

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Кафедра інформатики та програмної інженерії
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ Едуард ЖАРІКОВ
(підпис) (ім'я прізвище)

“ _____ ” _____ 2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інженерія програмного забезпечення
інформаційних систем»

спеціальності «121 Інженерія програмного забезпечення»

на тему: Імплементация API для платформы Schedule KPI

Виконав студентка IV курсу, групи IT-92
(шифр групи)

Бойко Єлизавета Костянтинівна
(прізвище, ім'я, по батькові)

(підпис)

Керівник доцент, к.т.н., доц., Фіногенов О. Д.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант
з графічної
документації ст. викл. Вітковська І. І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент доцент, к.т.н., доц., Філіпова М. В.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ –2023

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення
інформаційних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

Едуард ЖАРІКОВ
(ім'я прізвище)

(підпис)

“ ” _____ 2023 р.

ЗАВДАННЯ
на дипломний проєкт студенту

Бойко Єлизаветі Костянтинівні

(прізвище, ім'я, по батькові)

1. Тема проєкту Імплементація API для платформи Schedule KPI

керівник проєкту Фіногенов Олексій Дмитрович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «31» травня 2023 р. №2101-с

2. Термін подання студентом проєкту « 17 » червня 2023 року

3. Вихідні дані до проєкту: технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: загальні положення, основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі.

2) Моделювання та конструювання програмного забезпечення: опис структури бази даних, опис архітектури застосунку та його моделювання.

3) Аналіз якості та тестування програмного забезпечення.

4) Впровадження та супровід програмного забезпечення: розгортання та підтримка програмного забезпечення.

5. Перелік графічного матеріалу

1) Схема структурна варіантів використань _____

2) Схема структурна бізнес-процесу _____

3) Схема бази даних _____

4) Схема структурна класів програмного забезпечення _____

5) Креслення вигляду екранних форм _____

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «10» березня 2023 року _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення рекомендованої літератури	10.03.2023	
2	Розробка технічного завдання	03.03.2023	
3	Аналіз існуючих методів розв'язання задачі	19.03.2023	
4	Проектування структури та компонентів програмного забезпечення	30.03.2023	
5	Розробка програмного забезпечення	05.04.2023	
6	Тестування програмного забезпечення	10.04.2023	
7	Написання матеріалів текстової частини розробки	14.04.2023	
8	Написання матеріалів графічної частини розробки	20.04.2023	
9	Оформлення технічної документації проєкту	14.04.2023	
10	Подання ДП на попередній захист	08.05.2023	
11	Подання ДП рецензенту		
12	Подання ДП на основний захист		

Студент

(підпис)

Єлизавета БОЙКО

(ініціали, прізвище)

Керівник

(підпис)

Олексій ФІНОГЕНОВ

(ініціали, прізвище)

АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з чотирьох розділів, містить 30 таблиць, 26 рисунків та 5 джерел – загалом 65 сторінок.

Дипломний проєкт присвячений розробці системи розкладу занять для інтеграції з єдиним інформаційним середовищем університету, з можливістю замінити застарілу імплементацію платформи та адаптувати її для коректної роботи в умовах великих навантажень, помилок оновлення даних системи або перебоїв електропостачання.

Мета: виявлення програмного рішення для заміни минулої реалізації розкладу занять КПІ ім. Ігоря Сікорського з використанням оновленої технологічної бази відповідно до всіх сучасних тенденцій розробки веб-додатків, інтегроване з єдиним середовищем університету, налаштованим автоматичним розгортанням системи в хмарі та адаптоване для роботи в нестабільних умовах.

Об'єкт дослідження: організація навчального процесу університету.

Предмет дослідження: програмне забезпечення для надання розкладу занять та сесії здобувачам освітньої програми та викладачам КПІ ім. Ігоря Сікорського.

У першому розділі «Аналіз вимог до програмного забезпечення» надано детальний опис предметної області, що стосується розробки, а також визначено основні терміни та поняття, які використовуються під час реалізації проєкту. Було проаналізовано відомі технічні рішення, засоби розробки та відомі програмні продукти і обрано найбільш ефективні та корисні для даного продукту.

У другому розділі «Моделювання та конструювання програмного забезпечення» проведено детальний аналіз та дослідження різних аспектів програмного забезпечення. Розглянуто моделювання та аналіз програмного забезпечення, архітектуру програмного забезпечення, принципи роботи застосунку розкладу занять та конструювання програмного забезпечення.

У третьому розділі «Аналіз якості та тестування програмного забезпечення» було проведено аналіз якості та тестування програмного забезпечення розкладу занять університету.

У четвертому розділі «Впровадження та супровід програмного забезпечення» було описано процес розгортання та підтримки програмного забезпечення.

Програмне забезпечення впроваджено у єдине інформаційне середовище КПІ ім. Ігоря Сікорського.

КЛЮЧОВІ СЛОВА: СЕРВІС, INTELLIJ IDEA, RIDER, ВЕБ-ЗАСТОСУНОК, БАЗА ДАНИХ, СТАТИСТИКА, АНАЛІЗ.

ABSTRACT

The explanatory note of the diploma project consists of four sections, contains 30 tables, 26 figures and 5 sources – in total 65 pages.

The diploma project is devoted to the development of a classes schedule system for integration with the unified information environment of the university, with the possibility of replacing the outdated implementation of the platform and adapting it for correct operating in conditions of heavy loads, system data update errors or power outages.

Purpose: to identify a software solution to replace the past implementation of the class schedule of Igor Sikorsky Kyiv Politechnical Institute using an updated technological base in accordance with all modern trends in the development of web applications, integrated with the university's unified environment, configured by automatic deployment of the system in the cloud and adapted to work in unstable conditions.

Object of research: organization of the educational process of the university. Object of research: organization of the educational process of the university.

The subject of the study: software for providing the schedule of classes and exams to students of the educational program and teachers of Igor Sikorsky Kyiv Politechnical Institute.

The first section, "Analysis of software requirements", provides a detailed description of the subject area related to the development, as well as defines the main terms and concepts used during the implementation of the project. Known technical solutions, development tools and known software products were analyzed and the most effective and useful for this product were selected.

In the second chapter, "Software Modeling and Design", a detailed analysis and study of various aspects of the software is carried out. Software modeling and analysis, software architecture, operating principles of a class schedule application, and software design are considered.

In the third section, "Quality Analysis and Software Testing", the quality analysis and testing of the university's schedule software was conducted.

The fourth chapter, "Software Implementation and Support," described the process of software deployment and support.

The software is implemented in the unified information environment of Igor Sikorsky Polytechnical Institute.

KEYWORDS: SERVICE, INTELLIJ IDEA, RIDER, WEB APPLICATION, DATABASE, STATISTICS, ANALYSIS.

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ” _____ 2023 р.

ІМПЛЕМЕНТАЦІЯ АРІ ДЛЯ ПЛАТФОРМИ SCHEDULE KPI

Технічне завдання

КПІ.ІТ-9203.045440.01.91

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

Нормоконтроль:

_____ Ірина ВІТКОВСЬКА

Виконавець:

_____ Єлизавета БОЙКО

Київ – 2023

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
4.1	Вимоги до функціональних характеристик	6
4.1.1	Користувацького інтерфейсу	6
4.1.2	Для користувача:.....	15
4.1.3	Додаткові вимоги:	15
4.2	Вимоги до надійності.....	15
4.3	Умови експлуатації	16
4.3.1	Вид обслуговування.....	16
4.3.2	Обслуговуючий персонал.....	16
4.4	Вимоги до складу і параметрів технічних засобів	16
4.5	Вимоги до інформаційної та програмної сумісності	16
4.5.1	Вимоги до вхідних даних	17
4.5.2	Вимоги до вихідних даних	17
4.5.3	Вимоги до мови розробки	17
4.5.4	Вимоги до середовища розробки.....	17
4.5.5	Вимоги до представленню вихідних кодів.....	17
4.6	Вимоги до маркування та пакування.....	17
4.7	Вимоги до транспортування та зберігання	17
4.8	Спеціальні вимоги	18
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	19
5.1	Попередній склад програмної документації.....	19
5.2	Спеціальні вимоги до програмної документації	19
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	20
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ	21

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: ІМПЛЕМЕНТАЦІЯ АРІ ДЛЯ ПЛАТФОРМИ SCHEDULE КРІ.

Галузь застосування:

Наведене технічне завдання поширюється на розробку програмного забезпечення АРІ для платформи Schedule КРІ ІІ-9203.045440, котра використовується для надання платформи з зрозумілим користувацьким інтерфейсом, сучасною технологічною базою, налаштованим автоматичним оновленням та інтегровану з єдиним інформаційним середовищем університету та призначена для надання можливості перегляду розкладу занять та сесії здобувачів освітньої послуги та викладачів КПІ ім. Ігоря Сікорського.

					КПІ.ІТ-9203.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки імплементації API для платформи Schedule KPI є завдання на дипломне проектування, затверджене кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

					КПІ.ІТ-9203.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для інтеграції з єдиним інформаційним середовищем університету, призначене замінити застарілу імплементацію платформи розкладу занять та адаптоване для коректної роботи в умовах великих навантажень, помилок оновлення даних системи або перебоїв електропостачання.

Метою розробки є надання програмного забезпечення, яке призначене замінити минулу реалізацію розкладу занять КПІ ім. Ігоря Сікорського з використанням оновленої технологічної бази відповідно до всіх сучасних тенденцій розробки веб-додатків, інтегровану з єдиним середовищем університету, налаштованим автоматичним розгортанням системи в хмарі та адаптоване для роботи в нестабільних умовах.

					КПІ.ІТ-9203.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

Випадаючий список пошуку груп

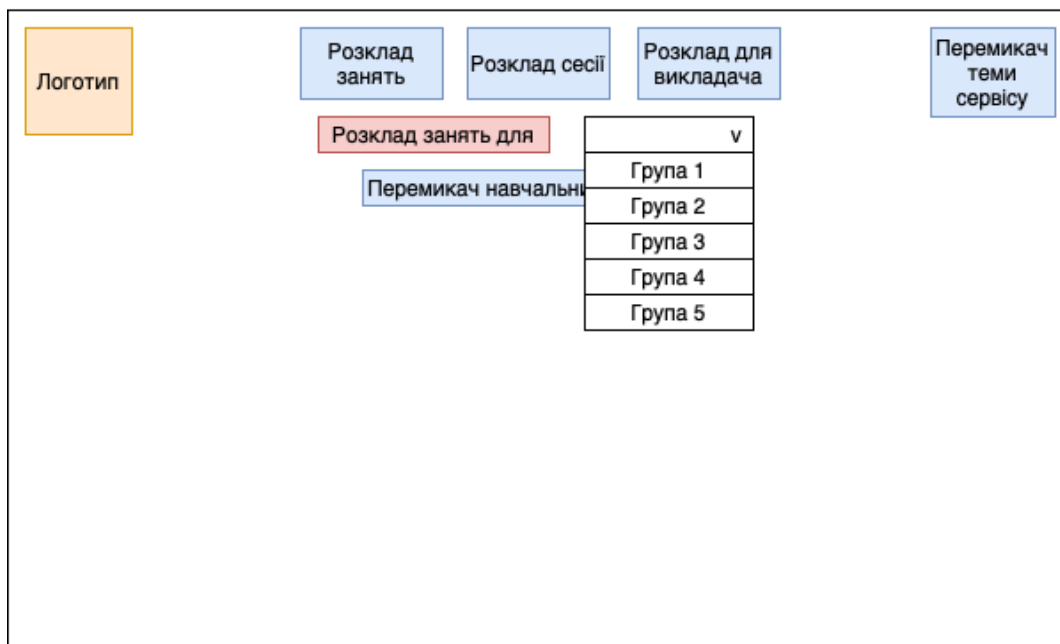


Рисунок 4.2 – Шаблон списку актуальних груп університету

– При виборі будь-якої групи зі списку та натисканні на її назву відобразиться розклад занять. Шаблон розкладу занять можна побачити нижче (рис. 4.3).

Розклад занять групи

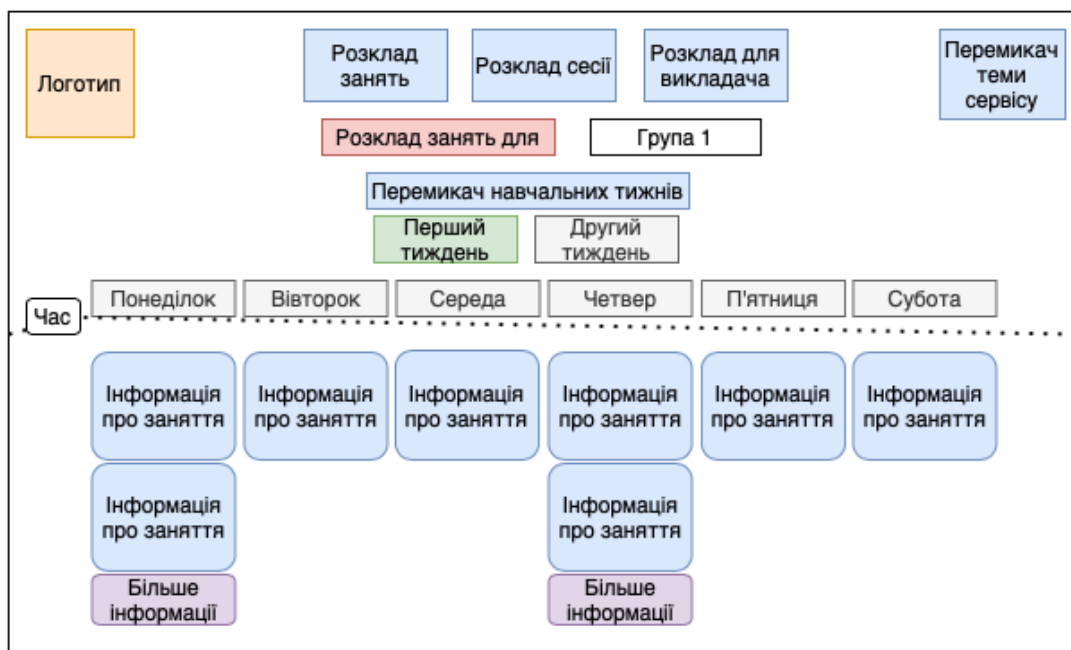


Рисунок 4.3 – Шаблон розкладу занять групи

– При натисканні на опції «Перший тиждень» або «Другий тиждень» можна переглянути розклад занять за обидва навчальні тижні, як продемонстровано на шаблоні (рис. 4.4).

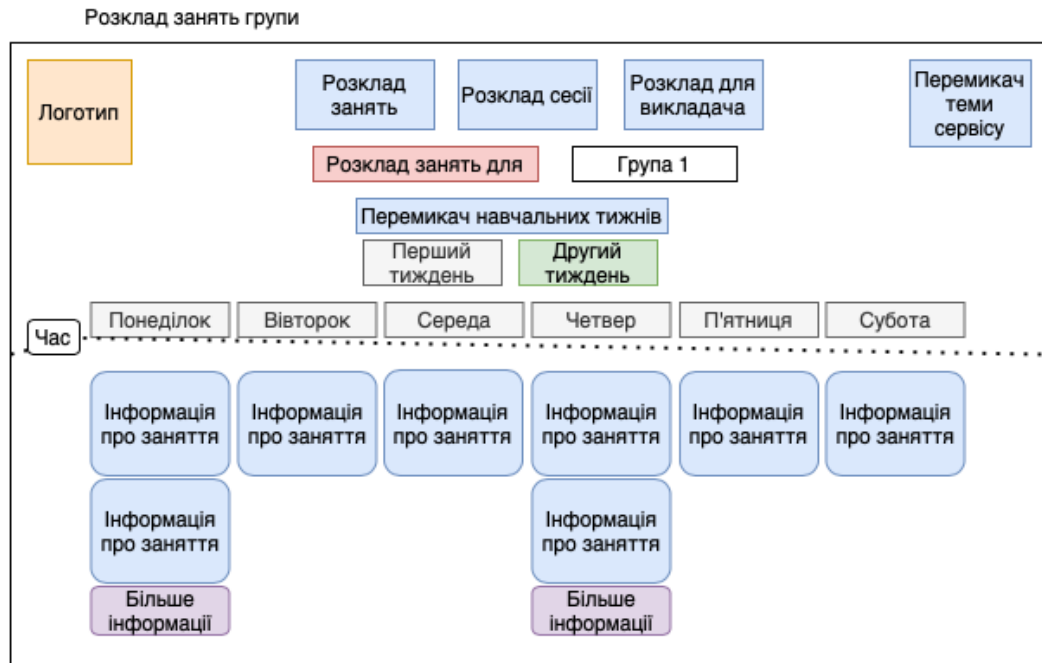


Рисунок 4.4 – Шаблон розкладу занять за інший тиждень

– При натисканні на опцію «Більше інформації», що знаходиться під переліком занять на день відобразиться повна інформація про заняття (рис. 4.5).

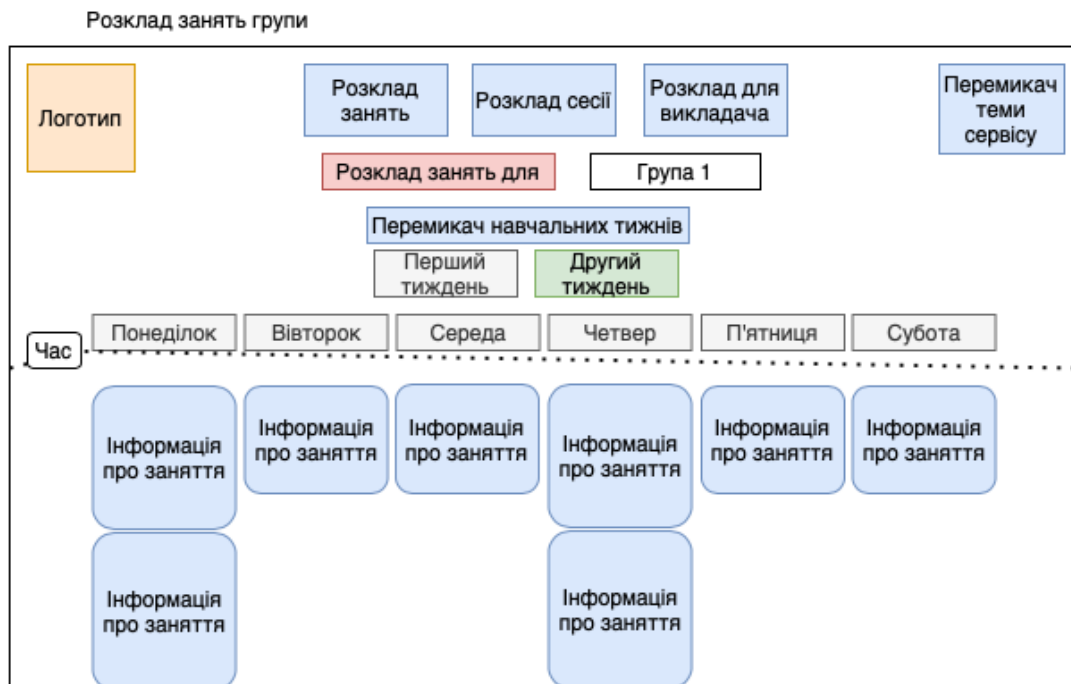


Рисунок 4.5 – Шаблон повної інформації розкладу занять групи

– При натисканні на опцію «Розклад сесії» відкриється сторінка розкладу сесії обраної групи (рис. 4.6).



Рисунок 4.6 – Шаблон розкладу сесії групи

– При вводиті назви групи в пошук, що знаходиться біля поля «Розклад занять для» відобразиться список в якому поступово будуть відображатися знайдені групи, що відповідають введеній назві. Такий самий пошук можна застосовувати і для пошуку розкладу занять. (рис. 4.7).

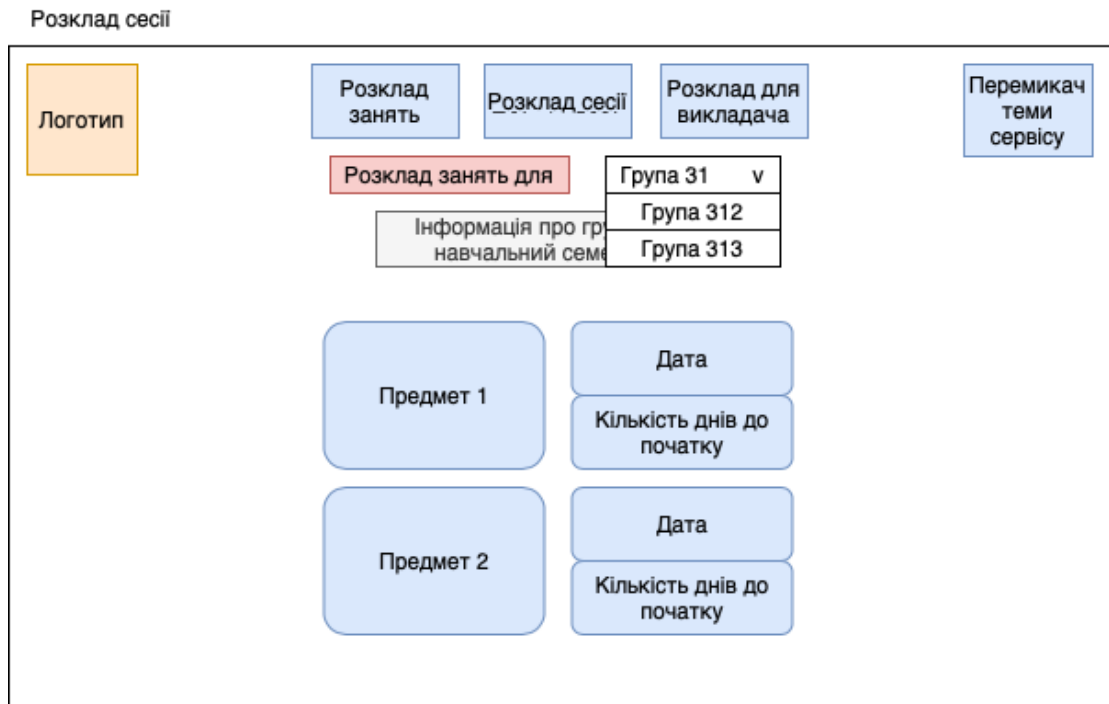


Рисунок 4.7 – Шаблон пошуку групи

– При натисканні на назву групи у випадяючому списку відобразиться її розклад сесії (рис. 4.8).

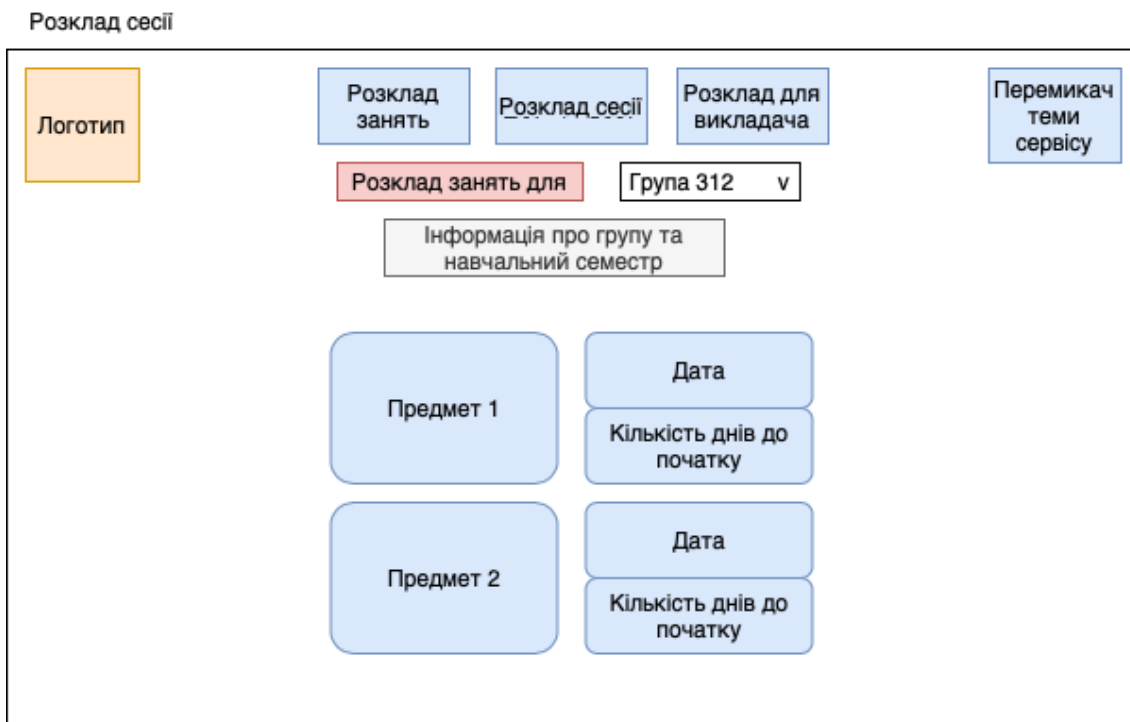


Рисунок 4.8 – Шаблон розкладу сесії шуканої групи

– При натисканні на опцію «Розклад для викладачів», відкриється сторінка для пошуку розкладу викладачів (рис. 4.9).

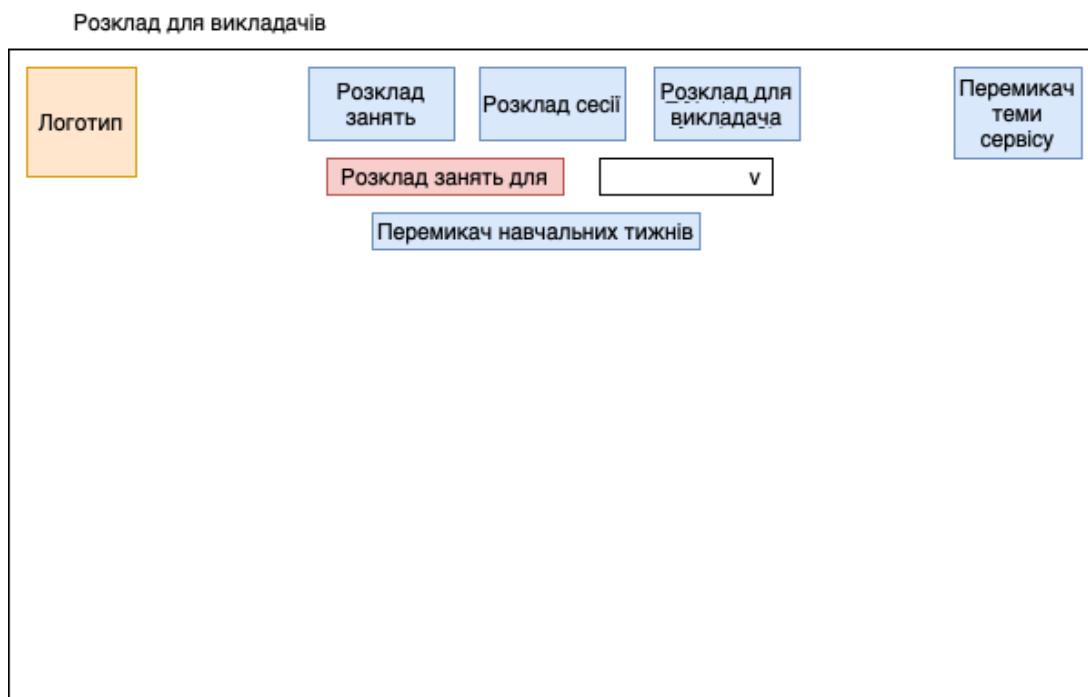


Рисунок 4.9 – Шаблон сторінки для пошуку розкладу викладача

– При натисканні на поле біля «Розклад занять для» відобразиться список актуальних викладачів університету, розклад яких можна передивитись (рис. 4.10).

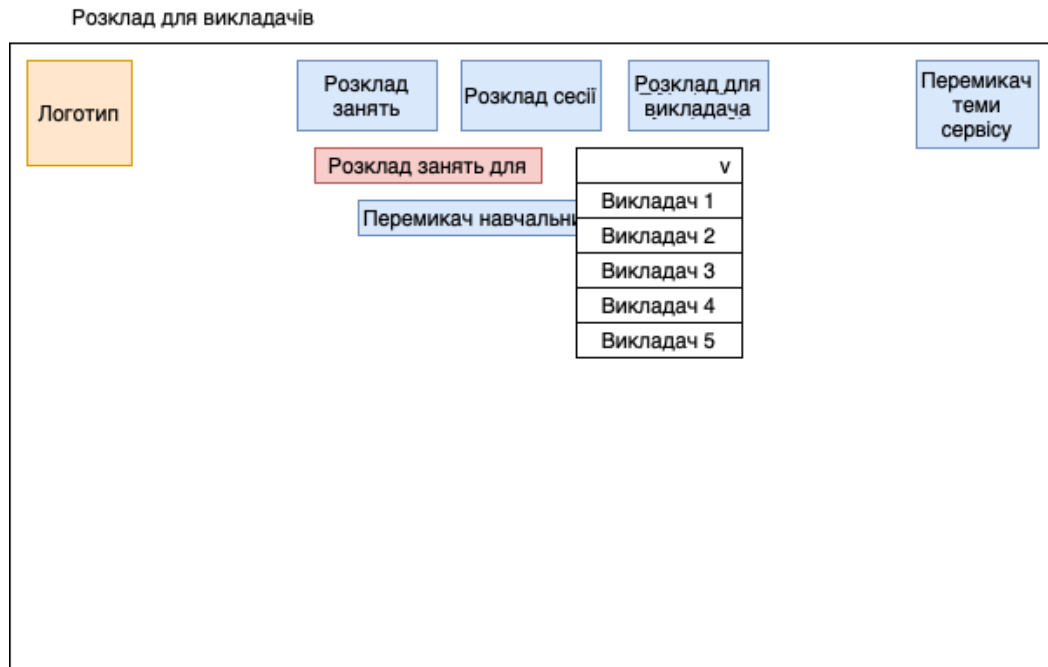


Рисунок 4.10 – Шаблон списку актуальних викладачів університету

– При натисканні на одного з викладачів зі списку, відобразиться розклад його занять (рис. 4.11).



Рисунок 4.11 – Шаблон розкладу викладача

– При натисканні на поле після поля «Розклад занять для» можна скористатися пошуком. Під час введення назви, відобразиться список в якому поступово будуть відображатися знайдені викладачі, що відповідають введеному ПІБ.(рис. 4.12).

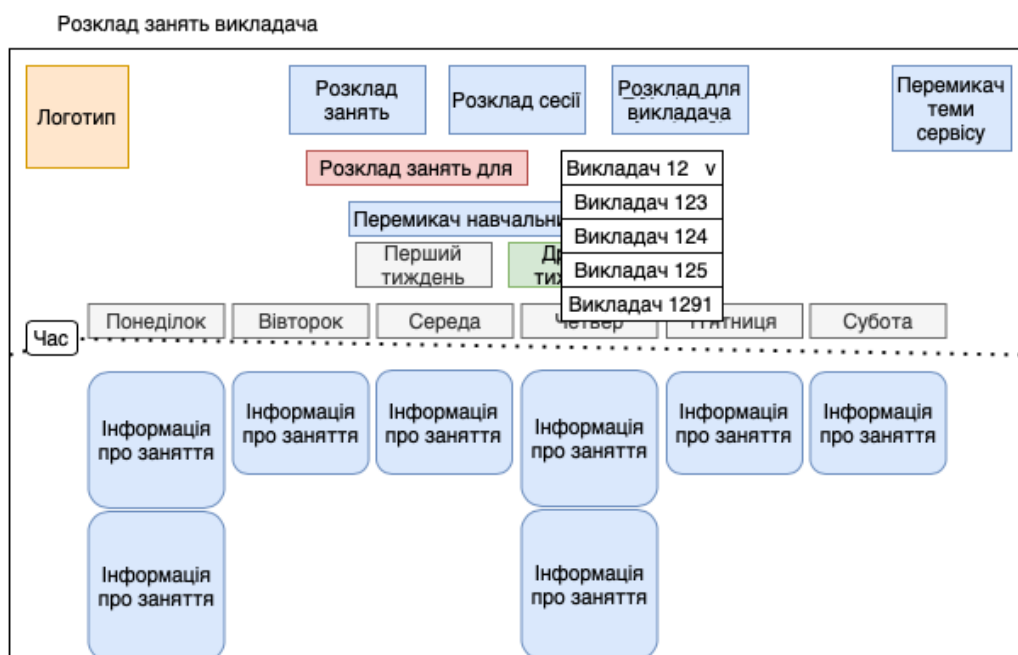


Рисунок 4.12 – Шаблон пошуку викладача

– При натисканні на опції «Перший тиждень» або «Другий тиждень» під час перегляду розкладу занять викладача, можна побачити розклад занять за обидва навчальні тижні (рис. 4.13).

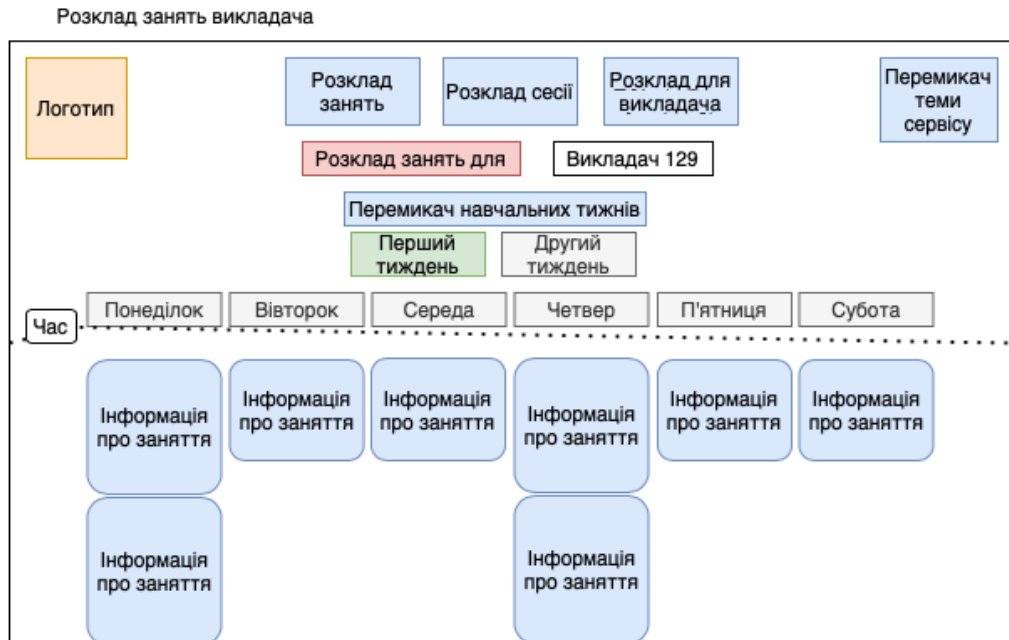


Рисунок 4.13 – Шаблон розкладу за перший тиждень

– При натисканні на назву групи, що вказана в інформації про заняття з розкладу викладача, відкриється вікно розкладу цієї групи (рис. 4.5). Крім того, при натисканні на ПІБ викладача, що вказано в інформації про заняття з розкладу групи, відкриється вікно розкладу цього викладача, як вказано на шаблоні(рис. 4.13).

– При натисканні на перемикач в правому куту екрану, тема додатку зміниться на світлу, якщо поточна є темною або на темну, якщо поточна є світлою (рис. 4.14).

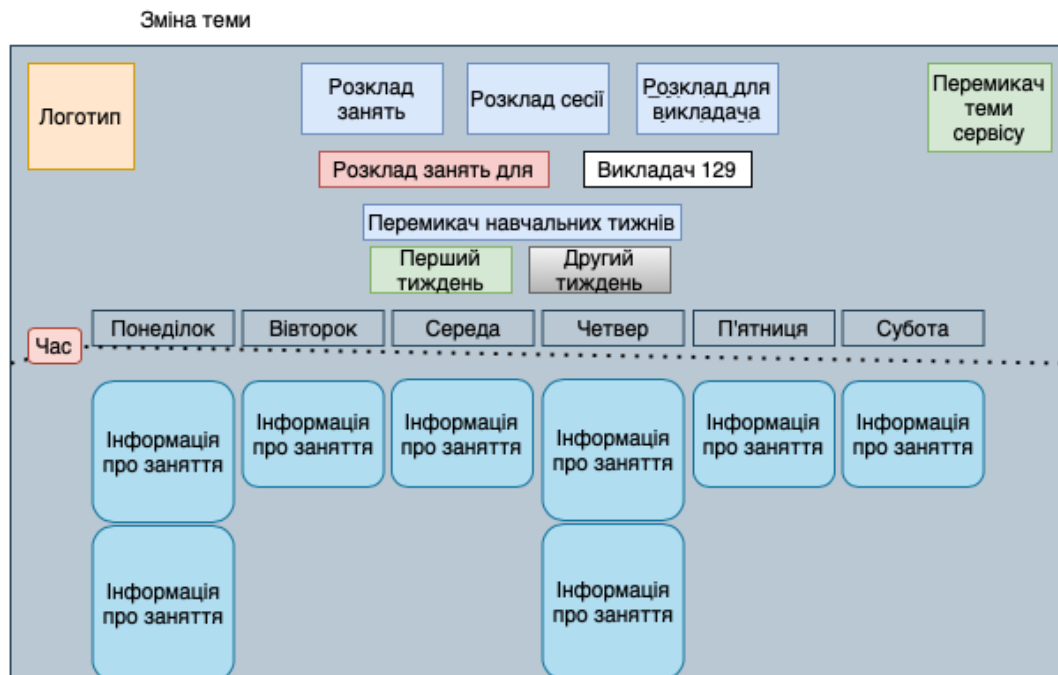


Рисунок 4.14 – Шаблон темної теми сервісу

4.1.2 Для користувача:

- забезпечити можливість перегляду розкладу занять групи;
- забезпечити можливість перегляду розкладу сесії групи;
- забезпечити можливість перегляду розкладу занять викладачів;
- забезпечити можливість зміни теми застосунку.

4.1.3 Додаткові вимоги:

- забезпечити можливість інтеграції системи з єдиним інформаційним середовищем університету;
- забезпечити можливість полегшеної підтримки веб-застосунку;
- забезпечити можливість заміни старого розкладу занять КПП ім. Ігоря Сікорського.

4.2 Вимоги до надійності

Передбачити контроль введення інформації та захист від некоректних дій користувача. Забезпечити цілісність інформації в базі даних. Забезпечити

коректну роботу системи під час збоїв оновлення сервісу або проблем з електропостачанням.

4.3 Умови експлуатації

Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.3.1 Вид обслуговування

Вимоги до виду обслуговування не висуваються.

4.3.2 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не висуваються.

4.4 Вимоги до складу і параметрів технічних засобів

Програмне забезпечення повинно функціонувати на комп'ютерах та телефонних пристроях.

Мінімальна конфігурація технічних засобів:

- тип процесору: Intel Core i5;
- об'єм ОЗП: 4 Гб;
- підключення до мережі Інтернет зі швидкістю від 20 мегабіт;

Рекомендована конфігурація технічних засобів < (та на якій виконувалась розробка) >:

- тип процесору: Intel Core i7;
- об'єм ОЗП: 8 Гб;
- підключення до мережі Інтернет зі швидкістю від 100 мегабіт

4.5 Вимоги до інформаційної та програмної сумісності

Програмне забезпечення повинно бути кросплатформеним.

4.5.1 Вимоги до вхідних даних

Обмін між серверною частиною та користувацьким інтерфейсом відбувається за допомогою JSON файлів, тому вхідні дані повинні бути представлені у форматі .json.

4.5.2 Вимоги до вихідних даних

Обмін між серверною частиною та користувацьким інтерфейсом відбувається за допомогою JSON файлів, тому вихідні дані повинні бути представлені у форматі .json, відповідно.

4.5.3 Вимоги до мови розробки

Користувацький інтерфейс розробки виконати на мові програмування JavaScript, а серверну частину на мові C#.

4.5.4 Вимоги до середовища розробки

Розробку виконати на платформах IntelliJ IDEA та Rider.

4.5.5 Вимоги до представлення вихідних кодів

Вихідний код програми має бути представлений у вигляді файлів з розширеннями .js, .cs та .jsx.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

					КПІ.ІТ-9203.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		17

4.8 Спеціальні вимоги

Згенерувати інсталяційну версію програмного забезпечення для її розгортання в хмарі.

					КПІ.ІТ-9203.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		18

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Попередній склад програмної документації

У склад супроводжувальної документації повинні входити наступні документи на аркушах формату А4:

- пояснювальна записка;
- технічне завдання;
- текст програми;
- програма та методика тестування;
- керівництво користувача;
- керівництво програміста;
- керівництво системного програміста;
- керівництво адміністратора.

Графічна частина повинна бути виконана на аркушах формату А3 та містити наступні документи:

- схема структурна варіантів використання;
- схема бази даних;
- креслення вигляду екранних форм;
- схема бізнес-процесів системи.

5.2 Спеціальні вимоги до програмної документації

Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення літератури за тематикою проекту	21.02	
2.	Розробка технічного завдання	03.03	Технічне завдання
3.	Аналіз вимог та уточнення специфікацій	19.04	Специфікації програмного забезпечення
4.	Проектування структури програмного забезпечення, проектування компонентів	30.04	Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму)
5.	Програмна реалізація програмного забезпечення	05.05	Тексти програмного забезпечення
6.	Тестування програмного забезпечення	10.05	Тести, результати тестування
7.	Розробка матеріалів текстової частини проекту	14.05	Пояснювальна записка
8.	Розробка матеріалів графічної частини проекту	20.05	Графічний матеріал проекту
9.	Оформлення технічної документації проекту	29.05	Технічна документація

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІТ-9203.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		21

**Пояснювальна записка
до дипломного проєкту**

на тему: Імплементація API для платформи Schedule KPI

КПІ.ІТ-9203.045440.02.81

Київ – 2023

ЗМІСТ

ВСТУП 4

1	АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
1.1	Загальні положення	6
1.2	Змістовний опис і аналіз предметної області	6
1.3	Аналіз існуючих технологій та успішних ІТ-проектів	7
1.3.1	Аналіз відомих алгоритмічних та технічних рішень.....	7
1.3.2	Аналіз допоміжних програмних засобів та засобів розробки	9
1.3.3	Аналіз відомих програмних продуктів	11
1.4	Аналіз вимог до програмного забезпечення.....	13
1.4.1	Розроблення функціональних вимог	20
1.4.2	Розроблення вимог до UI	23
1.5	Постановка задачі.....	24
	Висновки до розділу	24
2	МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	25
2.1	Моделювання та аналіз програмного забезпечення.....	25
2.2	Архітектура програмного забезпечення.....	27
2.3	Конструювання програмного забезпечення.....	28
2.4	Аналіз безпеки даних	36
	Висновки до розділу	36
3	АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 38	
3.1	Аналіз якості ПЗ	38
3.2	Опис процесів тестування	39
3.3	Опис контрольного прикладу.....	45
	Висновки до розділу	57
4	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.	59
4.1	Розгортання програмного забезпечення.....	59
4.2	Підтримка програмного забезпечення	60
	Висновки до розділу	60
	ВИСНОВКИ	61
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64
	ДОДАТКИ	65

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IDE	– Integrated Development Environment – інтегроване середовище розробки.
API	– Application programming interface - прикладний програмний інтерфейс.
SDK	– Software development kit - набір розробки програмного забезпечення.
ПЗ	– Програмне забезпечення.
REST	– Representational state transfer - репрезентативний стан передачі.
ER	– Операційна система.
БД	– База даних.
SSL	– Secure Sockets Layer - рівень захищених сокетів.
IT	– Інформаційні технології.
UI	– User interface – інтерфейс користувача.

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

ВСТУП

Навчання має велике значення для розвитку та успішної адаптації кожної особи у суспільстві. Отримання освіти дозволяє людині здобувати знання та навички в різних сферах, тим самим відкриваючи більше можливостей у житті.

Задля безперервного та сприятливого засвоєння матеріалу здобувачами освітньої послуги, процес навчання має бути чітко спланованим та організованим в логічній послідовності. Саме для цього і існує розклад занять, що дозволяє налагодити та контролювати цей підхід.

Останні кілька років навчальні заклади України стикнулись з новими умовами роботи через карантин та воєнну ситуацію у країні. Такі обставини призвели до введення обмежувальних заходів та вимусили школи та університети перейти на дистанційне навчання, щоб забезпечити безпеку учасників навчального процесу.

Забезпечення правильної організації такого підходу має вирішальне значення для надання ефективного освітнього процесу. Потрібно бути впевненим, що всі здобувачі освітньої послуги мають доступ до навчальних матеріалів і організація онлайн занять виконана вірно.

Одним із найважливіших питань в налаштуванні такого навчання є централізована система для організації занять. На початковому етапі впровадження онлайн навчання в Україні виникали значні труднощі щодо доступу до необхідних матеріалів з предметів, посилань на заняття та розкладу. Студентам доводилося використовувати різноманітні соціальні мережі, навчальні платформи та додатки зв'язку, що призводило до заплутаності і хаотичності інформації, а також ускладнювало організацію навчального процесу. Крім того, виникали певні технічні труднощі, такі як недосконалість технологій, відсутність синхронізації з наявними онлайн-сервісами університету та недостатня готовність університетських систем до великих навантажень.

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

Тому розробка системи для онлайн розкладу занять дозволить структурувати час здобувачів вищої освіти та їх навчальний процес, що сприятиме більш ефективному використанню часу та дозволить більш усвідомлено планувати свою роботу та навчальні завдання. Шляхом попереднього планування та визначення часових меж для кожного предмету можна уникнути ситуацій, коли здобувачу освітньої послуги доводиться обирати між двома важливими предметами або стикатися з застарілістю розкладу. Це забезпечить більш гармонійний розподіл навчального навантаження та зменшить напругу, пов'язану з тимчасовими труднощами. Адже, онлайн-розклад може бути легко оновлений та змінений за необхідності.

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Розроблюване програмне забезпечення презентує собою комплексну систему, спрямовану на забезпечення онлайн розкладу занять для здобувачів освітньої послуги та викладачів. Цей програмний продукт заснований на першочерговій ідеї створити ефективний інструмент, який надає користувачам зручні та структуровані можливості для відстеження та організації їх розкладу занять. Проте поступово працюючи над проектом та заглиблюючись в первинну ідею, було зроблено висновок, що даний програмний продукт може нести куди більшу користь учасникам навчального процесу.

Наприклад, якщо розглядати повноцінний процес здобуття освіти, то можна визначити що навчальний рік поділяється на два навчальних семестри. А кожний семестр закінчується сесією, відповідно. Тому було вирішено також відвести частину системи на відображення дат сесії для здобувачів освітньої послуги. Адже, знання дати сесії на початку семестру зможе допомогти освітньому учаснику краще планувати свій час, підготуватися до іспитів та забезпечити більш ефективне використання ресурсів, що сприятиме успішному складанню екзаменів та досягненню навчальних цілей.

Розроблена платформа вирішує проблеми користувачів здобувачів освітньої послуги, які вирішили переглянути розклад своєї групи та сесії, а також викладачів, які скористаються системою для перегляду свого розкладу викладання.

1.2 Змістовний опис і аналіз предметної області

Програмний продукт являє собою платформу для розкладу занять КПІ ім. Ігоря Сікорського та передбачає користування однієї ролі – користувача системи.

Для розробки даного програмного продукту було виділено кілька аспектів які включають в себе вивчення основних вимог та потреб, пов'язаних з налаштуванням онлайн розкладу занять та визначенням основних

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

компонентів та функцій платформи. Платформа має надавати можливість відстежувати порядок занять для різних груп та факультетів. Включає визначення часу, дати, локації аудиторії, назви предмета та імені викладача, що його веде. З боку викладача, програмний продукт надає можливість відстеження часу, дати, локації аудиторії, назви предмета та назви групи, у якій проводитиметься заняття. Також, розклад може включати різні типи занять, такі як лекції, практичні заняття або лабораторні роботи. Система може надавати можливість гнучкого налаштування розкладу відповідно до особливостей навчального процесу університету. Також, може містити визначення періодів занять та сесії.

1.3 Аналіз існуючих технологій та успішних ІТ-проектів

Проаналізуємо відомі на сьогодні технічні рішення у даній області, що допоможуть у реалізації платформи розкладу занять. Далі будуть розглянуті допоміжні програмні засоби та готові програмні рішення.

1.3.1 Аналіз відомих алгоритмічних та технічних рішень

Для реалізації якісної платформи важливим питанням постає вибір мов програмування. Проаналізувавши найпоширеніші технічні рішення, у результаті було обрано C# для написання API та JavaScript для написання візуальної частини.

C# - це сучасна, об'єктно-орієнтована мова програмування, яка привертає увагу своєю безпекою типів. Вона є незамінним інструментом для розробки надійних та безпечних програм, які виконуються на платформі .NET. C# надає розробникам конструкції, спеціально розроблені для безпосередньої підтримки цих концепцій, що робить її щонайкращою мовою для створення та використання програмних компонентів. Завдяки своїм можливостям та екосистемі, C# забезпечує ефективний процес розробки і дозволяє розробникам зосередитись на створенні високоякісних рішень. Ця мова є популярним вибором для розробки різноманітних додатків, від веб-сайтів до мобільних і десктопних програм [1].

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

Було обрано саме цю мову програмування тому, що вона забезпечує гнучкість та легкість перенесення при розгортанні бекенд-додатків. Тобто, C# надає можливість розробляти програми на різних операційних системах, таких як Windows, Linux та macOS. А за допомогою поєднання платформи .NET та C#, надається широкий набір інструментів та бібліотек, які значно спрощують розробку програмних продуктів. В додаток до цього, спрощується інтеграція з іншими технологіями, включаючи підтримку взаємодії з базами даних, використання сервісів хмарних платформ, роботу з API та іншими системами.

Для реалізації користувацького інтерфейсу було обрано мову програмування JavaScript. JavaScript є однією з основних технологій Всесвітньої павутини, разом з HTML і CSS. Згідно з останніми статистичними даними на 2022 рік, приблизно 98% веб-сайтів використовують JavaScript на стороні клієнта для надання інтерактивності та функціональності веб-сторінок. Часто розробники включають сторонні бібліотеки JavaScript для полегшення розробки і розширення можливостей своїх додатків [2].

Також, JavaScript є мовою з великим екосистемою, яка включає багато стандартних функцій, фреймворків та бібліотек, що дозволяють розробникам швидко реалізовувати складні задачі і забезпечувати багатofункціональність своїх додатків.

Таким чином, вибір JavaScript для розробки користувацького інтерфейсу веб-застосунку є досить популярним рішенням, оскільки ця мова має широкі можливості та велику кількість ресурсів для використання потужних інструментів та готових рішень задля ефективною розробки програмного забезпечення.

Мова програмування JavaScript підтримується всіма сучасними браузерами та надає можливість створювати різноманітні веб-сторінки та програми. Ця мова є стандартом веб-розробки та надає потужні можливості для роботи, обробки подій та маніпуляції елементами сторінки. Також, ця мова підтримує безліч бібліотек, фреймворків та інших інструментів, які спрощують розробку та додають функціональності програмним продуктам.

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		8

та дієві інструменти розробки користувацького інтерфейсу для створення багатофункціональних та інтерактивних додатків;

- надає гнучку систему налаштування на основі середовища, що дозволяє легко налаштовувати додатки для різних середовищ, включаючи роботу в хмарному середовищі;

- має вбудовану систему впровадження залежностей - Dependency Injection, яка дозволяє ефективно керувати залежностями між компонентами додатку;

- має ефективний конвеєр HTTP запитів, що сприяє швидкому обробленню запитів та зменшенню навантаження на сервер;

- додатки, розроблені на ASP.NET Core, можуть бути розгорнуті на різних серверах, включаючи Nginx, Apache, Docker та інші платформи;

- надає можливість паралельного створення версій додатків, що дозволяє розробникам зберігати та підтримувати різні версії своїх додатків;

- має багатий набір функцій та інструментів, які спрощують розробку веб-додатків, включаючи розширення для популярних інтегрованих середовищ розробки та інші механізми для налагодження, тестування та впровадження [3].

А для розробки інтерфейсу користувача, було вирішено скористатись однією з найпопулярніших бібліотек JavaScript – React. React є безкоштовною бібліотекою інтерфейсу з відкритим вихідним кодом, розробленою на JavaScript, і призначеною для зручного створення інтерфейсів користувача з використанням підходу на основі компонентів. Його можна використовувати для розробки одно-сторінкових, мобільних або серверних додатків із такими фреймворками, як Next.js. Оскільки React займається лише інтерфейсом користувача та відтворенням компонентів, програми React часто покладаються на бібліотеки для маршрутизації та інших функцій на стороні клієнта [4].

Так як, ця бібліотека заснована на компонентній архітектурі, вона дозволяє розбивати інтерфейс користувача на незалежні елементи, які можна

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		10

використовувати кілька разів у різних частинах імплементації, що значно полегшує підтримку коду та збільшує його зрозумілість.

Також, React підтримує віртуальний DOM, що є концепцією програмування, яка дозволяє ефективно оновлювати тільки необхідні частини інтерфейсу користувача. В той час, зручний синтаксис React забезпечує ефективну та легку розробку платформи.

Для роботи над серверною частиною було використано IDE – Rider, а для маніпуляції фронтенд проектом було використано IntelliJ IDEA. Так як саме ці інтегровані середовища розробки виявились найбільш зручними для роботи з обраними технологіями та базою даних.

1.3.3 Аналіз відомих програмних продуктів

За приклади готових відомих програмних продуктів можна взяти такі додатки, як Outlook, Teams та Roz.kpi.ua.

Суть Outlook та Teams полягає в тому, щоб надати користувачам розклад їх онлайн зустрічей, відображених у календарі. Всі користувачі мають бути учасниками компанії з придбаною ліцензією Microsoft та здебільшого ці рішення орієнтовані на планування роботи виключно в онлайн режимі (рис. 1.1 – 1.2).

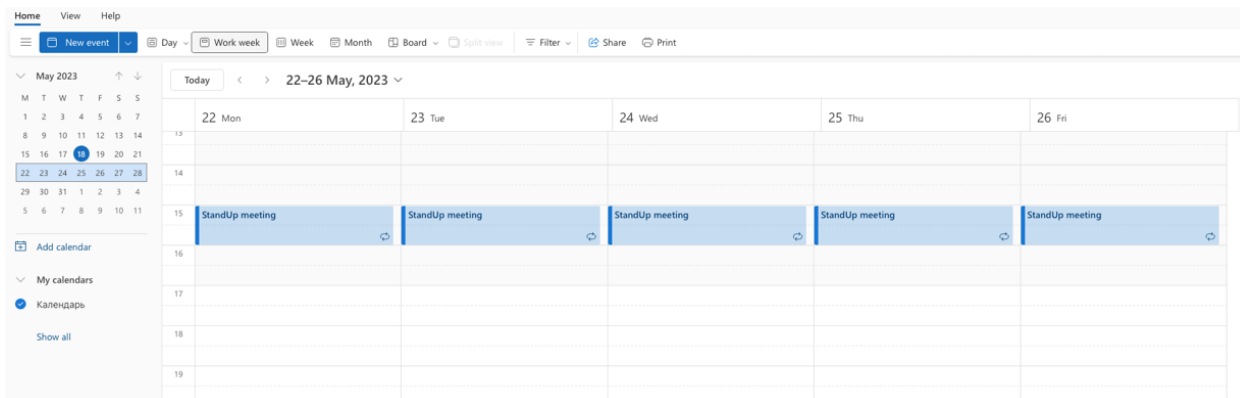


Рисунок 1.1 – Робота розкладу Microsoft Outlook

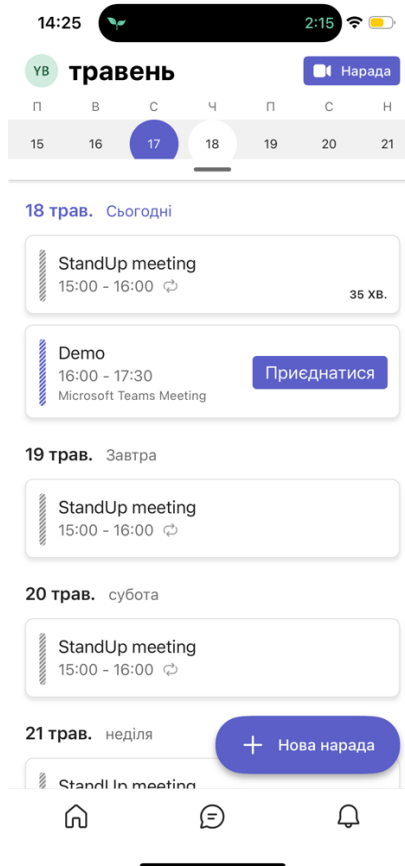


Рисунок 1.2 – Робота розкладу Microsoft Teams

Кажучи про Roz.kpi.ua, то ця платформа є минулою версією системи, що була імплементована в рамках даного дипломного проєкту. Перевагами такого додатку є те, що він є вузькоспеціалізованим, тобто він може використовуватися лише в рамках певного навчального закладу. Побачити роботу Roz.kpi.ua можна далі (рис. 1.3).

	Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота
1 08:30		Методи оптимізації та прийняття рішень пос. Журавовська О. С. -18 Лек on-line	Інформаційні системи та технології доц. Новітський В. П. -18 Лек on-line	Кибербезпека комп'ютерних мереж, Функційне програмування мовою Haskell, Управління виробництвом на базі застосування систем класу ERP, Основи науки про дані пос. Мironov Д. В., пос. Стативка Ю. І., доц. Новітський В. П., проф. Новозарський М. А. 212-18 Лаб on-line, 417-18 Лаб on-line, 422-18 Лаб on-line, -18 Лаб on-line		
2 10:25		Методи оптимізації та прийняття рішень пос. Журавовська О. С. -18 Лек on-line	Інформаційні системи та технології доц. Новітський В. П. -18 Лек on-line			
3 12:20		Мікропроцесорні технології інтернету речей пос. Стельмах О. П., доц. Жаріков Е. В. -18 Лек on-line				
4 14:15		Кибербезпека комп'ютерних мереж, Функційне програмування мовою Haskell, Управління виробництвом на базі застосування систем класу ERP, Алгоритми та математичні методи захисту інформації, Основи науки про дані, Системи автоматизованого проектування комп'ютерних систем.				Інформаційні системи та технології доц. Новітський В. П.

Рисунок 1.3 – Робота розкладу Roz.kpi.ua

Але програмний продукт розроблений в даній дипломній роботі є оновленою версією старої імплементації та має послужити їй заміною. Він включає в себе покращений та зрозуміліший інтерфейс користувача, поліпшену продуктивність системи, інтеграцію з єдиним середовищем університету та використання оновленої технологічної бази. Має налаштоване автоматичне оновлення сервісів при внесенні змін у кодову базу імплементації та їх автоматичне розгортання у хмарі. Також, спроектовану базу даних, що дозволяє підтримувати роботу веб-застосунку незважаючи на перебої та тимчасові відключення електроенергії.

1.4 Аналіз вимог до програмного забезпечення

Для створення use-case діаграми потрібно визначити акторів, що взаємодітимуть з програмним забезпеченням. Так як застосунок є веб-додатком для смартфонів та комп'ютерів, єдиним учасником системи є власник гаджету, тобто користувач.

Головною функцією програмного забезпечення є перегляд розкладу занять і екзаменів здобувачів освітньої послуги та розкладу викладачів (рис. 1.4).

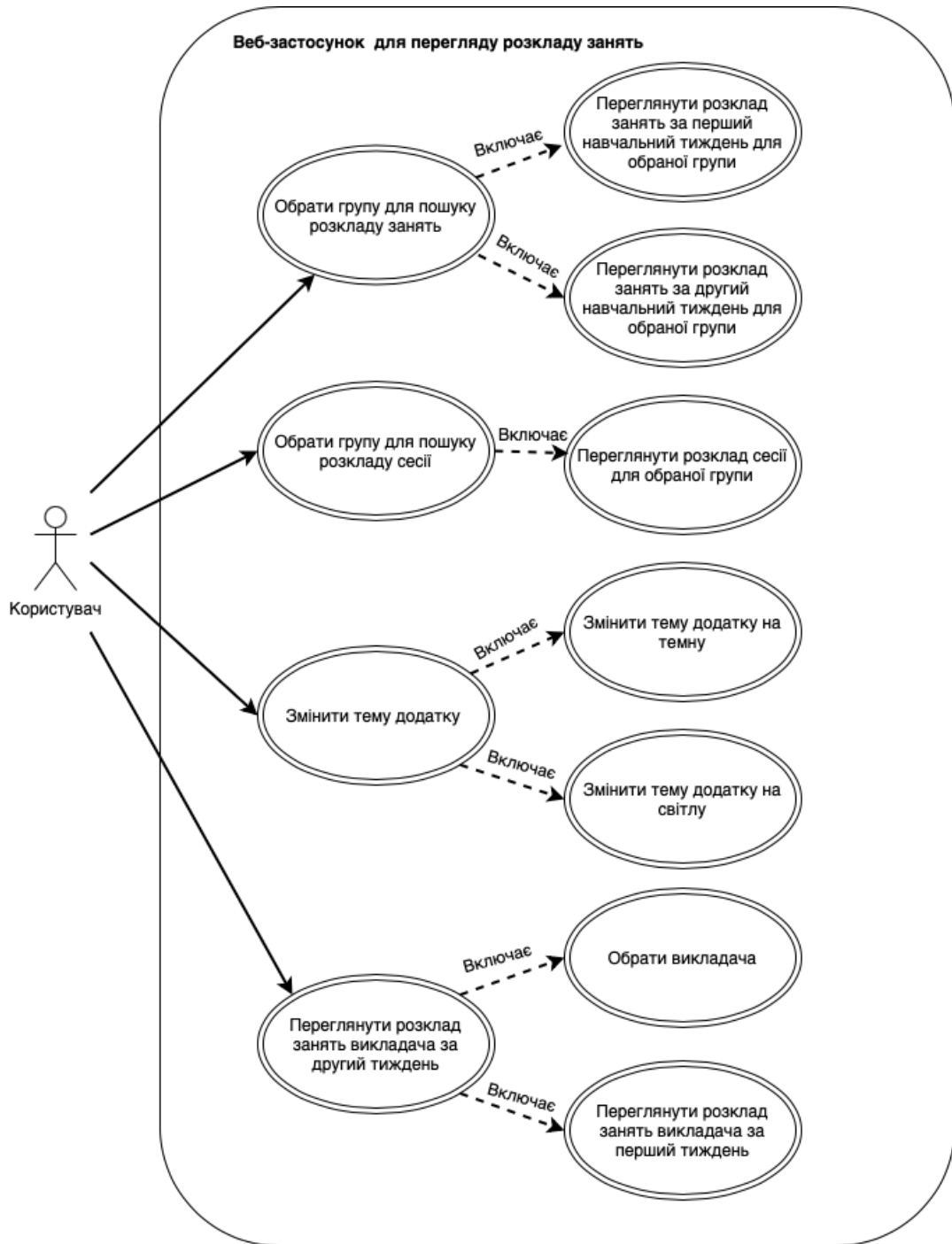


Рисунок 1.4 – Схема структурна варіантів використання

В таблицях 1.1 - 1.8 наведені варіанти використання програмного забезпечення.

Змін.	Арк.	№ докум.	Підп.	Дата.

Таблиця 1.1 - Варіант використання UC-1

Use case name	Вибір групи з списку
Use case ID	UC-01
Goals	Обрати групу для відображення розкладу занять
Actors	Гість (незарєєстрований користувач)
Trigger	Користувач бажає обрати групу для перегляду її розкладу
Pre-conditions	-
Flow of Events	Користувач заходить на сторінку веб-застосунку. В поле для групи вводяться відповідні дані: назва групи. Після заповнення даних, користувач натискає на назву групи в випадаючому списку. Після цього надсилається запит для пошуку розкладу занять обраної групи.
Extension	В випадку введення не коректних даних, випадаючий список не відображає доступних груп. Ніяких змін на сторінці не відбувається та запит не виконується.
Post-Condition	Перегляд розкладу занять для обраної групи.

Таблиця 1.2 - Варіант використання UC-2

Use case name	Перегляд розкладу занять для обраної групи
Use case ID	UC-02
Goals	Перегляд розкладу занять для обраної групи
Actors	Гість (незарєєстрований користувач)
Trigger	Користувач бажає переглянути розклад занять обраної групи
Pre-conditions	-

Продовження таблиці 1.2

Flow of Events	Користувач заходить на сторінку веб-застосунку. В випадковому списку з групами обирає бажану. Після цього надсилається запит для пошуку розкладу занять обраної групи та визначається навчальний тиждень. Знайдений план занять відображається на сторінці відповідно до поточного тижня.
Extension	В випадку введення не коректних даних, випадковий список не відображає доступних груп. Ніяких змін на сторінці не відбувається та запит не виконується.
Post-Condition	Перегляд розкладу сесії для обраної групи.

Таблиця 1.3 - Варіант використання UC-3

Use case name	Перегляд розкладу сесії для обраної групи
Use case ID	UC-03
Goals	Перегляд розкладу сесії для обраної групи
Actors	Гість (незарєєстрований користувач)
Trigger	Користувач бажає переглянути розклад сесії обраної групи
Pre-conditions	-
Flow of Events	Користувач заходить на сторінку веб-застосунку. Обирає опцію «Розклад сесії». В випадковому списку з групами обирає групу. Після цього надсилається запит для пошуку розкладу сесії обраної групи. Знайдений план сесії відображається на сторінці відповідно.
Extension	В випадку введення не коректних даних, випадковий список не відображає доступних груп. Ніяких змін на сторінці не відбувається та запит не виконується.
Post-Condition	Перегляд розкладу занять обраної групи.

Таблиця 1.4 - Варіант використання UC-4

Use case name	Вибір викладача з списку
Use case ID	UC-04
Goals	Обрати викладача для відображення розкладу занять
Actors	Гість (неzareєстрований користувач)
Trigger	Користувач бажає переглянути розклад сесії обраної групи
Pre-conditions	-
Flow of Events	Користувач заходить на сторінку веб-застосунку. Обирає опцію «Розклад для викладачів». В поле для пошуку вводяться відповідні дані: ПІБ викладача. Після заповнення даних, користувач натискає на ПІБ викладача в випадіючому списку. Після цього надсилається запит для пошуку розкладу занять обраного викладача.
Extension	В випадку введення не коректних даних, випадіючий список не відображає доступних викладачів. Ніяких змін на сторінці не відбувається та запит не виконується.
Post-Condition	Перегляд розкладу занять обраного викладача.

Таблиця 1.5 - Варіант використання UC-5

Use case name	Перегляд розкладу занять для обраного викладача
Use case ID	UC-05
Goals	Перегляд розкладу занять для обраного викладача
Actors	Гість (неzareєстрований користувач)
Trigger	Користувач бажає переглянути розклад занять обраного викладача
Pre-conditions	-

Продовження таблиці 1.5

Flow of Events	Користувач заходить на сторінку веб-застосунку та обирає опцію «Розклад для викладачів». В випадіючому списку з викладачами обирає викладача. Після цього надсилається запит для пошуку розкладу занять обраного викладача та визначається навчальний тиждень. Знайдений план занять відображається на сторінці відповідно до поточного тижня.
Extension	В випадку введення не коректних даних, випадіючий список не відображає доступних викладачів. Ніяких змін на сторінці не відбувається та запит не виконується.
Post-Condition	Перегляд розкладу занять групи за певний тиждень

Таблиця 1.6 - Варіант використання UC-6

Use case name	Перегляд розкладу занять групи за певний тиждень
Use case ID	UC-06
Goals	Перегляд розкладу занять групи для обраного тижня
Actors	Гість (незарєєстрований користувач)
Trigger	Користувач бажає переглянути розклад занять обраної групи за певний тиждень
Pre-conditions	-
Flow of Events	Користувач заходить на сторінку веб-застосунку та обирає опцію «Розклад занять». В випадіючому списку з групами обирає групу. Після цього надсилається запит для пошуку розкладу занять обраної групи та визначається навчальний тиждень. Знайдений план занять відображається на сторінці відповідно до поточного тижня. Після цього користувач може переглянути розклад занять за обидва тижні, натискаючи на опції «Перший тиждень» та «Другий тиждень».
Extension	В випадку введення не коректних даних, випадіючий список не відображає доступних груп. Ніяких змін на сторінці не відбувається та запит не виконується.

Продовження таблиці 1.6

Post-Condition	Перегляд розкладу занять викладачів за певний тиждень
----------------	---

Таблиця 1.7 - Варіант використання UC-7

Use case name	Перегляд розкладу занять викладачів за певний тиждень
Use case ID	UC-07
Goals	Перегляд розкладу занять викладача для обраного тижня
Actors	Гість (незарєєстрований користувач)
Trigger	Користувач бажає переглянути розклад занять обраного викладача за певний тиждень
Pre-conditions	-
Flow of Events	Користувач заходить на сторінку веб-застосунку та обирає опцію «Розклад для викладачів». В випадаючому списку з викладачами обирає вчителя. Після цього надсилається запит для пошуку розкладу занять обраного викладача та визначається навчальний тиждень. Знайдений план занять відображається на сторінці відповідно до поточного тижня. Після цього користувач може переглянути розклад занять за обидва тижні, натискаючи на опції «Перший тиждень» та «Другий тиждень».
Extension	В випадку введення не коректних даних, випадаючий список не відображає доступних викладачів. Ніяких змін на сторінці не відбувається та запит не виконується.
Post-Condition	Зміна теми веб-застосунку

Таблиця 1.8 - Варіант використання UC-8

Use case name	Зміна теми веб-застосунку
Use case ID	UC-08

Продовження таблиці 1.8

Goals	Перегляд розкладу занять у темній або у світлій темі
Actors	Гість (незарєєстрований користувач)
Trigger	Користувач бажає переглянути розклад занять у темній або у світлій темі
Pre-conditions	-
Flow of Events	Користувач заходить на сторінку веб-застосунку та натискає на перемикач в правому верхньому куту екрану. Після цього колір теми змінюється з темного в світлий або з світлого в темний, відповідно.
Extension	-
Post-Condition	-

1.4.1 Розроблення функціональних вимог

Програмне забезпечення розділене на модулі. Кожен модуль має свій певний набір функцій. Таблиці 1.9 - 1.16 містять опис функціональних вимог до програмного забезпечення.

Таблиця 1.9 – Загальна модель вимог

№	Назва функціональної вимоги	ID вимоги	Пріоритет	Ризики
1	Вибір групи	FR-1	1	Невизначений
2	Вибір тижня	FR-2	1	Невизначений
3	Вибір викладача	FR-3	1	Невизначений
4	Вибір теми веб-застосунку	FR-4	2	Невизначений

Продовження таблиці 1.9

5	Повернення розкладу занять відповідно до обраної групи	FR-5	1	Невизначений
6	Повернення розкладу занять відповідно до обраного викладача	FR-6	1	Невизначений
7	Повернення розкладу сесії відповідно до обраної групи	FR-7	2	Невизначений

Таблиця 1.10 – Функціональна вимога FR-1

Назва	Вибір групи
Опис	Система має забезпечувати користувачу можливість вибрати групу шляхом введення назви групи.

Таблиця 1.11 – Функціональна вимога FR-2

Назва	Вибір тижня
Опис	Система має забезпечувати можливість вибрати навчальний тиждень користувачеві шляхом натискання на опції «Перший тиждень» та «Другий тиждень».

Таблиця 1.12 – Функціональна вимога FR-3

Назва	Вибір викладача
Опис	Система має забезпечувати можливість вибрати викладача користувачеві шляхом введення його ПІБ.

Таблиця 1.13 – Функціональна вимога FR-4

Назва	Вибір теми веб-застосунку
Опис	Система має забезпечувати можливість вибору теми веб-застосунку користувачеві шляхом натискання на перемикач.

Таблиця 1.14 – Функціональна вимога FR-5

Назва	Повернення розкладу занять відповідно до обраної групи
Опис	Система має забезпечувати можливість перегляду розкладу занять користувачеві відповідно до обраної ним групи.

Таблиця 1.15 – Функціональна вимога FR-6

Назва	Повернення розкладу занять відповідно до обраного викладача
Опис	Система має забезпечувати можливість перегляду розкладу занять користувачеві відповідно до обраного ним викладача.

Таблиця 1.16 – Функціональна вимога FR-7

Назва	Повернення розкладу сесії відповідно до обраної групи
Опис	Система має забезпечувати можливість перегляду розкладу сесії користувачеві відповідно до обраної ним групи.

Наступним кроком було побудовано матрицю трасування вимог (табл. 1.17).

	FR-1	FR-2	FR-3	FR-4	FR-5	FR-6	FR-7
UC-1	+						
UC-2	+				+		
UC-3	+						+
UC-4			+				
UC-5			+			+	
UC-6	+	+					+
UC-7		+	+			+	
UC-8				+			

Таблиця 1.17 – Матриця трасування вимог

Виходячи з побудованої матриці трасування вимог, можна побачити відповідність між функціональними вимогами та варіантами використання.

1.4.2 Розроблення вимог до UI

Оскільки програмний продукт містить в собі різний функціонал, в процесі розробки потрібно створити:

- сторінку для відображення розкладу занять за групами;
- сторінку для відображення розкладу занять за викладачами;
- сторінку розкладу сесії.

Перелік вимог до UI:

- на кожній сторінці розмістити кнопки для переходу між опціями, для зміни теми, для вибору навчального тижня та поле для вводу розшукуваної групи чи викладача;

- перегляд плану занять має бути у вигляді таблиці з чітким позначенням часу занять та відображенням всіх полів шуканого предмету;
- поле пошуку викладачів та груп розробити у вигляді випадаючого списку;
- перемикач теми застосунку виконати у вигляді слайд бару;
- при натисканні на викладача у розкладі занять групи, переносити користувача на розклад обраного викладача.

1.5 Постановка задачі

Розробка призначена для вирішення проблеми організації дистанційного навчання, надаючи користувачам оновлений та легкий для розуміння веб-додаток розкладу занять, який є інтегрованим з онлайн сервісами університету. Програмний продукт надає можливість здобувачам освітньої послуги та викладачам вільно планувати свій робочий та навчальний час, а також забезпечує більш ефективне використання ресурсів, що сприятиме досягненню навчальних цілей.

Висновки до розділу

У цьому розділі надано детальний опис предметної області, що стосується розробки, а також визначено основні терміни та поняття, які використовуються під час реалізації проекту. Це дозволяє чітко уявити суть та контекст розробки, а також зазначити розуміння ключових термінів, які використовуються у процесі розробки програмного продукту. Також, було проаналізовано відомі технічні рішення, доступні допоміжні програмні засоби і засоби розробки та відомі програмні продукти і обрано найбільш ефективні та корисні для даного продукту. В додаток, було досліджено наявні на ринку аналоги. Зважаючи на виявлені недоліки та відмінності існуючих додатків, можна зробити висновок, що створення веб-додатку для розкладу КПІ ім. Ігоря Сікорського вирішує основну проблему організації навчального процесу під час дистанційного навчання в університеті, впроваджуючи новітню структуровану систему відстеження розпорядку занять для здобувачів вищої освіти та викладачів.

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Для моделювання бізнес-процесу програмного забезпечення використовується стандарт BPMN (Business Process Model and Notation). Ця модель дозволяє уявити послідовність дій, взаємозв'язки та взаємодії між різними елементами процесу (рис. 2.1).

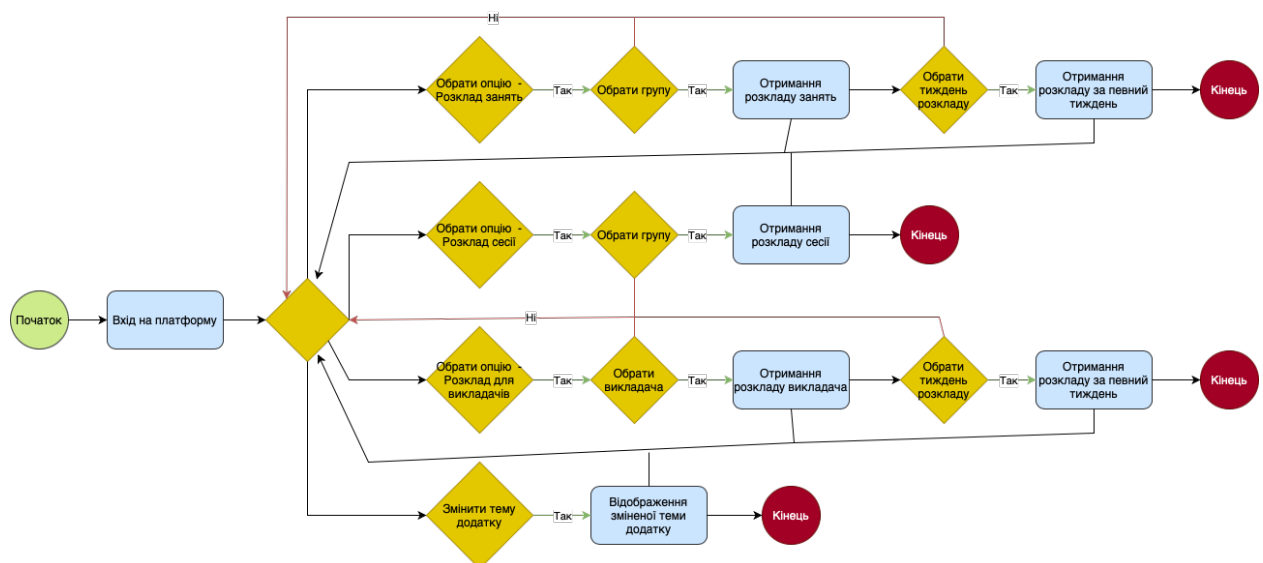


Рисунок 2.1 – Схема бізнес-процесу

Опис послідовності пошуку розкладу групи користувачем:

- користувач переходить на сторінку платформи;
- користувач натискає на опцію «Розклад занять»;
- користувач вводить назву групи у поле для пошуку;
- користувач обирає знайдену групу з випадуючого списку, натискаючи на її назву;
- якщо шуканої групи не існує, групу не буде знайдено у випадуючому списку, тому відповідно не буде знайдено розклад групи.

Опис послідовності пошуку розкладу викладача користувачем:

- користувач переходить на сторінку платформи;
- користувач натискає на опцію «Розклад для викладачів»;
- користувач вводить ПІБ викладача у поле для пошуку;
- користувач обирає знайденого викладача з випадаючого списку, натискаючи на її назву;

– якщо шуканого викладача не існує, викладача не буде знайдено у випадаючому списку, тому відповідно не буде знайдено розклад для нього.

Опис послідовності пошуку розкладу сесії користувачем:

- користувач переходить на сторінку платформи;
- користувач натискає на опцію «Розклад сесії»;
- користувач вводить назву групи у поле для пошуку;
- користувач обирає знайдену групу з випадаючого списку, натискаючи на її назву;

– якщо шуканої групи не існує, групу не буде знайдено у випадаючому списку, тому відповідно не буде знайдено розклад сесії.

Опис послідовності зміни тижня розкладу:

- користувач переходить на сторінку платформи;
- користувач натискає на опцію «Розклад занять» або «Розклад для викладача»;
- користувач вводить назву групи або ПІБ викладача у поле для пошуку;
- користувач обирає знайдену групу або викладача з випадаючого списку, натискаючи на знайдену опцію;
- користувач обирає та змінює тиждень відображуваного розкладу натискаючи на опції «Перший тиждень» та «Другий тиждень»;

– якщо шуканої групи або викладача не існує, дані не буде знайдено у випадаючому списку, тому відповідно не буде знайдено розклад.

Опис послідовності зміни теми веб-додатку:

- користувач переходить на сторінку платформи;

- користувач використовує перемикач теми, натискаючи на нього.

2.2 Архітектура програмного забезпечення

Головною задачею проєкту постало розроблення застосунку, який в подальшому замінить стару імплементацію розкладу занять КПІ ім. Ігоря Сікорського та буде інтегрований з єдиним середовищем університету. Для цього було вирішено додати імплементацію в закритий репозиторій КБ ІС, де зберігаються інші сервіси інфраструктури університету та розробити відокремлену базу даних для зберігання інформації потрібної для визначення розкладу користувачів застосунку.

Для розуміння структури розробленої БД, було створено ER діаграму, яка відображає основні сутності та їх взаємозв'язки використані в контексті даного проєкту (рис. 2.2).

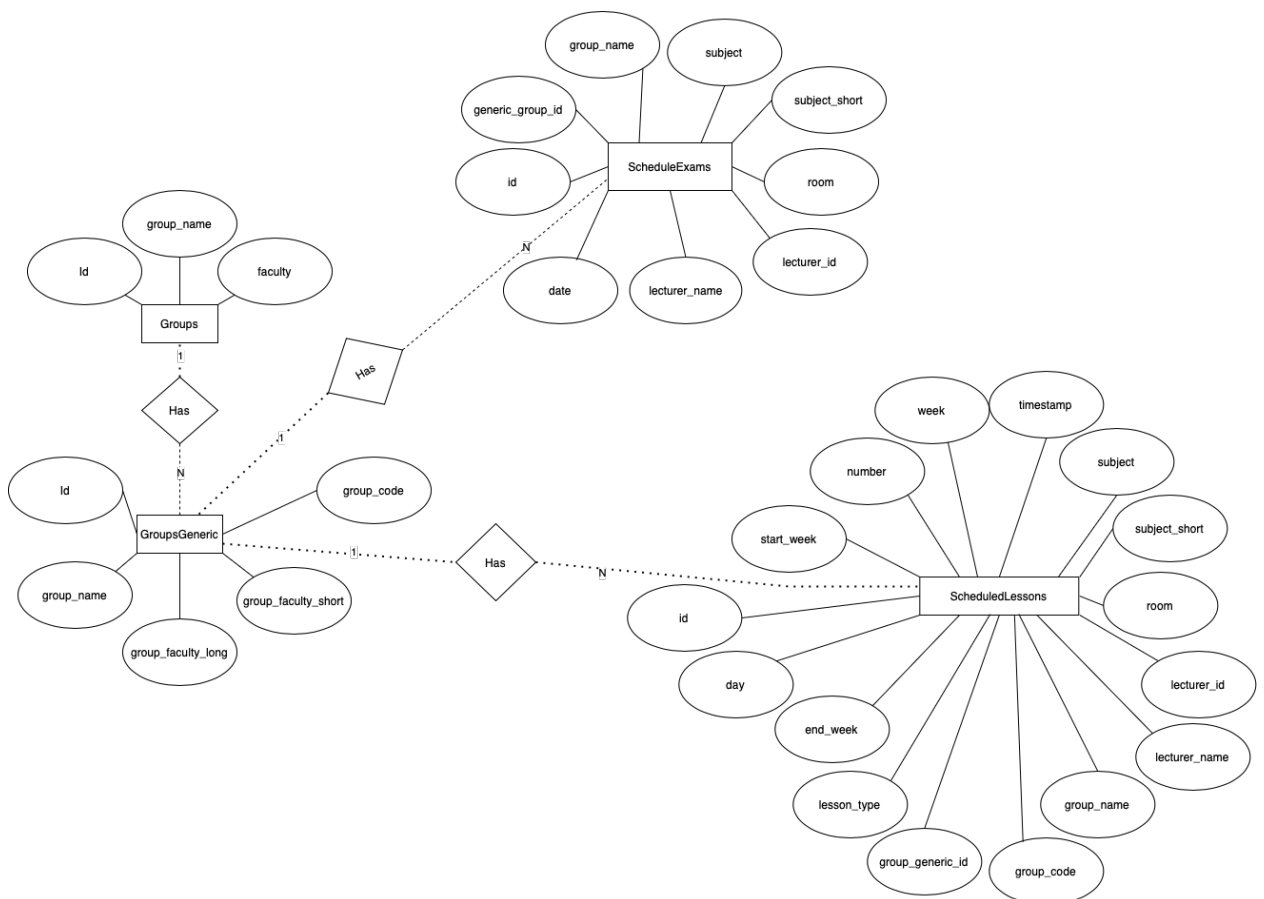


Рисунок 2.2 – ER діаграма сутностей

Виходячи з рисунку, можна побачити, що було виділено три основні сутності:

- сутність Groups представляє собою групу університету та містить в собі назву групи та факультету. Групи не є унікальною одиницею в межах університету та можуть мати однакові назви в рамках різних факультетів. Сутність груп має зв'язок з сутністю GroupsGeneric, що свідчить про те, що одній сутності групи може належати кілька сутностей GroupsGeneric;

- сутність GroupsGeneric представляє собою єдину унікальну сутність групи університету. Містить в собі унікальний ідентифікатор групи group_code, назву групи та інформацію про факультет. Так як ця сутність є унікальною, саме вона використовується для визначення групи та її розкладу;

- сутність ScheduleExams представляє собою сутність екзаменів та містить в собі код групи, інформацію про викладача, локацію екзамену та предмет. Сутність екзамену має зв'язок один до багатьох з сутністю групи, що свідчить про те, що одній сутності групи може належати кілька сутностей екзаменів;

- сутність ScheduledLessons представляє собою сутність занять та містить в собі інформацію про дату та час заняття, код групи, інформацію про викладача та предмет. Сутність заняття має зв'язок один до багатьох з сутністю групи, що свідчить про те, що одній сутності групі може належати кілька сутностей занять.

Кожна із таблиць містить всю необхідну інформацію для виконання запитів на пошук розкладу занять та сесії здобувачів освітньої послуги та розкладу викладачів. Додатково було використано таблиці для резервного копіювання даних з основних таблиць задля зберігання даних та їх використання під час перебоїв системи пов'язаних з великим навантаженням на додаток або перебоями електропостачання.

2.3 Конструювання програмного забезпечення

ASP.NET є потужною платформою для створення веб-додатків, особливо коли потребується робота з базою даних, аутентифікацією,

авторизацією та іншими функціями серверу. ASP.NET надає розширені можливості для роботи з даними, включаючи Object-Relational Mapping і можливість використання мови C# для програмування бізнес-логіки. Саме тому було вирішено обрати цей фреймворк для імплементації бекенд частини програмного продукту.

React JS є потужною JavaScript - бібліотекою для створення інтерфейсу користувача. Вона пропонує компонентний підхід до розробки, що дозволяє створювати окремі, перевикористовувані компоненти інтерфейсу користувача, що полегшує підтримку коду, поліпшує його читабельність та забезпечує більшу швидкість розробки.

Саме тому, комбінація ASP.NET та React JS може забезпечити високу продуктивність веб-застосунку. ASP.NET забезпечує швидку обробку запитів на серверному рівні та підтримує оптимізацію роботи з базою даних. А React JS, в свою чергу, дозволяє ефективно оновлювати лише необхідні компоненти на клієнтській стороні, що знижує навантаження на мережу та забезпечує швидке відображення змін.

Системою управління базами даних було обрано використовувати PostgreSQL. База даних призначена для зберігання і управління даними про розклад занять викладачів та здобувачів освітньої послуги, а також даними про їх сесії. Вона забезпечує надійне та ефективне зберігання і доступ до інформації, необхідної для функціонування системи (рис. 2.3).

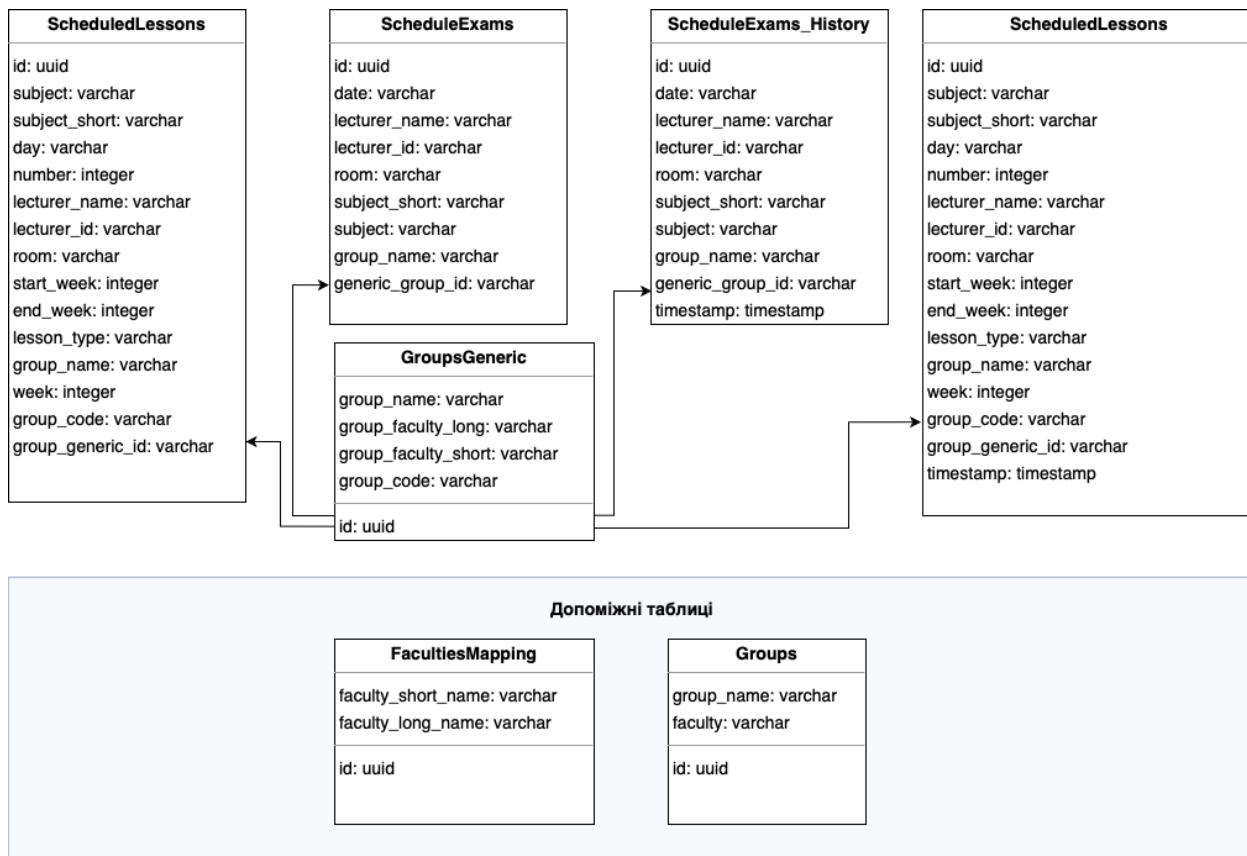


Рисунок 2.3 – Схема бази даних

Таблиці визначають структуру та поля даних, які зберігаються в базі даних. Кожна таблиця представляє певну сутність або аспект розкладу занять і містить відповідні поля, що описують цю сутність. Опис таблиць бази даних наведено у таблицях 2.1 - 2.10.

Таблиця 2.1 – Опис таблиці Groups

Поле	Тип даних	Опис
id	uuid	Універсальний унікальний ідентифікатор групи у форматі GUID - 128-розрядне ціле число, яке має дуже низьку ймовірність дублювання.
group_name	varchar	Назва групи
faculty	varchar	Назва факультету

Таблиця груп складається з назви групи та факультету. Завдяки такому комбінуванню даних, в БД зберігається унікальна група в межах певного факультету.

Таблиця 2.2 – Опис таблиці GroupsGeneric

Поле	Тип даних	Опис
id	uuid	Універсальний унікальний ідентифікатор групи у форматі GUID - 128-розрядне ціле число, яке має дуже низьку ймовірність дублювання.
group_name	varchar	Назва групи
group_faculty_long	varchar	Повна назва факультету групи
group_faculty_short	varchar	Скорочена назва факультету групи
group_code	varchar	Код групи у форматі GUID - 128-розрядне ціле число, яке має дуже низьку ймовірність дублювання.

Таблиця «GroupsGeneric» також містить інформацію про групи, але є доповненою інформацією про повну та скорочену назву факультету та унікальний ідентифікатор групи для визначення саме унікальних груп у рамках різних факультетів. За допомогою цієї таблиці і відбувається основний процес пошуку груп здобувачів освітньої послуги, розкладу їх плану занять та розкладу сесії.

Таблиця 2.3 – Опис таблиці ScheduledLessons

Поле	Тип даних	Опис
id	uuid	Універсальний унікальний ідентифікатор заняття у форматі GUID - 128-розрядне ціле число, яке має дуже низьку ймовірність дублювання.

2.4 Аналіз безпеки даних

Інформація про розклад занять зберігається в базі даних застосунку, і доступ до цієї бази має лише адміністратор системи, що забезпечує конфіденційність і захищеність даних від несанкціонованого доступу.

Система також автоматично створює резервні копії таблиць бази даних. Це дозволяє відновити дані у разі виникнення технічних проблем, таких як випадкове видалення або пошкодження даних або тимчасове відключення електроенергії. Резервні копії також зберігаються в тій самій базі даних з обмеженим доступом, що забезпечує збереження інформації та можливість відновлення даних.

Окрім цього, система забезпечує безпечний доступ до даних за допомогою надійних і перевірених мережних протоколів та SSL-сертифікатів, що дозволяють зашифрувати комунікацію між веб-сервером і користувачем та забезпечує захищене передавання і обробку даних між додатком та сервером бази даних. Адже, використання надійних мережних протоколів та сертифікатів гарантує конфіденційність та доступність даних у системі розкладу занять.

Висновки до розділу

У другому розділі проведено детальний аналіз та дослідження різних аспектів програмного забезпечення. Розглянуті наступні об'єкти: моделювання та аналіз програмного забезпечення, архітектура програмного забезпечення, принципи роботи застосунку розкладу занять та конструювання програмного забезпечення.

Побудування схеми бізнес-процесу допомогло візуалізувати послідовність дій між різним етапами роботи застосунку. За допомогою ER діаграми було визначено основні сутності системи та їх взаємозв'язки. Після чого було визначено структуру та логіку моделі БД.

Розглянуто основні принципи розробки програмного продукту, включаючи вибір утиліт та структуру програмного проєкту. Було проведено детальний огляд класів, методів та таблиць бази даних, які використовуються

в розроблюваній системі. Крім того, було проведено оцінку безпеки даних програмного продукту та розглянуті заходи для їх захисту, зокрема обмеження доступу до бази даних.

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		37

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

У даному підрозділі розглянемо аналіз якості програмного забезпечення за певними критеріями, що є процесом визначення і оцінки різних аспектів якості розробленого програмного продукту на основі певних вимірюваних параметрів та метрик. Цей аналіз допоможе виявити сильні та слабкі сторони програмного продукту, оцінити рівень його виконання, функціональності, ефективності, надійності та інших характеристик. Задля перевірки відповідності імплементації вимогам, виконаємо тестування системи відповідно до варіантів використання описаних у розділі 1.4. Наступним кроком буде перевірка відповідності результатів функціональним вимогам системи. Цей процес включає оцінку відповідності програмного продукту визначеним функціональним критеріям, які описують очікувану поведінку та можливості системи.

Для перевірки якості кодової бази програмного забезпечення було використано онлайн сервіс codescene.io [5]. Цей інструмент надає можливість аналізувати якість коду, визначати вразливості або шкідливий функціонал, покриття обробки помилок, читабельність коду та

Спочатку було проведено аналіз серверної частини програмного забезпечення (рис. 3.1).

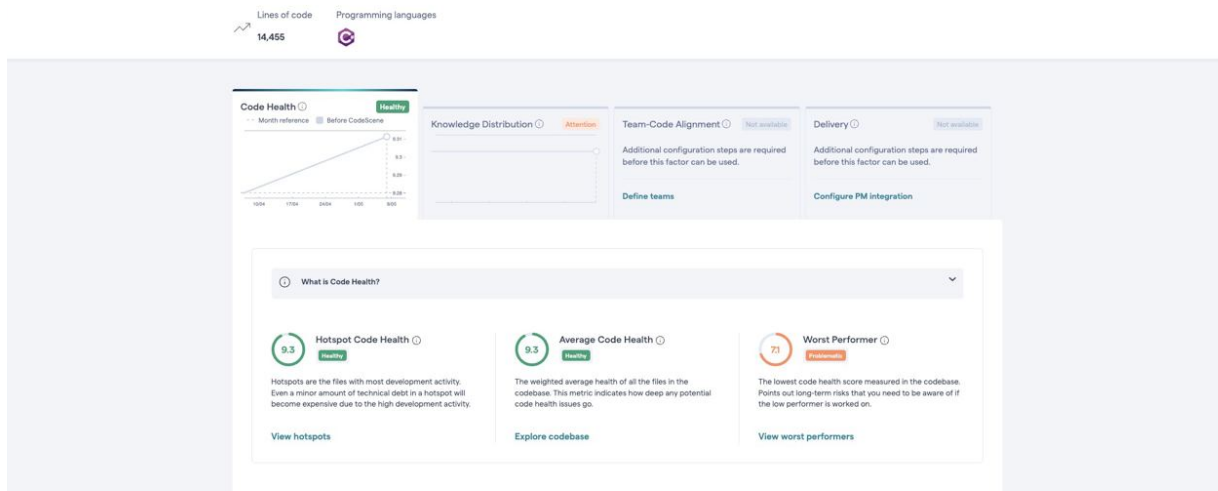


Рисунок 3.1 – Результат аналізу коду серверної частини

Після чого було проаналізовано візуальну частину застосунку (рис. 3.2).

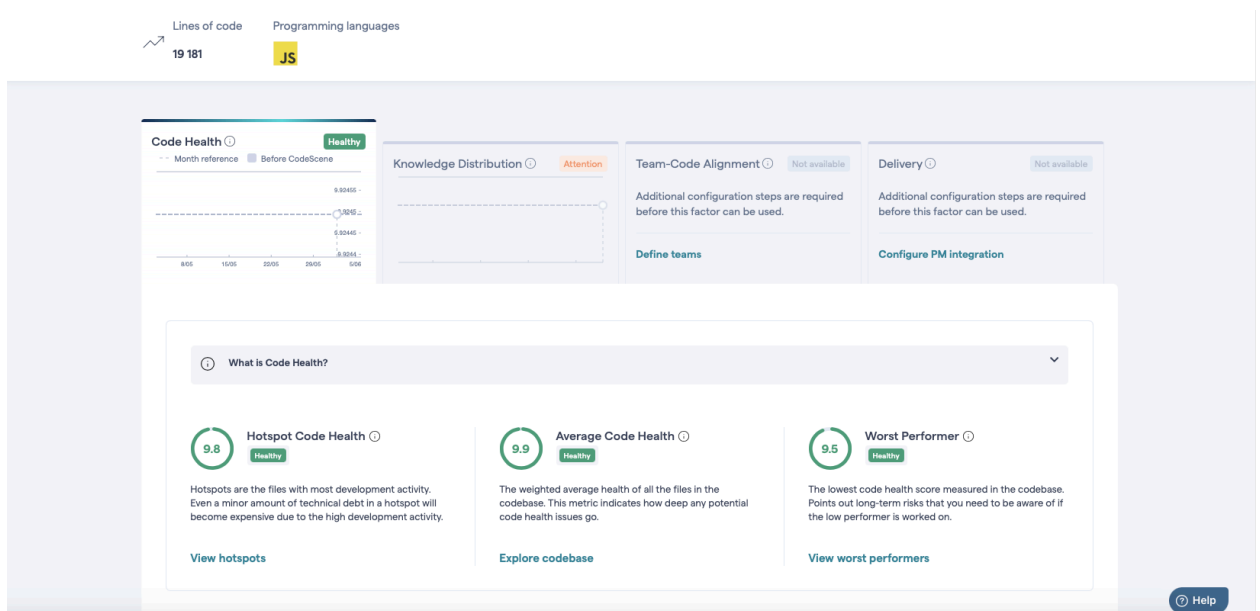


Рисунок 3.2 – Результат аналізу UI частини ПЗ

Виходячи з результатів аналізу кодової бази програмного продукту, можна сказати, що імплементація відповідає вимогам.

3.2 Опис процесів тестування

Тестування програмного забезпечення розкладу занять було проведено з боку користувача. Такий підхід передбачав послідовне введення вхідних даних, взаємодію з інтерфейсом та перевірку результатів, щоб переконатися в його відповідності вимогам. Мануальний вид тестування включає виконання

тестових сценаріїв та перевірку різних функціональних аспектів системи, щоб впевнитися в їх правильному функціонуванні та відповідності вимогам. Детальний опис відповідних тестів наведено нижче (табл. 3.1 – 3.8).

Таблиця 3.1 – Тест 1.1

Тест	Вибір групи з списку
Модуль	Інтерфейс користувача - Campus.Schedule.API
Номер тесту	1.1
Початковий стан системи	Користувач переходить до сторінки веб-застосунку.
Вхідні данні	Назва групи
Опис проведення тесту	У поле розташоване після «Розклад занять для» вводиться назва існуючої групи, розклад якої користувач бажає переглянути наступним кроком. Під час введення відображається список наявних груп, де в результаті відображається шукана група. Після цього натискається знайдена опція групи, яку шукали.
Очікуваний результат	Пошук групи відбувся успішно, групу знайдено у випадяючому списку.
Фактичний результат	Пошук групи відбувся успішно, групу знайдено у випадяючому списку.

Таблиця 3.2 – Тест 1.2

Тест	Перегляд розкладу занять для обраної групи
Модуль	Інтерфейс користувача - Campus.Schedule.API
Номер тесту	1.2
Початковий стан системи	Користувач переходить до сторінки веб-застосунку.
Вхідні данні	Назва групи

Продовження таблиці 3.2

Опис проведення тесту	У поле розташоване після «Розклад занять для» вводиться назва існуючої групи, розклад якої користувач бажає переглянути наступним кроком. Під час введення відображається список наявних груп, де в результаті відображається шукана група. Після цього натискається знайдена опція групи, яку шукали.
Очікуваний результат	Було знайдено групу, і її розклад занять відповідно до поточного тижня було успішно відображено.
Фактичний результат	Було знайдено групу, і її розклад занять відповідно до поточного тижня було успішно відображено.

Таблиця 3.3 – Тест 1.3

Тест	Перегляд розкладу сесії для обраної групи
Модуль	Інтерфейс користувача - Campus.Schedule.API
Номер тесту	1.3
Початковий стан системи	Користувач переходить до сторінки веб-застосунку.
Вхідні данні	Назва групи
Опис проведення тесту	На головній сторінці обирається опція «Розклад сесії». У поле розташоване після «Розклад занять для» вводиться назва існуючої групи, розклад сесії якої користувач бажає переглянути наступним кроком. Під час введення відображається список наявних груп, де в результаті відображається шукана група. Після цього натискається знайдена опція групи.
Очікуваний результат	Було знайдено групу, і її розклад сесії було успішно відображено.

Продовження таблиці 3.3

Фактичний результат	Було знайдено групу, і її розклад сесії було успішно відображено.
---------------------	---

Таблиця 3.4 – Тест 1.4

Тест	Вибір викладача зі списку
Модуль	Інтерфейс користувача - Campus.Schedule.API
Номер тесту	1.4
Початковий стан системи	Користувач переходить до сторінки веб-застосунку.
Вхідні данні	ПІБ викладача
Опис проведення тесту	На головній сторінці обирається опція «Розклад для викладачів». У поле розташоване після «Розклад занять для» вводиться ПІБ дійсного викладача, розклад занять якого користувач бажає переглянути наступним кроком. Під час введення відображається список дійсних вчителів, де в результаті відображається шукана особа. Після цього натискається знайдена опція викладача.
Очікуваний результат	Пошук викладача відбувся успішно, викладача знайдено у випадяючому списку.
Фактичний результат	Пошук викладача відбувся успішно, викладача знайдено у випадяючому списку.

Таблиця 3.5 – Тест 1.5

Тест	Перегляд розкладу занять для обраного викладача
Модуль	Інтерфейс користувача - Campus.Schedule.API
Номер тесту	1.5
Початковий стан системи	Користувач переходить до сторінки веб-застосунку.

Продовження таблиці 3.5

Вхідні данні	ПІБ викладача
Опис проведення тесту	На головній сторінці обирається опція «Розклад для викладачів». У поле розташоване після «Розклад занять для» вводиться ПІБ дійсного викладача, розклад занять якого користувач бажає переглянути наступним кроком. Під час введення відображається список дійсних вчителів, де в результаті відображається шукана особа. Після цього натискається знайдена опція викладача.
Очікуваний результат	Було знайдено викладача, і його розклад занять відповідно до поточного тижня було успішно відображено.
Фактичний результат	Було знайдено викладача, і його розклад занять відповідно до поточного тижня було успішно відображено.

Таблиця 3.6 – Тест 1.6

Тест	Перегляд розкладу занять групи за певний тиждень.
Модуль	Інтерфейс користувача - Campus.Schedule.API
Номер тесту	1.6
Початковий стан системи	Користувач переходить до сторінки веб-застосунку.
Вхідні данні	Назва групи
Опис проведення тесту	У поле розташоване після «Розклад занять для» вводиться назва існуючої групи, розклад якої користувач бажає переглянути наступним кроком. Під час введення відображається список наявних груп, де в результаті відображається шукана група. Після цього натискається знайдена опція групи, яку шукали. Далі, виконується вибір тижня для відображення розкладу занять, натискаючи на опції «Перший тиждень»/«Другий тиждень».

Продовження таблиці 3.6

Очікуваний результат	Було знайдено групу, і її розклад занять відповідно до натиснутої опції тижня було успішно відображено.
Фактичний результат	Було знайдено групу, і її розклад занять відповідно до натиснутої опції тижня було успішно відображено.

Таблиця 3.7 – Тест 1.7

Тест	Перегляд розкладу занять викладачів за певний тиждень.
Модуль	Інтерфейс користувача - Campus.Schedule.API
Номер тесту	1.7
Початковий стан системи	Користувач переходить до сторінки веб-застосунку.
Вхідні данні	ПІБ викладача
Опис проведення тесту	На головній сторінці обирається опція «Розклад для викладачів». У поле розташоване після «Розклад занять для» вводиться ПІБ дійсного викладача, розклад занять якого користувач бажає переглянути наступним кроком. Під час введення відображається список дійсних викладачів, де в результаті відображається шукана особа. Після цього натискається знайдена опція викладача. Наступним кроком, виконується вибір тижня для відображення розкладу занять, натискаючи на опції «Перший тиждень» та «Другий тиждень».
Очікуваний результат	Було знайдено викладача, і його розклад занять відповідно до натиснутої опції тижня було успішно відображено.
Фактичний результат	Було знайдено викладача, і його розклад занять відповідно до натиснутої опції тижня було успішно відображено.

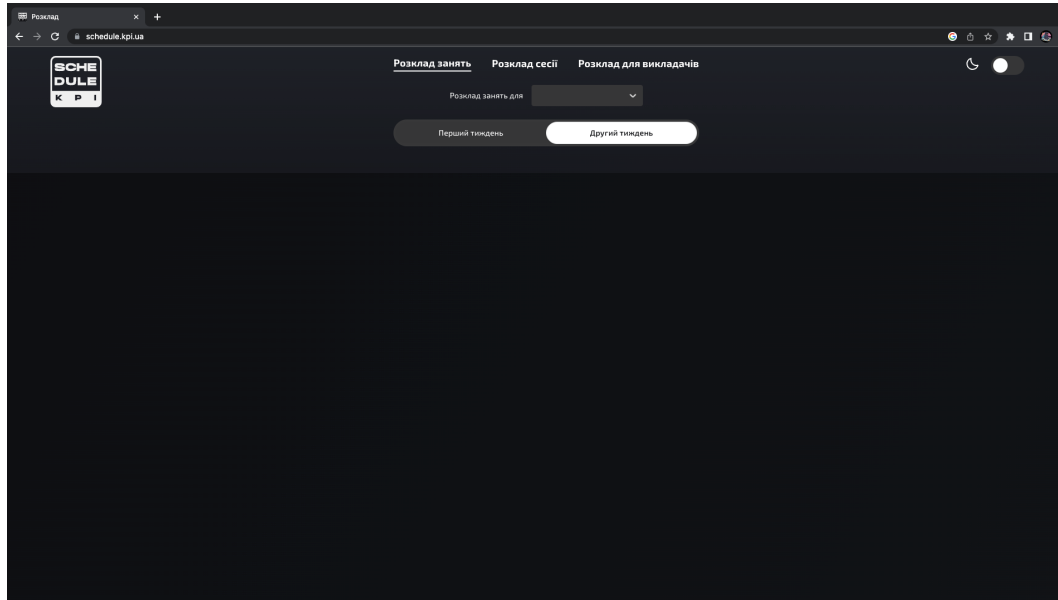


Рисунок 3.1 – Приклад головної сторінки веб-додатку розкладу занять

Після натискання на поле поряд з «Розклад занять для» буде відображено список актуальних груп університету, з якого можна вибрати бажану групу та в подальшому переглянути її розклад (рис. 3.2).

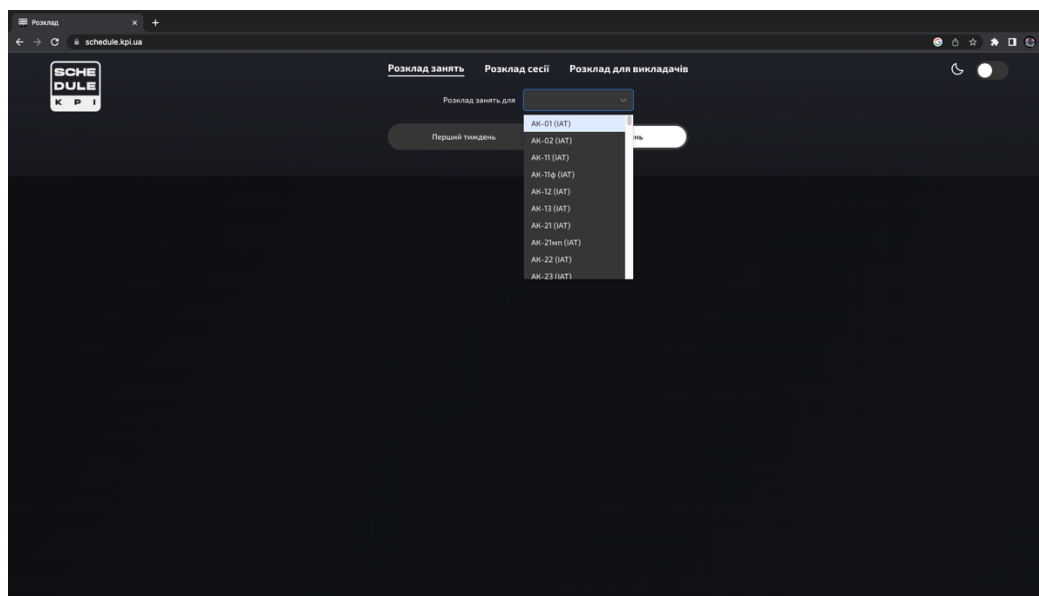


Рисунок 3.2 – Приклад списку доступних груп

Після натискання на одну з запропонованих груп, буде надіслано запит та відображено розклад занять обраної групи. Для демонстрації оберемо групу АК-01 (рис. 3.3).

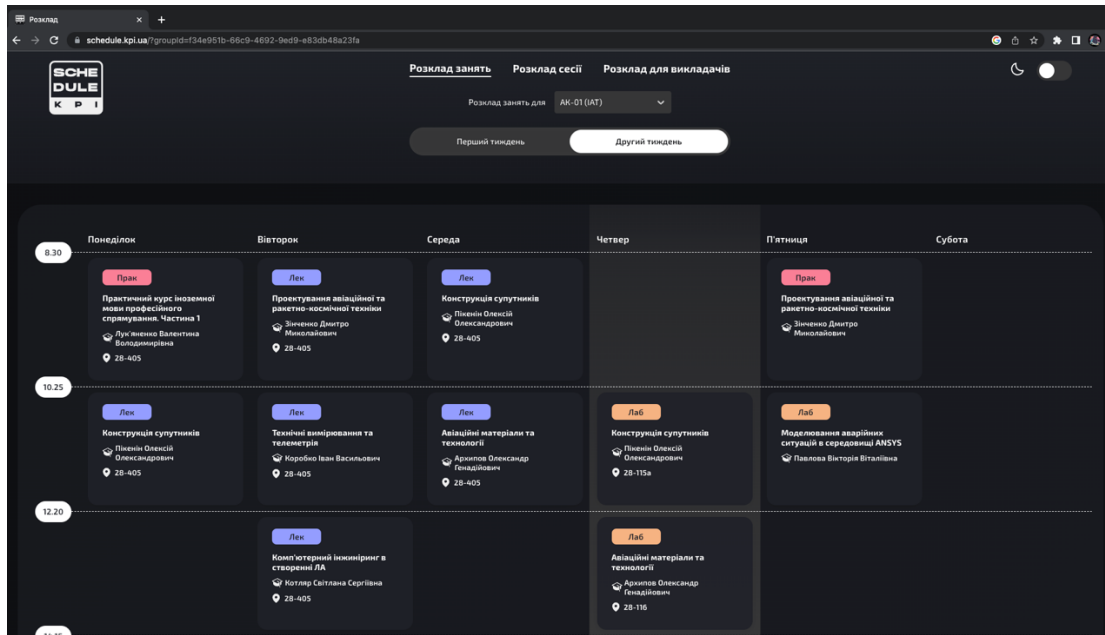


Рисунок 3.3 – Приклад розкладу занять групи АК-01

Також, для спрощення пошуку групи можна просто почати вводити її назву в поле з випадаючим списком та натиснути на її назву, як тільки вона відобразиться у списку. Для демонстрації спробуємо знайти розклад групи ІТ-92 (рис. 3.4 - 3.5).

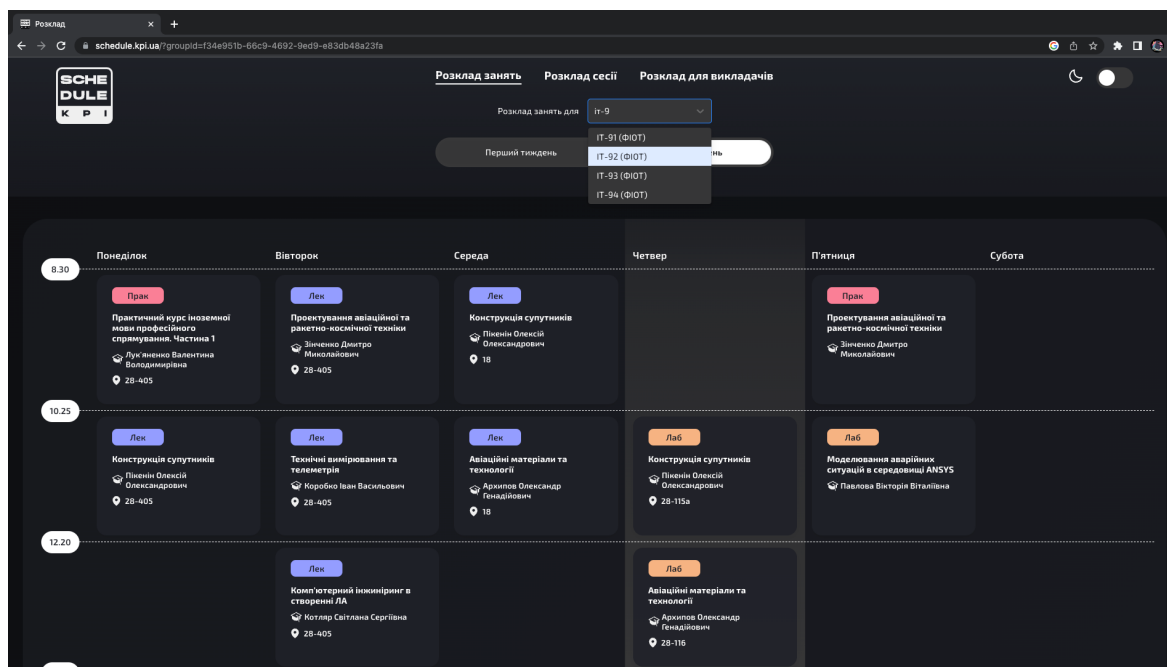


Рисунок 3.4 – Приклад пошуку групи ІТ-92

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

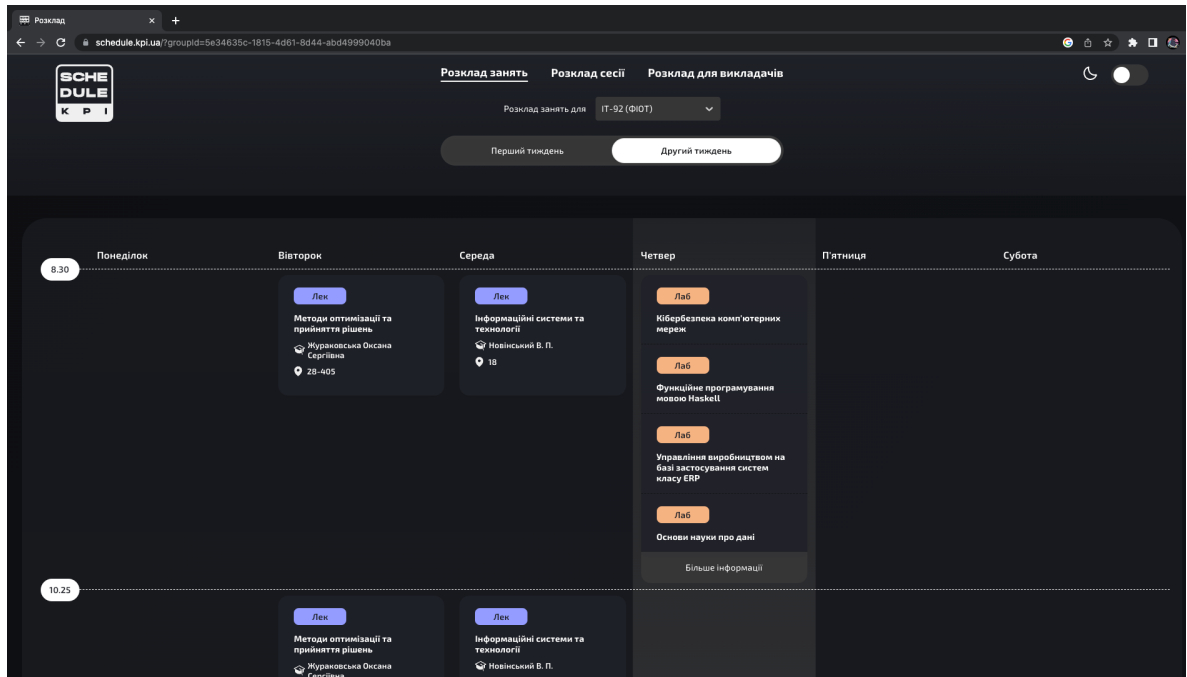


Рисунок 3.5 – Приклад відображення розкладу групи ІТ-92 після натискання на її назву у списку

Для перегляду розкладу занять за інший тиждень, можна скористатися перемикачем. Та натиснути на «Перший тиждень» для відображення розкладу за перший тиждень або на «Другий тиждень» для відображення розкладу за другий тиждень, відповідно (рис. 3.6).

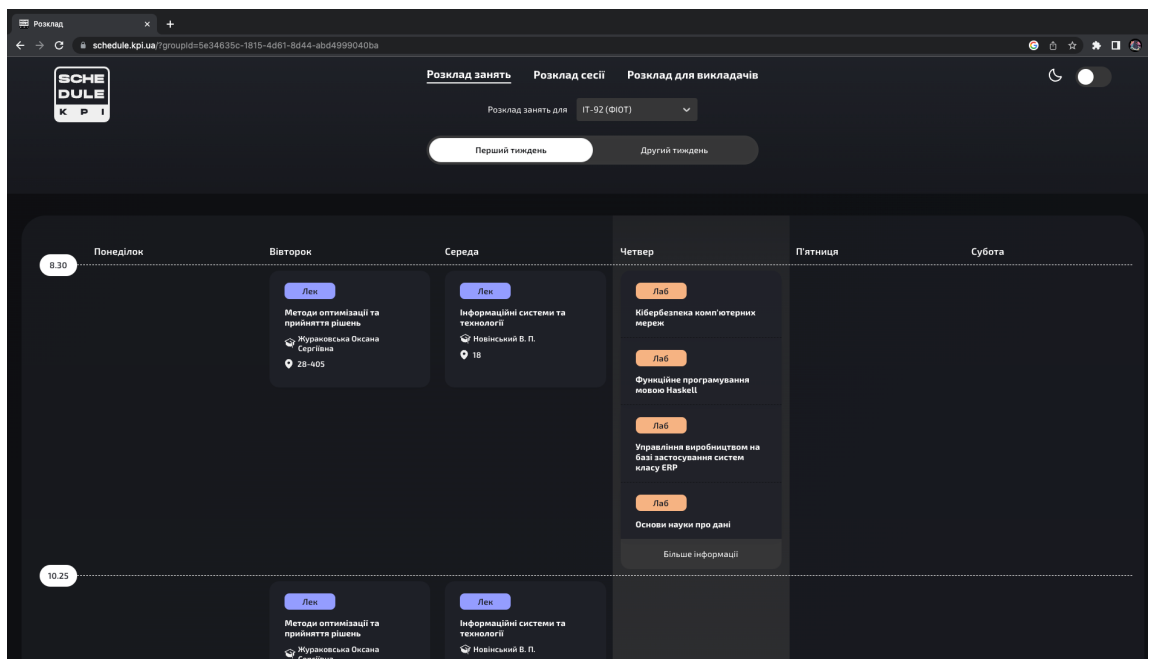


Рисунок 3.6 – Приклад розкладу занять групи за інший тиждень

Тепер перейдемо до опції «Розклад сесії». Якщо попередньо група була обрана для перегляду її розкладу занять, при переході до розкладу сесії, автоматично відобразиться розклад екзаменів попередньо обраної групи (рис. 3.7).

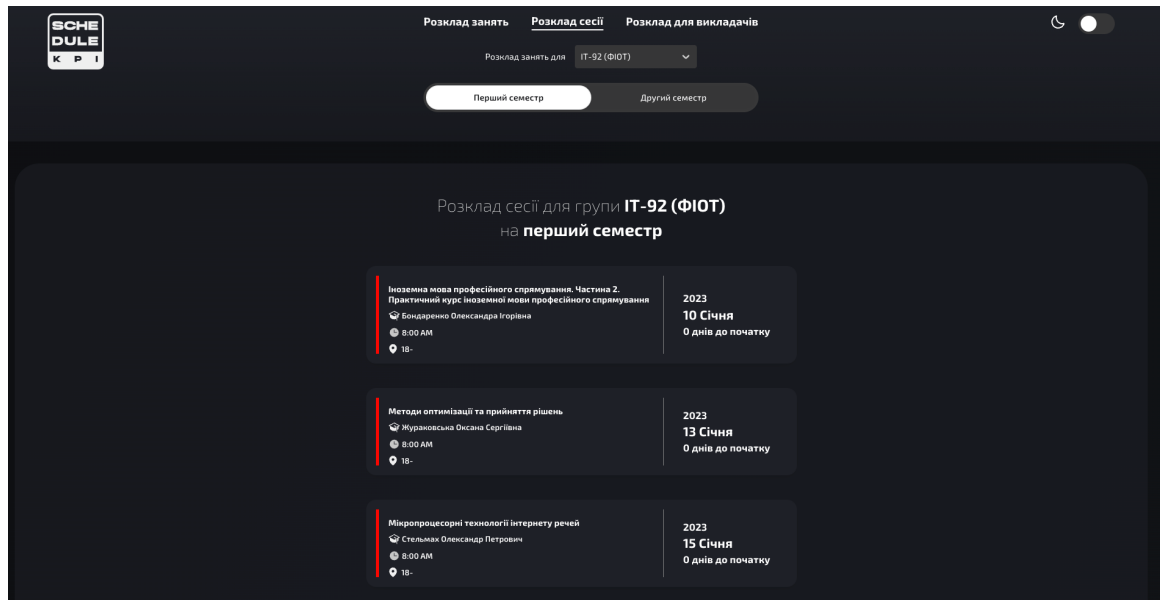


Рисунок 3.7 – Приклад розкладу занять сесії для попередньо обраної групи

Додатковою функцією застосунку є використання пошуку біля поля «Розклад занять для» для перегляду розкладу сесії будь-якої іншої існуючої групи. Переглянемо розклад сесії для групи ІС-91 (рис. 3.8).

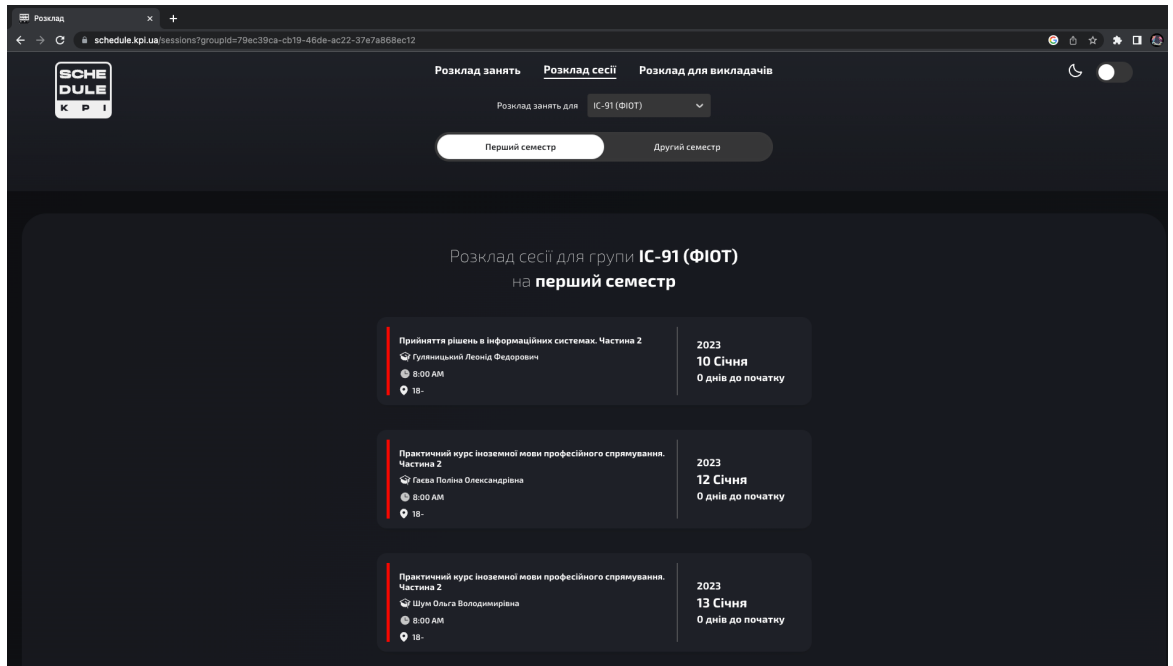


Рисунок 3.8 – Приклад розкладу сесії для групи IC-91

Система також надає можливість перегляду розкладу викладача, для цього треба натиснути на опцію «Розклад для викладачів». Після чого користувача перенесе на сторінку, де можна буде переглядати план занять вчителів (рис. 3.9).

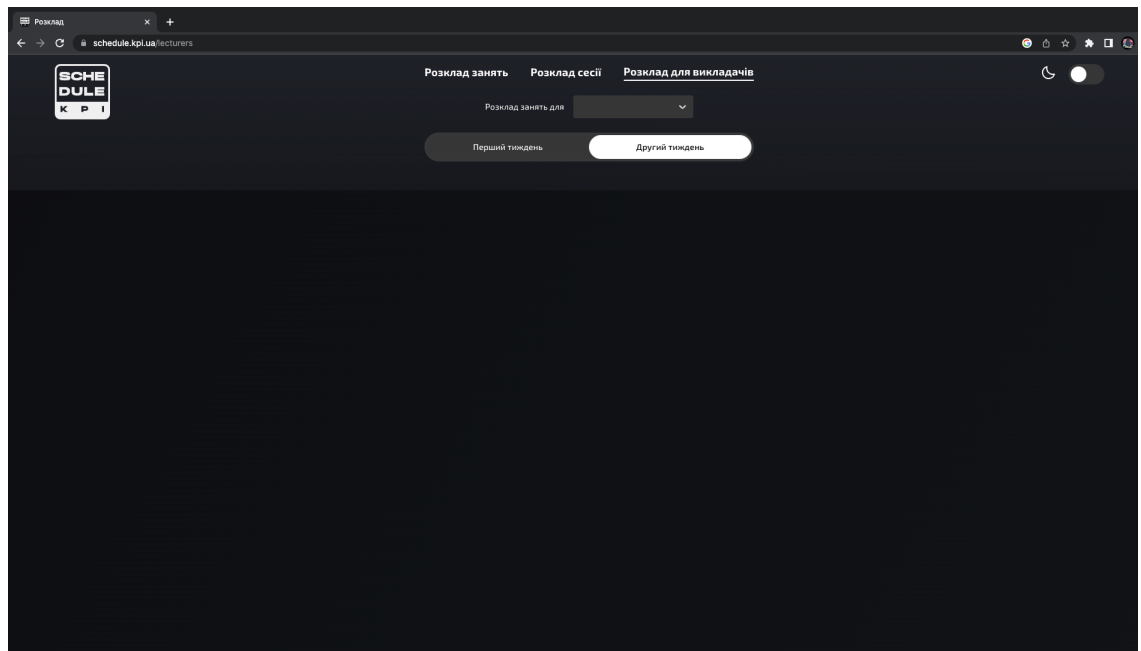


Рисунок 3.9 – Приклад первинної сторінки розкладу занять викладачів

Після натискання на поле поряд з «Розклад занять для» буде відображено список викладачів університету, з якого можна вибрати бажаного вчителя по його ПІБ та в подальшому переглянути його розклад (рис. 3.10).

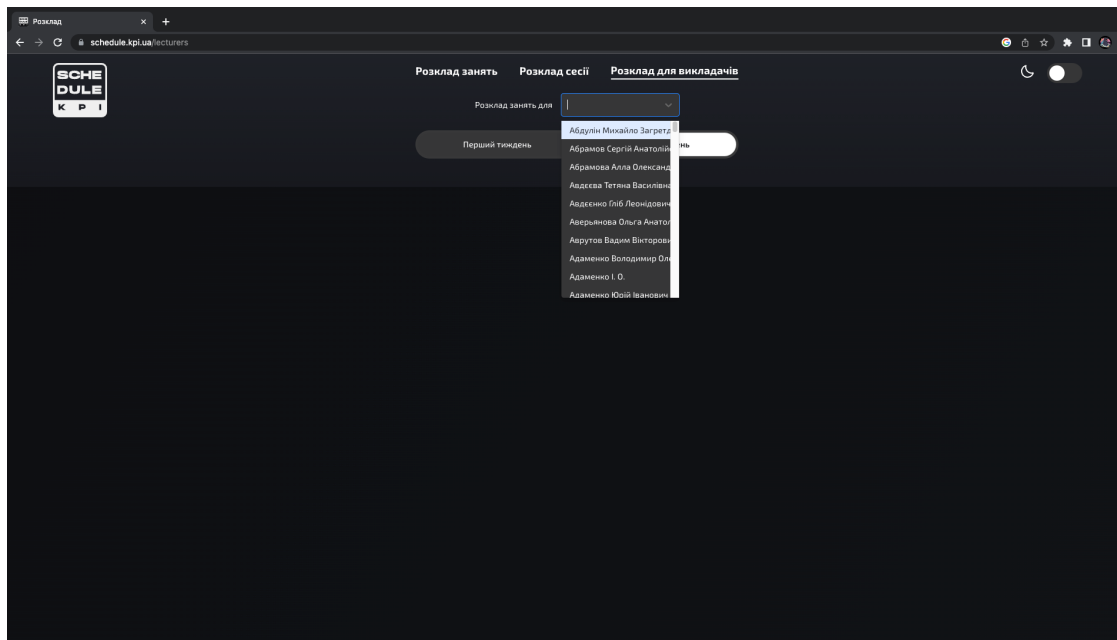


Рисунок 3.10 – Приклад актуального списку викладачів університету

Після натискання на одного з запропонованих викладачів, буде надіслано запит та відображено розклад його занять групи. Для демонстрації оберемо викладача «Корнага Ярослав Ігоревич» (рис. 3.11).

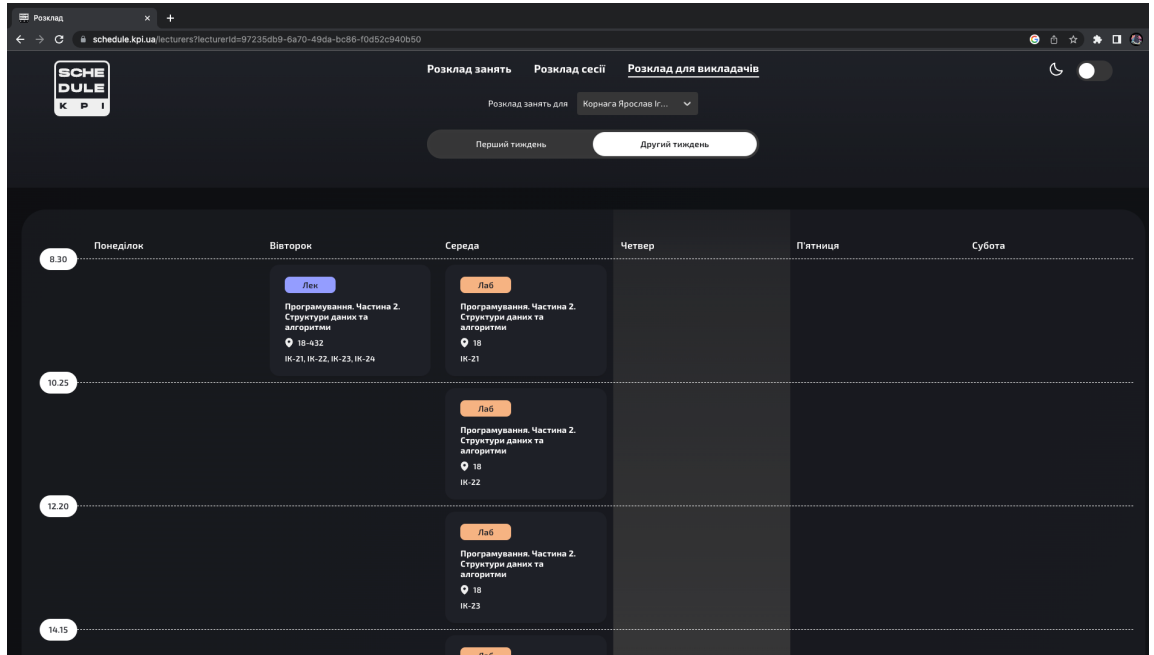


Рисунок 3.11 – Приклад розкладу занять викладача Корнаги Я.І.

Також, для спрощення пошуку викладача можна просто почати вводити його прізвище в поле з випаданим списком та натиснути на опцію, як тільки потрібний викладач відобразиться у списку. Для демонстрації спробуємо знайти розклад викладача Лісовиченка Олега Івановича (рис. 3.12 - 3.13).

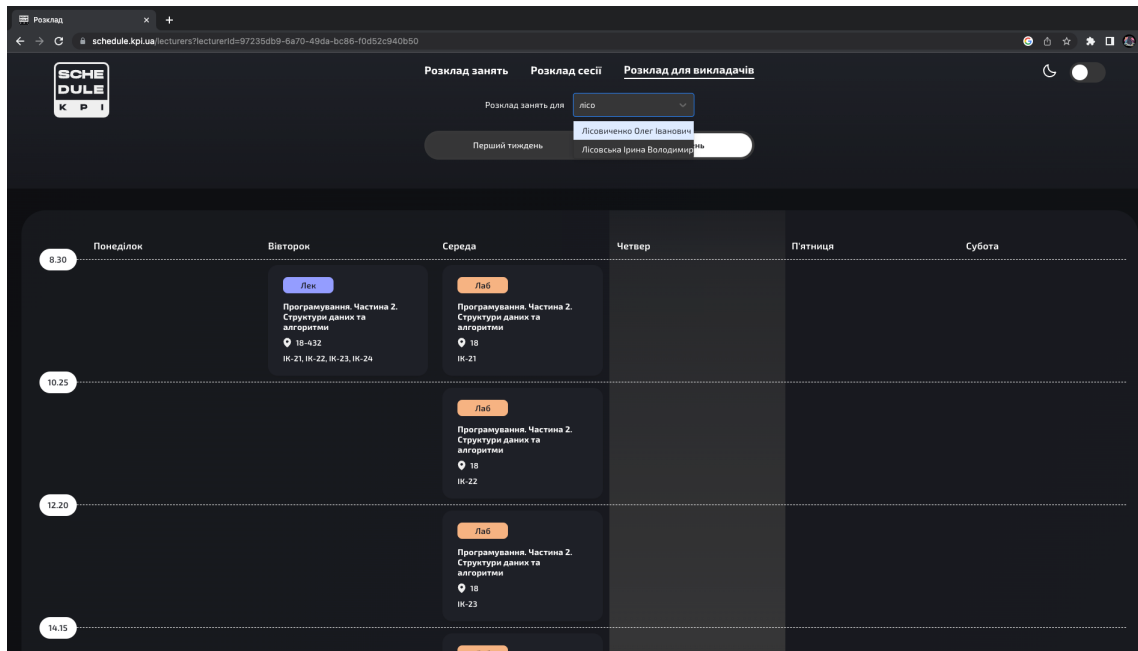


Рисунок 3.12 – Приклад пошуку викладача Лісовиченка О.І.

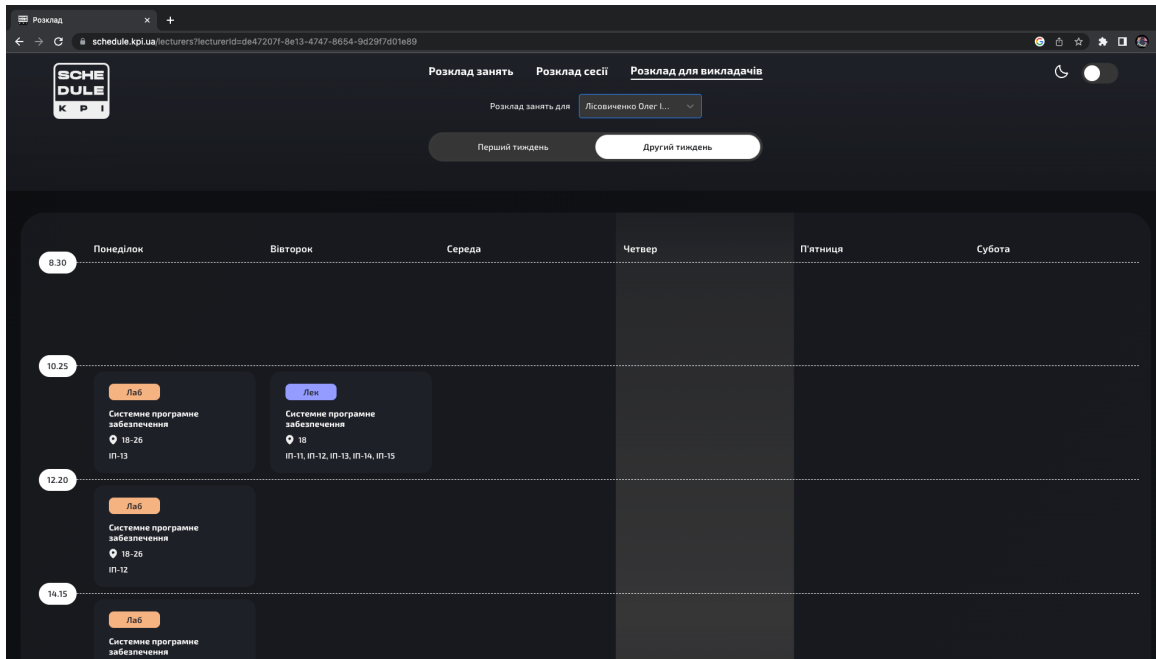


Рисунок 3.13 – Приклад відображення розкладу викладача після натискання на його ПІБ у списку

На сторінці для відображення розкладу вчителів також можна переглянути розклад занять за поточний та наступний тижні. Для перегляду розкладу занять за інший тиждень, можна скористатися перемикачем. Та натиснути на «Перший тиждень» для відображення розкладу за перший тиждень або на «Другий тиждень» для відображення розкладу за другий тиждень, відповідно (рис. 3.14).

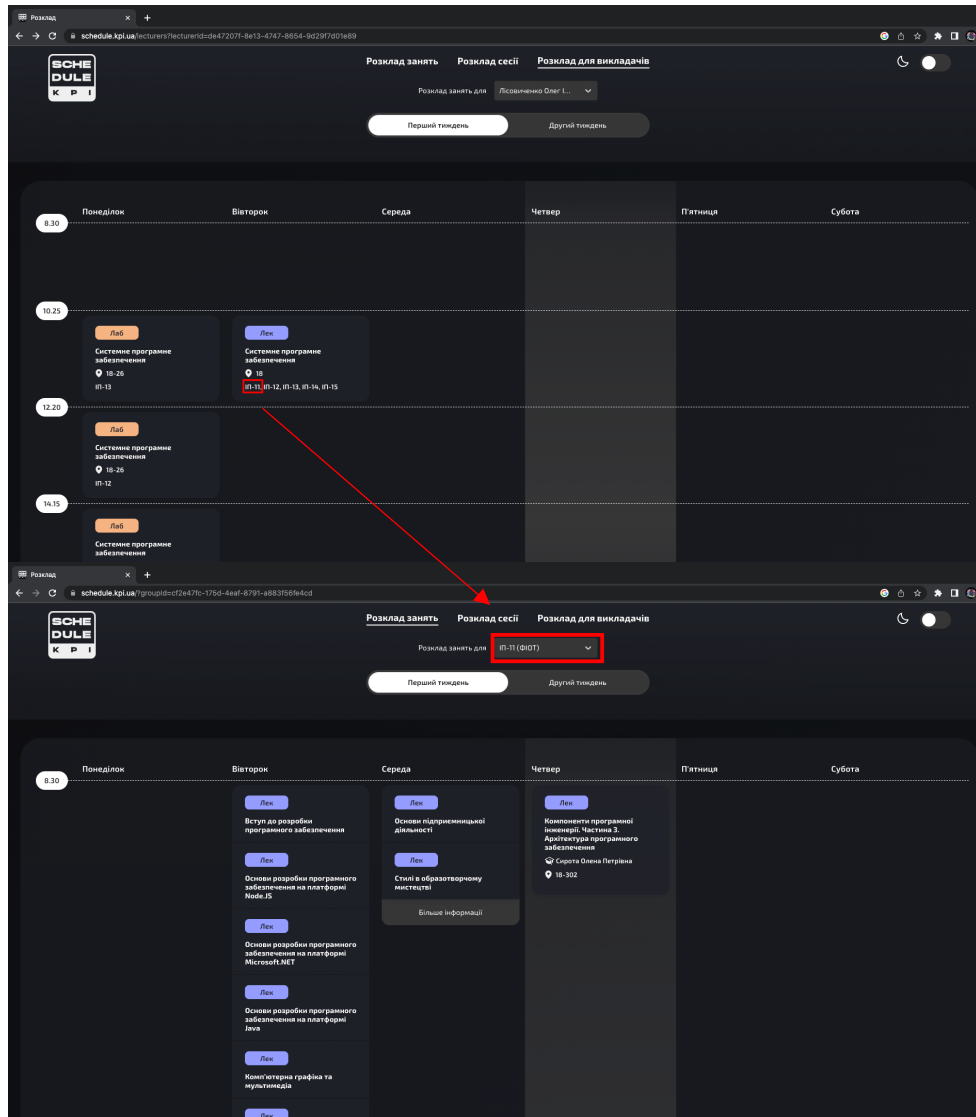


Рисунок 3.15 – Приклад переходу до розкладу групи з розкладу викладача

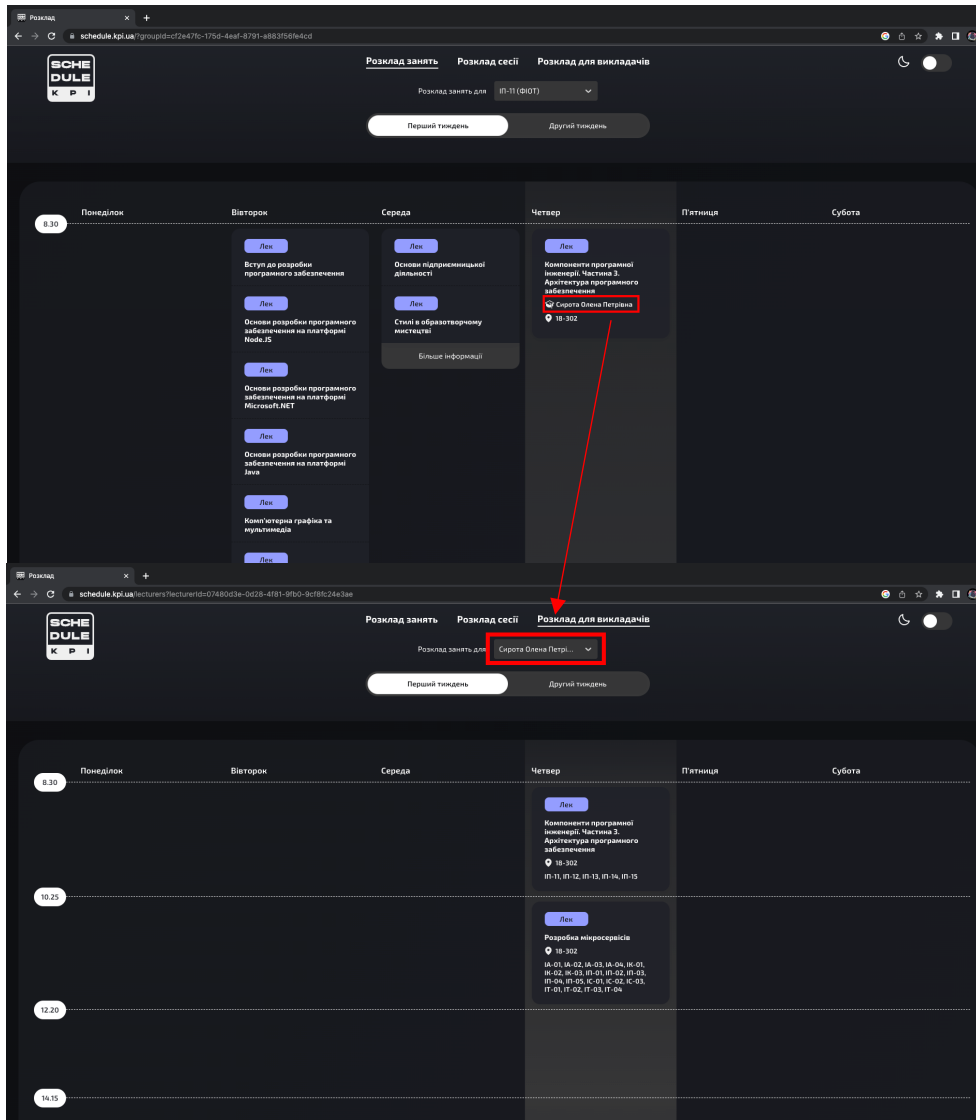


Рисунок 3.16 – Приклад переходу до розкладу викладача з розкладу групи

Веб-застосунок також надає можливість користувачу змінити тему застосунку з темного в світлий і з світлого в темний відповідно. Для цього треба скористатись перемикачем в правому верхньому куту додатку (рис. 3.17).

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

Після чого було проведено мануальне тестування системи з боку користувача, що дозволило перевірити функціональність та коректність роботи програмного забезпечення. Застосовуючи різні сценарії тестування доведено, що робота системи відповідає функціональним вимогам.

Також було детально описано контрольний приклад, який демонструє основні можливості системи. Він включав в себе вибір групи або викладача, перегляд розкладу занять та інші функції, яким користувач може скористатися під час роботи з системою.

Отже, результати проведеного аналізу якості та мануального тестування демонструють відповідність програмного забезпечення розкладу занять встановленим вимогам та підтверджують його функціональність та надійність. Подальші кроки роботи над проектом передбачають автоматизацію тестування та вдосконалення системи для забезпечення ще більшої ефективності та задоволення потреб користувачів.

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		58

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Для розгортання було використано популярне програмне забезпечення Docker, яке дозволяє створювати нові контейнери та маніпулювати ними. Він забезпечує ізольоване середовище для застосунків, що спрощує їх розгортання та забезпечує цілісність даних між різними середовищами.

Для автоматизації самого розгортання та забезпечення постійної інтеграції було використано сервіс GitHub Actions. Він дозволяє створювати та налаштовувати автоматизовані робочі процеси. Такі як: збирання, тестування та розгортання застосунків. Використання GitHub Actions спрощує сам процес розгортання та забезпечує високу якість програмного забезпечення завдяки автоматичному запуску тестів та перевірці якості коду.

Розгортання починається одразу після додавання нового коду до репозиторію застосунку. В цей момент, за допомогою середовища GitHub Actions та відповідного файлу конфігурації Dockerfile, створюється новий образ Docker. Такий образ є збіркою необхідних компонентів, середовища та залежностей, необхідних для виконання додатку.

Створення нового Docker-образу здійснюється на основі оновленого коду з репозиторію. Dockerfile містить інструкції, які визначають, які компоненти і налаштування повинні бути включені до образу, що містить в собі встановлення залежностей, налаштування середовища та копіювання відповідних файлів до контейнера.

Отриманий Docker-образ може бути далі розгорнутий на сервері або контейнерному середовищі, де додаток буде запущено. Після чого додаток стане доступним для використання. Таке розгортання образу дозволяє забезпечити послідовне та надійне розгортання додатку на потрібному сервері без необхідності його мануального конфігурування.

4.2 Підтримка програмного забезпечення

Завдяки використанню сервісу GitHub Actions, підтримка та оновлення програмного забезпечення виконується автоматизовано. Тобто, після додавання нового коду буде створено та розгорнуто новий образ Docker. Таким чином, користувачі зможуть якнайшвидше отримати нову версію додатку.

Висновки до розділу

Для розгортання системи було використано програмне забезпечення Docker, яке забезпечує ізольоване середовище для застосунків. Такий підхід спрощує процес розгортання та забезпечує цілісність даних між різними середовищами.

З метою автоматизації розгортання та забезпечення постійної інтеграції було використано сервіс GitHub Actions. Його можливості дозволили створити та налаштувати автоматизоване збирання та розгортання застосунку.

Використання Docker та GitHub Actions забезпечило швидке впровадження оновлень.

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		60

Після чого було розглянуто схему бізнес-процесу та визначено архітектуру програмного забезпечення, завдяки побудові ER діаграми сутностей. Було виділено три основні сутності та проаналізовано відношення між ними. Виходячи з цього було визначено структуру та поля таблиць бази даних.

Проаналізувавши найпоширеніші технічні рішення, у результаті було обрано C# ASP.NET для написання API та React JS для написання UI частини. ASP.NET забезпечує швидку обробку запитів на серверному рівні та підтримує оптимізацію роботи з базою даних. В свою чергу, React JS дозволяє ефективно оновлювати лише необхідні компоненти на клієнтській стороні, що знижує навантаження на мережу та забезпечує швидке відображення змін.

Системою управління базами даних було обрано використовувати PostgreSQL. Ця база даних призначена для зберігання і управління даними про розклад занять викладачів та здобувачів освітньої послуги, а також даними про їх сесії. Вона забезпечує надійне та ефективне зберігання і доступ до інформації, необхідної для функціонування системи.

Було обрано Rider та IntelliJ IDEA в якості середовищ розробки, адже вони виявились найбільш зручними для маніпулювання кодом користувацького інтерфейсу та серверної частини додатку. Також, Rider надав зручний інструмент Database Window для маніпулювання базою даних PostgreSQL.

Наступним кроком постало тестування якості коду ПЗ за допомогою онлайн сервісу та мануального тестування. А також, продемонстровано контрольний приклад, який перевірів працездатність системи перед впровадженням її в реальні умови. І останнім етапом роботи над дипломним проєктом постав аналіз процесу розгортання програмного забезпечення.

Переваги програмного забезпечення:

- впроваджено оновлений та зрозумілий користувацький інтерфейс;
- застосовано сучасну технологічну базу для імплементації;
- впроваджено інтеграцію з єдиним середовищем університету;

– налаштовано автоматичне оновлення системи та полегшено процес її підтримки.

Вперше:

– адаптація програмного забезпечення для коректної роботи в умовах великих навантажень, помилок оновлення даних системи або перебоїв електропостачання;

– автоматичне розгортання системи в хмарі при внесенні змін в репозиторій програмного забезпечення.

					КПІ.ІТ-9203.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		63

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) A tour of C# language Reference [Електронний ресурс] / Microsoft – Режим доступу до ресурсу <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- 2) JavaScript Reference [Електронний ресурс] / Wikipedia – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/JavaScript>
- 3) Overview of ASP.NET Core Reference [Електронний ресурс] / Microsoft – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core>
- 4) React_(software) Reference [Електронний ресурс] / Wikipedia – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/React_\(software\)](https://en.wikipedia.org/wiki/React_(software))
- 5) CodeScene Reference [Електронний ресурс] / Codescene – Режим доступу до ресурсу: <https://codescene.io>

ДОДАТКИ

ДОДАТОК А ЗВІТ ПОДІБНОСТІ



Ім'я користувача:
Лісовиченко Олег Іванович

Дата перевірки:
03.06.2023 13:31:34 EEST

Дата звіту:
03.06.2023 13:39:51 EEST

ID перевірки:
1015405710

Тип перевірки:
Doc vs Internet + Library

ID користувача:
76913

Назва документа: ІТ-92_Бойко_ПЗ (1)

Кількість сторінок: 60 Кількість слів: 9100 Кількість символів: 72400 Розмір файлу: 6.69 MB ID файлу: 1015069356

11.6% Схожість

Найбільша схожість: 4.98% з джерелом з Бібліотеки (ID файлу: 1015064292)

3.36% Джерела з Інтернету	265	Сторінка 62
11.5% Джерела з Бібліотеки	330	Сторінка 64

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

КПІ.ІТ-9203.045440.02.81

Арк.

65

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

Імплементація API для платформи Schedule KPI

Текст програми

КПІ.ІТ-9203.045440.03.12

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

Нормоконтроль:

_____ Ірина ВІТКОВСЬКА

Виконавець:

_____ Єлизавета БОЙКО

Київ – 2023

Файл ExamsControlles.cs

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Campus.Core.Repositories;
using Campus.Schedule.API.Models;
using Campus.Web;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;

namespace Campus.Schedule.API.Controllers;

[ApiController]
public class ExamsController : BaseApiController
{
    private readonly IScheduleRepository _repository;
    public ExamsController(ILogger<ExamsController> logger, IScheduleRepository repository) : base(logger)
    {
        _repository = repository;
    }

    [HttpGet]
    [Route("exams/group")]
    public async Task<IActionResult> GetExamsById([FromQuery] string groupId)
    {
        var exams = _repository.GetExamsByGenericGroupId(groupId);
        var examsPopulated = new List<ExamWithDeadline>();

        foreach (var exam in exams)
        {
            var daysLeft = (exam.Date - DateTime.Now).Days;

            if (daysLeft < 0)
            {
                daysLeft = 0;
            }

            examsPopulated.Add(new ExamWithDeadline(exam, daysLeft));
        }

        return await Result(examsPopulated);
    }
}
```

Файл GroupController.cs

```
using System.Threading.Tasks;
using Campus.Web;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;

namespace Campus.Schedule.API.Controllers;

[ApiController]
public class GroupController : BaseApiController
{
    private readonly Core.Repositories.IScheduleRepository _repository;

    public GroupController(Core.Repositories.IScheduleRepository repository, ILogger<GroupController> logger)
        : base(logger)
    {
        _repository = repository;
    }

    [HttpGet]
    [Route("schedule/groups")]
    public async Task<IActionResult> GetGroupsList([FromQuery] string filter)
```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

{
    var groups = filter != null ? _repository.GetGroupsGenericFiltered(filter) : _repository.GetGroupsGeneric();

    return await Result(groups);
}
}

```

Файл LecturersController.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Campus.Models.Schedule;
using Campus.Schedule.API.Models;
using Campus.Web;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;

namespace Campus.Schedule.API.Controllers;

[ApiController]
public class LecturersController : BaseApiController
{
    private readonly Core.Repositories.IScheduleRepository _repository;

    public LecturersController(Core.Repositories.IScheduleRepository repository, ILogger<GroupController> logger)
        : base(logger)
    {
        _repository = repository;
    }

    [HttpGet]
    [Route("schedule/lecturer/list")]
    public async Task<IActionResult> GetLecturersList([FromQuery] string filter)
    {
        var lecturers = filter != null
            ? await _repository.GetLecturersListFiltered(filter)
            : await _repository.GetLecturersList();

        return await Result(lecturers);
    }

    [HttpGet]
    [Route("schedule/lecturer")]
    public async Task<IActionResult> GetLessonsList([FromQuery] Guid lecturerId)
    {
        var lessons = _repository.GetLessonsByLecturerId(lecturerId);
        var days = Days.GetAll();
        var res = new LecturerScheduleResponse(lecturerId.ToString());

        for (var i = 1; i < 3; i++)
        {

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

foreach (var day in days)
{
    var lessonByWeekAndDay = lessons
        .Where(l => l.Week == i)
        .Where(l => l.Day == day)
        .Select(l => new LecturerLesson
            {
                Type = l.LessonType,
                Time = LessonTime.GetTimeByLessonNumber(l.Number),
                Name = l.Subject,
                Group = l.GroupName,
                Place = l.Room,
                Tag = GetTag(l.LessonType)
            })
        .ToList();

    var lessonsGroupedGroups = new List<LecturerLesson>();

    foreach (var lesson in lessonByWeekAndDay)
    {
        var lessonFromColl = lessonsGroupedGroups
            .FirstOrDefault(l => l.Name == lesson.Name
                && l.Type == lesson.Type
                && l.Time == lesson.Time);

        if (lessonFromColl != null)
        {
            continue;
        }

        var lessonByWeekDayTime = lessonByWeekAndDay.Where(l => l.Time == lesson.Time);
        var allGroups = string.Join(", ", lessonByWeekDayTime.Select(l => l.Group));

        lesson.Group = allGroups;
        lessonsGroupedGroups.Add(lesson);
    }

    var scheduleItem = new LecturerScheduleDayItem
    {
        Day = day,
        Pairs = lessonsGroupedGroups
    };

    if (i == 1)
    {
        res.ScheduleFirstWeek.Add(scheduleItem);
    }
    else
    {
        res.ScheduleSecondWeek.Add(scheduleItem);
    }
}
}

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

        return await Result(res);
    }

    private static string GetTag(string lessonType)
    {
        var lectureTypes = new[] { "1,5 Лек" , "Лек-в", "Лек", "0,5 Лек", "Лек on-line" };
        var labTypes = new[] { "0,5 Лаб" , "Лаб", "Лаб on-line","Лаб-в" , "Конс on-line", "1,5 Лаб"};
        var practiceTypes = new[] { "Прак-в", "Прак", "0,5 Прак", "0,4 Прак" , "1,5 Прак", "Прак on-line"};

        if (lectureTypes.Contains(lessonType))
        {
            return "lec";
        }

        if (labTypes.Contains(lessonType))
        {
            return "lab";
        }

        if (practiceTypes.Contains(lessonType))
        {
            return "prac";
        }

        return "";
    }
}

```

Файл ScheduleController.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Campus.Core.Repositories;
using Campus.Models.Schedule;
using Campus.Schedule.API.Models;
using Campus.Web;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;

namespace Campus.Schedule.API.Controllers;

[ApiController]
public class ScheduleController : BaseApiController
{
    private readonly IScheduleRepository _repository;

    public ScheduleController(IScheduleRepository repository, ILogger<ScheduleController> logger)
        : base(logger)
    {

```

					КПІ.ІТ-9203.045440.03.12	Арк. 5
Змін.	Арк.	№ докум.	Підп.	Дата.		

```

    _repository = repository;
}

[HttpGet]
[Route("schedule/lessons")]
public async Task<IActionResult> GetLessonsById([FromQuery] string groupId, [FromQuery] string groupName)
{
    if (string.IsNullOrEmpty(groupId) && string.IsNullOrEmpty(groupName))
    {
        return BadRequest($"One if required parameters `{nameof(groupId)}` or `{nameof(groupName)}` missing");
    }

    var lessons = await GetLessons(groupId, groupName);

    var days = Days.GetAll();

    var res = new ScheduleResponse(groupId);

    for (var i = 1; i < 3; i++)
    {
        foreach (var day in days)
        {
            var lessonByWeekAndDay = lessons
                .Where(l => l.Week == i)
                .Where(l => l.Day == day)
                .ToList();

            var scheduleItem = new GroupScheduleDayItem
            {
                Day = day,
                Pairs = lessonByWeekAndDay.Select(l => new GroupLesson
                {
                    Type = l.LessonType,
                    Time = LessonTime.GetTimeByLessonNumber(l.Number),
                    Name = l.Subject,
                    TeacherName = l.LecturerName,
                    Place = l.Room,
                    LecturerId = l.LecturerId,
                    Tag = GetTag(l.LessonType)
                }).ToList()
            };

            if (i == 1)
            {
                res.ScheduleFirstWeek.Add(scheduleItem);
            }
            else
            {
                res.ScheduleSecondWeek.Add(scheduleItem);
            }
        }
    }
}

```

Вмін.	Арк.	№ докум.	Підп.	Дата.

```

return await Result(res);
}

private async Task<ICollection<ScheduledLesson>> GetLessons(string groupId, string groupName)
{
    if (!string.IsNullOrEmpty(groupId))
    {
        return await _repository.GetLessonsByGroupId(groupId);
    }

    if (!string.IsNullOrEmpty(groupName))
    {
        return await _repository.GetLessonsByGroupName(groupName);
    }

    throw new ArgumentException($"One if required parameters `{nameof(groupId)}` or `{nameof(groupName)}` missing");
}

private static string GetTag(string lessonType)
{
    var lectureTypes = new[] { "1,5 Лек", "Лек-в", "Лек", "0,5 Лек", "Лек on-line" };
    var labTypes = new[] { "0,5 Лаб", "Лаб", "Лаб on-line", "Лаб-в", "Конс on-line", "1,5 Лаб" };
    var practiceTypes = new[] { "Прак-в", "Прак", "0,5 Прак", "0,4 Прак", "1,5 Прак", "Прак on-line" };

    if (lectureTypes.Contains(lessonType))
    {
        return "lec";
    }

    if (labTypes.Contains(lessonType))
    {
        return "lab";
    }

    if (practiceTypes.Contains(lessonType))
    {
        return "prac";
    }

    return "";
}
}

```

Файл TimeController.cs

```

using System;
using System.Threading.Tasks;
using Campus.Core.Repositories;
using Campus.Schedule.API.Models;
using Campus.Schedule.API.Services;
using Campus.Web;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;

```

					КПІ.ІТ-9203.045440.03.12	Арк.
Вмін.	Арк.	№ докум.	Підп.	Дата.		7

```

namespace Campus.Schedule.API.Controllers;

[ApiController]
public class TimeController : BaseApiController
{
    private readonly IScheduleRepository _repository;
    private readonly ScheduleWeekService _weekService;

    public TimeController(ILogger<TimeController> logger, IScheduleRepository repository,
        ScheduleWeekService weekService) : base(logger)
    {
        _repository = repository;
        _weekService = weekService;
    }

    [HttpGet]
    [Route("time/current")]
    public async Task<IActionResult> GetCurrentData()
    {
        return await Result(new CurrentDayResponse
        {
            CurrentWeek = _weekService.GetCurrentWeek(),
            CurrentDay = GetCurrentDay(),
            CurrentLesson = GetCurrentLesson()
        });
    }

    private static int GetCurrentDay() => (int)DateTime.Now.DayOfWeek;

    /// <summary>
    /// Returns 0 if no lesson was found
    /// </summary>
    /// <returns></returns>
    private int GetCurrentLesson()
    {
        // to Ukraine timezone
        var now = _weekService.GetCurrentTime();

        var lessonsTimes = new[] { "08:30", "10:25", "12:10", "14:15", "16:10", "18:30", "20:00" };

        for (var i = 0; i < lessonsTimes.Length; i++)
        {
            var currTimestamp = TimeOnly.Parse(lessonsTimes[i]);
            if (now > currTimestamp)
            {
                // last lesson
                if (i >= lessonsTimes.Length - 1)
                {
                    return 0;
                }

                var nextTimestamp = TimeOnly.Parse(lessonsTimes[i + 1]);
            }
        }
    }
}

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

        if (now < nextTimestamp)
        {
            return i + 1;
        }
    }
    return 0;
}
}

```

Файл Program.cs

```

using Campus.Core.Repositories;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;

namespace Campus.Schedule.API;

public class Program
{
    public static void Main(string[] args)
    {
        CreateHostBuilder(args).Build().Run();
    }

    public static IHostBuilder CreateHostBuilder(string[] args) =>
        Host.CreateDefaultBuilder(args)
            .ConfigureWebHostDefaults(webBuilder => { webBuilder.UseStartup<Startup>(); })
            .ConfigureServices( (_, services) =>
            {
                services.AddScoped<IScheduleRepository, ScheduleRepository>();
            })
            .ConfigureAppConfiguration((hostingContext, config) =>
            {
                config
                    .AddEnvironmentVariables();
            })
            );
}

```

Файл Settings.cs

```

namespace Campus.Schedule.API;

public sealed record Settings
{
    public string ConnectionString { get; set; }
    public string Version => "0.9.0-beta (2023-04-27 22:00)";
    public int WeekOrder { get; set; } // 1 or 2
    public bool SummerTime { get; set; }
}

```

Файл Startup.cs

					КПІ.ІТ-9203.045440.03.12	Арк.
Вмін.	Арк.	№ докум.	Підп.	Дата.		9

```

using Campus.Core.Data;
using Campus.Core.Mappers;
using Campus.Core.Repositories;
using Campus.DAL;
using Campus.DAL.Queries;
using Campus.Schedule.API.Services;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
using Microsoft.OpenApi.Models;

namespace Campus.Schedule.API;

public class Startup
{
    public IConfiguration Configuration { get; }
    public IWebHostEnvironment WebHostEnvironment { get; }

    public Startup(IConfiguration configuration, IWebHostEnvironment webHostEnvironment)
    {
        Configuration = configuration;
        WebHostEnvironment = webHostEnvironment;
    }

    // This method gets called by the runtime. Use this method to add services to the container.
    public void ConfigureServices(IServiceCollection services)
    {
        var settings = LoadSettings(Configuration);

        services
            .AddSingleton<Settings>(settings)
            .AddSwaggerGen(c =>
            {
                c.SwaggerDoc("v1", new OpenApiInfo { Title = "Campus.Schedule.API", Version = "v1" });
            })
            .AddSingleton<QueryManager>(p =>
            {
                var logger = p.GetService<ILogger<QueryManager>>();
                return new QueryManager(WebHostEnvironment.ContentRootPath, logger);
            })
            .AddSingleton<ScheduleQueryManager>()
            .AddSingleton<ScheduledMapper>()
            .AddSingleton<LecturerMapper>()
            .AddSingleton<GroupMapper>()
            .AddSingleton<FacultyMapper>()
            .AddScoped<IDatabase, Database>(p =>
            {
                var logger = p.GetRequiredService<ILogger<PostgresDatabase>>();

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

        return new PostgresDatabase(settings.ConnectionString, logger);
    })
    .AddScoped<IScheduleRepository, ScheduleRepository>()
    .AddScoped<ScheduleWeekService>()
    .AddControllers();
}

private static Settings LoadSettings(IConfiguration configuration)
{
    var settings = new Settings();
    configuration.Bind(settings);
    settings.ConnectionString = configuration.GetConnectionString("DefaultConnection");
    return settings;
}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }

    app.UseSwagger();
    app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.json", "Campus.Schedule.API v1"));

    // global cors policy
    app.UsePathBase(new PathString("/api"));

    app.UseCors(x => x
        .AllowAnyMethod()
        .AllowAnyHeader()
        .SetIsOriginAllowed(origin => true) // allow any origin
        .AllowCredentials()); // allow credentials

    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints => { endpoints.MapControllers(); });
}
}

```

Файл ScheduleWeekService.cs

```

using System;

namespace Campus.Schedule.API.Services;

public class ScheduleWeekService
{
    private readonly int _weekOrder;
    private readonly bool _summerTime;

    public ScheduleWeekService(Settings settings)

```

					КПІ.ІТ-9203.045440.03.12	Арк.
Вмін.	Арк.	№ докум.	Підп.	Дата.		11

```

{
    if (settings.WeekOrder != 1 && settings.WeekOrder != 2)
    {
        throw new Exception("Week set incorrectly");
    }

    _weekOrder = settings.WeekOrder;
    _summerTime = settings.SummerTime;
}

public int GetCurrentWeek()
{
    var semesterStartTime = new DateTime(2021, 9, 1);
    var weeks = (DateTime.Now - semesterStartTime).Days / 7;
    var weekNumber = (weeks % _weekOrder);

    return weekNumber;
}

public TimeOnly GetCurrentTime()
{
    var hoursToAdd = _summerTime ? 2 : 1;
    return TimeOnly.FromDateTime(DateTime.UtcNow.AddHours(hoursToAdd));
}
}

```

Файл CurrentDayResponse.cs

```
namespace Campus.Schedule.API.Models;
```

```

public record CurrentDayResponse
{
    public int CurrentWeek { get; set; }
    public int CurrentDay { get; set; }
    public int CurrentLesson { get; set; }
}

```

Файл ExamWithDeadline.cs

```
using Campus.Models.Schedule;
```

```
namespace Campus.Schedule.API.Models;
```

```

public record ExamWithDeadline : ScheduleExam
{
    public ExamWithDeadline(ScheduleExam baseExam, int daysLeft)
    {
        Id = baseExam.Id;
        Date = baseExam.Date;
        LecturerId = baseExam.LecturerId;
        LecturerName = baseExam.LecturerName;
        Group = baseExam.Group;
        DaysLeft = daysLeft;
        Room = baseExam.Room;
    }
}

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

        Subject = baseExam.Subject;
        SubjectShort = baseExam.SubjectShort;
    }

    public int DaysLeft { get; set; }
}

```

Файл ScheduleResponse.cs

```

using System.Collections.Generic;

namespace Campus.Schedule.API.Models;

public record ScheduleResponse
{
    public ScheduleResponse(string groupCode)
    {
        GroupCode = groupCode;
        ScheduleFirstWeek = new List<GroupScheduleDayItem>();
        ScheduleSecondWeek = new List<GroupScheduleDayItem>();
    }

    public string GroupCode { get; set; }
    public List<GroupScheduleDayItem> ScheduleFirstWeek { get; set; }

    public List<GroupScheduleDayItem> ScheduleSecondWeek { get; set; }
}

public record LecturerScheduleResponse
{
    public LecturerScheduleResponse(string lecturerName)
    {
        LecturerName = lecturerName;
        ScheduleFirstWeek = new List<LecturerScheduleDayItem>();
        ScheduleSecondWeek = new List<LecturerScheduleDayItem>();
    }

    public string LecturerName { get; set; }
    public List<LecturerScheduleDayItem> ScheduleFirstWeek { get; set; }

    public List<LecturerScheduleDayItem> ScheduleSecondWeek { get; set; }
}

public abstract record Lesson
{
    public string Type { get; set; }
    public string Time { get; set; }
    public string Name { get; set; }
    public string Place { get; set; }
    public string Tag { get; set; }
}

public record GroupLesson : Lesson

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

{
  public string TeacherName { get; set; }

  public string LecturerId { get; set; }
}

public record LecturerLesson : Lesson
{
  public string Group { get; set; }
}

public record GroupScheduleDayItem
{
  public string Day { get; set; }
  public IReadOnlyCollection<GroupLesson> Pairs { get; set; }
}

public record LecturerScheduleDayItem
{
  public string Day { get; set; }
  public IReadOnlyCollection<LecturerLesson> Pairs { get; set; }
}

```

Файл fullList.js

```

import Http from './index';

export const getAllLecturers = () => {
  return Http.get("/schedule/lecturer/list").then(({ data }) => data);
};

export const getAllGroups = () => {
  return Http.get("/schedule/groups").then(({ data }) => {
    return {
      data: data.data.map((row) => ({
        name: `${row.name.trim()} (${row.faculty.trim()}`,
        id: row.id,
      })),
    };
  });
};

```

Файл getCurrentDateValues.js

```

import Http from './index';

export const getCurrentDateValues = () => {
  return Http.get('/time/current')
    .then(res => res.data);
};

```

Файл index.js

```

import axios from 'axios';

let headers = {};

```

						КПІ.ІТ-9203.045440.03.12	Арк.
Вмін.	Арк.	№ докум.	Підп.	Дата.			14

```
export const Http = axios.create(  
  {  
    baseURL: "https://schedule.kpi.ua/api/",  
    headers  
  });
```

```
export default Http;
```

Файл schedule.js

```
import Http from './index';
```

```
export const getScheduleByLecturer = lecturerId => {  
  return Http.get('/schedule/lecturer?lecturerId=' + lecturerId)  
    .then(res => res.data);  
}
```

```
export const getScheduleByGroup = groupId => {  
  return Http.get('/schedule/lessons?groupId=' + groupId)  
    .then(res => res.data);  
}
```

```
export const getExamsByGroup = groupName => {  
  return Http.get('/exams/group?groupId=' + groupName)  
    .then(res => res.data);  
}
```

Файл App.js

```
import ScheduleRouter from './containers/router';  
import Navbar from './containers/navbar';  
import { Wrapper } from './app.style';  
import { WeekContextProvider } from './common/context/weekContext';  
import ThemeContextProvider from './common/context/themeContext';  
import { GroupContextProvider } from './common/context/groupContext';  
import { LecturerContextProvider } from './common/context/lecturerContext';  
import { PreloadedListsContextProvider } from './common/context/preloadedListsContext';  
import { useCurrentDateParams } from './common/utills/useCurrentDateParams';  
import ScrollToTop from './containers/scrollToTop/index';
```

```
function App() {  
  const { currentWeek } = useCurrentDateParams()  
  return (  
    <WeekContextProvider initialValue={currentWeek === 1 ? 'firstWeek' : 'secondWeek'}>  
      <PreloadedListsContextProvider>  
        <GroupContextProvider>  
          <ThemeContextProvider initialValue={null}>  
            <LecturerContextProvider>  
              <ScrollToTop>  
                <Wrapper>  
                  <Navbar />  
                  <ScheduleRouter />  
                </Wrapper>  
              </LecturerContextProvider>  
            </ThemeContextProvider>  
          </GroupContextProvider>  
        </PreloadedListsContextProvider>  
      </WeekContextProvider>  
    )
```

Вмін.	Арк.	№ докум.	Підп.	Дата.

```

    </ScrollToTop>
    </LecturerContextProvider>
    </ThemeContextProvider>
    </GroupContextProvider>
    </PreloadedListsContextProvider>
    </WeekContextProvider>
  );
}

```

```
export default App;
```

Файл app.style.js

```
import styled from 'styled-components';
import { getValueFromTheme } from '../common/utils/getValueFromTheme';
```

```
export const Wrapper = styled.div`
  background: ${getValueFromTheme('bgPrimary')};
  min-height: 105vh;
`;

```

Файл dayOptions.js

```
import { MODES } from '../common/constants/modes';
```

```
export const DAY_OPTIONS = {
  [MODES.EXTRA_SMALL]: [
    {label: 'ПН', value: '0-1'},
    {label: 'ВТ', value: '1-2'},
    {label: 'СР', value: '2-3'},
    {label: 'ЧТ', value: '3-4'},
    {label: 'ПТ', value: '4-5'},
    {label: 'СБ', value: '5-6'},
  ],
  [MODES.SMALL]: [
    {label: 'ПН-ВТ', value: '0-2'},
    {label: 'СР-ЧТ', value: '2-4'},
    {label: 'ПТ-СБ', value: '4-6'},
  ],
  [MODES.MEDIUM]: [
    {label: 'ПН-СР', value: '0-3'},
    {label: 'ЧТ-СБ', value: '3-6'},
  ]
};

```

Файл models.js

```
export const MODES = {
  EXTRA_SMALL: 'extraSmallMode',
  SMALL: 'smallMode',
  MEDIUM: 'mediumMode',
  BIG: 'bigMode',
};

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

Файл routes.js

```
export const routes = {
  GROUP: '/',
  LECTURER: '/lecturers',
  SESSION: '/sessions'
};
```

Файл scheduleParams.js

```
export const DAYS = [
  'Пн',
  'Вт',
  'Ср',
  'Чт',
  'Пт',
  'Сб'
];

export const TIME_POINTS = [
  '8.30',
  '10.25',
  '12.20',
  '14.15',
  '16.10',
  '18.30'
];
```

Файл selectOptions.js

```
export const getSelectCustomStyle = (theme) => ({
  option(base) {
    return {
      ...base,
      color: theme['primaryFontColor'],
    };
  },
  menu(base) {
    return {
      ...base,
      "& .fast-option-focused": {
        color: theme['hoveredOptionFontColor'],
      },
      "& .fast-option-selected": {
        color: theme['secondaryFontColor'],
        fontWeight: "bold"
        // background color can be also changed if needed
      },
      backgroundColor: theme['bgOptions'],
      zIndex: 1000,
    };
  },
  menuList(base) {
    return {
      ...base,
```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

    backgroundColor: theme['bgOptions'],
    zIndex: 1000,
  };
},
control(base) {
  return {
    ...base,
    backgroundColor: theme['bgOptions'],
    color: theme['primaryFontColor'],
    border: 'none',
  };
},
indicatorSeparator(base) {
  return {
    ...base,
    display: 'none'
  };
},
singleValue(base) {
  return {
    ...base,
    color: theme['primaryFontColor'],
  };
},
input(base) {
  return {
    ...base,
    color: theme['primaryFontColor'],
  };
}
});

```

Файл subjectTypes.js

```

import { ScheduleItemTypeLab, ScheduleItemTypeLec, ScheduleItemTypePrac } from
"../../components/scheduleItemContent/scheduleItemContent.style";

```

```

export const SUBJECT_TYPES = {
  lec: {
    component: ScheduleItemTypeLec,
    title: "Лек"
  },
  lab: {
    component: ScheduleItemTypeLab,
    title: "Лаб"
  },
  prac: {
    component: ScheduleItemTypePrac,
    title: "Прак"
  },
}

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

Файл theme.js

```
export const theme = {
  light: {
    bgPrimary: 'linear-gradient(90deg, #FFEFF2 0%, #DBE9FE 100%)',
    bgHeader: 'linear-gradient(180deg, #FFFFFF 0%, rgba(255, 255, 255, 0.5) 100%)',
    bgTable: 'rgba(255, 255, 255, 0.5)',
    bgCard: 'rgba(255, 255, 255, 0.9)',
    bgOptions: '#eff0f8',
    primaryFontColor: '#141518',
    secondaryFontColor: '#fff',
    hoveredOptionFontColor: '#141518',
    currentDayContainer: 'linear-gradient(180deg, rgba(223, 233, 253, 0) 0%, rgba(223, 233, 253, 0.5) 10.73%, #DFE9FD 55%, rgba(223, 233, 253, 0.5) 89.9%, rgba(223, 233, 253, 0) 100%)',
    svgPrimaryFilter: "",
  },
  dark: {
    bgPrimary: 'linear-gradient(120.84deg, #0E0F12 -0.88%, #111316 100%)',
    bgHeader: 'linear-gradient(180deg, #1F2128 0%, rgba(31, 33, 40, 0.5) 100%)',
    bgTable: 'rgba(31, 33, 40, 0.5)',
    bgCard: 'rgba(31, 33, 40, 0.9)',
    bgOptions: '#363637',
    primaryFontColor: '#fff',
    secondaryFontColor: '#21232a',
    hoveredOptionFontColor: '#21232a',
    currentDayContainer: 'linear-gradient(180deg, rgba(54, 54, 55, 0) 0%, rgba(54, 54, 55, 0.5) 10.73%, #363637 55%, rgba(54, 54, 55, 0.5) 89.9%, rgba(54, 54, 55, 0.29) 100%)',
    svgPrimaryFilter: 'invert(100%) sepia(0%) saturate(0%) hue-rotate(46deg) brightness(100%) contrast(103%)',
  }
};
```

Файл groupContext.js

```
import { createContext, useContext, useState } from "react";

const GroupContext = createContext(null);

export const useGroupContext = () => useContext(GroupContext);

export const GroupContextProvider = ({ initialValue, children }) => {
  const [group, setGroup] = useState(initialValue);

  const params = { setGroup, group };

  return (
    <GroupContext.Provider value={params}>{children}</GroupContext.Provider>
  );
};
```

Файл lecturerContext.js

```
import { createContext, useContext, useState } from 'react';

const LecturerContext = createContext(null);
```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```
export const useLecturerContext = () => useContext(LecturerContext);
```

```
export const LecturerContextProvider = ({children}) => {
```

```
  const [lecturer, setLecturer] = useState(null);
```

```
  const params = {setLecturer, lecturer};
```

```
  return (
```

```
    <LecturerContext.Provider value={params}>
```

```
      {children}
```

```
    </LecturerContext.Provider>
```

```
  );
```

```
};
```

Файл `preloadedListsContext.js`

```
import { createContext, useContext, useEffect, useState } from 'react';
```

```
import { getAllGroups, getAllLecturers } from '../api/fullList';
```

```
const PreloadedListsContext = createContext(null);
```

```
export const usePreloadedListContext = () => useContext(PreloadedListsContext);
```

```
export const PreloadedListsContextProvider = ({children}) => {
```

```
  const [lists, setLists] = useState({
```

```
    groups: [],
```

```
    lecturers: [],
```

```
  });
```

```
  const promises = [getAllGroups(), getAllLecturers()];
```

```
  useEffect(() => {
```

```
    Promise.all(promises)
```

```
      .then(([groups, lecturers]) => {
```

```
        setLists({
```

```
          groups: groups.data,
```

```
          lecturers: lecturers.data,
```

```
        });
```

```
      });
```

```
    }, []);
```

```
  return (
```

```
    <PreloadedListsContext.Provider value={lists}>{children}</PreloadedListsContext.Provider>
```

```
  );
```

```
};
```

Файл `sliceOptionsContext.js`

```
import { createContext, useContext } from 'react';
```

```
const SliceOptionsContext = createContext(null);
```

```
export const useSliceOptionsContext = () => useContext(SliceOptionsContext);
```

```
export default SliceOptionsContext.Provider;
```

Вмін.	Арк.	№ докум.	Підп.	Дата.

Файл themeContext.js

```
import { ThemeProvider } from 'styled-components';
import { createContext, useContext, useEffect, useState } from 'react';
import { theme } from '../common/constants/theme';
import { getLocalStorageItem, setLocalStorageItem } from '../utils/parsedLocalStorage';

const ThemeSelectorContext = createContext(null);

export const useThemeSelectorContext = () => useContext(ThemeSelectorContext);

const ThemeContextProvider = ({ children }) => {
  const [currentTheme, setTheme] = useState('light');
  useEffect(() => {
    const localStorageTheme = getLocalStorageItem("theme")
    const prefersLight = window.matchMedia('(prefers-color-scheme: light)').matches;
    if (localStorageTheme) {
      changeTheme(localStorageTheme === 'light')
    }
    else {
      changeTheme(prefersLight)
    }
  }, [])

  const changeTheme = isLightTheme => {
    const newTheme = isLightTheme ? 'light' : 'dark'
    setTheme(newTheme);
    setLocalStorageItem("theme", newTheme)
  }
  return (
    <ThemeProvider theme={theme[currentTheme]}>
      <ThemeSelectorContext.Provider value={changeTheme}>
        {children}
      </ThemeSelectorContext.Provider>
    </ThemeProvider>
  );
};

export default ThemeContextProvider;
```

Файл weekContext.js

```
import { createContext, useContext, useState } from 'react';

const WeekContext = createContext(null);

export const useWeekContext = () => useContext(WeekContext);

export const WeekContextProvider = ({ initialValue, children }) => {
  const [currentWeek, setCurrentWeek] = useState(initialValue);

  const params = { setCurrentWeek, currentWeek };

  return (
```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```
<WeekContext.Provider value={params}>
  {children}
</WeekContext.Provider>
);
};
```

Файл styles.js

```
import styled, { css } from 'styled-components';
import { flexbox, space } from 'styled-system';
import { MODES } from '../constants/modes';
import { getModeSize } from '../utils/getModeSize';
import { getValueFromTheme } from '../common/utils/getValueFromTheme';
import { Link } from 'react-router-dom';

export const Flex = styled.div`
  display: flex;
  ${space};
  ${flexbox};

  // Support for gap in flexbox
  gap: ${props => props.gap};
`;

export const Pictogram = styled.img`
  filter: ${getValueFromTheme('svgPrimaryFilter')};
`;

export const media = {
  [MODES.EXTRA_SMALL]: `@media(max-width: ${getModeSize(MODES.EXTRA_SMALL)})`,
  [MODES.SMALL]: `@media(max-width: ${getModeSize(MODES.SMALL)})`,
  [MODES.MEDIUM]: `@media(max-width: ${getModeSize(MODES.MEDIUM)})`,
  [MODES.BIG]: `@media(min-width: ${getModeSize(MODES.BIG)})`,
};

export const Logo = styled(Pictogram)`
  max-width: 92px;
  max-height: 92px;
  width: 100%;
`;

export const ScheduleItemMixin = css`
  background: ${getValueFromTheme('bgCard')};
  box-shadow: 0 4px 16px rgba(0, 0, 0, 0.04);
  border-radius: 16px;
`;

export const UnstyledLink = styled(Link)`
  text-decoration: none;
  color: ${getValueFromTheme('primaryFontColor')};
`;
```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

КПІ.ІТ-9203.045440.03.12

Арк.

22

Файл apiTransformers.js

```
import { unique } from './unique';

export const prepareList = (list) => {
  return unique(
    list.map(item => ({
      label: item.name,
      value: item.id,
    }))
    .filter(item => item.label !== '' && item.value !== '')
    .sort((a, b) => {
      return a.label.localeCompare(b.label, "uk")
    }),
    (item => item.label));
};
```

Файл generatedScheduleMatrix.js

```
import { DAYS, TIME_POINTS } from '../common/constants/scheduleParams';
import { getActiveTimePoint } from '../common/utils/getActiveTimePoint';
import moment from 'moment';

export const generateScheduleMatrix = weekSchedule => {
  const scheduleMatrix = new Array(TIME_POINTS.length).fill(null).map(() => new Array(DAYS.length).fill(null));
  const activePair = getActiveTimePoint();
  const currentDay = moment().day() - 1;

  weekSchedule.forEach(day => {
    const yIndex = DAYS.findIndex(item => item === day.day);

    day.pairs.forEach(pair => {
      const xIndex = TIME_POINTS.findIndex(item => item === pair.time);
      const cell = scheduleMatrix[xIndex][yIndex];
      let newCell = { ...pair, currentDay: activePair !== -1 && currentDay === yIndex && activePair === xIndex };

      if (cell) {
        let extendedCell = [];

        if (Array.isArray(cell)) {
          extendedCell = [...cell]
        } else {
          extendedCell = [cell]
        }

        extendedCell.push({ ...pair, currentDay: newCell.currentDay });
        newCell = extendedCell;
      }

      scheduleMatrix[xIndex][yIndex] = newCell;
    });
  });

  return scheduleMatrix;
};
```

Вмін.	Арк.	№ докум.	Підп.	Дата.

```
};
```

Файл `getActiveTimePoint.js`

```
import moment from 'moment';
import { TIME_POINTS } from '../common/constants/scheduleParams';

export const getActiveTimePoint = () => {
  const PAIR_DURATION_IN_MINUTES = 95;
  const now = moment();

  return TIME_POINTS.findIndex((timePoint, i) => {
    const timeNumbers = timePoint.split('.').map(item => +item);
    const pairStartDate = moment().set({hours: timeNumbers[0], minutes: timeNumbers[1], seconds: 0});
    const pairEndDate = moment(pairStartDate).set({minutes: timeNumbers[1] + PAIR_DURATION_IN_MINUTES});

    return now.isBetween(pairStartDate, pairEndDate);
  });
};
```

Файл `getApiFunction.js`

```
import { routes } from '../constants/routes';
import { getScheduleByGroup, getScheduleByLecturer, getExamsByGroup } from '../api/schedule';
import { getAllGroups, getAllLecturers } from '../api/fullList';

export const getScheduleApiFunction = route => {
  const functions = {
    [routes.GROUP]: getScheduleByGroup,
    [routes.SESSION]: getExamsByGroup,
    [routes.LECTURER]: getScheduleByLecturer,
  };

  return functions[route];
}

export const getListApiFunction = route => {
  const functions = {
    [routes.GROUP]: getAllGroups,
    [routes.SESSION]: getAllGroups,
    [routes.LECTURER]: getAllLecturers,
  };

  return functions[route];
}
```

Файл `getModeSize.js`

```
import { MODES } from '../constants/modes';

export const getModeSize = MODE => {
  const MODE_SIZES = {
    [MODES.BIG]: '1441px',
    [MODES.MEDIUM]: '1440px',
    [MODES.SMALL]: '988px',
  };
};
```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```
[MODES.EXTRA_SMALL]: '639px',
}
```

```
return MODE_SIZES[MODE];
}
```

Файл `getValueFromTheme.js`

```
export const getValueFromTheme = key => props => {
  return props.theme[key];
};
```

Файл `parsedLocalStorage.js`

```
export const setLocalStorageItem = (key, value) => {
  localStorage.setItem(key, JSON.stringify(value))
}
```

```
export const getLocalStorageItem = (key) => {
  try {
    return JSON.parse(localStorage.getItem(key))
  }
  catch {
    // No `key` local storage item, so return undefined
    return undefined
  }
}
```

Файл `unique.js`

```
export const unique = (array, getComparingItem = (item) => item) => {
  const uniqueFilter = (value, index, self) => {
    return self.map(e => getComparingItem(e)).indexOf(getComparingItem(value)) === index;
  }

  return [...array.filter(uniqueFilter)];
}
```

Файл `useCurrentDateParams.js`

```
import { useEffect, useState } from 'react';
import { getCurrentDateValues } from '../api/getCurrentDateValues';

export const useCurrentDateParams = () => {
  const [dateParams, setDateParams] = useState({currentWeek: null, currentDay: null, currentLesson: null});

  useEffect(() => {
    getCurrentDateValues()
      .then(({ data }) => setDateParams({...data, currentDay: data.currentDay - 1}));
  }, []);

  return dateParams;
};
```

Файл `useCurrentMode.js`

```
import { MODES } from "../constants/modes";
```

Вмін.	Арк.	№ докум.	Підп.	Дата.

```

import { getModeSize } from "./getModeSize";
import { useEffect, useState } from 'react';

export const useCurrentMode = () => {
  const [currentMode, setCurrentMode] = useState(null);

  useEffect(() => {
    detectCurrentMode();
    window.addEventListener('resize', detectCurrentMode);

    return () => window.removeEventListener('resize', detectCurrentMode);
  }, []);

  const detectCurrentMode = () => {
    for (let key in MODES) {
      if (window.innerWidth <= parseInt(getModeSize(MODES[key]))) {
        setCurrentMode(MODES[key]);
        return;
      }
    }

    setCurrentMode(MODES.BIG);
  }

  return currentMode;
}

```

Файл entitySearch.jsx

```

import React, { useEffect, useState } from "react";
import { Select } from "react-select-virtualized";

import { Label } from "./entitySearch.style";

import { useLecturerContext } from "../../common/context/lecturerContext";
import { useHistory, useLocation } from "react-router-dom";
import { usePreloadedListContext } from "../../common/context/preloadedListsContext";
import { useGroupContext } from "../../common/context/groupContext";
import { useTheme } from "styled-components";

import { prepareList } from "../../common/utills/apiTransformers";
import { routes } from "../../common/constants/routes";

import { getSelectCustomStyle } from "../../common/constants/selectOptions";
import "./entitySearch.scss";
import { getLocalStorageItem, setLocalStorageItem } from "../../common/utills/parsedLocalStorage";

const useQuery = () => {
  const { search } = useLocation();
  return React.useMemo(() => new URLSearchParams(search), [search]);
};

const EntitySearch = () => {

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

const theme = useTheme();
const location = useLocation();
const history = useHistory();

const [options, setOptions] = useState([]);
const { lecturer, setLecturer } = useLecturerContext();
const { group, setGroup } = useGroupContext();
const lists = usePreloadedListContext();

const isLecturer = location.pathname.includes(routes.LECTURER);
const list = isLecturer ? lists.lecturers : lists.groups;

const query = useQuery();

useEffect(() => {
  if (isLecturer) {
    setGroup(null);
  } else {
    setLecturer(null);
  }
}, [isLecturer, setGroup, setLecturer]);

useEffect(() => {
  if (isLecturer) {
    let lecturer = query.get("lecturerId");
    const localStorageLecturerId = getLocalStorageItem("lecturerId")
    if(!lecturer && localStorageLecturerId){
      lecturer = localStorageLecturerId
      history.replace("?lecturerId=" + localStorageLecturerId);
    }
    setLecturer(lecturer);
    setGroup(null);
  } else {
    let group = query.get("groupId");
    const localStorageLecturerId = getLocalStorageItem("groupId")
    if(!group && localStorageLecturerId){
      group = localStorageLecturerId
      history.replace("?groupId=" + localStorageLecturerId);
    }
    const groupObj = list.find((g) => g.id === group);
    setGroup(groupObj);
    setLecturer(null);
  }
  setOptions(prepareList(list));
}, [list, history, isLecturer, query, setGroup, setLecturer]);

const onOptionChange = (option) => {
  isLecturer ? setLecturer(option.value) : setGroup({ id : option.value, name : option.label });

  if (isLecturer) {
    history.push("?lecturerId=" + option.value);
    setLocalStorageItem("lecturerId", option.value)
  } else {

```

```

    history.push("?groupId=" + option.value);
    setLocalStorageItem("groupId", option.value)
  }
};
const initialValue =
  options.find((item) =>
    isLecturer ? item.value === lecturer : item.value === group?.id
  ) ?? null;
return (
  <Label alignItems="center" gap="15px">
    Розклад занять для
    <div style={{ width: "200px" }}>
      <Select
        options={options}
        onChange={onOptionChange}
        styles={getSelectCustomStyle(theme)}
        value={initialValue}
        isSearchable={true}
        isClearable={false}
        placeholder={null}
        name="color"
      />
    </div>
  </Label>
);
};

export default EntitySearch;

```

Файл exam.jsx

```

import { CardDate, CardMainData, CardWrapper, Divider, DividerRed, Location, Subject, Teacher, } from './exam.style';
import teacherIcon from '../assets/icons/teacher.svg';
import locationIcon from '../assets/icons/location.svg';
import clock from '../assets/icons/clock.svg';
import { Flex, Pictogram } from '../common/styles/styles';

const ExamComponent = ({data}) => {

  const subject = data && data.subject;
  const teacher = data && data.lecturerName;
  const location = data && data.room;
  const date = data && new Date(data.date);
  const daysLeft = data && data.daysLeft;
  const day = date.getDate();

  return (
    <div>
      <CardWrapper>
        <Flex>
          <DividerRed/>
          <CardMainData>
            <Subject>{subject}</Subject>

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

    <Teacher>
      <Pictogram src={teacherIcon} alt="teacher"/>
      {teacher}
    </Teacher>
  <Teacher>
    <Pictogram src={clock} alt="time"/>
    {date.toLocaleTimeString('en-US', {timeStyle: 'short'})}
  </Teacher>
  <Location>
    <Pictogram src={locationIcon} alt="location"/>
    {location}
  </Location>
</CardMainData>
</Flex>
<Flex>
  <Divider/>
  <CardDate>
    <h3>{date.getFullYear()}</h3>
    <h2> {day < 10 ? '0' + day : day} Січня</h2>
    { /* TODO calculate */ }
    <h3>{daysLeft} днів до початку</h3>
  </CardDate>
</Flex>
</CardWrapper>
</div>
);
};

export default ExamComponent;

```

Файл optionMultiplyToogler.jsx

```

import { ToggleOption, ToggleWrapper } from './optionMultipleToggle.style';
import { useEffect, useState } from 'react';

```

```

const OptionMultipleToggle = ({ options, onOptionChange, initialValue }) => {
  const [selectedItem, selectItem] = useState(initialValue);

```

```

  useEffect(() => {
    selectItem(initialValue);

```

```

    return () => onOptionChange(null);
  }, [initialValue]);

```

```

  useEffect(() => {
    onOptionChange(selectedItem);
  }, [selectedItem]);

```

```

const optionChangeHandler = (value, item) => () => {
  selectItem(state => {
    if (state !== value) {
      return value;
    }
  })
}

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

    return state;
  });

};

const togglerOptions = (options || []).map(item => {
  return (
    <ToggleOption key={item.value}
      active={selectedItem === item.value}
      onClick={optionChangeHandler(item.value)}>
      {item.label}
    </ToggleOption>
  );
});

return (
  <ToggleWrapper>
    {togglerOptions}
  </ToggleWrapper>
);
};

export default OptionMultipleToggle;

```

Файл `scheduleItemContent.jsx`

```

import { GroupName, Location, ScheduleItemCurrent, ScheduleItemHeader, Subject, Teacher, LocationLink } from
'./scheduleItemContent.style';
import teacherIcon from '../assets/icons/teacher.svg';
import locationIcon from '../assets/icons/location.svg';
import { Pictogram, UnstyledLink } from '../common/styles/styles';
import { routes } from '../common/constants/routes';
import { usePreloadedListContext } from '../common/context/preloadedListsContext';
import { unique } from '../common/utills/unique';
import { useEffect, useState } from 'react';
import { useLecturerContext } from '../common/context/lecturerContext';
import { useGroupContext } from '../common/context/groupContext';
import { setLocalStorageItem } from '../common/utills/parsedLocalStorage';
import { SUBJECT_TYPES } from '../common/constants/subjectTypes';

```

```

const ScheduleItemContent = ({scheduleItemData, collapsed}) => {
  const subject = scheduleItemData?.name;
  const teacher = scheduleItemData?.teacherName;
  const teacherId = scheduleItemData?.lecturerId;
  const groups = unique((scheduleItemData?.group || "").split(','));
  const tag = scheduleItemData?.tag;
  const allGroups = usePreloadedListContext().groups;
  const [location, setLocation] = useState(scheduleItemData && scheduleItemData.place)
  const {setLecturer} = useLecturerContext()
  const {setGroup} = useGroupContext()

  const ScheduleItemComponent = SUBJECT_TYPES[tag]?.component

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

const scheduleItemTitle = SUBJECT_TYPES[tag]?.title

useEffect(() => {
  if(location.endsWith('-')){
    setLocation(location.slice(0, -1))
  }
}, [location, setLocation])

const handleLecturerClick = () => {
  setLocalStorageItem("lecturerId", teacherId)
  setLecturer(teacherId)
}

const handleGroupClick = (groupId) => {
  return () => {
    const groupFound = allGroups.find(allGroupsItem => allGroupsItem.name.replace(" ", "") === groupId)
    if(groupFound){
      setLocalStorageItem("groupId", groupFound.id)
      setGroup(groupFound.id)
    }
  }
}

const getGroupIdByName = (groupName) => {
  const groupFound = allGroups.find(allGroupsItem => allGroupsItem.name.replace(" ", "") === groupName)
  if(groupFound){
    return groupFound.id
  }
  else{
    return ""
  }
}

const getGroup = (group, index, length) => {
  return `${group.split("(")[0]}${index === length - 1 ? "" : ","}`
}

const getLocationLink = () => {
  return `https://kpi.ua/k-${parseInt(location.split("-")[0])}`
}

const getGroupLink = (group) => {
  return `${routes.GROUP}?groupId=${getGroupIdByName(group)}`
}

return (
  <div>
    <ScheduleItemHeader>
      {ScheduleItemComponent && <ScheduleItemComponent>{scheduleItemTitle}</ScheduleItemComponent>}
      {scheduleItemData.currentDay && <ScheduleItemCurrent>3APA3</ScheduleItemCurrent>}
    </ScheduleItemHeader>
    <Subject>
      {subject}

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------


```

const ScheduleTypeTab = ({tabClick, activeTab, children, url}) => {
  const location = useLocation();

  const isActive = location.pathname === url;
  const { group } = useGroupContext();
  const { lecturer } = useLecturerContext()
  const [urlWithParams, setUrlWithParams] = useState(url)

  useEffect(() => {
    if(url.includes(routes.LECTURER)){
      const localStorageLecturerId = getLocalStorageItem("lecturerId")
      if(localStorageLecturerId){
        setUrlWithParams(`${url}?groupId=${localStorageLecturerId}`)
      }
    }
    else if(url.includes(routes.GROUP) || url.includes(routes.SESSION)){
      const localStorageGroupId = getLocalStorageItem("groupId")
      if(localStorageGroupId){
        setUrlWithParams(`${url}?groupId=${localStorageGroupId}`)
      }
    }
  }, [url, setUrlWithParams, isActive, group, lecturer])

  return (
    <Tab active={isActive} onClick={tabClick}>
      <UnstyledLink to={urlWithParams}>
        {children}
      </UnstyledLink>
    </Tab>
  );
};

export default ScheduleTypeTab;

```

Файл switch.jsx

```

import './switch.scss';
import { useTheme } from 'styled-components';
import { useEffect, useState } from 'react';
import clsx from 'clsx';
import { getLocalStorageItem } from '../../common/utils/parsedLocalStorage';

const Switch = ({onChange}) => {
  const theme = useTheme();

  const [checked, setChecked] = useState(undefined);

  useEffect(() => {
    const localStorageTheme = getLocalStorageItem("theme")
    const prefersLight = window.matchMedia('(prefers-color-scheme: light)').matches;
    if(localStorageTheme){

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

    setChecked(localStorageTheme === 'light')
  }
  else{
    setChecked(prefersLight)
  }
}, [])

const onCheckboxChange = event => {
  const newState = event.target.checked;
  setChecked(event.target.checked);

  onChange(newState);
};

return (
  <
  <label className="switch">
    <input onChange={onCheckboxChange} checked={checked === undefined ? true : checked} type="checkbox"/>
    <span style={{ 'backgroundColor': theme['bgOptions']}}
      className={clsx(
        { "slider round": true},
        { "hidden": checked === undefined }
      )}/>
    <span className={clsx(
      { "checker": true},
      { "hidden": checked === undefined }
    )}/>
  </label>
  </>
);
};

export default Switch;

```

Файл timeDivider.jsx

```
import { Container, Divider, InnerContainer, TimeCell } from './timeDivider.style';
```

```

const TimeDivider = ({children}) => {
  return (
    <Container>
      <InnerContainer alignItems="center">
        <TimeCell>{children}</TimeCell>
        <Divider/>
      </InnerContainer>
    </Container>
  );
};

```

```
export default TimeDivider;
```

Файл weekSwitch.jsx

```
import OptionMultipleToggler from "../components/optionMultipleToggler";
```

					КПІ.ІТ-9203.045440.03.12	Арк.
Вмін.	Арк.	№ докум.	Підп.	Дата.		34

```

import { useWeekContext } from "../../common/context/weekContext";
import { useEffect, useState } from "react";

const WeekSwitch = ({ type }) => {
  const { setCurrentWeek, currentWeek } = useWeekContext();

  const updateWeek = (value) => {
    if (value) {
      setCurrentWeek(value);
    }
  };

  const weeks = [
    { label: "Перший тиждень", value: "firstWeek" },
    { label: "Другий тиждень", value: "secondWeek" },
  ];

  const semesters = [
    { label: "Перший семестр", value: "firstWeek" },
    { label: "Другий семестр", value: "secondWeek" },
  ];

  return (
    <OptionMultipleToggler
      initialValue={currentWeek}
      options={type === "weeks" ? weeks : semesters}
      onOptionChange={updateWeek}
    />
  );
};

export default WeekSwitch;

```

Файл mainSettings.jsx

```

import ScheduleTypeTabs from "../../scheduleTypeTabs";
import EntitySearch from "../../components/entitySearch";
import WeekSwitch from "../../components/weekSwitch";
import { MainSettingsContainer } from "../../mainSettings.style";
import { useLocation } from "react-router-dom";
import { useEffect, useState } from "react";

const MainSettings = () => {
  const { pathname } = useLocation();
  const [weekAvailable, setWeekAvailable] = useState(true);

  useEffect(() => {
    if (pathname.includes("sessions")) {
      setWeekAvailable(false);
    } else {
      setWeekAvailable(true);
    }
  }, [pathname]);

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

return (
  <MainSettingsContainer
    flexDirection="column"
    alignItems="center"
    gap="24px"
  >
    <ScheduleTypeTabs />
    <EntitySearch />
    <div style={{ width: "100%" }}>
      <WeekSwitch type={weekAvailable ? "weeks" : "semesters"}></WeekSwitch>
    </div>
  </MainSettingsContainer>
);
};

export default MainSettings;

```

Файл navbar.jsx

```

import { Logo } from '../common/styles/styles';
import MainSettings from '../mainSettings';
import { NavBarInner, NavBarWrapper } from './navbar.style';
import Switch from '../components/switch';
import logo from '../assets/logo.svg';
import { useThemeSelectorContext } from '../common/context/themeContext';

const Navbar = () => {
  const selectTheme = useThemeSelectorContext();

  return (
    <NavBarWrapper>
      <NavBarInner justifyContent="space-between" mx={2} gap={'20px'}>
        <Logo src={logo} alt={'logo'} />
        <MainSettings />
        <Switch onChange={selectTheme} />
      </NavBarInner>
    </NavBarWrapper>
  );
};

export default Navbar;

```

Файл scheduleRouter.jsx

```

import { Redirect, Route, Switch } from 'react-router-dom';
import ScheduleWrapper from './scheduleWrapper';
import ExamsWrap from './scheduleExams';
import { routes } from '../common/constants/routes';
import { getScheduleApiFunction } from '../common/utills/getApiFunction';

const ScheduleRouter = () => {
  return (
    <Switch>
      <Route exact path={routes.GROUP}>

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

    <ScheduleWrapper contextType="group" getData={getScheduleApiFunction(routes.GROUP)}/>
  </Route>
  <Route exact path={routes.SESSION}>
    <ExamsWrap contextType="group" getData={getScheduleApiFunction(routes.SESSION)}/>
  </Route>
  <Route exact path={routes.LECTURER}>
    <ScheduleWrapper contextType="lecturer" getData={getScheduleApiFunction(routes.LECTURER)}/>
  </Route>
  <Route>
    <Redirect to={routes.GROUP}/>
  </Route>
</Switch>
);
};

export default ScheduleRouter;

```

Файл schedule.jsx

```

import React from 'react';
import TimeDivider from '../components/timeDivider';
import { CurrentDayContainer, GridContainer } from './schedule.style';
import ScheduleHeader from './scheduleHeader';
import { TIME_POINTS } from '../common/constants/scheduleParams';
import { useSliceOptionsContext } from '../common/context/sliceOptionsContext';
import { useCurrentDateParams } from '../common/utills/useCurrentDateParams';

const Schedule = ({children}) => {
  const sliceOptions = useSliceOptionsContext();
  const { currentDay } = useCurrentDateParams();

  const dynamicGeneratedTable = React.Children.map(children, (child, index) => {

    return (
      <
        <TimeDivider>{TIME_POINTS[index]}</TimeDivider>
        {child}
      </>
    );
  });

  const isDayInSlice = () => {
    if (sliceOptions) {
      return currentDay >= sliceOptions.start && currentDay < sliceOptions.end;
    }

    return true;
  };

  const gridDayStart = currentDay + 1;

  return (
    <GridContainer>

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

    {gridDayStart > 0 && isDayInSlice() ? <CurrentDayContainer start={sliceOptions ? gridDayStart - sliceOptions.start : gridDayStart}/> :
null}
    <ScheduleHeader/>
    {dynamicGeneratedTable}
  </GridContainer>
);
};

export default Schedule;

```

Файл scheduleDayToogler.jsx

```

import OptionMultipleToggler from '../components/optionMultipleToggler';
import { MODES } from '../common/constants/modes';
import { useCurrentMode } from '../common/utis/useCurrentMode';
import { DAY_OPTIONS } from '../common/constants/dayOptions';
import { ScheduleDayTogglerContainer } from './scheduleDayToggler.style';
import { useCurrentDateParams } from '../common/utis/useCurrentDateParams';

const ScheduleDayToggler = ({handler}) => {
  const mode = useCurrentMode();
  const {currentDay} = useCurrentDateParams()
  const getDayOption = () => {
    if(!currentDay || currentDay === 6){
      return 0
    }
    if(mode === MODES.EXTRA_SMALL){
      return currentDay
    }
    else if(mode === MODES.SMALL){
      return Math.floor(currentDay / 2)
    }
    else if(mode === MODES.MEDIUM){
      return Math.floor(currentDay / 3)
    }
    return 0
  }
  return (
    mode && mode !== MODES.BIG ?
    <ScheduleDayTogglerContainer>
      <OptionMultipleToggler initialValue={DAY_OPTIONS[mode] && DAY_OPTIONS[mode][getDayOption()].value}
onOptionChange={handler} options={DAY_OPTIONS[mode]}/>
    </ScheduleDayTogglerContainer> :
    null
  );
};

export default ScheduleDayToggler;

```

Файл scheduleExams.jsx

```

import { GridWrapper, Header, WordAccent } from './scheduleWrapper.style';
import Exam from '../components/examComponent';

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

import { useGroupContext } from '../common/context/groupContext';
import { useEffect, useState } from 'react';

const SchedulesExamsWrapper = ({ getData }) => {
  const [data, setData] = useState(null);

  const { group } = useGroupContext();

  useEffect(() => {
    const contextValue = group;

    if (contextValue) {
      getData(contextValue.id).then((res) => setData(res.data));
    } else {
      setData(null);
    }
  }, [group]);

  return (
    data ?
    <div style={{ overflow: 'hidden' }}>
      <GridWrapper>
        <Header>
          Розклад сесії для групи <WordAccent>{group.name}</WordAccent><br/>
          на <WordAccent>перший семестр{' '</WordAccent>
        </Header>
        {data && data.map((d, idx) => <Exam key={idx} data={d}/>)}
      </GridWrapper>
    </div> : null
  );
};

export default SchedulesExamsWrapper;

```

Файл scheduleHeader.jsx

```

import { WeekDay } from './scheduleHeader.style';
import { useSliceOptionsContext } from '../common/context/sliceOptionsContext';

const ScheduleHeader = () => {
  const sliceOptions = useSliceOptionsContext();

  const days = [
    {label: 'Понеділок', value: 'monday'},
    {label: 'Вівторок', value: 'tuesday'},
    {label: 'Середа', value: 'wednesday'},
    {label: 'Четвер', value: 'thursday'},
    {label: 'П'ятниця', value: 'friday'},
    {label: 'Субота', value: 'saturday'},
  ];

  const slicedDays = sliceOptions ? days.slice(sliceOptions.start, sliceOptions.end) : days;

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

const renderedWeekDays = slicedDays.map(item => {
  return <WeekDay key={item.value}>{item.label}</WeekDay>;
});

return (
  <>
    {renderedWeekDays}
  </>
);
};

export default ScheduleHeader;

```

Файл scheduleItem.jsx

```

import { ScheduleItemWrapper } from './scheduleItem.style';
import ScheduleItemContent from '../components/scheduleItemContent';

export const ScheduleItem = ({scheduleItemData}) => {
  return scheduleItemData ? (
    <ScheduleItemWrapper>
      <ScheduleItemContent scheduleItemData={scheduleItemData}/>
    </ScheduleItemWrapper>
  ) : null;
};

export default ScheduleItem;

```

Файл scheduleItemExtended.jsx

```

import { ScheduleItemExtendedUnit, ScheduleItemExtendedWrapper, CollapseItem } from './scheduleItemExtended.style';
import ScheduleItemContent from '../components/scheduleItemContent';
import { useEffect, useState } from 'react';

const ScheduleItemExtended = ({scheduleItemData}) => {
  const [collapsed, setCollapse] = useState(true);
  const [hasData, setHasData] = useState(true)

  useEffect(() => {
    const countEmpty = scheduleItemData.reduce((totalEmpty, currentItem) => {
      return totalEmpty + (currentItem.teacherName === "" && currentItem.place === "")
    }, 0)

    if(countEmpty === scheduleItemData.length){
      setHasData(false)
    }
  }, [scheduleItemData])

  const generateScheduleUnits = () => {
    return scheduleItemData.map((item, i) => {
      return (
        <ScheduleItemExtendedUnit key={i}>
          <ScheduleItemContent collapsed={collapsed} scheduleItemData={item}/>
        </ScheduleItemExtendedUnit>
      )
    })
  }

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

    );
  });
};

return scheduleItemData && scheduleItemData.length ? (
  <ScheduleItemExtendedWrapper items={scheduleItemData.length}>
    {generateScheduleUnits()}
    {hasData && <CollapseItem onClick={() => setCollapse(value => !value)}>{collapsed ? 'Більше інформації' : 'Менше
інформації'}</CollapseItem>}
  </ScheduleItemExtendedWrapper>
) : null;
};

export default ScheduleItemExtended;

```

Файл scheduleRow.jsx

```

import ScheduleItem from './scheduleItem';
import React from 'react';
import { EmptyElement } from './scheduleRow.style';
import ScheduleItemExtended from './scheduleItemExtended';

const ScheduleRow = ({dataset}) => {

  const generateScheduleItem = dataset => {
    return dataset.map((item, index) => {
      if (Array.isArray(item)) {
        return <ScheduleItemExtended key={index} scheduleItemData={item} />
      }

      return item ? <ScheduleItem key={index} scheduleItemData={item}/> : <EmptyElement key={index}/>;
    });
  };

  return (
    <
      {generateScheduleItem(dataset)}
    </>
  );
};

export default ScheduleRow;

```

Файл scheduleTypeTabs.jsx

```

import ScheduleTypeTab from './././components/scheduleTypeTab';
import { TabContainer } from './scheduleTypeTabs.style';
import { routes } from './././common/constants/routes';

const ScheduleTypeTabs = () => {
  const tabs = [
    {value: routes.GROUP, label: 'Розклад занять'},
    {value: routes.SESSION, label: 'Розклад сесії'},
    {value: routes.LECTURER, label: 'Розклад для викладачів'}
  ];
};

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

const TabComponents = tabs.map((tab) => {
  return <ScheduleTypeTab url={tab.value} key={tab.value}>{tab.label}</ScheduleTypeTab>;
});

return (
  <TabContainer justifyContent="space-between" alignItems="center" gap={'37px'}>
    {TabComponents}
  </TabContainer>
);
};

export default ScheduleTypeTabs;

```

Файл scheduleWrapper.jsx

```

import { useEffect, useState } from 'react';
import ScheduleRow from './scheduleRow';
import { GridWrapper } from './scheduleWrapper.style';
import SliceOptionsContext from '../common/context/sliceOptionsContext';
import ScheduleDayToggler from './scheduleDayToggler';
import Schedule from './schedule';
import { generateScheduleMatrix } from '../common/utills/generateScheduleMatrix';
import { useWeekContext } from '../common/context/weekContext';
import { useLecturerContext } from '../common/context/lecturerContext';
import { useGroupContext } from '../common/context/groupContext';

const ScheduleWrapper = ({getData, contextType}) => {
  const [sliceParams, setSliceParams] = useState(null);
  const [data, setData] = useState(null);

  const {currentWeek} = useWeekContext();
  const {lecturer} = useLecturerContext();
  const {group} = useGroupContext();

  const weekValue = {
    firstWeek: 'scheduleFirstWeek',
    secondWeek: 'scheduleSecondWeek'
  };

  useEffect(() => {
    const contextValue = contextType === 'lecturer' ? lecturer : group?.id;

    if (contextValue) {
      getData(contextValue)
        .then(res => setData(res.data));
    } else {
      setData(null);
    }
  }, [lecturer, group, contextType, getData]);

  const generateScheduleRows = (scheduleMatrix) => {
    return scheduleMatrix.map((item, i) => {

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```

const slicedDataset = sliceParams
  ? item.slice(sliceParams.start, sliceParams.end)
  : item;
return <ScheduleRow key={i} dataset={slicedDataset}/>;
});
};

const setSlice = (value) => {
  if (!value) {
    setSliceParams(null);
    return;
  }

  const ranges = value.split('-');
  setSliceParams({ start: +ranges[0], end: +ranges[1]});
};

return (
  data &&
  <div style={{overflow: 'hidden'}}>
    <GridWrapper>
      <SliceOptionsContext value={sliceParams}>
        <ScheduleDayToggle initialValue={'mon'} handler={setSlice}/>
        <Schedule>
          {generateScheduleRows(generateScheduleMatrix(data[weekValue[currentWeek]]))}
        </Schedule>
      </SliceOptionsContext>
    </GridWrapper>
  </div>
);
};

export default ScheduleWrapper;

```

Файл scrollToTop.jsx

```

import { useEffect, Fragment } from 'react';
import { withRouter } from 'react-router-dom';

// Scroll to top on location change
const ScrollToTop = ({ history, children }) => {
  useEffect(() => {
    const unlisten = history.listen(() => {
      window.scrollTo({
        top: 0,
        behavior: "smooth"
      });
    });
    return () => {
      unlisten();
    }
  }, [history]);

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```
    return <Fragment>{ children }</Fragment>;
  }
```

```
export default withRouter(ScrollToTop);
```

Файл tabs.jsx

```
import React, { useEffect, useState } from 'react';
```

```
const Tabs = ({ children, onTabChanged }) => {
  const [currentTab, setTab] = useState(0);

  useEffect(() => {
    if (typeof currentTab === 'number') {
      onTabChanged(currentTab);
    }
  }, [currentTab]);

  return (
    <>
      {React.Children.map(children, (child, index) => {
        return React.cloneElement(child, {
          tabClick: () => setTab(index),
          activeTab: currentTab === index,
        });
      })}
    </>
  );
};
```

```
export default Tabs;
```

Файл index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './app/App';
import ScopedCssBaseline from '@material-ui/core/ScopedCssBaseline';
```

```
import { BrowserRouter as Router } from 'react-router-dom';
import GlobalStyles from './index.style';
```

```
ReactDOM.render(
  <React.StrictMode>
    <ScopedCssBaseline>
      <GlobalStyles />
      <Router>
        <App />
      </Router>
    </ScopedCssBaseline>
  </React.StrictMode>,
  document.getElementById('root')
);
```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

Файл index.style.js

```
import { createGlobalStyle } from 'styled-components';
```

```
const GlobalStyles = createGlobalStyle`
```

```
* {
```

```
  font-family: 'Exo 2', sans-serif;
```

```
}
```

```
body {
```

```
  width: 100%;
```

```
  margin: 0;
```

```
  padding: 0;
```

```
}
```

```
h1, h2, h3, h4, h5, h6 {
```

```
  margin: 0;
```

```
}
```

```
`;
```

```
export default GlobalStyles;
```

Файл exam.style.js

```
import styled from "styled-components";
```

```
import { Flex } from "../../common/styles/styles";
```

```
import { getValueFromTheme } from "../../common/utils/getValueFromTheme";
```

```
import { ScheduleItemMixin } from '../../common/styles/styles';
```

```
export const ScheduleItemType = styled.div`
```

```
  width: 84px;
```

```
  height: 27px;
```

```
  line-height: 27px;
```

```
  text-align: center;
```

```
  background-color: #949dff;
```

```
  border-radius: 8px;
```

```
  color: ${getValueFromTheme("secondaryFontColor")};
```

```
  font-weight: 600;
```

```
`;
```

```
export const Subject = styled.div`
```

```
  font-weight: bold;
```

```
  font-size: 14px;
```

```
  line-height: 17px;
```

```
  color: ${getValueFromTheme("primaryFontColor")};
```

```
  margin-top: 13px;
```

```
`;
```

```
const Property = styled(Flex)`
```

```
  position: relative;
```

```
  font-weight: 600;
```

```
  font-size: 13px;
```

					КПІ.ІТ-9203.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		45

```
line-height: 16px;
color: ${getValueFromTheme("primaryFontColor")};
margin-top: 10px;
align-items: center;
`;
```

```
export const Teacher = styled(Property)`
  gap: 4px;
`;
```

```
export const Location = styled(Property)`
  gap: 7px;
`;
```

```
export const GroupName = styled(Property)`
  gap: 7px;
`;
```

```
export const ScheduleItemHeader = styled(Flex)`
  gap: 25px;
  align-items: center;
`;
```

```
export const ScheduleItemCurrent = styled.span`
  position: relative;
  font-weight: bold;
  color: #25cf9c;
```

```
&:before {
  content: "";
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
  left: -13px;
  display: block;
  background-color: #25cf9c;
  border-radius: 50%;
  width: 8px;
  height: 8px;
}
`;
```

```
export const CardWrapper = styled.div`
  ${ScheduleItemMixin};
  margin: 40px auto 0;
  padding: 16px 36px 16px 16px;
  display: flex;
  justify-content: space-between;
  max-width: 706px;
`;
```

```
export const CardMainData = styled.div`
  margin-left: 16px;
`;
```

Вмін.	Арк.	№ докум.	Підп.	Дата.

```
export const CardDate = styled.div`
  align-self: center;
  min-width: 150px;
  color: ${getValueFromTheme('primaryFontColor')};
`;
```

```
export const Divider = styled.div`
  border-right: 1px solid grey;
  margin-right: 2rem;
`;
```

```
export const DividerRed = styled.div`
  border-right: 5px solid red;
`;
```

Файл optionMultiplyToogler.style.js

```
import styled, { css } from 'styled-components';
import { getValueFromTheme } from '../common/utils/getValueFromTheme';
```

```
export const TogglerWrapper = styled.div`
  display: flex;
  padding: 4px;
  justify-content: space-around;
  min-height: 48px;
  background: ${getValueFromTheme('bgOptions')};
  box-shadow: inset 0 0 8px rgba(136, 136, 136, 0.08);
  border-radius: 42px;
  position: relative;
  z-index: 1;
`;
```

```
export const TogglerOption = styled.div`
  transition: .4s;
  display: flex;
  align-items: center;
  justify-content: center;
  width: 100%;
  cursor: pointer;
  font-weight: 500;
  font-size: 14px;
  letter-spacing: 0.01em;
  color: ${getValueFromTheme('primaryFontColor')};
  border-radius: 90px;
  ${props => props.active ? css`
    font-weight: 600;
    background-color: #fff;
    border: 0.5px solid rgba(20, 21, 24, 0.12);
    box-shadow: 0 0 2px rgba(0, 0, 0, 0.12);
    color: #141518;
  `}
`;
```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

Файл scheduleItemContent.style.js

```
import styled from 'styled-components';
import { Flex } from '../common/styles/styles';
import { getValueFromTheme } from '../common/utils/getValueFromTheme';

export const ScheduleItemType = styled.div`
  width: 84px;
  height: 27px;
  line-height: 27px;
  text-align: center;
  background-color: #949dff;
  border-radius: 8px;
  color: ${getValueFromTheme('secondaryFontColor')};
  font-weight: 600;
`;

export const ScheduleItemTypeLab = styled(ScheduleItemType)`
  background-color: #F6B382;
`;

export const ScheduleItemTypePrac = styled(ScheduleItemType)`
  background-color: #FF7F96;
`;

export const ScheduleItemTypeLec = styled(ScheduleItemType)`
  background-color: #949DFF;
`;

export const Subject = styled.div`
  font-weight: bold;
  font-size: 14px;
  line-height: 17px;
  color: ${getValueFromTheme('primaryFontColor')};
  margin-top: 13px;
`;

const Property = styled(Flex)`
  position: relative;
  font-weight: 600;
  font-size: 13px;
  line-height: 16px;
  color: ${getValueFromTheme('primaryFontColor')};
  margin-top: 10px;
  align-items: center;
`;

export const Teacher = styled(Property)`
  gap: 4px;
`;

export const Location = styled(Property)`
  gap: 7px;
```

Вмін.	Арк.	№ докум.	Підп.	Дата.

```

`;

export const LocationLink = styled.a`
  color: ${getValueFromTheme('primaryFontColor')};
  text-decoration: none;
`;

export const GroupName = styled(Property)`
  gap: 7px
`;

export const ScheduleItemHeader = styled(Flex)`
  gap: 25px;
  align-items: center;
`;

export const ScheduleItemCurrent = styled.span`
  position: relative;
  font-weight: bold;
  color: #25CF9C;

  &:before {
    content: "";
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    left: -13px;
    display: block;
    background-color: #25CF9C;
    border-radius: 50%;
    width: 8px;
    height: 8px;
  }
`;

```

Файл scheduleTypeTab.style.js

```

import styled from 'styled-components';
import { getValueFromTheme } from '../common/utils/getValueFromTheme';

export const Tab = styled.div`
  position: relative;
  font-weight: bold;
  font-size: 18px;
  letter-spacing: 0.01em;
  color: ${getValueFromTheme('primaryFontColor')};
  cursor: pointer;

  &:after {
    content: "";
    top: -12px;
    border-radius: 6px;
    height: 2px;
    background-color: ${getValueFromTheme('primaryFontColor')};
    display: ${(props) => (props.active ? 'block' : 'none')};
  }

```

Змін.	Арк.	№ докум.	Підп.	Дата.

```
}  
,
```

Файл switch.scss

```
@mixin switchPictogram($content_url) {  
  content: url($content_url);  
  position: absolute;  
  width: 28px;  
  height: 28px;  
  top: 50%;  
  transform: translateY(-50%);  
  transition: .4s;  
}  
  
.switch {  
  position: relative;  
  display: inline-block;  
  width: 60px;  
  height: 34px;  
}  
  
.switch input {  
  opacity: 0;  
  width: 0;  
  height: 0;  
}  
  
.slider {  
  position: absolute;  
  cursor: pointer;  
  top: 0;  
  left: 0;  
  right: 0;  
  bottom: 0;  
  transition: .4s;  
}  
  
.checker {  
  &:before {  
    @include switchPictogram("../assets/icons/moon.svg");  
    left: -45px;  
  }  
  
  &:after {  
    @include switchPictogram("../assets/icons/sun.svg");  
    right: -45px;  
  }  
}  
  
.slider:before {  
  position: absolute;  
  content: "";
```

Змін.	Арк.	№ докум.	Підп.	Дата.

КПІ.ІТ-9203.045440.03.12

Арк.

50

```

height: 26px;
width: 26px;
left: 4px;
bottom: 4px;
background-color: white;
transition: .4s;
}

input:checked+.slider:before {
  transform: translateX(26px);
}

input:not(:checked)~.checker:after {
  opacity: 0;
}

input:checked~.checker:before {
  opacity: 0;
}

/* Rounded sliders */
.slider.round {
  border-radius: 34px;
}

.slider.round:before {
  border-radius: 50%;
}

.hidden {
  display: none;
}

```

Файл timeDivider.style.js

```

import styled from 'styled-components';
import { Flex, media } from '../common/styles/styles';
import { getValueFromTheme } from '../common/utils/getValueFromTheme';

export const Container = styled.div`
  position: relative;
  align-self: start;
  grid-column: 1 / -1;
`;

export const InnerContainer = styled(Flex)`
  left: -90px;
  right: 0;
  top: -17px;
  position: absolute;

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

КПІ.ІТ-9203.045440.03.12

Арк.

51

```

    ${media.extraSmallMode} {
      left: -44px;
    }
  `;

export const TimeCell = styled.div`
  width: 64px;
  height: 36px;
  background: ${getValueFromTheme('primaryFontColor')};
  border-radius: 42px;
  color: ${getValueFromTheme('secondaryFontColor')};
  font-weight: 600;
  display: flex;
  justify-content: center;
  align-items: center;

  ${media.extraSmallMode} {
    background: none;
    color: ${getValueFromTheme('primaryFontColor')}
  }
  `;

export const Divider = styled.div`
  border-top: 1px dashed ${getValueFromTheme('primaryFontColor')};
  width: 100%;
  `;

```

Файл navbar.style.js

```

import styled from 'styled-components';
import { getValueFromTheme } from '../common/utils/getValueFromTheme';
import { Flex, media } from '../common/styles/styles';

export const NavBarWrapper = styled.header`
  padding: 16px 70px 48px;
  background: ${getValueFromTheme('bgHeader')};

  ${media.smallMode} {
    padding: 16px 5px 48px;
  }
  `;

export const NavBarInner = styled(Flex)`
  ${media.smallMode} {
    flex-direction: column;
    align-items: center;
  }
  `;

```

Файл schedule.style.js

```

import styled from 'styled-components';
import { media } from '../common/styles/styles';
import { getValueFromTheme } from '../common/utils/getValueFromTheme';

```

Змін.	Арк.	№ докум.	Підп.	Дата.

```

export const GridContainer = styled.div`
  padding-left: 100px;
  display: grid;
  position: relative;
  grid-template-columns: repeat(6, 1fr);
  grid-column-gap: 1.5rem;
  grid-row-gap: 10px;

  ${media.mediumMode} {
    grid-template-columns: repeat(3, 1fr);
  }
;

  ${media.smallMode} {
    grid-template-columns: repeat(2, 1fr);
  }

  ${media.extraSmallMode} {
    grid-template-columns: repeat(1, 1fr);
    padding-left: 23px;
  }
`;

export const CurrentDayContainer = styled.div`
  transform: scale(1.1, 1.2);
  position: absolute;
  background: ${getValueFromTheme('currentDayContainer')};
  grid-column: ${props => props.start} / span 1;
  width: 100%;
  top: 0;
  bottom: 0;
  z-index: 0;
;

```

Файл scheduleWrapper.style.js

```

import styled from 'styled-components';
import { getValueFromTheme } from '../common/utils/getValueFromTheme';

export const GridWrapper = styled.div`
  overflow: hidden;
  background-color: ${getValueFromTheme('bgTable')};
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
  border-radius: 40px;
  padding: 48px 20px 7rem;
  margin: 36px 24px 64px;
;

export const Header = styled.h1`
  /* padding: 20px; */
  color: ${getValueFromTheme('primaryFontColor')};
  text-align : center;

```

Вмін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

```
font-weight: lighter;
```

```
export const WordAccent = styled.span`
```

```
font-weight: bold;
```

Файл `scheduleItemExtended.style.js`

```
import styled, { css } from 'styled-components';
```

```
import { ScheduleItemMixin } from '../common/styles/styles';
```

```
import { getValueFromTheme } from '../common/utils/getValueFromTheme';
```

```
export const ScheduleItemExtendedWrapper = styled.div`
```

```
grid-row-start: span ${props => props.items};
```

```
z-index: 9;
```

```
`;
```

```
const cutOffPart = css`
```

```
content: ";
```

```
position: absolute;
```

```
width: 16px;
```

```
height: 16px;
```

```
border-radius: 50%;
```

```
top: 0;
```

```
`;
```

```
export const ScheduleItemExtendedUnit = styled.div`
```

```
`${ScheduleItemMixin};
```

```
padding: 20px 25px 16px;
```

```
position: relative;
```

```
&:not(:first-child):not(:last-child) {
```

```
border-radius: 0;
```

```
}
```

```
&:last-child {
```

```
border-top-left-radius: 0;
```

```
border-top-right-radius: 0;
```

```
}
```

```
&:first-child {
```

```
border-bottom-left-radius: 0;
```

```
border-bottom-right-radius: 0;
```

```
}
```

```
&:not(:first-child) {
```

```
border-top: 0.5px dashed #141518;
```

```
&:before {
```

```
  ${cutOffPart};
```

```
left: 0;
```

```
transform: translate(-50%, -50%);
```

```
}
```

Змін.	Арк.	№ докум.	Підп.	Дата.

```
&:after {
  ${cutOffPart};
  right: 0;
  transform: translate(50%, -50%);
}
}
`;

export const CollapseItem = styled.div`
  background-color: ${getValueFromTheme('bgOptions')};
  color: ${getValueFromTheme('primaryFontColor')};
  border-radius: 0 0 15px 15px;
  text-align: center;
  padding: 15px;
  cursor: pointer;
`
```

					КПІ.ІТ-9203.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		55

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ” _____ 2023 р.

ІМПЛЕМЕНТАЦІЯ API ДЛЯ ПЛАТФОРМИ SCHEDULE KPI

Програма та методика тестування

КПІ.ІТ-9203.045440.04.51

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

Нормоконтроль:

_____ Ірина ВІТКОВСЬКА

Виконавець:

_____ Єлизавета БОЙКО

Київ – 2023

ЗМІСТ

1	ОБ'ЄКТ ВИПРОБУВАНЬ.....	3
2	МЕТА ТЕСТУВАННЯ	4
3	МЕТОДИ ТЕСТУВАННЯ.....	5
4	ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ	6

					КПІ.ІТ-9203.045440.04.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		2

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Об'єктом випробування є сервіс для комп'ютерів та мобільних пристроїв для перегляду розкладу занять та сесії здобувачів освітньої послуги та викладачів.

					КПІ.ІТ-9203.045440.04.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		3

2 МЕТА ТЕСТУВАННЯ

Метою тестування є наступне:

- перевірка правильності роботи програмного забезпечення відповідно до функціональних вимог;
- перевірка сумісності веб-додатку з останніми версіями сучасних браузерів (Chrome, Opera, Safari);
- перевірка сумісності застосунку з різними операційними системами (Windows, Linux, MacOS);
- перевірка надання даних;
- знаходження проблем, помилок і недоліків з метою їх усунення;
- перевірка зручності графічного інтерфейсу.

					КПІ.ІТ-9203.045440.04.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		4

3 МЕТОДИ ТЕСТУВАННЯ

Для тестування програмного забезпечення використовуються такі методи:

- статичне тестування – перевіряється програма разом з усією документацією, яка аналізується на предмет дотримання стандартів програмування;
- функціональне тестування – полягає у перевірці відповідності реальної поведінки програмного забезпечення очікуваній;
- системне тестування – перевіряється усе програмне забезпечення в цілому;
- мануальне тестування – тестування без використання автоматизації, тест-кейси пише особа, що тестує програмне забезпечення.

					КПІ.ІТ-9203.045440.04.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		5

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується мануально з метою знаходження помилок та недоліків як у функціональній частині програмного забезпечення так і в зручності користування. Для того, щоб перевірити працездатність та відмовостійкість застосунку, необхідно провести наступні тестування:

- динамічне тестування на відповідність функціональним вимогам;
- тестування на мобільних пристроях з різною роздільною здатністю екрану;
- тестування у різних браузерях;
- тестування зміни орієнтації екрану;
- тестування інтерфейсу користувача;
- тестування зручності використання.

					КПІ.ІТ-9203.045440.04.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		6

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ” _____ 2023 р.

ІМПЛЕМЕНТАЦІЯ АРІ ДЛЯ ПЛАТФОРМИ SCHEDULE KPI

Керівництво користувача

КПІ.ІТ-9203.045440.05.34

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

Нормоконтроль:

_____ Ірина ВІТКОВСЬКА

Виконавець:

_____ Єлизавета БОЙКО

Київ – 2023

ЗМІСТ

1	ПРИЗНАЧЕННЯ ПРОГРАМИ.....	3
2	ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	4
2.1	Системні вимоги для коректної роботи	4
2.2	Завантаження застосунку	4
2.3	Перевірка коректної роботи	4
3	ВИКОНАННЯ ПРОГРАМИ.....	5

					КПІ.ІТ-9203.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

1 ПРИЗНАЧЕННЯ ПРОГРАМИ

Розроблений сервіс – це веб-платформа для комп’ютерів та мобільних пристроїв для перегляду розкладу занять та сесії здобувачів освітньої послуги та викладачів. Кінцевий результат представляє собою повністю готову для використання платформу, яка може надавати користувачу план занять та сесії актуальних груп та дійсних викладачів КПІ ім. Ігоря Сікорського, що є інтегрованою з єдиним середовищем університету та може слугувати заміною минулої імплементації платформи розкладу занять.

					КПІ.ІТ-9203.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

2 ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ

2.1 Системні вимоги для коректної роботи

Для успішної роботи даного застосунку необхідне виконання наступних вимог:

- наявність мобільного пристрою або комп'ютеру;
- наявність доступу до Інтернету;
- наявність браузеру.

2.2 Завантаження застосунку

На даний момент можна скористатися сервісом, перейшовши на адресу веб-сторінки <https://schedule.kpi.ua>. Для цього спершу необхідно відкрити браузер та ввести адресу в пошуковий рядок.

2.3 Перевірка коректної роботи

При переході на платформу має відобразитись початкова сторінка з опціями перегляду розкладу: «Розклад занять», «Розклад сесії» та «Розклад викладачів». Первинно опція «Розклад занять» має бути активною. У разі, відображення виключення, скоріш за все відбулась помилка з'єднання з інтернетом.

					КПІ.ІТ-9203.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

3 ВИКОНАННЯ ПРОГРАМИ

При переході на головну сторінку сервісу має відобразитися інтерфейс з наданням опцій для перегляду розкладу занять та сесії групи, а також розкладу занять викладачів. Первинно опція перегляду розкладу занять групи є активною (рис. 3.1).

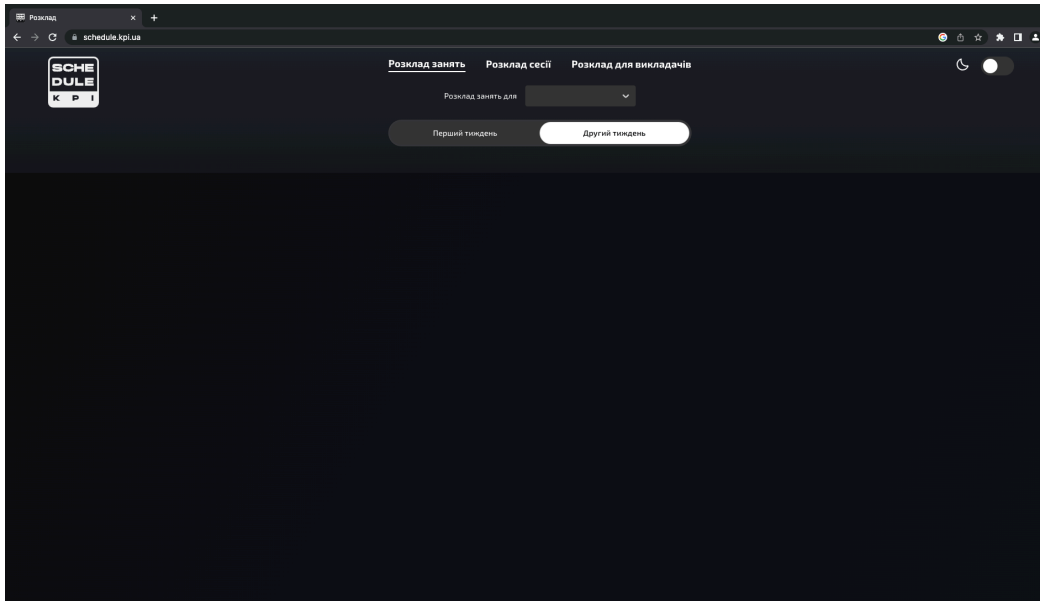


Рисунок 3.1 – Головна сторінка сервісу

На початковій сторінці сервісу відображено опції перекладу різних варіантів розкладів: «Розклад занять», що є опцією пошуку розкладу занять групи здобувачів освітньої послуги, «Розклад сесії», що є опцією пошуку розкладу сесії здобувачів освітньої послуги та «Розклад для викладачів», що є опцією пошуку розкладу занять викладачів університету.

Після натискання на поле розміщене після «Розклад занять для» буде відображено список актуальних груп університету для яких можна переглянути план занять (рис. 3.2).

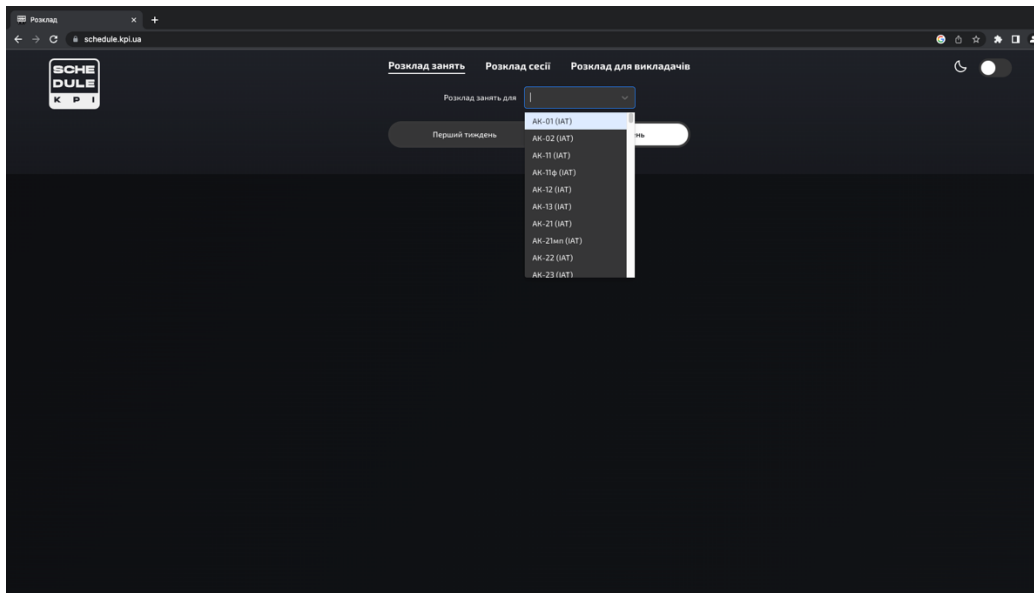


Рисунок 3.2 – Список актуальних груп університету

При виборі будь-якої групи зі списку та натисканні на її назву відобразиться розклад занять. Для прикладу взято групу ІТ-92 (рис. 3.3).

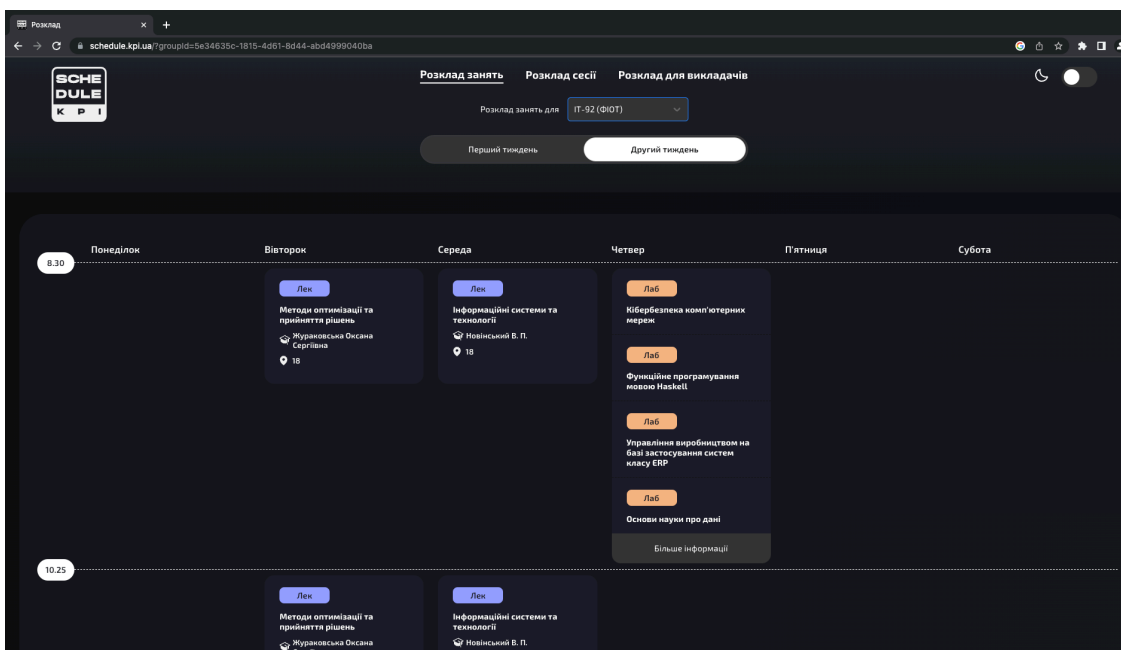


Рисунок 3.3 – Приклад розкладу занять групи

При натисканні на опції «Перший тиждень» або «Другий тиждень» можна переглянути розклад занять за обидва навчальні тижні (рис. 3.4).

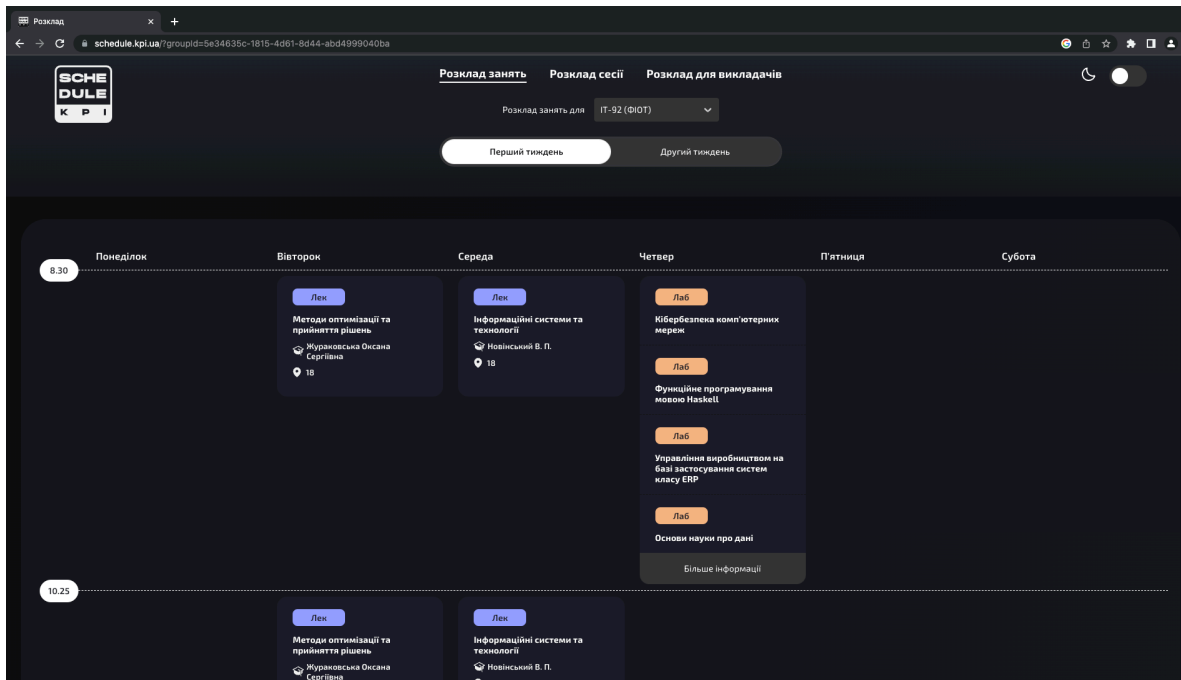


Рисунок 3.4 – Приклад розкладу занять за інший тиждень

Після натискання на опцію «Більше інформації», що знаходиться під переліком занять на день відобразиться повна інформація про заняття (рис. 4.5).

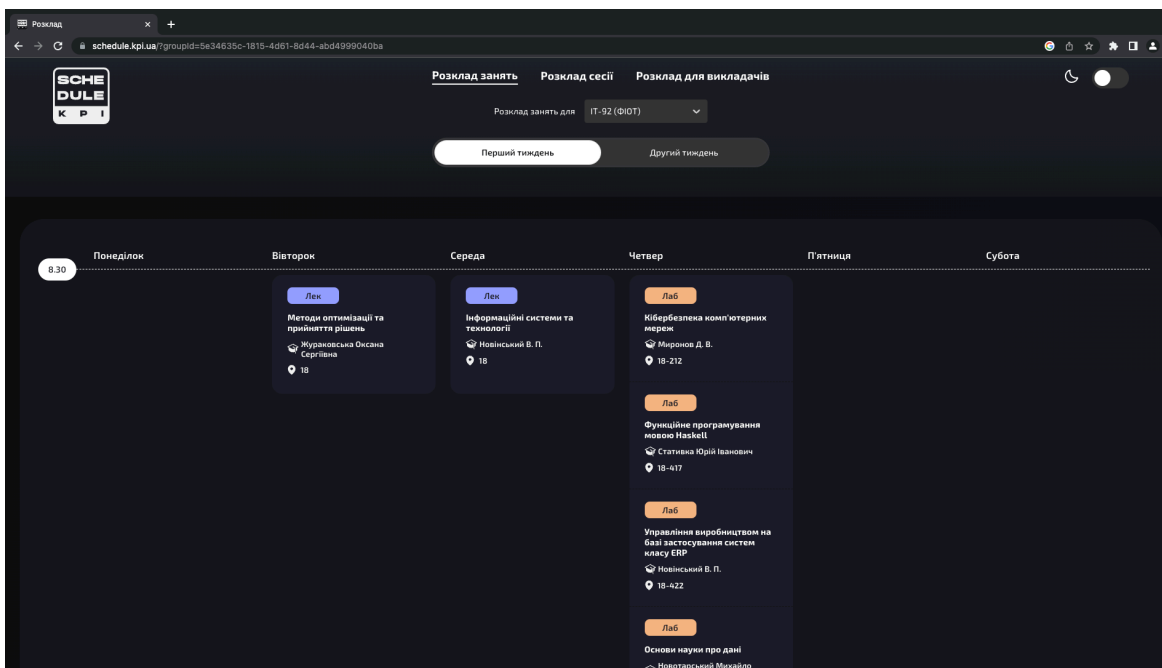


Рисунок 3.5 – Приклад повної інформації розкладу занять групи

При натисканні на опцію «Розклад сесії» відкриється сторінка розкладу сесії обраної групи (рис. 3.6).

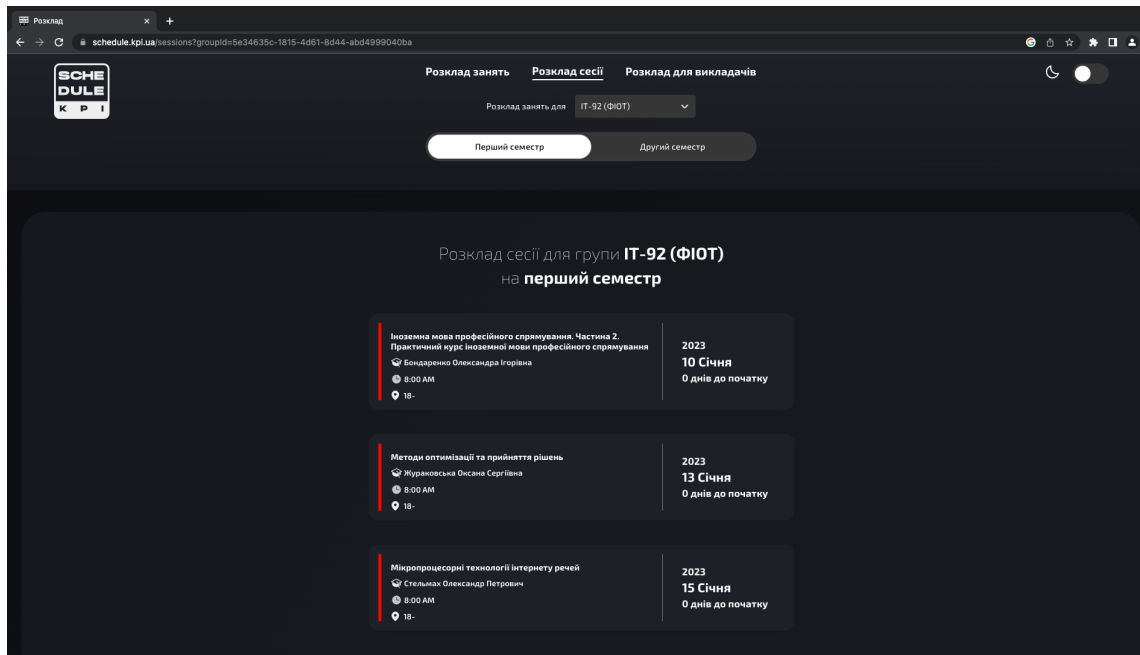


Рисунок 3.6 – Приклад розкладу сесії групи

При вводі назви групи в пошук, що знаходиться біля поля «Розклад занять для» відобразиться список в якому поступово будуть відображатися знайдені групи, що відповідають введеній назві. Такий самий пошук можна застосовувати і для пошуку розкладу занять. Для прикладу взято групу ІС-93 (рис. 3.7).

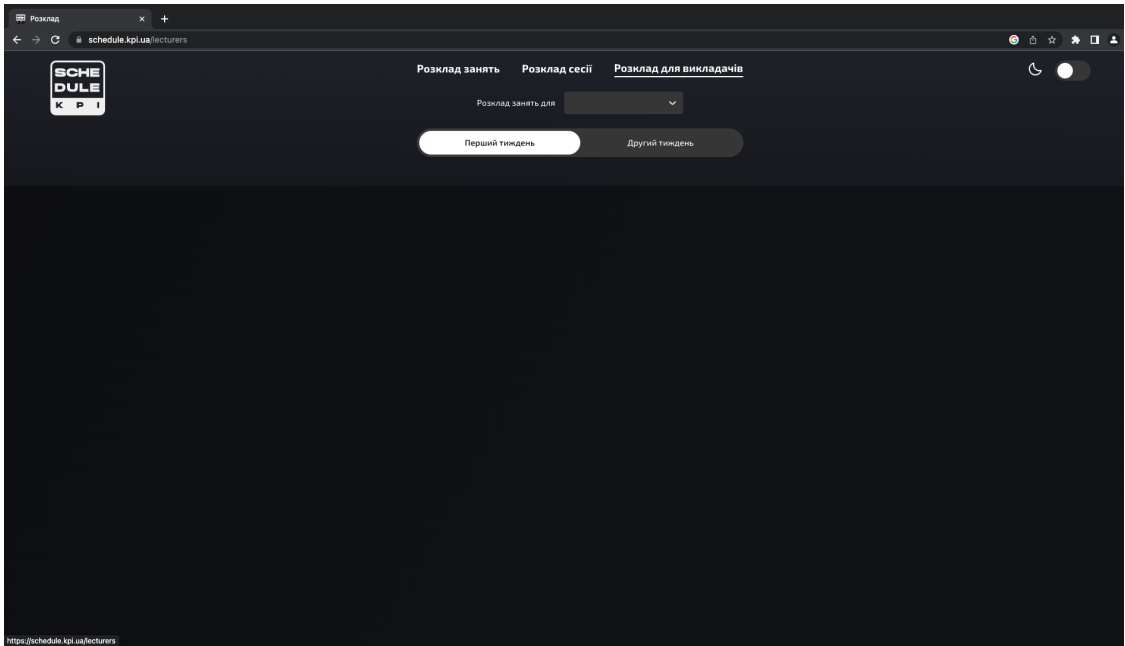


Рисунок 3.9 – Сторінка для пошуку розкладу викладача

При натисканні на поле біля «Розклад занять для» відобразиться список актуальних викладачів університету, розклад яких можна передивитись (рис. 3.10).

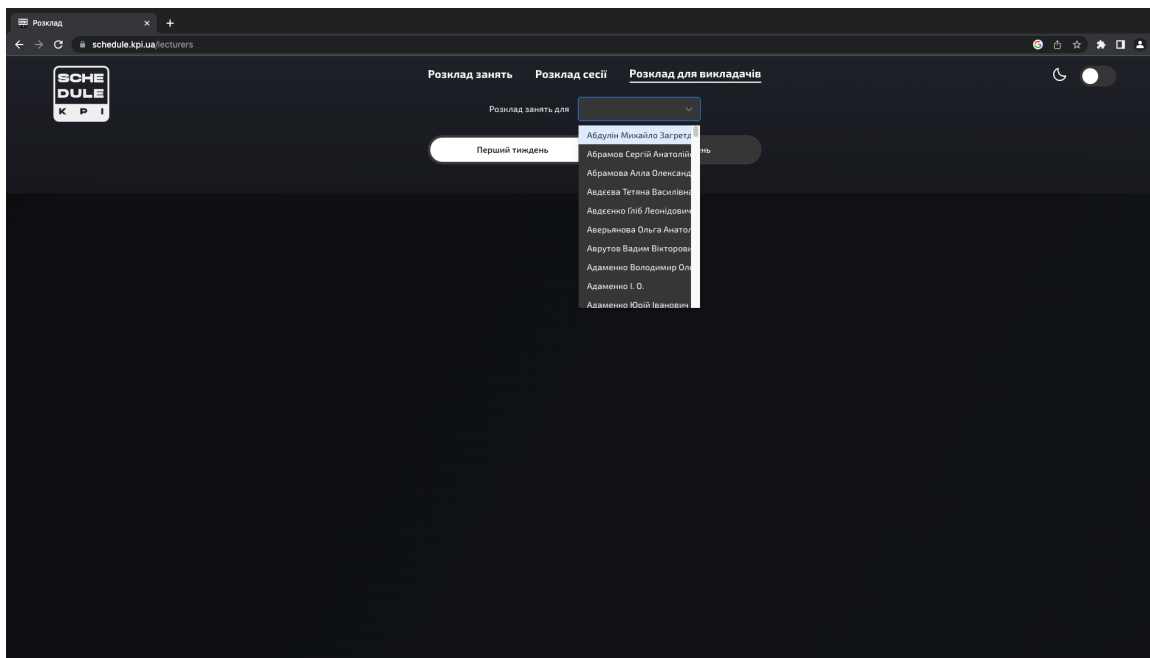


Рисунок 3.10 – Список актуальних викладачів університету

При натисканні на одного з викладачів зі списку, відобразиться розклад його занять (рис. 3.11).

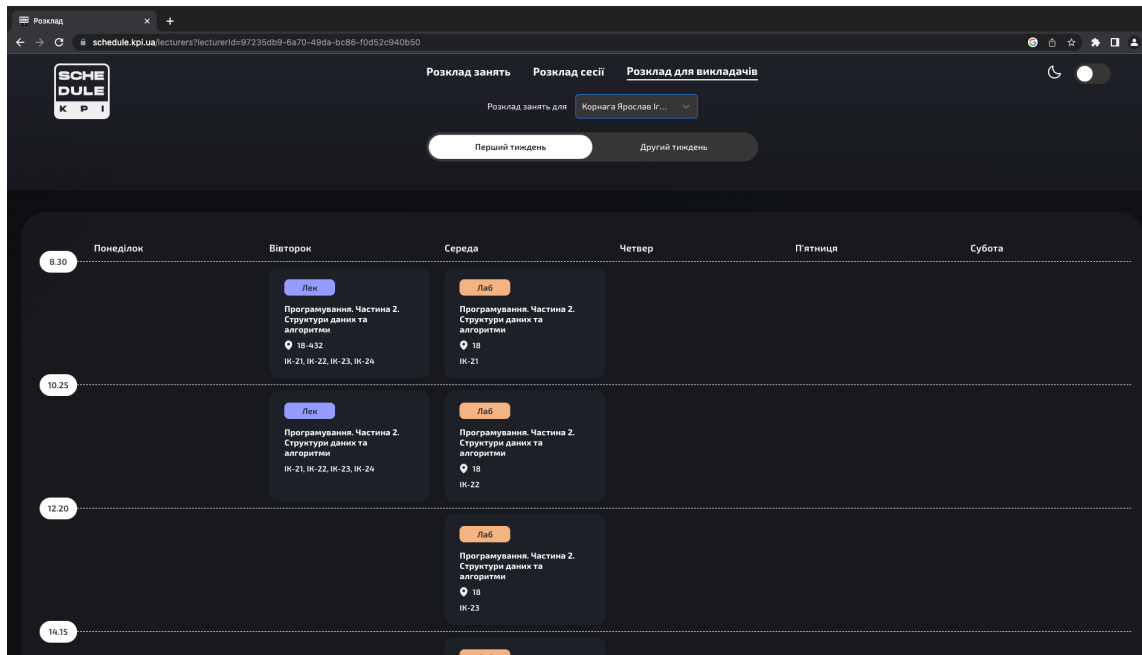


Рисунок 3.11 – Приклад розкладу викладача

При натисканні на поле після поля «Розклад занять для» можна скористатися пошуком. Під час введення назви, відобразиться список в якому поступово будуть відображатися знайдені викладачі, що відповідають введеному ПІБ. Для прикладу візьмемо – Лісовиченко О.І.(рис. 3.12 - 3.13).

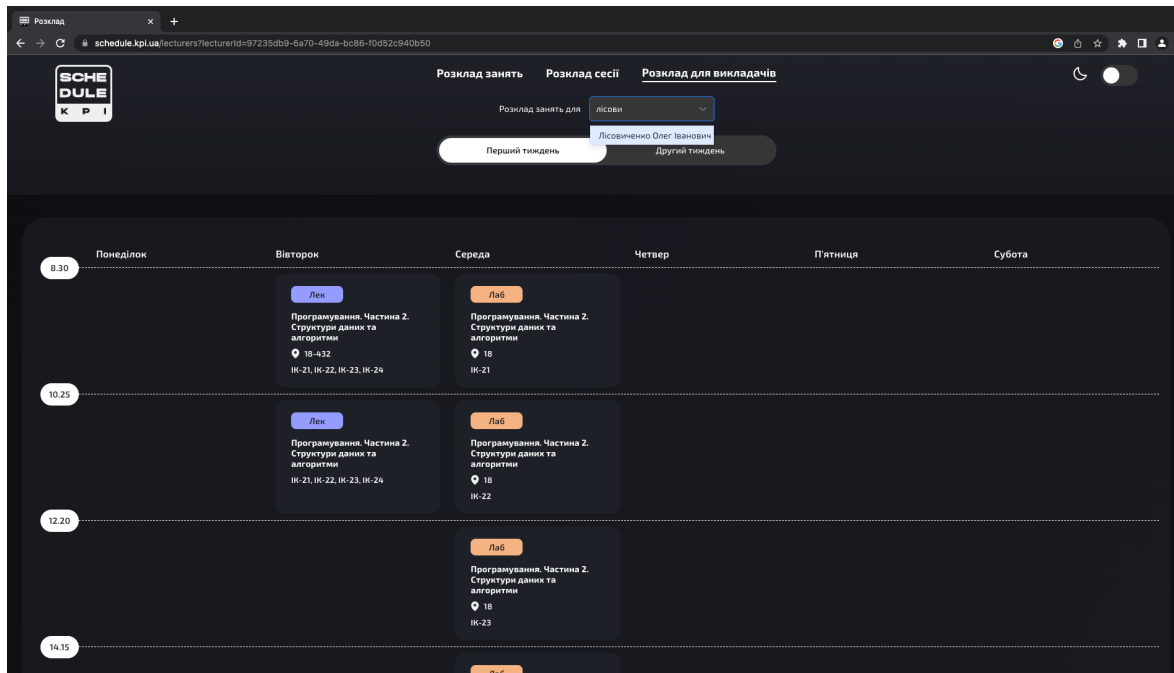


Рисунок 3.12 – Приклад пошуку викладача

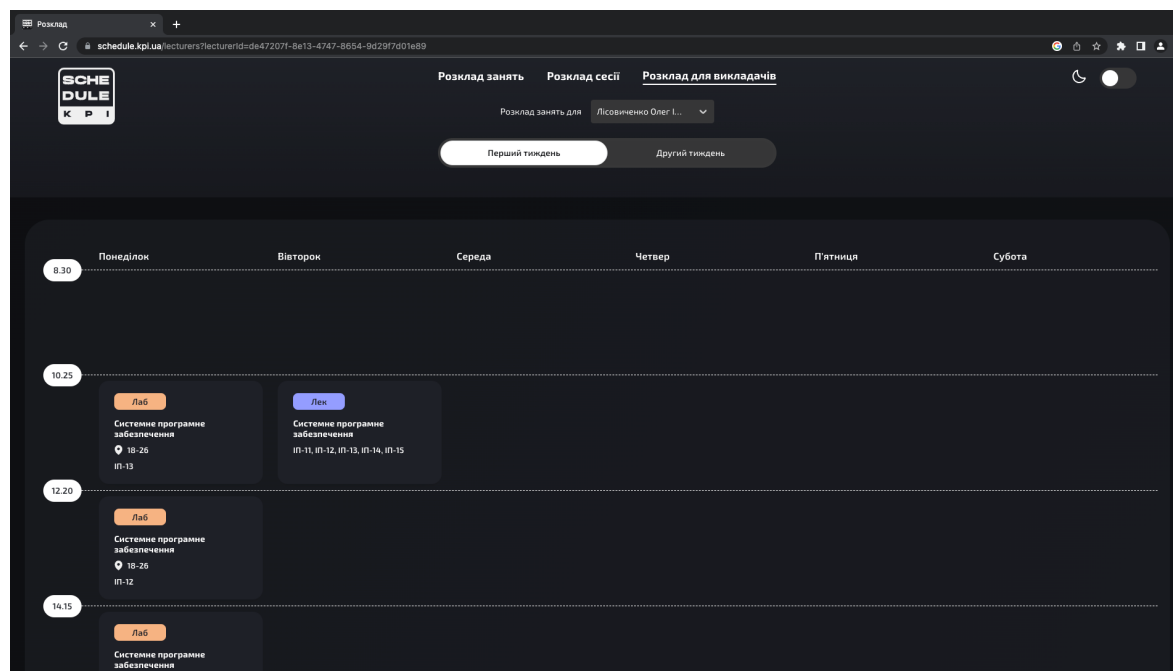


Рисунок 3.13 – Приклад розкладу шуканого викладача

При натисканні на опції «Перший тиждень» або «Другий тиждень» можна переглянути розклад занять за обидва навчальні тижні (рис. 3.14).

При натисканні на ПІБ викладача, що вказано в інформації предмету з розкладу, відкриється вікно розкладу цього викладача. Для прикладу натиснемо на «Сирота Олена Петрівна» з предмету «Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення», що викладається першою парою в четвер (рис. 3. 16).

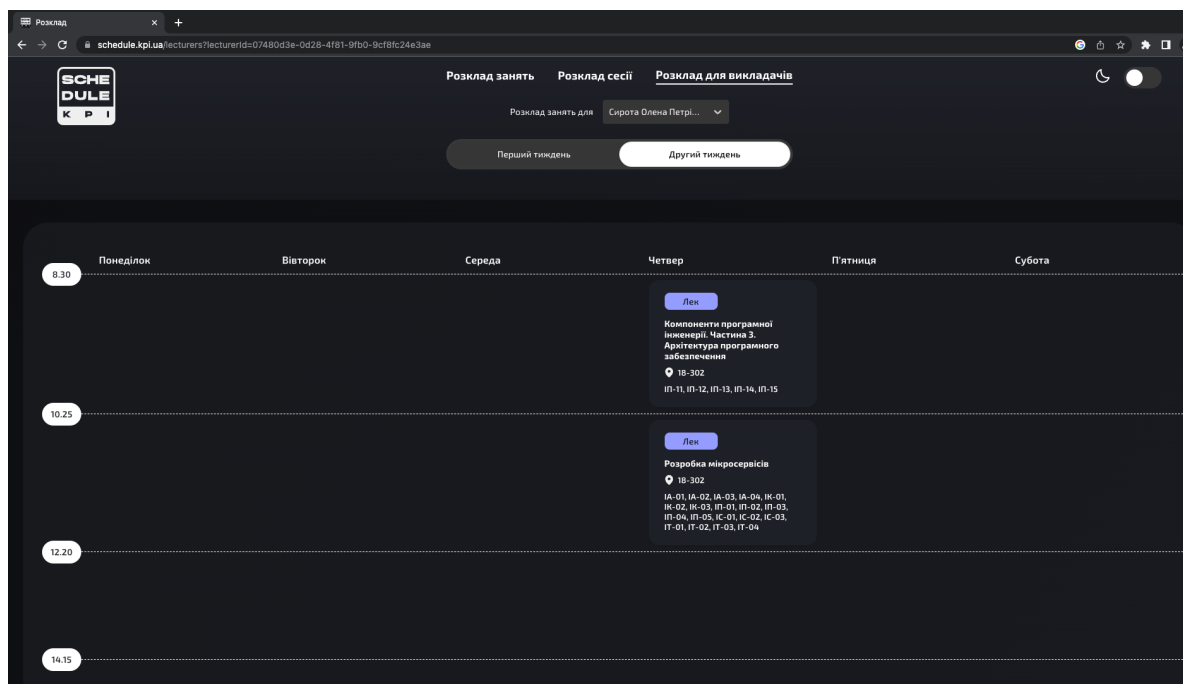


Рисунок 3.16 – Приклад розкладу викладача

При натисканні на перемикач в правому куту екрану, тема додатку зміниться на світлу, якщо поточна є темною або на темну, якщо поточна є світлою (рис. 3.17).

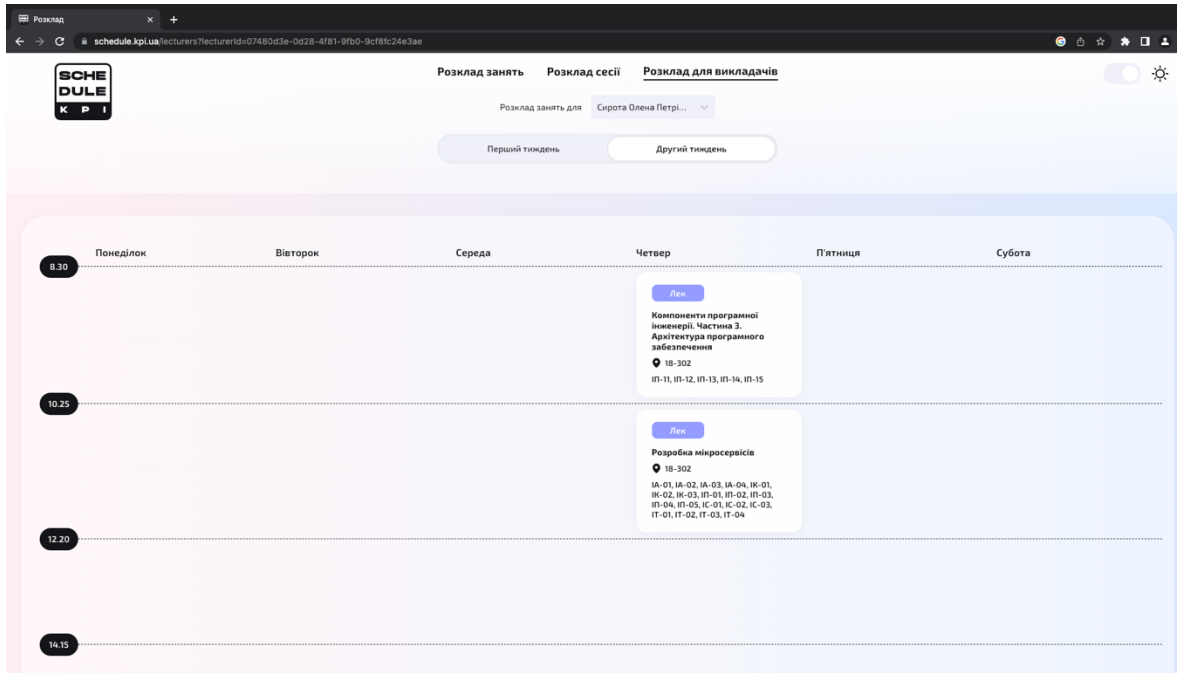


Рисунок 3.17 – Приклад світлої теми сервісу

Змін.	Арк.	№ докум.	Підп.	Дата.

КПІ.ІТ-9203.045440.05.34

Арк.

15

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

ІМПЛЕМЕНТАЦІЯ API ДЛЯ ПЛАТФОРМИ SCHEDULE KPI

Графічний матеріал

КПІ.ІТ-9203.045440.06.99

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

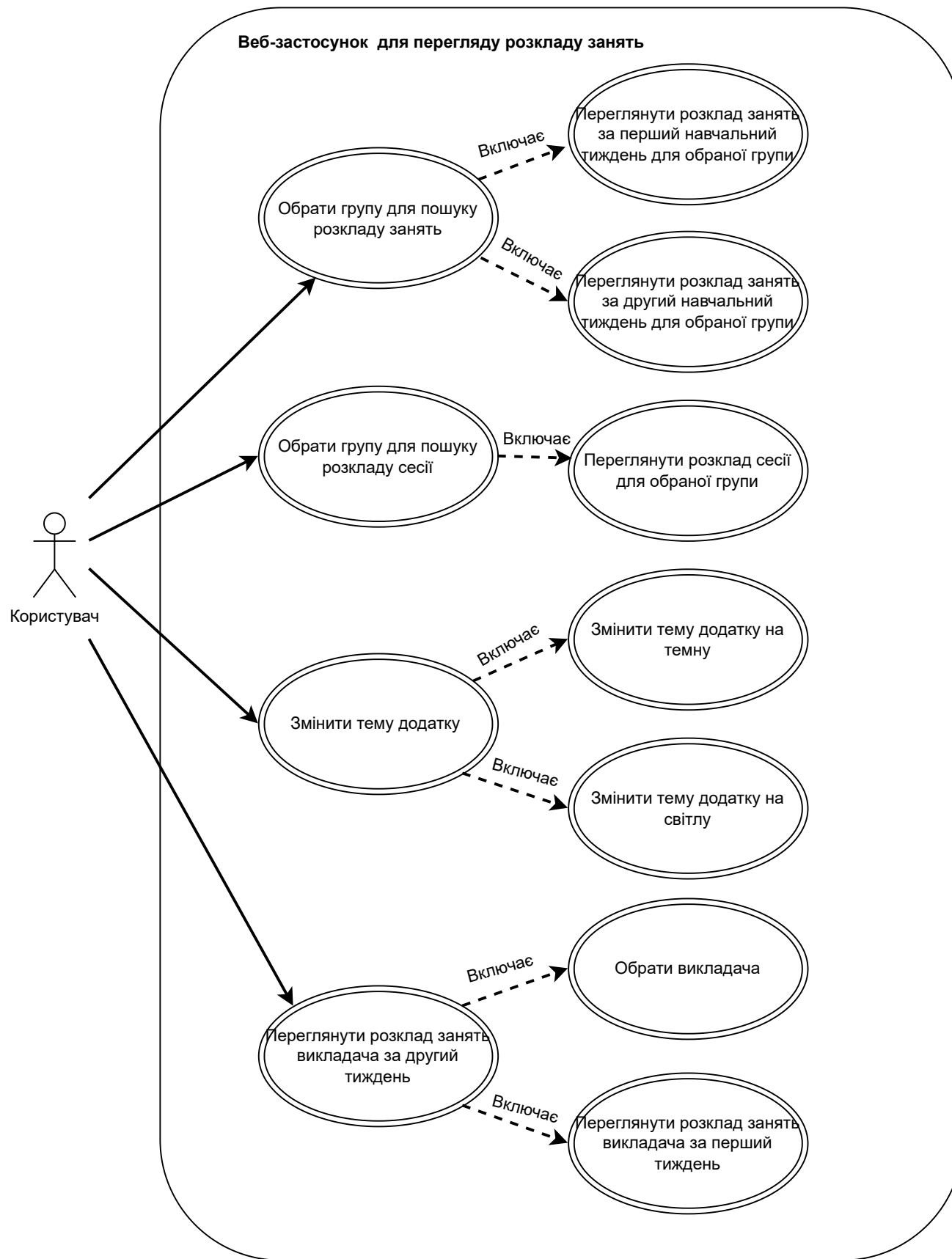
Нормоконтроль:

_____ Ірина ВІТКОВСЬКА

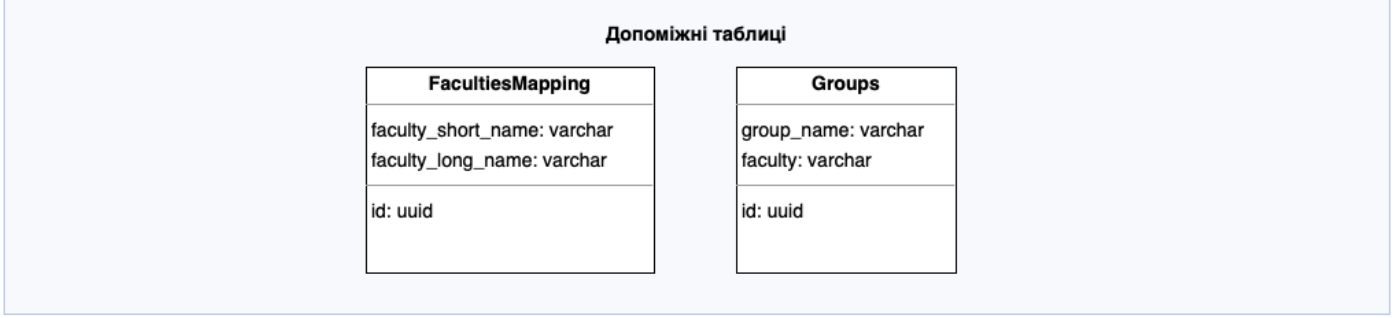
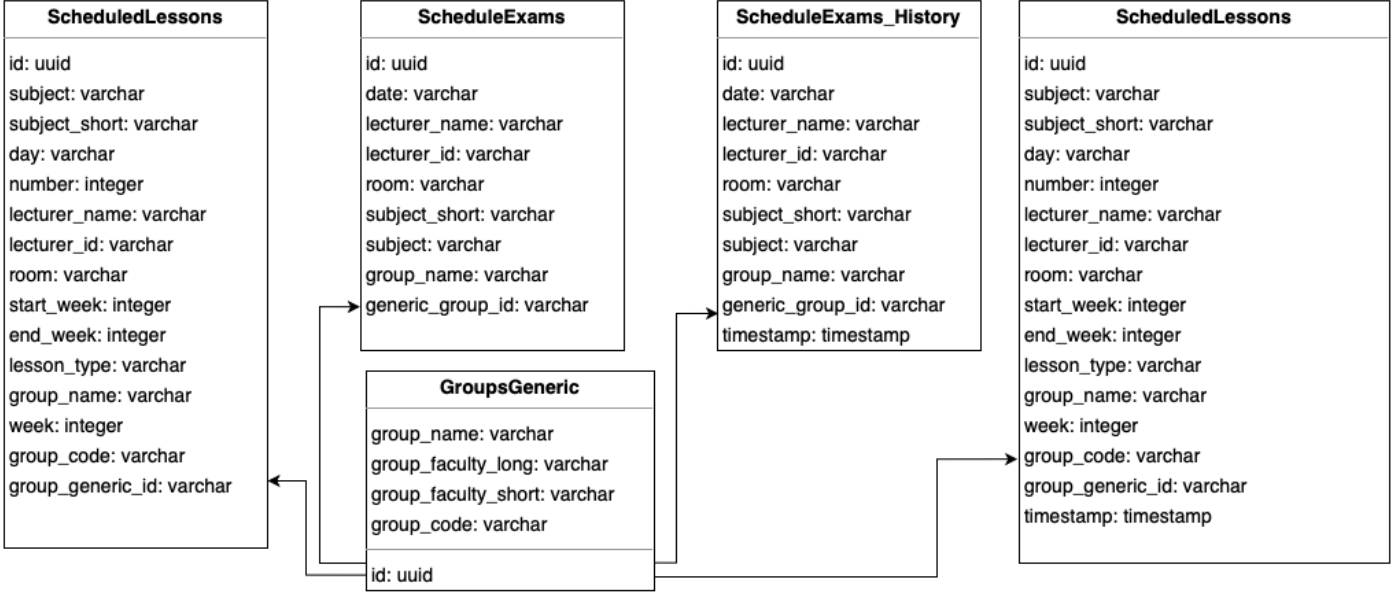
Виконавець:

_____ Єлизавета БОЙКО

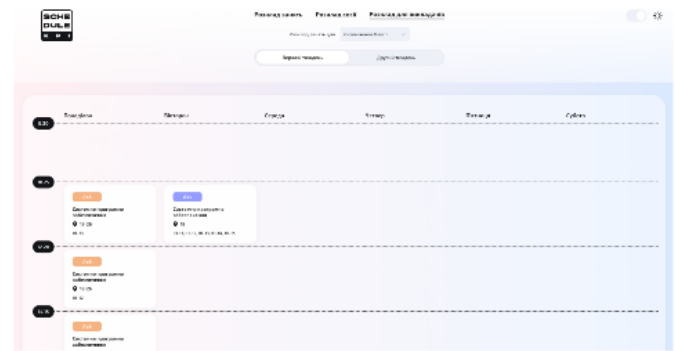
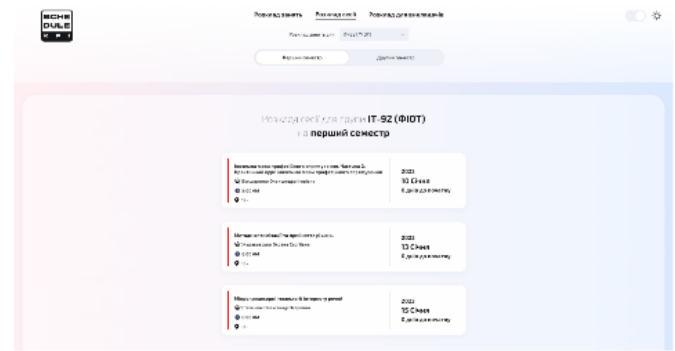
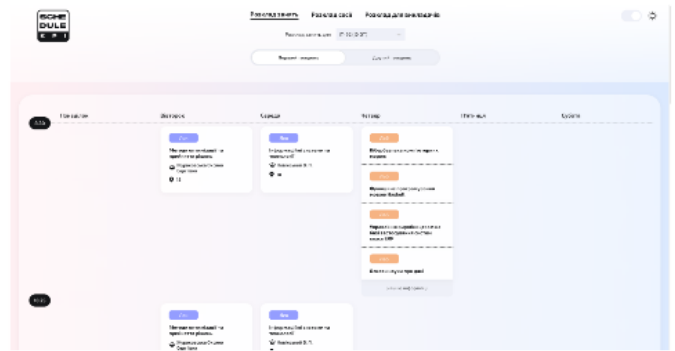
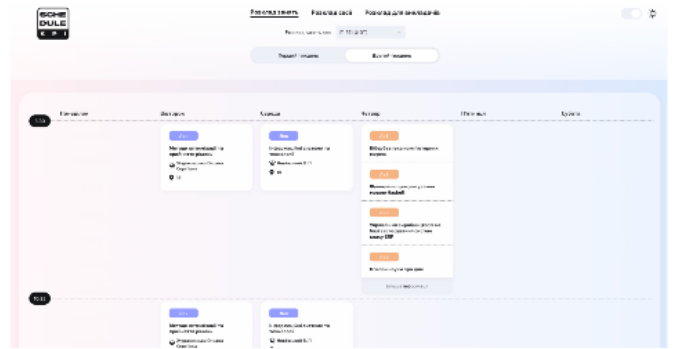
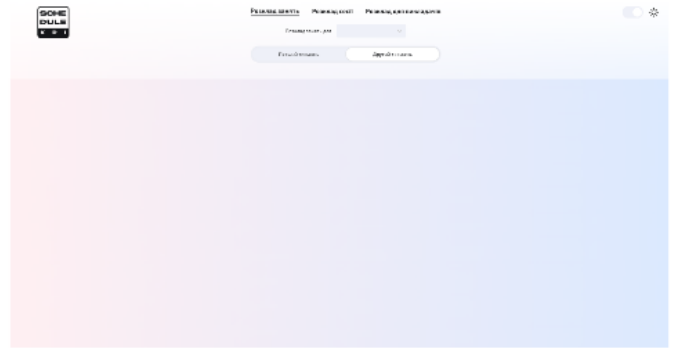
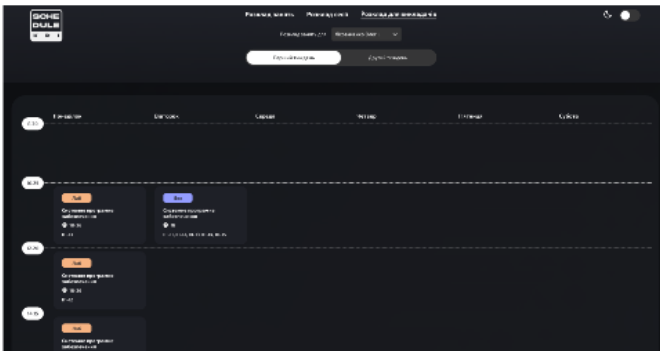
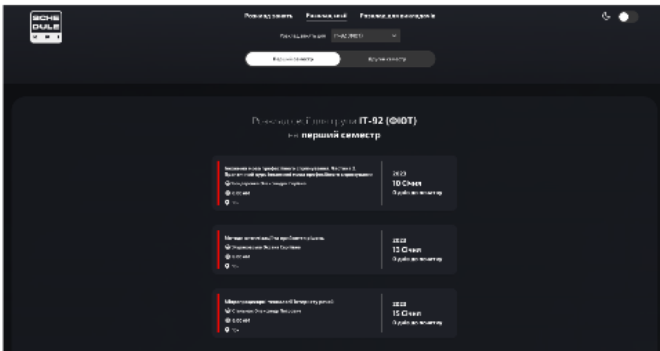
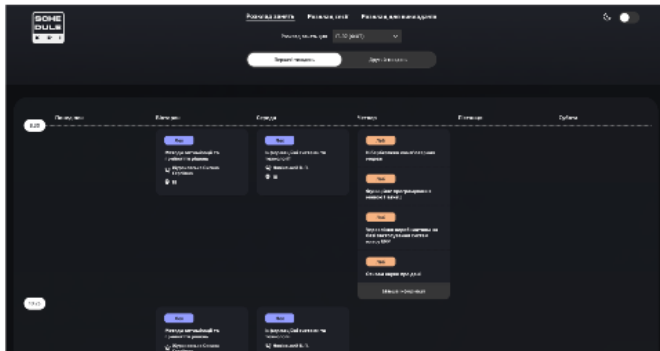
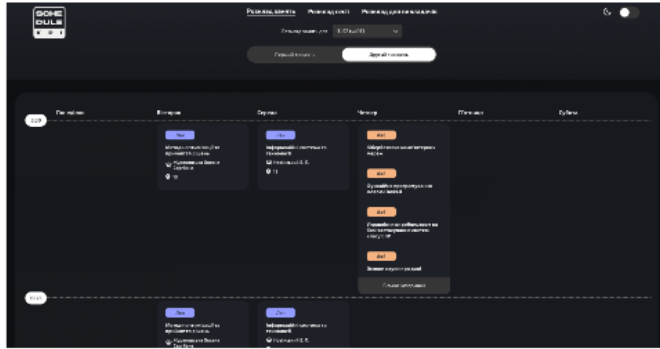
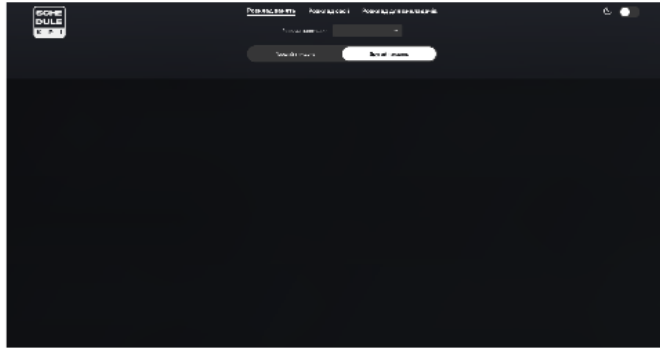
Київ – 2023



					КПІ.ІТ-9203.045440.06.99.СС			
					Схема структурна варіантів використання	<i>Лит.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>				1
<i>Розроб.</i>		Бойко Є.К.						
<i>Перев.</i>		Фіногенов О.Д.						
<i>Т. Кон.</i>								
<i>Н. Кон.</i>		Вітковська			Імплементація API для платформи Schedule KPI	КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-92		
<i>Затв.</i>		Фіногенов О.Д.				<i>Аркуш</i>	<i>Аркушів</i>	

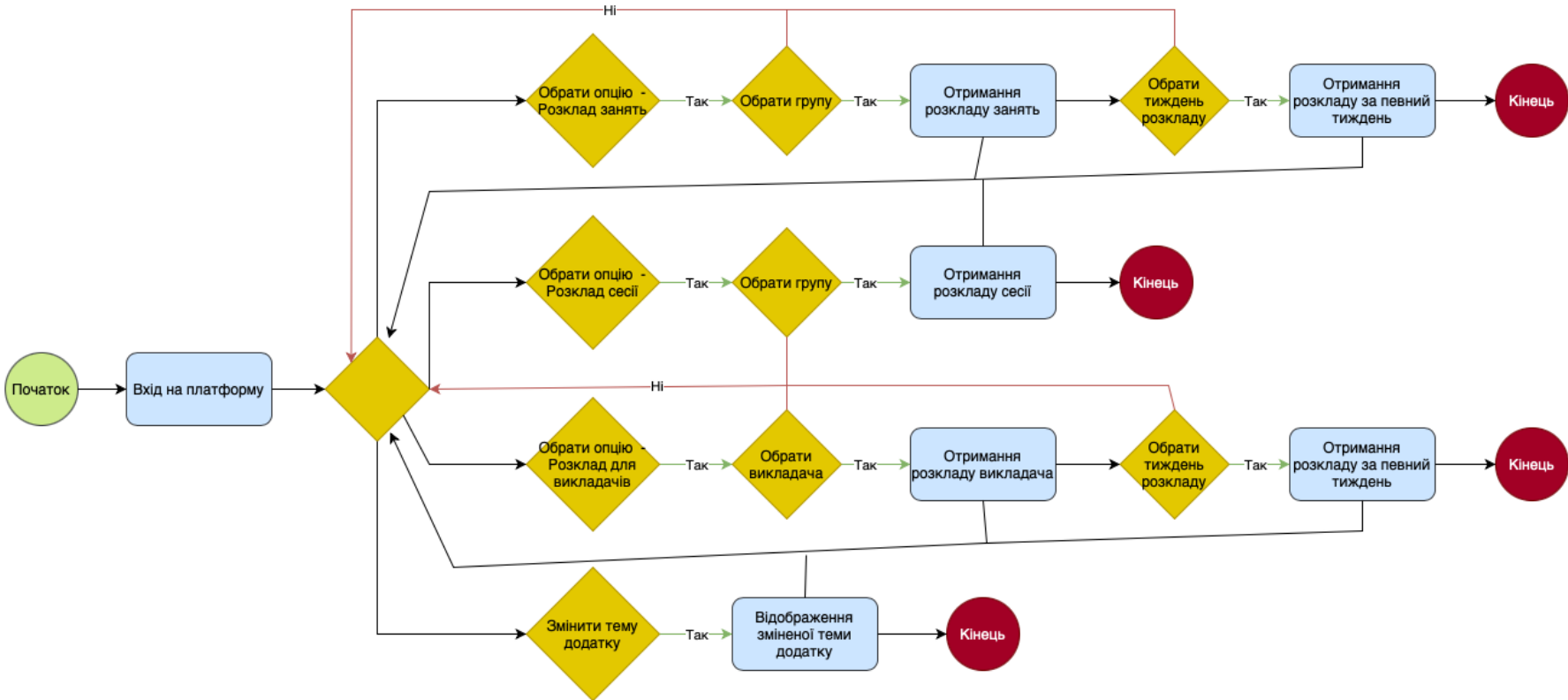


					<i>КПІ.ІТ-9203.045440.07.99.СБД</i>					
					Схема бази даних			Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив		Бойко Є.К.								
Перевірів		Фіногенов О.Д.								
Т. кон.							Аркуш	Аркушів		
Н. кон.		Вітковська І.І.			Імплементація API для платформи Schedule KPI			КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-92		
Затвердив		Фіногенов О.Д.								



Зм.	Арк.	№ докум.	Підп.	Дата	
Розроб.		Бойко Є. К.			
Перев.		Фіногенов О.Д.			
Т. Кон.					
Н. Кон.		Вітковська І.І.			
Затв.		Фіногенов О.Д.			

KPI.IT-9203.045440.08.99.KE			
Креслення вигляду екранних форм	Лит.	Арк.	Аркуші
			1
Імплементація API для платформи Schedule KPI	Аркуш	Аркуші	
	КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-9Х		



					<i>КПІ.ІТ-9203.045440.06.99.СБП</i>				
Зм.	Арк.	№ документа	Підпис	Дата	Схема бізнес-процесу		Літера	Маса	Масштаб
Розробив		Бойко Є.К.							
Перевірів		Фіногенов О.Д.							
Т. кон.							Аркуш	Аркушів	
Н. кон.		Вітковська І.І.			Імплементація API для платформи Schedule KPI		КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-92		
Затвердив		Фіногенов О.Д.							