

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Наталія АУШЕВА

«\_\_\_» \_\_\_\_\_ 2022 р.

**Дипломна робота**

на здобуття ступеня бакалавра

спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерний моніторинг та геометричне

моделювання процесів і систем»

на тему: «Система-довідник «Студент-викладач» кафедри»

Виконала:

студентка ІV курсу, групи ТР-82

Скорик Анна Юріївна



Керівник:

к.т.н., доц.

Недашківський Олексій Леонідович



Рецензент:



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

Студентка \_\_\_\_\_



Київ – 2022

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший

спеціальність 122 «Комп’ютерні науки»

освітня програма «Комп’ютерний моніторинг та геометричне моделювання процесів і систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Наталія АУШЕВА

(підпис)

” ” \_\_\_\_\_ 2022р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

Скорик Анни Юріївни

(прізвище, ім’я, по батькові)

1. Тема роботи «Система-довідник «Студент-викладач» кафедри».

керівник роботи доц., Недашківський Олександр Леонідович

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

затверджена наказом вищого навчального закладу від ”08” червня 2022р.

№ \_\_\_\_\_

2. Строк подання студентом роботи : 10.06.2022

3. Вихідні дані до роботи: загальна постановка задачі та індивідуальне завдання.

Задачею бакалаврської роботи є розробка програмного забезпечення для візуалізації тривимірного перетворення кулі з можливістю деформації в режимі роботи.

Індивідуальним завданням є створення зрозумілого та доступного інтерфейсу користувача з можливістю переглядати профілі викладачів, та переглядати/редагувати необхідну інформацію для навчального процесу студента.

4. Зміст розрахунково-пояснювальної записки

В пояснювальній записці будуть розглянуті питання:

а) аналіз проблеми та актуальності розроблюваного продукту

б) аналіз мов програмування, бібліотек для реалізації продукту

- в) конкретизація та декомпозиція поставленої задачі
- г) розробка програмного продукту
- д) представлення користувацького дизайну розробленого ресурсу
- е) висновки

5. Перелік ілюстративного матеріалу: скріншоти повного функціоналу готового продукту, скріншоти архітектури коду проекту.

6. Дата видачі завдання ” 9 ” лютого 2022 р.

### КАЛЕНДАРНИЙ ПЛАН

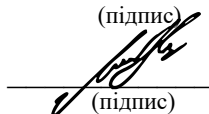
№ п/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	10.09.2021	
2.	Вивчення та аналіз задачі	02.05.2022-04.05.2022	
3.	Чітка постановка задачі; розробка архітектури бази даних та дизайну майбутньої системи	05.05.2022-11.05.2022	
4.	Розробка архітектури коду проекту та всіх компонентів	12.05.2022-14.05.2022	
5.	Програмна реалізація системи	15.05.2022-25.05.2022	
6.	Оформлення пояснювальної записки	02.05.2022-27.05.2022	
7.	Захист програмного продукту	22.05.2022-27.05.2022	
8.	Передзахист	6.06.2022-9.06.2022	
9.	Захист		

Студент



(підпис)

Керівник роботи



(підпис)

Скорик А.Ю.

(прізвище та ініціали,)

Недашківський О.Л.

(прізвище та ініціали,)

## **АНОТАЦІЯ**

Кількість сторінок роботи: 47; кількість зображень: 35. Робота містить 4 додатка та посилається на 13 бібліографічних найменувань.

Мета даної роботи – це розробка та створення довідникового ресурсу «Студент-викладач» в рамках кафедри університету. Для розробки логіки цього ресурсу були використані мови PHP, JavaScript та бібліотека JQuery. Для розробки візуальної частини використовувались JavaScript, CSS та PHP. Зв'язок з базою даних відбувається через SQL запити.

## **THE SUMMARY**

Number of pages of work: 47; number of images: 35. The work contains 4 appendices and refers to 13 bibliographic titles.

The purpose of this work is to develop and create a reference resource "Student-teacher" within the department of the university. The PHP, JavaScript and JQuery libraries were used to develop the logic of this resource. JavaScript, CSS and PHP were used to develop the visual part. The database is connected via SQL queries.

# ЗМІСТ

ВСТУП.....	7
1 ЗАДАЧІ РОЗРОБКИ СИСТЕМИ «СТУДЕНТ-ВИКЛАДАЧ».....	9
1.1 Створення бази даних .....	9
1.2 Створення модуля «Викладач» та його компонентів .....	9
1.3 Створення модуля «Студент» та його компонентів.....	10
1.4 Створення модуля «Перегляд викладача» та вікна «Логін».....	11
2 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ .....	12
2.1 КПІ Розклад.....	12
2.2 КПІ Кампус.....	13
2.3 Google classroom .....	14
2.4 Twitter.....	16
3 ЗАСОБИ РОЗРОБКИ .....	18
4 ОПИС РЕАЛІЗАЦІЇ ПРОДУКТУ .....	23
4.1 Створення концепції проекту.....	23
4.1.1 UML діаграми .....	23
4.1.2 Карта сайту.....	27
4.1.3 Загальна структура БД .....	29
4.1.4 Можливості ролей .....	30
4.2 Створення бази даних .....	32
4.3 Створення модуля «Викладач» та його компонентів .....	39

4.4	Створення модуля «Студент» та його компонентів.....	42
4.5	Створення модуля «Перегляд викладача» та вікна «Логін» .....	45
5	РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ .....	48
6	НАПРЯМКИ РОЗВИТКУ ПРОЕКТА .....	49
6.1	Розвиток модуля «Викладач».....	49
6.2	Розвиток модуля «Студент».....	50
6.3	Створення модуля «Адміністратор» .....	51
	ВИСНОВКИ.....	53
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
	ДОДАТОК А.....	56
	ДОДАТОК Б .....	64
	ДОДАТОК В.....	68
	ДОДАТОК Г .....	79

## ВСТУП

В сучасному світі якісна освіта – це ключова частина життя кожної людини. Однак за останні кілька років реалії децю змінилися та отримання якісних повноцінних знань в дистанційному форматі стало децю інакшим від всім звичного очного підходу. Саме тому зараз розробляється велика кількість різноманітних онлайн ресурсів для організації навчання онлайн. Стрімке збільшення платформ для навчання звичайно є хорошим показником напрямку розвитку технологій направленим на навчання, але чи завжди така варіативність буде зручною для одного студента?

Основною проблемою постає те, що немає жодного стандарту в межах нашого університету, що дозволяє узгодити єдині ресурси для моніторингу всієї необхідної інформації по навчальному процесу для студента. Звичайно вже є існуючі ресурси, такі як «КПІ Розклад», «КПІ кампус», сайт нашої кафедри, електронна пошта студента, електронна пошта групи, «Google classroom», «Google drive», соціальна мережа «Телеграм» та ще інші ресурси відповідно до індивідуальних вимог викладачів.

Це різноманіття часто створює проблеми з пошуком інформації в дистанційному режимі. Адже всі ці ресурси є вузько спеціалізованими або взагалі можуть покривати однаковий функціонал. Наприклад викладач може надіслати матеріали на «КПІ кампус», електронну пошту студента, електронну пошту групи, «Google classroom», «Google drive» або соціальну мережу «Телеграм».

Також існуючі ресурси від нашого університету, такі як «КПІ Розклад» та «КПІ кампус» є хорошим рішенням для своїх форматів. Однак для студента було б значно зручніше мати доступ до цієї інформації не переходячи за безліччю посиланнями, а використовуючи комплексне рішення, що узагальнює всю необхідну інформацію для швидкого отримання необхідної інформації для освітнього процесу.

Метою цього проекту є розробка єдиного зручного ресурсу для навчання, що об'єднає як потреби студента, так і потреби викладача в організації навчального процесу, його моніторингу, зв'язку та наданні всієї необхідної інформації, яку можна зручно та швидко знайти.

# **1 ЗАДАЧІ РОЗРОБКИ СИСТЕМИ «СТУДЕНТ-ВИКЛАДАЧ»**

Була поставлена задача реалізації системи, що буде зручним ресурсом-помічником для студента. Основні функції якої є організація власного розкладу, можливість переглядати поточні завдання до виконання (для викладача їх призначати), функцію переходу на сторінку викладача кафедри для отримання всієї необхідної інформації про нього. Для отримання різноманітної важливої довідкової інформації, а також листування з викладачем в онлайн режимі. Основним моментом є прозорість інформації. Тут мусять бути відсутні приватні повідомлення. Вся інформація доступна або ж для загального розгляду, або для конкретної групи.

Дана система є об'ємною, тому за допомогою методу декомпозиції, задачу можна поділити на створення окремих частин та налагодження їх сумісної роботи.

## **1.1 Створення бази даних**

Так як користувач в розроблюваній системі працює з даними в різних поданнях, можна сказати, що основа цього проекту є саме база даних.

Саме від цього етапу залежить кінцевий результат та логіка готового продукту. Для цієї мети краще за все підійде MySQL.

## **1.2 Створення модуля «Викладач» та його компонентів**

Модуль «Викладач» представляє собою сторінку, на яку потрапляє викладач при авторизації. Ця сторінка містить:

- Ім'я викладача;
- Кабінет викладача;

- Чекбокс для ідентифікування чи викладач знаходиться в кабінеті в поточний час;

- Особистий розклад викладача на два тижня;
- Поле для вибору групи та публікування для неї інформації;
- Вікно «Дати завдання» з дедлайнами студентам з обраної групи;
- Чат з студентами;
- Контакти;
- Вікно «Редагувати профіль»;
- Кнопка виходу зі сторінкою.

### **1.3 Створення модуля «Студент» та його компонентів**

Модуль «Студент» представляє собою сторінку, на яку потрапляє студент при авторизації. Ця сторінка містить:

- Ім'я студента;
- Групу студента;
- Особистий розклад студента на два тижня;
- Перелік завдань доступних до виконання студенту з можливістю зміни їх статусу з «Не виконано» на «Виконано»;
- Перелік викладачів на чиї сторінки можна перейти для отримання детальної інформації;
- Контакти;
- Вікно «Редагувати профіль»;
- Кнопка виходу зі сторінки.

## 1.4 Створення модуля «Перегляд викладача» та вікна «Логін»

Модуль «Перегляд викладача» представляє собою сторінку, на яку потрапляє студент при виборі викладача на якого він хоче перейти. Ця сторінка містить:

- Ім'я викладача;
- Кабінет викладача;
- Статус чи викладач знаходиться в кабінеті в поточний час;
- Особистий розклад викладача на два тижня;
- Поле з інформацією для групи авторизованого студента;
- Загальний чат з викладачем, де можна задати йому питання;
- Контакти викладача;
- Кнопка «Додому».

Модуль «Логін» представляє собою початкову сторінку при переході на даний ресурс. Ця сторінка містить:

- Поле «Логін»;
- Поле «Пароль»;
- Кнопку «Увійти».

## 2 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Університетські ресурси для узагальнення інформації студентам тематика не нова, та зокрема в нашому університеті є кілька подібних ресурсів. Кожен з яких вирішує конкретні питання та виконує поставлені задачі. Розглянемо вищезгадані «КПР Розклад» та «КПР Кампус».

### 2.1 КПР Розклад

Розглянемо ресурс «КПР Розклад» [5].

Переваги:

- 1) Зручний доступ до готового розкладу.
- 2) Перегляд розкладу усіх викладачів університету.
- 3) Перегляд дат екзаменів під час сесії.

Недоліки:

- 1) При зміні розкладу дані довгий час є невалідними.
- 2) Щоб подивитися розклад викладача потрібно чітко знати його повне прізвище, ім'я, по-батькові, що не завжди виявляється реальним. Доводиться довгий час шукати цю інформацію на сайтах кафедри.
- 3) Часто викладачі роблять тимчасові зміни в розкладі зі згоди студентів, що робить офіційний розклад невалідним.

Цей ресурс є орієнтованим лише на офіційні дані розкладу і лише розкладу. Більше ніякого стороннього функціоналу він не підтримує.

На рисунку 2.1 продемонстровано головний екран ресурсу «КПР Розклад».

департамент організації освітнього процесу

Розклад занять для TR-92

Перший тиждень

Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота
1 08:30 Методи та системи штучного інтелекту доц. Шаповалова С. І. 419-05 Лаб	Інформаційна безпека ст.вик. Радченко О. Г. Lex on-line	Інформаційна безпека ст.вик. Радченко О. Г. 419-05 Прак		Методи та системи штучного інтелекту доц. Шаповалова С. І. Lex on-line	
2 10:25 Новозна мова професійного спілкування 1. вик. Назаренко І. М. 325-05 Прак	Аналитика Интернету речей доц. Гусова І. І. Lex on-line	Вступ до інтелектуального аналізу даних ст.вик. Бандурка О. І. 421-05 Лаб		Технології розподілених систем та паралельних обчислень доц. Лабківський В. А. Lex on-line	
3 12:20 Аналитика Интернету речей ст.вик. Діачок О. А. 508-05 Лаб	Вступ до інтелектуального аналізу даних проф. Веремь А. А. Lex on-line	Кросплатформна розробка мобільних застосунків проф. Недашківський О. Л. 271-05 Лаб		Java-програмування доц. Теремський Ю. А. Lex on-line	
4 14:15	Кросплатформна розробка мобільних застосунків проф. Недашківський О. Л. Lex on-line			Предметно-орієнтовані мови програмування: розробка та реалізація доц. Стівка Ю. І. Lex on-line	
5 16:10					
6 18:30					

Другий тиждень

Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота
1 08:30 Методи та системи штучного інтелекту доц. Шаповалова С. І. 419-05 Лаб		Предметно-орієнтовані мови програмування: розробка та реалізація доц. Стівка Ю. І. 419-05 Лаб		Методи та системи штучного інтелекту доц. Шаповалова С. І. Lex on-line	
2 10:25 Новозна мова професійного спілкування 1. вик. Назаренко І. М. 325-05 Прак	Аналитика Интернету речей доц. Гусова І. І. Lex on-line	Технології розподілених систем та паралельних обчислень доц. Лабківський В. А. 524-05 Лаб		Технології розподілених систем та паралельних обчислень доц. Лабківський В. А. Lex on-line	
3 12:20 Java-програмування доц. Теремський Ю. А. 519-05 Лаб	Вступ до інтелектуального аналізу даних проф. Веремь А. А. Lex on-line			Java-програмування доц. Теремський Ю. А. Lex on-line	
4 14:15	Кросплатформна розробка мобільних застосунків проф. Недашківський О. Л. Lex on-line			Предметно-орієнтовані мови програмування: розробка та реалізація доц. Стівка Ю. І. Lex on-line	
5 16:10					
6 18:30					

Рисунок 2.1 – КПІ Розклад

Можемо бачити простоту користувацького інтерфейсу та вузьку направленість функціоналу.

## 2.2 КПІ Кампус

3 Розглянемо ресурс «КПІКампус» [6].

Переваги:

- 1) Перегляд особистих результатів в навчанні, сесії та заліках.
- 2) Перегляд інформації від кафедри (оголошення).
- 3) Перегляд матеріалів лекцій, що завантажили викладачі.

Недоліки:

- 1) Застарілий дизайн
- 2) Матеріали лекцій завантажує відносно мала кількість викладачів, тому часто ця інформація не переглядається/губиться з поміж інших джерел. А даний функціонал здебільшого не застосовується студентами.

Зрештою попри різні дані, що можна переглянути на цьому ресурсі, фактично він використовується лише як таблиць, що не покриває поставленої проблеми даної роботи.

На рисунку 2.2 продемонстровано головний екран ресурсу “КПІ Кампус”.

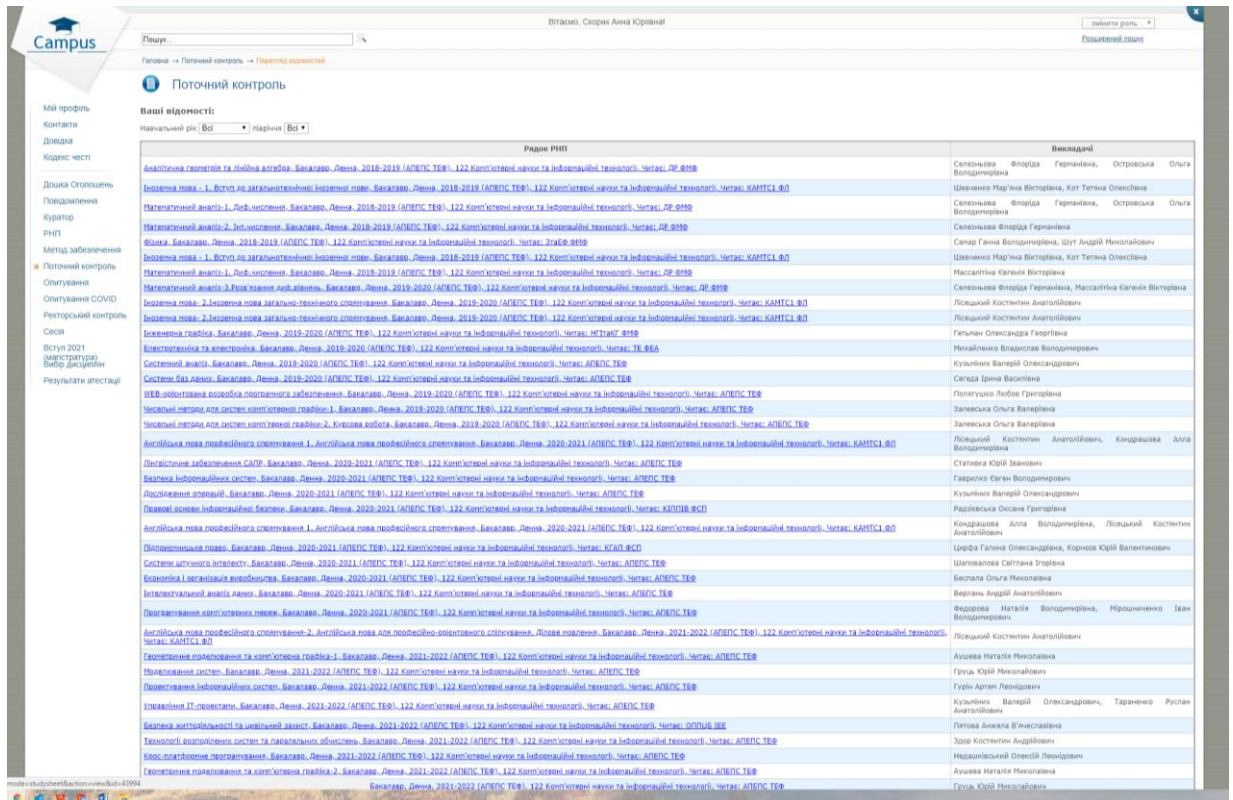


Рисунок 2.2 – КПІ Кампус

З цього рисунку можемо бачити функціонал ресурсу та навантаженість його візуалу.

## 2.3 Google classroom

Розглянемо ресурс «Google Classroom» [7].

Переваги:

- 1) Зручний перегляд та задача завдань
- 2) Можливість переглядати оцінки за здані роботи

3) Сповіщення на особисту електронну адресу про найближчий дедлайн або пропущений строк здачі.

4) Можливість прикріплювати матеріали до завдань.

Недоліки:

1) Ресурс більше направлений саме на проведення освітнього процесу, а не налагодження інформаційного потоку при дистанційному режимі.

На рисунку 2.3 продемонстровано головний екран ресурсу “Google Classroom”.

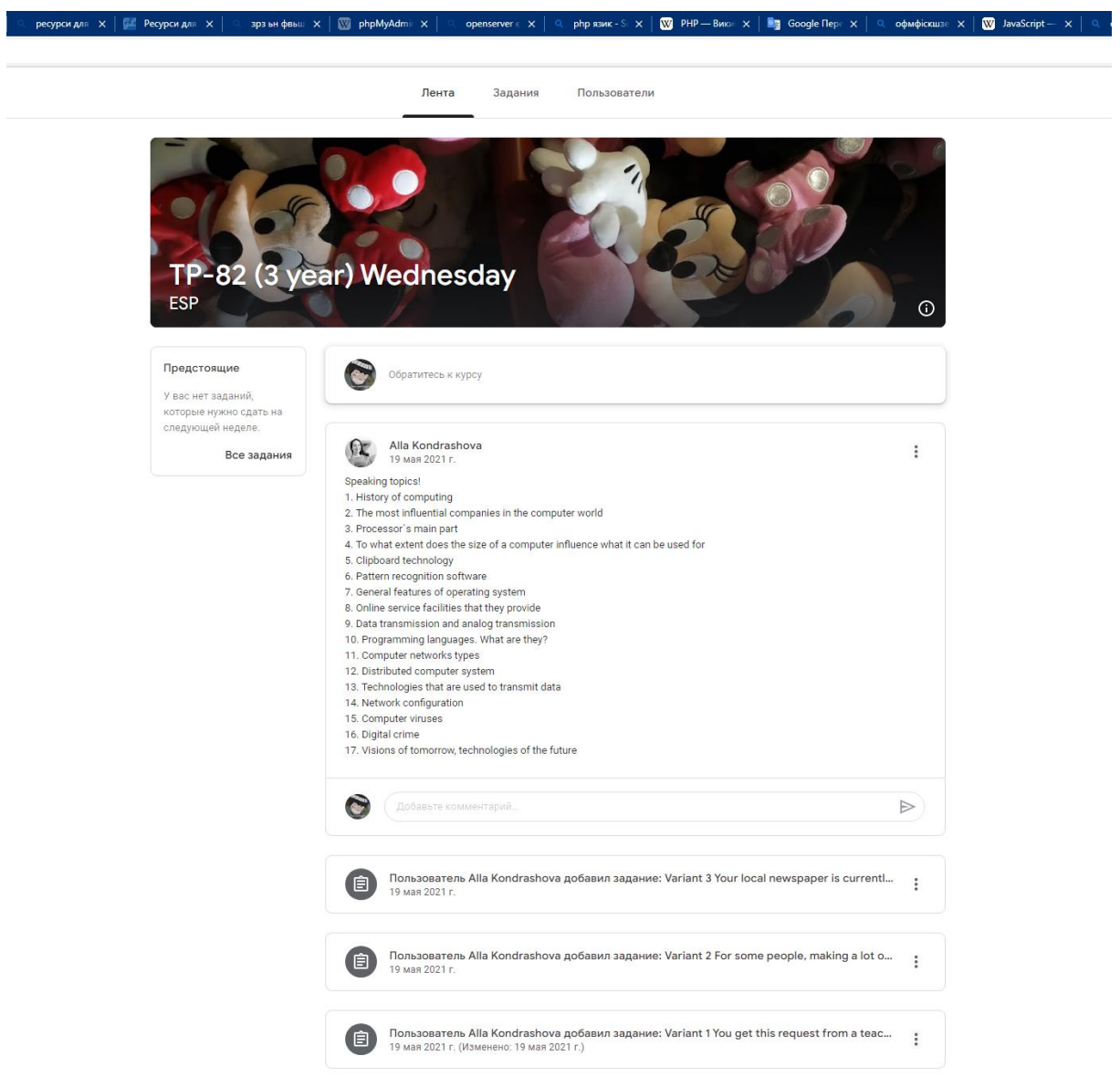


Рисунок 2.3 – Google Classroom

Даний рисунок демонструє сучасність дизайну та направленість функціоналу даного ресурсу.

## 2.4 Twitter

Розглянемо ресурс «Twitter» [8].

Попри те, що ресурс «Twitter» - це соціальна мережа, його концепція здається унікальною. Тому само його будемо використовувати як референс до розроблюваного ресурсу «Студент-викладач».

Як можна побачити на рис. 2.4, сторінка користувача демонструє дані лише про цього користувача у форматі стрічки з постів чи репостів. Така ж концепція буде застосовуватися до системи-довідника.

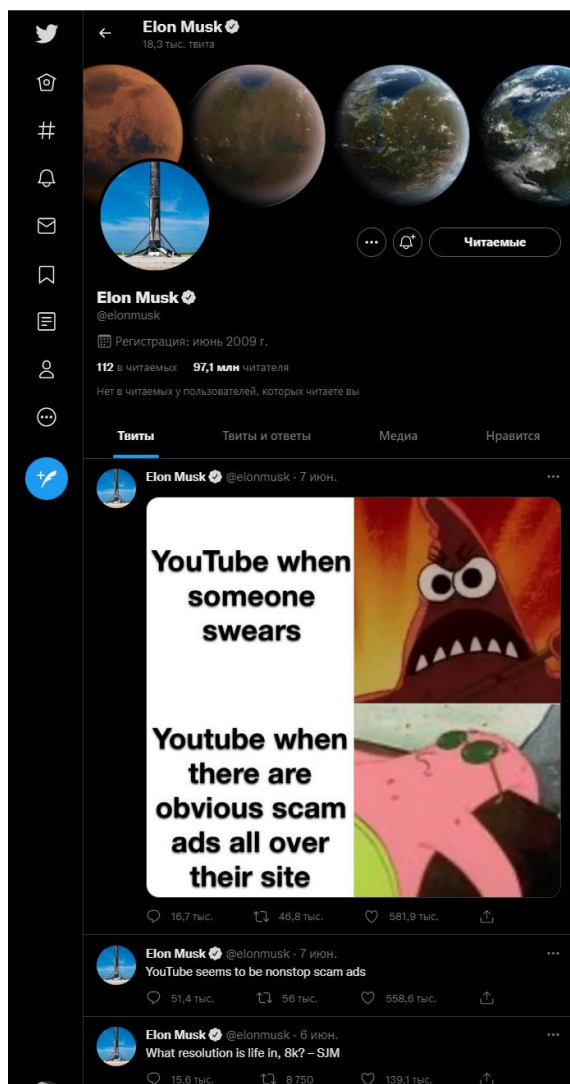


Рисунок 2.4 Вигляд стрічки соціальної мережі «Twitter»

Розроблюваний дизайн виглядатиме як стрічка з різною інформацією про користувача. Особливо чітко цю аналогію можна буде прослідкувати при перегляді викладача.

## 3 ЗАСОБИ РОЗРОБКИ

Під час виконання даного проекту було розглянуто та ознайомлено з значною кількістю технологій, мов та засобів програмування, СУБД. Під час ознайомленню з якими було знайдено такі технології та методи, при яких буде забезпечена найбільша продуктивність, гнучкість, простота користування системою. Також обрані методи розробки дають змогу без перешкод розширювати та доповнювати функціонал готового продукту.

За основу логіки було взято мову програмування Php [5] в поєднанні з JavaScript.

Php – це C-подібна скриптова мова, загального призначення, вона дуже активно застосовується для розробки веб-додатків. Наразі переважна більшість хостинг-провайдерів підтримує дану мову, що робить її беззаперечно чудовим вибором для написання сайту. Php є одним з лідерів серед всіх мов, яку використовують для написання динамічних веб-сайтів. Мова автоматично підтримує дані формату HTTP-Cookies відповідно до стандартів Netscape. Це дає можливість встановлювати та зчитувати невеликих порцій даних на стороні клієнта. Робота з Cookies організована за допомогою сесій. В них є термін дії, після завершення якого дані видаляються. У цих сесіях можна оперувати (зберігати чи редагувати) абсолютно різні типи вхідних даних (у тому числі серіалізовані дані та PHP-об'єкти).

Інтерпретатор складають основа (ядро) та модулі, які ще називають «розширення». Вони підключаються до проекту як динамічні бібліотеки та дозволяють розширяти базовий функціонал та можливості мови. Так можна отримати змогу працювати з різними інструментами, такими як бази даних [6], сокети, динамічна графіка, криптографічні бібліотеки, і навіть PDF документи.

Такі розширення може розробити та підключити будь-який розробник. Саме тому існує досить велика кількість розширень (не тільки стандартні, а і розроблені

сторонніми компаніями або програмістами-ентузіастами). Багато розширень є доступними для загального користування в репозиторії PECL.

На рис 3.1 зображена схема роботи будь-якого додатку, запрограмованого мовою PHP.

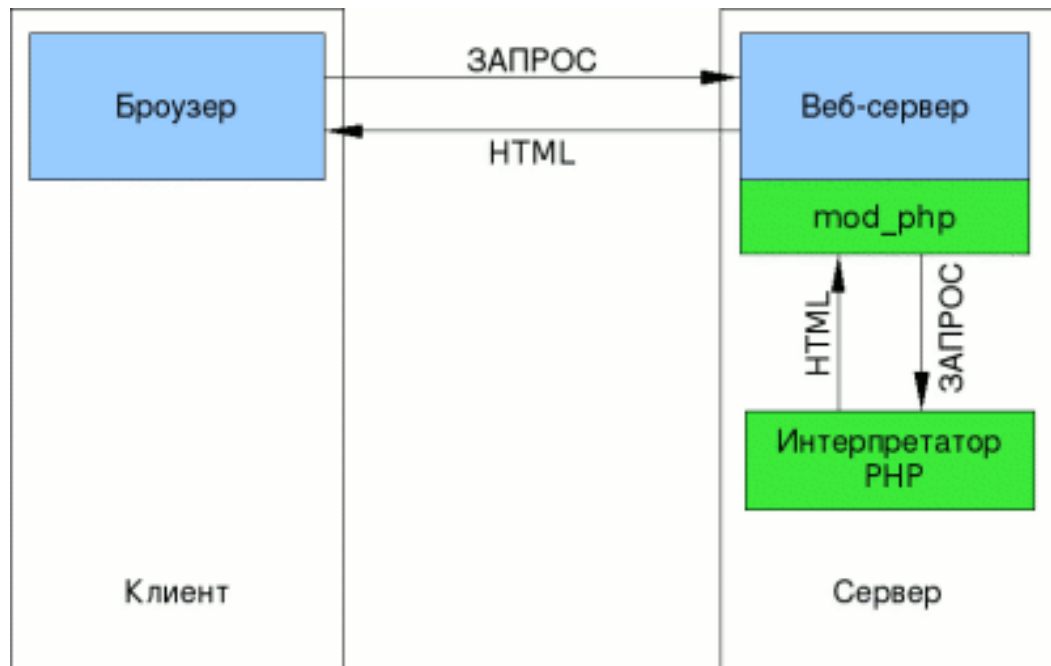


Рисунок 3.1 принцип роботи додатку мовою PHP

JavaScript [7] - це динамічна, об'єктно-орієнтована мова програмування. Найчастіше її використання – створення сценаріїв веб-сторінок сайтів. Це забезпечує можливість клієнта взаємодіяти з користувачем, асинхронно обмінюватися даними з сервером, виконувати дії в браузері, змінювати структуру/загальний вигляд сторінки.

JQuery [8] – це чи не найпопулярніша бібліотека JavaScript, яка дуже широко використовується на сьогодні. Синтаксис цієї бібліотеки націлений на створення анімацій, обробку подій, застосуванню AJAX [9] до застосунків. З JQuery розробники мають змогу створювати абстракції для взаємодії на низькому рівні та створювати анімацію для ефектів на високому рівні. Завдяки цьому

бібліотека JQuery користується такою популярністю та дає змогу створити справді потужний та універсальний веб-ресурс.

Принцип роботи JQuery полягає у використанню класу, який при зверненні повертає сам себе. Так можна побудувати логічний і послідовний ланцюжок методів.

При розробці даного проекту також застосовувався підхід асинхронних запитів AJAX.

AJAX – це підхід до побудови інтерфейсів веб-додатків, при яких веб-сторінка, не перезавантажуючись, у фоновому режимі посилає запити на сервер і сама звідти довантажує дані, що потрібні користувачу. AJAX – один із компонентів концепції DHTML.

Модель AJAX виглядає так (рис. 3.2) [10]:

1. Користувач, що перебуває на веб-сторінці натискає на будь-який елемент на ній.
2. Браузер відповідно відправляє запит до сервера.
3. Сервер повертає лише змінену частину документа.

## How AJAX Works

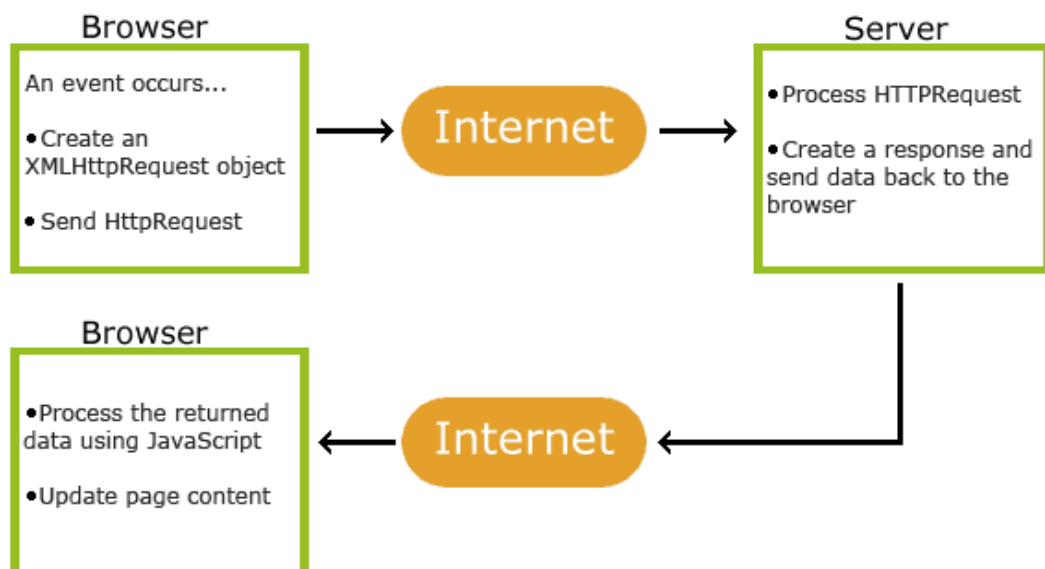


Рисунок 3.2 Модель AJAX

CSS [11] – походить від назви «каскадні таблиці стилів». Є мова для описання зовнішнього вигляду документа, що описаний однією з мов розмітки. За часту працює у зв'язці з HTML[12] та XHTML. Або ж до будь-яких документів формату XML. Ця мова застосовується в цілях задання особливих шрифтів, зручного та приємного користувацького інтерфесйу, стилів та розміщень окремих блоків на сторінці. Фактично файл CSS – це звичайний тестовий файл з переліком CSS правил для опису вигляду веб-сторінки.

Застосування CSS [13] базується на наслідуванні та каскадному принципі. Наслідування в даному випадку виражається тим, що властивості, застосовані для старших елементів наслідуються молодшими (нащадками).

Каскадний принцип застосовується лише в тому випадку, коли до одного елемента застосовані кілька властивостей одночасно. При їх конфлікті застосовується властивість з найвищим пріоритетом.

Далі пріоритети за зростанням:

- Найнижчий пріоритет - це стиль браузера;
- Налаштований стиль користувачем в налаштуваннях браузера;
- Стиль, що був заданий безпосередньо автором сторінки;

На рисунку 3.2. зображена схема розвитку стандартів мови CSS.



Рисунок 3.2 Стандарти мови CSS

Під час розробки даного проекту, буде використовуватися компіляція під час запуску сторінки на стороні клієнта (ресурсами JavaScript).

## 4 ОПИС РЕАЛІЗАЦІЇ ПРОДУКТУ

### 4.1 Створення концепції проекту

#### 4.1.1 UML діаграми

Розпочинаючи будь-який новий проект, важливо чітко продумати всі нюанси та проаналізувати можливі дії користувача з системою. Саме для таких цілей придумали незліченну кількість різноманітних діаграм для опису розроблюваного проекту.

В досліджуваному випадку дуже важливим є саме візуальне представлення процесів в системі, структури БД та функціоналу, доступного користувачу.

Діаграма прецедентів (або її ще називають діаграма UML, діаграма варіантів використання) демонструє відношення між акторами та прецедентами в системі, що наглядно демонструє всі дії користувача в системі.

На діаграмі (рис. 4.1) продемонстрована діаграма прецедентів для актора Студент.

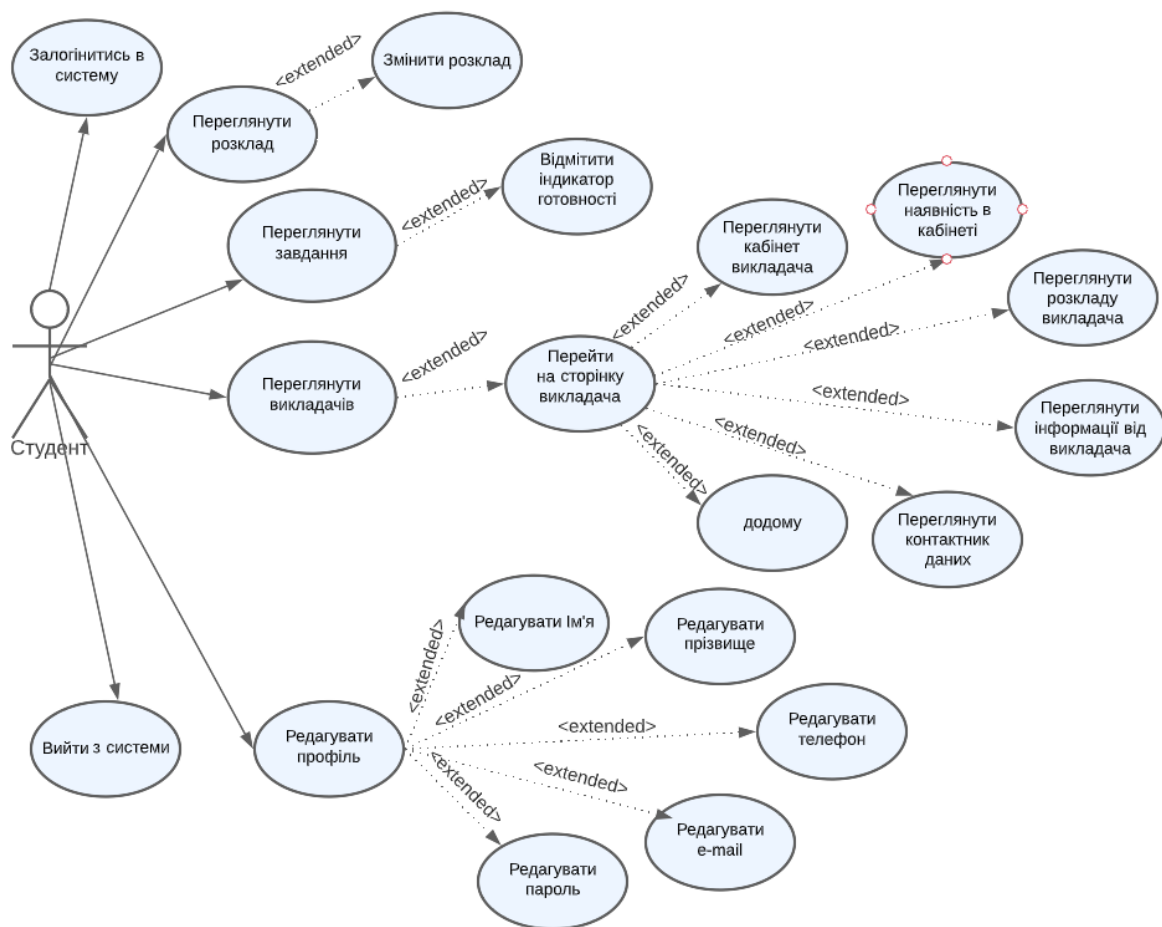


Рисунок 4.1. діаграма прецедентів для актора Студент.

Як можна помітити на рисунку вище, Студент може виконувати такі дії:

- Авторизуватись в систему;
- Переглянути розклад;
  - Змінити розклад;
- Переглянути завдання;
  - Відмітити ідентифікатор готовності;
- Переглянути викладачів;
  - Перейти на сторінку викладача;
    - Переглянути кабінет;
    - Переглянути наявність;
    - Перегляд розкладу;

- Перегляд інформації для групи;
  - Перегляд контактів;
  - Написати повідомлення в чат;
  - Повернутися додому
- Редагувати профіль
    - Редагувати ім'я;
    - Редагувати прізвище;
    - Редагувати телефон;
    - Редагувати e-mail;
    - Редагувати пароль;
  - Вийти з системи.

На діаграмі (рис. 4.2) Продемонстрована діаграма прецедентів для актора Викладач.

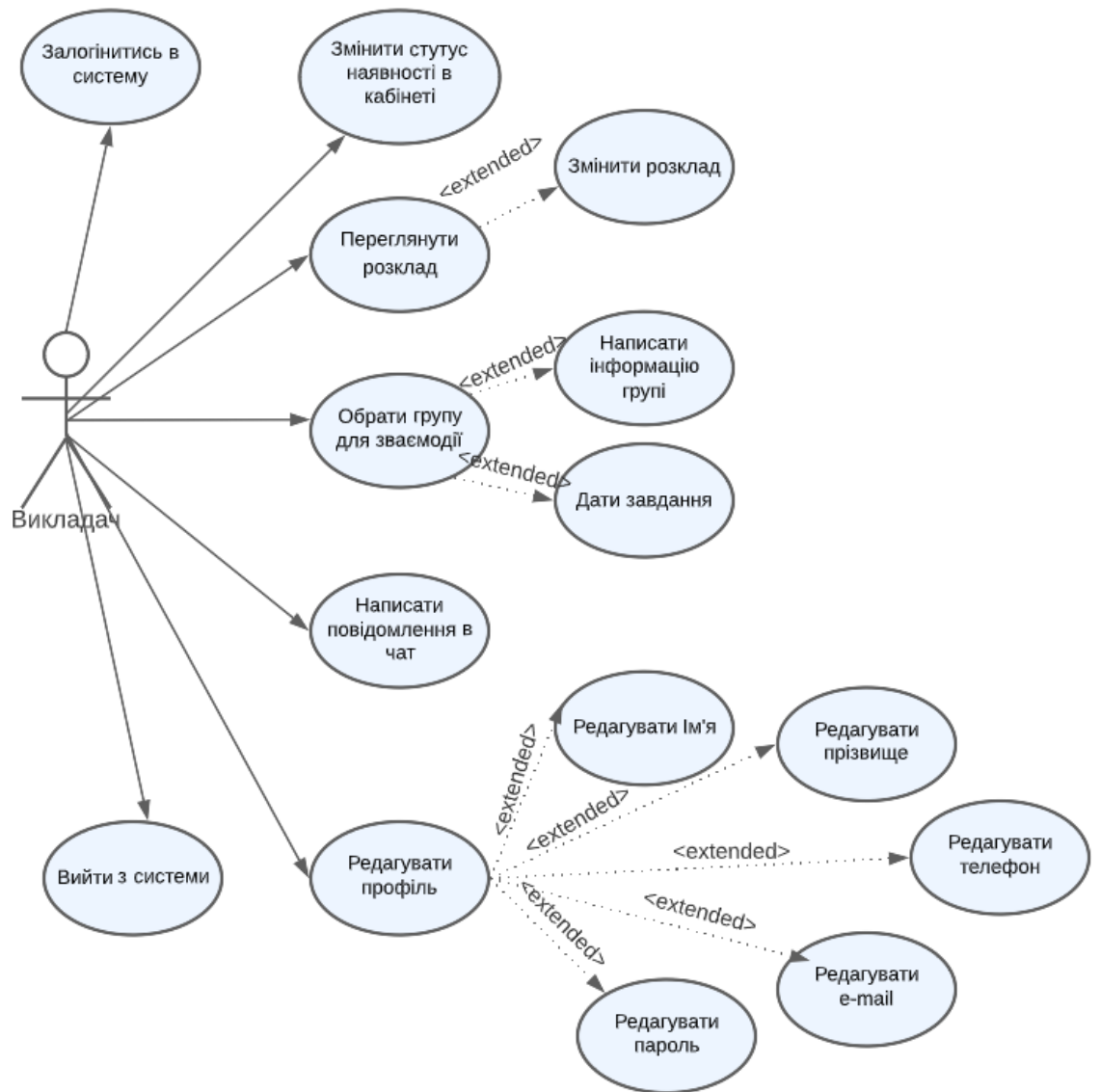


Рисунок 4.2 діаграма прецедентів для актора Викладач.

Відповідно до рисунку вище, Викладач може виконувати такі дії:

- Авторизуватись в систему;
- Переглянути розклад;
  - Змінити розклад;
- Змінити статус наявності в кабінеті;
- Обрати групу для взаємодії;
  - Написати інформацію групі;
  - Дати завдання групі;
- Написати повідомлення в чат;

- Редагувати профіль;
  - Редагувати ім'я;
  - Редагувати прізвище;
  - Редагувати телефон;
  - Редагувати e-mail;
  - Редагувати пароль;
- Вийти з системи.

#### **4.1.2 Карта сайту**

Для чіткого розуміння візуалу, логіки та структури проекту також необхідно описати безпосередньо взаємодію сторінок на навігацію між ними. Для цього використовують Карту сайту. Дана діаграма ілюструє зв'язки між сторінками та логіку використання ресурсу.

В межах даного проекту Карта сайта буде виглядати наступним чином (рис. 4.3)

Схема взаємодії та переходу сторінок в Системі-довіднику "Студент-викладач" кафедри

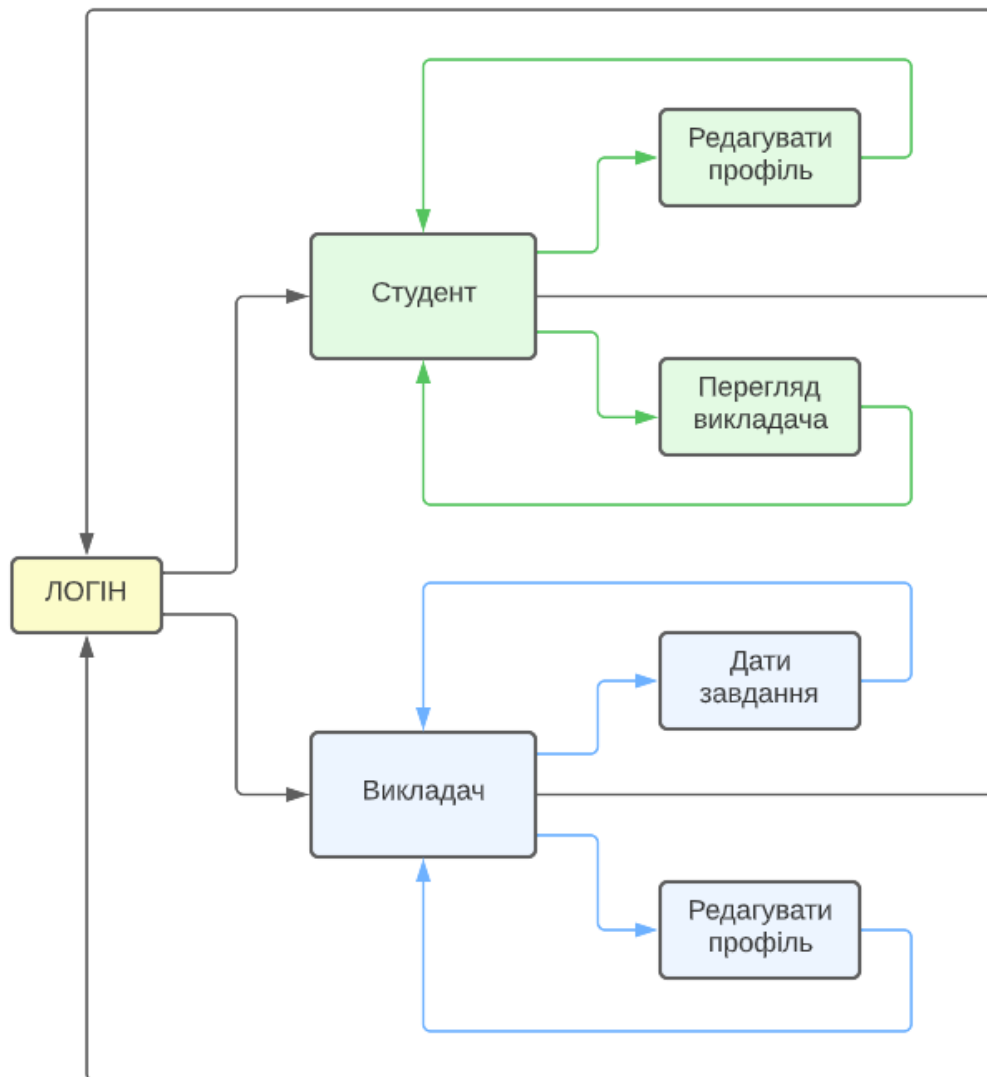


Рисунок 4.3 карта сайту «Студент-викладач».

Окремі блоки позначають доступні сторінки ресурсу.

Зеленим кольором продемонстровані взаємодії безпосередньо всередині модуля «Студент».

Синім кольором продемонстровані взаємодії всередині модуля «Викладач».

Чорні стрілки демонструють авторизацію та вихід зі сторінки користувача – повернення до сторінки «Логін».



Продемонстрована база даних містить в собі 8 таблиць, а саме:

- Користувачі;
- Студенти;
- Викладачі;
- Завдання;
- Групи;
- Інформація для групи;
- Чат;
- Розклад.

#### **4.1.4 Можливості ролей**

Для позначення та поділу дозволеного функціоналу для кожної з ролей системи, на рис. 4.5 зображена таблиця ролей та їх можливостей.

Можливості ролей	Викладач	Учень
логін	+	+
Відмітити статус наявності в кабінеті	+	-
Переглянути статус наявності в кабінеті	+	+
Задати розклад	+	+
Переглянути перший/другий розклад	+	+
Дати завдання	+	-
Переглянути завдання	-	+
Змінити статус завдання	-	+
обрати групу з випадуючого списку	+	-
Дати інформацію групі	+	-
Переглянути інформацію для групи	+	+
Переглянути контакти	+	+
Переглянути всіх викладачів у системі	-	+
Перейти на сторінку перегляду викладача	-	+
редагувати профіль	+	+
вихід з додаткових вкладок на домашню сторінку	+	+
вихід з акаунту	+	+
переписка в чаті	+	+

Рисунок 4.5 Можливості ролей

Як можна помітити, кількість доступного функціоналу для викладача еквівалентна кількості доступного функціоналу студента.

## 4.2 Створення бази даних

При створенні бази даних, було використано MySQL. Для адміністрування таблиць було використано ресурс PhpMyAdmin. Інтегрувати його в проект будемо за допомогою OpenServer. OpenServer є портативним програмним комплексом, створений для того, щоб допомогти у розробці, налагодженні та тестуванні сайтів безпосередньо на комп'ютері.

Після завантаження OpenServer, та додавання проекту в папку domains в кореневій директорії папки, запускаємо застосунок, клікнувши правою кнопкою миші на значок на панелі внизу (рис. 4.6).

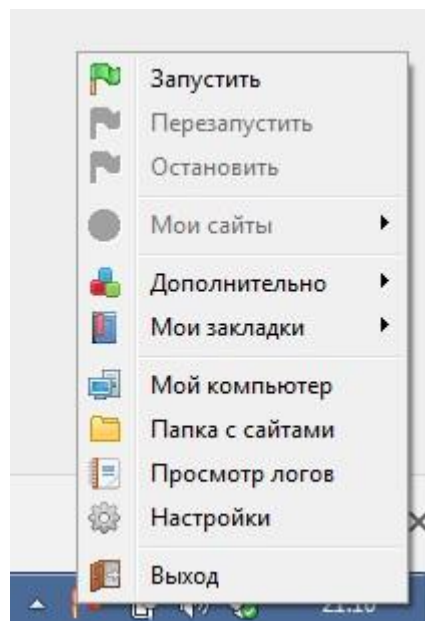


Рис. 4.6 Меню OpenServer.

Далі Переходимо у вкладку «Мої сайти», де після транспортування проекту, з'являються доступні опції (рис. 4.7).

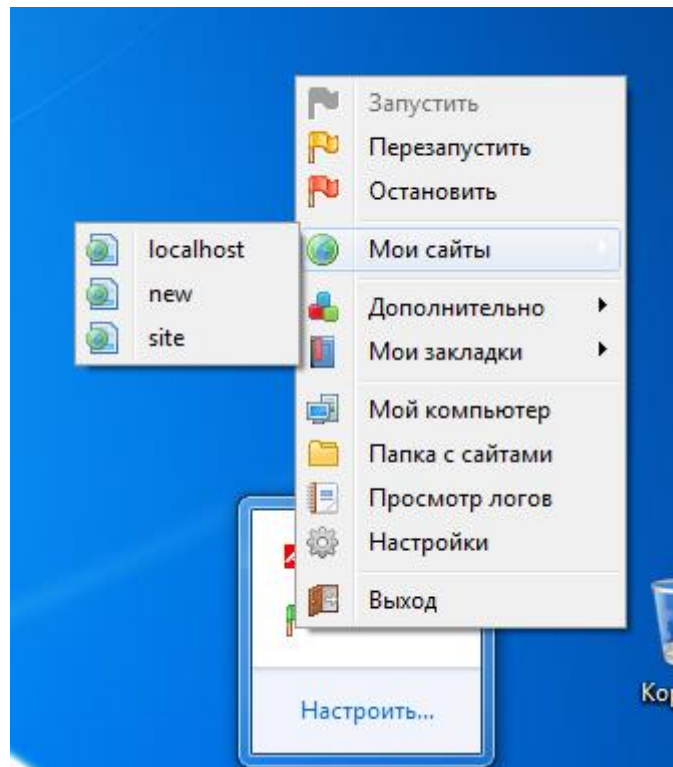


Рисунок 4.7 Відкриття веб-сторінки за допомогою OpenServer.

Для того, щоб під'єднати базу даних з PhpMyAdmin, необхідно перейти у вкладку Додатково – та обрати PhpMyAdmin (рис. 4.8).

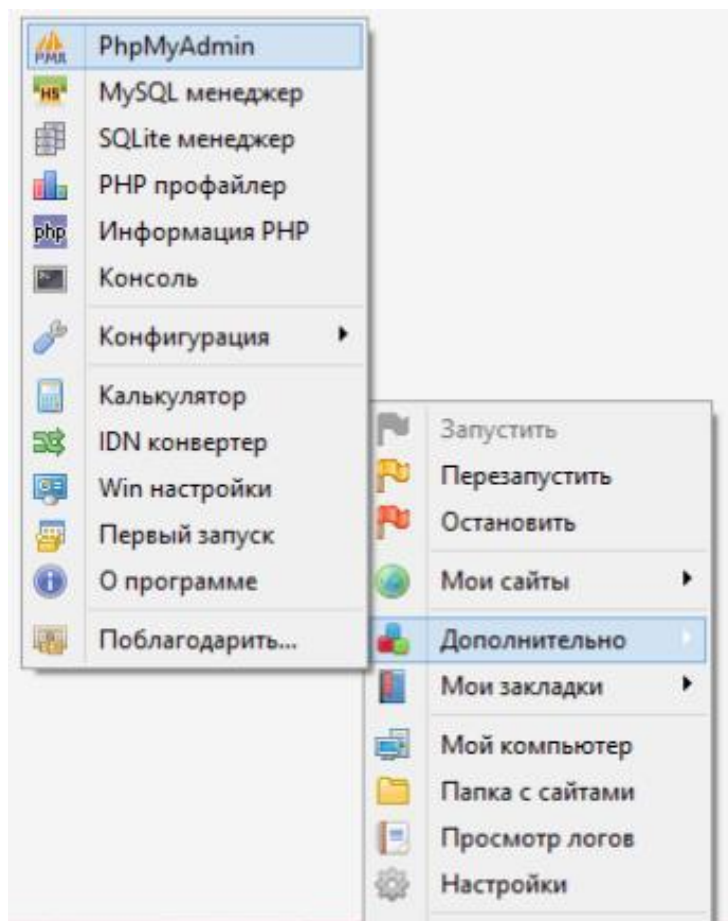


Рисунок 4.8 Приєднати БД

Після того як розібрались з процесом підключення всіх складових до OpenServer, розглянемо БД, що знаходиться на віддаленому ресурсі PhpMyAdmin.

При вході на цей ресурс, вказуємо логін “root”, поле паролю залишаємо порожнім (рис. 4.9).

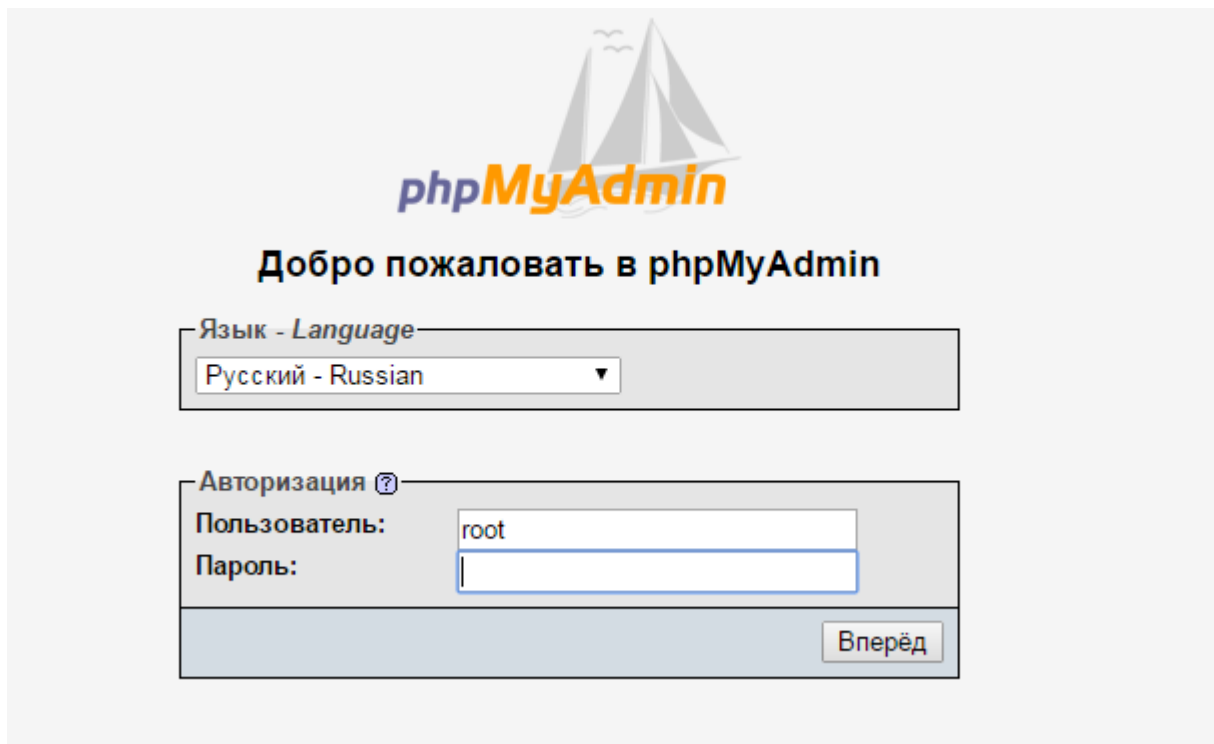


Рисунок 4.9 Вхід в PhpMyAdmin

Далі для створення безпосередньо самої БД, потрібно перейти на вкладку Бази даних, та натиснути на кнопку Створити (рис. 4.10).

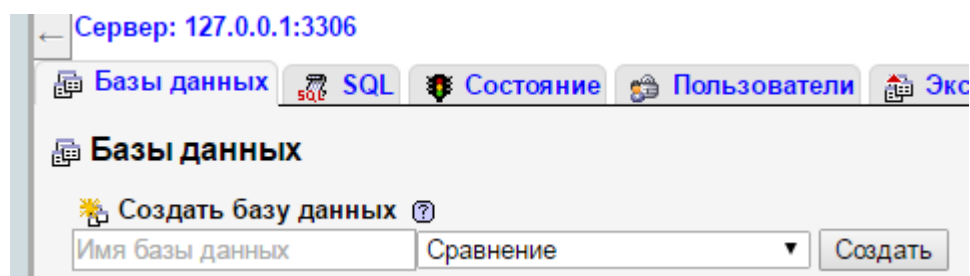


Рисунок 4.10 Створення БД

Після цих маніпуляцій збоку сторінки буде зображена щойно створена БД.

На рисунку 4.11 продемонстровано структуру БД розроблюваного ресурсу “Студент-викладач”.

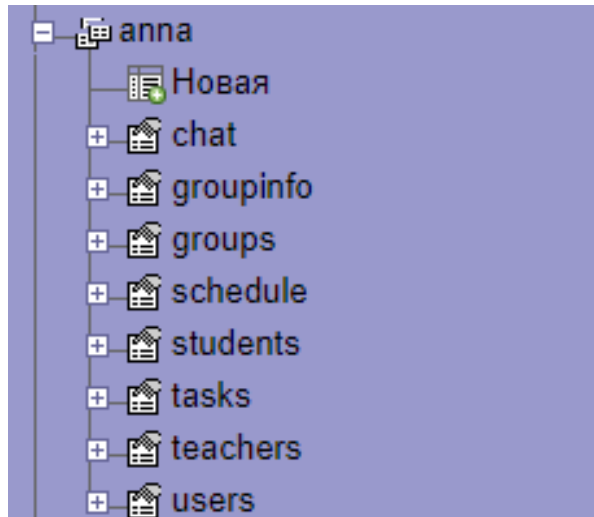


Рисунок 4.1 – Загальна структура БД

На рисунку 4.12 продемонстровано таблицю Чат.

	id	message	outcome_user_id	income_user_id	time
<input type="checkbox"/>	1	Доброго дня студенти, Для групи ТР-81 звіти скидат...	1	0	2022/05/25 11:40:46
<input type="checkbox"/>	2	Доброго дня! ви завтра будете в корпусі?	2	1	2022/05/25 12:11:29
<input type="checkbox"/>	3	Доброго дня! Підкажіть, будь ласка, для групи ТР-...	2	3	2022/05/25 12:12:45
<input type="checkbox"/>	4	Так робити!	3	0	2022/05/25 12:16:49

Рисунок 4.12 – Таблиця Чат

На рисунку 4.13 продемонстровано таблицю Інформації для групи.

	id	group_id	teacher_id	description
<input type="checkbox"/>	19	7	1	В завданні п.2 пропускаємо
<input type="checkbox"/>	21	2	1	пара переноситься на понеділок
<input type="checkbox"/>	22	7	3	Доброго дня студенти!
<input type="checkbox"/>	24	2	3	Через дорогу продають смачну каву!

Рисунок 4.13 – Таблиця Інформації для групи

На рисунку 4.14 продемонстровано таблицю Групи.

	id	name
<input type="checkbox"/>	1	TP-81
<input type="checkbox"/>	2	TP-82
<input type="checkbox"/>	3	TB-81
<input type="checkbox"/>	4	TB-82
<input type="checkbox"/>	5	TM-81
<input type="checkbox"/>	6	TM-82
<input type="checkbox"/>	7	TI-81
<input type="checkbox"/>	8	TI-82

Рисунок 4.14 – Таблица групп

На рисунку 4.15 продемонстрировано таблицу Розклад.

	id	description	teacher_id
<input type="checkbox"/>	59	{"first":{"Mo":["Вища математика Тр81", "", "", "", "", ...	1
<input type="checkbox"/>	62	{"first":{"Mo":["C++ TB82", "", "", "", "", "Tu":["...	3
<input type="checkbox"/>	66	{"first":{"Mo":["Пара1", "Пара1", "", "", "", "Tu":...	2
<input type="checkbox"/>	69	{"first":{"Mo":["", "", "", "", "", "Tu":["", "Астро...	4

Рисунок 4.15 – Таблица Розклад

На рисунку 4.16 продемонстрировано таблицу Студент.

	id	user_id	group_id
<input type="checkbox"/>	1	2	2
<input type="checkbox"/>	2	4	7

↑  Отметить все    С отмеченными:

Показать все | Количество строк: 50 ▾

Рисунок 4.16 – Таблица Студент

На рисунку 4.17 продемонстрировано таблицу Завдання.

id	teacher_id	deadline	description	group_id	subject
1	1	26.12.2000	Полити квіти	7	Лінійна алгебра
2	2	25.12.2020	Віджались 10 раз	7	C++
3	3	31.05.2023	Наліпити вареники	7	C++
4	3	31.05.2022	Прочитати параграф 8	7	C++
5	1	29.05.2022	Прибрати на кафедрі	7	Алалітична геометрія
8	1	30.05.2022	Помити дошку в аудиторії №2	7	Аналітична геометрія
9	1	26.12.2030	Заробити 20 грн	7	Лінійна алгебра
10	1	26.12.2000	Полити квіти	2	Лінійна алгебра
11	3	25.12.2020	Віджались 10 раз	2	C++
14	1	29.05.2022	Прибрати на кафедрі	2	Алалітична геометрія
15	1	30.05.2022	Помити дошку в аудиторії №2	2	Аналітична геометрія
16	1	26.12.2030	Заробити 20 грн	2	Лінійна алгебра

Рисунок 4.17 – Таблиця Завдання

На рисунку 4.18 продемонстровано таблицю Викладачі.

id	user_id	cabinet	is_online
1	1	207	1
2	3	604	0

Рисунок 4.18 – Таблиця Викладачі

На рисунку 4.19 продемонстровано таблицю Користувачі.

id	phone	name	lastname	login	password	email
1	+380930017311	Валерій	Конюшенко	teacher1	1001	valeii.koniushenko@gmail.com
2	+380333333335	Анна	Скорик	student1	2001	smorodina@gmail.com
3	+6663336663	Микола	Гребнонос	teacher2	1002	barbaris3000@gmail.com
4	+380689321055	Іван	Курочкін	student2	2002	ivan.kurochkin@gmail.com

Рисунок 4.19 – Таблиця Користувачі

Відповідно до цих рисунків можна чітко уявити структуру розроблюваного проекту. Дана база даних реалізована таким чином, щоб можна було легко доповнювати її новими таблицями, розширюючи функціонал системи.

### 4.3 Створення модуля «Викладач» та його компонентів

На рисунку 4.20 продемонстровано архітектуру програмного коду з виділеними методами для реалізації функціоналу «Викладач».

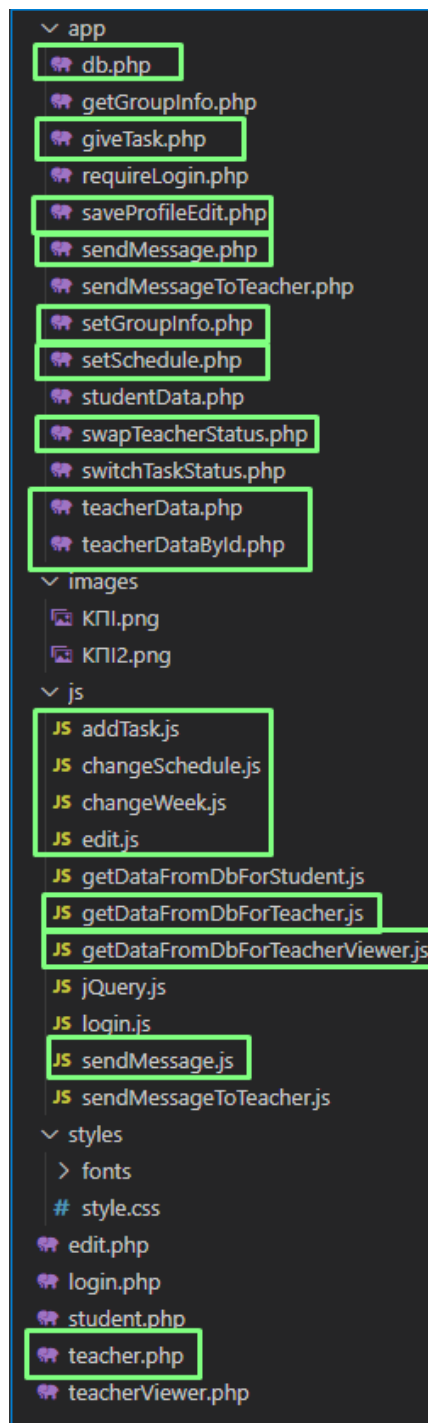


Рисунок 4.20 – Методи, що використовуються для реалізації функціоналу модуля «Викладач»

На рисунку 4.21 продемонстровано домашню сторінку «Викладач».

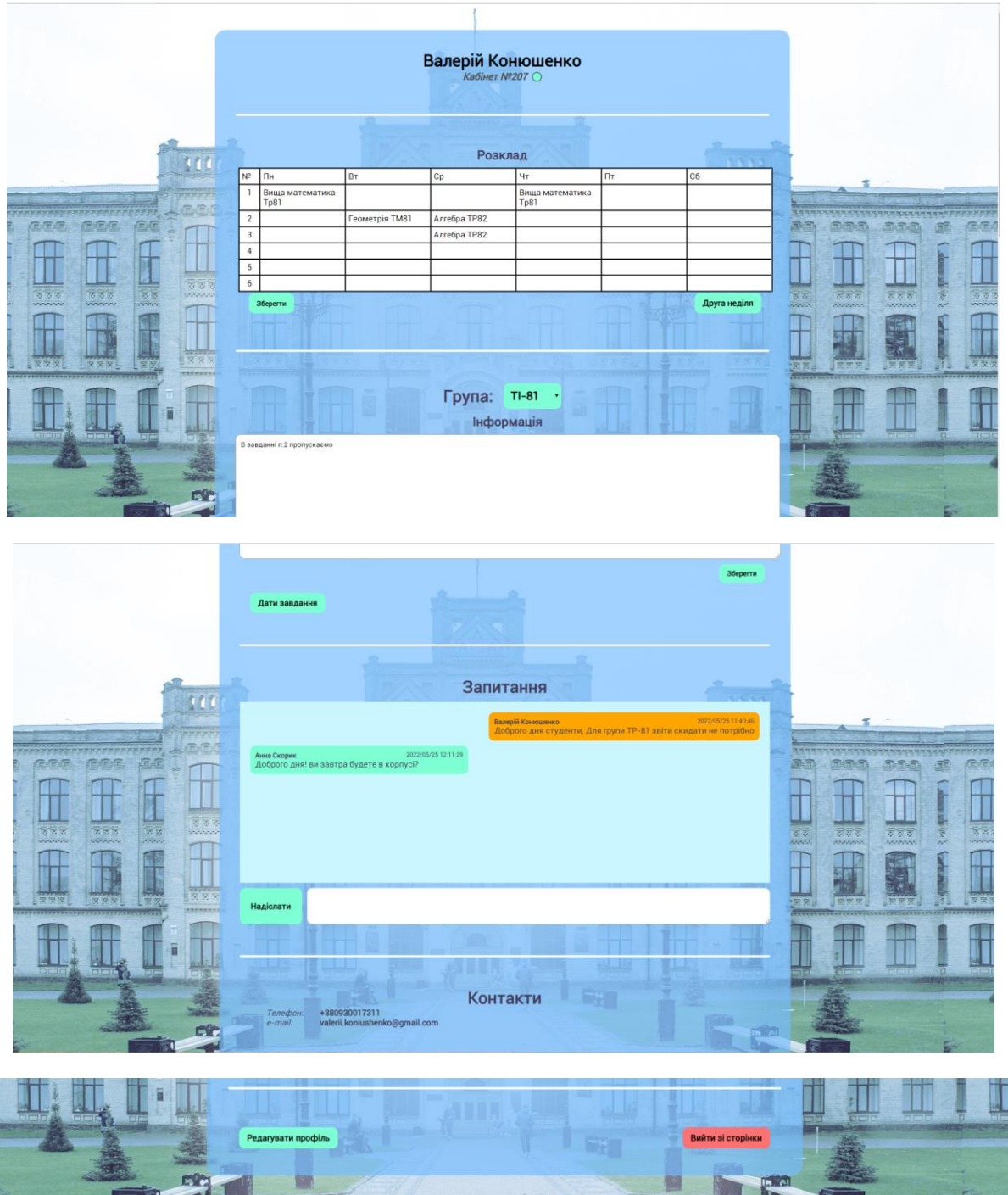


Рисунок 4.21 – Готова сторінка «Викладач»

На рисунку 4.22 продемонстровано сторінку «Дати завдання», що відкривається при натисканні відповідної кнопки.



Рисунок 4.22 – Вікно «Дати завдання»

Сторінка «Викладач» виглядає простою у розумінні для нового користувача, не залежно від його віку чи спеціалізації. Підтримуваний функціонал включає:

- редагування/перегляд розкладу (2 тижня)
- відмітка наявності викладача в кабінеті
- обирання назви групи з випадючого списку
- публікування інформації та дачі завдання для неї
- переписка зі всіма студентами у відкритому чаті
- редагування профілю
- вихід з акаунту.

## 4.4 Створення модуля «Студент» та його компонентів

На рисунку 4.23 продемонстровано архітектуру програмного коду з виділеними методами для реалізації функціоналу «Студент».

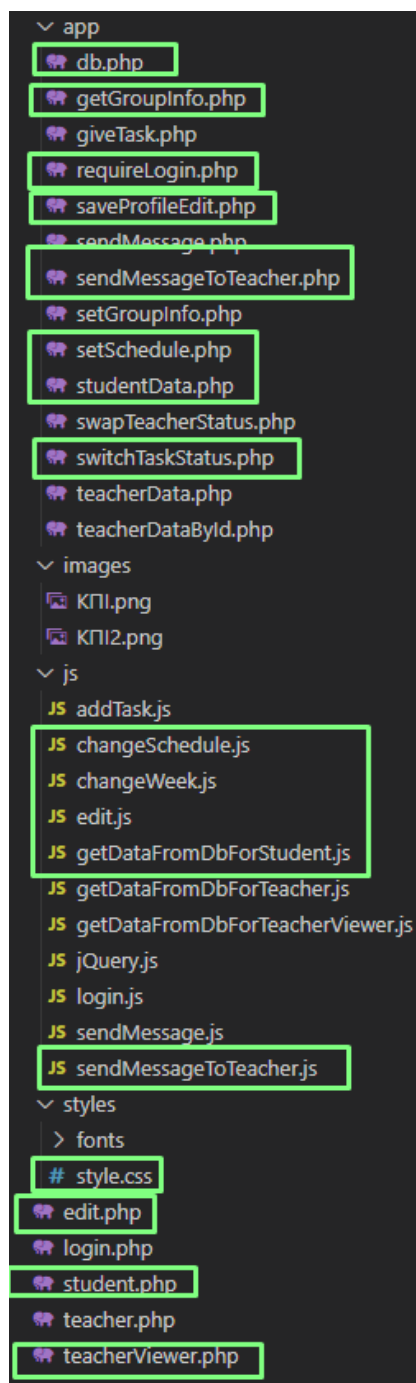


Рисунок 4.23 – Методи, що використовуються для реалізації функціоналу модуля «Студент»

На рисунку 4.24 продемонстровано домашню сторінку «Студент».

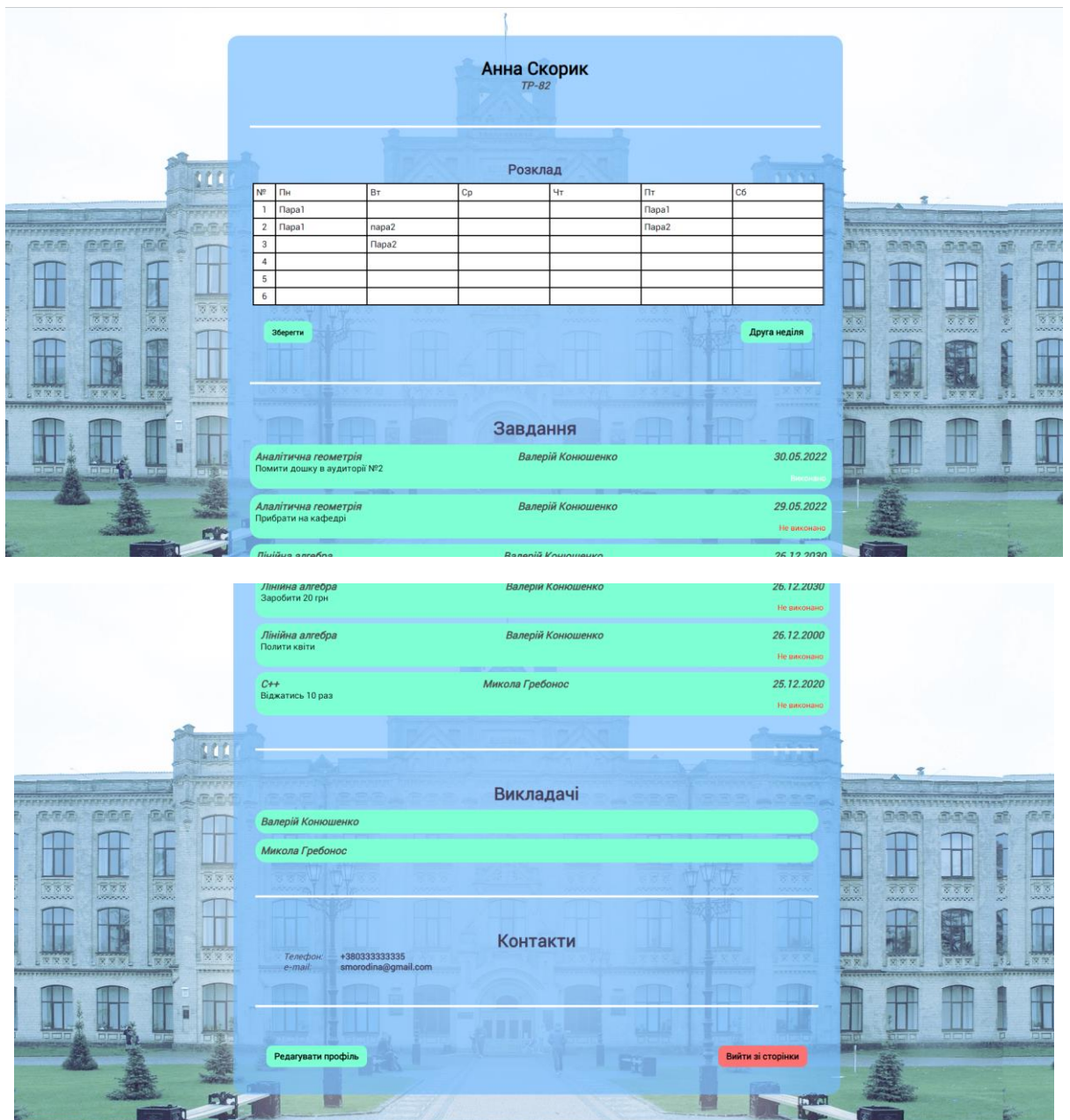


Рисунок 4.24 – Готова сторінка «Студент»

На рисунку 4.25 продемонстровано сторінку редагування профілю, що відкривається при натисканні відповідної кнопки.

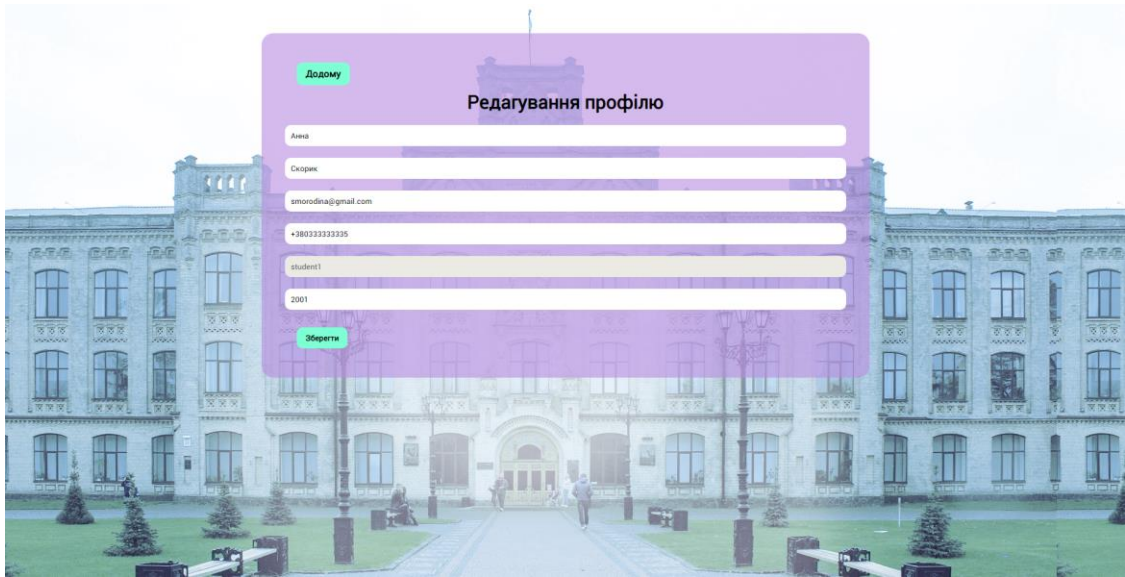


Рисунок 4.25 – Сторінка редагування профілю

Сторінка «Студент» включає:

- редагування/перегляд розкладу (2 тижня)
- Перегляд/відмічання виконаних/невиконаних завдань
- Перегляд всіх викладачів з можливістю переходу на їх сторінки
- редагування профілю
- вихід з акаунту.

Сторінка «Редагування профілю» включає:

- редагування полів інформації про користувача
- зберегти
- повернення на початкову сторінку.

## 4.5 Створення модуля «Перегляд викладача» та вікна «Логін»

На рисунку 4.26 продемонстровано архітектуру програмного коду з виділеними методами для реалізації функціоналу “Перегляд викладача” та “Логін”.

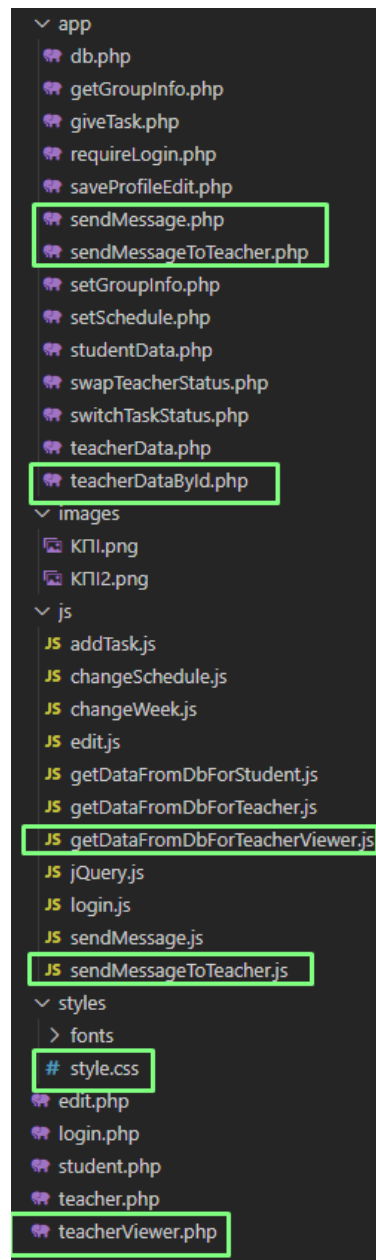


Рисунок 4.26 – Методи, що використовуються для реалізації функціоналу модуля «Перегляд викладача»

На рисунку 4.27 продемонстровано сторінку «Перегляд викладача».

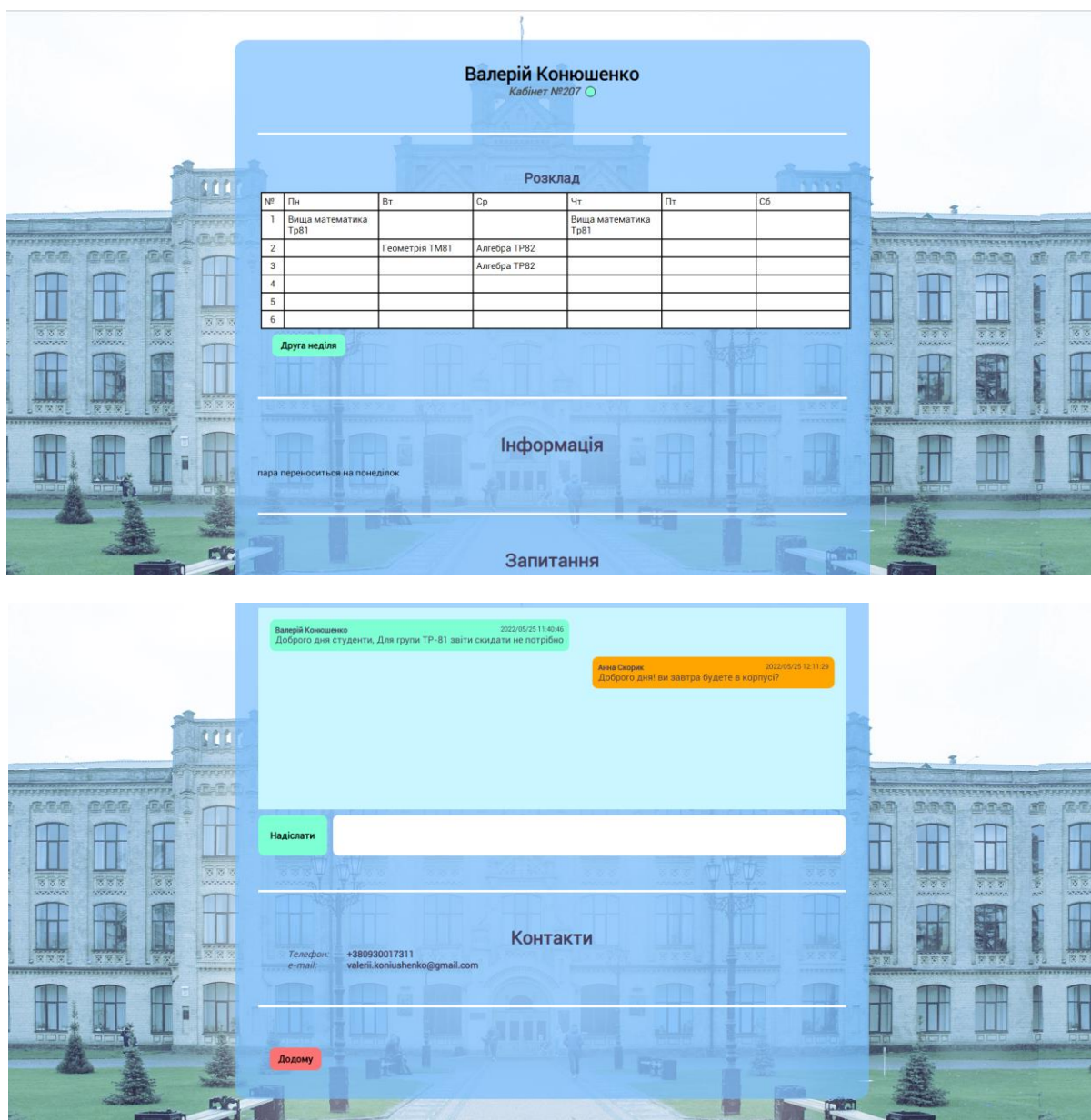


Рисунок 4.27 – Сторінка «Перегляд викладача»

На рисунку 4.28 продемонстровано сторінку «Логін».



Рисунок 4.28 – Сторінка «Логін»

Сторінка «Перегляд викладача» включає:

- перегляд розкладу (2 тижня)
- Перегляд інформації для групи авторизованого студента
- спілкування з викладачем у чаті
- перегляд контактів/ статусу чи в кабінеті викладач
- вихід на домашню сторінку.

Сторінка «Логін» включає:

- ввід логіну та паролю
- кнопку входу.

Система повідомляє користувача у разі невалідного логіну чи паролю при спробі зайти на ресурс.

## 5 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

Працювати з такою системою надзвичайно просто, оскільки вона є інтуїтивно зрозумілою для будь-якого користувача.

Дизайн спеціально продумувався без зайвих вікон, вкладок, кнопок і т.д.

Попри легкість у використанні, все ж є моменти на які варто звернути увагу. Перш за все це те, що зареєструватися власноруч в цю систему неможливо. Реєстрація студента/викладача відбувається лише один раз системним адміністратором. Це зроблено задля створення закритої мережі всередині університету. Також обмеження є на зміну даних профілю. Можна змінити всі поля окрім логіну. Логін також має змінювати виключно системний адміністратор за нагальної потреби. Це робиться для того, щоб акаунт неможливо було зламати і вкрасти в студента змінивши всі дані для входу.

## 6 НАПРЯМКИ РОЗВИТКУ ПРОЕКТА

Даний проект спрямований на повну заміну використання сторонніх ресурсів в навчальному процесі на базі НТУУ КПІ. «Система-довідник «Студент-викладач» кафедри» в майбутньому може позиціонувати себе як єдиний портал для зв'язку між викладачем та студентом, що безумовно буде хорошим кроком для систематизації та організації інформації. Однак звісно одним університетом дана система може не обмежуватись, та в подальшому, при низці вагомих допрацювань діяти на рівні країни, та покривати більшість ВНЗ країни.

Саме для цього розроблена система спроектована таким чином, щоб її можна було легко доповнювати новим обширним функціоналом, що гармонійно буде вести себе з вже існуючим.

### 6.1 Розвиток модуля «Викладач»

Модуль «Викладач» можна розвинути в даних напрямках:

- Перегляд заданих завдань та редагування їх;

Викладач мусить мати змогу переглядати завдання, що він задав студентам, редагувати та змінювати їх; видаляти.

- Завантаження необхідних матеріалів для виконання завдання, лекцій;

Для якісної освіти, студентам потрібні інформаційні джерела для засвоєння матеріалу та виконання лабораторних робіт. Оскільки даний ресурс цілком може стати єдиною платформою для навчання, то даний функціонал є необхідним.

- Перевірка зданих робіт студентами;

Аналогічно до попереднього підпункту. Ресурс повинен надавати якомога ширший спектр інформації для студента в одному місці.

- Оцінювання робіт та успішності студента;
- Проведення онлайн тестування з предмету для конкретної групи;

Дана функція дозволить завантажувати тестування для студентів певної групи. Система ж автоматично запустить його у вказаний час на вказану тривалість. По завершенню тестування система автоматично підрахує результат кожного студента та виставить оцінки до таблицю.

- Відповідь на повідомлення конкретного повідомлення з чату, та примусове видалення спаму з чату;

Розроблена демо-версія системи не передбачає саме відповідь на конкретне повідомлення з чату та функцію видалення повідомлення, у разі недопустимого контенту у ньому.

- Зв'язок зі службою підтримки.

При неполадках в системі, або іншим причинами, треба можливість з'єднання з службою підтримки (Системним Адміністратором) для узгодження проблем.

## **6.2 Розвиток модуля «Студент»**

Модуль «Студент» можна доповнювати наступним функціоналом:

- Завантаження зроблених студентом робіт на ресурс для оцінювання;

Для перевірки та оцінювання студента викладачем, йому необхідна офіційна підстава. Це і буде завантажені роботи на даний ресурс.

- Табелі з успішністю студента на його сторінці;

Зручно дивитися за своєю успішністю відразу та бачити цілу картину після перевірки кожного з завдань.

- Перегляд доданих джерел інформації до завдання відразу на сторінці студента;
- Зв'язок зі службою підтримки.

### 6.3 Створення модуля «Адміністратор»

Даний модуль є відсутній у системі. Системний Адміністратор керує БД системи за допомогою ресурсу PhpMyAdmin. Однак набагато зручніше для подальшої підтримки ресурсу – розробити модуль «Адміністратор» з найвищими правами доступу до системи.

Такий користувач повинен мати доступ до редагування всіх таблиць БД, оформленому в користувацькому інтерфейсу системи.

Також в нього мусить бути реалізована пошта, куди будуть надходити повідомлення з відміткою «Скарга», «Побажання» та «Зауваження». Ці листи будуть надсилати безпосередньо користувачі, при зверненні до служби підтримки.

Листи зі статусом «Скарга» - будуть найголовнішими в пріоритеті листів до розгляду. На вирішення яких, Адміністратор буде фокусувати переважну більшість свого робочого часу. «Скарга» - це некоректне поводження системи, яке перешкоджає користуванню функціоналом ресурсу.

«Зауваження» - це ідеї користувачів з приводу не доречності/не зручності тих чи інших функцій, що вже імплементовані в даний ресурс. Коли користувач надсилає лист з поміткою «Зауваження», він хоче поділитися своїми думками як можна покращити систему для зручності всіх користувачів. Саме таким чином може бути досягнутий результат справді конкурентоспроможного ресурсу з-поміж інших існуючих. Користувачі самі ж будуть навігувати напрям розвитку веб-порталу.

Листи зі статусом «Зауваження» - розглядаються в другу чергу.

Останній тип повідомлень – «Побажання». Цей тип дещо схожий на попередній, однак має низку відмінностей. Найголовніша відмінність – ця помітка позначає ідеї для ще не імплементованого функціоналу, який користувачі хотіли б бачити в цій системі. Ця опція також може скоригувати напрям розвитку порталу та можливо, навіть, відкрити зовсім нові методи підходу до розробок.

Саме така варіативність та гнучкість, прислухання до потреб молодого покоління дозволить розробити дійсно якісний продукт, що буде зручним та легким в освоєнні/використанні.

## **ВИСНОВКИ**

В сучасних реаліях тема даної роботи є дуже актуальна. З системним підходом, якісною реалізацією може стати дуже доречним проектом не лише на рівні кафедри чи університету, а і на Всеукраїнському рівні.

В майбутньому долучивши спонсорів, інвесторів, КПП та уряд можна збільшити ресурси та розвинути цю платформу як самостійний повноцінний портал. Адже наразі це лише демонстраційний варіант.

Попри це, є безліч ідей для покращення та розширення функціоналу, який буде дуже доречний в рамках даного проекту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. КПІ Розклад: веб-сайт.  
URL: <https://есampus.kpi.ua>.
2. КПІ Кампус: веб-сайт.  
URL: <https://есampus.kpi.ua>.
3. Google classroom: веб-сайт.  
URL: <https://classroom.google.com>.
4. Twitter: веб-сайт.  
URL: <https://twitter.com/>
5. Документація PHP: веб-сайт.  
URL: <https://www.php.net>
6. Tutorial Republic. *Ways of connecting to MySQL through PHP*: веб-сайт.  
URL: <https://www.tutorialrepublic.com/php-tutorial/php-mysql-connect.php>
7. Документація JavaScript: веб-сайт.  
URL: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>
8. Документація JQuery : веб-сайт.  
URL: <https://api.jquery.com/>
9. MDN. *Введення в AJAX*: веб-сайт.  
URL: <https://developer.mozilla.org/ru/docs/Web/Guide/AJAX>.
10. How to return AJAX Response : веб-сайт.  
URL: <https://dzone.com/articles/how-to-return-ajax-response-from-asynchronous-java>
11. Дронов В.О. HTML 5, CSS 3 та Web 2.0. Розробка сучасних Web-сайтів. *Основні принципи створення Web-сторінок. Мова HTML.*: довідник. Санкт-Петербург: БХВ-Петербург, 2011. С.25-32.  
URL:  
<https://lib.brsu.by/sites/default/files/books/%d0%94%d1%80%d0%be%d0%bd%d0%be%d0%b2.%d0%92.->

[.HTML .5.CSS .3.%d0%b8.Web%202.0.%d0%a0%d0%b0%d0%b7%d1%80%  
d0%b0%d0%b1%d0%be%d1%82%d0%ba%d0%b0.%d1%81%d0%be%d0%b2  
%d1%80%d0%b5%d0%bc%d0%b5%d0%bd%d0%bd%d1%8b%d1%85.Web-  
%d1%81%d0%b0%d0%b9%d1%82%d0%be%d0%b2.%d0%91%d0%a5%d0%2  
=  
%d0%9f%d0%b5%d1%82%d0%b5%d1%80%d0%b1%d1%83%d1%80%d0%b3  
.2011\\_0.pdf](#)

12. Довідник по HTML: веб-сайт.

URL:<http://htmlbook.ru/html>

13. Wikipedia. CSS: веб-сторінка.

URL: <https://ru.wikipedia.org/wiki/CSS>

# ДОДАТОК А

Система-довідник «Студент-викладач» кафедри

**Текст файлу „db.php”**

УКР.НТУУ”КПІ”\_ТЕФ\_АПЕПС\_ ТР82376\_22Б\_12-1

Листів 7

2022

**//Відкриття з'єднання (\$conn) з БД**

```
function openDb()
{
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "anna";
    $conn = new mysqli($servername, $username, $password, $dbname);
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    return $conn;
}
```

**//Перевірка при логіну чи юзер існує в БД**

```
function isUserExists( $conn, $login, $password )
{
    $sql = "SELECT * FROM users WHERE login='$login' AND password='$password'";
    $result = $conn->query($sql);
    return $result->num_rows > 0;
}
```

**//Взяти всі дані про користувача**

```
function getUserData( $conn, $login, $password )
{
    $sql = "SELECT * FROM users WHERE login='$login' AND password='$password'";
    $result = $conn->query($sql);
    return $result->fetch_assoc();
}
```

**//Взяти всі дані про користувача по ід**

```
function getUserDataById( $conn, $id )
{
    $sql = "SELECT * FROM users WHERE id=$id";
    $result = $conn->query($sql);
```

```
return $result->fetch_assoc();  
}
```

### **//Отримати номер кабінету**

```
function getTeacherCabinet( $conn, $userId )  
{  
    $sql = "SELECT * FROM `users` JOIN teachers ON user_id=$userId WHERE users.id=$userId";  
    $result = $conn->query($sql);  
    $row = $result->fetch_assoc();  
    return $row["cabinet"];  
}
```

### **//Перевірка чи викладач в кабінеті**

```
function isTeacherOnline( $conn, $userId )  
{  
    $sql = "SELECT * FROM `users` JOIN teachers ON user_id=$userId WHERE users.id=$userId";  
    $result = $conn->query($sql);  
    $row = $result->fetch_assoc();  
    return $row["is_online"];  
}
```

### **//Отримати список всіх груп**

```
function getAllGroups( $conn )  
{  
    $sql = "SELECT * FROM `groups` ORDER BY `groups`.`name` ASC";  
    $result = $conn->query($sql);  
    $out = array();  
    while( $row = $result->fetch_assoc() )  
        $out[] = $row["name"];  
    return $out;  
}
```

### **//Отримати ід групи по назві групи**

```
function getGroupIdByName( $conn, $groupName )  
{  
    $sql = "SELECT * FROM `groups` WHERE groups.name='$groupName'";  
    $result = $conn->query($sql);
```

```

$row = $result->fetch_assoc();

return $row["id"];
}

```

#### //Отримати інформацію для групи

```

function getGroupInfo( $conn, $groupName, $teacherId )
{
    $groupId = getGroupIdByName( $conn, $groupName );

    $sql = "SELECT * FROM groupinfo WHERE groupinfo.group_id=$groupId AND groupinfo.teacher_id=$teacherId";

    $result = $conn->query($sql);

    $row = $result->fetch_assoc();

    return $row["description"];
}

```

#### //Отримати інформацію для групи по її ід

```

function getGroupInfoByGroupId( $conn, $groupId, $teacherId )
{
    $sql = "SELECT * FROM groupinfo WHERE groupinfo.group_id=$groupId AND groupinfo.teacher_id=$teacherId";

    $result = $conn->query($sql);

    $row = $result->fetch_assoc();

    return $row["description"];
}

```

#### //Встановити інформацію для групи

```

function setGroupInfo( $conn, $groupName, $teacherId, $data )
{
    $groupId = getGroupIdByName( $conn, $groupName );

    $sql = "DELETE FROM `groupinfo` WHERE teacher_id=$teacherId AND group_id=$groupId";

    $conn->query($sql);

    $sql = "INSERT INTO `groupinfo`(`group_id`, `teacher_id`, `description`) VALUES ($groupId, $teacherId, '$data')";

    $conn->query($sql);
}

```

#### //Змінити статус наявності викладача в кабінеті

```

function swapTeacherStatus( $conn, $teacherId )
{
    $is = isTeacherOnline( $conn, $teacherId );
}

```

```

    $sql = "UPDATE `teachers` SET `is_online` = '".($is == '1' ? '0' : '1')." WHERE `user_id` = $teacherId";

    $conn->query($sql);

}

```

### //Встановити розклад

```

function setSchedule( $conn, $teacherId, $data )
{
    $sql = "DELETE FROM `schedule` WHERE teacher_id=$teacherId";

    $conn->query($sql);

    $sql = "INSERT INTO `schedule`(`description`, `teacher_id`) VALUES ('$data', $teacherId)";

    $conn->query($sql);

}

```

### //Отримати розклад

```

function getSchedule( $conn, $teacherId )
{
    $sql = "SELECT * FROM schedule WHERE teacher_id=$teacherId";

    $result = $conn->query($sql);

    return $result->fetch_assoc()["description"];

}

```

### //Отримати дані чату

```

function getChat( $conn, $teacherId )
{
    $sql = "

SELECT

    income_user_id,

    message,

    time,

    out_user.name AS out_name,

    out_user.lastname AS out_lastname,

    in_user.name AS in_name,

    in_user.lastname AS in_lastname

FROM

    (

        SELECT

```

```

        *
    FROM chat
    WHERE
        outcome_user_id=$teacherId OR
        income_user_id=$teacherId
    ) AS tmp
INNER JOIN
    users AS out_user
ON
    tmp.outcome_user_id=out_user.id
LEFT JOIN
    users AS in_user
ON
    tmp.income_user_id=in_user.id
";
$result = $conn->query($sql);
$chat = array();
while ( $row = $result->fetch_assoc() )
{
    $message = array();
    $message["message"] = $row["message"];
    $message["isIncome"] = $row["income_user_id"] == "$teacherId" ? 1 : 0;
    $message["out"] = $row["out_name"]." ".$row["out_lastname"];
    $message["in"] = $row["in_name"]." ".$row["in_lastname"];
    $message["time"] = $row["time"];
    $chat[] = $message;
}
return $chat;
}

```

**//Надіслати повідомлення в чат від викладача**

```

function sendMessageFromTeacher( $conn, $teacherId, $message, $time )
{
    $sql = "INSERT INTO `chat`(`message`, `outcome_user_id`, `income_user_id`, `time`) VALUES ('$message', $teacherId, 0, '$time')";
    echo $sql;
    $conn->query($sql);
}

```

```
}
```

### //Дати завдання студенту

```
function giveTask( $conn, $groupName, $userId, $description, $deadline, $subject )  
  
{  
  
    $groupId = getGroupIdByName( $conn, $groupName );  
  
    $sql = "INSERT INTO `tasks`(`teacher_id`, `deadline`, `description`, `group_id`, `subject`) VALUES ($userId, '$deadline', '$description',  
$groupId, '$subject')";  
  
    $conn->query($sql);  
  
}
```

### //Отримати всі завдання для групи

```
function getAllTasksForGroup( $conn, $groupName )  
  
{  
  
    $groupId = getGroupIdByName( $conn, $groupName );  
  
    $sql = "SELECT tasks.id AS task_id, group_id, subject, description, deadline, users.name, users.lastname, is_done FROM `tasks`  
  
INNER JOIN users ON teacher_id=users.id  
  
WHERE group_id=$groupId  
  
ORDER BY deadline DESC";  
  
    $result = $conn->query($sql);  
  
    $out = array();  
  
    while ( $row = $result->fetch_assoc() )  
  
    {  
  
        $task = array();  
  
        $task["teacher_name"] = $row["name"]." ".$row["lastname"];  
  
        $task["subject"] = $row['subject'];  
  
        $task["description"] = $row['description'];  
  
        $task["deadline"] = $row['deadline'];  
  
        $task["is_done"] = $row['is_done'];  
  
        $task["task_id"] = $row['task_id'];  
  
        $out[] = $task;  
  
    }  
  
    return $out;  
  
}
```

### //Отримати список всіх викладачів

```
function getTeachersList( $conn )
```

```

{
    $sql = "SELECT name, lastname, users.id FROM `teachers` INNER JOIN users ON teachers.user_id=users.id";
    $result = $conn->query($sql);
    $out = array();
    while ( $row = $result->fetch_assoc() )
    {
        $user = array();
        $user["id"] = $row["id"];
        $user["name"] = $row["name"]." ".$row["lastname"];
        $out[] = $user;
    }
    return $out;
}

```

**//Отримати номер групи**

```

function getGroup( $conn, $userId )
{
    $sql = "SELECT groups.name FROM students
    INNER JOIN users ON students.user_id=users.id
    INNER JOIN groups ON students.group_id=groups.id
    WHERE users.id=$userId";
    return $conn->query($sql)->fetch_assoc()["name"];
}

```

## ДОДАТОК Б

Система-довідник «Студент-викладач» кафедри

Текст файлів „**login.js**”, „**edit.js**” та „**addTask.js**”

УКР.НТУУ”КПІ”\_ТЕФ\_АПЕПС\_ТР82376\_22Б\_12-1

Листів 3

2022

### // login.js

//За допомогою аїах отримуємо введені дані про користувача та при вірних даних переадресуємося на наступну сторінку

```
$.ajax({
    method: "POST",
    url: "/app/requireLogin.php",
    data: {
        login: $('form input[name="login"]').val(),
        password: $('form input[name="password"]').val()
    }
}).done(function( msg ) {
    let json = JSON.parse(msg)
    let i = $('form i');
    if ( json["status"] == 'ok' )
    {
        document.cookie = "login = " + $('form input[name="login"]').val();
        document.cookie = "password = " + $('form input[name="password"]').val();
        document.cookie = "redirectTo = " + json["redirectTo"];
        i.text('Заходимо...').css('color', "green");
        window.location.replace("/" + json["redirectTo"]);
    }
    else
        i.text('Користувача не знайдено').css('color', "red");
});
});
```

### // edit.js

//функція діставання куки

```
function getCookie(name) {
    const value = `; ${document.cookie}`;
    const parts = value.split(`; ${name}=`);
    if (parts.length === 2)
        return parts.pop().split(';').shift();
}
$('a.button').attr('href', '/' + getCookie('redirectTo'));
```

// за отриманими куки визначаємо дані профілю для редагування та в асинхронному режимі працюємо з ними

```
if ( getCookie('redirectTo') == 'teacher.php' )
{
$.ajax({
method: "POST",
url: "/app/teacherData.php"
}).done((data)=>{
data = JSON.parse(data);
$('#name').val( data["name"] );
$('#lastname').val( data["lastname"] );
$('#email').val( data["email"] );
$('#phone').val( data["phone"] );
$('#password').val( getCookie("password") );
$('#login-input').val( getCookie("login") );
});
}
else
{
$.ajax({
method: "POST",
url: "/app/studentData.php"
}).done((data)=>{
data = JSON.parse(data);
$('#name').val( data["name"] );
$('#lastname').val( data["lastname"] );
$('#email').val( data["email"] );
$('#phone').val( data["phone"] );
$('#password').val( getCookie("password") );
$('#login-input').val( getCookie("login") );
});
}

$("form").submit((event)=>{
event.preventDefault();
```

```

// асинхронно зберігаємо змінені дані
$.ajax({
    method: "POST",
    url: "/app/saveProfileEdit.php",
    data: {
        password: $("#password").val(),
        email: $("#email").val(),
        phone: $("#phone").val(),
        lastname: $("#lastname").val(),
        name: $("#name").val(),
    }
});

});

// addTask.js
// асинхронно заповнюються та зберігаються дані нового завдання
$("#giveTaskButton").click(=>{
    $.ajax({
        method: "POST",
        url: "/app/giveTask.php",
        data: {
            groupName: $('#group select').find(":selected").text(),
            deadline: $("#tutor #giveTaskBlock input:first-of-type").val(),
            subject: $("#tutor #giveTaskBlock input:last-of-type").val(),
            description: $("#tutor #giveTaskBlock textarea").val()
        }
    });
});

```

## ДОДАТОК В

Система-довідник «Студент-викладач» кафедри

**Текст файлів „edit.php”, „login.php”, „student.php”, „teacher.php” та  
„teacherViever.php”**

УКР.НТУУ”КПІ”\_ТЕФ\_АПЕПС\_ТР82376\_22Б\_12-1

Листів 9

2022

## //edit.php

//візуальне представлення вікна редагування

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Редагування профілю</title>
  <link rel="stylesheet" href="styles/style.css">
  <script src="js/jquery.js"></script>
  <style> body{
    background: url(images/КПІ.png) fixed;
  }</style>
</head>
<body>
  <div class="wrapper" id="edit">
    <a class="button">Додому</a>
    <form>
      <h1>Редагування профілю</h1>
      <input type="text" placeholder="Ім'я" id="name">
      <input type="text" placeholder="Фамілія" id="lastname">
      <input type="text" placeholder="e-mail" id="email">
      <input type="text" placeholder="Номер телефону" id="phone">
      <input type="text" id="login-input" disabled>
      <input type="text" placeholder="Пароль" id="password">
      <button class="button" id="save-button">Зберегти</button>
    </form>
  </div>
  <script src="js/edit.js"></script>
</body>
</html>
```

## // login.php

//візуальне представлення вікна логіну

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Увійти</title>
  <link rel="stylesheet" href="styles/style.css">
  <script src="js/jquery.js"></script>
  <style> body{
    background: url(images/КПІ.png) fixed;
  }</style>
</head>
<body>
  <div class="wrapper" id="login">
    <form>
      <input type="text" placeholder="Лорін" name="login">
      <input type="password" placeholder="Пароль" name="password">
      <i></i>
      <button class="button">Увійти</button>
    </form>
  </div>
  <script src="js/login.js"></script>
</body>
</html>

```

## // student.php

//візуальне представлення вікна студента

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8"/>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<link rel="stylesheet" href="styles/style.css">
<title>Студент</title>
<script src="js/jquery.js"></script>
<style> body{
    background: url(images/КПІ.png) fixed;
}</style>
</head>
<body>
<div class="wrapper" id="tutor">
    <header>
        <h1>...</h1>
        <h3>...</h3>
    </header>
<div class="line"></div>
<main>
    <h2>Розклад</h2>
    <form action="" method="POST" id="scheduleForm">
        <div class="schedule schedule1">
            //Вивід розкладу
        </div>
        <div id="fakeSchedule"></div>
        <div class="buttons">
            <button class="button" id="save-schedule">Зберегти</button>
            <div class="button" id="btnChangeWeek">Друга неділя</div>
        </div>
    </form>
</main>
<div class="line"></div>
<div id="tasks">
    <h1>Завдання</h1>
</div>
<div class="line"></div>
<div id="teachers">
    <h1>Викладачі</h1>
    <ul id="teachersList">

```

```

    </ul>
</div>
<div class="line"></div>
<div id="contacts">
    <h1>Контакти</h1>
    <div>
        <section>
            <p>
                <i>Телефон:</i>
                <b id="phoneNumber">...</b>
            </p>
        </section>
        <section>
            <p>
                <i>e-mail:</i>
                <b id="email">...</b>
            </p>
        </section>
    </div>
</div>
<div class="line"></div>
<footer>
    <a href="edit.php" class="button">Редагувати профіль</a>
    <a href="login.php" class="button">Вийти зі сторінки</a>
</footer>
</div>
<script src="js/changeWeek.js"></script>
<script src="js/changeSchedule.js"></script>
<script src="js/getDataFromDbForStudent.js"></script>
</body>
</html>

```

## //teacher.php

// візуальне представлення вікна викладача

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles/style.css">
  <title>Студент-Викладач</title>
  <script src="js/jquery.js"></script>
  <style> body{
    background: url(images/КПІ.png) fixed;
  }</style>
</head>
<body>
  <div class="wrapper" id="tutor">
    <header>
      <h1>...</h1>
      <h3>
        Кабінет №<b>...</b>
        <form>
          <input type="checkbox">
          <span class="checkbox"></span>
        </form>
      </h3>
    </header>
    <div class="line"></div>
    <main>
      <h2>Розклад</h2>
      <form action="" method="POST" id="scheduleForm">
        <div class="schedule schedule1">
//Вивід розклада
        </div>
        <div id="fakeSchedule"></div>
        <div class="buttons">
          <button class="button" id="save-schedule">Зберегти</button>
          <div class="button" id="btnChangeWeek">Друга неділя</div>

```

```

    </div>

</form>

</main>

<div class="line"></div>

<div id="group">

    <h1>Група:

        <select id="selectGroup">

            </select>

        </h1>

        <form class="groupInfo" action="" method="GET">

            <h2>Інформація</h2>

            <textarea id="groupData"></textarea>

            <button class="button">Зберегти</button>

        </form>

    </div>

<div class="button" onclick="giveTask()">Дати завдання</div>

<div id="giveTaskBlock">

    <h2>Дати завдання</h2>

    <input type="text" placeholder="Кінцева дата">

    <input type="text" placeholder="Предмет">

    <textarea placeholder="Завдання"></textarea>

    <p>Для групи: <span id="forGroup"></span></p>

    <button class="button" id="giveTaskButton">Підтвердити</button>

</div>

<div class="line"></div>

<div id="questions">

    <h1>Запитання</h1>

    <div id="chat">

        <div id="chatBody"></div>

        <div class="writeMessage">

            <div class="button" id="sendMessage">Надіслати</div>

            <textarea id="messageBody"></textarea>

            <input type="text" id="tutorName" style="display: none">

        </div>

    </div>

</div>

```

```

</div>
<div class="line"></div>
<div id="contacts">
  <h1>Контакти</h1>
  <div>
    <section>
      <p>
        <i>Телефон:</i>
        <b id="phoneNumber">...</b>
      </p>
    </section>
    <section>
      <p>
        <i>e-mail:</i>
        <b id="email">...</b>
      </p>
    </section>
  </div>
</div>
<div class="line"></div>
<footer>
  <a href="edit.php" class="button">Редагувати профіль</a>
  <a href="login.php" class="button">Вийти зі сторінки</a>
</footer>
<div id="background"></div>
</div>
<script src="js/changeWeek.js"></script>
<script src="js/changeSchedule.js"></script>
<script src="js/sendMessage.js"></script>
<script src="js/getDataFromDbForTeacher.js"></script>
<script src="js/addTask.js"></script>
</body>
</html>

```

**//teacherViever.php**

## // візуальне представлення вікна перегляду викладача

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8"/>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles/style.css">
  <title>Перегляд Викладача</title>
  <script src="js/jquery.js"></script>
  <style> body{
    background: url(images/КПІ.png) fixed;
  }</style>
</head>
<body>
  <div class="wrapper" id="tutor">
    <header>
      <h1>...</h1>
      <h3>
        Кабінет №<b>...</b>
        <form>
          <input type="checkbox">
          <span class="checkbox"></span>
        </form>
      </h3>
    </header>
    <div class="line"></div>
    <main>
      <h2>Розклад</h2>
      //вивід розклада
      <div id="fakeSchedule"></div>
      <div class="buttons">
        <div class="button" id="btnChangeWeek">Друга неділя</div>
      </div>
    </form>
  </div>
</body>
```

```

</main>

<div class="line"></div>

<div id="info">

  <h1>Інформація</h1>

  <p id="info-box"></p>

</div>

<div class="line"></div>

<div id="questions">

  <h1>Запитання</h1>

  <div id="chat">

    <div id="chatBody"></div>

    <div class="writeMessage">

      <div class="button" id="sendMessage">Надіслати</div>

      <textarea id="messageBody"></textarea>

      <input type="text" id="tutorName" style="display: none">

    </div>

  </div>

</div>

<div class="line"></div>

<div id="contacts">

  <h1>Контакти</h1>

  <div>

    <section>

      <p>

        <i>Телефон:</i>

        <b id="phoneNumber">...</b>

      </p>

    </section>

    <section>

      <p>

        <i>e-mail:</i>

        <b id="email">...</b>

      </p>

    </section>

```

```
</div>
</div>
<div class="line"></div>
<footer>
<a class="button" href="/student.php">
    Додому
</a>
</footer>
</div>
<script src="js/changeWeek.js"></script>
<script src="js/getDataFromDbForTeacherViewer.js"></script>
<script src="js/sendMessageToTeacher.js"></script>
</body>
</html>
```

## ДОДАТОК Г

Система-довідник «Студент-викладач» кафедри

**Опис системи-довідника „ Студент-викладач.doc”**

УКР.НТУУ”КП”\_ТЕФ\_АПЕПС\_ТР82376\_22Б\_12-1

Листів 1

## **АНОТАЦІЯ**

Цей додаток призначений для розгляду таких питань як цільове призначення системи, правила користування ресурсом, вхідні та вихідні дані для користування ресурсом.

## **ЗМІСТ**

1. Загальні відомості.....	3
2. Призначення ресурсу.....	3
3. Опис логічної структури.....	3
4. Вхідні дані.....	5
5. Вихідні дані.....	5

## **1. ЗАГАЛЬНІ ВІДОМОСТІ**

Даний ресурс «Студент-викладач» реалізований засобами мов програмування Php, JavaScript, CSS, бібліотеки JQuery, запитами SQL та мовою гіперсилкової розмітки HTML. Використовувалось середовище Visual Studio Code. Операційна система Windows 10.

## **2. ПРИЗНАЧЕННЯ РЕСУРСУ**

Використовуючи даний ресурс студент має зручний та простий доступ до необхідної інформації щодо навчального процесу та зв'язку з викладачами. Викладач в свою чергу має можливість допомогти студентові та публічно оприлюднювати всю необхідну інформацію та задавати завдання для студентів.

## **3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ**

Код даного ресурсу складається з таких файлів:

Серверна частина:

Db.php – файл з основними запитам до БД;

getGroupInfo.php – отримання інформації про групу;

giveTask.php – дати завдання;

requireLogin.php – логін користувача;

saveProfileEdit.php – зберегти змінені дані;

sendMessage.php – відправлення повідомлення;

sendMessageToTeacher.php – відправлення повідомлення конкретному викладачу;

setGroupinfo.php – встановити інформацію для групи;

setSchedule.php – встановити розклад;

studentData.php – дані про студента;

swapteacherStatus.php – перемикання статусу наявності викладача;

swithtoStatus.php – статус завдання виконано/не виконано;

teacherData.php – дані про викладача;

teacherDataById.php – дані про викладача за ід;

Частина передачі даних серверу/взаємодія з клієнтом:

addtask.js – додавання завдання;

changeSchedule.js – зміна розкладу;

changeWeek.js – зміна тижня;

edit.js – взаємодія на сторінці редагування профілю;

getDatafromDbForStudent.js – отримання даних з БД для студента;

getDatafromDbForTeacher.js – отримання даних з БД для викладача;

getDatafromDbForTeacherViewer.js – отримання даних з БД для сторінки перегляду викладача;

login.js — взаємодія на сторінці логіну;

sendMessage.js – формування та надсилання повідомлення;

sendMessageToTeacher.js – формування та надсилання повідомлення конкретному викладачу;

Стилі візуалізації:

style.css

Візуальна частина та формування всіх контейнерів/елементів на екрані:

edit.php

login.php  
student.php  
teacher.php  
teacherViewer.php

#### **4. ВХІДНІ ДАНІ**

Вхідними даними є дані про зареєстрованого користувача в системі, букви латинського алфавіту та кирилиці, арабські цифри, розділові знаки та спеціальні символи.

#### **5. ВИХІДНІ ДАНІ**

Вихідними даними є результат, що отримує користувач на екрані відповідно до його дій в системі.