

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

ДО ЗАХИСТУ ДОПУЩЕНО

В.о. завідувача кафедри

Олександр КОВАЛЬ

« _____ » _____ 2023р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інженерія програмного забезпечення
інтелектуальних кібер-фізичних систем і веб-технологій»

спеціальності 121 Інженерія програмного забезпечення

на тему: «Програмний застосунок для вибору вибіркових дисциплін»

Виконала:

студентка IV курсу, групи ТВ-391

Пашкевич Анастасія Сергіївна

Керівник:

доцент, к. т. н., Діденко Олексій Олександрович

Консультант:

Рецензент:

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

Студентка

Київ – 2023

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

Навчально-науковий інститут атомної та теплової енергетики
Кафедра інженерії програмного забезпечення в енергетиці
Рівень вищої освіти перший (бакалаврський)
Спеціальність 121 Інженерія програмного забезпечення
Освітньо-професійна програма «Інженерія програмного забезпечення
інтелектуальних кібер-фізичних систем і веб-технологій»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр КОВАЛЬ

(підпис)

« ____ » _____ 2023р.

ЗАВДАННЯ

на дипломну роботу студенту

Пашкевич Анастасії Сергіївні

(прізвище, ім'я, по батькові)

1. Тема роботи: “Програмний застосунок для вибору вибіркового дисциплін”

керівник роботи: доцент, к. т. н., Діденко Олексій Олександрович

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

затверджені наказом по університету від “ ____ ” _____ 2023 року № _____

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи: фреймворк Angular, мова програмування TypeScript, середовище розробки WebStorm

4. Зміст (дипломної роботи) пояснювальної записки (перелік завдань, які потрібно розробити: розробити додаток для вибору вибіркового дисциплін.

5. Перелік ілюстративного матеріалу: Використані програмні засоби, Інтерфейс авторизації та реєстрації, інтерфейс головної сторінки, інтерфейс вибору вибіркового дисциплін, інтерфейс керування вибілковими дисциплінами

6. Дата видачі завдання «17» квітня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Строки виконання етапів роботи	Примітка
1	Отримання завдання	17.04.23	виконано
2	Дослідження предметної області	17.04.23-20.04.23	виконано
3	Постановка вимог до проєктування системи	17.04.23-20.04.23	виконано
4	Дослідження існуючих рішень	17.04.23-20.04.23	виконано
5	Розробка програмного продукту	20.04.23-01.05.23	виконано
6	Тестування	01.05.23-05.05.23	виконано
7	Захист програмного продукту	19.05.23	виконано
8	Оформлення дипломної роботи	20.05.23-25.05.23	виконано
9	Передзахист	05.06.23-11.06.23	виконано
10	Захист	19.06.23-25.06.23	виконано

Студентка



(підпис)

Пашкевич Анастасія

(ім'я, прізвище)

Керівник роботи



(підпис)

Діденко Олексій

(ім'я, прізвище)

РЕФЕРАТ

Структура та обсяг дипломної роботи. Робота містить 53 сторінки, 48 рисунків, 1 додаток та 10 посилань.

Метою роботи є спрощення процесу вибору вибіркових дисциплін і надання студентам зручного інструменту для цього.

Для досягнення поставленої мети виконано такі завдання:

- проаналізовано існуючі підходи та додатки для вибору вибіркових дисциплін;
- вивчено наукові джерела та досвід дослідників у галузі освіти та розробки додатків;
- розроблено зручний інтерфейс.

Розроблено систему, яка надає студентам можливість переглянути доступні вибіркові дисципліни, ознайомитися з їх описом та вимогами. Студенти можуть здійснити осмислений вибір, враховуючи свої інтереси, навички та кар'єрні цілі. Додаток пропонує зручний інтерфейс, який дозволяє студентам легко користуватися додатком, швидко знаходити потрібну інформацію та здійснювати вибір вибіркових дисциплін.

Практичне значення одержаних результатів полягає в отриманні моделі та методів для здійснення класифікації та вибору вибіркових дисциплін. Проведено огляд існуючих рішень та аналіз конкурентів. Реалізований базовий інтерфейс демонструє функціональні можливості програмного застосунку.

Ключові слова: додаток, вибіркові дисципліни, студенти.

ABSTRACT

Structure and scope of the work. The work contains 53 pages, 48 figures, 1 appendix and 10 references.

The purpose of the work is to simplify the process of choosing elective subjects and to provide students with a convenient tool for this.

To achieve the set goal, the following tasks were completed:

- existing approaches and applications for choosing optional disciplines were analyzed;
- national sources and experience of researchers in the field of education and application development were studied;
- a convenient interface has been developed.

A system has been developed that provides students with the opportunity to review available elective courses, familiarize themselves with their description and requirements. Students can make informed choices based on their interests, skills and career goals. The app offers a user-friendly interface that allows students to easily use the app, quickly find the information they need, and choose electives.

The practical significance of the obtained results lies in obtaining a model and methods for classification and selection of selective disciplines. An overview of existing solutions and analysis of competitors was conducted. The implemented basic interface demonstrates the functionality of the software application.

Keywords: application, optional subjects, students.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП	8
1 ПОСТАНОВКА ЗАДАЧІ ВИБОРУ ВИБІРКОВИХ ДИСЦИПЛІН	10
Висновки до розділу 1	11
2 ОГЛЯД ПОДІБНИХ ІСНУЮЧИХ СИСТЕМ.....	12
2.1 Модуль “Вибір дисциплін” системи Електронний кампус	12
2.2 Модуль “Вибір дисциплін” системи tu.kpi.ua	16
Висновки до розділу 2	21
3 ЗАСОБИ РОЗРОБКИ СИСТЕМИ	22
3.1 Середовище розробки WebStorm	24
3.2 Налаштування робочого оточення	25
3.2.1 Встановлення Node.js	25
3.2.2 Встановлення Angular CLI	27
3.2.3 Створення нового проекту	27
3.3 Фреймворк Angular	29
3.3.1 Компоненти	29
3.3.2 Модулі	30
3.3.3 Форми.....	31
3.3.4 Сервіси	31
3.3.5 Директиви	31
3.3.6 Переваги Angular.....	32
3.4 Бібліотека PrimeNg	33
3.5 База даних Firebase.....	34
Висновки до розділу 3	35
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ.....	37
4.1 Модель бази даних	37

Висновки до розділу 4	40
5 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ.....	41
5.1 Системні вимоги.....	41
5.2 Реєстрація та вхід у систему	41
5.3 Головна сторінка програмного продукту	43
5.4 Приклади робота користувача з програмним продуктом	43
Висновки до розділу 5	51
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТОК А.Текст програми.....	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – Application programming interface

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

HTTP – HyperText Transfer Protocol

URL – Uniform Resource Locator

IDE – Integrated Development Environment

CLI – Command Line Interface

SPA – Single-page application

DOM – Document Object Model

JSON – JavaScript Object Notation

ВСТУП

Сучасне освітнє середовище вимагає від студентів активного підходу до вибору своєї навчальної програми, зокрема вибіркового дисциплін, які дозволяють їм розширити свої знання та спеціалізуватися у певних галузях. Однак, процес вибору вибіркового дисциплін часто стає складним і заплутаним для студентів, оскільки вони зіштовхуються з багатьма факторами, такими як обсяг інформації, різноманітність дисциплін, вимоги та зміст курсів.

У зв'язку з цим, ця дипломна робота присвячена розробці додатку для вибору вибіркового дисциплін, який має на меті спростити і полегшити цей процес для студентів. Розробка такого додатку виправдана необхідністю нового інструменту, який допоможе студентам зробити осмислений та обґрунтований вибір вибіркового дисциплін на основі аналізу сучасного стану проблеми та врахуванням рекомендацій з вітчизняної та зарубіжної науково-технічної літератури.

Основними рішеннями, що розглядаються в роботі, будуть розробка зручного інтерфейсу для перегляду доступних вибіркового дисциплін, надання детальної інформації про кожну дисципліну, врахування індивідуальних потреб та інтересів студентів, а також можливість зберігання вибраних дисциплін для майбутнього орієнтування.

Результати цієї роботи матимуть застосування у сфері вищої освіти та допоможуть студентам зробити обґрунтований вибір вибіркового дисциплін, що відповідають їхнім інтересам, потребам і кар'єрним цілям. Завдяки розробленому додатку, студенти зможуть ефективно ознайомитися з доступними вибіркового дисциплінами, отримати повну інформацію про них, таку як опис, вимоги та навчальні матеріали, що допоможе їм зробити обдуманий вибір. Додаток також надасть можливість збереження обраних дисциплін для подальшого перегляду та редагування.

У подальшому, розробка додатку може сприяти розвитку сфери освіти та технологій, вносячи нові можливості для покращення навчального процесу та індивідуального навчання. Крім того, вона може стати вихідною точкою для

подальших досліджень і розробок у галузі розширення функціоналу додатку та впровадження нових інноваційних рішень в освітньому середовищі. Враховуючи швидкий розвиток технологій та зростання числа студентів, які шукають індивідуалізований підхід до навчання, додаток для вибору вибіркових дисциплін стане цінним інструментом для полегшення процесу планування навчального шляху та досягнення академічних та кар'єрних цілей.

Для досягнення поставленої мети розробки додатку було проведено аналіз сучасного стану проблеми, вивчено наукові джерела та досвід роботи вітчизняних та зарубіжних дослідників у галузі освіти та розробки додатків. На основі цього аналізу були сформульовані основні рішення та напрямки розробки, спрямовані на створення зручного інтерфейсу та функціоналу додатку.

Отже, ця дипломна робота пропонує новий підхід до вибору вибіркових дисциплін студентами, пропонуючи зручний та ефективний додаток, який спростить процес вибору та надасть студентам необхідну інформацію для прийняття обґрунтованих рішень. Розробка додатку є актуальною, оскільки вона відповідає потребам сучасних студентів, які шукають зручні та надійні інструменти для планування свого навчального шляху.

1 ПОСТАНОВКА ЗАДАЧІ ВИБОРУ ВИБІРКОВИХ ДИСЦИПЛІН

Метою цієї дипломної роботи є розробка додатку для вибору вибіркового дисциплін, який спростить процес вибору та надасть студентам необхідну інформацію для обґрунтованих рішень. Додаток повинен забезпечити зручний інтерфейс для перегляду доступних дисциплін, надавати детальну інформацію про кожен дисципліну та дозволяти обирати дисципліни.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Аналіз сучасного стану проблеми: провести огляд існуючих підходів та додатків для вибору вибіркового дисциплін, вивчити наукові джерела та досвід роботи вітчизняних та зарубіжних дослідників у галузі освіти та розробки додатків.
2. Розробка зручного інтерфейсу: створити інтуїтивно зрозумілий та зручний інтерфейс для перегляду доступних вибіркового дисциплін, їх сортування та фільтрації за різними параметрами (наприклад, за спеціалізаціями, семестрами, рейтингом тощо).
3. Надання детальної інформації: розробити функціонал для надання студентам докладної інформації про кожен вибірково дисципліну, включаючи опис, навчальні матеріали, вимоги та рекомендації.

Вимоги до розроблюваних у роботі програмних засобів:

1. Користувацький інтерфейс повинен бути зручним та інтуїтивно зрозумілим для студентів різних рівнів комп'ютерної грамотності.
2. Можливість створення облікового запису
3. Додаток повинен мати доступ до бази даних зі списком доступних вибіркового дисциплін, яка має бути оновлювана та розширювана.
4. Функціонал додатка повинен забезпечувати перегляд детальної інформації про кожен дисципліну, включаючи опис, вимоги та рекомендації.
5. Додаток повинен мати можливість збереження обраних дисциплін для

6. Додаток повинен бути розроблений з врахуванням найновіших стандартів та технологій розробки програмного забезпечення, забезпечувати безпеку даних та масштабованість.
7. Додаток повинен бути платформо незалежним і працювати на різних операційних системах (наприклад, Windows, macOS, Linux) та пристроях (комп'ютери, смартфони, планшети).

Висновки до розділу 1

В даному розділі було поставлено задачі вибору вибіркового дисциплін. Додаток повинен забезпечити зручний інтерфейс для перегляду доступних дисциплін, надавати детальну інформацію про кожну дисципліну та дозволяти обирати дисципліни.

2 ОГЛЯД ПОДІБНИХ ІСНУЮЧИХ СИСТЕМ

Тема додатку для вибору вибіркових дисциплін належить до сфери освіти і студентського самоврядування. Такий додаток може надати студентам зручний інструмент для вибору дисциплін, які найкраще відповідають їхнім інтересам, потребам та навчальним цілям. Основні функції та можливості такого додатку можуть включати:

1. Перегляд списку доступних вибіркових дисциплін: додаток має надати студентам можливість ознайомитись зі списком вибіркових дисциплін, які пропонуються у їхньому навчальному закладі.
2. Детальна інформація про дисципліни: студенти повинні мати можливість переглядати докладну інформацію про кожну дисципліну, включаючи назву, опис, викладача, кількість кредитів, рекомендації та вимоги.
3. Обрання дисциплін: додаток повинен надати можливість студентам обирати дисципліни.

2.1 Модуль “Вибір дисциплін” системи Електронний кампус

Електронний кампус – це система підтримки навчального процесу університету. На рисунку 2.1 показано систему Електронний кампус.

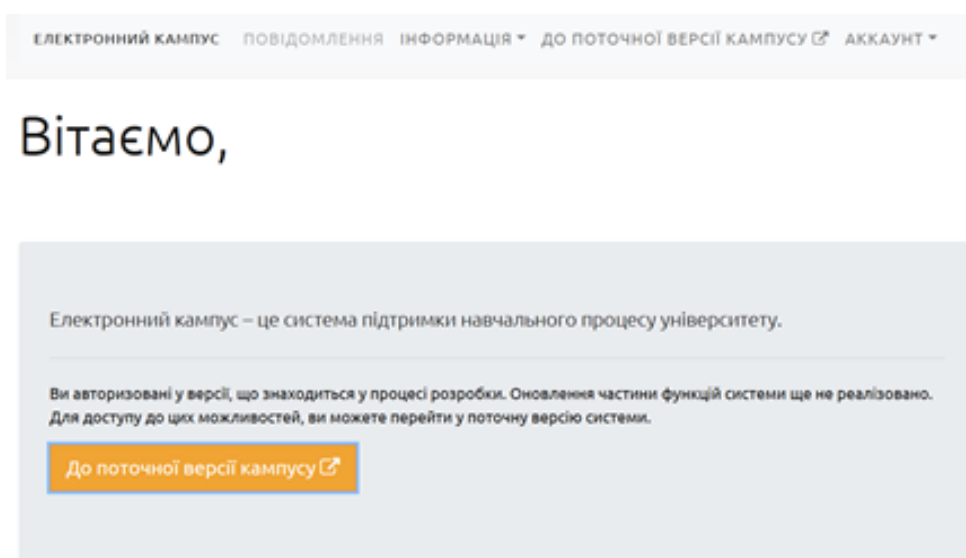


Рисунок 2.1 - Система Електронний кампус

Для того, щоб здійснити вибір дисциплін у системі Електронний кампус, потрібно обрати модуль “Вибір дисциплін”. Перед вами відкривається вікно з інформацією як показано на рисунку 2.2.

Обрані дисципліни 1

Шаблони для вибору

Назва шаблону	Курс	Семестр	ECTS	Тип контролю	Дисципліни																																										
ПВ1	2	4	4	залік	<table border="1"> <thead> <tr> <th>#</th> <th>Назва дисципліни</th> <th>Залишилось місць</th> <th>Чит. кафедра</th> <th>URL</th> <th>Дія</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td>AGILE методологія розробки програмного забезпечення</td> <td>60</td> <td>ІТ ФІОТ</td> <td>Переглянути</td> <td>Обрати</td> </tr> <tr> <td><input type="radio"/></td> <td>Python технологій у операційних системах</td> <td>60</td> <td>ІСТ ФІОТ</td> <td>Переглянути</td> <td>Обрати</td> </tr> <tr> <td><input type="radio"/></td> <td>Ігровий штучний інтелект</td> <td>60</td> <td>ІПІ ФІОТ</td> <td>Переглянути</td> <td>Обрати</td> </tr> <tr> <td><input type="radio"/></td> <td>Інженерія знань</td> <td>90</td> <td>ІПІ ФІОТ</td> <td>Переглянути</td> <td>Обрати</td> </tr> <tr> <td><input type="radio"/></td> <td>Інтелектуальні вбудовані системи</td> <td>60</td> <td>ІТ ФІОТ</td> <td>Переглянути</td> <td>Обрати</td> </tr> <tr> <td><input type="radio"/></td> <td>Автоматизація життєвого циклу програмних продуктів</td> <td>90</td> <td>ІПІ ФІОТ</td> <td>Переглянути</td> <td>Обрати</td> </tr> </tbody> </table>	#	Назва дисципліни	Залишилось місць	Чит. кафедра	URL	Дія	<input type="radio"/>	AGILE методологія розробки програмного забезпечення	60	ІТ ФІОТ	Переглянути	Обрати	<input type="radio"/>	Python технологій у операційних системах	60	ІСТ ФІОТ	Переглянути	Обрати	<input type="radio"/>	Ігровий штучний інтелект	60	ІПІ ФІОТ	Переглянути	Обрати	<input type="radio"/>	Інженерія знань	90	ІПІ ФІОТ	Переглянути	Обрати	<input type="radio"/>	Інтелектуальні вбудовані системи	60	ІТ ФІОТ	Переглянути	Обрати	<input type="radio"/>	Автоматизація життєвого циклу програмних продуктів	90	ІПІ ФІОТ	Переглянути	Обрати
#	Назва дисципліни	Залишилось місць	Чит. кафедра	URL	Дія																																										
<input type="radio"/>	AGILE методологія розробки програмного забезпечення	60	ІТ ФІОТ	Переглянути	Обрати																																										
<input type="radio"/>	Python технологій у операційних системах	60	ІСТ ФІОТ	Переглянути	Обрати																																										
<input type="radio"/>	Ігровий штучний інтелект	60	ІПІ ФІОТ	Переглянути	Обрати																																										
<input type="radio"/>	Інженерія знань	90	ІПІ ФІОТ	Переглянути	Обрати																																										
<input type="radio"/>	Інтелектуальні вбудовані системи	60	ІТ ФІОТ	Переглянути	Обрати																																										
<input type="radio"/>	Автоматизація життєвого циклу програмних продуктів	90	ІПІ ФІОТ	Переглянути	Обрати																																										

2 3 4 5 6

Рисунок 2.2 - Модуль “Вибір дисциплін”

В блоці 1 буде відображено обрані вами дисципліни після того, як ви здійсните їх обрання в блоці Шаблони для вибору.

В навчальних планах в частині вибіркових дисциплін прописуються освітні компоненти по типу «Дисципліна за вибором 1», «Дисципліна за вибором 2» або щось схоже.

Кожен з таких освітніх компонентів характеризується Курсом та Семестром, коли він буде вивчатись, Обсягом кредитів Єдиної кредитно-трансферної системи (ECTS) та Типом контролю (переважно залік, але може бути і екзамен). Всі ці параметри вказуються в блоці 2.

Назва шаблону в Електронному кампусі є простим ідентифікатором, який особливого значення для вас не має. На рисунку 2.3 показано шаблони для вибору.

Шаблони для вибору

Назва шаблону	Курс	Семестер	ECTS	Тип контролю
ПВ1	2	4	4	залік

2

Рисунок 2.3 - Шаблони для вибору

В рамках кожного такого освітнього компоненту, наприклад, «Дисципліна за вибором 1» кафедра передбачає декілька різних дисциплін, які власне і обирає студент. Ці дисципліни відрізняються за назвою та змістом – детальніше про суть дисциплін можна почитати в описах дисциплін, які розміщуються на сайті кафедри/факультету і посилання на які вказані в блоці 5 URL.

Обрання дисциплін здійснюється радіокнопками – блок 3 як показано на рисунку 2.4.

Назва шаблону	Курс	Семестер	ECTS	Тип контролю	Дисципліни					
ПВ1	2	4	4	залік	#	Назва дисципліни	Залишилось місьць	Чит. кафедра	URL	Дія
					<input type="radio"/>	AGILE методологія розробки програмного забезпечення	60	ІТ ФІОТ	Переглянути	Обрати
					<input checked="" type="radio"/>	Python технологій у операційних системах	60	ІСТ ФІОТ	Переглянути	Обрати

Рисунок 2.4 - Обрання дисциплін

Для того, щоб система запам'ятала ваш вибір, необхідно натиснути кнопку Обрати – будь-яку з блоку 6 як показано на рисунку 2.5. Всі кнопки блоку 6 є ідентичними – виокремлені виключно для зручності роботи з вікном у випадку великої кількості дисциплін у кожному із шаблонів.

#	Назва дисципліни	Залишилось місьць	Чит. кафедра	URL	Дія
<input type="radio"/>	AGILE методологія розробки програмного забезпечення	60	ОТ ФІОТ	Переглянути	Обрати
<input checked="" type="radio"/>	Python технологій у операційних системах	60	ІСТ ФІОТ	Переглянути	Обрати
<input type="radio"/>	Ігровий штучний інтелект	60	ІПІ ФІОТ	Переглянути	Обрати

6

Рисунок 2.5 - Обрання дисципліни

Після того, як ви натиснете кнопку Обрати – ви побачите вікно попередження щодо вибору, який ви здійснюєте. На рисунку 2.6 показано вікно попередження.

Обрані дисципліни

Shabl... Ви обрали Python технологій у операційних системах, яку читає кафедра ІСТ(ФІОТ).
Ви впевнені у своєму виборі?

OK Отмена

Назва шаблону	Курс	Семестер	ECTS	Тип контролю	Дисципліни																		
ПВ1	2	4	4	залік	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Назва дисципліни</th> <th>Залишилось місьць</th> <th>Чит. кафедра</th> <th>URL</th> <th>Дія</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td>AGILE методологія розробки програмного забезпечення</td> <td>60</td> <td>ОТ ФІОТ</td> <td>Переглянути</td> <td>Обрати</td> </tr> <tr> <td><input checked="" type="radio"/></td> <td>Python технологій у операційних системах</td> <td>60</td> <td>ІСТ ФІОТ</td> <td>Переглянути</td> <td>Обрати</td> </tr> </tbody> </table>	#	Назва дисципліни	Залишилось місьць	Чит. кафедра	URL	Дія	<input type="radio"/>	AGILE методологія розробки програмного забезпечення	60	ОТ ФІОТ	Переглянути	Обрати	<input checked="" type="radio"/>	Python технологій у операційних системах	60	ІСТ ФІОТ	Переглянути	Обрати
#	Назва дисципліни	Залишилось місьць	Чит. кафедра	URL	Дія																		
<input type="radio"/>	AGILE методологія розробки програмного забезпечення	60	ОТ ФІОТ	Переглянути	Обрати																		
<input checked="" type="radio"/>	Python технологій у операційних системах	60	ІСТ ФІОТ	Переглянути	Обрати																		

Рисунок 2.6 - Вікно попередження

Після натиснення ОК ви не зможете змінити свій вибір і обрана вами дисципліна зафіксується в системі і відобразиться в блоці 1 як показано на рисунку 2.7.

Обрані дисципліни

ShablonName	ForCourse	ForSemester	CountCredit	SemesterControlType	Хвиля	Дисципліна
ПВ1	2	4	4	залік	2	Python технологій у операційних системах

Рисунок 2.7 - Обрані дисципліни

У кожному шаблоні передбачено обрання лише 1 дисципліни, відповідно після того, як вона обрана, шаблон зникає. Залишаються лише ті шаблони, в яких дисципліни ще не обирались. На рисунку 2.8 показано шаблони для вибору.

Шаблони для вибору

Назва шаблону	Курс	Семестр	ECTS	Тип контролю	Дисципліни					
30-1	2	3	2	залік	#	Назва дисципліни	Залишилось місце	Чит. кафедра	URL	Дія
30-2	2	4	2	залік	#	Назва дисципліни	Залишилось місце	Чит. кафедра	URL	Дія
					<input type="radio"/>	Logic	50	КФ ФСП	Переглянути	Обрати

Рисунок 2.8 - Шаблони для вибору після обрання дисципліни

2.2 Модуль “Вибір дисциплін” системи ту.крі.уа

Система ту.крі.уа – система, для організації навчального процесу, практики та стажування.

На рисунку 2.9 показано систему ту.крі.уа.

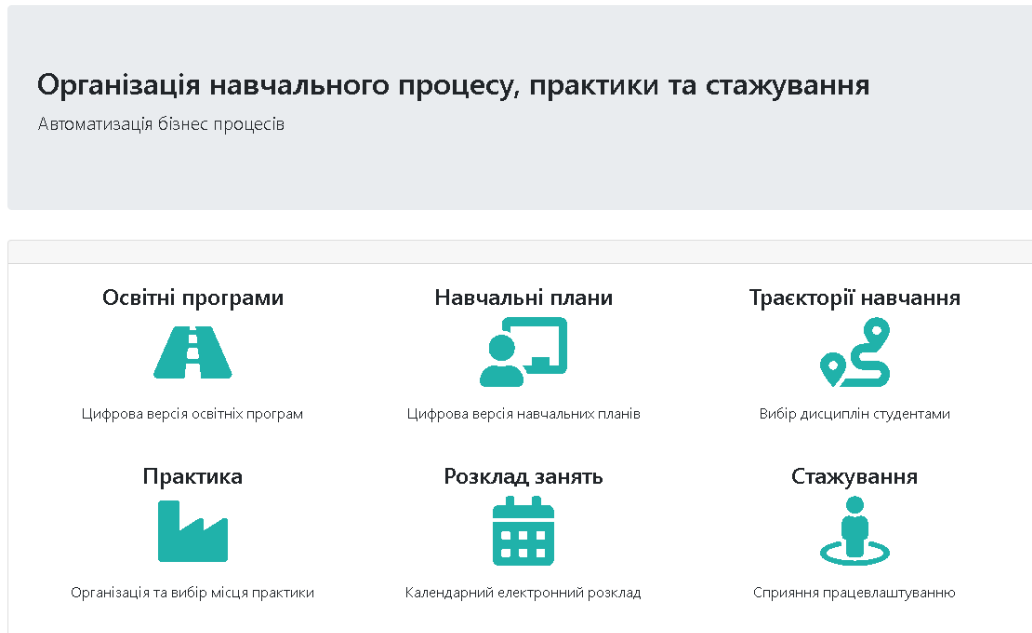


Рисунок 2.9 - Система my.kpi.ua

Для того, щоб здійснити вибір дисциплін у системі my.kpi.ua, потрібно натиснути кнопку “Вибір” у кабінеті студента. Кабінет студента показано на рисунку 2.10.

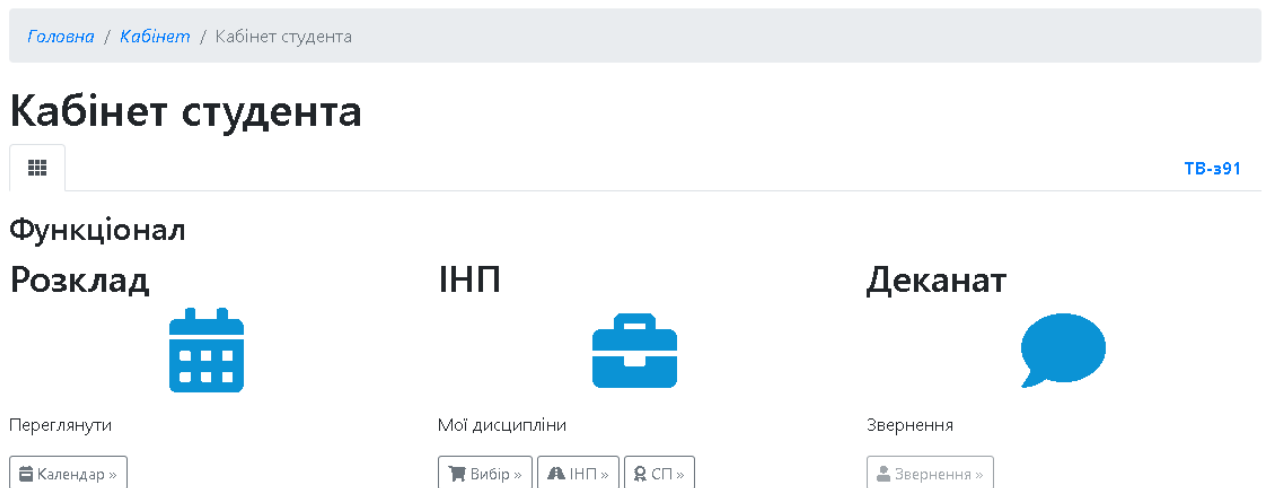


Рисунок 2.10 – Кабінет студента

Перед вами відкривається сторінка як показано на рисунку 2.11.

Вибір дисциплін



ТВ-991

2022 - 4-й курс

Рисунок 2.11 – Вибір дисциплін

Потрібно натиснути на посилання для обрання дисциплін на певний курс. Тоді, ви побачите сторінку зі списком освітніх компонент, як показано на рисунку 2.12.

Вибір дисциплін



ТВ-991

2022 - 4-й курс

Освітній компонент 14
Ф-каталогу

8 семестр



Проектування геоінформаційних си...

Ф-Каталог 121Б-ТВ14Ф-1

Освітній компонент 7
Ф-каталогу

7 семестр



Розподілені бази даних...

Ф-Каталог 121Б-ТВ7Ф-2

Освітній компонент 8
Ф-каталогу

7 семестр



Процеси неперервної інтеграції і...

Ф-Каталог 121Б-ТВ1Ф-2

Освітній компонент 9
Ф-каталогу

7 семестр

Освітній компонент 10
Ф-каталогу

7 семестр

Освітній компонент 11
Ф-каталогу

7 семестр

Рисунок 2.12 – Вибір дисциплін (освітні компоненти)

Обрані вами дисципліни будуть відображатися знизу кожної освітньої компоненти.

Якщо обрати якусь освітню компоненту то ви побачите список дисциплін цієї освітньої компоненти. На рисунку 2.13 показано каталог дисциплін.

← Ф-Каталог [Дорожня карта](#)

Каталог дисциплін, що доступні для вибору на місці освітньої компоненти "Освітній компонент 14 Ф-каталогу" програми підготовки "121Б Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем і веб-технологій ІПЗЕ 2019" у 8-му семестрі

Показані 1-25 із 79 записів.

#	Код	Назва дисц.	Викладачі	Категорія	Каф.	Вибір
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	151Б-В-306	Прикладна механіка і основи конструювання >>	...	(не задано)	ПГМ	<input type="button" value="Записатись"/>
2	151Б-В-307	Математичне моделювання в теплоенергетиці >>	...	(не задано)	АЕП	<input type="button" value="Записатись"/>
3	151Б-В-309	Технології відновлюваної енергетики >>	...	(не задано)	АЕП	<input type="button" value="Записатись"/>
4	151Б-В-310	Веб-технології >>	...	(не задано)	АЕП	<input type="button" value="Записатись"/>

Рисунок 2.13 – Каталог дисциплін

Для того, щоб обрати дисципліну, необхідно натиснути кнопку Записатись – як показано на рисунку 2.14. Всі кнопки є ідентичними – виокремлені виключно для зручності роботи з вікном у випадку великої кількості дисциплін у кожному із шаблонів.

1	151Б-В-306	Прикладна механіка і основи конструювання >>	...	(не задано)	ПГМ	<input type="button" value="Записатись"/>
---	------------	--	-----	-------------	-----	---

Рисунок 2.14 Обрання дисципліни

Після того, як ви натиснете кнопку Записатись – ви побачите вікно попередження щодо вибору, який ви здійснюєте. На рисунку 2.15 показано вікно попередження.

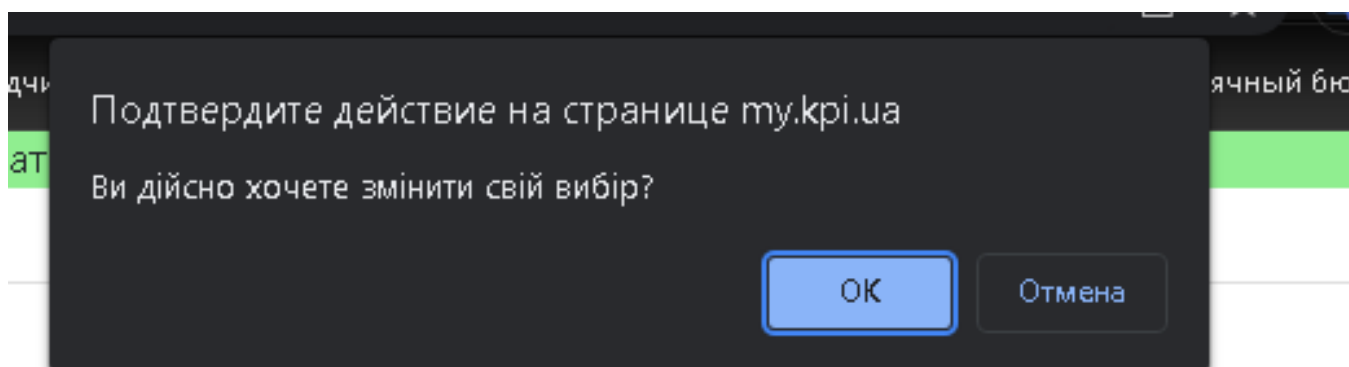


Рисунок 2.15 - Вікно попередження

Після натиснення ОК ви не зможете змінити свій вибір і обрана вами дисципліна зафіксується в системі і буде відображатися знизу освітньої компоненти. Щоб подивитися інформацію про обрану дисципліну можна натиснути зелену кнопку як показано на рисунку 2.16.

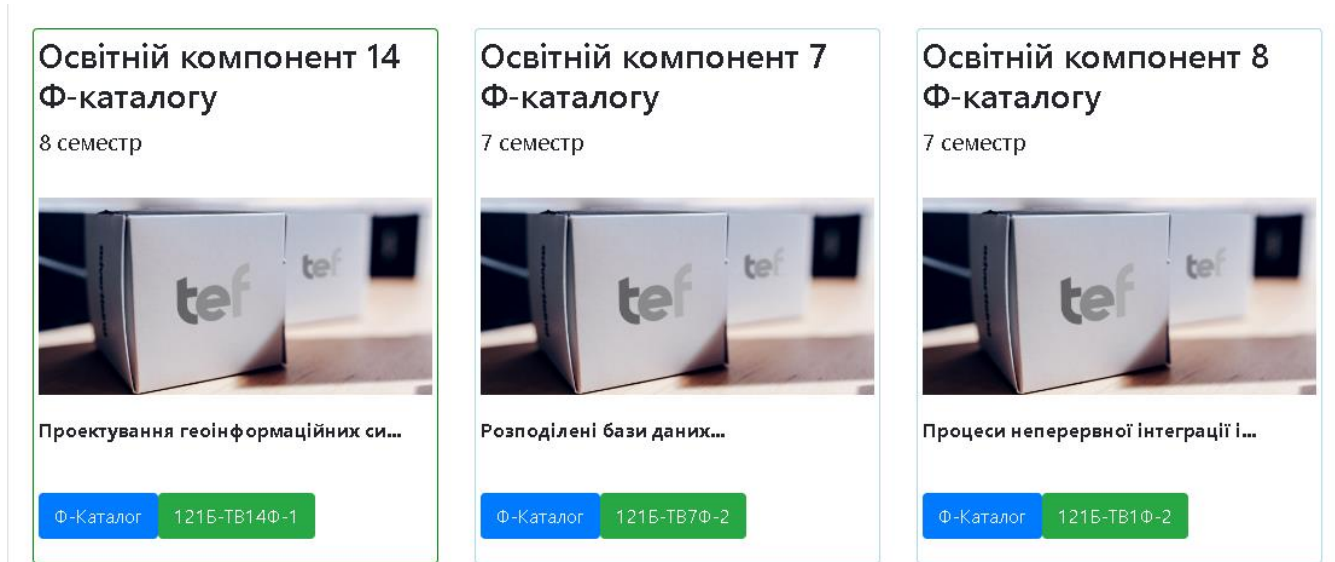


Рисунок 2.16 – Перегляд інформації про обрану дисципліну

Після цього ви побачите вікно з інформацією про обрану дисципліну. Там буде відображено інформацію про кафедру, чи можливе викладання англійською, вимоги до початку вивчення, що буде вивчатися, чому це цікаво, чому можна навчитися, як можна користуватися набутими знаннями і уміннями.

На рисунку 2.17 показано вікно з інформацією про обрану дисципліну.

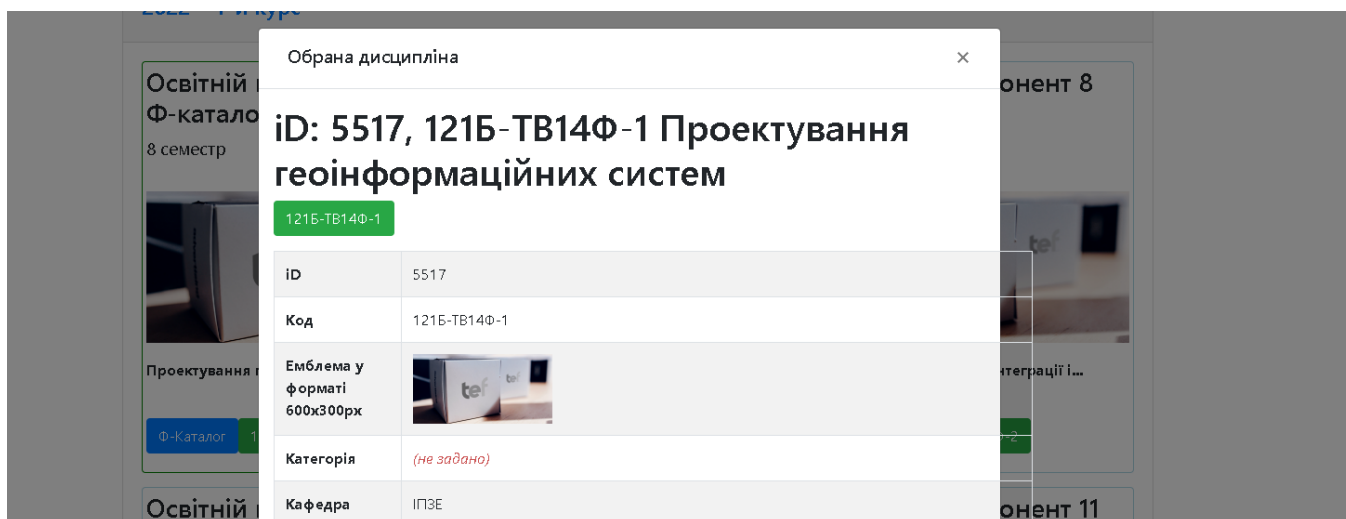


Рисунок 2.17 – Вікно з інформацією

Висновки до розділу 2

В даному розділі було розглянуто існуючі системи для вибору вибіркових дисциплін, а саме Електронний кампус та систему tu.kpi.ua.

Електронний кампус – це система підтримки навчального процесу університету.

Система tu.kpi.ua – система, для організації навчального процесу, практики та стажування.

3 ЗАСОБИ РОЗРОБКИ СИСТЕМИ

Даний додаток призначений для вибору вибіркового дисциплін університету. Вибіркові дисципліни є однією з головних складових частин університетської освіти, що дає можливість студентам самостійно формувати своє навчальне середовище та вибирати ті дисципліни, які їм цікаві та необхідні для отримання більш глибоких знань у конкретній сфері.

Проте, вибір вибіркового дисциплін може стати важливим викликом для студентів, особливо для першокурсників, які ще не мають достатнього досвіду в навчанні та визначенні своїх інтересів. Відсутність адекватної інформації про дисципліни, вимоги до них, а також рекомендації щодо їх вибору може призвести до невдачі у навчанні, зниження мотивації та інтересу до конкретної галузі знань.

Для вирішення цих проблем розроблено додаток, який дає можливість студентам отримати необхідну інформацію про вибіркові дисципліни та здійснити свій вибір з урахуванням своїх інтересів та потреб. Додаток забезпечує доступ до повної інформації про кожну вибірку дисципліну, включаючи її назву, опис, список необхідних попередніх знань, вимоги до кількості годин, форму проведення занять та інші деталі. Також додаток надає можливість здійснити пошук дисциплін за різними критеріями, такими як назва, кількість кредитів, факультет тощо. Крім того, додаток містить розділ з рекомендаціями щодо вибору дисциплін, які базуються на інтересах та потребах студента, а також на його попередніх знаннях та досвіді.

Додаток базується на технології Angular та Firebase, що дозволяє забезпечити його швидкість та надійність. Angular - це популярний фреймворк для розробки веб-додатків, який забезпечує швидку та ефективну роботу з більшою кількістю компонентів та модулів. Firebase - це платформа для розробки веб-додатків, яка надає послуги зберігання даних, аутентифікації, хостингу, аналітики та інших функцій, що дозволяють розробникам зосередитись на розробці функціоналу додатку.

Додаток також має можливість входу для студентів та адміністраторів. Адміністратор може додавати, редагувати та видаляти дисципліни з бази даних, а також керувати користувачами. Крім того, додаток має можливість збереження

списку вибраних дисциплін студента, щоб він міг зручно слідкувати за своїм навчанням та отримувати інформацію про зміни у розкладі занять та інші оновлення.

У загальному, додаток допомагає студентам зробити обдуманий вибір вибіркових дисциплін, що відповідає їхнім потребам та інтересам, та забезпечує зручну та доступну інформацію про кожен дисципліну. Такий інструмент дозволяє підвищити якість навчання студентів, сприяє їхньому успіху та підвищує їхню мотивацію до вивчення предметів. Крім того, додаток також допомагає адміністраторам ефективно керувати даними та користувачами, зменшуючи час та зусилля, потрібні для виконання цих завдань вручну.

Додаток може бути корисним не тільки для студентів вищих навчальних закладів, але і для викладачів та керівників навчальних програм. Викладачі можуть використовувати додаток для аналізу інтересів та потреб студентів та надавати рекомендації щодо вибору дисциплін, які відповідають їхнім потребам. Керівники навчальних програм можуть використовувати дані, що збираються додатком, для аналізу ефективності навчальних програм та вдосконалення їхньої структури.

У світі, де технології швидко розвиваються та змінюються, цей додаток може бути корисним інструментом для поліпшення якості навчання та забезпечення доступної та зручної інформації для студентів, викладачів та керівників навчальних програм. Застосування таких технологій, як Angular та Firebase, дозволяє створювати ефективні та надійні додатки, що забезпечують користувачам зручний та доступний інтерфейс для отримання необхідної інформації та управління даними.

Для розробки фронтенду додатку було використано фреймворк Angular. Angular - це відкритий фреймворк для розробки веб-додатків, який використовується для створення односторінкових додатків. Angular базується на TypeScript, який є суперсетом мови JavaScript, тому код стає більш структурованим та безпечнішим. TypeScript є мовою програмування, що розширює JavaScript та дозволяє розробникам створювати більш структурований та безпечний код.[7] Окрім того, Angular дозволяє розбити додаток на окремі компоненти, які можна повторно використовувати, що спрощує розробку та збільшує швидкість роботи додатку.

Для зберігання даних було використано Firebase - хмарну платформу від Google, яка дозволяє розробникам зосередитись на розробці функціоналу додатку, а не на налаштуванні сервера та бази даних. Firebase надає послуги зберігання даних, аутентифікації, хостингу, аналітики та інших функцій, що значно спрощує розробку та розвиток додатку.

Для розробки інтерфейсу користувача було використано HTML, CSS та JavaScript.[10] HTML використовувався для створення структури сторінок, CSS - для оформлення та стилізації елементів, а JavaScript - для додавання динамічного функціоналу.[9] Для забезпечення більшої зручності та ефективності розробки, було використано препроцесор CSS - Scss, який дозволяє використовувати змінні, міксіни та інші функції, що значно спрощує написання CSS-коду.[8]

Також було використано бібліотеку PrimeNG, яка надає готові компоненти інтерфейсу користувача для Angular. Завдяки PrimeNG можна швидко розробляти естетичний інтерфейс користувача, не витрачаючи багато часу на написання власних компонентів. Бібліотека PrimeNG містить такі компоненти, як таблиці, кнопки, форми, меню, діалогові вікна та багато іншого. Усі ці засоби розробки дозволили створити функціональний, зручний та безпечний додаток для обрання вибіркового дисциплін для студентів, який може ефективно працювати та забезпечувати користувачам зручний та безпечний інтерфейс для вибору дисциплін.

3.1 Середовище розробки WebStorm

Для розробки програмного коду було вибрано інтегроване середовище розробки WebStorm, яке надає зручні інструменти для покращення ефективності розробки. Інтегроване середовище розробки забезпечує автодоповнення коду, навігацію по коду, аналіз коду на синтаксичні помилки та інтеграцію з системою контролю версій. Використання інтегрованого середовища розробки значно полегшує процес розробки, дозволяючи розробникам працювати швидше і ефективніше. На рисунку 3.1 зображено вікно IDE WebStorm.

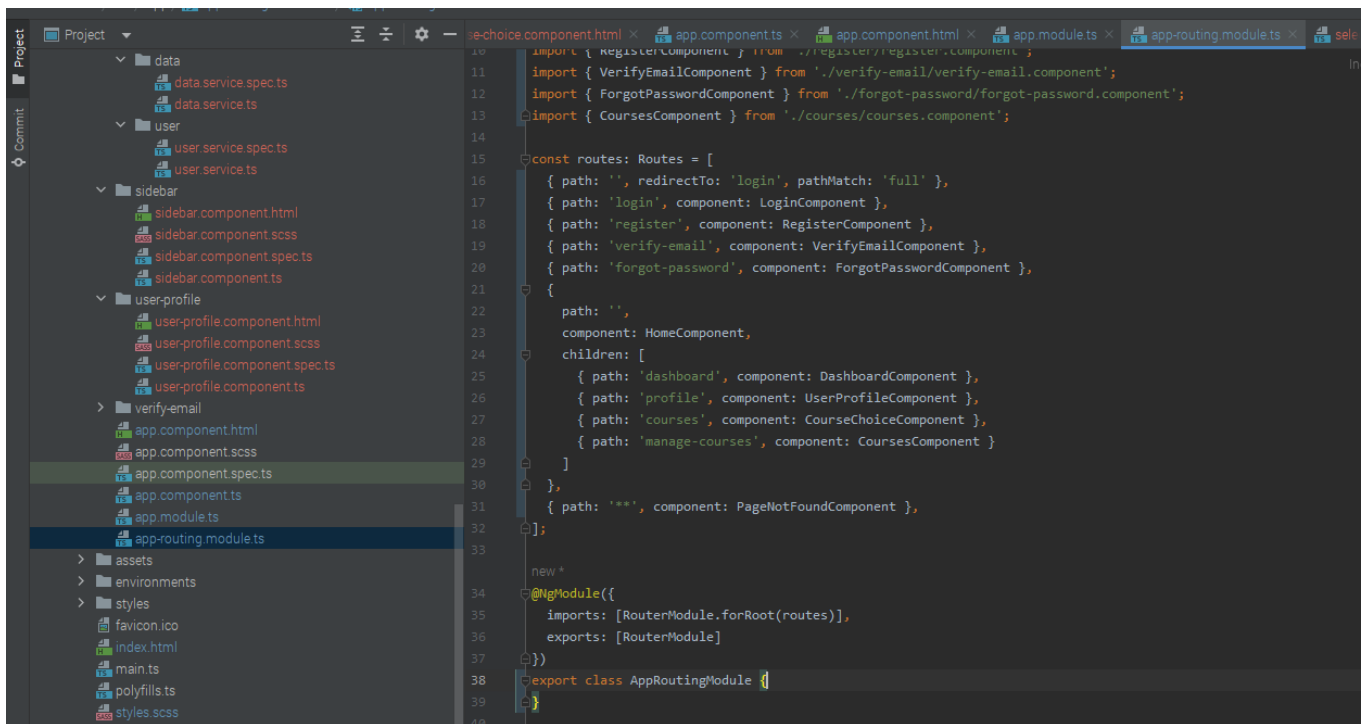


Рисунок 3.1- Вікно IDE WebStorm

WebStorm - це інтегроване середовище розробки, спеціально розроблене для розробки веб-додатків з наперед встановленими плагінами, які допомагають розробникам писати код на мовах програмування, таких як JavaScript, TypeScript, HTML та CSS. Крім того, середовище має підтримку різних фреймворків для створення інтерфейсів, таких як Angular, React та Vue.js, що робить процес розробки більш продуктивним та ефективним. Завдяки своїм функціональним можливостям, таким як автодоповнення коду, навігація по коду та аналіз наявних помилок, WebStorm допомагає розробникам писати чистий та ефективний код з меншим ризиком виникнення помилок. Крім того, середовище має інтеграцію з системою контролю версій, що дозволяє розробникам зберігати та керувати історією змін у своєму проєкті.

3.2 Налаштування робочого оточення

3.2.1 Встановлення Node.js

Для розробки клієнтської частини веб-застосунку з використанням Angular необхідно мати встановлене середовище Node.js. Node.js є середовищем виконання для JavaScript, яке дозволяє розробляти програми на мові JavaScript, які можуть функціонувати як поза браузером, так і на сервері. Фреймворк Node.js містить у собі API для роботи з файловою системою, доступу до баз даних та прослуховування запитів HTTP. На рисунку 3.2 зображено головну сторінку Node.js.[5]

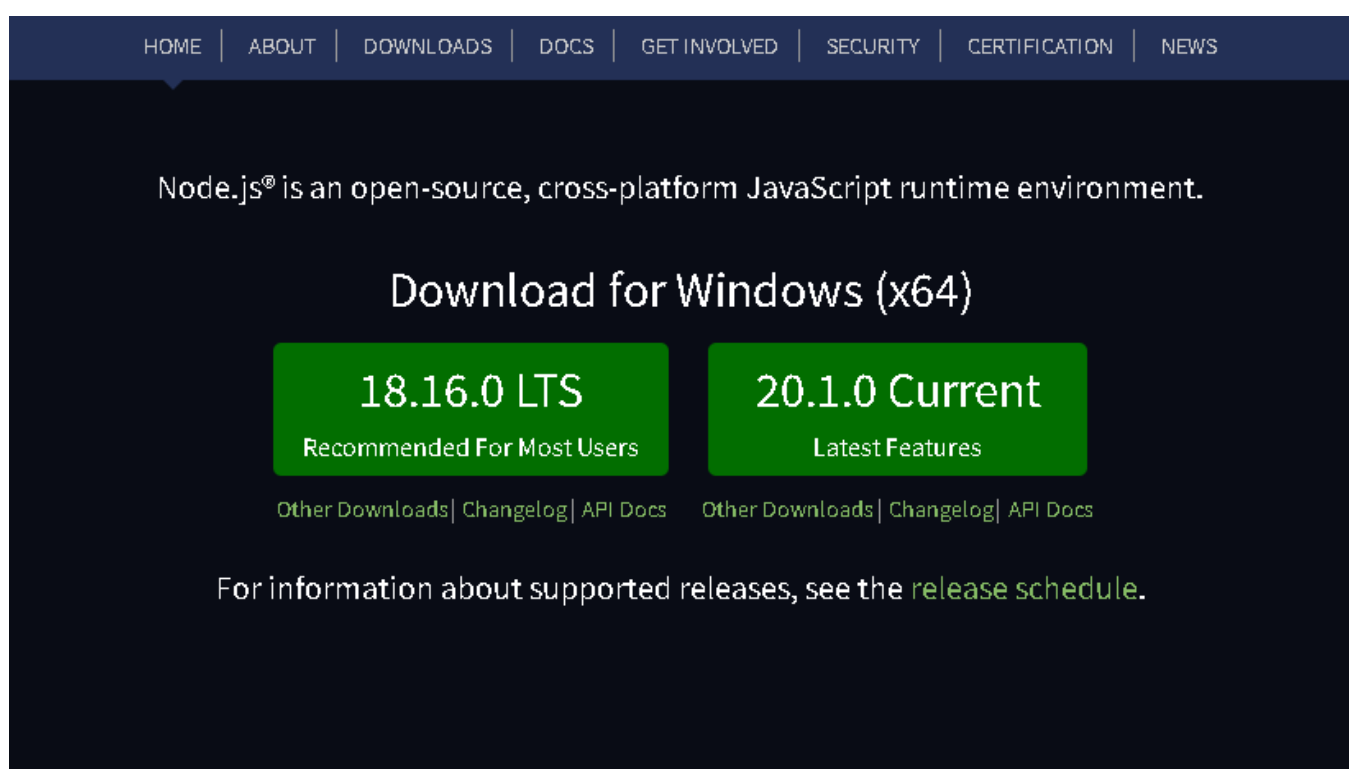


Рисунок 3.2 - Головна сторінка Node.js

Node.js включає в себе пакетний менеджер npm (Node Package Manager), який є потужним інструментом для управління залежностями проекту на мові JavaScript. Він дозволяє легко встановлювати, оновлювати та видаляти пакети, які знаходяться в реєстрі npm, а також керувати залежностями проекту. За допомогою npm можна швидко і легко встановити та оновити пакети, що використовуються в проекті, і забезпечити сумісність між різними версіями пакетів. Крім того, npm дозволяє зберігати локальні залежності програми чи інших пакетів, які використовуються,

для зручного управління проектом. Це дозволяє зберігати проект портативним та легко переносимим між різними середовищами розробки.

3.2.2 Встановлення Angular CLI

Для того щоб створити новий проект на Angular можна використати Angular CLI, що надає зручний інтерфейс командного рядка для управління розробкою Angular проекту. На рисунку 3.3 зображено встановлення Angular CLI.



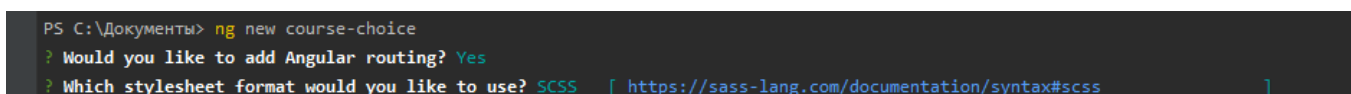
Рисунок 3.3 - Встановлення Angular CLI

Angular CLI надає інтерфейс командного рядка, який дозволяє виконувати різні операції з проектом, написаним на Angular. Деякі з основних команд включають:

- `ng new`: створює новий проект Angular
- `ng generate`: генерує нові файли коду Angular, такі як компоненти, сервіси, модулі тощо
- `ng serve`: запускає локальний сервер для розробки та перегляду проекту в браузері
- `ng build`: збирає проект для розгортання на сервері
- `ng test`: запускає модульні тести для проекту
- `ng lint`: перевіряє код на відповідність стандартам програмування.

3.2.3 Створення нового проекту

Команда `ng new`, викликана в терміналі, дозволяє створити новий проект Angular в поточній директорії. При цьому необхідно вказати назву проекту, вибрати стиль каскадних таблиць та опцію маршрутизації веб-застосунку. Після виконання команди в поточній директорії буде створена нова папка з назвою, яка була вказана раніше. Таким чином, буде створено новий Angular проект, готовий до подальшої розробки. На рисунку 3.4 зображено створення нового проекту.



```
PS C:\Документи> ng new course-choice
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? SCSS [ https://sass-lang.com/documentation/syntax#scss ]
```

Рисунок 3.4 – Створення нового проекту

Після створення нового проекту у Angular, всі необхідні конфігураційні файли та кореневий модуль програми будуть автоматично згенеровані. Також будуть встановлені всі пакети, що вказані у файлі `package.json`. Серед основних пакетів, що використовуються в застосунку Angular, можна виділити такі:

- `@angular/core`, який є найбільш важливим пакетом, містить основу Angular і його загальні елементи;
- `@angular/common`, який надає загальні функціональність, яка використовується у багатьох додатках. Він включає в себе різноманітні корисні сервіси, директиви і утиліти, такі як маршрутизація, обробка подій, маніпулювання даними форм, робота з URL-адресами, міжнародна локалізація, підтримка браузерних подій та багато іншого.;
- `@angular/compiler`, який відповідає за компіляцію і перетворення шаблонів Angular у виконуваний код JavaScript;
- `@angular/platform-browser`, який надає засоби для роботи з браузером, такі як маніпулювання DOM-елементами, керування заголовками сторінки, робота з URL-адресою та інші функції, пов'язані з веб-браузером. Використовується для платформозалежного розробки Angular додатків, які працюють у веб-браузері;

- @angular/platform-browser-dynamic, який надає засоби для динамічного компілювання і запуску Angular додатків у веб-браузері. Використовується для розгортання та виконання Angular додатків на клієнтській стороні ;

- @angular/http, який надає можливості для виконання HTTP-запитів і взаємодії зі сторонніми серверами у веб-додатках;

- @angular/router, який надає можливість визначати маршрути веб-додатку та керувати переходами між різними компонентами і сторінками в програмі.[4]

3.3 Фреймворк Angular

Angular - це фреймворк від компанії Google, який дозволяє створювати передові та безперервні (односторінкові) веб-додатки - SPA (односторінкові додатки) - за допомогою мов програмування TypeScript, JavaScript та Dart.[4] На рисунку 3.5 зображено архітектуру Angular програми.

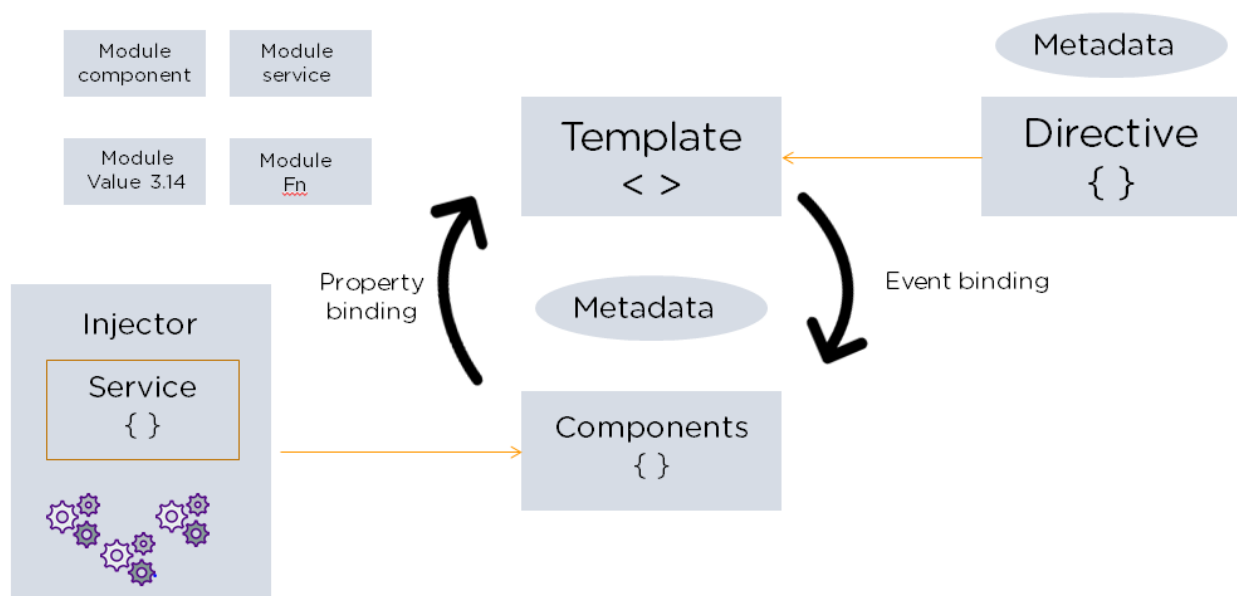


Рисунок 3.5 – Архітектура Angular програми

3.3.1 Компоненти

Компоненти - це великі складові частини додатка, які не залежать одна від одної.[6] Наприклад, один компонент може бути стрічкою новин, а інший - заголовком веб-сайту. Додаток будується з цих компонентів, як з блоків. Зазвичай

кожен компонент зберігається в окремому файлі. Для нього можна створити свій HTML-шаблон та CSS-стилі. Вони можуть знаходитися у тому ж файлі, що й компонент, або підключатися окремо. Створюється готовий блок інтерфейсу зі структурою, стилями та певною логікою поведінки. На рисунку 3.6 зображено ієрархію компонентів.

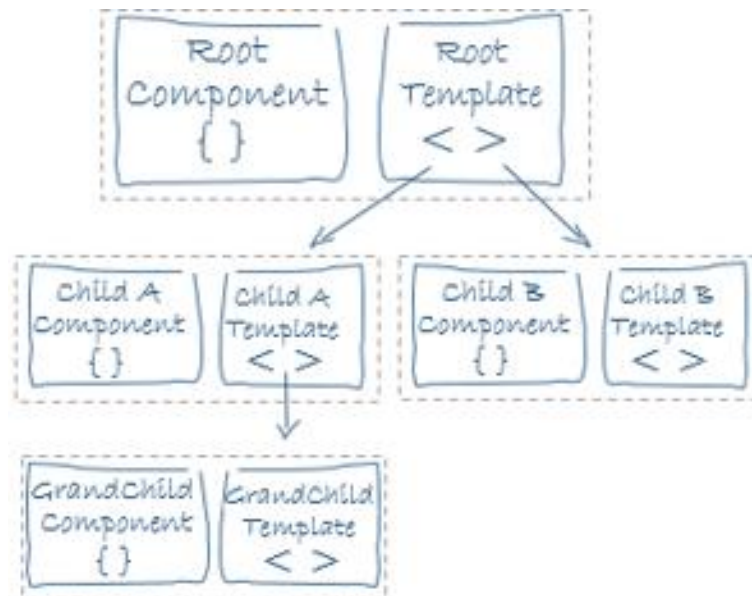


Рисунок 3.6 – Ієрархія компонентів

3.3.2 Модулі

Це ще одна складова частина додатка, але з іншим призначенням. Вони керують компонентами. Якщо компонент є окремою областю додатка, то модуль відповідає за його управління. Точка входу в додаток, код для анімації або навігації - все це модулі.

Основний модуль є в кожному проекті. Додаткові модулі додаються по мірі необхідності і виконують конкретні завдання. Вони потрібні, щоб не перевантажувати основний модуль зайвим функціоналом і не робити його надто важким. На рисунку 3.7 зображено приклад модуля.



Рисунок 3.7 – Модуль

3.3.3 Форми

Більшість додатків на Angular базуються на формах. Форма є структурою, в яку користувач вводить певні дані і потім надсилає їх на сервер. Наприклад, блок для написання коментарів або зворотного зв'язку є формою.

Angular спрощує роботу з формами, оскільки їх не потрібно писати з нуля. Вже створено шаблони, які потрібно адаптувати під нове завдання.

3.3.4 Сервіси

Сервіси схожі на компоненти, але більш спеціалізовані. Вони можуть бути визначені на рівні модуля, компонента або додатку. У сервісах реалізується спеціальна логіка. Вони підключаються до додатку як звичайні класи і використовуються для зберігання глобального стану додатку. Також вони використовуються як постачальники даних.

3.3.5 Директиви

Директиви є складовою частиною додатка, які змінюють структуру або поведінку сторінки. Компоненти також відносяться до директив. Однак, окрім компонентів, існують ще два типи директив: структурні директиви та директиви, що змінюють вигляд або поведінку елементів. Вони потрібні для застосування однієї дії

до всіх екземплярів певного компонента, наприклад, зміна валюти в усіх карточках товару.

3.3.6 Переваги Angular

Варто виділити основні переваги Angular.

1. Багатофункціональність

Angular надає можливість зв'язувати компоненти додатка один з одним, передавати дані, анімувати інтерфейси та багато іншого. Для простих проектів його функціонал може здаватися надмірним, але для складних односторінкових додатків він є незамінним.

2. Універсальність застосування

Цей фреймворк не обмежується створенням лише веб-додатків. З його допомогою можна писати код, який може бути адаптований для різних середовищ. Наприклад, додаток може працювати на мобільній або настільній операційній системі. За допомогою Angular навіть можна створити додаток для доповненої реальності.

3. Детальний стиль програмування

Особливістю Angular є докладна документація. Вона містить рекомендації щодо побудови та розробки додатків, включаючи посібник зі стилю програмування на Angular. Це зручно для розробників, які вперше зіштовхнулися з цим фреймворком. Єдність стилю допомагає програмістам краще розуміти код один одного.

4. Підтримка від Google

Розробники Angular є співробітниками Google, а підтримка великої корпорації допомагає фреймворку розвиватися. При цьому, завдяки вільній ліцензії та відкритому вихідному коду, його можуть розробляти й зовнішні розробники.

3.4 Бібліотека PrimeNg

PrimeNG - це популярна відкрита бібліотека компонентів інтерфейсу користувача для Angular додатків. Вона надає широкий набір готових до використання компонентів, які легко інтегруються в проекти Angular. PrimeNG пропонує великий вибір компонентів, включаючи вводи форм, таблиці даних, календарі, діаграми, діалогові вікна, меню та багато інших. Ці компоненти мають сучасний і адаптивний дизайн, надаючи однорідний та зручний інтерфейс для веб-додатків. PrimeNG також підтримує різноманітні опції налаштування, теми та можливості інтернаціоналізації. Його широко використовують розробники для покращення користувацького досвіду та прискорення процесу розробки в Angular додатках.

Бібліотека PrimeNG була використана у розробленій програмі з декількох причин:

1. Широкий набір готових компонентів: PrimeNG надає багато готових до використання компонентів, які значно спрощують розробку інтерфейсу користувача. Завдяки цьому, розробники можуть швидко додавати функціональність, таку як форми вводу, таблиці даних, діалогові вікна та інші, без необхідності писати код з нуля. Це дозволяє економити час і зусилля при розробці програми.
2. Сучасний та адаптивний дизайн: Компоненти PrimeNG мають сучасний дизайн, що дозволяє створювати привабливий та професійний вигляд програмного інтерфейсу. Вони також адаптивні, що означає, що їх можна легко пристосувати до різних розмірів екранів і пристроїв. Це забезпечує зручне користувацьке досвід незалежно від пристрою, на якому використовується програма.
3. Налаштування і теми: PrimeNG дозволяє розробникам налаштовувати компоненти відповідно до вимог і дизайну своєї програми. Вона надає багато опцій налаштування, які дозволяють змінювати вигляд, поведінку та функціональність компонентів. Крім того, PrimeNG має різноманітні

теми, які дозволяють швидко змінювати зовнішній вигляд програми, забезпечуючи їй унікальний стиль.

Загалом, використання бібліотеки PrimeNG у розробленій програмі дозволило спростити розробку інтерфейсу користувача, забезпечити сучасний та адаптивний дизайн, а також налаштовувати компоненти відповідно до потреб програми. Це дозволило зосередитись на функціональності програми та покращити користувацький досвід.

3.5 База даних Firebase

Firebase - це платформа для розробки веб- та мобільних додатків, яка надає інструменти та сервіси для зручної і швидкої реалізації різноманітних функцій. Вона пропонує набір хмарних послуг, які включають в себе базу даних, аутентифікацію, зберігання файлів, хостинг, розсилку повідомлень та багато іншого.[2]

Одним з ключових компонентів Firebase є Firebase Realtime Database, який є хмарною NoSQL базою даних.[1] Вона забезпечує зручний доступ до даних у реальному часі, що означає, що зміни, внесені в одному пристрої, автоматично синхронізуються з іншими підключеними пристроями без необхідності оновлення сторінки або додатка. Це робить Firebase Realtime Database ідеальним варіантом для розробки додатків, які потребують реалізації спільної роботи декількох користувачів, обміну повідомленнями або відстеження змін у режимі реального часу.

Крім того, Firebase надає широкий набір інших сервісів, які спрощують розробку програмного забезпечення. Наприклад, Firebase Authentication дозволяє реалізувати систему аутентифікації користувачів з різними методами входу, такими як електронна пошта, соціальні мережі та інші. Firebase Storage дозволяє зберігати та керувати файлами, такими як зображення або відео. Firebase Cloud Messaging дозволяє відправляти повідомлення на мобільні пристрої. Крім того, Firebase має інтеграцію з іншими сервісами Google, що розширює можливості розробки, такі як Google Analytics, Google Ads, Google Cloud Platform і багато інших.[3]

В цілому, Firebase є потужним інструментом для розробників, який пропонує широкий спектр сервісів із зручними інтерфейсами програмування додатків (API), що

дозволяє прискорити розробку, покращити функціональність та забезпечити надійність веб- та мобільних додатків.

Firebase було використано у розробленій програмі з кількох причин:

1. Реальний час: Firebase Realtime Database забезпечує можливість синхронізації даних в реальному часі. Це дозволяє користувачам бачити оновлення даних без необхідності оновлення сторінки або перезавантаження додатка. Це особливо важливо для додатків, які мають спільну роботу декількох користувачів або потребують швидкої передачі даних.
2. Зручність розробки: Firebase надає розробникам зручні інструменти та SDK, що спрощують процес розробки. Інтегровані сервіси, такі як аутентифікація користувачів, зберігання файлів, аналітика та розсилка повідомлень, забезпечують швидку і готову до використання функціональність, що дозволяє розробникам зосередитись на основній логіці додатку.
3. Масштабованість: Firebase має вбудовану масштабованість, що дозволяє додатку зростати разом з його користувацькою базою. База даних автоматично масштабується для виконання вимог навантаження, що робить Firebase ідеальним варіантом для додатків зі зростаючою кількістю користувачів.
4. Готовість для мобільних платформ: Firebase надає SDK для різних мобільних платформ, включаючи Android та iOS. Це дозволяє розробникам легко інтегрувати Firebase у свої мобільні додатки і отримати доступ до всіх його функціональних можливостей.

Загалом, використання Firebase у розробленій програмі дозволило забезпечити швидку і ефективну обробку даних, реалізацію реального часу та простоту розробки, що сприяє покращенню функціональності та користувацького досвіду.

Висновки до розділу 3

У даному розділі було розглянуто засоби розробки, які були використані при створенні системи для вибору вибіркового дисциплін. В результаті проведеного аналізу і вибору засобів, було прийнято рішення використовувати такі інструменти та технології:

1. Angular: Цей фреймворк був обраний для розробки клієнтської частини системи. Він надає широкі можливості для створення ефективного та масштабованого веб-інтерфейсу.
2. Firebase: Для зберігання та синхронізації даних між користувачами була використана база даних Firebase Realtime Database. Ця хмарна база даних забезпечує реальний час оновлення даних та оптимізована для використання в мобільних додатках.
3. PrimeNG: Ця бібліотека компонентів була використана для швидкої розробки інтерфейсу користувача. Вона надає готові компоненти, такі як таблиці, форми, кнопки, що спрощує процес розробки і покращує взаємодію з користувачем.
4. WebStorm: Інтегроване середовище розробки WebStorm було обране для написання програмного коду. Воно забезпечує потужні функції редагування, автодоповнення, налагодження та інші інструменти, що сприяють продуктивній розробці програмного забезпечення.

Використання цих засобів дозволило ефективно розробити систему для вибору вибіркового дисциплін зручним та функціональним інтерфейсом, а також забезпечити надійне зберігання та синхронізацію даних.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Розроблена система є веб-додатком, призначеним для вибору вибіркового дисциплін. Dodatok розроблено з використанням Angular та Firebase. Користувачі мають можливість створювати свій профіль, шукати дисципліни за критеріями, такими як назва, опис, викладач та інші, а адміністратори можуть додавати дисципліни до списку вибіркового дисциплін.

Dodatok використовує базу даних Firebase для зберігання даних користувачів, дисциплін та інших даних, що необхідні для його функціонування. Крім того, dodatok використовує можливості автентифікації та авторизації Firebase для захисту користувачів та їх даних.

Система має простий та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко знайти необхідну дисципліну та додати її до свого списку вибіркового дисциплін. Крім того, dodatok надає можливість переглядати опис дисципліни, дивитися її розклад та інші корисні дані.

Загалом, система є простою та зручною у використанні, що допомагає студентам знайти та вибрати необхідні вибірково дисципліни для свого навчання.

4.1 Модель бази даних

У моделі бази даних для розробленої системи використовується Firestore Database для зберігання та синхронізації даних в реальному часі. Firestore Database працює з даними у форматі JSON та надає можливість розподіленого доступу до них.

Структура моделі бази даних включає такі ключові елементи:

1. Колекції: Колекції відповідають основним сутностям додатку, таким як "дисципліни", та "користувачі". На рисунку 4.1 показано колекції розробленої системи.

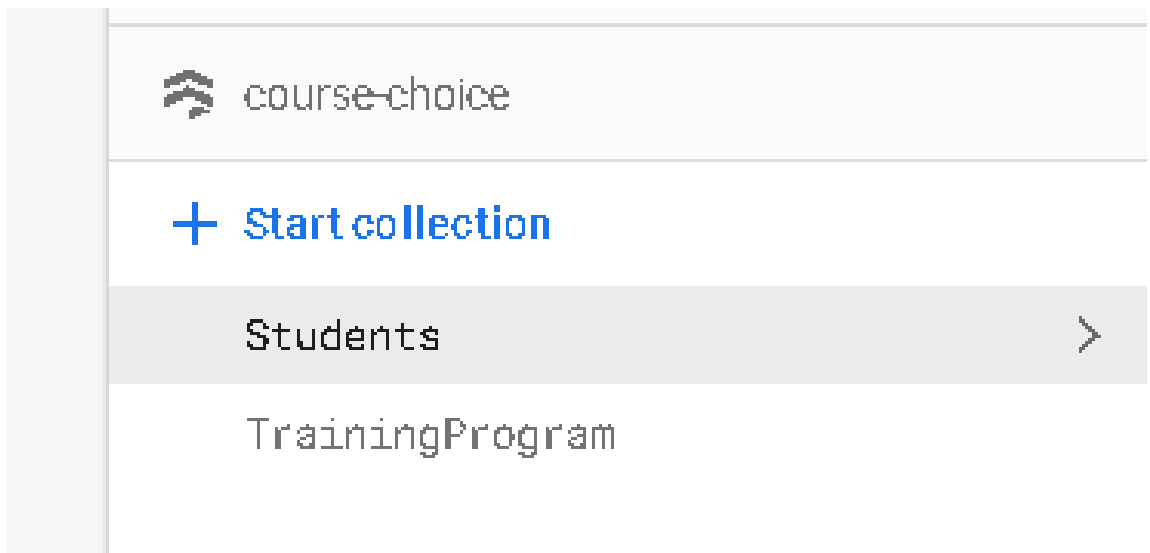


Рисунок 4.1 – Колекції

Кожна колекція містить набір документів, які представляють конкретні екземпляри цих сутностей. На рисунку 4.2 показано документи колекції Students.

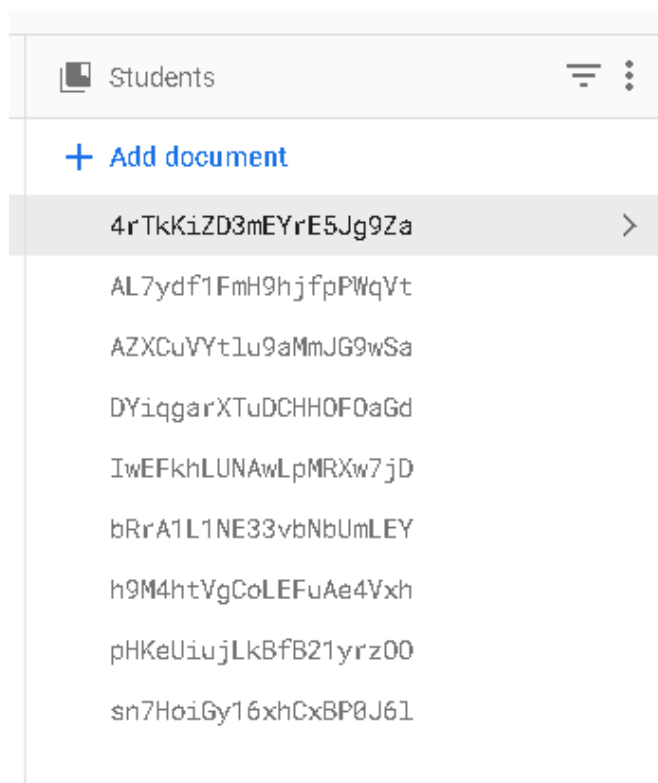


Рисунок 4.2 – Документи колекції Students

2. Документи: Кожен документ у колекції містить дані, представлені у форматі JSON. Наприклад, документ дисципліни може містити поля, такі як "назва", "опис", "викладач" та інші атрибути, що описують дисципліну. Документ

користувача може містити поля, такі як "ім'я", "прізвище", "email" та інші відповідні атрибути. На рисунку 4.3 показано приклад даних документу з колекції Students.

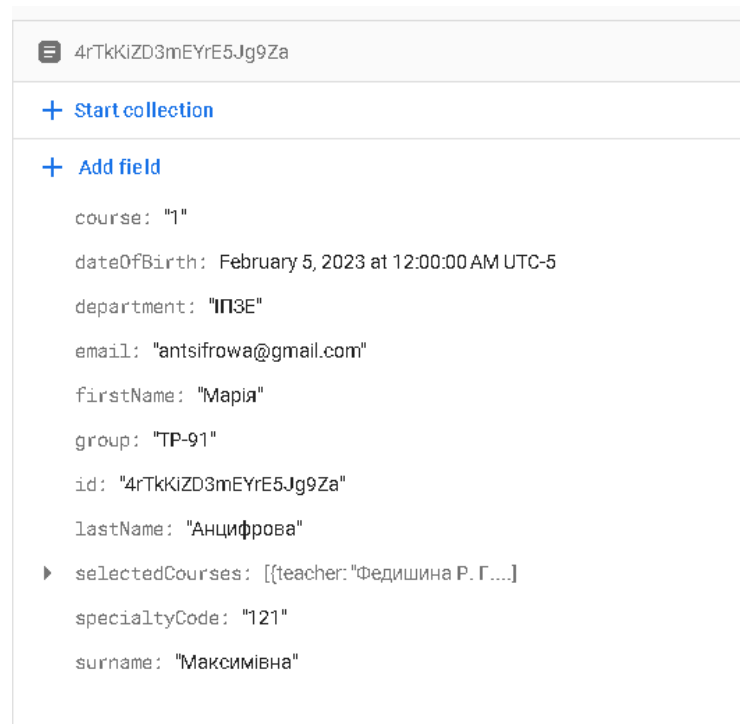


Рисунок 4.3 – Дані документу з колекції Students

На рисунку 4.4 показано приклад даних документу з колекції TrainingProgram.

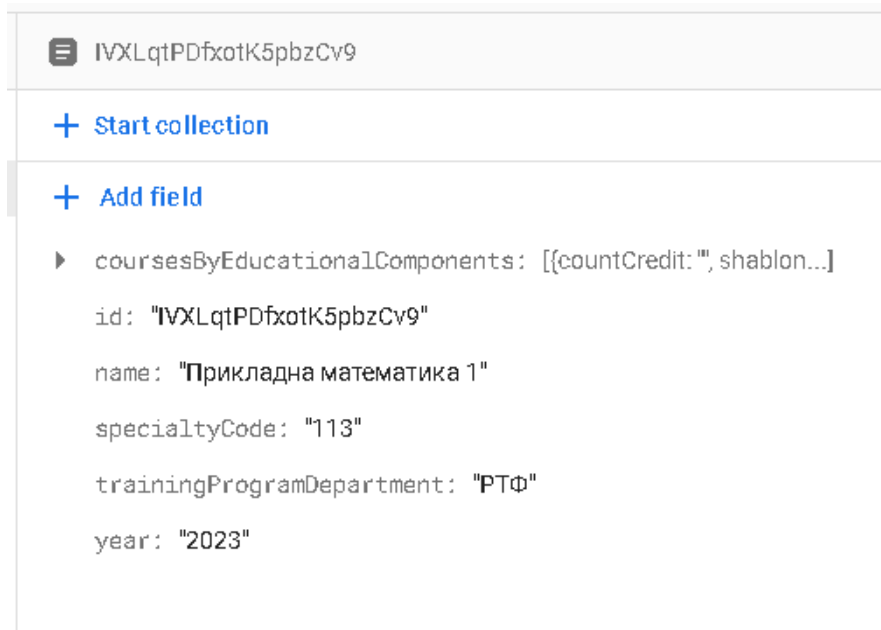


Рисунок 4.4 – Дані документу з колекції TrainingProgram

Висновки до розділу 4

У даному розділі було надано детальний опис програмної реалізації системи для вибору вибіркового дисциплін. В результаті розробки було отримано наступні висновки:

1. Розроблена система демонструє повноту та відповідність вимогам. Вона забезпечує зручний та інтуїтивно зрозумілий інтерфейс для користувачів, що дозволяє легко вибрати вибірково дисципліни зі списку доступних опцій.
2. Реалізована система підтримує авторизацію та аутентифікацію користувачів. Це забезпечує захист даних та забезпечує доступ до функціональності системи тільки автентифікованим користувачам.
3. Використання бази даних Firebase Database дозволяє зберігати та синхронізувати дані в реальному часі між різними пристроями користувачів. Це дозволяє користувачам бачити оновлені дані та спрощує процес обміну даними.
4. Розроблена система є масштабованою та гнучкою, що дозволяє легко розширювати її функціональність та впроваджувати нові можливості в майбутньому.

Загалом, програмна реалізація системи для вибору вибіркового дисциплін є успішною, відповідає вимогам та забезпечує зручну та ефективну роботу користувачів.

5 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

5.1 Системні вимоги

Розроблена програмна система є веб-додатком, який можна використовувати в браузері без необхідності встановлення на комп'ютері.

5.2 Реєстрація та вхід у систему

Для початку роботи з додатком користувач повинен зайти на веб-сайт і увійти в обліковий запис. На рисунку 5.1 зображено форму входу до кабінету.

The image shows a login form with the following elements:

- Title: **Авторизація**
- Label: Електронна пошта
- Input field for email
- Label: Пароль
- Input field for password
- Button: Увійти
- Links: [Реєстрація](#) and [Забули пароль?](#)

Рисунок 5.1- Форма входу до кабінету

Кнопка “Увійти” спочатку не працює, а коли користувач вводить необхідні дані, вона стає доступною. На рисунку 5.2 зображено заповнену форму входу до кабінету.

Авторизація

Електронна пошта

nastyapashkevich10@gmail.com

Пароль

Увійти

[Реєстрація](#)
[Забули пароль?](#)

Рисунок 5.2- Заповнена форма входу до кабінету

Якщо у користувача ще немає облікового запису, він може зареєструватися на сайті, ввівши необхідні дані. На рисунку 5.3 та 5.4 зображено форму реєстрації.

Реєстрація

Прізвище *

Семенов

Ім'я *

Семен

По батькові *

Семенович

Дата народження

04/14/2002

Кафедра *

Оберіть кафедру

Рисунок 5.3 - Форма реєстрації

Код спеціальності *

122

Курс *

Оберіть курс

Група

ІР-В1

Електронна пошта *

student@gmail.com

Пароль *

Зареєструватися

[Вже є акаунт?](#)

Рисунок 5.4 - Форма реєстрації (продовження)

Якщо користувач забув пароль, він може скористатися формою і ввести свою адресу електронної пошти, на яку йому прийде посилання для зміни пароля. На рисунку 5.5 зображено форму для зміни пароля .

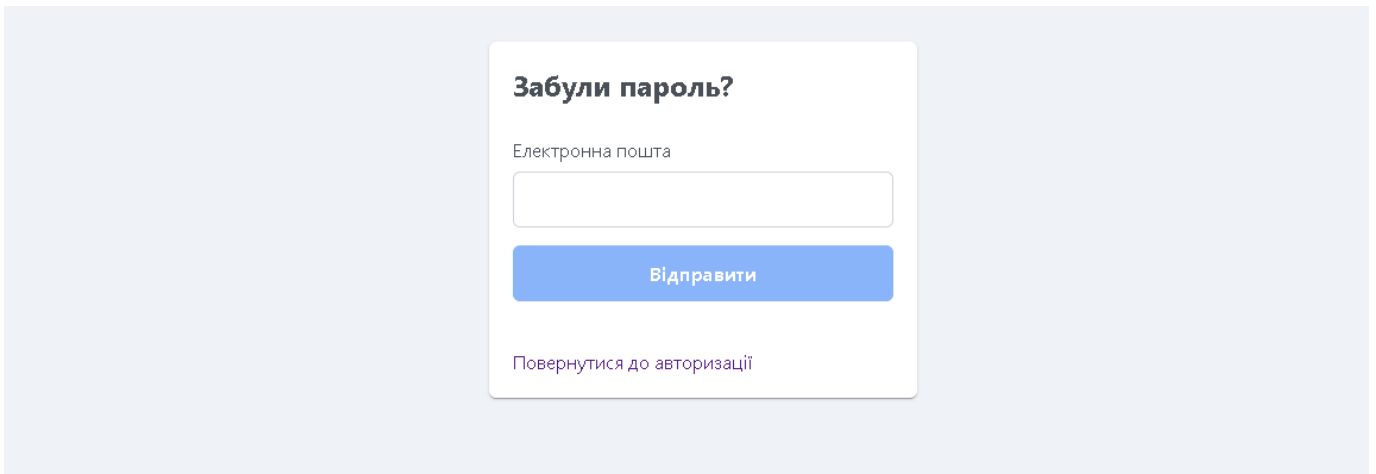


Рисунок 5.5 Форма для зміни пароля

5.3 Головна сторінка програмного продукту

Після реєстрації користувач потрапляє на головну сторінку де розповідається про додаток та його функції. На рисунку 5.6 зображено головну сторінку.

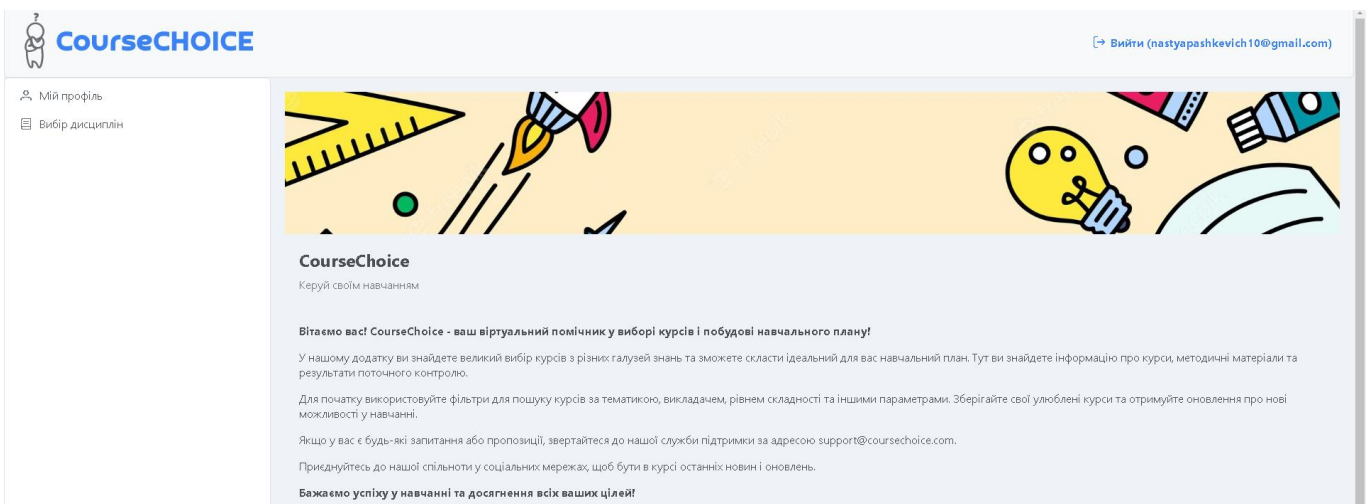


Рисунок 5.6 - Головна сторінка

5.4 Приклади роботи користувача з програмним продуктом

Користувач може подивитися інформацію про свій профіль, та змінити її, якщо потрібно. На рисунку 5.7 зображено профіль користувача.

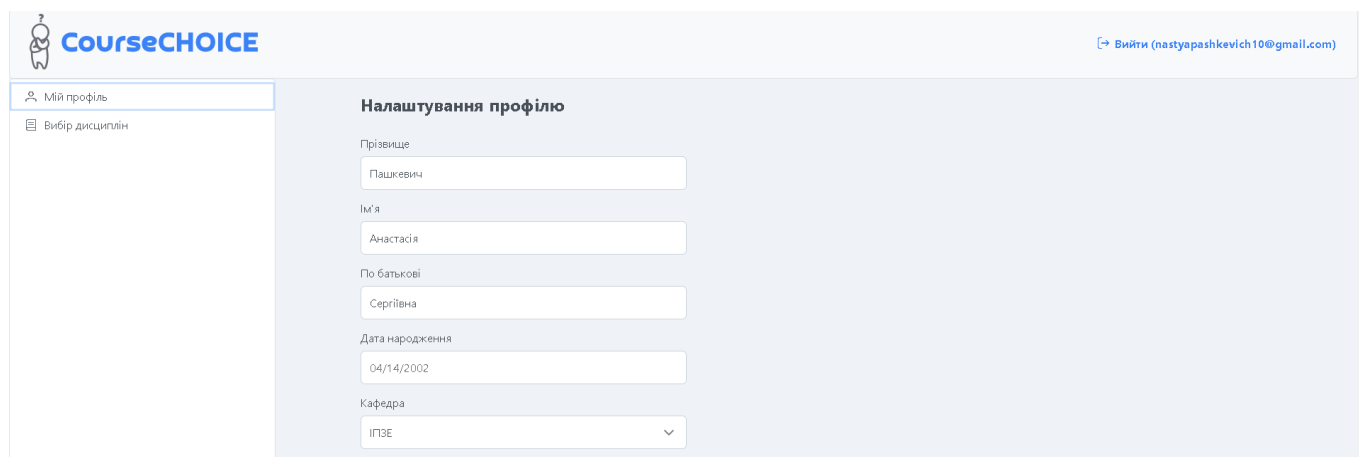
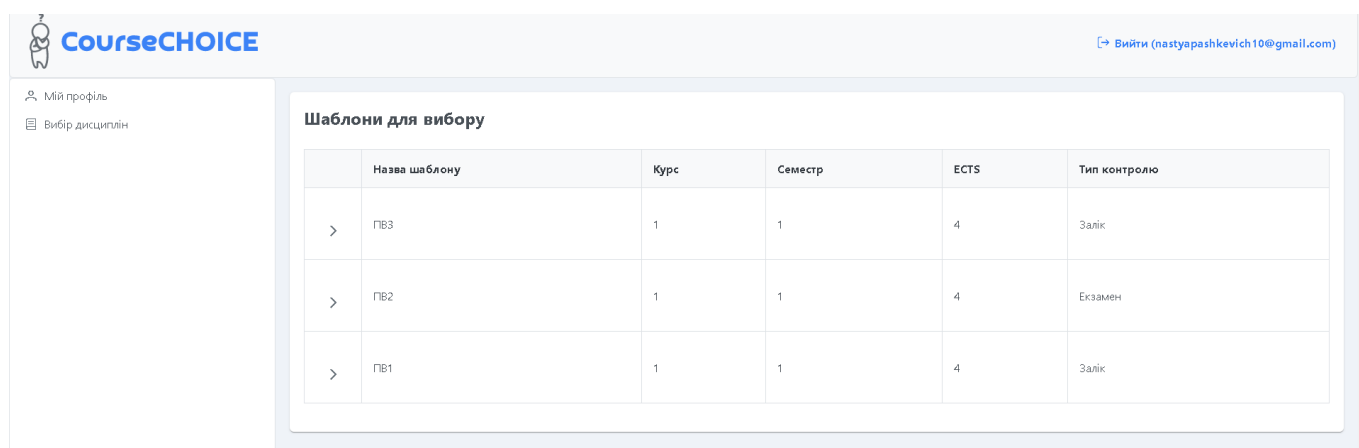


Рисунок 5.7 - Профіль користувача

Користувач може переглянути доступні вибіркові дисципліни на наступний семестр. Кожна дисципліна має назву, опис, та кількість місць, які є доступними для вибору.

Спочатку можна побачити список освітніх компонент. На рисунку 5.8 зображено список освітніх компонент .



	Назва шаблону	Курс	Семестр	ECTS	Тип контролю
>	ПВ3	1	1	4	Залік
>	ПВ2	1	1	4	Екзамен
>	ПВ1	1	1	4	Залік

Рисунок 5.8 - Список освітніх компонент

Натиснувши на стрілочку вниз можна розгорнути освітню компоненту та побачити список дисциплін для вибору. На рисунку 5.9 зображено список дисциплін.

Шаблони для вибору

	Назва шаблону	Курс	Семестр	ECTS	Тип контролю
▼	ПВ3	1	1	4	Залік

Назва дисципліни	Залишилось місць	Читає кафедра	URL	Вибір
Моделювання та оптимізація систем управління	60	ІХФ		+ Записатися
Основи операційної системи Linux	60	ІХФ		+ Записатися
Операційні системи та комп'ютерні мережі	60	ІХФ		+ Записатися

Рисунок 5.9 - Список дисциплін

Щоб записатися на курс потрібно натиснути на кнопку “Записатися”. Тоді обраний курс додається у верх сторінки у список обраних дисциплін, а освітня компонента видаляється зі списку. На рисунку 5.10 зображено вибір дисциплін.

Обрані дисципліни

Назва шаблону	Назва дисципліни	Залишилось місць	Читає кафедра	URL	Дія
ПВ3	Основи операційної системи Linux	60	ІХФ		X Відмінити вибір

Шаблони для вибору

	Назва шаблону	Курс	Семестр	ECTS	Тип контролю
>	ПВ2	1	1	4	Екзамен
>	ПВ1	1	1	4	Залік

Рисунок 5.10 – Вибір дисципліни

Коли користувач обрав дисципліну з кожної освітньої компоненти список з дисциплінами для вибору зникає, залишається лише список обраних дисциплін. На рисунку 5.11 зображено обрані дисципліни.

The screenshot shows the 'Обрані дисципліни' (Selected disciplines) section of the CourseCHOICE application. It features a table with the following data:

Назва шаблону	Назва дисципліни	Залишилось місць	Читає кафедра	URL	Дія
ПВ3	Основи операційної системи Linux	60	ІХФ		<input type="button" value="X Відмінити вибір"/>
ПВ2	Апаратне забезпечення комп'ютерних систем	60	РТФ		<input type="button" value="X Відмінити вибір"/>
ПВ1	Технології відновленої енергетики	60	ІХФ		<input type="button" value="X Відмінити вибір"/>

Рисунок 5.11 - Обрані дисципліни

Користувач може відмінити вибір дисципліни, натиснувши на кнопку "Відмінити вибір" поруч з назвою дисципліни. Після цього кількість доступних місць для вибору збільшиться, та користувач зможе вибрати іншу дисципліну. Та користувач зможе обрати іншу дисципліну.

Усі дані про вибрані дисципліни зберігаються в базі даних і можуть бути переглянуті в будь-який час користувачем.

Додаток також забезпечує безпеку даних шляхом шифрування і захисту від несанкціонованого доступу. Користувач повинен дотримуватися правил безпеки, таких як складний пароль для входу в обліковий запис, не надавати свої дані третім особам та не залишати комп'ютер без нагляду під час роботи з додатком.

У разі виникнення будь-яких проблем з додатком або потреби отримати додаткову інформацію користувач може звернутися до розділу підтримки на веб-сайті. Там він зможе знайти відповіді на питання щодо роботи з додатком, а також звернутися до технічної підтримки за допомогою спеціальної форми зв'язку.

Усі процеси вибору та реєстрації дисциплін у додатку здійснюються автоматично, що дозволяє значно економити час та зробити процес навчання більш зручним та доступним для користувачів.

Для адміністратора доступні додаткові функції, такі як додавання нових дисциплін, редагування та видалення дисциплін та перегляд інформації про обрані студентами дисципліни.

Спочатку адміністратор бачить список програм підготовки. На рисунку 6.3 зображено список програм підготовки. На рисунку 5.12 зображено список програм підготовки.

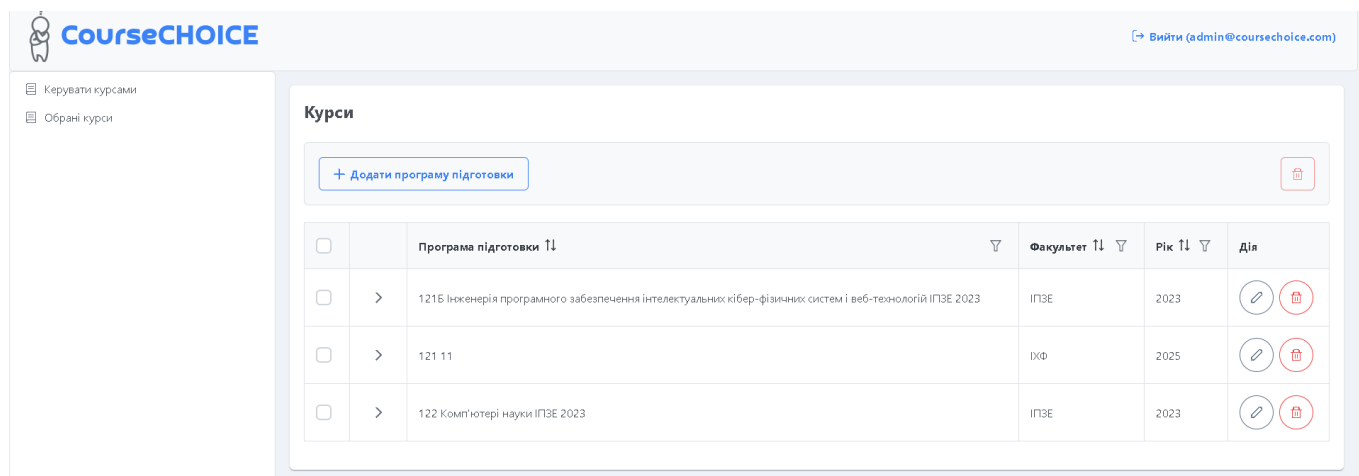


Рисунок 5.12 - Список програм підготовки

Натиснувши на стрілочку вниз можна побачити освітні компоненти обраної програми підготовки. На рисунку 5.13 зображено список освітніх компонент.

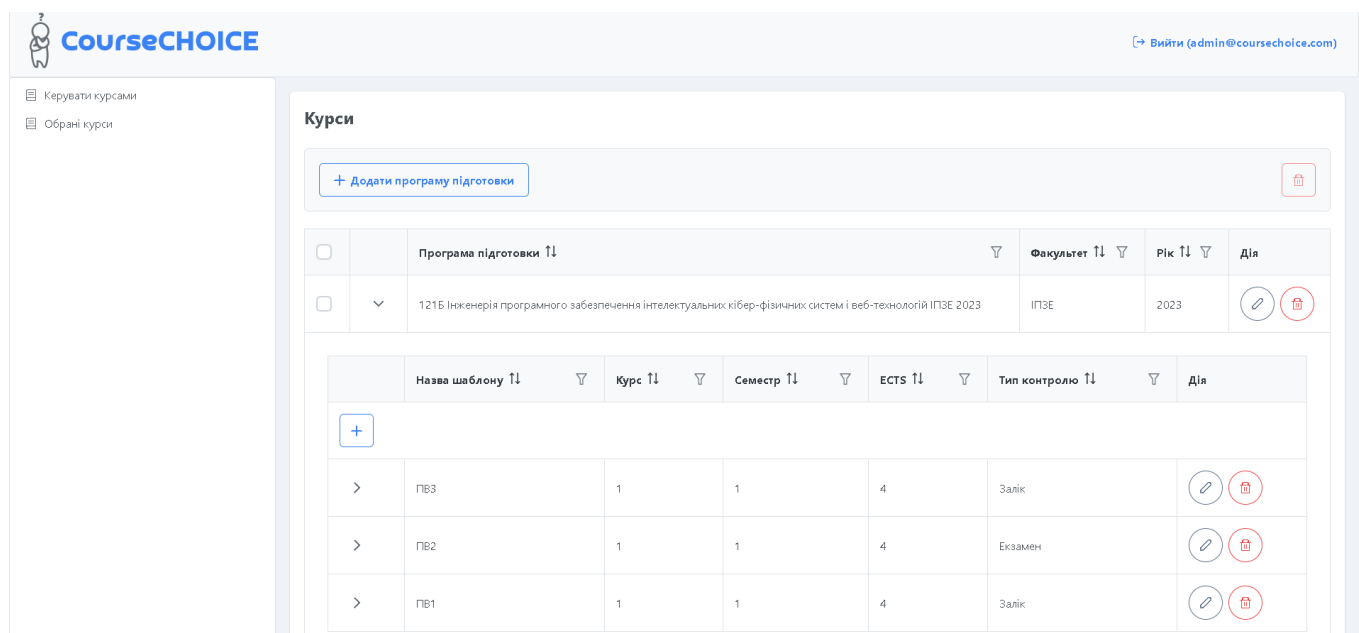


Рисунок 5.13 - Список освітніх компонент

Ще раз натиснувши на стрілочку вниз можна побачити дисципліни обраної освітньої компоненти. На рисунку 5.14 зображено список дисциплін.

<input type="checkbox"/>	Програма підготовки ↑↓	Факультет ↑↓	Рік ↑↓	Дія
<input type="checkbox"/>	121Б Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем і веб-технологій ІПЗЕ 2023	ІПЗЕ	2023	

<input type="checkbox"/>	Назва шаблону ↑↓	Курс ↑↓	Семестр ↑↓	ECTS ↑↓	Тип контролю ↑↓	Дія
<input type="checkbox"/>	ПВЗ	1	1	4	Залік	

<input type="checkbox"/>	Назва дисципліни ↑↓	Залишилось місць ↑↓	Читає кафедра ↑↓	URL	Дія
<input type="checkbox"/>	Моделювання та оптимізація систем управління	60	ІХФ		
<input type="checkbox"/>	Основи операційної системи Linux	60	ІХФ		
<input type="checkbox"/>	Операційні системи та комп'ютерні мережі	60	ІХФ		

Рисунок 5.14 - Список дисциплін

Адміністратор може додавати нові програми підготовки. Для цього потрібно ввести назву програми підготовки, факультет, рік та код спеціальності. На рисунку 5.15 зображено форма програми підготовки.

The screenshot shows the CourseCHOICE web application. A modal window titled 'Програма підготовки' (Program Preparation) is open over a list of courses. The modal contains the following elements:

- Title:** Програма підготовки
- Image:** An illustration of a person presenting to a group of people sitting at a table.
- Form Fields:**
 - Назва (Name):** A text input field.
 - Факультет (Faculty):** A dropdown menu with the text 'Оберіть факультет' (Select faculty).
 - Рік (Year):** A dropdown menu with the text 'Оберіть рік' (Select year).
- Buttons:** 'Відмінити' (Cancel) and 'Додати' (Add).

In the background, a table of courses is visible with columns for 'Програма', 'Факультет', 'Рік', and 'Дія'. The table contains three rows of data.

Рисунок 5.15 - Форма програми підготовки

Адміністратор може додавати нові освітні компоненти. Для цього потрібно ввести назву освітньої компоненти, курс, семестр та тип контролю. На рисунку 5.16 зображено форма освітньої компоненти.

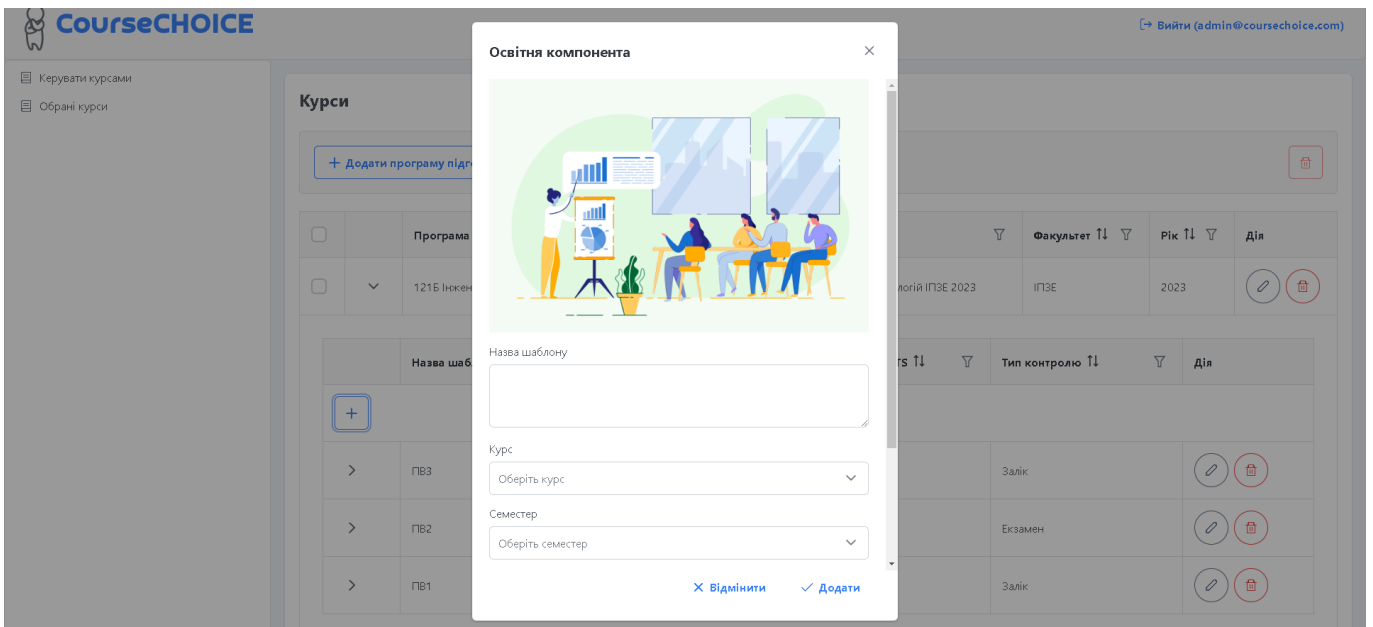


Рисунок 5.16 - Форма освітньої компоненти

Адміністратор може додавати нові дисципліни. Для цього потрібно ввести назву дисципліни, кількість місць та кафедру. На рисунку 5.17 зображено форма дисципліни.

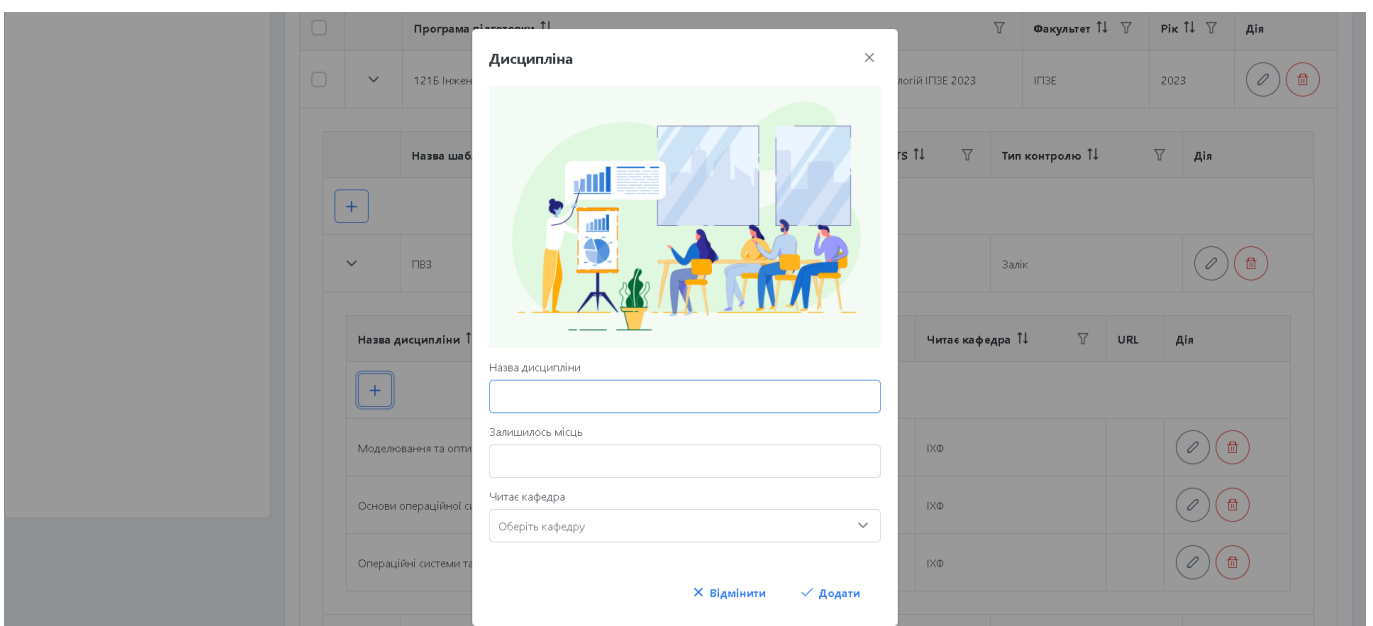


Рисунок 5.17 - Форма дисципліни

Крім того, адміністратор може видаляти програми підготовки, освітні компоненти та дисципліни. Для цього потрібно натиснути на значок “Смітєвого баку” поруч з елементом, який потрібно видалити. Після цього він побачить модальне вікно, на якому запитується чи справді він хоче видалити елемент. На рисунку 5.18 зображено видалення програми підготовки.

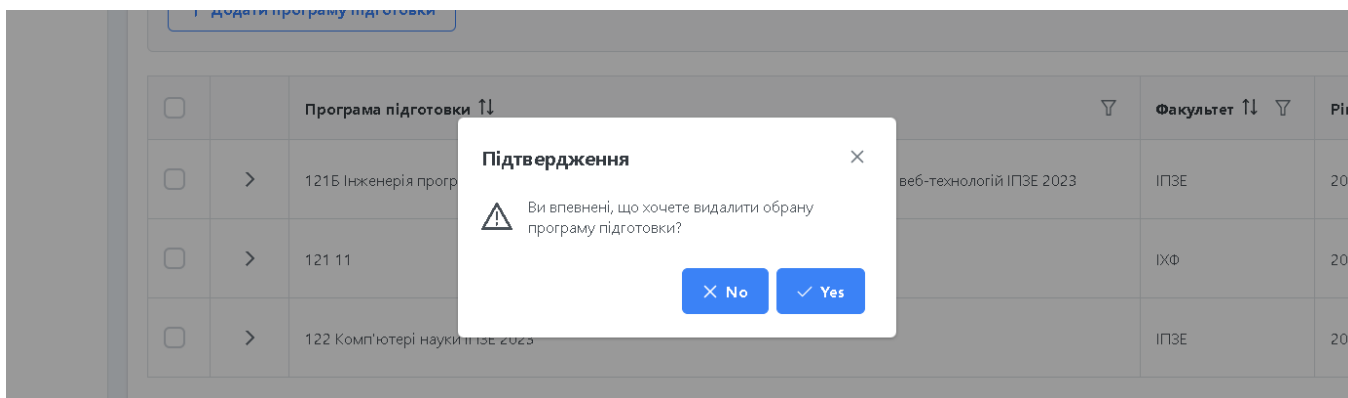


Рисунок 5.18 - Видалення програми підготовки

Якщо натиснути “Так” обраний елемент видалиться та з’явиться повідомлення про те, що елемент видалено. На рисунку 5.19 зображено повідомлення про успішне видалення.

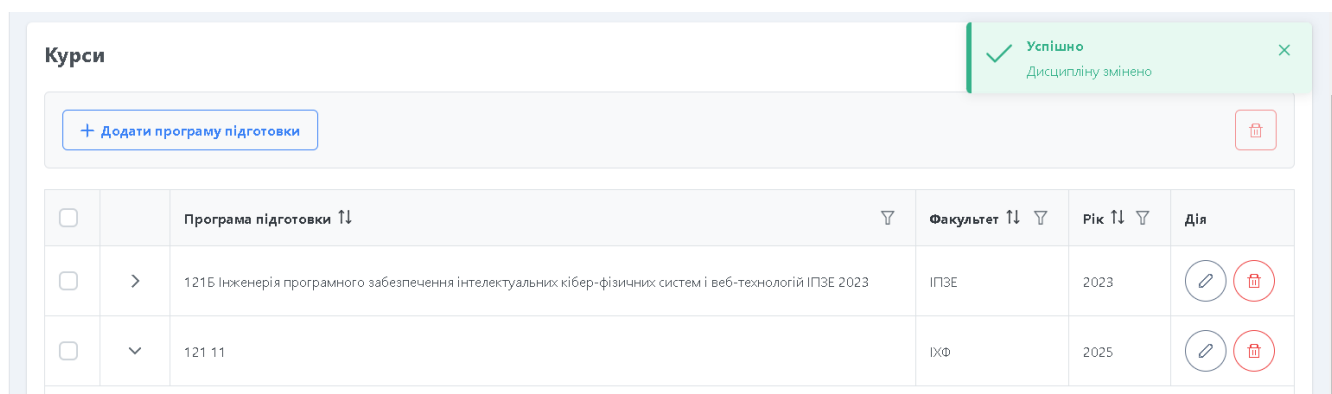


Рисунок 5.19 – Повідомлення про успішне видалення

Крім того, адміністратор може редагувати програми підготовки, освітні компоненти та дисципліни.

Адміністратор також може переглядати статистику вибору дисциплін студентами. Він може переглядати загальну кількість студентів, які зробили вибір, та кількість виборів кожної дисципліни. Також є можливість сортувати та фільтрувати дані таблиці. На рисунку 5.20 зображено обрані дисципліни.

Обрані дисципліни						
Прізвище Ім'я По- батькові ↑↓	Обрані дисципліни	Група ↑↓	Факультет ↑↓	Курс ↑↓	Код спеціальності ↑↓	Email ↑↓
<input type="text"/>		<input type="text"/>	Any	<input type="text"/>	<input type="text"/>	<input type="text"/>
Анастасія Пашкевич Сергіївна	ПВ3 - Моделювання та оптимізація систем управління ПВ2 - Апаратне забезпечення комп'ютерних систем ПВ1 - Математичне моделювання в теплоенергетиці	ТВ-91	ІПЗЕ	1	121	nastyapashkevich10@gmail.com

Рисунок 5.20 - Обрані дисципліни

Отже, додаток забезпечує зручний та безпечний спосіб вибору дисциплін для навчання, що дозволяє студентам швидко та легко обирати необхідні предмети та починати вивчати їх, а адміністраторам керувати дисциплінами.

Висновки до розділу 5

У даному розділі була детально описана робота користувачів з мобільним додатком для вибору вибіркового дисциплін. Показано, що додаток забезпечує зручний та безпечний спосіб вибору дисциплін для навчання, що дозволяє студентам швидко та легко обирати необхідні предмети та починати вивчати їх, а адміністраторам керувати дисциплінами.

ВИСНОВКИ

В ході проходження практики було розроблено систему для автоматизації процесу вибору дисциплін студентами в університеті. Система дозволяє студентам переглядати доступні дисципліни на наступний семестр, вибирати дисципліни та переглядати свої обрані дисципліни. Додаток також забезпечує безпеку даних.

Було проаналізовано існуючі програмні рішення схожої направленості та набуті навички практичного використання засобів розробки.

Було створено додаток, що працює на основі веб-технологій, до якого можна отримати доступ з пристрою, що має доступ до мережі інтернет через браузер.

У системі є функціонал для адміністратора, що дозволяє додавати нові дисципліни, керувати студентами та переглядати звіти про вибрані дисципліни студентами.

Для розробки веб-додатку було використано TypeScript - мову програмування зі строгою типізацією, що допомагає попередити помилки на етапі компіляції. Також використовувався Angular - фреймворк з багатим набором функцій. В якості бази даних та серверу використовувався Firebase - сервіс хмарового сховища даних, що надає API для доступу до даних з клієнтського застосунку.

Результатом проекту є готовий до використання додаток, що дозволить університету полегшити процес вибору дисциплін студентами та зменшити ручну роботу адміністраторів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Організація баз даних: практичний курс: Навч. посіб. для студ. / А. Ю. Берко, О. М. Верес; Нац. ун-т «Львів. політехніка». — Л., 2003. — 149 с.
2. Советов Б. Я. Базы данных / Б. Я. Советов, В. В. Цехановський, В. Д. Чертовський. — Санкт-Петербург, 2015. — 420 с.
3. Лаврищева К. М. Підручник «Програмна інженерія» / Катерина Михайлівна Лаврищева. — Київ, 2008. — 90 с.
4. Angular [Електронний ресурс] — Режим доступу до ресурсу: <https://angular.io/docs>;
5. Node.js [Електронний ресурс] — Режим доступу до ресурсу: <https://nodejs.org/uk/docs/>;
6. Wilken J. Angular in Action / Jeremy Wilken. — Shelter Island, New York: Manning Publications, 2018. — 320 с.
7. Файн Я. Angular и TypeScript. Сайтостроение для профессионалов / Я. Файн, А. Моисеев — СПб.: Питер, 2018. — 464 с.
8. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е изд. / Р. Никсон — СПб.: Питер, 2016. — 768 с.
9. Закас Н. JavaScript для профессиональных веб-разработчиков / Н. Закас — СПб.: Питер, 2015. — 960 с.
10. Clow M. Angular 5 Projects: Learn to Build Single Page Web Applications Using 70+ Projects / M. Clow — Apress, 2018. — 465 с.

ДОДАТОК Б

Додаток для вибору вибіркових дисциплін

Текст програмного модулю

Аркушів 5

Київ - 2023

Точка входу програми

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

Головний маршрутизатор програми

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

import { CourseChoiceComponent } from './course-choice/course-choice.component';
import { PageNotFoundComponent } from './page-not-found/page-not-found.component';
import { HomeComponent } from './home/home.component';
import { UserProfileComponent } from './user-profile/user-profile.component';
import { DashboardComponent } from './dashboard/dashboard.component';
import { LoginComponent } from './login/login.component';
import { RegisterComponent } from './register/register.component';
import { VerifyEmailComponent } from './verify-email/verify-email.component';
import { ForgotPasswordComponent } from './forgot-password/forgot-
password.component';
import { CoursesComponent } from './courses/courses.component';
import { SelectedCoursesComponent } from './selected-courses/selected-
courses.component';
import { LogoutComponent } from './logout/logout.component';

const routes: Routes = [
  {path: '', redirectTo: 'login', pathMatch: 'full'},
  {path: 'login', component: LoginComponent},
  {path: 'logout', component: LogoutComponent},
  {path: 'register', component: RegisterComponent},
  {path: 'forgot-password', component: ForgotPasswordComponent},
  {path: 'verify-email', component: VerifyEmailComponent},
  {
    path: '',
    component: HomeComponent,
    children: [
      {path: 'dashboard', component: DashboardComponent},
```

```

    {path: 'profile', component: UserProfileComponent},
    {path: 'courses', component: CourseChoiceComponent},
    {path: 'manage-courses', component: CoursesComponent},
    {path: 'selected-courses', component: SelectedCoursesComponent}
  ]
},
{path: '**', component: PageNotFoundComponent},
];

```

```

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule {
}

```

ГОЛОВНИЙ КОМПОНЕНТ ПРОГРАМИ

```
import { Component, OnInit } from '@angular/core';
```

```

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
}

```

Сервіс авторизації

```

import {Injectable} from '@angular/core';
import {Router} from '@angular/router';
import {AngularFireAuth} from '@angular/fire/compat/auth';

```

```

@Injectable({
  providedIn: 'root'
})

```

```
export class AuthService {
```

```

  constructor(private router: Router, private fireauth: AngularFireAuth) {
  }

```

```

  public login(email: string, password: string): void {
    this.fireauth.signInWithEmailAndPassword(email, password).then((res) => {
      localStorage.setItem('token', 'true');

```

```

      this.router.navigate(['/dashboard']);
    }, (err) => {

```

```

    alert(err.message);
    this.router.navigate(['/login']);
  });
}

```

```

public register(email: string, password: string): void {
  this.fireauth.createUserWithEmailAndPassword(email, password).then((res) => {
    alert('Registration Successful');
    this.router.navigate(['/login']);
  }, (err) => {
    alert(err.message);
    this.router.navigate(['/register']);
  });
};

```

```

public logout(): void {
  this.fireauth.signOut().then(() => {
    localStorage.removeItem('token');
    this.router.navigate(['/login']);
  }, (err) => {
    alert(err.message);
  })
}

```

```

public forgotPassword(email: string): void {
  this.fireauth.sendPasswordResetEmail(email).then(() => {
    this.router.navigate(['/verify-email']);
  }, (err) => {
    alert('Щось пішло не так ...!');
  });
};
}

```

Сервіс для робот з даними

```

import {Injectable} from '@angular/core';
import {AngularFirestore} from '@angular/fire/compat/firestore';
import {Student} from '../models/student.model';
import {CourseByEducationalComponent} from "../models/course-by-educational-
component.model";
import {TrainingProgram} from "../models/training-program.model";

```

```

@Injectable({
  providedIn: 'root'
})
export class DataService {

```

```

constructor(private afs: AngularFirestore) {
}

addStudent(student: Student) {
  student.id = this.afs.createId();
  return this.afs.collection('/Students').add(student);
}

addTrainingProgram(trainingProgram: any) {
  if (!trainingProgram.id) {
    trainingProgram.id = this.afs.createId();
  }

  trainingProgram.coursesByEducationalComponents.forEach((courseByEducationalComponent: CourseByEducationalComponent) => {
    courseByEducationalComponent.id = this.afs.createId();

    courseByEducationalComponent.courses.forEach((course) => {
      course.id = this.afs.createId();
    });
  });

  return this.afs.collection('/TrainingProgram').add(trainingProgram);
}

updateTrainingProgram(trainingProgram: TrainingProgram) {
  return
  this.afs.collection('/TrainingProgram').doc(trainingProgram.id).update(trainingProgram);
}

getAllStudents() {
  return this.afs.collection('/Students').snapshotChanges();
}

getAllTrainingPrograms() {
  return this.afs.collection('/TrainingProgram').snapshotChanges();
}

deleteTrainingProgram(trainingProgram: any) {
  return this.afs.doc('/TrainingProgram/' + trainingProgram.id).delete();
}

updateStudent(student: Student) {

```

```

return this.afs.collection('/Students').doc(student.id).update(student);
}

```

```

addCourseByEducationalComponent(trainingProgram: TrainingProgram) {
  trainingProgram.coursesByEducationalComponents.unshift({
    id: this.afs.createId(),
    shablonName: "",
    forCourse: "",
    forSemester: "",
    countCredit: "",
    semesterControlType: "",
    courses: [
      {
        id: this.afs.createId(),
        name: "",
        remainingSpots: "",
        department: "",
        url: "",
      }
    ],
  })
}

```

```

return
this.afs.collection('/TrainingProgram').doc(trainingProgram.id).update(trainingProgram);
}

```

```

addCourse(trainingProgram: TrainingProgram, courseByEducationalComponent:
CourseByEducationalComponent) {
  trainingProgram.coursesByEducationalComponents.forEach((item, index) => {
    if (item.id === courseByEducationalComponent.id) {
      trainingProgram.coursesByEducationalComponents[index].courses.unshift({
        id: this.afs.createId(),
        name: "",
        remainingSpots: "",
        department: "",
        url: "",
      });
    }
  })
}

```

```

return
this.afs.collection('/TrainingProgram').doc(trainingProgram.id).update(trainingProgram);
}
}

```