

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ

Навчально-науковий інститут автоматики, кібернетики та
обчислювальної техніки

Кафедра комп'ютерних наук та прикладної математики

«До захисту допущений»

Завідувач кафедри комп'ютерних наук
та прикладної математики

_____ Турбал Ю.В.

«_____» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розроблення інформаційної системи для освітніх програм»

Виконав: Коноплицький В. В.

(прізвище, ім'я, по батькові)

(підпис)

група КН-41

Керівник: професор Бомба А. Я.

(науковий ступінь, вчене звання, посада, прізвище та ініціали)

(підпис)

Національний університет водного господарства та природокористування

**Навчально-науковий інститут автоматичної, кібернетики та
обчислювальної техніки**

Кафедра комп'ютерних наук та прикладної математики

Рівень вищої освіти бакалавр

Галузь знань 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри
комп'ютерних наук та
прикладної математики
д.т.н., професор Турбал Ю.В.

«___» _____ 2023 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Коноплицькому Володимиру Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інформаційної системи для освітніх програм
керівник роботи професор Бомба Андрій Ярославович
затверджені наказом вищого навчального закладу від « 19 » квітня 2023 року
С №-449
2. Термін подання роботи студентом 25.05.2023
3. Вихідні дані до роботи технології, що необхідні для розробки інформаційної системи для освітніх програм
4. Зміст розрахунково-пояснювальної записки розділ I огляд предметної області, розділ II використані технології, розділ III практична реалізація
5. Перелік графічного матеріалу мультимедійна презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Розділ 1</i>	<i>Бомба А.Я., професор</i>	20.12.2022	20.12.2022
<i>Розділ 2</i>	<i>Бомба А.Я., професор</i>	05.01.2023	05.01.2023
<i>Розділ 3</i>	<i>Бомба А.Я., професор</i>	23.05.2023	23.05.2023

7. Дата видачі завдання 05.10.2022**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вивчення літератури за обраною тематикою	01.10.2021 – 31.11.2021	виконав
2	Формулювання постановки задачі	01.12.2021 – 07.12.2021	виконав
3	Вибір стеку технологій	30.12.2021 – 08.01.2022	виконав
4	Збір та аналіз даних	09.01.2022 – 10.02.2022	виконав
5	Програмна реалізація	11.02.2022 – 15.04.2022	виконав
6	Відлагодження програмного продукту	10.05.2022 – 13.05.2022	виконав
7	Загальні висновки до роботи	14.05.2022 – 23.05.2022	виконав
8	Підготовка звіту кваліфікаційної роботи	24.05.2022 – 07.06.2022	виконав
9	Підготовка мультимедійної презентації	08.06.2022 – 12.06.2022	виконав
10	Підготовка до виступу	13.06.2022 – 18.06.2022	виконав

Студент _____ (Коноплицький В.В)

Керівник кваліфікаційної роботи _____ (Бомба А.Я)

ЗМІСТ

РЕФЕРАТ.....	4
АНОТАЦІЯ	5
ВСТУП.....	6
РОЗДІЛ I ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Загальні відомості про інформаційні системи.....	7
1.2 Ключові поняття.....	8
1.3 Формулювання задачі	11
РОЗДІЛ II ВИКОРИСТАНІ ТЕХНОЛОГІЇ	13
2.1 Flutter та мова програмування Dart	13
2.1.1 Програмний каркас Flutter.....	13
2.1.2 Практичні застосунки мови Dart	15
2.2 Фреймворк Bootstrap.....	16
2.3 Середовище розробки Visual Studio Code.....	17
РОЗДІЛ III ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	18
3.1 Роль серверної і клієнтської частин у проєкті.....	18
3.2 Серверна частина проєкту.....	19
3.2.1 Фреймворк Flask та його особливості.....	19
3.2.2 Реалізація серверної частини на прикладі таблиці “Галузі знань” ..	20
3.3 Застосована база даних	22
3.4 Використана архітектура мобільного додатку	24
3.4.1 Bloc шаблон	24
3.4.2 Stateless та Stateful віджети	25
3.4.3 Використані сторонні пакети.....	26
3.5 Реалізація вебсайту з використанням Bootstrap table	26
3.6 Огляд інтерфейсу користувача вебсайту	28
3.6.1 Головна сторінка	28
3.6.2 Сторінка додавання елементів в таблицю	29
3.6.3 Сторінка редагування елементів в таблиці.....	30
3.7 Огляд інтерфейсу користувача мобільного додатку	31
3.7.1 Головна сторінка	31
3.7.2 Сторінка детальної інформації про навчальну програму.....	34
ВИСНОВОК	42
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	44

РЕФЕРАТ

Кваліфікаційна робота: 42 сторінки, 22 малюнки, 10 джерел.

Мета: Розробка інформаційної системи для освітніх програм.

Завдання:

- Створити план реалізації додатку (визначення технологій та функціоналу).
- Розробка функціоналу.
- Розробка інтуїтивно зрозумілого та привабливого користувацького інтерфейсу.

Результати: Розроблена інформаційна система для освітніх програм, яка надає зручний та ефективний користувацький інтерфейс, швидкий у освоєнні та функціонально потужний застосунок для користувачів.

Ключові слова: база даних, освітня система, серверна та клієнтська частина, вебсайт, мобільний додаток, Flutter, Dart, Bootstrap, Bootstrap Table, Flask.

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробленню інформаційної системи для освітніх програм з метою забезпечення зручного та ефективного доступу до інформації про освітні програми. У рамках роботи було реалізовано веб-сайт для адміністрування бази даних, яка містить повну інформацію про освітні програми. Сайт надає можливість зручного та ефективного керування даними, додавання та редагування інформації про галузі знань, спеціальності та освітні програми та їх складові.

Також у рамках роботи був розроблений мобільний додаток, який дозволяє користувачам переглядати освітні програми та отримувати детальну інформацію про них. Додаток забезпечує зручний та доступний інтерфейс для користувачів, де вони можуть швидко знайти необхідну інформацію про різні освітні програми.

Результатом роботи є інформаційна система, яка спрощує доступ до інформації про галузі знань, спеціальності та освітні програми. Розроблений веб-сайт та мобільний додаток дозволяють ефективно використовувати дані про освітні програми, забезпечуючи зручний та зрозумілий інтерфейс для користувачів. Ця інформаційна система може бути використана в освітніх установах та інших організаціях, які потребують доступу до інформації про освітні програми.

ВСТУП

У сучасному світі, де технології стають неодмінною частиною нашого повсякденного життя, сфера освіти також швидко розвивається, пристосовуючись до нових вимог і викликів. Розроблення інформаційних систем стає ключовим фактором у поліпшенні та оптимізації освітніх процесів.

Метою цієї кваліфікаційної роботи є розробка інформаційної системи для освітніх програм, яка складатиметься з веб-сайту для адміністрування всіх компонентів освітнього процесу та мобільного додатка для зручного перегляду освітніх програм та детальної інформації про них. Ця система буде сприяти автоматизації та ефективному управлінню освітніми програмами, забезпечуючи зручний доступ до актуальної інформації.

В сучасних умовах, коли кількість освітніх програм постійно зростає, викладачам доводиться мати справу зі значною кількістю адміністративних обов'язків, що створює додаткове навантаження та впливає на якість їх освітньої роботи. У таких умовах виникає потреба в ефективному інструменті, який забезпечуватиме комфорт та пришвидшення роботи з освітніми програмами, знижуючи навантаження на викладачів.

Актуальність даної роботи полягає саме в розробці інформаційної системи, яка спрощує та автоматизує процеси пов'язані з адмініструванням освітніх програм. Вона дозволить викладачам ефективно керувати та актуалізувати освітні програми, вносити зміни та оновлення в зручний спосіб.

Одним із ключових викликів, який потребує вирішення, є нестача централізованої системи для керування освітніми програмами. Звичайні процеси, пов'язані зі змінами в програмах, оновленнями та комунікацією між викладачами та студентами, можуть бути незручними та часовими затратами. Розроблена інформаційна система стане цінним інструментом для забезпечення централізованого контролю, зручного взаємодії та швидкого доступу до інформації про освітні програми.

РОЗДІЛ І ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальні відомості про інформаційні системи

Інформаційні системи є невід'ємною частиною сучасного світу. Вони відіграють ключову роль у зборі, обробці, зберіганні та передачі інформації для підтримки різноманітних ділових процесів та прийняття рішень. Завдяки розвитку технологій, інформаційні системи стали більш потужними, швидкими та ефективними.

Основна мета інформаційних систем полягає в тому, щоб забезпечити потрібну інформацію правильним людям у вірний час. Ці системи складаються з компонентів, таких як апаратне та програмне забезпечення, бази даних, мережі зв'язку, процедури обробки даних та людського фактора. Вони взаємодіють між собою для забезпечення потреб користувачів.

Інформаційні системи можуть бути різних типів, залежно від їхньої функціональності та масштабу застосування. Наприклад, операційні системи керують потоком даних та ресурсами комп'ютерної системи, електронні комерційні системи дозволяють здійснювати електронні торгівлі, управлінські інформаційні системи допомагають в прийнятті стратегічних рішень.

Ключовими характеристиками інформаційних систем є надійність, швидкодія, безпека та зручність використання. Надійність означає, що система повинна працювати стабільно та без збоїв. Швидкодія вимагає оптимізації обробки даних та швидкого доступу до інформації. Безпека забезпечує захист конфіденційності, цілісності та доступності даних. Зручність використання полягає в тому, щоб система була простою у використанні та мала зрозумілий інтерфейс.

Інформаційні системи постійно розвиваються і вдосконалюються, щоб відповідати зростаючим потребам суспільства та бізнесу. Ці системи стали невід'ємною частиною нашого повсякденного життя і грають важливу роль у підтримці ділових процесів та прийнятті рішень. У галузі освіти вони використовуються для автоматизації адміністративних завдань, управління

навчальними програмами та забезпечення комфорту в роботі з освітніми матеріалами.

Розроблена інформаційна система для освітніх програм буде включати вебсайт для адміністрування всіх компонентів освітнього процесу та мобільний додаток для зручного доступу до актуальної інформації про навчальні програми.

Оновлена інформаційна система для освітніх програм спростить процеси реєстрації та комунікації між студентами та викладачами, знизить навантаження на викладачів та забезпечить зручні умови для навчання. Вона стане вагомим кроком у поліпшенні якості освіти та підвищенні ефективності управління освітніми програмами.

Для початку необхідно визначити ключові поняття, які будуть спрямовані на полегшення розуміння процесу розробки інформаційної системи.

1.2 Ключові поняття

Інформаційна система - це комплексний набір взаємопов'язаних компонентів, включаючи апаратне та програмне забезпечення, бази даних, мережі зв'язку, процедури обробки даних та людський фактор, які спільно працюють для збору, зберігання, обробки та передачі інформації з метою підтримки різних бізнес-процесів та прийняття рішень. Основна мета інформаційних систем полягає в тому, щоб забезпечити потрібну інформацію правильним людям у вірний час. Це досягається шляхом збору, обробки та зберігання даних, а також їх подальшої передачі та використання для виконання завдань і досягнення цілей організації або системи. Інформаційні системи можуть бути різних типів, включаючи операційні системи, управлінські інформаційні системи, електронні комерційні системи, системи управління виробництвом та багато інших. Кожен тип інформаційної системи спрямований на конкретну область застосування та має свої особливості та функціональні можливості. Загалом, інформаційна система виступає як

важливий інструмент для забезпечення ефективного збору, обробки та передачі інформації, що сприяє поліпшенню прийняття рішень, збільшенню продуктивності та покращенню управління організаціями та процесами.

Мобільний додаток - це програмне забезпечення, розроблене для встановлення та використання на мобільних пристроях, таких як смартфони та планшети. Він призначений для виконання конкретних функцій або надання певних послуг користувачам безпосередньо на їхніх мобільних пристроях. Мобільні додатки можуть бути доступні через офіційні магазини додатків, такі як App Store для пристроїв Apple або Google Play Store для пристроїв з операційною системою Android. Користувачі можуть шукати та завантажувати додатки на свої пристрої з цих платформ. Мобільні додатки розробляються з урахуванням особливостей мобільних пристроїв, таких як екран на дотик, геолокація, камера та доступ до інших функцій пристрою. Вони можуть мати різні цілі, включаючи соціальні мережі, комунікацію, розваги, фітнес, фінанси, освіту та багато інших сфер застосування. Мобільні додатки надають користувачам зручну та швидку можливість використовувати різноманітні сервіси та функціональні можливості безпосередньо на їхніх мобільних пристроях. Вони стали невід'ємною частиною повсякденного життя і допомагають спростити багато різних задач та покращити продуктивність та зручність використання мобільних пристроїв.

Вебсайт - це колекція веб-сторінок, пов'язаних між собою гіперпосиланнями та розміщених на одному або декількох веб-серверах. Вебсайт доступний через Інтернет і призначений для відображення та передачі інформації користувачам. Кожна веб-сторінка включає текст, графіку, мультимедійний контент або інші елементи, які можуть бути відображені веб-браузером. Вебсайти можуть мати різну структуру та функціональність в залежності від їхньої мети та призначення. Вони можуть бути статичними, коли вміст змінюється рідко, або динамічними, коли вміст генерується на льоту з використанням програмних скриптів та баз даних. Вебсайти

використовуються для різних цілей, таких як представлення компаній, надання інформації, електронної комерції, спілкування, розваг та багато інших. Вони можуть бути особистими, комерційними, урядовими або громадськими. Вебсайти забезпечують користувачам можливість швидко знаходити та отримувати необхідну інформацію, взаємодіяти з різними сервісами, а також здійснювати комунікацію та спілкування з іншими користувачами. Вебсайти можуть мати різний дизайн, структуру та функціональність. Вони можуть бути адаптивними, тобто пристосованими до різних пристроїв, таких як комп'ютери, смартфони та планшети, або мати спеціалізовані версії для мобільних пристроїв. Вебсайти грають важливу роль у представленні інформації та забезпеченні взаємодії в онлайн-середовищі, і вони є невід'ємною частиною сучасного світу Інтернету.

Освітня програма - це структурований набір навчальних матеріалів, ресурсів та активностей, який визначає мету, зміст та послідовність навчання для досягнення певних освітніх цілей. Вона включає у себе навчальні плани, підручники, методичні матеріали, оцінювальні інструменти та інші ресурси, які сприяють засвоєнню знань, навичок та компетентностей студентами або учнями.

Освітні програми можуть бути розроблені на різних рівнях освіти, включаючи початкову школу, середню школу, вищу освіту або професійне навчання. Вони визначають те, що має бути вивчено, які цілі мають бути досягнуті, якими методами та стратегіями має бути організоване навчання та яким чином оцінюється успішність учнів або студентів. Освітні програми можуть бути загальноосвітніми, спрямованими на набуття загальних знань та навичок, або спеціалізованими, що фокусуються на певних предметах, галузях знань або професійних навичках. Вони враховують потреби та вимоги освітньої системи, регуляторних органів, роботодавців та суспільства загалом. Освітня програма визначає не лише академічні знання, але й розвиток загальних компетентностей, критичного мислення, творчого мислення, комунікаційних навичок та інших навичок, які сприяють гармонійному розвитку особистості.

Вона слугує основою для організації навчального процесу та створення умов для досягнення якісної освіти.

1.3 Формулювання задачі

Мета даної кваліфікаційної роботи полягає в розробці інформаційної системи для освітніх програм. Ця система буде складатися з веб-сайту, призначеного для адміністрування всіх компонентів освітнього процесу, а також мобільного додатка, що забезпечуватиме зручний перегляд освітніх програм та детальної інформації про них. Основною метою розробки системи є автоматизація та ефективне управління освітніми програмами, а також забезпечення зручного доступу до актуальної інформації. Система дозволить адміністраторам зручно керувати базою даних, включаючи галузі знань, спеціальності, навчальні програми та їх компоненти. Крім користувачі зможуть легко отримувати доступ до інформації про навчальні програми за допомогою мобільного додатка.

Розроблена програмна система надаватиме функціонал для користувача:

Вебсайту:

- Додавання, редагування та видалення даних про галузі знань.
- Додавання, редагування та видалення даних про спеціальності.
- Додавання, редагування та видалення даних про навчальні програми.
- Додавання, редагування та видалення даних про посилання на ресурси.
- Додавання, редагування та видалення даних про компоненти навчальної програми.
- Додавання, редагування та видалення даних про програмні компетентості.

- Додавання, редагування та видалення даних про зв'язки компетентості.

Мобільного додатку:

- Перегляд галузей знань.
- Перегляд спеціальностей.
- Перегляд навчальних програм.
- Перегляд детальної інформації про навчальну програму.

РОЗДІЛ II ВИКОРИСТАНІ ТЕХНОЛОГІЇ

2.1 Flutter та мова програмування Dart

Стає зрозуміло, якщо переглянути вище сказане, що пропускати одну з двох найпопулярніших операційних систем не є правильно при розробці. Проте витратити багато часу на розробку двох додатків на обох платформах не є найкращий варіант. В результаті цього і було обрано програмний каркас Flutter та мову програмування Dart.

2.1.1 Програмний каркас Flutter

Flutter - це безкоштовний, з відкритим кодом, мобільний фреймворк для створення мобільних додатків від компанії Google. Розроблений у 2017 році, Flutter дозволяє розробникам створювати мобільні додатки з єдиною кодовою базою і мовою програмування. Ця особливість робить додаток простим для мобільних пристроїв з операційними системами iOS та Android досить гнучким та швидким у роботі.

Flutter фреймворк складається з двох наборів для розробки програмного забезпечення (SDK) та бібліотеки інтерфейсу користувача на основі віджетів. Ця бібліотека складається з різних перевикористовуваних візуальних елементів, такі як кнопки та для вводу тексту.

Розробники будують мобільні застосунки з Flutter фреймворком з мовою програмування, яка називається Dart. З синтаксисом, як у JavaScript, Dart є строго типізованою мовою програмування, яка фокусується на інтерфейсній розробці.



Рис. 2.1. Іконка Flutter

Перша версія Flutter з кодовою назвою Sky, працювала тільки на операційній системі Android. Рушій Flutter був написаний на C++.

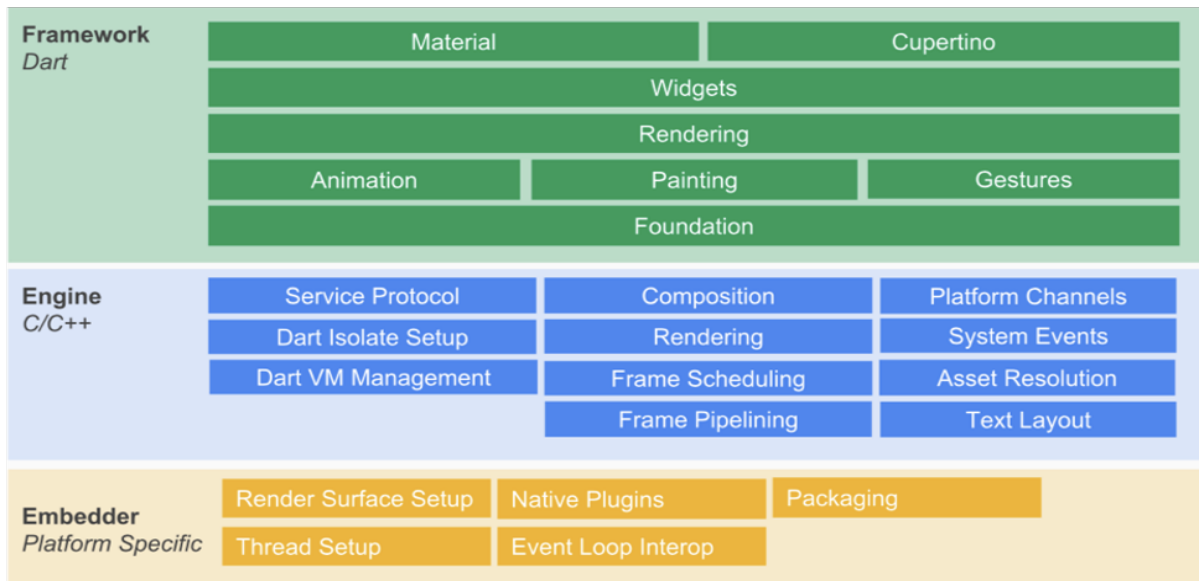


Рис. 2.2. Архітектура Flutter

Хоча Flutter є новим крос-платформним фреймворком, більше і більше компаній почали обрати Flutter замість таких фреймворків, такі як Xamarin, Cordova та React Native.

Деякі з топ причин чому команди розробників обираються Flutter включають:

- збільшена продуктивність
- легко вивчити
- крос-ефективність
- можлива розробка на різних середовищах розробки
- чудова документація та велике ком'юніті

2.1.2 Практичні застосунки мови Dart

Dart - мова програмування розроблення для клієнтської розробки, такої як, веб чи мобільні застосунки. Вона була розроблена компанією Google і може також використовуватися у розробці серверів чи застосунків для персональних комп'ютерів.

Вона є об'єктно-орієнтована, заснована на класах мова зі збиранням сміття з синтаксисом у стилі C. Також, можна скопіювати, як в рідному кодї чи JavaScript і інтерфейси, абстрактні класи та дженеріки.



Рис. 2.3. Іконка Dart

Крім того, Dart має велику підтримку асинхронності, а робота з генераторами та ітерабельними файлами надзвичайно проста.

Dart - це мова загального призначення, і ви можете використовувати її майже для будь-чого:

- У веб-додатках за допомогою бібліотеки: `html` та транслятора для перетворення коду Dart у JavaScript або за допомогою таких фреймворків, як `AngularDart`.
- На серверах, використовуючи бібліотеки: `http` і `io`. Існує також кілька фреймворків, якими можна скористатися, наприклад, `AngularDart`.
- У консольних додатках.
- У мобільних додатках завдяки `Flutter`.

2.2 Фреймворк Bootstrap

Bootstrap є потужним і популярним фреймворком для розробки веб-інтерфейсів. Він надає набір заздалегідь стилізованих компонентів, шаблонів, класів CSS і JavaScript, які дозволяють розробникам швидко та ефективно створювати сучасні, відзивчиві та стильні веб-додатки.

Однією з ключових переваг Bootstrap є його гнучкість та простота використання. Він дозволяє розробникам зосередитися на функціональності додатку, використовуючи готові компоненти та стилізовані елементи. Велика кількість готових класів дозволяє швидко застосовувати стилі до елементів і забезпечує їх консистентність на всій сторінці.

Bootstrap також пропонує адаптивний дизайн, що дозволяє створювати веб-додатки, які коректно відображаються на різних пристроях та екранах. Завдяки цьому фреймворку можна легко створити мобільно-дружні та відзивчиві веб-сторінки без необхідності писати власні медіа-запити або використовувати складні CSS-рішення.

Додатково, Bootstrap має велику спільноту розробників, що підтримує його активним розвитком та надає багато готових рішень, документацію, приклади коду та шаблони, які допомагають прискорити розробку веб-інтерфейсу.



Рис. 2.4. Іконка Bootstrap

2.3 Середовище розробки Visual Studio Code

Середовищем розробки для проєкту було обрано Visual Studio Code. Це є проєкт з відкритим вихідним кодом від компанії Microsoft. VS Code дозволяє розробку практично на всіх існуючих мовах програмування за допомогою додаткових компонентів для програми. Ці компоненти, також, можуть бути розробленими звичайними користувачами.

В опитуванні розробників Stack Overflow, Visual Studio Code визнано найпопулярнішим інструментом середовища розробника серед 82 000 респондентів, причому 70% повідомили, що вони ним користуються.

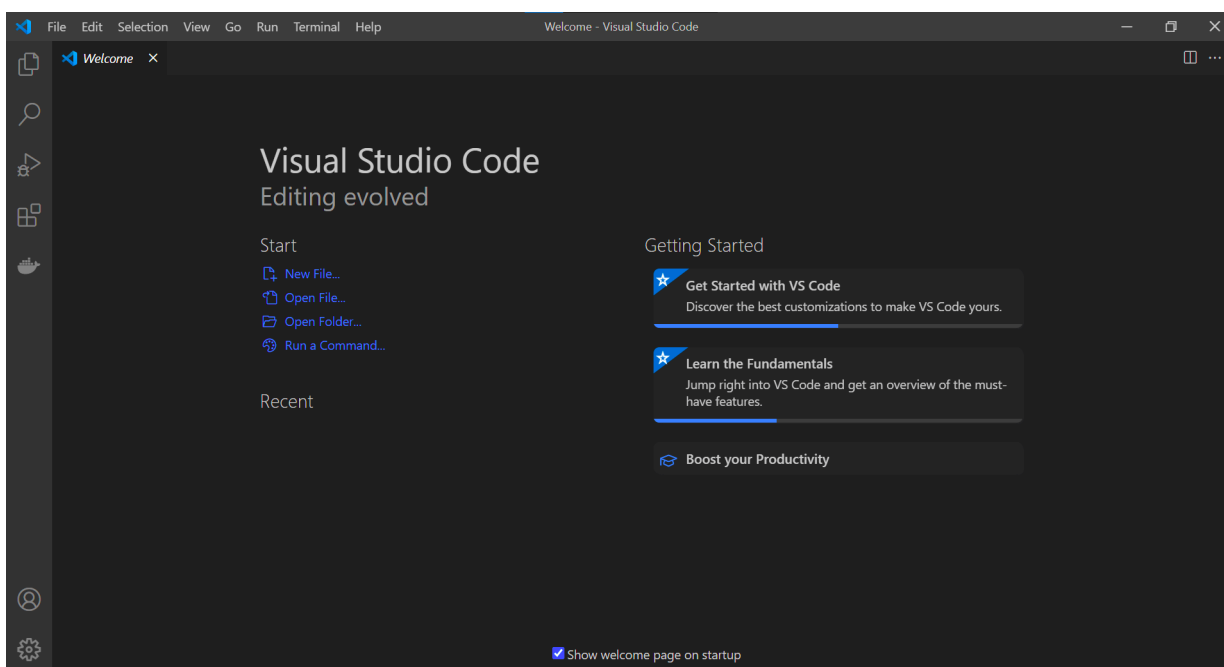


Рис. 2.5. Головний екран Visual Studio Code

РОЗДІЛ III ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Роль серверної і клієнтської частин у проекті

Для ефективної реалізації вебзастосунку необхідно чітко розуміти роль серверної та клієнтської частини. Серверна частина вебзастосунку виконується на сервері і відповідає за обробку запитів вебклієнтів, збереження та керування даними, реалізацію бізнес-логіки та надання відповідей на запити. Клієнтська частина вебзастосунку складається з програмного коду та графічного інтерфейсу, що працюють на боці користувача і відповідають за взаємодію з користувачем та відображення вебзастосунку на його пристрої.

Розробка клієнтської частини вебзастосунку може включати використання різних технологій, таких як HTML, CSS і JavaScript, а також фреймворків і бібліотек для розширення функціоналу та поліпшення користувацького досвіду. Клієнтська частина може включати такі елементи, як графічний інтерфейс, логіку взаємодії та взаємодію з базою даних.

У свою чергу, серверна частина вебзастосунку може включати різні компоненти, такі як вебсервер, база даних, мікросервіси, додаткові бібліотеки та фреймворки, а також спеціалізовані модулі для виконання конкретних завдань. Вона відповідає за обробку запитів, перевірку даних, доступ до бази даних, взаємодію з іншими системами та інші функції, необхідні для належної роботи вебзастосунку.

Основні завдання серверної частини вебзастосунку включають:

- Прийом та обробку запитів від вебклієнтів: серверна частина отримує HTTP-запити від клієнтів, аналізує їх та виконує необхідні дії відповідно до логіки застосунку.
- Роботу з базою даних: вебзастосунки часто потребують зберігання та керування даними. Серверна частина взаємодіє з базою даних, виконує запити до неї, отримує та зберігає дані.

Загалом, серверна частина вебзастосунку є ключовим елементом, який забезпечує обробку запитів користувачів, збереження даних та виконання бізнес-логіки. Розробка серверної частини має враховувати унікальні потреби та вимоги конкретного вебзастосунку.

3.2 Серверна частина проекту

3.2.1 Фреймворк Flask та його особливості

Flask - це легкий та гнучкий фреймворк для розробки веб-додатків на мові програмування Python. Він надає необхідні інструменти та бібліотеки для швидкої і простої розробки серверної частини вебзастосунків.

Основні особливості Flask:

- Мікрофреймворк: Flask є мікрофреймворком, що означає, що він має мінімальний набір функцій та залежностей. Це дозволяє розробникам бути більш гнучкими та вибирати самостійно бібліотеки та інструменти, які найкраще підходять для їх потреб.
- Простота використання: Flask пропонує простий та інтуїтивно зрозумілий інтерфейс для розробки веб-додатків. Він має зрозумілу структуру та легко вивчається. Flask дозволяє швидко встановити основний функціонал та розпочати розробку без зайвих зусиль.
- URL-маршрутизація: Flask дозволяє визначати URL-шляхи або маршрути та пов'язувати їх з функціями-обробниками. Це дозволяє серверу реагувати на конкретні URL-адреси та виконувати необхідну логіку.
- Вбудована робота з шаблонами: Flask має вбудовану підтримку для шаблонів, що дозволяє відокремити логіку від представлення. Розробники можуть використовувати шаблонізатор Jinja2 для створення динамічних сторінок змішуванням HTML та Python-коду.
- Розширюваність: Flask дозволяє легко розширювати функціональність за допомогою розширень (extensions). Розширення - це сторонні

бібліотеки, які надають додаткові функції та інтегруються з Flask. Наприклад, є розширення для роботи з базами даних, формами, аутентифікації та багато інших.

- Підтримка HTTP-протоколу: Flask добре інтегрується з протоколом HTTP. Він дозволяє обробляти різні типи запитів, встановлювати заголовки, отримувати доступ до параметрів запиту та відправляти відповіді клієнту.
- Розробка тестів: Flask має вбудовану підтримку для розробки тестів, що дозволяє автоматизувати перевірку функціональності вашого додатку. Ви можете тестувати маршрути, форми, базу даних та інші компоненти вашого додатку.

Флask є одним з найпопулярніших фреймворків для розробки веб-додатків на мові програмування Python. Він позначається своєю простотою, гнучкістю та швидкістю розробки. З його допомогою розробники можуть створювати потужні та масштабовані серверні частини вебзастосунків з невеликими зусиллями.

3.2.2 Реалізація серверної частини на прикладі таблиці “Галузі знань”

```
@app.route('/branch_table')
def branchs_table():
    connection = mysql.connector.connect(
        user='admin',
        host='127.0.0.1',
        database='edu_program_db')
    cursor = connection.cursor()
    cursor.execute('SELECT * FROM edu_program_db.branchs')
    data = cursor.fetchall()
    connection.close()
    data = list(data)
    data.sort()
    return render_template('branchs_table.html', list_data=data)
```

Функція `branch_table()` відповідає за відображення таблиці галузі знань.

```
@app.route('/add_branch', methods=['POST', 'GET'])
def add_branch():
    if request.method == 'POST':
        name = request.form['name']
```

```

number = request.form['number']
connection = mysql.connector.connect(
    user='admin',
    host='127.0.0.1',
    database='edu_program_db')
cursor = connection.cursor()
cursor.execute(
    f"INSERT INTO edu_program_db.branchs(name, number) VALUES
({name},{number})")
connection.commit()
cursor.close()
connection.close()
flash('Галузь знань додано', 'success')
return redirect(url_for('branchs_table'))
return render_template('add_branch.html')

```

Функція `add_branch()` відповідає за додавання нової галузі знань до бази даних.

```

@app.route('/edit_branch/<string:id>', methods=['POST', 'GET'])
def edit_branch(id):
    if request.method == 'POST':
        name = request.form['name']
        number = request.form['number']
        connection = mysql.connector.connect(
            user='admin',
            host='127.0.0.1',
            database='edu_program_db')
        cursor = connection.cursor()
        cursor.execute(
            f'UPDATE edu_program_db.branchs SET name="{name}", number="{number}" WHERE
id={id}')
        connection.commit()
        cursor.close()
        connection.close()
        flash('Дані про галузь знань оновлено', 'success')
        return redirect(url_for('branchs_table'))
    connection = mysql.connector.connect(
        user='admin',
        host='127.0.0.1',
        database='edu_program_db')
    cursor = connection.cursor()
    cursor.execute(f'SELECT * FROM edu_program_db.branchs WHERE id={id}')
    data = cursor.fetchall()
    cursor.close()
    connection.close()
    return render_template('edit_branch.html', data=data)

```

Функція `edit_branch()` відповідає за редагування даних про галузь знань.

```

@app.route('/delete_branch/<string:id>', methods=['GET'])
def delete_branch(id):
    connection = mysql.connector.connect(
        user='admin',

```

```
host='127.0.0.1',
database='edu_program_db')
cursor = connection.cursor()
cursor.execute(f'DELETE FROM edu_program_db.branchs WHERE id={id}')
connection.commit()
cursor.close()
connection.close()
flash('Галузь знань видалено', 'warning')
return redirect(url_for('branchs_table'))
```

Функція `delete_branch()` відповідає за видалення галузі знань з бази даних.

3.3 Застосована база даних

MySQL - це відкрите програмне забезпечення для управління базами даних, що використовує мову запитів SQL (Structured Query Language). Він є одним з найпопулярніших реляційних систем управління базами даних (СУБД) і широко використовується для зберігання, управління та отримання даних у веб-додатках.

Основні характеристики MySQL:

- Реляційна база даних: MySQL заснований на моделі реляційної бази даних, де дані організовані у відношеннях (таблицях) зі структурованими зв'язками між ними.
- Мова запитів SQL: MySQL використовує мову запитів SQL для взаємодії з базою даних. SQL дозволяє створювати, змінювати та видаляти дані в базі, а також виконувати складні операції фільтрації, сортування та об'єднання даних.
- Підтримка транзакцій: MySQL підтримує транзакції, які дозволяють виконувати групу операцій як єдину атомарну одиницю. Це забезпечує цілісність та надійність даних у випадку відмови системи або помилок.
- Масштабованість: MySQL має гнучку архітектуру, яка дозволяє масштабувати базу даних для великих обсягів даних і високого навантаження. Він може бути використаний як для невеликих проєктів, так і для великих підприємств систем.

- Підтримка багатокористувацького доступу: MySQL дозволяє керувати доступом до бази даних і таблиць шляхом надання різним користувачам різних рівнів дозволів. Це забезпечує безпеку та контроль над даними.
- Підтримка реплікації: MySQL надає можливість створювати резервні копії та реплікувати дані на кількох серверах. Це забезпечує високу доступність та швидкодію системи.

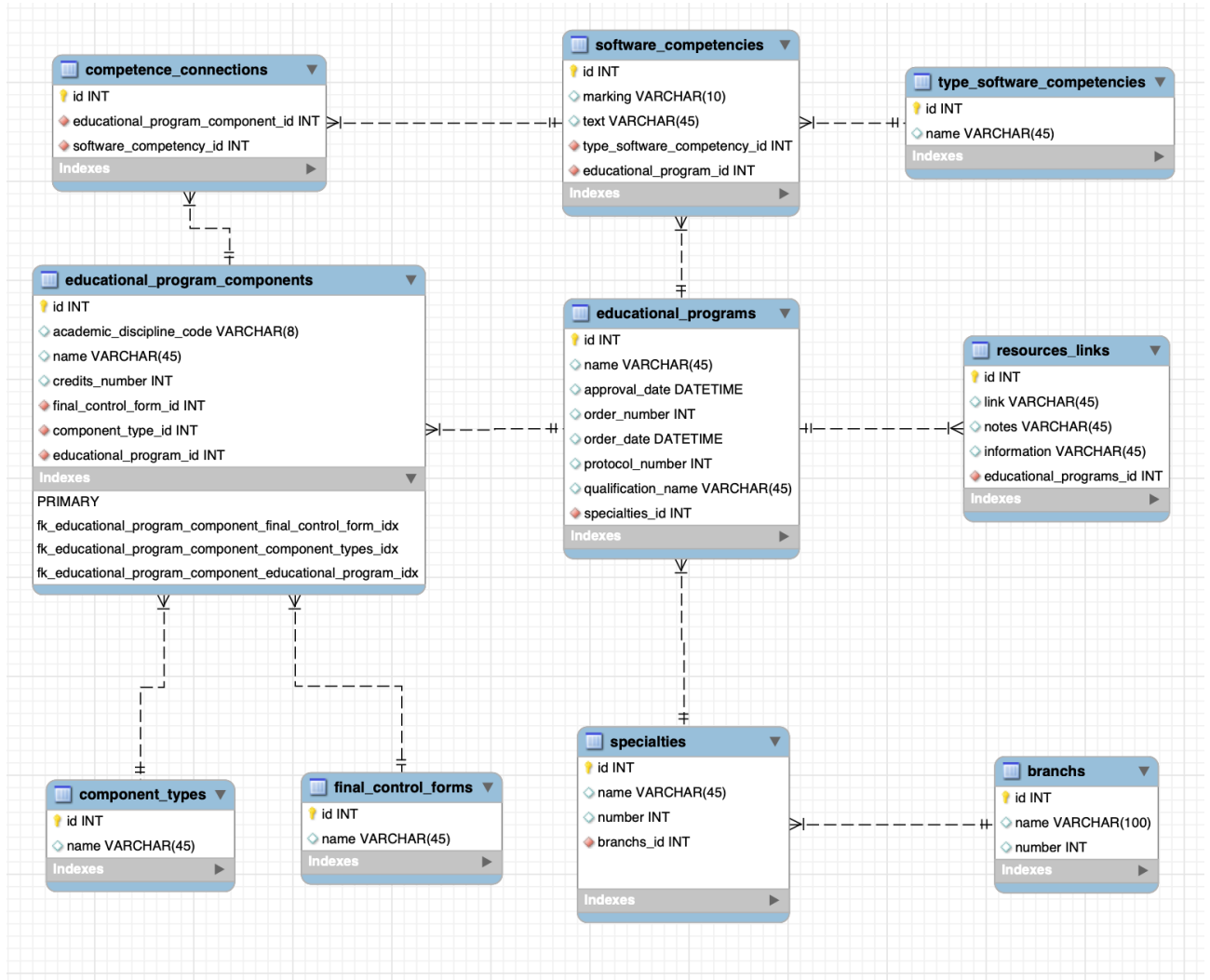


Рис. 3.1. Діаграми бази даних інформаційної системи

Наведена вище діаграма демонструє структуру даних для користувача у базі даних. Таблиця “branches” відповідає за дані про галузі знань, “specialties” – спеціальності, “educational_programs” – освітні програми, “resources_links” – посилання на ресурси, “educational_program_components” – компоненти освітньої програми, “component_types” – вид

компоненти((обовязкова, вільний вибір, вибірковий блок 1, вибірковий блок 2), “final_control_forms” – форма підсумкового контролю(іспит чи екзамен), “software_competencies” – програмні компетентності, “type_software_competencies” – типи програмних компетентостей(ЗК, ФК), “competence_connections” – зв’язки компетентості.

3.4 Використана архітектура мобільного додатку

3.4.1 Bloc шаблон

Для фреймворка Flutter існує багато шаблонів для розділення бізнес логіки та презентаційного шару. В даній частині проєкті було використано BLoC шаблон для розробки модуля логіки. В цьому модулі бізнес логіки існує декілька зв’язків, які є певними вхідними потоками для агрегування подій ззовні. В якості вихідних даних виступає Stream (потік), який визначає асинхронний формат даних для нашого користувацького інтерфейсу, тобто віджетів. Для використання модулів на рівні віджету використовується StreamBuilder який сам керує потоком даних і автоматично вирішує проблеми перемальовування дочірнього дерева віджетів.

BLoC (Business Logic Component) - це шаблон, який був розроблений компанією Google для оперування складними станами додатку, що ґрунтується на реактивній парадигмі.

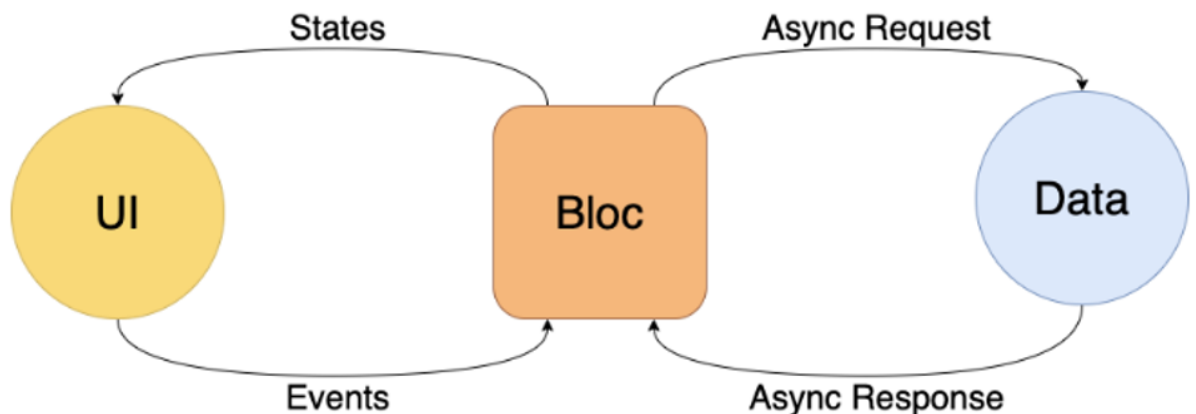


Рис. 3.2. Патерн BLoC

Переваги VLoC:

- великий програмний інтерфейс для роботи з потоками, що дозволяє легко їх групувати і трансформувати.
- групування логіки в одному місці.
- простота в тестуванні та додаванні нового функціоналу.
- мінімальна кількість перезбирання користувацького інтерфейсу за допомогою використання `StreamBuilder`.

3.4.2 Stateless та Stateful віджети

Стан програми можна дуже просто визначити як будь-що, що існує в пам'яті програми під час її роботи. Сюди входять усі віджети, які підтримують інтерфейс програми, включаючи кнопки, текстові шрифти, піктограми, анімацію тощо.

`StatelessWidget`: Віджети, стан яких не можна змінити після створення, називаються віджетами без стану. Ці віджети залишаються незмінними після створення, тобто будь-яка кількість змін у змінних, піктограмах, кнопках або отриманих даних не може змінити стан програми.

Приклад оголошення `StatelessWidget`:

```
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  @override  
  Widget build(BuildContext context) {  
    return Container();  
  }  
}
```

`StatefulWidget`: Віджети, стан яких можна змінювати після створення, називаються віджетами зі збереженням стану. Ці стани є змінними і можуть бути змінені кілька разів протягом свого життя. Це означає, що стан програми може змінюватися кілька разів з різними наборами змінних, вхідних даних і даних. Прикладами можуть бути поле для галочки, опціональний вибір, форма та поле для вводу тексту.

Приклад оголошення StatefulWidget:

```
class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

3.4.3 Використані сторонні пакети

В розробленій програмі використано пакети, за допомогою яких реалізують ряд функцій. Нижче представлені та описані декілька з них:

- auto_route – навігація по сторінках
- intl – локалізація застосунку
- dio – робота з REST API
- flutter_bloc - BloC архітектура
- retrofit - клієнт Dio, який полегшує використання Rest API
- freezed - дозволяє створювати незмінні моделі даних

Пакети є основною функцією менеджера пакетів pub у мові програмуванні Dart. Для роботи з ними, пакети потрібно вказувати у файлі pubspec.yaml.

3.5 Реалізація вебсайту з використанням Bootstrap table

Bootstrap Table - це потужне розширення для фреймворку Bootstrap, яке надає можливості створення стильних та функціональних таблиць на веб-сторінках. Це дозволяє розробникам легко та ефективно відображати дані у вигляді табличної структури і забезпечувати користувачам зручний спосіб навігації та взаємодії з цими даними.

Однією з головних переваг Bootstrap Table є його простота використання. Завдяки інтеграції з Bootstrap, створення таблиць з вбудованими стилями та класами є легким завданням. Ви можете легко використовувати попередньо визначені класи Bootstrap для стилізації таблиць та їх елементів, таких як заголовки, рядки, стовпці та кнопки.

Bootstrap Table надає широкий набір функціональних можливостей для зручної навігації та взаємодії з даними. Ви можете використовувати вбудовані опції для сортування, фільтрації, пагінації та пошуку в таблицях. Крім того, ви можете налаштовувати вигляд таблиці, додавати колонки з кнопками або іншими елементами керування, розміщувати додаткову інформацію в кожній комірці та використовувати розширені функціональність, таку як перетягування та зміна розмірів стовпців.

Bootstrap Table також підтримує адаптивний дизайн, що дозволяє таблицям належним чином відображатися на різних пристроях та розмірах екрану. Ви можете налаштовувати поведінку таблиць для мобільних пристроїв, таку як згортання рядків, приховування стовпців або відображення спеціальних кнопок для навігації.

Завдяки розширенню Bootstrap Table розробники можуть швидко і зручно створювати стильні та інтерактивні таблиці для відображення та взаємодії з даними на веб-сторінках. Він є потужним інструментом для роботи з даними та надає розробникам багато можливостей для налаштування та розширення функціональності таблиць у своїх проектах.

Приклад використання розширення Bootstrap Table на сторінці “Галузі знань”:

```
<table class='table table-bordered' data-toggle="table" data-pagination="true" data-search="true">
  <thead>
    <th data-sortable="true">№</th>
    <th data-sortable="true">Назва</th>
    <th data-sortable="true">Номер</th>
    <th data-searchable="false"></th>
```

```

        <th data-searchable="false"></th>
    </thead>
    <tbody>
    {% for data in list_data %}
        <tr>
            <td>{{data[0]}}</td>
            <td>{{data[1]}}</td>
            <td>{{data[2]}}</td>
            <td><a href='{{url_for("edit_branch",id=data[0])}}' class='btn btn-
primary'>Редагувати</a></td>
            <td><a href='{{url_for("delete_branch",id=data[0])}}' class='btn btn-
danger' onclick='return confirm("Ви впевнені?")'>Видалити</a></td>
        </tr>
    {%endfor%}
    </tbody>
</table>

```

3.6 Огляд інтерфейсу користувача вебсайту

3.6.1 Головна сторінка

На початку головної сторінки користувачу доступна панель навігації за допомогою якої він може перемикатися між таблицями бази даних. Усього доступно сім таблиць: Галузі знань, Спеціальності, Навчальні програми, Посилання на ресурси, Програмні компетентності, Компоненти освітньої програми, Зв'язки компетентності.

Нижче можна побачити кнопку “Додати спеціальність”(так як зараз активна вкладка Спеціальності). При натисненні на кнопку відкриється сторінка з можливістю додавання спеціальності, так само і для інших вкладок будуть відображатися свої кнопки з аналогічним функціоналом.

Далі розміщується власне сама таблиця з даними, а в кожному з рядків доступні дві кнопки – “Редагувати” та “Видалити”. Для швидкого та зручного доступу до потрібної інформації реалізовано пошук.

Інформаційна система для освітніх програм

Галузі знань Спеціальності Навчальні програми Посилання на ресурси Програмні компетентності Компоненти освітньої програми Зв'язки компетентності

Додати спеціальність

№	Назва	Номер	№ галузі знань		
1	Ком'ютерні науки	122	1	Редагувати	Видалити
2	Комп'ютерна інженерія	123	1	Редагувати	Видалити
3	Прикладна математика	113	1	Редагувати	Видалити
4	Інженерія програмного забезпечення	121	1	Редагувати	Видалити
5	Інформаційні системи та технології	126	1	Редагувати	Видалити
6	Автоматизація та комп'ютерно-інтегровані технології	151	1	Редагувати	Видалити
7	Будівництво та цивільна інженерія	192	1	Редагувати	Видалити

Рис. 3.3. Головна сторінка

Також в кінці сторінки реалізована пагінація, якщо кількість елементів в таблиці є зовеликою, для того, щоб поміститися на одній сторінці.

8	Екологія	101	1	Редагувати	Видалити
9	Екноміка	51	1	Редагувати	Видалити
10	Маркетинг	75	1	Редагувати	Видалити

Showing 1 to 10 of 12 rows 10 rows per page

< 1 2 >

Рис. 3.4. Продовження головної сторінки

Після успішного видалення рядка з таблиці вверху сторінки буде показане відповідне сповіщення.

3.6.2 Сторінка додавання елементів в таблицю

Для того щоб додати елемент в таблицю потрібно заповнити всі поля та натиснути кнопку підтвердити, якщо всі поля були правильно заповнені, після підтвердження користувача перекине на головну сторінку та вверху сторінки буде показане сповіщення про успішне додавання нового елементу в таблицю.

Галузі знань Спеціальності Навчальні програми Посилання на ресурси Програмні компетентності Компоненти освітньої програми Зв'язки компетенції

Додати навчальну програму

Назва

Дата затвердження

« Червень 2023 »						
Нд	Пн	Вт	Ср	Чт	Пт	Сб
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Назва кваліфікації

№ спеціальності

Рис. 3.5. Додавання елементів в таблицю

3.6.3 Сторінка редагування елементів в таблиці

Для того щоб редагувати дані елемента потрібно як і у випадку додавання правильно заповнити усі поля та натиснути кнопку підтвердити, після підтвердження користувача перекине на головну сторінку та вверху сторінки буде показане сповіщення про успішне редагування даних вибраного елемента таблиці.

Інформаційна система для освітніх програм

Галузі знань Спеціальності Навчальні програми Посилання на ресурси Програмні компетентності Компоненти освітньої програми Зв'язки компетенції

Редагувати дані про навчальну систему

Назва

Дата затвердження

Номер наказу

Дата наказу

Номер протоколу

Назва кваліфікації

№ спеціальності

Рис. 3.6. Додавання елементів в таблицю

3.7 Огляд інтерфейсу користувача мобільного додатку

3.7.1 Головна сторінка

На головній сторінці користувачу доступна панель навігації за допомогою якої він може перемикатися між вкладками. Усього доступно три вкладки: Галузі знань, Спеціальності, Навчальні програми.

Також користувачу може скористатися пошуком, якщо активна вкладка “Галузі знань”, то пошук буде відбуватися по галузям знань, аналогічна робота пошуку і на вкладках “Спеціальності” та “Навчальні програми”.

На вкладці “Галузі знань” відображений список усіх галузей знань, при натисненні на конкретну галузь знань, користувача перекине на вкладку спеціальності, на якій буде показано всі спеціальності даної галузі знань.

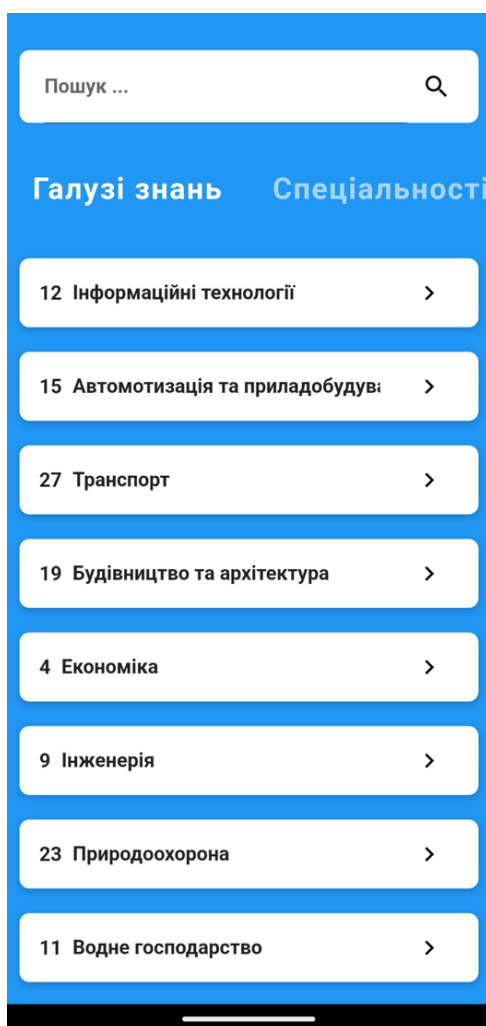


Рис. 3.7. Головна сторінка, список галузей знань

На вкладці “Спеціальності” відображений список усіх спеціальностей, при натисненні на конкретну спеціальність, користувача перекине на вкладку навчальні програми, на якій буде показано всі навчальні програми даної спеціальності.

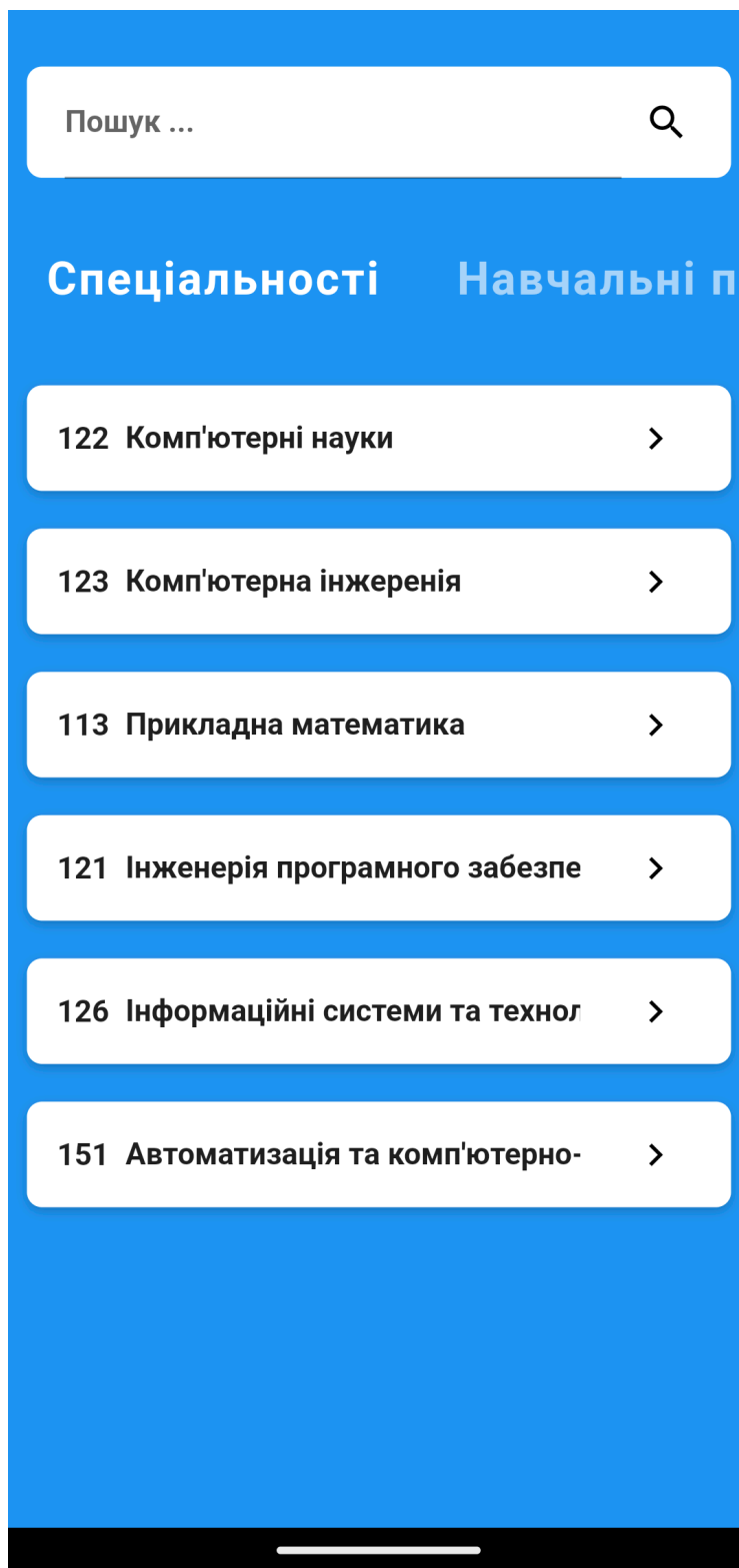


Рис. 3.8. Головна сторінка, список спеціальностей

На вкладці “Навчальні програми” відображений список усіх навчальних програм, при натисненні на начальну програму, користувача перекине на нову сторінку з детальною інформацією про дану начальну програму.

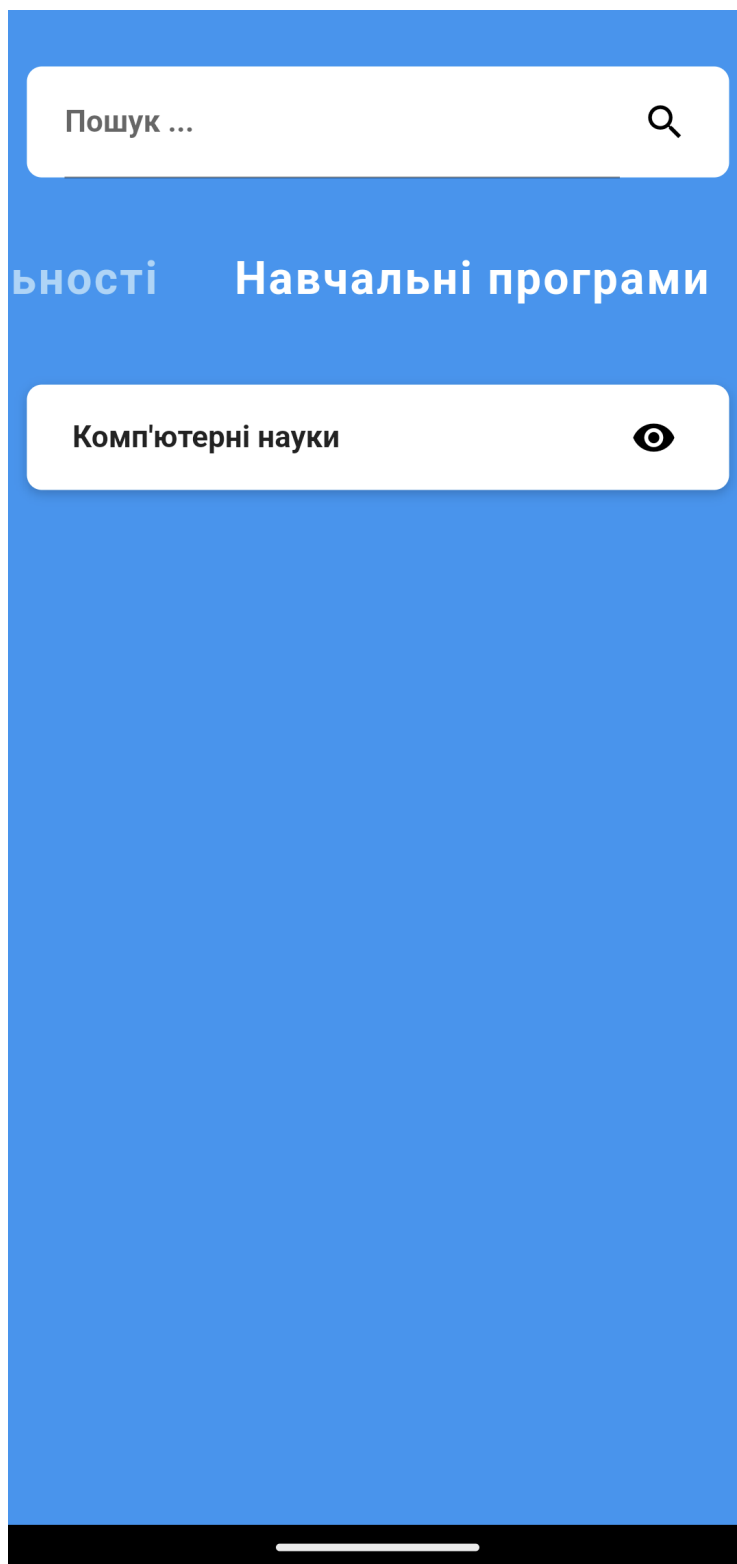


Рис. 3.9. Головна сторінка, список навчальних програм

3.7.2 Сторінка детальної інформації про навчальну програму

На початку сторінки сторінки з детальною інформацією про навчальну програму можна побачити таку інформацію про навчальну програму: назва, назва галузі знань, назва спеціальності, назва кваліфікації, номер наказу, дату наказу, номер протоколу і дату затвердження.



Рис. 3.10. Сторінка детальної інформації про навчальну програму, початок

Далі можна переглянути список загальних компетентостей навчальної програми.

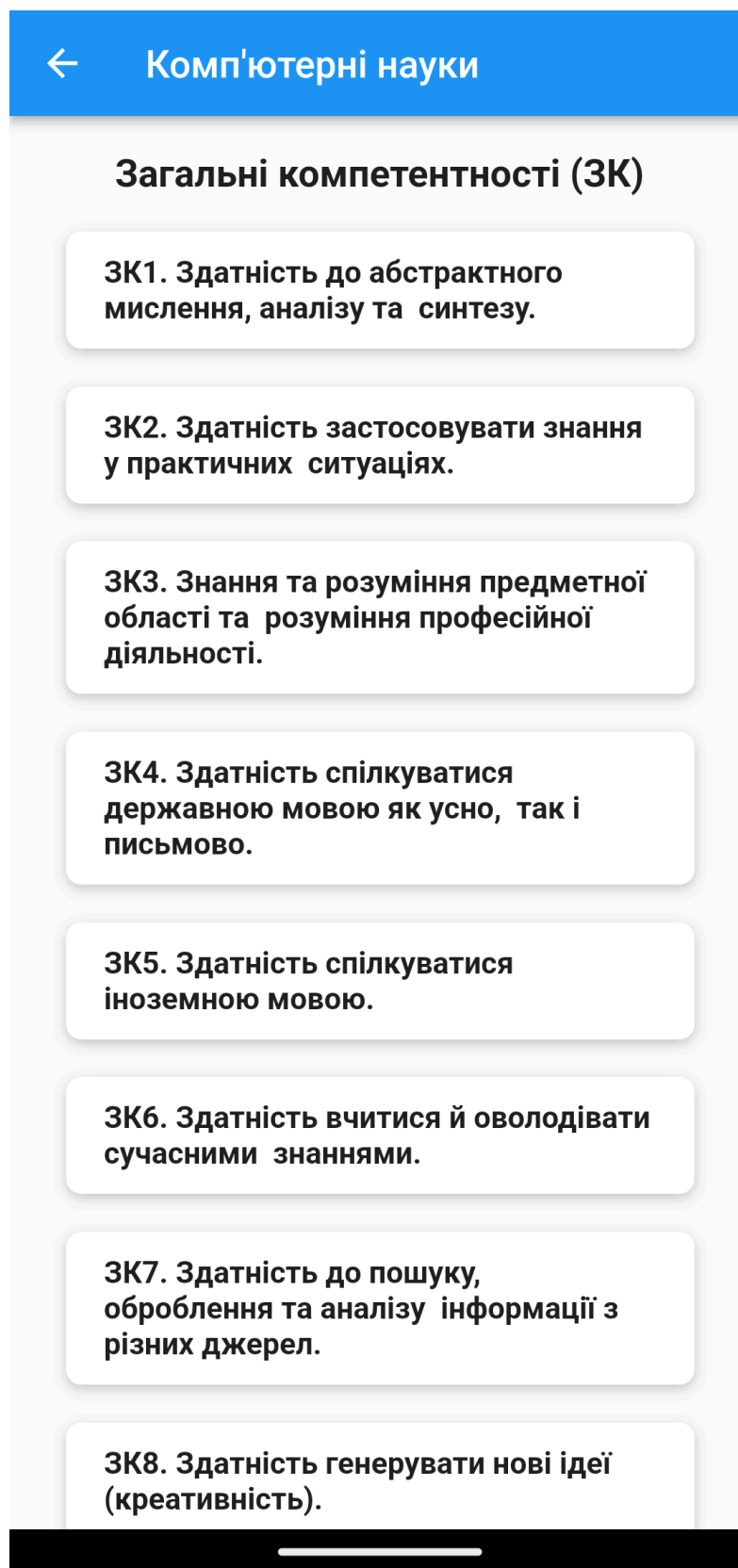


Рис. 3.11. Сторінка детальної інформації про навчальну програму, список загальних компетентостей

Далі можна переглянути список фахових компетентостей навчальної програми.

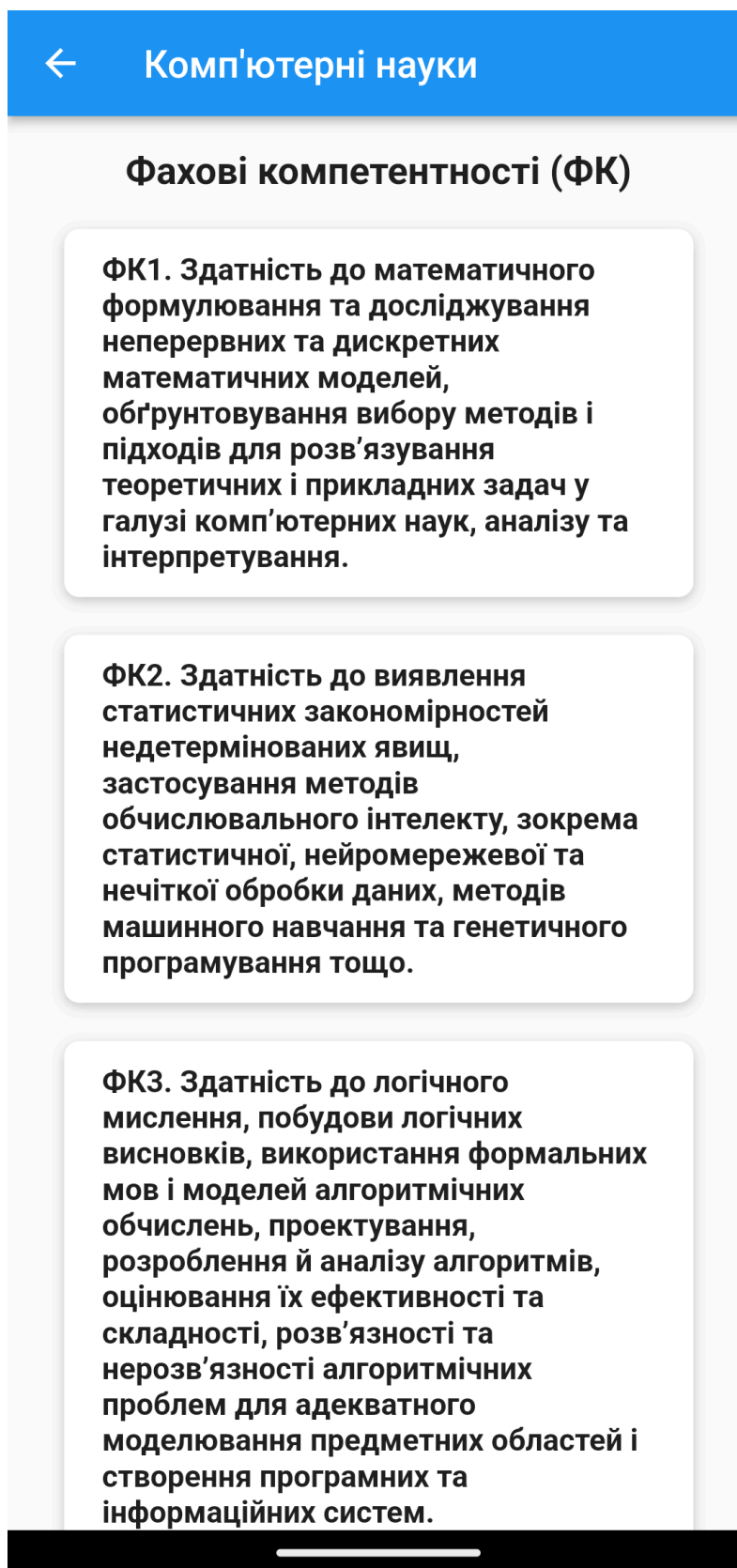
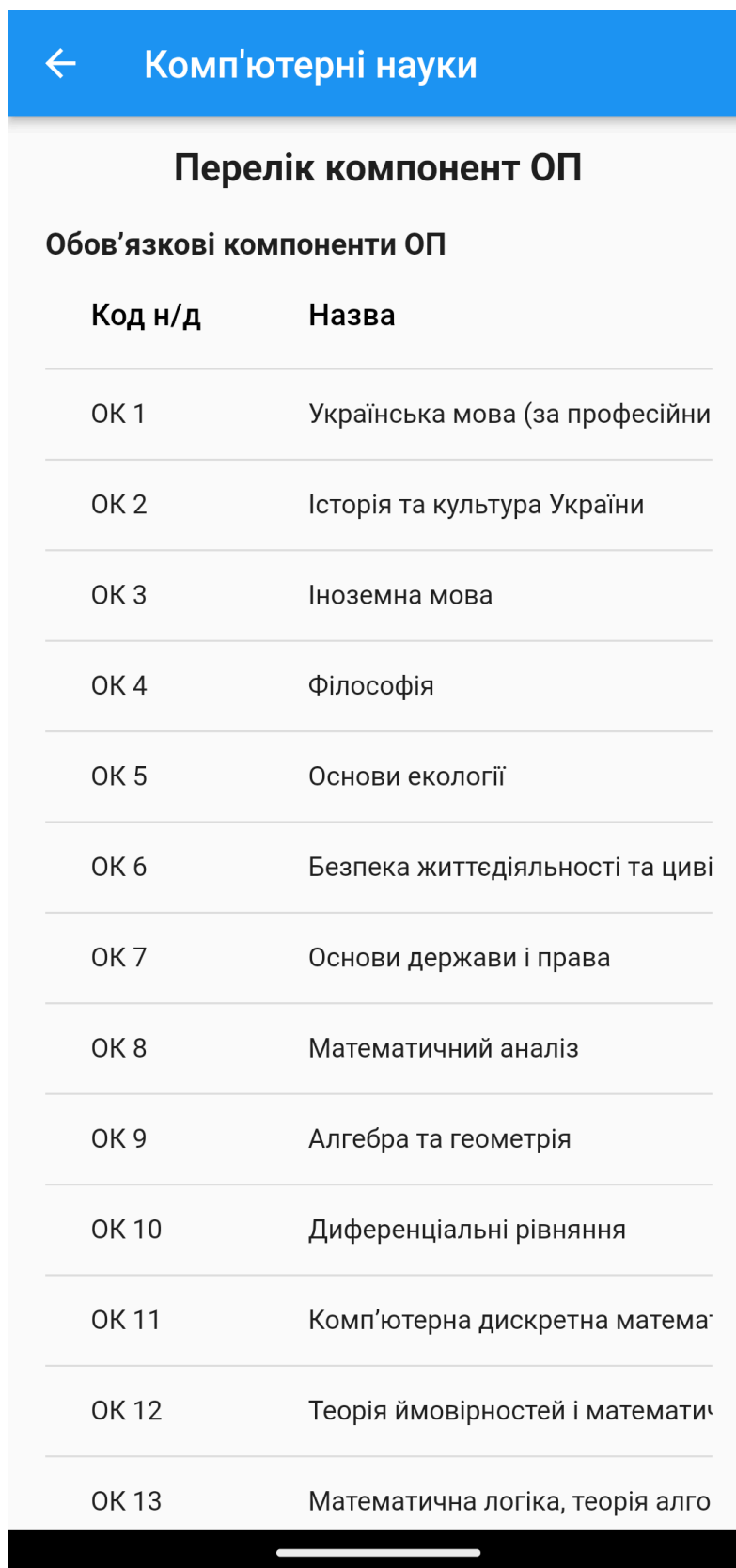


Рис. 3.12. Сторінка детальної інформації про навчальну програму, список фахових компетентностей

Далі можна переглянути перелік компонент, а саме обов'язкових компонент навчальної програми.



Код н/д	Назва
ОК 1	Українська мова (за професійни
ОК 2	Історія та культура України
ОК 3	Іноземна мова
ОК 4	Філософія
ОК 5	Основи екології
ОК 6	Безпека життєдіяльності та циві
ОК 7	Основи держави і права
ОК 8	Математичний аналіз
ОК 9	Алгебра та геометрія
ОК 10	Диференціальні рівняння
ОК 11	Комп'ютерна дискретна матема
ОК 12	Теорія ймовірностей і математи
ОК 13	Математична логіка, теорія алго

Рис. 3.13. Сторінка детальної інформації про навчальну програму, перелік обов'язкових компонент

Користувач може переглянути таку інформацію про компоненти навчальної програми – код і назву, а прокрутивши вправо інформацію, що не помістилася на екрані відразу – кількість кредитів та форму підсумкового контролю.



← Комп'ютерні науки	
Обов'язкові компоненти ОП	
Кількість кредитів	Форма підсумк. кс
3,0	екзамен
3,0	екзамен
5,5	екзамен
3,0	екзамен
3,0	залік
3,0	залік
3,0	залік
10,0	екзамен
8,0	екзамен
4,0	залік
5,0	екзамен
4,5	екзамен
5,0	залік
6,0	екзамен

Рис. 3.14. Сторінка детальної інформації про навчальну програму, перелік обов'язкових компонент, продовження

Далі можна переглянути перелік вибіркового компоненту навчальної програми, як і у випадку з обов'язковими компонентами доступна така інформація: кож, назва, кількість кредитів і форма підсумкового контролю.

← Комп'ютерні науки	
6,0	екзамен
3,0	залік
Загальний обсяг обов'язкових компонент: 180	
Вибіркові компоненти ОП	
Код н/д	Назва
ВВ 36	Методи глибинного аналізу дани Розробка спеціалізованих додат
ВВ 37	Випадкові процеси та їх моделю Моделювання важкоформалізо
ВВ 38	Програмування для мобільних д Методи автоматизованого розгс
ВВ 39	Математичне та комп'ютерне мс Математичне та комп'ютерне мс
ВВ 40	Інформаційні та комп'ютерні ме Комп'ютерні мережі, інтерфейси
ВВ 41	Командна розробка програмних Розробка ігор
Вибірковий блок 1	
Код н/д	Назва
ВВ 42	Управління, адміністрування та Інформаційні системи підприємс
ВВ 43	Рівняння математичної фізики Комп'ютерна математика
ВВ 44	Чисельні методи математичної с Оптимізація обчислень в методе

Рис. 3.15. Сторінка детальної інформації про навчальну програму, перелік вибіркового компоненту

Перелік вибірових компонент навчальної програми ділиться на частини: вибіровий блок 1 і вибіровий блок 2.

←
Комп'ютерні науки

ВВ 41	Командна розробка програмних Розробка ігор	
Вибірковий блок 1		
Код н/д	Назва	
ВВ 42	Управління, адміністрування та с Інформаційні системи підприємств	
ВВ 43	Рівняння математичної фізики Комп'ютерна математика	
ВВ 44	Чисельні методи математичної с Оптимізація обчислень в методах	
ВВ 45	Теорія систем, системний аналіз Прийняття рішень в умовах невизначеності	
ВВ 46	Спецкурс (за вибором)	
Вибірковий блок 2		
Код н/д	Назва	Кі
ВВ 47	Військова підготовка	29
ВВ 48	Військова підготовка	6,1

Загальний обсяг вибірових компонент: 60

Загальний обсяг освітньої програми: 240

Відкрити матрицю відповідності програмних компетентностей компонентам освітньої програми

Рис. 3.16. Сторінка детальної інформації про навчальну програму, перелік вибірових компонент, продовження

Внизу сторінки для перегляду доступна загальна інформація про компоненти навчальної програми, а також є можливість переглянути матрицю відповідності програмних компетентностей компонентам навчальної програми.

	ОК 1	ОК 2	ОК 3	ОК 4	ОК 5	ОК 6	ОК 7	ОК 8
ЭК 1				•				•
ЭК 2						•		
ЭК 3	•					•		
ЭК 4	•	•		•				
ЭК 5			•					
ЭК 6			•					
ЭК 7	•	•	•	•	•		•	•

Рис. 3.17. Матриця відповідності програмних компетентностей компонентам навчальної програми

ВИСНОВОК

У ході даної кваліфікаційної роботи була успішно розроблена інформаційна система для освітніх програм, яка має веб-сайт для адміністрування бази даних та мобільний додаток для зручного перегляду освітніх програм.

Розроблений веб-сайт дозволяє адміністраторам зручно та ефективно керувати даними про освітні програми, включаючи додавання, редагування та оновлення інформації про галузі знань, спеціальності та складові освітніх програм. Це дозволяє забезпечити актуальність та точність даних, що використовуються в системі.

Мобільний додаток, розроблений у рамках цієї роботи, надає користувачам зручний та доступний інтерфейс для перегляду освітніх програм та отримання детальної інформації про них. Користувачі можуть швидко знайти необхідну інформацію про різні освітні програми, що сприяє зручності та ефективності їх пошуку.

Завдяки розробленій інформаційній системі, викладачі та інші зацікавлені особи зможуть ефективно використовувати дані про освітні програми. Система забезпечує зручний доступ до актуальної інформації, що дозволяє швидко оновлювати та адаптувати освітні програми до змінних вимог та потреб.

Ця інформаційна система може бути використана в освітніх установах та інших організаціях, які мають потребу в доступі до інформації про освітні програми. Вона сприятиме полегшенню адміністративних процесів, підвищенню ефективності роботи викладачів та поліпшенню доступу до актуальної інформації про освітні програми для студентів та інших зацікавлених осіб.

Отже, розроблена інформаційна система має значний потенціал у сфері освіти, сприяючи автоматизації, ефективному управлінню та поліпшенню

доступу до інформації про освітні програми. Результати цієї роботи можуть бути використані для подальшого розвитку і вдосконалення системи, а також для впровадження її в реальні умови освітніх установ та організацій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. What is Flutter and why you should learn [електронний ресурс] –
<https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>
2. Flutter docs [електронний ресурс] –
<https://docs.flutter.dev/>
3. What is the Dart programming language? [електронний ресурс] -
<https://inlab.fib.upc.edu/en/blog/what-dart-programming-language>.
4. Flutter packages [електронний ресурс] –
<https://pub.dev/>
5. Bloc [електронний ресурс] –
<https://bloclibrary.dev/#/gettingstarted>.
6. Bootstrap docs [електронний ресурс] –
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
7. Bootstrap Table docs [електронний ресурс] –
<https://bootstrap-table.com/docs/getting-started/introduction/>
8. Flask docs [електронний ресурс] –
<https://flask.palletsprojects.com/en/2.3.x/>
9. Visual Studio Code [електронний ресурс] -
https://en.wikipedia.org/wiki/Visual_Studio_Code
10. Освітні програми НУВНП [електронний ресурс] -
<https://start.nuwm.edu.ua/osvitni-prohramy>