

Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Кваліфікаційна робота магістра
з теми: **«Розробка та програмна реалізація моделі
інформаційної системи здобувачів вищої освіти»**

Виконав: здобувач вищої освіти групи KN1-M24
спеціальності 122 Комп'ютерні науки
Шевчук Олександр Володимирович

Керівник: Пилипюк Тетяна Михайлівна, кандидат
фізико-математичних наук, доцент

Рецензент:

м. Кам'янець-Подільський – 2025 р.

ЗМІСТ

АНОТАЦІЯ.....	3
ВСТУП.....	5
РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ РІШЕНЬ	8
1.1 Змістовна постановка задачі	8
1.2 Порівняльний аналіз існуючих рішень	9
Висновки до розділу 1	10
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	11
2.1 Клієнт-серверна архітектура.....	11
2.2 Модель інформаційної системи.....	12
2.3 Інтеграція освітніх даних	15
2.4 Технологічний стек	20
Висновки до розділу 2.....	22
РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	23
3.1 Сховища даних	23
3.2 Клієнтські частини	26
3.2.1 Адміністративна панель	26
3.2.2 Мобільний застосунок.....	30
3.3 Серверна частина.....	39
Висновки до розділу 3.....	42
РОЗДІЛ 4 РОЗГОРТАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	43
4.1 Адміністративна панель і серверна частина	43
4.2 Мобільний застосунок.....	45
Висновки до розділу 4.....	45
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48

АНОТАЦІЯ

У даній кваліфікаційній роботі магістра представлено розробку моделі інформаційної системи здобувачів вищої освіти, що полегшує моніторинг освітнього процесу. Визначено пріоритетні складові цього процесу для здобувачі та описано інтеграцію освітніх даних в інформаційну систему у комплексі з АСУ «ВНЗ» через вебсервер для доступу до розкладу занять та електронний кабінет здобувача для доступу до індивідуальної навчальної інформації, а також за допомогою вебсторінок освітнього закладу.

Розроблена модель інформаційної системи складається з панелі адміністратора для керування системою та мобільного застосунку для моніторингу освітнього процесу здобувачами вищої освіти.

Під час створення цієї системи використано вебтехнології HTML і Tailwind CSS, мови програмування JavaScript, PHP і Kotlin, а також системи керування базами даних MySQL і SQLite. Для роботи із зазначеними технологіями застосовувалися середовища розробки VSCode та Android Studio.

Крім того, у кваліфікаційній роботі проаналізовано наявні мобільно-орієнтовані інформаційні системи, що отримують дані з різних автоматизованих систем, призначені для здобувачів вищої освіти. Висвітлено процес тестування компонентів інформаційної системи, розгортання вебмодуля системи на хостинг-провайдера, поширення версій мобільного застосунку через керування релізами в GitHub-репозиторії та встановлення даного застосунку на мобільний пристрій.

Ключові слова: моніторинг освітнього процесу, тайм-менеджмент освітнього процесу, студенто-орієнтована модель, цифровізація навчання, інтеграція освітніх даних, АСУ «ВНЗ».

ABSTRACT

This master's thesis presents the development of a students' information system model that facilitates the monitoring of the educational process. The thesis identifies the priority components of the educational process for students and describes the integration of educational data into the information system in combination with the «VNZ» automated management system, through a web server for access to the class schedule, and the student's electronic office for access to personal educational information, as well as through the university's web pages.

The developed information system model consists of an administrative panel for system management and a mobile application that allows students to monitor the educational process.

Web technologies HTML and Tailwind CSS, programming languages JavaScript, PHP and Kotlin, as well as database management systems MySQL and SQLite were used to create this system. The VSCodium and Android Studio development environments were used to work with these technologies.

Additionally, the thesis analyses existing mobile-oriented information systems designed for students that receive data from various automated systems. The process of testing information system components, deploying the system's web module on a hosting provider, distributing mobile application versions through release management in the GitHub repository, and installing the application on a mobile device is examined.

Keywords: monitoring of the educational process, time management of the educational process, student-oriented model, digitisation of education, integration of educational data, automated management system «VNZ».

ВСТУП

Актуальність. Відслідковування здобувачами власного освітнього процесу є важливим елементом сучасної освіти, що набуває особливого значення у формуванні ініціативності в навчанні. Здобувачі вищої освіти мають не лише пасивно сприймати інформацію, а й активно брати участь у своєму навчанні, відстежуючи та оцінюючи його ефективність. В результаті, це сприяє кращому розумінню своїх сильних і слабких сторін, розвитку самодисципліни, мотивації, встановленню амбітних цілей та формуванню навичок самоменеджменту для професійної й особистісної реалізації в майбутньому.

Відсутність простих і водночас ефективних інструментів забезпечення такого відслідковування, зокрема через планування та моніторинг освітньої діяльності здобувачів вищої освіти, ускладнює процес їхньої адаптації, особливо першокурсників, до освітнього процесу. Як наслідок, призводячи до низького рівня організованості, нераціонального використання часу та навчальних ресурсів. Слід зазначити, що для моніторингу здобувачі вищої освіти можуть використовувати офіційні вебсторінки навчального закладу, однак нерідко використання цих сторінок з мобільних пристроїв створює певні труднощі. Зазвичай їхня структура ускладнена й перенасичена численними елементами, що уповільнює пошук потрібних даних та змушує здобувачів регулярно переглядати окремі розділи, витрачаючи додатковий час на знаходження актуальної інформації. Для планування ж вони можуть застосовувати спеціалізовані застосунки чи онлайн-сервіси, які дають змогу встановлювати нагадування та контролювати прогрес виконання освітніх завдань. Отже, найзручнішим рішенням, зокрема для першокурсників, є використання централізованої платформи, яка містить усю необхідну інформацію про їхній освітній процес і дозволяє відслідковувати його.

Найбільш основними складовими освітнього процесу для здобувачів є:

1. Освітньо-професійна програма та структурований навчальний план, розроблені для підготовки фахівців за певною спеціальністю. Така програма

містить комплекс освітніх компонентів, спланованих і організованих для досягнення результатів навчання.

2. Графік освітнього процесу – визначає календарні терміни теоретичної й практичної підготовки, підсумкового контролю (заліки, екзамени), практики тощо, який складається на кожен семестр.

3. Розклад навчальних занять і екзаменаційної сесії – графік, який визначає дату і час проведення теоретичного та практичного навчання, а також екзаменів в освітньому закладі.

4. Робочі програми освітніх компонентів, силабуси – документи, які визначають зміст, цілі, завдання, форми контролю, оцінювання освітніх компонентів. Вони допомагають викладачам структурувати навчальний процес і забезпечують здобувачів чітким розумінням того, що від них очікується.

5. Відомості заліково-екзаменаційної сесії – підсумкові навчальні відомості поточного семестру, які містять результати досягнень освітніх компонентів здобувачами вищої освіти.

6. Рейтинг здобувачів вищої освіти – система оцінювання та ранжування успішності здобувачів на основі їхніх академічних досягнень для моніторингу та аналізу якості навчання.

Зрештою, при моніторингу освітнього процесу варто зосереджувати увагу на цих ключових аспектах. Адже важливим є використання розкладу навчальних занять, який зазвичай коригують протягом навчального року. Крім того, графік освітнього процесу доповнює план навчального процесу визначеними вибірковими освітніми компонентами, що формується на основі освітньо-професійної програми й навчального плану до початку кожного навчального року. Також систематичний моніторинг аудиторних годин й заліково-екзаменаційних відомостей допомагає здобувачам контролювати вчитку та підсумкову успішність.

Об'єктом дослідження є процес планування та моніторингу освітньої діяльності здобувачів вищої освіти в навчальних закладах, а **предметом** –

методи розробки та функціональні можливості інформаційних систем для реалізації моніторингу здобувачами вищої освіти власного навчального процесу з інтеграцією ресурсів освітнього закладу.

Мета полягає в розробці інформаційної системи для моніторингу освітнього процесу здобувачів вищої освіти з доступом до основних складових освітнього процесу та актуальної інформації про цей процес і навчальний заклад.

Для досягнення мети необхідно виконати такі **завдання**:

- оглянути існуючі рішення з аналізом їхніх переваг та недоліків;
- визначити основні засоби для розробки інформаційної системи;
- описати процес проєктування інформаційної системи;
- реалізувати інтеграцію освітніх даних в інформаційну систему;
- розробити адміністративну панель для закладу освіти;
- створити мобільний застосунок для здобувачів вищої освіти;
- реалізувати серверну частину для доступу до даних;
- протестувати та розгорнути інформаційну систему.

Практичне значення. Кваліфікаційна робота магістра полягає у створенні повноцінної моделі інформаційної системи здобувачів вищої освіти, спрямованої на відслідковування їхнього освітнього процесу, що сприяє ефективній взаємодії з освітнім закладом та цифровізації навчання.

Апробація результатів. Результати дослідження були оприлюднені у Віснику з фізико-математичних наук Кам'янець-Подільського національного університету імені І. Огієнка 2024 року [20] та на науковій конференції здобувачів вищої освіти за підсумками НДР у 2024-2025 н. р. 9-10 квітня 2025 року м. Кам'янець-Подільський [21].

Структура роботи. Кваліфікаційна робота магістра складається зі вступу, 4 розділів, висновків та списку використаних джерел, що налічує 34 найменувань, з яких 13 англomовних. Загальний обсяг роботи містить 50 сторінок, 42 рисунків і 10 таблиць.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ РІШЕНЬ

1.1 Змістовна постановка задачі

Модель інформаційної системи забезпечує здобувачів цифровим інструментом для моніторингу власного освітнього процесу. Вона автоматизує отримання освітніх даних, зокрема через реалізовану інтеграцію з компонентами АСУ «ВНЗ». Інформаційна система складається з двох основних компонентів – адміністративної панелі і мобільного застосунку.

Адміністративна панель відповідає за комплексне управління інформаційною системою на рівні всього університету. Вона дозволяє наповнювати і маніпулювати даними загальної інформації, що стосуються університету, окремих факультетів та їхніх вебджерел, а також здійснювати контроль доступу до системи, наприклад, обмежуючи певних користувачів. Завдяки інтеграції з АСУ «ВНЗ» додаткове введення даних про освітній процес та здобувачів виконувати не потрібно, бо необхідна інформація підтягується автоматично при кожному зверненні здобувача до інформаційної системи. Проте відповідальність за актуальність цих даних, внесених в функціонуючу автоматизовану систему, покладено на деканат, адже здобувачі даного факультету отримують доступ до інформації пов'язаної лише з ним. Адміністратор панелі персоналізує інформаційний контент, використовуючи вебджерела, надані факультетами для здобувачів вищої освіти.

Мобільний застосунок – цифровий студенто-орієнтований інструмент для моніторингу освітнього процесу. Як було викладено раніше, зокрема графік освітнього процесу, розклад навчальних занять, рейтинг успішності, заліково-екзаменаційні відомості є основними складовими освітнього процесу здобувача вищої освіти, які повинні відображатися в застосунку. Насамперед він авторизується через електронний кабінет здобувача освіти, який реалізований автоматизованою системою управління «ВНЗ», попередньо зареєструвавшись в ньому. Надалі отримує основну інформацію про себе, освітній процес і заклад при вході у мобільний застосунок.

1.2 Порівняльний аналіз існуючих рішень

Щоб проаналізувати програмні рішення, спершу потрібно скласти критерії, тому нижче зазначено критерії, за якими здійснювався аналіз.

1. Відображення фільтрованого розкладу – надає можливість переглядати актуальний розклад навчальних занять у відфільтрованому вигляді за різними параметрами (наприклад, показ розкладу за певним днем, приховування обраних підгруп та освітніх компонентів).

2. Моніторинг аудиторних годин – дозволяє відстежувати проведені заняття в аудиторіях за період семестру, тим самим контролюючи вчитку власних занять.

3. Перегляд особистої інформації та підсумкової успішності – надає можливість ознайомлюватися з наявними даними про себе та підсумковими навчальними відомостями за увесь навчальний період.

4. Планування завдань – допомагає організувати та контролювати власне навчання на основі поставлених завдань.

5. Перегляд вебджерел освітнього закладу – забезпечує доступ до актуальної та корисливої інформації, пов'язаної з освітнім закладом.

6. Сповіщення про події – забезпечує своєчасне інформування про важливі події (наприклад, зміни в розкладі, терміни виконання завдань).

7. Кросплатформленість – гарантує доступність програмного рішення на різних пристроях й операційних системах (Android та iOS).

8. Автоматизація – зменшує необхідність ручного введення інформації в програмне рішення.

Оглянувши ринок існуючих програмних рішень серед мобільно-орієнтованих інформаційних систем було складено таблицю 1.1. Вона демонструє порівняння за критеріями, які задовольняють («+»), не задовольняють («-») та частково задовольняють («±») рішення відносно очікуваного результату. Також таблиця містить інформацію про функціональні можливості мобільного застосунку інформаційної системи під назвою МЗІС.

Таблиця 1.1. Порівняльний аналіз безкоштовних програмних рішень

	<i>Критерій I</i>	<i>Критерій II</i>	<i>Критерій III</i>	<i>Критерій IV</i>	<i>Критерій V</i>	<i>Критерій VI</i>	<i>Критерій VII</i>	<i>Критерій VIII</i>
eUniversity [24]	±	–	+	–	+	–	+	+
My Study Life [26]	±	–	±	+	–	±	+	–
Skedy	±	–	–	–	–	–	–	+
UniSched [32]	+	–	–	–	+	–	+	+
uSched [33]	+	–	–	–	–	–	–	+
АСУ ВНЗ [22]	±	–	–	–	–	–	+	+
МЗІС	+	+	+	+	+	±	–	+

З табл. 1.1 видно, що існуючі мобільно-орієнтовані інформаційні системи, які отримують дані з певних автоматизованих систем, таких як АСУ «МКР» [4] (Skedy), ПП «Деканат» [15] (UniSched), АСУ «ВНЗ» [3] (uSched, АСУ ВНЗ), в основному зосереджені на роботі з розкладом занять і недостатньо орієнтовані на інформаційну підтримку здобувачів, а саме:

- відстеження аудиторних годин та фільтрування розкладу;
- огляд підсумкової успішності і планування персональних завдань;
- сповіщення про події та отримання своєчасно актуального розкладу;
- доступ до вебджерел освітнього закладу.

Висновки до розділу 1

У початковому розділі сформульовано постановку задачі, яка чітко визначає основні вимоги для досягнення мети кваліфікаційної роботи. Описано функціональні можливості основних компонентів інформаційної системи, а саме адміністративної панелі для керування системою та мобільного застосунку для моніторингу освітнього процесу. Крім того, проведено порівняльний аналіз безкоштовних мобільно-орієнтованих рішень, який виявив їхню обмежену функціональність відносно встановлених критеріїв, а також можливостей мобільного застосунку інформаційної системи для підтримки здобувачів в освітньому середовищі.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Клієнт-серверна архітектура

Архітектура клієнт-сервер – це мережева модель, у якій дві сутності, клієнт і сервер, взаємодіють між собою для виконання певних дій. Клієнт – програма, яка надсилає запити на отримання даних. Сервер – програма, яка обробляє ці запити, отримує відповідну інформацію та повертає її клієнту. Розрізняють дворівневу, трирівневу та багаторівневу клієнт-серверну архітектуру. У дворівневій архітектурі сервер приймає запит, самостійно його обробляє та надсилає відповідь клієнту без залучення сторонніх ресурсів. У трирівневій архітектурі між клієнтом і сервером використовується проміжний рівень, тоді як у багаторівневій – декілька таких рівнів. Вся бізнес-логіка реалізується на проміжному рівні, що розміщується на окремому сервері. Тобто суть багаторівневої архітектури полягає в тому, що запит клієнта обробляється декількома серверами [27; 11, с. 45-47].

Існують стандартизовані протоколи, які забезпечують зв'язок між клієнтом і сервером. Це протоколи передавання гіпертексту (HTTP), файлів (FTP), електронної пошти (SMTP) [27]. На основі протоколу HTTP працює Всесвітня павутина, що дає змогу користувачам відвідувати сайти через браузер, і переходити між сторінками, обмінюватися повідомленнями тощо. Взаємодія відбувається за принципом запит-відповідь, а ідентифікація ресурсів через глобальні URI. Даний протокол є безстановим, що означає, що кожен запит обробляється самостійно, без збереження інформації про попередні взаємодії. Основними методами HTTP-запитів є GET, POST, PUT, DELETE, які використовуються для отримання, надсилання, оновлення та видалення ресурсів відповідно. Кожен запит і відповідь супроводжуються заголовками, що дають змогу клієнту та серверу надсилати додаткову інформацію. Заголовки можуть містити дані про тип вмісту, метод автентифікації, код стану, спосіб кодування. Оскільки HTTP-протокол передає незашифровані дані, які можуть бути перехоплені, для підвищення безпеки

його розширили до HTTPS. Він поєднує запити й відповіді HTTP з TLS або SSL для шифрування переданих даних, забезпечуючи їхню конфіденційність та цілісність [19, с. 9-10; 16, с. 19-22].

Отже, загальна архітектура інформаційної системи складається з таких елементів (рис. 2.1):

1. Клієнтська частина – адміністративна панель і мобільний застосунок.
2. Серверна частина – доступ до даних.
3. Сховище даних – серверна база даних.

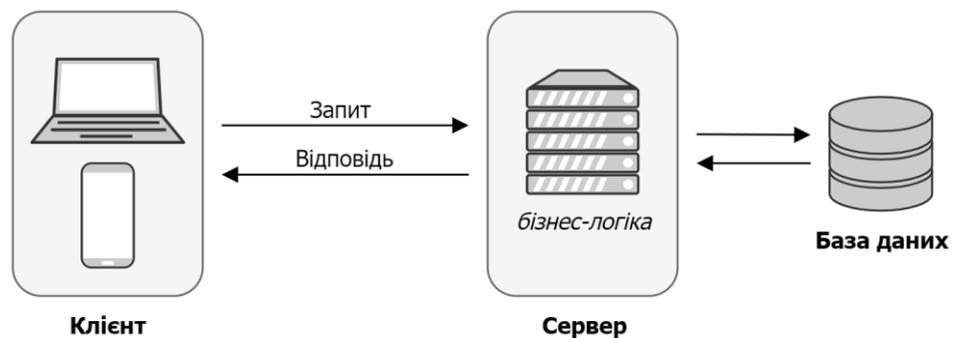


Рис. 2.1. Клієнт-серверна архітектура інформаційної системи

2.2 Модель інформаційної системи

UML – графічна мова моделювання, призначена для специфікації, візуалізації, проектування й документування артефактів програмних систем. Вона функціонує у парадигмі об’єктно-орієнтованого програмування, де будь-яка система розглядається як сукупність взаємодіючих об’єктів [1, с. 74-76].

Діаграма прецедентів є найпростішою серед поведінкових UML-діаграм, що відображає відношення між акторами (дійовими особами) і прецедентами (варіантами використання) у майбутній системі.

Діаграма послідовностей формалізує сценарій використання, зображуючи на єдиній часовій осі для певного набору об’єктів їхній життєвий цикл та взаємодію (надсилання запитів і отримання відповідей).

Діаграма діяльності візуалізує послідовність виконання дій подібно до блок-схеми, демонструючи, як одна дія призводить до іншої.

На рисунках 2.2-2.3 подано функціонал основних компонентів інформаційної системи через діаграму прецедентів, а на рисунках 2.4-2.5 – процес авторизації здобувачів вищої освіти в мобільному застосунку інформаційної системи за допомогою фрагментів діаграм послідовностей та діяльності.

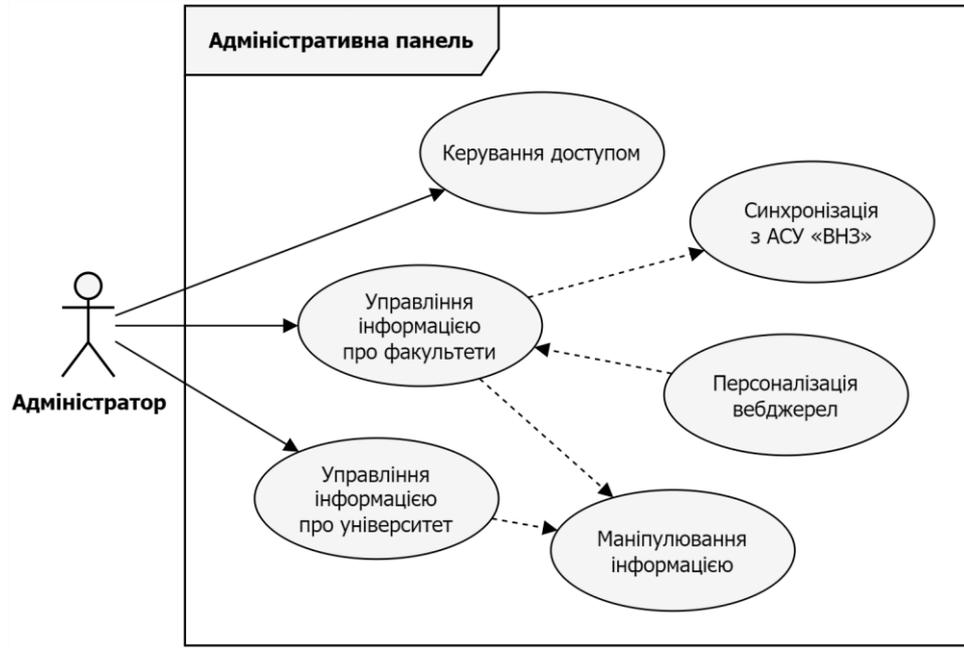


Рис. 2.2. Діаграма прецедентів адміністративної панелі

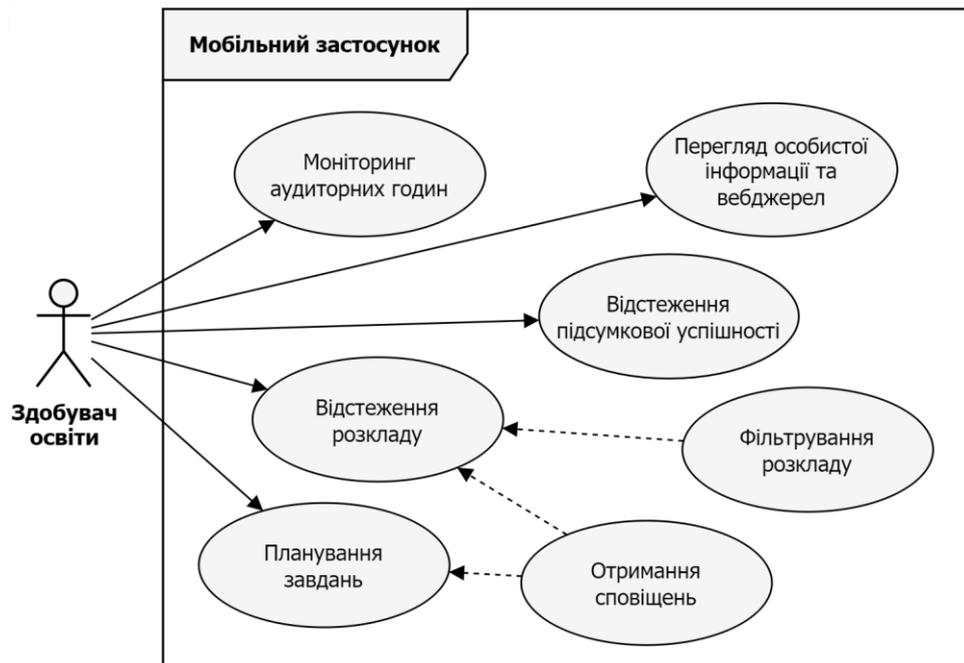


Рис. 2.3. Діаграма прецедентів мобільного застосунку

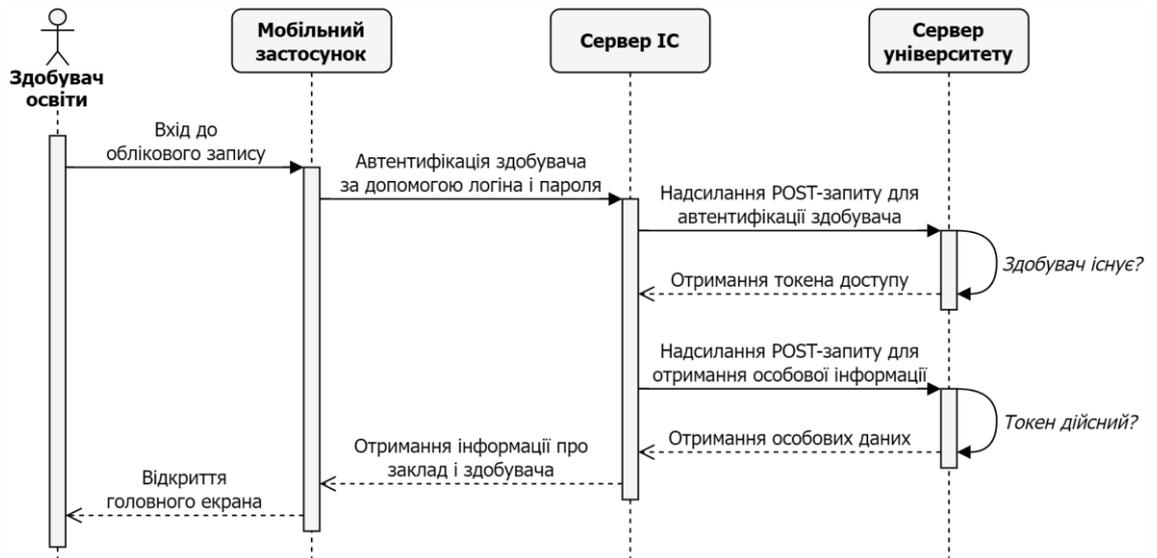


Рис. 2.4. Діаграма послідовності авторизації в мобільному застосунку

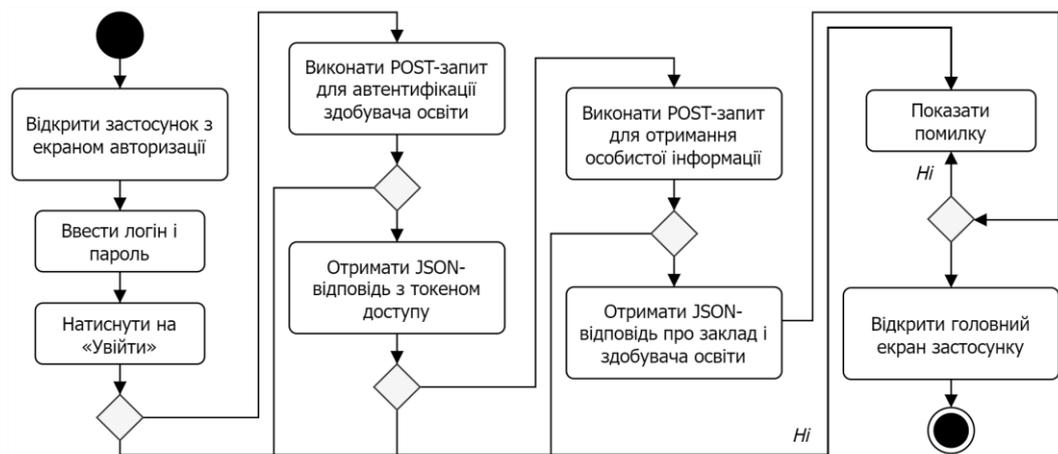


Рис. 2.5. Діаграма діяльності загальної авторизації в застосунку

У загальному вигляді можна виокремити такі компоненти інформаційної системи (рис. 2.6): мобільний застосунок, адміністративна панель, серверна частина і сховище даних, а додаткові – вебсервери університету і АСУ «ВНЗ», програма АС «Деканат».

Серверна частина інформаційної системи відповідає за забезпечення зв'язку між клієнтською частиною, сховищем даних та зовнішніми джерелами. Вона використовує прикладний програмний інтерфейс (API), який слугує каналом доступу до інформації для клієнтської частини. Також серверна частина охоплює бізнес-логіку системи, зокрема містить логіку взаємодії з серверною базою даних. Завдяки цьому існує певний рівень безпеки, оскільки клієнт позбавлений прямого доступу до даних.

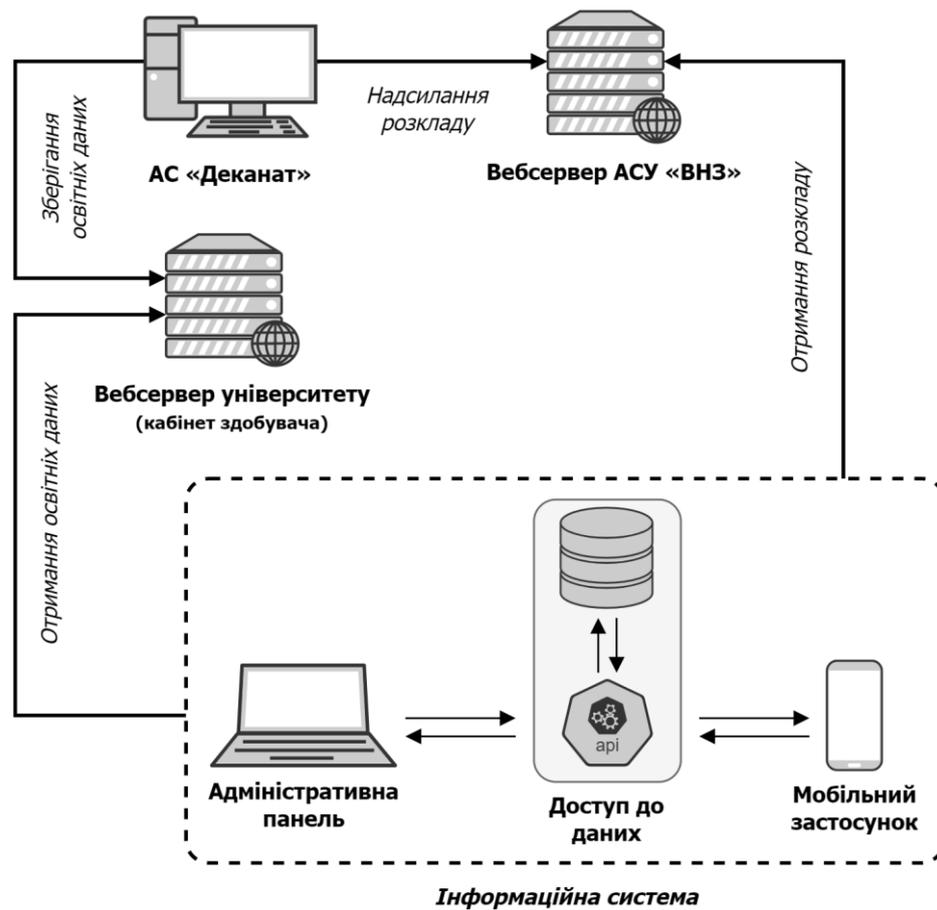


Рис. 2.6. Модель інформаційної системи

2.3 Інтеграція освітніх даних

Сьогодні більшість закладів вищої освіти має певне програмне забезпечення для автоматизації освітнього документообігу. У Кам'янець-Подільському національному університеті ім. І. Огієнка використовується автоматизована система управління «Вищий навчальний заклад», розроблена Науково-дослідним інститутом прикладних інформаційних технологій Кібернетичного центру Національної академії наук України. Вона призначена для підвищення ефективності управління освітнім закладом. Основними компонентами цієї системи є АС «Приймальна комісія», АС «Деканат» і АС «Студмістечко», а додатковими – АС «Тест», АС «Конструктор звітів», «Менеджер резервних копій», «Веб-розклад», «Веб Деканат» і «Система контролю доступу» [5].

Автоматизована система «Деканат» – це програмно-технологічний комплекс управління освітнім процесом закладу освіти, призначений для

організації роботи методистів та зменшення кількості документації на паперових носіях [2]. Дана АС функціонує на факультетах університету та забезпечує, зокрема створення, керування й поширення розкладу. Оскільки розповсюдження розкладу в основному відбувається через вебсервер АСУ «ВНЗ», отримання розкладу здійснюється шляхом виконання запитів до цього вебсервера. Важливо зазначити, що дана АСУ немає публічної документації API для отримання даних з вебсервера, тому дослідивши мережевий трафік, який використовується під час відображення розкладу на сайті, було знайдено приклади запитів для подальшого використання (рис. 2.7). Для цього скористався демонстраційним варіантом розкладу, що пропонує сама система [6]. Щоб проаналізувати мережеву активність на сайті, використав браузер Chrome, відобразив інструменти розробника та відкрив мережеву вкладку.

Окрім розкладу, аналогічним чином отримуються дані про особову інформацію здобувачів вищої освіти та їхні підсумкові навчальні результати за увесь період, які зафіксовані в АС «Деканат» і доступні через електронний кабінет здобувача вищої освіти «Веб Деканат», що розгорнутий на окремому вебсервері університету. Даний вебресурс є додатковим компонентом АСУ «ВНЗ», який призначений спростити процеси, пов'язані з навантаженням та успішністю здобувача, а також екзаменаційними відомостями й розкладом викладача через відповідні електронні кабінети [13].

Сформовані запити, інтегровані як зовнішні джерела в інформаційну систему, зображено на рисунках 2.8-2.19 з описом їхніх даних та адрес.

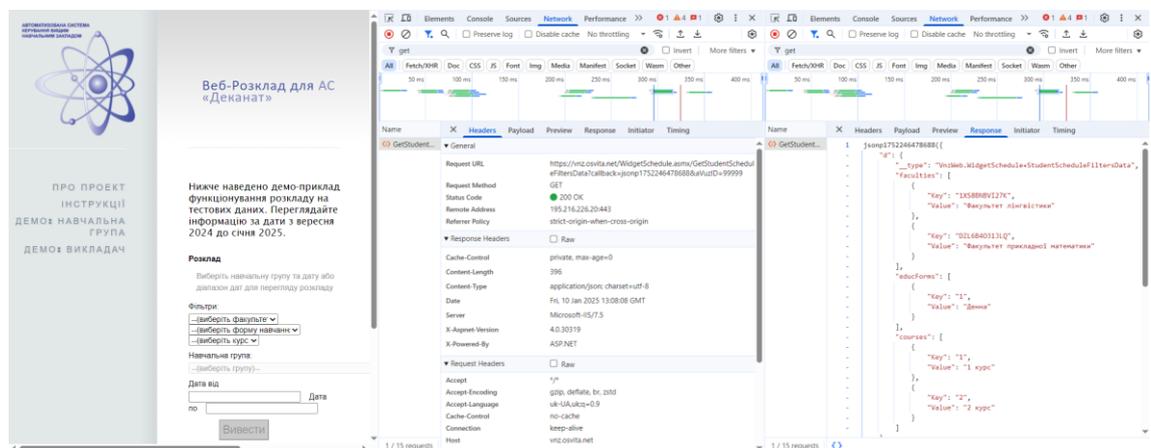


Рис. 2.7. Приклад знайденого запиту та відповіді на нього

GET `{{(base_url)}} /Faculties?VuzId= {{(vuz_id)}}` Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	VuzId	{{(vuz_id)}}	Код закладу		

Body Cookies Headers (9) Test Results 200 OK • 24 ms • 2.73 KB Save Response

{ } JSON Preview Visualize

	facultyDescription	name	facultyLongName	facultyAddress	id	facultyEngName
0	null	FK_PE	Природничо-економічний факультет	null	UZ76XHND0S93	null
1	null	FK_FIZMAT	Фізико-математичний факультет	м. Кам'янець-Подільський, вул. Уральська, 1	Q1W2ZPSPV13A	The Faculty of Physics and Mathematics
2	null	FK_FIZKILT	Факультет фізичної культури	м. Кам'янець-Подільський, вул.Огієнка, 61	6GQC4YG48ATU	The Faculty of Physical Culture

Рис. 2.8. Запит на отримання факультетів

GET `{{(base_url)}} /EducationForms?VuzId= {{(vuz_id)}}` Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	VuzId	{{(vuz_id)}}	Код закладу		

Body Cookies Headers (9) Test Results 200 OK • 49 ms • 288 B Save Response

{ } JSON Preview Visualize

	id	name
0	1	Денна

Рис. 2.9. Запит на отримання форм навчання

GET `{{(base_url)}} /Courses?VuzId= {{(vuz_id)}}` Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	VuzId	{{(vuz_id)}}	Код закладу		

Body Cookies Headers (9) Test Results 200 OK • 47 ms • 376 B Save Response

{ } JSON Preview Visualize

	id	name
0	1	1 курс
1	2	2 курс
2	3	3 курс

Рис. 2.10. Запит на отримання курсів

GET `{{(base_url)}} /Chairs?VuzId= {{(vuz_id)}} &FacultyID=Q1W2ZPSPV13A` Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	VuzId	{{(vuz_id)}}	Код закладу		
<input checked="" type="checkbox"/>	FacultyID	Q1W2ZPSPV13A	Код факультету		

Body Cookies Headers (9) Test Results 200 OK • 32 ms • 615 B Save Response

{ } JSON Preview Visualize

	id	name	chairShortName	facultyId
0	RYE1R1P6MDQA	Кафедра комп'ютерних наук	_19	Q1W2ZPSPV13A
1	RB148KWNIM3Z	Кафедра математики	_15	Q1W2ZPSPV13A
2	UZFF7VQDKEB3	Кафедра фізики	_18	Q1W2ZPSPV13A

Рис. 2.11. Запит на отримання кафедр

GET `{{base_url}}/Employees?VuzId={{vuz_id}}&FacultyID=Q1W2ZPSPV13A&ChairID=RYE1R1P6MDQA` Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
VuzId	{{vuz_id}}	Код закладу	
FacultyID	Q1W2ZPSPV13A	Код факультету	
ChairID	RYE1R1P6MDQA	Код кафедри	

Body Cookies Headers (9) Test Results 200 OK · 29 ms · 1.83 KB Save Response

{ } JSON Preview Visualize

id	lastName	firstName	middleName
0	E54RVRWD4S8X	Бараннік	Володимир Вікторович
1	OKDPDQQPE38C	Іванюк	Віталій Анатолійович
2	3WAEVS8UUM9M	Кух	Оксана Михайлівна

Рис. 2.12. Запит на отримання викладачів

GET `{{base_url}}/StudyGroups?VuzId={{vuz_id}}&FacultyID=Q1W2ZPSPV13A&EducationFormId=1&CourseId=1` Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
VuzId	{{vuz_id}}	Код закладу	
FacultyID	Q1W2ZPSPV13A	Код факультету	
EducationFormId	1	Код форми навчання	
CourseId	1	Код курсу	

Body Cookies Headers (9) Test Results 200 OK · 29 ms · 922 B Save Response

{ } JSON Preview Visualize

id	name	chairId	facultyId	educationFormId
0	24B9D91JMTL	F1-B25	UZFF7VQDKEB3	Q1W2ZPSPV13A
1	CIXILQFMB01P	F1-M25	UZFF7VQDKEB3	Q1W2ZPSPV13A
2	ATN9U1BNA485	KN1-B25	RYE1R1P6MDQA	Q1W2ZPSPV13A

Рис. 2.13. Запит на отримання академічних груп

GET `{{base_url}}/StudyTime?VuzId={{vuz_id}}` Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
VuzId	{{vuz_id}}	Код закладу	

Body Cookies Headers (9) Test Results 200 OK · 23 ms · 2.29 KB Save Response

{ } JSON Preview Visualize

id	name	shortName	begin	end
0	55793	08_30	08_30	1900-01-01T08:30:00
1	83598	08_45	08_45	1900-01-01T10:20:00
2	6985	10_10	10_10	1900-01-01T11:30:00

Рис. 2.14. Запит на отримання тривалості занять

GET `.../Schedule?VuzId={{vuz_id}}&IsByEmployeeNotByGroup=true&EmployeeOrGroup=OKDPDQQPE38C&StartDate=2025-09-01&EndDate=2025-09-07` Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description
VuzId	{{vuz_id}}	Код закладу
IsByEmployeeNotByGroup	true	Відобразити розклад викладача
EmployeeOrGroup	OKDPDQQPE38C	Код викладача
StartDate	2025-09-01	Дата початку розкладу
EndDate	2025-09-07	Дата кінця розкладу

Body Cookies Headers (9) Test Results 200 OK • 49 ms • 2.03 KB Save Response

{ } JSON Preview Visualize

date	lastRefreshDate	lessons																			
2025-09-02T00:00:00	2025-10-16T14:59:28	<table border="1"><thead><tr><th>studyTimeId</th><th>data</th></tr></thead><tbody><tr><td>6986</td><td><table border="1"><thead><tr><th>studyGroupId</th><th>studyGroupName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody><tr><td>0</td><td>QQ85GQ7Z7MHS</td><td>KN1-M24</td><td>8F6JC9RSDALR</td><td>176358</td><td>Сучасні методи аналізу даних з використанням R</td><td>40</td><td>Лекція</td></tr></tbody></table></td></tr></tbody></table>	studyTimeId	data	6986	<table border="1"><thead><tr><th>studyGroupId</th><th>studyGroupName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody><tr><td>0</td><td>QQ85GQ7Z7MHS</td><td>KN1-M24</td><td>8F6JC9RSDALR</td><td>176358</td><td>Сучасні методи аналізу даних з використанням R</td><td>40</td><td>Лекція</td></tr></tbody></table>	studyGroupId	studyGroupName	id	disciplineId	disciplineName	studyTypeId	studyType	0	QQ85GQ7Z7MHS	KN1-M24	8F6JC9RSDALR	176358	Сучасні методи аналізу даних з використанням R	40	Лекція
studyTimeId	data																				
6986	<table border="1"><thead><tr><th>studyGroupId</th><th>studyGroupName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody><tr><td>0</td><td>QQ85GQ7Z7MHS</td><td>KN1-M24</td><td>8F6JC9RSDALR</td><td>176358</td><td>Сучасні методи аналізу даних з використанням R</td><td>40</td><td>Лекція</td></tr></tbody></table>	studyGroupId	studyGroupName	id	disciplineId	disciplineName	studyTypeId	studyType	0	QQ85GQ7Z7MHS	KN1-M24	8F6JC9RSDALR	176358	Сучасні методи аналізу даних з використанням R	40	Лекція					
studyGroupId	studyGroupName	id	disciplineId	disciplineName	studyTypeId	studyType															
0	QQ85GQ7Z7MHS	KN1-M24	8F6JC9RSDALR	176358	Сучасні методи аналізу даних з використанням R	40	Лекція														
2025-09-05T00:00:00	2025-10-16T14:59:28	<table border="1"><thead><tr><th>studyTimeId</th><th>data</th></tr></thead><tbody><tr><td>55793</td><td><table border="1"><thead><tr><th>studyGroupId</th><th>studyGroupName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody><tr><td>0</td><td>QQ85GQ7Z7MHS</td><td>KN1-M24</td><td>B3LIPRPWEQ6I</td><td>176358</td><td>Сучасні методи</td><td>40</td><td>Лекція</td></tr></tbody></table></td></tr></tbody></table>	studyTimeId	data	55793	<table border="1"><thead><tr><th>studyGroupId</th><th>studyGroupName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody><tr><td>0</td><td>QQ85GQ7Z7MHS</td><td>KN1-M24</td><td>B3LIPRPWEQ6I</td><td>176358</td><td>Сучасні методи</td><td>40</td><td>Лекція</td></tr></tbody></table>	studyGroupId	studyGroupName	id	disciplineId	disciplineName	studyTypeId	studyType	0	QQ85GQ7Z7MHS	KN1-M24	B3LIPRPWEQ6I	176358	Сучасні методи	40	Лекція
studyTimeId	data																				
55793	<table border="1"><thead><tr><th>studyGroupId</th><th>studyGroupName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody><tr><td>0</td><td>QQ85GQ7Z7MHS</td><td>KN1-M24</td><td>B3LIPRPWEQ6I</td><td>176358</td><td>Сучасні методи</td><td>40</td><td>Лекція</td></tr></tbody></table>	studyGroupId	studyGroupName	id	disciplineId	disciplineName	studyTypeId	studyType	0	QQ85GQ7Z7MHS	KN1-M24	B3LIPRPWEQ6I	176358	Сучасні методи	40	Лекція					
studyGroupId	studyGroupName	id	disciplineId	disciplineName	studyTypeId	studyType															
0	QQ85GQ7Z7MHS	KN1-M24	B3LIPRPWEQ6I	176358	Сучасні методи	40	Лекція														

Рис. 2.15. Запит на отримання розкладу викладача

GET `.../Schedule?VuzId={{vuz_id}}&IsByEmployeeNotByGroup=false&EmployeeOrGroup=QQ85GQ7Z7MHS&StartDate=2025-09-01&EndDate=2025-09-07` Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description
VuzId	{{vuz_id}}	Код закладу
IsByEmployeeNotByGroup	false	Відобразити розклад групи
EmployeeOrGroup	QQ85GQ7Z7MHS	Код групи
StartDate	2025-09-01	Дата початку розкладу
EndDate	2025-09-07	Дата кінця розкладу

Body Cookies Headers (9) Test Results 200 OK • 86 ms • 8.69 KB Save Response

{ } JSON Preview Visualize

date	lastRefreshDate	lessons																												
2025-09-01T00:00:00	2025-10-16T14:59:28	<table border="1"><thead><tr><th>studyTimeId</th><th>data</th></tr></thead><tbody><tr><td>55793</td><td><table border="1"><thead><tr><th>employeeId</th><th>employeeName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody><tr><td>0</td><td>M3QM2UL3XDGT</td><td>Модик Ростислав Васильович</td><td>YJ4HSRD6VWF4</td><td>176357</td><td>Розробка проєктів для адаптивного навчання з використанням штучного інтелекту</td><td>40</td><td>Лекція</td></tr></tbody></table></td></tr><tr><td>6985</td><td><table border="1"><thead><tr><th>employeeId</th><th>employeeName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody></tbody></table></td></tr></tbody></table>	studyTimeId	data	55793	<table border="1"><thead><tr><th>employeeId</th><th>employeeName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody><tr><td>0</td><td>M3QM2UL3XDGT</td><td>Модик Ростислав Васильович</td><td>YJ4HSRD6VWF4</td><td>176357</td><td>Розробка проєктів для адаптивного навчання з використанням штучного інтелекту</td><td>40</td><td>Лекція</td></tr></tbody></table>	employeeId	employeeName	id	disciplineId	disciplineName	studyTypeId	studyType	0	M3QM2UL3XDGT	Модик Ростислав Васильович	YJ4HSRD6VWF4	176357	Розробка проєктів для адаптивного навчання з використанням штучного інтелекту	40	Лекція	6985	<table border="1"><thead><tr><th>employeeId</th><th>employeeName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody></tbody></table>	employeeId	employeeName	id	disciplineId	disciplineName	studyTypeId	studyType
studyTimeId	data																													
55793	<table border="1"><thead><tr><th>employeeId</th><th>employeeName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody><tr><td>0</td><td>M3QM2UL3XDGT</td><td>Модик Ростислав Васильович</td><td>YJ4HSRD6VWF4</td><td>176357</td><td>Розробка проєктів для адаптивного навчання з використанням штучного інтелекту</td><td>40</td><td>Лекція</td></tr></tbody></table>	employeeId	employeeName	id	disciplineId	disciplineName	studyTypeId	studyType	0	M3QM2UL3XDGT	Модик Ростислав Васильович	YJ4HSRD6VWF4	176357	Розробка проєктів для адаптивного навчання з використанням штучного інтелекту	40	Лекція														
employeeId	employeeName	id	disciplineId	disciplineName	studyTypeId	studyType																								
0	M3QM2UL3XDGT	Модик Ростислав Васильович	YJ4HSRD6VWF4	176357	Розробка проєктів для адаптивного навчання з використанням штучного інтелекту	40	Лекція																							
6985	<table border="1"><thead><tr><th>employeeId</th><th>employeeName</th><th>id</th><th>disciplineId</th><th>disciplineName</th><th>studyTypeId</th><th>studyType</th></tr></thead><tbody></tbody></table>	employeeId	employeeName	id	disciplineId	disciplineName	studyTypeId	studyType																						
employeeId	employeeName	id	disciplineId	disciplineName	studyTypeId	studyType																								

Рис. 2.16. Запит на отримання розкладу академічної групи

POST `.../auth_url` Send

Params Authorization Headers (8) Body • Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description
grant_type	password	Тип авторизації
username	{{username}}	Логін здобувача
password	{{password}}	Пароль здобувача
version	2.15	Версія кабінету здобувача

Body Cookies Headers (11) Test Results 200 OK • 45 ms • 727 B Save Response

{ } JSON Preview Visualize

access_token	Ckp55QwKc_Wi6iZD9BJz66lyd7p03g6OCe-DOiZUZsS9GuO7eGjQjIBCkX3o6fnBeceHXK4de862L-YZmZxnTwiVjF_GNTVoUTBw_5DJHULzoXeWz6-Tsk8DsNGSC4DCrIEZ9nIUHnSaHeOS5kh4q0gIZl9OUksSmCbFzFv0RQ_VyTtXa5uUjgkxS4oO_7YUsMj2AeXMiVf8ZFfFn4iQUHnVcng01e0R2QXQI-OvI9VHLmVfN2nATm43IHCG5VXHBxD92rG9QFv1oAcNfClr5hzG49SB4hNqU5PwDexsSrnx-sMNEOqVghF_anhThrmixFDUz2vresq-fm8g
token_type	bearer
expires_in	1209599

Рис. 2.17. Запит на отримання токена доступу до кабінету здобувача

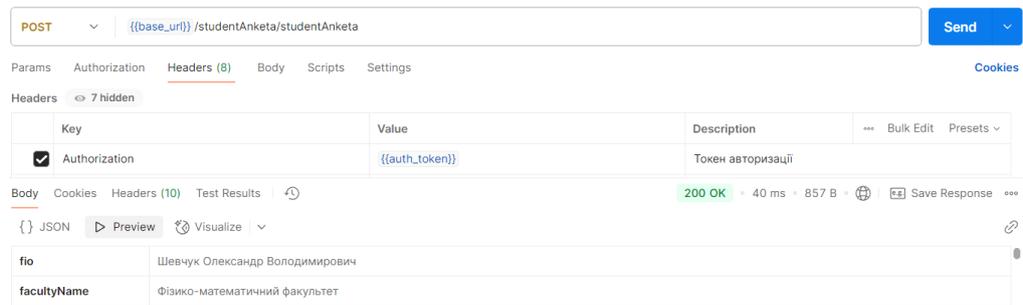


Рис. 2.18. Запит на отримання особистої інформації здобувача

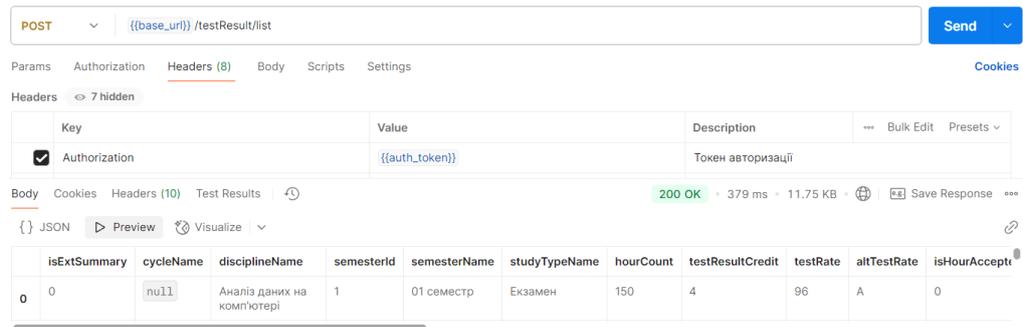


Рис. 2.19. Запит на отримання підсумкових оцінок здобувача

Також хмарним способом отримується інформація про освітню діяльність шляхом витягування даних із RSS-каналів сайту освітнього закладу та показу їхніх вебсторінок через відкриття переданих URL-адрес.

2.4 Технологічний стек

Архітектура системи складається з двох клієнтських і однієї серверної частин. Оскільки кожна частина має різні вимоги, технологічний стек для кожної з них буде відрізнятися, тому було проаналізовано (табл. 2.1) та підібрано відповідний набір основних технологій для їхньої реалізації.

Варто зазначити, що мобільний застосунок інформаційної системи орієнтований на операційну систему Android, бо вона є домінуючою серед мобільних пристроїв в Україні [21] та її застосунки простіше поширювати на Android-пристроях [29].

HTML і CSS – базові технології веброзробки, які відповідають за створення структури та зовнішнього вигляду інтерфейсу користувача. HTML формує каркас вебсторінки, а CSS визначає її стильові характеристики для візуального оформлення. Щоб прискорити роботу з каскадними таблицями стилів, використовують Tailwind CSS – утилітарний CSS-фреймворк, який

дозволяє швидко та гнучко оформлювати вебінтерфейси за допомогою попередньо визначених класів без написання власних стилів з нуля.

Мова програмування JavaScript додає інтерактивність і динамічність до інтерфейсу, забезпечуючи обробку подій, виконання логіки. Мова програмування PHP, у свою чергу, дозволяє створювати динамічні вебсторінки та вебзастосунки, ефективно взаємодіючи з HTML і виконуючи обробку на стороні сервера. MySQL – реляційна система управління базами даних, яка зберігає інформацію на сервері та забезпечує ефективну роботу із запитами до даних.

Kotlin – JVM-орієнтована, статично типізована мова програмування, яка забезпечує високий рівень продуктивності та надійності для розробки різноманітних застосунків. З 2019 року має статус рекомендованої мови для Android-розробки. SQLite – вбудована база даних, що забезпечує локальне зберігання даних у мобільному застосунку.

VSCodium – редактор коду, заснований на вихідних кодах Visual Studio Code, з акцентом на конфіденційність і відкритість програмного забезпечення. Він не містить телеметрії чи інших пропрієтарних компонентів Microsoft.

Android Studio – офіційне середовище розробки для створення Android-застосунків, розробником якого є компанія Google. Воно забезпечує розробників широким набором інструментів для створення, тестування та налагодження застосунків.

Таблиця 2.1. Переваги і недоліки визначених мов програмування

Мова	Переваги	Недоліки
<i>JavaScript</i>	<ul style="list-style-type: none"> – сумісність із різними пристроями та універсальність у використанні для клієнтської й серверної частин; – миттєвість виконання на стороні клієнта та доступність асинхронного програмування; – інтегрованість із HTML і CSS для створення інтерактивних сторінок; – наявність численних бібліотек і фреймворків, таких як Angular, Vue.js, Next.js тощо. 	<ul style="list-style-type: none"> – різноманітність інтерпретації різними браузерами; – нижча продуктивність порівняно з мовами C++ і Java; – видимість коду, обмеженість доступу до файлової системи.

Продовження таблиці 2.1

<i>PHP</i>	<ul style="list-style-type: none"> – легкість в освоєнні завдяки синтаксичній подібності до С; – поєднуваність з іншими технологіями, такими як HTML, CSS і JavaScript; – підтримуваність парадигми ООП і можливість роботи з різними СУБД, наприклад із MySQL, MongoDB, Oracle тощо; – наявність багатьох бібліотек і фреймворків, як-от Laravel, Symfony, CodeIgniter тощо. 	<ul style="list-style-type: none"> – обмеженість для великих систем через проблеми з масштабованістю й продуктивністю; – непослідовність у назвах функцій та їхніх параметрів; – можлива несумісність між різними версіями мови.
<i>Kotlin</i>	<ul style="list-style-type: none"> – сучасність синтаксису, компактність і читабельність коду; – можливість багатопотоковості й асинхронності в програмуванні; – безпечність щодо null-значень, наявність вбудованих функцій, підтримуваність функціонального та об'єктно-орієнтованого програмування; – повна сумісність із Java-кодом завдяки орієнтованості на віртуальну машину Java (JVM). 	<ul style="list-style-type: none"> – нижча швидкість компіляції та більша розмірність бінарного файлу порівняно з Java; – несумісність із ранніми бібліотеками й фреймворками Java; – обмеженість навчальних ресурсів.

Висновки до розділу 2

В другому розділі визначено клієнт-серверну архітектуру інформаційної системи, що складається з клієнтської та серверної частин, а також сховища даних. Спроектовано схему моделі інформаційної системи й UML-діаграм, зокрема діаграми прецедентів основних компонентів системи, які відображають їх ключовий функціонал. Описано процес інтеграції освітніх даних, що здійснюється шляхом отримання даних з таких джерел, як АСУ «ВНЗ» та вебсторінки навчального закладу. Також здійснено вибір технологічного стеку для реалізації моделі інформаційної системи здобувачів вищої освіти із зазначенням переваг і недоліків обраних мов програмування.

РОЗДІЛ 3

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Сховища даних

В інформаційній системі дані розподіляються між серверним і локальним сховищами. Серверне сховище забезпечує централізоване зберігання, керування, доступ клієнтських частин системи до даних через мережу. Локальне сховище, у свою чергу, реалізоване безпосередньо на пристрої користувача для маніпулювання даними мобільного застосунку. Обидва типи сховищ ґрунтуються на реляційній моделі даних. Суть реляційних баз даних полягає в тому, що вони організують дані у вигляді таблиць з чітко визначеними атрибутами та зв'язками, використовуючи зовнішні ключі для забезпечення цілісності даних, а також нормалізацію для усунення надмірності, що дозволяє ефективно керувати даними в конкретній СУБД [14, с. 44-46].

Всього в серверній базі даних реалізовано 3 таблиці, структура яких представлена у таблицях 3.1-3.3. Подана структура відношень містить найменування атрибутів та їхніх призначень, що функціонує в обраній СУБД MySQL. Реляційна схема цієї бази даних, яка визначає відношення (таблиці), їхні атрибути (поля), типи даних, взаємозв'язки між такими відношеннями та обмеження цілісності, зображена на рисунку 3.1.

Таблиця 3.1. Структура таблиці факультетів

Атрибут	Призначення
id	Унікальний ідентифікатор факультету
code	Присвоєний код факультету в АСУ «ВНЗ»
full_name	Назва факультету
address	Адреса факультету
is_blocked	Ознака блокування факультету в системі
last_modified	Дата і час останньої зміни запису

Таблиця 3.2. Структура таблиці контенту факультету

Атрибут	Призначення
id	Унікальний ідентифікатор контенту
faculty_id	Ідентифікатор факультету

title	Заголовок контенту
url	Кодована URL-адреса контенту
category	Номер категорії контенту
last_modified	Дата і час останньої зміни запису

Таблиця 3.3. Структура таблиці налаштувань

Атрибут	Призначення
id	Унікальний ідентифікатор налаштування
name	Унікальна назва параметра налаштування
value	Фактичне значення параметра налаштування

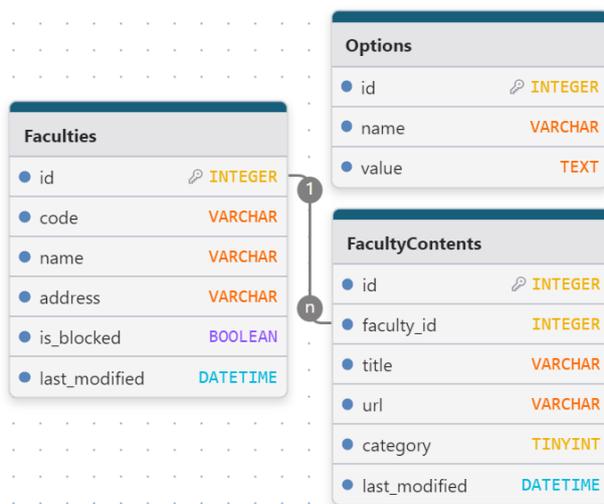


Рис. 3.1. Реляційна схема серверної БД

В локальній базі даних існує всього 5 таблиць. Нижче подано структуру цих таблиць (табл. 3.4-3.8), а також на рис. 3.2 зображено реляційну схему.

Таблиця 3.4. Структура таблиці розкладу

Атрибут	Призначення
id	Унікальний ідентифікатор заняття
date	Дата проведення заняття
time_start	Час початку заняття
time_end	Час завершення заняття
subject	Назва дисципліни
lesson_type	Тип заняття
classroom	Аудиторія, в якій проходить заняття
teacher	ПІБ викладача
subgroup	Підгрупа, для якої призначене заняття

Таблиця 3.5. Структура таблиці фільтра розкладу

Атрибут	Призначення
id	Унікальний ідентифікатор фільтра
target	Номер об'єкта фільтрації
name	Найменування фільтрованого об'єкта

Таблиця 3.6. Структура таблиці завдань

Атрибут	Призначення
id	Унікальний ідентифікатор завдання
name	Назва завдання
description	Опис завдання
start_datetime	Дата і час початку виконання завдання
category_id	Ідентифікатор категорії завдання
priority	Номер рівня пріоритету завдання
is_done	Ознака виконання завдання
is_archive	Ознака архівування завдання
created_at	Дата і час створення запису
updated_at	Дата і час останньої зміни запису

Таблиця 3.7. Структура таблиці категорії завдань

Атрибут	Призначення
id	Унікальний ідентифікатор категорії
name	Найменування категорії завдання

Таблиця 3.8. Структура таблиці налаштувань

Атрибут	Призначення
id	Унікальний ідентифікатор налаштування
name	Унікальна назва параметра налаштування
value	Фактичне значення параметра налаштування

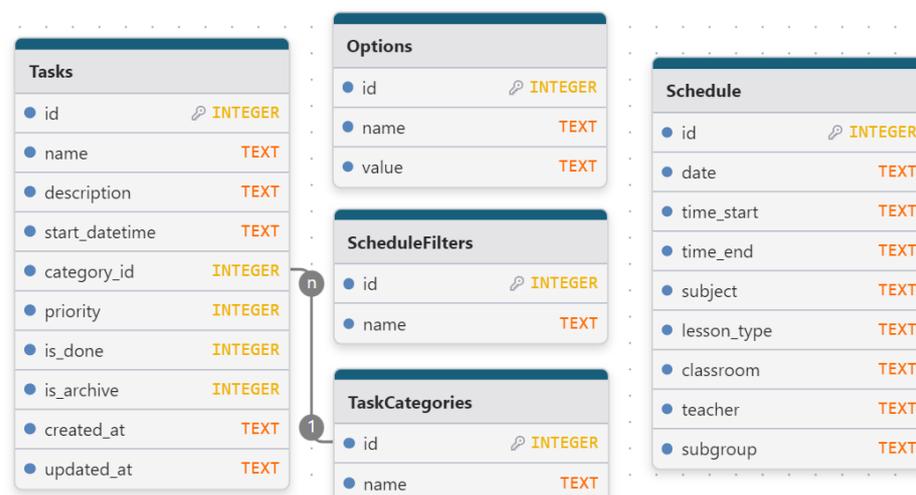


Рис. 3.2. Реляційна схема локальної БД

3.2 Клієнтські частини

Інформаційна система представлена двома клієнтськими частинами: панеллю адміністратора для управління системою, і мобільним застосунком здобувачів вищої освіти для моніторингу освітнього процесу.

3.2.1 Адміністративна панель

Першим кроком у розробці адміністративної панелі стало налаштування робочого середовища. Для цього було запущено VSCodium та створено кореневий каталог проекту із загальною структурою (рис. 3.3), що охоплює статичні ресурси, додаткові PHP-класи, вихідний код проекту, зовнішні бібліотеки, конфігураційні файли тощо. Слід зазначити, що панель адміністратора і серверна частина інформаційної системи взаємопов'язані, тому визначена структура застосовується обома частинами, які надалі будуть розгорнуті на одному вебвузлі.

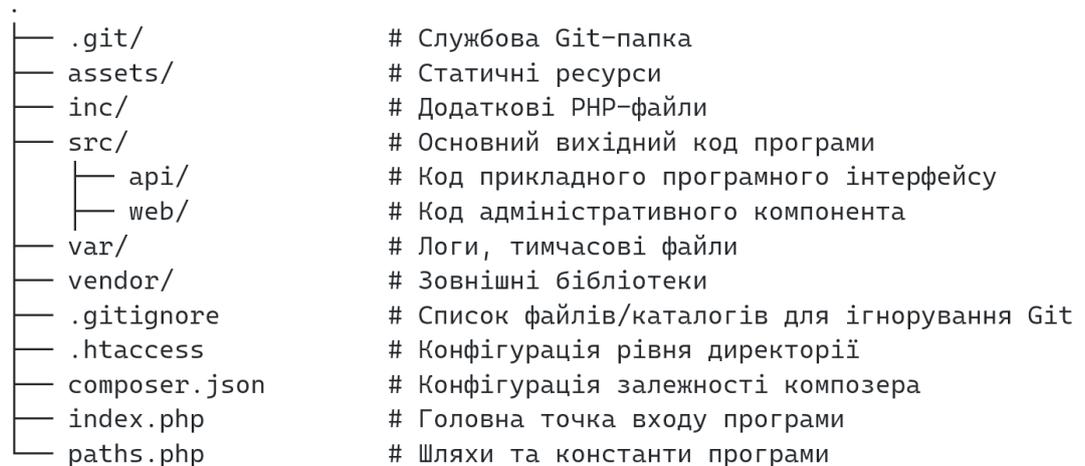


Рис. 3.3. Загальна структура вебмодуля інформаційної системи

У файлі `paths.php` визначено константи для базового та похідних шляхів до директорій, що забезпечує зручний доступ до ресурсів у різних частинах програми, а також виконується автоматичне підключення утилітарних функцій.

В директорії `inc` розміщено набір допоміжних PHP-файлів, кожен з яких виконує чітко визначену роль у функціонуванні вебмодуля, а саме:

- `config.php` містить конфігураційні параметри серверної бази даних та інші налаштування;
- `connection.php` реалізує механізм виконання HTTP-запитів за допомогою бібліотеки `cURL`, включаючи функції надсилання запитів `GET` і `POST`, оброблення помилок з'єднання тощо;
- `database.php` забезпечує взаємодію з серверною базою даних через бібліотеку `PDO`, надаючи функції встановлення з'єднання, виконання `SQL`-запитів, оброблення результатів тощо;
- `utils.php` охоплює набір статичних функцій, призначених для перевірки маршрутів, валідації полів, формування `JSON`-відповідей про успіх або помилку тощо;
- `schedule_api.php` містить опис `API`-запитів до розкладу вебсервера АСУ «ВНЗ», дозволяючи отримувати дані про факультети, кафедри, академічні групи, викладачів, розклади занять тощо;
- `student_api.php` містить опис `API`-запитів до електронного кабінету здобувача вищої освіти АСУ «ВНЗ», забезпечуючи автентифікацію, отримання особової інформації та підсумкових оцінок.

Структура директорії вихідного коду адміністративної панелі (`src/web`) організована наступним чином: тека `components` зберігає шаблони повторного використання, тека `views` містить сторінки інтерфейсу, тека `scripts` зберігає виконувані `PHP`-скрипти, файл `main.php` визначає параметри сторінок інтерфейсу, файл `router.php` налаштовує маршрутизацію, і файл `.htaccess` відповідає за конфігурацію цієї директорії.

Загалом при першому запуску панелі необхідно здійснити вхід в обліковий запис за допомогою логіну та паролю. Вхід, який включає в себе автентифікацію, тобто процес підтвердження справжності адміністратора, здійснюється шляхом надсилання його облікових даних із панелі на серверну логіку, де вони звіряються на відповідність у базі даних. В ній пароль зберігається у вигляді хеш-рядка, який використовує криптографічну хеш-функцію `SHA-384`. У разі успішної автентифікації особи, її буде

переспрямовано на відповідну сторінку, де надається доступ до панелі адміністратора.

Панель адміністратора містить меню, яке складається з розділів про заклад і факультети, а також підменю профілю адміністратора (рис. 3.4).



Рис. 3.4. Головне меню адміністративної панелі

У розділі «Заклад» адміністратор може задавати повну і коротку назву закладу, його ідентифікатор в АСУ «ВНЗ» тощо (рис. 3.5). Якщо такі відомості не будуть заповнені, розділ «Факультети» буде недоступним для адміністратора, а також неможливо буде виконати будь-які запити до прикладного програмного інтерфейсу. Щоб заповнити відомості, потрібно натиснути «Редагувати», після чого з'явиться модальне вікно (рис. 3.6), у якому необхідно внести дані та зберегти їх.

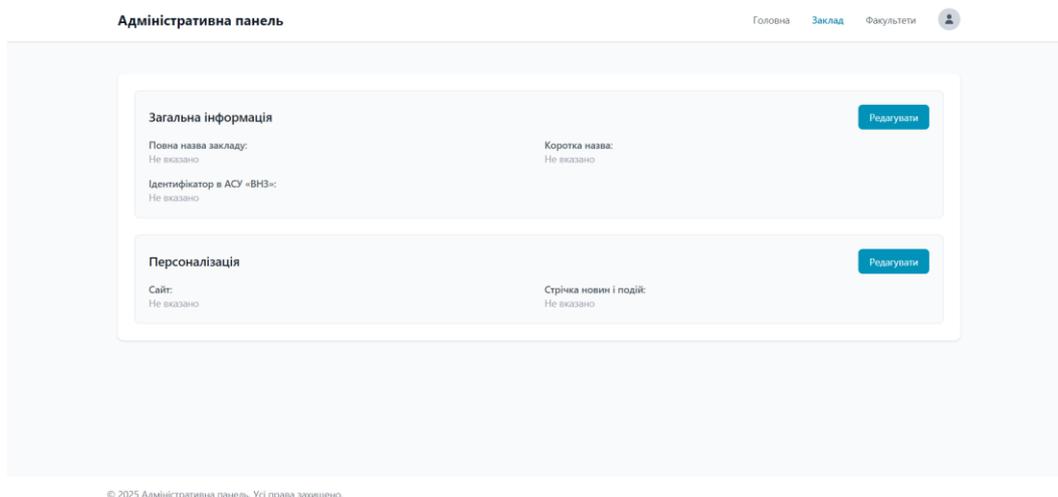


Рис. 3.5. Початкова сторінка закладу

Рис. 3.6. Заповнення загальної інформації закладу

Розділ «Факультети» містить список факультетів під'єднаних до інформаційної системи. Щоб під'єднати факультет, потрібно натиснути «Додати», після чого з'явиться список доступних факультетів для підключення, отриманих з АСУ (рис. 3.7). Вибравши факультети й натиснувши «Зберегти», їх буде відображено в списку під'єднаних до системи. Натискаючи на кожен під'єднаний факультет, адміністратор може переглядати і редагувати загальну факультетську інформацію (рис. 3.8-3.9), а також персоналізувати інформаційний контент для здобувачів вищої освіти, тобто додавати корисні посилання на вебджерела для відображення в мобільному застосунку.

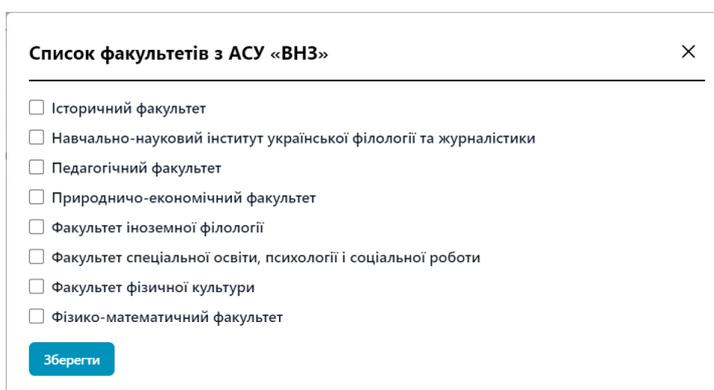


Рис. 3.7. Список доступних факультетів

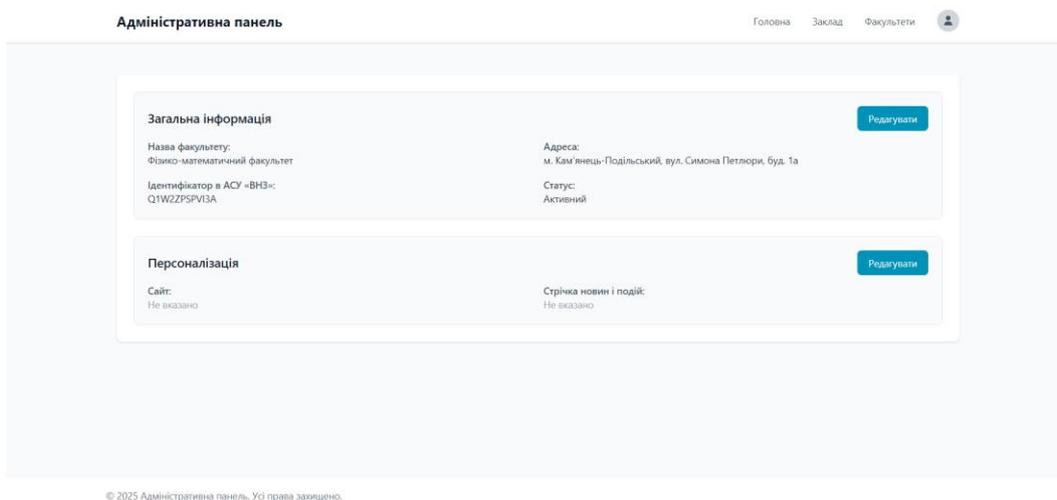


Рис. 3.8. Сторінка під'єданого факультету

У підменю профілю адміністратор може переходити у профіль для зміни облікових даних (імені, електронної пошти і пароля) та здійснювати вихід із адміністративної панелі.

Рис. 3.9. Редагування даних факультету

Тестування адміністративної панелі проводилося комплексно, поєднуючи мануальні та автоматизовані підходи. Для автоматизованої взаємодії з вебінтерфейсом панелі було обрано PlaywrightPHP – бібліотеку з відкритим вихідним кодом, яка дозволяє імітувати дії користувача в реальних браузерних середовищах. На основі цієї бібліотеки були реалізовані тест-кейси, написані мовою програмування PHP, які охоплювали ключові функціональні можливості адміністративної панелі та гарантували її коректну роботу в кросбраузерному режимі (Chromium, Firefox і WebKit).

Для локального розгортання вебмодуля інформаційної системи було використано OSPanel – безкоштовну портативну серверну платформу для Windows, яка забезпечує швидке створення повноцінного локального середовища з мінімальними зусиллями та має інтуїтивно зрозумілий інтерфейс, який дає змогу легко керувати всіма ключовими аспектами серверної інфраструктури. Під час встановлення цієї платформи на ОС Windows застосовувався вибірковий підхід, завдяки якому було інстальовано лише необхідні компоненти, такі як панель управління, вебсервер Apache, інтерпретатор PHP, СУБД MySQL і програмний інструмент phpMyAdmin.

3.2.2 Мобільний застосунок

Розробка мобільного застосунку інформаційної системи розпочалася із запуску Android Studio та створення нового проєкту. Для цього в головному вікні програми було натиснуто «New Project», після чого обрано шаблон «Empty Views Activity», який забезпечував просту структуру проєкту з

мінімальним функціоналом, і натиснуто «Next». Далі виконано налаштування проєкту: вказано назву та ім'я пакета, задано розташування файлів, обрано мову програмування Kotlin, встановлено мінімальний SDK на рівні API 26 для забезпечення сумісності з більшістю пристроїв, а також використано мову конфігурації збірки Kotlin DSL. Завершальним етапом стало натискання кнопки «Finish», після чого середовище розробки успішно завершило процес створення проєкту, автоматично згенерувавши його структуру.

Структура проєкту мобільного застосунку впорядкована та містить окремі каталоги для маніфестів, вихідного коду, ресурсів і Gradle-скриптів. У теці маніфестів розташовано конфігураційний XML-файл, який містить основну інформацію про застосунок, зокрема назву, цільову версію Android API, метадані, перелік різних компонентів. Тека вихідного коду включає спеціалізовані підкаталоги, призначені для розміщення кодів активностей, фрагментів, адаптерів і класів. Тека ресурсів охоплює всі некодові джерела, такі як локалізовані рядки, графічні елементи та дизайн-коди, розділені по відповідним каталогам. В теці скриптів Gradle розташовано ряд файлів, що використовуються для визначення конфігурації збірки проєкту.

Мобільний застосунок містить різні компоненти візуального інтерфейсу, серед яких основними є Activity (активність) – це повноцінний екран, з яким взаємодіє користувач, і Fragment (фрагмент) – це модульна частина активності, яку можна повторно використовувати. Одна активність може включати декілька фрагментів, кожен з яких має власний життєвий цикл, що напряду пов'язаний з життєвим циклом активності [25]. Найчастіше у процесі розробки даного застосунку замість звичайного вбудовування фрагмента в активність використовувався DialogFragment (діалоговий фрагмент), що належить до підкласу фрагментів і призначений для відображення у вигляді модального діалогового вікна, яке накладається поверх поточної активності [18, с. 283, 295].

Загалом реалізовано дві активності, перша з яких відповідає за логіку функціональної частини застосунку, а друга – за авторизацію користувача.

Робота мобільного застосунку розпочинається з головної активності, яка спершу ініціалізує splash-екран (вітальний екран), а потім перевіряє наявність авторизації. Для створення вітального екрана застосовувалася бібліотека `core-splashscreen` та стилі проєкту, написані завдяки дизайн-коду формату XML. Крім того, за допомогою таких кодів спроектовано головний інтерфейс застосунку, а також оформлення компонентів. При побудови дизайну використовувалась бібліотека `MDC-Android`.

Якщо під час запуску застосунку буде виявлено, що користувач не авторизований, головна активність переспрямує його на відповідну активність, де потрібно ввести логін і пароль від електронного кабінету здобувача вищої освіти та натиснути «Увійти» (рис. 3.10). У разі будь-якої помилки під час оброблення запиту на автентифікацію застосунок повідомить користувача, а після успішного виконання переспрямує його на головну активність, де можна переглянути насамперед розклад занять і знайти потрібну інформацію через панель навігації, натиснувши на кнопку меню (рис. 3.11). Всі необхідні дані зберігаються в локальній БД, а взаємодія з HTTP-протоколом відбувається за допомогою бібліотеки `OkHttp`. Вона забезпечує надсилання запитів до серверної частини інформаційної системи та вебсервера АСУ «ВНЗ», дозволяючи отримувати потрібну інформацію. Оскільки інформація повертається у форматі JSON, для її оброблення та аналізу використано вбудовані класи з пакета `org.json`.

Для зручного відображення розкладу реалізовано макети для елементів списку та заголовків, створено моделі даних для записів і групування за датами, а також розроблено відповідний адаптер під компонент `RecyclerView`, який використовується в дизайн-коді головної активності [21]. Даний адаптер найчастіше ініціалізується локальним розкладом, збереженим в базі даних. Подібний підхід було застосовано до відображення завдань, вичитки занять, підсумкової успішності тощо.

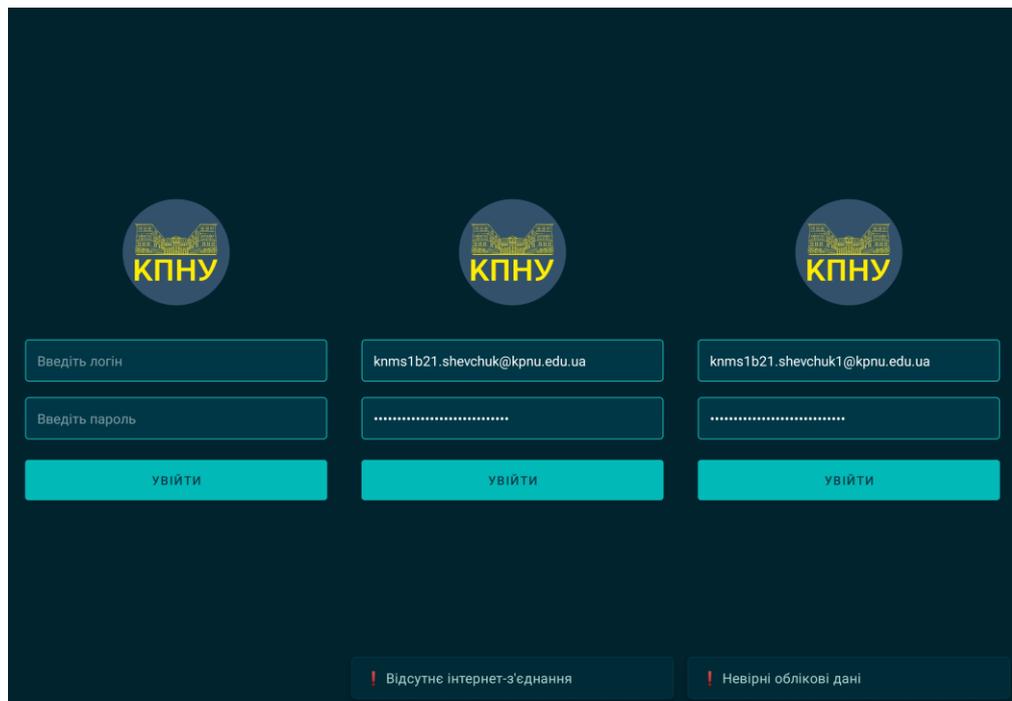


Рис. 3.10. Авторизація користувача з обробкою подій

Під час відкриття головної активності у фоновому режимі запускається перевірка актуальності розкладу. Якщо локальний розклад застарів, його буде оновлено, і з'явиться відповідне push-сповіщення. Варто зазначити, що під час першого запуску мобільного застосунку (після авторизації) розклад автоматично завантажується з вебсервера та зберігається в локальній БД. Натискаючи на кожне заняття розкладу, відображається спливаюча панель, яка містить розширену інформацію про заняття. Здобувач вищої освіти може переглядати розклад на поточний і наступний тиждень (за наявності), а натиснувши «Більше», отримує розширені можливості: перегляд свого розкладу без фільтрів на сьогодні, завтра чи вибрані дати, а також розклад конкретного викладача чи академічної групи. Доступне також фільтрування розкладу шляхом приховування підгруп та освітніх компонентів (рис. 3.12). Параметри приховування зберігаються до бази даних і застосовуються лише на головній активності. Щоб тимчасово скасувати їх, потрібно зняти відповідну позначку у меню «Більше». Для ручної перевірки актуальності локального розкладу здобувачу необхідно натиснути на відповідний значок на панелі інструментів, який запустить перевірку та сповістить його про хід виконання.

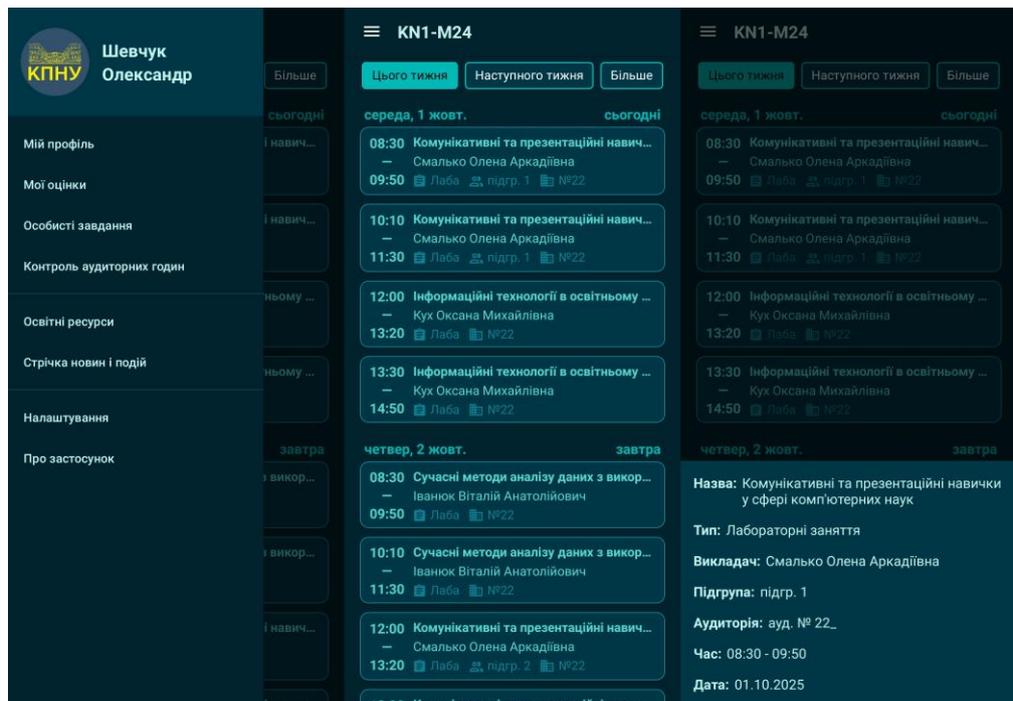


Рис. 3.11. Головний екран та панель навігації застосунку

Панель навігації застосунку, реалізована через компонент `NavigationBar`, містить меню з такими розділами: «Мій профіль», «Мої оцінки», «Особисті завдання», «Контроль аудиторних годин», «Освітні ресурси», «Стрічка новин і подій», «Налаштування» та «Про застосунок». Після натискання на певний його пункт, відкривається повноекранний діалоговий фрагмент цього розділу.

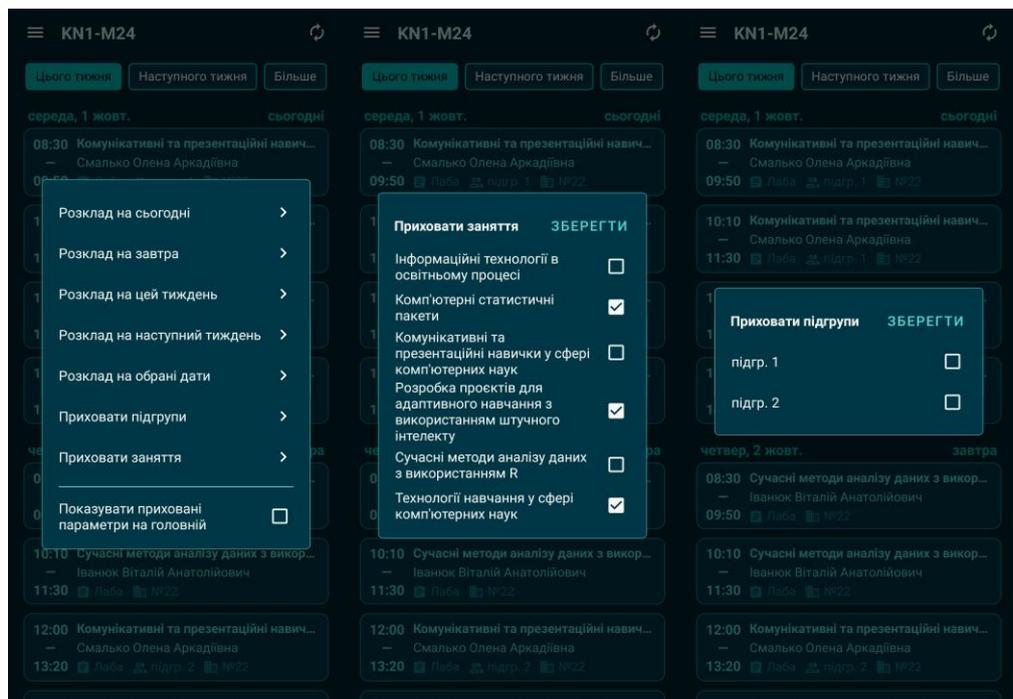


Рис. 3.12. Меню «Більше» із застосованими параметрами

У розділі «Мій профіль» здобувач вищої освіти може ознайомитись з наявними особовими даними, які містять прізвище, ім'я та по батькові (за наявності), дату народження, шифр групи, номер курсу, інформацію про факультет, спеціальність, джерело фінансування, освітній рівень здобуття, а також форму навчання (рис. 3.13).

Розділ «Мої оцінки» містить заліково-екзаменаційні відомості здобувача за вибраний семестр, включаючи повну назву освітнього компонента, форму підсумкового контролю, обсяг годин і кредитів ЄКТС, результати за 100-бальною шкалою, національною шкалою та шкалою ЄКТС, а також звання і скорочені ПІБ викладачів (рис. 3.13).

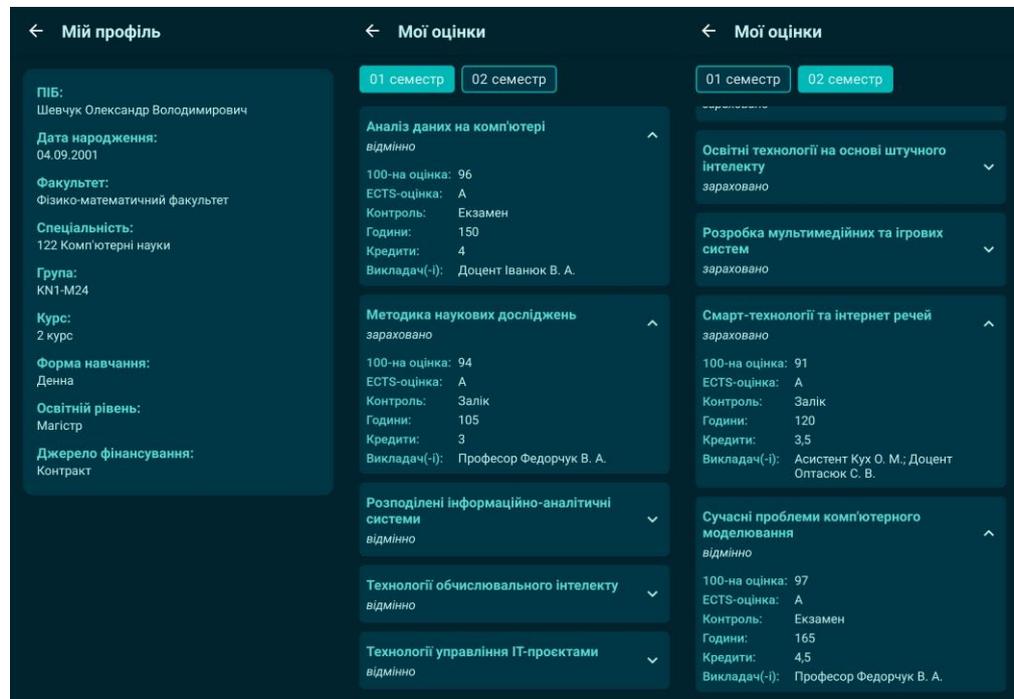


Рис. 3.13. Особиста інформація і підсумкова успішність

У розділі «Особисті завдання» відображаються завдання, які перебувають в роботі, а також доступний перегляд запланованих або виконаних завдань. Натиснувши на завдання, відкриється вікно для його перегляду та редагування. На сторінці завдань є можливість створити нове за допомогою відповідної кнопки (рис. 3.14). Загалом особисті завдання організовані за принципом простої Kanban-дошки, де всі завдання, які потрібно виконати, розміщені в розділі «Планується», завдання в процесі виконання – у розділі «В роботі», а завершені – у розділі «Виконано». До того

ж завдання можна розподіляти за пріоритетністю та категоріями, архівувати виконані. Завдання з низьким пріоритетом організовані без сповіщення, з середнім пріоритетом – зі сповіщенням про початок виконання, а з високим пріоритетом, окрім сповіщень, завдання закріплюються у верхній частині відповідного розділу.

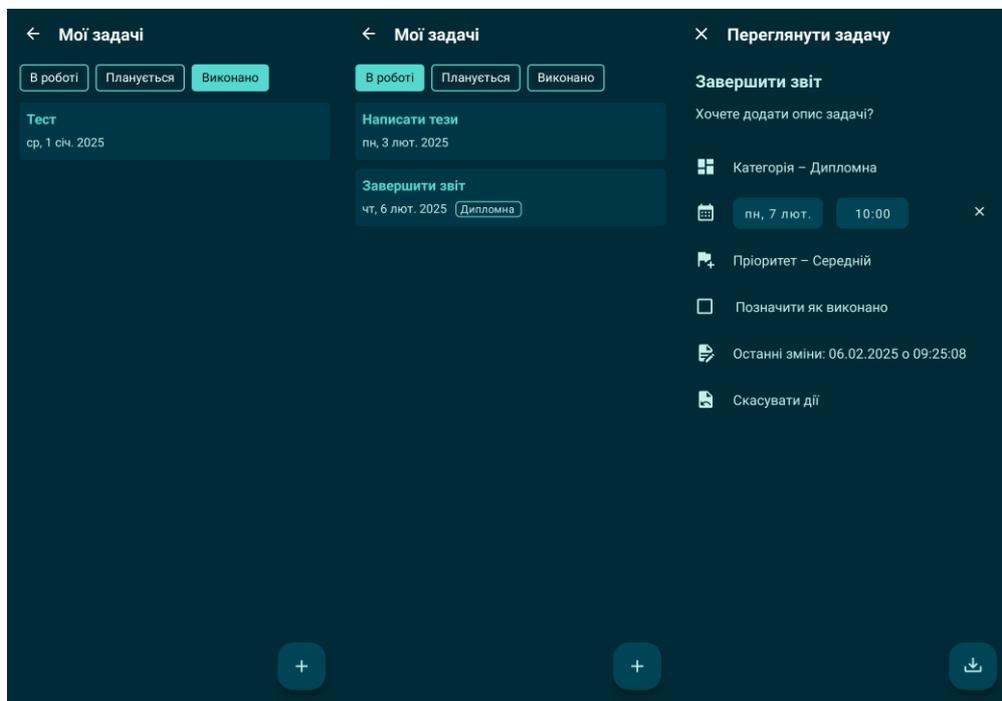


Рис. 3.14. Список завдань та їх перегляд

Розділ «Контроль аудиторних годин» дозволяє здобувачам вищої освіти контролювати вичитку занять, проведених в аудиторіях протягом поточного семестру (рис. 3.15). Він містить інформацію про кожне заняття, включаючи загальну кількість проведених занять та астрономічних годин, розподілених між різними викладачами, підгрупами та видами занять.

В розділі «Освітні ресурси» представлено вебджерела, організовані в основний і загальний інформаційні блоки, наповнення яких здійснюється факультетами через персоналізацію в адміністративній панелі. Вони можуть задати пріоритетні ресурси (наприклад, графік освітнього процесу, розклад екзаменів, ООП) для розміщення в основному блоці, а в загальному – перелік іншої корисної інформації (наприклад, сайти кафедр, сторінки в соцмережах). Розміщувана інформація в цьому розділі доступна у форматі вебсторінок (рис. 3.15). Перегляд таких сторінок здійснюється завдяки бібліотеки

androidx.browser, яка підтримує вбудовування користувацьких вкладок у мобільний застосунок, відкриваючи їх за допомогою стандартного браузера.

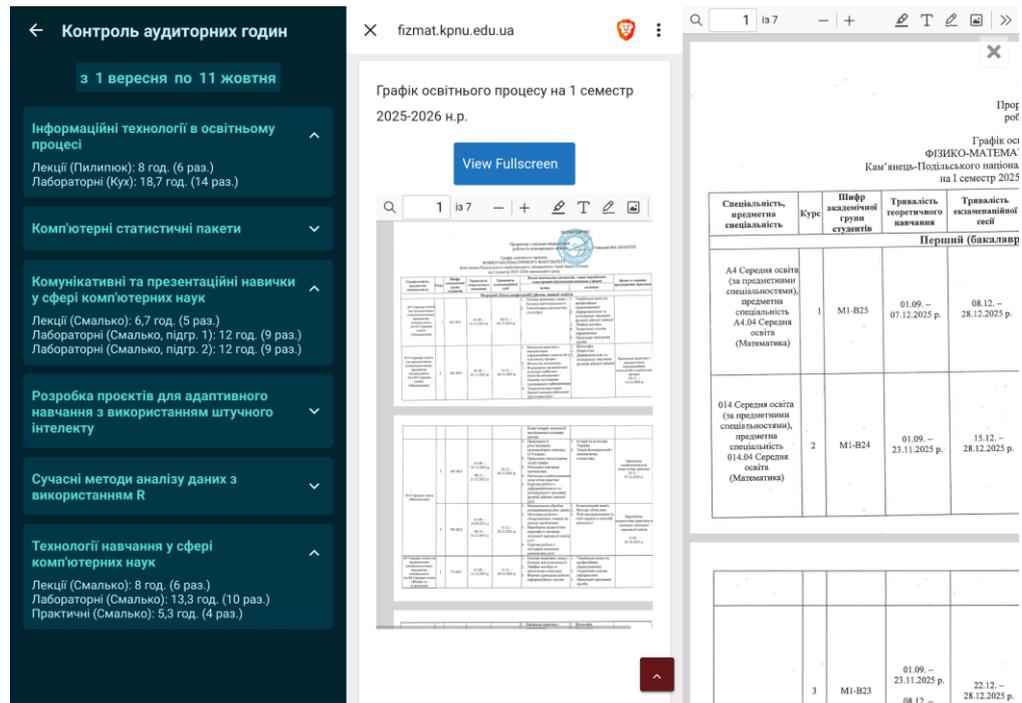


Рис. 3.15. Вичитка занять та перегляд вибраного вебджерела

Розділ «Стрічка новин і подій» відображає список новин і подій університету та факультету, який відфільтрований за датою. Після натискання на новину з'являється інформація про неї через вбудовану у застосунок вебсторінку (рис. 3.16).

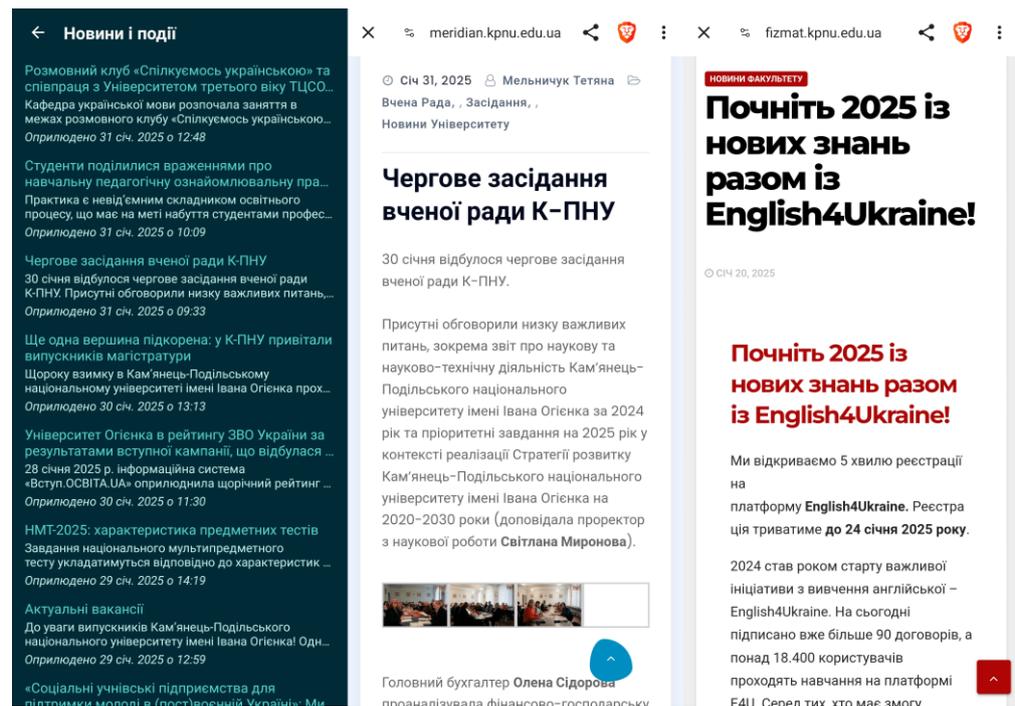


Рис. 3.16. Перегляд новин і подій

У розділі «Налаштування» можна ознайомитися з набором параметрів і внести необхідні зміни, а в розділі «Про застосунок» – переглянути інформацію про поточну версію застосунку, авторське право та рік розробки.

Тестування мобільного застосунку інформаційної системи відбувалося мануальним чином, тобто в ручну на фізичному пристрої Android. Для цього використовувався Android Debug Bridge (налагоджувальний міст Android) – це універсальний інструмент командного рядка, включений до пакета Android SDK Platform-Tools (доступний в Android Studio через SDK Manager), який дозволяє взаємодіяти з пристроєм. За його допомогою можна отримувати та змінювати системні налаштування пристрою, робити скріншоти й записувати відео екрана, імітувати торкання та натискання кнопок, працювати з файловою системою, видаляти, встановлювати і запускати застосунки з певними параметрами, а також підключатися до пристроїв через Wi-Fi чи USB.

Спершу Android-пристрій було під'єднано до бездротової мережі, потім у параметрах для розробників, увімкнувши «Бездротове налагодження» та надавши відповідний дозвіл, натиснуто «Прив'язати пристрій за допомогою коду прив'язування». Після цього в середовищі розробки Android Studio було відкрито термінал та введено такі дві команди:

- `adb pair <PHONE_IP>:<PHONE_PORT>`, команда, яка дозволяє прив'язати пристрій за вказаною IP-адресою і портом у діалоговому вікні прив'язування;
- `adb connect <PHONE_IP>:<PHONE_PORT>`, команда, яка дозволяє підключитися до пристрою за вказаною IP-адресою і портом на екрані налагодження після закриття діалогового вікна.

Після успішного виконання команд у середовищі розробки можна побачити назву мобільного пристрою в списку запущених пристроїв. Натиснувши кнопку «Run» на панелі інструментів Android Studio, було скомпільовано та запущено програму на фізичному пристрої. Таким чином, вдалося підключитися до Android-пристрою через бездротову мережу й запустити скомпільований мобільний застосунок інформаційної системи для

подальшого мануального тестування. Під час нього було виявлено декілька незначних візуальних недоліків у адаптивній верстці екранів різної роздільної здатності, які були оперативно виправлені шляхом оновлення відповідних дизайн-кодів мобільного застосунку.

3.3 Серверна частина

Як зазначалося раніше, серверна частина використовує прикладний програмний інтерфейс для надання мобільним клієнтам доступу до даних. Web API – це інтерфейс, який дозволяє клієнтам взаємодіяти з вебсервером через HTTP-запити. Він забезпечує доступ до функцій сервера, таких як робота з базою даних, передавання даних, доступ до ресурсів тощо. Взаємодія відбувається через визначені кінцеві точки (API-кінці) – конкретні URL-адреси, кожна з яких відповідає певним функціям. Дані зазвичай передаються у форматі JSON, що забезпечує зручність обміну між клієнтом і сервером. Таким чином, для реалізації прикладного програмного інтерфейсу було використано принципи RPC-архітектури (віддаленого виклику процедур). Реалізована модель API ґрунтується на використанні єдиного HTTP-методу POST для всіх типів запитів, від створення до виконання складних операцій. Кожна операція ідентифікується через унікальну кінцеву точку, що забезпечує зрозумілу структуру та логіку взаємодії. Усі параметри передаються в тілі запиту у форматі JSON, і серверна відповідь також надається в аналогічному форматі. Протокол HTTP забезпечує передавання даних, а результати операцій повідомляються через його зарезервовані коди стану.

Спершу розроблено логічну структуру організації файлів, розташованих в API-каталозі, у якому структура поділена на такі основні компоненти:

- .htaccess налаштовує правила перезапису URL для коректної маршрутизації запитів;
- index.php слугує точкою входу, яка приймає всі HTTP-запити та спрямовує їх до відповідних функцій;

- `service.php` включає функції, які взаємодіють з базою даних та зовнішніми джерелами, формуючи серверну відповідь.

Даний каталог розміщується на сервері в директорії вихідного коду (`src/api`), взаємодіючи з необхідними файлами адміністративної панелі. Особливо активно використовується директорія `inc`, яка слугує централізованим сховищем для важливих глобальних класів, зокрема відповідальних за організацію взаємодії з серверною базою даних та виконання різноманітних API-запитів до вебсерверів університету й АСУ «ВНЗ».

Кінцева точка формується з протоколу, доменного імені, шляху до каталогу API та назви конкретної функції, що реалізує потрібний функціонал. Кожен виклик такої точки супроводжується HTTP-заголовком, який вказує, що дані, передані у тілі такого запиту чи відповіді, мають формат JSON. Для доступу до особистих даних здобувача вищої освіти у тілі запиту необхідно зазначити його ідентифікатори – логін і пароль від відповідного електронного кабінету АСУ «ВНЗ». Такі ідентифікатори передаються виключно в зашифрованому вигляді єдиного кодованого рядка з використанням одного з найефективніших алгоритмів шифрування AES-128 за показниками швидкості, часу виконання, пропускну здатності та лавинного ефекту [31].

Нижче наведено реалізовані API-кінці, що дають змогу викликати відповідні функції прикладного програмного інтерфейсу. На рисунках 3.17-3.19 зображено успішне виконання запитів із отриманими відповідями.

POST api/authStud

Обробляється функцією `authenticateStudent(data)`, що приймає JSON-дані з ідентифікаторами здобувача. У разі успішної автентифікації повертаються основні дані з адміністративної панелі та АСУ «ВНЗ».

POST api/getStudProfile

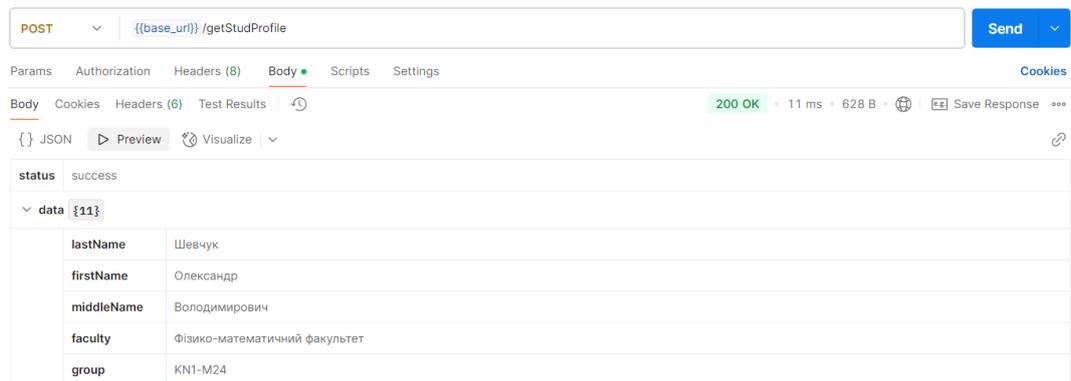
Обробляється функцією `getStudentProfile(data)`, що отримує JSON-дані з ідентифікаторами здобувача. Якщо запит оброблено успішно, повертається інформація про здобувача.

POST api/getStudGrades

Обробляється функцією `getStudentGrades(data)`, що отримує JSON-дані з ідентифікаторами здобувача. Якщо запит оброблено успішно, повертаються підсумкові оцінки здобувача.

POST api/getStudContent

Обробляється функцією `getStudentContent(data)`, що приймає JSON-дані з кодом факультету. У разі успіху повертається перелік інформаційного контенту (вебджерел) від університету та відповідного факультету.



```

POST {{base_url}} /getStudProfile
  
```

Params Authorization Headers (8) Body • Scripts Settings Cookies

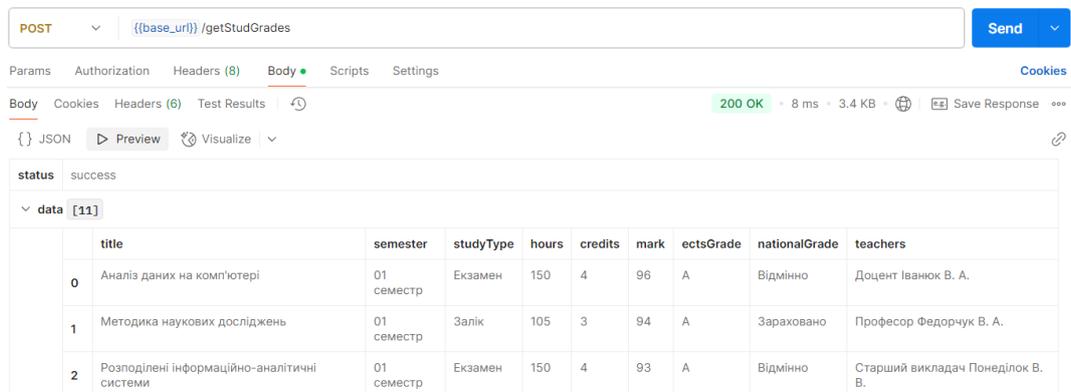
Body Cookies Headers (6) Test Results 200 OK • 11 ms • 628 B Save Response

{ } JSON Preview Visualize

```

{
  "status": "success",
  "data": {
    "lastName": "Шевчук",
    "firstName": "Олександр",
    "middleName": "Володимирович",
    "faculty": "Фізико-математичний факультет",
    "group": "KN1-M24"
  }
}
  
```

Рис. 3.17. Запит на отримання особистої інформації



```

POST {{base_url}} /getStudGrades
  
```

Params Authorization Headers (8) Body • Scripts Settings Cookies

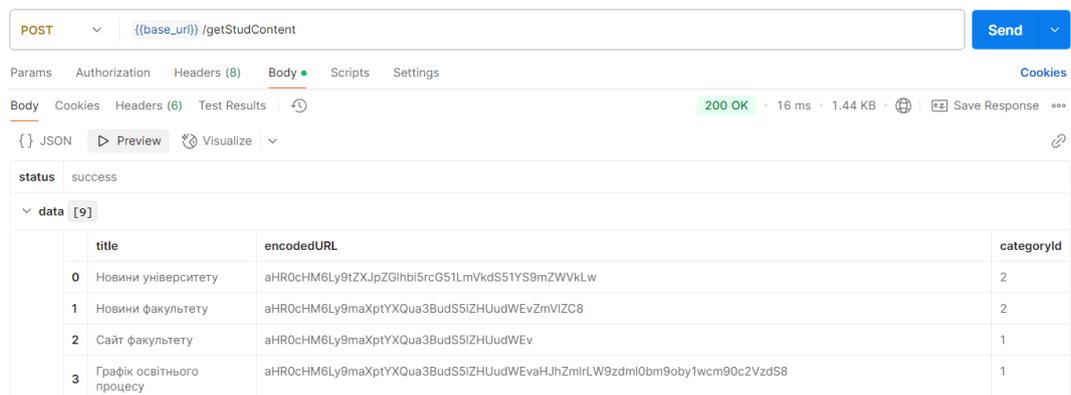
Body Cookies Headers (6) Test Results 200 OK • 8 ms • 3.4 KB Save Response

{ } JSON Preview Visualize

```

{
  "status": "success",
  "data": [
    {
      "title": "Аналіз даних на комп'ютері",
      "semester": "01 семестр",
      "studyType": "Екзамен",
      "hours": 150,
      "credits": 4,
      "mark": 96,
      "ectsGrade": "A",
      "nationalGrade": "Відмінно",
      "teachers": "Доцент Іванюк В. А."
    },
    {
      "title": "Методика наукових досліджень",
      "semester": "01 семестр",
      "studyType": "Залік",
      "hours": 105,
      "credits": 3,
      "mark": 94,
      "ectsGrade": "A",
      "nationalGrade": "Зараховано",
      "teachers": "Професор Федорчук В. А."
    },
    {
      "title": "Розподілені інформаційно-аналітичні системи",
      "semester": "01 семестр",
      "studyType": "Екзамен",
      "hours": 150,
      "credits": 4,
      "mark": 93,
      "ectsGrade": "A",
      "nationalGrade": "Відмінно",
      "teachers": "Старший викладач Понеділок В. В."
    }
  ]
}
  
```

Рис. 3.18. Запит на отримання підсумкових оцінок



```

POST {{base_url}} /getStudContent
  
```

Params Authorization Headers (8) Body • Scripts Settings Cookies

Body Cookies Headers (6) Test Results 200 OK • 16 ms • 1.44 KB Save Response

{ } JSON Preview Visualize

```

{
  "status": "success",
  "data": [
    {
      "title": "Новини університету",
      "encodedURL": "aHR0cHMGLy9tZkxjZGhbiS5rcG51LmVkdS51Y59mZWVklw",
      "categoryid": 2
    },
    {
      "title": "Новини факультету",
      "encodedURL": "aHR0cHMGLy9maXptYXQua3BudS5IZHUudWEVzZmVIZC8",
      "categoryid": 2
    },
    {
      "title": "Сайт факультету",
      "encodedURL": "aHR0cHMGLy9maXptYXQua3BudS5IZHUudWEV",
      "categoryid": 1
    },
    {
      "title": "Графік освітнього процесу",
      "encodedURL": "aHR0cHMGLy9maXptYXQua3BudS5IZHUudWEvaHJhZm90c2VzZD8",
      "categoryid": 1
    }
  ]
}
  
```

Рис. 3.19. Запит на отримання інформаційного контенту

Тестування прикладного програмного інтерфейсу проводилося автоматизованим способом за допомогою Postman – інструменту для розробки, тестування й документування API. Завдяки ньому було протестовано усі API-кінці: створено запити, впорядковано їх у колекції та виконано із застосуванням змінних середовища, пре-скриптів (сценаріїв попередньої обробки) і пост-тестів (тестових сценаріїв) для перевірки відповідей на відповідність очікуваним результатам. Такий підхід не лише прискорив процес тестування, дозволивши багаторазово запускати всю колекцію запитів одним натисканням на відповідну кнопку, а й забезпечив надійну перевірку функціональності API в різних сценаріях.

Висновки до розділу 3

У третьому розділі детально викладено ключові аспекти створення інформаційної системи. Спроектовано серверну та локальну бази даних на основі реляційного підходу в обраних системах керування БД – MySQL і SQLite. Для реалізації поставленої мети та завдань кваліфікаційної роботи використовувалися різноманітні мови програмування: JavaScript і PHP для створення вебмодуля, який включає адміністративну панель та серверну частину, та Kotlin для розробки мобільного застосунку. Побудовано оптимальну структуру для кожного компоненту інформаційної системи, і наведено перелік їхніх функціональних можливостей, які наочно проілюстровані на відповідних рисунках. Також описано процес тестування цих компонентів, де переважно застосовувався автоматизований підхід: PlaywrightPHP для тестування вебінтерфейсу панелі, і Postman для тестування прикладного програмного інтерфейсу. При цьому вебмодуль під час тестування було розгорнуто на локальному середовищі ОС Windows за допомогою платформи OSPanel, тоді як для мобільного застосунку встановлювалося з'єднання з фізичним Android-пристроєм через бездротову мережу за допомогою інструменту налагодження ADB.

РОЗДІЛ 4

РОЗГОРТАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Адміністративна панель і серверна частина

Вебмодуль інформаційної системи функціонує завдяки його розміщенню на хостингу провайдера Serv00 [30]. Серед його переваг над іншими безкоштовними хостингами було те, що він пропонував необмежений обсяг трафіку, відсутність реклами, 3 ГБ SSD-сховища, можливість розміщення до 100 сайтів і створення до 16 баз даних, надання SSL-сертифікатів Let's Encrypt, доступ через SSH із підтримкою тунелювання, щоденне резервне копіювання, розташування серверів в ЄС, а також використання сучасних технологій (PHP, Python, Ruby, Node.js тощо) та антивірусний захист.

Замість використання запропонованого субдомену провайдера було застосовано сервіс DigitalPlat FreeDomain [23], який дає змогу безкоштовно отримати унікальне доменне ім'я (на рік із можливістю поновлення) та прив'язати його до хостингу. Відповідно, у реєстраційному сервісі доменів було вказано DNS-сервери хостинг-провайдера (рис. 4.1), а в панелі керування хостингом додано домен, налаштовано SSL-сертифікат й рівень безпеки WAF (рис. 4.2). Крім того, через цю панель було створено та налагоджено функціонування бази даних MySQL. Запакувавши та розпакувавши zip-архів програмного коду адміністративної панелі та серверної частини через файл-менеджер хостингу, було задано початкові налаштування вебмодуля, вказавши конфігурацію створеної бази даних та інші параметри у файлі конфігурації (config.php).

Після переходу за доменним іменем та авторизації з початковими даними з конфігураційного файлу було змінено логін та пароль користувача, внесено повну і коротку назву закладу, його ідентифікатор АСУ «ВНЗ» у відповідному розділі, а також здійснено персоналізацію інформаційного контенту для доданих факультетів. Таким чином, адміністративну панель і серверну частину було встановлено та налаштовано на хостингу.

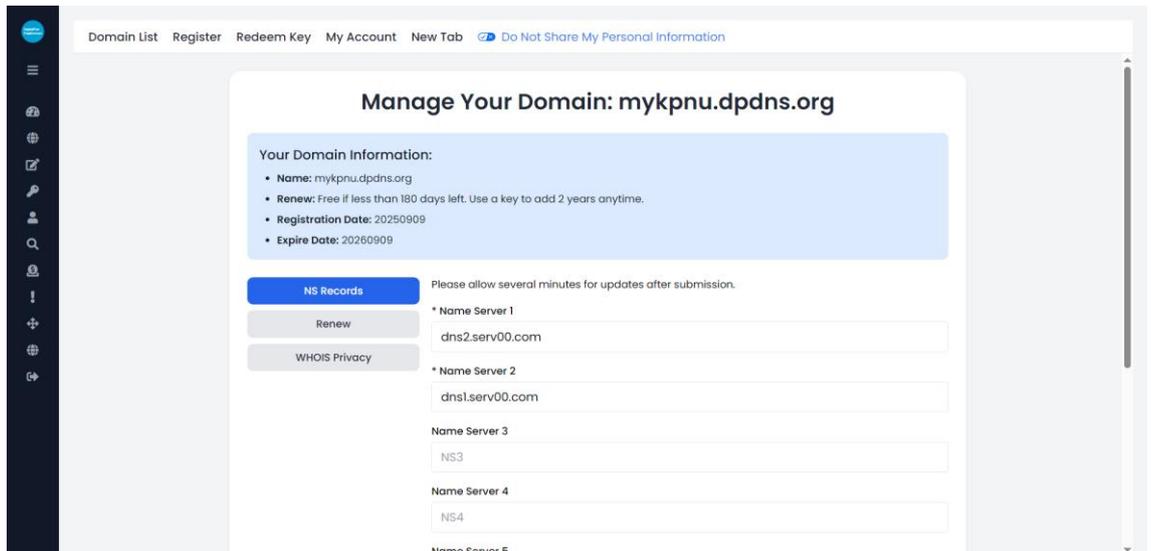


Рис. 4.1. Задані DNS-сервери домену в DigitalPlat FreeDomain

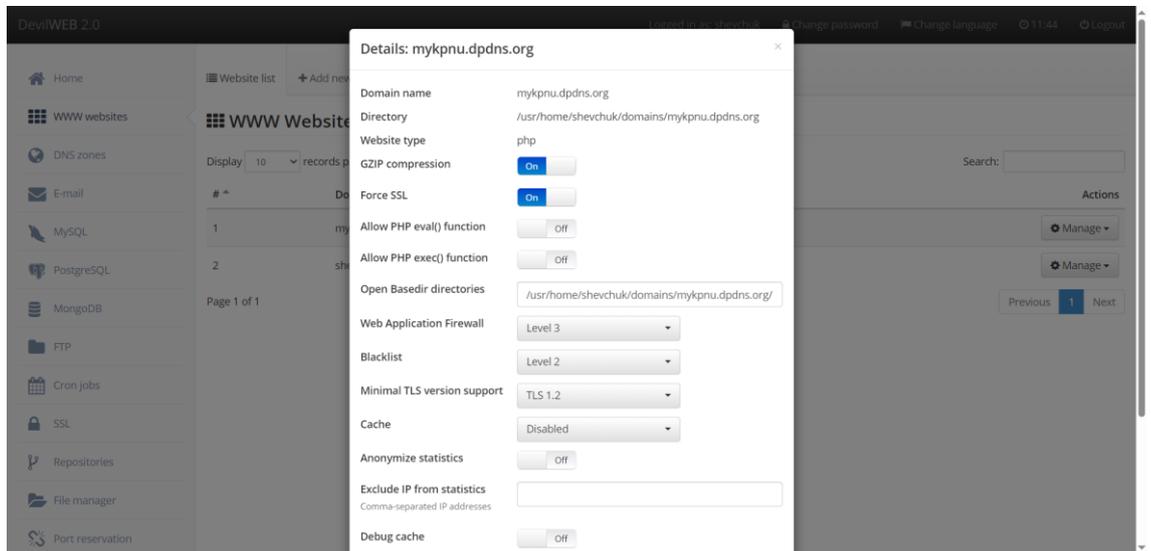


Рис. 4.2. Налаштування домену в Serv00

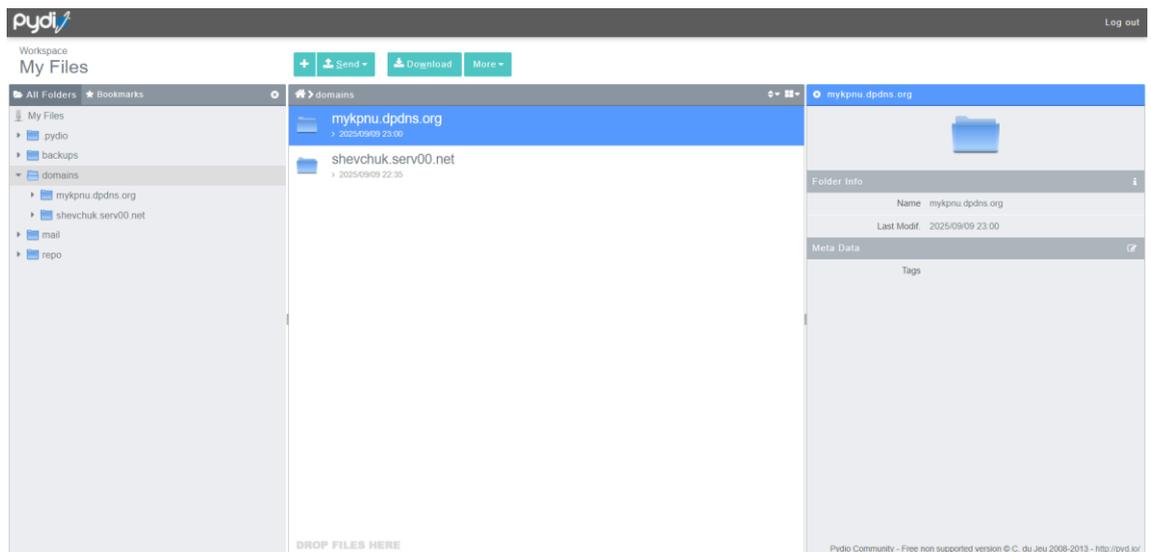


Рис. 4.3. Структура кореневої директорії домену

4.2 Мобільний застосунок

Мобільний застосунок інформаційної системи функціонує лише після безпосереднього встановлення на пристрій користувача у вигляді APK-файлу. Його поширення здійснюється через механізм керування релізами у GitHub-репозиторії, де всі доступні версії мобільного застосунку публікуються у вкладці «Releases» для завантаження. Генерація файлу APK виконується в середовищі розробки Android Studio через збирання проєкту та формування інсталяційного файлу за допомогою вбудованих інструментів.

Завантаживши актуальну версію мобільного застосунку, користувач відкриває відповідний файл і дозволяє встановлення програми. Після інсталяції першочергово відкривається splash-екран застосунку (рис. 4.4).

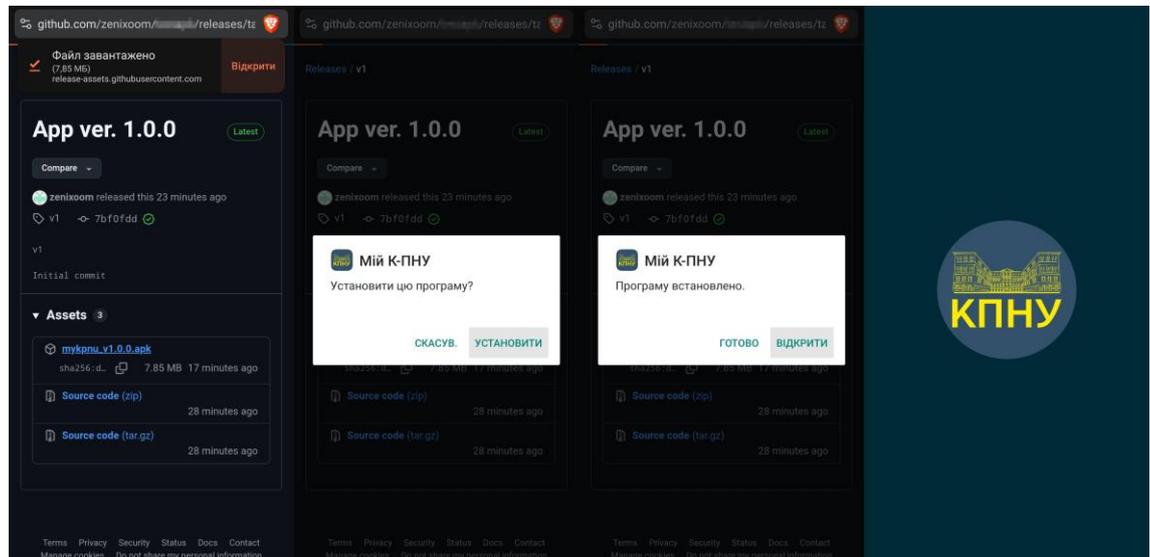


Рис. 4.4. Встановлення і запуск мобільного застосунку

Висновки до розділу 4

Прикінцевий розділ присвячений опису основних етапів розгортання інформаційної системи для загальнодоступного користування. Він містить інформацію про вибрані сервіси для функціонування вебмодуля системи (адміністративної панелі і серверної частини) та його встановлення на хостингу. Також вказується про доступність APK-файлу мобільного застосунку для опублікування у GitHub-репозиторії та зазначається про процес його встановлення на пристрій користувача.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи магістра було розроблено модель інформаційної системи здобувачів вищої освіти, спрямовану на відслідковування освітнього процесу. Розглянуто процес відстеження здобувачами власного навчального процесу та визначено їхні пріоритетні складові освітнього процесу, а також проаналізовано наявні мобільно-орієнтовані інформаційні системи, які забезпечують здобувачів вищої освіти інструментами для моніторингу навчання. Досліджено теоретичні аспекти проєктування інформаційної системи, що стали основою для розробки та програмної реалізації моделі цієї системи.

Запропонована модель інформаційної системи ґрунтується на інтеграції з компонентами АСУ «ВНЗ», які дозволяють отримувати через мережу розклад занять, особову інформацію та заліково-екзаменаційні відомості здобувачів вищої освіти, а також з вебсторінками закладу для доступу до інших складових освітнього процесу та корисної інформації. Таким чином, запропонована модель забезпечує мінімізацію ручного внесення даних до інформаційної системи та усуває процес накопичення дубльованої інформації з часом.

Розроблена модель інформаційної системи здобувачів вищої освіти поєднує в собі вебмодуль і мобільний застосунок. Вебмодуль складається з панелі адміністратора та серверної частини, що забезпечує управління системою на централізованому рівні та надає доступ до даних через створений прикладний програмний інтерфейс.

За допомогою адміністративної панелі налаштовується загальна інформація про університет та додані факультети, які попередньо вибірково синхронізовано через АСУ «ВНЗ». Також за її допомогою здійснюється персоналізація інформаційного контенту від кожного факультету, використовуючи задані конкретні посилання на вебджерела для показу здобувачам вищої освіти в мобільному застосунку. Такими джерелами можуть бути як RSS-канали (наприклад, налаштована на сайті стрічка новин і подій),

так і повноцінні електронні ресурси (наприклад, розташований на сайті PDF-файл графіка освітнього процесу).

За допомогою мобільного застосунку здійснюється безпосередній моніторинг освітнього процесу. Для отримання індивідуальної інформації здобувачі вищої освіти авторизуються в застосунку шляхом входу в обліковий запис відповідного електронного кабінету АСУ «ВНЗ» за допомогою логіна та пароля. Усі отримані дані з цього кабінету автоматизовано обробляються серверною частиною та повертаються у застосунок через взаємодію з прикладним програмним інтерфейсом системи, зважаючи на безпеку доступу до даних та актуальність API-запитів до зовнішніх джерел. В мобільному застосунку впроваджено можливості перегляду розкладу занять в офлайн-режимі з урахуванням вільного вибору освітніх компонентів і підгруп, а також відслідковування аудиторних годин для здійснення контролю за вчиткою, що є унікальною функцією серед проаналізованих мобільно-орієнтованих рішень. Крім того, реалізовано планування персональних завдань за дошкою Kanban, які можна розподіляти за пріоритетністю та категоріями, архівувати й відстежувати їхній прогрес.

Розроблення моделі інформаційної системи здобувачів вищої освіти здійснено з використанням таких технологій і середовищ: HTML, Tailwind CSS, JavaScript, PHP, MySQL і VSCode для вебмодуля, а також Kotlin, SQLite і Android Studio для мобільного застосунку. До того ж застосовувалися платформа OSPanel для локального розгортання, хостинг Serv00 для глобального розгортання, інструмент ADB для локального з'єднання з фізичним Android-пристроєм, репозиторій GitHub для поширення APK-файлу та інструменти PlaywrightPHP і Postman для автоматизованого тестування.

Загалом реалізована модель інформаційної системи відповідає всім вимогам, встановленим на етапі постановки задачі. Вона забезпечує здобувачів вищої освіти простим та ефективним цифровим інструментом для відслідковування освітньої діяльності й власного навчального процесу, що сприяє підтримці в освітньому середовищі та цифровізації навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Авраменко В. С., Авраменко А. С. Проектування інформаційних систем : навч. посіб. Черкаси : ЧНУ ім. Б. Хмельницького, 2017. 434 с.
2. АС «Деканат». URL: <https://vuz.osvita.net/as-dekanat> (дата звернення: 16.01.2025).
3. АСУ «ВНЗ». URL: <https://vuz.osvita.net> (дата звернення: 16.01.2025).
4. АСУ «МКР». URL: <https://mkr.org.ua> (дата звернення: 16.01.2025).
5. АСУ ВНЗ: Програмне забезпечення, призначення якого підвищити ефективність управління навчальним закладом та пришвидшити і спростити роботу співробітників ЗВО. URL: https://vuz.osvita.net/wp-content/uploads/2024/08/ASU_VNZ.pdf (дата звернення: 10.01.2025).
6. Веб-Розклад для АС «Деканат». URL: <https://vuz.osvita.net> (дата звернення: 10.01.2025).
7. Гавриляк М. С. Програмування мережних послуг : навч. посіб. Чернівці : ЧНУ ім. Ю. Федьковича, 2022. 179 с.
8. Діденко Д. Г., Черкас В. М. Методичний посібник: Деканат. Навчальні плани, навантаження та розклад. 3-тє вид. Київ : НДІ ПІТ, 2014. 160 с.
9. Діденко Д. Г., Черкас В. М. Методичний посібник: Деканат. Студенти та оцінки. 4-тє вид. Київ : НДІ ПІТ, 2014. 216 с.
10. Коваль Ю. В., Ставровський А. Б. Інформаційні мережі : навч. посіб. Київ : КНУ ім. Т. Шевченка, 2021. 84 с.
11. Мельник О. В. Геопросторові бази даних : конспект лекцій. Луцьк : Вежа-Друк, 2022. 156 с.
12. Методи представлення, збереження та аналізу даних інформаційних систем : колективна монографія / В. Ю. Щербань, С. М. Краснитський, Т. І. Астістова, В. М. Яхно. Київ : ТОВ «Фастбінд Україна», 2023. 473 с.
13. Онлайн додаток АСУ «ВНЗ» – «Веб Деканат»: Веб-ресурс призначений спростити процеси, що напряду пов'язані зі студентами, їх

навантаженням та успішністю. URL: https://vuz.osvita.net/wp-content/uploads/2021/06/ASU_VNZ_Web_Dekanat.pdf (дата звернення: 00.00.2025).

14. Організація баз даних : навч. посіб. / Я. І. Соколовський, М. В. Дендюк, І. М. Крошний, І. Б. Пірко, М. М. Паславський. Гданськ : GSW, 2023. 466 с.

15. Пакет програм «Деканат». URL: <http://www.politek-soft.kiev.ua/index.php?do=products&product=deanery> (дата звернення: 16.01.2025).

16. Пірог О. В. Безпека вебдодатків : навч. посіб. Житомир : Житомирська політехніка, 2025. 290 с.

17. Проектування інформаційних систем : конспект лекцій / уклад. І. А. Саченко. Київ : КНУБА, 2024. 88 с.

18. Розроблення мобільних застосунків : навч. посіб. / уклад. К. О. Радченко. Київ : КПІ ім. І. Сікорського, 2023. 546 с.

19. Серверні WEB-технології : навч. посіб. / уклад. О. С. Бунке. Київ : КПІ ім. І. Сікорського, 2023. 109 с.

20. Шевчук О. Актуальність розробки системи контролю освітнього процесу здобувачів вищої освіти. *Вісник Кам'янець-Подільського національного університету імені Івана Огієнка. Фізико-математичні науки.* Кам'янець-Подільський : К-ПНУ ім. І. Огієнка, 2024. № 17. С. 201-205.

21. Шевчук О. Розробка мобільного застосунку для відстеження розкладу занять здобувачами вищої освіти. *Збірник матеріалів наукової конференції за підсумками науково-дослідної роботи здобувачів вищої освіти фізико-математичного факультету Кам'янець-Подільського національного університету імені Івана Огієнка у 2024-2025 н.р., 9-10 квітня 2025 року.* Кам'янець-Подільський : К-ПНУ ім. І. Огієнка, 2025. С. 102-105.

22. ASU VNZ. Chrome-Stats. URL: <https://chrome-stats.com/d/com.zoria2010.ASDekanatApp?hl=en> (дата звернення: 00.00.2025).

23. DigitalPlat FreeDomain. URL: <https://domain.digitalplat.org> (дата звернення: 00.00.2025).

24. eUniversity. Chrome-Stats. URL: <https://chrome-stats.com/d/com.nupp.euniversity?hl=en> (дата звернення: 00.00.2025).
25. He J., Wu Z., Chen T. Formalization of Android Activity-Fragment Multitasking Mechanism and Static Analysis of Mobile Apps. *Formal Aspects of Computing*. 2025. Т. 37, № 2. DOI: [10.1145/3708562](https://doi.org/10.1145/3708562).
26. My Study Life – School Planner. Chrome-Stats. URL: <https://chrome-stats.com/d/com.virblue.mystudylife?hl=en> (дата звернення: 00.00.2025).
27. Nyabuto G. Client-server Architecture, a Review. *International Journal of Advanced Science and Computer Applications*. 2024. Т. 3, № 2. DOI: [10.47679/ijasca.v3i1.48](https://doi.org/10.47679/ijasca.v3i1.48).
28. Richards M., Ford N. Fundamentals of Software Architecture. Sebastopol : O'Reilly Media, 2020. 422 p.
29. Sahani A. Android v/s IOS – The Unceasing Battle. *International Journal of Computer Applications*. 2017. Т. 180, № 3. С. 23-26. DOI: [10.5120/ijca2017915990](https://doi.org/10.5120/ijca2017915990).
30. Serv00. URL: <https://www.serv00.com> (дата звернення: 00.00.2025).
31. Sood R., Kaur H. A Literature Review on RSA, DES and AES Encryption Algorithms. *Emerging Trends in Engineering and Management*. 2023. С. 57-63. DOI: [10.56155/978-81-955020-3-5-07](https://doi.org/10.56155/978-81-955020-3-5-07).
32. Unisched Chrome-Stats. URL: <https://chrome-stats.com/d/download.unisched.app?hl=en> (дата звернення: 00.00.2025).
33. uSched – розклад занять ЗВО. Chrome-Stats. URL: <https://chrome-stats.com/d/ua.in.usched.common?hl=en> (дата звернення: 00.00.2025).
34. Vandana, Nagwanshi K. K., Kumar A., Paliwal M., Fundamental Graphical User Interface Design of an Educational Android Application. *Advances in Signal Processing, Embedded Systems and IoT*. 2023. С. 519-529. DOI: [10.1007/978-981-19-8865-3_47](https://doi.org/10.1007/978-981-19-8865-3_47).