

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА
на тему: «**Розробка інтерактивної навчальної платформи**
"SkillHub" для вивчення ІТ-курсів»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

(підпис) Нікіта КАЗАНЦЕВ

Виконав: здобувач вищої освіти групи ПД-43

Нікіта КАЗАНЦЕВ

Керівник: _____
канд. фіз.-мат. наук Сергій КОМΠΑН

Рецензент: _____

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма « »

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2026 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Казанцеву Нікіті Євгенійовичу

1. Тема кваліфікаційної роботи: «Розробка інтерактивної навчальної платформи "SkillHub" для вивчення ІТ-курсів»

керівник кваліфікаційної роботи канд. фіз.-мат. наук Сергій КОМΠΑН,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «20» лютого 2026 року № 51.

2. Строк подання кваліфікаційної роботи «29» травня 2026 року.

3. Вихідні дані до кваліфікаційної роботи: теоретичні відомості про організацію онлайн-навчання з ІТ-напрямів, методи структурування електронних курсів, підходи до керування навчальними матеріалами, тестуванням та прогресом користувачів, опис принципів побудови клієнт-серверних web-застосунків, технічна документація щодо використання технологій Java, Spring Boot, PostgreSQL, React та TypeScript.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Огляд та аналіз предметної галузі онлайн-навчання з ІТ-напрямів.
2. Аналіз існуючих програмних засобів для організації онлайн-навчання, створення електронних курсів, проведення тестування та відстеження прогресу користувачів.
3. Огляд засобів реалізації web-застосунку SkillHub для організації онлайн-навчання з ІТ-напрямів.
4. Проектування структури web-застосунку, бази даних та основних модулів системи.
5. Програмна реалізація та опис функціонування web-застосунку SkillHub.
6. Тестування web-застосунку.
7. Апробація результатів дослідження.
8. Перелік графічного матеріалу: презентація.

5. Вимоги до web-застосунку SkillHub.

1. Логічна структура системи.
2. Діаграма прецедентів web-застосунку SkillHub.
3. Діаграма доменної моделі системи.
4. Діаграма класів або структура основних сутностей системи.
5. Діаграма компонентів web-застосунку.
6. Діаграма розгортання web-застосунку.
7. Взаємодія React, Spring Boot та PostgreSQL.
8. Екранні форми web-застосунку.
9. Схема та результати тестування програмного забезпечення.
10. Апробація результатів дослідження.

6. Дата видачі завдання «23» лютого 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури з теми онлайн-навчання, електронних курсів та web-застосунків для освіти	23.02–01.03.2026	
2	Аналіз та дослідження існуючих аналогів програмного забезпечення для організації онлайн-навчання з ІТ-напрямів	02.03–08.03.2026	
3	Аналіз предметної галузі онлайн-навчання з ІТ-напрямів та визначення вимог	09.03–16.03.2026	
4	Огляд засобів реалізації web-застосунку SkillHub для організації онлайн-навчання з ІТ-напрямів	17.03–23.03.2026	
5	Проектування web-застосунку SkillHub, логічної структури системи, бази даних та UML-діаграм	24.03–06.04.2026	
6	Програмна реалізація серверної та клієнтської частин	07.04–27.04.2026	
7	Тестування та оформлення результатів тестування	28.04–04.05.2026	
8	Оформлення роботи: вступ, висновки, реферат, список джерел та додатки	05.05–11.05.2026	
9	Розробка демонстраційних матеріалів	12.05–17.05.2026	
10	Попередній захист роботи	18.05–22.05.2026	

Здобувач вищої освіти

_____ (підпис)

Нікіта КАЗАНЦЕВ

Керівник
кваліфікаційної роботи

_____ (підпис)

Сергій КОМΠΑН

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 67 стор., 2 табл., 28 рис., 26 джерел.

Мета роботи – спрощення процесу організації онлайн-навчання з ІТ-напрямів за рахунок розробки веб-платформи, яка забезпечує централізований доступ до курсів, навчальних матеріалів, тестування та перегляду прогресу користувача.

Об'єкт дослідження – процеси організації онлайн-навчання з ІТ-напрямів.

Предмет дослідження – програмне забезпечення для організації онлайн-навчання з ІТ-напрямів.

Короткий зміст роботи: У роботі проаналізовано предметну галузь онлайн-навчання з ІТ-напрямів та визначено основні проблеми, пов'язані з розподіленням навчальних матеріалів між різними сервісами, складністю контролю прогресу студентів, недостатньою автоматизацією тестування, відсутністю єдиного середовища для проходження курсів та обмеженими можливостями аналізу результатів навчання. Проаналізовано існуючі програмні засоби для організації онлайн-навчання та роботи з електронними курсами: Moodle, Google Classroom, Coursera, Udemy та Stepik.

Розроблено структуру web-застосунку **SkillHub** та визначено його ключові функціональні можливості, зокрема: реєстрацію та авторизацію користувачів, розмежування ролей студента, автора курсу та адміністратора, перегляд каталогу ІТ-курсів, перегляд структури курсу, модулів і уроків, доступ до навчальних матеріалів, проходження тестування, автоматичне збереження результатів, відображення індивідуального прогресу студента, адміністрування навчального контенту та перегляд статистики навчання. У роботі використано Java та Spring Boot для реалізації серверної частини, PostgreSQL для зберігання даних, React і TypeScript для створення клієнтського інтерфейсу, а також REST API для організації взаємодії між клієнтською та серверною частинами системи.

Сферою використання застосунку є організація онлайн-навчання з ІТ-напрямів для студентів, початківців у сфері інформаційних технологій, авторів навчальних курсів та адміністраторів освітніх платформ, які потребують зручного інструменту для розміщення навчальних матеріалів, проходження тестування, контролю прогресу та централізованого керування навчальним процесом.

КЛЮЧОВІ СЛОВА: ОНЛАЙН-НАВЧАННЯ, ІТ-КУРСИ, SKILLHUB, WEB-ЗАСТОСУНОК, ЕЛЕКТРОННИЙ КУРС, ТЕСТУВАННЯ, ПРОГРЕС КОРИСТУВАЧА, JAVA, SPRING BOOT, POSTGRESQL, REACT, TYPESCRIPT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	11
ВСТУП.....	13
1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	15
1.1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПРЕДМЕТНОЇ ГАЛУЗІ.....	15
1.2 АКТУАЛЬНІСТЬ СТВОРЕННЯ ПЛАТФОРМИ ОНЛАЙН-НАВЧАННЯ	15
1.3 ЦІЛЬОВА АУДИТОРІЯ ТА ОСНОВНІ КОРИСТУВАЧІ СИСТЕМИ	16
1.4 ОСНОВНІ ФУНКЦІЇ ТА БІЗНЕС-ЛОГІКА СИСТЕМИ.....	17
1.5 СПОСОБИ ВИРІШЕННЯ ЗАДАЧІ ТА РОЛЬ ІНФОКОМУНІКАЦІЙНИХ ЗАСОБІВ	19
1.6 ОСНОВНІ ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ	21
1.7 АНАЛІЗ ПРОБЛЕМ І НЕДОЛІКІВ ІСНУЮЧИХ ПІДХОДІВ.....	21
2. АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОРГАНІЗАЦІЇ ОНЛАЙН- НАВЧАННЯ З ІТ-НАПРЯМІВ	23
2.1 MOODLE.....	23
2.2 GOOGLE CLASSROOM.....	25
2.3 COURSERA.....	26
2.4 UDEMY.....	28
2.5 ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ.....	29
3. ОГЛЯД ЗАСОБІВ РЕАЛІЗАЦІЇ ДОДАТКУ ДЛЯ ОНЛАЙН-НАВЧАННЯ З ІТ- НАПРЯМІВ.....	32
3.1 МОВА ПРОГРАМУВАННЯ JAVA	32
3.2 SPRING BOOT ТА SPRING FRAMEWORK	33
3.3 POSTGRESQL	34
3.4 REACT ТА TYPESCRIPT.....	35
4. ПРОЄКТУВАННЯ WEB-ЗАСТОСУНКУ SKILLHUB.....	36
4.1 ЛОГІЧНА СТРУКТУРА СИСТЕМИ	36
4.2 ДІАГРАМА ПРЕЦЕДЕНТІВ WEB-ЗАСТОСУНКУ SKILLHUB.....	38
4.3 ДІАГРАМА ДОМЕННОЇ МОДЕЛІ WEB-ЗАСТОСУНКУ SKILLHUB	40
4.4 ДІАГРАМИ ДІЯЛЬНОСТІ	42

4.4.1 ДІАГРАМА ДІЯЛЬНОСТІ ПРОЦЕСУ ПРОХОДЖЕННЯ КУРСУ СТУДЕНТОМ.....	42
4.4.2 ДІАГРАМА ДІЯЛЬНОСТІ ПРОЦЕСУ СТВОРЕННЯ НАВЧАЛЬНОГО КУРСУ	43
4.5 ДІАГРАМА КЛАСІВ WEB-ЗАСТОСУНКУ SKILLHUB.....	45
4.6 ДІАГРАМА КОМПОНЕНТІВ WEB-ЗАСТОСУНКУ SKILLHUB.....	47
4.7 ДІАГРАМА РОЗГОРТАННЯ WEB-ЗАСТОСУНКУ SKILLHUB.....	49
4.8 ПРОЄКТУВАННЯ БАЗИ ДАНИХ.....	51
5. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ WEB-ЗАСТОСУНКУ SKILLHUB.....	55
5.1 РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ	55
5.2 РЕАЛІЗАЦІЯ КЛІЄНТСЬКОЇ ЧАСТИНИ	58
5.3 ЕКРАННІ ФОРМИ WEB-ЗАСТОСУНКУ SKILLHUB.....	61
5.4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	67
5.5 РЕЗУЛЬТАТИ ТЕСТУВАННЯ	69
ВИСНОВКИ.....	73
ПЕРЕЛІК ПОСИЛАНЬ	76
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	79
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ.....	88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – інтерфейс прикладного програмування, що використовується для взаємодії між програмними компонентами системи.

REST – архітектурний стиль передавання даних між клієнтською та серверною частинами web-застосунку.

HTTP – протокол передавання даних у мережі Інтернет.

HTTPS – захищена версія протоколу HTTP, що забезпечує безпечний обмін даними.

JSON – текстовий формат обміну даними між клієнтською та серверною частинами системи.

UI – користувацький інтерфейс програмного забезпечення.

UX – користувацький досвід під час взаємодії з програмним продуктом.

HTML – мова розмітки гіпертексту, що використовується для створення структури web-сторінок.

CSS – мова стилів, що використовується для оформлення зовнішнього вигляду web-сторінок.

JavaScript – мова програмування, що використовується для створення інтерактивної поведінки web-сторінок.

TypeScript – мова програмування на основі JavaScript із підтримкою статичної типізації.

React – бібліотека JavaScript для створення компонентного користувацького інтерфейсу.

Java – об'єктно-орієнтована мова програмування, що використовується для реалізації серверної частини застосунку.

Spring Boot – фреймворк для швидкої розробки серверних web-застосунків мовою Java.

SQL – мова структурованих запитів для роботи з реляційними базами даних.

PostgreSQL – об'єктно-реляційна система керування базами даних.

DB – база даних, призначена для зберігання структурованої інформації.

ORM – технологія об'єктно-реляційного відображення, що забезпечує зв'язок між об'єктами програми та таблицями бази даних.

JPA – специфікація Java для роботи з реляційними базами даних через об'єктну модель.

CRUD – базові операції роботи з даними: створення, читання, оновлення та видалення.

JWT – формат токена, що може використовуватися для авторизації користувачів у web-застосунку.

SPA – односторінковий web-застосунок, у якому основна взаємодія з користувачем відбувається без повного перезавантаження сторінки.

LMS – система керування навчанням, призначена для організації електронного навчального процесу.

UML – уніфікована мова моделювання, що використовується для побудови діаграм програмної системи.

IDE – інтегроване середовище розробки програмного забезпечення.

Git – система контролю версій, що використовується для відстеження змін у програмному коді.

GitHub – веб-сервіс для зберігання репозиторіїв, спільної роботи з кодом та керування версіями проєкту.

Docker – платформа контейнеризації, що дозволяє запускати програмні компоненти в ізольованому середовищі.

ВСТУП

Актуальність теми. Онлайн-навчання з ІТ-напрямів є важливою складовою сучасної цифрової освіти, оскільки інформаційні технології швидко розвиваються, а користувачі потребують зручного доступу до навчальних матеріалів, практичних завдань, тестування та контролю власного прогресу. У сучасних умовах цифровізації освіти все більшого значення набувають web-застосунки, які дозволяють організувати навчальний процес у дистанційному форматі та забезпечити користувачу доступ до курсів незалежно від місця перебування [1; 13; 16].

Однією з поширених проблем онлайн-навчання є розподілення навчальних матеріалів між різними сервісами. Наприклад, теоретичні матеріали можуть зберігатися в одному ресурсі, тестові завдання — в іншому, а результати навчання — взагалі фіксуватися окремо. Через це користувачу складніше орієнтуватися в навчальному процесі, контролювати пройдені теми, бачити власний прогрес і розуміти, які матеріали потрібно вивчати далі. Також проблемою є недостатня автоматизація перевірки знань і обмежені можливості адміністрування навчального контенту [1; 13; 14].

Створення єдиної платформи онлайн-навчання може частково вирішити ці проблеми. Web-застосунок SkillHub дозволяє об'єднати в одному середовищі каталог ІТ-курсів, структуру курсу, модулі, уроки, тестові завдання, результати проходження тестів і відображення прогресу користувача. Такий підхід робить навчання більш послідовним, зрозумілим і керованим, а також спрощує взаємодію між студентом, автором курсу та адміністратором [13; 16; 17].

Web-застосунок SkillHub для онлайн-навчання з ІТ-напрямів може забезпечити зручний доступ до навчальних курсів, перегляд модулів і уроків, проходження тестування, автоматичне збереження результатів та контроль індивідуального прогресу. Адміністратор і автор курсу, у свою чергу, можуть керувати навчальним контентом, додавати нові курси, редагувати уроки,

формувати тестові завдання та переглядати статистику навчання. Враховуючи наведене, розробка web-застосунку SkillHub для організації онлайн-навчання з ІТ-напрямів є актуальною задачею [13; 16].

Мета роботи – спрощення процесу організації онлайн-навчання з ІТ-напрямів за рахунок розробки веб-платформи, яка забезпечує централізований доступ до курсів, навчальних матеріалів, тестування та перегляду прогресу користувача.

Об'єкт дослідження – процеси організації онлайн-навчання з ІТ-напрямів.

Предмет дослідження – програмне забезпечення для організації онлайн-навчання з ІТ-напрямів.

Методи дослідження. У бакалаврській роботі використовуються такі методи: аналіз предметної галузі, порівняльний аналіз існуючих програмних засобів для онлайн-навчання, об'єктно-орієнтоване проєктування, UML-моделювання, проєктування web-застосунку, моделювання структури бази даних, розробка клієнт-серверного програмного забезпечення та тестування [18; 21; 22].

Наукова новизна одержаних результатів. У ході виконання роботи одержано нове вирішення задачі організації онлайн-навчання з ІТ-напрямів — структуру web-застосунку SkillHub, який поєднує каталог курсів, навчальні модулі, уроки, тестування, ролі користувачів, збереження результатів і відстеження індивідуального прогресу.

Практична значущість одержаних результатів. Розроблений web-застосунок може бути використаний для організації дистанційного навчання з ІТ-напрямів, самостійного проходження курсів, перевірки знань студентів і керування навчальним контентом.

1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Загальна характеристика предметної галузі

В сучасних умовах, що супроводжуються цифровізацією освіти, особливої актуальності набувають програмні засоби, спрямовані на організацію дистанційного або змішаного типу навчання. Одним із найбільш вагомих напрямків є створення платформ для проведення онлайн-навчання [13; 16; 17].

Предметною галуззю кваліфікаційної роботи є організація онлайн-навчання за допомогою веб-платформи. Відображення навчальних матеріалів, можливість їх вивчення та тестування студентів є єдиною метою створення платформи [1; 13].

Загальна задача розробки подібного програмного забезпечення - створення веб-застосунку, що забезпечить можливість вивчення навчальних дисциплін, тестування та відстеження досягнень кожного зі студентів. Відмінною від традиційного навчання є можливість здобуття доступу до навчальних матеріалів у будь-який час та з будь-якого пристрою [1; 2].

Особливістю предметної галузі є можливість відображення навчальних матеріалів, тестування студентів та відстеження їх досягнень. Оскільки досягнення студентів можна відстежувати, то вирішується одна з основних задач [1; 2].

1.2 Актуальність створення платформи онлайн-навчання

Необхідність створення платформи онлайн-навчання в ІТ-напрямах сприймається як одна із проблем, що виникають при використанні розрізнених цифрових ресурсів. Програми, що вивчаються, тести, можливості взаємодії між викладачем та студентами розміщені в окремих платформах, що викликає навантаження для студентів [1; 13; 14].

Основними проблемами предметної галузі є наступні:

- відсутність єдиного середовища для вивчення ІТ-курсів;

- складність використання ресурсів для структурування навчальних матеріалів;
- недостатня автоматизація тестування та відстеження результатів;
- недостаток можливості відстеження досягнень кожного студента;
- необхідність швидкого отримання статистики відносно досягнень студентів;
- незручність використання декількох платформ одночасно.

Доцільністю створення веб-платформи є можливість вивчення курсів (вивчення матеріалу, тестування), створення аккаунтів для студентів та адміністраторів платформи, адміністрування навчальних матеріалів, оцінки досягнень студентів. Розробка платформи онлайн-навчання сприймається як рішення таких проблем, спрощення навчання для студентів та викладачів [13; 16].

1.3 Цільова аудиторія та основні користувачі системи

В основі створення платформи лежить потреба в організації онлайн-навчання для освоєння ІТ-дисциплін. Аудиторією платформи стануть початківці в галузі ІТ, студенти, а також ті, хто здобуває просто нові знання [2; 13].

Основними користувачами платформи стануть:

- Студенти - вивчають навчальний матеріал платформи, проходять тести та відстежують свої досягнення;
- Адміністратори платформи - вивчають курси та матеріали, що створені відомостями, відстежують досягнення студентів;
- Автори - створюють матеріали, що вивчаються студентами.

Студентам важливий простий інтерфейс платформи, адміністраторам - можливість вивчення відомостей, авторам на платформі - можливість створення та відстеження результатів. Програмний продукт має бути створений для відображення діяльності авторів, адміністраторів та студентів.

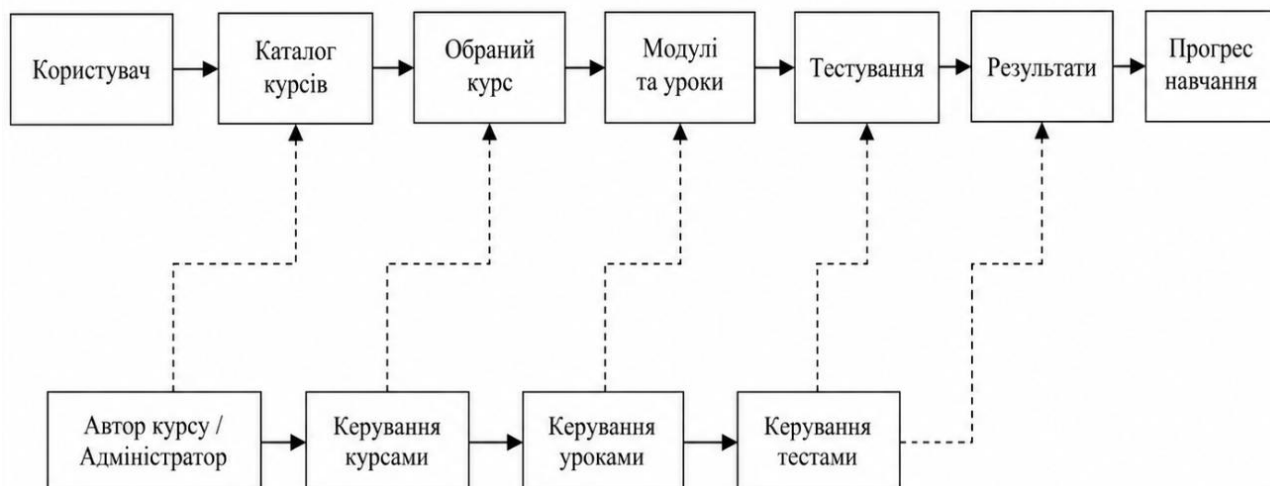


Рис. 1.1 Узагальнена схема організації онлайн-навчання на платформі SkillHub

1.4 Основні функції та бізнес-логіка системи

Розроблювана платформа повинна підтримувати базові процеси, характерні для систем онлайн-навчання. У загальному вигляді логіка роботи системи полягає в тому, що користувач проходить авторизацію, отримує доступ до каталогу курсів, обирає потрібний курс, вивчає матеріали та виконує контрольні завдання. Результати навчання зберігаються в системі й можуть бути використані для формування статистики прогресу [1; 2; 13].

До основних функцій системи доцільно віднести:

1. реєстрацію та авторизацію користувачів;
2. перегляд каталогу ІТ-курсів;
3. відображення структури курсу у вигляді модулів і уроків;
4. доступ до навчальних матеріалів;
5. проходження тестових завдань;
6. автоматичне збереження результатів тестування;
7. перегляд індивідуального прогресу;
8. адміністрування курсів, уроків, тестів і користувачів.

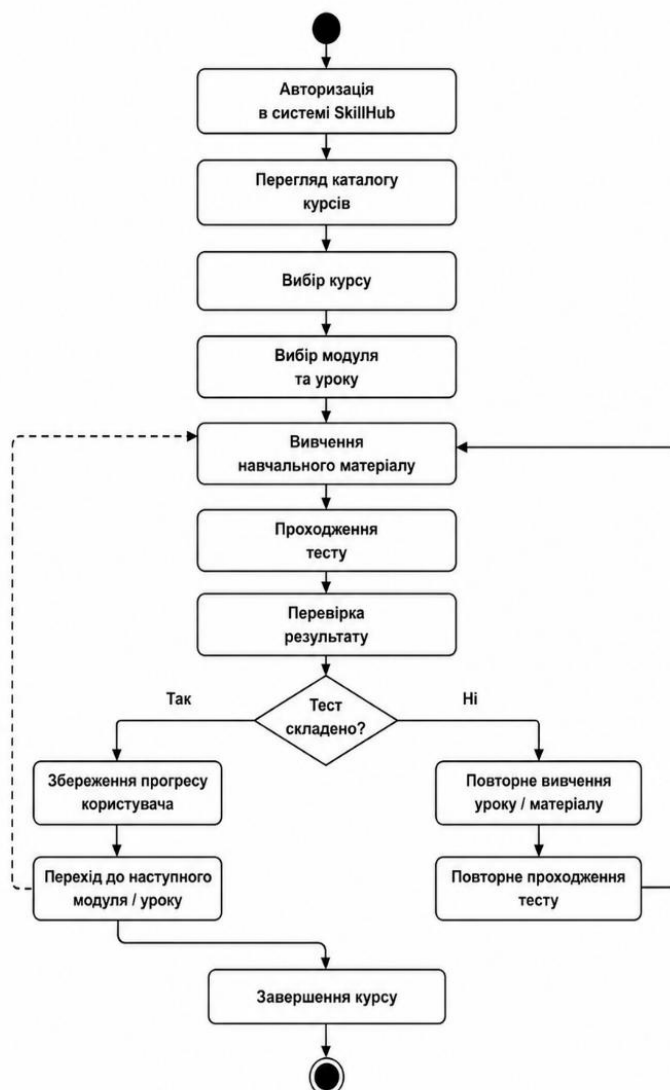


Рис. 1.2 Діаграма діяльності процесу проходження курсу та тестування на платформі

У межах предметної галузі важливою є правильна організація взаємозв'язків між сутностями системи. Основними інформаційними об'єктами платформи є курс, модуль, урок, тест, тестове питання, користувач, спроба тестування та результат. Саме на основі цих сутностей формується загальна модель системи [18; 21].

З позиції користувача типовий сценарій використання платформи виглядає так: студент обирає курс, переходить до певного модуля, переглядає урок, після чого проходить тестування для перевірки засвоєння матеріалу. Після завершення тесту система обробляє відповіді, обчислює результат і відображає його у профілі

користувача. Така послідовність дій формує базову бізнес-логіку навчального процесу [1; 2].

1.5 Способи вирішення задачі та роль інфокомунікаційних засобів

Задача організації онлайн-навчання може бути вирішена різними способами. Найпростішим підходом є використання декількох окремих сервісів: один ресурс застосовується для розміщення навчальних матеріалів, інший — для тестування, ще один — для обміну повідомленнями або зберігання статистики. Однак такий підхід не забезпечує цілісності освітнього процесу та ускладнює взаємодію між складовими системи [1; 13; 16].

Більш ефективним є створення єдиної веб-платформи, побудованої за клієнт-серверною моделлю. У такому випадку серверна частина відповідає за бізнес-логіку, зберігання та обробку даних, а клієнтська — за відображення інформації та взаємодію з користувачем. Подібний підхід забезпечує централізоване керування навчальними ресурсами, надійне збереження інформації та зручність використання [21; 22].

Інфокомунікаційні засоби в цій предметній галузі виконують ключову роль, оскільки саме вони забезпечують:

- доступ користувачів до системи через мережу Інтернет;
- передачу даних між клієнтською та серверною частинами;
- зберігання навчальних матеріалів і результатів тестування;
- інтерактивну взаємодію між користувачем і платформою;
- можливість масштабування та розвитку програмного продукту.

Отже, для ефективного вирішення задач предметної галузі найбільш доцільним є створення повноцінного веб-застосунку, в якому всі необхідні функції будуть інтегровані в єдину систему.

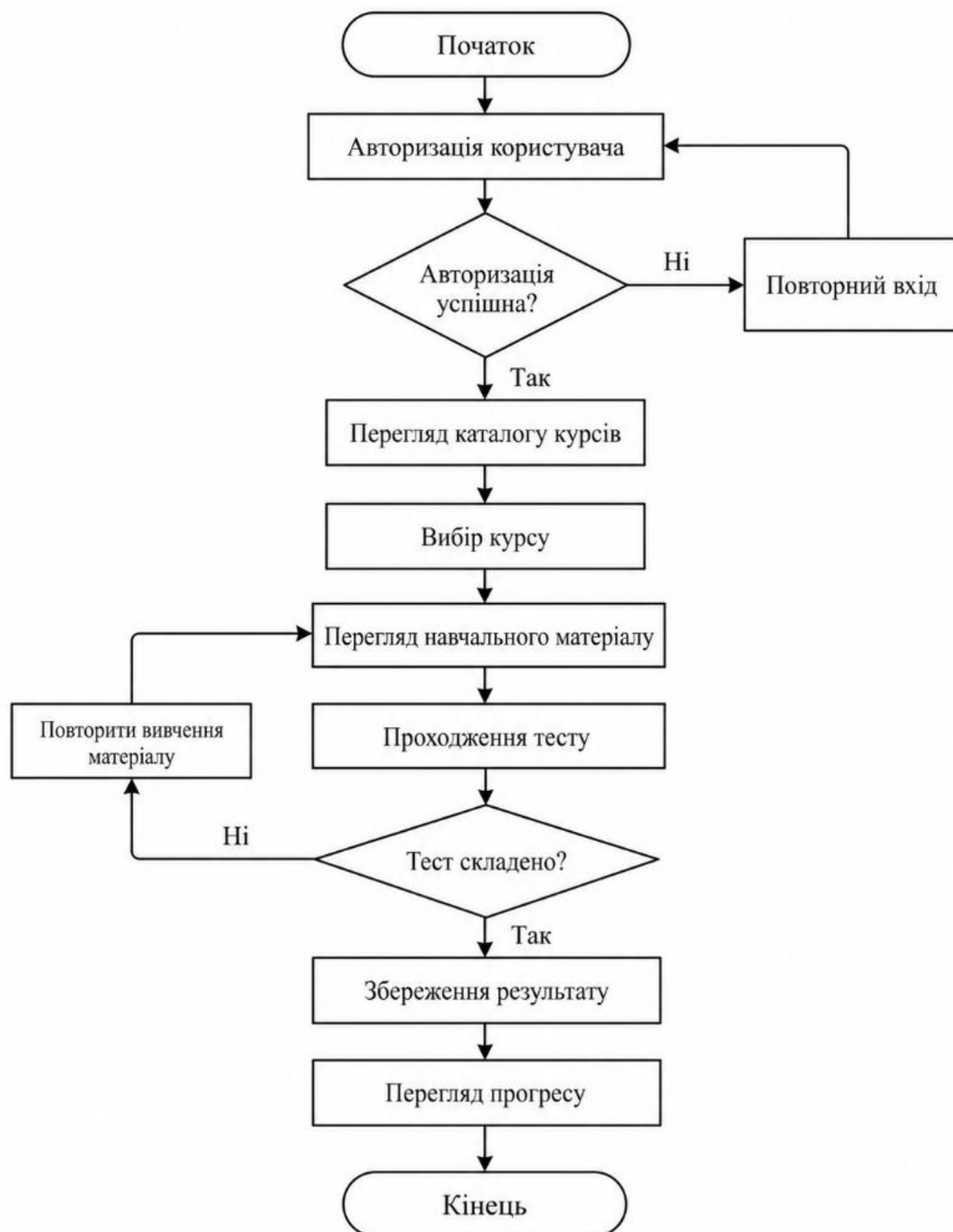


Рис. 1.3 Блок-схема процесу роботи користувача з платформою онлайн-навчання

1.6 Основні вимоги до програмного продукту

Для того щоб платформа відповідала потребам цільової аудиторії, вона повинна задовольняти як функціональні, так і нефункціональні вимоги [18; 21].

До **функціональних вимог** належать:

1. можливість реєстрації та входу користувача до системи;
2. перегляд каталогу курсів;
3. доступ до структури курсу, модулів і уроків;
4. проходження тестів і перегляд результатів;
5. збереження даних про прогрес користувача;
6. адміністрування навчального контенту;
7. керування обліковими записами користувачів.

До **нефункціональних вимог** належать:

1. зручність і зрозумілість інтерфейсу;
2. швидкодія системи;
3. надійність зберігання даних;
4. масштабованість;
5. безпечний доступ до інформації;
6. коректна робота на різних пристроях;
7. простота супроводу та оновлення системи.

Крім цього, важливими є обмеження, що можуть впливати на розробку системи. До них можна віднести необхідність доступу до мережі Інтернет, потребу в авторизації користувачів, а також необхідність підтримки актуальності навчального контенту [13; 16].

1.7 Аналіз проблем і недоліків існуючих підходів

Аналіз предметної галузі показує, що існуючі підходи до організації онлайн-навчання не завжди задовольняють потреби користувачів у повному обсязі. Частина систем має перевантажений інтерфейс, що ускладнює роботу нових

користувачів. Інші сервіси є надто універсальними, через що містять велику кількість зайвого функціоналу, який не є необхідним для вивчення ІТ-курсів [1; 13; 14].

Також досить поширеними недоліками є:

- недостатня гнучкість налаштування структури курсів;
- складність адміністрування контенту;
- незручне відображення результатів навчання;
- обмежені можливості аналітики прогресу;
- розподілення функцій між декількома окремими платформами.

Саме ці недоліки обґрунтовують доцільність розробки спеціалізованої веб-платформи онлайн-навчання, яка буде орієнтована на потреби користувачів у межах ІТ-напряму та забезпечуватиме більш цілісне, зручне і структуроване середовище навчання [1; 13; 16].

2. АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОРГАНІЗАЦІЇ ОНЛАЙН-НАВЧАННЯ З ІТ-НАПРЯМІВ

Програмні засоби для організації онлайн-навчання є важливою складовою сучасного освітнього процесу. Вони дозволяють розміщувати навчальні матеріали, формувати структуру курсів, організовувати тестування, контролювати результати студентів та забезпечувати взаємодію між учасниками навчання [1; 2; 13].

Для вирішення задач онлайн-навчання можуть застосовуватися різні типи програмних засобів:

- системи керування навчанням;
- освітні онлайн-платформи з готовими курсами;
- сервіси для створення навчальних класів і завдань;
- платформи для самостійного проходження курсів;
- спеціалізовані веб-застосунки для тестування та контролю прогресу [3; 4; 5; 6].

У межах даного розділу доцільно розглянути декілька популярних програмних засобів, які використовуються для організації онлайн-навчання та можуть бути аналогами майбутньої платформи. Для аналізу обрано такі системи: Moodle, Google Classroom, Coursera, Udemу [3; 4; 5; 6].

2.1 Moodle

Moodle є однією з найпоширеніх систем керування навчанням. Система використовується навчальними закладами, викладачами та самими організаціями. Moodle - це безкоштовна онлайн-система для відновлення навчального процесу, яка дозволяє створювати власний освітній простір з динамічними курсами [3].

Особливою перевагою платформи вважається велика кількість функцій, що спрощують організацію навчального процесу. Від відкриття і створення курсів до

додавання лекційних матеріалів, виконання завдань, тестів, розділів для дискусій та відстеження прогресу студентів - все це вміщує платформа Moodle [3].

Оскільки система розроблена для освітніх закладів з напрямком інформатики, то перехід до аналогу Moodle дозволяє відтворити основну структуру онлайн-освітньої платформи. Основні елементи платформи включають створення курсів, додавання матеріалів, створення тестів, виконання завдань, відстеження прогресу [3].

Переваги:

- велика кількість функцій для організації навчального процесу;
- можливість створення курсів будь-якої складності;
- можливість створення тестів, виконання завдань, дискусій та відстеження прогресу студентів;
- можливість використання платформи у навчальних закладах;
- відкритість та гнучкість структури вивчення.

Недоліки:

- складний інтерфейс для вивчення та вивчення для нових користувачів;
- велика кількість налаштувань, що дозволяє доокремо налаштувати кожен елемент;
- потреба в технічній підтримці та адмініструванні системи;
- надмірність функціоналу для такої малій платформи;
- потреба у налаштуванні дизайну до встановленого досвіду користувача.

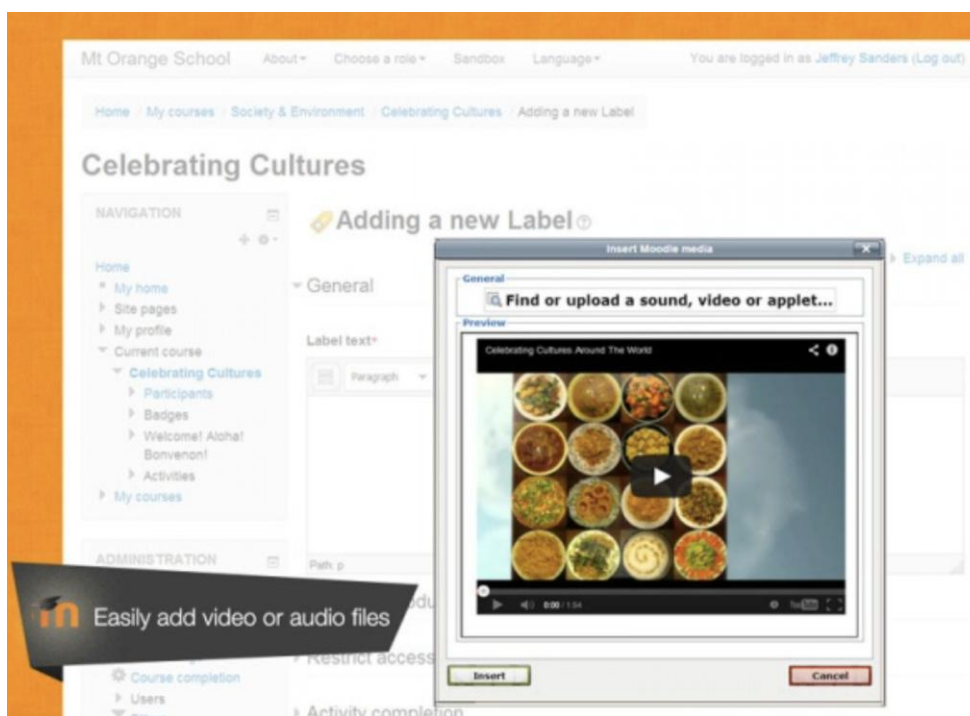


Рис. 2.1 Приклад інтерфейсу системи Moodle

2.2 Google Classroom

Google Classroom є веб-сервісом, що допомагає викладачам організувати класи, розміщувати та передавати навчальні матеріали студентам та іншим викладачам. Саме завдяки цьому інструменту відбувається спрощення взаємодії між викладачами та студентами. Викладач може створити класи, додати до них студентів, розмістити матеріали та навчальні завдання для перевірки [4].

Особливості Google Classroom відображаються в його простоті використання. Від студентів, які використовують Google Classroom, не вимагається технічна підготовка. У системі може бути просте створення і класів, і завдань, а також журнал оцінювання по відповідним курсам [4].

Переваги Google Classroom:

- Простий та зрозумілий інтерфейс;
- Швидке створення курсів та навчальних завдань;
- Є інтегрованим з Google Drive, Google Docs, Google Meet;
- Оцінювання студентів відбувається впродовж навчання;

- Доступність сервісу через веб-браузер.

Недоліки Google Classroom:

- Обмежується можливостями створення класів;
- Відсутність каталогу класів;
- Відсутність аналітики по класам;
- Залежність від Google;
- Можливості створення спеціалізованих ІТ курсів - обмежені.

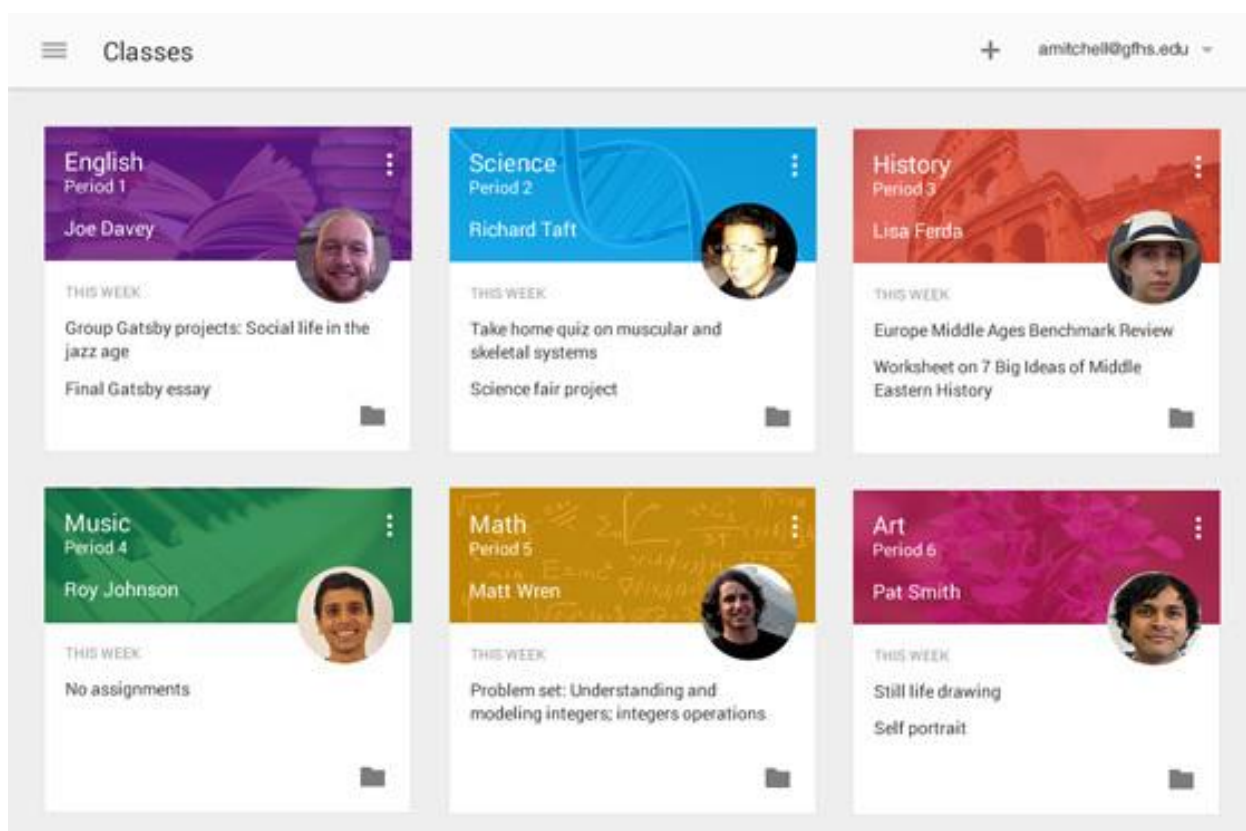


Рис. 2.2 Приклад інтерфейсу сервісу Google Classroom

2.3 Coursera

Coursera є одним із популярних інтернет-платформ для навчання, на якій компанії, школи або заклади вищої освіти викладають для студентів відеоуроки, матеріали до них, тестування та можливість отримання сертифікатів після їхнього успішного проходження [5].

Розробка платформи Coursera вказує на те, що цей сервіс відрізняється від інших і пропонує продукт з чіткою структурою курсу, зручним прогресом учнів та можливістю проходження курсів у відповідному власному напрямку [5].

Переваги платформи Coursera:

- велика кількість курсів і відеоуроків за різними напрямками;
- наявність багатьох курсів в галузі інформатики, сертифікатів та спеціалізацій;
- добре структуровані навчальні матеріали;
- тестування та виконання практичних завдань;
- можливість вивчення курсів з отриманням сертифікатів;
- зручна інтерфейс для відстеження відновленого прогресу учнів.

Недоліки платформи Coursera:

- більшість курсів та сертифікатів є платними для студентів;
- неможливість вільно створювати курси тільки для себе;
- платформа спрямована на масове навчання;
- труднощі у встановленні зміни функціоналу платформи за вимогами студентського проекту;
- складно виконувати всю платформу тільки в рамках студентського проекту.

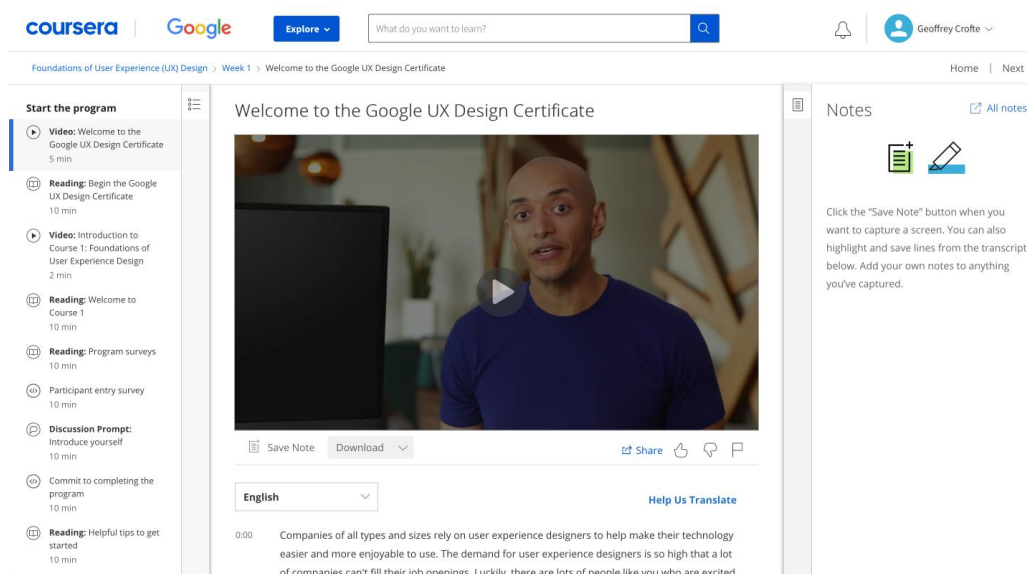


Рис. 2.3 Приклад інтерфейсу платформи Coursera

2.4 Udemу

Udemу — це платформа для проходження курсів. На платформі доступні різноманітні курси з різних напрямків, одним із яких є ІТ. Платформа орієнтована на дві основні групи: на студентів, які проходять курси, та на авторів, які створюють свої курси [6].

Udemу — це маркетплейс. Різні автори створюють курси, а учні самостійно обирають курси, що їх цікавлять. Udemу — один із найпопулярніших маркетплейсів у сфері освіти, де можна знайти різноманітні курси в галузі ІТ: Java, Python, веб-розробка, бази даних та інші [6].

На платформі Udemу можна знайти відеокурси, матеріали для завантаження, тести та сертифікати після проходження курсів. Важливою перевагою платформи є наявність практичних тестів, за допомогою яких учні можуть перевірити свої знання до або після проходження курсу [6].

Переваги платформи Udemу:

- Великий вибір курсів у сфері ІТ;
- Можливість для авторів створювати власні курси;
- Доступ до матеріалів;
- Наявність відеолекцій, тестів та практичних завдань;
- Можливість переглянути рейтинг та відгуки про курси перед тим, як обрати необхідний для себе курс;
- Видача сертифікатів після проходження курсу.

Недоліки платформи Udemу:

- Якість курсів залежить від авторів;
- Немає єдиної методики курсів;
- Деякі курси не містять необхідних практичних завдань;
- Обмежені можливості контролювати проходження курсів;
- Маркетплейс орієнтований на продаж курсів, а не на створення освітньої системи.

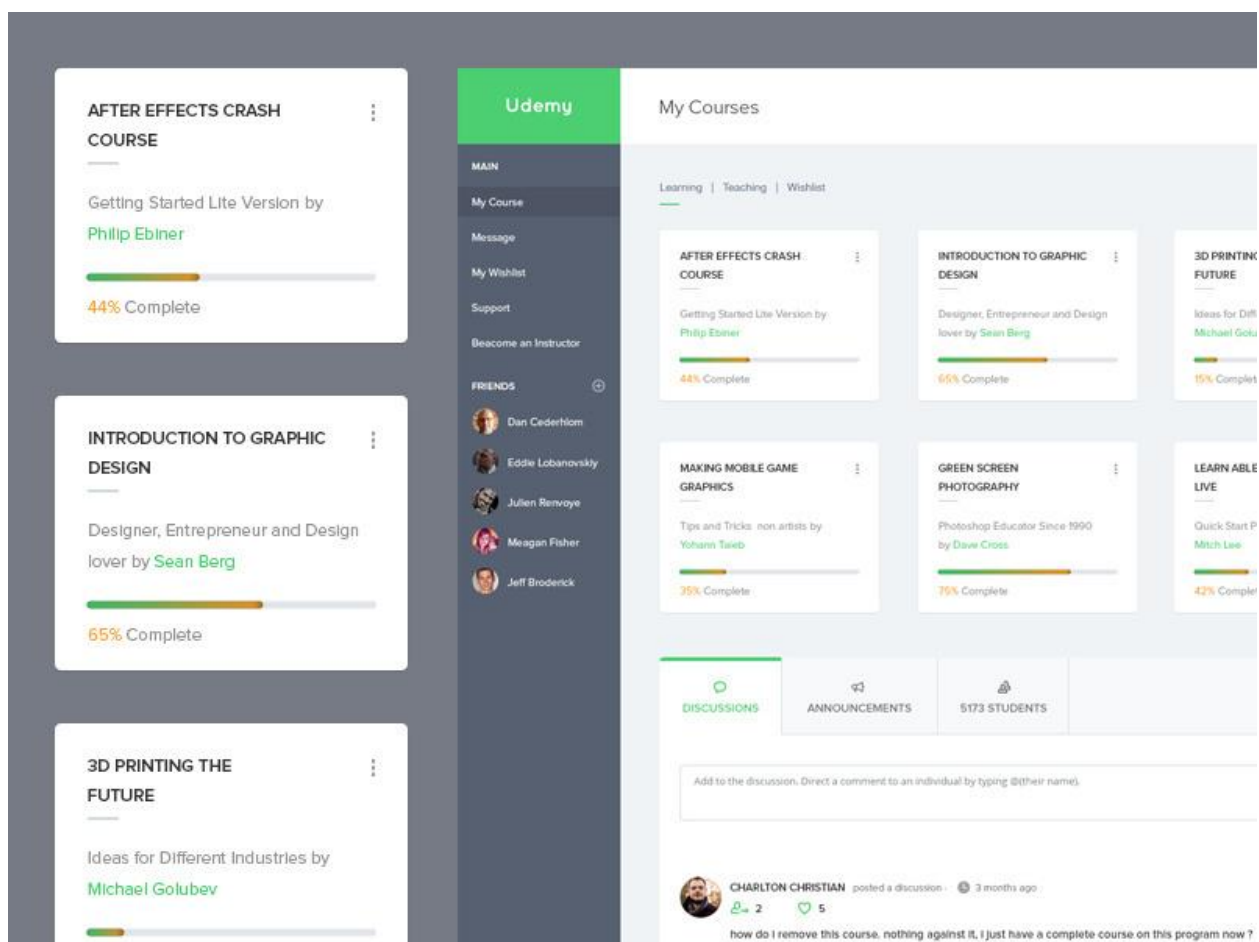


Рис. 2.4 Приклад інтерфейсу платформи Udeemy

2.5 Порівняльний аналіз програмних засобів

На основі проведеного аналізу можна зробити висновок, що існуючі програмні засоби мають достатньо широкий функціонал для організації онлайн-навчання. Moodle найбільше підходить для навчальних закладів, оскільки містить розвинені засоби керування курсами, тестами, оцінками та користувачами.

Google Classroom є простішим у використанні, однак має обмежені можливості для побудови повноцінної платформи з каталогом курсів, детальною структурою навчальних матеріалів та розширеною аналітикою прогресу. Coursera та Udeemy більше орієнтовані на масове самостійне навчання, доступ до готових курсів і сертифікацію [3; 4; 5; 6].

Водночас жоден із розглянутих засобів повністю не відповідає задачі розробки власної спеціалізованої платформи онлайн-навчання з ІТ-напрямів. Частина систем є надто складною або перевантаженою функціями, інші не дозволяють гнучко керувати власною структурою курсів, а деякі більше орієнтовані на комерційне розміщення готового контенту [3; 4; 5; 6].

Тому доцільною є розробка власного web-застосунку SkillHub, який міститиме необхідні функції для організації навчального процесу: каталог ІТ-курсів, модулі, уроки, тестування, збереження результатів, перегляд прогресу та адміністрування навчального контенту.

Унікальною особливістю SkillHub є можливість формування індивідуальної траєкторії навчання користувача на основі результатів проходження тестів і поточного прогресу. Це дозволяє не лише фіксувати факт проходження курсу, а й визначати, які теми вже засвоєні, які потребують повторного опрацювання та які матеріали доцільно проходити далі.

Таблиця 2.1

Порівняльна характеристика програмних засобів для онлайн-навчання

Показник	Moodle	Google Classroom	Coursera	Udemy	SkillHub
Каталог курсів	+	-	+	+	+
Структура курсу з модулями та уроками	+	частково	+	+	+
Ролі користувачів	+	частково	+	+	+
Створення власних курсів	+	частково	обмежено	+	+
Додавання уроків	+	+	обмежено	+	+
Тестування	+	частково	+	+	+
Відстеження прогресу	+	частково	+	+	+
Статистика навчання	+	частково	+	частково	+

Продовження таблиці 2.1

Порівняльна характеристика програмних засобів для онлайн-навчання

Показник	Moodle	Google Classroom	Coursera	Udemy	SkillHub
Індивідуальна траєкторія навчання	частково	-	частково	-	+
Визначення тем, які потребують повторення	частково	-	частково	-	+
Адміністрування навчального контенту	+	частково	обмежено	+	+
Зручність для початківця	середня	висока	висока	висока	висока
Гнучкість налаштування	висока	середня	низька	середня	висока
Орієнтація на IT-курси	частково	частково	+	+	+
Основні переваги	Повноцінна LMS, багато функцій, звіти, оцінювання	Простота, інтеграція з Google-сервісами	Якісна структура курсів, сертифікати	Великий вибір курсів, можливість авторам публікувати матеріали	Орієнтація на IT-курси, проста структура, тестування, прогрес, індивідуальна траєкторія навчання
Основні недоліки	Складність налаштування, перевантажений інтерфейс	Обмежена аналітика та структура курсів	Обмежене створення власних курсів, комерційна модель	Різна якість курсів, залежність від автора	Потребує подальшого розширення функцій після базової реалізації

3. ОГЛЯД ЗАСОБІВ РЕАЛІЗАЦІЇ ДОДАТКУ ДЛЯ ОНЛАЙН-НАВЧАННЯ З ІТ-НАПРЯМІВ

Для реалізації платформи онлайн-навчання з ІТ-напрямів необхідно обрати такі засоби програмної розробки, які дозволять створити зручний веб-інтерфейс, надійну серверну частину, базу даних для зберігання інформації та механізми взаємодії між усіма компонентами системи. Оскільки розроблюваний застосунок передбачає роботу з користувачами, курсами, модулями, уроками, тестами та результатами навчання, доцільним є використання клієнт-серверної архітектури [7; 8; 9; 11; 12].

У межах такої архітектури клієнтська частина відповідає за взаємодію з користувачем, відображення навчальних матеріалів, форм тестування, каталогу курсів і прогресу. Серверна частина реалізує бізнес-логіку системи, обробляє запити, перевіряє права доступу, працює з результатами тестування та передає дані до бази даних. База даних використовується для зберігання користувачів, курсів, модулів, уроків, тестових питань і результатів проходження тестів [8; 21; 22].

3.1 Мова програмування Java

Для реалізації серверної частини платформи доцільно використати мову програмування **Java**. Вона є однією з поширених мов для створення веб-застосунків, корпоративних інформаційних систем і серверного програмного забезпечення. Java має розвинену екосистему, підтримує об'єктно-орієнтований підхід і дозволяє створювати структурований, масштабований та зрозумілий програмний код. Офіційна документація Java містить навчальні матеріали, API-довідники та приклади, які можуть використовуватись для розробки і масштабування застосунків [7].

У межах платформи онлайн-навчання Java може бути використана для опису основних сутностей предметної галузі: користувачів, курсів, модулів, уроків,

тестів, питань і результатів. Завдяки об'єктно-орієнтованому підходу кожен елемент системи можна представити окремою моделлю, що спрощує подальшу розробку та підтримку програмного продукту [7; 20].

Перевагами використання Java для даної системи є:

- можливість створення складної серверної логіки;
- підтримка об'єктно-орієнтованого програмування;
- широка екосистема бібліотек і фреймворків;
- зручність роботи з базами даних;
- придатність для створення клієнт-серверних веб-застосунків;
- можливість подальшого масштабування системи.

3.2 Spring Boot та Spring Framework

Для спрощення створення серверної частини доцільно використати **Spring Framework** і **Spring Boot**. Spring Boot дозволяє швидко створювати веб-застосунки, налаштовувати серверну логіку та формувати REST API для взаємодії клієнтської частини з backend-рівнем. В офіційних матеріалах Spring показано, що застосунок Spring Boot може запускатися через основний клас із анотацією `@SpringBootApplication`, що спрощує початкове налаштування проєкту [8].

У розроблюваній платформі Spring Boot може використовуватися для створення API, через яке клієнтська частина отримуватиме дані про курси, модулі, уроки, тести, користувачів і результати навчання. Наприклад, окремі контролери можуть відповідати за роботу з курсами, авторизацією, тестуванням і статистикою. Такий підхід дозволяє розділити систему на логічні модулі та зробити її більш зручною для підтримки [8].

Spring Boot також добре підходить для реалізації RESTful-сервісів. Офіційний гайд Spring демонструє створення RESTful Web Service, що підтверджує доцільність використання цієї технології для побудови API веб-застосунку [8].

Переваги Spring Boot для даного проєкту:

- швидке створення серверної частини;
- зручна організація REST API;
- можливість розділення логіки на контролери, сервіси та репозиторії;
- інтеграція з базами даних;
- підтримка авторизації та безпеки;
- зручність подальшого розширення функціоналу.

3.3 PostgreSQL

Для зберігання даних платформи онлайн-навчання доцільно використати **PostgreSQL**. Це об'єктно-реляційна система керування базами даних із відкритим кодом, яка використовує та розширює мову SQL і призначена для надійного зберігання складних наборів даних [9].

У межах розроблюваної системи PostgreSQL може зберігати дані про користувачів, ролі, курси, модулі, уроки, тестові питання, варіанти відповідей, спроби проходження тестів і результати навчання. Така структура дозволяє організувати зв'язки між сутностями системи та забезпечити коректне збереження навчальної інформації [9; 10].

Окремою перевагою PostgreSQL є наявність широкого набору типів даних. В офіційній документації зазначено, що PostgreSQL має багатий набір вбудованих типів і дозволяє додавати власні типи даних. Це корисно для проєктів, де потрібно зберігати різні види інформації [10].

Переваги PostgreSQL для платформи онлайн-навчання:

- надійне зберігання структурованих даних;
- підтримка складних зв'язків між таблицями;
- використання SQL-запитів;
- можливість створення звітів і вибірок;
- придатність для зберігання результатів навчання;
- відкритий код і широка документація.

3.4 React та TypeScript

Для створення клієнтської частини платформи доцільно використати **React**. Це бібліотека для побудови користувацьких інтерфейсів на основі компонентного підходу. Офіційна документація React описує створення інтерфейсів із незалежних компонентів, які потім можуть поєднуватися у повноцінні сторінки та застосунки [11].

У платформі онлайн-навчання React може використовуватися для створення сторінки входу, каталогу курсів, сторінки курсу, сторінки уроку, тестової форми, профілю користувача та сторінки перегляду прогресу. Компонентний підхід дозволяє повторно використовувати окремі елементи інтерфейсу, наприклад картки курсів, кнопки, форми, списки модулів і блоки результатів [11].

Разом із React доцільно використовувати **TypeScript**, оскільки він додає типізацію до JavaScript-коду. Це дозволяє зменшити кількість помилок під час розробки та зробити структуру даних більш зрозумілою. Офіційна документація React зазначає, що TypeScript є популярним способом додавання типів до JavaScript-проектів, а також може використовуватися з React-компонентами [12].

Переваги React і TypeScript для даного проєкту:

- створення інтерактивного веб-інтерфейсу;
- компонентна структура клієнтської частини;
- повторне використання елементів інтерфейсу;
- зручна робота зі сторінками курсів, уроків і тестів;
- типізація даних за допомогою TypeScript;
- зменшення кількості помилок під час розробки.

4. ПРОЄКТУВАННЯ WEB-ЗАСТОСУНКУ SKILLHUB

4.1 Логічна структура системи

Web-застосунок SkillHub проєктується як клієнт-серверна система, що забезпечує організацію онлайн-навчання з ІТ-напрямів. Логічна структура системи передбачає поділ програмного продукту на окремі рівні, кожен з яких виконує визначену роль у роботі застосунку. Такий підхід дозволяє розділити користувацький інтерфейс, бізнес-логіку та роботу з даними, що спрощує розробку, тестування, супровід і подальше розширення системи [18; 21; 22].

У структурі web-застосунку SkillHub виділено три основні рівні: рівень представлення, рівень бізнес-логіки та рівень даних. Рівень представлення відповідає за взаємодію користувачів із системою. До нього належать web-інтерфейс студента та web-інтерфейс адміністратора або автора курсу. Студент використовує систему для перегляду каталогу курсів, проходження уроків, виконання тестових завдань і перегляду власного прогресу. Автор курсу та адміністратор працюють із функціями створення, редагування і керування навчальним контентом.

Рівень бізнес-логіки містить основні функціональні модулі системи. До нього належать модуль користувачів, модуль автентифікації, модуль курсів, модуль уроків, модуль тестування, а також модуль аналітики та прогресу. Модуль користувачів забезпечує роботу з обліковими записами та ролями. Модуль автентифікації відповідає за вхід до системи та перевірку прав доступу. Модуль курсів використовується для створення і перегляду навчальних курсів, а модуль уроків — для роботи з навчальними матеріалами. Модуль тестування забезпечує проходження контрольних завдань, перевірку відповідей і збереження результатів. Модуль аналітики та прогресу дозволяє відстежувати навчальні досягнення користувача та формувати дані щодо засвоєння матеріалу.

Рівень даних відповідає за доступ до бази даних і збереження основної інформації системи. Для взаємодії серверної частини з базою даних

використовується Spring Data JPA. У базі даних PostgreSQL зберігаються відомості про користувачів, курси, модулі, уроки, тести та результати проходження навчання. Така структура дозволяє організувати зв'язки між основними сутностями системи та забезпечити коректну роботу навчального процесу [8; 9; 10].

Логічна архітектура web-застосунку SkillHub наведена на рисунку 4.1.

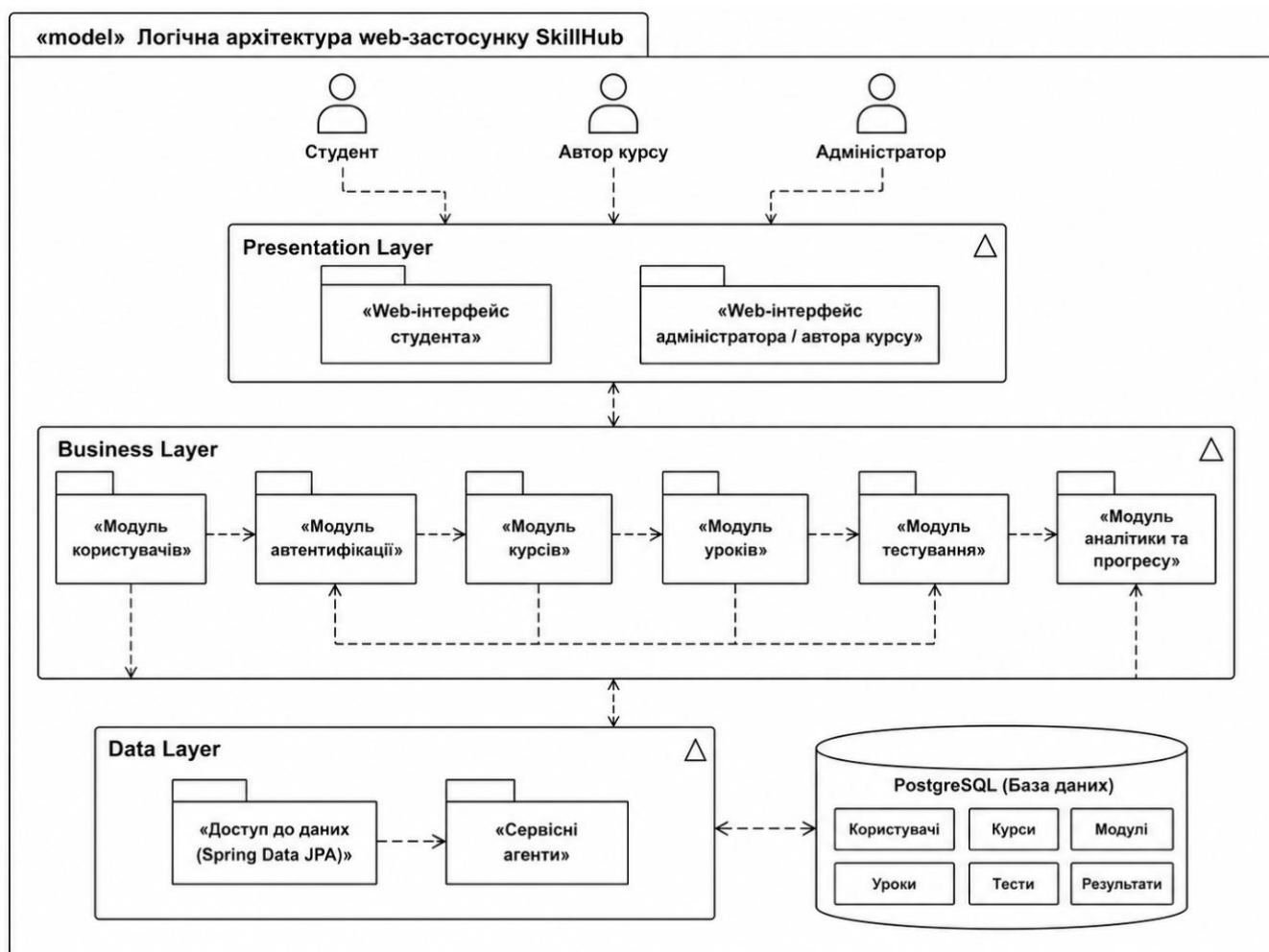


Рис. 4.1 Логічна архітектура web-застосунку SkillHub

4.2 Діаграма прецедентів web-застосунку SkillHub

Для опису функціональних можливостей web-застосунку SkillHub доцільно використати діаграму прецедентів. Вона дозволяє відобразити основних користувачів системи та показати, з якими функціями вони взаємодіють. У межах розроблюваної платформи виділено три основні ролі: студент, автор курсу та адміністратор [18].

Студент є основним користувачем платформи. Він може зареєструватися в системі, виконати вхід, переглядати каталог ІТ-курсів, відкривати сторінку обраного курсу, проходити навчальні модулі й уроки, виконувати тестові завдання та переглядати власні результати. Також студент має доступ до інформації про свій навчальний прогрес, що дозволяє йому контролювати проходження курсу та визначати, які теми вже опрацьовані.

Автор курсу відповідає за створення та наповнення навчального контенту. До його функцій належать створення нових курсів, додавання модулів, уроків, тестових завдань, редагування навчальних матеріалів та перегляд статистики навчання за створеними курсами. Така роль необхідна для того, щоб платформа могла поповнюватися новими навчальними матеріалами без прямого втручання адміністратора у кожен окремий елемент контенту.

Адміністратор має розширені права доступу до системи. Він може керувати користувачами, контролювати ролі, переглядати статистику навчання, редагувати або видаляти курси, а також здійснювати загальне адміністрування платформи. Наявність адміністративної ролі дозволяє забезпечити контроль за наповненням системи, коректністю навчальних матеріалів і роботою користувачів.

Діаграма прецедентів web-застосунку SkillHub наведена на рисунку 4.2.



Рис. 4.2 Діаграма прецедентів web-застосунку SkillHub

У результаті побудови діаграми прецедентів визначено основні сценарії взаємодії користувачів із системою. Найважливішими прецедентами для студента є перегляд курсів, проходження уроків, тестування та перегляд прогресу. Для автора курсу ключовими є створення і редагування навчального контенту. Для адміністратора основними функціями є керування користувачами, контроль курсів і перегляд статистики. Такий розподіл прецедентів дозволяє чітко визначити межі відповідальності кожної ролі та сформулювати основу для подальшого проєктування модулів системи [18].

4.3 Діаграма доменної моделі web-застосунку SkillHub

Для опису основних інформаційних об'єктів web-застосунку SkillHub доцільно побудувати діаграму доменної моделі. Така діаграма дозволяє визначити ключові сутності предметної галузі, їх основні властивості та взаємозв'язки між ними. Доменна модель є важливою частиною проєктування, оскільки на її основі надалі формується структура бази даних, серверні класи та логіка взаємодії між модулями системи [18; 21].

Основною сутністю системи є **користувач**. Користувач може мати одну з ролей: студент, автор курсу або адміністратор. Роль визначає доступні функції в системі. Студент має можливість переглядати курси, проходити уроки, виконувати тестові завдання та переглядати власний прогрес. Автор курсу створює та редагує навчальні матеріали. Адміністратор здійснює загальне керування платформою, користувачами та навчальним контентом.

Навчальний процес у системі побудований навколо сутності **курс**. Курс містить загальну інформацію про навчальну програму, її назву, опис, напрям, автора та список навчальних модулів. Кожен курс складається з окремих **модулів**, які, у свою чергу, містять **уроки**. Урок є структурною одиницею навчального матеріалу та може містити текстову інформацію, практичні пояснення, посилання або додаткові матеріали.

Для перевірки знань у системі передбачено сутність **тест**. Тест може бути пов'язаний із курсом, модулем або окремим уроком. Він складається з тестових питань, кожне з яких має варіанти відповідей. Після проходження тесту система створює запис про спробу тестування та зберігає результат користувача. Це дозволяє надалі відображати прогрес навчання та формувати статистику.

Окремою важливою сутністю є **прогрес користувача**. Він відображає, які курси, модулі або уроки вже були переглянуті студентом, які тести були пройдені та які результати отримано. На основі цих даних система може визначати рівень засвоєння матеріалу та формувати індивідуальну траєкторію навчання користувача.

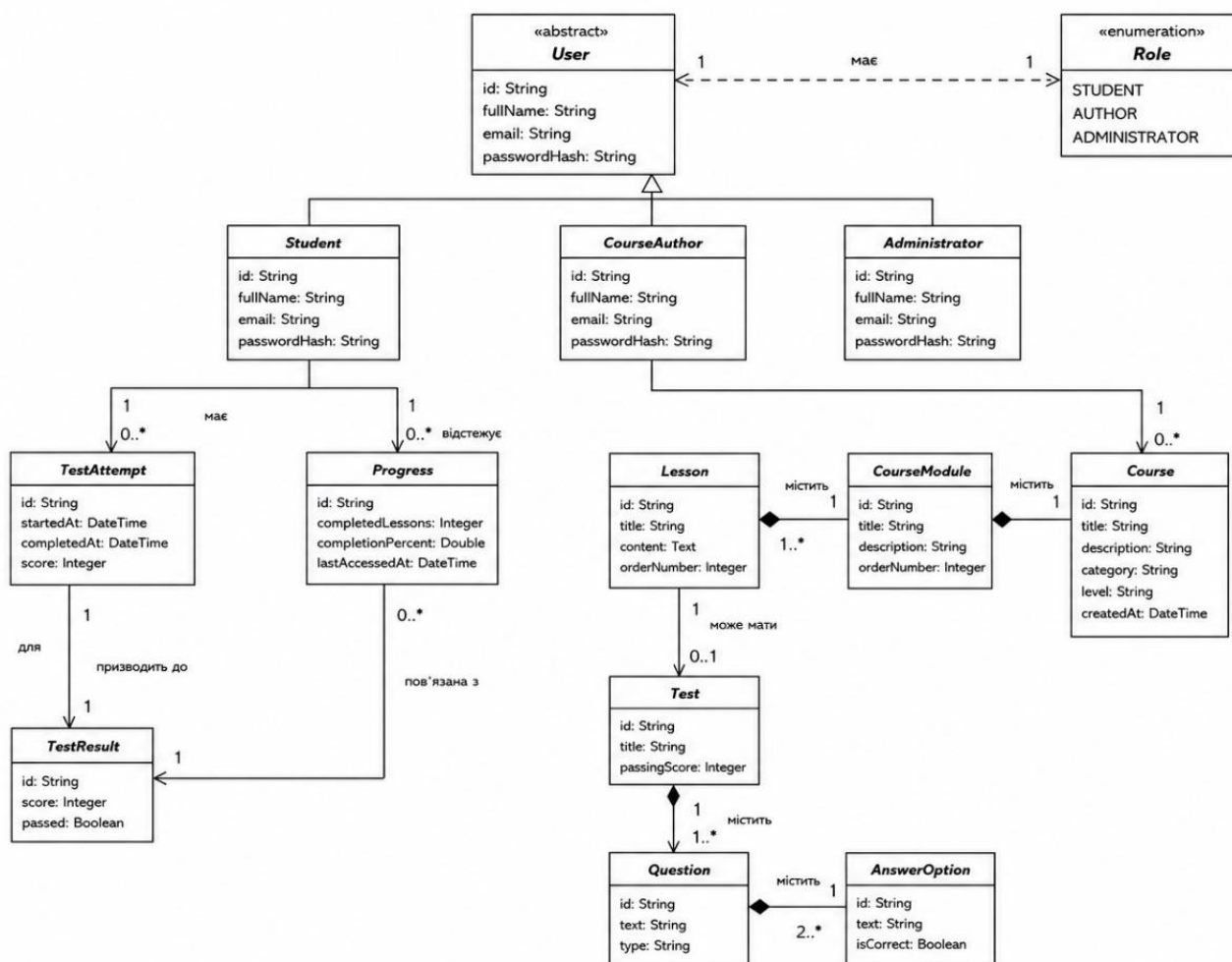


Рис. 4.3 Діаграма доменної моделі web-застосунку SkillHub

У доменній моделі web-застосунку SkillHub основними сутностями є User, Role, Course, CourseModule, Lesson, Test, Question, AnswerOption, TestAttempt, TestResult та Progress. Між ними існують логічні зв'язки: користувач має роль, курс містить модулі, модуль містить уроки, урок може мати тест, тест складається з питань, питання містить варіанти відповідей, а результати тестування та прогрес пов'язані з конкретним користувачем. Така структура дозволяє описати предметну галузь онлайн-навчання та підготувати основу для подальшого проєктування бази даних і класів програмної системи [18; 21].

4.4 Діаграми діяльності

Діаграми діяльності використовуються для опису послідовності дій, які виконуються користувачами під час роботи з web-застосунком SkillHub. Вони дозволяють показати логіку переходів між етапами, умови виконання окремих дій та результат взаємодії користувача із системою [18].

У межах проєктування платформи SkillHub доцільно розглянути два основні сценарії роботи системи: процес проходження курсу студентом та процес створення навчального курсу автором або адміністратором. Перший сценарій описує основну навчальну логіку платформи, другий — процес наповнення системи навчальним контентом.

4.4.1 Діаграма діяльності процесу проходження курсу студентом

Процес проходження курсу студентом починається з авторизації користувача в системі. Після успішного входу студент переходить до каталогу курсів, переглядає доступні навчальні програми та обирає потрібний курс. Далі система відкриває структуру курсу, яка складається з модулів, уроків і тестових завдань.

Після вибору модуля студент переходить до вивчення уроку. Якщо навчальний матеріал ще не опрацьовано, користувач може повернутися до повторного перегляду уроку. Після завершення вивчення матеріалу студент переходить до тестування. Система надає тестові питання, приймає відповіді користувача, виконує перевірку результатів і зберігає отримані дані в базі даних.

Після завершення тестування студент отримує результат і може переглянути оновлений прогрес за курсом. Якщо тест складено успішно, користувач може перейти до наступного уроку або модуля. Якщо результат є недостатнім, студент може повторно опрацювати матеріал і пройти тест ще раз.



Рис. 4.4 Діаграма діяльності процесу проходження курсу студентом

4.4.2 Діаграма діяльності процесу створення навчального курсу

Процес створення навчального курсу виконується автором курсу або адміністратором. Спочатку користувач проходить авторизацію та отримує доступ до адміністративної частини платформи. Після цього він обирає функцію створення нового курсу та вводить основну інформацію: назву, опис, категорію, рівень складності та інші параметри.

Після створення курсу автор формує його структуру. Для цього до курсу додаються навчальні модулі, а до кожного модуля — окремі уроки. Уроки наповнюються навчальним матеріалом, який може містити текстові пояснення, приклади, посилання або додаткові ресурси. Далі автор додає тестові завдання для перевірки засвоєння матеріалу.

Після заповнення курсу система може перевірити повноту введених даних. Якщо в курсі відсутні обов'язкові елементи, наприклад модулі, уроки або тестові питання, автор повертається до редагування. Якщо структура курсу заповнена коректно, курс може бути збережений і опублікований для студентів.

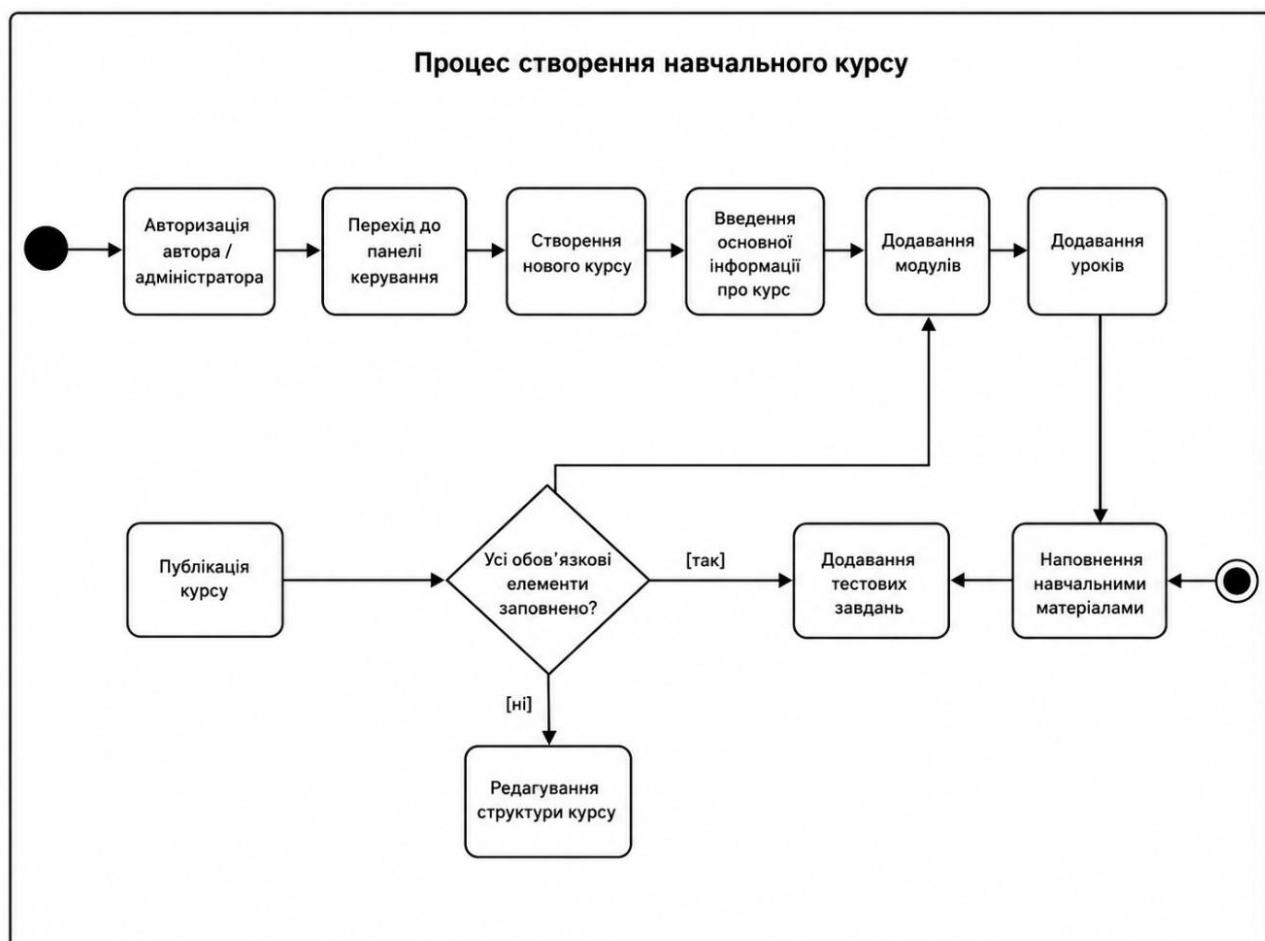


Рис. 4.5 Діаграма діяльності процесу створення навчального курсу

Побудова діаграм діяльності дозволяє деталізувати основні сценарії функціонування web-застосунку SkillHub. Завдяки цьому можна чітко визначити порядок дій користувачів, умови переходів між етапами та точки взаємодії з базою даних. Надалі ці сценарії можуть бути використані під час реалізації клієнтської частини, серверної логіки та тестування програмного забезпечення [18; 21].

4.5 Діаграма класів web-застосунку SkillHub

Діаграма класів використовується для опису програмної структури web-застосунку SkillHub, а також для відображення основних класів, їх методів і взаємозв'язків між компонентами серверної частини. На відміну від доменної моделі, яка описує сутності предметної галузі, діаграма класів показує логіку реалізації системи на рівні контролерів, сервісів і репозиторіїв [18; 20].

У web-застосунку SkillHub серверна частина побудована за багаторівневою структурою, де кожен рівень відповідає за окрему частину обробки даних. Контролери приймають запити від клієнтської частини, сервіси виконують основну бізнес-логіку, а репозиторії забезпечують взаємодію з базою даних. Такий підхід дозволяє розділити відповідальність між класами та зробити програмний код більш зрозумілим і зручним для подальшого супроводу [21; 22].

До рівня контролерів належать класи **UserController**, **AdminController**, **SkillPathController**, **AuthorController**, **DashboardController** та **CourseController**. Клас **UserController** відповідає за роботу з поточним користувачем, зокрема отримання даних профілю та його оновлення. **AdminController** використовується для адміністративного керування системою: отримання статистики, перегляду користувачів, зміни ролей, видалення користувачів, перегляду курсів та видалення курсів. **AuthorController** забезпечує роботу автора курсу з власними навчальними матеріалами, включаючи створення, оновлення та видалення курсів. **CourseController** відповідає за перегляд курсів, отримання конкретного курсу, завершення уроку та надсилання результатів тестування. **SkillPathController** використовується для отримання індивідуальної траєкторії навчання, а **DashboardController** — для формування даних панелі користувача.

Сервісний рівень представлений класами **AdminService**, **CurrentUserService**, **SkillPathService** та **CourseService**. Клас **CurrentUserService** забезпечує отримання поточного користувача, формування відповіді з даними користувача та оновлення профілю. **AdminService** реалізує логіку адміністративного керування користувачами, ролями, курсами та статистикою.

CourseService відповідає за основну роботу з курсами: отримання списку курсів, отримання курсу за ідентифікатором, завершення уроку, обробку результатів тестування, створення, оновлення та видалення курсу. **SkillPathService** реалізує логіку формування індивідуального навчального шляху користувача, пошуку наступного уроку, визначення слабких напрямів і розрахунку готовності користувача до подальшого проходження навчання.

Рівень доступу до даних представлений репозиторіями **UserRepository**, **QuizAttemptRepository**, **CourseRepository**, **LessonProgressRepository** та **LessonRepository**. Вони використовуються сервісами для отримання, збереження, оновлення та видалення даних у базі даних. Наприклад, **UserRepository** забезпечує доступ до даних користувачів, **CourseRepository** — до курсів, **LessonRepository** — до уроків, **LessonProgressRepository** — до прогресу проходження уроків, а **QuizAttemptRepository** — до результатів тестових спроб.

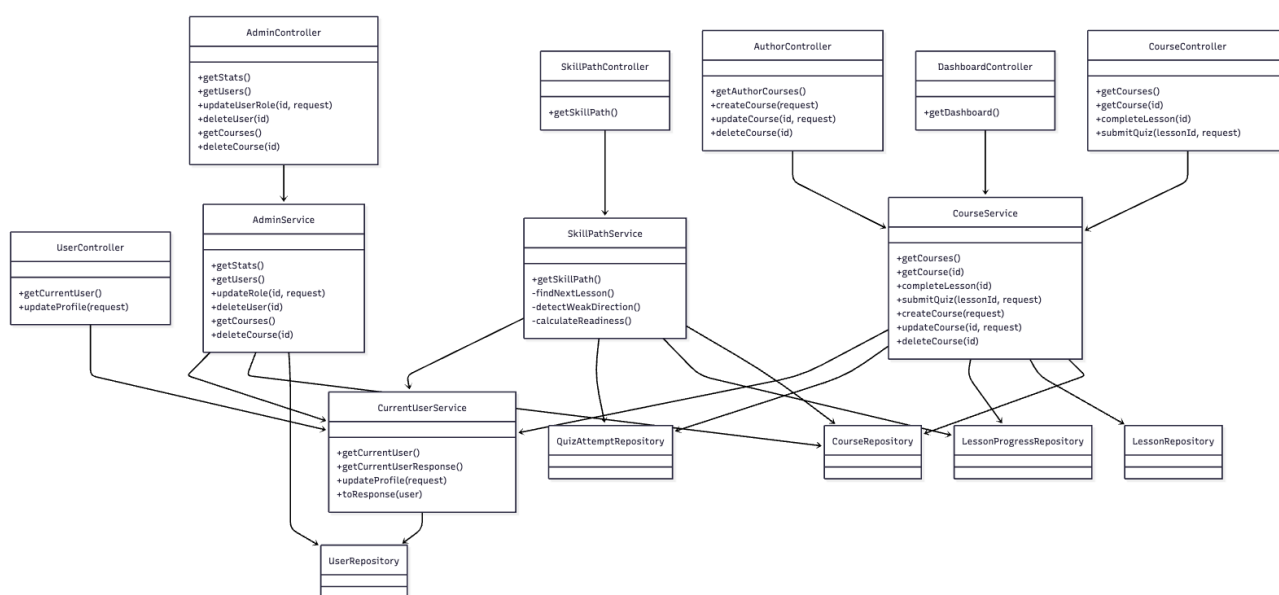


Рис. 4.6 Діаграма класів web-застосунку SkillHub

Побудована діаграма класів демонструє, що web-застосунок SkillHub має чітко розділену структуру. Контролери не працюють напряму з базою даних, а передають запити до сервісів. Сервіси, у свою чергу, виконують бізнес-логіку та

звертаються до відповідних репозиторіїв. Такий підхід відповідає принципу розділення відповідальності та спрощує подальше розширення системи, наприклад додавання нових типів курсів, розширеної аналітики прогресу або додаткових функцій для автора курсу й адміністратора [21; 22].

4.6 Діаграма компонентів web-застосунку SkillHub

Діаграма компонентів використовується для відображення загальної структури програмного забезпечення на рівні основних функціональних частин системи. Вона дозволяє показати, з яких компонентів складається web-застосунок SkillHub, як вони взаємодіють між собою та які технологічні рівні забезпечують роботу платформи [19; 21].

Web-застосунок SkillHub побудований за клієнт-серверною архітектурою. Основними компонентами системи є клієнтська частина, серверна частина та база даних. Клієнтська частина реалізується за допомогою React і TypeScript та відповідає за відображення інтерфейсу користувача. Через неї студент, автор курсу або адміністратор взаємодіє з функціями платформи: переглядає курси, проходить уроки, виконує тестування, переглядає прогрес або керує навчальним контентом [7; 8; 9; 11; 12].

Серверна частина реалізується за допомогою Java та Spring Boot. Вона приймає HTTP-запити від клієнтської частини, обробляє їх через REST API, виконує бізнес-логіку та звертається до бази даних.

До основних серверних компонентів належать контролери, сервіси та репозиторії. Контролери забезпечують приймання запитів і передавання відповідей клієнтській частині. Сервіси виконують логіку роботи з користувачами, курсами, уроками, тестами, прогресом і навчальним шляхом. Репозиторії відповідають за доступ до даних, які зберігаються у PostgreSQL [7; 8].

Окремими функціональними компонентами серверної частини є модуль користувачів, модуль адміністрування, модуль курсів, модуль уроків, модуль тестування, модуль прогресу та модуль індивідуальної траєкторії навчання.

Модуль користувачів забезпечує роботу з профілем користувача та його роллю. Модуль адміністрування дає змогу керувати користувачами, курсами та статистикою. Модуль курсів відповідає за створення, редагування, видалення та перегляд курсів. Модуль уроків забезпечує доступ до навчальних матеріалів і фіксацію проходження уроків. Модуль тестування обробляє відповіді користувача та зберігає результати. Модуль прогресу використовується для відображення стану проходження навчання. Модуль індивідуальної траєкторії навчання аналізує результати користувача та допомагає визначити наступні навчальні кроки.

База даних PostgreSQL використовується для зберігання основних даних web-застосунку: користувачів, курсів, модулів, уроків, тестів, результатів тестування та прогресу.

Взаємодія серверної частини з базою даних здійснюється через Spring Data JPA, що дозволяє працювати з даними на рівні репозиторіїв і спрощує реалізацію операцій створення, читання, оновлення та видалення [9; 10]. Даний підхід забезпечує зручність супроводу системи, спрощує тестування окремих частин застосунку та дозволяє надалі розширювати платформу новими функціями без повної зміни її архітектури (див. Рис. 4.7) [19; 21].

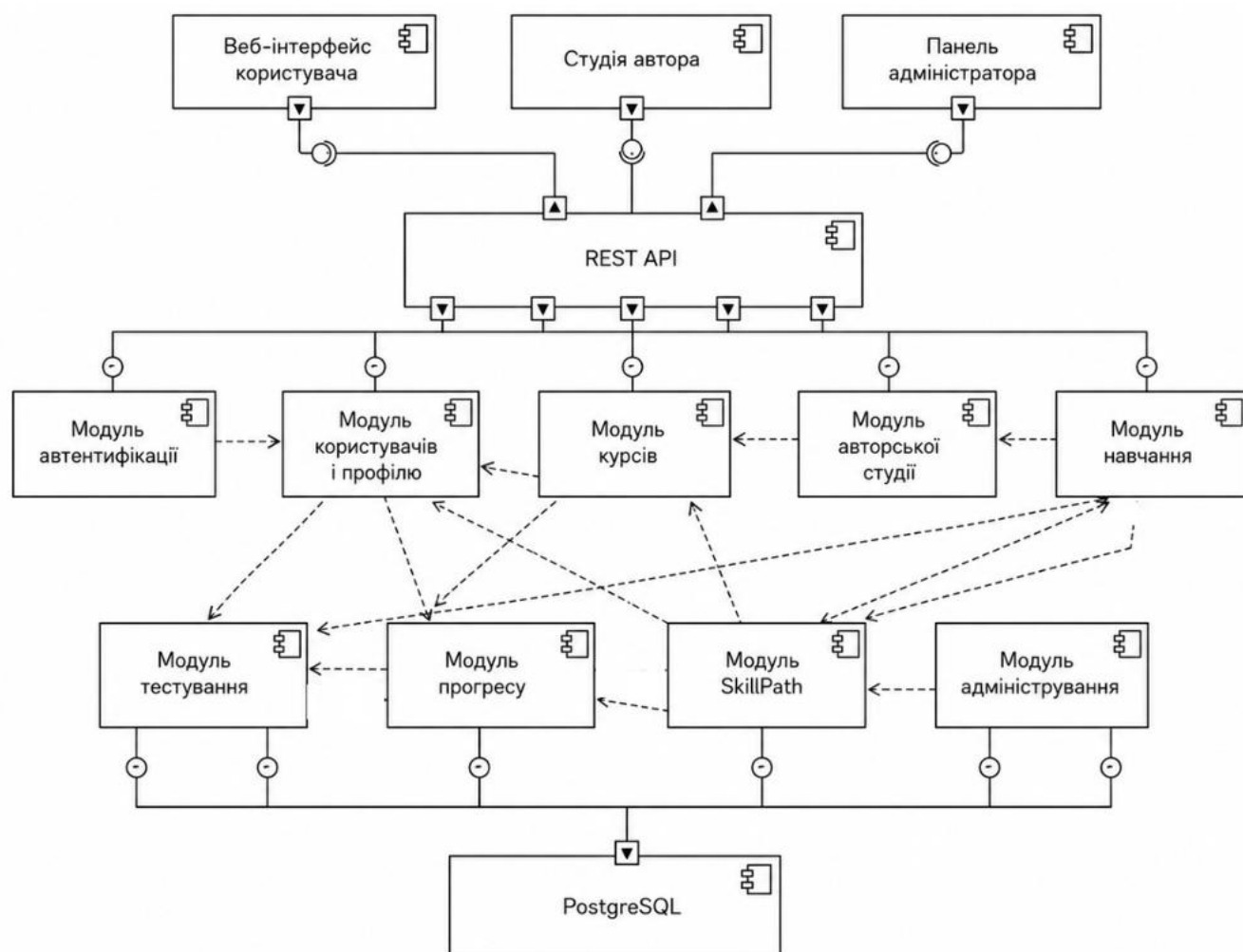


Рис. 4.7 Діаграма компонентів web-застосунку SkillHub

Побудована діаграма компонентів демонструє, що web-застосунок SkillHub має модульну структуру. Кожен компонент виконує окрему функцію та взаємодіє з іншими компонентами через визначені інтерфейси.

4.7 Діаграма розгортання web-застосунку SkillHub

Діаграма розгортання використовується для відображення фізичної структури розміщення web-застосунку SkillHub та показує, на яких вузлах виконуються основні компоненти системи. Вона дозволяє описати, як клієнтська частина, серверна частина та база даних взаємодіють між собою під час роботи програмного продукту [18; 21].

Web-застосунок SkillHub передбачає взаємодію користувача із системою через web-браузер. Користувач може працювати з платформою як студент, автор курсу або адміністратор.

Через браузер він відкриває клієнтську частину застосунку, яка реалізована за допомогою React та TypeScript. Клієнтська частина відповідає за відображення сторінок, форм, каталогу курсів, уроків, тестів, прогресу та адміністративних елементів [11; 12].

Клієнтська частина надсилає HTTP-запити до серверної частини через REST API. Серверна частина реалізована за допомогою Java та Spring Boot і розміщується на сервері застосунку. Вона обробляє запити, виконує бізнес-логіку, перевіряє права доступу, працює з курсами, уроками, тестами, результатами та прогресом користувачів [7; 8].

Для зберігання даних використовується сервер бази даних PostgreSQL. У базі даних зберігається інформація про користувачів, ролі, курси, модулі, уроки, тести, тестові спроби, результати та прогрес навчання.

Серверна частина взаємодіє з PostgreSQL через рівень доступу до даних, реалізований за допомогою Spring Data JPA [9; 10].

У процесі розгортання також можуть використовуватися допоміжні засоби, зокрема GitHub для зберігання програмного коду та Docker для контейнеризації окремих компонентів системи. Це спрощує перенесення застосунку між середовищами, тестування та запуск програмного продукту на різних пристроях або серверах [23; 25].

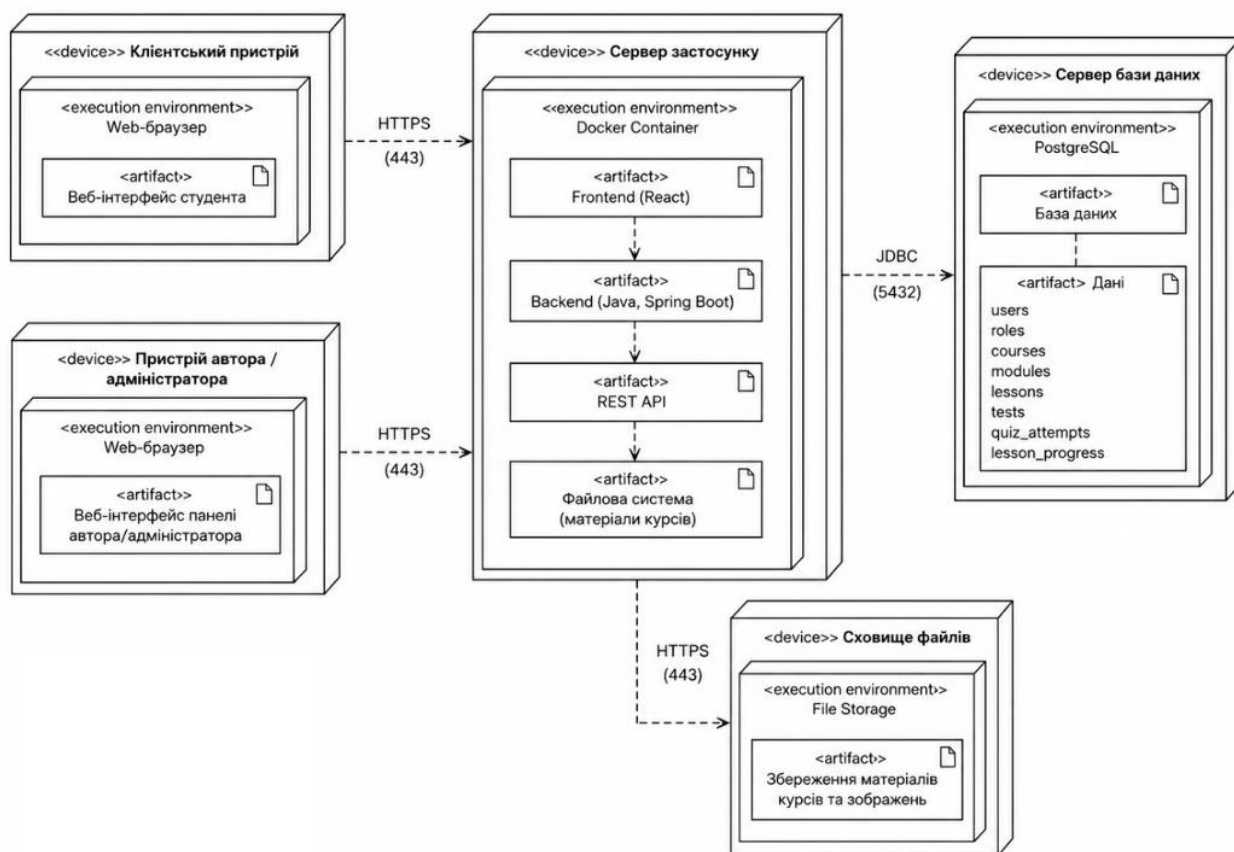


Рис. 4.8 Діаграма розгортання web-застосунку SkillHub

Побудована діаграма розгортання демонструє, що web-застосунок SkillHub має типову клієнт-серверну структуру. Користувач взаємодіє з системою через браузер, клієнтська частина передає запити до серверної частини, а серверна частина працює з базою даних.

Такий підхід забезпечує розділення відповідальності між компонентами, спрощує масштабування системи та дозволяє надалі розгортати платформу у локальному, хмарному або контейнеризованому середовищі [18; 21].

4.8 Проєктування бази даних

Для зберігання інформації web-застосунку SkillHub використовується реляційна база даних PostgreSQL. Структура бази даних побудована відповідно до логіки онлайн-навчання: користувачі створюють або проходять курси, курси

складаються з модулів, модулі містять уроки, а до уроків додаються тестові питання та варіанти відповідей. Окремо зберігаються дані про проходження уроків і результати тестування студентів [9; 10].

Основною таблицею системи є **USERS**, яка містить інформацію про користувачів платформи. У ній зберігаються email, пароль у хешованому вигляді, повне ім'я, коротка біографія, спеціалізація, навчальна мета, аватар, роль користувача та дата створення облікового запису. Поле *role* дозволяє розмежовувати користувачів за типами доступу: студент, автор курсу або адміністратор.

Для зберігання навчальних курсів використовується таблиця **COURSES**. Вона містить назву курсу, унікальний slug, напрям навчання, рівень складності, опис, тривалість, колір оформлення, обкладинку, статус публікації та посилання на власника курсу через поле *owner_id*. Це дозволяє визначити, який користувач створив конкретний курс.

Структура курсу деталізується за допомогою таблиць **COURSE_MODULES** та **LESSONS**. Таблиця **COURSE_MODULES** зберігає модулі курсу, їх назву, короткий опис і порядок відображення. Таблиця **LESSONS** містить уроки, що входять до складу модуля. Для кожного уроку зберігаються назва, основний навчальний матеріал, ключові тези, приклад коду, практичне завдання, орієнтовна тривалість і порядок відображення в межах модуля.

Для реалізації тестування використовуються таблиці **QUIZ_QUESTIONS** та **QUIZ_OPTIONS**. Таблиця **QUIZ_QUESTIONS** містить питання, прив'язані до конкретного уроку, пояснення до питання та порядок його відображення. Таблиця **QUIZ_OPTIONS** зберігає варіанти відповідей для кожного питання та позначку правильності відповіді.

Окремий блок бази даних пов'язаний із фіксацією навчального прогресу. Таблиця **LESSON_PROGRESS** зберігає інформацію про те, які уроки були завершені конкретним користувачем і коли саме це відбулося.

Таблиця **QUIZ_ATTEMPTS** використовується для збереження результатів проходження тестів: кількості правильних відповідей, загальної кількості питань, відсотка результату та дати завершення тестування.

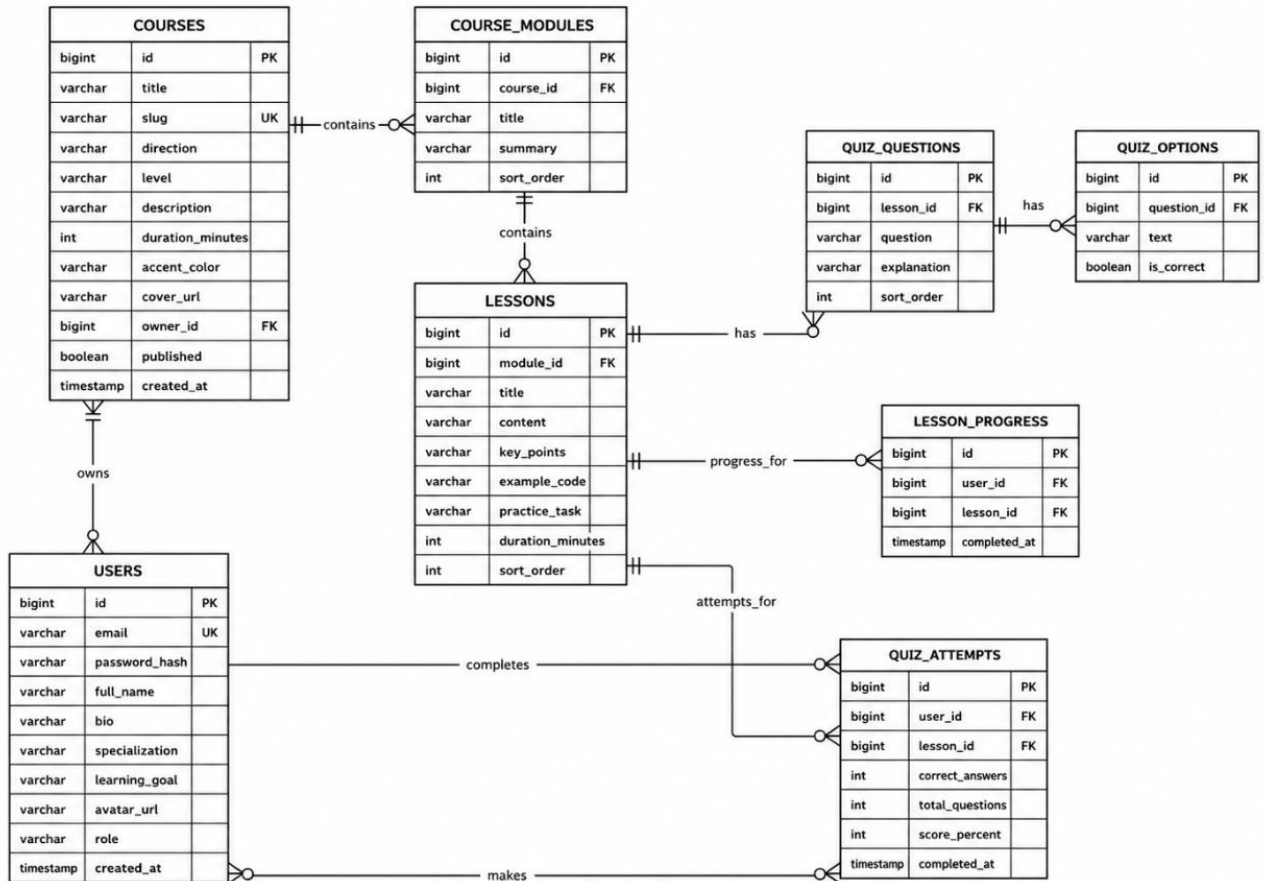


Рис. 4.9 Схема бази даних web-застосунку SkillHub

Зв'язки між таблицями реалізуються за допомогою зовнішніх ключів. Таблиця **USERS** пов'язана з таблицею **COURSES** через поле `owner_id`, що дозволяє визначити автора або власника курсу. Один користувач може створити декілька курсів, але кожен курс має одного власника [9; 10].

Таблиця **COURSES** пов'язана з таблицею **COURSE_MODULES** за допомогою поля `course_id`. Один курс може містити багато модулів. У свою чергу, таблиця **COURSE_MODULES** пов'язана з таблицею **LESSONS** через поле

`module_id`, що дозволяє побудувати послідовну структуру навчального матеріалу: курс → модуль → урок [9; 10].

Тестова частина системи реалізована через зв'язки між таблицями **LESSONS**, **QUIZ_QUESTIONS** та **QUIZ_OPTIONS**. Один урок може містити декілька тестових питань, а кожне питання може мати декілька варіантів відповідей. Поле `is_correct` у таблиці **QUIZ_OPTIONS** використовується для автоматичної перевірки правильності відповіді користувача [9; 10].

Для відстеження прогресу використовуються таблиці **LESSON_PROGRESS** та **QUIZ_ATTEMPTS**. Таблиця **LESSON_PROGRESS** пов'язує користувача з уроком і фіксує факт завершення навчального матеріалу. Таблиця **QUIZ_ATTEMPTS** пов'язує користувача з уроком, до якого належить тест, і зберігає результат проходження тестування. На основі цих даних система може визначати, які уроки вже пройдені, які тести виконані та який рівень засвоєння матеріалу має студент [9; 10].

Проектована структура бази даних відповідає задачам web-застосунку SkillHub і дозволяє реалізувати каталог курсів, ієрархію модулів і уроків, тестування, збереження результатів і відстеження прогресу користувача. Така модель є достатньо гнучкою для подальшого розширення системи, зокрема додавання сертифікатів, рейтингу курсів, коментарів, обраних курсів або персоналізованих рекомендацій [21; 22].

5. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ WEB-ЗАСТОСУНКУ SKILLHUB

5.1 Реалізація серверної частини

Серверна частина web-застосунку SkillHub реалізована з використанням мови програмування Java та фреймворку Spring Boot. Вона відповідає за обробку запитів від клієнтської частини, виконання бізнес-логіки, роботу з користувачами, курсами, уроками, тестуванням, прогресом навчання та взаємодію з базою даних PostgreSQL [7; 8; 9].

Основною задачею серверної частини є забезпечення стабільної роботи REST API, через яке клієнтський інтерфейс отримує та передає дані. Клієнтська частина надсилає HTTP-запити до backend-рівня, після чого відповідний контролер приймає запит, передає його до сервісу, а сервіс виконує необхідну логіку та звертається до репозиторію для роботи з базою даних [8].

Серверна частина має багаторівневу структуру, яка включає такі основні рівні:

- рівень контролерів;
- рівень сервісів;
- рівень репозиторіїв;
- рівень моделей даних;
- рівень взаємодії з базою даних.

Рівень контролерів забезпечує приймання запитів від клієнтської частини. У web-застосунку SkillHub використовуються такі основні контролери: **UserController**, **AdminController**, **CourseController**, **AuthorController**, **SkillPathController** та **DashboardController**.

UserController відповідає за роботу з поточним користувачем. Через нього користувач може отримати інформацію про власний профіль та оновити персональні дані. **AdminController** використовується для адміністративного керування системою, зокрема для перегляду статистики, списку користувачів, зміни ролей, видалення користувачів і курсів. **CourseController** забезпечує роботу

з курсами, отриманням списку курсів, відкриттям конкретного курсу, завершенням уроку та відправленням результатів тестування.

AuthorController використовується для функцій автора курсу. Він дозволяє отримувати список курсів автора, створювати нові курси, оновлювати дані курсу та видаляти курси. **SkillPathController** відповідає за формування індивідуальної траєкторії навчання користувача, а **DashboardController** — за отримання даних для панелі користувача.

Рівень сервісів містить основну бізнес-логіку застосунку. До нього належать **CurrentUserService**, **AdminService**, **CourseService** та **SkillPathService**. Сервіси не відповідають за приймання HTTP-запитів напряму, а виконують обробку даних після того, як запит був отриманий контролером.

CurrentUserService використовується для отримання поточного користувача, формування відповіді з даними профілю та оновлення інформації користувача. **AdminService** реалізує логіку керування користувачами, ролями, курсами та статистикою. **CourseService** забезпечує отримання курсів, створення, оновлення та видалення курсів, завершення уроків і обробку результатів тестування. **SkillPathService** відповідає за визначення наступного уроку, пошук слабких напрямів навчання та розрахунок готовності користувача до подальшого проходження курсу.

Рівень репозиторіїв відповідає за доступ до бази даних. У системі використовуються **UserRepository**, **CourseRepository**, **LessonRepository**, **LessonProgressRepository** та **QuizAttemptRepository**. Репозиторії працюють із відповідними таблицями PostgreSQL і дозволяють виконувати операції створення, читання, оновлення та видалення даних.

Для зберігання інформації використовується база даних PostgreSQL. Вона містить таблиці користувачів, курсів, модулів, уроків, тестових питань, варіантів відповідей, прогресу уроків і спроб проходження тестів. Взаємодія з базою даних реалізується через Spring Data JPA, що дозволяє працювати з даними на рівні об'єктів Java та зменшує кількість ручного SQL-коду [9; 10].

Загальна логіка роботи серверної частини полягає в тому, що клієнтський інтерфейс надсилає запит до певного endpoint. Контролер приймає цей запит і передає його до відповідного сервісу. Сервіс виконує перевірку даних, обробляє бізнес-логіку та звертається до репозиторію. Репозиторій отримує або змінює дані в базі даних, після чого результат повертається через сервіс і контролер назад до клієнтської частини.

Приклад структури взаємодії серверних класів наведено на рисунку 5.1.

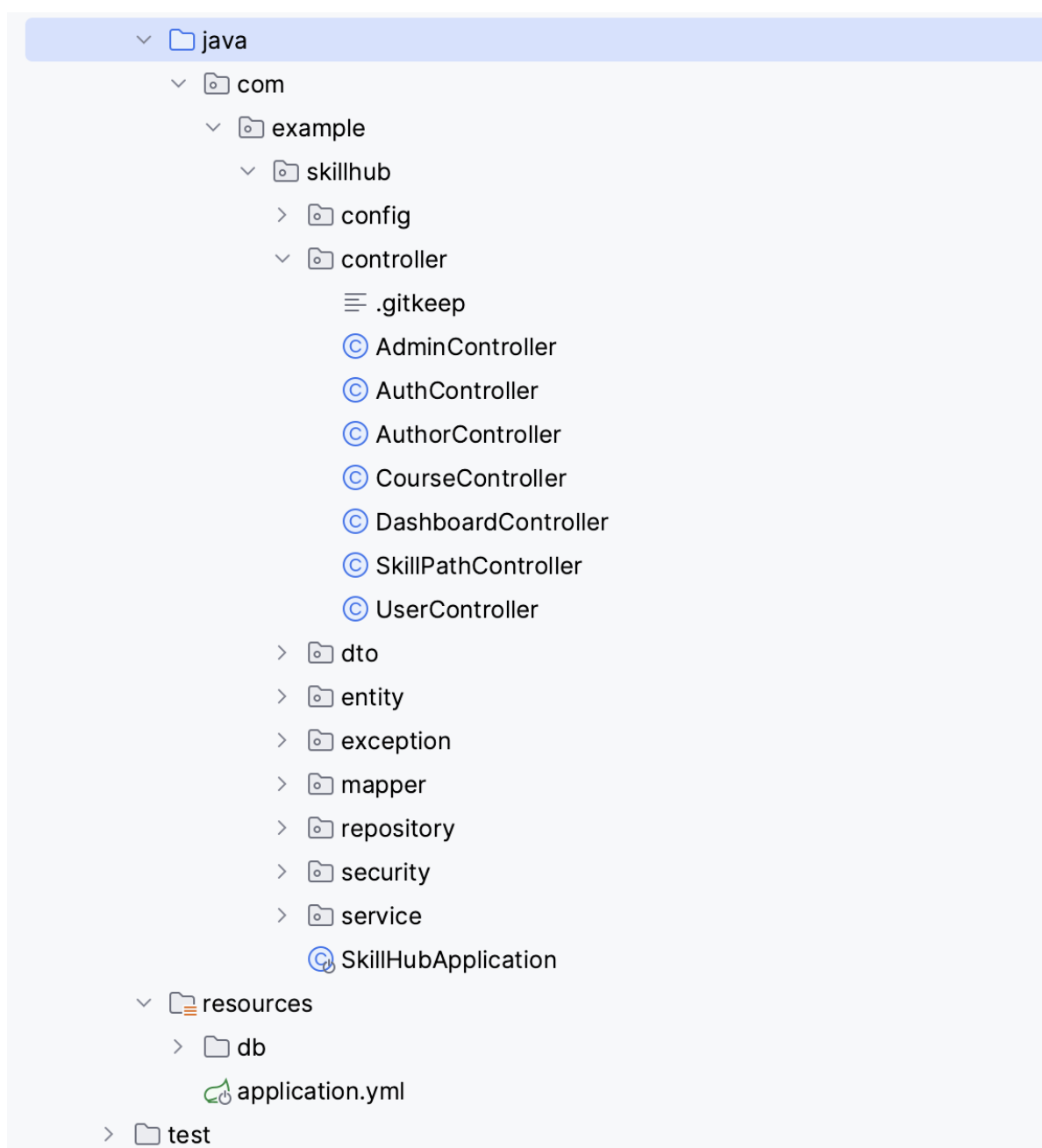


Рис. 5.1 Структура серверної частини web-застосунку SkillHub в середовищі IntelliJ IDEA

Таким чином, серверна частина web-застосунку SkillHub забезпечує основну логіку роботи платформи онлайн-навчання. Використання Spring Boot дозволяє організувати backend-рівень у вигляді зрозумілої модульної структури, де контролери відповідають за приймання запитів, сервіси — за бізнес-логіку, а репозиторії — за роботу з базою даних. Такий підхід спрощує подальше розширення системи та підтримку програмного коду [8; 21; 22].

5.2 Реалізація клієнтської частини

Клієнтська частина web-застосунку SkillHub реалізована з використанням бібліотеки **React** та мови **TypeScript**. Вона відповідає за відображення інтерфейсу користувача, взаємодію зі сторінками платформи, надсилання запитів до серверної частини та отримання даних через REST API. Основна задача клієнтської частини полягає в тому, щоб забезпечити зручну роботу студента, автора курсу та адміністратора з функціями системи [11; 12].

Інтерфейс web-застосунку побудований за компонентним підходом. Це означає, що окремі частини сторінок реалізовані у вигляді незалежних компонентів, які можуть повторно використовуватися в різних частинах системи. Наприклад, окремими компонентами можуть бути картка курсу, форма авторизації, список модулів, блок уроку, тестове питання, панель прогресу та елементи адміністративної панелі [11; 12].

У клієнтській частині передбачено декілька основних сторінок:

- сторінка авторизації та реєстрації користувача;
- головна сторінка платформи;
- каталог IT-курсів;
- сторінка перегляду окремого курсу;
- сторінка уроку;
- сторінка проходження тестування;
- сторінка прогресу користувача;

- панель автора курсу;
- адміністративна панель.

Сторінка авторизації використовується для входу користувача до системи. Після успішної авторизації користувач отримує доступ до функцій відповідно до своєї ролі. Студент може переглядати курси, проходити уроки й тести, автор курсу — створювати та редагувати навчальні матеріали, а адміністратор — керувати користувачами, курсами та статистикою платформи.

Каталог курсів є однією з основних сторінок клієнтської частини. На ньому користувач може переглядати доступні ІТ-курси, їх назву, напрям, рівень складності, короткий опис, тривалість і статус публікації. Після вибору курсу користувач переходить на сторінку детального перегляду, де відображається структура курсу: модулі, уроки та доступні тестові завдання.

Сторінка уроку призначена для відображення навчального матеріалу. Вона може містити основний текст уроку, ключові тези, приклади коду та практичне завдання. Після опрацювання матеріалу користувач може позначити урок як завершений або перейти до тестування, якщо до уроку додані тестові питання.

Сторінка тестування забезпечує проходження контрольних завдань. Користувач переглядає питання, обирає варіанти відповідей, після чого система передає результати на серверну частину. Після обробки відповідей користувач отримує результат тестування, відсоток правильних відповідей і оновлений прогрес навчання.

Окреме значення має сторінка індивідуальної траєкторії навчання. Вона дозволяє користувачу бачити поточний стан проходження курсу, завершені уроки, результати тестування та рекомендації щодо подальшого навчання. Така функція робить платформу більш персоналізованою і допомагає студенту краще орієнтуватися у власному навчальному процесі.

Панель автора курсу використовується для керування навчальним контентом. Автор може створювати нові курси, редагувати їх опис, додавати модулі, уроки, тестові питання та варіанти відповідей. Адміністративна панель має

ширші можливості та дозволяє працювати з користувачами, ролями, курсами й загальною статистикою платформи.

Для взаємодії з серверною частиною клієнтський застосунок використовує HTTP-запити до REST API. Дані між frontend та backend передаються у форматі JSON. Такий підхід дозволяє розділити клієнтську та серверну частини системи, а також спрощує подальше оновлення інтерфейсу без необхідності повністю змінювати серверну логіку [8; 11; 12].

Структура клієнтської частини web-застосунку SkillHub у середовищі розробки наведена на рисунку 5.2.

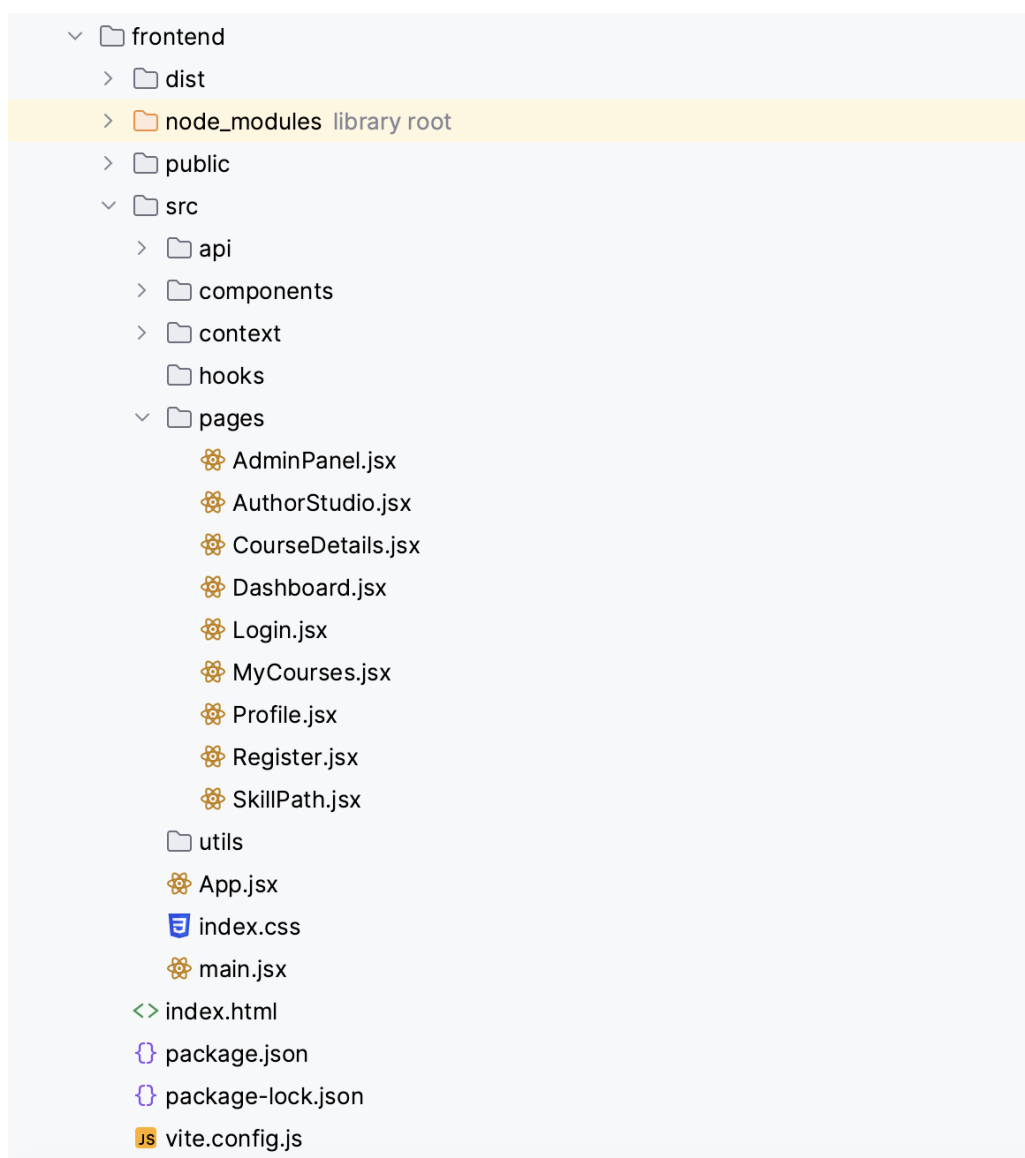


Рис. 5.2 Структура клієнтської частини web-застосунку SkillHub

Клієнтська частина SkillHub забезпечує основну взаємодію користувача з платформою. Використання React дозволяє створити гнучкий компонентний інтерфейс, а TypeScript підвищує надійність коду за рахунок типізації даних. Завдяки цьому інтерфейс системи є зручним для подальшого розширення, підтримки та додавання нових функцій [11; 12].

5.3 Екранні форми web-застосунку SkillHub

Екранні форми web-застосунку SkillHub відображають основні можливості системи та демонструють взаємодію користувача з платформою онлайн-навчання. Інтерфейс застосунку орієнтований на зручний доступ до курсів, навчальних матеріалів, тестування, прогресу користувача та функцій адміністрування.

Основною вимогою до інтерфейсу є простота використання. Користувач повинен мати можливість швидко перейти до потрібного курсу, переглянути структуру навчального матеріалу, виконати тестові завдання та побачити власні результати. Для автора курсу та адміністратора важливою є наявність зрозумілих форм для створення й редагування навчального контенту.

Першою екранною формою є сторінка авторизації користувача. Вона призначена для входу до системи за допомогою електронної пошти та пароля. Після успішної авторизації користувач переходить до основного інтерфейсу платформи відповідно до своєї ролі.

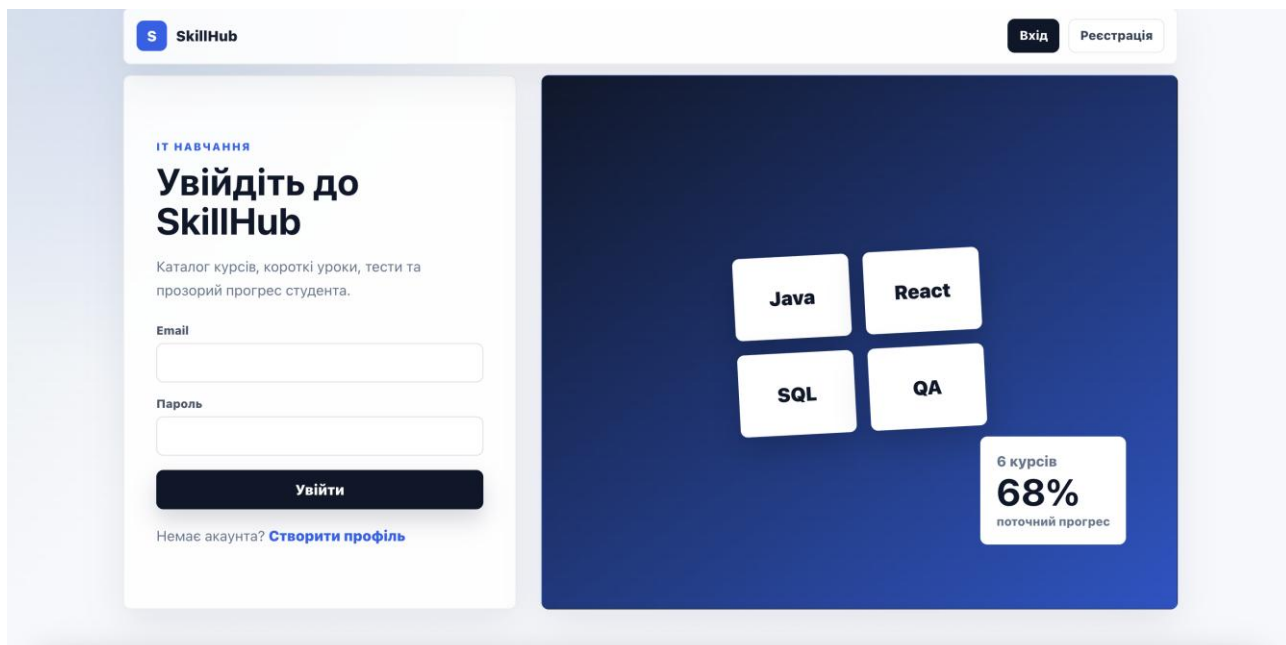


Рис. 5.3 Сторінка авторизації користувача web-застосунку SkillHub

Головна сторінка платформи містить коротку інформацію про систему, доступні навчальні напрями та основні можливості SkillHub. З цієї сторінки користувач може перейти до каталогу курсів, власного профілю або панелі керування, якщо має відповідні права доступу.

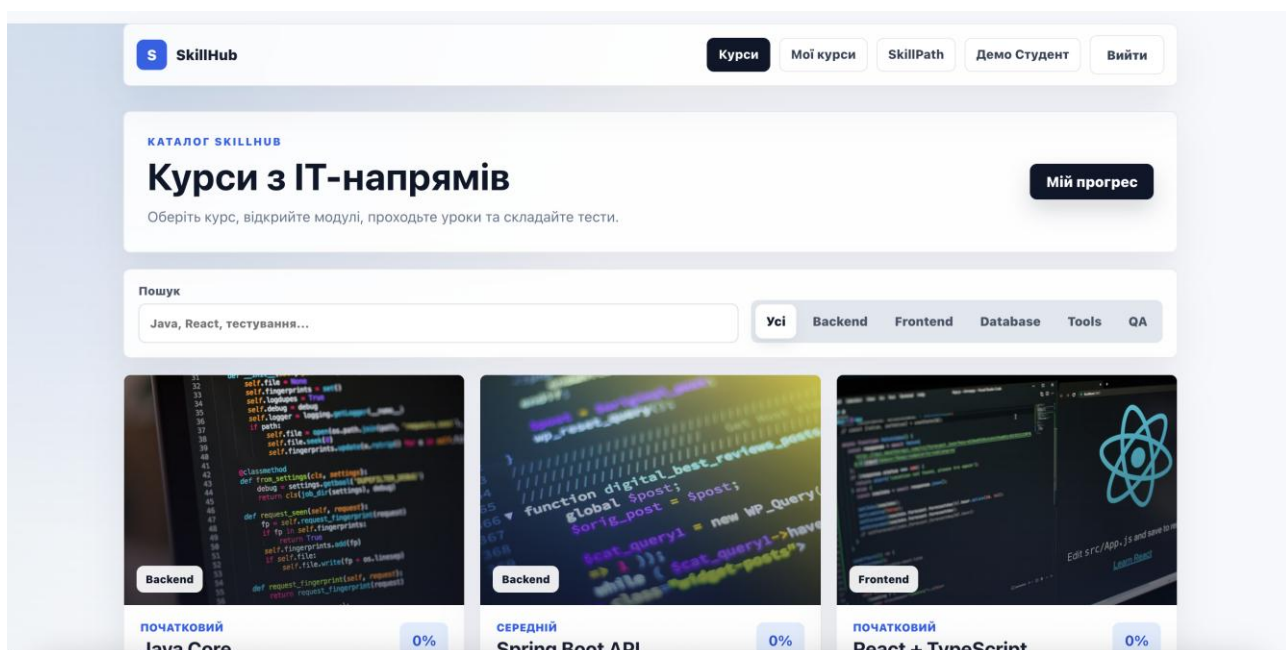


Рис. 5.4 Головна сторінка web-застосунку SkillHub

Каталог курсів є однією з основних сторінок системи. На ньому відображаються доступні ІТ-курси із зазначенням назви, напряму, рівня складності, тривалості та короткого опису. Такий підхід дозволяє користувачу швидко знайти потрібний курс і перейти до його проходження.

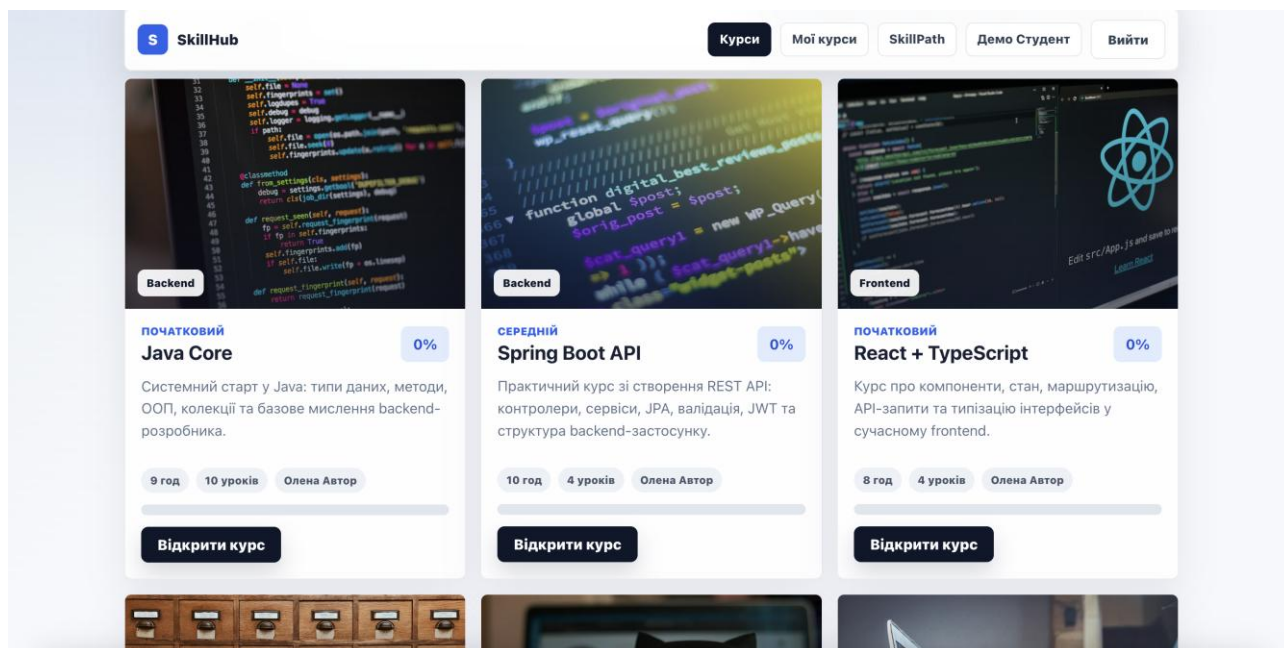


Рис. 5.5 Каталог ІТ-курсів web-застосунку SkillHub

Сторінка перегляду курсу містить детальну інформацію про обраний курс. На ній відображається опис курсу, список модулів, уроків, а також інформація про доступні тестові завдання. Завдяки цьому користувач може бачити загальну структуру навчання та послідовність проходження матеріалу.

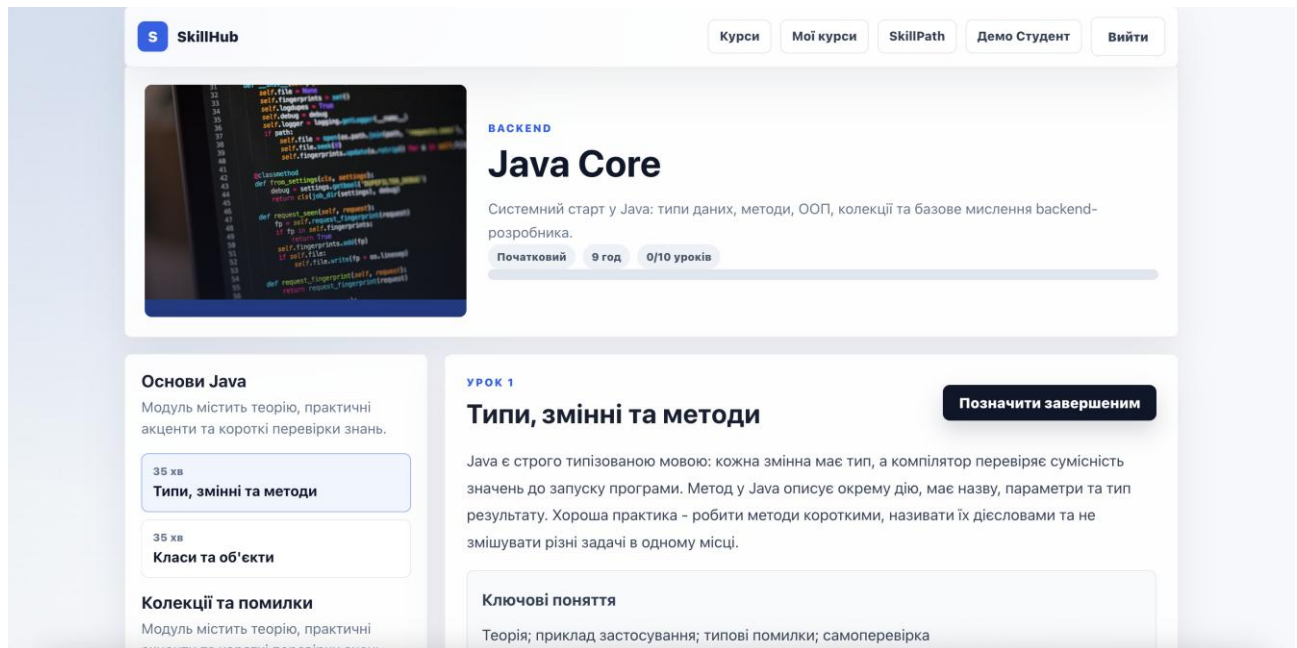


Рис. 5.6 Сторінка перегляду курсу web-застосунку SkillHub

Сторінка уроку призначена для вивчення навчального матеріалу. Вона може містити текст уроку, ключові тези, приклади програмного коду та практичне завдання. Після ознайомлення з матеріалом користувач може перейти до тестування або позначити урок як завершений.

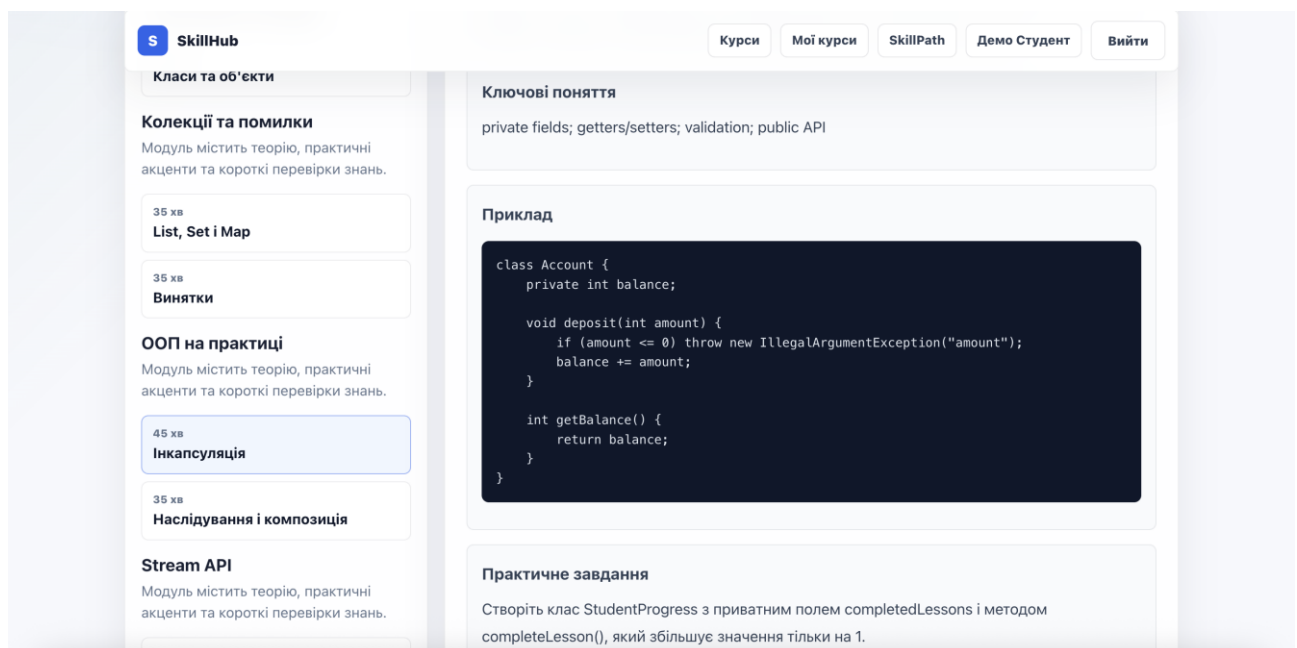


Рис. 5.7 Сторінка проходження уроку web-застосунку SkillHub

Сторінка тестування використовується для перевірки засвоєння матеріалу. Користувач переглядає питання, обирає варіанти відповідей і надсилає результат на перевірку. Після завершення тесту система обчислює кількість правильних відповідей і відображає результат проходження.

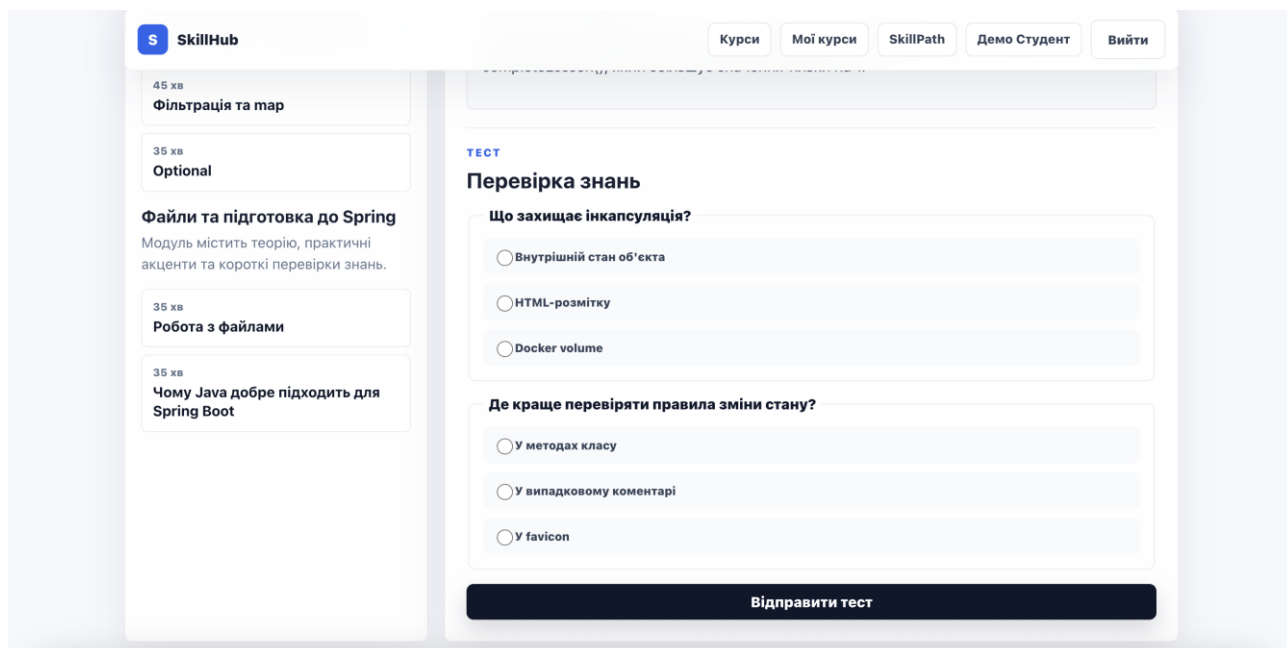


Рис. 5.8 Сторінка проходження тестування web-застосунку SkillHub

Сторінка прогресу користувача дозволяє відстежувати стан проходження навчання. На ній може відображатися кількість завершених уроків, результати тестів, відсоток проходження курсу та рекомендації щодо подальшого навчання. Ця функція є важливою для формування індивідуальної траєкторії користувача.

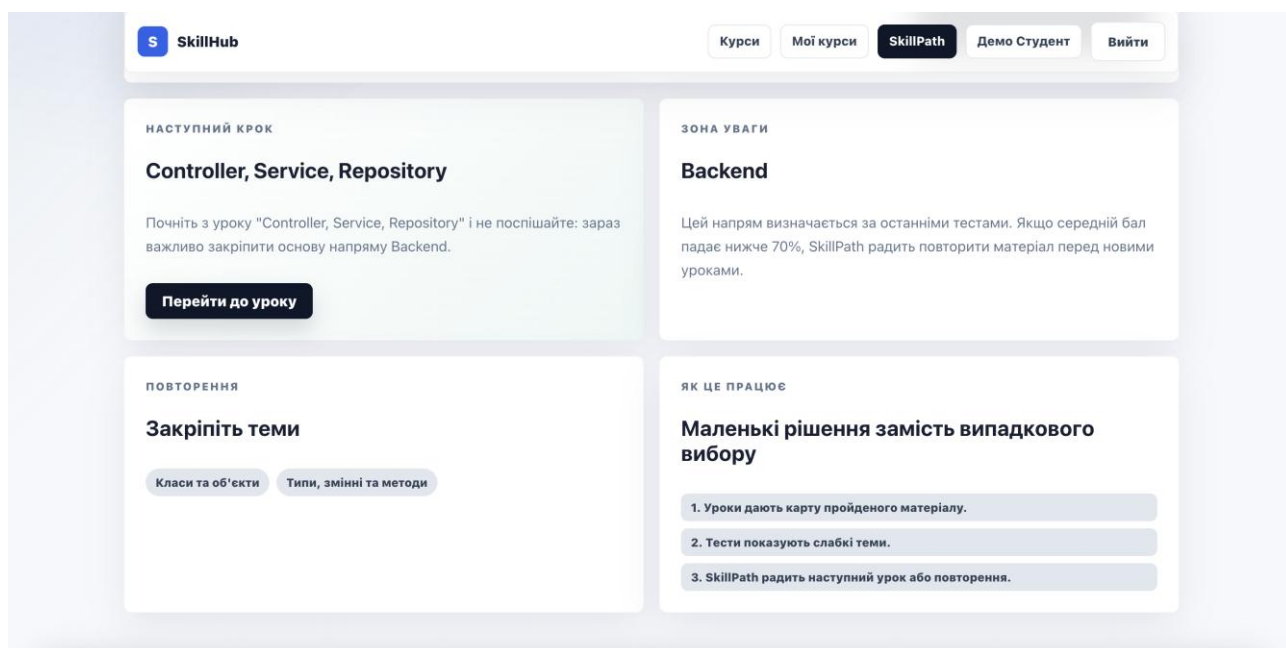


Рис. 5.9 Сторінка прогресу користувача web-застосунку SkillHub

Панель автора курсу використовується для створення та редагування навчальних матеріалів. Автор може додавати нові курси, формувати модулі, створювати уроки, додавати тестові питання та керувати публікацією навчального контенту.

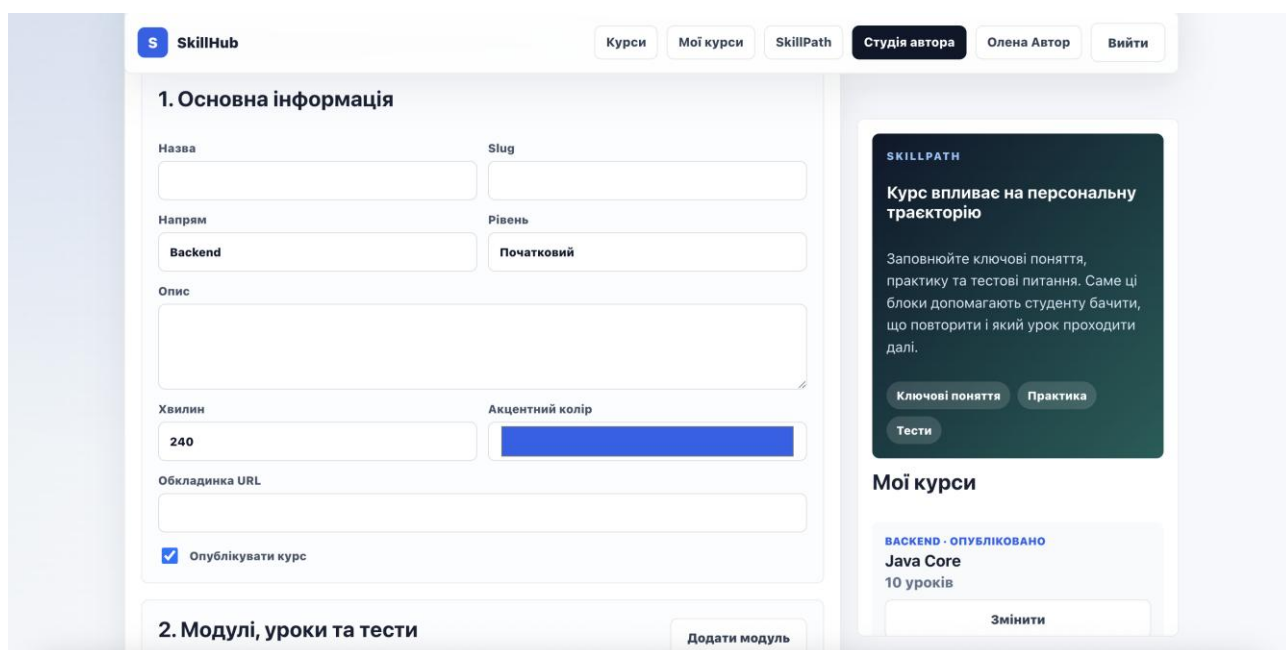


Рис. 5.10 Панель автора курсу web-застосунку SkillHub

Адміністративна панель призначена для керування користувачами, курсами та загальною статистикою платформи. Адміністратор може переглядати список користувачів, змінювати ролі, контролювати опубліковані курси та видаляти некоректні дані.

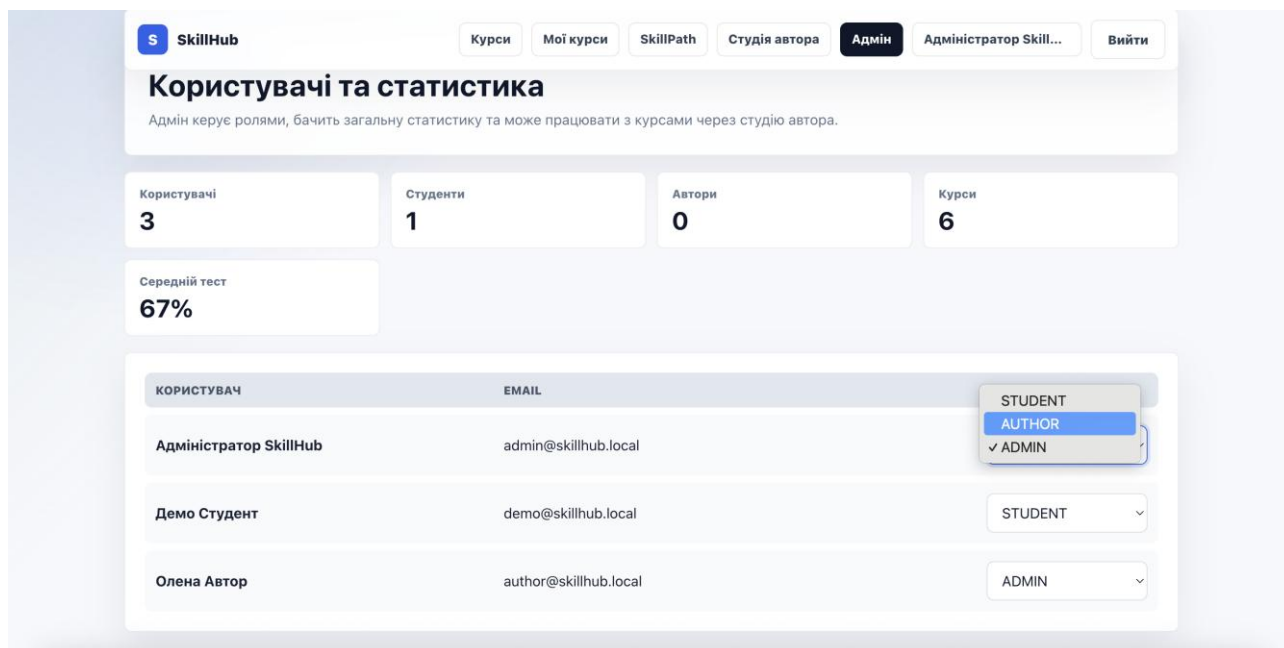


Рис. 5.11 Адміністративна панель web-застосунку SkillHub

Таким чином, екранні форми web-застосунку SkillHub демонструють основні сценарії роботи системи для різних ролей користувачів. Інтерфейс охоплює як навчальну частину для студента, так і функції керування контентом для автора курсу та адміністратора. Це забезпечує повноцінну роботу платформи онлайн-навчання в межах єдиного web-застосунку.

5.4 Тестування програмного забезпечення

Тестування web-застосунку SkillHub виконується для перевірки коректності роботи основних функцій системи, виявлення можливих помилок та підтвердження відповідності програмного продукту визначеним вимогам. Оскільки платформа SkillHub передбачає роботу з різними ролями користувачів,

навчальним контентом, тестуванням і прогресом, перевірка має охоплювати як клієнтську, так і серверну частини застосунку [21; 22].

Основною метою тестування є перевірка працездатності таких функціональних можливостей:

- реєстрація та авторизація користувача;
- розмежування доступу відповідно до ролі користувача;
- перегляд каталогу курсів;
- відкриття сторінки окремого курсу;
- перегляд модулів та уроків;
- проходження тестування;
- збереження результатів тесту;
- оновлення навчального прогресу;
- створення та редагування курсу автором;
- керування користувачами та курсами адміністратором.

Під час тестування перевіряється взаємодія між основними компонентами системи. Користувач виконує дію в інтерфейсі, після чого клієнтська частина надсилає HTTP-запит до серверної частини. Backend-рівень обробляє запит, звертається до бази даних PostgreSQL, отримує або змінює необхідні дані та повертає відповідь клієнтському застосунку. Після цього інтерфейс оновлюється відповідно до отриманого результату [8; 9].

У межах роботи доцільно використати функціональне тестування, тестування інтерфейсу, тестування API та перевірку роботи з базою даних. Функціональне тестування дозволяє перевірити, чи виконуються основні сценарії роботи системи.

Тестування інтерфейсу спрямоване на перевірку коректності відображення сторінок, форм, кнопок і повідомлень. Тестування API дає змогу переконатися, що запити до серверної частини обробляються правильно. Перевірка бази даних потрібна для підтвердження того, що дані про користувачів, курси, уроки, тести та прогрес зберігаються коректно [21; 22].

Процес тестування можна поділити на декілька етапів. Спочатку перевіряється запуск клієнтської та серверної частин застосунку. Далі виконується перевірка авторизації користувача та доступу до функцій відповідно до ролі.

Після цього тестуються основні навчальні сценарії: перегляд курсів, відкриття уроків, проходження тестування та збереження результатів. Окремо перевіряються функції автора курсу й адміністратора, зокрема створення, редагування та видалення навчального контенту.

Під час тестування також враховуються некоректні сценарії. Наприклад, введення неправильного пароля, спроба доступу до адміністративної панелі без відповідної ролі, відправлення порожньої форми, спроба проходження тесту без вибору відповіді або відкриття неіснуючого курсу. Перевірка таких ситуацій дозволяє оцінити стійкість системи до помилкових дій користувача.

Результатом тестування є підтвердження працездатності основних функцій web-застосунку SkillHub. У разі виявлення помилок вони мають бути зафіксовані, після чого виконується корекція програмного коду та повторна перевірка відповідного сценарію. Такий підхід дозволяє підвищити надійність системи та забезпечити її коректну роботу для студентів, авторів курсів і адміністраторів [21; 22].

5.5 Результати тестування

Результати тестування web-застосунку SkillHub оформлено у вигляді тест-кейсів. Для кожного тестового сценарію визначено послідовність дій користувача, очікуваний результат, фактичний результат і статус виконання. Такий підхід дозволяє системно перевірити основні функції програмного продукту та підтвердити його працездатність [21; 22].

Таблиця 5.1

Тест-кейси web-застосунку SkillHub

ID	Сценарій	Дії	Очікуваний результат	Фактичний результат	Статус
ТС-01	Авторизація користувача	Ввести коректний email і пароль, натиснути кнопку входу	Користувач успішно входить до системи	Відповідає очікуванню	Пройдено
ТС-02	Авторизація з некоректними даними	Ввести неправильний пароль і натиснути кнопку входу	Система відображає повідомлення про помилку авторизації	Відповідає очікуванню	Пройдено
ТС-03	Перегляд каталогу курсів	Перейти на сторінку каталогу курсів	Відображається список доступних ІТ-курсів	Відповідає очікуванню	Пройдено
ТС-04	Відкриття сторінки курсу	Обрати курс із каталогу	Відкривається сторінка курсу з описом, модулями та уроками	Відповідає очікуванню	Пройдено
ТС-05	Перегляд уроку	Обрати урок у структурі курсу	Відображається навчальний матеріал уроку	Відповідає очікуванню	Пройдено
ТС-06	Завершення уроку	Натиснути кнопку завершення уроку	Урок позначається як завершений, прогрес оновлюється	Відповідає очікуванню	Пройдено
ТС-07	Проходження тестування	Обрати відповіді на тестові питання та надіслати результат	Система перевіряє відповіді та обчислює результат	Відповідає очікуванню	Пройдено
ТС-08	Збереження результату тестування	Завершити тестування після вибору відповідей	Результат тесту зберігається в системі	Відповідає очікуванню	Пройдено
ТС-09	Перегляд прогресу користувача	Перейти на сторінку прогресу	Відображається інформація про завершені уроки та результати тестів	Відповідає очікуванню	Пройдено
ТС-10	Формування індивідуальної траєкторії	Відкрити сторінку навчального шляху користувача	Система відображає наступні рекомендовані кроки навчання	Відповідає очікуванню	Пройдено

Продовження таблиці 5.1

Тест-кейси web-застосунку SkillHub

ID	Сценарій	Дії	Очікуваний результат	Фактичний результат	Статус
ТС-11	Створення нового курсу автором	У панелі автора натиснути створення курсу, заповнити дані та зберегти	Новий курс створюється та відображається в списку курсів автора	Відповідає очікуванню	Пройдено
ТС-12	Редагування курсу	Змінити назву або опис курсу та зберегти зміни	Дані курсу оновлюються в системі	Відповідає очікуванню	Пройдено
ТС-13	Додавання модуля до курсу	Відкрити панель автора і додати новий модуль	Модуль додається до структури курсу	Відповідає очікуванню	Пройдено
ТС-14	Додавання уроку до модуля	Обрати модуль і створити новий урок	Урок додається до відповідного модуля	Відповідає очікуванню	Пройдено
ТС-15	Додавання тестових питань	Додати питання та варіанти відповідей до уроку	Тестові питання зберігаються і доступні студенту	Відповідає очікуванню	Пройдено
ТС-16	Керування користувачами адміністратором	Відкрити адміністративну панель і переглянути список користувачів	Адміністратор бачить список користувачів системи	Відповідає очікуванню	Пройдено
ТС-17	Зміна ролі користувача	Адміністратор змінює роль користувача	Роль користувача оновлюється	Відповідає очікуванню	Пройдено
ТС-18	Видалення курсу адміністратором	Обрати курс у адміністративній панелі та видалити його	Курс видаляється із системи	Відповідає очікуванню	Пройдено

За результатами тестування встановлено, що основні функціональні можливості web-застосунку SkillHub працюють коректно. Авторизація користувачів, перегляд каталогу курсів, відкриття навчальних матеріалів, проходження тестування, збереження результатів і відображення прогресу виконуються відповідно до очікуваної логіки.

✓ service (com.example.skillhub)	1 sec 351 ms
✓ CurrentUserServiceTest	1 sec 41 ms
✓ toResponse_mapsProfileFields()	999 ms
✓ updateProfile_changesProfileAndReturnsFreshToken()	30 ms
✓ updateProfile_whenEmailBelongsToAnotherUserThrowsConflict()	6 ms
✓ getCurrentUser_whenEmailMissingThrowsNotFound()	3 ms
✓ getCurrentUser_returnsUserFromSecurityContextEmail()	3 ms
✓ SkillPathServiceTest	161 ms
✓ currentPath_recommendsNextUnfinishedLessonAndWeakDirection()	157 ms
✓ currentPath_withoutCoursesReturnsCatalogRecommendation()	4 ms
> ✓ AuthServiceTest	61 ms
> ✓ CourseServiceTest	51 ms
✓ AdminServiceTest	37 ms
✓ updateRole_whenUserMissingThrows()	9 ms
✓ updateRole_changesUserRole()	4 ms
✓ users_mapsEntitiesToResponses()	5 ms
✓ stats_countsUsersCoursesProgressAndAverageScore()	19 ms

Рис. 5.12 Результати тестування в середовищі IntelliJ IDEA

Окремо було перевірено роботу функцій автора курсу та адміністратора. Створення курсів, додавання модулів, уроків і тестових питань, а також керування користувачами виконуються коректно. Перевірка обмеження доступу підтвердила, що користувачі без відповідної ролі не можуть виконувати адміністративні дії.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено web-застосунок **SkillHub**, призначений для організації онлайн-навчання з ІТ-напрямів. Розроблена система дозволяє централізувати навчальний процес, забезпечити доступ до електронних курсів, модулів, уроків, тестових завдань, результатів тестування та прогресу користувача.

Під час виконання роботи було проведено аналіз предметної галузі онлайн-навчання з ІТ-напрямів. Визначено основні проблеми, пов'язані з розподіленням навчальних матеріалів між різними сервісами, складністю контролю прогресу студентів, недостатньою автоматизацією тестування та відсутністю єдиного середовища для проходження навчальних курсів [1; 13; 14].

Було проаналізовано існуючі програмні засоби для організації онлайн-навчання, зокрема **Moodle**, **Google Classroom**, **Coursera** та **Udemy**. У результаті аналізу встановлено, що наявні рішення мають широкий функціонал, однак не завжди повністю відповідають задачі створення простої, спеціалізованої та гнучкої платформи саме для ІТ-навчання. Частина систем є перевантаженою, інші мають обмежені можливості адміністрування або орієнтовані переважно на готовий комерційний контент [3; 4; 5; 6].

У роботі було обґрунтовано вибір засобів реалізації web-застосунку. Для серверної частини використано **Java** та **Spring Boot**, для клієнтської частини — **React** і **TypeScript**, а для зберігання даних — **PostgreSQL**. Такий технологічний стек дозволив реалізувати клієнт-серверну архітектуру, забезпечити взаємодію через REST API та організувати збереження основних даних системи [7; 8; 9; 11; 12].

На етапі проєктування було розроблено логічну архітектуру системи, діаграму прецедентів, діаграму доменної моделі, діаграми діяльності, діаграму класів, діаграму компонентів, діаграму розгортання та ER-діаграму бази даних. Це

дозволило визначити основні модулі системи, ролі користувачів, структуру даних і логіку взаємодії між компонентами web-застосунку [18; 19; 21; 22].

У межах реалізації було описано серверну та клієнтську частини SkillHub. Серверна частина забезпечує роботу з користувачами, курсами, уроками, тестуванням, прогресом та індивідуальною траєкторією навчання. Клієнтська частина забезпечує зручний інтерфейс для студентів, авторів курсів і адміністраторів. Основними функціями застосунку є авторизація користувачів, перегляд каталогу курсів, проходження уроків, виконання тестів, збереження результатів, перегляд прогресу та керування навчальним контентом.

Особливістю розробленої системи є наявність функції індивідуальної траєкторії навчання. Вона дозволяє аналізувати результати проходження уроків і тестів, визначати слабкі напрями користувача та формувати рекомендації щодо подальшого навчання. Це підвищує практичну цінність платформи та відрізняє її від простих систем розміщення навчальних матеріалів.

Проведене тестування підтвердило працездатність основних функцій web-застосунку SkillHub. Було перевірено авторизацію користувачів, перегляд курсів, відкриття уроків, проходження тестування, збереження результатів, відображення прогресу, створення навчального контенту та адміністративне керування системою. За результатами тестування встановлено, що реалізований функціонал відповідає поставленим вимогам.

Отже, мету кваліфікаційної роботи досягнуто. Розроблений web-застосунок SkillHub забезпечує спрощення процесу організації онлайн-навчання з ІТ-напрямів за рахунок централізованого доступу до курсів, навчальних матеріалів, тестування, результатів і прогресу користувача. У подальшому систему можна розширити додаванням сертифікатів, рейтингу курсів, коментарів до уроків, розширеної аналітики, системи рекомендацій та інтеграції з зовнішніми освітніми сервісами.

Робота прийшла апробацію. За її результатами було опубліковано наступні тези доповіді:

1. Казанцев Н.Є., Худік Б.О. Розробка платформи для курсів з штучного інтелекту. Матеріали II Всеукраїнської науково-технічної конференції «Виклики та рішення в програмній інженерії», 26 листопада 2025 року, Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. – К.: ДУІКТ, 2025, С.464-466.
2. Казанцев Н.Є., Компан С.В. Особливості роботи інтерактивної навчальної платформи "SkillHub" для вивчення ІТ-курсів. Матеріали Всеукраїнської науково-технічної конференції «Технології цифрового розвитку: програмні системи та децентралізація», 1-3 травня 2026 року, К.: ДУІКТ.2026 Збірник тез. С.15-17.

ПЕРЕЛІК ПОСИЛАНЬ

1. Benefits and Challenges of E-Learning, Online Education, and Distance Learning [Електронний ресурс] // ResearchGate. – 2024. – Режим доступу: https://www.researchgate.net/publication/379274516_Benefits_and_Challenges_of_E-Learning_Online_Education_and_Distance_Learning
2. Impact of E-learning Platforms on Students' Interest and Academic Achievement in Data Structure Course [Електронний ресурс] // Academia.edu. – Режим доступу: https://www.academia.edu/43969850/Impact_of_E_learning_Platforms_on_Students_Interest_and_Academic_Achievement_in_Data_Structure_Course
3. Moodle LMS Features – Online Learning Platform For Everyone [Електронний ресурс] // Moodle. – Режим доступу: <https://moodle.com/solutions/lms/features/>
4. Google Classroom: Classroom Management Tools & Resources [Електронний ресурс] // Google for Education. – Режим доступу: <https://edu.google.com/workspace-for-education/products/classroom/>
5. Coursera: Courses, Professional Certificates, and Degrees [Електронний ресурс] // Coursera. – Режим доступу: <https://www.coursera.org/>
6. Udemy: Online Courses for Skills, Careers & AI [Електронний ресурс] // Udemy. – Режим доступу: <https://www.udemy.com/>
7. Java Documentation [Електронний ресурс] // Oracle. – Режим доступу: <https://docs.oracle.com/en/java/>
8. Building REST services with Spring [Електронний ресурс] // Spring. – Режим доступу: <https://spring.io/guides/tutorials/rest>
9. PostgreSQL: About [Електронний ресурс] // PostgreSQL Global Development Group. – Режим доступу: <https://www.postgresql.org/about/>
10. PostgreSQL Documentation: Data Types [Електронний ресурс] // PostgreSQL Global Development Group. – Режим доступу: <https://www.postgresql.org/docs/current/datatype.html>
11. React Documentation [Електронний ресурс] // React. – Режим доступу: <https://react.dev/>

12. Using TypeScript with React [Електронний ресурс] // React. – Режим доступу: <https://react.dev/learn/typescript>
13. Литвинова С. Г. Теоретико-методичні засади дистанційного навчання в закладах освіти [Електронний ресурс] / С. Г. Литвинова // Інститут цифровізації освіти НАПН України. – 2022. – Режим доступу: <https://lib.iitta.gov.ua/id/eprint/730426/1/%D0%A1%D0%A2%D0%90%D0%A2%D0%A2%D0%AF-%D0%9B%D0%B8%D1%82%D0%92%D0%9E%D0%94-%D0%94%D0%B8%D1%81%D1%82.pdf>
14. Кухаренко В. М. Система дистанційного навчання університету [Електронний ресурс] / В. М. Кухаренко // Інститут цифровізації освіти НАПН України. – Режим доступу: https://lib.iitta.gov.ua/id/eprint/716736/1/220-233_Kuharenko.pdf
15. Варава І.П. Використання технологій дистанційного навчання в професійній підготовці майбутніх техніків-програмістів [Електронний ресурс] / І. П. Варава // Інститут професійної освіти НАПН України. – 2018. – Режим доступу: <https://lib.iitta.gov.ua/id/eprint/715449/1/%D0%92%D0%B8%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D0%BD%D1%8F%20%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D0%B9%20%D0%B4%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D1%86%D1%96%D0%B9%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F.pdf>
16. Дистанційні та онлайн-сервіси в освіті [Електронний ресурс] // Міністерство освіти і науки України. – Режим доступу: <https://mon.gov.ua/distsiyni-ta-onlayn-servisi-v-osviti>
17. Цифрова платформа «Освіта для життя»: новий інструмент для вчителів [Електронний ресурс] // Міністерство освіти і науки України. – 2025. – Режим доступу: <https://mon.gov.ua/news/tsyfrova-platforma-osvita-dlia-zhyttia-novyi-instrument-dlia-vchyteliv>
18. Аналіз вимог та розробка UML-діаграм концептуального рівня проектування програмної системи [Електронний ресурс] // Електронний архів КПП ім. Ігоря

Сікорського. – Режим доступу:
https://ela.kpi.ua/bitstream/123456789/57036/1/stapvp_srs_uml.pdf

- 19.Марченко О. І. Компоненти програмної інженерії [Електронний ресурс] / О. І. Марченко // Електронний архів КПІ ім. Ігоря Сікорського. – 2022. – Режим доступу: <https://ela.kpi.ua/bitstreams/acfa5e81-b5ed-4307-9fab-59cadd80c0f2/download>
- 20.Алхімова С. М. Об'єктно-орієнтоване програмування. Частина 2 [Електронний ресурс] / С. М. Алхімова // Електронний архів КПІ ім. Ігоря Сікорського. – Режим доступу: https://ela.kpi.ua/bitstream/123456789/53747/1/OOP_Alkhimova_KR_pidruchnyk.pdf
- 21.Інженерія програмного забезпечення: навчальний посібник [Електронний ресурс] / В. І. Шинкаренко, О. В. Горбова, О. П. Іванов, В. О. Андрющенко, В. Я. Нечай. – Дніпро, 2019. – 140 с. – Режим доступу: https://duikt.edu.ua/uploads/p_2642_90357891.pdf
- 22.Основи інженерії програмного забезпечення [Електронний ресурс] // Державний університет інформаційно-комунікаційних технологій. – Режим доступу: https://duikt.edu.ua/uploads/p_163_89259109.pdf
- 23.Docker Documentation [Електронний ресурс] // Docker. – Режим доступу: <https://docs.docker.com/>
- 24.Git Documentation [Електронний ресурс] // Git. – Режим доступу: <https://git-scm.com/doc>
- 25.GitHub Docs [Електронний ресурс] // GitHub. – Режим доступу: <https://docs.github.com/>

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка інтерактивної навчальної платформи "SkillHub" для вивчення ІТ-курсів

Виконав студент 4 курсу
групи ПД-42
Казанцев Нікіта Євгенійович
Керівник роботи

канд. фіз.-мат. наук, доц. Компан Сергій Володимирович

Київ – 2026

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: спрощення процесу організації онлайн-навчання з ІТ-напрямів за рахунок розробки веб-платформи, яка забезпечує централізований доступ до курсів, тестування та перегляду прогресу користувача.

Об'єкт дослідження: процес організації онлайн-навчання з ІТ-напрямів.

Предмет дослідження: засоби, методи та технології автоматизації процесу онлайн-навчання з ІТ-напрямів у веб-застосунку SkillHub.

ЗАДАЧІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1. Провести аналіз предметної галузі онлайн-навчання з IT-напрямів, визначити основні проблеми організації навчального процесу та контролю прогресу користувачів.
2. Здійснити огляд та аналіз існуючих програмних засобів для організації онлайн-навчання, визначити їх переваги та недоліки.
3. Провести огляд засобів для розробки web-застосунку SkillHub.
4. Сформулювати функціональні та нефункціональні вимоги до web-застосунку для організації онлайн-навчання з IT-напрямів.
5. Спроектувати логічну структуру web-застосунку SkillHub та побудувати UML-діаграми для моделювання системи.
6. Розробити серверну та клієнтську частини web-застосунку SkillHub для керування курсами, уроками, тестуванням і прогресом користувачів.
7. Провести тестування web-застосунку SkillHub та проаналізувати результати перевірки основних функцій системи.

АНАЛІЗ АНАЛОГІВ

Показник	Moodle	Google Classroom	Coursera	Udemy	SkillHub
Ролі користувачів	+	частково	+	+	+
Створення власних курсів	+	частково	обмежено	+	+
Додавання уроків	+	+	обмежено	+	+
Тестування	+	частково	+	+	+
Відстеження прогресу	+	частково	+	+	+
Статистика навчання	+	частково	+	частково	+
Індивідуальна траєкторія навчання	-	-	-	-	+
Визначення тем, які потребують повторення	-	-	-	-	+
Адміністрування навчального контенту	+	частково	обмежено	+	+
Зручність для початківця	середня	висока	висока	висока	висока
Гнучкість налаштування	висока	середня	низька	середня	висока
Орієнтація на IT-курси	частково	частково	+	+	+
Основні переваги	Повноцінна LMS, багато функцій, звіти, оцінювання	Простота, інтеграція з Google-сервісами	Якісна структура з курсів, сертифікати	Великий вибір курсів, можливість авторам публікувати матеріали	Проста структура, тестування, прогрес, індивідуальна траєкторія навчання

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функціональні вимоги

1. Реєстрація та авторизація користувачів.
2. Розмежування ролей: студент, автор курсу, адміністратор.
3. Перегляд каталогу ІТ-курсів.
4. Перегляд структури курсу: модулі, уроки, тести.
5. Доступ до навчальних матеріалів.
6. Проходження тестування.
7. Автоматичне збереження результатів тестів.
8. Відображення прогресу студента.
9. Створення та редагування курсів, модулів і уроків.
10. Перегляд статистики навчання.

Нефункціональні вимоги

1. Перехід до курсу, уроку або тесту — не більше 3 кліків.
2. Завантаження основних сторінок — до 3 секунд.
3. Обробка стандартних API-запитів — до 1 секунди.
4. Захист даних: хешування паролів і доступ за ролями.
5. Збереження курсів, тестів і прогресу в PostgreSQL.
6. Підтримка сучасних браузерів: Chrome, Firefox, Edge, Safari.
7. Адаптивність інтерфейсу для екранів від 360 px до 1920 px.

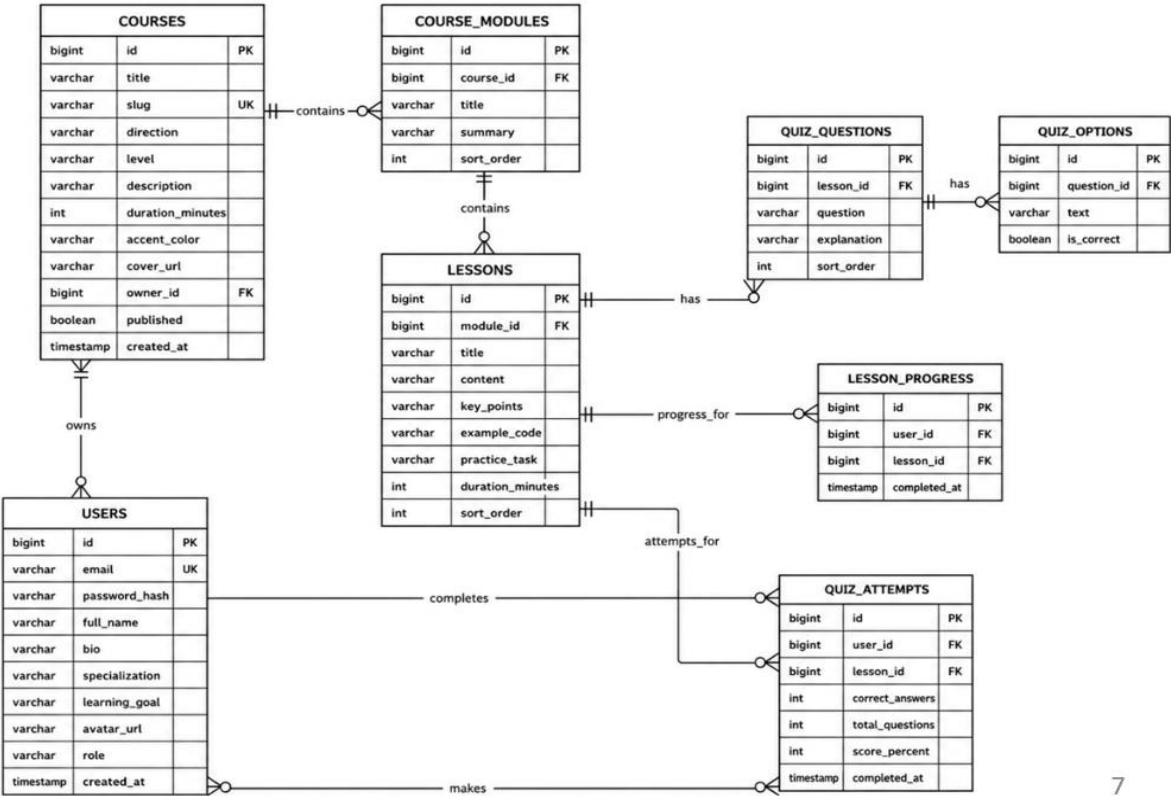
5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



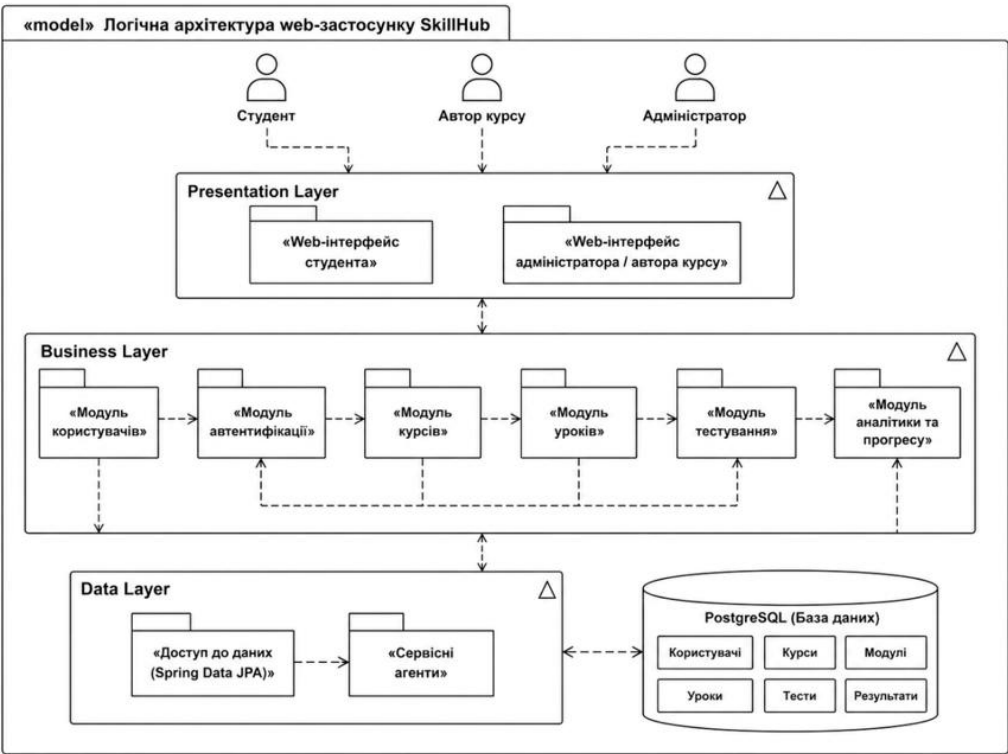
6

СХЕМА БАЗИ ДАНИХ



7

ЛОГІЧНА АРХІТЕКТУРА SkillHub



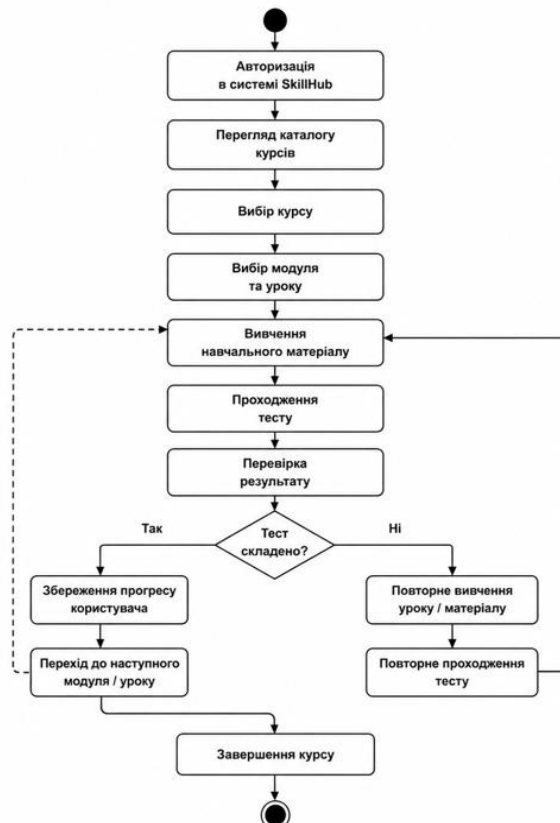
8

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



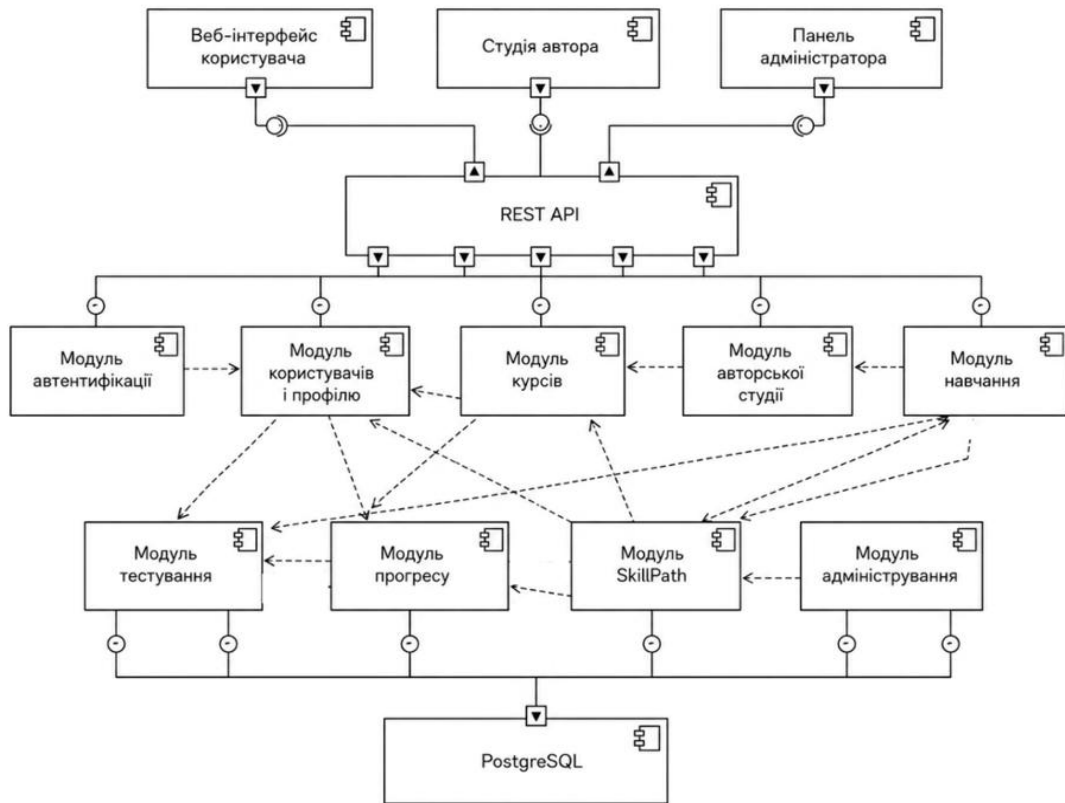
9

ДІАГРАМА ДІЯЛЬНОСТІ ПРОХОДЖЕННЯ КУРСУ



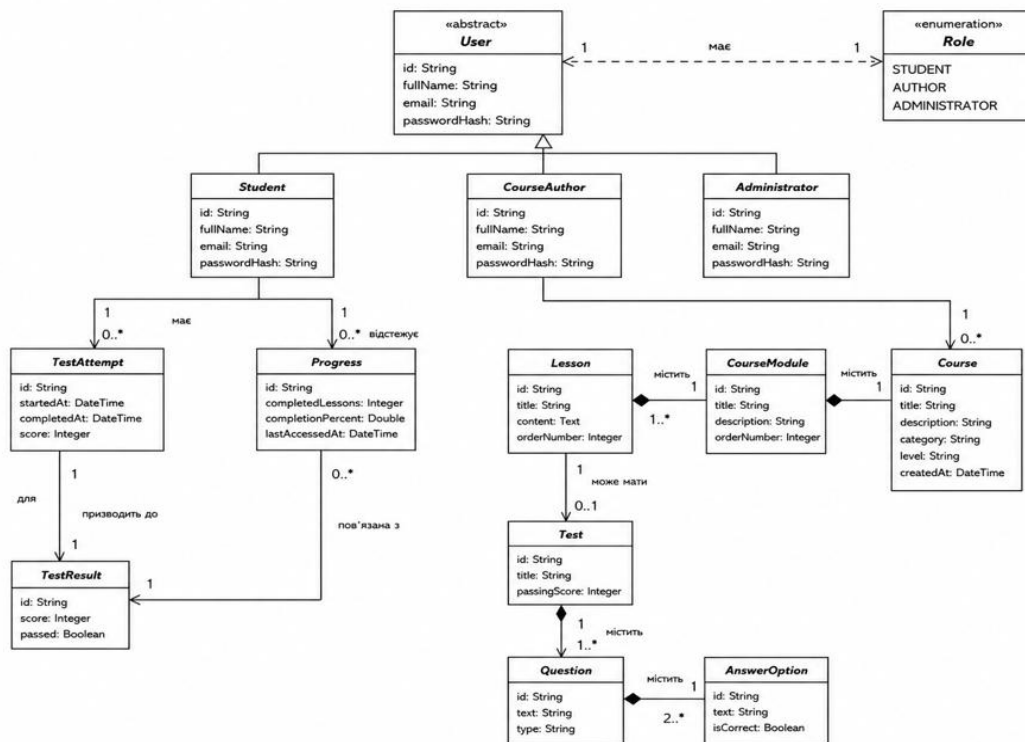
10

ДІАГРАМА КОМПОНЕНТІВ



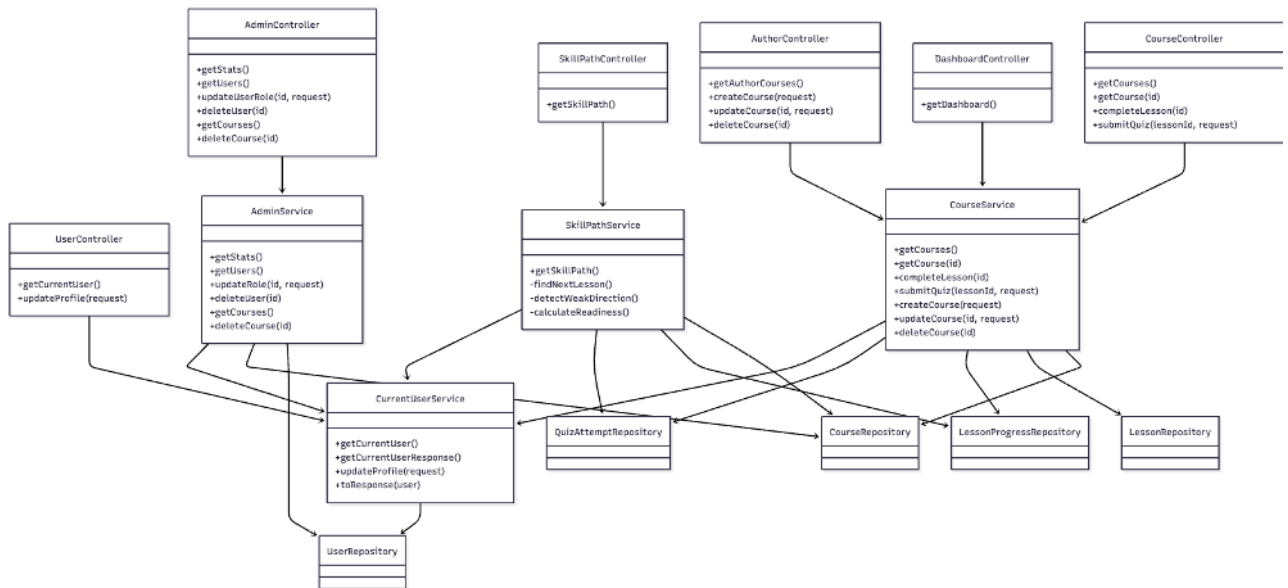
11

ДІАГРАМА ПРЕДМЕТНОЇ ГАЛУЗІ



12

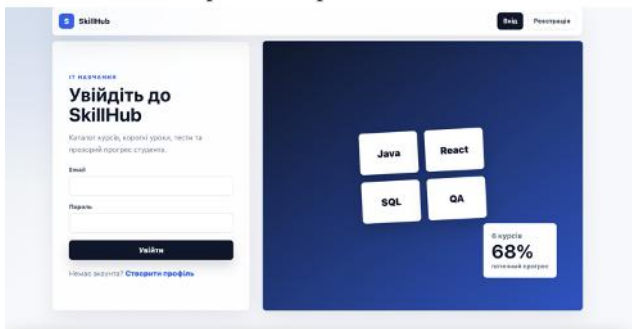
ДІАГРАМА КЛАСІВ



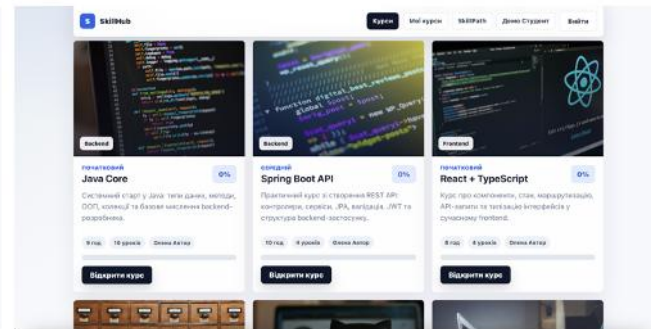
12

ЕКРАННІ ФОРМИ

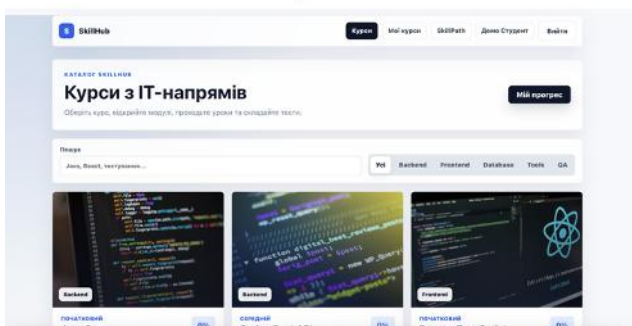
Сторінка авторизації



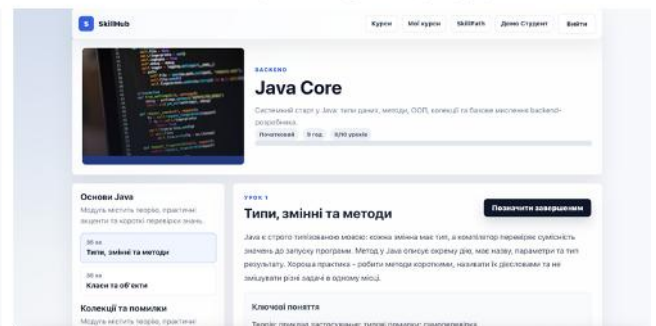
Каталог IT-курсів



Головна сторінка



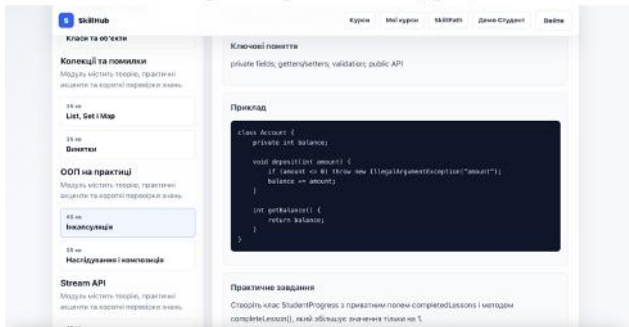
Сторінка перегляду курсу



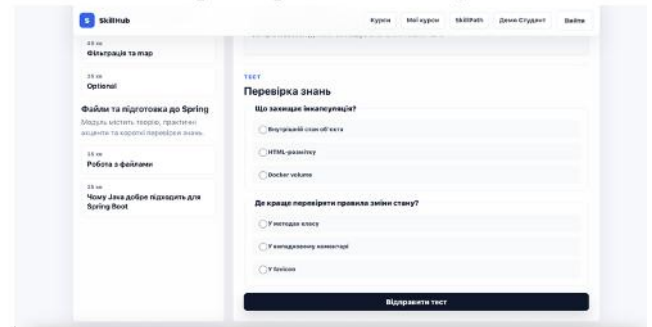
13

ЕКРАННІ ФОРМИ

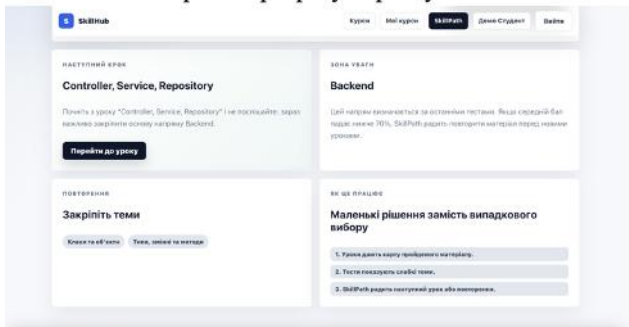
Сторінка проходження уроку



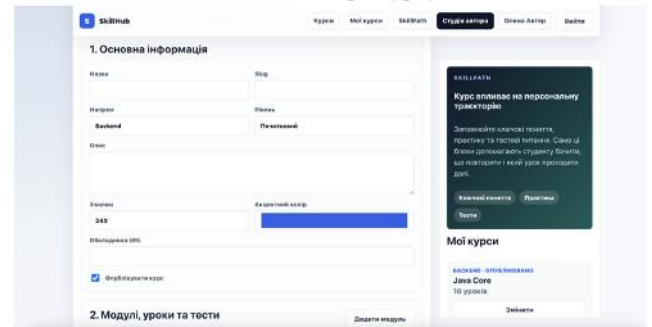
Сторінка проходження тестування



Сторінка прогресу користувача

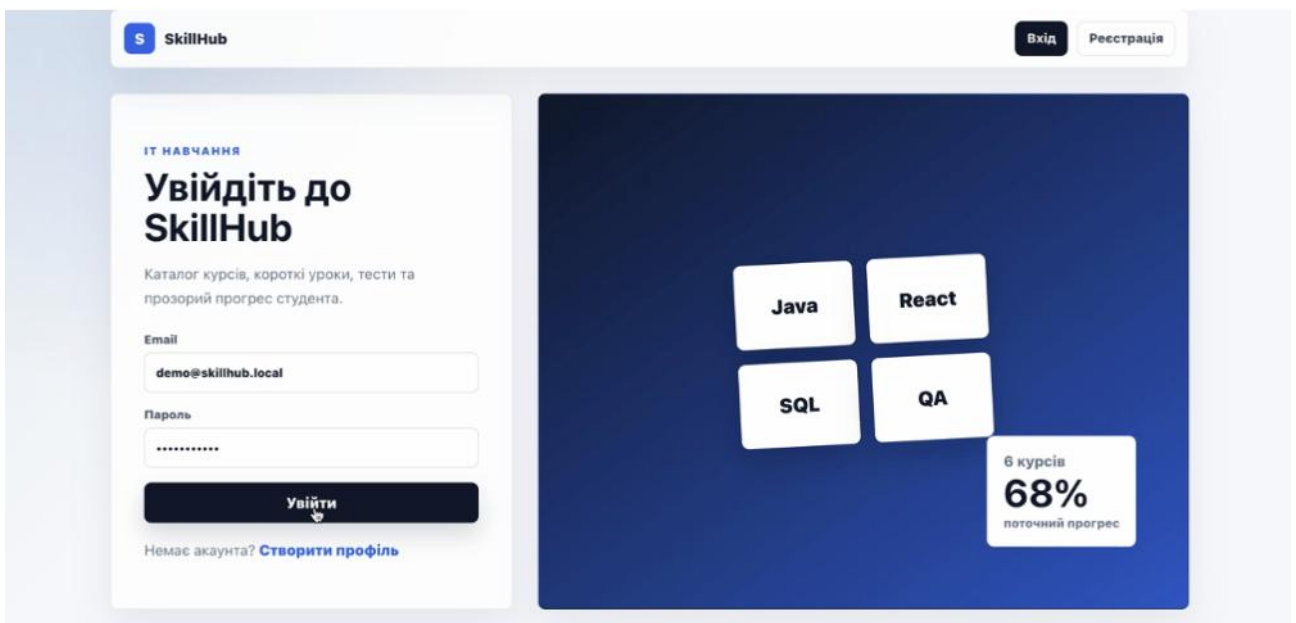


Панель автора курсу



14

ДЕМОНСТРАЦІЯ РОБОТИ ПРОГРАМИ



15

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Тези доповідей:

1. Казанцев Н.Є., Хулік Б.О. Розробка платформи для курсів з штучного інтелекту. Матеріали II Всеукраїнської науково-технічної конференції «Виклики та рішення в програмній інженерії», 26 листопада 2025 року, Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. – К.: ДУІКТ, 2025, С.464-466.

2. Казанцев Н.Є., Компан С.В. Особливості роботи інтерактивної навчальної платформи "SkillHub" для вивчення ІТ-курсів. Матеріали Всеукраїнської науково-технічної конференції «Технології цифрового розвитку: програмні системи та децентралізація», 1-3 травня 2026 року, Київ, ДУІКТ. Збірник тез. (подано до друку).

16

ВИСНОВКИ

1. Проведено аналіз предметної галузі онлайн-навчання з ІТ-напрямів. Визначено основні проблеми: розподілення матеріалів між різними сервісами, складність контролю прогресу, недостатня автоматизація тестування та відсутність єдиного навчального середовища.
2. Проаналізовано існуючі програмні засоби: Moodle, Google Classroom, Coursera, Udemu та Stepik. Визначено їх переваги й недоліки та обґрунтовано доцільність створення власної платформи SkillHub.
3. Обрано засоби реалізації web-застосунку: Java, Spring Boot, PostgreSQL, React та TypeScript. На їх основі створено клієнт-серверну систему з взаємодією через REST API.
4. Спроектовано структуру SkillHub, базу даних : прецедентів, доменної моделі, діяльності, класів, компонентів і розгортання.
5. Розроблено та протестовано web-застосунок SkillHub. Перевірено авторизацію, перегляд курсів, проходження уроків, тестування, збереження результатів і відображення прогресу. Основні функції системи працюють коректно.

18

ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

Лістинг 1. Сервіс роботи з курсами

```

@Service
@RequiredArgsConstructor
public class CourseService {

    private final CourseRepository courseRepository;
    private final LessonRepository lessonRepository;
    private final LessonProgressRepository
progressRepository;
    private final QuizAttemptRepository
attemptRepository;
    private final CurrentUserService
currentUserService;

    @Transactional(readOnly = true)
    public List<CourseSummaryResponse>
listCourses(String direction, String query) {
        User user = currentUserService.getCurrentUser();

        return courseRepository.findAll().stream()
            .filter(course -> course.isPublished() ||
canManage(user, course))
            .filter(course -> direction == null ||
direction.isBlank()
            ||
course.getDirection().equalsIgnoreCase(direction))
            .filter(course -> query == null ||
query.isBlank()
            ||
course.getTitle().toLowerCase().contains(query.toLow
erCase()))
            .map(course -> toSummary(course, user))
            .toList();
    }

    @Transactional
    public LessonResponse completeLesson(Long
lessonId) {
        User user = currentUserService.getCurrentUser();
        Lesson lesson =
lessonRepository.findById(lessonId)
            .orElseThrow(() -> new
ResourceNotFoundException("Урок не знайдено"));

        if
(!progressRepository.existsByUserAndLesson(user,
lesson)) {

progressRepository.save(LessonProgress.builder()
    .user(user)
    .lesson(lesson)
    .build());
        }

        return toLesson(lesson, true);
    }
}

```

```

@Transactional
public QuizResultResponse submitQuiz(Long
lessonId, QuizSubmissionRequest request) {
    User user = currentUserService.getCurrentUser();
    Lesson lesson =
lessonRepository.findById(lessonId)
        .orElseThrow(() -> new
ResourceNotFoundException("Урок не знайдено"));

    Map<Long, Long> answers =
request.getAnswers().stream()
        .collect(Collectors.toMap(
            QuizAnswerRequest::getQuestionId,
            QuizAnswerRequest::getOptionId
        ));

    int total = lesson.getQuestions().size();
    int correct = 0;

    for (QuizQuestion question :
lesson.getQuestions()) {
        Long selectedOptionId =
answers.get(question.getId());
        boolean answerCorrect =
question.getOptions().stream()
            .anyMatch(option ->
option.getId().equals(selectedOptionId)
            && option.isCorrect());

        if (answerCorrect) {
            correct++;
        }
    }

    int score = percent(correct, total);

    QuizAttempt attempt =
attemptRepository.save(QuizAttempt.builder()
        .user(user)
        .lesson(lesson)
        .correctAnswers(correct)
        .totalQuestions(total)
        .scorePercent(score)
        .build());

    boolean completed = score >= 60;
    if (completed &&
!progressRepository.existsByUserAndLesson(user,
lesson)) {

progressRepository.save(LessonProgress.builder()
    .user(user)
    .lesson(lesson)
    .build());
    }
}

```



```

        return new QuizResultResponse(attempt.getId(),
correct, total, score, completed);
    }
}

```

Лістинг 2. Сервіс персональної траєкторії SkillPath

```

@Service
@RequiredArgsConstructor
public class SkillPathService {

    private final CourseRepository courseRepository;
    private final LessonProgressRepository
progressRepository;
    private final QuizAttemptRepository
attemptRepository;
    private final CurrentUserService
currentUserService;

    @Transactional(readOnly = true)
    public SkillPathResponse currentPath() {
        User user = currentUserService.getCurrentUser();

        List<Course> courses =
courseRepository.findAll().stream()
        .filter(Course::isPublished)
        .toList();

        Set<Long> completedLessonIds =
courses.stream()
        .flatMap(course ->
lessonIds(course).stream())

.collect(Collectors.collectingAndThen(Collectors.toList(), ids ->

progressRepository.findByUserAndLessonIdIn(user,
ids).stream()
        .map(progress ->
progress.getLesson().getId())
        .collect(Collectors.toSet())));

        Course targetCourse = courses.stream()
        .filter(course -> lessonIds(course).stream()
        .anyMatch(id ->
!completedLessonIds.contains(id)))
        .min(Comparator.comparingInt(course ->
courseProgress(course,
completedLessonIds)))
        .orElse(courses.isEmpty() ? null :
courses.get(0));

        Lesson nextLesson = targetCourse == null ? null :
targetCourse.getModules().stream()
        .flatMap(module ->
module.getLessons().stream())
        .filter(lesson ->
!completedLessonIds.contains(lesson.getId()))
        .findFirst()
        .orElse(null);

        List<QuizAttempt> attempts =

```

```

attemptRepository.findTop8ByUserOrderByComplete
dAtDesc(user);

String weakDirection = attempts.stream()
        .filter(attempt -> attempt.getScorePercent() <
70)
        .map(attempt ->
attempt.getLesson().getModule().getCourse().getDirect
ion())
        .findFirst()
        .orElse(targetCourse == null ? "Java Core" :
targetCourse.getDirection());

List<String> repeatLessons = attempts.stream()
        .filter(attempt -> attempt.getScorePercent() <
70)
        .map(attempt ->
attempt.getLesson().getTitle())
        .distinct()
        .limit(3)
        .toList();

int averageScore = attempts.isEmpty() ? 0 :
Math.round((float) attempts.stream()

.mapToInt(QuizAttempt::getScorePercent)
        .average()
        .orElse(0));

int readiness = Math.min(100,
Math.round(courseProgress(targetCourse,
completedLessonIds) * 0.55f
+ averageScore * 0.45f));

return new SkillPathResponse(
    targetCourse == null ? null :
targetCourse.getId(),
    nextLesson == null ? null :
nextLesson.getId(),
    targetCourse == null ? "Каталог курсів" :
targetCourse.getTitle(),
    nextLesson == null ? "Оберіть новий курс"
: nextLesson.getTitle(),
    weakDirection,
    recommendation(readiness, weakDirection,
nextLesson),
    readiness,
    repeatLessons
);
}
}

```

Лістинг 3. Сервіс профілю користувача

```

@Service
@RequiredArgsConstructor
public class CurrentUserService {

    private final UserRepository userRepository;
    private final JwtTokenProvider tokenProvider;

    public User getCurrentUser() {
        String email =

```

```

SecurityContextHolder.getContext()
    .getAuthentication()
    .getName();

    return userRepository.findByEmail(email)
        .orElseThrow(() ->
            new
ResourceNotFoundException("Користувача не
знайдено"));
}

public ProfileUpdateResponse
updateProfile(UpdateProfileRequest request) {
    User user = getCurrentUser();

    userRepository.findByEmail(request.getEmail())
        .filter(existing ->
!existing.getId().equals(user.getId()))
        .ifPresent(existing -> {
            throw new ConflictException(
                "Email вже використовується: " +
request.getEmail()
            );
        });

    user.setFullName(request.getFullName());
    user.setEmail(request.getEmail());
    user.setBio(request.getBio());

    user.setSpecialization(request.getSpecialization());
    user.setLearningGoal(request.getLearningGoal());
    user.setAvatarUrl(request.getAvatarUrl());

    User saved = userRepository.save(user);

    return new ProfileUpdateResponse(
tokenProvider.generateToken(saved.getEmail()),
        toResponse(saved)
    );
}

public UserResponse toResponse(User user) {
    return new UserResponse(
        user.getId(),
        user.getEmail(),
        user.getFullName(),
        user.getRole().name(),
        user.getBio(),
        user.getSpecialization(),
        user.getLearningGoal(),
        user.getAvatarUrl(),
        user.getCreatedAt()
    );
}
}

```

Лістинг 4. Сервіс адміністратора

```

@Service
@RequiredArgsConstructor
public class AdminService {

```

```

    private final UserRepository userRepository;
    private final CourseRepository courseRepository;
    private final LessonProgressRepository
progressRepository;
    private final QuizAttemptRepository
attemptRepository;
    private final CurrentUserService
currentUserService;

    @Transactional(readOnly = true)
    public List<UserResponse> users() {
        return userRepository.findAll().stream()
            .map(currentUserService::toResponse)
            .toList();
    }

    @Transactional
    public UserResponse updateRole(Long userId,
UpdateRoleRequest request) {
        User user = userRepository.findById(userId)
            .orElseThrow(() ->
                new
IllegalArgumentException("Користувача не
знайдено"));

        user.setRole(request.getRole());

        return currentUserService.toResponse(user);
    }

    @Transactional(readOnly = true)
    public PlatformStatsResponse stats() {
        return new PlatformStatsResponse(
            userRepository.count(),
            userRepository.countByRole(UserRole.STUDENT),
            userRepository.countByRole(UserRole.AUTHOR),
            userRepository.countByRole(UserRole.ADMIN),
            courseRepository.count(),
            progressRepository.count(),
            attemptRepository.count(),
            (int)
Math.round(attemptRepository.averageScore())
        );
    }
}

```