

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра комп'ютерних наук та програмної інженерії

ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ СИСТЕМИ НАВЧАЛЬНОГО
ПРИЗНАЧЕННЯ «БУЛЕВІ ФУНКЦІЇ»

Кваліфікаційна робота (проект)
на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу 441 групи
Спеціальності: 121 Інженерія програмного
забезпечення
Освітньо-професійної програми:
Середня освіта. Інформатика
Баранов В.С.
Керівник: ст. викл. Черненко І.Є.
Співкерівник: Єрмолаєв В.А.
Рецензент: Лякутін В.В., ФОП

ЗМІСТ

ВСТУП.....	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
РОЗДІЛ 1.....	6
АНАЛІЗ ЗАДАЧ СИСТЕМ НАВЧАЛЬНОГО ПРИЗНАЧЕННЯ	6
1.1. Загальний аналіз завдань онлайн-систем навчального призначення.	6
1.2. Аналіз існуючих систем дистанційного навчання	8
1.3. Завдання СНП «Булеві функції»	10
РОЗДІЛ 2.....	12
ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СНП «БУЛЕВІ ФУНКЦІЇ».....	12
2.1. Проєктування бази даних на основі MySQL	12
2.2. Проєктування серверної частини СНП «Булеві функції».....	16
2.3. Проєктування мобільного клієнту на прикладі iOS	19
ВИСНОВКИ	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	23
ДОДАТОК А	26

ВСТУП

Актуальність теми. Дистанційне навчання – це комплекс програмних засобів, що дозволяє проводити навчання у віддаленому режимі. Така технологія набуває все більшого значення у світі, адже за наявності сучасних комп’ютерних технологій це стає не лише зручним та доступним, а й більш безпечним.

Мета дослідження: розроблення програмного засобу для системи навчального призначення за допомогою Node.js для backend-частини та мобільного клієнту на прикладі iOS.

Об’єкт дослідження: онлайн-системи навчального призначення.

Предмет дослідження: методи та технології реалізації систем навчального призначення.

Завдання дослідження:

- Проєктування системи начального призначення за напрямком алгебри логіки;
- Проаналізувати переваги та недоліки подібних систем
- Проаналізувати аспекти навчання з дисципліни алгебри логіки: лекції, практика, тестування
- Розглянути backend-фреймворк Node.js
- Набути умінь розроблення у середовищі Visual Studio Code;
- Проєктування бази даних у середовищі PHPMyAdmin;
- Розроблення клієнту для системи навчального призначення «Булеві функції»

Структура роботи:

Робота складається зі змісту, вступу, двох розділів, висновків, переліку умовних позначень і додатків.

У першому розділі проводиться теоритичний аналіз теми кваліфікаційної роботи. Буде розглянуто переваги та недоліки систем віддаленого навчання, порівняльний аналіз існуючих платформ, що забезпечують навчання. Буде розглянуто основні аспекти дистанційного навчання: лекції, практика, оцінювання.

У другому розділі буде розглянута практична частина роботи. Буде розглянуто процес проектування та програмування навчального призначення за допомогою Node.js як серверної частини та MySQL як база даних. Другий аспект розділу порушить питання реалізації клієнту на прикладі МОС iOS.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

СНП	Система навчального призначення
БД	База даних
СУБД	Система управління базами даних
МОС	Мобільна операційна система
UI	Інтерфейс користувача (User Interface)
API	Програмний інтерфейс (Application Programming Interface)
ПЗ	Програмне забезпечення

РОЗДІЛ 1

АНАЛІЗ ЗАДАЧ СИСТЕМ НАВЧАЛЬНОГО ПРИЗНАЧЕННЯ

1.1. Загальний аналіз завдань онлайн-систем навчального призначення.

Онлайн-системи навчального призначення (СНП) мають низку переваг. Ці переваги стосуються як студентів так і викладачів. Розглянемо найбільші переваги СНП:

- Наявність теоретичного матеріалу у великій кількості у режимі онлайн. Даний аспект дозволяє навчатись студентові або учню у будь-який час та у будь-якому місці;
- миттєве виставлення оцінок у режимі тестування. СНП здебільш реалізують програмний модуль для виконання тестування. Такі тести можуть мати велику кількість типів питань, наприклад: «вибір одного з багатьох», «вибір багато з багатьох»; такі питання можуть бути оціненими миттєво. Це дозволяє студентам та викладачам одразу побачити як вони впорались із завданням. Звільняючись від процесу оцінювання та перевірки, це дозволяє викладачам зосередитись на зворотньому зв'язку з рекомендаціями до своїх студентів;
- СНП можуть реалізовувати системи аналітики, а саме показники динаміки індивідуального зросту рівня знань студента. Варто відзначити, що онлайн-тести можуть вимірювати рівень знань студента з часом, можливо передбачити рівень знань студента у кінці курсу. Всі ці вимірювання статистичних даних забезпечуються функцією аналітики, що реалізована на сучасній платформі СНП. Одже, це дозволяє викладачам відстежувати рівень знань студентів і давати рекомендації;
- Допомога студентам із обмеженими можливостями. СНП дозволяє більш доступно навчання студентів, адже не вимагає візичної пррисутності у класі або аудиторії. Також, спеціальні програми дозволяють збільшити текст,

перетворити текст у голос або навпаки. Це дозволяє зробити процес навчання студентів з вадами зору, слуху більш зручним;

- Економія часу. СНП є більш ефективним методом під час оцінювання рівня знань студентів або учнів з економічної точки зору. Основні затрати направляються на вибір існуючої системи дистанційного навчання або створення нової, і це носить єдиноразовий характер. Затрати часу на проведення онлайн-тесту нижчі ніж проведення очного тестування.

Але, СНП є свої недоліки:

- розроблення якісного ПЗ – тривалий та коштовний процес;
- з одного боку навчання за допомогою СНП є більш економічним з точки зору часу, але СНП, здебільш, не дозволяють виконувати зворотній зв'язок у режимі реального часу;
- Дані, що отримують викладачі у результаті тестування, включають у себе інформацію про певні відсутності знань у питаннях, але не дозволяють встановити їх причину;
- Дистанційні тести не дозволяють перевірити та оцінити високі знання, пов'язані із творчістю;
- Забезпечення справедливості тесту. Треба проводити комплекс заходів із забезпечення конфіденціальності завдань тесту. Небажано використовувати один і той самий тест удвічі – тобто, проводити зміну питань;
- Елемент випадковості. Елемент випадковості заключається у випадковому обрані відповіді на тестове завдання. Це може спотворювати оцінку знань студента.

Одже, якісно розроблене ПЗ для дистанційного тестування однозначно розкриває всі переваги цієї форми навчання.

1.2. Аналіз існуючих систем дистанційного навчання

Станом на 2022 рік, кількість платформ дистанційного тестування більше 1000. Найбільш популярними платформами є: Moodle, iSpring, WebTutor, Teachbase, GetCourse, Memberluxe. Розглянемо більш детально ці платформи.

Moodle - безкоштовна платформа із широкими можливостями персоналізації, що була створена 20 серпня 2002 року. Встановлюється на сервер. Має велику кількість плагінів для розширення функціональності. Також, потребує вмінь веб-програмування та адміністрування. Moodle підходить для організації дистанційного навчання у вишах. Moodle дещо важка у налаштуванні. До особливостей цієї платформи можна віднести:

- налаштування через плагіни. Кількість плагінів більше 1500. Їх можна скачати з інтернету або створити власний плагін;
- Це система з відкритим кодом;
- Можлива інтеграція с іншими сервісами: можна поєднати з такими популярними платформами як WordPress, Zoom;
- Можливо завантаження презентацій, зображень, відео, аудіо та текстові файли;
- Присутній мобільний додаток для Android(Google Play), iOS(App Store).

iSpring – платформа, що орієнтована на корпоративний сектор. Готова до роботи одразу після реєстрації. До її особливостей можна віднести:

- конструктор курсів. За допомогою iSpring Suite можна створювати навчальний матеріал: курси, тести та ін.;
- Необмежений простір. До iSpring Learn можна завантажити необмежену кількість файлів;
- Інтеграція з іншими сервісами. iSpring Learn легко інтегрується з іншими системами клієнту через відкритий API;

- Швидкий запуск платформи. Немає потреби довго налаштовувати платформу. Достатньо зареєструватись, завантажити курси та покликати користувачів.

WebTutor – розроблення компанії WebSoft, за допомогою якої можна створити корпоративну платформу та атестації персоналу. Її особливості:

- налаштування платформи через модулі. WebTutor пропонує модулі – програми із функціональністю. Наприклад, модулі дистанційного навчання. Всього 12 модулів, кожен оплачується окремо;
- Інтеграція з іншими системами. WebTutor можливо інтегрувати із системами 1С, Oracle EBS, SAP HR та ін.;
- Конструктор курсів. У WebTutor є конструктор курсів CourseLab, який отримується окремо. За його допомоги можна створювати початковий контент.

Teachbase – платформа для дистанційної освіти. Можливо організувати як дистанційне навчання, так і продаж курсів. Особливості:

- Інтеграція з іншими системами. Teachbase можливо інтегрувати із сторонніми CRM та платіжними системами
- Платформа для вебінарів. Teachbase дозволяє проводити вебінари без сторонніх сервісів.

GetCourse – сервіс для дистанційного навчання та проведення вебінарів. Її особливості:

- продаж курсів. За допомогою GetCourse можливо створювати тренінги та вебінари;
- Автоматизація бізнес-процесів. GetCourse робить всю рутину роботи: нагадує учасникам про наступний веінар, перевіряє тести та видає учасникам сертифікати після успішного проходження курсу.

Memberlux – плагін для WordPress, що дозволяє продати навчальний контент через сайт: лекції, відеокурси, подкасти. Особливості:

- плагін для WordPress. Memberlux встановлює на сайт з WordPress, дозволяючи продаж курсів на сайті;
- Інтеграція з іншими сервісами. Memberlux можна інтегрувати з платіжними системами і сервісами розсилок;
- Захист матеріалів. В Memberlux працює шифрування відео: його не можна скачати чи побачити код.

1.3. Завдання СНП «Булеві функції»

Як і у більшості СНП, система «Булеві функції» реалізовує найважливіший функціонал СНП. До завдань СНП можна віднести наступні пункти:

- Аутентифікація користувача. Система «Булеві функції» пропонує 2 типи користувачів: студент та викладач. Кожна роль наділена певними правами. А саме, для студентів пропонуються права перегляду теоретичного матеріалу, що може складатись з лекції, аудіовідеозаписів. Можливість входження до групи та складання тестів. Роль викладача наділена правами створення навчальних матеріалів: лекцій, відеозаписів: тощо. Викладач має право перевіряти складання тестів, бачення динаміки та тенденції зростання знань студентів
- Переглядання теоретичних матеріалів. Як студент так і викладач має право переглядати навчальний матеріал, що може бути представлений у різних форматах: WEB-сторінки, PDF файли, відеолекції, або власний формат
- Складання тестових завдань. Одна з найголовніших функцій СНП. Система має пропонувати складання тестів із різними типами завдань (типами питань), а саме: вибір одного з багатьох, вибір багатьох з багатьох,

упорядкування, питання із відкритою відповіддю, питання із відкритою відповіддю.

Нижче зображено Use Case діаграму, що зображує взаємодію користувача із системою (Рисунок 1.1.):

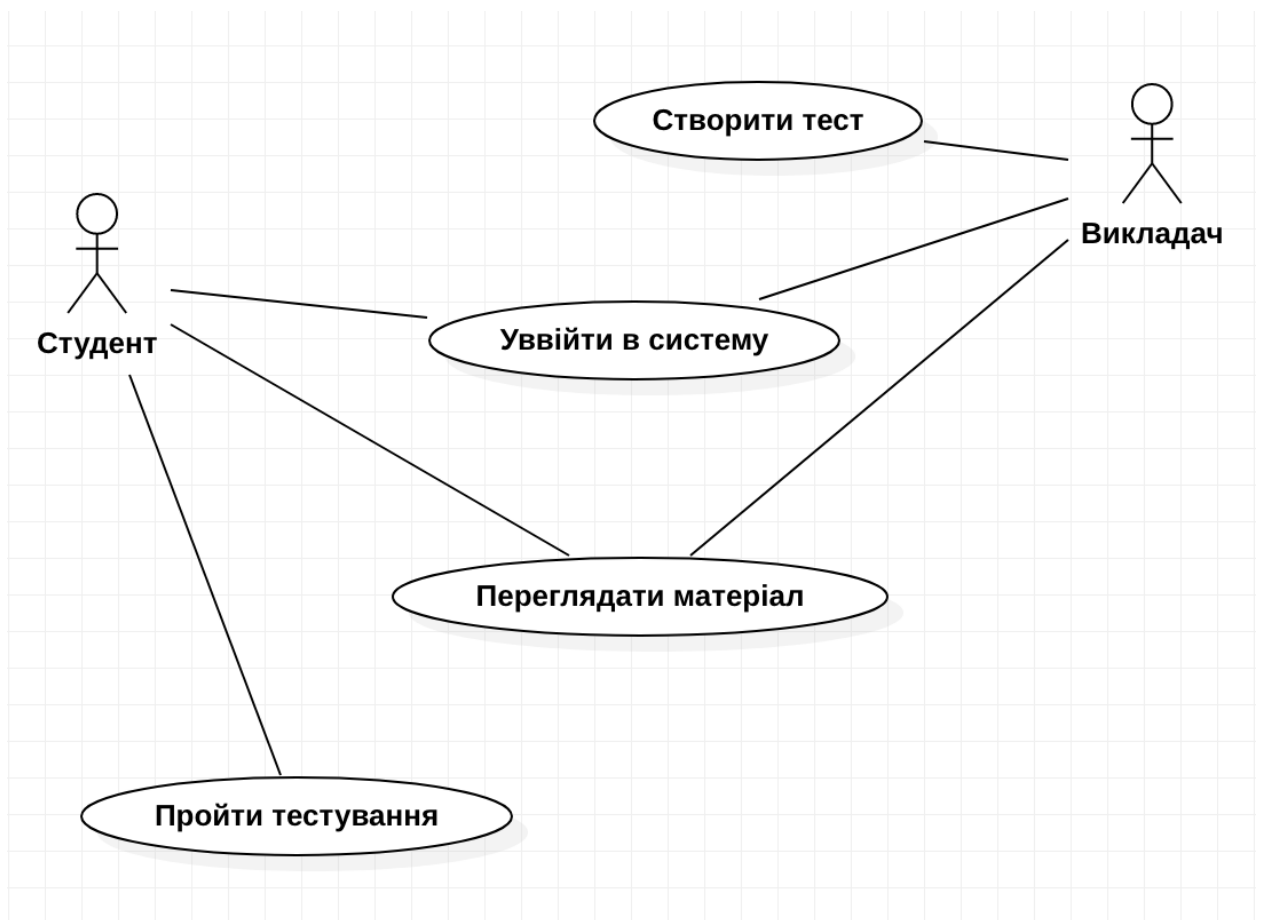


Рисунок 0-1.1 – UML-діаграма взаємодії із системою

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СНП «БУЛЕВІ ФУНКЦІЇ»

2.1. Проєктування бази даних на основі MySQL

Перш ніж переходити до створення серверної частини системи, слід було продумати схему бази даних, та сутності, що відносяться до неї. Під час проєктування БД було виділено наступні сутності: користувач (User), лекція (Lecture), тест (Test), оцінка (TestMark), стан проходження тесту (TestAnswerState), група користувачів (UserGroup). Розглянемо ці сутності більш детально.

User. Дана таблиця представляє список користувачів системи.

Нижче наведено перелік полів:

- Id (ідентифікатор). Для однозначної ідентифікації користувача буде використовуватись ідентифікатор у форматі UUID;
- Username (ім'я користувача). Це поле, що має рядковий тип даних. Ім'я користувача може також використовуватись для однозначної ідентифікації користувача;
- Password (пароль). Це поле, що має рядковий тип даних. Система аутентифікації передбачає використання або електронної пошти та паролю або ім'я користувача та паролю;
- Email (електронна пошта). Електронна пошта використовується під час створення облікового запису користувача. На електронну пошту може приходити деякі листи, що стосуються процесу навчання;
- Firstname (Ім'я). Ім'я використовується як звернення у клієнтській частині;
- Lastname (Прізвище) Прізвище використовується як звернення у клієнтській частині;
- dateCreated (дата створення користувача);

- lastOnline (дата останнього входження у систему);
- role (роль користувача). Система підтримує роботу з двома типами ролі: студент та викладач;
- country (країна користувача).

Нижче зображено схему таблиці User:

Field	Type
id	int(11)
username	varchar(64)
password	varchar(64)
email	varchar(128)
firstname	varchar(64)
lastname	varchar(64)
dateCreated	date
lastOnline	datetime
role	tinyint(4)
country	varchar(64)

Рисунок 2.1 – таблиця User

Lecture. Дана таблиця представляє список лекцій системи.

Нижче наведено перелік полів:

- id (ідентифікатор). Ідентифікатор використовується для однозначної ідентифікації лекції;
- title (назва). Назва лекції довжиною до 255 символів. Назва має повністю описувати призначення даної лекції;
- contentFile (файл). URL для файлу, що містить файл лекції. Це може бути WEB-сторінка (.html), pdf-файл, аудіовідеозапис, JSON-файл;
- createDate (дата створення);
- createdByUserId (ким створено). Ідентифікатор користувача, що створив дану лекцію.

Нижче наведено схему таблиці Lecture:

BooleanLearn Lecture	
id	int(11)
title	varchar(255)
contentFile	varchar(255)
createdDate	date
createdByUserId	int(11)

Рисунок 2.2 – Таблиця Lecture

Test. Дана таблиця представляє список тестів системи.

Нижче наведено перелік полів:

- id (ідентифікатор). Ідентифікатор використовується для однозначної ідентифікації тесту;
- contentFile (файл). URL для файлу, що містить файл тесту. Це JSON-файл, що має певний формат;
- createdDate (дата створення);
- createdByUserId (ким створено). Ідентифікатор користувача, що створив дану лекцію.

Нижче наведено схему таблиці Test:

BooleanLearn Test	
id	int(11)
createdDate	date
contentFile	varchar(255)
createdByUserId	int(11)

Рисунок 2.3 – Таблиця Test

TestAnswerState. Дана таблиця представляє список станів проходження тестів системи користувачем.

Нижче наведено перелік полів:

- id (ідентифікатор). Ідентифікатор використовується для однозначної ідентифікації стану тесту;

- `userId` (ким було пройдено тест). Ідентифікатор користувача, що пройшов даний тест.
- `contentFile` (файл). URL для файлу, що містить файл стану проходження тесту. Це JSON-файл, що має певний формат;
- `answerDate` (дата проходження тесту);

Нижче наведено схему таблиці `TestAnswerState`:

Column Name	Data Type
<code>id</code>	<code>int(11)</code>
<code>userId</code>	<code>int(11)</code>
<code>testId</code>	<code>int(11)</code>
<code>contentFile</code>	<code>varchar(255)</code>
<code>answerDate</code>	<code>date</code>

Рисунок 2.4 – Таблиця `TestAnswerState`

`TestMark`. Дана таблиця представляє список всіх оцінок, що були пройдені користувачами.

Нижче наведено перелік полів:

- `id` (ідентифікатор). Ідентифікатор використовується для однозначної ідентифікації оцінки;
- `answerId` (ідентифікатор стану). Ідентифікатор на стан проходження тесту користувачем;
- `markDate` (дата оцінювання);
- `mark` (оцінка). Це поле представляє собою мале число (`tinyint`);
- `answerDate` (дата проходження тесту).

Нижче наведено схему таблиці `TestMark`:

BooleanLearn TestMark	
id	: int(11)
answerId	: int(11)
markDate	: date
mark	: tinyint(4)
feedback	: varchar(255)

Рисунок 2.6. – Таблиця TestMark

2.2. Проектування серверної частини СНП «Булеві функції»

Для реалізації серверної частини було обрано технологію Node.js. Це платформа з відкритим кодом для програмування високонавантажених мережеских застосунків, написаних мовою JavaScript. Засновником платформи є *Раян Дал* (Ryan Dahl). Якщо раніше Javascript застосовувався для обробки даних в браузері користувача, то *node.js* надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їхнього виконання. Платформа *Node.js* перетворила JavaScript на мову загального використання з великою спільнотою розробників.

Node.js має наступні властивості:

- асинхронна single-threaded модель виконання запитів;
- неблокуючий ввід/вивід;
- система модулів CommonJS;
- рушій JavaScript Google V8.

Нижче наведено список запитів до серверу:

- /user/register. Це POST-запит, що виконує реєстрацію користувача. У якості вхідних даних поступає: Ім'я, Прізвище, Ім'я користувача, пароль, електронна пошта. Під час виконання запиту виконується перевірка на існування користувача, якщо користувача не існує, то створюється запис користувача у БД. Даний запит повертає ідентифікатор користувача, що буде використовуватись для подальших запитів (Рисунок 2.7.);

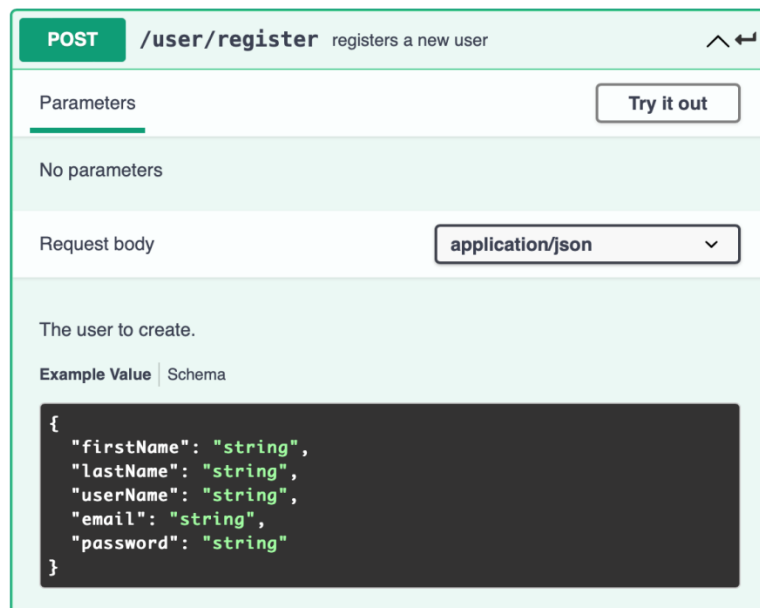


Рисунок 2.7. – Swagger для /user/register

- /user/login. Це POST-запит, що виконує входження у систему. У якості вхідних даних поступає ім'я користувача або електронна пошта та пароль. Під час виконання запиту перевіряється наявність користувача у БД, якщо його не існує, то повертається помилка. Інакше, повертається ідентифікатор користувача, що буде використовуватись для подальших запитів;
- /user/logout. Це PUT-запит, що виконує виходження з системи. Під час виконання запиту у БД оновлюється значення lastOnline для користувача (Рисунок 2.8.);

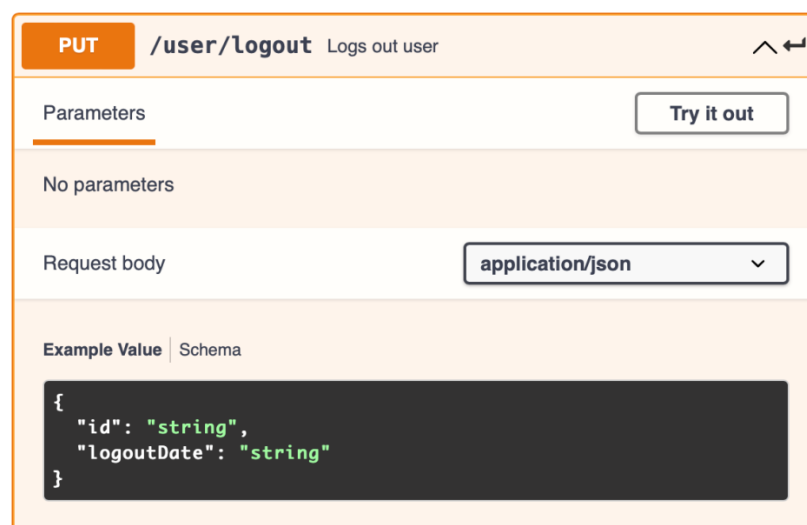


Рисунок 2.8. – Swagger для /user/logout

- /user/deactivate. Це PUT-запит, що деактивує користувача у системі. Користувач «зникає з усіх списків», але запис користувача та всі данні зберігаються у БД;
- /user/delete. Це DELETE-запит, що видаляє користувача із системи. Разом із цим користувач зникає з усіх груп, зникають всі оцінки користувача (Рисунок 2.9.);

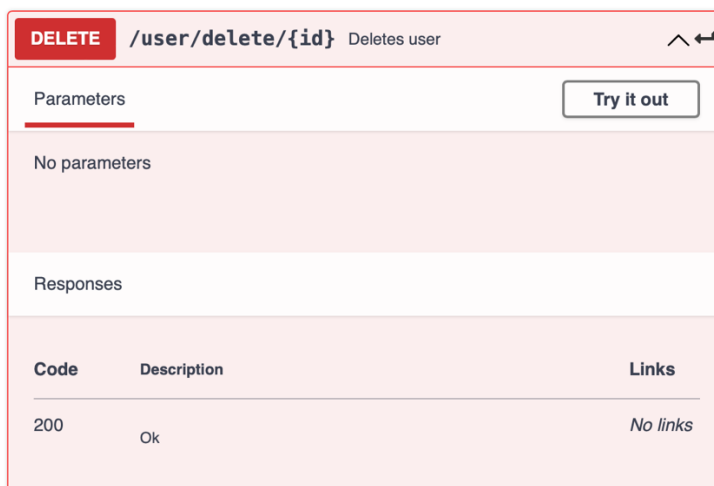


Рисунок 2.9. – Swagger для /user/delete

- /userlist. Це GET-запит, що повертає список користувачів у системі;
- /grouplist. Це GET-запит, що повертає список груп у системі;
- /testlist. Це GET-запит, що повертає список тестів у системі;
- /user/groups. Це GET-запит, що повертає список груп для використання у системі;
- /user/tests. Це GET-запит, що повертає список тестів у системі;
- /test/info. Це GET-запит, що повертає дані для тесту у JSON-форматі;
- /test/confirm. Це POST-запит, що є завершальним етапом, при проходженні тесту. Під час його виконання відбувається перевірка відповідей на правильність, та створення запису для оцінки тестування для поточного користувача (студента);

- /user/mark. Це POST -запит, що виконується після перевірки правильності відповідей;
- /user/marks. Це GET-запит, що повертає список оцінок для користувача (Рисунок 2.10.).

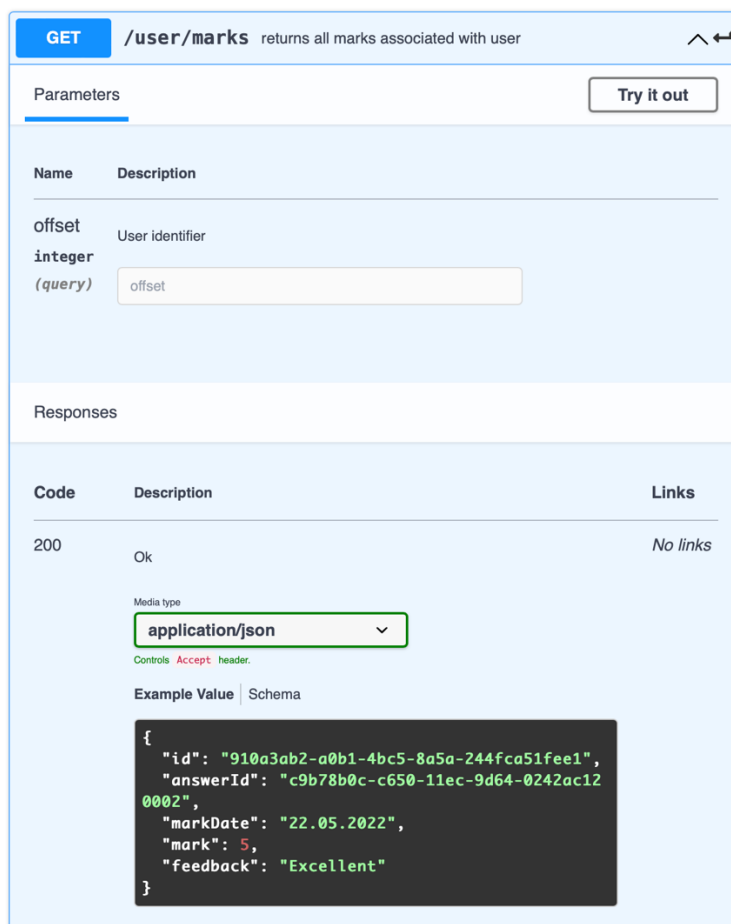


Рисунок 2.10. – Swagger для /user/marks

2.3. Проектування мобільного клієнту на прикладі iOS

Фінальним етапом реалізації СНП «Булеві функції» є створення frontend-частини системи. Одним із прикладом є мобільний додаток. Під час дослідження було запропоновано реалізувати мобільний додаток для МОС iOS за допомогою iOS SDK

Для розроблення додатку Boolean Functions було використано IDE Xcode. Xcode – це інтегроване середовище розроблення (IDE) програмного забезпечення для платформ Apple.

Це IDE включає у себе велику частину документації від Apple та Interface Builder – додаток, що використовується для створення графічних інтерфейсів. Пакет Xcode включає набір компіляторів GNU, та підтримує мови C, C++, Objective-C, Objective-C++, Swift, Java, AppleScript, Python, Ruby.

Для реалізації інтерфейсу було використано фреймворк SwiftUI. SwiftUI – це фреймворк, набір інструментів для створення інтерфейсу користувача, що дозволяє декларативно розробляти додатки.

Під час проєктування та програмування додатку було взято до уваги всі аспекти побудови UI: зручність, інтуїтивність, мінімалізм, увага до основного контенту.

Було реалізовано основні екрани для взаємодії серверної частини та клієнтом:

- Екран login, signup. Це перший екран, що бачить користувач при першому запуску. Користувач має 2 вибори: або увійти до системи або реєстрація або згадати пароль. Якщо користувач обирає login, то з'являються 2 поля: перше поле пропонує ввести ім'я користувача або електронну пошту, а друге поле пропонує ввести пароль. У разі обрання signup, користувач бачить 6 полів: поле ім'я, прізвище, ім'я користувача, пароль, підтвердження паролю, країна. У разі обрання поля «згадати пароль» користувачеві запропонується ввести електронну пошту та натиснути «перезавантажити пароль»;
- Головний екран. У разі успішного входження у систему, користувач переходить до головного екрану. На головному екрані користувач бачить список із лекціями та тестами, що заплановані. Якщо користувач натискає на лекцію, то користувач переходить до екрану лекції, після її завантаження. Якщо користувач натискає на тест, то користувач переходить до екрану початку тесту. Зверху є кнопка налаштувань, у разі натискання відбувається перехід до екрану налаштувань. Зокрема, користувач може вийти з системи

- Екран перегляду лекції. На даний екран можна перейти з головного. Даний екран може показувати лекцію як з Web-сторінки, pdf-файл або JSON файл;
- Екран початку тесту. На даному екрані користувач має побачити назву тесту, кількість питань, час виконання. Тут, також, є кнопка «почати тестування», у разі її натискання користувач переходить до екрану тестування;
- Екрани тестування. Даний екран показує користувачеві питання з тесту. Зверху є таймер, де користувач може бачити скільки часу пройшло або залишилось. Користувач може побачити тут саме питання та його номер. Варіанти відповідей, якщо це не питання з відкритою відповіддю. Знизу є кнопка «наступне питання»; користувач може перейти на наступне питання. В залежності від конфігурації тесту, користувач може перейти назад. На останньому питанні є кнопка «завершити тест», у разі натискання відбувається оцінювання.

ВИСНОВКИ

У процесі дослідження питання впровадження систем навчального призначення (СНП) було доведено важливість їх важливість у теперішній час. Було зпроектовано концепт СНП «Булеві функції» за напрямом алгебри логіки. Під час проектування та програмування було залучено технології веб-програмування, було виконано всі завдання. Можна зробити такі висновки.

По-перше, було проаналізовано переваги та систем навчального призначення (СНП). З одного боку, такі системи покращують ефективність навчання та оцінювання. Відбувається значна економія часу і ресурсів. Немає потреби фізичної присутності учасників опитування в аудиторіях. З іншого боку, СНП мають недоліки, а саме: відсутність прямої комунікації між викладачем та студентом, що може знизити встиг. Було проаналізовано основні типи питань у системах дистанційного тестування.

По-друге, було встановлено, що існує багато інструментів для реалізації систем навчального призначення (СНП), такі як Node.js та MySQL. Було спроектовано та реалізовано власну СНП «Булеві функції» засобами Node.js і MySQL у Visual Studio Code та його мобільний клієнт для МОС iOS засобами Xcode та iOS SDK.

По-третє, на основі проведеної роботи можна продовжити дослідження та винести невирішені питання дослідження, а саме:

- Універсалізація системи «Булеві функції» до інших освітніх напрямків;
- Програмування клієнтів для різних ОС;

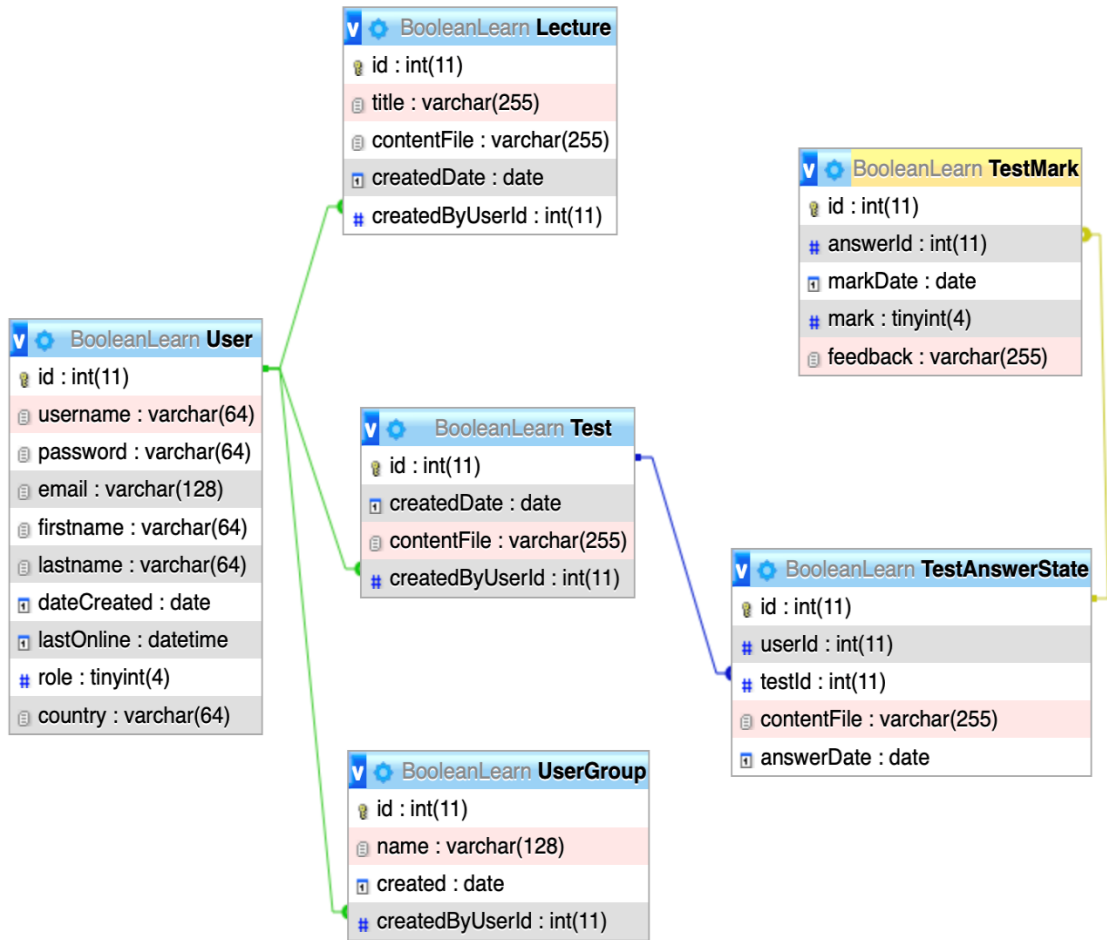
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. This chapter provides a tutorial introduction to MySQL by showing how to use the **mysql** client program to create and use a simple database [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/refman/8.0/en/tutorial.html>
2. Swift (programming language): [Електронний ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/Swift_\(programming_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language)) (дата останнього звернення: 10.05.2021).
3. SwiftUI helps you build great-looking apps: [Електронний ресурс]. – Режим доступу: <https://developer.apple.com/xcode/swiftui/> (дата останнього звернення: 10.05.2021).
4. SwiftUI by Example is the world's largest collection of SwiftUI examples: [Електронний ресурс]. – Режим доступу: <https://www.hackingwithswift.com/quick-start/swiftui> (дата останнього звернення: 10.05.2021).
5. Node.js is an open source server environment. Node.js allows you to run JavaScript on the server. [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com/nodejs/>
6. MySQL is a widely used relational database management system (RDBMS). MySQL is free and open-source. MySQL is ideal for both small and large applications. [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com/mysql/default.asp>
7. Introduction to Node.js. Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project! [Електронний ресурс]. – Режим доступу: <https://nodejs.dev/learn/introduction-to-nodejs>
8. Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web

- applications like video streaming sites, single-page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world. [Электронный ресурс]. – Режим доступа: <https://www.tutorialspoint.com/nodejs/index.htm>
9. The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers. [Электронный ресурс]. – Режим доступа: <https://www.tutorialspoint.com/http/index.htm>.
 10. What is Swagger? If you've ever worked with APIs, chances are, you've heard of Swagger. Swagger is the most widely used tooling ecosystem for developing APIs with the OpenAPI Specification (OAS). Swagger consists of both open source as well as professional tools, catering to almost every need and use case. [Электронный ресурс]. – Режим доступа: <https://swagger.io/tools/open-source/getting-started/>
 11. What's New in the iOS SDK Learn about the key technologies and capabilities available in the iOS SDK, the toolkit you use to build apps for iPhone, iPad, or iPod touch. For detailed information on API changes in the latest released versions, including each beta release, see the iOS & iPadOS Release Notes. [Электронный ресурс]. – Режим доступа: <https://developer.apple.com/ios/whats-new/>
 12. UIKit. Construct and manage a graphical, event-driven user interface for your iOS or tvOS app. [Электронный ресурс]. – Режим доступа: <https://developer.apple.com/documentation/uikit>
 13. Foundation. Access essential data types, collections, and operating-system services to define the base layer of functionality for your app. [Электронный ресурс]. – Режим доступа: <https://developer.apple.com/documentation/foundation>

14. URL Loading System. Interact with URLs and communicate with servers using standard Internet protocols. [Электронный ресурс]. – Режим доступа:
https://developer.apple.com/documentation/foundation/url_loading_system
15. A use case diagram is a graphical depiction of a user's possible interactions with a system. [Электронный ресурс]. – Режим доступа:
https://en.wikipedia.org/wiki/Use_case_diagram
16. The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. [Электронный ресурс]. – Режим доступа:
https://en.wikipedia.org/wiki/Unified_Modeling_Language
17. This part of the JavaScript section on MDN serves as a repository of facts about the JavaScript language. Read more about this reference. [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>
18. Building RESTful web services, like other programming skills is part art, part science. [Электронный ресурс]. – Режим доступа:
<https://www.restapitutorial.com>
19. Building a UIKit user interface programmatically. [Электронный ресурс]. – Режим доступа: <https://www.hackingwithswift.com/read/8/2/building-a-uikit-user-interface-programmatically>
20. In this SwiftUI tutorial, you'll learn how to layout the UI by declaring and modifying views, and how to use state variables to update your UI. [Электронный ресурс]. – Режим доступа:
<https://www.raywenderlich.com/3715234-swiftui-getting-started>

ДОДАТОК А



Додаток А 1 – Схема бази даних