

МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, програмної інженерії та комп'ютерних наук
Кафедра комп'ютерної інженерії та інноваційних технологій

Пояснювальна записка

до кваліфікаційної роботи
другого (магістерського) рівня

на тему **РОЗРОБКА WEB-ОРІЄНТОВАНОЇ E-LEARNING СИСТЕМИ З
ВИКОРИСТАННЯМ МЕТОДИК ГЕЙМІФІКАЦІЇ**

Виконав: студент 2 курсу, групи ІКК-2.1
спеціальності 123 Комп'ютерна інженерія

Пір'ян Ю.В.

Керівник Григор'єва Т.І.

Рецензент Русу О.П.

ДОВІДКА

кафедри КІ та ІТ про виконану магістерську роботу
студента 2 курсу ФКПІ та КН групи ІКК-2.1

Пір'яна Юрія В'ячеславовича

на тему: Розробка web-орієнтованої e-learning системи з використанням методик гейміфікації

Висновок нормоконтролера Позитивна записка до кваліфікаційної роботи
виконана з певною кількістю порушеннями ДСТУ. Оформлено згідно вимог
внутрішнього положення МРДУ.

Нормоконтролер доцент каф. КІ та ІТ, к.т.н. Педан В.В.
(науковий ступінь, вчене звання, посада) (підпис, дата) (і, б. прізвище)

Висновок відповідального за наявність плагиату згідно з сервісним
ІД 1015677900 унікальність роботи підтверджено.

Відповідальна особа доцент каф. КІ та ІТ, к.т.н. Педан В.В.
(науковий ступінь, вчене звання, посада) (підпис, дата) (і, б. прізвище)

Попередня експертиза (захист) магістерської роботи
(бакалаврської роботи чи магістерської роботи)

студ. Пір'ян Ю.В. проведена "12" чудня 2023 р.
(прізвище і б.)

Висновки Виконання МР відповідає завданню, усі
пункти виконано якісно та згідно вимог до
оформлення.

Оригінальність роботи: розроблена система, яка
дозволяє автору створювати курси та тренінги для
навчання за допомогою вироблених елементів 3D-ор.

МР відповідає вимогам до ВРР за завданням
спеціальності ІТС Комп'ютер.інжен та може бути
рекомендована до захисту в ДЕК.

Члени комісії

Лі к.т.н., доц. Цюпка А.Г.
(підпис) (науковий ступінь, вчене звання, посада)

Лі к.т.н., доц. Педан В.В.
(підпис) (науковий ступінь, вчене звання, посада)

Лі викл каф. КІ та ІТ Шибель Р.В.
(підпис) (науковий ступінь, вчене звання, посада)

МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, програмної інженерії та комп'ютерних наук
Кафедра комп'ютерної інженерії та інноваційних технологій
Освітній ступінь магістр
Галузь знань 12 Інформаційні технології
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри КІ та ІТ
к.т.н., доц.
Л.Г.Йона
“ 15 ” 09 2023 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ РОБОТУ

Пір'яну Юрію В'ячеславовичу

1. Тема роботи Розробка web-орієнтованої e-learning системи з використанням методик гейміфікації
керівник роботи Григор'єва Т.І., к.т.н., доцент кафедри ІТ
затверджені наказом закладу вищої освіти від 25.09.2023 р. № 1953
2. Строк подання студентом роботи 11.12.2023р
3. Вихідні дані до роботи: Виконати аналіз існуючих платформ для навчання.
Спроекувати e-learning додаток. Розробити e-learning додаток з методиками
гейміфікації
4. Зміст розрахунково-пояснювальної записки
Розділ 1: Огляд предметної області
Розділ 2: Огляд сучасних рішень
Розділ 3: Проектування системи
Розділ 4: Розробка e-learning ситеми використовуючи методики гейміфікації
5. Перелік графічного матеріалу (з зазначенням обов'язкових креслень)
Слайд 1 – Методики гейміфікації
Слайд 2 – Використані технології
Слайд 3 – Статистика умпінності користувача
Слайд 4 – Ігрова частина додатку
Слайд 5 – Результати
Слайд 6 – Майбутні плани

Слайд 7 – Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав	Завдання прийняв

7. Дата видачі завдання 25.09.2023р

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Вступ	25.09.2023-29.09.2023	<i>Вик</i>
2	Аналіз існуючих e-learning систем.	02.10.2023-11.10.2023	<i>Вик</i>
3	Проектування e-learning додатку	12.10.2023-30.10.2023	<i>Вик</i>
4	Розробка e-learning додатку	31.10.2023-20.11.2023	<i>Вик</i>
5	Висновки та рекомендації	21.11.2023-24.11.2023	<i>Вик</i>
6	Перелік джерел посилання	27.11.2023-28.11.2023	<i>Вик</i>
7	Додаток А	29.11.2023-01.12.2023	<i>Вик</i>
8	Додаток Б	04.12.2023-05.12.2023	<i>Вик</i>
9	Додаток В	06.12.2023-08.12.2023	<i>Вик</i>

Студент *Ю. В. Пір'ян* Ю. В. Пір'ян

Керівник роботи *Т. І. Григорєва* Т. І. Григорєва

ВІДГУК
наукового керівника на кваліфікаційну
магістерську роботу
на тему «Розробка web-орієнтованої e-learning системи з використанням
методик гейміфікації»
студента Пір'ян Ю.В.
Спеціальність: 123 Комп'ютерна інженерія

Однією з головних задач сучасної освіти є підвищення мотивації учнів, що свідчить про актуальність дослідження Пір'яна Ю.В. на тему «Розробка web-орієнтованої e-learning системи з використанням методик гейміфікації». Метою роботи є розробка e-learning web-додатку з методиками гейміфікації з можливістю отримання точної статистики успіхів студента та заохочення до навчання за допомогою методик гейміфікації.

Під час виконання дослідження Пір'ян Ю.В. проявив старанність у розв'язанні визначених завдань. Розроблений високофункціональний та універсальний e-learning web-додаток, що використовує методики гейміфікації для створення захопливих навчальних курсів. Розроблені такі функції, як відстеження статистики, можливість додавання курсів до своєї бібліотеки та в свої улюблені, гнучкість у створенні ігор. Розроблено сторінку користувача, де він отримує швидкий доступ до своєї тижневої статистики, новин та швидкого доступу до курсів. Завдяки цим функціям додаток є зручним та функціональним інструментом для вчителів та учнів.

Результати дослідження представлені у тезах доповідей ІХ Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих учених «Гуманітарний і інноваційний ракурс професійної майстерності: пошуки молодих вчених».

Виходячи з вище зазначеного вважаю, що робота виконана на достатньому методичному рівні та може бути допущена до захисту на засіданні екзаменаційної комісії.

Магістерська робота студента Пір'яна Ю.В. відповідає вимогам до кваліфікаційних робіт магістрів і заслуговує оцінки «відмінно». Студент Пір'ян Ю.В. заслуговує присвоєння за заявленою спеціальністю 123 «Комп'ютерна інженерія» кваліфікації магістр з комп'ютерної інженерії.

Науковий керівник:
завідувачка кафедри Інформаційних технологій,
кандидат технічних наук, доцент



Т.І.Григор'єва

РЕЦЕНЗІЯ

на магістерську роботу студента Пір'яна Ю.В.

Спеціальність: 123 Комп'ютерна інженерія

на тему «Розробка web-орієнтованої e-learning системи з використанням методик гейміфікації»

Магістерська робота студента Пір'ян Ю.В. присвячена розробці e-learning web-додатку, що використовує методи гейміфікації для створення захопливих навчальних курсів. Основна мета додатку полягає в побудові функціональної та ефективної системи електронного навчання з використанням методів гейміфікації, що сприяє покращенню мотивації та активності користувачів у процесі навчання. Розробка і розвиток освітніх веб-платформ набувають стрімкої популярності і попиту, що говорить про актуальність теми магістерської роботи.

В роботі проведено аналіз різноманітних популярних освітніх платформ. Запропонований освітній додаток відповідає останнім тенденціям в сфері e-learning і забезпечує ефективний та цікавий процес навчання для широкого спектру вікових груп – від дітей до дорослих.

Магістерська робота студента Пір'яна Ю.В. відповідає вимогам до випускних магістерських робіт та заслуговує оцінки «відмінно». Студент Пір'ян Ю.В. заслуговує присвоєння за заявленою спеціальністю 123 «Комп'ютерна інженерія» кваліфікації магістр з комп'ютерної інженерії.

Рецензент,
доцент кафедри
комп'ютерних наук,
кандидат технічних наук



О.П.Русу

Ім'я користувача:
Анна Серединко

ID перевірки:
1015995240

Дата перевірки:
11.12.2023 23:53:52 MSK

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
12.12.2023 00:03:48 MSK

ID користувача:
100001433

Назва документа: **Диплом_Пирьян**

Кількість сторінок: 89 Кількість слів: 10344 Кількість символів: 82395 Розмір файлу: 5.14 MB ID файлу: 1015677900

5.53% Схожість

Найбільша схожість: 0.43% з Інтернет-джерелом (https://ela.kpi.ua/bitstream/123456789/40956/1/Yurchenko_magistr.pdf).



0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 10

РЕФЕРАТ

Текстова частина магістерської роботи: 60 с., 55 рисунків, 1 таблиця, 3 додатки, 16 джерел.

ВІРТУАЛЬНА ОСВІТА, ГЕЙМІФІКАЦІЯ, РІВНІ СКЛАДНОСТІ, ЕЛЕМЕНТ, СТАТИСТИКА, DOM

Об'єкт дослідження – підходи до дистанційного навчання та тренінгів, а також можливості їх поліпшення за допомогою впровадження елементів ігор.

Мета роботи – розробка додатку для навчання в будь-якому місці де є інтернет, та заохочення учнів до навчання за допомогою різних ігор та точної статистики успіхів студента.

Метод дослідження – аналітичний з використання комп'ютерних технологій.

У магістерській роботі розроблена система яка дозволяє автору створювати курси за допомогою різних форм для текстового, відео-уроку тощо, та створювати завдання для перевірки засвоєння матеріалу студентом. Студент може проходити модулі тільки ті, які його цікавлять, може помічати їх виконаними тощо. Система має адаптивний дизайн для користування додатком як на ПК так і на телефоні. Додаток розташований на хостингу «Vercel» і надає можливість студентам відстежувати свій прогрес через особистий обліковий запис. Незалежно від того, де користувач проходить заняття, він може переглядати свою статистику та виконувати завдання.

ABSTRACT

Text part of the master's thesis: 60 pages, 55 figures, 1 table, 3 appendices, 16 sources.

VIRTUAL EDUCATION, GAMIFICATION, LEVELS OF COMPLEXITY, ELEMENT, STATISTICS, DOM

The object of the research is approaches to distance learning and training, as well as opportunities to improve them by incorporating game elements.

The aim of the work is to develop an application for learning anywhere with internet access and to encourage students to learn through various games and accurate statistics of student success.

The research method is analytical, using computer technologies.

The master's thesis presents a system that allows the author to create courses using various forms such as text, video lessons, etc., and to create assignments to assess the student's mastery of the material. Students can take modules that interest them, mark them as completed, and more. The system has an adaptive design for use on both PC and mobile devices. The application is hosted on «Vercel» and allows students to track their progress through a personal account. Regardless of where the user attends classes, they can view their statistics and complete assignments.

ЗМІСТ

Перелік скорочень та умовних позначень.....	11
Вступ.....	12
1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....	14
1.1 Актуальність теми.....	14
1.2 Мета та завдання дослідження.....	15
1.3 Аналіз переваг онлайн навчання та гейміфікації для покращення навчання....	16
1.4 Постановка задачі.....	19
1.5 Висновки до розділу.....	21
2 ОГЛЯД СУЧАСНИХ РІШЕНЬ.....	23
2.1 Udeу.....	23
2.2 LinkedIn Learning.....	25
2.3 Skillshare.....	27
2.4 Coursera.....	29
2.5 Khan Academy.....	31
2.6 Висновки до розділу.....	33
3 ПРОЕКТУВАННЯ СИСТЕМИ.....	35
3.1 Вибір технологій для розробки веб-орієнтованої e-learning системи.....	35
3.2 Налаштування проєкту для розробки.....	37
3.3 Проектування та налаштування бази даних для додатку.....	39
3.4 Налаштування аутентифікації.....	41
3.5 Висновки до розділу.....	42
4 РОЗРОБКА E-LEARNING СИТЕМИ ВИКОРИСТОВУЮЧИ МЕТОДИКИ ГЕЙМІФІКАЦІЇ.....	43
4.1 Реалізація backend частини проєкту.....	43
4.2 Реалізація frontend частини.....	46
4.3 Розробка ігрової частини додатку.....	58
4.4 Реалізація статистики успішності користувача.....	63
4.5 Тестування коду.....	65

4.6 Хостинг проєкту	66
4.7 Висновки до розділу	68
ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ	70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	72
Додаток А ПЕРЕЛІК КОПІЙ ДЕМОНСТРАЦІЙНОГО МАТЕРІАЛУ	74
Додаток Б ЛІСТИНГ КОДУ ДЛЯ НАЛАШТУВАННЯ БАЗИ ДАНИХ.....	77
Додаток В ЛІСТИНГ КОДУ ДЛЯ НАЛАШТУВАННЯ АУТЕНТИФІКАЦІЇ	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

HTML – HyperText Markup Language – мова розмітки гіпертексту

CSS – Cascading Style Sheets – каскадні таблиці стилів

tRPC – typed Remote Procedure Call – фреймворк (набір інструментів, бібліотек і правил, які визначають структуру та порядок розробки програмного забезпечення) для швидкої та ефективної реалізації типізованих віддалених процедур (RPC) з TypeScript, спрощуючи взаємодію між різними частинами веб-додатка.

API – Application Programming Interface – інтерфейс програмування застосунків, набір правил, які дозволяють одному програмному застосунку взаємодіяти з іншим, забезпечуючи обмін даними та функціональністю.

SEO – Search Engine Optimization – оптимізація для пошукових систем.

ВСТУП

Сучасні тенденції у сфері освіти свідчать про зростаючу важливість гейміфікації. Зростаюче використання цифрових технологій в освіті відкриває нові можливості для персоналізації навчання та адаптації його до індивідуальних потреб учнів і студентів. Орієнтація на індивідуалізацію та персоналізацію навчання вимагає від освітніх систем розробки нових підходів, які б могли задовольнити різноманітні потреби учнів і студентів. Потреба у підвищенні мотивації учнів і студентів також є однією з ключових тенденцій у сфері освіти. Традиційні методи навчання, засновані на передачі знань від учителя до учня, вже не здатні зацікавити та мотивувати учнів і студентів, які вирісши в епоху цифрових технологій, очікують від навчання захоплюючого досвіду.

Сучасний світ обрання освіти переживає період революційних змін, де традиційні методи навчання вже не відповідають вимогам динамічного та технологічного суспільства. У цьому контексті розробка e-learning систем, яка використовує методику гейміфікації, визначається як стратегічний крок у напрямку поліпшення якості та ефективності освіти. Сучасні учні та студенти вирісши в епоху цифрових технологій, очікують від навчання не лише засвоєння фактів, але й захоплюючого досвіду. Вони шукають стимул для саморозвитку, а гейміфікація в освіті саме надає цей стимул.

Спостерігаючи за динамікою сучасного розвитку, можна побачити, що традиційні методи навчання все більше втрачають свою привабливість для нового покоління. Учні і студенти виявляють відсутність зацікавленості та мотивації, що призводить до погіршення якості освіти та зниження результативності навчальних процесів. Гейміфікація, враховуючи психологічні та мотиваційні аспекти, може стати ефективним інструментом для залучення учнів та студентів до процесу навчання.

Метою даного дослідження є вивчення та аналіз можливостей розробки e-learning системи, що використовує методику гейміфікації, та визначення впливу такого підходу на якість навчання та розвиток учнівських та студентських компетенцій. Об'єктом дослідження є процес розробки та впровадження e-learning системи в контексті сучасної освітньої парадигми.

Зростання важливості цифрових технологій у сучасному суспільстві, спрямованість на індивідуалізацію та персоналізацію освіти, а також потреба у підвищенні мотивації учнів та студентів роблять тему розробки e-learning системи з використанням гейміфікації актуальною та перспективною. Цей дослід розкриє нові горизонти в сфері освіти та визначить переваги та виклики, пов'язані з впровадженням гейміфікації в навчальні процеси.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність теми

Дистанційне навчання, що стає все більш популярним у сучасному світі, виявляється ключовим елементом освіти, завдяки ряду факторів. Зростання технологій та загальна доступність інтернету відкривають нові можливості для студентів і викладачів.

Разом із зручністю дистанційного навчання приходять і свої виклики. Однією з переваг є гнучкість, яку надає дистанційна форма навчання. Студенти можуть вибирати час і місце для навчання, пристосовувати графік до власних потреб і економити час і кошти. Але, на жаль, це може призвести до меншої мотивації та взаємодії. Щоб вирішити ці питання, гейміфікація входить в гру.

Гейміфікація використовує елементи ігрового дизайну в невідповідних сферах, таких як навчання. Вона може стати відповіддю на виклики дистанційного навчання, зробивши його більш привабливим і ефективним. За допомогою гейміфікації можна підняти рівень мотивації студентів, залучити їх до процесу навчання і полегшити взаємодію з викладачами. Елементи гри, такі як бали, рівні, віртуальні нагороди, створюють стимул для досягнення цілей. Це може перетворити навчання в цікавий і викликовий процес. Гейміфікація стає не тільки інноваційним рішенням для проблем дистанційного навчання, але і інструментом, що робить освіту більш захопливою та результативною.

Журнал "The Wall Street Journal" [1] нещодавно опублікував цікаву статтю щодо стану онлайн-навчання. Зокрема, розглядалася динаміка розвитку електронного навчання серед учнів початкової та середньої школи, порівнюючи їхні досягнення з тими, хто отримує традиційну освіту в класах. Хоча жоден із студентів не входить до

категорії учнів K-12, цікавим є зростання кількості батьків, які обирають цей метод для навчання своїх дітей.

Вивчені результати були різноманітні: учні, які повністю зайняті онлайн-навчанням, часто набирали нижчі бали на стандартизованих тестах порівняно із тими, хто навчається у фізичних класах. Однак гібридні програми, які поєднують онлайн та фізичне навчання, демонстрували високі результати.

Зараз багато закладів впроваджують онлайн-навчання в шкільні програми, а кількість студентів, які обрали повноцінне віртуальне навчання, зросла на 40% за останні три роки. Навіть кілька мільйонів учнів беруть хоча б один онлайн-курс. Також зростає тенденція онлайн-освіти для професійного розвитку та підвищення кваліфікації.

Хоча не всі курси однаково ефективні, онлайн-навчання може бути доступним і ефективним засобом отримання необхідних навичок та знань. Інтернет залишається необхідною складовою нашого життя, і статистика свідчить про те, що майбутнє навчання буде все більше відбуватися онлайн.

1.2 Мета та завдання дослідження

Основною метою є проведення глибокого аналізу різних підходів до гейміфікації в освіті. Виокремлення конкретних елементів, таких як бейджі, рівні складності, системи винагород та інші, щоб визначити їхню ефективність та застосовність для e-learning.

Проектування e-learning платформи. Розроблення концепції платформи з використанням гейміфікації, узгодженого з результатами аналізу. Пояснення вибору конкретних гейміфікаційних елементів та їхнього впливу на мотивацію учнів.

Реалізація та тестування. Перенесення задуманої системи в реальний продукт, який може бути випробуваний користувачами. Пояснення обраного методу тестування, збору та аналізу даних, визначення метрик успішності.

Оцінка результатів. Аналіз отриманих даних, враховуючи показники успішності, участі та задоволення учнів. Пояснення висновків та визначення ключових аспектів, які визначають ефективність гейміфікації в навчанні.

Вдосконалення системи. Основано на аналізі результатів та отриманих відгуках, внесення коректив та доповнень до розробленої системи з метою її оптимізації та максимізації користувацького досвіду. Пояснення обраного шляху вдосконалення та передбачення його впливу.

1.3 Аналіз переваг онлайн навчання та гейміфікації для покращення навчання

Концепція традиційної освіти зазнала радикальних змін у останні роки. Більше не потрібно фізично знаходитися в класі, особливо зі зростанням Інтернету та нових технологій. Тепер ви маєте можливість отримати якісну освіту в будь-який час та в будь-якому місці, де є Інтернет.

Онлайн-освіта є розумним вибором як для підлітків, так і для дорослих [2]. Це корисний метод для вдосконалення навичок у складному предметі або вивчення нових навичок.

Основними перевагами є:

- гнучкість;
- широкий вибір програм;
- доступність;
- персоналізований досвід навчання;
- економічно вигідніше.

Гнучкість. Онлайн-освіта дозволяє вчителю та студенту встановлювати свій власний темп навчання, а також є можливість гнучкого графіку, який підходить всім. Це надає можливість краще збалансувати роботу та навчання, не відмовляючись від чого-небудь. Навчання онлайн вчить важливим навичкам управління часом, що полегшує досягнення гарного балансу між роботою та навчанням.

Широкий вибір програм. У просторі Інтернету, як інтернету, є безліч навичок і предметів для вивчення та навчання. Багато університетів та вищих навчальних закладів пропонують онлайн-версії своїх програм для різних рівнів та напрямків. Від композиції музики до квантової фізики - є варіанти для кожного типу студента. Навчання в програмі онлайн також є відмінною опцією для отримання офіційного сертифіката, диплома чи ступеня, не фізично відвідуючи університетський кампус.

Доступність. Онлайн-освіта дозволяє вам вивчати або викладати з будь-якого місця у світі. Це означає, що не потрібно подорожувати з одного місця в інше або дотримуватися жорсткого графіка. Крім того, ви економите час і гроші, які можна витратити на інші пріоритети. Віртуальний клас також доступний будь-де є Інтернет, і це чудовий спосіб використовувати його, подорожуючи. Немає причин відмовлятися від роботи чи навчання, під час подорожей.

Персоналізований досвід навчання. Онлайн-освіта також гнучка для індивідуальних вимог і рівня вмінь кожного студента. Онлайн-класи, як правило, менші за розміром, ніж традиційні, і в більшості випадків, онлайн-навчання дозволяє лише одному студентові одночасно, що забезпечує більшу взаємодію та зворотний зв'язок між вами та вашим викладачем. Зазвичай доступ до дуже різноманітного матеріалу, такого як відео, фотографії та електронні книги, і викладачі також можуть інтегрувати інші формати, такі як форуми чи обговорення, для поліпшення своїх уроків. І цей додатковий контент доступний у будь-який момент з будь-якого місця, що забезпечить вам більш динамічну та індивідуальну освіту.

Економічно вигідніше, ніж традиційна освіта. На відміну від освітніх методів у реальному часі, онлайн-освіта, як правило, є більш доступною. Часто є також широкий вибір варіантів оплати, які дозволяють платити частинами або за кожен клас. Це дозволяє краще управляти бюджетом. Багато з вас також можуть користуватися знижками чи стипендіями, тому ціна рідко є високою. Ви також можете заощадити гроші на дорогу та навчальні матеріали, які часто доступні безкоштовно. Крім того, існує безліч стипендій для онлайн-навчання. Іншими словами, грошові вкладення менше, але результати можуть бути кращими, ніж інші варіанти.

У сфері освіти гейміфікація стає все більш популярною, оскільки вона робить складний матеріал цікавим для вивчення, тим самим мотивуючи студентів. Основною метою гейміфікації в освіті є підвищення якості навчання шляхом збільшення залученості студентів. Розуміється, що люди продовжують робити щось, коли їм це подобається. Це стосується і теорії гейміфікації: учні краще вчаться, коли їм це весело. Крім того, учні намагаються досягти цілей, коли є завдання, цілі та досягнення, які вони можуть виконати в стилі гри. Відеоігри мають до себе пристрась, з якою спілкуються як діти, так і дорослі, тому педагоги можуть використовувати цей самий елемент залучення і застосовувати його до навчальних матеріалів, що дозволяє досягти оптимальних навчальних результатів.

Переваги гейміфікації в освіті Гейміфікація збільшує залученість студентів, використовуючи їхню внутрішню мотивацію [3]. Є різноманітні доказані переваги гейміфікації, такі як:

- власність/відповідальність студента за їхнє навчання;
- залучаюче середовище, що призводить до збільшення уваги та участі;
- навчання з показниками прогресу, такими як набуття балів, значків тощо;
- можливість викладати на різних рівнях когнітивного розвитку, тим самим виводячи студентів із зони комфорту та забезпечуючи більше зв'язків із навчальним матеріалом.

Всі ці елементи підтримують збільшення залученості, дозволяють студентам вивчати нову інформацію, використовуючи творчі методи, і також оцінюють їхні знання.. У багатьох іграх гравці заробляють бали за кожен виклик, який вони зустрічають. Гарна ідея нагороджувати студентів значками, коли вони піднімаються або виконують складні завдання. Приклади гейміфікації в навчанні Гейміфікація в навчанні розроблена для збільшення залученості студентів і вивчення, з'єднуючи освітні матеріали з динамікою гри та функціоналом гри. Ще одним прикладом є Kahoot, рисунок 1.1, інтерактивний веб-сайт, який є простим у використанні.

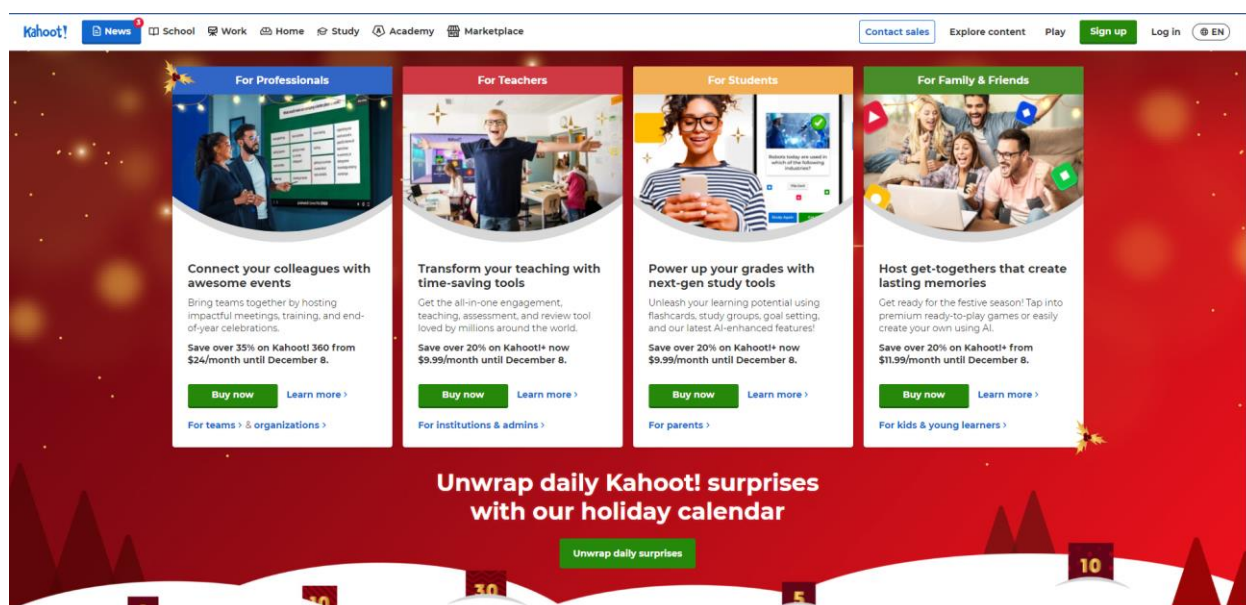


Рисунок 1.1 Веб ресурс Kahoot

Kahoot дозволяє вчителям створювати швидкі багатовибірні оцінювання за допомогою веб-сайту та легко обмінюваним посиланням або кодом. За легкістю використання цієї платформи студенти можуть використовувати мобільний телефон, планшет чи комп'ютер для участі в Kahoot.

Гейміфікація - це процес інтеграції дій за допомогою елементів гри. Метою гейміфікації в освіті є збільшення залученості за допомогою конкуренції, творчості,

вивчення, проведеного студентами та негайного зворотного зв'язку. Гейміфікація стає все більш популярною через її здатність зробити предметний матеріал цікавим, захоплюючим та мотивуючим. Легко використовувати гейміфікацію можна створити за допомогою Kahoot, веб-сайту чи додатку, який дозволяє створювати швидкі багатовибірні оцінювання вчителем і дозволяє студентам легко отримувати доступ за допомогою мобільного телефону, планшета чи комп'ютера.

1.4 Постановка задачі

Основна задача полягає створити додаток де вчитель зможе створювати свій власний курс, зі своїми модулями, уроками та уроками-тестами в одній ресурсі. Основними задачами для реалізації є:

а) Система реєстрації та входу користувача.

1) Мета: Створення зручного процесу реєстрації та входу користувача;

2) Завдання: Розробити та створити сторінки реєстрації та входу користувача.

Реалізувати функціонал для аутентифікації користувача на серверному боці;

б) Сторінка з переліком курсів:

1) Мета: Дозволити користувачам переглядати доступні курси.

2) Завдання: Розробити сторінку для відображення списку курсів.

Отримувати дані про курси з серверної частини та динамічно заповнювати сторінку.

в) Сторінка з деталями курсу:

1) Мета: Надавати детальну інформацію про обраний курс.

2) Завдання: Створити сторінку для відображення деталей курсу.

г) Отримувати та представляти дані про курс у структурованому форматі.

Включити функції, такі як опис курсу, дані про інструктора та відгуки. Сторінка профілю користувача:

1) Мета: Дозволити користувачам керувати своїми профілями та відстежувати прогрес.

2) Завдання: Розробити сторінку профілю користувача. Дозволити користувачам оновлювати інформацію та аватар профілю.

д) Забезпечити можливість перегляду та управління записаними курсами, прогресом та досягненнями.

1) Мета: Дозволити користувачам записуватися на курси.

2) Завдання: Реалізувати функціонал для запису на курс. Оновити інтерфейс користувача, відображаючи статус запису на курс.

е) Статистика користувача:

1) Мета: Надавати користувачам інформацію про їхню активність та прогрес.

2) Завдання: Реалізувати можливості для відстеження та відображення статистики користувача. Включити дані про відсоток завершення курсів, досягнення та прогрес.

є) Додаткові функції (за необхідності):

1) Мета: Покращити користувацький досвід та функціональність платформи.

2) Завдання: Визначити та реалізувати будь-які додаткові функції на основі потреб користувачів та вимог платформи.

ж) Розгортання:

1) Мета: Зробити платформу доступною для користувачів.

2) Завдання: Розгорнути онлайн-платформу для навчання в продакшен середовищі. Моніторити будь-які проблеми та забезпечити плавну роботу.

з) Документація:

1) Мета: Надати комплексну документацію для майбутнього використання.

2) Завдання: Задokumentувати код, API та архітектуру системи. Створити інструкції для адміністраторів, інструкторів та учасників.

Висноски до розділу

E-learning та гейміфікація в навчанні - це справжні інновації які допомагають студентам краще засвоювати матеріал. E-learning дає доступ до навчального матеріалу з будь-якого місця світу, роблячи навчання більш гнучким і доступним. Гейміфікація, у свою чергу, робить навчання цікавішим та захопливим, використовуючи елементи гри. Ці технології важливі, оскільки:

- доступність: забезпечує можливість навчатися на відстані, що особливо актуально у сучасному світі;
- мотивація: гейміфікація створює стимули для досягнення навчальних цілей, роблячи процес навчання цікавим і захоплюючим;
- інтерактивність: e-learning і гейміфікація дозволяють взаємодіяти з матеріалом, що сприяє кращому засвоєнню знань;
- оптимізація часу: можливість самостійно обирати темп і розклад навчання робить процес більш ефективним;
- створення позитивного досвіду: граючи, люди легше запам'ятовують і розуміють інформацію, отже, навчання стає ефективнішим.

2 ОГЛЯД СУЧАСНИХ РІШЕНЬ

Сучасний світ вимагає від нас постійно набувати нові знання, постійно підвищувати кваліфікацію. Дуже актуальною на сьогоднішній день є самоосвіта, тобто сучасна освітня платформа має надавати можливість не тільки здобувати нові знання, а й здійснювати самоконтроль і стимулювати саморозвиток, вона має бути ефективною і функціональною.

В Україні стратегічно важливим є впровадження інноваційних технологій, зокрема електронного навчання (e-learning), як ключового інструменту для досягнення цілей сталого розвитку в освіті [4].

На ринку є дуже багато e-learning ресурсів і конкуренція дуже велика, з численними проектами, які змагаються за увагу та користувачів. Ось кілька ключових аспектів конкуренції: Великі e-learning платформи: Компанії, такі як Coursera, Udemy, та інші, визначають стандарти для онлайн-навчання. Вони пропонують широкий вибір курсів від університетів та організацій з усього світу.

Існують проекти, спрямовані на конкретні галузі або навички. Наприклад, Codecademy фокусується на програмуванні, а LinkedIn Learning надає курси для розвитку кар'єри. Спеціалізовані платформи: Деякі ресурси спеціалізуються на конкретних аспектах навчання, таких як мови (Duolingo), дизайн (Skillshare), чи технічні навички (Pluralsight). Відкриті освітні ресурси (OER): Проекти, як Khan Academy або MIT OpenCourseWare, пропонують безкоштовний доступ до відкритих матеріалів, роблячи освіту більш доступною для всіх.

Багато компаній пропонують корпоративні навчальні платформи для підвищення кваліфікації персоналу. У цьому конкурентному середовищі проекти змагаються за якість контенту, цінність для користувачів, інновації в навчальних методах, а також за активність та взаємодію користувачів.

2.1 Udeуy

Udeуy — це онлайн-платформа для навчання, на якій кожен може створювати та продавати онлайн-курси (рисунок 2.1). Курси на Udeуy бувають різної тривалості та вартості, і їх можна знайти на будь-яку тему, від програмування та маркетингу до мистецтва та кулінарії.

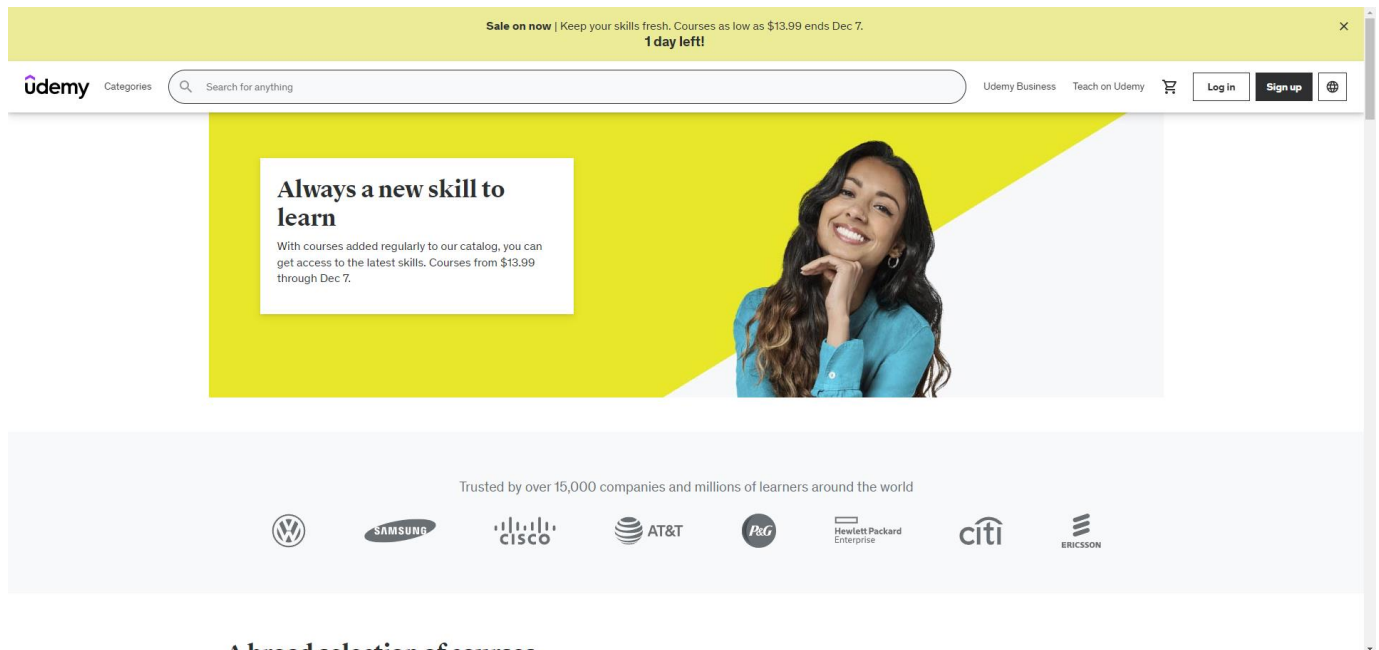


Рисунок 2.1 Навчальна платформа Udeуy

Створити курс на Udeуy може будь-хто, хто має знання та досвід у певній галузі. Для цього потрібно зареєструватися на платформі та заповнити заявку на створення курсу. Після того, як заявка буде схвалена, автор курсів може почати створювати контент.

Контент курсів на Udeуy може включати відео, текстові матеріали, презентації, інтерактивні вправи та інші елементи. Автори курсів також можуть використовувати

різні інструменти для створення контенту, такі як відеоредактори, текстові редактори та інструменти для створення презентацій.

Більшість курсів на UdeMy включають у себе тестування знань учнів. Тести можуть бути у вигляді:

- питань із вибором однієї відповіді;
- питань із вибором кількох відповідей;
- питань відкритого типу;
- практичних завдань.

Тестування знань учнів може проходити в режимі реального часу або після завершення курсу.

Навчання на UdeMy відбувається у власному темпі. Учні можуть переглядати відео, читати текстові матеріали та виконувати завдання в будь-який час і в будь-якому місці. Після завершення курсу учні отримують сертифікат, який підтверджує їхні знання.

Плюси UdeMy:

- широкий вибір курсів на будь-яку тему;
- можливість навчатися у власному темпі;
- має статистику.

Мінуси UdeMy

- якість курсів може бути різною;
- не всі курси включають у себе тестування знань учнів;
- курси платні.

UdeMy — це популярна платформа для онлайн-навчання, яка пропонує широкий вибір курсів на будь-яку тему. Навчання на UdeMy відбувається у власному темпі, і ціни на курси доступні. Однак, якість курсів може бути різною, тому перед покупкою курсу варто прочитати відгуки інших учнів.

2.2 LinkedIn Learning

LinkedIn Learning, раніше відомий як Lynda.com, — це онлайн-платформа для навчання, яка пропонує відеокурси, які викладають експерти галузі програмного забезпечення, творчих навичок і бізнес-навичок (рисунок 2.2). Це дочірня компанія LinkedIn.

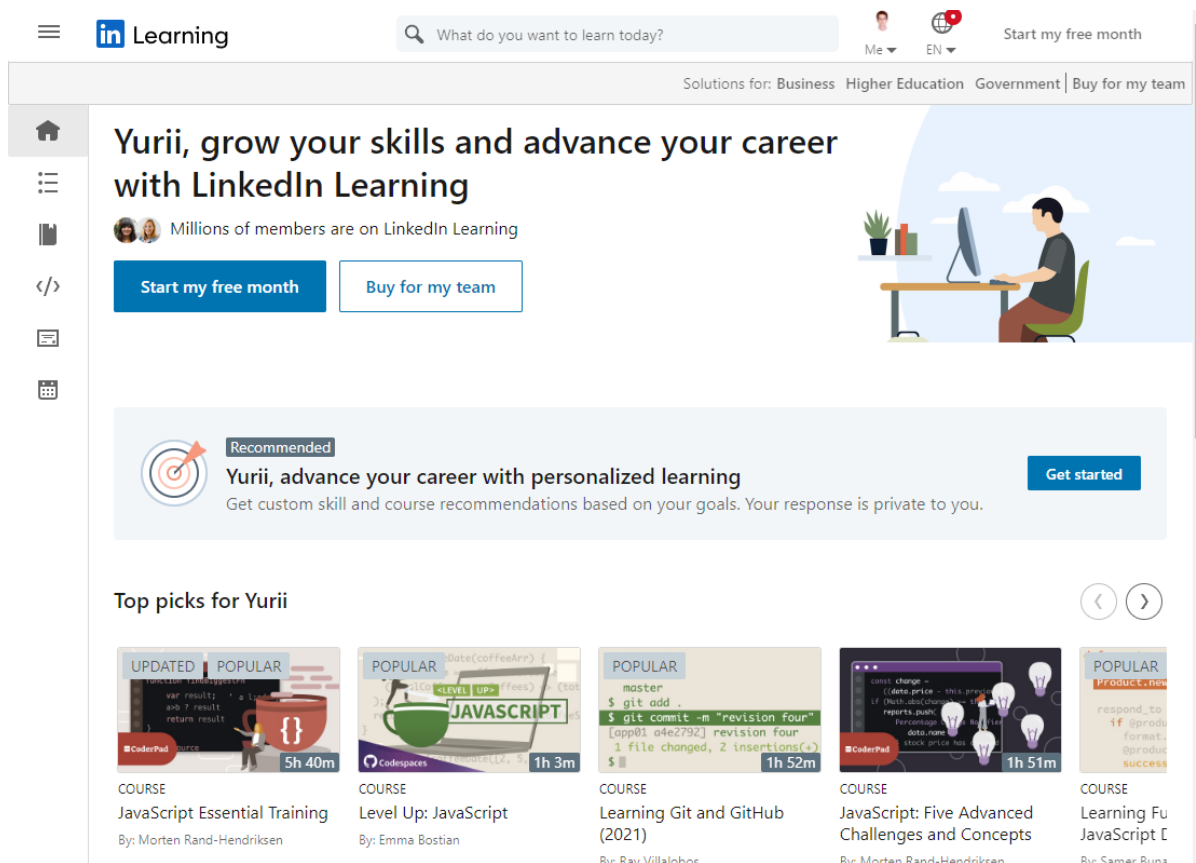


Рисунок 2.2 Навчальна платформа Coursera

Курси на LinkedIn Learning створюються експертами галузі, які мають досвід і знання в певній темі. Автори курсів використовують різні інструменти для створення

контенту, такі як відеоредактори, текстові редактори та інструменти для створення презентацій.

Більшість курсів на LinkedIn Learning включають у себе тестування знань учнів. Тести можуть бути у вигляді:

- питань із вибором однієї відповіді;
- питань із вибором кількох відповідей;
- питань відкритого типу;
- практичних завдань.

Тестування знань учнів може проходити в режимі реального часу або після завершення курсу. Навчання на LinkedIn Learning відбувається у власному темпі. Учні можуть переглядати відео, читати текстові матеріали та виконувати завдання в будь-який час і в будь-якому місці.

Після завершення курсу учні отримують сертифікат, який підтверджує їхні знання.

Плюси LinkedIn Learning

- широкий вибір курсів на теми бізнесу, технологій та творчості;
- високоякісні курси, викладені експертами галузі;
- можливість навчатися у власному темпі.

Мінуси LinkedIn Learning

- деякі курси можуть бути досить короткими;
 - не всі курси включають у себе тестування знань учнів
- Статистика успішності.

LinkedIn Learning пропонує студентам кілька видів статистики успішності, які вони можуть відстежувати. Ця статистика включає:

- прогрес у курсі;
- результати тестування;

– студенти можуть прочитати відгуки інших учнів;

LinkedIn Learning — це чудова платформа для навчання, яка пропонує широкий вибір високоякісних курсів. Вона особливо підходить для людей, які хочуть розвивати свої навички в галузі бізнесу, технологій та творчості.

2.3 Skillshare

Skillshare — це онлайн-платформа для навчання, яка пропонує відеокурси з різноманітних тем, включаючи мистецтво, дизайн, бізнес, маркетинг, технології та багато іншого. Курси на Skillshare створюються професійними викладачами та експертами в галузі, ресурс продемонстровано на рисунку 2.3.

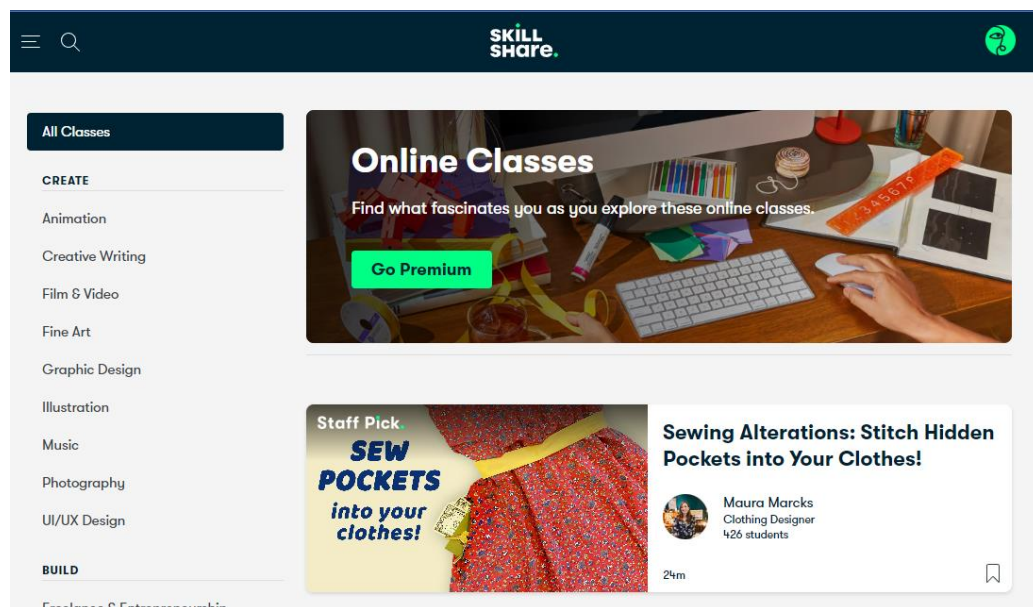


Рисунок 2.3 Навчальна платформа Coursera

Автори курсів використовують різні інструменти для створення контенту, такі як відеоредактори, текстові редактори та інструменти для створення презентацій.

Більшість курсів на Skillshare включають у себе тестування знань учнів. Тести можуть бути у вигляді:

- питань із вибором однієї відповіді;
- питань із вибором кількох відповідей;
- питань відкритого типу;
- практичних завдань.

Тестування знань учнів може проходити в режимі реального часу або після завершення курсу.

Навчання на Skillshare відбувається у власному темпі. Учні можуть переглядати відео, читати текстові матеріали та виконувати завдання в будь-який час і в будь-якому місці.

Після завершення курсу учні отримують сертифікат, який підтверджує їхні знання.

Плюси Skillshare

- широкий вибір курсів на різноманітні теми;
- високоякісні курси, викладені професійними викладачами;
- можливість навчатися у власному темпі.

Мінуси Skillshare

- деякі курси можуть бути досить короткими;
- не всі курси включають у себе тестування знань учнів.

Статистика успішності Skillshare пропонує студентам кілька видів статистики успішності, які вони можуть відстежувати. Ця статистика включає:

- прогрес у курсі;
- результати тестування;
- відгуки інших учнів.

Отже, це онлайн-платформа для навчання, яка пропонує широкий вибір високоякісних курсів на різноманітні теми. Вона особливо підходить для людей, які хочуть розвивати свої творчі навички.

2.4 Coursera

Coursera — це онлайн-платформа для навчання, яка пропонує курси від провідних університетів і компаній світу (рисунок 2.4). Курси на Coursera бувають різної тривалості та вартості, і їх можна знайти на будь-яку тему, від програмування та маркетингу до мистецтва та кулінарії.

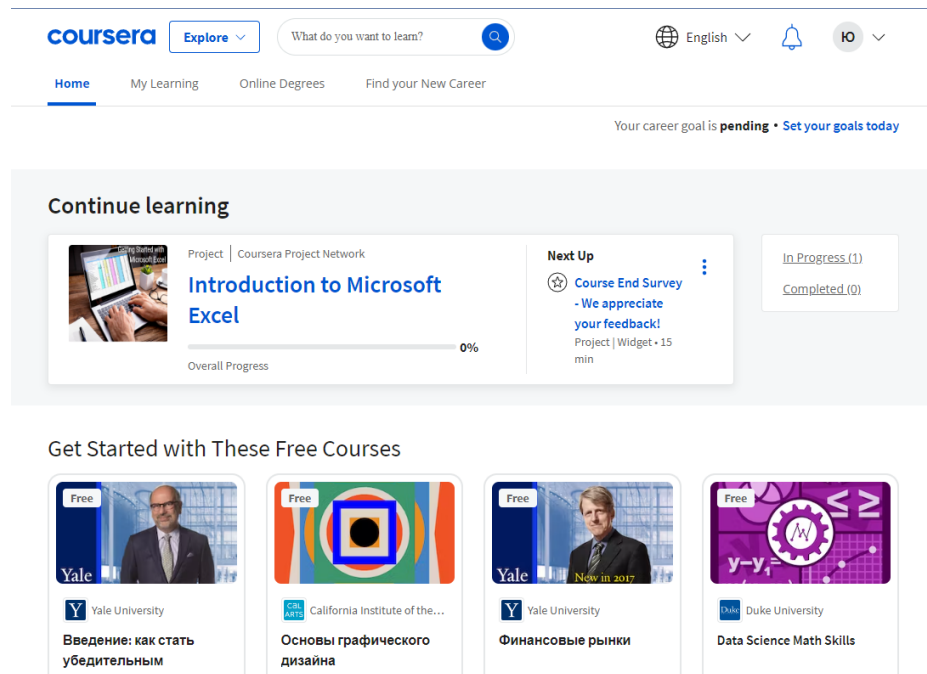


Рисунок 2.4 Навчальна платформа Coursera

Курси на Coursera створюються провайдерами, які можуть бути університетами, компаніями або іншими організаціями. Провайдери створюють курси,

використовуючи різні інструменти для створення контенту, такі як відеоредактори, текстові редактори та інструменти для створення презентацій.

Більшість курсів на Coursera включають у себе тестування знань учнів. Тести можуть бути у вигляді:

- питань із вибором однієї відповіді;
- питань із вибором кількох відповідей;
- питань відкритого типу;
- практичних завдань.

Тестування знань учнів може проходити в режимі реального часу або після завершення курсу.

Навчання на Coursera відбувається у власному темпі. Учні можуть переглядати відео, читати текстові матеріали та виконувати завдання в будь-який час і в будь-якому місці. Після завершення курсу учні отримують сертифікат, який підтверджує їхні знання.

Плюси Coursera

- широкий вибір курсів від провідних університетів і компаній світу;
- високоякісні курси, викладені експертами галузі;
- можливість навчатися у власному темпі.

Мінуси Coursera

- деякі курси можуть бути досить складними;
- не всі курси включають у себе тестування знань учнів.

Статистика успішності Coursera пропонує студентам кілька видів статистики успішності, які вони можуть відстежувати. Ця статистика включає:

- студенти можуть відстежувати свій прогрес у курсі;
- якщо курс включає тести, студенти можуть відстежувати свої результати тестування на окремій сторінці;

– Студенти можуть прочитати відгуки інших учнів.

Отже, це онлайн-платформа для навчання, яка пропонує широкий вибір високоякісних курсів від провідних університетів і компаній світу. Вона особливо підходить для людей, які хочуть розвивати свої професійні навички.

2.5 Khan Academy

Khan Academy — це некомерційна організація, яка пропонує безкоштовні онлайн-курси з різноманітних тем, включаючи математику, науку, історію, комп'ютерне програмування та інші інтерфейс продемонстровано на рисунку 2.5.

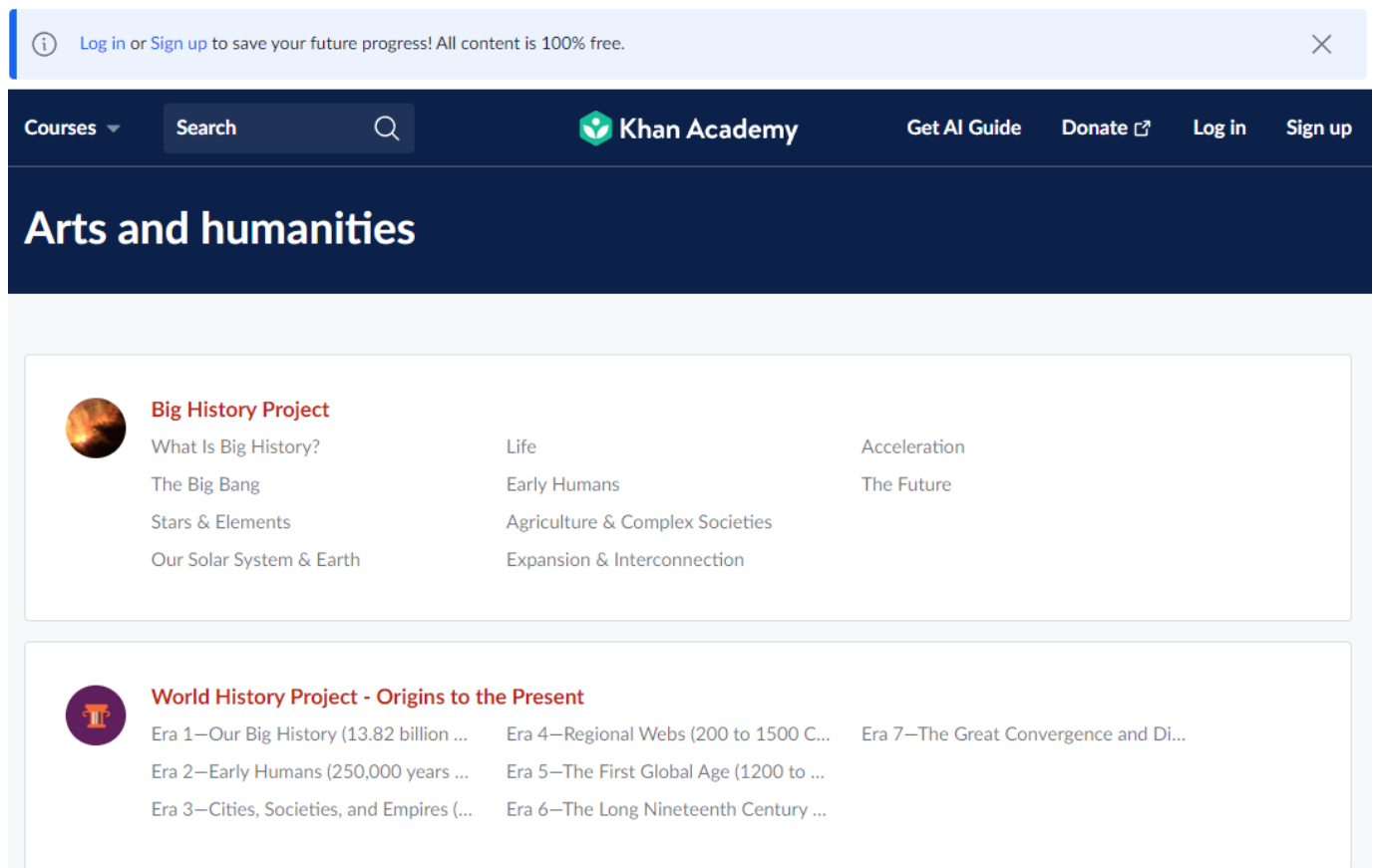


Рисунок 2.5 Навчальна платформа Khan Academy

Курси на Khan Academy створені Салманом Ханом, учителем математики, який хотів надати якісну освіту всім, хто її потребує. Курси на Khan Academy створюються командою викладачів, редакторів та інженерів.

Команда Khan Academy використовує різні інструменти для створення контенту, такі як відеоредактори, текстові редактори та інструменти для створення презентацій.

Khan Academy пропонує різні види тестування знань учнів, включаючи:

- питання із вибором однієї відповіді;
- питання із вибором кількох відповідей;
- питання відкритого типу;
- практичні завдання.

Тестування знань учнів може проходити в режимі реального часу або після завершення курсу.

Навчання на Khan Academy відбувається у власному темпі. Учні можуть переглядати відео, читати текстові матеріали та виконувати завдання в будь-який час і в будь-якому місці.

Плюси Khan Academy

- безкоштовні курси;
- широкий вибір курсів на різноманітні теми;
- високоякісні курси, викладені досвідченими викладачами;
- можливість навчатися у власному темпі.

Мінуси Khan Academy

- деякі курси можуть бути досить короткими;
- не всі курси включають у себе тестування знань учнів.

Статистика успішності Khan Academy пропонує студентам кілька видів статистики успішності, які вони можуть відстежувати. Ця статистика включає: Прогрес у курсі:

- студенти можуть відстежувати свій прогрес у курсі;
- якщо курс включає тести, студенти можуть відстежувати свої результати тестування на окремій сторінці;
- студенти можуть прочитати відгуки інших учнів;
- студенти можуть використовувати статистику успішності, щоб відстежувати свій прогрес, отримувати відгуки та приймати рішення про те, чи варто продовжувати курс.

Khan Academy — це чудова платформа для навчання, яка пропонує безкоштовні курси високої якості. Вона особливо підходить для людей, які хочуть розвивати свої базові навички в різних галузях знань.

Висновки до розділу

LinkedIn Learning, Skillshare, Coursera та Udemy — це всі чудові платформи для онлайн-навчання, які пропонують широкий вибір високоякісних курсів. Порівняння курсів продемонстровано в таблиці 2.1.

Таблиця 2.1 Порівняння сучасних e-learning ресурсів

Категорія	LinkedIn Learning	Skillshare	Coursera	Udemy	Khan Academy
Вибір курсів	Широкий вибір курсів на теми бізнесу, технологій та творчості	Широкий вибір курсів на різноманітні теми	Широкий вибір курсів від провідних університетів і компаній світу	Широкий вибір курсів на різноманітні теми	Широкий вибір курсів з різноманітних тем
Якість курсів	Високоякісні курси, викладені експертами галузі	Високоякісні курси, викладені професійними викладачами	Високоякісні курси, викладені експертами галузі	Високоякісні курси, викладені експертами галузі	Високоякісні курси, викладені професійними викладачами
Ціна	Платно	Платно	Платно	Платно	Безкоштовно
Режим навчання	Навчання у власному темпі	Навчання у власному темпі	Навчання у власному темпі	Навчання у власному темпі	Навчання у власному темпі
Перевірка знань	Присутнє	Присутнє	Присутнє	Присутнє	Присутнє
Статистика успішності	Присутнє	Присутнє	Присутнє	Присутнє	Присутнє

Розглянуті платформи мають різні сильні та слабкі сторони, які слід враховувати при виборі платформи. LinkedIn Learning найкраще підходить для людей, які хочуть розвивати свої професійні навички в галузі бізнесу, технологій або творчості. Skillshare найкраще підходить для людей, які хочуть розвивати свої творчі навички. Coursera найкраще підходить для людей, які хочуть отримати сертифікати від провідних університетів і компаній світу. Udemy найкраще підходить для людей, які хочуть навчатися з широкого вибору курсів на різноманітні теми, включаючи бізнес, технології, творчість, мистецтво, музику, здоров'я та багато іншого.

3 ПРОЕКТУВАННЯ СИСТЕМИ

3.1 Вибір технологій для розробки веб-орієнтованої e-learning системи

При виборі технологій для розробки веб-орієнтованої e-learning системи необхідно враховувати такі фактори: Функціональність системи. Система повинна забезпечувати всі необхідні функції для навчання, такі як:

- реєстрація та авторизація користувачів;
- навчання за курсами тестування та оцінювання;
- адміністрування системи;
- масштабованість системи. Система повинна бути масштабованою, щоб підтримувати зростання кількості користувачів і даних;
- безпека системи. Система повинна бути безпечною для зберігання даних користувачів і запобігання несанкціонованому доступу;
- легкість розробки та підтримки системи. Технології повинні бути добре відомими та підтримуватися широким співтовариством розробників;
- на основі цих факторів для розробки веб-орієнтованої e-learning системи були обрані такі технології:
 - Tailwind CSS. Tailwind CSS - це фреймворк CSS, який дозволяє швидко та легко створювати стильні веб-сайти. Він забезпечує широкий спектр готових компонентів та стилів, які можна використовувати для створення візуального оформлення системи;
 - tRPC. tRPC [4] - це фреймворк для створення RESTful API [14]. Він забезпечує простий спосіб створення API, які можна використовувати для взаємодії з серверною частиною системи;

– MongoDB. MongoDB - це документо-орієнтована база даних, яка добре підходить для зберігання даних e-learning системи [6]. Вона забезпечує швидкий доступ до даних та просту модель оновлення;

– Prisma. Prisma - це фреймворк для об'єднання MongoDB та tRPC. Він забезпечує простий спосіб створення API, які взаємодіють з даними в MongoDB [5];

– Next.js. Next.js - це фреймворк для створення статичного сайту з можливістю генерації динамічного контенту [7]. Він забезпечує високу продуктивність та хорошу SEO оптимізацію [15];

– TypeScript. TypeScript - це надмножина JavaScript [13], яка додає типізацію. Вона забезпечує надійність та безпеку коду.

Дані технології забезпечують високу функціональність, масштабованість, безпеку та легкість розробки та підтримки e-learning системи.

Особливості обраних технологій:

а) Tailwind CSS

- 1) простота використання;
- 2) широкий спектр готових компонентів та стилів;
- 3) налаштовуваність.

б) tRPC

- 1) простота використання;
- 2) підтримка RESTful API;
- 3) розширюваність.

в) MongoDB

- 1) швидкий доступ до даних;
- 2) проста модель оновлення;
- 3) масштабованість.

г) Prisma

- 1) простота використання;

- 2) підтримка MongoDB;
- 3) розширюваність.

д) Next.js

- 1) висока продуктивність;
- 2) хороша SEO оптимізація;
- 3) надійність.

е) TypeScript

- 1) надійність;
- 2) безпека;
- 3) налаштуваність.

Обрані технології забезпечують високу якість розробки веб-орієнтованої e-learning системи. Вони відповідають усім необхідним вимогам та забезпечують оптимальний баланс між функціональністю, масштабованістю, безпекою та легкістю розробки та підтримки.

3.2 Налаштування проєкту для розробки

Використовуючи Next.js для розробки можна легко створити стартовий набір для початку розробки за допомогою скрипта в терміналі "npx create-next-app@latest" далі буде створено стартові папки та налаштований webpack для розробки проєкту стартова структура проєкту продемонстровано на рисунку 3.1

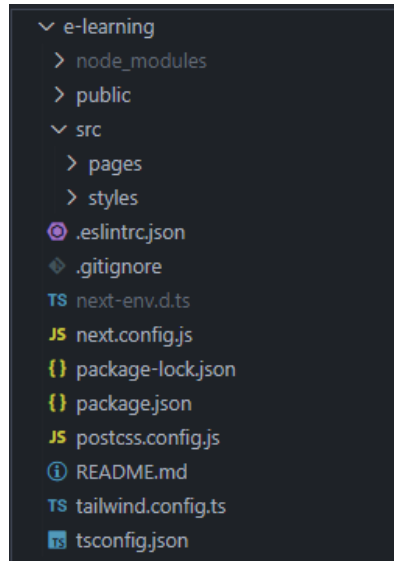


Рисунок 3.1 Структура проєкту після створення

У контексті забезпечення читабельності та зручності розробки для інших учасників команди рекомендовано використовувати архітектурну методологію. Обрана Feature-Sliced Design визначається розділенням проєкту на функціональні "фічі". Це створює чіткі межі між різними модулями та дозволяє кожному розробнику швидко зорієнтуватись у кодовій базі. Тепер структура проєкту продемонстровано на рисунку 3.2

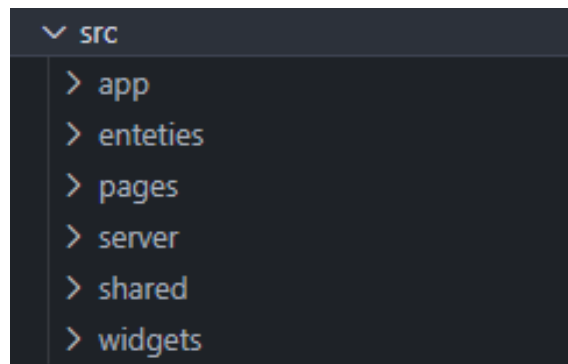


Рисунок 3.2 Структура проєкту

Для стилізації інтрефейсу будемо використовувати HTML та Tailwind.

HTML (Hypertext Markup Language) — це мова розмітки, використовувана для створення структури веб-сторінок. Вона визначає різні елементи на сторінці, такі як заголовки, абзаци, посилання та інші.

Tailwind CSS — це інструмент для створення стильового оформлення веб-сторінок шляхом використання набору готових класів, які визначають стилізацію елементів. Він пропонує набір заздалегідь визначених класів, які можна застосовувати безпосередньо до HTML-елементів для визначення їх стилів. Наприклад, клас "bg-blue-500" встановить фон елемента в синій колір.

Tailwind CSS має кілька вагомих переваг, які роблять його популярним серед розробників. Завдяки використанню готових класів, розробка стає швидше, оскільки не потрібно вручну писати CSS-стилі. Велика кількість готових класів покриває різні стилі та елементи. Tailwind надає гнучкість у виборі стилів і їх комбінації. Розробник може використовувати готові класи або налаштовувати свої власні, щоб відповідати конкретним потребам проекту. Можливість налаштування Tailwind дозволяє розробникам вибирати лише ті класи, які їм потрібні, що зменшує обсяг кінцевого CSS-коду.

3.3 Проектування та налаштування бази даних для додатку

MongoDB — це сучасна база даних, яка вирізняється своєю гнучкістю та можливістю легко адаптуватися до змін у структурі даних. Вона використовує документи у форматі JSON-подібних об'єктів для зберігання інформації і дозволяє додавати та змінювати поля без складних перетворень.

MongoDB маючи свою гнучку схему дозволяє розробникам додавати та змінювати поля у документах без необхідності проведення складних міграцій. Це особливо корисно в сучасних розробках, де вимоги до даних можуть змінюватися дуже швидко. Велика свобода в роботі зі структурою даних дозволяє більш ефективно

втілювати ідеї та вносити зміни у програмному забезпеченні. Додатково, MongoDB відзначається динамічними запитами, які дозволяють здійснювати виразне отримання даних. Запити можуть включати в себе вкладені структури та масиви, надаючи розробникам великі можливості у роботі з складними об'єктами. Це спрощує розробку, оскільки дозволяє отримувати та модифікувати дані у форматі, який найбільш природний для задач конкретного додатку.

Для зберігання даних використовується MongoDB, вона використовує документо-орієнтовану модель даних, що відрізняється від реляційних баз даних. У MongoDB не існує термінів "primary key" чи "secondary key", як у традиційних реляційних базах даних. Замість цього, кожен документ у колекції (еквівалент таблиці в реляційних базах) має унікальний ідентифікатор, який називається "_id". Це поле виступає в ролі унікального ключа для кожного документа в колекції. MongoDB також дозволяє індексувати будь-яке поле в документі для підвищення швидкодії пошуку.

Для додатку було створено 2 однакові бази даних під назвами "e-learning" для тестування коду на запити до бази даних та перевірки додатку в режимі розробки та "e-learning-prod" для фінальної не змінної бази даних для зберігання інформації користувачів продемонстровано на рисунку 3.3.

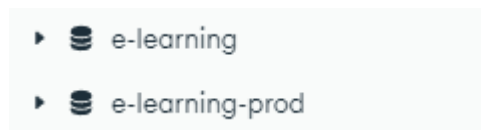


Рисунок 3.3 Бази даних для додатку

Для бази даних створимо сутності:

- users;
- courses;
- modules;

- lessons;
- news;
- reviews.

Створені сутності продемонстровано на рисунку 3.4

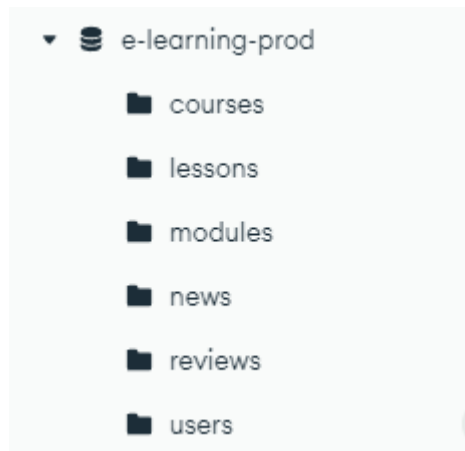


Рисунок 3.4 Сутності бази даних

Для кожної сутності потрібно створити поля де будуть зберігатися данні користувача. Використовуючи Prisma яка допомагає взаємодіяти MongoDB з tRPC. Для початку в терміналі потрібно використати команди:

- `npm install prisma --save-dev;`
- `npx prisma init.`

До бази даних потрібно підключитися за допомогою скрипту рисунок 3.5. В url замість "DATABASE_URL" потрібно використовувати свії "connection url"

```
datasource db {
  provider = "mongodb"
  url      = env("DATABASE_URL")
}

generator client {
  provider = "prisma-client-js"
}
```

Рисунок 3.5 Скрипт для підключення проєкту до бази даних

Далі потрібно написати сутності для того щоб запулити їх до бази даних код в додатку Б.

3.4 Налаштування аутентифікації

Для аутентифікації використано NextAuth.js — це бібліотека для автентифікації користувачів в Next.js додатках. Вона надає готові рішення для різних видів автентифікації, таких як вхід через соціальні мережі, локальний вхід з використанням електронної пошти та пароля, або використання вже існуючого облікового запису.

NextAuth.js легко інтегрується з Next.js проєктами і надає зручний інтерфейс для налаштування різних стратегій автентифікації. Вона підтримує різні стратегії автентифікації, такі як OAuth, JWT, або локальна автентифікація, що дозволяє вибрати найбільш підходящий спосіб для проєкту. Підтримка соціальних мереж: NextAuth.js має готові рішення для автентифікації через популярні соціальні мережі, такі як Google, Facebook, Twitter тощо. Вона надає можливість використовувати JWT (JSON Web Tokens) для забезпечення безпеки та авторизації користувачів.

Для встановлення потрібно в терміналі написати команду "npm install next-auth" та створити в папці api папку auth з файлом [...nextauth].ts в цьому файлі потрібно

описати які стратегії аутентифікації будуть використані та налаштування лістинг коду продемонстровано в додатку В. Щоб все працювало правильно основний компонент потрібно обернути в "Session provider" продуманостровано на рисунку 3.6

```
const App: FC<AppPropsWithLayout> = ({Component, pageProps}) => {
  const getLayout = Component.getLayout ?? (page => page);
  const router = useRouter();

  const [loading, setLoading] = useState(false);

  useEffect(() => {
    const handleStart = (url: string) => (url !== router.asPath) && setLoading(true);
    const handleComplete = (url: string) => (url !== router.asPath) && setTimeout(() => setLoading(false), 500);

    router.events.on("routeChangeStart", handleStart);
    router.events.on("routeChangeComplete", handleComplete);
    router.events.on("routeChangeError", handleComplete);

    return () => {
      router.events.off("routeChangeStart", handleStart);
      router.events.off("routeChangeComplete", handleComplete);
      router.events.off("routeChangeError", handleComplete);
    };
  }, [loading]);

  if (loading) {
    return <div className={"bg-light-background dark:bg-dark-background h-screen w-screen flex justify-center items-center"}><Loader /></div>;
  }

  return (
    <ReduxProvider>
      <SessionProvider session={pageProps.session}>
        {getLayout(<Component {...pageProps} />)}
      </SessionProvider>
    </ReduxProvider>
  );
};

export default trpc.withTRPC(App);
```

Рисунок 3.6 Основний компонент обернений в Session Provider

Висновки до розділу

Було вдосконалено навички з роботою над додатком електронної навчальної системи, в якому ми успішно використовували методики гейміфікації. Було обрано сучасні інструменти, такі як Next.js та TypeScript, для розробки веб-орієнтованої системи, тому що це ефективні інструменти для вирішення завдань.

Процес проєктування та налаштування бази даних з використанням MongoDB та Prisma допоміг з швидким налаштуванням середовища для роботи. Було додано систему аутентифікації з використанням бібліотеки NextAuth [17], що підвищує безпеку та зручність для користувачів.

4 РОЗРОБКА E-LEARNING СИТЕМИ ВИКОРИСТОВУЮЧИ МЕТОДИКИ ГЕЙМІФІКАЦІЇ

4.1 Реалізація backend частини проєкту

Backend — це невидима частина, яка відповідає за обробку запитів користувача, взаємодію з базою даних, виконання бізнес-логіки та відправку необхідних даних на "frontend". Backend забезпечує функціональність, яка необхідна для роботи додатка, але не відображається безпосередньо для користувача.

Одним із завдань бекенду є забезпечення ефективної обробки запитів від фронтенду, збереження та управління даними, забезпечення безпеки та виконання інших функцій, які роблять додаток повноцінним та працездатним.

Для серверної частини було обрано tRPC. tRPC (Typed RPC) — це бібліотека для створення зручних та типізованих API на Node.js. Вона забезпечує статичну типізацію для серверних та клієнтських запитів, що робить розробку більш надійною та зручною.

Однією з ключових переваг tRPC є статична типізація. Можна описати API-запити за допомогою TypeScript, що дозволяє виявляти помилки та недоліки на етапі розробки. tRPC генерує автоматичну документацію для API, що полегшує розуміння та використання API.

tRPC пропонує концепцію "zero-config", що робить процес розробки більш простим та менш витратним за часом. Також легко інтегрується з Prisma, що дозволяє зручно використовувати його для доступу до бази даних. Можна описати та типізовані запити для роботи з Prisma моделями. Підтримка різних типів передачі даних: tRPC підтримує передачу даних через JSON, GraphQL, або інші формати, забезпечуючи гнучкість та сумісність з різними стеками технологій. З використанням tRPC для

backend частини проекту можна отримати не тільки типізацію та зручність, але й швидкість розробки та покращену підтримку бази даних, зокрема з Prisma.

Для коректної роботи trpc потрібно створити в папці arі папку trpc з файлом [trpc].ts, код для обробки запитів продемонстровано на рисунку 4.1

```
import * as trpcNext from "@trpc/server/adapters/next";
import { appRouter } from "@server/routers/_app";
import { createContext } from "@server/context";

export default trpcNext.createNextApiHandler({
  router: appRouter,
  createContext,
  onError({ error }) {
    if (error.code === "INTERNAL_SERVER_ERROR") {
      // send to bug reporting
      console.error("Something went wrong", error);
    }
  },
});
```

Рисунок 4.1 Скрипт дял обробки запитів

Далі потрібно створити папку server продемонстровано на рисункку 4.2 де потрібно проініціалізувати tRPC та створити файл для контексту щоб було простіше потім використовувати "роути " .

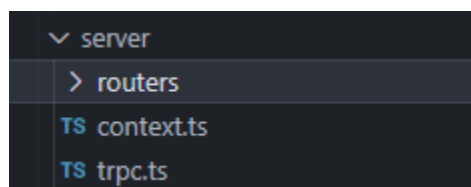


Рисунок 4.2 Структура папки server

В файлі trpc.ts потрібно написати скрипт для ініціалізації контексту код продемонстровано на рисунку 4.3


```
import { initTRPC } from "@trpc/server";
import { Context } from "../context";
import { transformer } from "@shared/utils/transformer";

const t = initTRPC.context<Context>().create({
  transformer
});

export const router = t.router;
export const procedure = t.procedure;
```

Рисунок 4.3 Скрипт для ініціалізації контексту

Для файлу context.ts потрібно написати скрипт для створення контексту, код продемонстровано на рисунку 4.4

```
import * as trpc from "@trpc/server";
import * as trpcNext from "@trpc/server/adapters/next";
import { prisma } from "@shared/utils/prisma";
import { CreateContextOptions } from "vm";

// create context based of incoming request
export const createContext = async (
  opts?: trpcNext.CreateNextContextOptions,
) => {
  return {
    req: opts?.req,
    prisma,
    user: prisma.users,
    course: prisma.courses,
    modules: prisma.modules,
    lessons: prisma.lessons,
    news: prisma.news,
    reviews: prisma.reviews
  };
};

export type Context = trpc.inferAsyncReturnType<typeof createContext>;

export async function createContextInner(_opts: CreateContextOptions) {
  return {};
}
```

Рисунок 4.4 Скрипт для створення контексту

В папці routers потрібно створити файли з скриптами для того щоб клієнт міг звертатися до API та отримувати данні з бази даних.

API — це скорочення від "Application Programming Interface" (Інтерфейс програмування додатків). Це набір правил і інструментів, які дозволяють різним програмам взаємодіяти одна з одною. API визначає, які запитання можна робити до програми, які дані можна отримати з програми, а також які дії можна виконати. У зв'язку з цим, можна уявити API як мовний посередник між різними програмами, який дозволяє їм розуміти одна одну та обмінюватися інформацією. Наприклад, у веб-розробці API може визначати, як фронтенд (клієнтська частина) може взаємодіяти з бекендом (серверною частиною) для отримання або надсилання даних.

API використовується в різних областях програмування, від веб-розробки до робототехніки, та є важливим інструментом для покращення взаємодії між програмами. Структура папки routers продемонстровано на рисунку 4.5

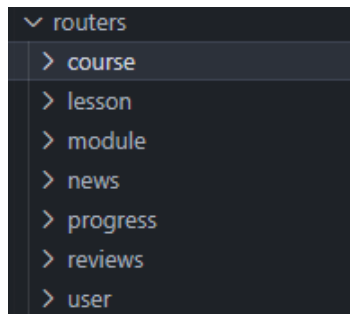


Рисунок 4.5 Структура папки routers

Кожна папка відповідає за свою область запитів до бази даних це зроблено для кращого розуміння коду.

4.2 Реалізація frontend частини

Frontend — це та частина, яку користувач бачить і з якою взаємодіє. Це візуальна та інтерактивна частина додатка, яка відображається у веб-браузері або іншому інтерфейсі користувача.

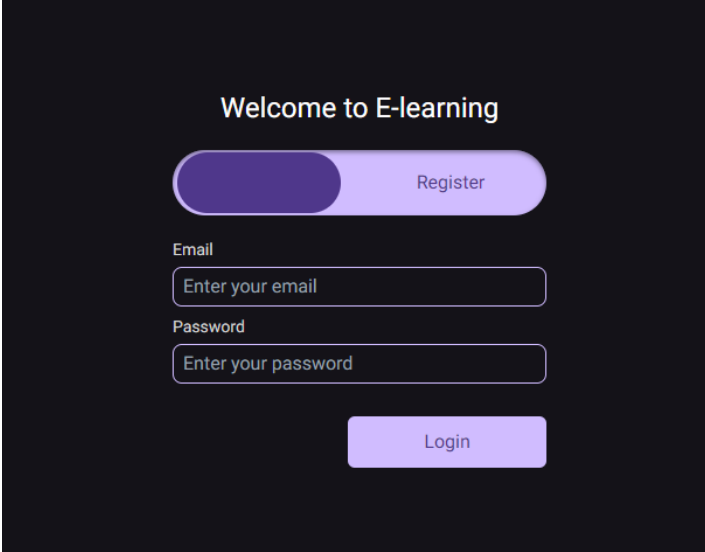
Frontend включає в себе різноманітні елементи, такі як текст, зображення, кнопки, форми та інші елементи інтерфейсу. Він відповідає за те, як виглядає та взаємодіє користувач з додатком. Також включає в себе розміщення та стилізацію елементів, анімацію, валідацію введених даних та інші аспекти, які стосуються зовнішнього вигляду та взаємодії.

Frontend часто використовує мови програмування та технології, такі як HTML, CSS та JavaScript. HTML відповідає за структуру сторінки, CSS - за стилізацію та вигляд, а JavaScript - за інтерактивність та динамічні функції.

Для додатку було розроблено такі сторінки:

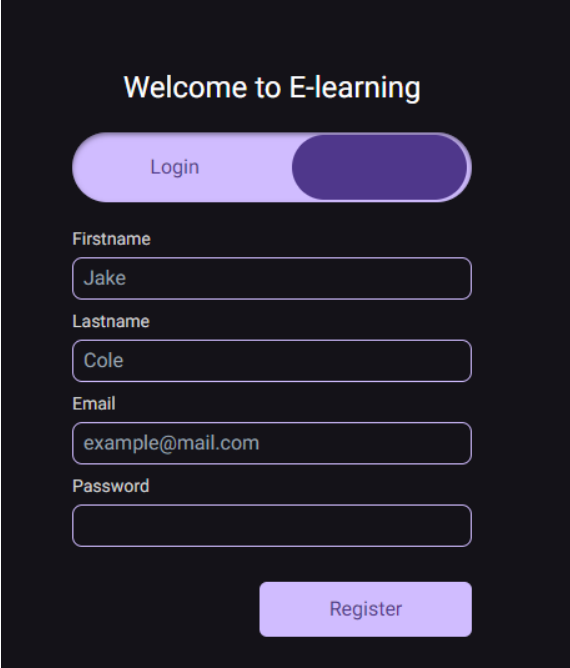
- сторінка авторизації та реєстрації;
- основна сторінка;
- сторінка з усіма курсами в додатку;
- сторінка профілю (якщо користувач авторизований);
- курси які проходить користувач;
- сторінка курсу;
- сторінка модулю;
- сторінка урока;
- прогрес користувача та оцінки;
- адмін панель (якщо користувач має роль "admin").

Сторінка авторизації дає можливість користувачу заходити на свій акаунт щоб проходити обрані курси та відстежувати свій результат, якщо користувач не має акаунту він може зареєструватися перемкнувши перемикач на сторінку реєстрації. Форми аутентифікації та реєстрації продемонстровано на рисунку 4.6 та рисунку 4.7



The screenshot shows a dark-themed login form titled "Welcome to E-learning". At the top, there is a toggle switch with the "Register" option selected. Below the toggle are two input fields: "Email" with the placeholder text "Enter your email" and "Password" with the placeholder text "Enter your password". A "Login" button is positioned at the bottom right of the form.

Рисунок 4.6 Форма аутентифікації

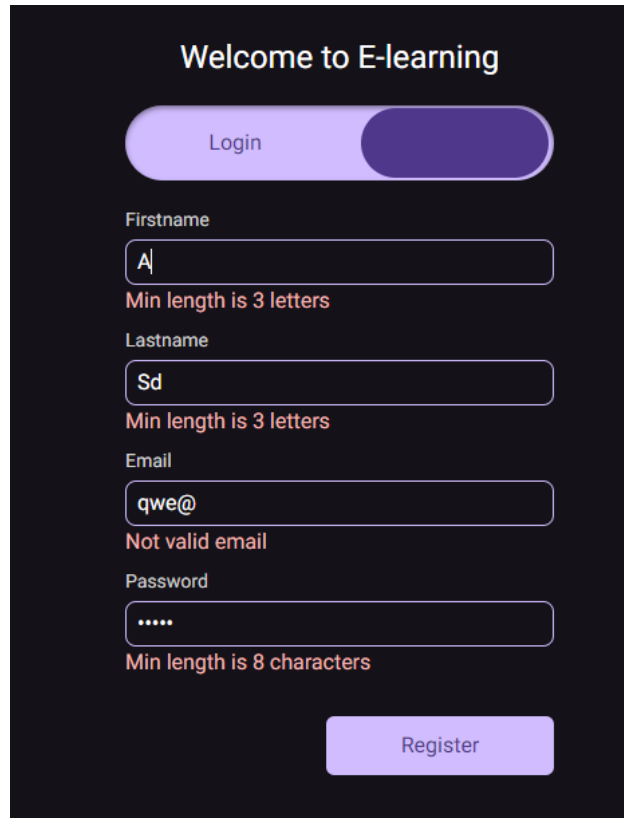


The screenshot shows a dark-themed registration form titled "Welcome to E-learning". At the top, there is a toggle switch with the "Login" option selected. Below the toggle are four input fields: "Firstname" with the value "Jake", "Lastname" with the value "Cole", "Email" with the value "example@mail.com", and "Password" which is currently empty. A "Register" button is positioned at the bottom right of the form.

Рисунок 4.7 Форма реєстрації

Для реєстрації також потрібно щоб користувач вводив валідні данні, щоб наприклад email мав відповідний вигляд, тому для форм також використовується валідація. Якщо користувач захоче написати невалідний email йому буде про це

повідомлено, також ім'я та прізвище повинно мати не менше 3 символів та пароль складатися не менше ніж з 8 символів, помилки продемонстровано на рисунку 4.8



The image shows a registration form on a dark background. At the top, it says "Welcome to E-learning". There are two buttons: "Login" (disabled) and "Register" (active). The form has four input fields with error messages below them:

- Firstname:** Input contains "A". Error message: "Min length is 3 letters".
- Lastname:** Input contains "Sd". Error message: "Min length is 3 letters".
- Email:** Input contains "qwe@". Error message: "Not valid email".
- Password:** Input contains ".....". Error message: "Min length is 8 characters".

Рисунок 4.8 Помилки при невалідних даних

Додатково потрібно перевіряти чи не має такого ж email в базі даних та повідомляти про це користувача. При вдалій реєстрації користувач може залогінитись після чого його відправляє на сторінку профіля.

Сторінка профілю допомагає користувачу отримати швидкий доступ до інформації по його статистиці за весь його шлях навчання. Користувач може швидко зайти на свій останній курс, перевірити кількість пройдених уроків за тиждень, подивитись останні новини додатку, допройти уроки які він додавав до вкладки "Complete later", обрати свій улюблений курс та переходити на нього по бажанню,

також переглядати інформації. По кількості пройдени курсів, модулів, уроків, сторінка провілю продемонстрована на рисунку 4.9

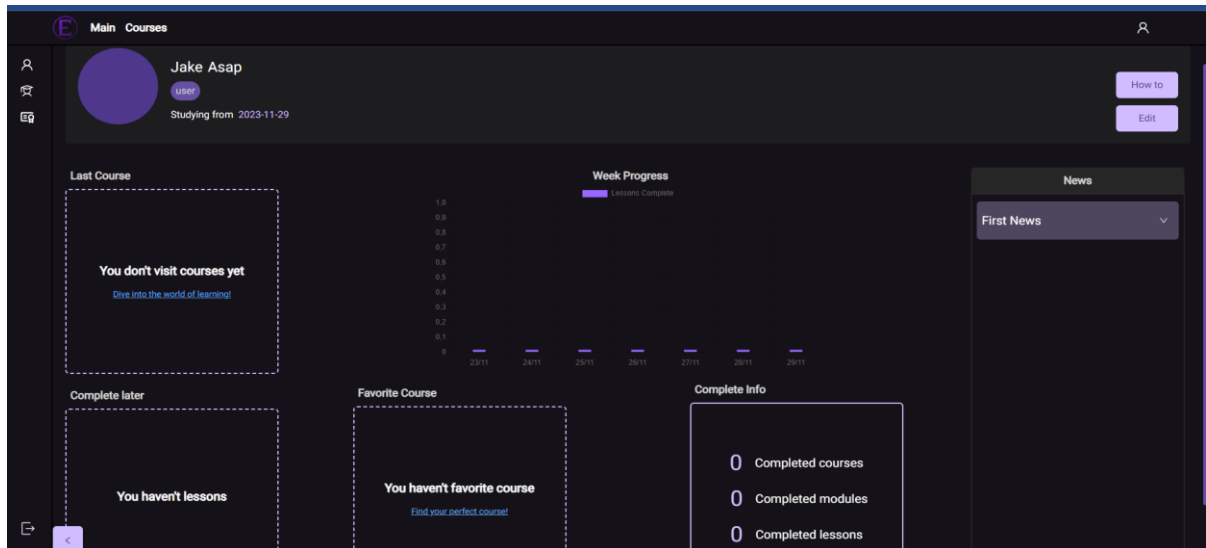


Рисунок 4.9 Сторінка профілю

На сторінці також киристувач може змінити собі аватар продемонстровано на рисунку 4.10 та змінювати свої данні за бажанням продемонстровано на рисунку 4.11

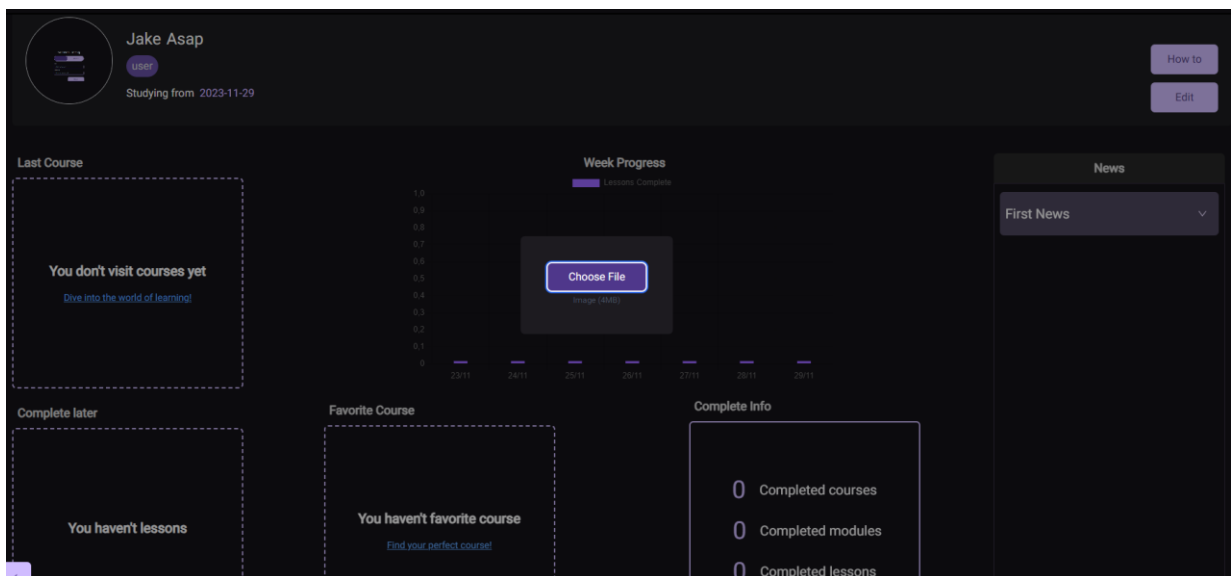


Рисунок 4.10 Форма для зміни аватара

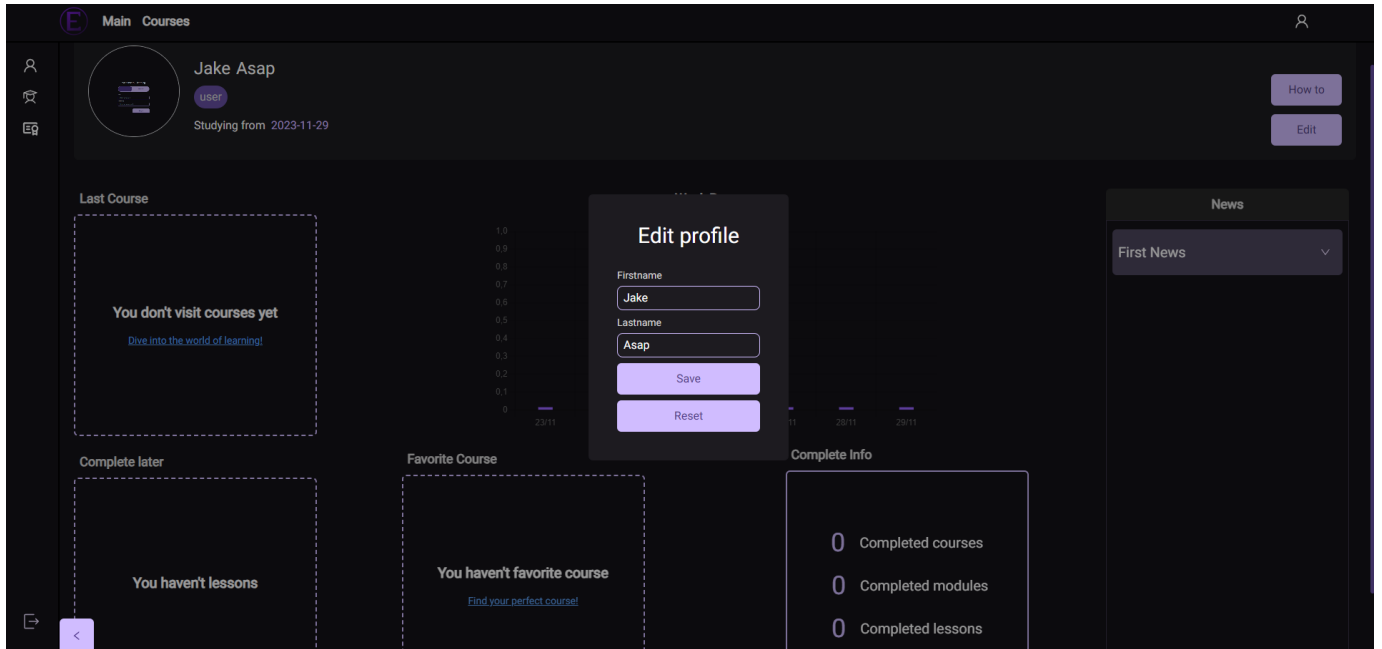


Рисунок 4.11 Форма для зміни даних користувача

Щоб користувачу було легко освоїтись з додатком є кнопка "How to" яка відкриває форму де все покроково описано як користуватись продемонстровано на рисунку 4.12

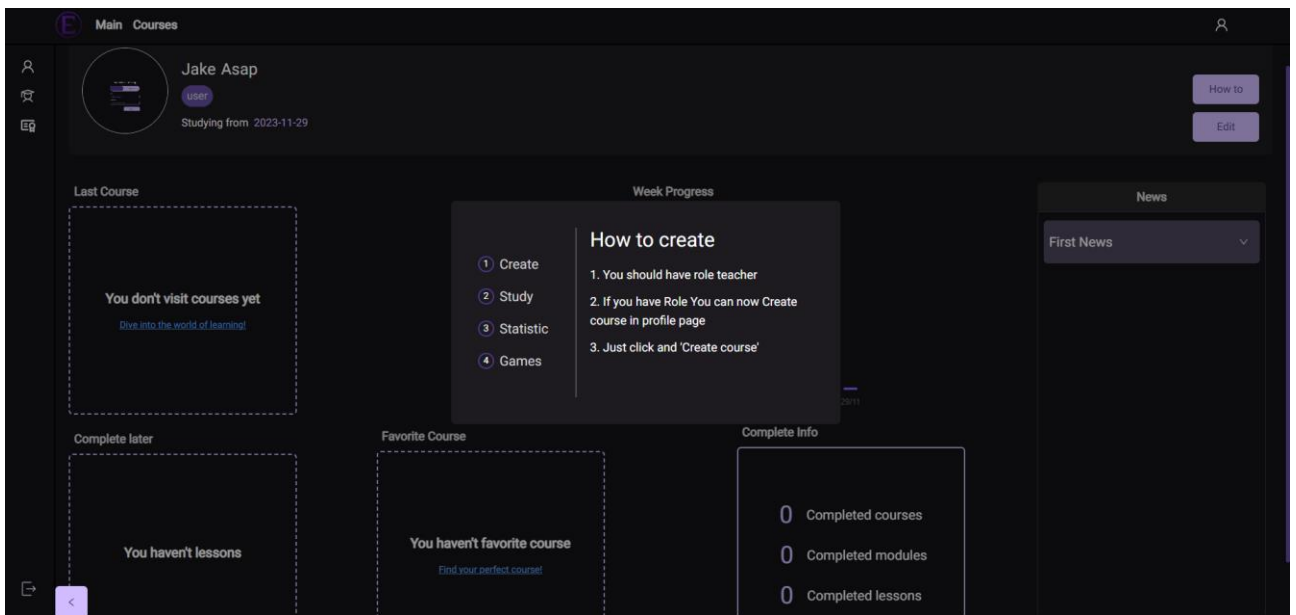


Рисунок 4.12 Інструкція користувача

Якщо користувач має роль "admin" він може створити курси або створювати новини, для такого користувача додається кнопка на сторінці профілю (рисунк 4.13)



Рисунок 4.13 Профіль для користувача admin

Коли користувач натискає кнопку "Create course" то з'являється форма продемонстрована на рисунку 4.14 форма також має валідації. На некоректні данні.

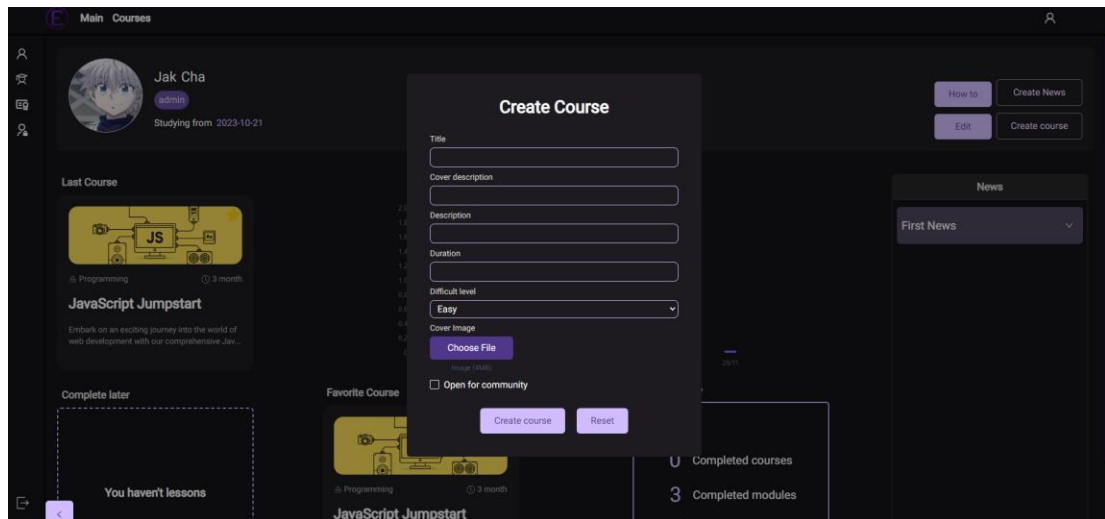


Рисунок 4.14 Форма створення курсу

Форма має поля для назви курсу, опису курсу, можна вказати тривалість курсу та рівень тяжкості, також форма має галку чи потрібно щоб користувачі його бачили чи ні.

Сторінка курсів користувача має вкладку "Subscribed" де він може дивитись свої підписки на курси, якщо користувач має роль "teacher" або "admin" він має додатково "My courses" де він може дивитись свої створені курси. Вкладки продемонстровано на рисунку 4.15

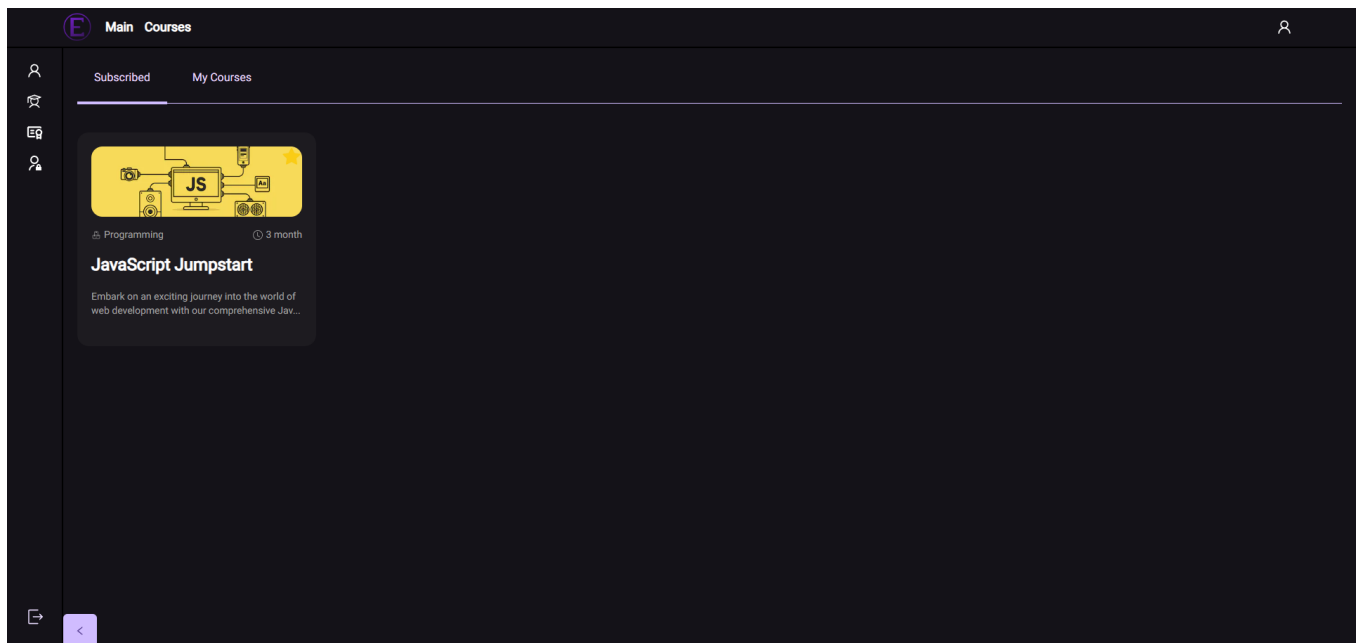


Рисунок 4.15 Сторінка з курсами користувача

Сторінка курсу має опис курсу, також користувач має кнопку додати до своїх курсів і якщо він автор то він може змінювати його поля такі як назва, картинка, тривалість опис на відкривати та зачиняти для інших людей.

Курс має 3 вкладки:

- About;
- Course content;

– Reviews.

Вкладка "About" має опис курсу, "Course content" має модулі курсу, "Reviews" тут можна поділитись своїми враженнями або дати пораду по покращенню курсу.

Course content має модулі де автор може додавати робити видимими/невидимими модулі для всіх, удаляти модулі міняти їх місцями за допомогою Drag`n`Drop. "Drag'n'Drop" (перетягування та відпускання) - це техніка інтеракції з користувачем, яка дозволяє переміщати об'єкти на екрані за допомогою миші чи екранного сенсора. Користувач може "захопити" об'єкт, натискати на нього кнопку миші (або торкатися екрану) та перетягувати його в інше місце на екрані, а потім відпустити кнопку миші (або палець на екрані), щоб відпустити об'єкт на новому місці. Автор також може перейменувати модулі а учень може помічати чи пройшов він модуль чи ні. Модуль продемонстровано на рисункку 4.16

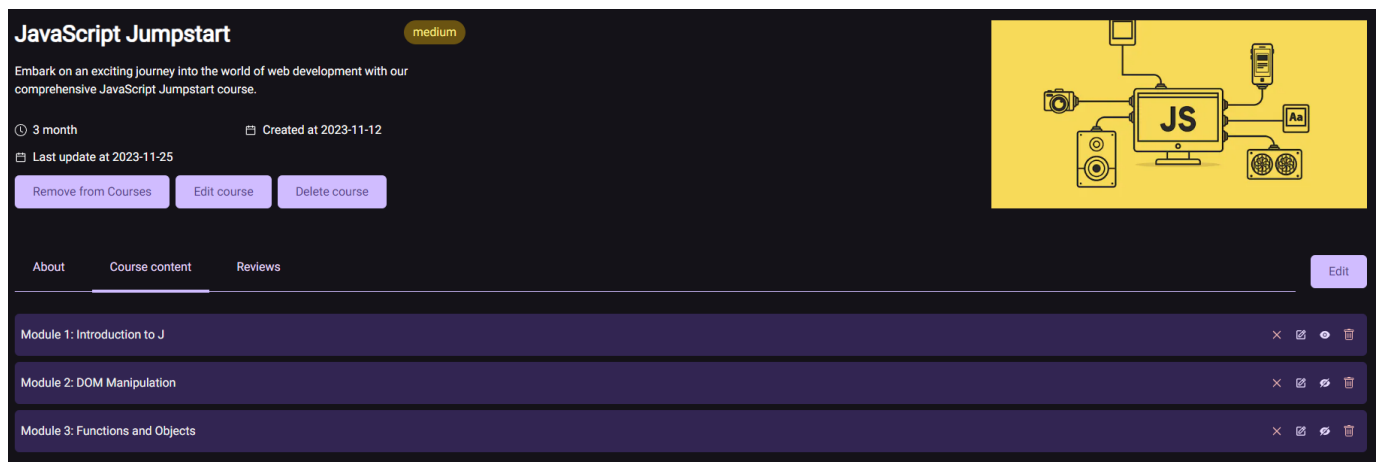


Рисунок 4.16 Модулі курсу

Вкладка "Reviews" з відгуками користувачів продемонстровано на риунку 4.17

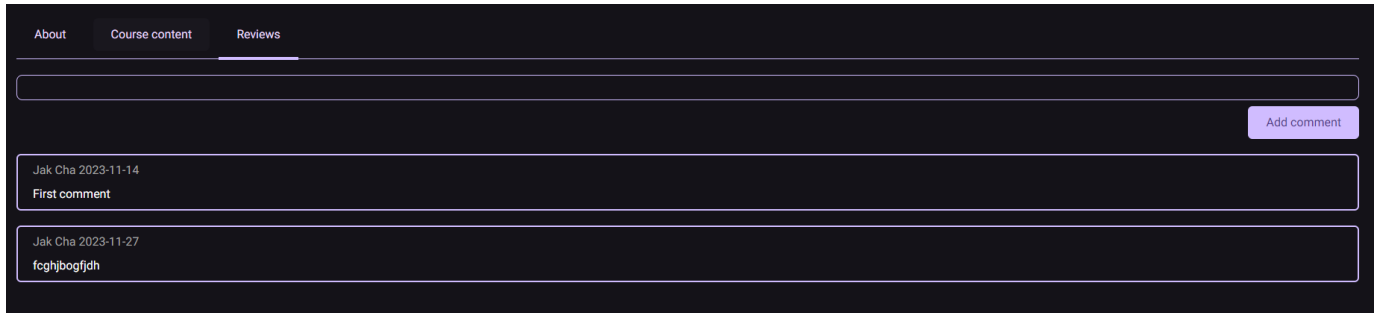


Рисунок 4.17 Вкладка з відгуками

Сторінка модуля має уроки які учень буде проходити уроки можуть бути як ігрові так і текстові, сторінка продумонстрована на рисунку 4.18



Рисунок 4.18 Сторінка модуля

Якщо автор хоче створити урок йому потрібно натиснути "Edit" та кнопку "Add Lesson" після чого буде форма для створення уроку (рисунок 4.19).

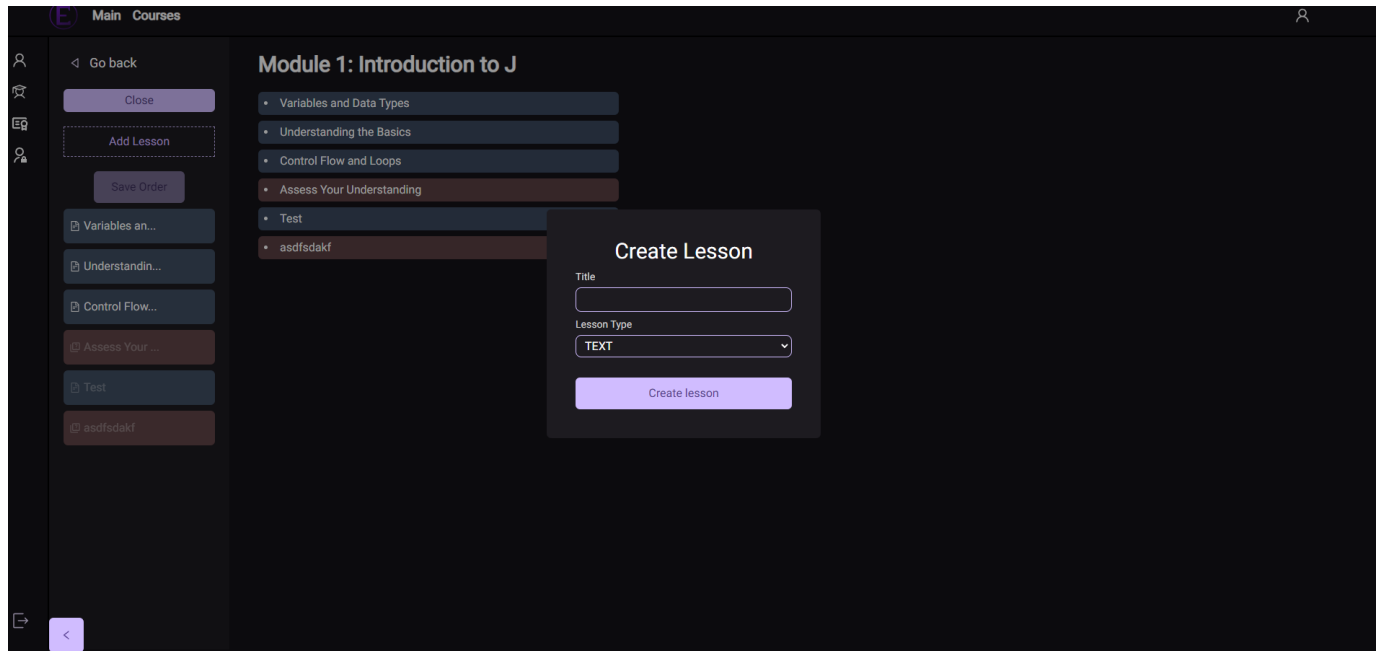


Рисунок 4.19 Форма створення уроку

Автор може обирати між Text або Quiz та написати назву урока після чого урок створиться. Уроки також підтримують Drag`n`Drop томму автор може їх переміщувати як завгодно.

Якщо це автор то він також може міняти видимість уроків або видаляти їх. Для всіх користувачів доступні кнопки для позначення чи виконаний урок чи ні та можна додати урок в контейнер "Complete later" рисунок 4.20

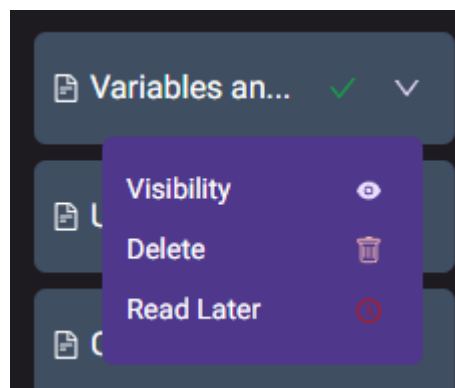


Рисунок 4.20 Маніпуляції з уроком

Кожен тип уроку має своє позначення для текстового це синій для ігрових червоні кольори.

Сторінка уроку для автора містить 2 кнопки для зміни типу уроку або назви, якщо автор хоче змінити тип уроку його буде попереджено про те що весь контент видалиться (рисунок 4.21).

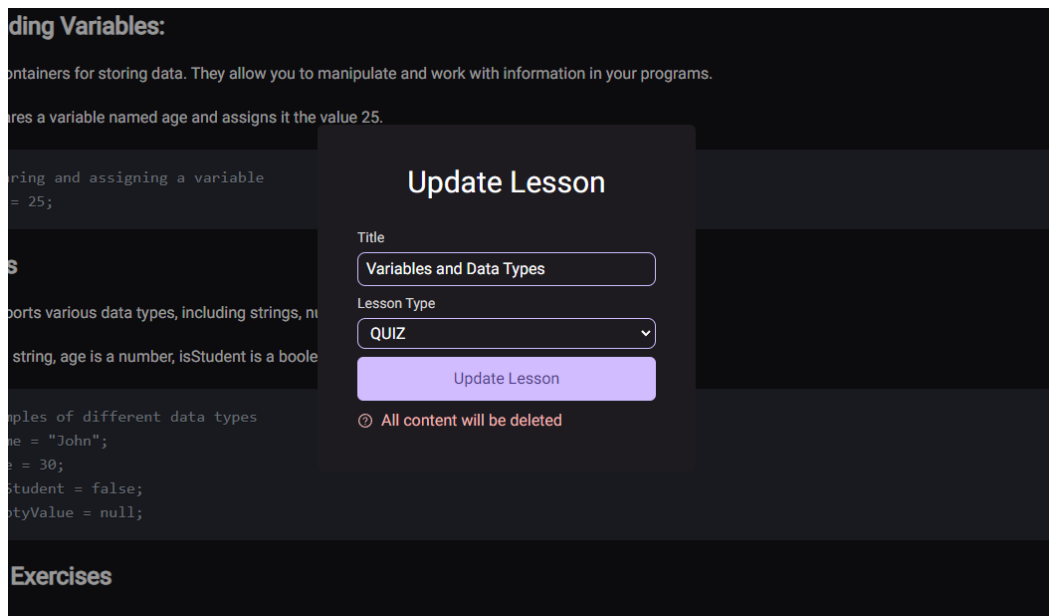


Рисунок 4.21 Зміна інформації уроку

Інша кнопка відкриває форму де автор може додавати блоки. Типи блоків залежать від типу уроку для текстового уроку:

- Text block;
- Image block;
- Code block;
- Video Block.

Для ігрового уроку:

- Question Answer;
- Alphabet Soup Quiz;

- Answer with fixed letters;
- True False;
- Question Two More Answer.

Сторінка прогресу має прогрес по всім курсам які учень додав собі. Сторінка продемонстрована на рисунку 4.22. На сторінці є картки з загальною інформацією по курсу чи виконаний він чи ні та його назва.

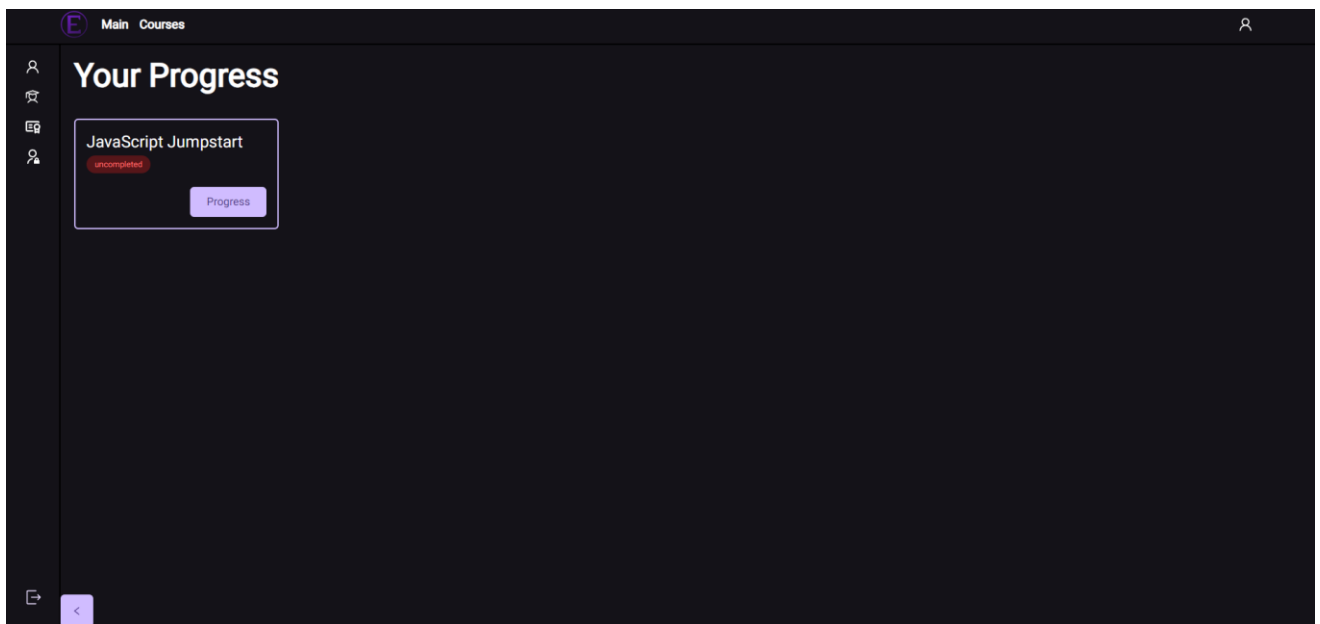


Рисунок 4.22 Сторінка прогресу

При переході на карту учень отримує повну інформацію по своєму курсу скільки відсотків пройдено свій середній бал скільки виконано уроків та які модулі виконано а які ні (рисунок 4.23).

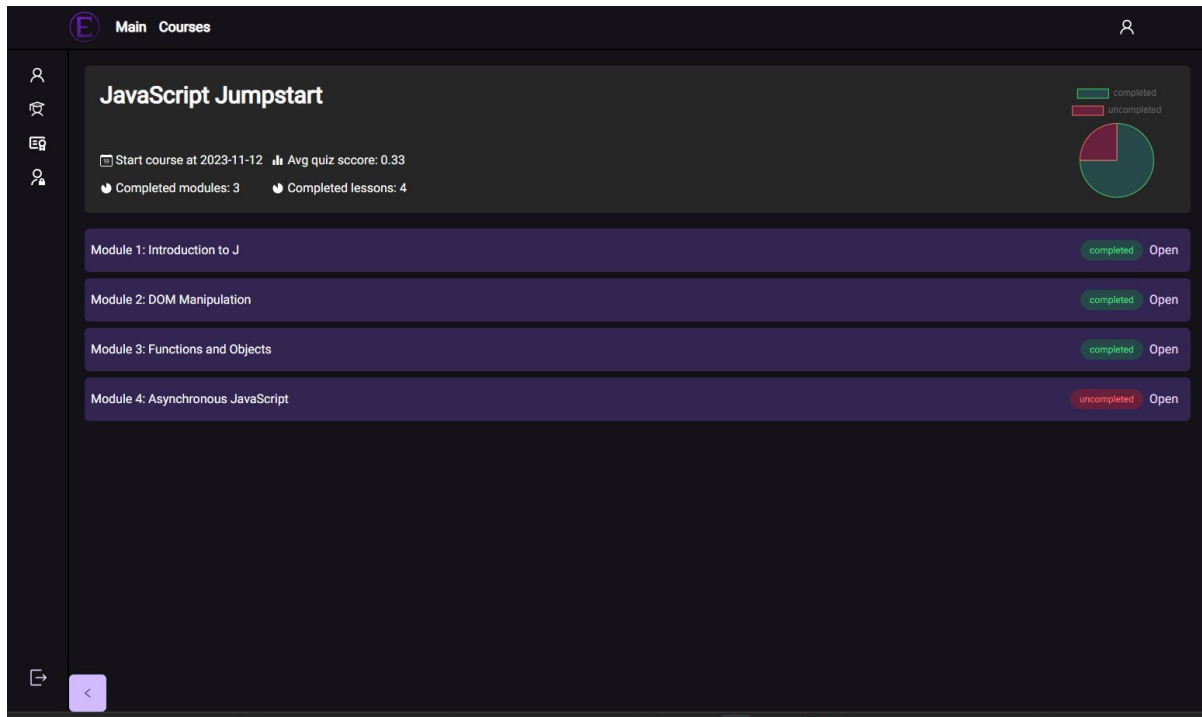
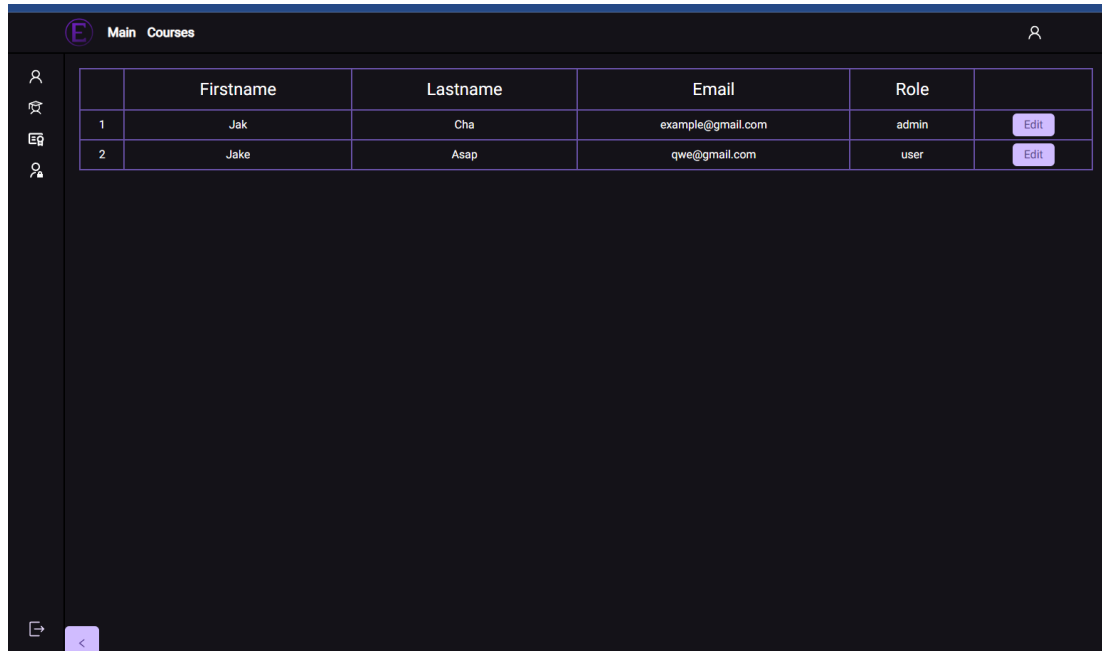


Рисунок 4.23 Загальна інформація успіхів

Сторінка Адмін панелі містить таблицю з усіма користувачами (рисунок 4.26). Кожен рядок має загальну інформацію про користувача та, якщо у користувача не відповідне ім'я або прізвище, адмін може легко його змінити. Якщо користувачів багато і їх потрібно відмальовувати і щоб не було великої навантаження на сервер при запиті на 100 користувачів було зроблено пагінацію. Пагінація - це метод організації великої кількості даних шляхом розділення їх на окремі сторінки. Це часто використовується в веб-розробці, де на одній сторінці не завжди можна відобразити всі дані. Замість цього дані розділяються на блоки або сторінки, і користувач може переходити між цими сторінками. Отже, коли користувач прокручує сторінку вниз, інші користувачі поступово з'являються перед ним.



The screenshot shows an admin panel interface. At the top, there is a navigation bar with a logo 'E' and the text 'Main Courses'. Below this is a table with the following data:

	Firstname	Lastname	Email	Role	
1	Jak	Cha	example@gmail.com	admin	Edit
2	Jake	Asap	qwe@gmail.com	user	Edit

The interface also includes a sidebar on the left with various icons and a bottom navigation bar with a back arrow.

Рисунок 4.26 Адмін панель

Для не зареєстрованих користувачів є дві сторінки: основна та сторінка з усіма курсами на ресурсі продемонстровано на рисунку 4.27 та рисунку 4.28. Основна сторінка відображає загальну інформацію по додатку з лозунгом. Сторінка курсів має карточки з курсами на, які користувач може натиснути для попереднього перегляду курсу, якщо він не аутентифікований його перекине на сторінку аутенфікації.

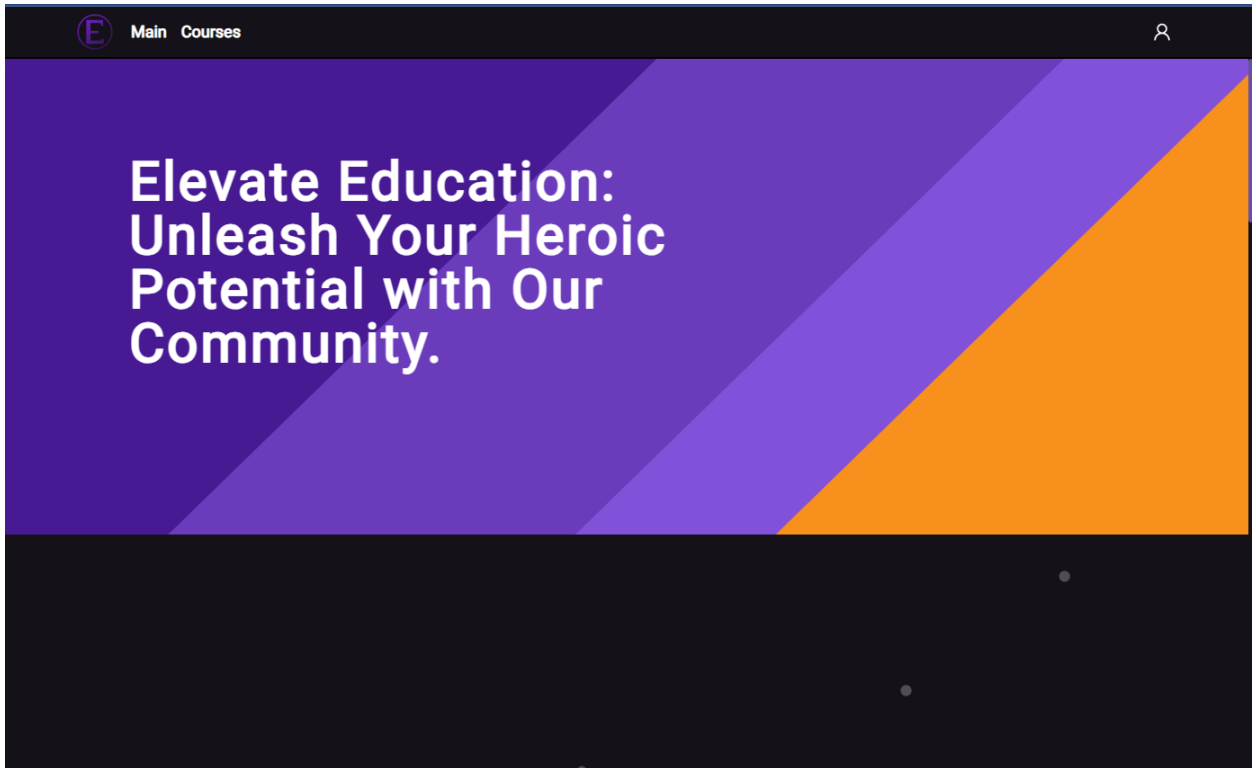


Рисунок 4.27 Основна сторінка додатку

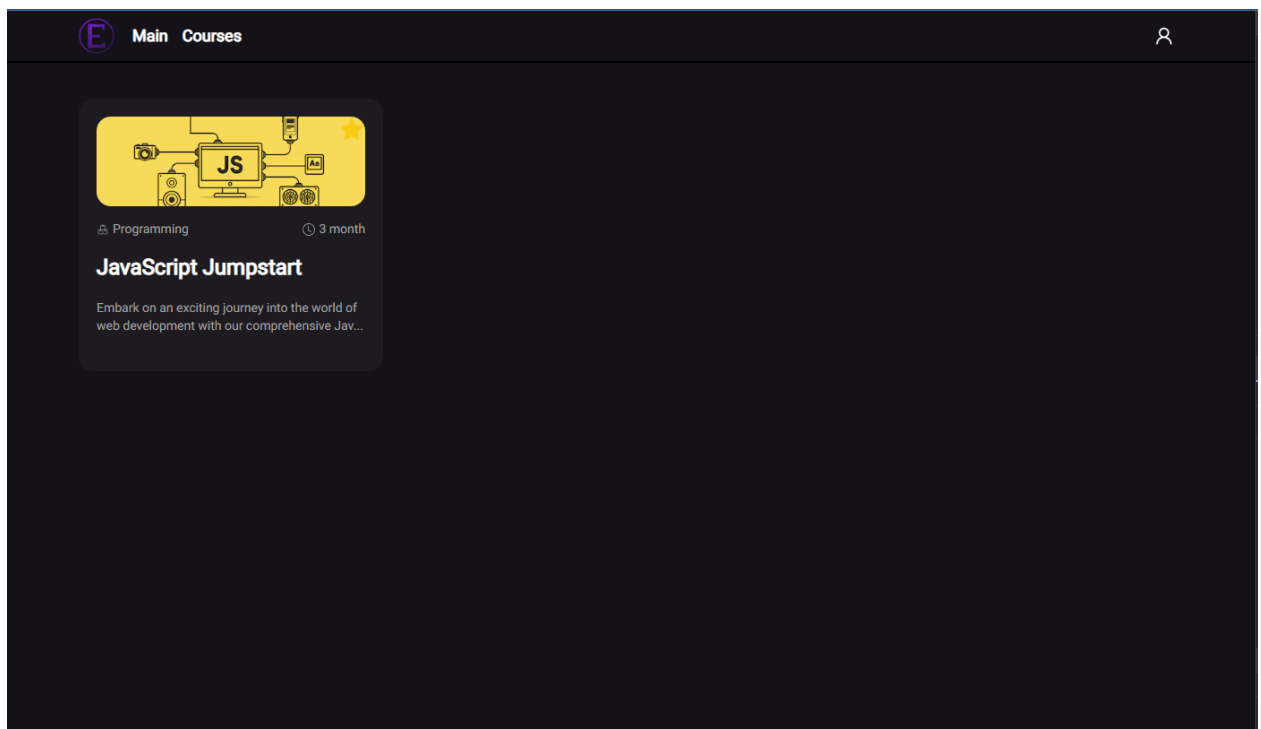


Рисунок 4.28 Сторінка всіх курсів додатку

4.3 Розробка ігрової частини додатку

Перевірка знань учнів через квізи для самоконтролю є ключовим елементом освітнього процесу. Цей підхід не лише надає можливість учням систематизувати свої знання, а й сприяє активному засвоєнню матеріалу. Навіть найбільш вмотивований учень може забути або пропустити деякі деталі, але завдяки регулярним квізам вони мають можливість вдосконалити свої знання та виявити слабкі місця. Крім того, цей процес сприяє розвитку навичок самоконтролю та відповідальності.

Для проєкту розроблено поки 5 типів питань:

- Question Answer;
- Alphabet Soup Quiz;
- Answer with fixed letters;
- True False;
- Question Two More Answer.

Question Answer — ця гра ставить учасникам конкретне питання, на яке вони повинні обрати одну правильну відповідь серед інших відповідей. Це може бути ефективним способом перевірити конкретні знання або розуміння концепцій (рисунок 4.29).

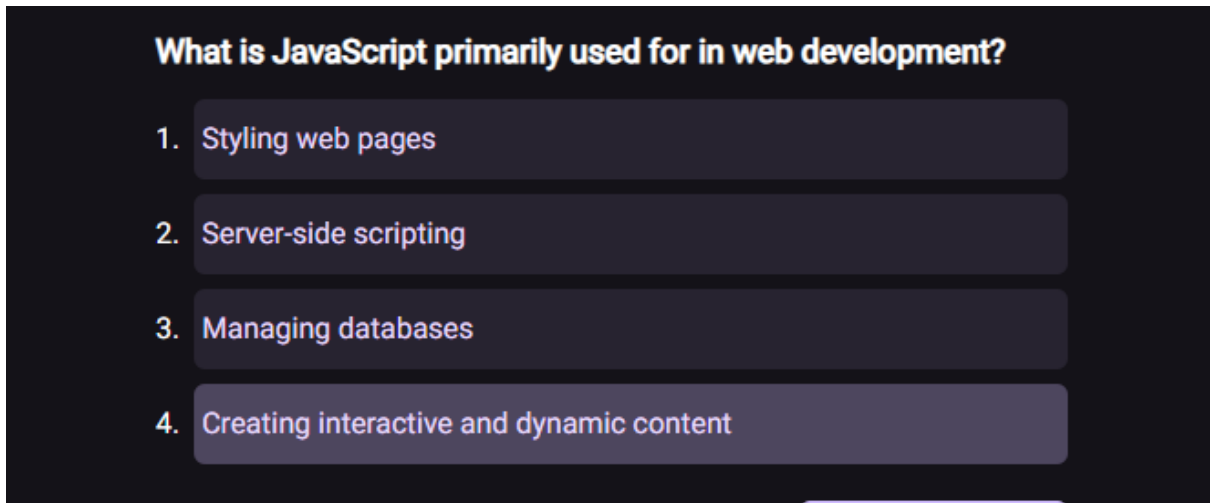


Рисунок 4.29 Гра "Question Answer"

Alphabet Soup Quiz — учаснику представлено перемішані літери, і йому потрібно впоратися з головоломкою, щоб визначити правильну відповідь (рисунок 4.30).

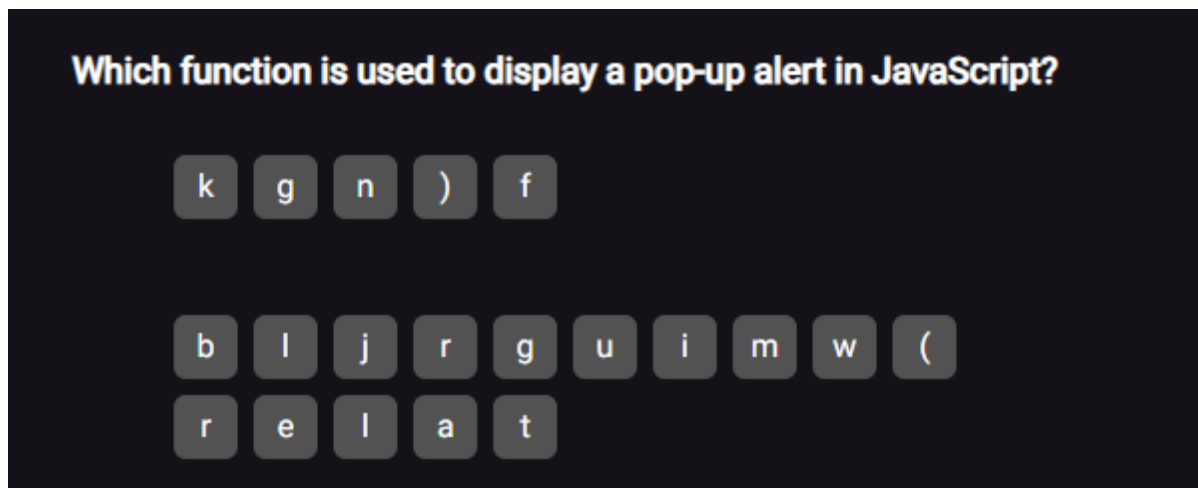


Рисунок 4.30 Гра "Alphabet Soup Quiz"

Answer with fixed letters, у цій грі може бути фіксована кількість комірок, які вказують кількість літер у слові (рисунок 4.31). Учень повинен правильно заповнити

ці комірки, враховуючи відоме слово. Це може допомагати в навчанні правопису та розширенні словникового запасу.

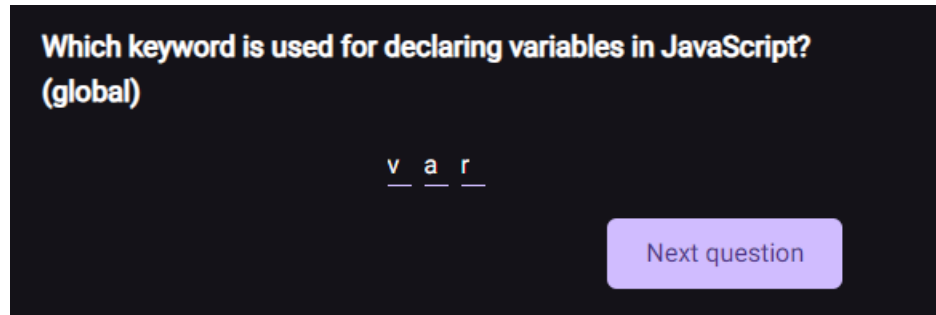


Рисунок 4.31 Гра "Answer with fixed letters"

True False тут висувається твердження, і учасник повинен визначити, чи воно вірне чи ні (рисунок 4.32). Це простий, але ефективний спосіб для швидкої перевірки розуміння концепцій.

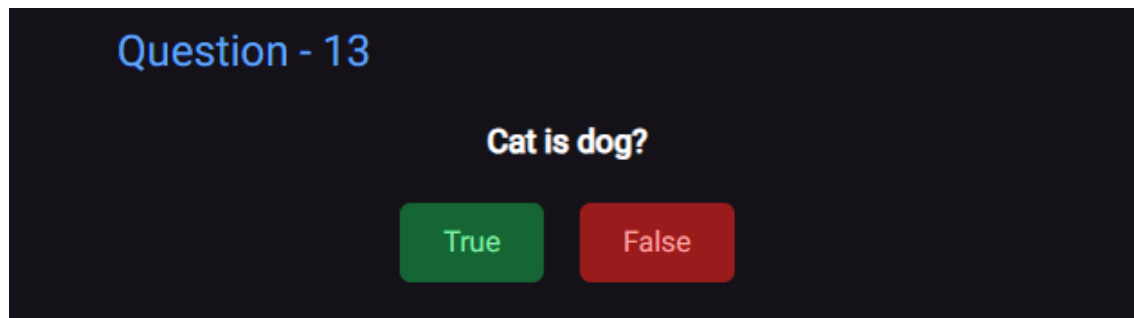


Рисунок 4.32 Гра "True False"

Question Two More Answer, у цій грі ставиться питання, і учень має не одну, а кілька відповідей на вибір (рисунок 4.33). Це розширює можливі варіантів відповідей.

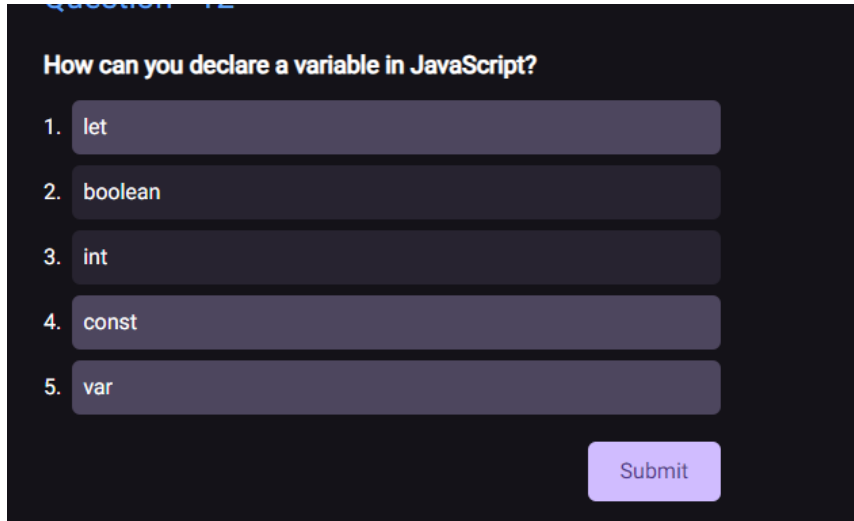


Рисунок 4.33 Гра "Question Two More Answer"

Після завершення квізу студент отримує результати своїх відповідей, він може побачити кількість балів, не правильних відповідей та точність відповідей (рисунки 4.34). Оцінка автоматично відправляється в статистику після проходження, але якщо учня не влаштовує оцінка він може пройти знову тест. Після першого проходження урок помічається як пройдений.

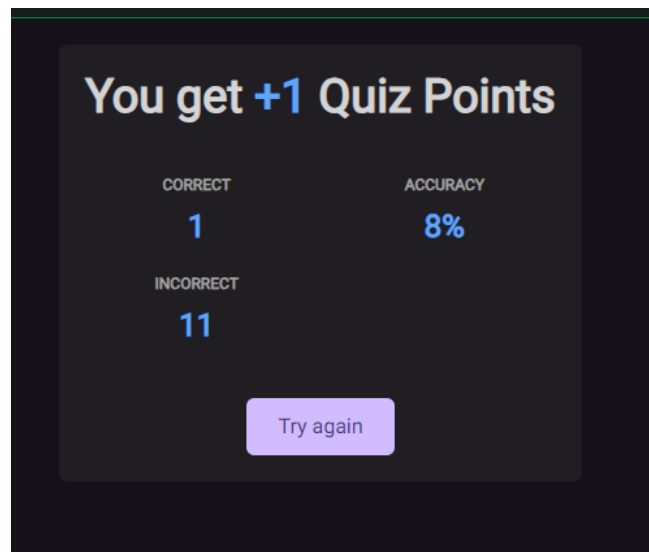
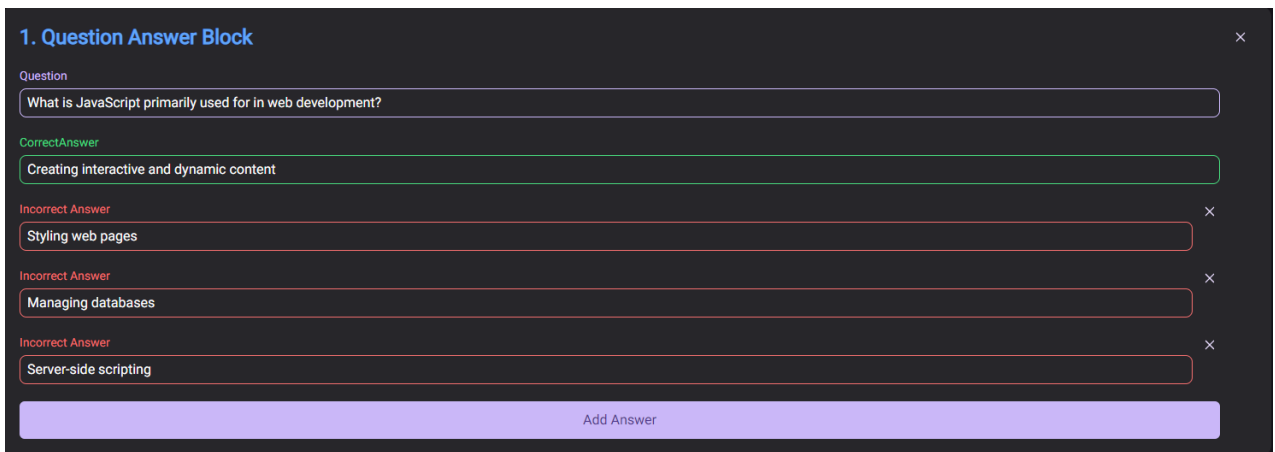


Рисунок 4.34 Результати учня

Для створення цих квізів автор курсу має форми для кожного типу уроку, кількість форм не обмежена.

Question Answer складається з поля для питання, правильної відповіді та полів які автор може додавати для неправильних відповідей (рисунок 4.35).



1. Question Answer Block

Question

What is JavaScript primarily used for in web development?

Correct Answer

Creating interactive and dynamic content

Incorrect Answer

Styling web pages

Incorrect Answer

Managing databases

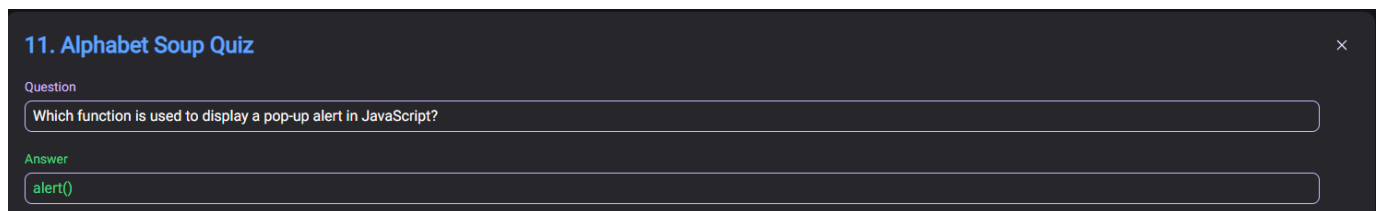
Incorrect Answer

Server-side scripting

Add Answer

Рисунок 4.35 Форма гри "Question Answer"

Alphabet Soup Quiz має поле для питання та для відповіді (рисунок 4.36). Для генерації випадкових літер написано невеликий скрипт він створює масив та заповнює його випадковими літерами у випадку, якщо вихідний масив має менше 20 елементів. Спочатку створюється масив із літер правильного слова, а потім у циклі додаються випадкові літери до масиву до тих пір, поки його довжина не стане рівною або більшою за 20. На завершення літери перемішуються та повертається новий масив.



11. Alphabet Soup Quiz

Question

Which function is used to display a pop-up alert in JavaScript?

Answer

alert()

Рисунок 4.36 Форма гри "Alphabet Soup Quiz"

Answer with fixed letters має поле для питання та відповіді (рисунок 4.37). Для подачі учню в правильному вигляді слово розбивається на масив окремих літер і потім генеруються форми вводу після чого учень вписує літери в кожну форму, слово склеюється і перевіряється з результатом на коректність.

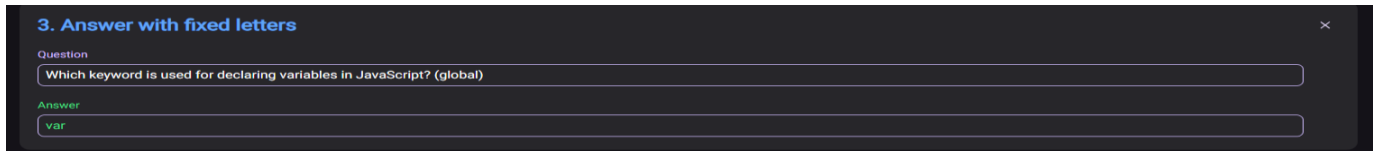


Рисунок 4.37 Форма гри "Answer with fixed letters"

True False має поле для питання та "radio button" для вибору правильної відповіді (рисунок 4.38).

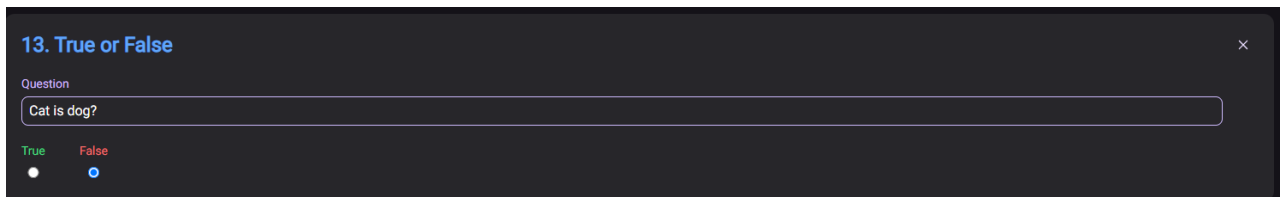


Рисунок 4.38 Форма гри "True False"

Question Two More Answer схожий на Question Answer але тут можна обирати кількість правильних відповідей (рисунок 4.39).

12. Question Two More Answer Block

Question

How can you declare a variable in JavaScript?

Correct Answer

var

Correct Answer

let

Correct Answer

const

Add Correct Answer

Incorrect Answer

int

Incorrect Answer

boolean

Add Incorrect Answer

Рисунок 4.39 Форма гри "Question Two More Answer"

4.4 Реалізація статистики успішності користувача

Оцінка власного прогресу — це ключовий елемент успішного навчання. Збір статистики дозволяє користувачам відстежувати свої сильні та слабкі сторони. Це не лише дозволяє зосередитися на сильних сторонах, але і виявляти слабкі місця для подальшого вдосконалення. Такий підхід сприяє не лише підвищенню якості навчання, а й створенню стимулюючого середовища для розвитку.

Тому для користувача створено сторінку де він може дивитись свої успіхи по кожному курсу. Сторінка з успіхом має "header" (рисунок 4.40), де є основна інформація по курсу така як:

- старт курсу;
- кількість виконаних модулів;
- середня оцінка по квізам;
- кількість виконаних уроків;
- діаграма на скільки пройдено курс у відсотках.



Рисунок 4.40 "Header" сторінки статистики

Також розроблено відображення інформації по кожному модулю, зображено на рисунку 4.41, він виглядає як "pop-up menu", тобто коли користувач відкриває інформацію по модулю, відкривається діаграма з успішністю учню по кожному уроку, (рисунок 4.42). На модулі зображено візуально чи пройдено його чи ні. Якщо модуль не має уроків користувач побачить про це повідомлення (рисунок 4.43).



Рисунок 4.41 Модуль курсу

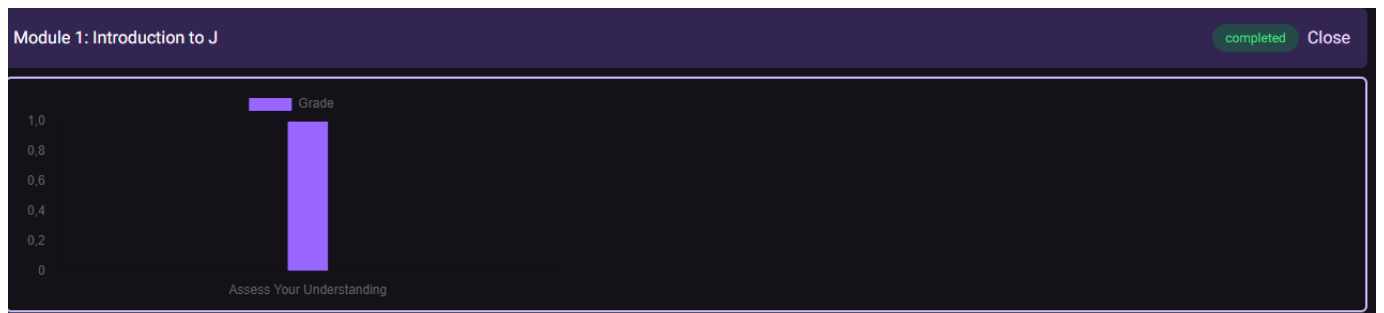


Рисунок 4.42 Діаграма успішності користувача

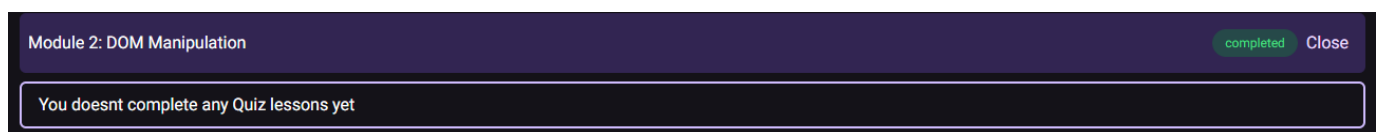


Рисунок 4.43 Повідомлення про відсутність уроків-квізів

4.5 Тестування коду

Тестування коду дуже важливе для запобігання непередбачуваним помилкам та забезпечення стабільності додатку. У веб-розробці, зокрема з використанням React, для цього використовуються інструменти, такі як Jest та React Testing Library.

Jest — це фреймворк для тестування JavaScript, який використовується для написання та запуску тестів [10]. Він надає можливості для створення unit-тестів, integration-тестів та snapshot-тестів. Jest має вбудовану підтримку для асинхронного коду, що робить його ідеальним вибором для тестування React додатків.

React Testing Library — це бібліотека, яка допомагає тестувати React компоненти з фокусом на їхню взаємодію з реальним DOM [11]. Замість того, щоб тестувати внутрішню реалізацію компонента, React Testing Library спрямована на тестування компонентів так, як їх бачить користувач. Це робить тести більш реалістичними та орієнтованими на користувача.

Тестування коду у React додатках відіграє критичну роль у забезпеченні його надійності та ефективності. Основні типи тестів включають unit-тести, які перевіряють окремі частини коду, та integration-тести, які тестують взаємодію між компонентами.

Unit-тести фокусуються на перевірці найменших частин коду, таких як функції чи компоненти. Вони дозволяють переконатися, що кожна одиниця коду працює так, як очікується, і виявляють можливі помилки на ранніх етапах розробки.

Integration-тести перевіряють, як різні частини додатку взаємодіють між собою. Наприклад, чи коректно працюють компоненти разом або чи правильно обробляються дані, які передаються між ними.

Під час тестування React додатків важливо не тільки перевіряти правильність виконання коду, а й враховувати взаємодію з реальними умовами використання. React

Testing Library спрямована саме на це, надаючи інструменти для тестування компонентів так, як їх бачить користувач.

В проєкті всі критичні файли коду покриті unit-тестами для запобігання помилок, для прикладу тести для елемента "Button" продемонстровано на рисунку 4.44. Даний код має два тести. Перший тест бере кнопку по її id та перевіряє чи відрендерилася кнопка в DOM-дереві. Другий тест перевіряє чи клікабельна кнопка і чи натискається вона вірно.

```
import {fireEvent, screen} from "@testing-library/react";
import {Button} from "../index";
import {ButtonThemes} from "./Button";
import {ComponentRender} from "../../config/test/test-utils";

describe("Button", () => {
  test("Button render in document", () => {
    ComponentRender(<Button theme={ButtonThemes.FILLED}>test</Button>);

    const buttonEl = screen.getByTestId("button");

    expect(buttonEl).toBeInTheDocument();
  });
  test("Button click event", () => {
    const mockClickHandler = jest.fn();
    ComponentRender(<Button theme={ButtonThemes.FILLED} onClick={mockClickHandler}>test</Button>);

    const buttonEl = screen.getByTestId("button");

    fireEvent.click(buttonEl);

    expect(mockClickHandler).toHaveBeenCalledTimes(1);
  });
});
```

Рисунок 4.44 Код для тестування елемента "Button"

Тести — це не тільки інструмент для виявлення помилок, але й засіб документації, який допомагає зрозуміти, як слід використовувати код. Таким чином,

тестування стає важливою складовою розробки, сприяючи покращенню якості та стабільності програмного забезпечення.

4.6 Хостинг проєкту

Щоб кожен користувач мав доступ до додатку його захощено на платформі Vercel. Vercel — це платформа для хостингу веб-додатків, яка є ідеальною для Next.js [12]. Вона пропонує безліч функцій, які полегшують розробку, розгортання та масштабування Next.js-додатків, включаючи:

- автоматичне виявлення Next.js;
- оптимізацію для продуктивності;
- підтримку HTTPS;
- швидке розгортання;
- глобальне масштабування.

Щоб захостити проєкт Next.js на Vercel, потрібно:

- зареєструватися в Vercel;
- підключити свій репозиторій Git;
- вибрати свій проєкт Next.js;
- натиснути кнопку "Deploy".

Vercel автоматично збудує проєкт Next.js і розгорне його на глобальній мережі серверів. Це займе лише кілька хвилин.

На веб-сайті Vercel потрібно натиснути кнопку "Get Started". Ввести свою електронну адресу та пароль, а потім натисніть кнопку "Sign up". Потім потрібно натиснути кнопку "Add New" (рисунок 4.46). Далі з'являється меню де потрібно вибрати "Project", після чого з'являться всі проєкти які є на github (рисунок 4.46).

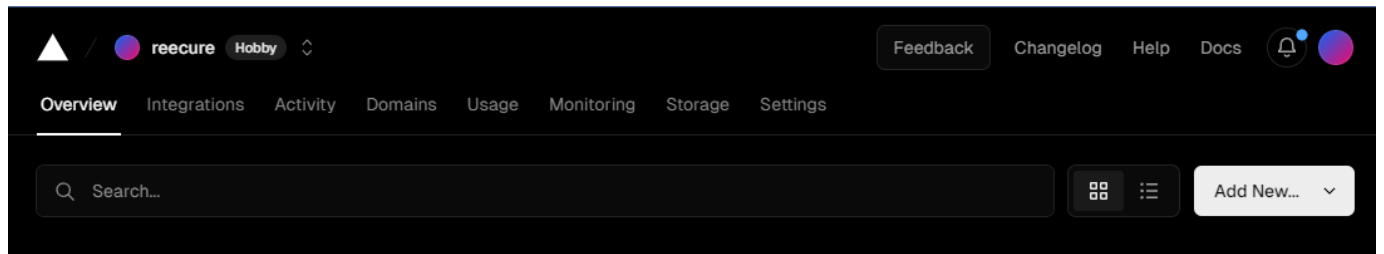


Рисунок 4.45 Інтерфейс Vercel

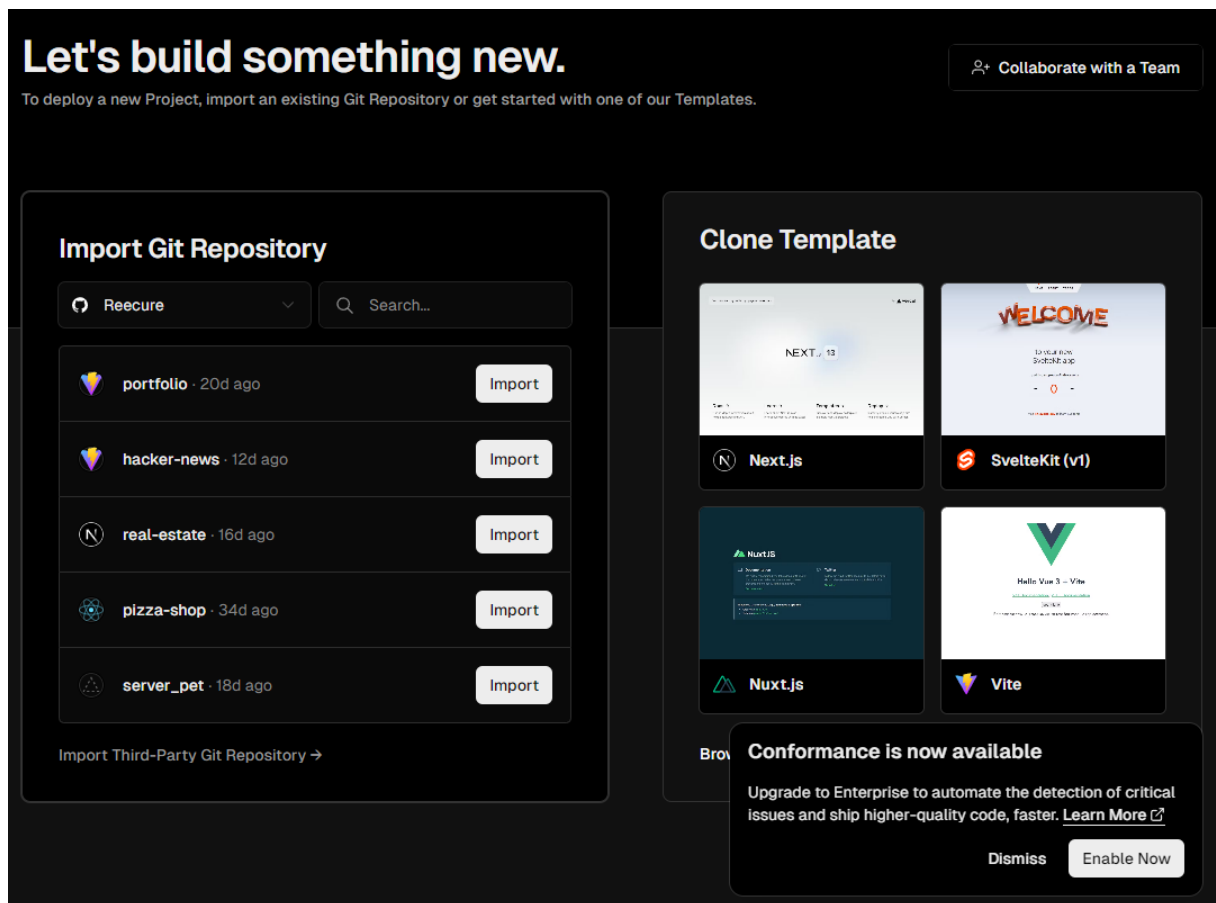


Рисунок 4.46 Інтерфейс Vercel

Після вибору проекту потрібно натиснути кнопку "Deploy". Vercel автоматично збудує проект Next.js і розгорне його на глобальній мережі серверів. Це займе лише кілька хвилин. Після того, як проект буде розгорнуто, його можна переглянути за URL-адресою, яка відображається в розділі "Deploys" на панелі керування Vercel.

Висноски до розділу

В ході розробки електронної навчальної системи з використанням методик гейміфікації з використанням tRPC допомогло ефективно структурувати та реалізовувати серверну частину системи. Це дозволить забезпечити надійність та ефективність обміну даними між сервером та клієнтами. Розробка ігрової частини для взаємодії з користувачем, додає інтерактивність та зацікавленість до навчального процесу. Реалізація статистики успішності користувача дозволяє ефективно відстежувати та візуалізувати прогрес учнів, забезпечуючи їм зручний засіб моніторингу своїх досягнень.

Вдосконалено навички по написанню тестів за допомогою Jest та React Testing Library та отримано нові вміння хостингу проєктів на Vercel.

ВИСНОВКИ

Розроблений e-learning web-додаток є високофункціональним та універсальним інструментом для навчання на різних рівнях. В ході роботи було:

1. Розроблено систему на основі сучасних технологій, яка відповідає останнім тенденціям в сфері e-learning.

2. Проведено аналіз різноманітних популярних освітніх платформ.

3. Створено робоче середовище для розробки з можливістю легкої розширюваності додатку.

4. За допомогою CSS-стилів було розроблено адаптивний дизайн. Він забезпечує доступність з будь-якого пристрою, роблячи навчання зручним для всіх користувачів.

5. Впроваджено різноманітні ігри для взаємодії з користувачем. Основна мета додатку полягає в створенні ефективних та захоплюючих навчальних курсів за допомогою методик гейміфікації гейміфікації. Це забезпечує не лише передачу інформації, але й стимулює учнів до активного навчання та досягнення поставлених цілей.

6. Додаток має функції, такі як моніторинг статистики, можливість додавання курсів до особистої бібліотеки та улюблених, а також гнучкість у створенні ігор, роблять додаток зручним та функціональним інструментом для вчителів та учнів.

7. Розроблено сторінку користувача, де він отримує швидкий доступ до своєї тижневої статистики, новин та швидкого доступу до курсів.

8. Розроблений за допомогою мови JavaScript додаток розташований на сервері, що дозволяє кожному користувачу з доступом до Інтернету навчатись в будь-якому місці географічно.

9. Додаток планує розширюватись у таких напрямках, як додавання стрімінгових

уроків, навчальних та перевірочних тестів, відзнак та чатів, зроблять e-learning web-додаток ще більш привабливим для користувачів, дозволяючи їм взаємодіяти та залучатися в навчальний процес більш активно. Це говорить про актуальність та перспективність роботи в контексті вдосконалення освітніх технологій.

Отже, під час роботи було розроблено високофункціональний та універсальний e-learning web-додаток, що використовує методи гейміфікації для створення захопливих навчальних курсів. Цей додаток є кроком в майбутнє освіти, пропонуючи ефективний та цікавий процес навчання для широкого спектру вікових груп – від дітей до дорослих.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 E-Learning Today [Електронний ресурс] : Режим доступу – <https://etraintoday.com/blog/e-learning-today/> – Назва з екрану.
- 2 5 Reasons Why Online Learning is the Future of Education in 2023 [Електронний ресурс] : Режим доступу – <https://www.educations.com/articles-and-advice/5-reasons-online-learning-is-future-of-education-17146> – Назва з екрану.
- 3 Gamification in Education | Definition, Benefits & Strategies [Електронний ресурс] : Режим доступу – <https://study.com/academy/lesson/what-is-gamification-in-education-definition-research-strategies.html#:~:text=Gamification%20in%20education%20is%20the,one%20big%20first%20person%20game> – Назва з екрану.
- 4 Заріцька С.І. Завдання розвитку освітніх технологій в контексті євроінтеграції / Заріцька С.І., Литвиненко Н.І. / Інформаційні технології і автоматизація – 2022. — 20 - 21 жовтня 2022 р. — 2022 — С. 111-113.
- 5 End-to-end typesafe APIs. [Електронний ресурс] : Режим доступу – <https://trpc.io/> – Назва з екрану.
- 6 Next-generation Node.js and TypeScript ORM. [Електронний ресурс] : Режим доступу – <https://www.prisma.io/> – Назва з екрану.
- 7 For the next generation of intelligent applications. [Електронний ресурс] : Режим доступу – <https://www.mongodb.com/> – Назва з екрану.
- 8 The React Framework for the Web. [Електронний ресурс] : Режим доступу – <https://nextjs.org/> – Назва з екрану.
- 9 Server-side Rendering (SSR). [Електронний ресурс] : Режим доступу – <https://nextjs.org/docs/pages/building-your-application/rendering/server-side-rendering> – Назва з екрану.

10 React The library for web and native user interfaces [Электронный ресурс] : Режим доступа – <https://react.dev/> – Назва з екрану.

11 Jest is a delightful JavaScript Testing Framework with a focus on simplicity. [Электронный ресурс] : Режим доступа – <https://jestjs.io/> – Назва з екрану.

12 Testing Library Simple and complete testing utilities that encourage good testing practices [Электронный ресурс] : Режим доступа – <https://testing-library.com/> – Назва з екрану.

13 Vercel Documentation [Электронный ресурс] : Режим доступа – <https://vercel.com/docs> – Назва з екрану.

14 JavaScript. The Definitive Guide. Master the World's Most-Used Programming Language 7th Edition – David Flanagan 2020.

15 RESTful Web APIs: Services for a Changing World 1st Edition. Leonard Richardson – Leonard Richardson, 2013.

16 The SEO Battlefield: Winning Strategies for Search Marketing Programs 1st Edition – nne Ahola War 2017.

17 NextAuth.js Authentication for Next.js [Электронный ресурс] : Режим доступа – <https://next-auth.js.org/> – Назва з екрану.

ДОДАТОК А

Випускна магістерська
робота з теми :
«РОЗРОБКА WEB-ОРІЄНТОВАНОЇ E-LEARNING
СИСТЕМИ З ВИКОРИСТАННЯМ МЕТОДИК
ГЕЙМІФІКАЦІЇ»

Виконав: студент групи ІКК Пир'ян Ю.В.
Керівник: к.т.н., доц. Т.І. Григор'єва

Мета: Розробка сучасної e-learning системи, яка підвищить мотивацію до навчання та ефективність засвоєння знань.

Завдання:

- 1) Розробити веб-інтерфейс системи, який буде простим у використанні та зручним для користувачів.
- 2) Запровадити в систему методи гейміфікації, які зроблять навчання більш цікавим та захоплюючим.
- 3) Налаштувати систему так, щоб вона відповідала потребам конкретних користувачів.

Для чого методики гейміфікації?

Гейміфікація - це використання елементів ігрового дизайну в неігрових контекстах.

Вона може підвищити мотивацію, зацікавленість і залученість користувачів.

Гейміфікація може бути використана для підвищення ефективності навчання, зробивши його більш цікавим та захоплюючим.

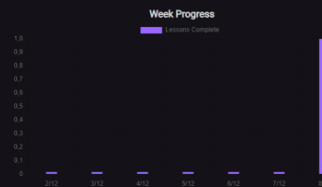
Технології

- Веб-технології: HTML, CSS, JavaScript, TypeScript
- Система керування базами даних: MongoDB
- Серверна частина: [tRPC](#)



Статистика користувача

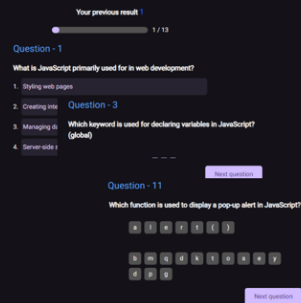
- Система надає користувачам детальну статистику їхнього прогресу в навчанні.
- Статистика відображається у вигляді радіальних діаграм та гістограм.
- Користувач може переглядати статистику за тиждень та по кожному курсу, модулю та уроку.



Ігрова частина додатку

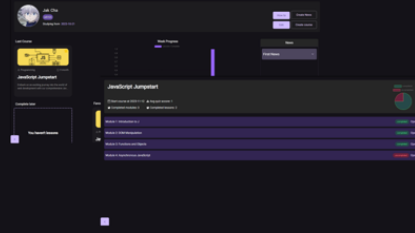
Додаток має різні варіації ігр:

- Question with one Answer
- Alphabet Soup Quiz
- Answer with fixed letters
- True False
- Question with Two More Answer



Який функціонал вийшов

- **Акаунт користувача:** Користувач може створювати та керувати своїм обліковим записом. У обліковому записі користувача зберігається його прогрес у навчанні, статистика успішності та інші дані.
- **Статистика успішності:** Система надає користувачам статистику їхньої успішності в навчанні. Ця статистика може використовуватися для відстеження прогресу користувача та мотивації його до подальшого навчання.
- **Ігри:** Система пропонує користувачам 5 різних видів ігор, які можна використовувати для вивчення матеріалу. Ігри допомагають зробити навчання більш цікавим та захоплюючим.
- **Адміністративна панель:** Адміністратор системи може використовувати адміністративну панель для управління даними користувачів.



Майбутні плани

До додатку є плани додати наступні функції:

- **Стрімінгові уроки:** користувачі зможуть дивитися онлайн-уроки від викладачів.
- **Нові види ігрових запитань:** користувачі зможуть розгадувати ребуси, головоломки та інші ігри, щоб вивчати матеріал.
- **Відзнаки:** користувачі зможуть отримувати відзнаки за досягнення, щоб підвищити свою мотивацію.
- **Чат для користувачів:** користувачі зможуть спілкуватися один з одним та обмінюватися думками про навчання.

Висновки

- Система була розроблена на основі сучасних технологій і відповідає останнім тенденціям в e-learning.
- Система має простий і зручний веб-інтерфейс, який легко освоїти.
- В систему впроваджено методи гейміфікації, які роблять навчання більш цікавим та захоплюючим.
- Гейміфікація є ефективним методом підвищення мотивації до навчання та ефективності засвоєння знань.
- Впровадження гейміфікації в e-learning системи може призвести до значного покращення результатів навчання.

Дякую за увагу

ДОДАТОК Б

```
generator client {  
  provider      = "prisma-client-js"  
  previewFeatures = ["referentialActions"]  
}
```

```
datasource db {  
  provider = "mongodb"  
  url      = env("DATABASE_URL")  
}
```

```
type LessonsLessonContent {  
  blocks LessonsLessonContentBlocks[]  
}
```

```
type ModuleLessons {  
  id          String  
  lesson_id  String  
  is_visible Boolean  
  title      String  
  lesson_type String  
  order      Int  
  author_id  String  
}
```

```
type CourseModules {  
  id          String  
  module_id  String  
  is_visible Boolean  
  title      String
```

```
order    Int
author_id String
}
```

```
type LessonsLessonContentBlocks {
  answer    Json?
  code      String?
  correctAnswer Json?
  id        String
  paragraphs LessonsLessonContentBlocksParagraphs[]
  listItems LessonsLessonContentBlocksListItems[]
  question  String?
  src       String?
  title     String?
  type      String
  url       String?
}
```

```
type LessonsLessonContentBlocksAnswer {
  otherAnswer String
}
```

```
type LessonsLessonContentBlocksParagraphs {
  id String
  text String
}
```

```
type LessonsLessonContentBlocksListItems {
  id String
  item String
}
```



```
type UsersCoursesProgress {  
  course_id      String  
  is_completed   Boolean  
  course_name    String  
  start_course   String?  
  complete_percentage Int?  
}
```

```
type UsersLessonsProgress {  
  lesson_name  String  
  complete_date DateTime @db.Date  
  read_later   Boolean?  
  is_completed Boolean  
  lessonType   String  
  module_id    String  
  lesson_id    String  
  quizScore    Int  
}
```

```
type UsersModulesProgress {  
  real_module_id String?  
  module_name    String  
  is_completed   Boolean  
  module_id      String  
  course_id      String  
}
```

```
model course_categories {  
  id String @id @default(auto()) @map("_id") @db.ObjectId
```

```

    @@map("course-categories")
}

```

```

model news {
    id      String  @id @default(auto()) @map("_id") @db.ObjectId
    author_id String  @db.ObjectId
    title   String
    description String
    creation_date DateTime @db.Date
}

```

```

model reviews {
    id      String  @id @default(auto()) @map("_id") @db.ObjectId
    course_id String  @unique
    user_id  String
    text     String
    creation_date DateTime @db.Date
}

```

```

model courses {
    id      String  @id @default(auto()) @map("_id") @db.ObjectId
    author_id  String  @db.ObjectId
    category_id String
    cover_description String
    modules    CourseModules[]
    cover_image String
    creation_date  DateTime @db.Date
    update_date    DateTime? @db.Date
    description    String
    difficulty_level String
    duration       String
}

```

```

is_visible    Boolean
rating       Int
title        String
}

```

```

model events_and_activities {
  id String @id @default(auto()) @map("_id") @db.ObjectId

  @@map("events-and-activities")
}

```

```

model grades_and_progress {
  id String @id @default(auto()) @map("_id") @db.ObjectId

  @@map("grades-and-progress")
}

```

```

model lessons {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  author_id   String          @db.ObjectId
  lesson_content LessonsLessonContent
  lesson_type String
  module_id   String          @db.ObjectId
  order       Int
  title       String
}

```

```

model modules {
  id      String      @id @default(auto()) @map("_id") @db.ObjectId
  course_id String      @map("course_id") @db.ObjectId
  author_id String      @db.ObjectId
  lessons ModuleLessons[]
  is_visible Boolean
}

```

```
    title String
}
model users {
    id String @id @default(auto()) @map("_id") @db.ObjectId
    avatar String
    courses String[]
    courses_progress UsersCoursesProgress[]
    email String @unique
    firstname String
    is_new_user Boolean
    lastname String
    favorite_course String?
    last_course String?
    lessons_progress UsersLessonsProgress[]
    modules_progress UsersModulesProgress[]
    password String
    registration_date DateTime @db.Date
    role String
}
```

ДОДАТОК В

```
import NextAuth from "next-auth";
import CredentialsProvider from "next-auth/providers/credentials";
import {PrismaClient} from "@prisma/client";
import bcrypt from "bcrypt";

const prisma = new PrismaClient();

export default NextAuth({
  providers: [
    CredentialsProvider({
      name: "Credentials",
      credentials: {
        // Define the expected fields here
        email: {label: "Email", type: "email"},
        password: {label: "Password", type: "password"},
      },
      async authorize(credentials, req) {
        const user = await prisma.users.findUnique({
          where: {email: credentials?.email},
        });
        if (
          user && (await bcrypt.compare(
            credentials !== undefined ? credentials.password : "",
            user.password,
          ))
        ) {
          const {id, role, email} = user;
          return {id, role, email};
        }
      }
    })
  ]
});
```

```
        throw new Error("Invalid Email or Password");
    },
    }),
],

pages: {
    signIn: "/auth/signin",
    signOut: "/",
    error: "/auth/error",
},
session: {
    strategy: "jwt",
},
callbacks: {
    async jwt({token, user}) {
        return {...token, ...user};
    },
    async session({session, token}) {
        session.user = token;
        return session;
    },
},
secret: "SECRETqwekasdkop1238iqwjasdjygl6ywge1",
});
```