МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК, ФІЗИКИ ТА МАТЕМАТИКИ
КАФЕДРА КОМ`ЮТЕРНИХ НАУК ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

РОЗРОБЛЕННЯ ОСВІТНЬОЇ ПЛАТФОРМИ ДЛЯ ЗОВНІШНІХ УЧАСНИКІВ ОСВІТНЬОГО ПРОЦЕСУ

Виконав: студент 12-231м курсу
Спеціальності:122
Комп'ютерні науки
Освітньо-професійної програми:
«Комп'ютерні науки»
Калитюк Іван Павлович
Науковий керівник: доктор фізико-математичних наук,
професор Співаковський Олександр Володимирович
Рецезент: Сенчишен Д.О., Middle Backend Developer, IT компанія DataArt

Херсон – 2023

ЗМІСТ

ВСТУП **3**
РОЗДІЛ 1 ФОРМУВАННЯ ВИМОГ ДО ВЕБ-АКАДЕМІЙ **6**
1.1 Вимоги до дизайну 6
1.2 Вимоги методів надання освіти 7
РОЗДІЛ 2 АНАЛІЗ ПОПУЛЯРНИХ ВЕБ-АКАДЕМІЙ **10**
2.1 Аналіз сайту Khan Academy 10
2.2 Аналіз Udacity 12
2.3 Аналіз Edx 14
2.4 Аналіз веб-академії CourseEra 16
2.5 Аналіз веб-академії Codecademy 18
Висновки 20
РОЗДІЛ 3. РОЗРОБЛЕННЯ ВЕБ-АКАДЕМІЇ **22**
3.1 Розроблення дизайну 22
3.2 Верстка дизайну 29
4.3 Розробка бекенду 31
3.4 Розробка фронтенду 35
3.5 Рефакторинг коду та виправлення помилок 40
ВИСНОВОК **41**
ВИКОРИСТАНІ ДЖЕРЕЛА **42**
ДОДАТКИ **46**

ВСТУП

В останнє десятиліття сфера освіти переживає значні зміни, спричинені ростом популярності веб-академій та онлайн-курсів. Coursera, edX, Udacity, Khan Academy та багато інших стають більш привабливим варіантом для студентів ніж університети. В порівнянні з традиційною освітою вони мають багато переваг:

- Доступність. Одна з основних причин популярності веб-академій. Люди з різних кутків світу можуть отримати доступ до високоякісної освіти без необхідності переїзду в інше місто або країну. Це особливо важливо для тих, хто не має достатньо фінансів та часу на відвідання освітніх закладів в очному або змішаному режимі.
- Гнучкість в розкладі. Веб-академії надають студентам можливість навчатися у зручний для них час. Це дозволяє працюючим студентам, батькам та тим, хто має зайнятий графік поєднувати навчання з іншими обов’язками.
- Широкий вибір курсів. Веб-академії пропонують багатий вибір курсів з різних галузей знань. Від онлайн-курсів з програмування та IT до освіти в галузі мистецтва та гуманітарних наук. Особливо важлива точковість в навчанні, тому що в університетах часто викладають ті предмети, які не потрібні в майбутній професії, а важливі знання подають обмежено.
- Інтерактивність та різноманітність курсів. Веб-академії надають різноманітні освітні ресурси, включаючи відео-лекції, тести, форуми для обговорення завдання. Це робить процес навчання більш інтерактивним та захоплюючим.
- Сертифікація та кар’єрні можливості. Багато веб-академій надають сертифікати і навіть дипломи. Більшість з них визнаються на ринку праці, це допомагає студентам покращити свої кар’єрні перспективи та конкурентоспроможність.

Як ми бачимо освітні веб-платформи можуть бути вигіднішими ніж традиційна освіта. Проте основні переваги університетів та шкіл полягають саме в сертифікації та професіоналізмі викладачів:

- Сертифікація. Проблемою деяких веб-академій є невизнання їх сертифікатів. Коли на співбесіді при працевлаштуванні між собою конкурують дві людини з однаковим рівнем знать, саме людина з дипломом університету, а не сертифікатом веб-академії частіше отримує посаду. Освіта в університеті вважається більш серйозною.
- Професіоналізм викладачів. Викладачі мають спеціальну педагогічну освіту на відміну від більшості веб-академій, при цьому нерідко університети залучають видатних вчених та професіоналів своєї справи.

І університети і веб-академії мають свої переваги, нашою задачею буде поєднати ці два типи здобуття знань та створити на її основі університетську веб-академію.

Об’єкт дослідження: веб-академії та освітні платформи для зовнішніх учасників навчального процесу

Предмет дослідження: розроблення платформи надання освітніх послуг академії університету

Мета: розробити веб-академію KSU Academy для забезпечення освітніх потреб зовнішніх користувачів освітніх послуг університету

Задачі:

1) Визначити вимоги до веб-академій
2) Проаналізувати існуючі веб-академії
3) Створити план розроблення сайту KSU Academy
4) Розробити сайт KSU Academy

Актуальність теми: полягає в створенні веб-академії на базі університету.

Структура роботи

Робота складається зі вступу, трьох розділів, висновку, списку використаних джерел та додатків.

Перший розділ містить інформацію про критерії аналізу веб-академій. В другому розділі ми аналізуємо веб-академії на основі критеріїв, визначених в першому розділі. Третій розділ має інформацію про розроблення сайту KSU Academy.

РОЗДІЛ 1 ФОРМУВАННЯ ВИМОГ ДО ВЕБ-АКАДЕМІЙ

1.1 Вимоги до дизайну

Дизайн є важливою та невід'ємною частиною успішної реалізації освітньої цілі веб-академій, тому що навчання проходить саме на сайтах. Правильно підібраний дизайн впливає на:

- Зручність користувача. Добре підібраний дизайн робить сайт зручним та привабливим для користувачів. Він створює інтуїтивно зрозумілий інтерфейс, що дозволяє легко знайти необхідну інформацію та використовувати освітні ресурси. Зручність користувача сприяє покращенню досвіду навчання та утримання студентів на платформі.
- Мотивація та залучення. Естетично приємний та функціональний дизайн може стимулювати мотивацію студентів. Інтерактивні елементи, креативне використання кольорів та мультимедійних матеріалів можуть зробити навчання більш привабливим та захоплюючим.
- Престижність. Так як весь освітній процес проходить саме на веб-сайті, його дизайн прямо впливає на престижність та допомагає створювати бренд навчального закладу чи платформи. Це важливо для залучення студентів та створення довіри до якості освітніх програм, що сприяє зміцненню іміджу.
- Адаптивність. Сучасні веб-академії повинні забезпечувати адаптивність для різних пристроїв та екранів. Адаптивний дизайн дозволяє студентам навчатись на планшетах, смартфонах та комп'ютерах, що важливо в умовах мобільності та різноманітності пристроїв.
- Навігація та структура. Добре підібраний дизайн допомагає оптимізувати навігацію на сайті. Чітка структура, легкий доступ до розділів курсів та матеріалів, а також інтуїтивно зрозумілі меню роблять процес навчання більш ефективним та приємним.[14][15]

Для інформаційних сайтів дизайнерами як правило використовується стиль матеріалізму або мінімалізму. Це допомагає зробити акцент саме на контенті сайту, а не на елементах дизайну. Також ці стилі впливають на легкість навігації, швидкому завантаженню сторінок, інклюзивності та легкості в розробці адаптивності. Не рекомендовано використовувати типографічний та мультимедійний стилі, вони надто перенасичені елементами, що може відволікати від навчання. Кольори, особливо в мінімалістичному та матеріалістичному дизайні мають вирішальне значення. Їх необхідно підбирати так, щоб вони пасували одно до одного і знаходились в одній кольоровій гаммі.[27]

Проте варто врахувати, що ці критерії щодо інформаційних сайтів є лише рекомендаціями до дизайнерів з точки зору дослідників. Веб-академії на сьогодні перебувають у великій конкуренції, тому їх розробники приймають рішення додавати особливості до дизайну, щоб виділятись серед інших. Тому оцінити доцільність використання певних стилів або дизайнерських рішень можна лише після власного ознайомлення.

У висновку, стиль веб-академії може бути різним, проте правильний дизайн освітньої платформи обов’язково повинен мати:

- Адаптивність дизайну
- Інтуїтивність навігаційної структури
- Сучасність дизайну
- Акцент саме на контенті сайту
- Дизайн має створювати бренд сайту

1.2 Вимоги методів надання освіти

Нам як розробникам при аналізі методів надання освіти необхідно звертати увагу саме на методи, а не на якість навчальних матеріалів. Веб-академії знаходяться в остійній конкуренції між собою, тому для

збільшення притоку користувачів вони можуть використовувати нетипові або експериментальні методи навчання. Проте завжди зберігається основа кожного методу – ефективне надання освіти. Під цим поняттям розуміється відношення якості освіти до часу, витраченого на його освоєння.

Веб-академії використовують відмінні від традиційного навчання методи. Наприклад використання відео або текстових лекцій буде мало ефективне при очному навчанні в школах або університетах. Проте в веб-академіях вони є найпопулярнішими та найефективнішими методами. [39]

Одним із найефективніших методів навчання є принцип маленьких кроків. Розділ, поділений на безліч маленьких лекцій, освоюється краще і швидше, ніж коли він поділений на великі за обсягом лекції. Проте варто врахувати, що не кожен матеріал можна подавати маленькими лекціями, нам як розробникам варто зробити зручним створення та освоєння лекцій будь якого методу, який буде заданий викладачем.[34]

Найважливішою частиною будь-якого методу надання освіти є практика. Навіть невеликі тести після проходження розділу є важливими для засвоєння інформації. Вважається що при читанні лекції студент поглинає але не використовує інформацію.[35][36][37]

Однією з суттєвих переваг традиційного навчання є прямий контакт з викладачем. Для ефективного засвоєння матеріалу студент у веб-академіях також повинен мати доступ до консультацій. Вони завжди проходять в онлайн форматі, тому їх ефективність можна порівняти з дистанційним навчанням.

Загалом можна виділити основні критерії до методів надання освіти:

- Загальна ефективність методу
- Можливість однаково ефективно використовувати різні методи

- Обов’язкова практична частина
- Доступ до консультації з викладачем. Відсутність зв’язку з викладачем може вплинути на ефективність навчання, так як в кожній темі може виникнути запитання, які не пояснюються в лекціях.

РОЗДІЛ 2 АНАЛІЗ ПОПУЛЯРНИХ ВЕБ-АКАДЕМІЙ

2.1 Аналіз сайту Khan Academy

Це сайт, який надає можливості студентам безкоштовно навчатись базовим предметам. Освіта проходить за планом: викладач створює клас, куди додає студентів та видає їм додані розробниками навчальні матеріали. [11]

Дизайн Khan Academy виконано в стилі мінімалізму. Він виконаний на високому рівні але з недоліками. По-перше, дизайн не створює бренду сайту, він виглядає так само, як і інші мінімалістичні сайти. По-друге, використано невдале поєднання кольорів, вони сильно відрізняються одне від одного по тональності. По-третє, використано багато елементів дизайну, які відрізняються одне від одного. Деякі елементи контенту в дизайнерському плані виглядають також погано, хоча в плані функціональності виконані добре. На сторінках лекцій багато зайвої інформації, яка створює заплутаність, більшість з неї дублюється на цій же сторінці. Навігація виконана на високому рівні. Сайт також добре виглядає на смартфонах та планшетах. (Рис 2.1, Рис 2.2)

Академія має три типи користувача. Студенти можуть записуватись в класи, де будуть проходити навчання під керівництвом викладача. Викладач може лише слідкувати за освітнім процесом, проте не створювати нові лекції. Третій тип користувача – батьки, можуть спостерігати за прогресом студента.

В навчальному матеріалі використовується метод маленьких кроків. Всі розділи поділені на невеликі за обсягом теми, а самі теми на окремі лекції. Лекції поділяють на два типи – текстові та відео з транскрипцією. Після кожної лекції йде практична частина, яка складається з тестів. Також після кожної теми є практична частина, а в кінці проходження розділу студента чекає фінальна робота. (Рис. 2.2, Рис 2.3)

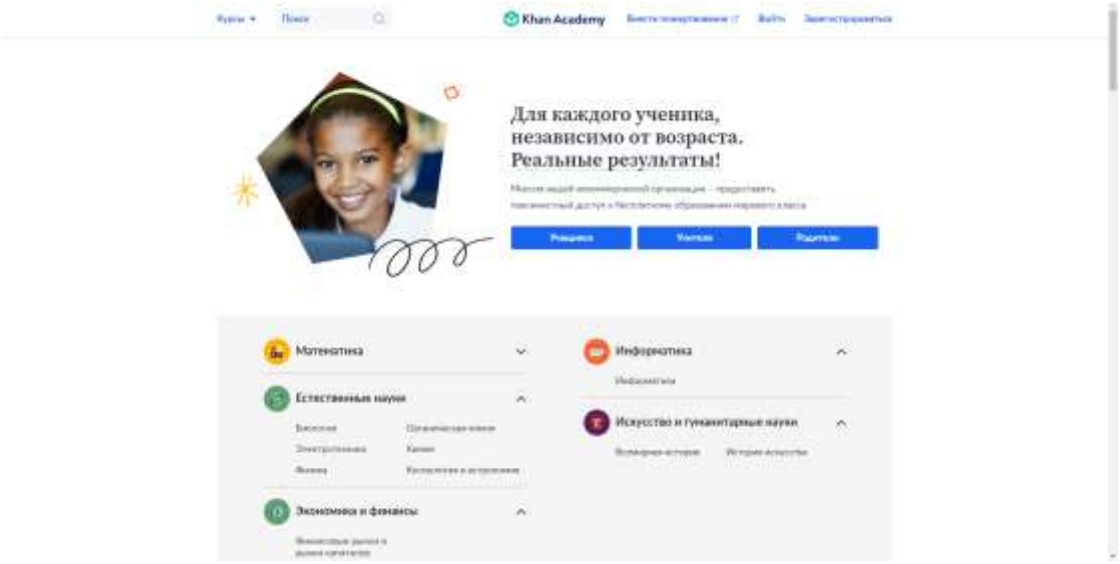

Рис. 2.1 Головна сторінка сайту Khan Academy

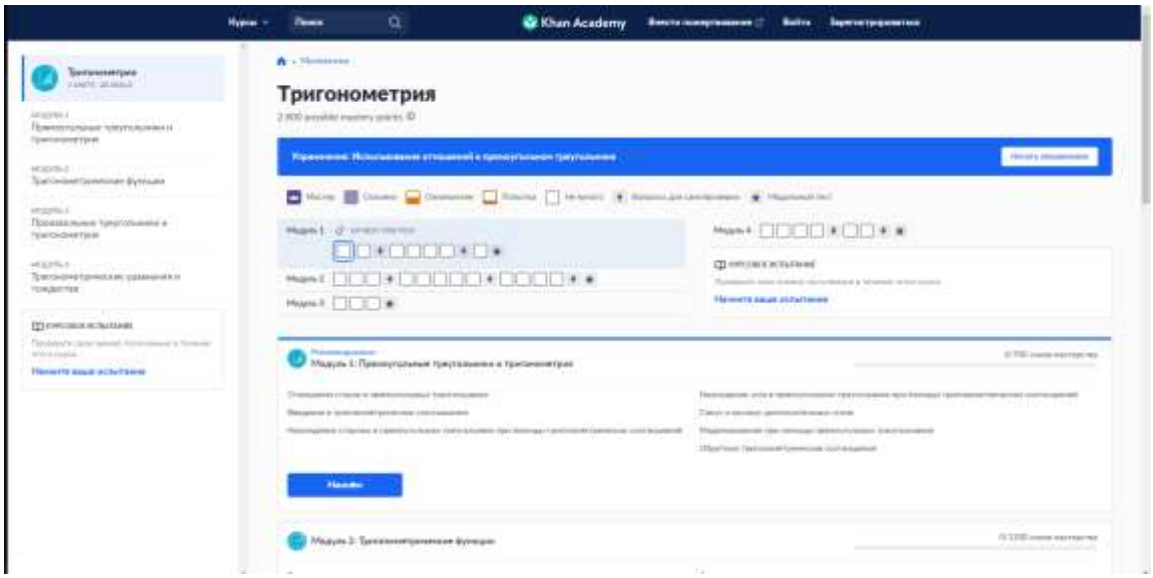

Рис. 2.2 Сторінка розділу навчального матеріалу сайту Khan Academy

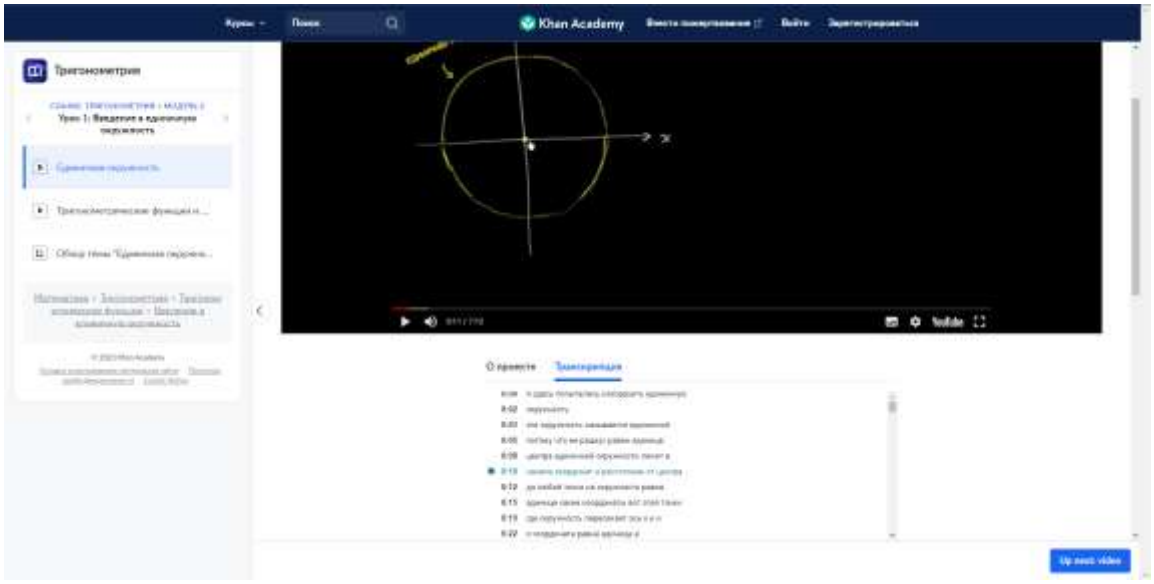

Рис. 2.3 Вигляд відео-лекції сайту Khan Academy

Висновок: з точки зору дизайну сайт має недоліки, він не створює бренду, не має особливостей, по ньому важко зрозуміти, що ти знаходишся саме на сайті Khan Academy. Сайт є адаптивним, навігація виконана на високому рівні як і загальна інтуїтивність. Методи вкладання матеріалу є ефективними – є текстові та відео лекції, виконується принцип маленьких кроків. Після кожної лекції та теми є опитування, що також підвищує ефективність. В цілому цей сайт через свою структуру і спосіб викладання матеріалу через класи не підходить нам, проте його освітні методи варто додати до нашої академії.

2.2 Аналіз Udacity

Веб-академія Udacity – одна з найпоширеніших академій. Користувачів приваблює досвідченість авторів лекцій. Самих лекцій мало, проте за відгуками студентів, вони виконані на високому рівні. Аудиторія сайту – студенти університетів, або люди, що хочуть опанувати нові знання для зміни професії.

Дизайн сайту Udacity виконаний в мінімалістичному стилі, з поєднанням елементів матеріалістичного стилю. Загалом дизайн виглядає несучасним, більшість сторінок має порожній вигляд. Палітра кольорів вибрана добре, вона навіть використана для фотографій на сайті. Дизайн загалом має недоліки, проте створює відчуття бренду і запам’ятовується. Навігація на сайті виконана на високому рівні. Дизайн сторінок лекцій не відволікає від навчання. Сайт є адаптивним та добре виглядає на смартфонах та планшетах. (Рис. 2.4)

Сайт має як безкоштовні так і платні матеріали. Всі матеріали створені викладачами університетів. Весь контент поділений на предмети, котрі поділені на теми, які поділені на лекції. На сайті присутні лише відео-лекції, більшість з них є короткими (менше трьох хвилин) проте є і довгі (більше десяти хвилин). Після проходження теми користувач повинен пройти тест. Особливістю також можна виділити

зворотній зв’язок з викладачем, який ведеться стосовно кожної лекції окремо. (Рис. 2.5)

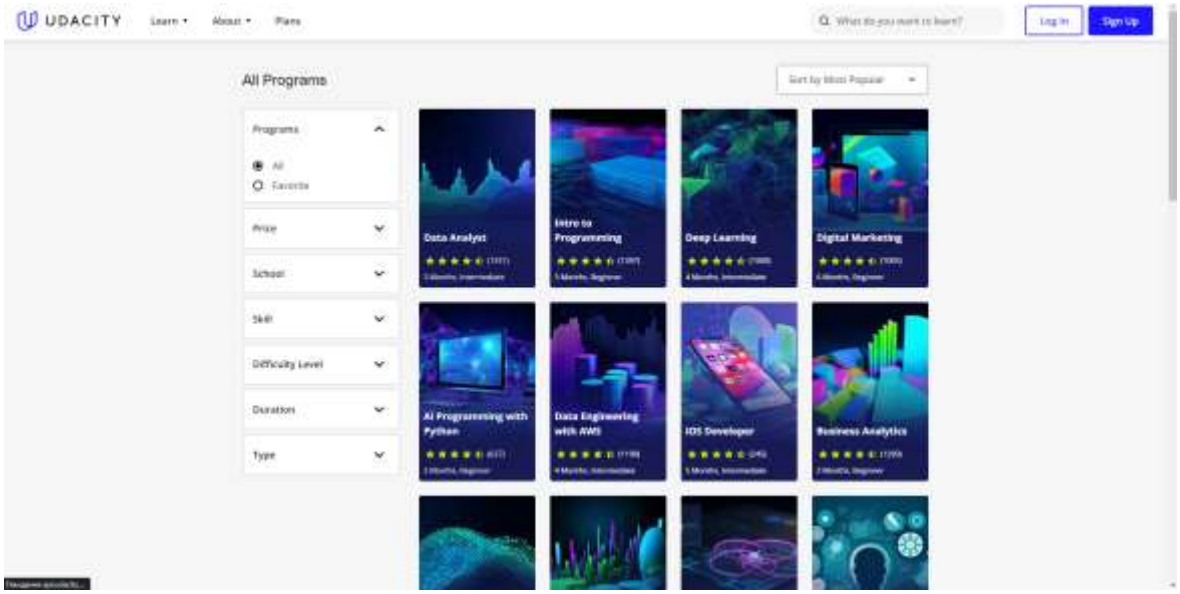

Рис. 2.4 Головна сторінка веб-академії Udacity

Рис. 2.5 Лекційна сторінка веб-академії Udacity

Висновок: сайт має недоліки з дизайном, проте загалом виконаний на достатньому рівні, він має інтуїтивну навігацію та є адаптивним. Особливістю є досвідченість людей, що створюють матеріали. Метод викладання матеріалу є ефективним, проте гіршим за попередній сайт – Khan Academy. Udacity не має текстових лекцій, що може стати проблемою так як не кожну лекцію можна пояснити за

допомогою відео. Проте сайт можна використати для подальшого створення нашої академії, особливо метод використання відео-лекцій.

2.3 Аналіз Edx

Edx – веб-академія, яка створена Массачусецьким технологічним інститутом та Гарвардським університетом. Всі лекції викладають професійні педагоги. Особливістю академії є досвідченість викладачів, безкоштовність матеріалів та різноманітність курсів.[10]

Дизайн сайту загально не має недоліків. Він виконаний в мінімалістичному стилі, при цьому фото відіграють велику роль в ньому, вони також підібрані згідно загального стилю. Також варто виділити відчуття бренду сайту та адаптивність для великих екранів. Проте недоліком варто виділити перенасичення другорядною інформацією сторінок. Особливо це стосується лекційних сторінок. Навігація виконана на високому рівні, є навіть перенасичення елементами, що відповідають за навігацію. (Рис 2.6, Рис 2.7)

Сайт має більше чотирьох тисяч курсів. Є два типи лекцій – текстові та відео. Відео лекції довгі за часом (більше п’яти хвилин). Текстові лекції великі за об’ємом. Стосовно кожної лекції є обговорення з викладачем, проте лише тоді, коли студент приєднався до вивчення в визначений день. В кінці теми є пропозиції стосовно практики. Тести або перевірка знань на сайті відсутня. Загалом навчання не ефективне з точки зору на розміри лекцій та відсутність будь-якої практики. (Рис 2.7)

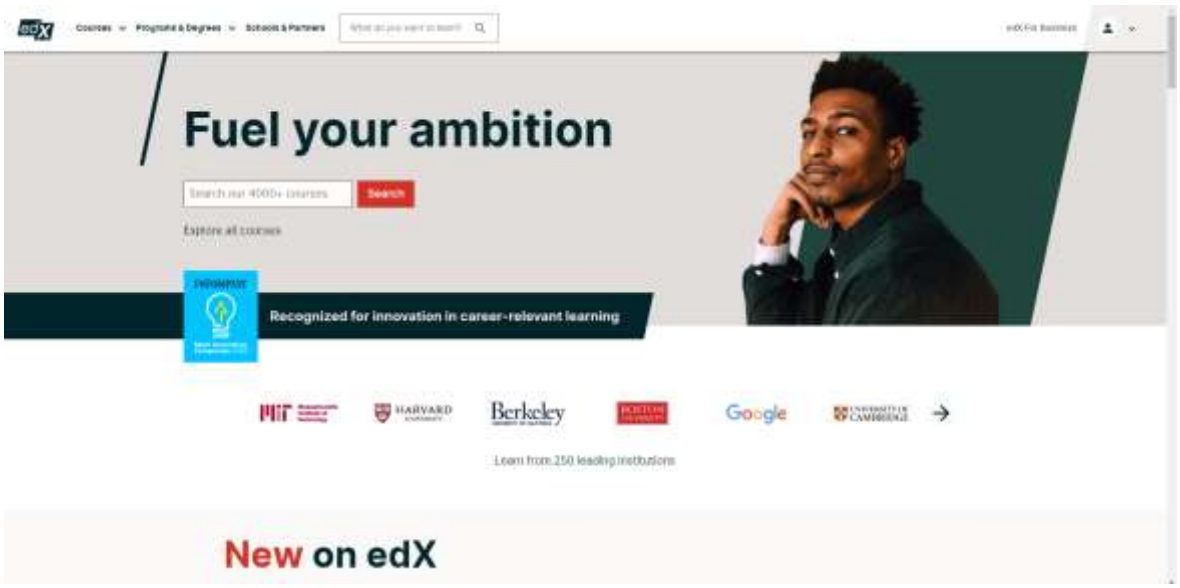

Рис. 2.6 Головна сторінка веб-академії Edx

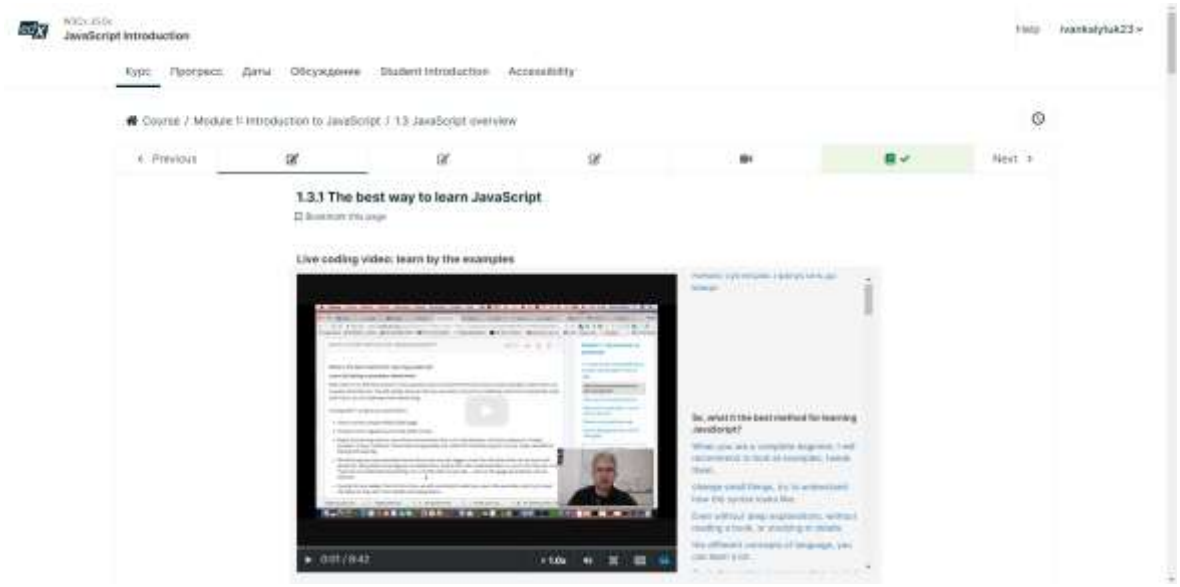

Рис. 2.7 Лекційна сторінка сайту Edx

Висновок: сайт має хороший дизайн, він створює бренд сайту, є адаптивним, сучасним та інтуїтивним. Проте суттєвою проблемою можна назвати перенасичення другорядними елементами дизайну. Метод навчання на сайті є неефективним – створені лекції великі за обсягом та не мають практичної частини. Сайт можна взяти за приклад для створення свого дизайну або структури.

2.4 Аналіз веб-академії CourseEra

CourseEra – це одна з найпопулярніших веб-академій. Вона розроблена на базі Стенфордського університету і містить курси, розроблені його викладачами. Особливістю академії є надання сертифікатів по закінченню навчання та безкоштовністю більшості курсів.

Сайт має хороший дизайн, проте він не має власних особливостей, тому не створює бренду. На сайті добре підібрана палітра кольорів. Навігація виконана на високому рівні. Проблемою дизайну можна назвати перенасиченість текстом та додатковими функціональними елементами дизайну, без яких сайт виглядав би краще. Адаптивність також виконана на високому рівні. (Рис 2.8)

Навчальні матеріали поділені на предмети, розділи та теми. В кожній темі є багато лекцій, виконується метод маленьких кроків. Лекції поділяються на відео та читання. Також присутні практичні завдання, які не мають вигляд тестів або інших видів опитування, а пропозицію зробити все індивідуально без перевірки. В платних версіях курсів викладач може перевіряти завдання та консультувати користувача. Також варто виділити багато освітніх програм, які не стосуються вивчення лекцій напряму, всі вони подані на головні сторінці сайту. (Рис. 2.10)

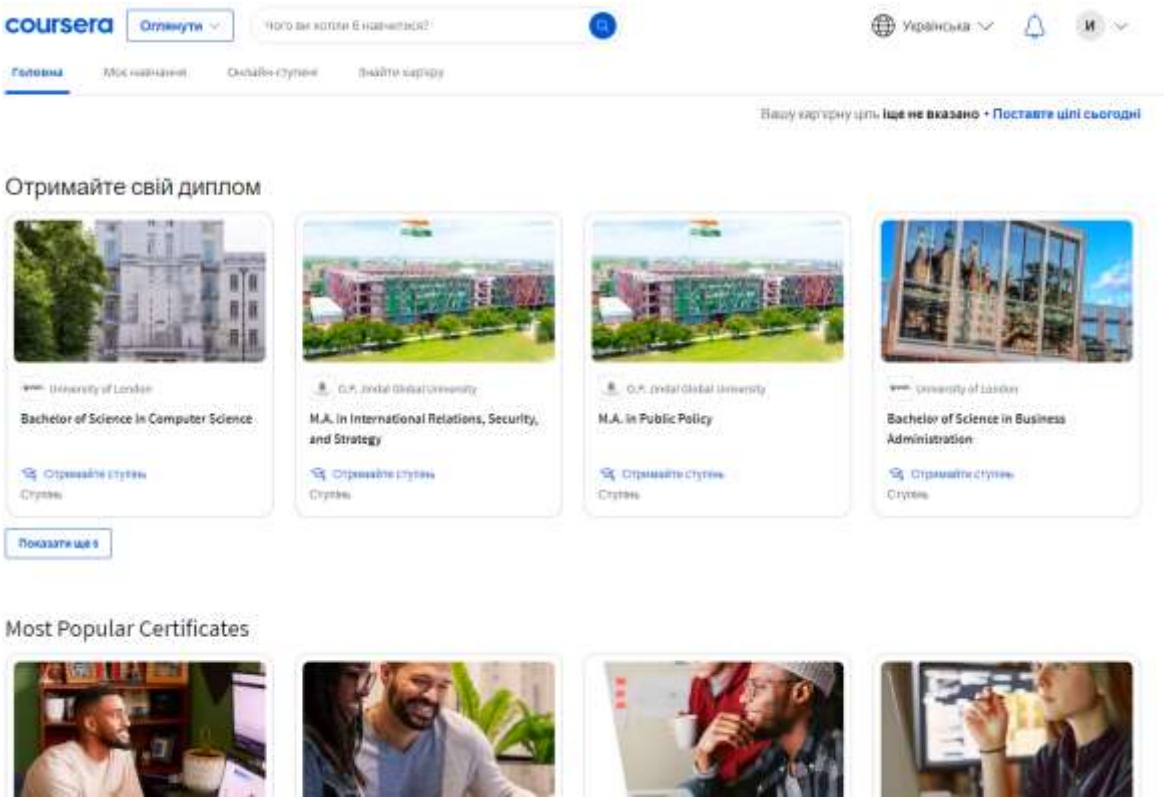

Рис 2.8 Головна сторінка сайту Coursera

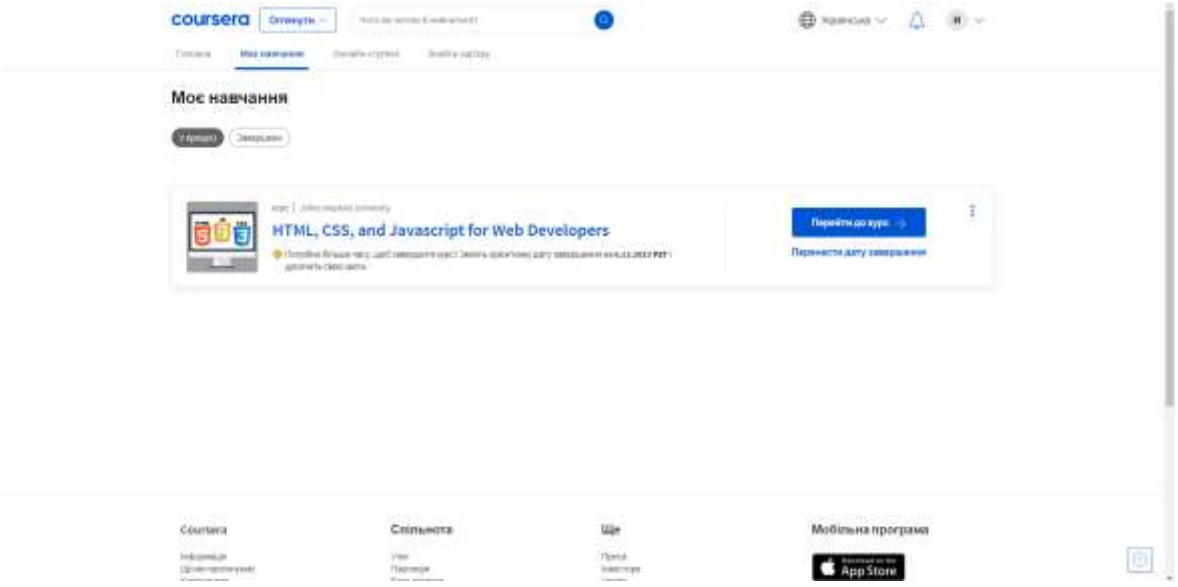

Рис. 2.9 Сторінка обраних курсів сайту Coursera

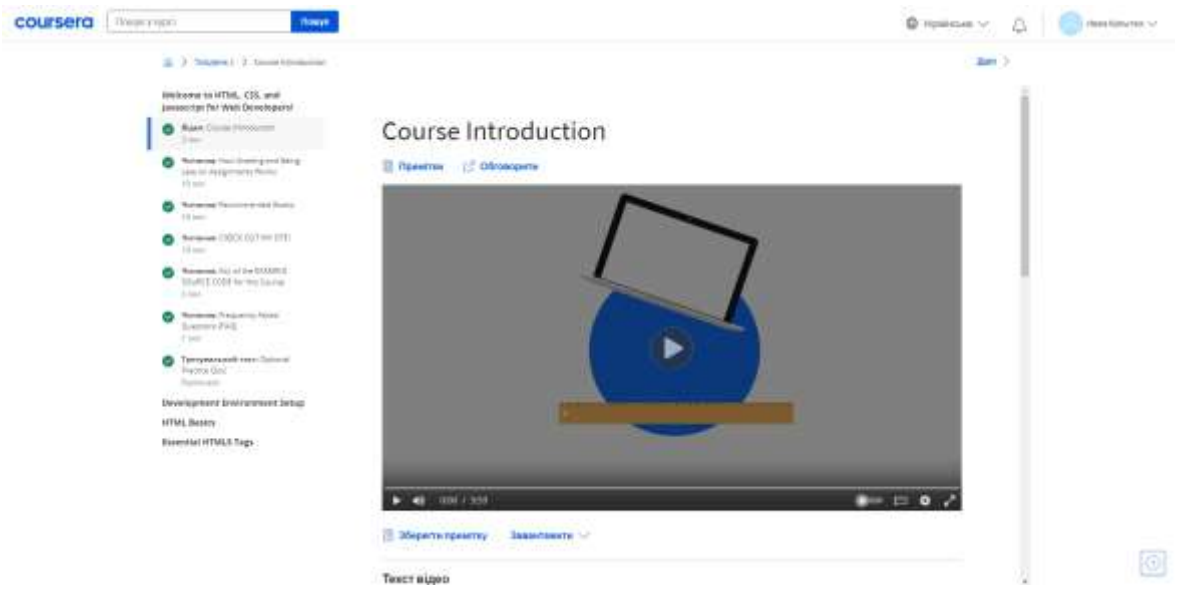

Рис. 2.10 Сторінка лекції сайту Coursera

Висновок: сайт має хороший, проте типовий дизайн. Він є інтуїтивним , сучасним та адаптивним. Начальні матеріали викладаються на високому рівні, їх можна порівняти з сайтом Khan Academy, проте на ефективність сильно впливають відсутність практики у вигляді тестів.

2.5 Аналіз веб-академії Codecademy

Веб-академія Codecademy спеціалізується на програмуванні, нею користуються як студенти так і вчителі. Сайт має безкоштовну та платну версії. [13]

Дизайн сайту виглядає не сучасно, проте це є особливістю, що виділяє його серед конкурентів. Він має інтуїтивну навігацію та добре виглядає на смартфонах та планшетах. Сайт не перенасичений інформацією. (Рис 2.11, Рис 2.12)

Метод викладання навчальних матеріалів є нетиповим для веб-академій. Лекції відбуваються паралельно практиці, а для практики сайт має поля для написання коду, який можна перевірити. Самі лекції невеликі за обсягом, є поділ на теми та розділи, що забезпечує метод маленьких кроків. (Рис. 2.13)

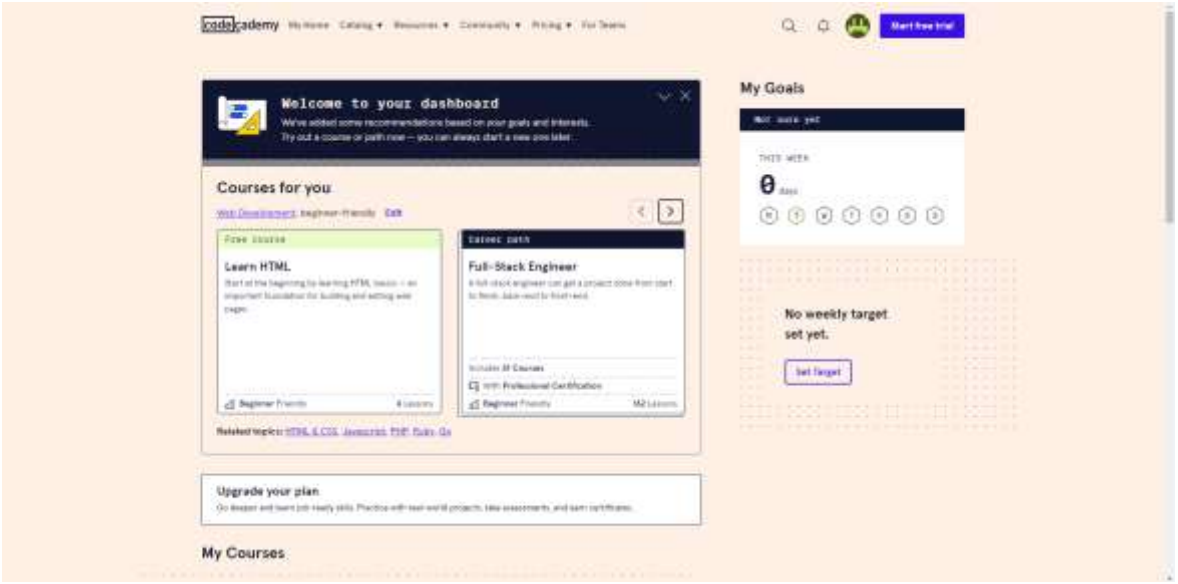

Рис. 2.11 Головна сторінка сайту Codecademy

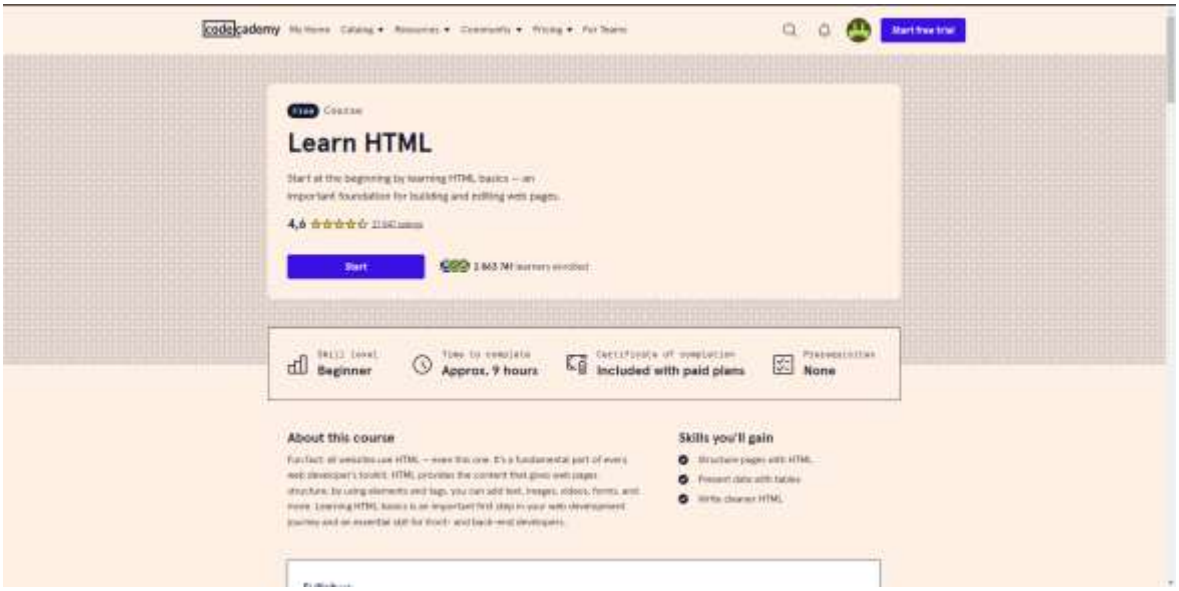

Рис. 2.12 Сторінка предмету сайту Codecademy

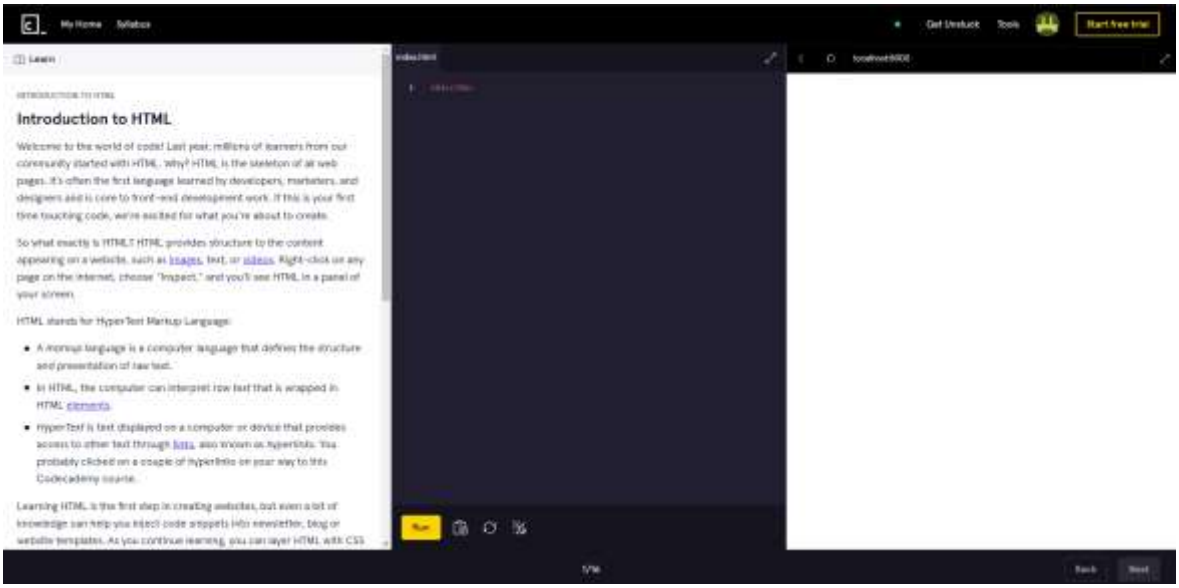

Рис 2.13 Сторінка лекції сайту Codecademy

Висновок: Сайт має нетиповий дизайн, який є його особливістю, також він адаптивний та інтуїтивний. Метод викладання матеріалу не підходить для наших задач через його вузьку спеціальність.

Висновки

Проаналізувавши дизайн веб-академій можна виявити ряд суттєвих недоліків, на які слід звернути уваги при розробці власного сайту:

- Перенасичення додатковими елементами дизайну. Такі академії як Khan Academy, Edx, Coursera в дизайні використовують елементи, які відволікають на процес навчання. У випадках коли вони виконують важливі функції їх треба спрощувати або робити непримітними щоб не впливати на концентрацію користувача.
- Відсутність бренду сайтів. Проблемою є типовість дизайну деяких веб-академій, який не запам’ятовується. Це може вплинути на приплив користувачів та загальне відчуття якості сайту.
- Загальні проблеми дизайну. Стилі деяких сайтів, або деякі окремі елементи не відповідають запитам сучасності.

Також варто виділити і ті критерії, які задовольняються на більшій частині проаналізованих сайтів. Щоб успішно конкурувати з ними нам обов’язково потрібно їх врахувати та покращити:

- Адаптивність. Всі проаналізовані академії є адаптивними, вони добре виглядають на смартфонах, планшетах та комп’ютерах. Покращити її можна за допомогою створення чуйної адаптивності.
- Навігація. Інтуїтивність в навігації на кожному з проаналізованих сайтів створена на високому рівні.

При аналізі методів надання освіти нами було виявлене суттєві відмінності методів на різних сайтах. Деякі з них мають свої недоліки, тому для створення власних навчальних матеріалів нам необхідно їх

поєднувати, обираючи найефективніші. В приклад можна поставити Khan Academy, яка є найефективнішою академією з проаналізованих. В ній є як відео, так і текстові лекції, після кожної лекції, теми та розділу є тести. Цей сайт модна взяти як за приклад для створення власної академії.

РОЗДІЛ 3. РОЗРОБЛЕННЯ ВЕБ-АКАДЕМІЇ

3.1 Розроблення дизайну

Дизайн сайту було виконано у стилі мінімалізму. Також було додано елементи матеріалізму, тому що вони роблять загальний вигляд сайту більш насиченим і відповідають стилю мінімалізму. Також використання геометричних фігур добре підходить до освітньої тематики сайту. На створення дизайну суттєво вплинув аналіз сайтів Edx та Khan Academy, які мали недолік в перенасиченні додатковими елементами дизайну. Врахувавши цю проблему в нас вийшов дизайн, що відповідає критеріям за якими ми проводили аналіз. (Рис. 3.3, Рис. 3.4) [7]

Для створення бренду та зв’язку з Херсонським державним університетом було використано його символи, такі як логотип університету, який став фавіконом та частиною логотипу сайту, офіційні кольори університету та кіт Василь, який став основою бренду. (Рис. 3.2, Рис. 3.4) [6] [4]

Дизайн було розроблено у програмі Figma, окремі елементи дизайну розроблялись в програмах Adobe Photoshop та Adobe Illustrator.

Спочатку було розроблено головну сторінку сайту, щоб відштовхуючись від неї створити всі інші. На ній ми розташували загальну інформацію про сайт, його особливості, та список найпоширеніших курсів. (Рис. 3.1)

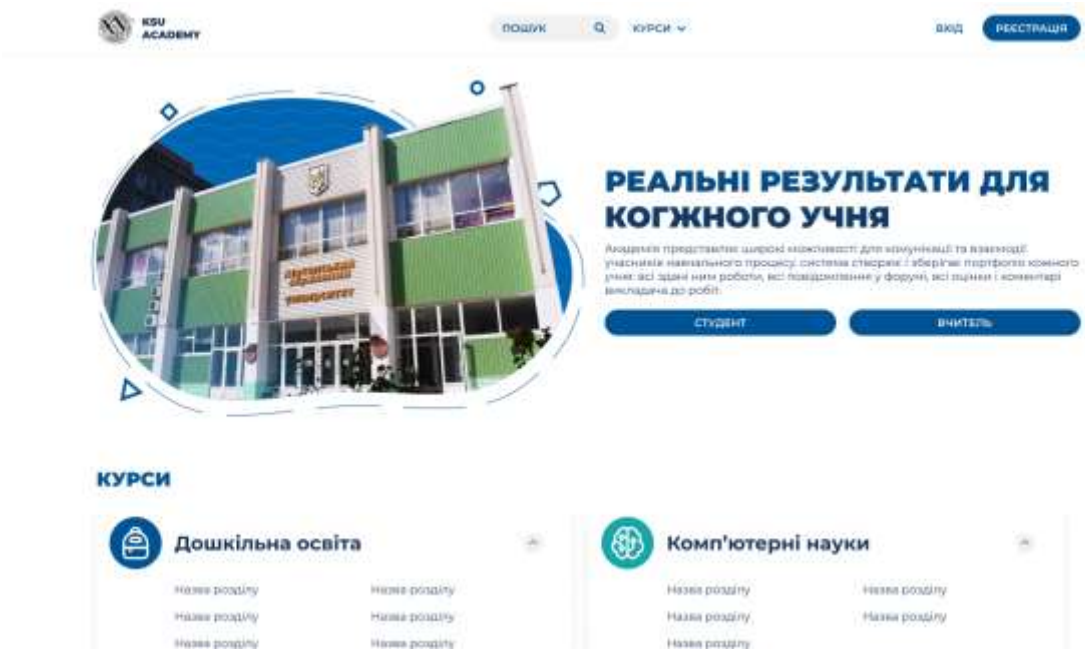

Рис. 3.1 Дизайн головної сторінки сайту KSU Academy

Рис. 3.2 Головна сторінка сайту KSU Academy

Наступною було розроблено сторінку предмета. На ній знаходяться всі розділи, які знаходяться в предметі та їх теми. Нами не було використано додаткових елементів дизайну як у інших сайтів за їх непотрібністю. При натисканні на назву розділу чи однієї з його тем користувач може перейти до сторінки розділу. (Рис. 3.3)

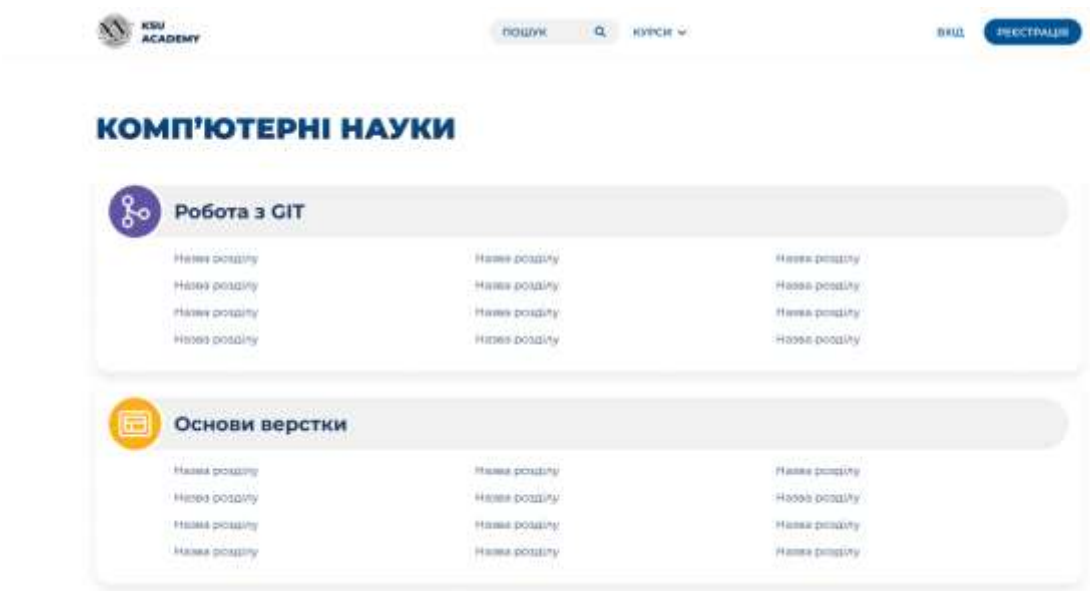

Рис. 3.3 Дизайн сторінки предмета

Сторінка розділу складається з трьох частин. Першою є його назва та відображення прогресу студента по темам. Другою частиною є перелік тем, що мають вигляд блоків з лекціями та тестами. Третьою частиною є заліковий тест розділу, після якого студенту зараховується проходження розділу. (Рис. 3.4)

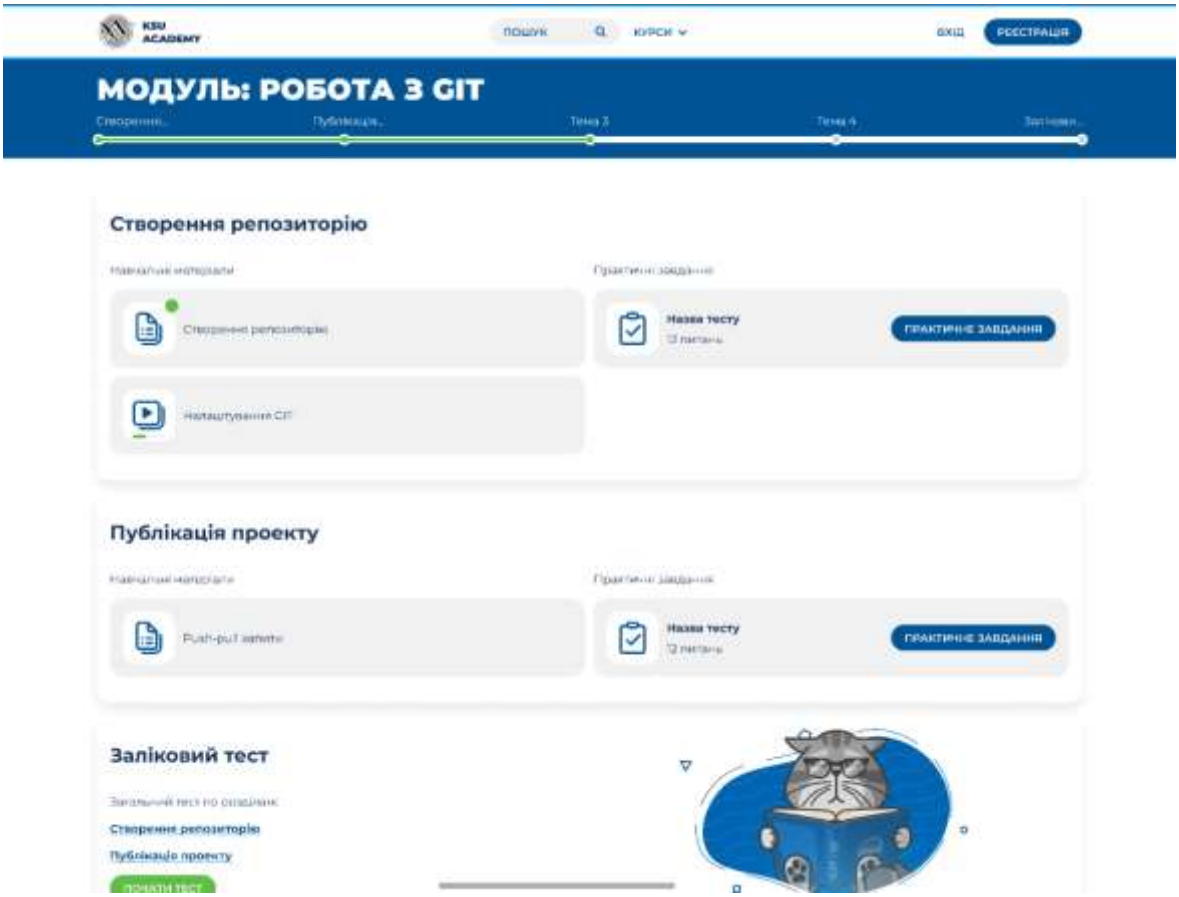

Рис. 3.4 Дизайн сторінки розділу

Лекційна сторінка може бути або у вигляді відео, або у вигляді тексту. Вони можуть містити фото, посилання, заголовки.(Рис. 3.5)

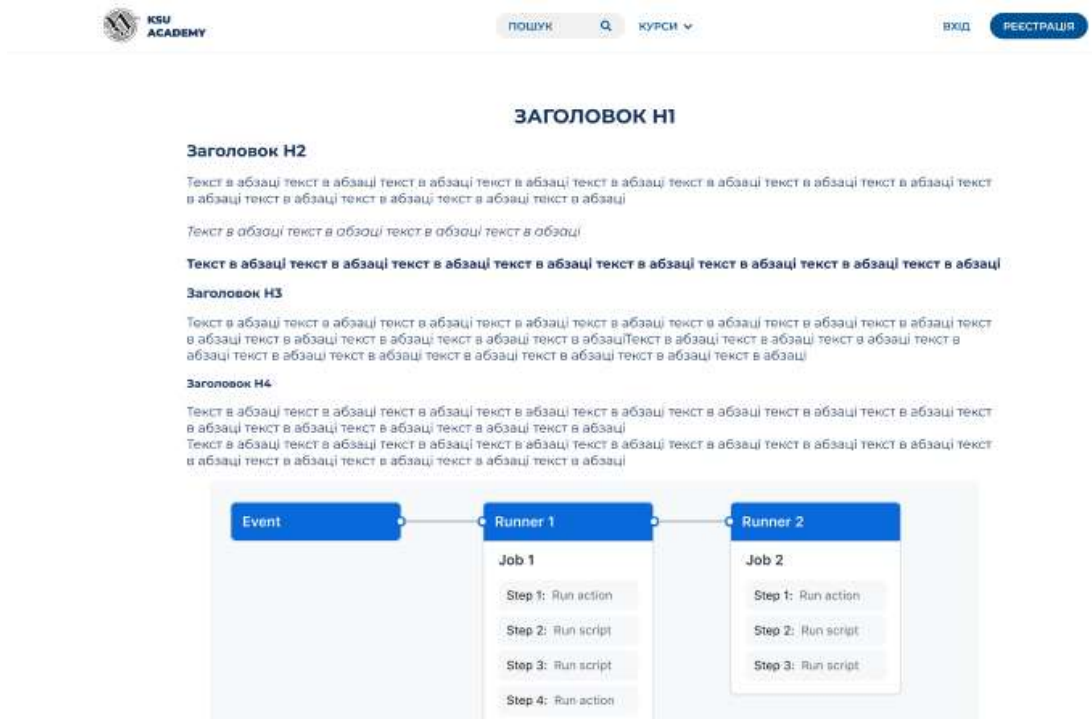

Рис. 3.5 Дизайн сторінки текстової лекції

Так як тести суттєво впливають на ефективність навчання, тому їх було розроблено як окремий компонент. Тест буде присутній на кожній лекції, в кінці проходження теми, та розділу. Він може мати одну або декілька правильних відповідей, питання до тесту може містити фото. (Рис. 3.6)

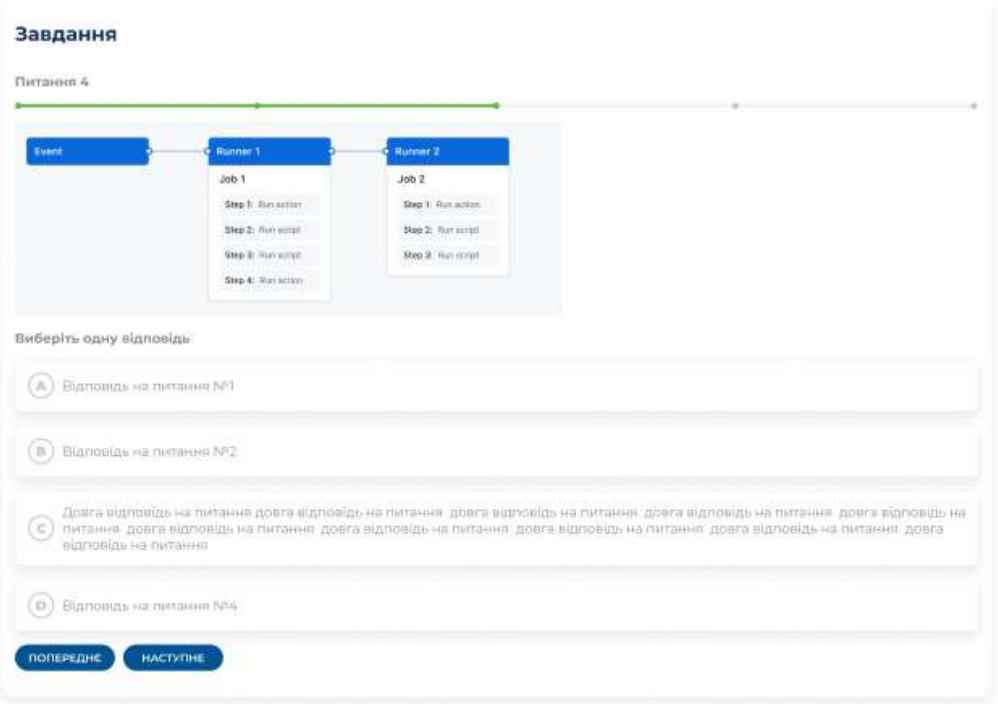

Рис. 3.6 Дизайн тесту

Сторінка профіля має інформацію про користувача. Окремо розроблено дизайн для викладача та студента, так як вони повинні мати доступ до різної інформації. Студент має інформацію про власний профіль, кількість пройдених курсів, сертифікати та таблицю пройдених курсів. Профіль викладача має інформацію про результати студентів, підписаних на їх курси, кількість створених курсів та посилання на сторінку для створення нового курсу. Також є окреме місце для додаткової інформації, яка може з’явитись в ході розробки. (Рис. 3.7, Рис. 3.8)

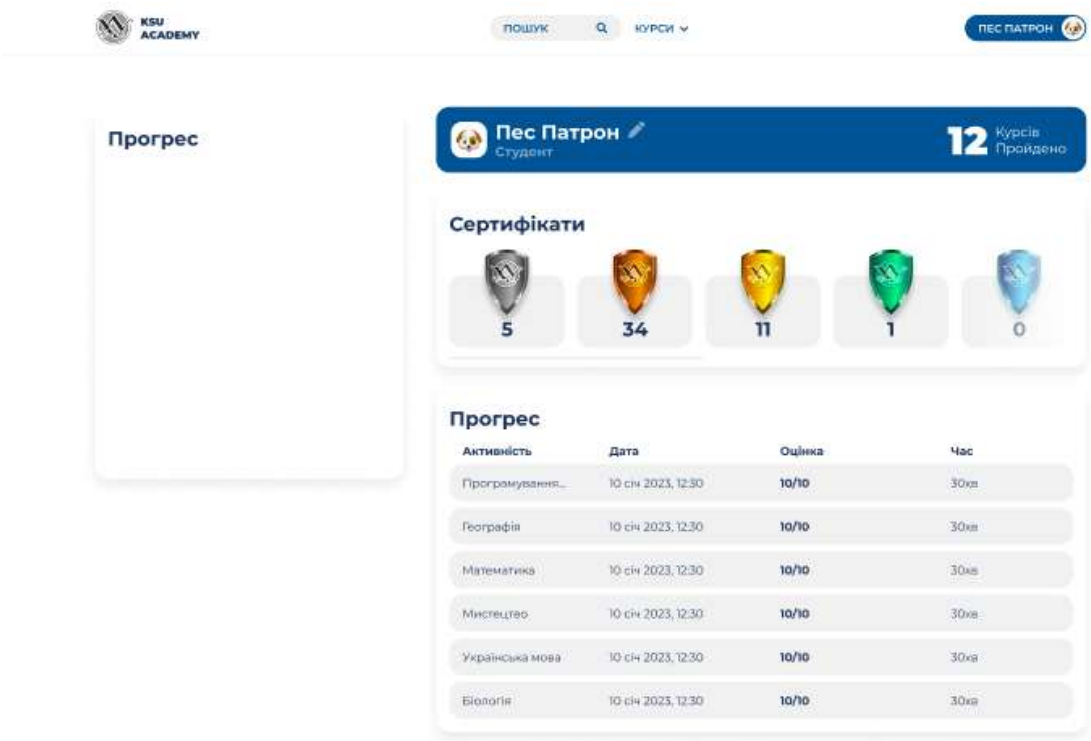

Рис. 3.7 Дизайн профіля студента

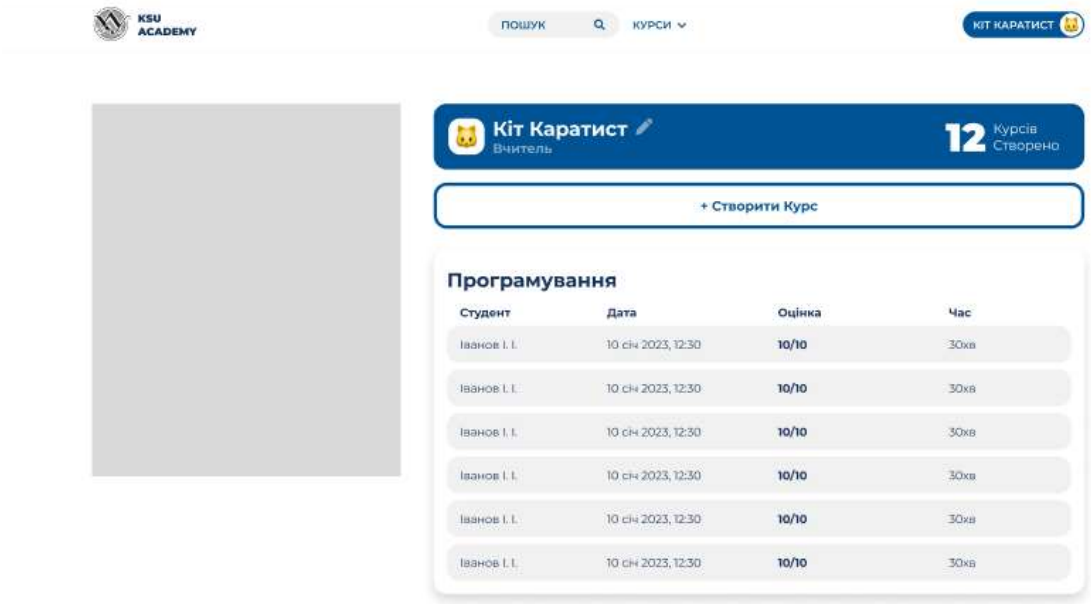

Рис. 3.8 Дизайн профіля викладача

Було також розроблено сторінку реєстрації та входу в профіль. В ньому подана можливість реєстрації через Google, Facebook та Apple. В майбутньому при реєстрації буде виконуватись перевірка на правильність написання пошти та паролю, неправильно введені поля будуть позначатись, а помилка коментуватись поруч з полями вводу. (Рис. 3.9) [30]

Рис 3.9 Компонент для реєстрації користувача

Окремо було розроблено адаптивний дизайн, проте лише для головної сторінки, щоб розуміти як повинен виглядати адаптивний дизайн загально. Також маленькі сторінки, наприклад для контактних даних не були розроблені, так як їх контент – це текст та посилання, які можна розробити по загальному шаблону. (Рис. 3.10)

Рис. 3.10 Головна сторінка сайту для телефонів

3.2 Верстка дизайну

Весь сайт ми розробляли за допомогою бібліотеки React, тому верстку ми написали одразу на ньому. Також ми використовували

препроцесор sass для більш зручної стилізації та бібліотеку React-Router-Dom для переходу між сторінками. (Рис. 3.11)

```
<div className="App">
  <Header></Header>

    <div className="pageContent">
      <Routes>
        <Route path='/' Component={MainPage} />
        <Route path='/course/:id' Component={CoursePage} />
        <Route path='/profile' Component={ProfilePage} />
        <Route path='/registration' Component={RegistrationPage} />
        <Route path='/course/:id/theme/:idr' Component={ThemePage}/>
        <Route path='/SubjectDevelopment' Component={SubjectDevelopmentPage} />
        <Route path='/SubjectDevelopment/ThemeDevelopment/:id' Component={ThemeDevelopmentPage} />
      </Routes>
    </div>

  <Footer></Footer>
```

Рис. 3.11 Код routes для переходу між сторінками

Верстка була розроблена адаптивною, з використанням медіа тегів та гнучкого макету flex-box, щоб покращити використання сайту на всіх типах екранів незалежно від їх розмірів:

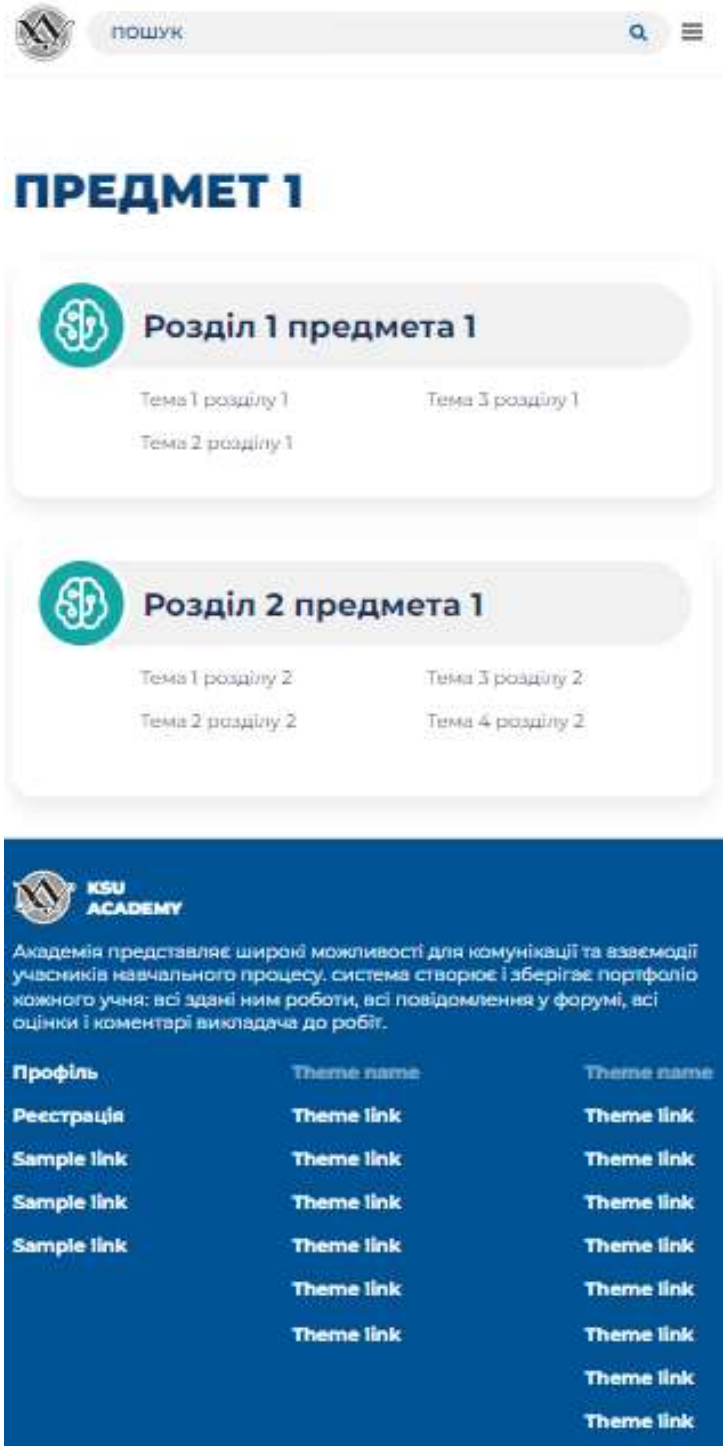

Рис. 3.12 Приклад адаптивності сторінки предмета

4.3 Розробка бекенду

Для розробки бекенду була використана мова програмування NodeJS для серверної частини та SQL для бази даних. Серверна частина була написана за допомогою Express, нами також використовувались бібліотеки Nodemon – для швидкого внесення змін та перезавантаження серверу, mysql2 – для встановлення зв’язку з базою даних MySQL,

CORS – для виправлення помилки cors, express-validator – для створення перевірки даних.

Таблиці бази даних MySQL мають велику вкладеність, це вдалось досягти за допомогою зовнішніх ключів. В таких таблицях як предмет та викладач ключі дублюються щоб покращити зв'язок між ними при об'єднанні. (Рис. 3.13, рис. 3.14)

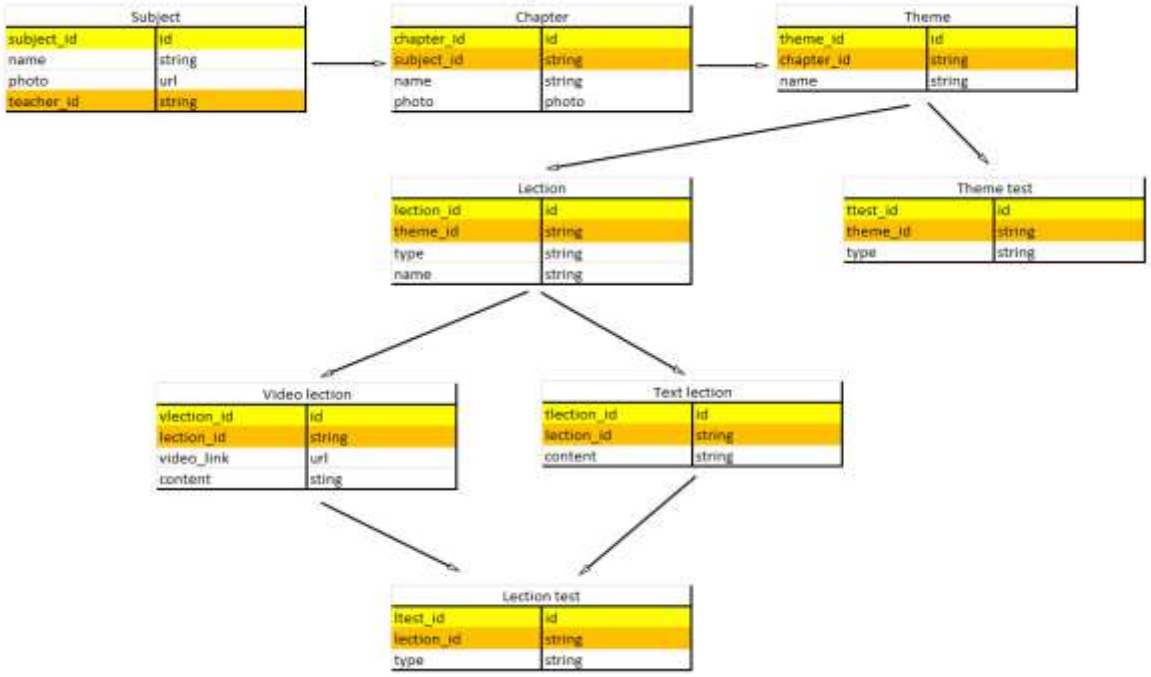

Рис. 3.13 Вигляд структури бази даних для предмета

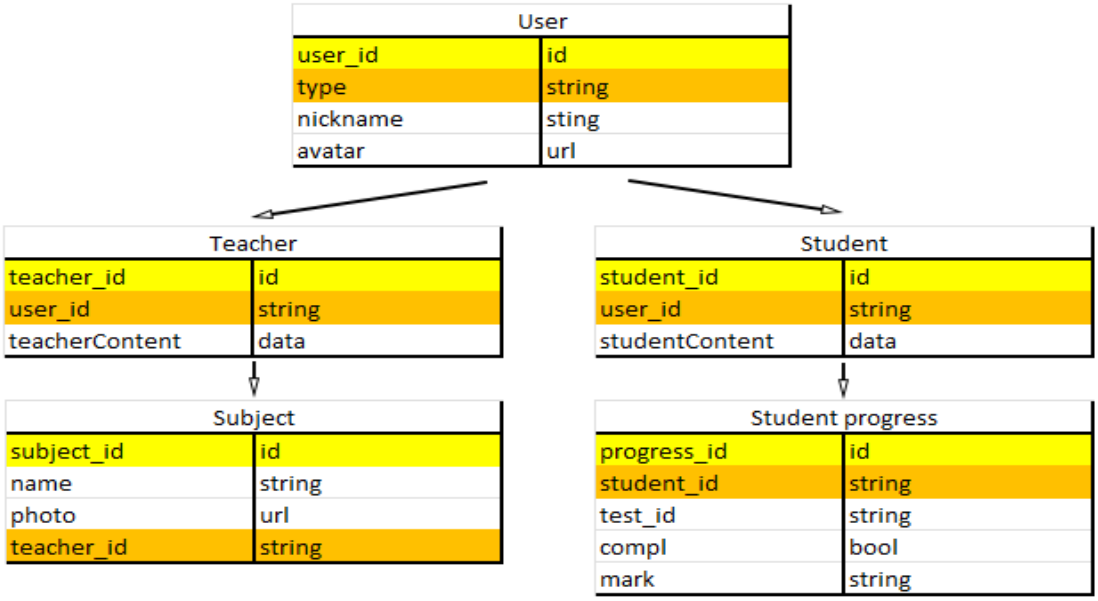

Рис. 3.14 вигляд структури бази даних для користувачів

Для кожних даних, що приходять на сервер було розроблено перевірку на правильність за допомогою бібліотеки Express-validator. [38]

```
import { body } from 'express-validator';  194.3k (gzipped: 60.3k)

export const registerValidator = [
  body('email').isEmail(),
  body('password').isLength({ min: 5 }),
  body('fullName').isLength({ min: 3 }),
  body('avatarUrl').optional().isURL(),
];
```

Рис. 3.15 Використання Express-validator для перевірки даних реєстрації

Обробка запитів була написана за допомогою методу CRUD (Create, Read, Update, Delete) для кожної таблиці. Ці дії були створені за допомогою надсилання до бази даних SQL коду. [40]

```
//отримання предметів
app.get('/subject', async (req, res) => {
    try{
        const rows = await db.execute('SELECT * FROM subject')
        res.json(rows[0])
    } catch (error){
        console.error(error)
        res.status(500).json({error: "Ошибка при получении данных"})
    }
})
```

Рис. 3.16 Отримання всіх предметів по методу CRUD (Read)

```
//отримання предмету по id
app.get('/subject/:id', async (req, res) => {
    try{
        let subject_id = req.params.id
        const rows = await db.execute('SELECT * FROM subject WHERE subject_id = ?', [subject_id])
        res.json(rows[0])
    } catch (error){
        console.error(error)
        res.status(500).json({error: "Ошибка при получении данных"})
    }
})
```

Рис. 3.17 Отримання одного предмета по методу CRUD (Read)

```
//оновлення даних предмета по id(params)
app.put('/subject/:id', async (req, res) => {
    try {
      const {subject_id} = req.query;
      const newValues = req.body;

      const [results] = await connection.execute(
        'UPDATE subject SET ? WHERE id = ?',
        [newValues, subject_id]
      );

      res.json({ message: 'Запись успешно обновлена' });
    } catch (error) {
      console.error(error);
      res.status(500).json({ error: 'Ошибка при обновлении записи' });
    }
});
```

Рис. 3.18 Оновлення предмета по методу CRUD (Update)

```
// видалення даних
app.delete('/subject/:id', async (req, res) => {
    try {
      const {subject_id} = req.query;

      const [results] = await connection.execute(
        'DELETE FROM subject WHERE id = ?',
        [subject_id]
      );

      res.json({ message: 'Запись успешно удалена' });
    } catch (error) {
      console.error(error);
      res.status(500).json({ error: 'Ошибка при удалении записи' });
    }
  });
```

Рис. 3.19 Видалення предмета по методу CRUD (Delete)

```
//створення предмету
app.post('/subject', async (req, res) => {
    try{
        const {teacher_id, name} = req.body

        if (!teacher_id || !name){
            return res.status(400).json({error: 'Не введено айди пользователя или название предмета'})
        }

        const [result] = await db.execute('INSERT INTO subject (teacher_id, name) VALUES (?, ?)', [teacher_id, name])
        res.json({ message: 'Предмет успешно добавлен', id: result.insertId });
    } catch(err){
        res.status(500).json({error: "Ошибка при добавлении предмета"})
    }
})
```

Рис. 3.20 Додавання предмета по методу CRUD (Create)

3.4 Розробка фронтенду

Для розробки фронтенду ми використали наступні бібліотеки:

- Axios для запитів на сервер
- Redux-tool-kit для зберігання даних, котрі необхідні по всьому додатку
- React-quill для створення текстового редактора
- React-select для створення списків (selects)
- Oauth20 для авторизації через Google акаунт

Було створено Redux сховище для зберігання інформації, яка необхідна в різних частинах додатку. Такими даними є інформація користувача (логін, тип, пошта) та предмети, їх розділи та теми. Лекції та тести не можна помістити в сховище, так як вони займають багато місця, що вплинуло б на швидкодію додатку.

Дані внесені до сховища будуть оновлюватись лише один раз – на етапі завантаження сайту, тому ми прописуємо можливість лише на отримання даних. [20][21]

```
import { createSlice } from '@reduxjs/toolkit';

const initialState = {
  email: '',
  userType: '',
  nickname: '',
};

const userSlice = createSlice({
  name: 'user',
  initialState,
  reducers: {
    setUser: (state, action) => {
      return action.payload;
    },
    clearUser: (state) => {
      return initialState;
    },
  },
});

export const { setUser, clearUser } = userSlice.actions;
export default userSlice.reducer;
```

Рис. 3.21 Приклад використання Redux для зберігання даних

Для отримання та відправки даних до серверу ми використали бібліотеку Axios. В ній ми окрім отримання даних також змінюємо статус їх надходження, який ми встановили за допомогою хука useState. Це було зроблено через особливості React. Дані приходять в момент до створення сторінки за допомогою хука useEffect. Проте коли DOM дерево вже створене, а дані ще не з’явились виникають помилки. Асинхронна функція відобразить наші дані тоді коли вони прийдуть, проте сама помилка з’являється незалежно від цього.(Рис. 3.22) [23]

```
//оновлення даних предмета
let [subjects, setSubjects] = useState([]);

//статус завантаження
let [loading, setLoading] = useState(false);

//функція запиту
async function fetchData() {
    setLoading(true)
    let res = await axios.get('http://localhost:1000/subject').then(({data}) => data).finally(setLoading(false))
    return res
}

//функція, що відбувається при першому рендері
useEffect(() => {
    fetchData().then((data) => setSubjects(data))
}, []);
```

Рис. 3.22 Приклад отримання предметів

Для запобігання помилок, поки асинхронна функція не отримала і не додала дані нам ми будемо відображати пусте поле. (Рис. 3.23)

```
<div className="mainPage_courses_columns" >
    {
        loading ? (
            (null)
        ) : (
            subjects.map((item) => (
                <Subject name = {item.name} key = {item.subject_id} id = {item.subject_id}></Subject>
            ))
        )
    }
</div>
```

Рис. 3.23 Приклад відображення отриманих предметів

Авторизацію було розроблено як для тих, хто не має google акаунта, так і для тих хто його використовує. В першому випадку при реєстрації ми перевіряли його пароль та пошту а також на сервері створювали веб-токен за допомогою бібліотеки JWT. При авторизації цей веб-токен перевірявся і тільки після цього користувач міг використовувати наш додаток. В другому випадку ми використовували дані, котрі нам передала бібліотека Oauth20. Також було розроблено перевірку даних ще до відправки на сервер, щоб не робити запити з неправильною інформацією.

```
const validateEmail = (email) => {
    return String(email)
        .toLowerCase()
        .match(
            /^(([^<>()[\]\\.,;:\s@"]+(\.[^<>()[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/
        );
};
```

Рис. 3.24 функція для перевірки email

Для розробки лекцій ми використали бібліотеки react-select та react-quill. Викладач після переходу на сторінку профіля має можливість перейди до розроблення навчальних матеріалів. Для цього необхідно натиснути на посилання “+ Створити курс”, після чого користувач перейде на сторінку розробки предмету.

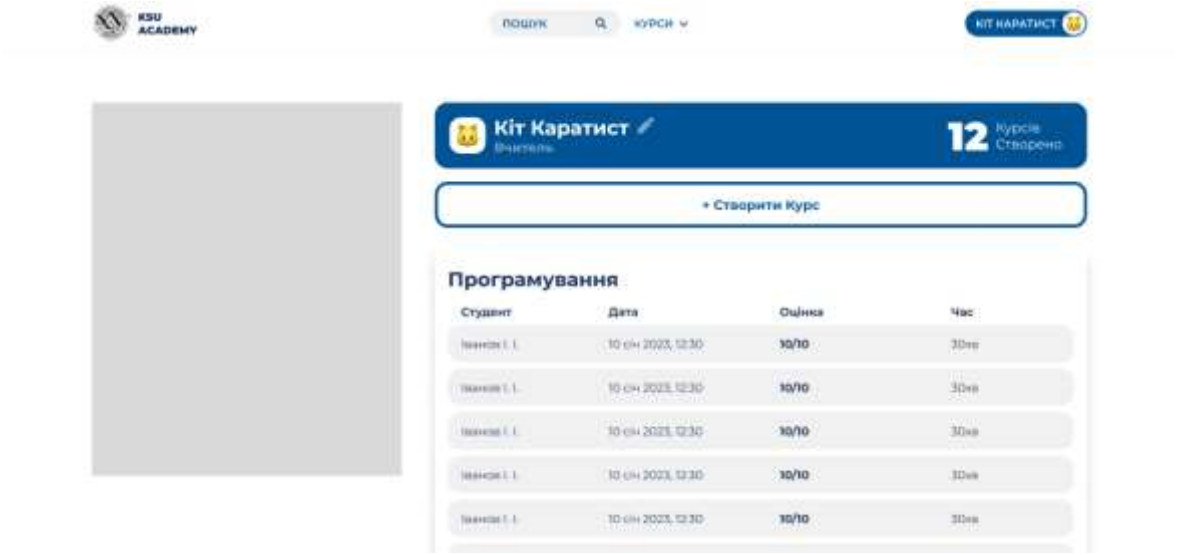

Рис. 3.25 Сторінка профіля викладача з можливістю створити курс

Сторінка розробки предмета виглядає схожою на сторінку предмету. Тут викладач може обрати раніше створений ним предмет з селектору, після обрання котрого з’явиться список всіх його розділів. Також викладач може створити новий предмет або розділи до нього. Щоб перейти до подальшої розробки викладач обирає розділ і переходить на сторінку розробки тем. Всі дані на сторінках розробки надходять не з сховища Redux, а з серверної частини, щоб користувач не оновляв сторінку для завантаження нових даних в сховище. [29]

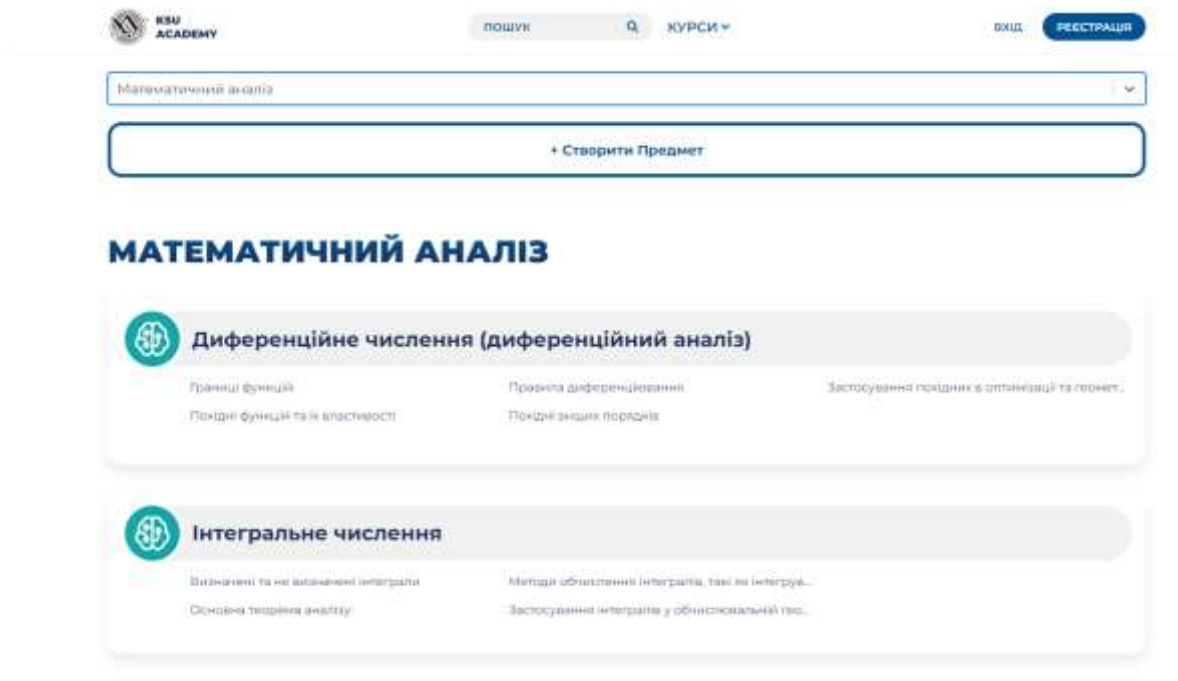

Рис 3.26 Сторінка розробки предмету

При переході на сторінку розробки тем користувач так само може створити теми, лекції та тести. Розроблення тестів та лекцій відбувається за допомогою текстового редактора Quill. В його властивості ми додали зміну шрифту, позиціонування тексту, вставку фотографій, створення посилань, створення відео-матеріалу за посиланням, створення формул та створення заголовків. Якщо викладачі звернуться щодо додання нових властивостей, їх можна буде швидко розробити через їх гнучкість. [25]

Всі дані, які ми отримуємо з текстового редактору мають вигляд HTML коду. Фотографії додані до тексту перетворюються в закодований файл, тому ми можемо без бібліотеки Multer зберігати його в базі даних у вигляді звичайного тексту.

Розроблення тестів також було створене за допомогою текстового редактора Quill, в ньому викладач повинен написати питання, а потім створити маркований список, в якому будуть відповіді. Правильні та неправильні відповіді викладач позначає в дужках – 1 для правильної, 0 для неправильної.

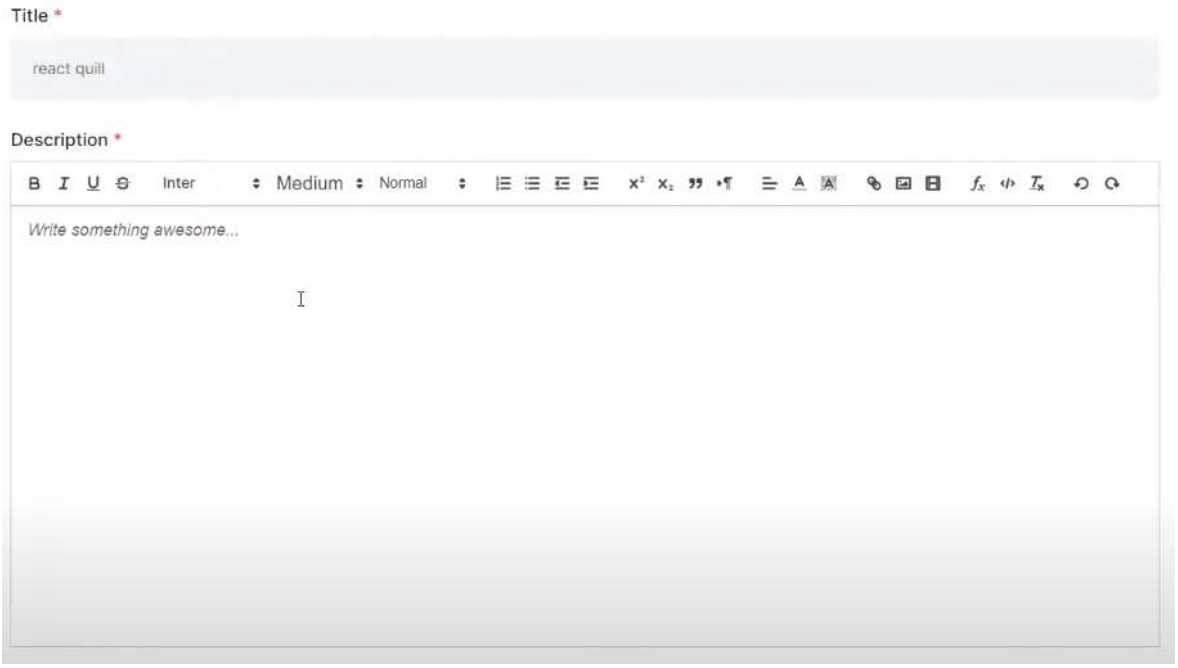

Рис 3.27 Приклад редактору з усіма можливими властивостями

Редагування вже розроблених лекцій та тестів відбувається за допомогою оберненої процедури повернення HTML коду до текстового редактора, після чого його можна змінити як звичайний текст та підтвердити зміни.

Відображення лекцій відбувається за допомогою властивості “dangerouslySetInnerHTML” та передачі в неї HTML коду. Ця властивість є аналогом “innerHTML” для звичайного JavaScript. Відображення тестів відбувається за допомогою функції map, призначеної для масиву тестів та відповідей всередині компоненту тесту. [22][1]

3.5 Рефакторинг коду та виправлення помилок

Після закінчення розробки нам необхідно зробити рефакторинг коду для зручності його читання та зберігання файлової структури. Всі запити на серверній частині ми об’єднали в папки для кожної таблиці. Ті самі дії були створені для фронтенду.

В більшості випадків частина рефакторингу відбувалась саме під час розробки, проте у фінальній стадії створення проекту необхідно пройти весь проект для пошуку помилок та виправлення помилок. [28]

ВИСНОВОК

В ході дослідження були сформовані основні вимоги до освітньої платформи, які враховують потреби зовнішніх учасників освітнього процесу та викладачів. Проведена робота була спрямована на визначення вимог ефективної веб-академії, аналіз аналогів та розробку власної академії.

Розробка вимог допомогла визначити чіткий план для аналізу інших веб-академій та майбутньої розробки сайту KSU Academy.

Аналіз успішних веб-академій згідно визначених в першому розділі критеріїв показав, що існуючі освітні платформи мають як переваги, так і обмеження та недоліки. Враховуючі ці результати було створено план по якому розроблявся сайт KSU Academy.

Розробка освітньої платформи займала п’ять етапів, що включали створення дизайну, верстку, бекенд, фронтенд та тестування. Розроблена академія на основі університету відповідає вимогам сучасності та ефективності навчання, що допоможе користувачам отримувати якісну освіту або створювати освітні матеріали для його надання.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Властивість для внесення HTML коду в елемент React (онлайн ресурс) URL: https://ru.legacy.reactjs.org/docs/dom-elements.html#dangerouslysetinnerhtml

2. Вимоги до мінімалізму в дизайні інформаційних та освітніх сайтів (онлайн ресурс) URL: https://web-valley.ru/articles/tpost/4a6sxgp4b1-minimalizm-v-veb-dizaine

3. Мінімалізм в дизайні сайтів [он-лайн ресурс]. URL: http://seo-design.net/design/minimalism-in-design-good-or-bad

4. Фірмовий стиль університету ХДУ [он-лайн ресурс]. URL: https://www.kspu.edu/About/Identity.aspx?lang=uk

5. Створення оптимальної структури сайту [он-лайн ресурс]. URL: https://seranking.ru/blog/seo/struktura-sajta/

6. Роль сайту в формуванні іміджа освітнього закладу [он-лайн ресурс]. URL: https://cyberleninka.ru/article/n/rol-sayta-v-formirovanii-imidzha-vuza-na-primere-sayta-innovatsionnogo-evraziyskogo-universiteta-pavlodara

7. Основи мінімалістичного дизайну [он-лайн ресурс] URL: https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BD%D0%B8%D0%BC%D0%B0%D0%BB%D0%B8%D0%B7%D0%BC_(%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD)

8. Використання матеріалістичного дизайну на прикладі Google [он-лайн ресурс] URL: https://itfuture.online/material-design-genialnyj-hod-google/

9. Chat-GPT за допомогою якого була написана частина коду [он-лайн ресурс] URL: https://chat.openai.com/

10. Веб-академія Udacity [он-лайн ресурс] URL: https://www.udacity.com/

11. Веб-академія Edx [он-лайн ресурс] URL: https://www.edx.org/

12. Веб-академія Khan Academy [он-лайн ресурс] URL: https://ru.khanacademy.org/

13. Веб-академія Codecademy [он-лайн ресурс] URL: https://www.codecademy.com/learn

14. Створення зручної навігації [он-лайн ресурс] URL: https://tilda.education/articles-navigation

15. Документація щодо використання flexbox (онлайн ресурс) URL: https://developer.mozilla.org/ru/docs/Learn/CSS/CSS_layout/Flexbox

16. Документація щодо використання grid layout (онлайн ресурс) URL:https://developer.mozilla.org/ru/docs/Web/CSS/CSS_grid_layout/Basic_concepts_of_grid_layout

17. Використання хука useEffect (онлайн ресурс) URL: https://ru.legacy.reactjs.org/docs/hooks-effect.html

18. Використання хука useState (онлайн ресурс) URL: https://legacy.reactjs.org/docs/hooks-state.html

19. Асинхронні запити (онлайн ресурс) URL: https://developer.mozilla.org/ru/docs/Web/API/XMLHttpRequest/Synchronous_and_Asynchronous_Requests

20. Документація щодо використання Redux (онлайн ресурс) URL: https://redux.js.org/introduction

21. Документація щодо використання Redux Toolkit (онлайн ресурс) URL: https://redux-toolkit.js.org/introduction

22. Властивість innerHTML в JavaScript (онлайн ресурс) URL: https://developer.mozilla.org/ru/docs/Web/API/Element/innerHTML

23. Virtual DOM в React (онлайн ресурс) URL: https://legacy.reactjs.org/docs/faq-internals.html

24. Валідація пошти на для авторизації (онлайн ресурс) URL: https://stackoverflow.com/questions/46155/how-can-i-validate-an-email-address-in-javascript

25. Використання React-quill (онлайн ресурс) URL: https://codeat21.com/react-quill-powerful-rich-text-editor/

26. Form та отримання даних з них в React (онлайн ресурс) URL: https://legacy.reactjs.org/docs/forms.html

27. Роль сайту в формуванні іміджу та бренду освітнього закладу (онлайн ресурс) URL: https://cyberleninka.ru/article/n/rol-sayta-v-formirovanii-imidzha-vuza-na-primere-sayta-innovatsionnogo-evraziyskogo-universiteta-pavlodara

28. Рефакторинг коду та його важливість (онлайн ресурс) URL: https://blog.ithillel.ua/ru/articles/zachem-i-kak-provodit-refaktoring-koda

29. Використання React-select (онлайн ресурс) URL: https://react-select.com/home

30. Використання React-router (онлайн ресурс) URL: https://v5.reactrouter.com/web/guides

31. Деструктуризація параметрів за допомогою хука useParams в React-router-dom (онлайн ресурс) URL: https://v5.reactrouter.com/web/api/Hooks/useparams

32. Документація щодо створення веб-токену JWT (онлайн ресурс) URL: https://jwt.io/

33. Методи дистанційного навчання (онлайн ресурс) URL: https://dspace.nuft.edu.ua/jspui/bitstream/123456789/5006/3/methods7.pdf

34. Використання відео-лекцій у очному навчанні (онлайн ресурс) URL: https://elibrary.kubg.edu.ua/id/eprint/26993/1/D_Nasts_KSHS_7(151)_Ndlio.pdf

35. Вплив тестів на ефективність навчання (онлайн ресурс) URL: https://ela.kpi.ua/bitstream/123456789/8576/1/29.pdf

36. Тестування як метод навчання (онлайн ресурс) URL: http://blog.ed-era.com/iak-tiestuvannia-spriiaie-iefiektivnomu-navchanniu/

37. Вимоги до тестування у педагогіці США (онлайн ресурс) URL: https://studentam.net.ua/content/view/7860/97/

38. Документація до бібліотеки Express-validator (онлайн ресурс) URL: https://express-validator.github.io/docs

39. Відмінності онлайн та дистанційного навчання (онлайн ресурс) URL: https://finacademy.net/materials/article/chem-otlichaetsya-onlajn-obuchenie-ot-distantsionnogo-obucheniya

40. Операції CRUD (онлайн ресурс) URL: https://www.freecodecamp.org/news/crud-operations-explained/

ДОДАТКИ

А. Кодекс академічної доброчесності

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Калитюк Іван Павлович, учасник(ця) освітнього процесу Херсонського державного університету, **УСВІДОМЛЮЮ**, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу. ЗАЯВЛЯЮ, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:

– дотримуватися:

• вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;

• принципів та правил академічної доброчесності;

• нульової толерантності до академічного плагіату;

• моральних норм та правил етичної поведінки;

• толерантного ставлення до інших;

• дотримуватися високого рівня культури спілкування; – надавати згоду на:

• безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;

• оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;

• використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;

– самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;

– надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;

– не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;

– своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;

– не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;

– підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі; – поважати честь, гідність та особисту недоторканність

особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;

– не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;

– відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науководослідницькі завдання;

– запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;

– не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;

– не підроблювати документи;

– не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;

– не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;

– не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;

– не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;

– не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;

– не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;

– не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам. УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.

29.10.2023		Іван Калитюк
(дата)	(підпис)	(ім'я, прізвище)