

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

Проектування веб-додатку для створення, публікації та керування відео-
курсами онлайн

Кваліфікаційна робота (проєкт)

на здобуття ступеня вищої освіти «бакалавр»

Виконав: здобувач
Спеціальності 121, Інженерія програмного
забезпечення
Освітньо-професійної програми
Лесів Олександр Іваннович

Керівник кандидат фізико-математичних
наук, Вейцблїт Олександр Йосипович
Рецензент старший (Senior) розробник в
компанії “ЕРАМ Systems” Кириєв Євген
Ігорович

ЗМІСТ

ЗМІСТ	3
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП	5
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ ВЕБ-ДОДАТКУ ДЛЯ СТВОРЕННЯ, ПУБЛІКАЦІЇ ТА КЕРУВАННЯ ВІДЕО-КУРСАМИ ОНЛАЙН	7
1.1 Аналіз предметної області. Обґрунтування необхідності розробки веб-додатку	7
1.2 Визначення переваг офлайн навчання та впровадження їх в онлайн навчання	9
1.3 Порівняльний аналіз аналогів веб-додатку для створення, публікації та керування відео-курсами онлайн	10
1.4 Технічне завдання на розробку веб-додатку.....	12
РОЗДІЛ 2 АНАЛІЗ І ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ	14
2.1 Обґрунтування вибору засобів для вирішення поставлених задач	14
2.2 Проектування логіки, архітектури та поведінки системи	18
РОЗДІЛ 3 РОЗРОБКА ВЕБ-ДОДАТКУ	25
3.1 Алгоритм обробки та аналізу даних	25
3.2 Проектування бази даних.....	28
3.3 Розробка інтерфейсу системи	32
3.4 Програмна реалізація логіки веб-додатку	34
РОЗДІЛ 4 ТЕСТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ СТВОРЕННЯ, ПУБЛІКАЦІЇ ТА КЕРУВАННЯ ВІДЕО-КУРСАМИ ОНЛАЙН.....	38
ВИСНОВКИ.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41
ДОДАТОК А ЛІСТИНГ КОДУ ВЕБ-ДОДАТКУ	44
ДОДАТОК Б ІНСТРУКЦІЯ КОРИСТУВАЧА.....	63
ДОДАТОК С КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ	68

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

BEM	– Block-Element-Modifier, з англ. – блок-елемент-модифікатор
CSS	– Cascading Style Sheets, з англ. – каскадні таблиці стилів
ER	– Entity-relationship – модель сутність-зв'язок
HTML	– HyperText Markup Language — мова розмітки гіпертексту
IDE	– Integrated development environment, з англ. – інтегроване середовище розробки
ІІС	– Internet Information Services, з англ. – інформаційний сервер в інтернеті
LESS	– Leaner Style Sheets – компактна таблиця стилів
MVC	– model-view-controller, з англ. – модель-представлення-контролер – архітектура розробки інформаційних систем
PHP	– Hypertext Preprocessor, з англ. – гіпертекстовий процесор, скриптова мова програмування для генерації HTML коду на стороні сервера.
SASS	– Syntactically Awesome Stylesheets – метамова на основі CSS
UI	– User Interface, з англ. – користувацький інтерфейс
UX	– User Experience, з англ. – користувацький досвід
БД	– База даних
ЗВО	– Заклад вищої освіти
ООП	– Об'єктно-орієнтоване програмування
СУБД	– Система управління базами даних

ВСТУП

Актуальність теми. Онлайн навчання дає здобувачам освіти можливість навчатися у зручний для них час та з будь-якої точки світу. Воно дозволяє відвідувати заняття, коли є вільний час, а також не потребує фізичної присутності в аудиторії. Дає змогу навчатися в своєму темпі, повторювати матеріал, який був не зрозумілим, та отримувати доступ до різноманітних джерел інформації, що допомагає зрозуміти предмет більш глибоко. Ситуація, пов'язана з пандемією та війною, примусила багато установ, у тому числі шкіл, коледжів і університетів, переходити на онлайн навчання. Такий перехід дозволив забезпечити безпеку та здоров'я студентів та викладачів. Онлайн навчання дозволяє використовувати різноманітні технології, такі як відео конференції, онлайн тестування та інтерактивні платформи, які роблять навчання більш цікавим та ефективним.

Отже, онлайн навчання є актуальним у зв'язку зі зміною сучасного світу та технологічного розвитку, що дозволяє студентам здобувати знання в більш зручний та ефективний спосіб. Воно також дозволяє широкій аудиторії студентів з різних куточків світу здобувати якісну освіту, незалежно від місця проживання. Більше того, онлайн навчання може стати цінним інструментом для навчання дорослих, які мають роботу та обмежений час для навчання, дозволяючи їм підвищувати свої кваліфікації та отримувати нові навички у своїй галузі.

Нарешті, онлайн навчання може допомогти розвивати та змінювати традиційну систему навчання, що забезпечить більш ефективне та прогресивне навчання в майбутньому. Все більше університетів та шкіл переходять на онлайн форму навчання та навчання за допомогою технологій, що свідчить про його актуальність та важливість у сучасному світі.

Мета і завдання дослідження. Метою дослідження є розробка веб-додатку для створення, публікації та керування відео-курсами онлайн. Завдання дослідження:

- проаналізувати предметну область;
- зробити аналіз програм-аналогів;
- розробка технічного завдання на розробку веб-додатку;
- провести аналіз та опис логіки і архітектури веб-додатку;
- розробити веб-додаток;
- провести тестування веб-додатку.

Об’єкт дослідження: процес навчання та взаємодії здобувачів освіти з викладачами.

Предмет дослідження: веб-додаток для створення, публікації та керування відео-курсами онлайн.

Методи дослідження. Отримані результати дослідження базуються на методах спостереження, порівняння, системного аналізу та дедукції.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ ВЕБ-ДОДАТКУ ДЛЯ СТВОРЕННЯ, ПУБЛІКАЦІЇ ТА КЕРУВАННЯ ВІДЕО-КУРСАМИ ОНЛАЙН

1.1 Аналіз предметної області. Обґрунтування необхідності розробки веб-додатку

В останні декілька років, відбувається переорієнтація освіти у всьому світі, Україна не є виключенням. Це спричинено самоізоляцією під пандемії коронавірусної інфекції COVID-19. Не встигли українці оголоситись від хвороби, як виникла нова проблема, пов'язана з повномасштабним вторгненням ворога на територію України. Життя продовжується і відповідно потрібно вирішувати питання, як продовжувати навчання зі школярами та студентами. Міністерство освіти і науки України низкою документів (наказ МОН №406 від 16 березня 2020 р.; офіційний лист №1/9-176 від 25 березня 2020 р.) зобов'язує заклади освіти функціонувати у форматі дистанційного навчання (Міністерство освіти і науки України, 2020). Система дистанційного навчання не є новою, але закладам освіти, педагогам бракує досвіду роботи в умовах карантину й повної ізоляції. [1].

У таких умовах стає більш актуальною проблема створення й керування онлайн відео-курсами, посилюється увага до їх характеристик та технологічно-методичних можливостей для освіти здобувачів.

Ідея розробки онлайн відео-курсів не є новою, її дослідженням займалися Т. Шарова [2], Г.М. Кузьменко та О.В. Хорольський [3], О. Самойленко [4], В. Беніньо та Г. Трентін [5], К. Гарріс та Б. Флек [6] та інші.

Онлайн курси є дуже популярними серед учнів (наприклад, як засіб підготовки до ЗНО), студентів та дорослих, оскільки вони допомагають підготуватися до ЗНО та підвищувати свою конкурентоспроможність, а також розширювати діапазон своїх компетенцій [7].

Онлайн освіта дає змогу викладачам та студентам проводити та відвідувати заняття у будь-який зручний час та в будь-якому зручному місці. Це спричиняє більшу продуктивність серед студентів, тому що вони самі можуть визначити час, в який вони найбільш продуктивні, а також дає змогу поєднувати навчання з іншими справами. Доступ та обміну інформацією відбувається легше та швидше. У зв'язку з розвитком сучасного суспільства та робочого середовища, навички цифрової грамотності стають все більш важливими, і онлайн-освіта може допомогти їх розвитку [8]. Діти на окупованих територіях не мають доступу до української освіти, яка може заборонятись для вивчення в школах. Військові, які ще недавно були студентами, а сьогодні боронять нашу країну – також втратили можливість навчатися, поповнювати свої знання фаховими знаннями і підтримувати свої мізки в тонусі. Саме онлайн відео-курси можуть надати змогу таким здобувачам освіти отримувати знання незалежно від місця знаходження та часу, коли можуть скористатися освітніми послугами.

Інтернет є універсальною платформою, що надає широкий спектр послуг (інформаційних, комунікаційних, фінансових, консультативних тощо), які є необхідними для сучасного життя. Розвиток можливостей гіпертекстової розмітки в Інтернеті призводить до з'яви все більшої кількості сайтів, ресурсів та платформ з різною структурою та тематикою [9]. Інтернет є зручним засобом обміну інформацією та універсальним інструментом для розміщення великої кількості інформації, тому створення веб-сайтів та додатків має велике значення. Вони створюють спеціальний інтерфейс взаємодії між користувачами та провайдерами послуг.

Можна зазначити, що використання онлайн-ресурсів у навчальному процесі може вирішити багато завдань, пов'язаних з отриманням нової автентичної інформації. Комбінація тексту, графіки, відео, звукових ефектів та анімації має вплив на емоційну та пізнавальну сфери, що сприяє більш ефективному засвоєнню матеріалу. [10].

Незважаючи на те, що існує велика кількість досліджень серед іноземних та вітчизняних науковців, на сьогодні відсутнє дослідження, у якому були б системно проаналізовані онлайн відео-курси у контексті їх практичного використання. Саме тому доцільно провести аналіз сучасних веб-додатків, визначити їх переваги та недоліки і сформулювати вимоги до веб-додатку опираючись на власний аналіз.

1.2 Визначення переваг офлайн навчання та впровадження їх в онлайн навчання

Офлайн-навчання – це навчання в реальному часі у звичайному класі чи аудиторії, тобто звичний спосіб навчання яким люди користувались до появи навчання в дистанційному режимі. Саме для того, щоб зробити якісний веб-додаток для організації онлайн відео-курсів потрібно визначити переваги та недоліки офлайн навчання і поєднати ці переваги з перевагами використання веб-додатків та інтернету в житті людей. До таких переваг можна віднести наступне [11]:

- візуальний контакт з викладачем та можливість живого спілкування під час пояснення матеріалу;
- живе спілкування з однокурсниками;
- чітко сформований розклад занять і заняття відбуваються і спеціально відведеній для цього аудиторії, в результаті чого випрацьовується систематичність та розвивається асоціативне мислення та рефлексорні дії;
- більший обсяг практичного досвіду;
- легше знайти потрібну та актуальну інформацію;
- менторська підтримка.

Отже, визначено переваги офлайн навчання. А також відомі переваги онлайн навчання, такі як: вільний графік навчання, не потрібно бути

прив'язаним до місця проведення навчання, часто вартість навчання дешевше, ніж офлайн.

Таким чином, веб додаток для створення, публікації та керування онлайн відео-курсами має бути інтерактивним. Тобто такий режим функціонування додатку, коли передбачено практично негайну відповідь системи на запит користувача. Важливо, щоб здобувач освіти мав підказки на всі запитання, які можуть виникнути під час навчання, тому викладач має передбачити корисні матеріали, практичні приклади, мультимедіа, домашні завдання та інше. Плюсом буде спілкування з викладачами в реальному часі, але тут виникає перепона в вигляді часу, так як викладач не може бути постійно онлайн, щоб відповісти студенту відразу після поставленого запитання.

1.3 Порівняльний аналіз аналогів веб-додатку для створення, публікації та керування відео-курсами онлайн

Розробка будь-якого веб-додатку або сайту є творчим і пошуковим процесом. Для створення високоякісного та унікального веб-додатку необхідно вивчити наявні рішення, провести їх аналіз, виявити переваги та недоліки інших ресурсів та врахувати це при створенні власного веб-додатку.

Було проаналізовано значну кількість веб-ресурсів для організації онлайн відео-курсів, а саме: Університет онлайн, Prometheus, EdEra, ВУМ online, MOODLE, Google Meet, Zoom, ВШО та інші. Переваги та недоліки їх буде описано далі.

Першою платформою в Україні для організації онлайн курсів стала платформа Університет онлайн, яка була запущена в 2013 році. Курси мали складатись з відеолекцій, тести та форум для спілкування студентів між собою та з викладачем. Однак продукт не став популярним.

2014 році було запущено український веб-ресурс Prometheus [12], який працює на базі неприбуткової платформи з відкритим кодом OpenEdx, що була розроблена та впроваджена Массачусетським технологічним інститутом та

Гарвардським університетом. Курси на платформі створені українською мовою, безкоштовні й доступні в мережі у будь-який час. Після проходження курсів здобувачі освіти отримують сертифікати.

Після початку карантину серед найпопулярніших додатків для проведення онлайн-занять були такі, як ZOOM [13] та Google Meet [14]. Ці два додатки мають спільну функцію - організацію відеоконференцій в онлайн-режимі. За допомогою цих додатків користувачі можуть створювати зустрічі та приєднуватись до них з будь-якого місця, де є доступ до Інтернету і запрошення на конференцію. Крім того, користувачі можуть спілкуватися один з одним в режимі онлайн, бачити один одного і демонструвати свої екрани, щоб показувати презентаційний матеріал. У додатках також є чат, де користувачі можуть обмінюватися текстовими повідомленнями та файлами. Варто зазначити, що хоча ці додатки мають зручний функціонал, вони призначені виключно для організації відеоконференцій.

Для організації дистанційного навчання багато ЗВО України MOODLE [15]. MOODLE є платформою управління навчанням, яка призначена для об'єднання адміністрації, викладачів та студентів/учнів. Ця система має значний функціональний потенціал, включаючи такі можливості, як:

- здача завдань;
- форуми для дискусій;
- завантаження файлів;
- оцінювання знань;
- обмін повідомленнями;
- календар подій;
- тестування онлайн та ін.

Це не вичерпний перелік функцій платформи, але важливим обмеженням є відсутність можливості для викладачів проводити онлайн відеоконференції зі своїми учнями/студентами.

Twiddla, Miro, AWWApp та На урок. Кожен з них має свій функціонал, який може бути корисним для проведення онлайн занять, такі як

відеоконференції, створення занять або інтерактивні функції. Однак, недоліком цих додатків є те, що весь необхідний функціонал для організації та проведення якісних онлайн курсів розміщується в різних додатках. Необхідно зберігати файли та навчальні матеріали в одному місці, планувати розклад і графік уроків в іншому місці, а проведення відеоконференцій – в третьому. Тому необхідно створити веб-додаток, який міститиме весь необхідний функціонал в одному місці, для того, щоб забезпечити якісне проведення онлайн курсів.

1.4 Технічне завдання на розробку веб-додатку

Веб-додаток для створення, публікації та керування відео-курсами онлайн є додатком для взаємодії викладачів зі здобувачами освіти з наступним функціоналом та можливостями:

- ролі користувачів: адміністратор – користувач, який має можливість керувати налаштуваннями веб-додатку та слідкує за дотриманням правил користувачами; викладач – користувач, який може створювати навчальні відео-курси, завантажувати супроводжуючі матеріали та оцінювати здобувачів освіти; здобувач освіти – користувач, який має доступ до навчальних курсів та матеріалів, обраних ним курсів;
- інформативний інтерфейс користувача – розробка адаптивного веб-додатку, який буде валідний за стандартом W3C та матиме велику швидкість загрузки як на мобільних пристроях, так і на комп'ютерах. Важливо, щоб він коректно працював у всіх сучасних браузерах, включаючи їх останні 3 версії. Інформація доступна лише авторизованим користувачам, для гостей доступна тільки ознайомча інформація про веб-додаток та його цілі;
- система керування контентом – розробка зручної, та простої в користуванні системи для редагування контенту з можливістю управління інформацією для гостей, контактною інформацією та

керуванням користувачами, блокуванням за порушення, створення нових навчальних курсів, завантаження супроводжуючих матеріалів, організація відео-конференцій, завантаження домашніх робіт та ведення журналу оцінювання;

- реєстрація та авторизація користувачів – розробка кабінету користувачів персонально для кожної ролі з доступом до нього при попередній реєстрації з можливістю зміни параметрів входу;
- система запису на навчальні курси – розробка форми для запису здобувачів освіти на навчальні курси;
- обговорення – реалізація форуму/чату для спілкування здобувачів освіти зі здобувачами освіти або з викладачем.

Обов'язково, розробляючи форми введення інформації має бути розроблена перевірка введених полів для забезпечення безпечності введених даних. Має бути передбачена анімація контенту сайту та інтерактиву для взаємодії користувача з веб-додатком та веб-додатку з користувачем.

Передбачити оптимізацію написаного коду та зображень, які будуть використовуватись. Надійність програмного забезпечення повинна бути забезпечена за наступними напрямками:

- забезпечення працездатності компонентів програмно-технічної платформи;
- збереження даних.

Для кращої індексації пошуковими системами має бути розроблено карта сайту sitemap.xml та файл robot.txt. Перевагою буде використання семантичної верстки та розмітки microdata де вона доречна.

При розробці програмного забезпечення перевага надається:

- клієнтська частина: HTML5, CSS3, JavaScript;
- серверна частина: PHP, MySQL.

Мова інтерфейсу українська, з можливістю в подальшому розробки мультимовності, за необхідності.

РОЗДІЛ 2

АНАЛІЗ І ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ

2.1 Обґрунтування вибору засобів для вирішення поставлених задач

Перед початком розробки веб-додатку дуже важливим є визначення засобів для його реалізації. Знаючи вже, які технології будуть використовуватись при розробці веб-додатку – можна визначитись з середовищем розробки IDE, сервером та модулями, які будуть використовуватись.

Вибір IDE залежить від вибору мови програмування, стеку технологій, але також не менш важливим фактором є особисті враження та відчуття при роботі, так як саме на розробку веб-додатку витрачається немала кількість часу і цей процес має бути приємним. Серед актуальних IDE є такі наступні:

- Sublime Text [16] – кросплатформний текстовий редактор та середовище розробки з відображенням синтаксису та можливістю використання розширень. За замовчуванням Sublime Text не містить локалізацій, відмінних від англійської. Має швидкий та простий інтерфейс.
- Visual Studio Code [17] – потужний редактор коду, який доступний на Windows, macOS, Linux. Він підтримує JavaScript, Node.js та інше, а також має багату екосистему розширень для інших мов. Він надає додаткові можливості користувачу, завдяки бібліотеці розширень, які безкоштовно доступні для всіх.
- PhpStorm [18] – інтелектуальний редактор для PHP, HTML і JavaScript, який може аналізувати код на ходу, запобігаючи помилкам у вихідному коді. Цей редактор має автоматизовані засоби рефакторингу для PHP і JavaScript. Крім того, автодоповнення коду в редакторі підтримує специфікації PHP

5.3/5.4/5.5/5.6/7.0/7.1/8.0, включаючи такі елементи, як генератори, співпрограми, простори імен, замикання, типажі та синтаксис коротких масивів. Крім цього, він містить повноцінний SQL-редактор з можливістю редагування результатів запитів."

Є і інші редактори, але ці найбільш популярні серед веб-розробників. Для розробки веб-додатку було обрано PhpStorm, тому що він має безкоштовну ліцензію для студентів і дуже великі функціональні можливості [19], які зробляють кращою якість роботи та пришвидшують процес написання коду:

- програма підтримує останні версії PHP і має вбудовану підтримку різних синтаксичних можливостей, таких як генератори, співпрограми, простори імен, замикання, типажі, синтаксис коротких масивів, доступ до членів класу при інстанціюванні, розіменування масивів при виклику функцій, бінарні літерали, вирази в статичних викликах та інші;
- автоматичне доповнення коду враховує класи, методи, назви змінних, ключові слова PHP, а також часто використовувані імена полів і змінних, залежно від їхнього типу;
- має підтримку стандартів оформлення коду такі як PSR1/PSR2, Drupal, Symfony2, Zend;
- можливість підтримки PHPDoc в PhpStorm дозволяє забезпечити автоматичне доповнення коду, враховуючи анотації @property, @method і @var;
- детектор дубльованого коду;
- PHP Code Sniffer (phpcs), що може перевіряти код на ходу;
- "операції зміни коду (перейменування, введення змінної/константи/поля, вбудовування змінної) для покращення організації коду та його читабельності;

- підтримка редагування шаблонів Smarty включає підсвічування синтаксичних помилок, автодоповнення функцій і атрибутів Smarty, а також автоматичну вставку парних дужок, лапок і закриваючих тегів;
- MVC подання для фреймворків Symfony2 і Yii;
- можливість розпізнавання коду, який міститься в PHAR-архівах.

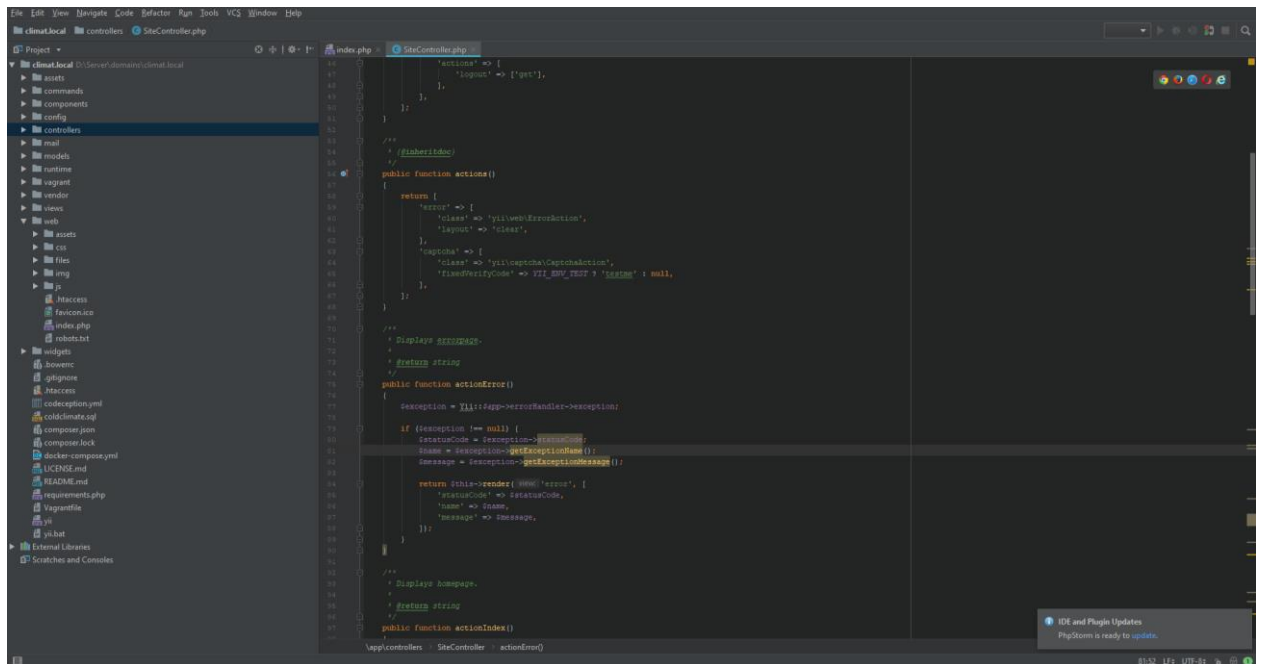


Рисунок 1.2 – Інтерфейс середовища розробки PhpStorm

PhpStorm містить увесь інструментарій WebStorm для роботи з HTML, CSS та JavaScript, що дозволяє виконувати такі дії, як:

- автоматичне доповнення коду для мов програмування JavaScript, HTML та CSS, зокрема для тегів, ключових слів, міток, змінних, параметрів та функцій;
- підтримка HTML5;
- Можливість миттєвого перегляду змін у кодї без перезавантаження сторінки в браузері за допомогою Live Edit;
- підтримка CSS/SASS/SCSS/LESS, що включає автодоповнення коду, підсвічування помилок та валідацію;

- можливість навігації по коду та пошуку використання (здійснення переходу до оголошення/ідентифікатора, пошук використань);
- підтримка ECMAScript Harmony;
- виконання рефакторингу для JavaScript, включаючи перейменування, виділення змінних та функцій, вбудовування змінних та функцій, переміщення та копіювання, безпечне вилучення, а також витяг вбудованого скрипту в окремий файл;
- підтримка відлагоджування JavaScript, а також інтеграція з фреймворками для модульного тестування JavaScript;
- Emmet.

Цей редактор підійде як для розробки серверної так і клієнтської частини веб-додатку. Редактор має зручний інтерфейс, який можна персоналізувати під свої вподобання. Зручно користуватись темною темою, тому що очі не так втомлюються при тривалій роботі як при роботі на світлій темі, де проміння світла монітору більше подразнюють очі.

Веб-додаток, який розроблено на PHP та має підключення до бази даних має бути розміщеним на веб-сервері, для того щоб ці скрипти працювали і можна було тестувати додаток, не завантажуючи його на хостинг. Зараз найпопулярніші веб-сервери, це:

- Apache – веб-сервер з відкритим початковим кодом, використовується як в Unix Windows системах;
- IIS – веб-сервер від компанії Microsoft, і працює він тільки на операційній системі Windows.

Окрім сервера потрібно встановити PHP, MySQL та налаштувати їх коректну роботу. Тому для того, щоб не втрачати час на налаштування локального серверу – існують готові пакети програм, такі як:

- Denver;
- XAMPP;
- AppServ;

– OpenServer.

В основу цих локальних серверів входять Apache, PHP, MySQL, phpMyAdmin. Найбільш зручним є OpenServer, він має неймовірно зручну і продуману керуючу утиліту (рис. 2.2), яка має значні можливості для адміністрування та налаштування всіх компонентів, які вона містить. Локальний сервер, що надається, включає в себе такі модулі, як Apache, Bind, FTP FileZilla, Ghostscript, Nginx, NNCron Lite та Sendmail..

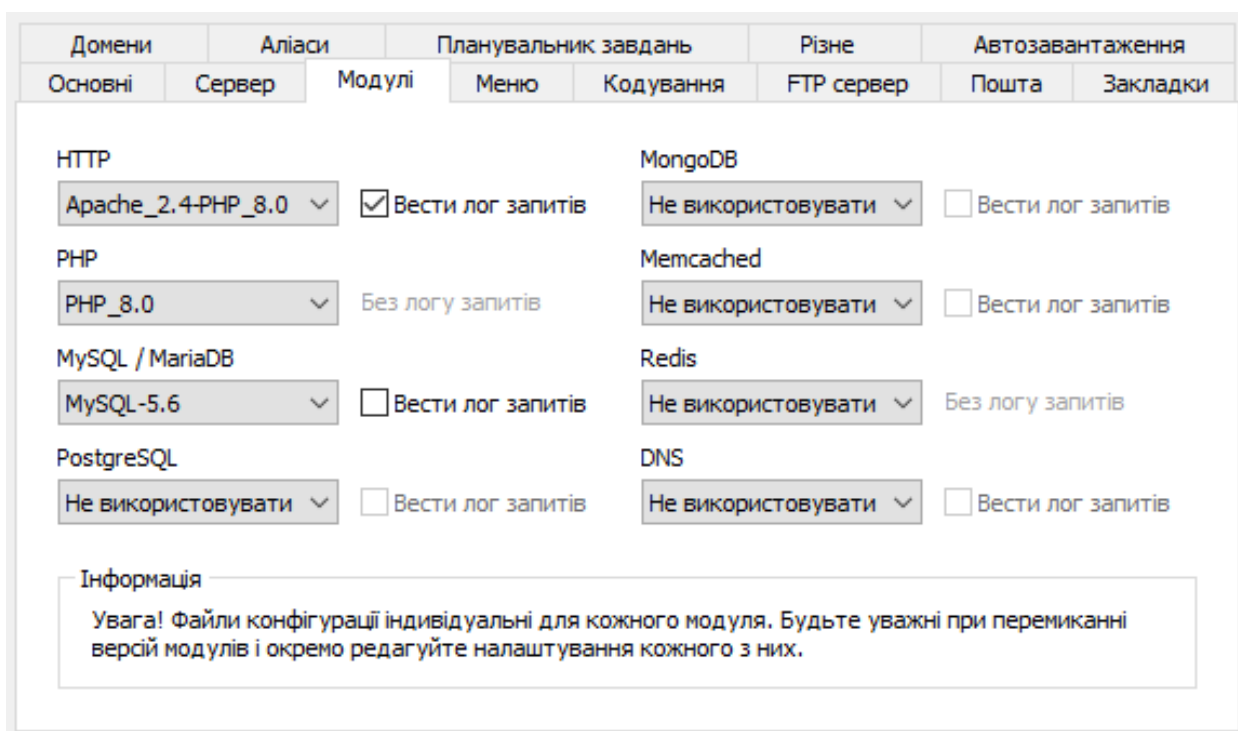


Рисунок 2.2 – Налаштування модулів Open Server Panel

2.2 Проектування логіки, архітектури та поведінки системи

На етапі прототипування системи відбувається проектування логіки, архітектури та поведінки системи. На цьому етапі проводиться визначення структури веб-додатку. Дослідивши в попередньому розділі аналоги було розроблено структуру для інформаційної веб-додатку для створення, публікації та керування відео-курсами онлайн.

Перед тим як отримати фінальну версію зовнішнього вигляду додатку, дизайнер повинен опрацювати значну кількість матеріалів. На першому етапі

визначається цільова аудиторія майбутнього продукту, вказується контекст його використання, описуються типові користувачі та сценарії їх роботи з веб-сервісом, будується інформаційна архітектура. Далі розробляється схематичне розміщення елементів сторінки (wireframes), вибирається найоптимальніший ескіз у процесі тестування. Після цих етапів можна приступати до розробки дизайну інтерфейсу. Ці етапи разом називають проектуванням досвіду взаємодії (UX дизайн) [20].

Цільову аудиторію можна розділити за наступними признаками:

- *стать* – 50% жінки, 50% чоловіки;
- *вік* – здобувачі освіти 12-35 років, викладачі 25-55 років;
- *рівень доходів* – середній та високий дохід;
- *місцезнаходження* – люди з будь-якого міста та країни.

Після визначення цільової аудиторії необхідно надати описи сценаріїв використання продукту користувачами. За допомогою функціоналу веб-додатку здобувач освіти має дізнатися все про навчальні курси, з легкістю знайти курс, який його цікавить, після чого після авторизації він заповнив форму на запис на навчальний курс. Викладач має з легкістю додавати навчальні курси, завантажувати супровідні матеріали.

Після визначення історії користувача та користувацьких сценаріїв проводиться розробка інформаційної архітектури. Це є структура веб-сайту, яка допомагає користувачеві знайти потрібну інформацію та виконати завдання на сайті. Результатом є карта сайту з ієрархією сторінок та їх категоризацією, що полегшує навігацію по сайту.

Цей етап дозволяє визначити як розробку інтерфейсу, так і програмування веб-додатку. Інформаційна архітектура визначає, які компоненти необхідно створити в дизайні та вказує на ті, які будуть згенеровані автоматично, використовуючи розроблені зразки оформлення.

Інформаційна структура веб-додатку зображено на рисунку 2.3. Вона описує ієрархію сторінок веб-додатку з детальним описом їх складових та

можливостей, що дає вже певне розуміння UI дизайнеру та програмісту, які задачі потрібно вирішити і яким способом.

Конкретизація зовнішнього вигляду інтерфейсу відбувається з розробки прототипу (wireframes), результатами є схематичні представлення блоків, що визначають функціонал конкретної сторінки.

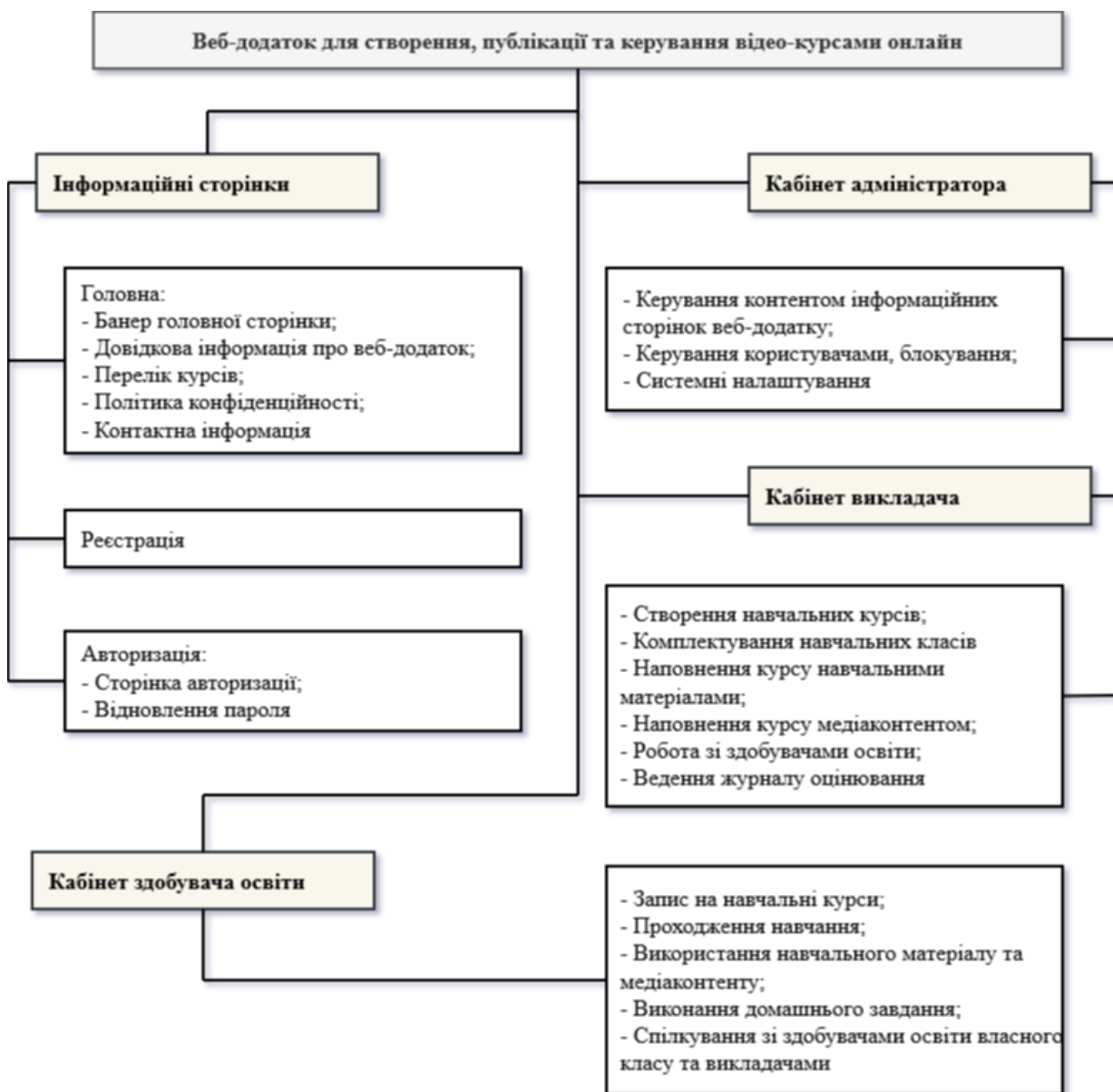


Рисунок 2.3 – Інформаційна структура веб-додатку

Структура веб-додатку, яка буде розроблюватись складається з чотирьох основних компонентів: інформаційні сторінки, які будуть доступні всім типам користувачів та гостям; кабінету адміністратора, який доступний авторизованим користувачам з роллю «Адміністратор»; кабінет викладача, який доступний авторизованим користувачам з роллю «Викладач»; кабінет здобувача освіти, який доступний для авторизованих користувачів з роллю «Здобувач освіти». В залежності від ролі користувача – функціонал кабінету буде відрізнятись.

На рисунку 2.4 представлено UseCase діаграму, події, які виконуються при взаємодії з веб-додатком.



Рисунок 2.4 – UseCase діаграма

ПРЕЦИДЕНТ: НАЛАШТУВАННЯ РОБОЧОГО СЕРЕДОВИЩА

Ектор: адміністратор.

Передумова: перший запуск веб-додатка.

Післяумова: веб-додаток готовий для роботи, наповнення навчальними курсами та проведення онлайн занять.

Сценарій:

1. Веб-додаток встановлюється на сервері.
2. Налаштування параметрів веб-дodatка (тема, часовий пояс та інше).
3. Запуск веб-дodatка та початок роботи.

ПРЕЦИДЕНТ: ПІДТРИМКА КОРИСТУВАЧІВ ВЕБ-ДОДАТКУ

Ектор: адміністратор.

Передумова: отримання заявки від користувачів.

Післяумова: вирішена проблема та задоволений користувач.

Сценарій:

1. Отримання нового повідомлення в чаті підтримки.
2. Відповідь користувачу, який надіслав запит.
3. Вирішення проблеми.

ПРЕЦИДЕНТ: ВЕДЕННЯ БЛОКУ З ОГОЛОШЕННЯМИ ТА НОВИНАМИ

Ектор: адміністратор.

Передумова: є інформація яку потрібно донести до користувачів веб-дodatку.

Післяумова: розміщена інформація в блоці новин та оголошень.

Сценарій:

1. Заповнення полів форми для додання оголошень або новин.
2. Відправка форми та додання інформації до бази даних.
3. Відображення інформації в блоці оголошень та новин.

ПРЕЦИДЕНТ: СТВОРЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Ектор: викладач.

Передумова: дисципліни немає в базі.

Післяумова: дисципліна існує в базі.

Сценарій:

1. Заповнення інформації про дисципліну в форму.

2. Додання інформації про дисципліну в базу даних.

ПРЕЦИДЕНТ: СТВОРЕННЯ УРОКІВ ТА ЗАВАНТАЖЕННЯ ФАЙЛІВ

Ектор: викладач.

Передумова: дисципліна без навчальних уроків.

Післяумова: дисципліна заповнена навчальними уроками та супровідними навчальними матеріалами.

Сценарій:

1. Додання навчальних уроків в базу даних.

2. Наповнення уроків супровідною інформацією, файлами.

3. Створення графіку навчання.

**ПРЕЦИДЕНТ: СТВОРЕННЯ ТА ПРОВЕДЕННЯ ОНЛАЙН
ВІДЕОКОНФЕРЕНЦІЙ**

Ектор: викладач.

Передумова: потрібно провести відеоконференцію із здобувачами освіти.

Післяумова: створене посилання на відеоконференцію, яка відбудеться в запланований час.

Сценарій:

1. Створення посилання на відеоконференцію з паролем входу.

2. Відправка запрошення на конференцію здобувачам освіти.

**ПРЕЦИДЕНТ: ВИКОНАННЯ ТА ЗАВАНТАЖЕННЯ ДОМАШНЬОГО
ЗАВДАННЯ**

Ектор: здобувач освіти

Передумова: отримано домашнє завдання після проведеного заняття.

Післяумова: викладач отримує виконане домашнє завдання.

Сценарій:

1. Завантаження виконаного домашнього завдання на сервер.

2. Перевірка завдання викладачем.

3. Внесення оцінки до журналу оцінювання.

Логіка роботи веб-додатку проста: до запуску веб-додатку адміністратор наповнює інформаційні сторінки. Далі, коли веб-додаток буде в роботі – користувач заходить на сайт, знайомиться з інформацією і переходить на сторінку реєстрації. Викладач створює сторінку викладача і додає навчальні курси. Здобувач освіти створює сторінку здобувача освіти, обирає курс, який йому сподобався, записується на нього і проходить навчання.

РОЗДІЛ 3

РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Алгоритм обробки та аналізу даних

Однією з найбільш поширених парадигм програмування для розробки програмного забезпечення є об'єктно-орієнтоване програмування. Об'єктно-орієнтовані мови програмування надаються з фіксованим набором операцій над об'єктами, що дозволяє зручно додавати нові об'єкти до коду. Ця парадигма програмування базується на чотирьох основних концепціях: інкапсуляція, успадкування, поліморфізм та абстракція. Ці концепції можна реалізувати, створюючи нові класи, що відтворюють існуючі методи, або успадковуючи властивості від інших класів. Таким чином, структура програмного забезпечення може змінюватися, додаючи нові класи та об'єкти, при цьому існуючий код залишається незмінним [21].

Об'єктно-орієнтоване проектування є методологією програмування, яка дозволяє розробникам створювати моделі реальних сценаріїв, що сприяє більш точному і адаптованому до потреб користувачів написанню коду. Особливістю ООП є прямий зв'язок між об'єктами, що існують у реальному світі та кодом, що відображає їх функціональність. Парадигма ООП використовує об'єкти для репрезентації речей у програмі. Наприклад, існує об'єкт під назвою «User» (Користувач), який має різні атрибути користувача додатку, такі як ім'я, по батькові, прізвище, e-mail тощо.

Для веб-додатку для створення, публікації та керування відео-курсами онлайн було виділено такі об'єкти:

- користувачі;
- ролі користувачів;
- навчальні курси;
- навчальні уроки;
- медіа-матеріали до уроку;
- домашні роботи;

- форум;
- запис студентів на навчання;
- обрані курси;
- студентські групи;
- розклад занять.

Модель-вид-контролер (MVC) було обрано як архітектурний шаблон для розробки веб-додатку. MVC – це скорочення від Model-View-Controller, що є популярною архітектурною парадигмою для розробки програмного забезпечення. MVC використовується для розділення логіки програми на три компоненти: модель, представлення і контролер:

- Model (модель) – представляє даний, який програма обробляє і маніпулює. Цей компонент зазвичай містить бізнес-логіку, а також функції для збереження і зчитування даних з бази даних або інших джерел.
- View (представлення) – відповідає за відображення даних для користувача. Це може бути HTML-сторінка, форма, вікно програми або будь-який інший інтерфейс, який користувач може використовувати для взаємодії з додатком.
- Controller (контролер) – це посередник між моделлю і представленням. Він отримує запити від користувача, обробляє їх і виконує відповідні дії в моделі. Після цього контролер оновлює представлення, щоб відобразити зміни.

Застосування MVC-архітектури дозволяє покращити структуру веб-додатку, зробити його більш модульним, зменшити залежність між компонентами і полегшити його тестування та підтримку. Вона також дозволяє розробникам працювати окремо над різними компонентами програми, що спрощує процес розробки і підтримки коду. За допомогою концепцій ООП можна забезпечити повторне використання коду та уникнути дублювання. Шаблон архітектури розробки MVC представлений на рисунку 3.1.

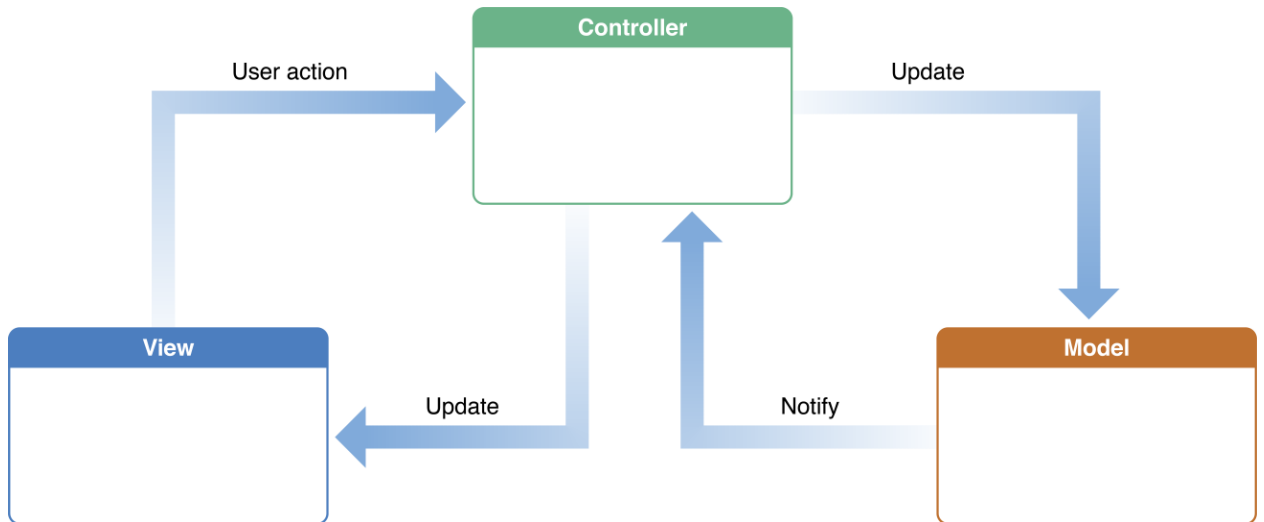


Рисунок 3.1 – Схема роботи архітектури MVC [22]

На діаграмі показано, що користувач взаємодіє з веб-додатком, дані обробляються у контролері та надсилається запит в модель. Модель вносить зміни до бази даних і після виконання запиту повідомляє контролер, що запит виконано. Контролер змінює інтерфейс користувача, забезпечуючи зворотний зв'язок між користувачем та додатком.

При розробці веб-додатку було використано PHP фреймворк Yii2, який підтримує архітектуру MVC. Встановлення фреймворку та інструкція з налаштування описано на офіційному сайті [23]. Файлову структуру додатку представлено на рисунку 3.2.

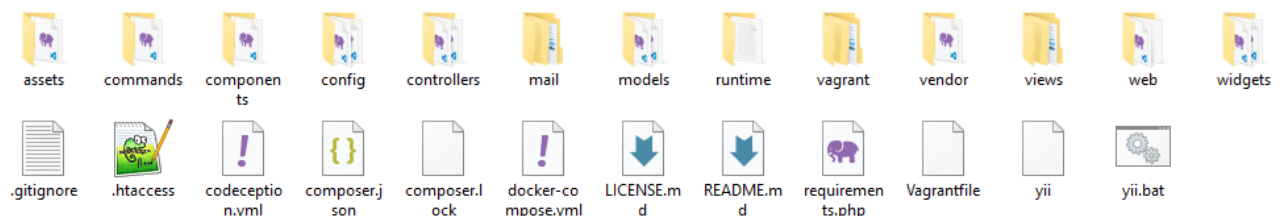


Рисунок 3.2 – Файлова структура проекту

Опис файлової структури проекту представлено в таблиці 3.1.

Таблиця 3.1

Директорія	Опис
<i>assets</i>	Папка з файлами підключення таблиць стилів, скриптів.
<i>commands</i>	Консольні додатки для реалізації фонових та супроводжувальних задач.
<i>components</i>	Компоненти веб-додатку, методи, або класи, до яких можна звертатись в різних частинах коду.
<i>config</i>	Налаштування сайту, підключення компонентів, модулів, підключення до бази даних.
<i>controllers</i>	Контролери.
<i>mail</i>	Налаштування шаблонів e-mail повідомлень.
<i>models</i>	Моделі.
<i>runtime</i>	Тимчасові файли, такі як логи помилок, кеш та інше.
<i>vagrant</i>	Налаштування для роботи з віртуальною машиною.
<i>views</i>	Представлення.
<i>web</i>	Папка зі статичними файлами: зображення, таблиці стилів, шрифти, скрипти та інше.
<i>widgets</i>	Віджети проекту.

Циклічний процес характеризує розробку веб-додатку. Після налаштування фреймворку Yii2, створюється таблиця даних в базі даних, про що можна докладніше прочитати в пункті 3.2 «Проектування бази даних». Для цієї таблиці створюється модель, а потім контролер та представлення.

3.2 Проектування бази даних

Проектування бази даних є одним з найбільш складних етапів створення бази даних, оскільки він не має чіткого початку і кінця. Цей етап є менш структурованим, порівняно з аналізом вимог до бази даних або розробленням додатків, на що зазначають багато провідних фахівців. Якщо всі

етапи створення бази даних можуть перекриватися в послідовності, то проектування бази даних може перекриватися з усіма іншими етапами.

Таблиці, які будуть включені до моделі даних, були визначені і представлені у таблиці 3.2. При цьому, для кожної таблиці слід розглянути три головні питання:

1. Для збору, якої інформації потрібна ця таблиця?
2. Хто повинен заповнювати її?
3. У чому полягає основна мета її використання?

Таблиця 3.2

№ п/п.	Назва таблиці	Інформація	Дії з даними
1.	auth_assignment	Таблиця, в якій зберігається інформація про ролі користувачів, а саме співвідношення ідентифікатора користувача і його ролі. Інформація в цю таблицю додається при реєстрації нового користувача.	додання редагування видалення
2.	auth_item	Таблиця, в якій зберігається інформація про ролі користувачів, які доступні в додатку. Керувати нею може лише розробник веб-додатку.	додання редагування видалення
3.	classes	Таблиця, в якій зберігається інформація про навчальні групи до курсу. Інформацією в цій таблиці може керувати тільки викладач для власного навчального курсу.	додання редагування видалення
4.	class_students	Таблиця, в якій зберігається інформація про нові записи студентів на навчальні курси, інформація в які додається автоматично при записі на курс. Редагувати інформацію можуть викладачі власних курсів.	додання редагування видалення
5.	content	Таблиця, в якій зберігається контентна інформація головної сторінки, політики сайту та контактної інформації.	редагування видалення

№ п/п.	Назва таблиці	Інформація	Дії з даними
		Інформацією керувати може адміністратор сайту.	
6.	courses	Таблиця, в якій зберігається інформація про навчальні курси. Керувати таблицею можуть викладачі.	додавання, редагування, видалення
7.	favorites	Таблиця, в якій зберігається інформація про обрані студентом курси. Керувати таблицею можуть здобувачі освіти.	додавання, редагування, видалення
8.	forum	Таблиця, в якій зберігається інформація форуму навчального курсу. Керувати таблицею можуть викладачі та здобувачі освіти.	додавання, редагування, видалення
9.	homework	Таблиця, в якій зберігається інформація про виконані домашні роботи студентами та відповіді на ці роботи від викладача.	додавання, редагування, видалення
10.	lessons	Таблиця, в якій зберігається інформація про навчальні уроки до курсу. Інформацією в цій таблиці може керувати тільки викладач для власного навчального курсу.	додавання, редагування, видалення
11.	marks	Таблиця, в якій зберігаються оцінки здобувачів освіти за відповідний урок. Керувати таблицею можуть викладачі.	додавання, редагування, видалення
12.	media	Таблиця, в якій зберігається інформація до навчального уроку (текст, зображення, відео, файли, трансляції). Інформацією в цій таблиці може керувати тільки викладач для власного навчального курсу.	додавання, редагування, видалення
13.	timetable	Таблиця, в якій зберігається інформація про початок та кінець навчального уроку для відповідної групи. Інформацією в цій таблиці може керувати тільки викладач для власного навчального курсу.	додавання, редагування, видалення

№ п/п.	Назва таблиці	Інформація	Дії з даними
14.	user	Таблиця, в якій зберігається інформація про користувачів веб-додатку, а саме їх персональні дані для входу в кабінет. Редагувати, інформацію може лише користувач з доступом до персонального кабінету.	додавання, редагування, видалення

На рисунку 3.3 показано повну модель бази даних, що відображає взаємозв'язок між таблицями.

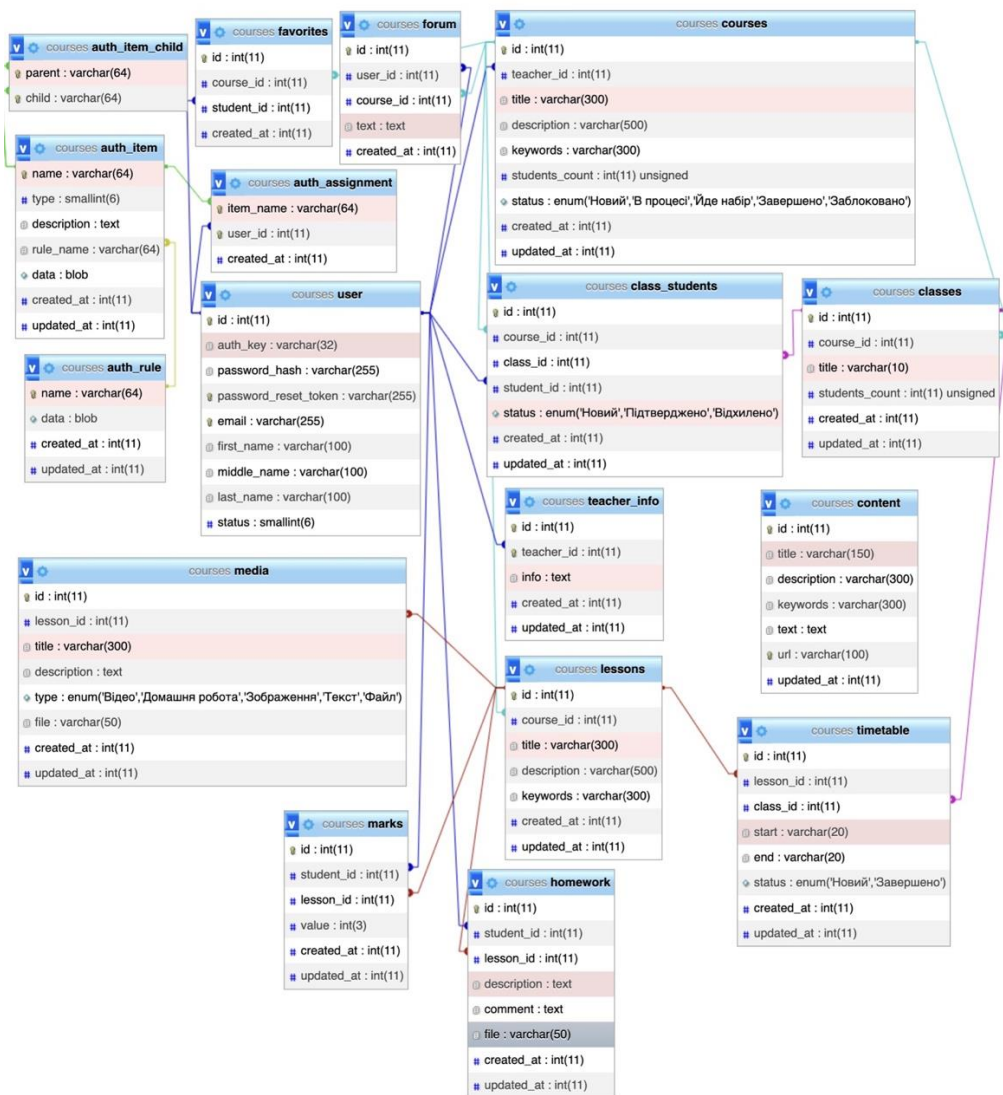


Рисунок 3.3 – ER-діаграма, повна модель бази даних

Сучасні вимоги до веб-програм полягають у можливості зберігати, обробляти та надавати користувачу певну інформацію. Для цих в якості СУБД було обрано реляційну БД MySQL. Для роботи з такою БД в об'єктно-орієнтованій архітектурі існує патерн ActiveRecord – шаблон проектування, що використовується при реалізації доступу до реляційних баз даних [24]. Для зручного користування БД використовується – зручний веб-інтерфейс для керування БД MySQL.

3.3 Розробка інтерфейсу системи

При розробці веб-додатку особливу увагу вимагає інтерфейс. Дизайн додатку та його якісне оформлення – це показник іміджу компанії, вираз її відношення до потенційних клієнтів. Якщо продукт виглядає красиво і оригінально, значить, його власнику важливо справити хороше враження на клієнтів, і він зацікавлений у вашій увазі і оцінці.

Використовуючи сучасні технології можна створити якісну програму, акуратно і красиво оформлену, приємну для очей. Що стосується азів, то тут важливо врахувати вагу графічних файлів, оскільки при високій вазі графіки сторінки сайту будуть довго завантажуватися, а це негативно відіб'ється на ефективності роботи користувачів.

Крім того, при оформленні дизайну важливим є правильне поєднання кольорів, стежити за колірним балансом, грамотно підбирати пропорції, освітлення і обсяг графічних елементів. Дизайн додатку повинен розроблятися з урахуванням тематики і цілей функціонування веб-ресурсу. Основними кольорами було обрано такі:

- основний колір – #7EC384;
- основний альтернативний колір – #3F7856;
- колір заголовків – #101D40;
- колір тексту – #6D8C8E;
- колір тексту світлий – #D5D5D5;
- колір фону – #FFFFFF.

Основним шрифтом веб-сайту було обрано «Montserrat».

Розробку інтерфейсу можна розділити на 2 етапи. Спочатку було розроблено загальні шаблони користувацького інтерфейсу, де було зроблено макети сторінок без основного контенту: шапка, сайдбар, контент та підвал сайту. На рисунку 3.5 представлено макет сторінок сайту, який буде спільний для всіх сторінок сайту.

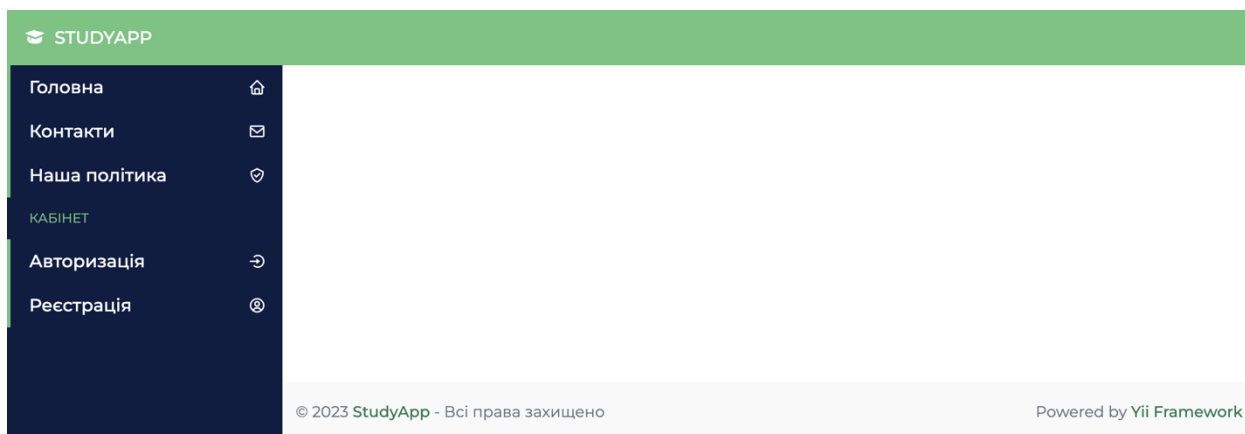


Рисунок 3.5 – Макет веб-додатку (зелений блок – шапка, темно-синій блок – сайдбар, білий блок – контент, сірий блок – підвал)

Заголовок сторінок та хлібні кришки сайту генеруються автоматично для кожної сторінки. В верстці хлібних кришок використано мікророзмітку microdata. Спільними елементами на всіх сторінках є логотип, меню, заголовок сторінки, шапка та підвал сайту. Меню відрізняється в залежності від ролі користувача. Лістинг коду хлібних кришок представлено в лістингу 3.1.

Лістинг 3.1 – Код хлібних кришок

```
<ul class="breadcrumb" itemscope=""
itemtype="http://schema.org/BreadcrumbList">
  <li class="breadcrumb__item" itemprop="itemListElement" itemscope=""
itemtype="http://schema.org/ListItem">
    <a class="breadcrumb__link" href="/" itemprop="item">
      <span class="breadcrumb__name" itemprop="name">Головна</span>
    </a>
    <meta itemprop="position" content="1">
  </li>
  <li class="breadcrumb__item" itemprop="itemListElement" itemscope=""
itemtype="http://schema.org/ListItem">
    <a class="breadcrumb__link" href="/content" itemprop="item">
```



```
        <span class="breadcrumb__name" itemprop="name">Контент</span>
    </a>
    <meta itemprop="position" content="2">
</li>
</ul>
```

Після розробки макету сторінок розроблювались макети контенту. При розробці інтерфейсів верстка робилась за методологією ВЕМ, контент розбивався на незалежні блоки, які можна використовувати в будь-якій частині коду і вони будуть відображатись однаково. Кожен компонент – це блок, який складається з елементів та може мати модифікатори, класи мають наступний вигляд: `block__element--modifier`. Більш детальний опис користувацьких інтерфейсів буде продемонстровано в розділі 4 «Тестування веб-додатку».

При розробці клієнтської частини для верстки було використано HTML5, CSS3 та JavaScript (бібліотека jQuery). Інтерфейс простий і зрозумілий для користувачів цільової аудиторії сайту. Дані для контенту беруться з бази даних, та виводяться в представленні за допомогою контролерів. Для кожної моделі розроблено представлення з відповідною назвою.

3.4 Програмна реалізація логіки веб-додатку

Розробку веб-додатку, а саме реалізацію логіки можна описати наступним чином: після того як спроектовано базу даних та встановлено зв'язки між таблицями – відбувається опис моделей основних об'єктів. Такими об'єктами можуть бути: модель сховища даних, яка описує таблицю з бази даних; модель для зв'язку користувача з додатком, наприклад, опис форми.

Було розроблено моделі таблиць баз даних, в яких описано функцію `tableName()`, яка повертає назву таблиці бази даних. Описується метод `rules()`, яка задає властивості параметрів полів таблиці: тип даних, довжина поля, обов'язковість введення полів та інше. У методі `attributeLabels()` вказуються підписи атрибутів, які будуть використовуватись в формі введення даних, таблиці виводу і т.д. В методі `behaviors()` вказуються події, які будуть

виконуватись після виконання певних дій, в нашому випадку після створення нового елемента, або після редагування існуючого. Також існують методи `getModelName()`, які є гетерами – методи назви яких починаються на «get». Використовуються ці методи для отримання даних зі зв'язаних таблиць. Приклад лістингу коду моделі представлено нижче.

Лістинг 3.2 – Приклад коду моделі таблиці

```
class Content extends \yii\db\ActiveRecord
{
    public static function tableName()
    {
        return 'content';
    }
    // Правила полів
    public function rules()
    {
        return [
            [['title', 'text', 'url'], 'required'],
            [['text'], 'string'],
            [['updated_at'], 'integer'],
            [['title'], 'string', 'max' => 150],
            [['description', 'keywords'], 'string', 'max' => 300],
            [['url'], 'string', 'max' => 100],
            [['url'], 'unique'],
        ];
    }
    // Атрибути полів
    public function attributeLabels()
    {
        return [
            'title' => 'Назва сторінки',
            'description' => 'Опис',
            'keywords' => 'Ключові слова',
            'text' => 'Вміст',
            'url' => 'URL',
            'updated_at' => 'Відредаговано',
        ];
    }
}
```

У випадку з моделями для опису користувацьких форм, таких як форма зворотного зв'язку, авторизації, реєстрації і відновлення паролю описуються ті ж самі методи, крім методу `tableName()`, а також описуються властивості моделі.

Після того як описано моделі – в папці `controllers` описуються контролери, а в папці `views` знаходяться представлення для всіх сторінок. Таким чином в контролері відбувається звернення до моделі і отримана інформація виводиться на головній сторінці. Таким чином при переході на сторінку `http://courses.local/site/contact` відбудеться звернення до контролера `SiteController.php`, в якому виконуються функції в `actionContact()`, а далі в залежності від того, яка сторінка вказана для виводу представлення – буде відкрито файл з цією назвою в папці «`views`» - «`contact`». Папка називається так як контролер. Також є віджети – такі компоненти, які запускаються в представленні і мають своє власне представлення. Ці компоненти дають змогу не дублювати однаковий код, таким чином генерація хлібних кришок і виведення інформативних повідомлень виконана за допомогою віджетів. Лістинг коду буде представлено в додатку А.

Важливим компонентом для SEO оптимізації сайту є генерація метатегів та `OpenGraph`. Лістинг 3.3 описує генерацію `OpenGraph` для соціальних мереж `Facebook`, `Google+` та `Twitter`.

Лістинг 3.3 – Код компонента генерації `OpenGraph`

```
<?php
namespace app\components;

use Yii;

class SeoComponent{
    public static function putOpenGraphTags($properties) {
        foreach($properties as $item => $value)
        {
            Yii::$app->view->registerMetaTag([
                'property' => $item,
                'content' => $value,
            ]);
        }
    }
}
```

```

    }

    public static function putGooglePlusMetaTags($properties) {
        foreach($properties as $itemprop => $content)
        {
            Yii::$app->view->registerMetaTag([
                'itemprop' => $itemprop,
                'content' => $content,
            ]);
        }
    }

    public static function putTwitterMetaTags($properties) {
        foreach($properties as $name => $content)
        {
            Yii::$app->view->registerMetaTag([
                'name' => $name,
                'content' => $content,
            ]);
        }
    }
}

```

РОЗДІЛ 4

ТЕСТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ СТВОРЕННЯ, ПУБЛІКАЦІЇ ТА КЕРУВАННЯ ВІДЕО-КУРСАМИ ОНЛАЙН

Перед тестуванням додатку необхідно ретельно продумати всі можливі варіанти його використання, щоб відповідно спланувати процес тестування. Умовно процедуру можна розділити на наступні етапи:

1. тестування розташування інтерфейсу;
2. тестування функцій;
3. тестування безпеки.

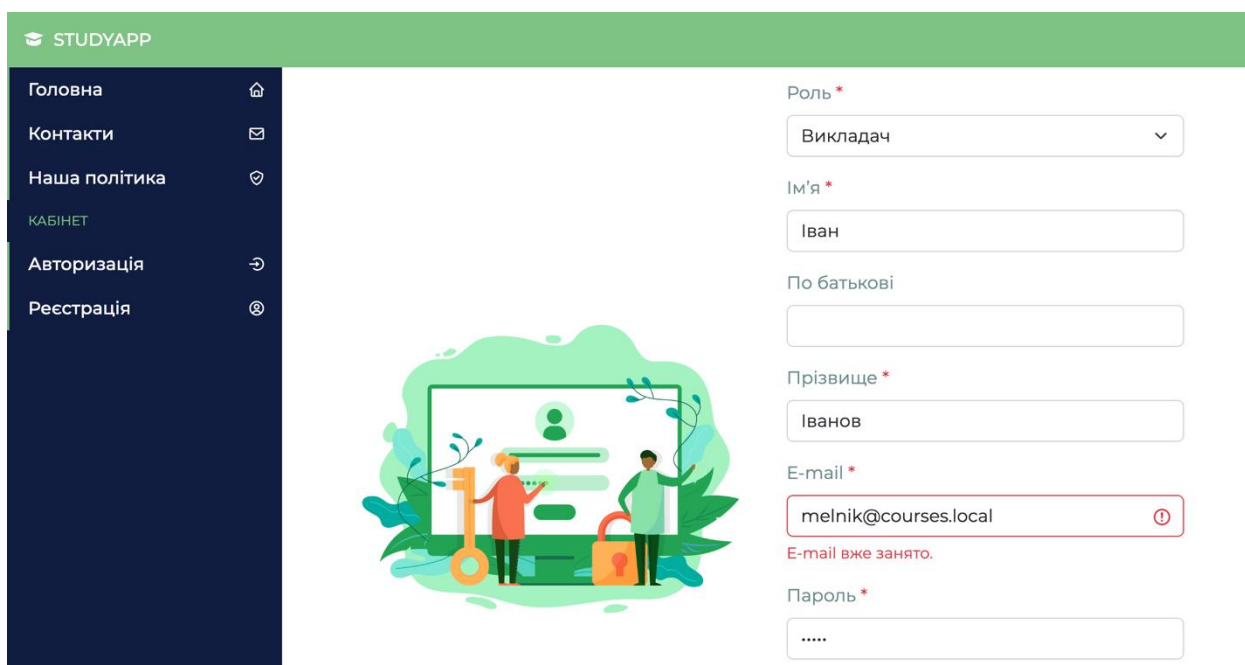
Визначившись з основними етапами – можна провести тестування, провести аналіз і перевірити чи все правильно працює. Однією із переваг навчання онлайн є те, що можна навчатися в будь-якому місці і з будь-якого пристрою, тому важливо, щоб інтерфейс користувача відображався на всіх пристроях коректно.

Було проведено тестування атаптивності, кросбраузерності і валідності коду. Веб-додаток коректно відображається на всіх пристроях, починаючи з мобільних пристроїв з шириною екрану від 320px. Було протестовано у більшості сучасних браузерів і останніх їх 2-3 версіях, це охоплює близько 90% користувачів – веб-додаток коректно працював з усіма ними. На валідність код було перевірено в валідаторі W3C, тест пройдено успішно. Крім цього було проведено тестування на швидкість завантаження сторінок. Додаток знаходиться в зеленій зоні Google Page Speed на всіх пристроях. Помилки в розташуванні елементів не виявлено, тому можна перевіряти функції веб-додатку.

Було перевірено роботу всіх веб сторінок та елементів взаємодії з користувачами. Таким чином було перевірено основні функції: авторизація, реєстрація користувача, відновлення паролю, створення нового навчального курсу та наповнення його основною інформацією, виконано перевірку запису на навчання, розподіл по навчальним групам здобувачів освіти. Здобувачі

освіти можуть успішно проходити навчальні курси. Інформація зберігається в базі даних. Персональні дані, типу пароля хешуються, для того щоб не відбулося взлому персональних сторінок користувачів.

Перевіривши функціонал – було проведено тестування безпеки. Особлива увага приділяється тестуванню форм. При реєстрації користувача відбувається перевірка і пошук користувачів з таким e-mail, а після проходження реєстрації надходить повідомлення про проходження реєстрації. Якщо введено якісь дані неправильно користувача сповіщають інформативні поля, або повідомлення, приклад на рисунку 4.1.



The screenshot shows the 'STUDYAPP' registration interface. On the left is a dark blue navigation menu with items: Головна, Контакти, Наша політика, КАБІНЕТ, Авторизація, and Реєстрація. The main content area features a registration form with the following fields: 'Роль *' (role) with a dropdown menu set to 'Викладач'; 'Ім'я *' (name) with the value 'Іван'; 'По батькові' (patronymic) which is empty; 'Прізвище *' (surname) with the value 'Іванов'; 'E-mail *' with the value 'melnik@courses.local', which is highlighted in red and has a red error icon; and 'Пароль *' (password) with masked characters '.....'. Below the email field, a red error message reads 'E-mail вже занято.' (Email is already taken). In the center of the form is an illustration of two people standing in front of a screen displaying a user profile, with a key and a padlock symbolizing security.

Рисунок 4.1 – Проведення тестування безпеки веб-додатку

Таким чином тестування було успішно виконано, додаток коректно відображається для більшості користувачів, всі функції коректно працюють і безпека системи на належному рівні. Поля форм в разі помилок підсвічуються, інтерфейс інтерактивний і сповіщає користувача повідомленнями, після виконання дій.

ВИСНОВКИ

В результаті виконання дипломної роботи було досягнуто всі поставлені задачі. Спочатку в ході роботи було досліджено предметну область, вивчено сферу онлайн навчання, на скільки розвинена нині дана сфера та з якими проблемами стикаються викладачі та здобувачі освіти. Провівши аналіз предметної області було обґрунтовано чому важлива розробка веб-додатку для створення, публікації та керування відео-курсами онлайн.

Далі було проведено аналіз існуючих сайтів та додатків. Виділено основні переваги та недоліки даних систем, які в подальшому були враховані при розробці веб-додатку для організації відео-курсів. Після проведення аналізу аналогів було описано технічне завдання на розробку веб-додатку, виділено типи користувачів та функціонал додатку.

Далі було розроблено прототип, який детально описує логіку та архітектуру веб-додатку. Для цього було проведено дослідження цільової аудиторії, поведінку користувачів та інше. Отриманий прототип допоможе не тільки дизайнерам в розробці макету інтерфейсу, а й розробникам алгоритмів для правильного розуміння завдання.

В якості архітектури було обрано MVC. Веб-додаток розроблено за допомогою Yii2 фреймворку. Він зручний тим, що його легко розширювати, тому додати новий функціонал не визве ніяких труднощів, а також легше буде працювати в команді, коли розробники будуть працювати кожен зі своїм модулем.

Завершальним етапом дипломної роботи є розробка програмного продукту: верстка інтерфейсів, проектування бази даних та розробка алгоритмів. Розроблений програмний продукт було протестовано і зроблено висновок, що веб-додаток зручний, стильний та функціональний. Охоплює весь спектр послуг, необхідних для проведення онлайн відео-курсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Петренко С. В. СУТНІСТЬ ТА ОСОБЛИВОСТІ УКРАЇНСЬКИХ ПЛАТФОРМ МАСОВИХ ВІДКРИТИХ ОНЛАЙН-КУРСІВ (МВОК) // Інноватика у вихованні. 2020., вип. 11 Т. 2. С. 165-173. DOI: <https://doi.org/10.35619/iuu.v2i11.260>
2. Шарова Т. Цифровий формат освіти та комунікація: огляд онлайн курсів / Т. Шарова; Академічна культура дослідника в освітньому просторі: європейський та національний досвід. – м. Суми. – 144-149 с.
3. Кузьменко Г. М., Хорольський О. В. Масові відкриті онлайн-курси у контексті трансформації вищої освіти України. – 2015.
4. Самойленко О. МООС-платформи як інструмент інформальної освіти дорослих // Педагогічні науки: теорія, історія, інноваційні технології. – 2019. – №. 4. – С. 103-116.
5. Benigno V., Trentin G. The evaluation of online courses // Journal of computer assisted learning. – 2000. – vol. 16. – №. 3. – pp. 259-270. DOI: <https://doi.org/10.1046/j.1365-2729.2000.00137.x>
6. Garris C. P., Fleck B. Student evaluations of transitioned-online courses during the COVID-19 pandemic // Scholarship of Teaching and Learning in Psychology. – 2022. – vol. 8. – №. 2. – pp. 119. DOI: <https://doi.org/10.1037/stl0000229>
7. Коваленко А. С., Шарова Т. М. ОНЛАЙН НАВЧАННЯ НА ВУМ: ОСОБЛИВОСТІ ТА ПЕРЕВАГИ // XXII Міжнародна науково-практична інтернет-конференція "НАУКА ТА ОСВІТА: ДОСЯГНЕННЯ ТА СТРАТЕГІЇ РОЗВИТКУ" / ГРОМАДСЬКА ОРГАНІЗАЦІЯ «НАУКА ТА ОСВІТА БЕЗ КОРДОНІВ» Ч.1. Запоріжжя, 2019.№ 1. С. 51-56.
8. Вдовиченко С. В., Устенко І. В., Апенько Н. В. Онлайн навчання // Proceedings of the 13th International Scientific and Practical Conference «Science and Practice: Implementation to Modern Society» / Scientific Collection «InterConf». Manchester, 2022.№ 128. С. 79-80.

9. Лісовська А. Ю., Калита А. А. Контент веб-сайтів і їхня структура // Молодий вчений. 2019., №74(10). С. 166-170.
10. Карпа І. Використання online-ресурсів при вивченні англійської мови студентами ВНЗ / І. Карпа // Молодь і ринок. 2013. № 2. С. 62-65.
11. Навчання: Оффлайн чи Онлайн? // Комп'ютерна школа Hillel: [Веб-сайт]. URL: <https://blog.ithillel.ua/articles/navchannia-offlain-chy-onlain> (дата звернення: 15.02.2023).
12. Prometheus: [Веб-сайт]. URL: <https://prometheus.org.ua/> (дата звернення: 15.02.2023).
13. Zoom Video Communications, Inc: [Website]. URL: <https://zoom.us/> (viewed on: 15.02.2023).
14. Google Meet: [Website]. URL: <https://meet.google.com/> (viewed on: 15.02.2023).
15. Moodle LMS: [Website]. URL: <https://moodle.org/> (viewed on: 15.02.2023).
16. Sublime Text: [Website]. URL: <https://www.sublimetext.com/> (viewed on: 18.02.2023).
17. Visual Studio Code: [Website]. URL: <https://code.visualstudio.com/> (viewed on: 18.02.2023).
18. PhpStorm // JetBrains: [Website]. URL: <https://www.jetbrains.com/phpstorm/> (viewed on: 18.02.2023).
19. PhpStorm – Вікіпедія // Wikipedia: [Веб-сайт]. URL: <https://uk.wikipedia.org/wiki/PhpStorm> (дата звернення: 18.02.2023).
20. Мосіюк О. О. UX проектування сайтів як основа підготовки спеціалістів web-дизайну // Наукові записки Бердянського державного педагогічного університету. Сер.: Педагогічні науки. – 2017. – №. 3. – С. 201-206.
21. Порівняння об'єктно-орієнтованої та функційної парадигм програмування у проектуванні програмного забезпечення / А. І. Коваль, О. М.

Яшина, Г. І. Радельчук, Ю. В. Форкун // Вісник Хмельницького національного університету. Технічні науки. - 2021. - № 3. - С. 34-38.

22. Коваль А. І. та ін. Порівняння об'єктно-орієнтованої та функційної парадигм програмування у проектуванні програмного забезпечення. 2021.

23. Yii PHP Framework: [Web-site]. URL: <https://www.yiiframework.com> (accessed: 27.02.2023).

24. Ханджанов Д. О., Єфіменко А. А. ОСНОВНІ ПАТТЕРНИ РОБОТИ З РЕЛЯЦІЙНОЮ БАЗОЮ ДАНИХ В ВЕБ АРХІТЕКТУРІ. – 2017.

ДОДАТОК А

ЛІСТИНГ КОДУ ВЕБ-ДОДАТКУ

Компонент генерації OpenGraph, “SeoComponent.php”

```
<?php
namespace app\components;
use Yii;

class SeoComponent
{
    public static function putOpenGraphTags($properties)
    {
        foreach ($properties as $item => $value) {
            Yii::$app->view->registerMetaTag([
                'property' => $item,
                'content' => $value,
            ]);
        }
    }

    public static function putGooglePlusMetaTags($properties)
    {
        foreach ($properties as $itemprop => $content) {
            Yii::$app->view->registerMetaTag([
                'itemprop' => $itemprop,
                'content' => $content,
            ]);
        }
    }

    public static function putTwitterMetaTags($properties)
    {
        foreach ($properties as $name => $content) {
            Yii::$app->view->registerMetaTag([
                'name' => $name,
                'content' => $content,
            ]);
        }
    }
}
```

Віджет повідомлень, “widgets/Alert.php”

```
<?php
namespace app\widgets;
use Yii;

/**
 * Віджет сповіщення відображає повідомлення з сеансу flash. Відображаються
 всі флеш-повідомлення
 *
 */
class Alert extends \yii\bootstrap5\Widget
{
    /**
     * @var масив конфігурації типів сповіщень для флеш-повідомлень.
     */
    public $alertTypes = [
        'error'    => 'alert-danger',
        'danger'   => 'alert-danger',
        'success'  => 'alert-success',
        'info'     => 'alert-info',
        'warning'  => 'alert-warning'
    ];

    public function run()
    {
        $session = Yii::$app->session;
        $appendClass = isset($this->options['class']) ? ' ' . $this->options['class'] : '';

        foreach (array_keys($this->alertTypes) as $type) {
            $flash = $session->getFlash($type);

            foreach ((array) $flash as $i => $message) {
                echo \yii\bootstrap5\Alert::widget([
                    'body' => $message,
                    'closeButton' => $this->closeButton,
                    'options' => array_merge($this->options, [
                        'id' => $this->getId() . '-' . $type . '-' . $i,
                        'class' => $this->alertTypes[$type] . $appendClass,
                    ]),
                ]);
            }
        }
    }
}
```

```

    }

    $session->removeFlash($type);
}
}
}
}

```

Віджет повідомлень, “widgets/BreadcrumbsWidget.php”

```

<?php

namespace app\widgets;
use yii;
use yii\base\InvalidConfigException;
use yii\helpers\ArrayHelper;
use yii\helpers\Html;
use yii\widgets\Breadcrumbs;

class BreadcrumbsWidget extends Breadcrumbs
{
    public $itempropPosition = 1;

    public function run()
    {
        if (empty($this->links)) {
            return;
        }

        $links = [];

        $replacement = ['>{link}' => ' class="breadcrumb__item"
itemprop="itemListElement" itemscope
itemtype="http://schema.org/ListItem">{link}{position}'];
        $this->itemTemplate = strtr($this->itemTemplate, $replacement);
        $this->activeItemTemplate = strtr($this->activeItemTemplate,
        $replacement);

        if ($this->homeLink === null) {
            $links[] = $this->renderItemMarkup(
                [
                    'label' => Yii::t('yii', 'Home'),
                    'url' => Yii::$app->homeUrl,
                ],
            );
        }
    }
}

```

```

        ],
        $this->itemTemplate,
        $this->itempropPosition
    );
} elseif ($this->homeLink !== false) {
    $this->homeLink['template'] = isset($this->homeLink['template']) ? strstr($this->homeLink['template'], $replacement) : $this->itemTemplate;
    $links[] = $this->renderItemMarkup(
        $this->homeLink,
        $this->homeLink['template'],
        $this->itempropPosition
    );
}

foreach ($this->links as $link) {
    if (!is_array($link)) {
        $link = ['label' => $link];
    }

    $links[] = $this->renderItemMarkup(
        $link,
        isset($link['url']) ? $this->itemTemplate : $this->activeItemTemplate,
        ++$this->itempropPosition
    );
}

echo Html::tag(
    $this->tag,
    implode('', $links),
    array_merge(
        $this->options,
        ["itemscope itemtype" =>
"http://schema.org/BreadcrumbList"]
    )
);
}

protected function renderItemMarkup($link, $template, $position)
{

```

```

        $encodeLabel = ArrayHelper::remove($link, 'encode', $this->encodeLabels);

        if (array_key_exists('label', $link)) {
            $label = Html::tag('span', $encodeLabel ?
                Html::encode($link['label']) : $link['label'], ['itemprop' => 'name']);
        } else {
            throw new InvalidConfigException('Елемент "label" є обов'язковим для кожного посилання.');
```

```

        }

        if (isset($link['template'])) {
            $template = $link['template'];
        }

        if (isset($link['url'])) {
            $options = $link;
            unset($options['template'], $options['label'],
                $options['url']);
            $link = Html::a($label, $link['url'], array_merge($options,
                ['class' => 'breadcrumb__link', 'itemprop' => 'item']));
        } else {
            $link = Html::tag('span', $label, ['itemprop' => 'item']);
        }

        return strtr(
            $template,
            [
                '{link}' => $link,
                '{position}' => Html::tag('meta', '', ['itemprop' =>
                    'position', 'content' => $position])
            ]
        );
    }
}

```

Модель «Користувачі», файл «models/User.php»

```

<?php
namespace app\models;
use Yii;
use yii\base\NotSupportedException;

```

```

use yii\db\ActiveRecord;
use yii\web\IdentityInterface;

class User extends ActiveRecord implements IdentityInterface
{
    const STATUS_DELETED = 0;
    const STATUS_ACTIVE = 10;
    const STATUS_BANNED = 99;
    public $password;

    public static function tableName()
    {
        return '{{%user}}';
    }

    public function rules()
    {
        return [
            [['password_hash', 'email', 'first_name', 'last_name'],
            'required'],
            [['password_hash', 'password_reset_token', 'email'], 'string',
            'max' => 255],
            [['auth_key'], 'string', 'max' => 32],
            [['first_name', 'middle_name', 'last_name'], 'string', 'max' =>
            100],
            [['password'], 'string', 'min' => 4],
            ['status', 'default', 'value' => self::STATUS_DELETED],
            ['status', 'in', 'range' => [self::STATUS_ACTIVE,
            self::STATUS_BANNED, self::STATUS_DELETED]],
            [['password_reset_token'], 'unique'],
            [['email'], 'email'],
            [['email'], 'unique'],
        ];
    }

    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'auth_key' => 'Ключ авторизації',
            'password' => 'Пароль',
            'password_hash' => 'Пароль',
        ];
    }
}

```



```

        'password_reset_token' => 'Токен скидування паролю',
        'email' => 'E-mail',
        'first_name' => 'Ім'я',
        'middle_name' => 'По батькові',
        'last_name' => 'Прізвище',
        'status' => 'Статус',
    ];
}

public static function findIdentity($id)
{
    return static::find()->where(['id' => $id])->andWhere(
        [
            'or',
            ['status' => self::STATUS_ACTIVE],
            ['status' => self::STATUS_BANNED]
        ]
    )->one();
}

public static function findIdentityByAccessToken($token, $type = null)
{
    throw new NotSupportedException('"findIdentityByAccessToken" не
впроваджено.');
```

```

}

public static function findByEmail($email)
{
    return static::find()->where(['email' => $email])->andWhere(
        [
            'or',
            ['status' => self::STATUS_ACTIVE],
            ['status' => self::STATUS_BANNED]
        ]
    )->one();
}

public function getId()
{
    return $this->getPrimaryKey();
}

```

```

public function getAuthKey()
{
    return $this->auth_key;
}

public function validateAuthKey($authKey)
{
    return $this->getAuthKey() === $authKey;
}

public function generatePasswordResetToken()
{
    $this->password_reset_token = Yii::$app->security-
>generateRandomString() . '_' . time();
}

public function validatePassword($password)
{
    return Yii::$app->security->validatePassword($password, $this-
>password_hash);
}

public function setPassword($password)
{
    $this->password_hash = Yii::$app->security-
>generatePasswordHash($password);
}

public function generateAuthKey()
{
    $this->auth_key = Yii::$app->security->generateRandomString();
}

public function checkEmail()
{
    $query = User::find()->where(['email' => $this->email])->andWhere(
        [
            'or',
            ['status' => 10],
            ['status' => 99]
        ]
    )->andWhere(['<>', 'id', $this->id]);
}

```

```

        if ($query->count() == 0) {
            return true;
        }
        return false;
    }

    public function getAuthAssignments()
    {
        return $this->hasMany(AuthAssignment::class, ['user_id' => 'id']);
    }

    public function getClassStudents()
    {
        return $this->hasMany(ClassStudents::class, ['student_id' => 'id']);
    }

    public function getCourses()
    {
        return $this->hasMany(Courses::class, ['teacher_id' => 'id']);
    }

    public function getForums()
    {
        return $this->hasMany(Forum::class, ['user_id' => 'id']);
    }

    public function getHomeworks()
    {
        return $this->hasMany(Homework::class, ['student_id' => 'id']);
    }

    public function getItemNames()
    {
        return $this->hasMany(AuthItem::class, ['name' => 'item_name'])-
>viaTable('auth_assignment', ['user_id' => 'id']);
    }

    public function getMarks()
    {
        return $this->hasMany(Marks::class, ['student_id' => 'id']);
    }

```

```

public function getTeacherInfo()
{
    return $this->hasOne(TeacherInfo::class, ['teacher_id' => 'id']);
}
}

```

Контролер, файл «models/User.php»

```

<?php
namespace app\controllers;
use app\models\AuthAssignment;
use Yii;
use yii\filters\AccessControl;
use yii\web\Controller;
use yii\web\Response;
use yii\filters\VerbFilter;
use app\models\LoginForm;
use app\models\ContactForm;
use app\models\Content;
use app\models\ForgotPasswordForm;
use app\models\SignUpForm;
use app\models\User;
use yii\web\NotFoundHttpException;

class SiteController extends Controller
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::class,
                'only' => ['logout'],
                'rules' => [
                    [
                        'actions' => ['logout'],
                        'allow' => true,
                        'roles' => ['@'],
                    ],
                ],
            ],
            'verbs' => [
                'class' => VerbFilter::class,

```

```

        'actions' => [
            'logout' => ['get'],
        ],
    ],
];
}

public function actions()
{
    return [
        'error' => [
            'class' => 'yii\web\ErrorAction',
        ],
        'captcha' => [
            'class' => 'yii\captcha\CaptchaAction',
            'fixedVerifyCode' => YII_ENV_TEST ? 'testme' : null,
        ],
    ];
}

public function actionIndex()
{
    return $this->render('index', [
        'content' => $this->findContent(),
    ]);
}

public function actionLogin()
{
    if (!$yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $model = new LoginForm();
    if ($model->load($yii::$app->request->post()) && $model->login()) {
        return $this->goBack();
    }

    $model->password = '';
    return $this->render('login', [
        'model' => $model,
    ]);
}

```

```

    });
}

public function actionLogout()
{
    Yii::$app->user->logout();

    return $this->goHome();
}

public function actionSignUp()
{
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $protocol = isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] === 'on' ?
'https://' : 'http://';
    $link = $protocol . $_SERVER['HTTP_HOST'];
    $model = new SignUpForm();

    if ($model->load(Yii::$app->request->post())) {
        if ($model->checkEmail()) {
            if ($model->checkPassword()) {
                if ($model->term) {
                    if (($user = User::findOne(['email' => $model->email,
'status' => 0])) === null) {
                        $user = new User();
                    }

                    $user->email = $model->email;
                    $user->first_name = $model->first_name;
                    $user->middle_name = $model->middle_name == '' ? null
: $model->middle_name;
                    $user->last_name = $model->last_name;
                    $user->password_hash = Yii::$app->security-
>generatePasswordHash($model->password);
                    $user->generateAuthKey();
                    $user->generatePasswordResetToken();

```

```

        if ($user->save()) {
            $user_id = User::findOne(['email' => $user->email])->id;

            if (($add_role =
AuthAssignment::findOne(['user_id' => $user_id])) === null) {
                $add_role = new AuthAssignment();
            }
            $add_role->item_name = $model->role;
            $add_role->user_id = $user_id;

            if ($add_role->save()) {
                $subject = 'Реєстрація на сайті "' .
Yii::$app->name . '"';
                $body = '<p><strong>Реєстрація на сайті "' .
Yii::$app->name . '"</strong></p>
                <p>Підтвердіть реєстрацію перейшовши
за посиланням нижче.</p>
                <hr>
                <p><u>Посилання</u>: <a href="' .
$link . '/site/confirmation/' . $user->password_reset_token . '">' . $link .
'/site/confirmation/' . $user->password_reset_token . '</a></p>';

                if ($model->contact($subject, $body)) {
                    Yii::$app->session->setFlash('success',
'Заявку надіслано на ваш e-mai, підтвердіть її.');
```

```

        $errorMsg = 'Паролі відрізняються.';
        $model->addError('password', $errorMsg);
    }
} else {
    $errorMsg = 'E-mail вже занято.';
    $model->addError('email', $errorMsg);
}
}

return $this->render('/site/sign-up', [
    'model' => $model,
]);
}

public function actionConfirmation($id)
{
    if (($model = User::findOne(['password_reset_token' => $id])) !==
null) {
        $model->password_reset_token = null;
        $model->status = 10;

        if ($model->save()) {
            Yii::$app->session->setFlash('success', 'Пошту підтверджено,
тепер ви можете авторизуватись.');
```

```

            return $this->redirect(['login']);
        }
    }

    throw new NotFoundHttpException('Запитувана сторінка не існує.');
```

```

}

public function actionForgotPassword()
{
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $model = new ForgotPasswordForm();
    $protocol = isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] === 'on' ?
'https://' : 'http://';
    $link = $protocol . $_SERVER['HTTP_HOST'];

```



```

        if (($id = Yii::$app->request->get('id')) !== null) {
            if (($user = User::findOne(['password_reset_token' => $id])) !==
null) {
                $chars =
'qazxswedcvftrtgbnhyujmkiolp1234567890QAZXSWEDCVFRTGBNHYUJMKIOLP';
                $max = 8;
                $size = strlen($chars) - 1;
                $password = null;

                while ($max--)
                    $password .= $chars[rand(0, $size)];

                $user->password_reset_token = null;
                $user->setPassword($password);

                $email = $user->email;

                if ($user->save()) {
                    $subject = 'Відновлення паролю в додатку "' . Yii::$app-
>name . '"';
                    $body = '<p><strong>Відновлення паролю в додатку "' .
Yii::$app->name . '"</strong></p>
                    <p>Ваш новий пароль для входу в систему.
<b>Нікому не передавайте його.</b></p>
                    <hr>
                    <p><u>Пароль</u>: ' . $password . '</p>';

                    Yii::$app->mailer->compose()
                        ->setTo($email)
                        ->setFrom([Yii::$app->params['senderEmail'] =>
Yii::$app->name . ' mailer'])
                        ->setSubject($subject)
                        ->setHtmlBody($body)
                        ->send();

                    Yii::$app->session->setFlash('success', 'Новий пароль
надіслано на ваш e-mail.');
```

```

                    return $this->redirect('/site/login');
                }
            }
        }
    }
}

```

```

    }

    throw new NotFoundHttpException('Запитувана сторінка не існує.');
```

```

}
```

```

if ($model->load(Yii::$app->request->post())) {
    if (!$model->checkEmail()) {
        $user = User::findOne(['email' => $model->email]);
        $user->generatePasswordResetToken();

        if ($user->save()) {
            $subject = 'Відновлення паролю в додатку "' . Yii::$app->name . '"';
            $body = '<p><strong>Відновлення паролю в додатку "' . Yii::$app->name . '"</strong></p>
                <p>Підтвердіть відновлення перейшовши за посиланням нижче.</p>
                <hr>
                <p><u>Посилання</u>: <a href="' . $link . '/site/forgot-password/' . $user->password_reset_token . '">' . $link . '/site/forgot-password/' . $user->password_reset_token . '</a></p>';

            if ($model->contact($subject, $body)) {
                Yii::$app->session->setFlash('success', 'Заявку надіслано на ваш e-mail, підтвердіть її.');
```

```

                return $this->redirect('login');
```

```

            }
```

```

        }
```

```

    } else {
```

```

        $errorMsg = 'Користувача з таким e-mail не знайдено.';
```

```

        $model->addError('email', $errorMsg);
```

```

    }
```

```

}
```

```

return $this->render('forgot-password', [
    'model' => $model,
]);
```

```

}
```

```

public function actionContact()
{
```

```

$model = new ContactForm();

if ($model->load(Yii::$app->request->post())) {
    $subject = 'Зворотній зв'язок додатку "' . Yii::$app->name . '"';
    $body = '<p><strong>Зворотній зв'язок додатку "' . Yii::$app->name . '"</strong></p>
        <hr>
        <p><b>ПІБ</b>: ' . $model->name . '</p>
        <p><b>E-mail</b>: ' . $model->email . '</p>
        <hr>
        <p>' . $model->body . '</p>';

    if ($model->contact($subject, $body)) {
        Yii::$app->session->setFlash('success', 'Дякуємо за ваше звернення. Ми відповімо вам якомога швидше.');
```

```

        return $this->refresh();
    } else {
        Yii::$app->session->setFlash('error', 'Виникла помилка, спробуйте надіслати форму ще раз.');
```

```

    }
}

return $this->render('contact', [
    'content' => $this->findContent(),
    'model' => $model,
]);
}

public function actionPolicy()
{
    return $this->render('policy', [
        'content' => $this->findContent(),
    ]);
}

protected function findContent()
{
    $url = '/' . Yii::$app->controller->id . '/' . Yii::$app->controller->action->id;

    if (($model = Content::findOne(['url' => $url])) !== null) {

```

```

        return $model;
    }

    throw new NotFoundHttpException('Запитувана сторінка не існує.');
```

Файл конфігурації “config/web.php”

```

<?php

$params = require __DIR__ . '/params.php';
$db = require __DIR__ . '/db.php';

$config = [
    'id' => 'basic',
    'basePath' => dirname(__DIR__),
    'bootstrap' => ['log'],
    'aliases' => [
        '@bower' => '@vendor/bower-asset',
        '@npm' => '@vendor/npm-asset',
    ],
    'components' => [
        'authManager' => [
            'class' => 'yii\rbac\DbManager',
        ],
        'request' => [
            'cookieValidationKey' => 'RK2C3CHSQPMfp06urlPM7raO',
            'baseUrl' => '',
        ],
        'cache' => [
            'class' => 'yii\caching\FileCache',
        ],
        'user' => [
            'identityClass' => 'app\models\User',
            'enableAutoLogin' => true,
        ],
        'errorHandler' => [
            'errorAction' => 'site/error',
        ],
        'mailer' => [
            'class' => \yii\symfonymailer\Mailer::class,
```

```

        'viewPath' => '@app/mail',
        // send all mails to a file by default.
        'useFileTransport' => true,
    ],
    'log' => [
        'traceLevel' => YII_DEBUG ? 3 : 0,
        'targets' => [
            [
                'class' => 'yii\log\FileTarget',
                'levels' => ['error', 'warning'],
            ],
        ],
    ],
    'db' => $db,
    'urlManager' => [
        'enablePrettyUrl' => true,
        'showScriptName' => false,
        'rules' => [
            '/' => 'site/index',
            '<controller:\w+>/' => '<controller>/index',
            '<controller:\w+>/<action:(\w|-)+>/<id:(\w|-)+>' =>
            '<controller>/<action>',
            '<controller:\w+>/<action:(\w|-)+>/<id:\d+>' =>
            '<controller>/<action>',
            '<module:\w+>/<controller:\w+>/<action:(\w|-)+>' =>
            '<module>/<controller>/<action>',
            '<controller:\w+>/<action:(\w|-)+>' =>
            '<controller>/<action>',
            '<controller:\w+>/<action:\w+>/' => '<controller>/<action>',
            'sitemap.xml' => 'site/sitemap',
        ],
    ],
    'seo' => [
        'class' => 'app\components\SeoComponent',
    ],
],
'params' => $params,
];

return $config;

```

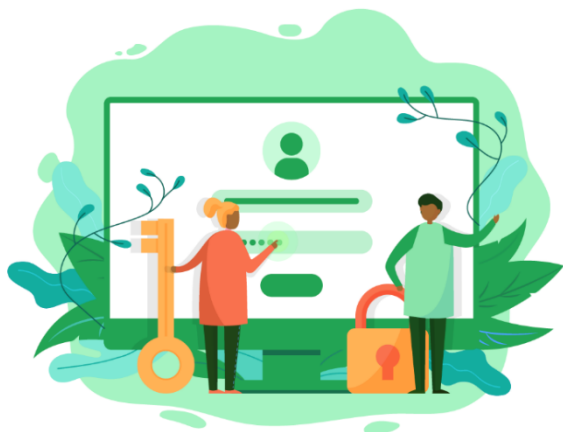
ДОДАТОК Б

ІНСТРУКЦІЯ КОРИСТУВАЧА

1. Гість

Незарєєстрований користувач може перейти до веб-додатку за посиланням та ознайомитись з основною інформацією про сервіс, контактною інформацією та політикою.

Гість може створити власний акаунт, для того, щоб авторизуватись на сайті. Для цього потрібно відкрити сторінку «Реєстрація» (рисунок Б.1).



Роль *

Не вказано ▾

Ім'я *

По батькові

Прізвище *

E-mail *

Пароль *

Повторіть пароль *

Користувацьку угоду прочитано

[➤ Зареєструватись](#)

Рисунок Б.1 – Форма реєстрації

Червоною зірочкою позначено поля обов'язкові для заповнення. Після відправки форми потрібно підтвердити реєстрацію за посиланням, яке прийде на пошту. Далі можна авторизуватись за вказаними e-mail та паролем (рисунок Б.2). Натиснувши «Запам'ятати мене» - сесія буде збережена на 14 днів, не бажано натискати, користуючись чужим пристроєм.



E-mail *

Пароль *

Запам'ятати мене

Якщо ви забули пароль - ви можете відновити його, перейшовши за посиланням.

Рисунок Б.2 – Форма авторизації

У випадку, якщо користувач забув пароль – він може відновити його, надіславши запит на відновлення. На пошту прийде повідомлення з підтвердженням.

2. Авторизований користувач

Будь-який авторизований користувач має доступ до вкладки «Мій профіль», де може переглянути особисту інформацію та редагувати її. У викладача стає доступною форма для заповнення детальної інформації, аби додати туди свої навички, досвід та освіту, щоб здобувачі освіти могли ознайомитись, хто саме буде вести навчальний курс. Приклад особистого профілю користувача показано на рисунку Б.3. Інтерфейс примітивний і

доступний, кнопки з діями супроводжуються іконками для кращого розуміння функціоналу.

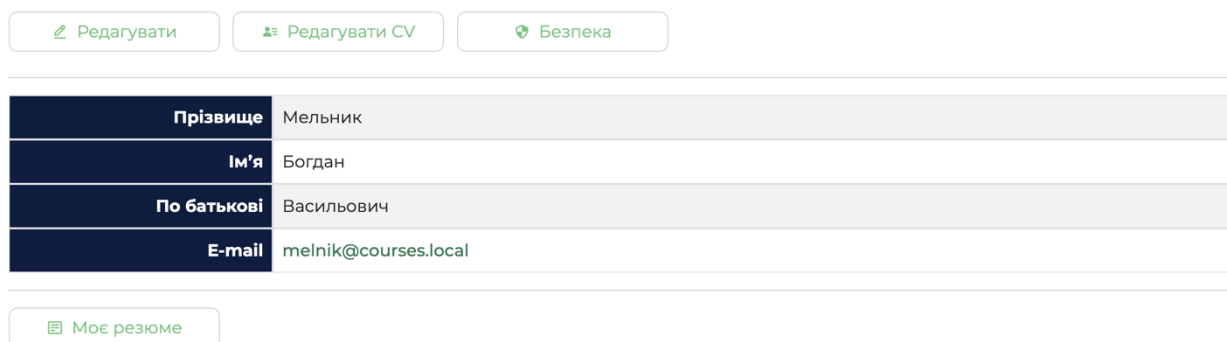


Рисунок Б.3. – Сторінка «Мій профіль»

2.1. Адміністратор

Меню адміністратора, окрім загальнодоступних посилань, має ще вкладки:

- Викладачі;
- Здобувачі освіти;
- Чорний список;
- Контент.

Сторінки «Викладачі», «Здобувачі освіти» мають таблицю з відповідними користувачами, приклад таблиці на рисунку Б.4. Над таблицею кнопка «Фільтри» для розширеного пошуку значень за параметрами. В кожній таблиці праворуч поруч з кожним елементом розміщено кнопки дій.

Викладачі

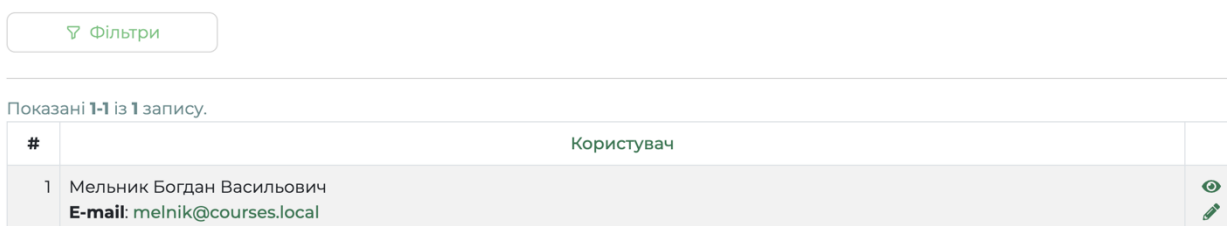


Рисунок Б.4 – Таблиця з користувачами

Користувача можна заблокувати за порушення правил сервісу. Потрібно натиснути на кнопку з іконкою ока «Переглянути» і натиснути «Заблокувати». Він потрапить на сторінку «Чорний список». Натиснувши на іконку ручки «Редагувати» - можна редагувати основну інформацію про користувача.

Сторінка «Контент» містить перелік сторінок, які адміністратор може редагувати, а саме контент цих сторінок та метатеги (рисунок Б.5).

Контент

Показані 1-3 із 3 записів.







#	Сторінка	Відредаговано	
1	<p>Сторінка: Інформація про сервіс відео-курси онлайн URL: http://courses.local/site/index Ключові слова: StudyAdd, відео-курси онлайн, інформація, Yii2 Опис: Офіційний веб-додаток для публікації, керування онлайн відео-курсами створений для учнів, студентів, які користуються перевагами онлайн-освіти: обирайте теми, які вас цікавлять, отримуйте мегакорисну інформацію від провідних учителів та працівників галузі освіти з усієї України.</p>	15.03.2023 08:17:57	 
2	<p>Сторінка: Контакти URL: http://courses.local/site/contact Ключові слова: StudyAdd, веб-додаток, контакти, форма зворотнього зв'язку, Yii2 Опис: Контактна інформація для зв'язку з адміністрацією веб-додатку. Будь ласка, не соромтеся зв'язатися з нами, якщо у вас виникли якісь запитання.</p>	03.03.2023 17:46:11	 
3	<p>Сторінка: Політика конфіденційності URL: http://courses.local/site/policy Ключові слова: StudyAdd, веб-додаток, політика конфіденційності, захист даних, Yii2 Опис: Ця політика конфіденційності пояснює, як ми збираємо, використовуємо та захищаємо вашу особисту інформацію, коли ви відвідуєте наш веб-додаток. Ми зобов'язані забезпечувати конфіденційність вашої інформації та захищати вашу приватність.</p>	03.03.2023 13:24:04	 

Рисунок Б.5 – Сторінка «Контент»

2.2. Викладач

Меню викладача, окрім загальнодоступних посилань, має ще вкладку:

– Мої курси.

Для початку роботи викладачеві потрібно додати свій курс і наповнити його інформацією в наступному порядку:

1. Створити новий навчальний курс;
2. Створити навчальні групи (сума здобувачів освіти у всіх групах не більше загальної кількості);
3. Додати навчальні уроки;

4. Наповнити уроки інформацією, а саме текстовим і медіа-матеріалом. Для організації прямого ефіру, викладач має створити ефір на платформі YouTube та додати посилання через «Відео»;
5. Для навчальних груп окремо для кожного уроку додати час початку уроку та завершення;
6. Після наповнення курсу інформацією змінити його статус на «Йде набір», тепер здобувачі освіти зможуть подавати заявки на навчальні курси;
7. Набрати потрібну кількість здобувачів освіти на курс та змінити статус на «В процесі» - здобувачам освіти будуть доступні медіаматеріали уроку.
8. Викладач має змогу писати в форумі відповідного класу, а також переглядати домашні роботи, які йому надіслали та відповідати на них.

2.3. Здобувачі освіти

Меню адміністратора, окрім загальнодоступних посилань, має ще вкладки:

- Всі курси;
- Обрані курси;
- Мої курси;
- Мої запити.

Сторінка «Всі курси» містить інформацію про всі доступні курси, якщо курс новий – його можна додати в обране, коли статус курсу «Йде набір» - можна подати заявку на проходження курсу.

Сторінка «Обрані курси» - перелік обраних курсів.

Сторінка «Мої запити» - перелік курсів, на які подано заявки. Відповідь про одобрення, або відхилення приходить здобувачу освіти на пошту.

Сторінка «Мої курси» - перелік курсів, на які було прийнято заявку для проходження курсу.

ДОДАТОК С
КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ХЕРСОНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Лесів Олександр Іванович,
учасник 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;

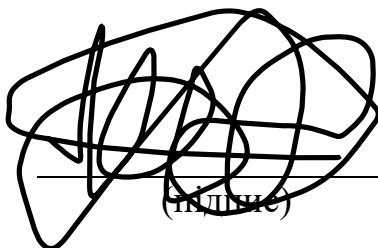
- самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;
- надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;
- не використовувати результати досліджень 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.

12.09.2019

(дата)



(підпис)

Олександр Лесів

(ім'я, прізвище)