

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

РОЗРОБКА ВЕБ-СЕРВІСУ ОНЛАЙН ВИСТАВОК ЦИФРОВОГО
МИСТЕЦТВА З МОЖЛИВОСТЯМИ СОЦІАЛЬНОЇ МЕРЕЖІ

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу 441 групи

Спеціальності: 121 Інженерія

програмного забезпечення

Освітньо-професійної програми:

Інженерія програмного забезпечення

першого (бакалаврського) рівня освіти

Пяцько С.С

Керівник: доктор педагогічних наук,

професор кафедри комп'ютерних наук та

програмної інженерії ХДУ

Круглик В.С

Рецензент:

Морозов В'ячеслав Євгенійович

Tech Lead Forex Tester Software

Херсон – 2023

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. АНАЛІЗ РИНКУ ВИСТАВКОВИХ ОНЛАЙН СИСТЕМ ТА ВИЯВЛЕННЯ ПОТРЕБ НА РЕАЛІЗАЦІЮ ПРОЄКТУ.	5
1.1 Аналіз ринку та особливостей виставкових онлайн систем.	5
1.2 Формування вимог до нового проєкту.	6
ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ	10
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБ-СЕРВІСУ ВИСТАВКОВИХ ОНЛАЙН СИСТЕМ.....	11
2.1 Архітектура веб-сервісу.	11
2.2 Технічне завдання для веб-сервісу.	12
2.3 Структура системи.	14
ВИСНОВКИ ДО ДРУГОГО РОЗДІЛУ	18
РОЗДІЛ 3. РОЗРОБКА ПРОТОТИПУ ВИСТАВКОВОЇ ОНЛАЙН СИСТЕМИ.	19
3.1 Вибір технологій для розробки програми.	19
3.2 Імплементация серверної частини	22
3.3 Розробка інтерфейсу користувача.	24
3.4 Підготовка до тестування та деплою.	27
3.5 Плани подальшого розвитку проєкту.	28
ВИСНОВКИ ДО ТРЕТЬОГО РОЗДІЛУ	30
ВИСНОВКИ	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	32
ДОДАТКИ.....	35
ДОДАТОК А. КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ	35
ДОДАТОК Б. ДІАГРАМИ ПОСЛІДОВНОСТЕЙ.	36

ВСТУП

Актуальність дослідження: Дослідження проводиться у зв'язку зі попитом інтеграції світу мистецтва у цифровий простір. Зростання популярності виставок цифрового мистецтва потребує створення централізованої платформи, яка буде надавати художникам та колекціонерам зручний сервіс для демонстрації, перегляду та купівлі цифрових картин. Крім того, інтеграція інструментів та можливостей соціальної мережі дозволяє зручну взаємодію та співпрацю між членами мистецької спільноти та користувачами платформи.

Об'єкт: Веб-сервіс виставка.

Предмет: Веб-сервіс виставки з можливостями соціальної мережі.

Мета роботи: Метою завдання є проектування та створення веб-сервісу, який буде дозволяти користувачам брати участь у виставках цифрового мистецтва та спілкуватися з іншими людьми за допомогою вбудованого функціонала соціальної мережі.

Задачі:

1. Провести аналіз ринку, щоб визначити цільову аудиторію та виділити особливості програм аналогів.
2. Виокремити основні особливості та переваги які будуть вирізняти продукт з поміж конкурентів.
3. Визначити технічні вимоги до програми, включаючи вебдизайн, архітектуру бази даних та архітектуру REST API.
4. Спроекувати структуру майбутнього продукту.
5. Визначити технології які будуть використовуватися під час розробки програми.
6. Розробити прототип продукту згідно визначеними вимогам та завданням.
7. Описати стратегії розвитку продукту у майбутньому.

У першому розділі буде описаний аналіз ринку систем типу онлайн виставок за для визначення особливостей програм конкурентів та потреби цільової аудиторії. За результатами аналізу будуть сформовані фінальні вимоги до нового проєкту.

У другому розділі буде зроблений глибокий аналіз архітектури веб-сервісів та на його основі сформоване технічне завдання для майбутнього прототипу продукту. Також у цьому розділі буде спроектована структура розроблюваної системи.

У третьому розділі детально описані технології, які будуть використовуватися для створення веб-сервісу, обґрунтовано їх вибір та описані їх особливості. Також у цьому розділі детально описана імплементація серверної частини а також розробка інтерфейсу користувача. Буде описана підготовка до тестування та деплою.

РОЗДІЛ 1. АНАЛІЗ РИНКУ ВИСТАВКОВИХ ОНЛАЙН СИСТЕМ ТА ВИЯВЛЕННЯ ПОТРЕБ НА РЕАЛІЗАЦІЮ ПРОЄКТУ.

1.1 Аналіз ринку та особливостей виставкових онлайн систем.

Ринок веб-сервісів галерей має велику базу потужних платформ, які відповідають потребам багатьох користувачів та надають ряд можливостей для створення, демонстрації та монетизації цифрових витворів. На цей час існує декілька великих веб-сервісів як мають свій набір інструментів для взаємодії з художніми ресурсами. Для того аби зрозуміти основні стратегії які використовують сучасні популярні сайти які пов'язані з художньою тематикою, було проведено аналіз найпопулярніших з них, таких як: Pinterest, DeviantArt, ArtStation.

Pinterest - є популярною соціальною мережею, яка дозволяє користувачам зберігати та організовувати зображення, що їх цікавлять. Тут можна знайти велику кількість цифрових зображень різних жанрів та напрямків. У якості цікавих особливостей цієї платформи було виділено: легкий у використанні інтуїтивний інтерфейс який відмінно працює як на десктоп, так і на мобільних пристроях та можливість створювати публічні та приватні дошки зображень, що дозволяє користувачам легко організовувати картинки по тематичним колекціям. На відміну від мого проєкту Pinterest не спеціалізований виключно для мистецтва, тут можна знайти різноманітний контент який зазвичай слугує для отримання ідей та «рефернесів» які не завжди можуть відповідати потребам художників [1].

DeviantArt - є однією з найбільш популярних онлайн-спільнот для мистецтва, яка об'єднує багатьох художників та інших професіоналів мистецтва зі всього світу. На відміну від Pinterest цей сервіс є спеціалізованою платформою для мистецьких творів, на якій можна

знайти велику кількість цифрових витворів та широку професійну спільноту художників з усього світу. Користувачі цієї платформи мають можливість створювати портфоліо для подальшого пошуку роботи. Серед потенційних недоліків DeviantArt має надмірну кількість контенту різної якості який може ускладнити пошук конкретної тематики та зорієнтуватися серед спільноти. Також місцева система монетизації дає невеликий розмір безплатного простору для зберігання та виставлення творів [2].

ArtStation - є сервісом, що спеціалізується на цифровому мистецтві та використовується професіоналами мистецтва та візуальних ефектів для реклами, фільмів та ігор. Цей сервіс надає високу якість та професіоналізм мистецьких творів. Тут також присутня можливість створювати портфоліо як і в DeviantArt, але на відміну, ArtStation охоплює більший спектр професійних компаній що при цьому породжує вищі вимоги до якості та оформлення творів що публікуються на сервісі. Серед недоліків цієї платформи можна виділити повну недозволеність щодо демонстрації нехудожніх витворів таких як відео або фотографії, а також деякі важливі функції, такі як можливість відгукуватися на роботи та брати участь в конкурсах, доступні лише платним користувачам [3].

1.2 Формування вимог до нового проєкту.

Ідея проєкту полягає у розробці веб-сервісу галереї з функціоналом соціальної мережі. Для реалізації проєкту, було проведено аналіз кількох популярних веб-сервісів та на його основі були виділені які саме особливості дозволять створити конкурентоспроможний продукт.

Основною метою розробки свого сервісу було зроблено створення повноцінної системи виставок яка буде дозволяти користувачам з усього світу створювати та відвідувати тематичні події на яких будуть представлені художні витвори по певним темам. Були виділені такі сильні сторони прототипу продукту:

- Комфортний функціонал соціальної мережі, що дозволяє користувачам додавати контент, ділитися ним з іншими та взаємодіяти між собою.
- Повноцінна система онлайн виставок, що дозволяє користувачам створювати тематичні виставки зі своїми творами та відвідувати інші виставки.
- Можливість використання новітніх технологій у майбутньому, таких як NFT [4] і кооперація з відомими художніми спілками.

Недоліки, проєкту сформовані на основі аналізу ринку конкурентів виставкових онлайн систем та можливостей які на цю мить існують для розробки проєкту. Серед основних слабких сторін майбутнього продукту були виділені такі як:

- Недостатня кількість розробників може спричинити затримки у розробці та вплинути на якість продукту.
- Повільне інтегрування нових технологій може затримати розвиток проєкту та позиції на ринку.
- Висока конкуренція на ринку веб-сервісів галерей може стати перешкодою для залучення користувачів та розвитку продукту.

Основними діями для мінімізування слабких сторін проєкту є залучення більшої кількості фінансів, а у наслідку цього, розробників для розробки веб-сервісу. Для того аби майбутній продукт міг бути конкурентоспроможним, він має повною мірою реалізовувати його основний функціонал, який і буде привертати увагу потенційної цільової аудиторії.

Після аналізу ринку виставкових онлайн систем, були сформовані загрози для майбутнього продукту, які можуть слугувати перешкодами для розробки та розвитку веб-сервісу:

- Зміна вимог та технічних можливостей може призвести до виникнення конкурентів, які запропонують нові технології та функціонал, що можуть бути більш привабливими для користувачів.
- Недостатня кількість розробників та велике навантаження на технічну підтримку можуть призвести до затримок у розробці та обслуговуванні проекту.
- Інтернет-цензура та обмеження свободи слова можуть стати перешкодою для розширення проекту на міжнародному рівні.

Усі ці загрози будуть враховані при подальшій розробці проекту, а також будуть виконані дії за для їх уникання та мінімізування наслідків від їх впливу у майбутньому.

Після формування вимог були розглянуті можливості проекту після випуску його на відкритий ринок. Вони є результатом аналізу спроектованого функціоналу та цільової аудиторії:

- Зростаюча популярність онлайн-галерей із зображеннями може привести до збільшення кількості користувачів та збільшення прибутків.
- Застосування нових технологій, таких як NFT, може стати конкурентною перевагою та привести до збільшення кількості користувачів та інтересу до продукту.
- Колаборація з відомими художніми спілками може привести до залучення ширшої аудиторії та збільшення інтересу до продукту.

За для візуалізації усіх факторів свого продукту був створений SWOT аналіз – «аналіз дозволяє провести детальне дослідження зовнішнього й внутрішнього середовища. Результатом раціонального SWOT-аналізу, спрямованого на формування взагальненого інформаційного потенціалу, повинні з'явитися *ефективні рішення*, що стосуються відповідної реакції (впливу) суб'єкта (слабкої, середньої й

сильної) відповідно до сигналу (слабкому, середньому або сильному) зовнішнього середовища»[5] у якому зазначені сильні сторони, слабкі сторони, можливості та загрози (Діаграма 2.1).

SWOT - аналіз

Strengths (Сильні сторони)	Weaknesses (Слабкі сторони)
<ul style="list-style-type: none"> - Комфортний функціонал соціальної мережі - Повноцінна система онлайн виставок - Можливість використання новітніх технологій у майбутньому, таких як NFT і колаборація з відомими художніми спілками 	<ul style="list-style-type: none"> - Недостатня кількість розробників - Повільне інтегрування нових технологій - Висока конкуренція на ринку веб-сервісів галерей
Opportunities (Можливості)	Threats (Загрози)
<ul style="list-style-type: none"> - Зростаюча популярність онлайн-галерей - Застосування нових технологій - Колаборація з відомими художніми спілками 	<ul style="list-style-type: none"> - Зміна вимог та технічних можливостей - Недостатня кількість розробників та велике навантаження

Діаграма 1.1 SWOT аналіз мого прототипу продукту

ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ

У першому розділі був проведений глибокий аналіз ринку та особливостей виставкових онлайн систем таких як: Pinterest, DeviantArt, ArtStation. Дослідження дозволило визначити їх переваги, недоліки та який ряд потреб основної цільової аудиторії вони задовольняють.

За результатами аналізу ринку були сформовані вимоги до нового розроблювального проєкту, була описана основна мета та сильні сторони розроблювального проєкт. Був оцінений ряд можливостей для проєкту у майбутньому, як результат аналізу основних особливостей. Були враховані слабкі сторони, а також потенційні загрози при майбутній реалізації продукту на ринку, наслідки від яких повинні бути мінімізовані.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБ-СЕРВІСУ ВИСТАВКОВИХ ОНЛАЙН СИСТЕМ.

2.1 Архітектура веб-сервісу.

Поняття веб-сервісу є ключовим у контексті розробки сучасного програмного забезпечення та передачі даних через мережу інтернет. Веб-сервіс визначають як «програмна система, що ідентифікується URI, і публічні інтерфейси та прив'язки якої визначені та описані мовою XML. Опис цієї програмної системи може бути знайдено іншими програмними системами, які можуть взаємодіяти з нею відповідно до цього опису з використанням повідомлень, що базуються на XML та передаються за допомогою інтернет-протоколів. Термін «вебслужба» введено організацією W3C і застосовується до багатьох різних систем, але в основному термін стосується клієнтів та серверів, що взаємодіють за допомогою повідомлень протоколу SOAP. В обох випадках припускається, що існує також опис доступних операцій у форматі WSDL. Хоча наявність цього опису не є вимогою SOAP, а радше передумовою для автоматичного генерування коду на платформах Java та .NET на стороні клієнта.»[6]. Веб-сервіс можна розглядати як набір функцій, доступних в мережі Інтернет, які можна використовувати для розв'язання конкретних задач.

Основними перевагами веб-сервісів є великий ряд можливостей та універсальність. Вони дозволяють, попри платформу або пристрій, надавати увесь спектр функцій які надають інші системи. Крім того, веб-сервіси дозволяють створювати системи з різних компонентів, мереж та платформ.

Як інструмент для вирішення завдань різної складності, веб-сервіси застосовуються в багатьох галузях включаючи електронну комерцію, фінансові послуги, транспортні системи та інші.

У сучасному проектуванні та розробці веб-сервісів використовуються популярні архітектурні стилі моноліт та мікро сервіси. Моноліт є традиційним підходом, де всі компоненти програмного забезпечення об'єднані в одному великому додатку, тоді як мікро сервіси використовують розбиття програмного забезпечення на невеликі, незалежні компоненти, що працюють як окремі сервіси та взаємодіють один з одним через API [7]. Після аналізу обох цих архітектур, була обрана архітектура мікро сервісів, адже для розробки веб-сервісу онлайн виставок вони мають ряд переваг: гнучкість – що дозволяє вдосконалювати сервіси та змінювати їх, масштабованість – мікро сервіси можна легко масштабувати горизонтально, розподіленість розробки – яка дозволяє декомпонувати програмне забезпечення на незалежні частини, які можуть розроблюватися декількома командами паралельно та легкість в інтегруванні нових технологій.

Існує ряд протоколів які використовуються для взаємодії між різними пристроями, найпоширеніші з них це: REST, SOAP, GraphQL, WebSocket та інші, серед яких була обрана REST – «підхід до архітектури мережевих протоколів, які надають доступ до інформаційних ресурсів. Був описаний і популяризований 2000 року Роем Філдінгом, одним із творців протоколу HTTP. В основі REST закладено принципи функціонування Всесвітньої павутини і, зокрема, можливості HTTP. Філдінг розробив REST паралельно з HTTP 1.1 базуючись на попередньому протоколі HTTP 1.0.» [11], та різні мови програмування як Java, C#, JavaScript, Ruby, PHP та інші. Ці мови можуть використовуватися сумісно один з одним при написанні комплексного додатка.

2.2 Технічне завдання для веб-сервісу.

Після аналізу ринку веб-сервісів галерей та окреслені особливостей мого проєкту, було розпочате написання чіткого технічного завдання нового прототипу продукту, де розписані пункти які повинні бути виконані при розробці програмної частини проєкту. Кожен пункт ТЗ був створений з урахуванням результатів проєктування усіх аспектів програми.

Технічне завдання:

- Розробити систему виставок. Основний функціонал виставок та тематичних подій буде надавати можливість для відвідування та створення виставок (увійшов у прототип).
- Розробити систему повідомлень. Основним завданням цієї системи, буде оповіщення користувачів про нові коментарі, які були залишені під створеними ними постами та про нові виставки які наближаються (увійшов у прототип).
- Спроектувати систему чату. Цей функціонал буде надавати користувачам можливість контактувати один з одним та ділитися своїми напрацюваннями у художній сфері (увійшов у прототип).
- Розробити навігацію на сторінку виставки, де будуть відображатися всі картини, що належать до цієї виставки, опис про виставку, період її проведення та ім'я автора (увійшов у прототип).
- Розробити сортувальну систему на сторінку всіх постів та на сторінку виставки (увійшов у прототип).
- Розробити календар, на якому буде відображатися розклад проведення усіх виставок, інформація про ці виставки та період часу у який вони проводяться (увійшов у прототип).
- Розробити шаблон платних підписок (не увійшов у прототип).

- Розробити пагінацію даних на сторінці постів, щоб забезпечити зручний перехід між сторінками (не увійшов у прототип).

2.3 Структура системи.

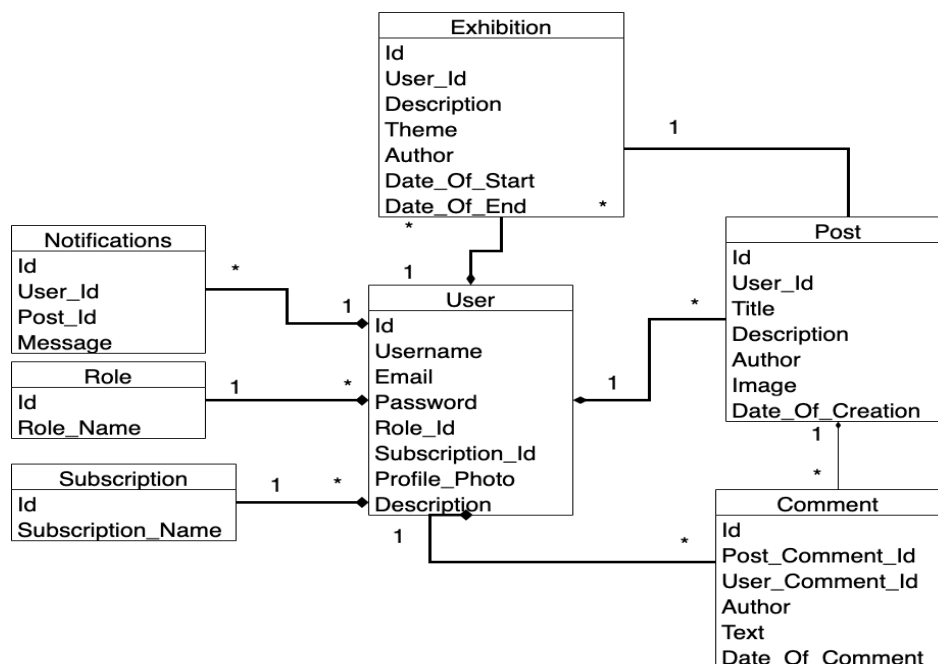
При розробці структури свого веб-сервісу, була спроектована база даних у PostgreSQL [12], та визначені відповідні таблиці які будуть утворювати систему сховища даних програми:

- Таблиця «User» міститиме інформацію про зареєстрованих користувачів веб-сервісу. Ця таблиця буде містити поля, такі як ім'я, прізвище, email, роль користувача, дата реєстрації, дата останнього входу в систему тощо.
- Таблиця «Post» буде зберігати інформацію про пости, які створюють користувачі на веб-сервісі. Ця таблиця буде містити поля, такі як id автора посту, назва посту, опис посту, дата створення, дата останнього оновлення тощо.
- Таблиця «Role» буде містити список доступних ролей користувачів, таких як адміністратор, модератор, автор тощо. Ця таблиця буде використовуватись для надання доступу до певних функцій на веб-сервісі в залежності від ролі користувача.
- Таблиця «Subscription» буде зберігати дані які будуть використовуватися для перевірки наявності платних підписок у користувача та надання йому відповідних привілеїв.
- Таблиця «Comment» буде використовуватись для зберігання коментарів користувачів під постами на веб-сервісі. Ця таблиця буде містити такі поля як id автора коментаря, id посту, текст коментаря, дата створення тощо.
- Таблиця «Exhibition» буде містити інформацію про виставки цифрового мистецтва на веб-сервісі. Ця таблиця буде містити

поля, такі як назва виставки, опис виставки, дата старту виставки, дата закінчення виставки тощо. Крім того, у цій таблиці будуть зберігатись інформація про автора виставки, теги виставки та посилання на картинки, що представляють виставку.

- Таблиця «Notifications» буде використовуватись для зберігання повідомлень про дії на сайті які будуть надходити до користувача. Ця таблиця буде містити поля як id повідомлення, id користувача якому відноситься це повідомлення, id посту та сам контент повідомлення.

Під час процесу проєктування структури програми була створена діаграма класів – «статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм [13].» БД програми (Діаграма 2.2).

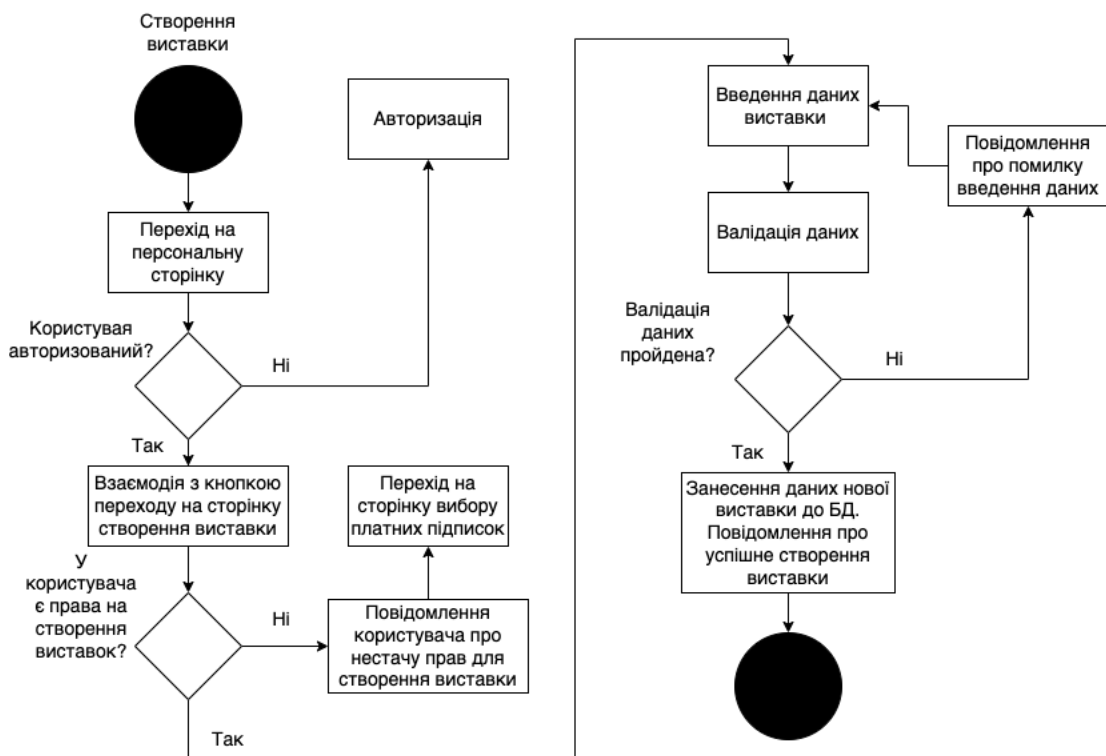


Діаграма 2.1 Діаграма класів бази даних

Після створення структури бек-енд частини своєї програми, було розпочате проектування основних процесів роботи додатка. Для того аби зрозуміти яким саме чином будувати фронт-енд частину сервісу, була розпочата розробка сценаріїв та шляхів якими будуть слідувати користувачі.

Для візуалізації послідовностей дій користувачів були створені діаграми послідовностей - «різновид діаграми в UML. Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність надісланих повідомлень [14]», в яких будуть відображатися процеси авторизації, реєстрації, створення постів, створення виставок та інші.

Далі у вигляді прикладу представлена діаграма послідовності яка відображає процес створення нової виставки на сайті.



Діаграма 2.2 Діаграма послідовностей створення виставки

На діаграмі 2.3 представлена послідовність дій користувача при створенні виставки, завдяки діаграмі такого типу можна враховувати усі етапи та перевірки які потребує конкретний процес для коректної роботи.

Функціонал створення нової виставки передбачає перевірку на наявність у користувача платного статусу, у випадку якщо користувач не володіє жодними привілеями які б надавали б йому право на створення виставок на веб-сервісі, його буде пере направлено на сторінку вибору плану підписки та запропоновано один із варіантів який містить у собі таку можливість, а також перевірку даних на правильність заповнення.

ВИСНОВКИ ДО ДРУГОГО РОЗДІЛУ

У другому розділі кваліфікаційної роботи були проаналізовані архітектури які використовуються для створення веб-сервісів. Серед них була обрана архітектура мікро сервісів та описане обґрунтування такого вибору.

З метою сформулювання чітких вимог до розробки програмного продукту було складено технічне завдання, яке містить перелік ключових функціональних частин проєкту. Ці частини мають допомогти реалізувати потенціал веб-сервісу повною мірою та задовольнити потреби користувачів.

Далі було перейдено до етапу проєктування структури програми. Це включало побудову плану бази даних, бек-енд та фронт-енд частини, а також діаграм послідовностей, необхідних для правильної роботи веб-сервісу.

РОЗДІЛ 3. РОЗРОБКА ПРОТОТИПУ ВИСТАВКОВОЇ ОНЛАЙН СИСТЕМИ.

3.1 Вибір технологій для розробки програми.

На сьогодні веб-сервіси створюються за допомогою великої кількості різних технологій та мов програмування, був проаналізований увесь спектр цих інструментів та обрані саме ті, які найбільше всього відповідали моїм потребам та задачам.

Під час вибору системи для зберігання даних, були проаналізовані ряд популярних баз даних, таких як: MySQL, Microsoft SQL, Oracle Database, PostgreSQL та MongoDB серед яких була обрана PostgreSQL - потужна та надійна база даних, що дозволила зберігати великі обсяги інформації та забезпечувала високу швидкість доступу до даних. Ця база даних була вибрана з декількох причин:

- **Надійність:** PostgreSQL є однією з найнадійніших БД оскільки має вбудовані механізми відновлення та захисту від втрати даних.
- **Масштабованість:** Вона може обробляти великий обсяг даних та може бути масштабована горизонтально та вертикально.
- **Продуктивність:** Сховище даних забезпечує високу продуктивність завдяки оптимізації запитів та мінімізації блокування.
- **Відкритість:** PostgreSQL є відкритим програмним забезпеченням, що означає, що його можна використовувати безкоштовно, розповсюджувати, модифікувати та використовувати в комерційних проектах.
- **Підтримка JSON:** База даних має вбудовану підтримку JSON, що дозволяє легко зберігати та обробляти документи за

допомогою цього формату даних без необхідності у використанні сторонніх бібліотек.

Для створення бек-енду виставкової системи була використана ASP Net. Core Web API[15] у середовищі розробки Rider, що дозволило ефективно створювати та підтримувати алгоритми серверної частини своєї програми. Цей ряд технологій був вибраний з цих причин:

- Масштабованість: ASP.Net Core Web API дозволяє розширювати та масштабувати веб-сервіси відповідно до потреб бізнесу.
- Підтримка HTTP протоколу: Web API дає можливість взаємодіяти з клієнтами за допомогою протоколу HTTP, що дозволяє виконувати RESTful запити.
- Швидкість: Ця технологія дозволяє створювати веб-сервіси, які працюють швидше і забезпечують швидкий відгук.
- Сумісність зі сторонніми бібліотеками: ASP.Net Core Web API підтримує стандарти веб-сервісів, такі як SOAP і WSDL, та дозволяє легко інтегрувати з іншими бібліотеками та сервісами.
- Наявність готових модулів: в ASP.Net Core Web API доступний великий вибір готових модулів для забезпечення безпеки, обробки запитів, валідації даних і т. д.
- Підтримка розробки на різних платформах: дана технологія підтримує розробку веб-додатків на різних платформах, таких як Windows, Linux, а також мобільних платформах.
- Відкритий код: ASP.Net Core Web API - це відкрите програмне забезпечення, що означає, що його можна використовувати безкоштовно та модифікувати на свій розсуд.

- Легко підтримувати: платформа .Net є досить простою у використанні та легко підтримується, що зменшує час і кошти на розробку та підтримку веб-додатків.

Для розробки фронтенду використовувалася бібліотека React Js [16]. У середовищі розробки Web Storm, ця бібліотека була вибрана з декількох причин:

- Компонентна архітектура, яка дозволяє розбивати основні блоки функціонала додатку на різні, незалежні один від одного, компоненти, які потім можна легко покривати тестами та пере використовувати.
- Використання Virtual DOM. React використовує віртуальне дерево структури сайту що дозволяє маніпулювати HTML-деревом без необхідності повної перерендеризації сторінки. Це дозволяє зменшити витрати ресурсів та збільшити продуктивність додатка.
- Можливість використання JSX. Бібліотека дозволяє використовувати JSX – спеціальний синтаксис який дозволяє поєднувати HTML та JavaScript код в одному файлі, що забезпечує більш зрозумілий та читабельний код
- Велика кількість сторонніх бібліотек та інструментів. React має велику екосистему бібліотек та інструментів, що дозволяє розширити можливості фреймворку та прискорити розробку.

Для того аби швидко та якісно створювати стилі на клієнтській частині додатка була використана MUI React [17]. Вибір цієї бібліотеки стилів був зроблений з наведених причин:

- Матеріальний дизайн: MUI React дозволяє легко і швидко створювати веб-інтерфейси з матеріальним дизайном, що забезпечує їм модний вигляд та сучасний стиль.

- Модульність: бібліотека складається з набору модулів, які можна використовувати окремо або разом для створення потрібного інтерфейсу. Це дозволяє забезпечити швидке та ефективне створення проекту.
- Кросбраузерність: вона працює на будь-яких сучасних веб-браузерах та забезпечує зручну розробку на різних платформах.
- Адаптивність: MUI React дозволяє створювати інтерфейси, які адаптуються до різних екранних розмірів, що забезпечує їх відповідність на мобільних пристроях.
- Компоненти: вона має широкий вибір компонентів, які можна використовувати для створення різних елементів інтерфейсу, таких як кнопки, текстові поля, меню, списки та інші.
- Підтримка: ця бібліотека стилів є популярним фреймворком з активною спільнотою розробників, що забезпечує підтримку та оновлення фреймворку відповідно до змін у технологіях.

3.2 Імплементация серверної частини

Розробка веб-сервісу виставкових онлайн систем була розпочата зі створення серверної частини – «це сервер, що приймає HTTP-запити від клієнтів, зазвичай веббраузерів, видає їм HTTP-відповіді, зазвичай разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними. Вебсервер — одна із основ Всесвітньої павутини.

Вебсервером називають як програмне забезпечення, що виконує функції вебсервера, так і комп'ютер, на якому це програмне забезпечення працює. Клієнти дістаються вебсервера за URL-адресою потрібної їм вебсторінки або іншого ресурсу.[18]». При розробці бек-енд частин веб-сервісів використовуються різні шаблони проектування програмного

забезпечення, які дозволяють ефективніше розробляти, тестувати та підтримувати програми. Існує ряд найбільш популярних шаблонів: Model-View-Controller (MVC), Model-View-ViewModel (MVVM), Repository Pattern, Dependency Injection (DI). Після глибокого аналізу цих шаблонів, та порівняння їх переваг та недоліків, був обраний шаблон MVC, оскільки він дозволяє розділяти логіку додатка на логічно відокремлені компоненти: Model – компонент що містить логіку, пов'язану з обробкою даних, View – надає інформацію користувачеві та містить шаблони відображення даних, Controller – обробляє запити користувача та взаємодіє зі збереженими даними. Такий підхід полегшує розробку та підтримку коду, та надає можливість розробникам працювати над різними частинами додатка паралельно, що забезпечує гнучкість та зменшує залежність між компонентами.

За шаблоном MVC у серверній частині були зроблені моделі до кожної таблиці бази даних, які містять їх властивості та конфігурації.

Після цього, були створені директиви які містять методи для взаємодії з базою даних, такими як запит на отримання всіх значень відповідної таблиці або запит на додання нового елемента у відповідну таблицю.

Коли усі методи та моделі були заповнені, була проведена міграція бази даних – «це керування інкрементними, зворотніми змінами схемиреляційної бази даних. Міграція схеми виконується над базою даних коли необхідно оновити, або повернути схему бази даних до якоїсь новішої чи старішої версії. Міграції виконуються програмно за допомогою інструменту міграцій (англ. *schema migration tool*). При виклику інструменту міграції з вказуванням бажаної версії схеми, інструмент автоматично застосовує або відкочує міграції в правильній послідовності аж поки не приведе базу даних до бажаного стану.[19]». Створення та оновлення бази даних таким методом, є дуже зручним та

ефективним, оскільки відпадає потреба в оновлені та створенні БД вручну, а також таке керування дозволяє змінювати серверну частину та БД синхронно. Після цієї дії була створена база даних PostgreSQL з відповідними таблицями.

У контролерах був прописаний ряд HTTP запитів, які реалізують взаємодію з клієнтською частиною.

Для того аби провести тестування роботи HTTP методів веб-сервісу перед їх використанням клієнтською частиною та сформувавши документацію програми, була згенерована Swagger документація – «це специфікація машиночитабельних файлів з інтерфейсами, для опису, створення, використання і візуалізації REST веб сервісів. Існують різноманітні інструменти що можуть генерувати код, документацію і тести за файлом з описом інтерфейсу [20]». Специфікація дозволяє, використовуючи користувальницький інтерфейс, протестувати які результати, у вигляді об'єктів JSON, XML або тексту, будуть повертати методи серверної частини.

3.3 Розробка інтерфейсу користувача.

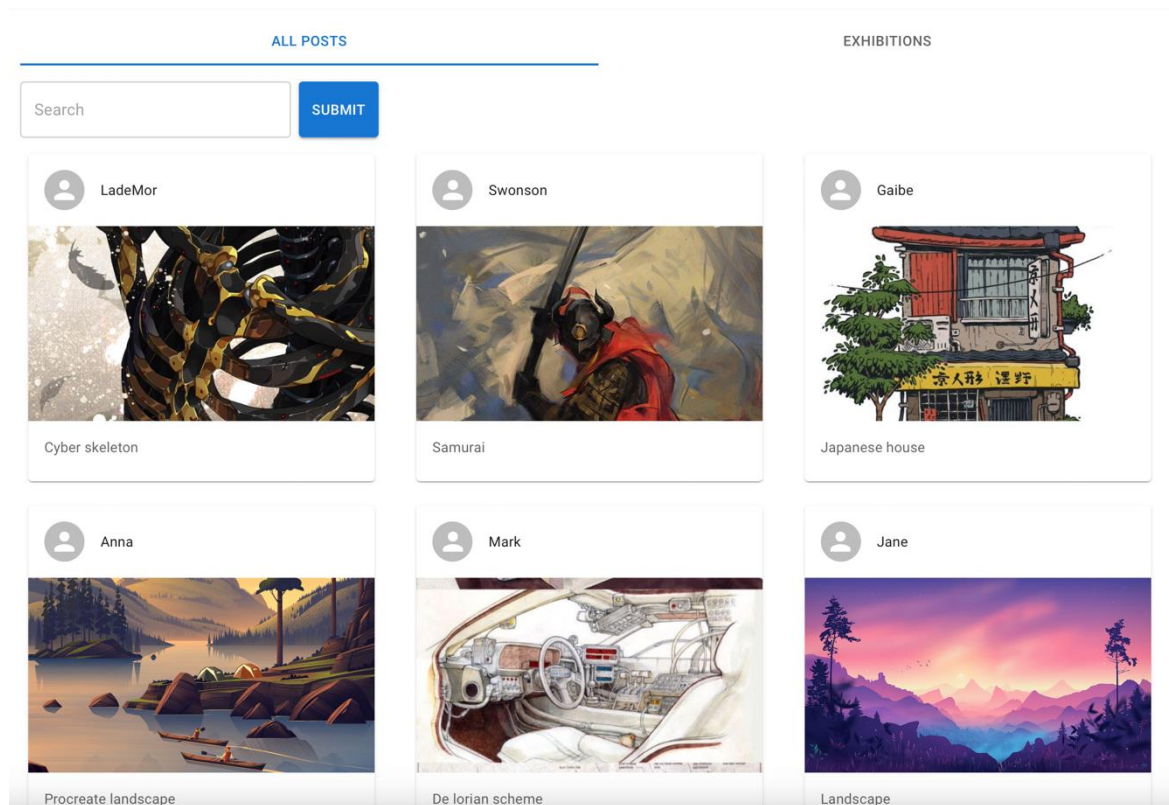
Після того як основні запити для роботи та взаємодією з базою даних програми були реалізовані на бек-енді, була розпочата робота над створенням клієнта. Він буде містити у собі алгоритми обробки запитів з сервера, а також додаткову логіку пошукових та сортувальних систем постів та виставок.

Після налаштування React проєкту, був створений компонент, який містить у собі навігаційну логіку між іншими компонентами, для його створення була використана бібліотека react-router-dom [21]. Її функціонал дозволяє створювати одно сторінкові додатки з декларативним роутингом.

Структура фронт-енд частини сервісу складається з декількох основних компонентів які були створені з урахуванням технічного завдання.

Хедер який статично знаходиться у будь-якому місці клієнта та відображає ряд посилань для навігації по додатку: логотип який за сумісництвом є посиланням на головну сторінку сайту, посилання з назвою «Main» яке теж є посиланням на головну сторінку, посилання «Registration» яке перенаправляє користувача на сторінку реєстрації, посилання «Login» яке перенаправляє на сторінку авторизації зареєстрованого користувача, іконки зі зображеннями календаря, дзвіночка та поштового листа які є посиланнями на сторінки за календарем виставок, повідомлень користувача та чату відповідно, а також зображення профілю користувача при натисканні на яку з'являється випадаючий список через який користувач може перейти на сторінку свого акаунта, вийти зі свого акаунта або перейти на сторінку адміністратора у випадку якщо користувач має відповідні права.

Компонент головної сторінки містить блок з двома вкладками на вибір, перша відображає список постів користувачів на сайті, при натисканні на кожний з них буде відкриватися сторінка конкретного посту, друга відображає список наявних на цей час виставок, при натисканні на будь-яку відображається сторінка виставки з поточною інформацією, а також зі список картин які на ній демонструються. Зовнішній вигляд головної сторінки веб-сервісу зображений на «Скріншот 3.1».



Скріншот 3.1 Головна сторінка веб-сервісу

Компоненти реєстрації та авторизації містять набір полів для введення даних для конкретної процедури, у разі помилки або успіху при валідації, користувачу будуть продемонстровані відповідні повідомлення.

Компонент календаря надає доступ до календаря на якому відображається інформація про розклад виставок, при натисканні на дату виставки буде відображатися інформація про конкретну подію.

Компоненти повідомлень та чату створені для реалізації функціонала соціальної мережі у веб-сервісі. Повідомлення надходять у вигляді елементів списку та містити інформацію про залишені коментарі під постами користувача або про майбутні виставки. У чаті користувачі будуть мати можливість спілкуватися один з одним.

Сторінка акаунту містить інформацію про користувача, відображати зроблені ним пости, а також надавати доступ до сторінок створення постів або виставок.

Сторінка адміністратора буде доступна для користувачів з правами «адміністратор», та надавати доступ до списку усіх зареєстрованих користувачів на сайті, можливість переходити на кожного користувача та передивлятись інформацію про нього. Адміністратор має право на блокування будь-якого користувача назавжди або на встановлений проміжок часу.

У кожному компоненті який містить у собі той чи інший вид списку: пости, виставки, список користувачів тощо, реалізований функціонал пошуку та сортувальна система.

3.4 Підготовка до тестування та деплою.

Після завершення написання основного функціонала виставкової онлайн системи, був розглянутий ряд варіантів для тестування та деплою майбутнього додатку.

Тестування програмного забезпечення – «це процес технічного дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому його мають використовувати. Техніка тестування також включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових із метою оцінки [19]». Для тестування веб-сервісів існує багато різних варіантів для реалізації, вони повинні охоплювати перевірку коду на правильність виконання функцій, перевірку на наявність потенційних загроз, перевірку на продуктивність, перевірку на взаємодію з іншими системами, перевірку взаємодії користувача з інтерфейсом клієнтської частини. Для створення автоматизованих перевірок розглядаються фреймворки та інструменти Selenium, Postman, JMeter та інші. Серед переглянутих програм, пріоритет у майбутньому використанні надається Postman[20], за інтуїтивній інтерфейс та велику кількість корисних функцій.

Деплой веб-сервісу (розгортання програмного забезпечення) – «це усі дії, що роблять програмну систему готовою до використання. Даний процес є частиною життєвого циклу програмного забезпечення. Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника, так і з боку споживача [21]». Для реалізації розгортки серверу існує декілька варіантів.

Перший це деплой на фізичному сервері який знаходиться в дата-центрі. Для таких операцій використовуються технології Docker або Kubernetes, що дозволяють запускати веб-сервіси у контейнерах.

Другим варіантом є використання хмарного сховища, реалізувати який можна за допомогою сервісів Amazon Web Services або Microsoft Azure. Деплой подібним шляхом надає можливість швидко налаштувати сервер та забезпечити потрібні ресурси, проте для такого варіанту потрібно вивчити наявні плани підписок на ці сервіси аби розподілити фінанси проекту без зайвих витрат. Серед сервісів розгортки з використанням хмарного сховища розглядається Microsoft Azure[22].

3.5 Плани подальшого розвитку проекту.

З огляду на те що майбутній продукт буде виходити на масштабний ринок веб-сервісів галерей, він повинен дотримуватися тенденції регулярних оновлень та розвитку. Своєчасна розробка та реалізація нового функціоналу буде підтримувати інтерес цільової аудиторії, а також надавати більше унікальних особливостей продукту та виділяти його з поміж конкурентів. Серед функціоналу запланованого на майбутнє виділено декілька ідей з найбільшим пріоритетом:

- Інтегрування платіжних систем. Цей функціонал буде підтримувати платіжні картки різних міжнародних банків, а також надавати можливість прив'язувати їх до облікового запису.
- Колаборація з відомими художніми спілками. У веб-сервісі будуть створені особливі сторінки та виставки, якими будуть керувати відомі мистецькі діячі.
- Розширення функціонала створення виставок. Оскільки кожна виставка представляє відповідну тематику їх автори будуть зацікавлені у більш розширеному функціоналі створення онлайн виставок, які б надавали унікальних аспектів кожній. До такого функціонала входить: встановлення кастомного фону, вибір основних та другорядних кольорів для теми виставки, вибір шрифтів тощо.
- Підтримка мов різних країн світу. За для більшої інтеграції сервісу на міжнародний рівень, інтерфейс буде перекладений на різні мови світу які будуть встановлюватися в залежності від регіону у якому проживає користувач.

Окрім функціонала який розрахований на загальний вектор розвитку додатка, він також буде отримувати нововведення в залежності від змін потреб та інтересів цільової аудиторії.

ВИСНОВКИ ДО ТРЕТЬОГО РОЗДІЛУ

У третьому розділі були описані усі етапи розробки виставкової онлайн системи, а саме: вибір технологій для створення додатка, розробка серверної та розробка клієнтської частини.

Перераховані технології які використовувались при розробці проекту, а також описані їх переваги за якими вони були обрані.

Описана імплементація серверної частини, вибір шаблону проектування бек-енду, а також описана структура основного функціонала.

Описаний користувацький інтерфейс веб-сервісу, перераховані компоненти які були реалізовані та їх взаємодія один з одним.

ВИСНОВКИ

Наприкінці роботи повністю розроблено прототип програмного продукту.

За допомогою аналізу ринку була визначена цільова аудиторія та особливості подібних програм, що дозволило створити конкурентоспроможний продукт.

Визначені технічні вимоги та створені схеми для створення безперебійної роботи баз даних, та взаємодії користувачів на самій платформі.

Обрані основні технології для розробки клієнтської частини, серверної частини та бази даних.

Розроблена система виставок, яка дозволяє користувачам демонструвати та переглядати цифрові витвори на спеціальних тематичних подіях, а також система платних підписок яка дозволяє розподіляти можливості користувачів на сайті.

Платформа також включає функціонал соціальної мережі, який дозволяє налагодити взаємодію між авторами та звичайними користувачами.

У результаті розробки веб-сервіс підготовлений для тестування потенційними користувачами, та підготовлені технології для розгортки майбутнього продукту на виділеному сервері.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pinterest. URL: <https://www.pinterest.com>. (дата звернення: 04.04.2023).
2. DeviantArt. URL: <https://www.deviantart.com> (дата звернення: 04.04.2023).
3. ArtStation. URL: <https://www.artstation.com> (дата звернення: 04.04.2023).
4. Невзаємозамінний токен визначення, Вікіпедія. URL: https://uk.wikipedia.org/wiki/Невзаємозамінний_токен (дата звернення: 04.04.2023).
5. SWOT – аналіз визначення, Вікіпедія. URL: <https://uk.wikipedia.org/wiki/SWOT-аналіз> (дата звернення: 04.04.2023).
6. Веб-сервіс визначення, Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Вебслужба> (дата звернення: 04.04.2023).
7. API визначення, Вікіпедія. URL: https://uk.wikipedia.org/wiki/Прикладний_програмний_інтерфейс (дата звернення: 04.04.2023).
8. REST визначення, Вікіпедія. URL: <https://uk.wikipedia.org/wiki/REST> (дата звернення: 04.04.2023).
9. PostgreSQL 15 Documentation. URL: <https://www.postgresql.org/docs/current/> (дата звернення: 04.04.2023).
10. Діаграма класів визначення, Вікіпедія. URL: https://uk.wikipedia.org/wiki/Діаграма_класів (дата звернення: 04.04.2023).

11. Діаграма послідовності визначення, Вікіпедія. URL: https://uk.wikipedia.org/wiki/Діаграма_послідовності (дата звернення: 04.04.2023).
12. ASP.Net documentation, Microsoft. URL: <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0> (дата звернення: 04.04.2023).
13. React. JavaScript бібліотека для створення користувацьких інтерфейсів, ReactJs. URL: <https://uk.reactjs.org> (дата звернення: 04.04.2023).
14. Material UI – Overview, MUI Core. URL: <https://mui.com> (дата звернення: 04.04.2023).
15. Веб сервер(серверна частина веб-сервісу), Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Вебсервер> (дата звернення: 05.04.2023).
16. Міграція бази даних визначення, Вікіпедія. URL: https://uk.wikipedia.org/wiki/Міграція_бази_даних (дата звернення: 05.04.2023).
17. Специфікація OpenAPI/Swagger визначення, Вікіпедія. URL: <https://uk.wikipedia.org/wiki/OpenAPI> (дата звернення: 05.04.2023).
18. Feature overview, React Router. URL: <https://reactrouter.com/en/main/start/overview#feature-overview> (дата звернення 05.04.2023).
19. Тестування програмного забезпечення визначення, Вікіпедія. URL: https://uk.wikipedia.org/wiki/Тестування_програмного_забезпечення (дата звернення: 06.04.2023).
20. Postman, Generating API Documentation. URL: <https://learning.postman.com/docs/publishing-your-api/documenting-your-api/> (дата звернення: 07.04.2023).
21. Розгортання програмного забезпечення визначення, Вікіпедія. URL:

https://uk.wikipedia.org/wiki/Розгортання_програмного_забезпечення (дата звернення: 06.04.2023).

22. Cloud Computing Services, Microsoft Azure. URL: <https://azure.microsoft.com/en-us> (дата звернення: 07.04.2023).

ДОДАТКИ

ДОДАТОК А. КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Пяцько Сергій Сергійович, учасник освітнього процесу Херсонського державного університету, **УСВІДОМЛЮЮ**, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

ЗАВЯЛЮ, що у своїй освітній і науковій діяльності **ЗОВОВ'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)

Сергій Пяцько
(ім'я, прізвище)



ДОДАТОК Б. ДІАГРАМИ ПОСЛІДОВНОСТЕЙ.

