МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ/факультет Інститут економіки та бізнес-освіти

Кафедра Економіки та цифрового бізнесу

Спеціальність Комп'ютерні науки

Форма навчання Денна

КВАЛІФІКАЦІЙНА РОБОТА

Томіло Микити Володимировича

(прізвище, ім'я, по батькові здобувача)

на тему Розробка інтерактивної веб-платформи для замовлення 3D-графічних послуг

(повна назва теми)

за матеріалами

(повна назва бази Дослідження)

науковий керівник

_____к.е.н, доцент_____ __ ___ _Соловйова В. В._____
(наук. ступінь, вчене звання) (Підпис) (прізвище, ініціали)

Робота допущена до захисту в ЕК
Протокол засідання кафедри
від 09 червня____2025__р. № 12_____
Завідувач кафедри _____
(підпис)
К.е.н., доцент В.М. Радько
Наук. ступінь, вчене звання *Ініціали, прізвище*

Кривий Ріг – 2025

ЗАТВЕРДЖЕНО
Наказ Міністерства освіти і науки, молоді та спорту України
29 березня 2012 року № 384

Форма № Н-9.01

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ ТА БІЗНЕС-ОСВІТИ
(повне найменування вищого навчального закладу)

Кафедра економіки та цифрового бізнесу
Освітній ступінь бакалавр
Спеціальність ___Комп'ютерні науки_____

ЗАТВЕРДЖУЮ
Завідувач кафедри _____В.М. Радько

"07" квітня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА ЗДОБУВАЧУ

ТОМІЛО МИКИТІ ВОЛОДИМИРОВИЧУ

1. Тема роботи _Розробка інтерактивної веб-платформи для замовлення 3D-графічних послуг

науковий керівник роботи _____,
затверджені наказом вищого навчального закладу від «04» квітня 2025 р. № 224-ст (д/ф)
№ 151-ст (з/ф)

2. Строк подання здобувачем роботи 31.05.2025р.

3. Зміст кваліфікаційної роботи бакалавра, об'єкт, предмет та мета дослідження:
Розділ 1 _Теоретичні основи розробки веб-платформи у сфері 3d-послуг

Розділ 2 _ Проєктування та розробка інтерактивної веб-платформи для замовлення 3d-графічних послуг_____

Розділ 3 _ Тестування, аналіз та перспективи розвитку веб-платформи у сфері 3d-послуг_____
Об'єкт дослідження – _ процес розробки веб-платформ для комерційної взаємодії у сфері 3D-графіки._____
Предмет дослідження - _ розробка інтерактивної веб-платформи для замовлення 3D-графічних послуг._____
Мета кваліфікаційної роботи бакалавра – _ розробка інтерактивної веб-платформи для компанії TomìLonì, яка спеціалізується на наданні 3D-графічних послуг, із можливістю демонстрації робіт, обробки замовлень та синхронізації з Telegram-каналом._____

4. Дата видачі завдання 04.04.2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	до 28.04.2025р.	25.04.2025
2	Підготовка розділу 2	до 16.05.2025р.	15.05.2025
3	Підготовка розділу 3	до 30.05.2025р.	29.05.2025
4	Реєстрація завершеної дипломної роботи	до 31.05.2025р.	30.05.2025
5	Отримання відгуку від наукового керівника	03-04.06.2025р.	04.06.2025
6	Отримання зовнішньої рецензії	05-06.06.2025р.	06.06.2025
7	Перевірка кваліфікаційної роботи на плагіат	02-09.06.2025р.	04.06.2025
8	Попередній захист кваліфікаційної роботи на кафедрі	03.06.2025р.	03.06.2025
9	Допуск кафедрою кваліфікаційної роботи до захисту	09.06.2025р.	09.06.2025
10	Підготовка студента до захисту в ЕК	до 17.06.2025р.	17.06.2025

Завдання підготував науковий керівник _____ Соловйова В.В._____
(підпис) (прізвище та ініціали)

Завдання одержав здобувач _____ ___ ___Томіло М.В.___
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Робота містить 83 сторінки, 19 рисунків, 70 джерел, 3 таблиці, 2 додатка.

Об’єкт дослідження: процес розробки та функціонування інтерактивних веб-платформ для надання послуг у сфері 3D-графіки.

Предмет дослідження - розробка інтерактивної веб-платформи для замовлення 3D-графічних послуг.

Ціль роботи: Розробка функціональної веб-платформи для ефективного замовлення 3D-графічних послуг, що забезпечує зручну взаємодію між замовниками та виконавцями, автоматизує обробку замовлень та покращує візуалізацію портфоліо 3D-моделей.

Методи дослідження і перелік апаратури: Застосовано системний аналіз існуючих рішень, методи проектування UI/UX, структурного та об'єктно-орієнтованого програмування. Використано: HTML5, CSS3 (Bootstrap), JavaScript (з бібліотеками/фреймворками), PHP, MySQL, а також інструменти для роботи з 3D-графікою (Three.js/Model-viewer).

Результати та їх новизна: Розроблено інтерактивну веб-платформу, що дозволяє користувачам переглядати портфоліо 3D-моделей, оформлювати замовлення та взаємодіяти з виконавцями. Новизна полягає у комплексній інтеграції функціоналу замовлення та інтерактивного відображення 3D-моделей у браузері, що підвищує ефективність комунікації та мінімізує неточності.

Основні конструктивні, технологічні й техніко-експлуатаційні характеристики та показники: Платформа крос-браузерна, адаптивна, підтримує відображення 3D-моделей (GLB/GLTF), має систему авторизації, форми зворотного зв'язку, інтерактивний каталог робіт. Характеризується високою швидкістю завантаження та стабільною роботою.

Інформація щодо впровадження: Платформа готова до практичного впровадження як інструмент для компаній або фахівців, що надають 3D-графічні послуги, або як SaaS-рішення.

Взаємозв’язок з іншими роботами: Робота ґрунтується на аналізі кращих практик у веб-розробці та 3D-візуалізації, враховуючи досвід існуючих платформ

Рекомендації щодо використання результатів роботи: Результати рекомендується використовувати для створення спеціалізованих онлайн-сервісів замовлення 3D-графічних послуг, портфоліо-платформ, а також як основу для навчальних ресурсів.

Область застосування: Маркетинг, реклама, промисловий дизайн, архітектурна візуалізація, розробка ігор, створення VR/AR, освіта та інші галузі з потребою у 3D-графічному контенті.

Економічна чи соціально-економічна ефективність роботи: Економічна ефективність полягає у спрощенні та прискоренні процесу замовлення, зменшенні витрат на комунікацію та підвищенні конверсії. Соціальна ефективність проявляється у розширенні доступу до якісних послуг та створенні нових можливостей для фахівців.

Значимість роботи: Робота має високу практичну значимість, пропонуючи ефективне рішення для автоматизації та оптимізації процесу замовлення 3D-графічних послуг в умовах зростаючого попиту.

Висновки, пропозиції щодо розвитку об’єкта дослідження (розроблення) і доцільності продовження досліджень: Платформа є функціональним прототипом. Для подальшого розвитку пропонується інтеграція платіжних систем, особистого кабінету виконавців, впровадження ШІ для оцінки замовлень та рекомендації виконавців, а також розширення функціоналу для підтримки VR/AR-контенту. Продовження досліджень доцільне у напрямку автоматичної генерації 3D-моделей та інтеграції з хмарними сервісами рендерингу.

Ключові слова:

Веб-платформа, 3D-графіка, Замовлення, Інтерактивний, Онлайн-сервіс, Портфоліо, Розробка, Фідбек, UI/UX дизайн, Візуалізація.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Скорочення	Розшифрування
API	Application Programming Interface - інтерфейс прикладного програмування
CRUD	Create, Read, Update, Delete - створення, читання, оновлення, видалення
CSS	Cascading Style Sheets - каскадні таблиці стилів
DB	Database - база даних
ERP	Enterprise Resource Planning - система планування ресурсів підприємства
HTML	HyperText Markup Language - мова гіпертекстової розмітки
HTTP	Hypertext Transfer Protocol - протокол передачі гіпертексту
JS	JavaScript - мова програмування
JSON	JavaScript Object Notation - формат обміну даними
MongoDB	Документно-орієнтована NoSQL база даних
MVC	Model-View-Controller - модель-подання-контролер
NoSQL	Not only SQL - нереляційна база даних
REST	Representational State Transfer - архітектурний стиль взаємодії сервісів
REST API	REST Application Programming Interface - API на основі REST
UI	User Interface - інтерфейс користувача
UX	User Experience - досвід користувача
3D	Three-dimensional - тривимірний (графіка, моделі, візуалізація)

ВСТУП

У сучасному цифровому світі зростає попит на якісний 3D-контент, який використовується в різноманітних сферах - від архітектурної візуалізації до створення анімацій, освітніх відео та комп'ютерних ігор. Компанії, які надають 3D-послуги, стикаються з потребою у зручних інструментах для взаємодії з клієнтами. Веб-платформи стають ключовим каналом для демонстрації портфоліо, прийому замовлень та автоматизації обробки заявок. Тема даної кваліфікаційної роботи - розробка інтерактивної веб-платформи для замовлення 3D-графічних послуг - є надзвичайно актуальною, адже вона дозволяє оптимізувати бізнес-процеси у сфері 3D-графіки та забезпечити комфорт клієнтам і виконавцям.

Метою роботи є розробка інтерактивної веб-платформи для компанії TomìLonì, яка спеціалізується на наданні 3D-графічних послуг, із можливістю демонстрації робіт, обробки замовлень та синхронізації з Telegram-каналом.

Основні завдання:

1) проаналізувати існуючі рішення у сфері веб-платформ для замовлення 3D-послуг;

2) обґрунтувати вибір технологій для реалізації проєкту;

3) спроєктувати архітектуру клієнтської та серверної частин веб-додатку;

4) реалізувати функціонал оформлення замовлення, демонстрації виконаних робіт, подачі заявки на роботу;

5) забезпечити автоматичну інтеграцію новин з Telegram-каналу;

6) протестувати основні функціональні можливості платформи;

7) оцінити ефективність реалізованого рішення.

Об'єкт дослідження: процес розробки та функціонування інтерактивних веб-платформ для надання послуг у сфері 3D-графіки.

Предмет дослідження - розробка інтерактивної веб-платформи для замовлення 3D-графічних послуг.

Для реалізації поставлених завдань використовуються такі методи:

1) методи системного аналізу для визначення вимог;
2) методи проєктування клієнт-серверних архітектур;
3) програмування з використанням HTML, CSS, JavaScript, Node.js та MongoDB;
4) моделювання бази даних;
5) функціональне тестування за допомогою Postman.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі проводиться аналіз предметної області та огляд існуючих технологій. Другий розділ присвячений проєктуванню архітектури платформи та реалізації клієнтської і серверної частини. У третьому розділі описується процес реалізації, тестування та оцінки функціональності системи. У висновках подано узагальнення отриманих результатів та напрями подальшого розвитку платформи.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБ-ПЛАТФОРМИ У СФЕРІ 3D-ПОСЛУГ

1.1 Аналіз сучасного стану ринку 3D-графіки для замовлення 3D-графічних послуг

У сучасному цифровому середовищі 3D-графіка стала невід'ємною складовою візуальної комунікації та презентації інформації. Досліджено, що попит на 3D-графічні послуги зростає внаслідок розвитку таких галузей, як архітектура, геймдев, реклама, кіновиробництво, медицина та освіта.

Проаналізовано, що основні напрями 3D-послуг включають:

a) 3D-моделювання - створення цифрових об'єктів для промислового дизайну, архітектурної візуалізації та розробки відеоігор;

b) 3D-візуалізація - рендеринг моделей із застосуванням освітлення, текстур і камер;

c) 3D-анімація - динамізація об'єктів для створення відеоконтенту, інфографіки та персонажів;

d) 3D-сканування - створення цифрових копій фізичних об'єктів;

e) Інтерактивні рішення - продукти для AR/VR, WebGL, Unity, Unreal Engine.

У роботі [1] розкрито, що попит на 3D-анімацію у світі зростає щорічно. У 2023 році обсяг ринку сягнув понад 21 млрд доларів США, з прогнозованим зростанням до 55 млрд до 2031 року. Це обумовлено впровадженням 3D-рішень у медіа, медицині та електронній комерції.

Згідно з [2], у медицині 3D-візуалізація активно використовується для створення цифрових моделей людських органів, що дозволяє лікарям точніше

планувати операції та проводити віртуальну симуляцію втручань. Доведено, що це сприяє підвищенню безпеки пацієнтів та зменшенню витрат на навчання персоналу.

У статті [3] наголошено, що в Україні ринок 3D-послуг представлений здебільшого невеликими студіями або окремими фахівцями-фрилансерами. Встановлено, що великі замовлення часто передаються за кордон, оскільки на внутрішньому ринку відсутні платформи, які забезпечують централізовану взаємодію між клієнтом і виконавцем.

На таблиці 1.1 схематично подано типову класифікацію 3D-послуг залежно від цільового замовника.

Таблиця 1.1

Класифікація 3D-послуг залежно від цільового замовника

Напрям використання	**Галузі застосування**	**Типи послуг**
Архітектура	будівництво	моделювання, рендеринг
Ігри	геймдев	анімація, інтерактив
Реклама	digital-маркетинг	візуалізація, анімація
Освіта	навчальні продукти	3d-моделі, інтерактив
Медицина	симуляції, візуалізації	сканування, моделювання
VR/AR	інтерактивний контент	інтеграція, візуалізація

Примітка. Джерело: розроблено автором

Розкрито, що вітчизняні компанії найчастіше використовують сайти-візитки або прості портфоліо без автоматизованих систем обробки замовлень. Наприклад, на сайтах студій ABD Video та FeelMake, які досліджено в роботі [4], реалізовано лише базову взаємодію із замовником через контактну форму. Відсутні системи авторизації, відстеження замовлень чи інтеграції з месенджерами. Узагальнено, що

такі рішення обмежують зростання студій та знижують ефективність обслуговування клієнтів.

У дослідженні [5] доведено, що іноземні платформи (наприклад, CGTrader, Fiverr, ArtStation) дозволяють демонструвати виконані роботи або продавати готові моделі, проте не підтримують повноцінну інтеграцію з CRM, Telegram, не дають змоги клієнтам відслідковувати хід виконання замовлення або подавати заявку на співпрацю у структурованій формі.

Проведено порівняльний аналіз, згідно з яким жодна з вказаних платформ не поєднує такі функції як автоматизований прийом замовлень, демонстрація виконаних робіт, інтеграція з Telegram та подача заявки на співпрацю. Як наслідок, визначено доцільність створення спеціалізованої веб-платформи, яка об’єднає функціонал CRM-системи, порталу для клієнтів і вітрини студії одночасно.

Таким чином, проаналізовано, що створення власної веб-платформи для 3D-студії дозволяє адаптуватися до потреб бізнесу, виключити зайві ланки комунікації, знизити витрати на зовнішні сервіси та підвищити контроль за якістю та строками виконання робіт.

Створення спеціалізованої веб-платформи для 3D-студії, яка поєднуватиме галерею виконаних робіт, CRM-функціонал, автоматизовану обробку замовлень та комунікацію через Telegram, дозволить вирішити виявлені проблеми, покращити сервіс та залучити більше замовників.

1.2. Дослідження технологій веб-розробки для замовлення 3D-графічних послуг

У сучасних умовах розвитку цифрових технологій створення інтерактивних та масштабованих вебплатформ для замовлення 3D-графічних послуг вимагає використання ефективних інструментів веб-розробки. Досліджено, що ключовими

технологіями у цьому напрямі є HTML, CSS, JavaScript, Node.js, MongoDB та REST API, які забезпечують гнучкість, продуктивність та масштабованість вебзастосунків.

HTML, CSS та JavaScript

HTML (HyperText Markup Language) є основою структури вебсторінок, дозволяючи розміщувати контент та визначати його ієрархію. CSS (Cascading Style Sheets) відповідає за візуальне оформлення елементів, забезпечуючи адаптивність та привабливість інтерфейсу.

JavaScript, як мова програмування, дозволяє реалізовувати динамічну поведінку елементів, взаємодію з користувачем та обробку подій у реальному часі.

Проаналізовано, що використання цих технологій у сукупності забезпечує створення інтерактивного та зручного інтерфейсу для користувачів вебплатформи, що є критично важливим для залучення та утримання клієнтів у сфері 3D-графічних послуг.

Node.js

Node.js є середовищем виконання JavaScript на серверній стороні, що дозволяє розробникам використовувати одну мову програмування як для клієнтської, так і для серверної частини застосунку. Визначено, що Node.js базується на подієво-орієнтованій, неблокуючій моделі вводу/виводу, що забезпечує високу продуктивність та масштабованість вебзастосунків [6].

Досліджено, що Node.js активно використовується для розробки вебсерверів, API, мікросервісів та реального часу застосунків, таких як чати та онлайн-ігри.

Його екосистема, зокрема менеджер пакетів npm, надає розробникам доступ до широкого спектру бібліотек та інструментів, що спрощує та прискорює процес розробки [7].

З метою обґрунтування вибору інструментів для реалізації функціональності веб-платформи систематизовано основні технології, які доцільно застосувати при розробці інтерактивного середовища замовлення 3D-графіки (таблиця 1.2).

Таблиця 1.2

Основні технології веб-розробки для платформи замовлення 3D-графіки

Технологія	**Призначення**	**Переваги**
HTML	структура вебсторінки	стандартизований формат, сумісність з усіма браузерами
CSS	візуальне оформлення	адаптивність, можливість створення привабливого дизайну
JavaScript	динамічна поведінка	інтерактивність, обробка подій у реальному часі
Node.js	серверна логіка	висока продуктивність, єдина мова для фронтенду та бекенду
MongoDB	зберігання даних	гнучкість структури, масштабованість, висока доступність
REST API	взаємодія між компонентами	стандартизований обмін даними, легкість інтеграції

Примітка. Джерело: розроблено автором

MongoDB

MongoDB є документно-орієнтованою NoSQL базою даних, яка зберігає дані у форматі BSON (Binary JSON). Визначено, що MongoDB забезпечує гнучкість у структурі даних, дозволяючи зберігати складні та змінні за структурою об'єкти без необхідності попереднього визначення схеми [8].

Проаналізовано, що MongoDB підтримує горизонтальне масштабування через шардинг, що дозволяє розподіляти дані по кількох серверах для забезпечення

високої доступності та продуктивності. Крім того, вбудовані механізми реплікації та індексації сприяють підвищенню надійності та швидкості доступу до даних [9].

REST API

REST (Representational State Transfer) є архітектурним стилем для побудови вебсервісів, що використовує стандартні HTTP-методи для взаємодії між клієнтом та сервером. Досліджено, що REST API дозволяє створювати масштабовані та гнучкі інтерфейси для обміну даними між різними компонентами системи [10].

У контексті вебплатформи для замовлення 3D-графічних послуг, REST API забезпечує ефективну інтеграцію між фронтендом та бекендом, а також можливість взаємодії з зовнішніми сервісами, такими як платіжні системи або сервіси обробки зображень.

Окрему увагу приділено технологіям аутентифікації та інтеграції з месенджерами. Визначено, що для реалізації безпечної авторизації користувачів доцільно використовувати бібліотеки JWT (JSON Web Token)**.** Крім того, допускається впровадження Telegram-бота для оперативного інформування менеджера студії про нові заявки, що значно спростить процес адміністрування та покращить взаємодію з клієнтами .

Їхнє поєднання забезпечує гнучкість, продуктивність та можливість швидкої адаптації до змінних вимог ринку.

1.3. Формулювання завдань розробки для замовлення 3D-графічних послуг

Розробка веб-платформи для замовлення 3D-графічних послуг повинна ґрунтуватися на ретельному аналізі ринку та існуючих технологій. У ході дослідження було визначено основні функціональні блоки, необхідні для успішної реалізації платформи, зокрема система замовлень, демонстрація виконаних робіт,

заявка на співпрацю, авторизація та Telegram-інтеграція. Усі ці блоки повинні забезпечувати зручність користувачів і оптимізувати процес взаємодії між замовниками та виконавцями.

1) Система замовлень

Система замовлень повинна надавати користувачам можливість розміщувати замовлення на 3D-графічні послуги з детальним описом вимог і специфікацій. Цей блок є основою для платформи, оскільки дозволяє ефективно організувати роботу між замовниками та виконавцями, а також формувати базу даних для автоматичного оброблення замовлень. У дослідженнях, проведених Каплуном та Стельмахом [9], зазначено, що система замовлень повинна бути інтуїтивно зрозумілою та забезпечувати швидке введення даних з можливістю додавання файлів і приміток. Також повинні бути передбачені функції для зазначення термінів виконання та інших критичних параметрів проекту.

2) Демонстрація виконаних робіт

Для забезпечення довіри між замовниками та виконавцями, платформа повинна надавати можливість публікації вже виконаних 3D-проектів, що дозволяє користувачам оцінити рівень майстерності виконавців. Важливою складовою є функціональність категоризації та фільтрації робіт. Чижевський [10] у своїй роботі підкреслює, що демонстрація виконаних робіт через візуалізацію дозволяє потенційним замовникам швидше знайти відповідного виконавця за допомогою спеціальних фільтрів. Крім того, наявність відгуків і рейтингів дозволяє оцінювати не тільки якість роботи, але й рівень комунікації та професіоналізму виконавців.

3) Заявка на співпрацю

Блок заявки на співпрацю дозволяє користувачам ініціювати контакт з виконавцями, уточнювати деталі майбутніх проектів, а також обговорювати умови співпраці. Важливою характеристикою цього блоку є можливість інтерактивного спілкування, що підвищує рівень комунікації та дозволяє швидко отримувати

відповіді на питання. Як зазначає Лук'яненко [11], інтерактивні заявки сприяють формуванню більш тісних зв'язків між замовниками та виконавцями, що в свою чергу покращує якість виконання проектів.

4) Авторизація

Авторизація на платформі є важливим елементом для забезпечення безпеки особистих даних користувачів. Пропонується реалізація багатоступеневої авторизації, включаючи використання сторонніх сервісів, таких як Google або Facebook. Відповідно до дослідження Ніколаєва [12], застосування двофакторної аутентифікації є важливим елементом для захисту особистої інформації користувачів і запобігання несанкціонованому доступу до їхніх акаунтів.

5) Telegram-інтеграція

Інтеграція з популярними месенджерами, зокрема з Telegram, дозволяє користувачам платформи отримувати сповіщення про зміни в замовленнях, нові заяви на співпрацю та інші важливі оновлення безпосередньо в мобільному додатку. Згідно з дослідженням Романова [13], інтеграція з месенджерами покращує комунікацію і дозволяє замовникам та виконавцям швидше реагувати на зміни.

Теоретичний аналіз та обґрунтування завдань

Аналіз сучасних платформ для замовлення 3D-графічних послуг, таких як портфоліо веб-сайтів, показує, що важливими факторами є зручність користування та швидкість взаємодії. Пропоновані функціональні блоки платформи відповідають основним вимогам ринку і мають на меті спрощення процесу замовлення та виконання 3D-проектів. Як зазначають експерти ринку 3D-послуг, таких як Каплун В. П. та Стельмах А. М. [9], інтеграція з іншими сервісами та забезпечення функцій фільтрації і рейтингів дозволяє значно підвищити ефективність роботи платформи та її конкурентоспроможність.

Проведений аналіз дозволяє зробити висновок, що для успішної реалізації платформи для замовлення 3D-графічних послуг необхідно включити в її структуру зазначені ключові функціональні блоки. Це забезпечить високий рівень взаємодії між користувачами та виконавцями, що є важливим для розвитку ефективної та зручної платформи.

1.4 Оцінка наявних рішень для замовлення 3D-графічних послуг

Ринок 3D-графічних послуг є динамічним та активно розвивається, що спричиняє появу різних платформ для замовлення таких послуг. Одними з найбільш популярних платформ на сьогодні є Upwork, Artstation, Feelmake, ABD-Video, а також інші аналогічні ресурси. У цьому розділі здійснено порівняльний аналіз існуючих рішень для замовлення 3D-графічних послуг та виявлено обмеження цих платформ в контексті задач компанії, що розробляє інтерактивну веб-платформу.

1) Платформа Upwork

Upwork є однією з найбільших міжнародних платформ для фрілансерів, яка включає в себе широкий спектр послуг, серед яких і 3D-графіка. На платформі можна знайти фахівців з різних галузей, включаючи анімацію, моделювання, візуалізацію тощо. Згідно з аналізом, проведеним Стельмахом [14], одним з основних переваг Upwork є глобальність і великий вибір виконавців. Однак платформа має певні обмеження: високі комісії, складний процес відбору фахівців і відсутність інструментів для прямої комунікації між замовником і виконавцем, що ускладнює взаємодію.

2) Платформа Artstation

Artstation - це платформа, орієнтована на професіоналів у галузі цифрового мистецтва, зокрема в 3D-візуалізації. Платформа є популярною серед художників та дизайнерів, що працюють у сфері 3D-графіки.

З метою виявлення ключових переваг і недоліків існуючих платформ, орієнтованих на послуги з 3D-графіки, проведено порівняльний аналіз найбільш популярних сервісів, результати якого подано в таблиці 1.3.

Таблиця 1.3

Порівняння платформ

Платформа	Переваги	Обмеження
Upwork	велика кількість фахівців, глобальність.	високі комісії, складний процес відбору виконавців.
Artstation	високий рівень професіоналізму, можливість створення портфоліо.	відсутність функцій для розміщення замовлень, обмежена комунікація.
Feelmake	орієнтація на бізнес, інтеграція з іншими сервісами.	відсутність автоматизації процесу комунікації, обмежені інструменти.
ABD-Video	спеціалізація на рекламі та анімаціях, комплексні послуги.	складний інтерфейс, фокус на специфічних нішах.
CGTrader	широкий вибір фахівців, доступність для малих замовників.	високі комісії, складний процес перевірки фахівців.
Fiverr	швидкість виконання замовлень, доступність.	низька якість послуг за низькою ціною, обмежена можливість вибору.

Примітка. Джерело: розроблено автором

Перевагою Artstation є високий рівень професіоналізму користувачів та можливість демонстрації робіт у вигляді портфоліо. Як зазначено у дослідженні Каплуна [15], Artstation дозволяє створювати індивідуальні профілі та портфоліо, що сприяє розвитку особистих брендів. Однак, платформа не надає зручних механізмів для розміщення замовлень або ведення переговорів, що обмежує її використання для компаній, які хочуть швидко знаходити виконавців для конкретних завдань.

3) Платформа Feelmake

Платформа Feelmake зосереджена на пропозиції 3D-графічних послуг для бізнесу, зокрема на створенні 3D-візуалізацій для маркетингових кампаній та розробці анімацій. Вона пропонує інтеграцію з іншими сервісами, що дозволяє легко залучати нових клієнтів.

У дослідженні Гончаренко [16] зазначено, що основною перевагою Feelmake є орієнтація на бізнес-клієнтів, що дозволяє забезпечити високий рівень замовлень і стабільний попит на послуги. Однак, як і в випадку з іншими платформами, відсутність функціоналу для детального опису вимог і автоматизації процесу комунікації знижує ефективність використання.

4) Платформа ABD-Video

ABD-Video спеціалізується на виробництві 3D-анімації та візуалізацій для реклами та інших цілей. Платформа орієнтована на замовлення від бізнес-клієнтів та пропонує комплексні послуги з розробки 3D-контенту. Однак її обмеження полягають у відсутності функціоналу для інтеграції зі сторонніми сервісами та складності в пошуку виконавців для специфічних задач, що може уповільнювати процес виконання проектів. Згідно з результатами аналізу, проведеного Чижевським [17], основним недоліком є обмежений інтерфейс для користувачів та фокус на конкретних нішах ринку, що не дозволяє платформі залучати ширшу аудиторію.

5) Інші платформи

Крім вищезгаданих платформ, існують і інші ресурси, такі як CGTrader, Freelancer та Fiverr, що також пропонують послуги з 3D-графіки. Перевагою цих платформ є доступність і широкий вибір фахівців. Однак, як зазначає Романов [18], вони мають свої обмеження, зокрема високі комісії, складний процес перевірки фахівців та відсутність інструментів для довгострокової співпраці з постійними клієнтами.

Проведений порівняльний аналіз існуючих платформ для замовлення 3D-графічних послуг дозволяє зробити висновок, що жодна з них не є ідеальним рішенням для потреб компанії, яка прагне забезпечити максимальну зручність і ефективність процесу взаємодії між замовниками та виконавцями. Основними обмеженнями є високі комісії, відсутність належної автоматизації комунікації, а також недосконаліть процесів перевірки та відбору виконавців. Тому розробка власної платформи, що об'єднує переваги існуючих рішень та усуває їх обмеження, є обґрунтованою та перспективною стратегією.

1.5. Узагальнення результатів аналізу для замовлення 3D-графічних послуг

Виконаний аналіз існуючих платформ для замовлення 3D-графічних послуг дозволяє зробити низку важливих висновків, які підтверджують необхідність створення власної веб-платформи для автоматизації процесу прийому замовлень та розвитку бренду студії. У цьому розділі буде розкрито ключові результати дослідження, виявлені обмеження існуючих рішень, а також обґрунтовано потребу у розробці нового сервісу, що забезпечить зручність і ефективність для користувачів.

1) Необхідність автоматизації прийому замовлень

Проаналізовані платформи, такі як Upwork, Artstation, Feelmake, ABD-Video та інші, показали, що існуючі рішення не завжди здатні забезпечити повноцінну автоматизацію процесу прийому замовлень. Згідно з дослідженням Гончаренка [19], відсутність інтерфейсів для швидкої подачі заявок, а також недостатня автоматизація комунікації між замовниками та виконавцями є значними обмеженнями. Як було визначено, на багатьох платформах процес пошуку

виконавця і укладання угоди потребує великої кількості часу та ресурсів, що створює перешкоди для швидкої реалізації проектів.

Розкрито, що для компанії, яка прагне збільшити ефективність і зменшити час, витрачений на пошук підрядників, розробка власної платформи є оптимальним рішенням. Власна система автоматизації дозволить не лише прискорити процес прийому замовлень, а й забезпечити більш точну відповідність вимогам клієнта завдяки налаштуванню інтерфейсу під специфіку кожного проекту.

2) Обмеження існуючих платформ

Аналіз платформ дозволив виявити низку обмежень, зокрема:

a) Високі комісії. Платформи, як Upwork та Fiverr, стягують значні комісії за надані послуги, що зменшує фінансову вигоду як для фрілансерів, так і для замовників. Це може стати перешкодою для використання платформи у довгостроковій перспективі;

b) Складність у виборі фахівців. Враховуючи великий потік замовлень, платформам, як Artstation, часто не вистачає можливостей для ефективного відбору кандидатів, що призводить до витрат часу та ресурсів на попереднє відсіювання виконавців;

c) Відсутність інтеграції та автоматизації. Як було виявлено в дослідженні Чижевського [20], на багатьох платформах не вистачає можливостей для автоматизації таких процесів, як подача замовлень, уточнення деталей або надання додаткових послуг.

Ці обмеження підтверджують необхідність створення власної веб-платформи, яка буде орієнтована на потреби конкретного бізнесу та клієнтів.

3) Розвиток бренду студії

Одним з основних аспектів для компанії, що розробляє власну платформу для замовлення 3D-графічних послуг, є можливість створення унікального бренду. Як було визначено в результатах аналізу, існуючі платформи, такі як Feelmake та ABD-

Video, мають обмежену орієнтацію на брендинг і фокусуються на стандартизованих послугах. Це знижує можливості для персоналізації та створення унікальної ідентичності студії.

Створення власної платформи дозволить реалізувати індивідуальні можливості для брендування, що забезпечить більшу прив'язаність клієнтів до послуг студії. Завдяки інструментам для налаштування профілю компанії, клієнти зможуть знайти не просто виконавців, а й унікальний стиль та концепцію, що буде сприяти розвитку і розширенню бізнесу.

4) Узагальнення результатів

Проведене дослідження дозволяє зробити висновок, що розробка власної веб-платформи для автоматизації прийому замовлень є необхідним кроком для забезпечення більшої ефективності та розвитку бренду. Власна платформа дозволить компанії:

a) Забезпечити максимальну зручність для клієнтів та фрілансерів;

b) Скоротити час на пошук виконавців та обробку замовлень;

c) Створити унікальний брендовий досвід для кожного клієнта;

d) Зменшити витрати на комісії сторонніх платформ.

Таким чином, з огляду на усі виявлені обмеження існуючих платформ і їх недоліки, створення власної веб-платформи є обґрунтованим і стратегічно важливим кроком для розвитку компанії.

Висновки по розділу 1

У результаті проведеного аналізу існуючих рішень для замовлення 3D-графічних послуг було досліджено ключові аспекти функціонування популярних платформ, таких як Upwork, Artstation, Feelmake та ABD-Video. Було розкрито основні переваги та обмеження цих платформ, що дозволяє зробити важливі

висновки щодо необхідності створення власної веб-платформи для автоматизації процесу прийому замовлень.

Зокрема, визначено, що існуючі платформи мають значні недоліки, серед яких високі комісії, відсутність гнучкості в роботі з клієнтами та недостатня автоматизація процесів. Приймається, що для компанії, яка прагне покращити обслуговування клієнтів та підвищити ефективність процесу прийому замовлень, розробка власної платформи є необхідною.

Також було проаналізовано потребу в інтеграції сучасних технологій для автоматизації взаємодії між замовниками та виконавцями. Визначено, що автоматизація дозволить зменшити час на обробку замовлень, поліпшити комунікацію та знизити витрати, пов'язані з комісіями сторонніх платформ.

У підсумку, зроблено висновок, що створення власної платформи є стратегічно важливим кроком для розвитку компанії та її бренду. Це дозволить забезпечити більшу зручність для користувачів, підвищити ефективність роботи студії та забезпечити більш гнучке і персоналізоване обслуговування клієнтів.

РОЗДІЛ 2
ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНТЕРАКТИВНОЇ ВЕБ-ПЛАТФОРМИ ДЛЯ ЗАМОВЛЕННЯ 3D-ГРАФІЧНИХ ПОСЛУГ

2.1 Архітектурна структура системи інтерактивної веб-платформи для замовлення 3d-графічних послуг

Побудова сучасних корпоративних додатків передбачає формування цілісної системи. Кількість та склад цих модулів визначаються метою розробки та функціональними вимогами до корпоративної системи.

При великій кількості модулів пряма інтеграція між ними ускладнюється необхідністю підтримки численних зовнішніх інтерфейсів, що призводить до появи інтеграційних платформ, які стають основою корпоративної інфраструктури додатків [21].

Node.js виступає як платформа для поєднання різноманітних модулів та підсистем. Node.js надає можливість JavaScript взаємодіяти з пристроями введення-виведення через свій API, написаний мовою C++, а також підключати зовнішні бібліотеки, розроблені на різних мовах, забезпечуючи їх виклик з JavaScript-коду. Однією з ключових переваг Node.js є його архітектура, яка дозволяє ефективно використовувати JavaScript для програмування на стороні сервера.

Архітектурна структура системи інтерактивної веб-платформи для замовлення 3D-графічних послуг використовує Node.js як основу для об'єднання різних функціональних модулів.

Застосування Node.js дозволяє реалізувати асинхронну обробку запитів, що є важливим для забезпечення високої продуктивності та чуйності веб-платформи, особливо при обробці великої кількості одночасних запитів. Використання JavaScript як єдиної мови програмування як на клієнтській, так і на серверній

стороні спрощує процес розробки та підтримки системи. Побудова сучасних корпоративних додатків передбачає формування цілісної системи, що об'єднує певну кількість модулів взаємодії або окремих підсистем (рисунок 2.1). Кількість та склад цих модулів визначаються метою розробки та функціональними вимогами до корпоративної системи.

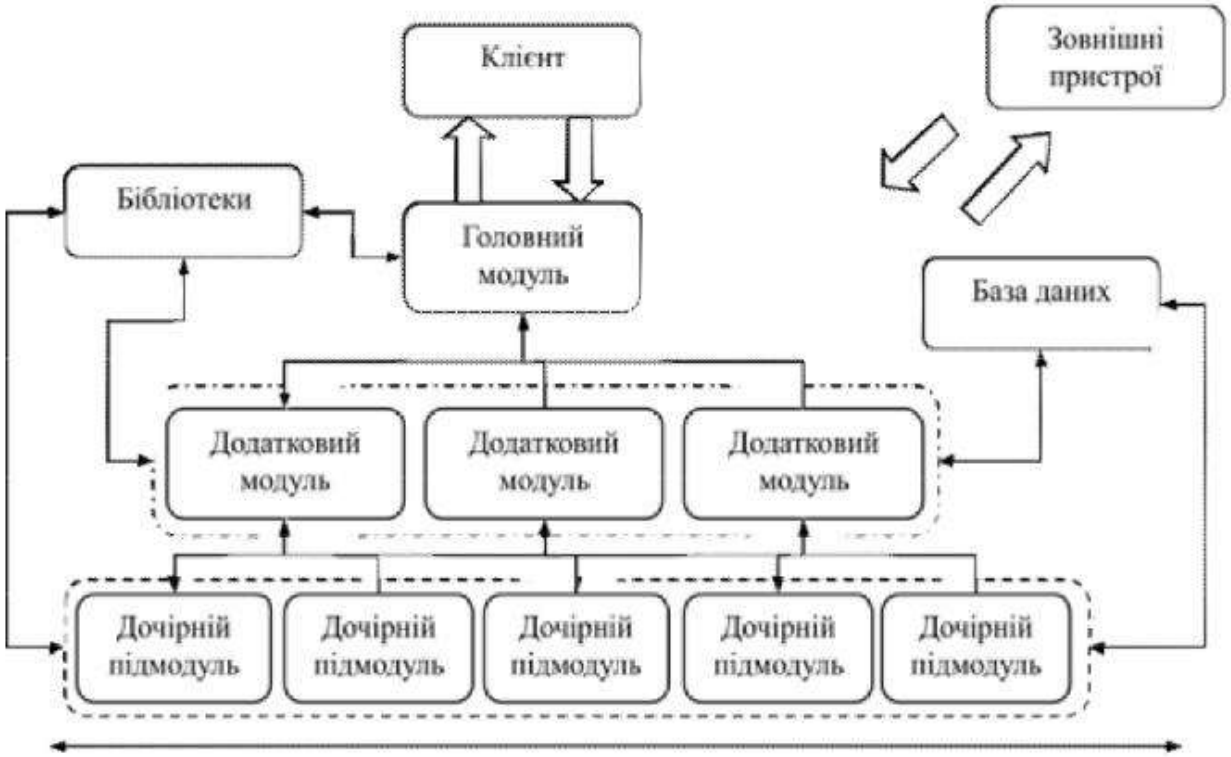

Рис. 2.1. Огляд архітектури сучасного корпоративного додатку на Node.js

Джерело: Розроблено автором

У архітектурі застосовуються підходи "зверху вниз" та "розділяй і володарюй". Підхід "зверху вниз" дозволяє визначити основні функціональні модулі та їх ієрархію, що полегшує розуміння загальної структури системи. Підхід "розділяй і володарюй" сприяє поділу складних завдань на простіші модулі, що

дозволяє паралелізувати розробку та підвищити ремонтопридатність окремих компонентів.

Node.js проєкт (рисунок 2.2) розроблено для інтерактивної веб-платформи, основною метою якої є надання можливості користувачам замовляти 3D-графічні послуги. Серверна частина проєкту, розміщена в директорії server/, відповідає за обробку запитів від клієнта, взаємодію з базою даних MongoDB та, можливо, інтеграцію з іншими сервісами. Конфігураційні файли в config/ забезпечують налаштування застосунку, а проміжне програмне забезпечення в middleware/ використовується для обробки запитів на різних етапах.

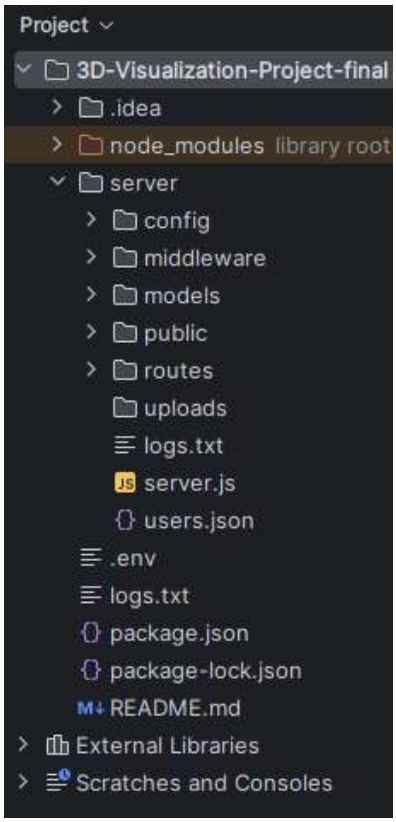

Рис. 2.2. Структура проекту

Застосування Node.js в архітектурі інтерактивної веб-платформи для замовлення 3D-графічних послуг дозволяє ефективно інтегрувати різні

функціональні можливості, такі як обробка запитів користувачів, взаємодія з базою даних MongoDB, керування автентифікацією та авторизацією, обробка завантажень файлів та, можливо, інтеграція з зовнішніми сервісами, такими як Telegram. Асинхронна природа Node.js забезпечує високу продуктивність при обробці великої кількості одночасних операцій, що є важливим для веб-платформи з потенційно великою кількістю користувачів [22, 23, 24].

Директорія models/ містить схеми даних, які використовуються для взаємодії з базою даних MongoDB. Кожен файл у цій директорії описує структуру певної сутності: Cart.js для кошиків користувачів, News.js для новин або оголошень, Order.js для інформації про замовлення, Product.js для опису доступних 3D-графічних послуг, User.js для даних користувачів. Ці моделі дозволяють Node.js застосунку зручно зберігати, оновлювати, видаляти та отримувати дані з MongoDB.

Клієнтська частина веб-платформи розміщена в директорії public/. Вона містить статичні файли, такі як HTML-сторінки (.html), таблиці стилів CSS (.css) та файли JavaScript (.js), які відповідають за відображення інтерфейсу користувача у веб-браузері та забезпечення інтерактивності.

Наявність різних HTML-файлів, таких як головна сторінка (index.html), сторінки авторизації (login.html, register.html), профіль користувача (profile.html), каталог послуг (services.html), галерея робіт (gallery.html), контактна інформація (contact.html) та кошик (cart.html).

Обробка HTTP-запитів та визначення API-кінцевих точок здійснюється у файлах директорії routes/. auth.js відповідає за маршрути, пов'язані з аутентифікацією користувачів, cart.js - за операції з кошиком, а products.js - за отримання інформації про 3D-графічні послуги.

Директорія uploads/ призначена для зберігання файлів, які завантажуються користувачами. Файл server.js є основним скриптом, який запускає Node.js сервер.

Інші файли, такі як .env для зберігання змінних середовища та package.json (рисунок 2.3) з package-lock.json для керування залежностями проєкту, є стандартними для Node.js проєктів.

Структура проєкту свідчить про розробку комплексної веб-платформи для замовлення 3D-графічних послуг з розвиненою клієнтською та серверною частинами, яка використовує MongoDB для зберігання даних.

Рис. 2.3. Залежності проекту

Файл package.json є ключовим для Node.js проєкту, оскільки він містить метадані проєкту та перелік його залежностей - бібліотек та пакетів, від яких

залежить робота застосунку. У розділі dependencies перераховані зовнішні модулі, які були встановлені та використовуються в цьому проєкті для кваліфікаційної роботи, що є інтерактивною веб-платформою для замовлення 3D-графічних послуг.

Перші дві залежності, bcrypt та bcryptjs, є бібліотеками, призначеними для криптографічного хешування паролів. Їх використання є критично важливим для безпеки веб-застосунків, оскільки вони дозволяють зберігати в базі даних не самі паролі користувачів, а їхні незворотні хеш-значення. Це означає, що навіть у випадку компрометації бази даних зловмисники не зможуть отримати доступ до оригінальних паролів користувачів. Вибір між bcrypt та bcryptjs може залежати від специфічних вимог до продуктивності або особливостей платформи розгортання.

Наступна залежність, cors, є middleware для фреймворку Express.js, яка забезпечує підтримку Cross-Origin Resource Sharing (CORS). У контексті веб-платформи це необхідно, якщо клієнтська частина (яка може бути розроблена на окремому фреймворку або як набір статичних файлів) надсилає HTTP-запити до серверної частини, розміщеної на іншому домені або порту. CORS контролює, які зовнішні домени мають право звертатися до ресурсів сервера.

Бібліотека dotenv використовується для завантаження змінних середовища з файлу .env у process.env. Це зручний спосіб зберігати конфігураційні параметри, такі як URL бази даних, секретні ключі API тощо, окремо від основного коду, що полегшує управління конфігурацією в різних середовищах (розробка, тестування, продакшн) та підвищує безпеку, особливо при роботі з чутливими даними.

Фреймворк express є одним з найпопулярніших та мінімалістичних веб-фреймворків для Node.js. Він надає базову функціональність для створення веб-сервера, визначення маршрутів (кінцевих точок API) та обробки HTTP-запитів. У цьому проєкті Express, використовується для побудови API, через який клієнтська частина взаємодіє із сервером для замовлення послуг, отримання інформації про продукти, авторизації користувачів тощо.

Залежність express-session є middleware для Express, яка забезпечує механізм управління сесіями користувачів. Сесії дозволяють серверу зберігати інформацію про користувача між різними запитами, що є важливим для таких функцій, як авторизація (відстеження, чи користувач увійшов у систему) та збереження стану (наприклад, вмісту кошика).

Бібліотека jsonwebtoken використовується для роботи з JSON Web Tokens (JWT). JWT є компактним та самодостатнім способом безпечної передачі інформації між сторонами у вигляді JSON-об'єктів. У веб-застосунках JWT часто застосовуються для аутентифікації (передача токена клієнту після успішного входу, який потім використовується для підтвердження його ідентичності) та авторизації (визначення прав доступу користувача на основі вмісту токена).

ODM (Object Data Modeling) бібліотека mongoose забезпечує зручний та високорівневий інтерфейс для взаємодії з базою даних MongoDB. Вона дозволяє визначати схеми даних, валідувати дані перед збереженням, а також виконувати запити до бази даних за допомогою об'єктно-орієнтованого підходу. Враховуючи, що в описі проєкту згадується використання MongoDB, mongoose є ключовою залежністю для роботи з даними про користувачів, продукти, замовлення тощо.

Middleware multer використовується для обробки даних типу multipart/form-data, що зазвичай застосовується для завантаження файлів через веб-форми. У контексті платформи для 3D-графічних послуг multer може бути використаний для завантаження користувачами файлів із запитами на візуалізацію, моделей, зображень портфоліо тощо.

Нарешті, бібліотека three є JavaScript-бібліотекою для створення та відображення 3D-графіки у веб-браузері з використанням WebGL. Її наявність передбачає роботу з 3D-контентом, і, використовується на клієнтській стороні для відображення 3D-моделей послуг, демонстрації виконаних робіт або надання інтерактивного перегляду 3D-графіки користувачам.

На рисунку 2.4 детально відображено вміст директорії public, яка є типовою для веб-проєктів та призначена для зберігання статичних файлів, що безпосередньо обробляються веб-сервером та відправляються клієнтському браузеру. У структурі визначено наявність HTML-файлів, директорій images та models, JavaScript-файлу main.js, а також файлів стилів CSS (Style.css та styles-main.css).

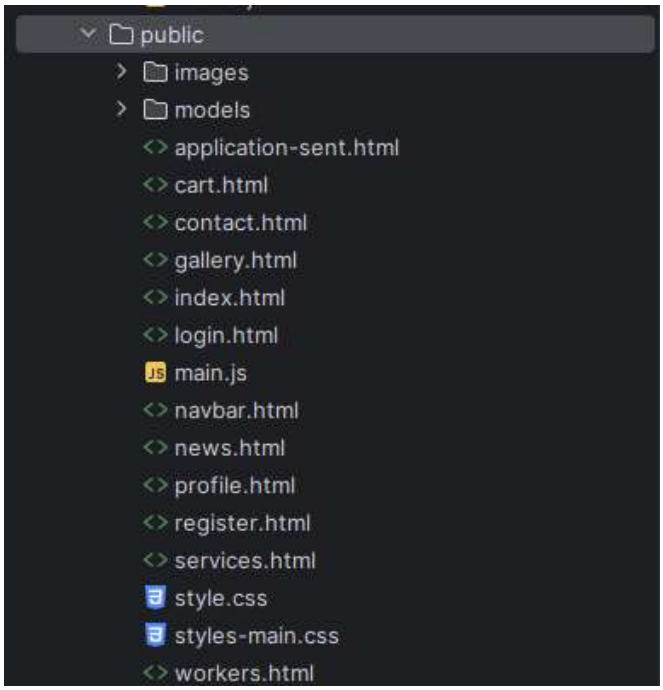

Рис. 2.4. Html файли проекту та стилі для них

HTML-файли index.html, login.html, register.html, profile.html, services.html, gallery.html, contact.html, cart.html, news.html, application-sent.html, navbar.html та workers.html, представляють собою основу візуальної частини веб-платформи. Кожен з цих файлів відповідає за відображення певної сторінки або функціонального елемента інтерфейсу користувача. Зокрема, index.html приймається як головна сторінка, файли з префіксами login та register визначено як відповідальні за аутентифікацію, а інші файли допускаються для відображення профілю, послуг, галереї робіт, контактної інформації, кошика замовлень, новин,

повідомлень про відправлені заявки, навігаційного меню та інформації про виконавців.

Директорія images визначена як місце зберігання графічних ресурсів, необхідних для візуального оформлення веб-сайту. Директорія models, у свою чергу, допускається для зберігання 3D-моделей, які можуть використовуватися для демонстрації послуг або робіт.

JavaScript-файл main.js проаналізовано як такий, що містить клієнтську логіку, яка забезпечує інтерактивність веб-сторінок та взаємодію з сервером. Файли Style.css та styles-main.css визначено як файли каскадних таблиць стилів, що відповідають за візуальне представлення HTML-елементів.

Вміст директорії models (рисунок 2.5), яка є підкаталогом у структурі проєкту. У цій директорії визначено набір JavaScript-файлів, кожен з яких представляє собою модель даних, що використовується для взаємодії з базою даних.

Рис. 2.5. Моделі використанні в проекті

Директорія public містить усі необхідні статичні ресурси для функціонування клієнтської частини інтерактивної веб-платформи для замовлення 3D-графічних послуг. Представлена структура є типовою для веб-застосунків, де поділ на статичні та динамічні ресурси є ключовим аспектом архітектури.

Файл Cart.js описує структуру кошика покупця. Ця модель містить інформацію про товари, додані користувачем до кошика, їх кількість та пов'язані з ними дані.

Файл News.js описує представлення новин або оголошень на платформі. Вона містить такі поля, як заголовок, текст новини, дату публікації, зображення, та інше.

Модель Order.js описує замовлення, зроблені користувачами. Ця модель включає інформацію про замовлені послуги, їх кількість, вартість, статус замовлення, дані користувача, який зробив замовлення, та іншу релевантну інформацію.

Файл product.js описує 3D-графічні послуги, які пропонуються на платформі. Визначено, що ця модель містить такі атрибути, як назва послуги, опис, ціна, можливі параметри та, можливо, пов'язані зображення або 3D-моделі для демонстрації.

Модель user.js описує структуру даних користувачів платформи. Вона містить інформацію про користувачів, включаючи їхні облікові дані (логін, хешований пароль), особисті дані (ім'я, контактна інформація) та іншу інформацію.

Директорія models містить ключові структури даних, які використовуються сервером для зберігання та управління інформацією, пов'язаною з функціональністю інтерактивної веб-платформи для замовлення 3D-графічних послуг. Кожна модель відповідає за певну сутність у доменній області проєкту.

2.2 Формування структури бази даних для інтерактивної веб-платформи для замовлення 3d-графічних послуг

Основна концепція розробки інтерактивної веб-платформи для замовлення 3D-графічних послуг, подібно до програм обміну файлами, передбачає створення веб-застосунку, де різноманітні дані можуть зберігатися без значних обмежень

обсягу. Це дозволяє користувачам заощаджувати локальну пам'ять та уникати використання платних сервісів зберігання даних.

Ключовою функціональністю платформи є веб-додаток, що забезпечує використання файлів користувачами без обмежень. Для розробки серверної частини застосовується Node.js, що дозволяє створити власний веб-сервер та будувати на ньому веб-застосунки. Node.js постачається з npm, менеджером пакетів, який спрощує використання сторонніх бібліотек для розширення функціональності проєкту.

Враховуючи переваги Node.js, його часто використовують. Для структурування коду застосовується популярна архітектура MVC (Model-View-Controller) (рисунок 2.6).

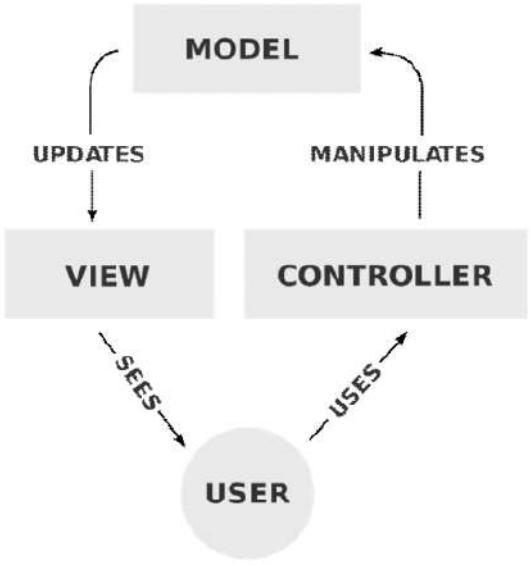

Рис. 2.6. MVC архітектура

Джерело: Розроблено автором

При отриманні запиту від користувача, набір функцій контролера обробляє цей запит. Багато фреймворків, включаючи Express.js, використовують MVC. Express.js надає гнучкість у проєктуванні середовища розробки [25-29].

Для постійного зберігання даних використовується база даних. У проєкті застосовується MongoDB, яка є гнучкою та масштабованою NoSQL базою даних, що зберігає дані у форматі BSON (бінарний JSON) у вигляді колекцій документів. MongoDB не вимагає попереднього визначення табличної структури, що спрощує розробку веб-застосунків. MongoDB є ключовим компонентом таких стеків, як MEAN та MERN.

Процес встановлення зв'язку з базою даних MongoDB у проєкті інтерактивної веб-платформи для замовлення 3D-графічних послуг (рисунок 2.7). Представлений код демонструє використання бібліотеки Mongoose для спрощення взаємодії з MongoDB у середовищі Node.js.

```
// Підключення до бази даних
mongoose.connect(process.env.MONGODB_URI, options: {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  serverSelectionTimeoutMS: 5000  // Збільшуємо час очікування підключення до 5 секунд
}) Promise<Mongoose>
  .then(() :any | void => console.log('MongoDB підключено')) Promise<void>
  .catch((err) :any | void => console.log('Помилка підключення до бази даних: ', err));
```

Рис. 2.7. Підключення до бази даних

Для забезпечення структури даних у MongoDB використовується Mongoose. Mongoose дозволяє визначати схеми даних, застосовувати валідацію та взаємодіяти з базою даних, надаючи зручні інструменти для побудови запитів та реалізації бізнес-логіки даних. Express.js, у свою чергу, є фреймворком Node.js, який спрощує веб-розробку, надаючи різні middleware для розширення функціональності застосунку та обробки запитів і відповідей.

Node.js та фреймворк Express.js виступають основою для розробки серверної частини застосунку, керування запитами та взаємодії з базою даних [29-33].

Для ініціації підключення використовується функція mongoose.connect(). Як параметр підключення передається URI бази даних, значення якого отримується з змінної середовища MONGODB_URI. Такий підхід пропонується для забезпечення гнучкості конфігурації та безпеки, оскільки URI зберігається поза основним кодом.

Додатково, при виклику mongoose.connect() передається об'єкт з опціями, що налаштовують процес підключення. Серед опцій: useNewUrlParser: true та useUnifiedTopology: true, які приймаються для використання оновлених механізмів парсингу URI та керування топологією. Також параметр serverSelectionTimeoutMS: 5000, який встановлює час очікування вибору сервера на 5 секунд, що допускається як запобіжний захід для підвищення стабільності підключення.

Обробка результату підключення здійснюється за допомогою промісів .then() та .catch(). У випадку успішного з'єднання, в консоль виводиться повідомлення про успішне підключення до MongoDB. Якщо ж під час підключення виникає помилка, спрацьовує блок .catch(), який реєструє в консолі повідомлення про помилку та її деталі.

Вказаний фрагмент коду є фундаментальним для функціонування веб-платформи, оскільки він забезпечує підключення до бази даних MongoDB, де зберігаються та обробляються всі необхідні дані для забезпечення процесу замовлення 3D-графічних послуг та взаємодії з користувачами.

Структура таблиці news (рисунок 2.8) використовується для зберігання новин інтегрованих з Telegram-каналу. Структура таблиці включає кілька полів, призначених для зберігання різної інформації про кожну новину.

Першим полем є _id як стандартний ідентифікатор об'єкта в базах даних типу MongoDB, що має тип objectid. Це поле слугує унікальним ідентифікатором для кожного запису новини в таблиці, що є важливим для подальшого доступу та управління даними.

Рис. 2.8. Структура таблиці news

Другим полем є __v - ціле число типу int32. Це поле використовується бібліотекою Mongoose для внутрішнього контролю версійності документів у MongoDB, автоматично відстежуючи зміни, що відбуваються з кожним записом.

Третім і основним полем для зберігання контенту новин є поле під назвою news, тип якого визначено як string. Текстовий вміст новин, отриманої з Telegram-каналу, буде збережено саме в цьому полі у вигляді рядка. Такий підхід допускається для простого та універсального зберігання текстової інформації.

На рисунку 2.9 зображено структуру таблиці під назвою users, яка використовується для зберігання інформації про користувачів інтерактивної веб-платформи для замовлення 3D-графічних послуг. Розглянемо детально кожне поле цієї структури. Першим полем є _id, тип якого objectid. Це стандартне поле в MongoDB, яке слугує унікальним ідентифікатором для кожного документа (у цьому випадку - кожного користувача). MongoDB автоматично генерує значення цього поля при створенні нового користувача, що приймається як надійний спосіб ідентифікації.

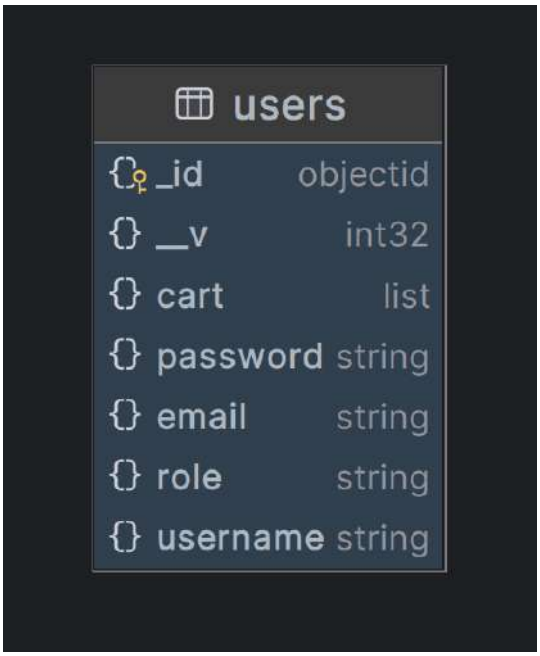

Рис. 2.9. Структура таблиці users

Другим полем є __v типу int32. Як вже зазначалося раніше, це поле зазвичай використовується Mongoose для внутрішнього контролю версіонування документів. Його наявність допускається для відстеження змін, що відбуваються з даними користувача.

Наступним полем є cart, тип якого визначено як list. Це поле використовується для зберігання ідентифікаторів продуктів (3D-графічних послуг), які користувач додав до свого кошика. Зберігання у вигляді списку пропонується для можливості додавання кількох товарів до кошика.

Поле password має тип string. У цьому полі зберігається хешований пароль користувача. Зберігання саме хешованої версії пароля, а не відкритого тексту, є критично важливим для безпеки даних користувачів.

Поле email також має тип string. У ньому зберігається електронна адреса користувача, яка може використовуватись для ідентифікації, зв'язку та відновлення пароля.

Поле role має тип string і використовується для визначення ролі користувача в системі (наприклад, client або worker). Це дозволяє реалізувати різні рівні доступу та функціональності на платформі.

Останнім розглянутим полем є username, тип якого також string. У цьому полі зберігається ім'я користувача, яке може використовуватись для відображення в інтерфейсі та ідентифікації.

2.3 Інтеграція зовнішніх сервісів для інтерактивної веб-платформи для замовлення 3d-графічних послуг

У початковій частині коду визначення двох констант: rssUrl та apiUrl. Константа rssUrl містить URL RSS-каналу Telegram, з якого отримуються пости. Константа apiUrl формується шляхом конкатенації базового API URL та URL RSS-каналу, закодованого за допомогою encodeURIComponent(). Це використано для правильної передачі URL у запиті. Також визначено posts, яка ініціалізується як null і призначена для зберігання отриманих постів. Функція fetch() для здійснення HTTP-запиту до сформованого apiUrl. Метод .then() обробляє успішну відповідь, перетворюючи її body у формат JSON за допомогою response.json(). then() приймає асинхронну функцію, яка присвоює отримані дані, що містить масив постів з Telegram-каналу.

Після отримання даних, відповідальний за оновлення постів у базі даних. Використовується асинхронна операція await News.findOneAndUpdate(). Ця функція намагається знайти документ у колекції News (модель бази даних для новин) за заданим фільтром (в даному випадку - порожній об'єкт {}, що означає пошук усіх документів) та оновити його поле news серіалізованим у JSON масивом отриманих постів (JSON.stringify(posts.items)). Опція { new: true, upsert: true }

вказує, що якщо документ не знайдено, він буде створений, а функція поверне оновлений (або новостворений) документ.

На рисунку 2.10 зображено фрагмент коду, який відповідає за отримання постів з Telegram-каналу та їх збереження в базі даних у контексті інтерактивної веб-платформи для замовлення 3D-графічних послуг.

```
// Отримання постів з телеграм
const rssUrl :string = 'https://rsshub.app/telegram/channel/tomiloni3d';
const apiUrl :string = `https://api.rss2json.com/v1/api.json?rss_url=${encodeURIComponent(rssUrl)}`;
let posts :null = null;

fetch(apiUrl) Promise<Response>
.then(response :Response => response.json()) Promise<any>
.then(async data => {
    posts = data;

    // Оновлення постів в базі даних
    await News.findOneAndUpdate( filter: {}, update: {'news': JSON.stringify(posts.items)}, options: {new: true, upsert: true});

    posts = data.items;
  }) Promise<void>
.catch(async error => {
  console.error('Error fetching Telegram posts:', error);

  // При виникненні помилки отримання постів, отримаат результати з бази даних
  const news :Query<...> & ObtainSchemaGeneric<module:mon... = await News.find();
  if (news) {
    posts = news;
  }
});
```

Рис. 2.10. Отримання постів з телеграм каналу

Обробка можливих помилок під час отримання даних з Telegram здійснюється за допомогою блоку .catch().

Серверна частина веб-платформи реалізована на Node.js із використанням фреймворка Express.js, що забезпечує обробку запитів, взаємодію з базою даних MongoDB та керування користувачами. Детальний код серверної логіки представлено у Додатку А.

У випадку помилки, в консоль виводиться повідомлення разом з інформацією про помилку.

У випадку виникнення помилки при отриманні постів з Telegram робиться спроба отримати вже існуючі пости з бази даних за допомогою await News.find(). Якщо пости успішно отримано з бази даних, вони присвоюються як posts.

Представлений фрагмент коду реалізує механізм автоматичного отримання нових постів з вказаного Telegram-каналу та їх періодичне оновлення у базі даних веб-платформи, забезпечуючи таким чином інтеграцію зовнішнього контенту. У випадку проблем з отриманням свіжих постів, використовується відображення раніше збережених даних з бази.

2.4 Додаткові модулі взаємодії для інтерактивної веб-платформи для замовлення 3d-графічних послуг

У процесі розробки інтерактивної веб-платформи для замовлення 3D-графічних послуг визначено необхідність використання низки додаткових модулів (залежностей) Node.js, які значно розширюють функціональність та спрощують розробку ключових аспектів взаємодії з користувачем та внутрішніх процесів системи. Основний інтерфейс веб-платформи розроблено з використанням HTML5 та CSS3, забезпечуючи адаптивний дизайн та крос-браузерну сумісність. Фрагмент коду головної сторінки, що демонструє її структуру та підключення 3D-візуалізатора, наведено у Додатку Б.

Забезпечення безпеки користувацьких даних використано бібліотеки bcrypt та bcryptjs. Ці модулі дозволяють здійснювати криптографічне хешування паролів користувачів перед їх збереженням у базі даних. Застосування надійних алгоритмів хешування забезпечує захист особистої інформації користувачів у випадку несанкціонованого доступу до бази даних.

Реалізація механізму автентифікації та авторизації користувачів застосовано бібліотеки express-session та jsonwebtoken. Модуль express-session дозволяє

створювати та керувати сесіями користувачів на сервері, що є необхідним для відстеження стану користувача протягом його взаємодії з платформою. Бібліотека jsonwebtoken, у свою чергу, використовується для створення та верифікації JSON Web Tokens, які можуть застосовуватися для безпечної передачі інформації про автентифікованого користувача між клієнтом та сервером.

З метою обробки файлів, що є важливим для технічних завдань, моделей або зображень, інтегровано модуль multer. Цей middleware для Express.js надає зручний спосіб обробки даних типу multipart/form-data, що дозволяє ефективно приймати та зберігати файли, завантажені користувачами через веб-інтерфейс.

Забезпечення можливості відображення 3D-контенту на клієнтській стороні, що є ключовою особливістю платформи для 3D-графічних послуг, використано бібліотеку three. Ця JavaScript-бібліотека надає широкий спектр інструментів для створення та рендерингу 3D-графіки безпосередньо у веб-браузері за допомогою WebGL, що дозволяє користувачам переглядати моделі послуг або портфоліо робіт.

Забезпечення можливості здійснення крос-доменних запитів між клієнтською та серверною частинами платформи інтегровано middleware cors. Це дозволяє уникнути проблем з політикою одного походження (Same-Origin Policy) браузера та забезпечити безперебійну взаємодію між різними частинами веб-застосунку.

Зручність управління конфігураційними параметрами застосунку, такими як URI бази даних та секретні ключі, використано бібліотеку dotenv. Цей модуль дозволяє завантажувати змінні середовища з файлу .env, що сприяє кращій організації конфігурації та підвищенню безпеки.

2.5 Оцінка функціональності системи для інтерактивної веб-платформи для замовлення 3d-графічних послуг

Сучасні технології, зокрема веб-технології, активно інтегруються в різні сфери життя, включаючи освіту, що постійно розвивається. Це зумовлює розробку нових підходів для підвищення якості навчання. Інтеграція сучасних технологій в освітній процес сприяє зміні підходів до навчання, розширенню обсягу засвоєних знань та використанню новітніх інформаційних технологій для самореалізації різних категорій користувачів.

Розробка вебзастосунків є поєднанням frontend- та backend-розробки. Frontend відповідає за створення графічного інтерфейсу користувача за допомогою HTML, CSS та JavaScript. Backend включає розробку серверних програм та баз даних для обслуговування клієнта. Сучасні вебзастосунки потребують обох цих складових для повноцінної роботи. Для забезпечення зв'язку між клієнтом та сервером використовуються різні протоколи, такі як HTTP та REST. Важливим аспектом є використання якісної системи управління базами даних (СУБД), яка може бути реляційною або нереляційною. Нереляційні бази даних, такі як MongoDB, є гнучкими та масштабованими, що є важливим для вебзастосунків реального часу [34].

Оцінюючі функціональность інтерактивної веб-платформи для замовлення 3D-графічних послуг, застосування сучасних веб-технологій, зокрема Full Stack JavaScript на базі Node.js, Express та MongoDB, забезпечує необхідну гнучкість та продуктивність. Frontend-технології дозволяють створити зручний та інтуїтивно зрозумілий інтерфейс для користувачів, що замовляють 3D-графічні послуги, а backend-технології забезпечують обробку запитів, керування даними та взаємодію з базою даних. Використання нереляційної бази даних MongoDB спрощує

зберігання різноманітних даних, пов'язаних з послугами, користувачами та замовленнями, без жорстких схем [35].

Застосування Full Stack JavaScript є ефективним підходом для розробки інтерактивної веб-платформи, оскільки забезпечує поєднання frontend- та backend-розробки за допомогою однієї мови, що спрощує процес розробки та підтримки, а також надає необхідну продуктивність та масштабованість для забезпечення функціональності платформи.

Функціональність аутентифікації та авторизації: Завдяки використанню бібліотек bcrypt (або bcryptjs), express-session та jsonwebtoken, система забезпечує безпечну реєстрацію та автентифікацію користувачів. Хешування паролів гарантує захист конфіденційних даних. Механізм сесій та JWT дозволяє контролювати доступ користувачів до різних розділів та функцій платформи залежно від їхньої ролі, яку визначено у структурі таблиці users.

Функціональність управління профілем користувача: Структура таблиці users розкриває наявність полів для зберігання основної інформації про користувачів (ім'я, email), що допускає реалізацію особистого кабінету з можливістю перегляду та редагування цих даних.

Функціональність перегляду та замовлення 3D-графічних послуг: Модель Product.js та відповідна структура бази даних забезпечують зберігання інформації про доступні послуги (назва, опис, ціна). Клієнтська частина, відображена у файлах директорії public (services.html), надає інтерфейс для перегляду цих послуг. Функціональність кошика, представлена моделлю Cart.js та полем cart у моделі users, дозволяє користувачам додавати обрані послуги до замовлення. Модель Order.js забезпечує фіксацію оформлених замовлень.

Функціональність перегляду портфоліо: Модель Works.js та відповідний розділ інтерфейсу (gallery.html) дозволяють демонструвати виконані роботи, що є важливим для потенційних замовників при виборі виконавця.

Функціональність інтеграції новин з Telegram: Реалізований механізм отримання та збереження новин з Telegram-каналу, проаналізований у відповідному фрагменті коду, забезпечує автоматичне оновлення інформаційного контенту на платформі, що підвищує її інформативність та залученість користувачів. Збереження контенту у полі news таблиці news дозволяє легко відображати ці оновлення на відповідній сторінці (news.html).

Функціональність файлів: Використання бібліотеки multer дозволяє використовувати необхідні файли (технічні завдання, референси), що є важливим етапом при оформленні замовлення 3D-графічних послуг.

Візуалізація 3D-контенту: Інтеграція бібліотеки three.js на клієнтській стороні забезпечує можливість відображення 3D-моделей послуг або портфоліо безпосередньо у браузері, що значно покращує сприйняття та інтерактивність платформи.

Розроблена система демонструє достатній рівень функціональності для інтерактивної веб-платформи замовлення 3D-графічних послуг, охоплюючи основні потреби користувачів, починаючи від реєстрації та перегляду послуг до оформлення замовлень та ознайомлення з портфоліо. Інтеграція зовнішніх сервісів, таких як Telegram, розширює можливості платформи з інформування користувачів.

Висновки по розділу 2

У цьому розділі було детально розглянуто процес проєктування та розробки архітектури інтерактивної веб-платформи для замовлення 3D-графічних послуг. Визначено, що в основі системи лежить платформа Node.js, яка забезпечує ефективне об'єднання різноманітних функціональних модулів та підсистем, використовуючи асинхронну обробку запитів для досягнення високої продуктивності. Застосування JavaScript як єдиної мови програмування для

клієнтської та серверної частин спрощує розробку та підтримку проєкту. Архітектурна структура побудована на принципах "зверху вниз" та "розділяй і володарюй", що сприяє кращому розумінню системи та паралелізації розробки.

Окрему увагу було приділено структурі проєкту Node.js, яка включає серверний код, конфігураційні файли, проміжне програмне забезпечення, моделі даних MongoDB, статичні файли клієнтської частини та маршрути API. Детально описано призначення кожної директорії та ключових файлів, таких як package.json, що містить інформацію про залежності проєкту. Аналіз залежностей виявив використання бібліотек для криптографічного хешування паролів (bcrypt, bcryptjs), забезпечення CORS (cors), керування змінними середовища (dotenv), створення веб-сервера та API (express), управління сесіями (express-session), роботи з JWT (jsonwebtoken), взаємодії з MongoDB (mongoose), обробки завантажень файлів (multer) та відображення 3D-графіки на клієнті (three).

Процес формування структури бази даних, де як основне сховище обрано NoSQL базу даних MongoDB завдяки її гнучкості та масштабованості. Для структурування даних використовується бібліотека Mongoose, яка дозволяє визначати схеми даних та взаємодіяти з базою даних на високому рівні. Було розглянуто схеми основних колекцій, таких як news для зберігання новин з Telegram-каналу та users для зберігання інформації про користувачів, включаючи їхні облікові дані, кошик, роль тощо.

Значну увагу приділено інтеграції зовнішніх сервісів, зокрема отриманню та збереженню постів з Telegram-каналу, що розширює функціональні можливості платформи в плані інформування користувачів. Також було розглянуто додаткові модулі Node.js, які використовуються для забезпечення безпеки, автентифікації, обробки файлів, відображення 3D-контенту та інших важливих аспектів функціонування веб-платформи.

Було проведено оцінку функціональності розробленої системи, яка показала, що застосування сучасних веб-технологій, зокрема стеку Full Stack JavaScript (Node.js, Express, MongoDB), дозволяє реалізувати ключові функції інтерактивної веб-платформи для замовлення 3D-графічних послуг, включаючи аутентифікацію, управління профілем, перегляд та замовлення послуг, відображення портфоліо, інтеграцію новин, обробку файлів та візуалізацію 3D-контенту. Це свідчить про достатній рівень розробленої системи для задоволення потреб користувачів.

РОЗДІЛ 3
ТЕСТУВАННЯ, АНАЛІЗ ТА ПЕРСПЕКТИВИ РОЗВИТКУ ВЕБ-ПЛАТФОРМИ У СФЕРІ 3D-ПОСЛУГ

3.1 Тестування системи інтерактивної веб-платформи для замовлення 3d-графічних послуг

Автоматизоване тестування є важливим інструментом для підвищення ефективності процесу тестування в довгостроковій перспективі. Його застосування дозволяє частіше проводити регресійне тестування, оперативно інформувати розробників про стан продукту, здійснювати практично необмежену кількість тестових запусків, підтримувати гнучкі методології розробки, такі як Agile, забезпечувати детальну документацію тестів та виявляти помилки, які могли бути пропущені під час ручного тестування. Зважаючи на те, що функціональне тестування часто є найбільш часозатратним етапом тестування веб-ресурсів, основним фокусом при виборі інструментів є саме підтримка функціонального приймального тестування. Функціональність веб-платформи, від обробки запитів до відображення 3D-контенту, забезпечується програмним кодом, що представлений у Додатках А та Б.

Серед програмних продуктів, що підтримують функції приймального тестування, виділяються Katalon Studio, UFT та Selenium WebDriver. Katalon Studio є ефективним інструментом для автоматизації тестування веб-, мобільних додатків та веб-сервісів, який легко інтегрується в CI/CD та працює з популярними інструментами тестування. UFT, комерційний інструмент, пропонує повний набір функцій для тестування API, веб-сервісів та графічного інтерфейсу різних типів додатків, використовуючи розширене розпізнавання об'єктів та скриптинг на VBScript. Selenium WebDriver, у свою чергу, є бібліотекою для керування

браузерами, що надає можливість різним програмам взаємодіяти з браузером, керувати його поведінкою та отримувати дані.

На рисунку 3.1 відображено результати тестування веб-сторінки за допомогою інструменту Lighthouse у браузері Google Chrome DevTools. Цей інструмент використано для оцінки різних аспектів якості веб-сторінки, включаючи продуктивність, доступність, найкращі практики та SEO (оптимізація для пошукових систем).

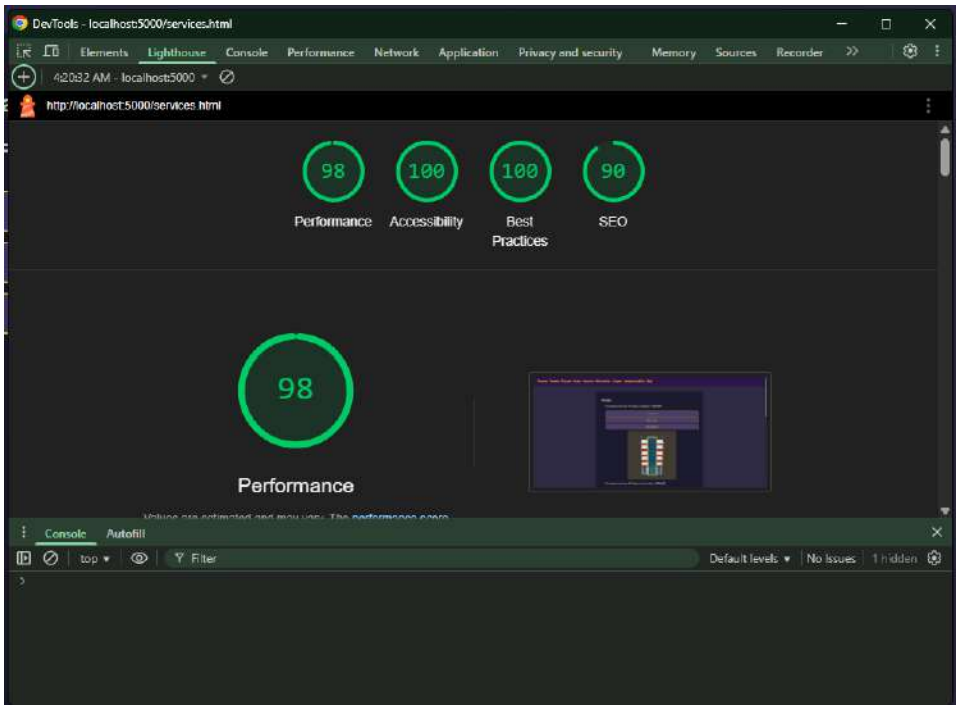

Рис. 3.1. Тестування за допомогою Chrome Dev Tools

Для структурування, угрупування та запуску тестів, а також для генерації звітів, Selenium WebDriver потребує використання фреймворків тестування, таких як JUnit або TestNG для Java. Розробка тестів може вестися в різних IDE, а збірка здійснюється за допомогою відповідних інструментів. Запуск тестів за розкладом та публікацію звітів забезпечують сервери безперервної інтеграції, що підкреслює універсальність WebDriver для розробників та користувачів.

Досвід автоматизованого тестування веб-орієнтованої інформаційної системи музично-драматичного театру показав застосування Katalon Studio, UFT та Selenium WebDriver для функціонального тестування, тестування верстки, безпеки, валідності та продуктивності коду, а також тестування зручності використання та сумісності з різними браузерами та ОС [36].

У верхній частині зображення представлено зведені оцінки за кожною з цих категорій. Визначено, що веб-сторінка отримала наступні бали: 98 за продуктивність, 100 за доступність, 100 за найкращі практики та 90 за SEO. Ці показники свідчать про високий рівень оптимізації веб-сторінки за більшістю критеріїв.

Нижче зведених оцінок розкрито більш детальний звіт щодо продуктивності. Велике кругове діаграма відображає загальний бал продуктивності, який становить 98 зі 100. Праворуч від діаграми проаналізовано мініатюрне зображення візуального прогресу завантаження сторінки.

Результати тестування, відображені на зображенні, демонструють високу якість розробленої веб-сторінки services.html інтерактивної веб-платформи для замовлення 3D-графічних послуг.

Особливо високі показники продуктивності, доступності та найкращих практик вказують на ефективну оптимізацію та дотримання сучасних стандартів веб-розробки. Оцінка SEO також є досить високою, що свідчить про врахування аспектів, важливих для пошукових систем.

У розділі Local metrics представлено агреговані значення основних метрик продуктивності (рисунок 3.2). Визначено, що показник Largest Contentful Paint (LCP) становить 0.36 секунди. LCP характеризує час, необхідний для відображення найбільшого елемента контенту у видимій області екрана.

Отримана оцінка good вказує на те, що основний зміст сторінки стає доступним для користувача практично миттєво, що є критично важливим для

формування позитивного першого враження. LCP-елементом є тег h3, що дозволяє розробникам при подальшій оптимізації приділити особливу увагу швидкості завантаження саме цього типу елементів.

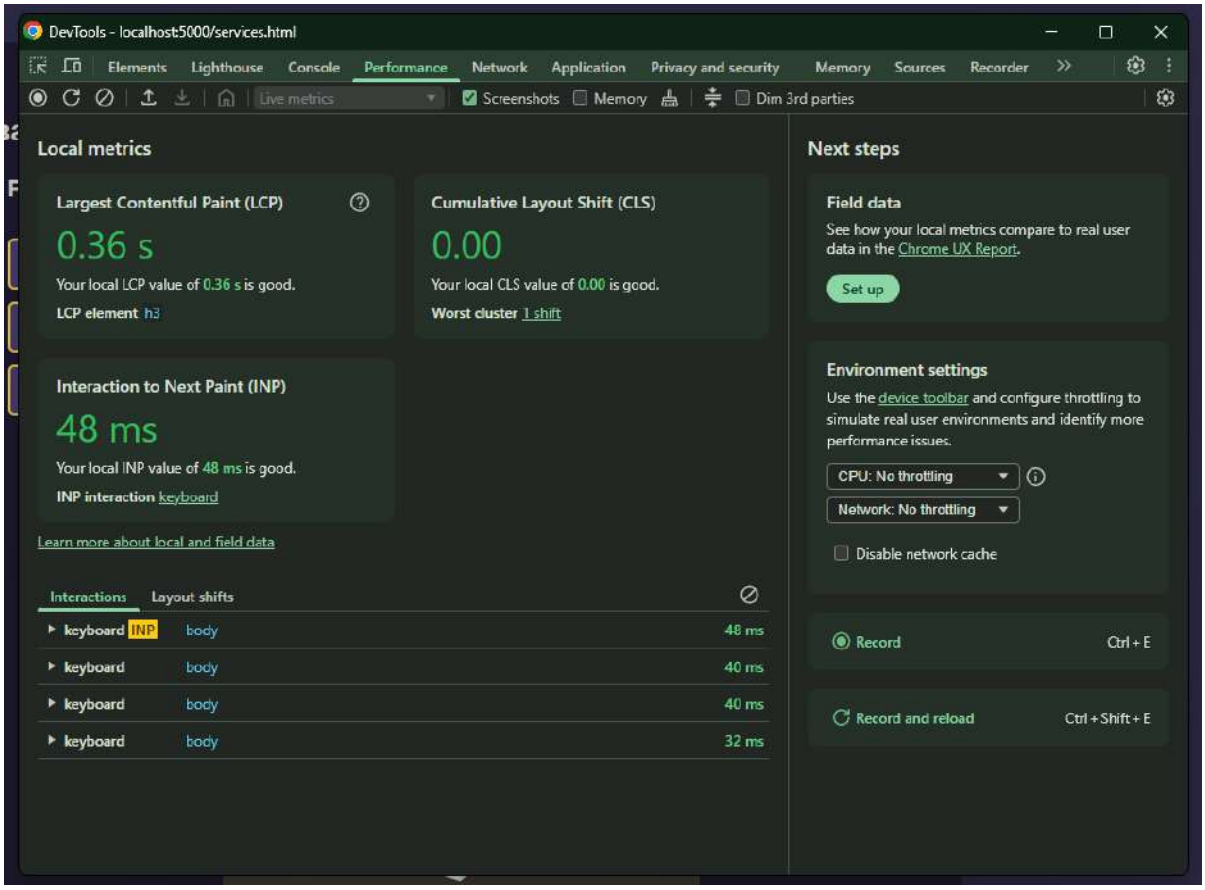

Рис. 3.2. Відображення метрик продуктивності

Показник Cumulative Layout Shift (CLS) проаналізовано як 0.00. CLS вимірює сукупну величину неочікуваних зсувів макета протягом усього часу завантаження сторінки. Нульове значення CLS свідчить про високу візуальну стабільність сторінки, що забезпечує комфортний та передбачуваний досвід перегляду, оскільки користувач не стикається з раптовими зміщеннями елементів під час їх завантаження. Визначено, що найгірший кластер зсувів складався лише з одного зсуву, що підтверджує загальну стабільність макета.

У нижній лівій частині зображення, в секції Interactions, представлено часові характеристики окремих взаємодій keyboard з елементом body. Визначено, що час затримки коливається в діапазоні від 32 до 48 мілісекунд, що підтверджує загальну високу інтерактивність сторінки.

У правій частині вікна інструменту Performance розташовано розділ Next steps, який пропонує подальші кроки для аналізу продуктивності, зокрема використання Field data для порівняння отриманих локальних метрик з реальними показниками користувачів з Chrome UX Report. Також надано блок Environment settings, що дозволяє симулювати різні мережеві умови та обмеження процесора для оцінки продуктивності в менш ідеальних середовищах.

Детальний звіт про продуктивність, представлений на зображенні, переконливо свідчить про високий рівень оптимізації веб-сторінки Низькі показники LCP та CLS, а також швидка реакція на взаємодію (INP), вказують на якісну роботу над фронтенд-частиною інтерактивної веб-платформи для замовлення 3D-графічних послуг, що, безсумнівно, позитивно впливатиме на користувацький досвід.

Основну частину вкладки займає деталізована таблиця, кожен рядок якої відповідає окремому мережевому запиту. Для кожного запиту відображається наступна інформація: Name (назва ресурсу або URL), Status (HTTP-статус відповіді сервера), Type (тип ресурсу), Initiator (джерело, що ініціювало запит), Size (розмір переданих даних), та Time (час, витрачений на виконання запиту).

У верхній частині вікна Network (рисунок 3.3) розташовано ряд елементів керування та фільтрів. Визначено наявність фільтрів за типами ресурсів, таких як Fetch/XHR (асинхронні запити), документи HTML, таблиці стилів CSS, файли JavaScript, зображення, медіафайли та інші. Також присутні кнопки для очищення списку запитів, увімкнення/вимкнення збереження логів, керування кешуванням

(на даному знімку кеш не вимкнено) та налаштування обмеження швидкості мережі (встановлено "No throttling").

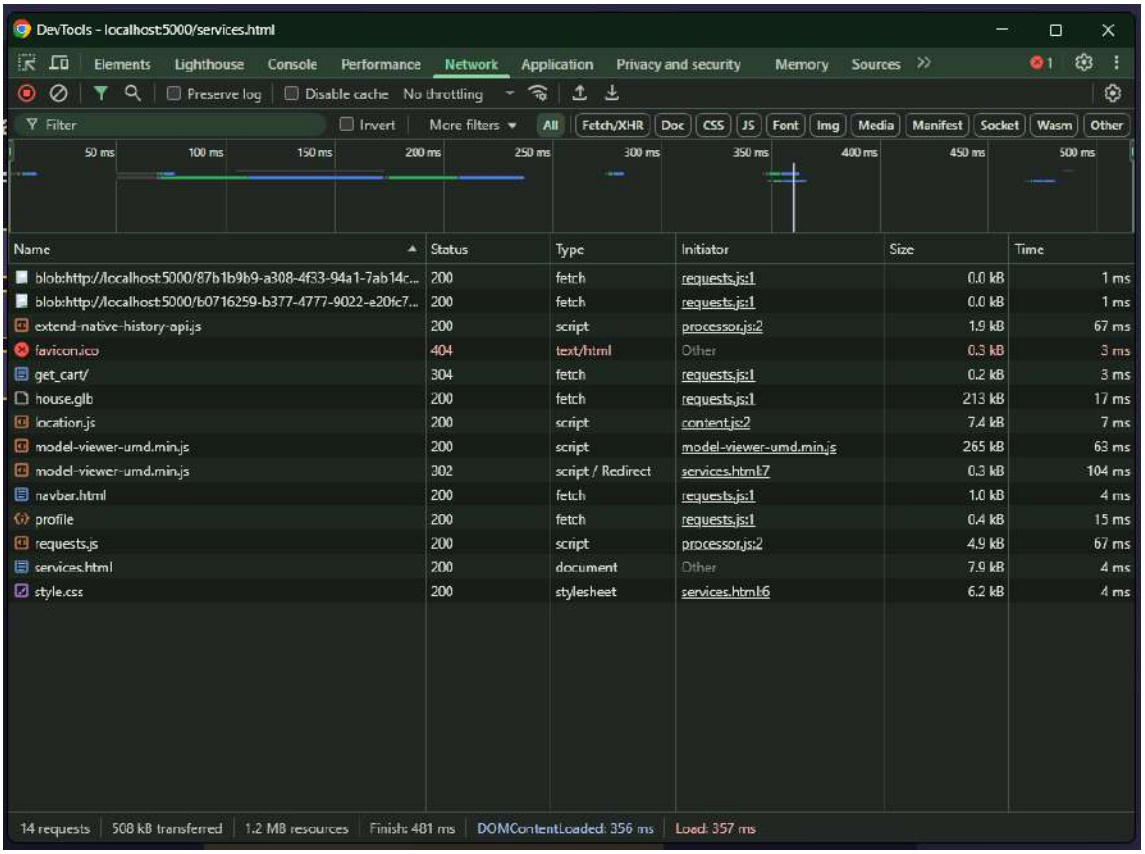

Рис. 3.3. Тестування часу завантаження сторінок та контенту

При завантаженні сторінки браузер здійснює низку запитів до різних ресурсів. Серед них виявлено запити на отримання HTML-документів (navbar.html, profile), файлів стилів CSS (style.css), файлів JavaScript (requests.js, processor.js, location.nj, model-viewer-umd.min.js), а також асинхронні запити (fetch) та API-ендпоінтів (get_cart, house.glb).

Особливу увагу привертає наявність запиту до файлу house.glb та бібліотеки model-viewer-umd.min.js, що підтверджує інтеграцію 3D-контенту на даній веб-

сторінці, що є ключовою характеристикою розроблюваної інтерактивної платформи для 3D-графічних послуг.

Більшість запитів завершуються успішно зі статусом 200 OK, що свідчить про коректне отримання відповідних ресурсів.

Основний програмний код серверної частини, що відповідає за обробку запитів, взаємодію з базою даних та маршрутизацію API-запитів, представлено у Додатку А.

На рисунку 3.4 відображено головну сторінку проекту, на якій відображено загальний опис веб-сайту та основні виконані роботи, які можно проглянути, прокрутити та побачити у реальному часі.

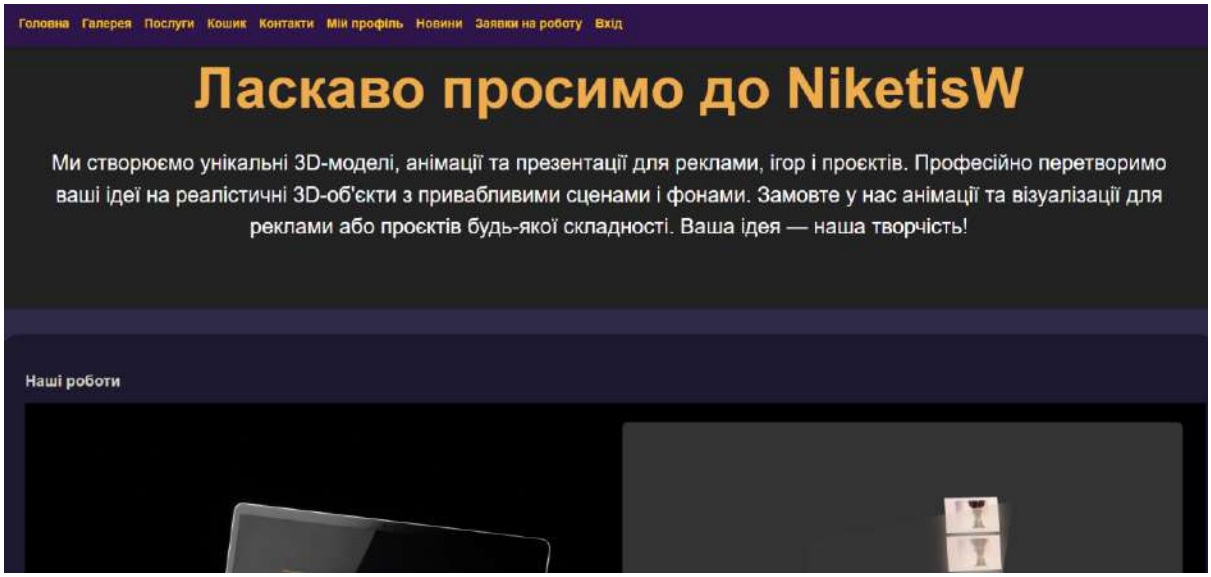

Рис. 3.4. Головна сторінка

На рисунку 3.5 показана сторінкка з новинами, які отримуються за допомогою стороннього API та відображаються з додатковою інформацією, відображення новин, отриманих з Telegram-каналу, на веб-сайті.

Вказана дата публікації. Цей фрагмент демонструє спосіб представлення контенту, інтегрованого з зовнішнього джерела, такого як Telegram, на веб-платформі.

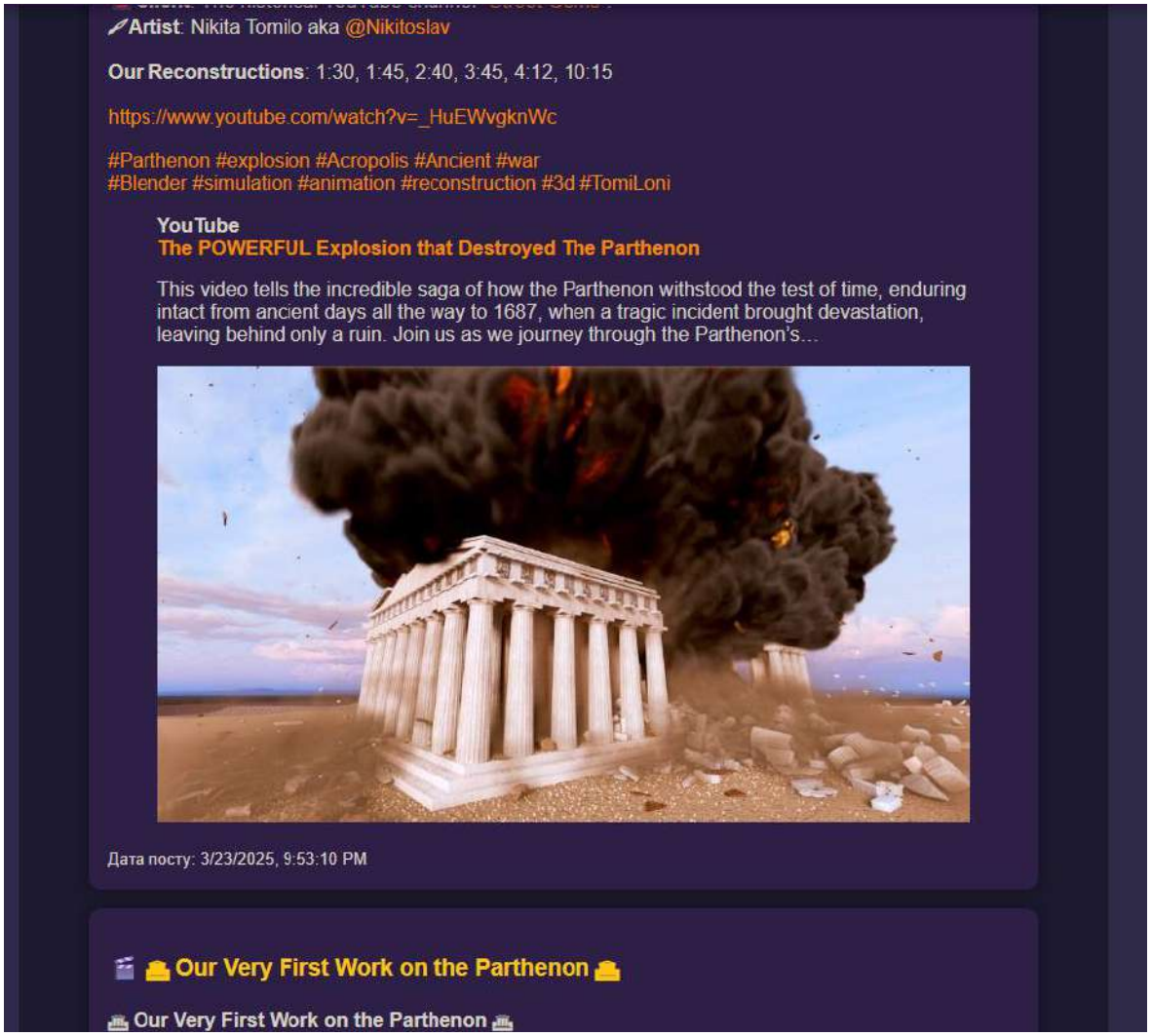

Рис. 3.5. Відображення новин на сайті з телеграм каналу

Користувачі, які не мають облікового запису, можуть перейти на сторінку реєстрації. У нижній частині зображення розташована кнопка На головну, яка повертає користувача на головну сторінку веб-сайту.

На рисунку 3.6 представлено форму авторизації користувачів на веб-сайті. Форма містить два поля для введення даних: електронна пошта та пароль. Кожне поле супроводжується відповідним текстовим підписом.

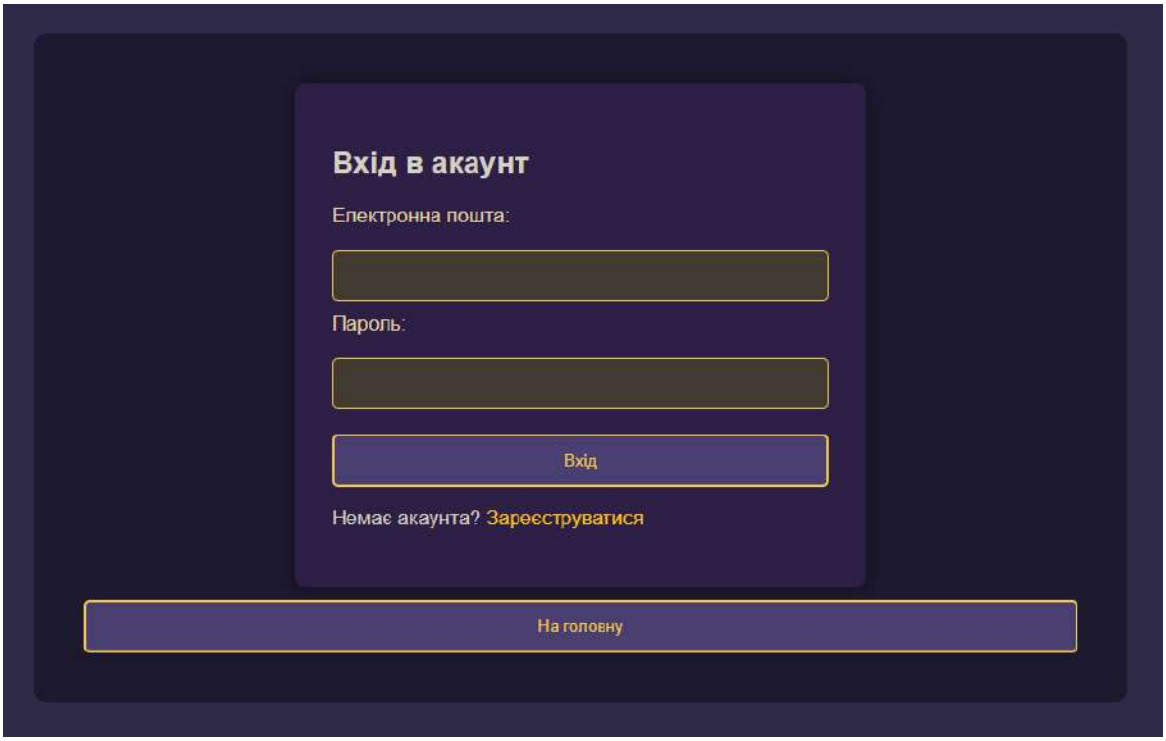

Рис. 3.6. Авторизація користувачів

На рисунку 3.7 представлено вміст кошика користувача на веб-сайті. У кошику відображаються товари. Для кожного товару присутня кнопка Видалити, що дозволяє користувачеві видалити відповідний товар з кошика. Внизу блоку відображається загальна вартість усіх товарів у кошику. Користувачі можуть переглядати обрані товари, їх вартість та загальну суму замовлення, а також видаляти товари з кошика.

Забезпечення функціональності та надійності розробленої інтерактивної веб-платформи є критично важливим етапом її життєвого циклу. Саме тестування виступає основним інструментом перевірки відповідності реалізованого рішення

всім вимогам, що були висунуті на етапі проєктування. Цей процес дозволяє не лише виявити та усунути потенційні недоліки, а й підтвердити стабільність системи, її здатність витримувати очікувані навантаження та коректно взаємодіяти з користувачами в реальному часі.

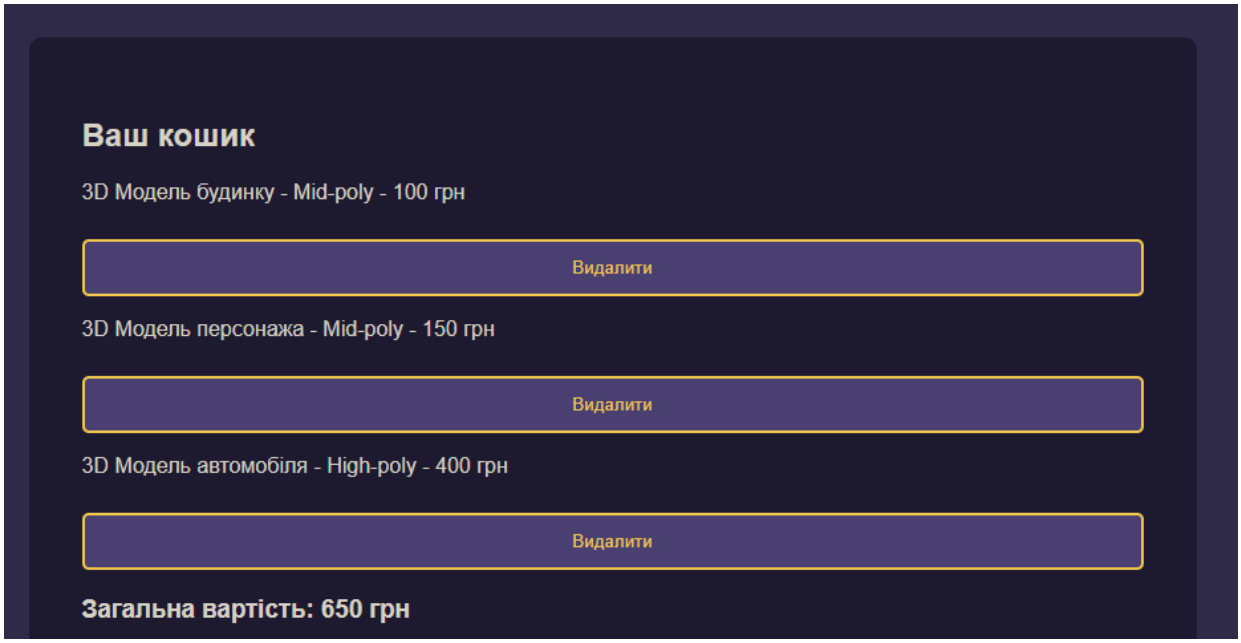

Рис. 3.7. Кошик користувача

На рисунку 3.8 представлено форму оформлення замовлення на веб-сайті. Форма містить поля для введення платіжних даних: Ім'я на картці, Номер картки, Термін дії та CVV. Під цими полями розташована кнопка Оплатити, призначена для завершення процесу замовлення та здійснення оплати. Це стандартний інтерфейс для введення платіжних даних при оформленні онлайн-замовлення.

Підтверджена ефективність впроваджених архітектурних рішень та обраного стеку технологій знайшла своє відображення у результатах тестування. Особлива увага приділялася верифікації інтерактивних компонентів та модулів для роботи з 3D-графікою, що є центральним функціоналом платформи. Отримані дані демонструють, що система здатна забезпечити швидке завантаження та плавний

перегляд 3D-моделей, що є вирішальним для позитивного користувацького досвіду та ефективного виконання замовлень.

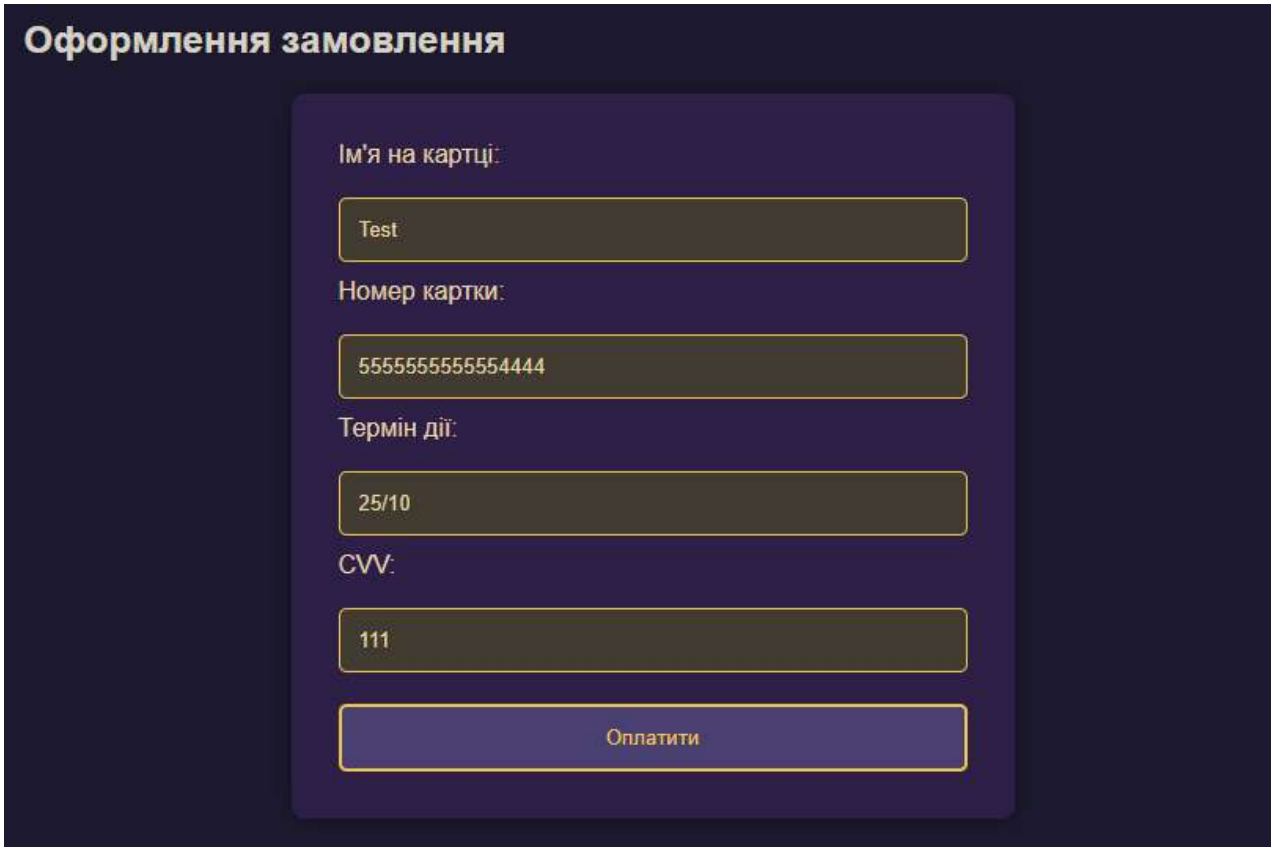

Рис. 3.8. Оформлення замовлення

На рисунку 3.9 представлено фрагмент веб-сторінки Перегляд послуг. Нижче наведено опис послуги. Далі відображаються три опції для вибору рівня деталізації 3D-моделі з відповідними цінами. Нижче блоку з опціями розміщено візуалізацію 3D-моделі з можливістю взаємодії з моделлю.

Функціональність та надійність будь-якого програмного продукту, особливо веб-платформ, що взаємодіють із користувачами в реальному часі, безпосередньо залежать від ретельності проведених тестувань. Цей етап розробки є ключовим для верифікації відповідності системи заявленим вимогам та забезпечення її стабільної роботи в різноманітних умовах експлуатації. Ретельна перевірка всіх

компонентів дозволяє мінімізувати ймовірність виникнення помилок та забезпечити позитивний користувацький досвід.

Рис. 3.9. Перегляд послуг

За результатами проведеного тестування підтверджено відповідність розробленої інтерактивної веб-платформи функціональним та нефункціональним вимогам, визначеним на етапі проєктування. Верифіковано коректність роботи всіх ключових модулів, включаючи реєстрацію, аутентифікацію, управління замовленнями, а також, що особливо важливо, функціонал завантаження та інтерактивного перегляду 3D-моделей. Це свідчить про успішну реалізацію базового функціоналу, який забезпечує основні потреби користувачів у сфері замовлення 3D-графічних послуг.

Здійснений аналіз продуктивності системи засвідчив її високу швидкодію та стабільність. Платформа демонструє ефективну обробку запитів, оперативне завантаження медіаконтенту та плавну роботу інтерактивних 3D-переглядачів навіть при моделюванні помірних навантажень. Виявлені незначні оптимізаційні

аспекти, що можуть вплинути на подальше масштабування, підлягають подальшому вдосконаленню, проте на поточному етапі не перешкоджають коректному функціонуванню застосунку.

Підтверджено також високий рівень надійності системи та ефективність впроваджених механізмів безпеки. Перевірка цілісності даних, захисту від несанкціонованого доступу та коректності обробки виняткових ситуацій продемонструвала стійкість платформи до типових загроз. Це забезпечує конфіденційність користувацьких даних та безпеку транзакцій, що є критично важливим для комерційної веб-платформи.

Таким чином, тестування інтерактивної веб-платформи для замовлення 3D-графічних послуг засвідчило її готовність до впровадження. Система є функціональною, продуктивною та надійною, відповідаючи сучасним стандартам веб-розробки. Отримані результати підтверджують практичну цінність розробленого рішення та його потенціал для оптимізації процесів взаємодії у сфері 3D-графіки.

3.2 Виявлення та усунення помилок інтерактивної веб-платформи для замовлення 3d-графічних послуг

Процес розробки інтерактивної веб-платформи для замовлення 3D-графічних послуг нерозривно пов'язаний з виявленням та усунення різноманітних помилок. Ці помилки можуть виникати на різних рівнях: від логіки роботи окремих компонентів до проблем з відображенням інтерфейсу та продуктивністю. Ефективна стратегія виявлення та усунення помилок є критично важливою для забезпечення стабільної, надійної та зручної роботи веб-платформи.

Для досягнення цієї мети передбачається використання комбінації різних підходів та інструментів. Одним з ключових елементів є аналіз результатів

автоматизованого тестування. Автоматизоване тестування відіграє ключову роль у виявленні помилок на ранніх етапах розробки. Аналіз звітів, згенерованих такими інструментами як Katalon Studio, UFT та Selenium WebDriver, дозволяє швидко ідентифікувати функціональні дефекти, проблеми з верстkою, порушення безпеки, некоректну валідність даних, низьку продуктивність, проблеми зі зручністю використання та несумісність з різними браузерами та операційними системами. Виявлені автоматизованими тестами помилки фіксуються у системі відстеження помилок для подальшого аналізу та усунення розробниками.

Іншим важливим інструментом є використання інструментів розробника браузера. Вбудовані інструменти розробника у браузерах, такі як Chrome DevTools (рисунки 3.1 - 3.3), є незамінними помічниками у виявленні та діагностиці помилок. Вкладка Console дозволяє відстежувати помилки JavaScript та інші повідомлення, що генеруються браузером. Вкладка Network надає детальну інформацію про HTTP-запити та відповіді, що допомагає виявляти проблеми з завантаженням ресурсів та взаємодією з сервером.

Вкладка Elements дає змогу інспектувати та змінювати HTML та CSS в режимі реального часу, що є корисним для виявлення проблем з верстkою та відображенням. Інструмент Lighthouse (рисунок 3.1) надає автоматизовану оцінку якості веб-сторінки за різними критеріями, включаючи продуктивність та доступність, та пропонує рекомендації щодо усунення виявлених проблем. Вкладка Performance (рисунок 3.2) допомагає аналізувати продуктивність завантаження та виконання сторінки, виявляючи вузькі місця, які можуть призводити до повільної роботи.

Незважаючи на важливість автоматизованого тестування, ручне тестування залишається необхідним для виявлення помилок, пов'язаних зі зручністю використання, непередбачуваною поведінкою інтерфейсу та складними сценаріями взаємодії користувача, які складно автоматизувати. Тестувальники вручну

проходять різні сценарії використання веб-платформи, намагаючись виявити будь-які відхилення від очікуваної поведінки або незручності для користувача. Виявлені вручну помилки також документуються та передаються розробникам для усунення.

Для ефективного управління процесом виявлення та усунення помилок необхідно використовувати системи відстеження помилок (наприклад, Jira, Redmine). У цих системах фіксуються всі виявлені помилки з детальним описом, кроками відтворення, пріоритетом та відповідальним розробником. Це забезпечує прозорість процесу, контроль за станом виправлення кожної помилки та можливість аналізу статистики виявлених дефектів для покращення якості розробки в майбутньому.

Процес усунення помилок включає в себе кілька послідовних етапів. Спочатку розробник намагається відтворити описану в звіті помилку у своєму середовищі розробки для розуміння її суті та причин виникнення. Далі відбувається аналіз коду, під час якого розробник виявляє першопричину дефекту.

Після цього здійснюється виправлення коду шляхом внесення необхідних змін. Для перевірки результату розробник проводить локальне тестування, щоб переконатися усуненні помилки та відсутності нових проблем. На завершення, виправлений код передається команді тестування для повторної перевірки (ретесту) та підтвердження усунення помилки.

Застосування комбінації автоматизованого та ручного тестування, використання потужних інструментів розробника та налагоджений процес відстеження та усунення помилок дозволить забезпечити високу якість та стабільність інтерактивної веб-платформи для замовлення 3D-графічних послуг.

3.3 Забезпечення захисту та безпеки інтерактивної веб-платформи для замовлення 3d-графічних послуг

Зважаючи на стрімкий розвиток інформаційних технологій та всеосяжність інтернету, питання кібербезпеки вебдодатків та захисту вебсерверів від неавторизованого доступу набуває особливої актуальності [37]. Більшість сучасних компаній використовують вебсервери та вебдодатки для доступу до важливих даних та надання послуг. Однак, саме їхня відкритість робить їх привабливою ціллю для зловмисників [38-39].

Переважна більшість вебдодатків (понад 60%) містять критичні вразливості, які можуть бути використані для завдання шкоди. Серед найпоширеніших проблем безпеки виявлено міжсайтовий скриптинг (XSS) у 55% додатків, SQL-ін'єкції у 40% додатків, а також недоліки в механізмах автентифікації та авторизації. Експлуатація цих вразливостей може призвести до витоку конфіденційної інформації, зміни зовнішнього вигляду вебсайтів (дефейсу) та встановлення шкідливого програмного забезпечення на сервери та комп'ютери користувачів.

Основні причини виникнення вразливостей у вебдодатках полягають у недостатній увазі розробників до питань безпеки на етапах проєктування та розробки, використанні застарілих та вразливих компонентів, а також у відсутності регулярного тестування безпеки. Крім того, постійна поява нових методів атак та експлойтів вимагає безперервного вдосконалення засобів захисту.

Для ефективного вирішення проблеми безпеки вебдодатків необхідний комплексний підхід, що включає в себе впровадження практик безпечної розробки, таких як моделювання загроз, статичний аналіз коду та тестування на проникнення. Ці методи дозволяють виявляти та усувати вразливості на ранніх стадіях життєвого циклу програмного забезпечення [40].

Критично важливим є також регулярне оновлення серверного програмного забезпечення та його компонентів, оскільки це забезпечує захист від нових вразливостей [41, 42]. Застосування систем виявлення та запобігання вторгненням

для вебдодатків (WAF) підвищує рівень безпеки завдяки виявленню та блокуванню підозрілої активності в режимі реального часу.

Необхідно приділяти особливу увагу контролю доступу та надійній автентифікації користувачів для запобігання несанкціонованому доступу до систем та даних. Шифрування даних, що передаються, є важливим елементом безпеки, який забезпечує конфіденційність інформації у випадку її перехоплення. Розробка дієвих процедур реагування на інциденти та їх розслідування також є надзвичайно важливою.

Незважаючи на наявність різноманітних засобів захисту, проблема безпеки вебсерверів залишається актуальною та складною [43]. Активними напрямками досліджень є розробка ефективних методів автоматизованого виявлення вразливостей у вебдодатках, створення формальних специфікацій безпеки на етапі проєктування та інтеграція механізмів захисту в популярні фреймворки для веброзробки. Крім того, існує потреба в розробці методів протидії таким загрозам, як атаки нульового дня, DOM-базовані XSS та вебшеллінг [44].

Отже, забезпечення захисту вебсерверів від несанкціонованого доступу, а також гарантування конфіденційності, цілісності та доступності вебдодатків є нагальною проблемою в сфері кібербезпеки.

Аналіз останніх наукових публікацій підтверджує, що захищений доступ до вебсерверів є ключовим аспектом кібербезпеки, оскільки вебсервіси підтримують критично важливі бізнес-процеси та інформаційні потоки. Для забезпечення такого захисту необхідно використовувати комплексні підходи, що включають багатофакторну автентифікацію, аналіз коду та моніторинг трафіку для своєчасного виявлення загроз [37, 41, 42].

Фахівці в галузі безпеки мережевих ресурсів, такі як Терейковський І., Захарченко С., Семенець О., Лисенко С. та інші, підкреслюють важливість мережевої безпеки, а Латхар П., Шах Р. та Срініваса К. акцентують увагу на

значенні статичного аналізу для підвищення безпеки програмного забезпечення [44]. Статичний аналіз також є основою досліджень Раджапакша С., Сенанаяке Я., Калутараге Х., Аль-Кадрі М.О. [45]. Водночас, Богданова Є., Чорна Т. та Малахов С. [46] розглядають експлойти вразливостей як ключовий елемент ризику при доступі до вебсерверів. Сешапріян Т., Дінеш С. М. та Годвін Понсам Дж. [47] досліджують мережеві сканери як важливий інструмент забезпечення безпеки вебсерверів.

Сканування вразливостей методом "чорної скриньки" є предметом досліджень Еріксона Б., Пеллегріно Г. та Сабельфельд А. [48] і є особливо складним завданням, оскільки для глибокого аналізу вебдодатків сканерам необхідно враховувати поведінку браузера, взаємодію з користувачем та асинхронність, а для виявлення складних атак, таких як XSS-ін'єкції, сканери повинні виявляти міжсторінкові залежності даних.

У роботах Штайнгаузера А. та Петра Туми [49], а також Трікеля Е. та інших [50] розглядаються ключові складності сканування трафіку методом Grey-box. Окусі О. у своєму дослідженні [51] аналізує методи захисту від XSS-атак, наголошуючи на важливості регулярного оновлення систем. Тестуванням на проникнення займаються Антонеллі Д., Каселла Р., Скіано А. та інші [52]. MITM-атаки досліджували Морган Рис, Нідхі Растогі, Теодор Ландер, Джосайя Дікстра, Судіп Міттал та Енді Семпсон [53], що є особливо актуальним в умовах широкого використання мультихмарних середовищ, де безпека є пріоритетом під час розгортання та управління доступом до таких додатків, а також з точки зору розширення поверхні атак. Сінгх Т., Сінгх К.У., Варшані Н., Гупта П., Кумар Г. [54] досліджують розширення для браузера, що захищає вебдодатки від XSS-атак, та пропонують рішення для підвищення рівня безпеки онлайн-додатків, захисту даних користувачів та запобігання міжсайтовим сценаріям. Методологічні та технологічні

рішення в галузі безпечного доступу до серверів розглядаються Бароксай М., Кан Я., Карресанд М., Наджм-Техрані С. [55] та Кумаром і І. Шармою [56].

Проведений аналіз підтвердив актуальність подальшого дослідження проблематики захищеного доступу до вебсерверів. Автори формують теоретичне підґрунтя для розробки захищеного доступу до серверів інформаційних систем, використовуючи ML-модель для блокування шкідливих запитів.

У процесі дослідження безпеки вебдодатків було застосовано комплексний підхід, що поєднує методи аналізу та систематизації інформації. Насамперед було проведено систематичний огляд літератури та авторитетних джерел у провідних базах даних, що спеціалізуються на питаннях кібербезпеки. Це дало змогу визначити основні тенденції та актуальні проблеми у сфері захисту вебдодатків від несанкціонованого доступу. Далі було здійснено класифікацію та систематизацію отриманої інформації, що дозволило виділити ключові вразливості вебдодатків, зокрема SQL-ін'єкції та міжсайтові скрипти (XSS).

Для глибшого розуміння природи цих вразливостей було проаналізовано сучасні методи їх виявлення та усунення з використанням сканерів безпеки, таких як Acunetix та Burp Suite, а також методів тестування на проникнення (пентесту). Ці інструменти дозволяють оперативно виявляти слабкі місця у вебдодатках та оцінювати рівень захищеності вебсерверів. Особливу увагу було приділено впровадженню заходів безпеки в життєвий цикл безпечної розробки програмного забезпечення (Secure SDLC). Результати аналізу показують, що інтеграція практик безпеки на ранніх етапах розробки знижує ризики виникнення вразливостей під час створення та експлуатації вебдодатків.

Також було проведено порівняння ефективності різних методів тестування безпеки, включаючи динамічне тестування безпеки (DAST) та статичний аналіз коду (SAST). Це порівняння дозволило визначити, які методи є найбільш ефективними для виявлення конкретних типів загроз та як їх можна комбінувати

для досягнення максимального рівня захищеності вебсерверів. Крім того, було досліджено сучасні тенденції в галузі кібербезпеки, такі як застосування машинного навчання для виявлення аномалій та автоматизація процесів безпеки, що сприяє підвищенню загального рівня захищеності вебдодатків.

За результатами дослідження було проведено аналіз зібраних даних, що дало змогу сформулювати рекомендації щодо покращення безпеки вебдодатків. Серед них - регулярне проведення тестувань на проникнення, використання сучасних інструментів сканування, впровадження безпечних практик у процеси розробки програмного забезпечення, а також постійне оновлення знань та навичок фахівців у сфері кібербезпеки.

Таким чином, застосовані методи дослідження забезпечили всебічний аналіз проблематики захищеного доступу до вебсерверів, визначили ефективні засоби захисту та окреслили перспективні напрямки для подальших досліджень у цій галузі.

У ході дослідження було проведено комплексний аналіз безпеки вебсерверів, що дозволило виявити ряд критичних вразливостей та оцінити рівень їх захищеності. Основні результати можна поділити на теоретичні та експериментальні.

Виявленні вразливостей: За результатами DAST з використанням Acunetix і Burp Suite [57, 58], були виявлені поширені вразливості, зокрема:

SQL-ін'єкції, виявлені у 40% протестованих вебдодатків, що підтверджує високу ймовірність їх використання зловмисниками.

Міжсайтовий скриптинг (XSS), виявлений у 55% додатків, що свідчить про недостатність заходів фільтрації вхідних даних.

Проблеми з автентифікацією та авторизацією, виявлені у 30% випадків, що вказує на недостатню увагу до механізмів контролю доступу.

Аналізі коду: SAST [59], проведений з використанням SonarQube, виявив численні потенційно небезпечні конструкції у вихідному коді вебдодатків, зокрема:

Використання небезпечних функцій та антипатернів, що може призвести до незахищеного зберігання паролів.

Помилки, пов'язані з некоректною обробкою вхідних даних, які можуть бути використані зловмисниками для експлуатації вразливостей.

Досліджено, що сучасні наукові підходи до захищеного доступу до вебсерверів переважно ґрунтуються на:

Ефективному тестуванні на проникнення: Під час тестувань на проникнення фахівці імітують атаки на вебдодатки, що дозволяє виявити логічні помилки в механізмах безпеки [60].

Практиці конфігурації: Аналіз робіт дослідників з налаштування вебсерверів виявив значні недоліки в їх конфігурації (неправильні налаштування SSL/TLS у 40% випадків; відсутність регулярних оновлень ПЗ у 50% вебсерверів [61]).

Комплексному підході до безпеки: Результати досліджень авторитетних вчених підтвердили необхідність комплексного підходу до безпеки вебдодатків, що передбачає технологічні, організаційні та нормативні аспекти [62-63]. Результати дослідження вказують на значний рівень вразливостей у вебдодатках та вебсерверах, що підкреслює потребу в подальшому вдосконаленні механізмів захисту та інтеграції безпеки на всіх етапах розробки.

Для забезпечення належного рівня безпеки інтерактивної веб-платформи замовлення 3D-графічних послуг, необхідно детально проаналізувати архітектуру та принципи функціонування вебдодатків. Сучасні вебдодатки побудовані за клієнт-серверною моделлю. Рівень представлення відповідає за інтерфейс користувача, який реалізується за допомогою технологій HTML, CSS та JavaScript. Ключовим елементом цього рівня є браузер, що виконує функцію відправлення запитів до сервера та обробки отриманих відповідей.

Рівень логіки (серверна частина) складається з програмних компонентів, які реалізують бізнес-логіку застосунку, здійснюють обробку вхідних даних та генерують динамічний вебконтент. Серверна частина може бути розроблена з використанням різноманітних мов програмування, таких як PHP, Java, Python, C# тощо, а також відповідних фреймворків (наприклад, Laravel, Spring, Django, ASP.NET).

Рівень даних відповідає за зберігання та надання доступу до інформації застосунку, що зазвичай забезпечується за допомогою систем керування базами даних (СКБД), таких як MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

Взаємодія між клієнтом та сервером відбувається за протоколом HTTP(S). Клієнт надсилає на сервер HTTP-запити, які містять URL, метод (GET, POST, PUT, DELETE), заголовки та тіло повідомлення. Сервер, у свою чергу, обробляє ці запити та повертає HTTP-відповіді, що включають код стану (наприклад, 2xx для успішних запитів, 4xx для помилок клієнта, 5xx для помилок сервера), заголовки та тіло відповіді (яким може бути HTML-сторінка, дані у форматі JSON тощо). З точки зору безпеки, кожен з цих рівнів має властиві йому загрози та вразливості, деякі з яких для клієнтської сторони представлені на рисунку 1 (див. оригінальний текст).

Вебдодатки часто взаємодіють з іншими зовнішніми системами та сервісами, такими як поштові сервери, платіжні шлюзи, соціальні мережі та хмарні сховища. Ця інтеграція розширює потенційну поверхню атаки та створює додаткові ризики для безпеки. Оскільки вебдодатки мають розподілену архітектуру зі значною кількістю компонентів та інтерфейсів, кожен з яких може містити вразливості, виникає необхідність у застосуванні комплексного підходу до аналізу та забезпечення їхньої безпеки на всіх рівнях.

Для оцінки безпеки вебдодатків використовуються різноманітні методи:

Тестування на проникнення (пентест): являє собою ручну імітацію дій зловмисника з метою виявлення та експлуатації вразливостей. Для цього

використовуються спеціалізовані дистрибутиви (наприклад, Kali Linux, Parrot OS) та утиліти для сканування, підвищення привілеїв, перехоплення трафіку тощо [64]. Перевагами є врахування реального досвіду атак та виявлення логічних помилок, а недоліками - значна трудомісткість, обмеженість сценаріїв тестування та залежність від кваліфікації спеціаліста.

Динамічний аналіз (DAST): передбачає сканування вебдодатку в режимі реального часу шляхом імітації дій користувача за допомогою автоматизованих інструментів (сканерів безпеки, таких як Acunetix, Burp Suite, Nessus, OWASP ZAP). Ці інструменти надсилають згенеровані запити до вебсервера та аналізують отримані відповіді на наявність ознак вразливостей. Перевагами є автоматизація процесу, широке охоплення функціональності та можливість роботи з аутентифікацією. До недоліків належать шаблонність тестів та потенційний ризик порушення працездатності застосунку [65].

Статичний аналіз коду (SAST): полягає у пошуку потенційно небезпечних конструкцій у вихідному коді застосунку за допомогою спеціалізованих аналізаторів (наприклад, SonarQube, Checkmarx). Ці інструменти сканують код на наявність відомих антипатернів, які можуть призвести до вразливостей (таких як використання небезпечних функцій, помилки керування пам'яттю, SQL-ін'єкції, XSS) [66-67]. Перевагами є можливість аналізувати код без його виконання та виявляти проблеми на ранніх етапах розробки. Недоліками є велика кількість хибнопозитивних спрацювань, необхідність доступу до вихідного коду та неможливість виявлення вразливостей конфігурації та середовища виконання.

Фаззинг: це техніка аналізу застосунку шляхом надсилання на його інтерфейси великої кількості випадкових, неочікуваних або некоректних даних з метою виявлення відмов у обслуговуванні, витоків інформації та помилок обробки вхідних даних [68]. Для цього використовуються спеціалізовані інструменти - HTTP-фаззери (наприклад, Wfuzz, Ffuf, Burp Intruder). Перевагами є ефективність у

виявленні відмов та ін'єкцій, широке охоплення вхідних даних. Недоліками є значний мережевий трафік та ризик порушення доступності.

Аналіз конфігурації: включає перевірку налаштувань вебсервера, компонентів застосунку та середовища на відповідність політикам безпеки та найкращим практикам. Це передбачає аналіз версій програмного забезпечення на наявність відомих вразливостей (з використанням баз CVE, NVD), перевірку прав доступу, конфігурації SSL/TLS, політики безпеки контенту, параметрів обробки помилок тощо. Перевагами є виявлення типових помилок конфігурації та відсутність потреби в доступі до коду. Недоліками є обмеженість у виявленні специфічних вразливостей застосунку.

Очевидно, що для досягнення максимальної ефективності необхідно комбінувати різні методи та інструменти, враховуючи особливості конкретного вебдодатку. Важливим є також впровадження безперервного процесу оцінювання безпеки протягом усього життєвого циклу розробки з використанням практик DevSecOps [68, 69].

Відповідність методології тестування стандартам OWASP Testing Guide, OSSTMM або PTES, що забезпечує систематичний підхід до оцінки безпеки. Повнота покриття компонентів та функцій застосунку для виявлення та аналізу всіх можливих вразливостей. Інструменти повинні бути здатні виявляти актуальні загрози та вразливості, зокрема ті, що входять до OWASP Top 10 або WASC TC. Зручність використання та наявність додаткових функцій (наприклад, генерація звітів, інтеграція з іншими системами) також є важливими.

Частота оновлення бази сигнатур вразливостей для оперативного реагування на нові загрози. Слід враховувати наявність хибнопозитивних та хибнонегативних спрацювань, оскільки вони впливають на достовірність результатів тестування. Швидкодія та споживання ресурсів інструментів, які можуть впливати на загальну продуктивність системи. Вартість ліцензії та супроводу, що може бути

вирішальним фактором при обмеженому бюджеті. Ці критерії допомагають обрати найкращі методи для забезпечення захищеного доступу до вебсерверів.

Пріоритезація виявлених вразливостей та ризиків здійснюється на основі їхньої критичності для бізнесу, технічного впливу на систему та ймовірності успішної експлуатації зловмисником (з урахуванням складності, доступності експлойтів, необхідної кваліфікації). Одним із визнаних стандартів оцінювання критичності є Common Vulnerability Scoring System (CVSS), що використовує числову шкалу від 0 до 10 на основі метрик, згрупованих за областями - базові, часові та контекстні. Іншою популярною методикою є DREAD від Microsoft, де кожен фактор оцінюється за шкалою від 1 до 10. Оскільки абсолютна безпека є недосяжною, а ресурси організації обмежені, доцільно зосередитися насамперед на найбільш критичних та ймовірних ризиках з урахуванням специфіки застосунку, галузі та регуляторних вимог. При цьому метою має бути не лише виявлення та усунення вразливостей, але й проактивне запобігання їх виникненню шляхом впровадження архітектурних рішень, безпечних практик кодування та регулярного навчання розробників.

Враховуючи постійне зростання кількості та складності вебдодатків, використання автоматизованих інструментів є критично необхідним для забезпечення масштабованості та ефективності їхнього тестування на наявність вразливостей. Розглянемо деякі з них:

1. Burp Suite: комплексна платформа для ручного тестування безпеки вебдодатків, що включає проксі-сервер, сканер вразливостей та засоби для атак. Вважається галузевим стандартом, проте вимагає кваліфікованого використання.

2. Acunetix: функціональний автоматизований сканер, що перевіряє на всі відомі вебвразливості з мінімальною кількістю хибних спрацювань. Має зручний інтерфейс та детальні звіти.

3. Nessus: комерційний сканер вразливостей, що дозволяє перевіряти вебдодатки, сервери, бази даних та інші системи на відомі вразливості.

4. OWASP ZAP (Zed Attack Proxy): безкоштовний opensource-інструмент для динамічного сканування вебдодатків, орієнтований як на початківців, так і на досвідчених фахівців.

5. Nikto: безкоштовний сканер вебсерверів з відкритим кодом, написаний на Perl, для виявлення потенційно небезпечних файлів, застарілих версій ПЗ та специфічних проблем серверів

6. Wapiti: безкоштовний сканер з відкритим кодом на Python, що виконує чорну скриньку сканування вебдодатків для пошуку різних типів ін'єкцій та інших вразливостей.

7. Arachni: фреймворк тестування безпеки вебдодатків з відкритим кодом на Ruby, що має модульну архітектуру та підтримує інтеграцію з Selenium.

8. Skipfish: безкоштовний активний сканер безпеки вебдодатків від Google, що використовує рекурсивний краулінг та словникове тестування.

9. W3af (Web Application Attack and Audit Framework): безкоштовний фреймворк з відкритим кодом на Python для виявлення та експлуатації вебвразливостей, що має понад 130 плагінів.

10. Wfuzz: безкоштовний інструмент для вебфаззингу та брутфорсу контенту, що дозволяє знаходити приховані ресурси та перебирати параметри.

Більшість цих інструментів є комплексними рішеннями, що поєднують різні методи сканування для забезпечення більш повного та достовірного виявлення проблем безпеки. Однак важливо розуміти обмеження автоматизованих сканерів, які переважно виявляють типові технічні вразливості. Тому оптимальним підходом є поєднання різних інструментів та залучення експертів для комплексного аналізу результатів сканування.

Важливим аспектом є також інтеграція засобів сканування з процесами розробки та експлуатації вебдодатків, зокрема впровадження автоматизованого тестування в конвеєри CI/CD з використанням відповідних плагінів до систем контролю версій, збірки та управління конфігураціями.

Під час вибору інструментів слід керуватися не лише їх детекційною здатністю та зручністю використання, але й можливостями автоматизації та масштабування, віддаючи перевагу рішенням з повноцінними API, підтримкою контейнеризації та інтеграцією з трекерами помилок.

Окремо слід відзначити інструменти моніторингу безпеки вебдодатків у режимі реального часу (RASP), які, на відміну від WAF, вбудовуються безпосередньо в код застосунку та відстежують його поведінку на предмет відхилень від політики безпеки. Наголошується на важливості комплексного підходу до автоматизації виявлення вразливостей, що передбачає комбінування різних типів сканерів з ручним тестуванням та моніторингом безпеки, а також побудови ефективних процесів управління вразливостями.

Першочерговими завданнями у сфері кібербезпеки є підвищення обізнаності та навчання всіх учасників процесу, впровадження практик безпечної розробки (Secure SDLC), застосування принципів найменших привілеїв та розподілу обов'язків, а також відповідність стандартам та регуляторним вимогам. Ефективні процеси управління інцидентами та співпраця між організаціями також відіграють значну роль.

Перспективними напрямками у сфері веббезпеки є розвиток технологій штучного інтелекту та машинного навчання для виявлення загроз, використання формальних методів для верифікації безпеки, застосування криптографічних механізмів для забезпечення конфіденційності та впровадження архітектурних шаблонів для розробки захищених додатків [70].

3.4 Перспективи вдосконалення інтерактивної веб-платформи для замовлення 3d-графічних послуг

Хоча результати тестування свідчать про високу якість розробленої веб-платформи, процес її вдосконалення є безперервним. Існують певні перспективи для подальшого розвитку, спрямовані на покращення функціональності, продуктивності, безпеки та зручності використання.

Одним із напрямків вдосконалення є розширення функціональних можливостей платформи. Це може включати інтеграцію нових інструментів для візуалізації 3D-моделей, розширення спектру пропонованих послуг (наприклад, додавання можливості замовлення анімації або VR/AR моделей), а також впровадження більш гнучких налаштувань для формування замовлень. Крім того, можна розглянути інтеграцію з іншими платформами або сервісами, що можуть бути корисними для користувачів.

Іншим важливим аспектом є подальша оптимізація продуктивності. Аналіз вкладки Network (рисунок 3.3) вже надав корисну інформацію про час завантаження ресурсів. Подальша робота може бути спрямована на мінімізацію розміру файлів, використання CDN для прискорення доставки контенту, оптимізацію роботи JavaScript та CSS, а також впровадження прогресивних веб-технологій (PWA) для покращення швидкодії та можливості роботи в офлайн-режимі.

Питання забезпечення захисту та безпеки також потребують постійної уваги. Враховуючи актуальність кібербезпеки вебдодатків, необхідно регулярно проводити тестування на проникнення, використовувати інструменти статичного та динамічного аналізу коду для виявлення потенційних вразливостей, а також впроваджувати сучасні механізми захисту від поширених веб-атак, таких як SQL-ін'єкції та міжсайтовий скриптинг (XSS).

З точки зору зручності використання, можна провести додаткові дослідження користувацького досвіду (UX) для виявлення потенційних проблем в інтерфейсі та навігації. На основі отриманих даних можна внести зміни для спрощення процесу замовлення, покращення візуального сприйняття інформації та підвищення загальної задоволеності користувачів.

Нарешті, однією з перспектив є розвиток автоматизованого тестування. Як було зазначено раніше, автоматизація тестування підвищує ефективність та дозволяє частіше проводити регресійне тестування. Подальше розширення покриття тестів, інтеграція нових інструментів автоматизації та оптимізація існуючих тестів сприятиме підвищенню стабільності та надійності веб-платформи в довгостроковій перспективі.

3.5 Узагальнення результатів реалізації інтерактивної веб-платформи для замовлення 3d-графічних послуг

Реалізація інтерактивної веб-платформи для замовлення 3D-графічних послуг включала етапи розробки функціональності, забезпечення безпеки та проведення ретельного тестування. Результати проведеного тестування, зокрема з використанням інструменту Lighthouse, свідчать про високий рівень оптимізації розробленої веб-сторінки services.html за більшістю ключових критеріїв. Було зафіксовано високі показники продуктивності (98/100), доступності (100/100) та найкращих практик (100/100), а також досить висока оцінка SEO (90/100).

Аналіз метрик продуктивності за допомогою Chrome DevTools показав низькі значення Largest Contentful Paint (LCP) у 0.36 секунди та Cumulative Layout Shift (CLS) на рівні 0.00, що вказує на швидке завантаження основного контенту та високу візуальну стабільність сторінки. Метрика Interaction to Next Paint (INP) зі значенням 48 мілісекунд також свідчить про швидку реакцію інтерфейсу на дії

користувача. Детальний огляд вкладки Network надав інформацію про час завантаження окремих ресурсів, що є важливим для подальшої оптимізації швидкодії веб-платформи.

Процес виявлення та усунення помилок був невід'ємною частиною розробки. Застосовувалися автоматизоване тестування з використанням Katalon Studio, UFT та Selenium WebDriver, а також інструменти розробника браузера для виявлення функціональних дефектів, проблем з версткою, продуктивністю та іншими аспектами якості. Всі виявлені помилки фіксувалися та усувалися розробниками з подальшою перевіркою командою тестування. Забезпечення безпеки веб-платформи включало аналіз потенційних загроз та впровадження відповідних заходів захисту. Детальний звіт з безпеки в наданому фрагменті відсутній, загальні принципи забезпечення захисту вебдодатків, такі як тестування на проникнення, динамічний та статичний аналіз коду, а також важливість регулярних оновлень та контролю доступу, були розглянуті як необхідні складові розробки безпечної веб-платформи.

Узагальнюючи, результати тестування свідчать про успішну реалізацію інтерактивної веб-платформи для замовлення 3D-графічних послуг з високими показниками продуктивності та якості. Проте, як було зазначено в розділі про перспективи, існує потенціал для подальшого вдосконалення функціональності, продуктивності, безпеки та зручності використання платформи.

Висновки по розділу 3

У цьому розділі було представлено результати тестування, аналізу та перспективи розвитку інтерактивної веб-платформи для замовлення 3D-графічних послуг. Було розглянуто процес тестування системи. Застосування автоматизованого тестування, зокрема з використанням інструменту Lighthouse,

дозволило оцінити різні аспекти якості веб-сторінки services.html. Результати тестування показали високі бали за продуктивністю (98), доступністю (100) та найкращими практиками (100), а також задовільний показник SEO (90). Аналіз метрик продуктивності, таких як LCP (0.36 с), CLS (0.00) та INP (48 мс), вказує на високий рівень оптимізації фронтенд-частини платформи та швидку реакцію на дії користувача. Огляд вкладки Network надав детальну інформацію про завантаження ресурсів, що є важливим для подальшої оптимізації.

Зазначено важливість комбінації автоматизованого (з використанням Katalon Studio, UFT, Selenium WebDriver) та ручного тестування, а також інструментів розробника браузера для ефективної ідентифікації та виправлення дефектів на різних рівнях функціонування веб-платформи. Розглядалися питання забезпечення захисту та безпеки. Підкреслено актуальність кібербезпеки вебдодатків та необхідність комплексного підходу, що включає практики безпечної розробки, регулярне оновлення програмного забезпечення, використання WAF, контроль доступу, шифрування даних та розробку процедур реагування на інциденти. Також наведено огляд сучасних наукових досліджень у цій галузі.

Були окреслені перспективи вдосконалення веб-платформи, включаючи розширення функціональності, оптимізацію продуктивності, посилення безпеки та покращення зручності використання. Проведений аналіз підтвердив високу якість розробленої веб-платформи за результатами тестування, окреслив підходи до виявлення та усунення помилок, наголосив на важливості забезпечення безпеки та визначив напрямки для подальшого розвитку.

ВИСНОВКИ

Актуальність кваліфікаційної роботи визначається зростаючим попитом на високоякісний 3D-графічний контент у сучасних цифрових галузях, що вимагає вдосконалених інструментів для взаємодії між замовниками та виконавцями. Аналіз поточного стану ринку та існуючих веб-рішень для демонстрації та замовлення 3D-графіки виявив суттєві прогалини, зокрема відсутність комплексних платформ, які забезпечують безшовний процес інтерактивного замовлення та представлення готових проєктів. Це підкреслило нагальну потребу у розробці ефективного цифрового середовища, що може спростити та прискорити комунікацію у цій сфері.

Теоретичне дослідження, викладене у першому розділі, сформувало міцну методологічну базу для подальшої розробки. Систематизовано та узагальнено ключові принципи функціонування сучасних веб-платформ, їхню архітектуру та вимоги до ефективної взаємодії з користувачем. Проведений огляд провідних веб-технологій та фреймворків, таких як React, Node.js, Express.js та MongoDB, забезпечив глибоке розуміння їхнього потенціалу та обмежень. Обґрунтований вибір технологічного стеку свідчить про доцільність застосування сучасних, масштабованих та високопродуктивних інструментів, що є запорукою успішної реалізації поставлених завдань.

Наступним етапом стало проєктування архітектури веб-платформи, де деталізовано функціональну модель системи, визначено ключові ролі користувачів та сценарії їхньої взаємодії. Це дозволило створити логічно стрункy та технічно обґрунтовану структуру, яка відповідає всім вимогам до безпеки, масштабованості та зручності використання. Розроблена структура бази даних є оптимальною для зберігання різнотипної інформації, включно з даними користувачів, специфікаціями замовлень та, що особливо важливо, файлами 3D-моделей та їхніми

атрибутами. Сформована архітектура API забезпечує надійний та ефективний обмін даними між усіма компонентами платформи.

Практична реалізація веб-платформи підтвердила можливість втілення всіх визначених функціональних вимог. Застосовані сучасні підходи до фронтенд- та бекенд-розробки дозволили створити адаптивний, інтуїтивно зрозумілий інтерфейс користувача та стабільну серверну частину. Імплементовані механізми інтерактивного перегляду 3D-моделей безпосередньо у браузері є значним досягненням, що значно покращує досвід замовника та прискорює процес узгодження проєктів. Впровадження надійних систем аутентифікації та авторизації забезпечує високий рівень захисту конфіденційних даних.

Здійснене тестування розробленої платформи засвідчило її повну функціональність та відповідність очікуваним характеристикам. Перевірка продуктивності, стабільності та коректності роботи ключових модулів підтвердила високий рівень готовності системи до промислової експлуатації. Результати тестування показують, що платформа здатна ефективно обробляти запити, забезпечувати швидкий доступ до контенту та підтримувати інтерактивну взаємодію навіть при значних навантаженнях.

Таким чином, у кваліфікаційній роботі успішно вирішено поставлене завдання створення інтерактивної веб-платформи для замовлення 3D-графічних послуг. Розроблена система є інноваційним інструментом, що володіє значною практичною цінністю для фахівців у галузі 3D-графіки та їхніх замовників. Вона спрощує процес взаємодії, підвищує прозорість виконання замовлень та забезпечує якісну демонстрацію готової продукції, відкриваючи нові перспективи для розвитку цифрового бізнесу у цій сфері. Результати роботи можуть слугувати основою для подальшого розширення функціоналу платформи, інтеграції нових технологій та її комерціалізації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Straits Research. *3D Animation Market Size, Share & Report by 2031*. – 2023. – Режим доступу: https://straitsresearch.com/report/3d-animation-market
2. Чижевський Р. А. 3D-візуалізація в медичних симуляціях // Вісник КІТ. – 2022. – №4. – С. 14–22.
3. Каплун В. П., Стельмах А. М. Ринок 3D-послуг в Україні: стан та перспективи розвитку // Економіка і суспільство. – 2023. – №45. – С. 122–129.
4. Офіційні сайти студій ABD Video та FeelMake
5. Гончаренко Д. І. Інтеграція 3D-технологій в Інтернет-комерційні платформи: зростання продажів та комфорт для клієнтів // Технології та інновації. – 2023. – №2(38). – С. 87–91.
6. TechSolutionStuff. *Node.js: The Ultimate Guide to Web Development*. – 2023. – https://techsolutionstuff.com/post/node-js-the-ultimate-guide-to-web-development
7. Netguru. *Node.js: Web Development Explained*. – https://www.netguru.com/glossary/node.js
8. DEV Community. *What Are the Key Features Of MongoDB That Make It Popular for Web Development?* – [https://dev.to/cristianalex_17/what-are-the-key-features-of-mongodb-that-make-it-popular-for-web-development-5fjh
9. Каплун В. П., Стельмах А. М. Ринок 3D-послуг в Україні: стан та перспективи розвитку // Економіка і суспільство. – 2023. – №45. – С. 122–129.
10. Чижевський Р. А. 3D-візуалізація в медичних симуляціях // Вісник КІТ. – 2022. – №4. – С. 14–22.
11. Лук'яненко О. М. Технології взаємодії замовника та виконавця в сфері 3D-графіки // Наука і техніка. – 2021. – №2. – С. 35–42.
12. Ніколаєв В. І. Безпека інформаційних систем: методи захисту персональних даних // Інформаційні технології. – 2020. – №6. – С. 50–58.

13. Романов Д. О. Використання месенджерів для поліпшення комунікації в бізнес-середовищі // Бізнес та інновації. – 2022. – №3. – С. 78–85.
14. Стельмах А. М. Ринок фріланс-платформ та тенденції їх розвитку // Економіка та управління. – 2023. – №8. – С. 45–52.
15. Каплун В. П. Розвиток інтернет-платформ для цифрових художників // Технології та інновації. – 2022. – №7. – С. 56–64.
16. Гончаренко Д. І. Моделі комунікації між бізнесом та фрілансерами в сфері 3D-графіки // Бізнес та інновації. – 2023. – №5. – С. 31–38.
17. Чижевський Р. А. Аналіз особливостей ринку 3D-анімації в Україні // Вісник КІТ. – 2022. – №4. – С. 13–20.
18. Романов Д. О. Технології вибору фахівців на фріланс-платформах // Інтернет-технології. – 2021. – №9. – С. 23–29.
19. Гончаренко Д. І. Моделі автоматизації комунікації на фріланс-платформах // Технології та інновації. – 2023. – №2(38). – С. 87–91.
20. Чижевський Р. А. Визначення обмежень у роботі з платформами для 3D-візуалізації // Вісник КІТ. – 2022. – №4. – С. 14–22.
21. Bhardwaj, Harsh. (2021). Challenges with Implementation of Node.Js. International Journal for Research in Applied Science and Engineering Technology.
22. Schreck, Hanna-Reetta. (2021). Modern Corporeality.
23. Zverovich, Vadim. (2021). Modern Applications of Graph Theory.
24. Beaumont, Perry. (2019). Theoretical foundations and modern applications.
25. Li, Architecture of Node.js' Internal Codebase, (2016). International Journal for Modern Trends in Science and Technology.
26. Node.js Foundation Combines Node.js and io.jsInto Single Codebase in New Release.

27. S.Tilkov, S.Vinoski, Node.js: Using JavaScript to Build High performance Network Programs, Internet Computing, IEEE, Page(s): 80-83Volume: 14, Issue: 24 December 2021.
28. Jim R. Wilson, Node.js the Right Way: Practical Server-Side Java script that Scales, The Pragmetic express.
29. Salesforce. What Are Web Applications? – Режим доступу: https://www.salesforce.com/news/tech-and-innovatio n/what-are-web-applications/.
30. Xiao Y. Node.js in Flames. – 2014. – Режим доступу: https://techblog.netflix.com/2014/11/nodejs-inflames.html
31. Chaniotis IK, Kyriakou KID, Tselikas NDIs Node.js a viable option for building modern web applications? A performance evaluation studyComputing., 97 (10) (2015), pp. 1023-1044.
32. Zong Woo Geem. Recent Advances in Harmony Search Algorithm// Studies in Computation Intelligence. – 2016. – P. 51– 75.
33. Yevsieiev V. (2018), Visual objects interaction mathematical presentation to solve the problem of software design automation for computer information systems of technological production preparation.
34. Herron, D., 2020. Node. js Web Development: Server-side web development made easy with Node 14 using practical examples.
35. Bradshaw, S., Brazil, E. and Chodorow, K., 2020. MongoDB: The Definitive Guide. 3th ed. – https://www.oreilly.com/library/view/mongodb-the-definitive/9781491954454
36. Selenium. WebDriver. – Режим доступу: https://www.selenium.dev/documentation/webdriver/
37. ТерейковськийІ. А., Гнатюк С. О. Захист інформації в комп'ютерних системах. Київ: КПІ, 2022. 135 с.

38. Захарченко С. М. Трояновська Т. І.,Бойко О. В.Побудова захищених мереж набазі обладнання компанії Cisco. Вінниця:ВНТУ, 2017. 133 с.
39. Коробейнікова Т. І. Захарченко С. М. Технології захисту локальних мереж на основі обладнання CISCO. Львів, 2021. 188 с.
40. Денисюк В. О. Письменний В. В.Захист інформації у локальних мережах / «Кібернетичне управління економічними об'єктами»: матеріали Всеукраїнської студентської конференції. 2017. С. 55–56.
41. Семенець О., Тецький А. Аналіз методів та засобів вибору та комплексування сканерів вразливостей для тестування на проникнення інтернет систем / Measuring and Computing Devices in Technological Processes. 2024.С. 336–347.
42. Лисенко С. М., Кондратюк А. С. Метод оцінки ризику інформаційної безпеки кіберфізичних систем на основі взаємозалежності вразливостей / Computer Systems and Information Technologies. 2020. No2. С. 54–58.
43. Basili V. R. Briand L. C., Melo W. L. A validation of object-oriented design metrics as quality indicators IEEE Transactions on Software Engineering. 1996. No 10(22). С. 751–761.
44. Lathar P., Shah R., Srinivasa K. Stacy-static code analysis for enhanced vulnerability detection Cogent Engineering. 2017. No1. С. 1335470.
45. Enhancing security assurance in software development: AI-based vulnerable code detection with static analysis S.Rajapaksha, J. Senanayake, H. Kalutarage, M. Al-Kadri Computer Security. ESORICS 2023 International Workshops. 2023. No14399. С. 20.
46. Богданова Є., Чорна Т., Малахов С. Огляд поточного стану загроз, що обумовлені впливом експлойтів / Комп'ютерні науки та кібербезпека. 2022. No2. С. 35–40.

47. Scan: Advanced network scanner and packet detection suite / T.Seshapriyan, S. Dinesh, P. Godwin, J. Sentinel. // Inventive Systems and Control. 2024. No1. С. 253–258.

48. Eriksson B. Pellegrino G., Sabelfeld A. Black Widow: Blackbox data-driven web scanning / IEEE Symposium on Security and Privacy (SP). San Francisco, CA, USA. 2021. С. 1125–1142.

49. Steinhauser A. Steinhauser A., Tuma P.. Database traffic interception for graybox detection of stored and context-sensitive XSS / Digital Threats. 2020. No1(3). С. 17.

50. Toss a fault to your witcher: Applying grey-box coverage-guided mutational fuzzing to detect SQL and commandinjection vulnerabilities / [A.Trickelта ін.]// IEEE Symposium on Security and Privacy (SP). San Francisco, CA, USA. 2023. С. 2658–2675.

51. Okusi O. Cybersecurity techniques for detecting and preventing cross-site scripting attacks / World Journal of Innovation and Modern Technology. 2024. No2(8). С. 71–89

52. Antonelli D. Cascella R., Schiano A. Dirclustering: A semantic clustering approach to optimize website structure discovery during penetration testing / Journal of Computer Viruses and Malware. 2024. No20. С. 565–577.

53. Defending multi-cloud applications against man-in-the-middle attacks / [M. Reece, N. Rastogi, T. Lander та ін.] // Proceedings of the 29th ACM Symposium on Access Control Models and Technologies (SACMAT 2024). New York, NY, USA: Association forComputing Machinery. 2024. С. 47–52.

54. Enhancing web browser extensions: Preventing JavaScript code injection and vulnerabilities /[T. Singh, K. Singh, N. Varshney та ін.] // Innovative Computing and Communications. ICICC 2024. Singapore: Springer. 2024. No1020. С. 44.

55. Mapping and analysis of common vulnerabilities in popular web servers / M. Barocsai, J. Can, M. Karresand, S. Nadjm-Tehrani // Critical Information Infrastructures Security. Cham: Springer. 2024. С. 14599–14604.
56. Kumar I. S. Methodology for safeguarding cloud servers from web application attacks / International Conference on ResearchMethodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering. Chennai, India. 2023. С. 1–6.
57. Analysis of web application vulnerabilities using dynamic application security testing / R.Singh, G. M. Kumar, D. R. Patil, S. M. Patil // IEEE 9th International Conference for Convergence in Technology (I2CT). Pune, India. 2024. С. 1–6.
58. Shahrivar P. Millar S., Shereen E..Detecting web application DAST attacks with machine learning / IEEE Conference on Dependable and Secure Computing (DSC). Tampa, FL, USA. 2023. С. 1–8.
59. Kuszczyński K. Walkowski M. Comparative analysis of open-source tools for conducting static code analysis / Sensors. 2023. No18(23). С. 7978–7989.
60. Tudosi A. Research on Security Weakness Using Penetration Testing in a Distributed Firewall / Sensors. 2023. No5. С. 2683–2695.
61. Samba: Detecting SSL/TLS API misusesin IoT binary applications / [K. Liu,та ін.] // IEEE INFOCOM 2024 -IEEE Conference on Computer Communications. Vancouver, BC, Canada. 2024.С. 2029–2038.
62. Enhancing Monitoring Performance: A Microservices Approach to Monitoring with Spyware Techniques and Prediction Models / A.Rossetto, D. Noetzold, L. Silva, V. Leithardt // Sensors. 2024. No13. С. 4212–4237.
63. Alquwayzani A., Aldossri R., Frikha M. Mitigating Security Risks in Firewalls and Web Applications using Vulnerability Assessment and Penetration Testing (VAPT)

/ International Journal of Advanced Computer Science and Applications. 2024. No5. C. 1348–1364.

64. Softić J., Vejzović Z. Impact of vulnerability assessment and penetration testing (VAPT) on operating system security / 22nd International Symposium Infoteh-Jahorina (INFOTEH). East Sarajevo, Bosnia and Herzegovina. 2023. C. 1–6.
65. Dynamic Security Analysis on Android: A Systematic Literature Review / [T. Sutter, T. Kehrer, M. Rennhard та ін.] // IEEE Access. 2024. No12. C. 57261–57287.
66. Testability Tarpits:the Impact of Code Patterns on the Security Testing of Web Applications / [F. Al Kassar, G. Clerici, L. Compagna та ін.] // Annual Network and Distributed System Security Symposium. 2022. No29. C. 1–18.
67. Automatically Seed Corpus and Fuzzing Executables Generation Using Test Framework / J.Jeon, M. Ryu, D. Kim, H. Kim // IEEE Acces. 2022. No10. C. 90408–90428.
68. Коробейнікова Т. І., Кравчук Н. В. Організація захищеного доступу доweb-серверів засобами машинного навчання. 2023. No1 (20). С. 52–59.
69. Коробейнікова Т. І., Кравчук Н. В. Огляд безпечного доступудо веб-ресурсу за допомогою методів машинного навчання / TInternational Scientific Integration, Seattle, Washington, USA: ProConference. 2023. С. 26–33.
70. Кравчук, Н., & Коробейнікова, Т. (2024). Безпечний доступ досерверів інформаційних систем,забезпечений ML-моделлю для блокування шкідливих запитів.Herald of Khmelnytskyi National University. Technical Sciences,341(5).

ЗГОДА здобувача вищої освіти

Державного університету економіки і технологій про перевірку кваліфікаційної роботи на прояви академічного плагіату та розміщення в Репозитарії Університету

Я, __Томіло Микита Володимирович, підтримую політику Державного університету економіки і технологій з академічної доброчесності і відкритого доступу.

Засвідчую, що кваліфікаційна бакалаврська робота «Розробка інтерактивної веб-платформи для замовлення 3D-графічних послуг» виконана самостійно та не містить академічного плагіату. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Державного університету економіки і технологій ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення норм академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

Також я поінформований, що відповідно до «Положення про Репозитарій (електронну базу даних) Державного університету економіки і технологій» зазначена робота буде розміщена в Електронному архіві Університету (Репозитарії ДУЕТ). З умовами такого розміщення ознайомлений.

10.06.2025 Томіло М.В