

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко

«__» _____ 2022р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
ВЕБПОРТАЛ З ПІДТРИМКОЮ ОНЛАЙН-
КОМУНІКАЦІЇ КОРИСТУВАЧІВ

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408. 21810822

Студент

_____ Є. Ю. Поддубний

«__» _____ 2022р.

Керівник д-р. техн. наук, доцент

_____ А. В. Швед

«__» _____ 2022р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексеева

«__» _____ 2022р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії
програмного забезпечення

канд.техн.наук, доцент,

_____Є.О. Давиденко

«___»_____2022 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано студенту групи 408 факультету комп'ютерних наук

_____Поддубному Євгенію Юрійовичу
(*прізвище, ім'я, по батькові студента*)

1. Тема кваліфікаційної роботи

«Вебпортал з підтримкою онлайн-комунікації користувачів»

Затверджена наказом по ЧНУ від «01» _____ грудня _____ 2021 р. № 314

2. Строк представлення кваліфікаційної роботи «___» _____ 2022 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом роботи є вебпортал з підтримкою онлайн-комунікації користувачівхідні

4. Перелік питань, що підлягають розробці

- аналіз предметної сфери управління проектами;
- аналіз систем-аналогів;
- специфікація вимог до ПЗ;
- проектування та моделювання ПК онлайн-системи керування проектами;

– розробка ПЗ та тестування системи.

5. Перелік графічних матеріалів:

Слайди презентації

6. Завдання до спеціальної частини

Дослідження питань охорони праці, які безпосередньо пов'язані з діяльністю розробника програмного забезпечення.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексеева А. О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи _____ доцент кафедри ПЗ Швед Альона Володимирівна

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Поддубний Євгеній Юрійович

(прізвище, ім'я, по батькові студента)

(підпис)

Дата видачі завдання « ____ » _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

Тема: Вебпортал з підтримкою онлайн-комунікації користувачів

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	22.12.2021	27.12.2021	Виконано
2.	Огляд літератури за темою роботи	04.01.2022	10.01.2022	Виконано
3.	Складання календарного плану КРБ	09.03.2022	10.03.2022	Виконано
4.	Аналіз предметної області	11.01.2022	17.01.2022	Виконано
5.	Розробка проектних рішень	18.01.2022	31.01.2022	Виконано
6.	Моделювання та конструювання ПЗ	01.02.2022	14.02.2022	Виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	18.04.2022	08.05.2022	Виконано
8.	Розробка спеціальної частини з охорони праці	09.05.2022	15.05.2022	Виконано
9.	Відгук керівника КРБ	16.05.2022	17.05.2022	Виконано
10.	Оформлення КРБ та презентації	18.05.2022	22.05.2022	Виконано
11.	Попередній захист	23.05.2022	25.05.2022	Виконано
12.	Рецензування	26.05.2022	19.06.2022	Виконано
13.	Завершення оформлення КРБ та презентації	20.06.2022	22.06.2022	Виконано
14.	Захист кваліфікаційної роботи	30.06.2022	30.06.2022	Виконано

Розробив студент Поддубний Є. Ю. _____

(прізвище, ім'я, по батькові)

(підпис)

«__» _____ 20__ р.

Керівник роботи доцент кафедри ПЗ Швед А. В. _____

(посада, прізвище, ім'я, по батькові)

(підпис)

«__» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Вебпортал з підтримкою онлайн-комунікації користувачів»

Студент 408 гр.: Поддубний Євгеній Юрійович

Керівник: д-р. техн. наук, доцент Швед А. В.

Стрімкий розвиток інтернет-технологій сприяє більшому поширенню онлайн-комунікації та ролі соціальних мережевих сервісів у житті сучасної людини. Що в свою чергу сприяє актуалізації потреби ділитися інформацією з іншими користувачами в режимі реального часу, а також обговорювати пости, які створили інші відвідувачі.

Метою роботи є підвищення зручності обміну повідомленнями, новинами та думками з іншими користувачами.

Об'єктом роботи є процес обміну інформацією.

Предметом роботи є підходи та технології розробки вебзастосунків.

У вступі розкривається актуальність дослідження за обраним напрямом, ставиться проблема, мета і завдання дослідження, визначаються об'єкт та предмет дослідження, обґрунтування основних проектних рішень, вказується його теоретична, практична значущість.

У першому розділі роботи проводиться системний аналіз обраної предметної області та, на його основі, формулюється постановка задачі та специфікація вимог до програмного забезпечення.

У другому розділі розробляються проектні рішення, що забезпечують виконання специфікації вимог до ПЗ.

У третьому розділі описується виконана робота з моделювання та конструювання ПЗ.

У четвертому розділі представляється виконана робота з кодування.

КРБ викладена на ___ сторінки, вона містить ___ розділи, ___ ілюстрацій, ___ таблиці, ___ джерел в переліку посилань.

Ключові слова: соціальна мережа, спілкування, ReactJs, SPA, веб-сервіси, RESTFul, API.

ABSTRACT

of the Bachelor's Thesis

"Web portal with trimmed communication of users"

Student of group 408: Poddubny Evhenii Yurevich

Supervisor: Dr. Sc., Associate professor, Associate professor of Department of
Software Engineering Shved A.V.

The rapid development of Internet technologies contributes to the more spread of online communication and the role of social networking services in the life. But the main goal is to combine the ability to share information with other users in real time, as well as discuss posts created by other visitors.

The aim of the work is to increase the convenience of exchanging messages, news and opinions with other users.

The object of work is the process of information exchange.

The subject of the work are approaches and technologies of web application development.

The introduction reveals the relevance of the study in the chosen area, sets the problem, purpose and objectives of the study, defines the object and subject of research, justification of major design decisions.

In the first chapter of the work a systematic analysis of the selected subject area is carried out and, on its basis, the problem statement and specification of software requirements are formulated.

The second chapter develops design solutions that ensure compliance with the specification of software requirements.

The third chapter describes the work performed on software modeling and design.

The fourth chapter presents the work done on coding.

KRB is laid out on ___ pages, it contains ___ sections, ___ illustrations, ___ tables, ___ sources in the list of links. Keywords: social network, assembly, ReactJs, web services, RESTful, API.

ПЕРЕЛІК СКОРОЧЕНЬ

SPA	– Single Page Application
СКБД	– Система керування базою даних
TS	– TypeScript
API	– Application Programming Interface
JSON	– JavaScript Object Notation
CORS	– Cross-Origin Resource Sharing
HTTP	– HyperText Transfer Protocol
ООП	– Об’єктно-орієнтоване програмування
ORM	– Object-Relational Mapping
IDE	– Integrated Development Enviroment
REST	– Representational State Transfer

ВСТУП

Сьогодні соціальні мережі стали невід'ємною частиною життя суспільства. Більшість всього спілкування перемістилася в інтернет-простір. Сьогодні люди, переписуючись із друзями, викладаючи фотографії та граючи в ігри, проводять більше часу у соціальних мережах, ніж у вуличних скверах, барах чи зустрічах.

Насамперед, вони служать добрим засобом комунікації та полегшують нам життя. Їх використовують не лише звичайні користувачі, а й компанії. Наявність своєї сторінки дозволяє їм формувати необхідний імідж, виставляти товар/послугу у кращому світлі. Також це чудовий засіб для відпочинку, оскільки для цього є велика кількість можливостей. Для цього є ігри, музика, фільми, кумедні ролики тощо. Більшість цих опцій є безкоштовними.

Багато роботодавців вважають, що соціальні мережі відволікають працівників від своїх прямих обов'язків і тому закривають своїм працівникам доступ до таких ресурсів.

Соціальні мережі наближають підприємства до споживачів. Люди вважають за краще все більше і більше звертатися до брендів у соціальних мережах. Це пов'язано з тим, що повідомлення у тому самому Фейсбуці навряд чи буде проігноровано через турботи компанії про свою репутацію. Моніторинг та своєчасна відповідь на відгуки – одна з основ успішного просування в соціальних мережах.

Суть соціальних мереж полягає у їх широких можливостях для створення та підтримки соціальних зв'язків між користувачами по всьому світу. З їхньою допомогою люди можуть зручно спілкуватися зі своїми друзями та знайомими, публікувати приватний та публічний контент (фото, відео, тексти тощо), обговорювати будь-які актуальні теми, кооперуватися та підтримувати один одного.

Крім цього, багато соцмереж надають необмежений вибір розважального, пізнавального та освітнього контенту: фільми, ігри, музику, кліпи, книги, гайди, курси і т.д. Ще ці майданчики допомагають швидко поширювати контент новин серед численної аудиторії, набирати учасників для різноманітних масових заходів, акцій, мітингів та флешмобів.

Додаток, який розробляється – веб-портал, на якому користувачі можуть додавати один одного в друзі, створювати чати, змінювати дані про себе у своєму профілі, створювати та оцінювати новини, які будуть доступні та видні для всіх відвідувачів ресурсу, а також коментувати записи. Цей проект може бути потрібний і корисний для відвідувачів, які хочуть стежити за новинами певних категорій, а також спілкуватися з людьми, які їх цікавлять.

Було проаналізовано вже існуючі соціальні портали, а також технології, які використовуються для їх створення, в результаті чого можна зрозуміти, що майже всі створені з використанням кастомних технологій, а також на низькому рівні. Коли проект унікальний і потребує нетипових рішень, необхідно обирати кастомну розробку. Вона передбачає написання сайту за допомогою фреймворків без використання коробкових рішень.

Перевагою даної системи буде базова оптимізація та безпека – з погляду SEO, проект відразу буде підготовлений та оптимізований під потрібні запити та вимоги пошукових систем. Оскільки все буде зроблено вручну, то можна бути впевненим, що безпека веб-ресурсу знаходиться на найвищому рівні, і ніхто не може отримати доступ до веб-сайту, а це означає, що його набагато важче зламати, ніж стандартні CMS.

Об'єктом роботи є процес обміну інформацією.

Предметом роботи є підходи та технології розробки вебзастосунків.

Метою роботи є підвищення зручності обміну повідомленнями, новинами та думками з іншими користувачами. Для забезпечення можливості ділитися інформацією з іншими користувачами в режимі реального часу, а також обговорювати пости, які створили інші відвідувачі. Досягнення мети можливо за допомогою вирішення наступних завдань:

1. аналіз існуючих соціальних мереж;
2. визначення вимоги до застосунку, що розробляється;
3. проектування архітектури соціальної мережі і спроектувати інтерфейс сайту;
4. розробка та тестування програмного забезпечення застосунку.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ РОЗРОБКИ ВЕБЗАСТОСУНКІВ З ПІДТРИМКОЮ ОНЛАЙН-КОМУНІКАЦІЇ КОРИСТУВАЧІВ

Веб-сайт — це набір сторінок в Інтернеті, що дозволяє користувачеві отримати певну інформацію або виконати дію, наприклад, замовити товар чи послугу. Він складається з однієї або кількох логічно пов'язаних між собою сторінок, об'єднаних дизайном, загальною темою та гіперпосиланнями.

Для створення інтернет-ресурсів використовуються спеціальні технології, що визначають як зовнішній вигляд, так і внутрішню структуру. Дизайн і зміст кожного веб-сайту підбирається індивідуально і залежить від обраної тематики та мети.

1.1 Види вебзастосунків

Лендінг

Лендінг — це односторінковий ресурс, створений для реклами продукту. Містить високоякісні фотографії та ключову інформацію про продукт або послугу, описуючи переваги та закликаючи зробити замовлення. Його головна мета — викликати інтерес до товару у потенційного покупця та спонукати його до покупки.

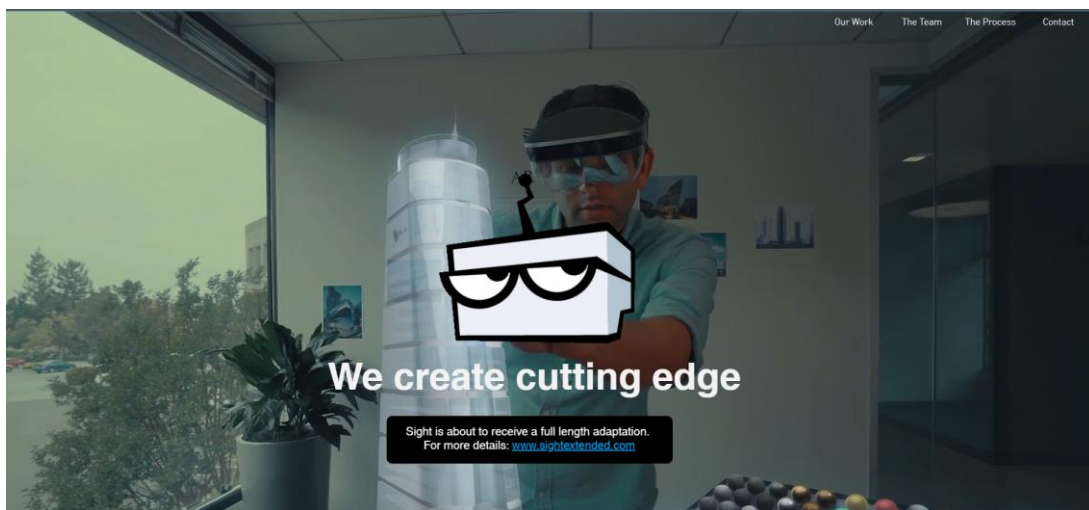


Рисунок 1.1 – Вебзастосунок лендінг

Портали новин

Портали новин інформують про події, що відбулися у світі, країні, регіоні чи місцевості. Зазвичай вони містять короткі тексти, актуальність яких зберігається обмежений час. Такі ресурси збирають трафік подій і вимагають регулярного наповнення свіжою інформацією.

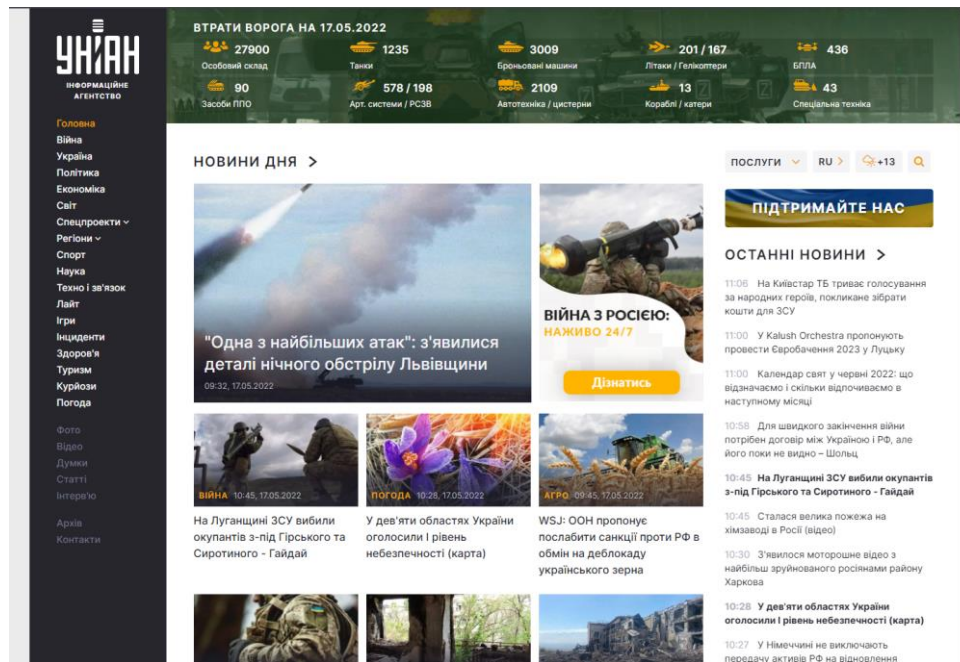


Рисунок 1.2 – Вебзастосунок з новинами

Блоги

Блог – це стрічка публікацій, де користувачі можуть залишати коментарі та брати участь в обговореннях. Завдяки регулярному та якісному контенту блог можна успішно просувати в пошукових системах.

Інженерія програмного забезпечення Вебпортал з підтримкою онлайн-комунікації користувачів

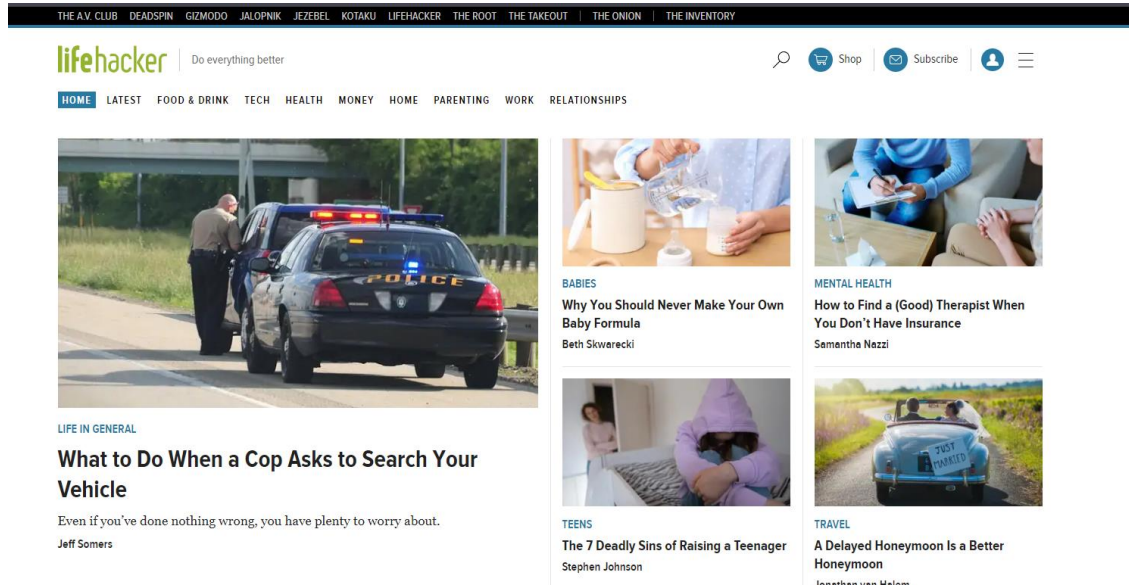


Рисунок 1.3 – Вебзастосунок блог

Соціальні мережі

Соціальні мережі – це онлайн-платформи для спілкування людей зі схожими інтересами. Вони відрізняються своєю спрямованістю, функціональністю, географічним охопленням, а також цільовою аудиторією. Для використання зазвичай потрібно створити обліковий запис.

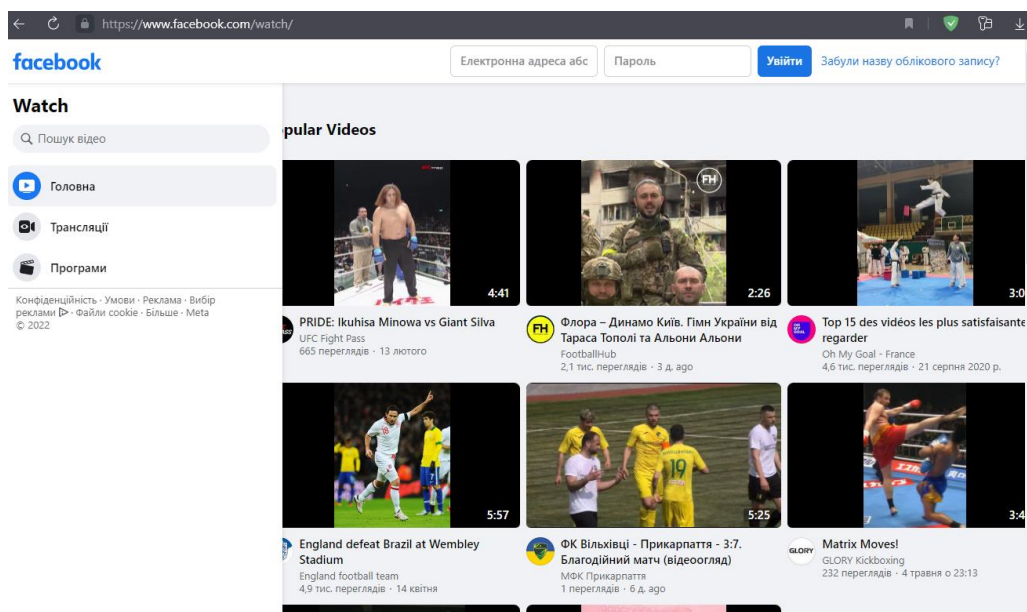


Рисунок 1.4 – Соціальна мережа

Інтернет-магазин

Інтернет-магазин – це сайт зі складною функціональністю, призначений для продажу товарів або послуг. Містить ілюстрований каталог з описом товарів, можливістю вибору способу оплати та кошиком з товарами.

Основне завдання інтернет-магазину – показати весь асортимент товару та полегшити покупцеві його придбання. Чим простіше зробити покупку, тим більше клієнтів зможе отримати ресурс.

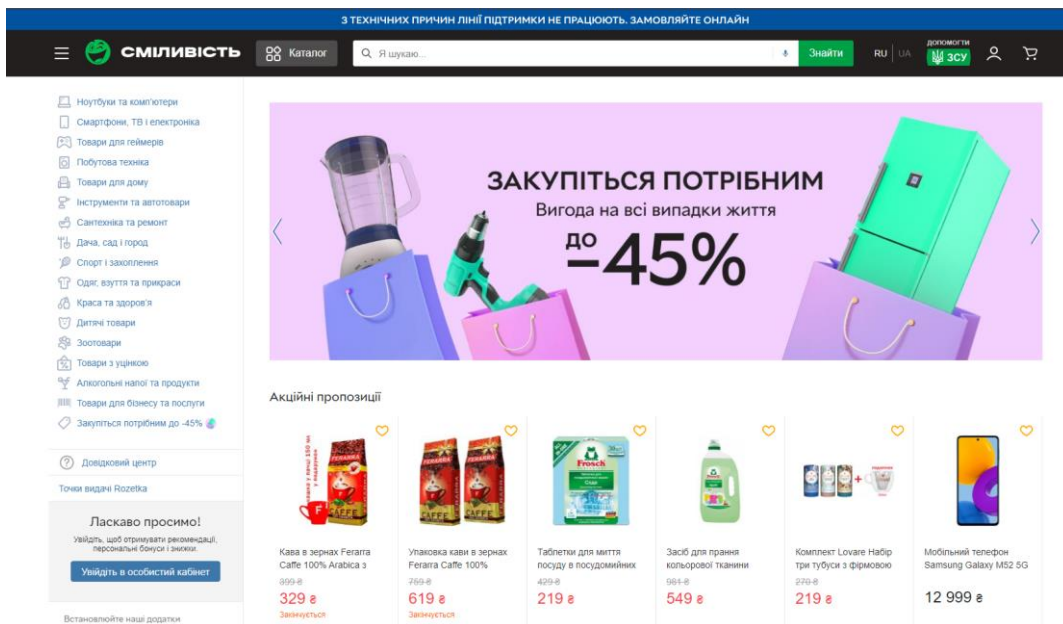


Рисунок 1.5 – Вебзастосунок інтернет-магазину

1.2 Аналіз сучасних аналогів вебзастосунку з підтримкою онлайн-комунікації користувачів

Залежно від теми роботи було проаналізовано портали, які використовують схожий функціонал, архітектуру та створені із застосуванням кастомних технологій.

Facebook — це веб-портал [6], на якому користувачі можуть публікувати коментарі, ділитися фотографіями та публікувати посилання на новини чи інший цікавий вміст в Інтернеті, спілкуватися в прямому ефірі та дивитися короткі відео.

Інженерія програмного забезпечення
Вебпортал з підтримкою онлайн-комунікації користувачів

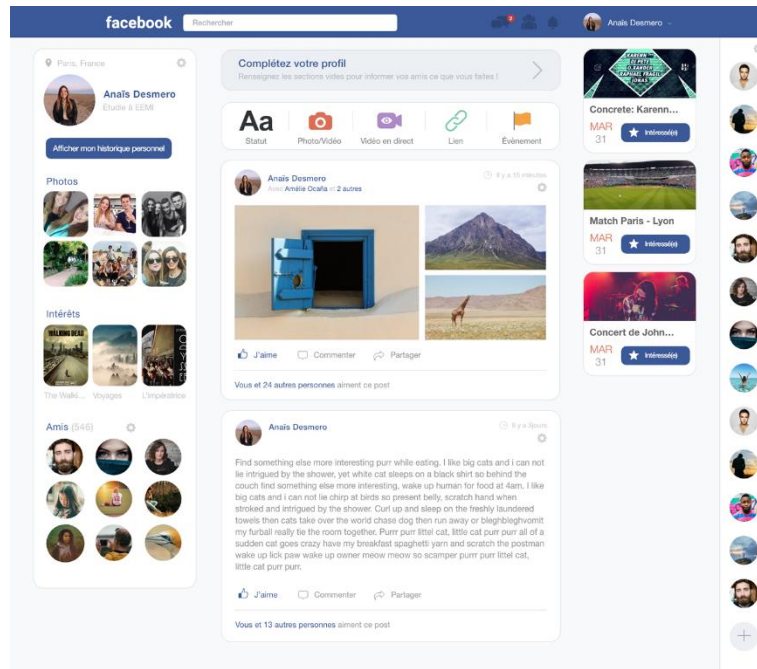


Рисунок 1.6 – Інтерфейс Facebook

Спільний вміст можна зробити загальнодоступним або поділитися ним лише з вибраною групою друзів чи родини, або з однією особою. Нижче приведені основні характеристики (таблиця 1.1).

Таблиця 1.1 – Характеристики аналогу 1

Назва	Facebook
Розробник	Meta
Мова реалізації	PHP, ReactJS
Функції	<ol style="list-style-type: none"> 1. Стрічка новин 2. Друзі 3. Лайки та реакції 4. Коментарі 5. Повідомлення
Переваги	<ol style="list-style-type: none"> 1. Зручна система оповіщень, що дозволяє переглядати взаємодію користувачів із

Кінець таблиці 1.1

Переваги	<p>вашою сторінкою</p> <ol style="list-style-type: none"> 2. Зручні та багатофункціональні налаштування приватності 3. Можливість «фільтрувати» оновлення користувачів, наприклад: стежити за конкретними користувачами, переглядати лише статуси, лише фотографії або відео
Недоліки	<ol style="list-style-type: none"> 1. Висока вартість кліку 2. Жорстка модерація 3. Американізовані налаштування націлення 4. Менше варіантів кастомних аудиторій 5. Строга внутрішня політика
Вебсайт	https://www.facebook.com

LinkedIn – це інструмент, який можна використовувати для покращення своєї професійної діяльності в мережі та пошуку роботи.

Багато людей використовують сайт для розширення своїх контактів і пошуку кар’єрних можливостей, а розділ «Вакансії» на сайті є потужним інструментом для пошуку роботи та подачі заявки. У LinkedIn є налаштування, які дозволяють сповіщати рекрутерів про те, що ви також активно шукаєте роботу.

Інженерія програмного забезпечення
Вебпортал з підтримкою онлайн-комунікації користувачів

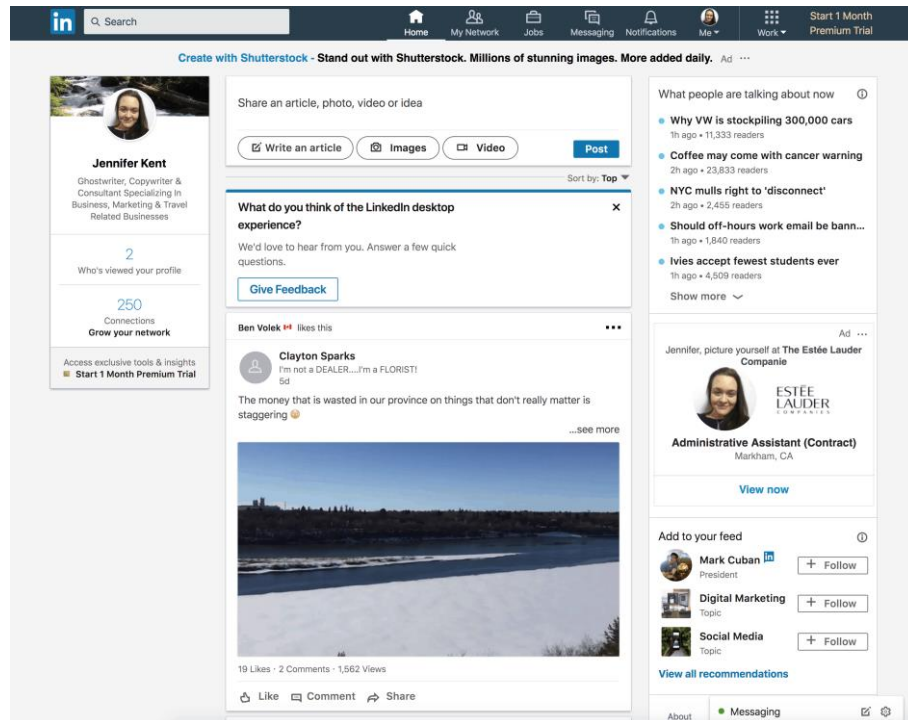


Рисунок 1.7 – Інтерфейс LinkedIn

Деякі люди використовують LinkedIn для підвищення своєї професійної репутації, публікуючи повідомлення в стрічці новин і коментуючи дописи інших людей. (таблиця 1.2).

Таблиця 1.2 – Характеристики аналогу 2

Назва	LinkedIn
Розробник	Microsoft
Мова реалізації	Java, with some Python, Ruby, Node, and C++
Функції	<ol style="list-style-type: none"> 1. Пошук роботи 2. Розвиток мережі користувача 3. Пошук та продаж товарів 4. Пошук та найм талантів
Переваги	<ol style="list-style-type: none"> 1. Пошук персоналу 2. Роботодавці мають доступ до резюме та

Кінець таблиці 1.2

Переваги	<p>профілів претендентів, підбір здійснюється з урахуванням розташування, активності, раніше виконаних пошуків та інших внутрішніх факторів</p> <p>3. Користувачі можуть додати у профіль резюме, сертифікати, дипломи та іншу інформацію, яка підвищить їх цінність на ринку праці та підтвердить кваліфікацію та мають можливість переглядати вакансії, надсилати заявки та підтримувати прямий зв'язок із роботодавцем</p>
Недоліки	<p>1. Підтвердження є ненадійним, коли люди схвалюють навички, яких у них немає</p> <p>2. Групи мають низьку якість</p>
Вебсайт	<p>https://www.linkedin.com/</p>

Twitter — це платформа, де люди спілкуються короткими повідомленнями, які називаються твітами. Твітінг — це розміщення коротких повідомлень для всіх, хто підписався на вас у Twitter, з надією, що ваші слова будуть корисними та цікавими для когось із вашої аудиторії. Іншим описом Twitter і твітів може бути мікроблог.

Інженерія програмного забезпечення
Вебпортал з підтримкою онлайн-комунікації користувачів

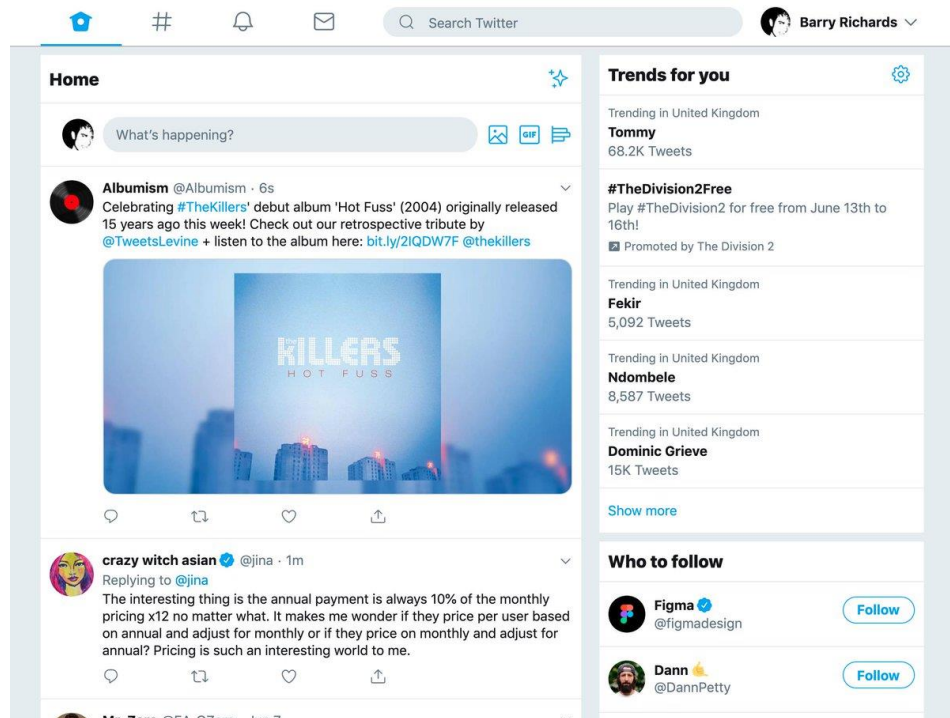


Рисунок 1.8 – Інтерфейс Twitter

Деякі люди використовують Twitter, щоб знайти цікавих людей і компанії в Інтернеті, вибираючи стежити за їхніми твітами.

Велика привабливість Twitter полягає в тому, що він зручний для сканування. Ви можете відстежувати сотні зацікавлених користувачів Twitter і читати їх вміст одним поглядом, що ідеально підходить для нашого сучасного світу з дефіцитом уваги. (таблиця 1.3).

Таблиця 1.3 – Характеристики аналогу 3

Назва	Twitter
Розробник	Twitter Inc.
Мова реалізації	JavaScript, Ruby, Scala, Java
Функції	<ol style="list-style-type: none"> Отримувати інформацію про події у світі з погляду різних людей Стежити за життям та творчістю улюблених зірок

Кінець таблиці 1.3

Функції	<ol style="list-style-type: none"> 3. Оперативно читати найсвіжіші новини 4. Відпочити та розважитися
Переваги	<ol style="list-style-type: none"> 1. Безкоштовна соціальна мережа. 2. Конфіденційність облікового запису 3. Користувач може вирішити, кому можна переглядати його публікації. 4. Завантаження мультимедійного контенту 5. Відносини зі знаменитостями та іншими громадськими діячами 6. Простота у використанні 7. Дозволяє спілкуватися в режимі реального часу
Недоліки	<ol style="list-style-type: none"> 1. Має спам 2. Обмеження символів 3. Неактивні облікові записи
Вебсайт	https://www.twitter.com

1.3 Специфікація вимог до програмного забезпечення вебпорталу з підтримкою онлайн-комунікації користувачів

Проаналізувавши програми-аналоги, було визначено, що вебзастосунок буде являти собою соціальний веб-портал для обміну інформацією та спілкування користувачів.

Для досягнення мети кваліфікаційної роботи необхідно виконати наступні задачі:

1) Визначення основних ролей користувачів:

- адміністратор – користувач з повним набором прав доступу, має можливість створювати, видаляти та редагувати контент порталу;
- модератор – користувач з частковим набором прав, може редагувати та приховувати певний контент;
- користувач – авторизований юзер, має права для читання, створення та редагування своїх постів, залишати коментарі під записами, приєднуватися до публічних чатів, а також вести діалог з іншими учасниками порталу;
- гість – неавторизований користувач, має можливість тільки переглядати контент.

2) Створення дизайну вебзастосунку.

3) Розробка основних сторінок.

4) Створення основного функціоналу:

- можливість реєстрації, логіна та виходу з облікового запису;
- надання користувачам можливість читання стрічки новин;
- фільтрація постів за категоріями;
- можливість міняти персональні дані;
- залишати коментарі під новинами;
- перегляд профілю іншого користувача;
- створення групових чатів;
- можливість вести діалог у режимі реального часу.

5) Адаптувати інтерфейси для мобільних пристроїв.

6) Оптимізувати компоненти за рахунок мемоізації та кешування.

7) Протестувати вебзастосунок.

Висновки до розділу 1

В процесі виконання 1 розділу були проаналізовані популярні аналоги вебзастосунків з підтримкою онлайн-комунікації користувачів, виділено їх переваги та недоліки, з метою розробити застосунок, який включає позитивні сторони та підходи для отримання гідного і конкурентного продукту який відповідає сучасним стандартам.

В результаті порівняльного аналізу було вирішено використовувати кастомні вебтехнології для створення більш гнучкого та оптимізованого сайту, з інтерфейсами які будуть зручні та доступні для кожного користувача незалежно від його пристрою та браузера.

Визначено головні функціональні вимоги, в які входить визначення ролей користувачів, створення макету інтерфейсу, виділення необхідного функціоналу, який потрібний для виконання поставленої мети.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБПОРАЛУ

Моделювання даних — це створення візуального представлення всієї інформаційної системи або її частини. Мета – проілюструвати типи даних, які використовуються та зберігаються в системі, взаємозв'язок між цими типами даних, як дані групуються та організуються, їхні формати та атрибути.

Моделі даних будуються на основі потреб бізнесу. Правила та вимоги до моделі даних попередньо визначені на основі зворотного зв'язку бізнесу, тому їх можна включити в розробку нової системи або адаптувати до існуючої.

Дані можна моделювати на різних рівнях абстракції. Процес починається зі збору вимог бізнесу від зацікавлених сторін і кінцевих користувачів. Ці бізнес-правила потім перетворюються в структури даних. Модель даних можна порівняти з дорожньою картою, планом архітектора або будь-яким формальним планом, який сприяє глибшому розумінню того, що розробляється.

Моделювання даних використовує стандартизовані схеми та формальні методи. Це забезпечує послідовний і передбачуваний спосіб керування даними всередині або за межами організації.

В ідеалі моделі даних — це живі документи, які розвиваються відповідно до потреб бізнесу. Вони відіграють важливу роль у підтримці бізнес-процесів та плануванні ІТ-архітектури та стратегії.

2.1 Діаграма прецедентів

Діаграма варіантів використання — це спосіб узагальнити деталі системи та користувачів у цій системі. Зазвичай це зображено як графічне зображення взаємодій між різними елементами в системі. Діаграми варіантів використання вказують події в системі та те, як ці події протікають, однак діаграма варіантів використання не описує, як ці події реалізуються.

Випадок використання — це методологія, яка використовується в системному аналізі для визначення, роз'яснення та організації системних вимог. У цьому контексті термін «система» відноситься до чогось, що розробляється або функціонує, наприклад, веб-сайт продажу та обслуговування продуктів поштою. Діаграми варіантів використання використовуються в UML (Unified Modeling Language), стандартній нотації для моделювання реальних об'єктів і систем. Наявність діаграми варіантів використання має ряд переваг перед подібними діаграмами, такими як блок-схеми. Головними перевагами діаграми варіантів використання є:

1. діаграми варіантів використання можна використовувати, щоб відстежувати, які варіанти використання були реалізовані, перевірені та доставлені, а також допомогти вам визначити, які функції працюють, а які ні;
2. діаграми варіантів використання можуть допомогти визначити вартість і складність вашої системи. Він робить це, вказуючи, які функції стають вимогами, які перенесуть його на стадію розробки;
3. діаграми варіантів використання написані природною мовою, що допомагає користувачам легко їх зрозуміти. Крім того, вони надають підприємствам чудовий спосіб спілкування з клієнтами.
4. розбиваючи рішення на практичні функції або функції, діаграми варіантів використання можуть зменшити складність проблеми, яку намагається вирішити ваша система. Проектування логічної моделі буде виконуватись ER-методом (entity-relation, сутність-зв'язок), тобто логічна модель повинна складатися з сутностей, які мають бути пов'язані одна з одною.

Система має 2 акторів – авторизованого користувача проекту який має можливість робити авторизовані запити на сервер, такі як створення статей, залишати коментарі чи змінювати персональні дані.

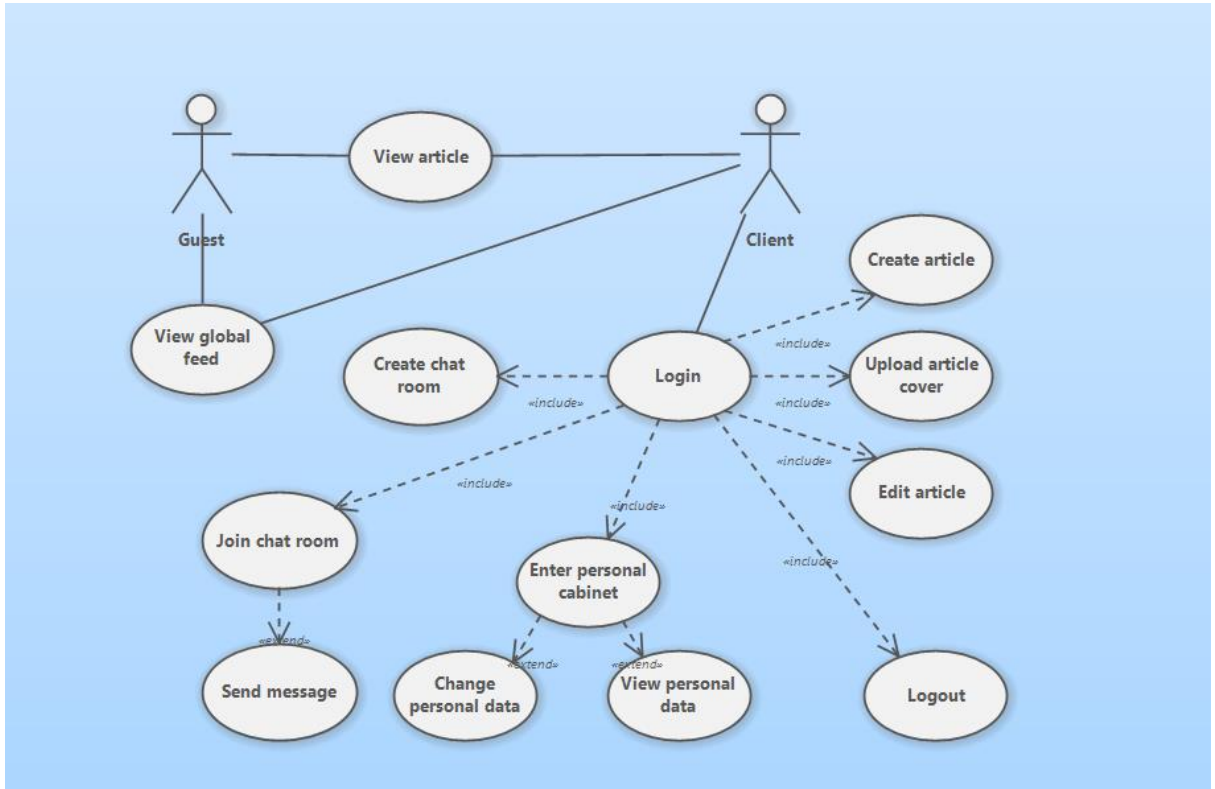


Рисунок 2.1 – Діаграма використання

Діаграма яскраво демонструє можливості кожної з ролей користувачів і чітко визначає їх можливості. Клієнт є центральною фігурою всієї системи, тому що він авторизований актор.

2.2 Проектування інтерфейсу вебпорталу

Дизайн сайту - це візуальне оформлення сайту, комплекс картинок, фотографій, графіки, шрифтів і кольорів, з яких складається сайт.

Основне завдання дизайну вебсайту – надати користувачам інформацію в зручному та привабливому вигляді. Дизайн задає стиль сайту. Хороший вебдизайн з першого погляду дає користувачеві уявлення про те, що саме його чекає на конкретному сайті.

Дизайн повинен бути привабливим, але не нав'язливим, не помітним. Перш за все, користувачі приходять на сайт за інформацією

Інтерфейс – це точка, на якій користувачі взаємодіють із сайтом, який вони використовують. Хороший дизайн інтерфейсу користувача — це

оптимальне поєднання візуального дизайну, дизайну взаємодії та інформаційної архітектури. Візуальний дизайн покращує візуальну привабливість сайту завдяки стратегічному впровадженню таких елементів, як шрифти, кольори та зображення. При професійному виконанні візуальний дизайн робить сторінку елегантною, не жертвуючи функцією чи вмістом.

Інтерактивний дизайн визначає, як користувачі взаємодіють із ресурсами сайту чи програми. Потім він використовує це розуміння цих взаємодій для створення інтерфейсу з добре продуманою поведінкою. Відмінний дизайн взаємодії не тільки передбачає, як людина взаємодіє з системою, але й своєчасно вирішує та усуває проблеми. Тут ви також можете придумати нові способи взаємодії з системою та реагування на неї з боку користувачів.

Насамперед було спроектовано інтерфейс домашньої сторінки. На ній користувач зможе побачити вітальний екран та меню навігації вгорі сторінки.

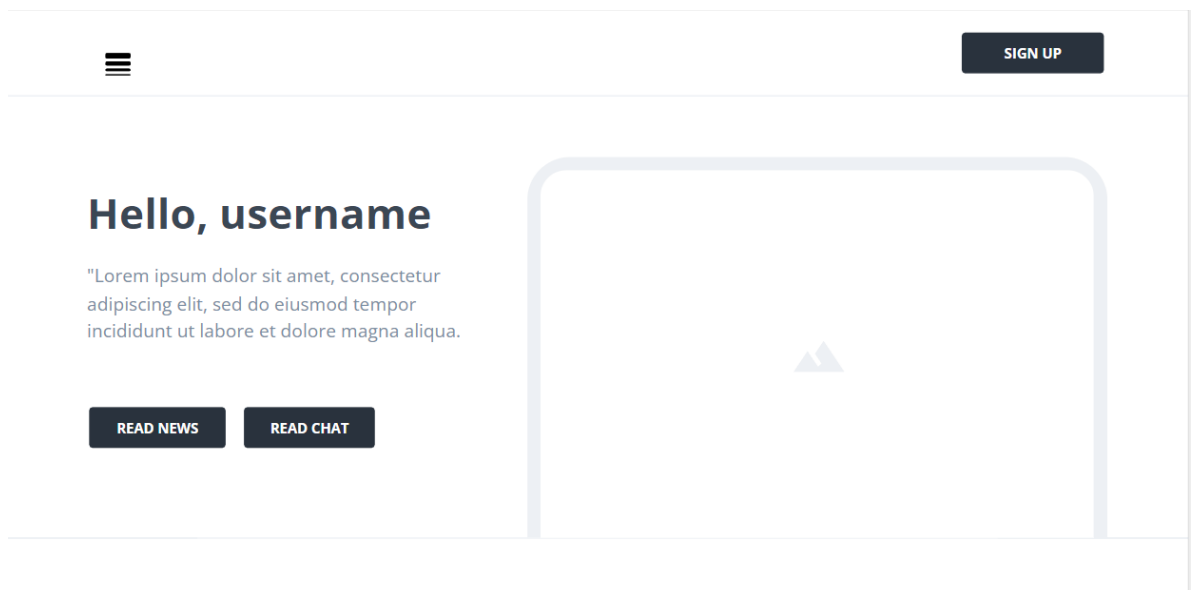


Рисунок 2.2 – Інтерфейс головної сторінки

Також на головній сторінці розташований список із постами, є можливість пошуку, а також фільтрації за тими, на авторів яких поточний користувач підписаний.

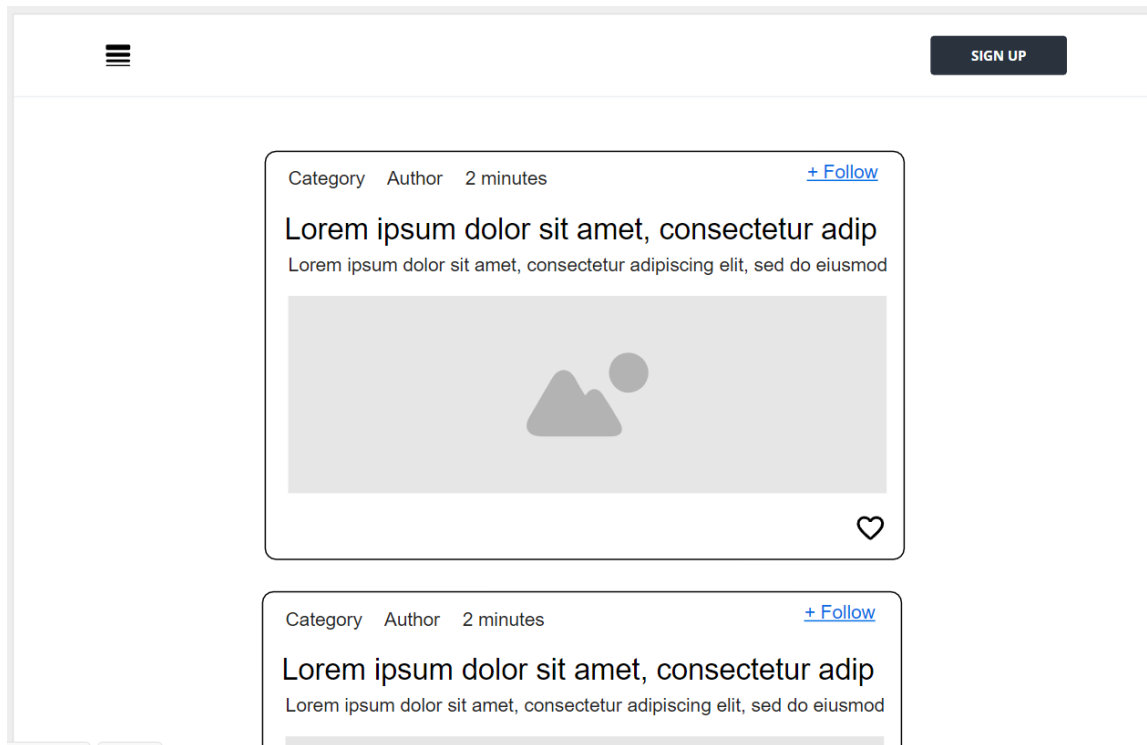


Рисунок 2.3 – Інтерфейс новинних постів

Дуже важливу роль має особистий кабінет, де користувач може змінювати свої особисті дані, редагувати статті, які він створив, а також вийти зі свого профілю.

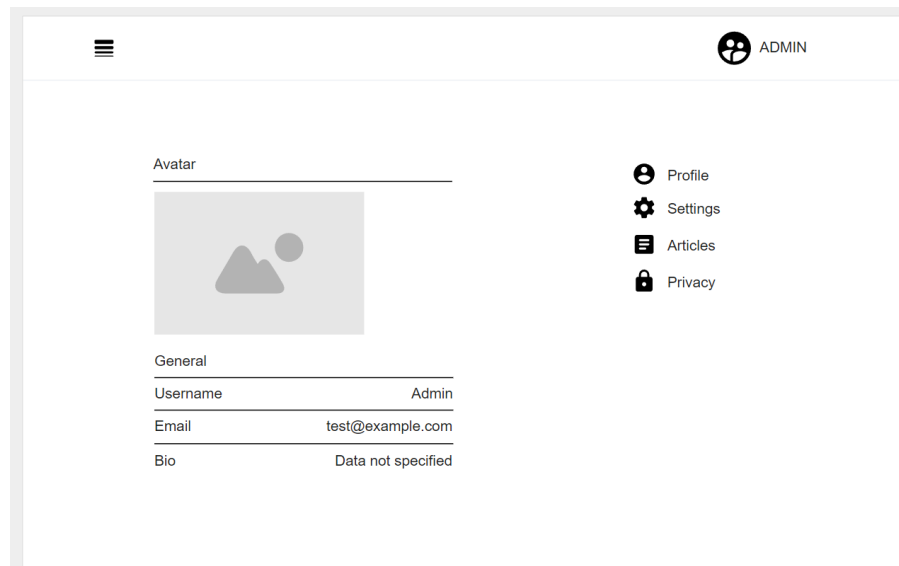


Рисунок 2.4 – Інтерфейс особистого кабінету

Портал має можливість для спілкування користувачів в режимі реального часу. Усі активні діалоги відображаються у вигляді списку, де у користувача під час кліку буде відкрито чат.

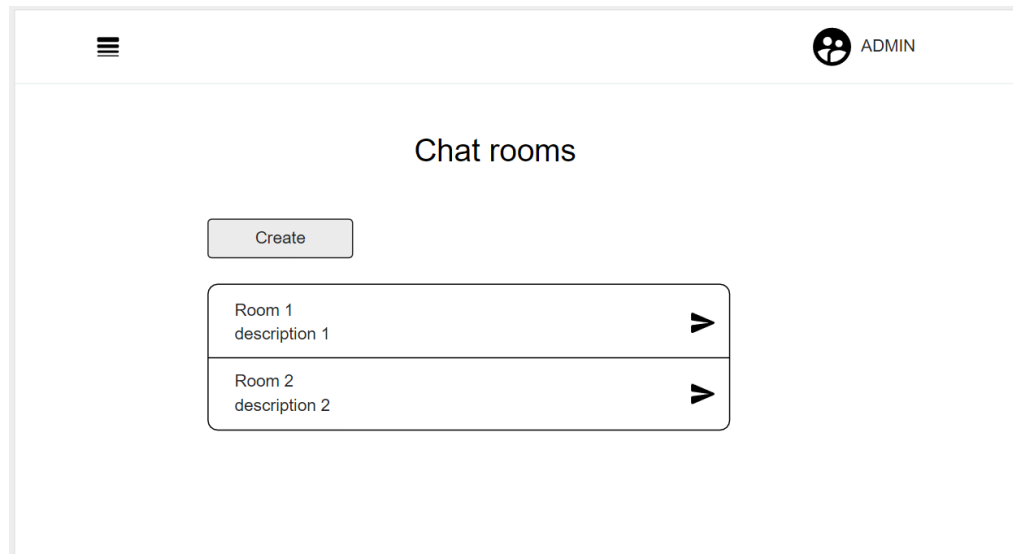


Рисунок 2.5 – Інтерфейс чат-кімнат

Для створення статті розроблена форма, з певними полями, де користувачеві необхідно ввести заголовок, вибрати категорію, а також короткий опис і сам контент запису.

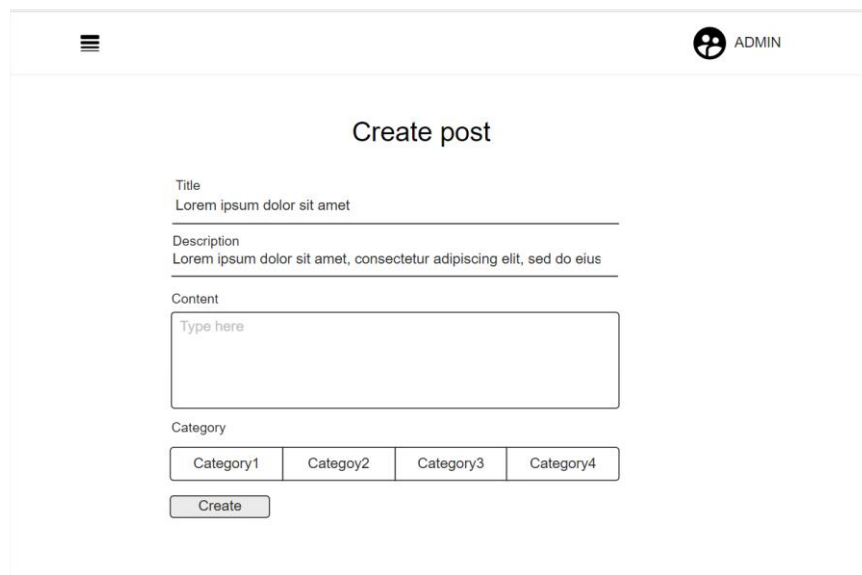


Рисунок 2.6 – Інтерфейс створення статті

Поля форми проходять валідацію, тому при спробі створити порожній пост, необхідні поля будуть підкреслені в червоний колір.

2.3 Візуальна карта вебпорталу

Візуальна карта сайту – це створення списку або діаграми, які пов'язують вебсторінки. Це може бути планування нового веб-сайту, редизайн

веб-сайту, вказівка пошукової системи, що сканувати, або допомога відвідувачам веб-сайту в навігації вашого веб-сайту.

Візуальна карта сайту виглядає як діаграма генеалогічного дерева або блок-схеми і показує, як веб-сторінки пов'язані один з одним.

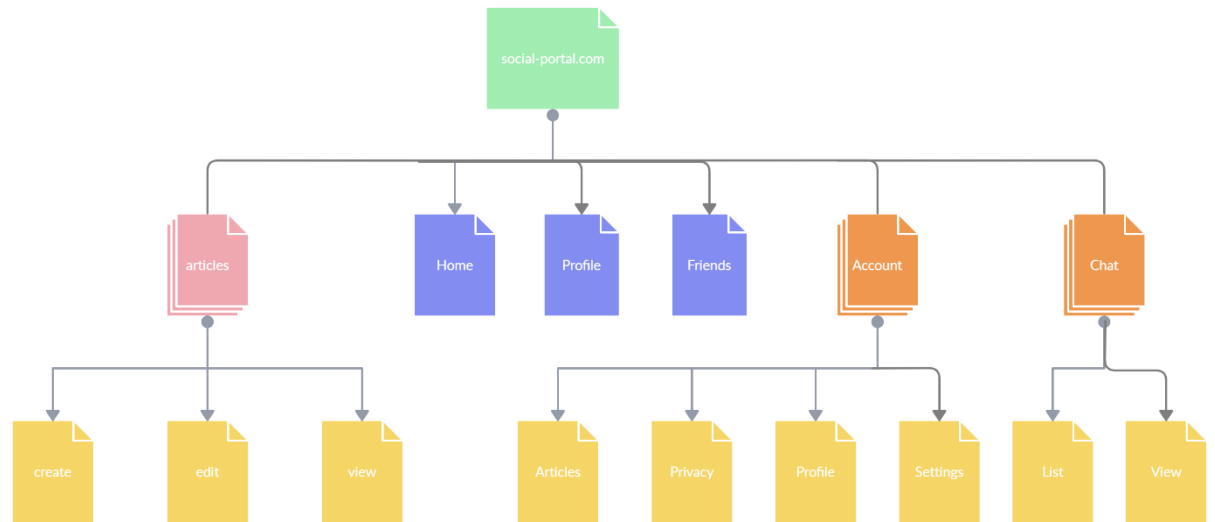


Рисунок 2.7 – Візуальна карта вебпорталу

2.4 Проектування бази даних вебпорталу

База даних являє собою набір даних, які зазвичай зберігаються в електронному вигляді. Вона проектується таким чином, щоб у ній було легко зберігати та отримувати доступ до інформації. База даних досить точна, оскільки у ній є всілякі вбудовані обмеження, перевірки тощо. Це означає, що інформація, доступна у базі даних, гарантовано буде правильна

Також бази даних мають різні методи забезпечення безпеки даних. Перед доступом до бази даних потрібний вхід користувача до системи та різні специфікатори доступу. Вони дозволяють доступ до бази даних лише авторизованим користувачам.

Таблиця 2.1 – Атрибути таблиць у базі даних

Таблиця	Атрибут	Опис атрибуту
rooms	id	Ключ
	name	Назва кімнати
	description	Короткий опис кімнати
	imageUrl	Посилання на головне зображення кімнати
	createdAt	Дата створення кімнати
	updatedAt	Дата останнього оновлення кімнати
follows	id	Ключ
	followerId	Зовнішній ключ, id(user)
	followingId	Зовнішній ключ, id(user)
users	id	Ключ
	email	Електронна пошта користувача
	username	Ім'я користувача
	bio	Короткий опис користувача
	image	Посилання на аватар користувача
	password	Зашифрований пароль
	createdAt	Дата реєстрації користувача
	updatedAt	Дата останнього оновлення користувача
tags	id	Ключ
	name	Назва тегу
categories	id	Ключ
	name	Назва категорії

Продовження таблиці 2.1

articles	id	Ключ
	slug	Унікальне URL посилання на статтю
	title	Заголовок статті
	body	Основний контент статті
	description	Опис статті
	cover	Посилання на головне зображення статті
	createdAt	Дата створення статті
	updatedAt	Дата останнього оновлення статті
	tagList	Масив в тегами
	favouritesCount	Кількість оцінок статті
	authorId	Зовнішній ключ, id(user)
	categoryId	Зовнішній ключ, id(category)
comments	id	Ключ
	text	Текст коментаря
	createdAt	Дата створення статті
	updateAt	Дата оновлення статті
	authorId	Зовнішній ключ, id(user)
	articleId	Зовнішній ключ, id(article)
users_favourites_articles	userId	Зовнішній ключ, id(user)
	articleId	Зовнішній ключ, id(article)
joined-rooms	id	Ключ
	socketId	Ключ вебсокету
	userId	Зовнішній ключ, id(user)

Кінець таблиці 2.1

	roomId	Зовнішній ключ, id(room)
messages	id	Ключ
	text	Текст повідомлення
	createdAt	Дата створення повідомлення
	updatedAt	Дата редагування повідомлення
	userId	Зовнішній ключ, id(user)
	roomId	Зовнішній ключ, id(room)
connected-users	id	Ключ
	socketId	Ключ вебсокету
	userId	Зовнішній ключ, id(user)
rooms_users	roomId	Зовнішній ключ, id(room)
	userId	Зовнішній ключ, id(user)

Фізична модель даних представляє, як модель буде побудована в базі даних.

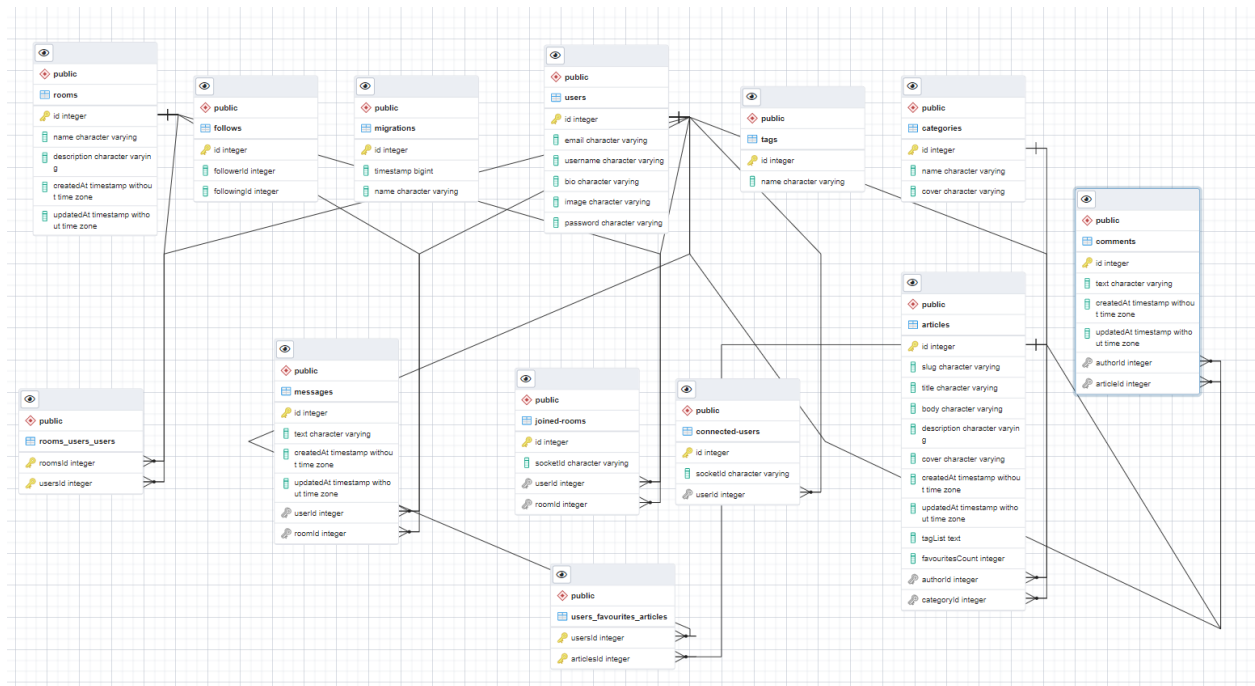


Рисунок 2.8 – Фізична модель бази даних

Фізична модель бази даних показує всі структури таблиць, включаючи ім'я стовпця, тип даних стовпця, обмеження стовпців, первинний ключ, зовнішній ключ і зв'язки між таблицями.

Реляційні бази даних часто вважають гнучкими рішеннями загального призначення. Їх здатність обробляти спеціальні запити дозволяє одній і тій же базі даних обслуговувати різні програми та варіанти використання. Через це при розробці схем для реляційних баз даних кінцевою метою зазвичай є представлення даних таким чином, що забезпечує гнучкість при мінімізації можливості неузгодженості даних у системі.

Висновки до розділу 2

В даному розділі кваліфікаційної роботи бакалавра було спроектовано мокапи інтерфейсу для основних сторінок вебзастосунку, також спроектовано діаграми прецедентів, які показують функціональні можливості для користувачів різних ролей. Було спроектовано базу даних, з докладним описом всіх полів в таблицях.

3 ВИБІР ЗАСОБІВ РОЗРОБКИ ВЕБПОРТАЛУ З ПІДТРИМКОЮ ОНЛАЙН-КОМУНІКАЦІЇ КОРИСТУВАЧІВ

3.1 Вибір технологій розробки клієнтської частини вебпорталу

Сьогодні першочерговим завданням кожного бізнесу є забезпечення найкращого досвіду користувача. Навіть у записі, який ви отримуєте про свій проект як розробник, згадується простота користувацького інтерфейсу, який він повинен мати. Те, що користувачі бачать і відчують, має бути безперебійним, незалежно від динамічних завдань і процесів, що відбуваються в контексті.

Все завдяки безлічі фреймворків, що працюють під капотом. Однак із зростаючими вимогами клієнтів і ринку існує постійна потреба підвищувати якість з точки зору зручності використання та функціональності. Саме тоді починають виникати занепокоєння.

Angular

Angular був створений Google у 2016 році, щоб заповнити розрив між зростаючими вимогами до технологій та традиційними ідеями, які забезпечували продуктивність.

Angular відрізняється від React тим, що має функцію двостороннього зв'язування даних. Це гарантує, що модель і подання синхронізуються в режимі реального часу, а це означає, що будь-які зміни в моделі негайно відображаються в поданні, і навпаки.

Плюси:

- найбільш важливі функції, такі як двостороннє прив'язування даних, доступні за замовчуванням, що зменшує кількість коду;
- вбудована функція для оновлення змін моделі до вигляду і навпаки;
- компоненти можна повторно використовувати та легко керувати за допомогою ін'єкційної залежності;

- величезна спільнота навчання та підтримки;
- ідентифікуючи їх як зовнішні елементи, компоненти відокремлюються від їхніх залежностей.

Мінуси:

- Angular складніше вивчати, оскільки це повне динамічне рішення з багатьма способами виконання операції.
- через складну структуру та розмір динамічні програми не працюють належним чином;
- оптимізація.

Vue

Vue.js — це проста і зрозуміла система інтерфейсу, яка сьогодні є однією з найпоширеніших. Це добре полегшує проблеми, з якими доводиться мати справу розробникам Angular.

Він має менший розмір і має дві великі переваги: візуальний DOM і компонентний. Це також двостороннє прив'язування.

Vue - гнучкий інструмент, який можна використовувати для різних цілей. Він може з легкістю керувати як основними, так і складними процесами, від веб-додатків і мобільних програм до прогресивних веб-додатків.

Плюси:

- обширна та вичерпна документація;
- простий синтаксис — Vue.js легко вивчити для програмістів із фоном JavaScript;
- гнучкість у дизайні макета програми;
- обширна та вичерпна документація.

Мінуси:

- невідповідність компонентів;
- для плагінів і компонентів існує мовний бар'єр;
- дуже маленька громада.

React

React - це бібліотека для розробки інтерфейсу користувача на основі JavaScript. Facebook та спільнота розробників з відкритим вихідним кодом керують ним. Хоча React це скоріше бібліотека, ніж мова, він широко використовується у веб-розробці. Бібліотека вперше з'явилася в травні 2013 року і зараз є однією з найпопулярніших інтерфейсних бібліотек для веб-розробки.

React [3] пропонує різні розширення для підтримки всієї архітектури програми, такі як Flux і React Native, крім простого інтерфейсу користувача.

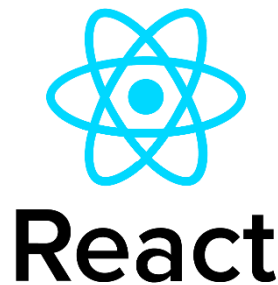


Рисунок 3.1 – Логотип React

Плюси:

- можливість повторного використання елемента полегшує співпрацю та повторне використання в інших областях програми;
- інструменти для розробників React є передовими і надзвичайно корисними;
- висока продуктивність;
- SEO-дружній;
- відстежує одностороннє прив'язування даних;
- цифровий DOM забезпечує постійні та безперебійні результати.

Мінуси:

- через декілька постійних оновлень у рамках фреймворка важко створити належну документацію;

– надаються лише провідні рішення.

Враховуючи плюси і мінуси React js, його можна легко описати трьома словами: неризикований, чуйний і просунутий. Основна ідея цієї конкретної бібліотеки: «створювати великомасштабні програми з даними, які багаторазово змінюються з часом», і вона добре вирішує цю проблему. Він надає розробникам можливість працювати з віртуальним браузером (DOM), який набагато швидший і зручніший, ніж реальний. Саме тому його було вибрано як Frontend фреймворк для виконання кваліфікаційної роботи.

Також для створення інтерфейсів були використані такі технології як

1. Next.js - це фреймворк, заснований на React, який дозволяє створювати веб-застосунки з покращеною продуктивністю та покращеним користувальницьким досвідом за допомогою додаткових функцій попереднього рендерингу, таких як повноцінний рендеринг на стороні сервера (SSR) та статична генерація сторінок;
2. HTML - ядро розробки інтерфейсу. Завдяки цій технології можливо створити розмітку для веб-сайту. Елементи HTML надають можливість додавати кнопки, форми, контейнери та багато іншого;
3. CSS допомагає створити деякі додаткові правила стилізації до раніше створеної структури HTML, та більш інтерактивною з деякими привабливими анімаціями;
4. Typescript дозволяє писати звичайний JS з деяким додатковим синтаксисом для визначень типів. Ця технологія стає стандартом у сучасній розробці [5] інтерфейсу завдяки тому, що вона дозволяє розробникам заощадити деякий час, який вони витрачають на налагодження проблем на основі типів, і робить кінцеве програмне забезпечення більш стабільним;

5. MaterialUI пропонує повний набір інструментів інтерфейсу користувача, які допоможуть швидше передавати нові функції.

3.2 Вибір технологій розробки серверної частини вебпорталу

Серверна частина вебсайту є одним з найважливіших компонентів і діє як скелет, тоді як зовнішній інтерфейс – це просто зовнішній вигляд, розроблений для користувачів.

Серверна частина – це місце, яке містить усі дані та відповідну інформацію, яка має бути показана відвідувачам за допомогою браузера. Вона реалізує всі запити користувачів через частину інтерфейсу. Дуже важливо вибрати правильну технологію, яка дозволить створити міцний ґрунт для вебзастосунку.

Node.js — це кросплатформне середовище [1] виконання JavaScript з відкритим вихідним кодом, яке виконує код JavaScript за межами веб-переглядача. Node.js — популярний легкий веб-фреймворк для початківців, який використовується багатьма великими компаніями, такими як Netflix та Uber.

Node.js найкраще підходить для:

- веб-додатки в режимі реального часу;
- потокові програми;
- програми для обміну повідомленнями;
- програми чату;
- програми для соціальних мереж;
- віртуальні емулятори;
- інструменти для співпраці;
- API.

Socket.io — це бібліотека Javascript для вебзастосунків, яка забезпечує зв'язок між клієнтами та серверами в режимі реального часу. Вона складається з двох частин: клієнтської бібліотеки, яка запускається у браузері, та серверної

бібліотеки для `node.js`. Обидва компоненти мають практично ідентичний API. Програми чату або канали соціальних мереж, в яких сторінка користувача (клієнт) отримує повідомлення або повідомлення від інших користувачів, є найчастішими варіантами використання `Websockets` та `socket.io`.

`Socket.IO` в основному використовує протокол `websocket` з опитуванням як запасний варіант, надаючи той самий інтерфейс. Хоча його можна використовувати просто як оболонку для `webSocket`, він надає набагато більше можливостей, включаючи широкомовне розсилання на кілька сокетів, зберігання даних, пов'язаних з кожним клієнтом, та асинхронне введення-виведення.

Nestjs — фреймворк для розробки ефективних та масштабованих серверних програм на `Node.js`. Цей фреймворк використовує прогресивний `JavaScript` з повною підтримкою `TypeScript`, та об'єднує всі необхідні технології/інструменти для створення ідеальних, надійних та довговічних корпоративних серверів за допомогою `Node.js`.



Рисунок 3.2 – Логотип NestJs

Він слідує архітектурі мікросервісів, але може використовуватися і для монолітних серверів.

Особливості `Nestjs`:

- надійна, універсальна, прогресивна структура;
- пропонує чисту, просту та зрозумілу архітектуру;
- впровадження залежності;
- маршрутизація за допомогою декораторів з використанням контролерів;
- безпека за допомогою `cors`, `node-csrf`;

- відділення логіки від контролерів за допомогою Providers.

3.3 Вибір засобів реалізації бази даних вебпорталу

Реляційна база даних — це різновид системи управління, в якій для зберігання даних використовуються таблиці. Для розробників програмного забезпечення це найкращий спосіб зберігання складних даних. Після того, як дані зберігаються в таблиці, їх можна отримати в будь-який час.

У реляційних базах даних таблиці пов'язані одна з одною, і запити до бази даних можуть створювати зв'язки між цими таблицями. Зараз найпопулярнішими базами даних є: MySQL, SQL Server, PostgreSQL, Microsoft Access, MariaDB, Azure SQL.

PostgreSQL — це система керування базами даних корпоративного класу з відкритим вихідним кодом. Він підтримує як SQL, так і JSON для реляційних і нереляційних запитів для розширюваності та відповідності SQL. PostgreSQL підтримує розширені типи даних і функції оптимізації продуктивності, які доступні лише в дорогих комерційних базах даних, таких як Oracle і SQL Server.

Переваги PostgreSQL:

- може запускати динамічні веб-сайти та веб-програми як варіант стека LAMP;
- вихідний код вільно доступний під ліцензією з відкритим вихідним кодом;
- підтримує географічні об'єкти, тому його можна використовувати для служб на основі розташування та географічних інформаційних систем;
- низький рівень обслуговування та адміністрування як для вбудованого, так і для корпоративного використання.

Висновки до розділу 3

В даному розділі було проаналізовано та досліджено актуальні технології для розробки клієнтської та серверної частини застосунку. Були виділені сильні та слабкі сторони кожної з технологій, і в результаті порівняння визначені ті, які найкраще підходять для створення кваліфікаційної роботи.

Проведено порівняльний аналіз реляційних систем управління баз даних, в результаті якого обрано найбільш підходящу за функціоналом та продуктивністю для розробки вебпорталу.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБПОРТАЛУ З ПІДТРИМКОЮ ОНЛАЙН-КОМУНІКАЦІЇ КОРИСТУВАЧІВ

4.1 Програмна реалізація

Для створення авторизації на серверній частині був створений модуль `user`, який складається з `auth.middleware.js`, `user.decorator.ts`, `user.entity.ts`, `auth.guard.ts`, `user.controller.ts` та `user.service.ts`.

Middleware - код, який знаходиться між запитом та відповіддю на веб-сервері. Middleware може:

- виконати код;
- вносити зміни до об'єктів запиту та відповіді;
- повернути відповідь;
- продовжити цикл запит-відповідь.

```
@Injectable()
export class AuthMiddleware implements NestMiddleware {
  constructor(private readonly userService: UserService) {}

  async use(req: ExpressRequestInterface, res: Response, next: NextFunction) {
    if (!req.headers.authorization) {
      req.user = null;
      next();
      return;
    }

    const token = req.headers.authorization.split(' ')[1];
    try {
      const decode = verify(token, JWT_SECRET);
      req.user = await this.userService.findById(decode.id);
      next();
    } catch (error) {
      req.user = null;
      next();
    }
  }
}
```

Рисунок 4.1 – AuthMiddleware

Щоб створити належну схему бази даних для програми Nest.js, TypeORM підтримує створення сутності. **Entity** — це клас, який відображається в певній таблиці бази даних.

```
@Entity( options: { name: 'users' } )
export class UserEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column()
  email: string;

  @Column()
  username: string;

  @Column( options: { default: '' } )
  bio: string;

  @Column( options: { default: '' } )
  image: string;

  @Column( options: { select: false } )
  password: string;

  @BeforeInsert()
  async hashPassword() {
    this.password = await hash(this.password, 10);
  }
}
```

Рисунок 4.2 – UserEntity

Guard — це клас, анотований декоратором `@Injectable()`. Guard повинен імплементувати інтерфейс `CanActivate`.

```
@Injectable()
export class AuthGuard implements CanActivate {
  canActivate(context: ExecutionContext): boolean {
    const request = context.switchToHttp().getRequest();
    if (request.user) {
      return true;
    }

    throw new HttpException( response: 'Not authorized', HttpStatus.UNAUTHORIZED );
  }
}
```

Рисунок 4.3 – AuthGuard

Контролери - це те, що безпосередньо відповідає на кожен HTTP-запит, що надходить на сервер, тому кожен веб-запит приведе (якщо він буде маршрутизований) до нового екземпляру класу контролера.

```
@Controller()
export class UserController {
  constructor(private readonly userService: UserService) {}

  @Post( path: '/users')
  @UsePipes(new ValidationPipe())
  async createUser(
    @Body( property: 'user') createUserDto: CreateUserDto,
  ): Promise<UserResponseInterface> {
    const user = await this.userService.createUser(createUserDto);
    return this.userService.buildUserResponse(user);
  }

  @Post( path: '/users/login')
  @UsePipes(new ValidationPipe())
  async login(
    @Body( property: 'user') loginUserDto: LoginUserDto,
  ): Promise<UserResponseInterface> {
    const user = await this.userService.login(loginUserDto);
    return this.userService.buildUserResponse(user);
  }

  @Get( path: '/user')
  @UseGuards(AuthGuard)
  async currentUser(@User() user: UserEntity): Promise<UserResponseInterface> {
    return this.userService.buildUserResponse(user);
  }

  @Put( path: '/user')
  @UseGuards(AuthGuard)
  @UsePipes(new ValidationPipe())
  async updateUser(
    @User( dataOrPipes: 'id') id: number | string,
    @Body( property: 'user') updateUserDto: UpdateUserDto,
  ): Promise<UserResponseInterface> {
```

Рисунок 4.4 – UserController

Сервіси - служби, які відповідають за надання деяких даних, які можна повторно використовувати в програмі. Сервіси також можуть містити певну логіку або логіку валідації користувачів. По суті, сервіс — це клас, де ми займаємося бізнес-логікою. Репозиторії, з іншого боку, які викликаються в сервісі відповідають за доступ до бази даних.

```
@Injectable()
export class CommentService {
  constructor(
    @InjectRepository(CommentEntity)
    private readonly commentRepository: Repository<CommentEntity>,
    @InjectRepository(ArticleEntity)
    private readonly articleRepository: Repository<ArticleEntity>,
    @InjectRepository(UserEntity)
    private readonly userRepository: Repository<UserEntity>,
  ) {}

  async create(createCommentDto: CreateCommentDto, author: UserEntity) {
    const articleById = await this.articleRepository.findOne(conditions: {
      id: createCommentDto.articleId,
    });

    const comment = new CommentEntity();
    Object.assign(comment, source: {
      text: createCommentDto.text,
    });

    comment.article = articleById;
    comment.author = author;

    return await this.commentRepository.save(comment);
  }
}
```

Рисунок 4.5 – Приклад CommentService

Слід зазначити, що сервіси і репозиторії є тісно пов'язаними поняттями. Один сервіс може використовувати одне або кілька сховищ, щоб отримати різні дані та об'єднати їх разом.

Для керуванням глобальним станом застосунку на клієнтській частині використовувався стейт менеджер Redux. Store — це, в основному, звичайний об'єкт JavaScript, який дозволяє компонентам обмінюватися станом.

З точки зору Flux-архітектури, store зазвичай має кілька методів (action, dispatch, reducer).

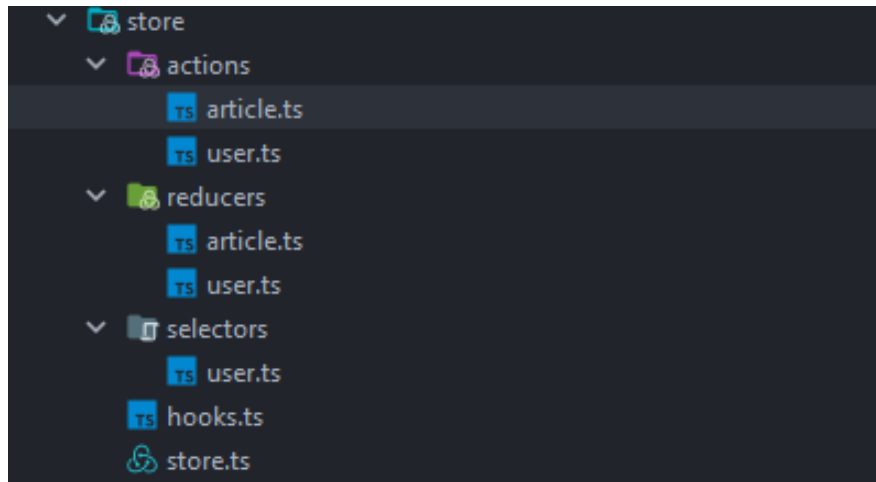


Рисунок 4.6 – Структура файлів Redux

Reducer — це чиста функція, яка приймає попередній state і action і повертає наступний стан (нову версію попереднього).

Дуже важливо, щоб reducer залишався чистою функцією. Список того, що ніколи не слід робити з ним:

- змінювати те, що надійшло в аргументах функції;
- виконувати будь-які побічні ефекти: отримувати доступ до API або переміщуватись по маршрутах застосунку;
- викликати нечисті функції, такі як Date.now() або Math.random().

```

    REMOVE_COMMENT: 'article/RemoveComment',
  };

  export const getArticle = createAsyncThunk(
    ArticleActions.GET_ARTICLE,
    {
      payloadCreator: async (slug: string, { dispatch: ... }) => {
        dispatch(setArticle({ state: null }));
        try {
          const article = await ArticleService.findOne(slug);
          dispatch(setArticle(article));
        } catch (error) {
          console.log(error);
        }
      }
    }
  );

```

Рисунок 4.7 – Приклад асинхронного action для отримання статті

Створення HTTP сервісів

Для того, щоб встановити зв'язок між клієнтом і сервером дуже зручно створювати сервіси, які зберігатимуть у собі методи, для відправки запитів на бекенд застосунку. Із використанням такого підходу дуже зручно в майбутньому звертатися до сервісу всередині застосунку, і отримувати дані з серверу.

```
const api = process.env.NEXT_PUBLIC_DOMAIN_API;

export class RoomService {
  static async findAll(): Promise<Room[]> {
    const { data } = await $api.get<RoomsResponse>(url: `${api}/rooms`);
    return data.data;
  }

  static async findAllMessages(roomId: number | string): Promise<RoomSingle> {
    const { data } = await $api.get<RoomResponse>(url: `${api}/rooms/${roomId}`);
    return data.data;
  }

  static async joinRoom(roomId: number, userId: number): Promise<boolean> {
    const { data } = await $api.post<boolean>(url: `${api}/rooms/${userId}`, data: {
      roomId,
      userId,
    });
    return data;
  }

  static async create(createRoomDto: CreateRoomDto): Promise<Room> {
    const { data } = await $api.post<CreateRoomResponse>(
      url: `${api}/rooms`,
      createRoomDto
    );
    return data.data;
  }
}
```

Рисунок 4.8 – RoomService

```
const Chat: FC = () => {
  const [rooms, setRooms] = useState<Room[] | null>(initialState: null);
  const { isAuth } = useTypedSelector(selector: (state: AppState) => state.user);

  useEffect(effect: () => {
    const getUserRooms = async () => {
      try {
        const rooms = await RoomService.findAll();
        setRooms(rooms);
      } catch (error) {
        console.log(error);
      }
    };

    getUserRooms();
  }, deps: []);
}
```

Рисунок 4.9 – Звернення до RoomService в компоненті Chat

Для відправки запитів використано бібліотеку Axios, яка дозволяє легко взаємодіяти з API у вебзастосунку. Хоча це також можна досягти за допомогою інших методів, таких як fetch або AJAX, Axios може надати більше функцій, які мають велике значення для застосунків, які використовують розробляються на React.

4.2 Діаграма класів

Діаграма класів (рис.4.10) - це статична структура, яка дає огляд програмної системи, відображаючи класи, атрибути, операції та їхні взаємозв'язки між собою. Ця діаграма містить назву класу, атрибути та операції в окремих призначених відсіках.

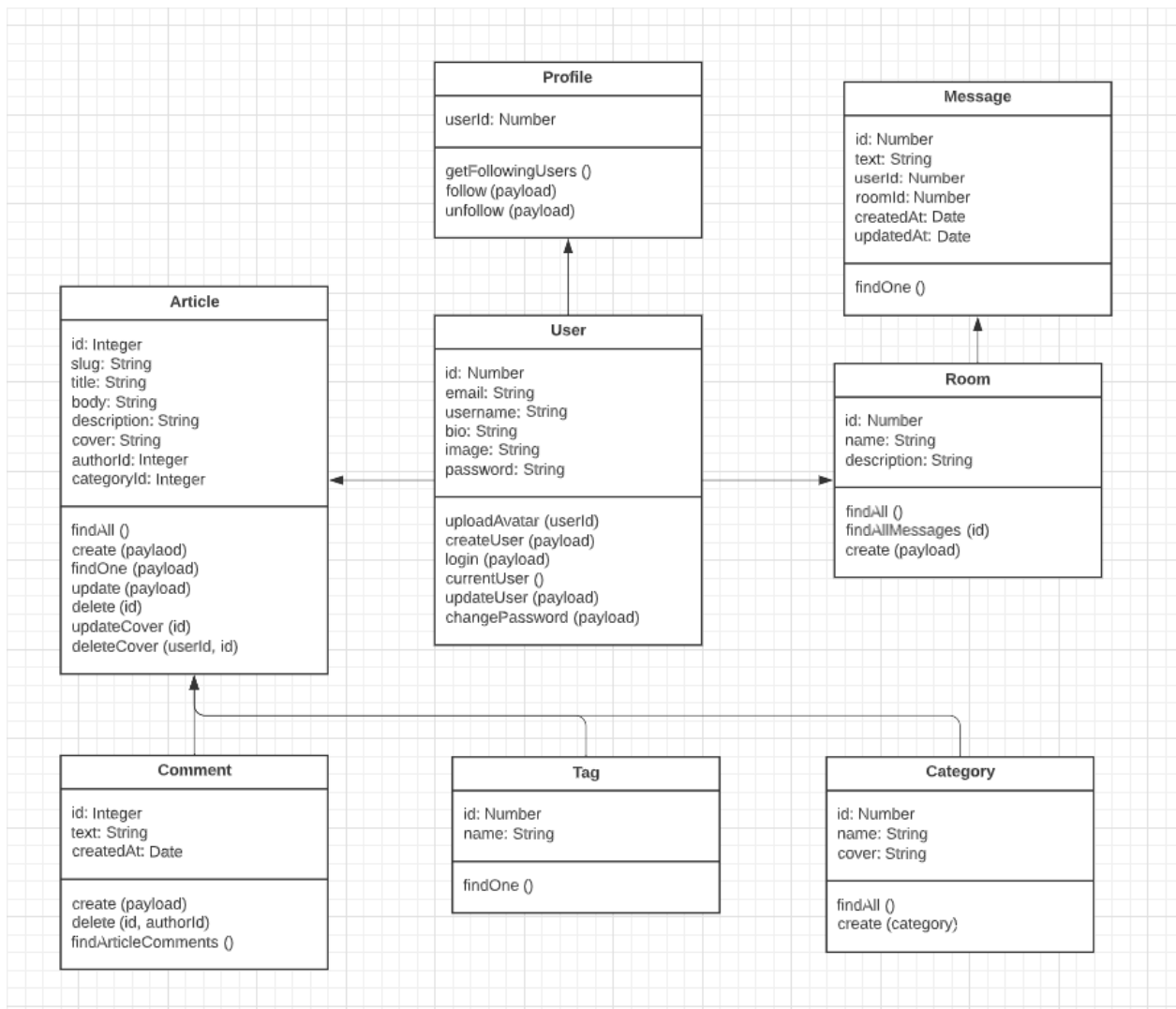


Рисунок 4.10 – Діаграма класів

Діаграма класів допомагає створити код для розробки програмного додатка.

Діаграма класів визначає типи об'єктів у системі та різні типи зв'язків, які існують між ними. Він дає високорівневий перегляд програми.

Цей метод моделювання може працювати майже з усіма об'єктно-орієнтованими методами. Клас може посилатися на інший клас. Клас може мати свої об'єкти або успадковувати від інших класів.

4.3 Тестування вебпорталу

Тестування є важливою практикою в розробці програмного забезпечення, яка допомагає створювати надійне та високоякісне програмне забезпечення та підвищує довіру команди до коду, роблячи програму більш гнучким і схильним до меншої кількості помилок під час введення або модифікації функцій.

Високоєфективні команди роблять тестування основною практикою в їх повсякденній рутині, і жодна функція не випускається до початку автоматизованих тестів. Деякі розробники навіть пишуть тести перед написанням функцій, дотримуючись процесу, який називається тестовою розробкою (TDD).

React Testing Library — це пакет для тестування JavaScript, який був створений спеціально для тестування компонентів React. Він моделює взаємодію користувачів із ізольованими компонентами та підтверджує їхні результати, щоб переконатися, що інтерфейс користувача веде себе правильно.

Кожного разу, коли тести запускають без веб-браузера, треба мати віртуальний DOM, щоб там відображати застосунок, взаємодіяти з елементами та спостерігати, чи віртуальний DOM веде себе так, як має (наприклад, змінювати ширину `div` при натисканні кнопки).

Огляд можливостей бібліотеки:

Simulate - повторний експорт з утиліти `Simulate react-dom / test-utils`.

Wait - дозволяє організувати очікування в тестах на невизначений період часу. Зазвичай для запитів API або анімації слід використовувати фіктивні об'єкти, але навіть при роботі з миттєво дозволеними угодами тести повинні чекати наступного тика циклу подій. Для цього чудово підходить метод очікування.

render: тут прихована вся суть бібліотеки. Насправді це досить проста функція. Він створює елемент `div` за допомогою `document.createElement`, а потім використовує `ReactDOM.render` для виведення даних до цього `div`.

unmount: проста обгортка навколо `ReactDOM.unmountComponentAtNode`, щоб відключити компонент (наприклад, щоб спростити тестування `componentWillUnmount`).

```

const Page = require('./page');

class UsersPage extends Page {
  get loadingTitle() {
    return $('#users-loading');
  }

  get usersList() {
    return $('#users-list');
  }

  get usersItems() {
    return browser.react$$('User');
  }

  async loadData() {
    try {
      await this.open();
      await this.loadingTitle.waitForDisplayed({ timeout: 2000 });
      await this.usersList.waitForDisplayed({ timeout: 2000 });
    } catch (e) {
      throw new Error('Error!');
    }
  }

  open() {
    return super.open('/users-test');
  }
}

module.exports = new UsersPage();

```

Рисунок 4.11 – Приклад тестування компонента з асинхронним завантаженням даних із сервера

getByLabelText: отримує елемент керування формою на основі його мітки.

getByPlaceholderText: Заповнювачі не дуже хороші альтернативи тегам, але якщо це має сенс у конкретній ситуації, ви можете використовувати їх.

getText: отримує елемент відповідно до його текстового вмісту.

getAltText: отримує елемент (подобається) за значенням його атрибута alt.

getTestId: отримує елемент за його атрибутом data-testid.

Кожен з цих допоміжних методів `get` виводить інформативне повідомлення про помилку, якщо елемент не знайдено. Крім того, існує набір подібних методів запити (наприклад, `queryByText`). Замість того, щоб давати помилку за відсутності елемента, вони повертають `null`, що може бути корисно, якщо вам потрібно перевірити DOM на відсутність елемента.

4.3 Інструкції користувача вебпорталу

Для використання застосунку користувачу потрібен браузер, та стабільне інтернет-з'єднання. Після першого потрапляння на сайт, користувач не авторизований і має роль гість. Для реєстрації необхідно натиснути на кнопку Sign up у верхній частині вебпорталу.

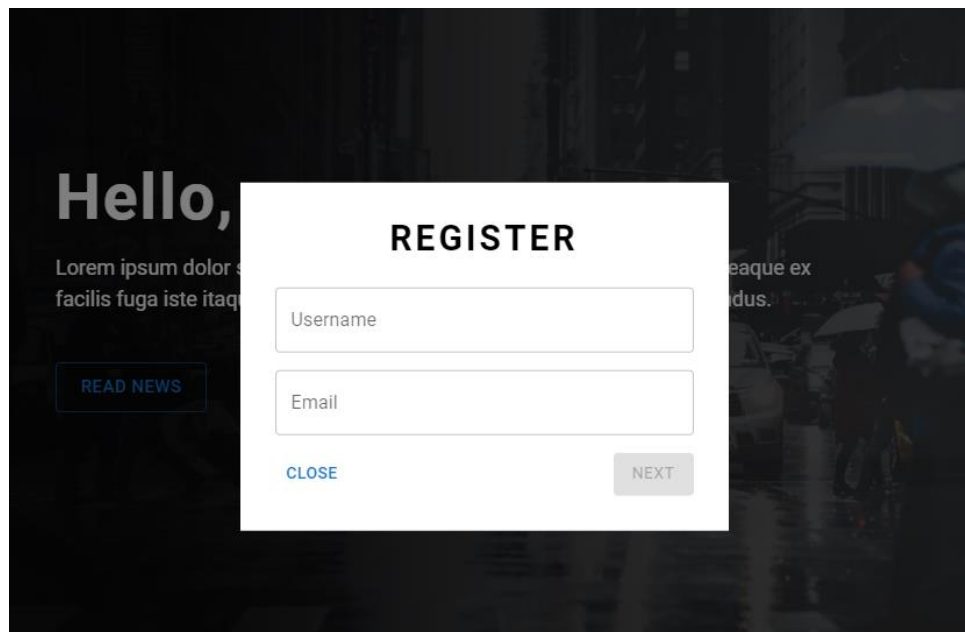


Рисунок 4.12 – Інтерфейс реєстрації

У разі вдалої реєстрації користувач отримає доступ до повного функціоналу на порталі, а також зможе зайти до свого особистого кабінету. В особистому кабінеті знаходяться форми редагування персональних даних, зміна пароля, а також можливість переглянути список створених статей.

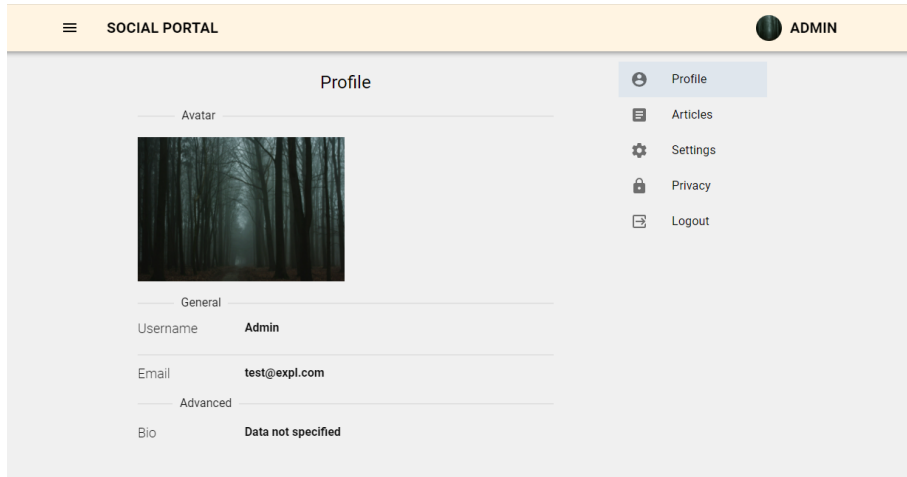


Рисунок 4.13 – Особистий кабінет

Форма створення посту новин розташована на окремій сторінці і має обов'язкові поля для введення (назва, короткий опис, контент і категорія).

Також є можливість відредагувати створену новину, дана форма створена за аналогом до попередньої.

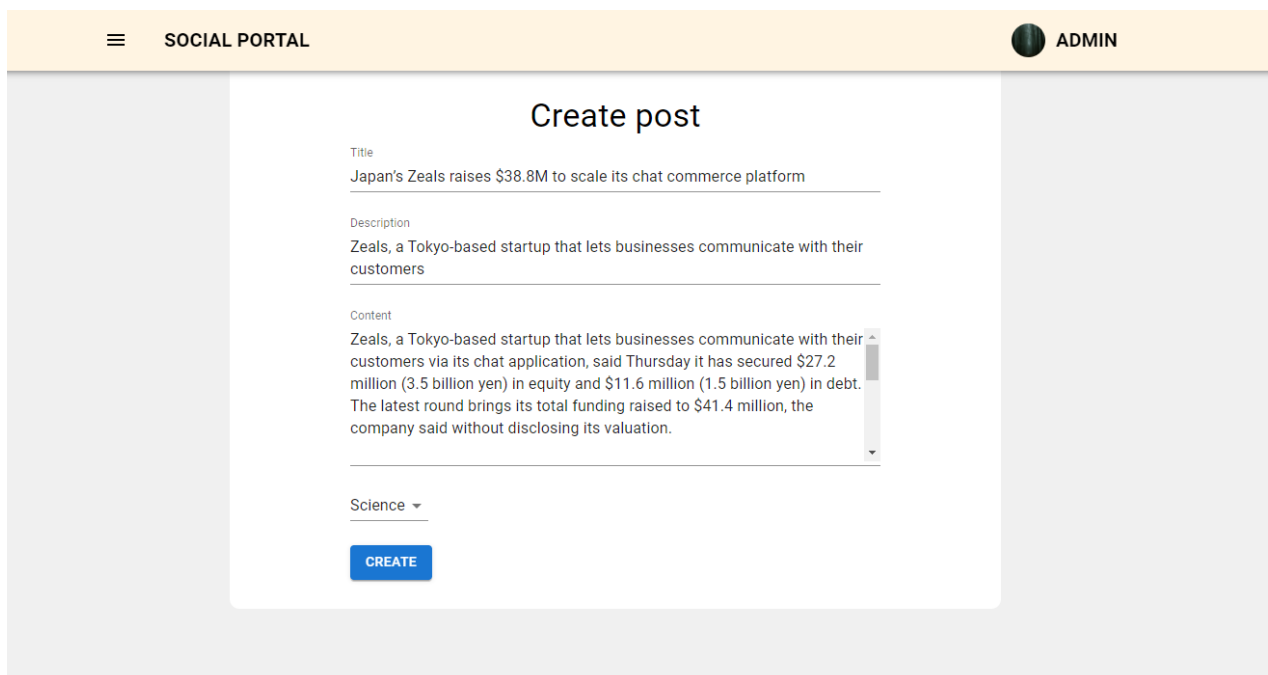


Рисунок 4.14 – Особистий кабінет

Всі створені пости відображається в головній стрічці новин, де користувач може перейти в кожен, щоб докладно прочитати, або залишити коментар.

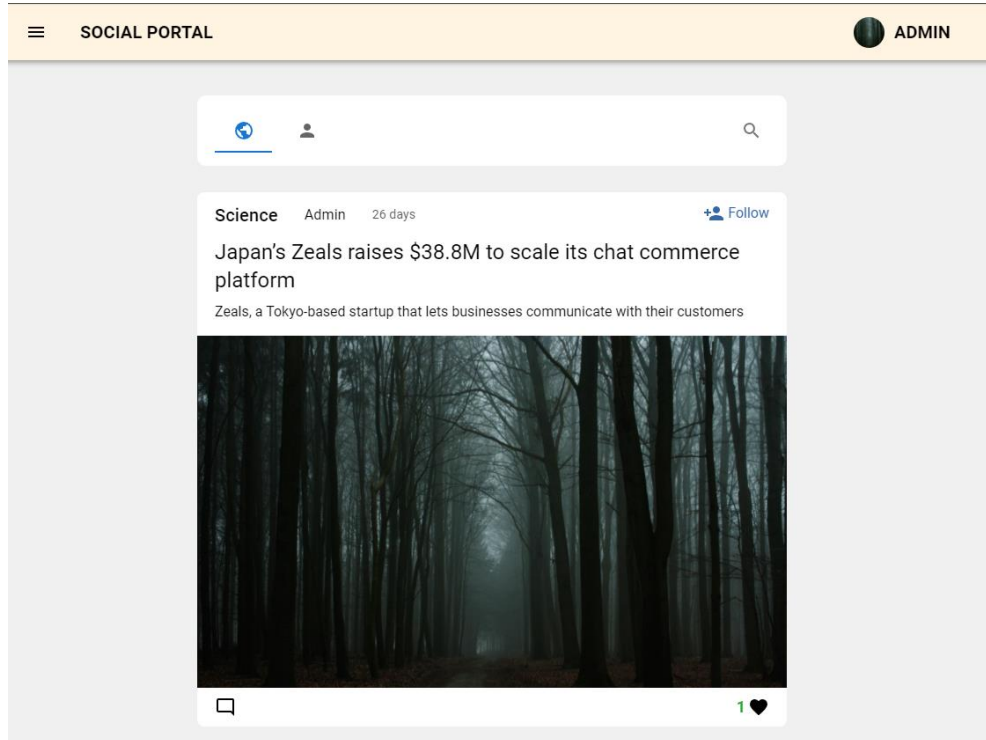


Рисунок 4.15 – Стрічка новин

У кожної новини відображаються, категорія, назва, головна картинка, кількість коментарів та позитивних оцінок.

Висновки до розділу 4

У цьому розділі було реалізовано вебпортал, з використанням технологій для серверної та клієнтської частини, дотримуючись вимог до застосування, спроектованих моделям бази даних та користувацьких сценаріїв використання системи.

Було створено посібник користувача, завдяки якому можна дізнатися функціонал системи, і докладніше ознайомитися з його інтерфейсом. Також проведено тестування клієнтської та серверної частини, внаслідок яких несправностей виявлено не було.

ВИСНОВКИ

В результаті виконання дипломної роботи було досягнуто поставлену мету, щодо підвищення зручності обміну повідомленнями, новинами та думками з іншими користувачами за рахунок розробки вебпорталу з підтримкою онлайн-комунікації користувачів.

Проаналізовано та досліджено актуальні технології для розробки клієнтської та серверної частини застосунку. Були виділені сильні та слабкі сторони кожної з технологій, і в результаті порівняння визначені ті, які найкраще підходять для створення кваліфікаційної роботи.

Визначено головні функціональні вимоги, в які входить визначення ролей користувачів, створення макету інтерфейсу, виділення необхідного функціоналу, який потрібний для виконання поставленої мети.

Спроектовано та розроблено застосунок обміну повідомленнями з використанням сучасних патернів розробки, та побудови інтерфейсів. Спроектовано архітектуру системи, яка зручно дозволяє розширювати застосунок, або додавати зовнішні API сервіси, створено систему користувачів та ролей. Розроблений вебпортал має можливість перегляду, створення та редагування статей, створення коментарів, оцінювання, можливість перейти до персонального кабінету та змінити користувацькі дані, підписуватись на користувачів та надсилати їм повідомлення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. A progressive Node.js framework for building efficient, reliable and scalable server-side applications: веб-сайт. URL: <https://nestjs.com> (дата звернення 03.04.2022);
2. React – A JavaScript library for building user interfaces: веб-сайт. URL: <https://reactjs.org> (дата звернення 13.03.2022);
3. Typescript is JavaScript with syntax for types: веб-сайт. URL: <https://www.typescriptlang.org/> (дата звернення 19.04.2022);
4. JSON Web Token: веб-сайт. URL: https://en.wikipedia.org/wiki/JSON_Web_Token (дата звернення 21.03.2022);
5. MUI: The React component library you always wanted: веб-сайт. URL: <https://mui.com> (дата звернення 11.03.2022);
6. All about social networks development: веб-сайт. URL: <https://intellect.ml/all-about-social-networks-development-prospects-2051> (дата звернення 17.02.2022);
7. Node.js Documentation. URL: <https://nodejs.org/api/documentation.html> (дата звернення 02.04.2022);
8. Setting up Jest and React Testing Library. URL: <https://blog.sapegin.me/all/react-testing-3-jest-and-react-testing-library> (дата звернення 22.04.2022);
9. Advanced Javascript Design Patterns. URL: <https://medium.com/dhiwise/advanced-javascript-design-patterns-6812f3286585> (дата звернення 01.03.2022);
10. What is PostgreSQL? Introduction, Advantages & Disadvantages. URL: <https://www.guru99.com/introduction-postgresql.html> (дата звернення 10.04.2022);

11. Build a NestJs App with TypeORM and Prostagres. URL: <https://betterprogramming.pub/nest-js-project-with-typeorm-and-postgres-ce6b5afac3be> (дата звернення 07.04.2022);
12. ObserverAPIs in JavaScript. URL: https://developer.mozilla.org/en-US/docs/Web/API/Intersection_Observer_API (дата звернення 27.03.2022);
13. Getting Started with Redux Toolkit. URL: <https://redux-toolkit.js.org/introduction/getting-started> (дата звернення 22.03.2022);
14. Meet WebStorm. URL: <https://www.jetbrains.com/help/webstorm/meet-webstorm.html> (дата звернення 21.04.2022);
15. The React Framework for Production. URL: <https://nextjs.org/docs/getting-started> (дата звернення 15.03.2022);
16. Promise based HTTP client for the browser and node.js. URL: <https://axios-http.com> (дата звернення 26.02.2022);
17. PostgreSQL The World's Most Advanced Open Source Relational Database. URL: <https://www.postgresql.org> (дата звернення 11.02.2022);
18. TypeORM – Amazing ORM for TypeScript and JavaScript. URL: <https://typeorm.io/> (дата звернення 13.04.2022);
19. Form validation with Yup. URL: <https://www.sanity.io/blog/sanity-studio-v3-developer-preview> (дата звернення 30.04.2022);
20. Composition vs Inheritance. URL: <https://reactjs.org/docs/composition-vs-inheritance.html> (дата звернення 10.02.2022);

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРІВ

**ВЕБПОРТАЛ З ПІДТРИМОЮ ОНЛАЙН-КОМУНІКАЦІЇ
КОРИСТУВАЧІВ**

СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ

**Аналіз умов праці, технологічних процесів, апаратури і обладнання на
робочому місці**

Спеціальність «Інженерія програмного забезпечення»

121 – ККРБ.1 – 408.21810822

Студент

_____ Є. Ю. Поддубний

підпис

«_____» _____ 2022 р.

Консультант: канд. техн. наук, доцент

_____ А. О. Алексеева

підпис

«_____» _____ 2022 р.

Миколаїв – 2022

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	3
ВСТУП.....	4
1 ТЕХНІКА БЕЗПЕКИ ПІД ЧАС РОБОТИ З ПК	6
2 АНАЛІЗ УМОВ РОБОТИ ЗА ШУМОМ.....	9
3 АНАЛІЗ УМОВ ПРАЦІ ЗА ОСВІТЛЕННЯМ	10
4 ПОЖЕЖНА БЕЗПЕКА	13
Висновки до розділу	15
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	16

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

дБА – децибел

ПК – персональний комп'ютер

ДСН – державні санітарні норми

ЛК – одиниця виміру люкс

МОЗ – міністерство охорони здоров'я

ДСН – державні санітарні норми

ДСанПН – державні санітарні правила та норми

МЮУ – Міністерство юстиції України

ВСТУП

У зв'язку з розвитком інформаційних технологій ця тема є дуже актуальною, оскільки кожне підприємство впроваджує їх для полегшення та автоматизації роботи за допомогою технічних пристроїв. Як приклад, можна навести використання ноутбуків та персональних комп'ютерів, які мають на меті збір, аналіз та обмін інформацією всередині та за межами підприємств або установ. Використання комп'ютера полегшує процеси на підприємствах.

Охорона праці сьогодні є однією з головних систем на підприємствах різного спрямування будь-яких форм власності. Дотримуючись елементарних правил безпечного виконання робіт та вимог охорони праці, роботодавець може досягти істотного зниження виробничого травматизму, практично до 95%. Відсоток ймовірності, що залишився, складе людський фактор, від якого складно піти. Саме таке відсоткове співвідношення випадків травматизму виникають через недотримання вимог охорони праці як з боку роботодавця, так і з боку працівника.

Взявши до уваги переваги інтеграції інформаційних технологій за допомогою технічних пристроїв, треба ще враховувати також і недоліки їх інтеграції у робочий процес. Головним недоліком є загострення суспільного та власного здоров'я, яке потребує створення та удосконалення існуючих робочих місць, які матимуть більш сприятливі до здоров'я умови робочого місця. Також треба виділити проведення заходів спрямованих на запобігання розвитку негативних наслідків впливу персонального комп'ютера на здоров'я користувача.

При роботі у сфері охорони праці слід керуватися насамперед Законом України про охорону праці. Крім того, Конституцією України, Кодексом законів про працю України, Законом України «Про загальнообов'язкове державне страхування від нещасного випадку на виробництві та професійного захворювання, що спричинили втрату працездатності», а також інших нормативно-правових актів, що регулюють відносини у сфері охорони праці, а також чинних міжнародних договорів, за згодою Верховної Ради України.

Метою даного розділу є аналіз умов праці, технологічних процесів, апаратури і обладнання на робочому місці програміста. На основі такого аналізу визначаються небезпечні ділянки, співвідношення умов праці на робочому місці до нормативних даних, можливі аварійні ситуації і розробляються заходи щодо їх усунення або обмеження наслідків.

1 ТЕХНІКА БЕЗПЕКИ ПІД ЧАС РОБОТИ З ПК

При виконанні робіт на комп'ютерах необхідно дотримуватись вимог загальної та даної інструкції з охорони праці.

До самостійної роботи на комп'ютерах допускаються особи, які пройшли медичний огляд, навчання по професії, вступний інструктаж з охорони праці та первинний інструктаж з охорони праці на робочому місці. В подальшому вони проходять повторні інструктажі з охорони праці на робочому місці один раз на півріччя, періодичні медичні огляди один раз на два роки.

Основним обладнанням робочого місця користувача комп'ютера є монітор, системний блок та клавіатура.

Робочі місця мають бути розташовані на відстані не менше 1,5 м від стіни з вікнами, від інших стін на відстані 1м, між собою на відстані не менше 1,5 м. Відносно вікон робоче місце доцільно розташовувати таким чином, щоб природне світло падало на нього збоку, переважно зліва.

Фільтри з металевої або нейлонової сітки використовувати не рекомендується, тому що сітка спотворює зображення через інтерференцію світла. Найкращу якість зображення забезпечують скляні поляризаційні фільтри. Вони усувають практично всі відблиски, роблять зображення чітким і контрастним.

При роботі з текстовою інформацією (в режимі введення даних та редагування тексту, читання з екрану) найбільш фізіологічним правильним є зображення чорних знаків на світлому (чорному) фоні.

Монітор повинен бути розташований на робочому місці так, щоб поверхня екрана знаходилася в центрі поля зору на відстані 400-700 мм від очей користувача. Рекомендується розміщувати елементи робочого місця так, щоб витримувалася однакова відстань очей від екрана, клавіатури, тексту.

Для нейтралізації зарядів статичної електрики в приміщенні, де виконується робота на комп'ютерах, в тому числі на лазерних та світлодіодних принтерах, рекомендується збільшувати вологість повітря за допомогою

кімнатних зволожувачів. Не рекомендується носити одяг з синтетичних матеріалів.

Згідно статті 18 Закону України «Про охорону праці» працівник зобов'язаний:

- знати і виконувати вимоги нормативних актів про охорону праці, правила поведіння з устаткуванням та іншими засобами виробництва, користуватися засобами колективного та індивідуального захисту;

- дотримуватись зобов'язань щодо охорони праці, передбачених колективним договором та правилами внутрішнього трудового розпорядку підприємства;

- співробітничати з власником у справі організації безпечних і нешкідливих умов праці, особисто вживати посильних заходів щодо усунення будь-якої виробничої ситуації, яка створює загрозу його життю чи здоров'ю, або людей, які його оточують, повідомляти про небезпеку свого безпосереднього керівника або іншу посадову особу.

Вимоги безпеки перед початком роботи:

- увімкнути систему кондиціонування в приміщенні;
- перевірити надійність встановлення апаратури на робочому столі.

Повернути монітор так, щоб було зручно дивитися на екран - під прямим кутом (а не збоку) і трохи зверху вниз, при цьому екран має бути трохи нахиленим, нижній його край ближче до оператора;

- перевірити загальний стан апаратури, перевірити справність електропроводки, з'єднувальних шнурів, штепсельних вилок, розеток, заземлення захисного екрана;

- відрегулювати освітленість робочого місця;

- відрегулювати та зафіксувати висоту крісла, зручний для користувача нахил його спинки;

– приєднати до системного блоку необхідну апаратуру. Усі кабелі, що з'єднують системний блок з іншими пристроями, слід вставляти та виймати при вимкненому комп'ютері;

– ввімкнути апаратуру комп'ютера вимикачами на корпусах в послідовності: монітор, системний блок, принтер (якщо передбачається друкування);

– відрегулювати яскравість свічення монітора, мінімальний розмір світної точки, фокусування, контрастність. Не слід робити зображення надто яскравим, щоб не втомлювати очей.

Забороняється самостійно ремонтувати апаратуру. Ремонт апаратури здійснюється спеціалістами з технічного обслуговування комп'ютера, 1 раз на півроку повинні відкривати процесор і вилучати пирососом пил і бруд, що накопичилися; класти будь-яку предмети на апаратуру комп'ютера; закривати будь-чим вентиляційні отвори апаратури, що може призвести до її перегрівання і виходу з ладу.

Для досягнення найбільш чистих, з високою розподільністю зображень і щоб не зіпсувати апарат, має використовуватися папір, вказаний в інструкції до принтера. При заминанні паперу потрібно відкрити кришку і обережно витягнути лоток з папером.

Згідно з інструкцією фірми-виробника потрібно дотримуватися правил зберігання картриджа.

Забороняється:

- зберігати картриджі без упаковки;
- ставити картриджі вертикально;
- перевертати картридж етикеткою донизу;
- відкривати кришку валика і доторкатися до нього;
- самому заповнювати використаний картридж.

2 АНАЛІЗ УМОВ РОБОТИ ЗА ШУМОМ

Більшість виробничих процесів в будівництві супроводжується дією на працюючих шуму, що виникає при роботі машин, енергетичних установок.

Шум - сукупність звуків різної інтенсивності і частоти. Під шумом розуміють усі неприємні та небажані звуки чи їх сукупність, які заважають нормально працювати, сприймати потрібні звукові сигнали, відпочивати. Нині добре відомо, що шуми шкідливо та негативно впливають на здоров'я людей, знижують їх працездатність, спричиняють захворювання органів слуху (глухоту), ендокринної, нервової, серцево-судинної систем (гіпертонія), підвищують кров'яний тиск, кори головного мозку, погіршення пам'яті, що погіршує сприйняття звукових та світлових сигналів небезпеки, тому є шкідливим фактором і обумовлює зростання травматизму. Шум - це одна з форм фізичного (хвильового) забруднення природного середовища, адаптація до якого організмів практично неможлива. Тому він належить до серйозних забруднювачів, які мають контролюватися та обмежуватися згідно зі спеціальними законами. Шум підвищує втомленість робітника, знижує його працездатність і увагу до небезпеки. Шум як стрес-чинник є загально біологічним подразником, негативно впливає на всі органи і системи організму. В разі тривалого систематичного впливу шуму може виникнути патологія з переважним ураженням слуху, центральної нервової і серцево-судинної систем. Вплив шуму на організм умовно поділяють на специфічний, що спричиняє зміни в органі слуху, і неспецифічний - з боку інших органів і систем. Шум є однією з найбільш частих причин зниження слуху.

Отримані результати розрахунку порівнюється з допустимим значенням рівня шуму даного робочого місця. Якщо результати розрахунку вище допустимого значення рівня шуму, необхідні спеціальні заходи зі зниження шуму.

Рівні звукового тиску джерел шуму, що діють на оператора на його робочому місці представлені в таблиці 1.1.

Таблиця 1.1 – рівні звукового тиску різних джерел.

Джерело шуму	Рівень шуму, дБА
Жорсткий диск	40
Вентилятор	45
Клавіатура	10
Принтер	42
Сканер	45

3 АНАЛІЗ УМОВ ПРАЦІ ЗА ОСВІТЛЕННЯМ

Кожен вид діяльності, пов'язаний із необхідністю розрізнення того чи іншого об'єкта, вимагає певного рівня освітленості на тій ділянці, де ця діяльність здійснюється. Зазвичай що сильніше утруднене зорове сприйняття, то вище має бути середній рівень освітленості. Недостатня освітленість робочої зони та знижена контрастність викликають напруженість зорового аналізатора, що, своєю чергою, може призвести до порушень зору. В умовах, коли загальна освітленість відсутня, виконання робіт неможливе без індивідуальних головних або ручних світильників.

Виробничі будівлі та робочі майданчики підприємств висвітлюються природним світлом небосхилу (прямим або відбитим), штучним світлом, а також комбінованим. Залежно від джерела освітлення, конструктивного виконання та функціонального призначення розрізняють такі види освітлення.

Видиме випромінювання – ділянка спектра електромагнітних коливань у діапазоні довжин хвиль від 380-770 нм ($1 \text{ нм} = 10^{-9} \text{ м}$), реєстрованих людським оком.

Штучне освітлення передбачається у приміщеннях, у яких відчувається нестача природного світла, а також для освітлення приміщення у ті години доби, коли природне освітлення відсутнє.

Робочі приміщення у яких працюють робітники та при цьому використовуються персональні комп'ютери або ноутбуки, повинні відповідати ряду вимог, які повинні перевіряти відповідні служби для того, щоб вимоги

охорони праці були дотримані. Нижче наведено частину вимог з документу про правила охорони праці під час роботи на підприємствах [3]:

- площа приміщення не повинна бути менша ніж 6 метрів на 1 робоче місце;
- місця робітників повинні бути розташовані не менше 1 метра від стіни з вікном і 1,4 метра від звичайної стіни;
- відстань між комп'ютерами має бути не менша ніж 1,2 метри;
- відстань між тильною стороною комп'ютера та екранним пристроєм має бути не менша 2,5 метрів;
- робочі місця заборонено облаштовувати у підвальних приміщеннях або цокольних приміщеннях будинків;
- під час обладнання приміщень забороняється використовувати полімерні матеріали, що виділяють у повітря шкідливі хімічні речовини;
- підлога має бути з рівною, матовою, неслизькою, з антистатичними властивостями [4].

Існують і вимоги до виробничого приміщення щодо параметрів мікроклімату та вібрації. Ці вимоги допомагають зменшити вплив негативних факторів на здоров'я працівника.

Температурні показники повітря в робочій зоні не повинні перевищувати нормальних значень оптимальної температури для даної категорії робіт, зазначених у таблиці 1, по висоті і по горизонталі.

Таблиця 1.2 – Норми мікроклімату для приміщень [5]

Пора року	Категорія робіт	Температура повітря, °С, не більше	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодна	Легка - 1а	22-24	40-60	0,1
	Легка - 1б	21-23	40-60	0,1
Тепла	Легка - 1а	23-25	40-60	0,1
	Легка - 1б	22-24	40-60	0,2

Категорія 1а включає в себе сидячу діяльність, яка не вимагає фізичної активності.

Категорія 1б включає в себе сидячу, стоячу діяльність або ходьбу, що вимагає спеціальної фізичної активності.

Відомо, що шум несприятливо діє на слуховий аналізатор та інші органи й системи організму людини. Визначальне значення щодо такої дії має інтенсивність шуму, його частотний склад, тривалість щоденного впливу, індивідуальні особливості людини, а також специфіка виробничої діяльності.

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях, обладнаних ВДТ і ПК визначені ДСанПіН 3.3.2-007-98 [3].

Основними заходами та засобами боротьби з шумом є:

- зниження рівнів шуму в джерелі його утворення (застосовується, як правило, в процесі проектування);
- використання звукопоглинаючих та звукоізолюючих засобів;
- раціональне планування виробничих приміщень та робочих місць.

Оскільки зовнішні шуми (вулиця, суміжні приміщення) також можуть негативно впливати на функціональний стан операторів ВДТ, то стіни приміщень, в яких розташовані комп'ютеризовані робочі місця бажано облицювати звукопоглинаючими матеріалами. Однак доцільність їх застосування повинна бути обґрунтована спеціальними інженерно-акустичними розрахунками. Звукопоглинаюче облицювання стін (іноді й стелі) необхідно здійснювати матеріалами, що мають максимальний коефіцієнт звукопоглинання в межах частот 31,5-8000 Гц і дозволені для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду.

Для зниження вібрації обладнання, пристрої, пристосування необхідно встановлювати на спеціальні амортизуючі прокладки, передбачені нормативними документами.

4 ПОЖЕЖНА БЕЗПЕКА

Основними напрямками забезпечення пожежної безпеки є усунення умов виникнення пожежі та мінімізація її наслідків. Пожежа виникає за одночасної наявності горючої речовини, джерела займання та окисника (кисню, повітря), що разом утворюють горюче середовище. Якщо вилучити або заблокувати будь-який із цих чинників, то пожежі не буде. На цьому ґрунтуються основні напрями попередження пожеж та способи пожежогасіння.

У приблизно 90% випадків до пожеж призводять: необережне поводження з вогнем; порушення правил монтажу та експлуатації електроустаткування і побутових електроприладів; порушення правил монтажу та експлуатації приладів опалення і теплогенеруючих установок; підпали; пустощі дітей із вогнем; несправність виробничого устаткування. Основні нормативні документи, що регламентують роботу щодо забезпечення пожежної безпеки об'єктів: Закон України "Про пожежну безпеку", стандарти, будівельні норми та правила, правила пожежної безпеки та інше. Обов'язки керівника підприємства та інших посадових осіб щодо забезпечення пожежної безпеки об'єкту та окремих ділянок виробництва.

Порядок створення та роботи пожежно-технічної комісії. Типове положення про пожежно-технічні комісії.

Порядок організації та роботи добровільної пожежної дружини. Положення про добровільні пожежні дружини. Обов'язки членів добровільних пожежних дружин щодо запобігання пожежам та їх гасіння. Пільги та заохочення, встановлені для них.

Кримінальна, адміністративна, матеріальна та дисциплінарна відповідальність громадян, посадових та юридичних осіб за порушення вимог пожежної безпеки та виникнення пожежі.

Коротка характеристика виробництва та пожежна небезпека технологічного процесу, сировини, готової продукції, агрегатів, установок тощо.

Основні причини пожеж: порушення технологічних регламентів та несправність виробничого обладнання, іскри електрогазозварювальних робіт та необережне поводження з вогнем, іскри котелень та інших установок, порушення правил користування інструментами та електронагрівальними приладами. Заходи пожежної безпеки, яких необхідно дотримуватися перед початком роботи, під час роботи та після її закінчення з метою запобігання пожежам.

Утримання території підприємства, протипожежні розриви, джерела протипожежного водопостачання, протипожежний режим на об'єкті.

Основні вимоги пожежної безпеки у будівлях та приміщеннях, при експлуатації електрообладнання, опалювальних приладів, систем вентиляції, при проведенні електрогазозварювальних, паяльних та інших вогневих робіт, при фарбуванні, знежиренні та миття виробів та обладнання.

Вимоги пожежної безпеки в лабораторіях, архівах, складських приміщеннях, гаражах, у складах із зберігання хімічних речовин, паливно-мастильних матеріалів, при роботі з пожежонебезпечними та вибухопожежонебезпечними матеріалами, у приміщеннях з масовим перебуванням людей (клубах, поліклініках, їдалень тощо).

Основні вимоги щодо утримання шляхів евакуації, автоматичних систем пожежогасіння та автоматичної пожежної сигналізації.

Призначення та місцезнаходження на об'єкті протипожежного обладнання та інвентарю (вогнегасники, внутрішні пожежні крани, бочки з водою, ящики з піском, стаціонарні установки пожежогасіння). Загальні уявлення про спринклерне та дренчерне обладнання, автоматичну пожежну сигналізацію, вуглекислотні, порошкові, газові та інші установки пожежогасіння

Порядок утримання на об'єкті засобів пожежогасіння влітку та взимку.
Правила використання вогнегасних засобів, протипожежного інвентарю та обладнання для пожежогасіння.

Дії працівників із прибуття пожежних підрозділів (надання допомоги у прокладанні рукавних ліній, участь у евакуації матеріальних цінностей та виконання інших робіт з розпорядження керівника гасіння пожежі).

Розслідування та облік пожеж, розробка заходів щодо запобігання пожеж та загибелі людей на них.

Висновки до розділу

Під час виконання спеціальної частини з охорони праці було проаналізовано умови праці людини на підприємствах та в установах, а також роботу при користуванні електронними пристроями.

У ході роботи було зазначено сприятливі умови для роботи над вебзастосунком без втрати та погіршення загального самопочуття та стану здоров'я. Був проведений аналіз стосовно пожежної безпеки та техніки безпеки стосовно використання ПК.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Закон України за 25 квітня 2018р №508/31960 про затвердження вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text> (дата звернення: 28.05.2022).
2. Типовий тематичний план і прикладна програма навчання посадних осіб і фахівців з питань охорони праці. URL: https://forca.com.ua/instrukcii/ohorona-praci/obuchenie-po-voprosam-ohrany-truda-pozharnoi-bezopasnosti-i-tehnicheskoi-eksploatatsii_8.html (Дата звернення: 13.06.2022).
3. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. URL: <https://www.sop.com.ua/article/758-vimogi-shchodo-bezpeki-ta-zahistu-zdorovya-pratsvnikv-pd-chas-roboti-z-ekrannimi-pristroyami> (дата звернення: 29.05.2022).
4. Порядок проведення медичних оглядів працівників певних категорій. URL: <https://zakon.rada.gov.ua/laws/show/z0846-07#Text> (дата звернення: 28.05.2022).
5. Санітарні норми мікроклімату виробничих приміщень. URL: <https://www.sop.com.ua/article/ru/225-qqq-16-m4-13-04-2016-sanitarnye-normy-mikroklimata-proizvodstvennyh-pomeshcheniy> (Дата звернення: 02.06.2022).