МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЛЕСІ УКРАЇНКИ
Кафедра комп’ютерних наук та кібербезпеки

На правах рукопису

КОНІЩУК ОЛЕКСАНДР МИХАЙЛОВИЧ

COLLABPAINT: РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ МАЛЮВАННЯ ОНЛАЙН

Спеціальність: 122 Комп’ютерні науки
Освітньо-професійна програма: Комп’ютерні науки та інформаційні технології
Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр»

Науковий керівник:
ОНИЩУК ОКСАНА ОЛЕКСАНДРІВНА,
Доцент кафедри комп'ютерних наук та кібербезпеки,

РЕКОМЕНДОВАНО ДО ЗАХИСТУ
Протокол №_____
засідання кафедри комп’ютерних наук та кібербезпеки від
_____ 2025 р.
Завідувач кафедри

(_____) Гришанович Т. О.

ЛУЦЬК 2025

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ТА ТЕРМІНІВ

IDE	– Integrated Development Environment, Інтегроване середовище розробки
ТЗ	– Технічне Завдання
Agile	– Це методологія Для Розробки ПЗ, Який Робить всю працю Над Проектом Послідовною, Гнучкою Та Адаптованою До Змін.
CSS	–Мова Для Стилізації Веб-сторінок. Cascading Style Sheets
Node.js	– Це Середовище Виконання З Відкритим Вихідним Кодом. Однопоточне Кросплатформове. Використовується Також Саме Для Запуску Вебдодатків На Javascript, Поза Браузером Клієнта.
FireBase	– хмарний бекенд-сервіс, що використовується для зберігання бази даних та синхронізації в режимі реального часу
JS	– JavaScript – Мова Програмування Використовується Для Анімації Та Надання Динамічності Веб-сторінкам
WebSockets	– Це Протокол, Що Призначений Для Обміну Інформацією Між Браузером Та Вебсервером В Режимі Реального Часу
WebRTC	– Інтернет-Протокол Із Відкритим Кодом, Призначений Для Організації Голосового Та Відеозв'язку Через Інтернет У Режимі Реального Часу.
CORS	– Cross-Origin Resource Sharin — Механізм Безпеки Сучасних Веб Браузерів, Дозволяє Вебсторінкам Використати Дані, Які Знаходяться На Інших Доменах.
Сокет	– Програмний Інтерфейс, Який Забезпечує Обмін Даними Між Процесами.
Фреймворк	– Інструмент Або Основа Для Створення Та Підтримки Веб-Додатків.
Git	– Система Контролю Версій, Що Дозволяє Зберігати Та Керувати Кодом У Віддалених Сховищах.

ВСТУП

Актуальність теми**.** У сучасному ландшафті цифрових інструментів, веб-додатки для спільної роботи стали критично важливою частиною сучасної комунікації, командної роботи та креативності. Зі зростанням попиту на віддалену взаємодію, особливо серед творчих спільнот, навчальних закладів та команд розробників, на перший план вийшли додатки, що дозволяють співпрацю в режимі реального часу. Серед них браузерні платформи для спільного малювання пропонують потужне, але доступне рішення для візуальної комунікації, мозкового штурму, створення прототипів та спільної творчості.

На відміну від традиційних графічних редакторів, ці інструменти не потребують встановлення чи високоякісного обладнання. Користувачі можуть малювати, коментувати та взаємодіяти з іншими миттєво, використовуючи лише веб-браузер. Це особливо актуально в епоху хмарних сервісів та віртуальної командної роботи, де простота та миттєва доступність важливіші, ніж будь-коли. CollabPaint відповідає на цей попит, пропонуючи середовище, де кілька користувачів можуть одночасно малювати , спілкуватися в чаті в режимі реального часу та взаємодіяти на спільному полотні за допомогою інтуїтивно зрозумілих інструментів та швидкої синхронізації.

Мета роботи**.** Метою цієї роботи було розробити та впровадити сучасний веб-додаток для спільного малювання під назвою CollabPaint, який дозволяє користувачам малювати та спілкуватися в режимі реального часу. Метою було створити платформу, яка є не лише багатофункціональною та адаптивною, але й доступною на різних пристроях, надаючи пріоритет зручності використання та безперебійній співпраці.

Завдання**.** Для досягнення поставленої мети були виконані такі завдання:

- Вивчення існуючих рішень для оцінки ключових функцій та шаблонів UX в інструментах для спільної роботи над малюванням.
- Вибір таких технологій, як Vue.js, Firebase Realtime Database, Tailwind CSS та HTML5 Canvas, для оптимальної продуктивності та візуального рендерингу в режимі реального часу.
- Розробка основного функціоналу , включаючи малювання від руки, інструменти для роботи з фігурами (прямокутник, коло), вибір кольору, товщину лінії, стирання, скасування дій та очищення полотна.
- Реалізація співпраці в реальному часі за допомогою Firebase для синхронізації штрихів, фігур та повідомлень чату між користувачами.
- Створення системи чату, інтегрованої із середовищем малювання, для підтримки живого спілкування між співробітниками.
- Тестування та оптимізація для забезпечення продуктивності, точності синхронізації в реальному часі та адаптивності до різних розмірів екранів та швидкостей з’єднання.

Об'єкт дослідження**.** Об'єктом цього дослідження є розробка веб-додатку для спільного малювання в режимі реального часу, побудованого з використанням сучасних фронтенд-технологій (Vue.js, Tailwind CSS та HTML5 Canvas) та бекенд-сервісів у режимі реального часу (Firebase).

Предмет дослідження**.** Предметом цього дослідження є реалізація синхронізації малювання в реальному часі, механізмів взаємодії з користувачем та зручних принципів UI/UX у багатокористувацькому веб-середовищі

Апробація результатів**.** Було проведено ретельне тестування для оцінки надійності синхронізації малювання в реальному часі , інтеграції чату та взаємодії з полотном за різних навантажень користувачів та мережевих умов. Програму було протестовано вручну та за допомогою автоматизованих наскрізних тестів (наприклад, за допомогою Cypress) для імітації одночасного малювання кількома користувачами. Тести підтвердили, що синхронізація штрихів, рендеринг фігур та

обмін повідомленнями працювали стабільно, а інтерфейс залишався адаптивним. Успішне впровадження темних/світлих тем та відстеження штрихів для користувача ще більше покращило взаємодію. Загалом, результати підтвердили, що CollabPaint відповідає своїм цілям доступності, зручності використання та стабільної співпраці в реальному часі.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ЗАСАДИ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ

1.1 Теоретичні дослідження систем редагування таблиць

Розробка сучасних веб-систем спільного малювання, таких як CollabPaint, ґрунтується на кількох теоретичних і технічних принципах. До них належать синхронізація даних у реальному часі, продуктивність рендерингу на стороні клієнта та чуйність інтерфейсу. Успішна реалізація таких систем вимагає інтеграції технологій, які забезпечують швидку, безперебійну та синхронізовану співпрацю між кількома користувачами через Інтернет.

Спільне редагування у реальному часі

Основною проблемою в додатках для спільного малювання є забезпечення одночасного редагування кількома користувачами, гарантуючи при цьому, що дії кожного користувача будуть точно і миттєво відображені в усіх активних сеансах. CollabPaint вирішує цю проблему, використовуючи Firebase Realtime Database, хмарну базу даних NoSQL, розроблену спеціально для додатків, що працюють у режимі реального часу.

Firebase автоматично синхронізує зміни даних між клієнтами за мілісекунди, не вимагаючи від розробників управління низькорівневою логікою паралелізму. Кожен мазок пензля, фігура або зміна стану полотна записується в базу даних і майже миттєво передається всім підключеним клієнтам. Такий підхід дозволяє уникнути складності традиційних моделей паралелізму, таких як Operational Transformation (OT) або Conflict-Free Replicated Data Types (CRDT), забезпечуючи при цьому надійну узгодженість для спільних сесій.

Firebase вирішує конфлікти, визначаючи пріоритетність замовлень на запис і використовуючи слухачі в реальному часі для відображення оновлень в інтерфейсі

користувача. Це робить його особливо ефективним для таких додатків, як CollabPaint, де користувачі часто виконують дії малювання, що перетинаються.

Протоколи зв'язку в режимі реального часу

На відміну від реалізацій на основі WebSocket, які вимагають створення власної серверної інфраструктури, Firebase абстрагується від мережевого рівня і забезпечує розповсюдження даних, кероване подіями. Це робить Firebase чудовою альтернативою для додатків, які потребують масштабованості, комунікації з низькою затримкою та простоти розгортання.

Основні переваги включають:

- Автоматичне оновлення в режимі реального часу без ручного керування сокетами.
- Підтримка офлайн і локальне кешування, що забезпечує безперервну інтерактивність навіть при переривчастому з'єднанні.
- Правила безпеки та автентифікація користувачів, що забезпечують безпечний багатокористувацький доступ до спільних полотен.
- Хоча Firebase не використовує традиційні протоколи сокетів, його продуктивність порівнянна з WebSockets для більшості випадків використання в режимі реального часу.

WebSockets відіграють ключову роль у додатках реального часу, забезпечуючи стабільне двостороннє з’єднання між клієнтом і сервером. Такий підхід дозволяє передавати дані миттєво, що критично важливо для інтерактивних функцій, таких як спільне малювання — де кожна дія користувача має відображатися у реальному часі. На відміну від традиційних HTTP-запитів, WebSockets усувають необхідність постійного перевстановлення з’єднань, знижуючи затримки та підвищуючи ефективність у колаборативних середовищах.

WebRTC: WebRTC зазвичай застосовується для потокової передачі аудіо та відео, але його можливості однорангової (P2P) комунікації також відкривають потенціал для спільних додатків. Однак для таких рішень, як CollabPaint, WebSockets часто виявляються зручнішими завдяки простоті інтеграції у клієнт-серверну архітектуру та меншій складності впровадження порівняно з децентралізованою моделлю WebRTC.

Технології рендерингу графіки: Візуальна адаптивність має вирішальне значення для роботи користувачів у додатках для малювання. CollabPaint використовує HTML5 Canvas для візуалізації інтерфейсу малювання, який забезпечує високу продуктивність і сумісність з усіма основними браузерами.

Canvas API надає розробникам низькорівневий скриптовий інтерфейс для малювання 2D-графіки. У CollabPaint він є звичним:

- Відтворення мазків, ліній, прямокутників і кіл від руки.
- Динамічно оновлювати полотно у відповідь на введення користувача у реальному часі.
- Реалізовувати такі функції, як стирання, скасування та очищення полотна.
- Підтримуйте покадрову узгодженість для всіх учасників спільної роботи.

Поєднання Canvas і системи реактивності Vue.js гарантує, що оновлення інтерфейсу будуть візуально плавними і логічно точними.

Короткий огляд ключових технологій у CollabPaint

CollabPaint використовує декілька з вищезгаданих веб-технологій для створення надійного середовища для спільного малювання. Хоча CollabPaint ще не використовує WebGL, він залишає місце для майбутнього розширення до графіки з апаратним прискоренням або більш просунутих ефектів рендерингу (наприклад, чутливість до натискання, текстуровані пензлі).

1.2 Огляд та аналіз подібних програмних розробок

У цьому розділі ми порівняємо та зіставимо існуючі платформи для спільного малювання, зосередившись на їхніх можливостях, технологічних стеках та обмеженнях. Розглянемо аналіз таких інструментів, як:

Miro: це спільна онлайн-платформа , розроблена для підтримки команд в інноваціях, плануванні та виконанні в різних місцях та часових поясах. Вона використовується для різних цілей, включаючи мозковий штурм, гнучке планування, картування шляху клієнта, дизайн продукту та дистанційні семінари. Станом на 2025 рік платформою користуються понад 90 мільйонів користувачів та понад 250 000 організацій по всьому світу.

Розробник: Miro Inc.
Архітектура: Веб-додаток та мобільні застосунки
Техстек: JavaScript (з підтримкою інтеграцій для різних фреймворків)

Ключові функції:

- Спільна робота на інтерактивній дошці у реальному часі (мультикористувацький режим).
- Розширений набір інструментів для візуалізації ідей (ментальні карти, діаграми, схеми процесів).
- Готові шаблони для бізнес-аналітики, дизайн-сесій та планування.
- Інтеграція зі Slack, Jira, Google Workspace та іншими сервісами.
- Вбудовані відеоконференції та чат для комунікації.

Переваги:

- Інтуїтивний інтерфейс із підтримкою складних проектів.
- Потужні інструменти для командного мозкового штурму.
- Гнучкість завдяки інтеграціям зі сторонніми сервісами.

Недоліки:

- Висока вартість преміум-підписки.
- Обмежені можливості налаштування у безкоштовному тарифі.

Джерело інформації: https://miro.com

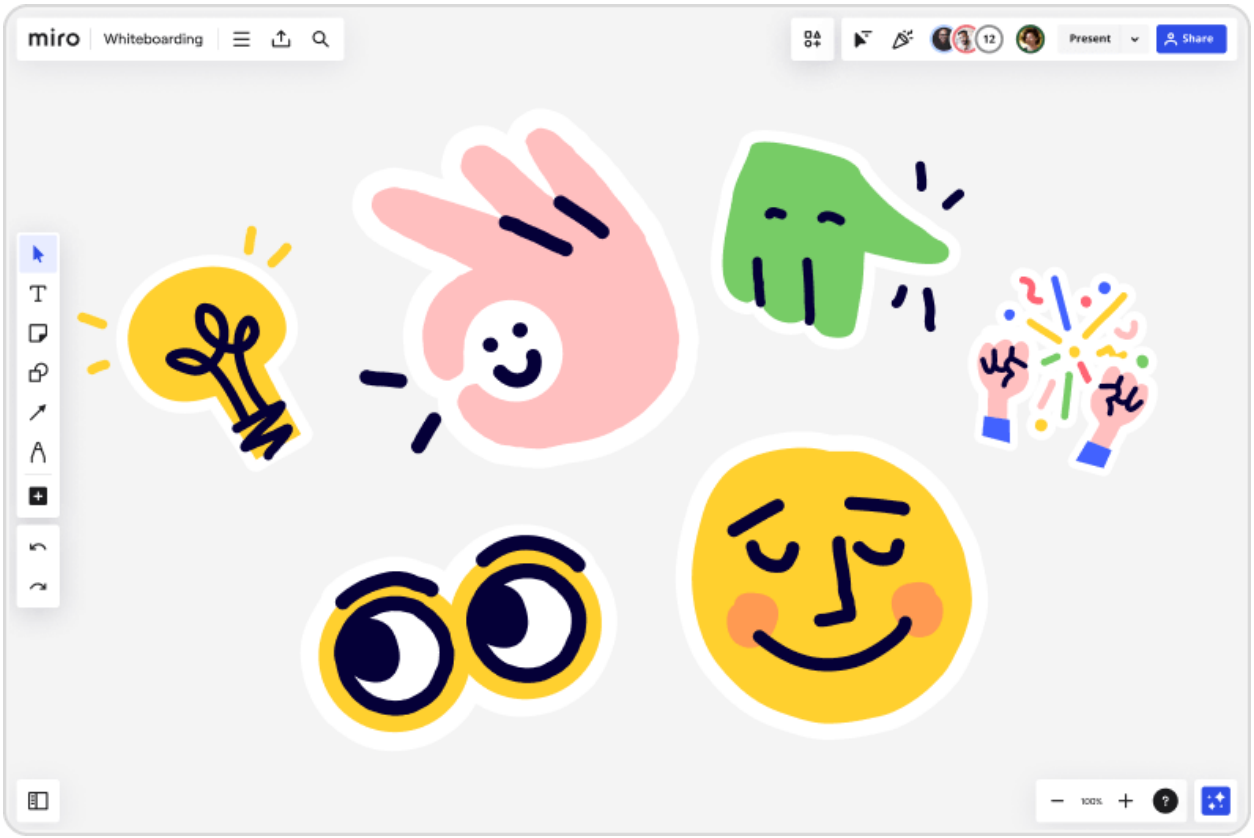

Рис. 1.1 – вигляд Miro

Google Jamboard: Це безкоштовний онлайн-сервіс від Google, який дозволяє створювати та ділитися інтерактивними цифровими дошками для спільної роботи. Jamboard можна використовувати для мозкових штурмів, планування, онлайн-навчання та інших видів діяльності, де потрібна взаємодія та творчий процес.

Розробник: Google LLC

Архітектура: Веб-додаток, фізична інтерактивна дошка, мобільні додатки

Мова реалізації: JavaScript та власні технології Google

Функції та характеристики:

- Базові інструменти для малювання (ручки, фігури, текст)

- Глибока інтеграція з Google Workspace для спільної роботи
- Інтуїтивно зрозумілий інтерфейс, орієнтований на нетехнічних користувачів
- Багатокористувацька підтримка з можливістю спільної роботи в реальному часі
- Доступ до хмарного зберігання Google Drive

Переваги:

- Оптимізована робота з іншими сервісами Google (Docs, Sheets, Meet)
- Мінімалістичний та зручний інтерфейс для швидкого старту
- Безкоштовний для користувачів з акаунтом Google

Недоліки:

- Обмежений функціонал для професійної графіки чи складного дизайну
- Повна залежність від інфраструктури Google
- Відсутність розширених інструментів редагування

Джерело інформації: https://workspace.google.com/products/jamboard

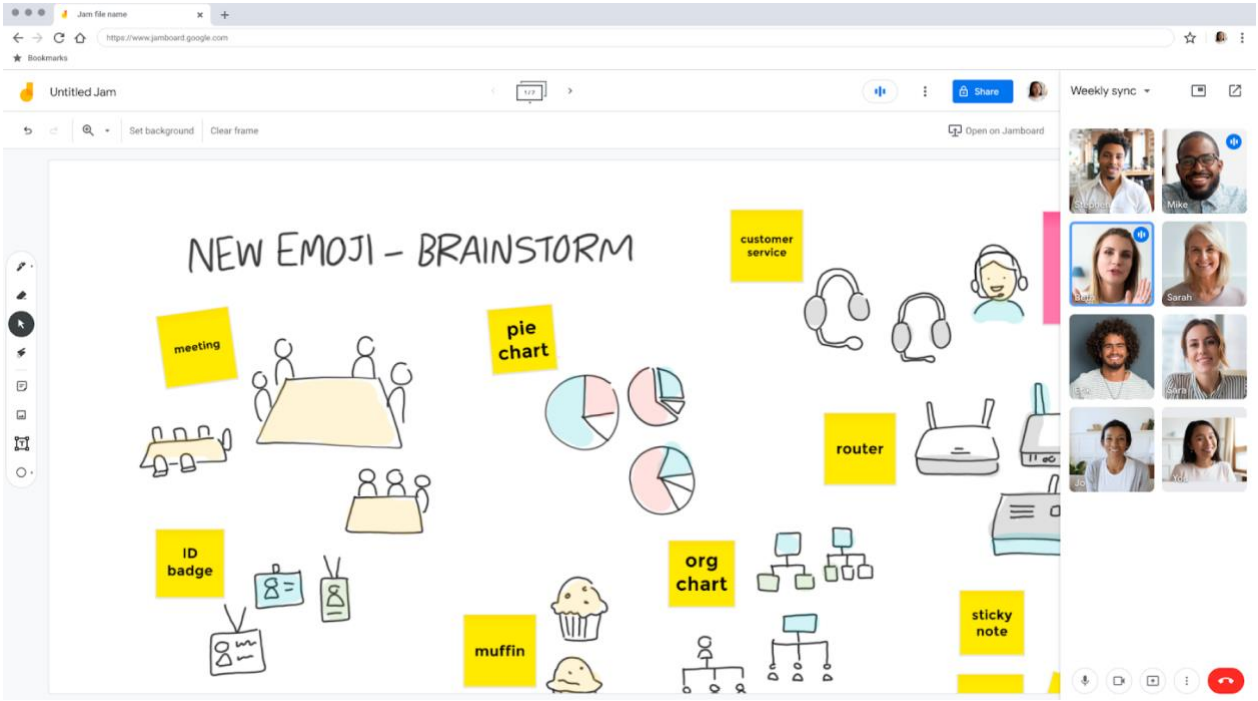

Рис. 1.2 – вигляд Google Jamboard

Aggie.io: Онлайн платформа з функціоналом для колаборативного малювання . Спочатку створена Agamnentzar, оригінальним творцем та власником гри Pony Town, і названа "Aggie.io", розміщена на однойменному домені.

Розробник: Magma Studio

Архітектура: Веб-додаток

Мова реалізації: JavaScript

Функції та характеристики:

- Справжній режим спільної роботи з можливістю одночасного редагування полотна кількома користувачами.
- Просунута робота з шарами для більш гнучкого процесу малювання
- Широкий вибір настроюваних інструментів та пензлів
- Можливість почати роботу без необхідності реєстрації або входу в систему

- Функція експорту готових робіт у популярні графічні формати

Переваги:

- Спеціалізоване рішення для спільної творчої роботи
- Простий та інтуїтивно зрозумілий інтерфейс
- Повноцінна робота прямо у веб-браузері без необхідності встановлення додаткового ПЗ

Недоліки:

- Періодичні затримки синхронізації при груповій роботі
- Дещо обмежений набір інструментів порівняно з професійними графічними редакторами

Відсутність розширених функцій для складних проектів

Джерело інформації: https://aggie.io

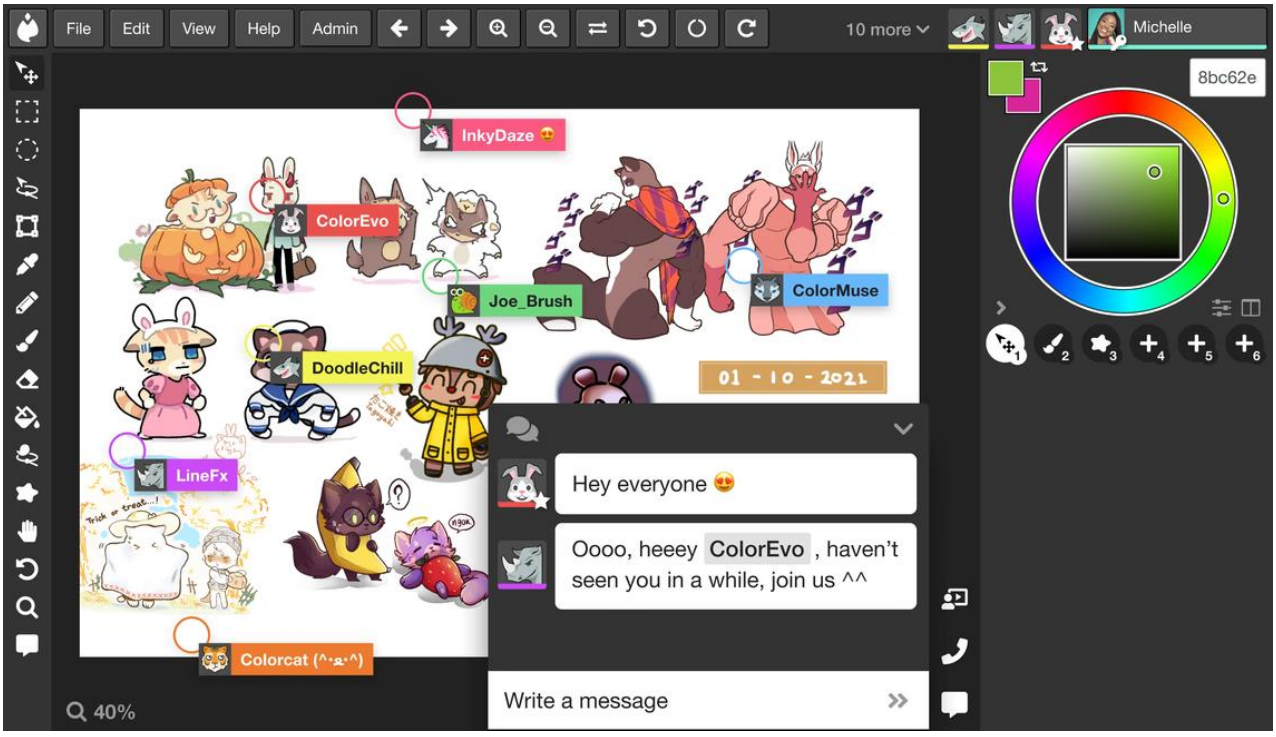

Рис. 1.3 – вигляд Aggie.io

Кожен з розглянутих інструментів пропонує унікальні можливості і пристосований до конкретних випадків використання. Наприклад, Miro та Google Jamboard ідеально підходять для мозкового штурму та освітніх цілей, тоді як Aggie.io орієнтований на творчих художників, які потребують можливостей для спільного малювання. Однак обмеження в доступності, складність інструментів або вартість підкреслюють потребу в легкому, браузерному додатку, такому як CollabPaint, який фокусується на простоті, співпраці в реальному часі і легкості використання.

РОЗДІЛ 2
COLLABPAINT: РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ МАЛЮВАННЯ ОНЛАЙН З БАЗОВИМИ ІНСТРУМЕНТАМИ

2.1 ТЗ та вимоги до програмного забезпечення CollabPaint

У сучасну цифрову епоху інструменти для спільної роботи стали невід'ємною частиною творчих і професійних робочих процесів. Можливість для кількох користувачів взаємодіяти зі спільним контентом у режимі реального часу, особливо у сфері онлайн-малювання чи роботи на дошці, стає дедалі ціннішою для віддалених команд, освітніх платформ та творчих спільнот. Однак існуючі рішення часто або надто складні, або вимагають платної підписки, або не мають зручних інтерфейсів, пристосованих до базових потреб малювання та спілкування в режимі реального часу.

Метою цього проекту є розробка веб-додатку для спільного малювання - CollabPaint, що дозволяє декільком користувачам одночасно малювати на спільному полотні, спілкуватися в чаті та ефективно керувати своїми сесіями. Система має бути доступною, інтуїтивно зрозумілою та швидко реагувати на запити користувачів, дозволяючи їм співпрацювати в режимі реального часу, не вимагаючи від них встановлення додаткового програмного забезпечення.

Щоб досягти цієї мети, додаток повинен відповідати наступним вимогам до розробки:

- Багатокористувацька співпраця: Підтримка одночасного малювання кількома користувачами на одному полотні в режимі реального часу.
- Забезпечення спілкування користувачів в кімнаті в реальному часі через чат на сторінці.
- Аутентифікація користувачів: Безпечний вхід та реєстрація для ідентифікації та управління користувачами.
- Керування сеансами: Створення унікальних кімнат для малювання, до яких можуть приєднатися кілька користувачів.

- Постійне зберігання даних: Збереження сеансів малювання та повідомлень у чаті за допомогою хмарних баз даних.
- Синхронізація в реальному часі: Миттєве відображення взаємодії користувачів (малювання та обмін повідомленнями) на всіх підключених клієнтах.
- Адаптивний UI/UX: сучасний, інтуїтивно зрозумілий користувальницький інтерфейс, який адаптується до різних розмірів екранів і пристроїв.
- Продуктивність і масштабованість: Ефективна обробка потоків даних у реальному часі та користувацьких з'єднань з мінімальною затримкою.

2.2 Вибір моделі розробки для програмного засобу CollabPaint

Для забезпечення ефективної, гнучкої та ітеративної розробки програмного інструменту CollabPaint було обрано методологію розробки Agile. Agile забезпечує безперервний зворотній зв'язок, часту доставку функціональних компонентів і швидку адаптацію до змін - все це має вирішальне значення при розробці сучасної веб-платформи для спільної роботи в режимі реального часу. Проект реалізовувався у вигляді коротких спринтів, кожна ітерація яких фокусувалася на впровадженні та тестуванні певної функції або вдосконалення (наприклад, синхронізація полотна, функціонал чату або автентифікація користувачів).

Технологічний стек та використані інструменти

У процесі розробки використовувався сучасний та ефективний набір технологій, фреймворків та бібліотек, які добре узгоджуються з гнучким підходом та системними вимогами.

Фронтенд

Nuxt.js : Прогресивний фреймворк JavaScript, що використовується для створення реактивного користувацького інтерфейсу. API композиції Vue уможливив кращу модуляризацію та повторне використання логіки.

Tailwind CSS: Фреймворк CSS, який дозволив швидко створювати прототипи інтерфейсу та узгоджувати стилі без необхідності писати кастомний CSS з нуля.

Pinia: Бібліотека керування станом для Vue 3, яка замінила Vuex, забезпечуючи простіший та ефективніший спосіб керування спільним станом програми, таким як дані користувача та інформація про кімнату/сеанс.

Vite: Сервер швидкої розробки та інструмент збірки, який забезпечує миттєву гарячу заміну модулів та швидкий запуск, що значно покращує досвід розробки.

Бекенд та функції реального часу

Firebase: Використовується як основна платформа backend-as-a-service для автентифікації, бази даних та хостингу.

Аутентифікація Firebase: Забезпечує безпечну реєстрацію користувачів та вхід за допомогою електронної пошти та пароля.

База даних Firebase в режимі реального часу: Вибрана за вбудовані можливості синхронізації в реальному часі. Вона зберігає дані про штрихи, повідомлення в чаті та метадані кімнати, забезпечуючи безперешкодну співпрацю між користувачами.

Firebase Storage: (необов'язково/заплановано) Для зберігання та обслуговування завантажених користувачем зображень профілів, якщо це необхідно.

Інші інструменти розробки

Git та GitHub: Для контролю версій. Для управління якістю коду та відстеження прогресу використовуються гілки функцій та запити на витягування.

ESLint + Prettier: Забезпечив узгоджене форматування коду та мінімізував помилки через синтаксичні проблеми.

Postman (опціонально): Використовувався під час тестування інтеграції та налагодження кінцевих точок Firebase REST API.

Чому обрали Agile

Розробка CollabPaint передбачала часте тестування, коригування інтерфейсу та налаштування функцій у реальному часі, якими важко керувати в рамках жорстких моделей, таких як Waterfall. Agile забезпечив це:

Постійна інтеграція зворотного зв'язку після кожного прототипу.

Пріоритетність основних функцій (малювання, чат, автентифікація), а потім вдосконалення (наприклад, редагування профілю користувача, видалення кімнати).

Гнучка дорожня карта, що дозволяє адаптуватися до технічних обмежень або кращих рішень у міру розвитку.

Таким чином, поєднання методології Agile, сучасних фреймворків JavaScript та хмарних сервісів, таких як Firebase, дозволило швидко розробити, протестувати та ітерації інструменту CollabPaint, забезпечивши створення надійної та зручної платформи для спільного малювання.

2.3 Загальний опис проекту CollabPaint

Проєкт CollabPaint - це веб-додаток для спільного малювання, який дозволяє кільком користувачам малювати разом у реальному часі на спільному полотні, спілкуючись за допомогою інтегрованої системи чату. Розроблена в першу чергу для освітніх і творчих цілей, система підкреслює простоту використання, швидкість реагування в реальному часі і доступність через сучасні веб-браузери.

CollabPaint надає користувачам інтуїтивно зрозуміле онлайн-середовище, де вони можуть створювати, спілкуватися та співпрацювати візуально без необхідності встановлювати програмне забезпечення. Система включає в себе реєстрацію та вхід користувачів, створення та видалення кімнат, малювання на полотні з синхронізацією в реальному часі та чат в режимі реального часу. Крім того, користувачі можуть керувати інформацією свого профілю та взаємодіяти з іншими в сеансовому середовищі.

Архітектура та структура проекту

CollabPaint - це односторінковий додаток (SPA), розроблений за допомогою Vue.js 3 та організований у модульні компоненти, подання та композити. Він структурований відповідно до найкращих практик для масштабованості та супроводу.

1. Структура проєкту

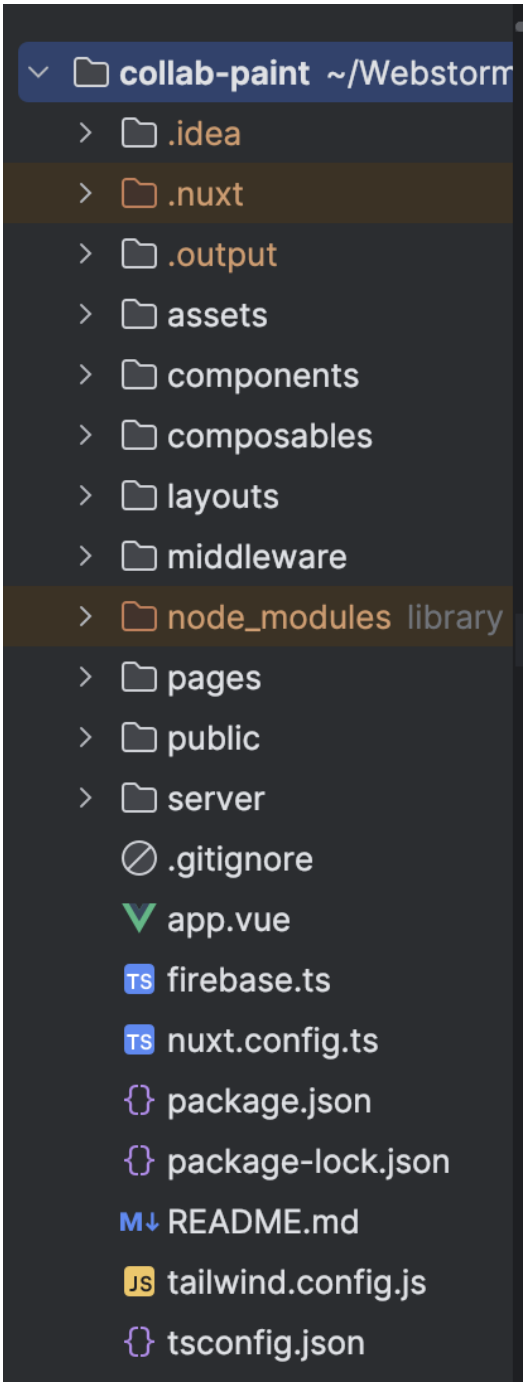

Рис. 1.1 – структура проєкту

2. Ключові сторінки та їх функціональність

/ (Головна сторінка)

- Цільова сторінка з основною інформацією або перенаправлення в залежності від стану автентифікації.
- Авторизовані користувачі перенаправляються на сторінку /rooms або /profile.

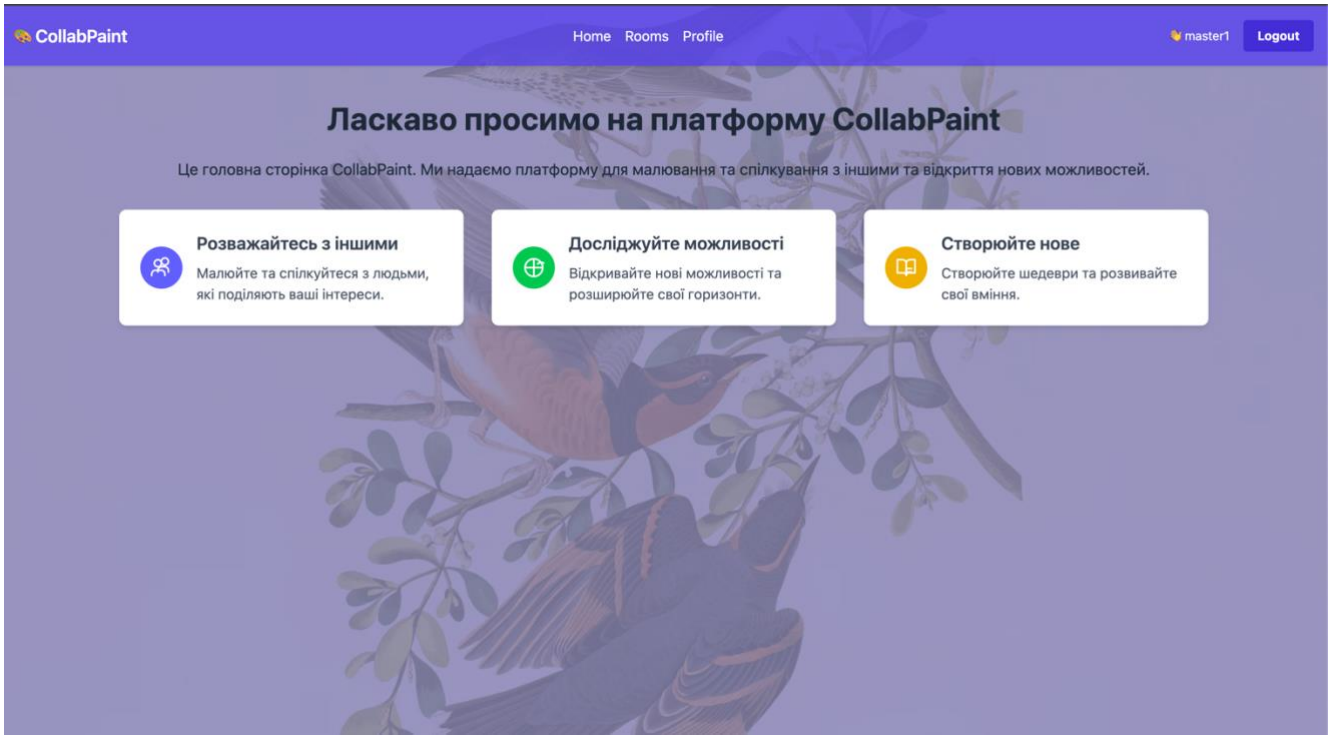

Рис. 1.2 – Головна сторінка

/login

- Форма для входу за допомогою електронної пошти/пароля.
- Перенаправляє на головну панель після успішного входу.
- Включає обробку помилок і валідацію.
- Реєстраційна форма для введення адреси електронної пошти/пароля.
- Зберігає дані користувача в Firebase Authentication.

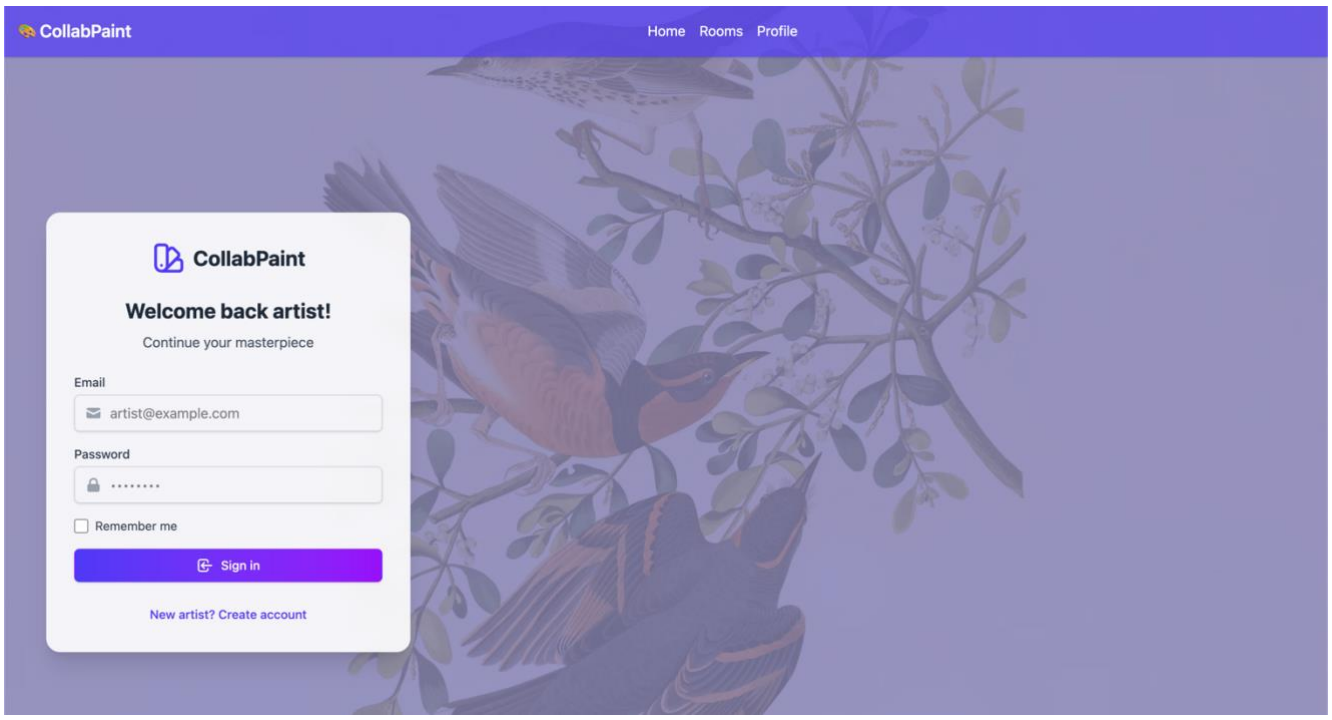

Рис. 1.3 – Сторінка входу/реєстрації

/profile

- Відображає поточну інформацію про користувача (ім'я, електронну пошту, аватар).
- Дозволяє редагувати дані користувача (з підтвердженням).
- Попередній перегляд зображення перед збереженням (якщо користувач завантажив фото).
- Зберігає зміни через Firebase Authentication API.

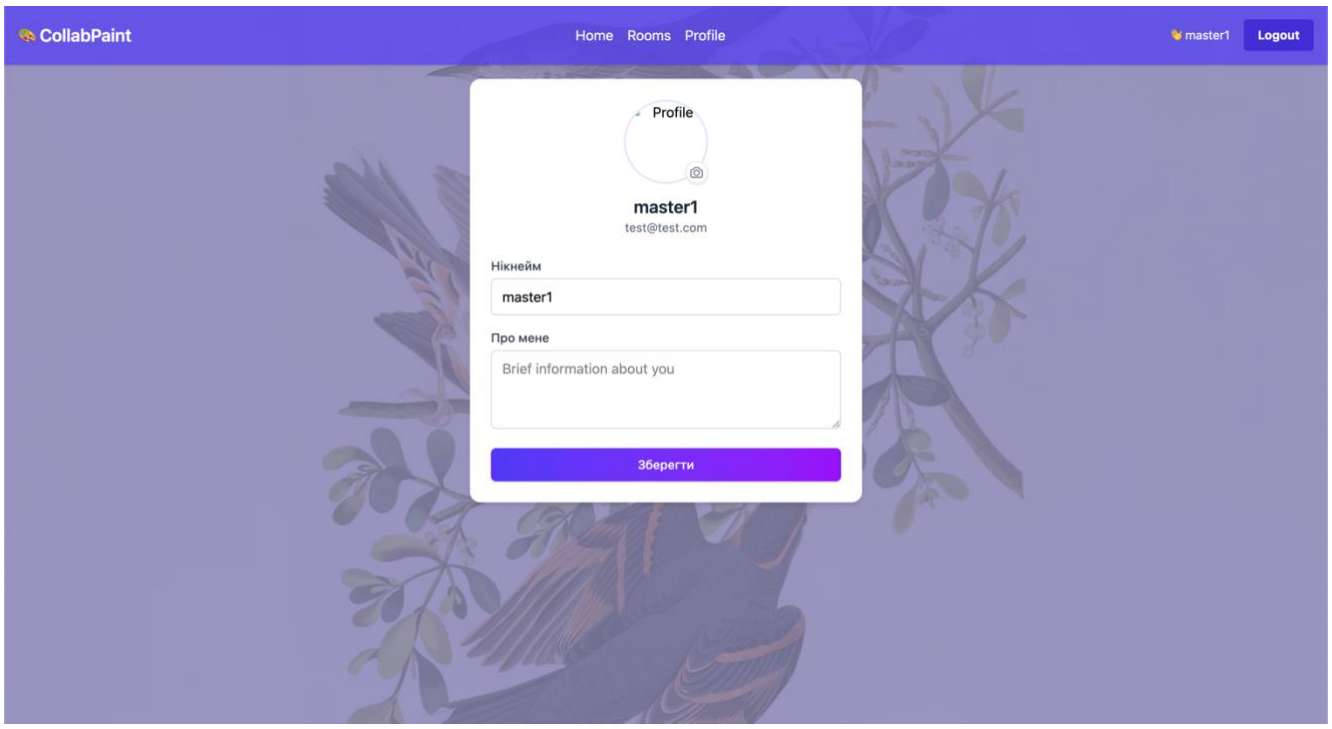

Рис. 1.4 – Сторінка профілю

/rooms

- Показує форму для створення кімнати разом з іншим користувачем
- Показує список доступних віталень (необов'язково, може використовуватися для навігації або управління).
- Може бути розширений для підтримки публічних/приватних кімнат або запрошень.

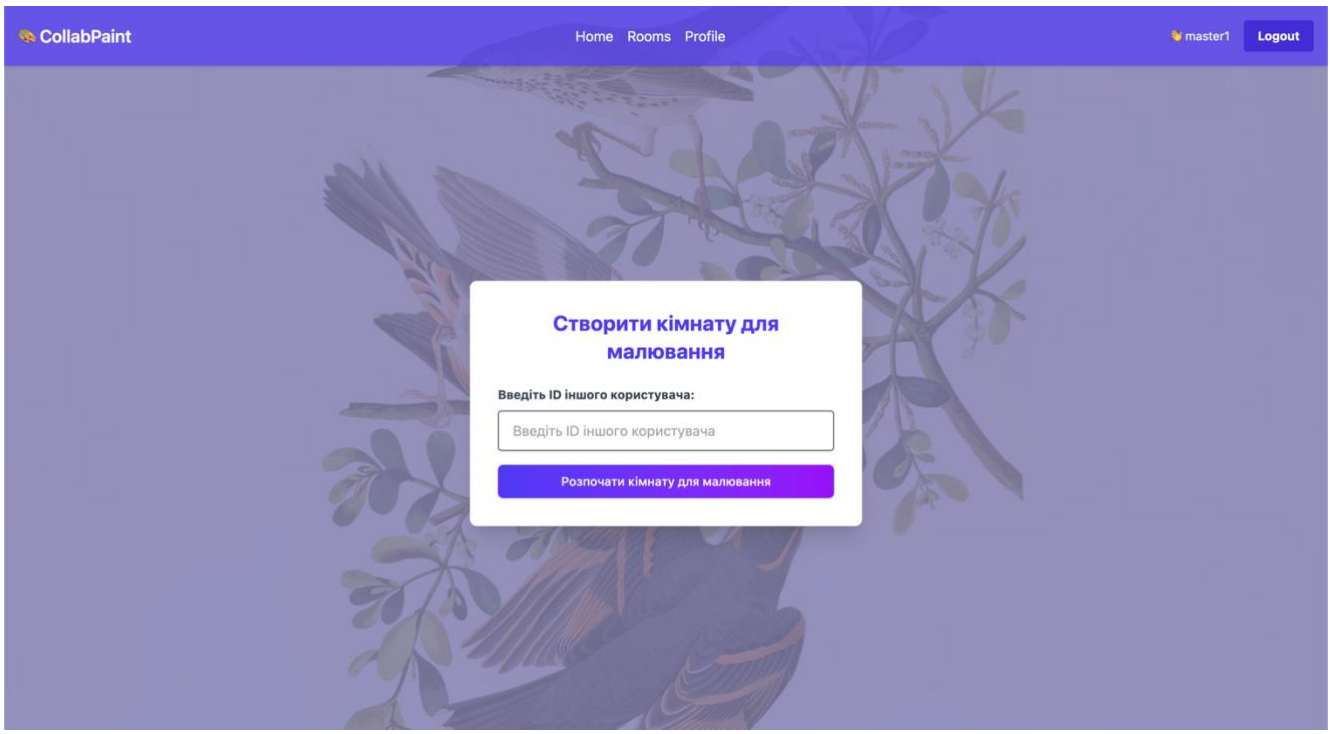

Рис. 1.5 – Сторінка створення кімнати

/draw/[roomId]

- Основний робочий простір для спільного малювання.

Можливості:

- Полотно для малювання в реальному часі за допомогою елемента <canvas>.
- Синхронізація штрихів з базою даних Firebase в реальному часі.
- Інтегрована панель чату з можливістю обміну повідомленнями в реальному часі.
- Оповіщення про присутність користувача (опціонально).
- Кнопка "Видалити кімнату (x)" для видалення кімнати з бази даних.
- Оптимізований адаптивний макет:
- Полотно займає весь доступний простір зліва.
- Панель чату закріплена праворуч з мінімальною шириною.

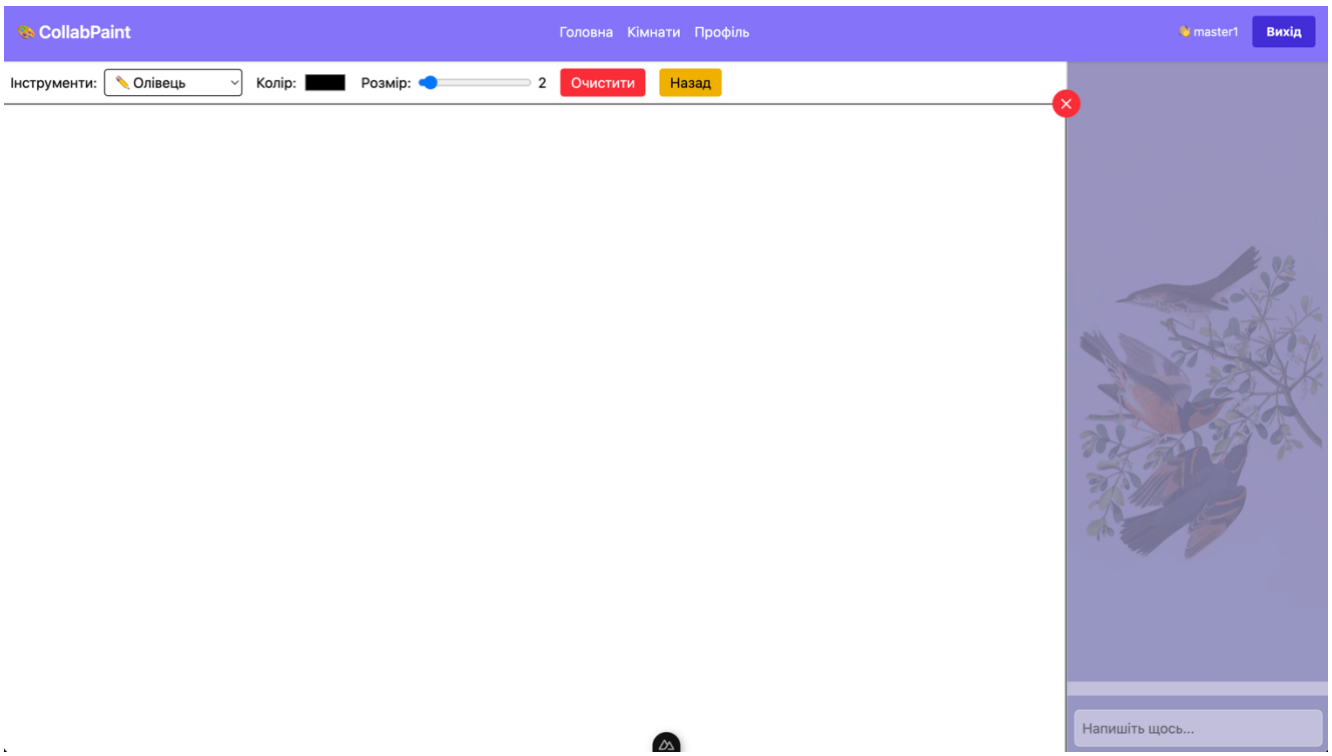

Рис. 1.6 – Основна сторінка CollabPaint

3. Компоненти

CollaborativeCanvas.vue

- Обробляє логіку малювання (події миші, збереження штрихів).
- Відображає елемент <canvas>.
- Використовує watchEffect для оновлення полотна на основі віддалених змін.

ChatPanel.vue

- Відображає повідомлення з іменем відправника.
- Автоматично прокручує до останнього повідомлення.
- Поле введення для надсилання повідомлень через Firebase.

ProfileEditor.vue

- Форма для редагування імені користувача та фотографії.
- Перегляд зображення перед збереженням.
- Обробляє відповіді на помилки від Firebase.

4. Інтеграція з Firebase

Автентифікація: Використання Firebase Auth (електронна пошта/пароль).

База даних в режимі реального часу:

/rooms/{roomId}/strokes: Зберігає дані малювання.

/rooms/{roomId}/messages: Зберігає повідомлення чату.

Правила безпеки: Визначені для того, щоб дозволити користувачам читати/записувати лише власні дані та кімнати, в яких вони перебувають.

Хостинг Firebase (необов'язково): Додаток можна розгорнути за допомогою Firebase CLI для глобального доступу.

5. Модель даних

Кімната

{
"strokes": [{ "x": 0, "y": 0, "color": "#000" }],
"messages": [{ "text": "Hello", "senderId": "abc123", "userName": "John" }],
"users": ["user1", "user2"]
}

Користувач

Зберігається в Firebase Auth:

{
"uid": "abc123",
"email": "user@example.com",

"displayName": "John",

}

Стилістика та адаптивність. Весь інтерфейс користувача побудований за допомогою Tailwind CSS, що забезпечує послідовну систему дизайну, орієнтовану насамперед на мобільні пристрої.Макет додатку є адаптивним і добре адаптується під десктопні, планшетні та мобільні версії. Компоненти використовують класи утиліт для інтервалів, вирівнювання, станів наведення та переходів.

Маршрутизація та навігація. Vue Router налаштовано з динамічними маршрутами (наприклад, /draw/:roomId) та навігаційними захисниками, щоб користувачі могли отримати доступ до певних сторінок лише після автентифікації. Додаток використовує router.push() для програмної навігації після дій користувача (наприклад, вхід, створення кімнати).

Загальна структура CollabPaint відображає масштабований, підтримуваний і орієнтований на користувача веб-додаток, який інтегрує сучасні технології інтерфейсу з можливостями бекенда в режимі реального часу. Модульний дизайн компонентів, узгоджений фреймворк інтерфейсу та надійний бекенд Firebase разом забезпечують повноцінну та адаптивну платформу для спільного малювання та спілкування.

2.5 Особливості програмної реалізації CollabPaint

Реалізація програмного інструменту CollabPaint включає в себе широкий спектр функцій, які працюють разом, щоб забезпечити спільне малювання та спілкування в реальному часі. Нижче наведено основні функції, кожна з яких детально пояснюється разом з фрагментами коду та архітектурними рішеннями.

Автентифікація користувачів та управління профілями

Автентифікація здійснюється за допомогою Firebase Authentication, що підтримує реєстрацію та вхід за допомогою електронної пошти/пароля.

Приклад входу:

```
const handleAuth = async () => {  Show usages  alex Ko *
  error.value = ''
  try {
    if (isLogin.value) {
      await signInWithEmailAndPassword(auth, email.value, password.value)
      await navigateTo( to: '/')
      console.log('Logged in!')
    } else {
      await createUserWithEmailAndPassword(auth, email.value, password.value)
      console.log('Registered!')
    }
  } catch (e: any) {
    error.value = e.message
  }
}
```

Рис. 2.1 – вхід або реєстрацію користувача за допомогою Firebase Authentication на основі значення isLogin.

Функція handleAuth керує процесом входу користувача в систему або реєстрації нового користувача, залежно від поточного стану. Спочатку вона очищає всі попередні повідомлення про помилки. Якщо користувач вирішив увійти, функція намагається зареєструвати його, використовуючи його електронну пошту та пароль. Якщо це вдається, вона перенаправляє користувача на домашню сторінку.

Якщо користувач замість цього реєструється, функція створює новий обліковий запис з вказаною електронною поштою та паролем. Після реєстрації він реєструє повідомлення з підтвердженням. Якщо щось піде не так під час входу або реєстрації (наприклад, неправильні облікові дані або слабкий пароль), функція фіксує помилку і зберігає повідомлення, щоб його можна було показати користувачеві.

Функція goToRoom відповідає за створення або перехід до спільної кімнати для малювання між двома користувачами в додатку CollabPaint. Основна мета полягає в тому щоб забезпечити двом користувачам спільний простір для малювання

(кімнату), де вони можуть малювати разом в режимі реального часу. Якщо кімната вже існує, він просто переходить до неї; в іншому випадку, він створює її.

```
async function goToRoom() {  Show usages  alex Ko
  if (!user.value || !otherUserId.value) return

  const myId = user.value.uid
  const theirId = otherUserId.value.trim()
  const sorted = [myId, theirId].sort()
  const roomId = `${sorted[0]}_${sorted[1]}`

  const roomRef = dbRef(database, path: `rooms/${roomId}`)

  // Check if room exists in Realtime Database
  const snapshot = await get(roomRef)
  if (!snapshot.exists()) {
    // Create the room if it doesn't exist
    await set(roomRef, value: {
      createdAt: Date.now(),
      strokes: [],
    })
  }

  roomLink.value = `/draw/${roomId}`
  router.push(roomLink.value)
}
```

Рис. 2.2 – Функція goToRoom

if (!user.value || !otherUserId.value) return. Якщо поточний користувач відсутній або ідентифікатор іншого користувача не введений, немає сенсу продовжувати. Ця перевірка запобігає помилкам і гарантує, що обидва учасники будуть ідентифіковані.

Сортування гарантує, що одна і та ж пара користувачів завжди отримує однаковий ідентифікатор кімнати, незалежно від того, хто її ініціював. Це гарантує використання однієї спільної кімнати замість створення дублікатів.

Посилання на кімнату в базі даних Firebase Realtime Database, це створює шлях до місця, де дані кімнати будуть збережені або отримані в базі даних Firebase.

Створюється кімната, якщо вона не існує, якщо кімнати ще не існує, програма створює її з позначкою часу і порожнім масивом штрихів (тобто малюнків). Це створює середовище для малювання.

```
onMounted( hook: () => {
  resizeCanvasToDisplaySize()
  window.addEventListener( type: 'resize', resizeCanvasToDisplaySize)

  if (canvas.value) {
    ctx = canvas.value.getContext('2d')
  }

  onChildAdded(strokesRef, callback: (snapshot) => {
    const stroke = snapshot.val()
    strokes.value.push({ ...stroke, id: snapshot.key })

    if (stroke.type === 'line') {
      drawLine(stroke.x1, stroke.y1, stroke.x2, stroke.y2, stroke.color, stroke.size)
    } else if (stroke.type === 'rectangle' || stroke.type === 'circle') {
      drawShape(stroke)
    }
  })
})
```

Рис. 2.3 – Ініціалізації полотна canvas

Цей код виконується під час монтування компонента:

– Змінює розмір полотна відповідно до розміру екрана та оновлює його при зміні розміру вікна.

- Ініціалізує контекст малювання полотна (ctx).
- Перевіряє наявність нових штрихів у базі даних Firebase Realtime Database (strokesRef).
- Коли додається новий штрих, він малюється на полотні (лінія, прямокутник або коло) і зберігається локально.

```
function startDrawing(e: MouseEvent) {  Show usages  alex Ko
  if (!canvas.value) return
  const rect = canvas.value.getBoundingClientRect()
  const x = e.clientX - rect.left
  const y = e.clientY - rect.top

  isDrawing.value = true
  prev.value = { x, y }
  startPos.value = { x, y }
}
```

Рис 2.4 - Зберігання початкової точки у параметрах prev і startPos для відстеження руху малювання

Ця функція startDrawing спрацьовує, коли користувач починає малювати на полотні (наприклад, при наведенні миші):

- Перевіряє, чи існує полотно - повертається достроково, якщо ні.
- Отримує позицію полотна - getBoundingClientRect() повертає позицію полотна на екрані.
- Обчислює координати миші відносно полотна - від позиції миші віднімається верхній лівий кут полотна.
- Встановлює стан малювання:
- isDrawing.value = true: сигналізує про початок малювання.

Зберігає початкову точку у параметрах prev і startPos для відстеження руху малювання.

```
function draw(e: MouseEvent) {  Show usages  alex Ko *
  if (!isDrawing.value || !canvas.value) return
  lastMouseEvent.value = e

  const rect = canvas.value.getBoundingClientRect()
  const current = {
    x: e.clientX - rect.left,
    y: e.clientY - rect.top,
  }

  if (tool.value === 'pencil' || tool.value === 'eraser') {...}
  else if (tool.value === 'rectangle' || tool.value === 'circle') {
    previewShape.value = {
      type: tool.value,
      x1: startPos.value.x,
      y1: startPos.value.y,
      x2: current.x,
      y2: current.y,
      color: color.value,
      size: lineWidth.value,
    }
    redrawCanvas()
    drawShape(previewShape.value)
  }
}
```

Рис 2.5 – Основна функція малювання

Функція малювання запускається під час переміщення миші по полотну:

Вийти достроково, якщо малювання не розпочалося або полотно не існує.

Обчислює поточну позицію миші відносно полотна за допомогою getBoundingClientRect().

Якщо інструментом є олівець або гумка:

– Визначити колір (для гумки використовується білий).
– Намалюйте лінію від попередньої позиції до поточної.
– Збережіть штрих у базі даних Firebase Realtime Database (strokesRef) з метаданими (тип, позиція, колір, розмір, користувач, час).
– Оновіть попередню позицію, щоб продовжити плавне малювання.

Якщо інструмент прямокутник або коло:

– Збережіть початкову та поточну позиції фігури у previewShape для попереднього перегляду в реальному часі.
– Очистіть і перемалюйте полотно, а потім покажіть попередній перегляд фігури (ще не зберігаючи її).

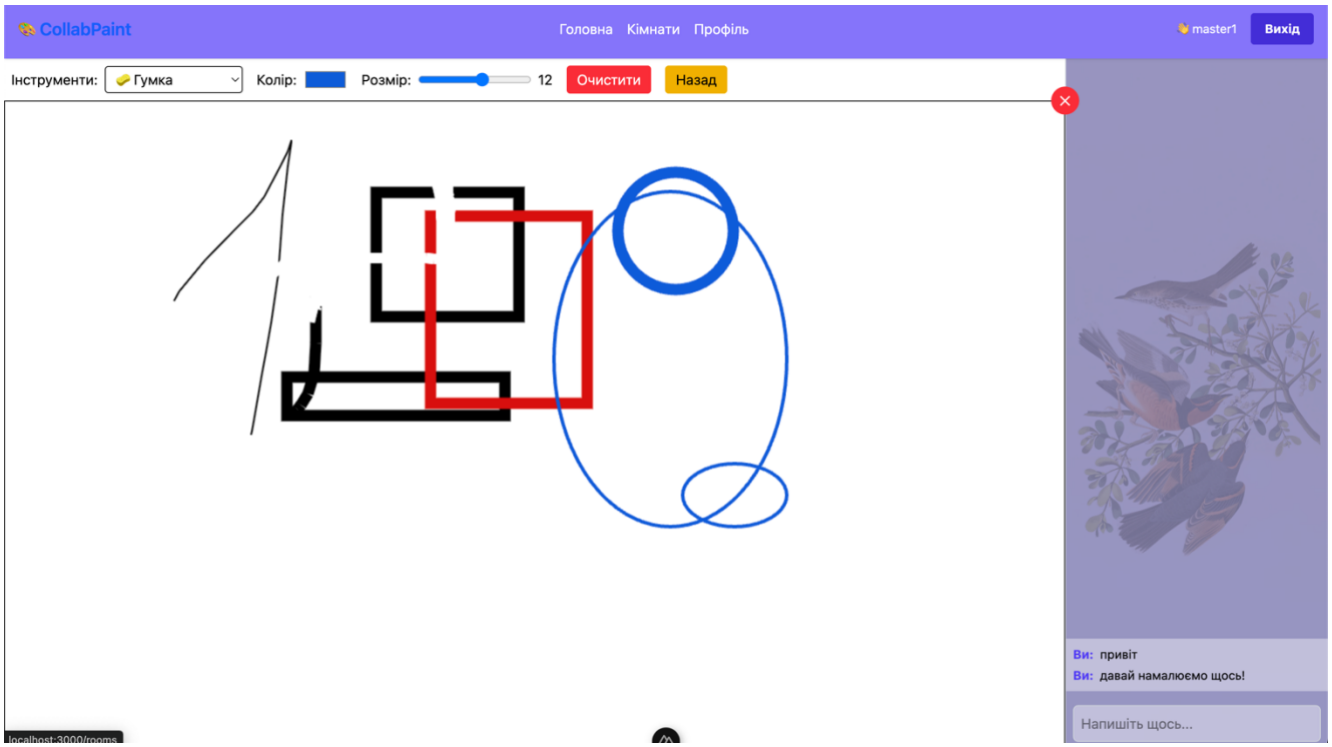

Рис 2.4 – робочий простір CollabPaint

```
function stopDrawing() {  Show usages  alex Ko
  if (!isDrawing.value || !canvas.value || !user.value) return
  isDrawing.value = false

  if (tool.value === 'rectangle' || tool.value === 'circle') {
    const rect = canvas.value.getBoundingClientRect()
    const current = {
      x: lastMouseEvent.value!.clientX - rect.left,
      y: lastMouseEvent.value!.clientY - rect.top,
    }

    const shape = {
      type: tool.value,
      x1: startPos.value.x,
      y1: startPos.value.y,
      x2: current.x,
      y2: current.y,
      color: color.value,
      size: lineWidth.value,
      user: user.value.uid,
      timestamp: Date.now(),
    }

    push(strokesRef, shape)
    previewShape.value = null
  }
}
```

Рис 2.5 – функція stopDrawing

Функція stopDrawing завершує дію малювання:

Вийти достроково, якщо малювання не розпочато, немає полотна або користувача.

Встановіть isDrawing у false - зупиніть малювання.

Якщо поточним інструментом є прямокутник або коло:

– Обчислити позицію миші, на якій зупинилося малювання.
– Створіть об'єкт-фігуру з позицією, стилем, інформацією про користувача та міткою часу.
– Збережіть фігуру у Firebase (strokesRef).
– Очистіть область попереднього перегляду фігури.
– Це завершує роботу з фігурами, намальованими за допомогою інструментів прямокутника або кола, зберігаючи їх назавжди.

```
function clearCanvas() {  Show usages  alex Ko
  if (!ctx || !canvas.value) return
  ctx.clearRect( x: 0,  y: 0, canvas.value.width, canvas.value.height)

  // Clear strokes in Firebase to an empty array, not null
  set(strokesRef,  value: [])

  // Clear local strokes
  strokes.value = []
}

function undoStroke() {  Show usages  alex Ko
  if (!user.value) return
  const last = [...strokes.value].reverse().find((s) => s.user === user.value?.uid)
  if (!last) return
  const strokeRef = dbRef(db,  path: `rooms/${props.roomId}/strokes/${last.id}`)
  remove(strokeRef)
  strokes.value = strokes.value.filter((s) => s.id !== last.id)
  redrawCanvas()
}
```

Рис 2.6 – функції clearCanvas та undoStroke

clearCanvas()

– Повністю очищає полотно за допомогою clearRect.
– Скидає мазки Firebase, встановлюючи список мазків у порожній масив.
– Спустошує локальний масив мазків, щоб відобразити очищене полотно.

undoStroke()

- Знаходить останній мазок користувача (останній, зроблений поточним користувачем).
- Видаляє його з Firebase за ідентифікатором та шляхом посилання.
- Видаляє його з локального масиву штрихів.
- Перемальовує полотно з урахуванням вилучення.

Поруч з полотном розміщено чат у реальному часі, створений за допомогою Firebase та реактивних компонентів Vue.

```
function sendMessage() {  Show usages  alex Ko
  if (!newMessage.value.trim()) return
  push(messagesRef,  value: {
    text: newMessage.value,
    senderId: user.value?.uid,
    timestamp: Date.now(),
    userName: user.value.displayName || user.value.email
  })
  newMessage.value = ''
}
```

Рис 2.7 – чат функція sendMessage

Перевіряє, чи повідомлення не порожнє (ігнорує пробіли).

Надсилає повідомлення до Firebase за допомогою push(), зберігаючи його:

- Текст повідомлення
- Ідентифікатор відправника
- Мітку часу
- Ім'я або email відправника

Очищає поле введення після відправлення.

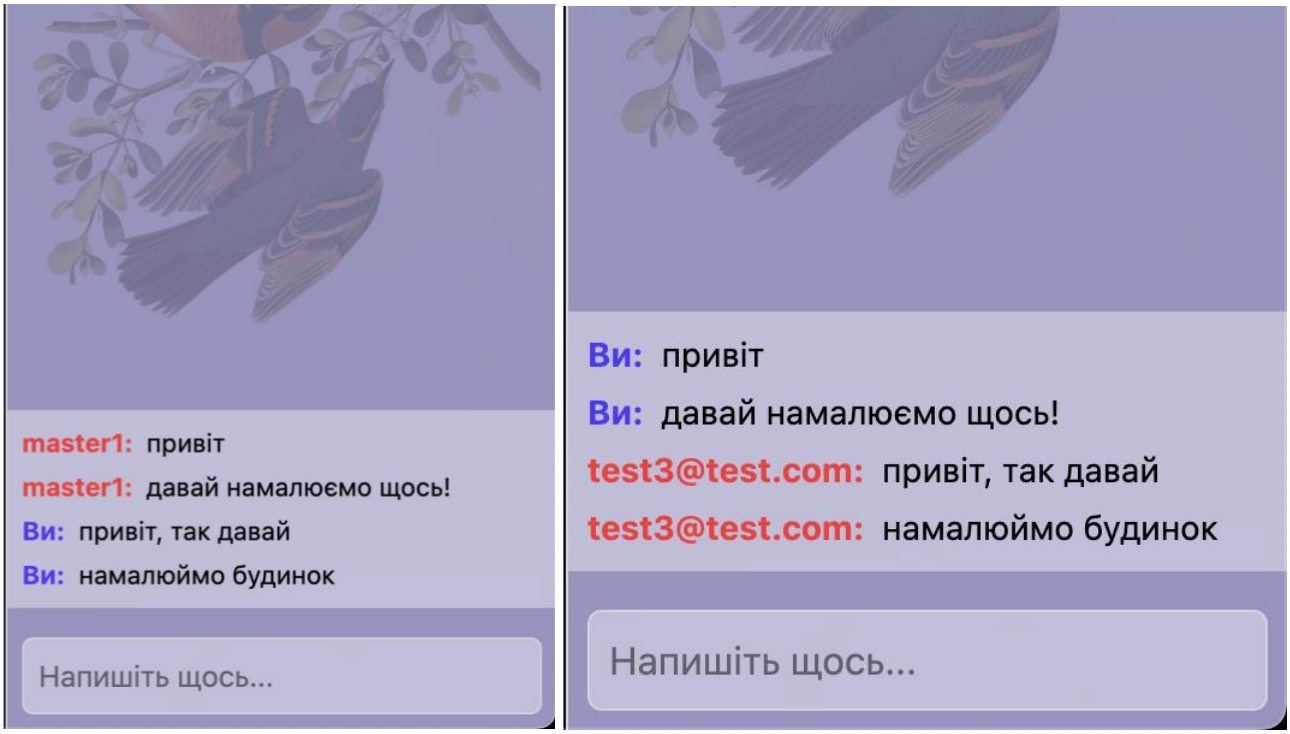

Рис 2.8(а, б) – вигляд повідомлень

```
onMounted( hook: () => {
  onChildAdded(messagesRef,  callback: (snapshot) => {
    const msg = snapshot.val()
    messages.value.push({
      id: snapshot.key,
      ...msg,
    })

    // Scroll to bottom
    nextTick( fn: () => {
      if (messagesContainer.value) {
        messagesContainer.value.scrollTop = messagesContainer.value.scrollHeight
      }
    })
  })
})
```

Рис 2.9 – Прослуховування нових повідомлень

Прослуховування нових повідомлень. Повідомлення відображаються з ідентифікацією відправника і стилізовано відповідно до того, хто їх надіслав.

Коли компонент змонтовано (onMounted):

– Він прослуховує нові повідомлення, додані до Firebase messagesRef (onChildAdded).
– Кожне нове повідомлення (знімок) витягується і додається до локального масиву повідомлень зі своїм унікальним ID.
– Після додавання повідомлення, він чекає на оновлення DOM (nextTick), а потім прокручує контейнер повідомлень донизу, переконуючись, що найновіше повідомлення є видимим.

```
async function deleteRoom() {  Show usages  alex Ko *
  if (!roomId.value) return

  const confirmDelete = confirm('Ви впевнені що хочете видалити кімнату?')
  if (!confirmDelete) return

  const roomDbRef = dbRef(db,  path: `rooms/${roomId.value}`)

  try {
    await remove(roomDbRef)
    router.push('/rooms')
  } catch (error) {
    console.error('Error deleting room:', error)
    alert('Failed to delete room.')
  }
}
```

Рис 2.10 – функція deleteRoom

Функція deleteRoom:

– Перевіряє, чи існує roomId; якщо ні, зупиняється.
– Просить користувача підтвердити, що він дійсно хоче видалити кімнату.

- Якщо підтверджено, намагається видалити дані кімнати з бази даних (rooms/{roomId}).
- У разі успіху перенаправляє користувача на сторінку /rooms.
- Якщо під час видалення трапляється помилка, записує її в журнал і показує попередження.

Події синхронізації полотна

Коли користувач починає малювати на полотні, натискаючи кнопку миші, активується функція startDrawing. Спочатку вона перевіряє, чи існує полотно, а потім обчислює положення миші відносно меж полотна. Це важливо, оскільки початкові координати миші відносяться до всього вікна браузера, а для малювання вам потрібні координати всередині самого полотна. Після встановлення початкового положення, програма позначає малюнок як активний і зберігає цю початкову точку, щоб відстежувати, де починається малювання.

Коли користувач переміщує мишу, утримуючи кнопку натиснутою, функція малювання перебирає на себе створення візуальних позначок. Ця функція постійно відстежує положення миші відносно полотна. Якщо вибраним інструментом є олівець або гумка, вона малює невеликі відрізки ліній між попередньою і поточною точками миші, створюючи безперервний штрих. Колір змінюється залежно від того, чи користувач малює, чи стирає (білий для гумки, вибраний колір для олівця). Кожен з цих невеликих сегментів лінії зберігається у Firebase, включаючи всі необхідні дані, такі як координати, колір, розмір штриха, ідентифікатор користувача та мітку часу. Це дає змогу транслювати малюнок усім учасникам у режимі реального часу, гарантуючи, що кожен бачить те, що малюється, миттєво.

Якщо інструмент є інструментом фігури, наприклад, прямокутника або кола, малювання поводиться по-іншому. Замість того, щоб негайно зафіксувати фігуру на полотні, функція створює «попередній перегляд» фігури, який динамічно оновлюється під час переміщення миші. Вона використовує початкову точку

відліку і поточну позицію миші для обчислення розмірів і щоразу перемальовує все полотно, показуючи попередній перегляд фігури, але не зберігаючи її остаточно. Таким чином, користувачі отримують плавний попередній перегляд розміру і положення фігури в реальному часі перед тим, як завершити її створення.

Коли користувач відпускає кнопку миші, функція stopDrawing завершує дію. Для малювання від руки відрізки ліній вже зберігаються безперервно в тому вигляді, в якому вони були намальовані. Але для фігур ця функція обчислює остаточні розміри, виходячи з початкового і кінцевого положень миші, а потім надсилає ці дані до Firebase, офіційно додаючи фігуру до спільного малюнка. Попередній перегляд зникає, щоб повідомити, що фігуру створено.

Щоб візуально синхронізувати полотно з усіма діями користувачів, додаток використовує події бази даних Firebase у реальному часі. Коли додаток завантажується, функція життєвого циклу onMounted приєднує слухача для нових штрихів малювання. Щоразу, коли користувач додає штрих, програма отримує його і негайно малює на локальному полотні. Це гарантує, що навіть якщо користувач приєднався пізніше, він отримає повну поточну картину і продовжить спільну роботу, нічого не пропустивши.

Крім того, користувачі можуть очистити все полотно за допомогою функції clearCanvas. Ця функція стирає візуальний вміст з полотна і видаляє всі штрихи з Firebase, фактично скидаючи спільний простір для малювання для всіх. Користувачі також мають функцію undoStroke, яка дозволяє видалити останню дію малювання. Ця функція виконує пошук у зворотному порядку, щоб знайти останній штрих, намальований поточним користувачем, видаляє його з бази даних, оновлює локальний список штрихів, а потім перемальовує полотно, щоб відобразити зміни.

Collab Paint також включає систему чату. Функція sendMessage дозволяє користувачам надсилати текстові повідомлення іншим користувачам у кімнаті. Вона надсилає повідомлення, інформацію про відправника та мітку часу до Firebase, де слухач onChildAdded виявляє нові повідомлення і додає їх до інтерфейсу чату в

режимі реального часу. Щоб зробити інтерфейс чату зручним для користувача, він автоматично прокручується вниз, щоб показати останнє повідомлення.

Нарешті, користувачі, які керують сеансом малювання, можуть використовувати функцію deleteRoom, щоб видалити весь спільний простір. Після підтвердження наміру ця функція видаляє дані кімнати з Firebase і перенаправляє користувача до списку доступних кімнат, гарантуючи, що сеанс буде належним чином закрито і очищено.

Разом ці функції створюють бездоганне середовище для спільної роботи, де кілька користувачів можуть малювати, стирати, переглядати фігури, скасовувати свої дії, спілкуватися в чаті та керувати сеансами - і все це з оновленнями в реальному часі на основі Firebase, що робить Collab Paint динамічним та інтерактивним веб-додатком.

Другорядні функції Collab Paint

Окрім основних функцій малювання та співпраці, Collab Paint включає кілька другорядних функцій, які покращують загальний користувацький досвід і роблять програму більш універсальною, інтерактивною та зручною. Ці функції можуть бути не основними, але є важливими для плавної роботи та кращої зручності використання.

Однією з важливих другорядних функцій є зміна розміру та адаптивність полотна. Оскільки програма є веб-орієнтованою, і користувачі можуть отримувати до неї доступ на різних пристроях з різними розмірами екранів, вкрай важливо, щоб полотно динамічно підлаштовувалося під розмір дисплея. Це обробляється функцією, яка прослуховує події зміни розміру вікна та автоматично змінює розмір полотна відповідно. Коли розмір вікна змінюється, піксельні розміри полотна перераховуються для збереження чіткості та правильних пропорцій. Це запобігає спотворенням малювання та гарантує, що користувачі завжди мають чітке робоче місце правильного розміру незалежно від розміру екрана чи орієнтації. Зміна

розміру зазвичай запускається під час запуску програми та щоразу, коли змінюється розмір вікна браузера.

Ще однією тонкою, але важливою функцією є попередній перегляд малювання для фігур, таких як прямокутники та кола. Коли користувач перетягує мишу для створення фігури, програма не фіксує фігуру одразу на полотні чи в базі даних. Натомість вона використовує систему попереднього перегляду, яка візуально відображає тимчасовий контур або заповнену фігуру як зворотний зв'язок. Цей попередній перегляд оновлюється в режимі реального часу на основі положення курсора користувача, дозволяючи йому точно побачити, як виглядатиме фігура, перш ніж остаточно її намалювати. Ця функція підвищує точність і впевненість користувачів, запобігаючи постійному малюванню випадкових або неправильних фігур.

Ідентифікація користувачів та керування сеансами – ще одна ключова вторинна функція. Collab Paint відстежує, хто малює або надсилає повідомлення, пов'язуючи кожен штрих або повідомлення з унікальним ідентифікатором користувача, а іноді з іменами користувачів або електронними адресами. Це критично важливо для таких функцій, як скасування лише власних штрихів або персоналізація повідомлень чату. Програма керує станом користувача за допомогою реактивних змінних, які зберігають інформацію про поточного користувача, яка часто перевіряється під час надсилання або отримання штрихів або повідомлень. Ця система ідентифікації дозволяє персоналізувати керування та покращує співпрацю, чітко пов'язуючи дії з користувачами.

Сама функція скасування, хоча й пов'язана з малюванням, також є тонкою вторинною функцією, оскільки вона передбачає вибіркове видалення лише останнього штриха поточного користувача. Щоб реалізувати це, програма підтримує локальний список штрихів і переглядає минуле, щоб знайти найновіший штрих, виконаний користувачем. Потім цей штрих видаляється з серверної частини (Firebase), а полотно перемальовується без нього. Це вибіркове скасування

запобігає випадковому видаленню роботи інших користувачів і допомагає підтримувати цілісність співпраці.

Крім того, програма реалізує поведінку прокручування вниз у режимі реального часу в інтерфейсі чату. Коли нові повідомлення надходять з Firebase, вікно чату автоматично прокручується, щоб відобразити останні повідомлення, без необхідності вручну прокручувати вниз користувачеві. Ця невелика, але важлива функція забезпечує плавність розмов і гарантує, що користувачі ніколи не пропустять вхідні повідомлення. Вона працює, вимірюючи висоту прокручування контейнера повідомлень і оновлюючи позицію прокручування щоразу, коли додається нове повідомлення.

Діалогові вікна підтвердження для конфіденційних дій, таких як видалення кімнати або очищення полотна, також є другорядними, але критично важливими функціями. Ці діалогові вікна пропонують користувачам підтвердити свій намір перед виконанням незворотних дій. Це зменшує випадкову втрату даних і забезпечує захист від помилок користувача. Підтвердження реалізовано за допомогою простих модальних підказок, які зупиняють виконання, доки користувач не підтвердить або не скасує.

Ще одним покращенням є керування розміром штриха та кольором. Під час малювання користувачі можуть вибирати різні кольори та ширину ліній, які програма зберігає та застосовує послідовно. Це передбачає реактивні змінні стану, які оновлюються, коли користувачі вибирають різні параметри в інтерфейсі користувача. Коли штрихи малюються або зберігаються, ці налаштування додаються до даних штрихів, тому зовнішній вигляд залишається незмінним під час відтворення або перегляду іншими. Ця функція підвищує творчу гнучкість та дозволяє створювати більш насичені, персоналізовані малюнки.

Нарешті, програма може включати оптимізацію продуктивності, таку як обмеження або усунення подій малювання, щоб запобігти надмірному запису в базу даних або перемальовування полотна. Наприклад, малювання від руки швидко створює багато маленьких сегментів ліній, тому програма балансує плавне малювання з

ефективним використанням ресурсів, контролюючи частоту оновлення серверної частини або візуального полотна.

Підсумовуючи, ці вторинні функції — зміна розміру полотна, попередній перегляд фігур, відстеження користувачів, вибіркове скасування, чат з автоматичною прокруткою, діалогові вікна підтвердження, налаштовувані стилі штрихів та оптимізація продуктивності — працюють непомітно за лаштунками. Вони підвищують зручність використання, точність та співпрацю в Collab Paint, роблячи програму не лише функціональний, але й приємний та надійний для кількох користувачів, які працюють разом у режимі реального часу.

Ініціалізація для локального або серверного розгортання

В основі ініціалізації лежить налаштування змінних середовища та залежностей. Наприклад, оскільки Collab Paint використовує Firebase як бекенд для автентифікації бази даних та користувачів у режимі реального часу, потрібно надати деталі конфігурації Firebase, такі як ключі API, ідентифікатори проектів, URL-адреси бази даних та налаштування автентифікації. Ці значення зазвичай зберігаються у файлах середовища (файлах .env) або конфігурації на стороні сервера, що забезпечує безпеку конфіденційних облікових даних та їх відокремлення від вихідного коду.

Ця конфігурація створює надійну основу для запуску програми для спільного малювання як локально для розробки, так і віддалено для користувачів у всьому світі.

```
vuefire: {
  auth: {
    enabled: true,
    sessionCookie: false
  },
  config: {
    apiKey: process.env.NUXT_PUBLIC_FIREBASE_API_KEY,
    authDomain: process.env.NUXT_PUBLIC_FIREBASE_AUTH_DOMAIN,
    projectId: process.env.NUXT_PUBLIC_FIREBASE_PROJECT_ID,
    storageBucket: process.env.NUXT_PUBLIC_FIREBASE_STORAGE_BUCKET,
    messagingSenderId: process.env.NUXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID,
    appId: process.env.NUXT_PUBLIC_FIREBASE_APP_ID,
    measurementId: process.env.NUXT_PUBLIC_FIREBASE_MEASUREMENT_ID,
    databaseURL: process.env.NUXT_PUBLIC_FIREBASE_DATABASE_URL,
  }
}
```

Рис 2.11 – функція deleteRoom

2.6 Тестування та налагодження програмної розробки

Під час розробки CollabPaint ретельне тестування та налагодження були важливими для забезпечення високоякісного користувацького досвіду та надійної функціональності. Процес тестування включав комбінацію наскрізних (E2E) тестів з використанням Cypress, модульних тестів та сеансів ручного налагодження. E2E-тести були зосереджені на моделюванні реальних сценаріїв користувачів, щоб перевірити, чи працює програма належним чином від початку до кінця. Це включало тестування дій користувача, таких як вхід та реєстрація, приєднання або створення кімнат для малювання, вибір інструментів (олівець, гумка, прямокутник, коло) та виконання дій малювання на полотні. Тести також перевіряли співпрацю в реальному часі, імітуючи одночасное малювання кількох користувачів та забезпечуючи миттєве та правильне відображення штрихів для всіх учасників.

Особливу увагу було приділено тестуванню синхронізації між клієнтом та серверною частиною Firebase, підтверджуючи, що штрихи та повідомлення чату були належним чином надіслані, оновлені та видалені з бази даних. Були протестовані критично важливі функції, такі як скасування штрихів та очищення

полотна, щоб переконатися, що вони послідовно оновлюють як інтерфейс користувача, так і стан сервера. Тести також підтвердили, що елементи інтерфейсу користувача, такі як вибір інструментів, засоби вибору кольору та налаштування ширини лінії, працювали правильно та зберігали свій стан протягом сеансів.

Окрім автоматизованих тестів, налагодження виконувалося за допомогою інструментів розробника браузера для моніторингу слухачів подій, рендерингу полотна та мережевої активності. Правила та структура даних Firebase на стороні сервера регулярно переглядалися та тестувалися для забезпечення цілісності та безпеки даних. Продуктивність також контролювалася для виявлення та вирішення проблем затримки під час синхронізації малювання. Протягом розробки широко використовувалося ведення журналу для відстеження помилок та взаємодії з користувачами, що сприяло швидкому виявленню та вирішенню помилок. Ця комплексна система тестування та налагодження була вирішальною для забезпечення надійного, адаптивного та безперебійного спільного малювання для користувачів.

2.7. Рекомендації по впровадженню та використанню програмної розробки

Впровадження та використання програмного забезпечення здійснювалися відповідно до набору найкращих практик та перевірених рекомендацій, які забезпечили успіх проекту від початку до кінця. Спочатку було завершено комплексний етап планування та аналізу вимог, який включав детальні обговорення із зацікавленими сторонами для повного розуміння потреб користувачів та бізнес-цілей. Ця ретельна підготовка заклала міцну основу для розробки програмного забезпечення, яке ефективно вирішувало поставлені проблеми, чітко узгоджуючи очікування між усіма залученими сторонами.

Протягом усього процесу розробки було дотримування встановлених стандартів кодування та підтримувала чистий, узгоджений код. Системи контролю версій ефективно використовувалися для управління змінами, сприяння співпраці та

забезпечення швидкого відкату за необхідності. Архітектура програмного забезпечення була розроблена модульною та масштабованою, що дозволяло легко вносити вдосконалення в майбутньому та гарантувало, що система може зростати без необхідності значних переписів.

Тестування було безперешкодно інтегровано в робочий процес розробки з самого початку. Були впроваджені автоматизовані модульні, інтеграційні та системні тести, що дозволило виявляти дефекти на ранній стадії та підтримувати високу якість коду. Регулярно проводилися експертні перевірки коду, що сприяло обміну знаннями та запобігало багатьом потенційним проблемам до того, як вони потрапили у виробництво.

дозволило швидко виявляти та вирішувати будь-які проблеми під час виконання.

Впровадження користувачами підтримувалося за допомогою добре підготовлених навчальних матеріалів, включаючи посібники та навчальні посібники, які допомогли користувачам швидко опанувати програму. Були створені канали зворотного зв'язку, що надавали цінну інформацію, яка спрямовувала подальші вдосконалення. Заходи безпеки були вбудовані протягом усього життєвого циклу програмного забезпечення, включаючи безпечну автентифікацію, перевірку вхідних даних, шифрування даних та регулярні оновлення залежностей для зменшення вразливостей.

Вимоги на стороні клієнта були ретельно розглянуті, щоб забезпечити оптимальний користувацький досвід на різних пристроях та браузерах. Програмне забезпечення було оптимізовано для продуктивності, швидкодії та доступності, забезпечуючи безперебійну взаємодію навіть на низькопродуктивному обладнанні та в різних мережевих умовах. Були проведені тести на сумісність браузерів, щоб гарантувати стабільну функціональність, а легкі фреймворки та ефективний код мінімізували час завантаження та споживання ресурсів.

Крім того, були впроваджені процедури резервного копіювання та аварійного відновлення для захисту цілісності даних та забезпечення доступності системи. Програмне забезпечення перейшло у фазу обслуговування, де постійні оновлення виправляли помилки, оптимізували продуктивність та адаптували програму до

змінюваних вимог користувачів та технологічних змін. Відкрита комунікація між розробниками, користувачами та зацікавленими сторонами сприяла постійному вдосконаленню та забезпечувала актуальність та надійність програмного забезпечення.

На завершення, весь процес розробки та впровадження програмного забезпечення був виконаний ретельно та професіонально, що призвело до створення високоякісного, безпечного та зручного для користувача продукту. Успіх проекту став прямим результатом ретельного планування, дисциплінованого кодування, інтегрованого тестування, автоматизованого розгортання, комплексної підтримки користувачів та проактивного обслуговування — все це з урахуванням потреб клієнта для забезпечення відмінного досвіду кінцевого користувача.

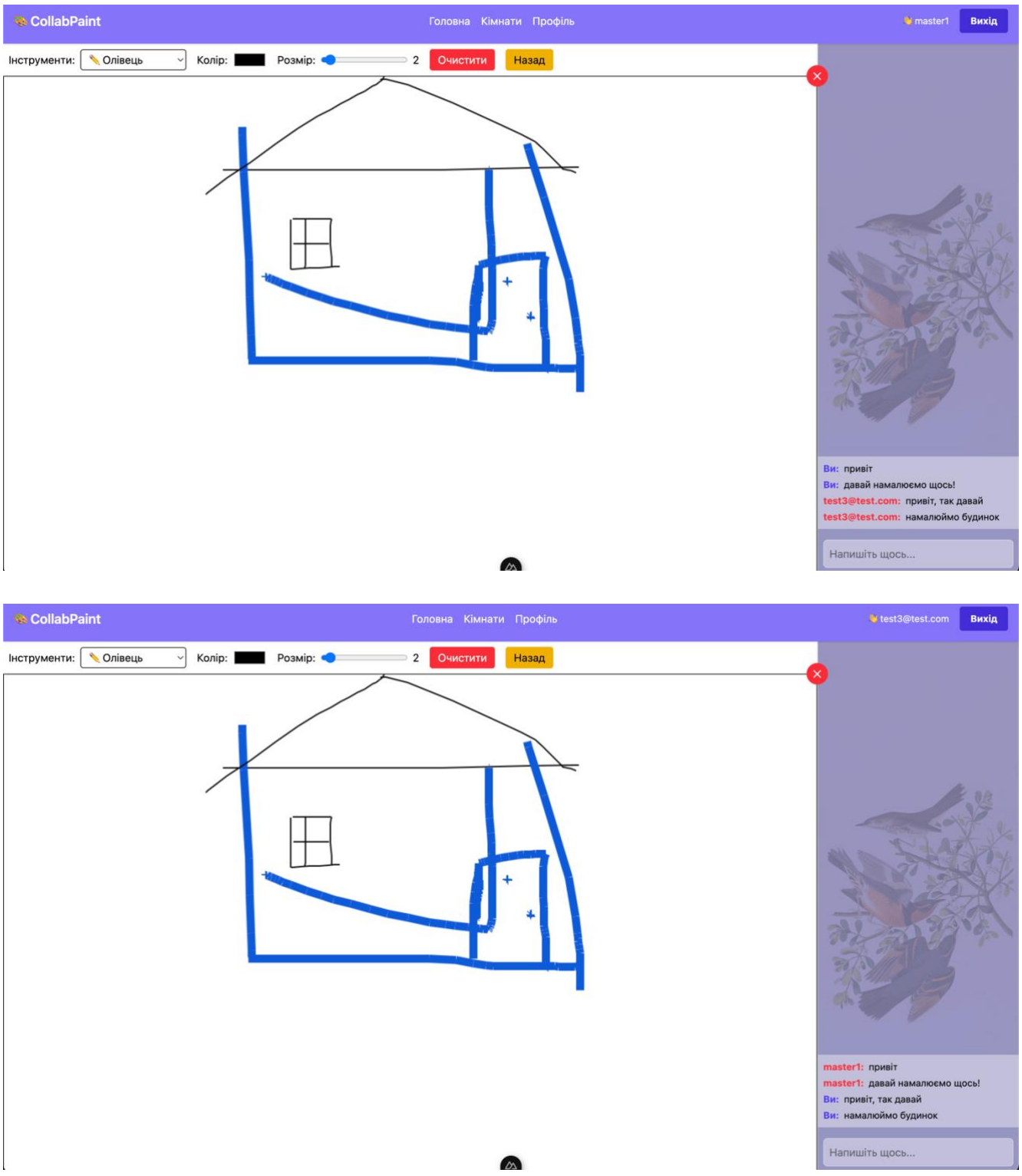

Рис. 2.12(а, б) – приклад співпраці двох юзерів у Collab Paint

ВИСНОВОК

Впровадження розробки програмного забезпечення пройшло успішно, продемонструвавши найкращі практики, що призвели до створення надійного, зручного в обслуговуванні та користувача продукту. Продумана інтеграція планування, тестування, розгортання та підтримки користувачів створила міцну основу для постійної еволюції програмного забезпечення та його практичного застосування. Такий підхід не лише виконав початкові цілі проекту, але й встановив сталий робочий процес для майбутньої розробки та вдосконалення. Пильна увага до відгуків користувачів та постійне вдосконалення забезпечили актуальність та ефективність програмного забезпечення у задоволенні потреб користувачів. Крім того, акцент на безпеці та продуктивності створив надійне середовище, що є важливим для ширшого впровадження. Поєднання автоматизованих процесів та ретельної документації спростило розробку та сприяло безперебійному впровадженню нових учасників. Загалом, проект продемонстрував, як добре організований та дисциплінований життєвий цикл розробки програмного забезпечення може призвести до високоякісного рішення, яке є адаптивним, масштабованим та готовим до довгострокового успіху в динамічному технологічному середовищі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mozilla Developer Network (MDN) — Canvas API. Детальна документація та приклади використання Canvas API для 2D-малювання у веббраузерах. https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API
2. Документація бази даних Firebase Realtime. Офіційний посібник з використання бази даних Firebase Realtime для зберігання та синхронізації даних програм у режимі реального часу. https://firebase.google.com/docs/database
3. Автентифікація Firebase. Документація з автентифікації Firebase для обробки входу користувачів та безпеки. https://firebase.google.com/docs/auth
4. Довідник Firebase JavaScript SDK. Довідники API для взаємодії з сервісами Firebase з веб-застосунків. https://firebase.google.com/docs/reference/js
5. Офіційна документація Socket.IO. Інформація про налаштування двонаправленого зв'язку між клієнтами та сервером у режимі реального часу. https://socket.io/docs/v4/
6. Офіційна документація Node.js.Вичерпна документація API Node.js, що використовується для створення бекенд-сервера. https://nodejs.org/en/docs/
7. Посібник з Vue.js. Офіційна документація Vue.js, що використовується для розробки реактивного інтерфейсу фронтенду. https://vuejs.org/guide/introduction.html
8. W3Schools — Підручник з HTML5 Canvas.Посібник з малювання за допомогою елемента HTML5 Canvas, зручний для початківців. https://www.w3schools.com/html/html5_canvas.asp

9. Stack Overflow — Програми для спільного малювання. Різні питання та рішення спільноти щодо проблем у впровадженні функцій спільного малювання. https://stackoverflow.com/questions/tagged/canvas+collaboration
10. Eloquent JavaScript (онлайн-книга). Поглиблені пояснення та практичні приклади концепцій JavaScript, що стосуються малювання та обробки подій. https://eloquentjavascript.net/
11. WebSocket API — MDN. Документація щодо WebSocket для зв'язку в режимі реального часу через TCP. https://developer.mozilla.org/en-US/docs/Web/API/WebSocket
12. Правила безпеки Firebase. Посібники щодо захисту доступу до бази даних Firebase у реальному часі з правилами захисту даних користувачів. https://firebase.google.com/docs/database/security

АНОТАЦІЯ

Коніщук О. М. – **CollabPaint: Розробка веб-платформи для малювання онлайн**

Кваліфікаційна робота за спеціальністю 122 Комп'ютерні науки. – Волинський національний університет імені Лесі Українки, Луцьк, 2025.

Дипломна робота присвячена розробці та дослідженню веб-додатку для спільного малювання - CollabPaint. Цей проект досліджує сучасні технології створення односторінкових додатків (SPA), які підтримують взаємодію між користувачами в режимі реального часу через Інтернет. У роботі проаналізовано теоретичні та практичні аспекти побудови таких систем з акцентом на спільному редагуванні, синхронізації даних у реальному часі та розробці адаптивного користувацького інтерфейсу.

Кінцевий програмний продукт, CollabPaint, - це сучасний веб-додаток для малювання, який дозволяє декільком користувачам одночасно взаємодіяти на спільному полотні. Додаток створено з використанням Vue.js для фронтенду, Tailwind CSS для стилізації та HTML5 Canvas для графічного рендерингу. Співпраця в реальному часі реалізована через базу даних Firebase Realtime Database, яка забезпечує зв'язок з низькою затримкою та синхронізацію дій малювання та повідомлень у чаті.

У цій роботі описано всі етапи розробки програмного забезпечення - від дослідження ринку та технологій, через дизайн інтерфейсу та реалізацію, до тестування та налагодження. В результаті розроблена система демонструє стабільне та ефективне середовище для спільного малювання, яке можна використовувати для творчої роботи, освіти або візуальної комунікації.

Ключові слова: веб-додаток для спільної роботи, малювання в реальному часі, Vue.js, Firebase, HTML5 Canvas, онлайн, односторінковий додаток (SPA)

ABSTRACT

Konishchuk O. M. **– CollabPaint: Development of a web platform for online drawing**

Qualification work in the specialty 122 Computer Science. – Lesya Ukrainka Volyn National University, Lutsk, 2025.

The thesis is devoted to the development and research of a web application for collaborative drawing – CollabPaint. This project explores modern technologies for creating single-page applications (SPA) that support real-time interaction between users via the Internet. The thesis analyzes the theoretical and practical aspects of building such systems with an emphasis on collaborative editing, real-time data synchronization, and the development of an adaptive user interface.

The final software product, CollabPaint, is a modern web application for drawing that allows multiple users to interact simultaneously on a shared canvas. The application was created using Vue.js for the frontend, Tailwind CSS for styling, and HTML5 Canvas for graphics rendering. Real-time collaboration is implemented through the Firebase Realtime Database, which provides low-latency communication and synchronization of drawing actions and chat messages.

This work describes all stages of software development, from market and technology research, through interface design and implementation, to testing and debugging. The resulting system demonstrates a stable and efficient environment for collaborative drawing that can be used for creative work, education, or visual communication.

Keywords: collaborative web application, real-time drawing, Vue.js, Firebase, HTML5 Canvas, online, single-page application (SPA)