

Національний університет «Полтавська політехніка імені Юрія Кондратюка»  
Навчально-науковий інститут інформаційних технологій та робототехніки  
Кафедра комп'ютерних та інформаційних технологій і систем

**Пояснювальна записка  
до дипломного проекту (роботи)  
магістра**

на тему  
«Створення онлайн інструменту для графічної обробки фото»

Виконав: студент 2 курсу, групи 601-ТН  
спеціальності 122 Комп'ютерні науки  
Острянин Вадим Володимирович  
Керівник Васюта В.В.  
Рецензент Підяшенко В.Є.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І  
СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

**спеціальність 122 «Комп'ютерні науки»**

**на тему**

**«Створення онлайн інструменту для графічної обробки фото»»**

**Студента групи 601-ТН Острянина Вадима Володимировича**

Керівник роботи кандидат  
технічних наук, доцент  
Васюта В.В.

Консультант кандидат технічних  
наук, доцент Васюта В.В.

Завідувач кафедри кандидат  
фізико-математичних наук,  
Двірна О.А.

## РЕФЕРАТ

ДР: 53 с., 15 рис., 15 джерел

*Ключові слова:* Редактор зображень із штучним інтелектом, веб-редагування зображень, сучасна веб-розробка, Інтерактивні інтерфейси користувача, React.js, TypeScript, Fabric.js, Tailwind CSS, Radix UI, React Hooks, Llava model, Ollama AI

Розробка передових веб-додатків досягла безпрецедентних висот, інтегруючи штучний інтелект (ШІ) для переосмислення досвіду користувачів. Цей проект зосереджений на створенні веб-редактора зображень з розширенням у вигляді функції аналізу зображення штучним інтелектом. Завдяки використанню надійного технічного стеку та інноваційним функціям цей редактор зображень призначений як для творчих професіоналів, так і для ентузіастів, забезпечуючи ефективно редагування.

Редактор зображень працює на основі ретельно підібраного технічного стеку для забезпечення продуктивності, масштабованості та взаємодії з користувачем. React.js у поєднанні з TypeScript утворює основу додатка, забезпечуючи надійну та зручну для обслуговування архітектуру. Для роботи з полотном Fabric.js пропонує потужні інструменти для візуалізації та взаємодії, тоді як Tailwind CSS забезпечує гнучкий стиль для візуально привабливого та адаптивного дизайну. Інтеграція можливостей штучного інтелекту стала можливою завдяки Ollama, яка використовує модель Llava для розширеного аналізу та покращення зображень. Щоб забезпечити інтуїтивно зрозуміле керування станом, використовуються React Hooks, тоді як Radix UI містить багаторазово доступні та настроювані компоненти інтерфейсу користувача.

Додаток може похвалитися набором інноваційних функцій, спрямованих на покращення досвіду редагування. Удосконалена система пензлів включає різні типи пензлів. Керування шарами з функцією перетягування спрощує складні редагування,

забезпечуючи гнучкість і контроль. Фільтри та ефекти зображень у режимі реального часу дають змогу користувачам динамічно експериментувати, а аналіз зображень за допомогою штучного інтелекту надає розумні пропозиції для вдосконалення. Редактор підтримує темні/світлі теми, задовольняючи вподобання користувача та зменшуючи навантаження на очі. Спеціальний засіб вибору кольорів із збереженими палітрами підвищує продуктивність, а можливості імпорту/експорту зображень забезпечують сумісність із різними форматами файлів.

Хоча поточна реалізація пропонує надійний набір функцій, майбутні розробки спрямовані на подальше покращення її функціональності. Запровадження співпраці в режимі реального часу дозволить декільком користувачам одночасно редагувати одне зображення, сприяючи командній роботі та творчості. Інтеграція хмарного сховища забезпечить плавний доступ до проектів на різних пристроях. Розширення можливостей штучного інтелекту може включати автоматичне видалення об'єктів, заповнення з урахуванням вмісту та вдосконалене відновлення фотографій. Забезпечення швидкості реагування на мобільні пристрої розширить доступність, а система плагінів дозволить користувачам розширювати функції програми відповідно до своїх потреб.

Веб-редактор зображень, котрий розширений штучним інтелектом, являє собою значний стрибок у сфері онлайн-інструментів редагування зображень. Поєднуючи найсучасніші технології з інноваційними функціями та дизайном, орієнтованим на користувача. Цей проект відповідає зростаючому попиту на потужні, доступні та інтелектуальні платформи редагування. Його зосередженість на дослідженнях і перспективному розвитку гарантує, що він є не лише інструментом для сьогодення, але й основою для нових потреб творчої спільноти.

## ESSAY

Page Count: 53, Figures: 15, References: 15

*Keywords:* AI-Powered Image Editor, Web Image Editing, Modern Web Development, Interactive User Interfaces, React.js, TypeScript, Fabric.js, Tailwind CSS, Radix UI, React Hooks, Llava Model, Ollama AI

The development of advanced web applications has reached unprecedented heights by integrating artificial intelligence (AI) to redefine user experiences. This project focuses on creating a web-based image editor with an AI-powered image analysis feature. Leveraging a robust tech stack and innovative functionalities, this image editor is designed for both creative professionals and enthusiasts, enabling efficient editing.

The image editor operates on a carefully selected tech stack to ensure performance, scalability, and seamless user interaction. React.js combined with TypeScript forms the foundation of the application, delivering a reliable and maintainable architecture. Fabric.js provides powerful tools for canvas rendering and interaction, while Tailwind CSS ensures flexible styling for visually appealing and responsive designs. AI capabilities are integrated via Ollama, employing the Llava model for advanced image analysis and enhancement. React Hooks enable intuitive state management, while Radix UI offers reusable and customizable user interface components.

The application boasts a range of innovative features aimed at enhancing the editing experience. An advanced brush system includes various brush types, while layer management with drag-and-drop functionality simplifies complex edits, providing flexibility and control. Real-time image filters and effects allow dynamic experimentation, and AI-powered image analysis offers intelligent suggestions for improvements. The editor supports dark/light themes to cater to user preferences and reduce eye strain. A dedicated color picker with saved palettes enhances productivity, while image import/export capabilities ensure compatibility with various file formats.

While the current implementation delivers a robust feature set, future developments aim to further enhance its functionality. Real-time collaboration will enable multiple users to edit the same image simultaneously, fostering teamwork and creativity. Cloud storage integration will ensure seamless access to projects across devices. Expanded AI capabilities may include automatic object removal, content-aware fill, and advanced photo restoration. Optimizing responsiveness for mobile devices will broaden accessibility, and a plugin system will empower users to extend the app's features to meet their specific needs.

The AI-augmented web image editor represents a significant leap forward in online image editing tools. By combining cutting-edge technologies with innovative features and user-centric design, this project meets the growing demand for powerful, accessible, and intelligent editing platforms. Its focus on research and forward-thinking development ensures that it serves not only as a tool for today but also as a foundation for evolving creative needs in the future.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	9
ВСТУП.....	10
РОЗДІЛ 1 СФЕРА ЗАСТОСУВАННЯ .....	11
РОЗДІЛ 2 АНАЛІЗ ЗАСТОСУНКІВ З РЕДАГУВАННЯ ЗОБРАЖЕНЬ ТА ДИЗАЙНУ .....	13
2.1 Adobe Photoshop (веб-версія) .....	13
2.2 Canva .....	14
2.3 Figma .....	15
2.4 Photopea.....	16
2.5 Krita (веб-бета).....	17
РОЗДІЛ 3 МОДЕЛЬ ТА МЕТОД РЕДАГУВАННЯ ЗОБРАЖЕНЬ .....	19
3.1 Маніпулювання зображеннями на основі полотна.....	19
3.2 Розширена система пензлів.....	22
3.3 Фільтри та ефекти зображень у реальному часі .....	24
3.4 Основні функції аналізу зображень за допомогою AI .....	26
3.5 Спеціальне керування кольором.....	29
3.6 Підтримка імпорту/експорту та форматування файлів .....	30
РОЗДІЛ 4 ОГЛЯД ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ЩОДО ПРОЄКТУВАННЯ ТА РОЗРОБКИ ПРОГРАМНИХ ПРОДУКТІВ .....	33
4.1 Засоби розробки інтерфейсу .....	33
4.2 Інструменти інтеграції ШІ .....	34
4.3 Інструменти розробки та проектування .....	35
4.4 Інструменти тестування та налагодження .....	35
4.5 Інструменти оптимізації продуктивності.....	36
4.6 Інструменти розгортання та масштабованості .....	36
РОЗДІЛ 5 ОПИС РОЗРОБЛЕНОГО В РАМКАХ МАГІСТЕРСЬКОЇ ДИПЛОМНОЇ РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	38
5.1 Мета і завдання роботи .....	38

5.2 Вимоги до програмного забезпечення .....	39
5.3 Опис результатів розробки програмного забезпечення .....	39
РОЗДІЛ 6 ІНТЕРФЕЙС КОРИСТУВАЧА.....	41
ВИСНОВКИ .....	52
СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ .....	53

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

**ШІ** – штучний інтелект.

**ІС** – інформаційна система.

**ПЗ** – програмне забезпечення.

**ОС** – операційна система.

**UI** – користувацький інтерфейс.

**UX** – користувацький досвід.

**PNG** – Portable Network Graphics.

**JPEG** – Joint Photographic Experts Group.

**SVG** – Scaleable Vector Graphic.

## ВСТУП

В епоху, коли процвітає цифрова творчість, зріс попит на потужні, але доступні інструменти для підтримки художнього вираження та співпраці. Редагування зображень, яке колись обмежувалося спеціальним програмним забезпеченням на спеціальних машинах, перетворилося на веб-діяльність, доступну практично з будь-якого місця. Проект веб-редактора зображень із розширенням штучним інтелектом має на меті сприяти цій трансформації, поєднуючи передові технології з інтуїтивно зрозумілими принципами дизайну.

Використовуючи сучасний стек веб-розробки, включаючи React.js, TypeScript, Fabric.js і Tailwind CSS, цей проект пропонує передове рішення як для окремих творців.

Основні інновації редактора, такі як аналіз зображень за допомогою штучного інтелекту, передові системи пензлів і динамічне керування шарами, демонструють його зосередженість на покращенні взаємодії з користувачем. Завдяки дослідженням таких областей, як оптимізація продуктивності та сучасна архітектура додатків, проект не лише забезпечує негайну цінність, але й закладає основу для майбутніх досягнень.

## РОЗДІЛ 1

### СФЕРА ЗАСТОСУВАННЯ

Веб-редактор зображень, розширений штучним інтелектом, створений для широкого кола користувачів і галузей, пропонуючи різноманітну функціональність і адаптивність. Сфера його застосування охоплює різні сфери, що робить його цінним інструментом для творчих професіоналів, педагогів і команд. Нижче наведено ключові сфери, де програма може мати значний вплив:

#### 1) Творчі індустрії та індустрії дизайну:

- графічні дизайнери: дозволяють детально маніпулювати зображеннями, накладати шари та точне редагування, що є важливим для брендингу, реклами та проектів веб-дизайну;

- ілюстратори та цифрові художники: пропонує розширені системи пензлів і спеціальні засоби вибору кольорів для легкого створення складних ілюстрацій;

- фотографи: надає інструменти на основі штучного інтелекту для ретушування фотографій, покращення та застосування ефектів.

#### 2) Освітньо-тренінгові платформи:

Навчання мистецтву та дизайну: забезпечує практичну навчальну платформу для студентів, щоб практикувати цифрове мистецтво та методи дизайну.

#### 3) Електронна комерція та створення контенту:

- інтернет-магазини: допомагає редагувати фотографії продуктів для списків за допомогою швидких фільтрів, ефектів;

- творці вмісту та впливові особи: спрощує створення привабливих візуальних елементів для соціальних мереж, блогів і мініатюр відео.

#### 4) Дослідження та розробки:

Дослідження штучного інтелекту та обробки зображень: служить випробувальним майданчиком для експериментів із моделями штучного інтелекту, методами покращення зображення та технологіями обробки полотна.

#### 5) Кросплатформний доступ:

Гнучкість на основі браузера: усуває необхідність інсталяції спеціального програмного забезпечення, роблячи його доступним з будь-якого пристрою з браузером.

Задовольняючи потреби різноманітних груп користувачів і галузей, цей проект підкреслює потенціал поєднання ШІ та технологій редагування зображень для створення універсальної та масштабованої платформи для редагування зображень. Його адаптивність забезпечує відповідність як поточним, так і новим цифровим творчим робочим процесам.

## РОЗДІЛ 2

### АНАЛІЗ ЗАСТОСУНКІВ З РЕДАГУВАННЯ ЗОБРАЖЕНЬ ТА ДИЗАЙНУ

Розробка веб-редактора зображень із розширенням штучним інтелектом потребує комплексного аналізу існуючих рішень, щоб визначити їхні сильні та слабкі сторони та області, які потрібно вдосконалити. У цьому розділі розглядаються відомі аналоги у сферах веб-редагування зображень і інструментів для спільної роботи, надаючи розуміння того, чим запропонована система вирізняється.

#### 2.1 Adobe Photoshop (веб-версія)

Сильні сторони:

- можливості редагування галузевого стандарту з розширеними функціями, такими як керування шарами та широкий вибір пензлів;
- хмарна інтеграція для доступності на різних пристроях;
- добре розроблений інтерфейс користувача та обширна документація.

Слабкі сторони:

- крута крива навчання для нових користувачів;
- обмежені можливості співпраці в реальному часі у веб-версії;
- ціни на основі передплати можуть бути доступними не всім користувачам.

На рисунку 2.1 можна побачити головну сторінку застосунку [1].

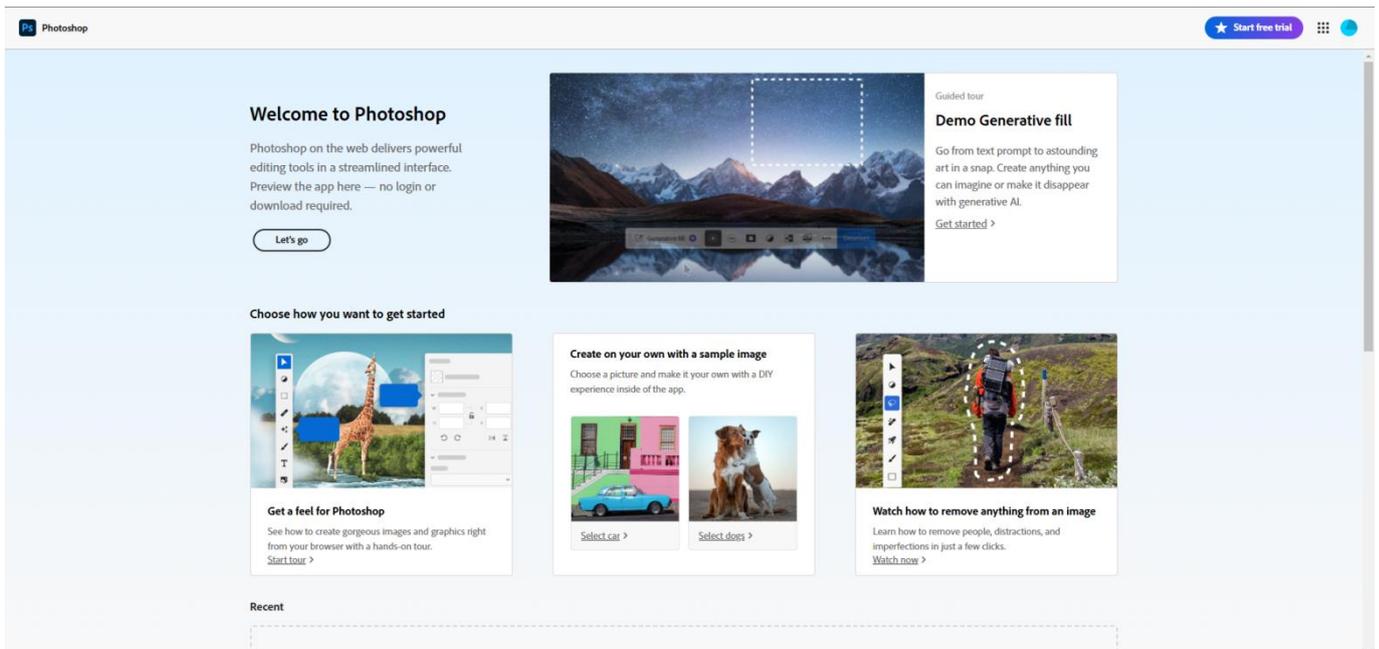


Рисунок 2.1 - Приклад сторінки у «Adobe Photoshop (веб-версія)»

Пропонований редактор може націлюватися на покращення доступності та зручності використання, полегшуючи використання розширених функцій для початківців. Співпраця в реальному часі є ключовою відмінністю.

## 2.2 Canva

Сильні сторони:

- зручний інтерфейс для початківців;
- сильні функції співпраці з редагуванням у режимі реального часу для кількох користувачів;
- попередньо розроблені шаблони для швидкого створення вмісту.

Слабкі сторони:

- обмежені розширені функції обробки зображень (наприклад, без спеціальних пензлів або детального керування шарами).
- більше покладається на шаблони, ніж на творчі інструменти вільної форми.
- можливості штучного інтелекту відносно базові, зосереджені на створенні тексту та пропозиціях дизайну.

На рисунку 2.2 можна побачити сторінку з шаблонами застосунку [2].

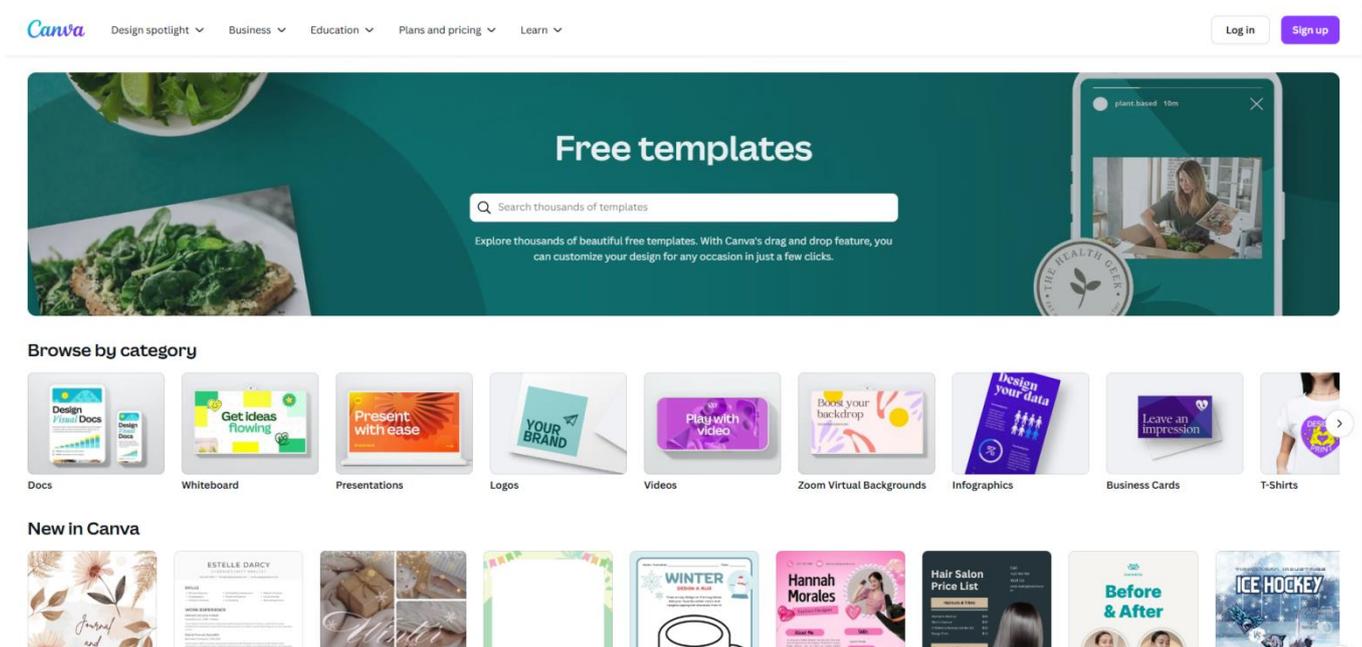


Рисунок 2.2 - Приклад сторінки з шаблонами у «Canva»

Незважаючи на те, що Canva вирізняється простотою використання та співпраці, відсутність розширених інструментів редагування дає можливість запропонованому редактору задовольнити більш досвідчених користувачів без шкоди для зручності використання.

## 2.3 Figma

Сильні сторони:

- найкращі в своєму класі функції співпраці в режимі реального часу для дизайнерських проєктів;
- хмарна платформа, доступна з будь-якого браузера;
- система проєктування на основі компонентів для модульних робочих процесів.

Слабкі сторони:

- зосереджено на дизайні UI/UX, не призначеному для редагування растрових зображень або функцій, керованих штучним інтелектом;

- обмежені засоби налаштування пензля та обробки зображень.

На рисунку 2.3 можна побачити сторінку з робочим простором застосунку[3].

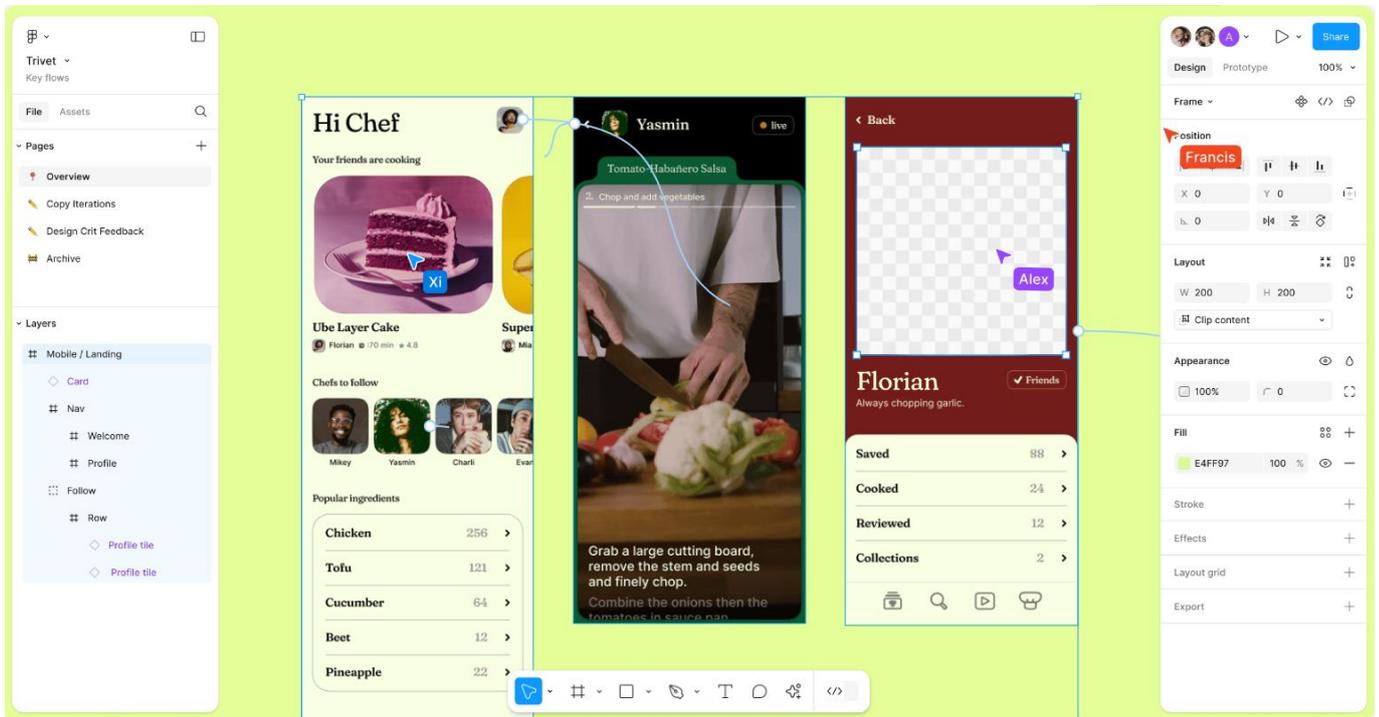


Рисунок 2.3 - Приклад сторінки з шаблонами у «Figma»

Пропонований редактор може прийняти структуру співпраці в режимі реального часу Figma, розширюючи свою увагу до растрового редагування з надійною інтеграцією ШІ.

## 2.4 Photopea

Сильні сторони:

- повністю базується на браузері та є безкоштовним у використанні;
- пропонує роботу, подібну до Photoshop, із підтримкою шарів, маскуванням і розширеними інструментами редагування;
- підтримує кілька форматів файлів, включаючи PSD.

Слабкі сторони:

- немає функцій на основі ШІ для інтелектуального редагування чи аналізу;

- інтерфейс не настільки інтуїтивно зрозумілий для непрофесійних користувачів;
- функції співпраці відсутні.

На рисунку 2.4 можна побачити сторінку з робочим простором застосунку[4].

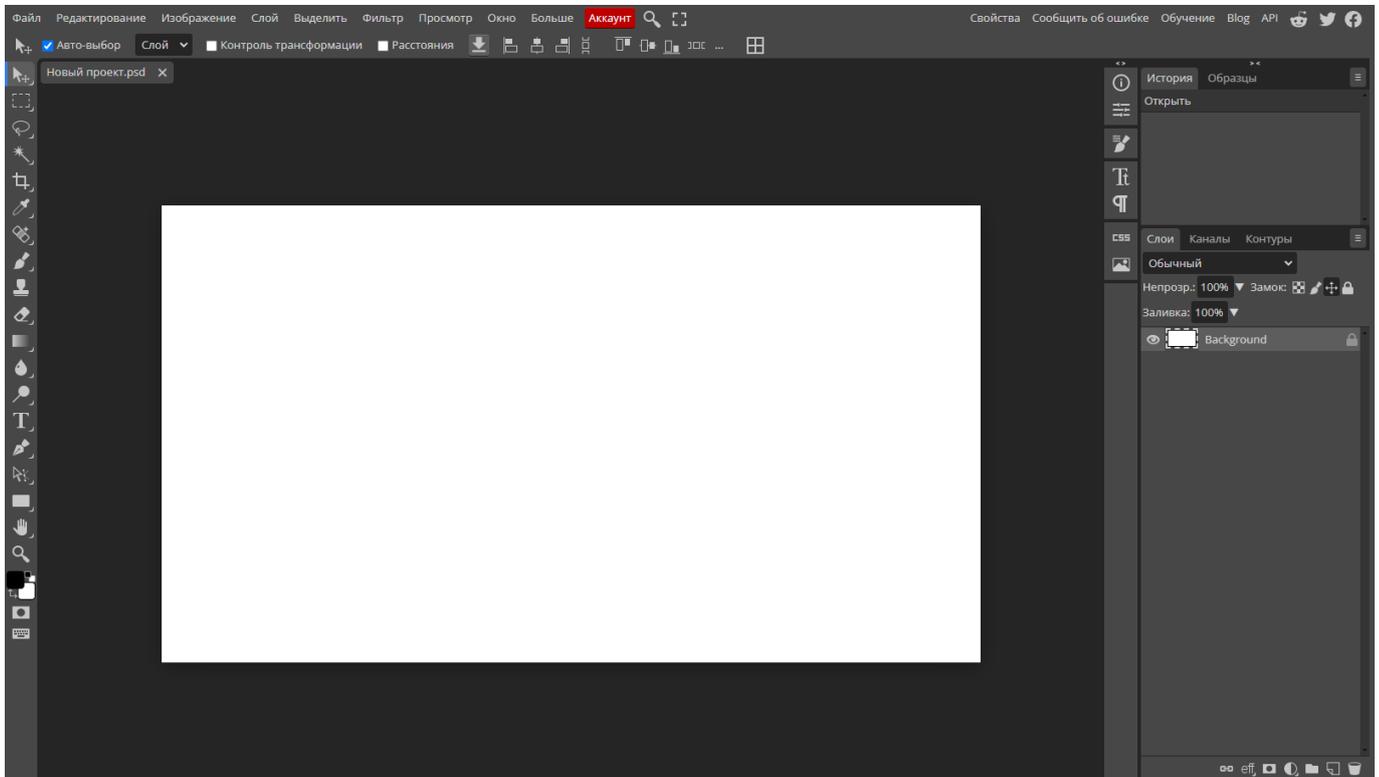


Рисунок 2.4 - Приклад сторінки з новим проектом у «Photoroa»

Хоча Photoroa є сильним аналогом з точки зору функціональності на основі браузера, відсутність штучного інтелекту та функцій співпраці залишає місце для більш багатofункціональної альтернативи.

## 2.5 Krita (веб-бета)

Сильні сторони:

- зосередьтеся на розширених художніх інструментах, зокрема пензлях, які можна налаштувати;
- підтримує багатoshарове редагування та детальні художні робочі процеси;
- платформа з відкритим кодом і активною спільнотою.

Слабкі сторони:

- веб-версія все ще є експериментальною та не має повної функціональності порівняно з версією для комп'ютера;
- немає вбудованих можливостей AI або функцій співпраці.

На рисунку 2.5 можна побачити сторінку з робочим простором за стосунку[5].

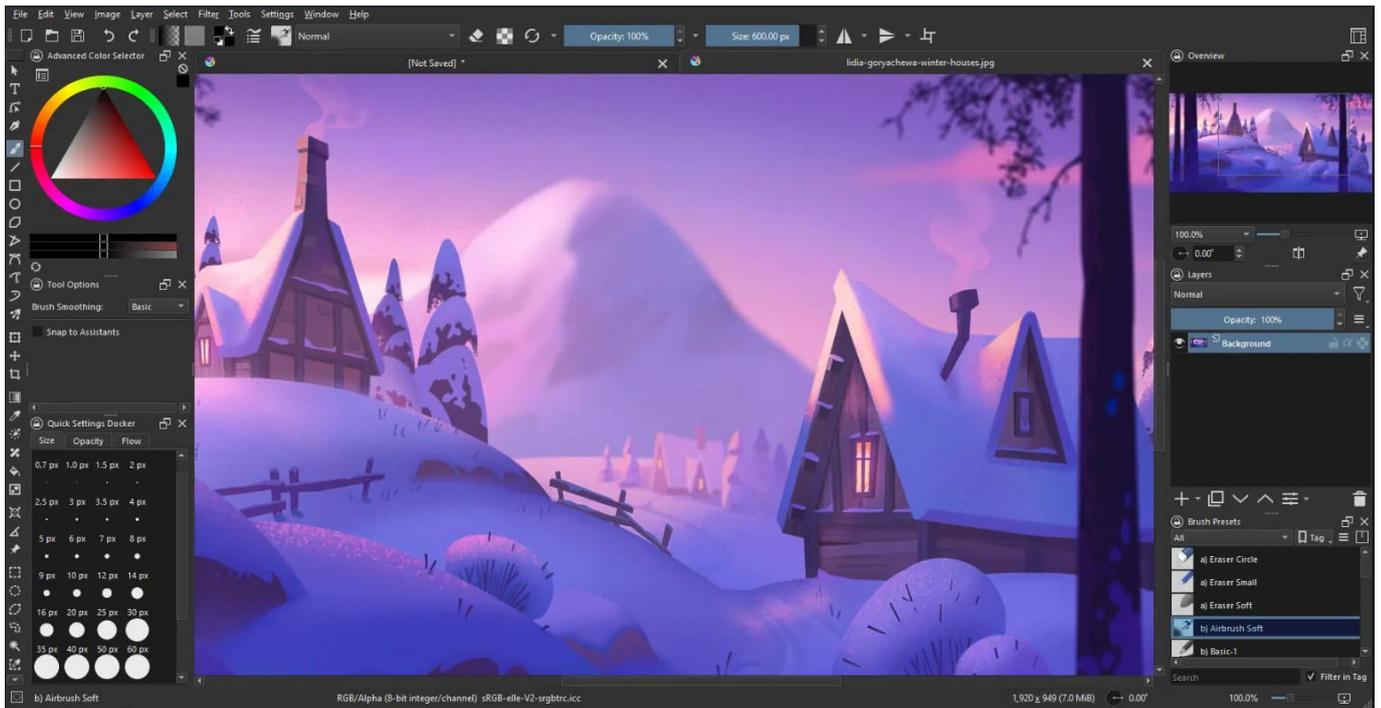


Рисунок 2.5 - Приклад сторінки з проектом у «Krita»

Зосередженість Krita на професійних художніх інструментах узгоджується з частиною цілей запропонованої системи. Однак додавання функцій, керованих штучним інтелектом, і функціональних можливостей для співпраці робить запропонований редактор помітним.

## РОЗДІЛ 3

### МОДЕЛЬ ТА МЕТОД РЕДАГУВАННЯ ЗОБРАЖЕНЬ

#### 3.1 Маніпулювання зображеннями на основі полотна

Маніпулювання зображеннями на основі полотна є основою веб-редактора зображень, удосконаленого штучним інтелектом. Використовуючи бібліотеку Fabric.js, цей підхід дозволяє редактору виконувати складні завдання редагування зображень із гнучкістю, ефективністю та високою продуктивністю в веб-середовищі. Елемент HTML5 <canvas> служить основою для візуалізації та обробки зображень у динамічний та інтерактивний спосіб. Нижче наведено більш детальне дослідження основних принципів і методів, які використовуються в маніпуляції зображеннями на основі полотна в рамках проекту.

#### **Об'єктно-орієнтоване редагування зображень**

У маніпуляції зображеннями на основі полотна кожен візуальний елемент на полотні розглядається як окремий об'єкт. Цей об'єктно-орієнтований підхід дозволяє редактору легко маніпулювати зображеннями, формами та текстовими елементами, не змінюючи все полотно. Основні переваги цього методу:

- модульне редагування: кожен елемент є незалежним об'єктом (наприклад, зображення, фігура або текст), який можна вибирати, переміщувати, повертати, масштабувати або видаляти незалежно від інших;
- керування шарами: організовуючи об'єкти в шари, користувачі можуть застосовувати різні операції редагування до різних елементів, не впливаючи на інші. Шари можна змінити, заблокувати або приховати, щоб полегшити створення складних композицій;
- інтерактивне виділення та трансформація: користувачі можуть безпосередньо взаємодіяти з об'єктами, вибираючи та застосовуючи такі трансформації, як обертання, зміна розміру або гортання за допомогою інтуїтивно зрозумілої функції перетягування.

## Трансформації полотна

Перетворення полотна дозволяють маніпулювати зображеннями та об'єктами в реальному часі. За допомогою Fabric.js різні геометричні трансформації можна застосовувати до об'єктів на полотні, роблячи процес редагування гнучким та інтерактивним:

- масштабування: об'єкти можна змінювати пропорційно або непропорційно, перетягуючи їх обмежувальні рамки або використовуючи спеціальні елементи керування введенням;
- обертання: обертання об'єкта можна здійснити, перетягнувши ручку обертання, увімкнувши обертання у довільній формі або на фіксований кут. Це перетворення є важливим для таких завдань, як вирівнювання зображення та графічний дизайн;
- перекис і викривлення: об'єкти можна перекошувати вздовж осей X і Y, щоб створити ефекти перспективи або відкоригувати кут зображення чи фігури;
- перевертання: простий спосіб дзеркального відображення об'єктів горизонтально чи вертикально, що особливо корисно для таких завдань, як дублювання об'єктів або дизайн на основі симетрії.

Ці перетворення надають дизайнерам високу гнучкість під час компоновання та редагування зображень, дозволяючи як прості налаштування, так і складні спотворення.

## Багаторівневе редагування зображень

Нанесення шарів є важливою технікою редагування зображень, що дозволяє користувачам керувати різними компонентами композиції окремо. Маніпуляції на основі Canvas підтримують керування шарами, що дає змогу розташовувати об'єкти в певному порядку:

- упорядкування шарів: користувачі можуть відправляти об'єкти на передню або задню частину стека, забезпечуючи відображення потрібних елементів зверху або позаду інших, якщо потрібно;
- групування: кілька об'єктів можна згрупувати разом, що дозволяє одночасно маніпулювати кількома елементами одночасно. Наприклад, користувач

може згрупувати всі компоненти логотипу та змінити розмір або перемістити всю групу, не змінюючи окремих частин;

- **видимість шару:** шари можна зробити видимими або невидимими відповідно до потреб користувача. Ця функція допомагає ізолювати об'єкти для редагування, одночасно запобігаючи порушенню інших частин композиції;
- **керування непрозорістю:** для кожного об'єкта на полотні можна налаштувати непрозорість, що корисно для створення прозорих шарів або додавання ефектів, таких як тіні та світіння.

Редагування на основі шарів забезпечує неруйнівні робочі процеси, коли користувачі можуть експериментувати з різними композиціями, не змінюючи остаточно вихідний вміст.

### **Інтерактивність Canvas**

Однією з головних переваг редактора на основі полотна є його здатність забезпечувати високоінтерактивне середовище редагування. Користувачі можуть легко керувати зображеннями безпосередньо на полотні:

- **події миші та дотику:** редактор підтримує як мишу, так і дотик для інтерактивного малювання, перетягування, масштабування та обертання. Ця універсальність робить додаток придатним як для настільних, так і для мобільних користувачів;
- **контекстні меню та елементи керування на панелі інструментів:** контекстні меню або контекстні меню, а також кнопки на панелі інструментів надають набір інструментів для взаємодії з об'єктами. Ці інструменти містять параметри вирізання, копіювання, вставлення та видалення вибраних елементів, а також застосування різноманітних трансформацій і ефектів;
- **напрямні прив'язки та вирівнювання:** щоб підвищити точність, редактор може включити прив'язку на основі сітки або інтелектуальні напрямні, які допомагають вирівняти об'єкти один відносно одного або в певному місці на полотні;
- **зворотній зв'язок у режимі реального часу:** коли користувачі налаштовують елементи на полотні, вони можуть бачити результати в режимі

реального часу, забезпечуючи плавне редагування, яке сприяє творчості та експерименту.

### **Імпорт/експорт зображень і підтримка форматів**

Маніпуляції на основі Canvas також підтримують імпорт і експорт різних форматів зображень, що робить його дуже гнучким для різних робочих процесів:

1. Імпорт зображень: користувачі можуть імпортувати зображення в різних форматах (наприклад, PNG, JPEG, SVG) на полотно, де ними можна керувати, як і будь-яким іншим об'єктом. Fabric.js дозволяє неруйнівне редагування зображень, тобто імпортовані зображення можна трансформувати, маскувати та змінювати розмір без впливу на вихідний файл.

2. Експорт зображень: після редагування користувачі можуть експортувати відредаговане зображення у бажаному форматі. Редактор може містити опції для експорту всього полотна або лише окремих шарів чи об'єктів. Поширені формати експорту включають PNG, JPEG, SVG і PDF.

3. Фільтри зображень: такі фільтри, як яскравість, контраст, насиченість і тони сепії, можна застосовувати безпосередньо до імпортованих зображень на полотні, покращуючи процес редагування, не виходячи з інтерфейсу редактора.

## **3.2 Розширена система пензлів**

Розширена система пензлів є ключовою функцією веб-редактора зображень, яка пропонує користувачам багатий набір інструментів для творчого вираження та точної обробки зображень. Незалежно від того, чи є користувач цифровим художником, ілюстратором чи графічним дизайнером, здатність налаштовувати пензлі та імітувати різні техніки малювання має вирішальне значення для досягнення високоякісних професійних результатів. Ця система розроблена, щоб забезпечити універсальність, реалістичність і оперативність, задовольняючи широкий спектр художніх потреб.

## **Налаштування пензля**

Головною особливістю розширеної системи щіток є широкі можливості налаштування, які дозволяють користувачам створювати пензлі відповідно до їхніх конкретних художніх потреб. Ці варіанти включають:

1. Розмір і непрозорість пензля: користувачі можуть регулювати розмір і непрозорість пензлів у режимі реального часу, забезпечуючи гнучкість як для широких мазків, так і для детальної роботи. Це особливо корисно для створення тонких градієнтів, ефектів змішування або сміливих виразних штрихів.

2. Твердість пензля: твердість пензля визначає чіткість країв. Залежно від бажаного ефекту користувачі можуть змінювати жорсткість пензлика, щоб досягти чітких ліній або м'якших країв, що розмиваються.

## **Кілька типів пензлів**

Щоб адаптувати різні художні стилі та техніки, розширена система пензлів включає широкий набір типів пензлів, кожен з яких підходить для конкретних завдань:

- жорсткі пензлі: ці пензлі мають гострі чіткі краї, що робить їх ідеальними для точних ліній і деталей, наприклад контурів або деталізації тонких елементів зображення;

- м'які пензлі: ці пензлі мають м'які змішані краї, ідеальні для розтушовування, змішування кольорів або створення плавних градієнтів і переходів між тонами;

- пензлі-розпилювачі: для створення більш художнього ефекту, схожого на графіті, система включає пензлі-розпилювачі, які випадковим чином розкидають частинки кольору, створюючи унікальний текстурований вигляд, який можна використовувати для фонів або абстрактних малюнків.

## **Розширена підтримка шарів для пензлів**

Пензлі повністю інтегровані з системою керування шарами, що забезпечує неруйнівне редагування та більший творчий контроль:

- пензлі для окремих шарів: користувачі можуть застосовувати пензлі до окремих шарів, гарантуючи, що ефекти пензля не впливатимуть на інші частини

зображення. Це забезпечує більш контрольований і організований процес редагування, оскільки користувачі можуть працювати з різними елементами ілюстрації окремо;

- непрозорість і маскування: пензлі можна використовувати в поєднанні з непрозорістю шарів і масками шарів, щоб покращити їх застосування. Наприклад, художник міг використовувати м'який пензель, щоб намалювати маску, поступово відкриваючи або приховуючи ділянки зображення з плавним, контрольованим переходом.

Розширена система пензлів у веб-редакторі зображень пропонує дуже гнучкий, динамічний і настроюваний набір інструментів, які дають змогу користувачам створювати складні, виразні твори мистецтва та дизайни. Для детальних ілюстрацій, ретушування фотографій чи абстрактного мистецтва розширена система щіток забезпечує універсальність і точність, необхідні для втілення творчих задумів.

### 3.3 Фільтри та ефекти зображень у реальному часі

Система фільтрів і ефектів зображень у реальному часі в розширеному веб-редакторі розроблена для надання миттєвого візуального зворотного зв'язку, що дозволяє користувачам легко застосовувати широкий спектр трансформацій і покращень до своїх зображень. Ці фільтри та ефекти не тільки швидкі, але й дуже інтерактивні, що дозволяє користувачам миттєво побачити вплив своїх коригувань. Використання WebGL та інших методів оптимізації продуктивності гарантує, що користувачі отримають плавні, високоякісні результати, навіть якщо мають справу з великими зображеннями або кількома шарами.

У цьому розділі ми досліджуємо основні та розширені функції, які роблять фільтри та ефекти в реальному часі невід'ємною частиною процесу редагування зображень у програмі.

## **Неруйнівне редагування з попереднім переглядом у реальному часі**

Однією з найважливіших особливостей системи фільтрів і ефектів у реальному часі є неруйнівне редагування. Коли користувачі застосовують фільтр або ефект, зміни не записуються безпосередньо до вихідних даних зображення. Натомість вони наносяться як шар поверх зображення, що дозволяє користувачам налаштовувати, видаляти або коригувати ефект, не змінюючи назавжди базове зображення. Це забезпечує гнучкість і більш контрольований процес редагування.

1. Миттєвий попередній перегляд: коли користувачі регулюють інтенсивність або параметри фільтра чи ефекту, зміни миттєво застосовуються до зображення, пропонуючи попередній перегляд результату в реальному часі. Цей інтерактивний зворотний зв'язок дозволяє користувачам точно налаштовувати свої зміни, доки вони не досягнуть бажаного результату.

2. Зворотні ефекти: оскільки система є неруйнівною, користувачі можуть легко повернутися до вихідного зображення або налаштувати параметри фільтра в будь-який час, не турбуючись про втрату попередньої роботи. Кілька шарів фільтрів можна складати в стопку, і кожен шар можна редагувати окремо.

### **Загальні фільтри та ефекти зображень**

Система фільтрів у реальному часі надає повний набір поширених покращень зображення, починаючи від простих налаштувань і закінчуючи складними художніми ефектами. Деякі з ключових фільтрів включають:

1. Налаштування яскравості та контрастності: ці основні, але важливі фільтри дозволяють користувачам змінювати загальну освітленість і темність зображення. Налаштування яскравості збільшує або зменшує експозицію, тоді як налаштування контрастності регулюють різницю між світлими та темними ділянками зображення.

2. Насиченість і зміна відтінку: ці фільтри дозволяють користувачам змінювати яскравість кольорів зображення (насиченість) і зміщувати відтінок, дозволяючи коригувати колір, градувати колір або створювати абсолютно нові настрої, змінюючи загальну палітру кольорів.

3. Збільшення різкості та розмиття: користувачі можуть підвищити різкість зображення, щоб підкреслити деталі та виділити текстури, або застосувати ефект розмиття, щоб пом'якшити зображення чи створити ефект глибини різкості. Система може включати певні типи розмиття, наприклад розмиття за Гауссом, розмиття в русі та розмиття, кожен з яких служить різним художнім або практичним цілям.

4. Віньєтування: цей ефект затемнює або освітлює краї зображення, привертаючи увагу до центру та створюючи сфокусований драматичний візуальний ефект. Це може бути особливо корисним під час портретної зйомки або для підкреслення об'єкта в композиції.

5. Відтінки сірого та сепія: фільтри, які перетворюють зображення на чорно-біле (відтінки сірого) або додають теплий, вінтажний відтінок (сепія). Вони часто використовуються, щоб викликати позачасове або ностальгічне відчуття в зображенні.

### **3.4 Основні функції аналізу зображень за допомогою AI**

Інтеграція моделей аналізу зображень на основі штучного інтелекту в веб-редактор розширює можливості програми. Ці моделі використовують передові методи машинного навчання, такі як глибоке навчання та комп'ютерне бачення, щоб аналізувати зображення та пропонувати інформацію.

#### **Виявлення та розпізнавання об'єктів**

Моделі штучного інтелекту можуть ідентифікувати та класифікувати об'єкти на зображенні, наприклад людей, тварин, транспортні засоби або певні орієнтири.

#### **Аналіз стилю та композиції**

Моделі AI аналізують загальний стиль і композицію зображення, пропонуючи пропозиції для покращення візуальної привабливості:

- пропозиції щодо гармонії кольорів: система оцінює палітру кольорів і пропонує коригування для покращення гармонії або досягнення певних настроїв, наприклад теплих, холодних або яскравих тонів;

- правило третин і допомога при обрізанні: аналізуючи композицію, AI може запропонувати оптимальне кадрування на основі правила третин, золотого перерізу або інших рекомендацій щодо композиції, забезпечуючи візуально збалансовані результати;
- оптимізація освітлення та контрастності: система може визначати області з поганим освітленням або контрастністю та запропонувати покращення, такі як налаштування яскравості в тінях або покращення деталей у пересвітлених областях.

### **Моделі та методи III**

Реалізація аналізу на основі штучного інтелекту базується на передових моделях машинного навчання [7].

#### **Згорткові нейронні мережі (CNN)**

CNN широко використовуються в аналізі зображень для таких завдань, як виявлення об'єктів, семантична сегментація та розпізнавання стилів. Їх багаторівнева архітектура розроблена для ефективного визначення шаблонів і особливостей на зображеннях.

- фреймворки виявлення об'єктів: такі моделі, як YOLO (You Only Look Once) або Mask R-CNN, використовуються для виявлення та сегментації об'єктів у реальному часі;
- виявлення країв: CNN можуть аналізувати краї зображення, щоб точно ідентифікувати форми, об'єкти та текстури.

#### **Генеративні змагальні мережі (GAN)**

GAN використовуються для завдань, які потребують синтезу зображення та передачі стилю:

- передача стилю: GAN дозволяють застосовувати художні стилі до зображень, наприклад перетворювати фотографію на картину, що нагадує роботу відомого художника;

- відновлення зображень: GAN можна використовувати для таких завдань, як малювання (заповнення відсутніх ділянок), усунення шумів або покращення зображень із низькою роздільною здатністю.

### **Трансформери зору (ViTs)**

Vision Transformers, нова архітектура комп'ютерного зору, може обробляти складні зв'язки в зображеннях, зосереджуючись на глобальних і локальних функціях одночасно. Вони використовуються для:

- розуміння сцени: ViTs може аналізувати цілі сцени, надаючи розширені пропозиції редагування на основі контексту зображення;
- відносини кількох об'єктів: ці моделі розуміють просторові та функціональні зв'язки між кількома об'єктами, що робить їх ідеальними для багатосарових композицій.

Переваги аналізу зображень за допомогою ШІ:

- ефективність: автоматизація повторюваних завдань економить час, дозволяючи користувачам зосередитися на творчих аспектах, а не на ручних налаштуваннях;
- точність: аналіз штучного інтелекту забезпечує більш точний вибір і редагування, мінімізуючи помилки та невідповідності;
- доступність: навіть початківці користувачі можуть досягти результатів професійної якості за допомогою інтелектуальних інструментів і пропозицій;
- креативність: штучний інтелект відкриває нові творчі можливості, пропонуючи стилі, фільтри та редагування, які користувачі могли не враховувати.

Моделі аналізу зображень на базі штучного інтелекту є наріжним каменем веб-редактора зображень, що забезпечує розумніші, швидші та креативніші робочі процеси. Використовуючи передові методи виявлення об'єктів, семантичної сегментації, аналізу стилів і GAN, система пропонує потужні інструменти автоматизації та точності, які перетворюють редагування зображень на інтуїтивно зрозумілий і захоплюючий досвід. Ці моделі долають розрив між технічною складністю та доступністю для користувача, надаючи можливість як звичайним

користувачам, так і професіоналам досягати приголомшливих результатів без зусиль.

### **3.5 Спеціальне керування кольором**

Настроюване керування кольором у веб-редакторі зображень пропонує комплексну систему вибору керування кольорами, адаптовану до різноманітних потреб художників, дизайнерів і звичайних користувачів. Ця функція надає інтуїтивно зрозумілі інструменти та розширені функції для спрощення процесу роботи з кольорами, забезпечуючи послідовність і креативність у кожному проекті. У цьому розділі розглядаються основні компоненти та переваги системи, підкреслюється її роль у покращенні взаємодії з користувачем.

#### **Основні функції спеціального керування кольором**

##### **Розширений вибір кольорів**

Додаток інтегрує складний вибір кольорів, який виходить за рамки базової функціональності. Можливе налаштування відтінку, насиченості та яскравості (HSB): користувачі можуть регулювати відтінок, насиченість і яскравість за допомогою повзунків або візуального колірної колеса, забезпечуючи точний контроль вибору кольору.

##### **Збережені кольорові палітри**

Щоб підтримувати узгодженість між проектами, редактор підтримує створення, зберігання та повторне використання власних палітр кольорів. Спеціальні палітри для проекту: користувачі можуть зберігати палітри, специфічні для проекту, забезпечуючи легкий доступ до тих самих кольорів протягом усього процесу проектування.

#### **Доступність і зручність використання**

##### **Візуальне налаштування**

Інтерфейс адаптується до вподобань користувача щодо взаємодії кольорів:

- Оптимізація темного/світлого режимів: інструменти керування кольором оптимізовано як для темних, так і для світлих тем, забезпечуючи стабільну роботу;
- Регульовані елементи інтерфейсу користувача: повзунки, палітри та колеса можна змінювати або реорганізовувати відповідно до робочого процесу користувача.

### **Переваги спеціального керування кольором**

- творчість: дозволяє користувачам досліджувати та впроваджувати унікальні колірні схеми з мінімальними зусиллями;
- ефективність: спрощує робочі процеси, пропонуючи швидкий доступ до збережених кольорів;
- точність: забезпечує точне збігання кольорів між інструментами, шарами;
- доступність: робить розширені функції кольору інтуїтивно зрозумілими для початківців, а професіоналам пропонує детальні елементи керування.

Спеціальна система керування кольором у веб-редакторі є наріжним каменем функціональності програми, яка поєднує в собі інтуїтивно зрозумілі інструменти, розширені функції та доступність для розширення можливостей користувачів будь-якого рівня кваліфікації. Незалежно від того, створюєте складне цифрове мистецтво, редагуєте фотографії чи розробляєте брендингові ресурси, система забезпечує точність і гнучкість, необхідні для втілення творчих бачень у життя

### **3.6 Підтримка імпорту/експорту та форматування файлів**

Ефективна функція імпорту/експорту файлів і підтримка широкого формату є ключовими компонентами веб-редактора зображень, що забезпечує безперебійну взаємодію з різноманітним програмним забезпеченням і платформами. Ці функції дають змогу користувачам працювати з різними типами файлів, керувати великими проектами та передавати свої творіння між пристроями та програмами з

мінімальними труднощами. Нижче наведено детальне дослідження того, як обробка файлів і підтримка форматів реалізовані в проекті.

### **Можливості імпорту файлів**

Редактор підтримує широкий спектр форматів файлів, задовольняючи різні потреби користувачів:

- растрові формати: для імпорту стандартних зображень підтримуються такі широко використовувані формати, як PNG, JPEG.

#### **Попередня обробка зображення**

Перш ніж зображення буде повністю завантажено в редактор, етапи попередньої обробки оптимізують їх для редагування:

- налаштування роздільної здатності: зображення можна змінювати або зменшувати автоматично відповідно до розміру полотна або вподобань користувача.

#### **Імпорт кількох файлів**

Користувачі можуть імпортувати кілька файлів одночасно, які організовані в шари або як окремі полотна для багатопроєктних робочих процесів.

### **Можливості експорту файлів**

Користувачі можуть експортувати свої проекти у стандартному форматі для публікації в Інтернеті або поширення в соціальних мережах.

#### **Архітектура обробки файлів**

Редактор використовує ефективні бібліотеки та API та перетворення форматів файлів:

- растрові бібліотеки: такі бібліотеки, як API Sharp або Canvas, точно обробляють формати растрових зображень;
- векторні бібліотеки: синтаксичний аналіз і візуалізація SVG підтримуються за допомогою таких бібліотек, як SVG.js або власних API браузера;
- декодери необроблених файлів: спеціалізовані декодери забезпечують сумісність із форматами RAW, вилучаючи дані найвищої якості.

#### **Майбутні вдосконалення**

- користувацькі налаштування експорту: користувачі зможуть зберігати бажані налаштування експорту для швидкого доступу;

- рекомендації щодо формату за допомогою штучного інтелекту: штучний інтелект запропонує найкращі формати та налаштування на основі сценарію використання проекту;
- розширена підтримка RAW: глибша інтеграція з виробниками камер для підтримки нових форматів RAW;
- формати 3D-файлів: підтримка таких форматів, як OBJ або GLTF, для розширення робочих процесів 3D-проекування.

Надійна система імпорту/експорту файлів у веб-редакторі забезпечує сумісність, ефективність і гнучкість для різноманітних потреб користувачів. Завдяки підтримці широкого діапазону редактор забезпечує безперешкодний досвід імпорту, редагування та експорту файлів. Майбутні вдосконалення ще більше розширять його можливості, зробивши його універсальним інструментом як для професіоналів, так і для любителів.

## РОЗДІЛ 4

### ОГЛЯД ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ЩОДО ПРОЄКТУВАННЯ ТА РОЗРОБКИ ПРОГРАМНИХ ПРОДУКТІВ

Розробка веб-редактора зображень із розширенням штучним інтелектом потребує сучасного, ефективного підходу до проектування та розробки програмного забезпечення. Цей проект використовує ряд передових інструментів, фреймворків і бібліотек для оптимізації створення, тестування та розгортання програми. Нижче наведено огляд використовуваних інструментів і технологій, класифікованих за їх ролями в життєвому циклі розробки програмного забезпечення.

#### 4.1 Засоби розробки інтерфейсу

##### **React + TypeScript**

- **React:** декларативна бібліотека JavaScript на основі компонентів, яка спрощує створення динамічних інтерфейсів користувача. Віртуальний DOM React забезпечує ефективне відтворення, що є критичним для редагування зображень у реальному часі [8];
- **TypeScript:** покращує JavaScript, додаючи статичну типізацію, що покращує підтримку коду, зменшує кількість помилок і забезпечує кращу підтримку інструментів [9].

##### **Fabric.js**

Потужна бібліотека canvas для роботи з зображеннями в інтернеті та їх відтворення [10]. Fabric.js забезпечує вбудовану підтримку для:

- **маніпуляції об'єктами:** обробляє такі перетворення, як масштабування, обертання та нахил;
- **інструменти малювання:** сприяють розширеним системам пензлів і спеціальним ефектам;
- **можливості експорту:** дозволяє експортувати вміст полотна в декілька форматів.

## **Tailwind CSS**

CSS-фреймворк, перш за все утиліти, який прискорює процес проектування, надаючи попередньо визначені класи. Tailwind забезпечує послідовний і адаптивний стиль, одночасно зменшуючи роздутість CSS [11].

## **Radix UI**

Бібліотека доступних і нестилізованих компонентів інтерфейсу користувача. Radix UI використовується для:

- створення повторно використовуваних компонентів, таких як модалі, повзунки та спадні меню;
- забезпечення доступності для різних пристроїв і інтерфейсів користувача.

## **4.2 Інструменти інтеграції ШІ**

### **Ollama з моделлю Llava**

- Ollama: платформа для інтеграції великих мовних моделей у програми. Це полегшує:
  - аналіз зображення: надає інтелектуальну інформацію, як-от виявлення об'єктів і інтерпретація сцени;
  - обробка даних: підтримує взаємодію на основі природної мови для створення пропозицій дизайну;
  - модель Llava: спеціалізована модель штучного інтелекту для мультимодальних завдань, що дозволяє редактору інтерпретувати та обробляти зображення разом із текстовим введенням.

### **TensorFlow.js і ONNX Runtime**

Незважаючи на те, що це явно не зазначено, такі інструменти, як TensorFlow.js або ONNX Runtime, можна інтегрувати для оптимізації моделей штучного інтелекту для виконання на основі браузера, підвищення продуктивності та зменшення затримки.

### 4.3 Інструменти розробки та проектування

#### **Figma**

Інструмент для спільного проектування, який використовується для створення прототипів і дизайну UI/UX. Інтерактивні макети: Візуалізація інтерфейсу програми перед впровадженням.

#### **Code Visual Studio (code VS)**

Легкий та налаштований редактор коду [12], який підтримує:

- інтеграція TypeScript і React: пропонує IntelliSense і навігацію кодом;
- розширення: інструменти для лінігування, налагодження та тестування.

#### **GitHub**

Платформа контролю версій, яка спрощує спільну розробку за допомогою:

- хостинг і керування вихідним кодом проекту;
- інтеграція конвеєрів CI/CD для автоматизованого тестування та розгортання.

### 4.4 Інструменти тестування та налагодження

#### **Бібліотека тестування Jest і React**

- Jest: платформа тестування JavaScript для модульного та інтеграційного тестування;
- бібліотека тестування React: полегшує тестування компонентів React, імітуючи взаємодію користувача та гарантуючи, що інтерфейс користувача поводить належним чином.

#### **Інструменти розробника браузера**

Вбудовані інструменти браузера для усунення вузьких місць продуктивності, перевірки мережевих запитів і оптимізації відтворення.

#### **Lighthouse**

Інструмент із відкритим вихідним кодом для аналізу продуктивності веб-додатків, доступності та SEO. Це допомагає:

- Визначте слабкі місця продуктивності.
- Забезпечте дотримання найкращих практик.

## 4.5 Інструменти оптимізації продуктивності

### Webpack і Vite

- Webpack: збірник модулів, який оптимізує завантаження активів для кращої продуктивності [13];
- Vite: інструмент для збирання нового покоління, який забезпечує блискавичну HMR (гарячу заміну модулів) і мінімальний час збирання [14].

### Chrome Profiler

Інструмент профілювання для вимірювання виконання, відтворення та використання пам'яті JavaScript у реальному часі.

## 4.6 Інструменти розгортання та масштабованості

### Docker

Використовується для контейнеризації програми, забезпечення узгодженого середовища на етапах розробки, тестування та виробництва.

### AWS і Firebase

- хмарні платформи для розміщення додатків, зберігання даних користувачів і забезпечення функцій у реальному часі;
- AWS S3 або Firebase Storage: використовується для ефективного керування великими файлами зображень і даними проекту.

Інструменти, використані при розробці веб-редактора зображень із розширенням штучним інтелектом, відображають сучасний підхід до створення програмного забезпечення. Завдяки використанню найсучасніших зовнішніх фреймворків, платформ інтеграції ШІ, інструментів для спільної роботи та бібліотек оптимізації продуктивності, цей проект досягає балансу між функціональністю, зручністю використання та масштабованістю. Ці інструменти не тільки підвищують

ефективність розробки, але й гарантують, що кінцевий продукт відповідає потребам різноманітних користувачів і випадків використання.

## РОЗДІЛ 5

### ОПИС РОЗРОБЛЕНОГО В РАМКАХ МАГІСТЕРСЬКОЇ ДИПЛОМНОЇ РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 5.1 Мета і завдання роботи

Основною метою магістерської роботи є розробка та реалізація веб-редактора зображень із розширенням штучним інтелектом. Це програмне забезпечення має на меті забезпечити надійну платформу для редагування зображень із можливістю аналізу зображень штучним інтелектом, створену як для професійних, так і для звичайних користувачів.

Цілі:

- впровадити інтуїтивно зрозумілий інтерфейс користувача: забезпечити зручність використання для користувачів із різним рівнем досвіду завдяки доступному дизайну та функціональності;
- включити розширені інструменти обробки зображень: розробити такі функції, як спеціальні пензлі, керування шарами та ефекти в реальному часі;
- використовувати штучний інтелект для розширеної функціональності: інтегрувати моделі штучного інтелекту для аналізу зображень, виявлення об'єктів і інтелектуальних пропозицій редагування;
- забезпечити сумісність між платформами: забезпечити адаптивне рішення на основі веб-переглядача, що усуває потребу в установці чи спеціальному обладнанні;
- оптимізувати продуктивність: реалізувати ефективну візуалізацію, керування станом і обробку даних, щоб забезпечити зручну роботу користувача;
- підтримка сумісності файлів: полегшення імпорту/експорту в різних форматах для професійних робочих процесів.

## 5.2 Вимоги до програмного забезпечення

### Функціональні вимоги

Функції редагування зображень:

- удосконалена система пензлів із кількома типами пензлів і параметрами, що настроюються;
- керування шарами, включаючи зміну порядку перетягування та параметри змішування шарів;
- фільтри та ефекти в реальному часі, такі як корекція кольору, розмиття та художні стилі;
- варіанти темної та світлої теми для комфортного використання.

Можливості ШІ:

- аналіз зображень за допомогою ШІ для розпізнавання об'єктів;
- розуміння сцени.

Обробка файлів:

- підтримка кількох форматів файлів імпорту (наприклад, PNG, JPEG, SVG, RAW);
- імпорт/експорт результату редагування.

Підтримка між платформами:

- програма на основі браузера, сумісна з настільними та мобільними пристроями;
- чуйний дизайн для бездоганної роботи на екранах різних розмірів.

## 5.3 Опис результатів розробки програмного забезпечення

### Основні функції

Програмне забезпечення надає повний набір інструментів для редагування зображень та аналізу їх за допомогою ШІ. Основні реалізовані функції включають:

- спеціальна система пензлів: користувачі можуть створювати складні дизайни з різними типами пензлів, регульованим розміром, непрозорістю та динамічними ефектами;
- керування шарами: зміна порядку перетягування шарів, режими змішування та групові функції для професійних робочих процесів;
- фільтри в реальному часі: користувачі можуть динамічно застосовувати фільтри зображень, миттєво переглядаючи зміни без затримок;
- інструменти на основі штучного інтелекту: аналіз результату редагування штучним інтелектом для швидкого створення опису зображення.

### **Інтерфейс користувача**

Програма має чистий, сучасний інтерфейс користувача, створений за допомогою Tailwind CSS і Radix UI, що забезпечує:

- швидкість реагування: оптимізовані макети для настільних комп'ютерів і мобільних пристроїв;
- налаштовані теми: світлі та темні теми, створені для зручності користувача.

### **Інтеграція ШІ**

Модель Ollama «Llava» виконує детальний аналіз зображення.

## РОЗДІЛ 6

### ІНТЕРФЕЙС КОРИСТУВАЧА

На головній сторінці сайту (рисунок 6.1) користувач має можливість імпортувати зображення для редагування на полотні та аналізу зображення за допомогою ШІ. Програмне забезпечення при першому відкритті має порожнє полотно, увімкнений темний режим сторінки за замовчанням та розміри полотна 800 на 600 пікселів.

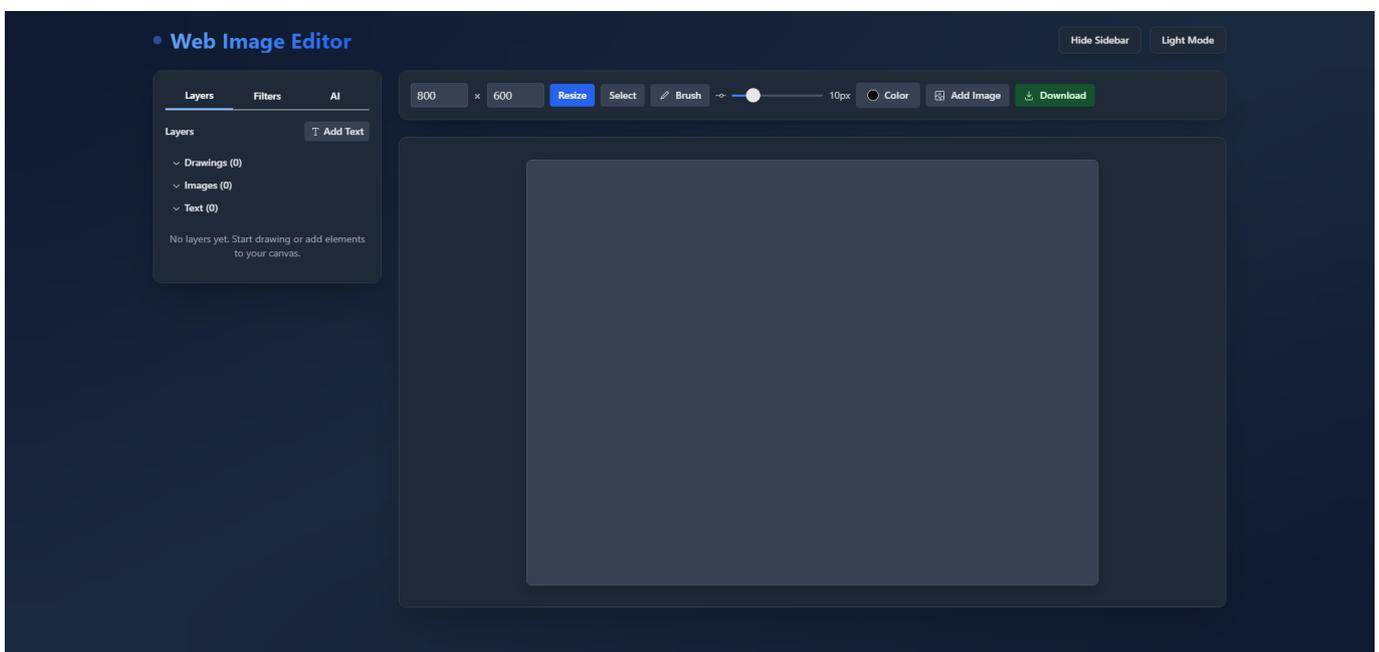


Рисунок 6.1 – Головна сторінка сайту

На цій сторінці користувач може виконати: зміну розміру зображення (рисунок 6.2), зміну позиції шарів на полотні через перетягування їх (рисунок 6.3), додавати текст до зображення (рисунок 6.4), обирати пензлі та малювати ними (рисунок 6.5), завантажувати результат редагування у .PNG форматі, сховати бокове меню (рисунок 6.6), перейти у «Світлий режим» (рисунок 6.7).

Опис кнопок головної сторінки:

- Add image – Дозволяє додати зображення на полотно для редагування;
- Resize – Виконує зміну розміру робочого полотна за вказаними розмірами у пікселях;

- Brush – Відкриває список доступних пензлів для малювання у робочій зоні;
- Select – Повертає курсор користувача у режим взаємодії, якщо до цього він мав режим пензля;
- Color – Відкриває меню вибору кольору для пензлів та тексту з настроюваною палітрою;
- Add Text – Дозволяє додати текст на робочу зону;
- Download – Завантажує кінчений результат редагування з робочої зони користувача;
- Hide Sidebar – Прибирає бокову панель з вкладками з інтерфейсу;
- Light Mode/Dark Mode – Перемикає «Світлий» або «Темний» режим сайту.

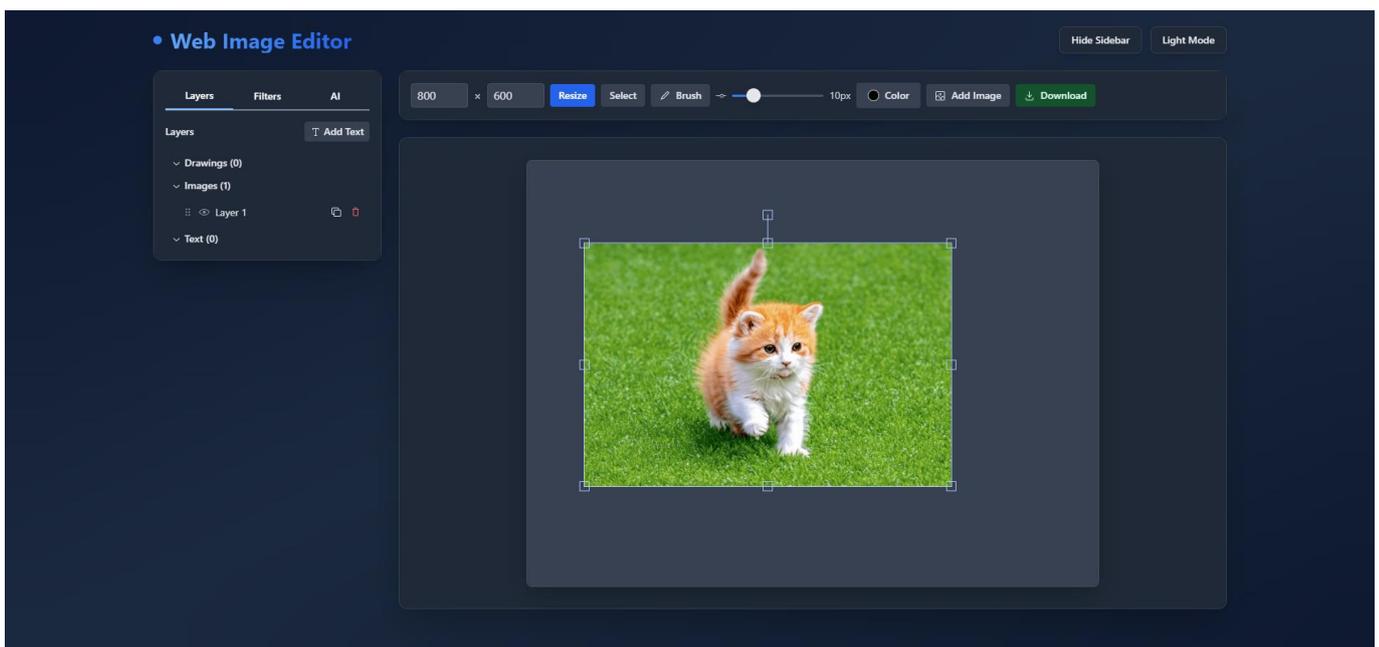


Рисунок 6.2 – Редагування розміру зображення

Приклад коду реалізації функціоналу редагування розміру зображення на рисунку 6.2 з файлу «Toolbar.tsx»:

```
// Inside Toolbar component
const [canvasWidth, setCanvasWidth] = useState(800);
const [canvasHeight, setCanvasHeight] = useState(600);

// Handle input changes
```

```

const handleSizeChange = (dimension: 'width' | 'height', value: string) => {
  const numValue = parseInt(value) || 0;
  if (dimension === 'width') {
    setCanvasWidth(numValue);
  } else {
    setCanvasHeight(numValue);
  }
};

// Apply new size to canvas
const applyCanvasSize = () => {
  if (!canvas) return;
  canvas.setDimensions({ width: canvasWidth, height: canvasHeight });
  canvas.renderAll();
};

// JSX for the resize controls
<div className="flex flex-wrap items-center gap-2 w-full sm:w-auto">
  <input
    type="number"
    value={canvasWidth}
    onChange={(e) => handleSizeChange('width', e.target.value)}
    className={`w-20 px-2 py-1 border rounded ${
      darkMode
        ? 'bg-gray-700 border-gray-600 text-gray-200'
        : 'bg-white border-gray-300 text-gray-700'
    }`}
    placeholder="Width"
  />
  <span className={darkMode ? 'text-gray-300' : 'text-gray-700'}></span>
  <input
    type="number"
    value={canvasHeight}
    onChange={(e) => handleSizeChange('height', e.target.value)}
    className={`w-20 px-2 py-1 border rounded ${
      darkMode
        ? 'bg-gray-700 border-gray-600 text-gray-200'
        : 'bg-white border-gray-300 text-gray-700'
    }`}
    placeholder="Height"
  />
  <button
    onClick={applyCanvasSize}
    className={`px-3 py-1.5 text-sm font-medium rounded ${
      darkMode
        ? 'bg-blue-600 text-white hover:bg-blue-700'
        : 'bg-blue-500 text-white hover:bg-blue-600'
    }`}
  >
    Resize
  </button>
</div>

```

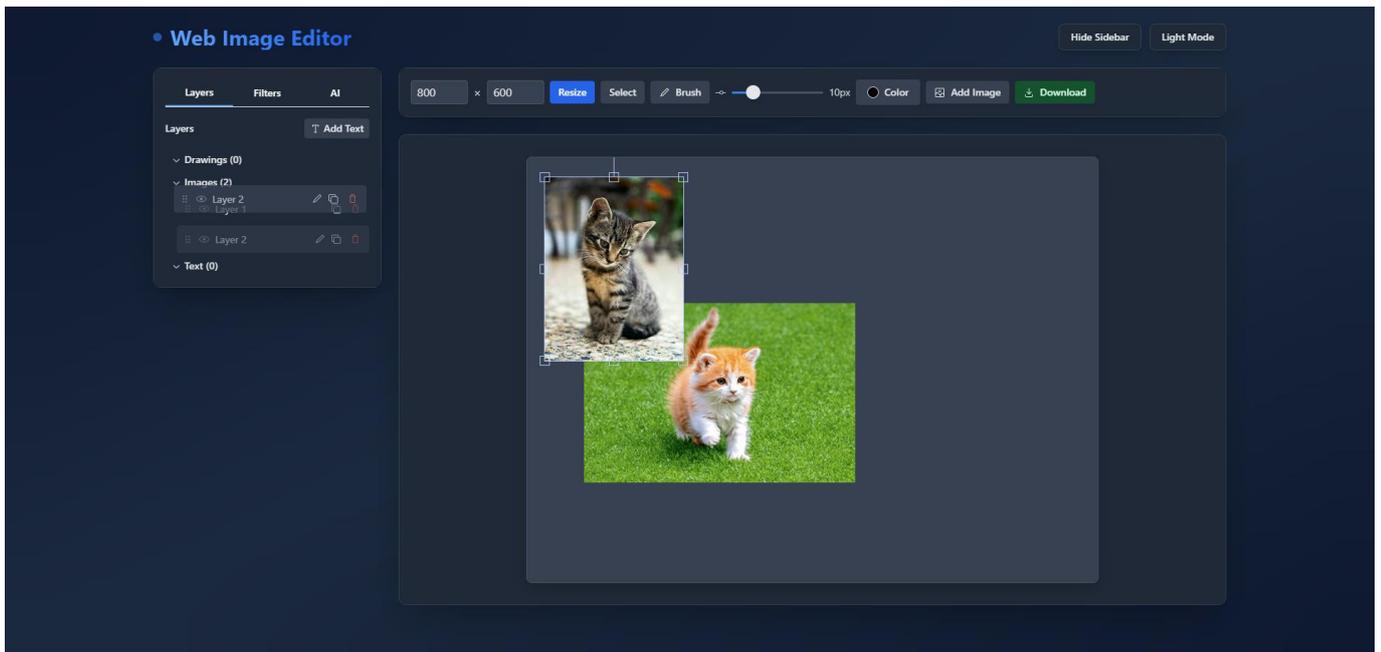


Рисунок 6.3 – Зміна позиції шарів через перетягування

Приклад коду реалізації функціоналу зміни позицій шарів через перетягування на рисунку 6.3 з файлу «LayersList.tsx»:

```
// Layer drag and drop handlers
const handleDragStart = (e: React.DragEvent, groupIndex: number, layerIndex: number) => {
  setDraggedItem({ groupIndex, layerIndex });
  e.dataTransfer.effectAllowed = 'move';
  e.currentTarget.classList.add('opacity-50');
};

const handleDragEnd = (e: React.DragEvent) => {
  e.currentTarget.classList.remove('opacity-50');
  setDraggedItem(null);
};

const handleDragOver = (e: React.DragEvent) => {
  e.preventDefault();
  e.dataTransfer.dropEffect = 'move';
};

const handleDrop = (e: React.DragEvent, targetGroupIndex: number, targetLayerIndex: number)
=> {
  e.preventDefault();

  if (!draggedItem || !canvas) return;

  const { groupIndex: sourceGroupIndex, layerIndex: sourceLayerIndex } = draggedItem;

  if (sourceGroupIndex === targetGroupIndex && sourceLayerIndex === targetLayerIndex) {
    return;
  }

  const sourceLayer = layerGroups[sourceGroupIndex].layers[sourceLayerIndex];
  const targetLayer = layerGroups[targetGroupIndex].layers[targetLayerIndex];
```

```

const sourceIndex = canvas.getObjects().indexOf(sourceLayer.object);
const targetIndex = canvas.getObjects().indexOf(targetLayer.object);

canvas.moveTo(sourceLayer.object, targetIndex);
canvas.renderAll();

updateLayerGroups();
};

// Usage in JSX
<div
  draggable={!editingLayer}
  onDragStart={(e) => handleDragStart(e, groupIndex, layerIndex)}
  onDragEnd={handleDragEnd}
  onDragOver={handleDragOver}
  onDrop={(e) => handleDrop(e, groupIndex, layerIndex)}
  className="layer-item"
>
  {/* Layer content */}
</div>

```

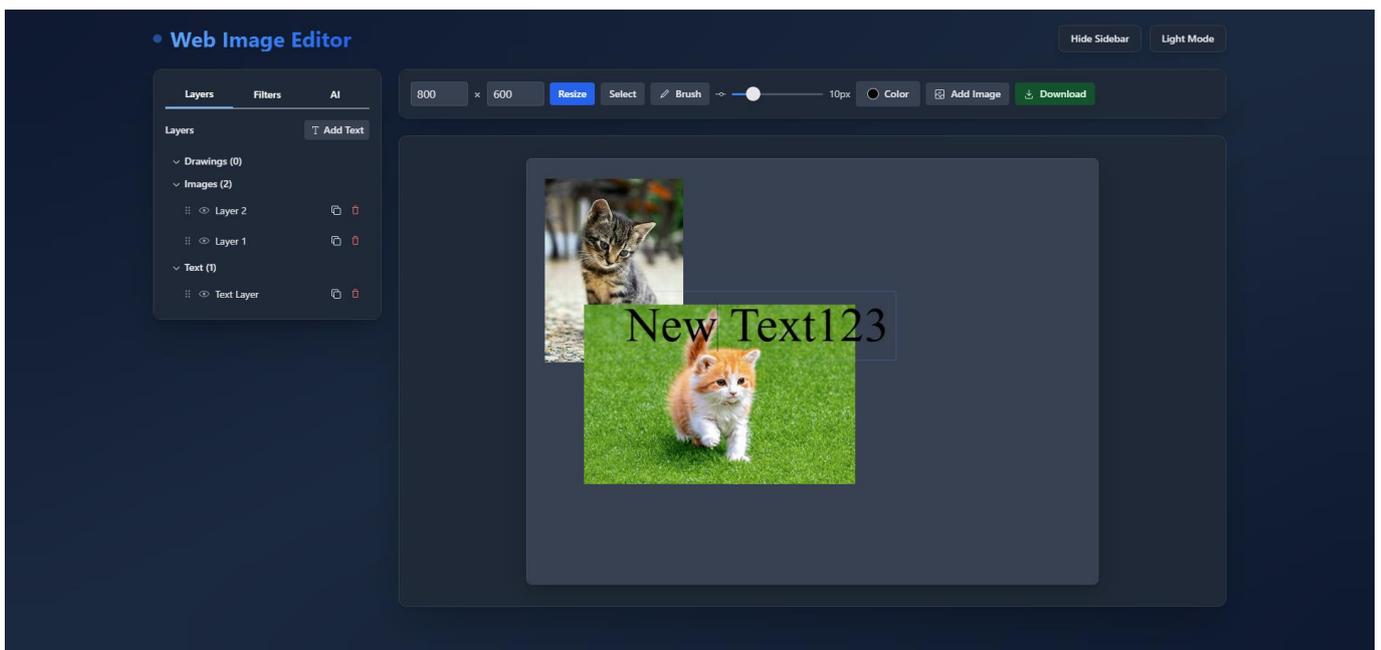


Рисунок 6.4 – Додавання тексту до зображення

Приклад коду реалізації функціоналу додавання тексту до зображення на рисунку 6.4 з файлу «LayersList.tsx»:

```

const handleAddText = () => {
  if (!canvas) return;

  const text = new fabric.IText('New Text', {
    left: canvas.width ? canvas.width / 2 : 400,
    top: canvas.height ? canvas.height / 2 : 300,
    originX: 'center',
    originY: 'center',
  });

```

```

    fontSize: 20,
    fill: selectedColor,
    name: 'Text Layer',
    editable: true,
    cursorWidth: 2,
    cursorColor: selectedColor,
    cursorDuration: 600,
    selectionColor: 'rgba(17, 119, 255, 0.3)',
    padding: 8
  });

  canvas.add(text);
  canvas.setActiveObject(text);
  canvas.renderAll();
  text.enterEditing();
  text.selectAll();
};

// Usage in JSX
<button onClick={handleAddText}>
  <TextIcon />
  Add Text
</button>

```

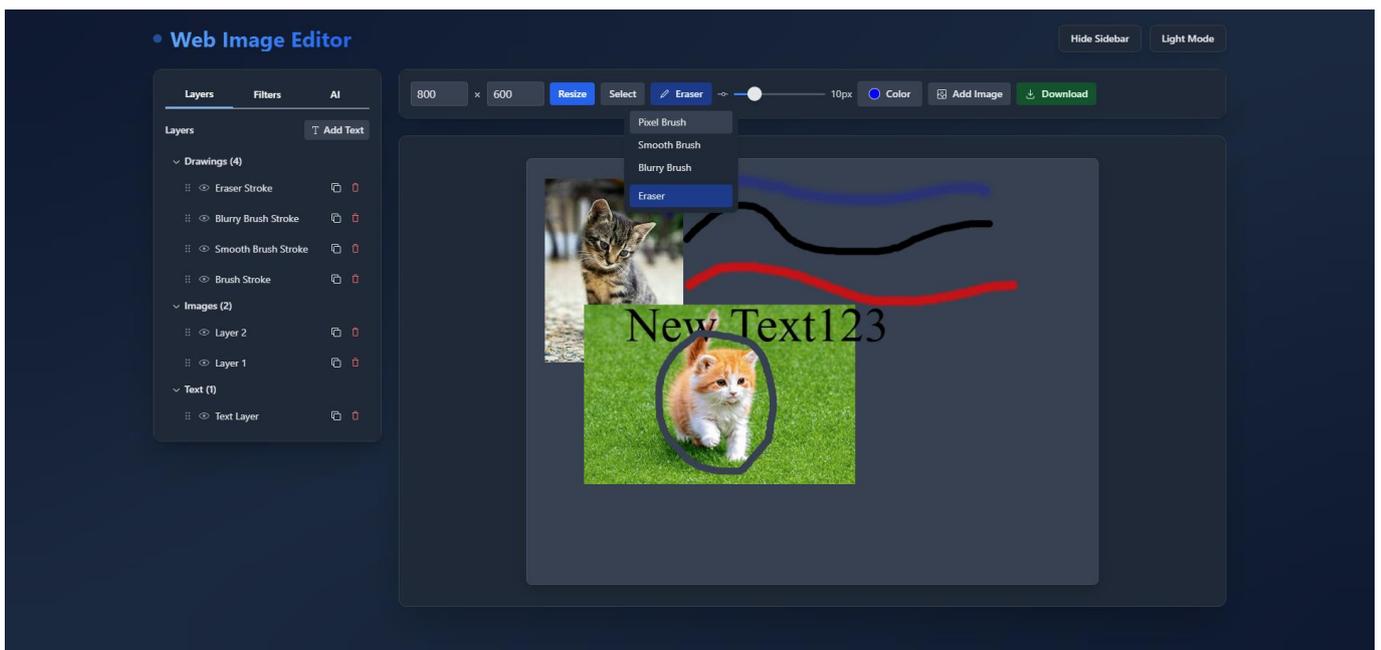


Рисунок 6.5 – Вибір пензлів та малювання ними з різними кольорами

Приклад коду реалізації функціоналу додавання тексту до зображення на рисунку 6.5 з файлу «Toolbar.tsx»:

```

const setBrushTool = (type: string) => {
  if (!canvas) return;

  canvas.isDrawingMode = false;

  switch (type) {
    case 'pixel':

```

```

    canvas.freeDrawingBrush = new PixelBrush(canvas);
    canvas.isDrawingMode = true;
    break;
case 'smooth':
    canvas.freeDrawingBrush = new SmoothBrush(canvas);
    canvas.isDrawingMode = true;
    break;
case 'blurry':
    canvas.freeDrawingBrush = new BlurryBrush(canvas);
    canvas.isDrawingMode = true;
    break;
case 'eraser':
    canvas.freeDrawingBrush = new EraserBrush(canvas);
    canvas.isDrawingMode = true;
    break;
}

if (canvas.freeDrawingBrush) {
    canvas.freeDrawingBrush.width = brushSize;
    canvas.freeDrawingBrush.color = type === 'eraser' ? '#ffffff' : selectedColor;
}
};

const handleColorChange = (color: string) => {
    onColorChange(color);
    if (canvas?.freeDrawingBrush && activeTool !== 'eraser') {
        canvas.freeDrawingBrush.color = color;
    }
};

```

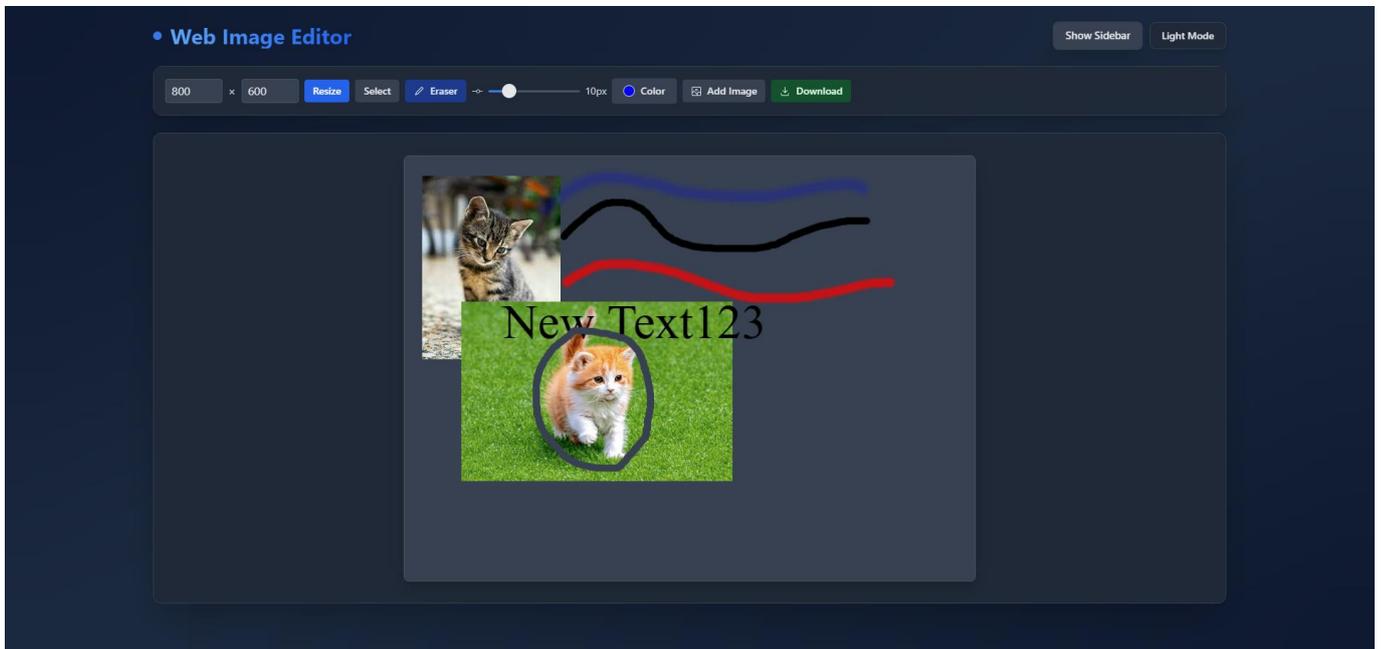


Рисунок 6.6 – Головна сторінка зі схованим боковим меню



Рисунок 6.7 – Головна сторінка зі «Світлим» режимом

Для налаштування фільтрів на зображенні користувач має натиснути на вкладку «Filters» та обрати потрібне йому зображення у робочій зоні (рисунок 6.8).

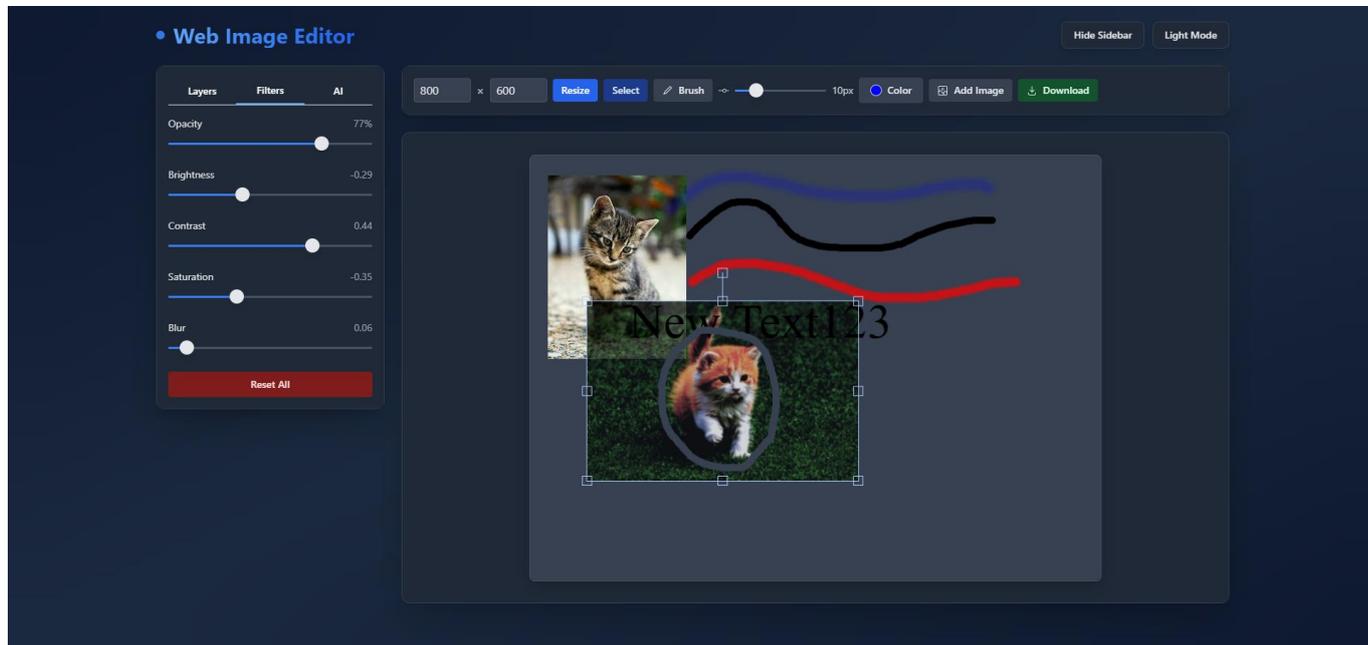


Рисунок 6.8 – Налаштування фільтрів на зображенні

Приклад коду реалізації налаштування фільтрів до зображення на рисунку 6.8 з файлу «FilterPanel.tsx»:

```
useEffect(() => {
```

```

if (!canvas) return;

const handleSelection = () => {
  const activeObject = canvas.getActiveObject();
  setSelectedObject(activeObject || null);

  if (activeObject) {
    setFilterValues({
      brightness: 0,
      contrast: 0,
      saturation: 0,
      blur: 0,
      opacity: activeObject.opacity ?? 1
    });
  } else {
    setFilterValues({
      brightness: 0,
      contrast: 0,
      saturation: 0,
      blur: 0,
      opacity: 1
    });
  }
};

canvas.on('selection:created', handleSelection);
canvas.on('selection:updated', handleSelection);
canvas.on('selection:cleared', handleSelection);

return () => {
  canvas.off('selection:created', handleSelection);
  canvas.off('selection:updated', handleSelection);
  canvas.off('selection:cleared', handleSelection);
};
}, [canvas]);

const applyFilters = (newValues: Partial<FilterValues>) => {
  if (!canvas || !selectedObject) return;

  const updatedValues = { ...filterValues, ...newValues };
  setFilterValues(updatedValues);

  // Apply opacity separately as it's not a filter
  selectedObject.set('opacity', updatedValues.opacity);

  const filters: fabric.IBaseFilter[] = [];

  if (updatedValues.brightness !== 0) {
    filters.push(new fabric.Image.filters.Brightness({
      brightness: updatedValues.brightness
    }));
  }

  if (updatedValues.contrast !== 0) {
    filters.push(new fabric.Image.filters.Contrast({
      contrast: updatedValues.contrast
    }));
  }

  if (updatedValues.saturation !== 0) {
    filters.push(new fabric.Image.filters.Saturation({

```

```

        saturation: updatedValues.saturation
    }));
}

if (updatedValues.blur !== 0) {
    filters.push(new fabric.Image.filters.Blur({
        blur: updatedValues.blur
    }));
}

if ('filters' in selectedObject) {
    selectedObject.filters = filters;
    (selectedObject as fabric.Image).applyFilters();
}

canvas.renderAll();
};

```

Перелік кнопок на вкладці фільтрів:

- Reset All – Повертає параметри фільтрів за замовчанням.

Для використання функції аналізу зображення за допомогою ШІ з підключенням до локального серверу «Ollama» та вибором моделі ШІ користувачу необхідно перейти на вкладку «AI» та натиснути кнопку «Connect» (рисунок 6.9).

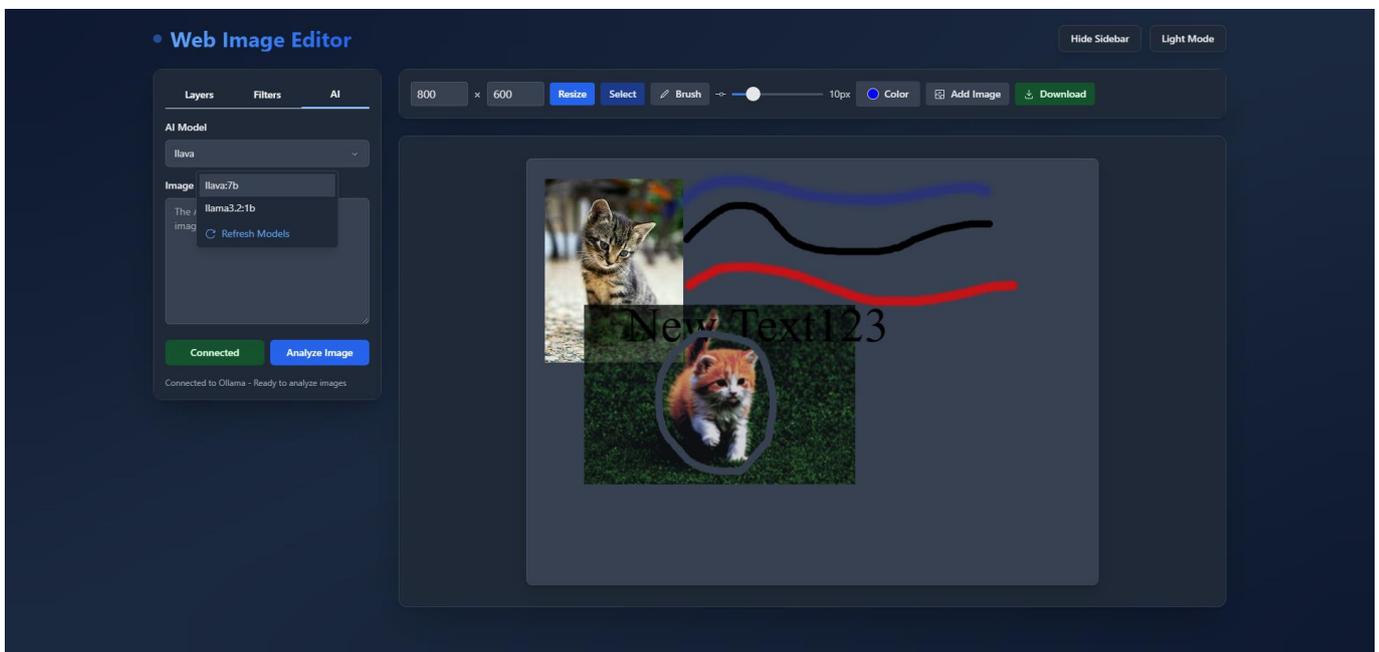


Рисунок 6.9 – Вибір моделі для аналізу зображення після під'єднання до локального серверу «Ollama»

Після натискання кнопки «Analyze Image» ШІ автоматично просканує всю робочу зону полотна та згенерує текст опису зображення у текстовому полі «Image Description» (рисунок 6.10).

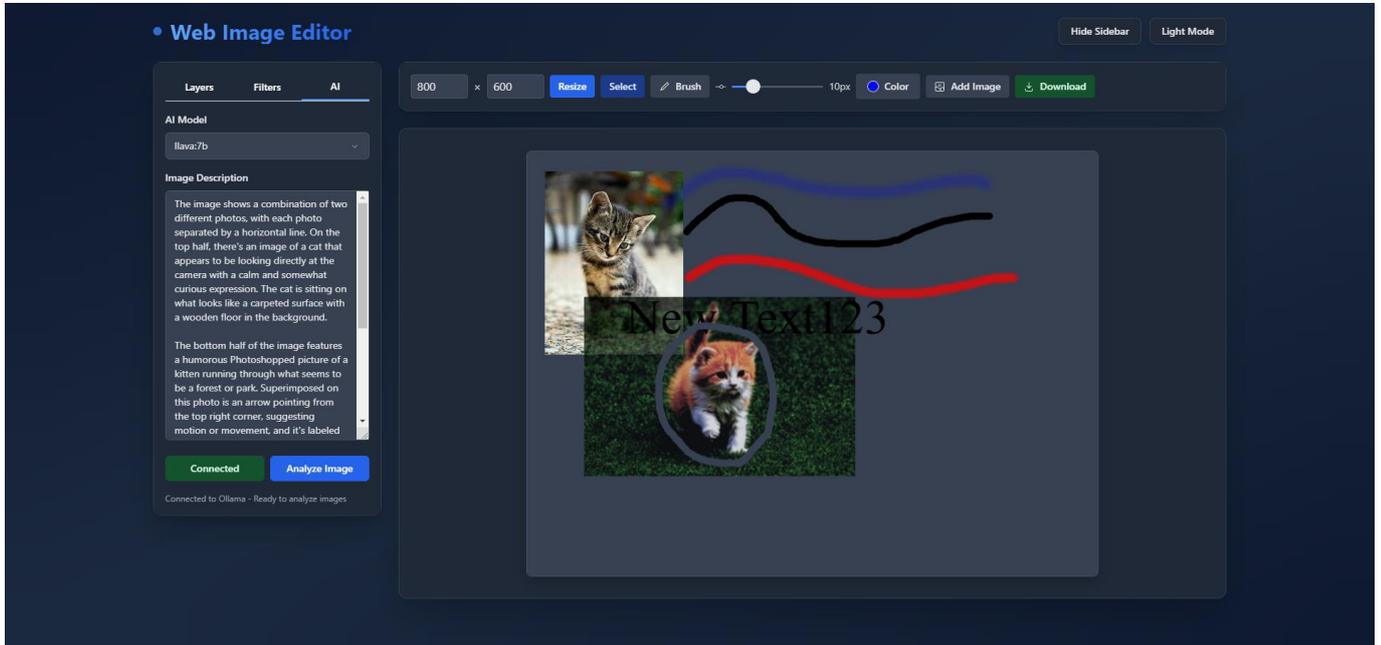


Рисунок 6.10 – Результат генерації опису зображення на робочому просторі після натискання кнопки «Analyze Image»

Перелік кнопок на вкладці ШІ:

- Connect – Автоматично під’єднує веб-застосунок до локального серверу «Ollama», що дозволяє обрати встановлену на пристрої ШІ модель для подальшого аналізу зображень на робочій зоні;
- Analyze Image – Починає процес аналізу робочої зони з подальшим виводом тексту опису проаналізованих об’єктів на полотні у текстове поле «Image Description».

## ВИСНОВКИ

Розробка веб-редактора зображень з розширенням у вигляді аналізу результату редагування за допомогою ШІ є значним досягненням у створенні сучасного, ефективного цифрового інструменту, орієнтованого на користувача. Завдяки інтеграції розширених можливостей редагування зображень та функцій штучного інтелекту цей проект відповідає зростаючому попиту на доступні професійні інструменти дизайну, які задовольняють різноманітні потреби користувачів. Реалізація таких функцій, як спеціальні пензлі, керування шарами та фільтри в реальному часі, демонструє ефективність використання передових технологій, таких як Fabric.js, React і TypeScript, для створення адаптивного та надійного середовища редагування. Крім того, інтеграція моделі штучного інтелекту, такої як Ollama's Llava, покращує функціональність програми, надаючи інтелектуальні функції аналізу та опису результату редагування зображення, що значно підвищує ефективність роботи користувача.

Кросплатформна доступність програми та підтримка різноманітних форматів файлів ще більше підвищують її практичність і зручність, роблячи її універсальним рішенням як для професіоналів, так і для звичайних користувачів. Ця дипломна робота не лише досягає поставлених цілей, але й робить внесок у ширшу область інструментів дизайну, закладаючи основу для майбутніх розробок, таких як розширені можливості штучного інтелекту, швидкість реагування на мобільні пристрої та системи плагінів для розширення функціональності програмного забезпечення. Цей проект є цінним прикладом того, як сучасні технології та продуманий дизайн можуть об'єднатися для вирішення реальних проблем у створенні цифрового контенту.

Усі поставлені задачі було виконано, а ціль роботи досягнута.

## СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Adobe Photoshop URL: <https://www.adobe.com/products/photoshop/online.html> (дата звернення: 26.12.2024).
2. Canva. URL: <https://www.canva.com/en/> (дата звернення: 26.12.2024).
3. Figma. URL: <https://www.figma.com/> (дата звернення: 26.12.2024).
4. Photopea. URL: <https://www.photopea.com/> (дата звернення: 26.12.2024).
5. Krita (web beta). URL: <https://krita.org/en/> (дата звернення: 26.12.2024).
6. Jones, A., & Smith, R. (2022). "Performance Optimization in Browser-Based Canvas Applications." Proceedings of the ACM Web Conference. URL: <https://dl.acm.org/doi/10.1145/example-link> (дата звернення: 27.12.2024).
7. Ollama Documentation for AI Integration. URL: <https://ollama.ai> (дата звернення: 28.12.2024).
8. React Official Documentation. URL: <https://reactjs.org/docs> (дата звернення: 28.12.2024).
9. Goodman, D. (2021). Modern Web Development with React and TypeScript. URL: <https://example.com/book-link> (дата звернення: 29.12.2024).
10. Fabric.js Documentation. URL: <https://fabricjs.com/docs> (дата звернення: 30.12.2024).
11. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 30.12.2024).
12. Visual Studio Code. URL: <https://code.visualstudio.com/docs> (дата звернення: 30.12.2024).
13. Webpack. URL: <https://webpack.js.org> (дата звернення: 03.12.2025).
14. Vite. URL: <https://vitejs.dev> (дата звернення: 03.01.2025).
15. LogRocket Blog: Performance Monitoring and Optimization Techniques. URL: <https://blog.logrocket.com> (дата звернення: 04.01.2025).