**Форма № Н-9.02**

Національний університет «Пролтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

\_Навчально науковий інститут інформаційних технологій та робототехніки\_

(повна назва факультету)

\_Кафедра комп’ютерних та інформаційних технологій і систем\_

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проекту (роботи)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_бакалавра\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(освітньо-кваліфікаційний рівень)

на тему

Створення онлайн інструменту для графічної обробки фото

Виконав: студент 4 курсу, групи 401-ТН

спеціальність

122 Комп’ютерні науки

(код і назва спеціальності)

Матвійчук А.Є.

(прізвище та ініціали)

Керівник Демиденко М.І.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ПОЛТАВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ЮРІЯ КОНДРАТЮКА**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

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

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

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

**на тему**

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

**Студента групи 401-ТН Матвійчука Арсенія Євгеновича**

Керівник роботи

старший викладач

Демиденко М.І.

Завідувач кафедри

доктор технічних наук,

доцент Головко Г.В.

Полтава – 2021

**РЕФЕРАТ**

Кваліфікаційна робота бакалавра: 78 с., 21 малюнка, 1 додаток, 15 джерел.

**Об’єкт дослідження:** процеси розробки та функціонування онлайн інструментів для графічної обробки фото.

**Мета роботи:** розробка онлайн інструменту для графічної обробки фото, що забезпечить можливість завантаження зображення, примінення ефекту двотонового накладання кольору на фото та збереження зображення до файлової системи користувача.

**Методи:** розробка алготму графічної обробки зображень засобами мови програмування JavaScript, створення макетів користувацького інтерфейсу, ручне тестування, технології створення вебсторінок HTML та CSS, застосування принципів розробки Rich Internet Application.

**Ключові слова:** графічна обробка фото, онлайн застосунок, Rich Internet Application, JavaScript.

**ABSTRACT**

Bachelor's qualification work: 78 pages, 21 drawings, 1 addition, 15 sources.

**Object of research:** processes of development and operation of online tools for graphic photo processing.

**Purpose:** development of an online tool for graphic photo processing, which will provide the ability to upload an image, apply the effect of two-tone color overlay on the photo and save the image to the user's file system.

**Methods:** development of an algorithm for graphic image processing using JavaScript programming language, creation of user interface layouts, manual testing, technologies for creating web pages HTML and CSS, application of the principles of Rich Internet Application development.

**Keywords:** графічна обробка фото, онлайн застосунок, Rich Internet Application, JavaScript.

**ЗМІСТ**

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ 4](#_Toc75005952)

[ВСТУП 5](#_Toc75005953)

[РОЗДІЛ 1 ОГЛЯД ОНЛАЙН ІНСТРУМЕНТУ ДЛЯ ГРАФІЧНОЇ ОБРОБКИ ФОТО 7](#_Toc75005954)

[1.1 Опис предметної області 7](#_Toc75005955)

[1.2 Огляд існуючих програмних рішень 8](#_Toc75005956)

[1.3 Функціональні вимоги 12](#_Toc75005957)

[РОЗДІЛ 2 ПРОЕКТУВАННЯ 15](#_Toc75005958)

[2.1 Функціонал та структура системи 15](#_Toc75005959)

[2.2 Розробка дизайну 15](#_Toc75005960)

[РОЗДІЛ 3 РОЗРОБКА ОНЛАЙН ІНСТРУМЕНТУ ДЛЯ ГРАФІЧНОЇ ОБРОБКИ ФОТОГРАФІЙ 22](#_Toc75005961)

[3.1 Вибір засобів розробки 22](#_Toc75005962)

[3.2 Алгоритм обробки зображень 22](#_Toc75005963)

[3.3 Розробка користувацького інтерфейсу 32](#_Toc75005964)

[РОЗДІЛ 4 ТЕСТУВАННЯ 37](#_Toc75005965)

[4.1 Вибір виду тестування 37](#_Toc75005966)

[4.2 Проведення тестування 37](#_Toc75005967)

[ВИСНОВОК 42](#_Toc75005968)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 43](#_Toc75005969)

[ДОДАТОК А ВИХІДНИЙ КОД 45](#_Toc75005970)

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

**HTML** – Hypertext Markup Language

**CSS** – Cascading Style Sheets

**PNG (Portable Network Graphics)** –растровий формат збереження графічної інформаці, що використовує стиснення без втрат

**JPG (Joint Photographic Experts Group)** –растровий формат збереження графічної інформації, що використовує стиснення з втратами якості зображення

**WebP** – формат ущільнення зображень з втратами і без втрат якості

ВСТУП

Тема дипломної роботи присвячена розробці онлайн інструменту для графічної обробки фото за допомогою мов веб-програмування.

Фотографія відіграє важливу роль у сучасному світі. Це один із найдоступніших засобів передачі візуальної інформації, тому їй відведено чільне місце у масовій культурі людтсва. Фотографії широко розповсюджені у засобах масової інформації, а із початком ери інтернету і смартфонів, оснащених якісними фотокамерами, для десятків мільйонів користувачів соціальних мереж стало буденністю щоденно створювати і поширювати чималу кількість зображень.

Паралельно із поширенням фотографії зростає потреба у графічних редакторах. Завдяки їм користувачі можуть суттєво покращити зображення, як для естетичного задоволення, так і для практичних або комерційних цілей.

Темпи появи нових засобів для обробки фотографій дозволяють припустити, що сфера онлайн графічних редакторів і надалі буде демонструвати стрімке зростання.

Метою даної дипломної роботи є розробка графічного редактора, що обробляє зображення за допомогою двотонового накаладання кольору. Для створення конкурентного продукту можна виділити наступні вимоги:

* простота використання – для користування не потрібно мати навичок у обробці фотографії;
* чіткий та зрозумілий інтерфейс, що сприяє полегшенню роботи з редактором;
* швидкість обчислень, алгоритм опрацьовує зображення у реальному часі без затримок.

Бурхливий розвиток веб-технологій сприяє переходу багатьох сфер настільних програм у онлайн, і графічні редактори не стали винятком. Рішення створення онлайн-інструменту має ряд переваг, таких як швидкість розробки, відсутність потреби у портуванні програми під різні операційні системи, простота доступу для користувача.

Для успішної реалізації завдання виділено наступні етапи розробки:

1. Проаналізувати ринок онлайн графічних редакторів, їх функціонал та доступність для користувача.

2. На основі проведеного аналізу створити функціональні вимоги для власного проекту.

3. Розробити дизайн користувацього інтерфейсу.

4. Реалізувати програмний алгоритм обробки зображень.

5. Створити онлайн-застосунок засобами мов веб-програмування.

6. Провести тестування на коректну роботу програми та відповідність поставленим вимогам.

РОЗДІЛ 1  
ОГЛЯД ОНЛАЙН ІНСТРУМЕНТУ ДЛЯ ГРАФІЧНОЇ ОБРОБКИ ФОТО

1.1 Опис предметної області

Комп’ютерна графіка являє собою одну із сучасних технологій створення

різноманітних зображень за допомогою апаратних і програмних засобів комп’ютера, відображення їх на екрані монітора і потім зберігання у файлі або друці на принтері.

Одним з найбільш поширених типів програм для роботи з графікою є графічні редактори. Графічні редактори – це прикладні програми, призначені для створення й обробки графічних зображень на комп’ютері в діалоговому режимі [1].

Rich Internet application (RIA, насичений інтернет-додаток) це клієнт-серверний додаток, в якому в якості клієнта виступає браузер, а в якості сервера – веб-сервер, доступний через Інтернет, що володіє функціональністю традиційних програм для настільних ком'ютерів.

Перехід від традиційних ком’ютерних програм до повнофункціональних веб-додатків зумовлений стрімким розвитком веб-технологій за останні десятиріччя, чому сприяло значне покращення швидкості передачі даних через Інтернет, а також істотне збільшення користувачів Всесвітньої павутини.

Завдяки своїй специфіці, насичені інтернет-застосунки мають ряд ключових переваг, в порівнянні зі звичайними комп’ютерними програмами.

До сильних сторін можна віднести відсутність потреби завантаження програми на персональний комп’ютер, що розкривається одразу через декілька аспектів.

По-перше, простий та швидкий доступ до веб-додатку з різних платформ, в тому числі мобільних, що значно заощаджує час на розробку застосунку для різних операційних систем. Також це розширює коло потенційних клієнтів веб-додатку до всіх користувачів всесвітньої мережі Інтернет.

По-друге, персональний комп’ютер клієнта необов’язково має бути потужним, адже обчислення частково або повністю можуть бути виконані на сервері.

По-третє, оновлення веб-додатку відбувається одразу на сервері, тому команді розробників непотрібно слідкувати за підтримкою та виправленням більш старих версій.

На противагу, насичені інтернет-застосунки мають деякі слабкі сторони.

По-перше, зазвичай для роботи веб-додатку потрібне підключення скриптів Javascript, яке може бути вимкнуте у браузері користувача. У такому випадку, робота застосунку буде некоректною.

По-друге, пошуковим системам дещо складніше індексувати та представляти у пошуковій видачі онлайн-застосунки, на відміну від звичайних багатосторінкових сайтів.

По-третє, веб-додатки потребують стабільного інтернет-з’єднання, адже інакше не зможуть працювати. В той же час, вільний доступ до Всесвітньої мережі давно ввійшов в повсякденне життя сучасної людини.

Беручи до уваги вищесказане, насичені інтернет-додатки це перспективний напрям розвитку прогрограмного забезпечення.

1.2 Огляд існуючих програмних рішень

Доволі поширеними сучасними онлайн-застосунками з обробки зображень є Adobe Photoshop Express та Pixlr Editor. Розглянемо їх інструменти та дизайн рішення.

Adobe Photoshop Express – багатофункціональний онлайн-сервіс з графічної обробки зображень, розроблений американською компанією Adobe Systems, та вперше представлений у 2008 році.

В порівнянні з комп’ютерною програмою Photoshop від однойменної компанії, яка призначена для професійного використання, Photoshop Express значно менш функціональний, але за рахунок цього має набагато нижчий поріг входу для користувача.

Спрощений інтерфейс та мінімальна кількість інструментів роблять застосунок зрозумілим та легким в освоєнні (Рисунок 1.1).

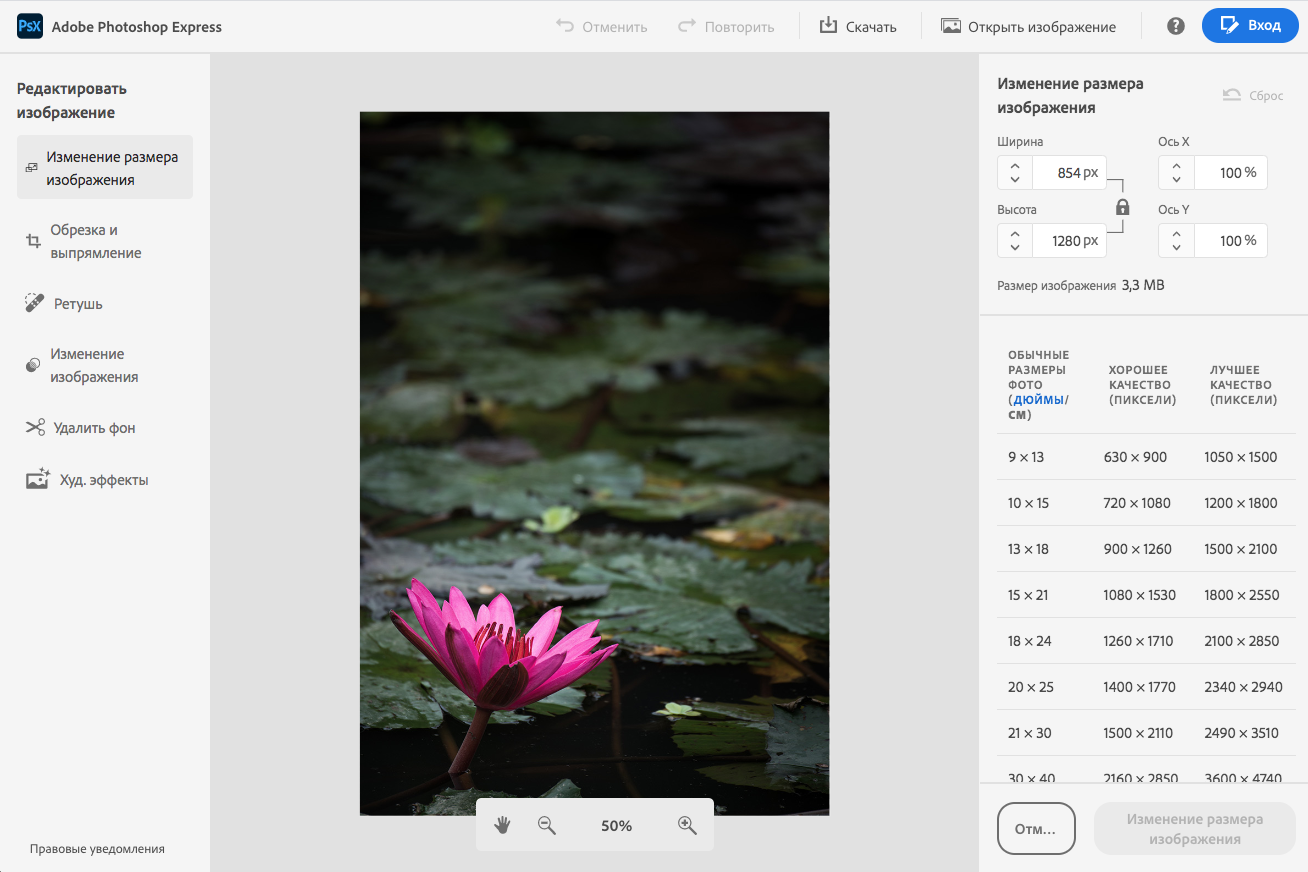


Рисунок 1.1 – Інтерфейс Photoshop Express

Photoshop Express безкоштовний, але для користування засобами редагування зображень потребує реєстрації облікового запису в екосистемі Adobe.

Даний графічний редактор надає базові інструменти редагування, серед яких:

* налаштування світла: зміна експозиції, контрасту, світлих областей та тіней;
* налаштування кольору: зміна температури, відтінку, барвистості та насиченості;
* налаштування ефектів: зміна чіткості та додавання “віньєтки” – ефекту, що зменшує рівень яскравості по краям зображення, виділяючи об’єкт у центрі [2].

Adobe Photoshop Express дозволяє змінювати ширину та висоту зображення, здійснити поворот та віддзеркалення по горизонтальній або вертикальній осі. Також доступна функція обрізки – користувач може задати значення розміру вихідного зображення у пікселях, або виділити потрібну область безпосередньо на фотографії.

Ключова функція це видалення фону. За допомогою технологій комп’ютерного зору графічний редактор може визначити головний об'єкт на фотографії та видалити, замінити або розмити іншу частину зображення. Даний інструмент доволі непогано працює з простими формами, але не завжди показує задовільний результат за наявності декількох предметів у композиції.

На вибір користувача, засобами Adobe Photoshop Express можна додати художній ефект на зображення з переліку запропонованих, серед яких інверсія кольорів, ефекти “масляна фарба”, “мультфільм”, “поп-арт”, “пікселізація”, “відблиск об’єктиву”.

Результатом обробки фотографії є файли растрових форматів збереження графічної інформації JPEG або PNG, які можуть бути завантажені на персональний комп’ютер користувача. За потреби онлайн-застосунок надає можливість вибору якості та зміни назви файлу.

Загалом, Adobe Photoshop Express зручний у використанні, але потреба реєстрації та невелика кількість доступних налаштувань може відштовхнути деяких користувачів від вибору на користь даного інструмента.

Наступним програмним аналогом за темою розробки є хмарний графічний редактор Pixlr Editor, який дозволяє редагувати фотографії або інші растрові зображення. Передусім призначений для користувачів, що не мають базової підготовки та спеціальних знать в області комп’ютерної графіки. Даний сервіс має простий інтерфейс та зрозумілі алгоритми роботи, що дозволяє швидко і без

проблем його освоїти, але в той же час дає можливість проводити широкий спектр маніпуляцій з зображеннями[3].

Першу версію Pixlr Editor було створено та представлено шведським розробником програмного забезпечення Олою Севандерсоном у 2008 році. Наразі підтримка онлайн-застосунку здійснюється міжнародним агентством Inmagine Group.

Незважаючи на відносну простоту інтерфейсу, Pixlr Editor має чималу кількість інструментів для редагування зображення. Вигляд інтерфейсу наведено на рисунку 1.2.

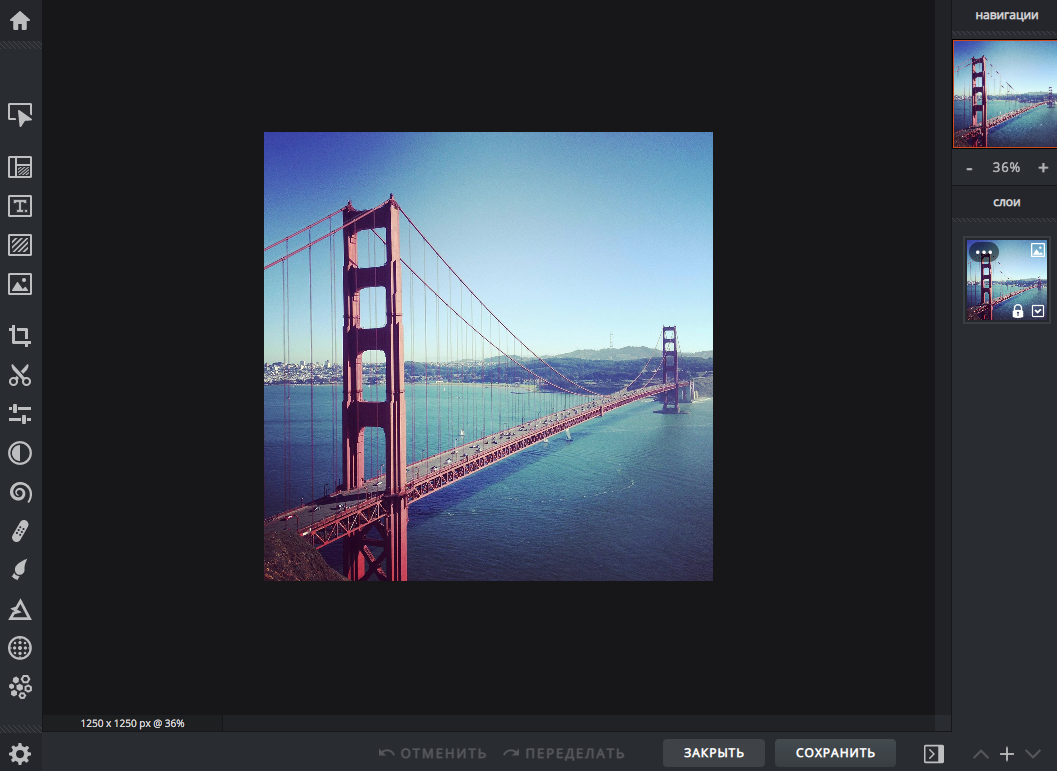


Рисунок 1.2 – Інтерфейс Pixlr Editor

Веб-застосунок дозволяє створювати прості форми — квадрат, коло, трикутник, зірка, лінія.

Pixlr Editor надає можливості додати текст з-поміж запропонованих стилів або створити власний. Серед доступних модифікацій тексту — зміна шрифту, прозорості, кольору заливки, розміру, міжрядкового інтервалу, відстані між літерами, додавання тіні або контуру. Доступне вирівнювання по центру, лівому або правому краю.

Вихідні зображення можуть буті завантажені в форматах PNG, JPG, WEBP або PXZ – нативному форматі, розробленому для збереження повної структури проекту та налаштувань в Pixlr Editor, що дозволяє за потреби повноцінно продовжити редагування фотографії.

Pixlr Editor частково безкоштовний. Деякими функціями можна користуватись безкоштовно, а для роботи з іншими потрібно мати платну підписку, що не найкращим чином підходить для разового використання. Тому, незважаючи на потужний функціонал, через комерційну направленість Pixlr Editor не є універсальним засобом графічних зображень.

1.3 Функціональні вимоги

Виходячи з аналізу наявного функціоналу основних конкурентів, їх основних переваг та недоліків можна сформувати ряд функціональних вимог, які необхідні для створення програмного забезпечення.

Спершу розробимо сценарій використання взаємодії системи та користувача (Таблиця 1.1).

Таблиця 1.1 – Сценарій використання

|  |  |
| --- | --- |
| Назва | Обробка зображення. |
| Опис | Користувач системи завантажує зображення з персонального ком’ютера та обробляє його інструментами, наданими системою. |
| Учасники | Користувач системи. |

Продовження таблиці 1.1

|  |  |
| --- | --- |
| Основний сценарій | 1) Користувач відкриває сторінку веб-додатку у браузері на персональному ком’ютері.  2) Користувач завантажує зображення з файлової системи.  3) Система відображає завантажене зображення |
| Розширення сценаріїв | 3.1) Система виявляє, що даний файл не є зображенням  3.2) Система виявляє, що розмір файлу зображення перевищує максимально допустимий |

Таблиця 1.2 – Опис функціональної вимоги REQ001

|  |  |
| --- | --- |
| Номер | REQ001 |
| Назва | Завантаження зображення |
| Опис | Система дозволяє завантажити зображення, обране користувачем у файловій системі для подальшої графічної обробки |

Таблиця 1.3 – Опис функціональної вимоги REQ002

|  |  |
| --- | --- |
| Номер | REQ002 |
| Назва | Відображення зображення |
| Опис | Система відображає щойно завантажене зображення на екрані користувача |

Таблиця 1.4 – Опис функціональної вимоги REQ003

|  |  |
| --- | --- |
| Номер | REQ003 |
| Назва | Застосування пресетів |

Продовження таблиці 1.4

|  |  |
| --- | --- |
| Опис | Система довзволяє обрати та примінити один із запропонованих пресетів обробки завантаженого зображення |

Таблиця 1.5 – Опис функціональної вимоги REQ004

|  |  |
| --- | --- |
| Номер | REQ004 |
| Назва | Зміна стилю обробки |
| Опис | Система довзволяє обрати та примінити один із запропонованих стилів обробки завантаженого зображення |

Таблиця 1.6 – Опис функціональної вимоги REQ005

|  |  |
| --- | --- |
| Номер | REQ005 |
| Назва | Збереження зображення |
| Опис | Система довзволяє зберегти зображення до файлової системи персонального ком’ютера користувача |

РОЗДІЛ 2  
ПРОЕКТУВАННЯ

2.1 Функціонал та структура системи

Обробка проходить на локальному пристрої користувача, що дає ряд переваг як зі сторони розробника, так і клієнта:

1. Скрипт завантажено лише один раз під час першого заходу на веб-сторінку.

2. Застосунок зможе продовжити роботу навіть за втрати інтернет-з’єднання в процесі обробки.

3. Суттєво зменшується навантаження на сервер та знижується об’єм передачі даних між клієнтом та сервером.

Даний інструмент надає функціонал для завантаження фото з файлової системи користувача, подальшу графічну обробку та збереження вихідного зображення.

У рамках графічної обробки онлайн-застосунок дозволяє змінювати кольори для двотонового накладання, приміняти пресети та змінювати стилі обробки. Пресет – набір заздалегідь визначених параметрів конфігурації ефектів. Їх застосування значно спростить роботу з застосунком для потенційного користувача.

2.2 Розробка дизайну

Розробимо макети сторінок інформаційної системи за допомогою векторного графічного редактора Figma.

Використання темної теми це поширений тренд у проектуванні дизайну графічних редакторів, адже це дозволяє зменшити напругу очей під час тривалої роботи, краще зчитувати інформацію при слабкому освітлені та значно зменшити енергоспоживання акумулятора на пристроях з сучасними OLED-дисплеями[4].

При розробці дизайну функціональні елементи системи було виділено бірюзовим кольром. Основний текст пофарбовано білим кольором. Допоміжний текст має привертати менше уваги, тому його виконано у сірому кольорі (Рисунок 2.1).



Рисунок 2.2 – Вікно завантаження зображення

Після завантаження зображення з’являється робоча область обробки фотографії. Її макет представлено на Рисунку 2.2.

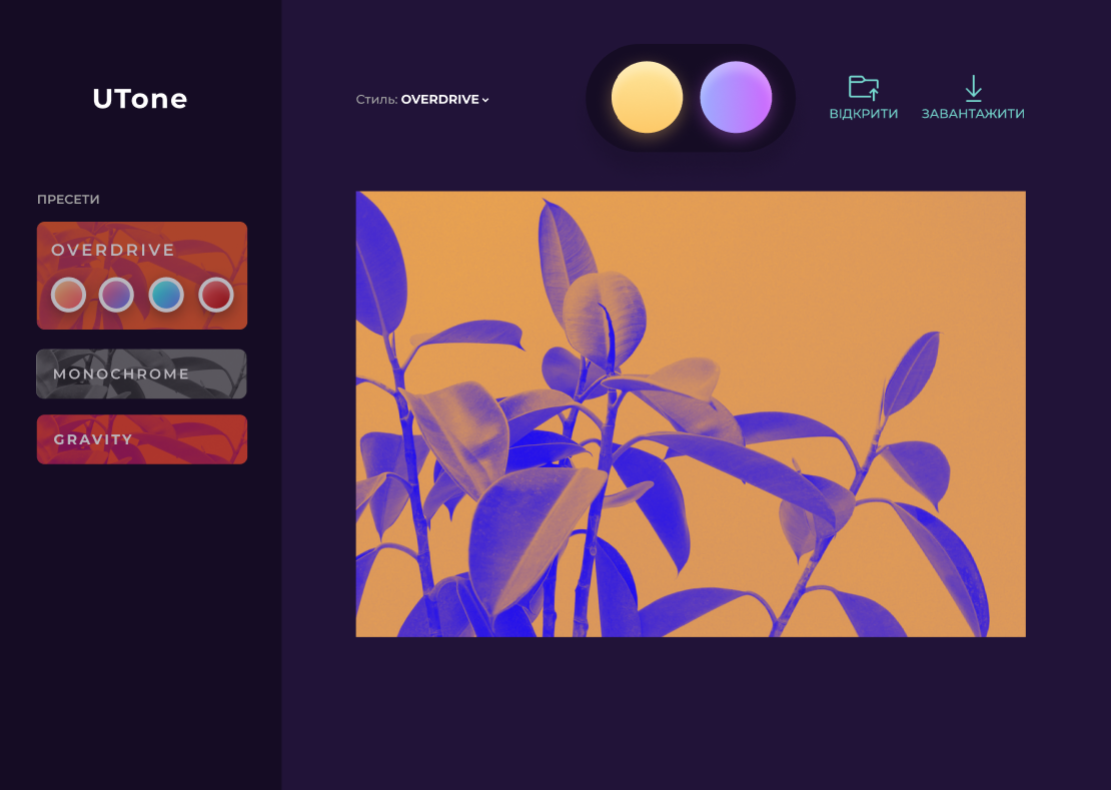


Рисунок 2.2 – Робоча область

На бічній панелі справа розташовано пресети – заздалегідь відібрані та налаштовані ефекти, що можуть бути примінені до завантаженого зображення. Їх використання має значно спростити і пришвидшити обробку фотографії для користувача.

На верхній панелі розташований список стилів для обробки, з яких користувач може обирати той, що краще відповідає посталеним цілям у графічній обробці. Центральний елемент верхньої панелі та робочої області загалом це палітра для двотонового фарбування зображення. З метою покращення естетичного вигляду, кольори на палітрі мають ефект об’ємності, що досягається завдяки додаванню світіння та тіням.

Також на верхній панелі розташовані функціональні кнопки для завантаження або збереження вихідного зображення.

З метою уникнення перевантаження інтерфейсу, деякі елементи з’являються при наведенні курсора. Наприклад, пресети на боковій панелі розкриваються та стають яскравішими. Макет пресетів при наведені курсора зображено на рисунку 2.3.



Рисунок 2.3 – Пресети при наведені курсора

За тим же принципом при наведені курсора на активний стиль з'являється випадаючий список з переліком всіх доступних для вибору стилів (Рисунок 2.4).



Рисунок 2.4 – Випадаючий список стилів при наведені курсора

На думку Маценко Г.В. для оцінки кольору людині зручно використовувати такі атрибути[5]:

1. Тон (відтінок) кольору асоціюється в людській свідомості з забарвленням предмета певним типом фарби. Саме він дозволяє людині відрізняти кольори між собою. Фізично тон кольору можна визначити переважаючою (усередненою) довжиною хвилі в спектрі випромінювання. Наприклад, світло, в якому переважає хвиля з довжиною 450 нм, буде сприйматися як відтінок синього кольору. Людське око спроможне розрізняти 350 тис. різних кольорів, хоча є й інші дані. Для характеристики відтінків вводять поняття яскравості і насиченості.

2. Яскравість (світлість) визначається енергією, інтенсивністю випромінювання на одиницю площі і виражає кількість сприйнятого світла. Значення світлості залежить від амплітуди електро-магнітних коливань. Наприклад, білий колір має максимальну яскравість (100%), а чорний колір навпаки дає повну відсутність яскравості (0%), тобто чим менша яскравість, тим темніший відтінок.

3. Насиченість характеризує рівень чистоти кольору і визначає кількість білого у відтінку того чи іншого кольору. Саме вона виражає співвідношення між основною домінуючою компонентою світла і рештою хвиль, що формують колір, тобто показує, наскільки даний колір відрізняється від білого. Чим вища насиченість, тим сильніше і ясніше відчувається тон кольору. Зниження насиченості приводить до того, що колір стає нейтральним без чітко вираженого тону. В ідеально чистому кольорі домішки білого відсутні (насиченість 100%). Якщо, наприклад, до чистого червоного кольору додати білий, то одержимо світлий блідо-червоний колір (низька насиченість). При насиченості, що дорівнює 0%, будь-який колір стає білим. Цей атрибут у людській свідомості пов’язаний із кількістю пігменту, фарби.

За цим принципом побудовано колірну модель HSL, в якій кожен колір задано чисельним значенням відтінку в діапазоні 0-360, чисельним значенням освітленості та насиченості у діапазонах 0-100. Саме її буде використано у веб-застосунку для задавання кольору на палітрі при наведені курсору, що передбачено при розробці макетів. Значення можуть бути задані текстом у числому полі або за допомогою повзунків (Рисунок 2.5).



Рисунок 2.5 – Панель вибору кольору

Для веб-сторінки шириною менше за 1350 пікселів система адаптує елементи для кращого відображення вмісту, що передбачено у дизайні застосунку. Наприклад, зникає текст біля кнопок, натомість іконки стають більшими. Макет представлено на рисунку 2.6.

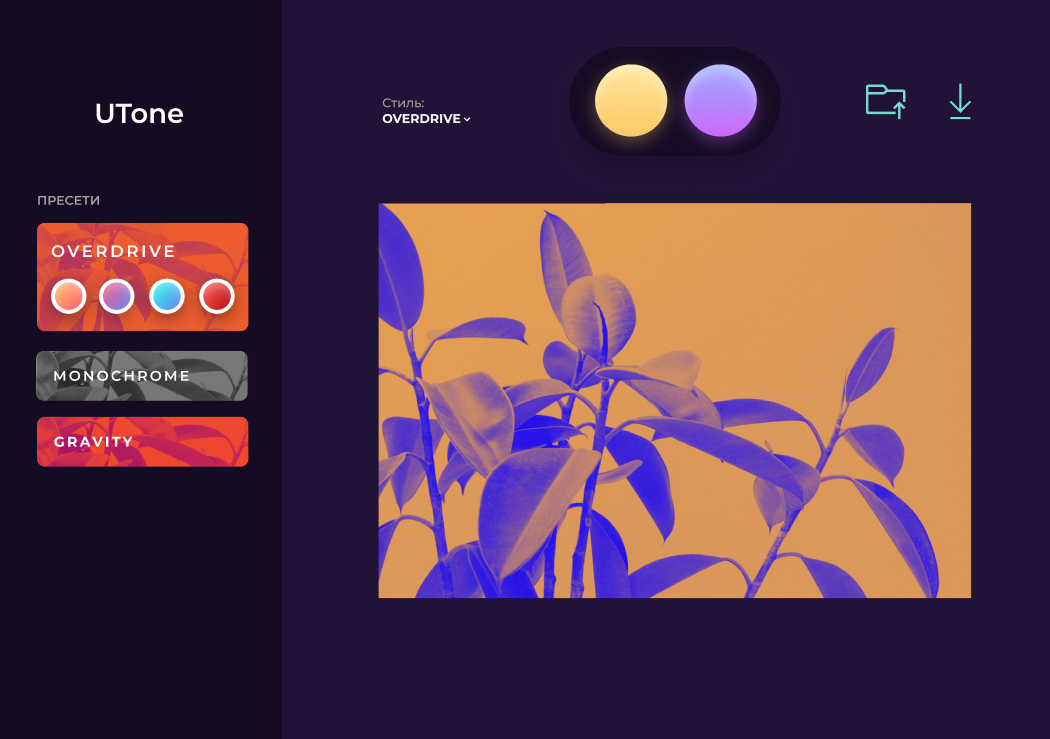


Рисунок 2.6 – Макет робочої область для веб-сторінки, вужчої за 1350px

РОЗДІЛ 3  
РОЗРОБКА ОНЛАЙН ІНСТРУМЕНТУ ДЛЯ ГРАФІЧНОЇ ОБРОБКИ ФОТОГРАФІЙ

3.1 Вибір засобів розробки

Для розробки веб-застосунку використані технології HTML, CSS та JavaScript.

HTML це стандартизована мова опису структури веб-сторінок у мережі Інтернет, що використовує спеціальні оператори – теги. За допомогою тегів створюється розмітка сторінки, з якою користувач зможе взаємодіяти у браузері [6].

CSS це мова опису зовнішнього стилю веб-сторінок. Завдяки їй можна втілити створені макети, задаючи потрібні кольори, шрифти, тіні, розміщення елементів та інше [6].

JavaScript це зручна та багатофункціональна об'єктно-орієнтована мова програмування. У цьому проекті її використано для реалізації алгоритму обробки зображень та анімацій веб-сторінки.

Для компресії зображень після редагування використано безкоштовну Javascript бібліотеку Compress.js, створену американським розробником Скотом Ананяном. Її функціонал дозволяє змінювати розмір вихідного зображення, задавати для нього максимальну висоту та ширину.

3.2 Алгоритм обробки зображень

Маніпуляції з зображенням будуть проведені на HTML-елементі canvas, що використовується для відображення графічних об’єктів на веб-сторінці засобами мов програмування.

Щоб відтворити та мати можливість в подальшому обробляти завантажене зображення на canvas, спершу потрібно конвертувати його з форматів JPG або PNG до base64. Base64 це поширений спосіб кодування довільних двійкових даних в ASCII текст.

Стандартний Javascript об’єкта FileReader довзоляє веб-застосункам асинхронно читати вміст файлів на пристрої користувача. За допомогою його методу readAsDataURL завантажене зображення декодується в Base64-рядок.

Перший крок алгоритму обробки це конвертування вхідного зображення до відтітків сірого. На відміну від колірної моделі RGB, в якій кожен колір визначається градаціями каналів червоного, зеленого та синього кольорів [7], кожен піксель у моделі відтінків сірого кодує інформацію лише про свою інтенсивність. Існує декілька формул, що використовуються для конвертації зображення з RGB до відтінків сірого [8]:

1. Метод середнього значення

, (3.1)

де *G* — вихідне значення відтінків сірого;

*R* — інтенсивність каналу червоного кольору;

*G* — інтенсивність каналу зеленого кольору;

*B* — інтенсивність каналу синього кольору;

1. Метод яскравості

(3.2)

1. Метод освітленості

(3.3)

Кожен з методів по різному обраховує значення, що позначається на кінцевому зображенні. Результати конвертування зображень за різними методами наведено на рисунках 3.1 – 3.2.

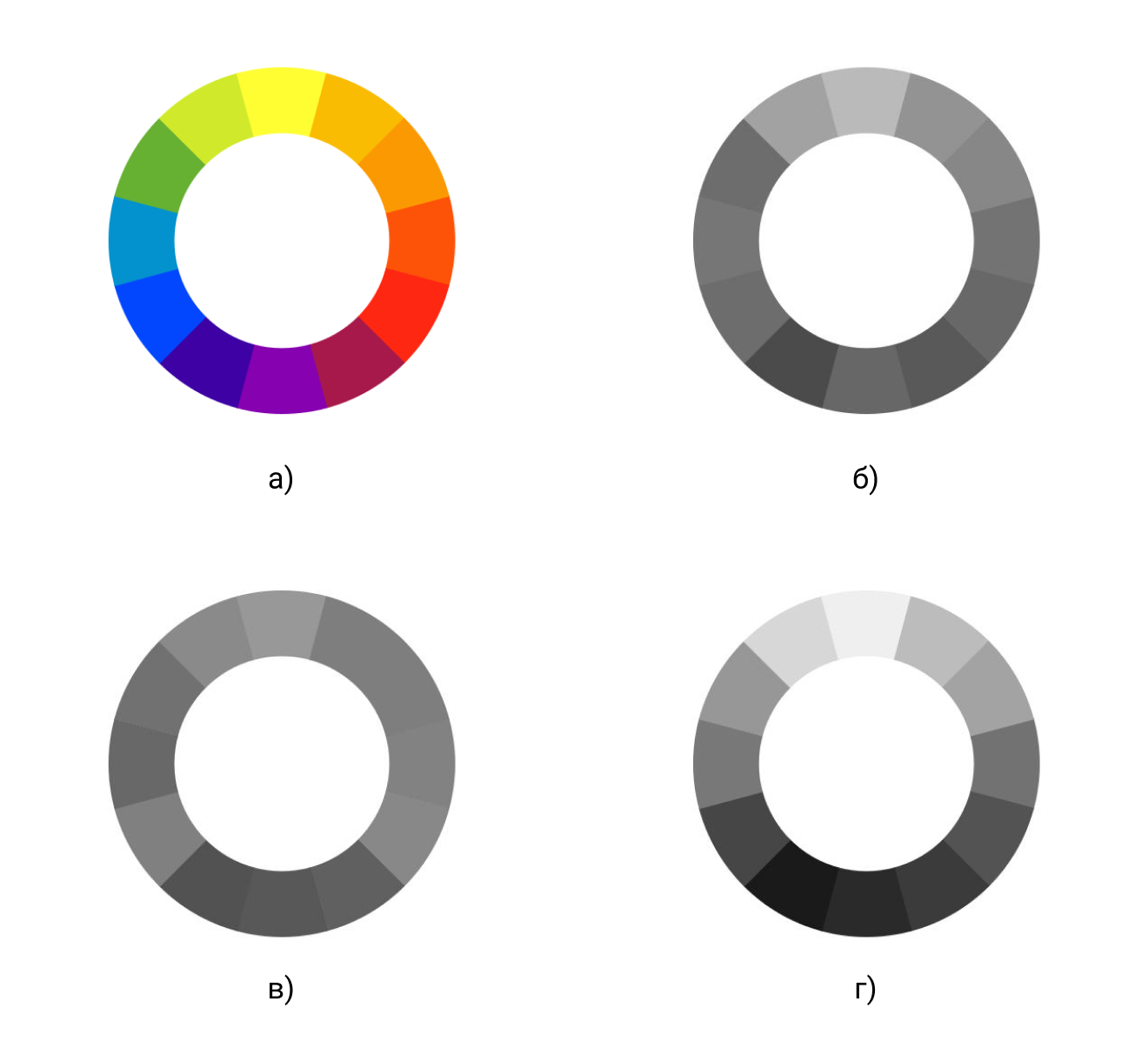


Рисунок 3.1 Результати конвертування:

а – зображення в кольорі; б – зображення оброблене методом середнього значення; в – зображення оброблене методом яскравості; г – зображення оброблене методом освітленості;

Як бачимо, метод освітленності найбільш контрастний та краще сприймається людським оком — кольори спектру не змішані, а поступово градуйовані. Оскільки він має суттєву перевагу серед інших методів, його використання у алгоритмі буде доцільним.



Рисунок 3.2 Результати конвертування:

а – зображення в кольорі; б – зображення оброблене методом середнього значення; в – зображення оброблене методом яскравості; г – зображення оброблене методом освітленості;

Для конвертації зображення використано Javascript метод getImageData(), який повертає ImageData об’єкт[9]. Властивість data об’єкту ImageData надає одновимірний масив, що містить послідовно розташовані елементи з даними інтенсивності каналів червоного, зеленого, синього кольорів та значення прозорості для кожного пікселя елементу canvas. Таким чином, за допомогою циклу for можна покроково обрахувати нові значення інтенсивності у масиві та вивести оновлене зображення.

Результат виконання першого кроку наведено на рисунку 3.3.



Рисунок 3.3 – Перший крок роботи алгоритму

Наступний крок – накладання першого кольору на зображення.

За допомогою globalCompositeOperation можна задати параметри режиму накладання або режиму змішування шарів на елементі canvas.

Режим композиції задає параметри накладання вмісту двох шарів один на одного. На таблиці 3.1 наведено перелік налаштувань режиму композиції, що можуть бути використані при обробці зображення.

Таблиця 3.1 – Режими композиції

|  |  |
| --- | --- |
| source-over | Це налаштування за замовчуванням і наносить новий шар поверх існуючого вмісту canvas. |
| source-in | Нова шар з’являється лише там, де і новий шар, і полотно призначення перекриваються. Все інше стає прозорим. |
| source-out | Новий шар з'являється лише там там, де він не перекриває існуючий вміст полотна. |
| source-atop | Новий шар з'являється лише там там, де він перекриває наявний вміст полотна. |
| destination-over | За наявним вмістом полотна з'являється новий шар. |
| destination-in | Існуючий вміст полотна зберігається там, де новий шар та наявний вміст canvas перекриваються. Все інше стає прозорим. |
| destination-out | Існуючий вміст зберігається там, де він не перекриває новий шар. |
| lighter | Існуюче полотно зберігається лише там, де воно перекриває нову шар, який з'являється за вмістом полотна. |
| copy | Показано лише новий шар. |
| xor | Шари стають прозорими там, де перекриваються. |

На таблиці 3.2 наведено перелік налаштувань режиму змішування кольорів, що можуть бути примінені при графічній обробці зображення.

Таблиця 3.2 – Режими змішування кольорів

|  |  |
| --- | --- |
| multiply | Пікселі верхнього шару множаться з відповідним пікселем нижнього шару. Колір, що отримується у результаті завжди темніший за ті, що змішувались між собою. Чорний колір залишається чорним на фінальному шарі, а білий не змінює зображення. |
| screen | Пікселі шарів інвертуються, потім множаться і знову інвертуються. У результаті отримується більш світле зображення. |
| overlay | Поєднання режимів multiply та screen. Темні частини нижнього шару стають темнішими, а світлі навпаки світлішими. |
| darken | При перетинанні залишається найтемніший піксель з нижнього та верхнього шару. |
| lighten | При перетинанні залишається насвітліший піксель з нижнього та верхнього шару. |
| color-dodge | Значення пікселів нижнього шару діляться на інвертовані значення пікселів верхнього шару. |
| color-burn | Значення пікселів нижнього шару діляться на інвертовані значення пікселів верхнього шару, а потім інвертуються. |
| hard-light | Поєднання режимів multiply та screen, з заміною верхнього та нижнього шару. |
| soft-light | Так, як і hard-light, але вихідне зображення менш контрастне. |

Продовження таблиці 3.2

|  |  |
| --- | --- |
| difference | Від більшого значення нижнього або верхнього шару віднімаються менші значення з обох шарів. |
| exlusion | Так, як і difference, але вихідне зображення менш контрастне. |
| hue | Збережена яскравість та насиченість нижнього шару, відтінок з верхнього шару. |
| saturation | Збережена яскравість та відтінок нижнього шару, значення насиченості з верхнього шару. |
| color | Збережена яскравість нижнього шару, значення відтінку та насиченості з верхнього шару. |
| luminosity | Збережена насиченість та відтінок нижнього шару, значення яскравості з верхнього шару. |

На рисунку 3.4 зображено приклад накладання пофарбованих шарів з різними налаштуваннями режиму змішування.

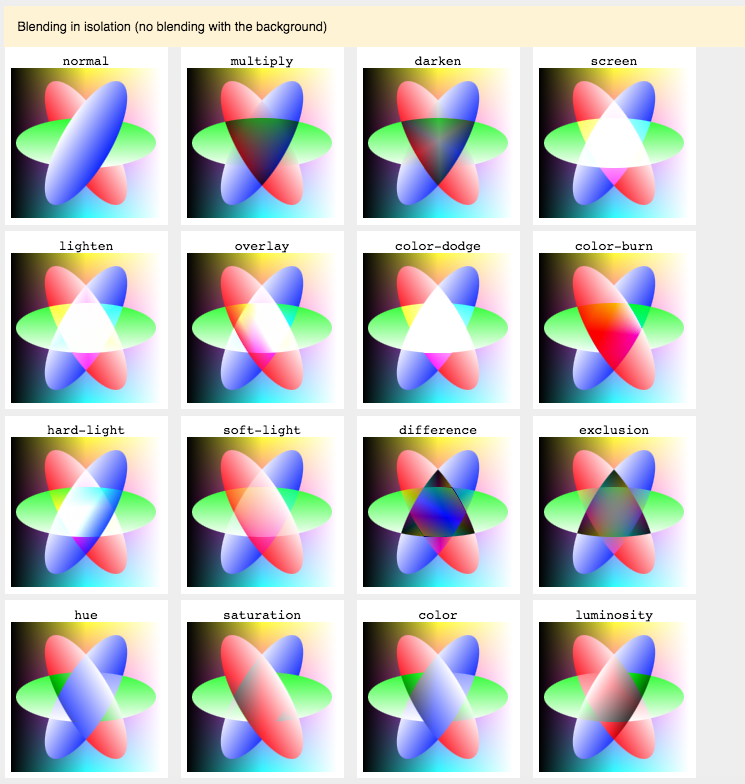


Рисунок 3.4 – Режими змішування

Поєднуючи різні налаштування режиму змішування та режиму композиції маємо можливість накладати поверх зображення у відтінках сірого пофарбовані шари та отримувати ефекти. Результат накладання першого кольору з режимом композиції source-over та режимом змішування multiply представлено на рисунку 3.5.



Рисунок 3.5 – Результат накладання першого кольору

Результат накладання першого кольору з режимом композиції source-over та режимом змішування multiply представлено на рисунку 3.6.



Рисунок 3.6 – Результат накладання другого кольору

3.3 Розробка користувацького інтерфейсу

Засобами технологій HTML та CSS макети інтерфейсу були втілені у веб-застосунок.

Нижче зображено вигляд завантажувального екрану. Напис анімовано, спершу він з’являється знизу з прозорістю 0, рухається догори, набираючи насиченість, а через 2 секунди розчиняється. Загальна тривалість завантажувального екрану становить 3 секунди (Рисунок 3.7).



Рисунок 3.7 – Завантажувальний екран

Нижче приведено вигляд екрану завантаження зображення. Користувач може завантажити фотографію перетягнувши її на виділену область, або настиснути на кнопку “Відкрийте” та обрати потрібний файл з файлової системи ком’ютера (Рисунок 3.8).

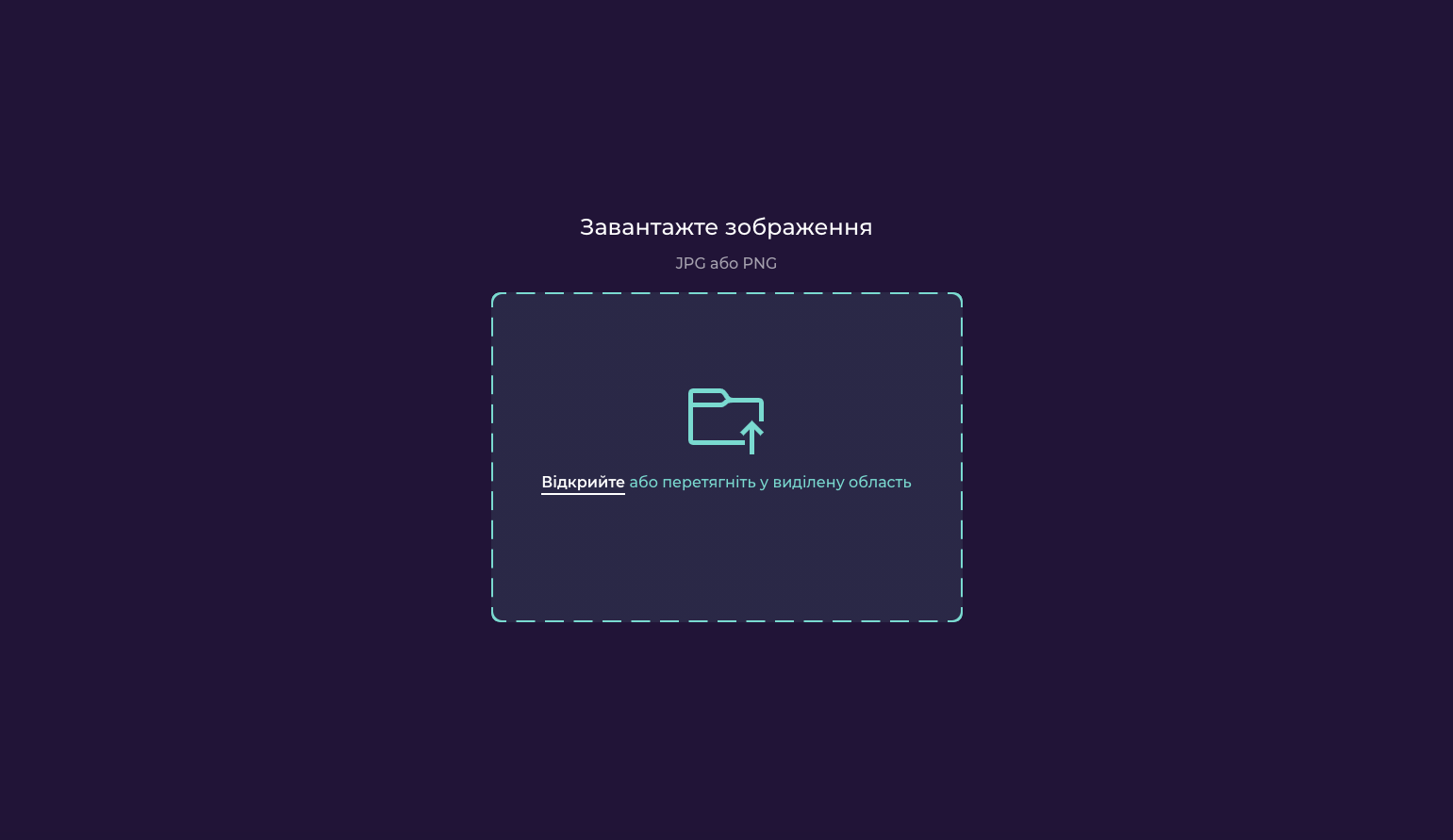


Рисунок 3.8 – Екран завантаження зображення

На рисунку 3.9 показний вигляд робочої області.



Рисунок 3.9 – Робоча область

Після завантаження зображення відбувається перехід до робочої області. Верхня панель має анімовану появу зверху, відповідно бічна панель плавно з’являється з лівого боку.

При наведені курсора на будь-який колір на палітрі з’являється панель, на якій користувач задає параметри відтінку, насиченості та освітленості. Тривалість анімації її появи становить 1 секунду (Рисунок 3.10).



Рисунок 3.10 – Панель задання кольору

Нижче продемонстровано панель задання стилю.(Рисунок 3.11)



Рисунок 3.11 – Панель задання стилю

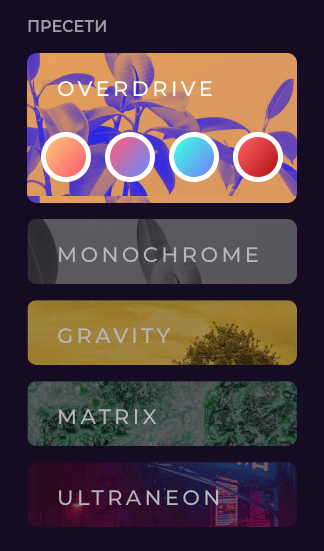


Рисунок 3.12 – Пресети при наведенні курсора

Як і було передбачено у макеті інтерфейсу, для екранів шириною менше за 1350px веб-застосунок адаптує зовнішній вигляд кнопок та стилю, що продемонстровано на рисунку 3.13.

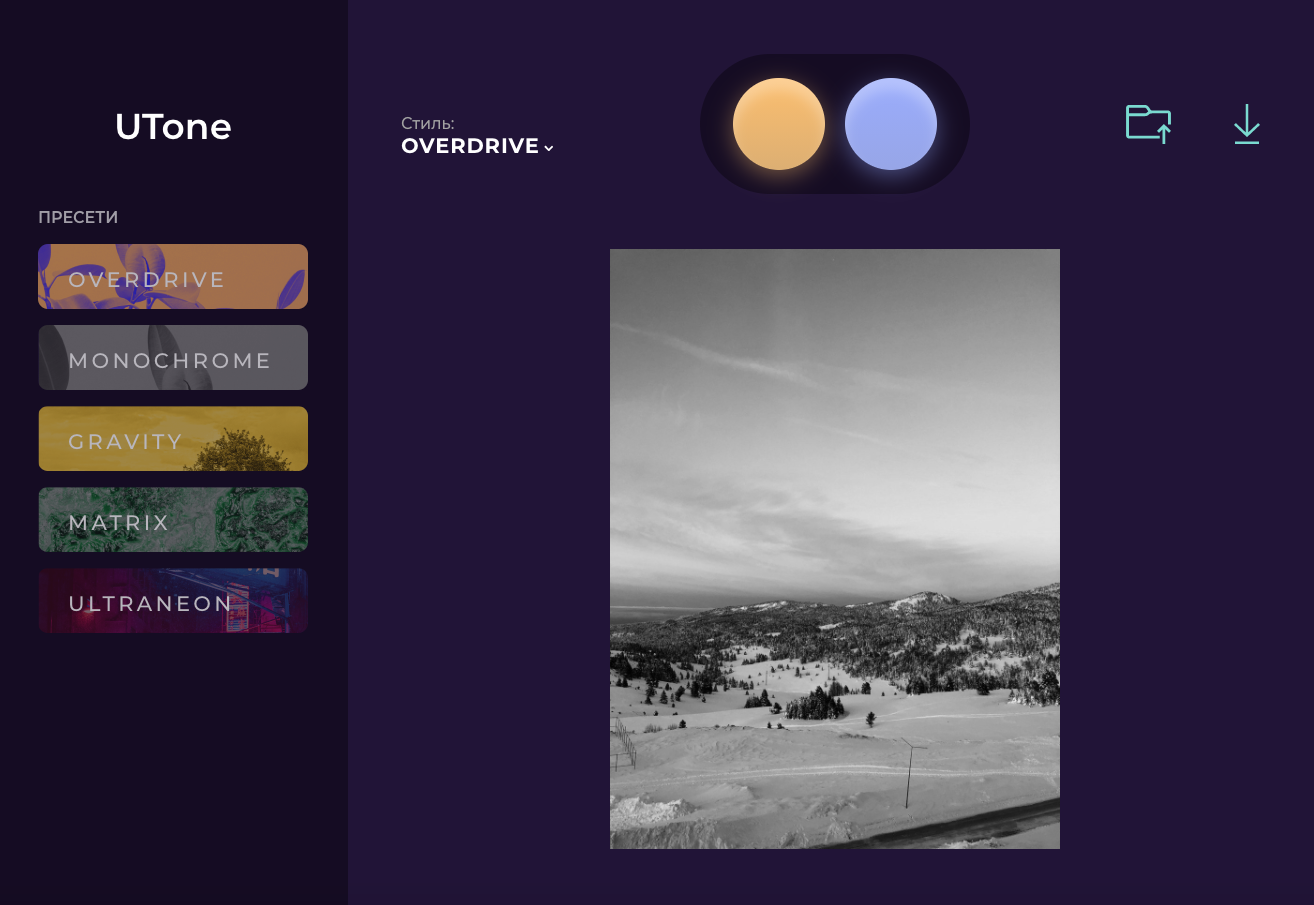


Рисунок 3.13 – Вигляд робочої область для веб-сторінки, вужчої за 1350px

РОЗДІЛ 4  
ТЕСТУВАННЯ

4.1 Вибір виду тестування

Тестування програм чи систем – це спосіб семантичної перевірки програми, який полягає в опрацюванні програмою послідовності різноманітних контрольних наборів тестів з відомими результатами. У процесі дослідження якості програмного забезпечення головна мета полягає в розробці ефективних методів та забезпеченні довготривалої працездатності системи із заданими функціональними характеристиками[10].

Тестування це заключний етап розробки програмного забезпечення, який має вагомий вплив у розробці якісного програмних засобів. Техніка тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою оцінки.

У ході розробки було примінено ручне тестування для перевірки коректності роботи графічного редактора.

4.2 Проведення тестування

Створено тест-кейси для основних тестових випадків системи. У таблиці 4.1 «Тестові випадки розробленого застосунку» наведено перелік всіх тестових випадків онлайн-застосунку.

Таблиця 4.2 – Тестові випадки розробленого застосунку

|  |  |
| --- | --- |
| Номер тестового випадку | Назва тестового випадку |
| ТC01 | Завантаження зображення |
| ТC02 | Обробка зображення |
| ТC03 | Зміна стилю |

Продовження таблиці 4.2

|  |  |
| --- | --- |
| ТC04 | Примінення пресетів |
| ТC05 | Збереження зображення |

Нижче розглянуто кожен тестовий випадок окремо та більш детально.

У таблиці 4.2 «Опис тестового випадку TC01» розглянуто тестовий

випадок TC01.

Таблиця 4.2 – Тестовий випадок ТC01

|  |  |
| --- | --- |
| Тестовий випадок | Завантаження зображення |
| Опис | Користувач завантажує зображення на сайт для подальшої графічної обробки |
| Передумови |  |
| Очікувана поведінка | 1. Користувач завантажує один файл формату JPG або PNG;  2. Система декодує зображення та відображає на екрані користувача. |
| Результат | Пройдено |

Нижче у таблиці 4.3 «Опис тестового випадку TC02» розглянуто тестовий

випадок TC02.

Таблиця 4.3 – Тестовий випадок ТВ02

|  |  |
| --- | --- |
| Тестовий випадок | Обробка зображення |
| Опис | Користувач задає параметри кольору для обробки, система обраховує і виводить оновлене зображення |
| Передумови | Користувач завантажив зображення |

Продовження таблиці 4.3

|  |  |
| --- | --- |
| Очікувані дії | 1. Користувач наводить курсор ком’ютерної миші на палітру кольорів на верхній панелі робочої області;  2. Система розкриває панель для задання параметрів кольору;  3. Користувач змінює значення відтінку, насиченості або освітленості за допомогою повзунків або вводить чисельне значення;  4. Система приймає нові значення параметрів, на їх основі виконує алгоритм обробки зображення;  5. Система відображає оновлене зображення на екрані користувача. |
| Результат | Пройдено |

Нижче у таблиці 4.4 «Опис тестового випадку TC03» розглянуто тестовий

випадок TC03.

Таблиця 4.4 – Тестовий випадок ТВ03

|  |  |
| --- | --- |
| Тестовий випадок | Зміна стилю |
| Опис | Користувач змінює стиль обробки зображення, система обраховує і виводить оновлене зображення |
| Передумови | Користувач завантажив зображення |
| Очікувані дії | 1. Користувач наводить курсор ком’ютерної миші на назву стилю на верхній панелі робочої області;  2. Система розкриває випадаючий список з переліком доступних стилів;  3. Користувач обирає один із стилів;  4. Система приймає нове значення стилю та виконує алгоритм обробки зображення; |

Продовження таблиці 4.4

|  |  |
| --- | --- |
|  | 5. Система відображає оновлене зображення на екрані користувача. |
| Результат | Пройдено |

Нижче у таблиці 4.5 «Опис тестового випадку TC04» розглянуто тестовий

випадок TC04.

Таблиця 4.5 – Тестовий випадок ТВ04

|  |  |
| --- | --- |
| Тестовий випадок | Примінення пресетів |
| Опис | Користувач обирає один із запропонованих пресетів, система обраховує і виводить оновлене зображення |
| Передумови | Користувач завантажив зображення |
| Очікувані дії | 1. Користувач наводить курсор ком’ютерної миші на групу пресетів на боковій панелі робочої області;  2. Система розкриває панель доступних пресетів;  3. Користувач обирає один із пресетів;  4. Система виконує алгоритм обробки зображення на основі значень, що передбачені у пресеті;  5. Система відображає оновлене зображення на екрані користувача. |
| Результат | Пройдено |

Нижче у таблиці 4.6 «Опис тестового випадку TC05» розглянуто тестовий

випадок TC05.

Таблиця 4.6 – Тестовий випадок ТВ05

|  |  |
| --- | --- |
| Тестовий випадок | Завантаження зображення |
| Опис | Користувач зберігає оброблене зображення на персональний ком’ютер |
| Передумови | Користувач завантажив зображення |
| Очікувані дії | 1. Користувач натискає на кнопку “Завантажити” на верхній панелі робочої області;  2. Система конвертує зображення у формат JPG;  3. Система проводить компресію зображення;  4. Система завантажує зображення на ком’ютер користувача. |
| Результат | Пройдено |

ВИСНОВОК

Мета дипломної роботи полягала у розробці онлайн інструменту для графічної обробки фото, що забезпечить можливість завантаження зображення, примінення ефекту двотонового накладання кольору на фото та збереження зображення до файлової системи користувача.

У ході виконання дипломного проектування було проаналізовано ринок онлайн-графічних редакторів, окреслено їх недоліки та переваги, що дозволило встановити функціональні вимоги для власної розробки.

Також було створено зручний дизайн користувацького інтерфейсу у темних тонах, що дозволяє зменшити напругу очей під час тривалої роботи та краще зчитувати інформацію при слабкому освітлені. Його реалізацію було втілено за допомогою технологій HTML, CSS та JavaScript.

Для обробки фотографій було створено програмний алгоритм, засобами якого відбувається конвертування фото у відтінки сірого, а потім проходить покрокове накладання кольорів на вхідне зображення.

Розроблено стилі та пресети для зручного користування, що відповідає вимогам створення програмного забезпечення, для якого непотрібно мати спеціальних навичок в сфері графічної обробки.

Створений застосунок було протестовано на відсутність програмних помилок та зручність використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Власій О.О Комп’ютерна графіка. Обробка растрових зображень: Навчально-методичний посібник / О. О. Власій, О. М. Дудка. – ІваноФранківськ: ДВНЗ «Прикарпатський національний університет імені Василя Стефаника», 2015. – 72 с.
2. Гілета І. В. Вагомі технологічні операції опрацювання ілюстрацій для публікації в поліграфічному виданні / І. В. Гілета, І. З. Миклушка, В. В. Пилип’юк // Наукові записки [Української академії друкарства]. — 2016. — No 1 (52). — С. 268.
3. Гришанина О. Н. Обзор графических онлайн редакторов, применимых в курсе информатики // Информационные технологии. Проблемы и решения: материалы Международной научно-практической конференции. Уфа, 2018. № 1 (5). С. 487—493.
4. Современные подходы к проектированию пользовательских интерфейсов [Електронний ресурс] / І. Д. Васильченко, А.Д. Васильченко // Столица науки. – 2020. – № 3. – Режим доступу до журн. : https://www.voicecapital.ru/mar2020/24032020.pdf
5. Маценко В.Г. Комп’ютерна графіка : [навчальний посібник] / В.Г. Маценко. – Ч. : Рута, 2009. – 343 с.
6. Квинт И. Создаем сайты с помощью HTML, XHTML и CSS на 100%. 2-е изд. – СПб.: Питер, 2012. – 448 с.
7. Федік Л. Ю. Аналіз колірних моделей RGB, CMY, CMYK / Л. Ю. Федік // Наукові нотатки. - 2010. - Вип. 29. - С. 217-221.
8. Three algorithms for converting color to grayscale. [Електронний ресурс] : [Веб-сайт] – Режим доступу: ohndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale (дата звернення 30.05.2021). – Назва з екрана.
9. Developer.mozilla.org [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://www.developer.mozilla.org/en-US/docs/Web/API /CanvasRenderingContext2D/globalCompositeOperation (дата звернення 31.05. 2021). – Назва з екрана.
10. Шамрай О. В. Методи тестування програмного забезпечення / О. В. Шамрай, Н. Н. Шаповалова // Гірничий вісник. - 2012. - Вип. 95. - С. 153-156.
11. Документація по HTML5 [Електронний ресурс] – Режим доступу до ресурсу: http://htmlbook.ru/html.
12. Блінова Т.О., Порєв В.М. Комп’терна графіка / За ред. В.М. Порєва. – К.: Видавництво «Юніор», 2004. – 456 с.
13. Горобець С.М. Основи ком’ютерної графіки: Навч.пос. / За ред. М.В. Левківського. – К.: Центр навчальної літератури, 2006. – 232 с.
14. Основні види тестування програмного забезпечення [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: http://lib.mdpu.org.ua/e-book/vstup/L11.htm (дата звернення 26.05.2021). – Назва з екрана.
15. HTML5 Canvas [Електронний ресурс] / W3Schools.com – Режим доступу: http://www.w3schools.com/html/html5\_canvas.asp (дата звернення 02.06.2021). – Назва з екрану.

ДОДАТОК А  
ВИХІДНИЙ КОД

1. index.js

let srcData;

let image = {};

const compress = new Compress();

let result;

let loaded = true;

let id = "duotone";

let secondary =

"hsl(" +

document.getElementById("secondary\_hue").value +

"," +

document.getElementById("secondary\_saturation").value +

"%," +

document.getElementById("secondary\_luminocity").value +

"%)";

let primary =

"hsl(" +

document.getElementById("primary\_hue").value +

"," +

document.getElementById("primary\_saturation").value +

"%," +

document.getElementById("primary\_luminocity").value +

"%)";

let background = true;

let primary\_blend = "multiply";

let secondary\_blend = "overlay";

let primary\_dark =

"hsl(" +

document.getElementById("primary\_hue").value +

"," +

(document.getElementById("primary\_saturation").value - 35) +

"%," +

(document.getElementById("primary\_luminocity").value - 5) +

"%)";

let primary\_dark\_hsla =

"hsla(" +

document.getElementById("primary\_hue").value +

"," +

(document.getElementById("primary\_saturation").value - 35) +

"%," +

(document.getElementById("primary\_luminocity").value - 10) +

"%, 0.5)";

document.getElementById("primary\_color").style.background =

"linear-gradient(180deg, " + primary + " 0%, " + primary\_dark + " 100%)";

document.getElementById("primary\_color").style["boxShadow"] =

"0 6px 23px " +

primary\_dark\_hsla +

", inset 0px 12px 12px hsla(100, 100%, 100%, 0.27)";

var secondary\_dark =

"hsl(" +

document.getElementById("secondary\_hue").value +

"," +

(document.getElementById("secondary\_saturation").value - 35) +

"%," +

(document.getElementById("secondary\_luminocity").value - 5) +

"%)";

var secondary\_dark\_hsla =

"hsla(" +

document.getElementById("secondary\_hue").value +

"," +

(document.getElementById("secondary\_saturation").value - 35) +

"%," +

(document.getElementById("secondary\_luminocity").value - 10) +

"%, 0.5)";

document.getElementById("secondary\_color").style.background =

"linear-gradient(180deg, " + secondary + " 0%, " + secondary\_dark + " 100%)";

document.getElementById("secondary\_color").style["boxShadow"] =

"0 6px 23px " +

secondary\_dark\_hsla +

", inset 0px 12px 12px hsla(100, 100%, 100%, 0.27)";

function encodeImageFileAsURL() {

var filesSelected = document.getElementById("uploadFile").files;

let upload = document.getElementById("uploadFile");

if (filesSelected.length > 0) {

var fileToLoad = filesSelected[0];

var fileReader = new FileReader();

fileReader.readAsDataURL(fileToLoad);

fileReader.onload = function (fileLoadedEvent) {

srcData = fileLoadedEvent.target.result;

let filename = upload.files[0].name;

image.size = upload.files[0].size / 1000000;

console.log(filename + image.size); // <--- data: base64

Duotone();

};

}

}

function Duotone() {

let canvas = document.getElementById("duotone");

let ctx = canvas.getContext("2d");

let downloadedImg = new Image();

downloadedImg.onload = function () {

if (downloadedImg.height > downloadedImg.width) {

document.getElementById("duotone").style.height = "600px";

document.getElementById("duotone").style.width = "auto";

} else {

document.getElementById("duotone").style.height = "auto";

document.getElementById("duotone").style.width = "700px";

}

canvas.width = downloadedImg.width;

canvas.height = downloadedImg.height;

ctx.drawImage(downloadedImg, 0, 0, canvas.width, canvas.height);

imageData = ctx.getImageData(0, 0, canvas.width, canvas.height);

const pixels = imageData.data;

for (let i = 0; i < pixels.length; i += 4) {

const red = pixels[i];

const green = pixels[i + 1];

const blue = pixels[i + 2];

// Using relative luminance to convert to grayscale

const avg =

//(Math.max(red, green, blue) + Math.min(red, green, blue)) / 2;

//(red + green + blue) / 3;

Math.round((0.2126 \* red + 0.7152 \* green + 0.0722 \* blue) \* 1);

//const avg = Math.round((0.299 \* red + 0.587 \* green + 0.114 \* blue) \* 1);

pixels[i] = avg;

pixels[i + 1] = avg;

pixels[i + 2] = avg;

}

ctx.putImageData(imageData, 0, 0);

ctx.globalCompositeOperation = primary\_blend;

ctx.fillStyle = primary; // colour for highlights

ctx.fillRect(0, 0, canvas.width, canvas.height);

ctx.globalCompositeOperation = secondary\_blend;

ctx.fillStyle = secondary; // colour for shadows

ctx.fillRect(0, 0, canvas.width, canvas.height);

};

if (loaded) {

document.getElementById("drag-upload-wrap").style.display = "none";

document.getElementById("canvas\_container").className =

"duotone-animation canvas\_container";

document.getElementById("sidenav").className = "sidenav-animation sidenav";

document.getElementById("container").className =

"container-animation container";

document.getElementById("container").style.zIndex = "2000";

loaded = false;

}

downloadedImg.src = srcData;

}

function changeLight() {

primary = document.getElementById("colorLight").value;

Duotone();

}

function changeDark() {

secondary = document.getElementById("colorDark").value;

Duotone();

}

function preset1() {

primary = "#FFB23F";

secondary = "#FF6B00";

Duotone();

}

function preset2() {

primary = "#1cff86";

secondary = "#52136e";

document.getElementById("colorLight").value = "#1cff86";

secondary = document.getElementById("colorDark").value = "#52136e";

Duotone();

}

function primarycolorchange(h, s, l) {

primary = "hsl(" + h + "," + s + "%," + l + "%)";

primary\_dark = "hsl(" + h + "," + (s - 35) + "%," + (l - 5) + "%)";

primary\_dark\_hsla = "hsla(" + h + "," + (s - 35) + "%," + (l - 5) + "%, 0.5)";

document.getElementById("primary\_saturation").style.background =

"linear-gradient(90deg, #808080 0%, hsl(" +

document.getElementById("primary\_hue").value +

",100%,50%) 100%)";

document.getElementById("primary\_luminocity").style.background =

"linear-gradient(90deg, #000000 0%, hsl(" +

document.getElementById("primary\_hue").value +

",100%,50%) 50%, #FFFFFF 100%)";

document.getElementById("primary\_color").style.background =

"linear-gradient(180deg, " + primary + " 0%, " + primary\_dark + " 100%)";

document.getElementById("primary\_color").style["boxShadow"] =

"0 6px 23px " +

primary\_dark\_hsla +

", inset 0px 12px 12px hsla(100, 100%, 100%, 0.27)";

Duotone();

}

function secondarycolorchange(h, s, l) {

secondary = "hsl(" + h + "," + s + "%," + l + "%)";

secondary\_dark = "hsl(" + h + "," + (s - 35) + "%," + (l - 5) + "%)";

secondary\_dark\_hsla =

"hsla(" + h + "," + (s - 35) + "%," + (l - 5) + "%, 0.5)";

document.getElementById("secondary\_saturation").style.background =

"linear-gradient(90deg, #808080 0%, hsl(" +

document.getElementById("secondary\_hue").value +

",100%,50%) 100%)";

document.getElementById("secondary\_luminocity").style.background =

"linear-gradient(90deg, #000000 0%, hsl(" +

document.getElementById("secondary\_hue").value +

",100%,50%) 50%, #FFFFFF 100%)";

document.getElementById("secondary\_color").style.background =

"linear-gradient(180deg, " +

secondary +

" 0%, " +

secondary\_dark +

" 100%)";

document.getElementById("secondary\_color").style["boxShadow"] =

"0 6px 23px " +

secondary\_dark\_hsla +

", inset 0px 12px 12px hsla(100, 100%, 100%, 0.27)";

Duotone();

}

function primary\_inputfield\_sync() {

let input\_hue = document.getElementById("input\_primary\_hue").value;

let input\_saturation = document.getElementById(

"input\_primary\_saturation"

).value;

let input\_luminocity = document.getElementById(

"input\_primary\_luminocity"

).value;

if (input\_hue == "") {

input\_hue = "0";

} else if (input\_hue > 360) {

input\_hue = "360";

document.getElementById("primary\_hue").value = "360";

} else {

document.getElementById("primary\_hue").value = input\_hue;

}

if (input\_saturation == "") {

input\_saturation = "0";

} else if (input\_saturation > 100) {

input\_saturation = "100";

document.getElementById("primary\_saturation").value = "100";

} else {

document.getElementById("primary\_saturation").value = input\_saturation;

}

if (input\_luminocity == "") {

input\_luminocity = "0";

} else if (input\_luminocity > 100) {

input\_luminocity = "100";

document.getElementById("primary\_luminocity").value = "100";

} else {

document.getElementById("primary\_luminocity").value = input\_luminocity;

}

primarycolorchange(

document.getElementById("primary\_hue").value,

document.getElementById("primary\_saturation").value,

document.getElementById("primary\_luminocity").value

);

}

function secondary\_inputfield\_sync() {

let input\_hue = document.getElementById("input\_secondary\_hue").value;

let input\_saturation = document.getElementById(

"input\_secondary\_saturation"

).value;

let input\_luminocity = document.getElementById(

"input\_secondary\_luminocity"

).value;

if (input\_hue == "") {

input\_hue = "0";

} else if (input\_hue > 360) {

input\_hue = "360";

document.getElementById("secondary\_hue").value = "360";

} else {

document.getElementById("secondary\_hue").value = input\_hue;

}

if (input\_saturation == "") {

input\_saturation = "0";

} else if (input\_saturation > 100) {

input\_saturation = "100";

document.getElementById("secondary\_saturation").value = "100";

} else {

document.getElementById("secondary\_saturation").value = input\_saturation;

}

if (input\_luminocity == "") {

input\_luminocity = "0";

} else if (input\_luminocity > 100) {

input\_luminocity = "100";

document.getElementById("secondary\_luminocity").value = "100";

} else {

document.getElementById("secondary\_luminocity").value = input\_luminocity;

}

secondarycolorchange(

document.getElementById("secondary\_hue").value,

document.getElementById("secondary\_saturation").value,

document.getElementById("secondary\_luminocity").value

);

}

function primary\_range\_sync() {

document.getElementById("input\_primary\_hue").value =

document.getElementById("primary\_hue").value;

document.getElementById("input\_primary\_saturation").value =

document.getElementById("primary\_saturation").value;

document.getElementById("input\_primary\_luminocity").value =

document.getElementById("primary\_luminocity").value;

primarycolorchange(

document.getElementById("primary\_hue").value,

document.getElementById("primary\_saturation").value,

document.getElementById("primary\_luminocity").value

);

}

function secondary\_range\_sync() {

document.getElementById("input\_secondary\_hue").value =

document.getElementById("secondary\_hue").value;

document.getElementById("input\_secondary\_saturation").value =

document.getElementById("secondary\_saturation").value;

document.getElementById("input\_secondary\_luminocity").value =

document.getElementById("secondary\_luminocity").value;

secondarycolorchange(

document.getElementById("secondary\_hue").value,

document.getElementById("secondary\_saturation").value,

document.getElementById("secondary\_luminocity").value

);

}

/\* Download \*/

function downloadImage() {

console.log("button click");

image = dataURItoBlob(

document.querySelector("#duotone").toDataURL("image/jpg")

);

compressImage(image);

console.log(image);

console.log(result);

}

function dataURItoBlob(dataURI) {

var binary = atob(dataURI.split(",")[1]);

var array = [];

for (var i = 0; i < binary.length; i++) {

array.push(binary.charCodeAt(i));

}

return new Blob([new Uint8Array(array)], { type: "image/jpeg" });

}

function downloadFinal() {

console.log("final");

var link = document.createElement("a");

link.download = "filename.jpg";

link.href = result;

let { original, source } = image;

console.log("ready");

link.click();

}

function compressImage(image) {

console.log("start compress");

const files = [image];

compress

.compress(files, {

size: image.size,

maxWidth: 1920,

maxHeight: 1920,

resize: false,

})

.then((images) => {

console.log("im here");

const img = images[0];

console.log("yeah");

result = img.prefix + img.data;

console.log("finish compress");

console.log(result);

console.log(img.prefix);

console.log(image);

console.log(result);

downloadFinal();

});

}

function setstyle(id, primary\_style, secondary\_style) {

primary\_blend = primary\_style;

secondary\_blend = secondary\_style;

Duotone();

}

2. index.html

<html>

<head>

<meta http-equiv="Content-Type" content="text/html;charset=utf-8" />

<title>UTone</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<!-- Intro -->

<div id="logocontainer">

<p class="logo">UTone</p>

</div>

<!-- Input -->

<div id="drag-upload-wrap" class="drag-upload-wrap">

<div class="drag-upload-container">

<p class="upload-text-major">Завантажте зображення</p>

<p class="upload-text-minor">JPG або PNG</p>

<div class="drag-area">

<div id="drag-box-border"class="drag-box-border">

</div>

<div id="drag-box"class="drag-box">

<div class="drag-area-content">

<div class="drag-area-upload-icon">

</div>

<div class="drag-area-text"><span class="white-bold-text">Відкрийте</span> або перетягніть у виділену область

</div>

<input type="file" onChange="encodeImageFileAsURL()" id="uploadFile" accept="image/x-png,image/jpeg" ondragover="drag()" ondragleave="dragleave()" />

</div>

</div>

</div>

</div>

</div>

<!-- Main -->

<div id="sidenav" class="sidenav">

<div class="title">

<p>UTone</p>

</div>

<div class="preset\_container">

<p class="system\_text">ПРЕСЕТИ</p>

<div id="preset\_group\_1" class="preset\_group" >

<div class="preset\_name">

<p class="preset\_text">OVERDRIVE</p>

</div>

<div id="preset-1" class="preset">

</div>

<div id="preset-2" class="preset">

</div>

<div id="preset-3" class="preset">

</div>

<div id="preset-4" class="preset">

</div>

</div>

<div id="preset\_group\_2" class="preset\_group">

<div class="preset\_name" >

<p class="preset\_text">MONOCHROME</p>

</div>

<div id="preset-1" class="preset">

</div>

<div id="preset-2" class="preset">

</div>

<div id="preset-3" class="preset">

</div>

<div id="preset-4" class="preset">

</div>

</div>

<div id="preset\_group\_3" class="preset\_group">

<div class="preset\_name">

<p class="preset\_text">GRAVITY</p>

</div>

<div id="preset-1" class="preset">

</div>

<div id="preset-2" class="preset">

</div>

<div id="preset-3" class="preset">

</div>

<div id="preset-4" class="preset">

</div>

</div>

<div id="preset\_group\_4" class="preset\_group">

<div class="preset\_name">

<p class="preset\_text">MATRIX</p>

</div>

<div id="preset-1" class="preset">

</div>

<div id="preset-2" class="preset">

</div>

<div id="preset-3" class="preset">

</div>

<div id="preset-4" class="preset">

</div>

</div>

<div id="preset\_group\_5" class="preset\_group">

<div class="preset\_name">

<p class="preset\_text">ULTRANEON</p>

</div>

<div id="preset-1" class="preset">

</div>

<div id="preset-2" class="preset">

</div>

<div id="preset-3" class="preset">

</div>

<div id="preset-4" class="preset">

</div>

</div>

</div>

</div>

<!-- Panel -->

<div id="main\_container">

<div id="container" class="container">

<div class="style-container">

<div class="style-wrapper">

<div class="style-text">Стиль:</div>

<div class="style-label">

<span class="style-label-text">OVERDRIVE</span>

<div class="arrow">

</div>

</div>

<div id="style-menu-container">

<div id="style-menu">

<p class="style-menu-text" onclick="setstyle('overdrive', 'soft-light', 'screen')">OVERDRIVE</p>

<p class="style-menu-text" onclick="setstyle('monochrome', 'screen', 'color-dodge')">MONOCHROME</p>

<p class="style-menu-text" onclick="setstyle('gravity', 'multiply', 'screen')">GRAVITY</p>

<p class="style-menu-text" onclick="setstyle('matrix', 'multiply', 'lighten')">MATRIX</p>

<p class="style-menu-text" onclick="setstyle('ultraneon', 'color-dodge', 'darken')">ULTRANEON</p>

</div>

</div>

</div>

</div>

<div class="palette-container">

<div id="palette">

<div id="colors\_container">

<div id="primary\_color" class="color">

</div>

<div id="primary\_color\_dropdown" class="color\_dropdown">

<div id="dropdown">

<div class="range">

<p class="range-text">ВІДТІНОК</p>

<input id="primary\_hue" class="hue input\_range" type="range" min="0" max="360" step="1" value="35" oninput="primary\_range\_sync()">

<input type="text" id="input\_primary\_hue" onkeyup="primary\_inputfield\_sync()" value="40" maxlength="3" >

</div>

<div class="range">

<p class="range-text">НАСИЧЕНІСТЬ</p>

<input id="primary\_saturation" class="s input\_range" type="range" min="0" max="100" step="1" value="97" oninput="primary\_range\_sync()">

<input type="text" id="input\_primary\_saturation" onkeyup="primary\_inputfield\_sync()" value="100" maxlength="3" >

</div>

<div class="range">

<p class="range-text">ОСВІТЛЕНІСТЬ</p>

<input id="primary\_luminocity" class="l input\_range" type="range" min="0" max="100" step="1" value="70" oninput="primary\_range\_sync()">

<input type="text" id="input\_primary\_luminocity" onkeyup="primary\_inputfield\_sync()" onchange="input\_fill()" value="50" maxlength="3" >

</div>

</div>

</div>

<div id="secondary\_color" class="color">

</div>

<div id="secondary\_color\_dropdown" class="color\_dropdown">

<div id="dropdown">

<div class="range">

<p class="range-text">ВІДТІНОК</p>

<input id="secondary\_hue" class="hue input\_range" type="range" min="0" max="360" step="1" value="230" oninput="secondary\_range\_sync()">

<input type="text" id="input\_secondary\_hue" onkeyup="secondary\_inputfield\_sync()" value="40" maxlength="3">

</div>

<div class="range">

<p class="range-text">НАСИЧЕНІСТЬ</p>

<input id="secondary\_saturation" class="s input\_range" type="range" min="0" max="100" step="1" value="100" oninput="secondary\_range\_sync()">

<input type="text" id="input\_secondary\_saturation" onkeyup="secondary\_inputfield\_sync()" value="40" maxlength="3">

</div>

<div class="range">

<p class="range-text">ОСВІТЛЕНІСТЬ</p>

<input id="secondary\_luminocity" class="l input\_range" type="range" min="0" max="100" step="1" value="80" oninput="secondary\_range\_sync()">

<input type="text" id="input\_secondary\_luminocity" onkeyup="secondary\_inputfield\_sync()" value="40" maxlength="3">

</div>

</div>

</div>

</div>

</div>

</div>

<div class="button1-container" onclick="document.getElementById('inputFileToLoad').click();">

<div class="upload-button">

<div class="upload-icon">

</div>

<div class="button-text">Відкрити

</div>

</div>

</div>

<div class="button2-container" onclick="downloadImage()">

<div class="download-button">

<div class="download-icon">

</div>

<div class="button-text">Завантажити

</div>

</div>

</div>

</div>

<div id="canvas\_container" class="canvas\_container">

<canvas id="duotone" class="duotone"></canvas>

</div>

</div>

<script>

setTimeout(cleanlogo,4000);

function drag(){

document.getElementById('drag-box').className = "dragover drag-box";

}

function dragleave(){

document.getElementById('drag-box').className = "drag-box";

}

function cleanlogo(){

document.getElementById("logocontainer").style.display = "none";

}

function drop(){

document.getElementById('drag-upload-wrap').style.display = "none";

document.getElementById('sidenav').className = "sidenav-animation sidenav";

document.getElementById('container').className = "container-animation container";

}

</script>

</script>

<script type="text/javascript" src='compress.js'></script>

<script type="text/javascript" src="Duotone.js"></script>

</body>

</html>

3. style.css:

html,

body {

height: 100%;

width: 100%;

margin: 0;

background-color: #211338;

font-family: Montserrat;

font-style: normal;

-webkit-font-smoothing: antialiased;

overflow-y: hidden;

}

/\* animations \*/

@keyframes fadeOut {

from {

opacity: 1;

}

to {

opacity: 0;

}

}

@keyframes fadeInOut {

0% {

display: inline-block;

opacity: 0;

transform: translate3d(0, 60%, 0);

}

33% {

opacity: 1;

transform: translate3d(0, 0, 0);

}

66% {

opacity: 1;

transform: translate3d(0, 0, 0);

}

100% {

opacity: 0;

transform: translate3d(0, -60%, 0);

}

}

@keyframes inputfadeIn {

0% {

transform: translate3d(0, 5%, 0);

opacity: 0;

}

100% {

transform: translate3d(0, 0, 0);

opacity: 1;

}

}

@keyframes slideIn {

0% {

transform: translate3d(-100%, 0, 0);

opacity: 0;

}

100% {

transform: translate3d(0, 0, 0);

opacity: 1;

}

}

@keyframes slideInDown {

0% {

transform: translate3d(0, -100%, 0);

opacity: 0;

}

100% {

transform: translate3d(0, 0, 0);

opacity: 1;

}

}

/\* main container \*/

p {

margin: 0;

}

#main\_container {

height: 100%;

margin-left: 350px;

z-index: 100;

}

.container {

width: 100%;

opacity: 0;

height: 250px;

display: none;

position: relative;

z-index: 1000;

}

.container-animation {

animation: slideInDown;

animation-duration: 1s;

display: flex;

justify-content: center;

align-items: center;

-webkit-animation-fill-mode: forwards;

}

/\* style dropdown \*/

.style-container {

width: auto;

height: 20px;

margin-right: auto;

margin-left: 4vw;

}

.style-wrapper {

width: 250px;

height: 30px;

display: inline-block;

}

.style-text {

display: inline-block;

margin-right: 6px;

font-weight: 500;

color: #a3a3a3;

}

.style-label {

display: inline-block;

}

.style-label-text {

display: inline-block;

font-weight: 700;

letter-spacing: 0.04em;

font-size: 21px;

color: #ffffff;

}

#style-menu {

margin-top: 10px;

transform: none;

padding: 20px;

background-color: #150c24;

color: #fff;

text-align: center;

border-radius: 30px;

padding-top: 10;

animation-name: learn;

animation-duration: 0.8s;

}

#style-menu-container {

display: none;

position: absolute;

margin-left: 60px;

padding-top: 10px;

width: auto;

}

.style-label:hover + #style-menu-container {

display: block;

}

#style-menu-container:hover {

display: block;

}

.arrow {

position: relative;

top: -2px;

width: 9px;

height: 6px;

background: url(arrow.svg) no-repeat;

display: inline-block;

margin-left: 0px;

}

.style-menu-text {

margin-top: 12px;

font-weight: 600;

opacity: 70%;

font-size: 16;

cursor: pointer;

}

.style-menu-text:hover {

opacity: 100%;

}

/\* palette \*/

.palette-container {

position: absolute;

height: 140px;

width: 270px;

margin-right: auto;

margin-left: auto;

}

#palette {

width: 380px;

display: flex;

justify-content: center;

width: auto;

}

#colors\_container {

height: 140px;

width: 270px;

background: #150c24;

border-radius: 80px;

display: flex;

justify-content: center;

align-items: center;

}

.color {

width: 92px;

height: 92px;

border-radius: 50%;

margin: 10px;

}

#secondary\_color {

background: linear-gradient(180deg, #fee69c 0%, #fdc968 100%);

box-shadow: 0px 6.22303px 23.3364px rgba(254, 201, 106, 0.34),

inset 0px 12.4461px 12.4461px rgba(255, 255, 255, 0.27);

}

#primary\_color {

background: linear-gradient(180deg, #9cacfe 0%, #cd68fd 100%);

box-shadow: 0px 6px 23px hsla(100, 100%, 100%, 1),

inset 0px 12px 12px hsla(100, 100%, 100%, 0.27);

}

/\* color dropdown \*/

.color\_dropdown {

position: absolute;

margin-top: 355px;

padding-top: 30px;

min-width: 160px;

animation-name: learn;

animation-duration: 0.8s;

}

#primary\_color\_dropdown {

display: none;

margin-right: 115px;

}

#secondary\_color\_dropdown {

display: none;

margin-left: 115px;

}

#primary\_color:hover + #primary\_color\_dropdown {

display: block;

}

#secondary\_color:hover + #secondary\_color\_dropdown {

display: block;

}

#secondary\_color\_dropdown:hover {

display: block;

}

#primary\_color\_dropdown:hover {

display: block;

}

#dropdown {

background-color: #150c24;

border-radius: 30px;

/\*box-shadow: 0px 25px 25px rgba(223, 219, 233, 0.55); \*/

z-index: 1;

margin-top: 30px;

padding: 10px 20px 20px 20px;

width: 205px;

}

#dropdown::after {

content: "";

position: absolute;

top: 40px;

left: 110px;

border-width: 10px;

border-style: solid;

border-color: transparent transparent #150c24 transparent;

}

/\* dropdown info \*/

.range {

margin-top: 16px;

}

.range-text {

font-size: 14px;

font-weight: 600;

margin-bottom: 0px;

color: #a19ea7;

}

input[type="text"],

input[type="number"] {

-webkit-appearance: none;

height: 25px;

border: none;

background: #211338;

border-radius: 3px;

width: 40px;

text-align: center;

font-family: Montserrat;

font-style: normal;

font-weight: 600;

font-size: 14px;

color: #fff;

margin-left: 5px;

}

input[type="text"]:focus,

input[type="number"]:focus {

outline: 0;

background: #2d184f;

}

input[type="number"]::-webkit-inner-spin-button,

input[type="number"]::-webkit-outer-spin-button {

opacity: 1;

}

input[type="range"] {

-webkit-appearance: none;

height: 8px;

border-radius: 10px;

width: 150px;

}

input[type="range"]::-webkit-slider-thumb {

-webkit-appearance: none;

appearance: none;

width: 22px;

height: 22px;

border-radius: 50%;

border: 4px solid #ffffff;

box-sizing: border-box;

filter: drop-shadow(0px 2.54655px 2.54655px rgba(0, 0, 0, 0.25));

cursor: pointer;

}

input[type="range"]:focus {

outline: 0;

}

.hue {

background: linear-gradient(

90deg,

#ff0000 0%,

#ffe600 21.64%,

#00ffb2 48.2%,

#1400ff 72.16%,

#ff0000 100%

);

}

.s {

background: linear-gradient(90deg, #808080 0%, #ffa800 100%);

}

.l {

background: linear-gradient(90deg, #000000 0%, #ffa800 50%, #ffffff 100%);

}

@keyframes learn {

from {

opacity: 0;

transform: 0, -100%, 0;

}

to {

opacity: 1;

transform: 0, 0, 0;

}

}

/\* buttons \*/

.button1-container {

margin-left: auto;

margin-right: 4vw;

width: auto;

height: auto;

text-align: center;

cursor: pointer;

}

.button2-container {

margin-right: 4vw;

width: auto;

height: auto;

text-align: center;

cursor: pointer;

}

.button-text {

margin-top: 6px;

color: #76dad0;

text-transform: uppercase;

font-size: 16px;

font-weight: 600;

}

.upload-icon {

display: block;

width: 33px;

height: 30px;

margin-left: auto;

margin-right: auto;

background: url(upload\_icon.svg) no-repeat;

background-position: center;

background-size: contain;

}

.download-icon {

display: block;

width: 33px;

height: 30px;

margin-left: auto;

margin-right: auto;

background: url(download\_icon.svg) no-repeat;

background-position: center;

background-size: contain;

}

#inputFileToLoad {

display: none;

}

/\* canvas \*/

.duotone {

width: 500px;

height: auto;

margin-top: 0px;

z-index: 0;

position: relative;

}

.duotone-animation {

animation: inputfadeIn;

animation-duration: 1s;

position: relative;

-webkit-animation-fill-mode: forwards;

}

.canvas\_container {

position: relative;

display: flex;

justify-content: center;

opacity: 0;

}

/\* resulution <1350 \*/

@media (max-width: 1350px) {

.download-icon {

width: 45px;

height: 40px;

}

.upload-icon {

width: 45px;

height: 40px;

}

.button-text {

display: none;

}

.style-wrapper {

width: 200px;

}

}

/\* intro \*/

#logocontainer {

position: absolute;

width: 100%;

height: 100%;

display: flex;

justify-content: center;

align-items: center;

background: #150c24;

animation: fadeOut;

animation-duration: 1s;

animation-delay: 2s;

-webkit-animation-fill-mode: forwards;

z-index: 101;

}

.logo {

opacity: 0%;

font-family: Montserrat;

font-style: normal;

font-weight: 600;

font-size: 72px;

color: #fff;

animation-name: fadeInOut;

animation-duration: 2s;

-webkit-animation-fill-mode: forwards;

}

/\*photo input\*/

.drag-box-border {

/\*display:none;\*/

border-radius: 10px;

width: 500px;

height: 350px;

margin: 0 auto;

text-align: center;

display: inline-block;

background: rgba(118, 218, 208, 0.1);

box-sizing: border-box;

border-radius: 10px;

overflow: hidden;

position: relative;

text-align: center;

}

.drag-box-border:before {

content: "";

position: absolute;

border-radius: 10px;

border: 10px dashed #76dad0;

box-sizing: border-box;

top: -4px;

bottom: -4px;

left: -4px;

right: -4px;

}

input[type="file"] {

position: absolute;

height: 100%;

width: 100%;

opacity: 0;

top: 0;

left: 0;

}

.drag-box {

position: relative;

top: -348px;

left: 2px;

width: 496px;

height: 346px;

border-radius: 10px;

background: linear-gradient(

0deg,

rgba(118, 218, 208, 0.1),

rgba(118, 218, 208, 0.1)

),

#211338;

z-index: 100;

display: flex;

justify-content: center;

}

.drag-upload-wrap {

height: 100%;

width: 100%;

display: flex;

justify-content: center;

align-items: center;

background-color: #211338;

z-index: 100;

}

.drag-upload-container {

opacity: 0;

text-align: center;

animation: inputfadeIn;

animation-duration: 1s;

animation-delay: 2.5s;

-webkit-animation-fill-mode: forwards;

}

.drag-area {

overflow: hidden;

height: 360px;

width: 500px;

margin-top: 20px;

}

.upload-text-major {

margin: 0;

font-weight: 500;

font-size: 24px;

line-height: 29px;

color: #ffffff;

}

.upload-text-minor {

margin: 0;

margin-top: 15px;

font-weight: 500;

font-size: 16px;

line-height: 20px;

color: #a6a1af;

}

.drag-area-upload-icon {

margin: 0;

display: block;

width: 80px;

height: 70px;

background: url(upload-icon.svg);

margin-left: auto;

margin-right: auto;

margin-top: 100px;

}

.drag-area-text {

margin-top: 20px;

font-weight: 500;

font-size: 16px;

line-height: 20px;

color: #76dad0;

}

.white-bold-text {

font-weight: 600;

color: #fff;

padding-bottom: 2px;

border-bottom: 2px solid #fff;

z-index: 1000;

}

.drag-area-content {

height: auto;

width: auto;

}

.dragover {

background: linear-gradient(

0deg,

rgba(118, 218, 208, 0.2),

rgba(118, 218, 208, 0.2)

),

#211338;

}

/\* sidenav \*/

.sidenav {

opacity: 0;

height: 100%;

min-height: 100%;

width: 350px;

position: fixed;

top: 0;

left: 0;

overflow-x: hidden;

padding-top: 20px;

background-size: cover;

background: #150c24;

}

.sidenav-animation {

animation: slideIn;

animation-duration: 1s;

-webkit-animation-fill-mode: forwards;

}

.title {

margin-top: 85px;

text-align: center;

color: #fff;

font-family: Montserrat;

font-style: normal;

font-weight: 600;

font-size: 36px;

line-height: 44px;

}

/\* presets \*/

.preset\_container {

width: 350px;

height: auto;

text-align: center;

}

.preset\_group {

width: 270px;

opacity: 0.7;

height: 65px;

overflow-y: hidden;

background-color: rgba(242, 67, 28, 0.7);

background-image: url(pic.png);

background: linear-gradient(

to right,

rgba(242, 67, 28, 0.7),

rgba(242, 67, 28, 0.7)

),

url(pic.png);

border-radius: 10px;

text-align: center;

margin-top: 16px;

margin-left: 40px;

-webkit-transition: height 0.7s;

}

#preset\_group\_1 {

background: url(pic.png);

}

#preset\_group\_2 {

background: url(pic2.png);

}

#preset\_group\_3 {

background: url(pic3.png);

}

#preset\_group\_4 {

background: url(pic4.png);

}

#preset\_group\_5 {

background: url(pic5.png);

}

@keyframes open {

from {

height: 65px;

}

to {

height: 150px;

}

}

.preset\_group:hover {

height: 150px;

opacity: 1;

-webkit-transition: height 0.7s;

transition-delay: 0.1s;

opacity: 1;

}

.preset {

display: inline-block;

width: 40px;

height: 40px;

border-radius: 50%;

background: linear-gradient(129.84deg, #28ffe5 6.04%, #6f84f7 91.37%);

border: 5px solid #ffffff;

margin: 5px;

}

.preset:hover {

box-shadow: 0px 6.10428px 22.8911px rgba(255, 255, 255, 0.43),

inset 0px 12.2086px 12.2086px rgba(255, 255, 255, 0.27);

}

#preset-1 {

background: linear-gradient(129.84deg, #ffc271 6.04%, #ff606d 91.37%);

}

#preset-2 {

background: linear-gradient(129.84deg, #fb5c7e 6.04%, #6f84f7 91.37%);

}

#preset-4 {

background: linear-gradient(129.84deg, #fb5c5c 6.04%, #ba1313 91.37%);

}

.preset\_text {

margin: 0;

font-family: Montserrat;

font-style: normal;

font-weight: 500;

font-size: 21px;

line-height: 26px;

letter-spacing: 0.16em;

text-align: left;

margin-left: 30px;

opacity: 100%;

color: #ffffff;

}

.preset\_name {

margin-top: 23px;

margin-bottom: 25px;

opacity: 100%;

}

.system\_text {

text-align: left;

font-family: Montserrat;

font-style: normal;

font-weight: 600;

font-size: 16px;

line-height: 20px;

text-transform: uppercase;

color: #a19ea7;

margin-left: 40px;

margin-top: 60px;}