COMSATS University Islamabad,
Abbottabad Campus

A Generative AI Web Tool Using Neural Style Transfer for Fashion Image Generation

By

Haleema Bint-e-Jamil	**CIIT/FA20-BSE-005/ATD**
Marwa Zulfiqar	**CIIT/FA20-BSE-006/ATD**
Tooba Syed	**CIIT/FA20-BCS-130/ATD**

Supervisor

Dr. Mazhar Ali

Bachelor of Science in Computer Science (2020-2024)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.

COMSATS University, Islamabad Pakistan

A Generative AI Web Tool Using Neural Style Transfer for Fashion Image Generation

A project presented to

COMSATS Institute of Information Technology, Islamabad

In partial fulfillment

of the requirement for the degree of

Bachelor of Science in Computer Science (2020-2024)

By

Haleema Bint-e-Jamil	**CIIT/FA20-BSE-005/ATD**
Marwa Zulfiqar	**CIIT/FA20-BSE-006/ATD**
Tooba Syed	**CIIT/FA20-BCS-130/ATD**

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied the report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Haleema Binte Jamil

Marwa Zulfiqar

Tooba Syed

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS/SE) "A Generative AI Web Tool Using Neural Style Transfer for Fashion Image Generation" was developed by **HALEEMA BINTE JAMIL (CIIT/FA20-BSE/TN-005)**, **MARWA ZULFIQAR (CIIT/FA20-BSE/TN-006)** and **TOOBA SYED (CIIT/FA20-BCS/TN-130)** under the supervision of "DR. MAZHAR ALI" that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

Supervisor

External Examiner

Head of Department

(Department of Computer Science)

EXECUTIVE SUMMARY

In the ever-evolving world of fashion, the demand for innovative and unique designs is incessant. Traditional fashion design processes are often labor-intensive, time-consuming, and fraught with subjectivity and creativity blocks. To address these challenges, we present the "AI Style Bot: Where Artistry Meets Innovation," a Generative AI Web Tool utilizing Neural Style Transfer (NST) to revolutionize the fashion design process. This tool aims to provide fashion designers with a powerful, automated solution to create unique, visually captivating designs by merging various elements such as textures, colors, patterns, and accessories from different images.

The AI Style Bot significantly enhances the efficiency of fashion design by reducing the time and effort required to generate new designs. Designers can quickly produce a plethora of unique, aesthetically pleasing designs, thereby eliminating the iterative and often tedious traditional design process. By integrating advanced AI techniques, the tool not only speeds up the design process but also inspires creativity, offering designers an extensive range of styles, colors, and patterns to experiment with, fostering innovation in the fashion industry.

Our project aligns with the core modules of a software engineering degree, encompassing key areas such as Software Project Management, Software Engineering Concepts, Artificial Intelligence, and Human-Computer Interaction. By leveraging the knowledge gained from these modules, we have meticulously designed and developed a user-friendly, reliable, and efficient web-based application. The procedural approach adopted ensures simplicity, ease of maintenance, and resource efficiency, facilitating quick prototyping and seamless integration with third-party libraries.

The AI Style Bot not only caters to the needs of individual designers but also offers fashion brands and businesses the ability to create distinct styles that enhance brand recognition and customer loyalty. By automating the design process and providing a vast repository of creative ideas, the AI Style Bot stands as a transformative tool in the fashion industry, driving innovation and setting new standards in fashion design.

ACKNOWLEDGEMENT

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Mazhar Ali”. Without his personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to him for his encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Haleema Binte Jamil

Marwa Zulfiqar

Tooba Syed

ABBREVIATIONS

SRS	Software Require Specification
PC	Personal Computer
SDA	Software Design and Architecture
DFD	Data Flow Diagram
AI	Artificial Intelligence
SDLC	Software Development Life Cycle
HCI	Human Computer Interaction
UI	User Interface
SPM	Software Project Management
SEC	Software Engineering Concepts
SRE	Software Requirement Engineering
SQE	Software Quality Engineering
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
JS	Java Script
NST	Neural Style Transfer

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

As the world is progressing, we also need to change the way we think and adapt new techniques and methods to solve modern world problems. This tool, which we are going to build using neural networks and machine learning algorithms, will help users to get cool and modern ideas for designs.

1.1. Brief

In this project, we are using advanced AI tools like Neural Style Transfer (NST) to create completely new fashion designs. These technologies allow us to combine different elements from images in unique ways. For example, NST helps us take textures from one image and mix them seamlessly with the colors from another.

This approach is not just about mixing things together—it is about using technology to make fashion design more creative and efficient. Each design that comes out of this process is not just a blend of parts; it is a new creation that brings together different elements in a fresh and exciting way. By using NST, we are aiming to push the boundaries of what is possible in fashion, offering designs that are both cutting-edge and attractive to a wide range of people.

1.2. Relevance to Course Modules

"A Generative AI Web Tool Using Neural Style Transfer for Fashion Image Generation" is highly relevant to the course modules we studied in a software engineering degree. It is widely used in software development and can greatly enhance the efficiency and productivity. The course include are:

1.1.1. Software Project Management

This course provides the foundational knowledge and skills necessary to develop Generative AI Web Based Tool. By adhering to the principles e.g., project scope and objectives, requirement analysis, project planning, task division, team condition, risk management, quality assurance and testing, security and privacy, deployment and maintenance, the project team can ensure a well-organized, efficient, and successful development process that meets user requirements while addressing potential challenges and risks. SPM also helped in time management of project and tasks. We learned that how team will meet the user requirements and support effective project Management.

1.1.2. Software Engineering Concepts

The software development life cycle SDLC is a key aspect of software engineering Concepts. Knowledge of this process essential in developing a Web Based Software that is usable, reliable, and meets user requirements. We studied different SDLC stages in this course and help to choose the best model. We applied Agile model in our project to facilitate iterative development, enhance flexibility, and allow for continuous user feedback.

1.1.3. Software Requirement Engineering

SRE emphasizes the importance of gathering and analysing user requirements. From this course we applied the concepts of understanding of the needs and requirements of "Generative AI Web Tool Using Neural Style Transfer" in our project. SRE helped in making the requirements document that is SRS which involves gathering and documenting user requirements.

1.1.4. Web Technologies

To develop "Generative AI Web Tool Using Neural Style Transfer", it is important to have knowledge of relevant software development tools and technologies. We studied different tools and techniques in this course, and we used HTML, CSS, JS, PHP for making our project website.

1.1.5. Artificial Intelligence

Our course of Artificial Intelligence gave us a base and grip of python and different models to get started with development of "Generative AI Web Tool Using Neural Style Transfer", along with learning the use of different libraries and frameworks to create advanced AI solutions.

1.1.6. Software Quality Engineering

From this course we learned how to apply SQE concepts to emphasize quality assurance which are helpful for building our project high-quality that meets user requirements. This helped to make our web reliable, user-friendly, and efficient.

1.1.7. Human Computer Interaction

From this course we understood the importance of UI, design and mistakes leading to unattractive interface. It helped us in building the best possible design and kept the interface between users and computer systems as friendly as it could. This course also helped us in building a UI which is more interactive with the clients.

1.1.8. Software Design and Architecture

From this course we learned methodologies for designing and architecting software. This course helped to our project by applying a well-designed and architected system that can visualize the workflow and track progress.

1.3. Project Background

Our software, called "AI Style Bot: where Artistry Meets Innovation," handles some important problems in the fashion industry.

Firstly, it will make the process of fashion designing easier and faster. Normally, creating new fashion designs takes a lot of time and effort. But with our system, which will use advanced technology called Neural Style Transfer (NST), designers can make unique, attractive, and user-friendly designs quickly. This means they do not have to go through lots of design changes, saving time and money.

Secondly, "AI Style Bot" is like a big source of creative ideas for fashion designers. In the creative world of fashion, it can be hard to come up with new and exciting designs all the time. Our tool will help designers by letting them try out many different styles, colors, and patterns. This can inspire them to create new and interesting designs that their customers will Love.

Lastly, our software helps fashion brands and businesses create their own special styles. This can make them stand out and be more recognized by their customers. When people recognize and like a brand's style, they are more likely to stay loyal to that brand.

1.4. Literature Review

Generative AI web tools using neural style transfer for fashion image generation are becoming increasingly prominent in the digital creative landscape. RunwayML, Picsart, Adobe Firefly, and Playform are notable platforms offering these capabilities. RunwayML employs various AI models. Picsart uses deep neural networks to apply artistic styles to user-uploaded images. Adobe Firefly incorporates a sophisticated generative AI model trained on extensive datasets to ensure high-quality style transfers. Playform provides automated style transfer tools designed for artists, using algorithms that transform image sets into consistent artistic series. The neural style transfer algorithm originally proposed by Gatys et al. (2016) forms the basis for many of these applications. Key scientists in this field include Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge, who pioneered the neural style transfer technique.

1.5. Analysis from Literature Review

The analysis of the literature reveals that generative AI tools utilizing neural style transfer are highly effective in fashion image generation, offering innovative solutions for designers and artists. These tools leverage deep learning algorithms to blend

content and style from different images, enabling the creation of unique and visually appealing outputs. The contributions of Gatys et al. have been instrumental in advancing this technology, providing a foundation that contemporary platforms like RunwayML, Picsart, Adobe Firefly, and Playform have built upon. The integration of these tools into creative workflows demonstrates their versatility and potential to revolutionize digital fashion design, making it more accessible and efficient.

1.6. Methodology and Software Lifecycle for This Project

The choice of design methodology and software process model plays a crucial role in the success of a software system. Design Methodology used in this software is: Procedural Approach.

1.6.1. Rationale behind Selected Methodology

Procedural Approach is a highly suitable design methodology for this project. It is a widely adopted approach in software development, particularly for projects driven by the need for simplicity, ease of maintenance and a straightforward workflow.

By opting for a procedural approach in the context of web development, our team aims to achieve the following advantages:

- Readability and Maintainability:
  The procedural paradigm promotes a clear and linear code structure, making it easier for developers to read, understand, and maintain the codebase. This is especially beneficial for projects with limited complexity.
- Ease Of Debugging:
  Debugging is simplified in a procedural model, as developers can trace the flow of execution step by step. This can expedite the identification and resolution of issues during the development process.
- Quick Prototyping:
  Procedural programming is often conducive to rapid prototyping. It allows for a straightforward implementation of features, facilitating the timely creation of functional prototypes to demonstrate and validate concepts.
- Resource Efficiency:
  In scenarios where the website's functionality does not require intricate object-oriented structures, a procedural approach can be more resource efficient. It allows for a leaner codebase without the overhead of unnecessary abstractions.

- Compatibility with Third-Party Libraries:

Procedural programming aligns well with many third-party libraries and frameworks commonly used in web development. This compatibility can streamline the integration of external tools and resources into our project.

2. Problem Definition

In the dynamic world of fashion, staying ahead requires continuous innovation and creativity. However, traditional methods of fashion design often led to significant challenges for designers. The process of conceptualizing, iterating, and finalizing designs is not only time-consuming but also prone to human errors and creative limitations. Designers face the daunting task of generating new ideas, translating them into drafts, and refining them repeatedly, which can stifle creativity and restrict design diversity. Moreover, the manual nature of design creation contributes to high costs and lengthy turnaround times, posing additional barriers to scalability and adaptability in an ever-changing market.

2.1. Problem Statement

The fashion industry relies heavily on the ingenuity and skills of designers to produce captivating and trend-setting designs. However, the conventional approach to fashion design is beset by several challenges. Firstly, the manual design process is labor-intensive and prone to human errors, leading to inefficiencies and delays. Secondly, designers often struggle with creative blocks and the repetitive nature of design creation, limiting the variety and novelty of designs produced. This reliance on manual processes not only hampers scalability but also imposes significant costs and time constraints on design firms.

Furthermore, the subjective nature of manual design can lead to inconsistencies in style and quality, further complicating efforts to maintain a cohesive brand identity or meet evolving consumer preferences. These challenges underscore the need for a transformative solution that enhances design innovation, reduces time-to-market, and mitigates the risks associated with manual design processes. By leveraging advanced technologies such as Generative AI and Neural Style Transfer, there is an opportunity to revolutionize fashion design by automating parts of the creative process, fostering greater design diversity, and empowering designers to explore new creative horizons without the constraints of traditional methods.

2.2. Deliverables and Development Requirements

Deliverables refer to the tangible outcomes or results that are produced during this project.

Following are the deliverables of our project are:

- The user will be able to register/signup.
- The user will be able to login.
- The user will be able to generate Designs.
- The user will be able to view the generated design.
- The user will be able to download the designs.
- The user will be able to customize the design to be generated.
- The user will be able to view their previously generated designs.
- The user will be able to upload reference images.
- The user will be able to select dimensions for images to be generated.

- The user will be able to select color and similarity ratio.

Following are the developments of A Generative AI Web Tool Using Neural Style Transfer for Fashion Image Generation:

- MongoDB
- React
- Python
- Neural Style Transfer
- Tensor Flow

3. Requirement Analysis

Requirement analysis involves a thorough understanding of the customer's requirements and translating them into actionable work items. Use case diagram, functional requirements, and non-functional requirements of "A Generative AI Web Tool Using Neural Style Transfer for Fashion Image Generation" included in requirement analysis chapter.

3.1. Use Cases Diagram(s)

Fig 3.1 illustrates the fully dressed use case diagram of "Generative AI Web Tool".

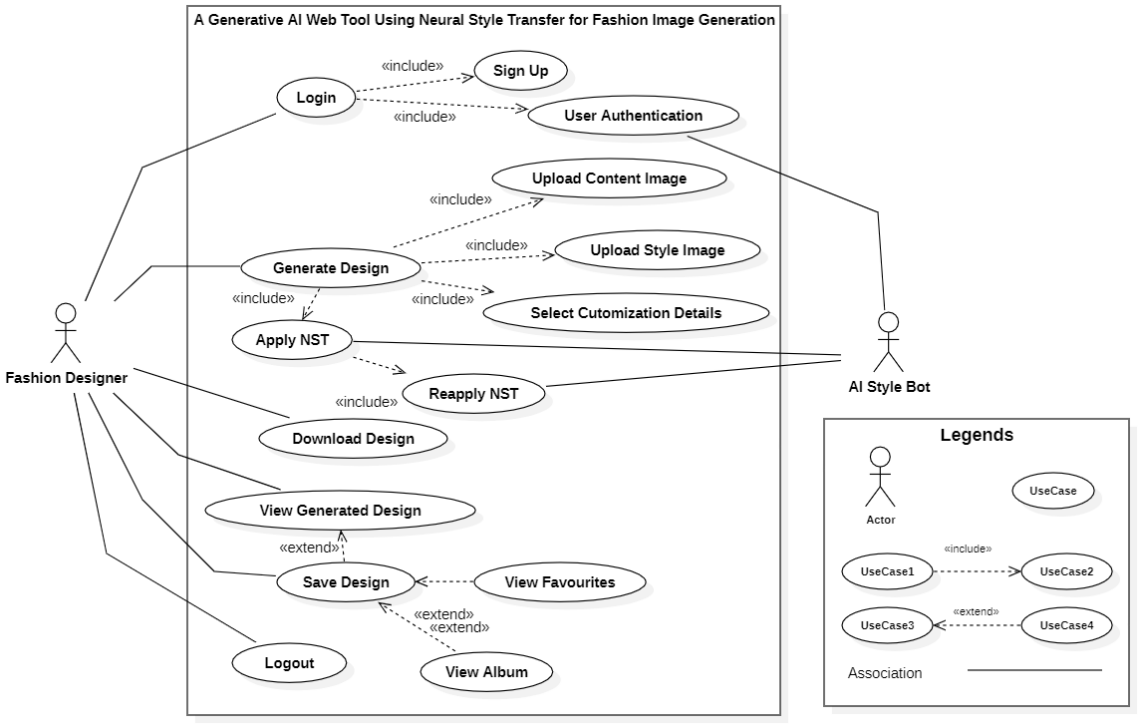

Figure 3.1: Use Case Diagram

3.2. Detailed Use Case

Detailed use cases of project provide a comprehensive description of how users interact with a software website to achieve specific goals. Following are the tables of fully dressed use cases of "Generative AI Web Tool Using NST for Fashion Image Generation".

3.2.1. UC-01 Sign Up

Table 3.1: UC-01 Sign Up

Use Case ID:	UC-01
Use Case Name:	Sign Up
Actors:	User
Description:	This use case describe how user will Sign Up to the system
Trigger:	The user presses the sign-up button to register because he wants to create the profile.
Preconditions:	PRE-1. The user is not registered with same email address already exist. PRE-2. A page is created to display user profile.
Post conditions:	POST-1. User will successfully be registered after providing first name, last name email and password.
Normal Flow:	1. User opens the website. 2. After pressing Sign up button, the user enters their first name, last name, new email and password. 3. Actor details are checked to figure out that the email has not been used before. 4. The new actor profile is created.
Special Requirements:	1. Actor has an active internet connection. 2. Actor must have a device to use this website.

3.2.2. UC-02 Login

After signup only authenticated users can login the website. Table 3.2 illustrates the fully dressed use case of login.

Table 3.2: UC-02 Login

Use Case ID:	UC-02
Use Case Name:	Login
Actors:	User
Description:	This use case describe how actor will login to website.
Trigger:	User wants to login, so he presses the login button.
Preconditions:	PRE-1. The user login web page is displayed to the user. PRE-2. The user is already registered.
Post conditions:	POST-1. Actor is successfully logged in to website.
Normal Flow:	1. User opens the website. 2. On the registration interface the Actor presses the login button. 3. Actor details are checked to determine that the user is already registered or not.
Special Requirements:	1. Actor has an active internet connection. 2. Actor must have a device to use this website. 3. Actor must register before login.

3.2.3. UC-03 Generate Fashion Designs

Users can generate designs after logging into the website. Table 3.3 illustrates the fully dressed use case of generate designs.

Table 3.3: UC-03 Generate Designs

Use Case ID:	UC-03
Use Case Name:	Generate Fashion Design
Actors:	User
Description:	This use case allows the Fashion Designer to use the AI Style Bot to generate a new and unique fashion design based on selected parameters and inspiration.
Trigger:	The user presses the sign-up button to register because he wants to create the profile.
Preconditions:	PRE-1. AI Style Bot is running. PRE-2. User is logged into web application i.e., AI Style Bot. PRE-3. The designer has uploaded or selected relevant images and design preferences.
Post conditions:	POST-1. The generated design is ready for further customization or export. POST-2. Uploaded images, generated design and customization specifications are stored in database.
Normal Flow:	1. User opens the website. 2. User Login/Signup in this application. 3. The Fashion Designer accesses the "Generate Fashion Design" feature. 4. The AI Style Bot processes the input data, including selected images and design preferences. 5. The system applies Neural Style Transfer (NST) and Generative Adversarial Network (GAN) techniques to generate a fashion design. 6. The AI Style Bot presents the generated design to the designer for review.
Special Requirements:	1. Actor has an active internet connection. 2. Actor must have a device to use this website.

3.2.4. UC-04 Download Design

Users can download the designs after generating. Table 3.4 illustrates the fully dressed use case of download design.

Table 3.4: UC-04 Download Design

Use Case ID:	UC-04
Use Case Name:	Download Design
Actors:	User
Description:	This use case enables the users to download the design they have already created using AI Style Bot.
Trigger:	The user presses the “Download” button.

Preconditions:	PRE-1. AI Style Bot is running. PRE-2. User is logged into web application i.e., AI Style Bot. PRE-3. System has a database of designs user has already generated.
Post conditions:	POST-1. The design file is successfully downloaded and is now available in the specified download location or directory. POST-2. The user is notified of the successful download, either through an on-screen confirmation message or a notification.
Normal Flow:	1. User opens the website. 2. User Login/Signup in this application. 3. The user accesses the "Designs" feature. 4. The AI Style Bot provides a user interface with all the generated designs. of current user.
Special Requirements:	The system has multiple download formats (e.g., JPEG, PNG, SVG, PDF) to accommodate different user preferences.

3.2.5. UC-05 View Generated Design

Table 3.5: UC-05 View Generated Design

Use Case ID:	UC-05
Use Case Name:	View Generated Design
Actors:	User
Description:	This use case enables the users to view the generated design they have already created using AI Style Bot.
Trigger:	The user presses the “Save” button on generated page.
Preconditions:	PRE-1. AI Style Bot is running. PRE-2. User is logged into web application i.e., AI Style Bot. PRE-3. The Design has been generated using AI Style Bot. PRE-3. System has a database of designs user has already generated.
Post conditions:	POST-1. The design is successfully displayed on the user interface, making it visible to the user. POST-2. User will be able to Zoom and Download the design.
Normal Flow:	1. User opens the website. 2. User Login/Signup in this application. 3. The user accesses the "Designs" feature. 4. The AI Style Bot provides a user interface with all the generated designs of the current user. 5.The user selects a specific design from the application interface expressing the intent to view it.
Special Requirements:	1. Actor has an active internet connection. 2. Actor must have a device to use this website.

3.2.6. UC-06 Logout

Table 3.6: UC-06 Logout

Use Case ID:	UC-06
Use Case Name:	Logout
Actors:	User
Description:	This use case enables the users to Logout from AI Style Bot.
Trigger:	The user presses the “Logout” button on navigation bar.
Preconditions:	PRE-1. AI Style Bot is running. PRE-2. User is logged into web application i.e., AI Style Bot. PRE-3. The user must be currently authenticated and logged into the system. Logout is applicable only if there is an active user session.
Post conditions:	POST-1. The user's session is successfully terminated, and the user is no longer considered authenticated within AI Style Bot.
Normal Flow:	1. User opens the website. 2. User Login/Signup in this application. 3. The user decides to log out and initiates the logout process. This could be triggered by clicking on a "Logout" button or selecting a logout option from the user interface.
Special Requirements:	1. Actor has an active internet connection. 2. Actor must have a device to use this website.

3.3. Functional Requirements

Functional requirements are focused on the functionality and characteristics of the project. Following are the functional requirements of this project:

FR-01: The user will be able to register/signup.

FR-02: The user will be able to login.

FR-03: The user will be able to generate Designs.

FR-04: The user will be able to view generated design.

FR-05: The user will be able to download the designs.

FR-06: The user will be able to customize the design.

FR-07: User will be able to view their previously generated designs.

FR-08: The user will be able to upload reference images.

FR-09: The user will be able to select dimensions for images.

FR-10: The user will be able to select color and check similarity ratio.

3.4. Non-Functional Requirements

User-Friendly

The website is going to be user friendly and not provide any complex UI (user interface).

Portability

The website will work on windows.

Security

It's a Generative toll and it provides security to user's information and designs. Application will be 99.9% reliable. System will have proper identification process by authenticating accounts i.e., user first register and their account verify through email.

Performance

Overall performance of the app will be good.

Usability

Interface will be friendly and easy to use. Every icon and buttons have proper labelling that will allow user to easily understand its functionalities, user can easily remember its interface. Allowing users to use the system to their specific styles, colors and design parameters will make the toll more user centric and will allow the users to do customization.

Responsiveness

Application will support multiple screen sizes for different devices.

4. Design and Architecture

The design and architecture of "A Generative AI Web Tool Using Neural Style Transfer for Fashion Image Generation" involves the overall structure, components, and interactions that enable the effective implementation and operation of this software. The diagram of system architecture, activity diagram and design model show the design and architecture of the project.

4.1. System Architecture

The system architecture comprises several components that work together to support the implementation and functioning of the Generative AI Web Tool. Fig 4.1 illustrates the System Architecture diagram of the project.

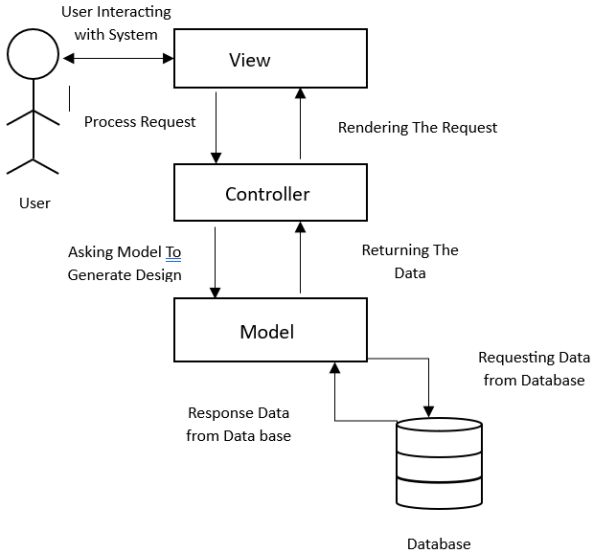

Figure 4.1: Architecture Diagram

4.2. Process Flow/Representation

In Generative AI Web Tool, it provides a platform for generating Fashion Designs from uploaded reference images and selected colour.

4.2.1. Activity Diagram

Activity diagram shows the workflow, identifying potential bottlenecks and communicating the process to users involved in project. Fig 4.2 illustrates the activity diagram of project for the users.

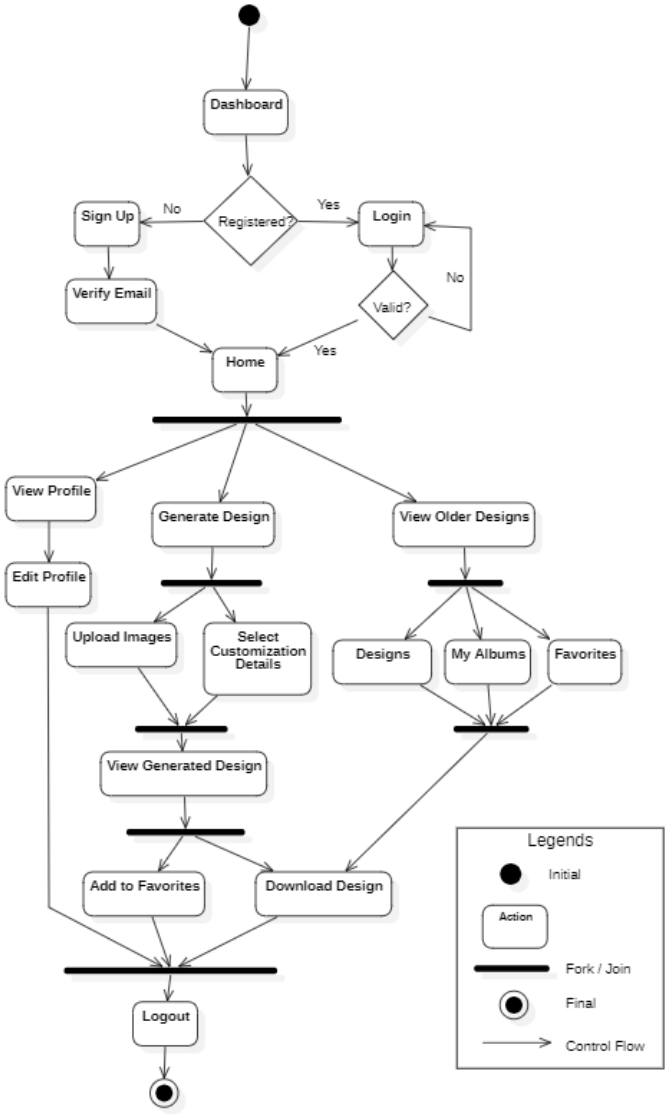

Figure 4.2: Activity Diagram

4.3. Design Models

4.3.1. Data Flow Diagram

4.3.1.1. Level 0

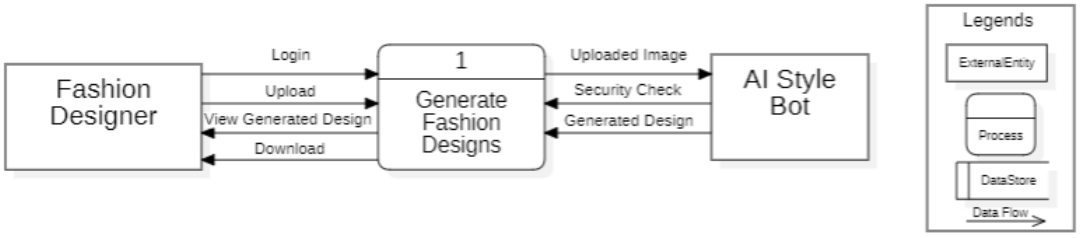

Figure 4.3: Level 0 DFD

4.3.1.2. Level 1

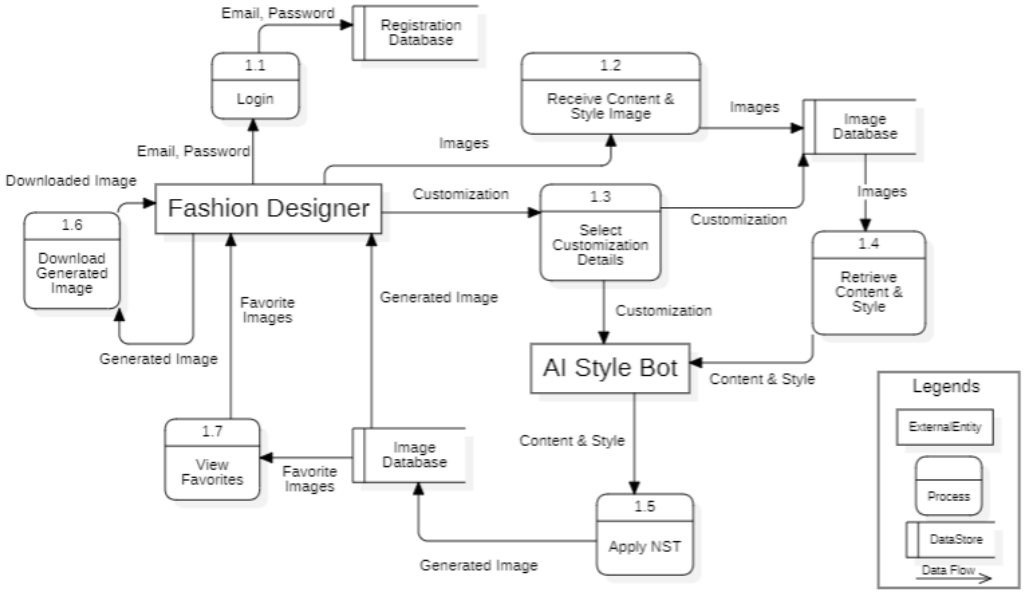

Figure 4.4: Level 1 DFD

4.3.2. Sequence Diagram

4.3.2.1. Sign Up

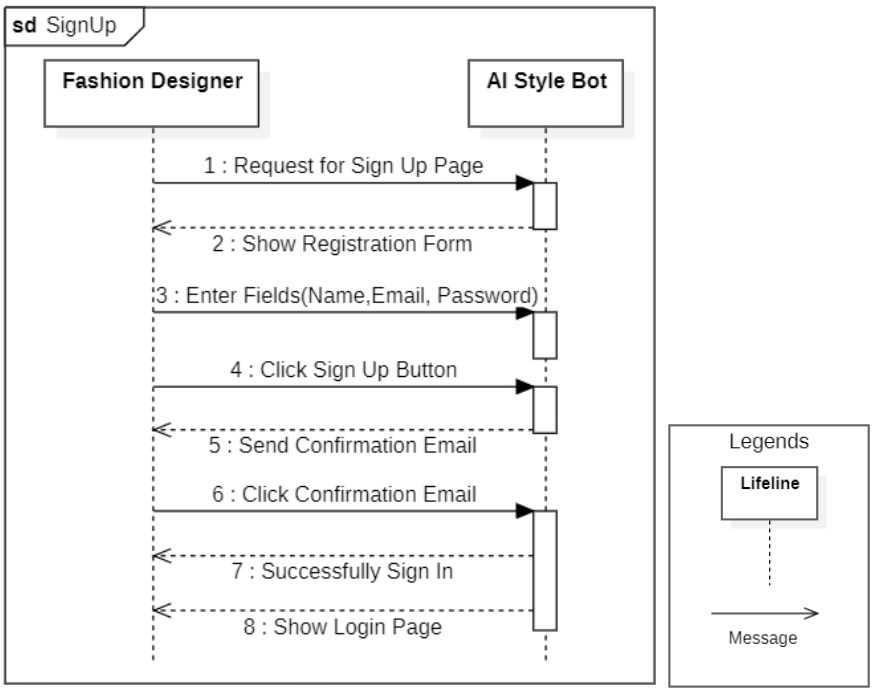

Figure 4.5: Sequence Diagram for Sign Up

4.3.2.2. Login

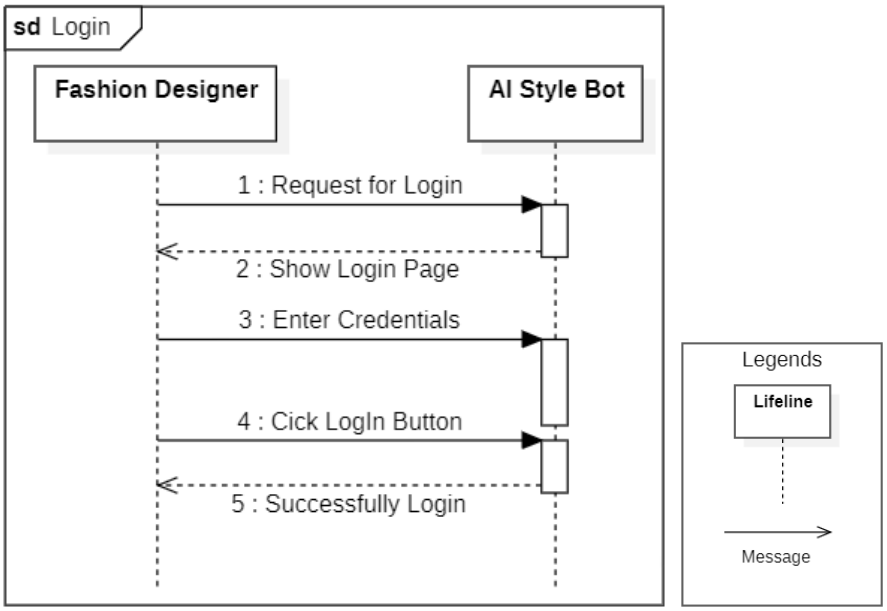

Figure 4.6: Sequence Diagram for Login

4.3.2.3. Generate Designs

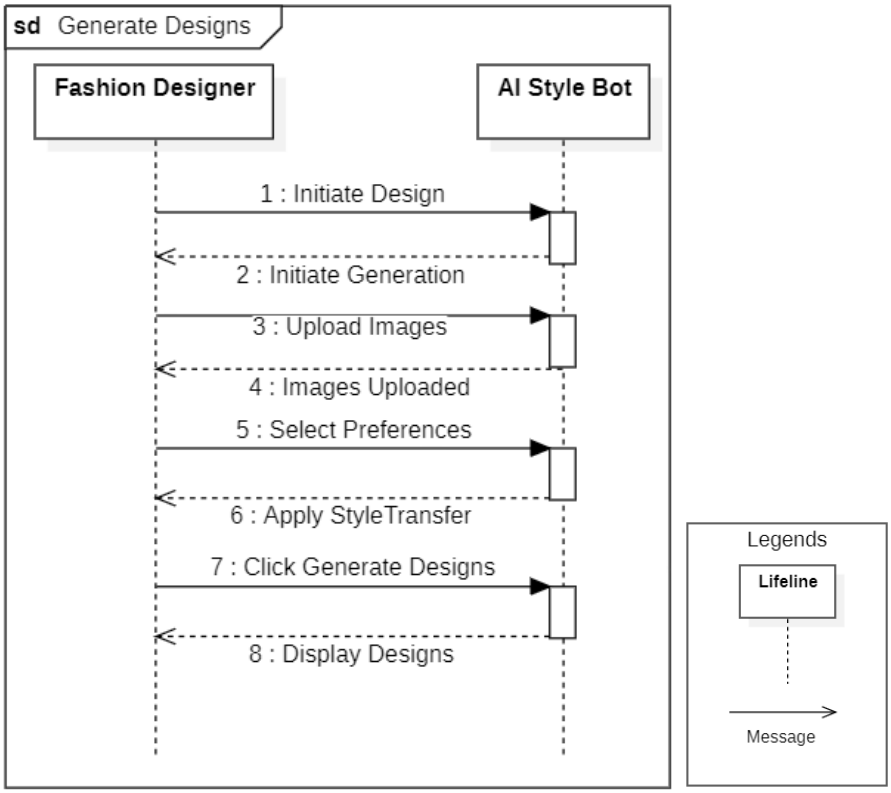

Figure 4.7: Sequence Diagram for Generate Design

4.3.2.4. View Design

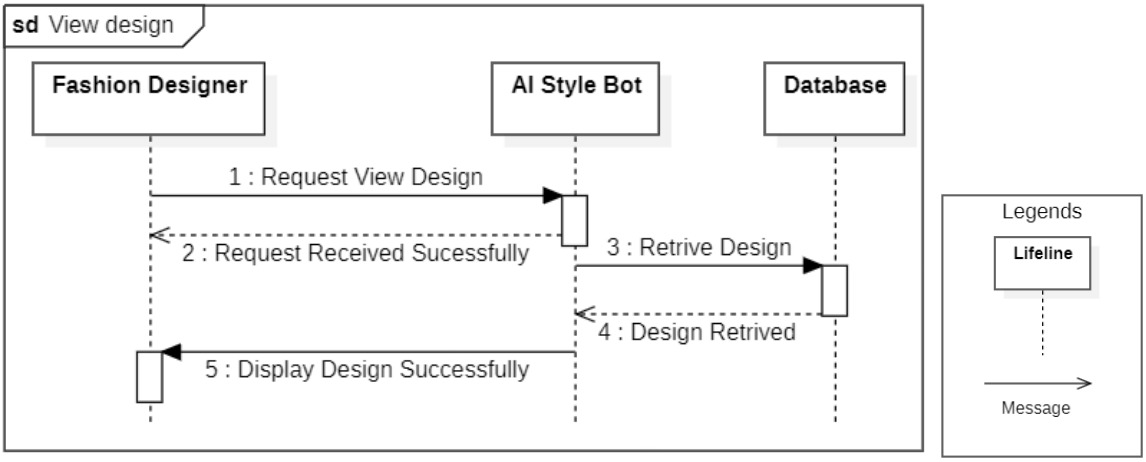

Figure 4.8: Sequence Diagram for View Design

4.3.2.5. Export Design

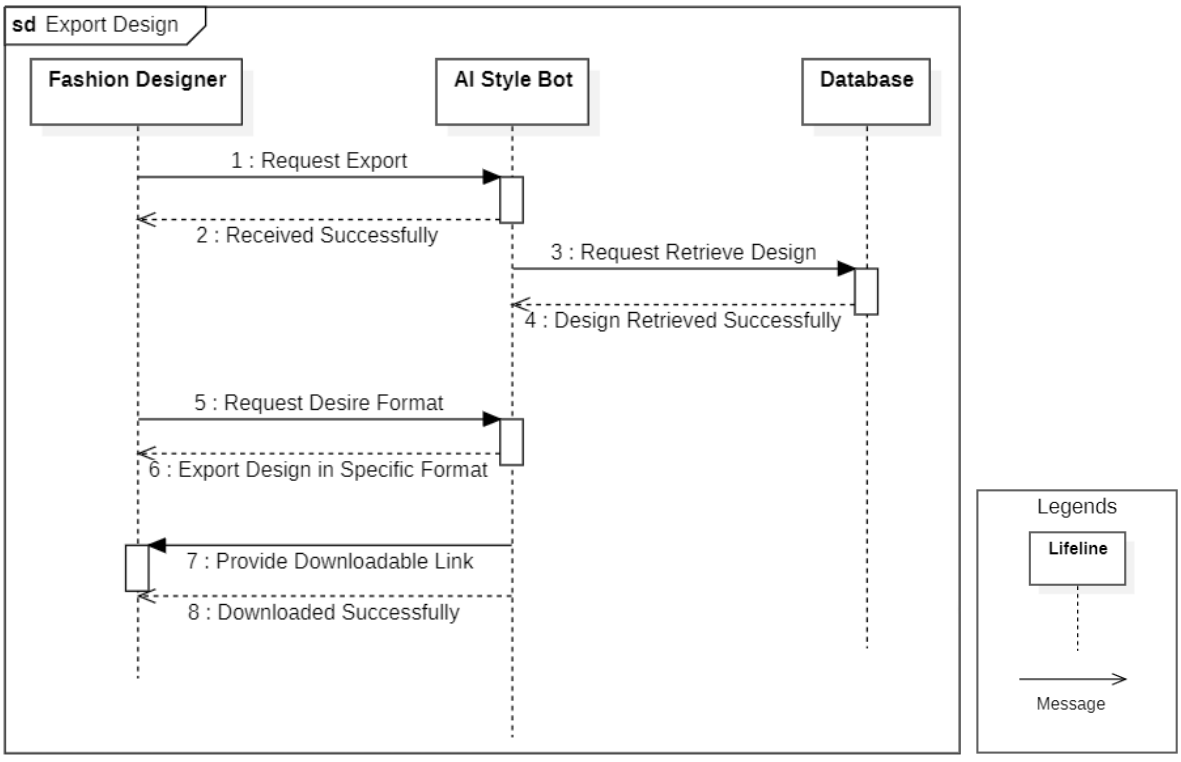

Figure 4.9: Sequence Diagram for Export Design

4.3.2.6. Logout

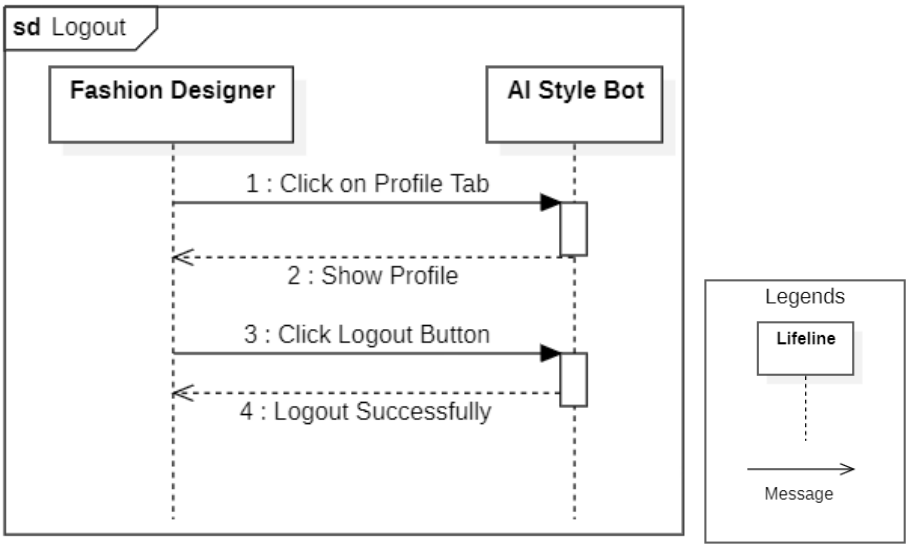

Figure 4.10: Sequence Diagram for Log Out

5. Implementation

The Implementation include the algorithms that used to build the project and user interfacethrough which the user interacts with the system.

5.1. Algorithm

```
import sys
import tensorflow as tf
import tensorflow_hub as hub
import random
import json
import numpy as np
from skimage.metrics import structural_similarity as ssim

// for image crop . to make center square
def crop_center(image):
    shape = image.shape
    new_shape = min(shape[1], shape[2])
    offset_y = max(shape[1] - shape[2], 0) // 2
    offset_x = max(shape[2] - shape[1], 0) // 2
    image = tf.image.crop_to_bounding_box(image, offset_y, offset_x, new_shape,
new_shape)
    return image
// for loading image and process image
def load_image(image_path, image_size=(256, 256), preserve_aspect_ratio=True):
    img = tf.io.decode_image(tf.io.read_file(image_path), channels=3,
dtype=tf.float32)[tf.newaxis, ...]
    img = crop_center(img)
    img = tf.image.resize(img, image_size, preserve_aspect_ratio=True)
    return img
// image flip or rotate etc
def random_transform(image):
    image = tf.image.random_flip_left_right(image)
    image = tf.image.random_flip_up_down(image)
    angles = [0, 90, 180, 270]
    angle = random.choice(angles)
    image = tf.image.rot90(image, k=angle // 90)
    return image

// generating a random number
def output_random_number(min_value=1, max_value=10000000):
    """
    Generates and returns a random number between min_value and max_value.
```

```
    Args:
    - min_value (int): The minimum value for the random number.
    - max_value (int): The maximum value for the random number.

    Returns:
    - int: A random number between min_value and max_value.
    """
    return random.randint(min_value, max_value)
// to apply colour on image
def apply_color_filter(image, color_hex):
    color_rgb = [int(color_hex[i:i+2], 16) / 255.0 for i in (1, 3, 5)]
    color_rgb = tf.constant(color_rgb, shape=[1, 1, 1, 3], dtype=tf.float32)
    color_mask = tf.ones_like(image) * color_rgb
    return image * color_mask

// applying model
def stylize_image(content_image, style_image, alpha, preserve_color=True,
color_hex=None):
    hub_handle = 'https://www.kaggle.com/models/google/arbitrary-image-stylization-
v1/TensorFlow1/256/2'
    hub_module = hub.load(hub_handle)
    stylized_image = hub_module(tf.constant(content_image),
tf.constant(style_image))[0]
    if preserve_color:
        stylized_image = tf.image.adjust_hue(stylized_image, 0.1)
    if color_hex:
        stylized_image = apply_color_filter(stylized_image, color_hex)
    return alpha * stylized_image + (1 - alpha) * content_image

// image similarity calculation
def calculate_similarity(image1, image2):
    image1_np = image1.numpy().squeeze()
    image2_np = image2.numpy().squeeze()
    similarity_score, _ = ssim(image1_np, image2_np, full=True, multichannel=True,
win_size=3, channel_axis=-1, data_range=image1_np.max() - image1_np.min())
    return similarity_score * 100  # Convert to percentage

if _name_ == "_main_":
    content_image_path = sys.argv[1]
    style_image_path = sys.argv[2]
    alpha = float(sys.argv[3])
    color_hex = sys.argv[4] if len(sys.argv) > 4 else None
    num_images = int(sys.argv[5]) if len(sys.argv) > 5 else 4
```

```
    content_image = load_image(content_image_path, (384, 384))
    style_image = load_image(style_image_path, (256, 256))
    style_image = tf.nn.avg_pool(style_image, ksize=[3, 3], strides=[1, 1],
padding='SAME')

    alphas = np.linspace(0.3, 0.9, num_images)  # Generate alpha values based on the
number of images
    output_images = []

    for i, alpha in enumerate(alphas):
        transformed_style_image = random_transform(style_image)
        stylized_image = stylize_image(content_image, transformed_style_image,
alpha, color_hex=color_hex)
        similarity_score = calculate_similarity(content_image, stylized_image)
        random_number = output_random_number()
        output_image_path = f'uploads/output_{i + random_number + 1}.png'
        tf.keras.preprocessing.image.save_img(output_image_path, stylized_image[0])
        output_images.append({'image_path': output_image_path, 'similarity':
similarity_score})

    print(json.dumps(output_images))
```

5.2. External APIs

APIs used in this project are described in table 5.1.

Table 5.1: Details of APIs used in the project.

Name of API	**Description of API**	**Purpose of usage**	**List down the function/class name in which it is used**
TensorFlow (tf)	Open-source platform for machine learning and deep learning	Reading images, resizing, transformations, saving images	crop_center, load_image, random_transform, apply_color_filter, stylize_image
TensorFlow Hub	Repository and library for reusable machine learning models	Loading pre-trained neural style transfer model	stylize_image
NumPy	Library for numerical computations in Python	Generating a range of alpha values	Main script execution (alphas = np.linspace(...))

Random	Python standard library for generating random numbers and choices	Random transformations on images, generating random numbers	random_transform, output_random_number
skimage.metrics	Library for image processing tasks, part of scikit-image	Calculating structural similarity index (SSIM)	calculate_similarity

5.3. User Interface

The user interface for our website consists of pages that represent the workflow and functions which the user can perform on our site. The User Interface for our site is designed to be user friendly. Users can easily see and access all features of the system.

5.3.1. Home Page

Fig 5.1 illustrates the homepage of our site.

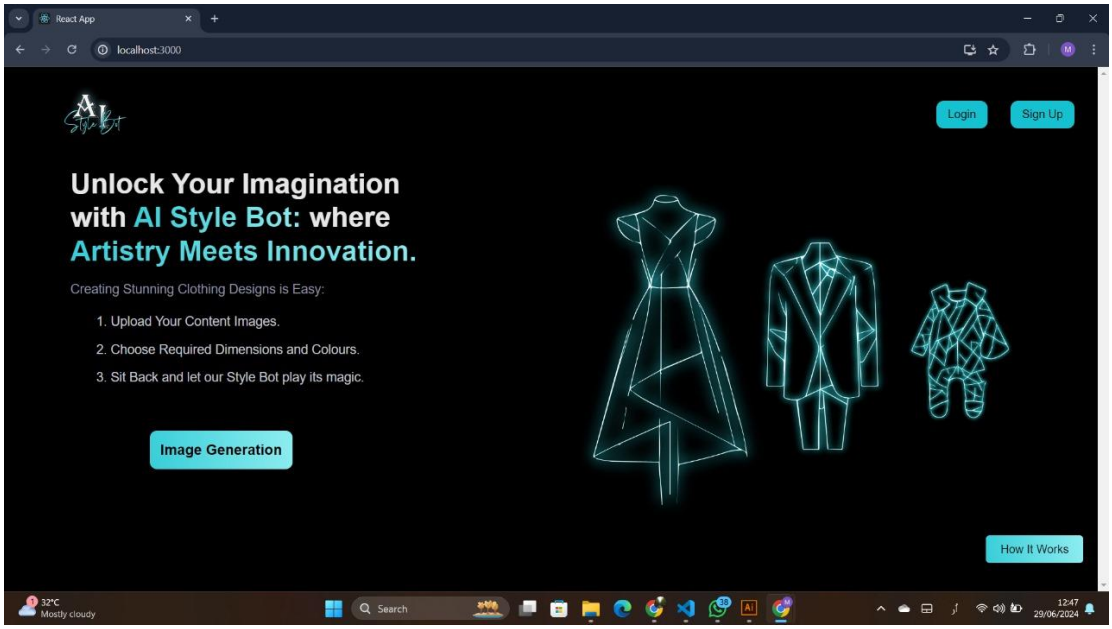

Figure 5.1: Home Page UI

5.3.2. Sign Up Page

Fig 5.2 illustrates the registration page of our site.

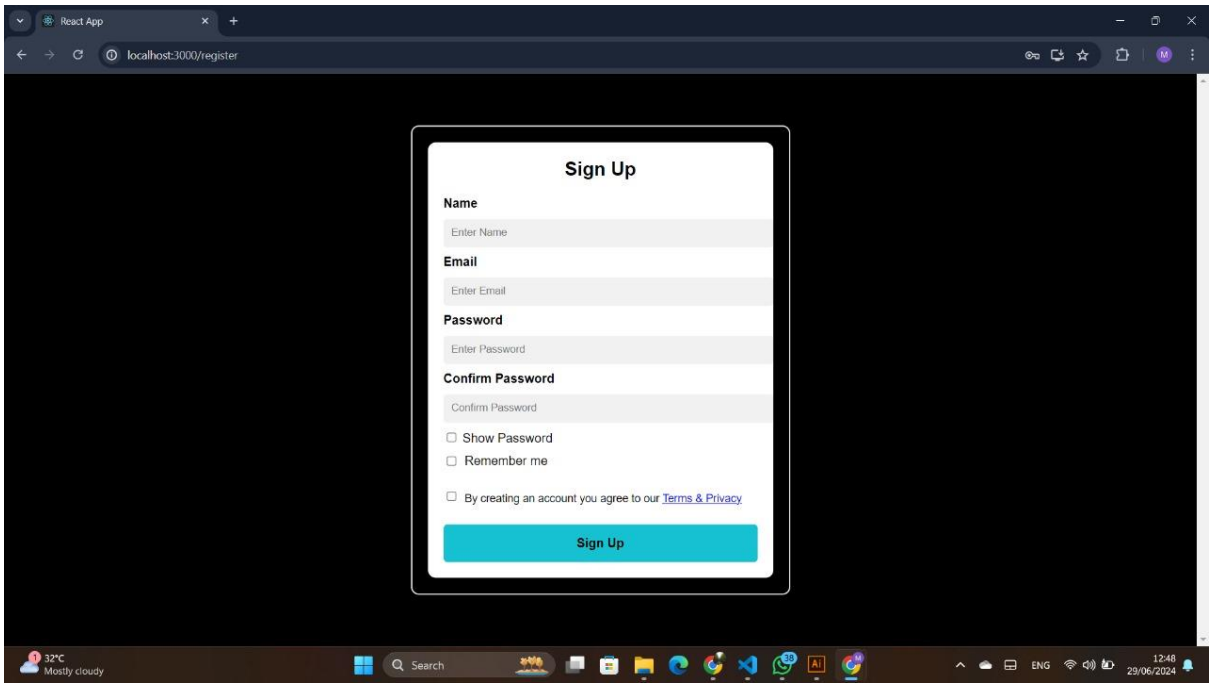

Figure 5.2: Sign Up Page UI

5.3.3. Login Page

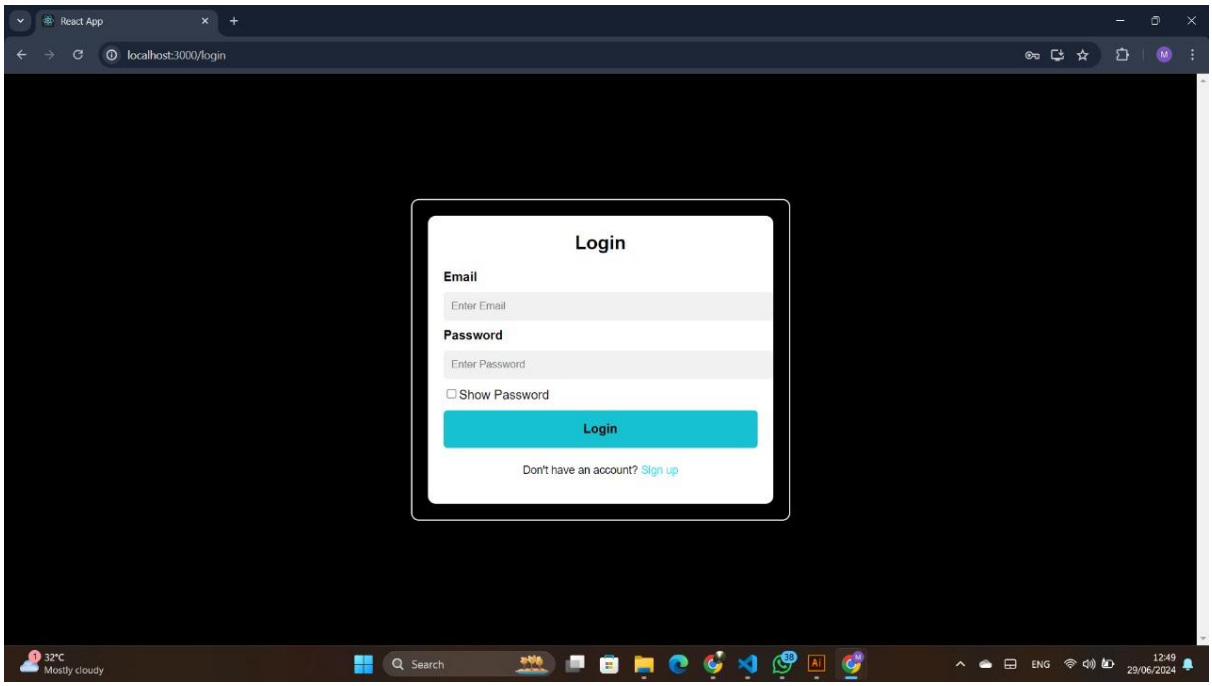

Figure 5.3: Login Page UI

5.3.4. Generate Page

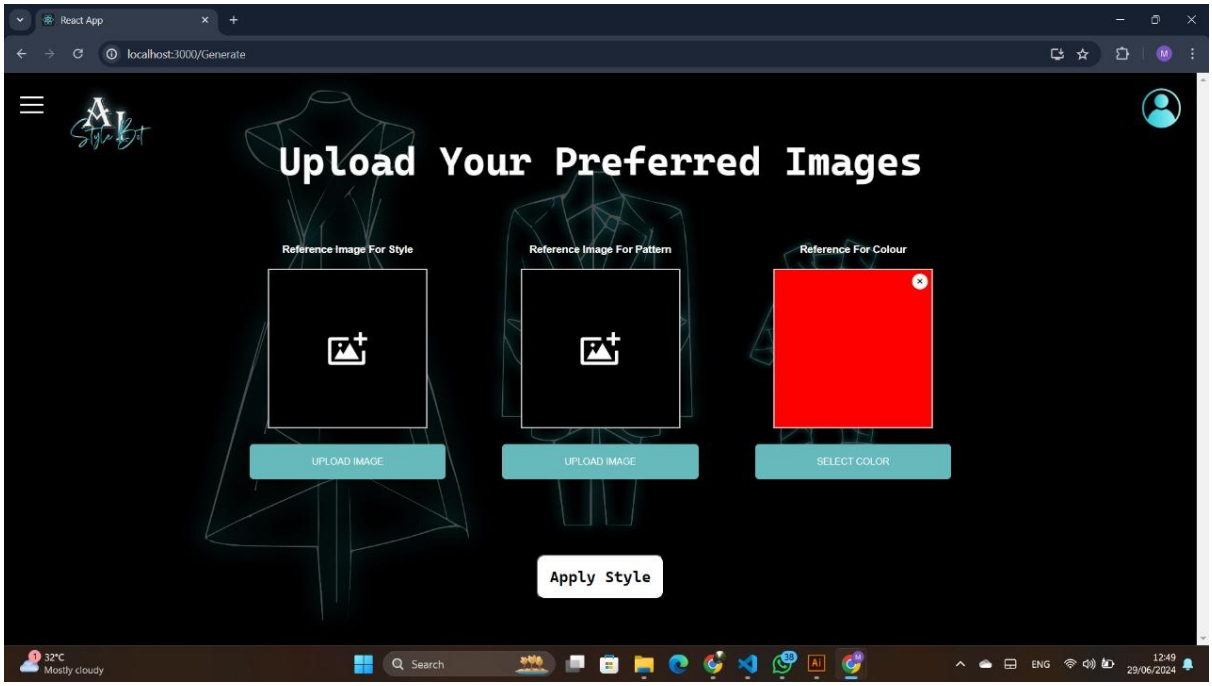

Figure 5.4: Generate Page UI

5.3.5. Loading Page

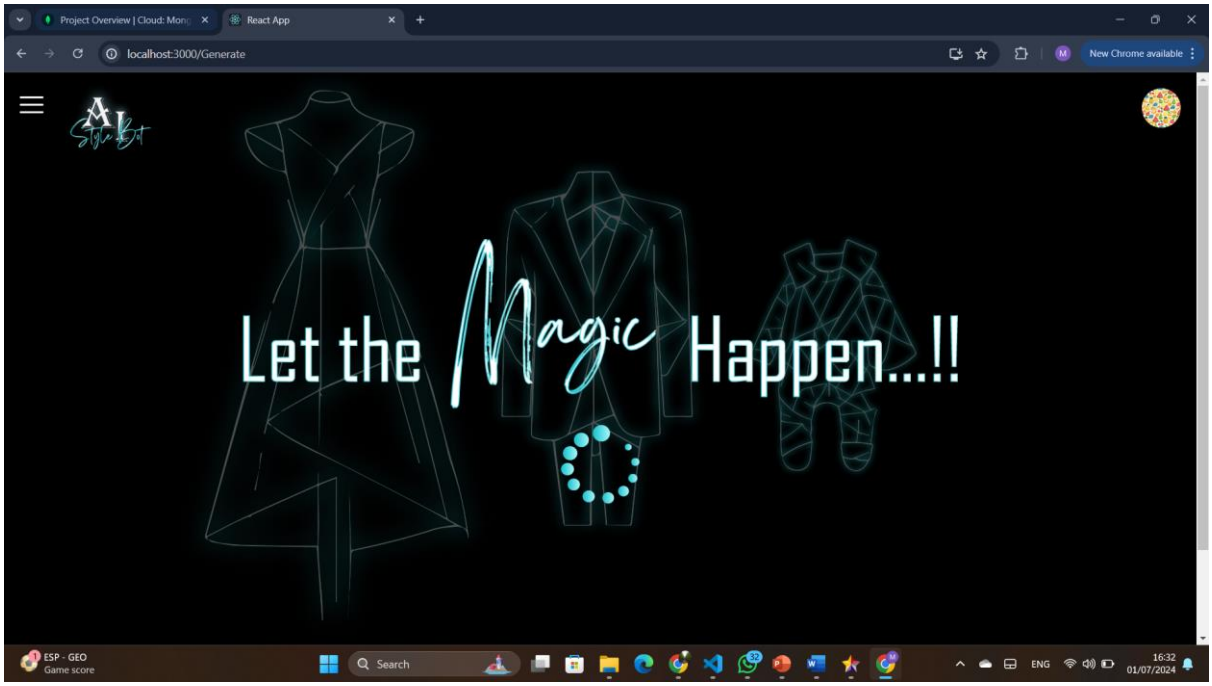

Figure 5.5: Loading Page UI

5.3.6. Output Page

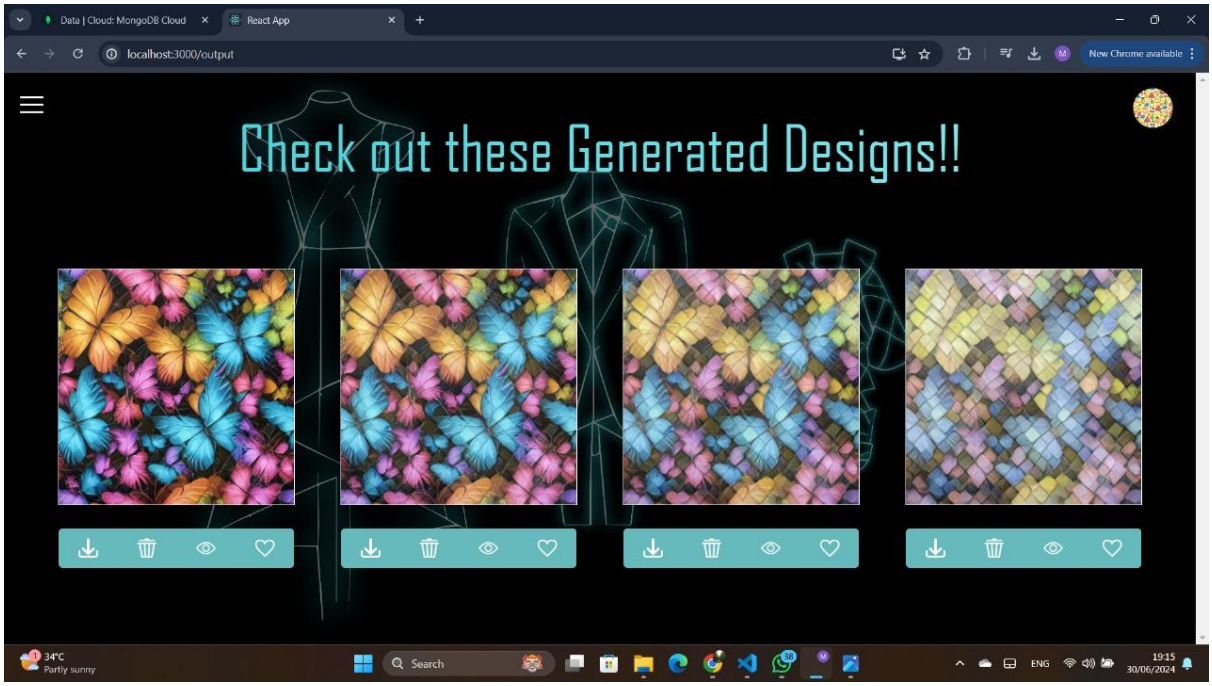

Figure 5.6: Output Page UI

5.3.7. My Designs Page

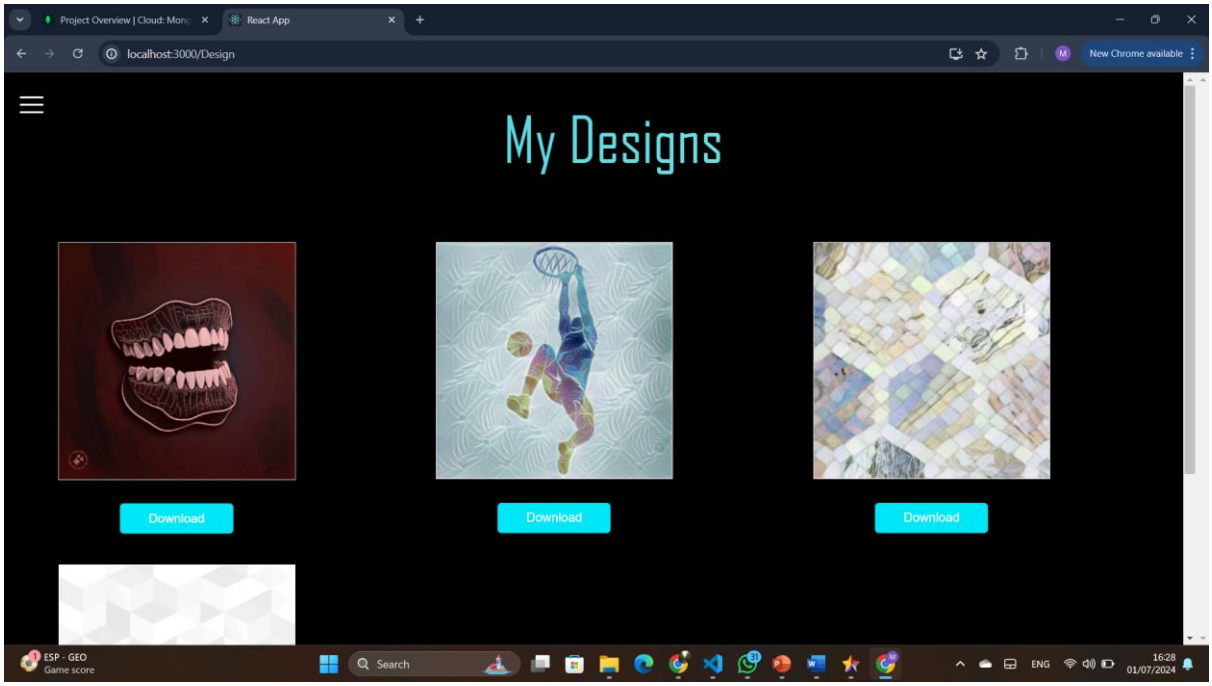

Figure 5.7: My Designs Page UI

5.3.8. My Album Page

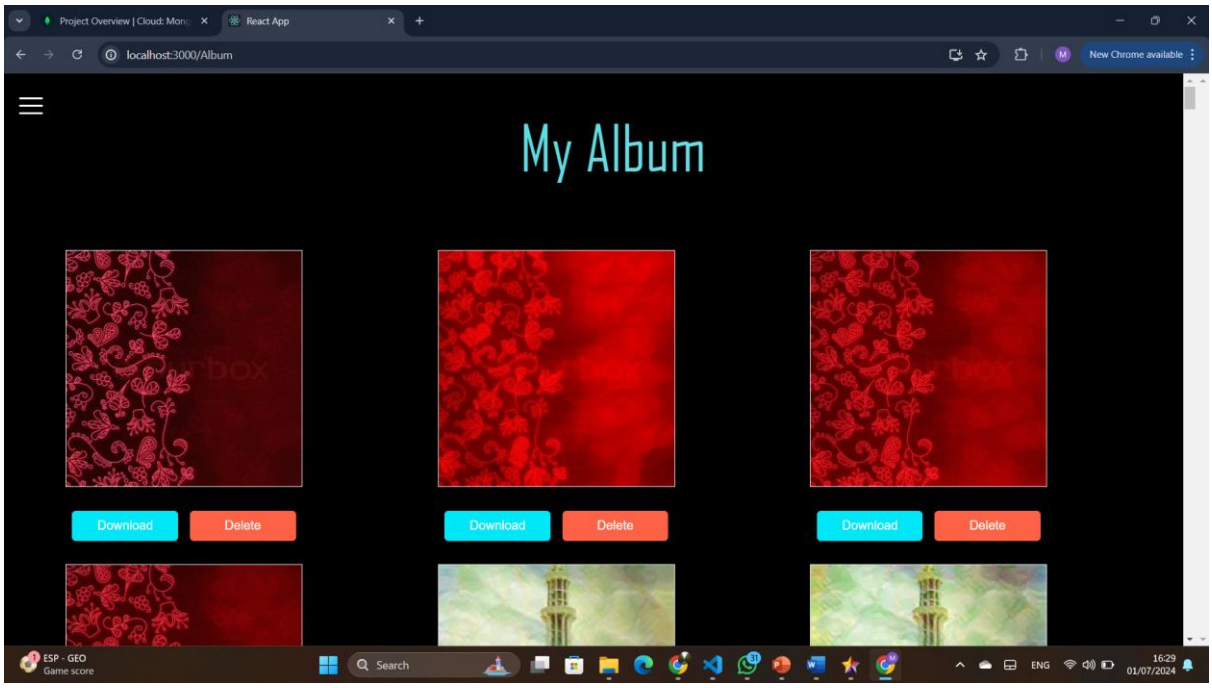

Figure 5.8: My Album Page UI

5.3.9. Favorites Page

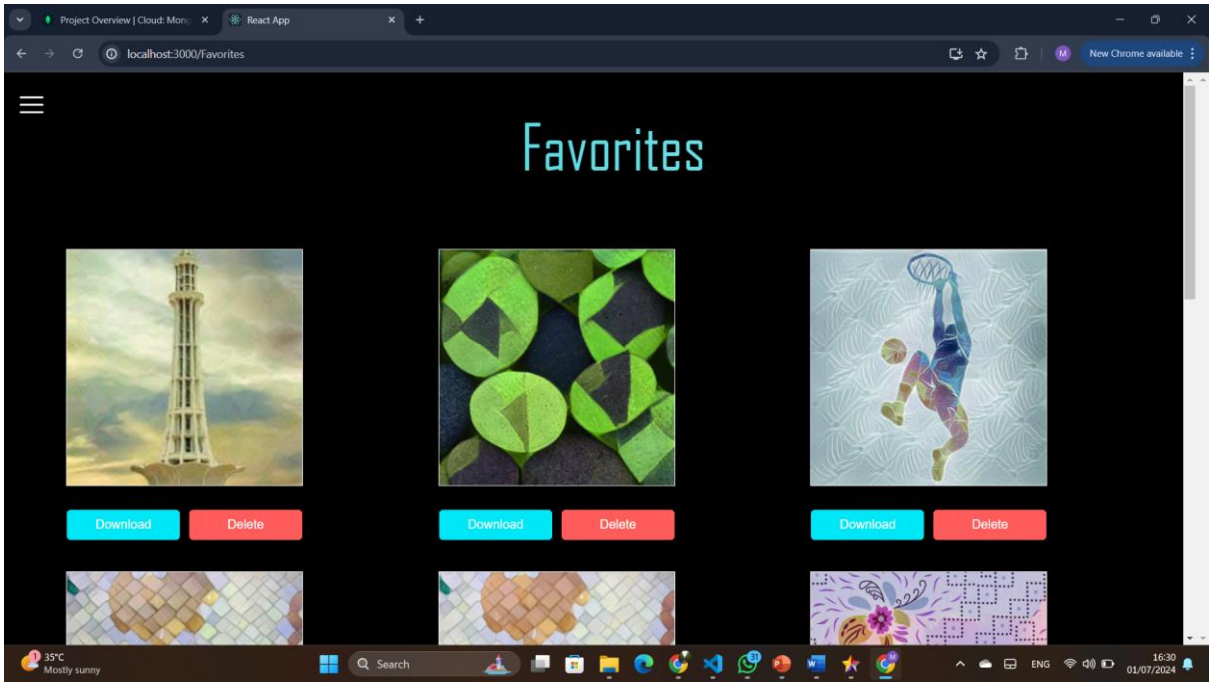

Figure 5.9: Favorites Page UI

5.3.10. How It Works Page

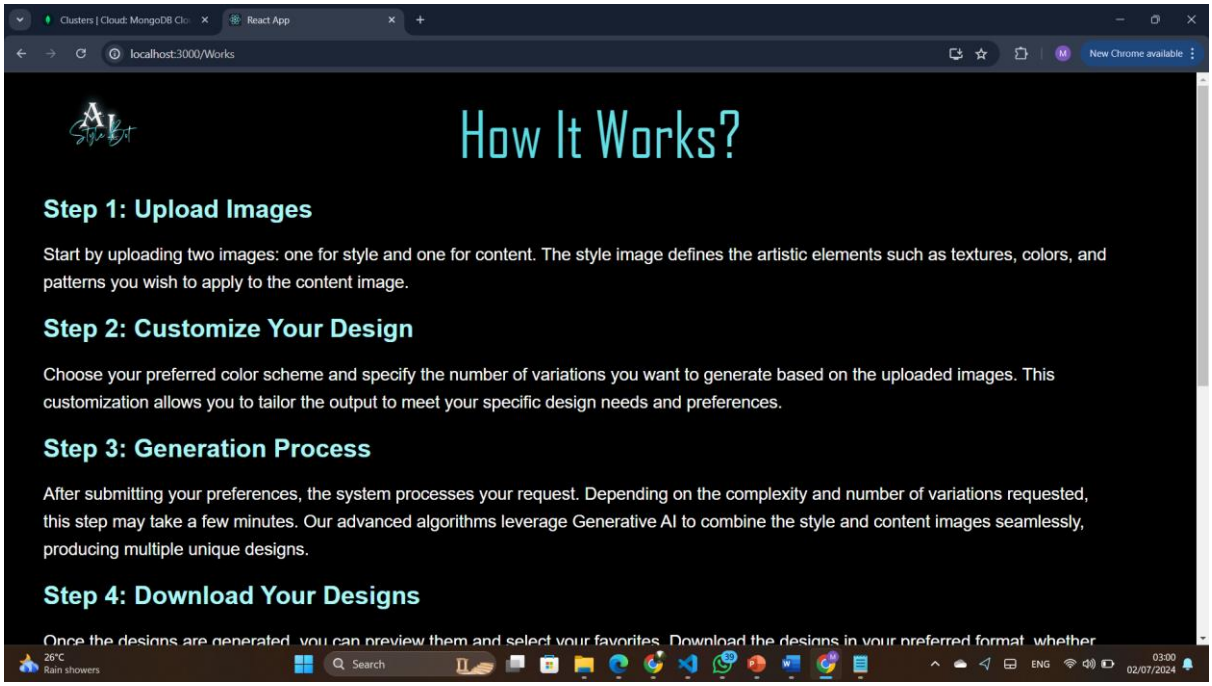

Figure 5.9: How It Works? Page UI

6. Testing and Evaluation

Testing in Generative AI Web Tool involves validating the functionality, performance, and usability of the system. It includes various types of testing, such as functional testing to ensure that the website and its features work as intended, integration testing to verify seamless integration.

6.1. Manual Testing

Manual testing is a type of software testing where the testing activities are carried out manually by human testers without the use of automated testing tools. It involves the manual execution of test cases. Following are the types of manual testing that we performed:

6.1.1. System testing

After the Generative AI Web tool has been constructed, system testing must be carried out to confirm that it performs as expected and conforms to the requirements that were originally established. The system's smooth and dependable operation is ensured by the testing phase's identification of any defects or issues that the user might not notice at first. All testing efforts (unit testing, functional testing, and integration testing) must be finished before releasing the decentralized marketplace to users. This lessens the likelihood of introducing bugs or missing features that might negatively affect the user experience or get in the way of effective project management.

6.1.2. Unit Testing

Unit testing involves creating test cases for each unit of code and executing them independently. Following are the tables of Unit Testing:

- **Unit Testing 1:** Sign Up
  Testing Objective: To validate that a user can successfully register for the system and create its profile. Table 6.1 illustrates the unit testing of registration.

Table 6.1: User Sign Up Test Cases

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify that the user is successfully register and directed to the login page.	Name: Haleema Email: haleema.jamil6@gmail.com Password: haleema.6 Confirm Password: haleema.6	The user should receive a code through entered email for verification.	Pass

2.	Verify if the system correctly prompts the user to enter a value in the Username field during sign-up when left empty.	Name field is left empty	System should prompt the user "Please fill out this field."	Pass
3.	Verify if the system correctly prompts the user to enter a value in the Password field during sign-up when left empty.	Password field is left empty.	System should prompt the user "Please fill out this field."	Pass
4.	Verify if the system correctly prompts the user to enter a value in the Email field during sign-up when left empty.	Email field is left empty.	System should prompt the user "Please fill out this field."	Pass
5.	Verify if the system correctly validates the email format during sign-up.	Email: haleema.jamil6.gmail.com	System should prompt the user "Please include an '@' in the email address."	Pass
6.	Verify if the system correctly checks that the Password and Confirm Password fields match during sign-up.	Password: haleema.6 Confirm Password:haleema6	System should prompt the user "Passwords do not match."	Pass
7.	Verify if the system correctly validates the length of the password during sign-up.	Password: 12345 Confirm Password: 12345	System should prompt the user "Password must be at least 8 Characters long."	Pass

- **Unit Testing 2:** Upload Images
  Testing Objective: To validate if system is taking the images as required to generate designs.

Table 6.2: Upload Images Test Cases

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify if the system correctly validates the file format of the uploaded images.	Upload images in '.bmp' format.	System should prompt the user "Please upload a valid PNG or JPG image file."	Pass
2.	Verify if the system allows the successful upload of images in PNG format.	Upload images in '.png' format.	Images should be uploaded successfully.	Pass
3.	Verify if the system allows the successful upload of images in JPG format.	Upload images in '.jpg' format.	Images should be uploaded successfully.	Pass
4.	Verify if the system allows the user to proceed without selecting a color.	Upload two valid images and do not select a color.	Designs should be generated without any errors when 'Apply Style' button is clicked.	Pass
5.	Verify if the system prompts the user to upload two images when only one image is uploaded.	Upload one valid images.	System should prompt the user "Please upload both Content and Style image."	Pass
6.	Verify if the system prompts the user to upload two images when no image is uploaded.	Do not upload any images.	System should prompt the user "Please upload both Content and Style image."	Pass
7.	Verify if the system correctly generates the specified number of images with the selected color applied.	Upload two valid images, select a color, and specify the number of images to be generated as 6.	System should generate 6 images with the selected color applied.	Pass

- **Unit Testing 3:** Add to Favourites
  Testing Objective: To validate if system is adding the generated designs in Favorites

Table 6.3: Download Generated Designs Test Case

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify if a user can successfully add a generated design to the favorites.	Click on the 'Heart/Favorite' Icon below the output images.	The designed image should be added to favorites folder.	Pass

- **Unit Testing 4:** Zoom Designs
  Testing Objective: To validate if system is showing zoom in designs that are generated.

Table 6.4: Zoom Generated Designs Test Case

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify if a user can successfully view a zoom version of generated design.	Click on the 'View' Icon below the output images.	The design should be opened in a new tab.	Pass

- **Unit Testing 5:** Delete Designs
  Testing Objective: To validate if system is deleting the designs that are generated.

Table 6.5: Delete Generated Designs Test Case

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify if a user can successfully delete a generated design.	Click on the 'Delete' Icon below the output images.	The designed image should be deleted.	Pass

6.1.3. Functional Testing

Functional testing will take place after the unit testing. In this functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements.

- **Functional Testing 1:** User Login
  Objective: Verify that the user can login successfully into the system.

Table 6.6: User Login Test Cases

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify user login after click on the 'Login' button on login form with correct input data.	Username: Haleema Password: haleema.6	Successfully log onto the main page of the system as Haleema.	Pass
2.	Verify user login after click on the 'Login' button on login form with incorrect input data	Username: 12345 Password:a1b2	Gives Error: Invalid Credentials	Pass

- **Functional Testing 2:** Generate Design
  Objective: Verify that the system generates the designs successfully when images are uploaded.

Table 6.7: Generate Design Test Cases

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify if the system correctly applies the selected color to the generated designs.	Upload two valid images. Select Color (#FF5733)	System should apply the selected color to the generated designs.	Pass
2.	Verify if the system generates 4 images by default when the number of images to be generated is not specified.	Upload two valid images and do not specify the number of images to be generated.	System should generate 4 images by default.	Pass
3.	Verify if the system generates the specified number of images.	Upload two valid images and specify the number of images to be generated as 2.	System should generate 2 images.	Pass
4.	Upload large image file	Large Image file	Upload should fail with an error message indicating the file size is too large.	Pass

- **Functional Testing 3:** Download Designs
  Testing Objective: To validate if system is downloading the designs that are generated.

Table 6.8: Download Generated Designs Test Case

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify if a user can successfully download a generated design from the system.	Click on the 'Download' Icon and choose the format.	The design file should be downloaded in the selected format to the local machine.	Pass

- **Functional Testing 4:** Check Similarity
  Testing Objective: To validate if system is showing the similarity between input images and generated images.

Table 6.9: Check Similarity Test Case

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify if system is showing similar generated images with input images when slider is moved.	Move the slider bar towards the start.	The system will start showing the images that are closest to input images.	Pass

- **Functional Testing 5:** Album
  Testing Objective: To validate if system adds all the generated designs to My Album Folder.

Table 6.10: My Album Test Case

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify if all the designs that are generated are added to My Album Folder.	Generate a design and check 'My Album'.	The designs generated should be available to view in My Album.	Pass

6.1.4. Integration Testing

Integration testing involves testing the interfaces and interactions between various software modules, subsystems, or external systems to ensure that they work correctly together. This Section shows the Integration Testing:

- **Integration Testing 1:** Check Previous Designs
  Testing Objective: To validate if system has previously generated designs of the user.

Table 6.11: Check Previous Designs Test Cases

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Sign in	Enter Valid Credentials that were used to Sign Up.	Login successful and directto the home page.	Pass
2.	Verify that the user can view, download and delete the previously generated images.	Check the My Album, design and Favorites folder for previous designs.	All the designs previously generated should be as it.	Pass

- **Integration Testing 2:** Design Generation
  Testing Objective: To validate if the user can successfully generate designs after logging into the system again.

Table 6.12. Generate Design Test Cases

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Verify if the system correctly applies the selected color to the generated designs.	Upload two valid images. Select Color (#FF5733)	System should apply the selected color to the generated designs.	Pass
2.	Verify if the system generates 4 images by default when the number of images to be generated is not specified.	Upload two valid images and do not specify the number of images to be generated.	System should generate 4 images by default.	Pass
3.	Verify if the system generates the specified number of images.	Upload two valid images and specify the number of images to be generated as 2.	System should generate 2 images.	Pass

7. Conclusion and Future Work

In conclusion, implementing a Generative AI Web Tool for Fashion Designing can bring significant benefits to the fashion design industry. In future, we can enhance the user experience by adding more customization features, more options to view the generated designs as a dress etc. In future, going to add cross-platform compatibility and collaborations with international artists.

7.1. Conclusion

The AI Style Bot project has successfully demonstrated the potential of artificial intelligence in revolutionizing the fashion industry. By integrating cutting-edge AI techniques like Neural Style Transfer (NST), the project has created a tool that automates the fashion design process, enabling designers to generate unique and aesthetically pleasing designs with ease. This automation not only significantly reduces the time and effort traditionally required for fashion design but also unleashes a new level of creativity and innovation. Designers can now experiment with a vast array of styles and patterns, pushing the boundaries of their creativity without the constraints of manual design processes. Furthermore, the robust and user-friendly web application ensures that both seasoned designers and novices can seamlessly navigate and utilize the tool, making high-quality fashion design accessible to a broader audience.

The meticulous alignment of the AI Style Bot with core software engineering principles has resulted in a highly efficient and reliable application. By addressing key challenges in fashion design, such as the need for rapid prototyping and customization, the AI Style Bot offers substantial benefits not only to individual designers but also to fashion brands and businesses looking to streamline their design workflows. The project's success underscores the transformative potential of AI in creative industries, paving the way for future innovations. As the fashion industry continues to evolve, tools like the AI Style Bot will play an increasingly important role in shaping the future of fashion design, fostering a dynamic and responsive design environment that can adapt to the ever-changing trends and preferences of consumers.

7.2. Future Work

Looking ahead, the development and enhancement of the AI Style Bot can be further advanced through several key initiatives. One significant area for improvement is the implementation of more sophisticated customization features. By allowing users to have finer control over specific design elements such as textures, patterns, and accessory placements, the tool can cater to more detailed and personalized design requirements. This would enable designers to create even more distinctive and tailored fashion pieces, enhancing the overall value and appeal of the generated designs. Additionally, integrating more advanced AI models and algorithms, potentially

incorporating techniques from reinforcement learning or evolutionary algorithms, could further enhance the quality and variety of the designs produced by the tool.

Another promising direction for future work is the development of collaborative functionalities. By enabling multiple designers to work on the same project simultaneously, the AI Style Bot can facilitate teamwork and the sharing of ideas, fostering a more collaborative and creative design environment. Direct integration with e-commerce platforms is another strategic enhancement, allowing designers to showcase and sell their generated designs directly to consumers, thereby bridging the gap between design and market. The creation of a mobile application version of the AI Style Bot would also make the tool more accessible to designers on the go, increasing its usability and reach. Establishing a feedback mechanism where users can provide input on generated designs will help continuously refine and improve the AI models based on real-world usage and preferences. Expanding the library of styles, patterns, and design elements from various cultural and historical contexts will offer users a broader range of inspiration, making the AI Style Bot an even more valuable resource for fashion designers worldwide. By pursuing these enhancements, the AI Style Bot can continue to evolve and maintain its position at the forefront of innovation in the fashion design industry.

8. References

1. Haykin, S. (1998). Neural Networks: A Comprehensive Foundation (2nd ed.). Prentice Hall.
2. Dilmegani, C. Generative AI in the Fashion Industry: 5 Use Cases with Case Studies. Aimultiple. Retrieved from https://research.aimultiple.com/
3. Lau, L. M. S., Curson, J., Drew, R., Dew, P., & Leigh, C. (1999). Use Of VSP Resource Rooms to Support Group Work in a Learning Environment. ACM 99, 2.
4. Nakanishi, H., Yoshida, C., Nishmora, T., & Ishada, T. (1996). FreeWalk: Supporting Casual Meetings in a Network. ACM, 308-314. Retrieved from http://www.acm.org/pubs/articles/proceedings/cscw/240080/p308-nakanishi.pdf
5. Behforooz, A., & Hudson, F. J. (1996). Software Engineering Fundamentals. Oxford University Press.
6. Page Author. Page Title. Retrieved from http://www.bt.com/bttj/archive.htm
7. Scott, W. (2005). Agile Modeling: Artifacts for UML (OOP). Retrieved from Agile Modeling
8. Ullman, J. D. (1982). The Complete Book of Database Systems. Retrieved from ELTE Database Systems
9. Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.
10. Ambler, S. W. (2005). Data Flow Diagrams (DFD). Retrieved from Agile Modeling
11. Sommerville, I. (2010). Software Engineering (9th ed.). Addison-Wesley.
12. Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Pearson.