**Persona Styler 24-FYP-302**



**Session 2020-2024**

**BACHELOR OF SCIENCE**

**IN**

**INFORMATION TECHNOLOGY**

**SUBMITTED BY**

Maryam Rehman 20-NTU-CS-1154

Meerat Ijaz 20-NTU-CS-1155

Sahar Sajjad 20-NTU-CS-1180

**SUPERVISED BY**

Dr. Rehan Ashraf

**CO-SUPERVISED BY**

Dr. Sajida Parveen

**DEPARTMENT OF COMPUTER SCIENCE**

**NATIONAL TEXTILE UNIVERSITY, FAISALABAD**

**CERTIFICATION**

It is to attest that this project titled “Persona Styler” satisfies the requirements for degree BSIT by the Department of Computer Science, National Textile University.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor: Date**

Dr. Rehan Ashraf

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**External Examiner Date**

Name:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Examiner (FYP-Member) Date**

Name:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**FYP-Convener Date**

Dr. M. Abdul Qayyum

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Head of Department Date**

Dr. Muhammad Asif Malik

**DECLARATION**

We hereby declared that this document is completely written by Maryam Rehman, Meerat Ijaz, and Sahar Sajjad under the supervision of our supervisor, Dr. Rehan Ashraf, and co-supervisor, Dr. Sajida Parveen, and it is totally our effort and no one from outside of our group have copied it. This report is written in accordance with our project.

**Group Members:**

Maryam Rehman \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

20-NTU-CS-1154 Signature Date

Meerat Ijaz \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

20-NTU-CS-1155 Signature Date

Sahar Sajjad \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

20-NTU-CS-1180 Signature Date

**ACKNOWLEDGEMENT**

We developed this project by putting our maximum effort and devotion. During the development of this project. We are very thankful to those who rewarded us with their precious time, guidance, and kind advice.

We are very thankful to Allah Almighty for countless blessings, wisdom, knowledge, information, and guidance which he blessed during this duration. We are nothing without these countless efforts of Allah Almighty. My group members and seniors played a very major role. They always guided, helped, strengthened, encouraged, and supported us.

We pay our appreciable acknowledgment to the most admirable Faculty Members of our Department. They gave us valuable time from their busy schedules and guided us in every aspect. We would like to thank our **Supervisor Dr. Rehan Ashraf** and **Co-Supervisor Dr. Sajida Parveen** who guided us regarding every problem of the project during the development phase. They were always available for us to guide, to give suggestions and teach us how more improvements could be made to the project.

**ABSTRACT**

The "Persona Styler" project aims to change the way people navigate the complex world of fashion. Its goal is to address the common struggle of choosing the right clothes among the vast and diverse fashion industry offerings. By utilizing high and leverage technologies including machine learning, computer vision and modeling techniques the project seeks to provide a personalized fashion app. This app will allow users to provide customized clothing suggestions modified to their unique style. Factors such as skin tone, facial structure, color choices and type of event will be taken into consideration when providing personalized fashion advice. Using machine learning capabilities, the app will constantly adjust its recommendations based on user feedback to ensure a reliable experience. The development of the "Persona Styler" application will use databases to manage user data and preferences, in addition to using Python for coding, Visual Studio code for development, and web development tools to create a simple user interface. Ultimately, the "Persona Styler" project aims to give users the confident ability to make fashion choices effortlessly. Seamlessly combining advanced technology with easy-to-use principles, the app will provide a simple and effective experience that will change the way people view fashion in the digital age.

Table of Contents

Descriptions Page No

Table of Contents

[CHAPTER 1 1](#_Toc160135771)

[INTRODUCTION 1](#_Toc160135772)

[**1.1 Background** 1](#_Toc160135773)

[**1.2 Problem Statement** 1](#_Toc160135774)

[**1.3 Proposed Solution** 1](#_Toc160135775)

[**1.4 Purpose** 2](#_Toc160135776)

[**1.5 Project Objectives** 2](#_Toc160135777)

[**1.6 Project Scope** 2](#_Toc160135778)

[**1.7 Gantt Chart** 2](#_Toc160135779)

[**1.8 Risk Management** 3](#_Toc160135780)

[**1.9 Risks and Risk Mitigation** 3](#_Toc160135781)

[**1.9.1 Cost Risk** 3](#_Toc160135782)

[**1.9.2 Performance Risk** 4](#_Toc160135783)

[**1.9.3 Time Risk** 4](#_Toc160135784)

[**1.9.4 Technical Risk** 4](#_Toc160135785)

[**1.9.5 Data Risk** 4](#_Toc160135786)

[CHAPTER 2 5](#_Toc160135787)

[LITERATURE REVIEW 5](#_Toc160135788)

[**2.1 Related Work** 5](#_Toc160135789)

[**2.1.1 Reeta Koshy(2021)** 5](#_Toc160135790)

[**2.1.2 B Dahunsi (2023)** 5](#_Toc160135791)

[**2.1.3 Zong, Wenjia (2022)** 5](#_Toc160135792)

[**2.1.4 Huizhong Chen, Andrew Gallagher & Bernd Girod(2012)** 5](#_Toc160135793)

[**2.1.5 David Ian Perrett (2021)** 6](#_Toc160135794)

[**2.1.6 Yasser A. Nada (2014)** 6](#_Toc160135795)

[**2.1.7 CD Kokane (2023)** 6](#_Toc160135796)

[**2.1.8 D. Vogiatzis (2012)** 6](#_Toc160135797)

[**2.1.9 D. Pierrakos (2009)** 6](#_Toc160135798)

[**2.1.10 P Wickramarathne (2019)** 6](#_Toc160135799)

[**2.2 Reasons to Develop** 6](#_Toc160135800)

[CHAPTER 3 8](#_Toc160135801)

[METHODOLOGY 8](#_Toc160135802)

[**3.1 Methodology for Software Development** 8](#_Toc160135803)

[**3.2 Agile Methodology** 8](#_Toc160135804)

[**3.3 Reasons for Choosing Agile Methodology** 9](#_Toc160135805)

[**3.3.1 Flexibility and Adaptability** 9](#_Toc160135806)

[**3.3.2 Incremental Development** 9](#_Toc160135807)

[**3.3.3 Risk Mitigation** 9](#_Toc160135808)

[**3.3.4 Timely Delivery** 9](#_Toc160135809)

[**3.4** **Why not Other Methodologies?** 9](#_Toc160135810)

[CHAPTER 4 10](#_Toc160135811)

[SYSTEM REQUIREMENTS 10](#_Toc160135812)

[**4.1 Hardware Requirements** 10](#_Toc160135813)

[**4.1.1 Server Infrastructure:** 10](#_Toc160135814)

[**4.1.2 Storage:** 10](#_Toc160135815)

[**4.1.3 Processing Power:** 10](#_Toc160135816)

[**4.1.4 Backup and Redundancy:** 10](#_Toc160135817)

[**4.1.5 Networking:** 10](#_Toc160135818)

[**4.1.6 Compliance and Security:** 11](#_Toc160135819)

[**4.2 Software Requirements** 11](#_Toc160135820)

[**4.2.1 Programming Languages and Frameworks:** 11](#_Toc160135821)

[**4.2.2 Integrated Development Environment (IDE):** 11](#_Toc160135822)

[**4.2.3 Database Management System:** 11](#_Toc160135823)

[**4.2.4 Computer Vision and Machine Learning Libraries:** 11](#_Toc160135824)

[**4.2.5 Image Processing Tools:** 11](#_Toc160135825)

[**4.3 Functional Requirements** 11](#_Toc160135826)

[**4.3.1 User Profile Management** 11](#_Toc160135827)

[**4.3.2 Clothing Recommendation System** 11](#_Toc160135828)

[**4.3.3 Color Combination Suggestions** 12](#_Toc160135829)

[**4.3.4 Event-Specific Outfit Recommendations** 12](#_Toc160135830)

[**4.3.5 Integration of Computer Vision and Machine Learning** 12](#_Toc160135831)

[**4.3.6 User Feedback Mechanism** 12](#_Toc160135832)

[**4.4 Non-Functional Requirements** 12](#_Toc160135833)

[**4.4.1 Performance** 12](#_Toc160135834)

[**4.4.2 Scalability** 12](#_Toc160135835)

[**4.4.3 Security** 12](#_Toc160135836)

[**4.4.4 Accuracy** 12](#_Toc160135837)

[**4.4.5 User Interface** 12](#_Toc160135838)

[**4.4.6 Reliability** 12](#_Toc160135839)

[**4.5 Use Case Diagram** 13](#_Toc160135840)

[**4.6 Activity Diagram** 14](#_Toc160135841)

[**4.7 Class Diagram** 15](#_Toc160135842)

[CHAPTER 5 16](#_Toc160135843)

[SYSTEM ARCHITECTURE 16](#_Toc160135844)

[**5.1 Components of the System Architecture** 16](#_Toc160135845)

[**5.1.1 User Interface (UI) Component** 16](#_Toc160135846)

[**5.1.2 Web Server Component** 16](#_Toc160135847)

[**5.2 Backend Services** 16](#_Toc160135848)

[**5.2.1 User Data Handling Module:** 16](#_Toc160135849)

[**5.2.2 Machine Learning Recommendation Engine:** 17](#_Toc160135850)

[**5.2.3 Computer Vision Analysis Module:** 17](#_Toc160135851)

[**5.2.4 Color Combination Generation Module:** 17](#_Toc160135852)

[**5.2.5 Event-Specific Outfit Recommendation Module:** 17](#_Toc160135853)

[**5.3 Sequence Diagrams** 17](#_Toc160135854)

[References 19](#_Toc160135855)

# List of Figures

[Figure 1.1 Gantt Chart 3](#figure11)

[Figure 3.1 Project Planning 8](#figure31)

[Figure 3.2 Agile Model 5](#figure32)

Figure 4.1 User Case Diagram of User 2

[Figure 4.2 User Case Diagram of Admin 3](#figure42)

[Figure 4.3 User Case Diagram of Full System 5](#figure43)

[Figure 5.1 Sequence Diagram of System 3](#figure51)

[Figure 5.2 Activity Diagram of Sign-Up 5](#figure52)

[Figure 5.3 Activity Diagram of Login 5](#figure53)

[Figure 5.4 Activity Diagram of Complete Sytem 5](#figure54)

[Figure 5.5 Database Design Diagram 5](#figure55)

# List of Tables

[Table 4.1 Description of User’s Sign-Up Use Case 13](#_Toc111292070)

[Table 4.2 Description of User’s Login Use Case 13](#_Toc111292070)

[Table 4.3 Description of User’s Image Upload Use Case 13](#_Toc111292070)

[Table 4.4 Description of User’s View Result Use Case 13](#_Toc111292070)

[Table 4.5 Description of User’s Send Message Use Case 13](#_Toc111292070)

[Table 4.6 Description of User’s Logout Use Case 13](#_Toc111292070)

[Table 4.7 Description of Admin’s Sign-Up Use Case 13](#_Toc111292070)

[Table 4.8 Description of Admin’s Login Use Case 13](#_Toc111292070)

[Table 4.9 Description of Admin’s Edit Main Page Use Case 13](#_Toc111292070)

[Table 4.10 Description of Admin’s Read Messages Use Case 13](#_Toc111292070)

[Table 4.11 Description of Admin’s Reply Messages Use Case 13](#_Toc111292070)

[Table 4.12 Description of Admin’s Search User Use Case 13](#_Toc111292070)

[Table 4.13 Description of Admin’s Logout Use Case 13](#_Toc111292070)

# CHAPTER 1

## INTRODUCTION

In this world, many peoples are confused about what they are going to wear for an event. So, they want to have some who help them in selecting clothes. That’s why we developed this app to solve users’ problems and the title is “Persona Styler”. And the aim of this app is revolutionizing the way people approach fashion sense by offering a personalized fashion app. Because the fashion world can be confusing, with so many different styles and options. This app uses technology to make things simpler. This app uses smart technology to help you pick clothes. It looks at your body shape, your skin complexion, and the colors you like. Then, it suggests outfits that would look good on you. It also considers the type of event you're going to, so that you can dressed up according to it. It's like having your own styling expert, that helps you to look best!

### 1.1 Background

The background of the “Persona Styler" project is to simplify fashion choices by creating a personalized app that helps users to select outfits according to their event. The fashion industry is vast, and it makes hard for people to find the right clothes according to their body type, style, and the occasion. This app uses technology to analyze the user's skin tone, facial structure, and color preferences to provide different recommendations. It also suggests color combinations and outfits based on the event type, whether it's a daytime or evening event. By using machine learning, computer vision, and image processing, the app offers fashion advices. It became easier for users to make stylish and different clothing combinations. Overall, the aim of this project is to revolutionize the fashion experience, making it more enjoyable and accessible to everyone.

### 1.2 Problem Statement

A personalized fashion app is designed to help users select the perfect outfit by utilizing machine learning and computer vision. It provides clothing recommendations based on skin complexion, facial structure, and body type, along with color combinations and event-specific suggestions. This app simplifies the decision-making process, offering designer fashion advice tailored to individual preferences and needs.

### 1.3 Proposed Solution

The main solution for the "Persona Styler" app is the Clothing Recommendation Module. This module lets users enter their image and the type of event they're dressing for (day or night). Then, it gives personalized outfit suggestions. For example, if someone has fair skin and a round face, it might suggest lighter colors and V-neck tops for daytime, and darker colors with more structured clothes for night-time. It also suggests color combinations based on the user's selected colors and skin tone. The module uses machine learning and computer vision to analyze images and give recommendations.

### 1.4 Purpose

The purpose of this project is to create a personalized fashion app that helps users to select outfits according to their body type, and the event they're attending. The fashion industry offers a wide range of choices, making it challenging for individuals to find the right clothes. This app aims to simplify the process by providing personalized recommendations based on the user's skin tone, facial structure, and color preferences. By using data and technology, the app offers amazing outfit suggestions for different occasions. It makes fashion choices easier and more enjoyable for users.

### 1.5 Project Objectives

The main objectives to develop this project is listed here:

* Aim to develop a personalized fashion app for users to choose outfits that suit their body shape, style, and event.
* This project utilizes data and technology to provide updated fashion choices.
* It also offers personalized clothing recommendations based on skin complexion and facial structure.
* This project also provides color combination suggestions based on selected colors and skin complexion.
* Offer event-specific outfit recommendations for daytime or evening events.
* Use machine learning and computer vision techniques for personalized recommendations.
* Train algorithms on a dataset of images to improve recommendation accuracy.
* Implement features for users to input specific events and desired styles.
* Include a color combination feature based on user-selected colors.

### 1.6 Project Scope

The scope of this project includes developing a web-based software application that provides personalized clothing recommendations. The app will offer suggestions based on the user's skin complexion, facial structure, selected colors, and event type (daytime or evening). Users will receive outfit recommendations for various occasions, such as weddings or casual outings. Additionally, the app will suggest color combinations that complement the user's skin tone and selected colors. The project will utilize machine learning and computer vision techniques to power the Clothing Recommendation Module, ensuring accurate suggestions. The aim of this app is to simplify the process of choosing outfits by providing personalized recommendations based on users’ preferences and attributes.

### 1.7 Gantt Chart

Schedule of activities and Gantt chart:

|  |  |
| --- | --- |
| **Activities** | **Tentative Date** |
| Research and Literature Review | 12 December 2023 – 15 January 2024 |
| User Interface Design | 11 January 2024 – 19 January 2024 |
| Dataset Collection and Preparation | 18 January 2024 – 23 January 2024 |
| Data Preprocessing | 26 January 2024 – 16 February 2024 |
| Model Training | 19 February 2024 – 01 March 2024 |
| Post Preprocessing | 04 March 2024 – 22 March 2024 |
| Validation and Optimization | 25 March 2024 – 16 April 2024 |
| Testing and Evaluation | 18 April 2024 – 15 May 2024 |

Figure 1.1 shows the Gant Chart.

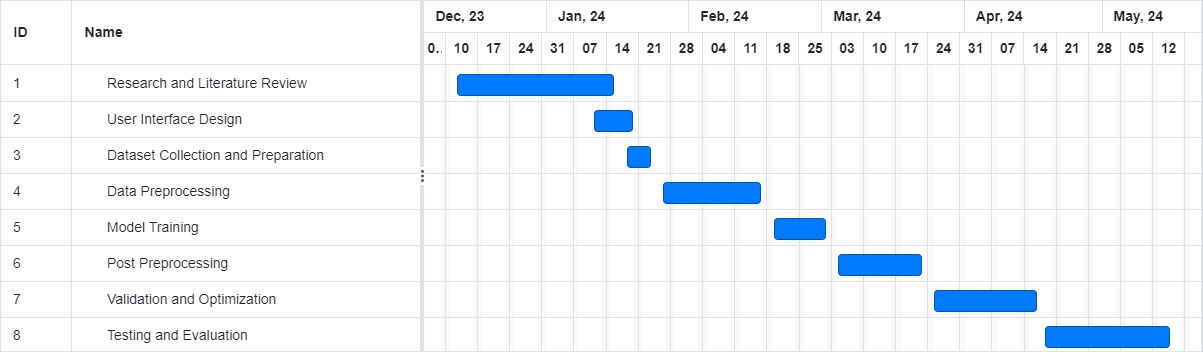
****

Figure 1.1 Gantt Chart

### 1.8 Risk Management

It involves identifying and modifying potential risks that could delay its success. One key risk is data privacy and security, ensuring that user information is protected. Another risk is algorithm accuracy, ensuring that machine learning and computer vision algorithms provide relevant recommendations. User adoption is also a concern, requiring the app to be user-friendly and meet user needs. Technical challenges may also arise during the development. Competition is another risk, requiring the app to offer unique features. Finally, legal and compliance risks, such as data protection laws, must be managed. By managing these risks, the chance of success can be increased.

### 1.9 Risks and Risk Mitigation

There are several risks that could potentially arise during the development of the Persona Styler software project. The risks outlined below are particularly relevant to this project:

### 1.9.1 Cost Risk

Cost risks may emerge due to various factors such as budget overruns associated with acquiring datasets, licensing fees for software tools, and additional resources needed for testing and validation. There is also a risk of underestimating costs for training machine learning models and maintaining the app. To mitigate these risks, careful budget planning, regular expense monitoring, and transparent communication with stakeholders are essential.

### 1.9.2 Performance Risk

Performance risks may arise from challenges in achieving the desired accuracy and reliability of the Persona Styler app. Factors such as inadequate training data, algorithmic limitations, and complexities in feature analysis could impact the app's ability to provide accurate recommendations. Additionally, real-time detection capabilities may face performance issues related to processing speed. To mitigate these risks, thorough testing, algorithm optimization, and continuous validation against datasets are crucial.

### 1.9.3 Time Risk

Time risks may manifest as delays in project milestones due to unexpected technical challenges, data acquisition delays, or complexities in algorithm development. To mitigate these risks, a detailed project plan with clearly defined milestones and deadlines should be developed. Regular progress monitoring and open communication with stakeholders can help manage and address any issues affecting project timelines.

### 1.9.4 Technical Risk

Technical risks may involve unforeseen challenges during the development and implementation of the Persona Styler app, such as algorithmic limitations or integration difficulties. Thorough research before starting the project, regular engagement with domain experts, and conducting various tests can help identify and address technical issues early in the development process.

### 1.9.5 Data Risk

Data risks may arise from insufficient or biased datasets, leading to poor performance of the app. To mitigate these risks, diverse datasets containing genuine and forged images should be carefully assembled. Data preprocessing techniques should be implemented to address biases and ensure dataset quality. Regular validation and refinement of datasets are also essential to enhance the app's performance and reliability.

# CHAPTER 2

## LITERATURE REVIEW

The rising demand for personalized fashion solutions due to challenges in attire selection. It highlights the fashion industry's complexity and the need for tailored recommendations based on factors like body type and event types. Technology, especially machine learning and computer vision, is seen as crucial for enhancing personalization. Current fashion apps are noted for their limitations in providing truly personalized recommendations, highlighting the gap the "Persona Styler" project addresses by leveraging advanced technologies for more user-centric solutions.

### 2.1 Related Work

### 2.1.1 Reeta Koshy(2021)

Reeta Koshy have used the nearest neighbor PageRank Algorithm for personalized fashion recommendations. A unique complexion-based outfit color recommender has been designed by authors using neural networks. A graph-based social media recommendation system has been prosed, the application of which can be extended to other applications as well. [1]

### 2.1.2 B Dahunsi (2023)

Fashion recommender systems assist users in selecting clothing by prioritizing items likely to interest them, but data scarcity hinders predictive accuracy. Expert knowledge exploration aims to define user profiles effectively, enhancing recommender system performance.[2]

### 2.1.3 Zong, Wenjia (2022)

Research develops body shape-based style recommendation system for apparel, aligning dress attributes with body shape attributes. Utilizes Female Figure Identification Technique (FFIT) validated against SizeUSA measurements. Despite inconclusive significance, reveals distinctive dress style preferences based on body shapes.[3]

### 2.1.4 Huizhong Chen, Andrew Gallagher & Bernd Girod(2012)

Proposing an automated system to generate nameable attributes for clothes on human bodies in images. Utilizes pose-adaptive low-level features and combines them for attribute classification. Employs a Conditional Random Field to capture mutual dependencies between attributes and enhance predictions, validated on a challenging dataset, with an additional application in dressing style analysis.[4]

### 2.1.5 David Ian Perrett (2021)

Fashion stylists advise clothing colors based on personal attributes like skin tone, hair, and eye color, yet these categories lack scientific definition and exhibit inconsistency. Recent studies reveal observer preferences for red and blue hues, with a preference for cool blues to match fair skin and warm orange/red hues for tanned skin, suggesting that skin tone influences clothing color preferences.[5]

### 2.1.6 Yasser A. Nada (2014)

An expert system using Forward Chaining and CLIPS is proposed for personalized clothing style selection, integrating expert knowledge on materials, colors, body types, and facial features to aid in careful outfit selection and enhance personal appearance.[6]

### 2.1.7 CD Kokane (2023)

Developing an ML-based Outfit Suggestion system leveraging deep learning, computer vision, and natural language processing techniques for personalized fashion recommendations based on body type and shape.[7]

### 2.1.8 D. Vogiatzis (2012)

Proposing a garment recommendation framework: 1) Utilizing owl ontology with fashion expert knowledge; 2) Incorporating community-based purchase behavior patterns.[8]

### 2.1.9 D. Pierrakos (2009)

This introduces a personalized clothing recommendation framework integrating fashion knowledge into an ontology, coupled with expert-defined style rules and user interaction data mining. It utilizes a personalization server to store rules as user stereotypes, enhancing recommendations through a recommendation engine based on these stereotypes.[9]

### 2.1.10 P Wickramarathne (2019)

"TrendiTex" is a user-friendly fashion design platform that recommends trending designs based on user preferences and body shape prediction, offering augmented fit-on features to streamline the selection process in today's fast-paced fashion landscape.[10]

### 2.2 Contribution

Our contribution in the “Persona Styler” project include:

**2.2.1** Personalized Fashion Recommendations: We provide personalized clothing recommendations based on user attributes such as skin complexion, facial structure, selected colors, and preferred event types (daytime or evening), ensuring that users receive tailored suggestions that match their unique style and preferences.

**2.2.2** Advanced Recommendation Algorithms: Leveraging machine learning and computer vision techniques, we have developed sophisticated algorithms to analyze user data effectively. These algorithms can handle data scarcity and provide accurate recommendations by learning from user feedback and trends in fashion.

**2.2.3** Body Shape-based Recommendations: Our system offers recommendations aligned with body shape attributes, ensuring that users receive suggestions that complement their physique and enhance their appearance.

**2.2.4** Color Combination Suggestions: We provide color combination suggestions based on user-selected colors and skin complexion, helping users create stylish and harmonious outfits.

**2.2.5** Event-specific Outfit Recommendations: Users can receive outfit recommendations tailored to specific events, ensuring that they always dress appropriately for any occasion.

**2.2.6** User-friendly Interface: Our web-based application features an intuitive and user-friendly interface, making it easy for users to input their preferences and navigate through the recommendation process effortlessly.

**2.2.7** Machine Learning and Computer Vision Integration: By integrating machine learning and computer vision technologies, we enhance the accuracy and efficiency of our recommendation system, ensuring that users receive high-quality suggestions that align with their preferences.

**2.2.8** Enhanced User Experience: We prioritize user satisfaction by providing curated suggestions, simplifying the decision-making process, and offering comprehensive support through documentation, marketing, and user assistance, ensuring a positive and fulfilling experience for our users.

### 2.3 Reasons to Develop

The inspiration behind Persona Styler's creation is rooted in the desire to confront the everyday struggles people face in the dizzying and complex world of fashion. The vast array of clothing, apparel, and accessories available makes it next to impossible to navigate through the countless outfit combinations that stand to flatter one’s body type, fit personal style, and work for a specific event. Persona Styler aims to resolve this dilemma in style, leveraging data and technology to produce fashion recommendations that are completely tailored to the specific attributes and preferences of each user.

At present, fashion apps are fraught with generic recommendations that don't always match the unique tastes or characteristics of individual users. The creators of Persona Styler believe that their app enjoys an important edge over the competition thanks to its emphasis on personalization.

# CHAPTER 3

## METHODOLOGY

This chapter outlines the methodology that is to be used in the development of this project. Development methodology plays a vital role in the development of any project. The methodology is structured to achieve the objectives that have been set forth in Chapter 1, including the creation of a clothing recommendation model.

### 3.1 Project Planning

*Figure 3.1 Project Planning*

### 3.2 Methodology for Software Development

The software development methodology that would be most suitable for our project, “Persona Styler,” would be “Agile Methodology.”

### 3.3 Agile Methodology

In Agile, we work in short cycles called sprints, and each sprint usually lasts for 2 weeks, where we focus on completing a small, specific part of the project. In agile methodology, meetings usually occur at predetermined times during the sprint cycle. The most common meetings are:

* Sprint Planning: This planning is done at the beginning of each sprint.
* Daily Standup: This is held daily during the sprint.
* Sprint Review: This is done at the end of each sprint.
* Sprint Retrospective: At the end of each sprint. It is just like the sprint review.

At the end of each sprint, we review what has been done and adjust the plans for the next sprint based on feedback and new priorities. The stages of agile methodology are:

* Plan
* Design
* Develop
* Test
* Deploy
* Review
* Launch

Figure 3.2 shows the Agile Methodology Diagram.

*A diagram of a software development process

Description automatically generated*

Figure 3.2 Agile Model

### 3.4 Reasons for Choosing Agile Methodology

### 3.4.1 Flexibility and Adaptability

An agile approach allows flexibility for changes and iterations throughout the development process. In a dynamic domain like fashion, where trends evolve rapidly, agility enables quick adjustments to user preferences and industry shifts.

### 3.4.2 Incremental Development

By breaking down the project into manageable increments or sprints, Agile enables the delivery of working features at regular intervals. Users can start benefiting from the application's functionality early on, and later iterations will add further enhancements.

### 3.4.3 Risk Mitigation

Agile methodology emphasizes early and frequent testing, allowing for the identification and mitigate risk early. This proactive approach reduces the likelihood of major issues arising later in the development cycle.

### 3.4.4 Timely Delivery

By prioritizing and repeatedly delivering value-based products, Agile ensures that the most important activities are implemented first. This approach ensures just-in-time delivery, allowing users to start reaping the benefits sooner.

### 3.5 Why not Other Methodologies?

Other methodologies like Waterfall or Spiral Model were not chosen for the "Persona Styler" project because they are too rigid and sequential for a dynamic domain like fashion where trends evolve rapidly. Additionally, the project involves complex technologies such as machine learning and computer vision, which require experimentation and adaptation to emerging trends. Agile methodology offers the flexibility and adaptability needed for iterative development and continuous feedback. It also promotes continuous improvements and risk management, making it the most suitable choice for developing a personalized fashion app that meets user expectations and adapts to changing market trends.

# CHAPTER 4

## SYSTEM REQUIREMENTS

System requirements serve as a blueprint for the design, development, and implementation of any project. It is considered as one of the basic steps. Clear and effective system requirements are crucial because if the requirements are not clear then we might not get the desired outcome. Thus, this chapter provides an in-depth analysis of the hardware and software components along with the functionals and non-functional requirements that are required to support the “Deepfake Image Analysis Using Spatial Features”. The details of these requirements are discussed further below.

### 4.1 Hardware Requirements

Hardware requirements specify the physical components which are necessary for the system’s operation. These requirements ensure that the system has all the necessary computational power, storage capacity, and connectivity which is essential for supporting the system’s functionality.

### 4.1.1 Server Infrastructure:

Hardware requirements include high-performance servers with reliable network connectivity, with plenty of storage to support machine learning, computer vision algorithms and easy user access.

### 4.1.2 Storage:

Storage solutions such as SSDs or HDDs must be scalable to meet current and future data needs, including application data, user profiles, images and multimedia content**.**

### 4.1.3 Processing Power:

Multi-core processors for processing concurrent requests, more RAM for large data processing, and possibly special GPUs or accelerators, especially for computer vision tasks, to speed up the image.

### 4.1.4 Backup and Redundancy:

Implement robust backup strategies and redundant server configuration, including cloud-based solutions, to reduce the risk of hardware failure or service disruption, ensuring data integrity and high availability.

### 4.1.5 Networking:

High-speed Internet connections and secure network protocols with encryption are essential to facilitate real-time transactions and protect sensitive user data during transmission.

### 4.1.6 Compliance and Security:

Industry-standard security practices, including regular audits and updates, are essential to protect user privacy and ensure data integrity against emerging threats and vulnerabilities.

### 4.2 Software Requirements

Software requirements outline the software components and platforms that are required for the system's development and deployment. These requirements ensure that the system operates effectively within its software environment and leverages the appropriate technologies and tools for implementation.

### 4.2.1 Programming Languages and Frameworks:

Software requirements include Python with django for backend development, and HTML, CSS, and JavaScript with potential frameworks like React or Angular for frontend development to enhance user interface functionality.

### 4.2.2 Integrated Development Environment (IDE):

Visual Studio Code or any other preferred IDE for coding and project management.

### 4.2.3 Database Management System:

Using a database structure such as SQLlite3 or MySQL to store user data, preferences, and other relevant information.

### 4.2.4 Computer Vision and Machine Learning Libraries:

Software requirements include integration with computer vision libraries such as OpenCV for image analysis and feature extraction, and machine learning libraries such as TensorFlow or PyTorch for recommendation algorithms and modeling.

### 4.2.5 Image Processing Tools:

Image processing tools to increase the quality and efficiency of user-centered clothing recommendations.

### 4.3 Functional Requirements

Functional requirements define the specific functions and features that the system must possess to meet the needs of the users. These requirements serve as the basis for system design and development. These requirements describe the system’s behavior and its interactions with the users, also describing the tasks that should be performed, data that should be processed, and outputs that should be generated.

### 4.3.1 User Profile Management

Users should be able to create and manage their own profiles, including personal profiles such as skin color and facial structure.

### 4.3.2 Clothing Recommendation System

The system should generate personalized clothing recommendations considering the user's skin tone, facial structure, selected event type (day or evening), color and style preferences.

### 4.3.3 Color Combination Suggestions

Users should have the ability to input their color code and receive color combinations suggestions based on their skin complexion and selected color.

### 4.3.4 Event-Specific Outfit Recommendations

The system should provide clothing recommendations tailored to specific events, considering factors such as the type of event and style preferences of the user.

### 4.3.5 Integration of Computer Vision and Machine Learning

The system should incorporate computer vision algorithms to analyze user features and clothing characteristics, while machine learning algorithms analyze user data to increase the accuracy of recommendation accuracy, spot trends, and recognize new fashion trends.

### 4.3.6 User Feedback Mechanism

Users should be able to provide feedback on recommended outfits, helping the system improve its recommendations over time.

### 4.4 Non-Functional Requirements

Non-functional requirements specify the attributes and performance characteristics that the system must exhibit beyond its functional capabilities. These requirements ensure that the system meets the user’s expectations in terms of efficiency, security, and user experience that contributes to its overall effectiveness and value.

### 4.4.1 Performance

The system must maintain an acceptable response time for user interactions and handle multiple simultaneous users without any degradation of performance, ensuring a smooth and seamless user experience.

### 4.4.2 Scalability

The system should be able to accommodate the increasing user traffic and data volume.

### 4.4.3 Security

Systems must implement strong security measures to protect user data, including encryption of sensitive information and secure authentication mechanisms.

### 4.4.4 Accuracy

Recommendation systems should aim to maintain a high level of accuracy, tailoring recommendations based on user characteristics, preferences, and changing fashion trends.

### 4.4.5 User Interface

The user interface should prioritize responsiveness and aesthetics by incorporating accessibility features to enhance usability.

### 4.4.6 Reliability

The system should be reliable and available for use without frequent downtime.

# CHAPTER 5

## SYSTEM ARCHITECTURE

The system architecture for a "Persona Styler" software project consists of interconnected elements that work in concert to achieve the objectives that have been described in Chapter 1. It acts as a conceptual framework describing the behavior, structure, and operation of the system. The core program revolves around providing personalized fashion recommendations to users through a web-based application. Modified machine learning algorithms and system characteristics analysis in the system are supported to help users select the user and paths that are most important to their specific asset are being activated in the intermediate plates. Individually perform tasks such as image analysis and implementation to provide recommendations.

### 5.1 Components of the System Architecture

The main components of the system architecture for “Persona Styler” are as follow:

### 5.1.1 User Interface (UI) Component

This feature constitutes the user-facing portion of the application, providing users with a simple web-based interface for interaction. It provides functionality such as user authentication, personal attribute input (such as skin color, facial settings, color preferences, and action type input), and personalized recommendations they look after the UI side to ensure a seamless and intuitive experience for individuals seeking fashion advice.

### 5.1.2 Web Server Component

The Web Server Component of the "Persona Styler" application acts as the backbone of the application, receiving and managing a web-based interface that processes incoming HTTP requests Provides seamless communication between users and the system, serving static content such as HTML, CSS, JavaScript files for the UI, as well as f Directs dynamic requests to back-end services responsible for processing user input and producing styling recommendations personalized types This feature ensures the smooth functioning of the application interface and contributes to an enhanced user experience.

### 5.2 Backend Services

There are several modules which are responsible for different tasks which are related of the backend services:

### 5.2.1 User Data Handling Module:

This module manages user input, including skin color, facial structure and color preferences. It ensures seamless transfer of user data to subsequent modules for analysis and recommendation generation.

### 5.2.2 Machine Learning Recommendation Engine:

At the core of the backend services is a recommendation engine powered by machine learning algorithms. This module analyzes user data to identify trends, patterns and preferences, enabling personalized clothing recommendations. Utilizing techniques such as collaborative filtering and content-based filtering, the recommendation engine ensures tailored recommendations that match the characteristics and style of individual users.

### 5.2.3 Computer Vision Analysis Module:

Working with a recommendation engine, the computer vision analysis module extracts feature from the user’s image, such as skin tone, facial structure, color attributes and advanced computer vision techniques with image segmentation and feature extraction in addition to implementation, this module enhances the accuracy and usefulness of the fashion recommendations delivered to the users.

### 5.2.4 Color Combination Generation Module:

This module focuses on suggesting color combinations based on the user’s selected color and skin tone. By applying algorithms that analyze color theory principles and user preferences, it suggests compatible colors to match the user’s color with the selected color, and thereby increasing the overall appeal of the recommended outfits.

### 5.3 Sequence Diagrams

A sequence diagram is a type of interaction diagram. It describes how and in what order a group of objects works together. In order words, it illustrates the interactions between different components of the system during specific user actions or system operations. Sequence diagrams are widely used by software developers to understand the requirements of a new system or the document of an existing process.*Figure 5.1 shows the Sequence Diagram of system.*

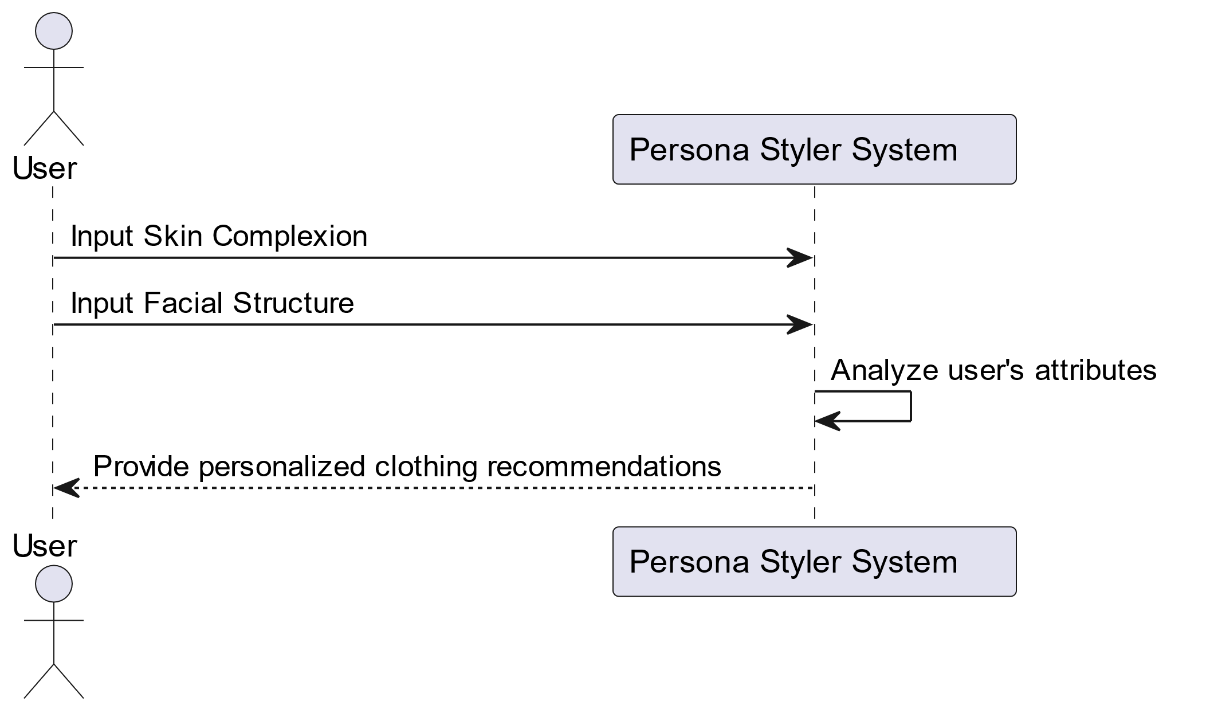
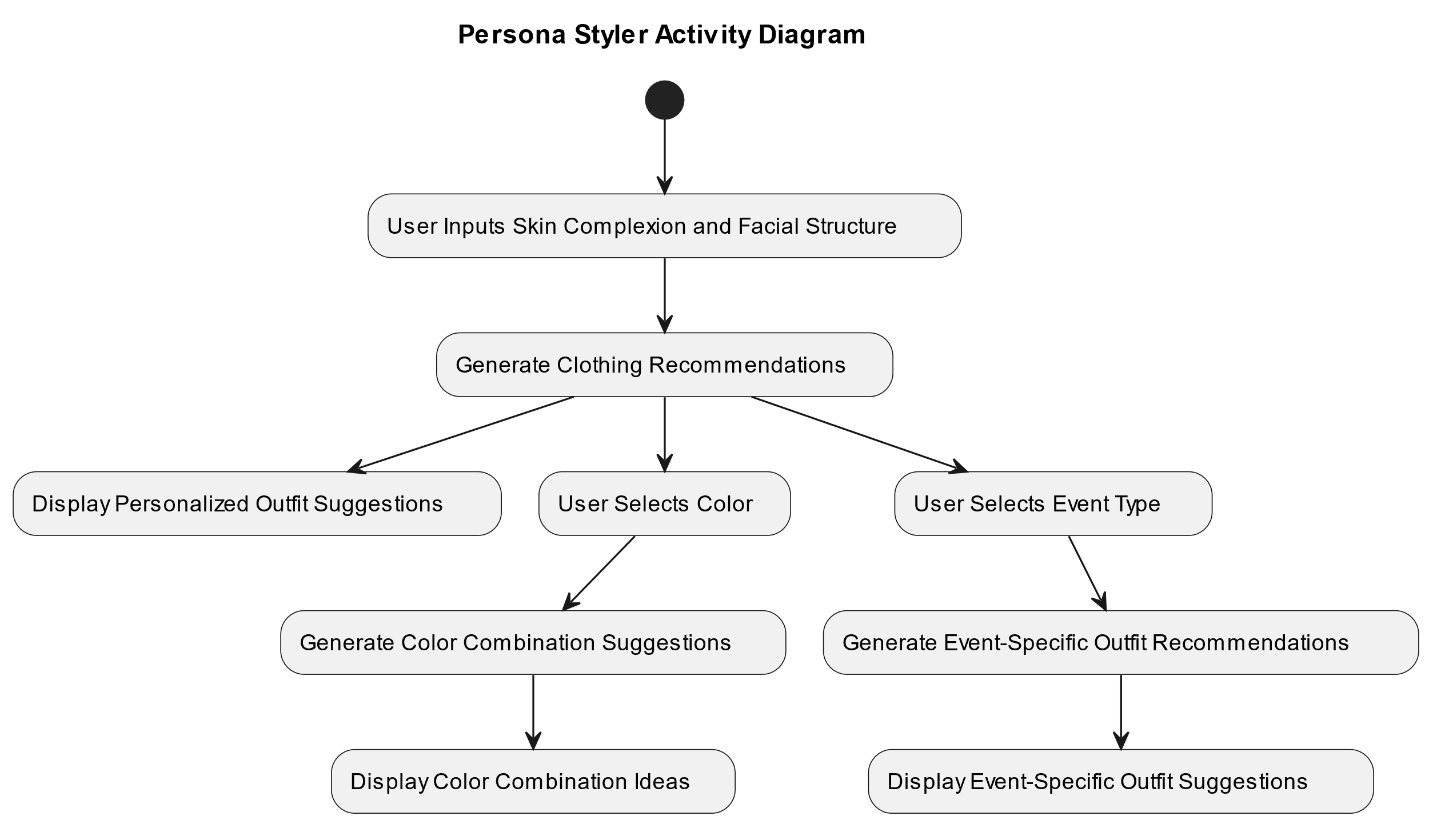


Figure 5.1: Sequence Diagram

### 5.4 Activity Diagram

Activity diagram is a graphical representation of workflows or processes, showing the sequence of actions or steps involved and their dependencies, facilitating visualization and analysis of system behavior and interactions.

Figure 5.2 shows the activity diagram.



*Figure 5.2: Activity Diagram*

# CHAPTER 6

## SYSTEM IMPLEMENTATION

The technical implementation details of the system are covered in this section. It explains the rationale behind our selection of a specific technology, the elements that comprise the system, and how everything is set up to best meet the needs of your setup. In the first part, we will cover the programming language we selected for the mobile application and the reason for its choice. We will then talk about the programming language we chose for our system's hardware setup.

### 6.1 System Tools and Technology

### 6.1.1 Python:

Python is a high-level, object-oriented, general-purpose programming language. Python is recognized for being user-friendly, having an extensive library and its simplicity that makes it ideal for both amateur as well as professional developers. One can blink the bulbs or flash the LEDs; it means that using some of the libraries and frameworks in Raspberry Pi’s vast ecosystem, one can access the device’s GPIO pins, manipulate actuators and sensors or communicate with numerous hardware components.

### 6.1.2 Django Firebase

Developers can create complex websites with ease thanks to Django - a high-level Python web framework. This simplifies development by structuring common web development tasks. In Django, there are Models, Views, Templates Forms and Admin Interface available. Django comes bundled with lots of features hence no need to write repetitive code.

### 6.1.3 Visual Studio Code (For Website Development)

Visual Studio Code (VS Code) is a powerful web development code editor specifically designed for. It is light weight, gratuitous cost wise and cross-platform; compatible with Linux OS X and windows operating system. VS Code will keep growing along with your experience in web programming thus making it a flexible tool. By adding more extensions for specific jobs like libraries and frameworks to VS Code will increase its usefulness further still.

### 6.2 Related Technology

### 6.2.1 Computer Vision

Computer vision is the branch of artificial intelligence (AI) that enables machines to perceive, interpret and understand visual information. This technology believes in what it sees, classifies accurately, and does this using digital photos and videos captured by cameras, along with deep learning models. The things this can do are generally as follows:

* **Image classification:** This task determines whether scenes or objects exist in an image. For example, a computer vision system may be trained to determine if an image contains cars, dogs or cats.
* **Object detection:** This is finding and identifying things from a picture or video. As an illustration, a computer vision system might be instructed to identify pedestrians in video streams.
* **Image segmentation:** An image is divided into several regions for this assignment each representing different object or parts of the scene that should be identified. For instance, a computer vision system may learn how to define separate areas of a person’s head, body, arms and legs in an image of that person
* **Facial recognition:** It involves recognizing someone from their picture or video. Face recognition technology is used by various sectors like security and surveillance among others.

### 6.2.2 Machine Learning

Artificial intelligence is a kind of machine learning that lets computers learn without having been explicitly programmed. It involves training algorithms on large volumes of data to recognize patterns and make predictions. These can use this for various purposes, such as sorting spam messages or encouraging purchases on e-commerce websites.

Here we have applied Supervised Learning in this project. A kind of learning called supervised learning, where the model is trained on labeled data. In other words, data points come with labels indicating what they represent like what was done in this Recommender System constructed using images based on facial structure and body type. After that it will be able to classify new unseen images.

### 6.2.3 Image Processing

## Image processing refers to manipulating and analyzing digital images, which entails the study of computer vision applications at large. Mainly we took out Image Processing key concept called Information Extraction: Objects can be recognized; distances computed as well as text within an image read through image processing.

### 6.3 Class Diagram

Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide.

Figure 4.6 shows the class diagram.



*Figure 4.6: class Diagram*

# CHAPTER 7

## SYSTEM TESTING

After the successful implementation of the system, the next step in system developing cycle is system testing. Before testing anything of the world, the tester should know about all the characteristics and specification of the system. This phase in system developing cycle is very important as, it is clearly used to verify the systems error by testing each and every parameter of the system. The primary goal of system testing is to validate the performance, accuracy, and reliability of the system.

## 7.1. Black Box Testing

The type of testing in which the tester may don’t have the knowledge of code and programming and tester only interacts to the user interface of the system for testing is called black box testing. The tester can or cannot be a developer in this type of testing. Here he doesn’t have to worry how the inputs and their corresponding outputs are generated. They must give inputs according to the test cases and then validate the test cases according to their corresponding outputs.

A screenshot of a computer

Description automatically generated with medium confidence

*Figure 7. 1 Black Box Testing*

### 7.1.1. Exhaustive testing

In it, we try to give all possible inputs to the system and then test the system’s behavior in accordance with those inputs. For example, if a function takes 1-50 integers as input, then in Exhaustive testing, we will give all these inputs one by one to test the system.

### 7.1.2. Equivalence Class Testing

In this method, we divide the input of the Module, which is being tested in two different portions. One portion is called “valid input” which includes all those inputs which are according to the defined Criteria of inputs for that system while the other portion is called “invalid input” which includes all these inputs which are not in accordance with the criteria of conditions defined for the inputs of that Module. Then we test that module also for valid and for invalid inputs.

### 7.1.3. Boundary Value Testing

## Boundary Value testing or BV testing is basically the up gradation of equivalence class testing. To apply boundary value testing to the objectives of developing a personalized fashion app, we'll focus on identifying, selecting, and executing test cases that examine the edge cases of input values. Here are three steps of boundary value testing tailored to the key objectives of the project

## 7.1.3.1 Identify the Boundaries

**Objective: Personalized Outfit Suggestions**

* **Boundary for Body Shape Input:** If body shape categories include Petite, Average, and Plus Size, the boundaries are at the transitions between these categories.
* **Boundary for Style Preferences:** If style preferences are scored on a scale from 1 to 5, the boundaries are 1 (min) and 5 (max).

**Objective: Color Combination Suggestions**

* **Boundary for Skin Complexion:** If skin complexion types are Fair, Medium, and Dark, the boundaries are at the transitions between these types.
* **Boundary for Color Selection:** If color selection is based on a predefined list, the boundaries are the first and last colors in the list.

## 7.1.3.2 Test Cases

 **Test Case 1 (Body Shape):**

* **Input**: Petite, Average, Plus Size
* **Boundary Values**: Petite to Average, Average to Plus Size

 **Test Case 2 (Style Preferences):**

* **Input**: 1, 3, 5
* **Boundary Values**: 1 (min), 5 (max), and values just inside (2, 4) and outside (0, 6) the boundaries
* **Test Case 3 (Skin Complexion):**
  + **Input**: Fair, Medium, Dark
  + **Boundary Values**: Fair to Medium, Medium to Dark
* **Test Case 4 (Color Selection):**
  + **Input**: First color in the list, Last color in the list
  + **Boundary Values**: Colors just inside and outside the list (e.g., if list is ["Red", "Green", "Blue"], test with "Red", "Blue" and invalid color "Yellow")

## 7.1.3.2 Execute and Evaluate

 **Execute Test Case 1:**

1. Test with inputs at the boundaries

* **Expected Result:** The app should handle transitions correctly and provide suitable outfit suggestions for these edge cases.

 **Execute Test Case 2:**

* Test with minimum (1) and maximum (5) values, and values just inside (2, 4) and outside (0, 6) the boundaries.
* **Expected Result:** The app should accept values within the range and provide appropriate style suggestions, while rejecting values outside the range.

 **Execute Test Case 3:**

* Test with inputs at the transitions: Fair-Medium and Medium-Dark.
* **Expected Result:** The app should provide accurate color combination suggestions for these edge cases.

 **Execute Test Case 4:**

* Test with first and last colors in the list and colors just outside the list.
* **Expected Result:** The app should accept valid colors and provide suitable combinations, while rejecting invalid colors.

## 7.2 Adopted Methodologies

We developed and reviewed the code while implementation and exception testing again and again, so we do not have review the code through white box testing. So according to the size, complexity, and type of this system, the methodology adopted for testing is: “Black Box Testing”

In Black Box Testing, the system is precisely tested at different levels as:

### 7.2.1. Unit Testing

Firstly, the whole system is tested in terms of unit testing by testing each specific module of the whole system with respect to functionality. It is regarded in terms of System verification. After testing all modules shifted to the next level of testing. Automated unit testing would run each unit or each small chunk of code in an automated, isolated manner.

### 7.2.2. Integrated Testing

After completing unit testing the next level of testing is called module testing where we tested dependent systems or modules at one time. It is regarded in terms of System verification. After completing this level, we shifted to the next level of testing.

### 7.2.3. System Testing

After completing Module testing of the whole system, we shifted to the next level of testing which is subsystem testing in which we test overall components specifically where each the component contains relevant modules. It is regarded in terms of System verification. After testing all components of the system, we shifted to the next level of testing.

### 7.2.4. Acceptance Testing

It is the last stage of testing and system development process where the system is validated by Stakeholders. The acceptance testing will be done by the Stakeholders to be sure whether the developed system is exactly according to the Expectations. After successfully passing the Acceptance test the System will be deployed in a very short time. This last stage of testing is regarded in terms of System validation.

### 7.3. Test Cases

Test cases are written in the testing phase before the implementation of the system that what are we going to test in the system. Test cases contains the expected input, output, and behavior of the system. While writing test cases we consider that what are the inputs and corresponding outputs according to the testing condition. Test cases usually written in the form of tables which contains the id of the test case, name of the test case and other terms are given as follows

* **Component Name**: to mention the name of the component.
* **Module Name**: to mention the name of the Module.
* **Condition Being Tested**: to provide the detail that we want to test.
* **Expected Result**: to check the expected results according to the condition.
* **Success Scenario**: if the test gives a positive result, then on what scenario or parameters this the result came, is mentioned here.
* **Failure Scenario**: if the test gives a negative result, then on what scenario or parameters this the result came, is mentioned here

### 7.3.1. Test Case for User Sign Up

Table 7.3.1 to 7.3.5 shows the detail of test cases of user.

*Table 7.3.1 Description of User Test Case*

|  |  |
| --- | --- |
| **Test Case ID** | 1 |
| **Module Name** | Persona Styler App |
| **Component Name** | User Sign Up |
| **Condition being tested** | The successful creation of a user account when valid information (username, email, and password) is provided. |
| **Expected Result** |  The app should create a new user account with the provided information.   Upon successful account creation, the user should be redirected to the app's home screen or a confirmation page.   Verify that no error messages are displayed indicating any issues with the sign-up process. |
| **Success Scenarios** |  The user account is successfully created without any errors.   The user is on the sign-up screen of the app. |
| **Failure Scenarios** |  The account creation fails due to errors or issues (e.g., invalid data format, server error).   Error messages appear indicating a problem with the sign-up process. |
| **Test Result** | Pass |

### 7.3.2. Test Case for User Login

*Table 7.3.2 Description of User Test Case*

|  |  |
| --- | --- |
| **Test Case ID** | 2 |
| **Module Name** | Persona Styler App |
| **Component Name** | User Login |
| **Condition being tested** | condition being tested is the successful authentication and access control mechanism of the app when a user attempts to log in. |
| **Expected Result** |  The app should verify the entered credentials.   If the credentials are correct, the user should gain access to personalized features.   Ensure the app navigates the user to the home screen or another appropriate screen upon successful login.   No error messages should be displayed indicating any issues with the login process. |
| **Success Scenarios** | The user successfully logs into the app without any errors. |
| **Failure Scenarios** | * The login fails due to incorrect credentials or other issues. * Error messages appear indicating a problem with the login process. |
| **Test Result** | pass |

### 7.3.3. Test Case for Providing Clothing Recommendations

*Table 7.3.3 Description of User Use Case*

|  |  |
| --- | --- |
| **Test Case ID** | 3 |
| **Module Name** | Persona Styler App |
| **Component Name** | Provide Clothing Recommendations |
| **Condition being tested** | App to accurately generate and present personalized clothing recommendations based on the user's provided inputs (skin complexion, facial structure, and preferences) |
| **Expected Result** |  The system should successfully generate personalized clothing recommendations based on the provided inputs.   The recommendations presented should be relevant to the user's skin complexion, facial structure, and selected preferences.   Ensure that the app navigates the user to a screen displaying the recommendations.   No errors should occur during the recommendation generation process. |
| **Success Scenarios** |  The system successfully generates personalized clothing recommendations based on the user's input (skin complexion, facial structure, preferences).   The recommendations presented are relevant and aligned with the user's attributes and selected preferences (e.g., casual style for casual preference).   The app navigates the user to a screen displaying the recommendations without any errors or interruptions.   No performance issues or significant delays occur during the recommendation generation process. |
| **Failure Scenarios** |  The system fails to generate personalized clothing recommendations despite valid user input.   The recommendations presented are not relevant or do not align with the user's attributes and preferences.   The app crashes, freezes, or encounters significant performance issues during the recommendation generation process. |
| **Test Result** | pass |

### 7.3.4. Test Case for Providing Color Combination Suggestions

*Table 7.3.4 Description of User Use Case*

|  |  |
| --- | --- |
| **Test Case ID** | 4 |
| **Module Name** | Persona Styler App |
| **Component Name** | Provide Color Combination Suggestions |
| **Condition being tested** | Verify that the Persona Styler app accurately provides color combination suggestions based on user input of selected color and skin complexion. |
| **Expected Result** |  The app should correctly accept and display the user's inputs.   The app should provide accurate and complementary color combination suggestions based on the input color and skin complexion. |
| **Success Scenarios** | Successfully provides accurate color combination suggestions based on the user's selected color and skin complexion. |
| **Failure Scenarios** | Invalid inputs or system errors while attempting to provide color combination suggestions. |
| **Test Result** | pass |

# CHAPTER 8

## APPLICATION PROTOTYPE

## 8.1 Interface

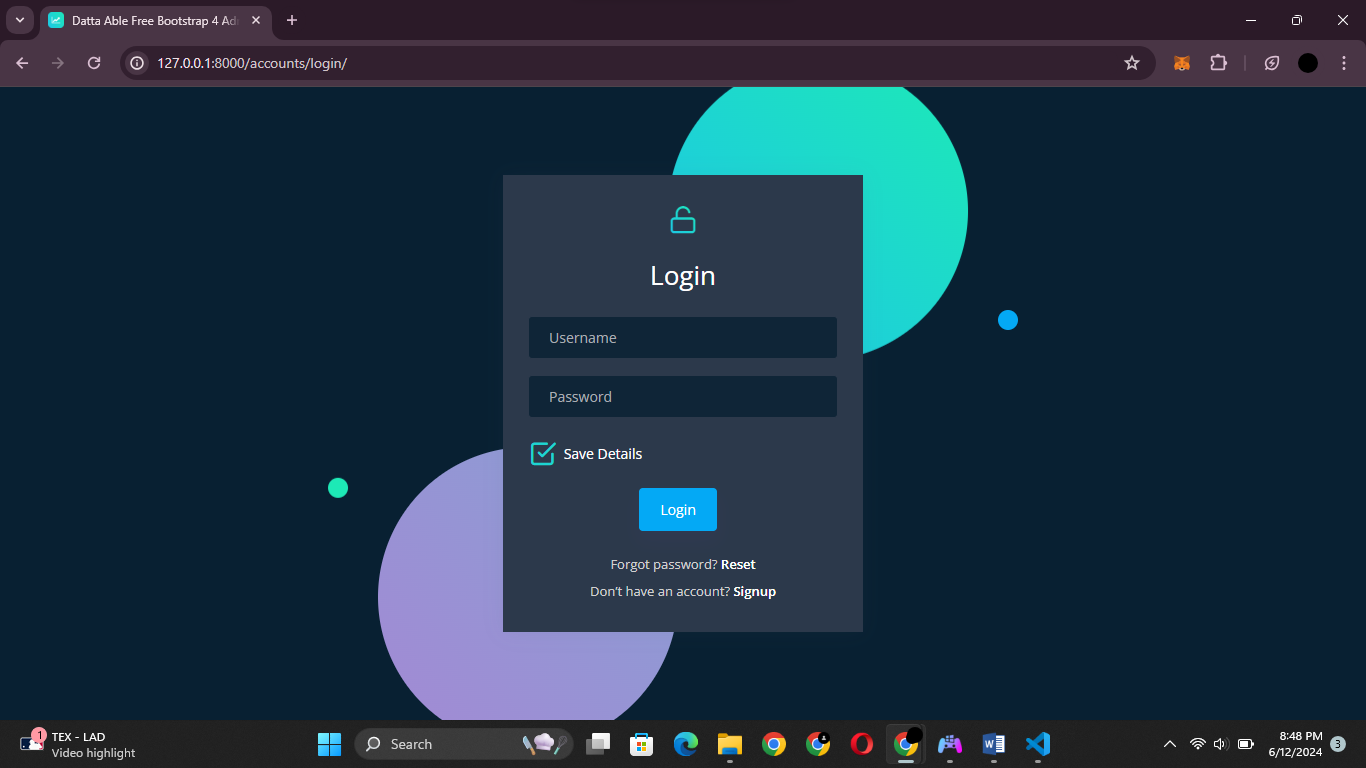
In this chapter, we present the prototype of our application, which serves as a representation of our conceptual design and functionality. Prototyping is a crucial step in the development process, allowing us to visualize and refine the user interface, interactions, and overall user experience before full-scale development.

Below is the interface of our website’s interface design

:

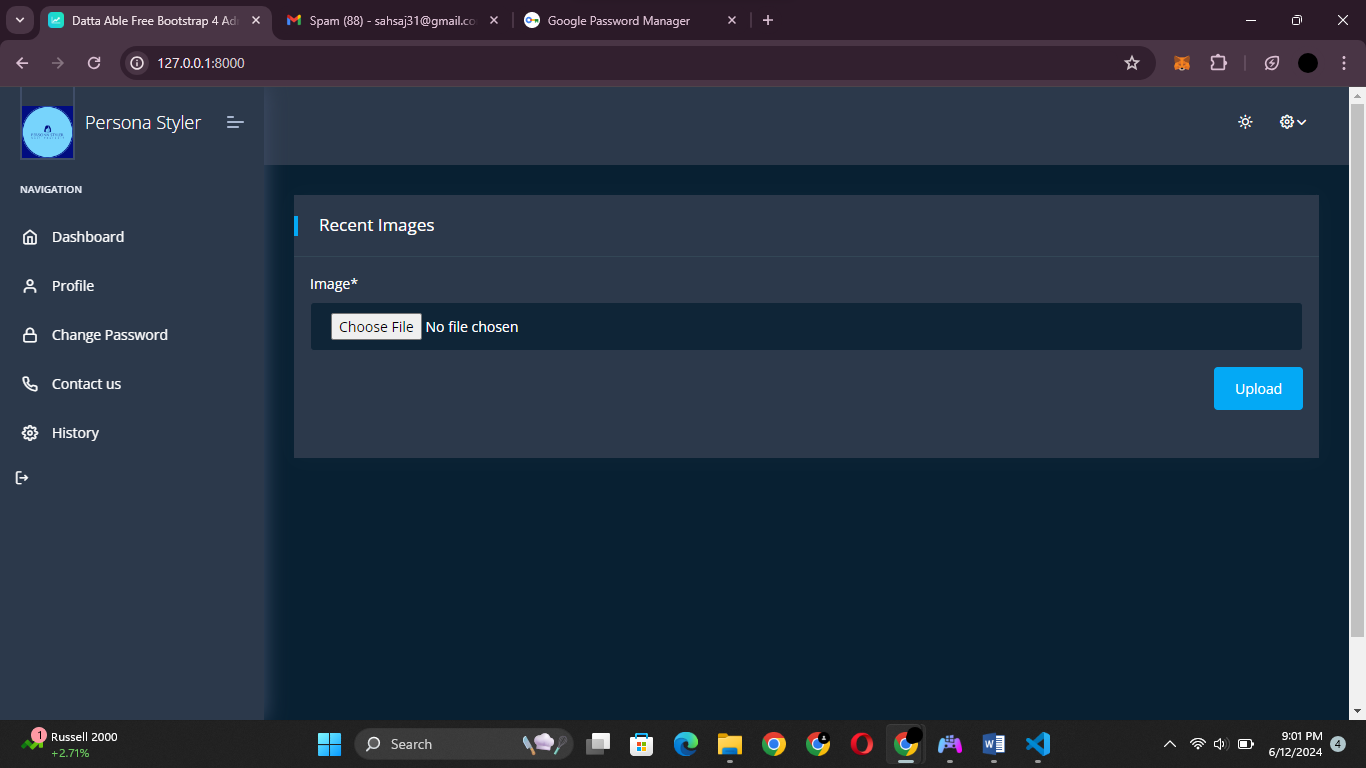
### 8.1.1. Login/Signup Page

Figure 8.1 shows the Login/Signup Page of application



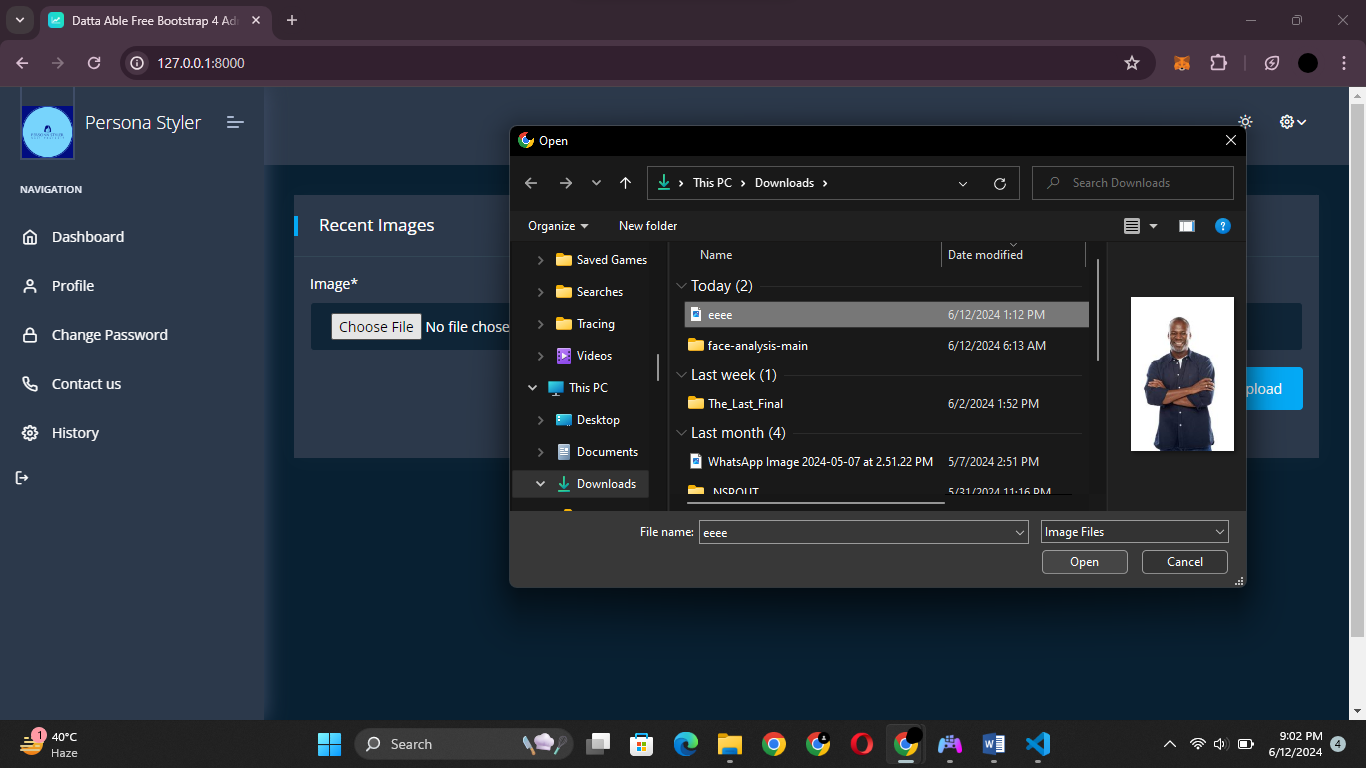
### 8.1.2. Dashboard

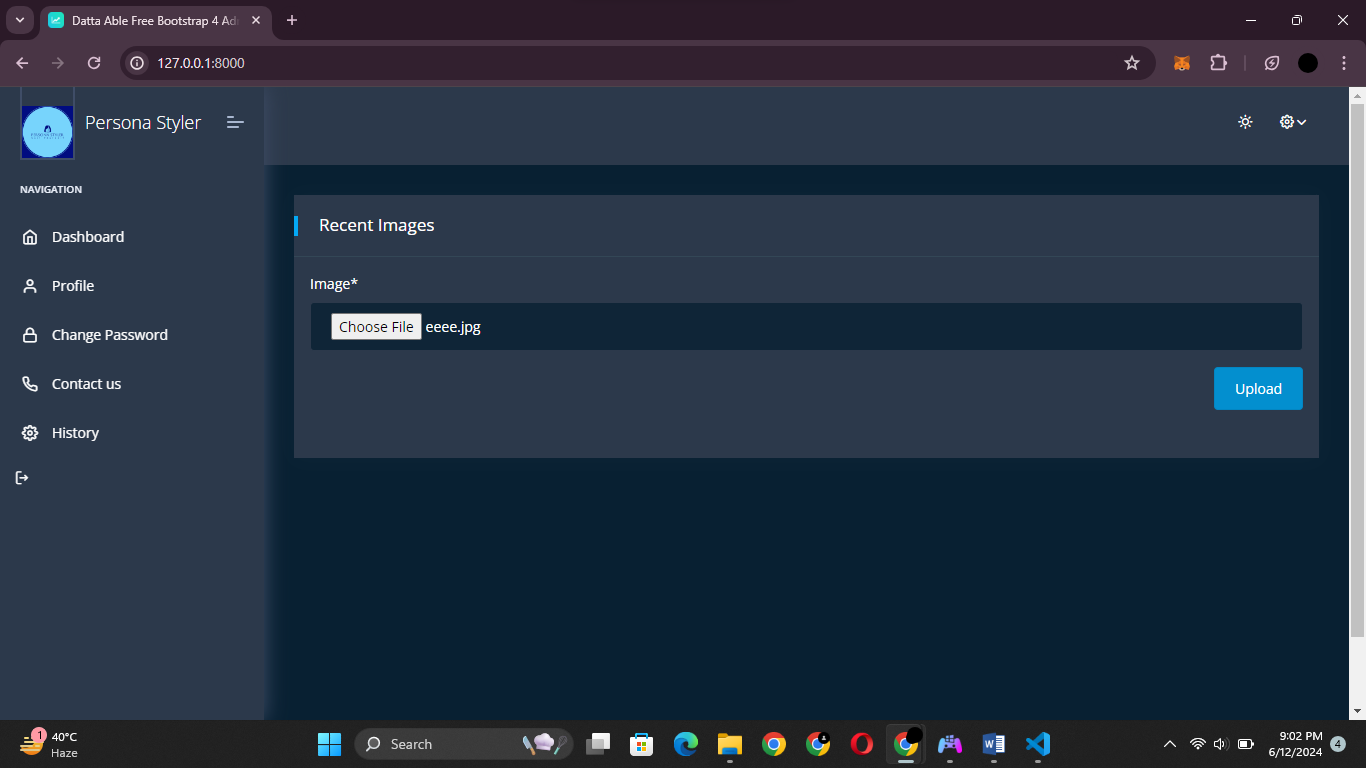
Figure 8.2 shows the dashboard



### 8.1.3. Adding Image For Analysis

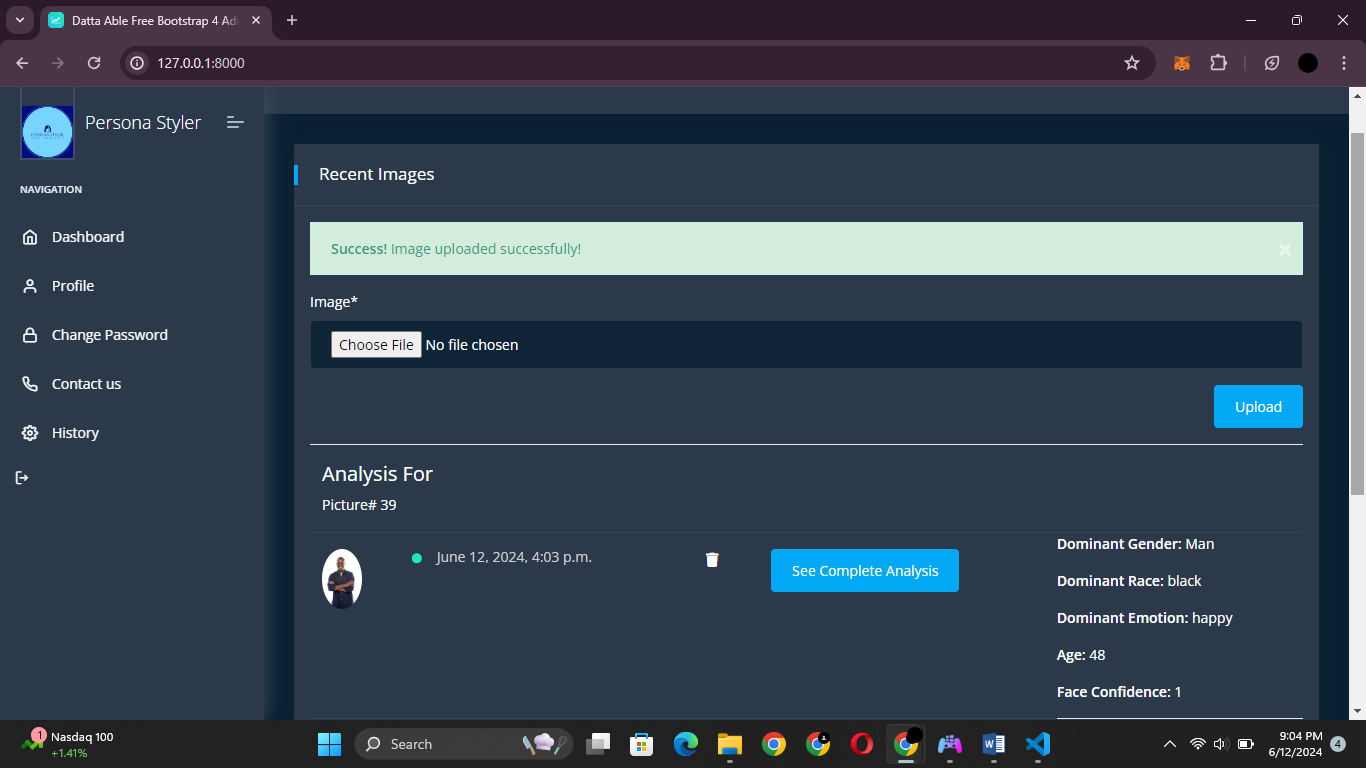
Figure 8.3 shows the Adding Image For Analysis





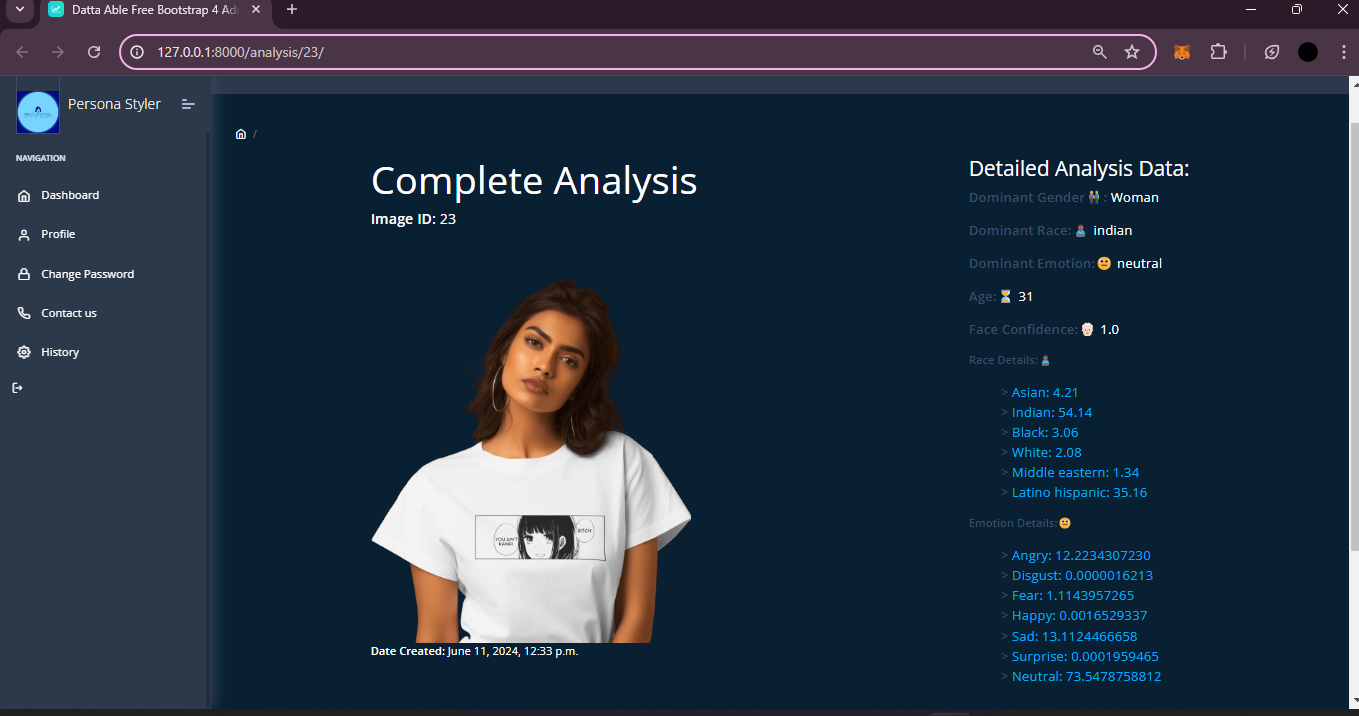
### 8.1.4. IMAGE Uploaded Successfully

Figure 8.4 shows the image upload successfully



### 8.1.5. Complete Analysis

Figure 8.5 shows the complete analysis



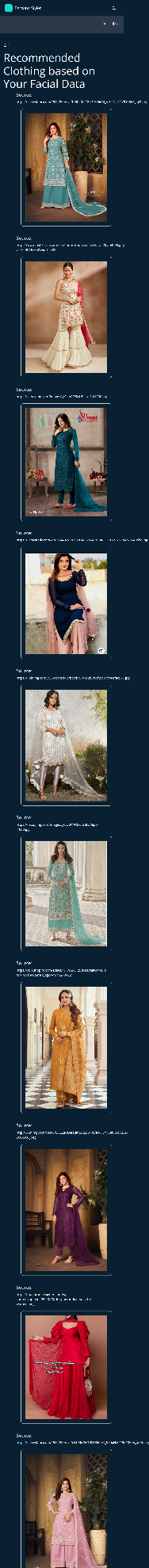
### 8.1.6. Feedback Analysis/Summary

Figure 8.6 shows the Feedback Analysis/Summary

### 

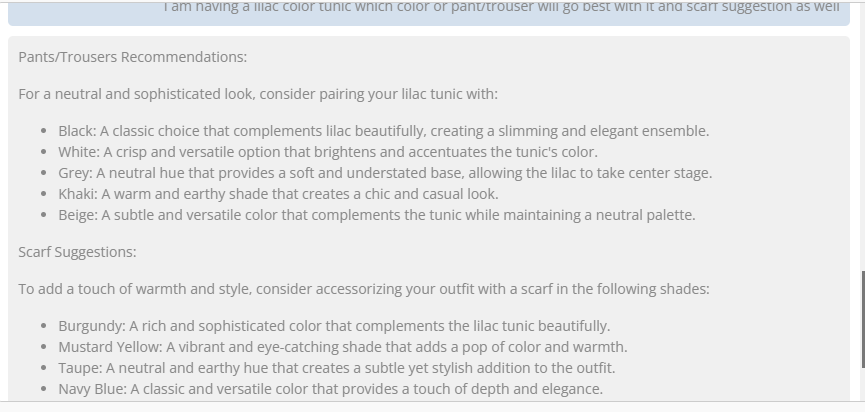
### 8.1.7. Recommended Clothing Type

Figure 8.7 shows the Recommended Clothing Type



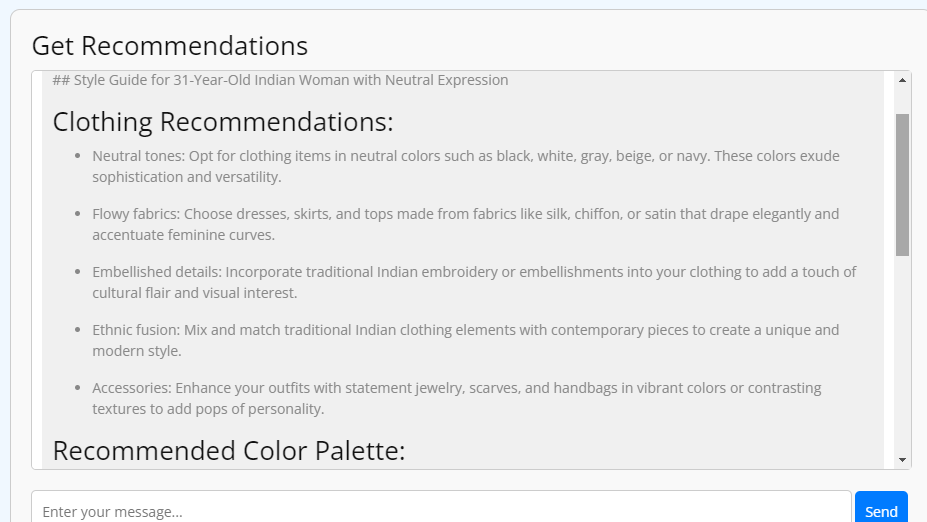
### 8.1.8. Color Suggestion Bot

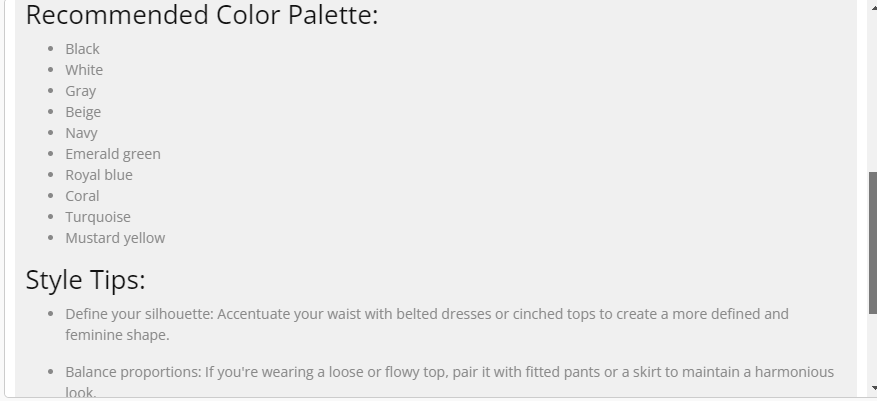
Figure 8.8 shows the color suggestion Bot

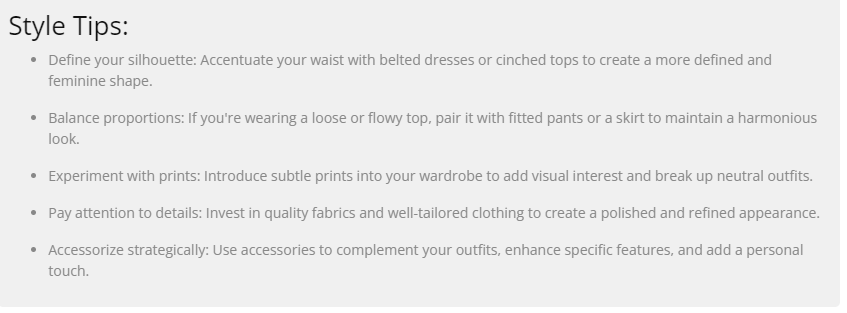


### 8.1.9. Recommendation Bot

Figure 8.9 shows the color suggestion Bot







### References

[1]

Koshy, R., et al. A Complexion based Outfit color recommender using Neural Networks. in 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). 2021. IEEE.

[2]

Dahunsi, B., et al., Review and synthesis of expert perspectives on user attribute and profile definitions for fashion recommendation. International Journal of Fashion Design, Technology and Education, 2023: p. 1-12.

[3]

Zong, W., Dress Style Recommendation Based on Female Body Shapes. 2022.

[4]

Chen, H., A. Gallagher, and B. Girod. Describing clothing by semantic attributes. in Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III 12. 2012. Springer.

[5]

Perrett, D.I. and R. Sprengelmeyer, Clothing Aesthetics: Consistent Colour Choices to Match Fair and Tanned Skin Tones. i-Perception, 2021. 12(6): p. 20416695211053361.

[6]

Nada, Y.A. and H. Meshref, Analysis, design, and implementation of intelligent expert system for clothes style selection. International Journal of Computer Applications, 2014. 105(4).

[7]

Kokane, C.D., et al., ML–Based Outfit Suggestion System. 2023.

[8]

Vogiatzis, D., et al., Expert and community-based style advice. Expert Systems with Applications, 2012. 39(12): p. 10647-10655.

[9]

Vogiatzis, D., et al., Exploiting knowledge about fashion to provide personalised clothing recommendations. 2009.

[10]

Wickramarathne, P., et al. Trenditex: An intelligent fashion designer. in 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). 2019. IEEE.