AI-POWERED VIRTUAL GARMENT TRIAL ROOM USING AUGMENTED REALITY

CONTENTS

ABSTRACT

The AI-Powered Virtual Garment Trial Room is an innovative solution designed to enhance the online shopping experience by enabling users to virtually try on apparel and accessories. The project addresses a major limitation of e-commerce: the inability to physically try products before purchase. Using augmented reality (AR) technology and advanced image processing, the system captures the user's image via a webcam and superimposes selected garments and accessories onto their body in real-time.

The system leverages Haar cascade datasets for body and face detection and convolutional neural networks (CNNs) for accurate alignment of apparel. The Flask framework integrates the back-end Python scripts with an interactive HTML front-end, allowing seamless user interaction. Users can register, shop, and virtually try on items, while administrators can manage the product catalog through an intuitive interface.

This cost-effective solution eliminates the need for expensive hardware, relying instead on efficient software tools like OpenCV and Dlib. Future enhancements include the integration of advanced networks, such as Pose Alignment Network (PAN) and Texture Refinement Network (TRN), to improve accuracy and realism. By bridging the gap between physical trials and online shopping, this project promises to revolutionize the e-commerce industry and enhance customer satisfaction.

Keywords: AI-Powered, Virtual Garment Trial Room, Online Shopping, Augmented Reality (AR), Image Processing, Real-time Superimposition, Convolutional Neural Networks (CNNs), Pose Alignment Network (PAN), Texture Refinement Network (TRN), Customer Experience, Virtual Try-On.

LIST OF FIGURES

LIST OF TABLES

LIST OF SCREENS

1. INTRODUCTION

1.1 MOTIVATION

The AI-Powered Virtual Garment Trial Room using Augmented Reality (AR) is inspired by the challenges of online apparel shopping, where customers cannot physically try on garments before purchase, leading to dissatisfaction, returns, and cancellations. This project aims to address these issues by leveraging augmented reality and image processing techniques to provide a virtual trial experience. Initially conceptualized from a problem statement in the Smart Gujarat Hackathon, the project seeks to offer an affordable and accessible alternative to expensive hardware-based solutions, such as Kinect motion sensors. By combining innovative technologies with user-centric design, it enhances customer confidence and satisfaction in e commerce platforms.

1.2 OBJECTIVE OF PROJECT

The main objective of this project is to provide an augmented reality-based solution for trying apparel and accessories online without the need for physical trials. This system reduces return rates and boosts customer satisfaction in e-commerce.

1.3 PROBLEM STATEMENT

The inability to try on apparel before purchase is a significant limitation of e-commerce platforms, leading to customer dissatisfaction, high return rates, and order cancellations. Traditional solutions, like Kinect motion sensors, are expensive and inaccessible for most users. This project aims to develop a cost-effective, AI-powered virtual garment trial room using augmented reality and image processing techniques. By allowing users to virtually try on garments and accessories through a webcam, the system enhances the online shopping experience, reduces return rates, and increases customer satisfaction, bridging the gap between physical and digital retail experiences.

1.4 SCOPE

The Virtual Trail Room project aims to develop a web-based platform that enhances online apparel shopping through virtual try-on features and efficient product management. Its scope includes:

- User Features: User Features: User registration, login, product browsing, cart management, and two try-on methods—AR-based (live video using OpenCV and Dlib) and image-based (via Segmind API).
- Admin Features: Secure login, adding, updating, and deleting clothing products with details (name, category, cost, image).
- Technical Scope: Implementation using Flask, MySQL for data storage, and integration of AR and image processing technologies.
- Deliverables: A functional web application with a basic frontend, backend logic, and database schema.

❖ **Inclusions:**

- Support for upper-body clothing try-ons (e.g., tops).
- Local file storage for product and user-uploaded images.
- Basic user and admin workflows.

❖ **Exclusions:**

- Full-body AR detection or support for lower-body clothing (e.g., pants).
- Advanced UI styling or mobile responsiveness.
- Payment gateway integration for checkout.
- Cloud deployment or scalability optimizations.

The project focuses on small to medium-scale apparel retail, targeting users and administrators seeking an interactive, proof-of-concept solution.

1.5 PROJECT INTRODUCTION

The Virtual Trail Room is an innovative web-based application designed to revolutionize the online shopping experience for clothing. By integrating augmented reality (AR) and image processing technologies, it allows users to virtually "try on" clothes before making a purchase, bridging the gap between physical and digital retail. This project aims to enhance user engagement, reduce return rates due to sizing or style mismatches, and provide an interactive platform for both customers and administrators.

Developed using Flask, a lightweight Python web framework, the application combines a robust backend with a simple, user-friendly frontend. It leverages MySQL for data management, OpenCV and Dlib for real-time AR capabilities. The system is divided into two primary modules: an Admin Panel for managing product inventory and a User Panel for shopping and virtual try-on experiences.

❖ **Purpose**:

The primary goal of the Virtual Trail Room is to empower users to make informed purchasing decisions by simulating the in-store fitting room experience online. For administrators, it provides an efficient tool to manage clothing products, including adding, updating, and deleting items with ease. The project addresses the common pain points of online apparel shopping—such as uncertainty about fit and appearance—through cutting-edge technology.

❖ **Key Features:**

- **Admin Capabilities:**
  - Secure login for administrators.
  - Add, update, and delete clothing products with details like name, category, cost, and images.
- **User Experience:**
  - User registration and authentication.
  - Browse a catalogue of clothes, add items to a cart, and manage cart contents.
  - Virtual try-on options:
  - AR Try-On: Overlay clothes on a live video feed using facial and upper-body detection.
  - Image Try-On: Upload a photo and see the selected clothing superimposed via an external API.
- **Technology Highlights:**
  - Real-time video processing for AR using OpenCV and Dlib.

- Seamless integration with a MySQL database for storing user and product data.

❖ **Target Audience:**

This application caters to online shoppers seeking a more immersive and confident buying experience, as well as e-commerce businesses looking to enhance their platforms with virtual fitting solutions. It is particularly suited for small to medium-scale apparel retailers aiming to adopt modern technologies without complex infrastructure.

The Virtual Trail Room is a step toward the future of e-commerce, blending convenience with interactivity. Whether you're a user exploring fashion options or an admin managing a digital store, this project offers a practical and engaging solution tailored to modern shopping needs.

1.6 OVERVIEW OF CORE TECHNOLOGIES

AUGMENTED REALITY (AR):

Augmented Reality (AR) is an interactive technology that enhances and enriches the real-world environment by overlaying digital content such as images, videos, sounds, 3D models, and data in real time. Unlike Virtual Reality (VR), which creates a fully immersive virtual environment, AR blends digital elements with the physical world, allowing users to interact with both simultaneously. AR applications typically use cameras, sensors, and software to recognize objects, locations, or surfaces in the real world and then superimpose contextually relevant digital information onto them through devices like smartphones, tablets, AR headsets, or smart glasses. This technology is widely used in various fields including gaming, education, healthcare, retail, and industrial design to provide immersive user experiences, enhance learning, improve productivity, and support complex tasks with real-time information.

KEY FEATURES OF AR

- Combines real and virtual environments
- Interactive in real time

- Registered in 3D (aligns with real-world dimensions)
- Enhances rather than replaces reality (unlike VR)

EXAMPLES OF AR

1. Pokémon GO (Gaming)

A mobile game that blends virtual Pokémon with the real-world environment, encouraging users to explore and catch Pokémon through their phone's camera.

2. IKEA Place (Retail)

An app that lets users visualize how IKEA furniture would look in their home before purchasing, by placing 3D models in real-world spaces using the phone camera.

3. Snapchat Filters (Social Media)

Snapchat uses AR to apply fun filters and effects to users' faces or surroundings in real-time, such as adding dog ears or changing hair color.

4. Google AR Animals (Search)

When you search for animals like lions or sharks on Google, it shows a 3D AR model of the animal, which you can place in your environment to see how it looks in real scale.

5. Sephora Virtual Artist (Beauty)

Sephora's Virtual Artist app allows users to try on makeup virtually, by applying makeup products to their faces using AR technology to see how they will look before buying.

VIRTUAL REALITY (VR)

Virtual Reality (VR) is a fully immersive technology that uses computer-generated simulations to create a realistic, interactive 3D environment that users can explore and interact with using specialized devices such as VR headsets, gloves, or motion controllers. In a VR experience, users are completely cut off from the physical world and placed into a digitally constructed environment, allowing them to feel as if they are "inside" a different world, whether it's a game, a simulation, or a virtual training space.

KEY FEATURES OF VR

- Immersive and fully virtual experience
- Interacts only with the digital world
- Requires VR headsets like Oculus, HTC Vive, PlayStation VR, etc.
- Often includes spatial audio and motion tracking for realism

EXAMPLES OF VR

1. Oculus Quest 2 (VR Headset)

A standalone VR headset that offers immersive experiences for gaming, fitness, and entertainment. With no need for a PC, users can experience VR games, apps, and movies in an entirely virtual environment.

2. Beat Saber (Gaming)

A VR rhythm game where players slice through blocks to the beat of the music using lightsabers. It combines music, action, and virtual reality for a highly immersive experience.

3. Google Earth VR (Virtual Exploration)

A VR version of Google Earth, allowing users to virtually travel anywhere in the world. You can explore cities, landmarks, and countries in 3D, as if you're flying over them.

4. VR Surgery Simulations (Healthcare)

VR is used in medical training to simulate surgeries. Medical professionals can practice surgical procedures in a risk-free environment using realistic, 3D simulations.

5. The VOID (Entertainment)

The VOID is a location-based VR experience that integrates physical environments with VR. Players can interact with physical objects while immersed in a fully virtual world (such as Star Wars: Secrets of the Empire).

ARTIFICIAL INTELLIGENCE (AI)

Artificial Intelligence (AI) is a branch of computer science that focuses on creating intelligent systems capable of performing tasks that typically require human intelligence. These tasks include learning from data, reasoning, problem-solving, understanding natural language, recognizing images and speech, and making decisions. AI systems are designed to analyze vast amounts of data, identify patterns, and adapt over time to improve their performance without being explicitly programmed for every task.

AI is broadly categorized into Narrow AI (specialized in a specific task like voice recognition or recommendation systems) and General AI (which aims to perform any intellectual task a human can do). Modern AI technologies leverage machine learning (ML), deep learning, neural networks, and natural language processing (NLP) to build smart applications across various industries such as healthcare, automotive, e-commerce, education, and entertainment.

EXAMPLES OF AI

1. Siri (Voice Assistant)

Siri is Apple's voice-activated AI assistant that performs tasks such as setting reminders, sending messages, and answering questions based on user voice commands.

2. Tesla Autopilot (Self-Driving Cars)

Tesla's Autopilot uses AI to enable semi-autonomous driving, allowing the car to make decisions like steering, braking, and accelerating in real-time.

3. Netflix Recommendation System

Netflix uses AI algorithms to analyze viewing habits and recommend movies and TV shows based on user preferences and previous interactions.

4. Chatbots (Customer Support)

AI-powered chatbots are used by businesses for customer service, answering queries, and providing support without human intervention.

5. Google Translate (Language Translation)

Google Translate uses AI to automatically translate text between different languages, improving the accuracy and fluency of translations over time.

WHY AI IS USED?

Artificial Intelligence (AI) is used in this project to enhance the accuracy, personalization, and intelligence of the virtual trial room system. AI enables body detection and pose estimation, helping the system understand the user’s body structure and measurements. This allows the virtual garments to be fitted realistically and accurately. AI also powers recommendation engines that suggest suitable outfits based on the user's preferences, body type, and trending fashion data. Furthermore, AI is used for fabric simulation, predicting how different clothing materials would behave on the user's body, such as stretching or flowing. Advanced gesture and facial recognition features driven by AI allow users to interact with the system naturally, such as changing clothes with a gesture or expression, making the experience smarter and more intuitive.

WHY AR IS USED?

Augmented Reality (AR) is used in this project to create an immersive and interactive virtual try-on experience. By using the device camera, AR overlays digital garments directly onto the user's live video feed, making it look like they are wearing the clothes in real time. This eliminates the need for physical trials and enhances safety and convenience, especially in online or contactless shopping environments. AR also enables users to view the outfit from multiple angles, helping them get a better sense of fit and appearance. The real-time nature of AR allows users to move, turn, and interact naturally, while still seeing the virtual clothes respond accordingly. This creates a highly engaging, fun, and futuristic way to try on clothes without stepping into a changing room.

KEY DIFFERENCES BETWEEN AR AND VR

AUGMENTED REALITY (AR)	**VIRTUAL REALITY (VR)**
AR enhances the real-world environment by overlaying digital elements such as images, 3D models, or animations in real time.	VR creates a completely artificial environment that immerses the user and blocks out the real world.
Offers a semi-immersive experience by blending virtual content with the real surroundings.	Provides a fully immersive experience where the user is entirely placed within a virtual world.
Depends on the physical world as a base and augments it with virtual content.	Entirely replaces the physical world with a digitally created simulation.
Requires smartphones, tablets, or AR glasses (e.g., Microsoft HoloLens, Magic Leap).	Requires VR headsets and motion controllers (e.g., Oculus Rift, HTC Vive, PlayStation VR).
Users can move freely in their physical surroundings while viewing virtual overlays.	User movement is limited to the virtual environment; physical movement may be restricted.
The real world remains visible and interactive.	The real world is completely blocked from view.
Virtual try-on, interior design previews, navigation, education, medical imaging.	Gaming, training simulations, virtual tourism, architecture walkthroughs, and therapy.

Table 2.1 - Key Differences B/W AR and VR

2. LITERATURE SURVEY

Ref. [1]: E. Nitasha et al. (2024)

In "Future of Fashion: AI-Powered Virtual Dressing for E-Commerce". This paper introduces a smart virtual dressing framework combining computer vision, gesture recognition, and intelligent recommendation. It allows users to interact using body movements and try garments virtually. The system enhances personalization, reduces returns, and provides fashion trend analysis.

Ref. [2]: Manjula Devarakonda Venkata et al. (2024)

In "AI-Enhanced Digital Mirrors". This research highlights AI-powered digital mirrors to improve shopping safety and hygiene, especially for women. The system enables virtual try-on using AR without needing physical fitting rooms. It also addresses privacy concerns, making it ideal for public spaces like malls.

Ref. [3]: B. S Rochana et.al (2024)

In "Virtual Dress Trials Using GANs". The paper leverages GANs for generating realistic virtual trials and includes size prediction and chatbot support. It enhances user experience with visual fitting and interactivity. The system reduces return rates and promotes eco-friendly shopping by minimizing physical inventory.

Ref. [4]: Qinghui Wang & Na Qu (2024)

In "Novel AI Model for Evaluating Clothing Fit Satisfaction". This work proposes a machine learning model using TSO-SLLR to predict how satisfied buyers are with clothing fit. It uses data from virtual and real fittings along with 3D body scans.

Ref. [5]: Xu et al. (2020)

Fashion Forward: A Deep Learning Approach for Fashion Understanding. The paper explores image recognition and fashion attribute classification using deep learning. It focuses on clothing segmentation and garment detection.

Ref. [6]: Ge et al. (2019)

Parser-Free Virtual Try-On via Distilling Appearance Flows. The paper removes the dependency on human parsing by learning appearance flows from paired images. It simplifies the pipeline, making the system faster and more efficient. The model offers high-quality garment fitting while saving computation time.

Ref. [7]: Dong et al. (2019)

FW-GAN: Flow-n-Warp GAN for Image-Based Virtual Try-On. This study introduces a GAN-based framework using dense flow estimation to warp clothes realistically over the target image. The network captures fine-grained textures and achieves accurate alignment. It offers a significant improvement in garment appearance realism in try-on systems.

Ref. [8]: Han et al. (2018)

M2E-Try on Net: Fashion from Model to Everyone. This paper proposes a model that transfers garments from fashion model images to user images using pose alignment. It utilizes deep convolutional networks to adapt clothing to new body shapes and orientations. The technique improves realism in virtual try-on scenarios. It’s a foundational work in AI-based fashion synthesis.

Ref. [9]: Chen et al. (2018)

VITON-HD: High-Definition Virtual Try-On Network. This improved version of VITON focuses on high-resolution output, delivering more detailed and visually appealing results. It preserves garment textures and enhances realism. It’s suited for commercial deployment where high visual quality is essential.

Ref. [10]: Zheng et al. (2017)

VITON: An Image-Based Virtual Try-On Network VITON presents a two-stage network for virtual try-on that first generates a coarse garment overlay and then refines it for detail. It maintains the user’s pose and body shape while fitting the garment. This model avoids 3D-modeling and is scalable for online applications.

Author	Year	Title	Outcome
E. Nitasha, S. Kumari, A. Kumar, R. Bhardwaj	2024	Future of Fashion: AI-Powered Virtual Dressing for E-Commerce Applications	AI-powered Virtual Dressing uses computer vision and machine learning to enhance online clothing shopping with realistic visualizations, accurate fit, and reduced returns, backed by positive consumer interest.
Manjula Devarakonda Venkata,	2024	AI-Enhanced Digital Mirrors: Empowering Women's Safety and Shopping Experiences	AI and AR-based digital mirrors enable virtual try-ons, minimizing physical fitting rooms and enhancing customer safety and privacy in malls by addressing traditional trial room security and usability issues.
B. S. Rochana and S. Juliet	2024	Virtual Dress Trials: Leveraging GANs for Realistic Clothing Simulation	A digitized fashion platform leverages GAN-driven virtual trials, advanced size recommendations, and AI chatbots to enhance online shopping. It ensures data security, supports reviews, and fosters e-commerce collaborations.
Qinghui Wang, Na Qu	2024	Novel AI Model for Evaluating Buyers' Fulfilment with Clothing Fit	A novel AI model uses 3D body scans and virtual fitting with TSO-SLLR machine learning to accurately predict customer satisfaction with clothing fit, enhancing online shopping experience and reducing returns.

Table 2.2 - Literature Search

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing systems for online apparel shopping primarily rely on static product images, size charts, and textual descriptions to assist users in making purchasing decisions. Advanced solutions, such as those implemented by large retailers like Amazon (e.g., Echo Look) and ASOS, incorporate virtual try-on features using augmented reality (AR) or image-based technologies powered by deep learning models like GANs. These systems often use mobile apps or specialized hardware to overlay clothing on users via AR or process uploaded photos for visualization.

3.2 DRAWBACKS OF THE EXISTING SYSTEM

- **High Cost:** Advanced AR and image-based systems require expensive hardware or cloud subscriptions, unaffordable for small retailers.
- **Complexity**: Full-body scanning and 3D-modeling demand significant computational resources and technical expertise.
- **Limited Accessibility:** Many solutions are mobile-app-based or hardware-dependent, excluding webcam-only users.
- **Scalability Issues:** Proprietary systems are not easily adaptable for small-scale or open-source use.
- **Lack of Real-Time Interaction:** Image-based try-ons often lack the immediacy of live AR, reducing user engagement.
- **Poor Fit Prediction:** Existing solutions may not accurately estimate garment fitting for diverse body types due to reliance on fixed size charts.
- **Privacy Concerns:** Systems that rely on cloud-based processing may raise user concerns about the storage and handling of personal images.

3.3 PROPOSED SYSTEM

The proposed Virtual Trail Room is a web-based application designed to democratize virtual try-on technology for online apparel shopping. Built using Flask, it integrates a lightweight AR module powered by OpenCV and Dlib for real-time clothing overlays on live video feeds, alongside an image-based try-on feature leveraging the Segmind Try-On Diffusion API for static photo processing. The system includes a dual-interface: an admin panel for managing products (add, update, delete) and a user panel for registration, browsing, cart management, and trying on clothes. Utilizing MySQL for data storage and a simple HTML frontend, it targets small to medium-scale retailers and users with basic hardware (e.g., a webcam), offering an accessible, interactive shopping experience without complex infrastructure.

The platform features a dual-interface:

- **Admin Panel:** Allows administrators to manage products, handle inventory (add/update/delete garments), and view customer activity.
- **User Panel:** Enables registration, login, browsing the product catalogue, cart management, and virtual trials.

3.4 ADVANTAGES OF THE PROPOSED SYSTEM

- **Cost-Effective:** Uses open-source tools and minimal hardware, reducing implementation costs.
- **Dual Try-On Modes:** Combines real-time AR and image-based options for flexibility and user preference.
- **Ease of Use:** Web-based platform requires no app installation, accessible via browsers.
- **Lightweight:** Optimized for basic systems (e.g., webcam and local storage), avoiding heavy computational demands.
- **Scalable for Small Retailers:** Simple admin tools and database integration suit small-scale operations effectively.
- **Privacy-Focused:** Images are processed securely without long-term storage, addressing user concerns regarding personal data.
- **Improved Engagement:** Real-time interaction boosts user engagement, reduces guesswork, and improves buying confidence.

- **Return Rate Reduction:** By visualizing fit and style beforehand, users are less likely to return products, improving seller profit margins.

3.5 WORK FLOW OF PROJECT

The workflow diagram of the AI-Powered Virtual Garment Trial Room outlines the logical sequence of operations from the moment a user accesses the platform to the point where virtual try-on is performed and data is managed. This project follows a modular and event-driven architecture, ensuring smooth interaction between frontend, backend, database, and AR/image-based processing systems.

The project workflow is divided into three major modules: Admin Workflow, User Workflow, and Try-On Options. Each module performs specific functions to ensure a smooth and interactive virtual garment trial experience.

1. USER INTERACTION LAYER (USER WORKFLOW)

Users access the platform through a web-based interface built using Flask and HTML/CSS.

- **Register & Login:** New users can sign up, and existing users can log in using email and password authentication.
- **Browse Products:** Users can explore the clothing catalogue, view available items, and add them to their shopping cart.
- **Add to Cart / View Cart / Remove from Cart:** Users can manage selected items for trial.

Once logged in, users can choose between two try-on options to view how garments would look on them.

2. PRODUCT CATALOGUE MANAGEMENT (ADMIN WORKFLOW)

All clothing products are maintained in a MySQL database, which includes product details like name, category (e.g., tops), image, and cost. Admin users can:

- **Add New Products:** Admins can add new clothing items, including product images, names, and prices.

- **Update Existing Products:** Modify details as necessary.
- **Delete Products:** Remove outdated or unwanted items.
- The changes made by the admin are reflected instantly on the user interface, ensuring up-to-date product availability.

3. TRY-ON FEATURE (TWO MODES)

The core functionality of the project lies in the virtual try-on experience, which is offered through two modes:

A. IMAGE-BASED TRY-ON:

- **Upload Photo:** Users upload a clear image of themselves.
- **Segmind API Processing:** The image is sent to a third-party try-on API (Segmind Try-On Diffusion).
- **Display Generated Image:** The processed image is returned, showing the user wearing the selected garment.

B. LIVE AR TRY-ON (REAL-TIME TRY-ON):

- **Live Video Try-On:** Users activate their webcam to access the real-time try-on module.
- **Detect Face/Upper Body**: Using OpenCV and Dlib, the system detects the user's upper body (face, jaw, shoulders) using Haar Cascades and 68-point landmark detection.
- **Overlay Clothing:** The selected garment is overlaid onto the live video feed using sprite overlay and alpha blending techniques, providing an interactive, real-time experience.
- **Display in Real-Time Tkinter Window:** The output is shown in a custom GUI window for a real-time fitting experience.

4. BACKEND PROCESSING & STORAGE

All user data, product information, cart contents, and try-on history are securely handled using Flask’s backend logic and stored in a MySQL database. Images are temporarily stored in local directories under /static/images/ and /saved_images/.

5. ADMIN MONITORING AND CONTROL

The admin has full control over product listings and can view activity logs. Admin interactions are kept separate from user functions for security and scalability.

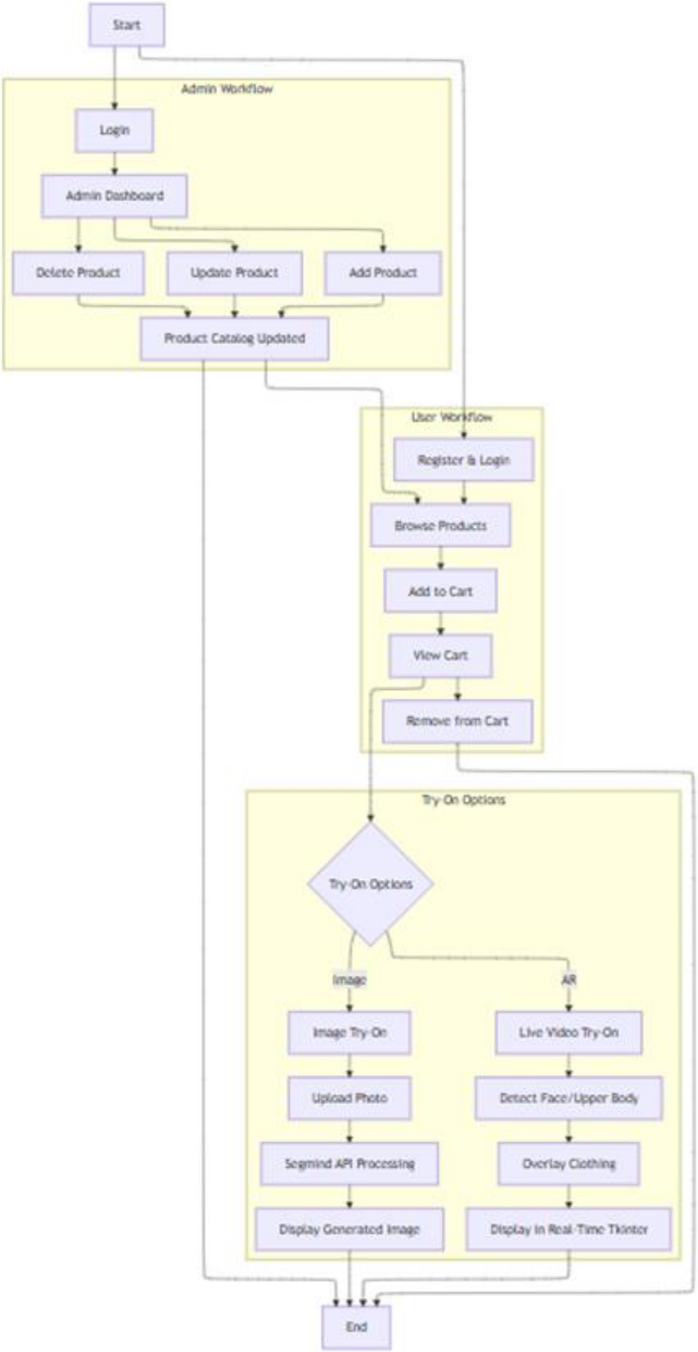


Figure 3.1: Work Flow of the Project

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

The process of requirements analysis is critical for assessing the success of a system or software project. These requirements help define the overall functionality and performance of the system. Requirements are generally categorized into Functional Requirements and Non-Functional Requirements, each focusing on different aspects of the system's behaviour and performance.

4.1.1 FUNCTIONAL REQUIREMENTS:

Functional requirements define the specific actions or functionalities that the system must perform. They are directly tied to the user's needs and describe how the system will behave under certain conditions. Functional requirements are typically more tangible and can be directly tested against the system. They provide a clear description of the system's expected behaviour, typically stated in terms of inputs, actions, and expected outputs.

These are generally considered as "what" the system should do and are closely tied to the features and capabilities that users will interact with. Functional requirements are often derived from the business requirements or user stories.

Examples of Functional Requirements:

1. **Authentication of User:** The system must verify the identity of the user upon login by checking their credentials against stored data.

2. **User Registration:** A new user should be able to create an account by providing necessary details, with the system validating and storing this information.

3. **Email Verification:** When a user registers for the first time, the system sends a verification email to their registered email address to confirm their identity.

4. **Product Search:** The system should allow users to search for specific products using predefined filters like category, size, and colour.

5. Order Placement: The system should allow users to add products to their cart, choose a payment method, and place an order successfully.

6. Data Backup: The system should automatically back up critical user and transaction data at regular intervals.

Functional requirements are usually represented by user stories, use cases, or technical specifications. They must be clear, testable, and achievable.

4.1.2 NON-FUNCTIONAL REQUIREMENTS:

Non-functional requirements define the quality attributes, system constraints, and overall performance characteristics that the system must meet. While functional requirements describe what the system should do, non-functional requirements focus on how the system should perform in various scenarios.

Non-functional requirements ensure that the system is built to be robust, scalable, secure, and responsive to the needs of its users. These requirements typically focus on aspects like performance, reliability, usability, and security, which are essential to ensure a smooth user experience and operational efficiency.

Non-functional requirements are often performance-based and may vary in importance depending on the nature of the project. These requirements can sometimes be less tangible than functional requirements but are equally critical for system success.

They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

1. Emails should be sent with a latency of no greater than 12 hours from such an activity.

2. The processing of each request should be done within 10 seconds

3. The site should load in 3 seconds whenever of simultaneous users are > 10000

4.2 HARDWARE REQUIREMENTS

Processor	- I3/Intel Processor
RAM	- 8GB (min)
Hard Disk	- 128 GB
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse
Monitor	- Any

4.3 SOFTWARE REQUIREMENTS

Operating System	:	Windows 7/8/10
Programming Language	:	Python
Libraries Used	:	CMake, Dlib, OpenCV, SciPy, Tkinter, NumPy, Flask
IDE	:	PyCharm
Technology	:	AI, Augmented Reality (AR)

4.4 ARCHITECTURE

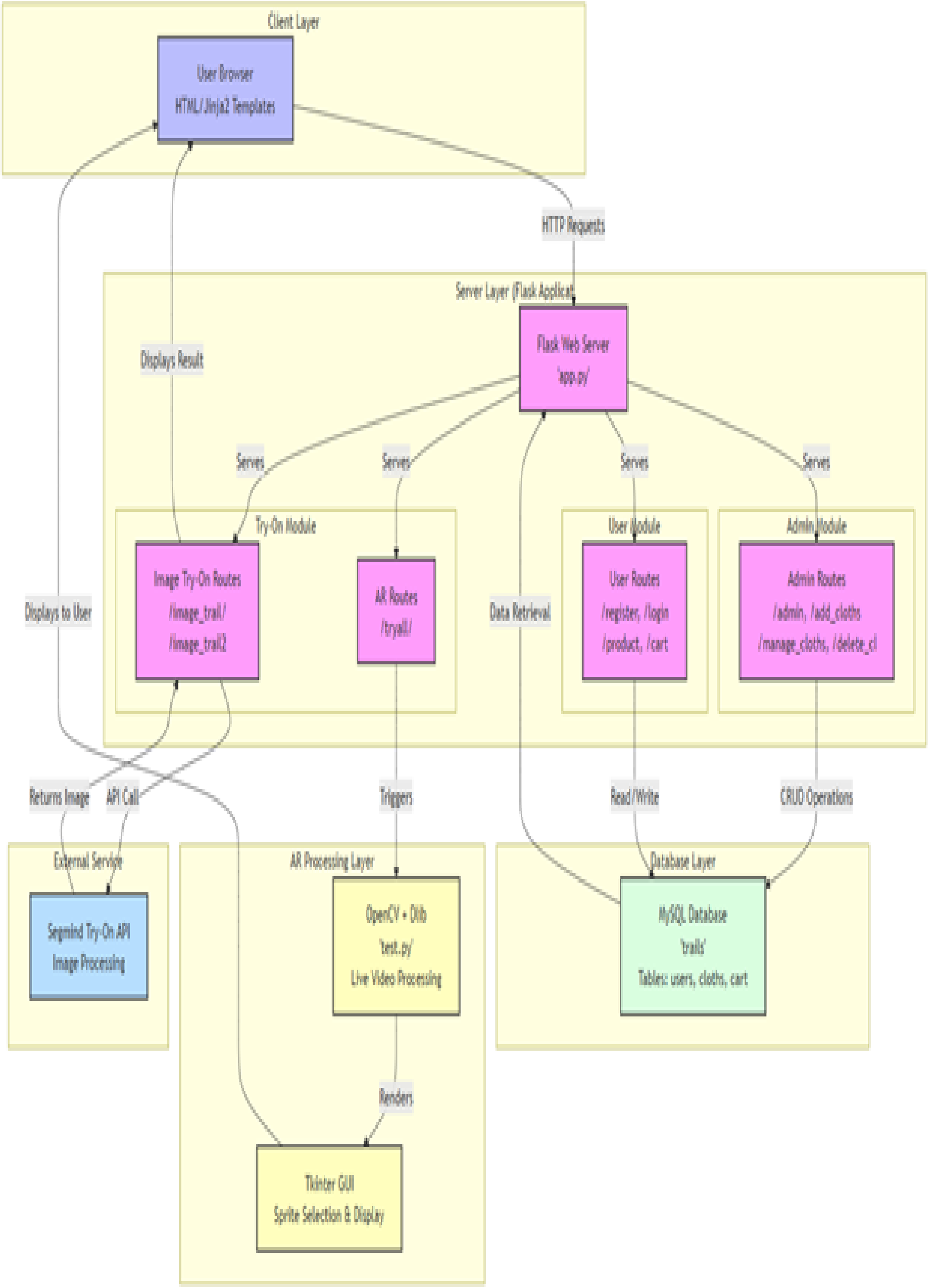


Figure 4.1 - Architecture of project

5. METHODOLOGY

5.1 Haar Cascade Algorithm (Live Video Try-On)

The Haar Cascade algorithm is used to detect key upper-body regions (such as the face, shoulders, and torso) in real-time video frames. This detection is crucial for positioning and overlaying clothing items onto the user's image during the AR-based virtual try-on process, ensuring accurate garment placement.

Steps:

1. Feature Extraction:

The algorithm slides a 24x24 pixel window across each video frame, extracting Haar-like features. These features are based on differences in intensity (light and dark regions) and are designed to capture edges or specific textures. The use of integral images helps speed up the process by allowing the fast calculation of these features, making it efficient for real-time applications.

2. AdaBoost Classifier:

AdaBoost is employed to improve the classifier's accuracy. It combines multiple weak classifiers (each focusing on different features) into a single strong classifier. Over several iterations, the algorithm focuses on harder-to-classify areas, selectively refining the model to better distinguish between body and non-body regions. This ensures the detection process remains reliable and efficient.

3. Cascade Structure:

The Haar Cascade classifier operates in a multi-stage cascade structure. In the first stage, simple classifiers quickly eliminate regions of the frame that do not contain a body. More complex classifiers are then applied to the remaining regions in subsequent stages. This early rejection of non-relevant regions reduces computational load and speeds up processing, making the system suitable for real-time performance.

4. Output:

Once the body regions are detected, bounding boxes are drawn around the face, shoulders, and torso. These bounding boxes help the system position the virtual clothing in the correct areas, ensuring a realistic representation of how the garment fits the user. The process is completed in real-time, allowing users to see the virtual try-on experience instantly.

5.2 Dlib Facial Landmark Detection (Live Video Try-On)

Dlib Facial Landmark Detection identifies specific facial landmarks in real-time to help accurately position accessories or clothing (e.g., tops) during AR-based virtual try-ons. The precise detection of facial features ensures that items like sunglasses, hats, and garments align perfectly with the user's facial structure and body.

Steps:

1. Face Detection:

The process begins by detecting the user's face in the video frame using Dlib’s pre-trained frontal face detector. This step identifies the location of the face in the image, setting the foundation for landmark extraction.

2. Landmark Localization:

Once the face is detected, the shape_predictor_68_face_landmarks.dat model is used to extract 68 specific facial landmarks. These landmarks correspond to key facial features such as the eyes, eyebrows, nose, and jawline. For example, landmarks 36-47 correspond to the eyes, landmarks 1-17 mark the jawline, and landmarks 6-8 represent the neck base.

3. Mapping Features:

Next, the system maps these landmarks to corresponding positions on the clothing or accessories. For tops, the jawline and neck (landmarks 1-17, 6-8) are used to align the garment. For accessories like glasses or hats, features like the eyes or ears (landmarks 36-47, 1-5) are mapped to ensure correct placement.

4. Output:

The algorithm outputs the coordinates of these landmarks, which are then used for accurate sprite placement on the user’s face or upper body. The precision of this mapping guarantees that the virtual try-on experience looks realistic, with items aligning perfectly with the user’s features.

5.3 OpenCV Sprite Overlay (Live Video Try-On)

OpenCV Sprite Overlay is responsible for placing clothing items (such as tops, dresses, or accessories) onto the user's body regions in video frames. The overlay occurs seamlessly with transparency effects, ensuring that the virtual clothing fits naturally on the user in real-time.

Steps:

1. Read Sprite:

The clothing sprite, typically an image file (e.g., PNG), is loaded into the system using OpenCV’s cv2.imread() function. The -1 argument ensures that the image is loaded with an alpha channel, which is essential for transparency. This allows the sprite to have transparent areas, such as around the edges of a garment.

2. Resize Sprite:

Next, the sprite is resized to fit the detected body region. For instance, the width of the clothing sprite is scaled based on the detected body part’s width (e.g., face or upper body). OpenCV’s cv2.resize() function is used to resize the sprite while maintaining its aspect ratio, ensuring the clothing doesn’t stretch or distort when placed on the user.

3. Overlay Logic:

The sprite is then blended with the video frame using alpha blending. The formula used for blending is:

$$\text{Output}=(\alpha\times\text{Sprite})+(1-\alpha)\times\text{Frame}$$

Where α\alphaα is the alpha channel of the sprite, determining its transparency. The system also adjusts the sprite’s position based on the landmarks or bounding boxes, ensuring that it overlays correctly on the user's body, such as aligning it with the shoulders or torso.

4. Output:

The final result is a video frame with the clothing seamlessly overlaid onto the user’s body in real-time. The transparency of the alpha channel ensures the clothing blends naturally with the video background, creating a realistic virtual try-on experience.

5.4 Segmind Try-On Diffusion API (Image-Based Try-On)

The Segmind Try-On Diffusion API allows for the generation of static images where clothing items are superimposed onto a user-uploaded photo. This enables a virtual try-on experience for users who prefer not to use real-time video features but still want to see how garments will look on them.

Steps:

1. Input Preparation:

The process begins by gathering the user-uploaded photo and the product image from the database. These two images are then prepared for the API request. To ensure compatibility, both images are converted to base64 format using a function like image_file_to_base64(). This step ensures that the images can be transmitted easily over the web as part of the API request.

2. API Request:

A POST request is sent to the Segmind Try-On Diffusion API. This request includes the following parameters:

- **model_image:** The user’s uploaded photo in base64 format.
- **cloth_image:** The clothing item’s image, also in base64.
- **category:** The type of clothing being tried on (e.g., "Upper body").

- **num_inference_steps:** The number of steps for the inference process (typically set to 35 for a good balance between quality and processing time).
- **guidance_scale:** A parameter to influence how strongly the model should adhere to the original input image (usually set to 2).

This step triggers the virtual try-on process, where the API processes the input and begins generating the composite image.

3. Image Processing:

Once the API receives the request, it performs the necessary image processing and returns a binary image response. This response is then saved as a new file (e.g., generated_image.jpg) in the static/saved_images/ directory. This processed image contains the user’s uploaded photo with the selected clothing item integrated onto the user’s body, aligned according to the category and features of the clothing.

4. Output:

Finally, the generated image is displayed on the front-end. A rendered HTML page is created to show the composite image of the user and the clothing. This allows the user to view how the clothing would appear on them, completing the image-based virtual try-on experience.

5.5 Pose Alignment Network (Future Enhancement for Live Video Try-On)

The Pose Alignment Network is a proposed future enhancement for the live video try-on functionality. By analyzing the user's body posture in real-time, this network will dynamically adjust the alignment of the clothing overlay to match the user's pose, improving the accuracy and realism of the virtual trial experience. This enhancement will allow the clothing to adapt based on the user's movements, ensuring that the garment fits naturally with their body in various positions.

Steps:

1. Pose Estimation:

To begin, a neural network model, such as OpenPose, will be used for pose estimation. This model detects key body landmarks, including the shoulders, elbows, knees, and wrists, in the user's live video feed. The keypoints captured by OpenPose provide detailed information about the user's body posture. This data forms the foundation for dynamic clothing adjustment, allowing the system to accurately understand the user's body position at any given moment.

2. Alignment:

Once the pose data is captured, the clothing sprite will undergo alignment adjustments based on the detected body keypoints. The system will calculate how the garment should be adjusted by considering factors such as:

- **Angle Adjustments:** The clothing will be rotated according to the angle of the user's body (e.g., shoulders, torso orientation).
- **Scaling:** The size of the clothing sprite will be modified based on the detected body parts' scale, ensuring that the clothing fits correctly to the user’s body dimensions in real-time.
- **Geometric Transformations:** Advanced geometric transformations will be applied, ensuring that the clothing adapts to posture-specific fitting. For example, if the user raises their arms or bends forward, the clothing overlay will adjust its position and size accordingly to avoid distortion.

3. Output:

The final output will be a dynamically aligned clothing overlay that moves and adjusts in real-time based on the user’s posture. As the user changes their position, the clothing will automatically align with the new pose, offering a more accurate and natural fitting experience. This enhancement aims to improve the realism of the virtual try-on process by making the clothing appear to interact with the user’s body in a more lifelike manner.

5.6 Texture Refinement Network (Future Enhancement for Image-Based Try-On)

The Texture Refinement Network is a proposed future enhancement for the image-based try-on functionality. Its goal is to improve the visual realism of the garment by refining the texture, edges, and overall fit of the clothing on the user’s uploaded image. This enhancement focuses on making the garment look more natural by enhancing fabric details, improving the blending of the garment with the user’s photo, and achieving a more seamless appearance.

Steps:

1. Garment Segmentation:

The first step in the Texture Refinement Network is to isolate the clothing region within the user-uploaded photo. This will be done using an image segmentation model, which will effectively detect and separate the garment from the rest of the image (i.e., the user’s body, background, etc.). This allows the system to focus specifically on the garment area and apply refinement techniques without affecting other parts of the image. Advanced segmentation models like U-Net or Deep Lab can be used for this purpose to accurately detect and delineate clothing boundaries.

2. Texture Transfer:

Once the garment is segmented, the next step is to enhance the fabric's texture mapping. The system will transfer the garment's texture details from the clothing image onto the user’s photo. This involves:

- **Texture Mapping:** Applying the fabric’s textures, such as patterns, wrinkles, or fabric weave, to the isolated clothing region, making sure these textures match the appearance of the user’s photo.
- **Edge Smoothing:** The edges of the clothing will be refined to ensure there is a natural blend between the garment and the user’s body. This process involves smoothing out harsh lines or mismatched boundaries, ensuring the garment doesn’t

look artificially inserted into the image. The goal is to create a seamless integration of the clothing with the user’s skin, body contours, and the overall photo.

3. Output:

The final output of the Texture Refinement Network will be a high-quality, realistic image that shows the user wearing the garment with improved texture details. The refined garment will have smoother edges, more accurate fabric representation, and a better fit, making the virtual try-on experience more lifelike. The system will output the final image, which will be displayed to the user on the platform, offering a more natural and appealing visual representation of how the clothing would look on them.

Each of these algorithms plays a crucial role in the successful implementation of the AI-powered virtual garment trial room. By combining real-time processing techniques, machine learning models, and advanced image processing, the system delivers an interactive and highly realistic virtual shopping experience. The algorithms used—ranging from Haar Cascade for body detection to Pose Alignment for dynamic clothing fitting—are carefully chosen to enhance both user interaction and the overall realism of the virtual garment fitting process. Together, these methodologies ensure that the project not only meets functional requirements but also provides a smooth, user-friendly experience.

6. SYSTEM DESIGN

6.1 INTRODUCTION TO INPUT DESIGN

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

- Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −
- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user’s attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens.

OBJECTIVES FOR INPUT DESIGN:

The objectives of input design are −

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

OUTPUT DESIGN:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

OBJECTIVES OF OUTPUT DESIGN:

The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions

6.2 UML DIAGRAMS:

- UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the object management group.
- The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.
- The unified modeling language is a standard language for specifying, visualization, constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex system.

6.2.1 USE CASE DIAGRAM:

A use case diagram in the Unified MODELING Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

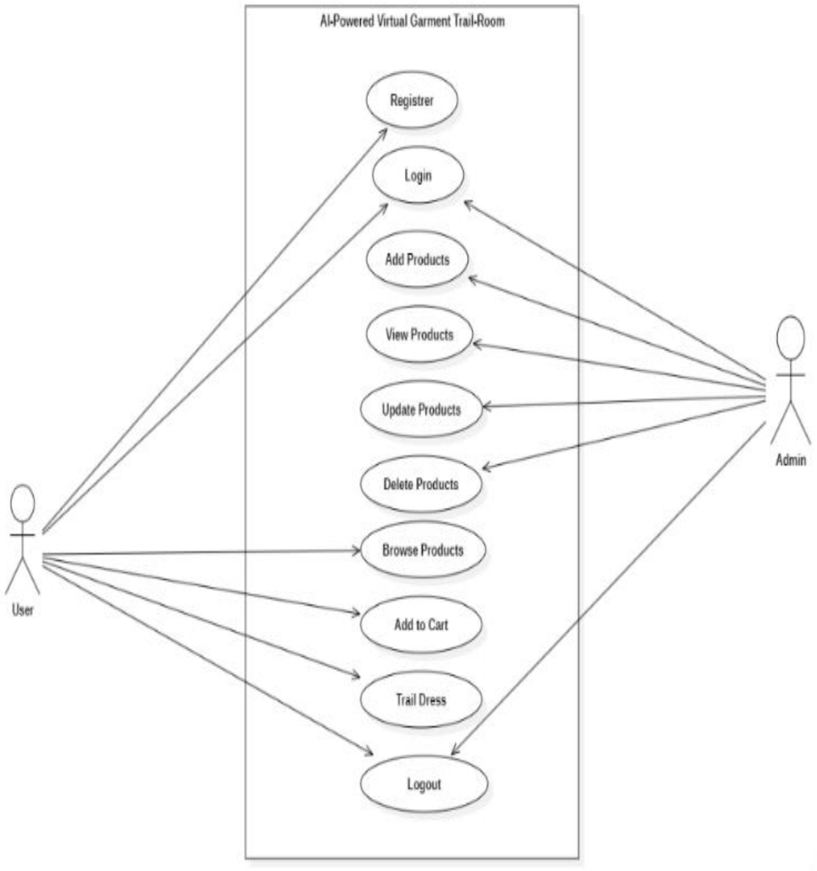


Figure 6.1 - USE CASE DIAGRAM

6.2.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified MODELING Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

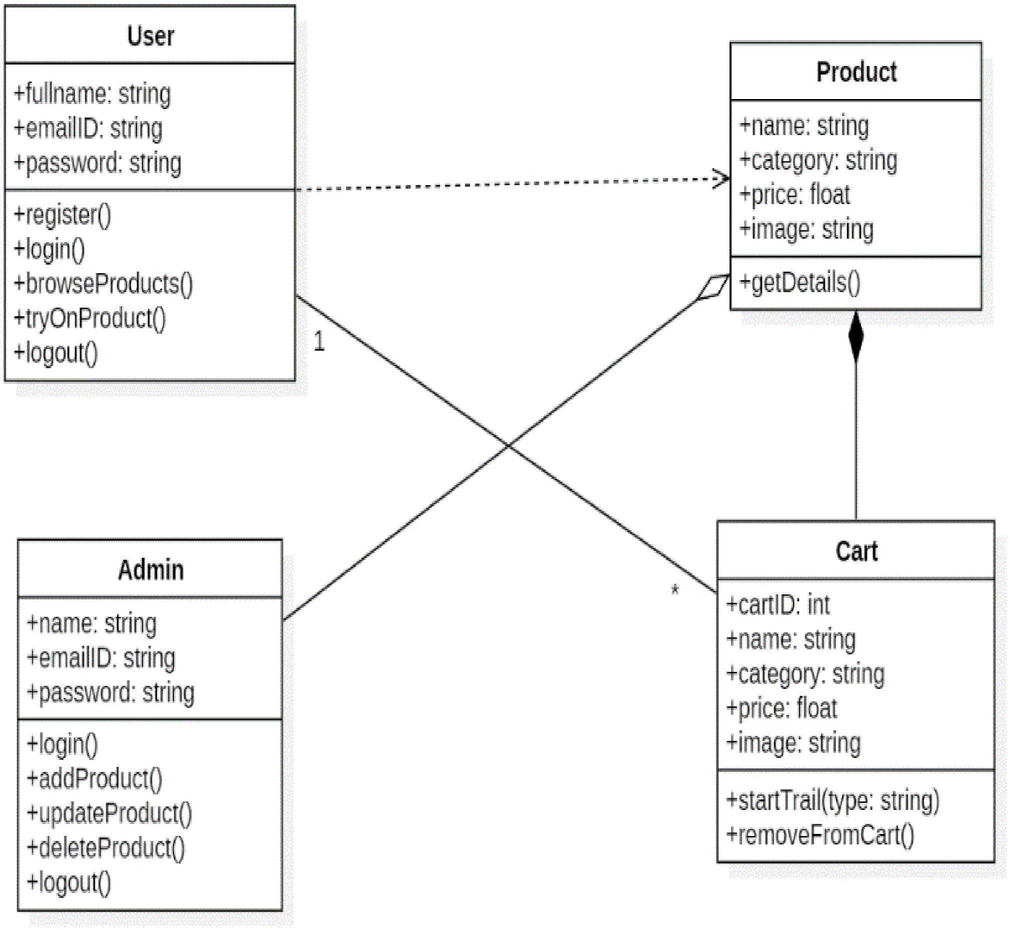


Figure 6.2 - CLASS DIAGRAM

6.2.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified MODELING Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

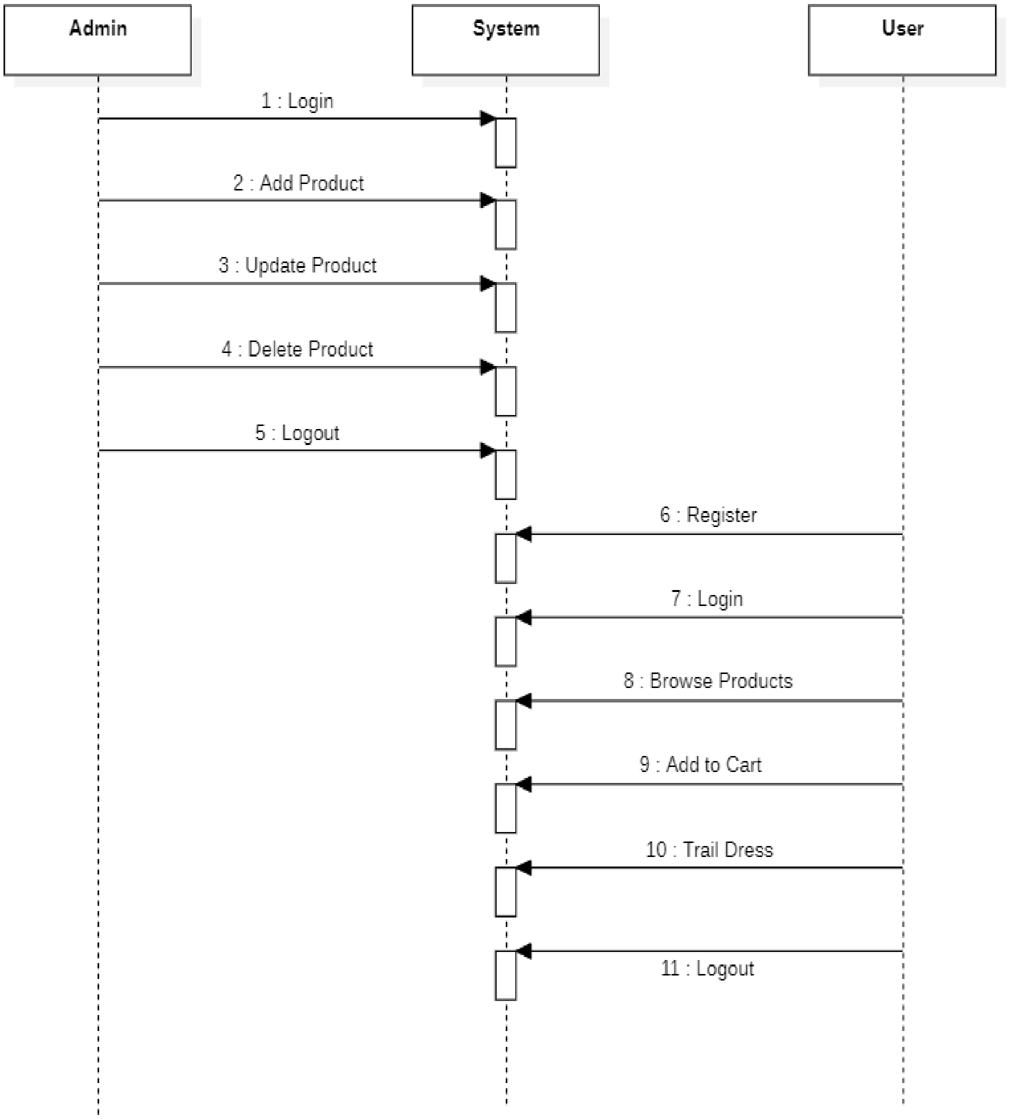


Figure 6.3 - SEQUENCE DIAGRAM

6.2.4 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified MODELING Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

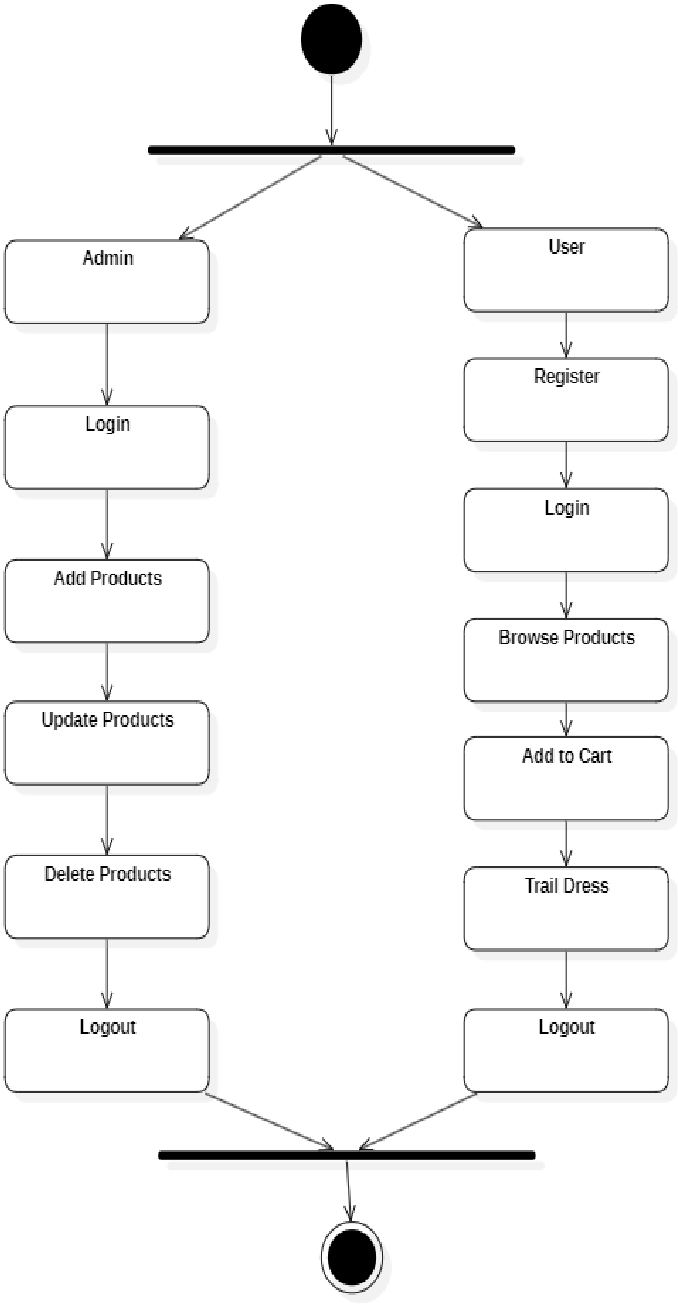


Figure 6.4 - ACTIVITY DIAGRAM

6.2.5 COMPONENT DIAGRAM:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.

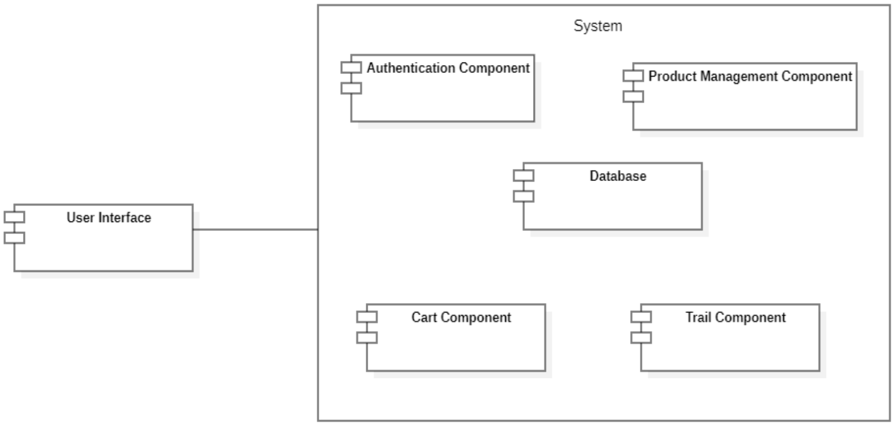


Figure 6.5 - COMPONENT DIAGRAM

6.2.6 DEPLOYMENT DIAGRAM:

A Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

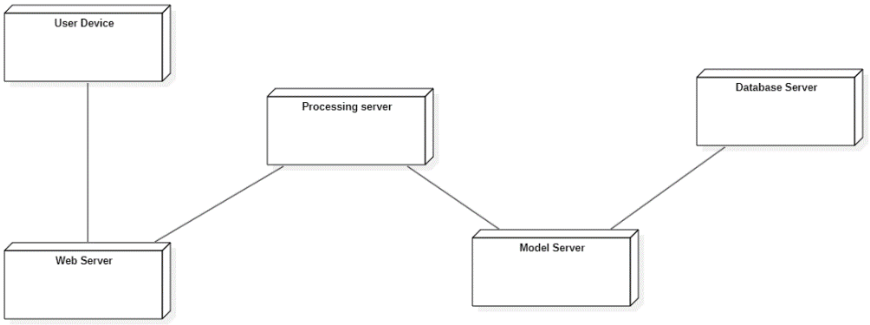


Figure 6.6 - DEPLOYMENT DIAGRAM

6.2.7 ER DIAGRAM:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

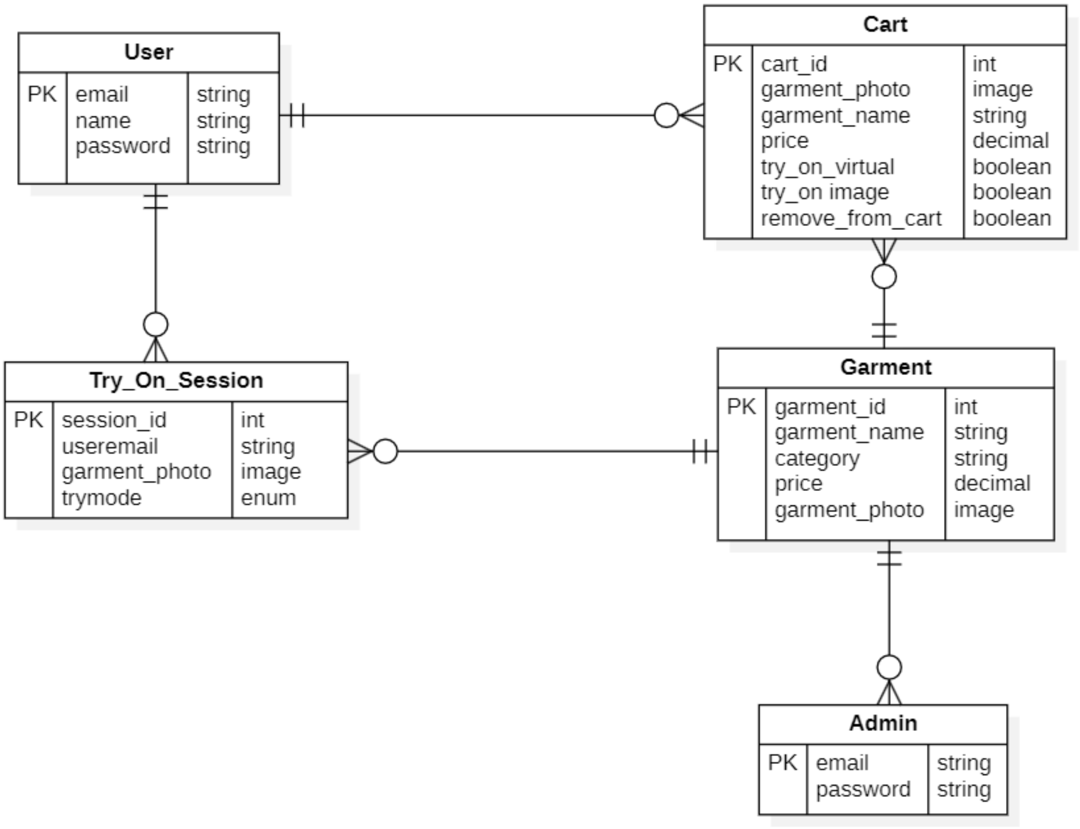


Figure 6.7 - ACTIVITY DIAGRAM

6.2.8 DFD DIAGRAM:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

CONTEXT DIAGRAM:

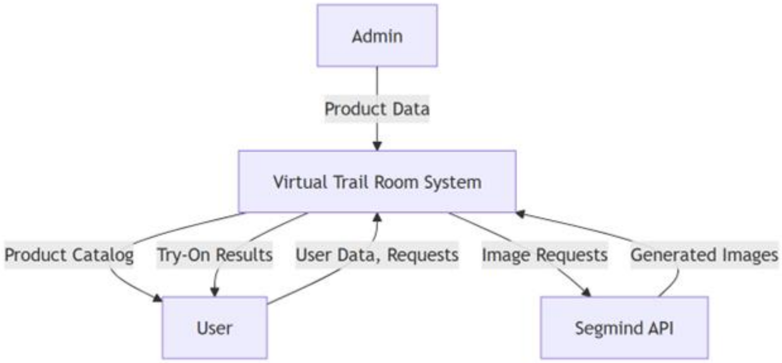


Figure 6.8 - DFD CONTEXT DIAGRAM

DFD LEVEL 1 DIAGRAM:

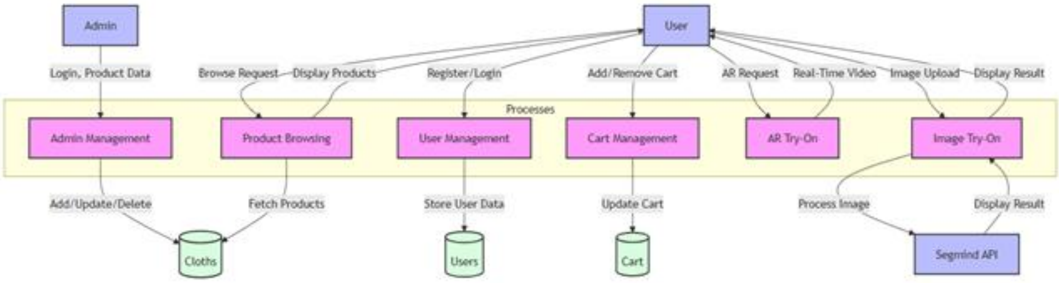


Figure 6.9 - DFD LEVEL 1 DIAGRAM

DFD LEVEL 2 DIAGRAM:

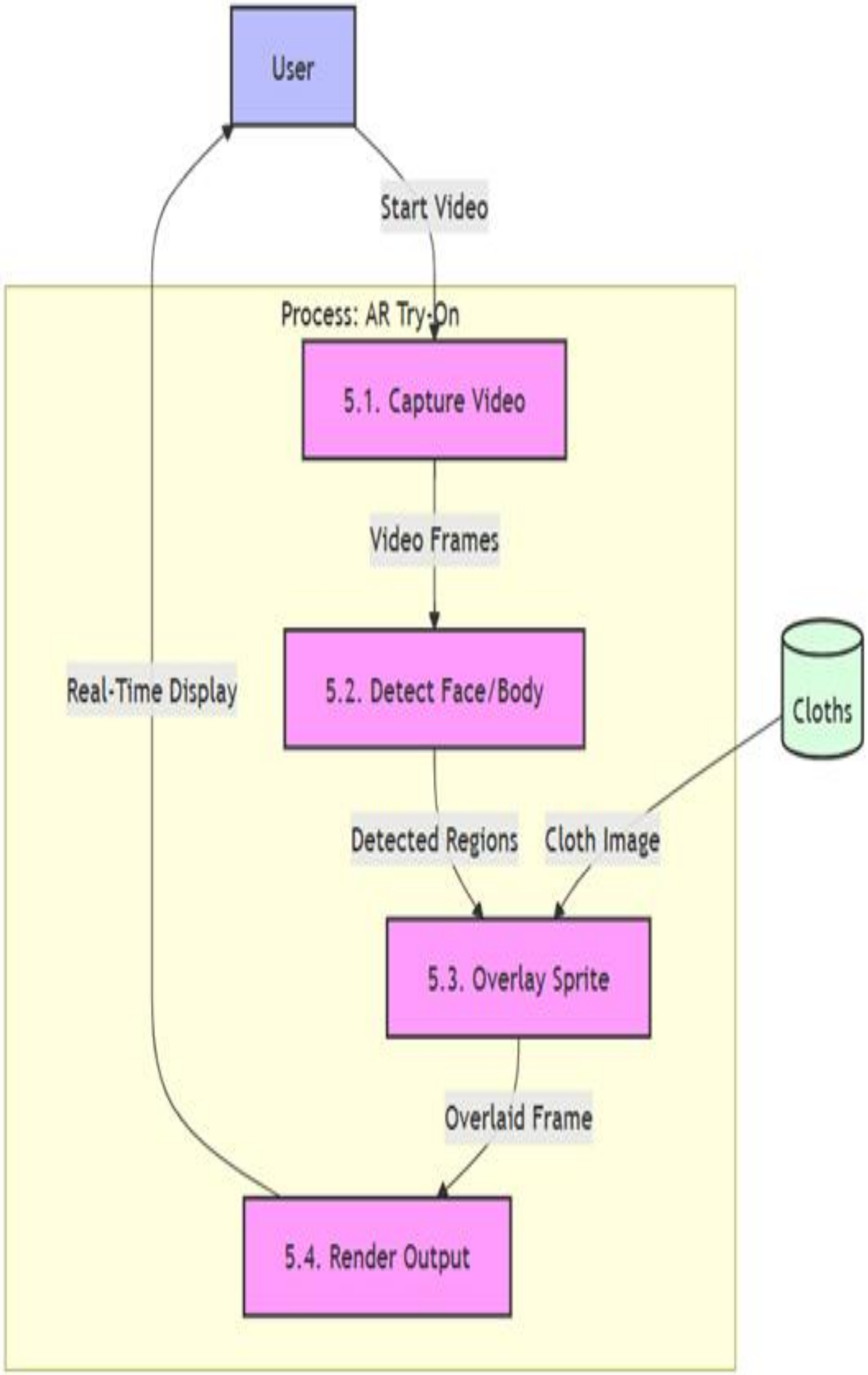


Figure 6.10 - UML DFD LEVEL 2 DIAGRAM

7. IMPLEMENTATION AND RESULTS

7.1 MODULES:

7.1.1 ADMIN MODULE

- ❖ **Purpose:** Manages the product catalogue for the virtual trial room, enabling administrators to maintain an up-to-date inventory.
- ❖ **Functionalities:**
  - **Login:** Secure access using predefined credentials (admin@gmail.com, admin).
  - **Add Product:**
    - o Upload new clothing items with:
      - Category (e.g., "Tops").
      - Name.
      - Cost.
      - Image (stored in static/images/Tops4/).
  - **Update Product:**
    - o Modify existing product details (e.g., name, cost, category, or replace image).
  - **Delete Product:**
    - o Remove outdated or unavailable items from the catalogue.
- ❖ **Flow:**

1.Admin logs in via /admin.

2.Performs CRUD operations through add_cloths, manage_cloths and delete_cloth

3.Changes are saved to the MySQL cloths table and reflected in the user interface instantly.

7.1.2 USER MODULE

- ❖ **Purpose:** Enables users to register, shop, and virtually try on clothes using AR or image-based methods.

❖ **Functionalities:**

- **Register:** Users create an account with name, email, and password (/register).
- **Login:** Secure login to access personalized features (/login).
- **Shop:** Browse products from the cloths table, displayed at /product.
- **Add to Cart:** Add selected items to a cart stored in the cart table (/product POST).
- **Try On:**
    - **AR:** Overlay clothes on live video feed via /tryall/<product_id>.
    - **Image:** Upload a photo for try-on via /image_trail/<product_id> and /image_trail2.

❖ **Flow:**

1.Users register and log in to access /home.

2.Browse products at /product, select items for AR (test.py) or image try-on (Segmind API).

3.Add desired items to the cart at /cart for review or removal

7.1.3 AR MODULE

❖ **Purpose:** Processes live video feeds to detect body parts and overlay clothing in real-time.

❖ **Functionalities:**

- **Live Video Feed:** Captures webcam input using OpenCV (cv2.VideoCapture).
- **Body and Face Detection:**
    - **Haar Cascades:** Detects upper body (via haarcascade_fullbody.xml).
    - **Dlib:** Identifies 68 facial landmarks for precise alignment.
- **Overlay:**
    - Resizes clothing sprites (e.g., from static/images/Tops4/) based on detected region size.
    - Applies alpha blending for transparency and adjusts position using landmark coordinates.

- **Flow:**

1.Webcam captures frames in test.py.

2.Haar cascades and Dlib detect upper body and facial landmarks.

3.Selected clothing is overlaid on the video feed, rendered via Tkinter GUI in real-time

7.1.4. IMAGE TRY-ON MODULE

After processing, users receive and view the classification results, which include the identified bird species and corresponding confidence levels. Results are displayed in an intuitive format, often accompanied by additional information such as species descriptions and images to enhance user understanding and engagement.

- **Purpose:** Processes static user-uploaded images to superimpose clothing using an external API.
- **Functionalities:**
  - **Image Upload:** Accepts user photos via /image_trail/<product_id> (person_img).
  - **API Integration:**
    - Uses Segmind Try-On Diffusion API to combine user photo with product image.
    - Parameters include model_image, cloth_image, and category ("Upper body").
  - **Output Generation:**
  - Saves the resulting composite image to static/saved_images/generated_image.jpg.
- **Flow:**

1. User uploads a photo and selects a product at /image_trail/<product_id>.

2. Data is sent to Segmind API via /image_trail2 for processing.

3. Generated image is displayed in image_trail2.html for user review.

7.2 SOFTWARE INSTALLATION:

To build and deploy the AI-Powered Virtual Garment Trial Room using Augmented Reality (AR), a variety of software tools were utilized to ensure seamless development across the front-end, back-end, and database systems. Key tools included Visual Studio Code (VS Code) for writing and debugging the project's code, Node.js for running JavaScript server-side to facilitate real-time interactions, and XAMPP with MySQL for efficient database management and back-end operations. These platforms were essential for creating the interactive, user-friendly experience needed for garment try-ons. In addition, Anaconda was used to manage the Python environment, particularly for running machine learning models and handling the image processing tasks necessary for the AR-based garment overlay. Each of these tools played a critical role in delivering a robust and dynamic application for users to try on clothes virtually.

1. VS Code:

Visual Studio Code (VS Code) is a free, open-source code editor that is widely used by developers for its flexibility, support for multiple programming languages, and rich extension ecosystem. It's especially useful for projects that involve web development, data science, and machine learning. With features like integrated Git support and debugging, VS Code enhances the overall development experience.

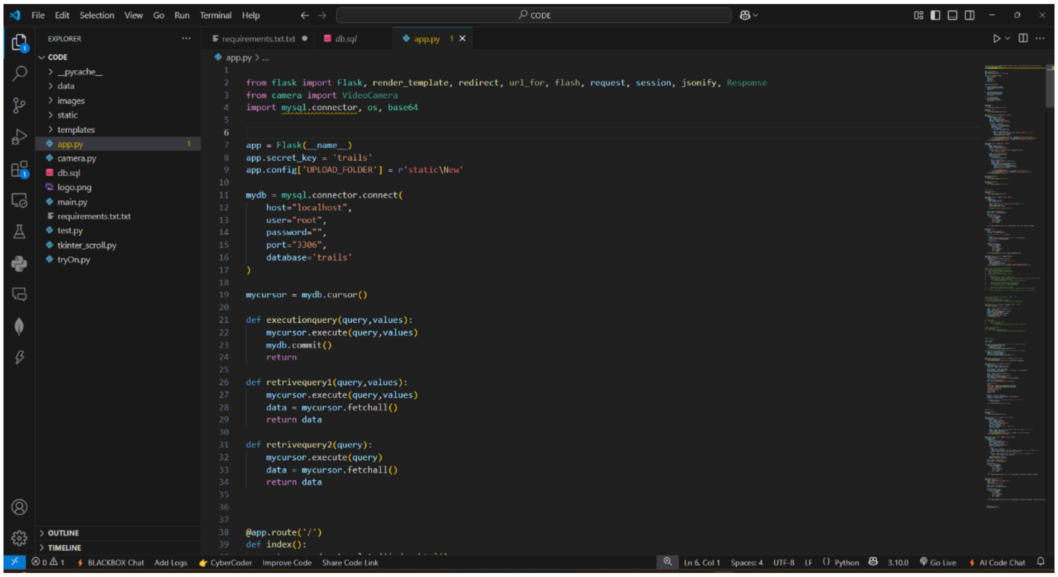
Key Features:

- Intelligent Code Editing: Provides intelligent code suggestions, auto-completion, and syntax highlighting, which speeds up development and reduces the chances of coding errors.
- Debugging Capabilities: Integrated debugging tools allow you to run your code and identify errors within the editor itself, making it easier to fix issues quickly.
- Extensions Marketplace: VS Code supports a wide range of extensions, such as Python, Jupyter Notebooks, GitHub integration, and many others, to tailor the editor to your development needs.

- Integrated Git: Version control is made easy with built-in Git support, allowing you to manage your code versions and collaborate with team members without leaving the editor.
- Cross-platform: Available for Windows, macOS, and Linux, ensuring compatibility across different operating systems.

Installation Steps:

1. Go to the Visual Studio Code download page.
2. Download the correct version for your operating system (Windows, macOS, or Linux).
3. Follow the installation wizard to complete the setup.
4. Once installed, open the editor and explore the Extensions Marketplace to install necessary extensions for Python, JavaScript, or any other languages you are working with.



```python

from flask import Flask, render_template, redirect, url_for, flash, request, session, jsonify, Response
from camera import VideoCamera
import mysql.connector, os, base64


app = Flask(__name__)
app.secret_key = 'trails'
app.config['UPLOAD_FOLDER'] = r'static\New'

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    port="3306",
    database='trails'
)

mycursor = mydb.cursor()

def executionquery(query,values):
    mycursor.execute(query,values)
    mydb.commit()
    return

def retrivequery1(query,values):
    mycursor.execute(query,values)
    data = mycursor.fetchall()
    return data

def retrivequery2(query):
    mycursor.execute(query)
    data = mycursor.fetchall()
    return data



@app.route('/')
def index():
```

Screen 7.1 - VS Code

Usage in the Project:

VS Code will serve as your primary development environment for coding. You will write Python scripts, implement machine learning models, work with web technologies like

HTML, CSS, and JavaScript, and integrate with databases using this editor. It simplifies the development workflow through real-time linting and error-checking.

2. Anaconda:

Anaconda is a Python distribution tailored for scientific computing and data science. It comes with pre-installed libraries for data analysis, machine learning, and deep learning, which is essential for building models and performing data manipulation tasks in the AI-Powered Virtual Garment Trial Room project. Anaconda makes it easy to manage environments, install dependencies, and deploy applications.

Key Features:

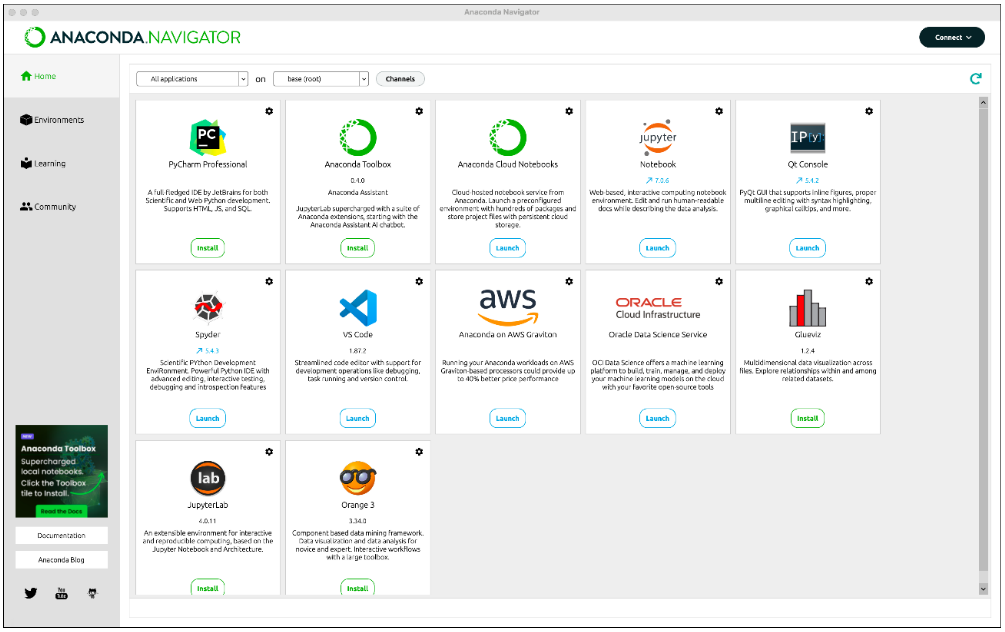
- **Package Management:** Anaconda uses conda as its package manager, making it easier to install and manage libraries such as NumPy, pandas, Matplotlib, TensorFlow, and PyTorch, which are crucial for data analysis and machine learning tasks.
- **Environment Management:** With Anaconda, you can create separate virtual environments for different projects, preventing dependency conflicts and ensuring a clean setup for each project.
- **Pre-installed Libraries:** Anaconda comes with over 1,500 packages pre-installed, including popular libraries like Pandas, NumPy, and Scikit-learn, which help with data manipulation, visualization, and machine learning.
- **Data Science Tools:** Anaconda also includes Jupyter Notebooks, which provide an interactive development environment for working with data, testing code, and visualizing results.

Installation Steps:

1. Visit the Anaconda website and download the latest version of Anaconda for your operating system.
2. Run the installer and follow the prompts to install the software.
3. After installation, open Anaconda Navigator or use the command line to create virtual environments and manage packages.

Usage in the Project:

Anaconda will be used to manage your Python environment, ensuring that all required libraries and dependencies are properly installed and isolated for your project. This is crucial when working on complex data science tasks like training machine learning models, processing images, and implementing computer vision techniques for the garment trial room.



PyCharm Professional

A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

Anaconda Toolbox 0.4.0 Anaconda Assistant

JupyterLab supercharged with a suite of Anaconda extensions, starting with the Anaconda Assistant AI chatbot.

Anaconda Cloud Notebooks

Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.

Notebook ↗ 7.0.6

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console ↗ 5.4.2

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder ↗ 5.4.3

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

VS Code 1.87.2

Streamlined code editor with support for development operations like debugging, task running and version control.

Anaconda on AWS Graviton

Running your Anaconda workloads on AWS Graviton-based processors could provide up to 40% better price performance

Oracle Data Science Service

OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools

Glueviz 1.2.4

Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab 4.0.11

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Orange 3 3.34.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Screen 7.2 - Anaconda Navigator Interface

3. XAMPP:

XAMPP is a free, open-source cross-platform web server solution package. It includes Apache HTTP Server, MySQL database, and PHP, among other components. XAMPP is particularly useful for local web development, allowing developers to test websites and web applications on their local machine before deploying them to a live server.

Key Features:

- **Web Server (Apache):** Apache serves as the backbone for serving web pages and hosting web applications locally, enabling developers to test applications before live deployment.

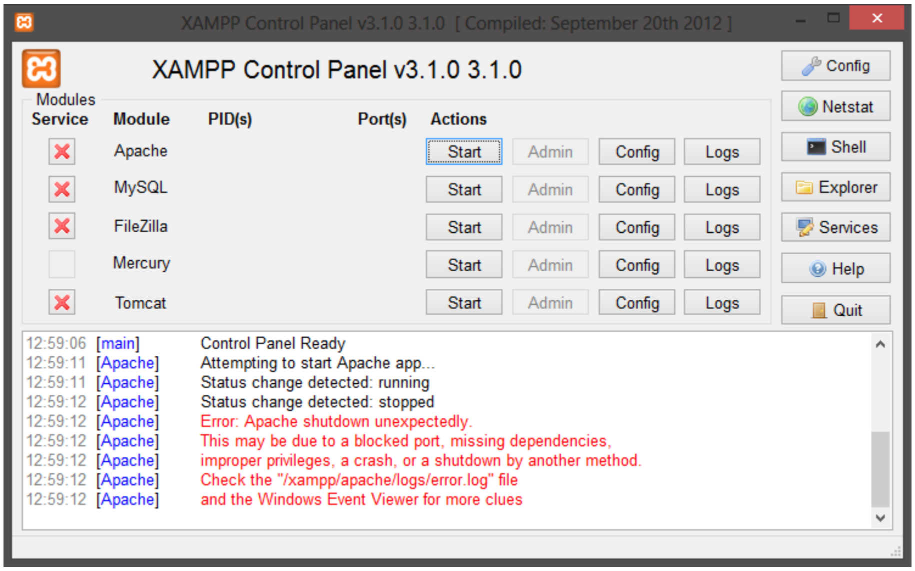
- **Database Management (MySQL):** XAMPP includes MySQL, an open-source relational database management system, which is widely used for storing and managing data in web applications.
- **PHP Support:** XAMPP supports PHP, a server-side scripting language essential for building dynamic websites and backend logic.
- **Cross-Platform:** XAMPP works seamlessly on all major operating systems (Windows, macOS, Linux), making it versatile for local development.

Installation Steps:

1. Download XAMPP from the Apache Friends website.
2. Run the installer and follow the instructions to complete the setup.
3. Once installed, open the XAMPP Control Panel to start services like Apache and MySQL, and test that everything is working.

Usage in the Project:

XAMPP will be used to set up a local web server environment for testing and running the web-based components of your project. For example, the user-facing interface, where they can upload their photos for the garment try-on, will be hosted locally during development and testing using XAMPP.



XAMPP Control Panel v3.1.0 3.1.0 [Compiled: September 20th 2012]

XAMPP Control Panel v3.1.0 3.1.0

Modules Service	Module	PID(s)	Port(s)	Actions			
☒	Apache			Start	Admin	Config	Logs
☒	MySQL			Start	Admin	Config	Logs
☒	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
☒	Tomcat			Start	Admin	Config	Logs

```
12:59:06  [main]     Control Panel Ready
12:59:11  [Apache]   Attempting to start Apache app...
12:59:11  [Apache]   Status change detected: running
12:59:12  [Apache]   Status change detected: stopped
12:59:12  [Apache]   Error: Apache shutdown unexpectedly.
12:59:12  [Apache]   This may be due to a blocked port, missing dependencies,
12:59:12  [Apache]   improper privileges, a crash, or a shutdown by another method.
12:59:12  [Apache]   Check the "/xampp/apache/logs/error.log" file
12:59:12  [Apache]   and the Windows Event Viewer for more clues
```

Screen 7.3 - XAMPP Dashboard

4. NODE.JS:

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It is designed for building scalable network applications, and is especially useful for backend development, handling asynchronous tasks, and building real-time applications. In this project, Node.js will be used to manage server-side tasks and handle real-time communication for live interactions.

Key Features:

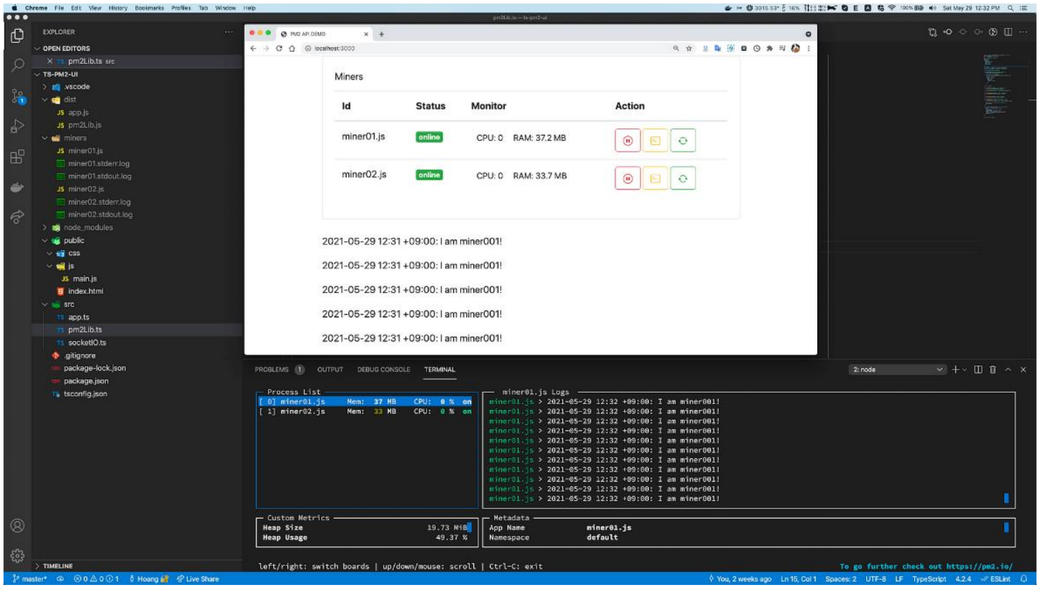
- **Non-blocking I/O:** Node.js is designed for building fast, scalable applications due to its non-blocking, event-driven architecture.
- **npm (Node Package Manager):** npm is included with Node.js and allows you to install and manage a wide range of libraries and tools for backend development.
- **Cross-Platform:** Node.js runs seamlessly on various operating systems (Windows, macOS, Linux).
- **Real-time Application Support:** Ideal for applications like live video processing or chat systems, where real-time communication is essential.

Installation Steps:

1. Go to the Node.js download page.
2. Download the LTS (Long-Term Support) version for your operating system.
3. Run the installer and follow the setup instructions.
4. After installation, verify by opening the terminal and typing node -v to check the installed version.

Usage in the Project:

Node.js will be utilized for building the backend of the live try-on feature, where real-time interactions and video processing need to be handled efficiently. It will also be used to manage HTTP requests, facilitate live video processing, and handle interactions between the server and the client application.



Miners

Id	Status	Monitor	Action
miner01.js	online	CPU: 0 RAM: 37.2 MB	
miner02.js	online	CPU: 0 RAM: 33.7 MB	

2021-05-29 12:31 +09:00: I am miner001!

2021-05-29 12:31 +09:00: I am miner001!

2021-05-29 12:31 +09:00: I am miner001!

2021-05-29 12:31 +09:00: I am miner001!

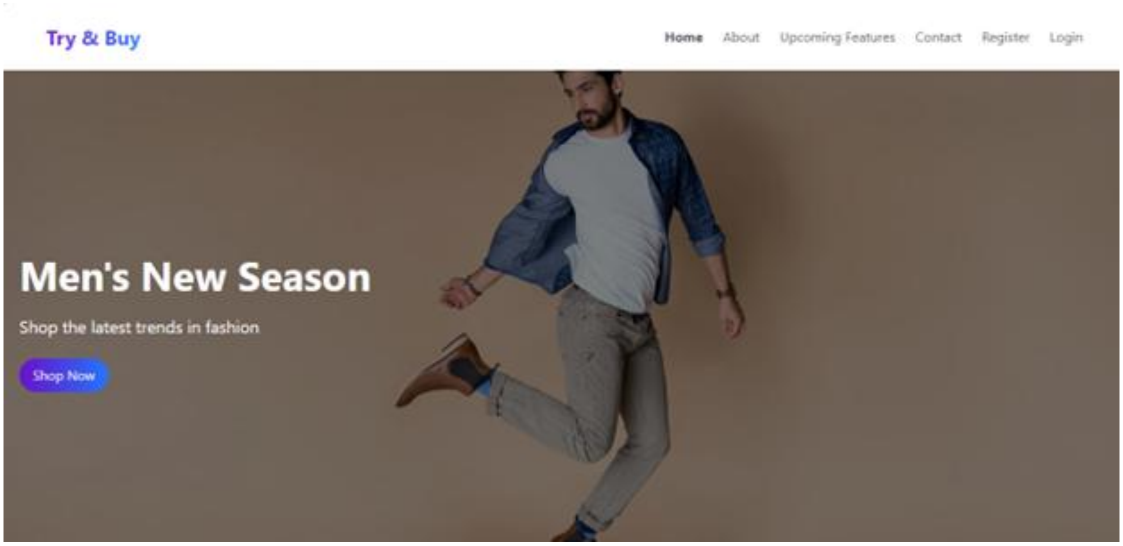
2021-05-29 12:31 +09:00: I am miner001!

```
Process List
[ 0] miner01.js     Mem:  37 MB    CPU:  0 %  on
[ 1] miner02.js     Mem:  33 MB    CPU:  0 %  on

miner01.js Logs
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!
miner01.js > 2021-05-29 12:32 +09:00: I am miner001!

Custom Metrics
Heap Size                      19.73 MiB
Heap Usage                     49.37 %

Metadata
App Name            miner01.js
Namespace           default

left/right: switch boards | up/down/mouse: scroll | Ctrl-C: exit        To go further check out https://pm2.io/
```

Screen 7.4 - Node.js

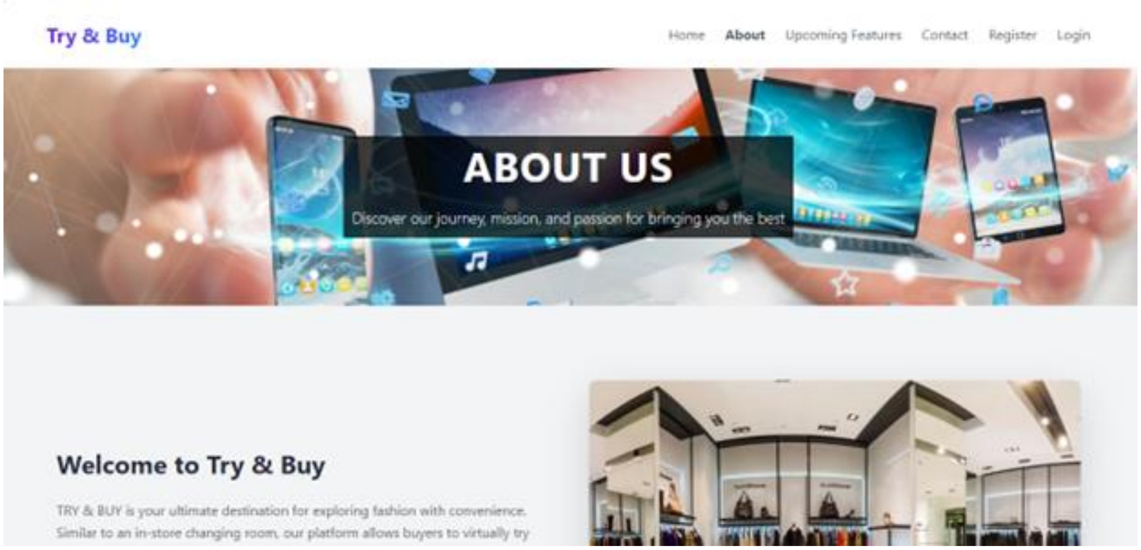
7.2 OUTPUT SCREENS:

HOME PAGE: This is the landing page.



Screen 7.5 - Home Page

ABOUT PAGE: Here information about this project will be there.



ABOUT US

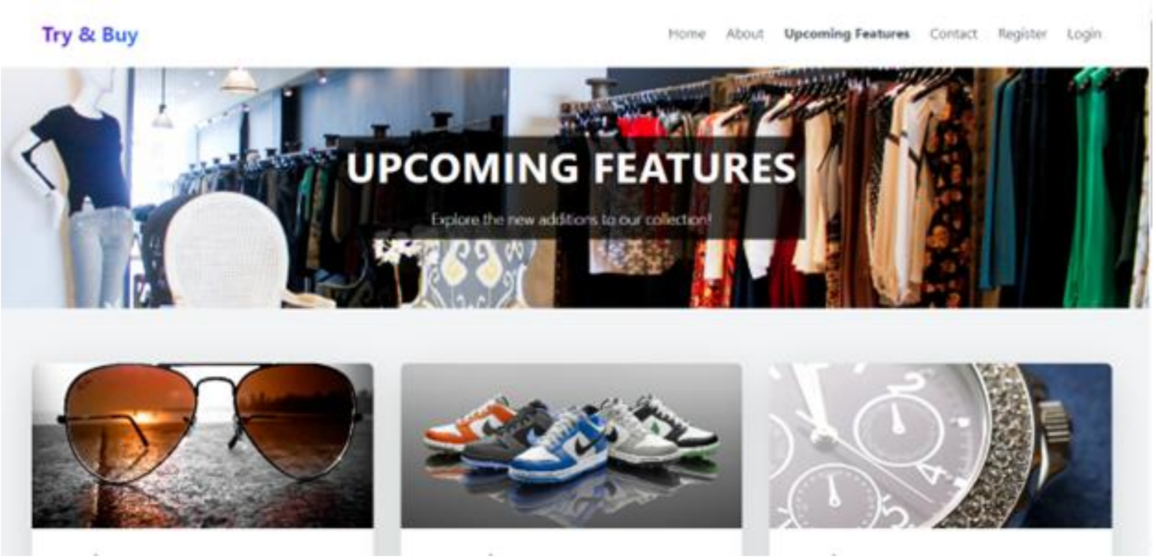
Discover our journey, mission, and passion for bringing you the best

Welcome to Try & Buy

TRY & BUY is your ultimate destination for exploring fashion with convenience. Similar to an in-store changing room, our platform allows buyers to virtually try

Screen 7.6 - About Page

FEATURES PAGE: Here upcoming features will be there.

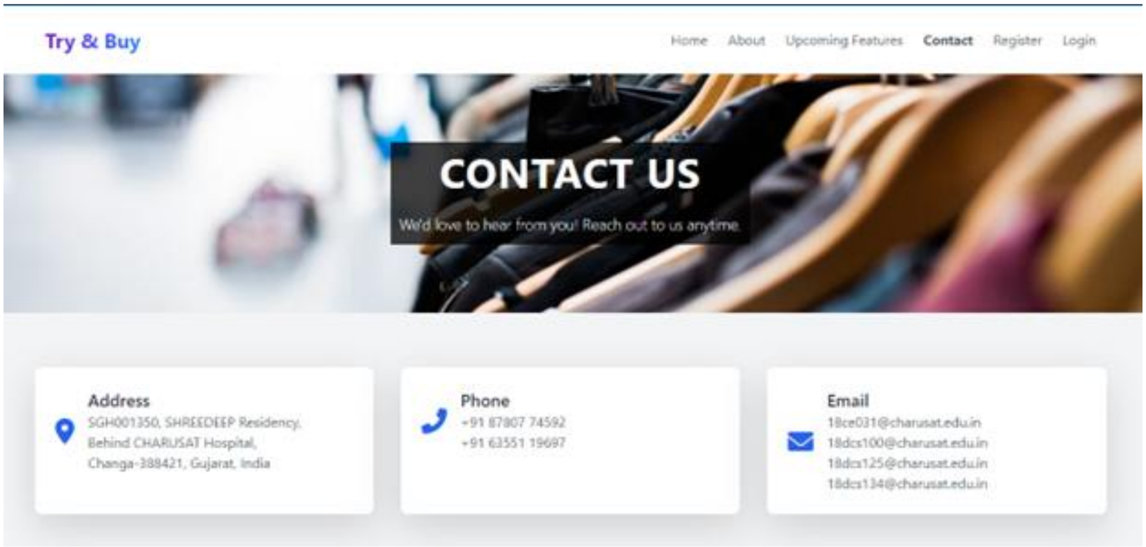


UPCOMING FEATURES

Explore the new additions to our collection!

Screen 7.7 - Registration Page

CONTACT PAGE: Here contact information of the website owners will be there



CONTACT US

We'd love to hear from you! Reach out to us anytime.

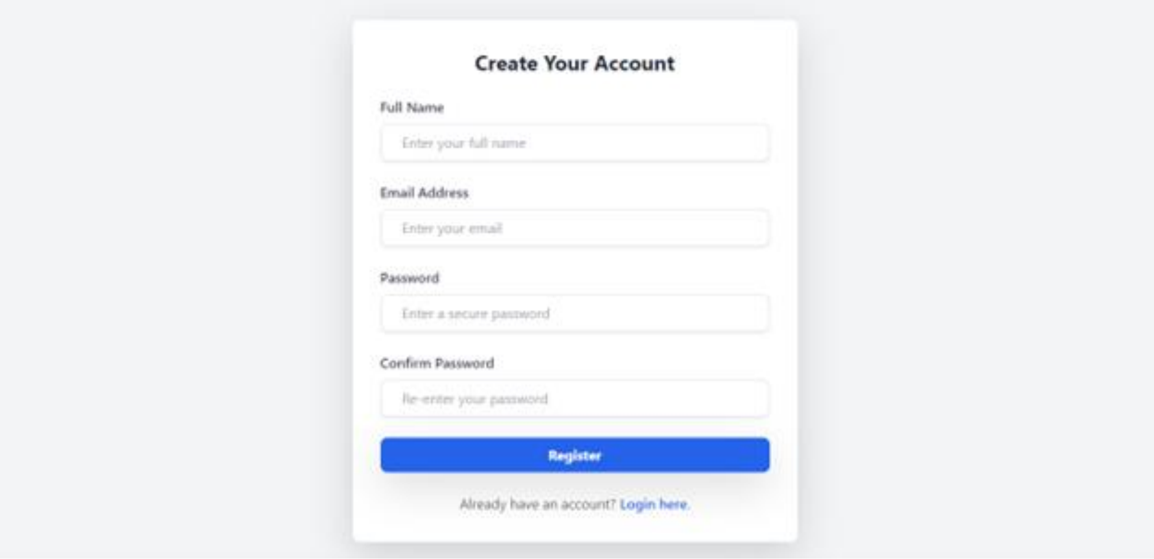
Address
SGH001350, SHREEDEEP Residency,
Behind CHARUSAT Hospital,
Changa-388421, Gujarat, India

Phone
+91 87807 74592
+91 63551 19697

Email
18ce031@charusat.edu.in
18dcs100@charusat.edu.in
18dcs125@charusat.edu.in
18dcs134@charusat.edu.in

Screen 7.8 - Contact Page

REGISTER PAGE: User can register with their credentials



Create Your Account

Full Name
Enter your full name

Email Address
Enter your email

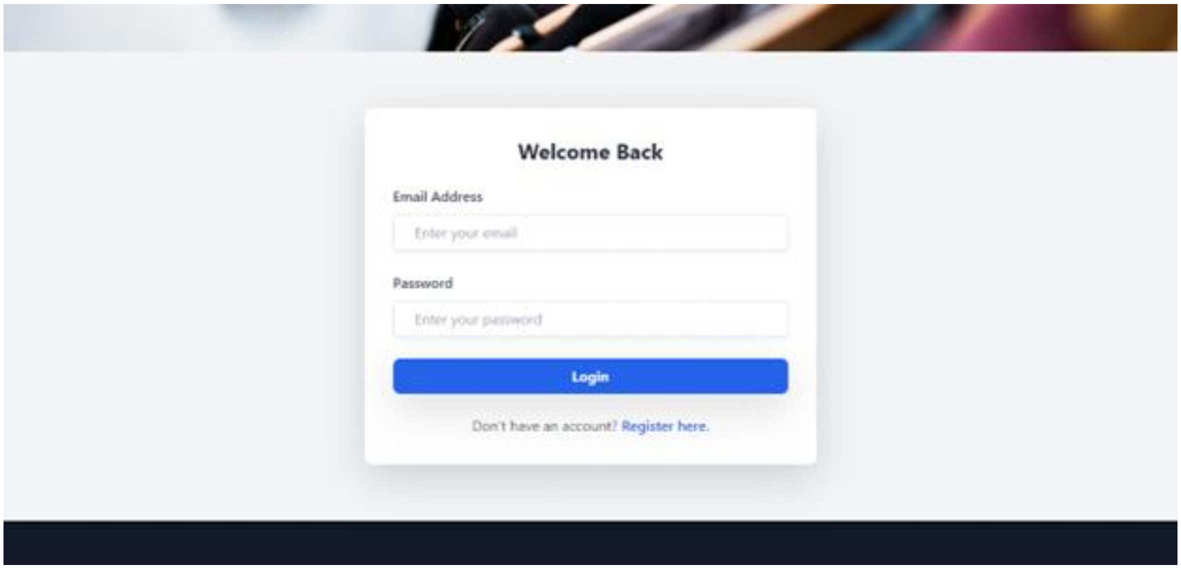
Password
Enter a secure password

Confirm Password
Re-enter your password

Already have an account? Login here.

Screen 7.9 - Register Page

LOGIN PAGE: User can login in here



Welcome Back

Email Address

Enter your email

Password

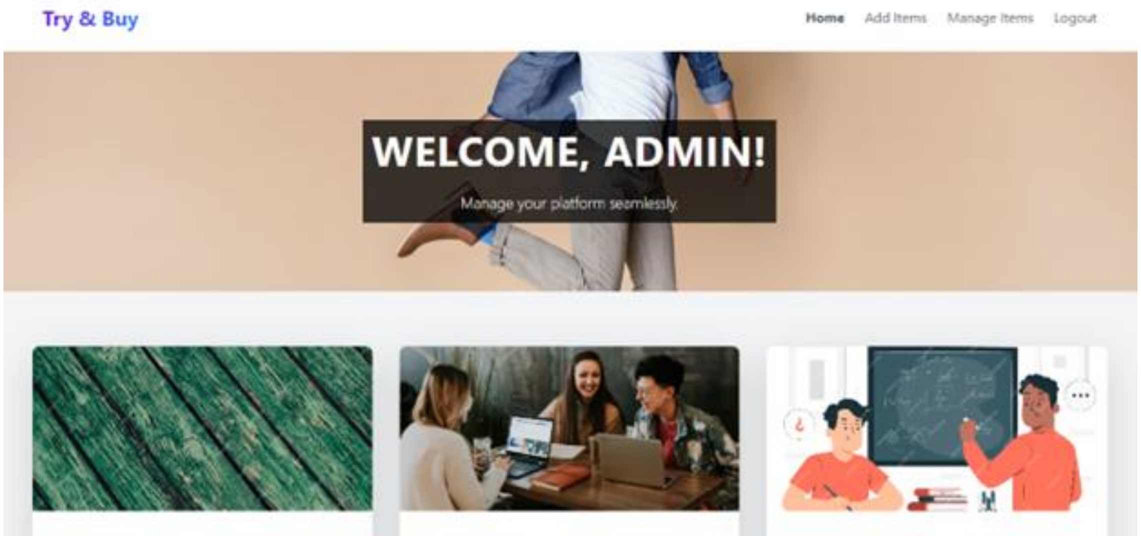
Enter your password

Login

Don't have an account? Register here.

Screen 7.10: - Login Page

ADMIN HOME PAGE: User can register with their credentials

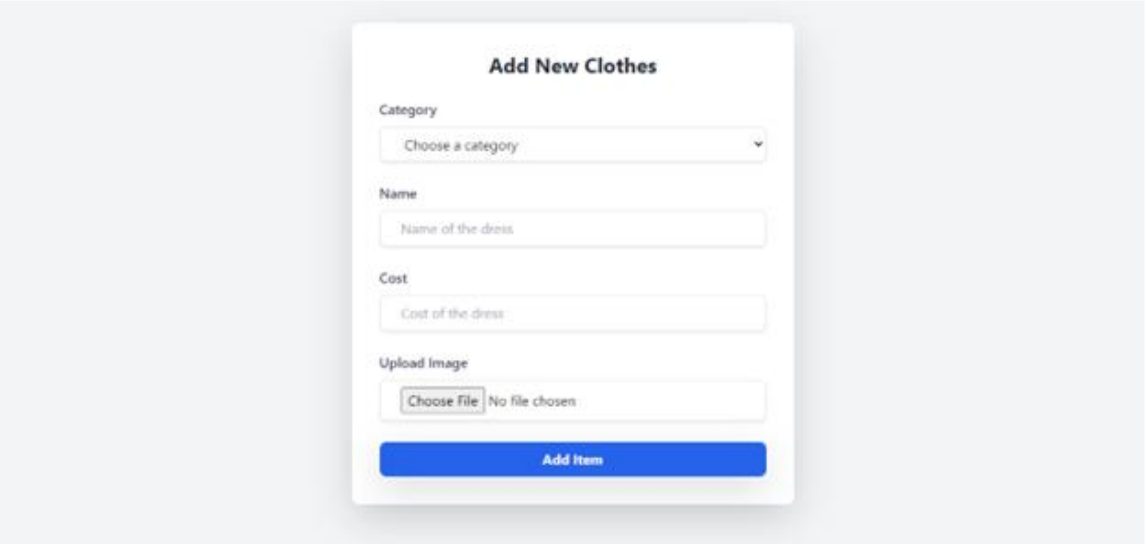


WELCOME, ADMIN!

Manage your platform seamlessly.

Screen 7.11 – Admin Home Page

***ADD CLOTHS PAGE*:** Here admin can add cloths and it's details.



Add New Clothes

Category

Choose a category

Name

Name of the dress

Cost

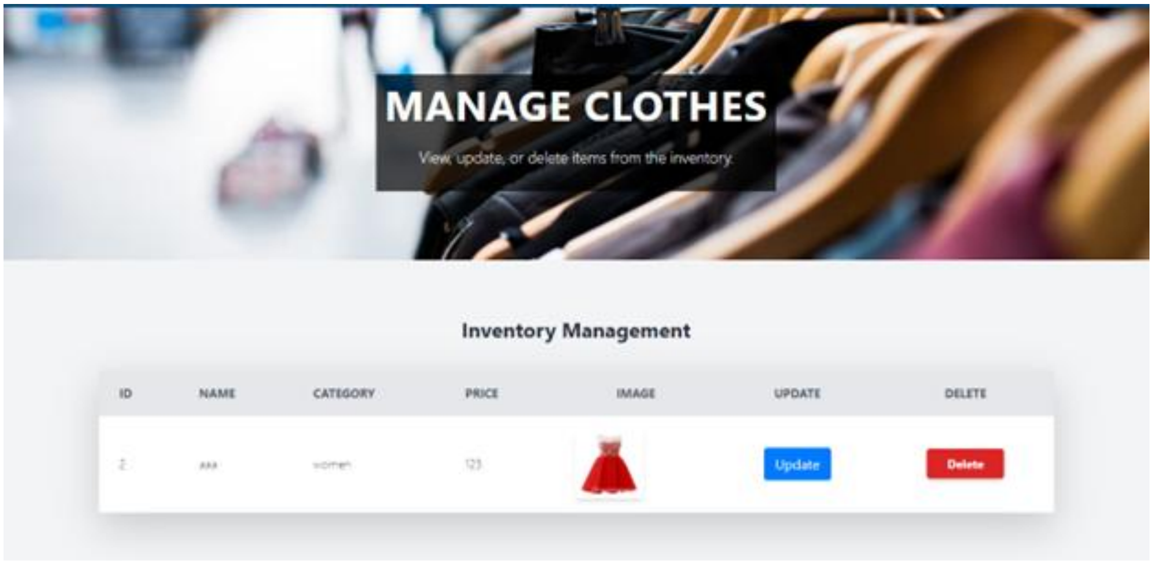
Cost of the dress

Upload Image

No file chosen

Screen 7.12 - Add Cloths Page

***MANAGE CLOTHS PAGE*:** Here admin can manage cloth details. He can update and delete.



MANAGE CLOTHES

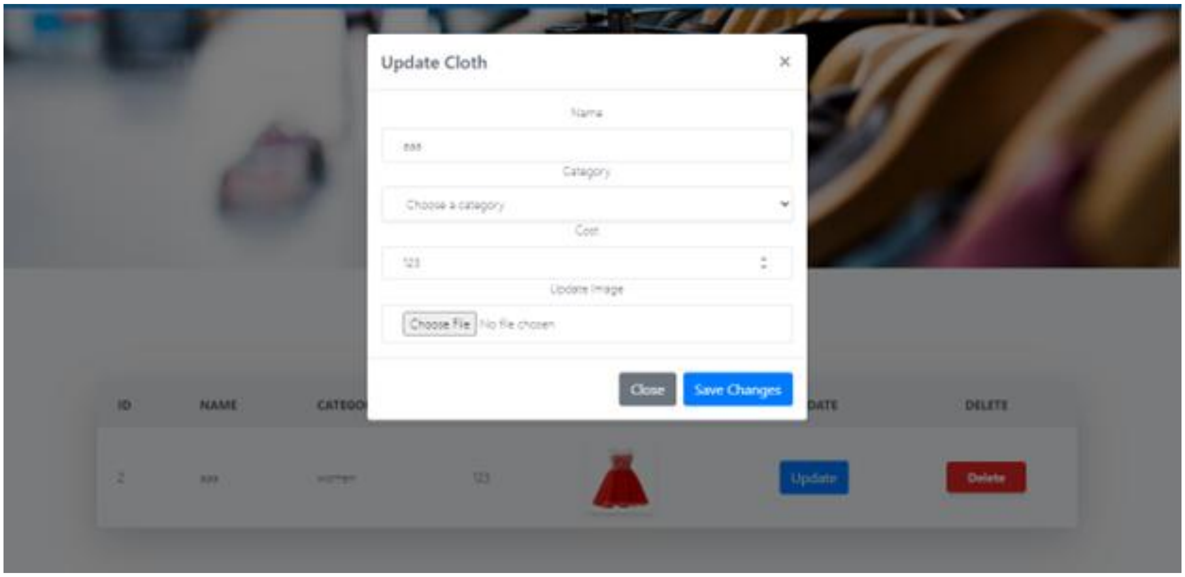
View, update, or delete items from the inventory

Inventory Management

ID	NAME	CATEGORY	PRICE	IMAGE	UPDATE	DELETE
2	aaa	women	123		Update	Delete

Screen 7.13 - Manage Cloths Page

UPDATE PAGE**:** This is the update section for a particular cloth.



Update Cloth

Name: aaa

Category: Choose a category

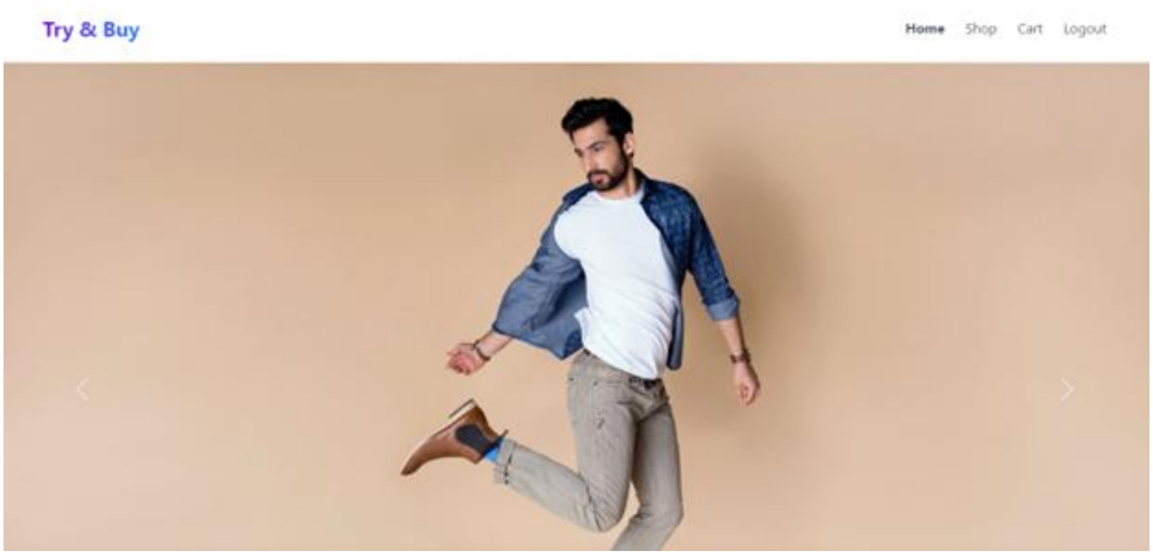
Cost: 123

Update Image: Choose File No file chosen

ID	NAME	CATEGO			DATE	DELETE
2	aaa	women	123		Update	Delete

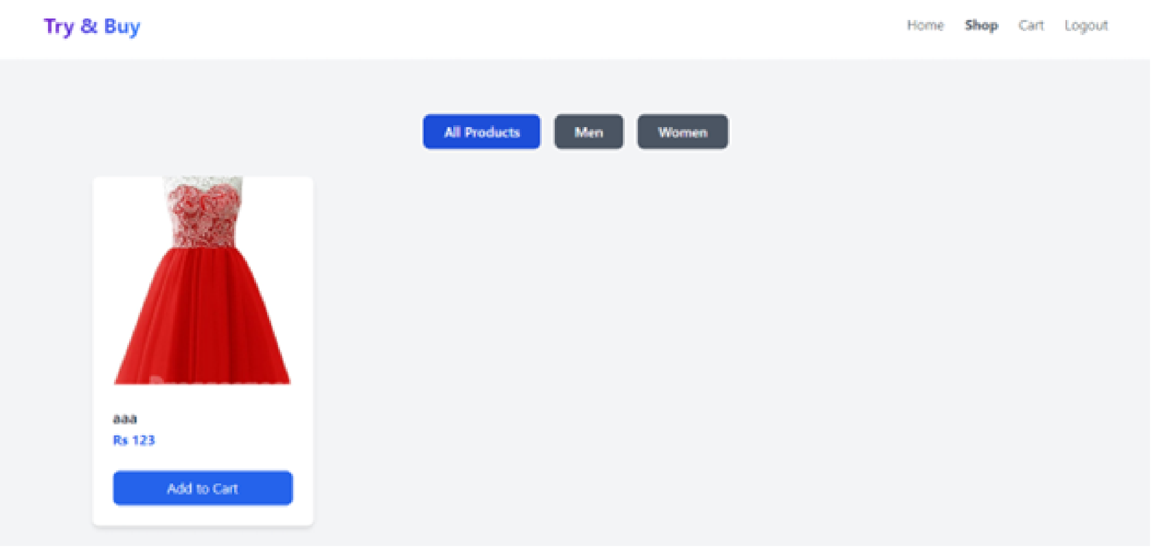
Screen 7.14 - Update Page

USER HOME PAGE**:** After user’s successful login this page will be display.



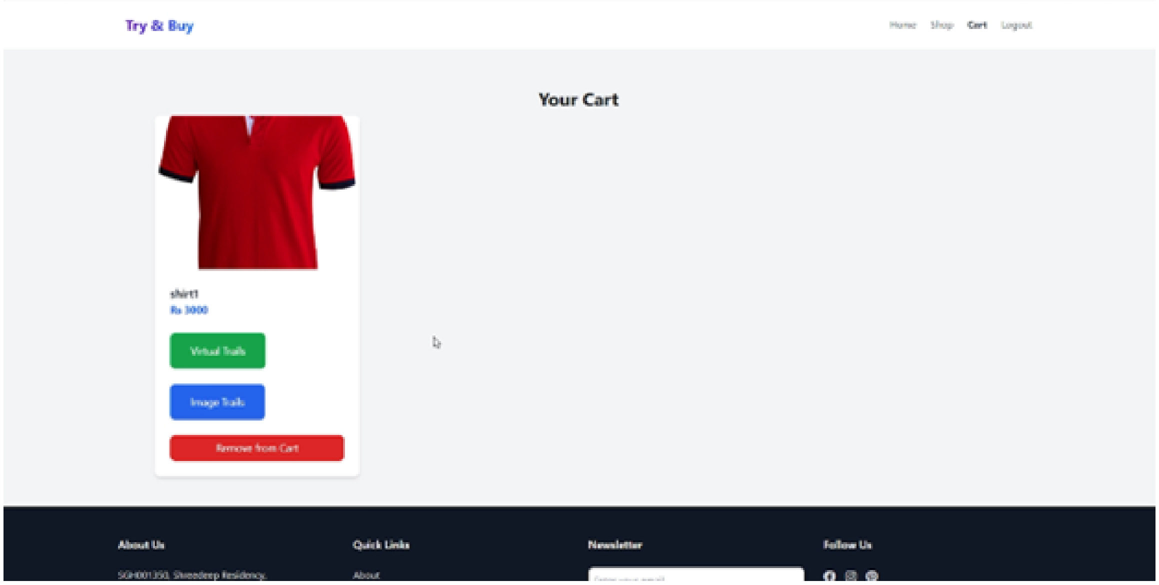
Screen 7.15 - User Home Page

***SHOP PAGE*:** Here all cloths and its information will be there. User can make it add to cart.



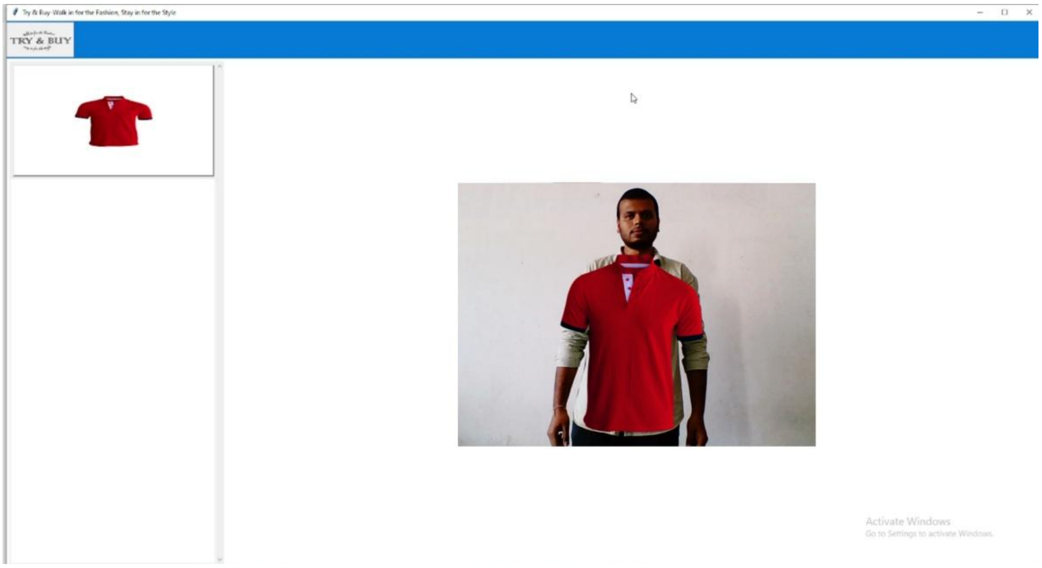
Screen 7.16 - Shop Page

CART PAGE: Here User can view the cloths which all are added by him and also, he can select a cloth for AI trail.



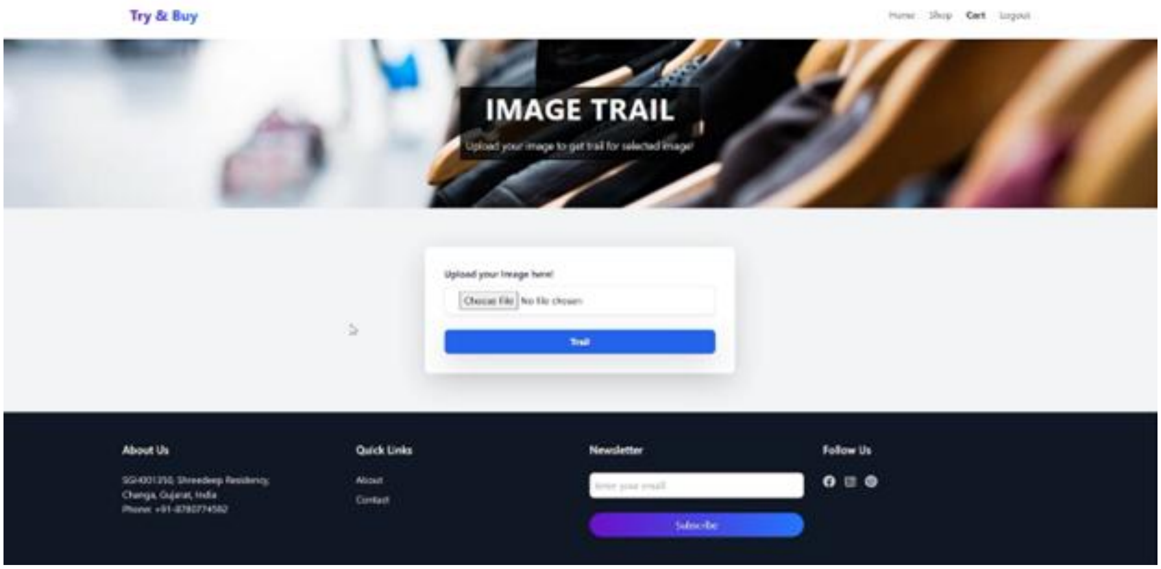
Screen 7.17 - Shop Page

***AI TRAIL ROOM PAGE*:** here user can trial with the cloth.


Try & Buy- Walk in for the Fashion, Stay in for the Style
TRY & BUY
Activate Windows
Go to Settings to activate Windows.

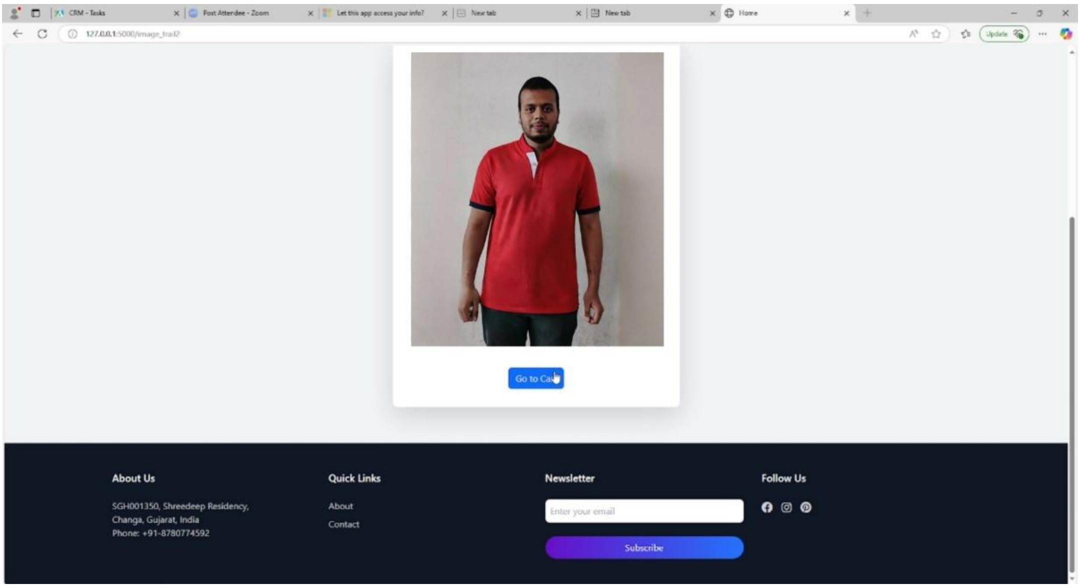
Screen 7.18 - AI Trail Room Page

***AI IMAGE TRAIL PAGE*:** here user can upload his / her image to trail with the dress.



Screen 7.19 - AI Image Trail Page

***AI IMAGE TRAIL RESULT PAGE*:** In here applied dress image will display.



SGH001350, Shreedeep Residency,
Changa, Gujarat, India
Phone: +91-8780774592

Screen 7.20 - AI Image Trail Result Pag

8.SYSTEM STUDY AND TESTING

8.1 FEASIBILITY STUDY

The feasibility study is an essential process for understanding whether the proposed system, the AI-Powered Virtual Garment Trial Room, can be successfully developed, deployed, and maintained. In this phase, we evaluate the system from multiple angles: economic, technical, and social feasibility. This ensures that all key aspects are well planned, reducing the risk of failure during the system's development or implementation stages.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Social feasibility

ECONOMICAL FEASIBILITY

The Economic feasibility is critical to ensure the system’s development stays within budget. This phase involves evaluating costs such as software licensing, hardware requirements, development costs, and ongoing maintenance. Given the project’s reliance on open-source tools like Visual Studio Code, Node.js, and XAMPP, the initial investment required for development is minimal. The project’s customizations, such as augmented reality (AR) libraries and specialized image manipulation tools, represent the primary financial investment. The development process also optimizes resources to ensure the project remains cost-effective, making it financially feasible even for small to medium-sized organizations.

TECHNICAL FEASIBILITY

For technical feasibility, we evaluate whether the proposed technologies can support the system's requirements without imposing excessive resource demands. The system incorporates multiple technologies, including Node.js for backend development, Visual Studio Code as the Integrated Development Environment (IDE), and XAMPP for local server management. These technologies are chosen because they are lightweight and compatible with standard computing environments, ensuring the system can be run on typical personal computers and servers. Furthermore, integration with AR technologies and image processing libraries has been thoroughly analyzed, with all components designed to function seamlessly together.

SOCIAL FEASIBILITY

The Social feasibility assesses how easily the system will be accepted by its target users. Since the system involves AI and augmented reality, it is crucial that the end users (clothing retailers, e-commerce companies, and customers) can comfortably adopt and use the system. This requires designing an intuitive interface and providing adequate user training. During the feasibility study, we considered the potential resistance to new technology and planned for adequate user education. To ensure ease of use, training modules, in-app tutorials, and customer support will be provided to increase user confidence. Furthermore, ongoing updates will be made based on feedback to improve user experience and ensure continued user satisfaction.

SYSTEM TESTING

System Testing is a crucial phase in the software development lifecycle, ensuring that the system not only meets its specifications but also performs reliably in real-world conditions. Given the complex nature of the AI-Powered Virtual Garment Trial Room, which integrates cutting-edge technologies such as Augmented Reality (AR), real-time video processing, and advanced image manipulation, the testing process becomes even more vital. These technologies must work seamlessly together, providing users with a smooth, intuitive experience.

Thorough testing is essential to identify any potential issues or performance bottlenecks early in the development process. It ensures that the system can handle diverse scenarios, from varying user inputs to different device capabilities. Furthermore, by performing a range of tests, we can validate that the system meets both functional and non-functional requirements, such as usability, security, and performance under load.

8.2 TYPES OF TESTING

8.2.1 UNIT TESTING

Unit testing involves validating individual components of the software to ensure they function correctly in isolation. This is essential for catching errors early and ensuring that each part of the AI-Powered Virtual Garment Trial Room works as intended before integrating with other system components.

In this project, unit testing focuses on components such as:

- Image processing for rendering virtual garments.
- Face detection to align virtual items on the user.
- Backend API services for user data processing.

Key aspects of unit testing include:

- **Test Case Design**: Ensures the component performs correctly with different inputs, such as various garment images or lighting conditions.
- **Code Coverage:** Validates that all paths in the code, including conditionals and loops, are tested.
- **Isolation:** Tests are written to ensure each unit is checked independently, minimizing dependencies on other components.
- **Automation:** Unit tests are automated to run continuously, providing quick feedback during development.
- **Regression Prevention:** Reusing unit tests throughout development ensures that new changes don’t break existing functionality.

Examples of tests include validating image processing with different inputs, checking error handling for invalid data, and verifying the proper alignment of virtual garments in varying conditions.

By conducting thorough unit testing, we ensure that the system’s components work correctly, reducing the risk of bugs and ensuring a stable foundation for integration and system testing.

8.2.2 INTEGRATION TESTING

Integration testing is a critical phase where individual software components are combined and tested as a group. It ensures that these integrated components work together as expected to form a cohesive system. In the context of the AI-Powered Virtual Garment Trial Room, integration tests verify the interaction between various system modules, such as image processing, real-time video analysis, and backend APIs.

Integration testing is primarily event-driven and focuses on:

- **Data Flow:** Ensuring data flows correctly between components like the user interface, image processing algorithms, and the backend database.
- **API Integration:** Verifying that requests and responses between the frontend and backend services are accurate and seamless.
- **User Interface Integration:** Testing how the UI components interact with the logic and backend to display the virtual garment properly in real-time.

The main goal of integration testing is to detect errors that might arise due to the interaction between different modules, which might not be apparent during unit testing. In this project, integration tests ensure that despite individual components performing correctly in isolation, their combination does not introduce inconsistencies or failures.

Examples of integration tests in this project include:

- Checking if virtual garments are accurately overlaid on the user’s image when the frontend UI interacts with the image processing module.
- Verifying that data retrieved from the backend is correctly reflected in the user interface.
- Ensuring smooth synchronization between video feed processing and real-time garment rendering.

By executing integration tests, we ensure that the system’s components work together efficiently and that the entire flow from the user’s input to the final output functions as intended.

Test Results:

All the test cases mentioned above were executed successfully, with no defects encountered during the integration testing phase.

8.2.3 FUNCTIONAL TESTING

Functional Testing involves verifying that the system operates in accordance with the defined business and technical requirements, as documented in system specifications and user manuals. It focuses on ensuring that each function of the software performs as expected and delivers the correct output for given inputs.

Functional testing is typically based on the following key aspects:

- **Valid Input:** The system must correctly accept all identified classes of valid input.
- **Invalid Input:** The system should properly reject all identified classes of invalid input.
- **Functions:** All intended functions must be triggered and executed correctly.
- **Output:** The application should produce the expected output for all tested scenarios.
- **Systems/Procedures:** All interfacing systems or dependent procedures must be properly invoked.

The organization and execution of functional tests are driven by the core requirements, critical business processes, and special or edge cases. This includes verifying:

- Business process flows
- Data fields and their validations
- Predefined and successive processes

Functional testing is usually performed during the system and acceptance testing phases and may involve both manual and automated test cases. It helps identify missing functionalities, incorrect processing, and interface errors.

As functional testing progresses, additional relevant test cases may be identified. The overall effectiveness of current test cases is also reviewed to ensure thorough coverage. Test scenarios are also updated based on user feedback and real-world conditions to simulate actual use cases more accurately.

8.2.4 WHITE BOX TESTING

White Box Testing is a software testing technique in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level. The tester uses the internal code structure to design test cases and validate the program’s flow of inputs and outputs.

It is mostly done at the unit testing level and focuses on verifying the correctness of internal operations. This includes checking loops, conditions, paths, and logic implemented in the code.

Test Objectives:

- Ensure all decision branches and conditions are executed correctly.
- Verify proper flow of data through different paths in the code.
- Identify hidden errors or unused code blocks.
- Confirm that all statements and functions are executed at least once.

White Box Testing helps in improving code quality and is usually performed by developers during early stages of testing.

8.2.5 BLACK BOX TESTING

Black Box Black Box Testing is a software testing method in which the tester evaluates the functionality of the software without any knowledge of its internal code, structure, or implementation. The system is treated as a "black box" — the focus is solely on inputs and the expected outputs, with no consideration for how the results are derived.

Test cases are derived from the software's specifications, requirements, and user expectations. This type of testing is especially useful for validating that the system behaves correctly under various scenarios.

Key objectives of Black Box Testing:

- All field entries must function as expected
- Navigational links must activate the correct pages
- System messages, alerts, and responses should be timely and appropriate
- The system should handle valid and invalid inputs correctly
- The overall user interface should be responsive and intuitive

Black Box Testing is primarily performed during system and acceptance testing phases and helps ensure that the end-user experience aligns with the requirements.

8.2.6 ACCEPTANCE TESTING

User Acceptance Testing (UAT) is a critical phase in the software development lifecycle where the end users validate the system to ensure it meets their expectations and business requirements. It is the final stage before the product is released and is typically conducted in a real-world environment that simulates actual usage.

- Acceptance Testing ensures:
- The software meets all specified functional and non-functional requirements
- The system behaves as expected under normal usage conditions
- End-users are comfortable using the system for their daily operations

Key activities in acceptance testing include:

- Creating real-world test scenarios based on actual business workflows
- Executing test cases with the involvement of end-users
- Verifying that there are no critical bugs or usability issues
- Any discrepancies found during UAT are reported for immediate resolution before deployment.

Test Results: All test cases mentioned above passed successfully. No defects encountered.

8.2.7 PERFORMANCE TESTING

Performance Testing is conducted to evaluate how well the application performs under various conditions, such as heavy load, stress, or concurrent user access. It helps identify bottlenecks, latency issues, and system limitations, ensuring that the application remains stable and responsive.

Key objectives of performance testing include:

- Determining the response time of key transactions
- Assessing system behavior under expected and peak loads
- Identifying memory leaks, crashes, or degradation under stress
- Validating system scalability and robustness

Performance testing typically includes:

- Load Testing – to measure system performance under expected user load
- Stress Testing – to test system limits beyond normal conditions
- Scalability Testing – to ensure the system can handle growth in workload
- Stability Testing – to verify the system can run continuously without failure

Testing is done using tools or scripts to simulate real-world usage patterns and multiple user interactions. It ensures that users experience smooth performance, even during high usage times or on low-spec devices.

Test Results: The application maintained stable performance across all test cases with no significant latency or resource issues observed.

8.3 TEST CASES MODEL BUILDING:

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Registration	User credentials	Successful registration	Successfully registered	P
2	Login	User’s registered credentials	Successful login	Successfully login	P
3	Add cloths details	Cloths details	Added successfully	Added successfully	P
4	Manage cloth details	Update or delete details	Successfully updated or deleted	Successfully updated or deleted	P
5	Add to cart	Add cloths to cart	Added successfully	Added successfully	P
6	AI virtual trial	Trail with dress on live camera	Image applied to the body successfully	Applied successfully	P
7	Image trail	Trail with dress on uploaded image	Image applied to the body successfully	Applied successfully	P

Table 2.3 - Test Cases Model Building

9. CONCLUSION

The Virtual Trail Room project successfully delivers an innovative web-based solution that enhances the online apparel shopping experience through virtual try-on capabilities. By integrating augmented reality (AR) using OpenCV and Dlib for real-time clothing overlays and the Segmind Try-On Diffusion API for image-based try-ons, the system effectively bridges the gap between physical and digital retail. The admin module streamlines product management, while the user module empowers customers to browse, try on, and manage purchases with ease. Built on Flask and MySQL, the application achieves its objective of improving user confidence and reducing uncertainty in online shopping, tailored for small to medium-scale retailers. Despite its focus on upper-body clothing and basic hardware requirements, it demonstrates a cost-effective, accessible approach to virtual fitting. The project lays a strong foundation for future enhancements, such as full-body detection and advanced texture refinement, promising broader applicability in e-commerce. Ultimately, the Virtual Trail Room showcases the potential of AR and image processing to transform digital apparel retail, offering a practical and engaging tool for modern shoppers and businesses alike.

10. FUTURE SCOPE OF ENHANCEMENTS

The Virtual Trail Room, while functional and effective for upper-body clothing try-ons, offers several opportunities for future development to broaden its scope and improve user experience. First, integrating full-body pose estimation using advanced models like OpenPose or MediaPipe could enable try-ons for lower-body garments (e.g., pants, skirts), providing a more comprehensive virtual fitting experience. Second, implementing a texture refinement network would enhance the realism of both AR and image-based overlays by improving garment texture mapping and edge smoothing, addressing current limitations in visual quality. Third, adding a payment gateway integration (e.g., Stripe, PayPal) would extend the system into a complete e-commerce platform, allowing users to finalize purchases directly. Fourth, enhancing the user interface with a responsive design using CSS frameworks (e.g., Bootstrap) and mobile optimization would improve accessibility across devices. Finally, deploying the application to a cloud platform (e.g., AWS, Heroku) with cloud storage for images would increase scalability, supporting larger product catalogues and user bases. These enhancements would elevate the Virtual Trail Room into a robust, market-ready solution for modern apparel retail.

11. REFERENCES

[1] E. Nitasha, S. Kumari, A. Kumar, R. Bhardwaj. (2024), Future of Fashion: AI-Powered Virtual Dressing for E-Commerce Applications.

[2] Manjula Devarakonda, Venkata. (2024), AI-Enhanced Digital Mirrors: Empowering Women's Safety and Shopping Experiences.

[3] B. S. Rochana and S. Juliet (2024), Virtual Dress Trials: Leveraging GANs for Realistic Clothing Simulation.

[4] Qinghui Wang, Na Qu. (2024), Novel AI Model for Evaluating Buyers' Fulfilment with Clothing Fit.

[5] Xu, Z., Liu, J., & Zhang, H. (2020), Fashion Forward: A Deep Learning Approach for Fashion Image Understanding.

[6] Ge, Y., Zhang, Z., Wang, W., Li, Z., Luo, P., & Tang, X. (2019), Parser-Free Virtual Try-On via Distilling Appearance Flows.

[7] Dong, H., Liang, X., Gong, K., Lai, H., Zhu, J., & Yin, J. (2019), FW-GAN: Flow-n-Warp GAN for Image-Based Virtual Try-On.

[8] Han, X., Wu, Z., Wu, Z., Yu, R., & Davis, L. S. (2018), M2E-Try On Net: Fashion from Model to Everyone.

[9] Chen, X., Xu, Z., & Zhang, L. (2018), VITON-HD: High-Definition Virtual Try-On Network.

[10] Zheng, Y., Fu, Y., & Yang, X. (2017), VITON: An Image-based Virtual Try-On Network.