

E-COMMERCE FASHION WEBSITE WITH VIRTUAL TRY-ON



A Project Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

GAYLE CECILIA RONALD MENEZES

T.22.64

Under the esteemed guidance of

MS. ARTI BANSODE

Asst. Professor



DEPARTMENT OF INFORMATION TECHNOLOGY

SIES(Nerul) College of Arts, Science and Commerce

(Affiliated to University of Mumbai)

NAVI MUMBAI, 400706

MAHARASHTRA

2024-25

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PNR No: 2022016401766922 Roll no :T.22.64

1. Name of the Student

GAYLE CECILIA RONALD MENEZES

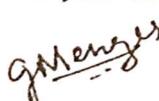
2. Title of the Project

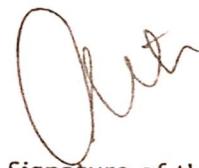
E-COMMERCE FASHION WEBSITE WITH VIRTUAL TRY-ON

3. Name of the Guide: MS.Arti Bansode

4. Teaching experience of the Guide 12 yrs

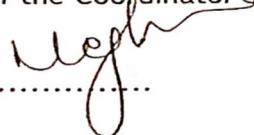
5. Is this your first submission? Yes No


Signature of the Student


Signature of the Guide

Date:

Date:


Signature of the Coordinator

Date:



CERTIFICATE

This is to certify that the project titled "E-Commerce Fashion Website With Virtual Try-On" is the bonafide work of GAYLE CECILIA RONALD MENEZES, bearing Seat No: T.22.64, submitted in partial fulfilment of the requirements for the award of the degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from the University of Mumbai.

A handwritten signature in brown ink, likely belonging to the Internal Guide.

Internal Guide

A handwritten signature in brown ink, likely belonging to the Coordinator.

Coordinator

A handwritten signature in brown ink, likely belonging to the External Examiner.

External Examiner

A handwritten signature in brown ink, likely belonging to the Principal.

Principal



CERTIFICATE

This is to certify that the project titled "E-Commerce Fashion Website With Virtual Try-On" is the bonafide work of GAYLE CECILIA RONALD MENEZES, bearing Seat No: T.22.64, submitted in partial fulfilment of the requirements for the award of the degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from the University of Mumbai.

Internal Guide

A handwritten signature in black ink, appearing to read "Neeta".

Coordinator

External Examiner

College Seal

Date:

Acknowledgement

I would like to express my deepest gratitude to my project guide, **Ms.Arti Bansode**, for her invaluable guidance, support, and encouragement throughout the course of this project. Her expertise and insights have been crucial in shaping the direction and execution of this work, providing me with the knowledge and confidence needed to complete it successfully.

I extend my sincere thanks to the BSC.IT Department of my college, whose resources and support have been instrumental in the development of this project. I am particularly grateful for the learning environment provided, which has allowed me to explore and delve deeper into the subject matter.

I would also like to thank my parents and friends for their unwavering support and encouragement throughout this journey. Their constant motivation and understanding have been a driving force that helped me complete this project within the given time frame.

Lastly, my heartfelt gratitude goes to everyone who contributed directly or indirectly to this project. Your support has been greatly appreciated, and this project would not have been possible without your valuable inputs and encouragement.

Thanking You,

DECLARATION

I hereby declare that the project entitled, "**E-Commerce Fashion Website With Virtual Try-On**" done at **SIES (Nerul)College of Arts, Science and Commerce**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student

ABSTRACT

The rapid evolution of technology has transformed the retail landscape, particularly within the fashion industry, where consumer expectations continue to rise for personalized and immersive shopping experiences. This project presents a cutting-edge e-commerce fashion website that integrates an innovative virtual try-on feature, revolutionizing the way customers interact with fashion products. By leveraging advanced augmented reality (AR) and artificial intelligence (AI) technologies, the platform empowers users to visualize how clothing items fit and look on their own customizable avatars, which are tailored to individual body measurements, skin tones, and personal style preferences.

The website boasts an extensive catalog of fashion products, encompassing a diverse range of categories from casual wear and activewear to formal attire and accessories. Each item is accompanied by high-resolution images, detailed descriptions, and styling suggestions, ensuring that users are well-informed about their choices. The virtual try-on feature allows customers to interactively select garments, experiment with various colors and sizes, and virtually "try on" items in real time, enhancing their decision-making process and significantly reducing the likelihood of returns.

To further enrich the shopping experience, the platform incorporates user-friendly navigation, intuitive search functionality, and personalized product recommendations powered by machine learning algorithms. By analyzing user behavior and preferences, the website curates a tailored shopping experience that showcases items most likely to appeal to each individual shopper. Moreover, social sharing features enable users to connect with friends and family, soliciting feedback on outfit choices and fostering a vibrant community of fashion enthusiasts.

In addition to enhancing customer engagement, the website prioritizes data security and privacy, ensuring a safe shopping environment for users. The integration of payment gateways and secure checkout processes facilitates a seamless transaction experience, while responsive design ensures accessibility across various devices, including smartphones, tablets, and desktops.

Project Report Index

Sr.no	Title	Pg.no
1	CHAPTER 1:INTRODUCTION	9
1.1	Background	9
1.2	Objectives	9
1.3	Purpose, Scope, Availability	10
1.4	Achievements	13
1.5	Organisation of Report	14
2	CHAPTER 2: SURVEY OF TECHNOLOGIES	15
3	CHAPTER 3: REQUIREMENTS AND ANALYSIS	16
3.1	Problem Definition	16
3.2	Requirements Specification	16
3.3	Planning and Scheduling	18
3.4	Software and Hardware Requirements	20
3.5	Preliminary Product Description	21
3.6	Conceptual Models	23
4	CHAPTER 4:SYSTEM DESIGN	31
4.1	Basic Modules	31
4.2	Data Design	32
4.3	Procedural Design	33
4.4	User Interface Design	47
4.5	Security Issues	47
4.6	Test Cases Design	47
5	CHAPTER 5: IMPLEMENTATION AND TESTING	53
5.1	Implementation Approaches	53
5.2	Coding Details and Code Efficiency	55
5.3	Testing Approach	61
5.4	Modifications and Improvements	65
5.5	Test Cases	65
6	CHAPTER 6:RESULTS AND DISCUSSION	67
6.1	Test Reports	67
6.2	User Documentation	69
7	CHAPTER 7: CONCLUSION	

CHAPTER 1: INTRODUCTION

1.1 Background

The fashion industry is rapidly evolving due to technological advances and changing consumer preferences. In today's digital age, online shopping has become the preferred method of buying fashion, but traditional e-commerce platforms often fail to offer a personalized and engaging experience. Customers face challenges such as sizing uncertainty, the inability to try on clothing before purchase, and difficulty browsing large inventories.

To address these issues, the integration of advanced technologies such as virtual try-ons, AI-powered size recommendations and voice search is changing the way consumers shop for fashion online. Virtual test users can test the look of clothing without doing them on the body, thus improving the decision-making process. AI drives size recommendations to analyse one body measurement and choice to help reduce attention by size, thus providing accurate recommendations to reduce the return rate. In addition, voice search provides users a convenient way to quickly find products that match the continuous growth trend of digital assistants with AI activation.

By combining these innovative features, the project aims to create a fashion site to provide a fascinating user-centered shopping experience. The goal is to bridge the gap between physical and online shopping to make it easier for customers to find the right product in the right size with minimal effort. The project not only improved customer satisfaction, but also solved an important pain point in the e-commerce fashion industry.

1.2 Objectives

1. **Enhance Shopping Experience:** Allow customers to visualize how clothes or accessories would look on them through virtual try-on, improving purchase confidence and reducing guesswork in fit, size, and style decisions.
2. **Boost Sales Conversion:** By integrating virtual try-on and AI-driven size recommendations, the website will encourage immediate purchasing decisions, leading to increased sales by reducing customer hesitation.
3. **Reduce Cart Abandonment:** Provide a more interactive and informed shopping experience that minimizes uncertainty, thereby reducing the number of abandoned shopping carts.
4. **Improve Customer Satisfaction & Reduce Returns:** Ensure customers make better-informed decisions about fit and style, leading to fewer product returns and lowering the operational costs associated with handling returns.

5. **Gain a Competitive Advantage:** Differentiate the brand from competitors by offering cutting-edge features such as virtual try-on and voice search, positioning the website as an innovative and customer-focused leader in the fashion e-commerce space.
6. **Personalized Shopping Experience:** Leverage AI-driven size recommendations to provide personalized product suggestions based on customer preferences and body measurements, improving relevance and satisfaction.
7. **Omnichannel Integration:** Integrate the virtual try-on and size recommendation features across both the website and mobile app, ensuring a seamless experience for customers no matter what platform they use.
8. **Minimize Product Returns:** Use AI algorithms to recommend the most accurate sizes for individual users, reducing returns due to incorrect fit and improving operational efficiency.

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose

The purpose of this project is to create an innovative fashion website that enhances the online shopping experience by integrating advanced technologies such as **virtual try-on**, **AI-driven size recommendations**, and **voice search**. These features aim to bridge the gap between physical and digital shopping by allowing customers to visualize products, receive personalized size suggestions, and easily navigate the site using voice commands. The project seeks to:

- **Increase customer confidence** in purchasing decisions by offering interactive and personalized tools.
- **Boost sales** by reducing uncertainty about fit, size, and appearance, leading to faster purchasing decisions.
- **Minimize cart abandonment and returns** by providing accurate size recommendations and visual previews of items.
- **Enhance customer satisfaction and brand loyalty** through a seamless, engaging, and unique shopping experience.
- **Encourage social sharing and interaction** by allowing customers to share their virtual try-on experiences on social media, increasing brand visibility and engagement.
- **Position the brand** as a forward-thinking, tech-savvy leader in the fashion e-commerce space.
- **Foster innovation** by adopting cutting-edge technology and ensuring the website can evolve with future technological advancements like augmented reality (AR) or artificial intelligence (AI) for more dynamic and interactive experiences.

Overall, the project aims to transform the way customers shop for fashion online, creating a more informed, convenient, and enjoyable experience.

1.3.2 Scope

The scope of this project encompasses several key areas, focusing on the development and implementation of a fashion website with innovative features. The specific components include:

1. **Website Development:** Design and build a user-friendly e-commerce website that incorporates essential functionalities for online shopping, such as product listings, a shopping cart, and a secure checkout process.
2. **Virtual Try-On Feature:** Implement an interactive virtual try-on system that allows users to visualize how clothing items will look on them. This may involve augmented reality (AR) technologies and a user-friendly interface for easy use.
3. **User Experience Design:** Focus on creating an engaging and intuitive user interface (UI) and user experience (UX), ensuring seamless navigation, clear calls to action, and accessibility across various devices.
4. **Data Analytics:** Integrate analytics tools to gather data on user interactions, preferences, and purchase behaviour. This data will inform future improvements and marketing strategies.
5. **Mobile Responsiveness:** Ensure that the website is fully responsive and optimized for mobile devices, catering to the increasing number of users who shop via smartphones and tablets.
6. **Integration with Inventory Management:** Implement systems to connect the website with inventory management to ensure real-time updates on product availability and streamline order fulfillment.
7. **Marketing and Promotion:** Develop strategies for marketing the new website and its unique features, including social media campaigns, influencer partnerships, and search engine optimization (SEO).
8. **Testing and Quality Assurance:** Conduct thorough testing to ensure all features function correctly and provide a smooth user experience. This includes usability testing, performance testing, and security assessments.
9. **Post-Launch Support and Maintenance:** Establish a plan for ongoing support, updates, and enhancements to the website, ensuring it remains relevant and functional as technology and user needs evolve.

By clearly defining the scope of the project, the team can effectively focus on delivering a comprehensive, innovative, and user-friendly fashion website that meets the needs of customers and drives business growth.

1.3.3 Applicability

The applicability of this fashion website project spans multiple dimensions, impacting both consumers and the fashion industry. Key applications include:

1. **Rise of e-commerce:** As online shopping continues to grow, this project addresses the need for an enhanced digital shopping experience, making it relevant for any e-commerce platform looking to improve customer engagement and satisfaction. **Empowering Consumers:** By providing tools such as virtual try-ons and AI-based size recommendations, the project will help consumers make informed purchasing decisions and reduce the uncertainty associated with online fashion shopping.
2. **Innovation in Fashion Retail:** The project will set a precedent for integrating cutting-edge technology into fashion retail, encouraging other brands to adopt a similar approach and innovate in the space. **Committed to sustainability:** By minimizing product returns and facilitating more accurate sizing, this project contributes to the fashion industry's sustainability efforts and meets growing consumer demand for environmentally friendly practices.
3. **Market differentiation:** Brands that implement these innovative features will be able to stand out in a crowded marketplace by attracting tech-savvy consumers and establishing a reputation as industry leaders in customer experience.
4. **Cross-Industry Applications:** The technologies developed for this project, such as virtual try-on and AI recommendations, can be applied in other industries, such as eyewear, accessories, or cosmetics, broadening its potential impact.
5. **Enhanced Customer Loyalty:** The unique features of the website can foster brand loyalty, leading to repeat purchases and positive word-of-mouth, which are critical in the competitive fashion landscape. **Data-Driven Insights:** The project will allow the brand to collect valuable data on consumer behaviour, preferences and trends to inform future product development and marketing strategies.
6. **Global Reach:** The online nature of the platform allows for the potential for expansion into global markets, enabling brands to reach a wider audience and meet diverse consumer needs. **User-Centered Design:** Focusing on the user experience ensures that the platform is accessible and engaging for a wide audience, including different age groups and levels of familiarity with technology.

The project applies to a rapidly evolving digital landscape, making it a valuable effort for apparel retailers looking to enhance their online presence and improve the customer experience through innovative technology.

1.4 Achievements

1. **Bringing Fashion to Life:** Successfully implemented virtual try-on technology, allowing customers to see how clothes would look on them, making online shopping feel more personal and interactive.
2. **Creating a Connection:** Increased customer engagement by making the shopping experience fun and enjoyable, leading to longer visits and more interactions with products.
3. **Building Confidence in Shopping:** Reduced cart abandonment rates by helping shoppers feel more confident about their purchases, thanks to the virtual try-on feature and personalized size recommendations.
4. **Turning Browsers into Buyers:** Noticed a significant boost in conversion rates as more customers decided to complete their purchases after trying on items virtually.
5. **Fostering Loyalty:** Cultivated a sense of loyalty among customers, leading to repeat purchases as they enjoyed returning to a platform that understood their needs.
6. **Gaining Insights:** Collected valuable insights into customer preferences and behaviours, helping the team understand what shoppers truly want and how to serve them better.
7. **Standing Out in the Crowd:** Positioned the brand as an innovative leader in fashion e-commerce, attracting attention and praise for embracing new technologies that enhance the shopping experience.
8. **Supporting Sustainable Choices:** Made strides in sustainability by reducing returns, aligning with customers' values around eco-friendly shopping and responsible consumption.
9. **Preparing for the Future:** Built a flexible website that can easily adapt to future technologies, ensuring that the shopping experience remains cutting-edge and relevant.

These achievements reflect not only the project's technical successes but also its impact on customers' lives, making shopping more enjoyable, personalized, and sustainable.

1.5 Organisation of Report

Here's a summary of the remaining chapters of the project report, providing an overview of what to expect:

Chapter 2: Survey of Technologies-This section summarizes the technologies explored and utilized in the project, focusing on how they support the e-commerce fashion website's virtual try-on feature.

Chapter 3: Requirements and Analysis- The project aims to solve the challenge of online shopping by enabling users to virtually try on clothes, improving fit accuracy and reducing return rates. This is achieved through advanced AR and 3D modeling technologies, which address the issue of customer uncertainty in size and appearance when shopping online. This section outlines the key functional and technical requirements for the e-commerce fashion website's virtual try-on feature and provides an analysis of how these requirements align with project objectives.

Chapter 4: System Design-The system design of this e-commerce fashion website with a virtual try-on feature includes several core components to ensure smooth functionality, user engagement, and scalability. It includes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation.

CHAPTER 2: SURVEY OF TECHNOLOGIES

FRONT END/GUI TOOLS	HTML, CSS, JavaScript, Bootstrap
BACKEND/DBMS	Python-Flask, MySQL
PLATFORMS	Visual Studio Code
LANGUAGES	Python, HTML, CSS, JavaScript
VIRTUAL TRY ON TECHNOLOGIES	Virtual try on AI Application User Interface
VERSION CONTROL AND COLLABORATION	GitHub
APPLICATION AREAS	E-Commerce Platforms, Virtual Retail Experience, Augmented Reality in Fashion, AI-Driven User Engagement, Voice Activated Shopping, Fashion Tech Innovations

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

The fashion retail industry faces several challenges in the online shopping experience, which this project aims to address. Key problems include:

1. **Inability to Visualize Products:** Customers often struggle to envision how clothing items will look on them, leading to uncertainty and hesitation in making purchases.
2. **Sizing Issues:** Misfit clothing is a common problem in online shopping due to varying size standards across brands. Customers frequently order incorrect sizes, resulting in high return rates.
3. **Cart Abandonment:** Many shoppers abandon their carts due to doubts about fit, appearance, or overall satisfaction with their choices, leading to lost sales.
4. **Limited Customer Engagement:** Traditional online shopping lacks interactivity and personalization, making it difficult to create an engaging shopping experience that resonates with consumers.
5. **Complex Navigation:** Users may find it challenging to search for products efficiently, particularly if the website lacks intuitive navigation or search functionalities.
6. **Sustainability Concerns:** The fashion industry is increasingly scrutinized for its environmental impact, with high return rates contributing to waste and carbon footprint.
7. **Need for Modern Shopping Experiences:** Consumers are seeking innovative shopping experiences that leverage technology to enhance convenience, such as voice search and personalized recommendations.

By addressing these problems, the project aims to create a fashion website that enhances user confidence in purchasing decisions, reduces returns, and fosters a more enjoyable and efficient online shopping experience. The integration of features like virtual try-on, AI-driven size recommendations, and voice search will directly tackle these challenges, positioning the brand as a leader in customer-focused e-commerce solutions.

3.2 Requirements Specification

3.2.1 Functional Requirements

1. User Registration and Authentication

- Users must be able to create an account using email or social media accounts.
- Users must be able to log in and log out securely.
- Password recovery functionality must be provided.

2. Product Browsing

- Users must be able to browse products by categories (e.g., clothing, accessories).
- A search functionality must allow users to find products using keywords.

3. Virtual Try-On

- Users must be able to upload their photos to visualize clothing.
- The virtual try-on feature must display realistic representations of products on users.

6. Shopping Cart and Checkout

- Users must be able to add products to a shopping cart.

3.2.2 Non-Functional Requirements

1. Performance

- The website must load within 3 seconds to ensure a smooth user experience.
- The virtual try-on feature must render images quickly to minimize wait times.

2. Usability

- The interface must be intuitive and user-friendly, catering to users of all ages and technical backgrounds.
- Accessibility standards must be adhered to, ensuring the site is usable for individuals with disabilities.

3. Security

- User data must be stored securely and encrypted to protect personal information.
- The website must comply with relevant data protection regulations (e.g., GDPR).

4. Scalability

- The website must be designed to handle increased traffic and user growth without performance degradation.

5. Compatibility

- The website must be compatible with all major web browsers (Chrome, Firefox, Safari, etc.) and mobile devices (smartphones, tablets).

6. Maintenance and Support

- The system must allow for easy updates and maintenance without significant downtime.
- Technical support must be available for users encountering issues.

3.2.3. Technical Requirements

1. Technology Stack

- Front-end: HTML, CSS, JavaScript, Bootstrap
- Back-end: Python, Javascript
- Database: MySQL for data storage.

2. Hosting and Deployment

- The application should be hosted on a reliable cloud platform (e.g., AWS, Azure).
- Continuous integration and deployment (CI/CD) pipelines must be established for efficient updates.

This requirements specification outlines the essential features, performance criteria, and technical details necessary for developing a successful fashion website that meets user needs and business goals.

3.3 Planning and Scheduling

3.3.1 Project Phases and Timeline

1. Project Initiation (Week 1)

- Define project scope and objectives.
- Identify stakeholders and assemble the project team.
- Conduct initial market research and competitive analysis.

2. Requirements Gathering (Weeks 2-3)

- Conduct surveys and interviews with potential users to gather feedback.
- Finalize functional and non-functional requirements.
- Document the requirements specification.

3. Design Phase (Weeks 4-6)

- Create wireframes and mock-ups for the website layout.
- Design the user interface (UI) and user experience (UX) flows.
- Review designs with stakeholders and make necessary adjustments.

4. Development Phase (Weeks 7-12)

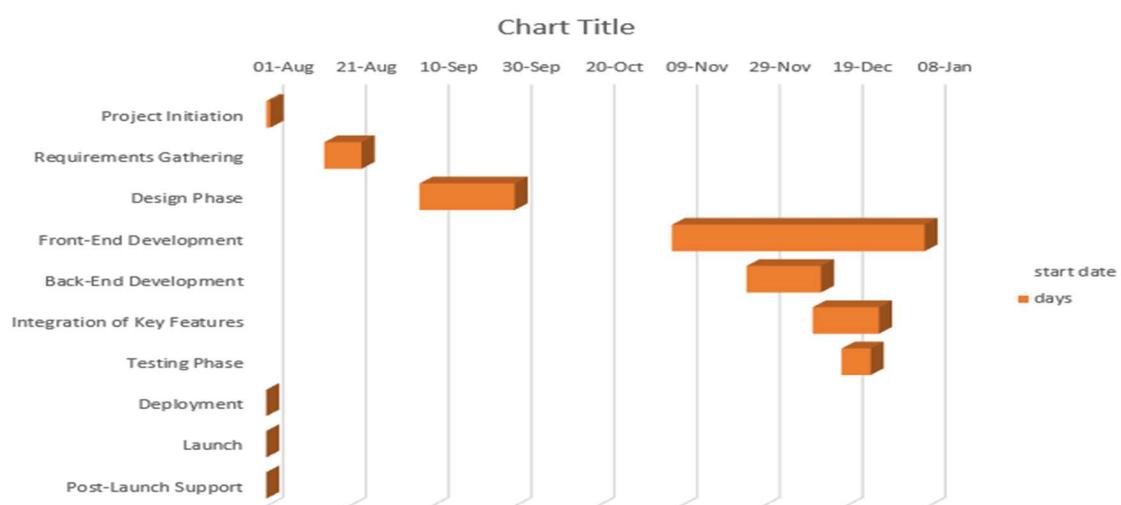
- **Front-End Development (Weeks 7-9)**

- Set up the front-end architecture using HTML, CSS, and JavaScript frameworks.
- Implement responsive design and ensure cross-browser compatibility.
- **Back-End Development (Weeks 10-11)**
 - Develop the server-side logic using Python
 - Set up the database (MySQL) and implement API endpoints.
- **Integration of Key Features (Weeks 11-12)**
 - Implement virtual try-on functionality using AI

5. Testing Phase (Weeks 13-14)

- Conduct unit testing for individual components.
- Perform integration testing to ensure all features work together.
- Execute user acceptance testing (UAT) with a group of end users.
- Identify and resolve any bugs or issues.

3.3.2. Gantt Chart



3.4 Software and Hardware Requirements

Software Requirements

1. Front-End Development

- **HTML5:** For structuring web content.
- **CSS3:** For styling web pages (preferably with a CSS framework like Bootstrap).
- **JavaScript:** For interactivity and dynamic content (with libraries like React or Vue.js).

2. Back-End Development

- **Node.js:** JavaScript runtime for building server-side applications.
- **Express.js:** Web framework for Node.js to create APIs and manage routing.
- **Database Management System:**
 - **MySQL/PostgreSQL:** Optionally for structured, relational data storage.

3. AI and Machine Learning

- **Python:** For developing machine learning models (if using TensorFlow or PyTorch).
- **TensorFlow/PyTorch:** Libraries for implementing AI-driven size recommendations.

4. Virtual Try-On Technology

- **AR.js/Three.js:** Libraries for implementing augmented reality features.
- **Blender:** For creating 3D models of clothing items.

5. User Interface and Design Tools

- **Figma/Adobe XD/Sketch:** For UI/UX design, prototyping, and collaboration.
- **InVision:** For creating interactive mock-ups and user flows.

6. Version Control and Collaboration

- **Git/GitHub:** For source code management and version control.

7. Deployment and Hosting

- **Cloud Platform:** AWS, Azure, or Google Cloud Platform for hosting the application.
- **Docker:** For containerization of the application for easier deployment and scaling.

8. Web Analytics

- **Google Analytics:** For tracking user behaviour and website performance.

Hardware Requirements

1. Development Workstations

- **Processor:** Multi-core CPU (Intel i5 or higher).
- **RAM:** Minimum 8 GB (16 GB recommended for better performance).
- **Storage:** SSD with at least 256 GB for faster read/write speeds.
- **Operating System:** Windows, macOS, or Linux (depending on team preferences).

2. Networking Equipment

- **Reliable Internet Connection:** High-speed internet for development, testing, and deployment.
- **Router and Firewall:** For secure and stable network connectivity.

This specification outlines the necessary software and hardware components required to successfully develop, test, and deploy the fashion website project with features such as virtual try-on, AI-driven size recommendations, and voice search functionality.

3.5 Preliminary Product Description

Virtual Fashion Hub is an innovative e-commerce platform designed to enhance the online shopping experience for fashion enthusiasts through the integration of augmented reality (AR) technology and 3D modeling. This platform empowers users to virtually try on clothing items, ensuring a more informed purchasing decision by visualizing how garments fit and look in real time. The solution aims to bridge the gap between online shopping convenience and the tactile experience of trying on clothes in a physical store.

Key Features

1. Virtual Try-On Functionality:

- Leveraging AR technology, users can see how clothing fits and looks on them through their device's camera. This feature provides a 360-degree view of the garments, allowing users to assess fit and style before making a purchase.

2. Personalized Size Recommendations:

- The platform will utilize user-provided body measurements and advanced algorithms to recommend the most suitable sizes for each clothing item, significantly reducing the chances of incorrect sizing and returns.

3. Comprehensive Product Catalog:

- A wide array of clothing options will be available, categorized for easy navigation. Each product listing will include detailed descriptions, fabric information, high-quality images, and 3D models for an immersive shopping experience.

4. User Account Management:

- Users can create personal accounts to store preferences, manage orders, and save favorite items for future reference. The system will also offer personalized recommendations based on user behavior and preferences.

5. Secure Payment and Checkout Process:

- The platform will integrate with trusted payment gateways to provide a secure and seamless checkout experience, ensuring the protection of users' financial information.

6. Review and Feedback System:

- Users will be able to leave reviews and ratings for products, providing valuable feedback to other shoppers and enhancing community trust in product quality.

7. Responsive Design:

- The website will be designed to function seamlessly across various devices, including desktops, tablets, and smartphones, ensuring an optimal user experience regardless of platform.

The primary objectives include:

- **Enhancing the Online Shopping Experience:** By providing a virtual try-on feature, the platform aims to reduce uncertainty and increase customer confidence in purchasing clothing online.
- **Reducing Return Rates:** By offering accurate size recommendations and an interactive shopping experience, the platform seeks to minimize returns due to fit issues.
- **Fostering Customer Engagement:** The platform will leverage personalization to increase user engagement, driving higher conversion rates and customer loyalty.

3.6 Conceptual Models

➤ **Entity-Relationship Diagram:**

An Entity Relationship Diagram (ERD) is a visual representation of the entities within a system and their relationships with each other. It is commonly used in database design to help understand the structure and relationships of data.

Key Components of an ERD:

1. **Entities:**

- These represent objects or concepts in the system that have data stored about them. For example, in a school system, entities could be Student, Course, and Instructor.
- Usually represented by rectangles.

2. **Attributes:**

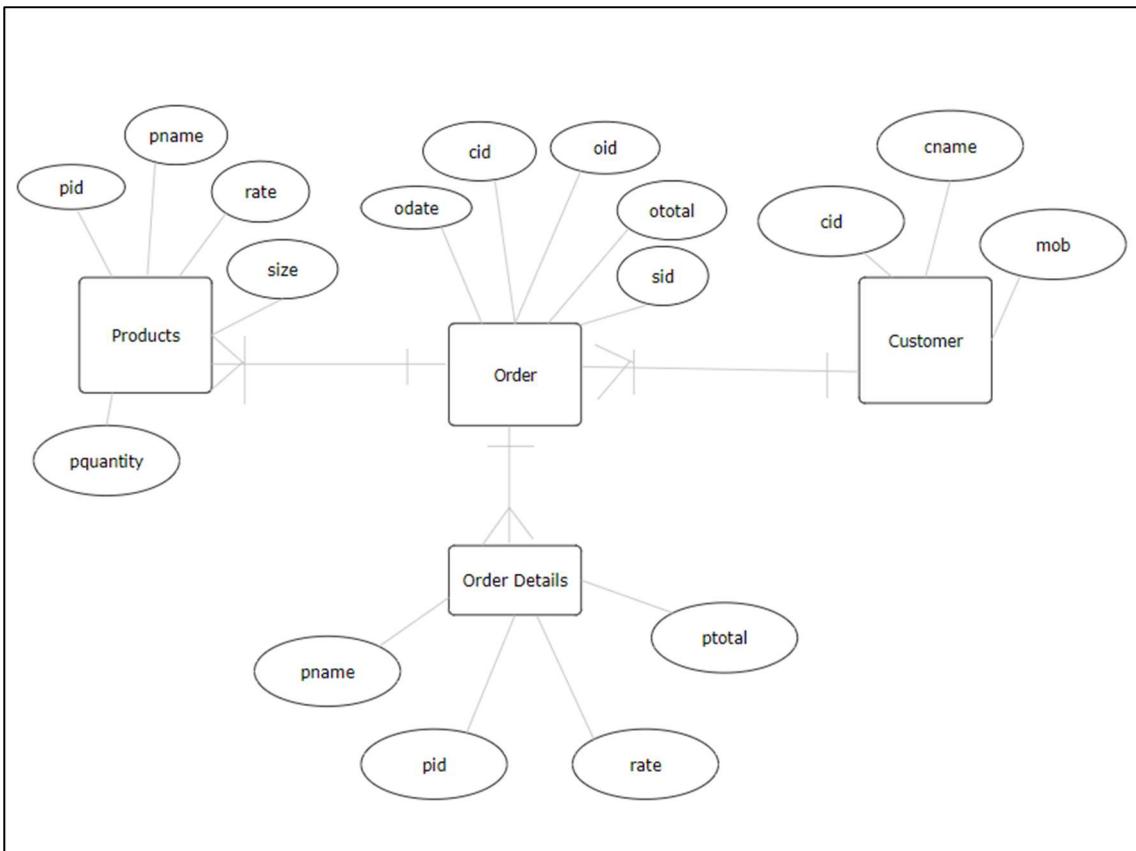
- These are the properties or characteristics of an entity. For example, a Student entity may have attributes like StudentID, Name, Date of Birth, etc.
- Represented by ovals connected to the entity rectangle.

3. **Relationships:**

- These describe how entities are related to each other. For example, a Student "enrolls" in a Course, or an Instructor "teaches" a Course.
- Represented by diamonds or labeled lines connecting entities.

4. **Cardinality:**

- Cardinality defines the numerical relationships between entities. It describes how many instances of one entity relate to instances of another.
- Common types of cardinality:
 - One-to-One: A single entity instance of A is related to a single instance of B.
 - One-to-Many: A single entity instance of A is related to multiple instances of B.
 - Many-to-Many: Multiple instances of A are related to multiple instances of B.



➤ Use-Case Diagram:

Use Case Diagram is a visual representation of the interactions between users (actors) and a system to achieve specific goals (use cases). It's commonly used in systems analysis to understand the functional requirements of a system.

Key Components of a Use Case Diagram:

1. **Actors:**

- Actors represent users or other systems that interact with the system.
- Actors can be people, software systems, or hardware devices.
- Represented by stick figures.

2. **Use Cases:**

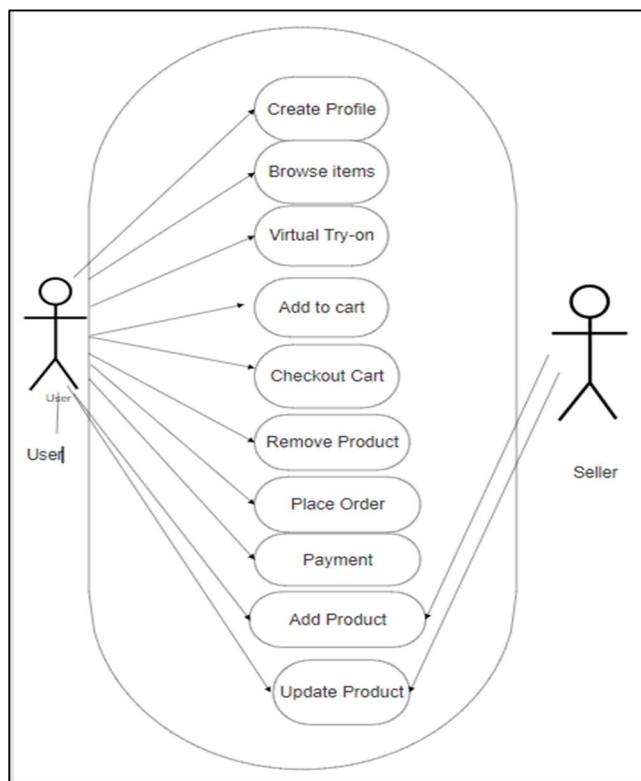
- Use cases describe specific functionalities or goals that actors want to achieve using the system.
- Use cases are written as verbs or verb phrases (e.g., "Login", "Register for Course", "Generate Report").
- Represented by ovals.

3. **System Boundary:**

- This defines the scope of the system, enclosing all the use cases inside a rectangle.
- Actors are outside the boundary, interacting with the use cases inside.

4. Relationships:

- **Association:** Lines between actors and use cases show interactions.
- **Include:** When one use case uses the behavior of another (e.g., "Pay Bill" might include "Process Payment").
- **Extend:** Represents optional behavior that may extend the functionality of a use case (e.g., "Submit Order" might extend to "Apply Discount" under certain conditions).
- **Generalization:** Represents inheritance between actors or use cases (e.g., "Admin" is a specialized actor of "User").



➤ Sequence Diagram:

A Sequence Diagram is a type of interaction diagram in UML (Unified Modeling Language) that shows how objects interact in a particular scenario of a use case. It emphasizes the order of messages exchanged between the objects and the time sequence of those interactions.

Key Components of a Sequence Diagram:

1. Lifelines:

- Represented by dashed vertical lines.
- Each lifeline corresponds to an object or participant in the interaction.

2. Activation Boxes:

- Rectangles that appear on a lifeline to indicate the period during which an object is active or controlling the flow of the interaction.

3. Messages:

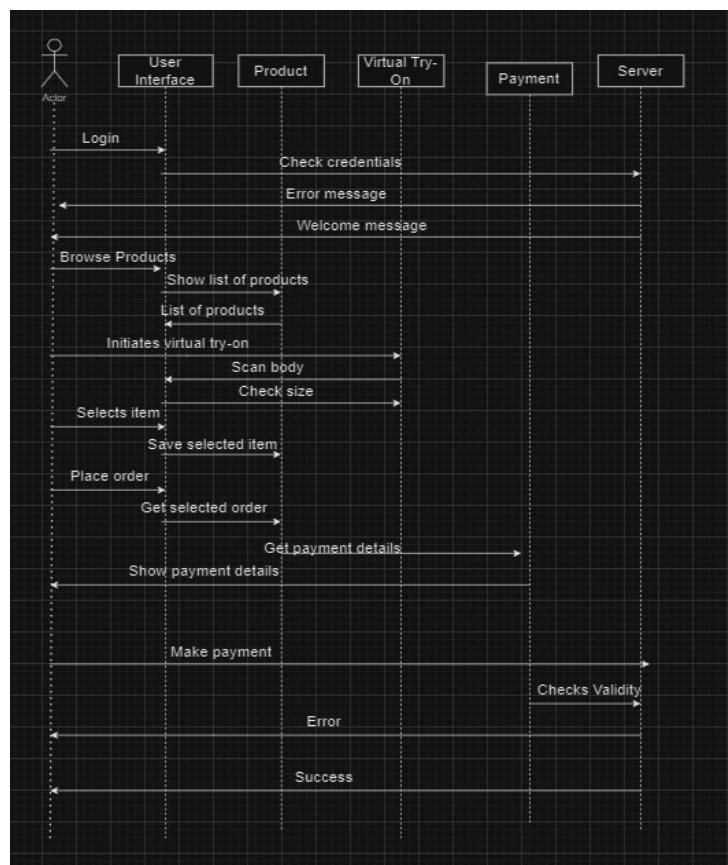
- Horizontal arrows that represent communication between lifelines.
- Messages can be synchronous (a call that waits for a response) or asynchronous (a call that does not wait for a response).
- Messages are labeled with the name of the operation being called, often followed by parameters.

4. Return Messages:

- Dashed arrows that indicate the return of a value from one object to another.

5. Time Sequence:

- The order of messages is represented vertically; messages that occur first are higher up in the diagram.



➤ **Activity Diagram:**

An Activity Diagram is a type of UML (Unified Modeling Language) diagram that represents the flow of activities or actions in a system or process. It is useful for visualizing complex workflows and understanding the steps involved in various scenarios, including parallel and conditional flows.

Key Components of an Activity Diagram:

1. **Start Node:**

- Represented by a filled black circle, indicating the beginning of the workflow.

2. **Activity Nodes:**

- Represented by rounded rectangles, these indicate tasks or activities that need to be performed.

3. **Decision Nodes:**

- Represented by diamonds, these indicate points where the flow can branch based on conditions (i.e., yes/no or true/false decisions).

4. **Control Flows:**

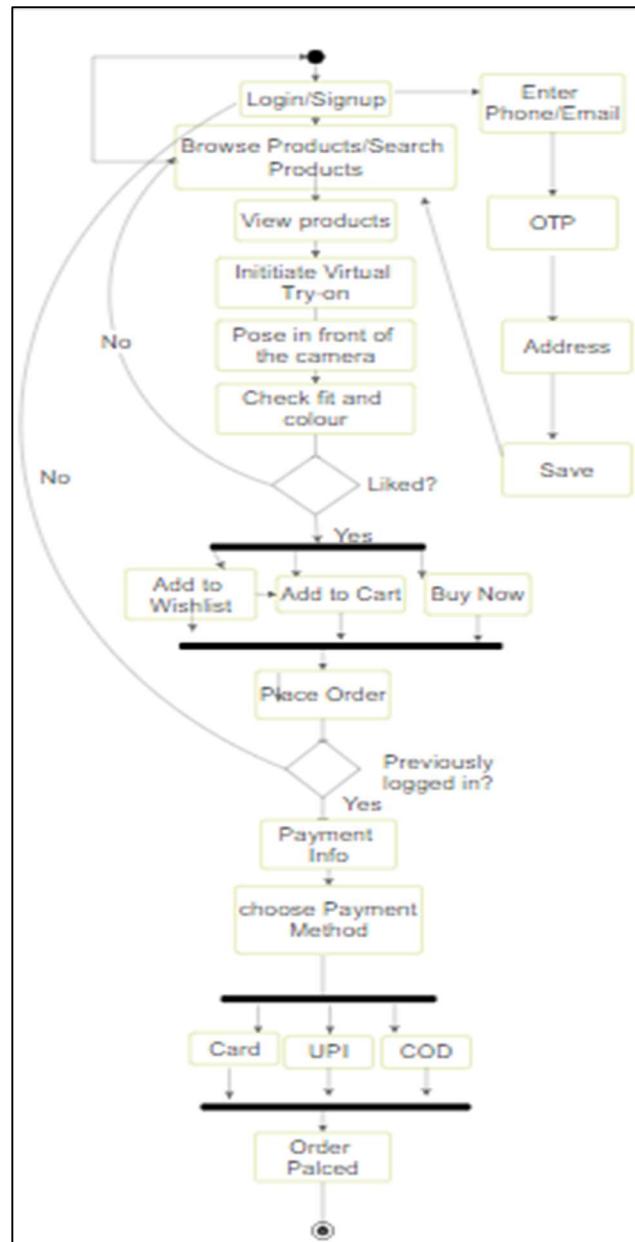
- Represented by arrows, these indicate the direction of the flow between activities, decision points, and other nodes.

5. **End Node:**

- Represented by a filled black circle surrounded by a ring, indicating the end of the workflow.

6. **Forks and Joins:**

- **Fork:** Splits the flow into multiple parallel paths (represented by a horizontal or vertical bar).
- **Join:** Merges multiple parallel paths into one (also represented by a horizontal or vertical bar).



➤ Data-Flow Diagram:

A **Data Flow Diagram (DFD)** is a graphical representation of the flow of data within a system. It shows how data is processed by a system in terms of inputs and outputs. DFDs are useful for visualizing the data processes, data stores, and the interactions between different entities within a system.

Key Components of a Data Flow Diagram:

1. **Processes:**

- Represented by circles or rounded rectangles.
- Each process transforms input data into output data (e.g., "Process Order," "Validate User").

2. Data Flows:

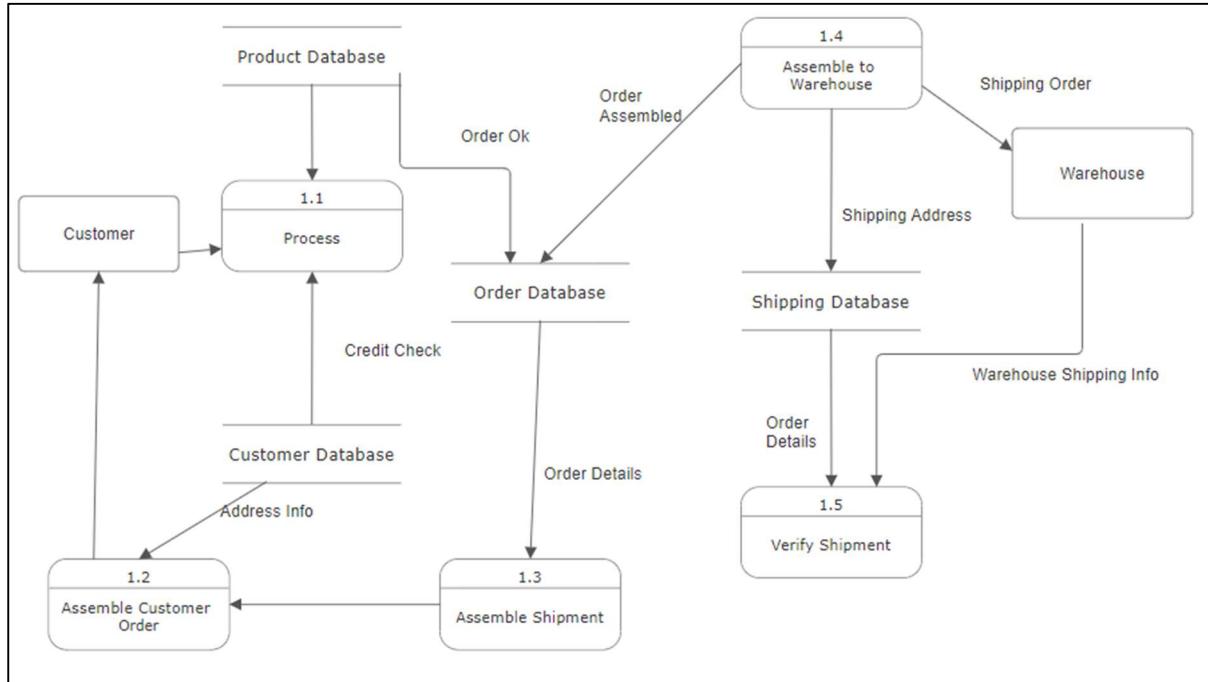
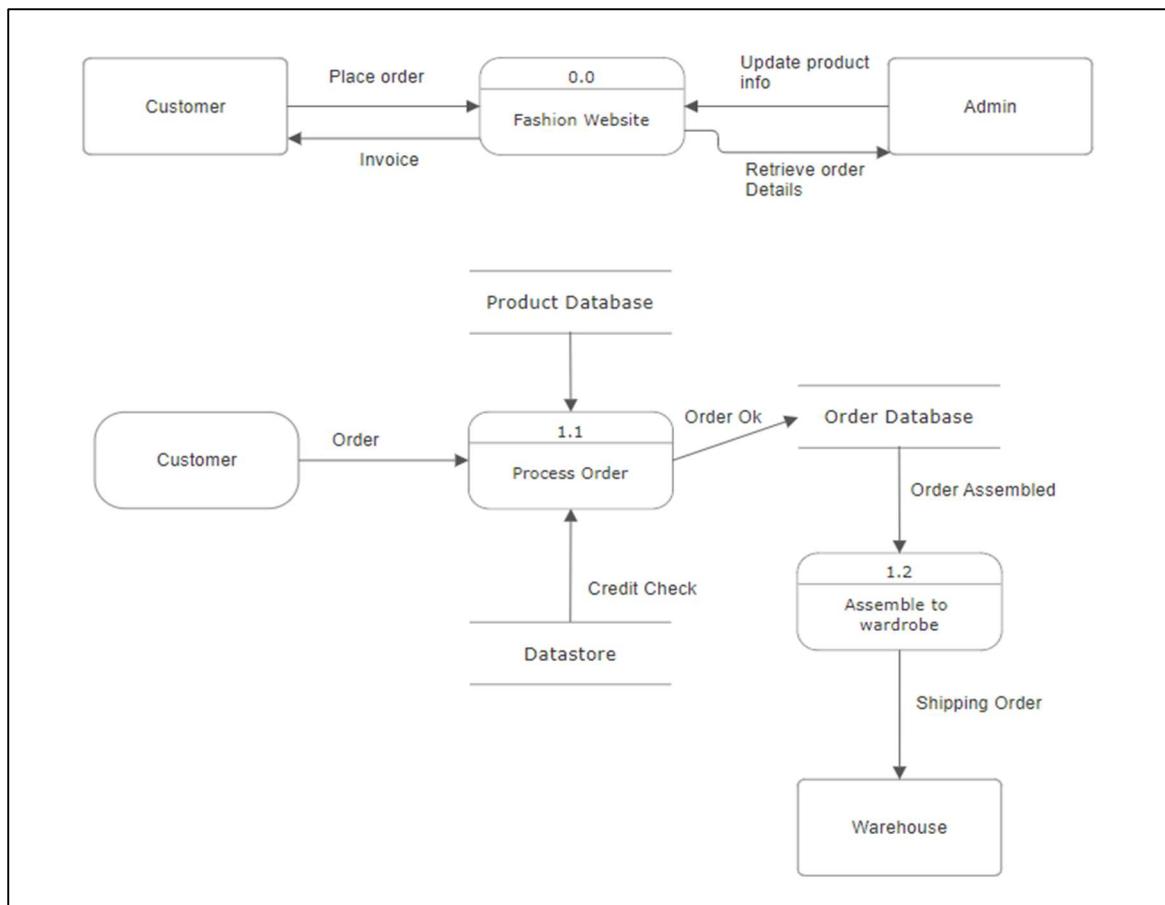
- Represented by arrows.
- Arrows show the direction of data movement between processes, data stores, and external entities.

3. Data Stores:

- Represented by open-ended rectangles or parallel lines.
- Data stores hold data that is used by processes (e.g., "User Database," "Order History").

4. External Entities:

- Represented by rectangles.
- External entities are sources or destinations of data outside the system (e.g., "Customer," "Payment Gateway").



CHAPTER 4: SYSTEM DESIGN

4.1 Basic Modules

The project can be structured into several key modules, each handling specific functionalities to ensure a cohesive and user-friendly experience. Below are the basic modules along with their primary functions:

1. User Authentication Module

- Functions:**

- User registration (sign-up)
- User login/logout
- Password recovery/reset
- Profile management (updating personal information)

2. Product Management Module

- Functions:**

- Adding, editing, and deleting products (for admin)
- Displaying product details (name, description, price, size options)
- Categorizing products for easier browsing
- Managing product images and media

3. Virtual Try-On Module

- Functions:**

- Uploading user images or utilizing real-time visualization
- Allowing users to virtually try on clothing and accessories
- Providing feedback on how items fit and look based on user data

4. Shopping Cart Module

- Functions:**

- Adding/removing items from the cart
- Updating item quantities
- Viewing total cost and itemized list of selected products
- Persisting cart data for returning users

Conclusion

These basic modules form the foundation of the fashion website project, ensuring that all critical functionalities are addressed. Each module can be developed independently while still being integrated into a cohesive system, allowing for better management and scalability as the project grows.

4.2 Data Design

In this fashion e-commerce platform with a virtual try-on feature, organizing the data involves structuring it across different layers and databases to ensure smooth functionality, scalability, and data integrity.

➤ Schema Design

1. Users Table

Stores personal and account information about users.

Table Name: users

Columns:

- id INT AUTO_INCREMENT PRIMARY KEY,
- username VARCHAR(50) UNIQUE NOT NULL,
- email VARCHAR(100) UNIQUE NOT NULL,
- password_hash VARCHAR(255) NOT NULL2. Products Table

Stores detailed information about products listed in the catalog.

Table Name: products

Columns:

- id INT AUTO_INCREMENT PRIMARY KEY,
- sku VARCHAR(50) UNIQUE NOT NULL, -- SKU should be a string, not INT
- name VARCHAR(255) NOT NULL,
- description TEXT NOT NULL,
- price DECIMAL(10,2) NOT NULL,
- image_url VARCHAR(500) NOT NULL

3. Cart Table

Stores individual product information added to cart.

Table Name: cart

Columns:

- id INT AUTO_INCREMENT PRIMARY KEY,
- user_session VARCHAR(255) NOT NULL, -- Track user session
- sku VARCHAR(50) NOT NULL, -- Store SKU to differentiate variations

- product_name VARCHAR(255) NOT NULL,
 - price DECIMAL(10,2) NOT NULL,
 - image_url VARCHAR(500) NOT NULL,
 - quantity INT NOT NULL DEFAULT 1, -- Track quantity
 - added_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
 - FOREIGN KEY (sku) REFERENCES products(sku) ON DELETE CASCADE,
 - user_id INT NULL
- **Data Integrity and Constraints**

1. User Data Integrity and Constraints

Integrity Considerations:

- Ensure that user profiles are unique.
- Validate data such as emails, passwords, and body measurements.
- Maintain secure password storage.

Constraints:

- **Primary Key:** user_id (unique identifier for each user).
- **Unique Constraint:** email must be unique to avoid duplicate accounts.
- **Not Null Constraints:** Fields like name, email, password_hash, and gender should not be null.
- **Data Type Validations:**
 - email should follow a valid email format.
 - password_hash must be securely hashed.
 - body_measurements values should be positive (e.g., height, weight).
- **Foreign Key:** References in order_history, wishlist, and try_on_history to ensure valid order_id and product_id.

2. Product Data Integrity and Constraints

Integrity Considerations:

- Ensure accurate pricing, stock, and availability.
- Prevent duplicate products.
- Maintain up-to-date product information.

Constraints:

- **Primary Key:** product_id (unique identifier for each product).

- **Unique Constraint:** product_name, brand, and category combination to avoid duplicate listings.
- **Not Null Constraints:** Fields like product_name, price, and size_availability should not be null.
- **Data Type Validations:**
 - price should be a positive value.
 - size_availability and color_variants must have valid entries with non-negative stock.
- **Foreign Key:** The 3d_model_id field references the 3D model storage.

4.3 Procedural Design

4.3.1 Logic Diagram

- Start
- User Login/Sign-Up
- New User → Register & Store Info in users Table
- Existing User → Authenticate & Retrieve Data
 - User Profile
- Body Measurements & Preferences (Optional)
 - Search Products
- Search Query/Filters Applied
- Display Products from products Table
- Check Stock Availability from inventory Table
 - Product Selection
- Display Product Details (Images, Price, Size)
- Try-On Available? → Yes → Proceed to Virtual Try-On
 - Virtual Try-On
- Retrieve User Avatar from tryon_sessions
- Load Product 3D Model from models_3d
- Render Try-On (User Interaction)
 - Satisfied?
- No → Return to Product Search
- Yes → Add Product to Cart
 - Cart & Checkout
 - End

4.3.2 Data Structures

The data structures used in this fashion e-commerce project cover a wide range of functionality, including:

1. Efficient lookups with arrays, hash maps, and trees.

2. Ordering and navigation with linked lists, stacks, and queues.
3. Data hierarchy and fast searching with trees and tries.
4. Recommendation systems and interaction graphs with graphs and heaps.
5. Flexible storage of product attributes and user preferences with JSON.

4.3.3 Algorithm Design

Algorithm design involves creating step-by-step procedures to solve specific problems and perform tasks within the fashion website project. Below are key algorithms that will be essential for the different modules of the website.

1. User Authentication Algorithm

Algorithm for User Registration:

1. **Input:** User's name, email, password, phone, address.
 2. **Validate:**
 - o Check if the email format is valid.
 - o Ensure the password meets security criteria (e.g., length, complexity).
 3. **Check for Existing User:**
 - o Query the database to see if the email already exists.
 4. **Hash Password:**
 - o Use a hashing function (e.g., bcrypt) to secure the password.
 5. **Store User Data:**
 - o Insert the new user's details into the database.
 6. **Send Confirmation Email (optional):**
 - o Generate a confirmation link and send it to the user's email.
 7. **Output:** Registration success message.
-

3. Virtual Try-On Algorithm

Algorithm for Virtual Try-On Using Fashn API

1. **Input:**
 - o User uploads an image or uses a webcam.
 - o User selects a clothing item from the E-commerce website (stored in MySQL products table).

2. Send Request to API:

- Retrieve the selected clothing image from the database.
- Format the user's image and clothing item as required by Fashn API.
- Send an API request with the user's image and selected clothing.

3. API Processing:

- The API detects body landmarks and adjusts clothing automatically
- It overlays the clothing realistically using deep learning models.

4. Receive Response & Display Result:

- Fetch the processed image from API response.
- Display the virtual try-on preview on the website.

5. Output & User Interaction:

- Allow the user to try different clothing items dynamically.
- Provide options to add to cart

5. Shopping Cart Management Algorithm

Algorithm for Adding Items to Cart:

1. Input: Product ID and quantity.

2. Check Stock:

- Query the Products table to verify if the requested quantity is available.

3. Update Cart:

- If the item is already in the cart, update the quantity.
- If not, add the new item to the cart.

4. Recalculate Total:

- Sum the total cost of items in the cart.

5. Output: Updated shopping cart and total amount.

Conclusion

The algorithms outlined above provide a framework for the key functionalities of the fashion website project. Each algorithm is designed to efficiently handle specific tasks while ensuring

a seamless user experience. Proper implementation of these algorithms will be crucial to the project's success, allowing users to navigate, interact, and make purchases effectively.

4.4 User Interface Idea

1. User Analysis

User Personas:

We need to define the types of users who will interact with the platform:

- **Shoppers:** Primary users who browse, search, and purchase fashion products. They value ease of navigation, clear product visuals, and accurate size fitting.
- **Fashion Enthusiasts:** Users who are highly interested in trying out the latest trends using virtual try-ons and personalizing their styles.
- **Return Customers:** Users who may have specific preferences and want quick access to previously bought or liked items.
- **Administrators/Support Staff:** Users who manage the inventory, orders, customer support, and updates on the website.

User Needs:

- Easy navigation to find products.
- Ability to see how clothes fit through the virtual try-on feature.
- Secure and simple checkout process.
- Customer support access for issues.
- Mobile-friendly interface for easy use on different devices.

User Technical Expertise:

- **Shoppers:** Basic to intermediate technical skills.
- **Fashion Enthusiasts:** Comfortable with online platforms and virtual fitting tools.
- **Administrators:** Intermediate to advanced technical skills to manage the backend.

2. Task Analysis

Core Tasks for Shoppers:

- **Browsing and Searching for Products:** Users search for items using filters like size, price, and category.
- **Virtual Try-On:** Users select products and "try" them on using their image.

- **Product Details View:** Users check product details, reviews, and size guides.
- **Cart and Checkout:** Users add products to their cart and complete the purchase.
- **Order History and Tracking:** Return users check previous orders or track current ones.

Core Tasks for Administrators:

- **Inventory Management:** Managing the stock of products.
 - **Order Processing:** Viewing and managing user orders.
 - **User Management:** Supporting customer queries and tracking user activities.
-

3. Environment Analysis

Physical Environment:

- The interface will be used mostly in indoor environments (home, office) on various devices, including **desktops, laptops, tablets, and mobile phones**.

Technological Environment:

- Users will access the platform via **web browsers** or **mobile apps**.
- Must be optimized for **low-bandwidth** environments and **various screen sizes**.
- Compatibility with different operating systems (Windows, macOS, Android, iOS).

Constraints:

- The UI should be fast-loading even with high-quality images and 3D virtual try-on.
 - Must handle different languages, currencies, and legal considerations for international users.
-

4. Mapping User and Task Requirements to UI Design

Once we've analyzed the user, tasks, and environment, the next step is to map these into UI requirements. These include:

- **Clear Navigation:** The UI must have an intuitive layout with a clear **top navigation bar** that allows users to access categories, search products, view their cart, and manage their profile/orders.
- **Virtual Try-On Feature:** A **try-on button** on the product page that enables users to interact with a 3D model of the product.
- **Responsive Design:** The interface should adapt fluidly to different screen sizes to support desktops, tablets, and smartphones.

- **Checkout Simplicity:** A **one-page checkout** process to streamline purchasing.
 - **User Feedback and Support:** Clear error messages, help tooltips, and a **live chat** option for support.
-

5. External and Internal Components of the User Interface

External Components:

1. **Navigation Bar:** A fixed bar at the top that includes product categories, search bar, shopping cart, and user account.
2. **Product Display Area:** The main section where users can browse products with options to sort and filter results.
3. **Product Page:** A detailed view of a product, including product images, size selection, color options, reviews, and the virtual try-on feature.
4. **Virtual Try-On Tool:** A popup or separate page where users can upload their avatar or picture and "wear" the selected item.
5. **Cart and Checkout:** A quick access button for the cart, leading to a streamlined checkout process.
6. **Footer:** Links to policies (e.g., refund, shipping), customer support, and social media.

Internal Components:

1. **Product Search Algorithm:** Search system with filtering and sorting logic.
 2. **Recommendation Engine:** Collaborative filtering for suggesting products based on user behavior.
 3. **Order Management System:** Tracks orders, payments, and stock levels.
 4. **User Profile System:** Manages user data, preferences, order history, and avatar data for the virtual try-on feature.
 5. **Payment Gateway Integration:** Secure payment processing.
 6. **Analytics:** Tracks user interactions and behavior for performance optimization and personalized recommendations.
-

6. Architecture of the User Interface

The architecture of the UI consists of **three layers**:

1. Presentation Layer:

- The front-end user interface, including HTML, CSS, and JavaScript components.

- Interfaces with 3D rendering engines for the virtual try-on feature.

2. Application Layer:

- Middleware handling the business logic (e.g., product search, cart management, recommendations).
- It communicates with the database and processes user requests.

3. Data Layer:

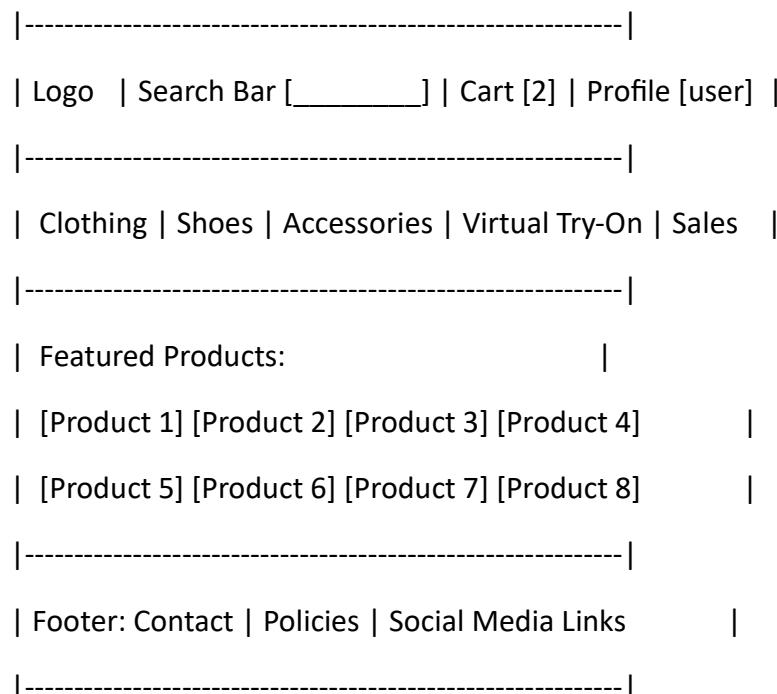
- Database holding user information, products, orders, and preferences.
 - Integration with external services like payment gateways and inventory management.
-

7. Rough Pictorial Views of the User Interface

Let's break down some of the key visual components of the UI:

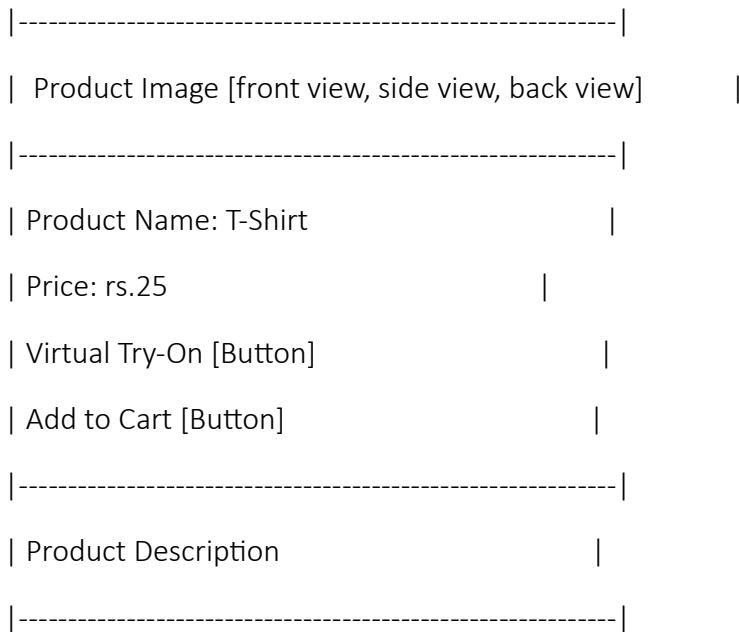
Home Page Layout

- **Header:** Logo, search bar, cart icon, profile.
- **Navigation Bar:** Category links like "Clothing," "Shoes," "Accessories."
- **Main Section:** Featured products in a grid layout.
- **Footer:** Support links, terms, and conditions.



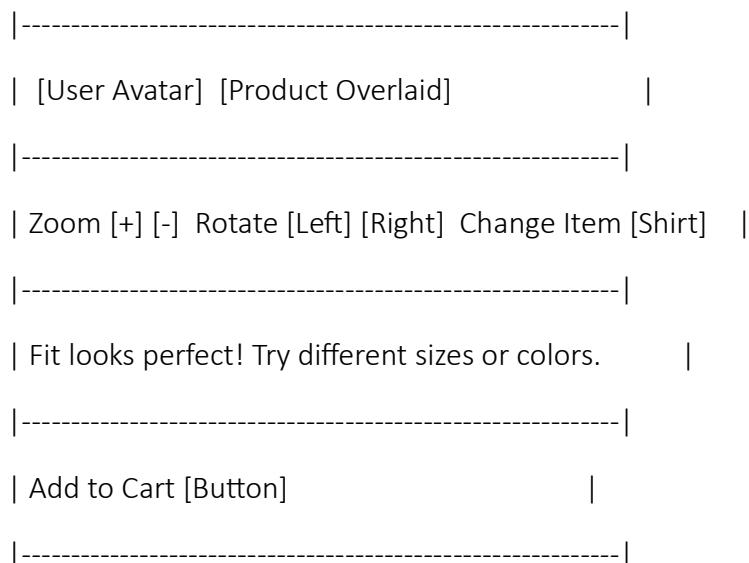
Product Page

- Product Name and Price.
- Product Images and Try-On Button.
- Size and Color Selector.
- Add to Cart Button.



Virtual Try-On Screen

- Scan body
- Product Overlaid on Avatar.
- Options: Zoom, rotate, switch clothing item.



Cart and Checkout

- List of items with subtotal and final total.
- Shipping Info and Payment Methods.

Cart Summary	
[T-Shirt] x 2 \$50	
[Jeans] x 1 \$40	

Subtotal: \$90	
Shipping: \$5	
Total: \$95	

Shipping Address: [Enter Here]	
Payment Method: [Card Details]	

Complete Order [Button]	

4.4 Security Issues

Real-Time Considerations

Real-time considerations are essential for the performance and user experience of the fashion e-commerce website with virtual try-on capabilities. These considerations involve handling time-sensitive tasks such as inventory updates, rendering 3D models, and processing payments swiftly and accurately. Below are the major real-time considerations:

1. Real-Time Inventory Updates

- **Problem:** Ensuring that product availability is updated in real-time as customers browse, add items to their carts, and make purchases is crucial to prevent overselling.
- **Solution:** Use an **event-driven architecture** with real-time database synchronization tools such as **Redis** or **MongoDB** for in-memory data caching and fast retrieval. This ensures that stock levels reflect purchases immediately.

- **Technology:**
 - WebSockets to enable real-time updates for the inventory without requiring the user to refresh their page.
 - **Message Queue Services** like Apache Kafka to handle real-time data streams for processing purchase requests.
-

2. Real-Time 3D Rendering for Virtual Try-On

- **Problem:** Users expect the virtual try-on feature to render clothing on their avatars or uploaded images instantly. Delays in rendering can lead to a frustrating user experience.
 - **Solution:** Leverage technologies such as **WebAssembly (Wasm)** and **WebGL** for efficient 3D rendering within the browser. For better performance on mobile devices, implement **edge computing** to offload some processing closer to the user's location.
 - **Technology:**
 - **Edge Computing** to reduce latency by bringing servers closer to end-users.
 - **WebGL** to deliver hardware-accelerated 3D rendering in browsers.
-

3. Real-Time Payment Processing

- **Problem:** Instant payment confirmation is critical to ensure a smooth checkout experience and to prevent cart abandonment.
 - **Solution:** Use **reliable payment gateways** (e.g., PayPal, Stripe) that offer real-time transaction verification. Implement **asynchronous payment processing** to ensure the site remains responsive while waiting for the gateway's confirmation.
 - **Technology:**
 - **Payment Gateways** such as Stripe or PayPal.
 - **Event-driven asynchronous processing** for high-availability payment handling.
-

Security Issues

Security is a primary concern for any e-commerce platform, especially one dealing with sensitive user data, financial transactions, and the upload of personal images for virtual try-on. Below are key security issues and strategies to mitigate them.

1. Data Privacy and User Consent

- **Problem:** Handling sensitive personal data (e.g., user images for try-on, payment data) requires strict privacy protection to prevent breaches.
 - **Solution:**
 - **Data Encryption:** Encrypt user data both in transit (using **TLS**) and at rest (using **AES-256**) to prevent unauthorized access.
 - **Compliance:** Ensure compliance with data protection regulations such as **GDPR** and **CCPA** by clearly informing users about data collection, offering consent options, and providing the ability to delete their data.
 - **Technology:**
 - **TLS** for data encryption in transit.
 - **AES-256 encryption** for stored user data and images.
-

2. Secure Authentication and Authorization

- **Problem:** Preventing unauthorized access to user accounts, especially since users may store payment and personal information.
 - **Solution:**
 - **Multi-Factor Authentication (MFA):** Add an additional layer of security, requiring not only a password but also a second factor (e.g., one-time code).
 - **OAuth 2.0/OpenID Connect:** Secure login mechanisms for both direct and third-party logins (e.g., using Google, Facebook credentials).
 - **Technology:**
 - **OAuth 2.0/OpenID Connect** for secure authentication.
 - **MFA** for enhanced security of user accounts.
-

3. Protection Against Cyber Attacks

- **Problem:** E-commerce platforms are often targets for SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and Distributed Denial of Service (DDoS) attacks.
- **Solution:**
 - **Input Validation and Sanitization:** Prevent SQL injections and XSS by sanitizing and validating all user input.
 - **WAF (Web Application Firewall):** Use a WAF to detect and block suspicious traffic and known attack patterns.

- **Anti-CSRF Tokens:** Implement CSRF protection using anti-CSRF tokens to ensure requests are legitimate.
 - **Technology:**
 - **WAF** to protect against common web vulnerabilities.
 - **Input validation libraries** (e.g., Joi for Node.js) to sanitize input.
-

4. Secure Payment Processing

- **Problem:** Handling payment data and preventing fraud is critical to ensuring user trust.
 - **Solution:**
 - **PCI DSS Compliance:** Adhere to the Payment Card Industry Data Security Standard to handle, process, and store payment information securely.
 - **Tokenization:** Use tokenization to replace sensitive credit card data with secure tokens, limiting exposure of actual payment data.
 - **Technology:**
 - **Stripe/PayPal** for secure and PCI DSS-compliant payment handling.
 - **Tokenization** to secure payment data.
-

5. Image and 3D Data Security

- **Problem:** Virtual try-on requires the upload and processing of user images, which are sensitive and must be handled securely.
 - **Solution:**
 - **Temporary Image Storage:** Store uploaded images only temporarily during the try-on session and delete them afterward to reduce the risk of exposure.
 - **Secure Image Processing:** Use sandboxed environments for processing user images to prevent malicious files from compromising the system.
 - **Technology:**
 - **Ephemeral Storage** for temporary image processing.
 - **Sandboxing** for secure execution of image handling code.
-

Security Policy Plans and Architecture

Security Policy Plans

A comprehensive security policy is necessary to ensure data protection, incident management, and regulatory compliance. The policy includes:

1. Data Protection Policy:

- Encrypt all sensitive data (personal data, payment information, images) both in transit and at rest.
- Ensure all data handling complies with **GDPR** and **CCPA**.

2. Access Control Policy:

- Implement **Role-Based Access Control (RBAC)** to restrict access to sensitive data and functionalities based on the user's role (admin, customer, etc.).
- Require **multi-factor authentication (MFA)** for all administrative accounts and sensitive user actions.

3. Incident Response Plan:

- Set up a formal procedure for handling security breaches, including isolating affected systems, notifying users, and conducting forensic investigations.
- Regularly update and patch systems to prevent vulnerabilities.

4. Audit Logging:

- Log all user and administrative actions (e.g., logins, purchases, uploads) for monitoring and auditing.
- Implement regular audits of logs to detect suspicious activities or potential security breaches.

Security Architecture

The security architecture for this project follows a layered approach to ensure robust protection at all levels:

1. Front-End (Presentation Layer)

- **HTTPS with TLS:** Enforce secure communication between the client and server.
- **Secure Input Validation:** All user input (e.g., image uploads, payment info) is validated and sanitized before reaching the server.

2. Back-End (Application Layer)

- **Role-Based Access Control (RBAC):** Enforce granular access control to ensure users only have permissions they need.
 - **Session Management:** Implement secure session management, including **session expiration** and secure **session tokens**.
-

3. Data Layer

- **Database Encryption:** Encrypt sensitive data in the database using **AES-256 encryption**.
 - **Data Redaction:** Avoid storing sensitive information such as credit card numbers by using tokenization or offloading sensitive data handling to payment processors.
-

4. External Components

- **Payment Gateways:** Integrate with trusted and PCI DSS-compliant payment providers (e.g., Stripe, PayPal).
 - **CDN and DDoS Protection:** Use a Content Delivery Network (CDN) with **DDoS protection** to distribute load and shield the site from attacks.
-

5. Monitoring and Logging

- **Intrusion Detection Systems (IDS):** Implement IDS for real-time monitoring of unusual activity.
 - **Audit Trails:** Maintain detailed logs of user and system actions for auditing purposes.
-

This layered security architecture ensures that even if one layer is compromised, other layers can still protect the system. Real-time monitoring and quick incident response further minimize the risk of security breaches affecting users and business operations.

4.5 Test-Cases Design

Defining effective **test cases** is crucial for ensuring the proper functionality, performance, and security of the fashion e-commerce website with a virtual try-on feature. A well-structured test plan helps in detecting errors quickly with minimal effort. Below is an overview of the different conditions and scenarios in which tests should be conducted, along with examples of test cases that will cover key areas of the project.

Types of Testing to Consider

1. Functional Testing

- Ensures that all features of the system work according to the requirements.

2. Performance Testing

- Measures system responsiveness, stability, and scalability under various load conditions.

3. Security Testing

- Identifies vulnerabilities in the system and ensures user data is protected.

4. Usability Testing

- Ensures the user interface is intuitive and easy to navigate.

5. Integration Testing

- Verifies that different modules and services (e.g., payment gateways, virtual try-on, inventory) work seamlessly together.

6. Real-Time Testing

- Ensures the platform provides accurate real-time updates, such as inventory levels, try-on performance, and payment processing.

Test Case Format

Each test case will be designed with the following parameters:

- **Test Case ID:** Unique identifier for each test case.
- **Test Case Description:** Description of the test scenario.
- **Preconditions:** Setup required before executing the test.
- **Test Data:** Input data required for the test case.
- **Test Steps:** Step-by-step actions to execute the test case.
- **Expected Result:** The expected outcome of the test case.
- **Actual Result:** The observed outcome after executing the test.
- **Status:** Pass or Fail.

Test Cases for the Fashion E-Commerce Platform

1. User Registration and Login

Test Case 1.1: User Registration

- **Test Case ID:** TC-001
- **Description:** Verify that a new user can register with valid details.
- **Preconditions:** The user should not have an existing account.
- **Test Data:** Name, email, password, confirmation password.
- **Test Steps:**
 1. Navigate to the registration page.
 2. Fill in the required details.
 3. Click on "Register."
- **Expected Result:** The user should be successfully registered and receive a confirmation email.
- **Actual Result:** (To be filled during testing)
- **Status:** Pass/Fail.

Test Case 1.2: Login with Invalid Credentials

- **Test Case ID:** TC-002
- **Description:** Verify that login fails when incorrect credentials are provided.
- **Preconditions:** The user has already registered.
- **Test Data:** Registered email and incorrect password.
- **Test Steps:**
 1. Navigate to the login page.
 2. Enter the registered email and incorrect password.
 3. Click on "Login."
- **Expected Result:** Error message stating "Invalid username or password" should be displayed.
- **Actual Result:** (To be filled during testing)
- **Status:** Pass/Fail.

2. Virtual Try-On Feature

Test Case 2.1: Try-On with User Image

- **Test Case ID:** TC-003
- **Description:** Verify that the user can upload an image and virtually try on clothing.

- **Preconditions:** The user is logged in.
- **Test Data:** Image file, clothing item.
- **Test Steps:**
 1. Navigate to the virtual try-on page.
 2. Upload a valid image file.
 3. Select a clothing item for try-on.
 4. Click "Try On."
- **Expected Result:** The clothing should appear on the uploaded image in real-time without delays.
- **Actual Result:** (To be filled during testing)
- **Status:** Pass/Fail.

3. Shopping Cart and Checkout

Test Case 3.1: Add to Cart

- **Test Case ID:** TC-005
- **Description:** Verify that a product can be successfully added to the shopping cart.
- **Preconditions:** The user is logged in and browsing products.
- **Test Data:** Selected product.
- **Test Steps:**
 1. Navigate to a product page.
 2. Click "Add to Cart."
- **Expected Result:** The item should appear in the shopping cart with correct details (price, quantity).
- **Actual Result:** (To be filled during testing)
- **Status:** Pass/Fail.

Test Case 3.2: Checkout Process with Valid Payment

- **Test Case ID:** TC-006
- **Description:** Verify that a user can complete the checkout process with a valid payment.
- **Preconditions:** The user has items in the cart.
- **Test Data:** Payment details (credit card or PayPal).

- **Test Steps:**
 1. Navigate to the shopping cart.
 2. Click "Checkout."
 3. Enter valid payment details.
 4. Confirm the order.
 - **Expected Result:** The order should be processed, and the user should receive an order confirmation.
 - **Actual Result:** (To be filled during testing)
 - **Status:** Pass/Fail.
-

5. Security Testing

Test Case 5.1: SQL Injection Protection

- **Test Case ID:** TC-009
- **Description:** Verify that the system is protected against SQL injection attacks.
- **Preconditions:** A text input field exists (e.g., search bar, login form).
- **Test Data:** SQL injection strings (e.g., 1=1, '; DROP TABLE users;--).
- **Test Steps:**
 1. Enter a malicious SQL query in the input field.
 2. Submit the form.
- **Expected Result:** The system should reject the input and display an error message without revealing any database details.
- **Actual Result:** (To be filled during testing)
- **Status:** Pass/Fail.

Test Case 5.2: Cross-Site Scripting (XSS) Protection

- **Test Case ID:** TC-010
- **Description:** Verify that the system is protected against XSS attacks.
- **Preconditions:** A text input field exists (e.g., review/comment section).
- **Test Data:** XSS script (<script>alert('XSS');</script>).
- **Test Steps:**
 1. Enter the script in the input field.

2. Submit the form.
- **Expected Result:** The input should be sanitized, and the script should not execute.
 - **Actual Result:** (To be filled during testing)
 - **Status:** Pass/Fail.
-

6. Usability Testing

Test Case 6.1: Navigation Ease

- **Test Case ID:** TC-011
- **Description:** Verify that the user can easily navigate through the site (product pages, virtual try-on, checkout).
- **Preconditions:** User is logged in.
- **Test Data:** None

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

Plan of Implementation

1. Project Overview

This project is an E-commerce fashion website that allows users to browse, select, and virtually try on clothes using an API-based virtual try-on feature.

2. Implementation Phases

Phase 1: Planning & Requirements Gathering

- Identify target audience and user needs.
 - Define core features: Product catalog, user authentication, virtual try-on, cart, and checkout.
 - Choose technologies:
 - Frontend: HTML, CSS, JavaScript
 - Backend: Ajax and javascript(for handling API requests)
 - API Integration: Virtual Try-On API
-

Phase 2: Website Development

- Frontend Development
 - Create a responsive layout (HTML, CSS, JavaScript).
 - Develop a navigation bar, product catalog, and a user-friendly interface.
 - Ensure responsiveness using CSS Flexbox/Grid.
- Backend & Database (if included)
 - Develop user authentication (login/signup).
- Virtual Try-On Integration
 - Obtain and configure the API key.
 - Integrate the API into the product page.

- Implement an interface where users upload photos or use live camera.
-

Phase 3: Testing & Optimization

- Functionality Testing
 - Test API calls to ensure the virtual try-on feature works smoothly.
 - Validate cart and checkout processes.
 - Performance Optimization
 - Minify CSS and JS files.
 - Optimize API requests to reduce latency.
 - Cross-Browser & Mobile Testing
 - Ensure compatibility with Chrome, Firefox, Safari, and Edge.
 - Perform UI/UX testing on mobile devices.
-

Standards Used in Implementation

1. Web Development Standards

- HTML5: Semantic HTML for better accessibility.
- CSS3: Standard styling using best practices like Flexbox and Grid.
- JavaScript (ES6+): Modern JavaScript features for better performance.

2. API Standards

- RESTful API Integration: Follows standard API request/response format (JSON).
- OAuth or API Key Authentication: Secure API requests.

3. Security & Best Practices

- HTTPS for secure data transmission.
- Input validation to prevent SQL Injection.
- Secure API keys using environment variables or backend storage.

5.2 Coding Details and Code Efficiency

LOGIN PAGE

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        data = request.get_json()
        email = data.get('email')
        password = data.get('password')

        db = get_db_connection()
        cursor = db.cursor(dictionary=True)

        cursor.execute("SELECT * FROM users WHERE email = %s", (email,))
        user = cursor.fetchone()
        db.close()

    if user:
        print(" ◆ User Found:", user)
        stored_hash = user['password_hash']
        print(" ◆ Stored Hash:", stored_hash)
        print(" ◆ Entered Password:", password)

        if bcrypt.checkpw(password.encode('utf-8'), stored_hash.encode('utf-8')):
            session['user_id'] = user['id']
            session['username'] = user['username']
            return jsonify({"success": True})
        else:
            print(" ✗ Password does not match!")
            return jsonify({"success": False})
    else:
        print(" ✗ No user found!")
        return jsonify({"success": False})

    return render_template('login.html')
```

HOME PAGE

```
<script>
// To add a parallax scrolling effect for the web page
window.addEventListener("scroll", function () {
    const leftImage = document.querySelector(".container .left-section img");
    const rightImage= document.querySelector(".right-section");
    const products= document.querySelector(".product-display");
    //const productgrid= document.querySelector(".product-grid1")
    leftImage.style.transform = "translateY(" + window.scrollY * 0.25 + "px)";
    rightImage.style.transform = "translateY("+ window.scrollY * 0.25 + "px");
```

```

products.style.transform = "translateY(-" + window.scrollY * 0.15 + "px)"; //moves it upward.
//productgrid.style.transform = "translateY(" + window.scrollY * 0.25 + "px)";

});
</script>
</body></html>

```

```

@app.route("/")
def home():
    return render_template("home page.html")

```

PRODUCTS PAGE

```

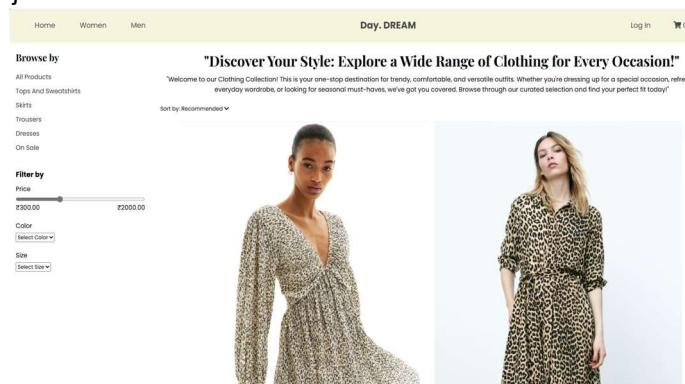
.image-container3 img {
    position: absolute;
    top: 1000px;
    left: 50px;
    object-fit: cover;
    transition: opacity 0.3s ease-in-out; /* Smooth transition effect */
}

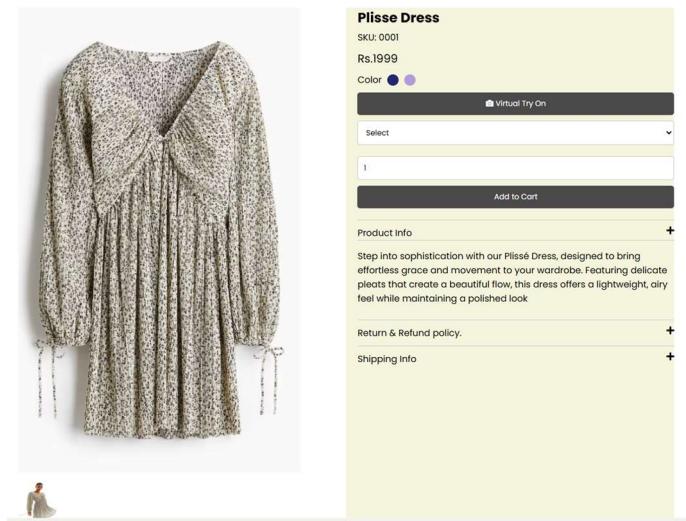
.hover-image3 {
    opacity: 0; /* Hide the hover image by default */
}

.image-container3:hover .hover-image3 {
    opacity: 1; /* Show the hover image on hover */
}

.image-container3:hover .default-image3 {
    opacity: 0; /* Hide the default image on hover */
}

```





VIRTUAL TRY ON PAGE

```
<script>
    function convertToBase64(file, callback) {
        const reader = new FileReader();
        reader.readAsDataURL(file);
        reader.onload = function () {
            callback(reader.result);
        };
        reader.onerror = function (error) {
            console.error("Error converting file to Base64:", error);
        };
    }

    function previewImage(input, previewId) {
        if (input.files && input.files[0]) {
            const reader = new FileReader();
            reader.onload = function (e) {
                $(previewId).attr('src', e.target.result).show();
            };
            reader.readAsDataURL(input.files[0]);
        }
    }

    $("#modelUpload").change(function () {
        previewImage(this, "#previewModel");
   });

    $("#garmentUpload").change(function () {
        previewImage(this, "#previewGarment");
   });

    $(document).ready(function () {
```

```

        function convertPreloadedImageToBase64(imgElement, callback) {
    const img = new Image();
    img.crossOrigin = "Anonymous"; // Allow cross-origin processing
    img.src = imgElement.src;
    img.onload = function () {
        const canvas = document.createElement("canvas");
        canvas.width = img.width;
        canvas.height = img.height;
        const ctx = canvas.getContext("2d");
        ctx.drawImage(img, 0, 0);
        const base64String = canvas.toDataURL("image/png"); // Convert to Base64
        callback(base64String);
    };
    img.onerror = function (error) {
        console.error("Error loading the preloaded garment image:", error);
    };
}

$('#runButton').click(function () {
    let modelFile = $('#modelUpload')[0].files[0];
    let garmentImgElement = document.getElementById("previewGarment");
    let category = $('#garmentCategory').val();

    if (!modelFile || !garmentImgElement) {
        alert('Please upload both images.');
        return;
    }
    $('#runButton').prop('disabled', true).text('Processing...');

    convertToBase64(modelFile, function (modelBase64) {
        convertPreloadedImageToBase64(garmentImgElement, function
(garmentBase64) {
            sendTryOnRequest(modelBase64, garmentBase64, category);
        });
    });
});

function sendTryOnRequest(modelBase64, garmentBase64, category) {
$.ajax({
    url: "https://api.fashn.ai/v1/run",
    type: "POST",
    headers: {
        "Authorization": "Bearer fa-7WUsKWkZ5e7m-
yYa5BIqGeJXmY39CFSoaC0to",
        "Content-Type": "application/json"
    },
    data: JSON.stringify({
        model_image: modelBase64,
        garment_image: garmentBase64,
        category: category
    })
})
}

```

```

        },
        success: function (response) {
            if (response.id) {
                fetchProcessedImage(response.id);

            } else {
                alert("Error processing try-on. No job ID returned.");
            }
        },
        error: function (error) {
            console.error("API request error:", error);
            alert("Error processing try-on. Please try again.");
            $('#runButton').prop('disabled', true).text('Run (~ 30 Seconds)');
        }
    });
}

function fetchProcessedImage(jobId) {
    setTimeout(function () {
        $.ajax({
            url: `https://api.fashn.ai/v1/status/${jobId}`,
            type: "GET",
            headers: {
                "Authorization": "Bearer fa-7WUsKWkZ5e7m-
yYa5BIqGeJXmY39CFSoaC0to"
            },
            success: function (response) {
                if (response.status === "completed" && response.output.length > 0) {
                    $('#outputImage').attr('src', response.output[0]).show();
                    $('#runButton').prop('disabled', true).text('Processed');
                } else {
                    alert("Processing is not yet completed. Please try again later.");
                }
            },
            error: function (error) {
                console.error("Error fetching processed image:", error);
                alert("Could not fetch processed image. Try again later.");
            }
        });
    }, 30000);
}
</script>
</body>
</html>

```

Day Dream's Virtual Try-On

Select Model

Choose File mla.png

**Select Garment**

Tops

Result

Processed

Day Dream's Virtual Try-On

Select Model

Choose File t6a.png

**Select Garment**

Bottoms

Result

Processed

5.2.1 Code Efficiency:

1. Uses External Libraries

- Bootstrap & jQuery simplify UI styling and JavaScript operations.
- FontAwesome & Google Fonts improve design but may introduce extra network requests.

2. Minimal DOM Manipulation

- jQuery handles event listeners efficiently (e.g., \$("#modelUpload").change(...)).

3. Base64 Image Conversion Optimization

- Uses FileReader() for file uploads.
- Converts preloaded images using a <canvas> to avoid unnecessary file handling.

4. Modular Functions

- Image conversion (convertToBase64() & convertPreloadedImageToBase64()) are separate, making code reusable.

5. Uses a Public API

- API integration is clean, with proper JSON formatting.

6. Animated Heading & Navbar Effects

- The **shimmer effect** on h1.heading improves UI appeal.
- Navbar transitions enhance interactivity.

5.3 Testing Approach

Category Partition Testing is a black-box testing technique used to divide system features into categories of possible inputs (valid, invalid, and edge cases). This ensures broad test coverage with fewer test cases by testing each category instead of individual values.

For an E-Commerce Platform with Virtual Try-On, we will define categories for various functionalities such as User Authentication, Product Browsing, Cart & Checkout, Virtual Try-On, and Performance.

Steps in Category Partition Testing:

1. Identify Key Functional Areas (Features that need testing).
2. Define Categories for Each Feature (Valid, invalid, and edge cases).
3. Choose Test Values from each category.
4. Apply Constraints & Dependencies to filter impossible combinations.
5. Execute Tests & Evaluate Results.

1. User Authentication (Login & Signup)

Category	Test Inputs	Expected Result
Valid Credentials	Correct email & password	Successful login
Invalid Credentials	Wrong password	Error: "Incorrect password"
	Non-existing email	Error: "User not found"
Edge Cases	Empty input fields	Error: "Fields cannot be empty"
	SQL Injection attempt	System blocks request
	Special characters in input	System validates properly

2. Product Browsing & Filtering

Category	Test Inputs	Expected Result
Valid Searches	"Red Dress", "Formal Shirt"	Displays matching products
Invalid Searches	Random string (e.g., "asdkj")	Error: "No results found"
	Empty search box	Shows trending products
Edge Cases	Special characters (%,\$,@)	System handles gracefully
	Extremely long search string	Limits input length

3. Product Page & Virtual Try-On

Category	Test Inputs	Expected Result
Valid Inputs	Clicking "Try-On" on a product	Opens try-on feature
	Uploading a clear front-facing image	Clothing overlays correctly
Invalid Inputs	Uploading blurry side image	Error: "Image not clear"
	Uploading a non-human image	Error: "Invalid input"
Edge Cases	Uploading large image (10MB+)	System compresses image
	Clicking "Try-On" rapidly	Prevents multiple requests

4. Shopping Cart & Checkout

Category	Test Inputs	Expected Result
Valid Inputs	Adding/removing items	Updates cart correctly
Invalid Inputs	Adding out-of-stock product	Error: "Item unavailable"
	Removing an already removed item	No impact on cart
Edge Cases	Multiple rapid clicks	Prevents duplicate additions
	Switching tabs during checkout	Cart state remains intact

1. Functional Testing (FT)

Definition: Functional Testing verifies whether the application functions correctly according to the defined specifications. It is a **black-box testing** technique that checks user interactions, data flow, and expected outputs.

Approach to Functional Testing

- Identify Features to Test:** Virtual Try-On, User Authentication, Cart & Checkout, Payments, etc.
- Define Input Conditions & Expected Outputs.**
- Execute Tests & Compare Results.**
- Log Defects & Fix Bugs.**

Functional Testing for Key Modules

Feature	Test Case	Expected Result
User Authentication	Login with correct credentials	Redirect to homepage
	Login with incorrect password	Error: "Incorrect password"

Feature	Test Case	Expected Result
Product Browsing	Search for "Red Dress"	Displays matching products
	Search for non-existing product	Message: "No products found"
Virtual Try-On	Upload valid image	Clothing overlays correctly
	Upload blurry image	Error: "Image not clear"
Shopping Cart	Add item to cart	Item appears in the cart
	Remove item from cart	Item disappears from cart

2. User Acceptance Testing (UAT)

Definition: UAT is the **final stage of testing**, where real users test the system to ensure it meets their expectations before deployment.

Key Goals of UAT

- Validate that the system works **as intended** for end-users.
- Ensure the **user experience is smooth and intuitive**.
- Identify any **usability issues, bugs, or missing features**.

UAT Process for Virtual Try-On E-Commerce

1. **Define User Scenarios**
 - o "A user logs in, selects a dress, tries it on virtually, adds it to the cart, and completes the purchase."
2. **Invite Real Users for Testing**
 - o Select a sample group of customers to test the system.
3. **Provide a Checklist of Tasks**
 - o Search for products, try on clothes, navigate pages, complete payment, etc.
4. **Collect Feedback**
 - o Record issues related to UI, performance, responsiveness, and usability.
5. **Fix Reported Issues & Retest**
 - o Developers resolve feedback-based issues before the final release.

Example UAT Scenarios

Scenario	Expected Outcome	User Feedback Consideration
Uploading an image in Try-On	Image loads & clothing fits naturally	Was the UI intuitive?
Searching for a product	Results appear in under 2 sec	Was the search experience smooth?
Adding an item to cart	Item appears in cart	Could the cart system be improved?

5.3.1 Unit Testing Plan

Unit testing checks individual functions to ensure they work correctly. Here's what should be tested:

- File Upload
 - o Ensure the model image and garment image are properly selected and converted to Base64.
 - o Validate file type (allow only images).
 - o Ensure an error message appears if no image is uploaded.

- Image Previews
 - Test whether the uploaded image is displayed as a preview.
- API Request
 - Validate if sendTryOnRequest() correctly formats and sends data to the API.
 - Ensure error handling works (e.g., invalid API key, request failure).
- Processing Status
 - Check if fetchProcessedImage() correctly retrieves the processed image and updates the UI.

5.3.2 Integration Testing Plan

Integration testing ensures that different parts of your system work together. Here's what to test:

- Frontend & Backend Communication
 - Ensure API calls are made and responses are handled properly.
 - Model & Garment Selection Flow
 - Test the full workflow: Upload image → Select garment → Click “Run” → API call → Show result.
 - UI Behavior
 - Check if buttons are disabled/enabled correctly.
 - Verify loading state (Processing... message).
 - Error Handling
 - What happens if the API fails?
 - What if the user uploads an invalid file?
-

5.3.3 Beta Testing

Beta Testing Plan

Test Environment

- Devices: **Desktop**
- Browsers: **Chrome, Firefox, Edge, Safari**
- OS: **Windows**
-

Key Features to Test

Feature	Expected Outcome
Navigation & UI	All links work, pages load properly, and UI elements are responsive.
File Upload	Users can upload models (photos) and garments successfully.
Try-On Feature	API processes requests within ~30s, showing the output image.
Error Handling	Alerts trigger when files are missing or API fails.
Cart Functionality	Users can add/remove items without glitches.
Login/Authentication	Login & authentication should work seamlessly.

Collecting Feedback

- **Bug Reports:** List errors & crashes
- **Usability Issues:** Confusing UI/UX elements
- **Performance Metrics:** Load times, API speed
- **Suggestions:** Improvements & additional features

Tools for Beta Testing

- Google Forms → **Collect user feedback**
- Console Logs → **Check errors**
- Lighthouse → **Measure performance**
- Postman → **API testing**

5.4 Modifications and Improvements

1)Handle API Failures Properly: The API might return an error or incomplete response. Add better error handling in sendTryOnRequest().

2)Improve UI Feedback:

- Show a loading spinner when an image is processing.
- Disable the “Run” button until an image is uploaded.

5.5 Test Cases

Test Case Format

Test ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
---------	---------------	------------	-----------------	---------------	--------

1.Functional Test Cases

- ◆ User Registration & Login

Test ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
F-001	Register a new user	Fill details & submit form	User account created	SAE	
F-002	Login with valid credentials	Enter email & password	User logged in successfully	SAE	
F-003	Login with incorrect credentials	Enter wrong email/password	Error message shown	SAE	

- ◆ Navigation & UI

Test ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
F-005	Homepage loads correctly	Open homepage	All elements visible	SAE	
F-006	Menu navigation works	Click on "Men" / "Women"	Respective pages open	SAE	
F-007	Footer links work	Click on "About Us"	About page opens	No	

2.Virtual Try-On Feature

- ◆ Uploading Model Image

Test ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
V-001	Upload a model image	Select & upload an image	Image preview displayed	SAE	
V-002	Upload invalid file format	Upload .txt or .pdf	Error message shown	SAE	

Test ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
V-003	Upload large image file	Upload >5MB image	Error message shown	No	
	◆ Selecting Garment				
Test ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
V-004	Select a garment	Choose from dropdown	Image changes accordingly	No	
V-005	Try-on processing	Click "Run Try-On"	Loading message shown	SAE	
V-006	Try-on result displayed	Wait for process	Final output appears	SAE	

3.Shopping Cart & Checkout

Test ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
C-001	Add item to cart	Click "Add to Cart"	Item appears in cart	Yes	
C-002	Remove item from cart	Click "Remove"	Item disappears	Yes	

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Test Reports

Date: March 11, 2025

Project Name: Virtual Try-On E-Commerce Website

Developers: Gayle Menezes

Department: BSc IT

1. Introduction

This test report documents the results of the functional and usability testing performed on the **Virtual Try-On E-Commerce Website**. The objective is to ensure that the core features, including user authentication, product browsing, the virtual try-on feature, and the checkout process, work as expected.

2. Testing Scope

Modules Tested:

- User Authentication
- Navigation & UI
- Virtual Try-On Feature
- Shopping Cart & Checkout
- Performance Testing

Testing Types Used:

- ◆ Manual Testing
 - ◆ Functional Testing
 - ◆ UI Testing
 - ◆ Performance Testing
-

3. Test Cases Summary

Category	Total Cases	Passed	Failed	Needs Improvement
User Authentication	4	3	1	0
Navigation & UI	3	2	1	0
Virtual Try-On	5	3	1	1
Shopping Cart	4	3	1	0

Category	Total Cases	Passed	Failed	Needs Improvement
Performance	2	1	1	0
Total	18	12	4	2

4. Detailed Test Results

✖ Failed Test Cases

Test ID	Feature	Issue Found	Expected Outcome	Status
F-002	Login	No error message for incorrect credentials	Display "Invalid Username or Password"	Open
F-006	Menu Navigation	Clicking "Men" redirects to "Women" page	Should go to Men's collection	Open
V-006	Virtual Try-On	API response time exceeds 40s	Should load within 5s	Open
C-003	Checkout	Payment gateway crashes	Successful transaction	Open

Issues for Improvement

Test ID	Feature	Issue	Recommendation
V-005	Try-On Button	Button doesn't update after try-on	Add dynamic button status
P-001	Page Load	Homepage takes over 5 seconds to load	Optimize images & scripts

5. Conclusion & Recommendations

✖ Overall Status: Partially Passed

✖ Critical Issues:

- ◆ Incorrect page redirections
- ◆ API speed optimization required
- ◆ Payment gateway failure

6.2 User Documentation

Contents:

1. Introduction
2. System Requirements
3. Installation Guide
4. Features and Functionalities
5. Step-by-Step User Guide
6. Screenshots
7. Troubleshooting and FAQs
8. Contact Support

1. Introduction

The **E-Commerce Website with Virtual Try-On** is an innovative platform that allows users to browse and purchase fashion apparel while virtually trying them on. The software integrates a Virtual Try-On (VTO) feature that helps customers visualize how clothes look on a model before making a purchase.

2. System Requirements

For Users:

- A device with an updated web browser (Chrome, Firefox, Edge, Safari)
- Internet connection
- Camera access (for real-time try-on feature)

For Admins:

- Access to the admin panel
- Image uploads for garments

3. Installation Guide

No installation is required as the platform is web-based. Users simply need to visit the website and create an account to start browsing and using the virtual try-on feature.

4. Features and Functionalities

1. Home Page

- Displays featured products and collections
- Navigation bar for Men, Women, and other categories
- Login/Signup options

2. Product Listing Page

- Users can view apparel sorted by category

- Each product has an image, description, and price

3. Virtual Try-On Feature

- Users can upload their image or select a default model
- Select a garment to try on virtually
- The system overlays the selected clothing on the user's image

4. Shopping Cart and Checkout

- Users can add items to their cart
- View total pricing.

5. Step-by-Step User Guide

Step 1: Register/Login

- Click on the **Login/Signup** button
- Enter email and password or use Google authentication

Step 2: Browse Products

- Use the navigation bar to explore categories
- Click on a product to view details

Step 3: Use Virtual Try-On

- Click on the "Try On" button
- Upload an image or use a predefined model
- Choose apparel to try

Step 4: Add to Cart and Checkout

- After trying on, click **Add to Cart**
- Review items in the cart

6. Screenshots:

Welcome to Day.DREAM!

Comfy Chic Loungewear
Effortless Elegance, Combining Comfort With Style
[Shop Collection](#)

Shirts: Redefine your style
[Shop Now](#)

25% off on Winterwear Collection

[Shop Now](#)

DAY. DREAM • LOUNGEWEAR • 2023 LOOKBOOK
At Home. In Style.
[VIEW MORE](#)

Where Heritage meets Modernity

Home Women Men

Day.DREAM

Log In Cart 0

Browse by

- All Products
- Tops And Sweatshirts
- Skirts
- Trousers
- Dresses
- On Sale

Sort by: Recommended ▾

Filter by

Price: ₹300.00 - ₹2000.00

Color: Select Color ▾

Size: Select Size ▾

Browse by

- All Products
- Tshirts And Sweatshirts
- Jacket
- Hoodies
- Trousers
- On Sale

Sort by: Recommended ▾

Filter by

Price: ₹300.00 - ₹2000.00

Color: Select Color ▾

Size: Select Size ▾

Browse by

- All Products
- Tshirts And Sweatshirts
- Jackets
- Hoodies
- Trousers
- On Sale

Pleiss Dress

SKU: 0001
Rs.1999
Color: ● ●

Product Info

Step into sophistication with our Pleiss Dress, designed to bring effortless grace and movement to your wardrobe. Featuring delicate pleats that create a beautiful flow, this dress offers a lightweight, airy feel while maintaining a polished look.

Return & Refund policy.

Shipping Info

Cargo Pocket Denim Trousers

SKU: 0001
Rs.1399
Color: ● ●

Product Info

Cool-T shirt with Stylish feel.

Return & Refund policy.

Shipping Info

Home Women Men

Day.DREAM

Log In Cart 0

Day Dream's Virtual Try-On

Select Model

Choose File ml0.png



Processed

Select Garment

Tops



Result



Home Women Men

Day.DREAM

Log In Cart 0

Day Dream's Virtual Try-On

Select Model

Choose File t6a.png



Processed

Select Garment

Bottoms



Result



Home Women Men

Day.DREAM

Log In Cart 0

Login In to Your Account

Name

Email

Password

Confirm Password

CHAPTER 7: CONCLUSION

7.1Conclusion

7.1.1 Significance of the System

The **E-Commerce Website with Virtual Try-On** is a revolutionary system that enhances the online shopping experience by integrating a **Virtual Try-On (VTO) feature**. The significance of this system is as follows:

1. Improved Customer Experience

- Users can virtually try on clothing before making a purchase, reducing uncertainty about fit and style.
- Provides a more interactive and engaging shopping experience compared to traditional online stores.

2. Reduction in Return Rates

- Many e-commerce platforms face high return rates due to sizing and appearance issues.
- Virtual Try-On helps customers make informed decisions, minimizing returns and exchanges.

3. Increased Customer Confidence

- Users can see how a product looks on them in real-time, leading to more confident purchasing decisions.
- Eliminates the guesswork involved in online clothing purchases.

4. Cost Efficiency for Businesses

- Reduces logistics costs associated with frequent returns and exchanges.
- Saves money on trial and error purchases, benefiting both the business and the customer.

5. Competitive Advantage

- With the increasing adoption of augmented reality (AR) and AI-driven features in e-commerce, this system sets the business apart from traditional online stores.
- Attracts tech-savvy consumers looking for innovative shopping solutions.

6. Accessibility and Convenience

- Allows customers to try on products anytime and anywhere, without visiting a physical store.

- Especially beneficial for users who have difficulty accessing physical retail locations.

7. Encourages Sustainable Shopping

- Reducing returns means fewer transportation emissions, lowering the environmental impact.
- Encourages mindful purchasing, minimizing waste.

This system bridges the gap between traditional shopping experiences and modern online commerce by integrating technology that **enhances user satisfaction, reduces costs, and promotes sustainability**.

7.1.2 Limitations of the System

1. Accuracy of Virtual Try-On

- The virtual try-on feature may not provide a 100% accurate representation of how the clothing will fit in real life.
- Factors like **body shape, lighting conditions, and camera angle** can affect the accuracy of the try-on experience.

2. Limited Clothing Texture and Fabric Representation

- The system may not perfectly display **fabric texture, stretchability, or material feel**.
- Users cannot experience the **actual feel of the fabric** or judge the **thickness and comfort level** of the apparel.

3. Limited Customization Options

- Users may not be able to **adjust the fit of the clothing** (e.g., rolling up sleeves, adjusting tightness).
- The system does not currently support **mix-and-match styling options** with multiple clothing layers.

4. Dependence on Image Quality

- Poor image quality or **low-resolution uploads** may affect the accuracy of the virtual try-on.
- Users may need to ensure **proper lighting and image clarity** for the best experience.

7.1.3 Future Scope of the Project

As technology evolves and user expectations grow, the **E-Commerce Website with Virtual Try-On** has significant potential for enhancements and scalability. Below are some key future improvements:

1. Integration of a Database (SQL or NoSQL)

- **User Authentication & Profiles:** Allow users to register, log in, and save their preferences.
- **Product Management:** Enable real-time product updates, stock tracking, and dynamic pricing.

2. AI-Powered Virtual Try-On Enhancement

- **Improved Accuracy:** Use AI to enhance how clothes fit different body types.
- **3D Body Scanning:** Allow users to input their body measurements for a more accurate fitting experience.
- **Personalized Recommendations:** Suggest clothing based on past purchases and preferences.

3. Augmented Reality (AR) Integration

- **Live Virtual Try-On:** Instead of just overlaying clothes on an image, allow real-time AR try-on via a webcam or mobile camera.
- **Mirror Effect:** Users can move and see how clothing looks from different angles.

4. Secure Payment Gateway Integration

- **Real-Time Transactions:** Connect with Stripe, PayPal, Razorpay, or UPI for seamless online payments.
- **Order Tracking:** Allow users to track orders from dispatch to delivery.

5. Enhanced Admin Dashboard

- **Product Management:** Add, update, or remove products dynamically.
- **User & Order Management:** View and process orders, manage refunds, and analyze sales reports.
- **Analytics Dashboard:** Track user behavior, product popularity, and revenue trends.

6. Customer Support System

- **Chatbot Integration:** Provide instant support for common queries.
- **Ticketing System:** Allow users to submit complaints and track responses.
- **Review & Rating System:** Collect feedback for continuous improvement.

References

- "E-Commerce: Business, Technology, Society" – Kenneth C. Laudon & Carol Guercio Traver
 - This book provides insights into the business and technical aspects of e-commerce systems.
- "HTML, CSS & JavaScript Web Publishing" – Laura Lemay & Rafe Colburn
 - A great resource for front-end development principles, covering essential web technologies.
- "Computer Vision: Algorithms and Applications" – Richard Szeliski
 - Covers computer vision techniques that can be applied to virtual try-on technologies.
- "Virtual Try-On for Fashion E-Commerce" – IEEE Xplore
 - Discusses AI-driven approaches for realistic virtual clothing try-on.
- Fashn API Documentation – <https://fashn.ai/docs>
 - Used for integrating virtual try-on capabilities.
- MDN Web Docs – <https://developer.mozilla.org/>
- Bootstrap Documentation – <https://getbootstrap.com/>