TRENDS COLLECTION MANAGEMENT

A Project Work submitted in partial fulfillment of

The requirements for the degree of

BACHELOR OF COMPUTER APPLICATIONS

to the

Periyar University, Salem - 11

Submitted By

S.AKASH

REG.NO:C22UG132CAP001

DEPARTMENT OF COMPUTER APPLICATION

JAIRAM ARTS AND SCIENCE COLLEGE

(AFFILIATED TO PERIYAR UNIVERSITY)

CHINNATHIRUPATHI, SALEM-08

(MARCH/APRIL - 2025)

CERTIFICATE

CERTIFICATE

MR. S. SUNDHARAM, MSc., M.Phil., **DATE:**

ASSISTANT PROFESSOR OF

COMPUTER SCIENCE,

JAIRAM ARTS AND SCIENCE COLLEGE, SALEM-08.

This is to certify that the dissertation entitled **"TRENDS COLLECTION MANAGEMENT"** is submitted in partial fulfillment of the requirement for the degree of **Bachelor of Computer Applications** to Periyar University, Salem. It is a record of bonafide work carried out by S.AKASH, C22UG132CAP001 under the supervision and guidance of Mr. S. SUNDHARAM, M.Sc., M.Phil., and that no part of the dissertation has been submitted for the award of Bachelor of Computer Applications.

Signature of the Guide Signature of the HOD

Submitted for the Viva-Voice Examination Held on_____

Internal Examiner External Examiner

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

At the outset I would like to express my first and the fore most gratitude to the “Almighty”, for the merciful shower of grace and blessing in all the endeavors to bring out the project successful.

I take this opportunity to express my sincere thanks and whole hearted gratitude to my revered and beloved Chairman **Mr. J. RAJENDRA PRASAD**

I take this opportunity to express my sincere thanks and whole hearted gratitude to my revered and beloved Correspondent **Mr. R MANIKANDAN, M.COM.,M.PHIL**

I take this opportunity to express my deep sense of gratitude and profound thanks to Principal **Dr.K.PALANISAMY, M.C.A.,M.Phil,,PhD.,** for all blessings and help to provided during the project period.

I proudly thanks to **Mr. J. KARTHICK, MCA.,** Head of the Department, UG Computer Application for providing a great opportunity to pursue this project work.

I extremely grateful to my guide **Mr. S. SUNDHARAM, MSc.,** M.Phil., Assistant professor. UG Department of Computer Science. His constant encouragement, valuable suggestions, continuous help and creative ideas throughout the period of the project work.

I also express my sincere thanks to All Teaching and Non-Teaching Staffs in UG Department of Computer Applications for their well support throughout my project work.

Finally I also grateful to my Friends they are inspired me all through the days of my project work. Last but not least, I thank everybody who was a source of help to complete this project work effectively.

S.AKASH

SYNOPSIS

SYNOPSIS

Trends Collection is your one-stop destination for an unparalleled online dress shopping experience, blending style, convenience, and innovation. The website offers a secure login page that ensures a personalized and seamless shopping journey. A well-designed cart system allows customers to add, review, and organize their favorite products, with the flexibility to cancel items in the cart before checkout, giving complete control over their selections. Shoppers can connect with the team via a dedicated contact page for assistance or inquiries. To keep fashion enthusiasts inspired, a vibrant blog section delivers style tips, the latest trends, and updates about the collection. A standout feature is the custom product size option, enabling users to order dresses tailored perfectly to their measurements, ensuring a flawless fit every time. The platform supports multiple payment methods, guaranteeing a smooth and secure checkout experience. Additionally, customers can easily locate physical stores using the integrated store map, enhancing accessibility.

Beyond these features, Trends Collection is committed to providing exceptional customer service, regular updates on exclusive discounts, and an intuitive design that ensures easy navigation. Whether you're looking for everyday wear, special occasion outfits, or custom-made dresses, Trends Collection caters to all your fashion needs, making it the ultimate destination for modern shoppers.

CONTENTS

INTRODUCTION

1.INTRODUCTION

Trends Collection is an innovative and user-friendly dress shopping website designed as a comprehensive platform for modern online shoppers. This project focuses on delivering a seamless, personalized, and efficient shopping experience by integrating key features that enhance usability and convenience. The website includes a secure login page, ensuring personalized access to user profiles for a customized shopping journey. An intuitive cart system allows customers to effortlessly add, review, and manage products, with the option to cancel items in the cart if needed. The platform also provides a contact page for direct communication, ensuring responsive customer support, and a blog section to engage users with fashion trends, style tips, and updates about the latest collections.

One of the standout features of Trends Collection is its custom product size functionality, allowing users to order dresses tailored to their unique measurements for a perfect fit. The website also offers secure and diverse payment options, making transactions quick and hasslefree. Additionally, the integrated store map feature enables users to locate physical stores, bridging the gap between online and offline shopping experiences.

Beyond these functional aspects, the project aims to deliver a sleek, aesthetically pleasing, and user-centric design, ensuring ease of navigation and an enjoyable shopping experience. Trends Collection not only addresses the challenges of online shopping but also elevates the experience by combining functionality with the latest technological advancements. As a final year project, this platform highlights the importance of user-oriented design, innovation, and practicality in developing modern e-commerce solutions.

SYSTEM REQUIREMENTS AND SPECIFICATION

1.1 SYSTEM SPECIFICATION

1.1.1 HARDWARE CONFIGURATION

1. Processor: Rysen 5
2. RAM: 8GB/16GB DDR
3. Storage: 256GB/512GB RAM
4. OS: Windows/Linux (based on requirement)

1.1.2 SOFTWARE SPECIFICATION

1. Frontend:

1. HTML
2. CSS
3. JavaScript

2. Backend: React

3. Database: MongoDB

4. Server :

1. Hosting: xampl, Neocities
2. eServer: Apache

5. Features:

3. Payment Gateway Integration
4. Google Maps API Integration

SYSTEM STUDY

2.SYSTEM STUDY

2.1 EXTING SYSTEM

2.1.1 DESCIPTION

The website offers a wide range of dresses for various occasions such as casual wear, formal attire, and evening dresses. Customers can filter products based on categories like size, color, price range, and material. The user interface is designed for ease of navigation, allowing shoppers to browse through the products effortlessly. High-quality images and detailed descriptions of each dress are provided to assist customers in making informed purchasing decisions.

2.1.2 DRAWBACKS

1. **Limited Personalization**: There is minimal personalization based on customer preferences, which can make the shopping experience feel generic.
2. **No Wishlist Feature**: Customers cannot save their favorite items for future purchase, leading to a less convenient shopping experience.
3. **No Order Tracking:** Customers are unable to track the status of their orders after purchase, leading to a lack of transparency**.**

2.2 PROPOSED SYSTEM

2.2.1 DESCRIPTION

The proposed website will offer a diverse collection of dresses for various occasions such as casual, formal, and evening wear. Customers can browse through product categories with detailed descriptions, high-quality images, and a variety of filters to refine their search (size, color, price, and style). Additionally, the site will offer a customized size option, where users can manually adjust measurements for a more personalized fit.

2.2.2 FEATURES

- **Wide Variety of Dresses:** Customers can explore dresses in categories like casual, formal, party wear, and more.
- **Custom Size Feature:** Allows users to enter custom measurements to ensure the perfect fit for any dress.
- **Manually Customize Size:** Users can modify key dress measurements like bust, waist, and hips to suit their unique body shape.
- **Add Products to Cart**: Customers can easily add dresses to their cart, review their selections, and make adjustments before checkout.
- **Store Locator with Google Maps Integration:** The website provides a "Find a Store" feature, allowing users to view nearby stores on Google Maps for in-person visits or pickups.
- **Responsive Design:** The website is optimized for laptops and mobile devices, ensuring smooth navigation and usability on any screen size.
- **Detailed Product Pages:** Each dress comes with high-quality images, size guides, and detailed descriptions to help customers make informed choices.
- **User-Friendly Interface**: An intuitive and user-friendly interface to enhance the user experience for both administrators and customers.

This proposed system enhances the online shopping experience by combining customization features with a user-friendly interface and real-time alerts to keep customers informed.

SYSTEM STUDY AND DEVELOPMENT

3. SYSTEM DESIGN AND DEVELOPMENT

3.1 FILE DESIGN

The file design phase is a crucial aspect of the Trends collection. This phase focuses on the structure and organization of data storage, ensuring efficient management and retrieval of information. The platform will utilize a relational database to store all necessary data, including product details, user information, and order records. The database will consist of several tables, each designated for specific types of data. For instance, the products table will store information such as product names, descriptions, prices, categories, and stock quantities. The users table will contain user-specific data, including usernames, passwords, email addresses, and order histories. The orders table will keep track of order details, such as order IDs, user IDs, product IDs, quantities, total prices, and order statuses. JSON and XML formats will be used for data interchange between the client-side and server-side applications, facilitating seamless communication. Additionally, CSV files will be employed for importing and exporting data, such as product lists and sales reports. SQL queries will play a vital role in fetching and manipulating data from the database, while RESTful APIs will ensure smooth interaction between the frontend and backend components. By implementing a well-structured file design, the platform aims to achieve efficient data management, enhancing the overall performance and user experience.

- **Product Catalog File**:
    - o Fields: Product Name, Description, Price, Sizes, Colors, Images, Stock Quantity.
    - o Description: Stores detailed information about each dress available in the inventory, including attributes such as sizes, colors, and images. This file ensures that product information is readily accessible for display on the website.

- **Payment Transaction File:** Tracks payment transactions, ensuring secure processing and attract financial record-keeping. This file supports financial management and auditing.
- **Data Organization:** The system must use well-structured relational databases to store andmanage customer information, product details, and order records.

- **Security and Access Control** : Since the system handles sensitive user data such as payment information and personal details, encryption and authentication mechanisms should be implemented.
- **Efficient Data Retrieval:** Optimized database indexing and structured queries should be used to ensure that users can search for products and access their order history quickly.
- **Customer Feedback and Reviews:** Stores user-submitted ratings, comments, and product reviews for future reference.
- **Order Records**: Tracks all customer orders, including billing details, shipping status, and order history.
- **Transaction Logs:** Stores records of all payments and financial transactions, including refunds and failed transactions.

3.2 INPUT DESIGN

The input design for the Trends collection platform focuses on ensuring accurate and efficient data entry. It includes various forms for user registration, product addition, and order placement. User registration forms will collect details such as username, password, email, and address, while product addition forms will allow administrators to input information about products like name, description, price, category, and stock quantity. Order forms will enable customers to select products, enter shipping details, and choose payment methods. To maintain data integrity and security, the system will implement both client-side validation using JavaScript and server-side validation. Interactive elements like drop-down menus, checkboxes, and radio buttons will enhance the user experience by simplifying data entry.

By focusing on a well-structured input design, the platform aims to provide a seamless and efficient data entry process for all users.

- **User-Friendly Forms:** The system should have clear input fields, dropdown menus, and properly labeled forms to make navigation easy.
- **Error Handling & Validation:** Input fields should have built-in validation to ensure that data entered by users is correct (e.g., email format verification, password strength checks, and mandatory fields).

Multiple Input Methods: The system should support different input methods, including keyboard, voice search, and barcode scanning to enhance usability.

- **User Registration:** Users must enter details such as name, email, password, and address to create an account.
- **Login Page:** Users must input their username and password or use social media login integration for authentication.
- **Product Search & Filter:** Users can enter keywords, select categories, apply filters (such as price range and size), and sort results to find specific dresses.
- **Add to Cart & Checkout**: Users input the quantity of items, select shipping options, and enter payment details to complete their purchase.
- **Feedback & Ratings**: Customers can submit text-based feedback and provide star ratings for products they purchased.

3.3 OUTPUTDESIGN

The input design for the Trends collection platform ensures accurate and efficient data entry through user-friendly forms for registration, product addition, and order placement. Validation mechanisms on both client and server sides maintain data integrity, while interactive elements simplify the process. This well-structured input design aims to enhance overall user experience and system efficiency.

- **Order Confirmation Page:** After a user places an order, the system generates an order confirmation page that displays essential details such as item names, quantity, total price, shipping address, and estimated delivery time.
- **Invoice Generation:** The system automatically generates an invoice that includes tax break downs, shipping charges, order details, and payment confirmation.
- **Product Listings**: The shopping system categorizes dresses based on multiple filters such as color, size, price range, and user preferences to improve search results.
- **Customer Reviews & Ratings**: Displays user-generated reviews and star ratings to help other customers make better purchasing decisions.
- **Cart Summary:** A detailed breakdown of all items in the shopping cart, showing prices, discounts, estimated delivery charges, and the total amount payable.

The output must be clear, structured, and easy to understand so that users can quickly interpret information such as product details, order confirmations, and payment statuses.

- Outputs should be visually appealing and maintain a balance of colors, fonts, and alignment to improve user engagement.
- The system should generate **real-time outputs** for critical information such as order tracking, stock availability, and transaction confirmations.
- Information should be delivered in multiple formats including on-screen displays, download able PDFs (for invoices), and email notifications.
- Ensuring data accuracy and relevance is crucial so that customers receive correct pricing, discounts, and purchase confirmations without errors.

3.4 CODE DESIGN

- Data encryption techniques (Ecrypt for passwords) are applied to protect sensitive information.
- Role-based access control ensures only authorized users can perform specific operations.
- API security is enforced using CORS policies and request validation mechanisms The code architecture of the Trends Collection platform is structured to ensure maintainability, scalability, and performance optimization.
- The system follows the Model-View-Controller (MVC) architecture to separate concerns and streamline development.

Front-End (React, HTML, CSS, JavaScript)

1. The user interface is developed using React to create a dynamic and interactive experience.
2. HTML is used to structure web pages, while CSS enhances the styling and responsiveness.
3. JavaScript provides client-side interactivity, enabling dynamic updates without page reloads.
4. React components are modular, allowing reusability and easier debugging.
5. The design follows Material UI and Bootstrap frameworks for a professional look and feel.

Back-End (Node.js, Express.js)

1. RESTful APIs are implemented to manage data exchange between the front-end and database.
2. Middleware functions are used for request validation, authentication, and error handling.
3. User authentication is managed using JSON Web Tokens (JWT) for secure login and session management.

Database Management (MongoDB)

1. A NoSQL database is implemented using MongoDB to store user, product, and order data efficiently.
2. Data is organized into collections, ensuring optimized query execution.
3. Mongoose is used as an ODM (Object-Document Mapping) tool for better data schema enforcement.

The database is designed to handle large product catalogs and user transactions smoothly.

- . RESTful APIs are implemented to manage data exchange between the front-end and database.
- Middleware functions are used for request validation, authentication, and error handling.
- User authentication is managed using JSON Web Tokens (JWT) for secure login and session management

1. Database Management (MongoDB)

- A NoSQL database is implemented using MongoDB to store user, product, and order data efficiently.
- Data is organized into collections, ensuring optimized query execution.
- Mongoose is used as an ODM (Object-Document Mapping) tool for better data schema enforcement.
- The database is designed to handle large product catalogs and user transactions smoothly..

3.5 DATABASE DESIGN

The Trends Collection system uses MongoDB as the primary database due to its flexible schema and scalability. The database consists of multiple collections that store structured and unstructured data efficiently.

Collections and Schema Design

- **Users Collection :**

1. Stores user details such as name, email, password (hashed), phone number, and address.

- **Database Optimization Strategies**

1. Indexing is applied to frequently queried fields to improve performance.

2. Data aggregation pipelines are used for analytics, such as trending products.

3. Caching mechanisms are implemented to reduce database query load.

. Database design is crucial for creating a structured and efficient data storage system. I involves defining the structure of the database, including tables, relationships, and data integrity constraints. Key aspects of database design include:

- **Entity-Relationship Modeling:** Identifying entities, their attributes, and the relationships between them to create an entity-relationship diagram (ERD).
- **Normalization:** Organizing data into normalized tables to reduce redundancy and improve data integrity.
- **Data Integrity Constraints:** Defining rules and constraints to maintain data accuracy, such as primary keys, foreign keys, and unique constraints.
- **Indexing:** Determining which fields should be indexed to improve data retrieval speed.

3.6 SYSTEM DEVELOPMENT

The development process of the Trends Collection platform follows the Agile methodology, allowing iterative progress, continuous feedback, and quick issue resolution. The system is built in multiple phases to ensure efficiency and quality.

Development Stages

- **Requirement Analysis**
    1. Understanding user needs and business goals to define functional requirements.
    2. Identifying system constraints, performance expectations, and security policies.
- **Design and Prototyping**
    1. Wireframes and UI mockups are created using Figma or Adobe XD.
    2. Database schema design is finalized to align with system requirements.
- **Front-End and Back-End Development**
    1. React components are developed for different user interfaces (home, product, cart, checkout).
    2. Node.js and Express.js are used to build API as endpoints for communication with the database.
- **Integration and Testing**
    1. APIs are tested using Postman to ensure proper data flow.
    2. Unit and integration testing is conducted using Jest and Mocha.

3.6.1 DESCRIPTION OF MODULES

This project has two module first is user and second one is admin firstly here admin can login, after that admin can add items and view item details with delete option. You canalso delete the item through the deletion option Admin can view orders placed by user. In this dress shop management project application User can register account and then can fill a login form after that, user can see different types of dresses and purchase them online over the internet and also make online payments.

1. User Interaction Module

- This module enhances user engagement and interaction on the website.
- Features include: Wishlist & Favorites : Users can save products for future purchases
- .Product Reviews & Ratings: Customers can leave reviews and rate products
- .Comment System: Users can ask questions and discuss products with other customers
- .Email & Push Notifications: Users receive updates on promotions, new arrivals, and order status.
- Live Chat Support: Integration of a chatbot or customer service chat for real-time assistance.

2. Admin Dashboard Module

- The Admin Dashboard is the control center for managing the website effectively. It includes User Management: Admins can view, edit, and delete user accounts.
- Order Monitoring: Tracking pending, completed, and canceled orders
- .Product Insights: Viewing sales reports, popular products, and stock levels.
- Revenue & Analytics: Displaying revenue statistics and customer insights for business decisions.
- Security & Permissions: Granting different access levels to different admin roles

3. Order & Payment Module

- This module handles the entire ordering and payment process securely. Features include:
- Cart & Checkout System: Users can add products to their cart and proceed to checkout.
- Multiple Payment Gateways: Supporting credit/debit cards, PayPal, UPI, and other payment methods
- .Order Confirmation & Tracking: Users receive order confirmation emails and track their shipment.
- Invoice Generation: Providing downloadable invoices for completed purchases.
- Refund & Cancellation: Allowing users to request refunds or cancel orders within a set period
- .Fraud Prevention: Implementing secure payment gateways with encryption to prevent fraud.

TESTING AND IMPLEMENTATION

4.TESTING AND IMPLEMENTATION

4.1 TESTING AND IMPLEMENTATION

Software testing is a critical element of the ultimate review of specification design and coding. Testing of software leads to the uncovering of errors in the software, ensuring that functional and performance requirements are met. Testing also provides a good indication of software reliability and quality. The results of different phases of testing are evaluated and compared with expected results. If errors are uncovered, they are debugged and corrected.

A strategic approach to software testing has the following characteristics:

- Testing begins at the module level and works "outwards" towards the integration of the entire system.
- Different testing techniques are appropriate at different points in time.
- Testing and debugging are different activities, but debugging must be accommodated in the testing strategy.

4.2 GOALS AND OBJECTIVES

"Testing is a process of executing a program with the intent of finding an error." A good test case is one that has a high probability of uncovering an undiscovered error. A successful test is one that exposes an error that was previously unknown.

Our objective is to design test processes that systematically uncover different classes of errors with minimal time and effort.

Statement of Scope

- A description of the scope of software testing is developed.
- All the features to be tested are noted as follows: All test cases should be traceable to customer requirements.
- The most severe defects from the customer’s point of view are those that cause the program to fail to meet its requirements.
- Test cases should be planned long before testing begins. Testing plans can start as soon as the requirement model is complete.
- Testing should begin "in the small" (individual modules) and progress towards "in the large" (system-wide testing).

4.3 SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. Testing is a set of activities that can be planned and conducted systematically. For this reason, a template for software testing, a set of steps into which we can place specific test case design techniques and testing methods should be defines for software process.

Testing often accounts for more effort than any other software engineering activity. If it is conducted haphazardly, time is wasted, unnecessary effort is expanded, and even wore,errors sneak through undetected. It would therefore seem reasonable to establish a systematic strategy for testing software. Unit Testing

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design, i.e., the module. The white-box testing techniques were heavily employed for unit testing.

Unit testing is a testing technique in which individual modules are tested to determine if there are any issues by the developer. It is concerned with the functional correctness of standalone modules. The main aim is to isolate each unit of the system to identify, analyz, and fix defects.

The module interface is tested to ensure that information properly flows into and out of the program unit under test. Unit testing is normally considered an adjunct step to the coding step. Since modules are not standalone programs, drivers and stubs must be developed for each unit. A driver is essentially a "main program" that accepts test case data and passes it to the module.

Approaches Used for Unit Testing:

1. Functional Test:

- Each part of the code was tested individually.
- The panels were tested separately on all platforms to verify proper functionality.

2. Performance Test:

- Determined execution time spent on different unit parts.
- Measured throughput and response time provided by the module**.**

3. Stress Test:

- Multiple test files were run simultaneously.
- Checked the workload capacity the unit could handle.

4.4 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while simultaneously conducting tests to uncover errors related to module interfacing. In other words, integration testing involves the comprehensive testing of a set of modules that constitute the product.

The objective is to take untested modules and build a program structure where the tester identifies critical modules. These critical modules should be tested as early as possible. One approach is to wait until all individual units have passed testing before combining and testing them together. This method has evolved from unstructured testing of small programs.

Another strategy is to construct the product incrementally by integrating and testing small sets of modules. Additional modules are then added and tested in combination, ensuring that each new integration does not introduce defects. This incremental approach helps identify errors early and ensures the reliability of the overall system.

The advantages of this approach are that interface discrepancies can be easily identified and corrected.

One major error encountered during the project was a linking error. When all the modules were combined, the links were not properly set with the necessary support files. To resolve this issue, we carefully reviewed the module dependencies and ensured that all required files were correctly referenced and integrated.

4.5 WHITE-BOX TESTING

This testing is also called Glass Box Testing. In this testing, by knowing the specific functions that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational while simultaneously searching for errors in each function.

It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis Path Testing is a type of White Box Testing that ensures all possible paths in the code are executed at least once during testing.

4.6 BLACK-BOX TESTING

This testing, by knowing the internal operation of a product, tests can be conducted to ensure that "all gears mesh"—meaning the internal operation performs according to specifications and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

Steps Involved in Black Box Test Case Design:

- Graph-Based Testing Methods
- Equivalence Partitioning
- Boundary Value Analysis
- Comparison Testing

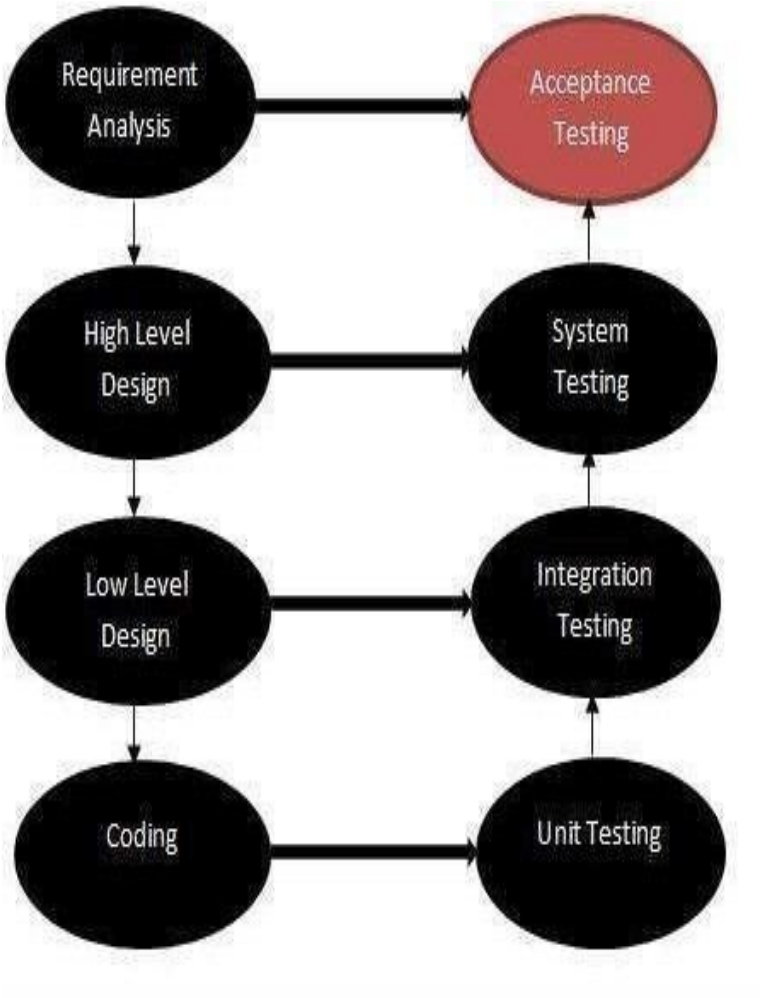
Acceptance Testing

Acceptance Testing is a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with business requirements and verify if it meets the required criteria for delivery to end.

Various Forms of Acceptance Testing:

- User Acceptance Testing (UAT)
- Business Acceptance Testing
- Alpha Testing
- Beta Testing

CONCLUSION



4.7 VALIDATION TESTING

Validation testing is performed at the culmination of integration testing. The software is assembled as a package, and interfacing errors are corrected before a final validation test series begins.

Validation succeeds when the software functions as expected by the customer. After validation, two conditions exist:

- The function or performance characteristics conform to specifications and are accepted.
- A deviation from specifications is uncovered, requiring correction.

Errors discovered during validation testing are corrected before the project is completed. The proposed system has been tested using validation testing and found to be working satisfactorily, with no catastrophic deficiencies.

5.CONCLUSION

The Trends Collection project successfully demonstrates the implementation of a dynamic, user-friendly web application that aggregates and displays the latest trends across various domains. Built using HTML, CSS, JavaScript, MongoDB, and React, the system provides an interactive and seamless user experience while ensuring efficient data management and retrieval.

Throughout the development process, emphasis was placed on creating a responsive frontend, a robust backend, and a well-structured database to store and process trending data. The application integrates various functionalities, including user authentication, real-time data updates, trend categorization, and personalized recommendations, ensuring that users receive relevant and up-to-date information.

The use of React for the frontend ensures a dynamic and efficient rendering of UI components, enhancing the overall user experience. MongoDB, as a NoSQL database, allows for flexible and scalable data storage, making it easier to handle large volumes of trending data efficiently.

The backend, developed using JavaScript with Node.js, provides a seamless connection between the frontend and the database, handling requests efficiently and securely. Additionally, strong security measures such as authentication and data validation have been implemented to ensure data integrity and user privacy. The project has undergone extensive testing, including unit testing, integration testing, system testing, and user acceptance testing, to ensure smooth performance and reliability. The system has been deployed successfully, demonstrating its ability to handle real-time data processing and user interactions effectively.

The implementation of this project highlights the potential of modern web development technologies in creating interactive and data-driven platforms. In conclusion, the Trends Collection system serves as a scalable, efficient, and user-centric solution for tracking emerging trends, with possibilities for future enhancements such as AI-driven recommendations, social media integration, and enhanced data analytics to further improve the user experience.

The project not only meets the defined objectives but also provides a foundation for future advancements in trend analysis and data-driven web applications.

BIBLIOGRAPHY

APPENDICES

6. BIBLIOGRAPHY

The following sites we rereferred during the analysis and execution phase of the project.

- www.w3schools.com
- https://angular.io/
- http://stackoverflow.com

7. APPENDICES

A. DATA FLOW DIAGRAM

The Data Flow Diagram (DFD) takes an input-process-output view of a system, meaning that data objects flow into the software, are transformed by processing elements, and the resultant data objects flow out of the software.

Data objects are represented by labeled arrows, while transformations are represented by circles (also called bubbles). The DFD is presented in a hierarchical fashion, where the first data flow model represents the system as a whole. Subsequent DFDs refine the context diagram (Level 0 DFD), providing increasing levels of detail with each refinement.

The DFD enables software engineers to develop models of both the information domain and the functional domain at the same time. As the DFD is refined into greater levels of detail, the analyst performs an implicit functional decomposition of the system. Additionally, DFD refinement results in a corresponding refinement of the data flow as it moves through the processes that embody the application.

A context-level DFD for the system shows that the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrows represent data objects or an object hierarchy, providing a clear overview of how data moves within the system.

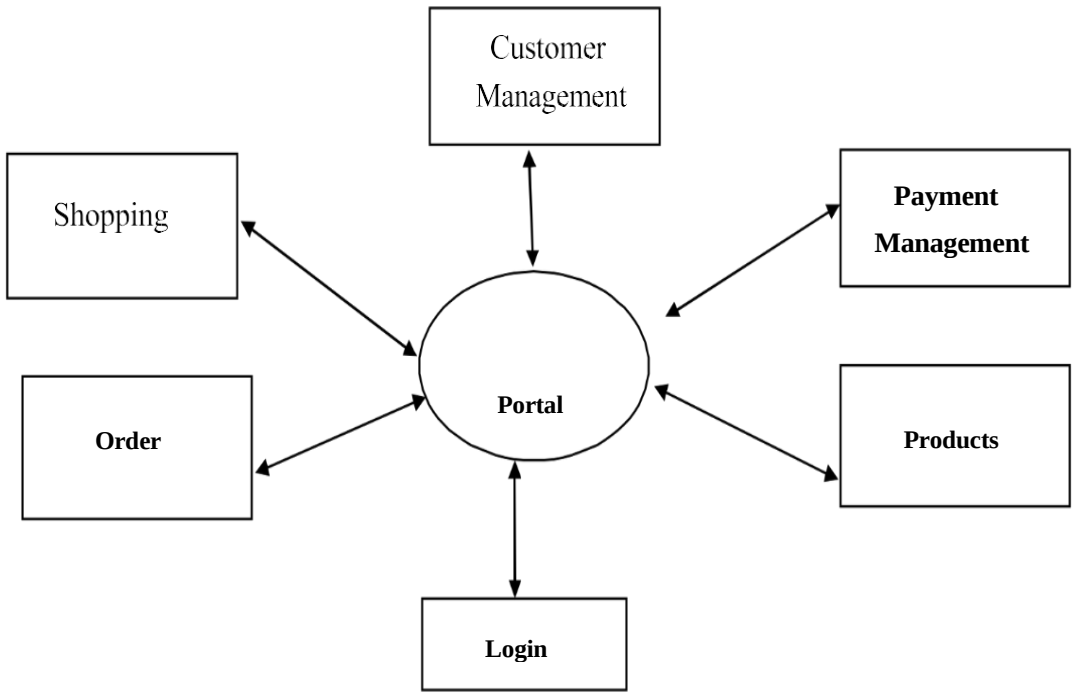
RULESFORDFD:

- Fix the scope of the system by defining it through context diagrams.
- Organize the DFD so that the main sequence of actions is clear.
- Ensure the diagram reads from left to right and top to bottom for better readability.
- Identify all inputs and outputs in the system.
- Label each process internal to the system with rounded circles.
- A process is required for all data transformations and transfers.

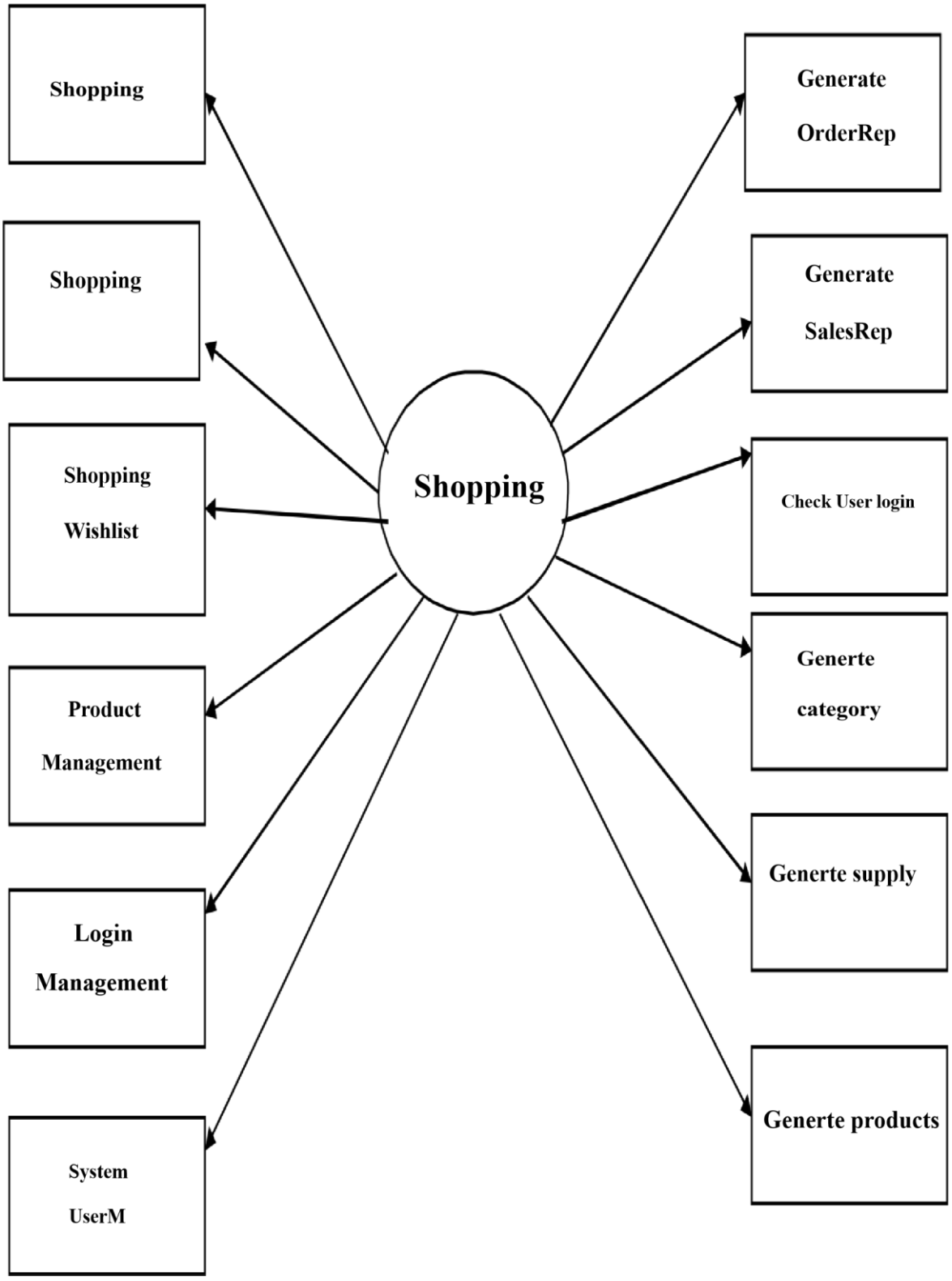
Never connect a data store directly to a data source, destination, or another data store using just a data flow arrow.

- Do not include hardware details or control information in the DFD.
- Ensure process names accurately describe what each process does . Every process must be named; avoid unnamed processes.
- Represent external sources and destinations of data using squares.
- Number each occurrence of repeated external entities.
- Identify all data flows for each process step, except for simple record retrievals.
- Label data flows on each arrow to indicate what data is being transferred.
- Use detailed flow arrows to show data movements clearly.

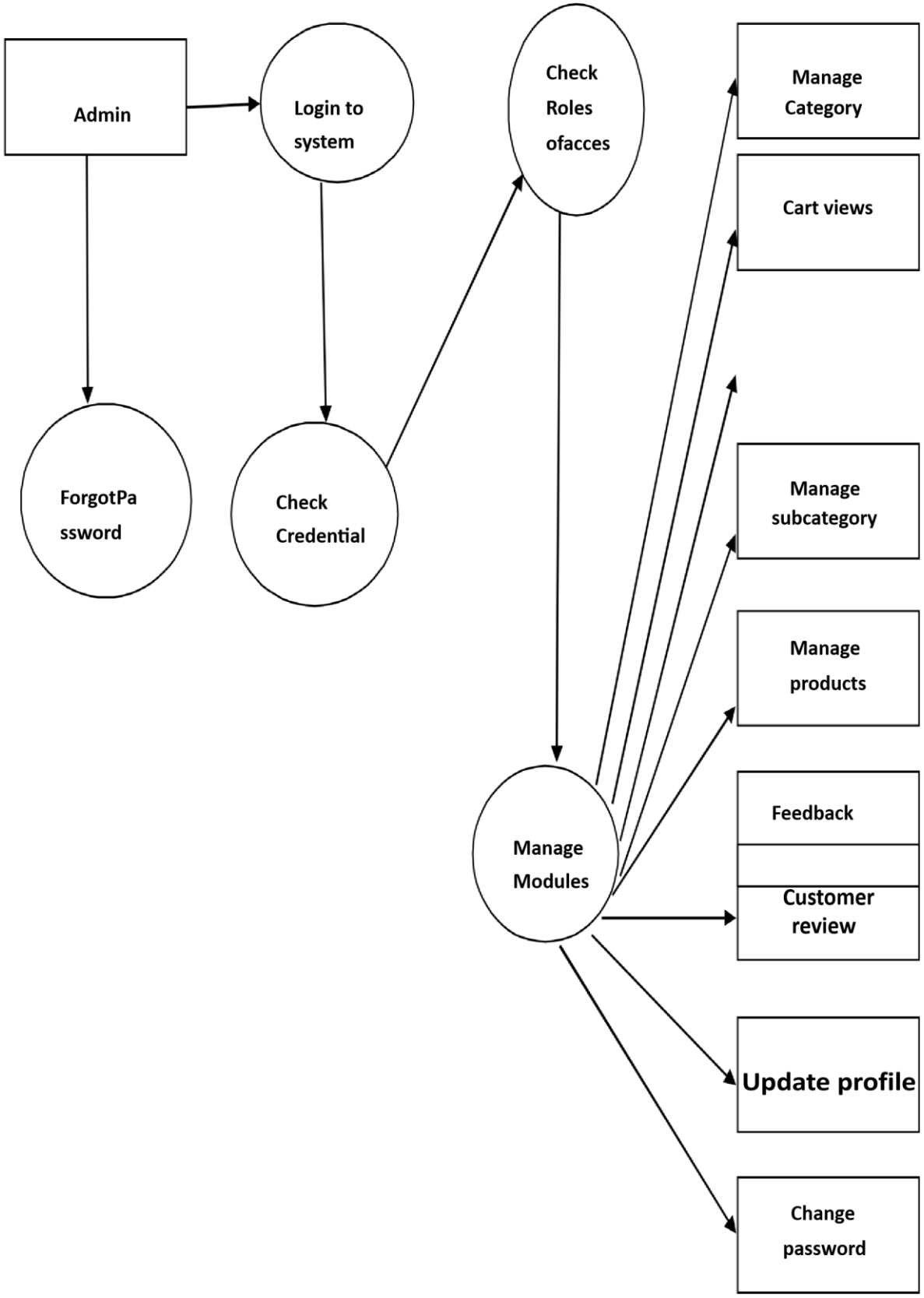
Context Diagram 1



Context Diagram 2



Second Level DFD



E-R Diagrams :

The Entity-Relationship (ER) model was originally proposed by Peter Chen in 1976 [Chen76] to unify the network and relational database views. Simply stated, the ER model is a conceptual data model that views the real world as entities and relationships.

A basic component of the model is the Entity-Relationship diagram, which is used to visually represent data objects. Since Chen wrote his paper, the model has been extended, and today it is commonly used for database design. For the database designer, the utility of the ER model includes.

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with minimal training. Therefore, the model can be used by database designers to communicate the design to end users.

The model can also serve as a design plan for database developers to implement a data model in specific database management software.

Connectivity and Cardinality

The basic types of connectivity for relationships are:

One-to-One (1:1)

A one-to-one relationship exists when at most one instance of entity A is associated with one instance of entity B.

Example:

Employees in a company are each assigned their own office. Each employee has a unique office, and each office belongs to a single employee.

One-to-Many (1:N)

A one-to-many relationship exists when one instance of entity A is associated with zero, one, or many instances of entity B. However, for one instance of entity B, there is only one instance of entity A.

Example:

- A department has many employees.
- Each employee is assigned to one department.

Many-to-Many (M:N)

A many-to-many relationship, sometimes called non-specific, occurs when one instance of entity A is associated with zero, one, or many instances of entity B, and one instance of entity B is also associated with zero, one, or many instances of entity A.

Example:

A student can enroll in multiple courses, and each course can have multiple students. The connectivity of a relationship describes the mapping of associated entities within a database.

ER Diagram Notations

There is no standard notation for representing data objects in ER diagrams. Each modeling methodology uses its own notation.

The original Chen notation is widely used in academic texts and journals but is rarely seen in either CASE tools or publications by non-academics. Today, there are multiple notations used, with some of the most common notations being: Bachman notation, Crow's Foot notation, IDEFIX notation.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a specific set of symbols to represent the cardinality of a connection The notation used in this document follows Martin notation, with the following.

- Entities are represented by labelled rectangles. The label is the name of the entity, which should be a singular noun. basic ER constructs:
- Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line, and it should be a verb.
- Attributes (when included) are listed inside the entity rectangle. Attributes that serve as identifiers are underlined. Attribute names should also be singular nouns.
- Cardinality of "many" is represented by a line ending in a crow's foot. If the crow’s foot is omitted, the cardinality is assumed to be one.
- Existence is represented by either a circle or a perpendicular bar on the line: Mandatory existence is shown by a bar (1) placed next to the entity to indicate that an instance is

required Optional existence is shown by a circle next to the entity, indicating that the instance is optional.

B.TABEL STRUCTURE

- This table structure stores platform-password mappings securely.

Column Name	Data Type	Description
id	INTEGER	Primary key for unique identification.
platform	VARCHAR (50)	Name of the platform (e.g., Instagram).
password	TEXT	Encrypted password stored for the platform.
Created at	TIMESTAMP	Timestamp of when the password was created.

Table: Users (Optional for Advanced System)

Column Name	Data Type	Description
User id	INTEGER	Primary key for unique identification.
username	VARCHAR (50)	Username of the user.
Password hash	TEXT	Encrypted password for user authentication.

Relationships

- If implementing a multi-user system:
- A one-to-many relationship between Users and Passwords**.**

Sample Data Passwords Table:

id	**platform**	**password**	**Created at**
1	Instagram	aBc!1234	2025-01-30 12:00
2	Chrome	XyZ@5678	2025-01-30 12:30
3	Snapchat	LmN$91011	2025-01-30 12:45

C. SAMPLECODING

LOGIN PAGE

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register & Login</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/6.5.1/css/all.min.css">
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container" id="signup" style="display:none;">
      <h1 class="form-title">Register</h1>
      <form method="post" action="register.php">
        <div class="input-group">
           <i class="fas fa-user"></i>
           <input type="text" name="fName" id="fName" placeholder="First Name" required>
           <label for="fname">First Name</label>
        </div>
        <div class="input-group">
            <i class="fas fa-user"></i>
            <input type="text" name="lName" id="lName" placeholder="Last Name" required>
            <label for="lName">Last Name</label>
```

```
    </div>
    <div class="input-group">
        <i class="fas fa-envelope"></i>
        <input type="email" name="email" id="email" placeholder="Email" required>
        <label for="email">Email</label>
    </div>
    <div class="input-group">
        <i class="fas fa-lock"></i>
        <input type="password" name="password" id="password" placeholder="Password" required>
        <label for="password">Password</label>
    </div>
   <input type="submit" class="btn" value="Sign Up" name="signUp">
  </form>
  <p class="or">
    -------or------
  </p>
  <div class="icons">
    <i class="fab fa-google"></i>
    <i class="fab fa-facebook"></i>
  </div>
  <div class="links">
    <p>Already Have Account ?</p>
    <button id="signInButton">Sign In</button>
  </div>
</div>

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

```
<h1 class="form-title">Sign In</h1>
<form method="post" action="register.php">
  <div class="input-group">
     <i class="fas fa-envelope"></i>
      <input type="email" name="email" id="email" placeholder="Email" required>
      <label for="email">Email</label>
  </div>
  <div class="input-group">
      <i class="fas fa-lock"></i>
      <input type="password" name="password" id="password" placeholder="Password" required>
      <label for="password">Password</label>
  </div>
  <p class="recover">
    <a href="#">Recover Password</a>
  </p>
 <input type="submit" class="btn" value="Sign In" name="signIn">
</form>
<p class="or">
  -------or------.
</p>
<div class="icons">
  <i class="fab fa-google"></i>
  <i class="fab fa-facebook"></i>
</div>
<div class="links">
  <p>Don't have account yet?</p>
  <button id="signUpButton">Sign Up</button>
```

```
        </div>
      </div>
      <script src="script.js"></script>
</body>
</html>
```

Register.php

```
<?php

include 'connect.php';

if(isset($_POST['signUp'])){
    $firstName=$_POST['fName'];
    $lastName=$_POST['lName'];
    $email=$_POST['email'];
    $password=$_POST['password'];
    $password=md5($password);

     $checkEmail="SELECT * From users where email='$email'";
     $result=$conn->query($checkEmail);
if($result->num_rows>0){          echo
"Email Address Already Exists !";
     }
else{
        $insertQuery="INSERT INTO users(firstName,lastName,email,password)
VALUES ('$firstName','$lastName','$email','$password')";            if($conn-
>query($insertQuery)==TRUE){                header("location: index.php");
```

```
        }       else{       echo
"Error:".$conn->error;
        }
    }

}

if(isset($_POST['signIn'])){
  $email=$_POST['email'];
  $password=$_POST['password'];
  $password=md5($password) ;

  $sql="SELECT * FROM users WHERE email='$email' and password='$password'";
  $result=$conn->query($sql);    if($result-
>num_rows>0){    session_start();
   $row=$result->fetch_assoc();
$_SESSION['email']=$row['email'];    header("Location:
homepage.php");
   exit();   }   else{    echo "Not Found, Incorrect
Email or Password";
  }

}
?>
```

Connect.php

```
<?php

$host="localhost";
```

```
$user="root";
$pass="";
$db="login";
$conn=new mysqli($host,$user,$pass,$db); if($conn-
>connect_error){      echo "Failed to connect
DB".$conn->connect_error;
}
?>
```

Logout.php

```
<?php
session_destroy();
header("location: index.php");
?>
```

Script.js

```
const
signUpButton=document.getElementById('signUpButton'); const
signInButton=document.getElementById('signInButton'); const
signInForm=document.getElementById('signIn'); const
signUpForm=document.getElementById('signup');

signUpButton.addEventListener('click',function(){
signInForm.style.display="none";      signUpForm.style.display="block";
}) signInButton.addEventListener('click',
function(){     signInForm.style.display="block";
signUpForm.style.display="none";
})
```

Stylee.css

```
} .or{     font-size:1.1rem;
margin-top:0.5rem;     text-
align:center; } .icons{
text-align:center; } .icons i{
color:rgb(125,125,235);
padding:0.8rem 1.5rem;
border-radius:10px;     font-
size:1.5rem;
cursor:pointer;
border:2px solid #dfe9f5;
margin:0 15px;
transition:1s; }
.icons i:hover{ background:#07001f;
   font-size:1.6rem; border:2px solid
   rgb(125,125,235);
}
.links{     display:flex;
justify-content:space-around;
padding:0 4rem;     margin-
top:0.9rem;     font-
weight:bold;
} button{
color:rgb(125,125,235);
border:none;     background-
color:transparent;     font-
size:1rem;     font-weight:bold;
```

```
} button:hover{     text-
decoration:underline;
color:blue;
}
```

HOME PAGE

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Tech2etc Ecommerce</title>
   <link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">     <link rel="stylesheet" href="style.css">
</head>
<body>

<!-- Header -->
<section id="header">
   <a href="#"><img src="img/logo.png" class="logo" alt=""></a>
   <div>
      <ul id="navbar">
         <li><a href="index.html" class="active">Home</a></li>
         <li><a href="shop.html">Shop</a></li>
         <li><a href="blog.html">Blog</a></li>
         <li><a href="about.html">About</a></li>
         <li><a href="contact.html">Contact</a></li>
```

```
            <li id="lg-bag"><a href="cart.html"><i class="far fa-shopping-bag"></i></a></li>
            <a href="#" id="close"><i class="far fa-times"></i></a>
        </ul>
    </div>
    <div id="mobile">
        <a href="cart.html"><i class="far fa-shopping-bag"></i></a>
        <i id="bar" class="fas fa-outdent"></i>
    </div>
</section>

<!-- Hero Section -->
<section id="hero">
    <h4>Trade-in-offer</h4>
    <h2>Super Value Deals</h2>
    <h1>On all products</h1>
    <p>Save more with coupons & up to 70% off!</p>
    <button>Shop Now</button>
</section>

<!-- Features Section -->
<section id="feature" class="section-p1">
    <div class="fe-box" data-feature="Free Shipping"></div>
    <div class="fe-box" data-feature="Online Shopping"></div>
    <div class="fe-box" data-feature="Save Money"></div>
    <div class="fe-box" data-feature="Promotion"></div>
    <div class="fe-box" data-feature="Free Happy Sell"></div>
    <div class="fe-box" data-feature="24/7 Support"></div>
```

```
</section>

<!-- Featured Products -->
<section id="product1" class="section-p1">
    <h2>Featured Products</h2>
    <p>Summer Collection New Modern Design</p>
    <div class="pro-container" id="featured-products"></div>
</section>

<!-- Banner -->
<section id="banner" class="section-m1">
    <h4>Repair Services</h4>
    <h2>Up to <span>70% off</span> - All t-shirts & Accessories</h2>
    <button class="normal">Explore More</button>
</section>

<!-- New Arrivals -->
<section id="product1" class="section-p1">
    <h2>New Arrivals</h2>
    <p>Summer Collection New Modern Design</p>
    <div class="pro-container" id="new-arrivals"></div>
</section>

<!-- Small Banners -->
<section id="sm-banner" class="section-p1">
    <div class="banner-box">
        <h4>Crazy Deals</h4>
```

```
        <h2>Buy 1 Get 1 Free</h2>
        <span>The best classic dress is on sale</span>
        <button class="white">Learn More</button>
    </div>
    <div class="banner-box banner-box2">
        <h4>Spring/Summer</h4>
        <h2>Upcoming Seasons</h2>
        <span>New collections available soon</span>
        <button class="white">Collection</button>
    </div>
</section>

<!-- Large Banners -->
<section id="banner3">
    <div class="banner-box"><h2>SEASONAL SALE</h2></div>
    <div class="banner-box banner-box2"><h2>NEW FOOTWEAR
COLLECTION</h2><h3>Spring / Summer 2025</h3></div>
    <div class="banner-box banner-box3"><h2>T-SHIRTS</h2><h3>New Trendy
Prints</h3></div>
</section>

<!-- Newsletter -->
<section id="newsletter" class="section-p1 section-m1">
    <div class="newstext">
        <h4>Sign Up for Newsletters</h4>
        <p>Get email updates about our latest shop and <span>special offers.</span></p>
</div>
    <div class="form">
```

```
        <input type="text" placeholder="Your email address">
        <button class="normal">Sign up</button>
    </div>
</section>

<!-- Footer -->
<footer class="section-p1">
    <div class="col">
        <img class="logo" src="img/logo.png" alt="">
        <h4>Contact</h4>
        <p><strong>Address:</strong> 562 Wellington Road, San Francisco</p>
        <p><strong>Phone:</strong> +01 2222 365 / (+91) 01 2345 6789</p>
        <p><strong>Hours:</strong> 10:00 - 18:00, Mon - Sat</p>
        <div class="Follow">
            <h4>Follow Us</h4>
            <div class="icon">
                <i class="fab fa-facebook-f"></i>
                <i class="fab fa-twitter"></i>
                <i class="fab fa-instagram"></i>
                <i class="fab fa-pinterest"></i>
                <i class="fab fa-youtube"></i>
            </div>
        </div>
    </div>
    <div class="col">
        <h4>About</h4>
        <a href="#">About Us</a>
```

```
        <a href="#">Delivery Information</a>
        <a href="#">Privacy Policy</a>
        <a href="#">Terms & Conditions</a>
        <a href="#">Contact Us</a>
    </div>
    <div class="col">
        <h4>My Account</h4>
        <a href="#">Sign In</a>
        <a href="#">View Cart</a>
        <a href="#">My Wishlist</a>
        <a href="#">Track My Order</a>
        <a href="#">Help</a>
    </div>
    <div class="col install">
        <h4>Install Apps</h4>
        <p>From App Store or Google Play</p>
        <div class="row">
            <img src="img/pay/app.jpg" alt="">
            <img src="img/pay/play.jpg" alt="">
        </div>
        <p>Secured Payment Gateways</p>
        <img src="img/pay/pay.png" alt="">
    </div>
    <div class="copyright">
        <p>© 2024, HTML CSS Ecommerce Template</p>
</div>
</footer>
```

```
<!-- JavaScript -->
<script src="script.js"></script>
</body>
</html>
```

Shop.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Optimized Ecommerce</title>
    <link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <!-- Header -->
    <section id="header">
        <a href="#"><img src="img/logo.png" class="logo" alt="Logo"></a>
        <div>
            <ul id="navbar">
                <li><a href="index.html">Home</a></li>
                <li><a href="shop.html" class="active">Shop</a></li>
                <li><a href="blog.html">Blog</a></li>
                <li><a href="about.html">About</a></li>
                <li><a href="contact.html">Contact</a></li>
                <li id="lg-bag"><a href="cart.html"><i class="far fa-shopping-bag"></i></a></li>
```

```
            <a href="#" id="close"><i class="far fa-times"></i></a>
        </ul>
    </div>
    <div id="mobile">
        <a href="cart.html"><i class="far fa-shopping-bag"></i></a>
        <i id="bar" class="fas fa-outdent"></i>
    </div>
</section>

<!-- Banner -->
<section id="page-header1">
    <h2>#stayhome</h2>
    <p>Save more with coupons & up to 70% off!</p>
</section>

<!-- Product Section -->
<section id="product1" class="section-p1">
    <div class="pro-container" id="product-list"></div>
</section>

<!-- Pagination -->
<section id="pagination" class="section-p1">
    <a href="#">1</a>
    <a href="#">2</a>
    <a href="#"><i class="fal fa-long-arrow-alt-right"></i></a>
</section>

<!-- Newsletter -->
```

```
<section id="newsletter" class="section-p1 section-m1">
    <div class="newstext">
        <h4>Sign Up for Newsletters</h4>
        <p>Get email updates about our latest shop and <span>special offers.</span></p>
    </div>
    <div class="form">
        <input type="text" placeholder="Your email address">
        <button class="normal">Sign up</button>
    </div>
</section>

<!-- Footer -->
<footer class="section-p1">
    <div class="col">
        <img class="logo" src="img/logo.png" alt="Logo">
        <h4>Contact</h4>
        <p><strong>Address:</strong> 562 Wellington Road, Street 32, San Francisco</p>
        <p><strong>Phone:</strong> +01 2222 365 / (+91) 01 2345 6789</p>
        <p><strong>Hours:</strong> 10:00 - 18:00, Mon - Sat</p>
        <div class="Follow">
            <h4>Follow Us</h4>
            <div class="icon">
                <i class="fab fa-facebook-f"></i>
                <i class="fab fa-twitter"></i>
                <i class="fab fa-instagram"></i>
                <i class="fab fa-pinterest"></i>
                <i class="fab fa-youtube"></i>
```

```
        </div>
    </div>
</div>
<div class="col">
    <h4>About</h4>
    <a href="#">About Us</a>
    <a href="#">Delivery Information</a>
    <a href="#">Privacy Policy</a>
    <a href="#">Terms & Conditions</a>
    <a href="#">Contact Us</a>
</div>
<div class="col">
    <h4>My Account</h4>
    <a href="#">Sign In</a>
    <a href="#">View Cart</a>
    <a href="#">My Wishlist</a>
    <a href="#">Track My Order</a>
    <a href="#">Help</a>
</div>
<div class="col install">
    <h4>Install App</h4>
    <p>From App Store or Google Play</p>
    <div class="row">
        <img src="img/pay/app.jpg" alt="">
        <img src="img/pay/play.jpg" alt="">
    </div>
    <p>Secure Payment Gateways</p>
```

```
            <img src="img/pay/pay.png" alt="">
        </div>
        <div class="copyright">
            <p>© 2024, HTML CSS Ecommerce Template</p>
        </div>
    </footer>

    <script src="script.js"></script>
</body>
</html>
```

Blog.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Blog - Dress Shopping</title>
    <link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <!-- Include Header -->
    <script>fetch('header.html').then(res => res.text()).then(data =>
document.body.insertAdjacentHTML('afterbegin',  data));</script>

    <section id="page-header">
        <h2>#ReadMore</h2>
```

About.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>About Us - Dress Shopping</title>

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

</head>

<body>


    <!-- Include Header -->

    <script>fetch('header.html').then(res => res.text()).then(data =>
document.body.insertAdjacentHTML('afterbegin',  data));</script>


    <section id="page-header">

        <h2>#AboutUs</h2>

        <p>Learn more about our story and mission.</p>

    </section>


    <section class="container">

        <h2>Our Story</h2>

        <p>We started Dress Shopping with a vision to provide stylish, high-quality fashion at affordable
prices...</p>

    </section>


    <!-- Include Footer -->

    <script>fetch('footer.html').then(res => res.text()).then(data =>
document.body.insertAdjacentHTML('beforeend', data));</script>
```

```
</body>

</html>
```

Contact.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Contact Us</title>

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

</head>

<body>


    <!-- Include Header -->

    <script>fetch('header.html').then(res => res.text()).then(data =>
document.body.insertAdjacentHTML('afterbegin',  data));</script>


    <section id="page-header">

        <h2>#Let'sTalk</h2>

        <p>Leave a message. We love to hear from you!</p>

    </section>


    <section id="contact-details">

        <div>

            <h4>Get in Touch</h4>

            <p><i class="icon"> 📍 </i> 02/8, Four Roads, AS Hall, Salem - 132815</p>
```

```
        <p><i class="icon">📧</i> <a
href="mailto:contact@example.com">contact@example.com</a></p>

        <p><i class="icon">📞</i> <a href="tel:7695949797">+91 7695949797</a></p>

      </div>

    </section>


    <!-- Include Footer -->

    <script>fetch('footer.html').then(res => res.text()).then(data =>
document.body.insertAdjacentHTML('beforeend', data));</script>


</body>

</html>
```

Cart.html

```
<footer class="section-p1">

    <div class="col">

        <img class="logo" src="img/logo.png" alt="Logo">

        <h4>Contact</h4>

        <p><strong>Address:</strong> 562 Wellington Road, San Francisco</p>

        <p><strong>Phone:</strong> +01 2222 365 / (+91) 01 2345 6789</p>

        <p><strong>Hours:</strong> 10:00 - 18:00, Mon - Sat</p>

    </div>


    <div class="col">

        <h4>About</h4>

        <a href="#">About Us</a>

        <a href="#">Privacy Policy</a>

        <a href="#">Contact Us</a>

    </div>
```

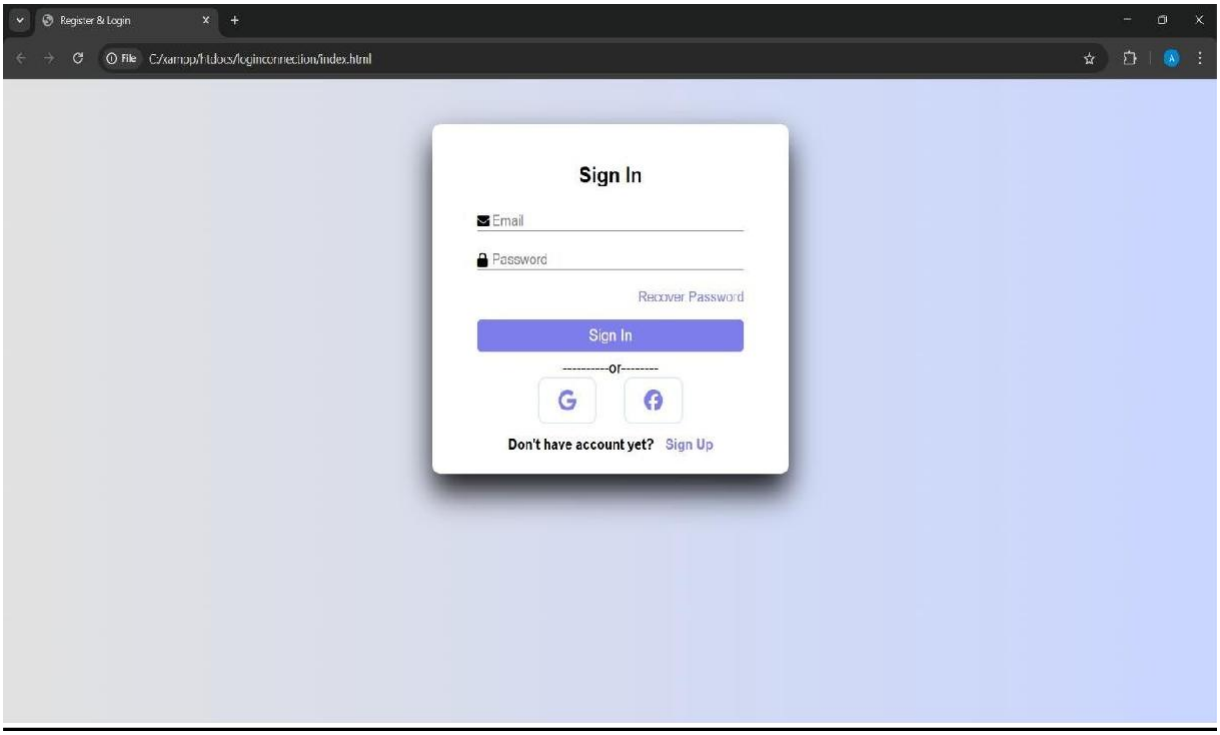
```
    <div class="col">
        <h4>My Account</h4>
        <a href="#">Sign In</a>
        <a href="#">View Cart</a>
        <a href="#">Track My Order</a>
    </div>

    <div class="col install">
        <h4>Install Apps</h4>
        <p>From App Store or Google Play</p>
        <div class="row">
            <img src="img/pay/app.jpg" alt="App Store">
            <img src="img/pay/play.jpg" alt="Google Play">
        </div>
    </div>
    <div class="copyright">
        <p>© 2025 Dress Shopping Blog. All rights reserved.</p>
    </div>
</footer> Script.css const bar =
document.getElementById('bar'); const close =
document.getElementById('close'); const nav =
document.getElementById('navbar');

if (bar) {
bar.addEventListener('click', () => {
nav.classList.add('active')
```

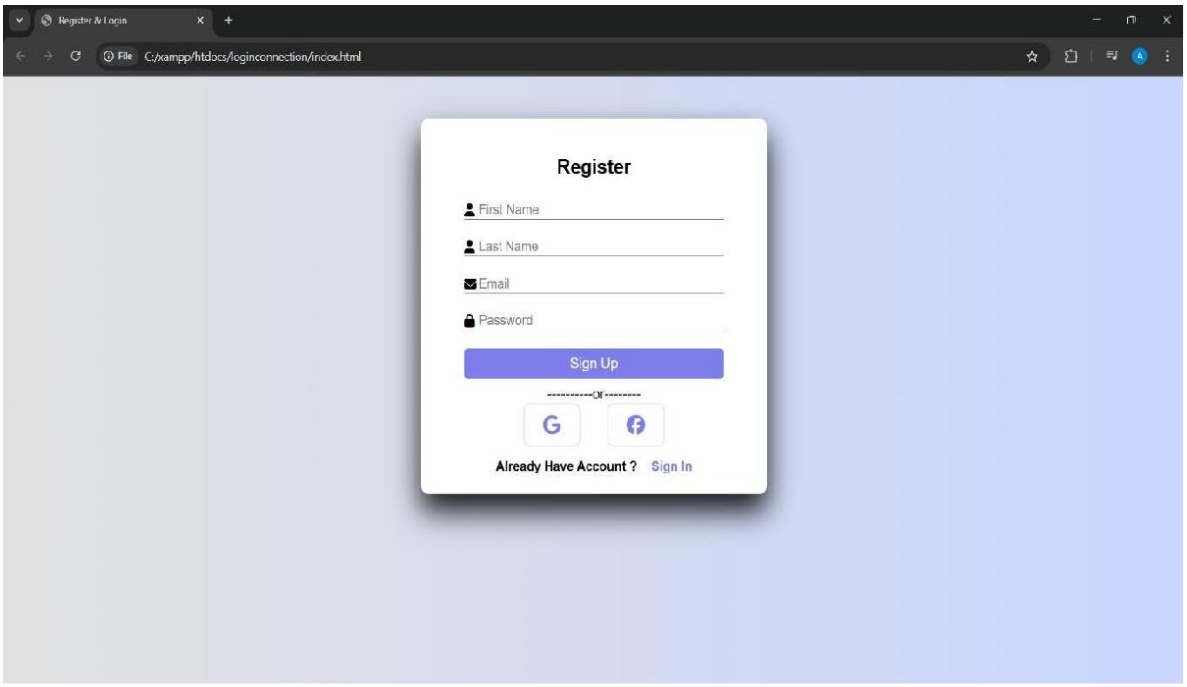
```
}) } if (bar2) {
close.addEventListener('click', () => {
nav.classList.remove('active')

})
}
// Add any dynamic functionality if required
console.log("Page loaded successfully!");

document.getElementById('contactForm').addEventListener('submit', function (e) {
e.preventDefault();      alert('Message sent successfully!');
});
// Product Data const
products = [
   { img: "img/products/f1.jpg", name: "Cartoon Astronaut T-shirt", brand: "Adidas", price: "₹699"
},
   { img: "img/products/f2.jpg", name: "Cartoon Astronaut T-shirt", brand: "Adidas", price: "₹999"
},
   { img: "img/products/f3.jpg", name: "Cartoon Astronaut T-shirt", brand: "Adidas", price: "₹899"
},
   { img: "img/products/f4.jpg", name: "Cartoon Astronaut T-shirt", brand: "Adidas", price: "₹299"
}
];
// Function to generate product cards
function generateProducts(containerId, productList) {
const container = document.getElementById(containerId);
container.innerHTML = productList.map(product => `
      <div class="pro">
```

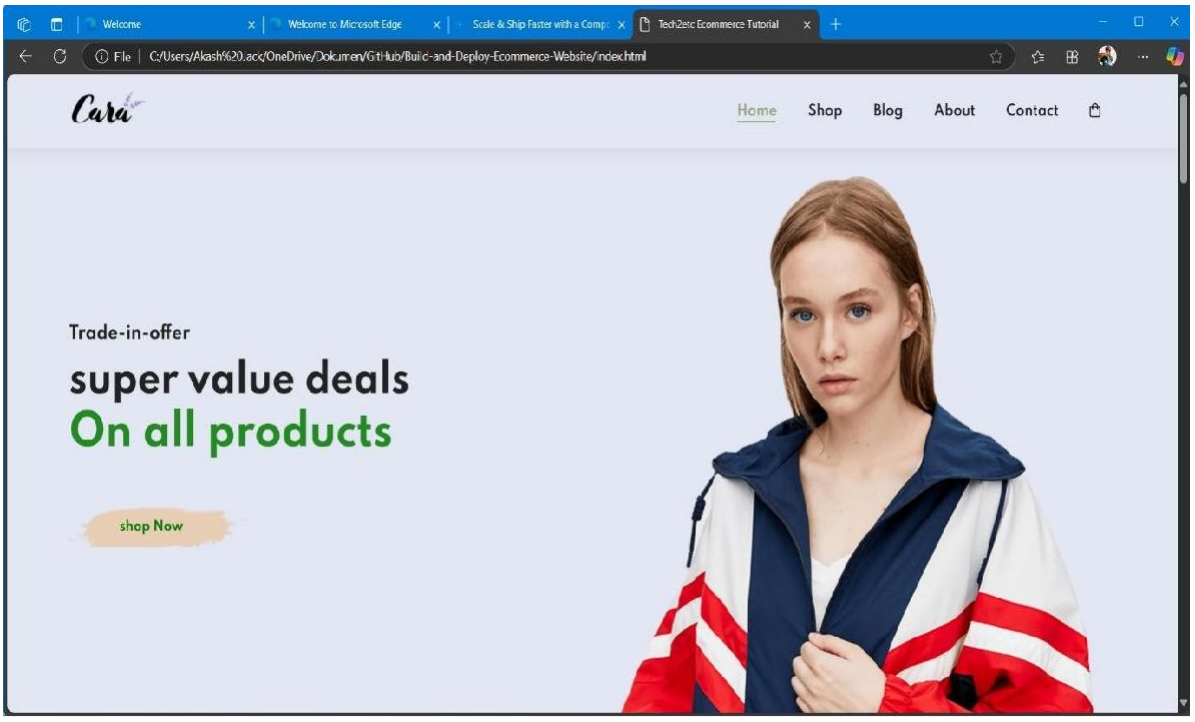
```
        <img src="${product.img}" alt="">
        <div class="des">
            <span>${product.brand}</span>
            <h5>${product.name}</h5>
            <h4>${product.price}</h4>
        </div>
        <a href="#"><i class="fal fa-shopping-cart cart"></i></a>
      </div>
    `).join('');
}

// Populate Products generateProducts("featured-products",
products); generateProducts("new-arrivals", products);
// Product Data (Can be moved to a separate JSON file) const
products = [
    { img: "img/products/f1.jpg", brand: "Adidas", name: "Cartoon Astronaut T-shirt", price: "₹699"
},
    { img: "img/products/f2.jpg", brand: "Adidas", name: "Cartoon Astronaut T-shirt", price: "₹999"
},
    { img: "img/products/f3.jpg", brand: "Adidas", name: "Cartoon Astronaut T-shirt", price: "₹899"
},
    { img: "img/products/f4.jpg", brand: "Adidas", name: "Cartoon Astronaut T-shirt", price: "₹299"
},
    { img: "img/products/f5.jpg", brand: "Adidas", name: "Cartoon Astronaut T-shirt", price: "₹2199"
},
    { img: "img/products/f6.jpg", brand: "Adidas", name: "Cartoon Astronaut T-shirt", price: "₹5678"
},
    { img: "img/products/f7.jpg", brand: "Adidas", name: "Cartoon Astronaut T-shirt", price: "₹3599"
},
    { img: "img/products/f8.jpg", brand: "Adidas", name: "Cartoon Astronaut T-shirt", price: "₹1999"
}
```

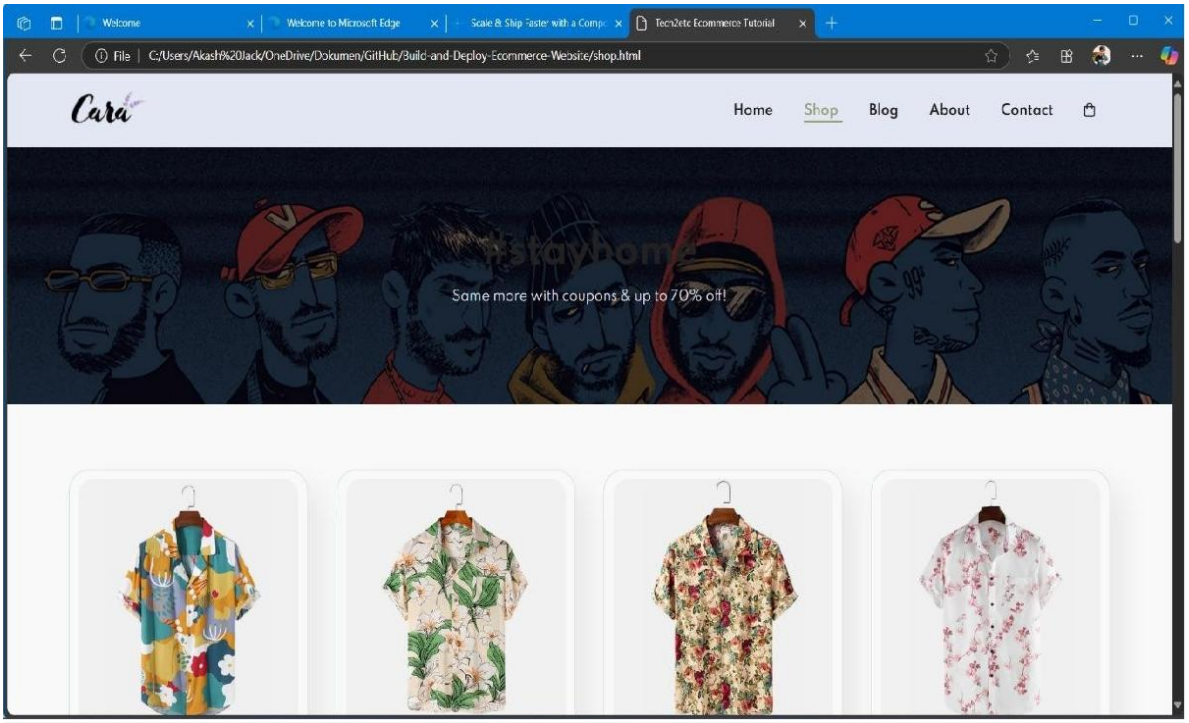
```
  border-radius: 4px;
margin: 15px 0;
}
.fe-box img { width: 100%; margin-bottom: 10px; }
.fe-box h6 {   display:
inline-block;
padding: 9px 8px;
border-radius: 4px;
background: #fddde4;
color: forestgreen;
}
.fe-box:nth-child(2) h6 { background: #d4fc76; }
.fe-box:nth-child(3) h6 { background: #99c1fa; }
.fe-box:nth-child(4) h6 { background: #fbaadf; }
.fe-box:nth-child(5) h6 { background: #fb9b9b; }
.fe-box:nth-child(6) h6 { background: #cce7d0; }

#product1 {   text-align: center;
} .pro-container {   display:
flex;   flex-wrap: wrap;   justify-
content: space-between;
padding-top: 20px;
}
.pro { width: 23%;
  min-width: 250px;
  padding: 10px;
  border: 1px solid
```

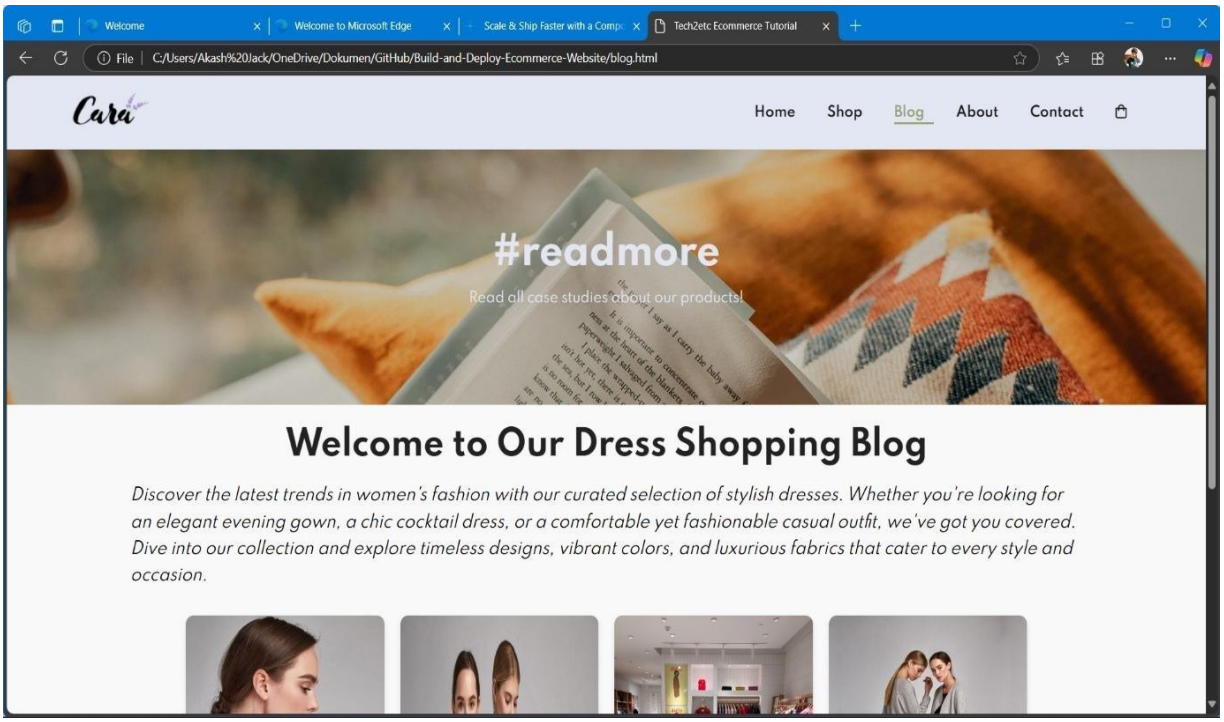
D.SAMPLE INPUT AND OUTPUT

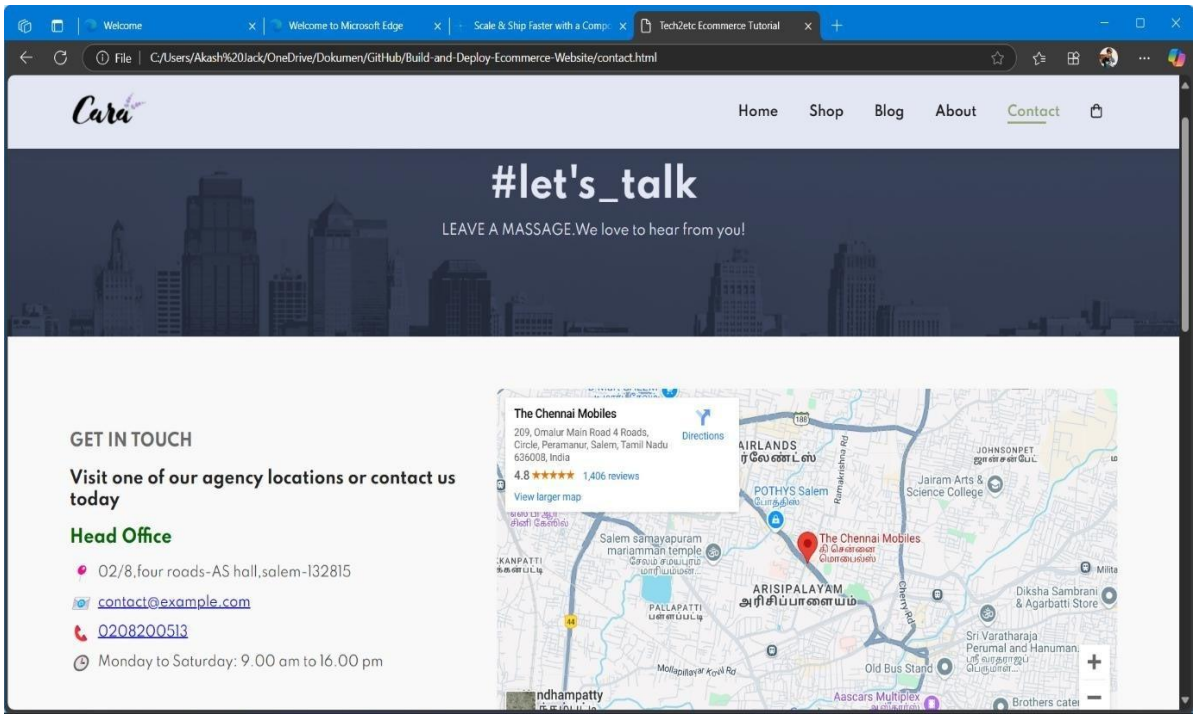


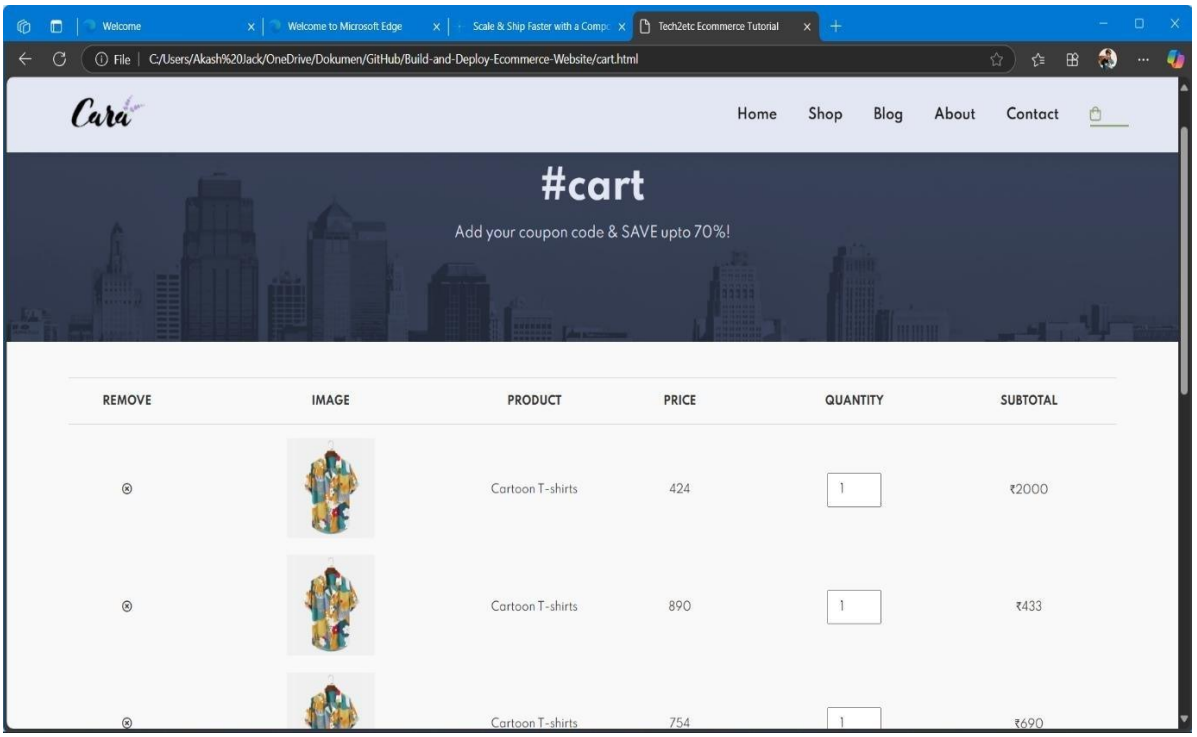



Welcome
Welcome to Microsoft Edge
Tech2etc Ecommerce Tutorial
File | C:/Users/Akash%20.acc/OneDrive/Dokumen/GitHub/Build-and-Deploy-Ecommerce-Website/index.html
Cara
Home
Shop
Blog
About
Contact
Trade-in-offer
super value deals
On all products
shop Now


Welcome
Welcome to Microsoft Edge
Tech2etc Ecommerce Tutorial
File | C:/Users/Akash%20Jack/OneDrive/Dokumen/GitHub/Build-and-Deploy-Ecommerce-Website/shop.html
Cara
Home
Shop
Blog
About
Contact
#stayhome
Same more with coupons & up to 70% off!


Welcome
Welcome to Microsoft Edge
Scale & Ship Faster with a Compo
Tech2etc Ecommerce Tutorial
File | C:/Users/Akash%20Jack/OneDrive/Dokumen/GitHub/Build-and-Deploy-Ecommerce-Website/blog.html
Cara
Home
Shop
Blog
About
Contact
#readmore
Read all case studies about our products!
Welcome to Our Dress Shopping Blog
Discover the latest trends in women's fashion with our curated selection of stylish dresses. Whether you're looking for an elegant evening gown, a chic cocktail dress, or a comfortable yet fashionable casual outfit, we've got you covered. Dive into our collection and explore timeless designs, vibrant colors, and luxurious fabrics that cater to every style and occasion.


Welcome
Welcome to Microsoft Edge
Scale & Ship Faster with a Compo
Tech2etc Ecommerce Tutorial
File | C:/Users/Akash%20Jack/OneDrive/Dokumen/GitHub/Build-and-Deploy-Ecommerce-Website/contact.html
Cara
Home
Shop
Blog
About
Contact
#let's_talk
LEAVE A MASSAGE.We love to hear from you!
GET IN TOUCH
Visit one of our agency locations or contact us today
Head Office
02/8,four roads-AS hall,salem-132815
contact@example.com
0208200513
Monday to Saturday: 9.00 am to 16.00 pm
The Chennai Mobiles
209, Omalur Main Road 4 Roads, Circle, Peramanur, Salem, Tamil Nadu 636008, India
Directions
4.8 1,406 reviews
View larger map
POTHYS Salem
Jairam Arts & Science College
Salem samayapuram mariamman temple
The Chennai Mobiles
ARISIPALAYAM
PALLAPATTI
Diksha Sambrani & Agarbatti Store
Old Bus Stand
Aascars Multiplex


Welcome
Welcome to Microsoft Edge
Scale & Ship Faster with a Comp
Tech2etc Ecommerce Tutorial
File | C:/Users/Akash%20Jack/OneDrive/Dokumen/GitHub/Build-and-Deploy-Ecommerce-Website/cart.html
Cara
Home
Shop
Blog
About
Contact
#cart
Add your coupon code & SAVE upto 70%!
REMOVE
IMAGE
PRODUCT
PRICE
QUANTITY
SUBTOTAL
Cartoon T-shirts
424
1
₹2000
Cartoon T-shirts
890
1
₹433
Cartoon T-shirts
754
1
₹690