

E-COMMERCE SITE FOR CLOTHING SHOP

ABSTRACT:

The **E-commerce site for clothing business** is a web-based application to expand the business through online. This project focuses on developing an e-commerce website tailored for a clothing business, aimed at providing a seamless online shopping experience for customers. The platform will feature an intuitive user interface, responsive design, and robust functionality to meet the demands of modern consumers. Key features include a dynamic product catalog, secure payment gateway integration, user-friendly navigation and personalized recommendations. The system is designed to enhance customer engagement, streamline operations, and support scalability, ultimately driving business growth in the competitive online marketplace.

By leveraging **modern web technologies (HTML, CSS, JavaScript, PHP, and MySQL)**, this platform aims to revolutionize the online fashion shopping experience, providing **convenience, security, and efficiency** for both customers and businesses.

1. INTRODUCTION

1.1. ABOUT THE PROJECT

The **E-Commerce Clothing Website**, "Yuv Clothing," is a comprehensive online shopping platform designed to cater to the evolving fashion needs of consumers. In the digital age, online shopping has transformed the retail industry by providing convenience, accessibility, and a wide range of choices to customers. The emergence of e-commerce platforms has enabled businesses to reach a broader audience while offering customers the ability to browse, select, and purchase products from the comfort of their homes.

The primary objective of this website is to bridge the gap between fashion retailers and customers by offering a **user-friendly, secure, and feature-rich online shopping experience**. The platform specializes in trendy fashion for Gen Z and includes an extensive collection of **shirts, t-shirts, pants, and combo outfits** from various brands. Customers can explore product categories, view high-quality images, check ratings and reviews, and make informed purchase decisions.

The **E-Commerce Clothing Website** integrates essential features such as **real-time cart updates, secure payment gateways, and seamless order tracking** to enhance the overall shopping experience. Additionally, the platform employs modern web technologies like **HTML, CSS, JavaScript, PHP, and MySQL** to ensure smooth navigation, fast loading times, and a responsive interface across all devices.

2. SYSTEM STUDY

2.1. EXISTING SYSTEM

The existing system for managing a clothing business primarily relies on traditional brick-and-mortar models and basic digital platforms with limited functionalities. Many businesses use manual processes or basic websites that lack advanced e-commerce features. Managing product inventory, order processing, customer communication, and promotional campaigns manually or through outdated systems results in inefficiencies and missed opportunities.

2.2. DISADVANTAGES OF EXISTING SYSTEM

- **Time-Consuming Processes** – Manually updating inventory, processing orders, and handling payments consumes significant time and effort.
- **Limited Scalability** – Traditional systems cannot effectively handle an increasing number of products, orders, and customers as the business grows.
- **Lack of Centralized Data** – Product details, order history, and customer information are often scattered, making data retrieval and analysis difficult.
- **Inventory Mismanagement** – Without an automated system, tracking inventory levels accurately can lead to stockouts or overstocking.
- **Inefficient Order Management** – Manual processing increases the risk of delays, errors, and missed orders, resulting in poor customer satisfaction.
- **Limited Customer Engagement** – Traditional platforms offer limited features for personalized product recommendations, promotions, and communication with customers.

2.3. PROPOSED SYSTEM

The proposed **Clothing E-Commerce Website** is a comprehensive digital platform designed to streamline and automate the management of online retail operations. It will replace the manual, error-prone processes of the existing system with an efficient, centralized, and user-friendly solution.

2.4. ADVANTAGE OF PROPOSED SYSTEM

- **Time-Saving Automation** – Automates critical operations such as inventory updates, order processing, and payment handling, reducing manual workload.
- **Scalability and Flexibility** – Supports business growth by accommodating an expanding product range and increasing customer base.
- **Centralized Data Management** – Stores all product, customer, and order data in a unified system, enabling easy retrieval and analysis.
- **Efficient Inventory Control** – Ensures accurate tracking of stock levels, preventing stockouts and minimizing overstocking.
- **Seamless Order Management** – Streamlines order fulfillment and reduces errors, ensuring faster delivery and higher customer satisfaction.
- **Enhanced Customer Engagement** – Provides personalized recommendations, automated notifications, and seamless communication to improve customer retention.

2.5. PROBLEM DEFINITION AND DESCRIPTION

Managing a clothing business manually or through outdated systems is inefficient, time-consuming, and prone to errors. Product listings, customer information, and order records are often managed through spreadsheets or simple databases, leading to data duplication, inaccuracies, and mismanagement. Inventory tracking is prone to errors, resulting in stock discrepancies and poor order fulfillment. Manual order processing delays order completion, leading to dissatisfied customers. Lack of personalized customer engagement and real-time order tracking further affects customer retention. Financial transactions are often managed manually, posing security risks and increasing the chance of payment errors. Therefore, an automated **Clothing E-Commerce Website** is required to enhance efficiency, accuracy, and overall business operations.

3. SYSTEM ANALYSIS

3.1. PACKAGES SELECTED

- Front End : HTML,CSS,JAVASCRIPT,BOOTSTRAP
- Back End : PHP,MYSQL

3.2. RESOURCE REQUIRED

Hardware Resource

- Processor : Intel processor 3.0 GHz
- RAM : 8GB
- Hard disk : 500 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard
- Mouse : Logitech mouse
- Monitor : 15 inch color monitor

Software Resource

- Front End : HTML,CSS,JAVASCRIPT,BOOTSTRAP
- Back End : PHP,MYSQL
- Server : WAMP
- Operating System : Windows OS
- System type : 32-bit or 64-bit Operating System

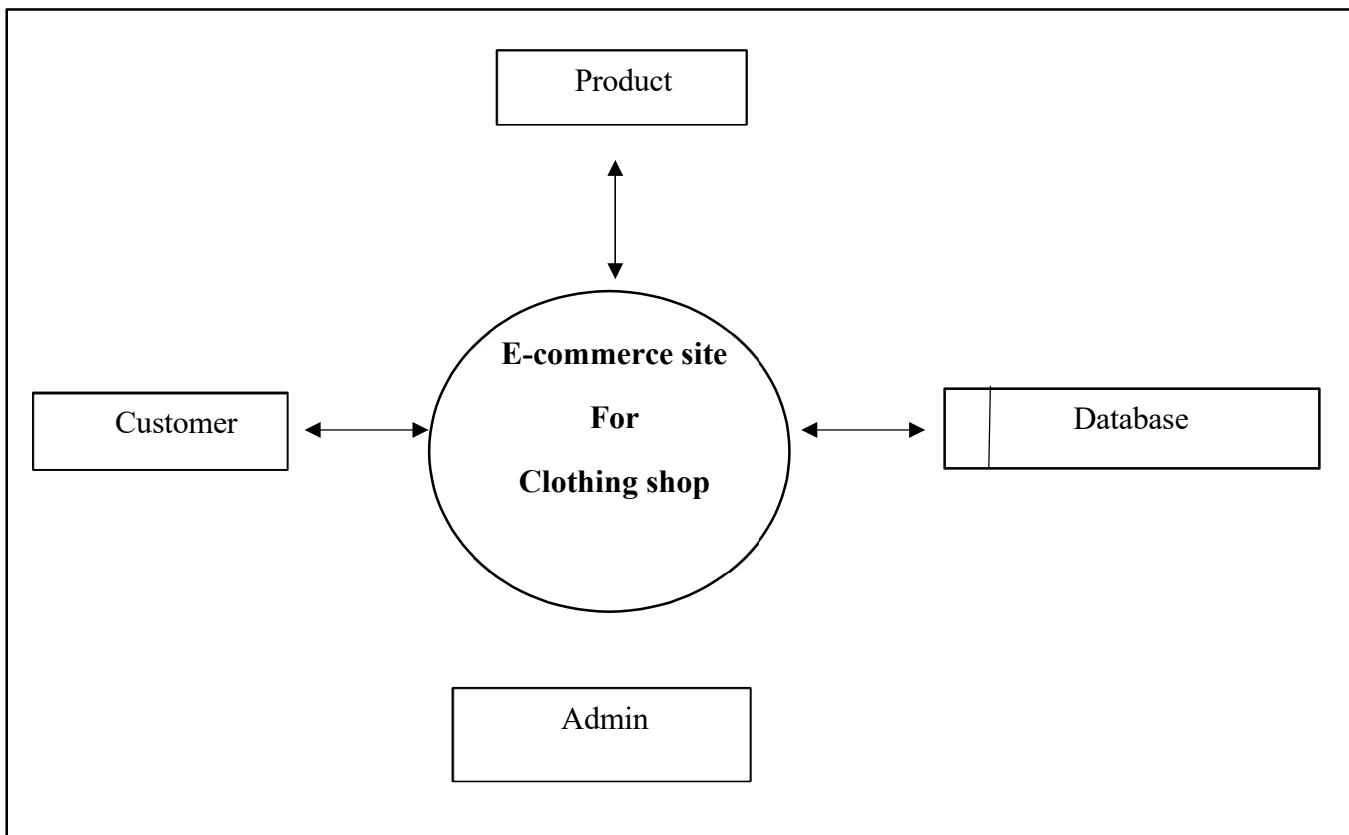
3.3. DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) visually represents the flow of data within a system. It illustrates how data moves between processes, data stores, and external entities. DFDs use standardized symbols such as circles (processes), arrows (data flow), and rectangles (external entities). They are hierarchical, starting with a high-level overview (Level 0) and expanding into detailed views (Level 1, 2, etc.). DFDs are useful for understanding system functionality, identifying data sources, and improving system design.

SYMBOL	DESCRIPTION
	An entity. A source of data or a destination for data.
	A process or task that is performed by the system.
	A data store, a place where data is held between processes.
	A data flow.

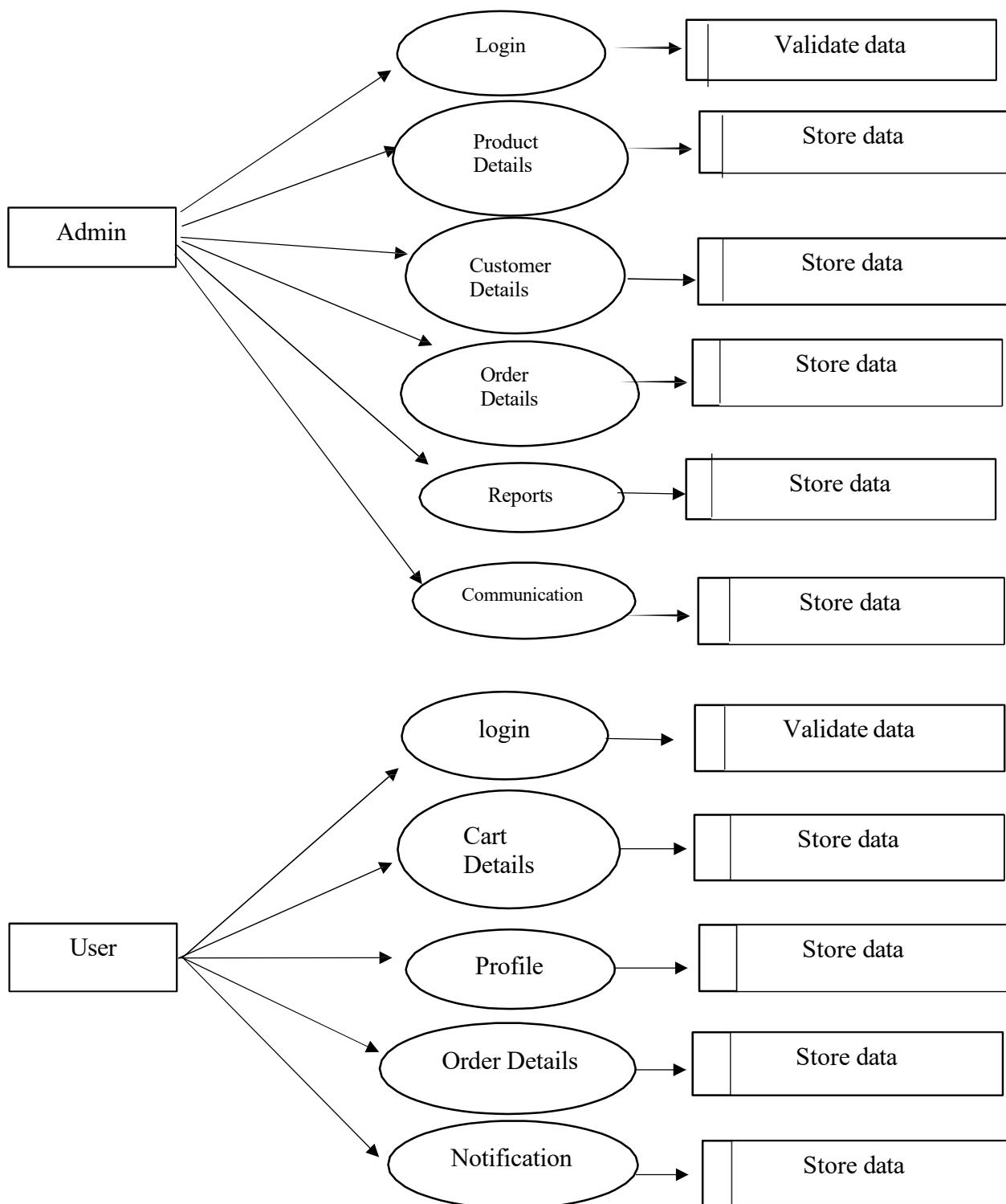
3.3.1. LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



3.3.2. LEVEL 1

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.



4. SYSTEM DESIGN

4.1. ARCHITECTURAL DESIGN

The **system architecture** of a clothing e-commerce website defines the conceptual framework that describes the structure, behavior, and interactions between various system components. It serves as a blueprint that guides the development of the platform to ensure seamless functionality, security, and scalability. The system architecture defines the interactions between front-end, back-end, database, and external services such as payment gateways and APIs. A well-structured architecture ensures that all system modules work together to deliver a smooth and secure user experience.

4.1.1. Various organizations define systems architecture in different ways, including:

- **Physical and Logical Arrangement:** A well-defined system architecture for an e-commerce website includes both physical infrastructure (servers, hosting, and databases) and logical structures (modules for product management, order processing, and payment systems). This arrangement provides a scalable and robust solution for meeting customer and business requirements.
- **Strategic Design Decisions:** The architecture includes essential design decisions related to front-end interfaces, back-end services, and database management. These decisions affect overall performance, security, and maintainability of the system.
- **Hardware, Software, and Network Specifications:** The system architecture documents current hardware and software capabilities, outlines plans for future upgrades, and describes network requirements for seamless data exchange.
- **Product and Process Lifecycle Management:** The architecture integrates design models for various modules, ensuring a cohesive system that supports business growth, data security, and continuous process improvements.

4.2. I/O FORM DESIGN

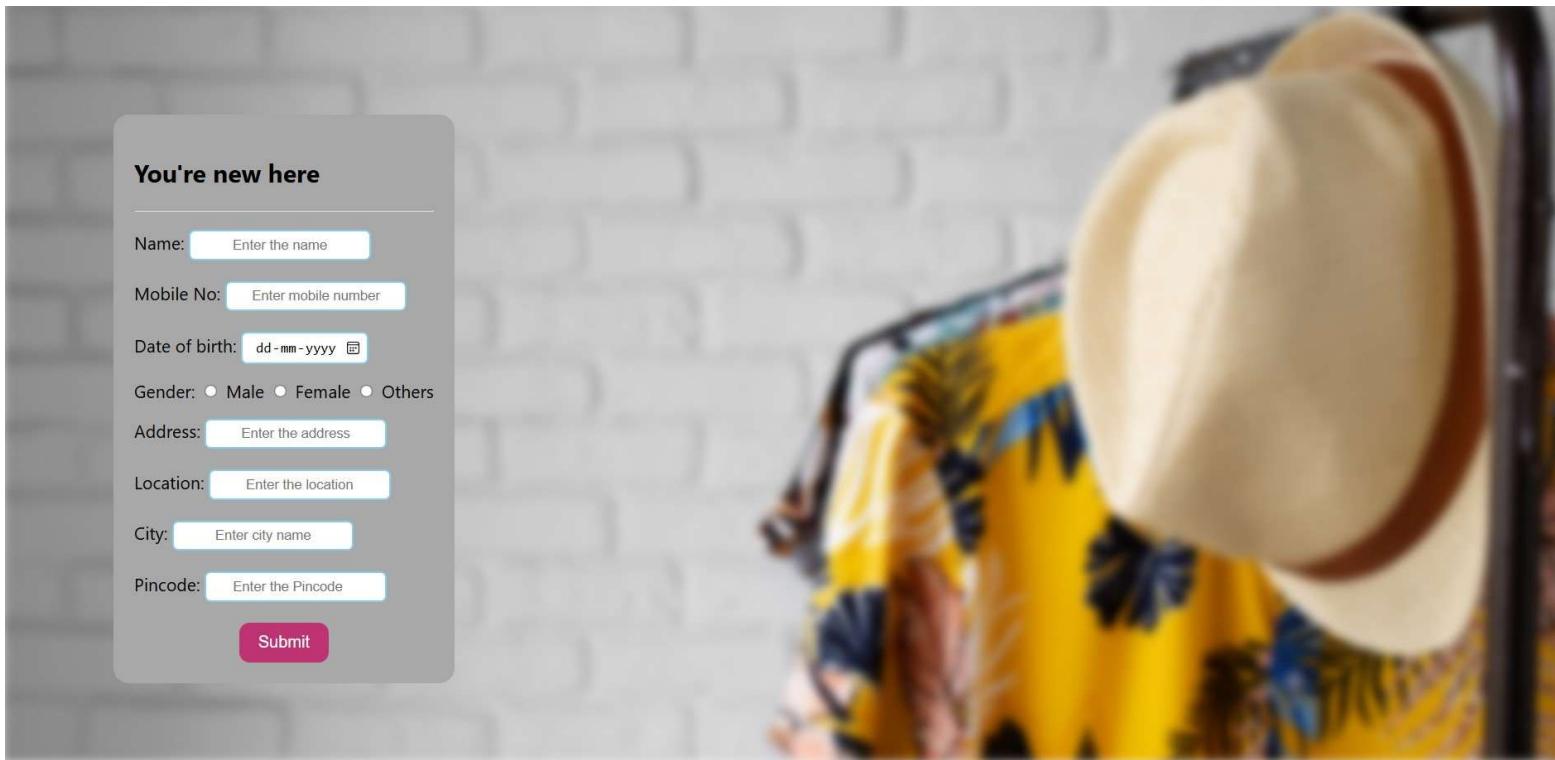
4.2.1 INDEX PAGE



4.2.2. ADD LOGIN FORM



4.2.3 ADD SIGN-IN FORM



4.3. TABLES

A table is a data structure that organizes information into rows and columns. It can be used to both store and display data in a structured format. For example, databases store data in tables so that information can be quickly accessed from specific rows. Websites often use tables to display multiple rows of data on page. Spreadsheets combine both purposes of a table by storing and displaying data in a structured format.

Databases often contain multiple tables, with each one designed for a specific purpose. For example, a company database may contain separate tables for employees, clients, and suppliers. Each table may include its own set of fields, based on what data the table needs to store. In database tables, each field is considered a column, while each entry (or record), is considered a row. A specific value can be accessed from the table by requesting data from an individual column and row.

Table 1 : **Admin**

Field	Type	Default
Name	VARCHAR(20)	NOT NULL
Pass_key	VARCHAR(20)	NULL
Secure key	VARCHAR(20)	NULL
Full_name	VARCHAR(50)	NOT NULL

Table 2: **Login**

Field	Type	Default
Mobile Number	INT	PRIMARY
Password	VARCHAR(20)	NOT NULL

Table 3 : **Product**

Field	Type	Default
Product name	VARCHAR(255)	NOT NULL
Price	INT	NOT NULL
Quantity	INT	NOT NULL
Size	VARCHAR(255)	NOT NULL

Table 4: **Order**

Field	Type	Default
Product Details	VARCHAR(255)	AUTO_INCREMENT
Customer Details	VARCHAR(255)	NOT NULL
Order Date	DATE	NOT NULL
Order Price	FLOAT	CURRENT_TIMESTAMP

Table 5 : **Cart**

Field	Type	Default
Product	INT	AUTO_INCREMENT
Size	VARCHAR(5)	NULL
Quantity	INT	NULL

Table 6 : **Sign-in**

Field	Type	Default
Mobile Number	INT	PRIMARY
Name	VARCHAR(255)	NOT NULL
DOB	DATE	NOT NULL
Address	VARCHAR(20)	NULL
Location	VARCHAR(20)	NULL
Pincode	INT	NULL
Password	VARCHAR(20)	NULL

4.4. NORMALIZATION

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy (repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

- Eliminating redundant (useless) data.
- Ensuring data dependencies make sense i.e data is logically stored.

4.4.1. NORMALIZATION RULE

Normalization rules are divided into the following normal forms:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form

4.4.2. FIRST NORMAL FORM (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single (atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Table 7 : Model of first normal form

ID	Name	Phone Number	DOB	Pin code
1	Sam	9003505907	11.12.2004	620101
2	Joe	9786158432	02.09.2006	620001

4.4.3. SECOND NORMAL FORM (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Table 8 : Model of second normal form

ID	Phone Number
1	9003505907
2	9786158432

4.4.4. THIRD NORMAL FORM (3NF)

A table is said to be in the Third Normal Form when,

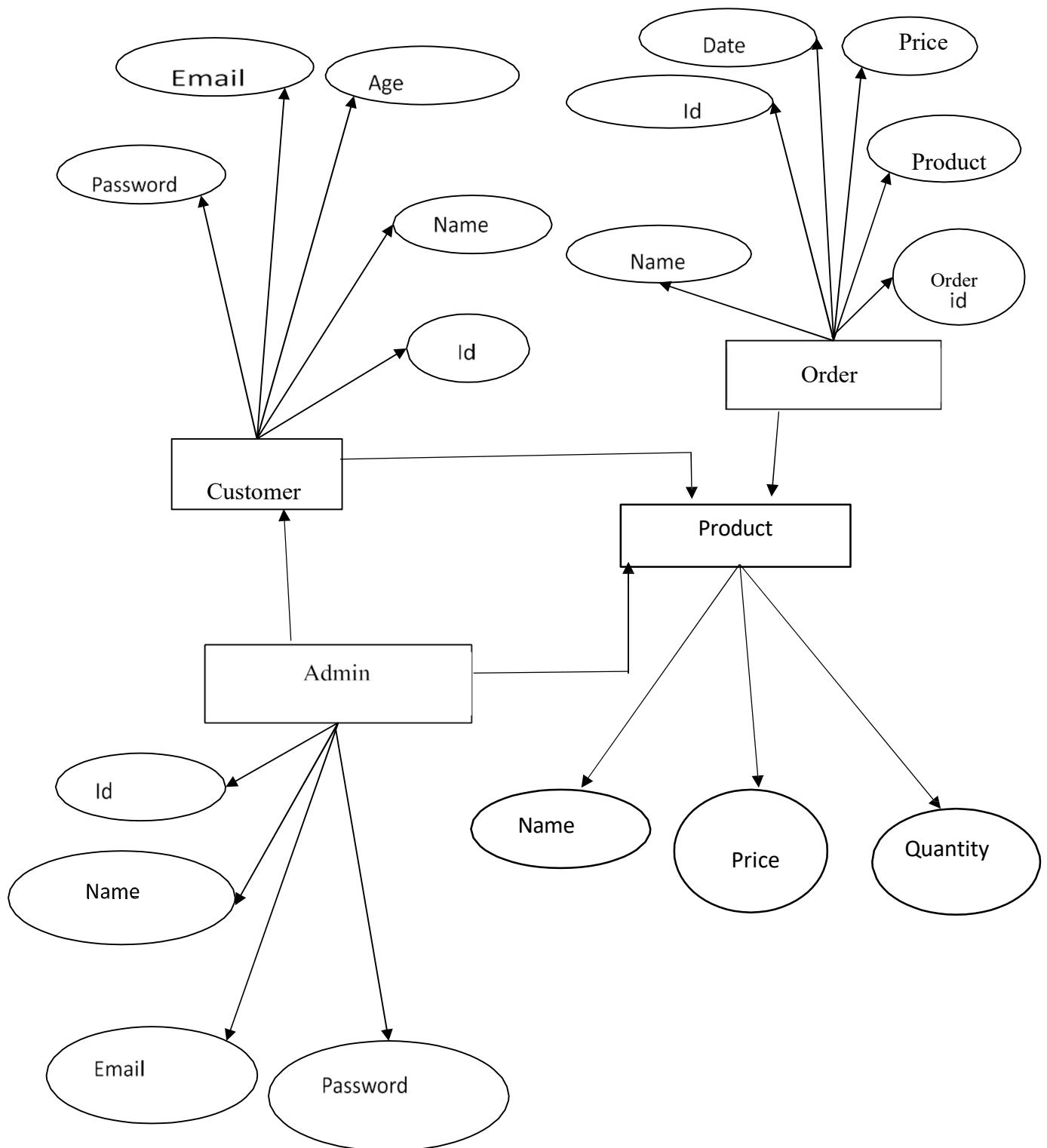
1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

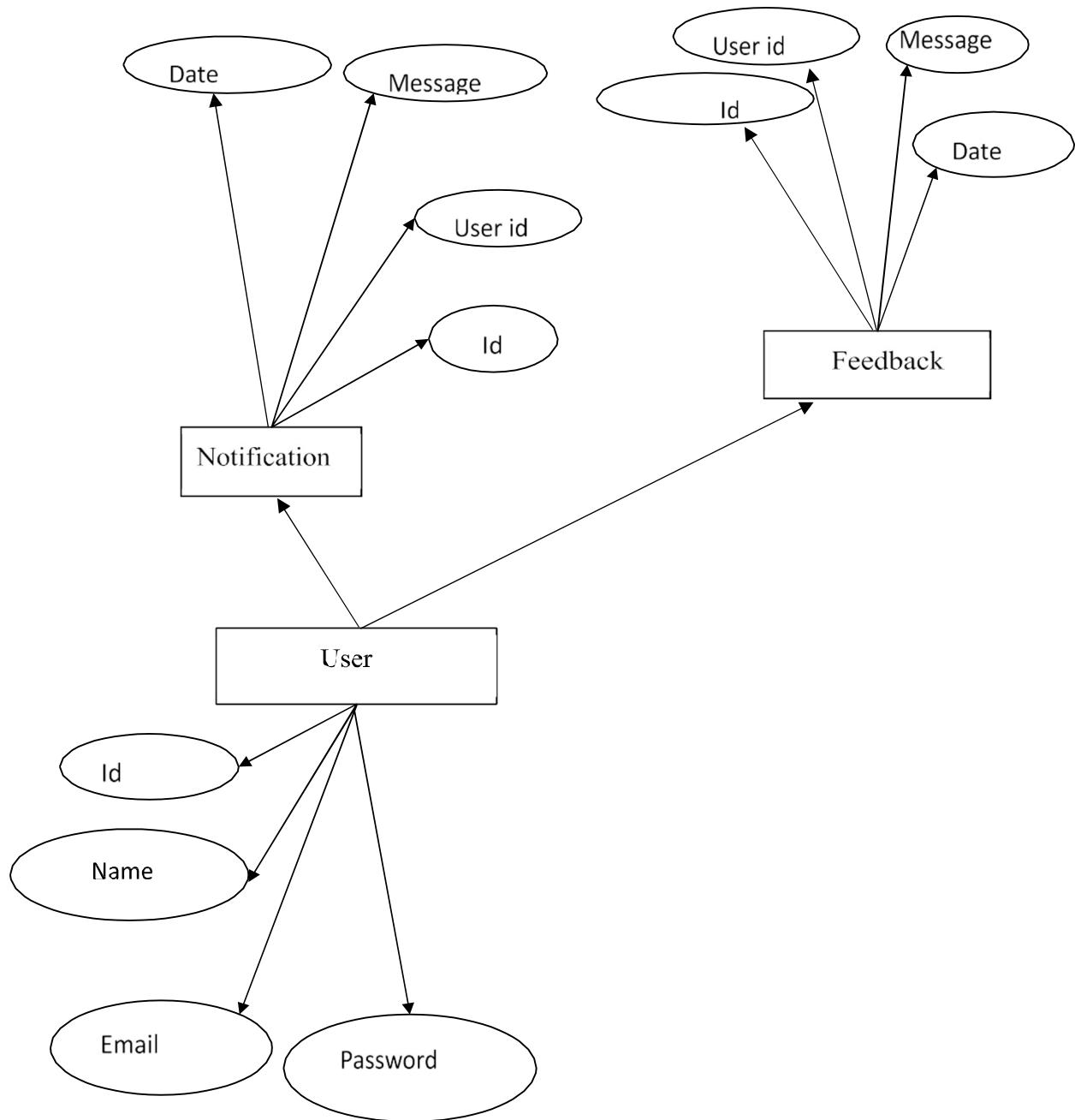
Table 9 : Model of Third Normal Form

Product	Name	Price
101	Jean pant	899
102	Party combo	1299

4.5. ENTITY RELATIONSHIP DIAGRAM

An Entity-Relationship Diagram (ERD) is a visual representation of data structures used in database design. It depicts entities (objects), attributes (properties of entities), and relationships (connections between entities). Entities are shown as rectangles, attributes as ovals, and relationships as diamonds or lines. ERDs help visualize data flow and identify database requirements.





4.6. DATA DICTIONARY

A data dictionary is a structured collection of information about data elements in a database. It defines data types, structures, constraints, and relationships. Each entry typically includes the field name, data type, allowed values, and description. Data dictionaries ensure consistency, improve data understanding, and support database management. They are crucial for developers, analysts, and database administrators.

Table 10 : Model of data dictionary

Table Name	Attribute Name	Data Type	Constraints	Description
Admin	Admin ID	INT	PK, Not Null	Unique identifier for Admin
Admin	Name	VARCHAR(100)	Not Null	Admin's full name
Admin	Email	VARCHAR(100)	Unique, Not Null	Admin's email address
Admin	Password	VARCHAR(100)	Not Null	Admin's login password
User	Mobile number	INT	PK, Not Null	Unique identifier for User
User	Name	VARCHAR(100)	Not Null	User's full name
User	DOB	DATE	Unique, Not Null	User's email address
User	Password	VARCHAR(100)	Not Null	User's login password
User	Address	VARCHAR(50)	Not Null	User's address
Product	price	FLOAT	PK, Not Null	Product price
Product	Name	VARCHAR(100)	Not Null	Product name
Product	Quantity	INT	Not Null	Quantity
Product	size	VARCHAR(50)	Not Null	Available size
Order	Product details	VARCHAR(50)	PK, Not Null	Multiple Product details
Order	Customer details	VARCHAR(50)	Not Null	Customer details
Order	Order price	FLOAT	Not Null	Total Bill price

5. SYSTEM DEVELOPMENT

5.1. FUNCTIONAL DOCUMENTATION MODULES

1. ADMIN MODULES:

- Login
- Product Details
- Customer Details
- Order Details
- Reports and Analytics
- Communication Module

2. USER MODULES:

- Login
- Cart Details
- Order Details
- Profile Management
- Offers and Discounts
- Notification

MODULE DESCRIPTION

Admin Modules

1. Login
 - Secure authentication for admins to access the system.
 - Features like password recovery, two-factor authentication (optional).
2. Product Details
 - Add, edit, or remove product details.
 - Maintain product details such as name, brand, original price and discounted price.
3. Match Scheduling
 - Collect and store the Customer details.
 - Maintain customer details such as name, mobile number and address,.

4. Order Details

- Collect the order details with customer id.
- Maintain and provide the order with order id.

5. Reports & Analytics

- Generate detailed reports on product details, monthly sales and profit stats.
- Visual data insights for better decision-making.

6. Communication Module

- Send notifications, offers and updates to products.
- Manage feedback and inquiries from users.

User Modules

1. Login

- Secure authentication for Clothing site.

2. Cart Details

- Re-store and display the users cart details.

3. Order Details

- Allows users to track the order details and update the delivery process.

4. Profile Management

- Enables users to view and update their personal details.

5. Offers

- Update the user about the brand offers, seasonal offers and festival sales.

6. Notification & Feedback

- Users receive product notification , share the purchase experience, and can provide feedback to the admin.

5.2. SPECIAL FEATURES OF THE LANGUAGE

5.2.1. FRONTEND

❖ HTML:

HTML, which stands for HyperText Markup Language, is the standard markup language used to create and structure content on the web. It is the backbone of web development and provides a way to describe the structure of web pages using markup. HTML consists of a set of elements, each represented by tags, which define the various parts and components of a webpage. Tags are used to define elements in HTML. They consist of an opening tag, content, and a closing tag. HTML documents are built using a variety of markup elements, such as headings, paragraphs, lists, images, links, forms, and more.

❖ CSS:

CSS, which stands for Cascading Style Sheets, is a style sheet language used to describe the presentation and formatting of a document written in HTML or XML. CSS allows web developers to control the visual appearance of web pages, making it a crucial component in the design and layout of websites. By separating content (HTML) from presentation (CSS), developers can achieve greater flexibility and maintainability in web development.

HTML provides the structure and foundation for web content. When combined with Cascading Style Sheets (CSS) for styling and JavaScript for interactivity, it forms a powerful trio of technologies used in modern web development. Web browsers interpret HTML documents to render the visual representation of web pages. CSS provides tools for creating complex layouts and positioning elements on a page. Techniques include floats, flexbox, and grid layout..CSS supports responsive design principles, allowing developers to create websites that adapt to different screen sizes and devices.

CSS is an essential technology for web development, providing the means to create visually appealing, responsive, and interactive websites. When combined with HTML and JavaScript, CSS forms the basis for the three core technologies used in front-end web development.

❖ **JAVA SCRIPT:**

JavaScript is a high-level, interpreted programming language that is widely used for creating dynamic and interactive web pages. It is a versatile scripting language that can be embedded directly into HTML code and executed by web browsers to enhance the functionality and behavior of websites. JavaScript allows developers to create features such as form validation, content updates, and interactive user interfaces.

❖ **BOOTSTRAP:**

Bootstrap is a popular front-end framework for building responsive and mobile-first websites. It provides pre-designed [CSS](#), [JavaScript](#) components, and utility classes to quickly create modern and consistent user interfaces. It includes pre-built responsive grid systems for mobile-first design. Offers a wide range of UI components like buttons, modals, and navbars. It provides built-in support for responsive typography, spacing, and utilities. Extensively customizable via Sass variables and Bootstrap's configuration.

❖ **Advantage of bootstrap**

- Automatically adjusts layout for mobile, tablet, and desktop screens using a grid system.
- Provides ready-to-use **buttons, modals, alerts, navigation bars, cards, and more**, saving development time.

5.2.2. BACKEND

❖ **PHP**

PHP (Hypertext Preprocessor) is a powerful, open-source server-side scripting language widely used for web development. Known for its simplicity and flexibility, PHP allows developers to embed code directly into HTML, making it easy to build dynamic and interactive web applications. It supports multiple databases like MySQL, PostgreSQL, and MongoDB, enabling seamless data management. PHP's extensive library support simplifies tasks such as file handling, image manipulation, and encryption. With features like error handling through try-catch blocks, robust session management,

and built-in security mechanisms to combat SQL injection and XSS attacks, PHP ensures secure web development.

Its compatibility across platforms like Windows, Linux, and macOS makes it versatile, while popular frameworks such as Laravel, CodeIgniter, and Symfony accelerate development with structured coding practices. PHP also offers excellent API integration capabilities, making it ideal for connecting external services. With features like email handling via the mail() function, command-line scripting for automation, and strong object-oriented programming support, PHP is a comprehensive solution for web development projects.

❖ **Advantage of PHP**

- Easy to Learn and Use: PHP has a simple syntax, making it beginner-friendly and easy to integrate with HTML.
- Open-Source and Cost-Effective: PHP is free to use, reducing development costs significantly.
- Cross-Platform Compatibility: PHP runs seamlessly on various operating systems like Windows, Linux, and macOS.
- Extensive Database Support: PHP efficiently connects with databases like MySQL, PostgreSQL, and MongoDB for dynamic web applications.
- Robust Frameworks: PHP offers powerful frameworks such as Laravel, CodeIgniter, and Symfony, which speed up development and ensure better code organization.
- Strong Community Support: PHP has a large, active community that provides extensive resources, documentation, and troubleshooting assistance.

5.2.3. DATABASE

❖ **MySQL**

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

INTER IMAGES

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

GRAPHICAL

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development(replacing MySQL Query Browser) and Database administration(replacing MySQL Administrator).

COMMAND LINE

MySQL ships with some command line tools. Third-parties have also developed tools to manage a MySQL server, some listed below. Maatkit - a cross-platform toolkit for MySQL, PostgreSQL and Memcached, developed in Perl. Maatkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Maatkit is included with several GNU/Linux distributions such as CentOS and Debian and packages are available for Programming.

MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO OpenServer, SCO

UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.^[32]

MySQL is written in C and C++. Its SQL parser is written in yacc, and a home-brewed lexical analyzer. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBC interimage called MyODBC allows additional programming languages that support the ODBC interimage to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL - URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

FEATURES

As of April 2009, MySQL offered MySQL 5.1 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server. MySQL 5.5 is offered under the same licenses. They have a common code base and include the following features:

- ❖ A broad subset of ANSI SQL 99, as well as extensions
- ❖ Cross-platform support
- ❖ Stored procedures
- ❖ Triggers
- ❖ Cursors
- ❖ Updatable Views
- ❖ Information schema
- ❖ Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)
- ❖ X/Open XAdistributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine

- ❖ Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
- ❖ Transactions with the InnoDB, and Cluster storage engines; savepoints with InnoDB
- ❖ SSL support
- ❖ Query caching
- ❖ Sub-SELECTs (i.e. nested SELECTs)
- ❖ Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master, no automatic support for multiple masters per slave.
- ❖ Full-text indexing and searching using MyISAM engine
- ❖ Embedded database library
- ❖ Unicode support (however prior to 5.5.3 UTF-8 and UCS-2 encoded strings are limited to the BMP, in 5.5.3 and later use utf8mb4 for full Unicode support)
- ❖ ACID compliance when using transaction capable storage engines (InnoDB and Cluster)

Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time): Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, Cluster, EXAMPLE, Maria, and InnoDB, which was made the default as of 5.5). Partner-developed storage engines (solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (formerly Brighthouse), Kickfire, XtraDB, IBM DB2). InnoDB used to be a partner-developed storage engine, but with recent acquisitions, Oracle now owns both MySQL core and InnoDB.

6. TESTING

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system. The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input.

6.1. TYPES OF TESTING DONE

6.1.1. SYSTEM TESTING

After a system has been verified, it needs to be thoroughly tested to ensure that every component of the system is performing in accordance with the specific requirements and that it is operating as it should including when the wrong functions are requested or the wrong data is introduced.

Testing measures consist of developing a set of test criteria either for the entire system or for specific hardware, software and communications components. For an important and sensitive system such as an electronic voting system, a structured system testing program may be established to ensure that all aspects of the system are thoroughly tested.

6.1.2. UNIT TESTING

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.

6.1.3. INTEGRATION TESTING

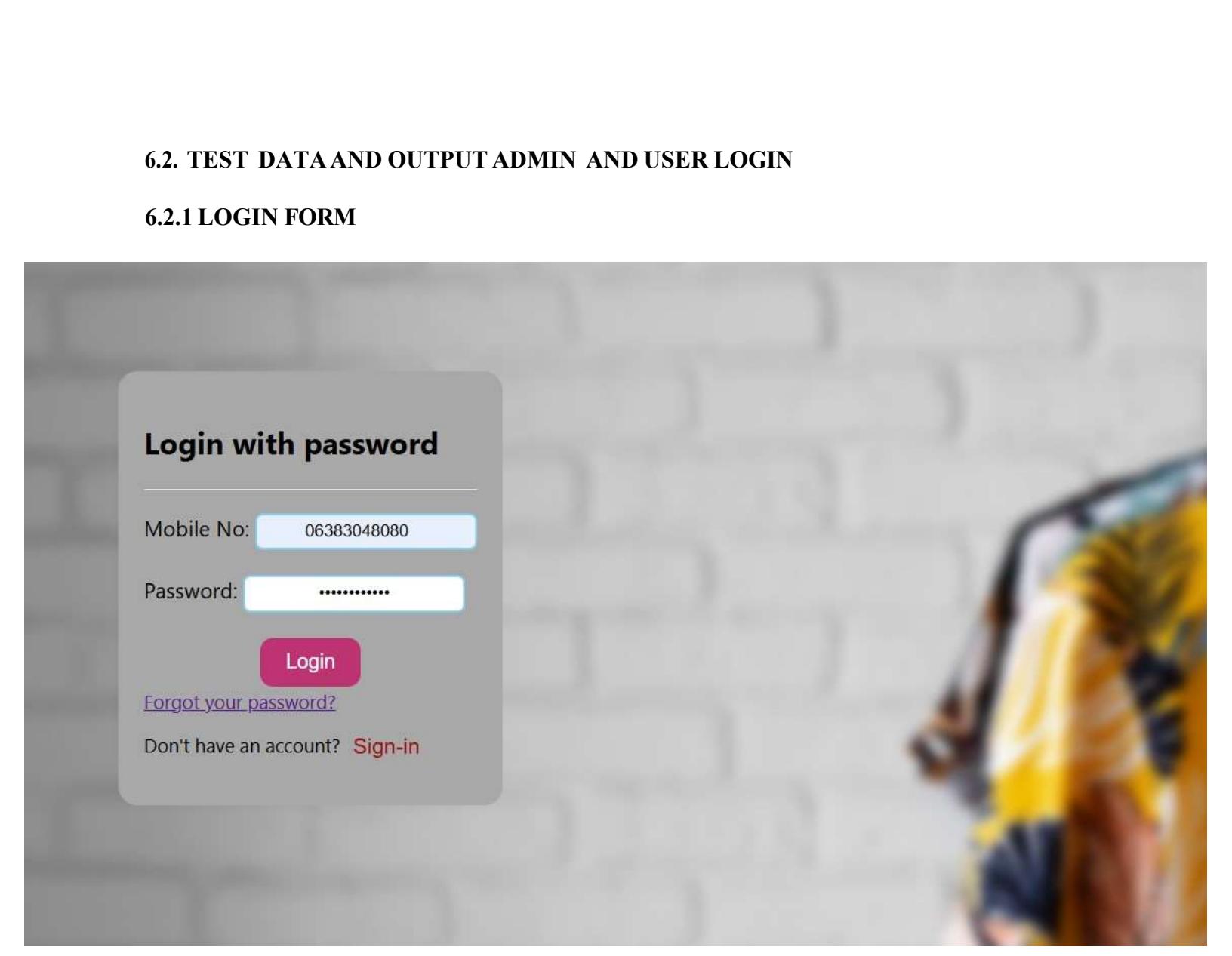
Testing in which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.

6.1.4. VALIDATION TESTING

Valid and invalid data should be created and the program should be made to process this data to catch errors. When the user of each module wants to enter into the page by the login page using the use rid and password .If the user gives the wrong password or use rid then the information is provided to the user like “you must enter user id and password”. Here the inputs given by the user are validated. That is password validation, format of date are correct, textbox validation. Changes that need to be done after result of this testing.

6.2. TEST DATA AND OUTPUT ADMIN AND USER LOGIN

6.2.1 LOGIN FORM



Login with password

Mobile No:

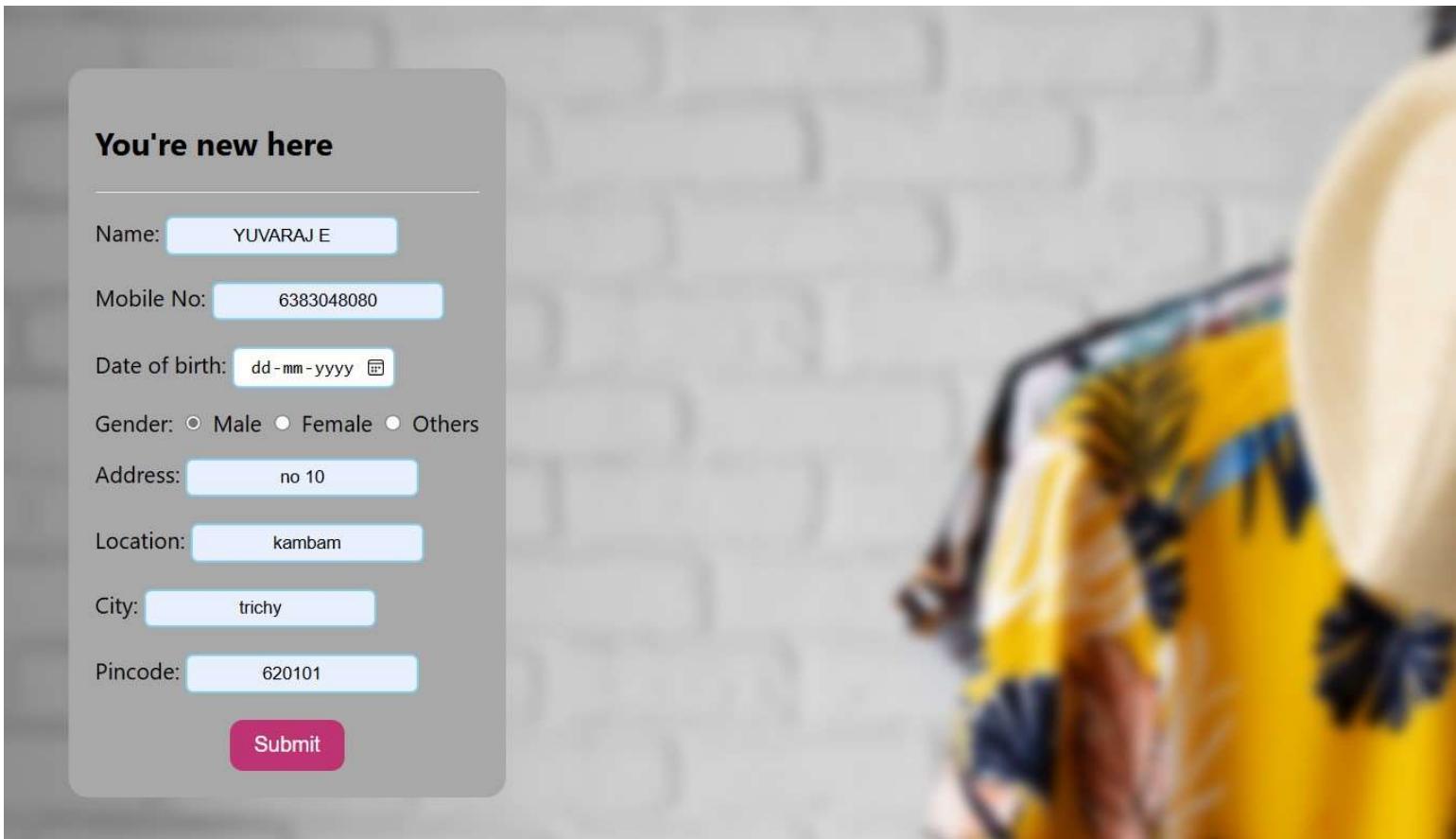
Password:

Login

[Forgot your password?](#)

Don't have an account? [Sign-in](#)

6.2.2. SIGN-IN LOGIN



7. USER MANUAL

The **Cricket Club Management System** is a comprehensive web-based platform designed to efficiently manage various aspects of a cricket club. This system caters to both **Admin** and **User** roles, ensuring seamless communication and organized club activities. The platform is built using **HTML, CSS, and JavaScript** for the front-end interface, while the back-end functionalities are powered by **PHP** to ensure dynamic content management and secure data handling.

7.1. HARDWARE REQUIREMENTS

- Processor : Intel processor 3.0 GHz
- RAM : 8GB
- Hard disk : 500 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard
- Mouse : Logitech mouse
- Monitor : 15 inch color monitor

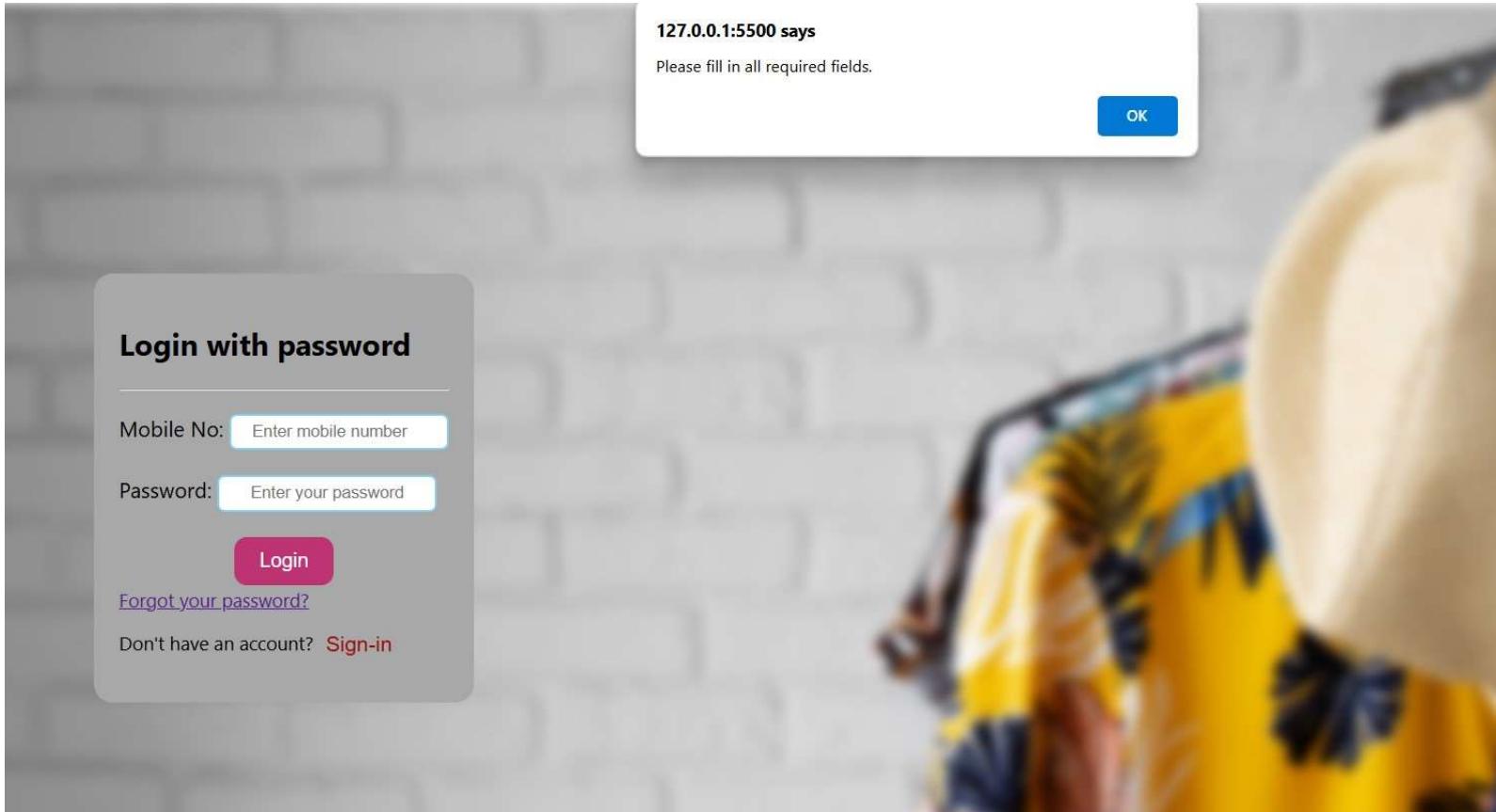
7.2. SOFTWARE SPECIFICATION

- Front End : HTML,CSS,JAVASCRIPT,BOOTSTRAP
- Back End : PHP,MYSQL
- Server : WAMP
- Operating System : Windows OS
- System type : 32-bit or 64-bit Operating System

7.3. INSTALLATION PROCEDURE

- Downloading Wamp Server Download the installer file for the latest version of Wamp Server, and save the file to your computer.
- Installing Wamp Server To start the installation process, you need to open the folder where you saved the file, and double-click the installer file. A security warning window will open, asking if you are sure you want to run this file. Click Run to start the installation process.

7.4 ERROR MESSAGE



8. CONCLUSION

8.1. SUMMARY OF PROJECT

The project "Design and Development of a Clothing E-Commerce Website" represents a significant advancement in enhancing the online shopping experience for customers and streamlining business operations for sellers. The implemented system replaces traditional retail limitations with an automated and efficient digital marketplace. This transition has not only improved the overall efficiency of online shopping but has also enhanced transparency, communication, and accessibility. Customers now have access to a wide range of products, secure payment options, real-time order tracking, and personalized recommendations, empowering them to make informed purchasing decisions. The integration of analytics and reporting tools equips sellers with valuable insights, enabling data-driven decisions for better inventory management, pricing strategies, and marketing efforts. The seamless interaction between sellers, administrators, and delivery personnel ensures a smooth and efficient order fulfillment process, reducing delays and enhancing customer satisfaction.

8.2. FUTURE ENHANCEMENTS

Looking forward, the project "Design and Development of a Clothing E-Commerce Website" has promising avenues for future enhancement. Integrating machine learning can offer personalized product recommendations based on customer behavior. Incorporating a chatbot for instant customer support and augmented reality for virtual try-ons are exciting possibilities. Blockchain integration can enhance transaction security and authenticity verification for premium brands. Expanding analytics provides deeper insights into customer preferences, and a dedicated mobile app can improve accessibility and user engagement. Gamified loyalty programs can boost customer retention, while advanced communication tools could enhance seller-buyer interactions. Strengthening partnerships with fashion brands and collaborating with influencers could expand the platform's reach. Managing inventory with AI-driven demand forecasting and establishing a continuous feedback loop for product reviews could further refine the system, ensuring its continual growth and adaptability in the evolving e-commerce landscape.

BIBLIOGRAPHY

BOOK REFERENCE

1. "E-Commerce Strategy, Technology, and Implementation" by Gary Schneider
2. "Fashion Marketing and E-Commerce" by Margaret Bruce and Christopher Moore
3. "Digital Retailing: Strategies for Omni-Channel Success" by Paul Doherty
4. "The Art of Digital Marketing" by Ian Dodson
5. "Mastering Shopify: Build and Customize Your Online Store" by Thomas F. Sheehan

WEBSITE REFERENCE

6. Shopify - Official website for building and managing online stores: shopify.com
7. WooCommerce - E-Commerce platform for WordPress websites: woocommerce.com
8. BigCommerce - Leading platform for enterprise-level online stores: bigcommerce.com