

BOOK STORE



Submitted for fulfilment of the requirement of **Main Project(4MCA1)** for

MASTER OF COMPUTER APPLICATIONS

Submitted by

KUNDAN KUMAR (P18CD23S126023)

Name of the Guide

Dr. SURENDRA H D

Associate Professor



COMMUNITY INSTITUTE OF MANAGEMENT STUDIES

MCA DEPARTMENT

Affiliated to Bengaluru City University

9th Cross, 9th Main, 2nd Block, Jayanagar, Bangalore-560011

COMMUNITY INSTITUTE OF MANAGEMENT STUDIES



CERTIFICATE

This is to certify that the report entitled "**BOOK STORE**" embodies the original work by **KUNDAN KUMAR(P18CD23S126023)** the fulfilment of the requirements for Main Project (**4MCA1**) for MCA, 4TH Semester course during the academic semester from July 2025 to November 2025 as prescribed by Bengaluru City University.

Project Guide

(Dr. SURENDRA H D)

Head of the Department

(Dr. MANJULA P S)

Place: Bangalore

Date:

COMMUNITY INSTITUTE OF MANAGEMENT STUDIES



CERTIFICATE

This is to certify that the report entitled "**BOOK STORE**" embodies the original work by **KUNDAN KUMAR(P18CD23S126023)** the fulfilment of the requirements for Mini Project (**4MCA1**) for MCA, 4TH Semester course during the academic semester from July 2025 to November 2025 as prescribed by Bengaluru City University.

External Examiner

1. _____

2. _____

DECLARATION

I, **KUNDAN KUMAR** do hereby declare that the project work entitled “**BOOK STORE**” is a Bonafide work carried out by me under the guidance of Assistant Professor **Dr. SURENDRA H D**. This project as presented in this report is our original work and has not been presented for any other University award. This project has been submitted as partial fulfilment of requirement for the Degree of Master of computer Applications IV Semester of Bengaluru City University.

KUNDAN KUMAR

Acknowledgement

I would like to express our gratitude and appreciation to all those who gave us the possibility to complete this report.

We take pleasure in expressing gratitude towards our institution for providing us the opportunity to work on this project as a part of our course. We would like to express our gratitude and sincere thanks to our management, we would like to express our gratitude to Principal & Director. **Dr. Janardhan T G** for providing the right atmosphere at the institute.

We express our heart full thanks to **Dr. Manjula P S**, Head of the Department, for being helpful through the project. We express our deep sense of gratitude to our project guide **Dr. SURENDRA H D**, Assistant Professor for her unfailing encouragement, valuable guidance, and suggestions to us during project work. We also express our heartfelt thanks to our parents for being supportive in all our activities without whom it would not have been possible for us to complete our project.

INDEX

SI NO	CONTENTS	PG NO
1.	Synopsis	1 - 3
2.	Introduction	4 - 9
3.	Literature Survey	10 - 13
4.	Hardware And Software Requirements	14 - 15
5.	System Design	16 - 28
6.	Database Design	29 - 34
7.	System Testing	35 - 37
8.	Implementation Details	38 - 41
9.	Coding	42 - 98
10.	Snapshots	99 - 106
11.	Developments Tools	107 - 110
12.	Testing	111 - 112
13.	System Evaluation and Discussion	113 - 114
14.	System Strength and Limitation	115 - 116
15.	Conclusion	117 - 118
16.	Future Enhancement	119 - 120
17.	Bibliography	-----

CHAPTER - 1

SYNOPSIS

SYNOPSIS

Title: BOOK STORE

Aim:

To develop an Online Book Store system for efficient book management, online browsing and purchasing, and seamless administration. The system provides secure accounts, order tracking, and Gmail notifications for registration, payments, and delivery updates, ensuring a smooth experience for both customers and administrators.

Description:

The Book Store Management System is a web-based application that provides an interactive platform for both customers and administrators. Customers can register, browse available books, view details, place orders, and make payments online. Administrators can manage book records, update inventory, track orders, and monitor deliveries. The system ensures easy access, efficient management, secure transactions, and a seamless user experience for all users.

Features

Customer Module

- User registration and login.
- Browsing books by category, author, and price
- Viewing book details and reviews
- Adding desired books to the cart
- Placing online orders with multiple payment options (Card, UPI, etc.)
- Viewing order history and tracking current orders.

Admin Module

- Managing the book catalog (add, update, delete books).
- Monitoring current stock and inventory levels.
- Processing customer orders and updating order statuses.
- Viewing transaction history.
- Track completed and pending orders.
- Managing user accounts and feedback.

Delivery Module

- Assign delivery agents to orders.
- Real-time tracking of deliveries.
- Update delivery status (pending, out for delivery, delivered).

Hardware Requirements:

- Processor: 11th Gen Intel(R) Core (TM) i3-1115G4
- RAM: 8GB
- Hard Disk: 500GB
- Internet Connection: Required for online operations

Software Requirements:

- Operating System: Windows 11
- Database: MySQL
- Backend: PHP, MySQL
- Frontend: HTML, CSS, JavaScript
- Server: Apache, Xampp

CHAPTER - 2

INTRODUCTION

2. INTRODUCTION

The Bookstore Management System is a web-based platform that allows customers to browse, search, and purchase books online securely. Administrators can manage book listings, inventory, and orders efficiently. The system includes an automated Gmail notification feature to inform users about registration, payments, and delivery updates. Built on an open-source platform, it ensures scalability, transparency, and a seamless experience for both customers and administrators.

2.1 Objective

The main objective of this project is to develop an online application that allows bookstore owners to manage their inventory and sales efficiently while giving customers an easy and secure way to browse and purchase books online. The system aims to enhance customer experience, increase sales, and reduce manual management tasks.

- **User-Friendly Interface:**

The **Bookstore Management System** offers a simple, intuitive, and user-friendly interface. Customers can easily browse books by category, author, or price range, view details, and make purchases securely.

- **Order Reporting:**

The system allows administrators to generate sales and order reports for any specific date or time period. This helps track order statuses, customer preferences, and business performance efficiently.

- **EmailNotificationSystem:**

The system is integrated with an automated Gmail notification feature that keeps users informed at every stage. Customers receive instant email alerts upon registration, order placement, payment confirmation, and delivery updates. Administrators also receive notifications for new orders and user registrations, ensuring smooth communication between the system and its users.

- **Minimal Paperwork:**

The system reduces paperwork by maintaining all records digitally. Reports can be generated automatically, saving time and improving accuracy. Since all data is stored in the system database, manual errors are minimized.

- **Controlled by Authorized Staff:**

The application is managed by authorized staff and administrators. Data entry and management are straightforward, reducing the risk of clerical errors and ensuring tasks are completed accurately and on time.

2.2 Modules

The **Bookstore Management System** consists of the following main modules:

- Administrator Module
- Customer Module

2.2.1 Administrator Module

The administrator has full control over the platform and can view and manage all book and user information. The key responsibilities include:

- Creating book categories (e.g., Fiction, Science, Technology, etc.)
- Managing book categories
- Adding and updating book records (title, author, price, stock, etc.)
- Managing user orders.
- Tracking payment status and delivery updates
- Viewing reports and transaction history
- Receiving automated email notifications when new user registrations or customer orders are placed

2.2.2 Customer Module

This module provides functionality for end users (customers), allowing them to:

- Viewing available book items and categories
- Register for a new account and log in securely
- Search for books by title, author, or genre
- Add books to the shopping cart
- Place book orders and make online payments
- View order history and track their book deliveries
- Receive email notifications for registration confirmation, order placement, payment success, and delivery status updates

1.2.3 Admin Module

The Admin Module provides complete control over the Bookstore Management System. The functionalities include:

- **Dashboard:**

The admin can view all key statistics at a glance, including:

- Total Pending Orders
- Total Completed Orders
- Total Orders
- Total Books Added
- Total Registered User Accounts
- Total Admins
- New Messages/Feedback

Each section allows the admin to view more details or take necessary actions, such as updating profiles or managing orders. Admins also receive email notifications for new customer orders or important updates.

- **Registered Users:**

Admin can view and manage all registered users, including updating customer details

and viewing their order history.

- **Book Category:**

Admin can add, update, or delete book categories for better organization.

- **Book Inventory:**

Admin can manage book listings by adding new books, updating stock levels, or removing unavailable items..

- **Orders:**

Admin can view all book order details and update their status (Pending, Processing, Shipped, Delivered).

- **Search Order:**

Admin can search for specific orders using order numbers or customer details.

- **Reports:**

Admin can generate detailed reports, including:

- Order Details
- Total Number of Orders
- Sales Reports (based on selected date range)

Additionally, the admin can:

- Update their profile
- Change their password
- Recover a forgotten password

2.2.4 User Module

This module allows customers to manage their online shopping experience. Key functionalities include:

- **Book Catalog:**

Customers can browse and view available books from different categories.

- **My Accounts:**

Users can update their profile, change passwords, and securely log out.

An email confirmation is sent upon profile update or password change.

- **My Orders:**

Customers can view their past order history and current order status after logging in.

- **Cart:**

Users can add selected books to their cart and proceed to checkout to complete their purchase.

After checkout, customers receive an email receipt and order confirmation automatically.

2.2.5 Brief Information About Homepage

The homepage of the Bookstore Management System offers the following features for different types of users:

Guest Users (Unregistered Visitors):

- Can browse the book catalog.
- Can search for books by name, author, or category. Can sign up to create a new account (and receive a registration confirmation email)

Registered Users:

- Can sign in to their account to place orders and manage their profile.
- Receive email alerts for order confirmation, payment, and delivery updates

All Users (Including Guests):

- Can track their order status without logging in, by using their order ID or tracking number.

CHAPTER - 3

LITERATURE SURVEY

2.1 AIM

The primary aim of developing the Bookstore Management System is to replace the traditional method of managing book sales and records with a computerized online platform. This system enables customers to browse, search, and purchase books online while allowing administrators to efficiently manage inventory, sales, and user data.

The system aims to provide a fast, accurate, and secure way to handle bookstore operations, improve customer satisfaction, and reduce manual errors. It also generates real-time reports that help bookstore owners make informed business decisions quickly and accurately.

Key Highlights:

- **Faster Operations:** Simplifies book sales and inventory management by automating manual tasks.
- **Data Security:** Uses MySQL Server as the backend to ensure safe and reliable data storage with minimal risk of data loss.
- **Automated Email Notifications:** Integrated with a Gmail-based notification system that automatically sends emails to users during registration, order placement, payment confirmation, and delivery updates.
- **Admin Alerts:** Administrators receive email alerts for new customer registrations, new orders, and stock updates.
- **Scalable and Lightweight:** Built on an open-source platform, the system is easy to scale and requires minimal resources.

2.2 Existing System

In the existing manual system, customers must visit physical bookstores to:

- Browse and search for books manually
- Inquire about book availability and pricing

- Purchase books by paying in cash or offline methods

This traditional approach consumes a lot of time, effort, and paperwork. Bookstore staff must maintain inventory and sales records in files or spreadsheets, which is prone to human error, data duplication, and loss.

Limitations of the Existing System:

- Requires physical presence to view or purchase books
- Manual record-keeping leads to inefficiency and errors
- No instant search or category filtering
- Reports and stock updates are difficult to generate
- No real-time communication or updates for customers
- Lack of data security and backup

Overall, managing large inventories and customer transactions manually is time-consuming, error-prone, and not suitable for modern digital needs.

2.3 Proposed System

The proposed Bookstore Management System is a web-based application that simplifies the process of buying and selling books through an online platform. It allows customers to register, browse book categories, search titles/authors, place online orders, and make secure payments — all from a single, user-friendly interface.

Once an order is placed, the system automatically updates the admin dashboard with the order details, stock adjustments, and payment status.

Key Features and Advantages

- **Online Registration & Login:** Users can register easily and receive a confirmation email after signup.
- **Book Catalog & Search:** Browse books by category, author, or price with quick search options.
- **Online Ordering & Payment:** Add books to cart, make secure payments, and receive

order confirmation via email.

- **Admin Dashboard:** Admins can view all orders, track payments, and manage inventory in real time.
- **Email Notifications:** Automatic Gmail alerts for registration, order placement, payment confirmation, and delivery updates.
- **Reduced Manual Work:** Automates sales, inventory, and customer management.
- **Role-Based Access:** Ensures data security for customers, staff, and admins.
- **Real-Time Status Updates:** Customers can track orders and receive live email updates.
- **Report Generation:** Admins can generate detailed sales and performance reports instantly.

CHAPTER – 4

HARDWARE AND SOFTWARE REQUIREMENTS

3.1 Hardware Requirement:

Hardware	- Pentium
Processor	- 11th Gen Intel(R) Core i3
RAM	- 8 GB or higher
Hard Disk	- 500GB or higher
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse
Monitor	- SVGA

3.1 Software Requirement:

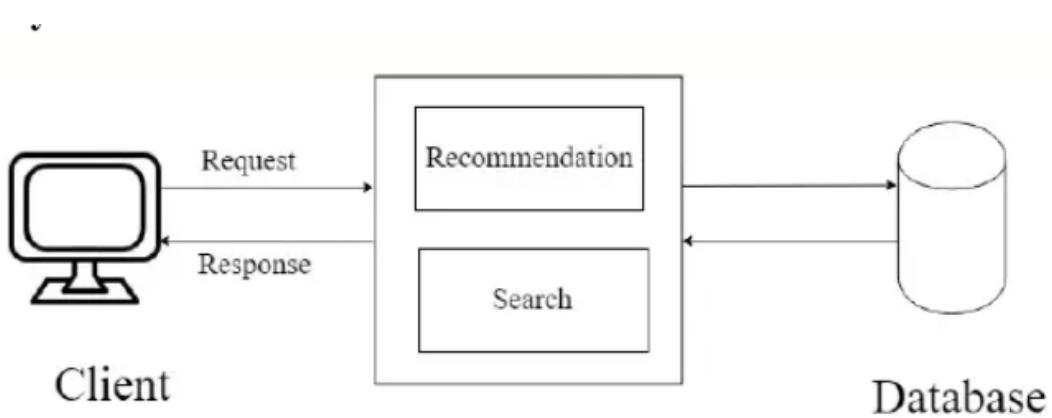
Operating System	: Windows
Technology	: PHP
Web Technologies	: Html, JavaScript, CSS
IDE	: Notepad, visualstudio code
Web Server one	: XAMPP
Database	: MySQL

CHAPTER – 5

SYSTEM DESIGN

4.1 INTRODUCTION:

System Architecture



Design is the first and most important phase of software development. It involves applying principles and techniques to define a system in sufficient detail to permit its successful implementation.

Once the software requirements have been gathered and analyzed, the software design phase begins. This phase includes three main technical activities—design, coding (implementation), and testing—which together ensure that the system meets user needs efficiently.

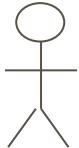
Among these, design plays a critical role. The design phase transforms the analyzed requirements into a structured system model. It determines the system architecture, database structure, data flow, and interaction between modules. These design decisions directly affect the system's performance, reliability, and maintainability.

In the Bookstore Management System, the design stage helps convert customer and admin requirements—like book browsing, order tracking, and email notifications—into well-defined modules. This ensures the final system is both scalable and user-friendly.

- **UML Diagrams:**

Actor:

Represents a user or another system that interacts with the Bookstore Management System (e.g., Customer, Administrator).



Use case:

A description of a sequence of actions that the system performs to achieve a particular goal (e.g., “Place Order,” “Add Book,” “Generate Report”).

UML (Unified Modeling Language) is used for specifying, visualizing, and documenting software systems. It helps developers and stakeholders understand how the system’s components interact.

Common UML diagrams used in this project include:

- Use Case Diagram
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- State Chart Diagram

4.2 USECASE DIAGRAMS:

Use Case Diagrams model the behavior of the system and illustrate what the user expects the system to do. They help identify system functionalities and user interactions in a clear and visual format.

Elements of a Use Case Diagram:

- **Actors:** Represent users such as customers or admins.
- **Use Cases:** Represent functionalities like “Search Book,” “Manage Orders,” or “Send

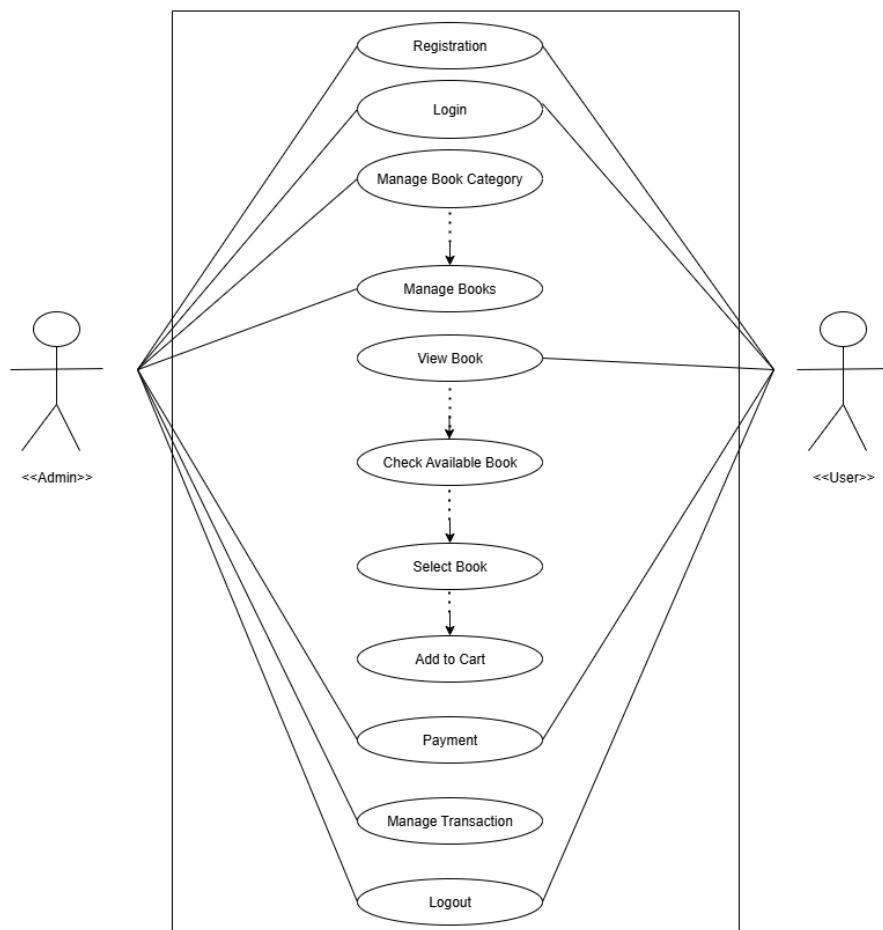
Email Notifications.”

- **System Boundary:** Defines the scope of the Bookstore Management System.

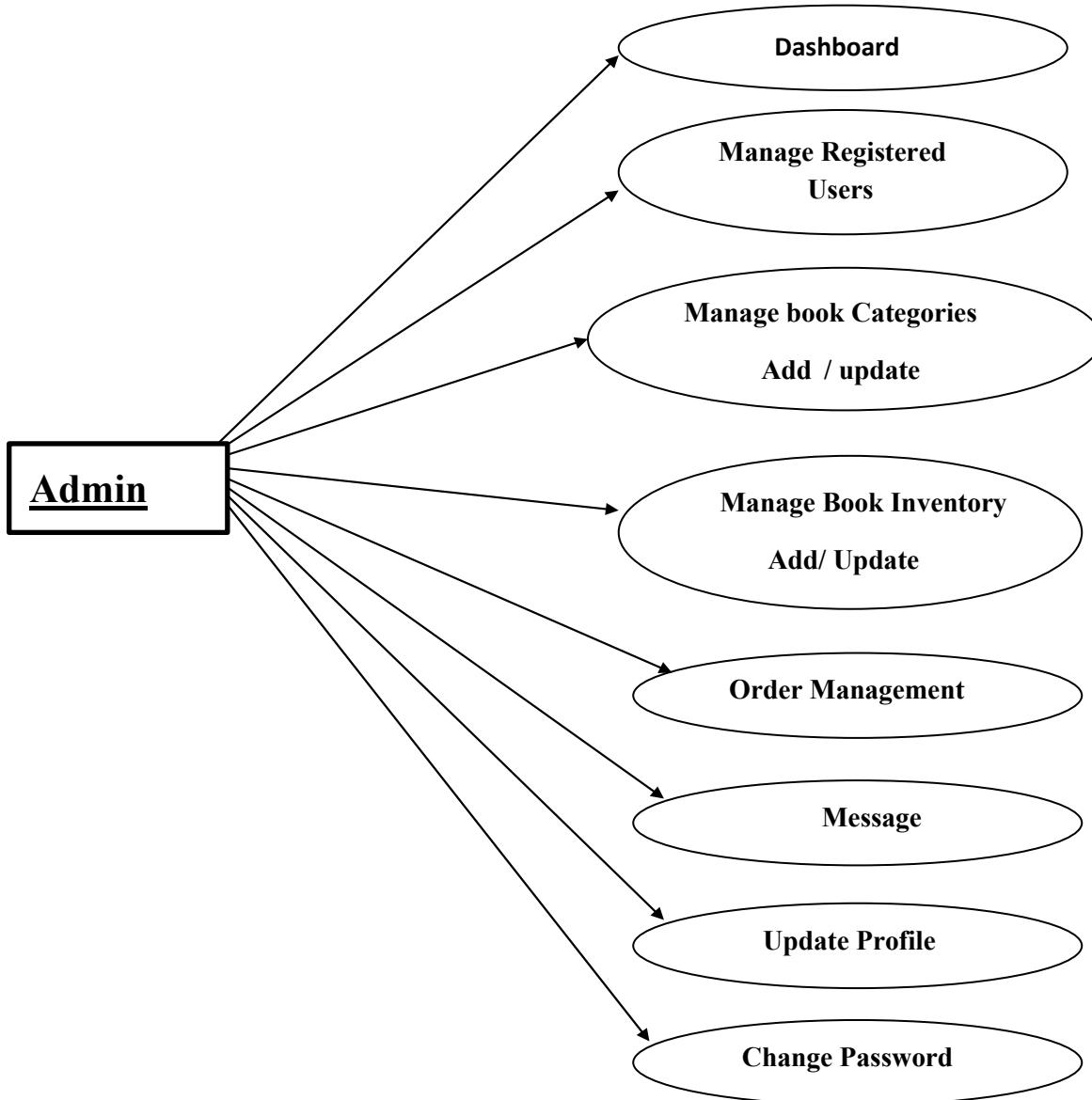
Use Case Diagrams are useful to show who can do what within the system and ensure that no essential function is missed.

USECASE DIAGRAM:

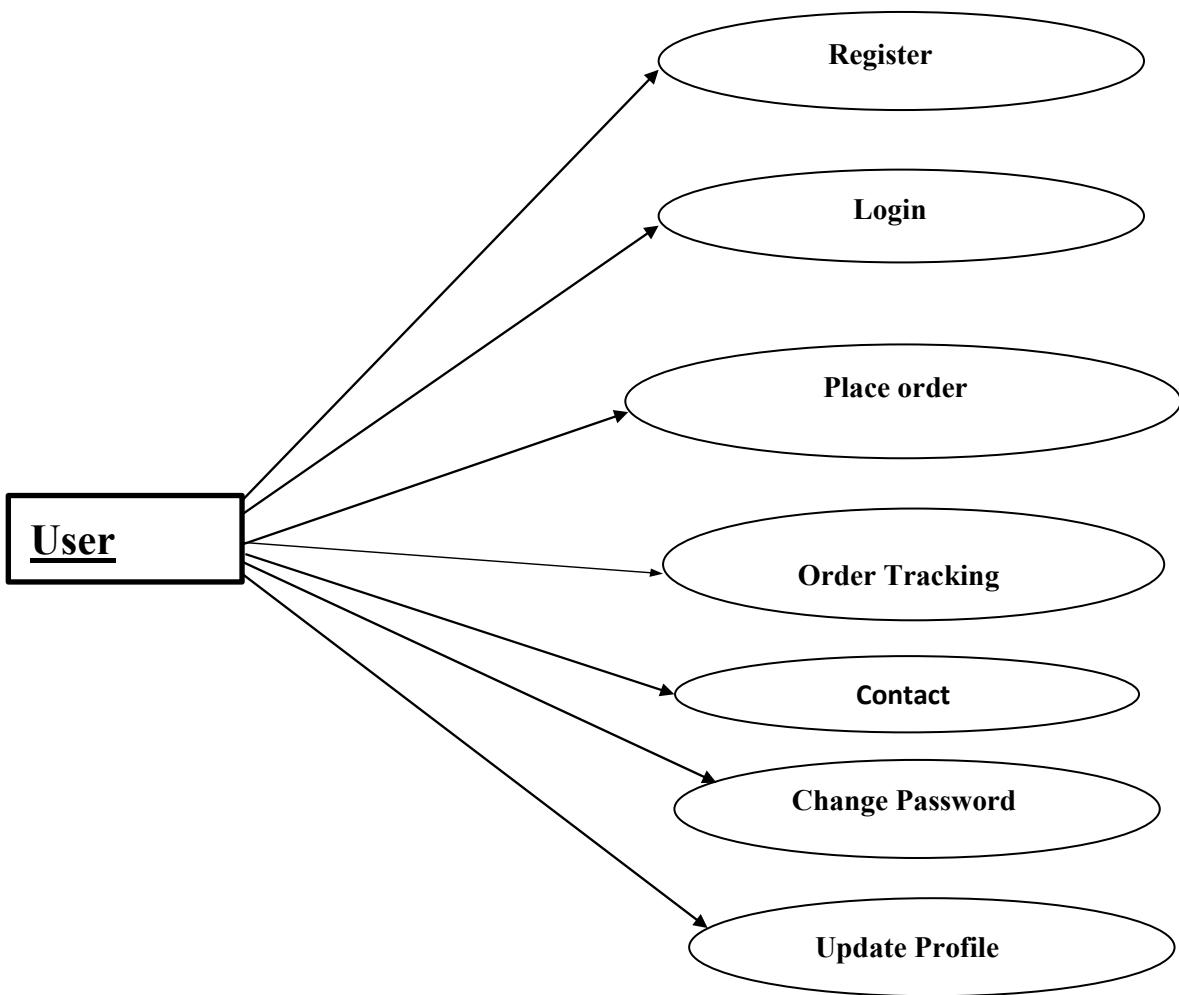
A use case is a description of set of sequence of actions, Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor Sender, Secondary Actor Receiver.



4.2.1 Admin Use Case Diagram:



4.2.2 User Use Case Diagram:



4.3 E-R Diagrams:

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 as a way 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 is:

- 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 a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

4.4 Connectivity and Cardinality

The basic types of connectivity for relations are: one-to-one, one-to-many, and many-to-many. A one-to-one (1:1) relationship is when at most one instance of entity A is associated with one instance of entity B. For example, "employees in the company" are each assigned their own office. For each employee there exists a unique office and for each office there exists a unique employee.

A one-to-many (1:N) relationship is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. An example of a 1:N relationship is a department has many employees; each employee is assigned to one department.

A many-to-many relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero , one , or many instances of entity A. The connectivity of a relationship describes the mapping of associated.

4.5 ER Notation

There is no standard for representing data objects in ER diagrams. Each modelling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used, among the more common are Bachman, crow's feet, and IDEFIX.

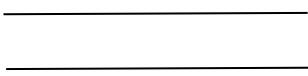
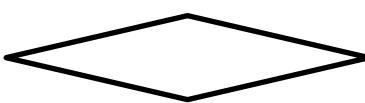
All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of n connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

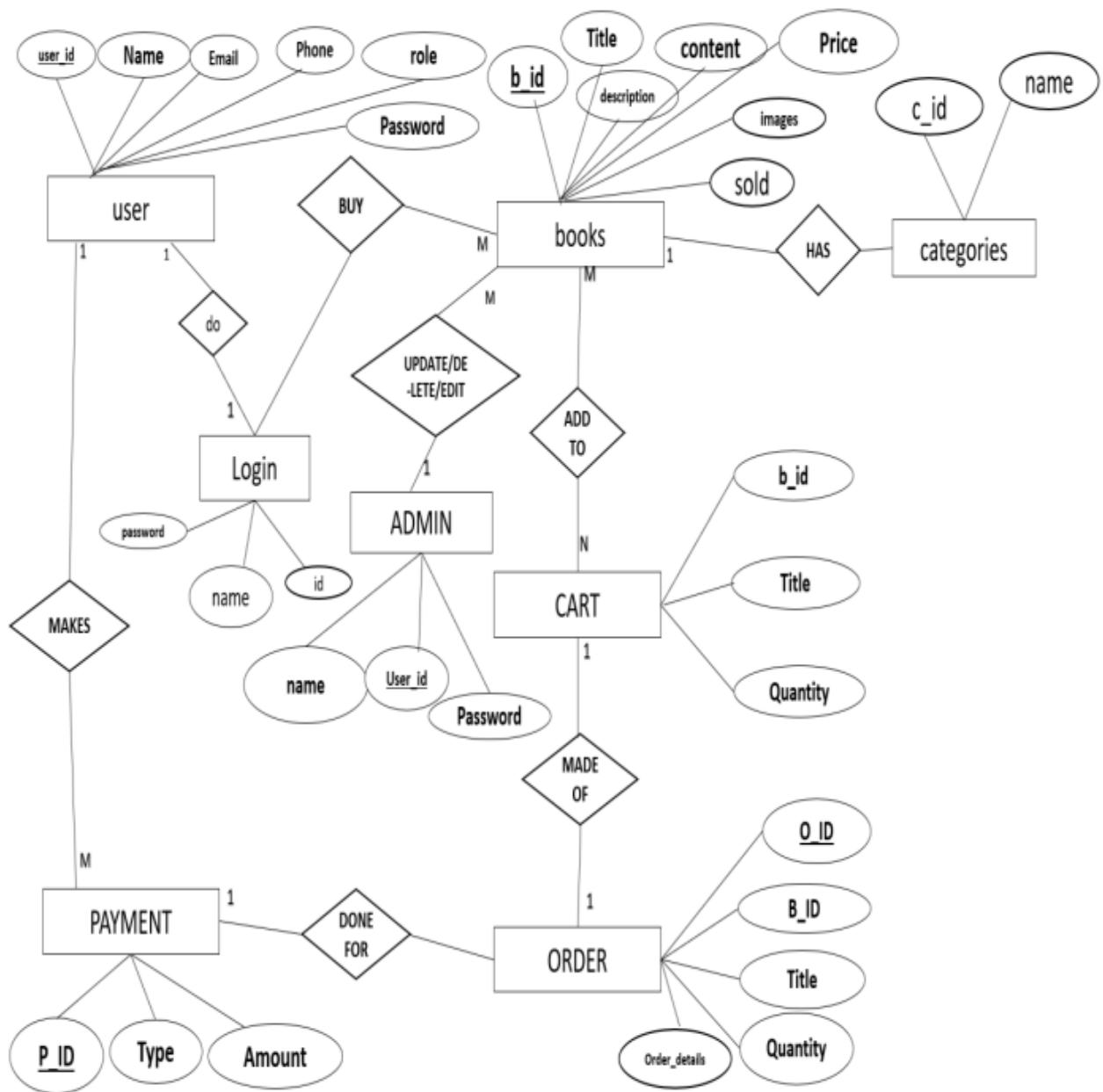
- **Entities** are represented by labelled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- **Relationships** represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs.
- **Attributes**, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should 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 one.
- **Existence** is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the har looks like a 1) next to the entity for an instance is required Optional existence is shown by placing a circle next to the entity that is option

Entity relationship diagrams are a way to represent the structure and layout of a database. An ERD is a model that identifies the concepts or entities that exist in a system and the relationships between those entities. An ERD is often used as a way to visualize a relational database: each entity represents a database table, and the relationship lines represent the keys in one table that point to specific records in related tables.

Objects There are three main objects on an ER Diagram:

1. Entities
2. Relations
3. Attributes.

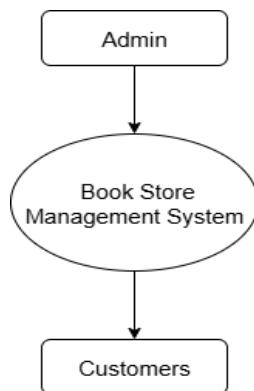
	Process -It tells the process state in a project.
	Data Flow -It tells the flow of data.
	Source/Sink -It is the project source or sink.
	Data Store -It stores the data in a particular file.
	Relationship -It shows the relationship between function units.

E-R Diagrams:

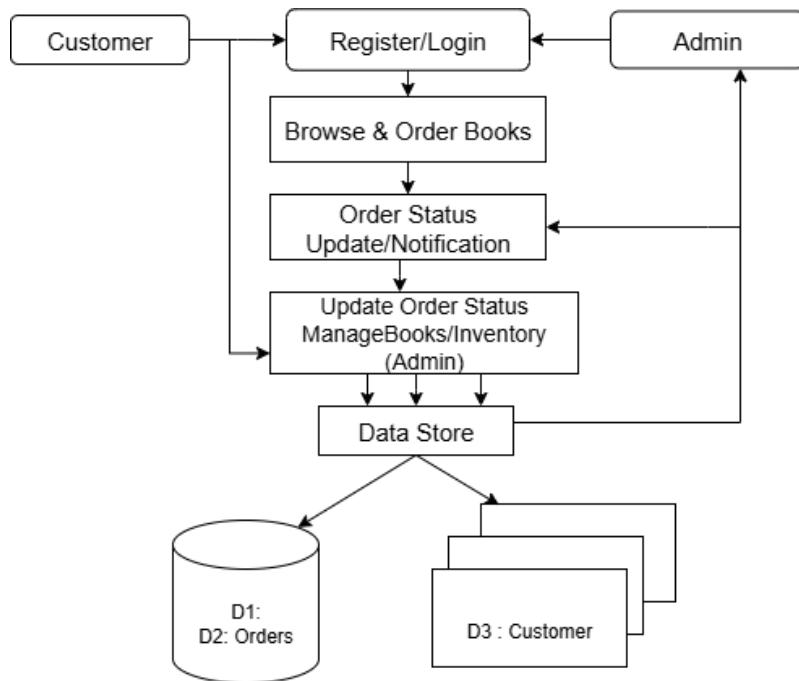
4.6 Data Flow Diagram:

Data Flow Diagram (DFD) represents the flow of data within information systems. Data Flow Diagrams (DFD) provide a graphical representation of the data flow of a system that can be understood by both technical and non-technical users. The models enable software engineers, customers, and users to work together effectively during the analysis and specification of requirements.

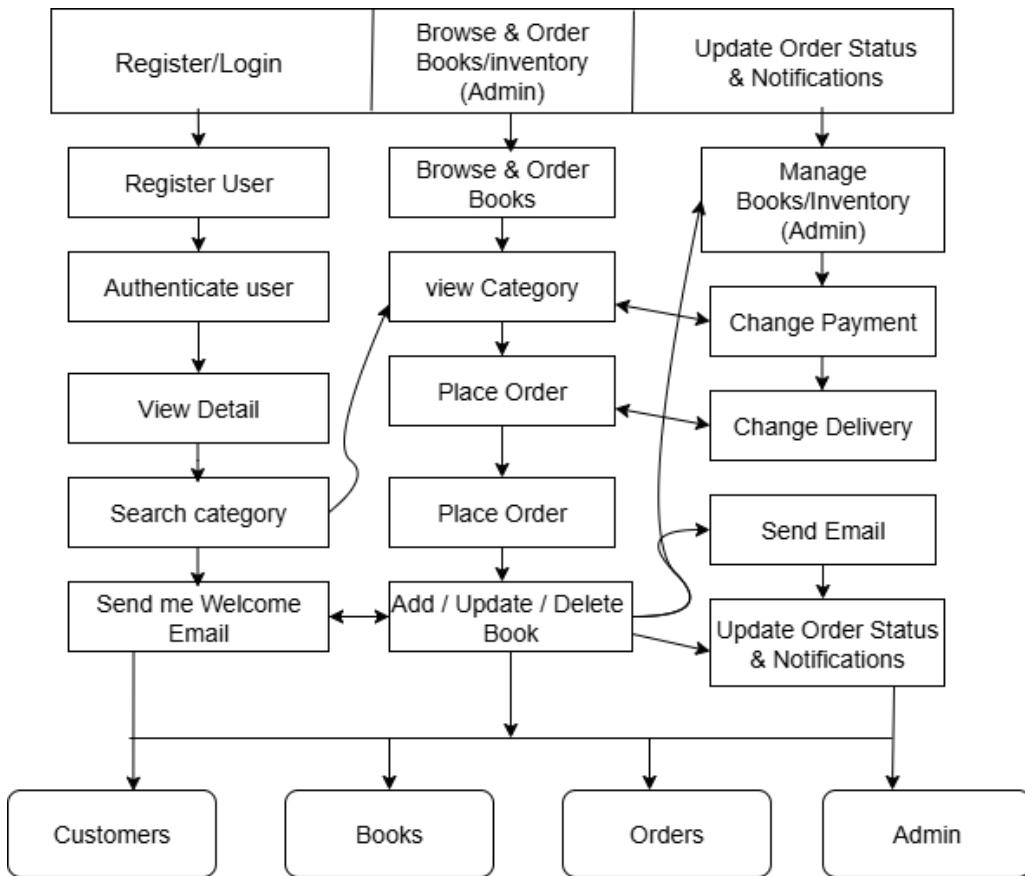
DFD Level 0:

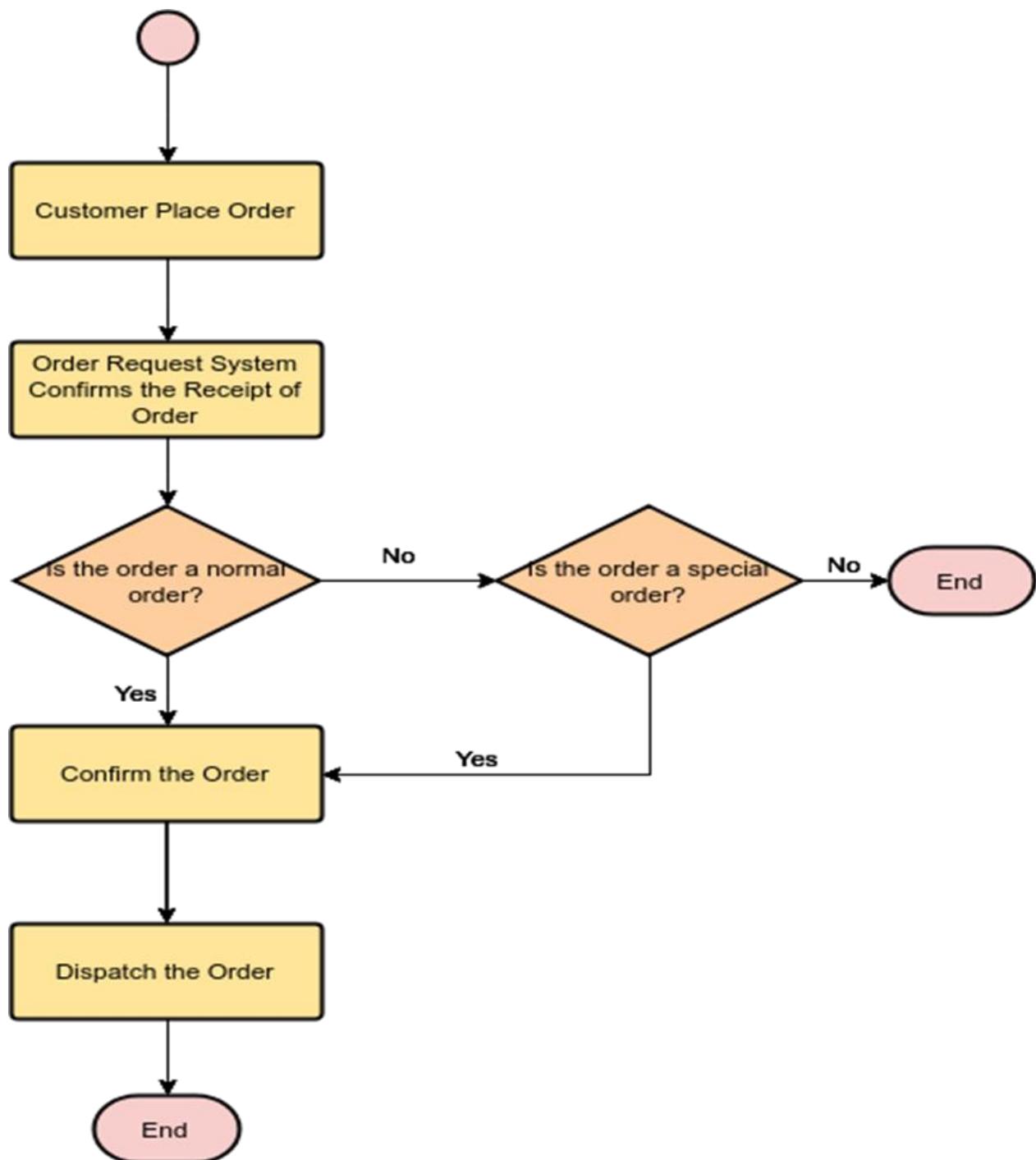


DFD Level 1:



DFD Level 2:





CHAPTER – 6

DATABASE DESIGN

The data in the Bookstore Management System must be efficiently stored and retrieved from a centralized database. Designing this database is a crucial part of the system design process. During the analysis phase, all data elements and structures—such as book details, user accounts, orders, and payment information—are identified. These are then organized and structured to create an efficient data storage and retrieval system.

A database is a collection of interrelated data organized with minimal redundancy to ensure fast, secure, and reliable access. The primary objective of this design is to make data management easy, quick, and efficient for both administrators and customers.

Relationships between entities such as Books, Users, Orders, Payments, and Admins are clearly defined to maintain consistency. Normalization techniques are applied to eliminate redundancy and ensure data integrity, stability, and optimized performance during updates and queries.

The Bookstore Management System uses a MySQL database to handle all data operations, including book inventory, customer records, order history, and payment transactions. This relational database design helps ensure accuracy, scalability, and security for real-time operations.

By structuring data in this way, the system provides:

- Faster data retrieval for book searches and order tracking.
- Reduced data redundancy and improved storage efficiency.
- Easier maintenance and scalability for future updates.
- Secure and reliable handling of user and payment information.

Thus, the database design forms the backbone of the Bookstore Management System, ensuring smooth operations, consistent data management, and enhanced performance across all modules.

5.1 Book Ordering System (BOS) contains 7 MySQL tables:

- **tbldatabase books:** This table store the Database books

Table	Action	Rows	Type	Collation	Size	Overhead
cart	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 Kib	-
message	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 Kib	-
orders	Browse Structure Search Insert Empty Drop	18	InnoDB	utf8mb4_general_ci	16.0 Kib	-
products	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 Kib	-
register	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	32.0 Kib	-
replies	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 Kib	-
users	Browse Structure Search Insert Edit Drop View	~0	InnoDB	utf8mb4_general_ci	-	-

- **tbladmin :** This table store the admin login details

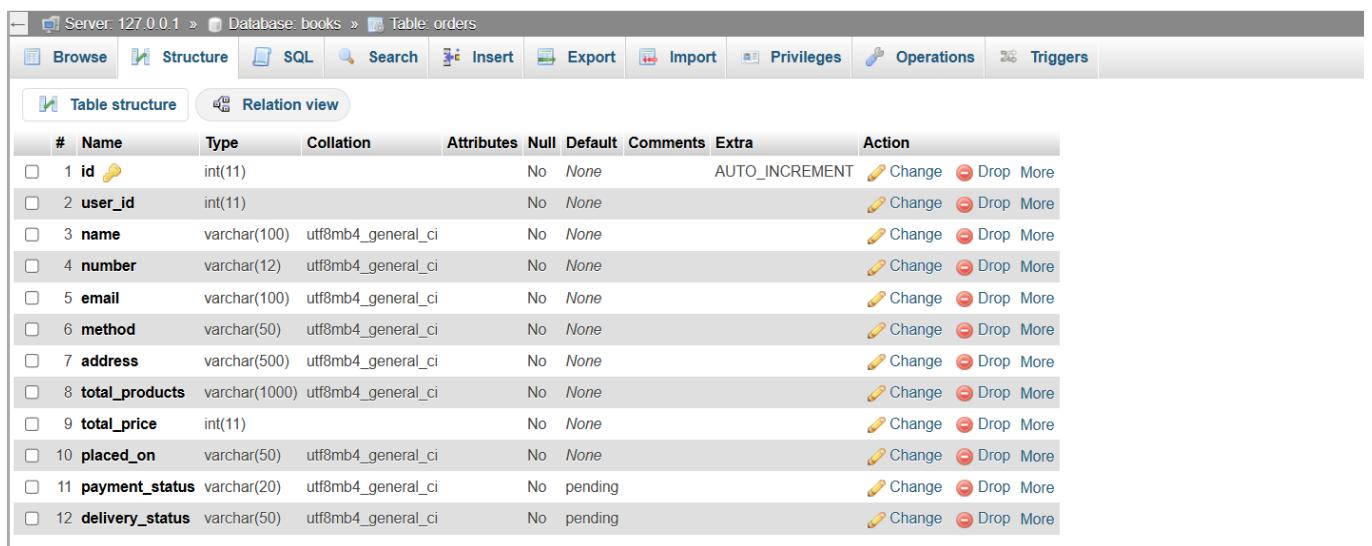
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	0		
2	name	varchar(100)	utf8mb4_general_ci		No	None		
3	email	varchar(100)	utf8mb4_general_ci		No	None		

- **tblproducts :** This table stores the food products

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT		Change Drop More
2	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	price	int(11)			No	None			Change Drop More
4	image	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
5	category	varchar(50)	utf8mb4_general_ci		No	General			Change Drop More

- **tblorders :** This Table store the order Details

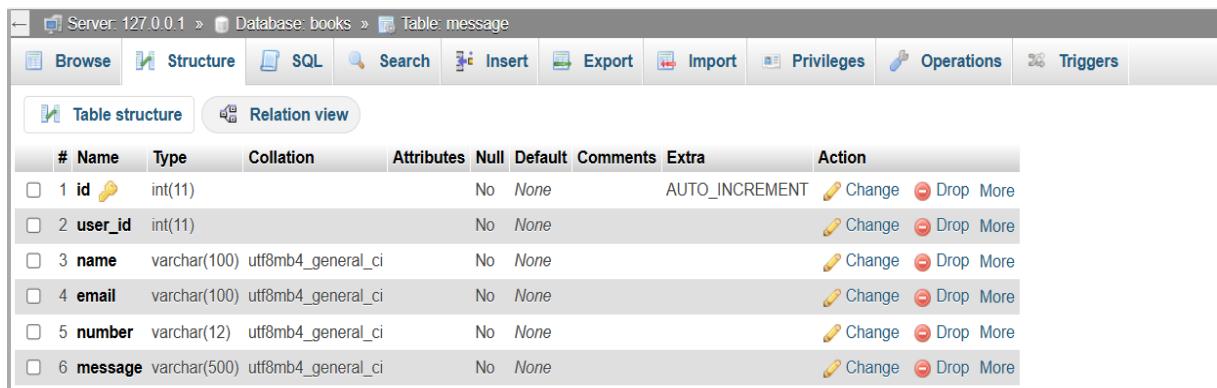
BOOK STORE



The screenshot shows the 'orders' table structure in MySQL Workbench. The table has 12 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	user_id	int(11)			No	None			Change Drop More
3	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	number	varchar(12)	utf8mb4_general_ci		No	None			Change Drop More
5	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
6	method	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
7	address	varchar(500)	utf8mb4_general_ci		No	None			Change Drop More
8	total_products	varchar(1000)	utf8mb4_general_ci		No	None			Change Drop More
9	total_price	int(11)			No	None			Change Drop More
10	placed_on	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
11	payment_status	varchar(20)	utf8mb4_general_ci		No	pending			Change Drop More
12	delivery_status	varchar(50)	utf8mb4_general_ci		No	pending			Change Drop More

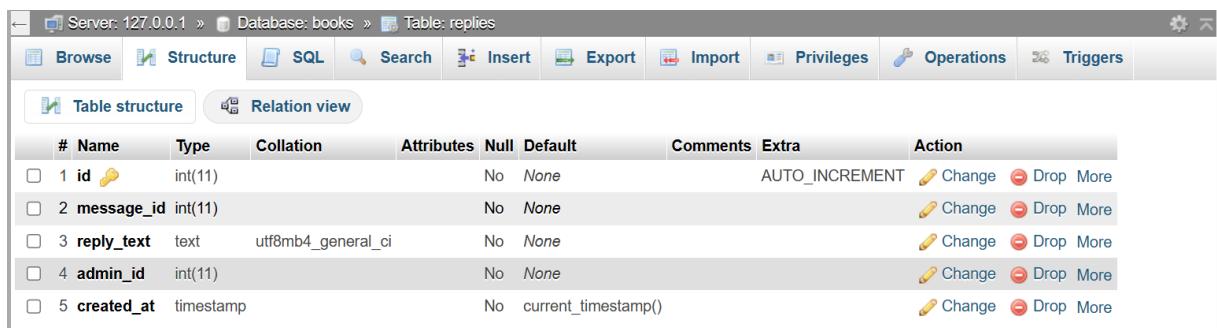
- **tblmessages** : This Table Store the Message Details



The screenshot shows the 'message' table structure in MySQL Workbench. The table has 6 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	user_id	int(11)			No	None			Change Drop More
3	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
5	number	varchar(12)	utf8mb4_general_ci		No	None			Change Drop More
6	message	varchar(500)	utf8mb4_general_ci		No	None			Change Drop More

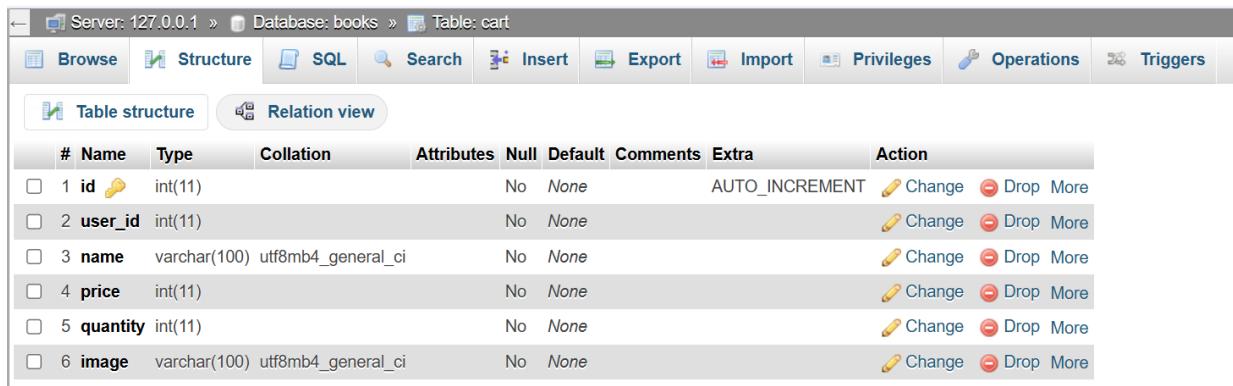
- **tblreplies** : This Table Store the Reply Details



The screenshot shows the 'replies' table structure in MySQL Workbench. The table has 5 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	message_id	int(11)			No	None			Change Drop More
3	reply_text	text	utf8mb4_general_ci		No	None			Change Drop More
4	admin_id	int(11)			No	None			Change Drop More
5	created_at	timestamp			No	current_timestamp()			Change Drop More

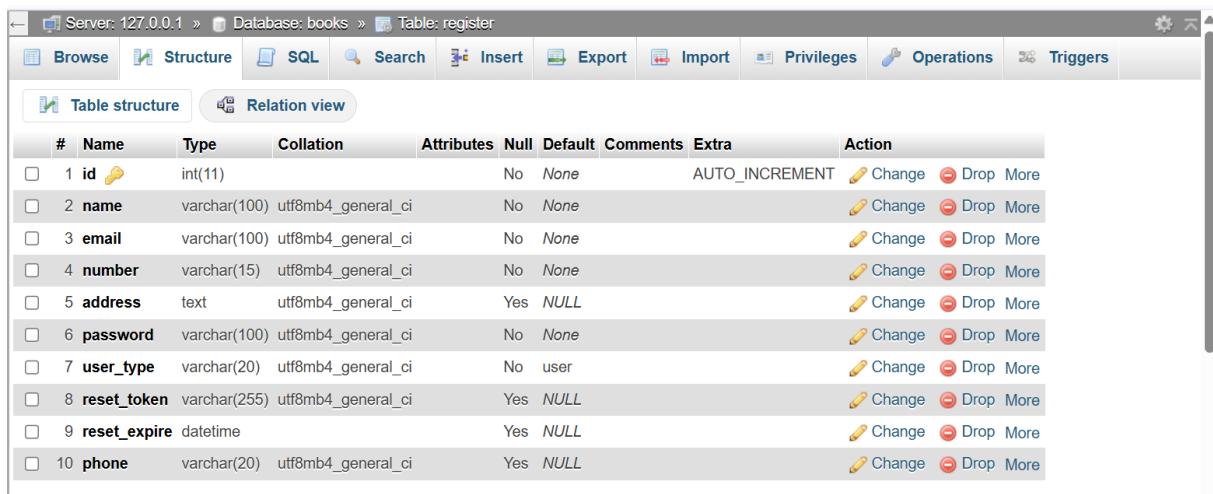
- **tblcart:** This Table Store the cart



The screenshot shows the 'Structure' tab for the 'cart' table in MySQL Workbench. The table has six columns: id, user_id, name, price, quantity, and image. The 'id' column is defined as int(11) with AUTO_INCREMENT, while the others are varchar(100) with utf8mb4_general_ci collation.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	user_id	int(11)			No	None			Change Drop More
3	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	price	int(11)			No	None			Change Drop More
5	quantity	int(11)			No	None			Change Drop More
6	image	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More

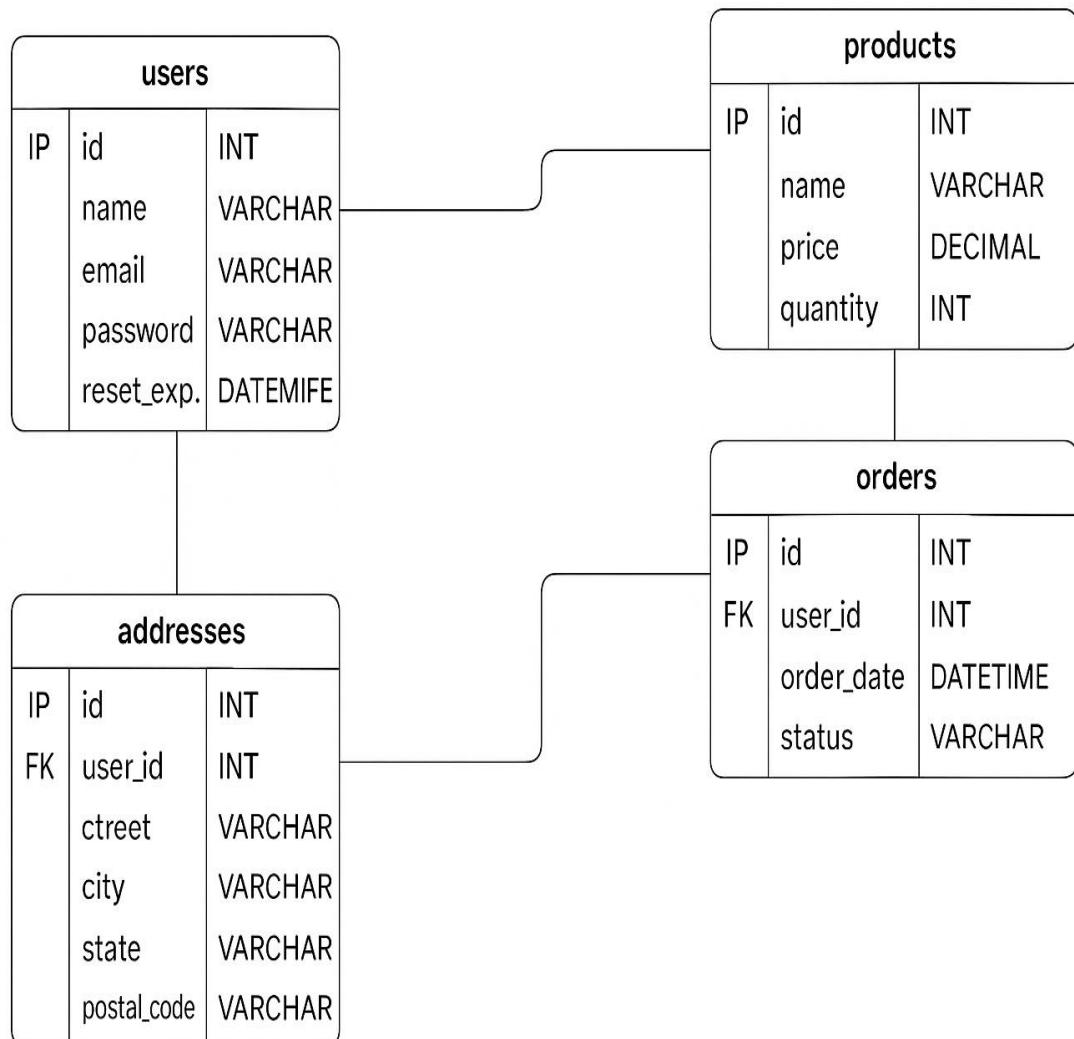
- **tblregister :** This Table Store the Register



The screenshot shows the 'Structure' tab for the 'register' table in MySQL Workbench. The table has ten columns: id, name, email, number, address, password, user_type, reset_token, reset_expire, and phone. The 'id' column is defined as int(11) with AUTO_INCREMENT, while the others are varchar or text types with utf8mb4_general_ci collation.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	number	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
5	address	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	password	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
7	user_type	varchar(20)	utf8mb4_general_ci		No	user			Change Drop More
8	reset_token	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
9	reset_expire	datetime			Yes	NULL			Change Drop More
10	phone	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More

5.2 Tables Relationship Diagram



CHAPTER – 7

SYSTEM TESTING

6.1 Project Implementation & Testing

At the end of the system design phase, the actual coding and development of the proposed system begins. During the development phase, the table structure of the database will be built first to define appropriate data types that suit the backend development. Following this, the system connection and data transfer mechanisms will be implemented.

Next, the computer-side client program will be developed, followed by the web services that allow the mobile client application to communicate with the server. Once the server-side components are complete, the mobile-side client program will be developed.

In the testing phase, several test cases will be executed to evaluate the system's reliability and accuracy. Based on these test cases, a system testing report will be generated to identify any weaknesses in the system and make necessary improvements.

During the deployment phase, training sessions will be provided to restaurant staff. These sessions will cover system operations, procedures for handling various scenarios, and instructions to ensure proper usage of the system.

6.2 Implementation Issues & Challenges

During the system implementation phase, several challenges must be confronted, as this phase involves end users testing the production system under various scenarios. The possible challenges that may be encountered include the following:

6.3 User without background

Users are required to have basic knowledge of how to operate a computer system and an Android mobile phone in order to use the system effectively. However, providing training to such users may be challenging, as it is often assumed that basic information can be easily obtained from the internet.

6.4 The screen size of different device

This is one of the issues that may be encountered during system implementation. Since users can download the mobile application from the official website and use it as a client device to place orders, inconsistencies may arise. If the screen size of the user's device is too small or too large, the convenience and user interface of the application may not appear consistent across all devices.

6.5 Server performance

During real-time system implementation, there may be a large number of clients accessing the server simultaneously. This can lead to slower connection speeds and reduced system performance. In extreme cases, it may even cause the server to crash if the load exceeds acceptable limits.

CHAPTER – 8

IMPLEMENTATION DETAILS

8.1: FRONT END



4.1.1 HTML

Hypertext Mark-up Language (HTML) is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.



4.1.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.



4.1.3 JavaScript

JavaScript is a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications.

4.2: BACK END

4.2.1 PHP



PHP is a server side scripting language that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages. PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only. A PHP file contains PHP tags and ends with the extension ".php". PHP: Hypertext Pre-processor (or simply PHP) is a general-purpose programming language originally designed for web development.

4.2.2 MySQL



MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). It is one part of the very popular LAMP platform consisting of Linux, Apache, My SQL, and PHP. Currently My SQL is owned by Oracle. My SQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows, or Mac OS. Wikipedia and YouTube use My SQL. These sites manage millions of queries each day. My SQL comes in two versions: My SQL server system and My SQL embedded system.

RDBMS TERMINOLOGY

Before we proceed to explain MySQL database system, let's revise few definitions related to

database.

- **Database:** A database is a collection of tables, with related data.
- **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column:** One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row:** A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- **Redundancy:** Storing data twice, redundantly to make the system faster.
- **Primary Key:** A primary key is unique. A key value cannot occur twice in one table. With a key, you can find at most one row.
- **Foreign Key:** A foreign key is the linking pin between two tables.
- **Compound Key:** A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index:** An index in a database resembles an index at the back of a book.
- **Referential Integrity:** Referential Integrity makes sure that a foreign key value always points to an existing row.

CHAPTER – 9

CODING

HOME PAGE

```

<?php
include 'config.php';
session_start();

$user_id = isset($_SESSION['user_id']) ? $_SESSION['user_id'] : null;

if (isset($_POST['add_to_cart'])) {
    if ($user_id) {
        $pro_name  = $_POST['product_name'];
        $pro_price = $_POST['product_price'];
        $pro_quantity = $_POST['product_quantity'];
        $pro_image = $_POST['product_image'];

        $check = mysqli_query($conn, "SELECT * FROM `cart` WHERE name='$pro_name' AND user_id='$user_id'" or die('query failed'));

        if (mysqli_num_rows($check) > 0) {
            $message[] = 'Already added to cart!';
        } else {
            mysqli_query($conn, "INSERT INTO `cart`(user_id, name, price, quantity, image)
VALUES
('$user_id','$pro_name','$pro_price','$pro_quantity','$pro_image')") or die('query2 failed');
            $message[] = 'Product added to cart!';
        }
    } else {
        $message[] = 'Please login to add products to cart!';
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bookstore - Home</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-

```

```

awesome/6.4.0/css/all.min.css">
<link rel="stylesheet" href="style.css">
<link rel="stylesheet" href="home.css">

<style>
.msg-box {
  margin: 20px auto;
  padding: 12px 20px;
  border: 1px solid #c02736ff;
  background: #eafaf1;
  color: #c02736ff;
  text-align: center;
  font-weight: bold;
  border-radius: 6px;
  max-width: 500px;
  animation: fadeOut 4s forwards;
}
@keyframes fadeOut {
  0% { opacity: 1; }
  70% { opacity: 1; }
  100% { opacity: 0; display: none; }
}

h2.section-title {
  text-align: center;
  font-size: 36px;
  font-weight: 900;
  margin: 0 auto 30px;
  text-transform: uppercase;
  border-bottom: 3px solid #111;
  width: fit-content;
  padding-bottom: 6px;
}

/* Category grid */
.category-grid {
  max-width: 1200px;
  margin: auto;
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));
  gap: 20px;
}

```

```

padding: 0 16px 50px;
}
.category-card {
border: 2px solid #111;
background: #fff;
padding: 28px 18px;
text-align: center;
text-decoration: none;
border-radius: 8px;
transition: 0.2s;
color: #111;
}
.category-card i {
font-size: 50px;
margin-bottom: 12px;
}
.category-card h3 {
font-size: 20px;
font-weight: 700;
}
.category-card:hover {
background: #000;
color: #fff;
}
.category-card:hover i,
.category-card:hover h3 {
color: #fff;
}

/* Latest Books */
.books-grid {
max-width: 1200px;
margin: auto;
display: grid;
grid-template-columns: repeat(auto-fit, minmax(260px, 1fr));
gap: 20px;
padding: 0 16px 50px;
}
.book-card {
background: #fff;
border: 1px solid #ddd;

```

```
border-radius: 8px;
padding: 15px;
text-align: center;
transition: 0.3s;
}
.book-card:hover {
  transform: translateY(-5px);
  box-shadow: 0px 6px 15px rgba(0,0,0,0.15);
}
.book-card img {
  width: 100%;
  height: 220px;
  object-fit: contain;
  margin-bottom: 10px;
}
.book-card h3 {
  font-size: 20px;
  font-weight: 600;
  margin: 10px 0;
  color: #111;
}

/* Category display under book */
.book-card .category {
  font-size: 15px;
  color: #555;
  margin-top: -5px;
  margin-bottom: 8px;
}
.book-card .category span {
  font-weight: 600;
  color: #c02736ff;
  text-transform: capitalize;
}

.book-card .price {
  font-size: 18px;
  font-weight: bold;
  color: #e67e22;
  margin-bottom: 15px;
}
```

```

.book-card input[type=number] {
    width: 70px;
    padding: 6px;
    margin-bottom: 12px;
    text-align: center;
    border: 2px solid #c02736ff;
    border-radius: 6px;
    font-weight: 600;
}

/* Unified button style */
.btn, .product_btn, .read-btn, .contact-btn {
    padding: 10px 22px;
    background: #c02736ff;
    color: #fff;
    border: 2px solid #c02736ff;
    border-radius: 6px;
    cursor: pointer;
    font-weight: 600;
    transition: all 0.3s ease;
    white-space: nowrap;
    display: inline-block;
    text-decoration: none;
}
.btn:hover, .product_btn:hover, .read-btn:hover, .contact-btn:hover {
    background: #fff;
    color: #c02736ff;
    border: 2px solid #c02736ff;
}

/* About / Contact */
.about_cont, .questions_cont {
    max-width: 1100px;
    margin: 60px auto;
    display: flex;
    align-items: center;
    gap: 30px;
    padding: 0 16px;
}
.about_cont img {
    width: 50%;
}

```

```

border-radius: 8px;
}
.about_descript h2, .questions h2 {
font-size: 28px;
font-weight: 800;
margin-bottom: 12px;
}
.about_descript p, .questions p {
color: #555;
line-height: 1.6;
margin-bottom: 18px;
}

@media(max-width: 768px){
.about_cont, .questions_cont { flex-direction: column; text-align: center; }
.about_cont img { width: 100%; }
}
</style>
</head>
<body>

<?php include 'user_header.php'; ?>

<?php if (isset($message)) { foreach ($message as $msg) { echo "<div class='msg-box'>$msg</div>"; } } ?>

<!-- Hero Section -->
<section class="home_cont">
<div class="main_descrip">
<h1>The Bookshelf</h1>
<p>Explore, Discover, and Buy Your Favorite Books</p>
<button class="product_btn" onclick="window.location.href='shop.php';">Discover More</button>
</div>
</section>

<!-- BOOK CATEGORY -->
<section>
<h2 class="section-title">BOOK CATEGORY</h2>
<div class="category-grid">
<a class="category-card" href="shop.php?category=Fiction"><i class="fa-solid fa-book-
```

```

open"></i><h3>Fiction</h3></a>
    <a class="category-card" href="shop.php?category=Non-Fiction"><i class="fa-solid fa-feather-pointed"></i><h3>Non-Fiction</h3></a>
        <a class="category-card" href="shop.php?category=Children"><i class="fa-solid fa-child"></i><h3>Children</h3></a>
            <a class="category-card" href="shop.php?category=Textbooks"><i class="fa-solid fa-user-graduate"></i><h3>Textbooks</h3></a>
        </div>
    </section>


<section>
    <h2 class="section-title">LATEST BOOKS</h2>
    <div class="books-grid">
        <?php
            $select_products = mysqli_query($conn, "SELECT * FROM `products` ORDER BY id DESC LIMIT 3") or die('query failed');
            if (mysqli_num_rows($select_products) > 0) {
                while ($fetch_products = mysqli_fetch_assoc($select_products)) {
                    ?>
                    <form action="" method="post" class="book-card">
                        
                        <h3><?php echo $fetch_products['name']; ?></h3>
                        <p class="category">Category: <span><?php echo $fetch_products['category']; ?></span></p>
                        <div class="price">₹<?php echo $fetch_products['price']; ?>/-</div>
                        <input type="number" name="product_quantity" min="1" value="1">
                        <input type="hidden" name="product_name" value="<?php echo $fetch_products['name']; ?>">
                        <input type="hidden" name="product_price" value="<?php echo $fetch_products['price']; ?>">
                        <input type="hidden" name="product_image" value="<?php echo $fetch_products['image']; ?>">
                        <input type="submit" value="Add to Cart" name="add_to_cart" class="btn">
                    </form>
                <?php } } else { echo '<p class="empty">No Books Added Yet!</p>'; } ?>
            </div>
        </section>


<section class="about_cont">
    

```

```
<div class="about_descript">
  <h2>Discover Our Story</h2>
  <p>At Book Store, we are passionate about connecting readers with captivating stories, inspiring ideas, and a world of knowledge. Our bookstore is more than just a place to buy books; it's a haven for book enthusiasts, where the love for literature thrives.</p>
  <a href="about.php" class="read-btn">Read More</a>
</div>
</section>

<!-- CONTACT -->
<section class="questions_cont">
  <div class="questions">
    <h2>Have Any Queries?</h2>
    <p>At Bookiee, we value your satisfaction and strive to provide exceptional customer service. If you have any questions, concerns, or inquiries, our dedicated team is here to assist you every step of the way.</p>
    <a href="contact.php" class="contact-btn">Contact Us</a>
  </div>
</section>

<?php include 'footer.php'; ?>
</body>
</html>
```

REGISTER PAGE

```
<?php
include 'config.php';
session_start();

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'vendor/autoload.php'; // PHPMailer autoload

if (isset($_POST['submit'])) {
  $name    = mysqli_real_escape_string($conn, $_POST['name']);
  $email   = mysqli_real_escape_string($conn, $_POST['email']);
  $number  = mysqli_real_escape_string($conn, $_POST['number']);
  $password = mysqli_real_escape_string($conn, password_hash($_POST['password'], PASSWORD_BCRYPT));
}
```

```

$cpassword = $_POST['cpassword'];
$user_type = $_POST['user_type'];

$select_users = mysqli_query($conn, "SELECT * FROM `register` WHERE email='$email'
OR number='$number'" ) or die(mysqli_error($conn));

if (mysqli_num_rows($select_users) > 0) {
    $message[] = 'Email or Phone number already exists!';
} else {
    if ($_POST['password'] !== $cpassword) {
        $message[] = 'Confirm password not matched!';
    } else {
        $insert      = mysqli_query($conn,      "INSERT      INTO
`register`(name,email,number,password,user_type)
VALUES('$name','$email','$number','$password','$user_type')") or
die(mysqli_error($conn));

        if ($insert) {
            // Send Email using PHPMailer (same as before)
            $mail = new PHPMailer(true);
            try {
                $mail->isSMTP();
                $mail->Host      = 'smtp.gmail.com';
                $mail->SMTPAuth  = true;
                $mail->Username   = 'onlinebookstore2025@gmail.com';
                $mail->Password   = 'yity rhwt tzly wttn';
                $mail->SMTPSecure = 'tls';
                $mail->Port       = 587;

                $mail->setFrom('onlinebookstore2025@gmail.com', 'Bookstore');
                $mail->addAddress($email, $name);

                $mail->isHTML(true);
                $mail->Subject = "Welcome to Bookstore, $name!";
                $mail->Body   = "Hello <b>$name</b>,<br><br>
                    Thank you for registering at <b>Bookstore</b>!<br>
                    We are excited to have you with us.<br><br>
                    <b>Happy Reading!</b><br>
                    - Bookstore Team";
                $mail->send();
            } catch (Exception $e) {

```

```

        error_log("Mailer Error: {$mail->ErrorInfo}");
    }

    $_SESSION['user_name'] = $name;
    $_SESSION['user_email'] = $email;

    header("Location: login.php");
    exit();
} else {
    $message[] = "Something went wrong!";
}
}

}

}

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Register | Book Store</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" />
    <link rel="stylesheet" href="style.css" />
<style>
.register-section{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    padding: 60px 20px;
}
.register-box{
    width: 100%;
    max-width: 400px;
    border: 1px solid #ddd;
    padding: 30px;
    text-align: center;
    border-radius: 8px;
    background: #fff;
    box-shadow: 0 4px 10px rgba(0,0,0,0.1);
}

```

```
}

.register-box h2{
    font-size: 22px;
    margin-bottom: 20px;
    font-weight: bold;
}

.register-box .inputbox{
    margin-bottom: 15px;
    text-align: left;
}

.register-box .inputbox input,
.register-box .inputbox select{
    width: 100%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    outline: none;
}

.register-box input[type="submit"] {
    display:block;
    width:100%;
    margin:10px 0;
    padding:12px;
    background:#d32f2f;
    color:#000;
    border:none;
    border-radius:5px;
    font-weight:bold;
    cursor:pointer;
    text-decoration:none;
    transition:0.3s;
}

.register-box input[type="submit"]:hover {
    background:#b71c1c;
    color:#fff;
}

.links{
    margin-top: 15px;
    font-size: 14px;
}

.links a{
```

```

font-weight: bold;
text-decoration: none;
margin-left: 5px;
color: #000;
}

.message{
position: relative;
margin:10px auto;
padding:10px 35px 10px 10px;
border:1px solid red;
color:red;
background:#ffe5e5;
border-radius: 5px;
max-width: 400px;
text-align: left;
}

.message .close-btn{
position: absolute;
right: 10px;
top: 50%;
transform: translateY(-50%);
font-weight: bold;
font-size: 18px;
color: red;
cursor: pointer;
}

</style>
</head>
<body>

<?php include 'user_header.php'; ?>

<section class="register-section">
<?php
if (isset($message)) {
foreach ($message as $msg) {
echo '<div class="message">
&times;

```

```

    }
}

?>

<div class="register-box">
    <h2>REGISTER NOW</h2>
    <form id="registerForm" action="" method="post" onsubmit="return validateForm();">
        <div class="inputbox">
            <input type="text" id="name" name="name" placeholder="Enter your name" required>
        </div>
        <div class="inputbox">
            <input type="email" id="email" name="email" placeholder="Enter your email" required>
        </div>
        <div class="inputbox">
            <input type="number" id="number" name="number" placeholder="Enter your phone number" required>
        </div>
        <div class="inputbox">
            <input type="password" id="password" name="password" placeholder="Enter your password" required>
        </div>
        <div class="inputbox">
            <input type="password" id="cpassword" name="cpassword" placeholder="Confirm your password" required>
        </div>
        <div class="inputbox">
            <select id="user_type" name="user_type" required>
                <option value="">Select user type</option>
                <option value="user">User</option>
                <option value="admin">Admin</option>
            </select>
        </div>
        <input type="submit" value="Register Now" name="submit" />
        <div class="links">
            Already have an account?
            <a href="login.php">Login now</a>
        </div>
    </form>
</div>
</section>

```

```

<?php include 'footer.php'; ?>

<script>
function validateForm() {
    const name = document.getElementById('name').value.trim();
    const email = document.getElementById('email').value.trim();
    const number = document.getElementById('number').value.trim();
    const password = document.getElementById('password').value;
    const cpassword = document.getElementById('cpassword').value;
    const user_type = document.getElementById('user_type').value;

    // Name validation: only letters and spaces, min length 3
    if (!/^[a-zA-Z\s]{3,}$/.test(name)) {
        alert("Name must be at least 3 characters and contain only letters and spaces!");
        return false;
    }

    // Email validation: basic pattern for .com emails
    if (!/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.(com)$/.test(email)) {
        alert("Email must be a valid .com address (e.g., example@gmail.com)!");
        return false;
    }

    // Phone number validation: exactly 10 digits
    if (!/^d{10}$/.test(number)) {
        alert("Phone number must be exactly 10 digits!");
        return false;
    }

    // Password match check
    if (password !== cpassword) {
        alert("Passwords do not match");
        return false;
    }

    // User type must be selected
    if (user_type === "") {
        alert("Please select user type");
        return false;
    }
}

```

```

// All validations passed
return true;
}

// Hide error messages after 5 seconds
setTimeout(() => {
  document.querySelectorAll('.message').forEach(msg => {
    msg.style.display = 'none';
  });
}, 5000);
</script>

</body>
</html>

```

LOGIN PAGE

```

<?php
include 'config.php';
session_start();

if (isset($_POST['submit'])) {
  $email = mysqli_real_escape_string($conn, $_POST['email']);
  $password = $_POST['password'];

  $select_users = mysqli_query($conn, "SELECT * FROM `register` WHERE email='$email' LIMIT 1") or die('query failed');

  if (mysqli_num_rows($select_users) > 0) {
    $row = mysqli_fetch_assoc($select_users);

    if (password_verify($password, $row['password'])) {
      if ($row['user_type'] == 'admin') {
        $_SESSION['admin_name'] = $row['name'];
        $_SESSION['admin_email'] = $row['email'];
        $_SESSION['admin_id'] = $row['id'];
        header('location:admin_page.php');
        exit();
      }
    }
  }
}

```

```

} elseif ($row['user_type'] == 'user') {
    $_SESSION['user_name'] = $row['name'];
    $_SESSION['user_email'] = $row['email'];
    $_SESSION['user_id'] = $row['id'];
    header('location:home.php');
    exit();
}
} else {
    $message[] = 'Incorrect password!';
}
} else {
    $message[] = 'No account found with this email!';
}
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <link rel="stylesheet" href="style.css"> <!-- common css -->
    <style>
        .login-section{
            display: flex;
            justify-content: center;
            align-items: center;
            flex-direction: column;
            padding: 60px 20px;
        }
        .login-box{
            width: 100%;
            max-width: 400px;
            border: 1px solid #ddd;
            padding: 30px;
            text-align: center;
            border-radius: 8px;
            background: #fff;
        }
    </style>

```

```
    box-shadow: 0 4px 10px rgba(0,0,0,0.1);
}

.login-box h2{
    font-size: 22px;
    margin-bottom: 20px;
    font-weight: bold;
}

.login-box .inputbox{
    margin-bottom: 15px;
    text-align: left;
}

.login-box .inputbox input{
    width: 100%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    outline: none;
}

.login-box .links{
    display: flex;
    justify-content: space-between;
    font-size: 14px;
    margin-bottom: 20px;
}

/* Button style */

.product_btn {
    width: 100%;
    padding: 12px;
    background: #c02736ff;
    color: #fff;
    border: 2px solid #c02736ff;
    border-radius: 6px;
    cursor: pointer;
    font-weight: 600;
    transition: all 0.3s ease;
    white-space: nowrap;
    text-decoration: none;
    text-align: center;
}

.product_btn:hover {
    background: #ffff;
```

```

    color: #c02736ff;
    border: 2px solid #c02736ff;
}
/* Message style */
.message {
    position: relative;
    max-width: 400px;
    margin: 15px auto;
    padding: 15px 45px 15px 15px;
    border-radius: 5px;
    border: 1px solid #f44336;
    background-color: #ffcdd2;
    color: #b71c1c;
    font-weight: bold;
    text-align: left;
}
.message .close-btn {
    position: absolute;
    right: 10px;
    top: 50%;
    transform: translateY(-50%);
    color: #b71c1c;
    font-size: 20px;
    font-weight: bold;
    cursor: pointer;
    user-select: none;
}
</style>
</head>
<body>
<?php include 'user_header.php'; ?>

<section class="login-section">
<?php
if (isset($message)) {
    foreach ($message as $msg) {
        echo '<div class="message">
        '.$msg.'
        <span      class="close-btn"
        onclick="this.parentElement.style.display=\\'none\\\'>&times;</span>
        </div>';
    }
}

```

```

        }
    }
?>
<div class="login-box">
    <h2>LOGIN NOW</h2>
    <form action="" method="post">
        <div class="inputbox">
            <input type="email" name="email" placeholder="Enter your email" required>
        </div>
        <div class="inputbox">
            <input type="password" name="password" placeholder="Enter your password" required>
        </div>
        <div class="links">
            <a href="register.php">Create account</a>
            <a href="forgot_password.php">Forgot password?</a>
        </div>
        <input type="submit" value="Login" name="submit" class="product_btn" />
    </form>
</div>
</section>

<?php include 'footer.php'; ?>

<script>
    // Auto-hide messages after 5 seconds
    window.addEventListener('DOMContentLoaded', function() {
        setTimeout(() => {
            document.querySelectorAll('.message').forEach(msg => {
                msg.style.transition = 'opacity 0.5s';
                msg.style.opacity = '0';
                setTimeout(() => msg.style.display = 'none', 500);
            });
        }, 5000);
    });
</script>
</body>
</html>

```

SHOP PAGE

```

<?php
include 'config.php';
session_start();

$user_id = isset($_SESSION['user_id']) ? $_SESSION['user_id'] : null;

// ✅ Handle Add to Cart
if (isset($_POST['add_to_cart'])) {
    if (!$user_id) {
        header('Location: login.php');
        exit();
    }

    $pro_name = $_POST['product_name'];
    $pro_price = $_POST['product_price'];
    $pro_quantity = $_POST['product_quantity'];
    $pro_image = $_POST['product_image'];

    $check = mysqli_query($conn, "SELECT * FROM `cart` WHERE name='$pro_name' AND user_id='$user_id'"") or die('query failed');

    if (mysqli_num_rows($check) > 0) {
        $message[] = 'Already added to cart!';
    } else {
        mysqli_query($conn, "INSERT INTO `cart`(user_id, name, price, quantity, image)
VALUES ('$user_id', '$pro_name', '$pro_price', '$pro_quantity',

```

```

'$pro_image")'

        or die('query2 failed');

$message[] = 'Product added to cart!';

}

}

//  Category Filter

$categoryFilter = "";

if (isset($_GET['category']) && $_GET['category'] !== "") {

    $categoryFilter = mysqli_real_escape_string($conn, $_GET['category']);

}

if ($categoryFilter !== "") {

    $select_products = mysqli_query($conn,
        "SELECT * FROM `products` WHERE category='$categoryFilter' ORDER BY id
DESC"
    ) or die('query failed');

} else {

    $select_products = mysqli_query($conn,
        "SELECT * FROM `products` ORDER BY id DESC"
    ) or die('query failed');

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Shop Page</title>

```

```

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

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

<style>

.msg-box {
    margin: 15px auto;
    max-width: 600px;
    padding: 12px;
    border-radius: 6px;
    text-align: center;
    font-weight: bold;
    animation: fadeOut 5s forwards;
}

.msg-success { background: #d4edda; color: #155724; border: 1px solid #c3e6cb; }

.msg-error { background: #f8d7da; color: #721c24; border: 1px solid #f5c6cb; }

@keyframes fadeOut {
    0% { opacity: 1; }
    80% { opacity: 1; }
    100% { opacity: 0; display: none; }
}

.category-bar {
    max-width: 1200px;
    margin: 20px auto;
    padding: 12px 16px;
    border: 1px solid #111;
}

```

```

background: #fafafa;
font-size: 16px;
}

.category-bar strong { color: #111; }

.category-bar a { text-decoration: underline; color: #721c24; margin-left: 10px; }

/* ✅ Product Grid */

.products_cont {
  max-width: 1200px;
  margin: auto;
  padding: 30px 16px;
}

.pro_box_cont {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(260px, 1fr));
  gap: 20px;
}

.pro_box {
  background: #fff;
  border: 1px solid #ddd;
  border-radius: 8px;
  padding: 15px;
  text-align: center;
  transition: 0.3s;
  position: relative;
}

.pro_box:hover {
  transform: translateY(-5px);
  box-shadow: 0px 6px 15px rgba(0,0,0,0.15);
}

```

```
}
```

```
.pro_box img {
```

```
    width: 100%;
```

```
    height: 220px;
```

```
    object-fit: contain;
```

```
    margin-bottom: 10px;
```

```
}
```

```
.pro_box h3 {
```

```
    font-size: 20px;
```

```
    font-weight: 600;
```

```
    margin: 10px 0 4px;
```

```
    color: #111;
```

```
}
```

```
.pro_box .category {
```

```
    font-size: 14px;
```

```
    color: #666;
```

```
    margin-bottom: 10px;
```

```
}
```

```
.price-tag {
```

```
    position: absolute;
```

```
    top: 10px;
```

```
    left: 10px;
```

```
    background: #c02736ff;
```

```
    color: #fff;
```

```
    padding: 5px 10px;
```

```
    border-radius: 6px;
```

```
    font-weight: 600;
```

```
    font-size: 15px;
```

```
}
```

```
/* ✅ Quantity & Button in one line */
```

```
.action-row {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    gap: 10px;  
}  
  
.action-row input[type=number] {  
    width: 70px;  
    padding: 6px;  
    text-align: center;  
    border: 2px solid #c02736ff;  
    border-radius: 6px;  
    font-weight: 600;  
}  
  
.action-row .product_btn {  
    padding: 10px 22px;  
    background: #c02736ff;  
    color: #fff;  
    border: 2px solid #c02736ff;  
    border-radius: 6px;  
    cursor: pointer;  
    font-weight: 600;  
    transition: all 0.3s ease;  
    white-space: nowrap;  
}  
  
.action-row .product_btn:hover {  
    background: #fff;
```

```

color: #c02736ff;
border: 2px solid #c02736ff;
}

</style>

</head>

<body>

<?php include 'user_header.php'; ?>

<!-- ✅ Message Show -->

<?php
if (isset($message)) {
    foreach ($message as $msg) {
        if ($msg == "Already added to cart!") {
            echo '<div class="msg-box msg-error">'.$msg.'</div>';
        } else {
            echo '<div class="msg-box msg-success">'.$msg.'</div>';
        }
    }
}
?>

<!-- ✅ Category Filter Bar -->

<?php if ($categoryFilter !== ""): ?>
<div class="category-bar">
    Showing category: <strong><?php echo htmlspecialchars($categoryFilter); ?></strong>
    <a href="shop.php">Clear</a>
</div>
<?php endif; ?>

```

```

<!-- ✅ Products Section -->

<section class="products_cont">
  <div class="pro_box_cont">
    <?php
      if (mysqli_num_rows($select_products) > 0) {
        while ($fetch_products = mysqli_fetch_assoc($select_products)) {
          ?>
          <form action="" method="post" class="pro_box">
            <span class="price-tag">₹<?php echo $fetch_products['price']; ?>/-</span>
            
            <h3><?php echo $fetch_products['name']; ?></h3>
            <div class="category">Category: <?php echo ucfirst($fetch_products['category']); ?></div>
          ?></div>

          <div class="action-row">
            <input type="number" name="product_quantity" min="1" value="1">
            <input type="hidden" name="product_name" value="<?php echo $fetch_products['name']; ?>">
            <input type="hidden" name="product_price" value="<?php echo $fetch_products['price']; ?>">
            <input type="hidden" name="product_image" value="<?php echo $fetch_products['image']; ?>">
            <input type="submit" value="Add to Cart" name="add_to_cart" class="product_btn">
          </div>
        </form>
      <?php
      }
    } else {
  
```

```
echo '<p class="empty">No Products Found!</p>';

}

?>

</div>

</section>

<?php include 'footer.php'; ?>

<script src="https://kit.fontawesome.com/eedbcd0c96.js"
crossorigin="anonymous"></script>

<script src="script.js"></script>

</body>

</html>
```

ABOUT PAGE

```
<?php
include 'config.php';
session_start();

if(isset($_SESSION['user_id'])){
    $user_id = $_SESSION['user_id'];
} else{
    $user_id = "";
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
```

```
width, initial-scale=1.0">
<title>About Us | Online Book Store</title>

<link rel="stylesheet"
href="https://unpkg.com/swiper@8/swiper-
bundle.min.css" />
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">
<link rel="stylesheet" href="style.css">

<style>
/* --- General Styling --- */
.about {
    padding: 60px 20px;
    max-width: 1200px;
    margin: auto;
}
.about .row {
    display: flex;
    align-items: center;
    flex-wrap: wrap;
    gap: 40px;
}
.about .row .image {
    flex: 1 1 45%;
    text-align: center;
}
.about .row .image img {
    max-width: 100%;
    border-radius: 12px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}
.about .row .content {
    flex: 1 1 45%;
}
.about .row .content h3 {
    font-size: 28px;
    margin-bottom: 15px;
    font-weight: bold;
    color: #333;
```

```

        }
.about .row .content p {
    margin-bottom: 10px;
    font-size: 16px;
    color: #444;
    line-height: 1.6;
}
.about .btn {
    display: inline-block;
    margin-top: 15px;
    padding: 10px 20px;
    background: #c02736;
    color: #fff;
    border-radius: 6px;
    text-decoration: none;
    transition: 0.3s;
}
.about .btn:hover {
    background: #fff;
    color: #c02736;
    border: 2px solid #c02736;
}
/* --- Simple Steps Section --- */
.steps {
    padding: 60px 20px;
    max-width: 1200px;
    margin: auto;
    text-align: center;
}
.steps h1 {
    font-size: 28px;
    margin-bottom: 30px;
    font-weight: bold;
    text-transform: uppercase;
}
.steps .box-container {
    display: grid;
    grid-template-columns: repeat(auto-fit,
    minmax(280px, 1fr));
    gap: 25px;
}

```

```

        }
.steps .box {
    background: #fff;
    padding: 25px;
    border: 1px solid #ddd;
    border-radius: 10px;
    box-shadow: 0 6px 12px rgba(0,0,0,0.1);
    transition: 0.4s ease;
    position: relative;
    overflow: hidden;
}
.steps .box:hover {
    transform: translateY(-8px);
    box-shadow: 0 10px 18px rgba(0,0,0,0.15);
}
.icon-container {
    width: 100%;
    height: 120px;
    display: flex;
    align-items: center;
    justify-content: center;
    margin-bottom: 15px;
    position: relative;
}
/* Floating animation (used in Choose & Enjoy) */
.floating {
    height: 100px;
    transition: transform 0.3s ease, filter 0.3s ease;
    animation: floatUpDown 3s ease-in-out infinite;
}
@keyframes floatUpDown {
    0%, 100% { transform: translateY(0); }
    50% { transform: translateY(-10px); }
}
.steps .box:hover .floating {
    transform: scale(1.1) rotate(3deg);
    filter: drop-shadow(0 5px 8px rgba(0,0,0,0.15));
}
/* Bike Animation */
.bike {

```

```

height: 100px;
animation: moveBike 3s linear infinite;
}
@keyframes moveBike {
0% { transform: translateX(-20px); }
50% { transform: translateX(20px); }
100% { transform: translateX(-20px); }
}

/* --- Reviews Section --- */
.reviews {
padding: 60px 20px;
max-width: 1200px;
margin: auto;
text-align: center;
}
.reviews h1 {
font-size: 28px;
margin-bottom: 30px;
font-weight: bold;
text-transform: uppercase;
}
.reviews .slide {
background: #fff;
padding: 20px;
border: 1px solid #ddd;
border-radius: 8px;
box-shadow: 0 6px 12px rgba(0,0,0,0.1);
}
.reviews .slide img {
width: 80px;
height: 80px;
border-radius: 50%;
margin-bottom: 15px;
}
.reviews .stars {
color: #ffcc00;
margin-top: 8px;
}
</style>
</head>

```

```

<body>

<?php include 'user_header.php'; ?>

<!-- About Section -->
<section class="about">
    <div class="row">
        <div class="image" data-aos="fade-right">
            
        </div>
        <div class="content" data-aos="fade-left">
            <h3>Why Choose Us?</h3>
            <p>Welcome to <strong>Online Book Store</strong> — your one-stop destination for exploring, discovering, and purchasing books. We are passionate about spreading knowledge and joy through books.</p>
        </div>
    </div>
    <div class="features">
        <p><strong>📚 Vast Collection:</strong> Choose from a wide range of genres and authors.</p>
        <p><strong>⚡ Fast Delivery:</strong> Get your favorite books delivered quickly and safely.</p>
        <p><strong>🤝 Customer Support:</strong> Friendly service for a smooth book-buying experience.</p>
        <p><strong>💻 Easy Access:</strong> Browse, order, and enjoy books from the comfort of your home.</p>
    </div>
    <p>At Online Book Store, we believe every book opens up a new world. Start your reading journey with us today!</p>
    <a href="shop.php" class="btn">Browse Books</a>
    </div>
    </div>
</section>

```

```

<!-- ✨ Animated Simple Steps Section -->
<section class="steps">
  <h1>Simple Steps</h1>
  <div class="box-container">

    <!-- 📚 Choose Your Book -->
    <div class="box choose">
      <div class="icon-container">
        
      </div>
      <h3>Choose Your Book</h3>
      <p>Browse our collection and pick your favorite books with ease.</p>
    </div>

    <!-- 🚲 Fast Delivery -->
    <div class="box">
      <div class="icon-container">
        
      </div>
      <h3>Fast Delivery</h3>
      <p>Your selected books are packed with care and delivered safely.</p>
    </div>

    <!-- 📖 Enjoy Reading -->
    <div class="box enjoy">
      <div class="icon-container">
        
      </div>
      <h3>Enjoy Reading</h3>
      <p>Relax with your books and dive into a world of imagination.</p>
    </div>

  </div>
</section>

```

```

<!-- ✨ Customer Reviews Section -->
<section class="reviews">
  <h1>Customer Reviews</h1>
  <div class="swiper reviews-slider">
    <div class="swiper-wrapper">
      <div class="swiper-slide slide">
        
        <p>This bookstore has an amazing collection! I always find what I need.</p>
        <div class="stars">
          <i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star-half-alt"></i>
        </div>
        <h3>Rahul Sharma</h3>
      </div>
      <div class="swiper-slide slide">
        
        <p>Fast delivery and excellent quality of books. Highly recommended!</p>
        <div class="stars">
          <i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i>
        </div>
        <h3>Priya Mehta</h3>
      </div>
      <div class="swiper-slide slide">
        
        <p>I love the user-friendly website. Buying books has never been easier.</p>
        <div class="stars">
          <i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i><i class="fas fa-star"></i>
        </div>
        <h3>Arjun Verma</h3>
      </div>
    </div>
    <div class="swiper-pagination"></div>
  </div>
</section>

```

```
<?php include 'footer.php'; ?>

<script src="https://unpkg.com/swiper@8/swiper-
bundle.min.js"></script>
<script>
var swiper = new Swiper(".reviews-slider", {
    loop:true,
    grabCursor: true,
    spaceBetween: 20,
    pagination: {
        el: ".swiper-pagination",
        clickable:true,
    },
    breakpoints: {
        0: { slidesPerView: 1 },
        700: { slidesPerView: 2 },
        1024: { slidesPerView: 3 },
    },
});
</script>
</body>
</html>
```

CONTACT PAGE

```
<?php
include 'config.php';
session_start();

$user_id = isset($_SESSION['user_id']) ? $_SESSION['user_id'] : null;

if(!$user_id){
    header('location:login.php');
}

if(isset($_POST['send'])){


```

```

$name = mysqli_real_escape_string($conn, $_POST['name']);
$email = mysqli_real_escape_string($conn, $_POST['email']);
$number = $_POST['number'];
$msg = mysqli_real_escape_string($conn, $_POST['message']);

$select_message = mysqli_query($conn, "SELECT * FROM `message` WHERE name='$name' AND email='$email' AND number='$number' AND message='$msg'"") or die('query failed');

if(mysqli_num_rows($select_message)>0){
    $message[] = 'Message already sent!';
} else{
    mysqli_query($conn, "INSERT INTO `message`(user_id, name, email, number, message) VALUES('$user_id','$name','$email','$number','$msg')") or die('query failed');
    $message[] = 'Message sent successfully!';
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Us | Book Store</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <link rel="stylesheet" href="style.css">
    <style>
        /* .heading {
            background: #222;
            padding: 30px 20px;
            text-align: center;
            color: #fff;
        } */
    </style>

```

```
.heading h3 {  
    font-size: 32px;  
    margin-bottom: 8px;  
}  
.heading p {  
    font-size: 16px;  
    color: #ddd;  
}  
.heading p a {  
    color: #c02736;  
    text-decoration: none;  
} */  
  
.contact {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    flex-wrap: wrap;  
    padding: 50px 20px;  
    max-width: 1200px;  
    margin: auto;  
    gap: 40px;  
}  
.contact .image {  
    flex: 1 1 45%;  
    text-align: center;  
}  
.contact .image img {  
    max-width: 100%;  
}  
.contact form {  
    flex: 1 1 45%;  
    background: #fff;  
    padding: 25px;  
    border: 1px solid #ddd;  
    border-radius: 8px;
```

```
    box-shadow: 0px 6px 12px rgba(0,0,0,0.1);
}

.contact form h3 {
    text-align: center;
    margin-bottom: 15px;
    font-size: 22px;
    font-weight: bold;
    color: #333;
}

.contact form .box {
    width: 100%;
    padding: 12px;
    border: 1px solid #ccc;
    border-radius: 6px;
    margin: 10px 0;
    font-size: 16px;
}

.contact form textarea {
    resize: none;
    height: 120px;
}

.contact form .btn {
    display: block;
    width: 100%;
    padding: 12px;
    background: #c02736;
    color: #fff;
    border: none;
    font-weight: bold;
    border-radius: 6px;
    cursor: pointer;
    transition: 0.3s;
}

.contact form .btn:hover {
    background: #fff;
    color: #c02736;
```

```

        border: 2px solid #c02736;
    }
</style>
</head>
<body>

<?php include 'user_header.php'; ?>

<!-- <div class="heading">
    <h3>Contact Us</h3>
    <p><a href="home.php">Home</a> / Contact</p>
</div> -->

<section class="contact">

    <div class="image">
        
    </div>

    <form action="" method="post">
        <h3>Tell Us Something!</h3>
        <input type="text" name="name" maxlength="50" class="box"
placeholder="Enter your name" required>
        <input type="text" name="number" maxlength="10" class="box"
placeholder="Enter your number" required>
        <input type="email" name="email" maxlength="50" class="box"
placeholder="Enter your email" required>
        <textarea name="message" class="box" required placeholder="Enter your
message" maxlength="500"></textarea>
        <input type="submit" value="Send Message" name="send" class="btn">
    </form>

</section>

<?php include 'footer.php'; ?>

```

```
<script src="https://kit.fontawesome.com/eedbcd0c96.js"
crossorigin="anonymous"></script>
</body>
</html>
```

USER ORDER PAGE

```
<?php
include 'config.php';
session_start();

if (!isset($_SESSION['user_id'])) {
    header('location:login.php');
    exit();
}

$user_id = $_SESSION['user_id'];
date_default_timezone_set('Asia/Kolkata');

// Handle cancel request
if (isset($_POST['cancel_order'])) {
    $order_id = intval($_POST['order_id']);

    // Check status before cancelling
    $check_status = mysqli_query($conn, "SELECT delivery_status FROM `orders` WHERE id='$order_id' AND user_id='$user_id'"") or die('query failed');

    if (mysqli_num_rows($check_status) > 0) {
        $order_data = mysqli_fetch_assoc($check_status);
        if ($order_data['delivery_status'] != 'delivered' && $order_data['delivery_status'] !=
```

```

'cancelled') {

    mysqli_query($conn, "UPDATE `orders` SET delivery_status='cancelled' WHERE
id='$order_id'" or die('query failed');

        echo     "<script>alert('Your      order      has      been      cancelled.');
window.location.href='orders.php';</script>";

        exit();

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Your Orders</title>




<!-- keep same css links as 1st code so cart/profile show correctly -->

    <link      rel="stylesheet"      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css"      integrity="sha512-
iecdLmaskl7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcenmpD1ypASozpmT/E0iPtmFIB46Zmdt
Ac9eNBvH0H/ZpiBw==" crossorigin="anonymous" referrerpolicy="no-referrer" />




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

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




<style>

/* Orders Page Scoped CSS */

.orders-page {
    max-width: 1000px;

```

```
margin: 40px auto;  
padding: 0 16px;  
}  
  
.orders-page h1.title {  
text-align: center;  
font-size: 26px;  
margin-bottom: 25px;  
font-weight: bold;  
}  
  
.orders-page .order-box {  
background: #fff;  
border: 1px solid #ddd;  
border-radius: 10px;  
padding: 20px;  
margin-bottom: 25px;  
box-shadow: 0 3px 8px rgba(0,0,0,0.08);  
overflow: hidden;  
}  
  
.orders-page .order-box p { margin: 6px 0; font-size: 15px; }  
.orders-page .order-box span { font-weight: bold; }  
  
/* Timeline */  
  
.orders-page .order-timeline {  
position: relative;  
display: flex;  
justify-content: space-between;  
align-items: center;  
margin: 30px 10px;  
}
```

```
.orders-page .order-timeline::before {  
    content: "";  
    position: absolute;  
    top: 20px;  
    left: 0;  
    right: 0;  
    height: 4px;  
    background: #ccc;  
    z-index: 1;  
}  
  
.orders-page .order-progress {  
    position: absolute;  
    top: 20px;  
    left: 0;  
    height: 4px;  
    background: #4CAF50;  
    z-index: 2;  
    transition: width 0.4s ease;  
}  
  
.orders-page .order-step {  
    position: relative;  
    text-align: center;  
    flex: 1;  
    z-index: 3;  
}  
  
.orders-page .order-circle {  
    width: 40px;  
    height: 40px;  
    line-height: 40px;
```

```

border-radius: 50%;

background: #ccc;
margin: 0 auto;
font-size: 18px;
display: flex;
justify-content: center;
align-items: center;
}

.orders-page .order-step.active .order-circle { background: #4CAF50; color: #fff; }

.orders-page .order-label { margin-top: 8px; font-size: 14px; color: #444; }

.orders-page .order-step.active .order-label { color: #4CAF50; }

/* Vehicle */

.orders-page .vehicle img {
width: 30px;
height: 30px;
}

.orders-page .vehicle.moving img {
animation: moveVehicle 0.8s ease-in-out infinite alternate;
}

@keyframes moveVehicle {
0% { transform: translateX(-5px); }
100% { transform: translateX(5px); }
}

/* Cancelled order */

.orders-page .cancelled .order-circle { background-color: red; color: white; }

.orders-page .cancelled .order-label { color: red; }

```

```

/* Buttons */

.orders-page .order-btn {
    display: inline-block;
    margin-top: 15px;
    padding: 10px 20px;
    background: #c02736;
    color: #fff;
    border-radius: 6px;
    text-decoration: none;
    font-weight: bold;
    border: none;
    cursor: pointer;
}

.orders-page .order-btn:hover { background: #fff; color: #c02736; border: 2px solid #c02736; }

</style>
</head>
<body>

<?php include 'user_header.php'; ?>

<section class="orders-page">
    <h1 class="title">Your Orders</h1>
    <?php
        $select_orders = mysqli_query($conn, "SELECT * FROM `orders` WHERE user_id='$user_id' ORDER BY id DESC") or die('query failed');
        if(mysqli_num_rows($select_orders) > 0){
            while($order = mysqli_fetch_assoc($select_orders)){
                $status = $order['delivery_status'];
            }
        }
    </?php
</section>

```

```

$progressWidth = 0;
if ($status == 'on the way') $progressWidth = 50;
if ($status == 'delivered') $progressWidth = 100;
?>

<div class="order-box">
    <p>Placed on : <span><?= $order['placed_on']; ?></span></p>
    <p>Name : <span><?= $order['name']; ?></span></p>
    <p>Email : <span><?= $order['email']; ?></span></p>
    <p>Number : <span><?= $order['number']; ?></span></p>
    <p>Address : <span><?= $order['address']; ?></span></p>
    <p>Payment method : <span><?= $order['method']; ?></span></p>
    <p>Your orders : <span><?= $order['total_products']; ?></span></p>
    <p>Total price : <span>₹<?= $order['total_price']; ?>/-</span></p>
    <p>Payment status :
        <span style="color:<?= $order['payment_status'] == 'pending' ? 'red' : 'green'; ?>">
            <?= ucfirst($order['payment_status']); ?>
        </span>
    </p>
    <p>Delivery status :
        <span style="color:
            <?= $status == 'delivered' ? 'green' :
                ($status == 'on the way' ? 'orange' :
                    ($status == 'cancelled' ? 'red' : 'gray')); ?>">
            <?= ucfirst($status); ?>
        </span>
    </p>
<!-- Timeline -->
<?php if ($status == 'cancelled'): ?>

```

```

<div class="order-timeline">
    <div class="order-step cancelled">
        <div class="order-circle">X</div>
        <div class="order-label">Cancelled</div>
    </div>
</div>

<?php else: ?>
    <div class="order-timeline">
        <div class="order-progress" style="width: <?= $progressWidth; ?>%;"></div>

        <!-- Step 1 -->
        <div class="order-step active">
            <div class="order-circle">✓</div>
            <div class="order-label">Order Placed</div>
        </div>

        <!-- Step 2 -->
        <div class="order-step <?= in_array($status, ['on the way','delivered']) ? 'active' : "inactive'">
            <div class="order-circle vehicle <?= $status == 'on the way' ? 'moving' : " " ?>">
                
            </div>
            <div class="order-label">On the Way</div>
        </div>

        <!-- Step 3 -->
        <div class="order-step <?= $status == 'delivered' ? 'active' : "inactive'">
            <div class="order-circle">📦</div>
            <div class="order-label">Delivered</div>

```

```

        </div>
        </div>
<?php endif; ?>

<!-- Cancel Button --&gt;
&lt;?php if ($status != 'delivered' &amp;&amp; $status != 'cancelled'): ?&gt;
&lt;form action="" method="post"&gt;
&lt;input type="hidden" name="order_id" value="&lt;?= $order['id']; ?&gt;"&gt;
    &lt;button type="submit" name="cancel_order" class="order-btn" onclick="return
confirm('Are you sure you want to cancel this order?');"&gt;
        Cancel Order
    &lt;/button&gt;
&lt;/form&gt;
&lt;?php endif; ?&gt;

&lt;a href="invoice.php?order_id=&lt;?= $order['id']; ?&gt;" class="order-btn"
target="_blank"&gt;Download Invoice&lt;/a&gt;
&lt;/div&gt;
&lt;?php
    }
} else {
    echo '&lt;p class="empty"&gt;No orders placed yet!&lt;/p&gt;';
}
?&gt;
&lt;/section&gt;

&lt;?php include 'footer.php'; ?&gt;
&lt;script
src="https://kit.fontawesome.com/eedbcd0c96.js"
crossorigin="anonymous"&gt;&lt;/script&gt;
</pre>

```

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

ADMIN DASHBOARD PAGE

```
<?php
include 'config.php';
session_start();

// ✅ Ensure only admin can access
if (!isset($_SESSION['admin_id'])) {
    header('location:login.php');
    exit;
}

$admin_id = $_SESSION['admin_id'];

// ✅ Fetch Admin Profile
$fetch_profile = null;
$select_profile = mysqli_query($conn, "SELECT * FROM `register` WHERE id = '$admin_id' AND user_type = 'admin' LIMIT 1") or die('query failed');
if (mysqli_num_rows($select_profile) > 0) {
    $fetch_profile = mysqli_fetch_assoc($select_profile);
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Admin Dashboard</title>

<!-- Font Awesome -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<!-- Custom CSS -->
<link rel="stylesheet" href="admin.css">
<link rel="stylesheet" href="style.css">

<style>
body {
    background: #f5f5f5;
    font-family: 'Poppins', sans-serif;
}
.dashboard {
    padding: 100px 20px 50px;
    text-align: center;
}
.dashboard h1 {
    font-size: 28px;
    font-weight: 700;
    margin-bottom: 30px;
    text-transform: uppercase;
    color: #222;
}
.box-container {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 25px;
    justify-content: center;
    align-items: stretch;
    max-width: 1200px;
    margin: 0 auto;
}
```

```
}
```

```
.box {
```

```
    background: #fff;
```

```
    border: 2px solid #000;
```

```
    border-radius: 10px;
```

```
    padding: 25px;
```

```
    box-shadow: 0 4px 12px rgba(0,0,0,0.1);
```

```
    transition: transform 0.3s ease, box-shadow 0.3s ease;
```

```
}
```

```
.box:hover {
```

```
    transform: translateY(-5px);
```

```
    box-shadow: 0 6px 16px rgba(0,0,0,0.15);
```

```
}
```

```
.box h3 {
```

```
    font-size: 22px;
```

```
    color: #c02736;
```

```
    margin-bottom: 10px;
```

```
}
```

```
.box p {
```

```
    font-size: 15px;
```

```
    color: #444;
```

```
    margin-bottom: 15px;
```

```
}
```

```
.box a.btn {
```

```
    display: inline-block;
```

```
    padding: 8px 16px;
```

```
    background: #c02736;
```

```
    color: #fff;
```

```
    font-weight: 600;
```

```
    border-radius: 6px;
```

```
    transition: 0.3s;
```

```
    text-decoration: none;
```

```
}
```

```
.box a.btn:hover {
```

```
    background: #fff;
```

```
    color: #c02736;
```

```

    border: 2px solid #c02736;
}
</style>
</head>

<body>

<?php include 'admin_header.php'; ?>

<section class="dashboard">
<h1>Admin Dashboard</h1>

<div class="box-container">

<!-- ✅ Admin Profile -->
<div class="box">
<h3>Welcome!</h3>
<?php if ($fetch_profile): ?>
<p><?= htmlspecialchars($fetch_profile['name']); ?></p>
<?php else: ?>
<p>Profile not found</p>
<?php endif; ?>
<a href="admin_update.php" class="btn">Update Profile</a>
</div>

<!-- ✅ Pending Payments -->
<div class="box">
<?php
    $total_pendings = 0;
    $select_pending = mysqli_query($conn, "SELECT total_price FROM `orders` WHERE payment_status = 'pending'") or die('query failed');
    while ($row = mysqli_fetch_assoc($select_pending)) {
        $total_pendings += $row['total_price'];
    }
?>
<h3>₹<?= number_format($total_pendings); ?></h3>

```

```

<p>Total Payments Pending</p>
<a href="admin_orders.php" class="btn">View Orders</a>
</div>

<!-- ✅ Completed Payments -->
<div class="box">
<?php
$total_completed = 0;
$select_completed = mysqli_query($conn, "SELECT total_price FROM `orders` WHERE payment_status = 'completed'") or die('query failed');
while ($row = mysqli_fetch_assoc($select_completed)) {
    $total_completed += $row['total_price'];
}
?>
<h3>₹<?= number_format($total_completed); ?></h3>
<p>Completed Payments</p>
<a href="admin_orders.php" class="btn">View Orders</a>
</div>

<!-- ✅ Orders -->
<div class="box">
<?php
$select_orders = mysqli_query($conn, "SELECT * FROM `orders`") or die('query failed');
$number_of_orders = mysqli_num_rows($select_orders);
?>
<h3><?= $number_of_orders; ?></h3>
<p>Orders Placed</p>
<a href="admin_orders.php" class="btn">View Orders</a>
</div>

<!-- ✅ Products -->
<div class="box">
<?php
$select_products = mysqli_query($conn, "SELECT * FROM `products`") or die('query failed');

```

```

$number_of_products = mysqli_num_rows($select_products);
?>
<h3><?= $number_of_products; ?></h3>
<p>Products Added</p>
<a href="admin_products.php" class="btn">View Products</a>
</div>

<!-- ✅ Registered Users -->
<div class="box">
<?php
    $select_users = mysqli_query($conn, "SELECT * FROM `register` WHERE user_type = 'user'" ) or die('query failed');
    $number_of_users = mysqli_num_rows($select_users);
?>
<h3><?= $number_of_users; ?></h3>
<p>Registered Users</p>
<a href="admin_users.php?type=user" class="btn">View Users</a>
</div>

<!-- ✅ Registered Admins -->
<div class="box">
<?php
    $select_admins = mysqli_query($conn, "SELECT * FROM `register` WHERE user_type = 'admin'" ) or die('query failed');
    $number_of_admins = mysqli_num_rows($select_admins);
?>
<h3><?= $number_of_admins; ?></h3>
<p>Registered Admins</p>
<a href="admin_users.php?type=admin" class="btn">View Admins</a>
</div>

<!-- ✅ Messages -->
<div class="box">
<?php
    $select_messages = mysqli_query($conn, "SELECT * FROM `message`" ) or die('query failed');

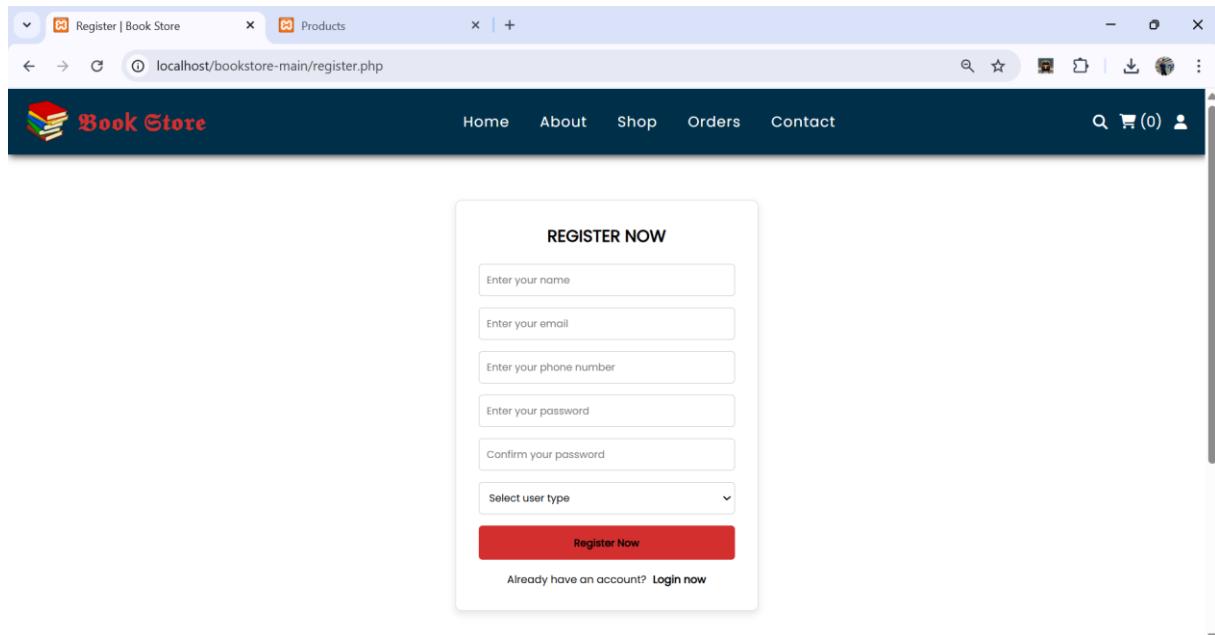
```

```
$number_of_messages = mysqli_num_rows($select_messages);  
?>  
<h3><?= $number_of_messages; ?></h3>  
<p>New Messages</p>  
<a href="admin_messages.php" class="btn">View Messages</a>  
</div>  
  
</div>  
</section>  
  
<script src="admin_js.js"></script>  
    <script src="https://kit.fontawesome.com/eedbcd0c96.js"  
crossorigin="anonymous"></script>  
</body>  
</html>
```

CHAPTER – 10

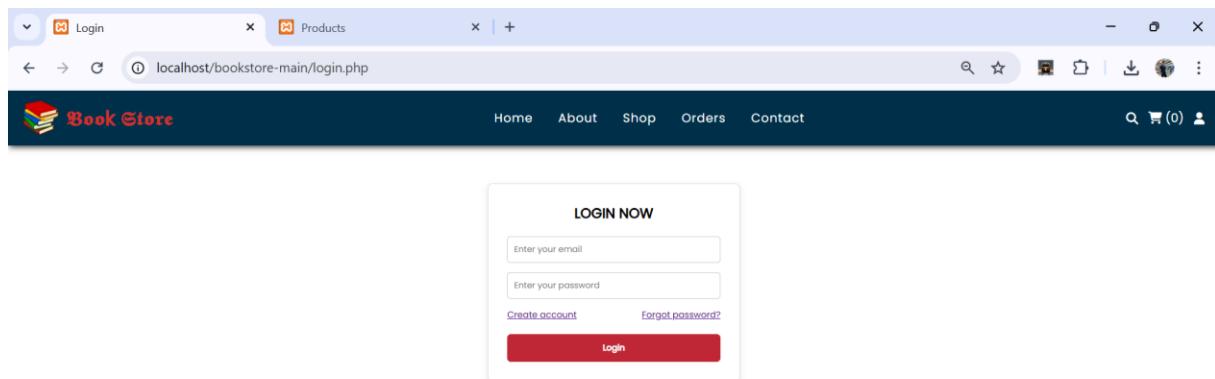
SNAPSHOT

User Register



A screenshot of a web browser showing the 'Register' page of a book store. The page has a white background with a light gray border. At the top center, it says 'REGISTER NOW'. Below that are six input fields: 'Enter your name', 'Enter your email', 'Enter your phone number', 'Enter your password', 'Confirm your password', and 'Select user type'. A red 'Register Now' button is at the bottom. Below the button, a link says 'Already have an account? [Login now](#)'.

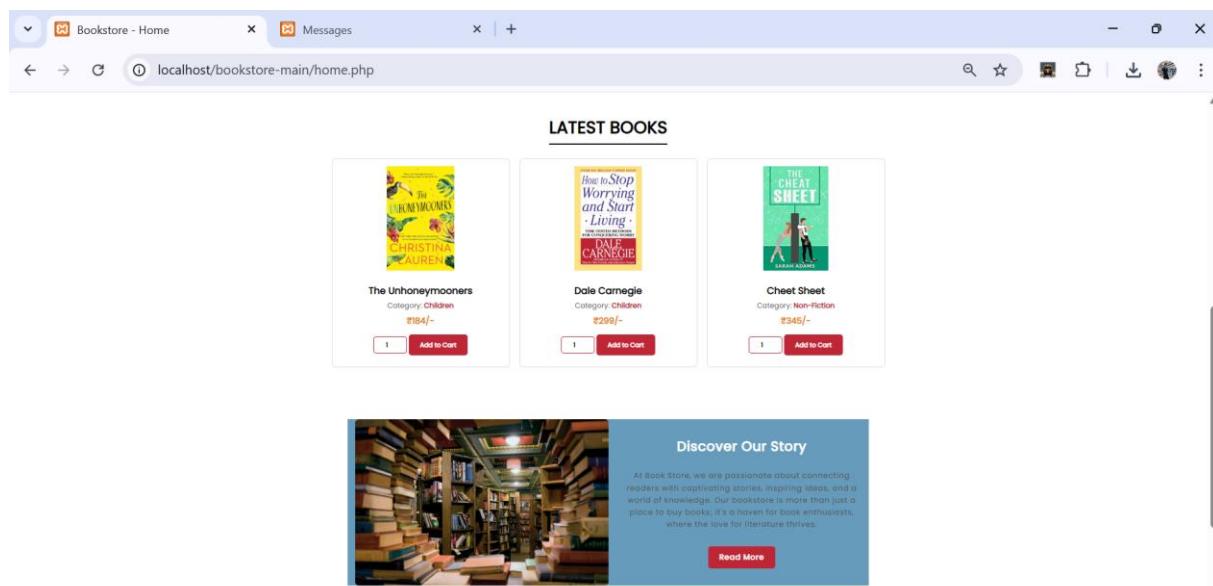
Login Page



A screenshot of a web browser showing the 'Login' page of a book store. The page has a white background with a light gray border. At the top center, it says 'LOGIN NOW'. Below that are two input fields: 'Enter your email' and 'Enter your password'. Below the fields are links for 'Create account' and 'Forgot password?'. A red 'Login' button is at the bottom. Below the button, a link says 'Already have an account? [Login now](#)'.



Home page



Admin Dashboard Page

The screenshot shows the Admin Dashboard page with the following data:

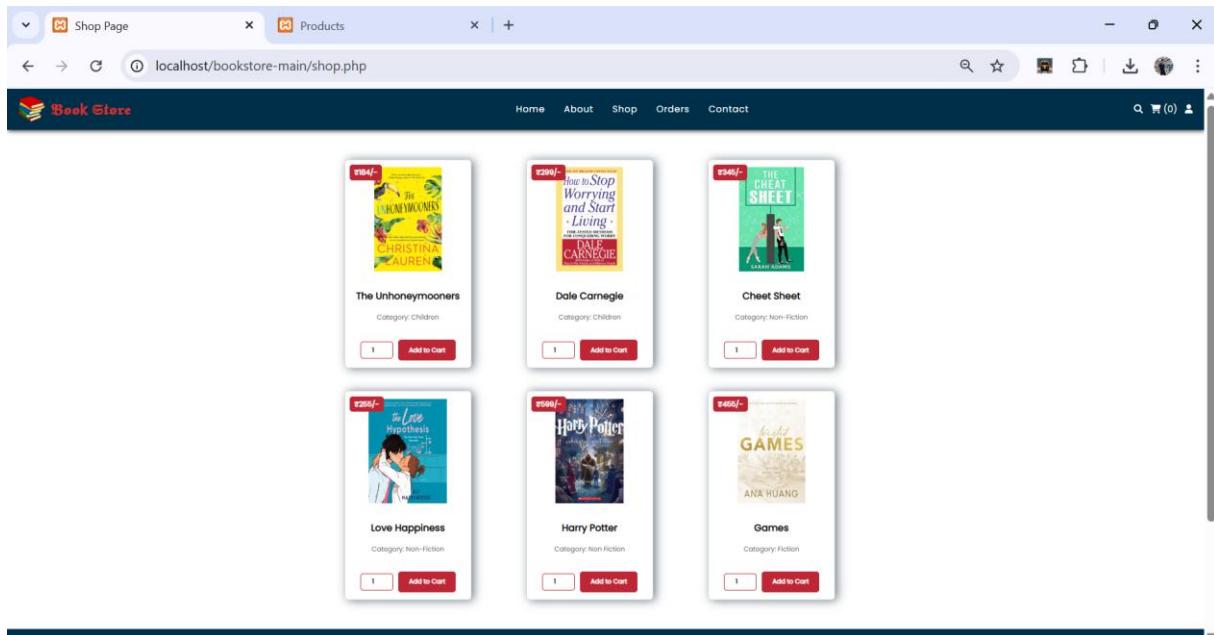
- Welcome!** BookStore [Update Profile](#)
- ₹5,846** Total Payments Pending [View Orders](#)
- ₹5,264** Completed Payments [View Orders](#)
- 18** Orders Placed [View Orders](#)
- 5** Products Added [View Products](#)
- 7** Registered Users [View Users](#)
- 2** Registered Admins [View Admins](#)
- 2** New Messages [View Messages](#)

Order Page

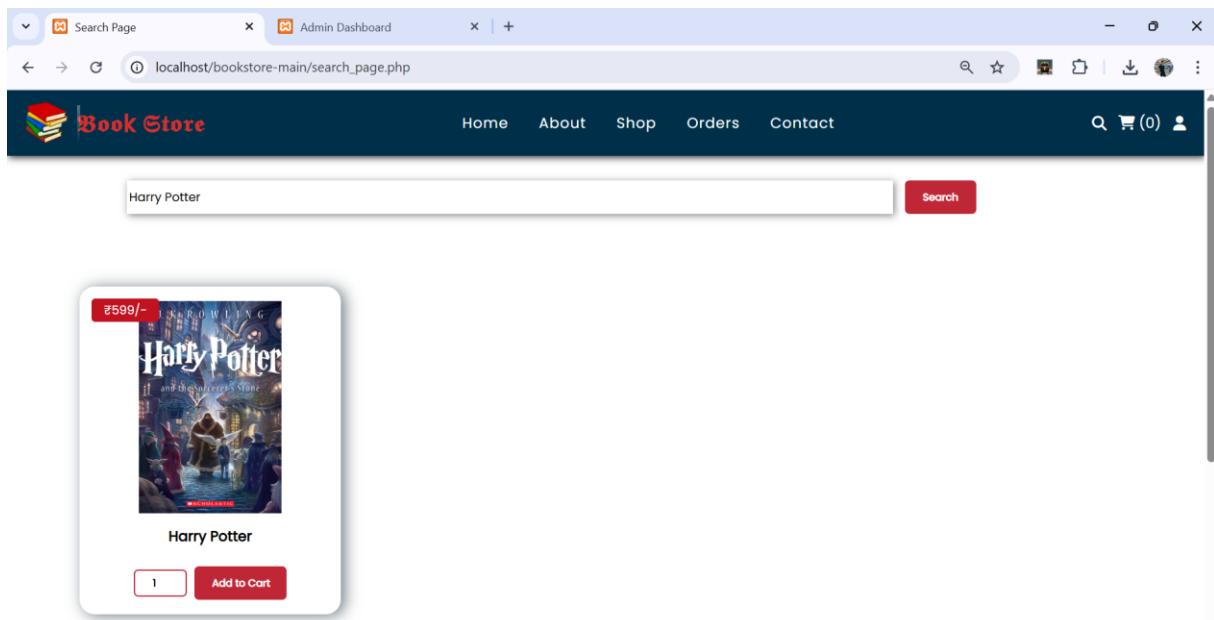
The screenshot shows the Order Page with the following data for each order:

Order Details	Status Options
Placed On : 2025-11-02 20:39:50 Name : Ramesh Email : raamk5499@gmail.com Number : 8974873445 Address : 85/1, 1st A cross 7th main road 2nd block Jayanagar,Bengaluru-560011 Your Books : Dale Carnegie (1) Total Price : ₹299/- Payment Method : paytm Payment Status : Completed Delivery Status : On the way	Order Placed On the Way Delivered <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Completed </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Payment </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> On the Way </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Delivery </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Download Invoice </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Delete </div>
Placed On : 2025-11-01 19:41:26 Name : Kundan Kumar Email : kundankumarcims204@gmail.com Number : 7217241543 Address : 85/1, 1st A cross 7th main road 2nd block Jayanagar,Bengaluru-560011 Your Books : Dale Carnegie (1), python (1), Cheet Sheet (1) Total Price : ₹345/- Payment Method : gpay Payment Status : Completed Delivery Status : Processing	Order Placed On the Way Delivered <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Completed </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Payment </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Processing </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Delivery </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Download Invoice </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Delete </div>
Placed On : 2025-11-01 11:10:29 Name : Thippesh Email : thippesh20010@gmail.com Number : 7483618803 Address : 85/1, 1st A cross 7th main road 2nd block Jayanagar,Bengaluru-560011 Your Books : Dale Carnegie (1), python (1), Cheet Sheet (1) Total Price : ₹988/- Payment Method : cash on delivery Payment Status : Completed Delivery Status : Delivered	Order Placed On the Way Delivered <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Completed </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Payment </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Delivered </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Delivery </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Download Invoice </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Delete </div>
Placed On : 2025-10-09 15:48:47 Name : Kundan Kumar Email : kundankumarcims204@gmail.com Number : 7217241546 Address : 85/1, 1st A cross 7th main road 2nd block Jayanagar,Bengaluru-560011 Your Books : Harry Potter (1) Total Price : ₹490/- Payment Method : gpay Payment Status : Completed Delivery Status : Delivered	Order Placed On the Way Delivered <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Completed </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Payment </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Delivered </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Delivery </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Download Invoice </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Delete </div>
Placed On : 2025-08-29 16:24:23 Name : Kundan Kumar Email : kundankumarcims204@gmail.com Number : 7217241546 Address : 85/1, 1st A cross 7th main road 2nd block Jayanagar,Bengaluru-560011 Your Books : Harry Potter (1) Total Price : ₹599/- Payment Method : gpay Payment Status : Completed Delivery Status : Cancelled	Order Placed On the Way Delivered <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Completed </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Payment </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Cancelled </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> Update Delivery </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Download Invoice </div> <div style="background-color: red; color: white; padding: 2px 10px; margin-top: 5px;"> Delete </div>

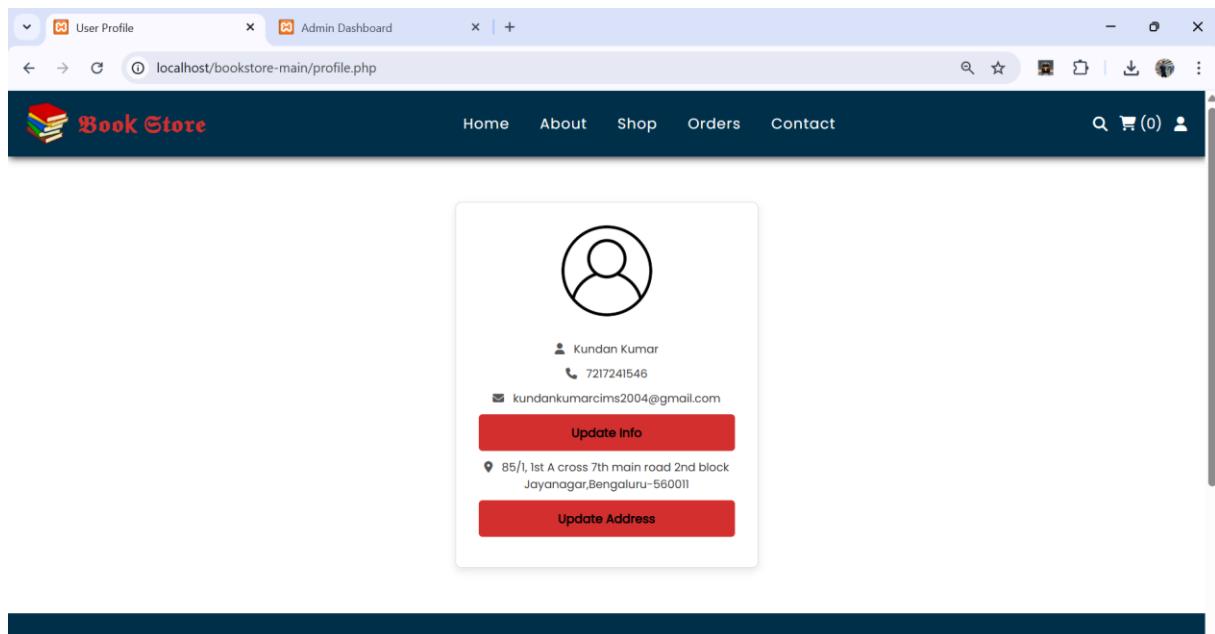
Shop Page



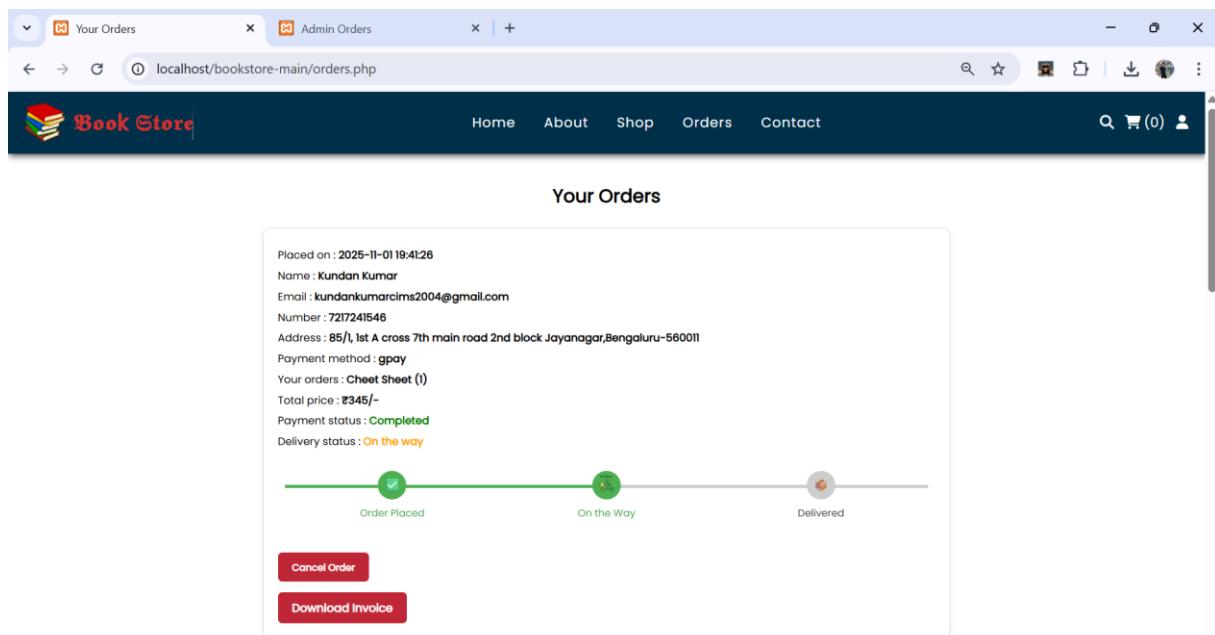
Search Book



User Profile



Order Tracking



About

Why Choose Us?

- Vast Collection:** Choose from a wide range of genres and authors.
- Fast Delivery:** Get your favorite books delivered quickly and safely.
- Customer Support:** Friendly service for a smooth book-buying experience.
- Easy Access:** Browse, order, and enjoy books from the comfort of your home.

At Online Book Store, we believe every book opens up a new world. Start your reading journey with us today!

[Browse Books](#)

SIMPLE STEPS

Choose Your Book
Browse our collection and pick your favorite books with ease.

Fast Delivery
Your selected books are packed with care and delivered safely.

Enjoy Reading
Relax with your books and dive into a world of imagination.

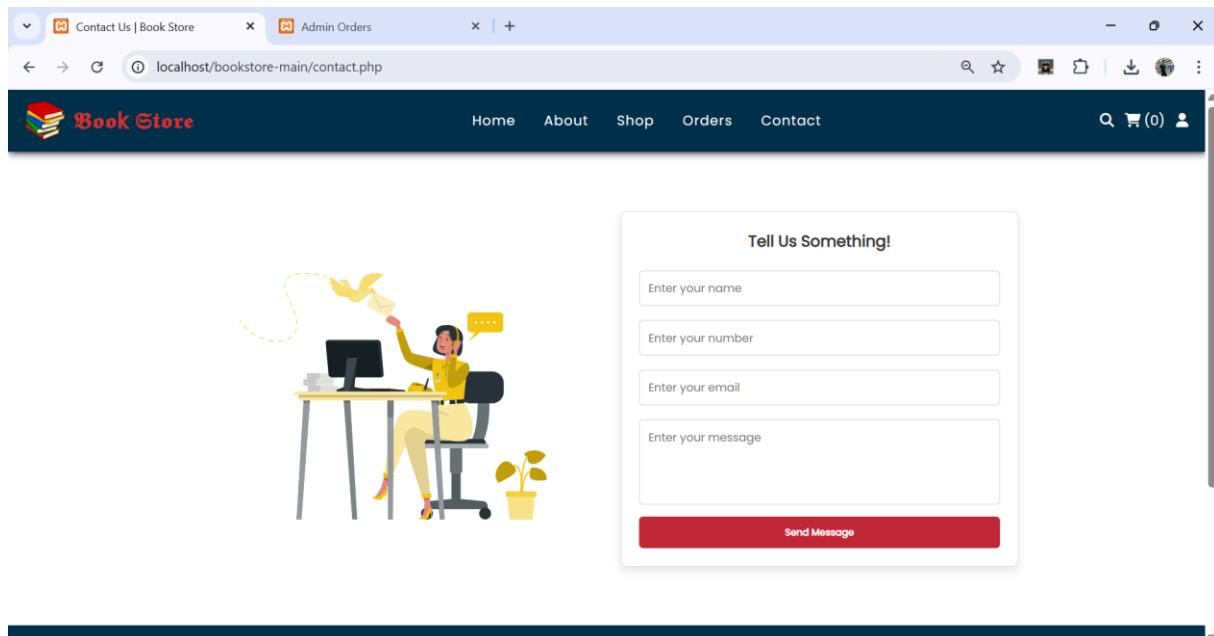
CUSTOMER REVIEWS

This bookstore has an amazing collection! I always find what I need.
★★★★★
Rahul Sharma

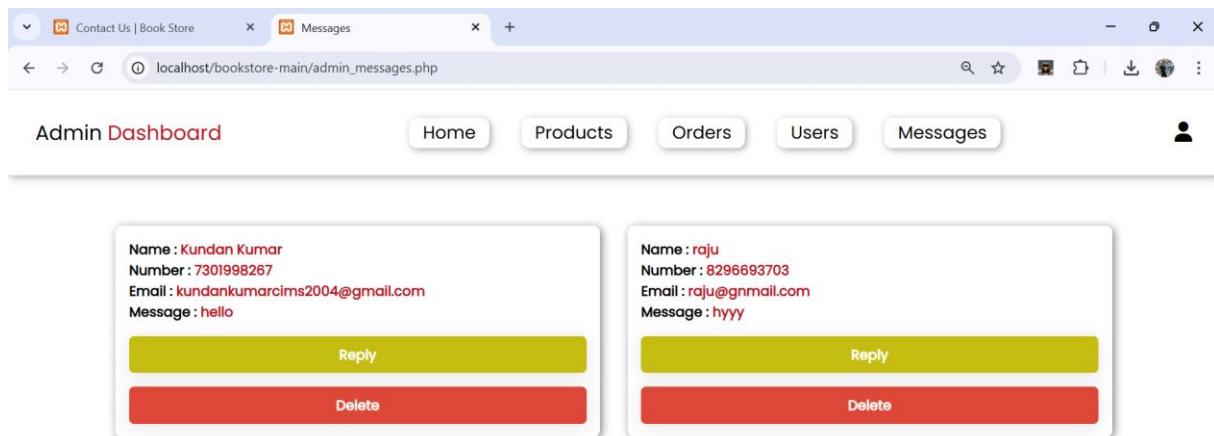
Fast delivery and excellent quality of books. Highly recommended!
★★★★★
Kundan Kumar

I love the user-friendly website. Buying books has never been easier.
★★★★★
Priya Verma

Contact



Messages



CHAPTER – 11

DEVELOPMENT TOOLS

10.1 Database Environment

The proposed system will need a database system to support in order to store the huge amount of data. MySQL database system will be chosen to support the proposed system because it is well-known open source relational database management system. Other than that, MySQL database system provides software developer with a client program with easy understanding graphical user interface that can communicate to the MySQL database server named as MySQL Workbench 6.3 CE. By doing so, software developer can interact with the database system with the user friendliness client program and without using the command prompt, at the end it may speed up the development progress.

10.2 Web Technology

The proposed system is a cross platform system, which include Windows desktop client and Android mobile phone client. Therefore, it requires to implement web services in order to support the Android mobile phone client device to fetch data from the database server and store data back to the database server. In this project, WampServer is used to support the issue that mention earlier, WampServer is a

Windows web development environment that built in comes together with the Apache, PHP and MySQL database. Furthermore, PHP will be the programming language that is used to create the web services that store in the server and it is able to call by the Android mobile phone client to perform particular task.

10.3 System Platform

The project is a cross platform-based system which includes Windows desktop client device and Android mobile phone client device in order to operate the proposed system. Therefore, iOS mobile phone client device will not be compatible with the proposed system.

8.4 Project Management Tool

For the rule of thumb to develop a system, it requires to have an intact system planning which serve as a guide line during the development phases. Microsoft Project is a project management software program and it has been used in the project to design the system planning, project timeline, managing of resources, tracking development progress and etc.

8.5 Visual Paradigm Community Edition

In this project, Visual Paradigm software has been used to create document for several system planning diagram such as Use Case diagram, Activity diagram, Class diagram and By doing so, software developer are able to visualize the system communicate information much more concisely and clearly.

8.6 Test Plan

After the system has been developed, it will move to system testing phase. In system testing phase, the developed system is required to install on appropriate devices for testing purpose.

After the system installation has been completed, the system testing task will be performed by different roles of user such as manager role and staff role. The purpose of system testing is to identify and determine the degree of system stability. At the same time, it is given an opportunity for developer to figure out error or bug that has not been raise and encounter during the system development phase. Those error or bug that has been found during the system testing activities will be solved before the system release. Each and every testing before system testing phases is actually tested by the system developer itself. Therefore, it might cause some biases toward the actually tested by the system developer itself. Therefore, it might cause some biases toward the testing due to the system developer have knowledge about the system software logics and lead the result to be inappropriate. There are four types of testing that will be used to test the developed system which includes unit testing, integration testing, system testing and acceptance testing

- Unit Testing**

First of all, unit testing will be the first testing method that used to test the developed system.

It consists of testing activities that test the system module by module which has not been integrated whole. By doing unit testing, developer is able to identify error and hug easily since it is finding the error and hog through a tit part of the system inther than finding error through the complete system. In addiñon, developer will test the unit part of the system with the validation and the correctness of data valoz. Valid and invalid input will he entering to test and ensure the system processes perform with an expected result.

- **Functional Testing**

After conduced the unit testing, fumetional testing will begin to test the developed system Functional testing is performed to verify that the system application processes that perform and functioning appropriately according to the design specifications. In functional testing, the core system application functions will be tested with several test cases in order to ensure that the entire system functioning as a whole and perform task that with the expected results.

CHAPTER – 12

TESTING

System testing of the software and hardware is a testing conducted on a system which is complete, integrated system that works as a whole. System testing is a critical testing procedure that must be conducted by software developer before the system released. During system testing it can evaluate the system's compliance with its specified requirements according to the system design. Furthermore, several testing activities in system testing test not only the design of the system, but also the behaviour and the believed expectations result from the customer. In addition, various complex test cases that used to test the system are according to the business process requirements which are collected from the user. Meanwhile, errors or bugs that detected during the testing is required software developer look into it from the initial step of the business process to the end of the process to ensure it have expected result in order to solve the errors or bugs to determine the degree of system stability.

9.1 Acceptance Testing

Last but not least, acceptance testing also known as user acceptance testing would be the final testing procedure that performs to test the developed software system. In acceptance testing, the testing activities are different compare to the testing activities that mentioned previously because the tester that tests the system will be the final user which do not have knowledge about the system logic. If the final user encountered an error while using the system, system developer are required to maintain the system as much as possible and release a new patch for the existing system to recover the error. Meanwhile, final user will use the system that visualized as to support their real business routine operation; therefore, software support team are required to stand by to provide technical support while final user need any help or support that regarding the system. If there are no errors detected by the final user while using the system for a long period, the development job of developer is considered as complete and the system will be a final system product.

CHAPTER – 13

SYSTEM EVALUATE AND DISCUSSION

10.1 Proposed System Completion

The proposed system is designed and developed to solve all the problem statements which are stated in chapter one of this report. First of all, the developed system provides a feature that is able to solve the problem of difficulties in food order ticket tracking and achieve the project objective of prevention of food serves not in sequence. By using the system, it allows staff to place order ticket through using the system and the system will automatically queue the food order information according to the first come first serve basis and kitchen staff is able to follow the food queue to serve customer accordingly. It also eliminates all the manual processes that involve in the traditional method of delivering food order ticket. In addition, the developed system allows manager to update all the food information as it is needed. This feature helps restaurant to eliminate duplicated physical menu card which contain misleading information and also allow staff and customer to view the latest updated food menu information through using the system. By doing so, it helps restaurant to solve the problem which regarding difficulties in updating menu card information, difficulties in providing appropriate updated food information and the potential of increase cost of operation as the system will automatically refresh all the updated information that is edited by manager. Furthermore, the project objective which regarding provides convenience for both employees and consumers has been achieved because the system allows consumer to view all the updated information through the mobile phone client devices and it reduces the number of manual work which restaurant staff need to be performed. Last but not least, the project objective of assisting restaurant to plan ahead has been achieved due to the system allowing manager to generate several types of report in order to assist the restaurant to plan ahead. By analysing the generated report, manager is able to carry out a planning for the next business routine of the restaurant in order to improve the restaurant operation efficiency. In a nutshell, the system has fulfilled all significant outcomes that according to all the problem statements and project objectives that are stated.

CHAPTER -14

SYSTEM STRENGTH AND LIMITATION

11.1 System Strength

The system provide customer with good dine in experience as it allow customer to view food information and place order through using the mobile phone device. The mobile application also provides ease of use because it has a simple understandable graphic user interface and minimal instruction to follow while placing order through the application. Next, it helps restaurant staff to serve their valued customer in minimal delay. Furthermore, the system can be implemented with low cost and is affordable for most of the small medium enterprise restaurant, because the hardware requirements are not required to be high end powerful in order to support the system. Lastly, server and client are communicating within the restaurant which is intranet therefore it does not need to have internet access.

11.2 System Limitation

The system is not compatible with IOS mobile devices as the mobile application is developed in Android environment. Therefore, for IOS mobile phone may not be able to install the application in their mobile phone and experience the system. Meanwhile, the restaurant has prepared some Android mobile devices which are used to resolve the issue mentioned above. Next, due to the mobile phone client devices are required to connect to the wireless intranet in order to communicate with the server. It is very important to identify a suitable location that is used to install and set up the wireless access point in order to ensure the wireless signal coverage is able to reach the entire restaurant area.

CHAPTER – 15

CONCLUSION

15.1 CONCLUSION

In conclusion, the advancement of technology over the past decade has significantly transformed how businesses operate, especially in the food and beverage (F&B) industry. The adoption of management systems has become a crucial trend, enabling restaurants to streamline operations, reduce manual labor, save time, and generate valuable reports for strategic planning. The proposed system in this project aims to enhance restaurant productivity and profitability by automating core business processes and minimizing operational costs. Obviously, the proposed system can help improve the productivity of the restaurant and thus directly impact the profitability of the restaurant. Furthermore, it can also help the restaurant to reduce the cost of operation in terms of manpower, because the system has already facilitated majority of the business process by using the system. Therefore, it is believed that the system can lead the restaurant's business grow from time to time.

Furthermore, the importance of portability in modern technology cannot be overlooked. With the increasing use of mobile devices, system accessibility and user convenience have become vital. Our proposed solution addresses this by integrating a mobile-based food ordering system with a computer-based restaurant management system. This combination allows customers to place orders seamlessly using smartphones or tablets, while restaurant staff can efficiently handle operations using the computer interface, which offers extended functionalities and hardware compatibility.

Overall, the integration of mobile and desktop platforms not only supports operational efficiency but also aligns with current user behavior and technological trends. This system is expected to support the continuous growth of restaurant businesses by providing a comprehensive, accessible, and user-friendly management solution.

CHAPTER – 16

FUTURE ENHANCEMENT

12.1 Future Enhancement

The Online Bookstore Management System has great potential for future enhancement. It can be improved with AI-based book recommendations, advanced sales analytics, and multi-branch management for larger operations.

Adding features like loyalty programs, courier integration, offline reading, and multi-language support can enhance user experience and efficiency. Expanding to e-books, audiobooks, and global markets will further increase its reach and growth potential.

- Mobile Application Development: A dedicated mobile app can allow users to browse, purchase, and manage their book orders conveniently from anywhere.
- AI-Based Personalization: Implementing AI to suggest books based on reading habits and interests will enhance user engagement and discovery.
- Subscription & Membership Plans: Offering book subscription boxes or premium memberships can encourage regular reading and increase customer loyalty.

CHAPTER – 17

BIBILOGRAPHY

BIBILOGRAPHY

Websites:

- <https://www.tutorialspoint.com/>
- <https://www.tpointtech.com/>
- <https://www.w3schools.com/>
- <https://html.com/>
- <https://canvasjs.com/>
- <https://fontawesome.com/>
- <https://www.geeksforgeeks.org/>
- <https://www.youtube.com/>
- <https://github.com/>

Books:

- *Web Technologies: HTML, CSS, JavaScript* by Uttam K. Roy
- *Internet & World Wide Web: How to Program* by Paul Deitel and Harvey Deitel
- *Database System Concepts* by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan
- *Beginning PHP and MySQL: From Novice to Professional* by W. Jason Gilmore
- *Learning PHP, MySQL & JavaScript* by Robin Nixon