Project Report

Name	Student Code
Hritik Kulkarni	**AF0481783**

Index

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my teachers Prof._____ and Prof_____. for their exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by them time to time shall carry me a long way in the journey of life on which I am about to embark.

Special gratitude I give to my respected head of the division Mr.XYZ, for allowing me to use the facilities available and also help me to coordinate my project
Furthermore, I would also like to acknowledge with much appreciation the crucial role of faculty members on this occasion.

Lastly, I thank almighty, my parents and friends for their constant encouragement without which this project would not be possible.

Abstract

The ***Easy Shop*** **e-commerce system** is a web-based platform designed to simplify online shopping for both customers and administrators. It provides a user-friendly interface that allows users to browse products by categories, view detailed product information, manage shopping carts, and place orders efficiently. The system supports secure user registration, login, and order tracking, enhancing the overall shopping experience.

For administrators, Easy Shop offers powerful tools to manage products, inventory, orders, and customer data through a centralized dashboard. It includes features such as product search, discount management, and sales reports, enabling better decision-making and operational control. Integrated payment gateway support ensures secure and seamless transactions.

The platform is built to be responsive and accessible across various devices, ensuring a smooth user experience on desktops, tablets, and smartphones. By automating key business processes, Easy Shop helps businesses reduce operational costs, expand customer reach, and improve service delivery. The system is scalable, making it suitable for small startups as well as growing enterprises in the digital retail space.

Introduction

In today's age of Information Communication and Technology, we are surrounded by technology at every moment. From the moment we wake up to the time we go to bed, technology plays a crucial role in our lives. One of the most impactful developments in this digital era is the rise of e-commerce platforms. These systems allow people to shop online conveniently, saving both time and effort while offering a wide range of products accessible from anywhere.

Our project aims to develop an e-commerce website named Easy Shop that allows users to browse and purchase products online quickly, securely, and efficiently. This platform provides a better alternative to traditional shopping methods by delivering a seamless and user-friendly digital shopping experience. Users can search for products, view details, add items to their shopping cart, and complete purchases through a secure checkout process.

The system also features an administrative panel that allows the management of products, inventory, orders, and customer information. With a responsive design and easy navigation, Easy Shop ensures accessibility on various devices, making online shopping more flexible and engaging for users.

1.1 Objective of the Present Work

The objectives of this project are as follows:

- To develop a web application for an **e-commerce website (Easy Shop)** that allows users to purchase products online conveniently.
- To provide a platform where users can **browse products, view details, and make secure purchases**.
- To enable **real-time updates** of product listings, pricing, and availability.
- To deliver a shopping experience that is **easy-to-use, visually appealing, and engaging**.

- To ensure **fast, efficient, and secure transactions** through an integrated checkout and payment system.

- To make it possible for **anyone, anywhere, at any time** to access the shop via the internet at a low cost.

- To create a **dynamic and scalable platform** where products, categories, and user data can be managed easily by administrators without complexity.

3. System Analysis

3.1 Problem Definition

Many small and medium-sized businesses struggle to establish a professional and functional online presence. Traditional shopping methods are time-consuming and often lack convenience, while some existing e-commerce platforms are expensive or too complex for smaller retailers to manage. Customers also face issues like poor website navigation, lack of product information, and insecure transactions.

The **Easy Shop** e-commerce system addresses these challenges by offering a dynamic, user-friendly platform that allows businesses to manage their products and sales efficiently. It enables customers to browse products, place orders, and make payments online securely. The system also supports easy administrative control over inventory, orders, and customer data, making online retail simpler and more accessible.

3.2 Preliminary Investigation

Purpose

The purpose of the Easy Shop project is to create an efficient, scalable, and responsive e-commerce platform that makes online shopping easier for users and online selling more manageable for business owners. It replaces manual selling processes with a digital platform that supports real-time product updates, secure transactions, and order tracking.

Benefits

The system offers several advantages:

- **Instant Product Access:** Customers can browse and search for products by categories, brands, or names.
- **Efficient Product Management:** Admins can easily add, update, and remove products, manage stock levels, and track sales.
- **User Convenience:** Customers can register, log in, shop, and make secure payments anytime, anywhere.
- **Streamlined Order Management:** Real-time
-

3.3 Feasibility Study

The feasibility study evaluates whether the **Easy Shop** e-commerce project is practical, achievable, and beneficial within the constraints of available time, resources, and technologies.

Types of Feasibility Analysis

- **Technical Feasibility**
  - The system is developed using **Python (Django)** for the backend and **HTML, CSS, JavaScript, and Bootstrap** for the frontend.
  - **MySQL** is used as the database for secure and scalable storage of product and user data.
  - The system is compatible with most browsers and is hosted on a reliable web server for consistent performance.
  - Responsive design ensures the website is accessible on mobile and desktop platforms.
- **Economic Feasibility**
  - The use of **open-source technologies** such as Django and MySQL reduces software licensing costs.
  - The system improves sales and customer retention, offering a strong return on investment for businesses.
- **Operational Feasibility**
  - The system is intuitive and requires minimal training for both users and administrators.
  - It supports smooth shopping workflows—from product browsing to checkout—enhancing customer satisfaction.
  - Admin features like product and order management ensure operational efficiency.
- **Schedule Feasibility**
  - The development follows a modular approach covering planning, design, development, testing, and deployment phases.
  - The estimated timeline is achievable, with key deliverables aligned with realistic deadlines.
- **Social Feasibility**
  - The system encourages digital shopping habits and supports local businesses in expanding online.
  - It increases customer convenience and accessibility, promoting broader social acceptance.
  - The platform is inclusive and can be used by people with minimal technical knowledge.

3.4 Project Planning

Purpose of Project Planning

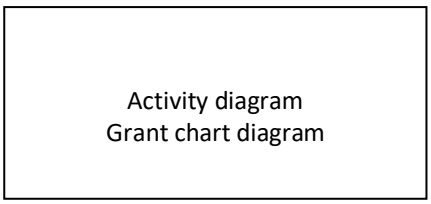
Project planning ensures that the development of the **Easy Shop** e-commerce platform follows a well-structured and strategic process. It helps define key activities, allocate resources, identify risks, and set timelines to ensure successful and timely delivery of the project.

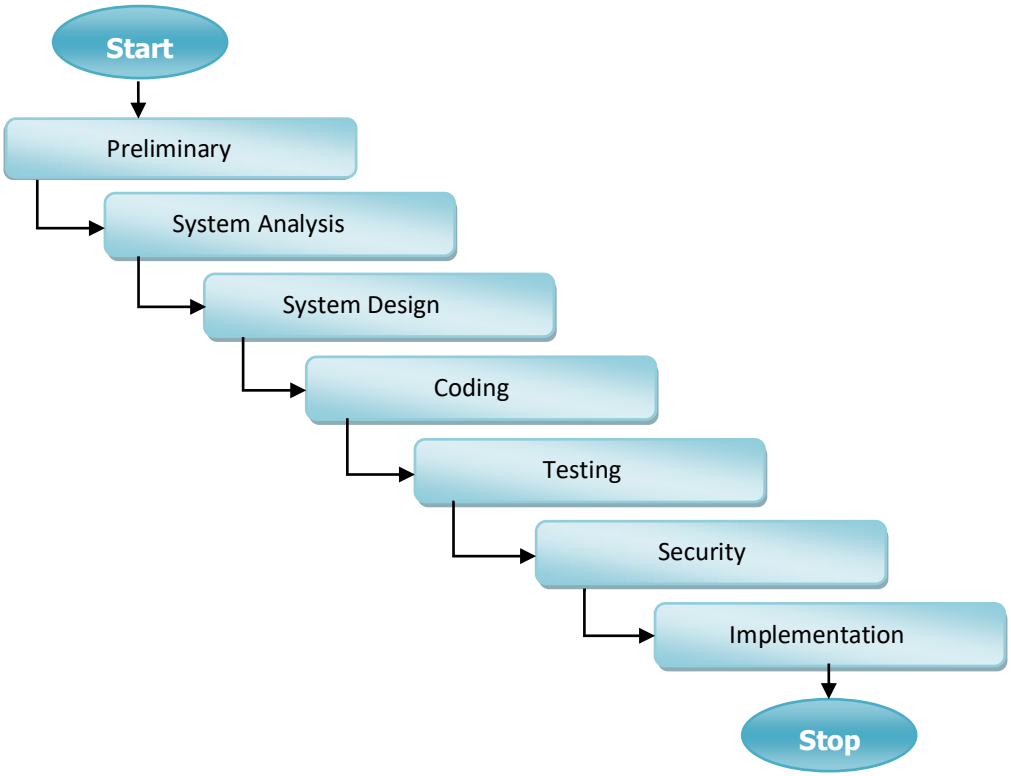
Phases Covered in the Plan

The project plan is divided into the following phases:

1. **Preliminary Investigation** – Understanding the scope, purpose, and objectives of the e-commerce system.
2. **System Analysis** – Identifying existing problems, collecting requirements, and proposing effective solutions.
3. **System Design** – Designing system architecture, database models, and user interface components.
4. **Coding** – Implementing backend logic using Django and developing frontend features for users and admins.
5. **Security** – Integrating authentication, data encryption, and secure payment methods.
6. **Testing** – Conducting unit testing, integration testing, and user acceptance testing to ensure system reliability.
7. **Implementation** – Deploying the final system on a live server and providing ongoing support for maintenance and upgrades.

3.5 Project Scheduling





3.6 Software Requirement Specification (SRS)

The Software Requirement Specification (SRS) defines the essential software and hardware requirements for the **Easy Shop e-commerce system**, ensuring the system performs efficiently, is user-friendly, and supports future scalability and maintenance.

System Overview

The **Easy Shop** platform provides users with a smooth online shopping experience and allows administrators to manage products, categories, orders, and customers through a secure and responsive web interface. The system is structured into three main modules:

1. **User Module** – Enables users to register, log in, browse products, add items to the cart, place orders, and make payments.

2. **Admin Module** – Provides tools for managing products, inventory, order processing, customer accounts, and viewing sales reports.
3. **Sub-Admin Module** – Grants limited administrative rights to manage specific categories or handle customer inquiries.

Software & Hardware Requirements

Software Requirements

- **Frontend:**
  - HTML, CSS, JavaScript, Bootstrap – for building a responsive and attractive user interface.
- **Backend:**
  - Python (Django) – for handling server-side logic, business processes, and database operations.
- **Database:**
  - MySQL – for storing user information, product data, orders, and payment records.
- **Payment Integration:**
  - Razorpay, PayPal, or Stripe – for secure and flexible online payments.
- **Web Server:**
  - Apache or Nginx – for deploying and hosting the web application.
- **Authentication & Security:**
  - Django's built-in authentication system with encryption for secure user access.

Hardware Requirements

- **Processor:** Intel Core i5 or higher
- **RAM:** Minimum 8GB
- **Storage:** At least 100GB to accommodate product images, customer data, and transaction logs
- **Connectivity:** Stable internet connection for real-time data access and updates
- **Display:** 1366x768 resolution or higher for optimal user experience

3.7 Functional Requirements

1. User Module

Users can:

- **Browse Products** – View items categorized under various sections such as Electronics, Fashion, Home Appliances, Groceries, etc.
- **Product Search** – Search for specific products using keywords or filters such as price, category, or brand.
- **User Registration/Login** – Create an account, log in securely, and manage profile and address information.
- **Shopping Cart** – Add, update, or remove items in the cart before proceeding to checkout.
- **Place Orders** – Complete orders using secure payment gateways and receive order confirmation.
- **Order Tracking** – View order history and track the delivery status of purchased items.
- **Product Reviews** – Leave feedback or rate products to help other users make informed choices.

---

2. Admin Module

Admins have complete control over the system and can:

- **Secure Login System** – Access the admin dashboard through authenticated login credentials.
- **Product Management** – Add, edit, delete, and restore products along with images, prices, and stock quantities.
- **Category & Inventory Management** – Organize products under categories and monitor inventory levels.
- **Order Management** – View, update, or cancel customer orders, and mark orders as shipped or delivered.
- **User Management** – View and manage customer accounts and activities.
- **Sub-Admin Management** – Create sub-admin accounts with role-based permissions.
- **Reports and Analytics** – Generate sales reports and monitor revenue, orders, and user activity.
- **Page Management** – Manage static pages like ‘About Us’, ‘Terms & Conditions’, and ‘Contact Us’.

---

3. Sub-Admin Module

Sub-Admins can:

- **Manage Products** – Add, update, or delete products under assigned categories.

- **Moderate Orders** – Process and manage orders within their assigned scope.
- **Limited Access** – Operate within permissions set by the admin without access to full system controls.
- **Customer Support Handling** – Respond to user queries and assist with product-related issues.

3.8 Software Engineering Paradigm

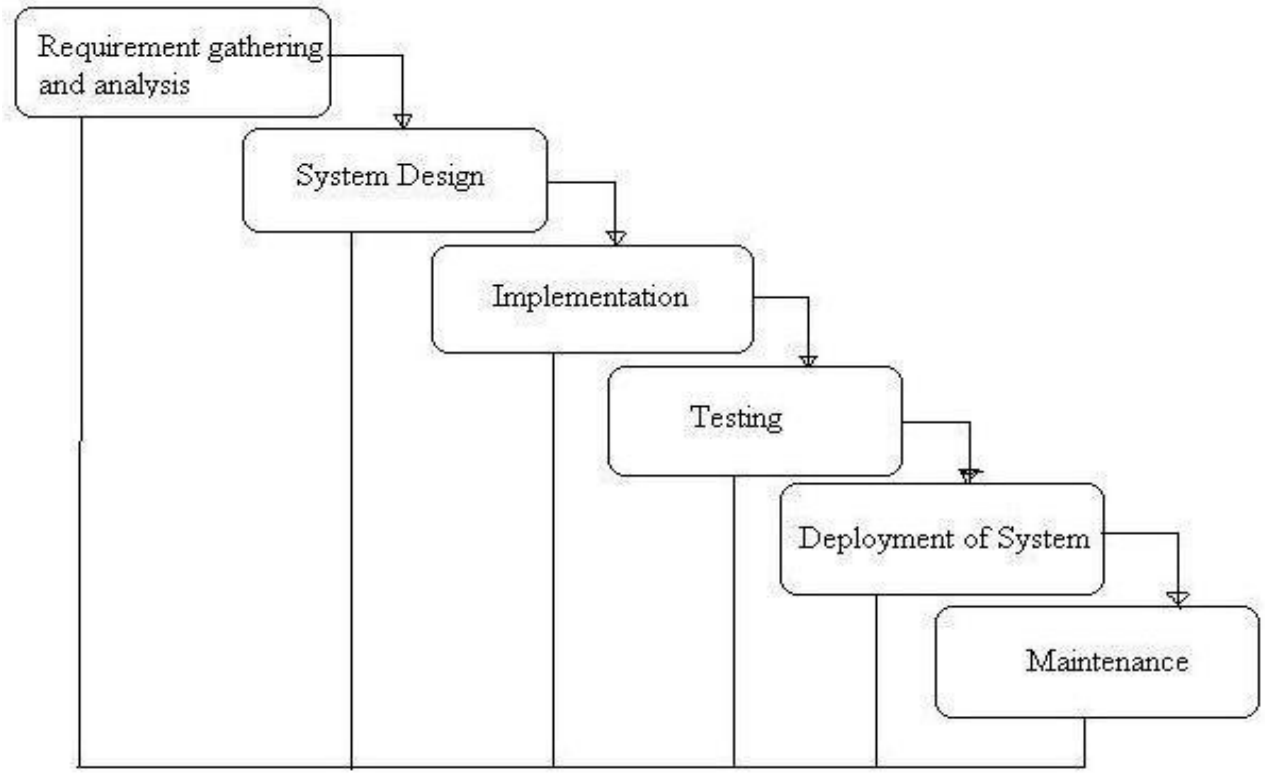
The development of the **Easy Shop** e-commerce system follows a structured approach to ensure efficiency, reliability, and maintainability. The chosen software engineering paradigm streamlines the project by defining clear phases while allowing iterative improvements for quality enhancement.

Development Model: Adapted Waterfall Model

The Waterfall Model is a traditional linear software development approach. For this project, an iterative feedback mechanism is incorporated to allow refinement of earlier phases based on insights gained during implementation and testing.

Key Adaptations in the Waterfall Model:

1. **Structured Phase Progression** – Each development phase follows a well-defined sequence, providing clarity and control over project progress.
2. **Iterative Refinements** – Feedback loops enable adjustments, especially between coding and testing phases, to improve functionality and fix defects early.
3. **Defined Milestones** – Clear deliverables and completion criteria are established for each phase before moving to the next.
4. **Flexible Adjustments** – Overlapping of certain activities is allowed when necessary to boost efficiency and accommodate changes.



Phases of Development

1. **Requirement Analysis & System Study**
    - Identify project goals, challenges, and functional requirements for the e-commerce system.
    - Gather stakeholder inputs and define core features such as product management, user accounts, and payment processing.
2. **System Design**
    - Design database schemas, system architecture, and module interactions.
    - Develop user interface layouts focused on usability and responsiveness for multiple devices.
3. **Implementation (Coding)**
    - Backend development using Python (Django) for business logic and data management.
    - Frontend development with HTML, CSS, JavaScript, and Bootstrap to create an engaging UI.
    - Database integration with MySQL for secure storage of product, order, and user data.
    - Integration of payment gateways and security measures.
4. **Testing & Debugging**
    - Conduct unit tests, integration tests, and user acceptance tests to validate system performance.

- Debug identified issues to improve reliability and user experience.

5. **Deployment & Maintenance**
   - Deploy the application on a scalable, secure web server.
   - Provide ongoing maintenance, including bug fixes, updates, and feature enhancements based on user feedback.

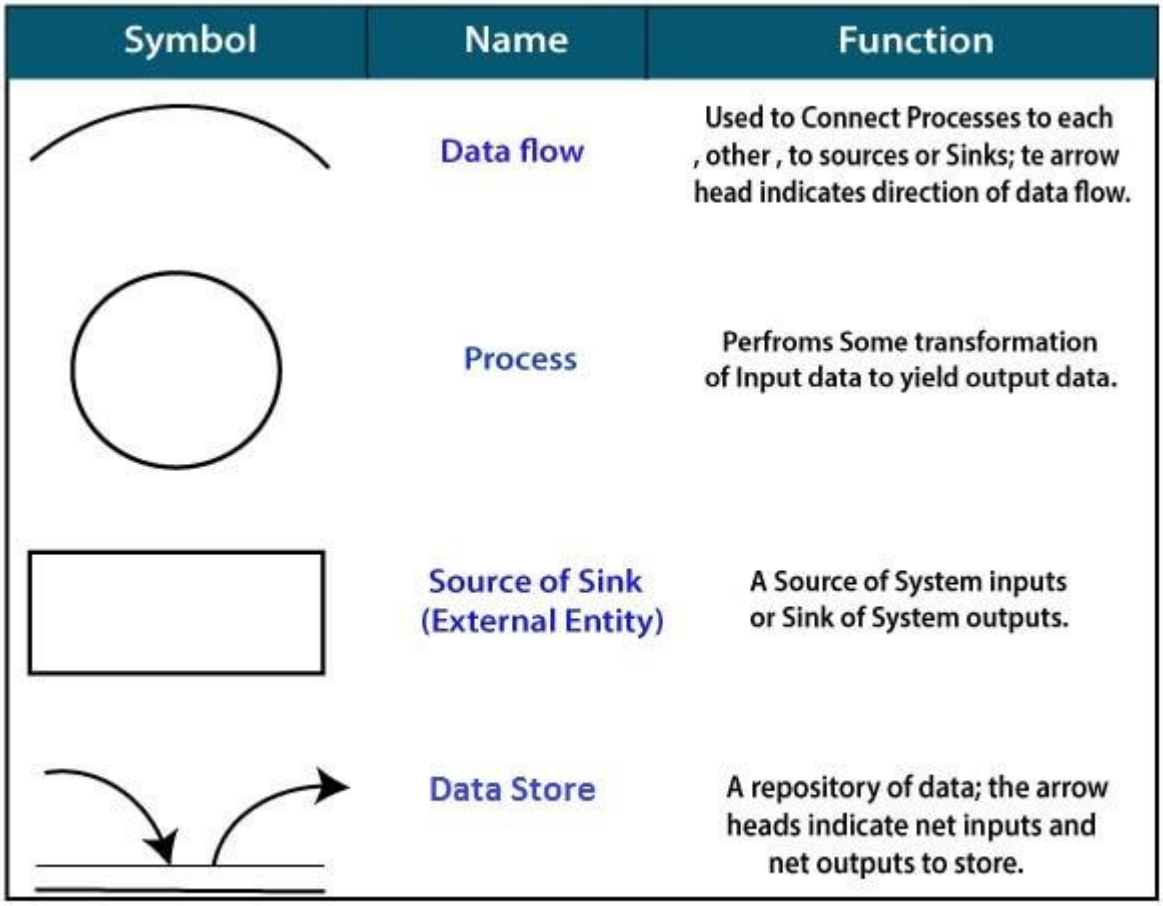
3.9 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a visual representation that illustrates how data moves through the Easy Shop e-commerce system. It provides a clear graphical view of how information enters, is processed, stored, and exits the system.
The main purpose of the DFD is to outline the scope and boundaries of the Easy Shop system and to serve as an effective communication tool between system analysts, developers, and stakeholders. It can also be used as a foundation for system redesign or enhancement.

Important Points about DFDs:

1. **Unique Naming:** Every element in the DFD (process, data store, external entity) should have a unique name for easy reference.
2. **Data Flow, Not Process Flow:** Unlike flowcharts, arrows in DFDs represent data flow, not the sequence of events or control flow.
3. **No Logical Decisions:** Avoid including decision points (diamonds) in a DFD since it does not represent control logic or conditional flows.

4. **Simplicity:** Avoid overloading the diagram with details. Defer error handling and exception flows until later stages of analysis.

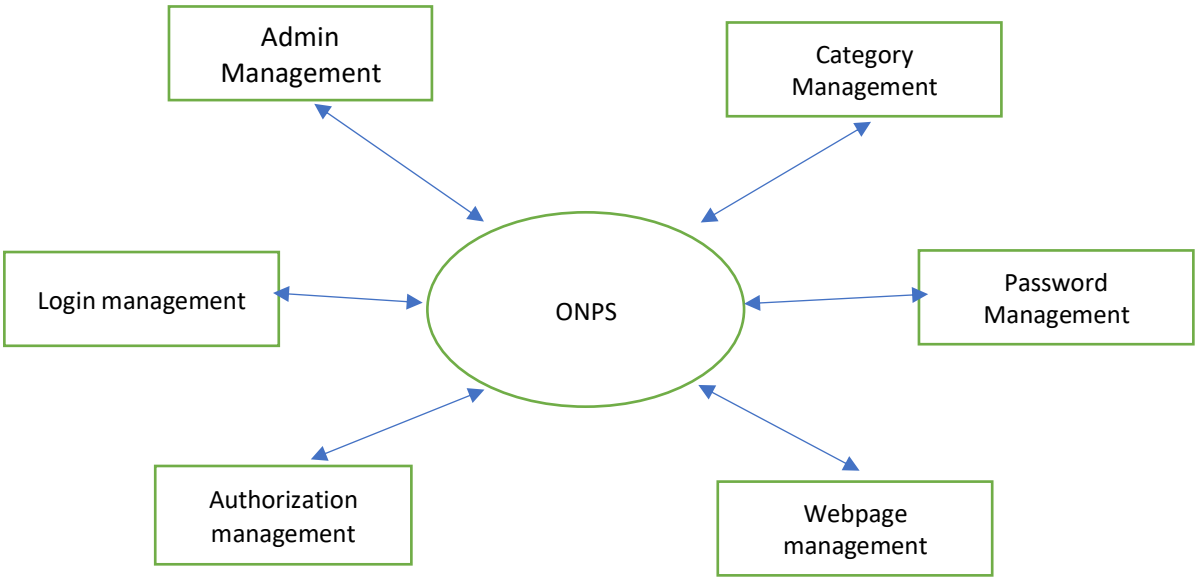


Symbols for Data Flow Diagrams

Standard Symbols Used in DFDs:

- **Process (Circle/Bubble):** Represents a process that transforms input data into output data.
- **Data Flow (Arrow):** Shows the movement of data between processes, data stores, and external entities.
- **Data Store (Parallel Lines):** Indicates storage locations where data is held for later use.
- **External Entity (Square/Rectangle):** Represents sources or destinations of data outside the system, such as customers, payment gateways, or suppliers.

Zero Level DFD of Easy Shop

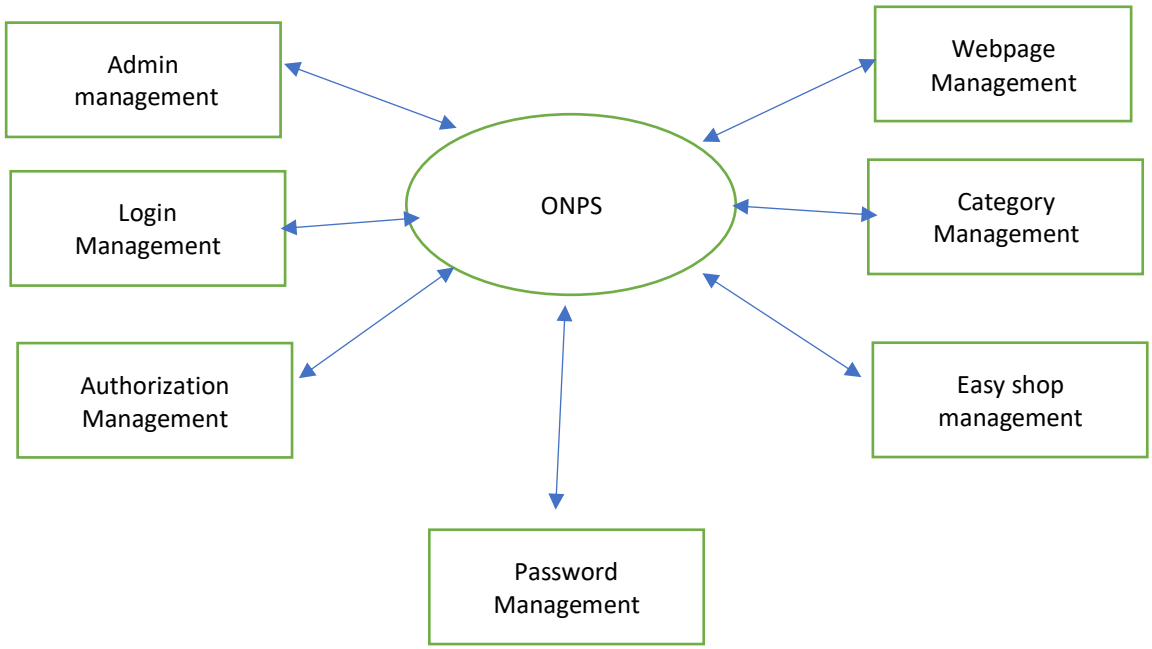


The Level 0 DFD (Context Diagram) of the Easy Shop system illustrates the key external entities interacting with the system and the major data flows. These include:

- **Users (Customers):** Browsing products, placing orders, making payments, and writing reviews.
- **Admin:** Managing products, orders, user accounts, and generating reports.
- **Payment Gateway:** Processing secure payment transactions.
- **Warehouse:** Managing inventory updates based on orders.
- **Shipping Service:** Handling delivery information and tracking.

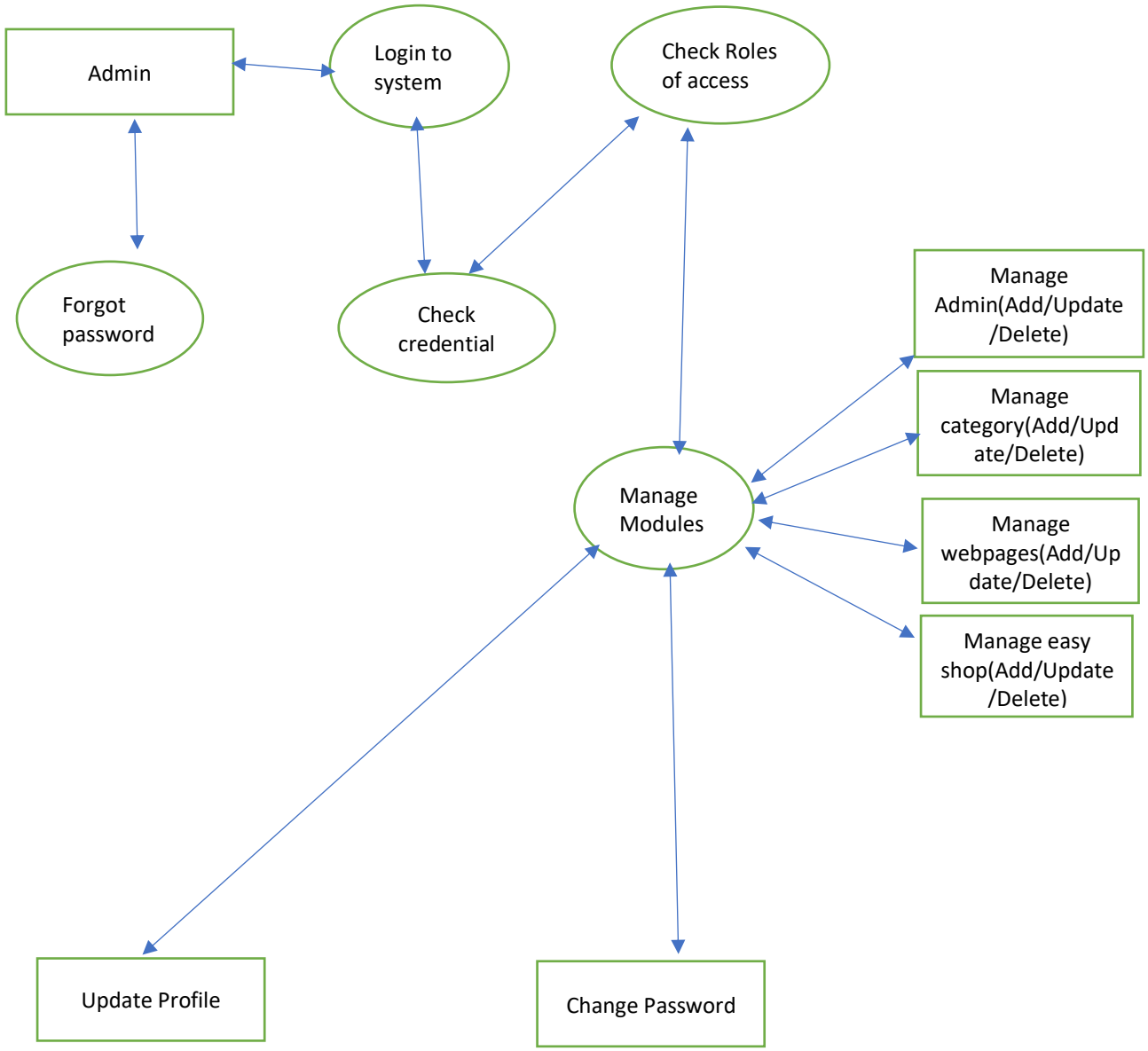
The diagram highlights how data such as product information, user details, orders, payment confirmation, and shipment status flows between these entities and the core Easy Shop system, showcasing the overall system workflow.

First-Level DFD

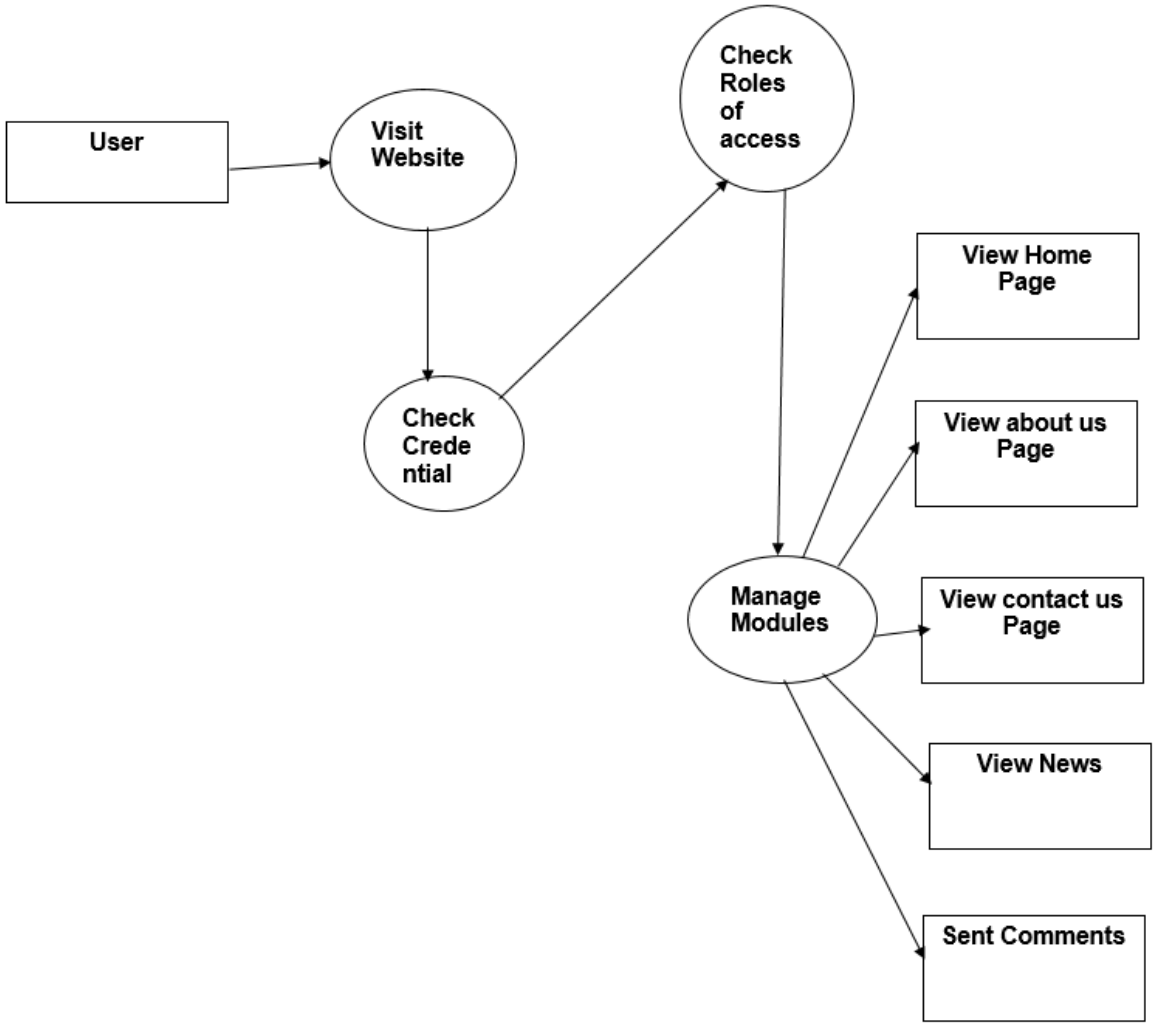


The **First-Level DFD** of ONPS shows how the system handles key functions like login, admin, sub-admin, authorization, and password management, and connects them to content-related modules such as news, category, subcategory, webpage, and user comments management for smooth portal operations

Second-Level DFD



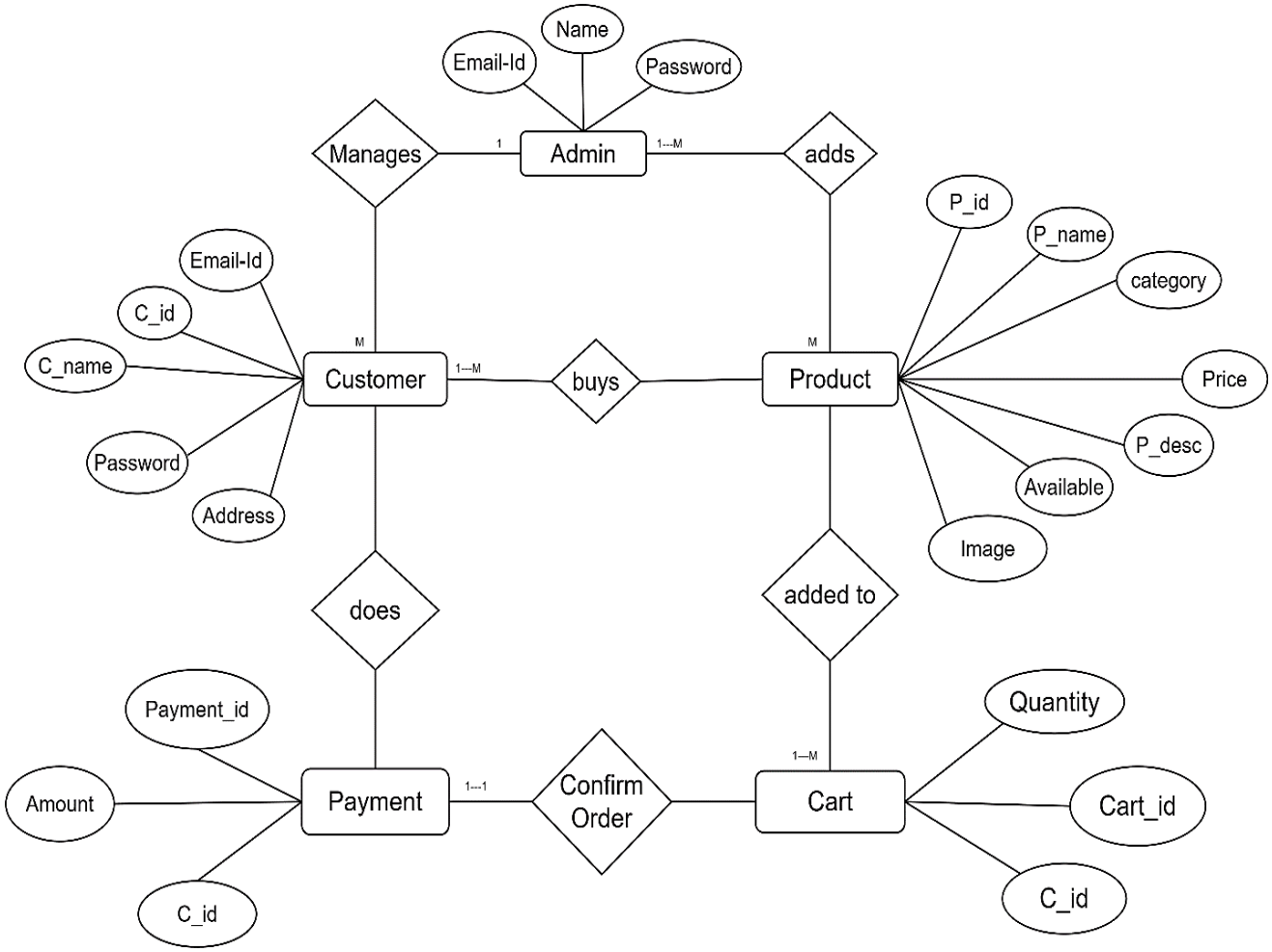
The **Second-Level DFD** shows the internal working of the admin in the ONPS system. It includes processes like login, credential checking, role-based access, and module management. The admin can manage subadmins, categories, subcategories, news, webpages, and user comments, along with profile updates, password changes, and performing sentiment analysis.



This **Second-Level DFD** illustrates the **User's interaction** with the ONPS system. When a user visits the website users can interact with various modules.

Entity Relationship Diagram(ERD)

The **ER Diagram** of the **Easy shop Ecommerce System** shows key relationships between **Users, Admins, News Articles, Categories, Comments, and Sentiment Analysis**, ensuring structured data management and insightful moderation



4.System Design

4.1 Modules

---

1. User Module

Users (customers) can:

- **Browse Products** categorized into Electronics, Fashion, Home Appliances, Groceries, and more.
- **Search for Products** using keywords, filters (price, brand, ratings), and categories.
- **Register and Log In** to create an account, manage profiles, and save preferences.
- **Add Products to Cart** and update quantities or remove items before checkout.
- **Place Orders** by providing shipping information and completing payments securely.
- **Track Orders** to view status updates from purchase to delivery.
- **Write Product Reviews** and rate items they have purchased.
- **Access Customer Support** for inquiries or complaints.

---

2. Admin Module

Admins have full control over the platform and can:

- **Secure Login System** – Access the admin dashboard with secure authentication and role-based access.
- **Dashboard Management** – Manage product categories, brands, inventory, and pricing.
- **Order Management** – View, update, and process customer orders including refunds and cancellations.
- **User Management** – Manage customer accounts, monitor activities, and address disputes.
- **Sub-Admin Management** – Create and manage sub-admin accounts with limited privileges.
- **Reports & Analytics** – Generate sales, inventory, and user activity reports for business insights.
- **Content Management** – Manage static pages like ‘About Us’, ‘Terms & Conditions’, and promotional banners.
- **Promotions & Discounts** – Create and manage special offers, coupon codes, and discounts.

---

3. Sub-Admin Module

Sub-admins can:

- **Manage Products** – Add, edit, or delete products within assigned categories.
- **Process Orders** – Handle order updates, shipping status, and customer queries related to their scope.
- **Moderate Reviews** – Approve or remove product reviews and ratings.
- **Limited Access** – Operate under permissions set by the admin without access to sensitive data or full system control.

4.2 DATA STRUCTURE OF ALL MODULES:

We have organized one database "**Easy shop"** for system design. It can be accessed directly or sequentially by registered. The database determines files, record, fields, and characters. It can be easily controlled and updated. **"Ecommerce system (Easy shop)" contains 15 MySQL tables(In this MySQL 6 table is customized and 9 table made by default in django) :**

Customized Tables Details

Admin and Sub-Admin Table(auth_user):

This store admin personal and login details.

```
      desc auth_user;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int          | NO   | PRI | NULL    | auto_increment |
| password     | varchar(128) | NO   |     | NULL    |                |
| last_login   | datetime(6)  | YES  |     | NULL    |                |
| is_superuser | tinyint(1)   | NO   |     | NULL    |                |
| username     | varchar(150) | NO   | UNI | NULL    |                |
| first_name   | varchar(150) | NO   |     | NULL    |                |
| last_name    | varchar(150) | NO   |     | NULL    |                |
| email        | varchar(254) | NO   |     | NULL    |                |
| is_staff     | tinyint(1)   | NO   |     | NULL    |                |
| is_active    | tinyint(1)   | NO   |     | NULL    |                |
| date_joined  | datetime(6)  | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
11 rows in set (0.01 sec)
```

Category Table: Table name(store_category)

This table store the details of category of shop

```
mysql> desc store_category;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| id    | bigint      | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50) | NO   |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
2 rows in set (0.01 sec)
```

Category Table: Table name(store_customer)

This table store the details of customer of shop

```
mysql> desc store_customer;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| id         | bigint       | NO   | PRI | NULL    | auto_increment |
| first_name | varchar(50)  | NO   |     | NULL    |                |
| last_name  | varchar(50)  | NO   |     | NULL    |                |
| phone      | varchar(15)  | NO   |     | NULL    |                |
| email      | varchar(100) | NO   |     | NULL    |                |
| password   | varchar(100) | NO   |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
```

Category Table: Table name(store_category)
This table store the details of order of shop

```
mysql> desc store_order;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | bigint       | NO   | PRI | NULL    | auto_increment |
| quantity    | int          | NO   |     | NULL    |                |
| address     | varchar(100) | NO   |     | NULL    |                |
| phone       | varchar(100) | NO   |     | NULL    |                |
| date        | date         | NO   |     | NULL    |                |
| status      | tinyint(1)   | NO   |     | NULL    |                |
| customer_id | bigint       | NO   | MUL | NULL    |                |
| product_id  | bigint       | NO   | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
8 rows in set (0.01 sec)
```

Category Table: Table name(store_product)
This table store the details of products of shop.

```
mysql> desc store_product;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| id          | bigint        | NO   | PRI | NULL    | auto_increment |
| name        | varchar(100)  | NO   |     | NULL    |                |
| price       | decimal(10,2) | NO   |     | NULL    |                |
| description | varchar(500)  | YES  |     | NULL    |                |
| image       | varchar(100)  | NO   |     | NULL    |                |
| category_id | bigint        | NO   | MUL | NULL    |                |
| is_sale     | tinyint(1)    | NO   |     | NULL    |                |
| sale_price  | decimal(10,2) | NO   |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
8 rows in set (0.01 sec)
```

Category Table: Table name(product_order)
This table store the details of product of shop

```
mysql> desc product_order;
ERROR 1146 (42S02): Table 'ecomm.product_order' doesn't exist
mysql> desc store_product;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| id          | bigint        | NO   | PRI | NULL    | auto_increment |
| name        | varchar(100)  | NO   |     | NULL    |                |
| price       | decimal(10,2) | NO   |     | NULL    |                |
| description | varchar(500)  | YES  |     | NULL    |                |
| image       | varchar(100)  | NO   |     | NULL    |                |
| category_id | bigint        | NO   | MUL | NULL    |                |
| is_sale     | tinyint(1)    | NO   |     | NULL    |                |
| sale_price  | decimal(10,2) | NO   |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
8 rows in set (0.01 sec)
```

Category Table: Table name(store_profile)
This table store the details of profile of shop.

```
mysql> desc store_profile;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | bigint       | NO   | PRI | NULL    | auto_increment |
| date_modified | datetime(6)  | NO   |     | NULL    |                |
| phone         | varchar(30)  | NO   |     | NULL    |                |
| address1      | varchar(200) | NO   |     | NULL    |                |
| address2      | varchar(200) | NO   |     | NULL    |                |
| city          | varchar(100) | NO   |     | NULL    |                |
| state         | varchar(100) | NO   |     | NULL    |                |
| zipcode       | varchar(100) | NO   |     | NULL    |                |
| country       | varchar(100) | NO   |     | NULL    |                |
| user_id       | int          | NO   | UNI | NULL    |                |
| old_cart      | varchar(100) | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
```

4.3 Procedural Design

4.3.1 User Panel Design

The User Panel in Easy Shop is designed to provide customers with seamless access to browse products, search, and manage orders. Upon landing on the homepage, users can select various options such as viewing product categories, searching for items, or logging into their account.

- Users navigate through product listings and select desired items to view details.
- They can add products to the shopping cart, update quantities, or remove items.
- When ready, users proceed to checkout, enter shipping and payment information, and place orders.
- Order confirmation and tracking information are then displayed.

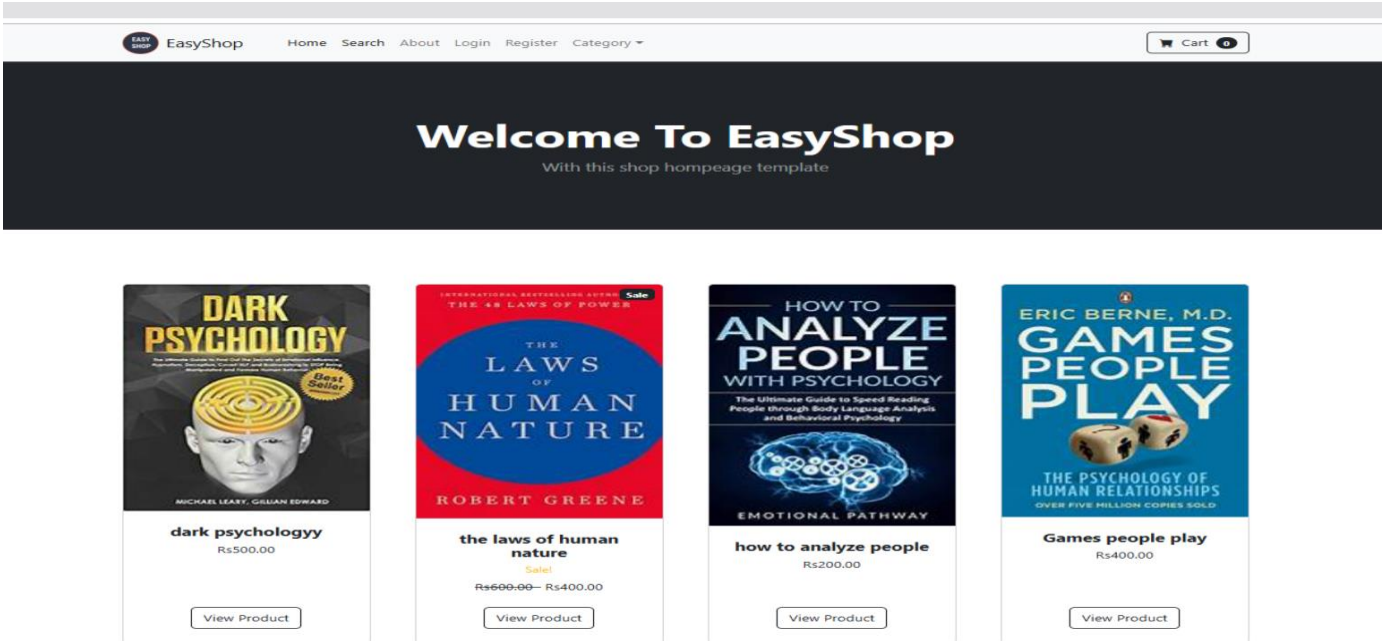
The flow of user actions from homepage to order completion is visually represented in the following flowchart:

4.3.2 Admin Panel Design

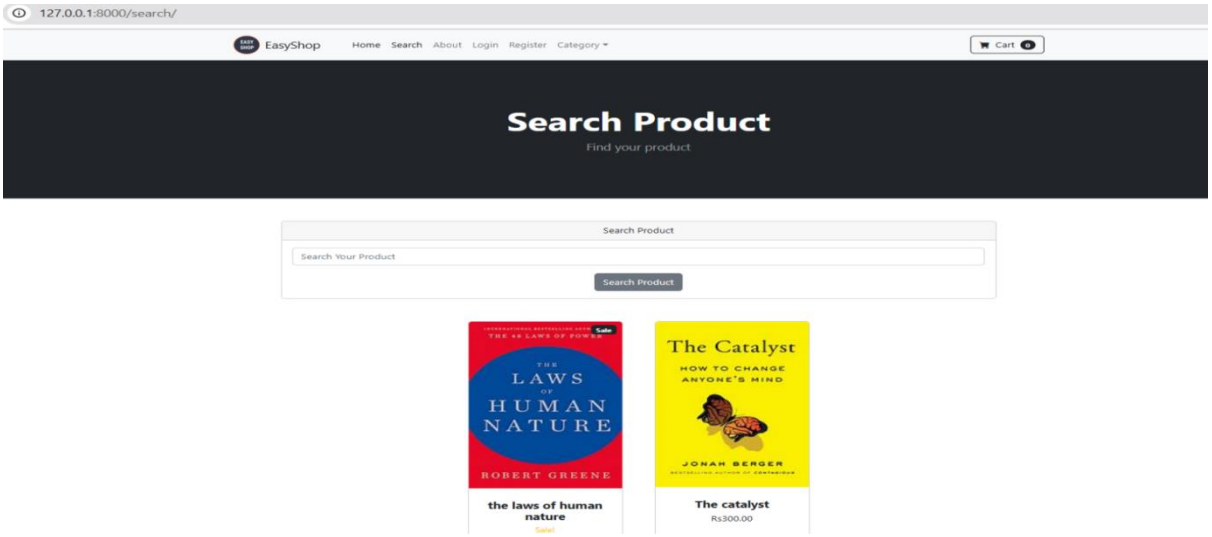
The Admin Panel includes a secure login system to protect sensitive data and administrative functionalities.

- The admin enters a username and password on the login page.
- Upon clicking the login button, the system verifies credentials against the database.
- If credentials are correct, a session starts, and the admin is redirected to the dashboard where they can manage products, orders, users, and reports.
- If credentials are invalid, an error message is displayed prompting retry.
- Access to the admin panel without successful login is prohibited.

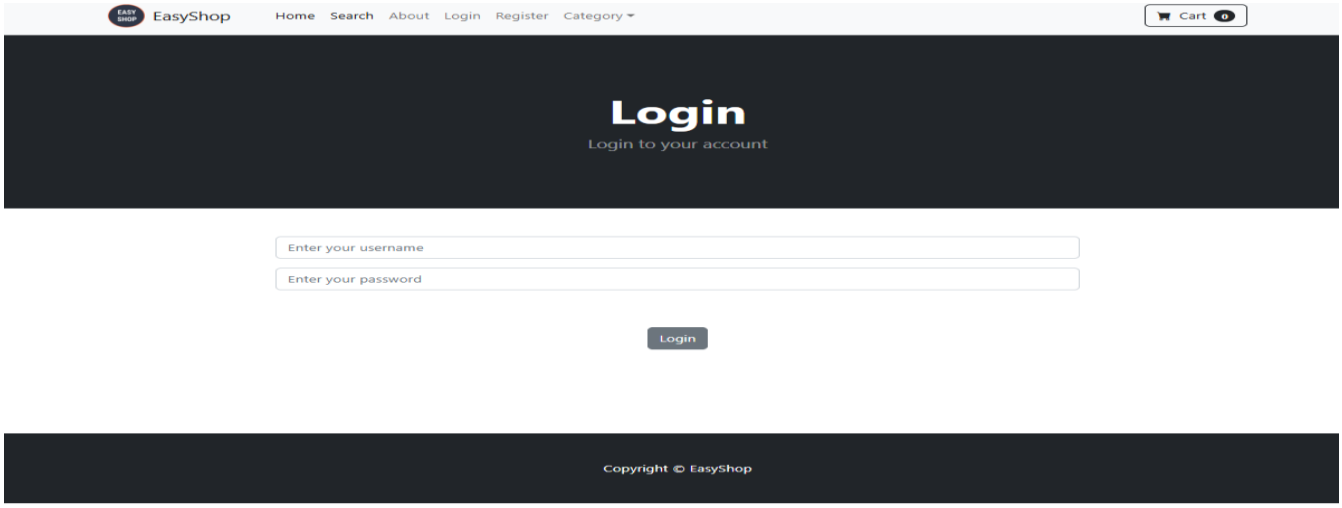
Screenshot



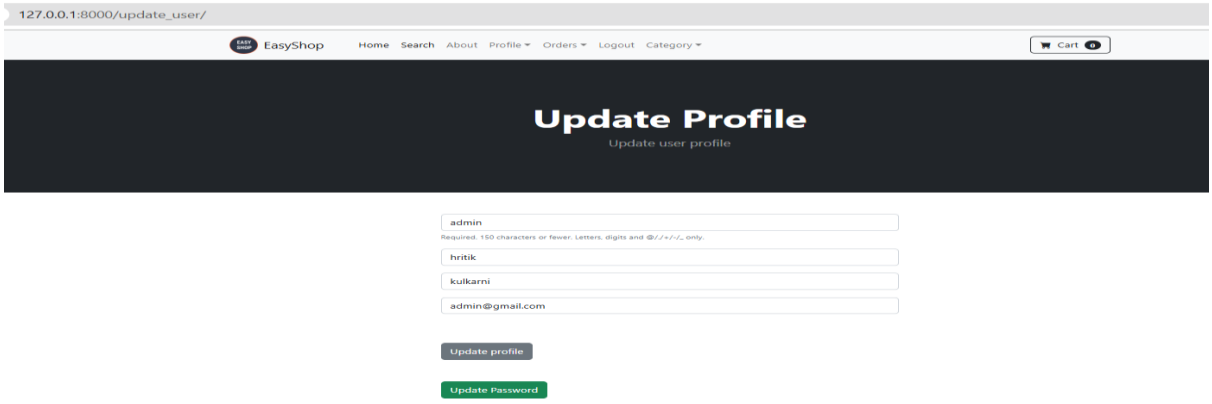
Search



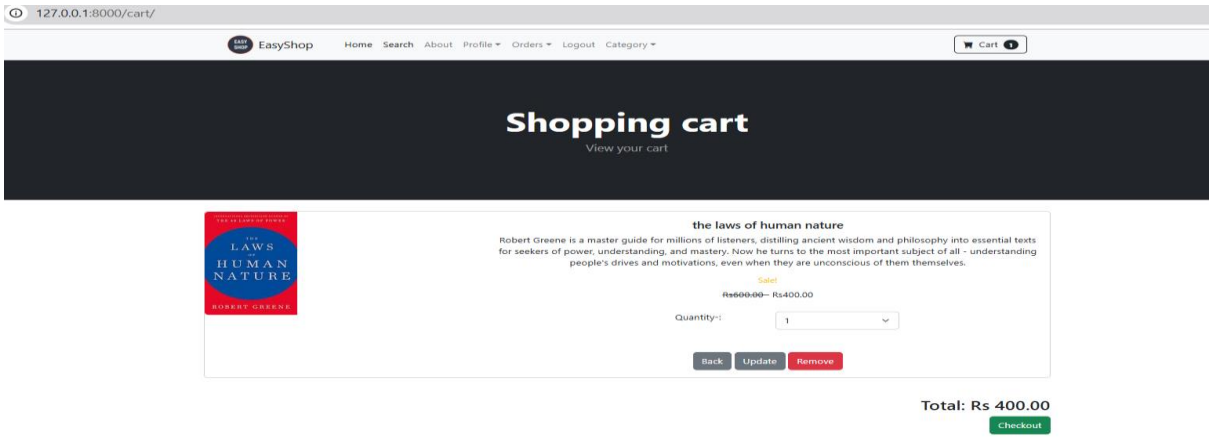
LOGIN



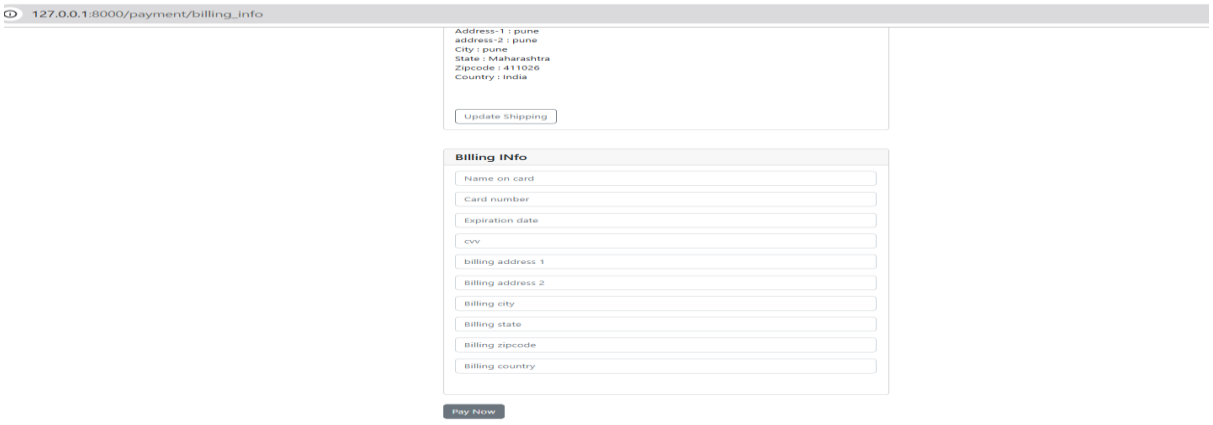
UPDATE PROFILE



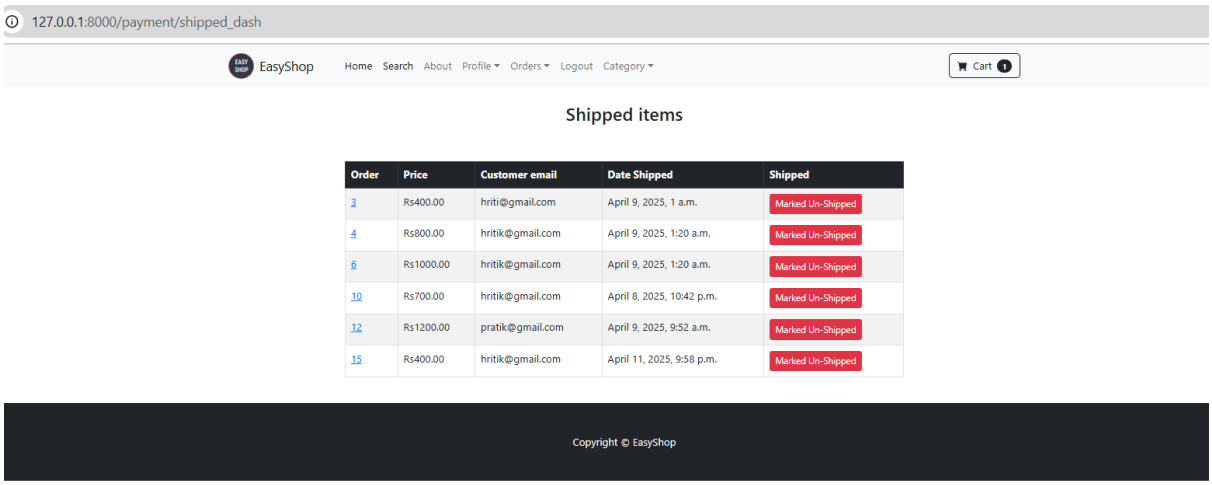
CART PAGE



Billing_page



Order_management



Conclusion

The EasyShop project is a full-stack e-commerce application built with Django and MySQL, designed to offer essential online shopping functionalities. It emphasizes clean code architecture, database integrity, and modular development using Django models, signals, and user authentication. This project highlights practical knowledge of backend development, user interaction flows, and database handling in a production-ready environment. Through this project, critical concepts such as order management, product categorization, user profiles, and shipping logistics are implemented effectively. Despite its limitations, the system serves as a powerful foundation for more complex e-commerce systems. With further enhancements—like secure payment integration, multi-vendor support, and improved UI—EasyShop could evolve into a scalable, real-world application.

Bibliography

Django Documentation – https://docs.djangoproject.com

MySQL Developer Guide – https://dev.mysql.com/doc/

Python Official Documentation – https://docs.python.org

W3Schools Django Tutorial – https://www.w3schools.com/django/
MDN Web Docs – https://developer.mozilla.org/

GeeksforGeeks – https://www.geeksforgeeks.org

Stack Overflow – https://stackoverflow.com

Real Python – https://realpython.com

References

Django ORM model relationships and signal handling practices

Python datetime and signals for event-driven actions

GitHub repositories of open-source Django projects for architecture inspiration

Blogs and tutorials on deploying and scaling Django e-commerce apps

Industry practices for building and testing web-based transactional systems