CHAPTER 1: INTRODUCTION

CHAPTER 1: INTRODUCTION

Comic books have been a significant part of popular culture for decades, offering a unique blend of storytelling, artistry, and entertainment. As the digital age progresses, traditional comic book stores are being supplemented—and in some cases, replaced—by online platforms that provide instant access to a vast collection of comics. ComicHub is an innovative website designed to cater to comic book enthusiasts by providing a centralized, digital platform for discovering, purchasing, and discussing comic books.

The primary purpose of ComicHub is to revolutionize the way users engage with comics, offering both digital and physical copies while integrating community engagement tools such as forums, ratings, and personalized recommendations. This website is tailored for a broad audience, from casual readers who enjoy graphic novels to serious collectors looking for rare issues and exclusive editions.

One of the major issues in the comic book industry is accessibility. Many comics are available only in select regions, limiting their reach. With ComicHub, users can browse and purchase comics from any location, ensuring that their favorite titles are always within reach.
Furthermore, the rise of independent comic creators has increased the demand for a platform that allows self-published artists to showcase their work without relying on major publishers. ComicHub addresses this need by offering a creator-friendly marketplace where independent artists can publish and sell their comics directly to consumers.

Additionally, ComicHub integrates modern technology to enhance the reading experience, with features like zoom-in panels, guided reading modes, dark mode, font customization, and offline reading options.
These features ensure that readers can enjoy their favorite comics in a format that suits their preferences.

Another key aspect of ComicHub is community engagement. Unlike traditional comic book stores, where readers interact with store clerks or fellow fans, ComicHub provides an online space where users can discuss comics, share reviews, and engage in fan theories. This sense of community enhances user experience and creates an immersive environment for comic book lovers.

Ultimately, ComicHub is a one-stop destination for comic book fans, collectors, and creators. It aims to bring the comic book industry into the digital era, offering an innovative, user-friendly, and interactive experience that caters to the diverse needs of its users.

1.1 **Background**

The comic book industry has experienced a significant transformation over the years, evolving from traditional print publications to digital platforms. The increasing adoption of digital technologies has changed how people read, purchase, and engage with comics, making online platforms an essential part of the industry. ComicHub is designed as an innovative digital platform that brings together comic book fans, collectors, creators, and publishers in one centralized hub.

The Evolution of Comic Books

Comic books have been a staple of entertainment for decades, with major publishers such as Marvel, DC Comics, Image Comics, and Dark Horse producing some of the most well-known superheroes and graphic novels. However, the comic book industry has faced several challenges, including:

- Declining physical sales due to digital alternatives.
- Limited availability of comics in certain regions.
- High costs of printing and distribution.
- Difficulties for independent creators in reaching a large audience.

With the rise of digital media, e-books, and online subscriptions, many traditional comic book stores have struggled to compete.
However, digital comic platforms have opened new doors, allowing
readers to instantly access thousands of titles without visiting a physical store.

The Need for an All-in-One Comic Platform

Despite the growing popularity of digital comics, many existing platforms are either limited in features or cater only to specific publishers. For example, Marvel Unlimited and DC Universe Infinite offer digital comics but only for their respective brands. Independent creators often struggle to distribute their work because mainstream platforms primarily focus on established publishers.

ComicHub aims to address these gaps by providing a platform that:

1. Offers a diverse collection of comics from both major publishers and independent creators.
2. Supports both digital and physical comic purchases, giving users the flexibility to choose how they consume their comics.
3. Creates a community-driven space where readers can interact, review, and discuss comics.
4. Empowers independent artists by providing a marketplace to sell their work directly to fans.

With advancements in cloud storage, AI-based recommendations, mobile compatibility, and interactive reading technologies, ComicHub integrates modern features to enhance the reading experience.

Features such as:

- Guided reading mode for an immersive panel-by-panel experience.
- Cross-device synchronization, allowing users to read comics on multiple devices.
- AI-based personalized recommendations to help users discover new comics.

By leveraging technology and offering a comprehensive solution, ComicHub seeks to become the ultimate platform for comic book enthusiasts worldwide.

1.2 **Objectives**

The primary goal of ComicHub is to bridge the gap between comic book publishers, independent creators, and readers, ensuring seamless access to a wide range of comics while fostering an engaging community. The following objectives outline the purpose and goals of ComicHub:

1.2.1 Provide a Comprehensive Comic Store

One of the core objectives of ComicHub is to create an extensive digital and physical comic store where users can:

- Browse thousands of comics by genre, publisher, and popularity.
- Purchase both digital and physical editions based on their preference.
- Subscribe to premium memberships for exclusive content.

1.2.2 Support Independent Creators

ComicHub aims to empower independent artists and writers by providing:

- Aself-publishing marketplace where creators can sell their work.
- Marketing and promotional tools to help them reach a wider audience.
- Revenue-sharing models to support independent artists.

1.2.3 Build an Interactive Community

Unlike traditional comic book stores, ComicHub integrates social engagement tools such as:

- User reviews and ratings to guide new readers.
- Community forums for discussions, debates, and fan theories.
- Live Q&A sessions with comic book creators.

1.2.4 Enhance Accessibility and User Experience

ComicHub is designed with user experience in mind, offering:

- A mobile-friendly and responsive interface for seamless reading.
- Dark mode, text resizing, and text-to-speech for accessibility.
- AI-driven recommendations to personalize content for users.

Byaddressing these objectives, ComicHub ensures that it provides a rich, engaging, and accessible platform for all comic book fans.

1.3 **Purpose and Scope**

1.3.1 Purpose

The purpose of ComicHub is to create an all-in-one digital platform that meets the needs of comic book fans, collectors, and creators.
Traditionally, comic book readers have relied on multiple platforms for different needs—one for reading comics, another for purchasing collector's editions, and another for discussing comics with the community. ComicHub combines all these features into one unified platform, making it easier for users to access and engage with their favorite comics.

The website aims to:

- Provide an easy-to-use digital marketplace for comics, ensuring users can instantly find and purchase the titles they love.
- Support creators by giving independent artists a platform to publish and sell their work, reducing their reliance on traditional publishers.
- Create a social space where fans can review, discuss, and connect over shared interests in comics.
- Increase accessibility by offering multiple purchase options, including one-time purchases, subscriptions, and bundled deals.

ComicHub seeks to revolutionize the way users experience comics, making it more inclusive, accessible, and community-driven than ever before.

1.3.2 Scope

The scope of ComicHub includes various functionalities and services that cater to a global audience of comic book enthusiasts. Below are the primary areas covered by the platform:

A. User Access & Membership

ComicHub is designed for different types of users, including:

- Casual readers who want to browse and read comics without long-term commitments.
- Collectors who seek rare editions, signed comics, and exclusive merchandise.
- Independent creators who want to publish and sell their comics. To cater to different

audiences, ComicHub provides:

- Free access to a limited selection of comics.
- Premium subscriptions for unlimited access to digital comics.
- Special memberships for collectors, providing access to exclusive deals and limited-edition prints.

B. Marketplace & Digital Library

The platform includes a robust marketplace where users can:

- Purchase comics from multiple publishers (Marvel, DC, Image, Dark Horse, etc.).
- Order collector's editions, rare prints, and signed copies.
- Buy, sell, and trade comics with other users.

C. Community & Engagement Features

ComicHub is more than just a store—it's a social hub for comic book fans. Users can:

- Review and rate comics to help guide new readers.
- Engage in discussions and forums on favorite characters, stories, and creators.
- Participate in fan events, live Q&Asessions, and exclusive interviews.

D. Technological Integration

ComicHub leverages modern technologies to improve user experience:

- AI-powered recommendations for personalized reading suggestions.
- Cloud storage for syncing digital libraries across devices.
- Mobile and tablet compatibility for reading on the go.

CHAPTER 2: SYSTEM ANALYSIS

2.1 Existing System

Introduction

The comic book industry has undergone significant changes over the years, transitioning from traditional physical comic book stores to online platforms.
However, the existing systems for distributing and consuming comics still present several challenges. The traditional comic book distribution model involves physical comic book stores, publisher-specific platforms, and third-party online marketplaces. While these systems have served the industry for years, they come with limitations that make them inefficient in the digital age.

Understanding the shortcomings of the existing system is crucial for developing an improved solution, such as ComicHub, which aims to provide an all-in-one platform for comic enthusiasts, creators, and publishers.

Traditional Comic Book Stores Historically, comic book stores have

been the primary source for purchasing and

collecting comics. These brick-and-mortar stores house physical copies of comics,

graphic novels, and collectibles, catering to local readers and collectors. While traditional comic stores
have played a significant role in comic book culture, they face several challenges:

1. Limited Availabilityand Selection
   - Due to space constraints, comic book stores can only stock a limited number of titles at any given time.
   - Readers looking for rare editions or older issues often struggle to find them in local stores.
   - Independent comics or small-publisher works are frequently overlooked in favor of major titles from Marvel, DC, and other big publishers.
2. Geographical Barriers
   - Many comic book enthusiasts do not have access to a dedicated comic book store in their area, especially in rural or remote locations.
   - International readers may find it difficult to obtain certain comics due to regional restrictions or high shipping costs.
   - Physical stores rely heavily on foot traffic, which limits their audience to a specific geographic area.
3. High Operational Costs
   - Running a physical store requires substantial investment in rent, utilities, and inventory management.

- These costs often lead to higher prices for consumers, making comics less accessible to casual readers.
- The decline of physical retail due to digitalization has further hurt independent comic book stores.

4. Lack ofTechnological Integration
   - Most traditional stores do not leverage modern technology, such as cloud-based digital libraries, AI-based recommendations, or interactive reader engagement tools.
   - Manual inventory tracking and checkout processes can be slow and inefficient.

Standalone Publisher Platforms

Several major comic book publishers, such as Marvel, DC Comics, and Image Comics, have created their own digital platforms to distribute comics. These platforms allow users to buy and read comics exclusively from their respective publishers. While these platforms have

helped bring comics into the digital space, theypresent several limitations:

1. Publisher Exclusivity
   - Platforms such as Marvel Unlimited and DC Universe Infinite provide access only to their respective brand's comics, preventing users from exploring other publishers' work.
   - Readers who enjoy comics from multiple publishers need to subscribe to multiple platforms, leading to inconvenience and higher costs.
2. Lack of Support for Independent Creators
   - These platforms focus solely on established publishers and do not provide an avenue for independent creators to publish their work.
   - Indie comic book artists must seek alternative distribution methods, such as crowdfunding or self- publishing on third-party platforms.
3. Minimal Community Engagement
   - Publisher platforms focus primarily on comic book sales and reading experiences but lack social features.
   - There is no space for readers to review, discuss, or interact with one another or with comic book creators.

Digital marketplaces such as Amazon Kindle, ComiXology, and eBay have expanded the availability of digital comics. These platforms allow users to buy, sell, and trade comic books, both digitally and physically. However, theyalso present significant drawbacks:

1. Lack of Exclusive Content
   - Many online marketplaces do not offer exclusive comics, making it difficult for indie creators to gain visibility.
   - Users often have to visit multiple platforms to find certain comics.
2. High Commission Fees
   - Marketplaces like Amazon and ComiXology take a significant percentage of each sale, reducing profit margins for creators.
   - Independent comic book artists often receive lower earnings compared to selling directly to consumers.
3. Minimal Social Interaction
   - These platforms lack dedicated comic book discussion forums, fan engagement features, or interactive storytelling options.
   - Readers cannot easily connect with other fans, discuss their favorite comics, or engage in community-driven activities.

Common Limitations Across the Existing Systems

1. Despite the various ways comics are distributed today, several fundamental issues persist: Fragmentation
   1. Readers must navigate multiple platforms to access content from different publishers and independent creators.
   2. No single platform provides an all- encompassing comic book experience.
2. Exclusivity Issues
   1. Publisher-specific platforms limit reader choices, restricting access to a full spectrum of comic book content.
3. Lack of Support for Indie Creators
   1. Independent artists struggle with distribution, visibility, and profitability.
   2. No mainstream platform is designed to empower indie creators without high commission fees.
4. Limited Community Features
   1. Existing systems do not foster strong reader engagement, fan interactions, or

creator-reader discussions.

5. High Costs for Readers

    1. Purchasing digital comics across multiple platforms becomes expensive, discouraging casual readers.

CHAPTER 3 SYSTEM DESIGN

ComicHub is a comprehensive platform designed to manage comic book collections, provide a seamless reading experience, facilitate purchases, and engage the community. To achieve this, the system is divided into several key modules. Each module has specific functionalities that contribute to the overall efficiency and user experience of the platform.

1. User Management Module

The User Management Module is responsible for handling user accounts, authentication, and roles within the system. It includes:

- User Registration & Authentication: Users can sign up using an email, username, and password. Authentication mechanisms such as OAuth and two-factor authentication (2FA) enhance security.
- Role-based Access Control (RBAC): Different user roles include Admin, User, and Retailer, each with specific permissions.
- Profile Management: Users can update their profiles, including profile pictures, personal details, and reading preferences.
- Password Recovery: Users can recover their passwords via email verification or security questions.

2. Comic Collection & Catalog Module

This module serves as the core of ComicHub by managing the comic book database, allowing users to browse, search, and filter collections.

- Comic Book Listings: Comics are stored with details such as title, author, publisher, and genre.
- Search & Filtering: Users can search bytitle, author, publisher, or genre.
- Comic Details & Metadata: Each comic has a detailed page with descriptions, issue numbers, and release dates.
- Wishlist & Favorites: Users can save comics they intend to read or purchase.

3. Digital Comics Reader Module

This module enhances the reading experience by offering a user-friendly digital reader.

- Online Comic Viewer: Ahigh-quality viewer allows users to read comics within the app.
- Zoom& Navigation: Readers can zoom in on pages and navigate seamlessly.
- Bookmarking & Reading History: Users can resume from where they left off.
- Offline Reading Mode: Subscribers can download comics for offline reading.

4. Retail & Subscription Module

This module enables purchases, subscriptions, and payment processing.

- Purchase & Subscription Plans: Users can buy individual comics or subscribe for unlimited access.

- Payment Gateway Integration: Supports credit cards, PayPal, and cryptocurrency.
- Order & Invoice Management: Generates invoices and order tracking.
- Discounts & Promotions: Retailers can create discount codes and promotions.

5. Community & Engagement Module

ComicHub fosters a community through engagement features.

- User Reviews & Ratings: Users can rate and review comics.
- Discussion Forums: Aplatform for discussing comics and storylines.
- Social Sharing: Users can share their favorite comics on social media.
- Achievements & Badges: Gamification elements reward user engagement.

6. Inventory & Retailer Management Module Designed for store owners and comic book retailers.

- Stock & Inventory Tracking: Retailers can manage their inventory.
- Sales Reports & Analytics: Provides insights into sales trends.
- Supplier & Distribution Management: Tracks comic book orders from suppliers.
- Multi-Store Support: Retailers with multiple locations can manage them centrally.

7. Admin Panel & Reports Module

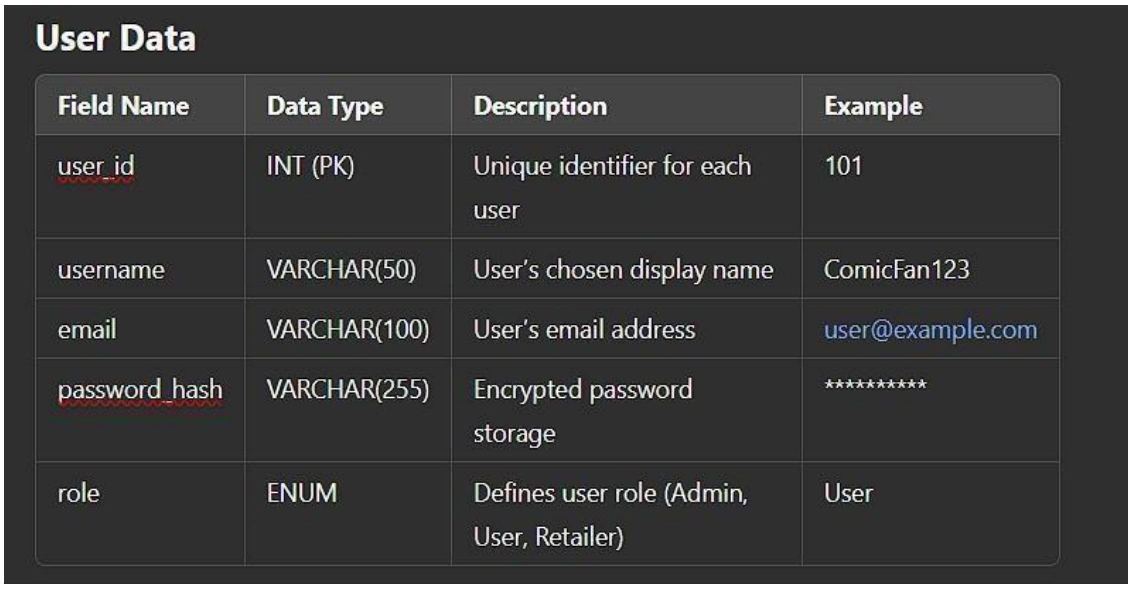
The Admin Panel ensures smooth platform management.

- Dashboard & Insights: Displays real-time user activity and system performance.
- Content Moderation: Admins can review and remove inappropriate content.
- System Logs & Reports: Tracks all platform activities for auditing.
- User & Retailer Support: Admins can respond to support tickets.

3.1 Modules Division

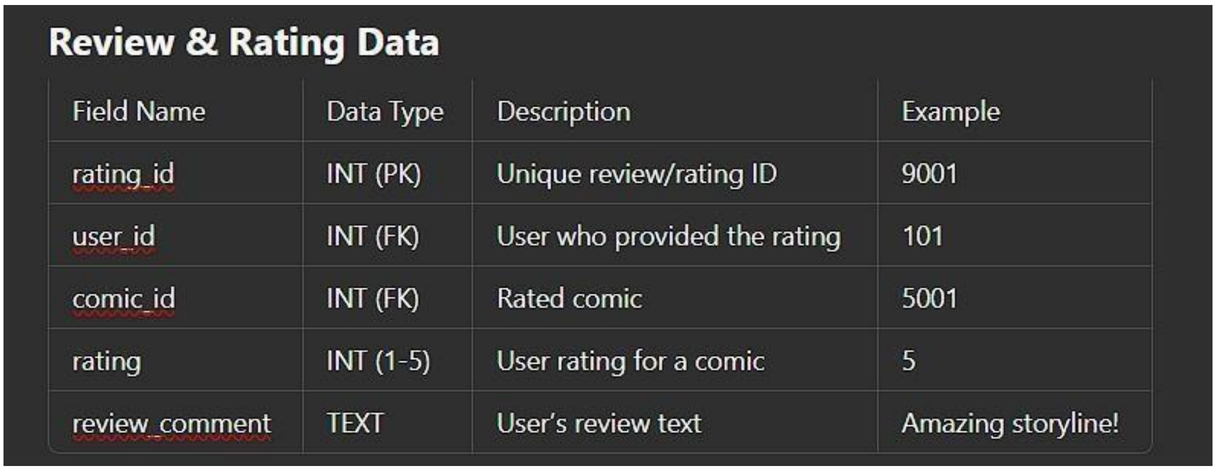
3.2 Data Dictionary

A data dictionary defines all essential fields in the ComicHub database, ensuring consistency in data handling.



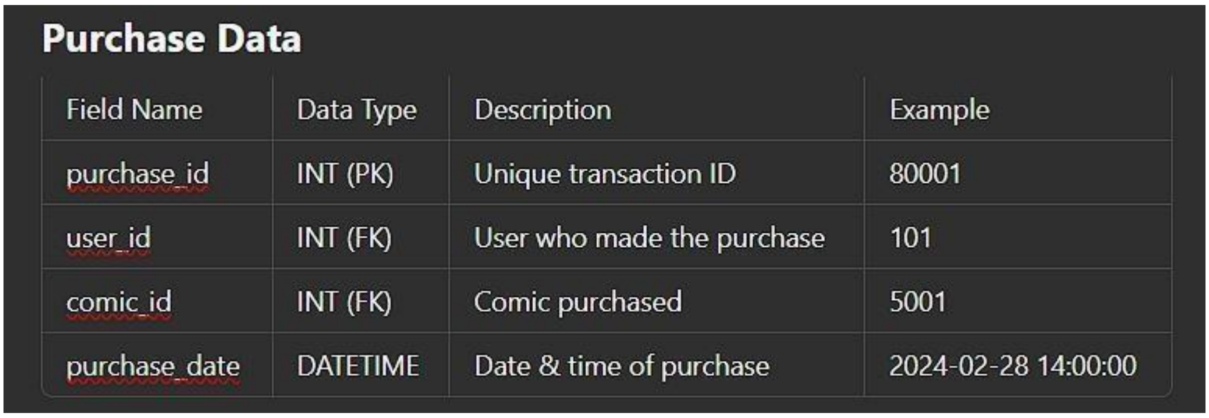
User Data

Field Name	Data Type	Description	Example
user_id	INT (PK)	Unique identifier for each user	101
username	VARCHAR(50)	User's chosen display name	ComicFan123
email	VARCHAR(100)	User's email address	user@example.com
password_hash	VARCHAR(255)	Encrypted password storage	**********
role	ENUM	Defines user role (Admin, User, Retailer)	User



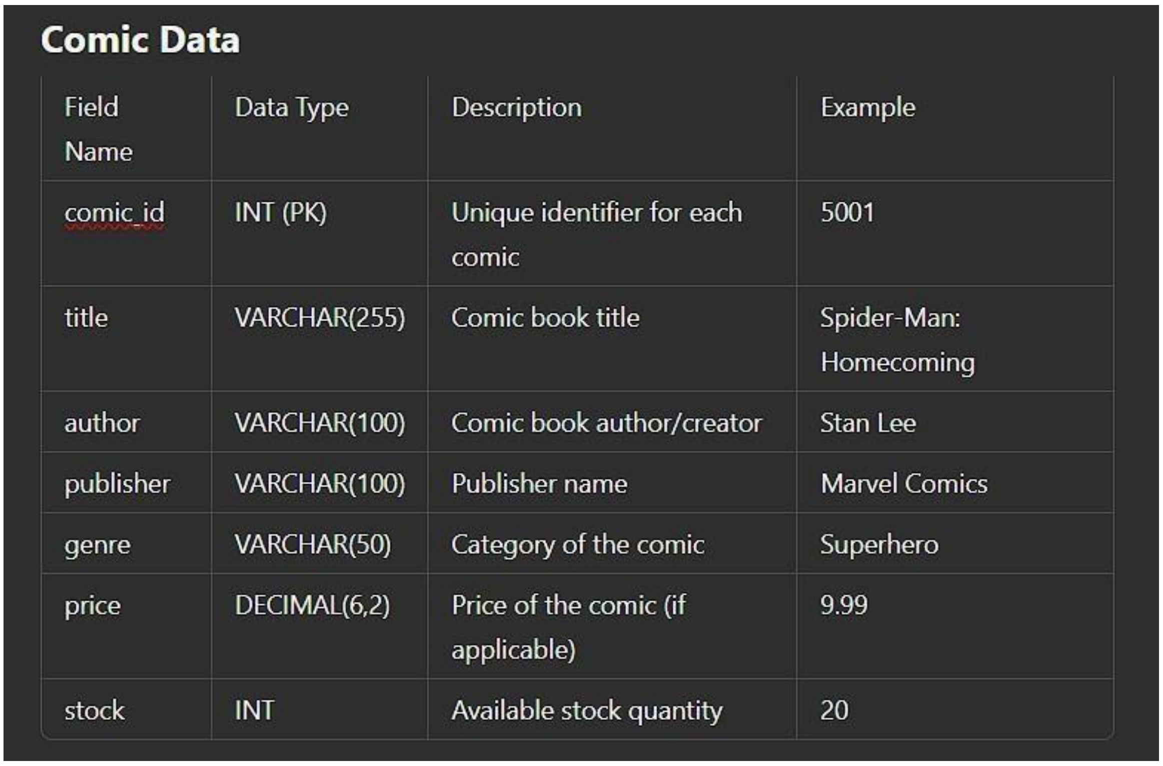
Review & Rating Data

Field Name	Data Type	Description	Example
rating_id	INT (PK)	Unique review/rating ID	9001
user_id	INT (FK)	User who provided the rating	101
comic_id	INT (FK)	Rated comic	5001
rating	INT (1-5)	User rating for a comic	5
review_comment	TEXT	User's review text	Amazing storyline!



Purchase Data

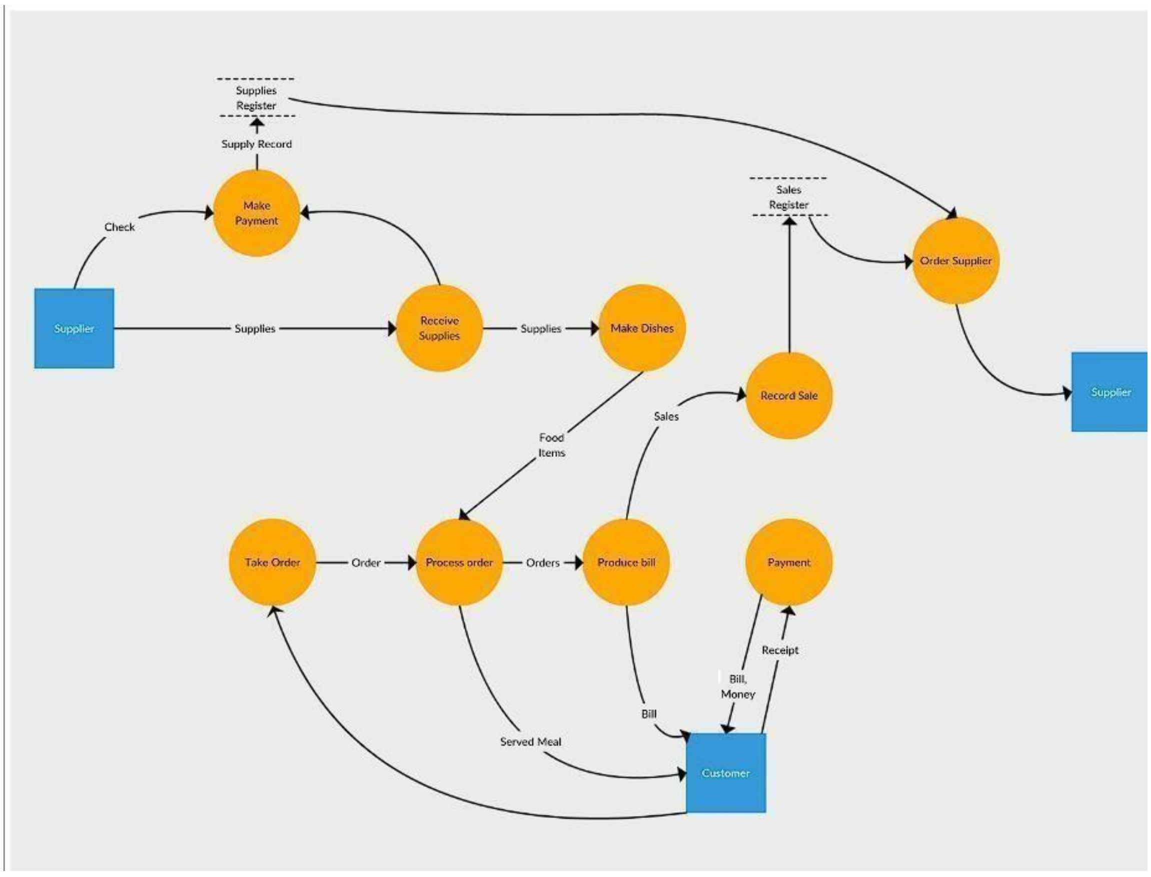
Field Name	Data Type	Description	Example
purchase_id	INT (PK)	Unique transaction ID	80001
user_id	INT (FK)	User who made the purchase	101
comic_id	INT (FK)	Comic purchased	5001
purchase_date	DATETIME	Date & time of purchase	2024-02-28 14:00:00



Comic Data

Field Name	Data Type	Description	Example
comic_id	INT (PK)	Unique identifier for each comic	5001
title	VARCHAR(255)	Comic book title	Spider-Man: Homecoming
author	VARCHAR(100)	Comic book author/creator	Stan Lee
publisher	VARCHAR(100)	Publisher name	Marvel Comics
genre	VARCHAR(50)	Category of the comic	Superhero
price	DECIMAL(6,2)	Price of the comic (if applicable)	9.99
stock	INT	Available stock quantity	20

DFD diagram:


Supplies Register
Supply Record
Make Payment
Check
Supplier
Supplies
Receive Supplies
Supplies
Make Dishes
Sales Register
Order Supplier
Supplier
Record Sale
Sales
Food Items
Take Order
Order
Process order
Orders
Produce bill
Payment
Receipt
Bill, Money
Bill
Served Meal
Customer

1.4) FrontEnd & BackEnd

Front End	**Bootstrap:** • Bootstrap is a free front-end framework for faster and easier web development • Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins. • Bootstrap also gives you the ability to easily create responsive designs.
Back End	**Python Django:** • Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. • Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. • It's free and open source. • Object Relational Database.

2. FEASIBILITY STUDY

2.1) Operational Feasibility

- Analyze the existing comic websites and identify how they can be adapted or transformed for online operations.
- Consider the impact of the online platform on day-to-day activities, such as customer service, sales diagnostics and order fulfilment.
- Evaluate the feasibility of implementing efficient workflows to handle online inquiries, bulk requests, and tracking of customer orders

2.2) Technical Feasibility

- Evaluate the compatibility of existing systems and infrastructure with the proposed online comic websites
- Consider the technical requirements for the online platform, such as hosting, security measures, data backup, and system maintenance.
- Assess the feasibility of integrating the online system with existing systems, such as inventory management, appointment scheduling, and customer relationship management (CRM) tools.

2.3) Economical Feasibility

- Evaluate the costs associated with implementing and maintaining the online garage management system, including software development, staff training, infrastructure upgrades, and ongoing technical support.
- Analyze the projected benefits and potential return on investment (ROI) in terms of increased customer reach, improved operational efficiency, reduced administrative tasks, and revenue generation.

3. REQUIREMENT ANALYSIS

3.1) Stakeholder

In project management, a stakeholder refers to any individual, group, or organization that has an interest or influence in the project, directly or indirectly. They can significantly impact or be impacted by the project's objectives, activities, or outcomes. Stakeholders can be both internal and external to the project.

- **Comic Owners/Managers**: As the primary stakeholders, they have a vested interest in the successful implementation of the online garage management system. Theywill benefit from improved operational efficiency, streamlined processes, and increased customer satisfaction.

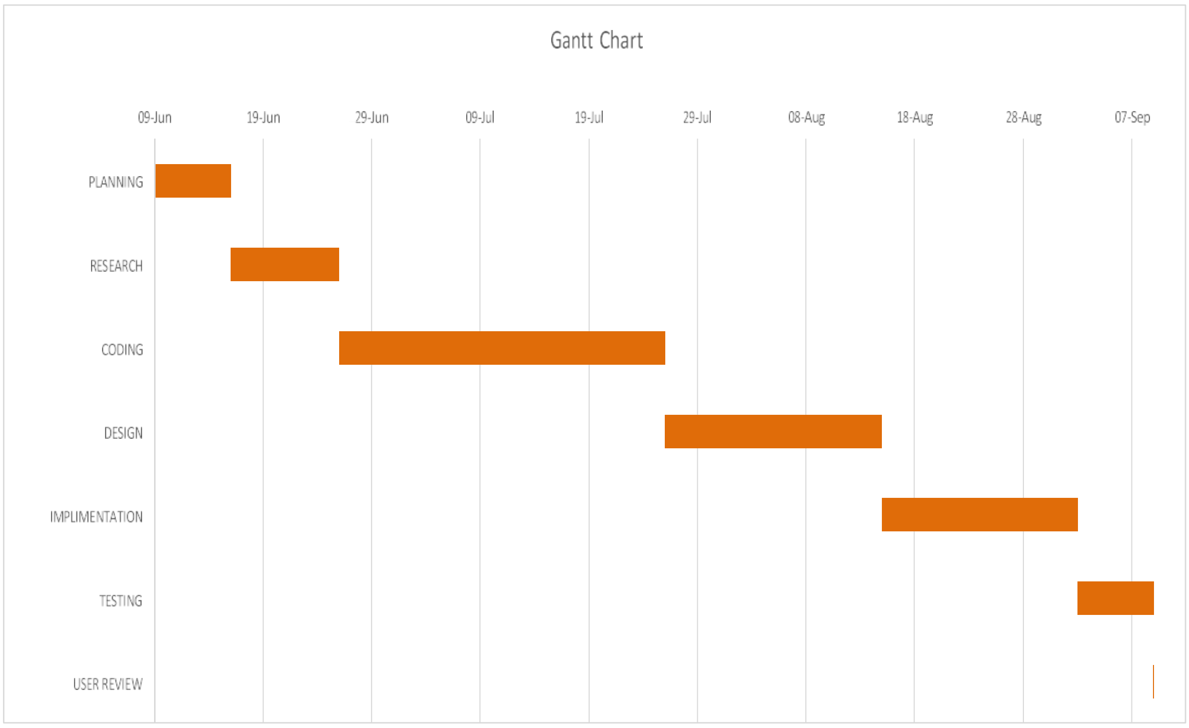
- **Technicians/Mechanics**: The frontline workers in the workshop who will directly interact with the online system. Their involvement and adoption of the platform are crucial for efficient service delivery and effective management of customer requests.

- **IT Department/Technical Team:** The internal technical team responsible for the development, implementation, and maintenance of the online system. They play a vital role in ensuring optimal performance, data security, and seamless integration with existing IT infrastructure.

- **Shipping and logistics providers**: Shipping and logistics providers are responsible for delivering comic to customers. They ensure that comic is delivered on time and in good condition.

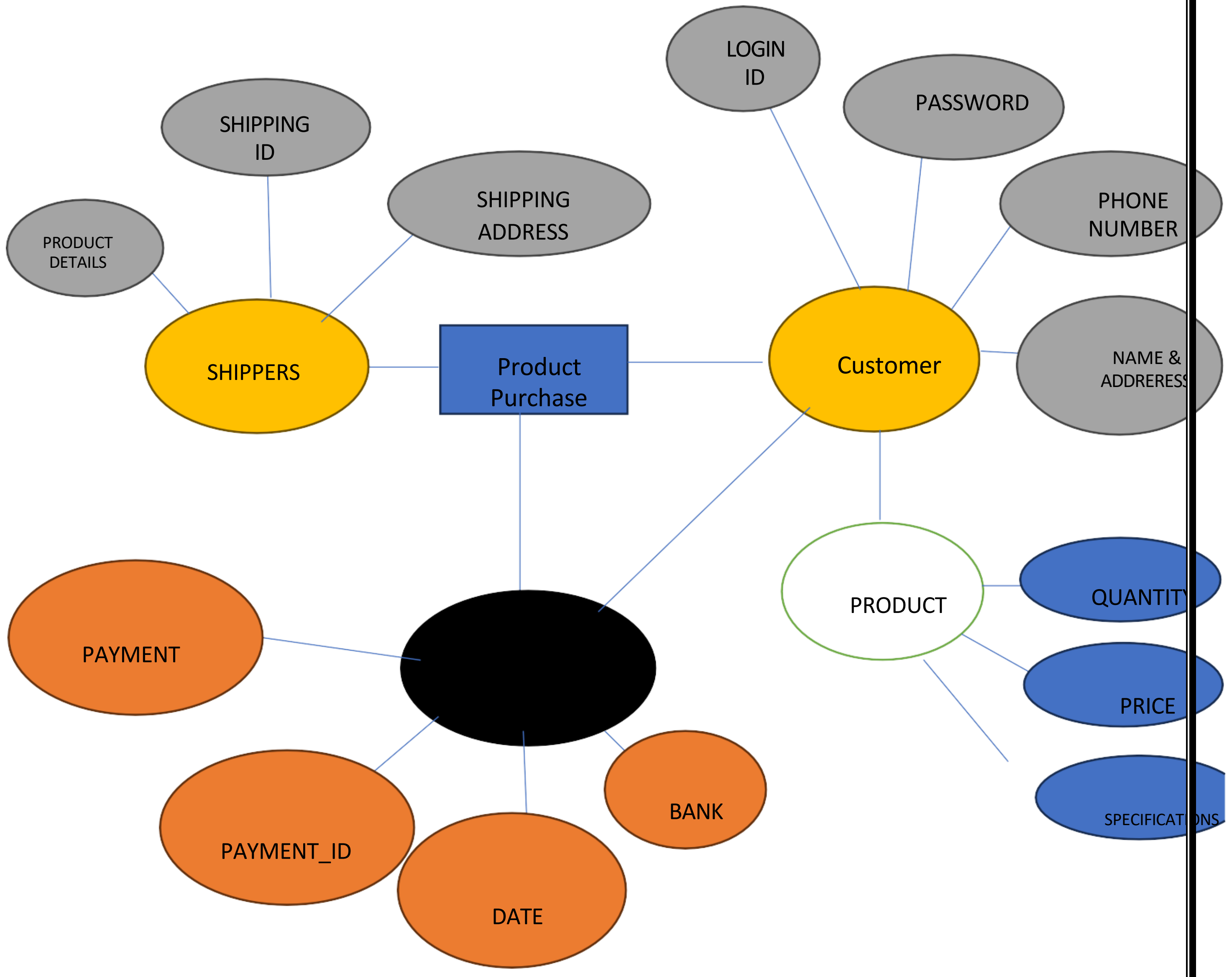
- **Marketing and advertising agencies:** Marketing and advertising agencies are responsible for promoting the website and its comic. They develop marketing strategies and campaigns to attract customers and increase sales.

3.3) Gantt Chart



4. SYSTEM DESIGN

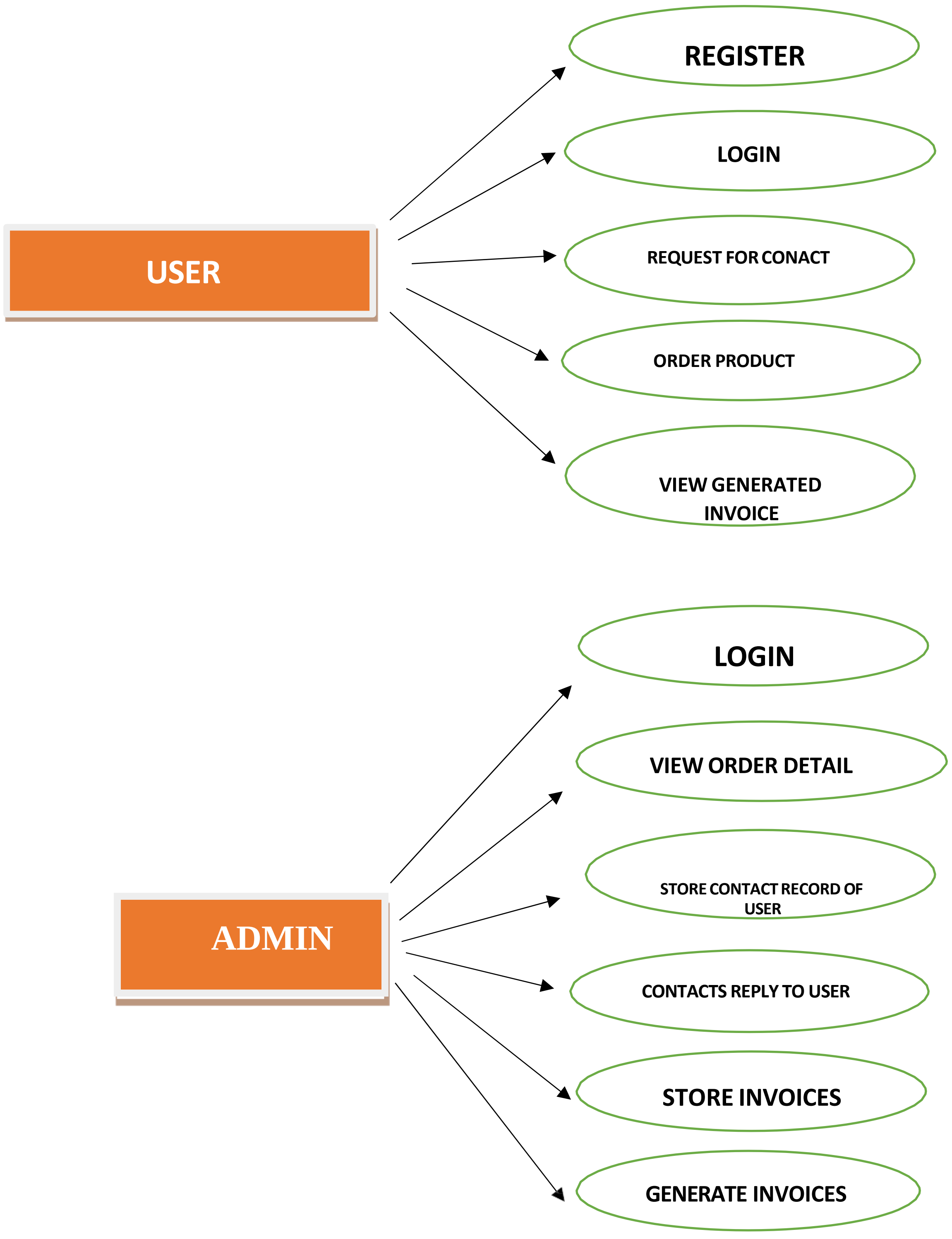
4.1) Entiy Relationship Diagram



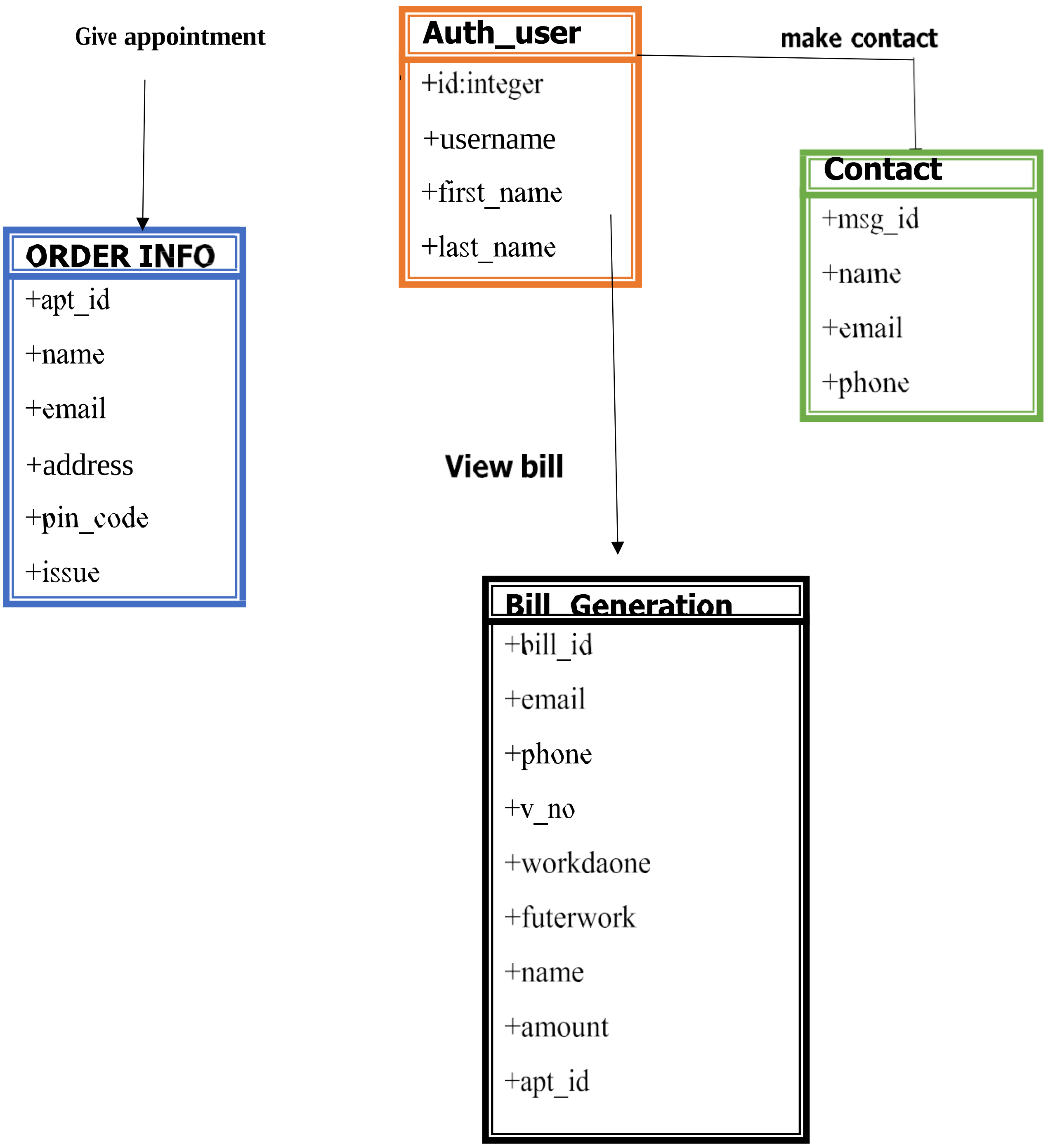
4.2) Event Table

EVENT	TRIGGER	SOURCE	ACTIVITY	RESPONSE	DESTINATION
Customer requests cancelation		Customer	Cancelation is requested	Cancelation is confirmed	management
Customer wants to Repeat order	Last order	Customer		Last order is repeated	Management
Yearly subscription plan is added	Subscription panel is added	Management	Subscription Plan details shared with customers	Customer buys the subscription	Customer
Promotional packages to be sent to customer	Package details	Marketing	Package details sent via sms	-	Customer
Report is requestedby customer	Report of the vehicle repair	Customer	Report is created		Management
Customer wants to change shipping address	Address	Customer	Oldaddress is deleted	Address is updated	Management
Late Shipment alert	Dates	Management	Customer is updated the delay in shipment		customer
Customer wants to cancel order	Order	Customer	Reason for cancelation	Cancelation is confirmed	
Customer requests invoice of earlier orders	Invoice registers	Customer	Invoices are created	Invoices are mailed	customer
Refund is requestedby customer	Refund amout	Management	Refund amount is calculated	Refund is successful	customer

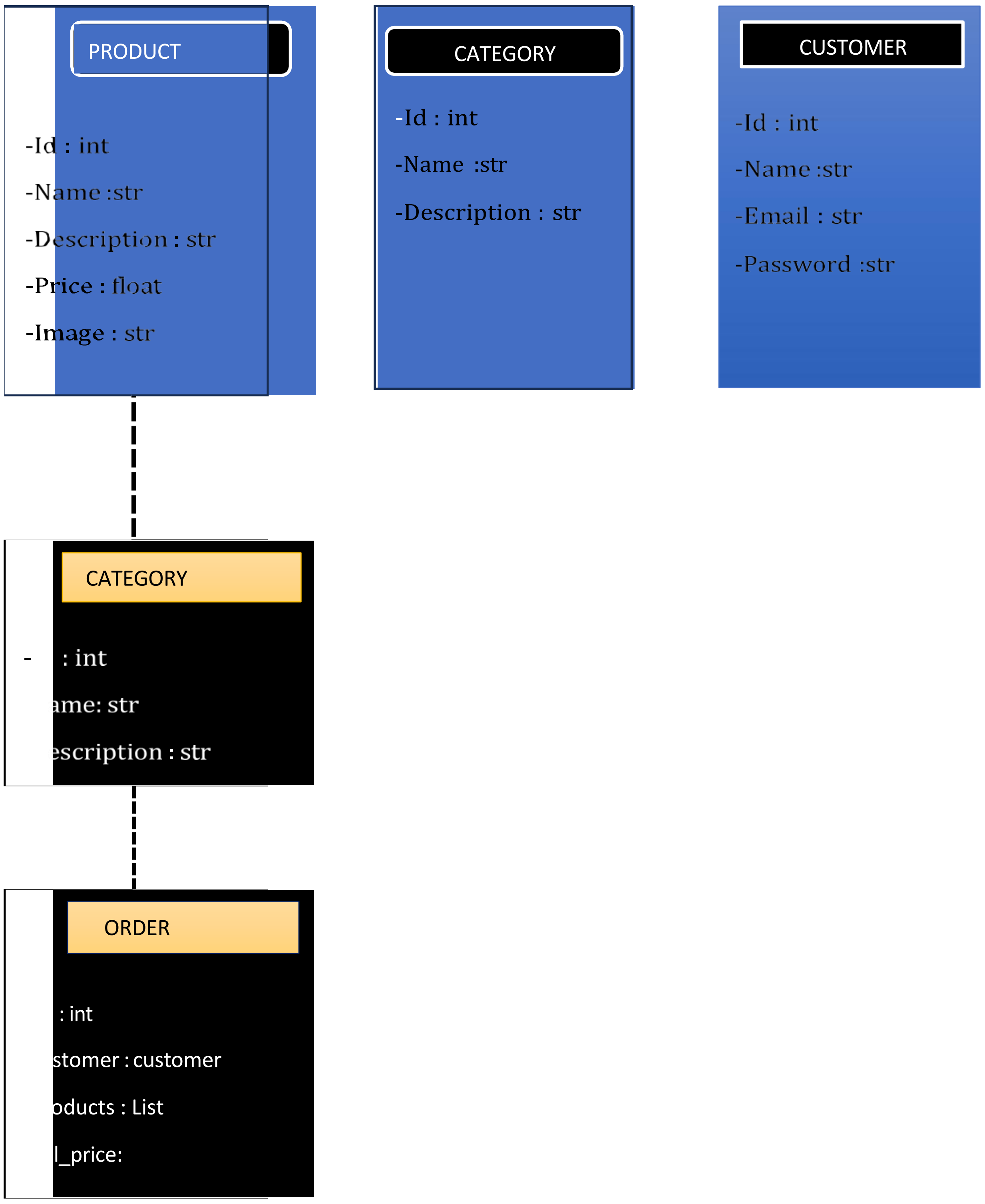
4.3) Use Case Diagram



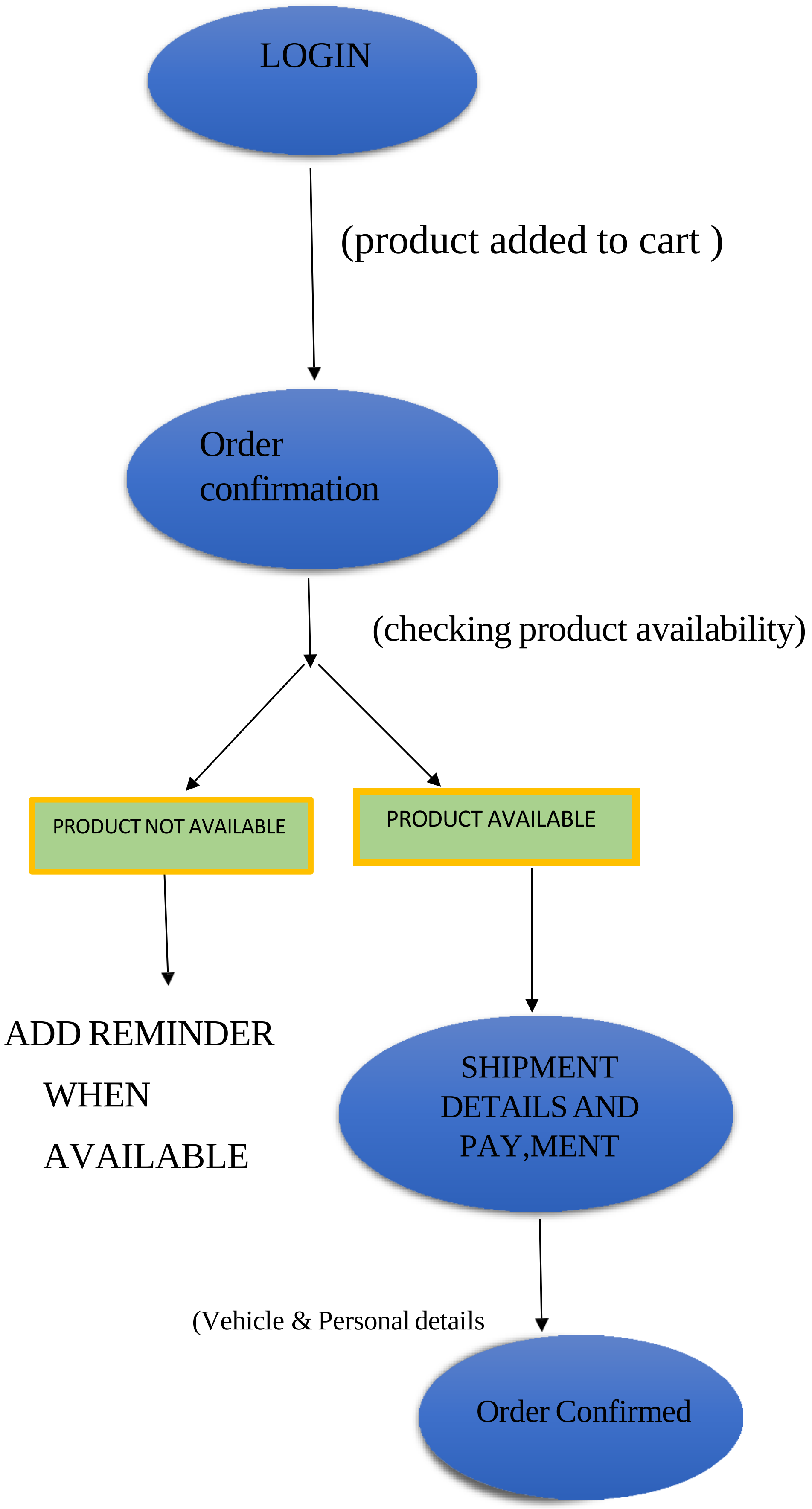
4.5) Class Diagram



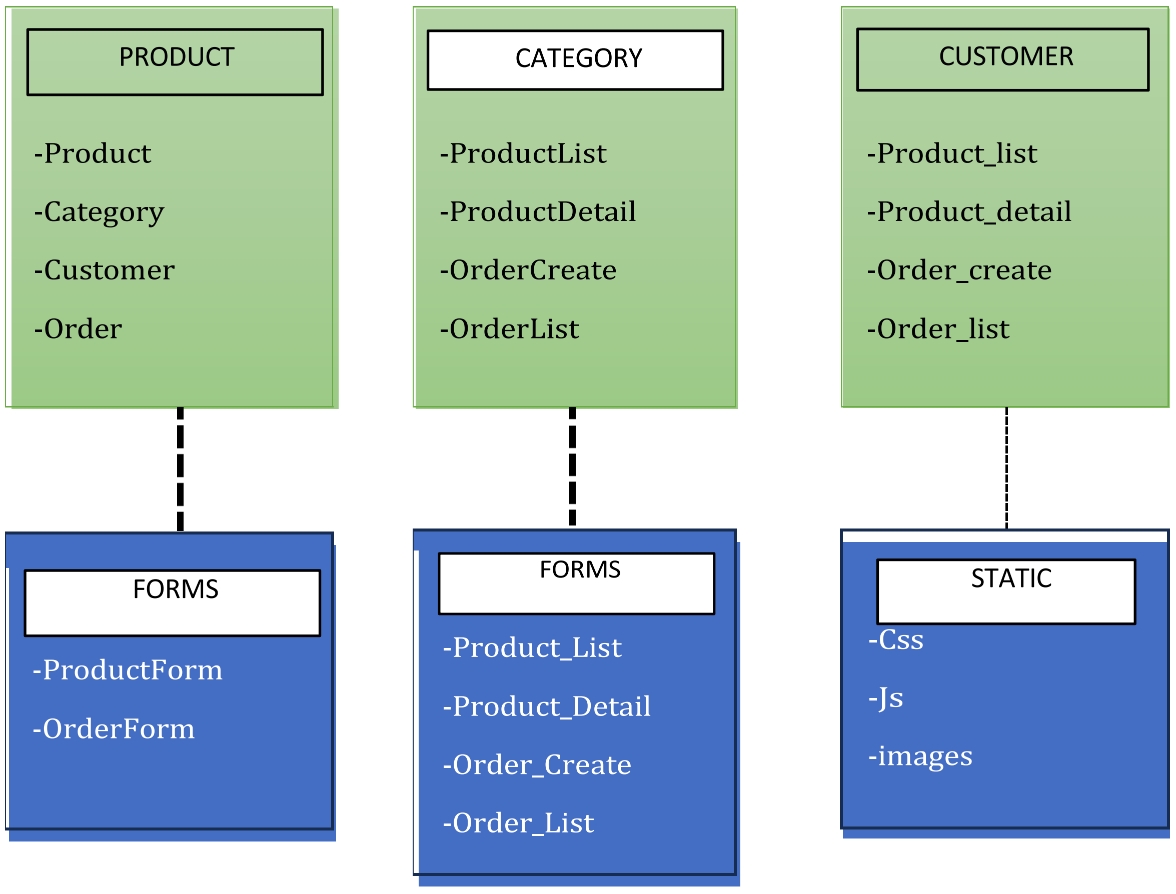
4.6) Object Diagram



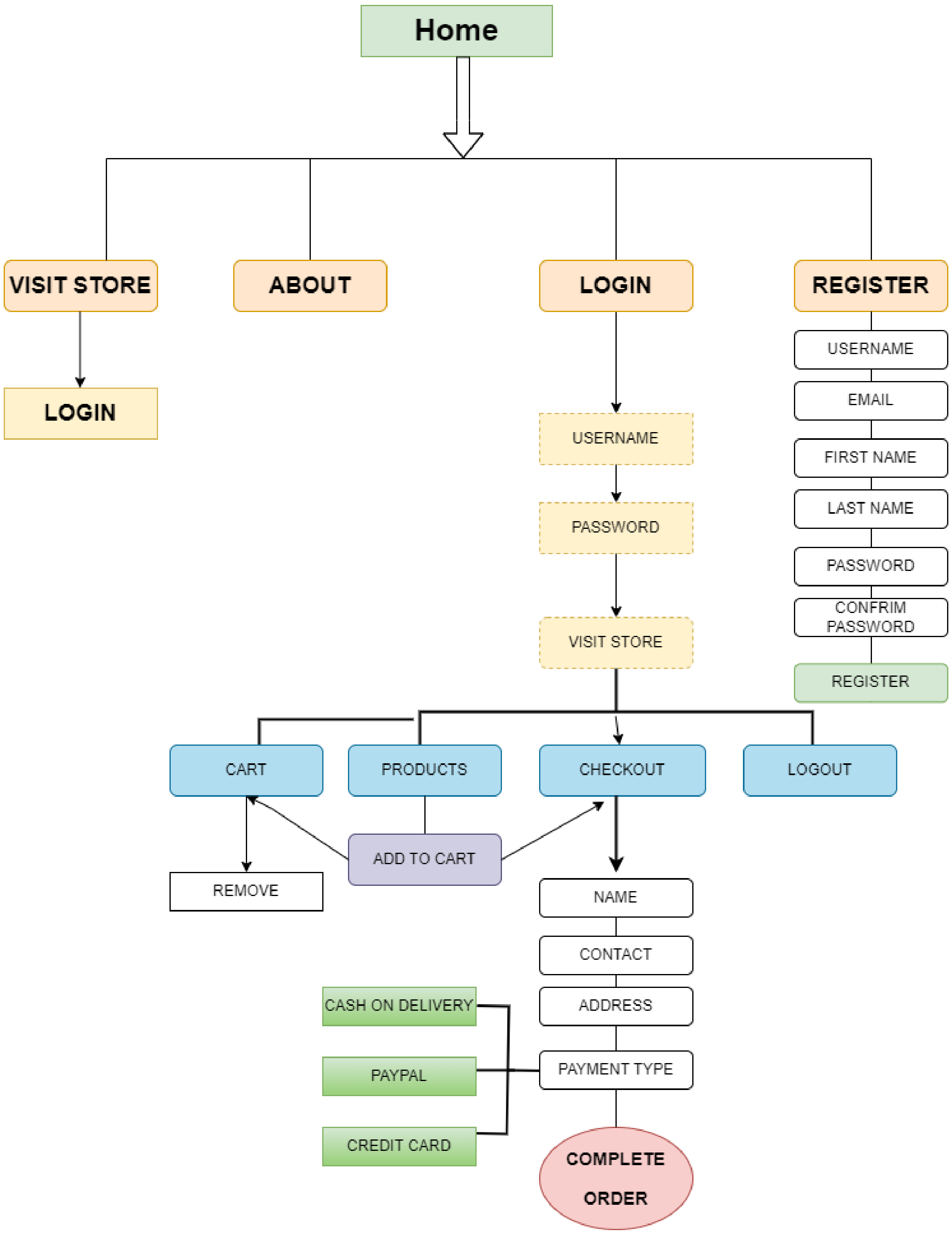
4.8) State Chart Diagram


LOGIN
(product added to cart)
Order confirmation
(checking product availability)
PRODUCT NOT AVAILABLE
PRODUCT AVAILABLE
ADD REMINDER WHEN AVAILABLE
SHIPMENT DETAILS AND PAY,MENT
(Vehicle & Personal details
Order Confirmed

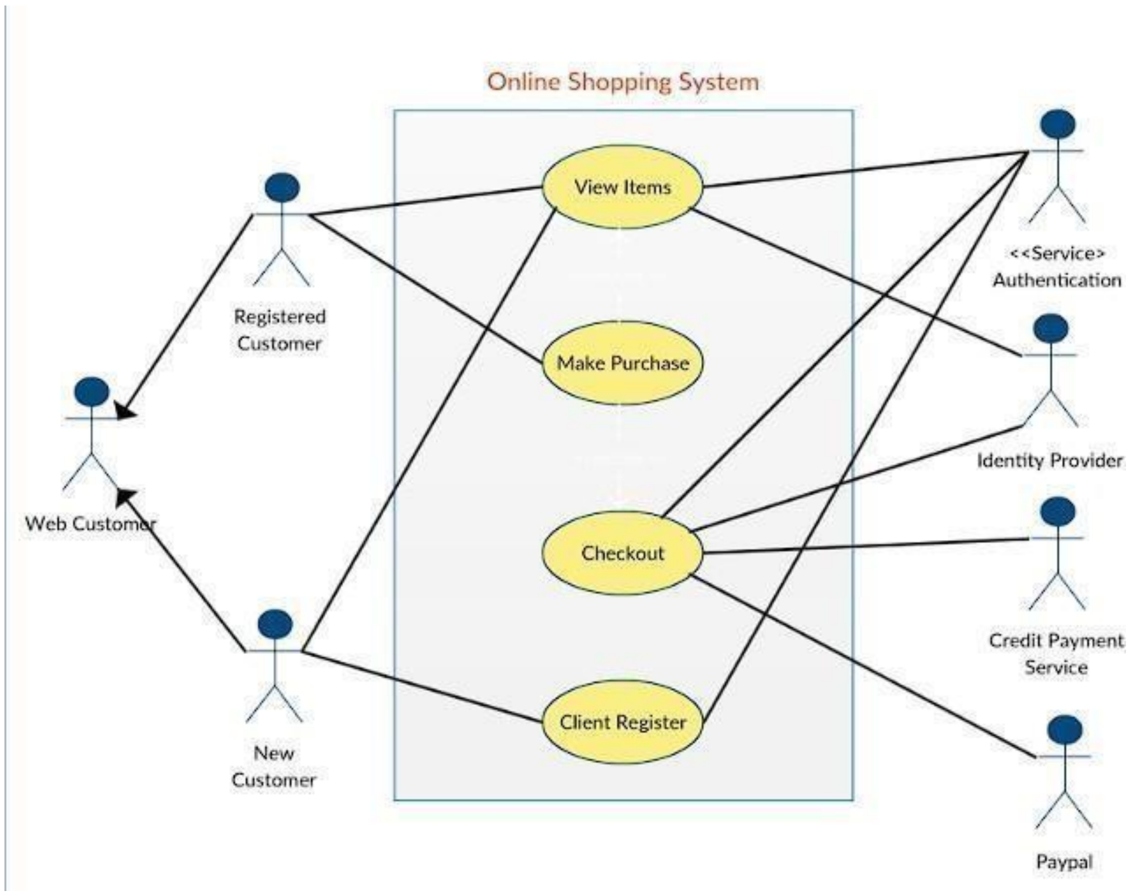
4.9) Package Diagram



MENU TREE


Home
VISIT STORE
ABOUT
LOGIN
REGISTER
LOGIN
USERNAME
EMAIL
FIRST NAME
LAST NAME
PASSWORD
CONFRIM PASSWORD
REGISTER
USERNAME
PASSWORD
VISIT STORE
CART
PRODUCTS
CHECKOUT
LOGOUT
ADD TO CART
REMOVE
NAME
CONTACT
ADDRESS
PAYMENT TYPE
CASH ON DELIVERY
PAYPAL
CREDIT CARD
COMPLETE ORDER

UML:


Online Shopping System
View Items
Make Purchase
Checkout
Client Register
Registered Customer
Web Customer
New Customer
<<Service> Authentication
Identity Provider
Credit Payment Service
Paypal

CHAPTER 4 TESTING APPROACH

4.2.1 UNIT TESTING (Test cases and Test Results)

4.2.2 INTEGRATION SYSTEM (Test cases and Test Results)

11.CODES

Base.html

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

<head>
    <!-- Required meta tags -->
    <meta  charset="utf-8">
    <metaname="viewport" content="width=device-width, initial-scale=1,  shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
        integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"   crossorigin="anonymous"
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
    <link rel="icon" type="image/png" href="{% static 'store/2.png' %}">
    <title>COMICHUB</title>
    <style>
        /* Additional CSS styles can be added here */
        /* For example, to customize the background color */ body {
            background-color:#f4f4f4;
        }

        .text-shadow {
            text-shadow: 0 0 10px rgba(255, 255, 255, 0.8);
        }
    </style>
</head>

<body>

    <nav class="navbar navbar-expand-md navbar-light" style="background-color: #f8f9fa;">
        <div class="container">
            <a class="navbar-brand d-flex align-items-center" href="{% url 'home' %}">
                <img src="{% static 'store/2.png' %}" width="100" height="100" class="d- inline-block align-top"
                    alt="Dokan Logo">
                <span class="ml-2 font-weight-bold text-shadow" style="font-size:  48px;">Comichub</span>
```

```
                    </a>
                    <button class="navbar-toggler" type="button" data-toggle="collapse" data- target="#navbarNav"
                        aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="collapse navbar-collapse" id="navbarNav">
                        <ul class="navbar-nav ml-auto">
                            <li class="nav-item">
                                <a class="nav-link" href="{% url 'cartview' %}">
                                    <img src="{% static 'store/cart.png' %}" width="30" height="30" class="d-inline-
                                        block align-top" alt="Cart Icon">
                                    <span class="text-dark">Cart :({{ cart_size }}) Items</span>
                                </a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link" href="{% url 'checkout' %}"><b class="text-
dark">Checkout</b></a>

                            <li class="nav-item">
                                <a class="btn btn-outline-dark my-2 my-sm-0" href="{% url 'about'
%}">About</a>
                            </li>

                            {% if user.is_authenticated %}

                            <li class="nav-item">
                                <a class="nav-link " href="{% url 'logout' %}">Logout</a>
                            </li>
                            {% else %}
                            <li class="nav-item {% block login %}{% endblock %}">
                                <a class="nav-link" href="{% url 'login' %}">Login</a>
                            </li>
                            <li class="nav-item {% block register %}{% endblock %}">
                                <a class="nav-link " href="{% url 'register' %}">Register</a>
                            </li>
                            {% endif %}
                        </ul>
                    </div>
                </div>
            </nav>




            <div class="container">
                {% block content %}{% endblock %}
```

```
        </div>

        <!-- jQuery, Popper.js, Bootstrap JS -->
        <script  src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
                integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
                crossorigin="anonymous"></script>
        <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-
                Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
                crossorigin="anonymous"></script>
        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-
                wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
                crossorigin="anonymous"></script>
</body>

</html>
```

Cart.html<br>

```
<styletype="text/css"> body {
        background-color: #252525; color:
        white;
    }

    .centerdv { width:
        80%;
        margin: 0 auto; text-
        align: center;
    }

    .imgscale {
        max-width: 64px;
        max-height: 64px;
    }

    table {
        border-collapse: collapse; width: 100%;
    }

    th,
    td {
        padding: 10px; text-
        align: auto;
        border-bottom: 1px solid #E6D9FE; color: white;
        /* Change text color to white */
    }

    tr:hover {
        background-color: #E6D9FE;
    }
</style>
<br>
<br>
<div class="container centerdv">
    <h2>Your Cart</h2>
    <div class="table-responsive">
        <tableclass="table table-bordered">
            <thead>
                <tr>
                    <th>Image</th>
                    <th>Product Name</th>
                    <th>Price</th>
                    <th>Remove</th>
                </tr>
            </thead>
            <tbody>
                {% for item in cart_item %}
                <tr>
                    <td><img class="imgscale" src="{{ item.image.url }}" alt="{{ item.name }}"></td>
                    <td>{{ item.name }}</td>
                    <td><spanclass="label label-primary"><b>Price: ${{ item.price
}}</b></span></td>
                    <td>
                        <form method="post" action="{% url 'removefromcart' %}">
                            {% csrf_token %}
                            <input type="hidden" name="obj_id" value="{{ item.id }}">
                            <buttontype="submit" class="btn btn-danger">Remove</button>
                        </form>
                    </td>
                </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
    <div class="text-center">
        <h3>Total Price: ₹{{ total }}</h3>
        <a href="{% url 'checkout' %}" class="btn btn-success">Checkout</a>
    </div>
</div>
{% endblock %}
```

Checkout.html

```
<styletype="text/css"> body {
        background-color: #252525; color: white;
    }

    .centerdv { width:
        60%;
        /* Adjust width to make it wider */ margin: 0 auto;
        text-align: center;
    }

    .imgscale {
        max-width: 64px;
        max-height: 64px;
    }
</style>

<divclass="container centerdv">
    <h3>Total Price :<br><spanclass="label label-success">
            {{ total }}</span></h3>
    <hr>
    <div class="well">
        <formmethod="post" action="{% url 'completeorder' %}" class="centerdv">
            {% csrf_token %}
            <h4>Billing Information</h4>
            <hr>
            <b>Name:</b>
            <inputtype="text" name="name" class="form-control">
            <br>
            <b>Contact No.:</b>
            <inputtype="text" name="phone" class="form-control">
            <br>
            <b>Address:</b>
            <input type="text" name="address" class="form-control">
            <br>
            <h4>Payment Information</h4>
            <hr>
            <b>Payment type:</b>
            <br>
            <select name="payment" class="form-control">
                <option value="Cash On Delivery">Cash on Delivery</option>
                <option value="Paypal">Paypal</option>
                <option value="credit card">Credit Card</option>
            </select>
            <br>
            <br>
            <b>Paypal Email/CC number [if applicable]</b>
            <br>
            <input type="text" name="payment_data" class="form-control">
            <br>
            <inputtype="submit" value="Complete Order" class="btnbtn-success">
        </form>
    </div>
</div>
```

Complete.html

```
<styletype="text/css"> body {
        background-color: #252525; color: white;
    }

    .centerdv { width:
        60%;
        margin: 0 auto; text-
        align: center;
    }

    .receipt { padding:
        20px;
        border: 1px solid#110d0d; border-
        radius: 5px; background-color:
        #3d3c3c;
    }
</style>

<divclass="container centerdv">
    <h3>Order Receipt</h3>
    <hr>
    <div class="receipt">
        <h4>Billing Information</h4>
        <hr>
        <p><b>Name:</b> {{ request.POST.name}}</p>
        <p><b>Contact No.:</b> {{ request.POST.phone }}</p>
        <p><b>Address:</b> {{ request.POST.address }}</p>
        <h4>Payment Information</h4>
        <hr>
        <p><b>Payment type:</b> {{ request.POST.payment }}</p>
        {% if request.POST.payment == "Paypal" %}
        <p><b>Paypal Email/CC number:</b> {{ request.POST.payment_data }}</p>
        {% endif %}
        <br>
        <p>
        <h3>Total Price :<br><spanclass="label label-success">
                {{ total }}</span></h3>
          </p>
       </div>
</div>
```

Home.html

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome to Our Anime Comic Store</title>
    <style> body{
            font-family: Arial, sans-serif; margin: 0;
            padding: 0;
            background-color: #1c1c1c;
            /* Darkbackgroundcolor */ color: #fff;
        }

        .header {
            background-color: #333; text-
            align: center; padding: 1rem;
        }

        .container {
            max-width: 1200px;
            margin: 0 auto;
            padding: 2rem;
        }

        .intro {
            text-align: center; padding: 4rem
            0;
        }

        .intro h1 {
            font-size:2.5rem; color:
            #FF9800;
            /* Orange color for anime theme */
            text-shadow: 0 0 10px rgba(255, 152, 0, 0.8);
            /* Glowing shadow effect */
        }

        .intro p {
            font-size: 1.2rem; max-
            width: 800px; margin: 0
            auto;
            text-shadow: 0 0 10px rgba(255, 255, 255, 0.1);
            /* Glowing shadow effect */
        }

        .shop-btn {
            display: inline-block; background-
            color: #FF9800;
            /* Orange color for anime theme */ color: #fff;
            padding: 1rem 2rem; font-
            size: 1.2rem; text-decoration:
            none; border-radius: 4px;
            transition: background-color 0.3s;
            box-shadow: 0 0 20px rgba(255, 152, 0, 0.4);
            /* Glowing shadow effect on hover */
        }

        .shop-btn:hover { background-color:
            #FFC107;
            /* Lighter orange color on hover */
        }
    </style>
</head>

<body>
    <div class="header">
        <img src="{% static 'store/1.jpg' %}" class="img-fluid" alt="Anime Logo" width="100%">
    </div>
    <div class="container">
        <div class="intro">
            <h1>Explore the World of Anime Merch</h1>
            <p>Indulge in our collection of anime comic and accessories. From your favorite series to
exclusive items,
                we have everything an anime fan could dream of.</p>
            <a href="{% url 'storem' %}" class="shop-btn">Visit the Store</a>
        </div>
    </div>
</body>

</html>
```

Merch.html

```
<html>
        <head>

        </head>

<p class="tip">Click on button in image container</p>
<div class="cont">
           <div class="form sign-in">
           <h2>Welcome back,</h2>
          <label>
            <span>Email</span>
            <input type="email" />
        </label>
        <label>
            <span>Password</span>
            <input type="password" />
        </label>
        <pclass="forgot-pass">Forgot password?</p>
        <buttontype="button" class="submit">Sign In</button>
        <button type="button" class="fb-btn">Connect with
<span>facebook</span></button>
    </div>
    <div class="sub-cont">
        <div class="img">
            <div class="img_text m--up">
                <h2>New here?</h2>
                <p>Sign up and discover great amount of new opportunities!</p>
            </div>
            <div class="img_text m--in">
                <h2>One of us?</h2>
                <p>If you already has an account, just sign in. We've missed you!</p>
            </div>
            <div class="img_btn">
                <span class="m--up">Sign Up</span>
                <span class="m--in">Sign In</span>
            </div>
        </div>
        <divclass="form sign-up">
            <h2>Time to feel like home,</h2>
            <label>
                <span>Name</span>
                <input type="text" />
            </label>
            <label>
                <span>Email</span>
                <input type="email" />
            </label>
            <label>
                <span>Password</span>
                <input type="password" />
            </label>
            <buttontype="button" class="submit">Sign Up</button>
            <buttontype="button" class="fb-btn">Join with
<span>facebook</span></button>
        </div>
    </div>
</div>

<a href="https://dribbble.com/shots/3306190-Login-Registration-form" target="_blank" class="icon-link">
    <img src="http://icons.iconarchive.com/icons/uiconstock/socialmedia/256/Dribbble-   icon.png">
</a>
<ahref="https://codepen.io/suez/pen/XWyBpre" target="_blank" class="link- footer">New 2023
Version</a>
<ahref="https://twitter.com/NikolayTalanov" target="_blank" class="icon-link  icon-link--twitter">
    <img src="https://cdn1.iconfinder.com/data/icons/logotypes/32/twitter-  128.png">
</a>
```

About.html

```
    <style>
          h3,
           p {
                padding-top: 2%;
          }

h1{
        font-style: italic rgb(248,248,248);
}
</style>



<div class="parallax" style='background-image :url("static/images/2.jpg");'>


                <!---header-wrapper--->
              <div class="container">
                      <div class="clear"></div>
                      <div class="page-content">
                            <div class="page">
                                    <div class="panel">
                                            <div class="title">

                                              <h1>SERVICES WE OFFER</h1>
                                       </div>
                                       <div class="content">




                                              <div class="card mb-3">
                                                      <imgclass="card-img-top" src="static/images/s4.jpg"  alt="Card
image cap">
                                                      <div class="card-body">
                                                          <h5 class="card-title">Shran Motors</h5>
                                                          <p class="card-text">Never Stop<br><br>We Let you to never
stop in your day to life because of the vehicle. So get Your Vehicle Serviced at our Garage and move without Stoping. </p>
                                                                      <p class="card-text"><small class="text-muted">Last updated 3
mins ago</small></p>
                                                      </div>
                                                  </div>




                                              <divclass="card-deck">
                            <div class="card">
                              <img  class="card-img-top" src="static/images/s1.jpg"
alt="Card image cap">
                              <div class="card-body">
                                <h5 class="card-title">Bike Repair</h5>
                                <p class="card-text">We Give You Best Bike Repair Service
at Reasonable Rates so whenerver you think that your bike needs a service We First Strike your mind.</p>
                              </div>
                            </div>
                            <div class="card">
                              <img  class="card-img-top"  src="static/images/s2"  alt="Card
image cap">
                              <div class="card-body">
                                <h5 class="card-title">4 Wheelers</h5>
                                <p class="card-text">Every Kind Of Car or 4 wheeler is
Repaired At our Garage , So you no need to go anywhere else for the service for your vehicle whether it is a Bus , Car or
any other.</p>
                              </div>
                            </div>
                            <div class="card">
                              <img  class="card-img-top" src="static/images/s3.jpg"
alt="Card image cap">
                              <div class="card-body">
                                <h5 class="card-title">Washing Centre</h5>
                                <p class="card-text">We have our own Washing Center so you
can get your vehicle Washed easily at a lower rate than others.<p>Normal Wash , Foam Wash</p></p>
                              </div>
                            </div>
                          </div>




                    </div>
                </div>
            </div>
            <div class="clear"></div>
        </div>
    </div>
</div>
```

Store.html

```
<styletype="text/css"> body {
        background-color: #0c0000;
        /* Dark background color */
    }

    .centerdv { width:
        45%;
        margin-left: auto;
        margin-right: auto; text-
        align: center;
    }

    .imgscale {
        max-width: 64px;
        max-height: 64px;
    }

    .sizef {
        font-size: 120px;
    }

    /* Add shadow glow effect to card borders */
    .card {
        box-shadow: 0 0 20px rgba(245, 221, 3, 0.6); border: none;
        background-color: #252525; color: #fff;
        margin-bottom: 20px;
        /* Add margin between cards */
    }

    /* Style the card title */
    .card-title {
        text-shadow: 0 0 10px rgba(0, 0, 0, 0.8);
    }
</style>

<h4 class="font-weighttext-center sizef" style="color: #0c0000; text-shadow:
0 0 20px rgba(245, 221, 3, 0.6);"> COMICHUB</h4>
<hr>

<div class="row">
    {% for product in store_item %}
    <divclass="col-lg-4 col-md-6">
```

```
        <div class="card">
            <img class="card-img-top" src="{{ product.image.url }}" alt="Card image">
            <div class="card-body">
                <h4 class="card-title">{{ product.name }}</h4>
                <hr>
                <p class="card-text">{{ product.description }}</p>
                <br>
                <span class="label label-primary"><b>Price : {{ product.price }}
</b></span>
                <br>
                <form method="post">
                    {% csrf_token %}
                    <input type="submit" value="Add to Cart" class="btn btn-primary">
                    <inputtype="hidden" name="obj_id" value={{ product.id}}>
                </form>
            </div>
        </div>
    </div>
    {% endfor %}
</div>
```

Login.html

```
<style> h2,
    h3, label
    {
        color: rgb(245, 237, 237); font-
        weight: bold;
    }
</style>

<div class="parallax" style='background-image :url("static/images/2.jpg");'>

    {% for message in messages %}
    <div class="alert alert-{{message.tags}} alert-dismissible fade show" role="alert">
        <strong>Message:</strong>{{message}}
        <button type="button" class="close rounded" data-dismiss="alert" aria- label="Close">
            <spanaria-hidden="true">&times;</span>
        </button>
    </div>
    {% endfor %}

     <divclass="container my-2">
         <div class="row justify-content-center">
             <divclass="title col-12 col-lg-8">
                <h2 style="font-weight: bold;" class="mbr-section-title align-center pb-3 mbr-fonts-style
display-4">
                    Log In Here
                </h2>
                <h3 style="color: rgb(240, 228, 228);">To Use our Service</h3>
            </div>
        </div>
    </div>
    <div class="container">
        <div class="row justify-content-center">
            <div class="media-container-column col-lg-8" data-form-type="formoid">
                <hr>
                <hr>

                <form class="mbr-form" action="/login" method="post" data-form- title="Mobirise Form">
                    {% csrf_token %}
                    <div class="row row-sm-offset">
                        <div class="col-md-6 multi-horizontal" data-for="loginusername">
                            <div class="form-group">
```

```
                                <label class="form-control-label mbr-fonts-styledisplay-7" for="loginusername-
form1-4">Username</label>
                                <inputtype="text" class="form-control" name="loginusername" data-form-
field="loginusername" required
                                    placeholder="Username" id="loginusername" />
                            </div>
                        </div>
                        <div class="col-md-6 multi-horizontal" data-for="password">
                            <div class="form-group">
                                <label class="form-control-label mbr-fonts-styledisplay-7" for="password-
form1-4">Password</label>
                                <input type="password" class="form-control" name="loginuserpassword" data-
form-field="password" required
                                    placeholder="password" id="loginuserpassword" />
                            </div>
                        </div>



                        <spanclass="input-group-btn"><buttontype="submit" class="btn btn-form btn-
warning display-4 my-3 mx-3">
                                Login
                            </button></span>

                    </div>
                </form>
                <hr>
                <hr>
            </div>
        </div>
    </div>
</div>



{% endblock %}


<style>
    /* Background and Text Styles */ body {
        background-color: #252525; color: #fff;
    }

    h2,
    h3, label
    {
        font-weight: bold;
```

```
}

/* Center the form */
.parallax {
    background-image: url("static/images/2.jpg"); background-
    size: cover;
    background-position: center; height:
    100vh;
    display: flex;
    align-items: center; justify-
    content: center; flex-direction:
    column; color: #fff;
    font-weight: bold;
}

/* Alert Messages */
.alert {
    margin-top: 10px; text-
    align: center; font-weight:
    bold;
}

/* Form Styling */
.mbr-form {
    background-color: rgba(0, 0, 0, 0.7);
    /* Semi-transparent blackbackground */ border-
    radius: 10px;
    padding: 20px;
    text-align: center;
    /* Center the form elements */
}

.form-grouplabel{ color:
    #fff;
    font-weight: bold;
}

/* Text Box Styles */
.form-control{ width:
    100%;
    /* Make text boxes 100% width */ padding: 10px;
    margin: 10px 0;
    background-color: rgba(255, 255, 255, 0.1);
    /* Semi-transparent white background */ border:
    none;
    border-radius: 5px;
```

```
        color: #fff;
        font-weight: bold;
    }

    /* Submit Button */
    .btn-form{
        background-color: #ff9800; color: #fff;
        font-weight: bold;
        border: none; padding:
        10px 20px; cursor:
        pointer; border-radius:
        5px;
    }

    /* Forget Password Link */
    .btn-danger {
        background-color: #dc3545; color:
        #fff;
        font-weight: bold;
        border: none; margin-
        top: 10px; padding: 5px
        10px; cursor: pointer;
        border-radius: 5px;
    }
</style>
```

Signup.html

```
</style>
<style>
  /*Global Styles */ body {
    background-color: #252525; color: #fff;
  }

  h2,
  h3, label
  {
    font-weight: bold;
  }

  /* Center the form */
  .parallax {
    background-image: url("static/images/2.jpg"); background-
    size: cover;
    background-position: center; height:
    100vh;
    display: flex;
    align-items: center; justify-
    content: center; flex-direction:
    column; color: #fff;
    font-weight: bold;
  }

  /* Alert Messages */
  .alert {
    margin-top: 10px; text-
    align: center; font-weight:
    bold;
  }

  /* Form Styling */
  .mbr-form {
    background-color: rgba(0, 0, 0, 0.7);
    /* Semi-transparent blackbackground */ border-
    radius: 10px;
    padding: 20px;
    text-align: center;
    /* Center the form elements */
  }

  .form-group label {
    color: #fff;
    font-weight: bold;
  }

  /* Text Box Styles */
  .form-control{ width:
    100%;
    /* Make text boxes 100% width */ padding: 10px;
    margin: 10px 0;
    background-color: rgba(255, 255, 255, 0.1);
    /* Semi-transparent white background */ border:
    none;
    border-radius:5px; color:
    #fff;
    font-weight: bold;
  }

  /* Submit Button */
  .btn-form{
    background-color: #ff9800; color: #fff;
    font-weight: bold;
    border: none; padding:
    10px 20px; cursor:
    pointer; border-radius:
    5px;
  }
</style>



<div class="parallax" style='background-image :url("static/images/2.jpg");'>


  {% for message in messages %}
  <div class="alert alert-{{message.tags}} alert-dismissible fade show" role="alert">
    <strong>Message:</strong>{{message}}
    <button type="button" class="close rounded" data-dismiss="alert" aria- label="Close">
        <spanaria-hidden="true">&times;</span>
    </button>
  </div>
  {% endfor %}
```

```
   <divclass="container my-2">
       <div class="row justify-content-center">
           <divclass="title col-12 col-lg-8">
               <h2 style="font-weight: bold;" class="mbr-section-title align-center pb-3 mbr-fonts-style
display-4">
                   Register Here
               </h2>
               <h3 style="color: rgb(248, 245, 245);">To Use our Service</h3>
           </div>
       </div>
   </div>
   <div class="container">
       <div class="row justify-content-center">
           <div class="media-container-column col-lg-8" data-form-type="formoid">
               <hr />

               <formclass="mbr-form" action="/register" method="post" data-form- title="Mobirise Form">
                   {% csrf_token %}
                   <div class="row row-sm-offset">
                       <div class="col-md-6 multi-horizontal" data-for="userid">
                           <div class="form-group">
                               <label class="form-control-label mbr-fonts-styledisplay-7"  for="userid-form1-
4">Username</label>
                               <input type="text" class="form-control" name="userid" data- form-
field="userid" required
                                   placeholder="username" id="userid" />
                           </div>
                       </div>

                       <div class="col-md-6 multi-horizontal" data-for="useremail">
                           <div class="form-group">
                               <label class="form-control-label mbr-fonts-styledisplay-7" for="useremail-
form1-4">Email</label>
                               <inputtype="email" class="form-control" name="useremail" data-form-
field="useremail" required
                                   placeholder="Email" id="useremail" />
                           </div>
                       </div>

                       <div class="col-md-6 multi-horizontal" data-for="username">
                           <div class="form-group">
                               <label class="form-control-label mbr-fonts-styledisplay-7" for="username-
form1-4">First Name</label>
                               <input type="text" class="form-control" name="username" data- form-
field="username" required
```

```
                                    placeholder="First Name" id="username" />
                                </div>
                            </div>
                            <div class="col-md-6 multi-horizontal" data-for="userlastname">
                                <div class="form-group">
                                    <label class="form-control-label mbr-fonts-styledisplay-7" for="userlastname-
form1-4">Last Name</label>
                                    <inputtype="text" class="form-control" name="userlastname" data-form-
field="userlastname" required
                                    placeholder="Last name" id="userlastname" />
                                </div>
                            </div>

                            <div class="col-md-6 multi-horizontal" data-for="password">
                                <div class="form-group">
                                    <label class="form-control-label mbr-fonts-styledisplay-7" for="password-
form1-4">Password</label>
                                    <inputtype="password" class="form-control" name="userpassword" data-form-
field="password" required
                                    placeholder="Password" id="userpassword" />
                                </div>
                            </div>
                            <div class="col-md-6 multi-horizontal" data-for="password">
                                <div class="form-group">
                                    <label class="form-control-label mbr-fonts-styledisplay-7" for="password-
form1-4"> Confirm
                                    Password</label>
                                    <inputtype="password" class="form-control" name="usercpassword" data-form-
field="password" required
                                    placeholder=" Confirm Password" id="usercpassword" />
                                </div>
                            </div>



                            <span class="input-group-btn"><button type="submit"
                                    class="btn btn-form btn-outline-primary display-4 my-3 mx-3"> Register
                                </button></span>
                        </div>
                    </form>
                </div>
            </div>
        </div>
</div>
```

Asgi.py

```
"""
ASGI config for estore project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/howto/deployment/asgi/ """

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'estore.settings') application =

get_asgi_application()
```

settings.py

```
"""
Django settings for estore project.

Generated by 'django-admin startproject' using Django 3.0.6.

For more information on this file, see https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/ """

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...) BASE_DIR =
os.path.dirname(os.path.dirname(os.path.abspath(  file_)))


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret! SECRET_KEY =
'mc@r^v6hi2$ibe=4#_#o4r9t)6k%8upkh_^xu_^*=l5*-c$1+a'

# SECURITY WARNING: don't run with debug turned on in production! DEBUG = True

ALLOWED_HOSTS = []


# Applicationdefinition  INSTALLED_APPS

= [
        'django.contrib.admin',
        'django.contrib.auth',
        'django.contrib.contenttypes',
        'django.contrib.sessions',
        'django.contrib.messages',
        'django.contrib.staticfiles', 'store',
]

MIDDLEWARE = [
        'django.middleware.security.SecurityMiddleware', 'django.contrib.sessions.middleware.SessionMiddleware',
        'django.middleware.common.CommonMiddleware', 'django.middleware.csrf.CsrfViewMiddleware',
        'django.contrib.auth.middleware.AuthenticationMiddleware',
        'django.contrib.messages.middleware.MessageMiddleware',
        'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF= 'estore.urls'

TEMPLATES = [
        {
                'BACKEND': 'django.template.backends.django.DjangoTemplates', 'DIRS': [],
                'APP_DIRS':True, 'OPTIONS': {
                        'context_processors': [ 'django.template.context_processors.debug',
                                'django.template.context_processors.request',
                                'django.contrib.auth.context_processors.auth',
                                'django.contrib.messages.context_processors.messages',
                        ],
                },
        },
]

WSGI_APPLICATION = 'estore.wsgi.application'


# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
        'default': {
                'ENGINE': 'django.db.backends.sqlite3', 'NAME':
                os.path.join(BASE_DIR, 'db.sqlite3'),
        }
}


# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password- validators

AUTH_PASSWORD_VALIDATORS = [
        {
                'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator'
,
        },
        {
                'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
        },
        {
                'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
        },
        {
                'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
        },
]


# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N   =   True

USE_L10N  =  True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
#  https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR,  'static')

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

Urls.py

```
"""estore URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home') Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home') Including another
URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
fromdjango.contrib import admin from
django.urls import path from store import
views
fromdjango.conf.urls.static import static from django.conf
import settings

urlpatterns = [
    path("", views.home, name="home"), path("admin/",
    admin.site.urls), path("storem/", views.storem,
    name="storem"), path("storem/", views.storem,
    name="storem"),
    path("cartview/", views.cartview, name="cartview"), path("login",
    views.hlogin, name="login"), path("logout", views.hlogout,
    name="logout"), path("register", views.register, name="register"),
    path("removefromcart/", views.removefromcart, name="removefromcart"), path("checkout/",
    views.checkout, name="checkout"), path("completeorder/", views.completeorder,
    name="completeorder"), path("about/", views.about, name="about"),
]

urlpatterns+=static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

wsgi.py

```
"""
WSGI config for estore project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/howto/deployment/wsgi/ """

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'estore.settings') application =

get_wsgi_application()
```

Migrations

```
# Generated by Django 3.0.6 on 2020-05-30 12:37 from django.db

import migrations, models


class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Order', fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False,
verbose_name='ID')),
                ('name', models.CharField(max_length=200)), ('address',
                models.TextField()),
                ('phone', models.CharField(max_length=100)), ('payment_method',
                models.CharField(max_length=400)), ('payment_date',
                models.DateTimeField(auto_now_add=True)),
            ],
        ),
        migrations.CreateModel(
            name='Product', fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False,
verbose_name='ID')),
                ('name', models.CharField(max_length=200)), ('price',
                models.FloatField()), ('description', models.TextField()),
                ('image', models.ImageField(upload_to='store/images')),
            ],
        ),
    ]
```

```
# Generated by Django 3.0.6 on 2020-05-30 12:51 from django.db

import migrations, models


class Migration(migrations.Migration):

    dependencies = [
        ('store', '0001_initial'),
    ]

    operations = [
        migrations.AddField(
            model_name='order', name='fullfilled',
            field=models.BooleanField(default=False),
        ),
        migrations.AddField(
            model_name='order',
            name='item',
            field=models.TextField(null=True),
        ),
    ]

# Generated by Django 3.0.6 on 2020-05-30 15:55 from django.db

import migrations, models


class Migration(migrations.Migration):

    dependencies = [
        ('store', '0002_auto_20200530_1851'),
    ]

    operations = [ migrations.AlterField(
            model_name='product', name='image',
            field=models.ImageField(upload_to='store/images/'),
        ),
    ]
```

Orderpaymentdata.py

```
# Generated by Django 3.0.6 on 2020-05-31 07:37 from django.db

import migrations, models


class Migration(migrations.Migration):

    dependencies = [
        ('store', '0003_auto_20200530_2155'),
    ]

    operations = [
        migrations.AddField(
            model_name='order', name='payment_data',
            field=models.TextField(null=True),
        ),
    ]
```

Admin.py

```
from django.contrib import admin from

.modelsimport Product,Order # Register your

models here.

admin.site.register(Product) admin.site.register(Order)
```

apps.py

```
fromdjango.apps import AppConfig class
StoreConfig(AppConfig):
    name = 'store'
```

models.py

```
from django.db import models


# Create your models here. class
Product(models.Model):
    name=models.CharField(max_length=200) price =
    models.FloatField()  description =
    models.TextField()
    image = models.ImageField(upload_to="store/images/")

    def __str__(self): return
        self.name


class Order(models.Model):
    name=models.CharField(max_length=200) address =
    models.TextField()
    phone = models.CharField(max_length=100) payment_method =
    models.CharField(max_length=400) payment_data =
    models.TextField(null=True) payment_date =
    models.DateTimeField(auto_now_add=True) item =
    models.TextField(null=True)
    fullfilled = models.BooleanField(default=False)

    def __str__(self): return
        self.name
```

views.py

```
from django.shortcuts import render, redirect from .models
import Product, Order
from django.contrib.auth import authenticate, login, logout from django.contrib
import messages
fromdjango.contrib.auth.models import User
fromdjango.contrib.auth.decorators import login_required


# Create your views here. def
hlogin(request):
    if request.method == "POST":
        loginusername = request.POST["loginusername"] loginpassword=
        request.POST["loginuserpassword"]

         user = authenticate(username=loginusername, password=loginpassword) if user is not None:
            login(request, user)
            messages.success(request, " Sucessfully Logged In") return redirect("/")
        else:
            messages.error(request, "Invalid Credentials, Please try again") # return redirect('/')
    returnrender(request, "login.html")


def hlogout(request):
    # if request.method == 'POST':
    logout(request)

    return redirect("home")


def register(request):
    if request.method == "POST":
        username = request.POST.get("userid") fname =
        request.POST.get("username") lname=
        request.POST.get("userlastname") email =
        request.POST.get("useremail")
        password = request.POST.get("userpassword") cpassword =
        request.POST.get("usercpassword")

        if not username:
            messages.error(request, " Username Required") elif len(username)
        < 4:
            messages.error(request, " Username must be under characters") elif
        User.objects.filter(username=username).exists():
```

```
                        messages.error(request, " Already User exist. Try Another
username")
                elif not fname:
                        messages.error(request, " First Name Required !!") elif len(fname) < 3 or
                len(fname) > 10:
                        messages.error(request, " First Name must be 4 char long or more") elif not lname:
                        messages.error(request, " Last Name Required..") elif len(lname)
                < 4 or len(lname) > 10:
                        messages.error(request, " Last Name must be 4 char long or more") elif len(email) < 5 or
                len(email) < 16:
                        messages.error(request, "Email must be 5 char long") elif
                User.objects.filter(email=email).exists():
                        messages.error(request, " Email Already exist. Try Another Email") elif not password:
                        messages.error(request, " Password Required") elif not
                cpassword:
                        messages.error(request, " Confirm Password Required") elif len(password) <
                6 or len(password) > 20:
                        messages.error(request, " Password must be 6 char long") elif password !=
                cpassword:
                        messages.error(request, " Password do not match")

                else:
                        myuser = User.objects.create_user(username, email, password) myuser.first_name =
                        fname
                        myuser.last_name=lname myuser.save()
                        messages.success(request, " Account has been sucessfully created") return
                        redirect("login")

        returnrender(request, "signup.html")


def home(request):
        if "cart" not in request.session: request.session["cart"]
                = []
        cart=request.session["cart"] request.session.set_expiry(0)
        ctx = {"cart": cart, "cart_size": len(cart)} returnrender(request,
        "store/home.html", ctx)


@login_required(login_url="login") def
storem(request):
        if "cart" not in request.session: request.session["cart"]
                = []
        cart=request.session["cart"] request.session.set_expiry(0)
```

```
    store_item = Product.objects.all()
    ctx = {"store_item": store_item, "cart_size": len(cart)} if request.method ==
    "POST":
        cart.append(int(request.POST["obj_id"])) return
        redirect("storem")
    return render(request, "store/store.html", ctx)


defcartitem(cart): items = []
    for item in cart: items.append(Product.objects.get(id=item))
    return items


def genItemsList(cart): cart_item=
    cartitem(cart) items_list = ""
    for item in cart_item: items_list+=
        str(item.name) items_list += ","
    return items_list


def totalcost(cart): cart_item=
    cartitem(cart) price = 0
    for item in cart_item: price +=
        item.price
    return price


def cartview(request):
    cart=request.session["cart"]
    request.session.set_expiry(0) ctx = {
        "cart": cart, "cart_size": len(cart),
        "cart_item": cartitem(cart), "total":
        totalcost(cart),
    }
    return render(request, "store/cart.html", ctx)


def removefromcart(request): request.session.set_expiry(0)
    obj_remove = int(request.POST["obj_id"])
    obj_index=request.session["cart"].index(obj_remove)  request.session["cart"].pop(obj_index)
```

```
        return redirect("cartview")


def checkout(request):
        request.session.set_expiry(0) cart=
        request.session["cart"]
        ctx = {"cart": cart, "cart_size": len(cart), "total": totalcost(cart)} return render(request,
        "store/checkout.html", ctx)


def completeorder(request):
        request.session.set_expiry(0) cart=
        request.session["cart"] order = Order()
        order.name = request.POST["name"] order.address =
        request.POST["address"] order.phone =
        request.POST["phone"] order.payment_method=
        request.POST["payment"]
        order.payment_data =request.POST["payment_data"] order.item =
        genItemsList(cart)
        order.save() request.session["cart"] =
        []
        returnrender(request, "store/complete.html")


def about(request): request.session.set_expiry(0)
        cart=request.session["cart"]
        ctx = {"cart": cart, "cart_size": len(cart)} returnrender(request,
        "store/about.html", ctx)


from django.shortcuts import render
from .models import Order  # Replace with your actual Order model


def orders(request):
        # Retrieve orders from the database or session, and pass them to the template
        orders = Order.objects.all() # Fetch orders using your model context = {"receipts":
        orders}
        return render(request, "store/orders.html", context)


from .models import Order


def order_history(request):
        orders = Order.objects.all().order_by("-timestamp")
        return render(request, "store/orders.html", {"orders": orders})
```

manage.py

```
#!/usr/bin/envpython
"""Django's command-line utility for administrative tasks.""" import os
import sys


def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',           'estore.settings')           try:
        fromdjango.core.managementimport execute_from_command_line except ImportError
    as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and " "available on your
            PYTHONPATH environment variable? Did you " "forget to activate a virtual
            environment?"
        )               from               exc
    execute_from_command_line(sys.argv)


if _name_== '_main_': main()
```

CHAPTER 5

RESULT AND DISCUSSIONS

Results:

The Comic Hub platform has been successfully structured into various modules that handle user management, comic cataloguing , digital reading, retail operations, and community engagement. Each module effectively contributes to a seamless user experience, ensuring that users can browse, read, purchase, and discuss comics efficiently.

Discussion:

The modular design of Comic Hub allows for scalability and easy maintenance. The integration of advanced features such as role-based access control, payment gateway, and digital comic reader enhances the overall user experience. However, potential challenges such as data security, content moderation, and performance optimization must be continuously monitored to maintain platform integrity and user satisfaction.

Future enhancements may include AI-driven recommendations, blockchain-based ownership verification for digital comics, and more advanced community engagement tools.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Comic Hub has successfully been designed as a multi-functional platform to cater to comic book enthusiasts, retailers, and publishers. Through its structured modules, it provides an engaging and efficient user experience. The implementation of user management, cataloguing , digital reading, and retail functionalities ensures that users can seamlessly browse, purchase, and enjoy their favorite comics. The platform also enhances community engagement through social features and discussions. With an organized data structure, Comic Hub can scale efficiently and maintain its robust performance.

However, challenges such as ensuring data security, managing high traffic loads, and moderating user content must be consistently addressed. The integration of advanced analytics, AI-driven recommendations, and blockchain for digital ownership could further improve Comic Hub's capabilities.

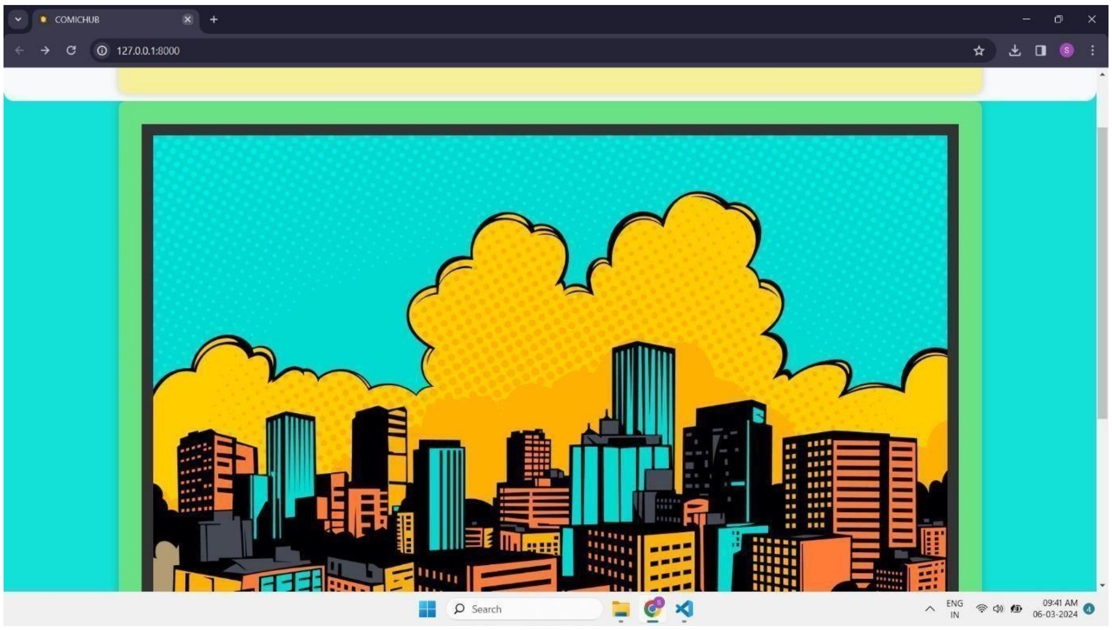
6.2 Future Work

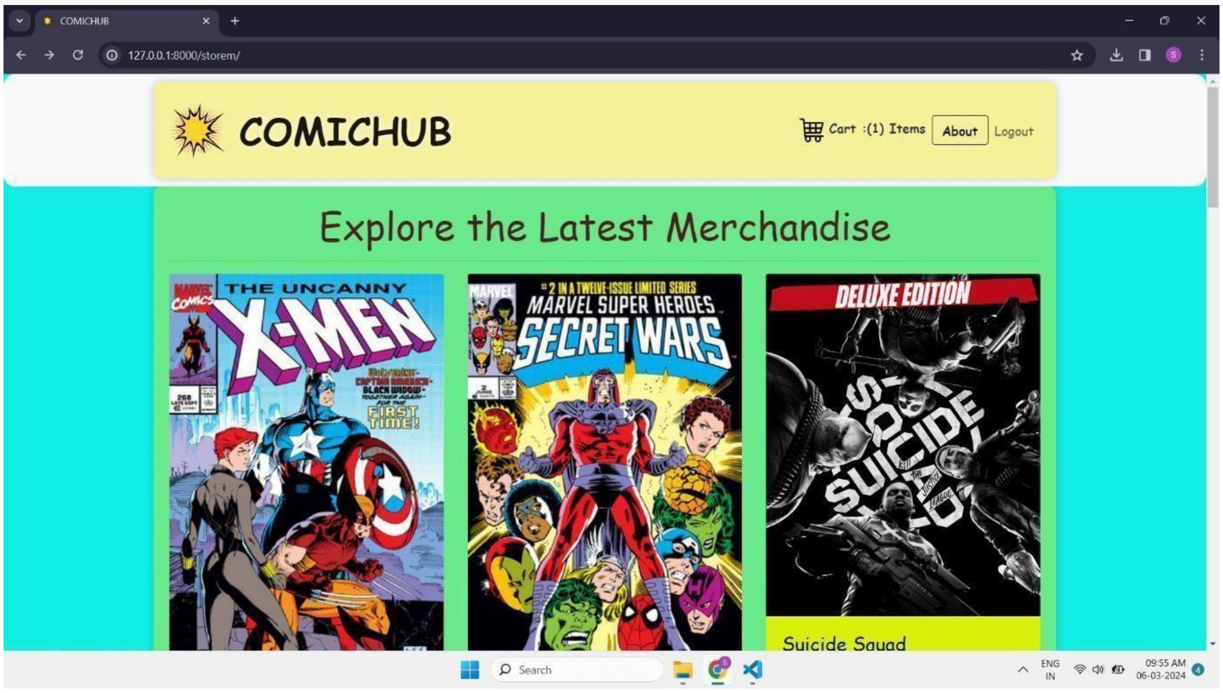
Future enhancements for Comic Hub include:

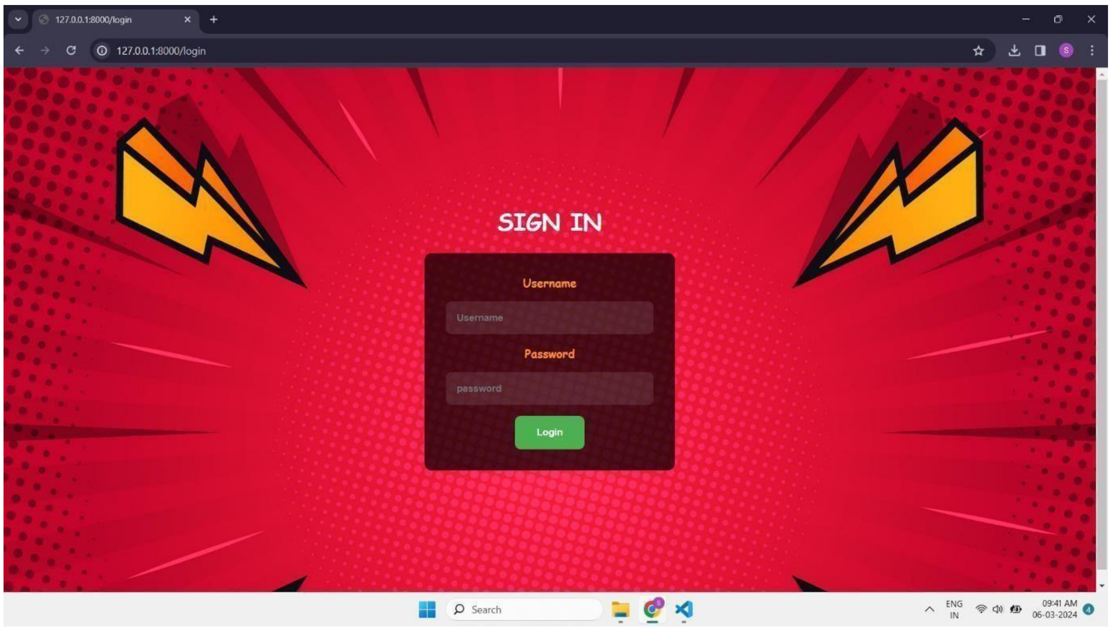
- **AI-powered personalized recommendations** to help users discover new comics.
- **Improved security measures**, including blockchain-based digital ownership tracking.
- **Expansion into mobile applications** for a better reading experience.
- **Integration of Augmented Reality (AR) features** for interactive storytelling.
- **Enhanced retailer support**, allowing direct publisher-to-retailer transactions.

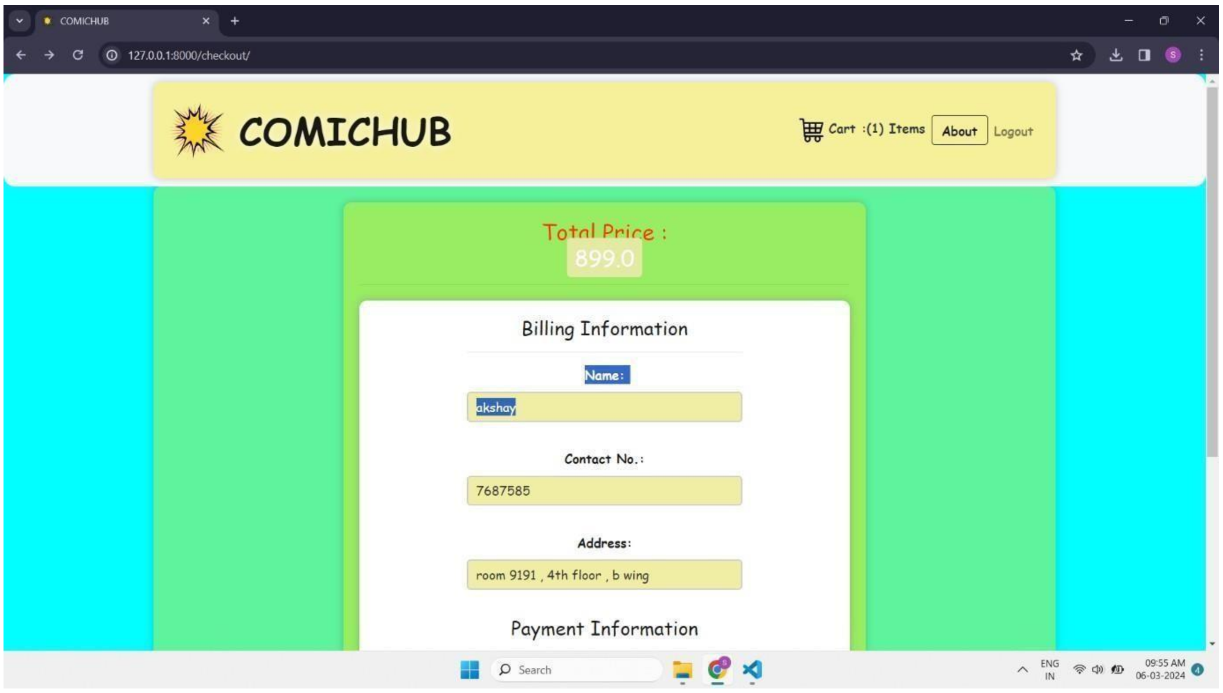
With these future enhancements, Comic Hub aims to revolutionize how comic book enthusiasts interact with digital content.

5. SCREEN LAYOUTS




COMICHUB
127.0.0.1:8000/storem/
COMICHUB
Cart :(1) Items
About
Logout
Explore the Latest Merchandise
Suicide Squad
Search
ENG IN
09:55 AM
06-03-2024


127.0.0.1:8000/login
127.0.0.1:8000/login
SIGN IN
Username
Username
Password
password
Login
Search
ENG IN
09:41 AM
06-03-2024



Total Price :
899.0

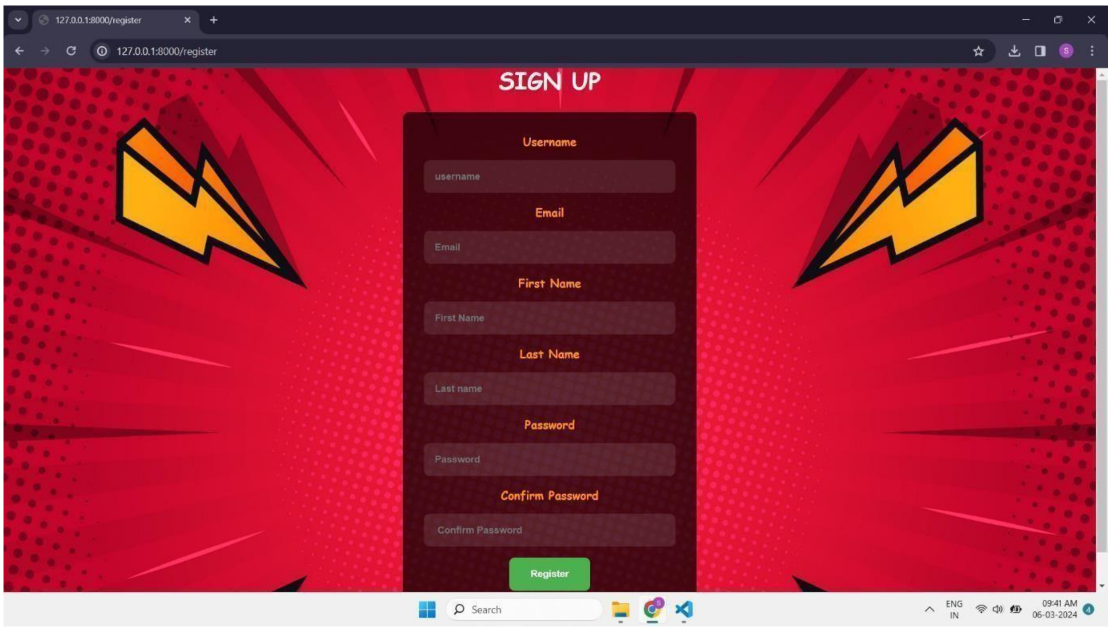
Billing Information

Name:
akshay

Contact No.:
7687585

Address:
room 9191 , 4th floor , b wing

Payment Information



SIGN UP

Username
username

Email
Email

First Name
First Name

Last Name
Last name

Password
Password

Confirm Password
Confirm Password

Register

CHAPTER 7 REFERENCES

1. "The Otaku Market in Japan: An Overview" by Patrick W. Galbraith, published in the Journal of Fandom Studies in 2014.
2. "Anime and Manga Fandom in the United States" by Andrea Wood, published in the Journal of Fandom Studies in 2016.
3. "The Anime Industry in Japan: A Brief Overview" by Rayna Denison, published in the International Journal of Cultural Policy in 2015.
4. "The Globalization of Anime: A Case Study of the U.S. Market" by Sharon Kinsella, published in the Journal of Popular Culture in 2000.
5. "The Business of Anime: A Comprehensive Overview" by Marc Hairston, published in the Journal of Media Economics in 2006.

WEBSITES:

1. *Crunchyroll Store - https://store.crunchyroll.com/*
2. *Tokyo Otaku Mode - https://otakumode.com/*
3. *Right Stuf Anime - https://www.rightstufanime.com/*
4. *AmiAmi - https://www.amiami.com/*
5. *J-List - https://www.jlist.com/*