A

PROJECT REPORT

ON

MEZBAANI CAFE FOR ONLINE FOOD ORDERING

DEVELOPED FOR

PASSION SOFTWARE SOLUTIONS

BY

TEJASWINI GOSAVI

FOR THE PARTIAL FULFILLMENT OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS

SUBMITTED TO

KCES's INSTITUTE OF MANAGEMENT AND RESEARCH, JALGAON

AFFILIATED TO

KAVIYATRI BAHINABAI CHAUDHARI

NORTH MAHARASHTRA UNIVERSITY, JALGAON

2024-2025

Certificate

This is to certify that **Ms. Tejaswini Kailasgir Gosavi** a student of MCA (Master of Computer Application) from Institute of Management and Research, Jalgaon has completed the full time Industrial Training with project titled **"Mezbaani Cafe for Online Food Ordering"** at **Passion Software Solutions.**

She has submitted satisfactory report in partial fulfillment of the requirement for the award of the degree of Master of Computer Application (MCA) during academic year 2024-2025.

Mrs. Rupali Narkhede	Mr. Uday Chatur
Internal Guide	Head, Dept. of MCA
Dr. Varsha Pathak	Prof. B. V. Pawar
Dean, School of Computer Applications	Director, IMR Jalgaon
External Examiner	External Examiner

Passion Software
SOLUTIONS
THE INSTITUTE OF SOFTWARE
DEVELOPMENT TRAINING CENTER, JALGAON
7972518721 9595544840

Ref No: 12/507 **Date:** 01/ 01 / 2025

To,

TEJASWINI KAILASGIR GOSAVI.

IMR College Jalgaon.

INTERNSHIP OFFER LETTER

Dear,

Passion Software Solution is pleased to offer you an Educational Python Internship Opportunity as a Software Development Internship. As per curriculum you will be receiving academic year for successful completion of this internship program. Additionally, you shall receive multiple benefits as a part of your internship program like, development of Technical Skills, Managerial Skills, Time Management Skills, Personality Development, Team Management Skills, Live Project Development.

Your schedule of training will be approximately of 25 to 30 hours per week beginning in **1st Jan 2025**. Your internship will conclude after completing 24 weeks of training.

Congratulations and welcome to the team.!

Mr. MANOJ A. KUMAVAT
Director
PASSION SOFTWARE SOLUTION

PASSION SOFTWARE SOLUTIONS Jalgaon

Mrs. APURVA M. KUMAVAT
HR
PASSION SOFTWARE SOLUTION

Behind Bank of Maharashtra, Near Gokul Hospital, Gandhi Nagar, Jilha Peth, Jalgaon.
infopassionsoftwares@gmail.com, www.passionsoftwares.com.

ACKNOWLEDGMENT

I would like to acknowledge all those without whom this project would not have been successful. Firstly, I would wish to thank my project guide **Mrs. Rupali Narkhede** mam who guided me throughout the project and gave his immense support, she made us understand how to successfully complete this project and without her, the project would not have been complete.

This project has been a source to learn and bright our theoretical knowledge to the real-life world. So, I would really acknowledge his help and guidance for this project.

I would also like to thank my parents who have always been there whenever needed. Once again, thanks to everyone for making this project successful.

TEJASWINI KAILASGIR GOSAVI

DECLARATION

I hereby declare that the work reported in this project work on "Mezbaani Cafe for Online Food Ordering System" submitted to Kavayitri Bahinabai Chaudhari North Maharashtra University is the record of authentic work carried out by me during the academic year 2024-2025 of the partial Requirement for the degree of Master of Computer Application (MCA) under the guidance of **Mrs. Rupali Narkhede**, KCES's Institute of Management and Research. The Material contained in the report has not been submitted to any University or Institution for the award of any degree.

TEJASWINI KAILASGIR GOSAVI

Index

Abstract

The purpose of Online Food Ordering System is to automate the existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with. The Online Food Ordering System's main purpose is to maintain rack of information such as Item Category, Food, Delivery Address, Order and Shopping Cart.

In today's fast-paced world, digitization of services has become essential for enhancing user convenience and streamlining business operations. This project is a web-based cafe management system designed to provide users with a seamless experience for browsing menus, placing online food orders, and booking tables in advance. Developed using Python and Django, this system aims to reduce customer wait time, optimize cafe management, and deliver a user-friendly interface for both customers and administrators. This report details the architecture, implementation methodology, system workflow, and key features of the Online platform.

The online food ordering system sets up a food menu online and customers can easily place the order as per their wish. Also with a food menu, customers can easily track the orders. This system also provides a feedback system in which user can rate the food items. Also, the proposed system can recommend Cafe, food, based on the ratings given by the user, the Cafe staff will be informed for the improvements along with the quality. The payment can be made online or pay-on-delivery system. For more secured ordering separate accounts are maintained for each user by providing them an ID and a password.

General Terms

Cloud Computing, Wi-Fi

Keywords

Automated Food Ordering System, Dynamic Database Management, Internet of Things, Smart Phones.

CHAPTER-01

INTRODUCTION

1.1 Introduction

The Mezbaani Cafe for Online Food Ordering system is web-based platform that allow customer to browse menu, place orders, and pay for food and beverages using their smartphones, tablets or computers. These systems have become increasingly popular in recent years, as more people are turning to online ordering for its convenience and ease of use. As the hospitality industry evolves, traditional food ordering and table booking methods are being replaced by digital platforms. Customers prefer convenience, speed, and reliability. This platform offers a modern approach to managing a cafe by integrating digital ordering and table reservations into a single, accessible web application.

The Online food ordering system typically consists of three main components: the customer-facing front-end, the cafe management back-end, and the payment gateway. The front-end interface allows customers to view manus, select items, customize orders, and make payments. The back-end management system enables restaurant owners to manage orders, track inventory, and manage customer data. The payment gateway is a secure payment processing system that allows customers to pay for their order online using credit cards, debit card or other payment methods.

Online food ordering systems provide a range of benefits for both customers and cafe owners. Customers can easily place orders from their homes or workplaces without the hassle of phone calls or waiting in line. They can also customize their orders and view real-time updates on the status of their orders.

1.2 Service Description

Mezbaani Cafe is a Django-based web application designed for “Online Food Ordering”. The services provided include:

- **Menu Browsing**: Customers can view a visually enriched menu categorized by food types, complete with item images, descriptions, and prices.
- **Cart and Checkout**: Customers can add items to a virtual cart, review their selections, and place an order using multiple payment options such as Cash on Delivery or online payment modes.
- **Table Booking**: A booking module allows users to reserve tables by selecting a date, time, and number of seats.
- **Order Management**: Orders are tracked and processed through a backend admin dashboard, ensuring accurate and timely fulfilment.

- **Admin Panel**: The admin interface provides features to manage categories, menu items, customer feedback, orders, and table reservations.

1.3 Reasons

An application that controls the entire business contributes to smooth management and makes it cost-efficient. Therefore, it is easier for you to grow your company faster, which may have taken longer with the normal way of investing time and making more expenditure. It can also ensure excellent customer services that helps to make the company more profitable leading to more purchases and loyal customers.

The key motivations for developing Mezbaani Cafe include:

- Minimizing manual errors in order and table management.
- Reducing customer wait time and improving satisfaction.
- Providing transparency in menu selection and pricing.
- Empowering business owners with admin-level insights.

Manage your inventory:

It is one of the key software required in any company that has a storage area. Inventory software is essential to help you maximize and control your inventory level, and to alert you against running out of stock. The Inventory management module of a cafe app knowns just how to handle your inventory and it takes care of your demand, inventory level and ensures that there is no lack of raw materials and demand.

Social Presence:

It doesn't matter how small or big is your cafe shop, having an online presence is the most powerful and remarkable tool one can possess while starting a business. Have a website or mobile app, connect your network with social platforms such as Facebook, Instagram, Twitter and more to promote and engage with tons of super fans.

1.4 Objectives

The objectives of an online food ordering system can vary depending on the specific needs and goals of the business. However, here are some common objectives that an online food ordering system can aim to archive:

1. To create a responsive online system for ordering and reservation.
2. To implement a secure, user-friendly portal for customers.
3. To simplify backend order and table management for cafe staff.
4. To provide real-time confirmation and recordkeeping for orders/ bookings.
5. Provide a convenient and user-friendly way for customers to order food online.
6. Streamline cafe operations by automating the ordering and payment process.
7. Increase sales and revenue by expanding the customer base and offering new sales channels.
8. Reduce errors and improve accuracy in order processing.
9. Gather customer data and feedback to improve service quality and marketing strategies.
10. Improving efficiency and reduce wait times for customers.
11. Implement loyalty programs to incentive repeat orders and increase customer retention.
12. Stay competitive in the food industry by keeping up with changing customer preferences and market trends.

Overall, the main objective of an online food ordering system is to provide a seamless and satisfying experience for customers while also helping restaurant owners streamline their operations and grow their business.

CHAPTER-02

SYSTEM ANALYSIS

2.1 Needs of Online Food Order and Table Booking

Helping customers in placing meal orders whenever they want. Customers will be able to order their preferred foods at any time, but as we've already mentioned, this is only a limited option. As a result, cafe need to have a specific system in place that will allow them to serve a large number of customers while streamlining operations. One of the best platforms is ordering, which offers all of these services in addition to a host of cutting-edge features that have helped countless small and large enterprises themselves as market leaders.

- Growing demand for online food services post-COVID-19.
- Enhancing customer experience with digital solutions.
- Avoiding overcrowding and miscommunication at cafes.
- Increasing order accuracy and transparency.
- Efficient order queue and resource management

2.2 Functionalities

- Provides search options based on a variety of criteria. Like Food Item, Customer, Order, and Order Confirmation.
- Online food ordering system also manages payment information for order details, order confirmation details and food items online.
- It keeps track of all the data regarding categories, Payments, Orders, etc.
- Manage the category's details.
- User Signup/Login
- Menu display with category filtering
- Cart management
- Order placement and tracking
- Table booking interface
- Admin management panel
- Order history per user
- Feedback collection
- Manage the order's information by combining all Confirm Order data.

2.3 Features

- Based on products and components.
- Django-based backend with SQLite.
- CSRF protection and form validation.
- Distinct admin and user roles.
- Easily creating and altering issues.
- User accounts are used to manage access and uphold security.
- Straightforward status & resolutions.
- Attachment & Additional Comments for more information.

2.4 Background

The development of Mezbaani stems from real-world challenges faced by cafes in managing orders and seating efficiently, especially during peak hours. This solution was conceptualized to merge traditional cafe service with digital innovation.

The research papers we considered while doing our analysis are listed below. In a wireless meal ordering system was designed and implemented together with consumer feedback for a restaurant. It makes it simple for restaurant operators to change menu presentations and set up the system in a Wi-Fi setting. The configurable wireless meal ordering system has linked a smart phone with real time customer feedback implementation to enable real-time contact between patrons of café and business owners.

2.5 Technology

The Mezbaani project is built using a combination of modern web development technologies, primarily Django for the backend and Bootstrap for the frontend. The development stack was chosen to ensure scalability, security, and ease of deployment.

2.5.1 Software Requirement

- **Frontend**: HTML5, CSS3, JavaScript, Bootstrap 4
- **Backend**: Python 3.x with Django Framework
- **Database**: SQLite (default Django database)
- **Browser**: Google Chrome, Mozilla Firefox
- **Editor/IDE**: Visual Studio Code or PyCharm
- **Operating System**: Windows 10/Linux

2.5.2 Hardware Requirement

- **Processor**: Intel i3 or higher
- **RAM**: Minimum 4 GB
- **Storage**: At least 2 GB of free disk space
- **Internet**: Required for deployment and updates
- **Display**: Minimum 1280×720 resolution

CHAPTER-03

FESIBILITY STUDY

3.1 Technical Feasibility

This aspect assesses whether the technology needed to develop and implement the system is available and practical.

- **Platform**: Python Django is a high-level, open-source web framework that encourages rapid development and clean, pragmatic design. It is well-suited for developing scalable, secure, and maintainable web applications like an online food ordering system.
- **Database**: PostgreSQL, MySQL, or SQLite (default with Django) can be used depending on the size and complexity of the application.
- **Hosting**: Platforms like Heroku, PythonAnywhere, or AWS provide excellent support for Django applications, enabling easy deployment and scalability.
- **Support Libraries**: Django REST framework, crispy forms, Celery for background tasks, and Bootstrap for responsive design offer extended functionality with active community support.
- **Skill Availability**: Developers skilled in Python and Django are readily available, and numerous learning resources exist, reducing the risk of technical issues.

3.2 Operational Feasibility

Operational feasibility considers whether the proposed system will function as intended in the real-world environment and whether it meets the user requirements.

- **Ease of Use**: The system offers intuitive navigation for users to browse menus, place orders, and track deliveries. Admins can easily manage menu items, orders, and customers.
- **Integration**: The system can integrate with payment gateways (e.g., Stripe, Razorpay), SMS/email notifications, and third-party delivery APIs if needed.
- **User Roles**: Separate dashboards for admins, restaurant staff, and customers ensure smooth operations and data flow.
- **Training**: Minimal training is required due to the user-friendly design and interface.

3.3 Economic Feasibility

This component evaluates whether the system is financially viable and cost-effective.

- **Development Costs**: Using open-source technologies like Python and Django significantly reduces development and licensing costs.
- **Maintenance**: Long-term maintenance is economical due to the robust and modular architecture of Django.
- **ROI**: By automating order processing, reducing manual errors, and enhancing customer convenience, restaurants can increase revenue and customer satisfaction.
- **Hosting and Domain**: Costs for domain registration and hosting (especially using platforms like Heroku or shared VPS) are minimal.

3.4 Legal Feasibility

This assesses whether the system conforms to all legal and regulatory requirements.

- **Data Privacy**: Django supports secure development practices, and compliance with privacy policies like GDPR can be ensured by following best practices in data storage and usage.
- **Digital Payments**: Proper integration with verified payment gateways ensures compliance with financial regulations.
- **Licensing**: Django is BSD-licensed, allowing commercial use without legal complexities.

3.5 Schedule Feasibility

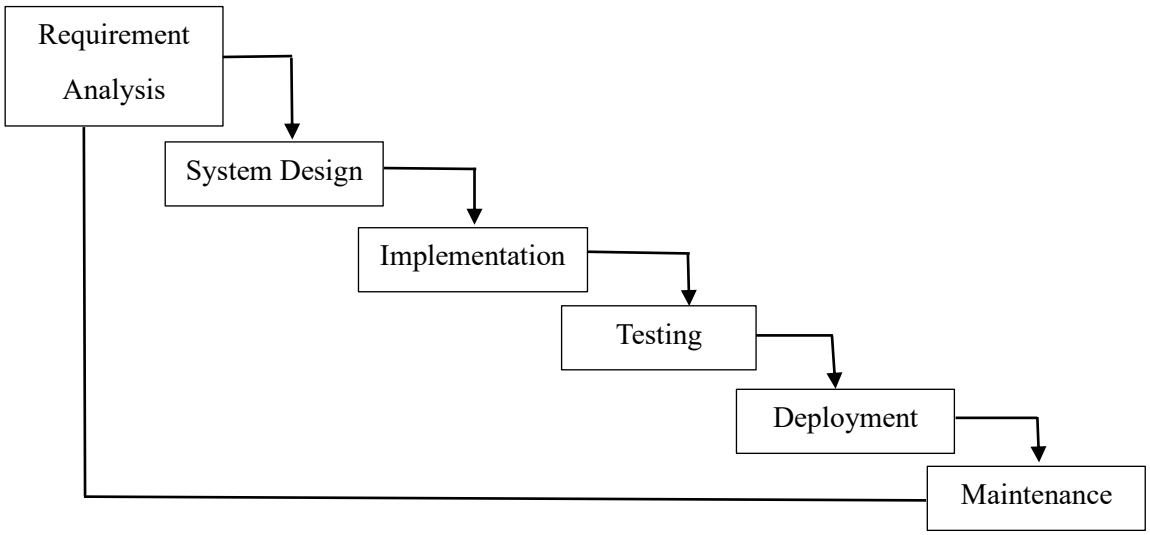
This evaluates whether the project can be completed within the available timeframe.

- **Development Timeframe**: With proper planning, the system can be developed within 8–12 weeks, depending on complexity.
- **Project Milestones**: Django's rapid development capabilities enable fast prototyping, early testing, and iterative improvements.
- **Team Productivity**: A small team can handle development, testing, and deployment efficiently using agile methodologies.

CHAPTER-04

SYSTEM DESIGN

4.1 Methodology Development Model



The Waterfall model's consecutive phases are:

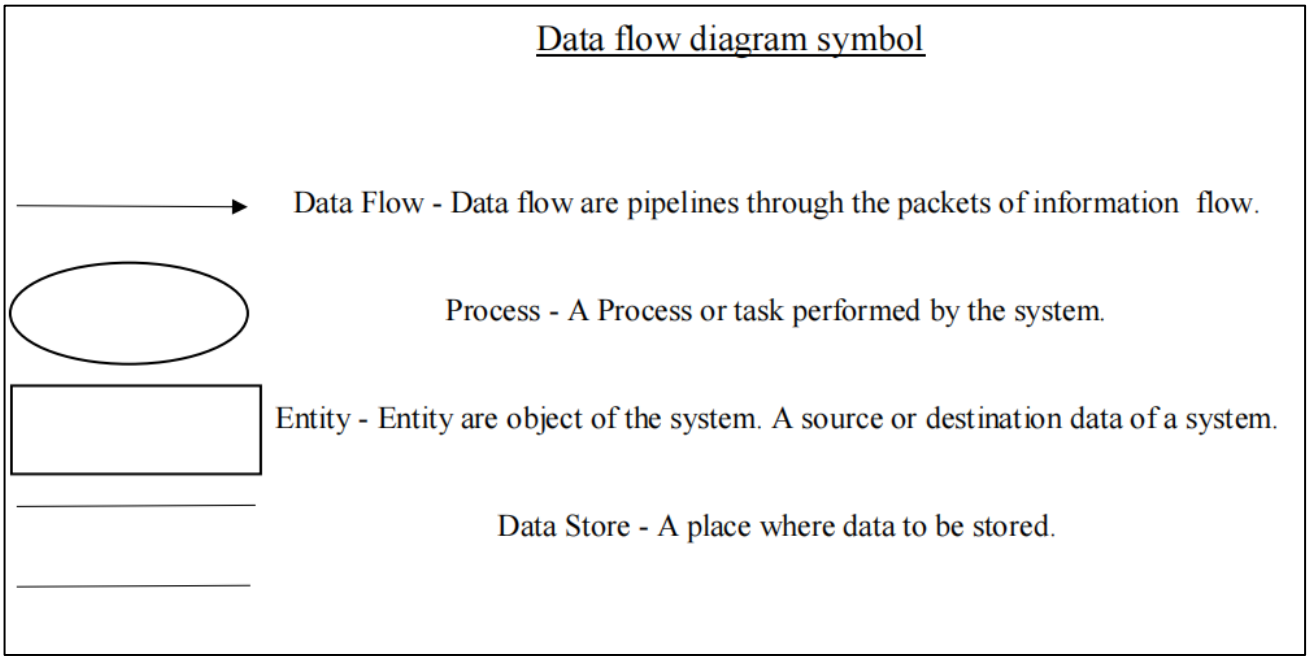
- **Requirement Gathering and analysis -**During this stage, all potential system needs are gathered and outlined in a requirement specification document.
- **System Design** -The System design is created in this phase after studying the requirement specifications from the first phase. This system design aids in determining the overall system architecture as well as the hardware and system requirements.
- **Implementation** -The system is initially built in discrete programs known as units, which are then combined in the following phase, using inputs from the system design. Unit testing is the process of developing and evaluating each unit for functionality.
- **Integration and Testing** -Following the testing of each unit created during the implementation phase, the entire system is merged. The entire system is tested for errors and failures after integration.
- **Deployment of system** -Once the product has undergone functional and non-functional testing, it is either published to the market or deployed in the customer's environment.
- **Maintenance** -Various problems can arise in a client environment. Patches are published to address certain problems. Additionally, improved versions of the product are issued. To bring about these changes in the surroundings of the consumer, maintenance is performed.

4.2 DATA FLOW DIAGRAM (DFD)

DFD is an important tool used by system analysis. A data flow diagram model, a system using external entities from which data flows to a process which transforms the data and create output data transforms which go to other processes or external entities such as files. The main merit of DFD is that it can provide an overview of what data a system would process.

SYMBOLS

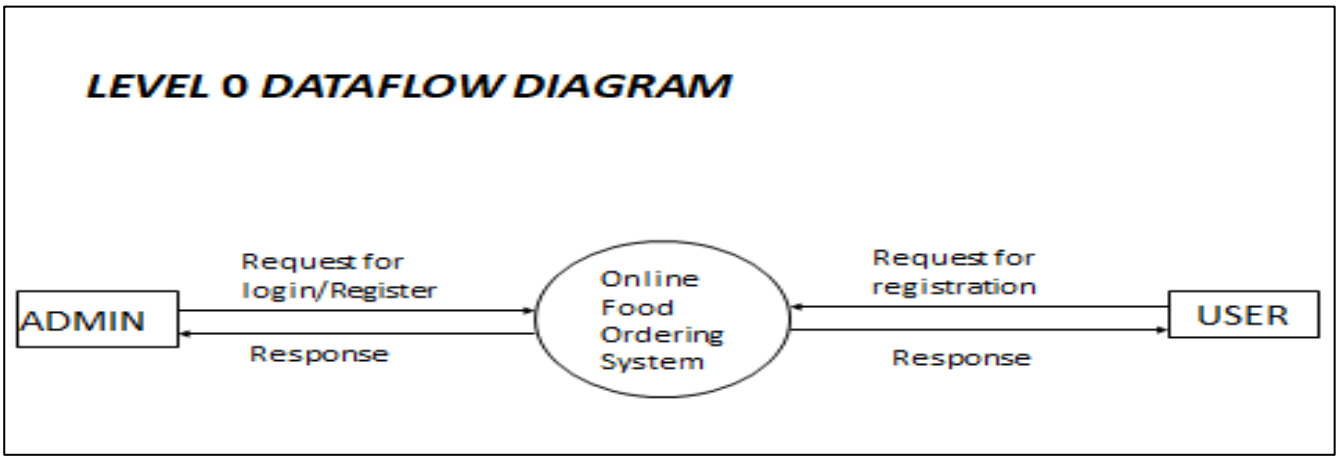
- A Circle represents a process that transforms incoming data flow into outgoing data flows.
- A Square defines a source or destination of system data.
- An Arrow identifies data flow direction. It is the pipeline through which the information flows.
- An Open Rectangle is a data store, data at rest or a temporary repository of data.



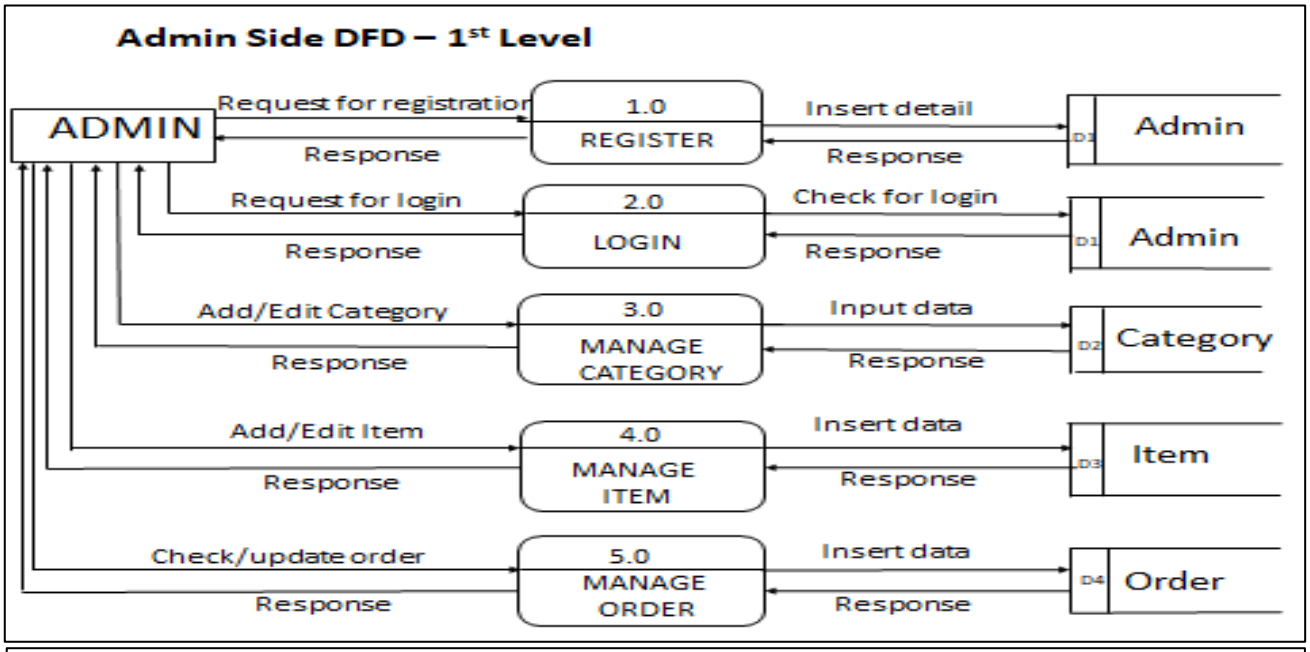
CONTEXT LEVEL DFD:

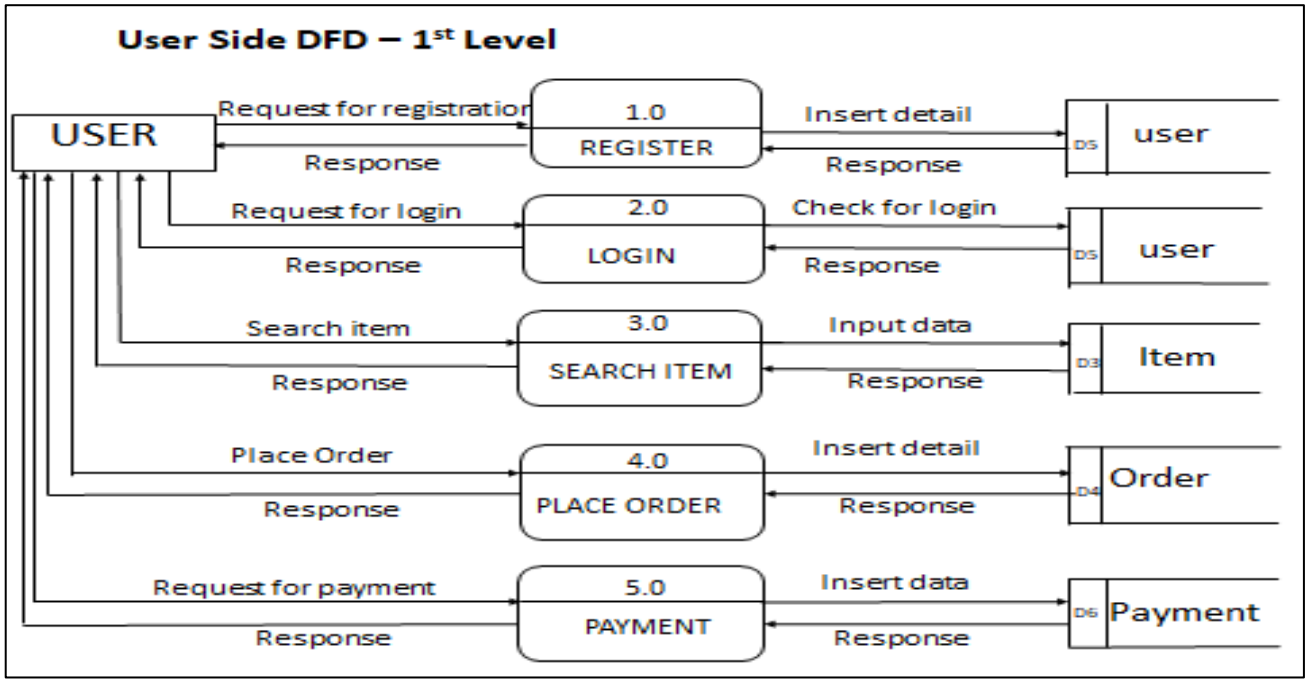
The context level data flow diagram (DFD) is Describe the whole system. The (0) level DFD describe the all-users module who operate the system. Below data flow diagram of online food ordering system site shows the two User can operate the system Admin and Member user.

4.2.1 DFD 0 Level

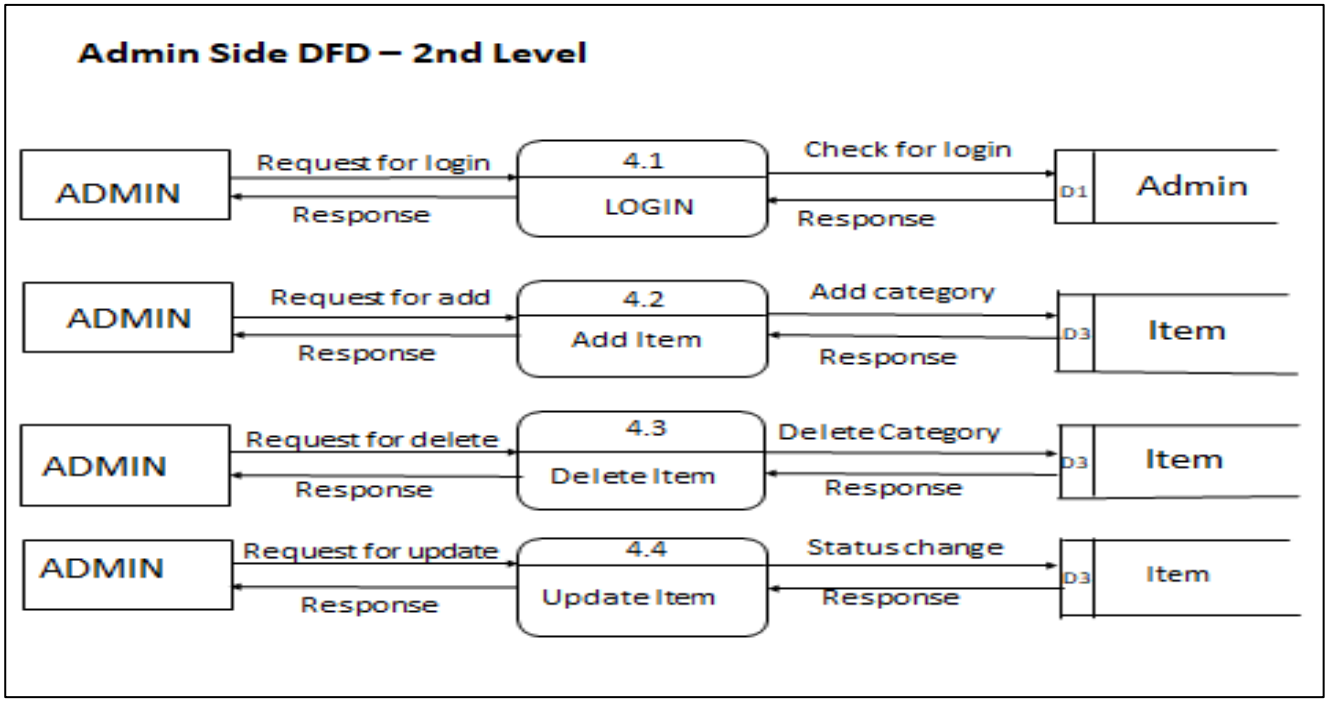


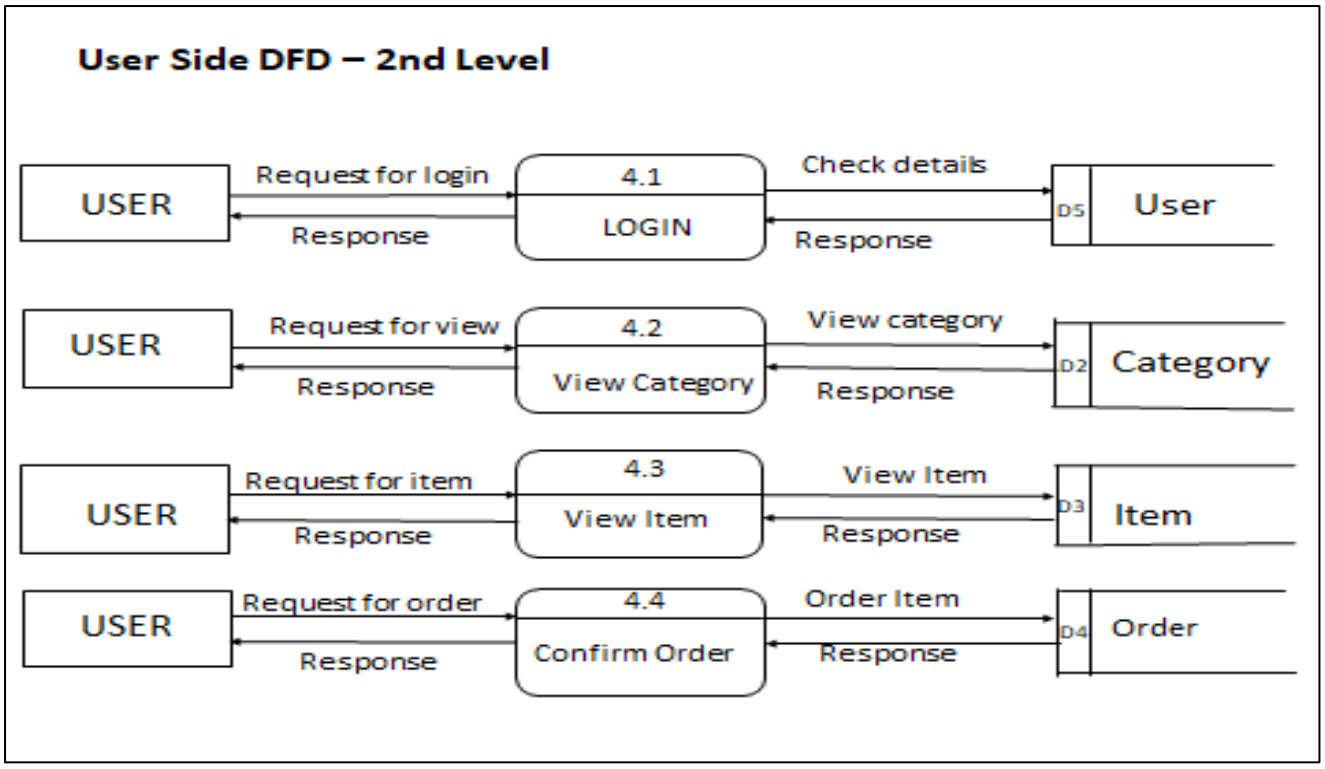
4.2.2 DFD 1 Level



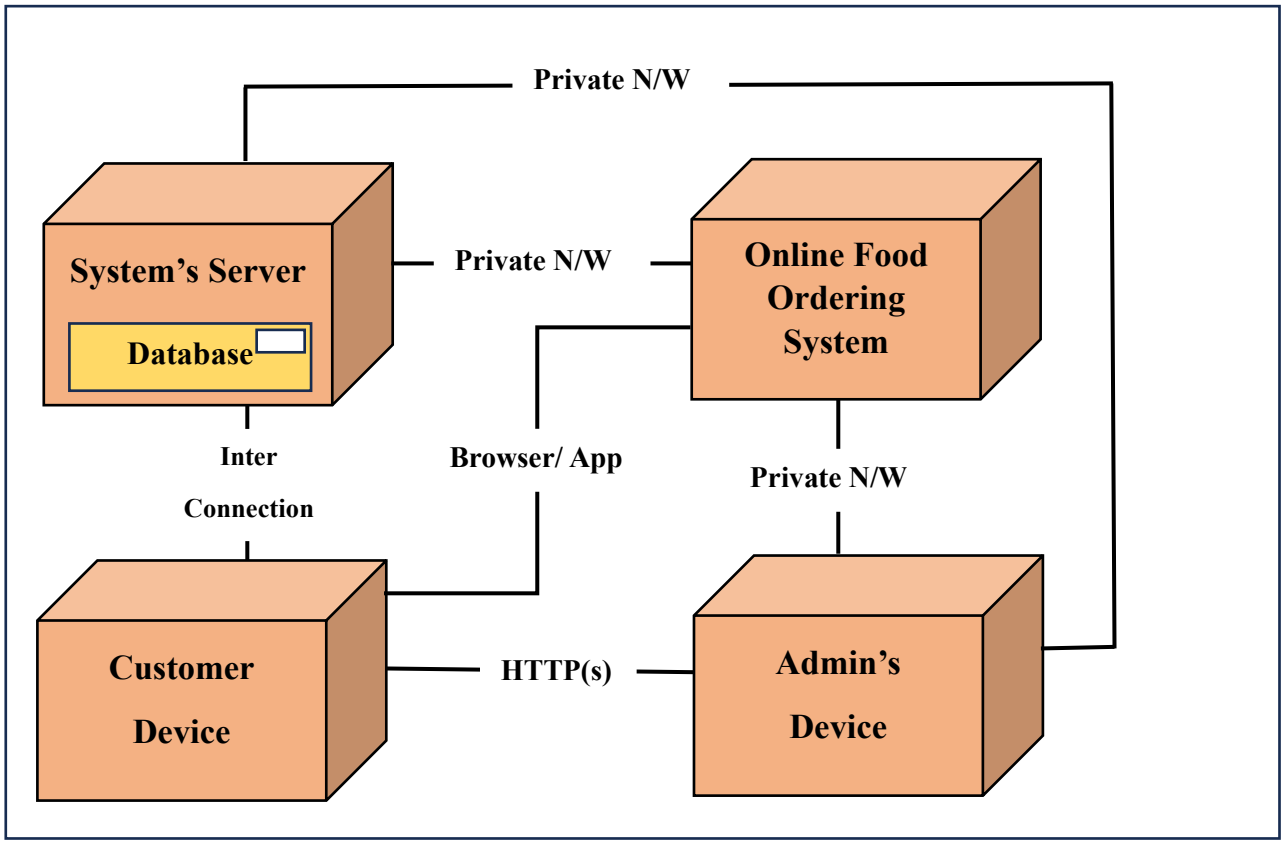


4.2.3 DFD 2 Level



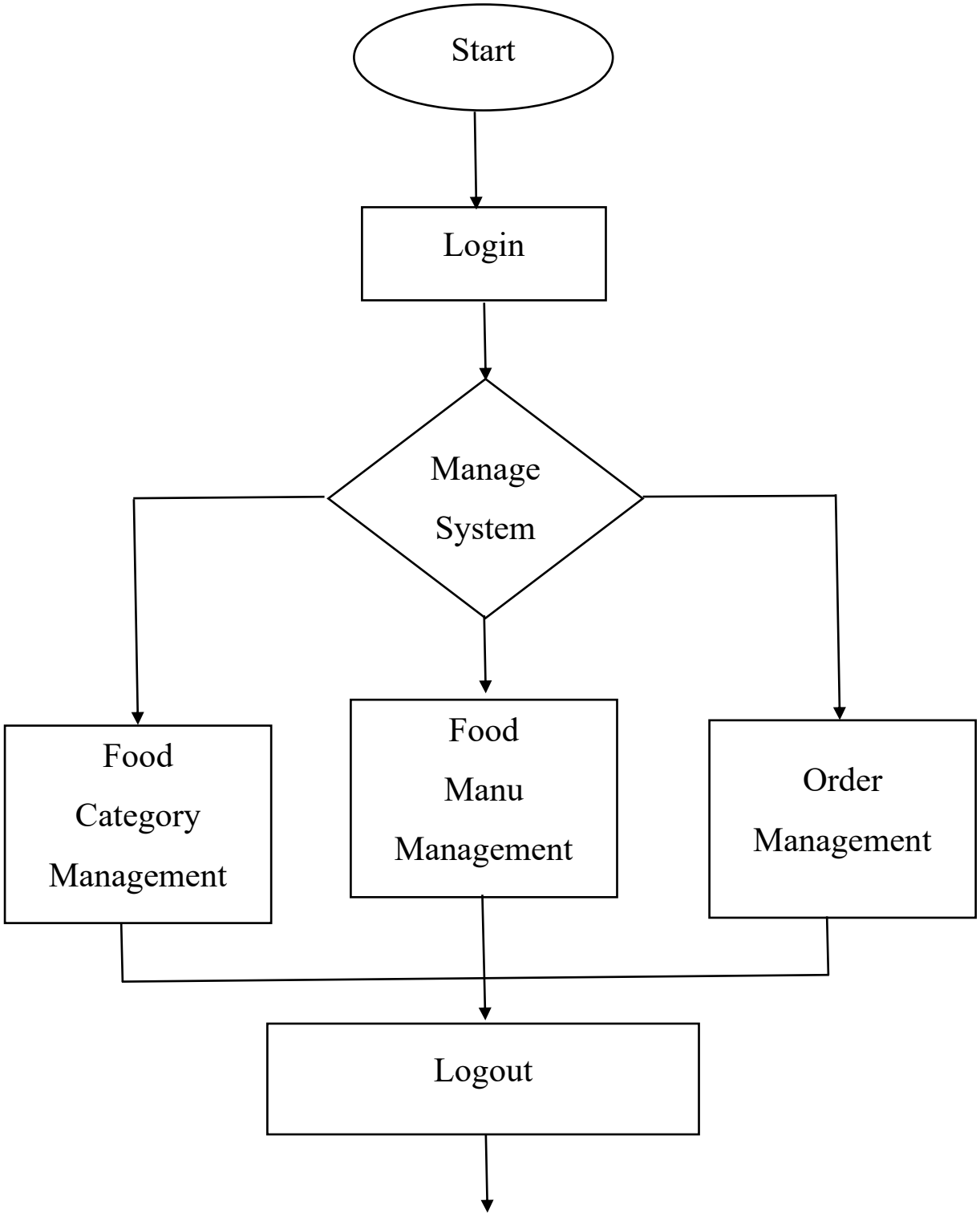


4.3 System Design Model

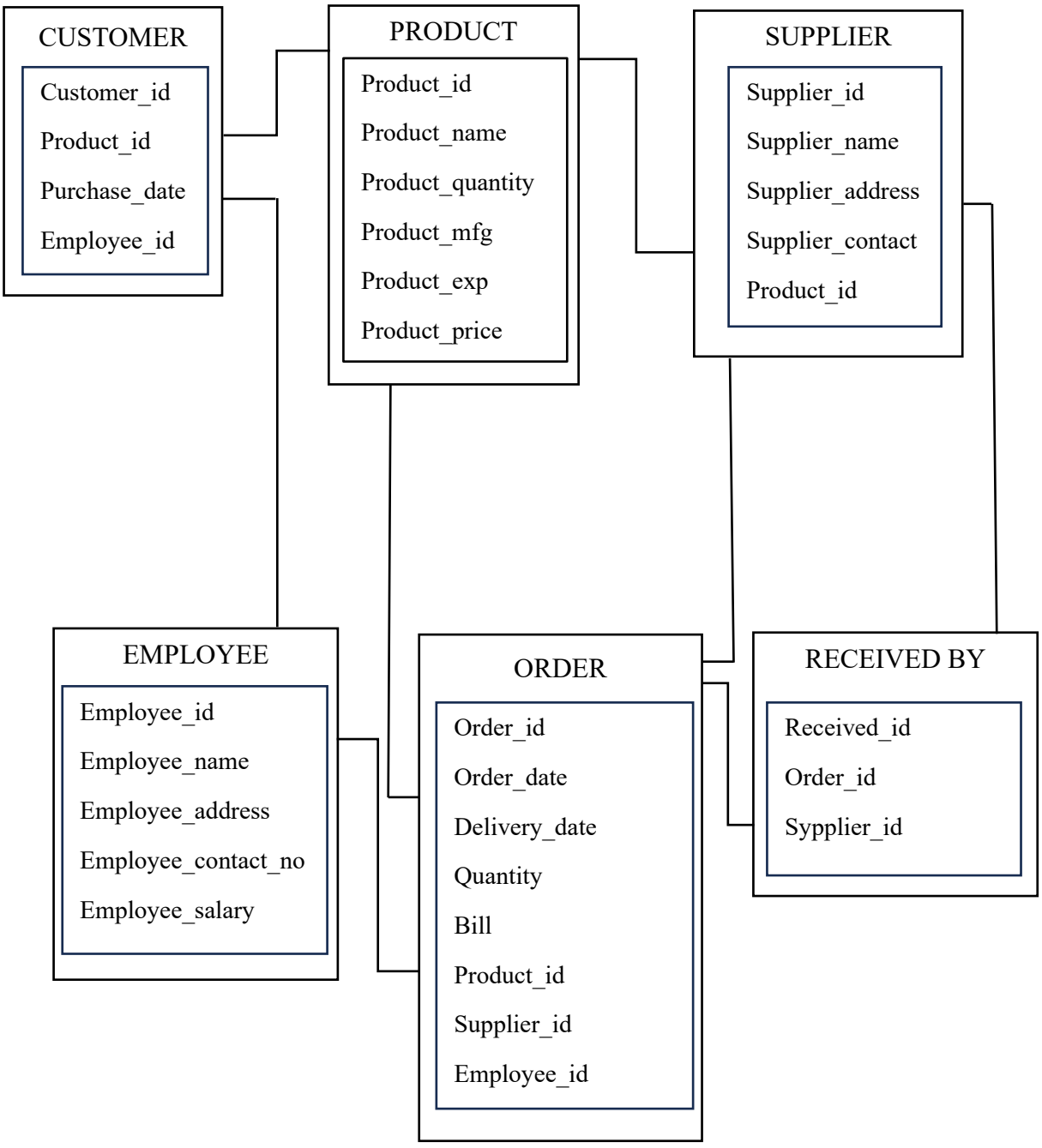


4.4 Admin Workflow Process

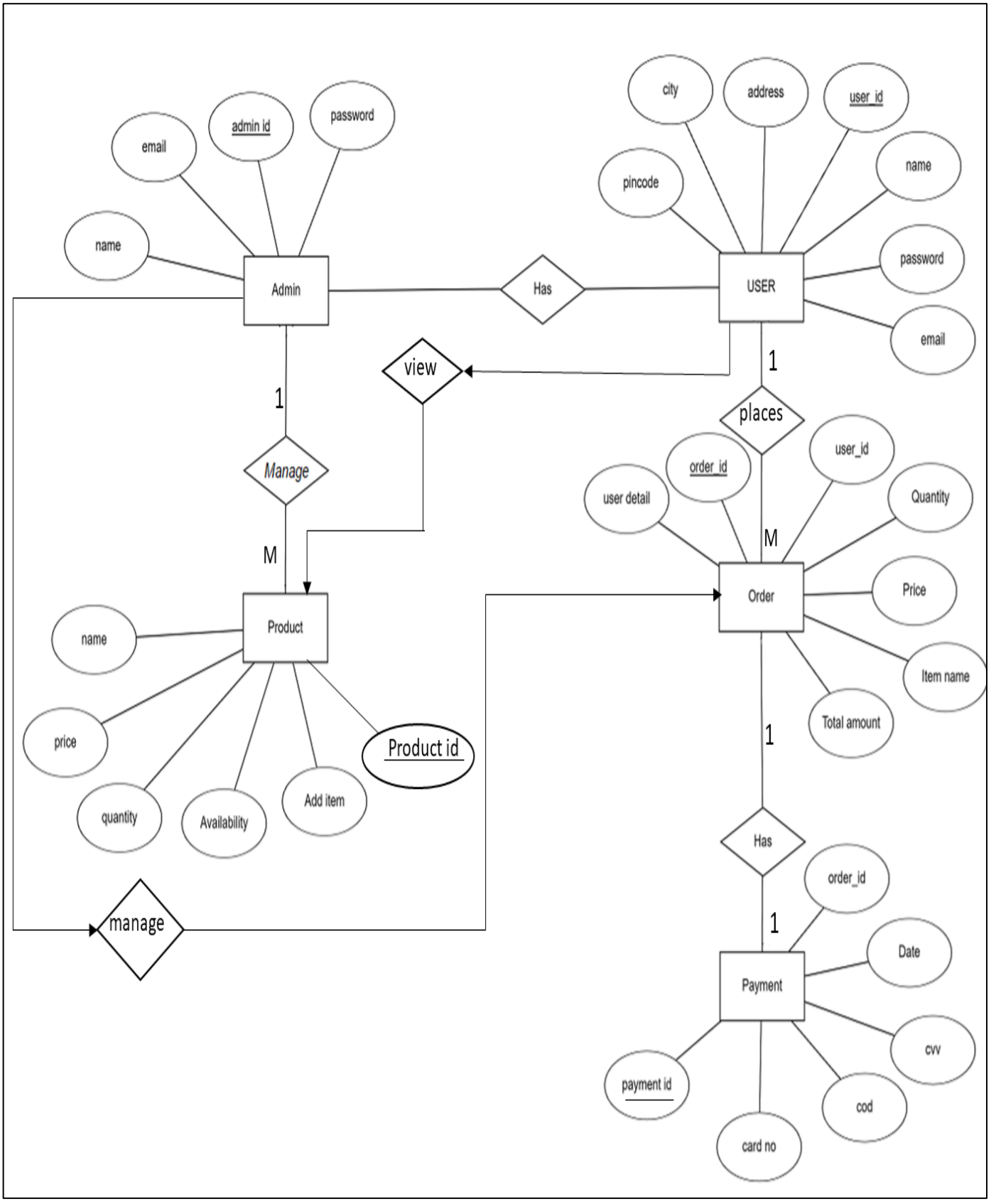
User goes to home page of the domain. If he/she has an account then he/she can login in cafe management system otherwise he/she need to register an account after successful registration, they can login in home page.



4.5 Conceptual Data Model



4.6 ER Diagram



4.7 Algorithm and Flow Charts of Modules

There are two points under this topic.

4.7.1 Algorithm

Step 1: Start i.e., Open the Website.

Step 2: Login.

Step 3: Already account.

Enter Password,

Enter Username,

Login Successfully.

Not have any account then go to step 4.

Step 4: Sign Up.

Create your account by signing into the website.

Sign Up Successfully.

Step 5: Take a look at all products available on the website.

Step 6: Add item as per user requirements in cart and user purchase click on order now button.

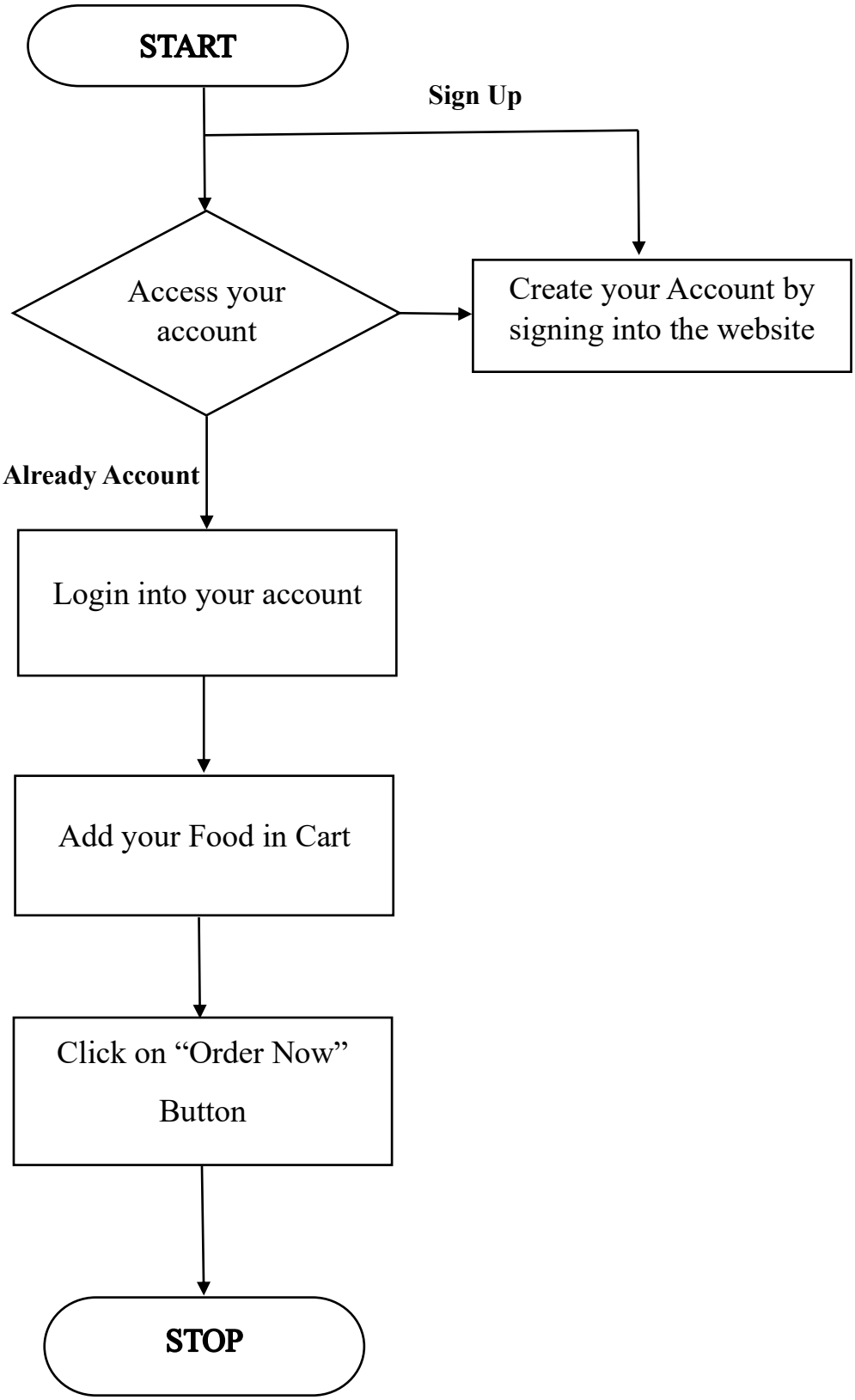
Step 7: Add Customer address.

Step 8: Click on Payment option mode.

Step 9: Show “Confirm Order Successfully” message on display.

Step 10: Stop.

4.7.2 Flow Chart



4.8 Tables

These are the following tables.

4.8.1 Signup Table

Base_App_signup		CREATE TABLE "Base_App_signup"
id	integer	"id" integer NOT NULL
fname	varchar(20)	"fname" varchar(20) NOT NULL
address	varchar(30)	"address" varchar(30) NOT NULL
email	varchar(30)	"email" varchar(30) NOT NULL
mobile	varchar(20)	"mobile" varchar(20) NOT NULL
pass1	varchar(20)	"pass1" varchar(20) NOT NULL
repass	varchar(20)	"repass" varchar(20) NOT NULL
gen	varchar(20)	"gen" varchar(20) NOT NULL

4.8.2 Table Booking Table

Base_App_book_table		CREATE TABLE "Base_App_book_table"
id	integer	"id" integer NOT NULL
Name	varchar(15)	"Name" varchar(15) NOT NULL
Phone_number	integer	"Phone_number" integer NOT NULL
Total_person	integer	"Total_person" integer NOT NULL
Booking_date	date	"Booking_date" date NOT NULL
Email	varchar(254)	"Email" varchar(254) NOT NULL

4.8.3 Feedback Table

Base_App_feedback		CREATE TABLE "Base_App_feedback"
id	integer	"id" integer NOT NULL
User_Name	varchar(15)	"User_Name" varchar(15) NOT NULL
Description	text	"Description" text NOT NULL
Rating	integer	"Rating" integer NOT NULL
Image	varchar(2083)	"Image" varchar(2083) NOT NULL

4.8.4 Cart Table

Base_App_cart		CREATE TABLE "Base_App_cart"
id	integer	"id" integer NOT NULL
created_at	datetime	"created_at" datetime NOT NULL
user_id	integer	"user_id" integer NOT NULL UNI

4.8.5 Cart Item Table

Base_App_cartitem		CREATE TABLE "Base_App_cartitem"
id	integer	"id" integer NOT NULL
quantity	integer unsign...	"quantity" integer unsigned NOT NUL
cart_id	bigint	"cart_id" bigint NOT NULL
item_id	bigint	"item_id" bigint NOT NULL

4.8.6 Item List Table

Base_App_itemlist		CREATE TABLE "Base_App_itemlist"
id	integer	"id" integer NOT NULL
Category_name	varchar(15)	"Category_name" varchar(15) NOT N

4.8.7 Items table

Base_App_items		CREATE TABLE "Base_App_items"
id	integer	"id" integer NOT NULL
Description	text	"Description" text NOT NULL
Price	integer	"Price" integer NOT NULL
Image	varchar(2083)	"Image" varchar(2083) NOT NULL
Category_id	bigint	"Category_id" bigint NOT NULL
Item_name	varchar(50)	"Item_name" varchar(50) NOT NU

4.8.8 Order Table

Base_App_order		CREATE TABLE "Base_App_order"
id	integer	"id" integer NOT NULL
customer_name	varchar(255)	"customer_name" varchar(255) NC
customer_address	text	"customer_address" text NOT NUL
payment_method	varchar(50)	"payment_method" varchar(50) NC
created_at	datetime	"created_at" datetime NOT NULL
item	varchar(255)	"item" varchar(255)
quantity	integer unsign...	"quantity" integer unsigned CHECK
customer_phone	varchar(20)	"customer_phone" varchar(20) NO

4.8.9 About Us Table

Base_App_aboutus		CREATE TABLE "Base_App_aboutus"
id	integer	"id" integer NOT NULL
Description	text	"Description" text NOT NULL

CHAPTER-05

TESTING

5.1 System Implementation Plane

A software design pattern called Model View Controller, or MVC as it is more formally known, is used to build online applications. There are three components to the Model View Controller pattern:

- **Model** – The lowest level of the pattern, is in charge of maintaining the data.
- **View** – This is in charge of showing the user all or part of the data.
- **Controller** – The computer program that controls how the Model and View interact.

MVC is well-liked because it provides for duty separation by separating the application logic and user interface layers. The Controller accepts all requests from the application and collaborates with the Model to prepare any necessary data for the View. The View then constructs a final presentable response using the data produced by the Controller. The following is a graphic representation of the MVC abstraction. Model of MVC (Model View Controller Flow)

5.2 Project Planning

The project was executed in the following phases:

Phase 1: Requirement analysis and feature listing

Phase 2: Database design and model creation

Phase 3: Frontend and backend integration

Phase 4: Admin panel implementation

Phase 5: Testing and debugging

Phase 6: Documentation and final report preparation

5.3 Features of System

- Fully functional **online food ordering system**
- Interactive **table booking module** with confirmation
- Secure **login/signup** for users and admins
- Clean, categorized **menu display** with search functionality
- Dynamic **cart management system**
- Admin backend for item, order, and reservation management
- **Feedback section** to gather user opinions
- **Mobile responsive design** for better accessibility

5.4 Facing Problem During Development the Project

During the construction of the web application "Online Food Order," the developer ran into a few issues. Here are a few issues in brief:

During development, several challenges were encountered:

I. **Session management**: Handling cart data for guest and logged-in users required session-based tracking logic.

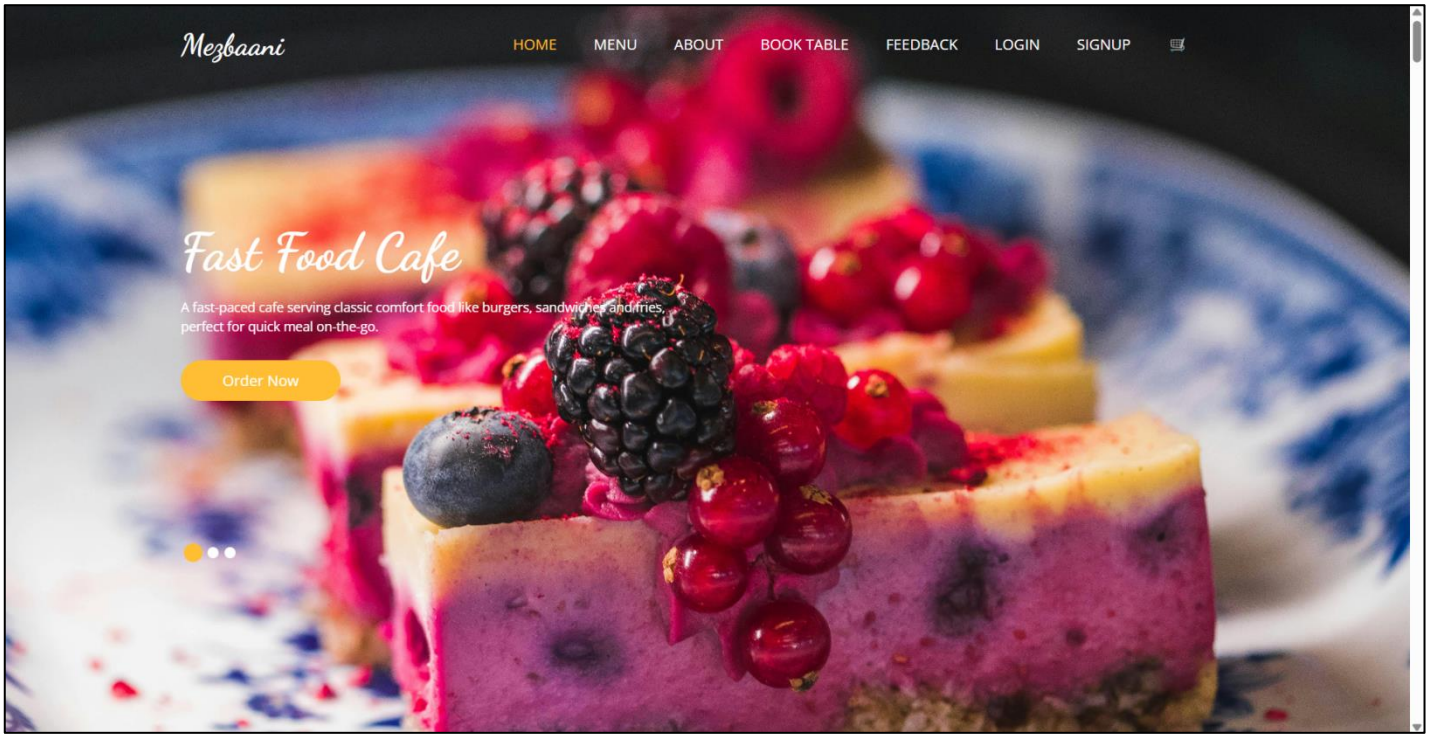
II. **Form validation**: Django's form handling required custom validators for phone numbers and reservation times.

III. **Image handling**: Managing menu item images required proper media root configuration and testing on various browsers.

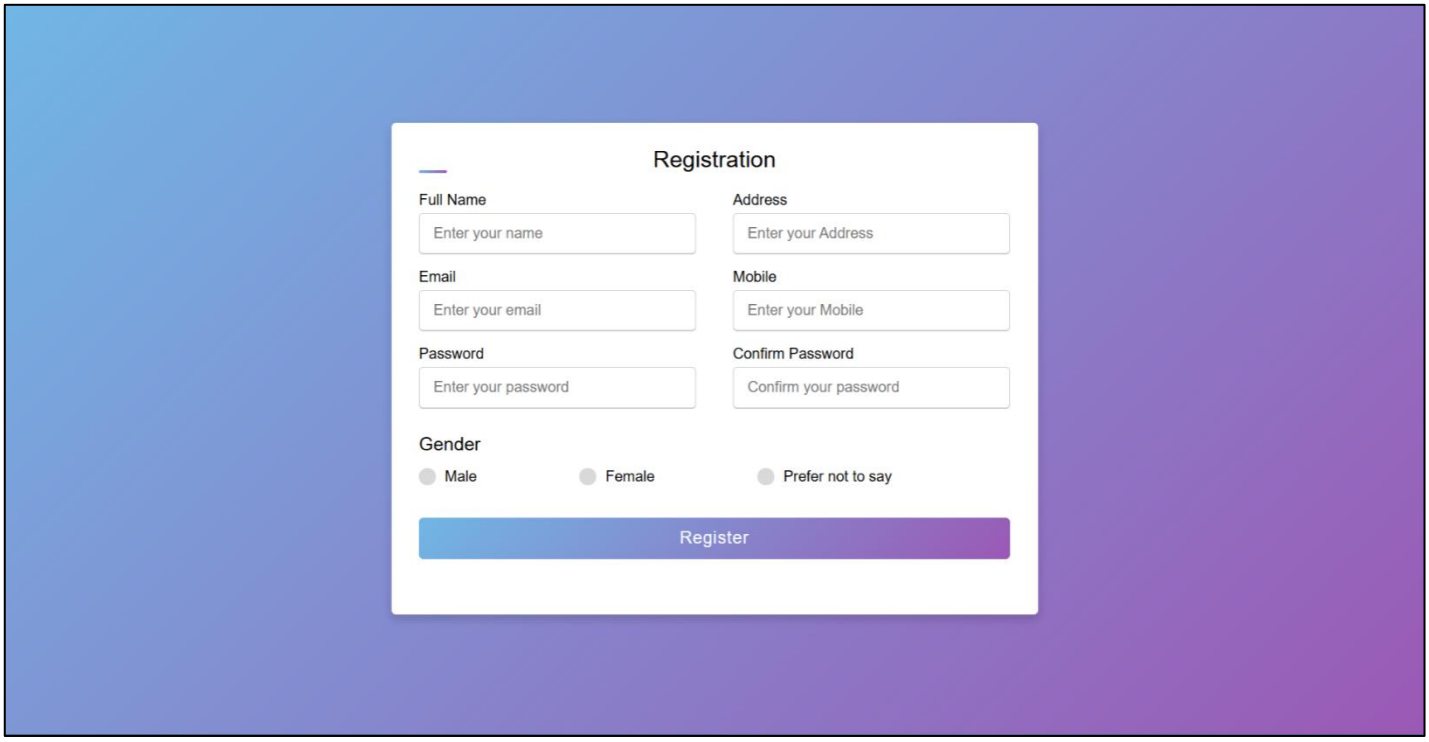
IV. **Admin panel access control**: Ensuring that only superusers could access the admin dashboard was implemented using Django's.

5.5 Output Screen of Project

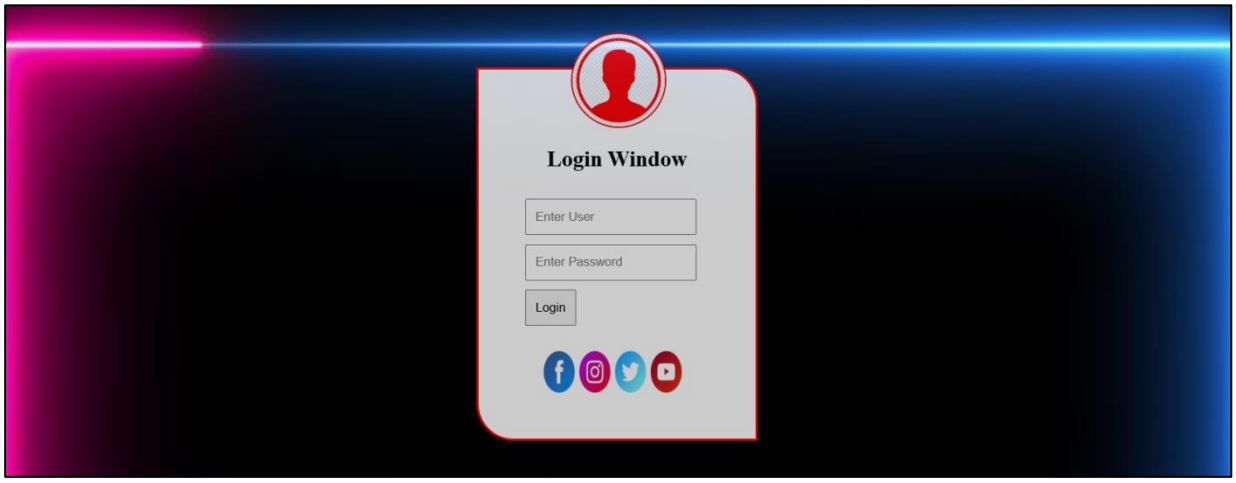
Home Page



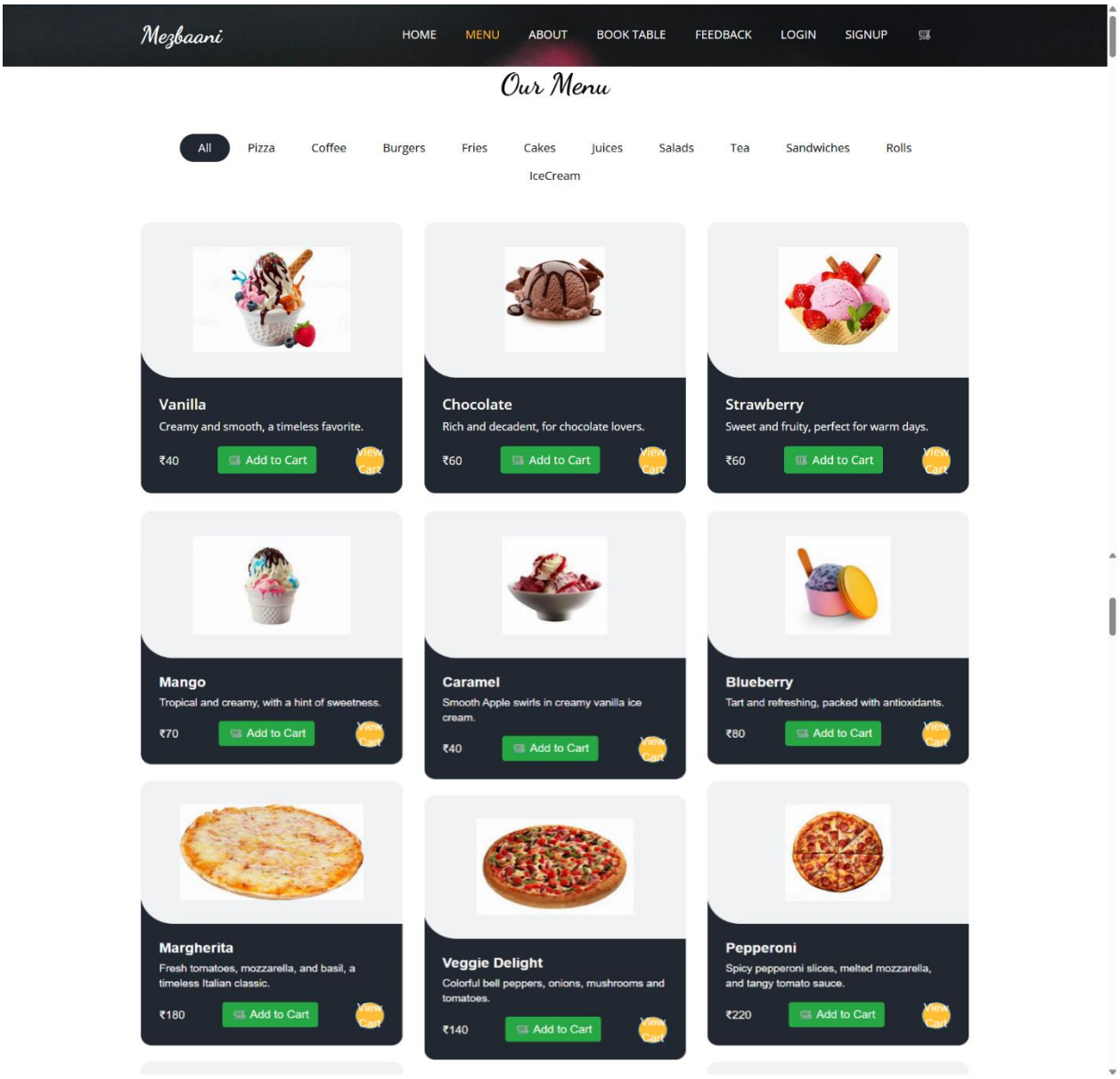
Registration Page



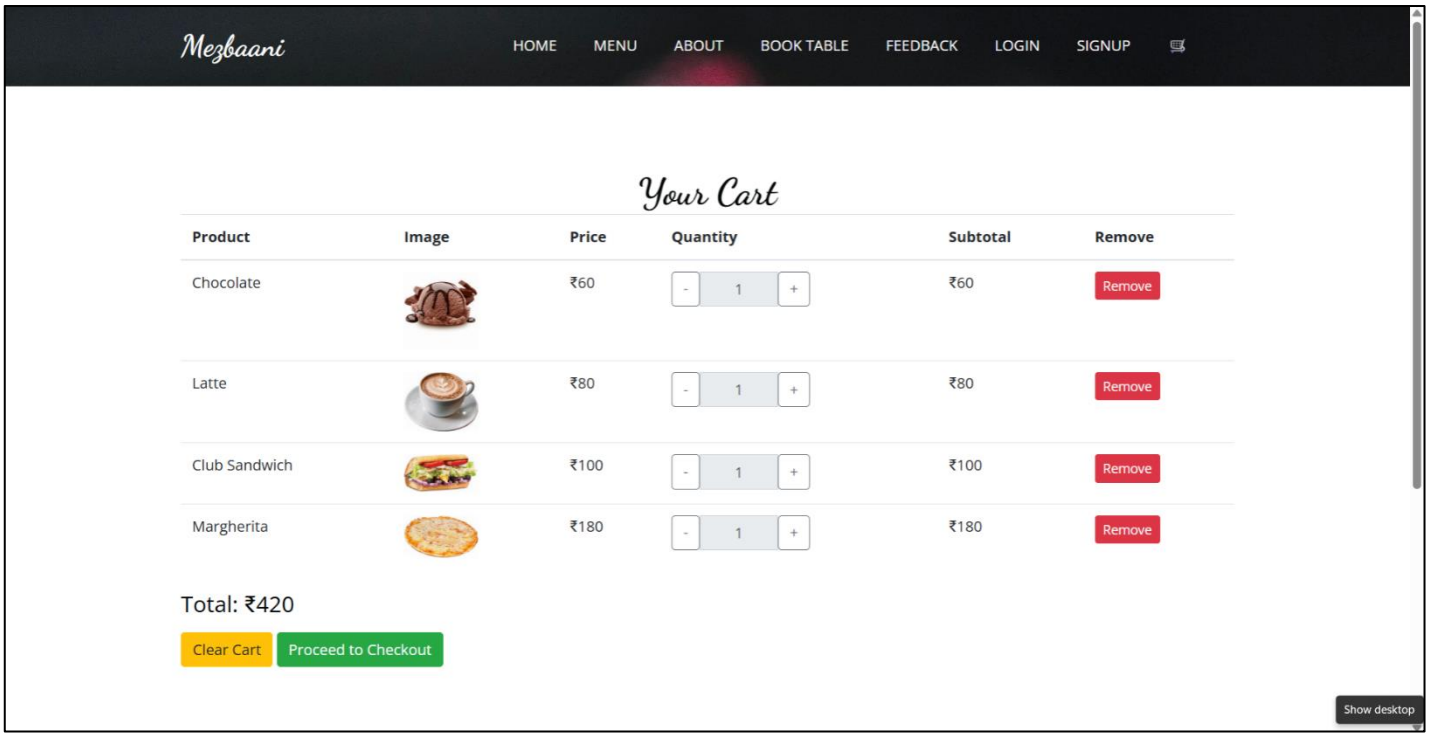
Login Page



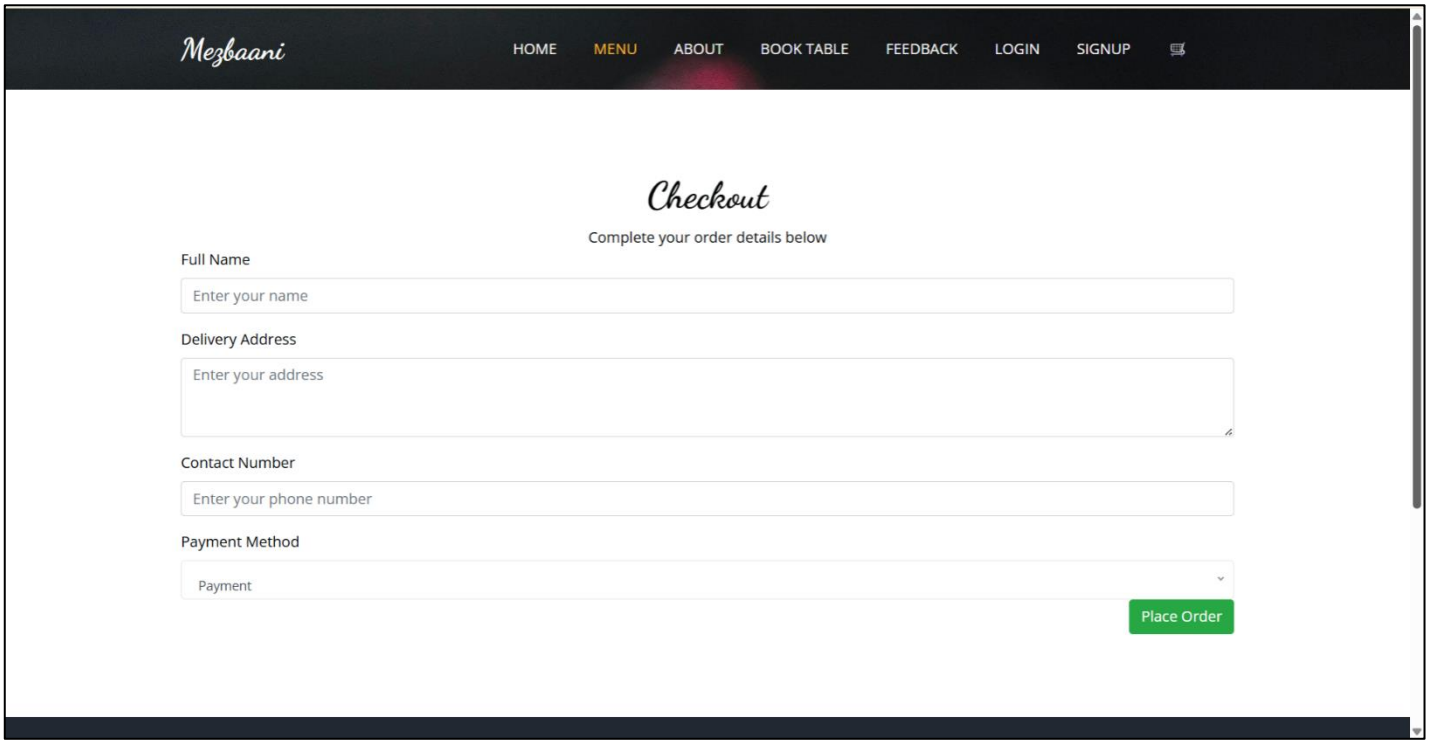
Menu Page



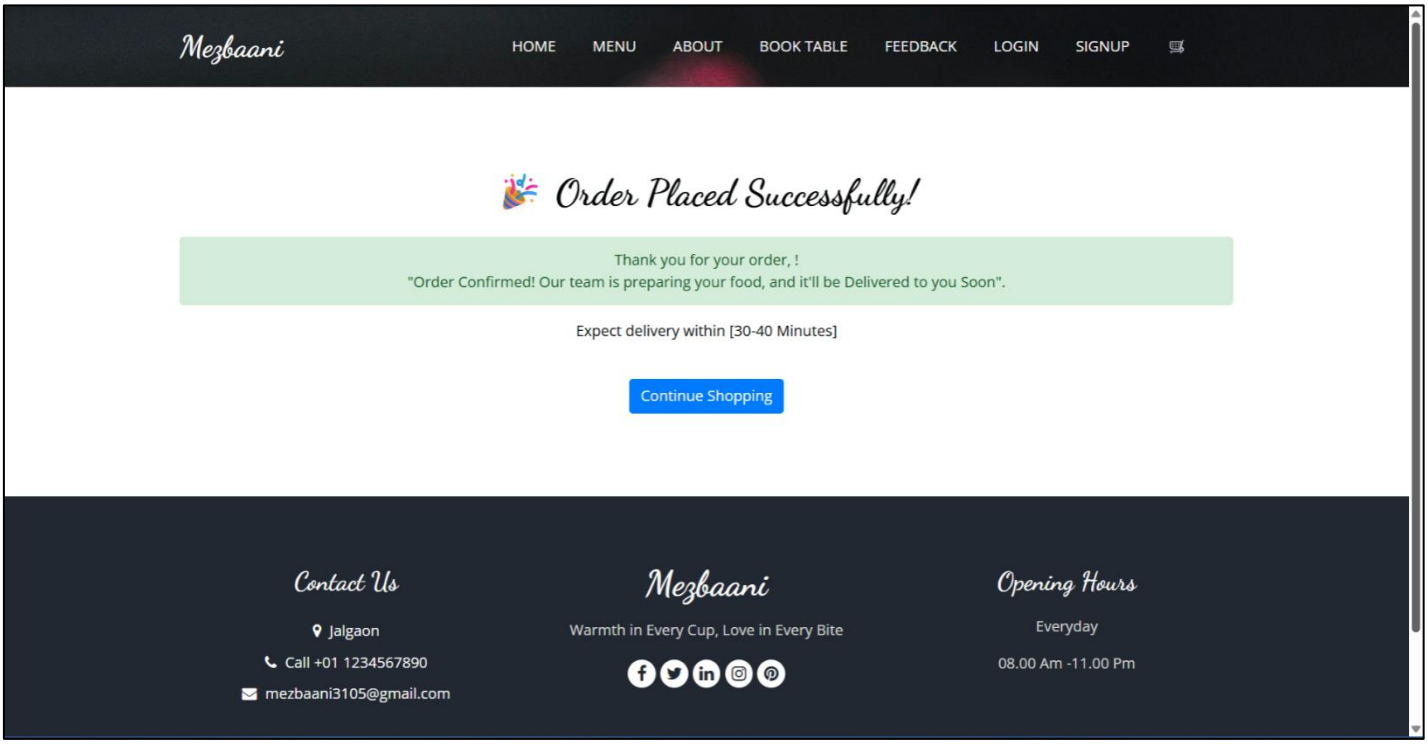
Cart Page



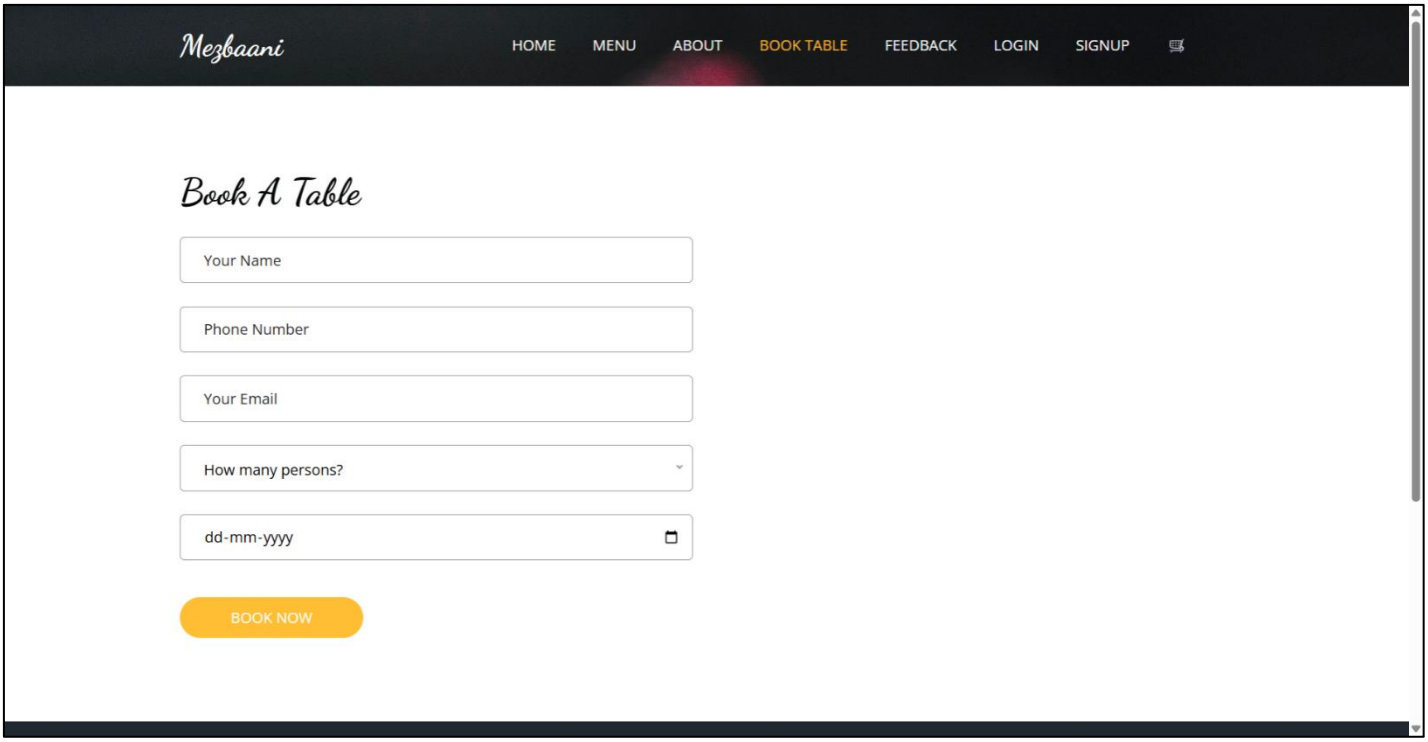
Checkout Page



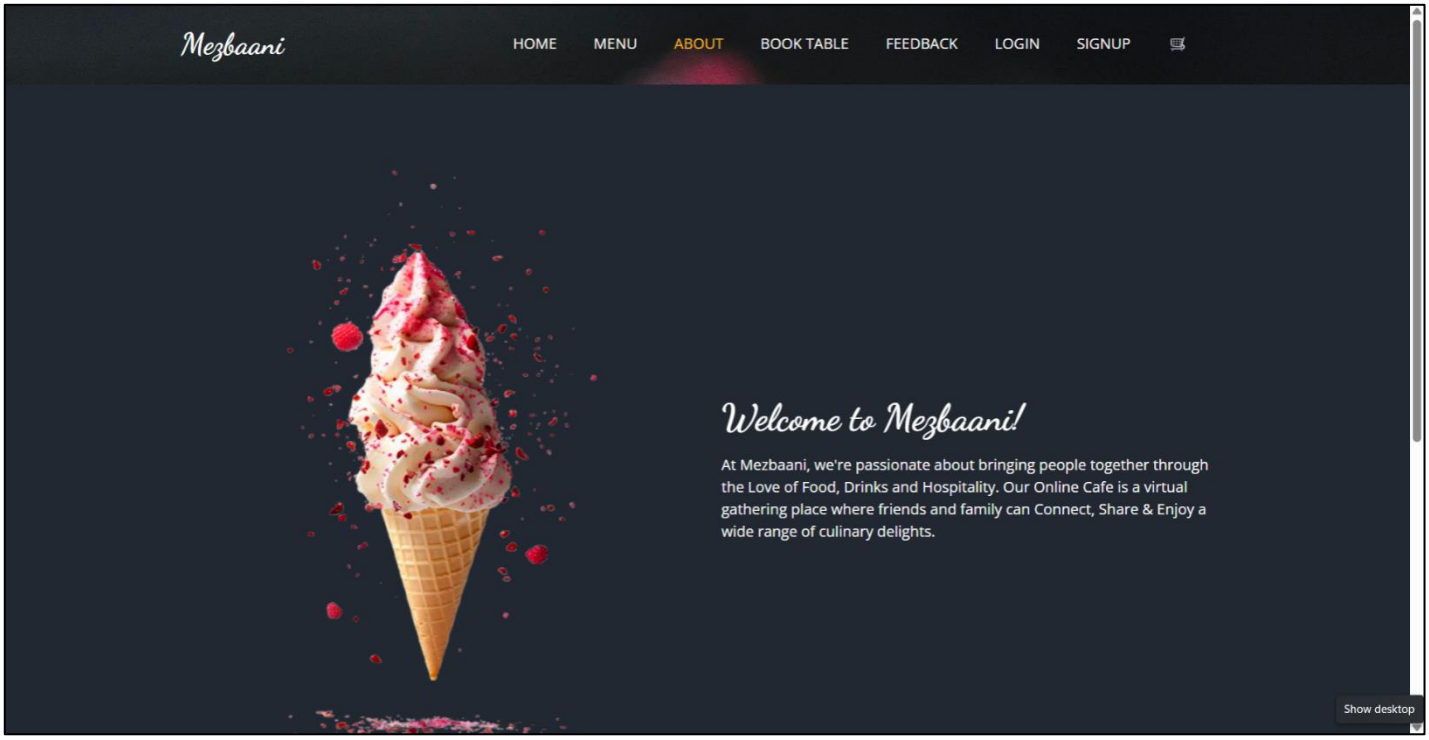
Order Confirmed Page



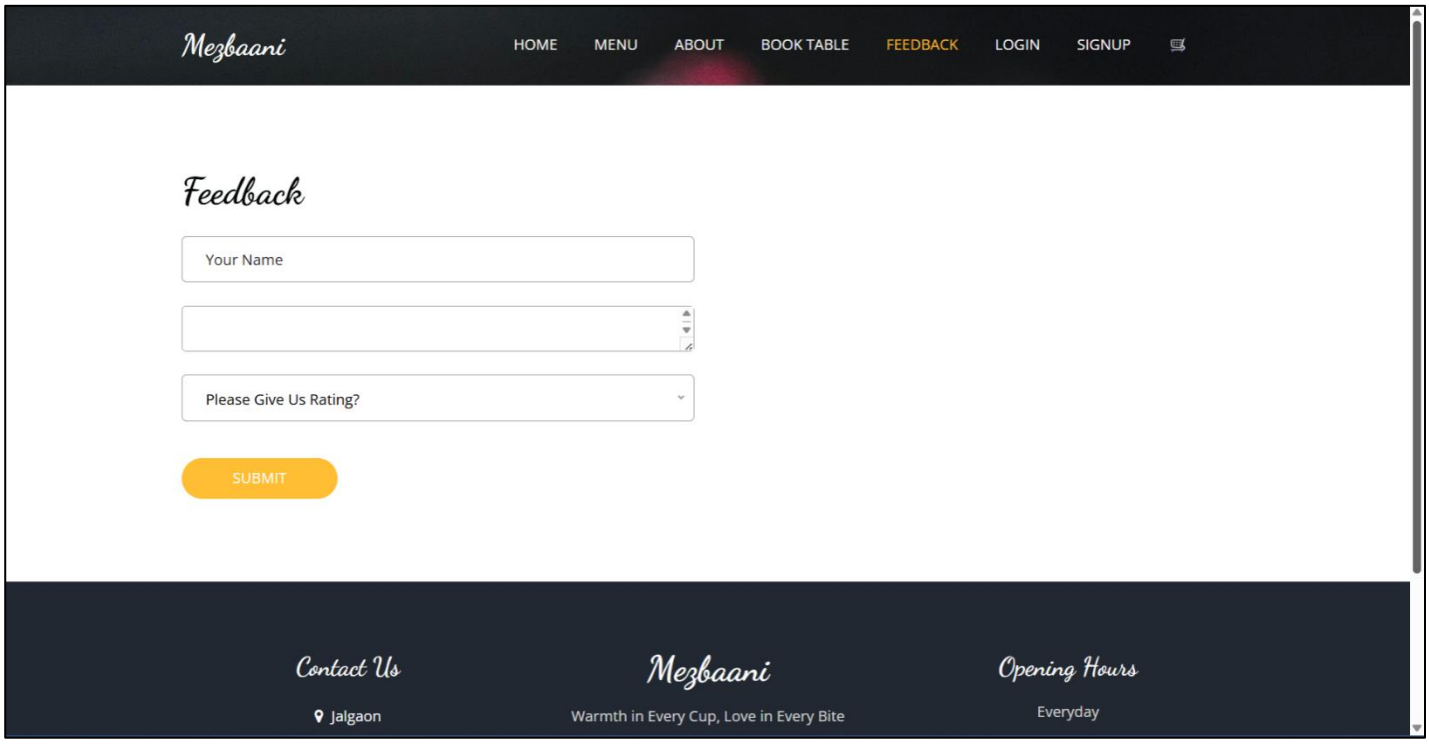
Booking Table Page



About Page



Feedback Page



CHAPTER-06

CONCLUDING REMARKS

6.1 Result

The final output is a complete **Django-based** Web based Cafe Management System, which can be used in any kind of cafe. This Cafe Management System can help to manage the Cafe more effectively, efficiently and smoothly. This is more secured and there will be speedy and well-ordered authentication procedure for the maintenance of records. At present time, in this Technology based world, people Likes and wants everything to be smooth and efficient through the use of data and information.

The Mezbaani project successfully met its defined objectives:

- Users were able to register, log in, and place orders without errors.
- Admins could manage all backend functions smoothly.
- Table bookings were recorded and updated in real-time.
- The interface remained user-friendly and mobile-compatible.
- System performance was reliable with minimal load time in testing scenarios.

6.2 Limitations of the System

The system has certain other restrictions as well. There are only a few basic functions in the system's shopping cart, and it can't be extensively customized. Additionally, practically all of the functionality of the application, including validation, is handled by server-side programming.

- The system currently supports **only one branch** of a cafe; multi-branch scalability is not implemented.
- **Email/SMS confirmation** for orders or bookings is not integrated.
- No support for **real-time order tracking** (e.g., delivery stages).
- No advanced **analytics dashboard** for business insights.
- System security is limited to Django's default protections without custom encryptions or multi-factor authentication.

6.3 Conclusions

The Mezbaani project has effectively demonstrated the feasibility and benefits of integrating digital technology into the traditional cafe environment. Through features such as online food ordering, table booking, and admin management, the platform enhances customer convenience and streamlines backend operations.

The Django framework provided a secure and scalable foundation for development, while Bootstrap ensured a responsive and accessible frontend. During testing, the system showed reliable performance, user-friendly navigation, and smooth integration between different modules.

Although there are areas for future enhancement, the current system full fills the core objectives: reducing wait times, minimizing manual errors, and improving service efficiency. Overall, Mezbaani stands as a robust prototype for cafes looking to modernize their operations through web-based solutions.

CHAPTER-07

SCOPE & FURTHER WORK

7.1 Scope

To enhance the Mezbaani system and prepare it for real-world deployment, the following improvements can be considered:

- **Multi-branch Support**: Extend the platform to support multiple cafe branches with centralized admin control.
- **Real-time Order Tracking**: Implement order status updates for delivery or in-store preparation.
- **Payment Gateway Integration**: Add secure online payment options such as Razorpay, Stripe, or PayPal.
- **Analytics Dashboard**: Develop advanced reports for sales trends, popular items, and user engagement.
- **Mobile Application**: Build a native mobile app for iOS and Android for greater accessibility.
- **Enhanced Security**: Implement two-factor authentication and data encryption for sensitive transactions.
- **Inventory Management**: Include inventory tracking for better stock control and automatic low-stock alerts.

7.2 Future Work

Each project should pay close attention to future development because it contains the system's most recent features. It lessens software issues and defects. It develops a close relationship with customers based on their comments or preferences. Developer will incorporate certain dynamic elements that are briefly described below into my restaurant management system. Reporting module with real time mechanism.

- Modern architecture with smooth transitions.
- System for email and mobile conformation.
- Selling Point.

CHAPTER-08

REFERENCE

References

[1] Django Documentation. https://docs.djangoproject.com

[2] Bootstrap Official Website. https://getbootstrap.com

[3] W3Schools – HTML, CSS, and JavaScript Tutorials. https://www.w3schools.com

[4] Stack Overflow – Developer Community. https://stackoverflow.com

[5] SQLite Documentation. https://www.sqlite.org/docs.html

[6] GitHub –Open-Source Django Projects. https://github.com

[7] Real Python – Django Web Framework Tutorials. https://realpython.com/tutorials/django/

[8] GeeksforGeeks – Python and Django Articles. https://www.geeksforgeeks.org

[9] ThemeWagon – UI Templates. https://themewagon.com

[10] Google Fonts & Icons – UI/UX Enhancements.