Food Ordering Website using MERN STACK with Recommendation System

Sourav Kumar Gupta (20013440087)
Pursottam (20013440060)
Nityanand Singh (20013440053)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
R.V.S. COLLEGE OF ENGINEERING AND TECHNOLOGY
JHARKHAND UNIVERSITY OF TECHNOLOGY

2020-2024

A
Project Report
On

FOOD ORDERING WEBSITE USING MERN STACK WITH RECOMMENDATION SYSTEM

Submitted in partial fulfillment of the requirements
for
8th Semester

Bachelor of Technology in
Computer Science and Engineering

by

1. Name: - Sourav Kumar Gupta	**University Roll No - (20013440087)**
2. Name: - Pursottam	**University Roll No - (20013440060)**
3. Name: - Nityanand Singh	**University Roll No - (20013440053)**

Under the guidance of
Prof. Sangeeta
Professor, Dept. of Computer Science and Engineering

Department Of Computer Science and Engineering
R.V.S. College of Engineering and Technology
Jamshedpur – 831012
2020 – 2024

Department of Computer Science and Engineering

CERTIFICATE

*This is to certify that the project work entitled "***Food Ordering Website using MERN STACK with Recommendation System***" is done by Sourav Kumar Gupta,* **Regd. No:- (20013440087)** , *Pursottam***, Regd. No:- (20013440060)** , *Nityanand Singh***, Regd. No:- (20013440053)** *in partial fulfilment of the requirements for the 8th Semester Sessional Examination of* **Bachelor of Technology** *in* **Computer Science & Engineering** *during the academic year 2020-24. This work is submitted to the department as a part of evaluation of 8th Semester Project.*

Prof. Sangeeta
Professor, Department of CSE
Project Guide

Prof. Jeevan Kumar
H.O.D, CSE

Signature of External

Date: /05/2024
Place: RVSCET, Jamshedpur

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1. Sourav Kumar Gupta
2. Pursottam
3. Nityanand Singh

ACKNOWLEDGEMENT

We avail this opportunity to express our profound sense of gratitude to **Prof. Sangeeta** for her constant supervision, guidance and encouragement right from the beginning till the completion of the project.

We wish to thank all the faculty members and supporting staffs of Dept. of Computer Science and Engineering for their valuable suggestions, timely advice and encouragement.

1. Sourav Kumar Gupta

2. Pursottam

3. Nityanand Singh

Abstract

The Food Ordering Website is proposed here which works on the food requesting process. The proposed framework shows a UI and updates the menu with all accessible choices so if facilitates the client's work. The client can pick more than one thing to make a request and can see request subtleties prior to logging off. The request affirmation is shipped off to the client. The request is put in the line and refreshed in the data set and returned continuously. This framework helps the staff to go through the orders continuously and process them proficiently with negligible mistakes.

The justification for Food Ordering Website is to motorize the manual structure with the help of electronic instruments and absolutely PC programming, accomplish their necessities, so their fundamental data/information can be taken care of for a more expanded time period by absolutely overseeing something almost identical. The normal programming and gear are really open and easy to work with.

Food Ordering Website, as portrayed above, is botch-free, secure, strong, and speedy organization structure. It can assist the client with zeroing in on their various activities rather to focus on the record-keeping. Thusly, it will help the relationship in better utilization of resources. The affiliation can stay aware of electronic records without silly segments. That infers that one need not be involved by information that isn't significant while having the choice to show up at the information. In a general sense, the endeavor depicts how to supervise great execution and better organization for clients.

Content

1. Introduction……………………………Pg. No. 01-06

1.1 Project Overview....................................Pg. No. 02
1.2 Project Scope..Pg. No. 04
1.3 Product Features.......................................Pg. No. 05-06

2. System Analysis………………………Pg. No. 07-10

2.1 Software Requirements ....Pg. No. 07-10
2.2 Hardware Requirements.........................Pg. No. 10
2.3 User (Questionnaires) ....Pg. No. 10

3. System Design & Specifications…….Pg. No. 11-21

High Level Design (HLD)

3.1 Project Model.. Pg. No. 11
3.2 Structure Chart...Pg. No. 11-12
3.3 DFD .... Pg. No. 13-16
3.4 E-R Diagram.. Pg. No. 17
3.5 UML (Use Case Diagram)........................Pg. No. 18

Low Level Design (LLD)

3.6 Screen-Shot Diagram ....Pg. No. 19-21

4. Coding…………………………………..Pg. No. 22-23

4.1 Sample Code...Pg. No. 22-23

5. Testing…………………………………..Pg. No. 24-25

5.1 Unit Testing.. Pg. No.24
5.2 Integration Testing .... Pg. No.24-25

6. Conclusion & Limitations…………..Pg. No. 26-27

7. Reference/Bibliography…………….Pg. No. 28

List of Figures

No. of Figure	Name of Figure	Page No. of Figure
Fig.3.1.2	Structure Chart	Page No.12
Fig.3.1.3.1	Context level 0 DFD	Page No.13
Fig.3.1.3.2	Context level 1 DFD	Page No.14
Fig.3.1.3.3	Context level 1 DFD	Page No.14
Fig.3.1.3.4	Context level 2 DFD	Page No.15
Fig.3.1.3.5	Context level 2 DFD	Page No.15
Fig.3.1.3.6	Context level 2 DFD	Page No.16
Fig.3.1.4	E R Diagram	Page No.17
Fig.3.1.5	UCD	Page No.18
Fig.3.2.1	Home Page	Page No.19
Fig.3.2.2	Login Page	Page No.20
Fig.3.2.3	Admin Dashboard	Page No.20
Fig.3.2.4	Food Cart	Page No.21
Fig.3.2.5	Cart Page	Page No.21

List of Table

No. of Table	Name of Table	Page No.
Table 1	Scope	Page No.05

Chapter 1

Introduction

The era we live in today, is the digital age. Day by day it changes the lifestyle of people and is very helpful for every part of life. You can even control your home appliances through digitally. In order to avoid the rush, which is a major issue in restaurants become ordering food takes a lot of time while orders are being prepared, we create an online website for a restaurant through which a customer can easily access all the thing belonging to a restaurant while sitting anywhere rather than going to the restaurant.

One of the newest features that the majority of fastfood businesses in the West are adopting is the online meal ordering system. This process involves ordering food online and having it delivered to the customer. The adoption of an electronic payment system makes it possible. Although credit card clients can be served befere they make a payment via cash or check, they typically pay with their credit cards. Customers will therefore be able to buy food online thanks to the technology created for this project.

Anyone can place an online order for any product and have it delivered right to their door. However, the focus is on the payment mechanism rather than the method of transmission of the products and services. In other words, how feasible is online payment for goods and services? The discussion of the economic effects of digital currency is then prompted by this. What are the implementations from an economic perspective? Since the globe is quickly becoming into a global village, communication of which telecommunication is a crucial component is a critical instrument for this process.

The electronic meal ordering system is maybe the newest aid that the majority of drive- through restaurants in the west are adopting. With this approach, the client gets informed about food on the internet. By using an electrical part structure, it is made practicable. Despite the fact that Visa customers can be served even before they make a payment in the form of cash or a check, customers pay using their Master cards. As a result, the design used for this project will encourage customers to search online for food demand.

Here we propose a "Food Ordering Website" that has been intended for Fast Food eatery, Take-Out or College Cafeterias. The framework can likewise be utilized in any food conveyance industry. This works on the course of food requests for both the client and the cafe, as the whole course of taking requests is mechanized.

1.1 Project Overview

The task sets to foster an internet requesting framework for eatery. Numerous enterprises are currently rapidly taking on advances. Café industry additionally embraces various kinds of advances which make day to day processes simpler and quicker. In any case, the reception of innovation in eatery industry is more slow contrasted with different businesses.

Online food requesting frameworks are turning into a well-known point. That is on the grounds that they are serving the consistently expanding interest for persuading. The principal motivation behind a web based requesting framework is to give clients to a method for putting in a request at a café over the web. With a site or portable application, clients can undoubtedly peruse every one of the dishes the eatery has accessible, modify dishes to their prerequisites and submit a request. It can likewise save their number one orders permitting them to effectively re-request that later on.

1.2 Project Aim & Objective

A web-based restaurant ordering system that automates orders is what the project aims to create. Manage online orders and check status will also be helpful to management. The system allows the management to add menus and take orders. The system also has a straightforward user interface that is mobile-friendly and usable on a variety of screens and devices. The application will integrate the Face book API so users can log in using their Face book accounts and like or share menus, which can help attract new users through word-of- mouth.

The following goals must be met in order to accomplish the aforementioned goal:

- Conduct a thorough literature review to learn about previous efforts to automate restaurant food ordering processes.
- The whole system will be designed and developed in a systematic manner using the appropriate methodologies.
- The backend and frontend of the system will be built using MERN Stack and Bootsrap.
- The system will in corporate social media. Customers can therefore login and register and then comment on the menu.

To ensure system quality, the system will undergo thorough testing.

1.3 Project Rationale

Web-based requests from eateries are getting more straightforward as versatile and web innovation advances. Food sweethearts presently will in general request food varieties through a web-based requesting framework as it is simpler and quicker. QSR Web found that computerized café requests are growing 300% quicker than eating in rush hour gridlock. In any case, most of the eatery don't give a fixation on showing up on the web. Subsequently, 51% of all versatile ventures on Google are for eateries, yet not many as 5% have portable improved sites.

1.4 Feasibility of Project

Except for a few minor flaws that are certain to surface during more thorough testing, the system is currently fully operational. Currently, a user can register, log in, and place an order on the website. The order retrieved desktop application accurately and totally displays that order. A complete web application for a restaurant that can handle online orders is needed for the project. This web application will be mobile-optimized so that it can be seen on both a PC and a mobile device. The system will be developed using open-sourced technologies including MERN Stack ,Bootstrap, and Apache Server. As a result, the technologies are free to utilize. All user requirements should be met by the suggested solution, and it should be adaptable enough to allow for future changes depending on new requirements.

- Economic Feasibility

  This is a crucial factor to take into account when creating a project. We chose the technology based on its lowest potential cost. The organization is required to pay the full cost of all hardware and software. Overall, we have calculated that the advantages the business will experience from the suggested system will undoubtedly outweigh the upfront fees and ongoing operating expenses for the system.

- Technical Feasibility

  This includes researching system performance, function, and limitations that would make it more difficult to develop a workable system. In order to conduct this feasibility analysis, we examined all of the system's intended features and determined whether or not it was feasible using several frontend and backend platforms.

- Operational Feasibility

  Without a sure, the suggested system is entirely GUI-based, very user-friendly, and all inputs to be taken are self-explanatory even to a layperson. Additionally, sufficient training has been provided to users so they are familiar with the basics of the system and feel at ease using it. According to our analysis, the system has reduced the clients' workloads and doing, so they are comfortable and content.

1. 5 Scope of Project

It might aid in meticulously gathering optimal management. The collection will be clear-cut and logical in a relatively short period of time. It will assist someone in fully understanding the management of the previous year. Additionally, it supports all ongoing initiatives involving the online food ordering system. The management & collection process will be efficient, and the cost of collection will be decreased. Our project aims at business process automation, so we have made an effort to computerize several Online Food Ordering System activities.

Generally speaking, an expanded degree of control has been displayed to prompt a more elevated level of consumer loyalty and a more noteworthy aim to utilize or suggest recommended assistance. Seen comfort of a self-administration framework likewise prompts an expansion in reception and fulfillment. On this occasion, the meaning of accommodation is connected principally to getting to comfort and exchanging accommodation. A client will look for a most loved eatery base on the client area, and browse accessible things. Installment can be among others either with MasterCard or cash.

It might help gather ideal administration exhaustively. In an exceptionally brief time frame, the assortment will be basic and reasonable. It will assist an individual with knowing the administration of a spent year flawlessly. It likewise helps in current all works comparative with Online Food Ordering System. It will be additionally decreased the expense of gathering the administration and the assortment system will continue without a hitch. Our venture focuses

on Business process robotization, for example, we have attempted to automate different cycles of the

- The computer system requires the user to fill out a variety of forms, and it is simple to produce multiple copies of each form at once.
- In a computer system, we can simply print the manifest instead of having to create one first, which saves us time.
- To help the personnel record the time spent on each of their assigned task areas.
- To increase resources' productivity through automation in order to use them more wisely.
- The system creates multiple information kinds that can be applied to a variety of situations.
- It satisfies the user's needs.
- Be simple for the user and operator to understand.
- Be simple to use.
- Have a user-friendly interface.

SCOPE IN	SCOPE OUT
Web application for restaurant.	IOT based food ordering system.
Online food ordering system.	Physical smart system for restaurant.
Online customer management.	Mobile application
Written report on project.	

Table 1 Scope

1.6 Purpose of Project

The goal of the online food ordering system is to automate the manual process so that data and information can be stored for a longer length of time using computerized technologies. The online ordering system is a convenient way for individuals to order food online without having to physically visit a restaurant. The internet is what makes this system possible; it is

what links the customer with the restaurant or food provider on the one hand. In essence, online ordering is a fast, dependable platform that is specifically designed for restaurants and handles everything from taking orders to updating customers on delivery. As consumer needs rise, this approach is currently incredibly profitable for food industry owners.

The Online Food Ordering System's primary goal is to manage the information for Item Category, Food, Delivery Address, Order, and Shopping Cart to control the ordering of fast food online. It enables clients to order desired products from any location and at any time. Additionally, pay upon delivery.

1.7 Proposed System

The suggested solution guarantees complete user autonomy by giving each user access to a personal system where they can log in and reserve the things they want.

The system must store fresh information. System must assist internal personnel in maintaining Category information and locating it in response to various questions.

The system must keep track of quantity.

The system must maintain the customer record and update and delete it as necessary.

A search area is also required by the system.

A security system is also required to protect data.

Only individuals who have registered with our system are able to book products and cancel them as needed. The entire work is done online under this method. Customers can use this site's feedback form to submit
questions and recommendations

1.8 Existing System

The tests are only performed physically in the existing framework; however, in the suggested framework, we must automate the tests by using this application

- The lack of information security.
- A greater labor pool.
- Requires a significant amount of paring labor.
- Manual estimations are required.
- The higher authorities don't have any immediate jobs

Chapter 2

System Analysis

System analysis involves understanding the software and hardware requirements necessary to develop and run the Food Ordering website effectively, as well as gathering user requirements through questionnaires.

2.1 Software Requirements

The software requirements for this project include:

- **Frontend:** React.js, Redux for state management, HTML5, CSS3, JavaScript, Axios for API calls.
- **Backend:** Node.js, Express.js, MongoDB, Mongoose for database interactions.
- **Payment Integration:** Stripe API for payment processing.
- **Development Tools:** Visual Studio Code, Git for version control, Postman for API testing.

- ### React

React is a popular JavaScript library developed by Facebook for building user interfaces, especially single-page applications (SPAs). It allows developers to create large web applications that can update and render efficiently in response to data changes.

Key Features

- Component-Based Architecture: Enables the building of encapsulated components that manage their own state and can be composed to create complex UIs.
- Virtual DOM: Improves performance by minimizing direct DOM manipulations and updating only the parts of the DOM that need to change.
- JSX Syntax: Allows developers to write HTML-like code within JavaScript, making the code more readable and easier to debug.

Benefits

- Reusable Components: Promotes code reuse, reducing development time and effort.
- Fast Rendering: Virtual DOM ensures high performance and quick updates.
- Strong Community Support: Extensive resources, libraries, and tools available for development.

- ## Redux

Redux is a predictable state container for JavaScript applications, commonly used with React for managing application state.

Key Features

- Single Source of Truth: The state of the entire application is stored in a single object.
- Immutability: State changes are made with pure functions called reducers, ensuring predictability and easier debugging.
- Middleware Support: Enhances functionality through middleware, such as logging, asynchronous requests, and more.

Benefits

- Predictable State Management: Simplifies the state management of complex applications.
- Debugging Capabilities: Tools like Redux DevTools help in tracking state changes and debugging.
- Scalability: Suitable for applications of all sizes due to its structured approach.

- ## Axios

Axios is a promise-based HTTP client for the browser and Node.js. It is used to make asynchronous HTTP requests to fetch or save data.

Key Features

- Promise-Based: Uses Promises to handle asynchronous operations, making the code cleaner and easier to manage.
- Request and Response Interceptors: Allows transforming requests and handling responses before they reach the application.
- Automatic Transformations: Automatically transforms JSON data.

Benefits

- Ease of Use: Simple API for making HTTP requests.
- Error Handling: Provides robust error handling capabilities.
- Wide Browser Support: Compatible with all major browsers and Node.js.

- ## Node.js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows the execution of JavaScript code server-side, making it possible to build scalable network applications.

Key Features

- Event-Driven Architecture: Non-blocking, asynchronous programming model.
- NPM (Node Package Manager): Vast ecosystem of libraries and packages available for use.

- High Performance: Efficiently handles concurrent connections with high throughput.

Benefits

- Scalability: Suitable for building scalable network applications.
- Unified Language: Enables full-stack JavaScript development (both frontend and backend).
- Community Support: Large and active community contributing to a rich ecosystem of modules.

Express.js

Express.js is a minimal and flexible Node.js web application framework that provides robust features for web and mobile applications.

Key Features

- Middleware: Uses middleware to handle HTTP requests and responses.
- Routing: Provides a robust set of features for handling different routes in the application.
- Template Engines: Supports various template engines to generate dynamic HTML pages.

Benefits

- Simplifies Development: Abstracts much of the complexity of server-side development.
- Flexibility: Highly customizable and can be tailored to the needs of the application.
- Performance: Lightweight framework with high performance.

MongoDB

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents, allowing for efficient querying and indexing.

Key Features

- Schema-less Design: Does not require a predefined schema, providing flexibility in data modeling.
- Scalability: Supports horizontal scaling through sharding.
- Rich Query Language: Offers a powerful query language for filtering and manipulating data.

Benefits

- Flexibility: Easily handles a variety of data types and structures.
- Performance: Optimized for high read and write throughput.
- Scalability: Suitable for handling large volumes of data and high-traffic applications.

Stripe API

The Stripe API is a powerful and user-friendly platform for integrating payment processing capabilities into web and mobile applications. It supports a variety of payment methods including credit/debit cards, bank transfers, and digital wallets like Apple Pay and Google Pay. Stripe is designed to handle both one-time payments and subscription billing, making it versatile for different business models.

\

Key Features

1. Payment Processing

- Supports global payments and multiple currencies.

- Handles both one-time and recurring payments.

2. Security

- PCI-DSS compliant to ensure transaction security.

- Utilizes tokenization to protect sensitive payment data.

- Supports 3D Secure for additional authentication.

3. Developer-Friendly

- Offers well-documented APIs and libraries for multiple programming languages (e.g., JavaScript, Ruby, Python).

- Provides a sandbox environment for testing integrations.

- Includes tools like Stripe.js and Stripe Elements for building secure and customizable payment forms.

2.2 Hardware Requirements

The hardware requirements for development and deployment include:

- **Development Machines**: Modern laptops or desktops with at least 8GB RAM, multi-core processors, and SSD storage for fast read/write operations.
- **Server Requirements**: A cloud-based server (e.g., AWS, DigitalOcean) with at least 2GB RAM and sufficient storage to host the backend server and database.

2.3 User (Questionnaires)

User requirements were gathered through questionnaires aimed at potential users (both customers and restaurant owners). The key questions included:

- **For Customers:**

  - What are the most important features you look for in an online food ordering platform?
  - How important is the ease of use and navigation of the website?
  - What payment methods do you prefer?
  - How important is order tracking to you?

Chapter 3

System Design & Specification

3.1 High-Level Design (HLD)

High-Level Design (HLD) involves the overall system architecture and the design of the primary components of the Food Ordering website using the MERN stack (MongoDB, Express.js, React.js, Node.js) and Stripe for payment processing. The following sections describe the project model, structure chart, Data Flow Diagram (DFD), Entity-Relationship (E-R) Diagram, and Unified Modeling Language (UML) diagrams.

3.1.1 Project Model

The project model outlines the architecture and the flow of data between the client and server. It is based on a typical three-tier architecture comprising:

1. **Presentation Layer (Frontend)**
   - o **Technologies:** React.js, Redux
   - o **Functionality:** User interface for customers and restaurant owners, including menu browsing, order placement, and order management.
2. **Application Layer (Backend)**
   - o **Technologies:** Node.js, Express.js
   - o **Functionality:** Handles business logic, API endpoints, user authentication, order processing, and payment integration with Stripe.
3. **Data Layer (Database)**
   - o **Technologies:** MongoDB
   - o **Functionality:** Stores user information, restaurant details, menu items, orders, and payment data.

3.1.2 Structure Chart

A structure chart represents the hierarchical decomposition of the system into modules and sub-modules.

- ➢ Main Application

  Frontend (React.js)
- ➢ User Interface

  State Management (Redux)

- ➢ Backend (Node.js + Express.js)

  API Endpoints

Authentication

Order Processing

- Payment Integration (Stripe)
- Database (MongoDB)

User Data

Menu Data

Order Data

Payment Data

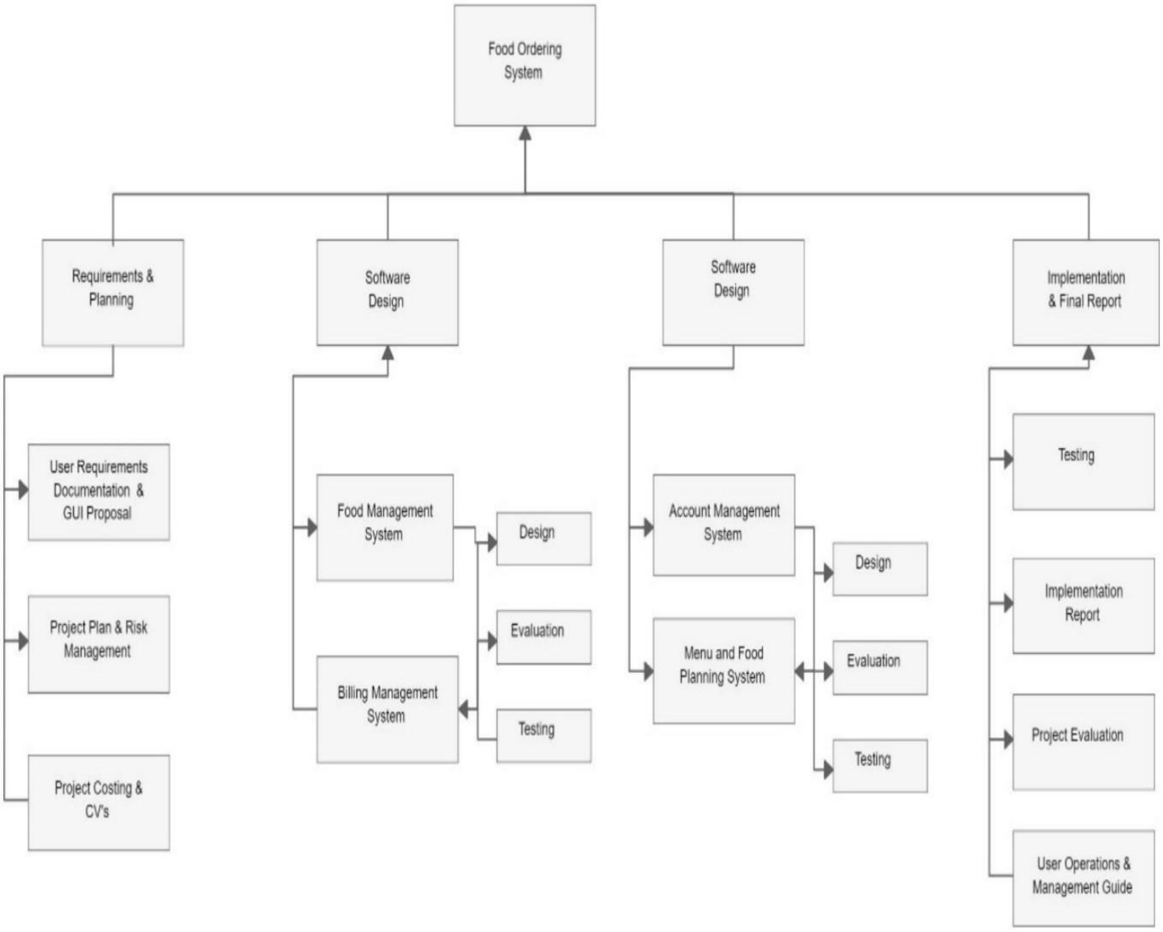

Food Ordering System
Requirements & Planning
Software Design
Software Design
Implementation & Final Report
User Requirements Documentation & GUI Proposal
Project Plan & Risk Management
Project Costing & CV's
Food Management System
Billing Management System
Design
Evaluation
Testing
Account Management System
Menu and Food Planning System
Design
Evaluation
Testing
Testing
Implementation Report
Project Evaluation
User Operations & Management Guide

Fig 3.1.2 Structure Chart

3.1.3 Data Flow Diagram(DFD)

A dataflow diagram is a graphical view of how data is processed in a system in terms of input and output.

Dataflow diagram symbol:

Symbol	Description
⟶	**Data Flow** – Data flow are pipelines through the packets of information flow.
⬭	**Process :** A Process or task performed by the system.
▭	**Entity :** Entity are object of the system. A source or destination data of a system.
═	**Data Store:** A place where data to be stored.

Context level DFD :

The context level data flow diagram (dfd) is describe the whole system. The (0) level dfd describe the all user module who operate the system. Below data flow diagram of online library management site shows the two user can operate the system Admin and Member user.

LEVEL 0

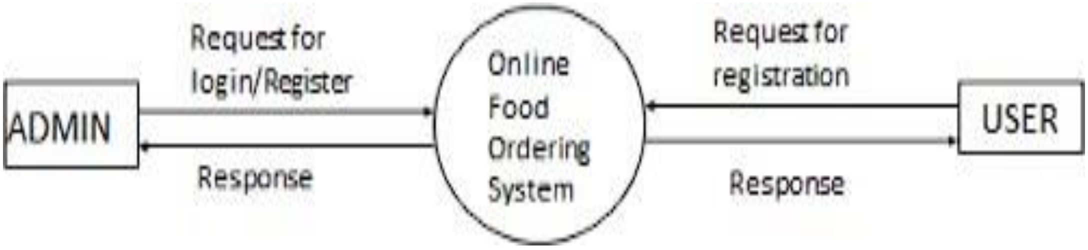


Figure3.1.3.1. Context level 0 DFD

LEVEL 1 Admin Side

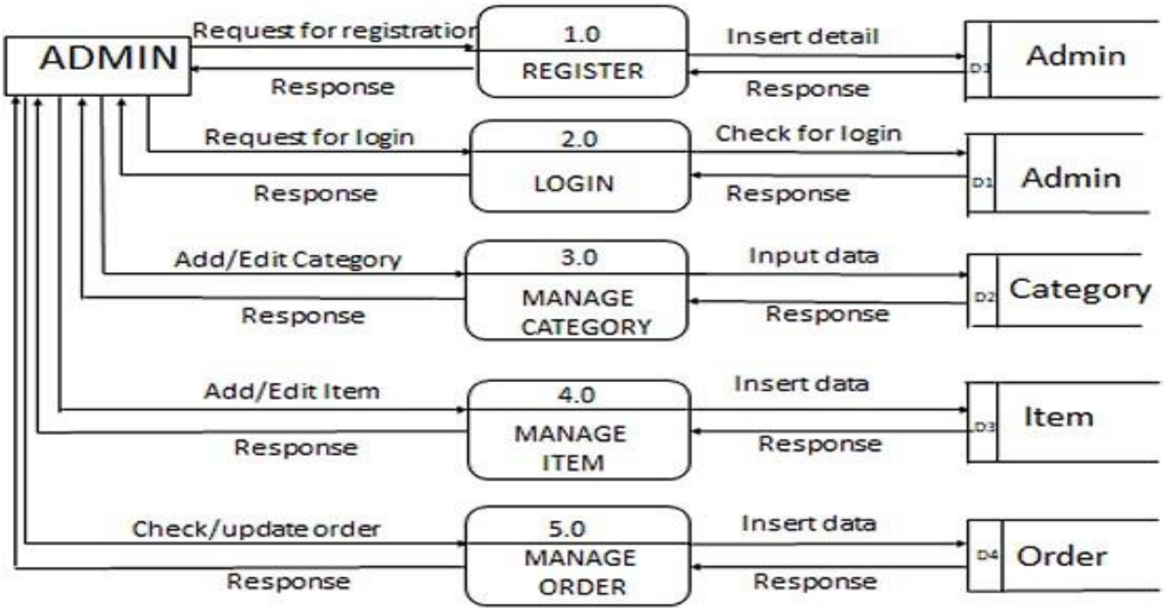


Figure 3.1.3.2 Context level 1 DFD

LEVEL 1 User Side

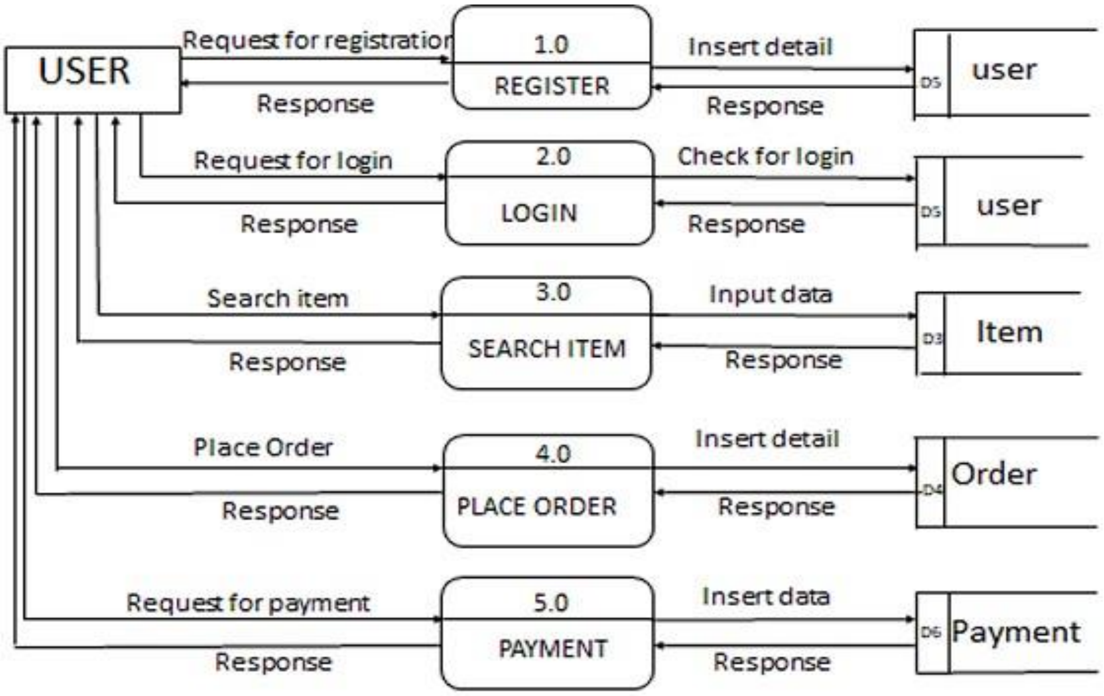


Figure 3.1.3.3 Context level 1 DFD

LEVEL 2 Admin Side

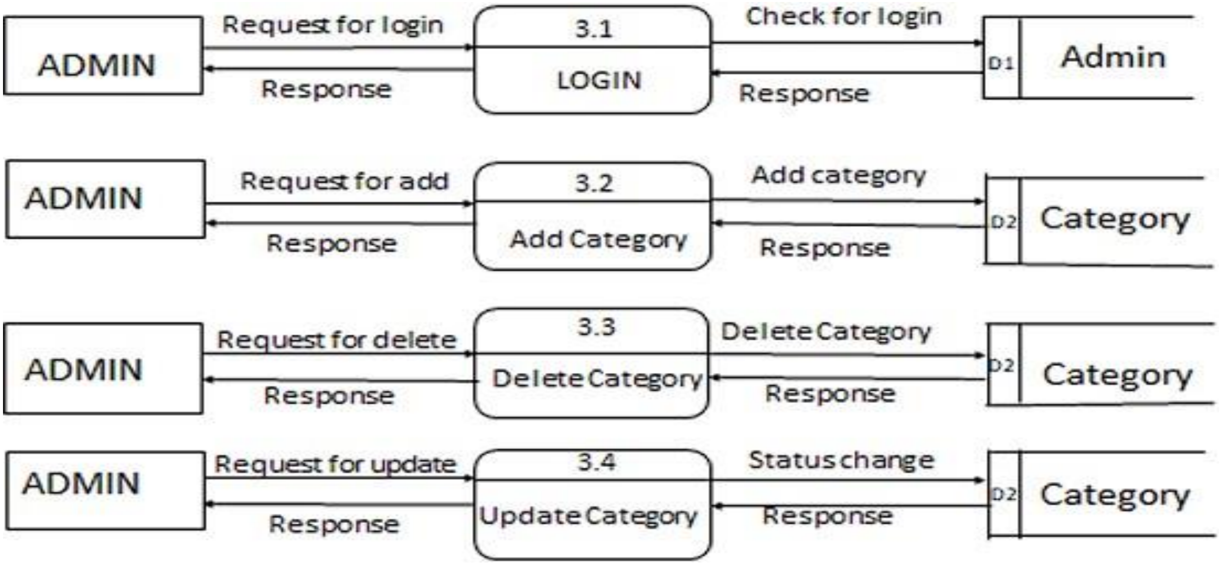


Figure 3.1.3.4 Context level 2 DFD

LEVEL 2 Admin Side

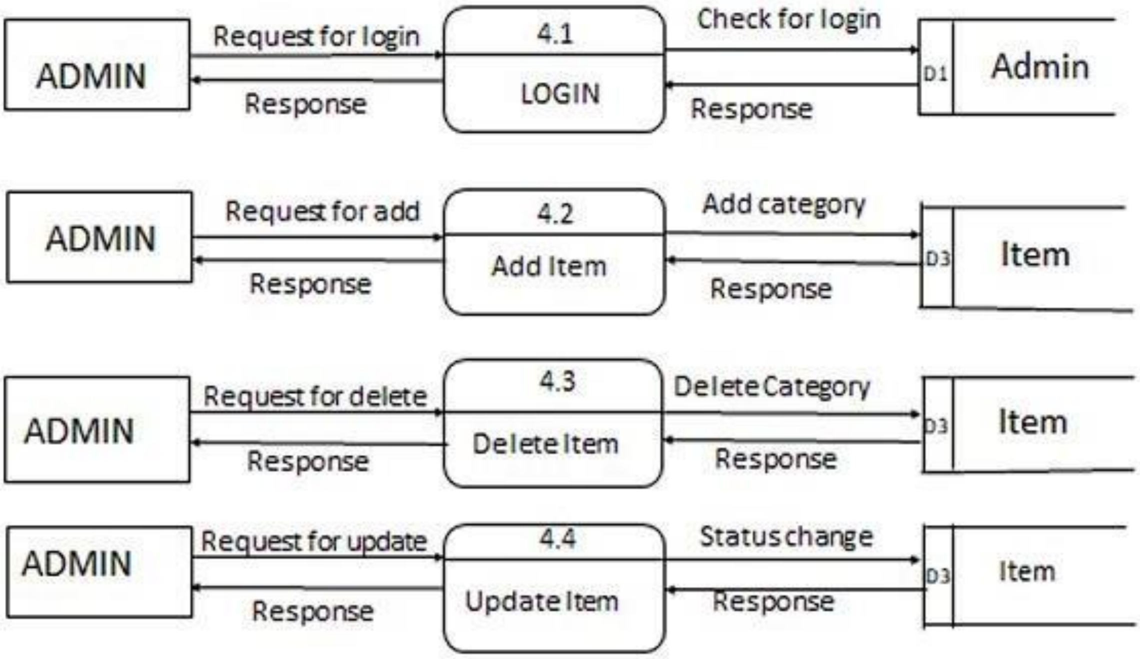


Figure 3.1.3.5 Context level 2 DFD

LEVEL 2 User Side

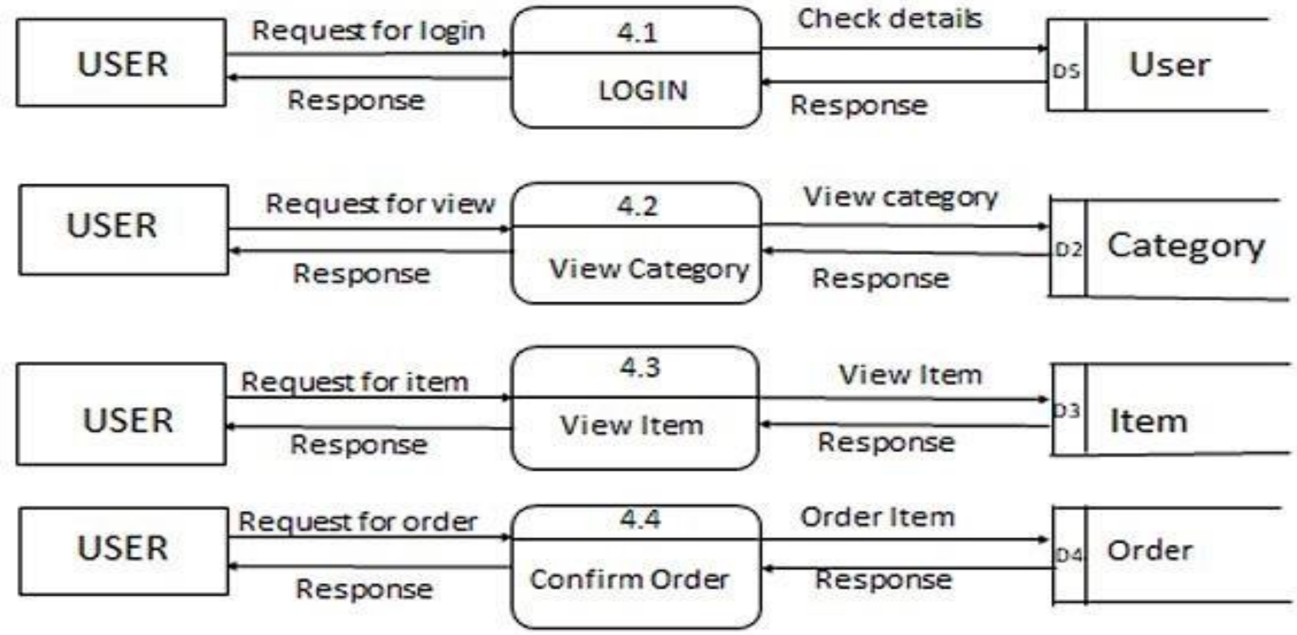


Figure 3.1.3.6 Context level 2 DFD

3.1.4 E- R Diagram

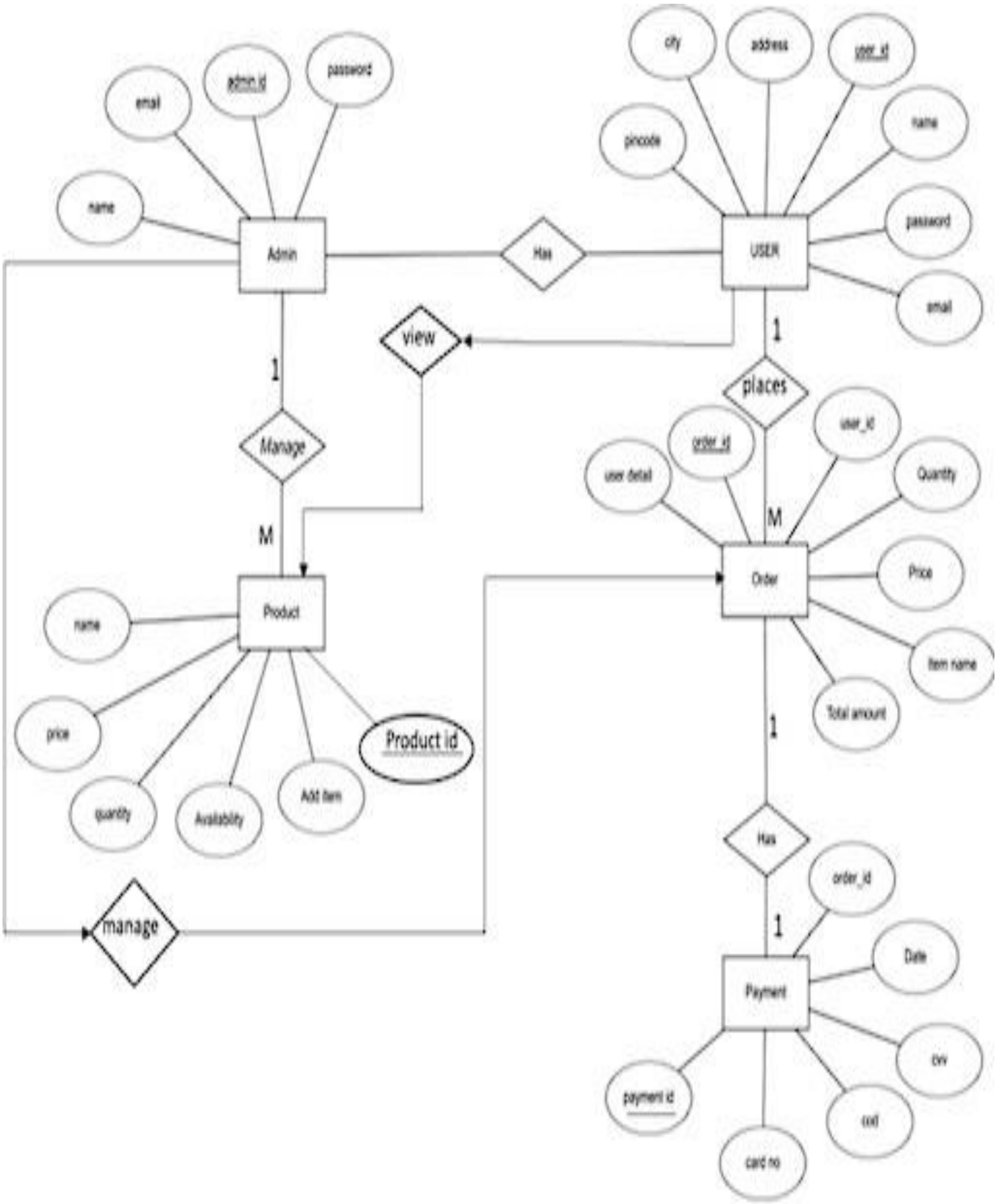


Figure 3.1.4 E R Diagram

3.1.5 UML (Use Case Diagram)

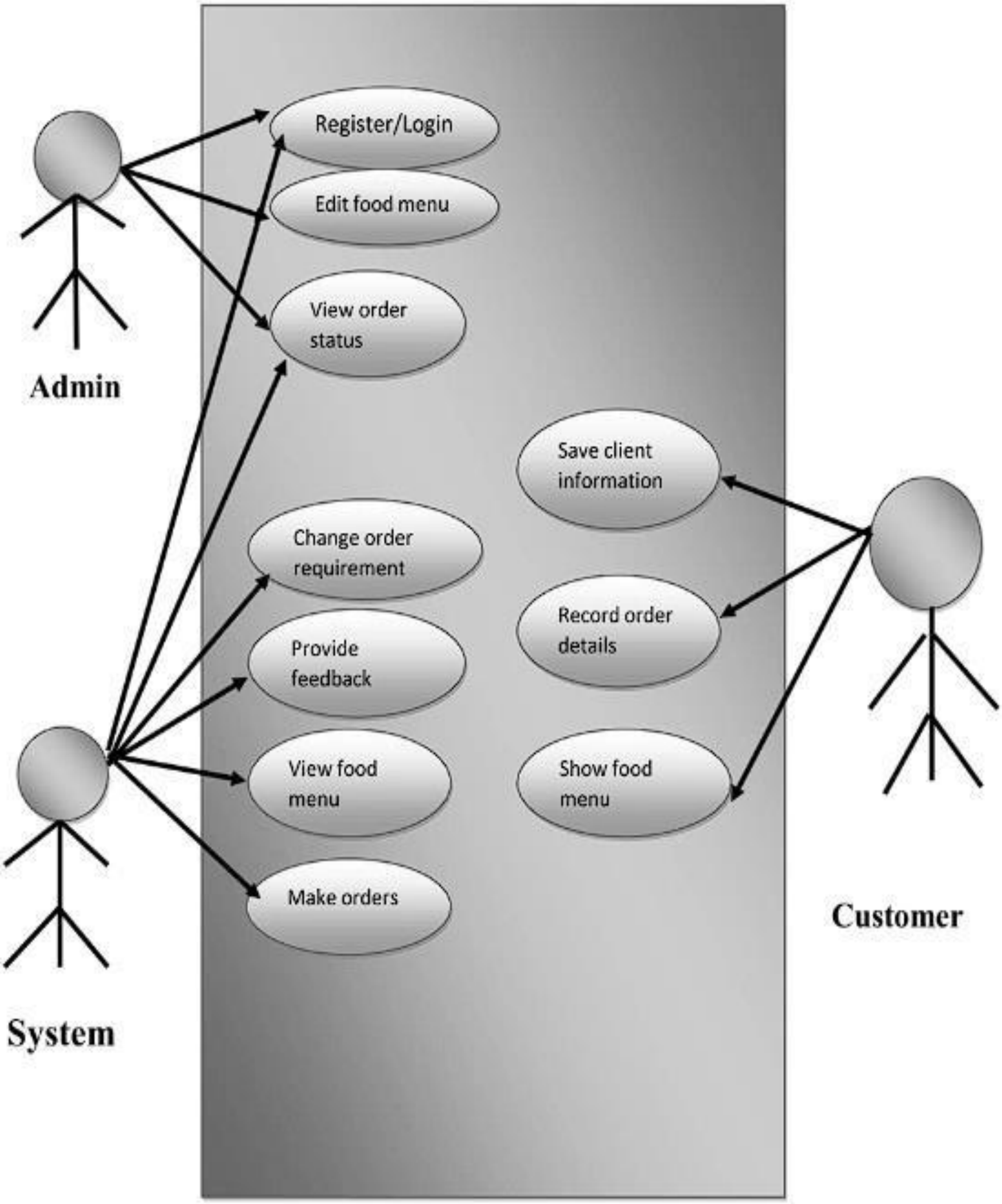
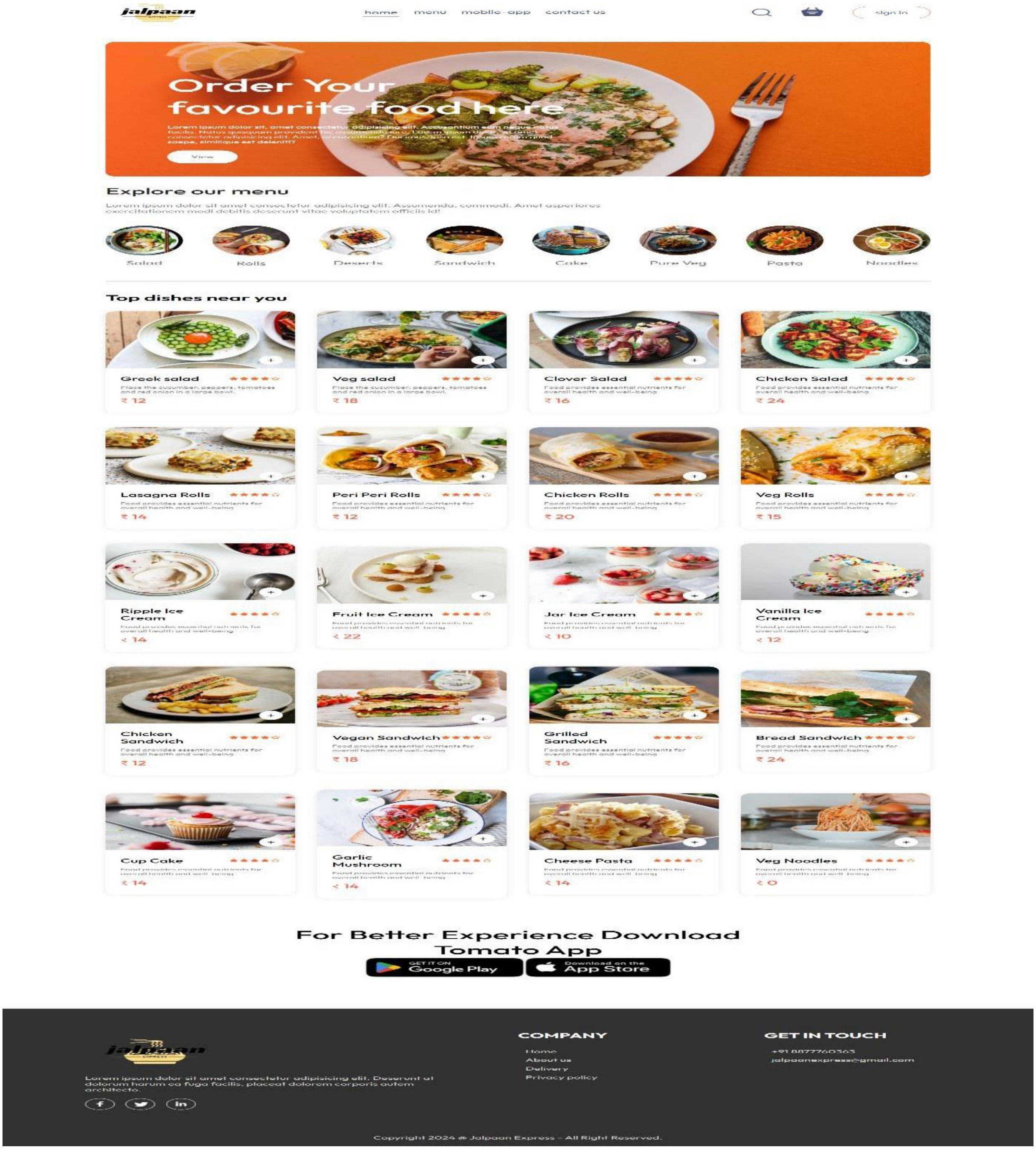


Figure 3.1.5 UCD

3.2 Low Level Design (LLD)

3.2.1 Screenshot Diagram:

Home Page



Lorem ipsum dolor sit, amet consectetur adipisicing elit. Accusantium eum neque natus facilis. Natus quisquam provident hic assumenda iure, Lorem ipsum dolor, sit amet consectetur adipisicing elit. Amet, accusantium? Ducimus, ipsa est. Ipsam earum quod saepe, similique est deleniti?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Assumenda, commodi. Amet asperiores exercitationem modi debitis deserunt vitae voluptatem officiis id!

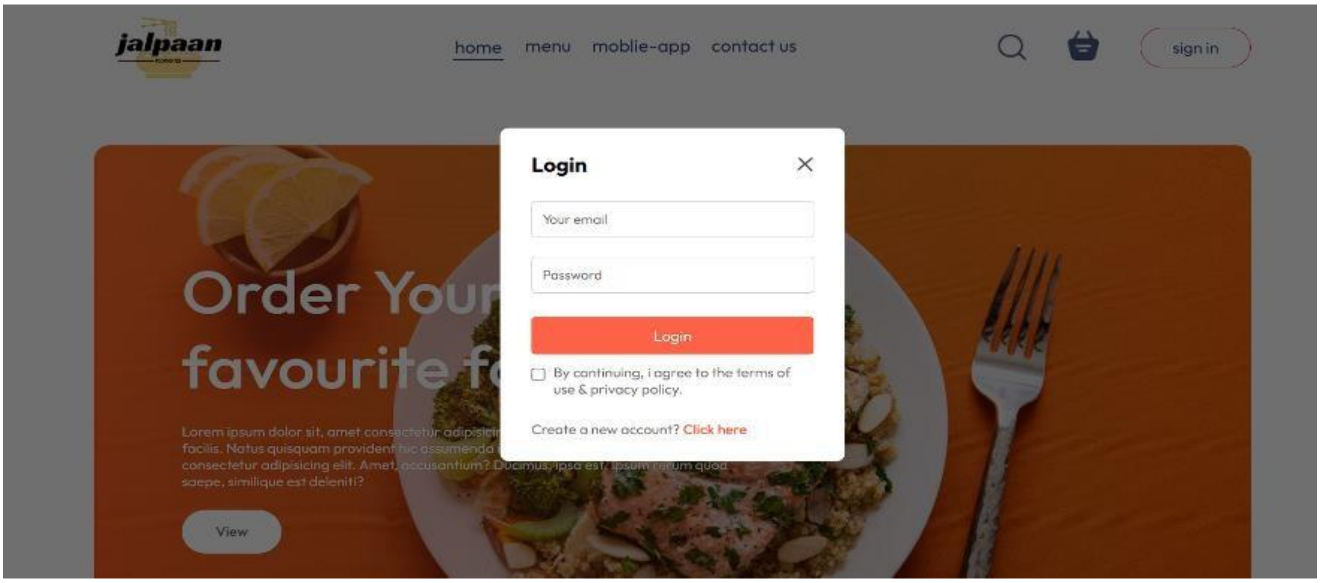
Place the cucumber, peppers, tomatoes and red onion in a large bowl.

Food provides essential nutrients for overall health and well-being

Lorem ipsum dolor sit amet consectetur adipisicing elit. Deserunt at dolorum harum ea fuga facilis, placeat dolorem corporis autem architecto.

Figure 3.2.1 Home page

Login



Order Your favourite f

Lorem ipsum dolor sit, amet consectetur adipisici facilis. Natus quisquam provident hic assumenda consectetur adipisicing elit. Amet, accusantium? Ducimus, ipsa est, ipsum rerum quod saepe, similique est deleniti?

□ By continuing, i agree to the terms of use & privacy policy.

Create a new account? Click here

Figure 3.2.2 Login page

Admin Dashboard

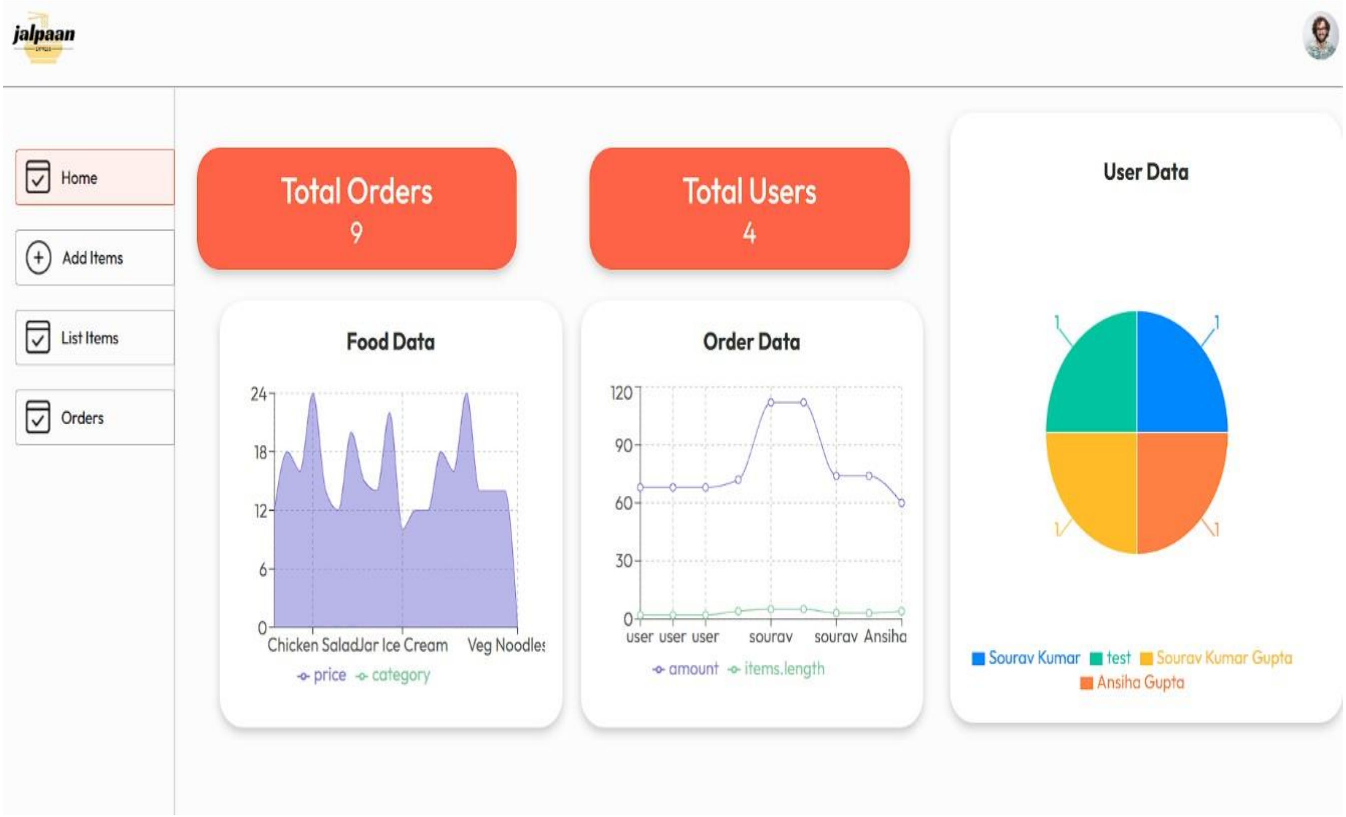
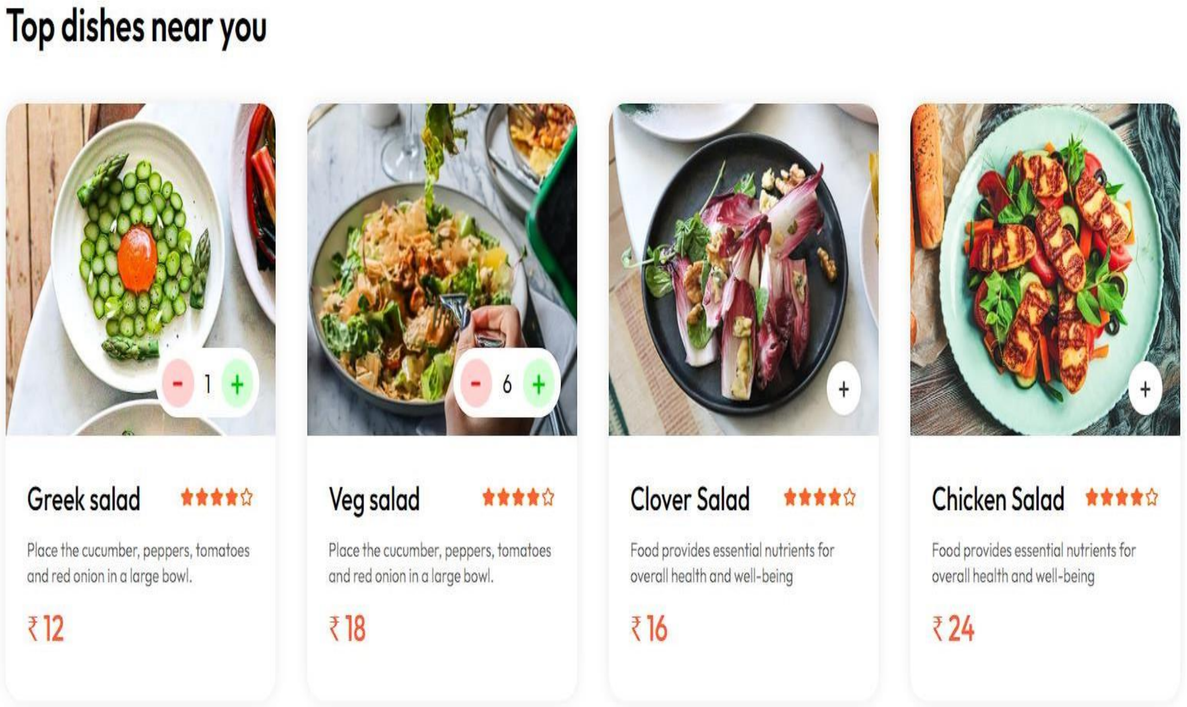


Figure 3.2.3 Admin Dashboard

Food Cart



Top dishes near you

Greek salad ★★★★☆

Place the cucumber, peppers, tomatoes and red onion in a large bowl.

₹ 12

Veg salad ★★★★☆

Place the cucumber, peppers, tomatoes and red onion in a large bowl.

₹ 18

Clover Salad ★★★★☆

Food provides essential nutrients for overall health and well-being

₹ 16

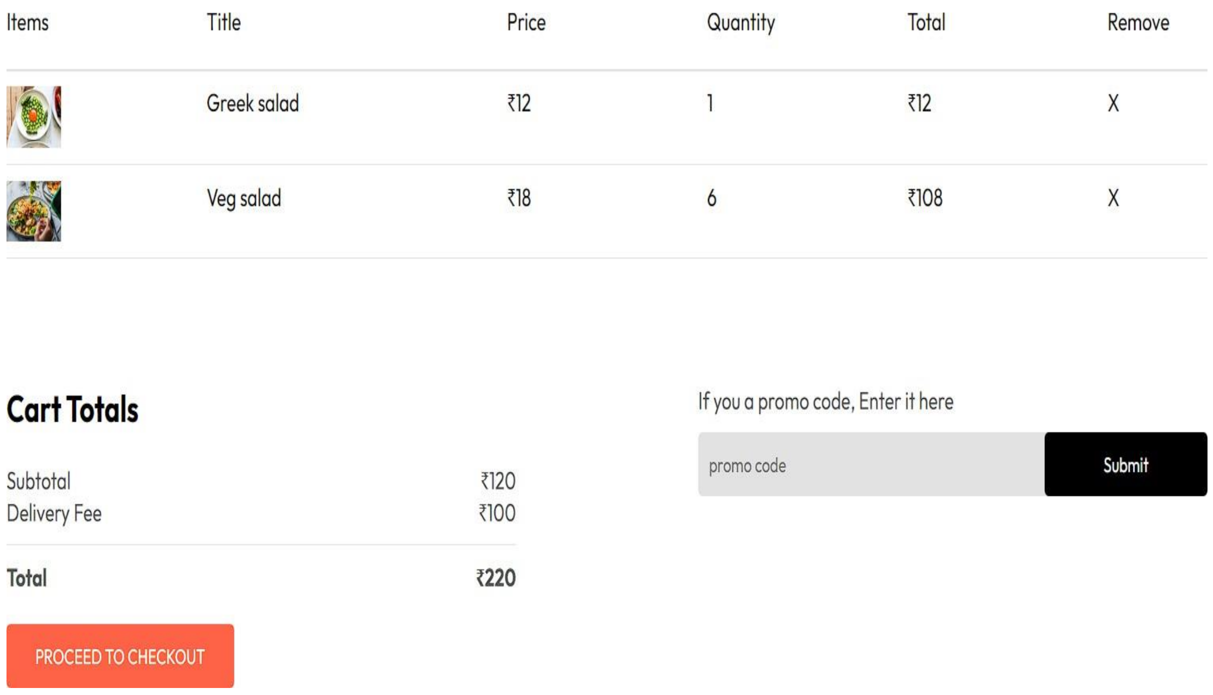
Chicken Salad ★★★★☆

Food provides essential nutrients for overall health and well-being

₹ 24

Figure 3.2.4 Food Cart

Cart Page



Items	Title	Price	Quantity	Total	Remove
	Greek salad	₹12	1	₹12	X
	Veg salad	₹18	6	₹108	X

Cart Totals

Subtotal	₹120
Delivery Fee	₹100
Total	₹220

If you a promo code, Enter it here

Figure 3.2.5 Cart Page

Chapter 4

Sample Code

Backend :

```
// we are using ES6 feature
import express from 'express'
import cors from 'cors'
import { connectDB } from './config/db.js'
import foodRouter from './routes/foodRoute.js'
import userRouter from './routes/userRoute.js'
import 'dotenv/config.js'
import cartRouter from './routes/cartRoute.js'
import orderRouter from './routes/orderRoute.js'
//app config
const app = express()
const port =4000
//middleware
app.use(express.json())
app.use(cors());
//db connection
connectDB();

//api end point
app.use("/api/food",foodRouter)
app.use("/image",express.static('uploads'))
app.use("/api/user",userRouter)
app.use("/api/cart",cartRouter)
app.use("/api/order",orderRouter)
app.get("/",(req,res)=>{
res.send("API  Working")
})
app.listen(port,()=>{
   console.log(`Server  Started  on  http://localhost:${port}`);
})
```

DataBase:

```
import mongoose from "mongoose";
export const connectDB = async ()=>{
   await
mongoose.connect('mongodb+srv://Food:Food12345@cluster0.zpwhhu3.mongodb.net/f
ood-del').then(()=>
      console.log("DB  Connected")
   );
}
```

Frontend

```
import React, { useState } from 'react'
import Navbar from './Component/Navbar/Navbar'
import { Route, Routes } from 'react-router-dom'
import LoginPopup from './Component/LoginPopup/LoginPopup'
import Verify from './pages/Verify/Verify'
import MyOrders from './pages/MyOrders/MyOrders'
import Profile from './pages/Profile/Profile'
const App = () => {
  const [showLogin,setShowLogin]=useState(false)
  return (
    <>
{showLogin?<LoginPopup  setShowLogin={setShowLogin}/>:<></>}
 <div className='app'>
 <Navbar  setShowLogin={setShowLogin}/>
 <Routes>
   <Route  path='/' element={<Home/>}/>
   <Route path='/card'element={<Card/>}/>
   <Route  path='/order'  element={<PlaceOder/>}/>
   <Route  path='/verify'  element={<Verify/>}/>
   <Route  path='/myorders'  element={<MyOrders/>}/>
   <Route  path='/profile'  element={<Profile/>}/>
  </Routes>
    </div>
    <Footer/>
    </>
export  default  App
```

Chapter 5

Software & Testing

SOFTWARE TESTING

General

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown errors.

TEST CASE

Testing, as already explained earlier, is the process of discovering all possible weak-points in the finalized software product. Testing helps to counter the working of sub-assemblies, components, assembly and the complete result. The software is taken through different exercises with the main aim of making sure that software meets the business requirement and user-expectations and doesn't fails abruptly. Several types of tests are used today. Each test type addresses a specific testing requirement.

Testing Techniques

A test plan is a document which describes approach, its scope, its resources and the schedule of aimed testing exercises. It helps to identify almost other test item, the features which are to be tested, its tasks, how will everyone do each task, how much the tester is independent, the environment in which the test is taking place, its technique of design plus the both the end criteria which is used, also rational of choice of theirs, and whatever kind of risk which requires emergency planning. It can be also referred to as the record of the process of test planning. Test plans are usually prepared with signification input from test engineers.

5.1 UNIT TESTING

In unit testing, the design of the test cases is involved that helps in the validation of the internal program logic. The validation of all the decision branches and internal code takes place. After the individual unit is completed it takes place. Plus, it is taken into account after the individual united is completed before integration. The unit test thus performs the basic level test at its component stage and test the particular business process, system configurations etc. The unit test ensures that the particular unique path of the process gets performed precisely to the documented specifications and contains clearly defined inputs with the results which are expected.

5.2 INTEGRATION TESTING

These tests are designed to test the integrated software items to determine whether if they really execute as a single program or application. The testing is event driven and thus is concerned with the basic outcome of field. The Integration tests demonstrate that the components were individually satisfaction, as already represented by successful unit testing, the components are apt and fine. This type of testing is specially aimed to expose the issues that come-up by the components combination.
documentation of the system and the user manual. configuration is taken place here to ensure predictable result and thus analysis of it. System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

5.3 WHITE BOX TESTING

The white box testing is the type of testing in which the internal components of the system software is open and can be processed by the tester. It is therefore a complex type of testing process. All the data structure, components etc. are tested by the tester himself to find out a possible bug or error. It is used in situation in which the black box is incapable of finding out a bug. It is a complex type of testing which takes more time to get applied.

5.4 BLACK BOX TESTING

The black box testing is the type of testing in which the internal components of the software is hidden and only the input and output of the system is the key for the tester to find out a bug. It is therefore a simple type of testing. A programmer with basic knowledge can also process this type of testing. It is less time consuming as compared to the white box testing. It is very successful for software which are less complex are straight-forward in nature. It is also less costly than white box testing.

Chapter 6

Conclusion & Limitations

Conclusion

The development of a Food Ordering website using the MERN stack (MongoDB, Express.js, React.js, Node.js) and Stripe for payment processing presents a robust, scalable, and user-friendly solution for online food ordering and delivery services. By leveraging the strengths of each technology, the system provides a seamless experience for customers, restaurant owners, and administrators.

Key Benefits

1. **Seamless User Experience:** React.js ensures a dynamic and responsive user interface, enhancing the customer experience.
2. **Efficient State Management:** Redux provides a predictable state container, improving application performance and maintainability.
3. **Asynchronous Communication:** Axios facilitates smooth and efficient data fetching and synchronization between the frontend and backend.
4. **Scalable Backend:** Node.js and Express.js enable the development of a scalable and high-performance server-side application.
5. **Flexible Database:** MongoDB offers a flexible and scalable database solution, ideal for handling diverse data types and large volumes of data.
6. **Secure Payments:** Integration with Stripe ensures secure and reliable payment processing, protecting user data and transaction information.

This comprehensive approach not only streamlines the process of ordering food online but also provides a solid foundation for future enhancements and scalability. The system's modular architecture allows for easy maintenance, updates, and the addition of new features, ensuring long-term viability and adaptability.

Limitations

Despite the numerous advantages, the Food Ordering website built using the MERN stack also faces several limitations:

1. **Initial Learning Curve:** Developers new to the MERN stack may face a steep learning curve due to the need to master multiple technologies (MongoDB, Express.js, React.js, Node.js) and their integration.
2. **Performance Overhead:** While MongoDB is flexible, its performance may not match that of relational databases for certain types of queries and large-scale transactional operations.

3. **Complex State Management:** As the application grows, managing state with Redux can become complex and may require additional libraries or patterns to handle efficiently.
4. **Security Risks:** Although Stripe provides secure payment processing, the overall system security must be diligently maintained to protect against potential vulnerabilities, such as cross-site scripting (XSS) or SQL injection attacks.
5. **Scalability Challenges:** While the MERN stack supports scalability, scaling each component (frontend, backend, database) independently can become complex, requiring careful planning and infrastructure management.
6. **Dependency Management:** The reliance on various libraries and frameworks means that keeping dependencies updated and compatible can be challenging, potentially leading to issues with library conflicts or deprecated functionalities.

Addressing Limitations

To mitigate these limitations, several strategies can be employed:

- **Continuous Learning and Training:** Providing developers with adequate training and resources to familiarize themselves with the MERN stack.
- **Performance Optimization:** Implementing best practices for database indexing, query optimization, and efficient data modeling in MongoDB.
- **Advanced State Management:** Utilizing advanced state management techniques and libraries, such as Redux-Saga or React Query, to handle complex state scenarios.
- **Security Best Practices:** Regularly auditing and updating the codebase for security vulnerabilities, implementing secure coding practices, and using automated security tools.
- **Scalable Architecture:** Designing the application architecture with scalability in mind, using containerization (e.g., Docker), microservices, and cloud-based services for efficient resource management.
- **Dependency Management Tools:** Leveraging tools like npm-check-updates and Dependabot to automate the process of keeping dependencies up-to-date.

Final Thoughts

The Food Ordering website using the MERN stack and Stripe represents a significant step forward in delivering efficient, scalable, and secure online food ordering solutions. By addressing the inherent limitations and continually optimizing the system, businesses can provide a top-tier user experience and stay competitive in the fast-paced digital marketplace.

References

[1.] Kirti Bhandge, Tejas Shinde, Dheeraj Ingale, Neeraj Solanki and Reshma Totare, "A Proposed System for Touchpad Based Food Ordering System Using Android Application", International Journal of Advanced Research in Computer Science Technology (IJARCST, 2015.)

[2.] Varsha Chavan, Priya Jadhav, Snehal Korade and Priyanka Teli, "Implementing Customizable Online Food Ordering System Using Web Based Application", International Journal of Innovative Science Engineering Technology(IJISET), 2015.

[3.] Resham Shinde, Priyanka Thakare, Neha Dhomne and Sushmita Sarkar, "Design and Implementation of Digital dining in Restaurants using Android", International Journal of Advance Research in Computer Science and Management Studies, 2014.

[4.] Ashutosh Bhargave, Niranjan Jadhav, Apurva Joshi, Prachi Oke and S. R Lahane, "Digital Ordering System for Restaurant Using Android", International Journal of Scientific and Research Publications, 2013.

[5.] K. Khairunnisa, J. Ayob, Mohd. Helmy A. Wahab, M. Erdi Ayob, M. Izwan Ayob and M. Afif Ayob, "The Application of Wireless Food Ordering System", MASAUM Journal of Computing, 2009.

[6.] Noor Azah Samsudin, Shamsul Kamal Ahmad Khalid, Mohd Fikry Akmal Mohd Kohar, Zulkifli Senin and Mohd Nor Ihkasan, "A customizable wireless food ordering system with real time customer feedback", IEEE Symposium on Wireless Technology and Applications(ISWTA), 2011.

[7.] Serhat Murat Alagoza and Haluk Hekimoglub, "A study on tam: analysis of customer attitudes in online food ordering system" in , Elsevier Ltd, 2012.

[8.] Patel Krishna, Patel Palak, Raj Nirali and Patel Lalit, "Automated Food Ordering System", International Journal of Engineering Research and Development (IJERD), 2015.

[9.] Mayur D. Jakhete and Piyush C. Mankar, "Implementation of Smart Restaurant with e-menu Card" International Journal of Computer Applications 2015 of Smart Restaurant with e-menu Card", International Journal of Computer Applications, 2015.

[10.] V. A. Bharadi, "Intelligent e-Restaurant using Android OS Mumbai", 2013.

[11.] M. Rajesh, "E-Restaurant: Online Restaurant Management", India: International Journal & Magazine of Engineering, 2015.

[12.] A. Bankar, "Intelligent Restaurant - Menu Ordering System", India: IOSR Journal of VLSI and Signal Processing(IOSR-JVSP), 2015.