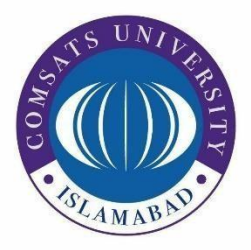



# COMSATS University Islamabad (CUI)

## Semester Project

## Semester Project Final Document

## for

## FeastFleet

Version 1.0

***By***

**Umar Hassan SP23-BSE-052**

**Zubaria Sajjad SP23-BSE-055**

**BSE-5-A**

***Supervisor Mr. Aamir Shabbir Pare***

***Bachelor of Science in Software Engineering (2023-2027)***


FeastFleet

## Contents

# PART 1 (SRS and Business Idea)

**Project Idea: FeastFleet Website**

FeastFleet is a web-based food delivery application designed to allow customers to browse nearby restaurants, place food orders, and track their deliveries in real time. It connects users with local eateries and simplifies the process of ordering food online using a modern MERN stack (MongoDB, Express.js, React.js, Node.js).

## Objectives

- Provide a seamless food ordering experience.
- Real-time order tracking and delivery updates.
- User profile and order history management.
- Filtering and searching restaurants by rating, category, and price.
- Secure authentication with persistent sessions.

## Scope

**In Scope:**

- Customer registration, login, and profile management
- Restaurant and menu browsing
- Real-time order placement and tracking
- Order history and reordering

**Out of Scope (Initial Phase):**

- Admin dashboard for restaurants
- Payment gateway integration
- Multi-vendor management

**Future Enhancements (Planned):**

- Online payments with Stripe/PayPal
- Push/email notifications
- Geolocation-based live tracking
- Admin management interface

## Modules and Features

### A. Authentication Module

- JWT-based login and registration
- Password hashing using bcrypt
- Secure session handling

### B. Restaurant Browsing

- Filter by location, category, rating, and price
- View menus with images and descriptions

### C. Order Management

- Add items to cart, checkout, and place orders
- Track order status: Preparing → Out for Delivery → Delivered

### D. User Profile

- Update name, email, and delivery address
- View and reorder previous orders

## Rationale

FeastFleet addresses the demand for a simplified and intuitive food delivery platform. With increasing reliance on web-based services, this application bridges the gap between customers and restaurants by delivering a seamless and real-time order management experience.

## Technology Stack

| **Component** | **Technology Used** |
|---|---|
| Frontend | React.js, Redux, Axios, Bootstrap/MaterialUI |
| Backend | Node.js, Express.js |
| Database | MongoDB |
| Auth/Security | JWT, bcrypt |
| Real-time Updates | Socket.io |

## System Architecture

- **Client:** Built with React, interacts with REST APIs via Axios
- **Server:** Express.js API handles routing and business logic
- **Database:** MongoDB collections for users, restaurants, orders
- **Real-time Layer:** Socket.io for instant updates (order tracking)

## Database Design

**MongoDB Collections:**

- Users
- { userId, name, email, passwordHash, address }
- Restaurants
- { restaurantId, name, cuisine, location, rating, menuItems[] }
- Orders
- { orderId, userId, restaurantId, items[], totalPrice, status, timestamp }
- menuItems
- { itemId, name, description, price, imageUrl }

# PART 2 (UI)

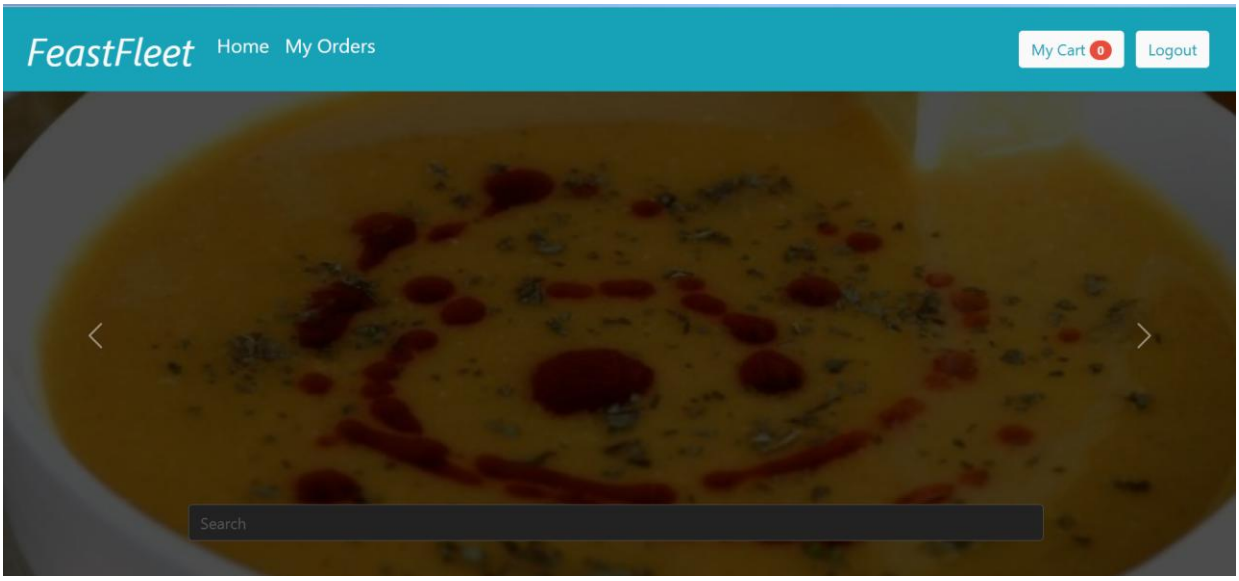



*Figure 1 Home Screen*

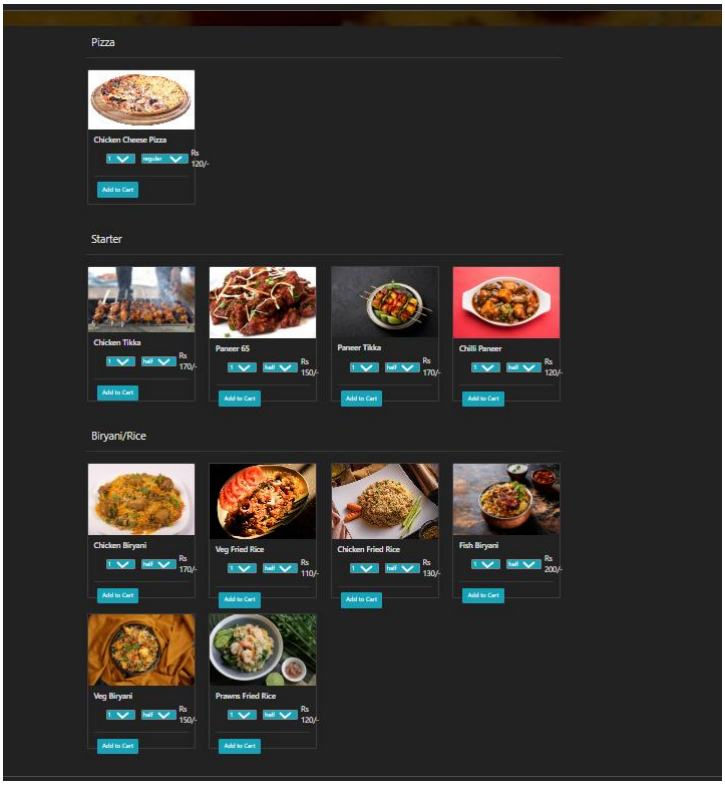



*Figure 2 Menu Section*

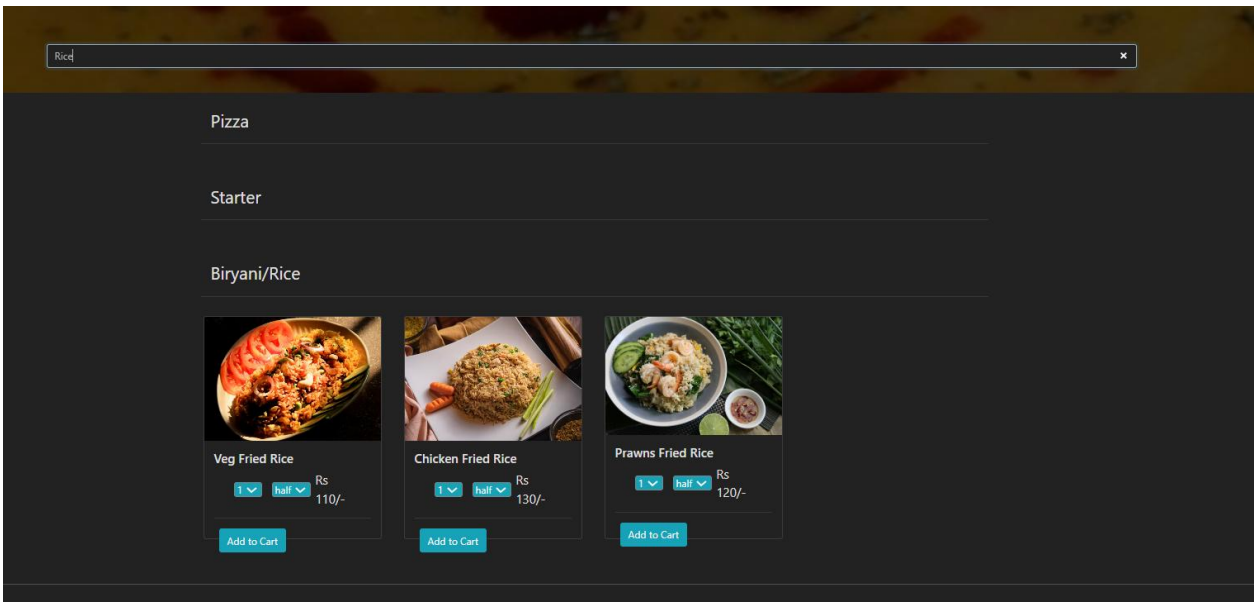



*Figure 3 Search Result Screen*

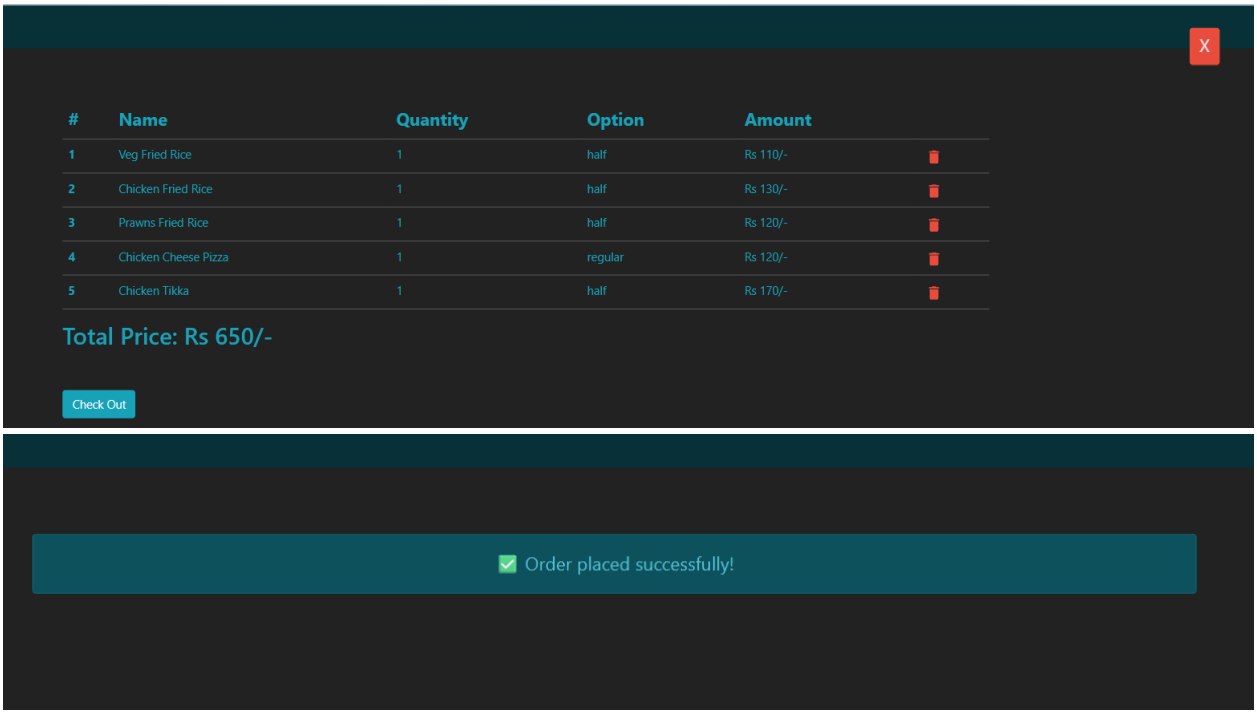
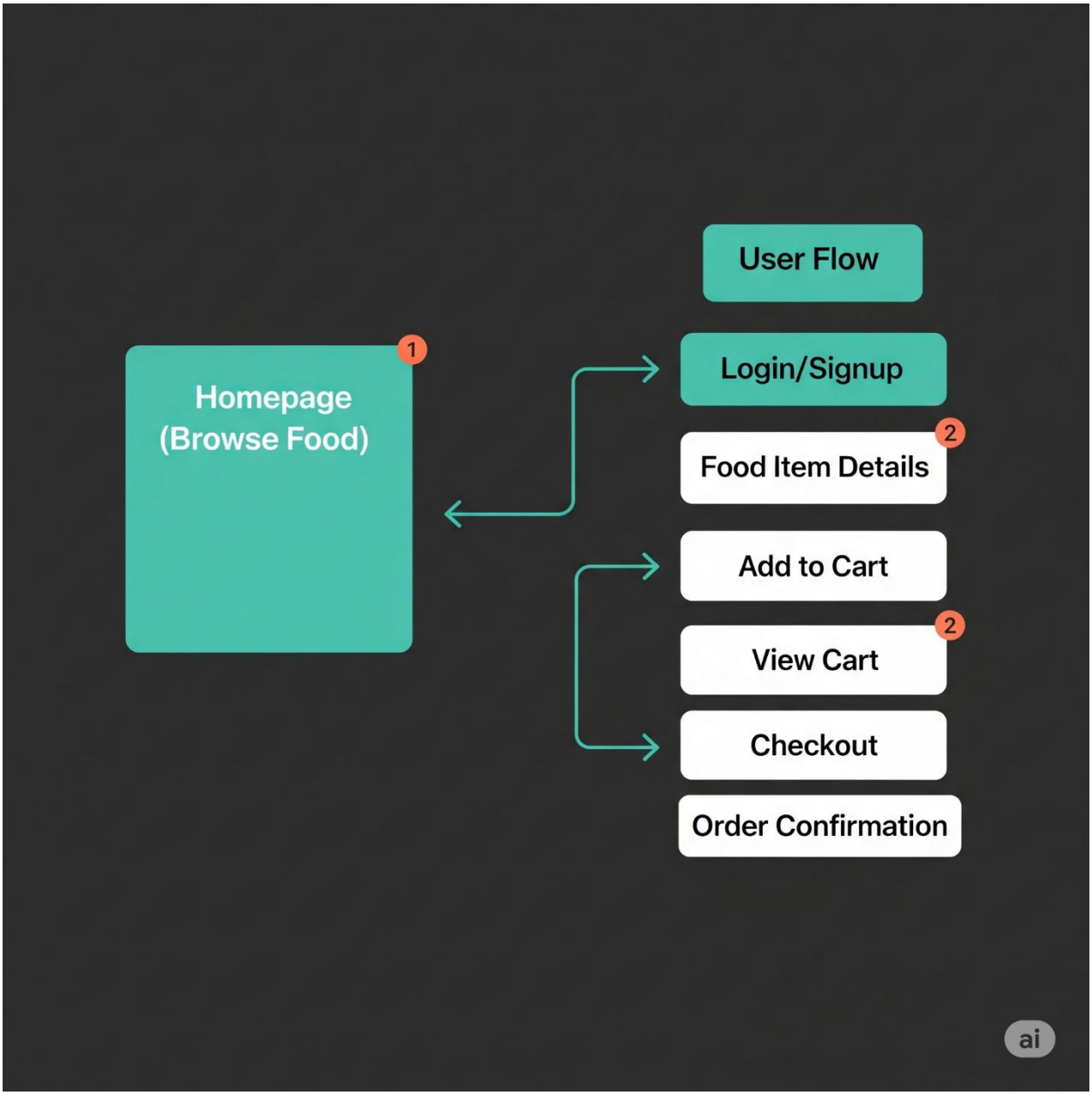
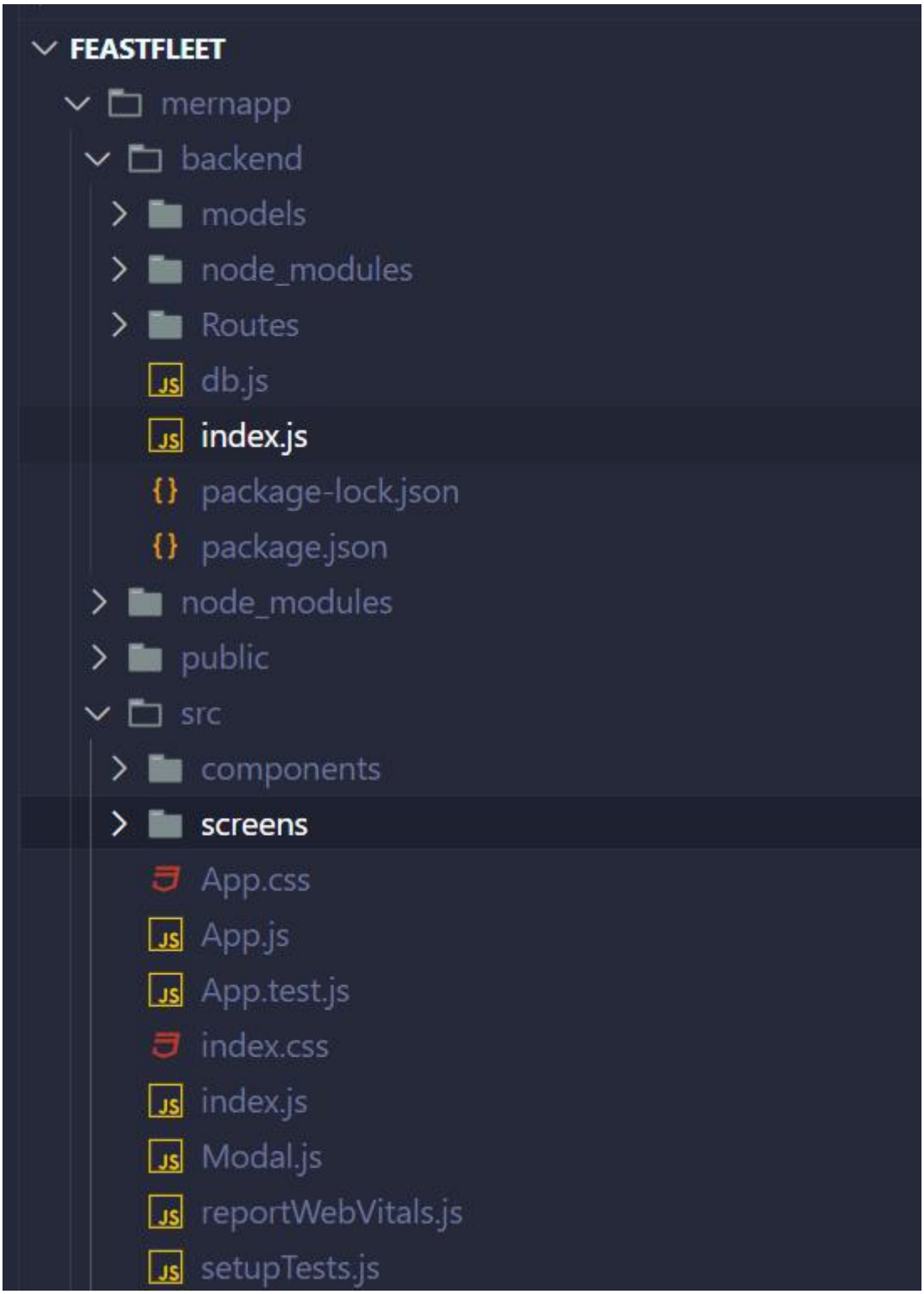



*Figure 4 Order Section Screen*

# Story Boarding



## PART 3 (File Structure)



# PART 4 (Frontend)

## Components:

### Card.js:

```
import React, { useRef, useState, useEffect } from 'react';
import { useDispatchCart, useCart } from './ContextReducer';
export default function Card(props) {
  let dispatch = useDispatchCart();
  let data = useCart();
  const priceRef = useRef();
  let options = props.options;
  let priceOptions = Object.keys(options);
  const [qty, setQty] = useState(1);
  const [size, setSize] = useState('');
  useEffect(() => {
    if (priceRef.current) {
      setSize(priceRef.current.value);
    }
  }, []);
  let finalPrice = qty * parseInt(options[size] || 0);
  const handleAddToCart = async () => {
    const existingFood = data.find(
      (item) => item.id === props.foodItem._id && item.size === size
    );
    if (existingFood) {
      await dispatch({
        type: 'UPDATE',
        id: props.foodItem._id,
        price: finalPrice,
        qty: qty,
        size: size,
      });
    } else {
      await dispatch({
        type: 'ADD',
        id: props.foodItem._id,
        name: props.foodItem.name,
        price: finalPrice,
```

```
      qty: qty,
      size: size,
    });
  }
};
return (
  <div className="card mt-3 mx-2" style={{ width: '18rem', maxHeight: '360px' }}>
    <img
      className="card-img-top"
      src={props.foodItem.img}
      alt={props.foodItem.name}
      style={{ objectFit: 'cover', height: '200px' }}
    />
    <div className="card-body">
      <h5 className="card-title">{props.foodItem.name}</h5>
      <div className="container w-100 d-flex align-items-center">
        <select
          className="m-2 h-100 bg-info rounded"
          value={qty}
          onChange={(e) => setQty(parseInt(e.target.value))}
        >
          {Array.from({ length: 6 }, (_, i) => (
            <option key={i + 1} value={i + 1}>
              {i + 1}
            </option>
          ))}
        </select>
        <select
          className="m-2 h-100 bg-info rounded"
          ref={priceRef}
          value={size}
          onChange={(e) => setSize(e.target.value)}
        >
          {priceOptions.map((option) => (
            <option key={option} value={option}>
              {option}
            </option>
          ))}
        </select>
        <div className="d-inline h-100 fs-5">Rs {finalPrice}/-</div>
```

```
      </div>
      <hr />
      <button className="btn btn-info ms-2" onClick={handleAddToCart}>
        Add to Cart
      </button>
    </div>
  </div>
 );
}
```

## Carousel.js:

```
import React from 'react';
export default function Carousel() {
 return (
  <div>
    <div id="carouselExampleFade" className="carousel slide carousel-fade custom-carousel" data-bs-ride="carousel" style={{objectFit:"contain !important"}}>
      <div className="carousel-inner" id='carousel'>
        <div className='carousel-caption' style={{zIndex:10}}>
        <form className="d-flex">
    <input className="form-control me-2" type="search" placeholder="Search" aria-label="Search"/>
    <button className="btn btn-outline-info text-white" type="submit">Search</button>
  </form>
        </div>
       <div className="carousel-item active">
        <img src="https://www.themealdb.com/images/media/meals/58oia61564916529.jpg" className="d-block w-100" style={{ filter: "brightness(30%)" }} alt="Burger" />
       </div>
       <div className="carousel-item">
        <img src="https://www.themealdb.com/images/media/meals/xrttsx1487339558.jpg" className="d-block w-100" style={{ filter: "brightness(30%)" }} alt="Pizza" />
       </div>
       <div className="carousel-item">
        <img src="https://www.themealdb.com/images/media/meals/wtsvxx1511296896.jpg" className="d-block w-100" style={{ filter: "brightness(30%)" }} alt="Pasta" />
       </div>
      </div>
```

```
      <button className="carousel-control-prev" type="button" data-bs-
target="#carouselExampleFade" data-bs-slide="prev">
        <span className="carousel-control-prev-icon" aria-hidden="true"></span>
        <span className="visually-hidden">Previous</span>
      </button>
      <button className="carousel-control-next" type="button" data-bs-
target="#carouselExampleFade" data-bs-slide="next">
        <span className="carousel-control-next-icon" aria-hidden="true"></span>
        <span className="visually-hidden">Next</span>
      </button>
    </div>
  </div>
 );
}
```

## ContextReducer.js:

```
import React, { createContext, useContext, useReducer } from 'react'
const CartStateContext = createContext();
const CartDispatchContext = createContext();
const reducer = (state, action) => {
  switch (action.type) {
    case "ADD":
      return [...state, { id: action.id, name: action.name, qty: action.qty, size: action.size,
price: action.price, img: action.img }]
    case "REMOVE":
      let newArr = [...state]
      newArr.splice(action.index, 1)
      return newArr;
    case "DROP":
      let empArray = []
      return empArray
    case "UPDATE":
      let arr = [...state]
      arr.find((food, index) => {
        if (food.id === action.id) {
          arr[index] = { ...food, qty: parseInt(action.qty) + food.qty, price: action.price +
food.price }
        }
        return arr
      })
```

```
            return arr
        default:
            console.log("Error in Reducer");
    }
};
export const CartProvider = ({ children }) => {
    const [state, dispatch] = useReducer(reducer, [])

    return (
        <CartDispatchContext.Provider value={dispatch}>
            <CartStateContext.Provider value={state}>
                {children}
            </CartStateContext.Provider>
        </CartDispatchContext.Provider>
    )
}
export const useCart = () => useContext(CartStateContext);
export const useDispatchCart = () => useContext(CartDispatchContext);
```

## Footer.js:

```
import React from 'react'
import {Link} from 'react-router-dom'
export default function Footer() {
  return (
    <div>
    <footer className="d-flex flex-wrap justify-content-between align-items-center py-3 my-4
border-top">
    <div className="col-md-4 d-flex align-items-center">
      <Link to="/" className="mb-3 me-2 mb-md-0 text-muted text-decoration-none lh-1">
      </Link>
      <span className="text-muted">© 2025 FeastFleet, Inc</span>
    </div>
  </footer>
    </div>
  )
}
```

## Navbar.js:

```
import React, { useState } from 'react';
import { Link } from 'react-router-dom';
import Badge from 'react-bootstrap/Badge';
import Modal from '../Modal';
import Cart from '../screens/Cart';
import { useCart } from './ContextReducer';
export default function Navbar() {
    let data = useCart();
    const [cartView, setCartView] = useState(false);
    const isLoggedIn = localStorage.getItem("authToken");
    const handleLogout = () => {
        localStorage.removeItem("authToken");
        localStorage.removeItem("userEmail");
        window.location.reload();
    };
    return (
        <div>
            <nav className="navbar navbar-expand-lg navbar-dark bg-info">
                <div className="container-fluid">
                    <Link className="navbar-brand fs-1 fst-italic" to="/">FeastFleet</Link>
                    <button className="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav"
                        aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                        <span className="navbar-toggler-icon"></span>
                    </button>
                    <div className="collapse navbar-collapse" id="navbarNav">
                        <ul className="navbar-nav me-auto mb-2">
                            <li className="nav-item">
                                <Link className="nav-link active fs-5" to="/">Home</Link>
                            </li>
                            {isLoggedIn && (
                                <li className="nav-item">
                                    <Link className="nav-link active fs-5" to="/myorder">My
Orders</Link>
                                </li>
                            )}
                        </ul>
                        <div className='d-flex'>
```

```
                    {!isLoggedIn ? (
                      <>
                        <Link className="btn bg-white text-info mx-1"
to="/login">Login</Link>
                        <Link className="btn bg-white text-info mx-1" to="/createuser">Sign
Up</Link>
                      </>
                    ) : (
                      <>
                        <button className='btn bg-white text-info mx-2' onClick={() =>
setCartView(true)}>
                          My Cart{" "}
                          <Badge pill bg="danger">{data.length}</Badge>
                        </button>
                        {cartView && (
                          <Modal onClose={() => setCartView(false)}>
                            <Cart />
                          </Modal>
                        )}
                        <button
                          className="btn bg-white text-info mx-1"
                          onClick={handleLogout}>
                          Logout
                        </button>
                      </>
                    )}
                  </div>
                </div>
              </div>
            </nav>
          </div>
        );
}
```

## Screens:

### Cart.js:

```
import React, { useState } from 'react';
import DeleteIcon from '@mui/icons-material/Delete';
import { useCart, useDispatchCart } from '../components/ContextReducer';
export default function Cart() {
  let data = useCart();
  let dispatch = useDispatchCart();
  const [orderSuccess, setOrderSuccess] = useState(false);
  if (data.length === 0 && !orderSuccess) {
    return (
      <div>
        <div className='m-5 w-100 text-center fs-3 text-info'>
          The Cart is Empty!
        </div>
      </div>
    );
  }
  const handleCheckOut = async () => {
    try {
      let userEmail = localStorage.getItem("userEmail");

      let response = await fetch("http://localhost:5000/api/orderData", {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json'
        },
        body: JSON.stringify({
          order_data: data,
          email: userEmail,
          order_date: new Date().toDateString()
        })
      });
      if (response.status === 200) {
        dispatch({ type: "DROP" });
        setOrderSuccess(true);
      } else {
        setOrderSuccess(false);
      }
```

```
    } catch (error) {
      setOrderSuccess(false);
    }
  };
  let totalPrice = data.reduce((total, food) => total + food.price, 0);
  return (
    <div>
      <div className='container m-auto mt-5 table-responsive table-responsive-sm table-
responsive-md'>

        {orderSuccess && (
          <div className="alert alert-info text-center fs-5" role="alert">
            ☑ Order placed successfully!
          </div>
        )}

        {!orderSuccess && (
          <>
            <table className='table table-hover'>
              <thead>
                <tr className='fs-4'>
                  <th scope='col' className='text-info'>#</th>
                  <th scope='col' className='text-info'>Name</th>
                  <th scope='col' className='text-info'>Quantity</th>
                  <th scope='col' className='text-info'>Option</th>
                  <th scope='col' className='text-info'>Amount</th>
                  <th scope='col' className='text-info'></th>
                </tr>
              </thead>
              <tbody>
                {data.map((food, index) => (
                  <tr key={index}>
                    <th scope='row' className='text-info'>{index + 1}</th>
                    <td className='text-info'>{food.name}</td>
                    <td className='text-info'>{food.qty}</td>
                    <td className='text-info'>{food.size}</td>
                    <td className='text-info'>Rs {food.price}/-</td>
                    <td>
                      <button type="button" className="btn p-0">
                        <DeleteIcon
```

```
                                    className="text-danger"
                                    onClick={() => dispatch({ type: "REMOVE", index })}
                                />
                            </button>
                        </td>
                    </tr>
                ))}
            </tbody>
        </table>
        <div>
            <h1 className='fs-2 text-info'>Total Price: Rs {totalPrice}/-</h1>
        </div>
        <div>
            <button className='btn bg-info text-white mt-5'
onClick={handleCheckOut}>
                Check Out
            </button>
        </div>
    </>
    )}
    </div>
  </div>
  );
}
```

## Home.js:

```
import React, { useEffect, useState } from 'react';
import Navbar from '../components/Navbar';
import Footer from '../components/Footer';
import Card from '../components/Card';
export default function Home() {
  const [search, setSearch] = useState('');
  const [foodCat, setFoodCat] = useState([]);
  const [foodItem, setFoodItem] = useState([]);
  const loadData = async () => {
    let response = await fetch("http://localhost:5000/api/foodData", {
      method: "POST",
      headers: {
        'Content-Type': 'application/json'
      }
    });
```

```
  response = await response.json();
  setFoodItem(response[0]);
  setFoodCat(response[1]);
 };
 useEffect(() => {
  loadData();
 }, []);
 return (
  <>
   <div><Navbar /></div>
   <div>
    <div className="carousel-inner" id='carousel'>
     <div className='carousel-caption' style={{ zIndex: 10 }}>
      <div className="d-flex justify-content-center">
       <input
        className="form-control me-2"
        type="search"
        placeholder="Search"
        aria-label="Search"
        value={search}
        onChange={(e) => setSearch(e.target.value)}
       />
      </div>
     </div>
     <div className="carousel-item active">
      <img src="https://www.themealdb.com/images/media/meals/58oia61564916529.jpg"
classNameName="d-block w-100" style={{ filter: "brightness(30%)" }} alt="Burger" />
     </div>
     <div className="carousel-item">
      <img src="https://www.themealdb.com/images/media/meals/xrttsx1487339558.jpg"
classNameName="d-block w-100" style={{ filter: "brightness(30%)" }} alt="Pizza" />
     </div>
     <div className="carousel-item">
      <img src="https://www.themealdb.com/images/media/meals/wtsvxx1511296896.jpg"
classNameName="d-block w-100" style={{ filter: "brightness(30%)" }} alt="Pasta" />
     </div>
     <button classNameName="carousel-control-prev" type="button" data-bs-
target="#carouselExampleFade" data-bs-slide="prev">
      <span classNameName="carousel-control-prev-icon" aria-hidden="true"></span>
      <span classNameName="visually-hidden">Previous</span>
```

```
        </button>
        <button className="carousel-control-next" type="button" data-bs-
target="#carouselExampleFade" data-bs-slide="next">
          <span className="carousel-control-next-icon" aria-hidden="true"></span>
          <span className="visually-hidden">Next</span>
        </button>
      </div>
    </div>
    <div className='container mt-4'>
      {
       foodCat.length !== 0
        ? foodCat.map((data) => (
          <div key={data._id} className='mb-5'>
            <div className="fs-3 m-3">{data.CategoryName}</div>
            <hr />
            <div className='row'>
              {
                foodItem.length !== 0
                 ? foodItem
                    .filter(
                      (item) =>
                        item.CategoryName === data.CategoryName &&
                        item.name.toLowerCase().includes(search.toLowerCase())
                    )
                    .map((filterItems) => (
                      <div key={filterItems._id} className='col-12 col-md-6 col-lg-3 mb-4 px-
2'>
                        <Card foodItem={filterItems}
                          options={filterItems.options[0]}
                        ></Card>
                      </div>
                    ))
                 : <div>No such data found.</div>
              }
            </div>
          </div>
        ))
        : <div>Loading categories...</div>
      }
    </div>
```

```
      <div><Footer /></div>
    </>
  );
}
```

## Login.js:

```
import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom';
export default function Login() {
  const [credentials, setcredentials] = useState({
    email: "", password: ""
  });
  const navigate = useNavigate();
  const handleSubmit = async (e) => {
    e.preventDefault();
    const response = await fetch("http://localhost:5000/api/loginuser", {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        email: credentials.email,
        password: credentials.password
      })
    });
    const json = await response.json();
    console.log(json);
    if (!json.success) {
      alert("Enter valid credentials");
    } else {
          localStorage.setItem("userEmail", credentials.email);
      localStorage.setItem("authToken", json.authToken);
      console.log(localStorage.getItem("authToken"))
      navigate("/");
    }
  };
  const onChange = (event) => {
    setcredentials({ ...credentials, [event.target.name]: event.target.value });
```

```
  };
  return (
    <div className='container'>
      <form onSubmit={handleSubmit}>
        <div className="mb-3">
          <label htmlFor="exampleInputEmail1" className="form-label">Email
address</label>
          <input type="email" className="form-control" name='email'
value={credentials.email} onChange={onChange} id="exampleInputEmail1" aria-
describedby="emailHelp" />
          <div id="emailHelp" className="form-text">We'll never share your email with anyone
else.</div>
        </div>
        <div className="mb-3">
          <label htmlFor="exampleInputPassword1" className="form-label">Password</label>
          <input type="password" className="form-control" name='password'
value={credentials.password} onChange={onChange} id="exampleInputPassword1" />
        </div>
        <button type="submit" className="m-3 btn btn-info">Submit</button>
        <Link to="/createuser" className='m-3 btn btn-info'>Register here</Link>
      </form>
    </div>
  );
}
```

## MyOrder.js:

```
import React, { useEffect, useState } from 'react';
import Footer from '../components/Footer';
import Navbar from '../components/Navbar';
export default function MyOrder() {
    const [orderData, setOrderData] = useState(null);

    const fetchMyOrder = async () => {
        const userEmail = localStorage.getItem('userEmail');
        if (!userEmail) {
            setOrderData([]);
            return;
        }
        try {
            const res = await fetch("http://localhost:5000/api/myorderData", {
```

```
            method: 'POST',
            headers: {
              'Content-Type': 'application/json'
            },
            body: JSON.stringify({ email: userEmail })
          });
          if (!res.ok) {
            throw new Error(`HTTP error! Status: ${res.status}`);
          }
          const response = await res.json();
          if (response.orderData && response.orderData.order_data) {
            setOrderData(response.orderData.order_data);
          } else {
            setOrderData([]);
          }
        } catch (error) {
          setOrderData([]);
        }
      };
      useEffect(() => {
        fetchMyOrder();
      }, []);
      let lastOrderDate = null;
      return (
        <div>
          <Navbar />
          <div className='container'>
            <div className='row'>
              {orderData === null ? (
                <div className="text-center mt-5 fs-3">Loading your orders...</div>
              ) : orderData.length > 0 ? (
                orderData.slice(0).reverse().map((orderGroup, groupIndex) => {
                  lastOrderDate = null;
                  return (
                    <React.Fragment key={groupIndex}>
                      {orderGroup.map((itemData, itemIndex) => {
                        if (itemData.Order_date) {
                          lastOrderDate = itemData.Order_date;
                          return (
```

```
                                <div key={`date-${groupIndex}-${itemIndex}`}
                                <h4 className='text-white'>{lastOrderDate}</h4>
                                <hr />
                              </div>
                            );
                          } else {
                            return (
                              <div key={`food-${groupIndex}-${itemIndex}`}
                                <div className="card mt-3" style={{ width: "16rem",
                                  <img src={itemData.img} alt={itemData.name} style={{ height: "120px", objectFit: "fill" }} />
                                  <div className="card-body">
                                    <h5 className="card-title">{itemData.name}</h5>
                                    <div className='container w-100 p-0' style={{ height: "38px" }}>
                                      <span className='m-1'>{itemData.qty}</span>
                                      <span className='m-1'>{itemData.size}</span>
                                      <div className='d-inline ms-2 h-100 w-20 fs-5'>
                                        Rs{itemData.price}/-
                                      </div>
                                    </div>
                                  </div>
                                </div>
                              </div>
                            );
                          }
                        })}
                      </React.Fragment>
                    );
                  })
                ) : (
                  <div className="text-center text-info mt-5 fs-3">No Orders Found. Start Feasting!</div>
                )}
              </div>
            </div>
            <Footer />
```

```
      </div>
    );
}
```

## Signup.js:

```
import React, { useState } from 'react';
import { Link } from 'react-router-dom';
export default function Signup() {
  const [credentials, setcredentials] = useState({
    name: "", email: "", password: "", geolocation: ""
  });
  const handleSubmit = async (e) => {
    e.preventDefault();
    const response = await fetch("http://localhost:5000/api/createuser", {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        name: credentials.name,
        email: credentials.email,
        password: credentials.password,
        location: credentials.geolocation
      })
    });
    const json = await response.json();
    console.log(json);
    if (!json.success) {
      alert("Enter valid credentials");
    }
  };
  const onChange = (event) => {
    setcredentials({ ...credentials, [event.target.name]: event.target.value });
  };
  return (
    <div className='container'>
      <form onSubmit={handleSubmit}>
        <div className="mb-3">
          <label htmlFor="Name" className="form-label">Name</label>
```

```
        <input type="text" className="form-control" name='name' value={credentials.name}
onChange={onChange} />
      </div>
      <div className="mb-3">
        <label htmlFor="exampleInputEmail1" className="form-label">Email
address</label>
        <input type="email" className="form-control" name='email'
value={credentials.email} onChange={onChange} id="exampleInputEmail1" aria-
describedby="emailHelp" />
        <div id="emailHelp" className="form-text">We'll never share your email with anyone
else.</div>
      </div>
      <div className="mb-3">
        <label htmlFor="exampleInputPassword1" className="form-label">Password</label>
        <input type="password" className="form-control" name='password'
value={credentials.password} onChange={onChange} id="exampleInputPassword1" />
      </div>
      <div className="mb-3">
        <label htmlFor="address" className="form-label">Address</label>
        <input type="text" className="form-control" name='geolocation'
value={credentials.geolocation} onChange={onChange} />
      </div>
      <button to type="submit" className="m-3 btn btn-info">Submit</button>
      <Link to="/login" className='m-3 btn btn-info'>Already a user</Link>
    </form>
  </div>
 );
}
```

# PART 5 (Backend)

## Models:

### Order.js

```
const mongoose = require('mongoose')
const { Schema } = mongoose;
const OrderSchema = new Schema({
    email: {
        type: String,
        required: true,
        unique: true
    },
    order_data: {
        type: Array,
        required: true,
    },
});
module.exports = mongoose.model('order', OrderSchema)
```

### User.js

```
const mongoose=require('mongoose')
const {Schema}=mongoose;
const UserSchema=new Schema({
    name:{
        type:String,
        required:true
    },
    location:{
        type:String,
        required:true
    },
     email:{
        type:String,
        required:true
    },
     password:{
        type:String,
        required:true
```

```
    },
    date:{
        type:Date,
        default:Date.now
    }
});
module.exports=mongoose.model('user',UserSchema)
```

## Routes:

### CreateUser.js:

```
const express = require('express');
const router = express.Router();
const User = require('../models/User');
const { body, validationResult } = require('express-validator');
const jwt = require("jsonwebtoken");
const bcrypt = require("bcryptjs");
const jwtSecret = "OurGroupNameIsDivAndConquer$#123";
router.post('/createuser',
  body('email').isEmail(),
  body('name', 'Name must be of 5 or more letters').isLength({ min: 5 }),
  body('password', 'Password Length is Short').isLength({ min: 5 }),
  async (req, res) => {
    const errors = validationResult(req);
    if (!errors.isEmpty()) {
      return res.status(400).json({ errors: errors.array() });
    }
    const salt = await bcrypt.genSalt(10);
    let secPassword = await bcrypt.hash(req.body.password, salt);
    try {
      await User.create({
        name: req.body.name,
        password: secPassword,
        email: req.body.email,
        location: req.body.location,
      });
      res.json({ success: true });
    } catch (error) {
      console.error('❌ Error creating user:', error.message);
```

```
      res.json({ success: false });
    }
  }
);
router.post('/loginuser',
  [
    body('email').isEmail(),
    body('password', 'Password must be at least 5 characters long').isLength({ min: 5 })
  ],
  async (req, res) => {
    const errors = validationResult(req);
    if (!errors.isEmpty()) {
      return res.status(400).json({errors: errors.array() });
    }
    let email= req.body.email;
    try {
      let userData = await User.findOne({ email });
      if (!userData) {
        return res.status(400).json({ errors: "Invalid email or password" })
      }
      const pwdCompare = await bcrypt.compare(req.body.password, userData.password)
      if (!pwdCompare) {
        return res.status(400).json({errors: "Invalid email or password" })
      }
      const data = {
        user: {
          id: userData.id
        }
      }
      const authToken = jwt.sign(data, jwtSecret)
      return res.json({ success: true, authToken:authToken })
    } catch (error) {
      console.log(error)
      res.json({ success: false});
    }
  }
)
module.exports = router;
```

## DisplayData.js

```
const express = require('express');
const router = express.Router();

router.post('/foodData', (req, res) => {
  try {
    res.send([global.food_items,global.foodCategory]);
  } catch (error) {
    console.error(error.message);
    res.status(500).send("Server Error");
  }
});

module.exports = router;
```

## OrderData.js

```
const express = require('express');
const router = express.Router();
const Order = require('../models/Orders');
router.post('/orderData', async (req, res) => {
    let data = req.body.order_data;
    const orderDate = req.body.order_date;
    const userEmail = req.body.email;
    await data.splice(0, 0, { Order_date: orderDate });
    try {
        let eId = await Order.findOne({ 'email': userEmail });
        if (eId === null) {
            await Order.create({
                email: userEmail,
                order_data: [data]
            });
            res.json({ success: true });
        } else {
            await Order.findOneAndUpdate(
                { email: userEmail },
                { $push: { order_data: data } }
            );
            res.json({ success: true });
        }
```

```
    } catch (error) {
        res.status(500).send("Server Error: " + error.message);
    }
});
router.post('/myorderData', async (req, res) => {
    try {
        let myData = await Order.findOne({ 'email': req.body.email });
        if (myData) {
            res.json({ orderData: myData });
        } else {
            res.json({ orderData: null });
        }
    } catch (error) {
        res.status(500).send("Server Error: " + error.message);
    }
});
module.exports = router;
```

## Conclusion

FeastFleet is designed to be a lightweight yet scalable food delivery solution. It combines realtime features with modern UI practices to offer an enhanced user experience. Its modular design allows easy integration of future enhancements like payment systems and admin dashboards.