DBMS LAB (CSL333) PROJECT

EATWISE RESTAURANT MANAGEMENT SYSTEM

Aravind A Kamath (CEC23CS041)
Arjun Manoj (CEC23CS043)
Anandhakrishnan S (CEC23CS033)
Aravind Sajeev (CEC23CS042)

Computer Science & Engineering (CSE)
Semester 5 (S5)
College of Engineering Cherthala

INDEX

INTRODUCTION

- **EatWise is a small restaurant management application to demonstrate relational database design, CRUD operations, and use of triggers to maintain derived/calculated columns**

- **Objective: to simplify the process of restaurant management (order/suborder/review/menu/customer handling) and improve its efficiency**

- **Tools and technologies used => ReactJS (HTML, CSS, JavaScript, JSX), ExpressJS (NodeJS), MySQL**

- **Purpose/motivation: to alleviate hassles like mixing up of orders, incorrect cash calculations, outdated menu items and other such issues in restaurants**

IMPLEMENTATION

- **A simple client-server restaurant management app: ReactJS frontend communicating with an ExpressJS backend using axios/CORS to manage menu, orders, users, reviews with CRUD flows and role-based homepages**

MODULES

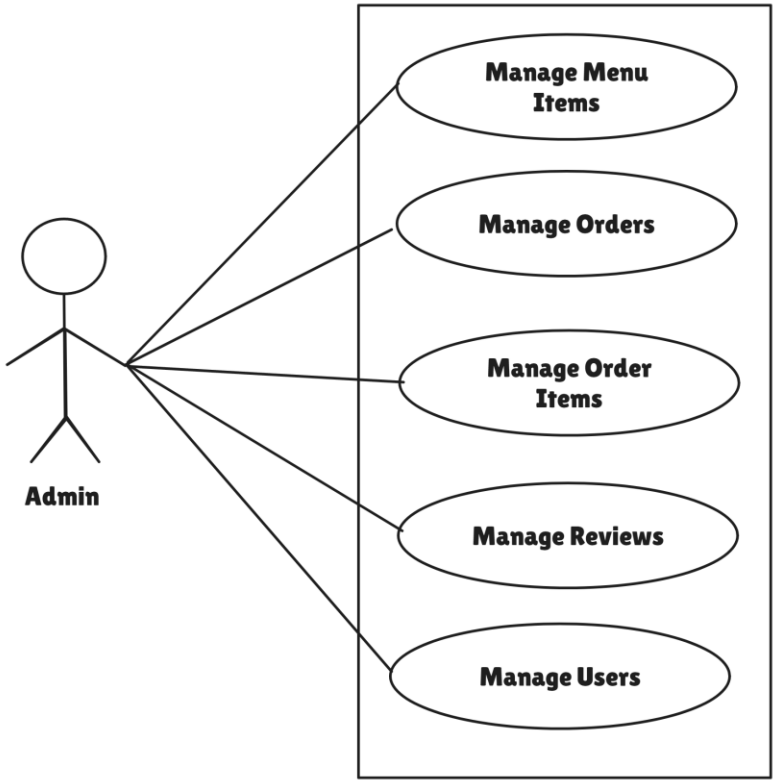
1. **Frontend -> App, Homepage, Hero, Login, Review, Credits, AdminLanding, CustomerLanding**
2. **Backend -> index.js (single backend file)**

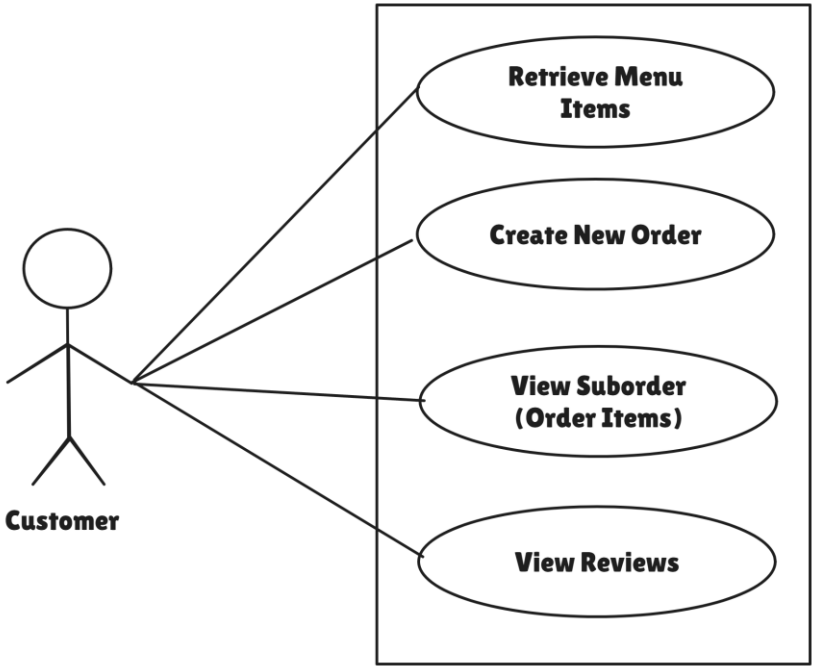
- **User roles – Customer and Admin**

- `users` (user_id PK, username, password, email, role)
- `menu_items` (item_id PK, item_name, category, price, item_picture)
- `orders` (order_id PK, user_id FK -> users.user_id, order_date_time, total_amount)
- `order_items` (order_items_id PK, order_id FK -> orders.order_id, item_id FK -> menu_items.item_id, quantity, suborder_price)
- `reviews` (review_id PK, user_id FK -> users.user_id, description, date, rating)

USE CASE DIAGRAM

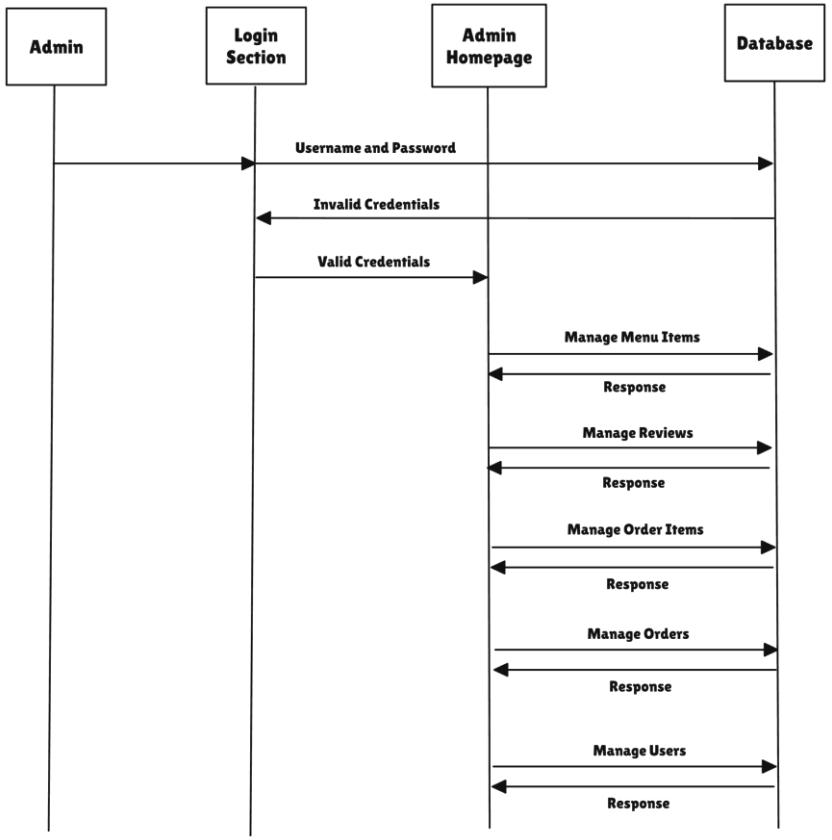
- Customer => view menu_items, create suborders
- Admin => manage all 5 tables with create/retrieve/update/delete (CRUD)

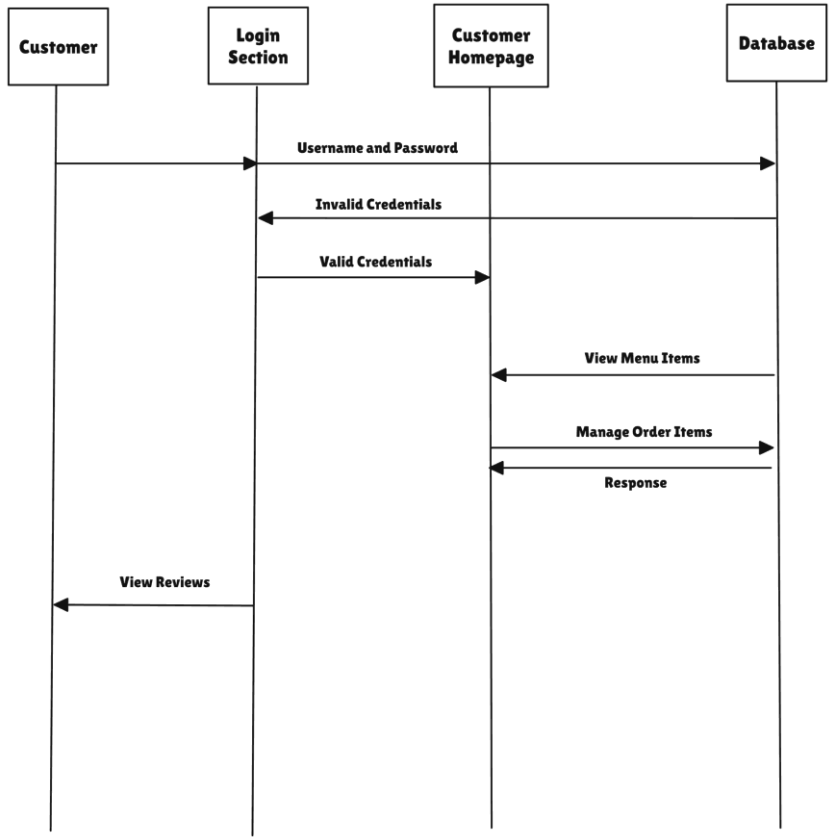




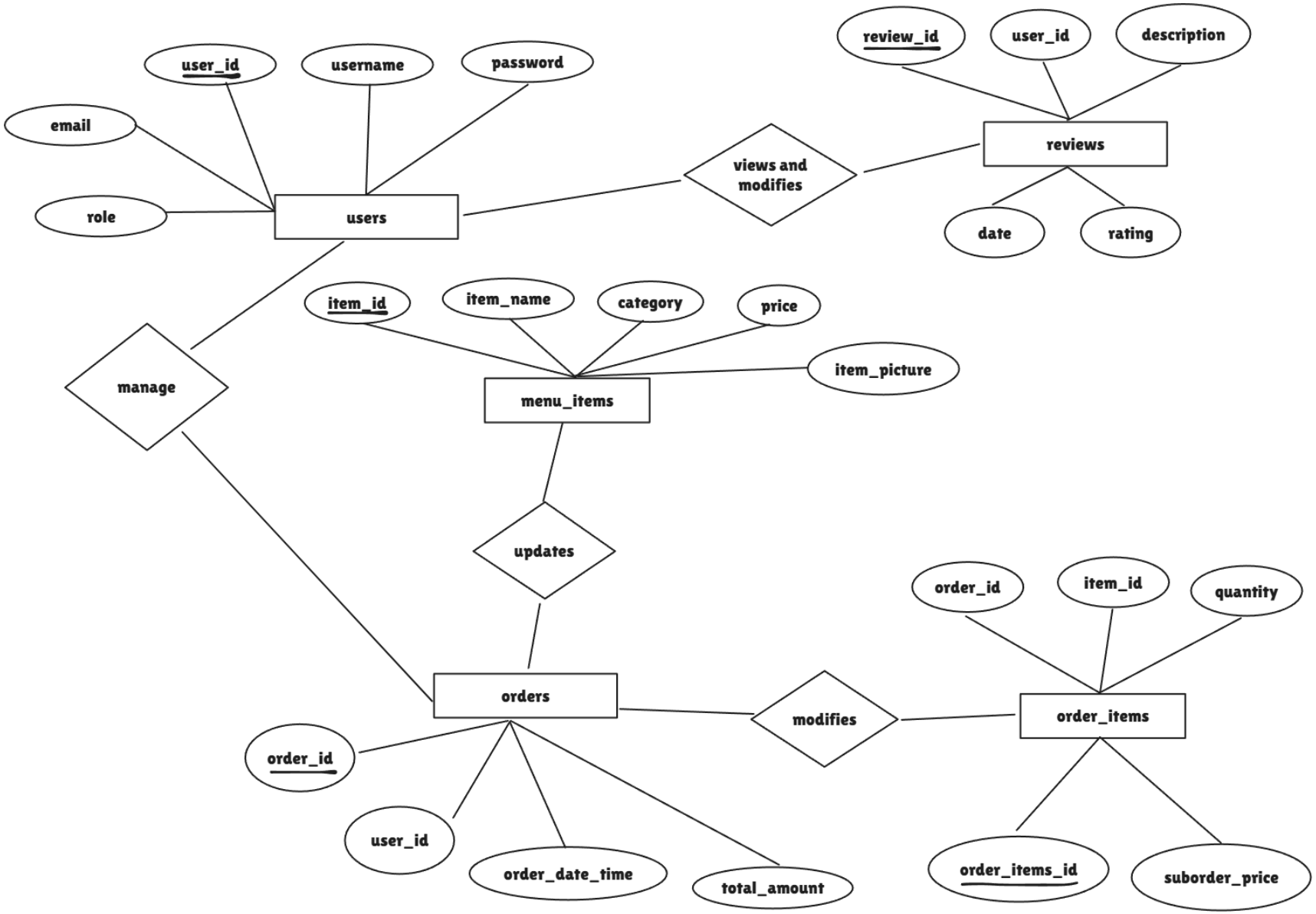
SEQUENCE DIAGRAM

- Customer => view menu_items, create suborders
- Admin => manage all 5 tables with create/retrieve/update/delete (CRUD)





ER DIAGRAM


user_id
username
password
email
role
users
review_id
user_id
description
reviews
date
rating
views and modifies
item_id
item_name
category
price
item_picture
menu_items
manage
updates
orders
order_id
user_id
order_date_time
total_amount
modifies
order_id
item_id
quantity
order_items
order_items_id
suborder_price

CODE

index.js (backend)

```javascript
// Login route
app.post('/login', (req, res) => {
    const { username, password } = req.body;
    const q = "SELECT * FROM users WHERE username = ? AND password = ?";
    db.query(q, [username, password], (err, results) => {
        if (err) return res.status(500).json({ error: 'Database error' });
        if (results.length === 0) {
            return res.status(401).json({ error: 'Invalid credentials' });
        }
        // Return role and user info
        const user = results[0];
        return res.json({ role: user.role, user_id: user.user_id, username: user.username });
    });
});
```

CODE

index.js (backend)

- **Creation of table menu_items**

```sql
CREATE TABLE `menu_items` (
  `item_id` int NOT NULL,
  `item_name` varchar(45) DEFAULT NULL,
  `category` varchar(45) DEFAULT NULL,
  `price` int DEFAULT NULL,
  `item_picture` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`item_id`),
  UNIQUE KEY `item_id_UNIQUE` (`item_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

CODE

index.js (backend)

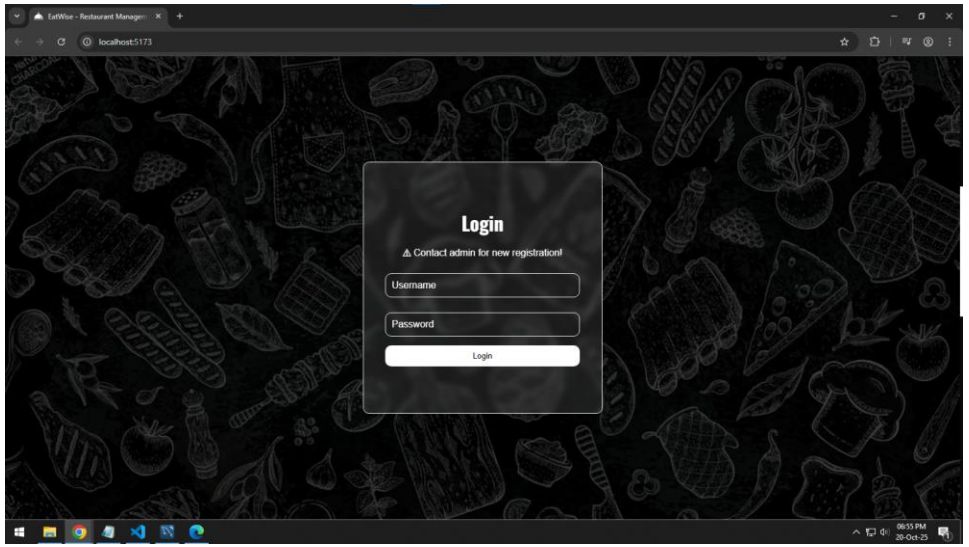
- **Updation of menu_items**

```javascript
// Update menu item by id
app.put('/menu_items/:id', (req, res) => {
    const menuItemId = req.params.id;
    const q = "UPDATE menu_items SET `item_name` = ?, `category` = ?, `price` = ?, `item_picture` = ? WHERE item_id = ?";
    const values = [
        req.body.item_name,
        req.body.category,
        req.body.price,
        req.body.item_picture
    ];
    db.query(q, [...values, menuItemId], (err, data) => {
        if (err) return res.json(err);
        return res.json("Menu item updated successfully!");
    });
})
```

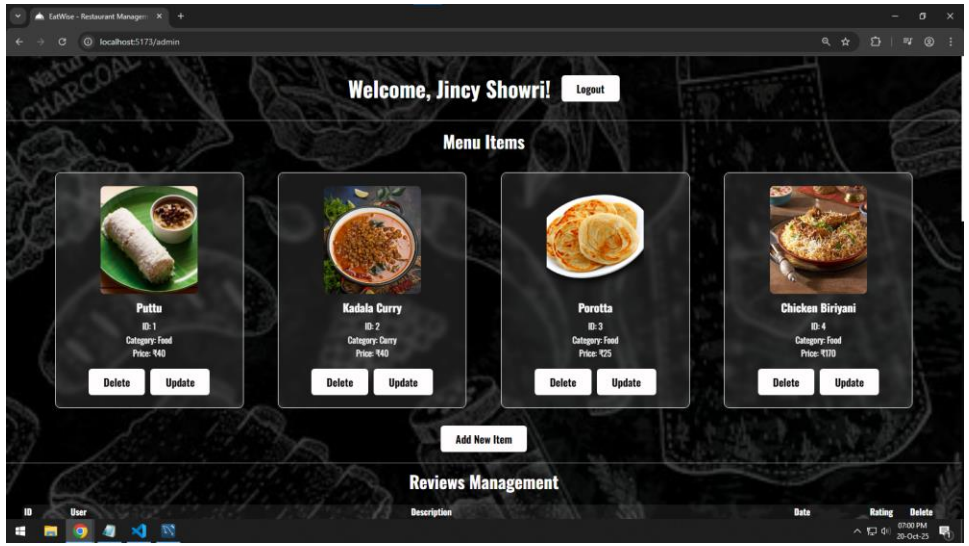
RESULTS

- **<u>Homepage</u> – simple layout with CSS flexbox and relative/absolute positioning**

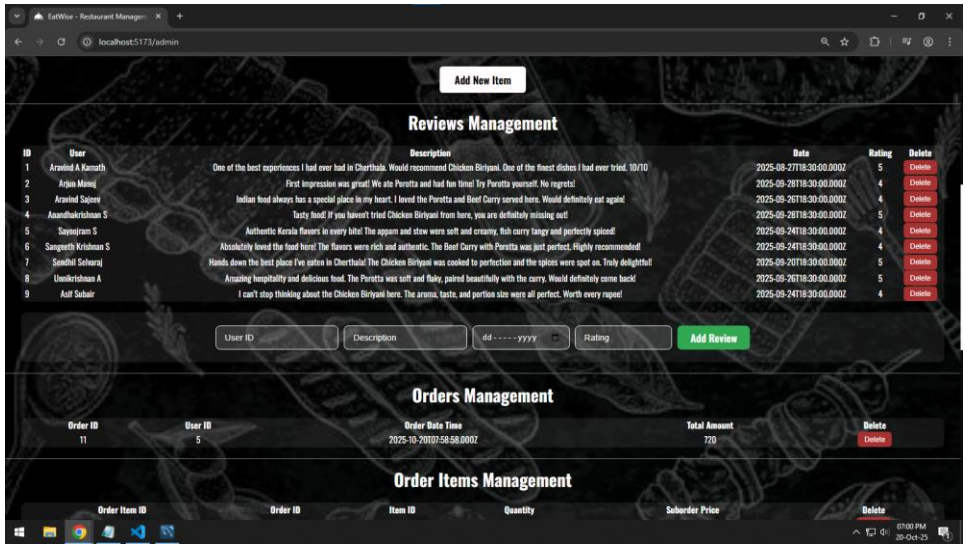
- **<u>Login section</u> with glassmorphism UI design with simple inputs and button**



ADMIN HOMEPAGE

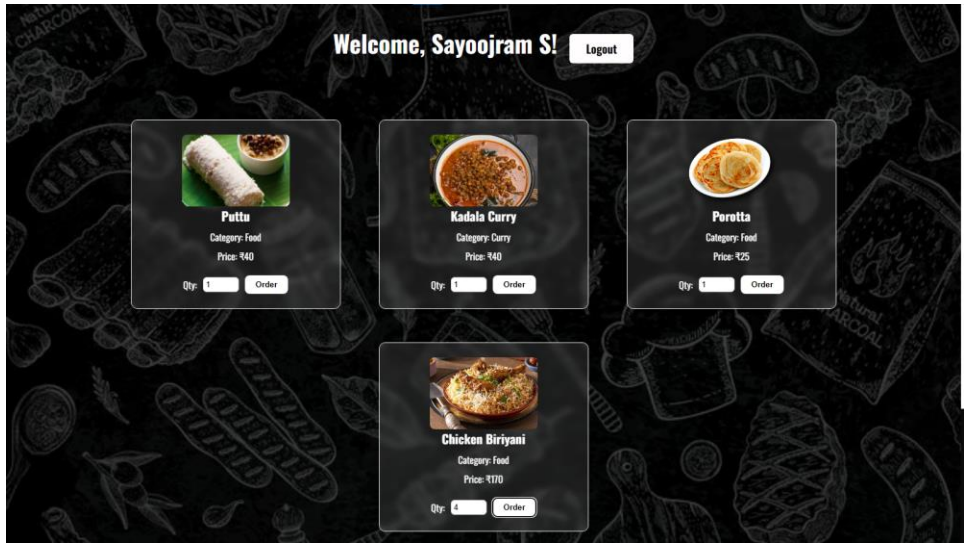


- **Ability to modify all 5 tables' data (menu_items section shown in screenshot)**



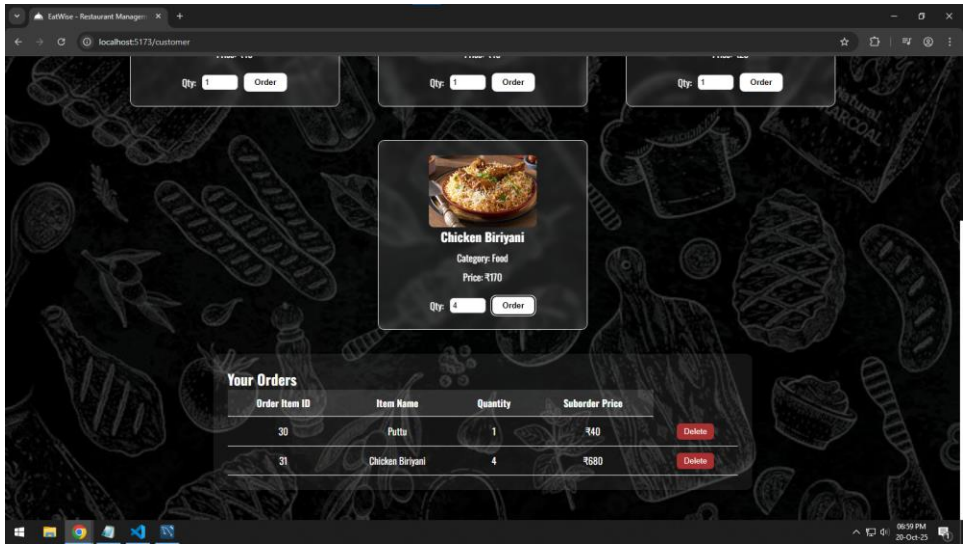
- **Ability to manage different tables including reviews, orders etc...**

CUSTOMER HOMEPAGE



- **Ability to view menu_items, select quantity and order**

- **Ability to view and delete suborders**



CONCLUSION

- **A basic but diversely feature-rich web app with vast technologies used for management of 5 tables with data consistency and objective to improve customer/admin experience in restaurant management system**

- **Technologies learned – ReactJS, React Router, React DOM, NodeJS, ExpressJS, MySQL, CORS, Nodemon, SQL Triggers**

- **Possible improvements – adding proper authentication with security, adding more options for table modifications**