



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Report File FULL STACK

Student Name: Palash Mathur

UID: 23BAI70673

Branch: BE-AIT-CSE

Section/Group: 23AIT-KRG-G2

Semester: 5th

Subject Code: 23CSP-339

---

### Project Report: Restaurant Website

#### 1. AIM

The aim of this project is to build a simple Responsive Restaurant Website where users can:

- Check their favorite restaurants and dishes
- They can place their orders.
- Stores user's details in secured database

The project helps beginners learn frontend (HTML, CSS, JS) and backend (Node.js with Express.js) basics, along with database usage.

---

#### 2. TECH STACK

##### Frontend

- HTML → Page structure.
- CSS → Styling and layout.
- JavaScript → Interactivity and fetching data from backend.

##### Backend

- Node.js → JavaScript runtime for backend.
- Express.js → Framework to handle routes and APIs. Database
- JSON file / MongoDB o Use a local JSON file to store posts.

##### Tools

- VS Code → Code editor.
- Browser → To run the application.

### 3. THEORY / BACKGROUND

The Website follows the client-server model:

- Client (HTML, CSS, JS) → Can login and search for their foods/restaurants
- Server (Node + Express) → Handles requests (create, read, update, delete user info).
- Database (JSON/MongoDB) → Stores User login details and history data.

We will design REST APIs:

- GET /posts → Fetch all details.
- POST /posts → Create a new user.
- PUT /posts/:id → Reset Password for user
- DELETE /posts/:id → Delete a user account.

---

### 4. PROCEDURE / IMPLEMENTATION

#### **Step 1: Setup Environment**

- Install Node.js and MySQL/SQL Express.
- Create a project folder `restaurant-website`.
- Initialize with `npm init -y`.
- Install dependencies:

```
npm install express mysql body-parser cors bcrypt jsonwebtoken
```

#### **Step 2: Database Setup**

- Create a database `restaurantDB`.
- Create tables:
  - `Users(user_id, name, email, password, address)`
  - `Menu(item_id, name, price, restaurant)`
  - `Orders(order_id, user_id, item_id, status, date)`

#### **Step 3: Backend Development**

- Build `server.js` to define routes and connect with MySQL.
- Implement authentication (JWT, bcrypt for hashing passwords).
- Implement APIs for users, menu, and orders.

#### **Step 4: Frontend Development**

- Create `index.html`, `style.css`, and `script.js`.
- Add forms for login/signup and dynamic menus using JavaScript fetch API.
- Handle order placement and order tracking.

#### **Step 5: Run Project**

- Start the backend with:

```
node server.js
```

- Open `index.html` in the browser to interact with the system.
- 

## 5. CONCLUSION

This project demonstrates how to build a full-stack restaurant website that:

- Allows users to create accounts, log in, and manage their profiles.
- Stores all data securely in SQL-based databases.
- Lets users browse menus, place orders, and track them.
- Combines frontend (HTML, CSS, JS) with backend (Node.js, Express) and database (MySQL/SQL Express) for practical web application development.

It builds a strong foundation in web technologies, authentication systems, and database integration, preparing developers for more complex projects in the future.