

Lab 1 Assignment: Using Node.js and ReactJS

Due: March 3rd, 2025. 11:59pm

This lab assignment covers developing REST services using Node.js (Express.js) and ReactJS. It is worth 30 points.

Prerequisites:

- You should be able to run basic Node.js and React applications.
- You must be familiar with JavaScript programming.

UberEATS Prototype Requirements:

Develop a Prototype of UberEATS using React for the frontend and Node.js (Express.js) for the backend. The application must support two main personas:

- Customer
- Restaurant

Required Features

Customer Features:

- Signup – Customer sign-up with name, email ID, and password (store securely using bcrypt.js)
- Login/Logout – Implement basic session-based authentication using Express-session.
- Profile Page - Display customer details, Update profile picture, Update profile information, Country field should be a dropdown list, state should be abbreviated
- Restaurant Page/ Dashboard- View restaurant details, including a brief description and menu. Select a dish to add to the cart. Finalize the order in the cart view.
- Favourites - Mark restaurants as favorites, Display a favourites tab

Restaurant Features:

- Signup – Restaurant sign-up with name, email ID, password, and location.
- Login/Logout – Implement session-based authentication.
- Profile Management -
View and update restaurant profile (name, location, description, contact info, images, timings).
Add or edit dishes with details (dish name, ingredients, image, price, description, category – Appetizer, Salad, Main Course, etc.).

View a list of added dishes.

- **Orders Management-**

View and filter customer orders by status: New, Delivered, Cancelled.

Update order delivery status: Order Received, Preparing, On the Way, Pick-up Ready, Delivered, Picked Up.

View customer profiles for each order.

Backend Development Requirements (Node.js + Express.js + MySQL)

- Node.js with Express.js will be used for the backend
- Database Options: MySQL
- Implement RESTful APIs for:
 - User authentication (session-based login/signup).
 - Restaurant and dish management.
 - Order placement and tracking.
 - Customer profile and favorite restaurants.
- Secure API endpoints with basic validation and error handling
- Handle errors with proper exception handling

Frontend Development Requirements (React)

- The frontend should be fully responsive, using CSS frameworks like Bootstrap or TailwindCSS.
- API calls should be managed using Axios or Fetch API.

API Documentation

You are required to document your API endpoints using Swagger or Postman.

1. Swagger- Use Swagger to generate API documentation. The Swagger UI should allow for testing API routes.
2. Postman Collection- Alternatively, export a Postman collection with descriptions for each endpoint, request parameters, headers, and sample responses. Submit the Postman collection along with your project.

Non-Functional Requirements

- Responsiveness: The web app must work across mobile, tablet, and desktop
- Accessibility: Implement semantic HTML, alt text, and keyboard navigation support
- Scalability: Optimize queries, avoid unnecessary data loading, and ensure efficient API response times.

Git Repository Guidelines

- Commit History – Add detailed commit messages describing changes.
- Dependencies – Do not submit node_modules. Instead, include package.json.
- README.md – Instructions to run the application.
- Private Repository – Invite leon.correia@sjsu.edu

Project Report Guidelines

Submit a brief report including:

- Introduction – Purpose and goals of the system
- System Design – Explain architecture (Node.js, Express, MongoDB/MySQL, React).
- Results – Screenshots of key screens and API test results.

Submission Guidelines:

Upload your project report (John_Lab1_Report.doc) on Canvas before the deadline.