Lab 1 Assignment: Using Node.js and ReactJS + AgenticAI

Due: October 20, 2025, 11:59 PM

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

Prerequisites

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

Airbnb Prototype Requirements

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

- Traveler
- Owner (Property Host)

Required Features

Traveler Features:

- Signup – Traveler signs 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, Upload profile picture, Update profile information (name, email, phone number, about me, city, country, languages, gender). Country field should be a dropdown list, state should be abbreviated
- Property Search Page/Dashboard – Search available properties by:
    - Location
    - Start Date, End Date
    - Number of Guests
- Property Details View – View details such as property name, type, amenities, pricing, bedrooms, bathrooms, and availability. Allow booking.

- Booking Flow – Create a booking request for selected dates/guests. A booking starts in Pending state until the Owner responds. Travelers can view Pending, Accepted, and Cancelled bookings.
- Favourites - Mark properties as favorites, Display a favourites tab
- Traveler History – Display a history tab for any previous bookings/trips taken.

- Agent AI: the agent's button should be accessible to the user, from the dashboard

Owner (Host) Features:

- Signup – Owner signs up with name, email ID, password, and location.
- Login/Logout – Implement session-based authentication.
- Profile Management – View/update profile details(name, location, contact info, images). Add/edit with details (property name, type, amenities, pricing, bedrooms, bathrooms, and availability.)
- Property Posting – Post property for rent with details:
    - Location, description, photos
    - Pricing, amenities, availability
- Booking Management – View incoming booking requests for your properties and either Accept or Cancel each request.
    - Accepting a booking changes its status to Accepted and should block the property for the requested dates.
    - Cancelling a booking changes its status to Cancelled and should release the dates.
- Owner Dashboard – Show previous bookings and recent requests.

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)
    - Traveler profile management

    - Property posting and management
    - Property search and booking
    - Traveler and Owner dashboards
- Bookings
    - Create booking (Traveler) → status PENDING
    - List bookings (Traveler & Owner)
    - Accept booking (Owner) → status ACCEPTED
    - Cancel booking (Owner or Traveler) → status CANCELLED
- 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.

Agent features(Python FastAPI + Langchain+MYSQL):

Given a booking (dates, location, party type), traveler preferences (budget, interests, mobility needs, dietary filters), and live/local context (weather, POIs, events), the agent produces:

- A day-by-day plan (morning/afternoon/evening blocks)
- Activity cards (title, address/geo, price tier, duration, tags, wheelchair/child-friendly flags)
- Restaurant recs filtered by dietary needs
- Packing checklist (weather-aware)

It supports NLU: users may submit a free-text ask (e.g., “we’re in {users booking should be passed in context} vegan, no long hikes, two kids”).

Use this services to do web searches https://www.tavily.com/#features

Inputs

- Booking context: dates, location, party type
- Preferences: budget, interests, mobility needs
- Local Point of Interest catalog & events (local events)

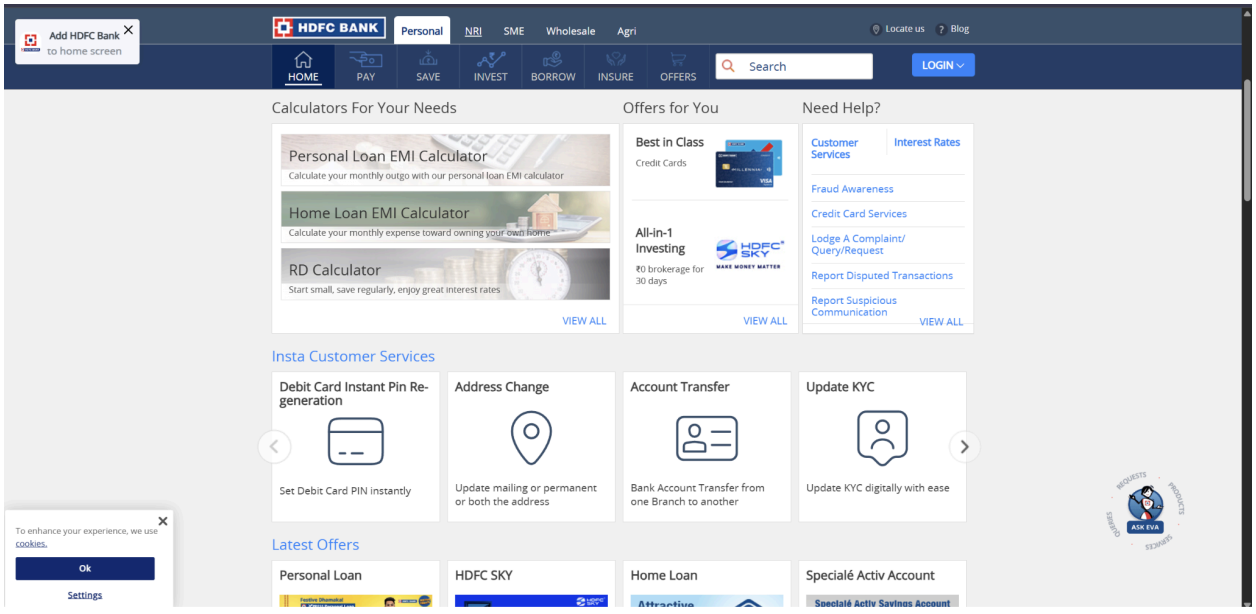
Outputs

- Day-by-day plan, activity cards, restaurant recs, packing checklist

User Interface (UI):

- In your Dashboard the agent service should show up as a button on the bottom right corner. Upon clicking the button, the layout should open on the right hand side of the screen. Feel free to add FAQ buttons(optional), your agentic service should be usable from UI.
- Create the AI service in python as a REST endpoint:
    - POST request to AI Concierge Agent -> returns suggestions in JSON

Example:



visit : https://www.hdfcbank.com/ to see the structure

API Documentation:

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

- Swagger- Use Swagger to generate API documentation. The Swagger UI should allow for testing API routes.
- 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: Application must work on mobile, tablet, and desktop.
- Accessibility: Implement semantic HTML, alt text, and keyboard navigation support.
- Scalability: Optimize queries, avoid unnecessary data loading, 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 tanyayadavv5@gmail.com and smitsaurabh20@gmail.com

Report Guidelines:

Submit a brief report including:

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

Submission Guidelines:

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