FS Developer Exam

Build a sales dashboard.

To build a sales dashboard with a split design and API, you'll need to follow a structured approach. Below, I outline the steps and components required to fulfill the given requirements:

1. Design:

The design part involves creating the user interfaces (UI) for the sales dashboard. This includes:

- Login Page: Allows sales users to log in.
- My Business Page:
  - List of Companies: Displaying existing companies with options to edit or delete.
  - Add New Company Form: Allows sales users to add new company data.
- Meetings Page:
  - List of All Meetings: Displaying a history of meetings with companies.
  - Add New Meeting Summary Form: Allows users to record new meetings, including date, location, business name, and summary.
- Meeting Statistics Page:
  - Graphs showing:
    - Number of meetings per day.
    - Number of meetings for the existing month.
    - Percentage of the number of meetings per day.

2. API:

The API handles database connections and logic. It should include endpoints for various functionalities such as:

- User Authentication: Handles user login.
- Company Management:
  - CRUD operations for companies.
- Meeting Management:
  - CRUD operations for meetings.
- Statistics:
  - Endpoints to retrieve data for meeting statistics.

3. Database:

You can use SQLite or any other database you're familiar with.

The database should contain tables for:

- Sales Users: Stores user credentials.
- Company Business: Stores information about companies.
- Meetings: Stores meeting details, including date, location, business name, and summary.

4. Admin Interface:

The admin interface is a part of the design but deserves special mention. It should provide a user-friendly environment for sales users to interact with the dashboard. This includes:

- Login functionality.
- Navigation links to different sections/pages (My Business, Meetings, Meeting Statistics).
- Forms for adding/editing companies and meetings.
- Visual representations of meeting statistics (graphs).

Technologies Recommendations: you can use any technologies you familiar.

- Frontend: HTML/CSS, JavaScript, a frontend framework/library like React.js or Vue.js.
- Backend: Node.js or Python with a framework like Express.js or Flask for the API.
- Database: SQLite or any other relational database management system (RDBMS) you're comfortable with.
- Additional Libraries/Tools: Charting libraries like Chart.js or D3.js for graphical representations.

Workflow:

1. Design UI mockups.
2. Set up backend API and database.
3. Implement frontend and backend functionalities.
4. Document explanation how to setup the project in “Stage ENV”