Task: Full Stack Development Assessment

Gym Management System Development

Figma Link- *https://www.figma.com/design/CYvDqqzOQUaUvFcCSC8fZH/Untitled?node-id=0-1&t=GQryrggKqDbqtu0N-1*

Objective

Develop a fully functional Gym Management System based on the provided Figma project using React (frontend), Node.js/Express (backend), and Firebase (database and authentication).

Frontend Requirements

General Points

- **Framework**: Use React.
- **Styling**: Incorporate Tailwind CSS and pure CSS where necessary for responsive and modern UI.
- **Structure**: Break the frontend into reusable JSX components.
- **API Integration**: Call backend API routes for CRUD operations.
- **Responsiveness**: Ensure all pages are mobile-friendly and adapt seamlessly to different screen sizes.

1. Login and Signup

- **Features**:
  - Create two forms:
    - Login: Email and Password fields with "Show/Hide Password" functionality and error handling.
    - Signup: Name, Email, Password, and Confirm Password fields with validation.
  - Use Firebase Authentication for user login and signup.
  - Display validation errors for incorrect credentials.
  - Redirect users to the Dashboard upon successful login.
- **Styling**: Clean, modern UI with fitness-themed colors and a branded logo.

2. Dashboard

- **Features**:
  - Display widgets/cards showing:
    - Total Members
    - Total Trainers
    - Active Classes

        - Monthly Revenue
    - Quick action buttons for:
        - Add Member
        - Add Trainer
        - Add Class
    - Navigation Sidebar with links to all major pages.
- **Styling**: Use a vibrant and professional theme. Include hover effects and transitions for navigation.

3. Package Creation and Management

- **Features**:
    - Form to create packages with:
        - Name
        - Price
        - Duration (e.g., Monthly, Yearly)
        - Description
    - Display a table of all packages with Edit/Delete options.
    - Implement pagination for the table.
- **API Calls**: Integrate backend routes to:
    - Create, fetch, update, and delete packages.
- **Styling**: Table and form design with Tailwind's utility classes for responsiveness.

4. Client Management

- **Features**:
    - Table to display all clients with:
        - Name
        - Membership Type
        - Status (Active/Inactive)
        - Join Date
        - Action buttons (Edit/View/Delete).
    - Add a "Search Clients" bar and filters for membership type/status.
    - "Add Client" form to register new members.
- **API Calls**: Integrate backend routes to:
    - Create, fetch, update, and delete clients.
- **Styling**: Use Tailwind CSS for modern table design and search functionality.

5. Trainers Creation and Management

- **Features**:
    - Form to add trainers with:
        - Name
        - Specialty
        - Assigned Classes
        - Contact Information
    - Table displaying all trainers with Edit/Delete options.

        - Calendar view for trainer schedules (optional for extra credit).
    - **API Calls**: Integrate backend routes to:
        - Create, fetch, update, and delete trainers.
    - **Styling**: Focus on clean forms and structured table views.

Backend Requirements

General Points

- **Framework**: Use Node.js with Express.js.
- **Database**: Utilize Firebase Firestore for data storage.
- **Authentication**: Use Firebase Authentication for managing user logins and roles.
- **Structure**:
    - **Controllers**: Handle business logic.
    - **Routes**: Define API endpoints.
    - **Middleware**: Add authentication checks for protected routes.
    - **Server**: Manage the app entry point and setup.

1. Login and Signup

- **Endpoints**:
    - **/auth/signup**: Handles user registration, validates input, and stores user data in Firestore.
    - **/auth/login**: Verifies credentials using Firebase Authentication and returns a JWT token.
- **Middleware**: Add token validation to secure routes

2. Dashboard

- **Endpoints**:
    - **/stats/members**: Fetch total members count.
    - **/stats/trainers**: Fetch total trainers count.
    - **/stats/classes**: Fetch active classes count.
    - **/stats/revenue**: Fetch monthly revenue details.
- **Logic**: Implement Firestore queries to gather statistical data.

3. Package Creation and Management

- **Endpoints**:
    - **/packages**: Create a new package (POST).
    - **/packages**: Fetch all packages (GET).
    - **/packages/:id**: Update a package by ID (PUT).
    - **/packages/:id**: Delete a package by ID (DELETE).
- **Validation**: Ensure all fields are validated before storing in Firestore.

4. **Client Management**
    - **Endpoints**:
        - **/clients**: Add a new client (POST).
        - **/clients**: Fetch all clients (GET).
        - **/clients/:id**: Update client information by ID (PUT).
        - **/clients/:id**: Delete a client by ID (DELETE).
    - **Search/Filter**:
        - /**clients/search**: Endpoint to search and filter clients based on parameters.

5. **Trainers Creation and Management**
    - **Endpoints**:
        - **/trainers**: Add a new trainer (POST).
        - **/trainers**: Fetch all trainers (GET).
        - **/trainers/:id**: Update trainer details by ID (PUT).
        - **/trainers/:id**: Delete a trainer by ID (DELETE).
    - **Schedule**:
        - /trainers/schedule: Endpoint to assign classes and fetch trainer schedules.

Assignment Deadline and Delivery Method

Deadline: January **08**, 2025, **11:59 PM**

Delivery Method:

1. **GitHub Repository**
    - Create a public GitHub repo and upload all project files.
    - Include the **assignment PDF** in the repository root.
    - Share the GitHub repo link.
2. **Assignment PDF**
    - Include:
        - Your **basic information**: Name, Email, Phone, GitHub profile link.
        - **GitHub repo link**.
        - **Screen recording link** (uploaded to Google Drive).
        - Confirm **Firebase collaboration** with **dynac.technologies@gmail.com**.

3. **Screen Recording**
    - Record and upload a video of your project setup and demonstration.
    - Showcase CRUD operations, Firebase integration, and Tailwind CSS UI.
    - Share the **Google Drive link** in the PDF.
4. **Firebase Collaboration**
    - Add **dynac.technologies@gmail.com** as a collaborator with access rights.

Email the completed **assignment PDF to careers@dynac.lk*