

# REEF

## Home Assignment for Fullstack React.js and Node.js Developer

### Objective:

Develop a simplified business management application that allows business owners to manage products, view sales reports, and handle customer orders. The app should demonstrate proficiency in both frontend and backend development using React.js and Node.js. The application must include a secure login system, a RESTful API, and a well-structured database.

---

### Requirements:

#### 1. App Structure and Navigation:

- Utilize modern React.js practices, including functional components and hooks.
  - Implement a responsive, user-friendly UI using any CSS framework.
  - Ensure seamless navigation across different sections of the app.
- 

#### 2. Features:

- **User Authentication:**
    - Implement a secure login and registration system using Node.js with JWT (JSON Web Tokens).
    - Ensure proper session management and token expiration handling.
  - **Product Management:**
    - Add features for creating, editing, and deleting products.
    - Each product should have a name, description, price, and image.
  - **Order Management:**
    - Display a list of customer orders including details such as order ID, customer name, order date, and status.
    - Enable updating the status of customer orders.
  - **API Integration:**
    - Design and implement a RESTful API using Node.js to manage backend functionality.
    - Ensure seamless integration between the frontend and backend.
-

### 3. Technical Requirements:

- **Frontend:**
    - Use React.js for building the UI, and manage state using Redux or Context API.
    - Implement form validation and error handling in the product and order management forms.
  - **Backend:**
    - Use Node.js for building the backend API.
    - Implement authentication and authorization for routes.
    - Utilize a database such as PostgreSQL to store user, product, and order data.
  - **Clean Code:**
    - Follow best practices for code organization with separation of concerns.
    - Write modular, reusable, and testable code.
  - **Performance:**
    - Ensure the app is performant and efficiently handles API requests.
  - **Testing:**
    - Write unit tests for critical sections of both the frontend and backend.
- 

### 4. Additional Features (Bonus Points):

- **Sales Reports:**
    - Generate sales reports based on customer orders, showing total sales, number of orders, and average order value.
    - Include filtering options for reports by date range.
  - **Search Functionality:**
    - Implement a search feature to filter products or orders.
  - **Notifications:**
    - Send notifications when a new order is placed or when the status of an order is updated.
  - **Admin Dashboard:**
    - Create an admin dashboard with statistics and charts showing key metrics.
  - **GraphQL:**
    - Implement a GraphQL API as an alternative to the RESTful API for more efficient data querying.
- 

### 5. Submission:

- We estimate this task should take approximately 4-6 hours to complete. We prefer fewer features that are well-implemented and written over trying to complete all features with lesser quality or taking too long to finish.
- Provide a GitHub repository link with the complete source code.

- Include a README file with setup and run instructions.
- 

### **Inspiration:**

Feel free to draw inspiration from applications like Uber Eats, but ensure your solution is unique and showcases your coding style and problem-solving abilities.

---

Good luck! We are excited to see your innovative solutions!