**Technical Assessment**

**Case Study: Software Development Assessment**

**Background:**

A leading tech company is seeking a skilled full-stack developer to assess their proficiency in building web applications using React.js, Next.js, Node.js, and PostgreSQL, and incorporating key concepts such as API integration, middleware, database interaction, and authentication.

**Objective:**

The candidate is tasked with creating a comprehensive web application with both front-end and back-end components. The application should showcase the ability to build interactive user interfaces, handle routing, integrate with APIs, interact with databases, and implement user authentication.

**Note:** Candidates can develop a basic-level application capable of performing specific functions, utilizing a local host for their server.

**Tasks:**

**Frontend (React.js and Next.js):**

**1. React.js Basics:**

**Objective:** Create a simple React component that displays a list of items (to-do list) and allows users to add new items.

**Approach:** Utilize state and props appropriately to manage the dynamic nature of the to-do list.

**2. Next.js Routing:**

**Objective:** Create a Next.js application with at least two pages and implement client-side navigation between them.

**Approach:** Leverage the capabilities of Next.js for efficient client-side routing.

**3. API Integration:**

**Objective:** Integrate a mock API endpoint with the Next.js app using tools like JSON server **Approach:** Fetch data from the API and display it on a page, demonstrating proficiency in frontend backend interaction.

**Backend (Node.js):**

**4. Node.js Server:**

**Objective:** Set up a basic Node.js server using Express.js and create an API endpoint

that returns a JSON response.

**Approach:** Establish a solid foundation for the backend, ensuring proper routing and response handling.

**5. Middleware:**

**Objective:** Implement a middleware function that logs information about incoming requests.

**Approach:** Create a middleware to enhance request logging, facilitating debugging and monitoring.

**6. Database Interaction:**

**Objective:** Connect the Node.js server to a PostgreSQL database, create a table for storing user information, and implement CRUD operations.

**Approach:** Demonstrate competence in database integration, schema design, and CRUD operations.

**Database (PostgreSQL):**

**7. Database Schema:**

**Objective:** Design a simple relational database schema for a blogging platform with tables for users, posts, and comments.

**Approach:** Develop a coherent schema considering relationships between entities.

**8. SQL Queries:**

**Objective:** Write SQL queries to retrieve posts by a specific user, and count comments on a post.

**Approach:** Showcase SQL proficiency in accessing and manipulating data within the designed schema.

**Additional Challenge: Authentication:**

**9. Authentication:**

**Objective:** Implement user authentication in the Node.js application using Passport.js. **Approach:** Ensure that only authenticated users can access specific API endpoints, demonstrating knowledge of secure authentication practices.

**Submission Instructions:**

• The candidate should organize code in a well-structured repository.

• Provide clear documentation and comments explaining thought processes and design decisions for each task.

• Write a summary of the approach taken for each task, highlighting key considerations and challenges faced.

**Assessment Criteria:**

The candidate will be evaluated based on the quality and completeness of the code and tasks, adherence to best practices, effective use of relevant technologies, documentation clarity, and the ability to address the specified requirements in each task.