Software Development Intern

Assessment Title

Role-based System Design and Implementation

Objective

To evaluate the intern's understanding of role-based access control, basic CRUD operations, and system design principles to implement a platform with three user types: Partner, Client, and Super Admin.

Please find the following design to assist the implementation UI Design. Please try to be strict with the given design to complete your task.

Scenario Overview

The system allows:

1. Partners to register and associate themselves with a company, providing services.
2. Clients to book services provided by the partners.
3. Super Admins to manage all users, companies, and bookings with full access.

Technology Stack

You are recommended to use **React** for the frontend, **Spring Boot** for the backend, and **MongoDB** as the database. However, you are free to choose any other technology stack you are comfortable with(AngularJS/ NodeJS, Vue JS, C#).

Assessment Tasks

Task 1: User Authentication and Role Assignment

- Implement a simple login and registration system with role-based access (Partner, Client, Super Admin).
- Store user information, including their roles, securely (e.g., use password hashing).
- Test that users can log in only to their respective dashboards.

Task 2: Partner Dashboard

- Create a dashboard where partners can:
    - Register a company (with details like company name, address, and contact info).
    - View and update their profile.

    - List the services they offer (e.g., service name, description, price).

Task 3: Client Dashboard

- Create a dashboard where clients can:
    - Browse available services offered by partners.
    - Book a service (provide booking details like date, time, and service selection).
    - View their booking history.

Task 4: Super Admin Dashboard

- Build a dashboard for the super admin to:
    - View a list of all users (partners, clients).
    - View all registered companies and services.
    - View all bookings with client and partner details.

Task 5: Role-based Authorization

- Ensure users can only access actions and data relevant to their roles. For example:
    - A client shouldn't access partner data.
    - A partner shouldn't see bookings made by clients of other partners.

Task 6: Additional Features (Optional)

- Add email notifications for booking confirmations to both the client and partner.
- Add pagination for large datasets (e.g., bookings list).

Submission Requirements

Email the following requirements to **recruitment@taurgo.co.uk** before the deadline

1. Source code (organized and commented).
    - Upload your code to a version control platform (e.g., GitHub, GitLab, or Bitbucket). Make sure to commit the code daily.
    - Ensure the repository is public or share access with the reviewer.
    - Include a meaningful README file that explains the project and setup instructions.
2. A document explaining:
    - System architecture and database design.
    - How role-based access control is implemented.
    - Steps to run the application.
3. Screenshots or a video demo of the application.
    - Logging in as different user roles.
    - Key features implemented for each role.

- Any unique aspects or improvements you've made.

Assessment Duration

You will be given 2-3 days to complete the task. Late submissions may not be evaluated unless prior approval has been given.

What we expect

1. **Focus on Completing the Core Requirements:**
   We understand that time may be limited, so prioritize implementing the essential features for each user role:
   - Partner: Add and manage company details and services.
   - Client: Book services and view bookings.
   - Super Admin: View and manage all users, services, and bookings.
2. **Demonstrate Your Skills:**
   The goal is to assess your understanding of:
   - Role-based access control.
   - CRUD operations.
   - Clean, modular, and scalable code.
   - Database integration (or working with mock data).
3. **Show Progress, Even if Incomplete:**
   If you cannot complete every feature, ensure that the parts you do implement are functional, well-structured, and showcase your best work.
4. **Present Your Solution:**
   Prepare to explain:
   - Your approach to solving the task.
   - Key challenges and how you addressed them.
   - Any additional features or improvements you made beyond the requirements.
5. **Document Your Work:**
   Submit a basic documentation file that includes:
   - Instructions to set up and run your application.
   - A description of the features you implemented.
   - Any limitations or known issues in your solution.

What Matters Most

- Quality over quantity: A partially completed but well-implemented solution is better than a rushed, incomplete one.
- Your problem-solving approach and how you prioritize tasks.
- Your ability to communicate and document your work clearly.

This way, you set clear, realistic expectations for the intern while encouraging them to showcase their skills and make meaningful progress.

Contact for Questions

If you face issues or have questions, reach out to **recruitment@taurgo.co.uk** before the deadline.

All the very Best!