

Full Stack JavaScript Engineer Role

Test Assignment

Objective:

The purpose of this assignment is to design and build a comprehensive web application for a smart elevator maintenance company. The application should offer a user-friendly dashboard that displays the operational status of all elevators, as recorded in the database, for the currently logged-in user.

The dashboard should provide the following information:

- The total number of elevators that are currently operational, those that are issuing warnings, and those that are out of service.
- A list of elevators that have been recently visited or inspected.

The application should also offer interactive functionality:

- Clicking on the count of elevators in any state (operational, warning, or out of service) will display a detailed list of those specific elevators.
- Clicking on any individual elevator from any list will present detailed information about that particular elevator's status and history, along with a chart displaying analytical data about its operations.

Requirements:

Security:

Authenticate users using Auth0¹.
Create two independent demo users.

Frontend:

Use ReactJS² for frontend development.

The dashboard should display:

- Counts of operational, warning, and out-of-service elevators.
- A list of recently visited elevators.
- On clicking any count, display a list of elevators in that state.
- On clicking a specific elevator, display its details (in all listings)

¹ <https://auth0.com/docs/quickstarts>

² <https://react.dev/reference/react>

Backend:

Use ExpressJS³ for backend development.

The API should provide endpoints that support frontend actions.

Database:

Use MongoDB⁴ as the database to store elevator and user data.

Tests:

Write tests for the frontend and backend. Use the testing framework you're comfortable with.

Documentation:

Document the API endpoints, frontend components, and database schema. Include setup instructions and any assumptions made.

Version Control:

Use GitHub⁵ for version control. Create a private repository and share it with us when you are done.

Infrastructure *(optional)*:

Write an Amazon CDK⁶ to build this infrastructure using API Gateway and AWS Lambda. Host the frontend in an S3 bucket.

Test Data:

Example elevator data is provided in two JSON files inside the “Example Data” folder.

Deliverables:

1. Link to the private GitHub repository containing the application code, tests, and documentation.
2. (Optional) Amazon CDK configuration.
3. Live demo of the application, locally hosted.

³ <https://expressjs.com/en/4x/api.html>

⁴ <https://www.mongodb.com/docs/launch-manage/>

⁵ <https://github.com/new>

⁶ https://docs.aws.amazon.com/cdk/v2/guide/getting_started.html

Evaluation:

Your submission will be evaluated on the following criteria:

- Code quality and organisation.
- Adherence to the requirements.
- Completeness of tests and documentation.
- Ability to reason the choices made in the development process.

Please let us know if you have any questions or need further clarification on any part of the assignment.

Good luck 🍀