

User Management Project - QA Automation Challenge

This project is developed as part of a QA Automation job application challenge. It demonstrates proficiency in backend API development, frontend React UI, API and UI test automation using Cypress, and API testing with Postman. The project includes user management functionalities such as creating, updating, deleting, and listing users, along with validation and error handling.

Project Overview

The project consists of three main parts:

- Backend API built with Express.js handling user data with REST endpoints.
- Frontend React application providing a User Management interface.
- Automated tests covering API and UI functionalities using Cypress.

Backend API

- Implemented using Express.js with endpoints for GET, POST, PUT, DELETE user operations.
- Validation to prevent duplicate emails and handle missing fields.
- Simple authentication via a login endpoint.

Frontend React UI

- Built with React functional components and hooks.
- User-friendly forms with inline validation messages.
- Styled with clean, modern CSS for a professional look and feel.
- Error messages displayed clearly on API or validation failures.

Automation Testing with Cypress

- End-to-end tests covering all critical user flows:
 - * Listing users
 - * Creating new users
 - * Updating existing users
 - * Deleting users
 - * Validating input fields and duplicate emails
- Tests are designed to run independently and clean up test data to avoid conflicts.
- Cypress configuration includes Mochawesome reporter for detailed HTML reports.

API Testing with Postman

- Postman collections created for all API endpoints.
- Automated tests written in Postman to verify correct API responses and error handling.
- Tests cover scenarios like successful user creation, duplicate email errors, update and delete operations.

Running the Project

1. Start the backend server by running `node server.js` (Express.js API on port 3001).
2. Start the React frontend app using `npm start` (default on port 3000).
3. Run Cypress tests with `npx cypress run --spec "cypress/e2e/UserManagement.cy.js"`.
4. View generated test reports in `cypress/reports` folder (Mochawesome HTML reports).

Summary

This project demonstrates strong proficiency in full-stack development, quality assurance, and automation testing. It covers:

Designing and implementing a RESTful API with robust validation.

Building a clean and responsive React UI.

Writing effective automated API tests using Postman.

Creating reliable end-to-end UI tests with Cypress, along with detailed reporting.

Delivering a maintainable and extensible codebase ready for real-world applications.

Prepared by Mehmet Furkan Ersamut