Project Documentation – QA Automation Challenge

This document explains how to install, configure, and run the automated tests developed for the User Management project, which consists of:

- A Node.js + Express backend (API)

- A React frontend UI

- Automated API tests using Postman and Newman

- Automated UI tests using Cypress

- Reporting with Mochawesome

- Code written in JavaScript

Technologies Used

Tool / Library	Purpose
Node.js	Backend server runtime
Express	API framework for Node.js
React	Frontend UI framework
Postman	Manual and automated API testing
Newman	CLI runner for Postman collections
Cypress	End-to-end UI test framework
Mochawesome	HTML report generation for Cypress
JavaScript	Programming language used in all layers

API Testing (qa-backend + postman-tests)

Installation

```
cd qa-backend

npm install

npm start
```

Backend will run at: http://localhost:3001

Run API Tests (with Newman)

```
cd ../postman-tests
```

```
newman run user-management.postman_collection.json -e
env.postman_environment.json -r cli,html --reporter-html-export report.html
```

This will:

- Run all API tests.
- Export a readable HTML report as report.html

Frontend Testing (qa-frontend + Cypress)

Installation

```
cd ../qa-frontend
```

```
npm install
```

```
npm start
```

Frontend will run at: http://localhost:3000

Run UI Tests with Cypress

Option 1: Cypress GUI

```
npx cypress open
```

Option 2: Headless mode with report

```
npx cypress run --reporter mochawesome --reporter-options
reportDir=cypress/reports,overwrite=false,html=true,json=true
```

This will:

- Run all tests automatically.
- Generate a detailed report in cypress/reports/

Test Coverage & Reports

UI Test Scenarios

- Login with valid/invalid credentials
- Create a new user

- Update an existing user

- Delete a user

- Prevent duplicate emails

- Validate required fields

- List users

- Handle API errors

- Form validation for empty fields

API Test Scenarios

- POST /login

- GET /users

- POST /users

- PUT /users/:id

- DELETE /users/:id

- Positive and negative cases for all endpoints

Quick Start

```
# Start Backend

cd qa-backend

npm install

npm start

# Start Frontend

cd ../qa-frontend

npm install

npm start

# Run API Tests

cd ../postman-tests

newman run user-management.postman_collection.json -e env.postman_environment.json -r cli,html --reporter-html-export report.html
```

# Run UI Tests (Headless)

```
cd ../qa-frontend
```

```
npx cypress run --reporter mochawesome --reporter-options reportDir=cypress/reports,html=true
```

Notes

- The backend and frontend must both be running for Cypress tests to work correctly.

- Postman and Newman tests require user-management.postman_collection.json and env.postman_environment.json.

- Cypress test reports are saved as HTML and JSON for easy review.