Project work for Software Design Engineer in Test at Scitara

We are looking for engineers who can join us in building of software by helping left shift the quality motto. In today's day and age, cent-percent automation is not a dream; it is an outcome of persistent effort of a driven team. And in the age of generative and agentic AI tools, this calls for engineers who are smart enough to master the tools and deliver the goods.

To aid our selection process, we invite you to attempt any one of the following projects prior to appearing for a Technical Evaluation round. You are welcome to leverage any AI tool you may have – this is test for problem solving and not syntax memorization. You are then expected to demonstrate the working of your code. Your knowledge of the code you present will be key factor in our selection process.

All project code may be submitted as a ZIP (excluding node modules) at least one hour before the Technical Evaluation timeslot

Do feel free to reach out to Scitara for clarifications; contact HR and an appropriate technical person will connect back.

Project 1: Backend API Service + Automation

Part A: REST API Service Development for User management (Candidate should use Node.js, In-Memory DB or JSON Mock Data)

Endpoint: /api/users

Method	Endpoint	Description
POST	/users	Create a new user
GET	/users	Get all users
GET	/users/{id}	Get user by ID
PUT	/users/{id}	Update user
DELETE	/users/{id}	Delete user

Mandatory API Behavior

Proper HTTP status codes (200, 201, 400, 404, 500)

Input validation

Error responses with meaningful messages

Part B: Automation for REST APIs

Tool choice (Any JavaScript based tool suitable for automation API's developed in Part A)

Expectation:

- 100% coverage of all REST endpoints

- Positive + negative test cases

- Schema validation

- Data-driven tests

Part C: Advanced Requirement (WebSocket OR Kafka Integration)

(This part is optional for candidates with less than 5 years of experience)

Option 1: Using WebSocket:-

WebSocket emits event when:-

- User is created

- Order status changes

Automation:

- Connect to WebSocket

- Validate messages/events

- Handle async behavior

Option 2: Using Kafka

/api/user POST should trigger Kafka event called "UserCreatedEvent" and then User should be saved (either in-memory or DB)

Consumer should consume this "UserCreatedEvent" and logs message that User is created successfully.

Automation: Should simulate the above flow

Project 2: React UI + Automation

Part A: Create a dummy E-Commerce UI

Pages

- Product Listing Page
  - Load products from Local JSON file OR Mock API
- Display (Product name, Price, Category, Add to cart button)
- Product Details Page
  - Click product → view details
  - Add to cart
- Cart Page
  - View selected items
  - Update quantity
  - Remove item
  - Display total price
- Checkout Page (Mock)
  - User details form
- Submit order

Part B: UI Automation

Tool Choice (Cypress / Playwright)

Expectation:

- 100% automation of all UI flows