

Build a polished **frontendonly prototype** (React/Angular/Vue — candidate's choice) that simulates a **AI Platform for Quality Engineering** user experience. It should not implement AI logic — instead, use static JSON mocks either provided here or candidate can create their own.

**The prototype should demonstrate:**

- UI/UX design skills
- Component breakdown
- Information architecture
- State management
- Responsive layout
- Interaction design (upload, navigation, viewing details)
- Frontend engineering quality

**Deliverable:**

- A working frontend app (no backend required)
- GitHub repo
- Deployed link (Netlify/Vercel/GitHub Pages)
- A short README explaining:
  - design decisions
  - component structure
  - how to run

**To Build:**

**1. "Upload" Screen (Simulated)**

A simple page with:

- A draganddrop upload box
- It doesn't parse files — clicking "Upload & Continue" loads the mock JSON data into the app state
- Show a minimal animation/loading state

**2. Requirements Viewer**

A page listing all requirements in a clear, structured form of "AIenhanced, contextrich requirements".

Required UI features:

- Filter by **FR / NFR / All**
- Search bar
- Requirement card/table row with:
  - ID (REQ001)
  - Type tag (FR/NFR)
  - Title
  - Statement
  - Badge showing number of linked tests

Bonus UX points:

- Hover reveals full statement
- Sorting
- Colorcoding by type

**3. Test Case Explorer**

A page that lists all auto-generate test cases from requirements.

Each test case card/row should display:

- Test ID (TESTXXX)
- Requirement it links to
- Title
- Collapsible section for:
  - Steps
  - Expected results

Interactions:

- Clicking the linked Requirement ID opens a rightside panel or modal showing REQ details
- Filters:
  - By requirement
  - Positive / negative / boundary (inferred via title keywords)

#### 4. Traceability Matrix (Visual View)

A core feature of such tools is explicit requirements-to-test traceability.

The candidate must design a visualization such as:

Option A — Table Matrix

Requirement	Linked Test Cases
REQ001	TEST001, TEST002
REQ002	TEST003
REQ003	— (show warning message)

Option B — Node Graph Visualization

- Left column: Requirements
- Right column: Test cases
- Lines drawn between them

This tests creativity + UI skills.

Mandatory UX:

- Highlight **uncovered requirements** (red badge)
- Toggle expansion/collapse
- Show coverage stats (e.g., "Coverage: 67%")

#### 5. Application Navigation (Routing)

Use a modern routing approach:

Tabs or sidebar:

- Upload
- Requirements
- Test Cases
- Traceability Matrix

#### 6. Responsiveness & Styling

- Must work beautifully on mobile & desktop
- Use a UI library (Material UI, Chakra, Tailwind, Bootstrap) OR custom components
- Show clean spacing, hierarchy, contrast, and alignment