

Assignment Overview

The candidate will create a **responsive UI form** mimicking the first two steps of the Udyam registration process

(<https://udyamregistration.gov.in/UdyamRegistration.aspx>).

Key Focus Areas:

1. **Web Scraping:** Extract form fields, validation rules, and UI structure from the Udyam portal.
 2. **Frontend Development:** Rebuild the UI with modern frameworks (React/Next.js preferred).
 3. **Backend Integration:** Handle form submissions, validation, and data storage.
-

Tasks

1. Web Scraping (Step 1 & 2)

- **Goal:** Identify all input fields, labels, validation rules (e.g., PAN/Aadhaar formats), and UI components (dropdowns, buttons) from the Udyam portal's first two steps.
- **Tools:** Use Python (BeautifulSoup/Scrapy) or JavaScript (Puppeteer/Cheerio).

Perform scraping on the first two steps i.e aadhaar + OTP validation and Pan Validation on the udyam website form -

<https://udyamregistration.gov.in/UdyamRegistration.aspx>

2. Responsive UI Development

- **Requirements:**
 - Replicate the Udyam form's layout with a **mobile-first approach** (ensure 100% responsiveness).
 - Use **React/Next.js** (TypeScript preferred) or vanilla HTML/CSS/JS.
 - Implement dynamic form rendering based on the scraped JSON schema.
 - Add real-time validation (e.g., PAN format: [A-Za-z]{5}[0-9]{4}[A-Za-z]{1}).
 - **UI Enhancements (Bonus):**
 - Auto-fill suggestions for city/state based on PIN code (use an API like PostPin).
 - Progress tracker showing Steps 1 & 2.
-

3. Backend Implementation

- **Requirements:**

- Build a REST API (Node.js/Python) to:
 - Validate form data against the scraped rules.
 - Store submissions in PostgreSQL.
 - Design a database schema matching the Udyam form fields.
 - **Tools:**
 - Node.js: Express + Prisma ORM.
 - Python: FastAPI + SQLAlchemy.
-

4. Testing

- Write unit tests for:
 - Form validation logic (e.g., invalid PAN triggers error).
 - API endpoints (e.g., POST /submit returns 400 for invalid data).
 - Tools: Jest (JavaScript), pytest (Python).
-

5. Deployment (Bonus)

- Dockerize the application.
 - Deploy frontend (Vercel/Netlify) and backend (Heroku/Railway).
-

Evaluation Criteria

Area	Key Metrics
Scraping	Accuracy of extracted fields, handling dynamic content (if any).
UI/UX	Pixel-perfect responsiveness, intuitive error messages, smooth transitions.
Backend	REST API correctness, validation logic, database schema design.
Code Quality	Clean architecture, modular code, proper comments, Git practices.
Testing	Coverage of edge cases (e.g., invalid Aadhaar, empty fields).