Final Hackathon Project

**Problem Statement: Identify Courses**

**Thematic**

* Streamline the discovery of beginner-friendly web development courses
* Pinpoint language learning opportunities
* Verify form error handling on an online education platform like Coursera.
* This project showcases high-level automation expertise
* Covering browser navigation, data retrieval, user interaction, and validation mechanisms.

**Automated Course Discovery & Form Validation**

**Core Features**

* Instant search for beginner-friendly web development courses.
* Extracts course names, duration, and ratings.
* Identifies available languages and learning levels.
* Automates form interactions, triggering and capturing errors.
* Handles dropdowns, navigation, and browser windows.

**Automation Scope**

* **Navigation:** Seamless movement between homepage, search results, and form pages.
* **Data Extraction:** Retrieves key course details and user options.
* **Form Handling:** Auto-fills and verifies input errors.
* **Dropdown Processing:** Parses selections for structured display.

**Key Tasks**

**1. Web Development Course Extraction**

**Goal:** Find top two beginner-friendly web dev courses with essential details. **Steps:**

1. Open Coursera.
2. Search for "Web Development."
3. Filter **Beginner** level & **English** language.
4. Extract:
   * Course name
   * Duration
   * Rating
5. Display the extracted data.

**2. Language Learning Options Extraction**

**Goal:** Identify all available languages & count learning levels. **Steps:**

1. Open "Language Learning" category.
2. List all supported languages.
3. Extract learning levels per language (Beginner, Intermediate, Advanced).
4. Count levels & format the output neatly.

**3. Enterprise Form Validation**

**Goal:** Test error handling on inquiry forms. **Steps:**

1. Open "For Enterprise" → Navigate to **Courses for Campus**.
2. Locate the **"Ready to Transform"** form.
3. Fill valid details but enter an incorrect email.
4. Submit & capture error messages.

**Technical Details**

* Uses **Selenium WebDriver** for automation.
* Structured data handling with **Java Collections / JSON/ Properties**.
* Robust error management for delays and missing elements.
* Console-based reporting for extracted information.

**Tools & Technologies**

* **Languages:** Java (TestNG) .
* **Framework:** Selenium.
* **Data Handling:** JSON / CSV.
* **Reporting:** Console output.

**How to Run**

1. Clone the repository.
2. Install dependencies (pom.xml or requirements.txt).
3. Configure browser drivers (e.g., **ChromeDriver**).
4. Run via TestNG:

testng.xml

1. View structured output in the console.

**Assumptions**

* The platform (Coursera) maintains public availability.
* Filters (level, language) remain consistent.
* Forms are accessible without login requirements.