QA Capstone Project Document

QA Capstone Project Document	
Application To be Tested	**Website Link:** Here. If the website is down, run the app locally by following these steps.
Target Modules	Login Module and Recruitment (Candidates and Vacancies Menu) Module
Objective	To help students revise and practice essential QA skills, tools, and techniques by working on real-world scenarios

Introduction:

- The QA and Testing Capstone Project is a comprehensive learning experience designed to bring together all the foundational concepts, tools, and practices of Quality Assurance (QA) and Software Testing in a practical, real-world scenario. This project provides students with the opportunity to apply their knowledge in a simulated or real environment, ensuring they develop the skills required for professional QA roles.

Objective:

- The primary objective of this QA and Testing capstone project is to provide students with a platform to consolidate and apply their knowledge of essential Quality Assurance (QA) skills, tools, and techniques in a practical, real-world context. This project is tailored to:

  - **Reinforce Core QA Concepts**: Strengthen students' understanding of QA principles through immersive, hands-on practice.

- **Simulate Real-World Scenarios**: To experience and address practical quality assurance challenges that mimic industry environments.
- **Enhance Problem-Solving Skills**: Foster critical thinking and decision-making abilities specific to QA processes and practices.
- **Gain Proficiency in Tools and Techniques**: Provide hands-on exposure to industry-standard QA tools and methodologies, such as TestLink, Postman, JIRA, Cypress, K6, and more.
- **Bridge Theory and Practice**: Equip students with the ability to translate theoretical QA concepts into actionable, professional-grade testing strategies.
- **Prepare for Professional QA Roles**: Develop the technical, analytical, and collaborative skills necessary for success in QA positions in the software industry.

Phase 1: Manual Testing and Test Management

➢ **Part A: Jira Project Setup**

1. **Create a Jira Project:**
    - Name the project **"QA Capstone Project"**.
    - Create tasks that you are going to work on this project
    - Assign roles for yourself
2. **Invite Instructors:**
    - You MUST share the Jira project link and invite instructors for monitoring and feedback.

➢ **Part B: Manual Testing Tasks**

1. **Prepare Test Cases:**
    - **You** are expected to create test cases for vacancies menu functionality based on the SRS document.
    - **Focus Areas to Test:**
        - **Recruitment Module (Vacancies Menu)**
            - Add new vacancies.
            - Search vacancies.
            - Error messages for mandatory fields.

2. **Log Artifacts to TestLink:**
    - Upload the **SRS**, **Test Plan**, and **Test Cases** into TestLink. **Note**: Make sure to upload the test cases that are provided to you and the test cases that you prepared which are both given and you will have prepared for vacancies menu functionality as stated in Part B above.
    - Organize test cases into folders in TestLink for Login Module, Recruitment Module (Candidates Menu) and Recruitment Module (Vacancies Menu).
3. **Manually Execute Test Cases:**
    - Execute the test cases (both test cases that are provided to you and that you created) in **TestLink** and mark results as **Pass** or **Fail**.
4. **Generate Test Report:**
    - **Export a test execution report from TestLink,** including Total test cases, number of passed, failed, and blocked test cases.

- **Resources**:
    - **QA Capstone Intro Video**: Link to the video
    - **SRS Document:** Link to SRS Document
    - **Test Plan Document**: Link to Test Plan Document
    - **Functional Test Case Document:** Link to Functional Test Cases Document
    - **Performance Test Case Document:** Link to Performance Test Cases Document
    - **Step-by-Step Guide to Install and Run OrangeHRM Application Locally**: If the online orange HRM website here is DOWN, please set up the project and launch it locally by:
        - Following these steps or
        - Watching the installation video
        - **Note**: There is no need to go through the above if their website is working.

Phase 2: Automation Testing with Cypress

- **Objective:** To automate End-to-End (E2E) testing for the Login Module and the Recruitment Module (Candidates and Vacancies Menu) for OrangeHRM website if the hosted demo is available unless use your local OrangeHRM app you installed before, ensuring the application works as expected by leveraging Cypress for efficient test execution and reporting.
- **Implementation includes:** The implementation process encompasses the following key components, with flexibility to include additional elements or tasks as necessary.

1. **Cypress Hooks:**
    - **before():** Used for one-time setup, such as loading environment variables or test data.
    - **beforeEach():** Used for repetitive setup tasks like navigating to the login page before each test.
    - **after():** Used for cleanup tasks like deleting test data or logging results.
    - **afterEach():** Used for performing tasks like taking screenshots for failed tests.

2. **Data-Driven Testing (DDT):**
    - Use external data files (Excel) for testing various input combinations such as to test valid and invalid login scenarios using multiple datasets.
3. **Parameterization:**
    - Pass test data dynamically for valid and invalid login scenarios. candidate creation with different details and vacancy creation with varied input combinations.
4. **Page Object Model (POM):** Organize test scripts for better readability and reusability by creating separate files for each page/module:
    - **Login Page**: Define locators for username, password, login button, and error message. Create reusable functions for login actions (e.g., entering credentials, clicking login).
    - **Recruitment Module (Candidates):** Define locators and functions for adding candidates, validating error messages, and searching for candidates.

- **Recruitment Module (Vacancies):** Define locators and functions for adding vacancies, validating error messages, and searching for vacancies.

5. **Retries:**
   - Enable test retries for flaky tests using Cypress's retry mechanism.
6. **Cypress Reporting:**
   - Generate reports using **Mochawesome Reports**.
7. **Cypress Cloud:**
   - Run and monitor test executions on **Cypress Cloud**.
   - Invite instructors for your Cypress cloud project.

➢ **Test Scenarios to Automate:**

1. **Login Module:** The login module ensures that users can access the application securely and that validation works for all input scenarios.
   - **Scenarios:**
     1. **Valid Login:**
        - Test successful login with correct username and password.
        - Assert that the user is redirected to the dashboard page upon successful login.
     2. **Invalid Login:**
        - Test unsuccessful login with incorrect username or password.
        - Validate that the application displays an error message: **"Invalid username or password"**.
        - Assert that the user remains on the login page.
     3. **Field Validation:**
        - Test login attempts with:
          - Empty username and/or password fields.
          - Invalid input formats (e.g., special characters or overly long inputs).
        - Validate that the application shows appropriate error messages.
2. **Recruitment Module (Candidates Menu):** This module allows users to manage candidates by adding, validating, and searching through candidate records.
   - **Scenarios:**
     1. **Add New Candidate:**

- Add a new candidate by filling in all mandatory fields with valid details.
- Assert that the candidate is successfully added to the candidate list.

2. **Mandatory Field Validation:**
   - Leave mandatory fields empty while trying to add a new candidate.
   - Validate that the system displays appropriate error messages indicating the missing information.
3. **Search Candidates:**
   - Search for candidates using the provided search filters, such as:
     - Job Title
     - Hiring Manager
     - Date of Application
   - Assert that the search results correctly display the candidates matching the specified criteria.

3. **Recruitment Module (Vacancies Menu):** This module enables users to manage job vacancies by adding, validating, and searching for vacancies.
   - **Scenarios:**
     1. **Add New Vacancy:**
        - Add a new job vacancy by filling in all required fields with valid details.
        - Assert that the vacancy is successfully added to the vacancy list.
     2. **Mandatory Field Validation:**
        - Leave mandatory fields empty while trying to add a new vacancy.
        - Validate that the system shows appropriate error messages highlighting the missing fields.
     3. **Search Vacancies:**
        - Search for vacancies using various search filters, such as:
          - Job Title
          - Hiring Manager
          - Number of Positions

- Assert that the search results accurately display the vacancies matching the applied filters.

Phase 3: Performance Testing with K6

- **Scope:** The primary focus of this phase is to conduct performance testing on the **Login Page** of the OrangeHRM application. If the hosted demo is not available, use the locally installed orange HRM web application. This phase evaluates the system's ability to handle concurrent users, ensuring responsiveness, stability, and reliability under load.
- **Objective:** The objective is to assess critical performance metrics, such as response time, error rate, and throughput, under simulated real-world scenarios, using K6, a modern load-testing tool, with data visualizations enabled through Grafana.
- **Implementation includes:**

1. **Parameterization:**
   - Dynamically read multiple sets of usernames and passwords from an external JSON file to simulate diverse user credentials.

2. **Assertions:**
   - Validate critical performance requirements to ensure system reliability under load:
3. **Scenarios:**
   - Simulate Ramping-up, Ramping-down and consistency concurrent users.
4. **Visualization with Grafana:**
   - Export performance metrics to Grafana and analyze response times, error rates, and throughput.
   - Visualize metrics such as:
     - Average and peak response times.
     - Error rates (percentage of failed requests).
     - Throughput (requests per second).

- **Test Script Overview:**
  - Simulate multiple users logging in with valid credentials.
  - Collect performance metrics (response time, error rate, throughput).

Deliverables:

- **Phase 1 Deliverables:**
  - Jira project setup with tasks and subtasks.
  - Test cases logged into TestLink.
  - Test execution report from TestLink.
- **Phase 2 Deliverables:**
  - Cypress test scripts adhering to best practices (POM, DDT, retries).
  - Execution reports (Mochawesome).
  - Results shared via Cypress Cloud.
- **Phase 3 Deliverables:**
  - K6 performance test scripts.
  - Grafana visualization of performance metrics.

Evaluation Criteria:

- **Phase 1: Manual Testing**
  - Completeness and accuracy of test cases.
  - Quality of test execution and reporting.
- **Phase 2: Automation Testing**
  - Adherence to automation best practices, such as Page Object Model (POM) and Data-Driven Testing (DDT).
  - Script readability and maintainability.
  - Successful execution on the Cypress Cloud platform.
  - Engagement with instructors for feedback and guidance.
- **Phase 3: Performance Testing**
  - Accuracy of test scripts created with K6.
  - Validity of test assertions.
  - Detailed test reports in JSON format.
  - Analysis and insights derived from Grafana visualizations.