**Project Title: Web Automation with Selenium (100 Marks)**

**Objective:**

Students will set up their automation environment, write and debug Selenium test scripts, handle various web elements, and apply the Page Object Model (POM) design pattern.

**Project Breakdown and Marking Scheme**

**Part 1: Environment Setup and Basics**

**1. Setup Eclipse and Java WebDriver**

* **Task:**
  + Install Eclipse IDE and Java Development Kit (JDK).
  + Set up Selenium WebDriver in Eclipse.
* **Deliverables:**
  + Screenshots or a short video showing installation and configuration steps.

**2. Understanding Object Identification (10 Marks)**

* **Task:**
  + Learn and document ways to identify web elements using XPath and CSS Selectors.
* **Deliverables:**
  + A report with examples and explanations of XPath and CSS Selectors.
* **Marking:**
  + Accuracy and completeness of explanations: 6 Marks
  + Quality of examples and clarity: 4 Marks

**3. Configuring WebDriver (10 Marks)**

* **Task:**
  + Download and configure Java WebDriver for Firefox and Chrome browsers.
* **Deliverables:**
  + Code snippets initializing WebDriver for both browsers.
* **Marking:**
  + Correct setup and usage of WebDriver: 6 Marks
  + Quality and clarity of code snippets: 4 Marks

**4. Hands-on Practice (15 Marks)**

* **Task:**
  + Write a Selenium script to automate basic navigation and interaction with the [example website](https://example.com) using id, name, linkText, className, and xpath.
* **Deliverables:**
  + A functional script demonstrating interaction with the website.
* **Marking:**
  + Correctness and functionality of the script: 10 Marks
  + Code organization and readability: 5 Marks

**Part 2: Advanced Web Element Handling**

**1. Handling Different Web Elements (15 Marks)**

* **Task:**
  + Write scripts to handle different types of web elements on the [DemoQA Forms page](https://demoqa.com/automation-practice-form): input boxes, buttons, list/selection/drop-down boxes, radio buttons, and checkboxes.
* **Deliverables:**
  + Scripts demonstrating interaction with each type of web element.
* **Marking:**
  + Accuracy and functionality of scripts: 10 Marks
  + Clarity and organization of code: 5 Marks

**2. Extracting Links and Synchronization (15 Marks)**

* **Task:**
  + Extract all links and other web elements from the [Wikipedia Main Page](https://www.wikipedia.org). Implement synchronization using implicit and explicit waits.
* **Deliverables:**
  + A script that extracts links and demonstrates synchronization.
* **Marking:**
  + Functionality of extraction and synchronization: 10 Marks
  + Correct implementation of waits: 5 Marks

**3. Simulating Keyboard and Mouse Actions (10 Marks)**

* **Task:**
  + Simulate keyboard and mouse actions on the [Google Search Page](https://www.google.com). For instance, perform a search and handle search suggestions.
* **Deliverables:**
  + A script demonstrating the use of the Actions class for keyboard and mouse actions.
* **Marking:**
  + Correct usage of Actions class: 7 Marks
  + Functionality and clarity of the script: 3 Marks

**4. Debugging Tests (60 mins) (10 Marks)**

* **Task:**
  + Debug a provided Selenium script that interacts with the [Selenium Official Site](https://www.selenium.dev) to identify and fix issues.
* **Deliverables:**
  + A debugged version of the script with explanations of issues and fixes.
* **Marking:**
  + Effectiveness of debugging and fixes: 7 Marks
  + Clarity of explanations: 3 Marks

**Part 3: Page Object Model (POM)**

**1. Understanding Page Object Model (POM) (10 Marks)**

* **Task:**
  + Learn and document the Page Object Model design pattern and its benefits.
* **Deliverables:**
  + A report explaining POM, its advantages, and its implementation.
* **Marking:**
  + Accuracy and completeness of the explanation: 6 Marks
  + Quality and clarity of the report: 4 Marks

**2. Implementing POM in Scripts (10 Marks)**

* **Task:**
  + Refactor existing scripts to use the Page Object Model. Use the DemoQA Forms page for this task.
* **Deliverables:**
  + Refactored scripts demonstrating the application of POM.
* **Marking:**
  + Correct application of POM principles: 5 Marks
  + Organization and readability of refactored code: 5 Marks

**Project Submission**

1. **Code Repository:**
   * Submit all code in a Git repository. Ensure the repository is well-organized and includes a README file.
2. **Documentation:**
   * Include detailed documentation on setup, usage, and explanations of each script.
3. **Final Report:**
   * A comprehensive report summarizing the learning outcomes, challenges faced, and solutions.
4. **Presentation:**
   * Optional: Prepare a short presentation summarizing the project work.

**Total Marks: 100**