Project Plan for ToDoMVC Testing

Week 1: Planning & Environment Setup

Define project objectives:

- Conduct Manual Testing to understand ToDoMVC functionalities.

- Implement Automation Testing using Selenium.

- Cover core features: adding, editing, deleting, and filtering tasks.

Set up the testing environment:

- Install Selenium WebDriver, TestNG.

- Set up browser drivers (Chrome،Firefox,Edge).

- Configure Git/GitHub for version control.

- Select a defect tracking tool (Jira).

Tools used:

- Manual Testing: Test case documentation (Jira).

- Automation Testing: Selenium, TestNG.

- Defect Tracking: Jira.

- Version Control: Git/GitHub.

Week 2: Manual Test Design & Execution

Identify manual test scenarios:

- UI Testing: Verify layout, buttons, input fields.

- Functional Testing: Ensure task addition, editing, deletion, and filtering work correctly.

- Negative Testing: Test empty inputs, duplicate tasks, and unexpected inputs.

Write manual test cases:

- Document test cases in Jira

- Define test data and expected results.

Execute manual test cases:

- Perform testing on Chrome & Firefox.

Week 3: Automation Test Script Implementation

Automate core test cases using Selenium:

- Adding a task → Verify task appears in the list.
- Editing a task → Ensure updates reflect correctly.
- Marking a task as completed → Validate checkbox functionality.
- Deleting a task → Confirm removal from the list.
- Filtering tasks (Active, Completed, All) → Check correct filtering behavior.

Implement Assertions & Locators:

- Use TestNG to validate results.

Code Management:

- Push test scripts to GitHub repository.
- Organize test cases using Page Object Model (POM) for maintainability.

Week 4: Test Execution & Debugging

- Execute both Manual & Automated test cases.
- Identify and log defects using Jira or a defect trackiTestinl.
- Debug failing test scripts and refine locators/logic.
- Perform Cross-Browser Testing (Chrome, Firefox).

Week 5: Performance & API Testing

Basic performance testing:

- Use JMeter to measure page load time.

API Testing :

- Use Postman to verify API requests (GET, POST, DELETE).

Week 6: Reporting & Final Evaluation

Generate reports:

- Manual Test Report: Document test cases, results, and defects.
- Automation Test Report: Generate TestNG/Extent Reports.

- Bug Report: Summary of identified issues.

Final Presentation:

- Showcase manual and automation testing workflow.
- Discuss challenges and key learnings.