**JIOMART**

**Playwright and TypeScript Automation**

**Project Overview – Planning Phase**

**Project Title:** E-Commerce Web Automation Testing using Playwright and TypeScript  
**Application Under Test (AUT):** [https://www.jiomart.com](https://www.jiomart.com/)

**Tech Stack:**

* **Automation Tool:** Playwright
* **Language:** TypeScript
* **Design Pattern:** Page Object Model (POM)
* **Test Runner:** Playwright Config file (playwright.config.ts)
* **Assertion:** Built-in Playwright ‘expect’ assertions

**Folder Structure:**

* **pages/** – Contains .ts files for each page (e.g., homePage.ts, loginPage.ts, productPage.ts)
* **tests/** – Contains spec.ts files with all test cases
* **utils/** – Includes config.ts with environment constants and reusable data
* **fixtures/** – Manages custom fixtures to inject page objects using POM

jiomart-playwright-automation/

├── pages/

│ ├── homePage.ts

│ ├── loginPage.ts

│ ├── productPage.ts

│ ├── cartPage.ts

│ └── wishlistPage.ts

├── tests/

│ └── jiomart.spec.ts

├── utils/

│ └── config.ts

├── fixtures/

│ └── fixtures.ts

├── playwright.config.ts

├── package.json

├── tsconfig.json

└── README.md

**Test Scenarios Covered:**

* Login/Logout – Valid login credentials
* Add to Cart – Proceed till checkout, Cart functionalities
* Add to Wishlist – Wishlist functionalities
* Product Search – Verify search and display functionalities
* Apply Filters – Apply different filters and verify results
* Apply Sort Options – Apply different sorts and verify the product list
* Handle Out-of-Stock Products – If it correctly displays out of stock products
* Validate Search Results List – Validate search with list functionality
* Profile page updates – Validating profile changes and settings changes

**Objectives:**

* Automate critical user flows on JioMart to ensure functional reliability
* Build a scalable and maintainable test architecture using Playwright
* Validate both positive and negative scenarios to ensure robustness