**AUTOMATION TESTING OF MAGICBRICKS WEB APPLICATION**

**NAME :** JYOTHIRMYE B

**DATE :** 13.09.2025

**ABSTRACTION :**

This project aims on conducting functional and regression testing of the Magicbricks web application, a leading digital platform for real-estate services. The primary aim is to validate that the core features — such as property search, user login, and interactive navigation , exploring, comparing localities— work seamlessly, reliably, and as per the business requirements. By ensuring the stability of these critical modules, the project contributes to enhancing user experience and maintaining application quality.

The testing effort was automated using Selenium WebDriver integrated with Java, Cucumber, and TestNG frameworks. This automation framework not only minimized repetitive manual work but also improved test accuracy, accelerated execution speed, and expanded test coverage across multiple scenarios. Furthermore, all defects encountered during execution were documented in a structured defect report, prioritized based on severity.

In summary, the project demonstrates how a well-structured automation testing approach can increase efficiency, ensure software reliability, and support the delivery of a robust web application. It demonstrates the behaviour-driven development (BDD) testing with reporting. It shows the use of extent reports, cucumber based reports and excel reader.

**TECH STACK :**

* **Programming Language :** Java
* **Automation Tool :** Selenium WebDriver
* **Testing Framework :** TestNG
* **(BDD) :** Cucumber(Gherkin feature files)
* **Build & Dependency :** Maven
* **IDE :** Eclipse
* **Version Control :** Git & GitHub
* **Reporting :** Extent Reports , Cucumber reports

**PURPOSE OF THE PROJECT :**

The primary purpose of this project is to automate the functional and regression testing of the Magicbricks web application. Manual testing is time-consuming, repetitive, and error-prone; therefore, automation was implemented to achieve the following goals:

* **Ensure core functionality works reliably**
* **Validate user journey** – simulate real user
* **Improve test efficiency** – execute tests faster, repeatedly, and across multiple scenarios without human error.
* **Generate systematic defect reports** – provide structured results for developers to fix issues efficiently**.**

**OUTCOMES :**

The project successfully delivered the following outcomes:

* **Improved Test Coverage** – All major functionalities of the Magicbricks web application (search, login, signup, filters, shortlisting, and invalid input validation , comparing localities, exploring localites) were covered.
* **Increased Reliability** – Automated scripts provided consistent, repeatable, and accurate test execution.
* **Early Defect Detection** – Issues in navigation, validation messages, and workflow inconsistencies were identified early and documented.
* **Reduced Manual Effort**
* **Reusable & Maintainable Test Suite** – The BDD approach with Cucumber feature files makes the test cases easily understandable.
* **End-to-End Validation** – Each user journey from homepage navigation to property shortlisting, exploring and comparing locality was validated.

In conclusion, the automation framework built for Magicbricks provided a robust testing solution, ensuring that the application meets quality standards, enhances customer experience, and supports continuous integration and delivery.

**PURPOSE OF EACH SCENARIO :**

**Scenario: Homepage Launch**

**TS\_MB\_01\_– Launch Homepage**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/Profile.feature>

* **Purpose:** To check whether the Magicbricks homepage URL opens correctly in the browser.
* **Outcome:** Website launched successfully; homepage is displayed with navigation and search bar.

**Scenario 2: Login with Valid Credentials**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/Profile.feature>

**TS\_MB\_02 – Valid Login**

* **Purpose:** To verify that a registered user can log in with correct phone number, captcha, and OTP.
* **Outcome:** User is redirected to the login/homepage successfully.

**TS\_MB\_02 – Valid Mobile Number**

* **Purpose:** To ensure the system accepts a valid registered mobile number.
* **Outcome:** No error message shown; OTP request initiated.

**TS\_MB\_02\_ – Captcha Validation**

* **Purpose:** To check that captcha entry is validated before OTP is generated.
* **Outcome:** No error message; page redirects to OTP entry page.

**TS\_MB\_02– OTP Entry**

* **Purpose:** To confirm login is successful with a valid OTP.
* **Outcome:** OTP accepted; login successful.

**Scenario 3: Login with Invalid / Missing Credentials**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/invalidlogin.feature>

**TS\_MB\_03– Invalid Mobile Number**

* **Purpose:** To ensure invalid or unregistered numbers cannot log in.
* **Outcome:** Error message “Invalid mobile number” displayed.

**TS\_MB\_03 – Blank Credentials**

* **Purpose:** To check mandatory field validation when no credentials are entered.
* **Outcome:** Error “Username/Password required” displayed; login denied.

**TS\_MB\_03– Missing Captcha**

* **Purpose:** To verify system prevents login when captcha is not entered.
* **Outcome:** Error “Invalid credentials” displayed; login fails.

**Scenario 4: Search Home**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/searchHome.feature>

**TS\_MB\_04** – Valid Location Search

* **Purpose:** To verify property results appear for a valid location (e.g., Chennai).
* **Outcome:** Correct list displayed and properties in Chennai are displayed.

**TS\_MB\_06** – Apply Residential Filter

* . **Purpose:** To shortlist houses/villas. Only house/villa properties displayed.
* **Outcome:** Only house/villa properties displayed

**TS\_MB\_07** – Apply 2,3 Bhk filter

* **Purpose:** To apply the 2,3 Bhk filters.
* **Outcome:** Correct range of properties are displayed

**TS\_MB\_08** – Apply Budget filter

* **Purpose:** To apply min and max filter (50 lac – 1.5cr)
* **Outcome:** Correct range of properties are displayed in the range.

**Scenario 5: Shortlist Property**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/shortlistProperty.feature>

**TS\_MB\_09** – Add Property to Shortlist

* **Purpose:** To verify property can be shortlisted.
* **Outcome:** Property added to shortlist successfully.

**TS\_MB\_10** – View Shortlisted Properties

* **Purpose:** To confirm shortlisted properties are viewable.
* **Outcome:** List of saved properties shown.

**Scenario 6: Invalid location**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/invalidexplorelocality.feature>

**TS\_MB\_05** – Blank Location Search

* **Purpose:** To check blank location validation.
* **Outcome:** Error message is displayed when no location is entered.

**TS\_MB\_05** – Invalid Location Search

* **Purpose:** To test invalid location input.
* **Outcome:** Error message appears and disappears quickly.

**Scenario 7: Location Explore**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/tipsandguides.feature>

**TS\_MB\_11**– Explore Valid Locality

* **Purpose:** To verify properties of a valid locality (e.g., Ambattur, Chennai).
* **Outcome:** Properties of Ambattur displayed and validated by rating

**Scenario 8: Invalid Locality**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/invalidexplorelocality.feature>

**TS\_MB\_12** – Invalid Locality Input

* **Purpose:** To check invalid locality handling.
* **Outcome:** Error displayed and validated.

**TS\_MB\_12** – Blank Locality Input

* **Purpose:** To verify blank input validation.
* **Outcome:** Error displayed and validated. “Please enter a valid locality.”

**Scenario 9: Compare Localities**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/tipscontinue.feature>

* **TS\_MB\_13–** Compare Two Valid Localities
* **Purpose:** To verify comparison page loads correctly.
* Outcome: Redirected and comparison validated with the help of rating.

**Scenario 10: Invalid Locality in Comparison**

Click here to see :

<https://github.com/jyothirmyeB/test_magicbricks/blob/482e2f247348f520b7aaa73bec836b416d8708b1/src/test/resources/Features/invalidcomparsion.feature>

**TS\_MB\_14** – Compare with One Invalid Locality

* **Purpose:** To ensure system rejects invalid input in comparison.
* **Outcome:** Error displayed and the validated.

**TS\_MB\_14**– Compare with Blank Locality

* **Purpose:** To verify error handling with missing input.
* **Outcome:** Error message is displayed

**TS\_MB\_14** – Compare with Two Invalid Localities

* **Purpose:** To test both inputs invalid.
* **Outcome:** Error displayed is displayed “Please enter a valid locality name”

**CONCLUSION :**

Automation testing of the Magicbricks web application ensured functionality, reliability, and improved efficiency.

1. Validated core modules like login, search, shortlist, profile, and locality comparison.
2. Covered both valid and invalid scenarios for better accuracy.
3. Reduced manual effort and saved time.
4. Identified issues like OTP and error message handling**.**