**Automation Framework experience:**

* **I** have **good exposure** in **designing** the **automation framework**. In automation framework, I have mostly used open source tools and technologies like **Selenium WebDriver**, **Cucumber, Java, Maven,** **Jenkins**, **Selenium Grid** and many more**.**
* In my previous experience. I have built various frameworks. The **latest framework** that I have designed is **Behaviour Driven Development framework**. This framework is **completely driven** by **feature files**.
* The **feature file** is a way of writing features based upon the acceptance criteria in **natural structured language** called “**Gherkin**” and each **scenario/test case** is named based upon on the its **test type** with the **annotations** **@Smoke**, **@Sanity** and **@Regression** and this gives an **advantage** of **different types executions** in **Jenkins pipeline**.
* Coming to **Step Definition implementation** as **every step** has **its own step definition** and **in this step** **definition** a **Selenium glue** will be **applied**.
* Coming to the **Selenium implementation** I have **followed** **Page Object design pattern** where each **Page object** is **stored** at **Page Classes**.
* Used **Java OOPs** concepts **in order** to **use** **selenium library.**
* This framework has **flexibility** to run on different **browsers** and best in handling synchronization problems as it has implicit and explicit waits.

The highlights of this framework:

* Handles **UI**, **API** and **mobile** tests.
* **Run** in **Jenkins pipeline.**
* **Generates reports**.
* **Handles alerts**, **popups**, **iframes.**
* **Logs using Log4j.**
* **Parallel testing.**
* **Screenshots** at the **test failure**.

**Challenges faced in automation** and **Solutions:**

* **Change of UI** or **UI** is **dynamic in nature**. (**Solution: using XPath**)
* Synchronization of the application and **application response time** vs **test execution**. (**Solution: using Implicit and Explicit waits**)
* Maintenance of automation framework if the **test cases are more than 20** (**Solution: using Page Object Model**).
* Generating enough **detailed level of reports** (**Solution: using User defined reports** and **Log4j for logging**).