Putting Together - Design the UI Test Framework

In this lesson, we will go through the components of a test automation framework.

For designing the UI test framework or any framework, we have to develop the below modules, which have already been discussed previously in the course.

- Driver Management – Managing your choice of browser/driver capabilities and related configuration
- Configuration Management – Setting configurations for your project which can be used across the framework code
- Loggers – Logging capability to your framework
- Test Data Management – Managing static test data as well as for multiple environments
- Utilities – Providing file read-write utilities, DB utilities and 3rd party connectors etc.
- POM design pattern – Designing Page Object Model for your application
- Creating an Abstract Base Page Class – Abstract base page class will be a superclass for all POM pages; it will have common functionalities used across the pages
- Creating an Abstract Base Test class – Abstract base test class will be a superclass for all test files; it will have driver initialization, navigating to launch Urls and will have common functionalities used across the test classes
- Testing framework – Using TestNG , Junit etc.
- Reporting – Creating TestNG reports, Allure reports, and etc.; it will help to understand the errors and debugging them

- Build Management – Providing support to compile and run code, e.g., via Maven or Gradle

---

In this lesson, we learned about the design constructs and their usage. In the next lesson, we will discuss the best practices while writing an automated test.