Framework Architecture   
Selenium WebDriver: For cross-browser UI automation.  
TestNG or JUnit: For parallel test execution and reporting.  
Apache POI: For reading/writing data from Excel sheets (Data-Driven).  
Page Object Model (POM): For separating UI interactions.  
Maven or Gradle: For dependency and build management.  
Log4j: For logging.  
ExtentReports/Allure: For rich test execution reports.  
Jenkins: CI/CD (cron jobs + pipeline triggers)  
IRetryAnalyzer: To Rerun Failed Tests via testng-failed.xml   
  
HybridAutomationFramework/  
│  
├── src/  
│  └── main/  
│    └── java/  
│      ├── base/        # Browser setup, driver init  
│      ├── pages/        # (locators, action methods) for each web page  
│      ├── tests/        # TestNG test classes  
│      ├── utils/        # Excel reader, config reader, waits, etc.  
│      └── retry/        # Retry logic for failed tests  
│  
├── testdata/  
│  └── LoginData.xlsx        # Excel sheet with input data  
│  
├── config/  
│  └── [**config.properties**](http://config.properties/)

# URL, browser, timeout, etc.  
│  
├── testng.xml            # Test suite runner config  
├── pom.xml             # Maven dependencies & plugins  
├── reports/             # Extent or Allure reports  
└── logs/              # Log4j output  
  
The Hybrid Framework combines the best of multiple methodologies ensuring flexibility, reusability, and maintainability. It allows teams to leverage the strengths of various approaches, providing a robust solution for efficient and scalable automation testing.