**Best Practices for Framework Design:**

**1. Avoid Hard-Coding Test Data:**

Ensure that test data is injected from Tests rather than being hard coded within test methods. This enhances the generic code maintainability

**2. Externalize Test Data:**

Feed test data from external files to maintain isolation between the test logic and data sources. This practice supports easier maintenance and scalability.

**3. Implement Page Object Model (POM):**

Apply the Page Object Design Pattern to separate page locators and actions from test files. This improves code reusability, readability, and maintainability.

**4. Centralize Reusable Code:**

Use common utilities (utils.py) or configuration files (conf.py) to store reusable code across the framework. This avoids redundancy and promotes consistency.

**5. Define Global Environment variables:**

Use global configuration values to drive the execution using core parameters of the test without changing the code.

**6. Apply Grouping/Tags to run targeted tests**

**7. Running Tests in Parallel Mode.**

**8. Generate HTML Reports:**

Integrate HTML reporting tools to create detailed test execution reports. These reports help analyse overall test coverage, success rates, and failures.

**9. Capture Logs & Screenshots:**

Configure your framework to automatically generate logs and capture screenshots for failed test cases. This helps in debugging and analysis.

**10. CI/CD Integration with Jenkins:**

Integrate the framework with Continuous Integration/Continuous Deployment (CI/CD) tools like Jenkins. Automate and schedule nightly builds to run tests regularly and ensure consistent feedback loops.

**11. Integrate Cucumber BDD:**

Incorporate Cucumber for Behavior Driven Development (BDD) to write Gherkin syntax-based feature files. This improves test readability and collaboration with non-technical stakeholders.