

AOOUT Project



- Goal: Build an end-to-end automation framework for Sauce Demo web app.
- Automate critical functionalities using Selenium
 + TestNG.
- Enable CI/CD pipeline with Jenkins for automatic test execution.
- Use GitHub for version control & collaboration.
- Track requirements, test cases, and bugs using
 Jira + Zephyr Board.



Core Automation & Framework

Java, Selenium, TestNG, JUnit, Maven

Collaboration & Project

Git/GitHub, JIRA + Zephyr

CI/CD

Jenkins

Development Environment

Eclipse IDE

Automation approach



Page Object Model separates page structure and test logic for maintainability

```
public class CartPage extends BasePage {
   // -----
   // o Constructor
   public CartPage(WebDriver driver) {
       super(driver);
       PageFactory.initElements(driver, this);
   // -----
   // o Dynamic Locators
   private static final String REMOVE_BTN_BY_PRODUCT_NAME =
           "//div[@class='cart item' and .//div[@class='inventory item name' and text()='%s']]
   private static final String PRODUCT TITLE BY NAME =
           "//div[@class='cart_item']//div[@class='inventory_item_name' and text()='%s']";
   // o Web Elements
   @FindBy(xpath = "//button[text()='Remove']")
   private List<WebElement> removeButtonElements;
   @FindBy(xpath = "//div[@class='inventory_item_name']")
   private List<WebElement> productTitleElements;
   @FindBy(xpath = "//button[text()='Continue Shopping']")
   private WebElement continueButtonElement;
   @FindBy(xpath = "//button[text()='Checkout']")
   private WebElement checkoutButtonElement;
   @FindBy(xpath = "//span[@class='title' and text()='Your Cart']")
   private WebElement cartTitle;
   // o Page Validations
```

Data Driven Testing use excel to run tests with multiple inputs

```
public static Object[][] getTestData(String filePath, String sheetName, String groupFilter) {
   List<Object[]> data = new ArrayList<>();
   DataFormatter formatter = new DataFormatter(); // handles all cell types
   try (FileInputStream fis = new FileInputStream(filePath);
        Workbook workbook = new XSSFWorkbook(fis)) {
       Sheet sheet = workbook.getSheet(sheetName);
       int rows = sheet.getPhysicalNumberOfRows();
       for (int i = 1; i < rows; i++) { // skip header
           Row row = sheet.getRow(i);
           if (row == null) continue;
           String testCaseId = formatter.formatCellValue(row.getCell(0));
           String username = formatter.formatCellValue(row.getCell(1));
           String password = formatter.formatCellValue(row.getCell(2));
           String expected = formatter.formatCellValue(row.getCell(3));
           String testType = formatter.formatCellValue(row.getCell(4));
           String groups = formatter.formatCellValue(row.getCell(5));
           String desc
                            = formatter.formatCellValue(row.getCell(6));
           // Filter by group (matches partial string like "sanity" inside "smoke, sanity, reg
           if (groups != null && !groups.isEmpty() &&
               groups.toLowerCase().contains(groupFilter.toLowerCase())) {
               data.add(new Object[]{
                   testCaseId, username, password, expected, testType, groups, desc
   } catch (Exception e) {
       e.printStackTrace();
   return data.toArray(new Object[0][0]);
```

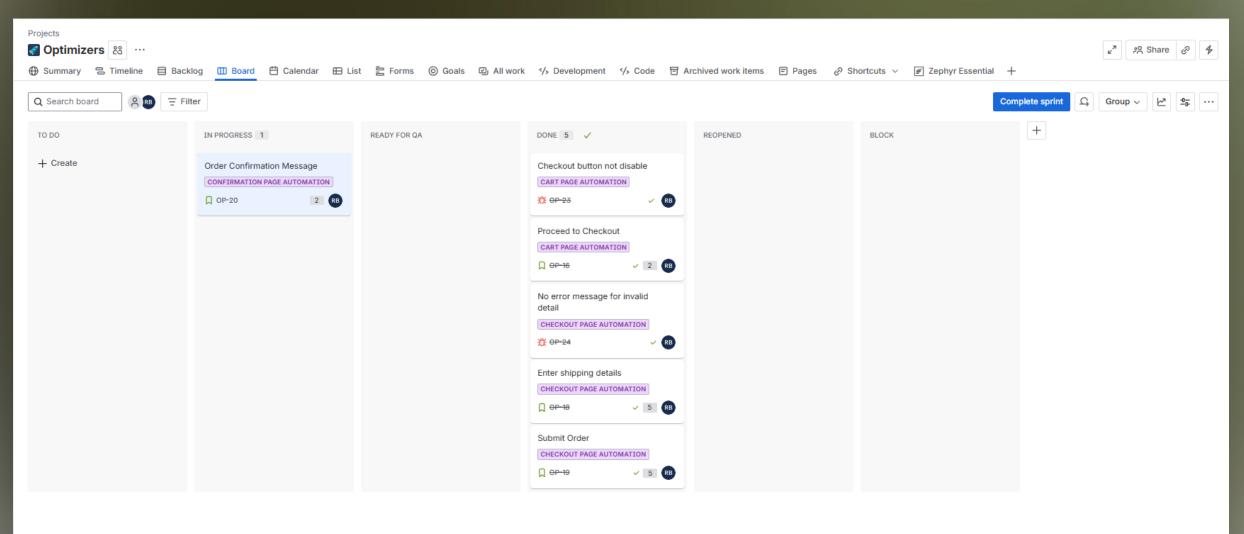
Cross Browser Testing validate application on Chrome, Firefox, etc.

```
// ===========
// o Test Setup / Teardown
// -----
* Setup WebDriver before any test class execution.
* @param browserString Browser name from testng.xml
@Parameters({ "browser" })
@BeforeClass(alwaysRun = true)
public void setUp(String browserString) {
   LogConfig.initLogs();
   Map<String, Object> prefs = new HashMap<>();
   prefs.put("credentials enable service", false);
   prefs.put("profile.password manager enabled", false);
   prefs.put("profile.password manager leak detection", false);
   log.info("======"");
   log.info("====== Test Execution Started =======");
   log.info("-----");
   log.info("Selected Browser: " + browserString);
    switch (browserString.toLowerCase()) {
       case "chrome":
           WebDriverManager.chromedriver().setup();
           ChromeOptions chromeOptions = new ChromeOptions();
           chromeOptions.setExperimentalOption("prefs", prefs);
           driver = new ChromeDriver(chromeOptions);
           break;
       case "firefox":
           WebDriverManager.firefoxdriver().setup();
           FirefoxProfile profile = new FirefoxProfile();
           profile.setPreference("signon.rememberSignons", false);
           profile.setPreference("signon.autofillForms", false);
           profile.setPreference("signon.generation.enabled", false);
           FirefoxOptions options = new FirefoxOptions();
           options.setProfile(profile);
           driver = new FirefoxDriver(options);
           break;
       case "edge":
           WebDriverManager.edgedriver().setup();
           EdgeOptions edgeOptions = new EdgeOptions();
           edgeOptions.setExperimentalOption("prefs", prefs);
           driver = new FdgeDriver(edgeOntions):
```

Board

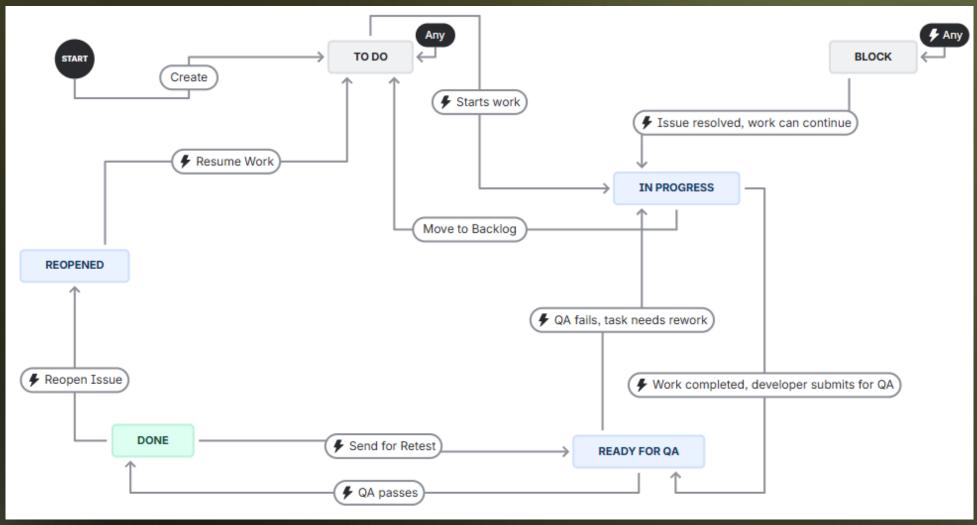


Board: Visual representation of tasks organized by status, enabling easy tracking of team work.



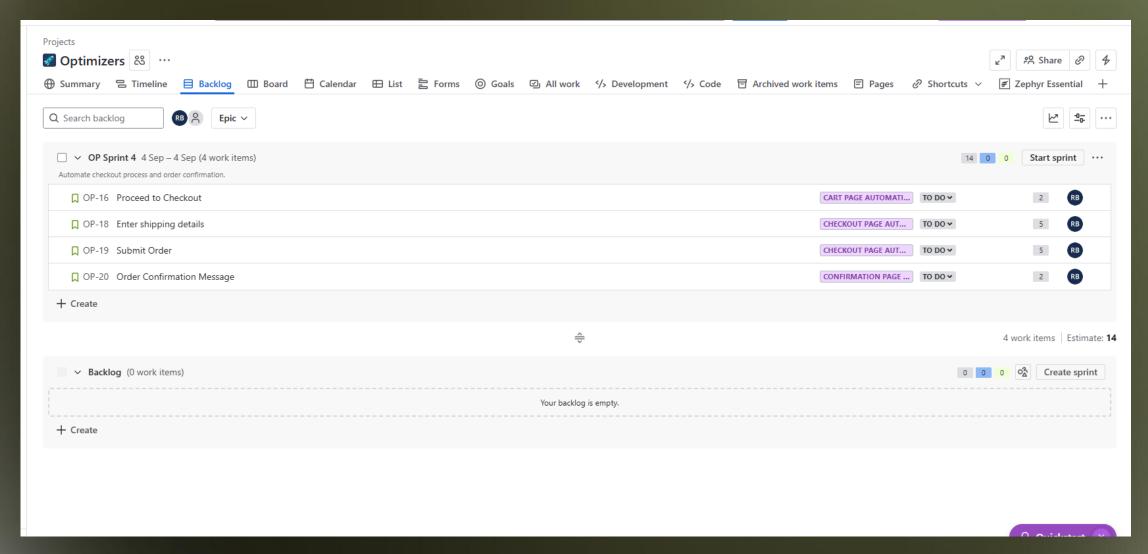


Workflow – Shows the path each story, task, or bug follows (e.g., To Do \rightarrow In Progress \rightarrow Review \rightarrow Done).



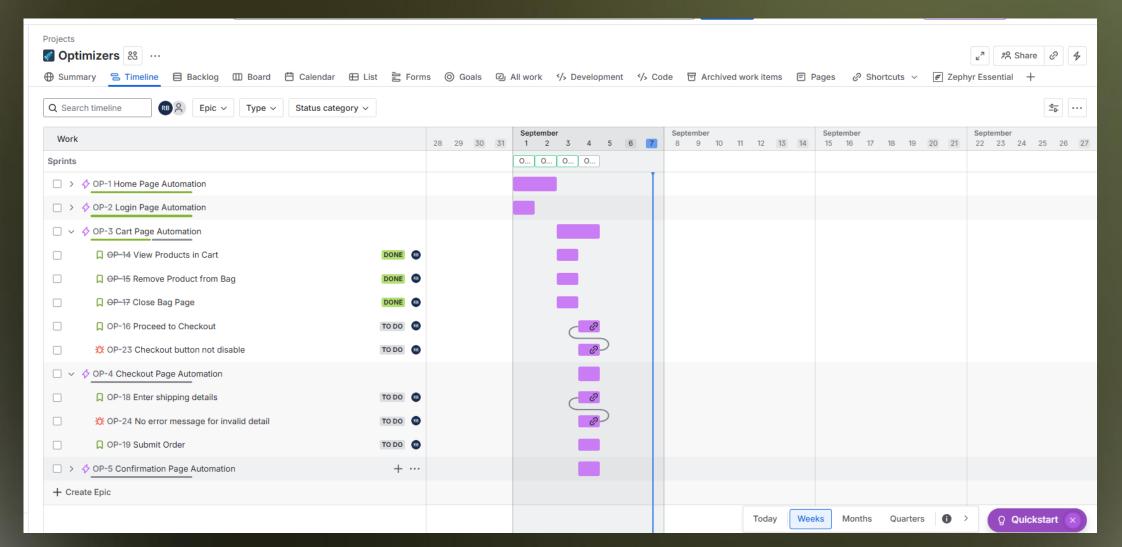


Backlog – List of all pending user stories and tasks that are yet to be planned into sprints.





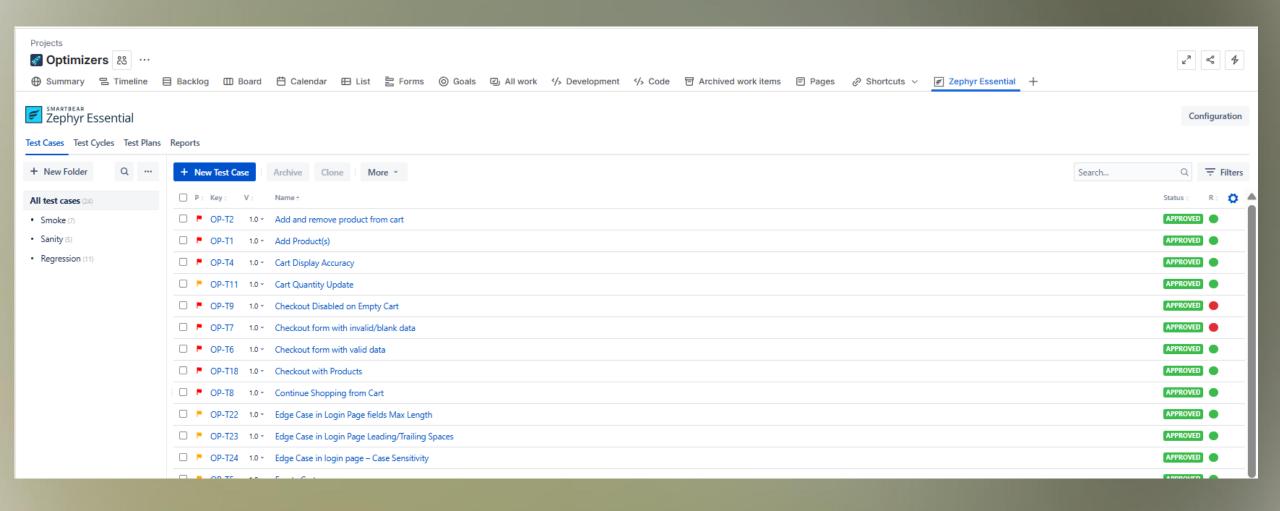
Timeline – Displays epics and stories along a timeline, helping track progress and deadlines clearly.



Zephyr Board

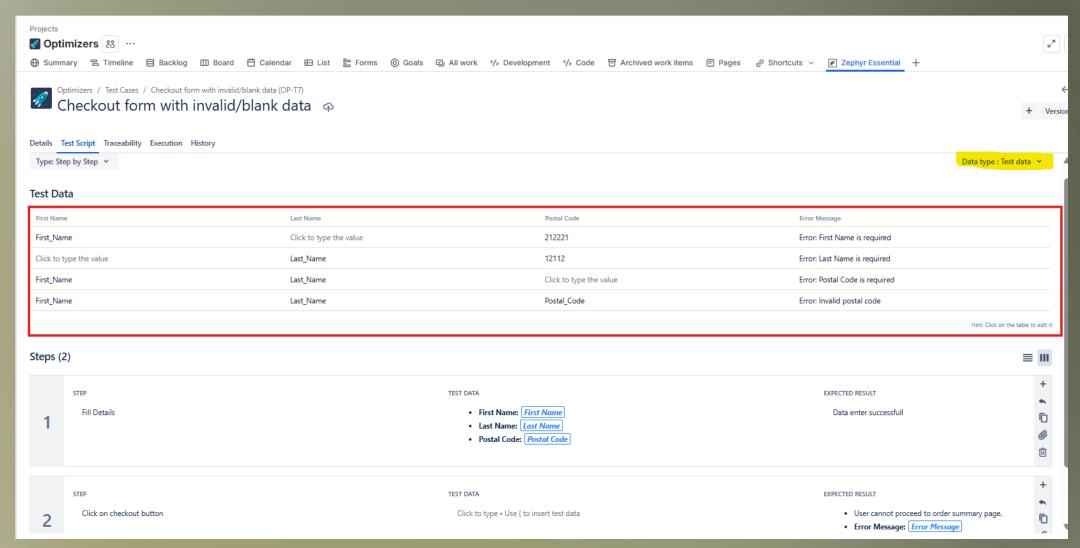


Test cases: Repository showing all created test cases.



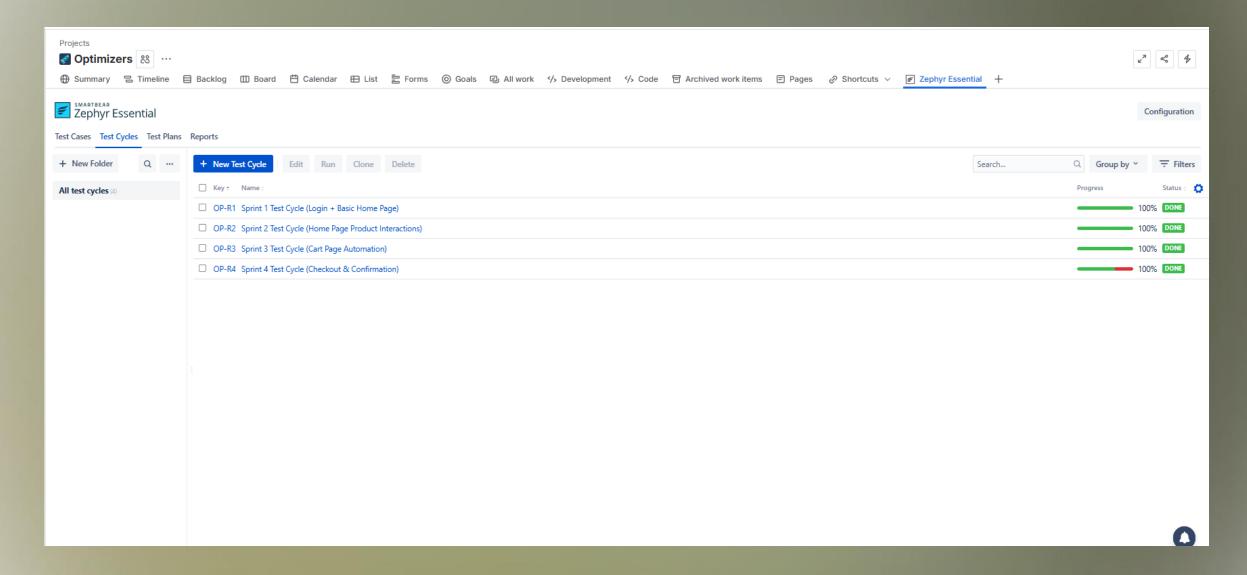


Each test case is written with clear steps. **Test Data** field is used to provide multiple inputs. Same test case can run with different sets of data (data-driven).



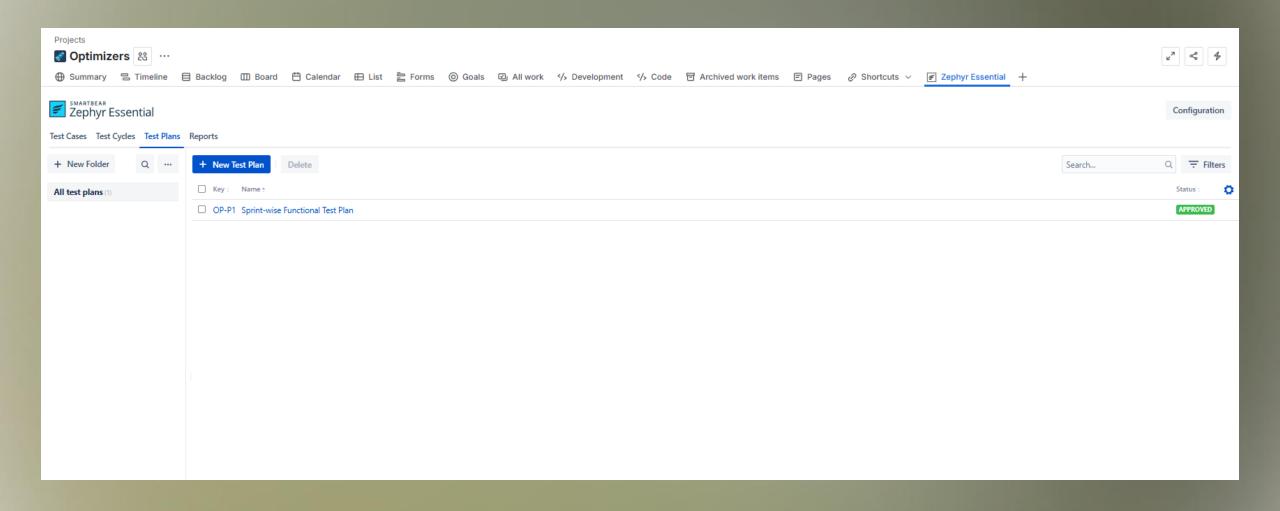


Test Cycles – Collection of test cases grouped per sprint.



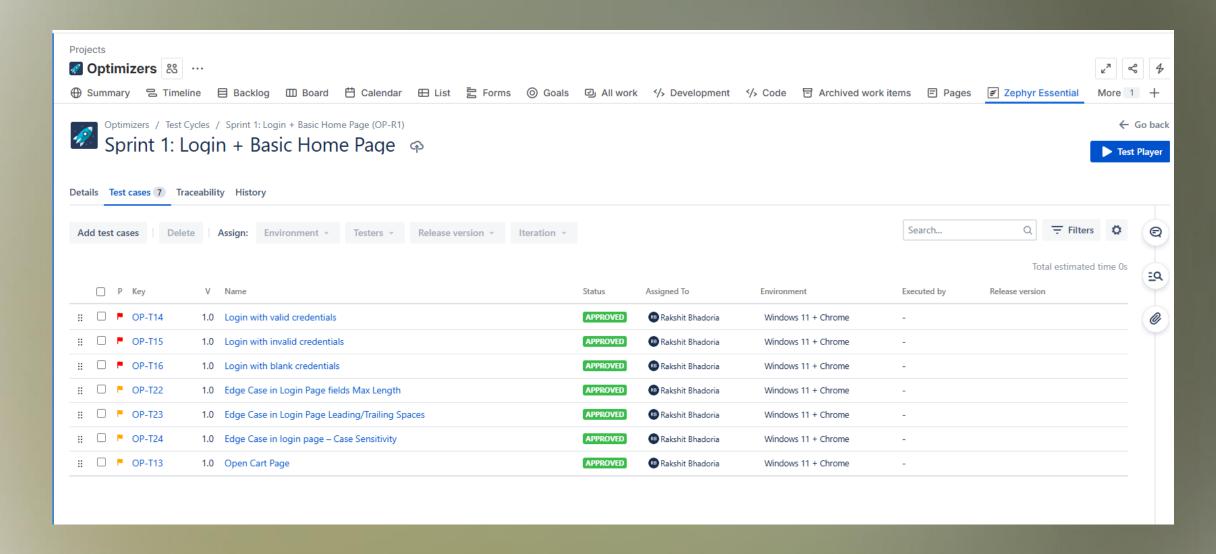


Test Plan – High-level plan combining multiple test cycles.



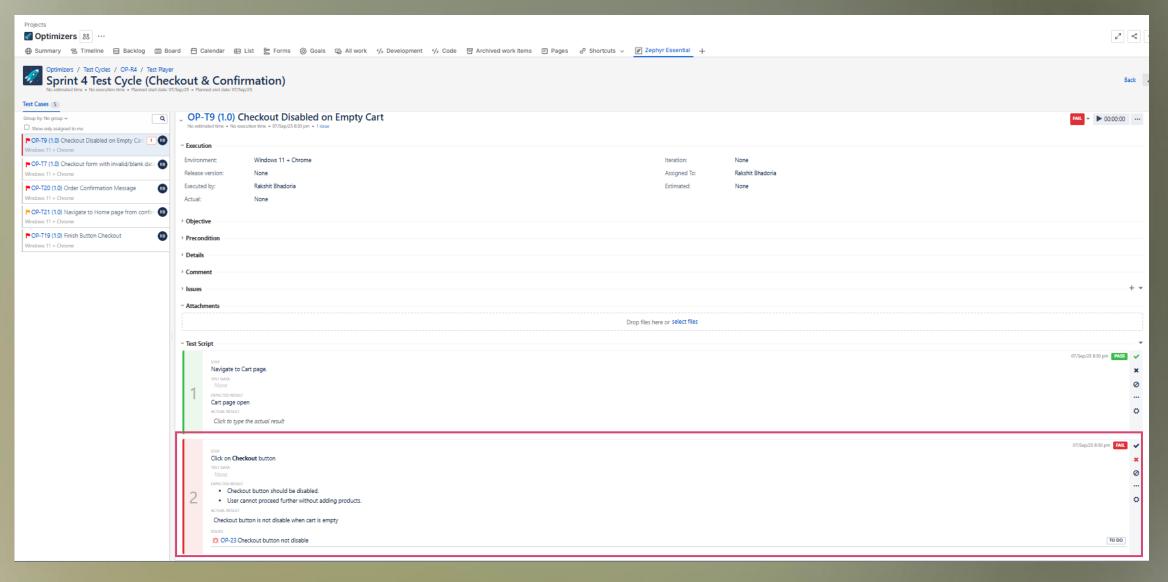


Test Cycle View – Shows one cycle with all test cases inside the test cycle



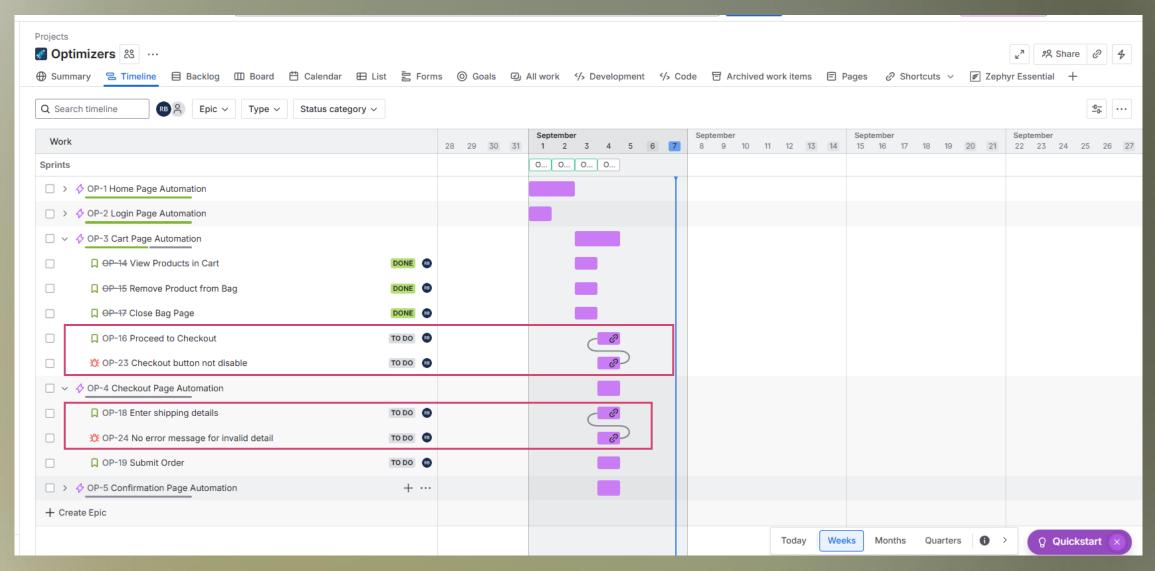


Failed Test Handling – When a test fails, we can create an issue/bug at that failed step.





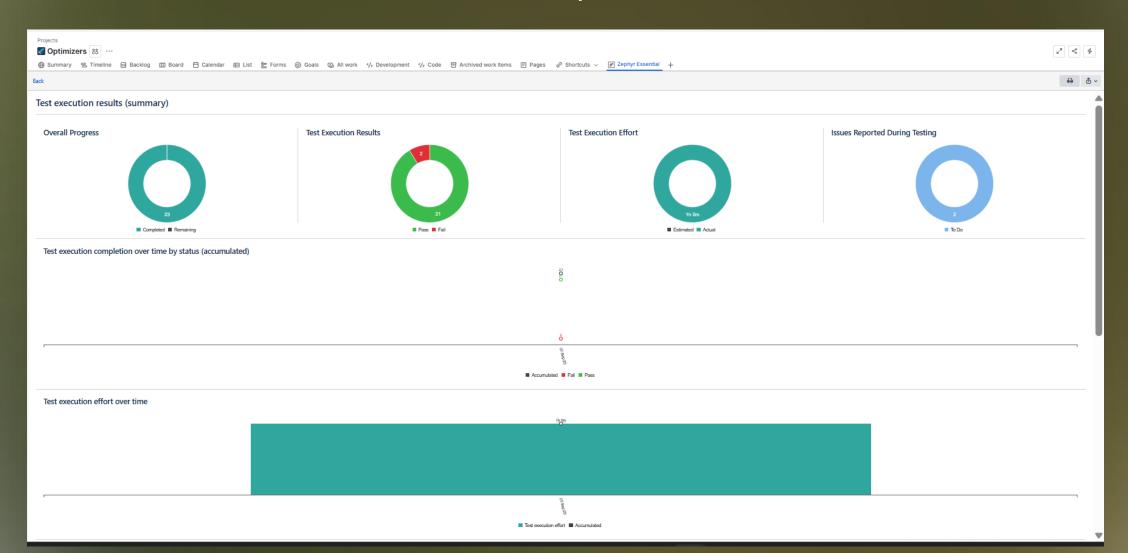
Issue Linking – Reported issues are linked to the respective user story they block.



Test Report



Zephyr Test Report – Provides execution status of test cases (Pass/Fail/Blocked), linked with user stories for better traceability.





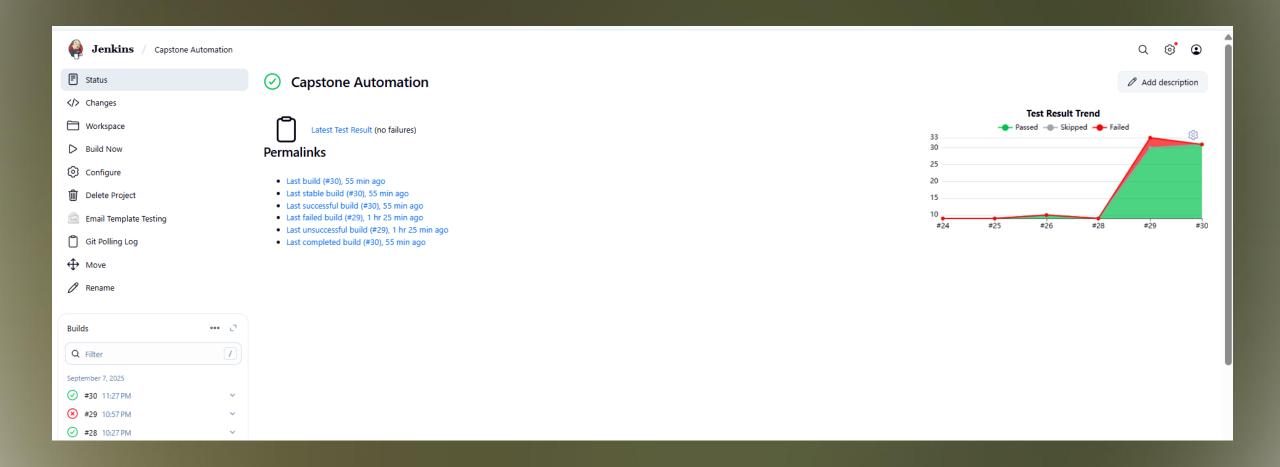
Test Report – Provides execution status of test cases (Pass/Fail/Blocked), linked with



Jenkins

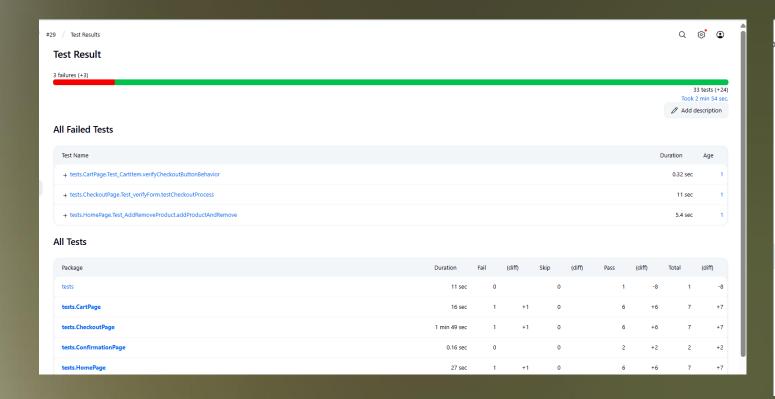


Job View (Capstone Automation) – Displays build history with a graph showing pass/fail trends for each build.





Post-Build Action (JUnit Reports) – Publishes detailed test results after every build for easy review.

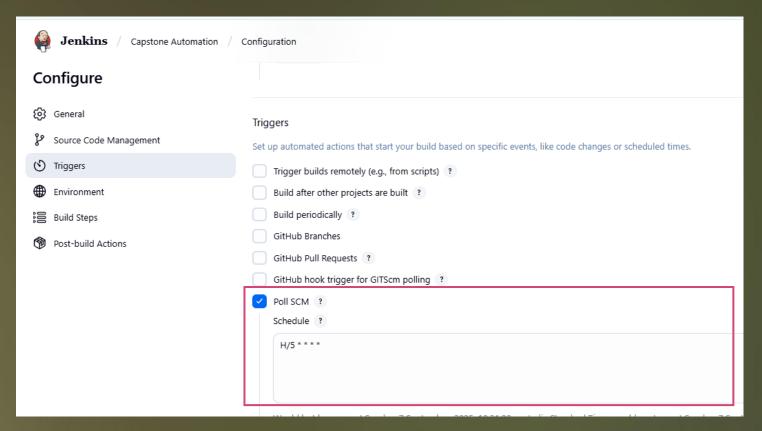


1 / 0	Configuration
	Post-build Actions
	Define what happens after a build completes, like sending notifications, archiving artifact
	■ Publish JUnit test result report ?
	Test report XMLs
	Fileset 'includes' setting that specifies the generated raw XML report files, such as 'my
	SauceDemoAutomation/target/surefire-reports/junitreports/*.xml
	Test output retention ?
	All tests
	keep all the properties
	Keep original test names, even when reporting from multiple stages



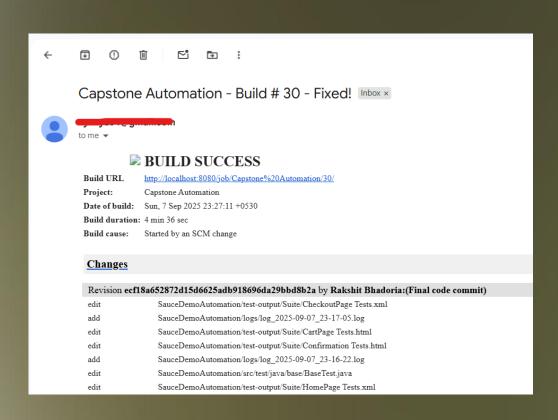
GitHub Trigger – Automatically starts a build whenever a new commit is pushed to the repository.

Poll SCM (Source Code Management)- Poll SCM is a Jenkins option that periodically checks for changes in the repo instead of building on a fixed schedule. If changes exist, it triggers the pipeline automatically.





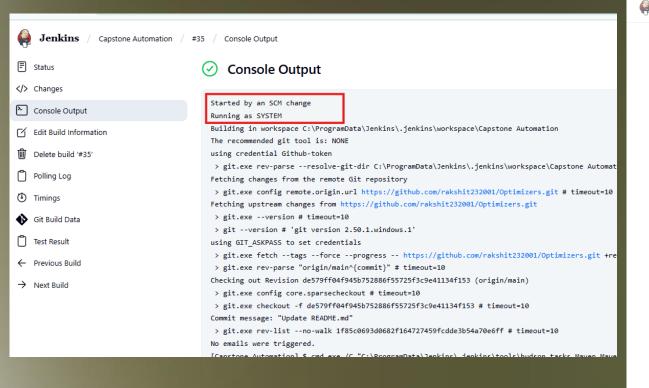
Email Notification – Sends an email after each build with logs attached, indicating success or failure.



Jenkins / Capstone Automation / Configuration		
Configure	■ Editable Email Notification ?	
General Source Code Management Triggers	Allows the user to disable the publisher, while maintaining the settings Disable Extended Email Publisher ? Project From	
Environment		
Build Steps Post-build Actions	Project Recipient List ? Comma-separated list of email address that should receive notifications for this project.	
	SDEFAULT_RECIPIENTS Project Reply-To List ? Comma-separated list of email address that should be in the Reply-To header for this project.	
	\$DEFAULT_REPLYTO	
	Content Type ?	
	Default Content Type	
	Default Subject ?	
	\$DEFAULT_SUBJECT	



Console Output of a build automatically trigger

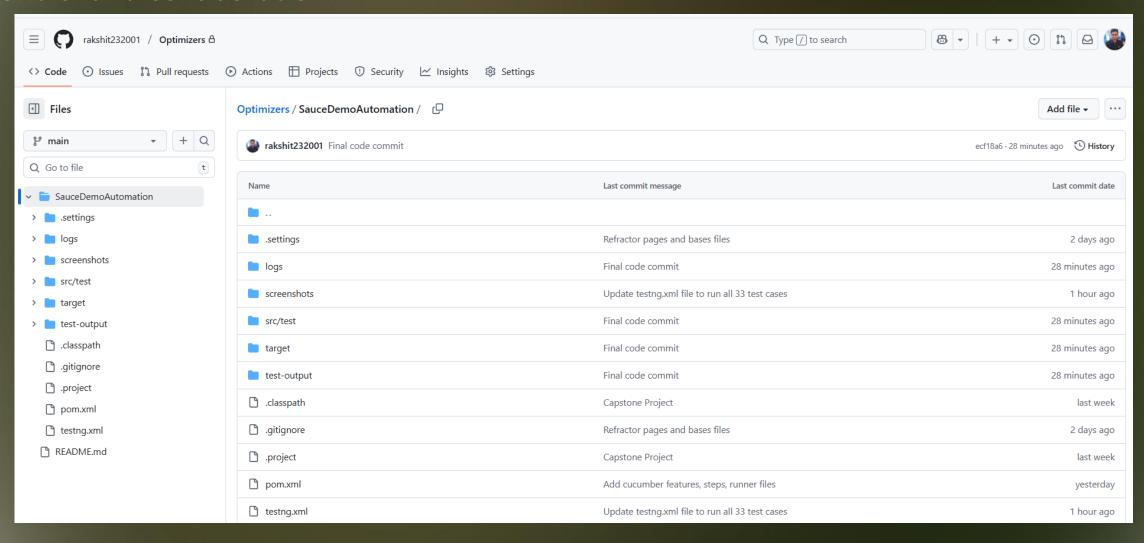


```
Jenkins / Capstone Automation / #35 / Console Output
                                             2025-09-08 13:02:48 [INFO] UT115.lestListener - ===== SIAKIING IESI: VerityUrderContirmationmessage =====
                                             2025-09-08 13:02:48 [INFO] base.BaseTest - Verifying order confirmation message is displayed.
                                             2025-09-08 13:02:48 [INFO] pages.ConfirmationPage - Verifying if order confirmation message is displayed.
                                             2025-09-08 13:02:48 [INFO] utils.TestListener - ? TEST PASSED: verifyOrderConfirmationMessage
                                             2025-09-08 13:02:48 [INFO] utils.TestListener - ===== STARTING TEST: verifyBackHomeNavigation =====
                                             2025-09-08 13:02:48 [INFO] base.BaseTest - Clicking Back Home button.
                                             2025-09-08 13:02:48 [INFO] pages.ConfirmationPage - Clicking on 'Back Home' button.
                                             2025-09-08 13:02:48 [INFO] pages.ConfirmationPage - Clicked on: Back Home button
                                             2025-09-08 13:02:48 [INFO] utils.TestListener - ? TEST PASSED: verifyBackHomeNavigation
                                             2025-09-08 13:02:49 [INFO] base.BaseTest - Browser closed.
                                             2025-09-08 13:02:49 [INFO] base.BaseTest - ==== Test Execution Finished =====
                                             2025-09-08 13:02:49 [INFO] utils.TestListener - ===== FINISHED TEST SUITE: Confirmation Tests =====
                                             [INFO] Tests run: 31, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 269.8 s -- in TestSuite
                                             [INFO] Results:
                                             [INFO]
                                             [INFO] Tests run: 31, Failures: 0, Errors: 0, Skipped: 0
                                             [TNFO] BUTLD SUCCESS
                                             [INFO] Total time: 04:53 min
                                             [INFO] Finished at: 2025-09-08T13:02:50+05:30
                                             Recording test results
                                             [Checks API] No suitable checks publisher found.
                                             Email was triggered for: Always
                                             Sending email for trigger: Always
                                             Not sending mail to unregistered user (id: noreply, email: noreply@github.com) because your SCM claimed this was associated with a user
                                             changes in your SCM plugin
                                             Sending email to: rakshitbhadoria5@gmail.com
                                             Finished: SUCCESS
```

ECIDSE GitHUO



GitHub Repository – Shows all project files and folders stored online for version control and collaboration.





Eclipse IDE – Displays the local folder structure and files in the workspace for easy development and debugging.

```
Edit Source Refactor Navigate Search Project Run Window Help

☑ CartPage.java ×
                                package pages;

✓ I SauceDemoAutomation [Or
                              3⊖ import java.util.List;
 > 乃 src/main/java
   src/main/resources
                                import org.junit.Assert;
                                import org.openqa.selenium.By;
 import org.openga.selenium.WebDriver;
    > 🔠 base
                                import org.openqa.selenium.WebElement;
    > A features
                                import org.openqa.selenium.support.FindBy;
    > 击 pages
                             10 import org.openqa.selenium.support.PageFactory;
                                import org.openga.selenium.support.ui.ExpectedConditions;
   > 📠 runners
                             12
   > # stepDefinitions
                             13 import base.BasePage;
   > 🏭 tests
                             14
   > # tests.CartPage
                             15\(\theta\) /**
                                  * Page Object representing the Cart Page in SauceDemo.
   tests.CheckoutPage
                                  * Contains methods to interact with cart elements and perform validations.
    > # tests.ConfirmationPag
                             18
   > # tests.HomePage
                             19
                                  * @Author: Rakshit Bhadoria
    > # tests.LoginPage
                                  * @version: Sept 2025
                             21
    > 📠 utils
                             22
                                 public class CartPage extends BasePage {
      23
  > R src/test/resources
                             24
  JRE System Library [JavaS
                             25
                             26
                                    // ===========
  > Maven Dependencies
                             27⊝
                                    public CartPage(WebDriver driver) {
 > 🚰 logs
                             28
                                        super(driver);
  >  a screenshots
                             29
                                        PageFactory.initElements(driver, this);
  > Am src
                             30
                             31
  > 🗁 target
                             32
                                    // -----
  > (a) test-output
                             33
                                     // ⋄ Dynamic Locators
    mx.mog
   testng.xm
```

EMO

Thank You!