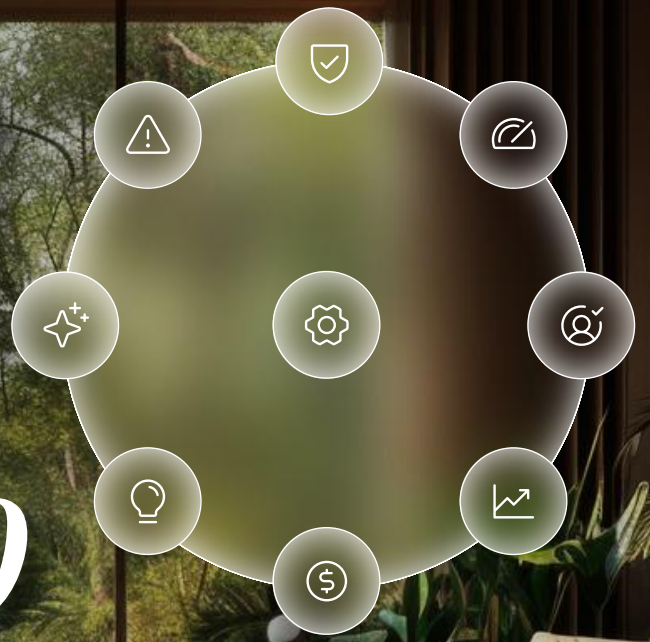
9 Aug 2025

Capstone Project

Sauce Demo



Presented by:
Rakshit Bhadoria (Java Selenium Batch – 3)

Table of *Contents*

9 Aug 2025

About Project

Automation approach

Zephyr Board

Jenkins

End

Tools Used

Jira Board

Test Report

Eclipse & GitHub

Presented by:
Rakshit Bhadoria (Java Selenium Batch – 3)

About 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**.

Tools Used

Tools Used

Core Automation & Framework

Java, Selenium,
TestNG, JUnit, Maven

Collaboration & Project

Git/GitHub,
JIRA + Zephyr

CI/CD

Jenkins

Development Environment

Eclipse IDE

Automation approach

Automation approach

Page Object Model

separates page structure and test logic for maintainability

```
public class CartPage extends BasePage {

    // ==============================
    // ⬦ Constructor
    // ==============================
    public CartPage(WebDriver driver) {
        super(driver);
        PageFactory.initElements(driver, this);
    }

    // ==============================
    // ⬦ 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']";

    // ==============================
    // ⬦ 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;

    // ==============================
    // ⬦ 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

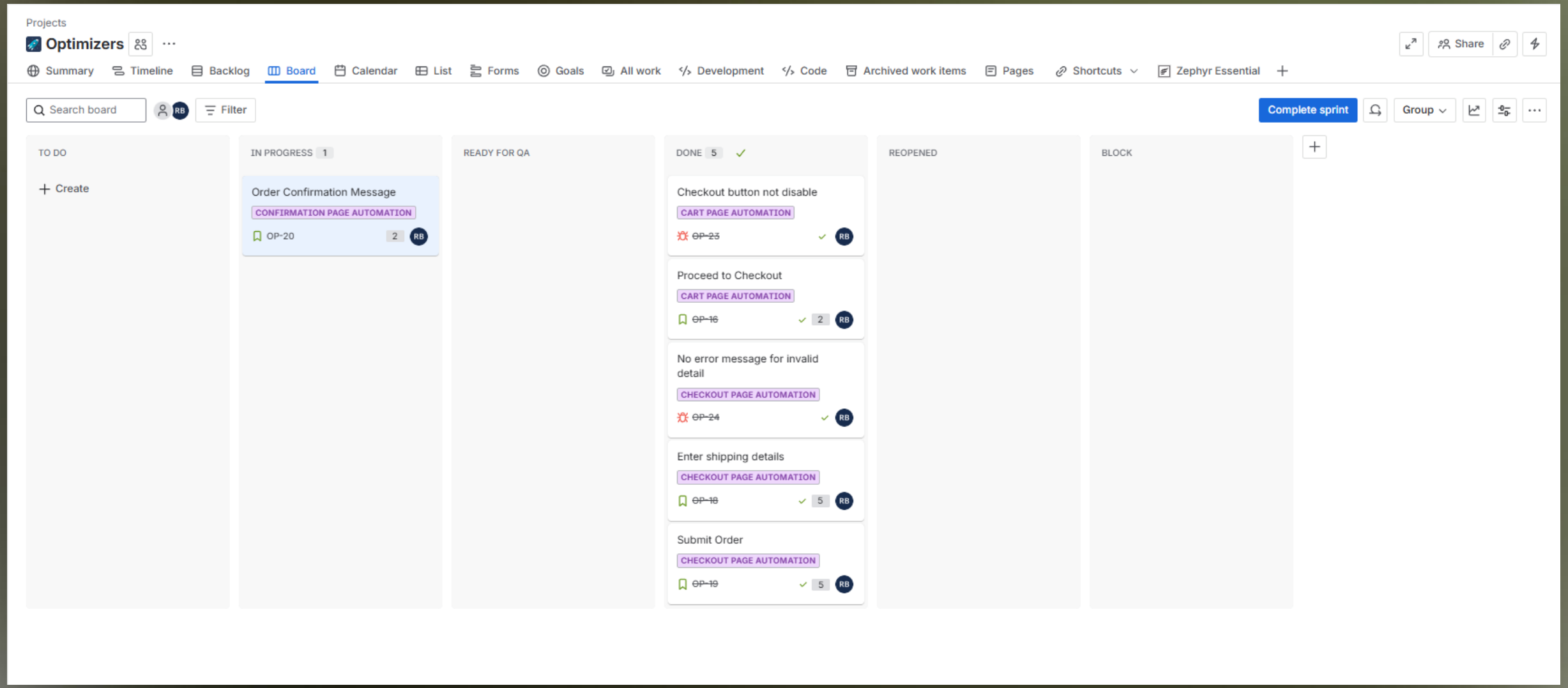
validate application on Chrome, Firefox, etc.

```
// ==============================
// ⬦ 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 EdgeDriver(edgeOptions);
```

Jira Board

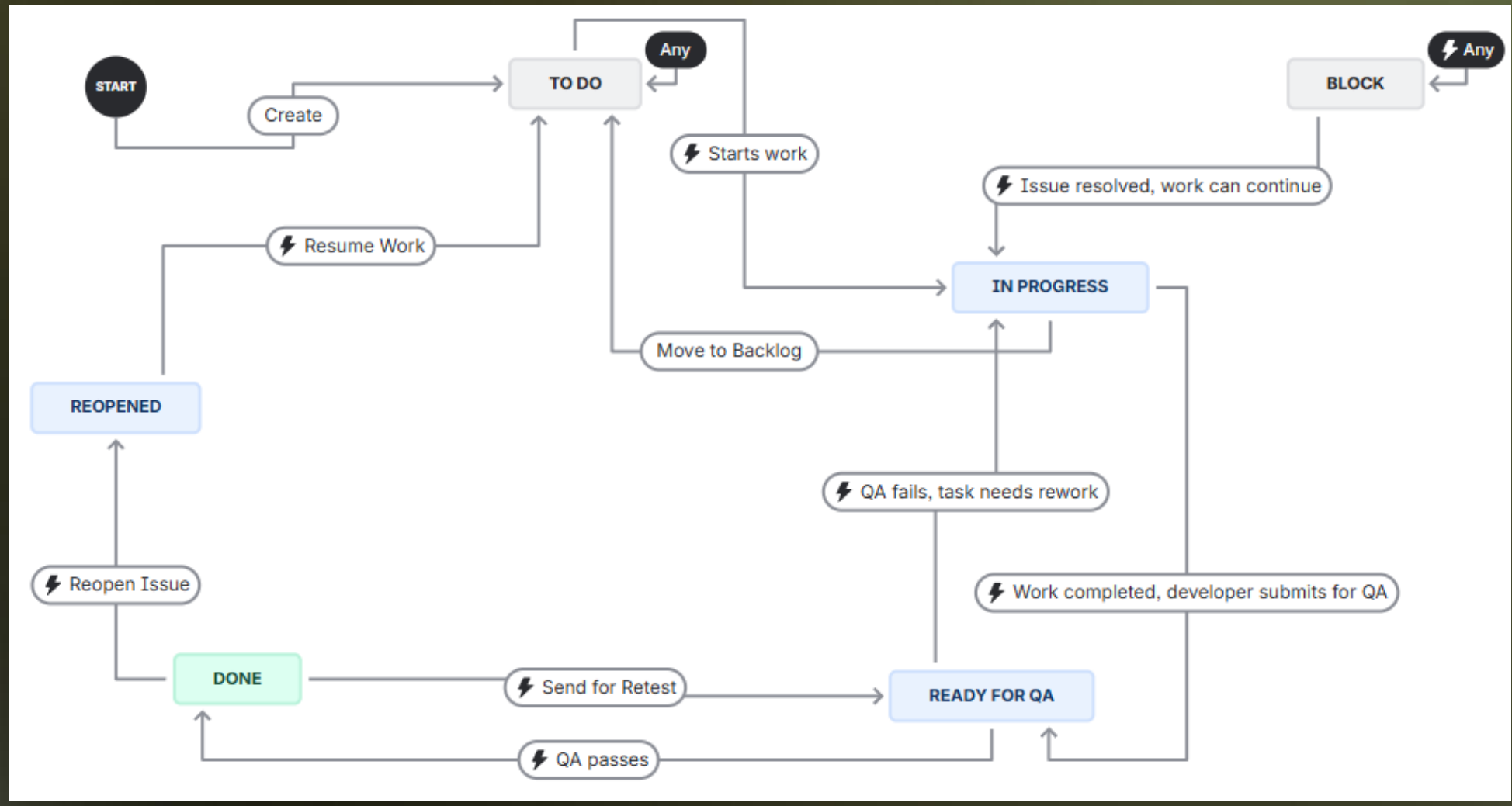
Link for Jira

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



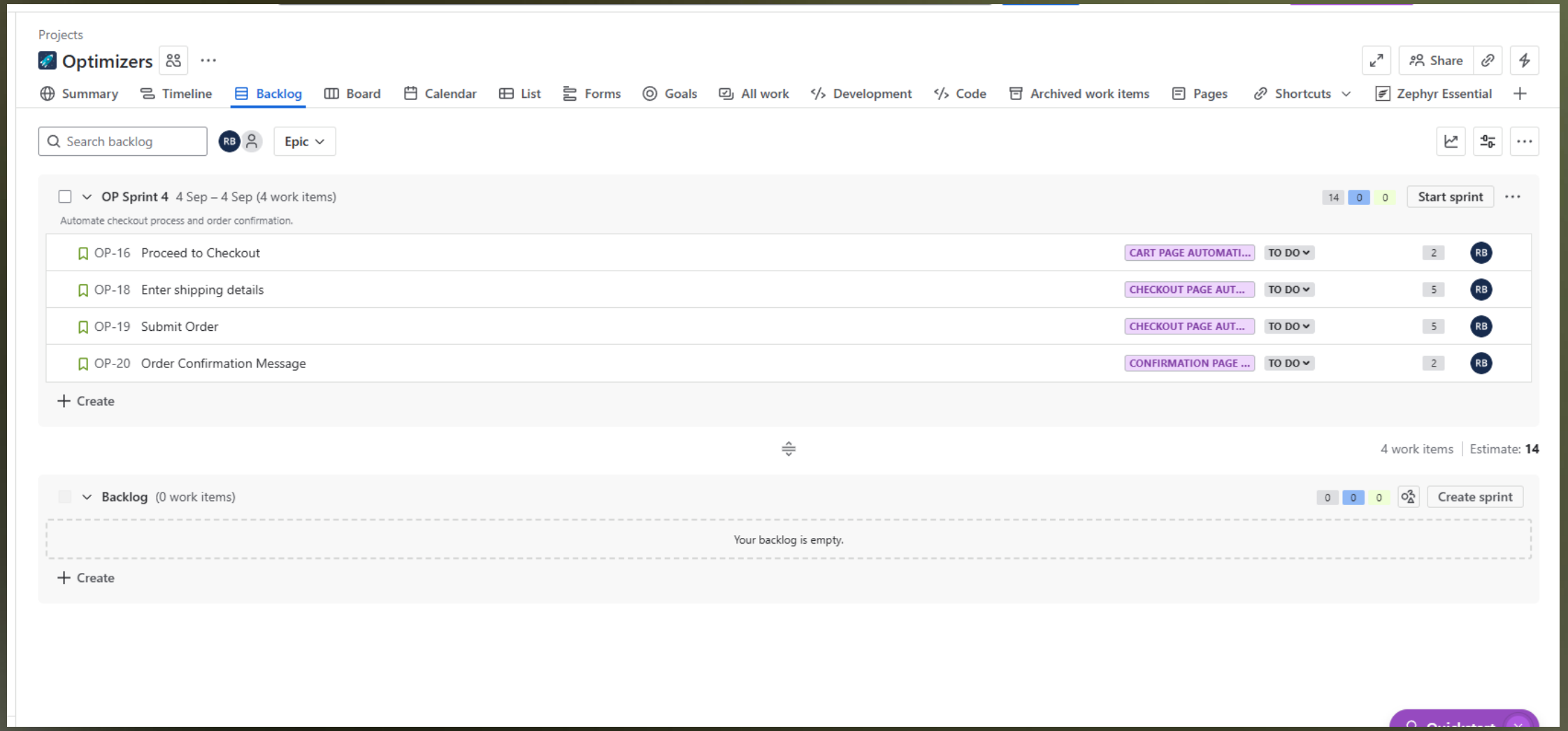
Link for Jira

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



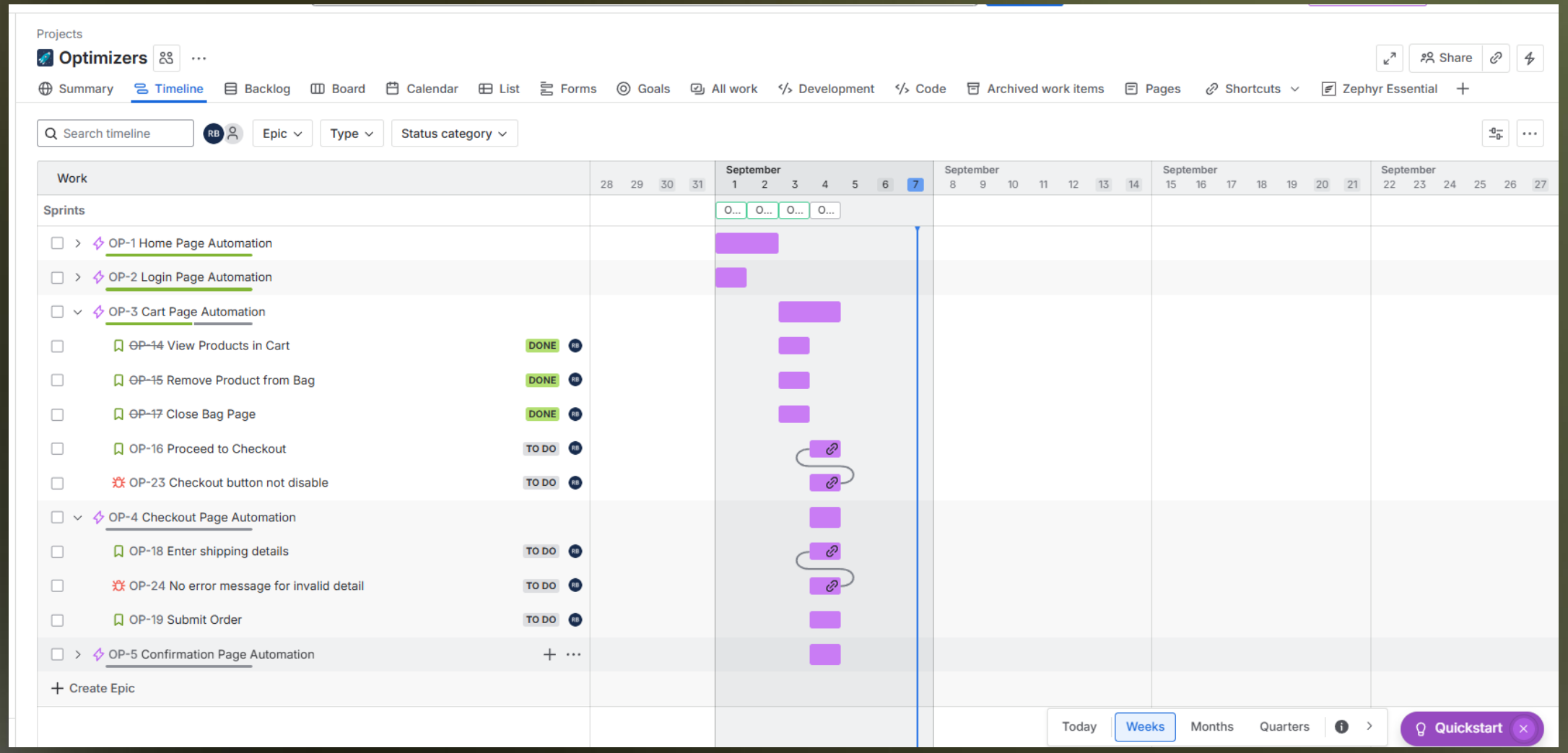
Link for Jira

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



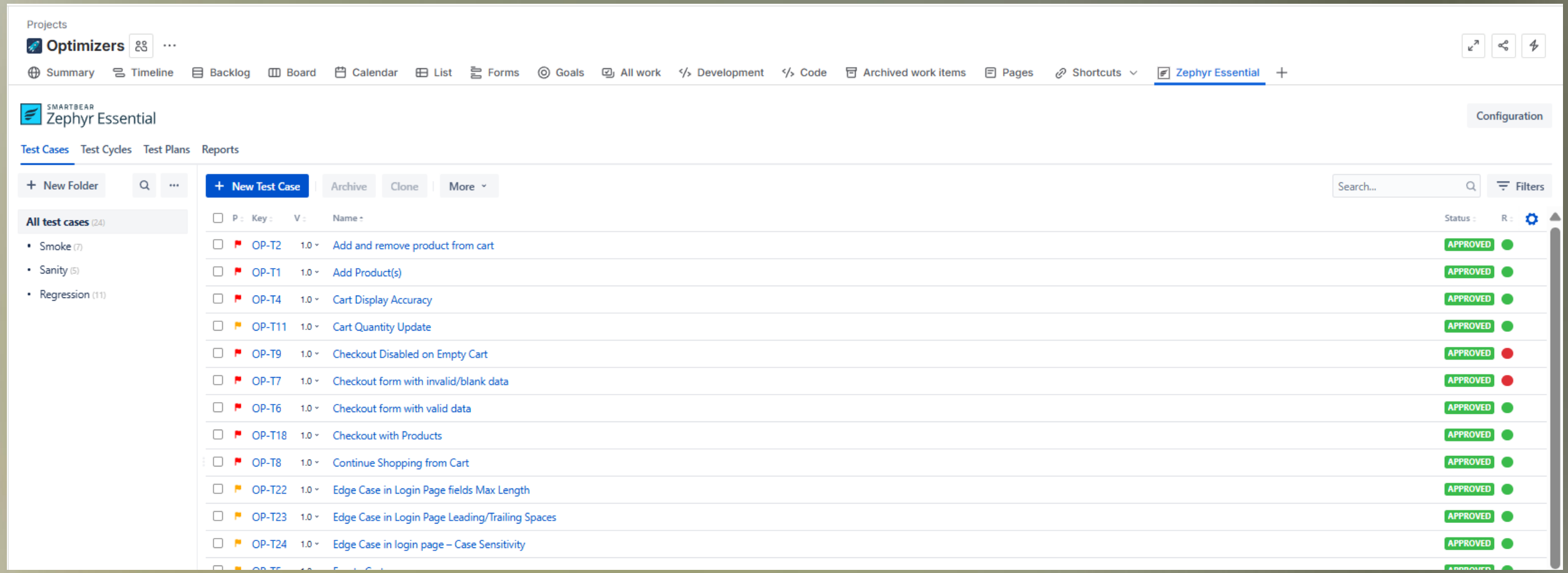
Link for Jira

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

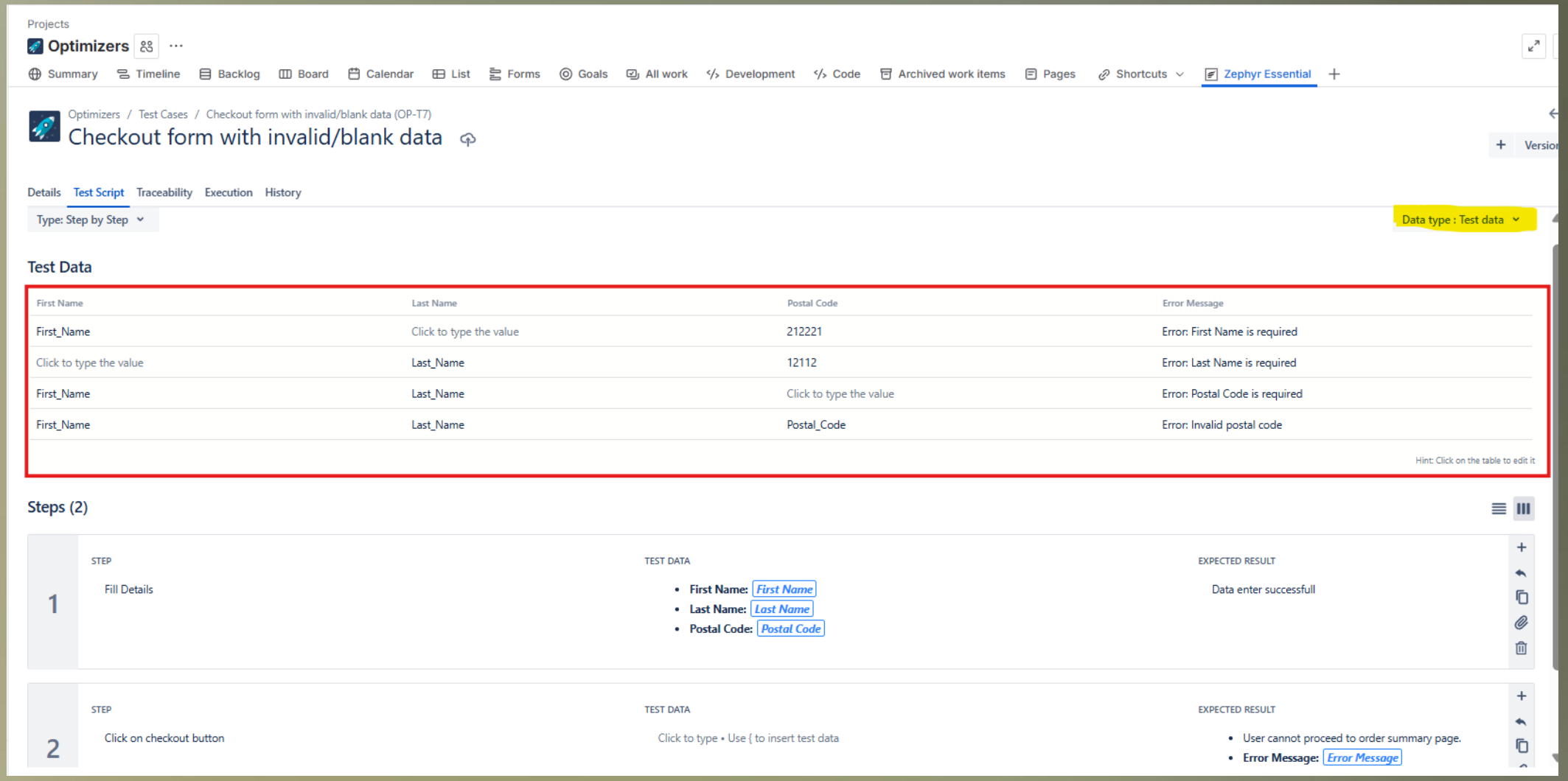


Zephyr Board

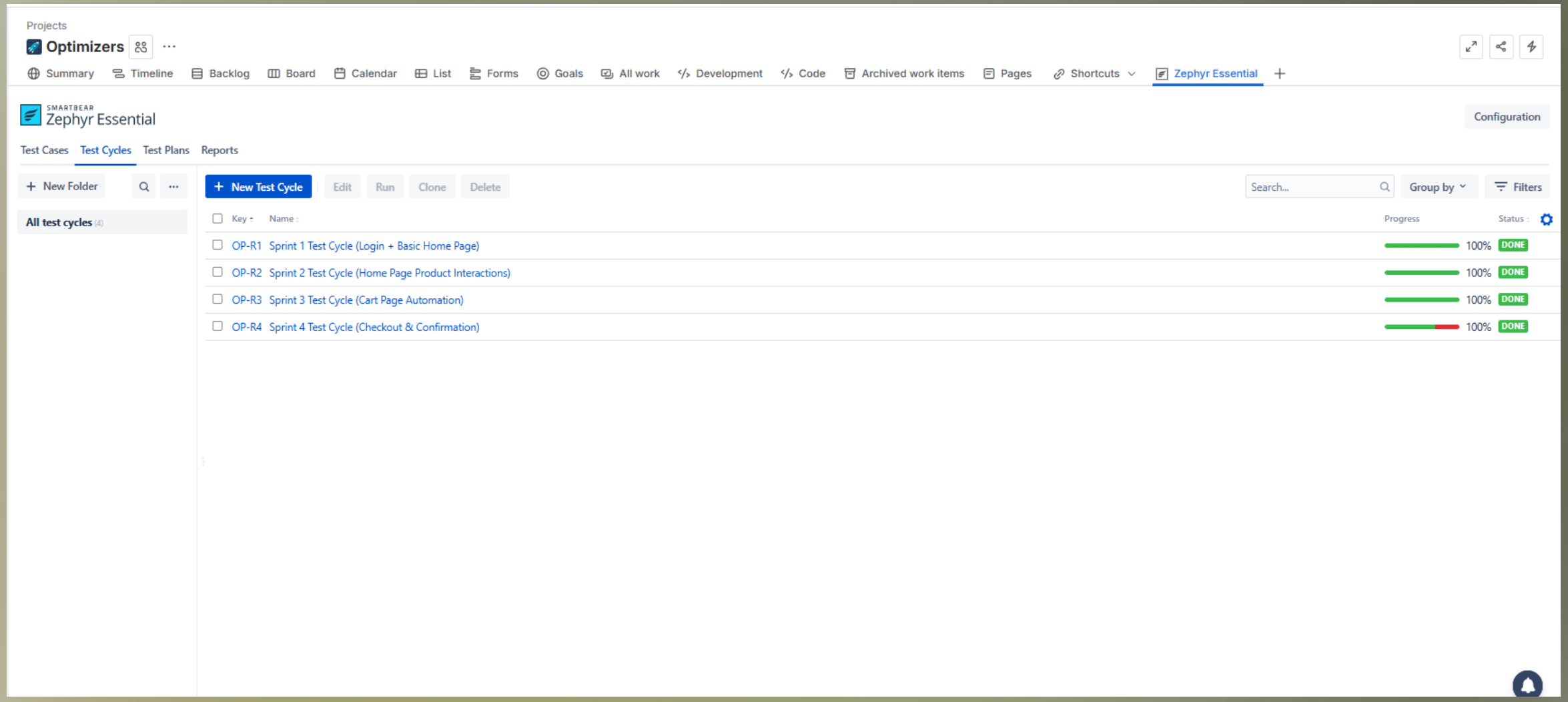
Test cases: Repository showing all created test cases.



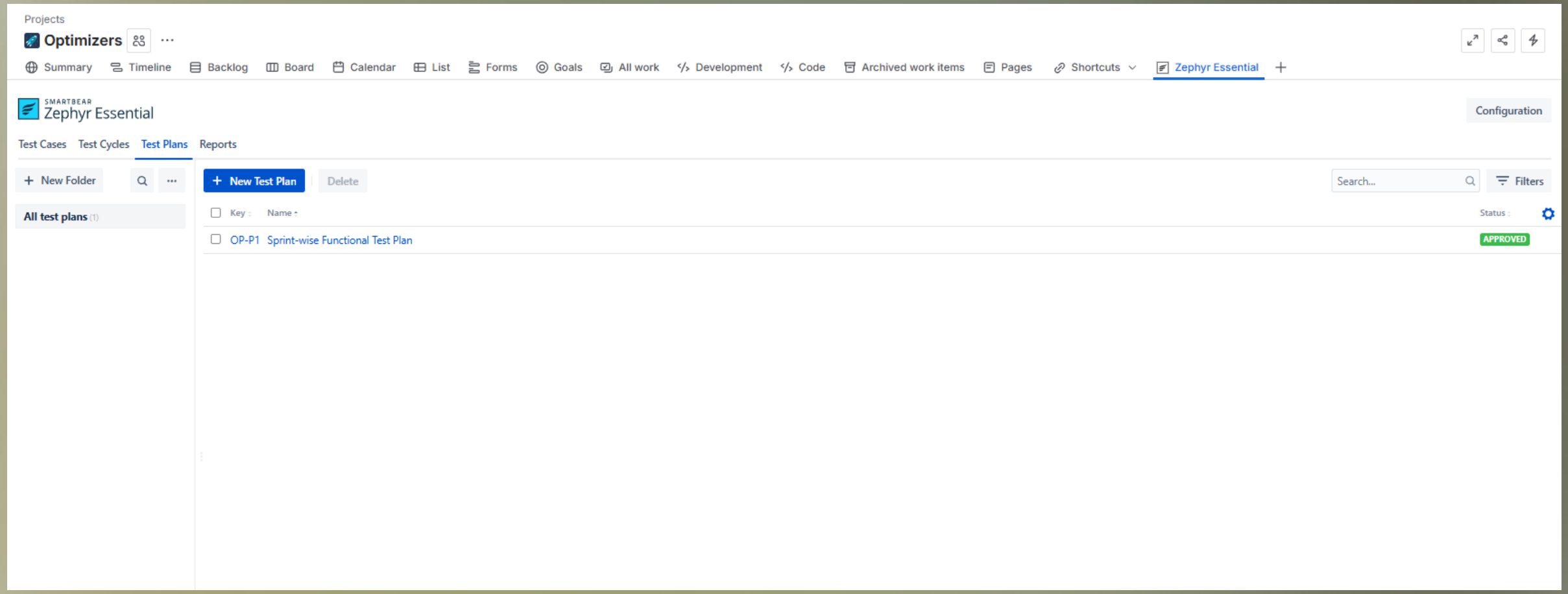
Each test case is written with clear steps. **Test Data** field is used to provide multiple inputs. Same test case can run with different sets of data **(data-driven).**



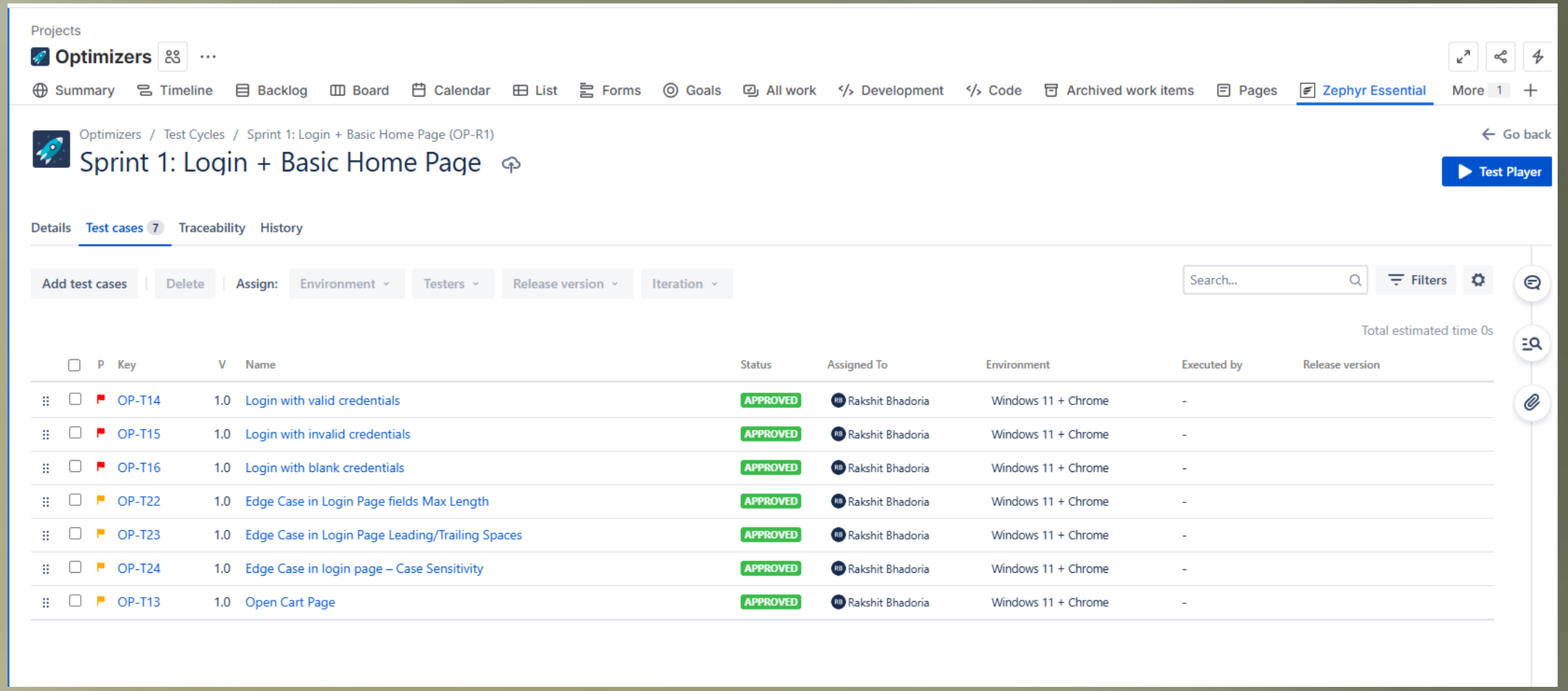
Test Cycles – Collection of test cases grouped per sprint.



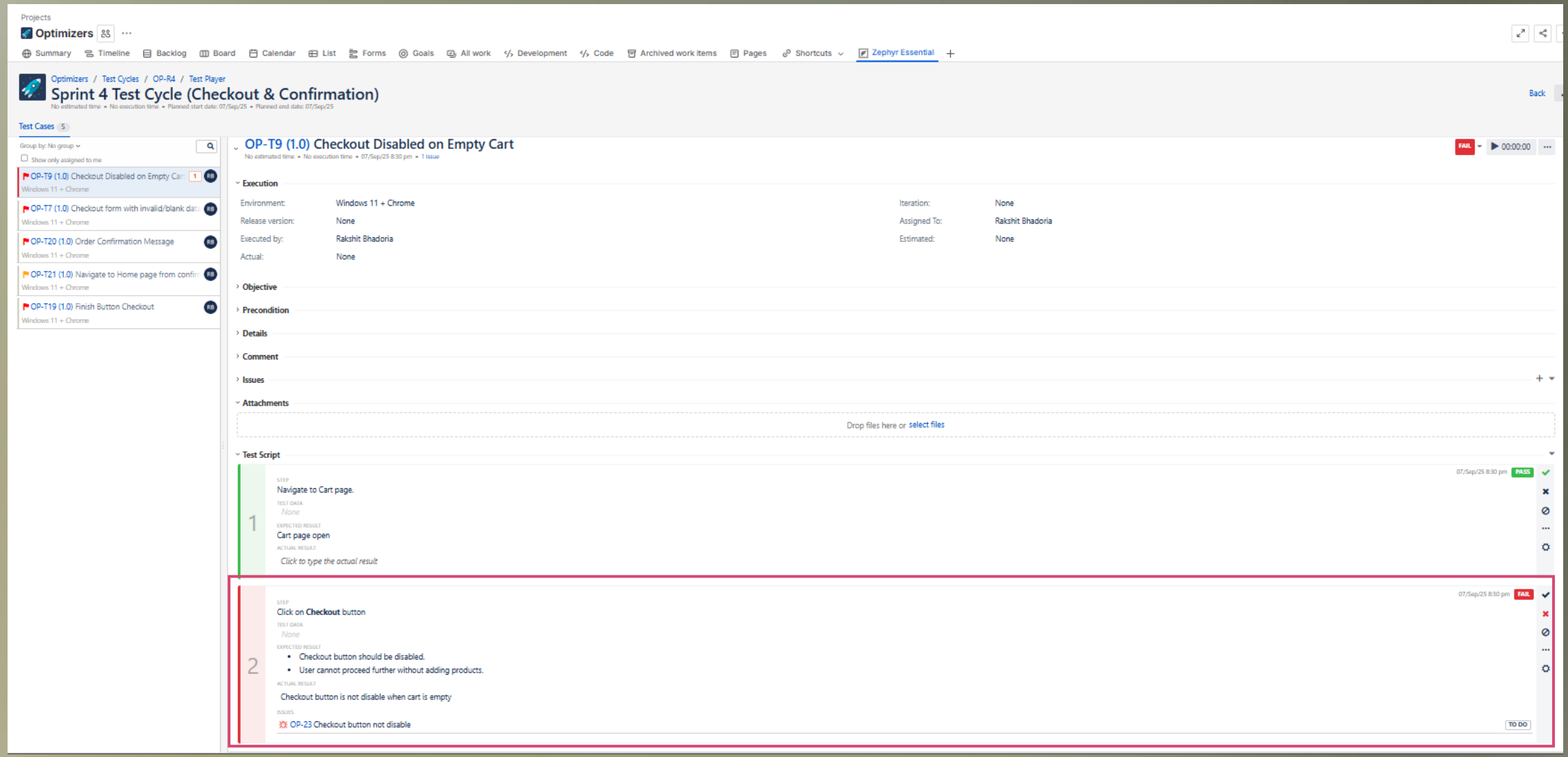
Test Plan – High-level plan combining multiple test cycles.



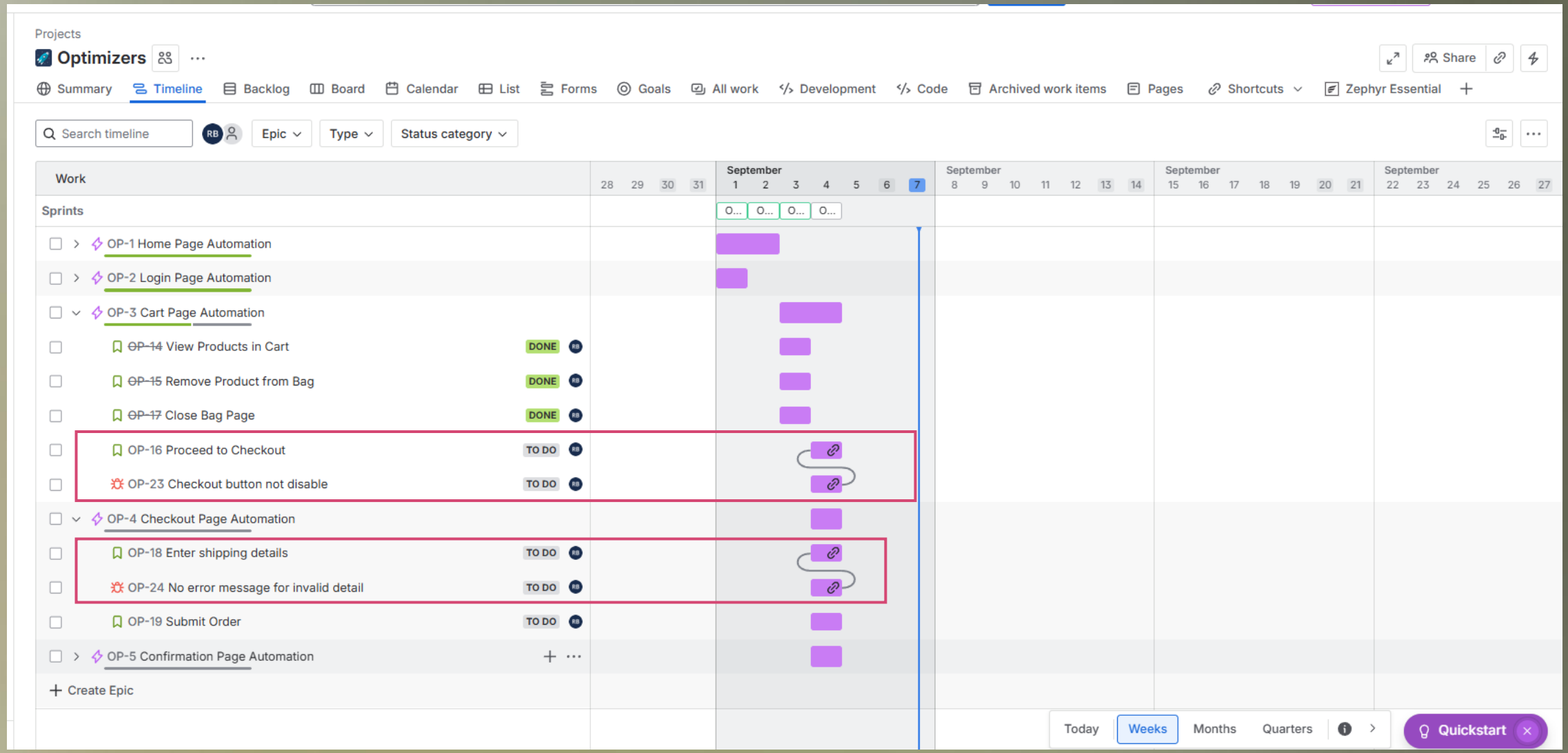
Test Cycle View – Shows one cycle with all test cases inside the test cycle



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

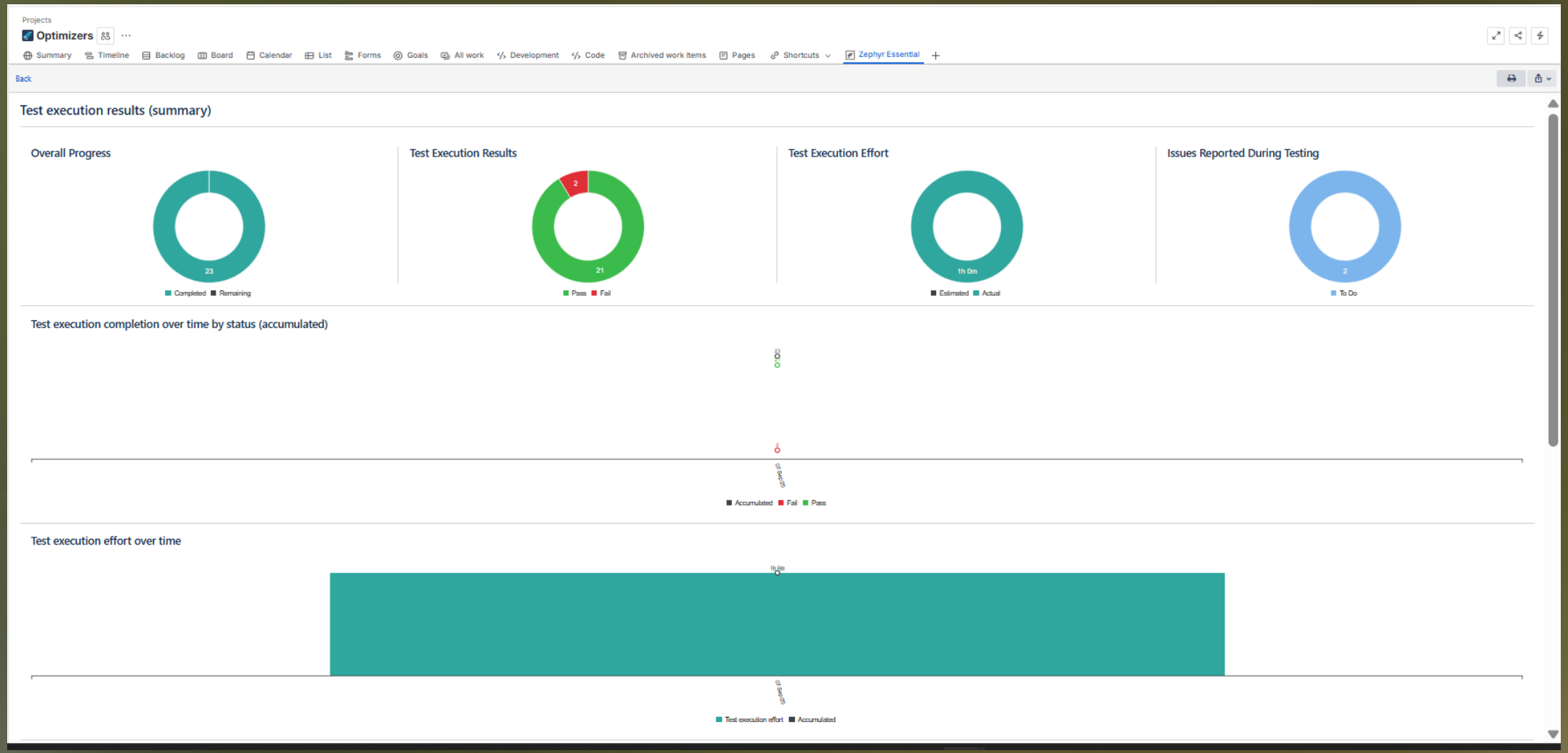


Issue Linking – Reported issues are linked to the respective user story they block.



Test Report

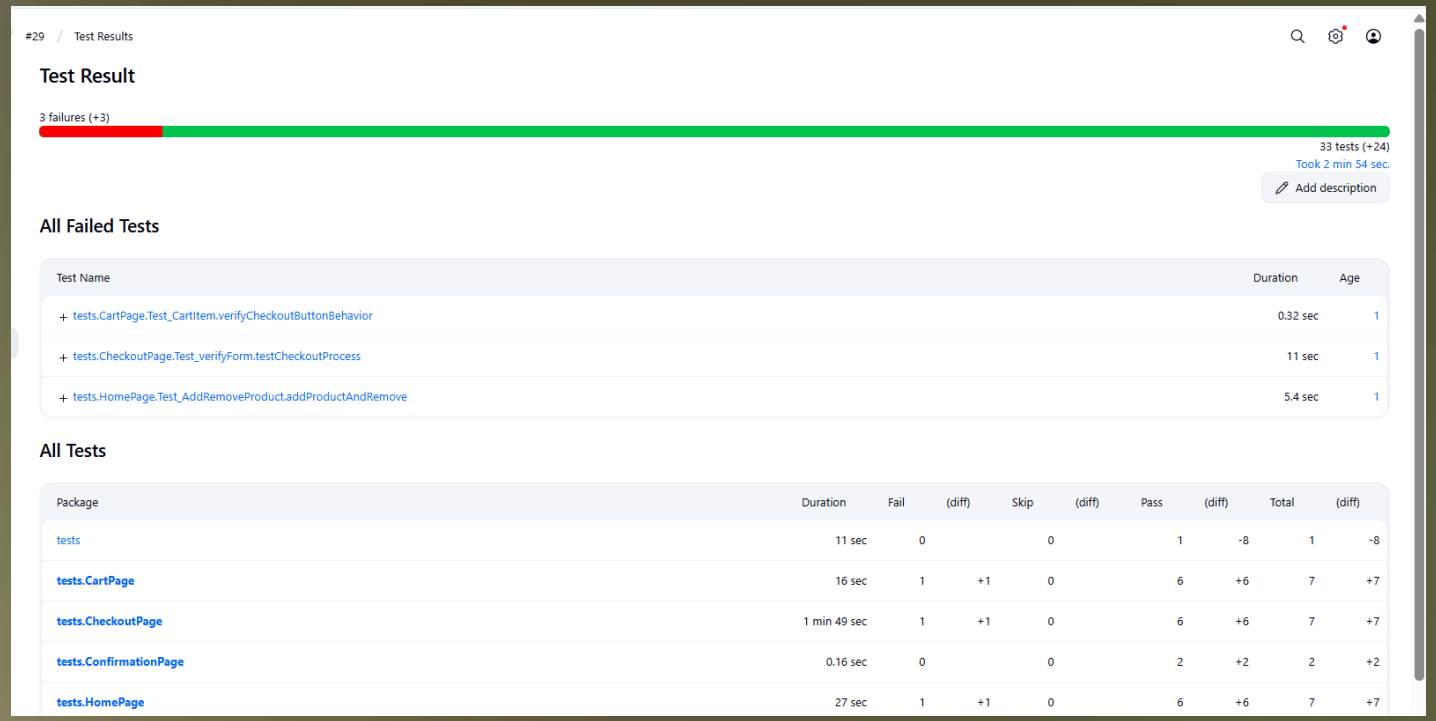
Zephyr Test Report – Provides execution status of test cases (Pass/Fail/Blocked), linked with user stories for better traceability.

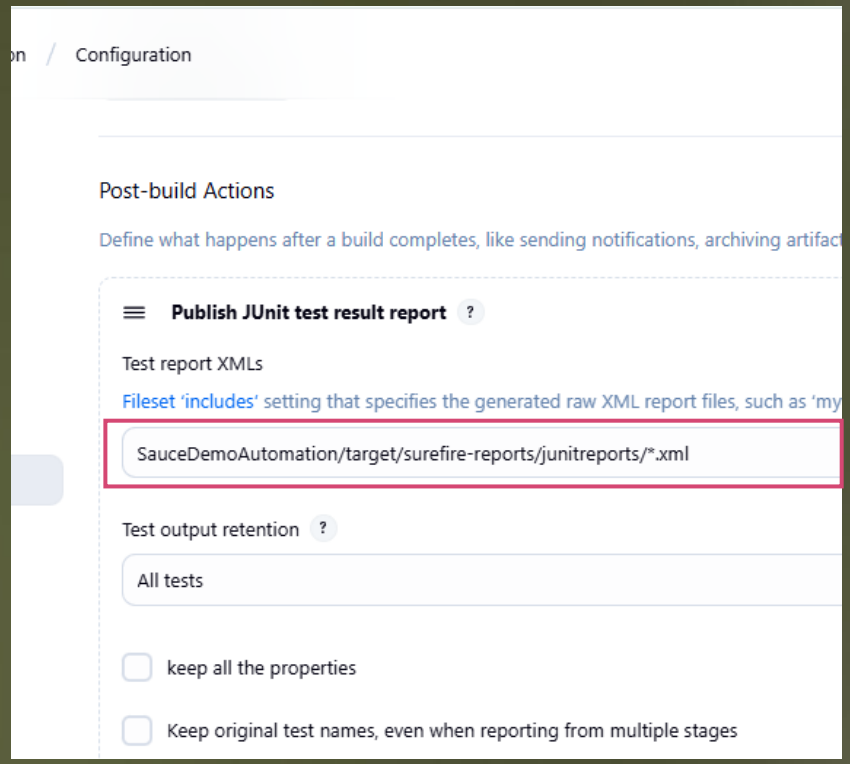


Jenkins

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

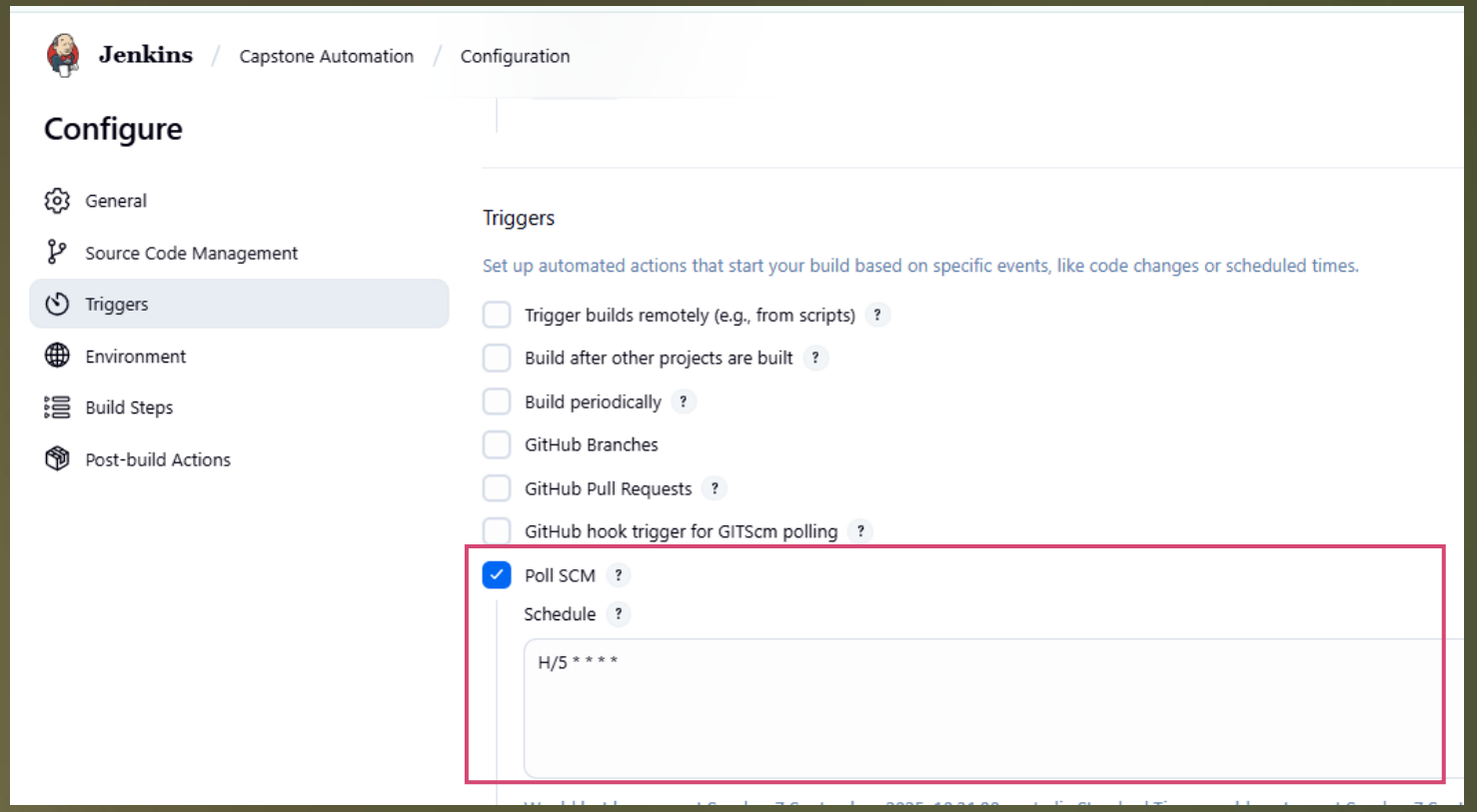
Post-Build Action (JUnit Reports) – Publishes detailed test results after every build for easy review.



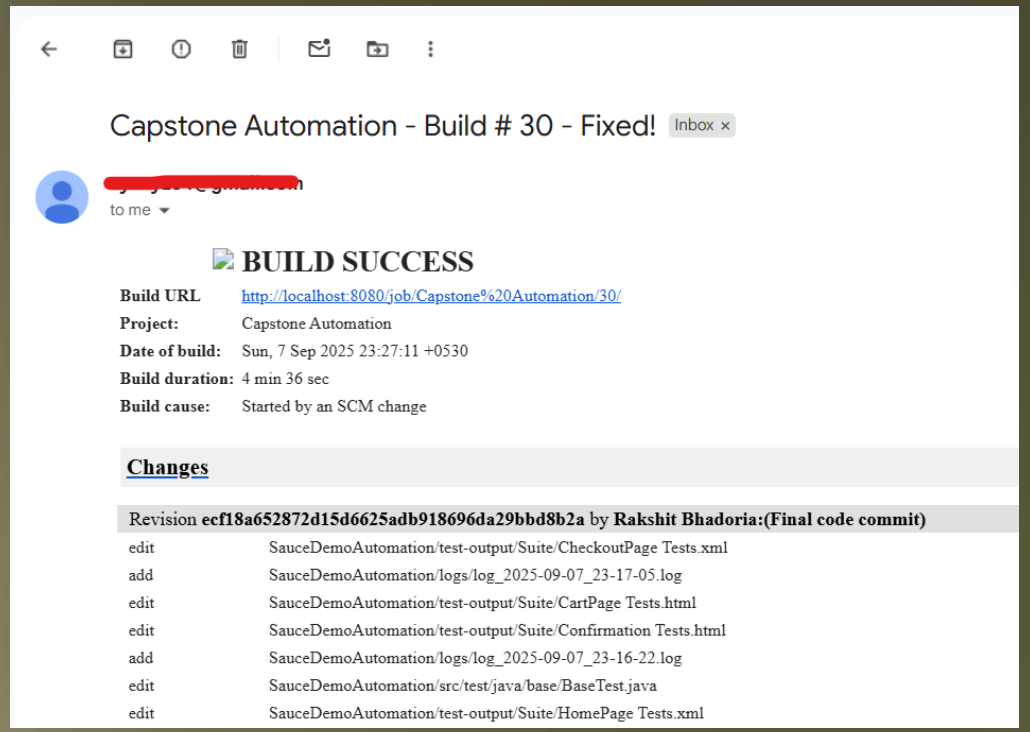


GitHub Trigger – Automatically starts a build whenever a new commit is pushed to the repository.

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.



Capstone Automation - Build # 30 - Fixed! Inbox ×

[redacted]
to me

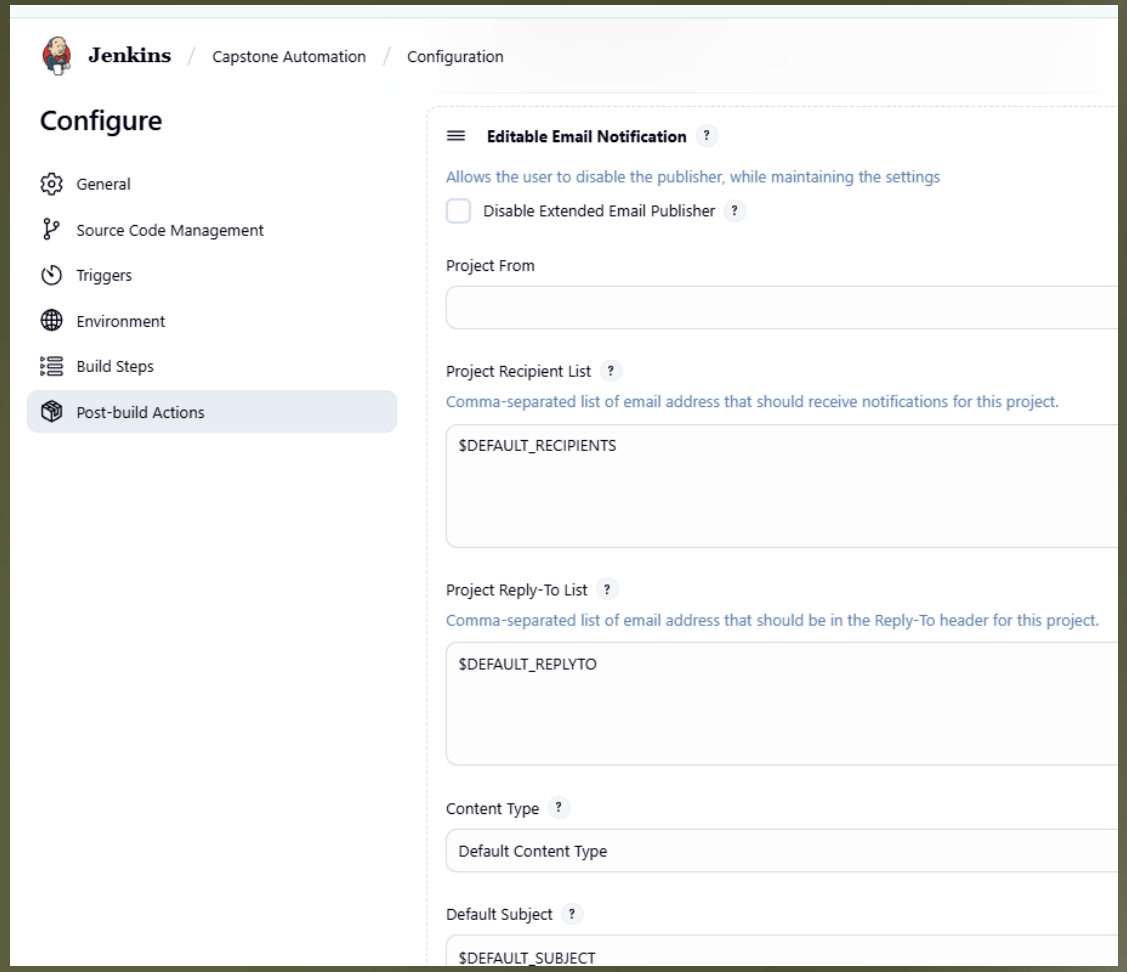
BUILD SUCCESS

Build URL	http://localhost:8080/job/Capstone%20Automation/30/
Project:	Capstone Automation
Date of build:	Sun, 7 Sep 2025 23:27:11 +0530
Build duration:	4 min 36 sec
Build cause:	Started by an SCM change

Changes

Revision **ecf18a652872d15d6625adb918696da29bbd8b2a** by **Rakshit Bhadoria:(Final code commit)**

edit	SauceDemoAutomation/test-output/Suite/CheckoutPage Tests.xml
add	SauceDemoAutomation/logs/log_2025-09-07_23-17-05.log
edit	SauceDemoAutomation/test-output/Suite/CartPage Tests.html
edit	SauceDemoAutomation/test-output/Suite/Confirmation Tests.html
add	SauceDemoAutomation/logs/log_2025-09-07_23-16-22.log
edit	SauceDemoAutomation/src/test/java/base/BaseTest.java
edit	SauceDemoAutomation/test-output/Suite/HomePage Tests.xml



Jenkins / Capstone Automation / Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

Editable Email Notification

Allows the user to disable the publisher, while maintaining the settings

Disable Extended Email Publisher

Project From

Project Recipient List

Comma-separated list of email address that should receive notifications for this project.

$DEFAULT_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

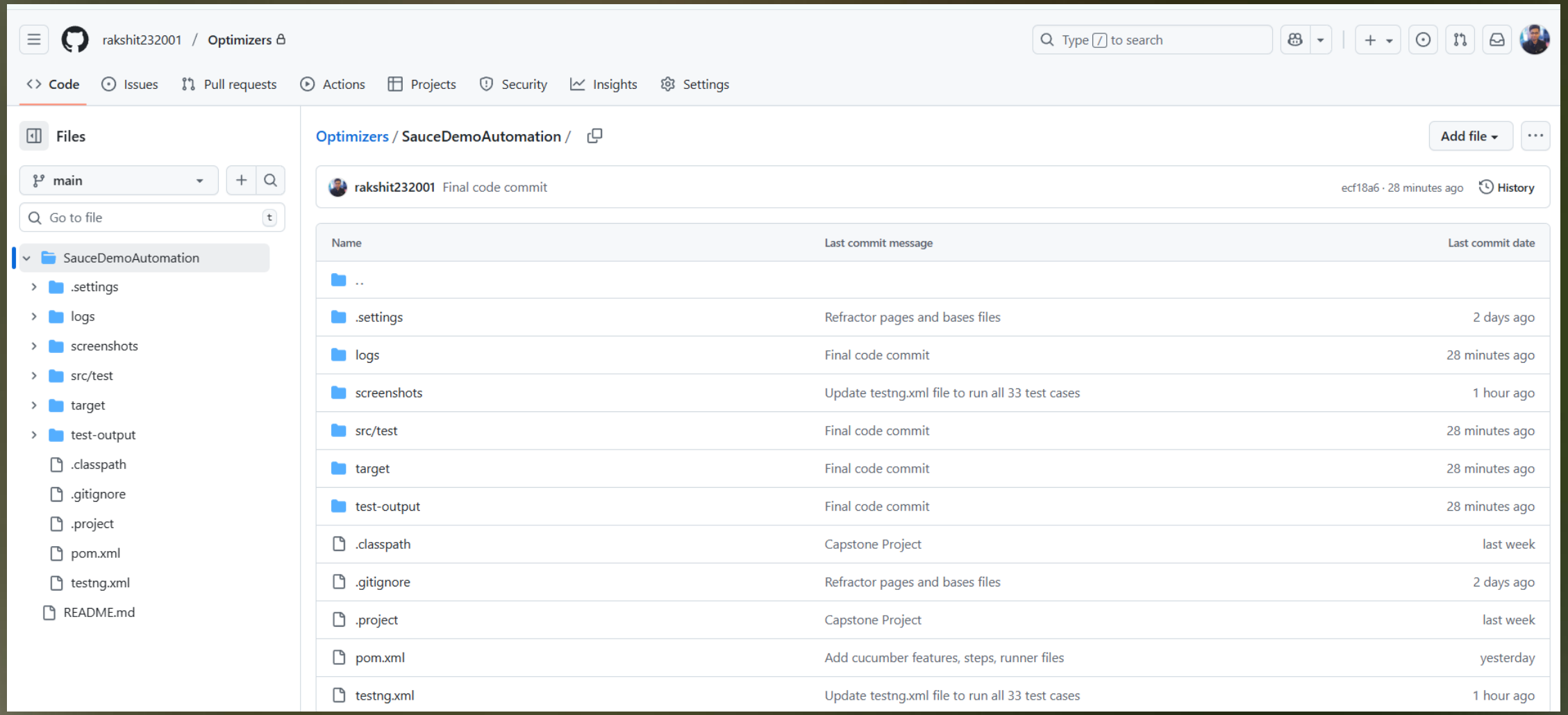
Default Content Type

Default Subject

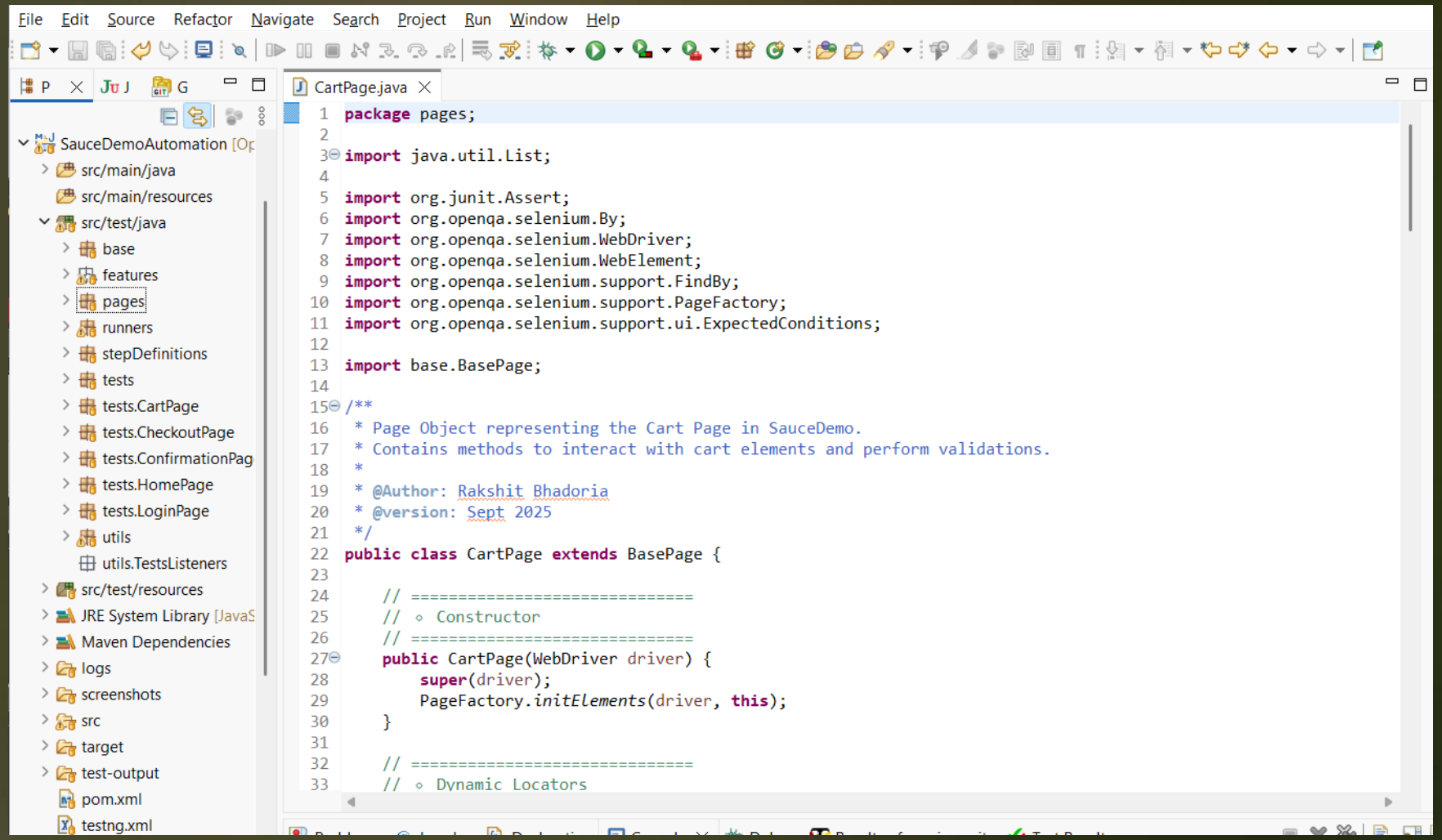
$DEFAULT_SUBJECT

Eclipse & GitHub

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.



End

Thank You!