

# Fitpeo assignment Documentation

**Name:** Tadi Naga Praveen Reddy, QA Engineer

**Note:** In step 8 of the assignment, the recurring total is to be validated for the value **820**, but as per the suggested steps, the value was updated to **560**. Hence, the script reassigns the value to **820** for accurate validation.

## Objective

The purpose of this project is to automate the testing of the **FitPeo Revenue Calculator** using Selenium WebDriver with Java and TestNG. The script navigates through the website, interacts with various web elements, and validates the output as per the assignment requirements.

---

## Prerequisites

- 1. Development Environment Setup:**
    - Install **Java Development Kit (JDK)** (version 8 or higher).
    - Install **Maven** for managing dependencies.
    - Install **IntelliJ IDEA** or any Java IDE of your choice.
    - Download and configure the latest **Selenium WebDriver**.
    - Add the required browser driver (e.g., ChromeDriver) to your system PATH.
    - Ensure the TestNG plugin is installed in your IDE.
  - 2. Dependencies:** Add the following dependencies to your pom.xml file:
    - selenium-java
    - log4j-api
    - log4j-core
    - testng
  - 3. Browser Requirements:**
    - Use the latest version of **Google Chrome** or any other supported browser.
- 

## How to Run the Script

1. Clone or download the project files from the repository.
  2. Open the project in IntelliJ IDEA or any other IDE.
  3. Verify the Maven dependencies by refreshing the pom.xml file.
  4. Execute the TestRevenueCalculator test class in src/test/java/testCases:
    - Right-click the file and select **Run 'TestRevenueCalculator'**.
-

## Test Steps Automated

1. **Navigate to FitPeo Homepage:**
    - Launch the browser and open the FitPeo homepage (<https://www.fitpeo.com/>).
  2. **Navigate to the Revenue Calculator Page:**
    - Click the "Revenue Calculator" link to navigate to the calculator page.
  3. **Scroll Down to Slider Section:**
    - Scroll until the slider is visible on the page.
  4. **Adjust the Slider:**
    - Adjust the slider value to **820** using the `moveSliderTo()` method.
  5. **Update the Text Field:**
    - Update the text field value to **560** and verify the slider's position.
  6. **Validate Slider Value:**
    - Confirm the text field value reflects the slider's position accurately.
  7. **Select CPT Codes:**
    - Check the required CPT checkboxes: **CPT-99091**, **CPT-99453**, **CPT-99454**, and **CPT-99474**.
  8. **Validate Total Recurring Reimbursement:**
    - Verify the total reimbursement value matches the expected **\$110700**.
- 

## Code Highlights

- **Page Object Model (POM):**
    - The project uses POM for better test maintenance and reusability.
    - Separate classes are defined for each webpage (HomePage and RevenueCalculatorPage).
  - **Logging:**
    - Logs are generated using the logger to track test execution steps and outcomes.
  - **Exception Handling:**
    - Used detailed exception handling to capture errors with meaningful messages for debugging.
- 

## Expected Output

- Test execution should navigate through the steps, log the progress, and validate the following:
    - Slider value and text field synchronization.
    - CPT codes selection.
    - Total reimbursement value display.
-

## Script Files Overview

1. **src/main/java/pageObjects:**
    - Contains POM classes for HomePage and RevenueCalculatorPage.
  2. **src/test/java/testCases/TestRevenueCalculator.java:**
    - Contains the test case logic and validation steps.
  3. **logs/:**
    - Stores logs for debugging and test result analysis.
  4. **pom.xml:**
    - Manages project dependencies and configurations.
-