# Task Documentation

#### ****1. AdjustSliderTo820.java****

* **Purpose**: This script automates setting a slider's value to 820 on the FitPeo Revenue Calculator page.
* **Steps**:
  1. Opens the FitPeo Revenue Calculator page.
  2. Waits for the slider input box to load using an explicit wait.
  3. Uses JavaScript to directly set the slider value to 820.
  4. Triggers input and change events to simulate user input.
  5. Verifies if the value has been updated and logs the result.
* **Use Case**: Automates scenarios where manual slider adjustment is required to test system behavior

#### ****2. NavigateToFitPeo.java****

* **Purpose**: Navigates to the FitPeo homepage and validates successful navigation.
* **Steps**:
  1. Opens the FitPeo homepage (https://www.fitpeo.com).
  2. Maximizes the browser window.
  3. Prints success or failure messages in the console.
  4. Closes the browser after execution.
* **Use Case**: Verifies the accessibility of the FitPeo homepage

#### ****3. NavigateToRevenueCalculator.java****

* **Purpose**: Navigates to the FitPeo Revenue Calculator page.
* **Steps**:
  1. Opens the Revenue Calculator page.
  2. Maximizes the browser window.
  3. Logs navigation status in the console.
  4. Closes the browser post-execution.
* **Use Case**: Confirms if the Revenue Calculator page loads correctly​

#### ****4. ScrollToSliderSection.java****

* **Purpose**: Scrolls to the slider section and adjusts its value.
* **Steps**:
  1. Navigates to the Revenue Calculator page.
  2. Locates the slider using its XPath.
  3. Scrolls to the slider using JavaScript.
  4. Adjusts the slider value to 1000 and verifies the update.
* **Use Case**: Ensures smooth interaction with slider elements even when they are off-screen​

#### ****5. SelectCPTCodes.java****

* **Purpose**: Selects CPT (Current Procedural Terminology) codes on the Revenue Calculator page.
* **Steps**:
  1. Loads the Revenue Calculator page.
  2. Scrolls to and selects specified checkboxes using JavaScript.
  3. Handles scenarios where checkboxes are already selected or take time to load.
* **Use Case**: Tests scenarios involving multiple checkbox selections​

#### ****6. UpdateTextFieldAndSlider.java****

* **Purpose**: Updates a text field and a slider simultaneously to ensure consistency.
* **Steps**:
  1. Navigates to the Revenue Calculator page.
  2. Updates a text field value to 0 and triggers input events.
  3. Updates the slider value to 560 and verifies the change.
  4. Validates synchronization between the text field and slider values.
* **Use Case**: Tests input synchronization between two dependent UI elements​

#### ****7. ValidateRecurringReimbursement.java****

* **Purpose**: Extracts and validates recurring reimbursement values for different CPT codes.
* **Steps**:
  1. Navigates to the Revenue Calculator page.
  2. Maps CPT codes to their corresponding reimbursement values.
  3. Extracts and logs reimbursement values for each CPT code.
* **Use Case**: Validates the correctness of reimbursement calculations on the page​

#### ****8. ValidateSliderValue.java****

* **Purpose**: Verifies that the slider reflects the value updated in the text field.
* **Steps**:
  1. Updates a text field with a value (e.g., 560).
  2. Verifies if the slider reflects the same value.
* **Use Case**: Ensures consistency between slider values and associated input fields​

#### ****9. VerifyTotalRecurringReimbursement.java****

* **Purpose**: Updates and validates header values related to recurring reimbursement.
* **Steps**:
  1. Navigates to the Revenue Calculator page.
  2. Dynamically assigns updated reimbursement values to header elements.
  3. Verifies the updates visually and programmatically.
* **Use Case**: Tests the dynamic update of key metrics displayed on the page​