**Unit Testing**

**Borrower Personal Details Page:**

* Verify that the input fields for personal details (name, date of birth, contact information) are rendered correctly.
* Test that the input fields validate user input (e.g., required fields, valid email format).
* Check that the data entered in the fields is correctly captured and stored in the state or passed to the API.

**Income Page:**

* Test that the input fields for income details (amount, frequency) are rendered correctly.
* Validate that the income amount field accepts only numeric values and validates the minimum and maximum values.
* Ensure that the selected frequency is correctly captured and stored in the state or passed to the API.

**Employment Page:**

* Verify that the input fields for employment details (job title, company name) are rendered correctly.
* Validate that the fields accept and store the entered data correctly.
* Test that the checkbox for current employment status works as expected.

**Assets Page:**

* Check that the input fields for asset details (type, description, value) are rendered correctly.
* Validate that the value field accepts only numeric values and validates the minimum and maximum values.
* Ensure that the entered asset data is correctly captured and stored in the state or passed to the API.

**Liabilities Page:**

* Verify that the input fields for liability details (type, description, amount) are rendered correctly.
* Validate that the amount field accepts only numeric values and validates the minimum and maximum values.
* Ensure that the entered liability data is correctly captured and stored in the state or passed to the API.

**Credit Information Page:**

* Check that the input fields for credit information (credit score, credit history) are rendered correctly.
* Validate that the credit score field accepts only numeric values and validates the minimum and maximum values.
* Ensure that the entered credit information is correctly captured and stored in the state or passed to the API.

**Product Selection Page:**

* Verify that the radio buttons or select dropdowns for selecting loan products and terms are rendered correctly.
* Test that the selected loan product and term are correctly captured and stored in the state or passed to the API.

**Property Information Page:**

* Check that the input fields for property information (address, property type, value) are rendered correctly.
* Validate that the value field accepts only numeric values and validates the minimum and maximum values.
* Ensure that the entered property information is correctly captured and stored in the state or passed to the API.

**Rate Locking Page:**

* Verify that the input fields for rate locking details (interest rate, expiration date) are rendered correctly.
* Validate that the interest rate field accepts only numeric values and validates the minimum and maximum values.
* Ensure that the selected expiration date is correctly captured and stored in the state or passed to the API.

These are just a few examples of the unit test cases that you can consider for each page or component. The specific test cases will depend on the functionalities and validations implemented in your application. It's important to cover different scenarios, edge cases, and error conditions to ensure the robustness and correctness of your code.