

Test Plan and Results

Overview

This document outlines the testing strategy, approach, test cases, and results for our project. The primary goal is to ensure that all functionalities of the components work as expected, covering aspects such as rendering, user interactions, error handling, and API integrations. This comprehensive testing strategy ensures the robustness and reliability of the application.

Testing Strategy

General Approach

1. Component Rendering:

- Verify that all components render correctly and display the expected initial state.
- Ensure that components correctly display data fetched from APIs.

2. User Interactions:

- Simulate user interactions such as clicks, form submissions, and navigation to verify the application's behavior.
- Test toggling of buttons and switching views to ensure state updates correctly.

3. Error Handling:

- Mock API failures and user errors to ensure that the application handles them gracefully and displays appropriate messages.

4. State Management:

- Verify that the component state updates correctly based on user interactions and API responses.

5. API Integration:

- Mock API calls to test the application's behavior without relying on real backend services.
- Ensure that data fetching, submission, and updates work as expected.

Tools and Libraries

- **React Testing Library:** For rendering components and simulating user interactions.
- **Jest:** For assertions and mocking functions.
- **Jest-DOM:** For additional DOM assertions.

Test Cases

Component Rendering

1. Component Rendering Test

- **Description:** Ensure that each component renders and displays the initial state correctly.
- **Expected Outcome:** The component should render without errors and display the expected content.

User Interactions

2. Form Display Test

- **Description:** Verify that clicking on elements displays the correct forms or views.
- **Expected Outcome:** The form or view should be displayed correctly with all necessary fields.

3. Form Submission with Valid Data

- **Description:** Submit the form with all required fields filled and verify successful submission.
- **Expected Outcome:** The form should submit successfully, and a success message should be displayed.

4. Form Submission with Invalid Data

- **Description:** Submit the form with missing or invalid data and verify validation.
- **Expected Outcome:** An error message should be displayed indicating that required fields must be filled or corrected.

5. Button Clicks and Toggle Views

- **Description:** Simulate button clicks and view toggles to verify state updates.
- **Expected Outcome:** The state should update correctly, and the view should change accordingly.

Error Handling

6. API Error Handling

- **Description:** Mock API failures and verify that the application handles errors gracefully.
- **Expected Outcome:** An appropriate error message should be displayed, and the application should remain functional.

API Integration

7. Data Fetching Test

- **Description:** Verify that data is fetched from the API and displayed correctly.
- **Expected Outcome:** The data should be displayed correctly in the component after fetching.

8. Data Submission Test

- **Description:** Verify that data is submitted to the API correctly.
- **Expected Outcome:** The data should be submitted without errors, and the response should be handled correctly.

Testing results:

PASS	src/components/Home.test.js	(14.68 s)					
File	% Stmt	% Branch	% Funcs	% Lines	Uncovered Line #s		
All files	84.68	74.19	78.88	84.58			
src	100	100	100	100			
firebase.js	100	100	100	100			
src/components	100	100	100	100			
About.js	100	100	100	100			
src/components/Admin-dashboard	81.6	73.52	78.18	80.67			
Admin-dashboard.js	70	56.25	37.5	70	26,35,53-74		
ManageFundManagers.js	83.92	73.07	84	82.69	34,46,65,75,85,91,100,111,176		
ManageUsers.js	83.67	84.61	86.36	82.97	27,39,60,70,81,87,96,101		
src/components/Managers-dashboard	86.58	80	75.86	87.5			
PostAds.js	95	50	92.3	95	39,53		
Review-applications.js	78.57	83.33	62.5	80	29,57,69,77-79,99-100		
src/components/Users-dashboard	90.9	60	100	90.9			
Users-home.js	90.9	60	100	90.9	26,42		


```
Test Suites: 10 passed, 10 total
Tests:      36 passed, 36 total
Snapshots:  0 total
Time:      20.615 s, estimated 32 s
Ran all test suites.
PS C:\Users\hans1\OneDrive\Desktop\funding-management\funding-management> █
```