

# Testing Policy Document

**Project:** Smart Student Handbook

---

## 1. Purpose

This document defines the testing policies and guidelines for the Smart Student Handbook project. It ensures all code is tested systematically to maintain quality, reliability, and performance.

---

## 2. Scope

This policy applies to:

- All components in the `components/` directory
- All pages and APIs in the `app/` directory
- Libraries and utilities in the `lib/` directory

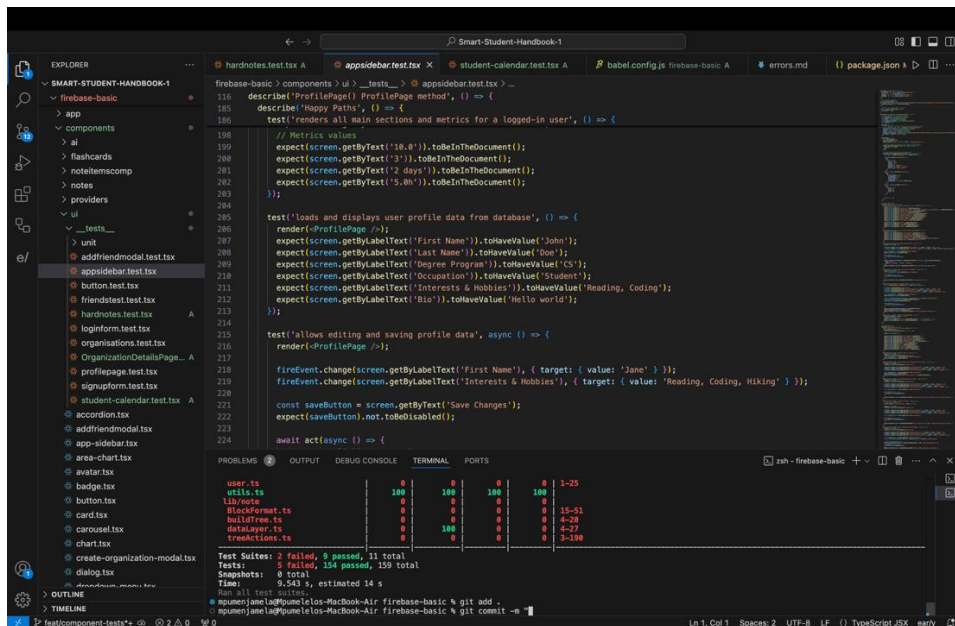
It covers **unit testing**, **integration testing**, **mocking of external dependencies**, and **code coverage analysis**.

---

## 3. Testing Strategy

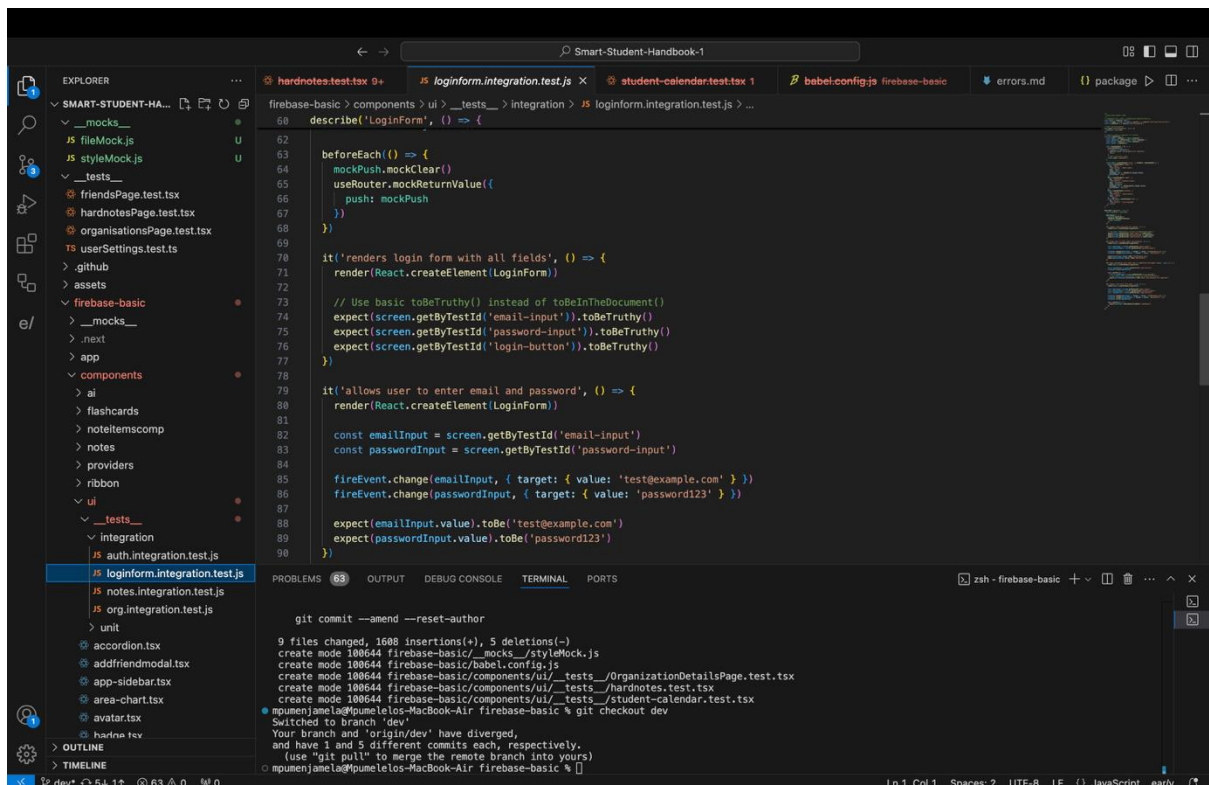
### 3.1 Unit Testing

- Frameworks: **Jest + React Testing Library**
- Focus on:
  - Component rendering
  - Event handling and user interactions
  - API mocks and asynchronous operations



## 3.2 Integration Testing

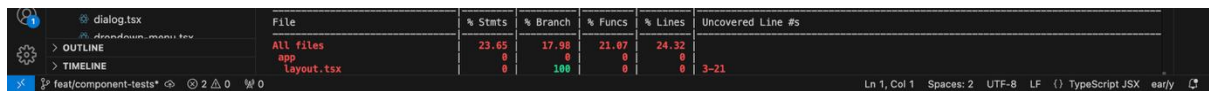
- Test interactions between components and pages
- Mock external dependencies like Next.js router or Firebase



## 3.3 Coverage Policy

### -Test the most important components

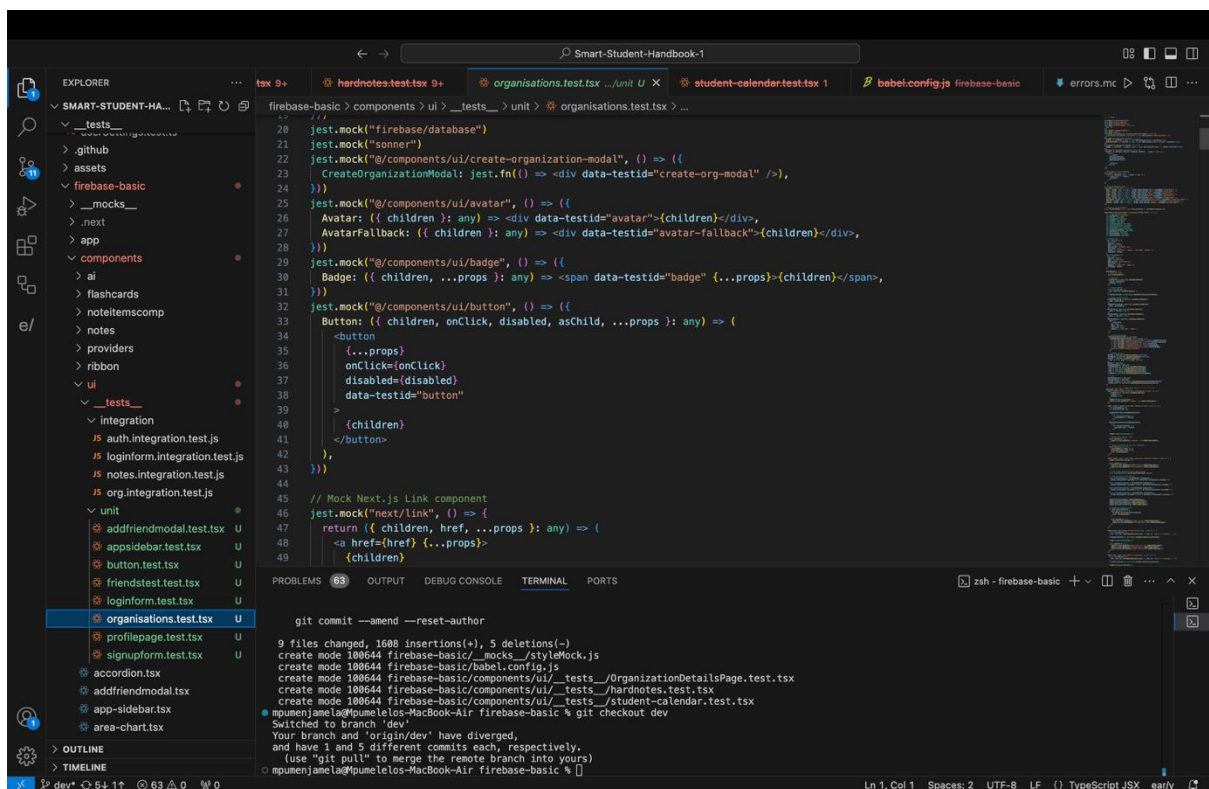
- Priority for testing:
  - Pages (app/) with complex logic
  - Core UI components (components/ui/)
  - Utility libraries (lib/)



File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	23.65	17.98	21.07	24.32	
app	0	0	0	0	
layout.tsx	0	100	0	0	3-21

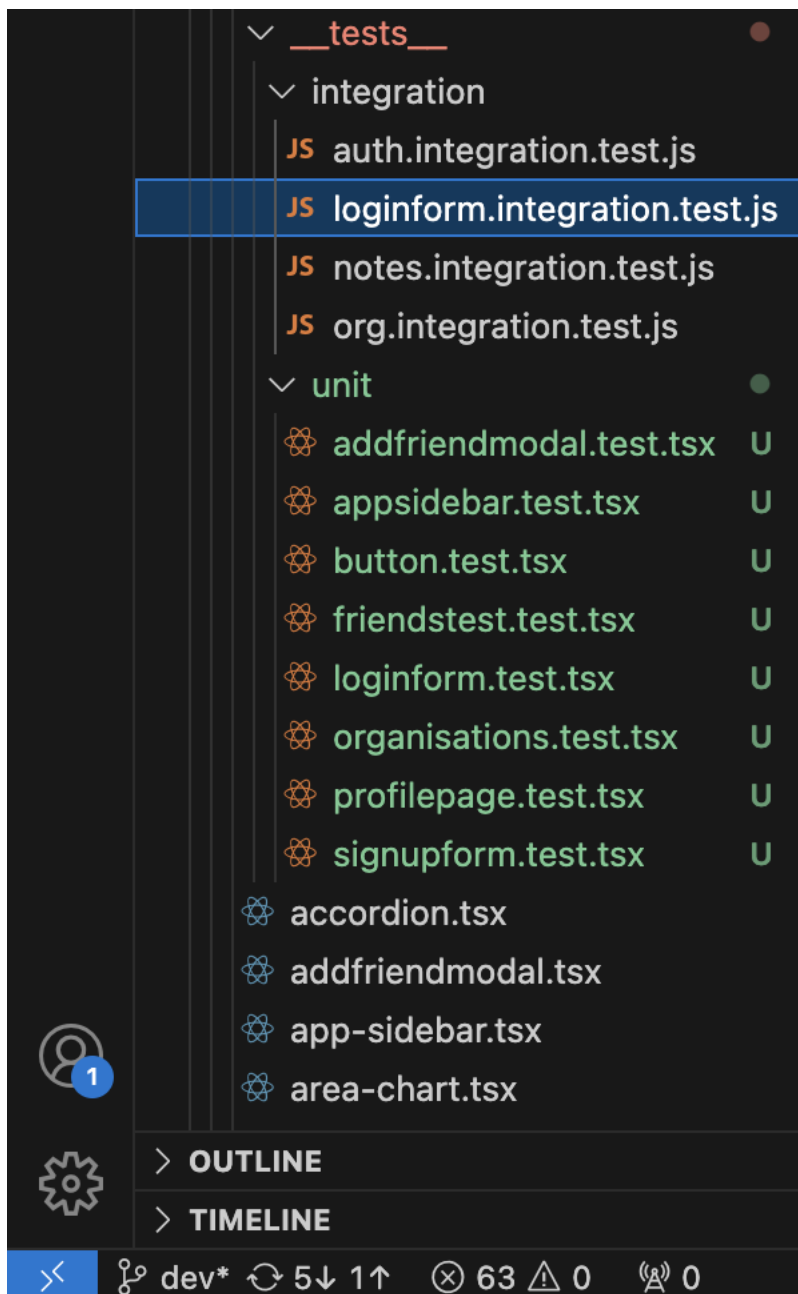
## 4.4 Mocking Guidelines

- **Next.js router:** `jest.mock('next/router')`
- **Firebase:** Mock Firestore, Auth, and Realtime Database calls
- **Global browser APIs:** `ResizeObserver`, `matchMedia`, `Response`, `TextEncoder`, `TextDecoder`



## 5. Test Naming and Location

- **Unit tests:** `components/__tests__/unit/`
- **Integration tests:** `__tests__/integration/`
- **File naming:** `<componentOrPage>.test.tsx`



## 6. Test Execution

- Run all tests:

```
npm test
```

- Run tests with coverage:

```
npm test -- --coverage
```

- Run a single test file:

```
npm test -- path/to/file.test.tsx
```

---

## 7. Reporting and Review

- All test results and coverage reports should be reviewed during **pull requests**
- Failures must be fixed before merging to main branches

---

## 8. Exceptions

- Temporary mocks or stubs for unimplemented APIs are allowed
- Features can be excluded with justification

---

## 9. References

- Jest Documentation
- React Testing Library
- Internal coding standards and branching policies