# Test Plan: Favorite (Bookmark) Feature

**Test Plan Report by: Arnav Kanade**

## 1) Test Plan Identifier

TP-FAV-001 - (Test Plan-Favorite System-001)

## 2) Introduction

This test plan outlines how we'll test the **Favorite (Bookmark) Feature** in the VerifyPair app. This feature allows users to save their favorite questions for quick access later. Instead of using localStorage, the data will be stored and managed using a **JSON web server**, ensuring it's persistent and accessible across sessions.

We'll test everything from how the feature looks and behaves on the screen, to making sure it communicates properly with the server, updates the state correctly, and is easy for everyone to use. Our goal is to ensure this feature works seamlessly and reliably for all users.

## 3) Test Items

- Bookmarking a Question: Users can click a button to add or remove a question from their favorites.

- Persistent Storage: Favorited questions remain saved even after refreshing the page.

- UI State Updates: The button should change dynamically when a question is favorited or unfavorited.

- Favorites Page: Users can view all bookmarked questions on a dedicated page.

## 4) Features to be Tested

- Favorite Toggle Button - Clicking a button should add or remove the question from favorites.

- Persistence - Favorited questions should remain saved in localStorage.

- UI Feedback - Button should change dynamically to indicate a question is favorited.
- Favorites Page - Only bookmarked questions should appear in the favorites route.

- Routing and Navigation - Users can navigate to the favorites page and return.

**5) Feature Not to be Tested**

- Creating or deleting questions (covered in a separate test).

-Creating or deleting answers

- User authentication (covered in login or logout tests).

-User Reputation

**6) Approach**

The testing will be conducted using:

- Unit Testing (Jest and React Testing Library) - Verify that the toggleFavorite function updates localStorage correctly.

- Integration Testing (Cypress) - Test that clicking the button correctly adds or removes the question from the favorites list.

- Manual Testing - Ensure UI updates reflect correct state changes.

**7) Item Pass or Fail criteria**

Pass Criteria:

- Clicking the favorite button adds or removes a question from favorites.

- Refreshing the page persists the favorited questions.

- The favorites page correctly displays all saved favorite questions.

- The UI reflects the favorite state dynamically.

Fail Criteria:

- Clicking the favorite button does not update the favorites list.

- Refreshing the page loses the favorited questions.

- A question that is not favorited appears in the favorites page.

- The favorite button does not change after clicking.

8) Suspension criteria or Resumption requirements

- Testing will be paused if localStorage does not work correctly.

- Testing will resume once browser storage is enabled or a mock API is used for persistence.

**9) Test Deliverables**

- Test cases document (Jest and Cypress test scripts).

- Test execution report (pass or fail status for each test case).

- Bug report (if any issues are found).

**10) Testing Tasks**

- Write unit tests to verify toggleFavorite updates localStorage correctly.

- Implement Cypress tests for UI interactions (adding or removing favorites).

- Perform manual testing to ensure UI reflects state changes.

- Log and report bugs for unexpected behavior.

**11) Environmental Needs**

- Browser: Google Chrome (latest version).

- Tools: Jest, React Testing Library, Cypress.

- Test Data: Pre-existing questions for favoriting or unfavoriting.

**12) Responsibilities**

- Frontend Developer: Ensures the UI correctly reflects favorite state changes.

- QA Tester: Runs Jest and Cypress tests and reports bugs.

- Team Lead: Reviews test results and approves fixes.

**13) Staffing and Training Needs**

- Familiarity with Jest and Cypress for writing tests.

- A training session on handling localStorage in React if needed.

**14) Schedule**

Task - Duration - Assigned To

Write unit tests - 1 day - Developer

Implement Cypress tests - 2 days - QA Tester

Manual Testing - 1 day - Tester

Bug Fixes and Review - 1 day - Team Lead

Final Approval - 1 day - Team Lead

## 15) Risk and Contingencies

Risk - Impact - Mitigation

- JSON web server downtime or failure - High - Implement a fallback mechanism, such as caching data temporarily on the client side using IndexedDB or a mock API.
- UI does not update after clicking - Medium - Ensure useState updates correctly
- Users cannot remove a favorite - Medium - Debug toggleFavorite logic
- Page crashes due to missing data - High - Add default empty array handling

## 16) Approvals

Project Lead: Arnav Kanade

QA Tester: Arnav Kanade

Professor: Prof. Peter Dillinger