Test Plan for Answer Management

Test Plan:
VerifyPair
Prepared by:
Manu Janardhana
14-03-2025

Table of Contents

1.0 Introduction

Module/Function Under Test: Answer Management and Voting

This module is responsible for displaying answers to a question, allowing users to post new answers, editing and deleting existing answers, and enabling users to upvote or downvote answers. It includes:

- Creating and submitting answers

- Editing and deleting answers

- Displaying a list of answers

- Upvoting/downvoting an answer

- Tracking the total score (net upvotes/downvotes) for each answer

Additionally, the answer list is sorted based on the number of upvotes, ensuring that the most popular answers are displayed first.

The Q&A platform allows users to view questions, post answers, edit or delete their answers, vote on answers, and see updated vote counts and sorted order.

2.0 Objectives and Tasks

2.1 Objectives

• Verify that the Answer Management and Voting module meets functional requirements: posting, editing, deleting answers, voting, and updating vote counts with sorted order.

• Ensure correct handling of user actions and error/edge cases (e.g., editing/deleting restrictions, voting without login, etc.).

2.2 Tasks

• Test Planning and preparation of this document.

• Design detailed test cases covering all functional, integration, and edge scenarios.

• Setup test environment (local/staging) matching production.

• Execute tests and log defects.

• Retest and confirm defect resolution, then produce a test summary report.

3.0 Scope

In-Scope:

• All functionalities related to Answer Management and Voting including posting, editing, deleting, displaying, and voting on answers.

• Ensuring the answer list is sorted based on upvote count.

Out-of-Scope:

• User authentication, question creation/editing, and UI theme styling.

4.0 Testing Strategy

4.1 Unit Testing

• Test individual functions (e.g., createAnswer, editAnswer, deleteAnswer, upvoteAnswer, downvoteAnswer) using frameworks like Jest.

• Use mocks to isolate external dependencies.

4.2 Integration Testing

• Verify interactions between front-end and back-end (API endpoints) in a test/staging environment.

• Check end-to-end flows from posting, editing, or deleting an answer to database update and UI reflection.

4.3 System (E2E) and Acceptance Testing

• Use tools like Cypress or Selenium to simulate real user scenarios.

• Validate UI updates, vote count accuracy, edit and delete permissions, and sorted order based on upvotes.

4.4 Performance and Stress Testing

• Simulate concurrent usage (using JMeter/Locust) to ensure system stability under load.

4.5 User Acceptance Testing (UAT)

• End-users test the feature in a real-world scenario to confirm usability and functionality.

5.0 Hardware Requirements

• Minimum: Developer/Tester machine with Node.js, modern browser (Chrome/Firefox/Edge).

• Recommended: At least 8GB RAM, stable internet connection.

6.0 Environment Requirements

• Backend: JSON Server or Node.js running on localhost or dedicated test URL.

• Frontend: React app running locally or on test domain.

• Database: JSON file or test database instance.

• Tools: Jest for unit tests, Cypress/Selenium for E2E tests, and an issue tracker like Jira/GitHub Issues.

7.0 Test Schedule

Milestone	Planned Date	Owner
Test Plan Review	March 20, 2025	QA Lead
Unit Test Development	March 21–23, 2025	Developers
Integration Test Setup	March 24, 2025	QA Engineer

System/E2E Test Execution	March 25–28, 2025	QA Engineer
User Acceptance Testing	March 29–30, 2025	Stakeholders
Test Closure & Sign-Off	March 31, 2025	QA Lead

8.0 Control Procedures

8.1 Problem Reporting

• Use an issue tracker (e.g., Jira or GitHub Issues) to log defects with a summary, steps to reproduce, expected vs actual results, and severity.

8.2 Change Requests

• All changes affecting the module must be approved by the QA Lead and Project Manager. Major UI/UX changes require design approval.

9.0 Features to Be Tested

• Answer Creation: Posting valid answers and handling empty/invalid input.

• Answer Editing: Editing existing answers and validating edit restrictions (e.g., only author can edit).

• Answer Deletion: Deleting answers and ensuring proper removal from the display and database.

• Answer Display: Retrieving and correctly sorting the list of answers based on upvote counts.

• Voting: Upvoting and downvoting answers, restricting multiple votes by the same user, and updating vote counts accurately.

• Edge Cases: Handling actions by unauthorized users, duplicate submissions, and server errors.

10.0 Features Not to Be Tested

• User authentication (login/logout) is handled separately.

• Question creation/editing and advanced features such as tagging or notifications.

11.0 Resources / Roles & Responsibilities

• QA Lead: Oversees test plan and test execution.

• Developers: Write unit tests, fix defects, maintain code coverage.

• QA Engineer: Designs and executes integration/system tests, logs defects.

• Stakeholders: Conduct UAT and provide final acceptance.

12.0 Schedules

• Test Plan: March 20, 2025

• Test Cases & Test Scripts: March 23, 2025

• Test Incident Reports: Ongoing

• Test Summary Report: March 31, 2025

13.0 Significantly Impacted Departments (SIDs)

• Frontend Development

• Backend/API Team

• Database Administration (if applicable)

14.0 Dependencies

• Stable API endpoints and test environment availability.

• Working user login system if voting requires authentication.

15.0 Risks/Assumptions

• Risk: Backend instability or delays may impact test execution. Mitigation: Use mock services.

• Risk: Inconsistent environment setups. Mitigation: Provide clear configuration instructions.

• Assumption: Module requirements are final during test execution.

16.0 Tools

• Testing Frameworks: Jest for unit tests, Cypress/Selenium for E2E tests.

• Bug Tracking: Jira or GitHub Issues.

• Version Control: GitHub/GitLab.

• CI/CD (Optional): Jenkins or GitHub Actions for automated test runs.

Appendix

Sample Test Cases:

• TC-01: Verify that posting a valid answer displays it immediately.

• TC-02: Verify that editing an answer updates the content correctly.

• TC-03: Verify that deleting an answer removes it from the display and database.

• TC-04: Verify that upvoting increments the vote count by 1.

• TC-05: Verify that downvoting decrements the vote count by 1.

• TC-06: Verify that the answer list is sorted based on upvote counts (highest first).

• TC-07: Verify that a user cannot vote multiple times on the same answer.

• TC-08: Verify error messages when attempting actions.