

# Test Report

## Overview

The system was tested to ensure full functionality of user registration, login, post creation, replies (including nested replies), voting, and search. Both frontend and backend components were validated, along with integration and edge case handling.

## Testing Methods

### Manual Testing (Frontend + Backend)

- Verified user registration, login/logout.
- Checked channel creation and listing.
- Created posts with and without images.
- Replied to posts multiple levels deep.
- Tested upvote/downvote buttons for accurate count.
- Performed content and user search.
- Validated top user search output.

### API Testing (Postman & Fetch Calls)

- Sent manual POST, GET, and DELETE requests to test endpoints like:
- /api/users/register, /login, /logout
- /api/channels, /api/posts
- /api/posts/search, /api/posts/vote
- Confirmed proper authentication middleware behavior.
- Ensured responses matched expected structure and status codes.

## Edge Case Testing

- Prevented creation of blank channels or posts.
- Blocked unauthenticated access to protected routes.
- Simulated deleted users to check fallback names.
- Submitted malformed JSON to test server error handling.

### Session & Security Tests

- Ensured sessions expire correctly on logout.
- Verified user sessions persist between page reloads.
- Protected admin-only routes and verified non-admins could not access them.

### Summary

All core features performed as expected across multiple scenarios. The interface handled both user errors and system failures gracefully. Image uploads and nested post logic were stable under repeated interaction. The search system reliably parsed and displayed relevant results. Session-based login and admin privilege checks functioned correctly.