



Department of Computer Science and Engineering
(IoT Cyber Security with Block Chain)
B.Tech. Sem: VII Subject: Software Testing and Quality Analysis

Experiment 1

Name: Hiren Darji

SAP ID: 60019230114

Aim: -

Test Plan for Cloud-based Collaborative Code Editor

1. Test Plan Identifier

TCCCE-001-2025

2. Introduction

This test plan outlines the strategy, resources, schedule, and scope for testing the Cloud-based Collaborative Code Editor **CodeCollab**. **CodeCollab** is a web application enabling multiple users to write, edit, and manage code files collaboratively with real-time synchronization, user authentication, cloud storage, secure access, and project/file management.

Testing ensures the application is robust, user-friendly, performant under load, and meets security and business requirements.

3. Test Items

- User Authentication (signup, login, logout).
- Real-time collaborative code editing.
- Project management (create, rename, delete).
- File system (folders, files: create, edit, delete, organize).



- Editor functionalities (syntax highlighting, autocompletion, code formatting, undo/redo).
 - Multi-language code support.
 - User profile management.
 - Cloud storage (Firebase integration, data persistence).
 - Network/disconnection handling.
 - Browser compatibility.
 - Security (auth/session, XSS/CSRF prevention).
 - UI/UX: dark/light modes, responsive design.
-

4. Features to be Tested

- **Authentication**
 - Valid/significant/invalid sign-up, login, logout.
 - Error and alert handling.
- **Dashboard**
 - Project listing, recent file history, quick project/file access.
- **Projects**
 - Create, rename, delete projects.
 - Add/remove collaborators.
 - Open and display correct project data.
- **File & Folder Management**
 - Create, view, rename, delete files and folders.
 - Drag-and-drop hierarchy (if supported).
 - File/folder organization.
 - File language type switching.
- **Code Editor**
 - Syntax highlighting for supported languages.
 - Real-time updates between multiple users (cursor, text, selection sync).
 - Conflict handling for simultaneous edits.
 - Autocomplete and code formatting.
- **Collaboration**
 - Multi-user edits in same file/project.



- Permission control on files/folders.
 - **Profile**
 - View/update display name, avatar.
 - Correct email display; email invariant.
 - **Error Handling & Feedback**
 - All action confirmations, loading indicators, and error popups.
 - **Cloud Storage**
 - Auto-save of files/projects to Firebase.
 - Data fetched correctly after refresh or from another device.
 - **Security**
 - Only authorized users access respective resources.
 - Secure API and data transmission.
 - **Cross-Browser & Responsive Design**
 - Chrome, Firefox, Edge, Safari.
 - Desktop and mobile device responsiveness.
-

5. Features Not to be Tested

- External API services not yet implemented (third-party file storage, advanced analytics).
-

6. Approach

- **Manual Functional Testing:**
 - Each UI flow and edge case exercised by QA.
 - Coverage for positive/negative scenarios.
- **Automated Testing:**
 - Regression suite for auth, project/file CRUD, API endpoint validation.
- **Performance Testing:**



- Simulate many concurrent users editing a file/project; measure sync latency, UI responsiveness, Firebase data throughput.
 - **Security Testing:**
 - Manual auth bypass attempts, fuzz testing on endpoints.
 - **Compatibility/Accessibility Testing:**
 - Use accessibility tools; test in multiple browsers/devices.
-

7. Entry and Exit Criteria

Entry Criteria:

- Code freeze for current release.
- All critical bugs from previous cycles resolved.
- Test environment (dev/staging) deployed and accessible.
- Approved, documented test cases available.

Exit Criteria:

- 100% execution of planned test cases.
 - No critical/major open defects blocking workflows.
 - Test summary report delivered and signed off by QA Lead and Project Manager.
-

8. Suspension & Resumption Criteria

- Test cycle suspended if any critical/blocker bug halts basic usage (e.g., login fails, editor doesn't load).
 - Testing resumes once blocking defects are resolved/verified and build is stable.
-



9. Test Deliverables

- Master Test Plan document.
 - Detailed Test Case documents.
 - Defect/bug logs and reports.
 - Daily/weekly test execution reports.
 - Final QA signoff summary.
-

10. Schedule

Phase	Start Date	End Date
Test Planning	2025-08-14	2025-08-17
Test Design	2025-08-18	2025-08-22
Environment Setup	2025-08-23	2025-08-25
Test Execution	2025-08-26	2025-09-09
Defect Reporting	2025-08-26	2025-09-09
Regression/Closure	2025-09-10	2025-09-12
Final Signoff	2025-09-13	2025-09-14



11. Roles and Responsibilities

Role	Name	Responsibilities	Signature
Test Manager	Hiren Darji	Oversees the entire testing strategy, ensuring alignment with project goals and timely delivery of quality assurance milestones	[Signature]
QA Lead	Shourya Tamboskar	Leads the QA team by defining test processes, coordinating test efforts, and ensuring comprehensive coverage of functional and non-functional requirements	[Signature]
Test Engineer	Devarsh Chandiwade	Designs and executes test cases to validate collaborative editing features and real-time synchronization functionality	[Signature]
Test Engineer	Parth Shethia	Focuses on automation and performance testing to ensure scalability, responsiveness, and seamless user experience in the cloud environment	[Signature]



12. Environmental Needs

- Latest Chrome, Firefox, Edge, Safari browsers.
 - Stable, fast Internet connection.
 - Firebase dev database with seeded test users and data.
 - Node.js server running latest build.
 - Multiple test accounts (admin, collaborator, regular user).
 - Desktop, tablet, and mobile test devices.
-

13. Risks and Mitigation

- **Real-time Sync:** Network lags or conflicts—mitigate via robust backend testing and fallback logic.
 - **Third-Party Libraries:** Unexpected updates/changes—lock dependency versions, frequent upgrade reviews.
 - **Concurrency Issues:** Data merge issues—implement/verify conflict resolution logic.
 - **Cloud Outage:** Firebase/Cloud error—graceful error messaging to client, retry/backoff.
 - **Security:** Unauthorized file/data access—periodic pentesting, strict permission checks.
-

14. Approval

Prepared by: Hiren Darji
Date: 2025-08-14
