Comprehensive Test Plan: MVM Breach Platform

# 1. Project Overview

Project Name: Breach

Client: Metaverse Magna (MVM)

Objective: To ensure the Breach platform meets expected functional, UX, and non-functional requirements across frontend, backend, and real-time components.

# 2. Scope of Testing

In Scope:

- User registration and login

- Display and filtering of posts

- User interest selection during onboarding

- WebSocket event stream validation

- Full API validation (Swagger)

- Browser and mobile screen size compatibility

- Load and concurrency testing

Out of Scope:

- Localization testing

- Third-party integrations (e.g., email delivery if mocked)

# 3. Test Objectives

- Identify functional, UI/UX, and integration issues.

- Validate WebSocket real-time stream behavior.

- Ensure error messages are correctly handled and user-friendly.

- Assess system behavior under load.

# 4. Test Deliverables

- Test Plan (this document)

- Functional and exploratory test cases

- Bug reports (Google Sheets)

- Video recordings of test sessions

- Screenshots for defects

- Test summary and sign-off report

# 5. Test Approach

a. Manual Testing:

- Exploratory testing based on Figma designs

- Role-based testing (guest, registered user)

- Browser and device compatibility tests

b. API Testing:

- Postman collections for each endpoint

- Auth flows and response schema checks

c. WebSocket Testing:

- Postman/WebSocket UI for connection and message validation

- Stream latency and reconnection testing

d. Automation (optional):

- Basic smoke tests for login, navigation, and API status

- Tools: Cypress or Playwright

# 6. Tools & Environment

- Browsers: Chrome (primary), Firefox, Safari, Edge

- OS: macOS, Windows 11

- Tools: Postman, JMeter, Chrome DevTools, Loom (recording), Google Sheets

- Frontend: https://breach-fe.qa.mvm-tech.xyz

- Backend Swagger: https://breach-api.qa.mvm-tech.xyz/swagger

- WebSocket: wss://breach-api-ws.qa.mvm-tech.xyz

# 7. Entry & Exit Criteria

Entry Criteria:

- Access to staging environment

- Functioning test accounts

- Figma design as test basis

Exit Criteria:

- 100% test cases executed

- No critical or high severity bugs left unresolved

- All bugs documented and reviewed

# 8. Risk & Mitigation

Risk: No PRD or documentation

Mitigation: Rely on exploratory testing from Figma

Risk: Unstable WebSocket stream

Mitigation: Retry, document exact cases

Risk: Load tests disrupt QA env

Mitigation: Limit virtual users; test off-peak

# 9. Schedule & Timeline

Task | Date

---------------------------|------------

Test Planning | April 29

Exploratory/Manual Testing| April 29–30

API & WebSocket Testing | April 30–May 1

Load Testing | May 1

Bug Report & Documentation| May 2

Final Review & Submission | May 2–3

# 10. Test Case Sources

- Figma UI screens

- Swagger API definitions

- WebSocket event documentation (explored via Postman)

- Functional user flows such as:

- Register/Login

- Select Interest

- Filter Posts

- View Stream

- UX behavior for responsiveness and failure states