PicVerse Testing Test Plan

1. TEST PLAN IDENTIFIER

ID: TP-PICVERSE-2025-V1.0

2. REFERENCES

a. Project Plan Document (PicVerse)
a. Software Requirements Specification (SRS-PicVerse)
b. Design Document (PicVerse.pptx)
c. UI Wireframes (Figma: PicVerse v1.0)
d. IEEE 829 Standard for Software Test Documentation
e. UML Diagrams : PicVerse_UML_Diagrams.pdf

3. INTRODUCTION

The purpose of this document is to define the testing strategy and planning necessary for validating the **PicVerse platform**, an Instagram-inspired social media application for content sharing and interaction. The plan includes objectives, scope, resources, scheduling, entry and exit criteria, and deliverables.

Scope

- Functional testing of user-driven actions: registration, login, profile management, posts, and messaging.
- Integration testing across modules: frontend (Html,CSS,Javascript,ReactJS), backend (Flask), and database (MySQL).
- Compatibility testing on modern browsers and devices.

Constraints

- No cloud storage features currently supported.
- Admin-side analytics are not publicly released; hence, excluded from test scope.

4. TEST ITEMS (FUNCTIONS)

Module	Test Items
User Authentication	Registration, Login, Session Management
Content Sharing	Create Post, View Feed, Like, Comment
Messaging	Direct Messages, Chat History
User Profile	View, Edit, Upload Picture
User Connections	Search, Send/Accept Request, Remove
Discussion Forum	Create Threads, Reply, Moderate

PicVerse Testing Test Plan

5. FEATURES TO BE TESTED

A. User Management

- User Registration with OTP verification
- Secure Login and Logout flow
- Session management, expiration, and redirection behavior

B. Content Sharing & Interaction

- Post creation (text + image uploads)
- Edit/Delete post functionality
- Like and comment on posts (including UI updates and database reflection)
- Real-time post updates in feed

C. Messaging

- 1:1 text-based messaging
- Message sending and chat history validation
- Concurrent messaging between users

D. Discussion Forums

- Create new discussion threads
- View and participate in existing threads
- Commenting and reply validation
- Thread ownership and update permissions

E. Badge System

- Badge assignment by admin
- Badge revocation and update visibility
- Badge display on user profile

F. Analytics

- Real-time chart rendering and metric updates
- Accuracy of user and post statistics over time

G. Admin Operations

- Ban and unban users (status update and impact)
- Role-based access restrictions (admin vs regular user)

- User and thread moderation (if active)

H. Profile Management

- View and update profile info (bio, name, links)
- Profile picture upload and rendering
- Data consistency across profile and post history

I. System-Wide Functions

- Search functionality for users and threads
- Form validation for all input fields (email, password, post content)
- Error messaging for invalid or failed actions
- Responsive UI behavior on multiple screen sizes (mobile, tablet, desktop)
- Database consistency across modules (posts, messaging, bad

PicVerse Testing Test Plan

6. FEATURES NOT TO BE TESTED

- Load/stress testing (out of current scope)
- Video upload and playback (planned for v2.0)
- Browser compatibility for IE < 11
- Payment gateway (not applicable)
- Cross-platform responsiveness beyond desktop and Android

7. APPROACH (STRATEGY)

A combination of black-box functional testing, integration testing, and GUI-based exploratory testing will be employed. Selenium and Postman will be used for automated test scripting of frontend and backend APIs respectively. Manual tests will validate GUI behaviors. State diagrams and activity flows will guide behavioral tests.

8. ITEM PASS/FAIL CRITERIA

- A test case passes if actual results match expected results for valid inputs.
- A test case fails if there's a deviation or crash, UI glitch, or inconsistent backend response.
- For APIs: HTTP 2xx expected for success, 4xx/5xx for failures.

9. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

- Testing will be suspended in case of database connection loss, major application crash, or OTP system failure.
- Resumption requires resolution confirmation and re-execution of previously failed test cases.

10. TEST DELIVERABLES

- Test Plan Document
- Test Cases (Excel/TestRail format)
- Test Data Sets (users, posts, messages)
- Bug Reports and Logs
- Automated test scripts (Python/Selenium, Postman)
- Final Test Summary Report

PicVerse Testing Test Plan

11. REMAINING TEST TASKS

- Write & execute test cases for Messaging and Forum modules
- Validate API responses for post creation, comments, likes
- Perform regression testing after each major patch
- Capture UI screenshots for evidence

12. ENVIRONMENTAL NEEDS

Component	Description
Web Browsers	Chrome v120+, Firefox v112+
OS	Windows 10 / Ubuntu 22.04 LTS
Backend Server	Flask on localhost:5000
Frontend Server	ReactJS on localhost:3000
Database Server	MySQL on port 3306
Tools	Git, Selenium, VSCode ,Figma,Taiga

13. STAFFING AND TRAINING NEEDS

As this is a student-led project, all team members are actively involved in both development and testing. No additional staffing or formal training is required.

Each member has hands-on experience with the technologies used in the project, including **Flask**, **Python**, and the **custom API for messaging**. Since the system was collaboratively designed and built, the team is already familiar with its structure and functionality, making additional onboarding unnecessary,

PicVerse Testing Test Plan

14. RESPONSIBILITIES

Role	Responsibility
QA Team	Execute test cases, report bugs
Development Team	Fix issues and support retesting
Product Owner	Review test outcomes and approve
Mentor/Guide	Supervise and evaluate testing docs

15. SCHEDULE

Phase	Activity	Tentative Timeline
Sprint 1	Test Planning and Test Case Design	Week 1
Sprint 2	Feature Testing: Registration & Login	Week 2
Sprint 3	Testing: Post Creation & Feed Display	Week 3–4
Sprint 4	Messaging and Interaction Testing	Week 5–6
Sprint 5	Forum and Analytics Feature Testing	Week 7–8
Sprint 6	Badge Management and Moderation Testing	Week 9–10
Final Sprint (System Testing)	End-to-End Testing, Load/Stress Testing	Week 11
Bug Fix & UAT	Bug Resolution, User Acceptance Testing	Week 12

PicVerse Testing Test Plan

16. PLANNING RISKS AND CONTINGENCIES

1. Lack of Personnel Resources During Test Phase

***Risk**:* Some QA or team members may not be available during the test window due to exams, academic pressure, or personal reasons.

Contingency:

- Reassign development members to assist with testing tasks.
- Slightly extend the testing timeline if it doesn't impact final submission deadlines.

2. Unavailability of Hardware, Software, or Tools

***Risk**:* Tools like Postman (if used), local Flask servers, mobile devices, or necessary browser versions may not be available or functional.

Contingency:

- Use alternate systems, browsers, or platforms.
- Replace unavailable services with mock APIs or offline test data.
- If a specific tool isn't used (e.g.,), skip tests dependent on it and focus on in-browser or unit-level testing.

3. Delayed Delivery of Software Components

Risk: Some modules (like Messaging, Forum, or Real-Time Features) may be incomplete or unstable when testing is scheduled to start.

Contingency:

- Mark such modules as "partially tested" in documentation.
- Prioritize test coverage for the remaining completed features.
- Reduce the number of test cases for late components.

4. Ambiguous or Misunderstood Requirements

Risk: Miscommunication between developers and mentors may lead to vague or incomplete feature expectations, especially for usability and edge cases.

Contingency:

- Clarify requirements through group discussions or tutor feedback.
- Update test cases dynamically as understanding improves.

PicVerse Testing Test Plan

5. Historical Defect Clustering in Certain Modules

Risk: Modules like Post Creation, Authentication, and Session Handling have shown repeated bugs in the past.

Contingency:

- Focus additional testing on historically buggy areas.
- Track past bugs and check for regressions.

6. Third-party Dependency: OTP Delivery

Risk: OTPs sent via email or SMS might be delayed due to network or external service issues.

Contingency:

- Allow OTP timeout buffers in testing.
- Simulate OTP verification during internal tests.

7. Image Upload Failures

Risk: Uploading large or unsupported image files may cause server or client errors.

Contingency:

- Validate file types and sizes in both frontend and backend.
- Perform network condition testing to check user impact.

8. Session and Token Issues

Risk: Improper session handling can result in security risks like unauthorized access or forced logouts.

Contingency:

- Add unit tests to verify token creation, expiry, and invalidation.
- Test login/logout flows under different browsers and tabs.

9. Data Inconsistency from Concurrent User Actions

Risk: Simultaneous user activity (like multiple users commenting or liking posts) might cause race conditions.

Contingency:

- Use optimistic concurrency or locks if required.
- Test under simulated load to catch synchronization issues.

10. Common Web Security Risks (XSS, SQLi, CSRF)

Risk: Poor input validation can expose the system to common attack vectors.
Contingency:

- Perform security-focused testing on form inputs and endpoints.
- Sanitize user input and use secure session handling techniques.

17. APPROVALS

Before the PicVerse testing process can move forward, this test plan needs to be reviewed and approved by key team members and mentors. These approvals confirm that the test coverage is complete, the plan makes sense, and the team is ready to begin formal testing.

The following individuals are responsible for reviewing and approving this test plan:

- **QA Lead – Tisha Jain**
  Responsible for reviewing the overall testing strategy to ensure thorough test case coverage and alignment with project goals.

- **Lead Developer – Kaival Shah**
  Confirms that the application is functionally stable, major features are implemented, and the codebase is ready for structured testing.

- **Group Leader – Dhruv Patel**
  Oversees coordination between development and testing, ensures that project milestones are being met, and reviews the test plan from a project management standpoint.

It's important that both technical and non-technical reviewers are involved, since developers may not always see issues from a user's perspective, and users may not fully understand the technical side. Involving both ensures a balanced and realistic plan.

Once all reviewers sign off, the testing phase can begin, followed by bug fixes, final review, and project submission.