**User Acceptance Testing (UAT) Template**

|  | |
| --- | --- |
| Date | 29 June 2025 |
| Team ID | SWUID20250177139 |
| Project Name | Social Media App |
| Maximum Marks |  |

**Project Overview:**

Project Name: Social Media App

Description: This social media platform is built using the MERN stack (MongoDB, Express.js, React.js, Node.js) and offers real-time communication through Socket.io. Designed for students, freelancers, and remote collaborators, the app features secure messaging, post and story sharing, user profiles, and notifications. It combines modern UI with strong data privacy to deliver a seamless and interactive online experience.

Project Version:v1.0.0

Testing Period: 2025-06-29 to 2025-07-02

**Testing Scope:**

User Authentication

* Test registration, login, logout, and password validation.
* Check role-based access and token/session management.

Real-Time Messaging

* Verify message delivery, typing status, and notifications using Socket.io.
* Check file/media sharing in chats.

Post and Story Functionality

* Test post/story creation, editing, and deletion.
* Validate story expiry after 24 hours.

Profile Management

* Verify updating profile picture, bio, and personal details.
* Ensure data persists and reflects across the app.

Save & Explore Posts

* Test post saving, unsaving, and exploring new content.
* Check trending posts and personalized suggestions.

Likes, Comments, Shares

* Test interactions and real-time updates.
* Verify comment threads and share functionality.

Notifications

* Validate in-app and real-time notifications for likes, comments, follows, and messages.

Mobile Responsiveness & UI

* Test the app across different screen sizes and devices.
* Validate responsive design and accessibility.

Data Security

* Ensure encryption in data transmission (e.g., JWT, HTTPS).
* Validate that private data is not exposed.

Performance Testing

* Test app behavior under multiple user connections.
* Measure server response time and load handling.

### **Testing Environment**

The testing environment includes all necessary hardware, software, and configurations required to test the application effectively.

**1. Frontend**

* **Technology**: React.js (with Socket.io-client, Axios)
* **Browser Support**: Chrome, Firefox, Edge
* **Device Testing**: Desktop, Tablet, Mobile (using responsive dev tools or emulators)

**2. Backend**

* **Technology**: Node.js, Express.js, Socket.io
* **Database**: MongoDB (local or cloud via MongoDB Atlas)
* **Authentication**: JWT-based session/token handling

**3. Tools**

* **Code Editor**: VS Code
* **Version Control**: Git, GitHub
* **Testing Tools**:
  + Postman (API testing)
  + Mocha/Chai or Jest (unit testing)
  + Browser Developer Tools (UI/debug testing)
  + Cypress or Selenium (optional: end-to-end testing)

**4. Environment Configuration**

* **Operating System**: Windows 10+ / macOS / Linux
* **Node Version**: >= 14.x
* **Package Manager**: npm or yarn
* **Ports Used**:
  + Client: localhost:3000
  + Server: localhost:5000
  + MongoDB: localhost:27017 or MongoDB Atlas URI

**Test Cases:**

| **Test Case ID** | **Test Case Description** | **Input** | **Expected Output** | **Status** |
| --- | --- | --- | --- | --- |
| TC001 | Register a new user | Valid username, email, password | User registered and redirected to login | Pass/Fail |
| TC002 | Login with valid credentials | Registered email and password | Login successful, redirected to homepage | Pass/Fail |
| TC003 | Create a new post | Text + Image | Post visible in user feed | Pass/Fail |
| TC004 | Like a post | Click like button | Like count increases by 1 | Pass/Fail |
| TC005 | Comment on a post | Enter comment text and submit | Comment appears under post | Pass/Fail |
| TC006 | Send real-time message | Message in chat input | Message appears in real-time in chat box | Pass/Fail |
| TC007 | Upload and view story | Upload image/video | Story displays for 24 hours | Pass/Fail |
| TC008 | Search and follow another user | Enter username in search | Follow button visible and functional | Pass/Fail |
| TC009 | Save a post | Click save icon | Post added to saved section | Pass/Fail |
| TC010 | Verify story auto-delete after 24 hours | Upload story, wait 24 hours | Story automatically deleted | Pass/Fail |

**Bug Tracking:**

| **Bug ID** | **Bug Description** | **Steps to Reproduce** | **Severity** | **Status** | **Additional Feedback** |
| --- | --- | --- | --- | --- | --- |
| BUG001 | Login form crashes on invalid input | 1. Go to Login page 2. Enter @#\*%! as username 3. Click Login | Medium | Open | Add input validation for special characters. |
| BUG002 | Real-time chat message not delivered | 1. Open chat 2. Send message during network drop 3. Restore connection | High | In Progress | Implement message queue for offline scenarios. |
| BUG003 | Post image not showing after upload | 1. Create post with image 2. Submit 3. Check feed | Medium | Resolved | Image URL wasn’t updating—fixed in latest patch. |
| BUG004 | Stories not expiring after 24 hours | 1. Upload story 2. Wait over 24 hours 3. Refresh page | High | Open | Set a background cron job or TTL-based deletion. |
| BUG005 | Notification not shown for new comment | 1. Comment on a post from another account 2. Wait for notification | Medium | In Progress | Check Socket event or database trigger issue. |
| BUG006 | Profile update not saving | 1. Edit bio or profile picture 2. Click Save 3. Refresh page | Medium | Open | Possibly failing to update user record in MongoDB. |
| BUG007 | Post save button unresponsive | 1. Click Save on a post 2. Refresh 3. Post not in saved section | Low | Open | Check frontend state vs backend save request status. |
| BUG008 | Like count not incrementing in real-time | 1. Like post 2. Open same post on another account 3. Count not updated instantly | Medium | Open | Socket event may not be emitting to all users. |

**Sign-off:**

Tester Name: Yerasi Lohitha

Date: 01-07-2025

Signature:Y.Lohitha

**Notes:**

* Ensure that all test cases cover both positive and negative scenarios.
* Encourage testers to provide detailed feedback, including any suggestions for improvement.
* Bug tracking should include details such as severity, status, and steps to reproduce.
* Obtain sign-off from both the project manager and product owner before proceeding with deployment.