**User Acceptance Testing (UAT)**

|  |  |
| --- | --- |
| Date | June 27, 2025 |
| Team ID | LTVIP2025TMID59169 |
| Project Name | Citizen AI – Intelligent Citizen Engagement Platform |
| Maximum Marks |  |

**Project Overview:**

**Project Name:** Citizen AI – Intelligent Citizen Engagement Platform

**Project Description:** **CitizenAI** is a smart chatbot and analytics system that helps citizens get instant answers to their everyday questions about government services, schemes, exams, and support—just by chatting.

This platform uses **Artificial Intelligence (AI)** to understand what people are asking and give them the most accurate answers. It also checks how people feel (happy, confused, or frustrated) through their messages and shows that as reports on a dashboard.

You can access CitizenAI through a **website, mobile app, or messaging** (like SMS or chat). It includes:

* A chatbot that talks like a human,
* A brain (AI model) that finds and gives answers smartly,
* A database that stores citizen questions and answers,
* A dashboard that shows how citizens feel and what topics matter most.

Whether a student asks how to apply for an exam, or a farmer wants to know about a government subsidy, **CitizenAI is like a 24/7 helper for everyone**.

Project Version: Version 1.0

Testing Period: June 7,2025 to June 26,2025

**Testing Scope:**

**1. List of Features and Functionalities to be Tested**

These are the main features or tools that users will interact with and you need to test if they work properly.

Example list for your **CitizenAI project**:

* Chatbot answers citizen queries accurately
* Sentiment analysis detects positive, negative, or neutral messages
* Dashboard shows graphs and trends correctly
* User input validation (e.g., empty messages, wrong formats)
* API connection to IBM Granite model works
* File uploads (if used) don’t crash the system

**2. List of User Stories or Requirements to be Tested**

These are short, simple statements that describe **what a user wants** and **what the system should do**.

Example list for your project:

* “As a citizen, I want to ask the chatbot questions so I can get instant help.”
* “As an admin, I want to see the user sentiment on a dashboard so I can track complaints.”
* “As a developer, I want the chatbot to respond within 3 seconds so users don’t wait long.”
* “As a user, I want the system to show errors if I leave the question field empty.”

**Testing Scope:**  
**Features to be Tested:**

* Chatbot Q&A
* Sentiment analysis
* Dashboard visualizations
* API response
* Input validation

**User Stories to be Tested:**

* Chatbot must give useful replies
* User emotions must be correctly shown
* Dashboard must display accurate data
* Errors must appear for wrong inputs

**Testing Environment:**

URL/Location: http://localhost:5000/

(If hosted online, it can be something like http://citizenai.live or your hosting link)

**Test Cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC-001 | |  | | --- | |  |  |  | | --- | | Verify chatbot responds to citizen questions | | 1. Open chatbot interface 2. Type a valid question like “How to apply for a PAN card?” 3. Click send | Chatbot provides accurate and relevant answer | Chatbot provided correct information on PAN card application | Pass |
| TC-002 | Verify chatbot handles empty message input | 1. Open chatbot interface 2. Leave input blank 3. Click send | Chatbot shows an error like “Please enter a message.” | Error message displayed correctly | Pass |
| TC-003 | |  | | --- | |  |  |  | | --- | | Check sentiment analysis of a positive message | | 1. Enter a message like “I love this service!” 2. Submit it | Sentiment should be detected as positive | Sentiment was shown as positive | Pass |

**Bug Tracking:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bug ID** | **Bug Description** | **Steps to reproduce** | **Severity** | **Status** | **Additional feedback** |
| BG-001 | Chatbot crashes when user uploads large PDF files | 1. Open the chatbot  2. Upload a PDF file larger than 10MB  3. Click on "Submit" | High | In Progress | App should show error message instead of crashing |
| BG-002 | Sentiment not detected for short messages | 1. Type "Good" or "Bad"  2. Send it | Low | Closed | Added logic to detect even short emotion words |
| BG-003 | Dashboard graphs not loading on slow internet | 1. Open dashboard on a slow connection  2. Check if graphs load properly | Medium | |  | | --- | | In Progress |  |  | | --- | |  | | |  | | --- | |  |  |  | | --- | | Add a message like “Loading…” while graphs are loading | |

**Sign-off:**

**Tester Name:** [Name of Tester]

**Date:** [Date of Test Completion]

**Signature:** [Tester's Signature]

**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.