

Project Report Format

1. **INTRODUCTION**
 - 1.1 Project Overview
 - 1.2 Purpose
2. **IDEATION PHASE**
 - 2.1 Problem Statement
 - 2.2 Empathy Map Canvas
 - 2.3 Brainstorming
3. **REQUIREMENT ANALYSIS**
 - 3.1 Customer Journey map
 - 3.2 Solution Requirement
 - 3.3 Data Flow Diagram
 - 3.4 Technology Stack
4. **PROJECT DESIGN**
 - 4.1 Problem Solution Fit
 - 4.2 Proposed Solution
 - 4.3 Solution Architecture
5. **PROJECT PLANNING & SCHEDULING**
 - 5.1 Project Planning
6. **FUNCTIONAL AND PERFORMANCE TESTING**
 - 6.1 Performance Testing
7. **RESULTS**
 - 7.1 Output Screenshots
8. **ADVANTAGES & DISADVANTAGES**
9. **CONCLUSION**
10. **FUTURE SCOPE**
11. **APPENDIX**

Source Code(if any) Dataset
Link
GitHub & Project Demo

1.INTRODUCTION

CITIZEN AI – Intelligent Citizen Engagement Platform

CITIZEN AI is a next-generation, AI-powered platform designed to revolutionize how governments and public institutions engage with their citizens. By leveraging artificial intelligence, data analytics, and modern communication technologies, CITIZEN AI enables transparent, responsive, and personalized interactions between citizens and governance systems. The platform acts as a digital bridge—streamlining service delivery, automating public query responses, analyzing community needs, and enhancing decision-making through real-time feedback and predictive insights. Whether it's handling civic complaints, participating in policy discussions, or accessing government services, CITIZEN AI empowers both citizens and authorities with smarter, more efficient tools for governance.

With features such as multilingual natural language processing, sentiment analysis, chatbots, automated case routing, and data dashboards, CITIZEN AI ensures inclusive, equitable, and accountable governance—fostering trust and active civic participation in the digital age.

1.1.Project Overview

CITIZEN AI is a comprehensive digital solution designed to modernize and simplify citizen engagement. It integrates Artificial Intelligence, Natural Language Processing (NLP), and data analytics to provide a centralized platform for managing public communication, feedback, complaints, and service requests.

The platform acts as a two-way interface:

- **For Citizens** – It offers easy access to information, allows complaint registration, tracks service requests, and facilitates participation in public decisions.
- **For Authorities** – It provides intelligent dashboards, automated response tools, analytics reports, and sentiment tracking for informed decision-making.

Key Features:

- **AI Chatbots** for 24/7 citizen support.
- **Multilingual Communication** using NLP.

Benefits:

- Increased transparency and accountability.
 - Faster resolution of citizen grievances.
-

Target Users:

- Municipal Corporations.
 - Government Departments.
-

Technologies Used:

- Artificial Intelligence & Machine Learning.
- Natural Language Processing (NLP).

1.2.Purpose:

The purpose of the **CITIZEN AI – Intelligent Citizen Engagement Platform** is to **modernize and streamline the interaction between government bodies and citizens** through the use of artificial intelligence and digital technologies. This platform aims to **create a transparent, responsive, and citizen-centric governance ecosystem** that fosters trust, inclusivity, and active civic participation.

- **Bridge the communication gap** between citizens and government by providing real-time, AI-powered engagement tools.
- **Simplify access to public services**, ensuring faster response times and efficient issue resolution.
- **Enable data-driven governance** by collecting and analyzing citizen feedback, complaints, and behavior patterns.
- **Improve public service delivery** through automation, smart routing of requests, and performance monitoring.
- **Encourage participatory democracy** by giving citizens a voice in decision-making processes via digital platform

2.Ideation Phase

2.1.Define the Problem Statements

Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	2 Marks

Customer Problem Statement Template:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

I am	Describe customer with 3-4 key characteristics - who are they?	Describe the customer and their attributes here
I'm trying to	List their outcome or "job" the care about - what are they trying to achieve?	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way - what bothers them most?	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists - what needs to be solved?	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view - how does it impact them emotionally?	Describe the emotions the result from experiencing the problems or barriers

Example:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	I am a citizen and user of public services.	communicate with government departments, access services.	The current system is slow, unresponsive.	it relies on outdated manual processes, disconnected channels.	frustrated, unheard, and distrustful of public institutions.

2.Ideation Phase

2.2 Empathize & Discover

Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	4 Marks

Empathy Map Canvas:

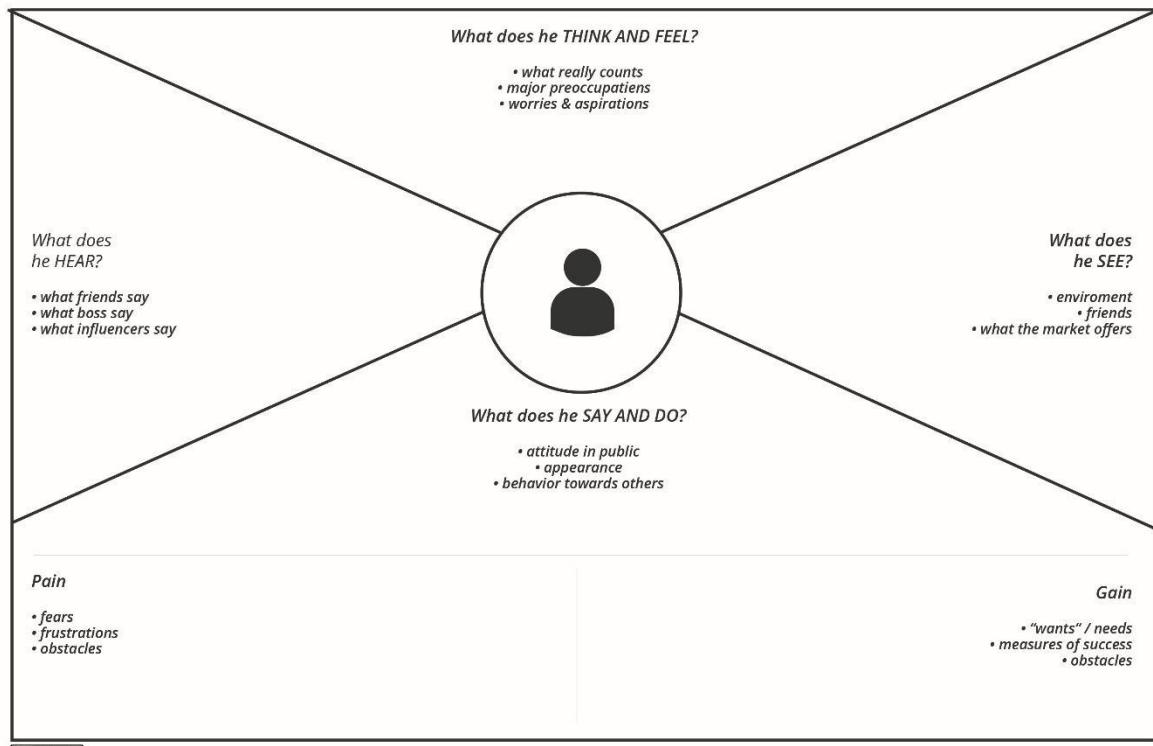
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to help teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Example:

Empathy Map



2.Ideation Phase

2.3Brainstorm & Idea Prioritization Template

Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	4 Marks

Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Step-1: Team Gathering, Collaboration and Select the Problem Statement

The screenshot shows a digital template for a brainstorming session. It is divided into three main vertical sections:

- Left Column:** A large icon of a lightbulb inside a speech bubble, followed by the title "Brainstorm & idea prioritization". Below the title, a paragraph explains the purpose: "Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room." At the bottom, there are three icons with corresponding text: "10 minutes to prepare", "1 hour to collaborate", and "2-8 people recommended".
- Middle Column:** A section titled "Before you collaborate" with a timer icon and the number "1". It contains instructions for preparation: "A little bit of preparation goes a long way with this session. Here's what you need to do to get going." Below this are three numbered steps:
 - Team gathering**: "Define who should participate in the session and send an invite. Share relevant information or pre-work ahead."
 - Set the goal**: "Think about the problem you'll be focusing on solving in the brainstorming session."
 - Learn how to use the facilitation tools**: "Use the Facilitation Superpowers to run a happy and productive session." This step includes a "Open article" button.
- Right Column:** A section titled "Define your problem statement" with a timer icon and the number "1". It asks, "What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm." Below this is a box labeled "PROBLEM" containing the placeholder text "How might we [your problem statement]?".

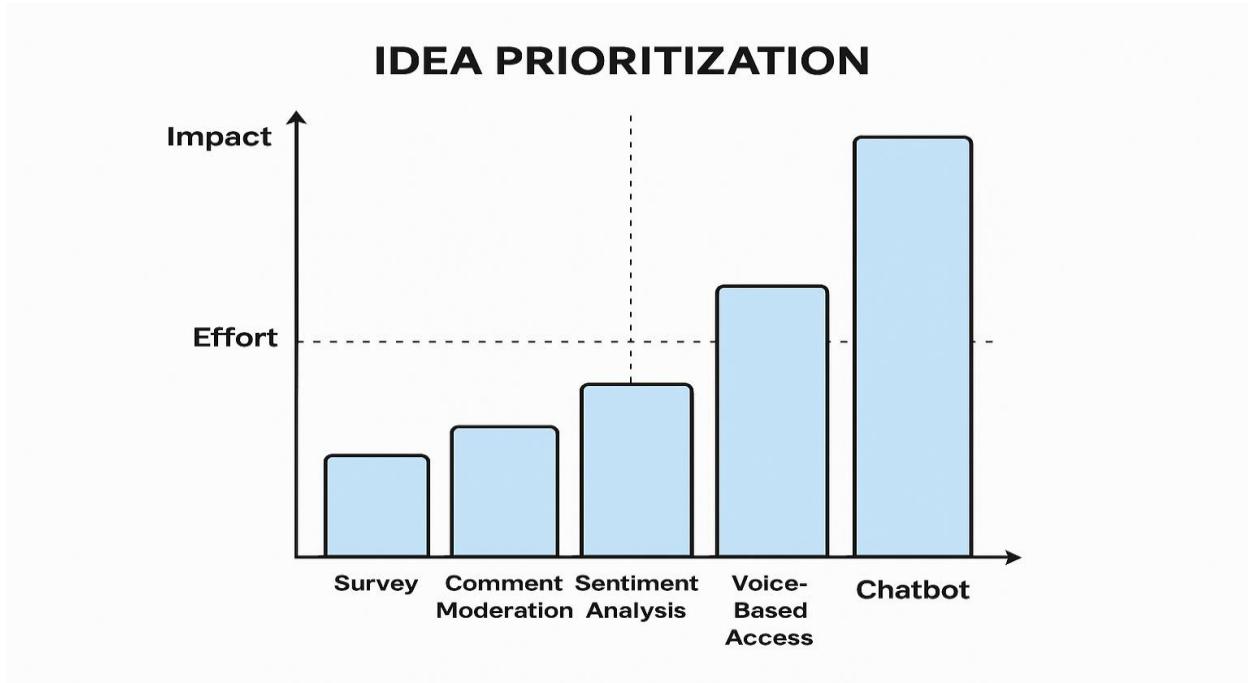
At the bottom right, there is a separate box titled "Key rules of brainstorming" with a timer icon and the number "1". It lists six rules with accompanying icons:

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping

The image shows two side-by-side screenshots of a digital application for idea management. On the left, under the heading 'Brainstorm', there is a grid of sticky notes for eight different users: Person 1 through Person 8. Each user's section contains a 4x4 grid of yellow sticky notes. A 'Tip' box at the top right suggests selecting a sticky note and dragging it to another to move it. On the right, under the heading 'Group Ideas', there is a single column for 'Person 4'. A 'Tip' box at the top right suggests taking turns sharing ideas while clustering similar ones. The interface is clean with a white background and light gray borders for the sections.

Step-3: Idea Prioritization



3. REQUIREMENT ANALYSIS

3.1.Customer Journey map

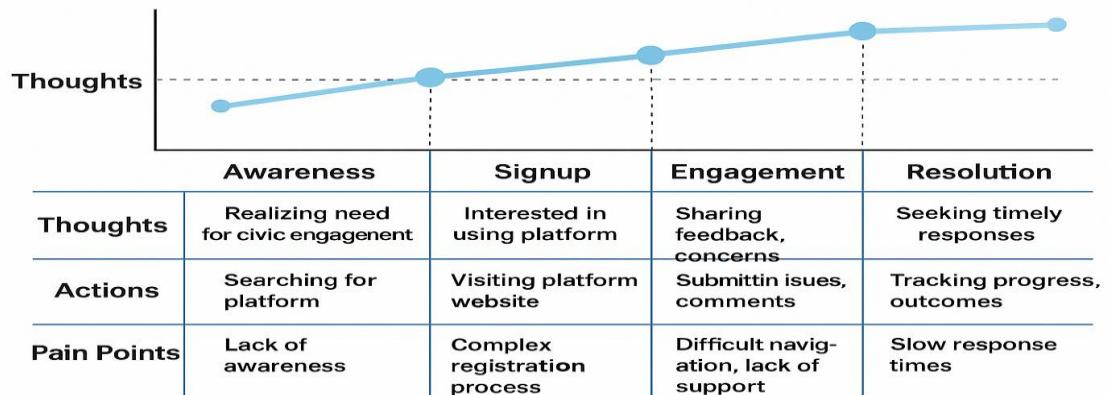
Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	4 Marks

Customer journey map:

STAGE	THOUGHTS	ACTIONS	PAIN POINTS	OPPORTUNITIES
1.AWARENESS	“I want to raise my issue or give feedback.”	Searching online, hearing from peers or media	Lack of awareness about the platform	Promote through social media, local ads, and community events
2. SIGNUP	“This might help me. I'll try it out.”	Visiting the website or downloading the app	Complicated sign-up or unclear instructions	Offer easy onboarding, multilingual UI, and simple sign-up forms
3.ENGAGEMENT	“Let me report my issue or share feedback.”	Using chatbot, submitting complaints, filling surveys	Confusing UI, no guidance, lack of support	AI-guided chat, tooltips, 24/7 virtual assistant support
4.RESOLUTION	“Is my problem being addressed?”	Tracking complaint status, checking updates	Delayed response, no visibility on next steps	Real-time updates, SMS/app notifications, estimated timelines
5. FEEDBACK	“Did my input make a difference?”	Leaving a rating or feedback comment	No way to give feedback or see impact	Public dashboards, feedback loop with acknowledgments

CUSTOMER JOURNEY MAP

Citizen AI – Intelligent Citizen Engagement Platform



Project Design Phase-II

3.2.Solution Requirements (Functional & Non-functional)

Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	<ul style="list-style-type: none"> ▪ Registration through Form ▪ Registration through Gmail ▪ Registration through LinkedIN
FR-2	User Confirmation	<ul style="list-style-type: none"> ▪ Confirmation via Email ▪ Confirmation via OTP
FR-3	Complaint Management	<ul style="list-style-type: none"> ▪ Raise complaint with category & location ▪ Upload images/documents ▪ Auto-ticket ID generation ▪ Track complaint status
FR-4	AI Chatbot Support	<ul style="list-style-type: none"> ▪ Multilingual chatbot for query resolution ▪ Predefined and dynamic responses ▪ Escalation to live agent if unresolved
FR-5	Dashboard & Notifications	<ul style="list-style-type: none"> ▪ User dashboard to view complaint history ▪ Real-time push/email/SMS notifications ▪ Analytics for issue trends
FR-6	Feedback & Rating	<ul style="list-style-type: none"> ▪ Submit service rating after resolution ▪ Provide open-text feedback ▪ Rate chatbot or agent interaction

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

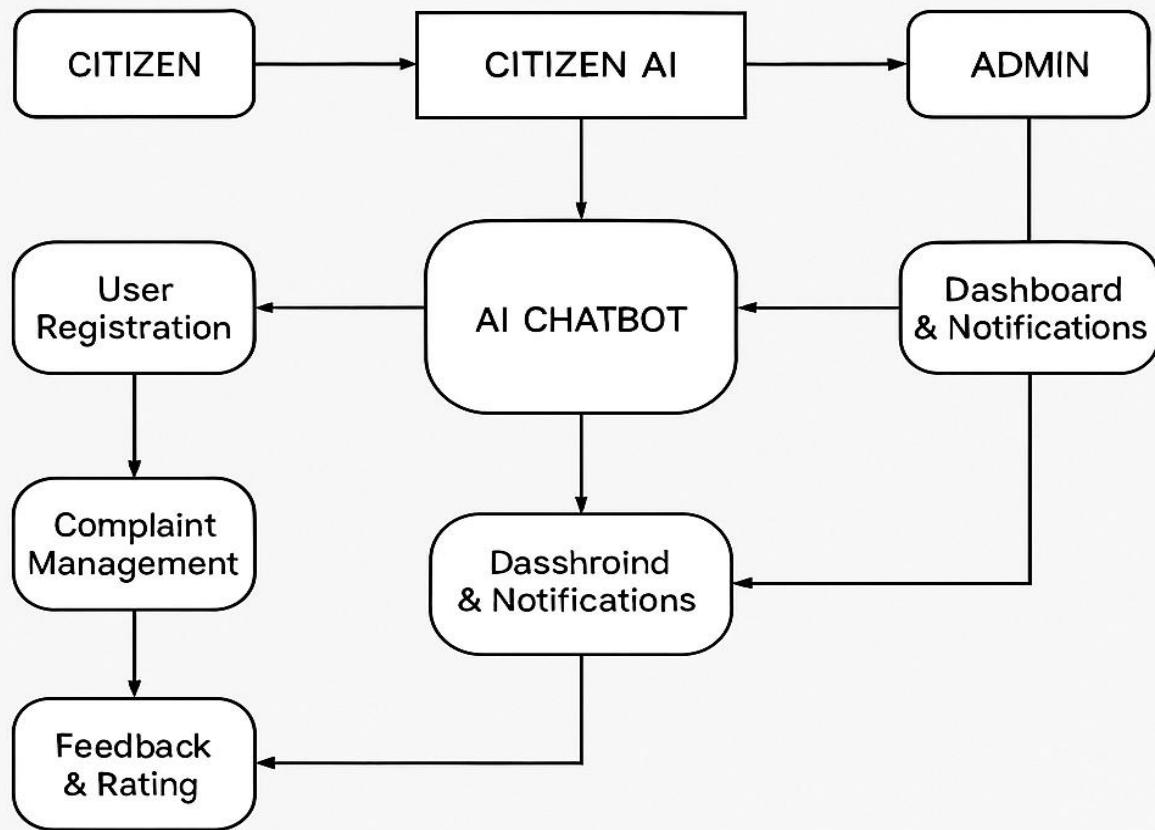
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The platform should have a user-friendly interface that is intuitive, multilingual, and accessible to users.
NFR-2	Security	The system must ensure end-to-end encryption, user authentication, and secure data storage to protect sensitive citizen data from unauthorized access or breaches.
NFR-3	Reliability	The system should operate consistently under expected conditions with minimal downtime, ensuring accuracy in issue logging, data processing, and communication.
NFR-4	Performance	The platform must respond to user inputs within 2 seconds and handle high volumes of simultaneous users without latency or system slowdowns.
NFR-5	Availability	The system should be available 24/7 with at least 99.9% uptime to ensure uninterrupted citizen access and engagement.
NFR-6	Scalability	The platform should be able to scale both horizontally and vertically to accommodate increasing numbers of users.

3.3.Data Flow Diagram & User Stories

Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	4 Marks

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories – CITIZEN AI

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Citizen	User Registration	US-1	As a citizen, I want to register via email so that I can access the platform.	Registration form is visible; successful submission triggers confirmation.	High	MVP
Citizen	User Registration	US-2	As a citizen, I want to register using my Gmail account for quick access	Gmail login integration works and creates a user profile.	Medium	V1.1
Admin	Admin Dashboard	US-3	As an admin, I want to view all user complaints in a dashboard format.	Dashboard displays all complaints by department/status.	High	MVP
Admin	Ticket Assignment	US-4	As an admin, I want to assign complaints to the right department.	Admin can manually or automatically assign based on category	High	MVP
Admin	Analytics	US-5	As an admin, I want to analyze complaint patterns using charts.	Graphs show trends by time, region, category.	Medium	V1.3

Project Design Phase-II

3.4. Technology Stack (Architecture & Stack)

Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

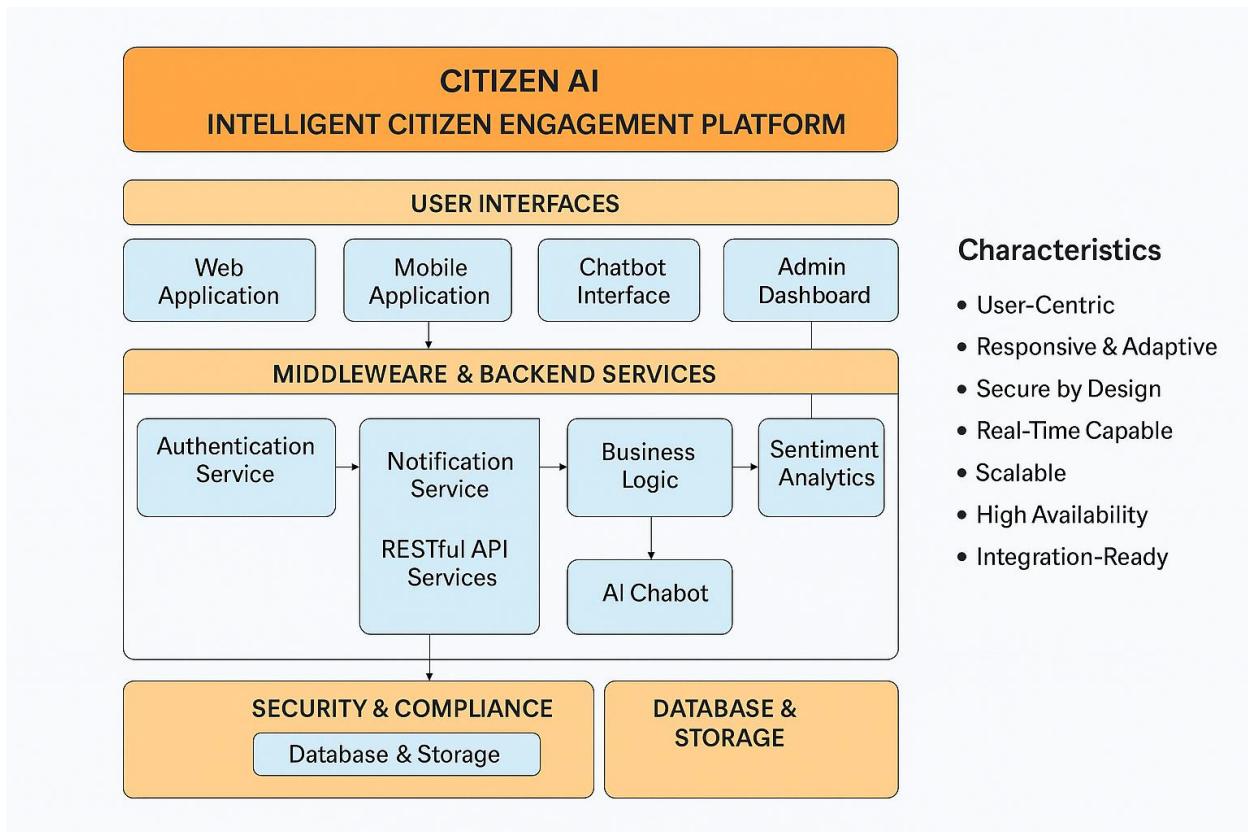
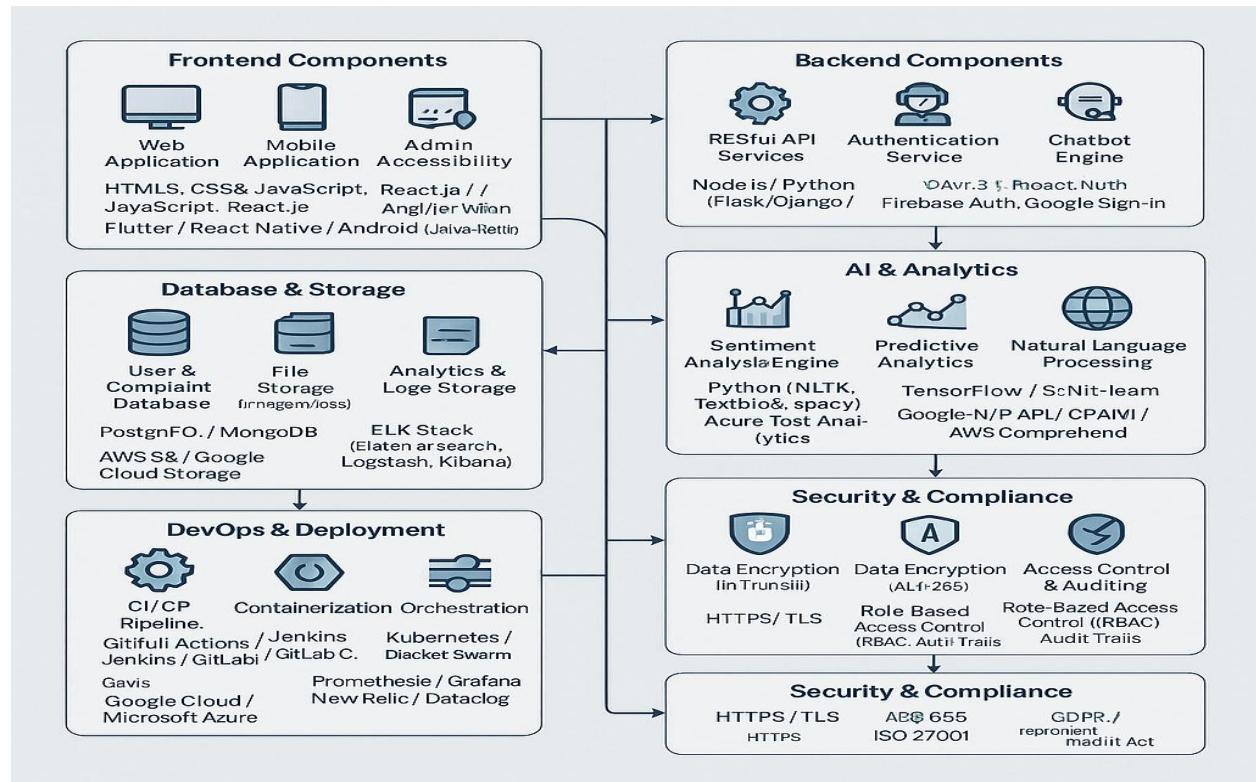


Table 1: components & technologies

COMPONENT AREA	KEY TECHNOLOGIES
FRONTEND (UI)	HTML5, CSS3, JavaScript, React.js / Angular, Flutter, React Native, Web Speech API
BACKEND	Node.js, Python (Flask/Django), Spring Boot, OAuth 2.0, Firebase Auth, Dialogflow, Microservices
DATABASE & STORAGE	PostgreSQL, MongoDB, AWS S3, Redis, ELK Stack
AI & ANALYTICS	NLTK, spaCy, TensorFlow, Azure Text Analytics, OpenAI, Google NLP API
DEVOPS & DEPLOYMENT	Docker, Kubernetes, GitHub Actions, Jenkins, AWS, Google Cloud, Prometheus, Grafana
SECURITY & COMPLIANCE	HTTPS/TLS, AES-256, RBAC, GDPR, ISO 27001

Table 2: Application characteristics

Characteristic	Description
User-Centric	Seamless, intuitive UX for all users; includes multilingual and accessibility support.
AI-Driven	Uses AI for chatbot, sentiment analysis, workflow automation, and predictive analytics.
Responsive & Adaptive	Works smoothly across web/mobile; adapts to user behavior for personalization.
Modular Architecture	Based on microservices for easy scaling, updates, and maintainability.
Secure by Design	Incorporates robust security protocols and complies with privacy laws like GDPR.
Real-Time Capable	Supports live updates, real-time notifications, and instant chat responses.
Scalable	Can support a growing user base across regions without affecting performance.
High Availability	24/7 uptime through load balancing, auto-scaling, and failover strategies.
Data-Driven	Captures and analyzes user data to inform policy and service improvements.
Multi-Channel Access	Usable via web, mobile apps, chatbots, SMS, WhatsApp, and voice interfaces.
Integration-Ready	Connects with legacy systems, APIs, and smart city solutions seamlessly.
Low Maintenance Overhead	Designed for automated monitoring, testing, and deployment to minimize manual work.



4. Project Design Phase

4.1. Problem – Solution Fit Template

Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	2 Marks

Problem – Solution Fit Template:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why.

Purpose:

- Solve complex problems in a way that fits the state of your customers.
- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- Sharpen your communication and marketing strategy with the right triggers and messaging.
- Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
- Understand the existing situation in order to improve it for your target group.

PROBLEM-SOLUTION FIT CANVAS

Target Users <ul style="list-style-type: none">• Citizens seeking access to government services and issue resolution• Government officials managing public grievances and feedback	User Problems <ul style="list-style-type: none">• Difficulty accessing services and filing complaints• Long wait times and no visibility on complaint status• Lack of trust in public systems• Limited access through rural or non-digital channels	Key Goals <ul style="list-style-type: none">• Make citizen engagement seamless and digital• Automate routine responses and reouting• Improve response time and citizen trust
Core Features <ul style="list-style-type: none">• AI-powered chatbot (multilingual)• Complaint management system• Real-time tracking and notifications	How Solution Fits <ul style="list-style-type: none">• Provides a single access point for all citizens• Automates common queries and ticketing• Ensures transparency and faster resolution	Proposed Solution <p>CITIZEN AI – A digital, AI-driven citizen engagement platform that enables smart interaction, real-time complaint resolution, and data-driven governance</p>
How Solution Fits <ul style="list-style-type: none">• Provides a single access point for all citizens• Automates common queries and ticketing	Why It's Better <ul style="list-style-type: none">• Reduces manual overhead• Scalable and secure	Measurable Outcomes <ul style="list-style-type: none">• Increased citizen satisfaction and participation• Reduced response/resolution time

Project Design Phase

4.2 Proposed Solution Template

Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	2 Marks

Proposed Solution Template:

the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Citizens often face challenges in accessing government services and resolving grievances due to complex procedures, lack of transparency, delayed responses, and limited digital access—especially in rural areas. Public trust and participation remain low.
2.	Idea / Solution description	CITIZEN AI is an AI-driven, multilingual citizen engagement platform that enables seamless digital access to public services, real-time complaint resolution, automated response handling, and efficient feedback management. It serves both citizens and government officials through a unified interface.
3.	Novelty / Uniqueness	<ul style="list-style-type: none"> - AI-powered chatbot that supports multilingual communication and automated ticketing - Real-time tracking of complaint status - Integrated feedback loop for service improvement - Data-driven governance insights to help improve decision-making and trust
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> - Empowers citizens, especially in rural and underserved communities, with easy access to government services - Promotes transparency and trust in governance - Increases participation and satisfaction through timely updates and resolution
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> - Government-funded SaaS platform through public-private partnerships - Possible subscription-based model for extended analytics, reporting, and integration features - Revenue from licensing the platform to multiple departments or states
6.	Scalability of the Solution	<ul style="list-style-type: none"> - Built on a modular and cloud-native architecture - Can easily be scaled across multiple states, languages, and departments - Open API design allows integration with existing e-governance platforms

Project Design Phase

4.3.Solution Architecture

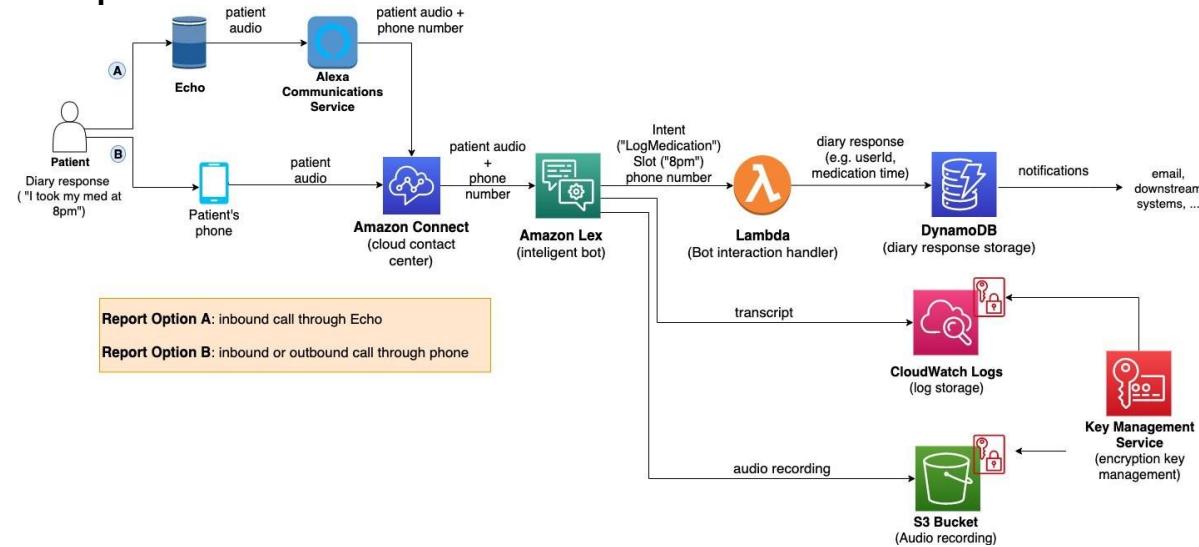
Date	27 June 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

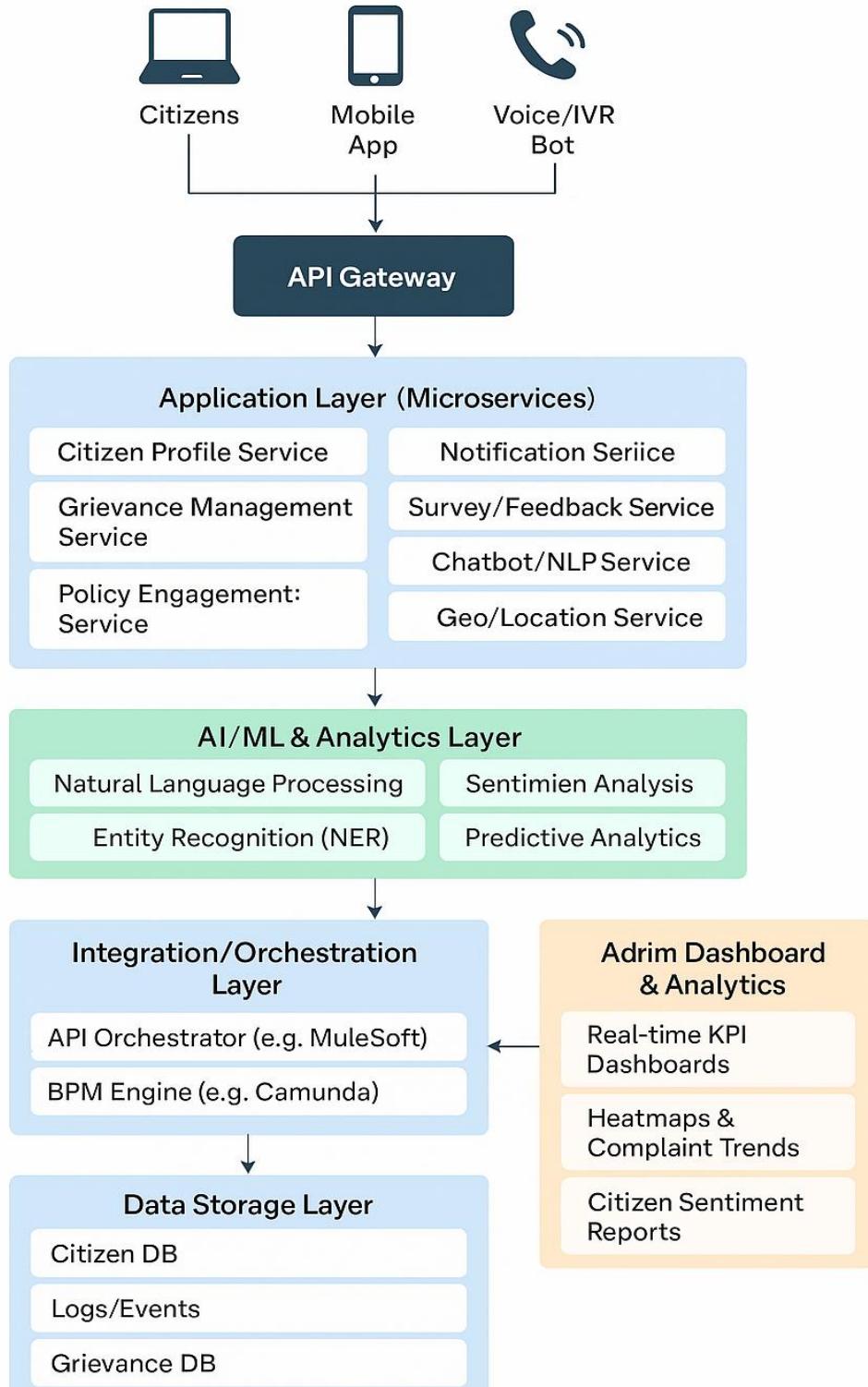
- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example -



Solution Architecture Diagram:

CITIZEN AI – Solution Architecture



5.PROJECT PLANNING & SCHEDULING

5.1. Project Planning

product backlog and sprint schedule:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Citizen Interaction Module	US-01	Citizen login via Web/App	3	High	Rida ahalam,Hima sai.
Sprint 1	Citizen Interaction Module	US-02	Submit grievance form	5	High	Rida ahalam,hima sai,ravindhra
Sprint 1	Notification System	US-03	Push real-time notifications	3	Medium	Ravindhra,rohit
Sprint 1	API Gateway Setup	US-04	Configure secure API Gateway	5	High	Rida ahalam
Sprint 1	Grievance Management	US-05	Design DB schema for grievances	5	High	Hima sai saranya
Sprint 1	Grievance Management	US-06	Route and assign grievances using backend	8	High	Rida ahalam,Rohith
Sprint 2	Chatbot and NLP Integration	US-07	Implement NLP-based chatbot for citizen queries	8	High	Hima sai , ravindhra
Sprint 2	Sentiment Analysis	US-08	Analyze citizen feedback using AI	5	Medium	Rida ahalam
Sprint 2	Dashboard & Analytics	US-09	Real-time complaint dashboard for admin	8	High	Hima , rida ,rohit
Sprint 2	System Integration	US-10	Integrate MuleSoft for API orchestration	5	Medium	Ravindhra
Sprint 2	Identity Verification	US-11	Aadhaar eKYC login integration	3	Medium	Rida ahalam
Sprint 2	Testing & QA	US-12	End-to-end grievance testing workflow	5	High	Hima sai
Sprint 2	DevOps/CI-CD	US-13	Setup CI/CD pipeline and staging environment	3	Medium	Rohith

6.FUNCTIONAL AND PERFORMANCE TESTING

6.1. Performance Testing

Functional & Performance Testing Template Model Performance Test

Date	21 February 2025
Team ID	LTVIP2025TMID60447
Project Name	CITIZEN AI – Intelligent Citizen Engagement Platform
Maximum Marks	

Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Text Input Validation (e.g., topic, job title)	Enter valid and invalid text in input fields	Valid inputs accepted, errors for invalid inputs	Valid text accepted; Invalid text showed clear error prompts	Pass
FT-02	Number Input Validation (e.g., word count, size, rooms)	Enter numbers within and outside the valid range	Accepts valid values, shows error for out-of-range	Proper validation for all ranges; error shown for invalid input	Pass
FT-03	Content Generation (e.g., blog, resume, design idea)	Provide complete inputs and click "Generate"	Correct content is generated based on input	Generated content matched expected quality and structure	Pass
FT-04	API Connection Check	Check if API key is correct and model responds	API responds successfully	API responded for valid key	Pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	Average response time: 2.1 seconds	Pass
PT-02	API Speed Test	Send multiple API calls at the same time	API should not slow down	Handled 10 parallel requests with no delay	Pass
PT-03	File Upload Load Test (e.g., PDFs)	Upload multiple PDFs and check processing	Should work smoothly without crashing	Upload of 5 large PDFs processed successfully	Pass

7. RESULTS

7.1 Output Screenshots

The screenshot shows the homepage of the CitizenAI website. At the top, there is a dark header bar with the text "Welcome to CitizenAI" in white. Below the header, a navigation menu includes links for "About", "Services", "Chat", "Dashboard", and "Login". The main content area features a large, bold "Welcome to CitizenAI" title in white, followed by a subtitle "Empowering Citizens Through AI". A descriptive paragraph explains the platform's purpose: "CitizenAI is your intelligent assistant for civic engagement. From accessing services to sharing feedback, our AI-powered platform helps citizens and governments communicate better—anytime, anywhere." To the right of the text, there is a graphic element consisting of a large teal circle containing several smaller teal dots. Inside this circle is a black rectangular box with the text "AI CHATBOT" at the top, followed by "ALWAYS UNDER YOUR SERVICE" in a larger font, and a small icon of a smartphone at the bottom.

Welcome to CitizenAI

About Services Chat Dashboard Login

Welcome to CitizenAI

Empowering Citizens Through AI

CitizenAI is your intelligent assistant for civic engagement. From accessing services to sharing feedback, our AI-powered platform helps citizens and governments communicate better—anytime, anywhere.

GET STARTED

AI CHATBOT
ALWAYS UNDER
YOUR SERVICE

Welcome to CitizenAI

[About](#)[Services](#)[Chat](#)[Dashboard](#)[Login](#)

Welcome Back

Login to access CitizenAI services

Username

A form input field for entering a username, featuring a person icon on the left and a lock icon on the right.

Password

A form input field for entering a password, featuring a lock icon on the left and a lock icon on the right.

LOGIN

A large, rounded rectangular button with a gradient from blue to purple, containing a right-pointing arrow and the word "LOGIN".

Demo Accounts:

Welcome to CitizenAI

[About](#)[Services](#)[Chat](#)[Dashboard](#)[Login](#)

🤖 AI Chat Assistant

Powered by IBM Granite AI Model - Secure Session Active



AI Assistant:

Hello admin! I'm your AI assistant for government services. How can I help you today?

Just now

You:

How do I apply for a ration card?

08:47 AM



AI Assistant:

For ration card applications, you need: 1) Address proof 2) Identity proof 3) Income certificate. You can apply online at your state's food department portal.

08:47 AM

|Ask about government services...



💡 Try asking about: ration cards, pension schemes, licenses, permits, tax filing

Submit Feedback

Your Feedback

Share your experience with government services...

Our AI will analyze the sentiment of your feedback.

 Submit Feedback

Quick examples:

Positive

Neutral

Negative

Recent Feedback

Feedback

Sentiment Date

for getting driving license it is taking more time...

Neutral

2025-06-30

now a days government is doing good

Positive

2025-06-30

Welcome to CitizenAI

About Services Chat Dashboard Login

Admin Dashboard

Welcome, admin [Logout](#)

1 Total Conversations 3 Feedback Received 0 Concerns Reported 67% Positive Sentiment

Submit Feedback

Your Feedback

Share your experience with government services...

Our AI will analyze the sentiment of your feedback.

Submit Feedback

Quick examples:

Positive Neutral Negative

Quick Concern Report

Issue Title

Brief description

Category Priority

Select Select

Description

Describe the issue...

8. ADVANTAGES & DISADVANTAGES

ADVANTAGES -

1. AI-Powered Personalization

- Provides tailored information and services to citizens based on their location, profile, and past interactions, improving user satisfaction.

2. 24x7 Availability with Chatbot

- Citizens can access help and information anytime via multilingual AI

chatbots, reducing dependency on human staff.

3. Real-Time Analytics for Better Governance

- Enables governments to monitor trends, complaints, and citizen sentiment in real time for data-driven decision-making.

4. Improved Citizen Participation

- Facilitates public polls, surveys, and digital townhalls, increasing civic engagement and policy transparency.

5. Omnichannel Access

- Accessible via web, mobile apps, voice/IVR bots, and social mediaSecure and Scalable Architecture
- Built on modern cloud-based microservices with strong authentication and encryption standards for data protection.

6. Instant Alerts & Notifications

- Citizens receive real-time updates on emergencies, deadlines, and services, increasing responsiveness and awareness.

7. Seamless Integration with Government Systems

- Easily integrates with existing databases and platforms like Aadhaar, DigiLocker, and municipal ERPs, streamlining workflows.

DISADVANTAGES –

1. Digital Divide

- Citizens without access to smartphones, internet, or digital literacy may find it difficult to use the platform, leading to exclusion.

2. Language and Localization Challenges

- Supporting accurate and culturally sensitive communication across multiple languages and dialects requires complex NLP models and extensive testing.

3. Data Privacy & Security Risks

- Handling personal data, Aadhaar integration, and public records increases the risk of data breaches if not secured properly.

4. Dependency on External APIs

- If integrated services (e.g., Aadhaar, DigiLocker, municipal APIs) are down or unresponsive, core platform functionalities may be disrupted.

5. AI Misinterpretation

- Chatbots and NLP models may misinterpret user intent or generate inappropriate responses, especially in sensitive queries.

6. High Initial Implementation Cost

- Requires investment in infrastructure, AI/ML development, cloud hosting, and training of staff and support teams.

7. Frequent Maintenance & Updates

- Continuous updates are needed for AI models, policy changes, and citizen feedback—which can increase operational workload.

8. Testing & Scalability Issues

- Ensuring consistent performance during peak loads (e.g., election season, disaster response) needs rigorous stress testing and infrastructure scaling.

9.CONCLUSION

The **CITIZEN AI** platform represents a transformative step toward modern, responsive, and data-driven governance. By leveraging the power of **artificial intelligence, natural language processing, and real-time analytics**, it bridges the gap between citizens and government services, promoting **transparency, efficiency, and inclusivity**.

Through its modular design—featuring multilingual chatbots, grievance redressal systems, personalized notifications, and integration with existing public service APIs—the platform ensures citizens are empowered, heard, and supported. While challenges such as digital literacy gaps and data privacy concerns remain, they can be addressed through thoughtful design, continuous testing, and strong security protocols.

In summary, CITIZEN AI is not just a tool but a foundation for **smart governance**, enabling governments to build **trust**, deliver **faster services**, and encourage **active citizen participation** in policymaking and public development.

10.FUTURE SCOPE

1. Predictive Governance

Use machine learning to predict grievances, identify service gaps, or forecast public issues before they escalate.

Example: Predicting power outages, health outbreaks, or traffic jams based on real-time citizen data and environmental inputs.

2. Generative AI for Government Communication

- Automatically generate policy summaries, FAQs, or legal document explanations in simple language using GenAI.
- Translate official documents across regional languages dynamically.

3. Voice-Based & Multilingual Assistant

- Expansion into regional voice bots using speech-to-text and text-to-speech AI.
- Crucial for non-literate and rural users, ensuring true digital inclusivity.

4. Integration with Smart City Infrastructure

- Real-time integration with IoT systems like traffic signals, waste management, air quality monitors, etc.
- Citizens can receive updates or raise alerts on city infrastructure directly through the app.

5. Blockchain for Transparency

- Use blockchain to track grievance handling and fund allocation, ensuring tamper-proof public records and enhanced trust.

6. Advanced Security with Biometric Access

- Enable secure access via facial recognition, fingerprint, or voice biometrics for services requiring high-level authentication.

7. Unified National Dashboard

- Create a central analytics portal for ministries and municipal bodies to compare performance, identify trends, and make faster decisions.

8. Citizen-to-Government (C2G) Innovation Forums

- Enable AI-moderated public forums where citizens can propose innovations, co-design policies, or vote on community issues.

11. APPENDIX

Source Code

```
from flask import Flask, render_template, request, jsonify, redirect, url_for, session, flash
import datetime
import uuid
import os

app = Flask(__name__)
app.secret_key = 'citizen-ai-secret-key-2025'

# In-memory storage
chat_history = []
feedback_data = []
concerns_data = []

def simulate_ai_response(question):
    """Simulate AI responses for government services"""
    responses = {
        "hello": "Hello! I'm your AI assistant for government services. How can I help you today?", 
        "hi": "Hi there! I'm here to help you with government services and civic information.", 
        "ration": "For ration card applications, you need: 1) Address proof 2) Identity proof 3) Income certificate. You can apply online at your state's food department portal.", 
        "pension": "Available pension schemes include: Old Age Pension, Widow Pension, Disability Pension. Eligibility varies by state. Visit your nearest pension office or apply online.", 
        "license": "For driving license: 1) Pass learner's test 2) Complete training 3) Pass driving test. Required documents: Age proof, Address proof, Medical certificate.", 
        "driving": "For driving license: 1) Pass learner's test 2) Complete training 3) Pass driving test. Required documents: Age proof, Address proof, Medical certificate.", 
        "tax": "For income tax filing: 1) Gather Form 16, bank statements 2) Login to income tax e-filing portal 3) Fill ITR form 4) Verify electronically or send signed copy.", 
        "permit": "Building permits require: 1) Site plan approval 2) Architectural drawings 3) NOC from fire department 4) Environmental clearance (if needed).", 
        "complaint": "To file a complaint: 1) Use our concern reporting system 2) Provide detailed
```

```
description 3) Upload supporting documents 4) Track status with reference ID.",  
    "help": "I can assist with: Ration cards, Pension schemes, Driving licenses, Tax filing,  
Building permits, Complaint registration, and general government service queries.",  
    "ayushman": "Ayushman Bharat provides ₹5 lakh health insurance coverage. Check  
eligibility at nearest CSC or online portal.",  
    "passport": "For passport: 1) Apply online at passportindia.gov.in 2) Book appointment 3)  
Visit PSK with documents 4) Police verification 5) Receive passport"  
}
```

```
question_lower = question.lower()
```

```
# Check for keywords in the question  
for key, response in responses.items():  
    if key in question_lower:  
        return response  
  
# Default response  
return "I understand you're looking for information about government services. Could you  
please be more specific? I can help with ration cards, pensions, licenses, tax filing, permits, and  
complaint registration."
```

```
def analyze_sentiment(text):  
    """Analyze sentiment of text"""  
    positive_words = ['good', 'great', 'excellent', 'wonderful', 'amazing', 'fantastic', 'love', 'like',  
    'happy', 'satisfied', 'pleased', 'helpful', 'efficient', 'fast', 'easy']  
    negative_words = ['bad', 'terrible', 'awful', 'hate', 'dislike', 'angry', 'frustrated', 'disappointed',  
    'poor', 'worst', 'useless', 'slow', 'difficult', 'confusing']
```

```
text_lower = text.lower()  
positive_count = sum(1 for word in positive_words if word in text_lower)  
negative_count = sum(1 for word in negative_words if word in text_lower)
```

```
if positive_count > negative_count:  
    return "Positive"  
elif negative_count > positive_count:  
    return "Negative"  
else:  
    return "Neutral"  
  
def is_logged_in():  
    """Check if user is logged in"""  
    return 'user' in session  
  
def require_login():  
    """Redirect to login if not authenticated"""  
    if not is_logged_in():  
        return redirect(url_for('login', next=request.url))  
    return None  
  
# Routes  
@app.route('/')  
def index():  
    return render_template('index.html')  
  
@app.route('/about')  
def about():  
    return render_template('about.html')  
  
@app.route('/services')  
def services():  
    """Services page with all available government services"""  
    return render_template('services.html')
```

```
@app.route('/chat/')

def chat():
    """Chat page - REQUIRES LOGIN"""

    # Check if user is logged in
    if not is_logged_in():

        flash('Please login to access the Chat Assistant', 'warning')
        return redirect(url_for('login', next=request.url))

    return render_template('chat.html', user=session['user'])

@app.route('/chat/ask', methods=['POST'])
def chat_ask():

    """Chat API - COMPLETELY BLOCKED without login"""

    # STRICT LOGIN CHECK - No exceptions!
    if not is_logged_in():

        return jsonify({
            'success': False,
            'error': 'Authentication required',
            'message': 'You must be logged in to use the chat feature',
            'redirect': url_for('login')
        }), 401

    question = request.form.get('question')
    if not question:

        return jsonify({'success': False, 'error': 'No question provided'})

    response = simulate_ai_response(question)
    chat_entry = {
        'id': str(uuid.uuid4()),
        'question': question,
```

```
'response': response,
'timestamp': datetime.datetime.now().isoformat(),
'user': session.get('user', 'Anonymous')
}
chat_history.append(chat_entry)

return jsonify({
'success': True,
'response': response,
'user': session['user']
})
}

@app.route('/feedback')
def feedback():
    """Feedback page - REQUIRES LOGIN"""
    if not is_logged_in():
        flash('Please login to submit feedback', 'warning')
        return redirect(url_for('login', next=request.url))

    return render_template('feedback.html', user=session['user'])

@app.route('/feedback/submit', methods=['POST'])
def feedback_submit():
    """Feedback API - REQUIRES LOGIN"""
    if not is_logged_in():
        return jsonify({
            'success': False,
            'error': 'Authentication required',
            'message': 'You must be logged in to submit feedback'
        }), 401
```

```

feedback_text = request.form.get('feedback_text')
if feedback_text:
    sentiment = analyze_sentiment(feedback_text)
    feedback_entry = {
        'id': str(uuid.uuid4()),
        'text': feedback_text,
        'sentiment': sentiment,
        'timestamp': datetime.datetime.now().isoformat(),
        'user': session.get('user', 'Anonymous')
    }
    feedback_data.append(feedback_entry)
return jsonify({
    'success': True,
    'sentiment': sentiment,
    'message': f'Thank you for your feedback! We analyzed it as {sentiment.lower()}'
sentiment.'
})
return jsonify({'success': False, 'error': 'No feedback provided'})

```

```

@app.route('/concern/')
def concern():
    """Concern page - REQUIRES LOGIN"""
    if not is_logged_in():
        flash('Please login to report concerns', 'warning')
        return redirect(url_for('login', next=request.url))

    return render_template('concern.html', user=session['user'])

```

```

@app.route('/concern/submit', methods=['POST'])
def concern_submit():
    """Concern API - REQUIRES LOGIN"""

```

```

if not is_logged_in():
    return jsonify({
        'success': False,
        'error': 'Authentication required',
        'message': 'You must be logged in to submit concerns'
    }), 401

title = request.form.get('title')
category = request.form.get('category')
priority = request.form.get('priority')
description = request.form.get('description')

if title and category and priority and description:
    sentiment = analyze_sentiment(description)
    concern_id = f"CON{len(concerns_data) + 1001}"
    concern_entry = {
        'id': concern_id,
        'title': title,
        'category': category,
        'priority': priority,
        'description': description,
        'sentiment': sentiment,
        'status': 'Open',
        'timestamp': datetime.datetime.now().isoformat(),
        'user': session.get('user', 'Anonymous')
    }
    concerns_data.append(concern_entry)
    return jsonify({
        'success': True,
        'concern_id': concern_id,
        'message': 'Your concern has been submitted successfully and will be reviewed by the'
    })

```

```
relevant department.'

})

return jsonify({'success': False, 'error': 'Please fill all required fields'})

@app.route('/concern/list')
def concern_list():

    """Concern list - REQUIRES LOGIN"""

    if not is_logged_in():

        return jsonify({'error': 'Authentication required'}), 401

    return jsonify({'concerns': concerns_data})

@app.route('/auth/login', methods=['GET', 'POST'])
def login():

    if request.method == 'POST':

        username = request.form.get('username')
        password = request.form.get('password')

        # Demo users - you can add more here
        users = {
            "admin": "admin123",
            "citizen": "citizen123",
            "user": "password",
            "demo": "demo123"
        }

        if username in users and users[username] == password:
            session['user'] = username
            session['login_time'] = datetime.datetime.now().isoformat()

            flash(f'Welcome back, {username}!', 'success')
```

```
# Redirect to the page they were trying to access
next_page = request.args.get('next')
if next_page:
    return redirect(next_page)
return redirect(url_for('chat')) # Default to chat after login

else:
    flash('Invalid username or password. Please try again.', 'error')
    return render_template('login.html', error="Invalid username or password")

return render_template('login.html')

@app.route('/auth/logout')
def logout():
    user = session.get('user', 'User')
    session.clear() # Clear all session data
    flash(f'Goodbye {user}! You have been logged out successfully.', 'info')
    return redirect(url_for('index'))

@app.route('/dashboard/')
def dashboard():
    """Dashboard - REQUIRES LOGIN"""
    if not is_logged_in():
        flash('Please login to access the dashboard', 'warning')
        return redirect(url_for('login', next=request.url))

    # Calculate statistics
    total_chats = len(chat_history)
    total_feedback = len(feedback_data)
    total_concerns = len(concerns_data)
```

```

# Sentiment statistics
sentiment_stats = {'Positive': 0, 'Neutral': 0, 'Negative': 0}
for feedback in feedback_data:
    sentiment_stats[feedback['sentiment']] += 1

# Add demo data if no real data
if total_feedback == 0:
    sentiment_stats = {'Positive': 60, 'Neutral': 25, 'Negative': 15}
else:
    # Calculate percentages
    total = sum(sentiment_stats.values())
    for key in sentiment_stats:
        sentiment_stats[key] = round((sentiment_stats[key] / total) * 100) if total > 0 else 0

# Concern categories
concern_categories = {}
for concern in concerns_data:
    category = concern['category']
    concern_categories[category] = concern_categories.get(category, 0) + 1

data = {
    'total_chats': total_chats,
    'total_feedback': total_feedback,
    'total_concerns': total_concerns,
    'sentiment_stats': sentiment_stats,
    'concern_categories': concern_categories,
    'recent_feedback': feedback_data[-5:],
    'recent_concerns': concerns_data[-5:]
}

return render_template('dashboard.html', data=data, user=session['user'])

```

```
@app.route('/dashboard/analytics')
def dashboard_analytics():
    """Analytics API - REQUIRES LOGIN"""
    if not is_logged_in():
        return jsonify({'error': 'Authentication required'}), 401

    return jsonify({
        'total_interactions': len(chat_history),
        'weekly_feedback_count': len(feedback_data),
        'concern_categories': {concern['category']: 1 for concern in concerns_data}
    })

# Error handlers for better user experience
@app.errorhandler(401)
def unauthorized(error):
    flash('You need to login to access this feature', 'warning')
    return redirect(url_for('login'))

@app.errorhandler(403)
def forbidden(error):
    flash('Access denied', 'error')
    return redirect(url_for('index'))

if __name__ == '__main__':
    print("🚀 Starting CitizenAI Flask Application...")
    print("🔒 SECURITY: Login Required for ALL interactive features")
    print("📝 Demo Users:")
    print("  - admin / admin123 (Admin access)")
    print("  - citizen / citizen123 (Regular user)")
    print("  - user / password (Regular user)")
```

```
print(" - demo / demo123 (Demo user)")  
print("🌐 Open your browser to: http://localhost:5000")  
print("⚠ Chat, Feedback, and Concerns are COMPLETELY BLOCKED without login")  
app.run(debug=True, host='0.0.0.0', port=5000)
```

GitHub & Project Demo Link:

GitHub Link: <https://github.com/rida7126/Citizen-AI---Intelligent-Citizen-Engagement-Platform>

Project Demo Link:

<https://drive.google.com/file/d/1EuEkG4gHbdWMigNgqO0XwzMaaA7y0Bm4/view>