

CITIZEN AI – INTELLIGENT CITIZEN ENGAGEMENT PLATFORM

Team ID : NM2025TMID01897

Project title: **CITIZEN AI**

Team member :

Name1:S.ISHWARYA(TL) NM ID:8C50752908430CF963E4113900981F5C

Name2: Z FATHIMA FARHANA. NM ID: D5641113C28B345AF4E4593CD00DB9AC

Name 3: J KARTHIKA. NM ID: 916C1FCB38CE8969E464E6BB59C736D2

Name 4: M RADHIKA. NM ID: CEDCD43A201F9A3AD3B4F4883AA24C9A

Name5: S RASMA .NM ID: DF31331F3B385719ADD497DF88099817

1. OVERVIEW OF CITIZEN AI

1.1 Overview

Citizen AI refers to an empowered approach where non-technical users, known as citizen developers, utilize AI capabilities within a platform to build, configure, and deploy AI-powered solutions without deep coding expertise. It democratizes AI access, enabling faster innovation, enhanced productivity, and more responsive services, especially in contexts such as public administration, project management, and citizen engagement. Instead of reliance on centralized IT, Citizen AI integrates AI tools into user-friendly, low-code platforms that allow users to address specific operational and strategic needs autonomously.

1.2 Mission and Goals

The mission of Citizen AI is to enable all stakeholders, including specialists from various domains and citizen developers, to leverage AI easily and securely, fostering an inclusive AI-driven environment. Key goals include accelerating digital transformation, reducing dependency on technical teams, ensuring data and security governance, improving decision-making through AI insights, and increasing efficiency by automating routine tasks.

1.3 Target Audience

Citizen AI platforms target a diverse user base:

Administrators: Manage AI infrastructure, security, and vendor configurations.

Specialists: Domain experts who design AI-driven workflows and use AI-assisted tools

Citizen Developers: Non-technical users who build AI-powered applications using low-code/no-code interfaces.

End Citizens: The ultimate beneficiaries who use AI-enabled services for engagement, feedback, and community interaction.

2. INTRODUCTION OF CITIZEN AI

2.1 Background and Rationale

As technological complexity grows, organizations face challenges in scaling AI adoption across departments. Traditional AI development requires specialized skills and long deployment cycles, alienating many knowledge workers. Citizen AI emerged as a response, aiming to empower non-technical users to leverage AI without waiting for IT bottlenecks. It bridges the gap between AI potential and practical application, making AI a strategic resource embedded within everyday workflows.

2.2 AI in Citizen Engagement

AI facilitates more dynamic, responsive citizen engagement by automating communication, analysing sentiment, and personalizing interactions. Chatbots handle inquiries 24/7, sentiment analysis gauge's public opinion, and dashboards provide real-time insights into community needs. This use of AI helps governments and organizations foster transparency, inclusiveness, and responsiveness in citizen services.

2.3 Problem Statement

The major challenges addressed by Citizen AI include the complexity of AI integration, lack of AI expertise in many departments, slow innovation cycles, uncontrolled use of external AI tools ("Shadow AI"), and security concerns regarding sensitive citizen data. Citizen AI seeks to overcome these through secure, governance-compliant, user-friendly AI platforms that empower internal stakeholders.

3. ARCHITECTURE OVERVIEW OF CITIZEN AI

3.1 Platform Structure

Citizen AI platforms typically feature a modular architecture with:

Frontend: User interface built with modern web technologies for accessibility and ease of use.

Backend: AI services, data processing, workflow engines, and integration layers managing the AI lifecycle.

Low-code/No-code Modules: Tools enabling citizen developers to assemble AI functionalities without writing code.

Security & Governance Layer: Authentication, authorization, and audit trails.

Backend & Frontend Technologies

Backend often uses scalable cloud infrastructure with AI/ML frameworks (e.g., TensorFlow, PyTorch), AI service APIs, and containerized microservices. Frontend utilizes frameworks like React or Angular, emphasizing responsive design and user experience.

3.2 Modular Components

AI Chatbots and virtual assistant

Sentiment and data analytics engines

Workflow automation module

Integration adapters for external systems and APIs

User role and permission management

4. CORE FEATURES

4.1 AI Chatbot

An intelligent conversational interface that handles inquiries, provides guidance, and automates routine responses. It can be customized with natural language understanding (NLU) and integrate with backend services to provide contextual and personalized answers.

4.2 Sentiment Analysis

Analyses text data from feedback forms, social media, and surveys to extract public mood and opinions. This helps in real-time monitoring and proactive decision-making.

4.3 Dashboard

A visual interface presenting key metrics, analytics, and system status. Users can monitor AI interactions, workflow progress, and sentiment trends.

4.4 Automation Tools

Tools to design and execute business workflows, task automation, and AI-driven triggers, enabling faster and error-free operations.

5. USER ROLES AND PERMISSIONS OF CITIZEN AI

Roles

In Citizen AI, roles and permissions are used to securely manage and control access based on the assigned user roles within the platform. The system respects the roles and permissions of cplace users, ensuring that access rights are granted according to predefined roles, which define what actions and features a user can use in the Citizen AI environment. This role-based access control (RBAC) model helps maintain security, control, and compliance by restricting actions users can perform based on their assigned roles.

5.1 Admin: Oversees system configuration, AI model selection, security, and compliance.

5.2 Specialist: Domain experts who utilize AI capabilities for analysis and content creation.

5.3 Citizen Developer: Builds AI-powered applications and workflows using no-code tools.

5.4 How Roles and Permissions Work in Citizen AI

Roles represent job functions or user types within the organization.

Permissions specify the exact actions a user can perform in the Citizen AI platform (e.g., managing AI agents, editing knowledge bases, or deploying workflows).

Permissions are assigned to roles, and users are assigned roles, thus inheriting the corresponding permissions.

This approach enables fine-grained control to tailor user access, support collaboration, and protect critical AI governance functions.

5.5 Benefits

Secure and scalable AI collaboration across teams.

Clear governance by restricting access only to authorized users.

Customized roles and permission sets that operate across AI agents and platform features.

6. AUTHENTICATION IN CITIZEN AI

Authentication in Citizen AI ensures that only authorized users, such as citizens and government officials, can securely access the platform. The system uses JWT (JSON Web Token) based authentication combined with role-based access control (RBAC). This mechanism guarantees secure communication, identity verification, and differentiated access to services

Key Features

6.1 User Registration

Citizens and officials register through /auth/register.

Required details: name, email/phone, government ID (e.g., Aadhaar or official ID).

Passwords are securely hashed before storage.

6.2 Login & Token Generation

Users log in via /auth/login with credentials.

On successful authentication, a JWT token is issued.

The token contains user role (citizen or official), ID, and session validity.

6.3 Secure Access

Each request to protected APIs (complaints, feedback, services) must include the token in the Authorization header:

Authorization: Bearer <JWT_TOKEN>

Tokens have expiration times for added security.

6.4 Role-Based Permissions

Citizens → Access chatbot, services, complaints, and feedback submission.

Officials/Admins → Access dashboards, analytics, and manage complaints.

6.5 Session Management

Expired tokens require re-login.

Refresh tokens may be used for longer sessions in production deployments.

Example Authentication Flow

1. User registers with Citizen AI.
2. User logs in → system validates credentials → generates JWT.
3. JWT is included in every API request.
4. Server validates JWT → grants or denies access based on role and permissions.

7. USER INTERFACE IN CITIZEN AI

The Citizen AI User Interface (UI) is designed to be simple, accessible, and inclusive so that both citizens and government officials can interact with the system easily. The platform is available as a web application, mobile application, and AI-powered chatbot, ensuring maximum reach and usability.

Key UI Features

7.1 Home Dashboard

Provides quick access to government services, announcements, and ongoing schemes.

Citizens can see personalized recommendations based on their profile and past interactions.

7.2 AI Chatbot Interface

A text and voice-enabled chatbot for answering citizen queries in multiple languages.

Supports conversational interaction for services, complaints, and FAQs.

7.3 Citizen Services Portal

Section for browsing, searching, and applying for government schemes or services.

Includes status tracking for applications and requests.

7.4 Complaint & Feedback Section

Easy-to-use form for submitting complaints and grievances.

Citizens can view the progress and resolution status of their issues.

7.5 Government Official Dashboard

Provides real-time analytics, complaint statistics, and citizen feedback trends.

AI-powered visualizations (graphs, charts) to help in decision-making.

7.6 Multilingual Support

Interfaces available in English and regional languages to ensure inclusivity.

7.7 Mobile-Friendly Design

Responsive UI for smartphones and tablets.

Lightweight design optimized for low-bandwidth areas.

8. TESTING IN CITIZEN AI

Testing in Citizen AI is carried out systematically to ensure the application is secure, reliable, and user-friendly for both citizens and government officials. Since the platform integrates AI modules, databases, and multi-role access, different testing strategies are applied at each stage.

8.1 Unit Testing

Tests individual components like authentication, chatbot responses, and complaint submission.

Example: Verify that /auth/login returns a valid JWT token for correct credentials.

8.2 Integration Testing

Ensures that modules work correctly when combined.

Example: Submitting a complaint through the UI → stored in the database → accessible by officials on the dashboard.

8.3 Functional Testing

Checks that all features meet requirements.

Example: Citizens can submit feedback, officials can update complaint status, chatbot gives correct responses.

8.4 AI Model Testing

Validates accuracy of NLP models, sentiment analysis, and recommendation engine.

Example: Check if chatbot understands queries in multiple languages.

8.5 Security Testing

Tests authentication, authorization, and data encryption.

Example: Ensure only officials can access analytics APIs.

8.6 Usability Testing

Conducted with real users to ensure the interface is simple and accessible

Example: Testing chatbot interaction on low-bandwidth mobile networks.

8.7 Performance & Load Testing

Simulates multiple users accessing the system simultaneously.

Example: 1,000 citizens logging complaints without downtime.

9. SCREENSHOTS

```
1 import gradio as gr
2 import torch
3 from transformers import AutoTokenizer, AutoModelForCausalLM
4
5 # Load model and tokenizer
6 model_name = "ibm-granite/granite-3.2-2b-instruct"
7 tokenizer = AutoTokenizer.from_pretrained(model_name)
8 model = AutoModelForCausalLM.from_pretrained(
9     model_name,
10     torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
11     device_map="auto" if torch.cuda.is_available() else None
12 )
13
14 if tokenizer.pad_token is None:
15     tokenizer.pad_token = tokenizer.eos_token
16
17 def generate_response(prompt, max_length=1024):
18     inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)
19
20     if torch.cuda.is_available():
21         inputs = {k: v.to(model.device) for k, v in inputs.items()}
22
23     with torch.no_grad():
24         outputs = model.generate(
25             **inputs,
26             max_length=max_length,
27             temperature=0.7,
28             do_sample=True,
29             pad_token_id=tokenizer.eos_token_id
30         )
```

Figure 9.1

```
31
32 response = tokenizer.decode(outputs[0], skip_special_tokens=True)
33 response = response.replace(prompt, "").strip()
34 return response
35
36 def city_analysis(city_name):
37     prompt = f"Provide a detailed analysis of {city_name} including:\n1. Crime Index and safety statistics\n2. Accident rates and traffic safety information\n3. Overall safety assessment\n\nCity: {city_name}"
38     return generate_response(prompt, max_length=1000)
39
40 def citizen_interaction(query):
41     prompt = f"As a government assistant, provide accurate and helpful information about the following citizen query related to public services, government policies, or civic issues:\n\nQuery: {query}"
42     return generate_response(prompt, max_length=1000)
43
44 # Gradio interface
45 gr.Blocks() as app:
46     r.Markdown("# City Analysis & Citizen Services AI")
47
48     with gr.Tabs():
49         with gr.TabItem("City Analysis"):
50             with gr.Row():
51                 with gr.Column():
52                     city_input = gr.Textbox(
53                         label="Enter City Name",
54                         placeholder="e.g., New York, London, Mumbai...",
55                         lines=1
56                     )
57                     analyze_btn = gr.Button("Analyze City")
58
59                 with gr.Column():
60                     city_output = gr.Textbox(label="City Analysis (Crime Index & Accidents)", lines=15)
61
62             analyze_btn.click(city_analysis, inputs=city_input, outputs=city_output)
```

Figure 9.2

```

63
64     with gr.TabItem("Citizen Services"):
65         with gr.Row():
66             with gr.Column():
67                 citizen_query = gr.Textbox(
68                     label="Your Query",
69                     placeholder="Ask about public services, government policies, civic issues...",
70                     lines=4
71                 )
72                 query_btn = gr.Button("Get Information")
73
74             with gr.Column():
75                 citizen_output = gr.Textbox(label="Government Response", lines=15)
76
77         query_btn.click(citizen_interaction, inputs=citizen_query, outputs=citizen_output)
78
79 aunch(share=True)

```

Figure 9.3

City Analysis & Citizen Services AI

City Analysis

Citizen Services

Enter City Name

Hyderabad

Analyze City

City Analysis (Crime Index & Accidents)

****Positive Aspects**:**

- ****Cosmopolitan Environment**:** The city's cultural diversity and openness contribute to a generally cooperative atmosphere.
- ****Strong Law Enforcement**:** Hyderabad boasts an active police force that is visible in public spaces, contributing to a deterrent effect against crime.
- ****Safety Infrastructure**:** Efforts to improve infrastructure, like better road systems and enhanced public transportation, can help reduce crime and accidents.

****Challenges**:**

- ****Crime Influx**:** The city's rapid growth and increasing wealth disparities sometimes lead to concentrated pockets of higher crime rates.
- ****Traffic Mayhem**:** Congestion and lack of proper road infrastructure exacerbate traffic-related dangers.
- ****Cybersecurity Concerns**:** As Hyderabad is a digital hub, the growing threat of cybercrimes is a pressing concern.

In conclusion, Hyderabad presents a complex safety picture. While significant strides have been made in enhancing road safety through infrastructure improvements and public awareness, persistent challenges remain in dealing with property and violent crimes, especially in

Figure 9.4

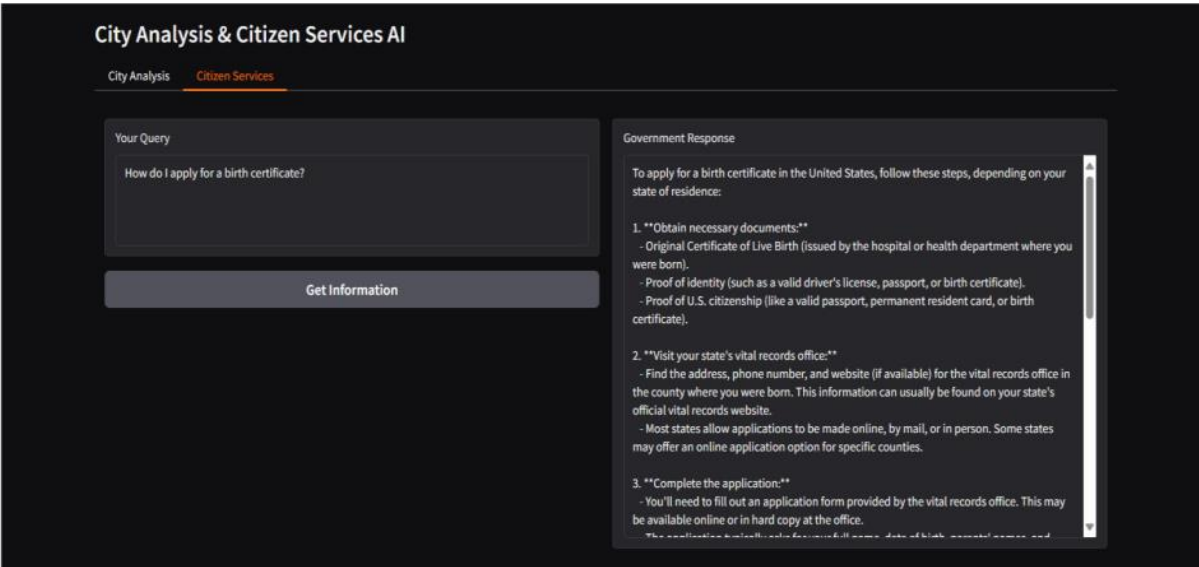


Figure 9.5

10. CONCLUSIONS IN CITIZEN AI

Citizen AI represents a significant step toward smart governance by combining artificial intelligence with citizen-centric services. The platform simplifies access to government schemes, improves grievance redressal, and enhances decision-making through AI-powered analytics. With features such as multilingual support, chatbot assistance, real-time dashboards, and secure authentication, Citizen AI empowers citizens to actively participate in governance while enabling officials to respond more effectively to public needs.

Although challenges such as AI accuracy, data quality, and scalability still exist, the project provides a strong foundation for continuous improvement. Future enhancements in areas like advanced NLP, cloud scalability, offline support, and stronger data privacy measures will further strengthen the system.

Ultimately, Citizen AI is more than just a technological solution—it is a bridge between citizens and governments, fostering transparency, inclusivity, and accountability. By leveraging AI for governance, the project promotes active citizen engagement, better policy formulation, and improved trust in public institutions, paving the way for a more participatory and sustainable digital future.

11. KNOWN ISSUES IN CITIZEN AI

While Citizen AI provides a robust framework for enhancing governance and citizen interaction, certain limitations and challenges still exist:

11.1 AI Accuracy Limitations

The chatbot and NLP models may misinterpret queries, especially in regional dialects or mixed-language inputs.

Sentiment analysis sometimes struggles with sarcasm or context-heavy feedback.

11.2 Data Quality Issue

If government databases are outdated or inconsistent, the system may provide incomplete or incorrect information

Lack of standardized formats for policies and schemes can affect data integration.

High user traffic during peak times (e.g., election periods, disaster management) may slow down response times.

AI model inference can become resource-intensive, impacting performance.

11.4 Security & Privacy Risks

Handling sensitive citizen data (IDs, complaints, feedback) requires strict compliance; any lapse can lead to privacy breaches.

Risk of unauthorized access if authentication tokens are not properly managed.

11.5 Limited Offline Access

The system heavily depends on internet connectivity, making it less effective in rural or low-bandwidth regions.

11.6 User Adoption Challenges

Citizens unfamiliar with digital platforms may find it difficult to use AI-driven services.

Trust in AI-based recommendations may take time to build.

12. FUTURE ENHANCEMENTS IN CITIZEN AI

To overcome existing limitations and improve overall efficiency, several enhancements can be implemented in future versions of Citizen AI:

12.1 Improved AI Models

Train NLP models with larger datasets in multiple regional dialects to enhance chatbot accuracy.

Use context-aware AI (like transformer-based models) to better understand complex citizen queries.

12.2 Data Standardization & Integration

Develop APIs to integrate directly with government databases in real time.

Implement data validation pipelines to ensure accuracy and consistency of policy and scheme information.

12.3 Scalability & Cloud Optimization

Adopt microservices and container orchestration (Kubernetes) for handling high traffic loads.

Use edge computing to improve response times in rural and remote areas.

12.4 Enhanced Security & Privacy

Introduce multi-factor authentication (MFA) for both citizens and officials.

Use blockchain or distributed ledger technologies to secure sensitive citizen data and audit logs.

12.5 Offline & Low-Bandwidth Support

Provide an SMS/IVR-based version of Citizen AI for areas with poor internet access.

Enable lightweight mobile applications that work in offline mode and sync data when connectivity is restored.

12.6 User Awareness & Training

Conduct digital literacy campaigns to increase citizen adoption.

Offer multilingual tutorials and guides within the application.

12.7 Advanced Analytics for Governance

Use AI to predict emerging social issues (e.g., water scarcity, health risks) based on citizen feedback.

Provide recommendation dashboards for policymakers with scenario-based simulations.