CITIZEN AI –

INTELLIGENT CITIZEN ENGAGEMENT PLATFORM

Project Documentation

1. Introduction

Citizen AI is an innovative platform that leverages artificial intelligence to enhance citizen engagement and government service delivery. It provides personalized services, intelligent query resolution, and easy access to information, promoting transparency and efficiency. By streamlining interactions, Citizen AI improves the overall citizen experience and fosters a more responsive government.

Citizen AI is a powerful platform that can revolutionize citizen engagement and government service delivery. By providing a personalized, intelligent, and efficient way for citizens to interact with government services, Citizen AI can improve the overall citizen experience and promote transparency and accountability.

Citizen AI is an intelligent platform that revolutionizes government-citizen interactions through AI-driven responses, sentiment analysis and real-time analytics. It provides personalized services, efficient query resolution and easy access to information, promoting transparency and trust. This platform enhances citizen engagement, fostering a more responsive government.

Project Title: Citizen AI - Intelligent Citizen Engagement Platform.

Team Member: AARTHI S

Team Member: ABARNA V

Team Member: VARSHA DEVI S

Team Member: VETHAMATHI G

Team Member: THARANIKA P

2. Project Overview

Purpose:

Citizen AI is designed to empower individuals and communities by promoting responsible, transparent, and inclusive use of artificial intelligence. It supports citizens in understanding AI-driven decisions, encourages ethical engagement, and provides accessible AI-driven services that improve governance, education, and social well-being.

Citizen AI is to empower citizens and governments with an intelligent engagement platform that fosters transparent communication, participatory governance, and community collaboration. By leveraging AI and real-time data, the platform helps governments understand citizen needs, gather feedback, and provide personalized services, while enabling citizens to actively participate in decision-making.

Citizen AI bridges the gap between citizens and officials by offering policy summaries, feedback loops, service updates, and predictive insights for improved governance. Ultimately, this assistant enhances civic trust, inclusivity, and responsiveness in modern smart societies.

Features:

Conversational Interface:

Natural language chat for citizens and officials to ask questions, get updates, and access services.

Policy Summarization:

Transforms lengthy government documents into simplified, actionable insights.

Citizen Feedback Loop:

Collects, analyzes, and reports citizen feedback to inform public policies and service delivery.

Community Engagement Insights:

Uses AI to detect citizen sentiment trends and provide recommendations for better governance.

Service Request Tracking:

Allows citizens to submit issues (waste, roads, water, etc.) and track their v. resolution status.

Predictive Governance (KPI Forecasting):

Forecasts governance performance indicators and public demand trends for strategic planning

Anomaly Detection:

Identifies irregularities in public services or citizen reports (e.g., unusual complaints, sudden spikes).

Multimodal Input Support:

Accepts text, PDFs, and CSVs for analyzing public documents, petitions, or surveys.

User-Friendly Dashboard (Streamlit/Gradio):

Provides an intuitive platform for both citizens and officials with chat, dashboards, and feedback tools.

3. Architecture

Frontend (Streamlit):

Interactive citizen portal with dashboards, chat interface, service forms, and feedback pages.

Backend (FastAPI):

REST APIs for processing feedback, generating policy summaries, and managing service requests.

LLM Integration (IBM Watsonx Granite):

For natural language queries, report generation, and policy simplification.

Vector Search (Pinecone):

Stores and retrieves public policy and citizen documents for semantic search.

ML Modules:

Forecasting citizen service demand and detecting anomalies in public sentiment.

4. Setup Instructions

Prerequisites:

- Python 3.9 or later
- pip and virtual environment tools
- API keys for IBM Watsonx and Pinecone
- Internet access to access cloud services

Installation Process:

- Clone the repository
- Install dependencies from requirements.txt
- Create a .env file and configure credentials
- Run the backend server using Fast API
- Launch the frontend via Stream lit

- Upload data and interact with the modules

5. Folder Structure

- ✓ app/ → All backend logic (FastAPI routes, models, utils).
- ✓ ui/ → All citizen-facing frontend components (Streamlit pages).
- ✓ tests/ → Unit/API tests.
- ✓ data/ → Example policies, citizen feedback, etc.
- ✓ docs/ → Documentation and guides

6. Running the Application

- ➢ Start FastAPI backend
- ➢ Launch Streamlit citizen dashboard
- ➢ Citizens upload petitions, submit feedback, or chat with the assistant
- ➢ Officials view reports, trends, and service prediction

7. API Documentation

- POST /chat/ask – Query policy, service, or governance issues
- POST /upload-doc – Upload petitions or documents for summarization
- GET /search-docs – Semantic search across public records
- POST /submit-feedback – Citizens submit structured or open feedback
- GET /citizen-insights – Returns aggregated sentiment a

8. Authentication

- Token-based authentication for citizens
- Role-based access (citizen, admin, policymaker)
- Optional OAuth2 integration

9. User Interface

- Citizen-friendly portal with chat, service forms, dashboards
- Feedback visualization and real-time updates
- Downloadable policy reports and service updates

10. Testing

- Unit tests for citizen queries and policy summaries
- API tests for service request handling
- Usability testing with citizen interaction scenarios

11.Screenshot:

Provide screenshot and Demo Link showcasing the application features and design. Here, The Project Documentation, demo link and screenshot have been added in the GitHub repository for better understanding of the systems.

12. Objectives

- Promote ethical and transparent Al adoption in governance and daily life.
- Improve citizen literacy in Al through interactive education.

- ❖ Build a trustworthy Al assistant for citizens, policymakers, and communities.
- ❖ Ensure fairness, inclusivity, and accountability in Al systems.
- ❖ Provide real-time insights into policies, bias detection, and social impact.

13. Scope

Citizens: Personalized guidance, Al literacy modules, easy-to-read policy summaries.

Policymakers: Decision support with forecasting, fairness analysis, and community feedback.

Researchers & NGOs: Data access for bias analysis, ethical Al benchmarking.

Global Applicability: Can be deployed in smart cities, education systems, and digital governance platforms.

14. Problem Statement

- With rapid Al adoption, citizens face challenges like:
- Lack of Al literacy and awareness.
- Difficulty in understanding complex policies and Al decisions.
- Risk of bias, discrimination, and misuse of Al.
- Limited channels for citizen participation in Al governance.
- Citizen Al addresses these problems by providing a transparent, inclusive, and accessible Al ecosystem.

15. Use Case Scenarios

Policy Explanation: A citizen uploads a complex Al-related policy 'n gets a simple summary.

Fairness Checker: A government uploads a hiring dataset 'n tool detects potential gender bias.

Al Literacy Hub: A student uses the chatbot to learn about responsible Al.

Community Feedback: Citizens submit feedback on a new Al-based public service 'n policymakers see summarized insights.

Impact Forecasting: Predicting how automation will affect jobs in a local community.

16. Technology Stack

Frontend: Streamlit / Gradio (interactive dashboards).

Backend: FastAPI (REST APIs, async performance).

LLM Integration: IBM Watsonx Granite for summarization & natural language interaction.

Database: Pinecone (vector DB for semantic search), PostgreSQL (user/feedback data).

ML Models: Scikit-learn, pandas, matplotlib (forecasting, anomaly detection).

Authentication: OAuth2, JWT tokens.

Deployment: Docker, Kubernetes, IBM Cloud.

17. Performance Metrics

- Accuracy of Summarization (e.g., BLEU/ROUGE scores for policy summaries).
- Bias Detection Precision/Recall (ability to identify unfair datasets).
- System Response Time (chatbot latency in ms).
- User Engagement Metrics (feedback responses, number of queries).
- Scalability (how many simultaneous citizens can use it).

18. Limitations

- Dependent on quality of training data 'n risk of Al misinterpretation.

- Cloud/API dependency may limit offline use.
- Bias detection is probabilistic, not absolute.
- High compute cost for large-scale real-time forecasting.
- User trust must be continuously built through transparency.

19. Known Issues

- ✓ Limited support for regional languages/local dialects.
- ✓ Large files slow down summarization.
- ✓ Occasional inaccuracies in bias/anomaly detection.
- ✓ Requires stable internet for API-based services.

20. Future Enhancements

- ❖ Multilingual support for all major world languages.
- ❖ Integration with voice-based interfaces for accessibility.
- ❖ Al explainability module (why the Al made a decision).
- ❖ Expansion to IOT devices for smart citizen services.
- ❖ Gamified Al literacy programs for students.

21. References

- IBM Watsonx Documentation
- EU Guidelines on Trustworthy Al (2020)
- UNESCO Report on Al Ethics (2021)

- IEEE Ethically Aligned Design Guidelines (2022)
- Open Al Responsible Al Practices

22. Conclusion

Citizen Al is a transformative initiative aimed at bridging the gap between artificial intelligence and society. By making Al transparent, inclusive, and ethical, it empowers citizens to trust, use, and shape Al technologies responsibly. Through its features-policy summarization, bias detection, forecasting, and citizen engagement-it creates a balanced ecosystem where Al serves the people, not the other way around.