

Citizen AI – Intelligent Citizen Engagement Platform

1. Introduction

The Citizen AI project is developed under the Naan Mudhalvan (NM) program. Its main goal is to provide an AI-powered chatbot that helps governments and organizations engage with citizens in an efficient and transparent way. The platform uses IBM Granite model (from Hugging Face) integrated with Gradio UI for user interaction in Google Colab.

2. Problem Statement

In today's world, citizens face several challenges when engaging with government systems: - Long waiting times for query resolutions. - Lack of transparency and trust. - Difficulty in accessing services in rural areas. - Language barriers between citizens and officials. - Inefficient grievance redressal mechanisms. Citizen AI is designed to solve these problems by enabling intelligent, 24/7 automated assistance.

3. Objectives

- To develop an AI chatbot that can handle citizen queries in natural language. - To use IBM Granite large language model for generating meaningful responses. - To build a user-friendly interface using Gradio. - To demonstrate real-time citizen engagement through AI.

4. System Requirements

Software Requirements: - Python 3.10+ - Google Colab - Libraries: Transformers, Torch, Gradio

Hardware Requirements: - Any system with internet access (Colab handles computation on cloud).

5. Methodology

1. Dataset/Model Selection – IBM Granite model (ibm-granite/granite-3.2-2b-instruct) from Hugging Face. 2. Environment Setup – Google Colab with required libraries (transformers, torch, gradio). 3. Model Loading – Pretrained Granite model loaded for text generation. 4. Function Creation – Python function ask_citizen_ai() processes user queries. 5. Interface Design – Gradio is used to create an interactive chatbot interface. 6. Deployment – Runs directly in Colab; Gradio provides a shareable demo link.

6. Implementation (Code Summary)

```
# Install dependencies !pip install transformers torch gradio -q # Import libraries from transformers
import AutoModelForCausalLM, AutoTokenizer, pipeline import gradio as gr # Load Granite model
model_name = 'ibm-granite/granite-3.2-2b-instruct' tokenizer =
AutoTokenizer.from_pretrained(model_name) model =
AutoModelForCausalLM.from_pretrained(model_name) # Pipeline generator =
pipeline('text-generation', model=model, tokenizer=tokenizer) # Function def ask_citizen_ai(query):
response = generator(query, max_length=200, num_return_sequences=1) return
response[0]['generated_text'] # Gradio UI interface = gr.Interface(fn=ask_citizen_ai, inputs='text',
outputs='text', title='Citizen AI - IBM Granite') interface.launch()
```

7. Results

Successfully created a chatbot that answers queries intelligently. Citizens can ask problem-based queries like: - “How can Citizen AI improve transparency?” - “How does AI help rural governance?” The chatbot provides contextual, AI-generated responses using IBM Granite.

8. Applications

- Smart city governance - Grievance redressal and complaint management - Public service delivery (healthcare, utilities, transport) - Multilingual citizen support - Transparency in government processes

9. Future Enhancements

- Voice-based interaction - Multi-language support - Integration with government databases for real-time updates - Analytics dashboard for policymakers

10. Conclusion

The project Citizen AI demonstrates how artificial intelligence can improve citizen engagement and governance. By combining IBM Granite AI with Gradio, the platform provides an interactive, intelligent, and user-friendly solution to modern citizen service challenges.

11. Demo Video

■ Watch here:

https://drive.google.com/file/d/1xVFQbDiw4d2KTfMITOolVrnnacAp_74c/view?usp=drivesdk

12. Authors

■■■ Nivetha ■■■ Savitha Ks ■■■ Rifaya Sulthana B ■■■ Nisma A