# Citizen AI – Project Documentation

## 1. Introduction

Project Title: Citizen AI – Intelligent Citizen Engagement Platform

Team Member 1: Sukanth A 22RCN35 (Leader)

## 2. Project Overview

Purpose:
Citizen AI is a generative AI-powered civic engagement assistant that provides quick, accurate answers about government services, policies, and civic issues. It also tracks public sentiment and presents simple dashboards for officials to review citizen feedback.

The project is implemented in Google Colab with IBM Granite models from Hugging Face and hosted on GitHub for collaboration.

Features:
- Citizen Query Chat – Answers questions about government schemes, services, and policies.
- Sentiment Analysis – Tracks public feedback and generates insights.
- Dashboard for Officials – Simple visualizations of citizen input.
- Colab Integration – Runs in Colab with GPU for smooth performance.
- Gradio UI – Provides an interactive, accessible interface.
- GitHub Hosting – Stores source code and documentation.

## 3. Architecture

Google Colab Environment:
- Provides execution with GPU (T4).
- Handles library installation, model loading, and Gradio app launch.

Frontend (Gradio):
- Includes tabs for citizen chat, sentiment analysis, and feedback dashboard.
- Generates a public link from Colab for easy use.

Backend (Python in Colab):
- Logic for analyzing queries and generating responses.
- Integrates Granite models for NLP tasks.

LLM Integration (IBM Granite):
- Model used: granite-3.2-2b-instruct.

Data Storage:
- GitHub used for storing code and docs.
- Citizen feedback can be logged locally or extended with a database.

## 4. Setup Instructions

Prerequisites:
- Google Account with Colab access
- Hugging Face account for IBM Granite models
- GitHub repository for project hosting
- Internet connection

Installation in Colab:

```
!pip install transformers torch gradio -q
```

## 5. Folder Structure (GitHub Repository)

```
citizen-ai/
├── citizen_ai.ipynb       # Main Colab notebook
├── requirements.txt       # Dependencies
├── model_integration.py   # Granite model loading
├── gradio_ui.py           # Gradio interface
├── utils/                 # Helper functions (sentiment, feedback)
├── README.md              # Documentation
```

## 6. Running the Application

1. Open Google Colab.
2. Clone GitHub repo:
```
  !git clone https://github.com/your-username/citizen-ai.git
  %cd citizen-ai
```
3. Install dependencies.
4. Run all cells in citizen_ai.ipynb.
5. A Gradio link will be generated.
6. Open the link to interact with the assistant.

## 7. API Documentation

Internal functions (executed inside Colab):
- citizen_chat(query) → Answers policy/service-related questions.
- analyze_sentiment(feedback) → Returns sentiment (positive/neutral/negative).
- generate_dashboard(data) → Produces simple visualizations for officials.

## 8. Authentication

- Hugging Face API key required for Granite models.
- GitHub account for hosting project files.

## 9. User Interface

- Tab 1 – Citizen Chat (ask questions).
- Tab 2 – Sentiment Analysis (analyze feedback).
- Tab 3 – Dashboard (visual summary for officials).

Runs inside Colab with a Gradio shareable link.

## 10. Testing

- Unit Testing: Checked sentiment and query response functions.
- Manual Testing: Verified in Colab with Gradio interface.
- Edge Cases: Tested blank queries, long text, irrelevant inputs.

## 11. Screenshots

- Colab notebook setup.
- Gradio link generated in Colab.
- Citizen chat example.
- Sentiment analysis output.
- Dashboard visualization.

## 12. Known Issues

- Colab session expires after ~12 hours.
- Internet required for Hugging Face model.
- Limited dashboard customization in current version.

## 13. Future Enhancements

- Add real-time feedback collection from citizens.
- Integrate with government datasets/APIs.
- Build mobile app version.
- Provide multi-language support.
- Enhance dashboard with advanced analytics.