# Citizen AI: Intelligent Citizen Engagement Platform

#### 1. Project Overview

Citizen AI is an AI-powered citizen engagement platform designed to:

- Provide quick, accurate responses about government services and civic issues.
- Track public sentiment and present insights on simple dashboards for officials.
- Run efficiently in Google Colab using IBM's Granite models for cost-effectiveness and performance.

#### **TEAM MEMBERS**

A.kalaivanan

A.Akash

R.Saravana kumar

#### 2. Pre-requisites

Before starting, ensure you are familiar with:

- 1. Gradio Framework https://www.gradio.app/guides/
- 2. IBM Granite Models https://huggingface.co/ibm-granite
- 3. Python Programming https://docs.python.org/3/
- 4. Version Control with Git https://git-scm.com/docs/git
- 5. Google Colab with GPU https://www.geeksforgeeks.org/python/how-to-use-gpu-ingoogle-colab/

#### 3. Project Workflow

#### **Activity 1: Explore Naan Mudhalvan Smart Internz Portal**

- 1. Search for Naan Mudhalvan Smart Internz in your browser.
- 2. Login  $\rightarrow$  Go to Projects  $\rightarrow$  Select Citizen AI.

- 3. Access Resources → Guided Project for step-by-step instructions.
- 4. Open Project Workspace to track progress & upload your demo link.

#### **Activity 2: Choose IBM Granite Model from Hugging Face**

- 1. Go to https://huggingface.co/.
- 2. Sign up / Login → Search for IBM Granite models.
- 3. For this project, use granite-3.2-2b-instruct (fast & lightweight).

#### **Activity 3: Run Application in Google Colab**

- 1. Open https://colab.research.google.com/.
- 2. Create a New Notebook → Rename to Citizen AI.
- 3. Set runtime: T4 GPU  $\rightarrow$  Save.
- 4. Install dependencies:

!pip install transformers torch gradio -q

- 5. Add your project code (provided in Guided Project or sample link).
- 6. Run cells  $\rightarrow$  Launch Gradio App  $\rightarrow$  Open URL to view the chatbot.

#### **Activity 4: Upload Your Project to GitHub**

- 1. Go to https://github.com/  $\rightarrow$  Sign in / Sign up.
- 2. Create a new repository (e.g., IBM-Project).
- 3. Enable Add README file.
- 4. From Google Colab: File  $\rightarrow$  Download  $\rightarrow$  Download as .py.
- 5. In GitHub repo  $\rightarrow$  Add File  $\rightarrow$  Upload Files  $\rightarrow$  Choose your project file.
- 6. Commit changes  $\rightarrow$  Project is now live on GitHub.

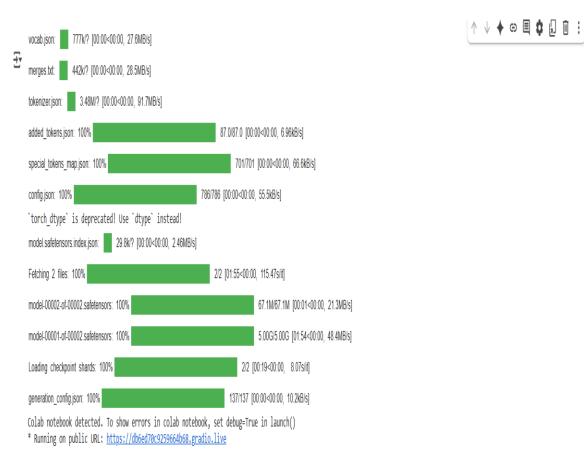
#### 4. Expected Output

- A working citizen engagement chatbot powered by IBM Granite.
- Gradio web UI for users to interact with the model.
- GitHub repo containing your code for version control.
- Dashboard-like insights into citizen queries & sentiment.

## **Project Screenshot**

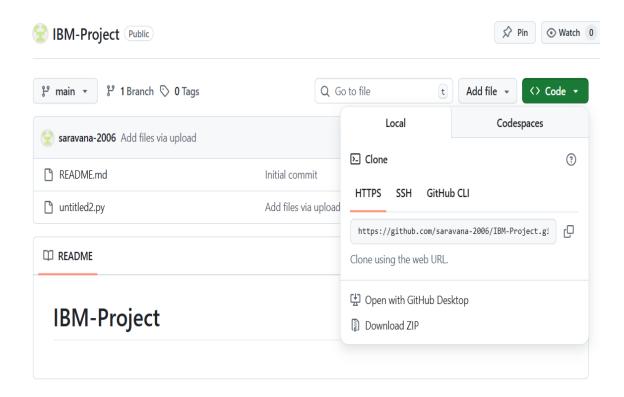
```
# run this project file in google collab by changing run type to T4 GPU
!pip install transformers torch gradio -q
import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
# Load model and tokenizer
model name = "ibm-granite/granite-3.2-2b-instruct"
tokenizer = AutoTokenizer.from pretrained(model name)
model = AutoModelForCausalLM.from pretrained(
    model name,
    torch_dtype=torch.float16 if torch.cuda.is available() else torch.float32,
    device map="auto" if torch.cuda.is available() else None
if tokenizer.pad_token is None:
    tokenizer.pad token = tokenizer.eos token
def generate response(prompt, max length=1024):
    inputs = tokenizer(prompt, return tensors="pt", truncation=True, max length=512)
    if torch.cuda.is available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}
```

### **Screenshot 2**



This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (

### **Screen Shot 3**



## Thank you