

# 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-in-google-colab/>

## 3. Project Workflow

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

1. Search for Naan Mudhalvan Smart Internz in your browser.
2. Login → Go to Projects → 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 → 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 → Launch Gradio App → Open URL to view the chatbot.

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

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

↑ ↓ ✦ ⌂ ⚙ 📄 🗑 ⋮

📄

vocab.json: 777k/? [00:00<00:00, 27.6MB/s]

merges.txt: 442k/? [00:00<00:00, 28.5MB/s]

tokenizer.json: 3.48M/? [00:00<00:00, 91.7MB/s]

added\_tokens.json: 100% 87.0/87.0 [00:00<00:00, 6.96KB/s]

special\_tokens\_map.json: 100% 701/701 [00:00<00:00, 66.6KB/s]

config.json: 100% 786/786 [00:00<00:00, 55.5KB/s]

`torch\_dtype` is deprecated! Use `dtype` instead!

model.safetensors.index.json: 29.8k/? [00:00<00:00, 2.46MB/s]

Fetching 2 files: 100% 2/2 [01:55<00:00, 115.47s/it]

model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:01<00:00, 21.3MB/s]

model-00001-of-00002.safetensors: 100% 5.00G/5.00G [01:54<00:00, 48.4MB/s]

Loading checkpoint shards: 100% 2/2 [00:19<00:00, 8.07s/it]


generation\_config.json: 100% 137/137 [00:00<00:00, 10.2kB/s]



Colab notebook detected. To show errors in colab notebook, set debug=True in launch()




\* Running on public URL: <https://db6ed70c9259664b68.gradio.live>



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


 IBM-Project Public


 Pin  Watch 0


 main  1 Branch  0 Tags

 Add file  Code

 saravana-2006 Add files via upload

 README.md Initial commit



 untitled2.py Add files via upload

 README


## IBM-Project

Local


Codespaces


 Clone 

HTTPS SSH GitHub CLI



Clone using the web URL.

 Open with GitHub Desktop

 Download ZIP

**Thank you**