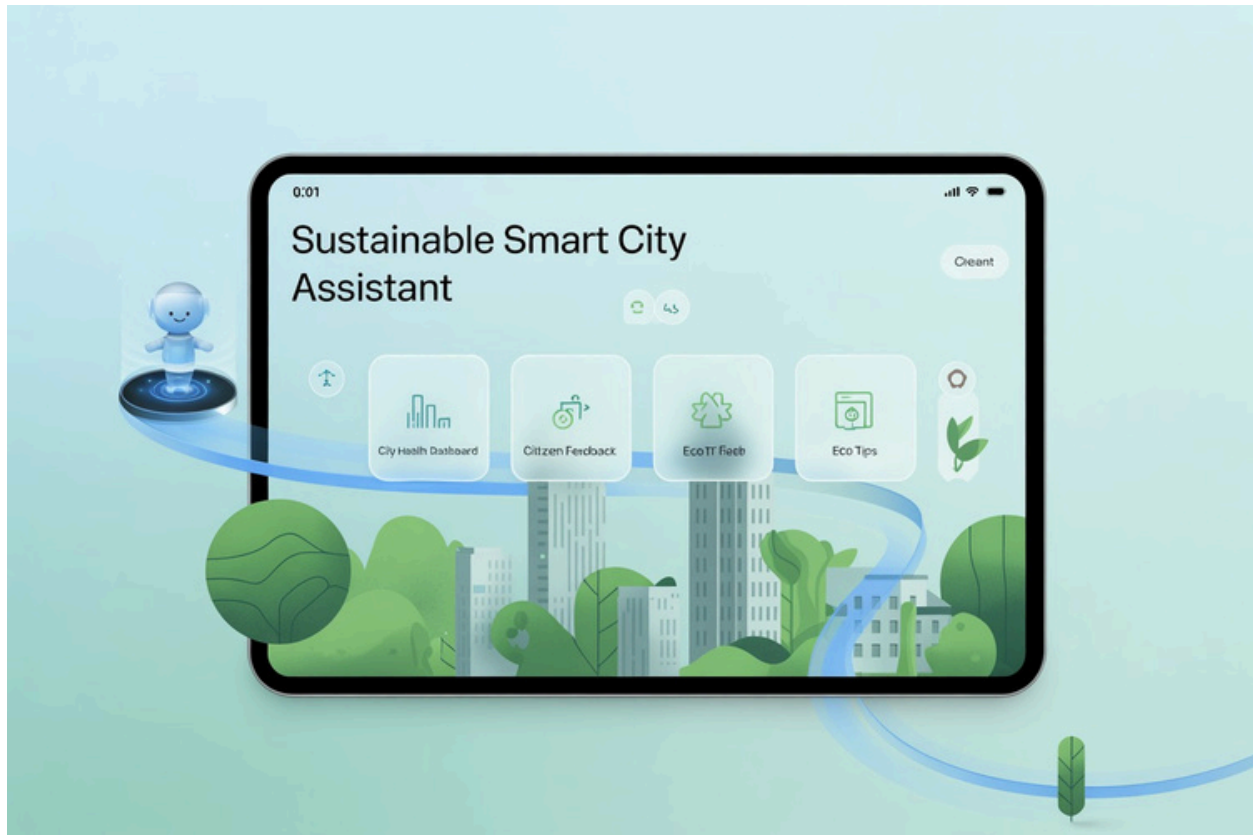


SUSTAINABLE SMART CITY ASSISTANT USING IBM GRANITE LLM

Generative AI with IBM



TEAM MEMBERS :

1. HAMSINI V
2. BHAVANIKA M E
3. PARTHIBAN G
4. ELANCHEZHIAN M

INTRODUCTION:

The Sustainable Smart City Assistant project is designed to address the pressing need for efficient city governance, sustainability practices, and citizen engagement. By leveraging IBM Granite LLM (Large Language Model) available on Hugging Face, this assistant acts as a digital companion for cities to monitor health indicators, gather citizen feedback, generate eco-friendly tips, and provide document summarization. The use of generative AI ensures that responses are context-aware, accurate, and tailored to the dynamic needs of a smart urban environment.

The project workflow involves four main stages: exploring the Naan Mudhalvan Smart Interz portal for resources, selecting an appropriate IBM Granite model from Hugging Face, deploying the application using Google Colab with GPU acceleration, and finally, uploading and version-controlling the project on GitHub. Each stage is carefully structured to help students gain practical exposure to modern AI tools while simultaneously contributing to the broader vision of sustainability in urban areas.

The uniqueness of this project lies in its practical application of Generative AI within the context of real-world challenges. For example, city administrators can rely on the assistant to track urban health metrics through a City Health Dashboard, while citizens can actively engage by sharing their concerns and receiving useful eco-friendly lifestyle suggestions. Moreover, document summarization capabilities simplify decision-making by condensing large reports into concise, actionable insights.

From an academic perspective, this project is highly relevant as it combines multiple domains—Artificial Intelligence, Sustainable Development, Cloud Computing, and Open-Source Collaboration. Students not only learn to build an AI-powered system but also gain experience in version control (GitHub), cloud-based execution (Colab), and ethical AI usage (sustainability-focused applications).

In essence, the Sustainable Smart City Assistant reflects how cutting-edge AI technologies like IBM Granite LLM can be harnessed to design intelligent urban solutions. It bridges the gap between theoretical knowledge and practical innovation, empowering students to contribute to a greener, smarter, and more sustainable future.

PROJECT DESCRIPTION:

SustainableSmartCity Assistant uses the Granite model from Hugging Face to help with city sustainability, governance, and citizen engagement. It includes quick tools for a City Health Dashboard, citizen feedback, document summaries and eco tips. This project will be deployed in Google Colab using Granite for easy setup and smooth performance.

PRE-REQUISITES:

1. Gradio Framework Knowledge: [Gradio Documentation](#)
2. IBM Granite Models (Hugging Face): [IBM Granite models](#)
3. Python Programming Proficiency: [Python Documentation](#)
4. Version Control with Git: [Git Documentation](#)
5. Google Collab's T4 GPU Knowledge: [Google collab](#)

PROJECT WORKFLOW:

Activity-1: Exploring Naan Mudhalavan Smart Interz Portal.

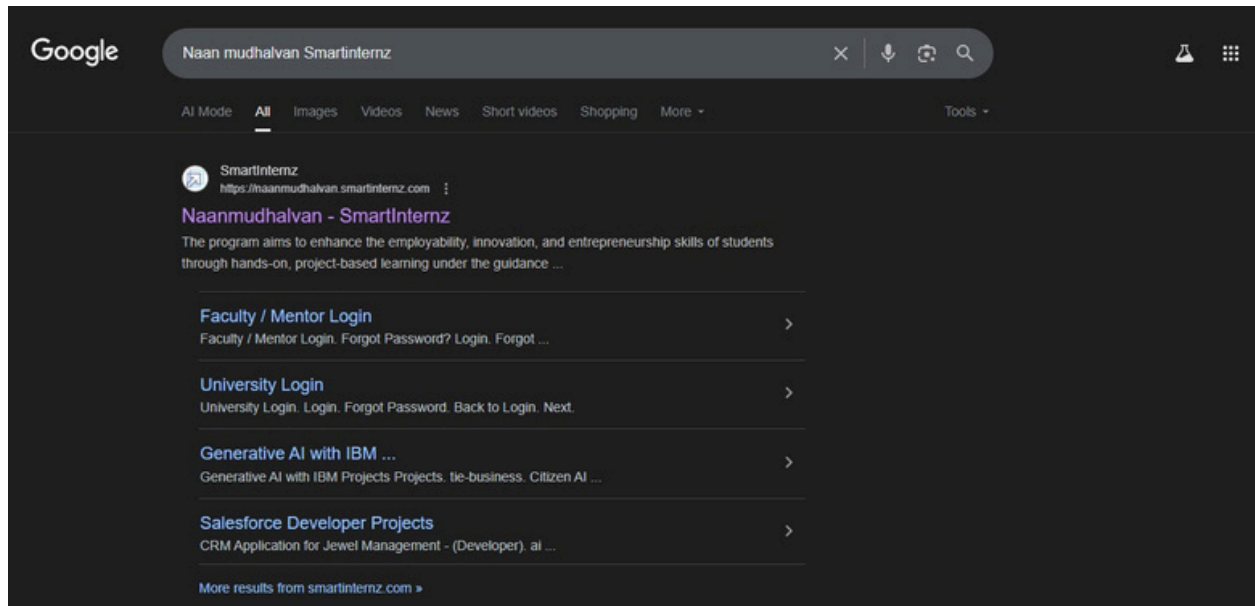
Activity-2: Choosing a IBM Granite Model From Hugging Face.

Activity-3: Running Application In Google Colab. Activity-4:

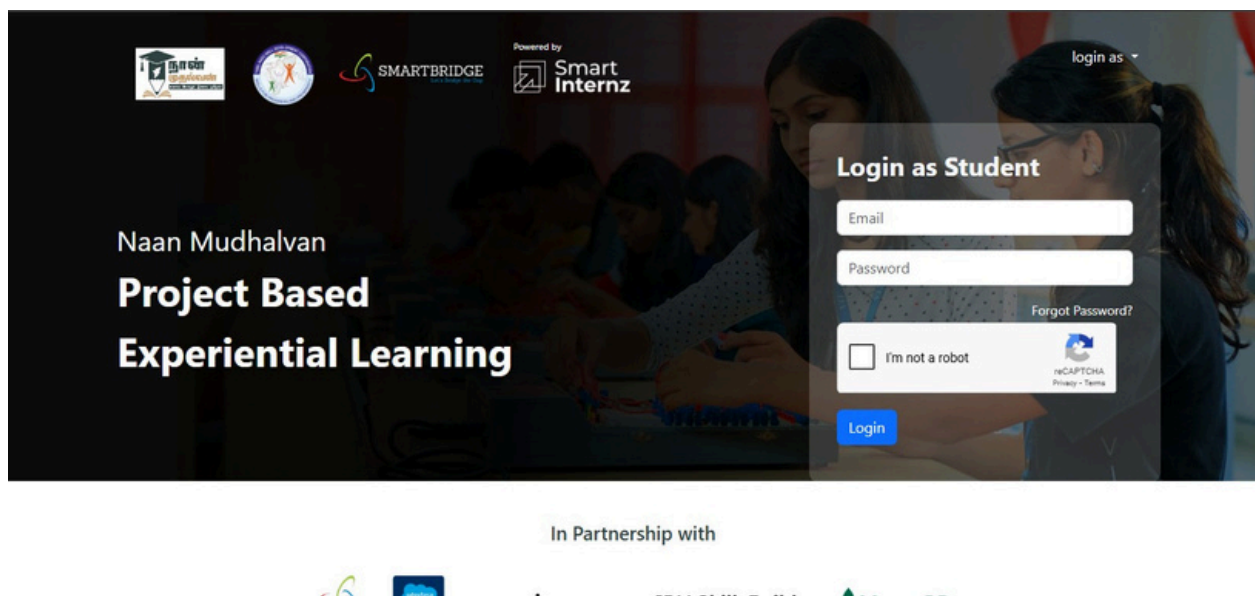
Upload your Project in Github.

Activity-1: Exploring Naan Mudhalavan Smart Interz Portal.

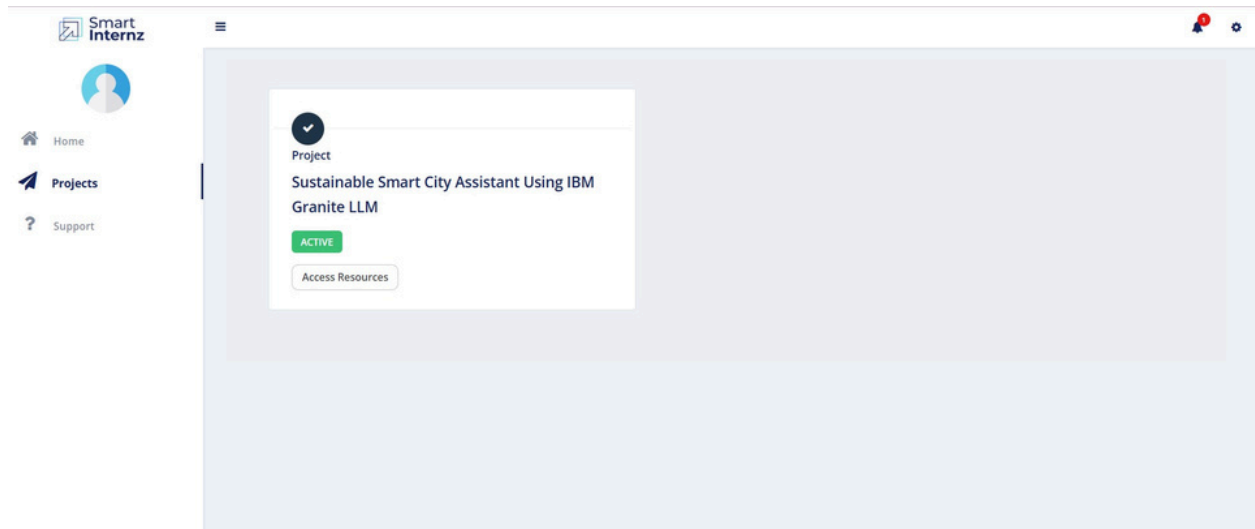
- Search for “Naan Mudhalavan Smart Interz” Portal in any Browser.



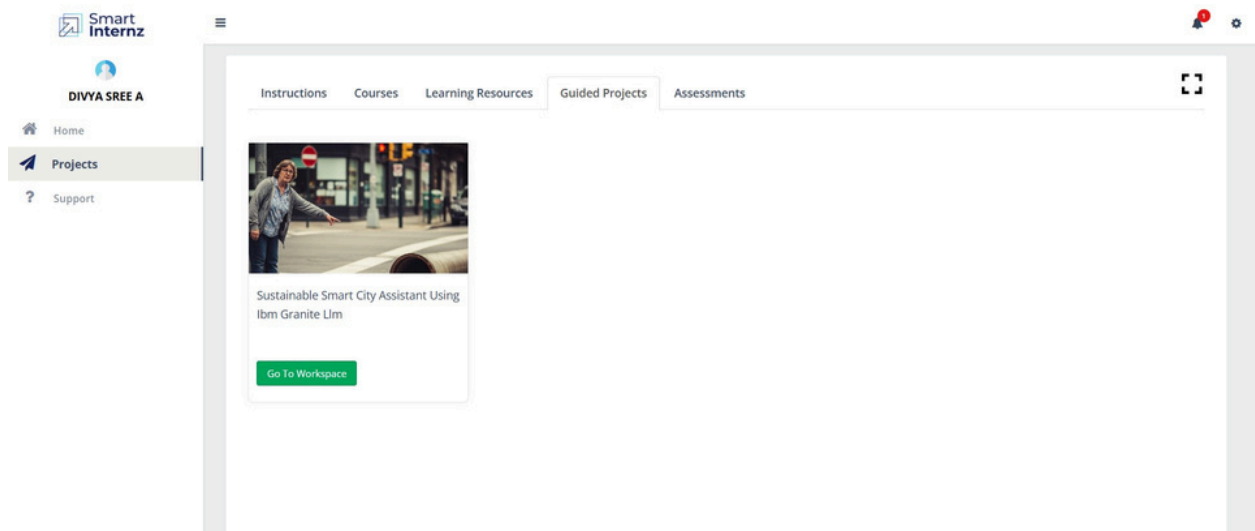
- Then Click on the first link. ([Naanmudhalvan Smartinternz](https://naanmudhalvan-smartinternz.com)) Then login with your details.



- Then you will be redirected to your account then click on “Projects” Section. There you can see which project you have enrolled in here it is “Health AI”.

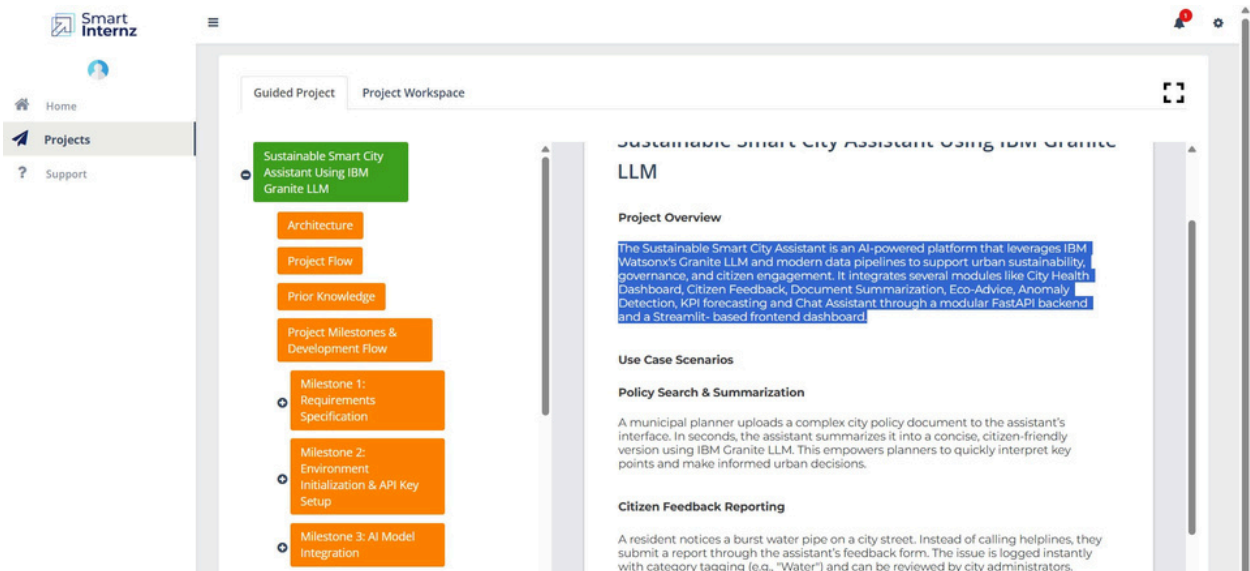


- Then click on “Access Resources” and go to the “Guided Project” Section.

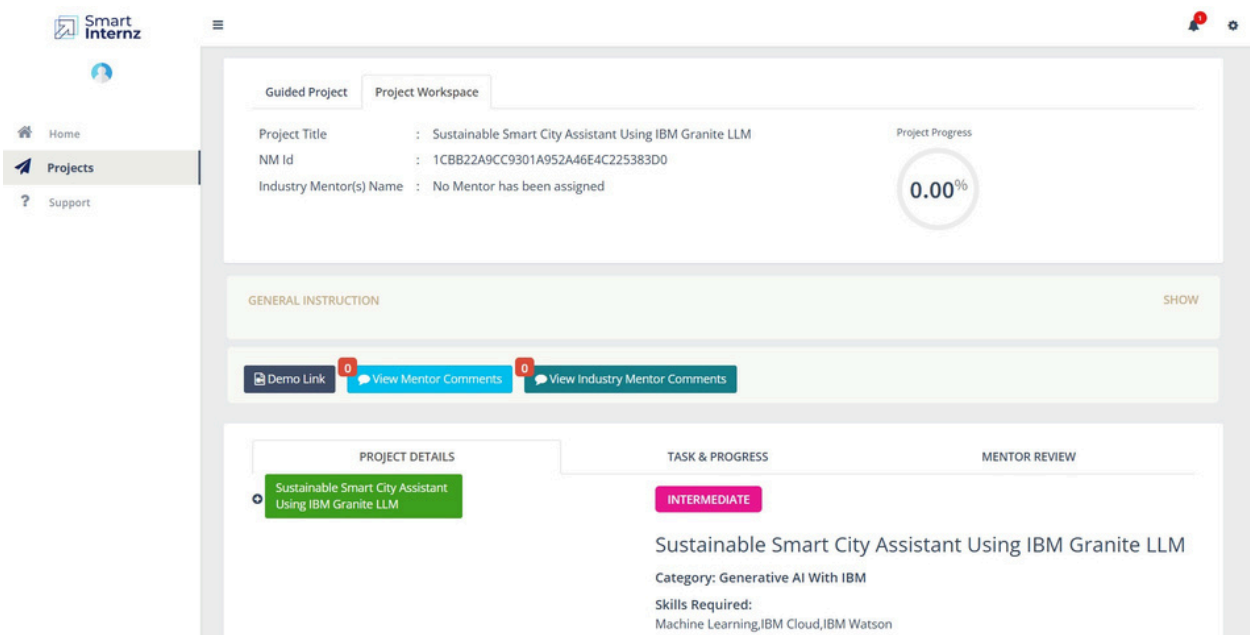


- Click on the “Go to workspace” section. Then you can find the detailed

explanation of Generative AI Project using IBM Watsonx API key.



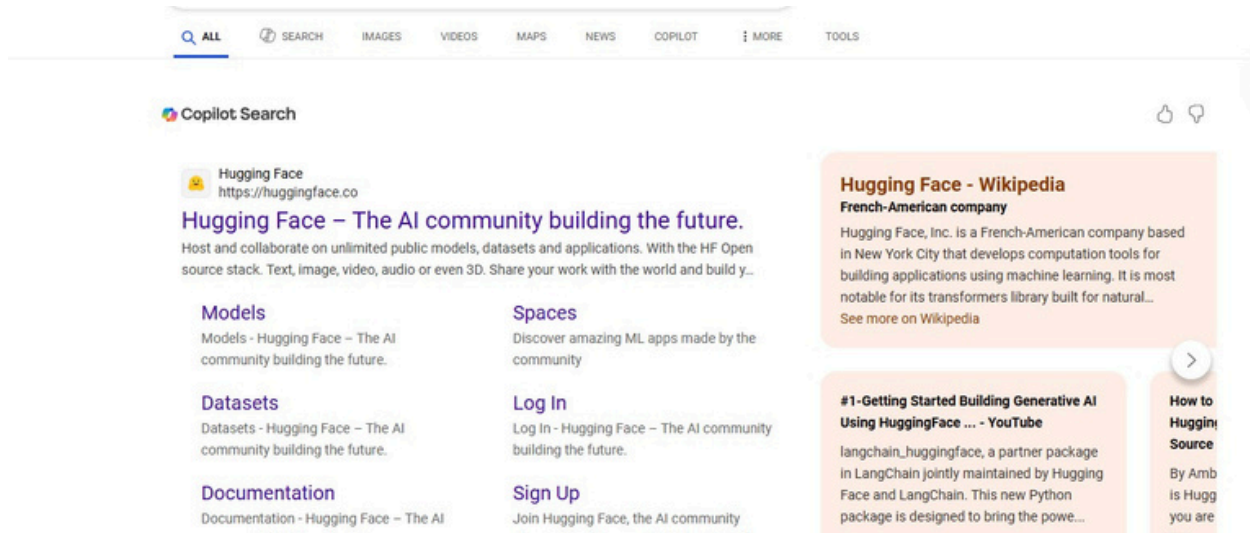
- Click on “Project Workspace”, there you can find your project progress and Place to upload “Demo link”.



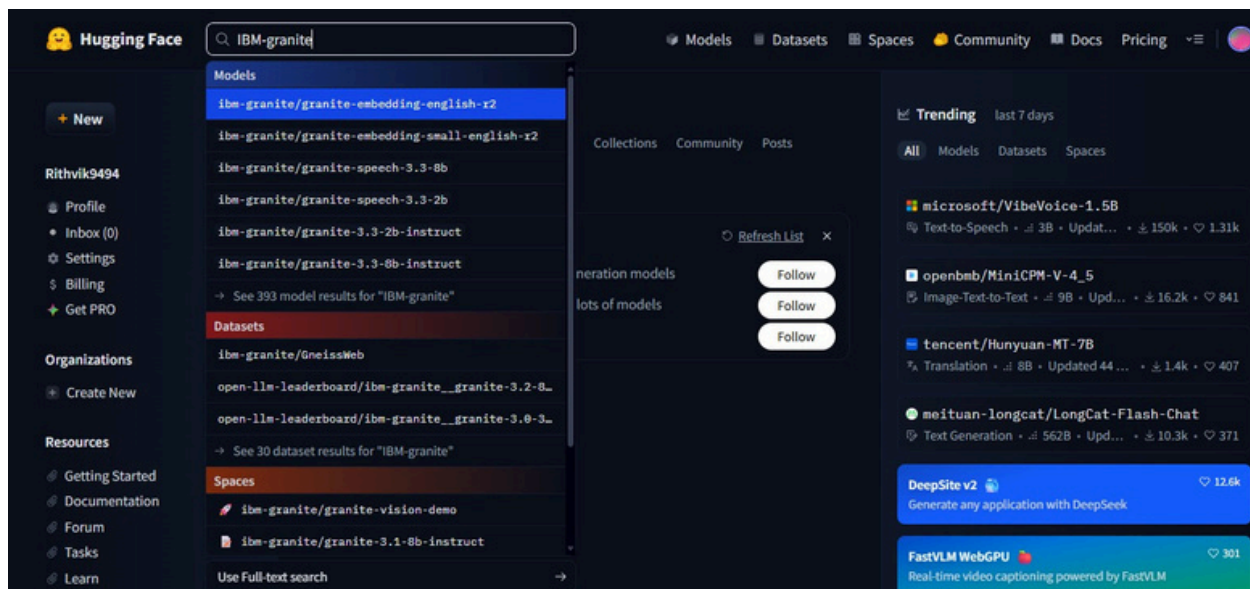
- Now we have gone through portal understanding, now lets find a IBM granite model from hugging face to integrate in our project.

Activity-2: Choose a IBM Granite model From Hugging Face.

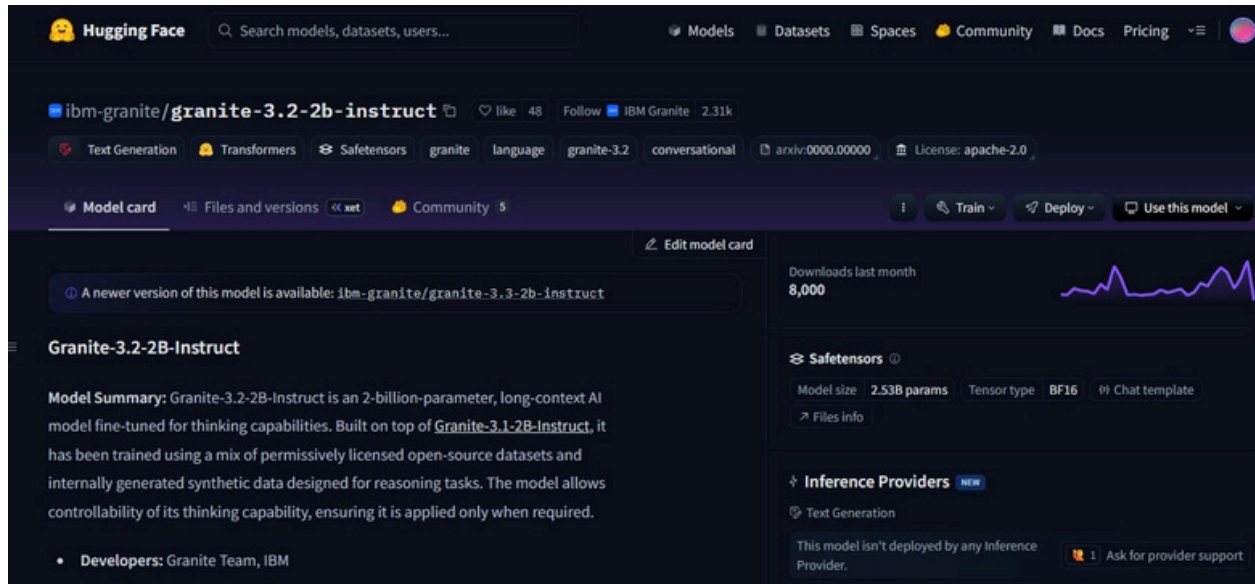
- Search for “Hugging face” in any browser.



- Then click on the first link ([Hugging Face](https://huggingface.co)), then click on signup and create your own account in Hugging Face. Then search for “IBM-Granite models” and choose any model.



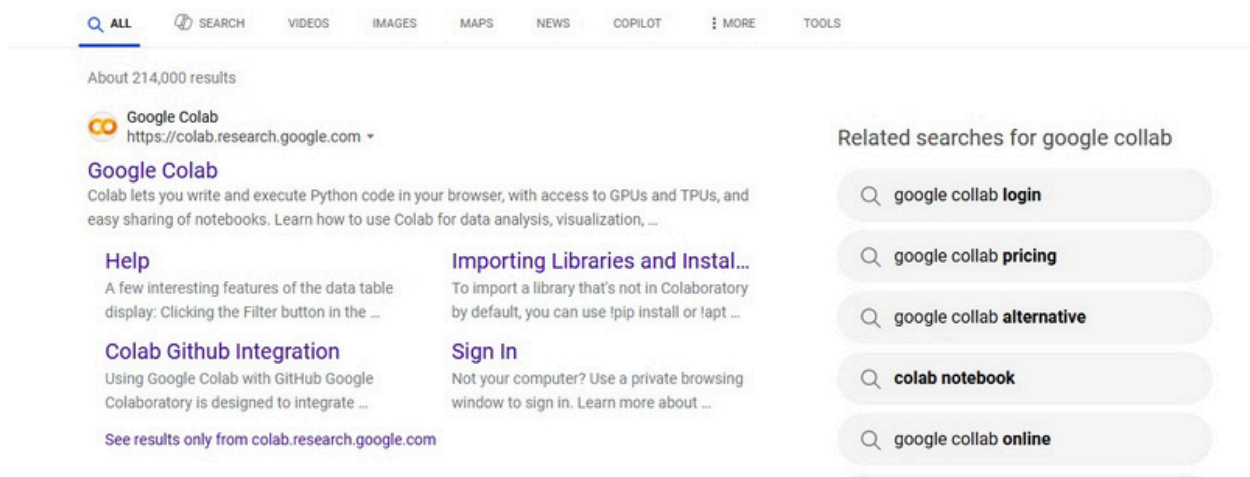
- Here for this project we are using “granite-3.2-2b-instruct” which is compatible fast and light weight.



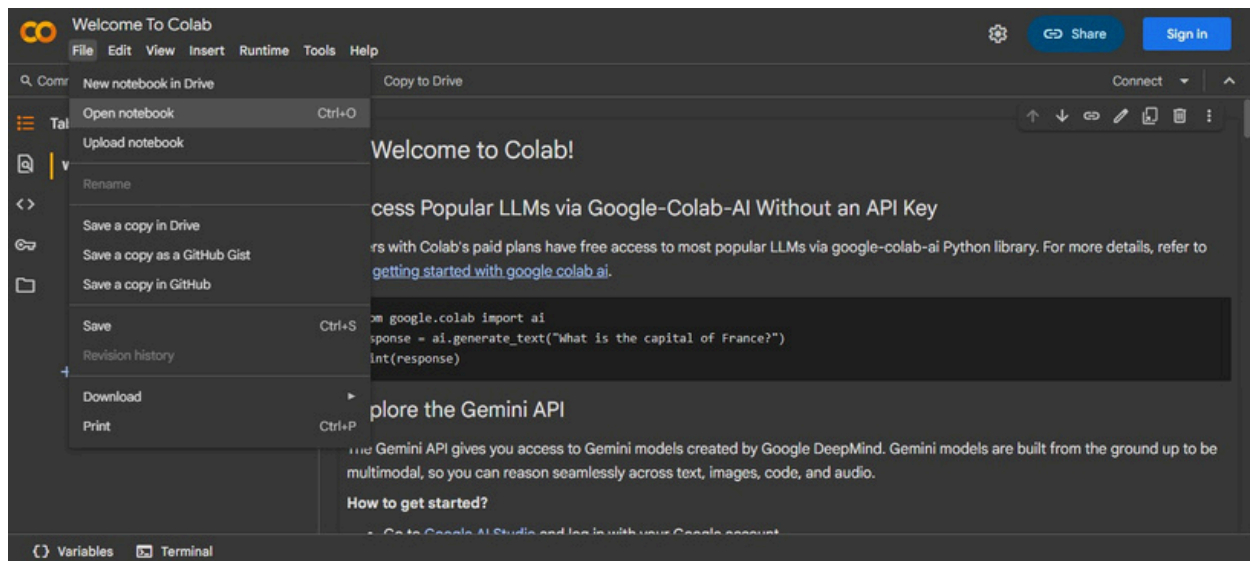
- Now we will start building our project in Google collab.

Activity-3: Running Application in Google Colab.

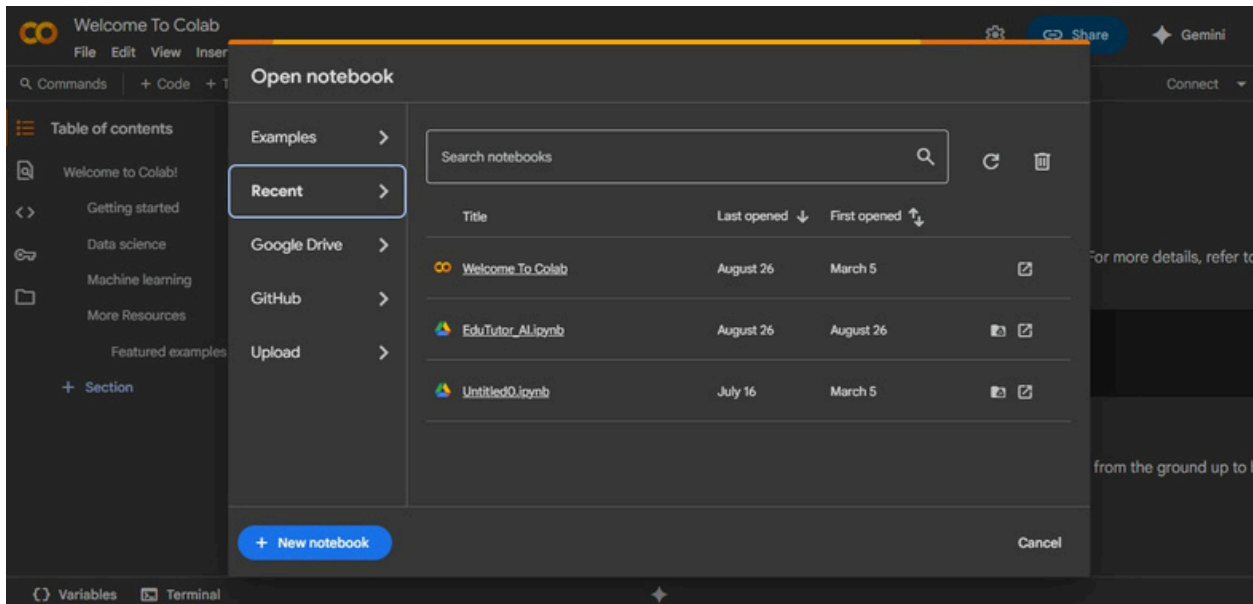
- Search for “Google collab” in any browser.



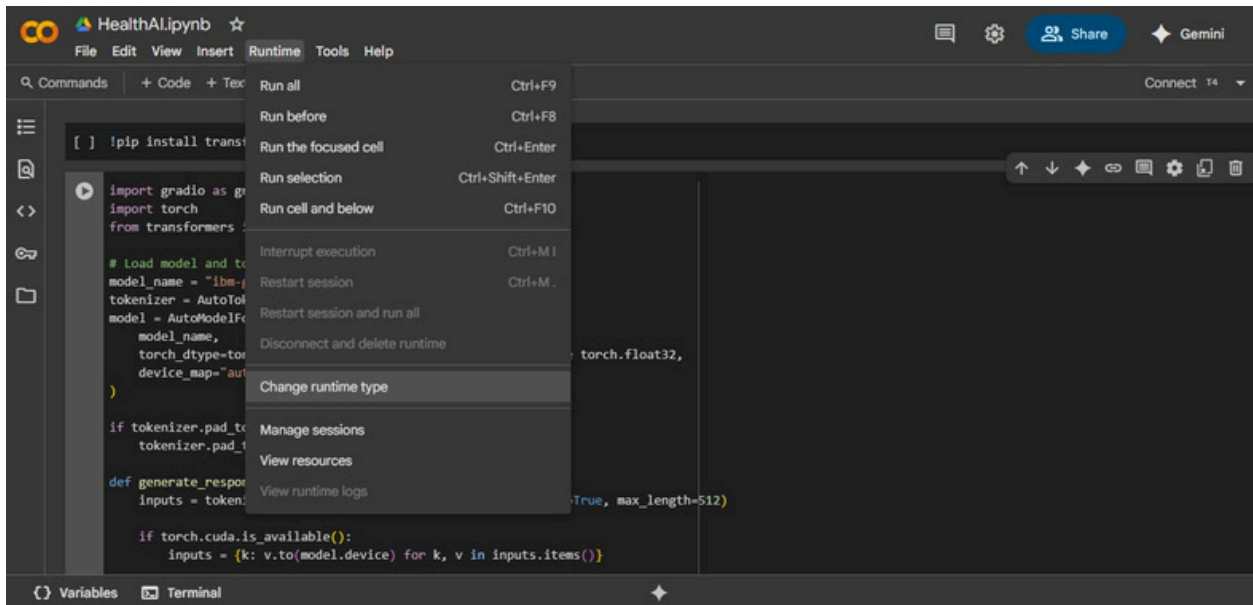
- Click on the first link ([Google Colab](https://colab.research.google.com)), then click on “Files” and then “Open Notebook”.



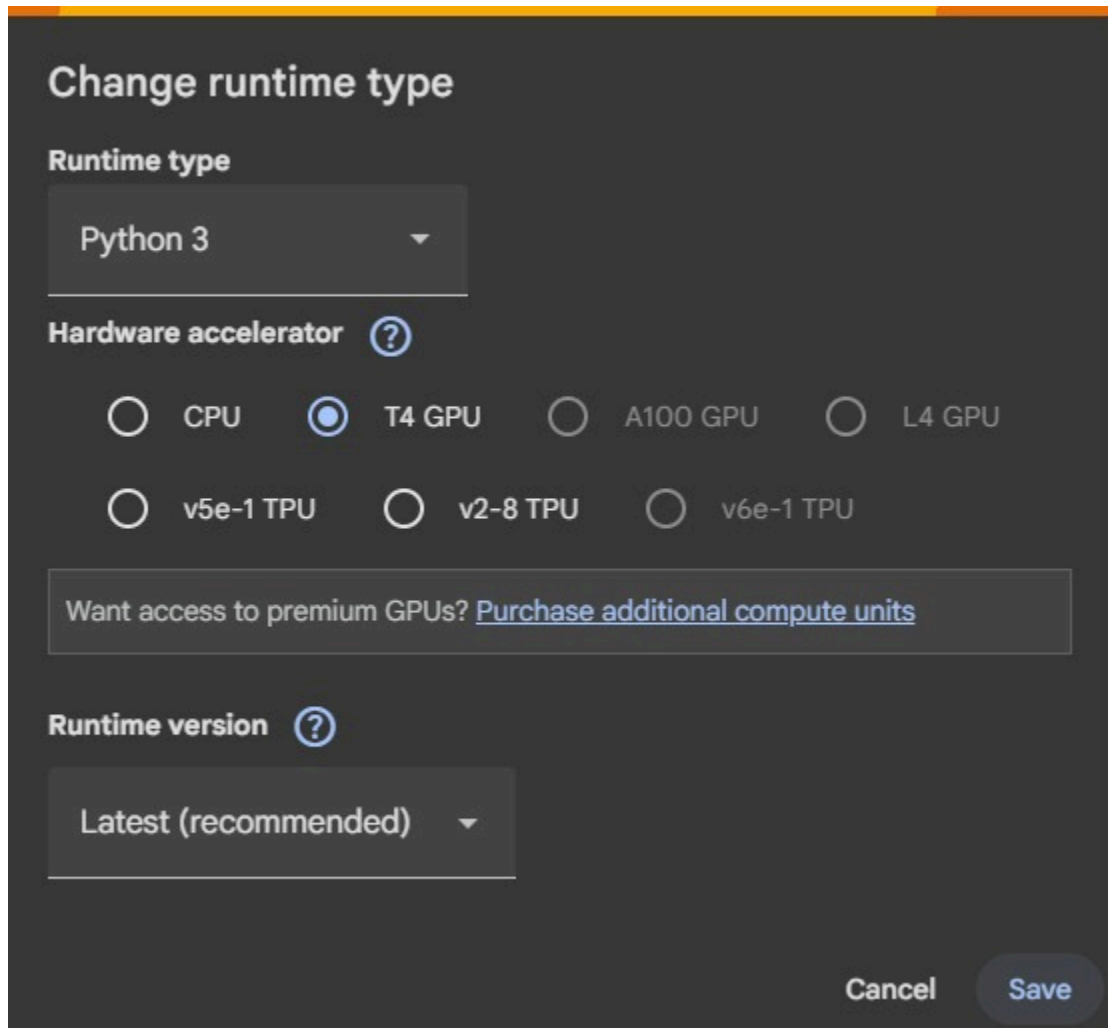
- Click on “New Notebook”



- Change the title of the notebook “Untitled” to “Health AI”. Then click on “Runtime”, then go to “Change Runtime Type”.



- Choose “T4 GPU” and click on “Save”



Change runtime type

Runtime type

Python 3 ▼

Hardware accelerator (?)

☐ CPU ☒ T4 GPU ☐ A100 GPU ☐ L4 GPU

☐ v5e-1 TPU ☐ v2-8 TPU ☐ v6e-1 TPU

Want access to premium GPUs? [Purchase additional compute units](#)

Runtime version (?)

Latest (recommended) ▼

Cancel Save

- Then run this command in the first cell “!pip install transformers torch gradio PyPDF2 -q”. To install the required libraries to run our application.



```
[ ] !pip install transformers torch gradio PyPDF2 -q
```

232.6/232.6 kB 12.3 MB/s eta 0:00:00

- Then run the rest of the code in the next cell.

```

1
2 import gradio as gr
3 import torch
4 from transformers import AutoTokenizer, AutoModelForCausalLM
5 import PyPDF2
6 import io
7
8 # load model and tokenizer
9 model_name = "ibm-granite/granite-3.2-2b-instruct"
10 tokenizer = AutoTokenizer.from_pretrained(model_name)
11 model = AutoModelForCausalLM.from_pretrained(
12     model_name,
13     torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
14     device_map="auto" if torch.cuda.is_available() else None
15 )
16
17 if tokenizer.pad_token is None:
18     tokenizer.pad_token = tokenizer.eos_token
19
20 def generate_response(prompt, max_length=1024):
21     inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)
22
23     if torch.cuda.is_available():
24         inputs = {k: v.to(model.device) for k, v in inputs.items()}
25
26     with torch.no_grad():
27         outputs = model.generate(
28             **inputs,
29             max_length=max_length,
30             temperature=0.7,
31             do_sample=True,
32             pad_token_id=tokenizer.eos_token_id
33         )
34
35     response = tokenizer.decode(outputs[0], skip_special_tokens=True)
36     response = response.replace(prompt, "").strip()
37     return response

```

```

38
39 def extract_text_from_pdf(pdf_file):
40     if pdf_file is None:
41         return ""
42
43     try:
44         pdf_reader = PyPDF2.PdfReader(pdf_file)
45         text = ""
46         for page in pdf_reader.pages:
47             text += page.extract_text() + "\n"
48         return text
49     except Exception as e:
50         return f"Error reading PDF: {str(e)}"
51
52 def eco_tips_generator(problem_keywords):
53     prompt = f"Generate practical and actionable eco-friendly tips for sustainable living related to: {problem_keywords}. Provide specific solutions and suggestions:"
54     return generate_response(prompt, max_length=1000)
55
56 def policy_summarization(pdf_file, policy_text):
57     # Get text from PDF or direct input
58     if pdf_file is not None:
59         content = extract_text_from_pdf(pdf_file)
60         summary_prompt = f"Summarize the following policy document and extract the most important points, key provisions, and implications:\n\n{content}"
61     else:
62         summary_prompt = f"Summarize the following policy document and extract the most important points, key provisions, and implications:\n\n{policy_text}"
63
64     return generate_response(summary_prompt, max_length=1200)
65
66 # Create Gradio interface
67 with gr.Blocks() as app:
68     gr.Markdown("# Eco Assistant & Policy Analyzer")
69
70     with gr.Tabs():
71         with gr.TabItem("Eco Tips Generator"):
72             with gr.Row():
73                 with gr.Column():
74                     keywords_input = gr.Textbox(
75                         label="Environmental Problem/Keywords",
76                         placeholder="e.g., plastic, solar, water waste, energy saving...",
77                         lines=3
78                     )
79                     generate_tips_btn = gr.Button("Generate Eco Tips")
80
81                 with gr.Column():
82                     tips_output = gr.Textbox(label="Sustainable Living Tips", lines=15)
83
84             generate_tips_btn.click(eco_tips_generator, inputs=keywords_input, outputs=tips_output)
85
86         with gr.TabItem("Policy Summarization"):
87             with gr.Row():
88                 with gr.Column():
89                     pdf_upload = gr.File(label="Upload Policy PDF", file_types=[".pdf"])
90                     policy_text_input = gr.Textbox(
91                         label="Or paste policy text here",
92                         placeholder="Paste policy document text...",
93                         lines=5
94                     )
95                     summarize_btn = gr.Button("Summarize Policy")
96
97                 with gr.Column():
98                     summary_output = gr.Textbox(label="Policy Summary & Key Points", lines=20)
99
100             summarize_btn.click(policy_summarization, inputs=[pdf_upload, policy_text_input], outputs=summary_output)
101
102 app.launch(share=True)
103
104

```

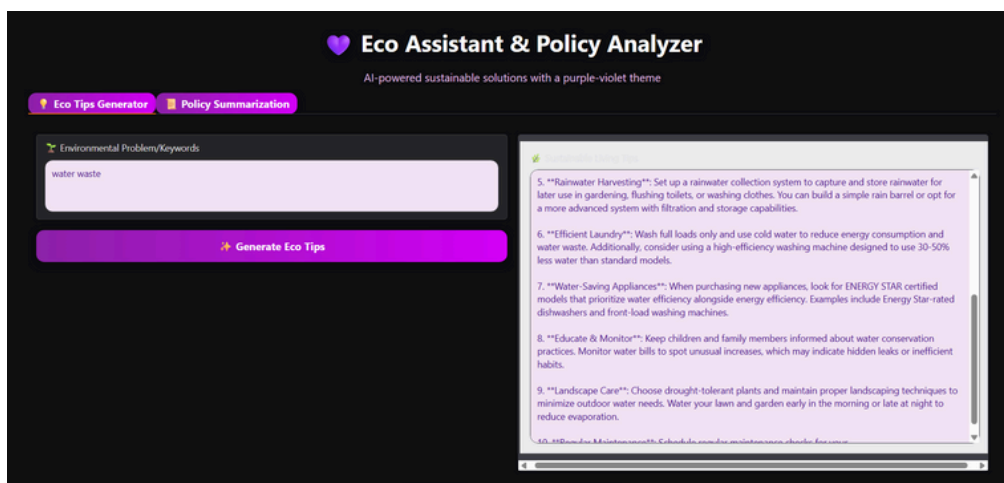
- You can find the code here in this link: [Sustainable Smart City Assistant](#)

OUTPUT:

- Now you can see our model is being Downloaded and application is running

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: <https://5475fe6c096b7ff650.gradio.live>

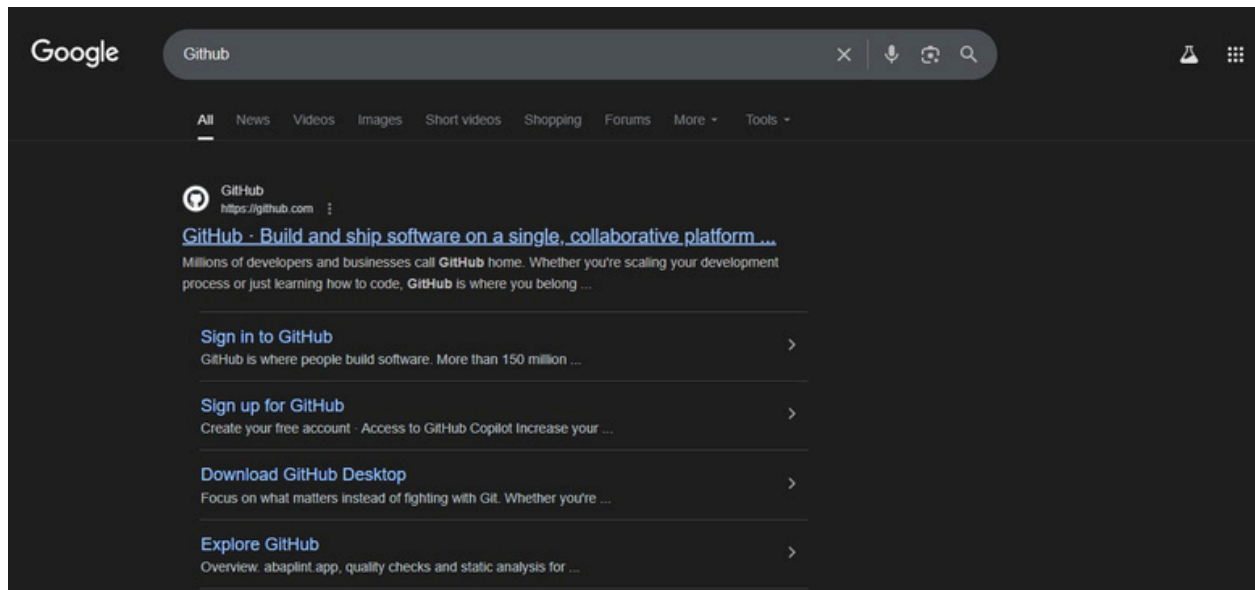
- Click on the URL to open the Gradio Application click on the link



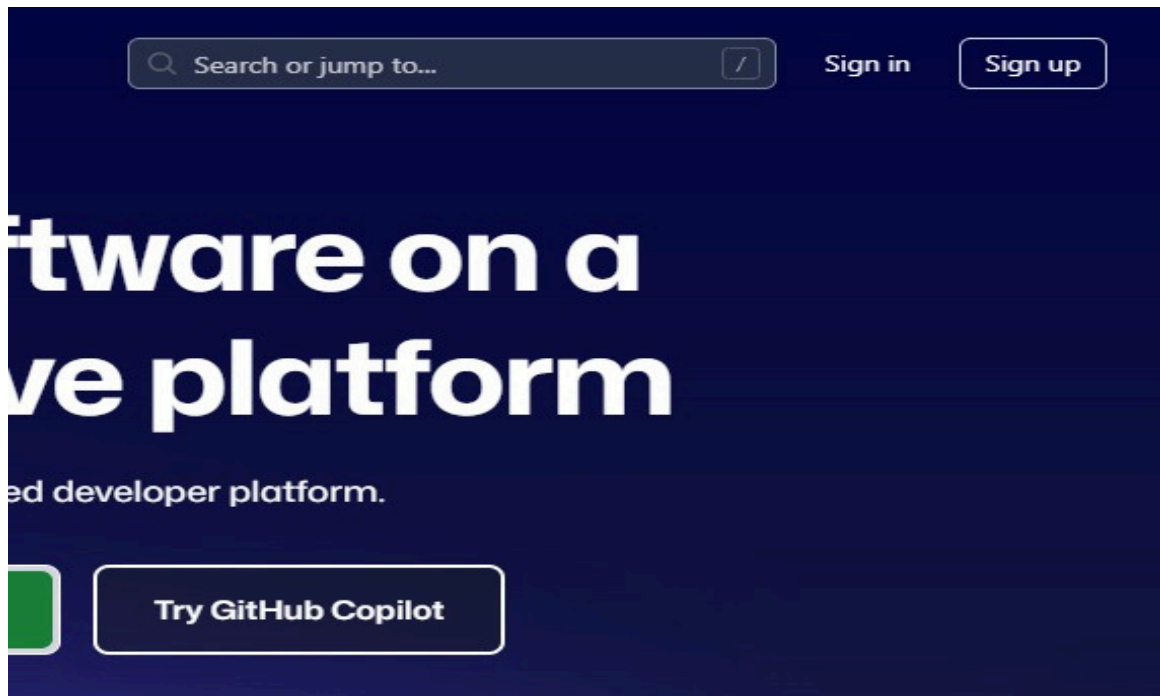
- You can View the Application is the running in the other tab

Activity-4: Upload Your Project in GitHub.

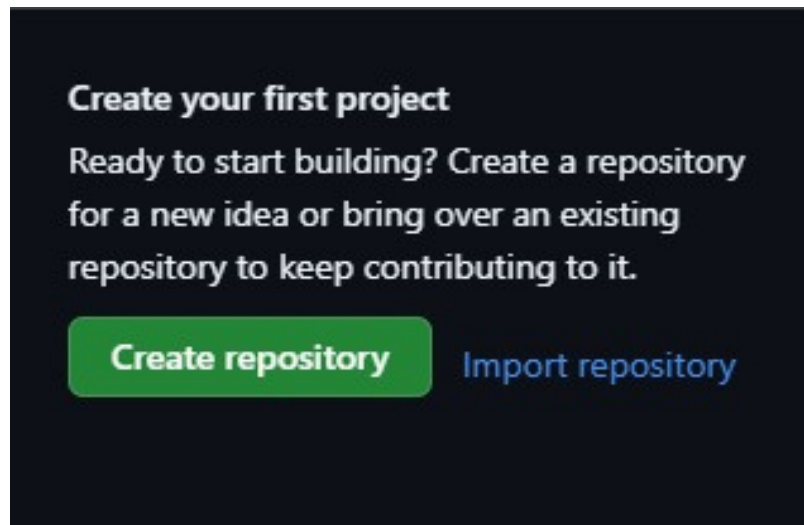
- Search for "GitHub" in any browser, then click on the first link ([GitHub](#)).



- Then click on “Signup” and create your own account in GitHub. If you already have an account click on “Sign in”

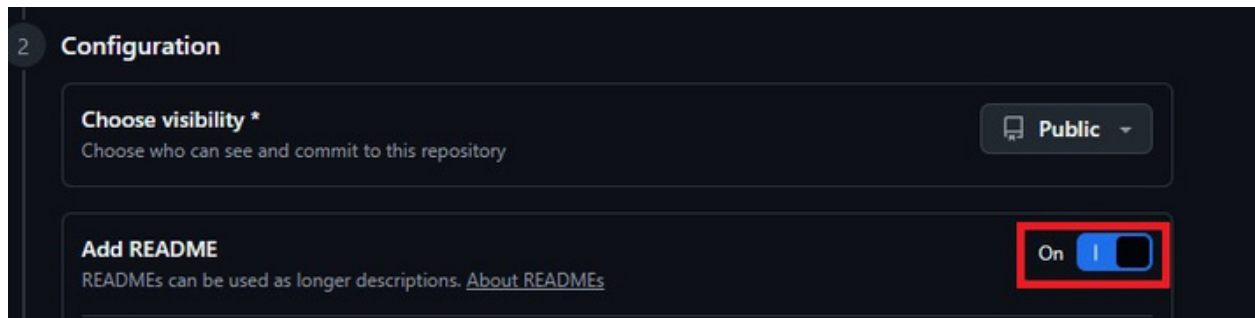


- Click on “Create repository”.

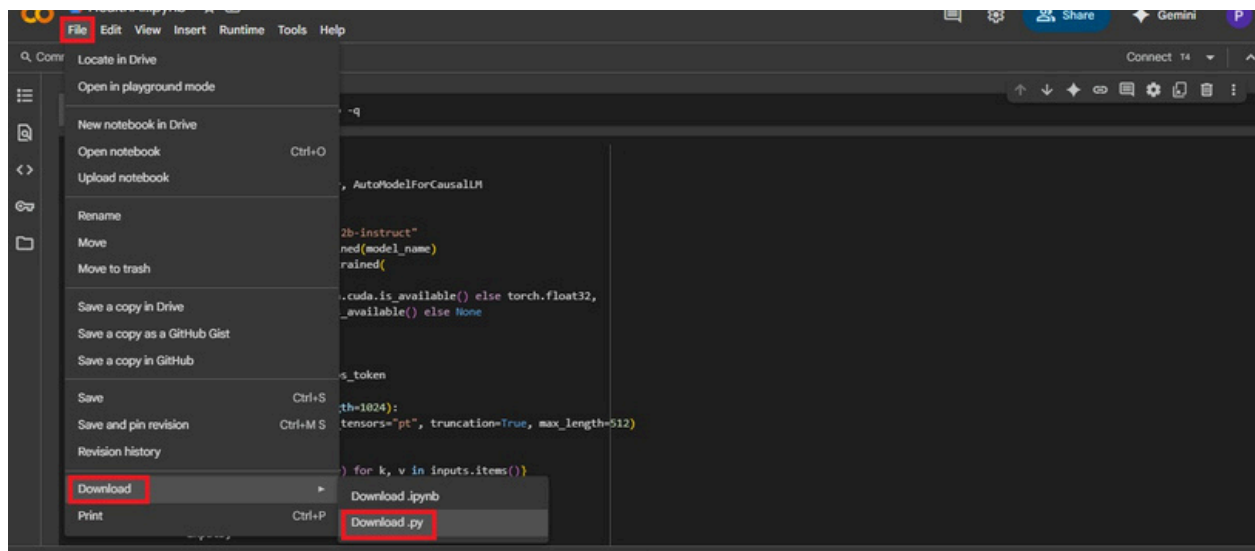


- In “General” Name your repo. (Here I have given “IBM-Project” as my repo name and it is available)

- In “Configurations” Turn On “Add readme” file Option.

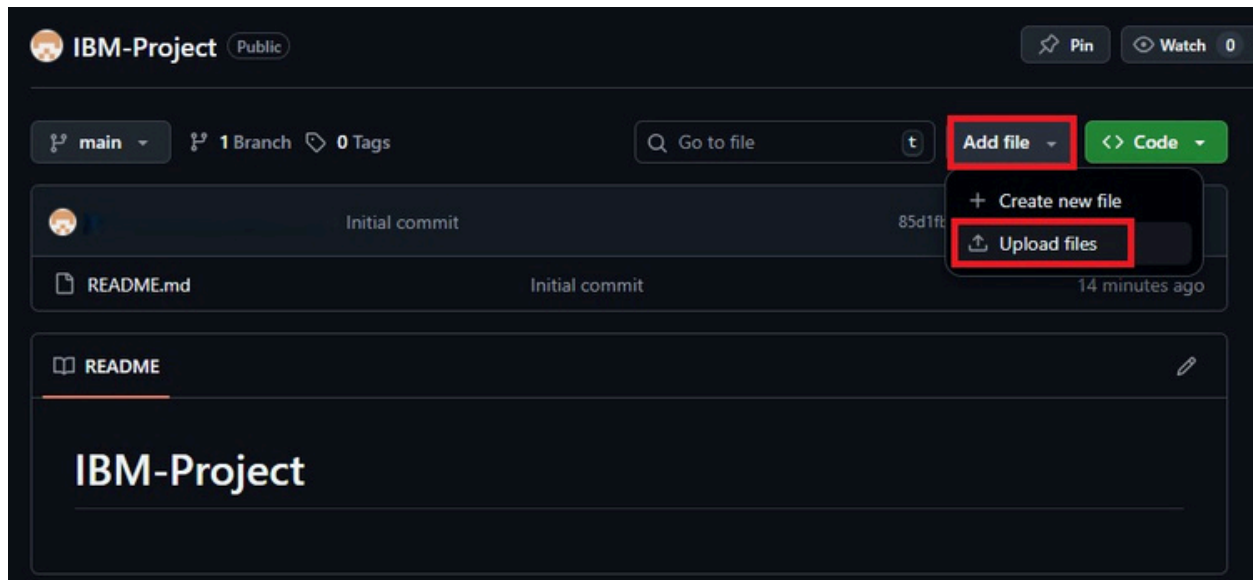


- Now Download your code from Google collab by Clicking on “File”, then Goto “Download” then download as “.py”.

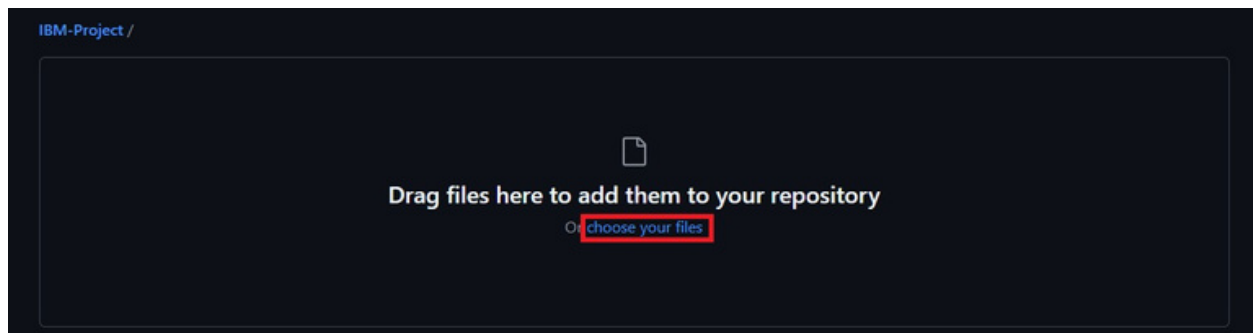


- Then your repository is created, then Click on “Add file” Option. Then Click

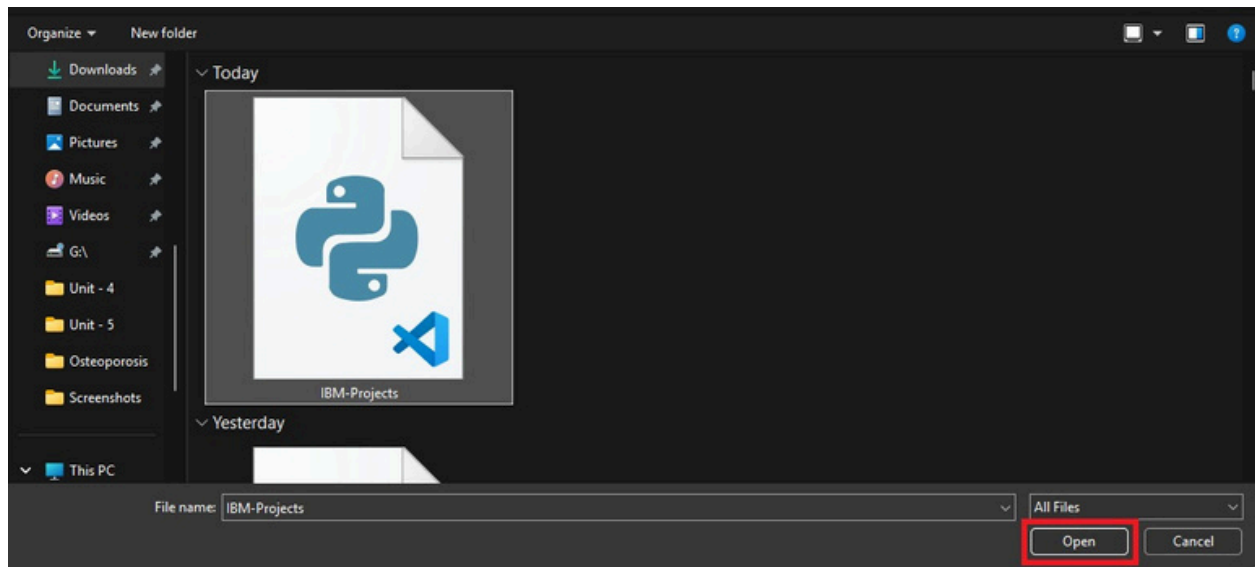
“Upload files” to upload your files.



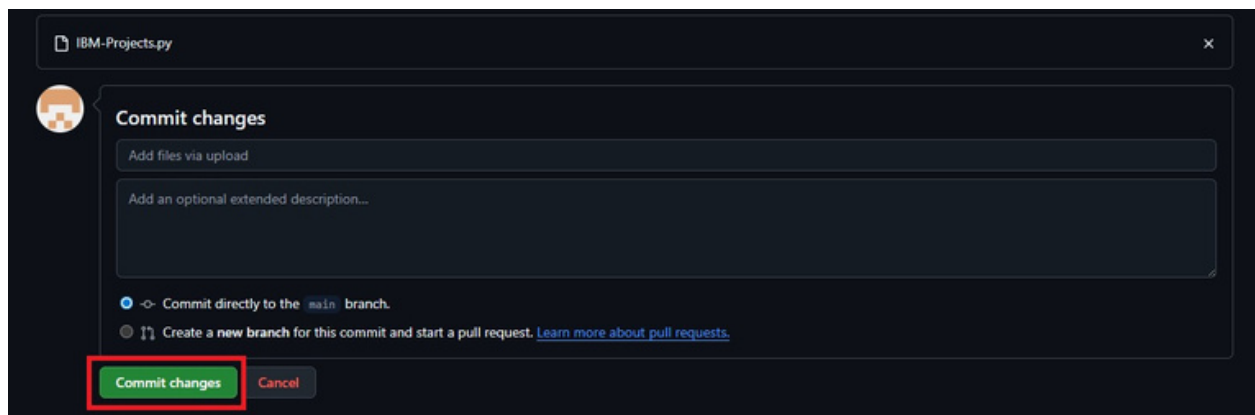
- Click on “choose your files”.



- Choose your project file and click on “Open”.



- After your file has Uploaded Click on “Commit changes”.



CONCLUSION :

The Sustainable Smart City Assistant demonstrates how AI, when applied thoughtfully, can make urban living more efficient, inclusive, and sustainable. By integrating IBM Granite LLM with tools such as Gradio, Google Colab, and GitHub, the project highlights the potential of Generative AI to assist in governance, environmental awareness, and citizen engagement. The assistant not only addresses immediate urban challenges like feedback management and eco-friendly practices but also lays a foundation for scalable solutions that can evolve with city needs.

Ultimately, this project proves that technology and sustainability can go hand in hand, offering students a meaningful learning experience while contributing to society's collective goal of building smarter, greener cities.