SUSTAINABLE SMART CITY ASSISTANT USING IBM GRANITE LLM

GenerativeAIwithIBM

TEAM MEMBERS::

1. Rajika V
2. Rajitha V
3. Adeline Abiah V

INTRODUCTION::

The Sustainable Smart City Assistant project is designed to address the pressing need forefficientcitygovernance,sustainabilitypractices,andcitizenengagement.By leveraging IBM Granite LLM (Large Language Model) available on Hugging Face, this assistantactsasadigitalcompanionforcitiestomonitorhealthindicators,gather citizenfeedback,generateecofriendlytips,andprovidedocumentsummarization.Theuseofgenerati veAIensuresthatres ponsesarecontext-aware,accurate,and tailored to the dynamic needs of a smart urban environment.

Theprojectworkflowinvolvesfourmainstages:exploringtheNaanMudhalvan SmartInterzportalforresources,selectinganappropriateIBMGranitemodelfrom HuggingFace,deployingtheapplicationusingGoogleColabwithGPUacceleration,andfinall y,uploadingandversion-controllingtheprojectonGitHub.Eachstageis carefully structured to help students gain practical exposure to modern AI tools while simultaneouslycontributingtothebroadervisionofsustainabilityinurbanareas.

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 assistanttotrackurbanhealthmetricsthroughaCityHealthDashboard,whilecitizenscanact ivelyengagebysharingtheirconcernsandreceivingusefulecofriendlylifestylesuggestions.Moreover, documentsummarizationcapabilitiessimplifydec ision-makingbycondensinglargereportsintoconcise,actionable insights.

From an academic perspective, this project is highly relevant as it combines multiple domains—ArtificialIntelligence,SustainableDevelopment,CloudComputing,and OpenSourceCollaboration.Studentsnotonlylearntobuildan AI-poweredsystem butalsogainexperienceinversioncontrol(GitHub),cloud-basedexecution(Colab), and ethical AI usage (sustainability-focused applications).

Inessence,theSustainableSmartCityAssistantreflectshowcutting-edgeAI technologieslikeIBMGraniteLLMcanbeharnessedtodesignintelligenturban solutions. It bridges the gap between theoretical knowledge and practical innovation, empoweringstudentstocontributetoagreener,smarter,andmoresustainablefuture.

PROjECTDEsCRIPTION::

SustainableSmartCityAssistantusestheGranitemodelfromHuggingFacetohelp 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. GradioFrameworkKnowledge:GradioDocumentation
2. IBMGraniteModels(HuggingFace):IBMGranitemodels
3. PythonProgrammingProficiency:PythonDocumentation
4. VersionControlwithGit:GitDocumentation
5. GoogleCollab'sT4GPUKnowledge:Googlecollab

PROjECTWORkFLOW:

Activity-1: Exploring Naan Mudhalavan Smart Interz Portal.

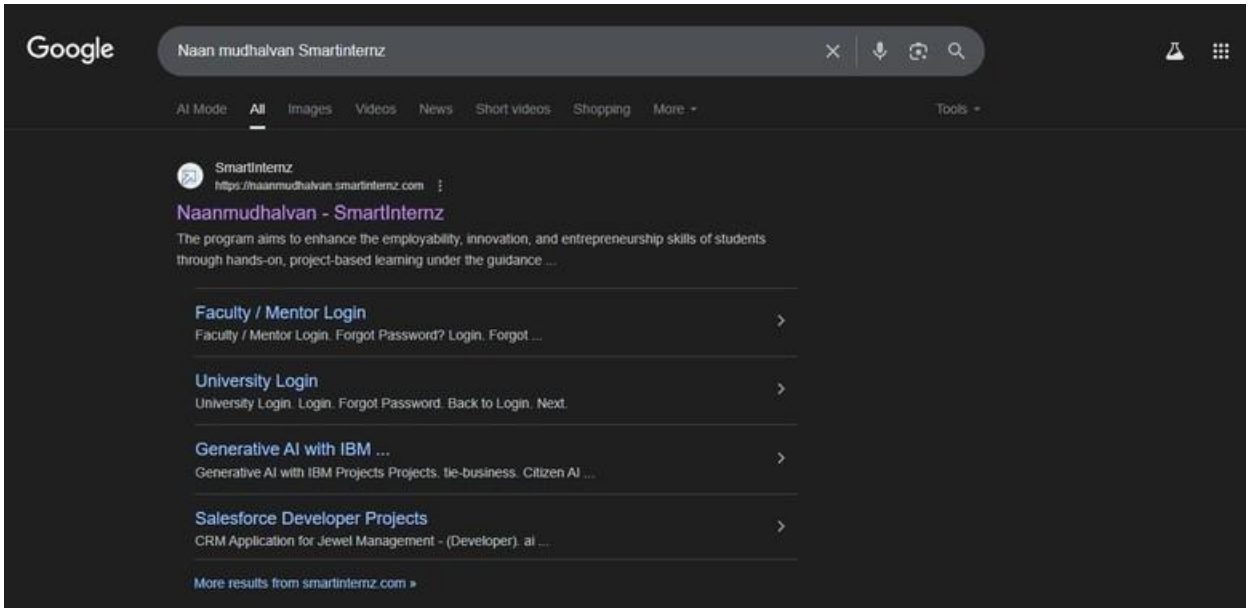
Activity-2: Choosing a IBM Granite Model From Hugging Face.

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

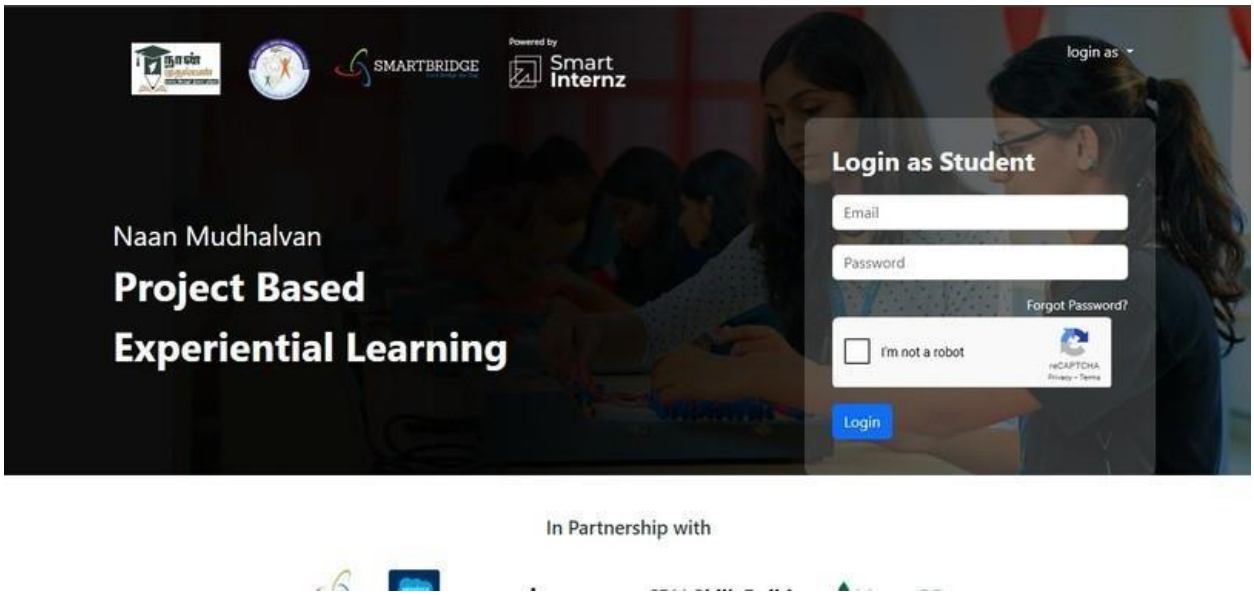
Upload your Project in Github.

Activity-1:ExploringNaanMudhalavanSmartInterzPortal.

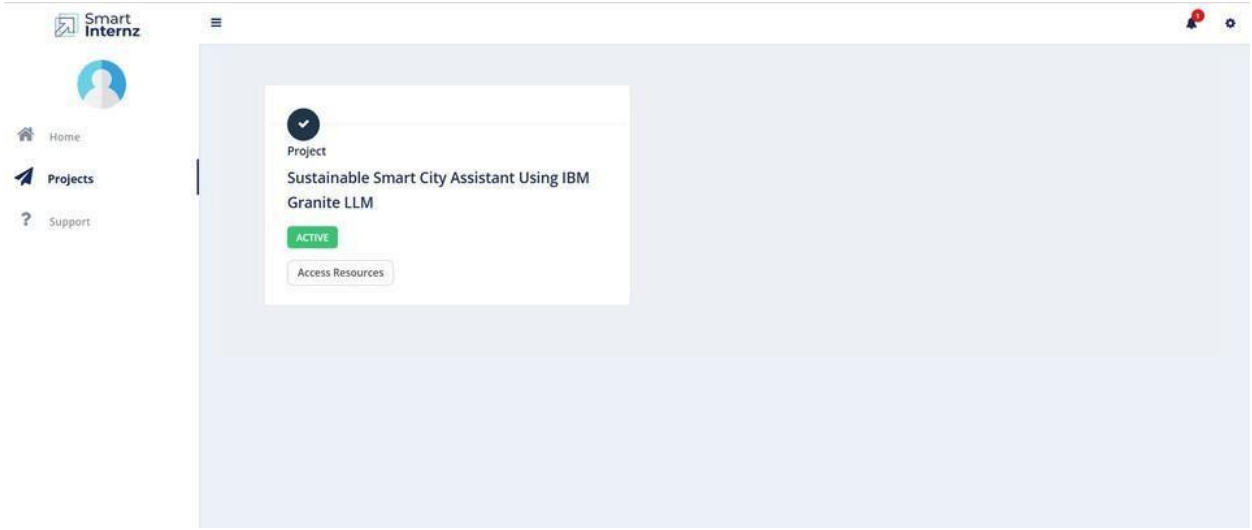
- Searchfor"NaanMudhalavanSmartInterz"PortalinanyBrowser.



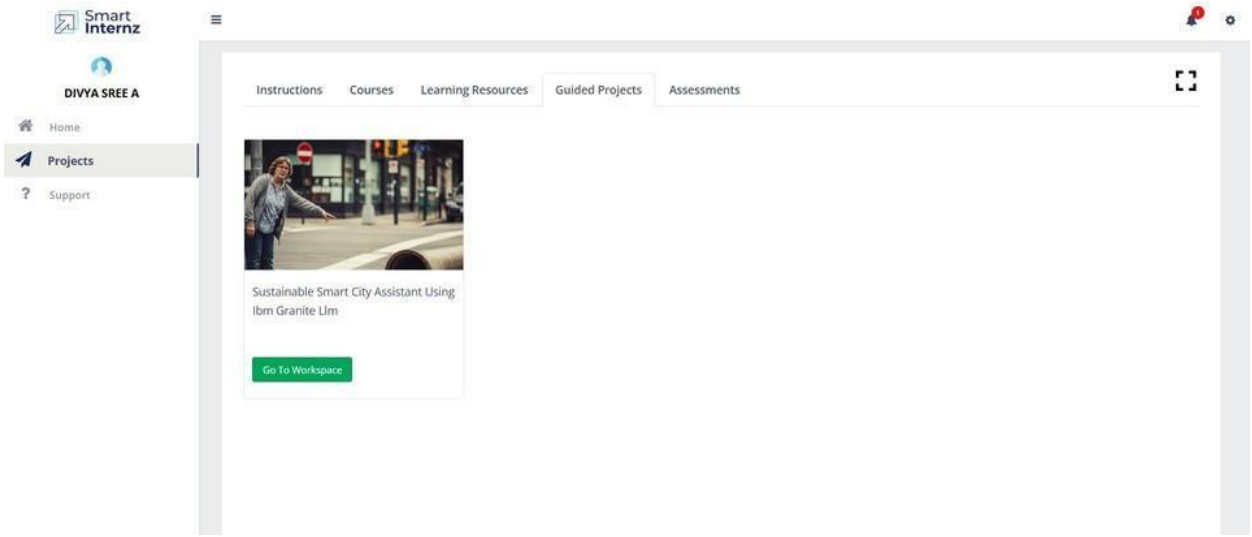
- Then Click on the first link. (Naanmudhalvan Smartinternz) Then login with your details.



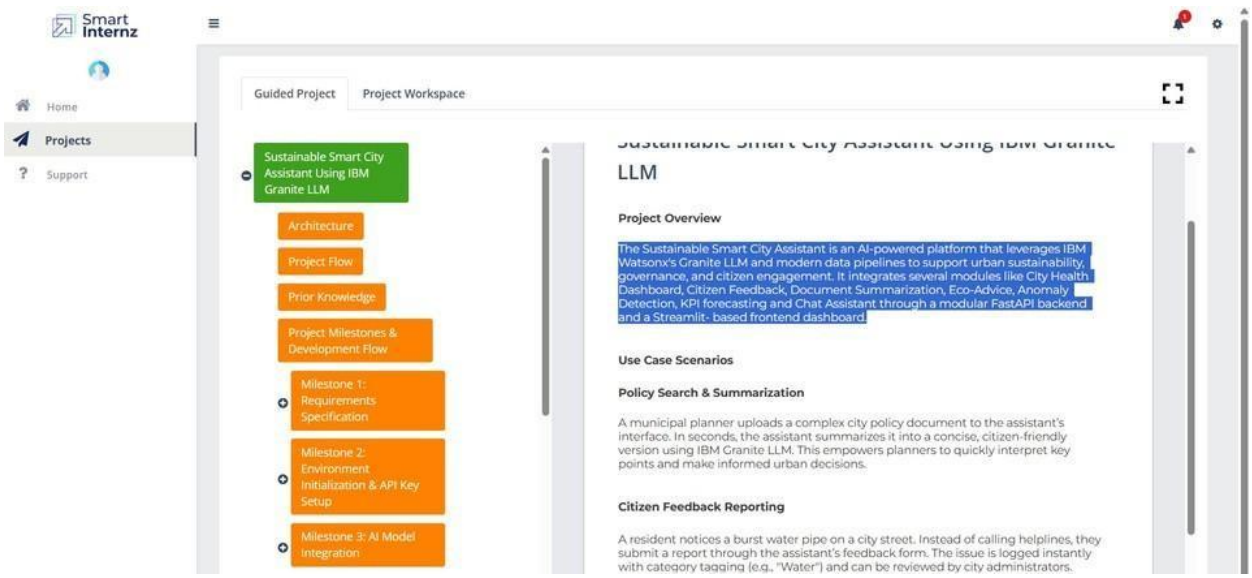
- Then you will be redirected to your account then click on “Projects” Section. There you can see which project you have enrolled in here it is “Health AI”.



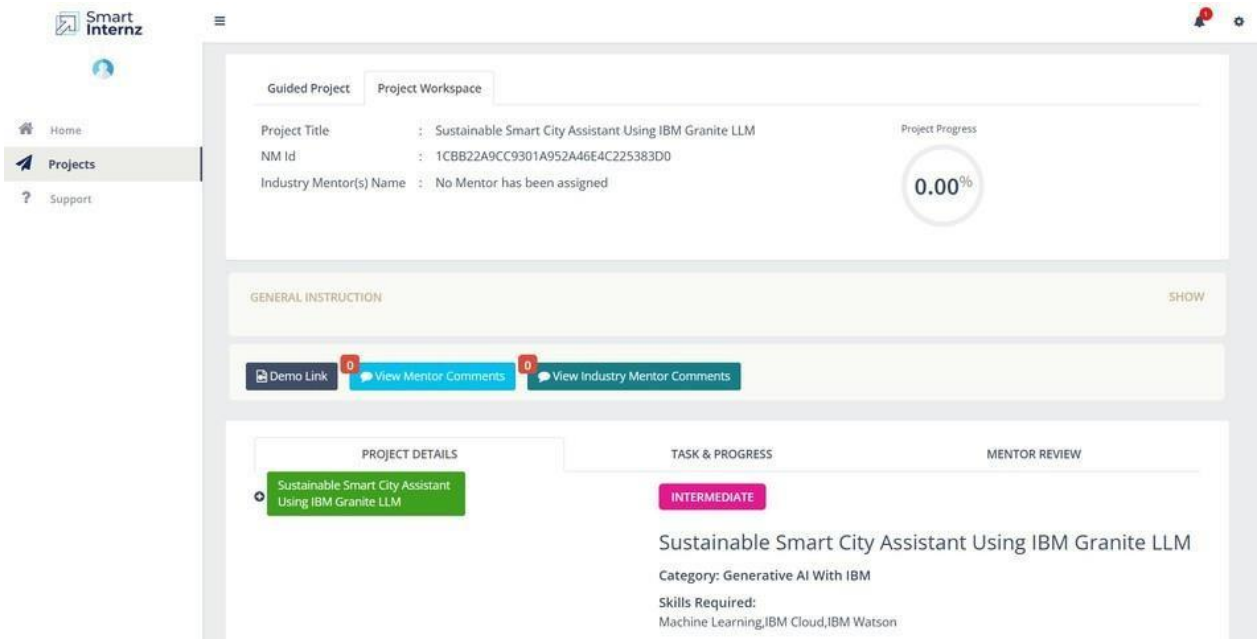
- Thenclickon“AccessResources”andgotothe“GuidedProject”Section.



- Clickonthe“Gotoworkspace”section.Thenyoucanfindthedetailed explanationofGenerativeAIProjectusingIBMWatsonxAPIkey.



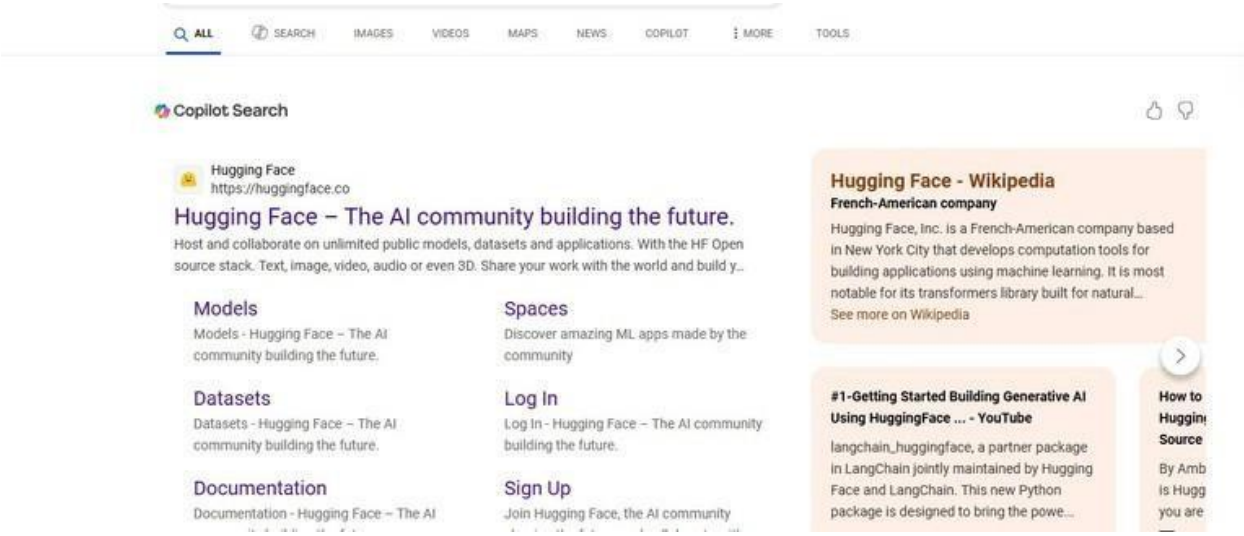
- Click on “Project Workspace”, there you can find your project progress and Place to upload “Demo link”.



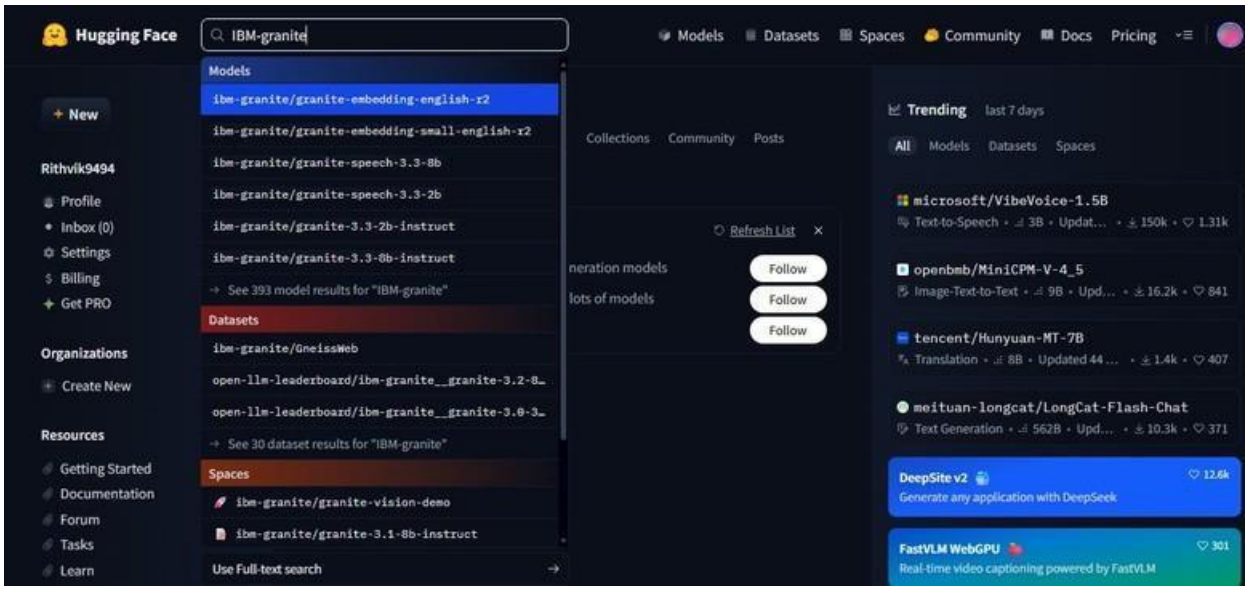
- Now we have gone through portal understanding, now lets find a IBM granite model from hugging face to integrate in our project.

Activity-2: ChooseaIBM GranitemodelFrom HuggingFace.

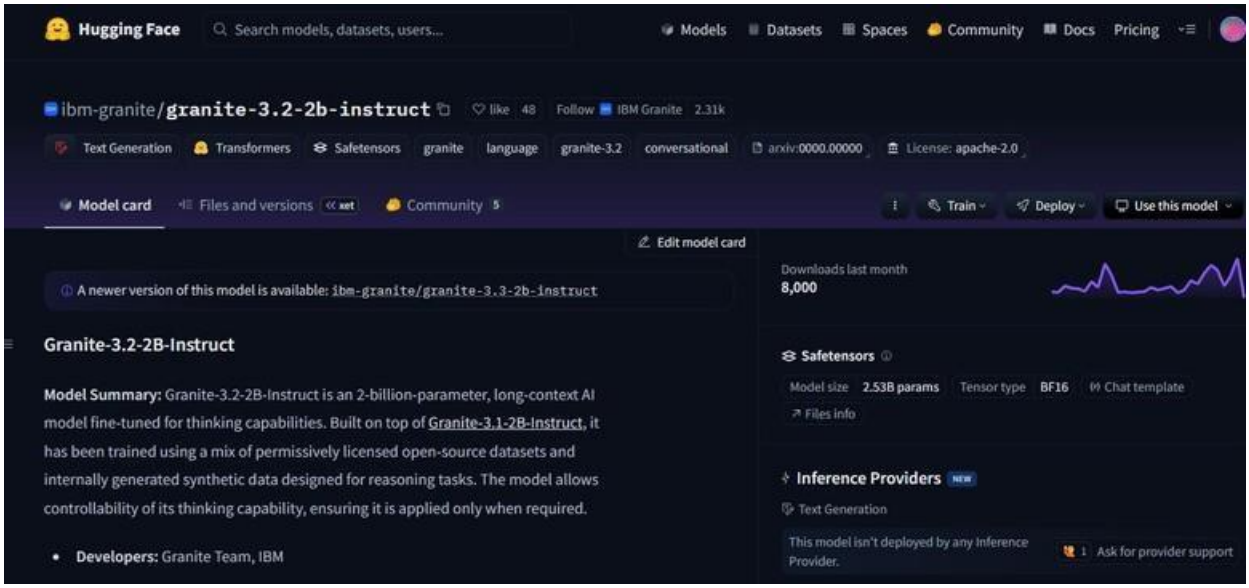
- Searchfor“Huggingface”inanybrowser.



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



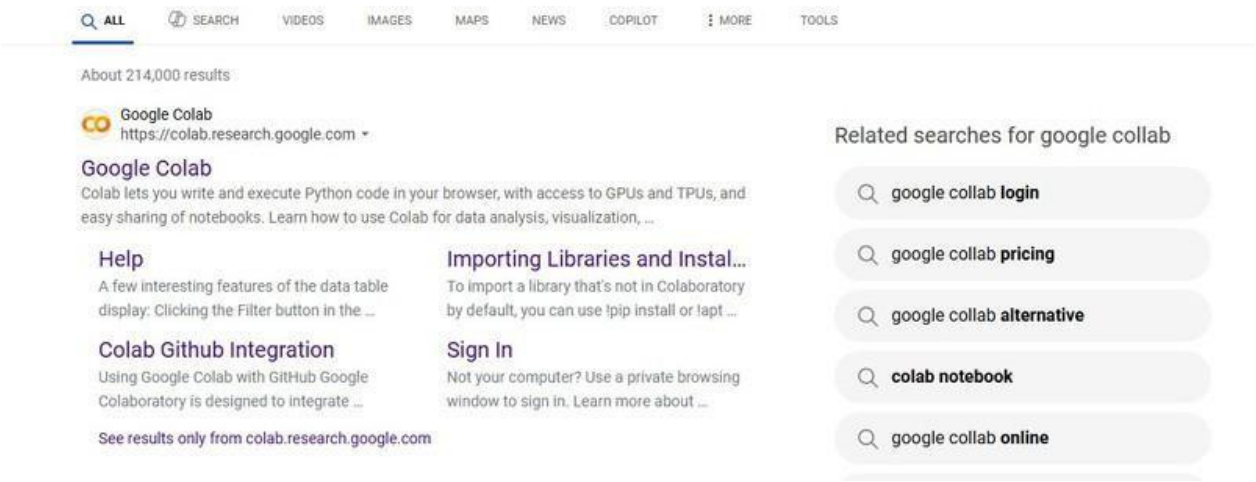
- Here for this project we are using “granite-3.2-2b-instruct” which is compatible fast and light weight.



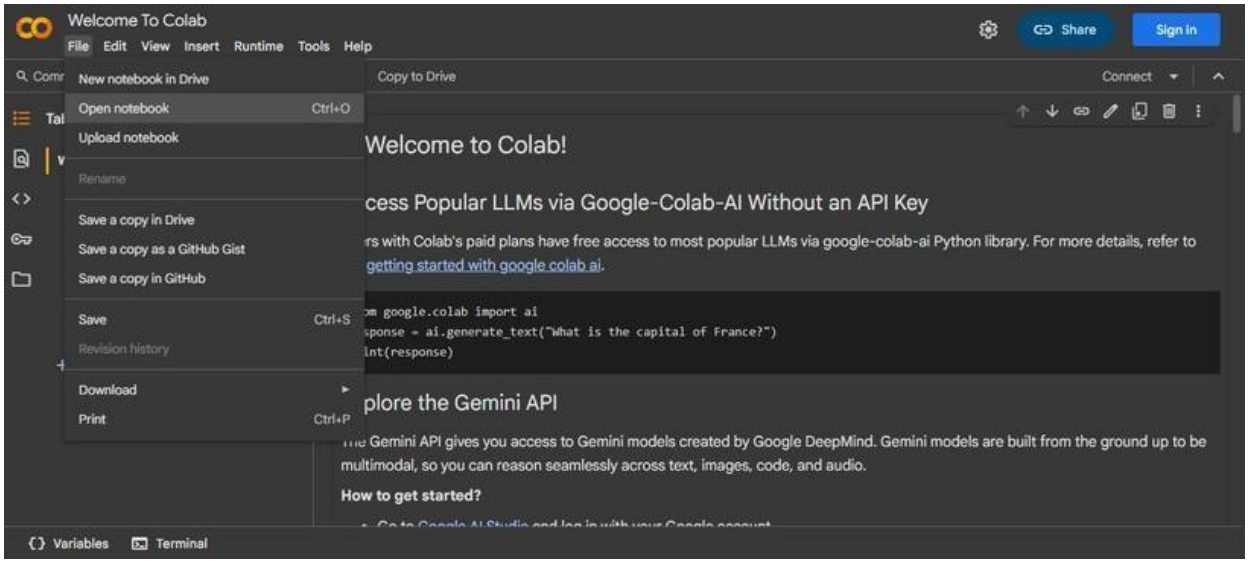
- Nowwewillstartbuildingourprojectin Googlecollab.

Activity-3:RunningApplicationinGoogleCollab.

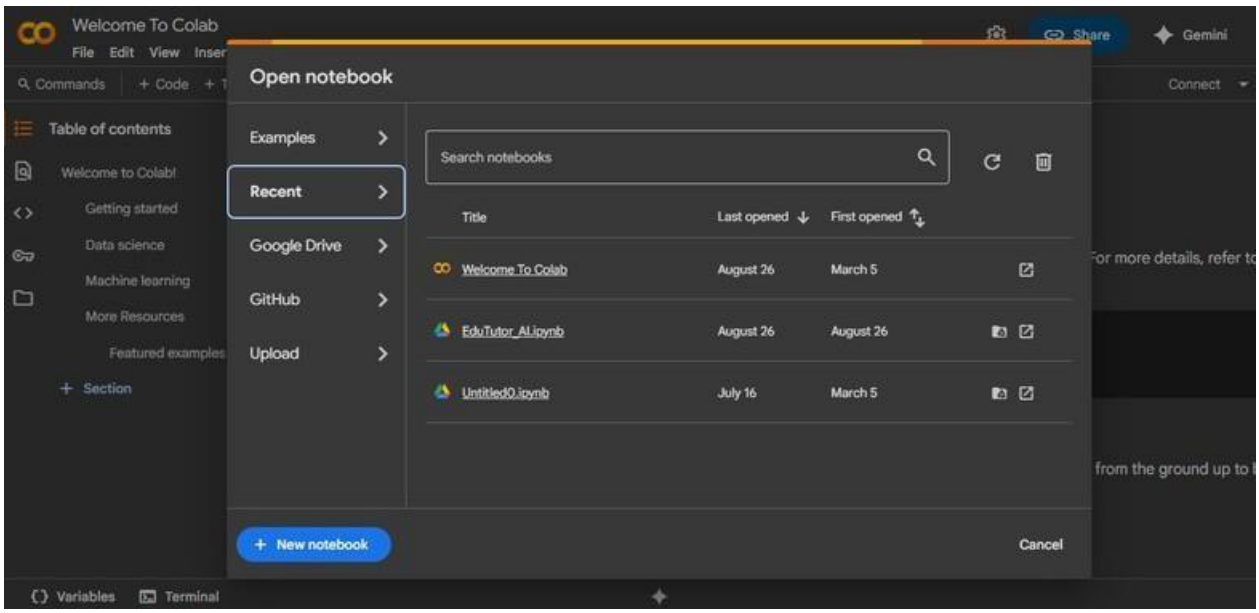
- Searchfor“Googlecollab”inanybrowser.



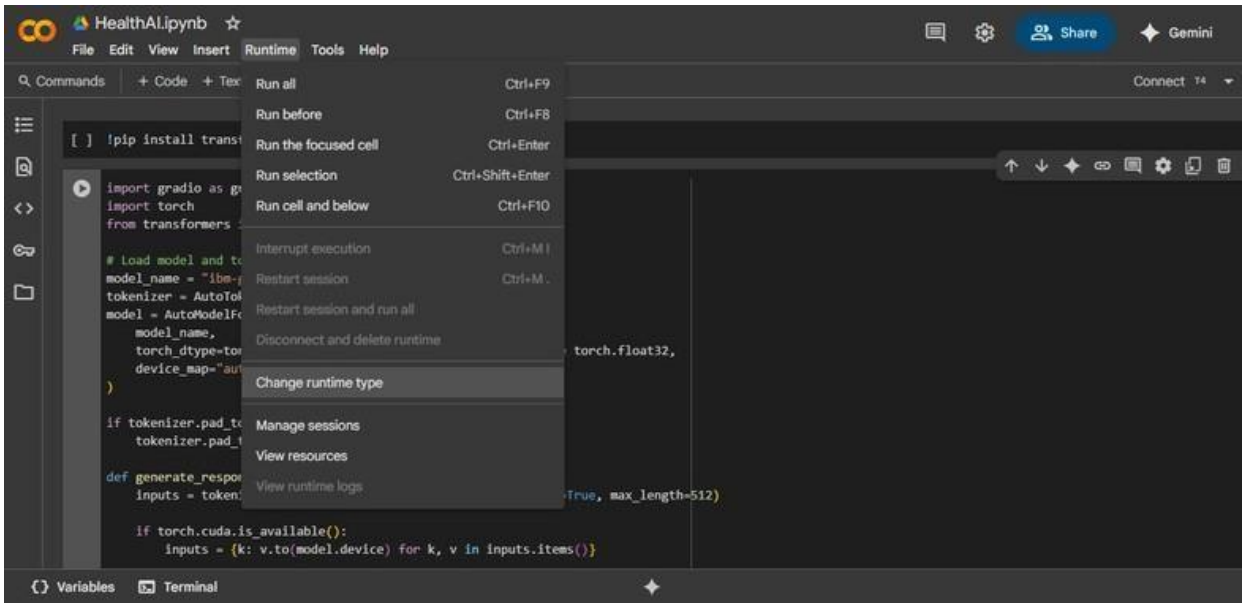
- Click on the first link (Google Colab), then click on “Files” and then “Open Notebook”.



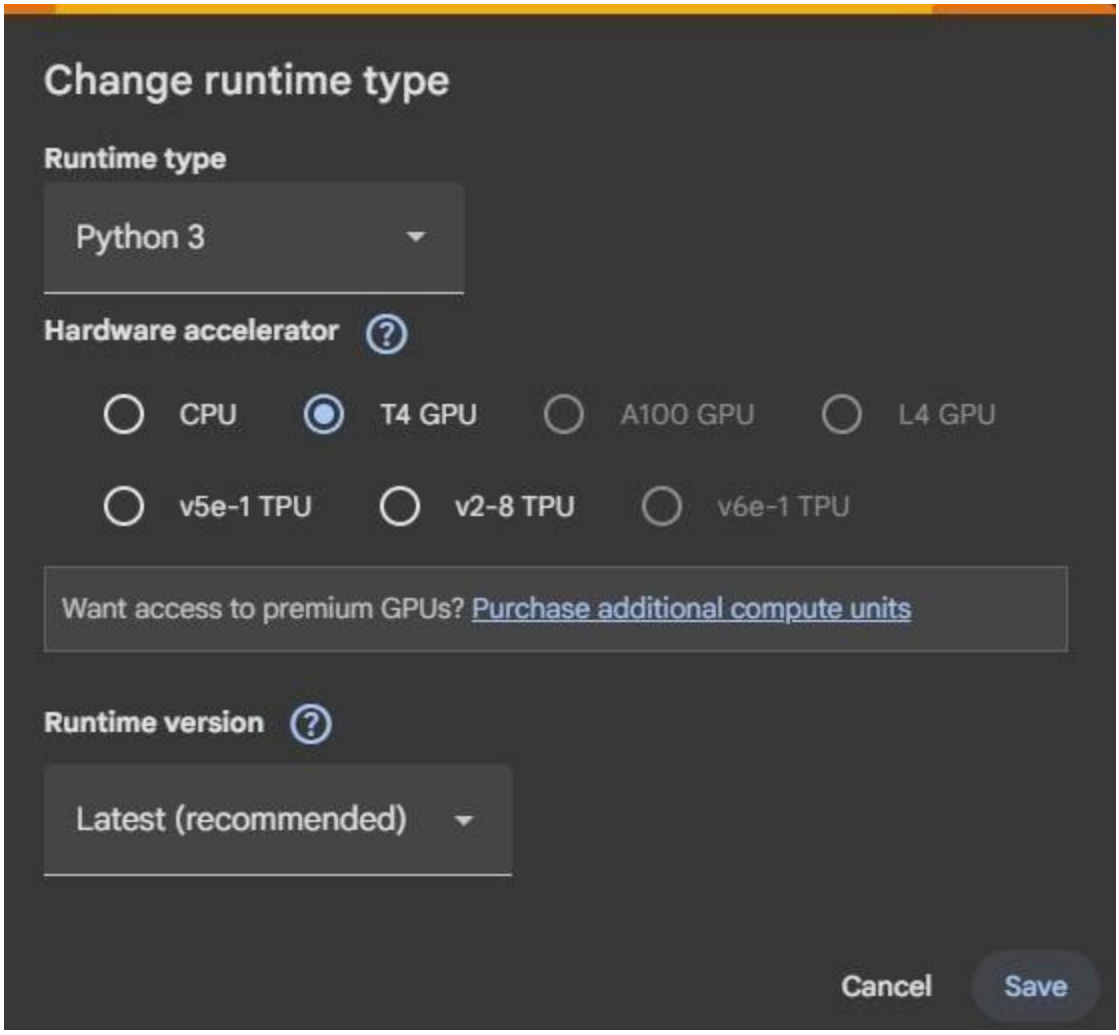
- Clickon“NewNotebook”



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



- Choose“T4GPU”andclickon“Save”

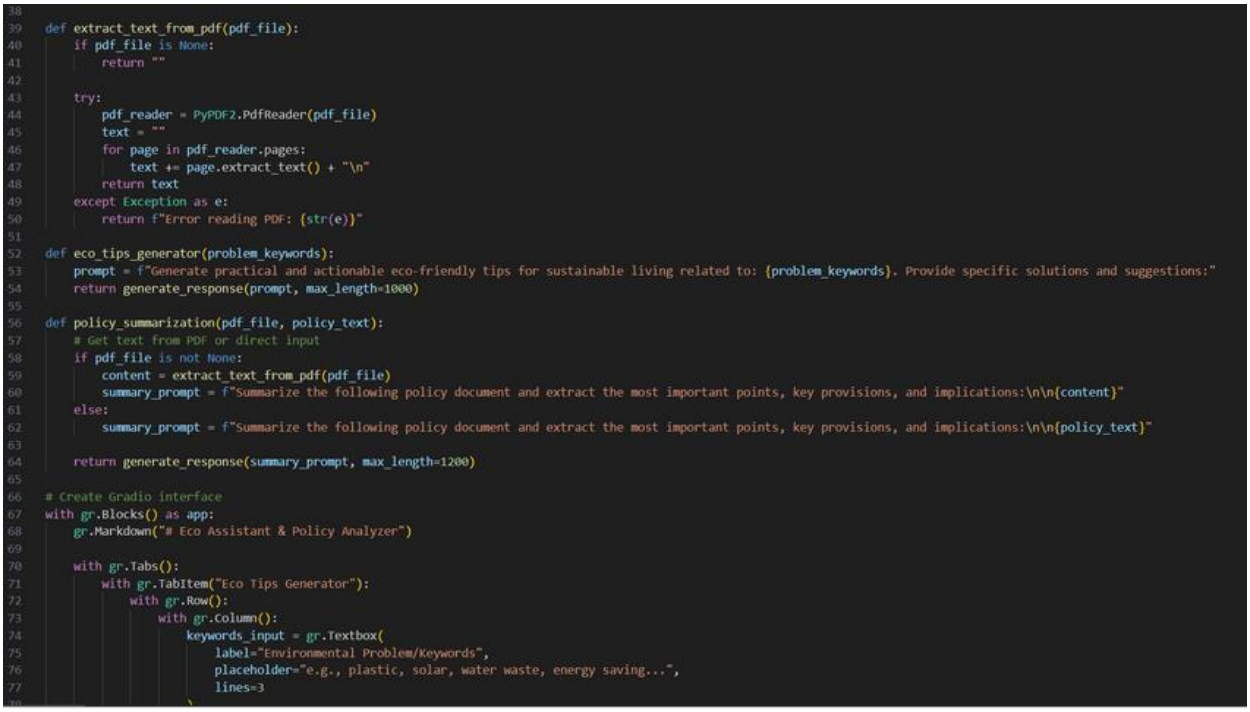


- Thenrunthiscommandinthefirstcell“!pipinstalltransformerstorch gradioPyPDF2-q”.Toinstalltherequiredlibrariestorunourapplication.

```
[ ] !pip install transformers torch gradio PyPDF2 -q
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 232.6/232.6 kB 12.3 MB/s eta 0:00:00
```

- Thenruntherestofthecodeinthenextcell.

```python

import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
import PyPDF2
import io

# 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()}

    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
            temperature=0.7,
            do_sample=True,
            pad_token_id=tokenizer.eos_token_id
        )

    response = tokenizer.decode(outputs[0], skip_special_tokens=True)
    response = response.replace(prompt, "").strip()
    return response
```



```python
def extract_text_from_pdf(pdf_file):
    if pdf_file is None:
        return ""

    try:
        pdf_reader = PyPDF2.PdfReader(pdf_file)
        text = ""
        for page in pdf_reader.pages:
            text += page.extract_text() + "\n"
        return text
    except Exception as e:
        return f"Error reading PDF: {str(e)}"

def eco_tips_generator(problem_keywords):
    prompt = f"Generate practical and actionable eco-friendly tips for sustainable living related to: {problem_keywords}. Provide specific solutions and suggestions:"
    return generate_response(prompt, max_length=1000)

def policy_summarization(pdf_file, policy_text):
    # Get text from PDF or direct input
    if pdf_file is not None:
        content = extract_text_from_pdf(pdf_file)
        summary_prompt = f"Summarize the following policy document and extract the most important points, key provisions, and implications:\n\n{content}"
    else:
        summary_prompt = f"Summarize the following policy document and extract the most important points, key provisions, and implications:\n\n{policy_text}"

    return generate_response(summary_prompt, max_length=1200)

# Create Gradio interface
with gr.Blocks() as app:
    gr.Markdown("# Eco Assistant & Policy Analyzer")

    with gr.Tabs():
        with gr.TabItem("Eco Tips Generator"):
            with gr.Row():
                with gr.Column():
                    keywords_input = gr.Textbox(
                        label="Environmental Problem/Keywords",
                        placeholder="e.g., plastic, solar, water waste, energy saving...",
                        lines=3
```

```python
                with gr.Column():
                    keywords_input = gr.Textbox(
                        label="Environmental Problem/Keywords",
                        placeholder="e.g., plastic, solar, water waste, energy saving...",
                        lines=3
                    )
                    generate_tips_btn = gr.Button("Generate Eco Tips")

                with gr.Column():
                    tips_output = gr.Textbox(label="Sustainable Living Tips", lines=15)

            generate_tips_btn.click(eco_tips_generator, inputs=keywords_input, outputs=tips_output)

        with gr.TabItem("Policy Summarization"):
            with gr.Row():
                with gr.Column():
                    pdf_upload = gr.File(label="Upload Policy PDF", file_types=[".pdf"])
                    policy_text_input = gr.Textbox(
                        label="Or paste policy text here",
                        placeholder="Paste policy document text...",
                        lines=5
                    )
                    summarize_btn = gr.Button("Summarize Policy")

                with gr.Column():
                    summary_output = gr.Textbox(label="Policy Summary & Key Points", lines=20)

            summarize_btn.click(policy_summarization, inputs=[pdf_upload, policy_text_input], outputs=summary_output)

app.launch(share=True)
```

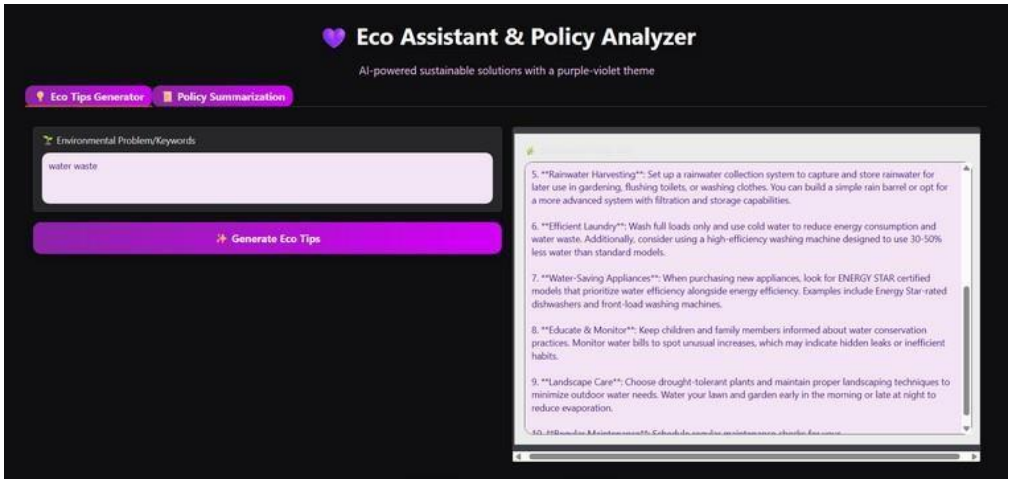
- Youcanfindthecodehereinthislink:SustainableSmartCityAssistant

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
```

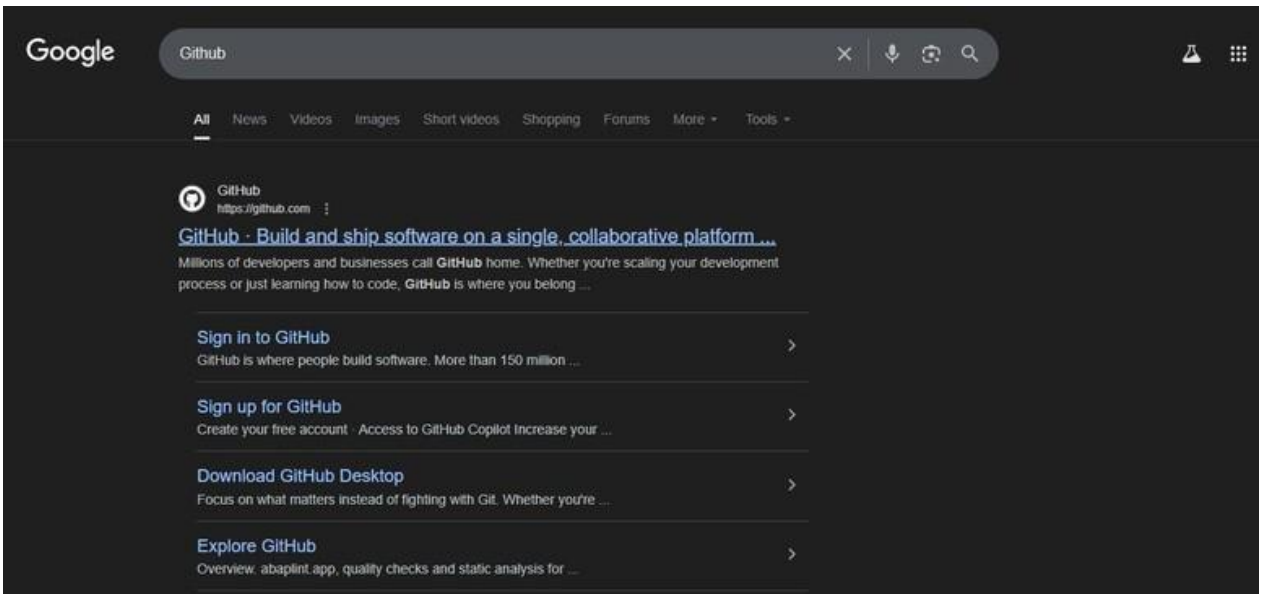
- ClickontheURltoopentheGradioApplicationclickonthelink



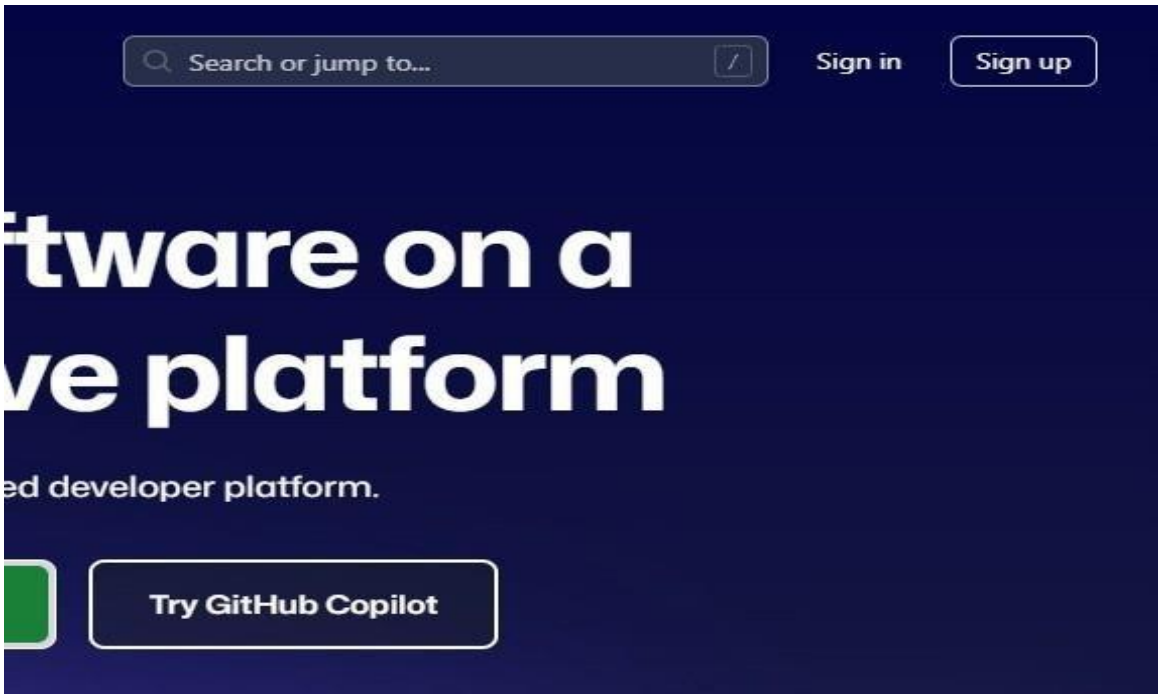
- YoucanViewtheApplicationistherunningintheothertab

Activity-4:UploadYourProjectinGitHub.

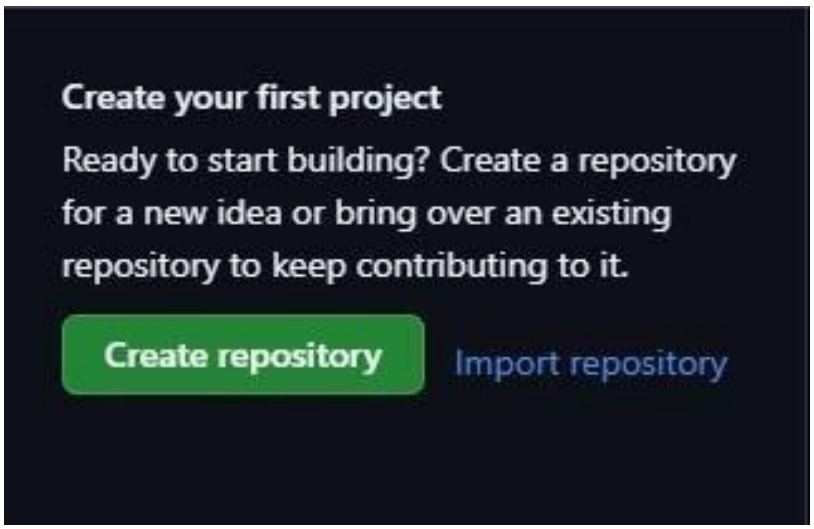
- Searchfor“GitHub”inanybrowser,thenclickonthefirstlink(GitHub).



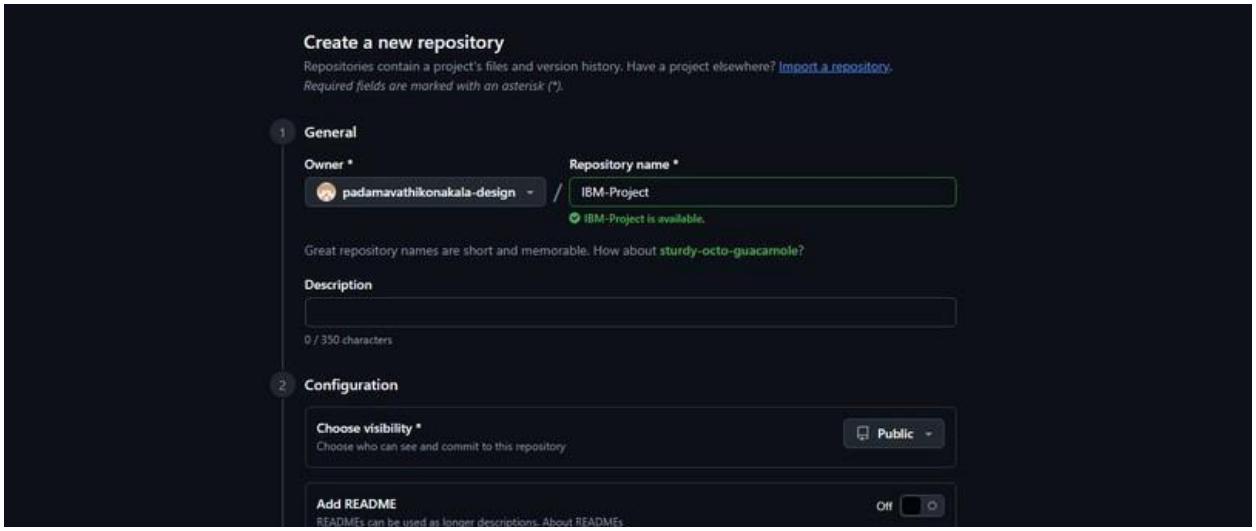
- Thenclickon“Signup”andcreateyourownaccountinGitHub.Ifyou already have an account click on “Sign in”



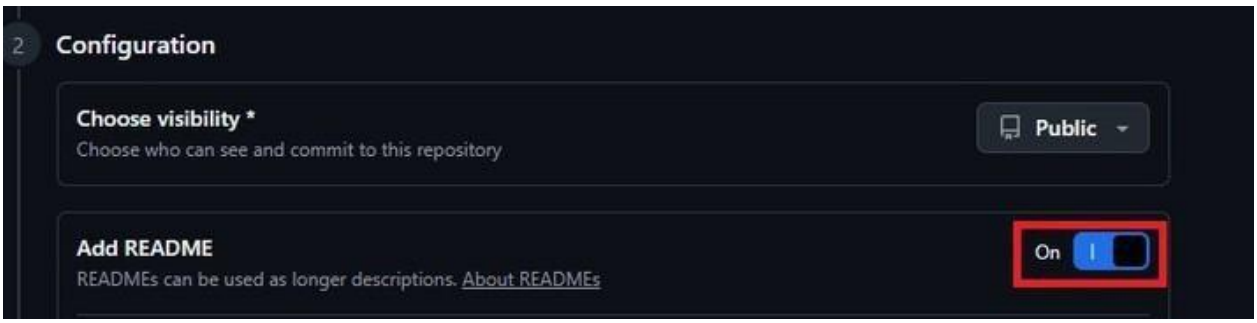
- Clickon“Createrepository”.



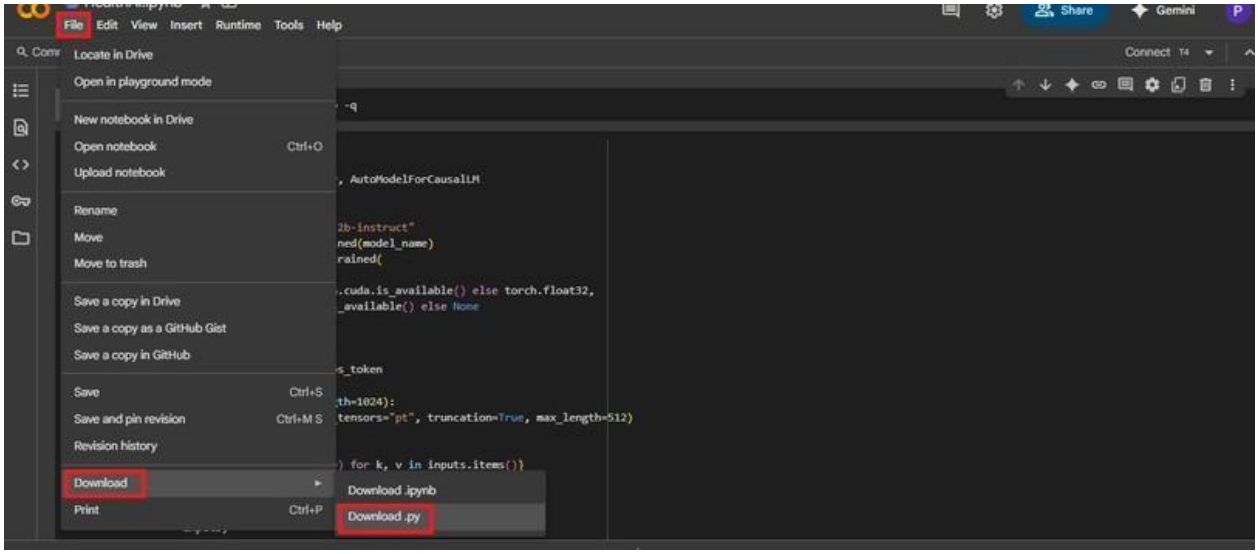
- In “General” Name your repo. (Here I have given “IBM-Project” as my repo name and it is available)



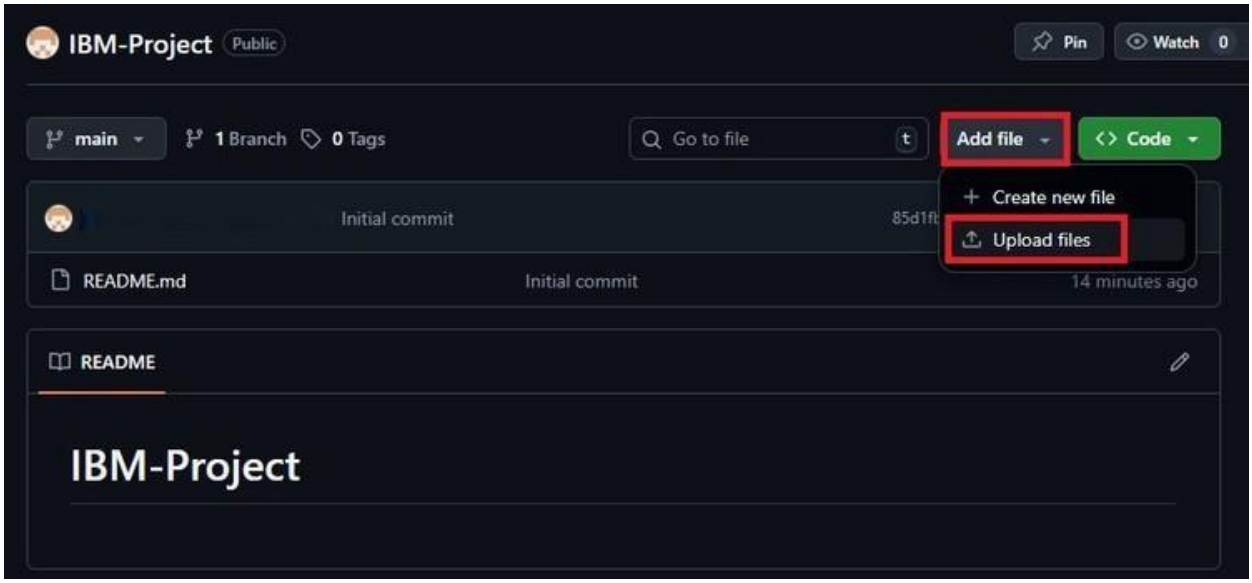
- In“Configurations”TurnOn“Addreadme”fileOption.



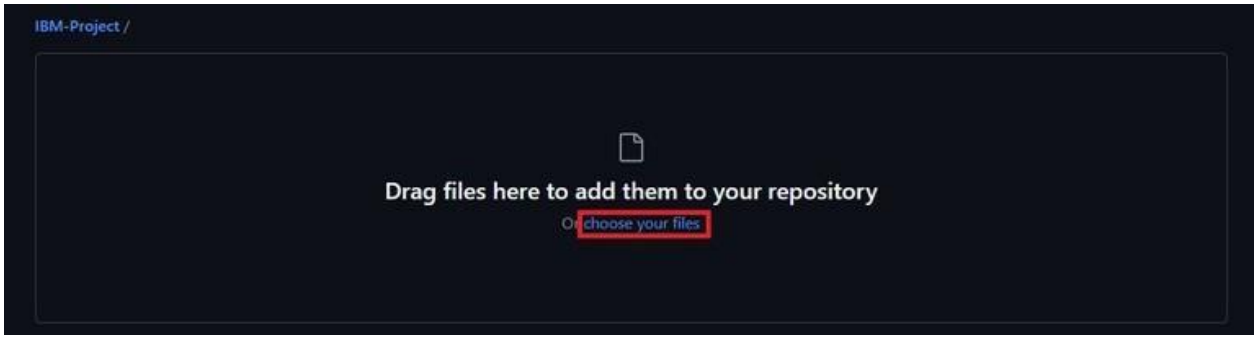
- Now Download your code from Google collab by Clicking on “File”, then Goto “Download” then download as “.py”.



- Thenyourrepositoryiscreated,thenClickon“Addfile”Option.ThenClick “Uploadfiles”touploadyourfiles.

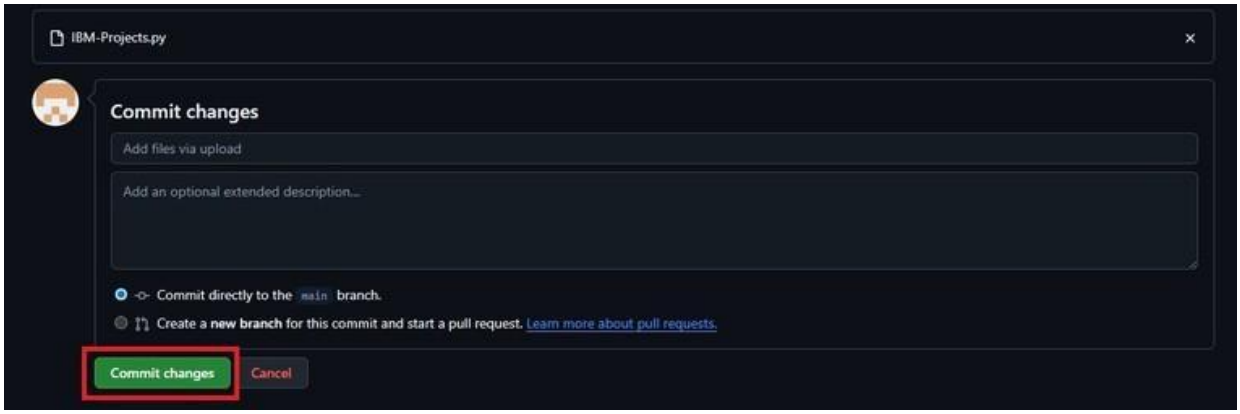


- Clickon“chooseyourfiles”.



- Chooseyourprojectfileandclickon“Open”.

- AfteryourfilehasUploadedClickon“Commitchanges”.



CONCLUSION::

TheSustainableSmartCityAssistantdemonstrateshowAI,whenapplied thoughtfully,canmakeurbanlivingmoreefficient,inclusive,andsustainable.By integrating IBM Granite LLM with tools such as Gradio, Google Colab, and GitHub, the projecthighlightsthepotentialofGenerativeAItoassistingovernance,environmental awareness, and citizen engagement. The assistant not only addresses immediateurbanchallengeslikefeedbackmanagementandeco-friendlypractices but also lays a foundation for scalable solutions that can evolve with city needs.

Ultimately,thisprojectprovesthattechnologyandsustainabilitycangohandin hand,offeringstudentsameaningfullearningexperiencewhilecontributingto society's collective goal of building smarter, greener cities.