****

**INDEX**

**Contents**

* Project overview
* Abstract

* Pre-requisites

* Project workflow

**Activity-1:** Exploring Naan Mudhalvan smart internz portal

**Activity-2:** Choose a IBM Granite Model From Hugging Face

**Activity-3**: Running Application in Google Collab

**Activity-4**: Upload Your Project in GitHub

* Future Enhancement



**PROJECT OVERVIEW**

The Software Development Lifecycle (SDLC) is the process of planning, creating, testing and developing software. Traditional SDLC often struggles with delays, rising costs, and human error. With the growth of Artificial Intelligence (AI), this issue can be minimized by automatic repetitive tasks and making software decisions.

SmartSDLC is a AI enhanced approach that integrates intelligent tools into every phase of development- from requirements gathering to maintenance. This project explores how SmartSDLC improves efficiency, reduce costs, and ensure better quality in software development.

**Abstract:**

This project is designed to make programming easier by helping users both generate code and analyze code. The system works in two simple ways: when a user gives a prompt, it generates the required code automatically; and when a user provides existing code, it analyzes the logic and gives explanations or suggestions. The goal of this project is to reduce the difficulty that beginners face while coding and to save time for experienced programmers. The frontend is developed using [your chosen frontend, e.g., HTML, CSS, JavaScript/React] to provide a clean and user-friendly interface, while the backend uses [your chosen backend, e.g., PHP/Java/Python with MySQL] to handle the processing and data management. By following the Software Development Life Cycle (SDLC) approach, the project was developed step by step—from problem identification to design, coding, testing, and deployment. The end result is a smart and reliable tool that helps learners understand programming better and assists developers in writing error-free code more quickly.

**PROJECT WORKFLOW**

**Activity-1**: Exploring Naan Mudhalvan smart internz portal

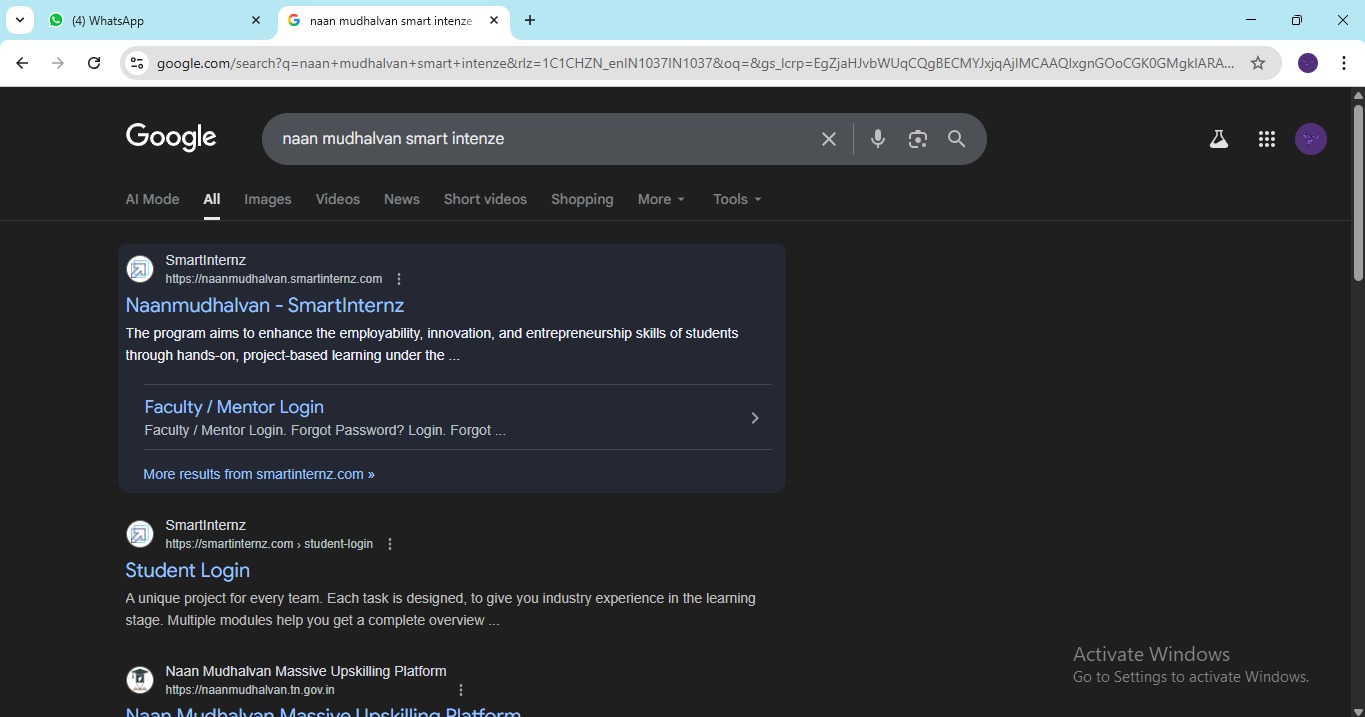
**Activity-2**: Choose a IBM Granite Model from Hugging Face

**Activity-3**: Running Application in Google Collab

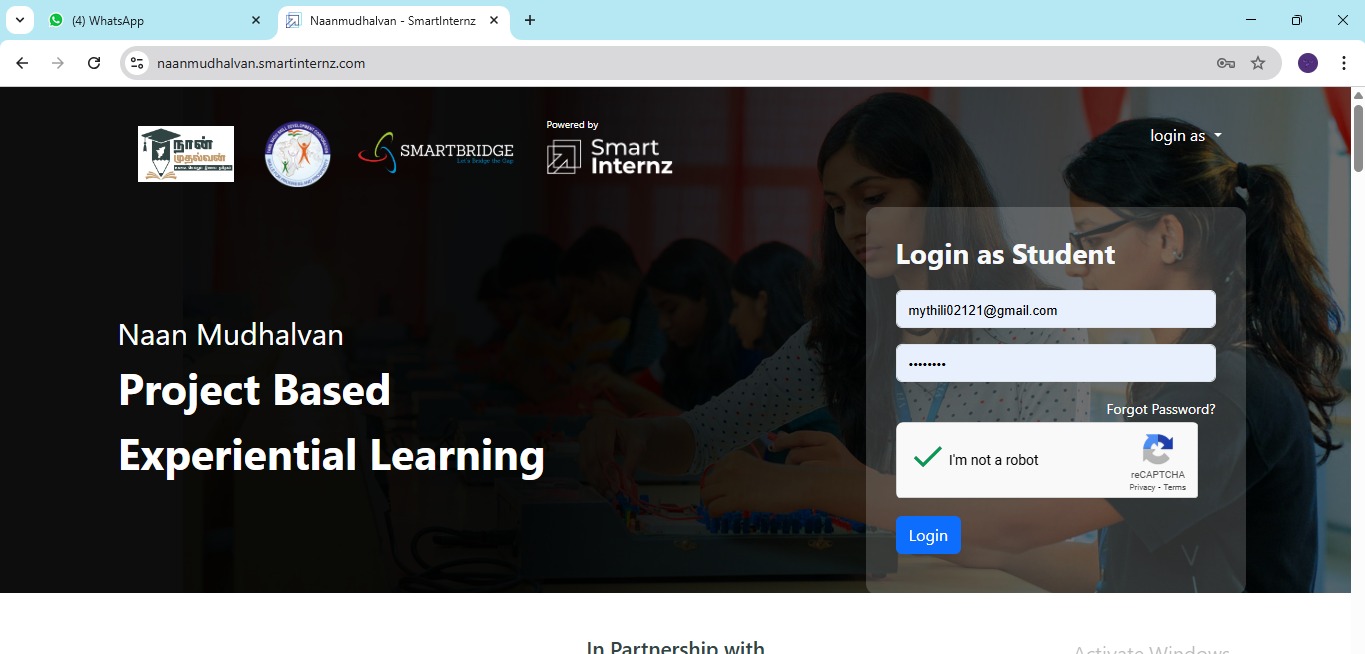
**Activity-4**: Upload Your Project in GitHub

Activity-1: Exploring Naan Mudhalavan Smart Interz Portal.

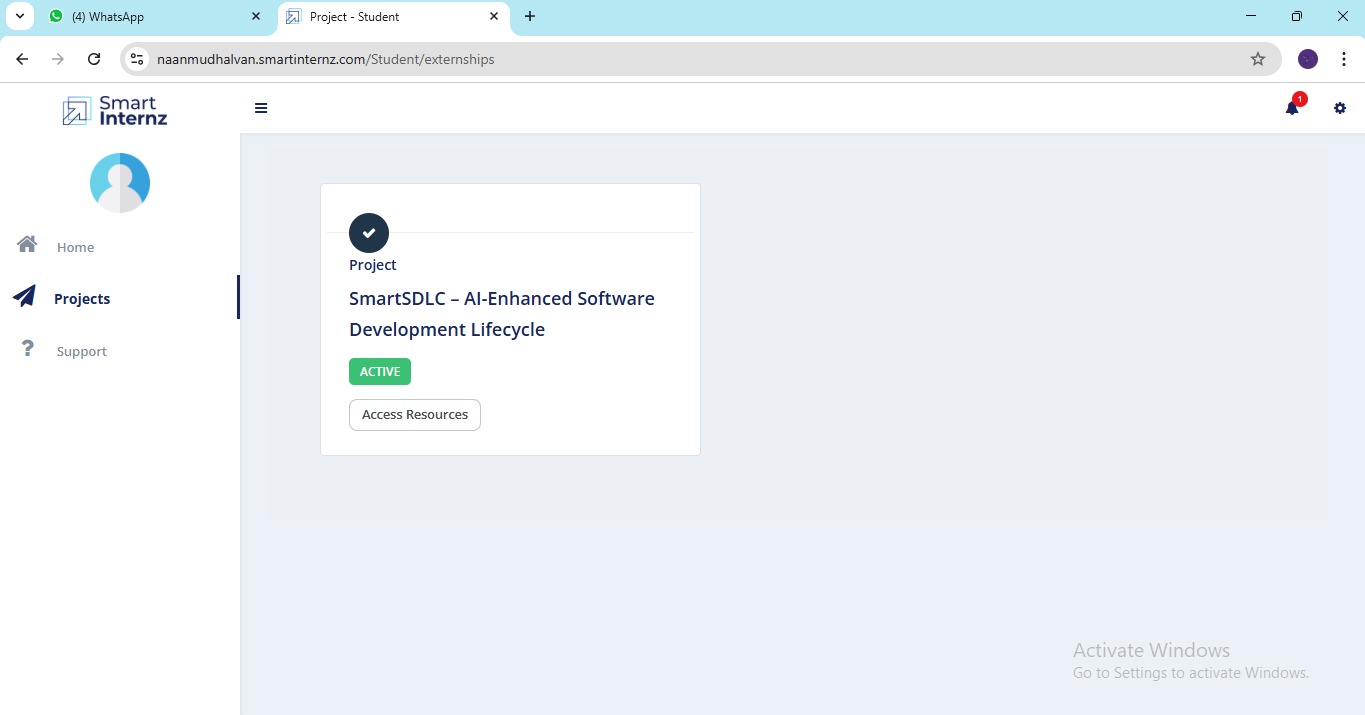
● Search for “Naan Mudhalavan Smart Interz” Portal in any Browser.



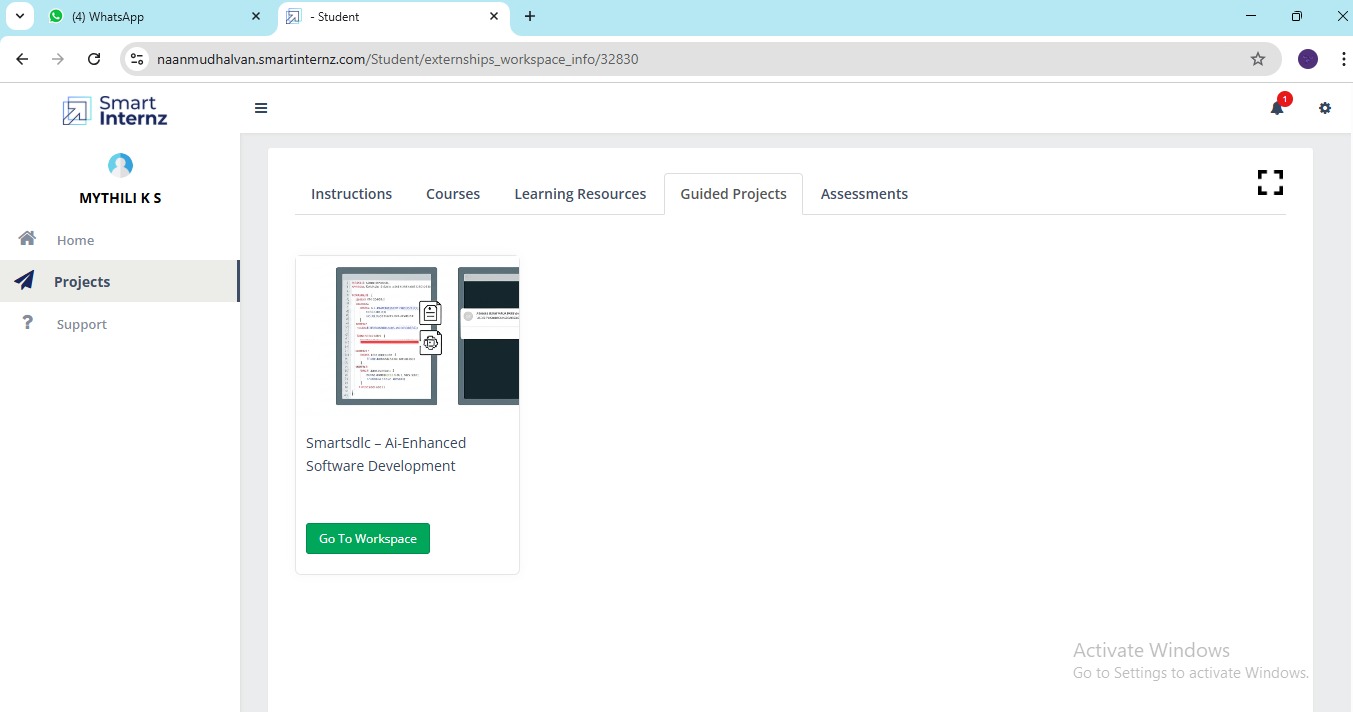
● Then Click on the first link. ([Naanmudhalvan Smartinternz](https://naanmudhalvan.smartinternz.com/)) Then login with your details.



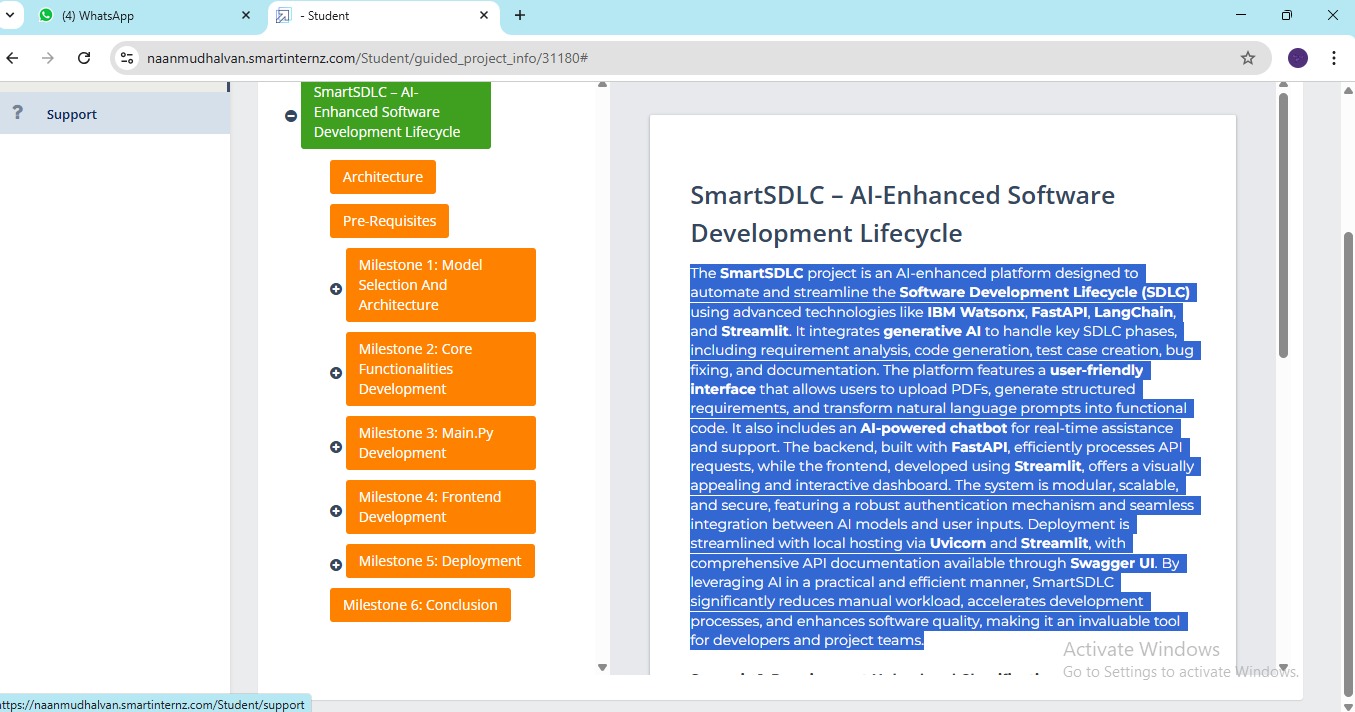
● Then you will be redirected to your account then click on “Projects” Section. There you can see which project you have enrolled in here it is “ SmartSDLC– AI-Enhanced Software Development Lifecycle”



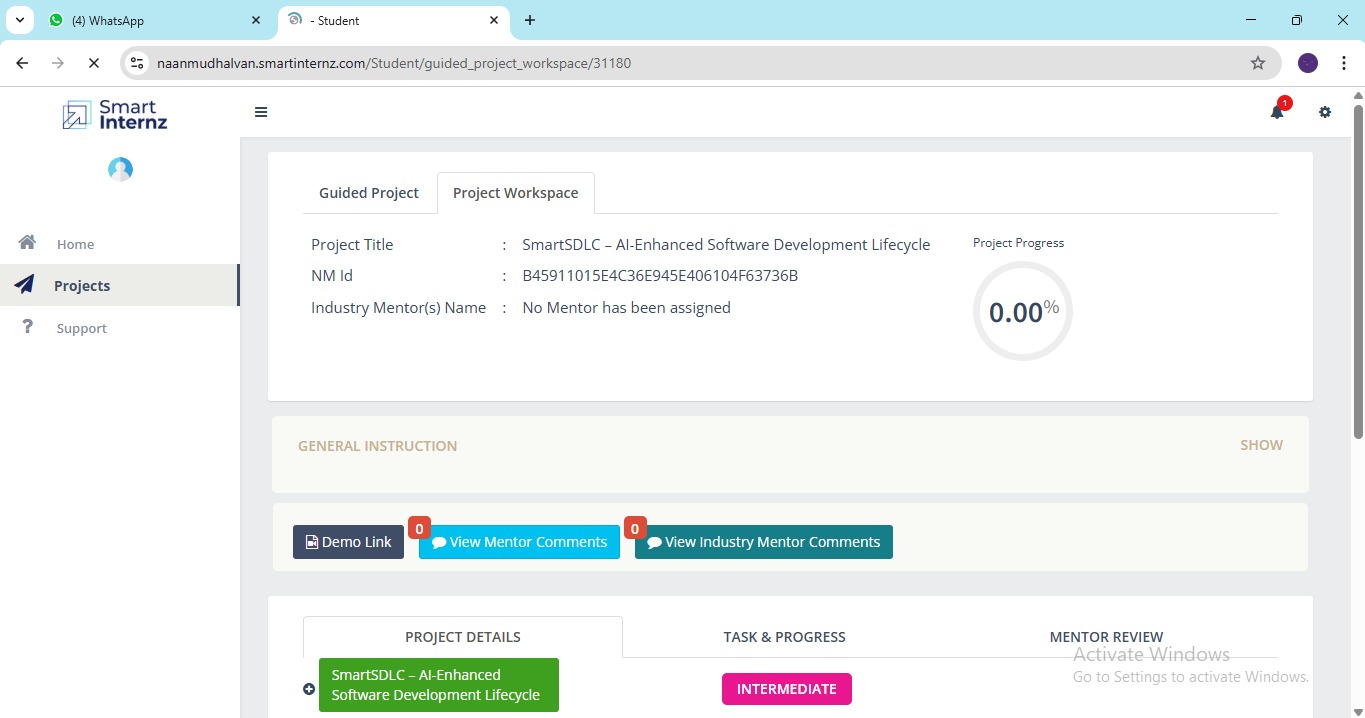
● Then click on “Access Resources” and go to the “Guided Project” Section



● Click on the “Go to workspace” section. Then you can find the detailed explanation of Generative AI Project using IBM WatsonX API key



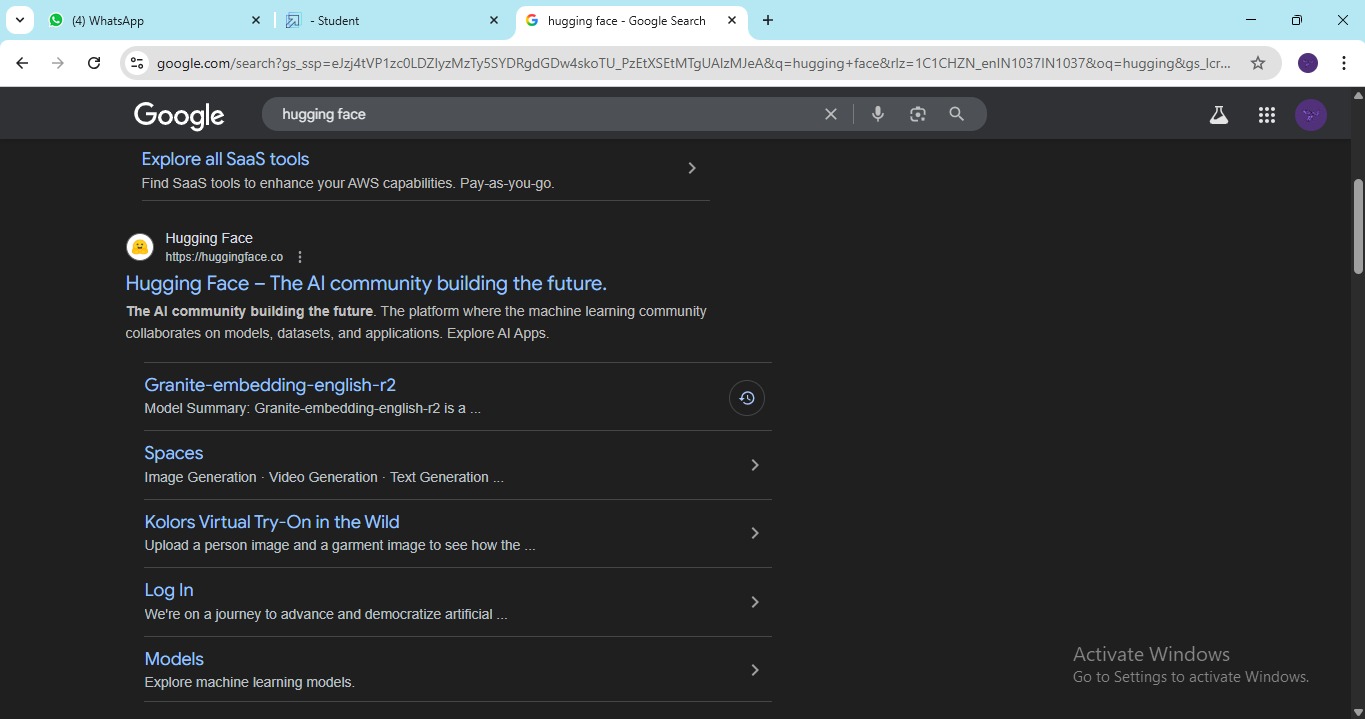
● Click on “Project Workspace”, there you can find your project progress and Place to upload “Demo link.



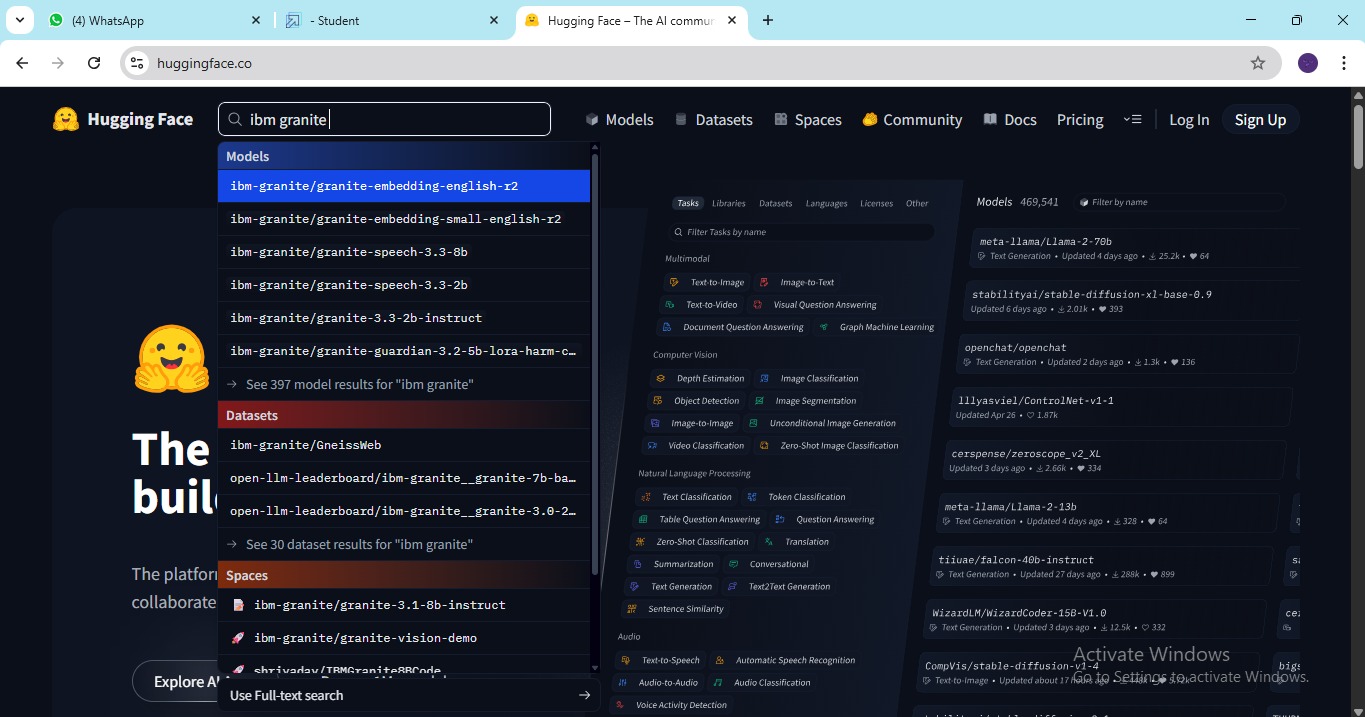
● Now we have gone through portal understanding, now lets find a IBM granite model from hugging face to integrate in our project.

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

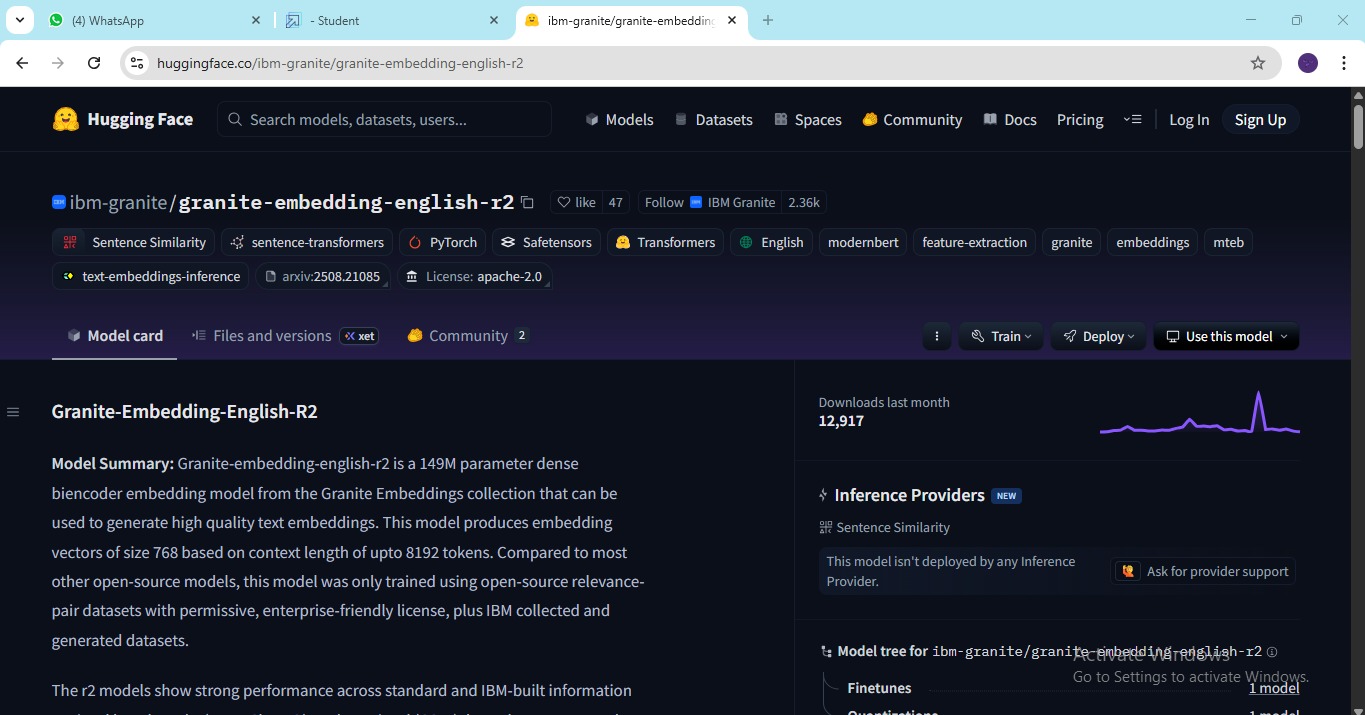
● Search for “Hugging face” in any browser.



● 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



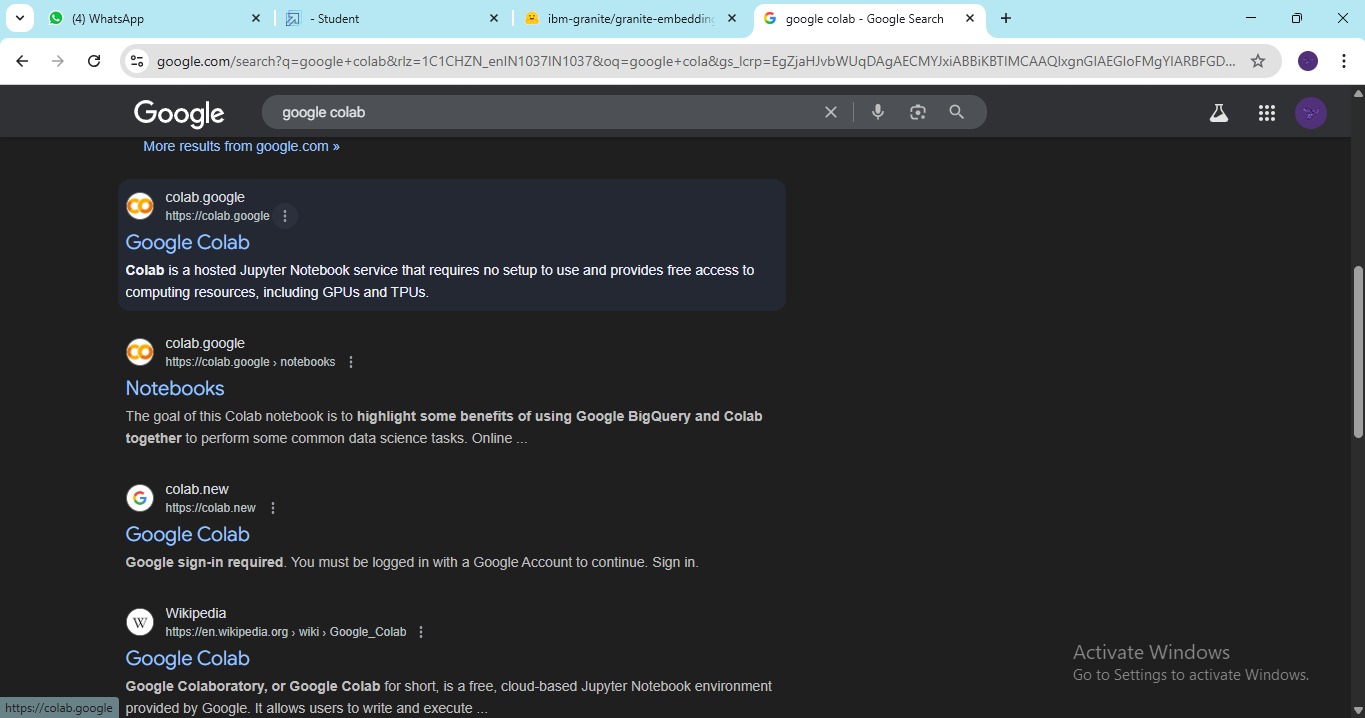
● Here for this project we are using “granite-3.2-2b-instruct” which is compatible fast and light weight.



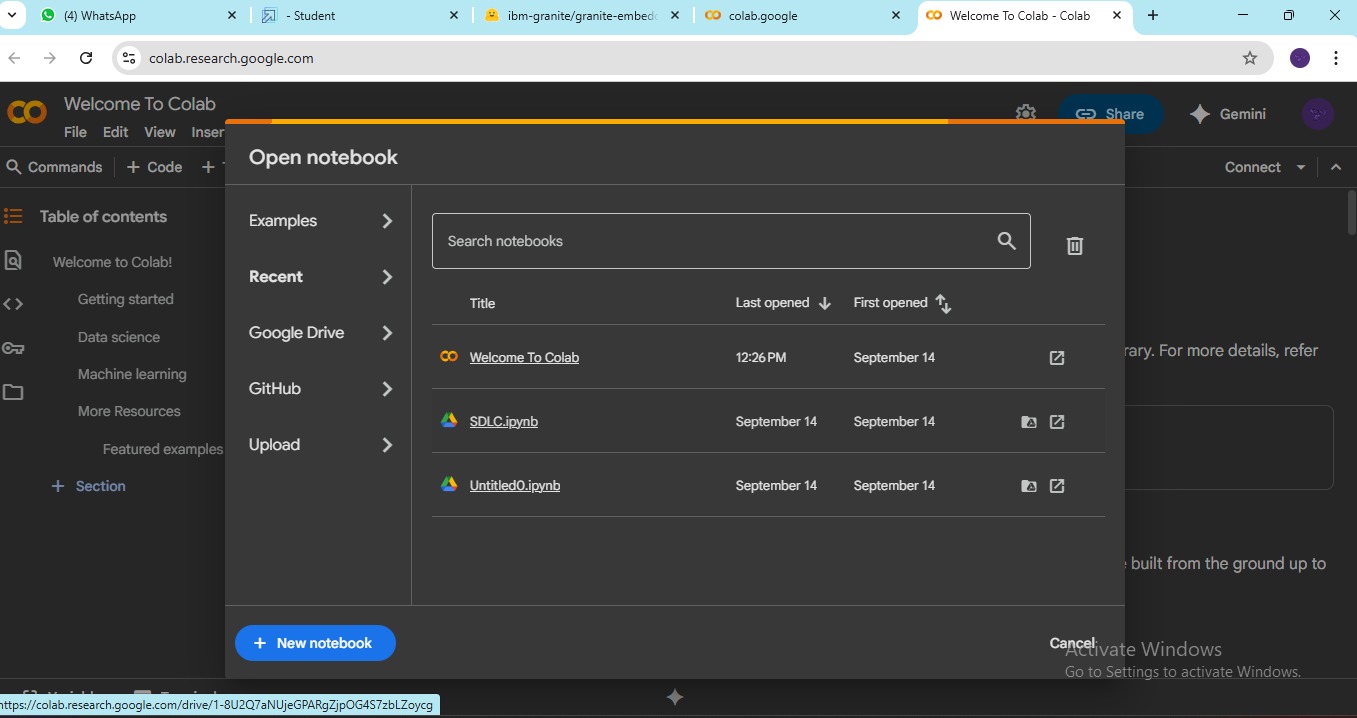
● Now we will start building our project in Google collab.

Activity-3: Running Application in Google Collab.

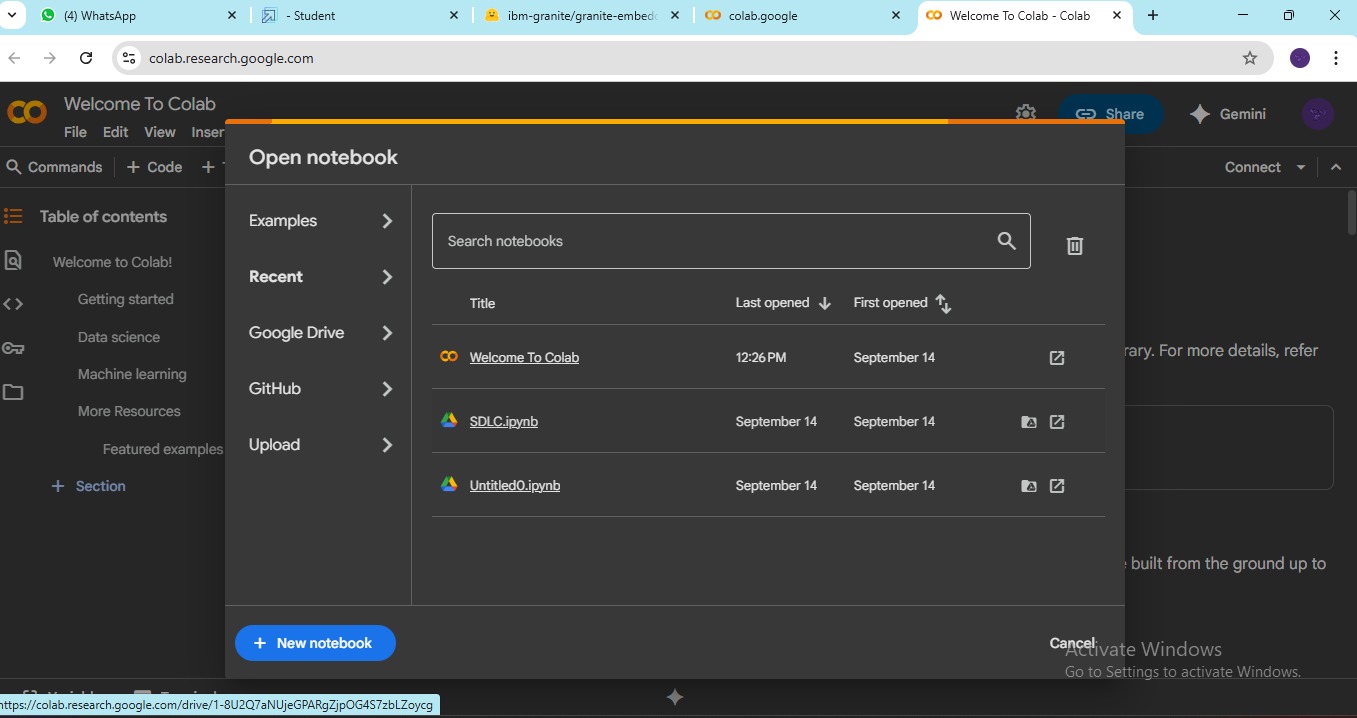
● Search for “Google collab” in any browser.



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

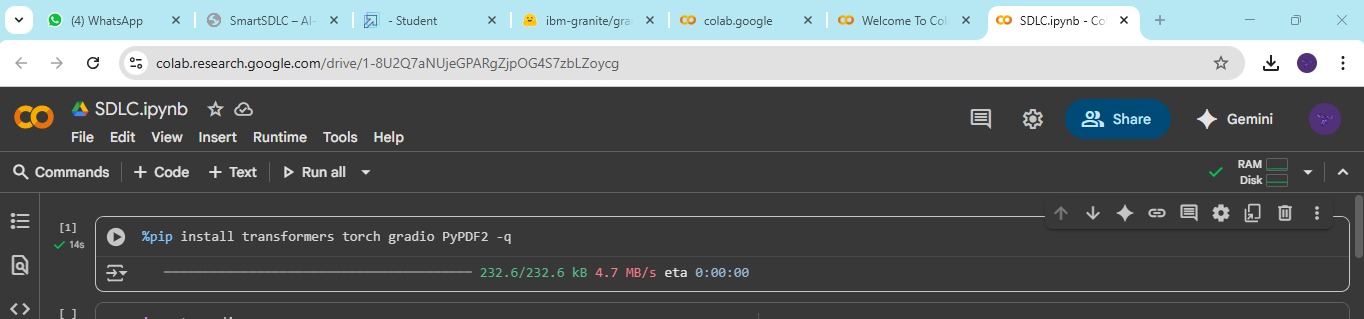


● Click on “New Notebook”

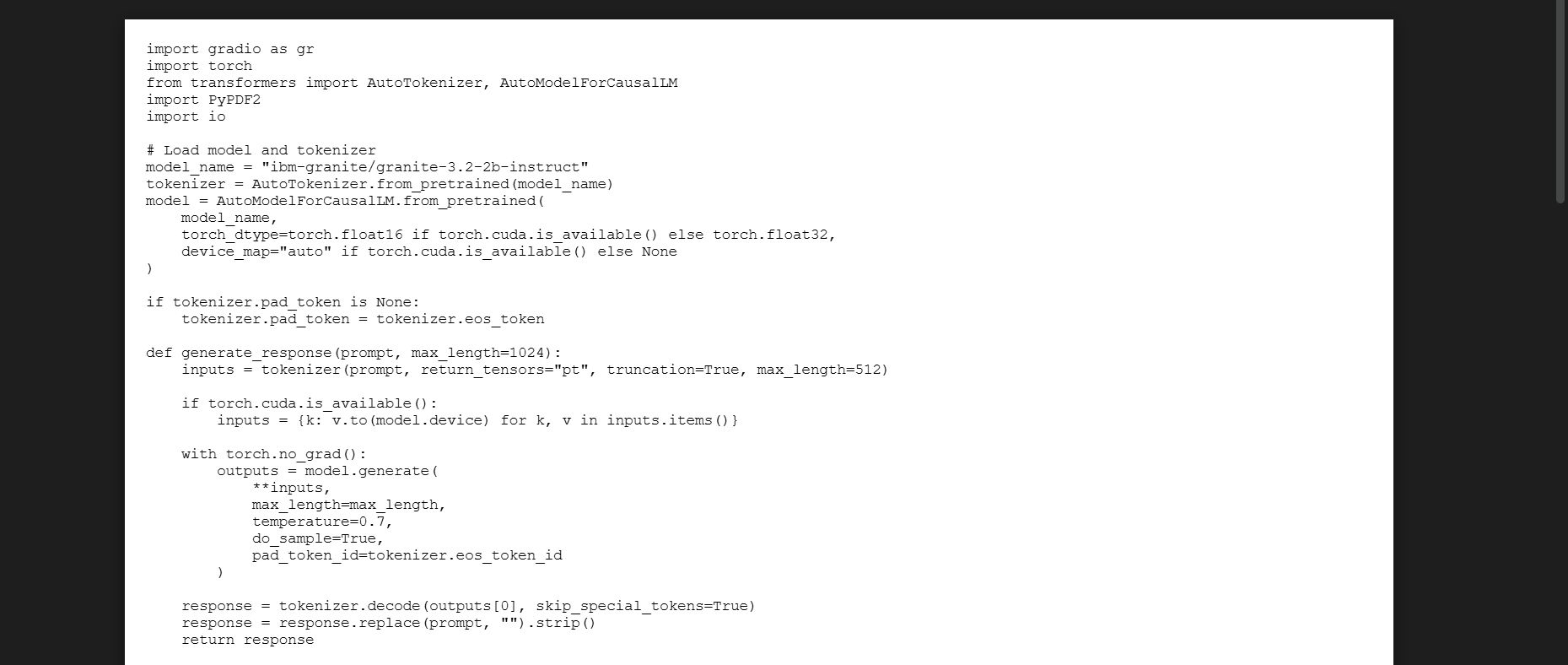


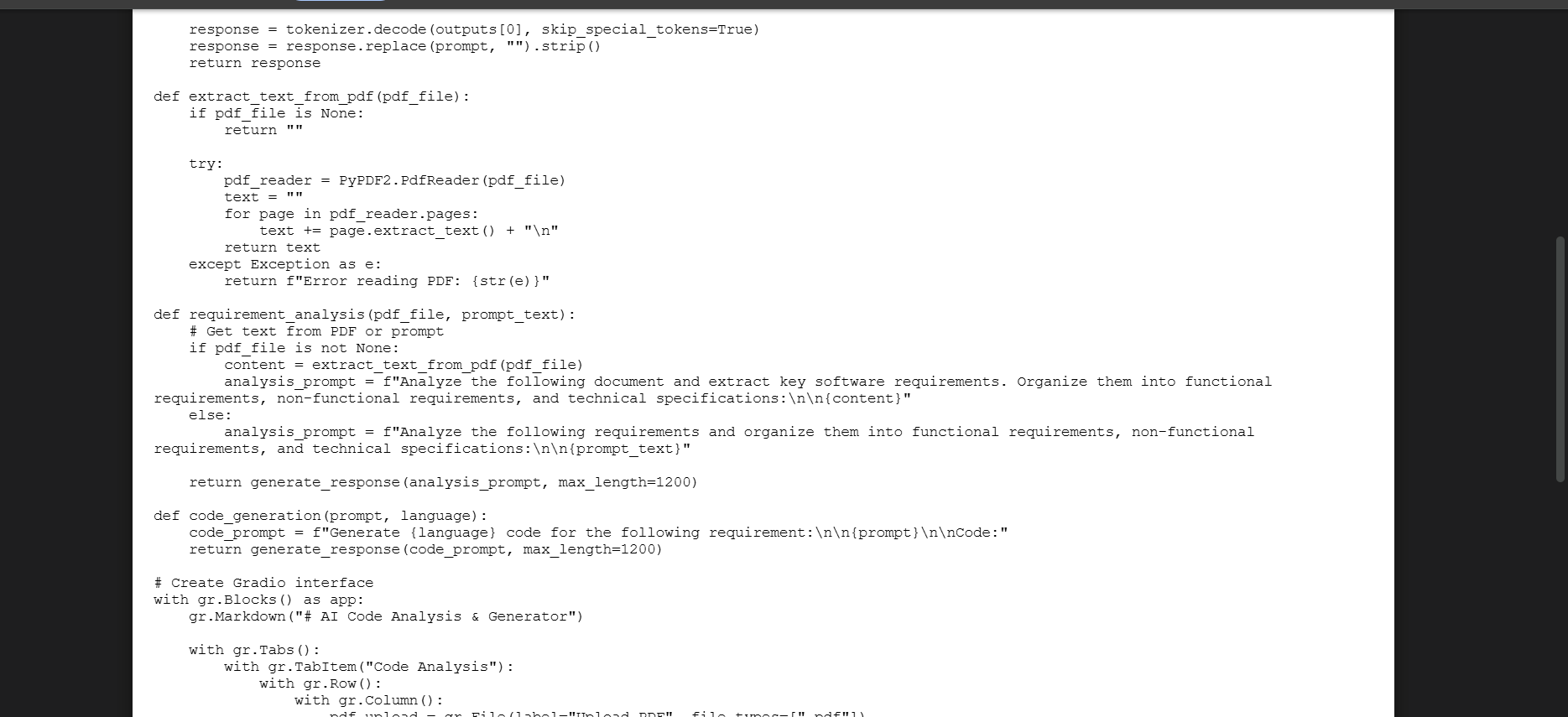
● Change the title of the notebook “Untitled” to “Health AI”. Then click on “Runtime”, then go to “Change Runtime Type”.

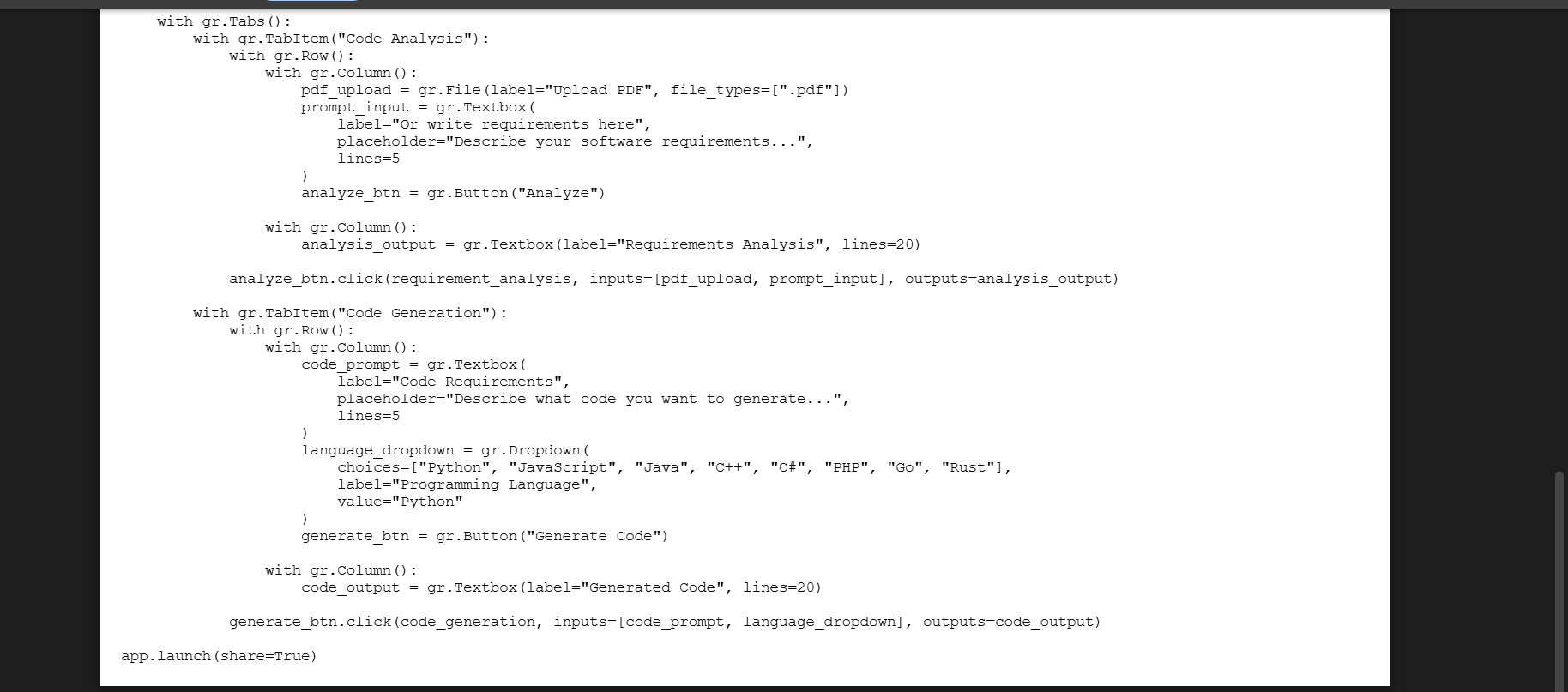
● Choose “T4 GPU” and click on “Save”



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

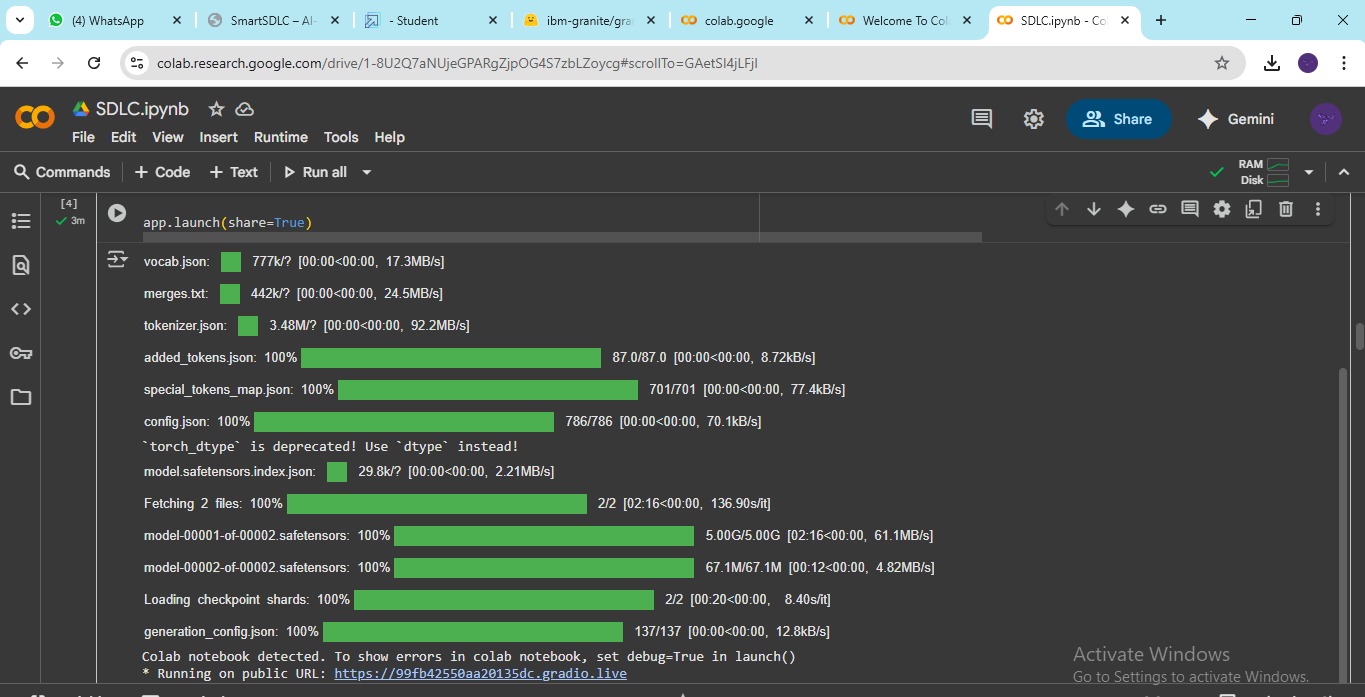






● You can find the code here in this link: [SmartSDLC](https://drive.google.com/file/d/1oZLJGObohBiJUjPPcQeCHPJYHxI9xBUS/view)

OUTPUT:

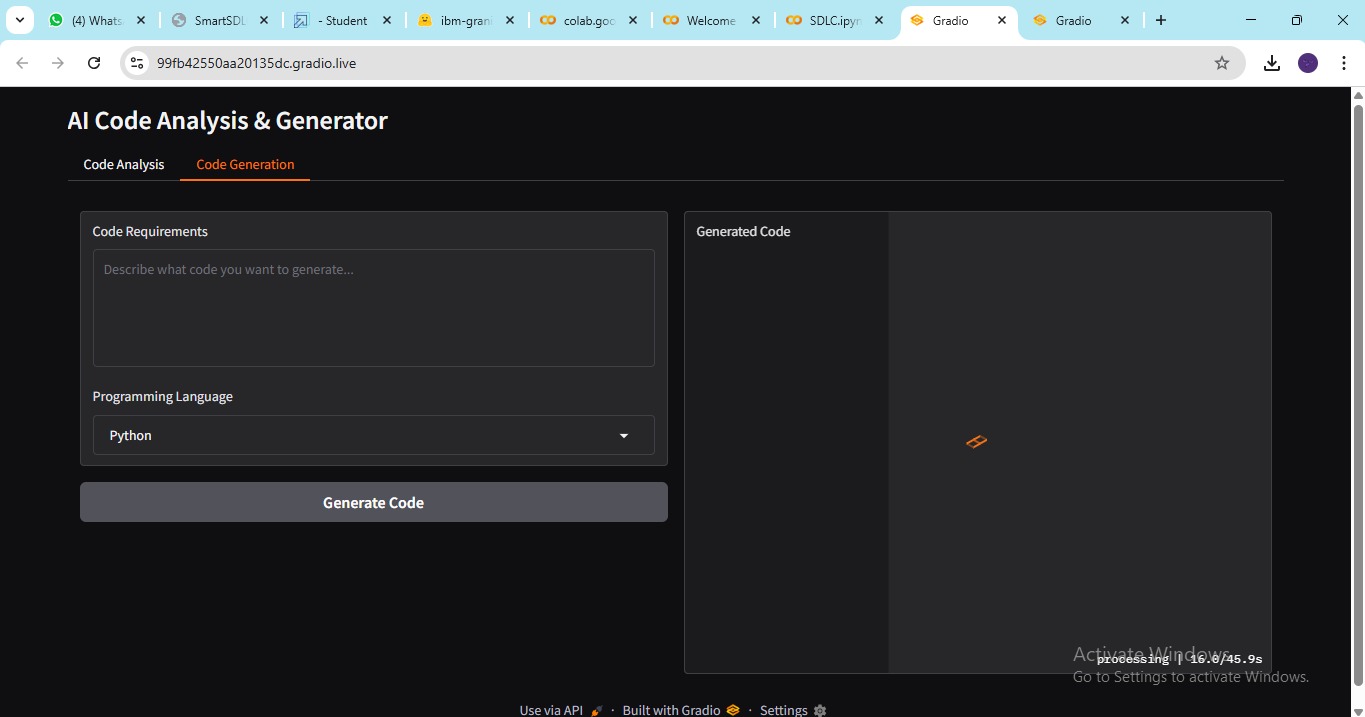


● Now you can see our model is being Downloaded and application is running

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

\*Running on public URL : https://92a1779dd5f3dd5ddc.gradio.live/

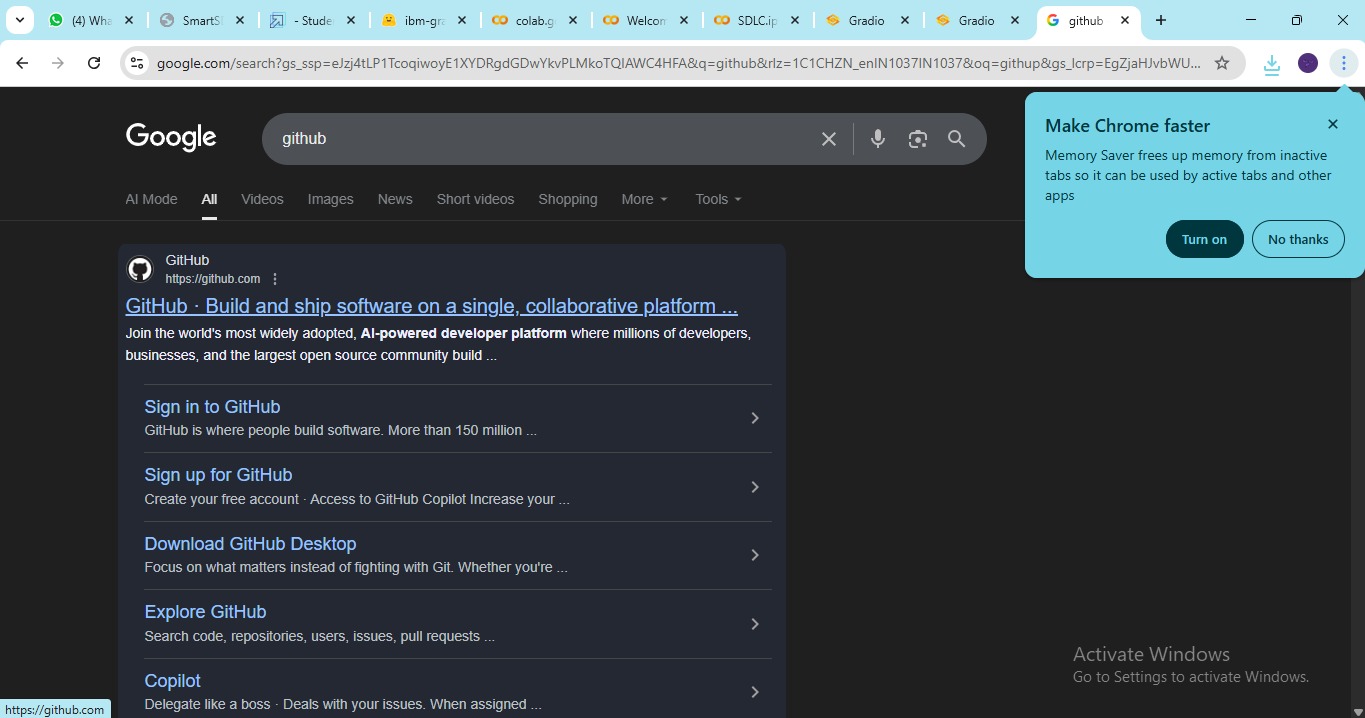
● Click on the URl to open the Gradio Application click on the link



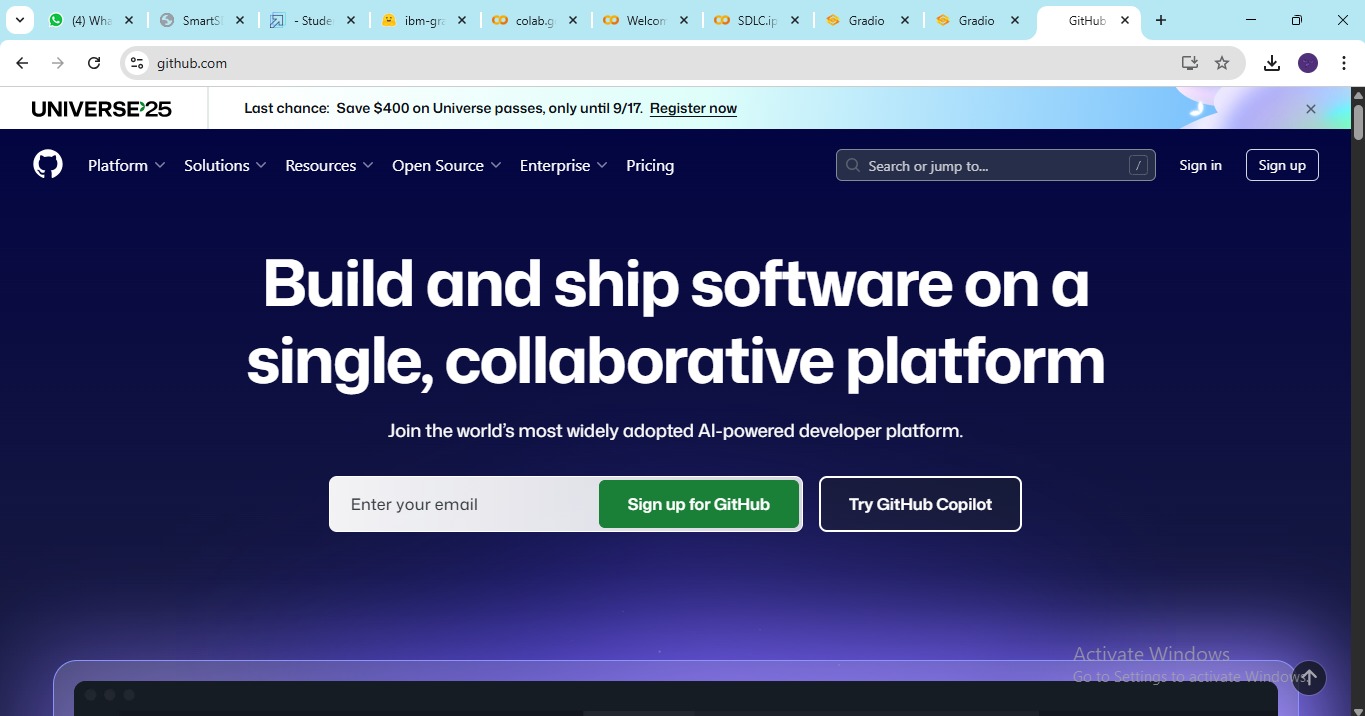
● You can View the Application is the running in the other tab.

Activity-4: Upload Your Project in GitHub.

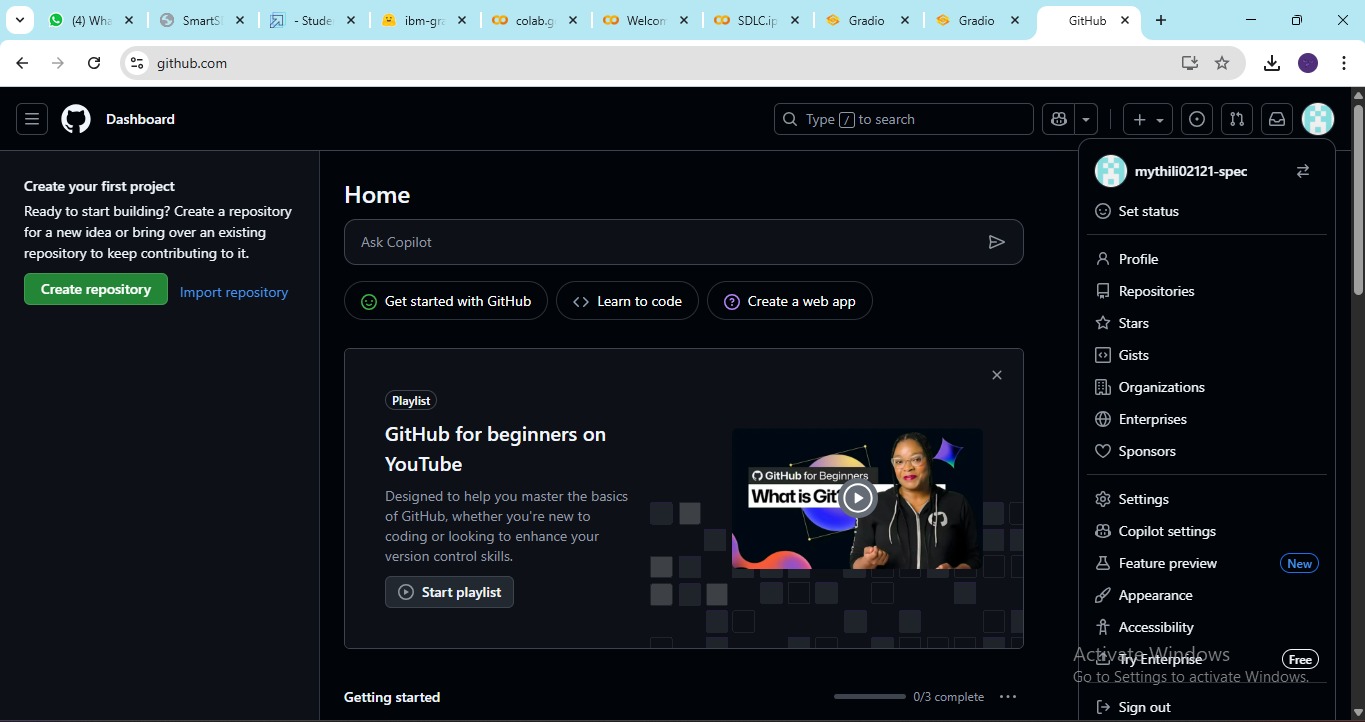
● Search for “GitHub” in any browser, then click on the first link ([GitHub](https://github.com/)).



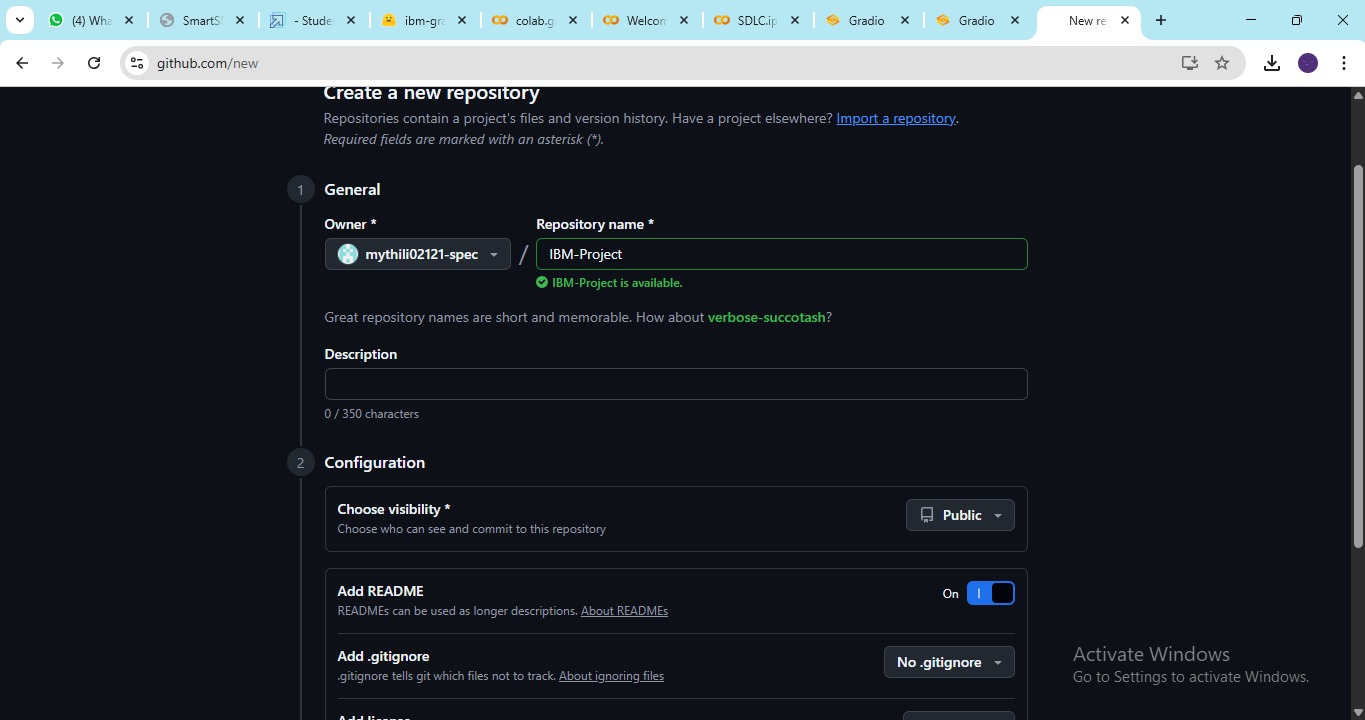
● Then click on “Signup” and create your own account in GitHub. If you already have an account click on “Sign in”.



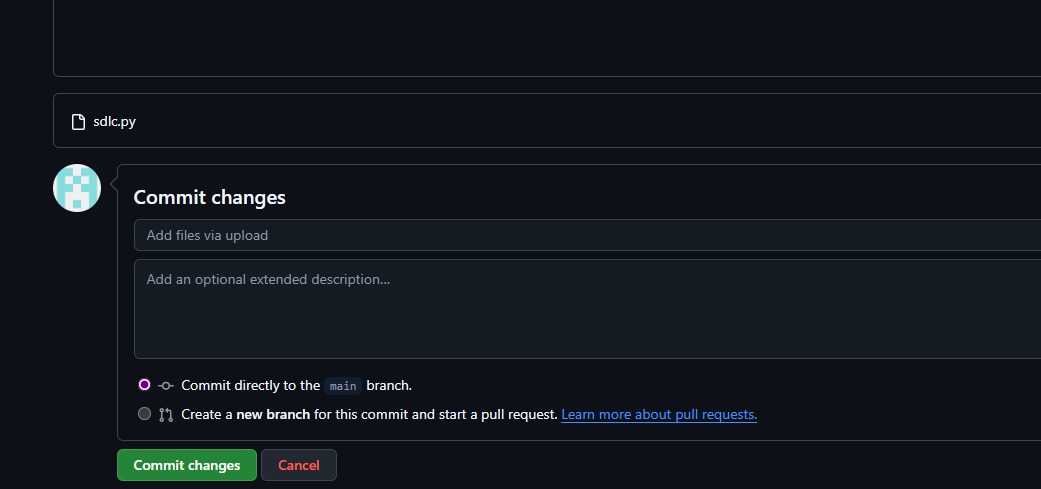
● Click on “Create repository”.



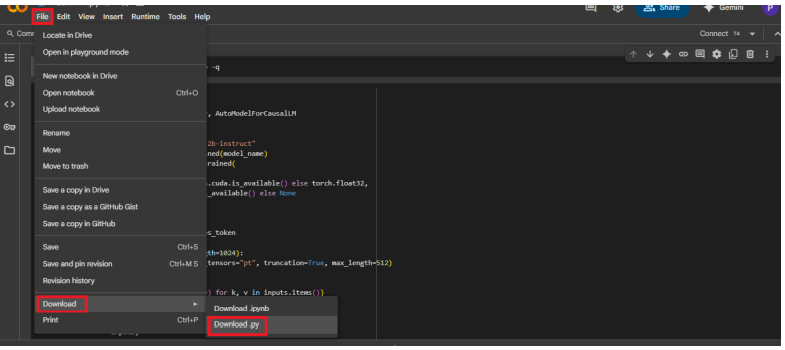
● In “General” Name your repo. (Here I have given “IBM-Project” as my repo name and it is available)



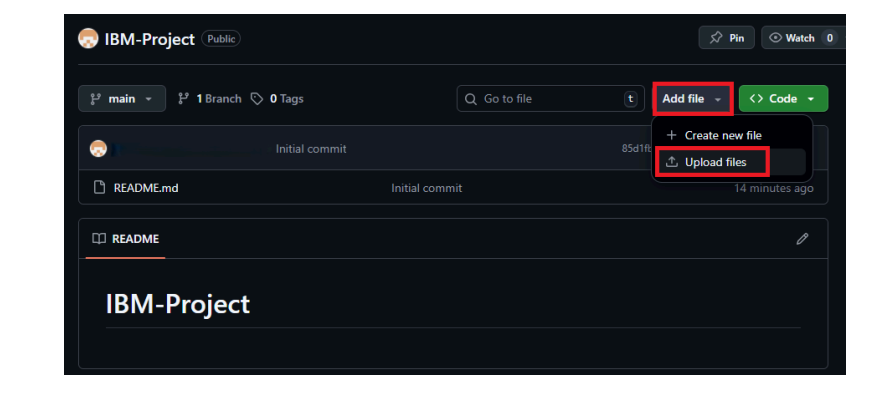
● In “Configurations” Turn On “Add readme” file Option.



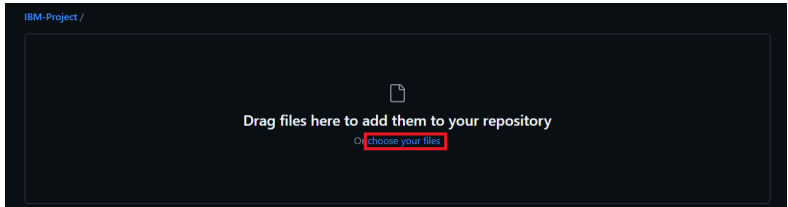
● Now Download your code from Google collab by Clicking on “File”, then Goto “Download” then download as “. Py ”.



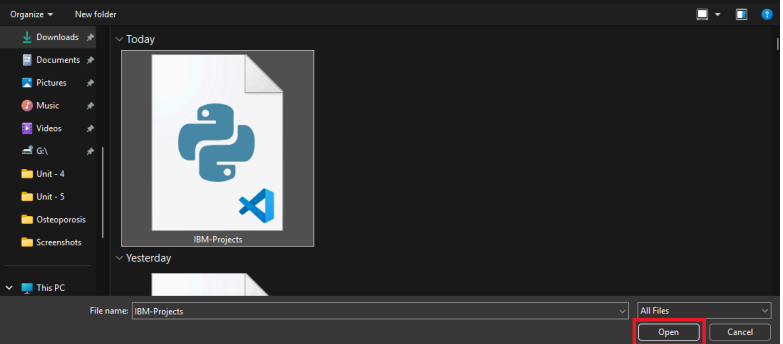
● Then your repository is created, then Click on “Add file” Option. Then Click 15 “Upload files” to upload your files.



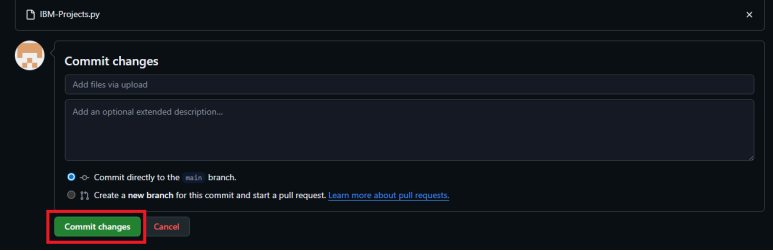
● Click on “choose your files”.



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



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



**Future Enhancement:**

* Add support for multiple programming languages instead of just one.
* Provide debugging tips and auto-correction for errors.
* Suggest optimized or better ways to write the same code.
* Use AI and machine learning so the system learns and improves with time.
* Allow real-time collaboration, where many users can work together on code.
* Develop a mobile app or cloud version for easy access from anywhere.

**Pre-requisites:**

* Gradio Framework Knowledge: [Gradio Documentation](https://www.gradio.app/docs)

* IBM Granite Models (Hugging Face):[Hugging face](https://huggingface.co/)

* Python Programming Proficiency: [Python Documentation](https://docs.python.org/3/)

* Version Control with Git:  [Git Documentation](https://git-scm.com/docs/git)

* Google Collab’s T4 GPU Knowledge: [Google Collab](https://www.geeksforgeeks.org/python/how-to-use-gpu-in-google-colab/)