

# Edu Tutor AI: Personalized Learning

*Generative AI with IBM*



## Team members

Aswini  
Gokul Saranya  
Janarthanan  
Hagusundaram

## **Project Description:**

EduTutor AI uses the Granite model from Hugging Face to create simple, personalized learning tools like concept explainers, quizzes generator and add more functionalities that you like. This project is deployed in Google Colab using Granite for low setup effort and reliable 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 colab](#)
- 

## **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.

The screenshot shows a Google search results page for the query "Naan mudhalavan Smartinternz". The top result is a link to "SmartInternz" with the URL "https://naanmudhalavan.smartinternz.com". Below the link, there is a snippet of text: "The program aims to enhance the employability, innovation, and entrepreneurship skills of students through hands-on, project-based learning under the guidance ...". There are several other links listed below, including "Faculty / Mentor Login", "University Login", "Generative AI with IBM ...", and "Salesforce Developer Projects". At the bottom of the list, there is a link to "More results from smartinternz.com".

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

The screenshot shows the login interface for the Naan Mudhalavan Smart Internz portal. The background features a dark image of students working on a project. On the right side, a modal window titled "Login as Student" is displayed, containing fields for "Email" and "Password", a "Forgot Password?" link, a reCAPTCHA checkbox labeled "I'm not a robot", and a "Login" button. The top of the page has the text "Naan Mudhalavan Project Based Experiential Learning" and logos for SMARTBRIDGE and Smart Internz. A "login as" dropdown menu is visible in the top right corner.

In Partnership with



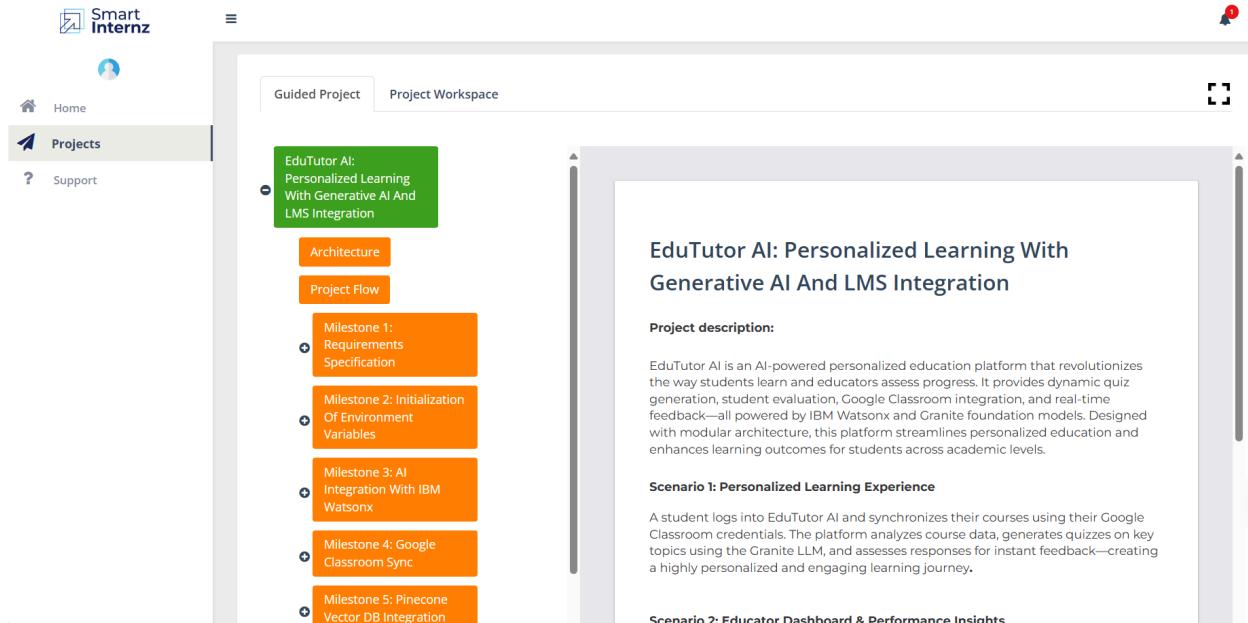
- Then you will be redirected to your account then click on “Projects” Section. There you can see which project you have enrolled in here it is “EduTutor AI”.

The screenshot shows the Smart Internz platform interface. On the left, there's a sidebar with a user profile icon and three menu items: "Home", "Projects", and "Support". The "Projects" item is highlighted with a blue background. The main content area has a title "Project" with a checkmark icon, followed by the project name "EduTutor AI: Personalized Learning with Generative AI and LMS Integration". A green button labeled "ACTIVE" is visible, along with a "Access Resources" button.

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

The screenshot shows the "Guided Projects" section of the Smart Internz platform. The top navigation bar includes tabs for "Instructions", "Courses", "Learning Resources", "Guided Projects" (which is currently selected and highlighted in blue), and "Assessments". Below the tabs, there's a card for the "Edututor Ai: Personalized Learning With Generative Ai And Lms" project, featuring a thumbnail image of four people interacting with AI interfaces. A green button labeled "Go To Workspace" is at the bottom of the card.

- 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”.

The screenshot shows the Smart Internz platform interface. On the left, there's a sidebar with 'Smart Internz' logo, user profile icon, and navigation links: Home, Projects (selected), and Support. The main area is titled 'Project Workspace' under 'Guided Project'. It displays project details: Project Title (EduTutor AI: Personalized Learning with Generative AI and LMS Integration), NM ID (3E531B76720F34C95B94C564E4909437), Industry Mentor(s) Name (No Mentor has been assigned), and a 'Project Progress' circle at 0.00%. Below this is a 'GENERAL INSTRUCTION' section with a 'SHOW' button. At the bottom, there are buttons for 'Demo Link' (0 comments), 'View Mentor Comments' (0 comments), and 'View Industry Mentor Comments'.

- 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.

The screenshot shows search results for "Hugging Face" on Copilot Search. The top navigation bar includes 'ALL', 'SEARCH', 'IMAGES', 'VIDEOS', 'MAPS', 'NEWS', 'COPILOT', 'MORE', and 'TOOLS'. The search bar shows "Copilot Search". The first result is "Hugging Face" with the URL <https://huggingface.co>. Below it, a summary says "Hugging Face – The AI community building the future." and "Host and collaborate on unlimited public models, datasets and applications. With the HF Open source stack. Text, Image, video, audio or even 3D. Share your work with the world and build y...". Other results include "Models", "Datasets", "Documentation", "Spaces", "Log In", "Sign Up", and a featured article "#1-Getting Started Building Generative AI Using HuggingFace ... - YouTube". To the right, there's a sidebar with "How to Huggin Source" and a snippet about "langchain\_huggingface".

- 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.

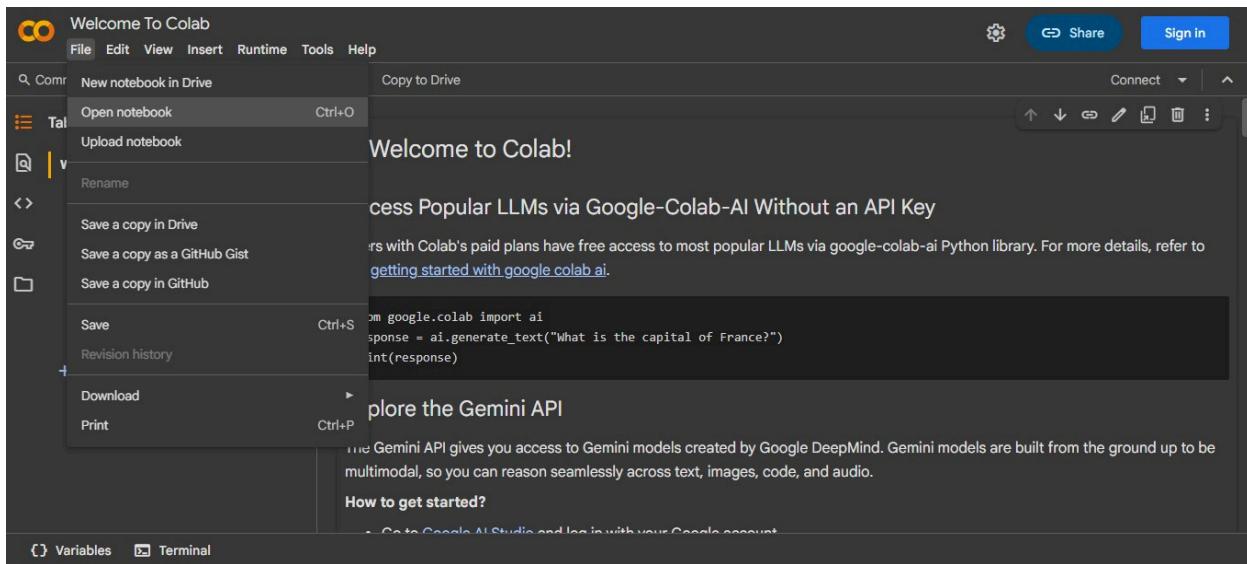
- Now we will start building our project in Google collab.

## Activity-3: Running Application in Google Collab.

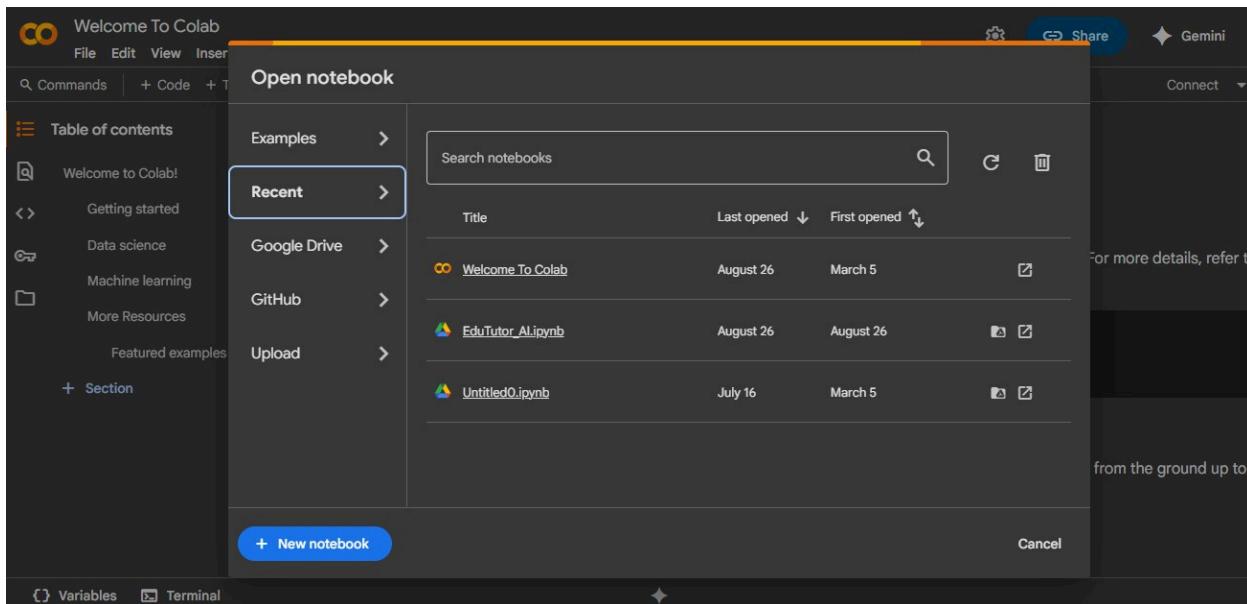
Search for “Google collab” in any browser.

The screenshot shows a search results page from a web browser. At the top, there is a navigation bar with tabs for ALL, SEARCH, VIDEOS, IMAGES, MAPS, NEWS, COPILOT, MORE, and TOOLS. Below the navigation bar, it says "About 214,000 results". The first result is a link to "Google Colab" with the URL "https://colab.research.google.com". A snippet of the page content for "Google Colab" reads: "Colab lets you write and execute Python code in your browser, with access to GPUs and TPUs, and easy sharing of notebooks. Learn how to use Colab for data analysis, visualization, ...". There are also links for "Help", "Importing Libraries and Instal...", "Sign In", and "Colab Github Integration". To the right of the search results, there is a sidebar titled "Related searches for google collab" with suggestions: "google collab login", "google collab pricing", "google collab alternative", "colab notebook", and "google collab online".

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



- Click on “New Notebook”

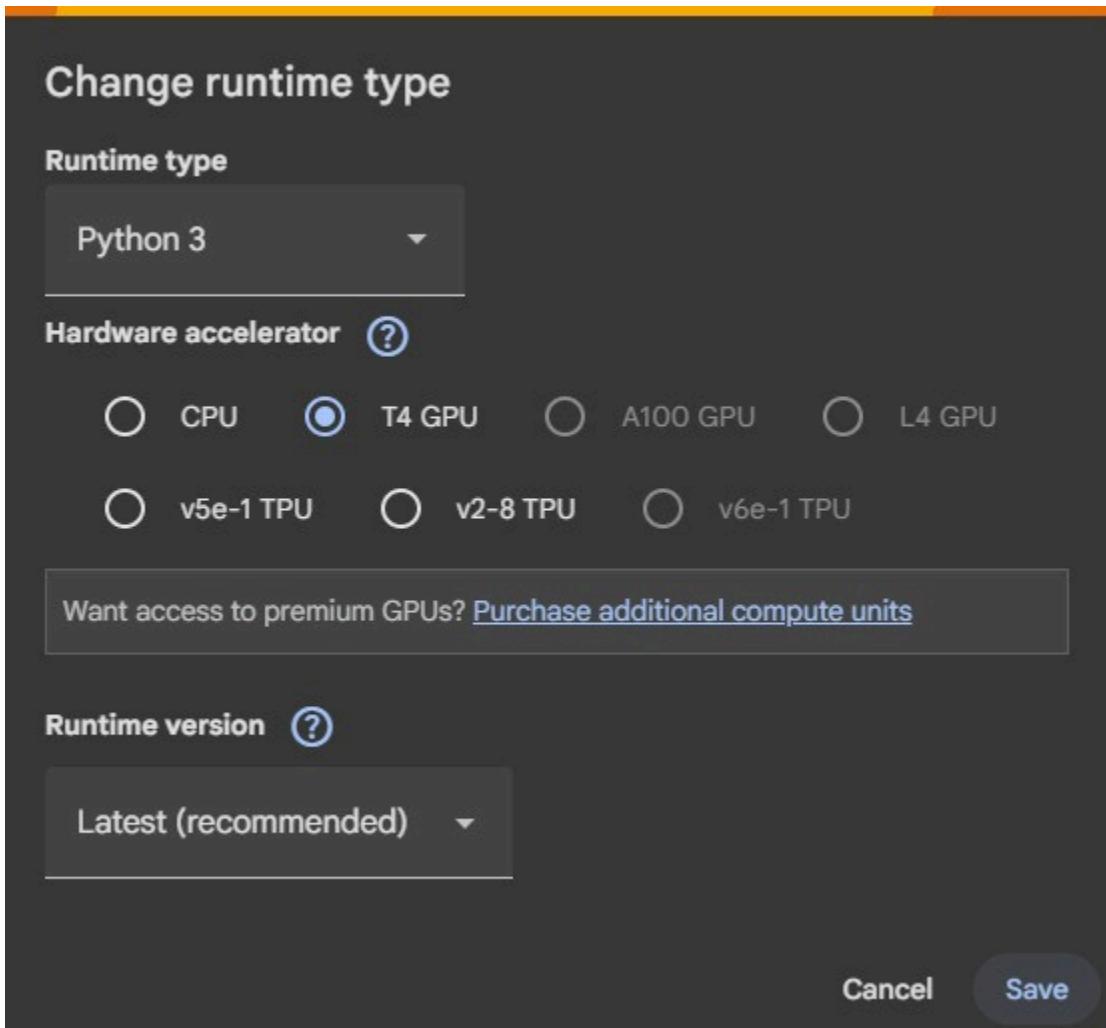


- Change the title of the notebook “Untitled” to “Health AI”. Then click on “Runtime”, then go to “Change Runtime Type”.

```
[ ] !pip install transformers
import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM
model_name = "t5-small"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name, torch_dtype=torch.float32,
                                              device_map="auto")
if tokenizer.pad_token_id is None:
    tokenizer.pad_token_id = 0
def generate_response(inputs):
    if torch.cuda.is_available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}

    outputs = model.generate(**inputs, max_length=512)
```

- Choose “T4 GPU” and click on “Save”



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

A screenshot of a Jupyter Notebook cell. The cell contains the command `!pip install transformers torch gradio -q`. Below the cell, a message says 'Run cell (Ctrl+Enter)' and 'cell has not been executed in this session'.

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

```

1 # Educational AI Application using IBM Granite Model
2 # Run this in Google Colab
3 !pip install transformers torch gradio -q

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

```

```

31
32     response = tokenizer.decode(outputs[0], skip_special_tokens=True)
33     response = response.replace(prompt, "").strip()
34     return response
35
36 def concept_explanation(concept):
37     prompt = f"Explain the concept of {concept} in detail with examples."
38     return generate_response(prompt, max_length=800)
39
40 def quiz_generator(concept):
41     prompt = f"Generate 5 quiz questions about {concept} with different question types (multiple choice, true/false, short answer). At the end, provide all the answers in a separate ANSWERS section."
42     return generate_response(prompt, max_length=1000)
43
44 # Create Gradio interface
45 with gr.Blocks() as app:
46     gr.Markdown("# Educational AI Assistant")
47
48     with gr.Tabs():
49         with gr.TabItem("Concept Explanation"):
50             concept_input = gr.Textbox(label="Enter a concept", placeholder="e.g., machine learning")
51             explain_btn = gr.Button("Explain")
52             explanation_output = gr.Textbox(label="Explanation", lines=10)
53
54             explain_btn.click(concept_explanation, inputs=concept_input, outputs=explanation_output)
55
56         with gr.TabItem("Quiz Generator"):
57             quiz_input = gr.Textbox(label="Enter a topic", placeholder="e.g., physics")
58             quiz_btn = gr.Button("Generate Quiz")
59             quiz_output = gr.Textbox(label="Quiz Questions", lines=15)
60
61             quiz_btn.click(quiz_generator, inputs=quiz_input, outputs=quiz_output)
62
63 app.launch(shallow=True)

```

- You can find the code here in this link: [Edu Tutor AI Code](#)

## OUTPUT:

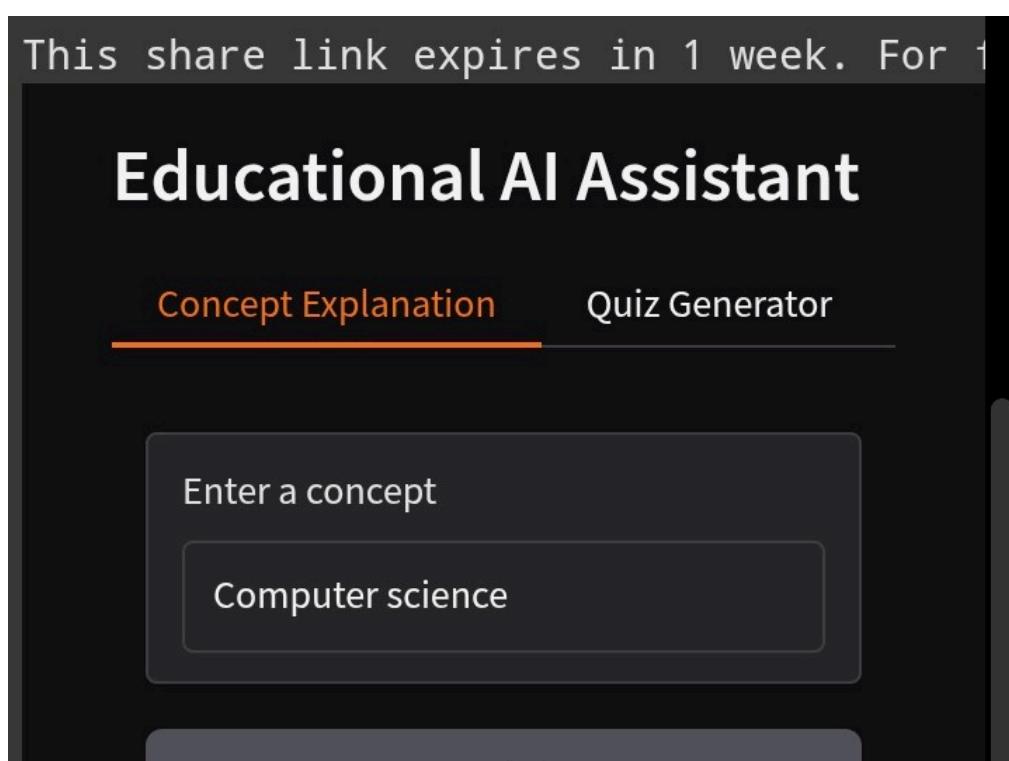
- Now you can see our model is being Downloaded and the application is running.

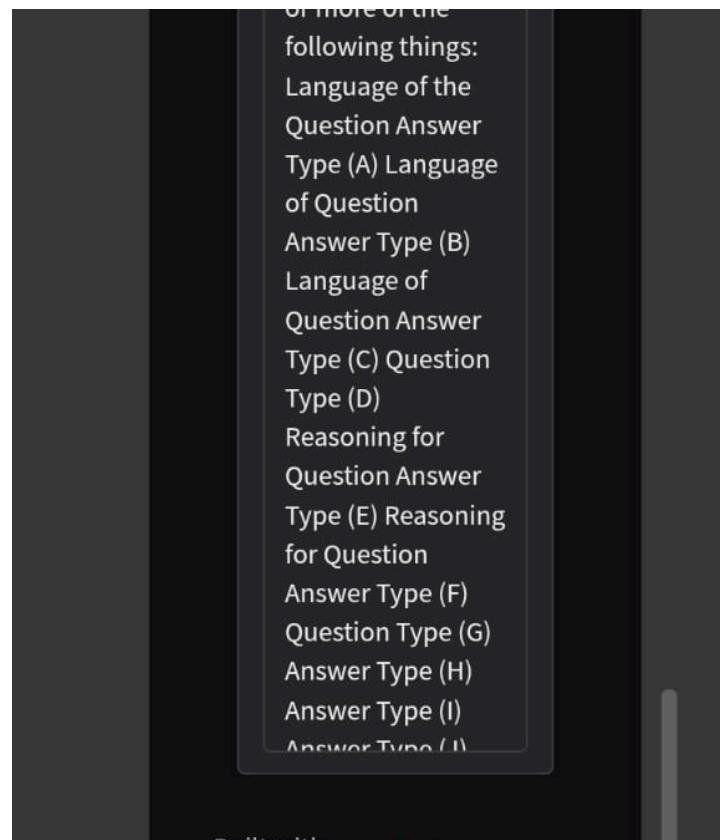
```
→ /usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:  
The secret 'HF_TOKEN' does not exist in your Colab secrets.  
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.  
You will be able to reuse this secret in all of your notebooks.  
Please note that authentication is recommended but still optional to access public models or datasets.  
warnings.warn(  
    tokenizer_config.json: 8.88kB? [00:00<00:00, 695kB/s]  
    vocab.json: 777kB? [00:00<00:00, 30.9MB/s]  
    merges.txt: 442kB? [00:00<00:00, 23.4MB/s]  
    tokenizer.json: 3.48MB? [00:00<00:00, 94.3MB/s]  
    added_tokens.json: 100% [00:00<00:00, 8.14kB/s]  
    special_tokens_map.json: 100% [00:00<00:00, 50.9kB/s]  
    config.json: 100% [00:00<00:00, 48.8kB/s]  
    model.safetensors.index.json: 29.8kB? [00:00<00:00, 2.54MB/s]  
    Fetching 2 files: 100% [00:00<00:00, 141.84kB/s]  
    model-00001-of-00002 safetensors: 100% [00:00<00:00, 50.7MB/s]  
    model-00002-of-00002 safetensors: 100% [00:00<00:00, 37.0MB/s]  
    Loading checkpoint shards: 100% [00:00<00:00, 10.58kB/s]  
    generation_config.json: 100% [00:00<00:00, 10.5kB/s]  
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()  
* Running on public URL: https://92320020f660b93f05.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 (https://huggingface.co)
```

- Click on the URI to open the Gradio Application click on the link.

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()  
\* Running on public URL: <https://92320020f660b93f05.gradio.live>

- You can View the Application is 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)).

Google

Github

All News Videos Images Short videos Shopping Forums More Tools

**GitHub** <https://github.com>

**GitHub · Build and ship software on a single, collaborative platform...**

Millions of developers and businesses call GitHub home. Whether you're scaling your development process or just learning how to code, GitHub is where you belong ...

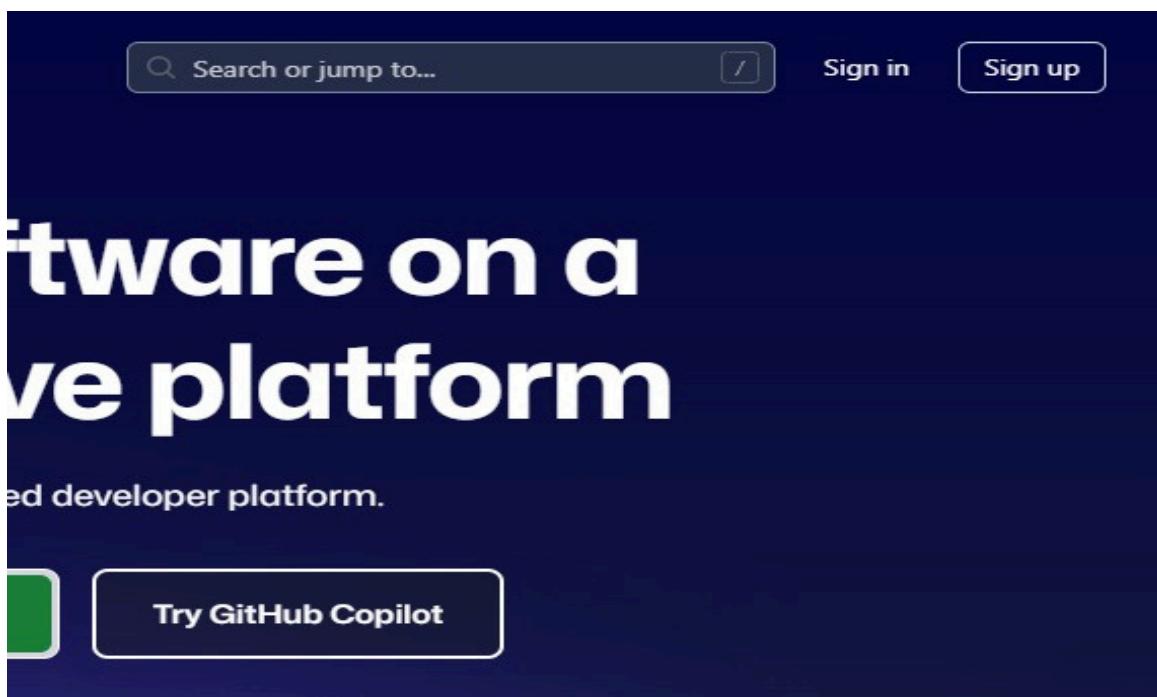
Sign in to GitHub

Sign up for GitHub

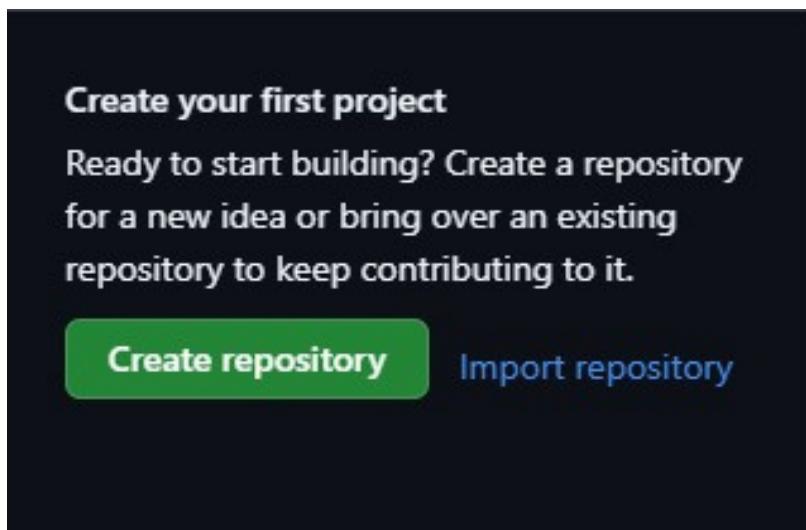
Download GitHub Desktop

Explore 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)

**Create a new repository**

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).  
Required fields are marked with an asterisk (\*).

**1 General**

Owner \*  padamavathikonakala-design / Repository name \* IBM-Project  IBM-Project is available.

Great repository names are short and memorable. How about [sturdy-octo-guacamole](#)?

Description   
0 / 350 characters

**2 Configuration**

Choose visibility \*  Public 

Add README  Off 

READMEs can be used as longer descriptions. [About READMEs](#)

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

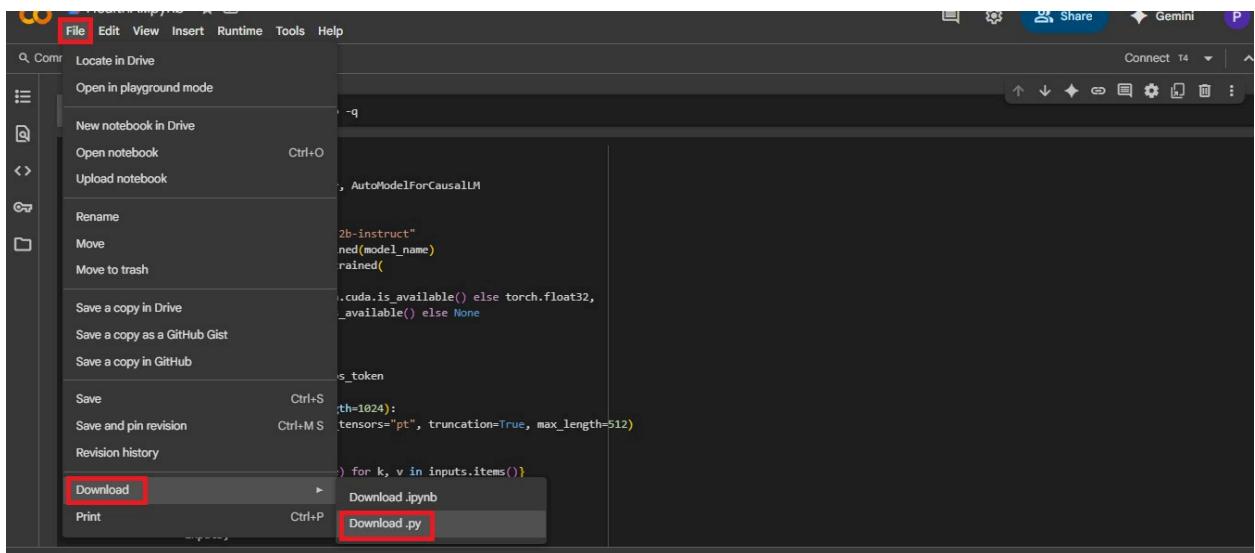
**2 Configuration**

Choose visibility \*  Public 

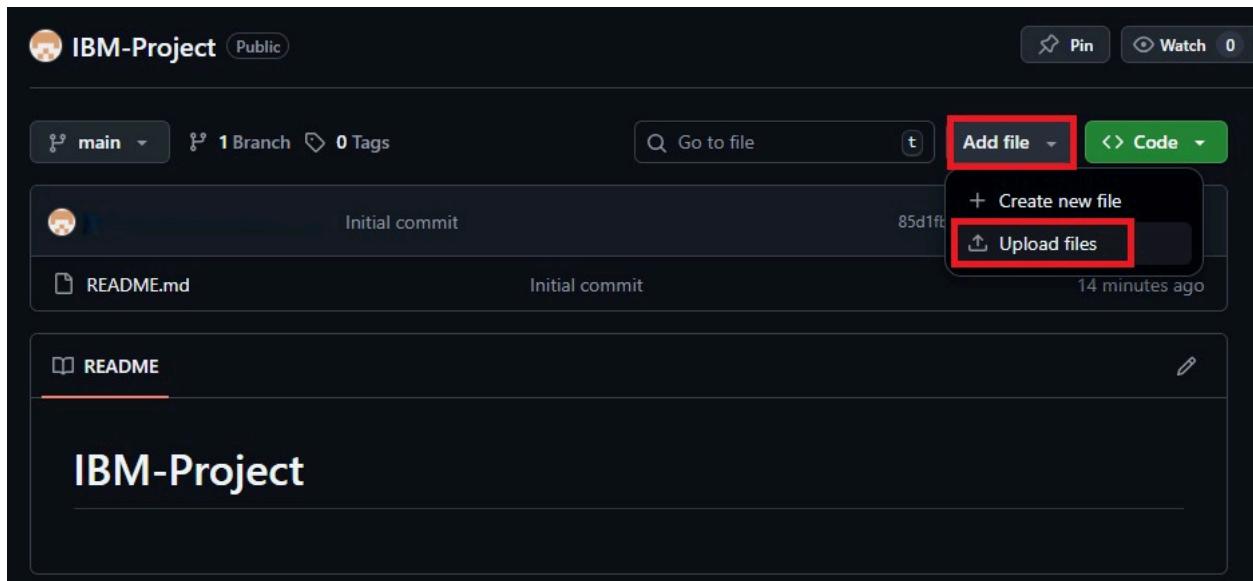
Add README  On 

READMEs can be used as longer descriptions. [About READMEs](#)

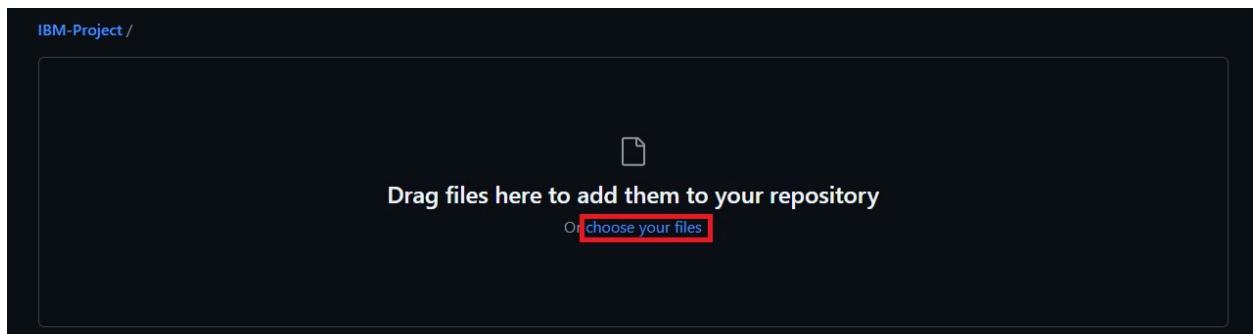
- Now Download your code from Google collab by Clicking on “File”, then Goto “Download” then download as “.py”.



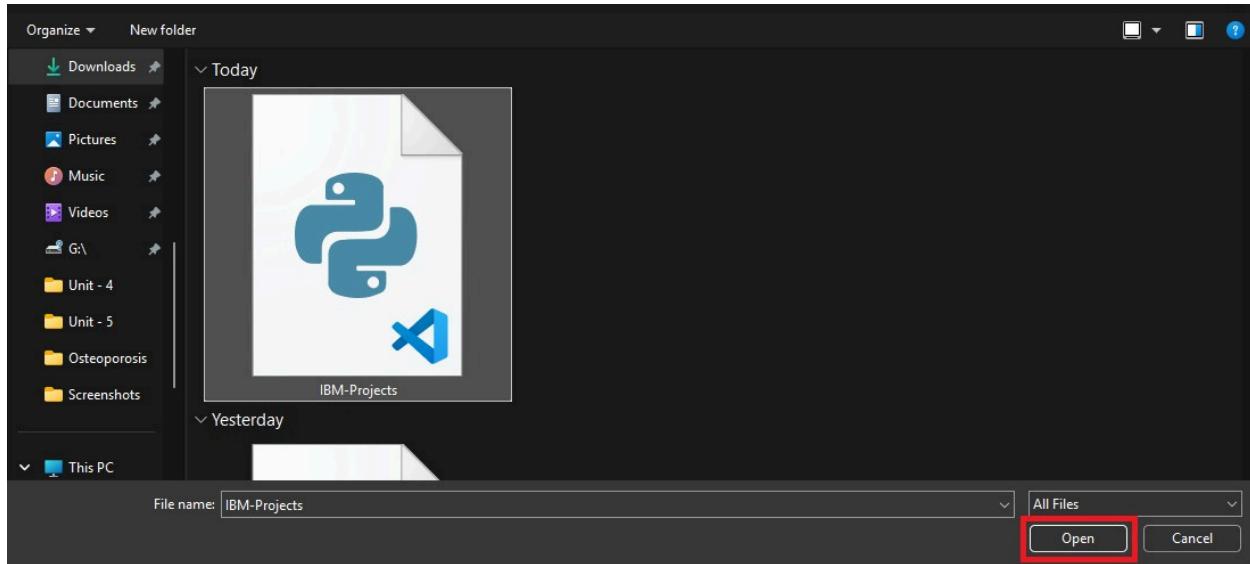
- Then your repository is created, then Click on “Add file” Option. Then Click “Upload files” to upload your files.



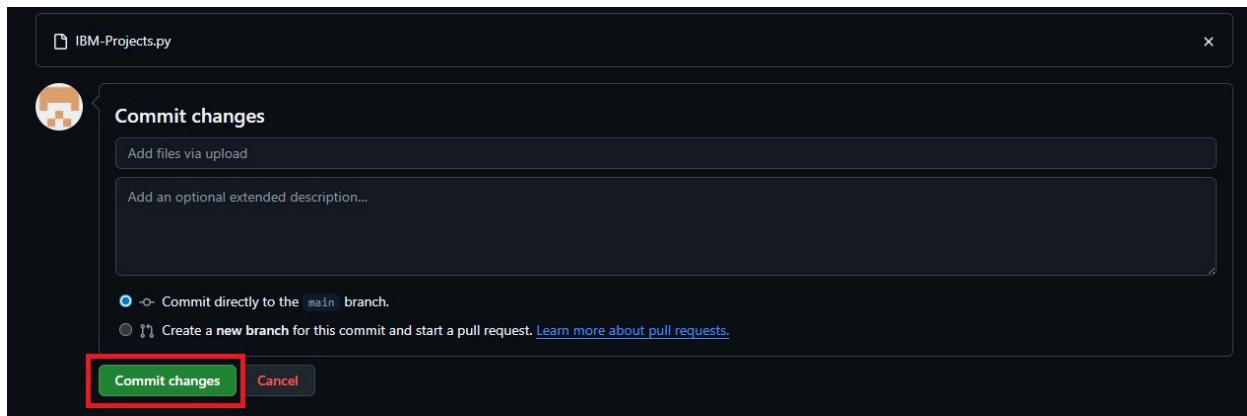
- Click on “choose your files”.



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



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



## Conclusion

EduTutor AI is a revolution\*Conclusion

EduTutor AI is a revolutionary educational tool that harnesses the power of AI to provide personalized learning experiences, transforming the way students learn and interact with complex concepts\*Conclusion\*

EduTutor AI is a revolutionary educational tool that harnesses the power of AI to provide personalized learning experiences, transforming the way students learn and interact with complex concepts.s.utionary educational tool that harnesses the power of AI to provide personalized learning experiences, transforming the way students learn and interact with complex concepts.