

Health AI with IBM

Project Documentation

Project Title: Health AI with IBM

Team Member: Priyadarshini.V

Team Member: Priyadharshani.V

Team Member: Rithika.K

Team Member: Saipriya. P

Introduction

Artificial Intelligence (AI) is transforming healthcare by improving patient care, assisting medical professionals, and enhancing access to medical information. Health AI with IBM is an innovative project that leverages IBM Granite models and generative AI to create an intelligent healthcare assistant capable of predicting diseases, suggesting treatment plans, and interacting with patients through a conversational interface. By integrating AI models from Hugging Face, deploying on Google Colab, and providing a user-friendly design with Gradio, this project demonstrates a practical real-world application of AI in the healthcare domain.

Overview

Health AI with IBM is an innovative project that leverages **Generative AI** to provide intelligent, accessible, and secure **healthcare assistance**. Built with IBM Granite models available on Hugging Face, this project is designed to simplify healthcare interactions for patients while also helping medical professionals manage queries more effectively.

The Health AI system is capable of:

- Patient-friendly **chat-based consultations**
- **Disease prediction** based on symptoms
- **Treatment plan recommendations**
- Integration of additional healthcare features as needed

Unlike traditional healthcare portals, Health AI offers **real-time AI-driven support**, making medical information easier to understand for the general public.

Objective

The core objectives of Health AI include:

- Providing **quick, reliable health advice** to patients
 - Assisting doctors by **automating basic symptom checks**
 - Creating **personalized treatment plans** using AI models
 - Ensuring **affordable deployment** through Google Colab's cloud environment
 - Maintaining project transparency and scalability using GitHub
-

2. Pre-requisites

Before beginning the project, learners must acquire certain **skills and tools**:

1. **Gradio Framework**
 - Gradio is used to build a simple and interactive user interface where patients can chat with the AI.
 - It allows healthcare chatbots to be deployed quickly without complex web development.
2. **IBM Granite Models (Hugging Face)**
 - IBM Granite models are pre-trained AI models optimized for performance and lightweight deployment.
 - Example: granite-3.2-2b-instruct is recommended as it runs efficiently even on Google Colab GPUs.
3. **Python Programming Skills**
 - Python is the primary language for implementing AI models.
 - Knowledge of libraries like transformers, torch, and gradio is essential.
4. **Git & GitHub**
 - Git helps in version control, while GitHub enables project sharing, collaboration, and documentation.
5. **Google Colab T4 GPU**
 - Provides free access to GPU resources.
 - Ensures smooth execution of AI models without requiring expensive hardware.

3. Project Workflow

The project is structured into **four main activities**, ensuring a logical flow from start to deployment:

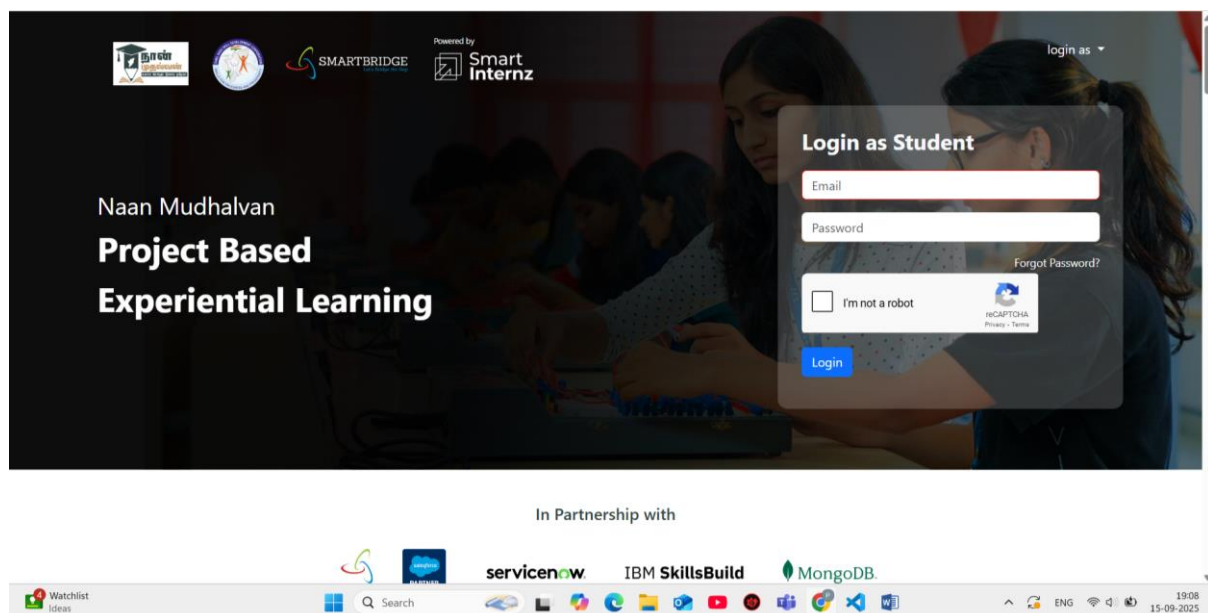
- **Activity 1:** Exploring the **Naan Mudhalavan Smart Interz Portal**
- **Activity 2:** Choosing an **IBM Granite Model** from Hugging Face
- **Activity 3:** Running the **Health AI Application** in Google Colab
- **Activity 4:** Uploading the **Project to GitHub**

This workflow allows learners to gradually understand the process — from accessing the project workspace to **developing, testing, and deploying** a working Health AI model.

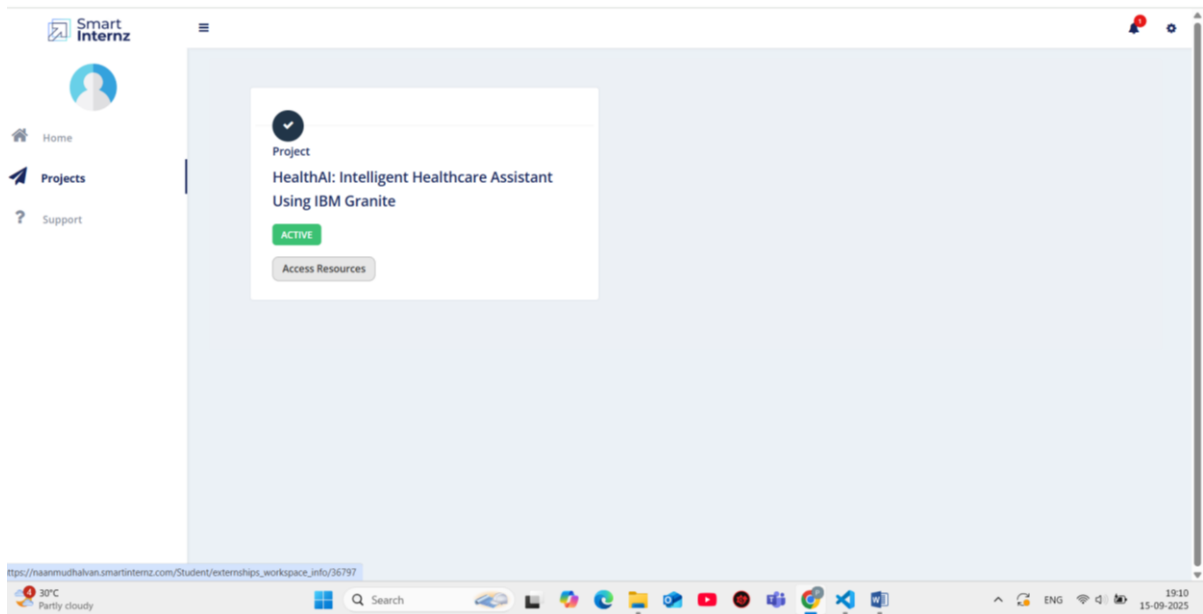
4. Activities in Detail

Activity 1: Exploring the Naan Mudhalavan Smart Interz Portal

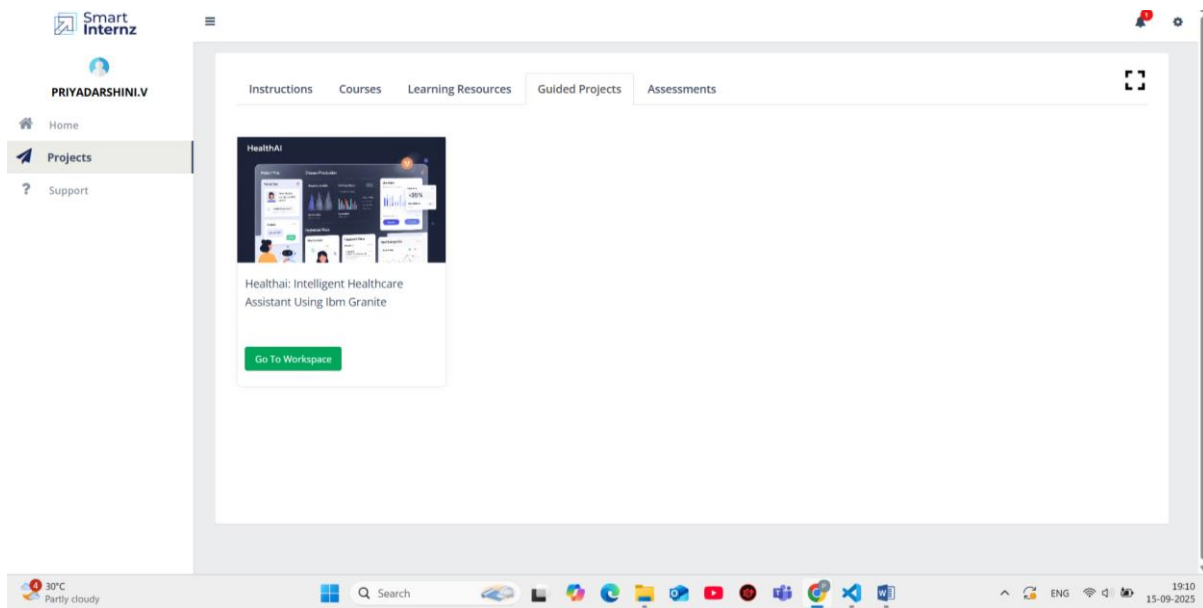
- Open the **Naan Mudhalavan Smart Internz Portal** in a browser.



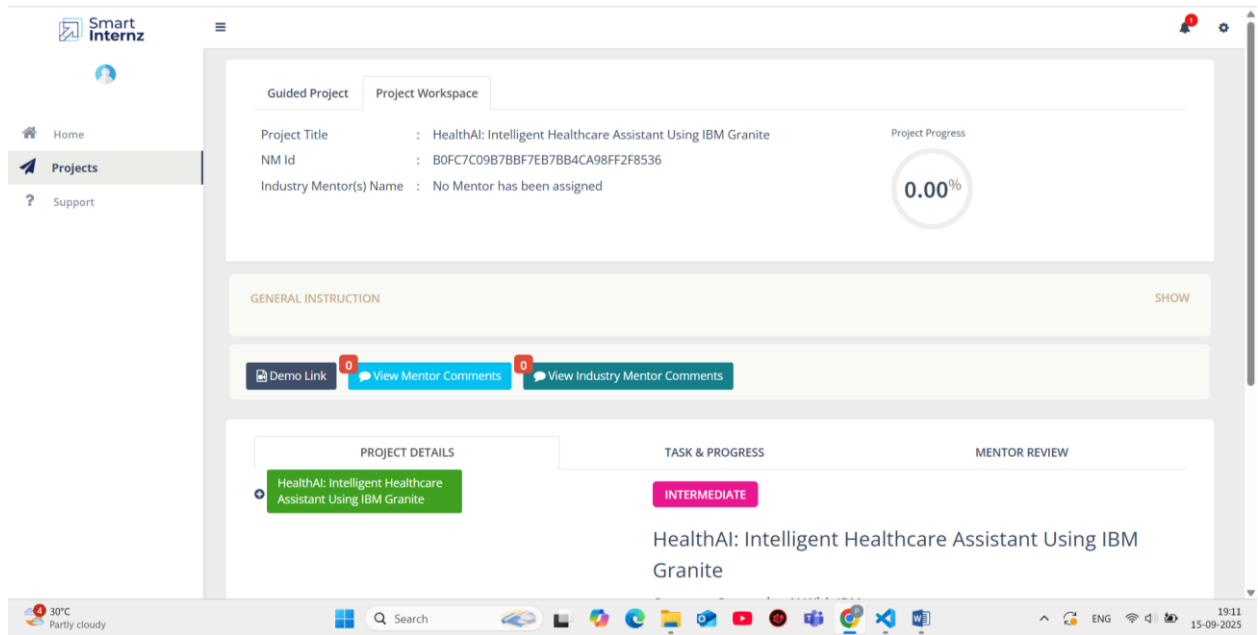
- Log in with registered credentials.
- Navigate to the **Projects Section** and identify the **Health AI project**.



- Access resources from the **Guided Project Section** to understand requirements.



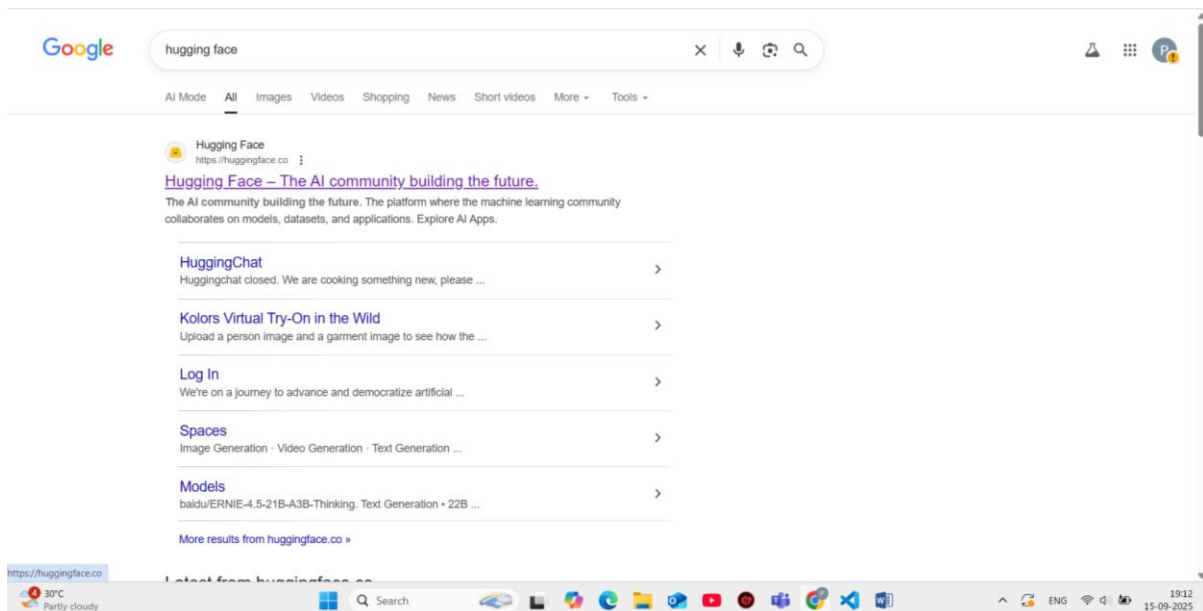
- Open the **Workspace** to track project progress and upload demo links.



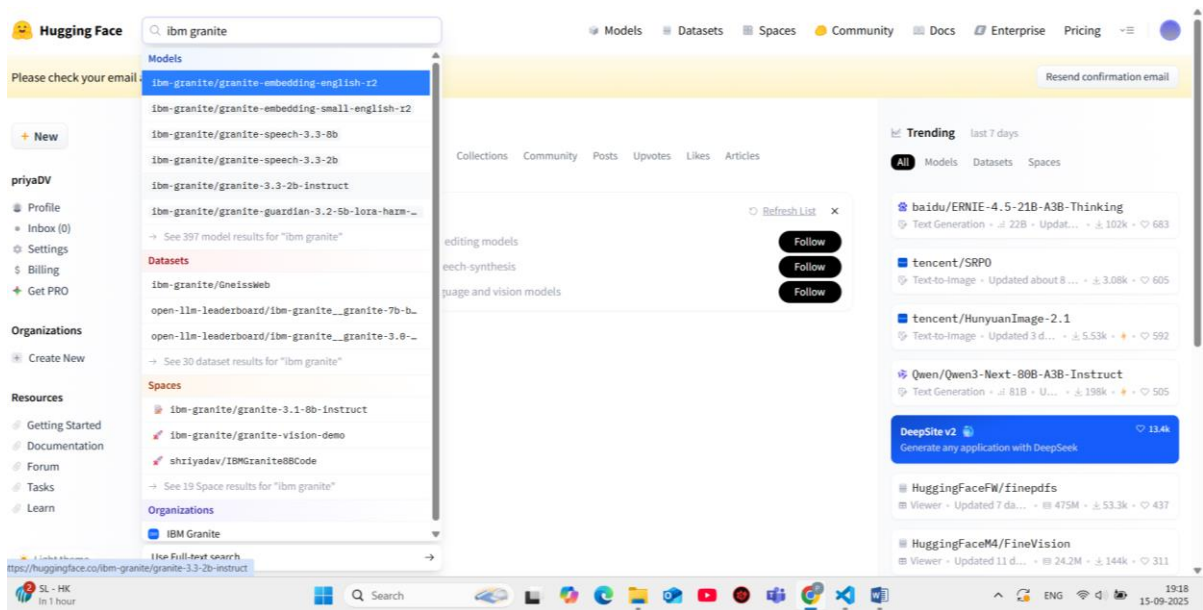
☐ This activity ensures that learners understand the project environment before diving into technical execution.

Activity 2: Choosing an IBM Granite Model

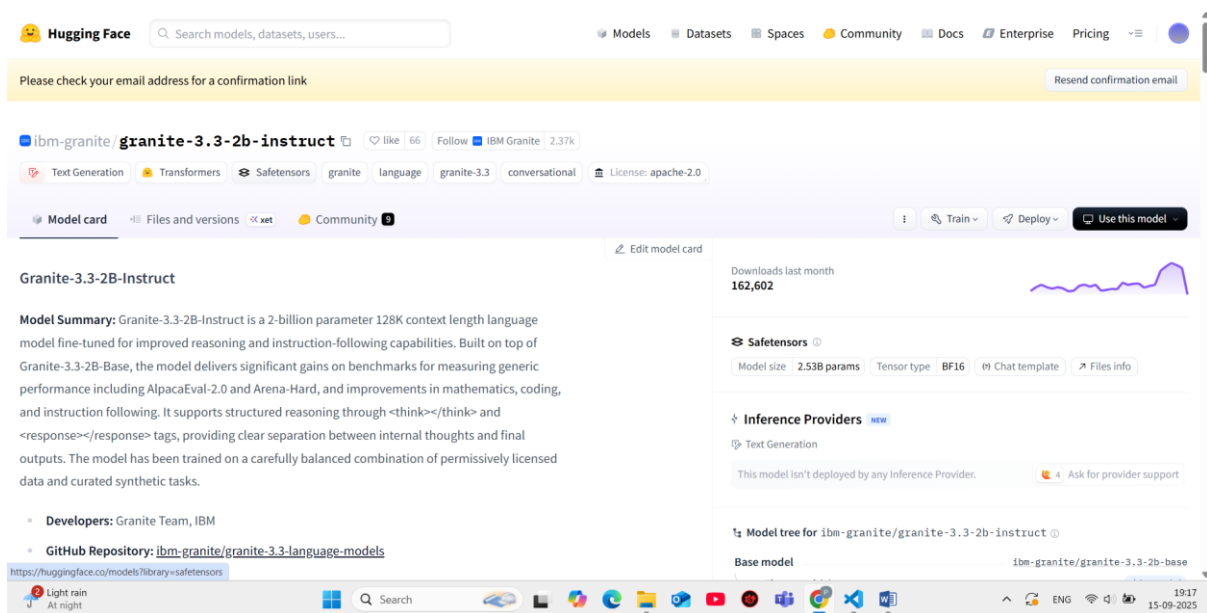
- Visit the **Hugging Face** website.



- Create an account or log in.
- Search for **IBM Granite Models**.



- Choose a suitable model (e.g., granite-3.2-2b-instruct) for this project.



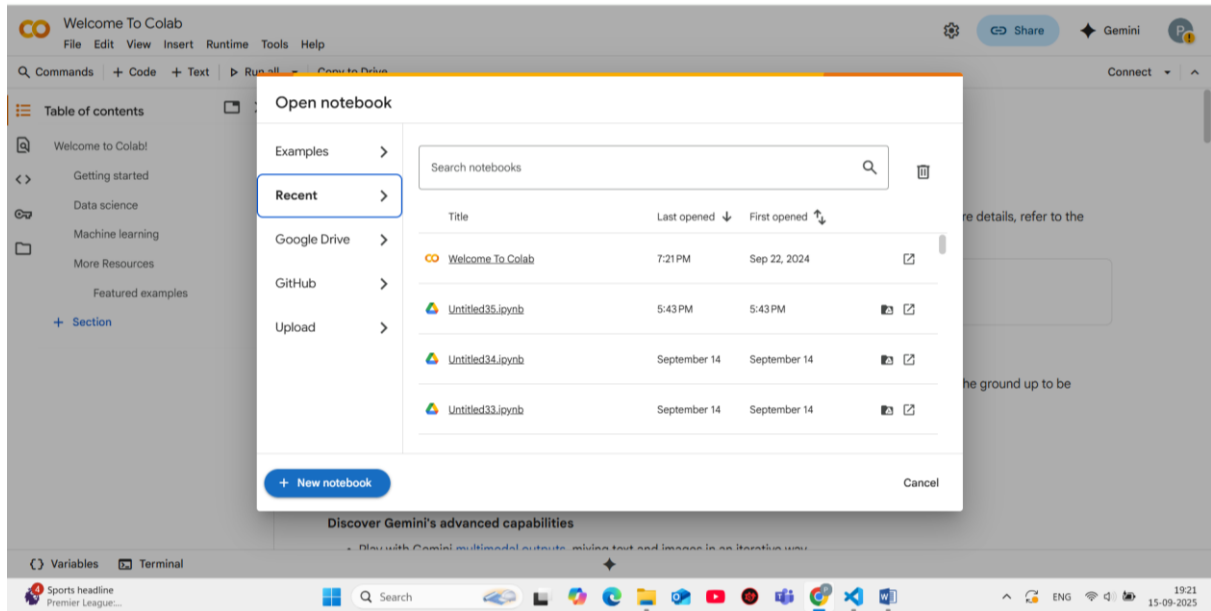
- This model balances **speed, accuracy, and lightweight performance**, making it perfect for medical AI tasks.

☐ *This step allows the project to be powered by a reliable AI model while remaining resource-efficient.*

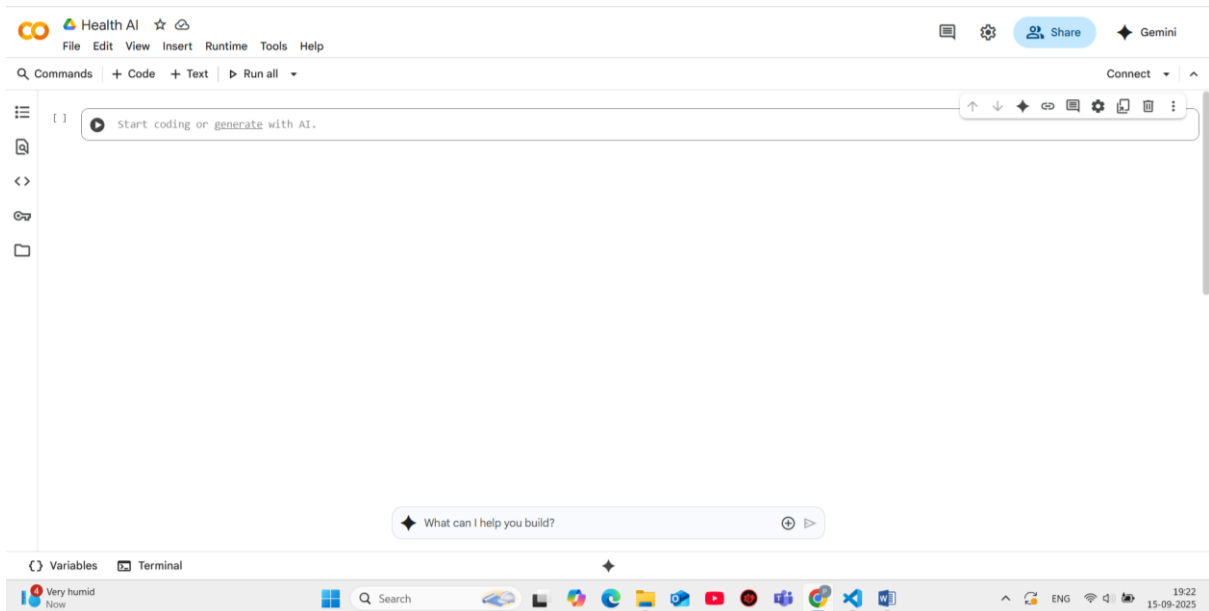
5. Implementation and Deployment

Activity 3: Running the Application in Google Colab

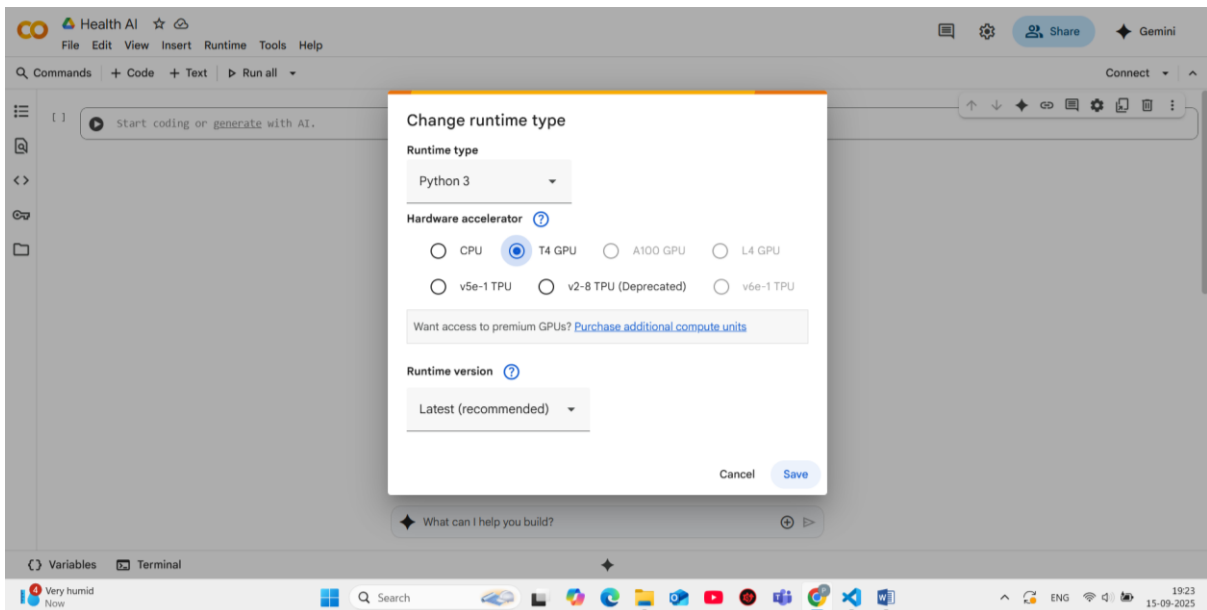
1. Open **Google Colab** and create a **new notebook**.



2. Rename the notebook to **Health AI**.

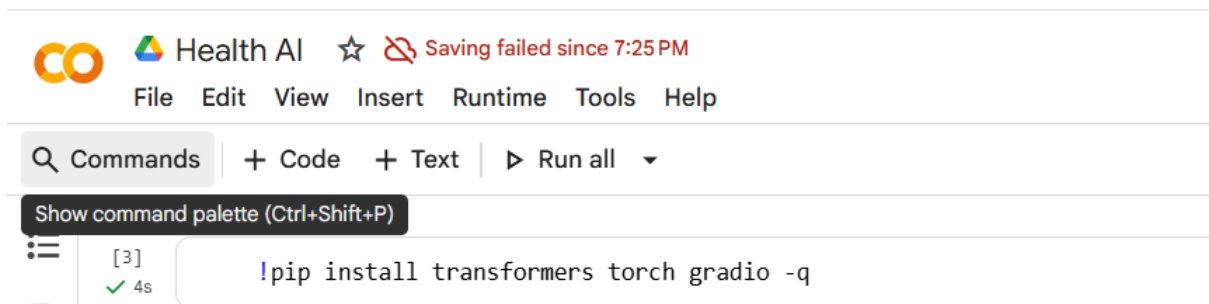


3. Go to Runtime → Change Runtime Type → Select **T4 GPU**.



4. Install required libraries using:
5. Copy and run the Health AI project code.

`!pip install transformers torch gradio -q`



Run the Health AI Code.




```
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

def disease_prediction(symptoms):
    prompt = f"Based on the following symptoms, provide possible medical conditions and general medication suggestions. Always emphasize the importance of consulting a doctor for\n\n"
    return generate_response(prompt, max_length=1200)

def treatment_plan(condition, age, gender, medical_history):
    prompt = f"Generate personalized treatment suggestions for the following patient information. Include home remedies and general medication guidelines.\n\nMedical Condition: {\n\n"
    return generate_response(prompt, max_length=1200)

# Create Gradio interface
with gr.Blocks() as app:
    gr.Markdown("# Medical AI Assistant")
    gr.Markdown("**Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.**")

    with gr.Tabs():
        with gr.TabItem("Disease Prediction"):
            with gr.Row():
                with gr.Column():
                    symptoms_input = gr.Textbox(
                        label="Enter Symptoms",
                        placeholder="e.g., fever, headache, cough, fatigue...",
                    )

                    predict_btn = gr.Button("Analyze Symptoms")

                with gr.Column():
                    prediction_output = gr.Textbox(label="Possible Conditions & Recommendations", lines=20)

            predict_btn.click(disease_prediction, inputs=symptoms_input, outputs=prediction_output)

        with gr.TabItem("Treatment Plans"):
            with gr.Row():
                with gr.Column():
                    condition_input = gr.Textbox(
                        label="Medical Condition",
                        placeholder="e.g., diabetes, hypertension, migraine...",
                        lines=2
                    )
                    age_input = gr.Number(label="Age", value=30)
                    gender_input = gr.Dropdown(
                        choices=["Male", "Female", "Other"],
                        label="Gender",
                        value="Male"
                    )
                    history_input = gr.Textbox(
                        label="Medical History",
                        placeholder="Previous conditions, allergies, medications or None",
                        lines=3
                    )
```

Output:

6. Once executed, a **Gradio** link will appear.

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: <https://575c0e9b9ca25d1e64.gradio.live>

7. Open the link to interact with the **Health AI chatbot**.

Medical AI Assistant

Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.

Disease Prediction

Treatment Plans

Enter Symptoms

Fever, headache, cough, fatigue...

Analyze Symptoms

Possible Conditions & Recommendations

The symptoms presented—fever, headache, cough, and fatigue—are common to various infectious diseases. These can include:

- Influenza (Flu)**: Often characterized by sudden onset, fever, severe headache, body aches, and extreme fatigue. Cough may or may not be present.
- Common Cold**: Typically starts gradually with a milder fever, sore throat, and nasal congestion. Cough usually develops later in the course of the illness.
- Pneumonia**: Can present with fever, chills, severe cough, and difficulty breathing. May require antibiotic treatment.
- COVID-19**: Symptoms can range from mild to severe, but often include fever, cough, and fatigue. Some may also experience headache and gastrointestinal issues.

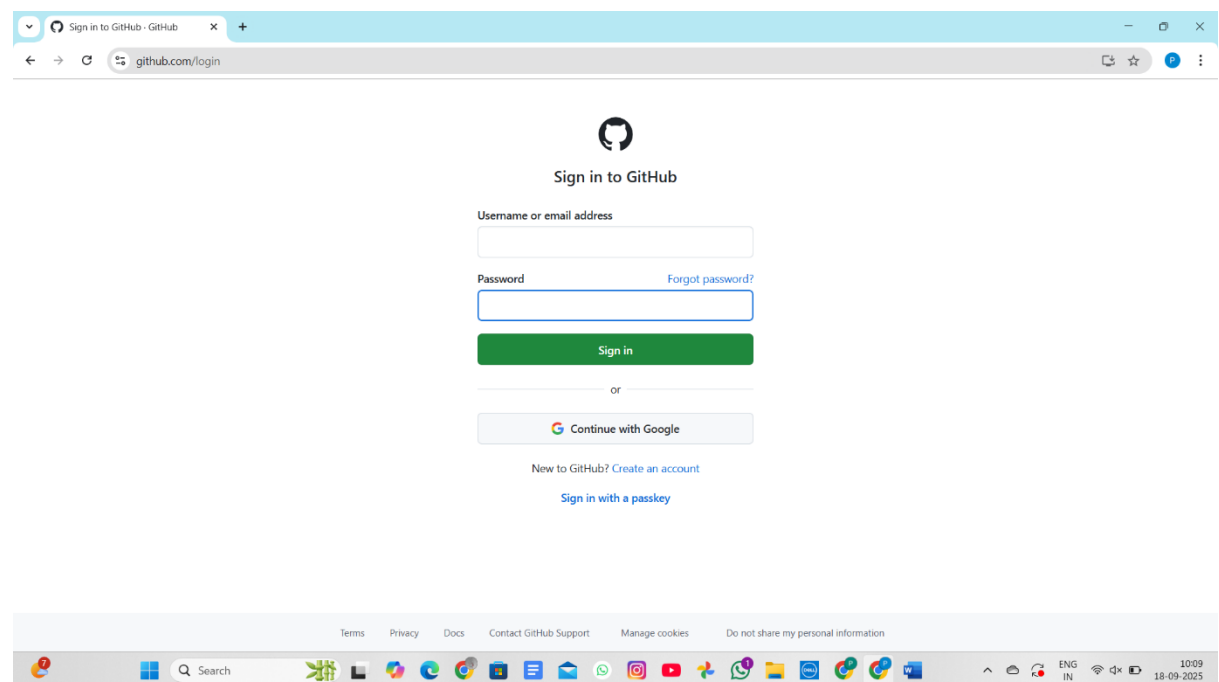
Medication suggestions (general and non-prescription):

- Acetaminophen (Tylenol) or ibuprofen (Advil, Motrin) for managing fever and pain.
- Over-the-counter decongestants or antihistamines for nasal congestion relief.
- Plenty of fluids to stay hydrated.
- Rest is crucial for recovery.

□ *At this stage, the system can accept patient queries, analyze symptoms, and provide guidance.*

Activity 4: Uploading the Project to GitHub

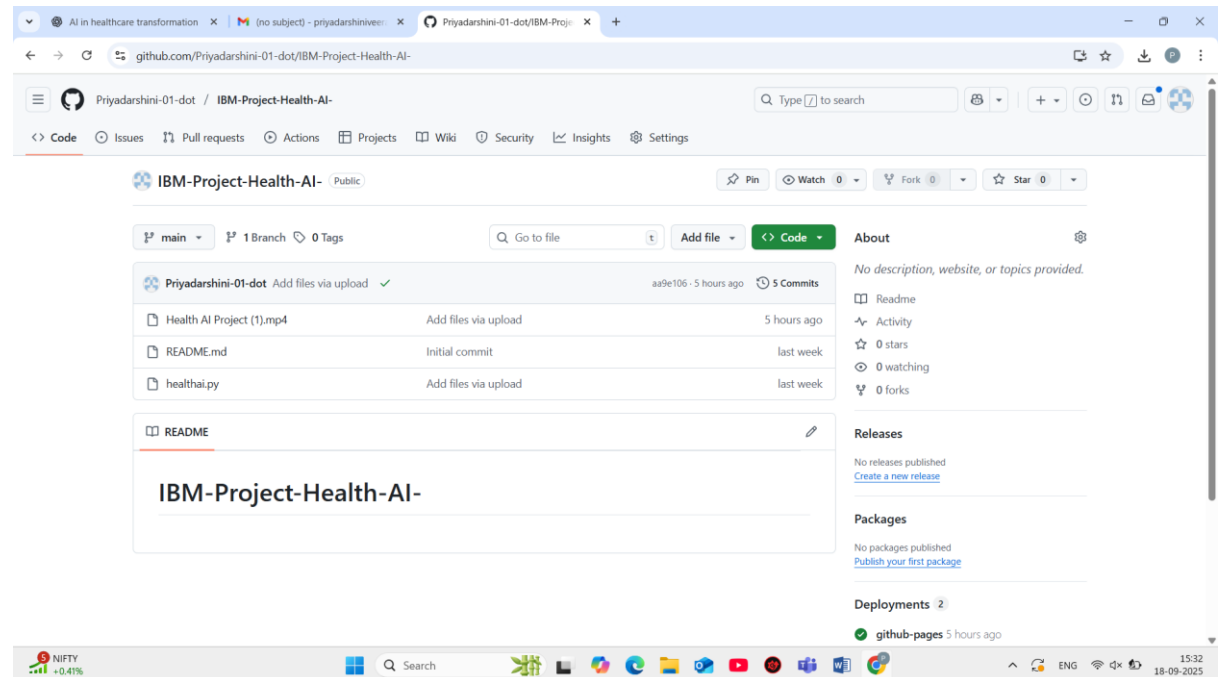
1. Go to **GitHub** and sign in (or create a new account).



2. Create a **new repository** (e.g., Health-AI-IBM).

3. Enable the **Add README** option.

4. From Google Colab, download your notebook as a .py file.
5. Upload this file to the repository using **Add File → Upload Files**.



6. Commit changes to finalize.

☐ *This ensures the project is safely stored, version-controlled, and ready for collaboration or demonstration.*

6. Conclusion

Health AI with IBM demonstrates how **AI can transform healthcare services** by bridging the gap between patients and medical professionals.

Key takeaways:

- **Patient-friendly chatbot** improves accessibility to medical information.
- **AI-driven disease prediction** provides faster preliminary assessments.
- **Google Colab deployment** ensures cost-effectiveness.
- **GitHub integration** makes the project transparent, collaborative, and easy to maintain.

Future Enhancements

- Multi-language support for diverse patient communities.
- Integration with **wearable devices** (e.g., smartwatches, health monitors).

- Advanced **predictive analytics dashboards** for doctors.
- Real-time **telemedicine integration** with healthcare providers.