Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**

MRI Impression Generation

Introduction

The project focuses on developing an assistive tool that supports doctors as well as radiologists in generating consistent and time-efficient MRI impressions. The system is not meant to replace medical expertise but to act as a supportive layer that reduces repetitive work and ensures accuracy.

Platform Selection

For the live deployment of the MRI Impression Assistant, we chose Hugging Face Spaces as our hosting platform. This decision was made for a few key reasons. First, its integration with the Hugging Face Hub, where our fine-tuned BioBART model is stored, is seamless.

- Second, Spaces offers native support for Streamlit applications, which meant we wouldn't have to manage complex server configurations.
- Finally, its generous free tier provided a perfect, cost-effective environment for hosting and sharing our academic project. This allowed us to focus on the application's functionality rather than on complex cloud infrastructure.

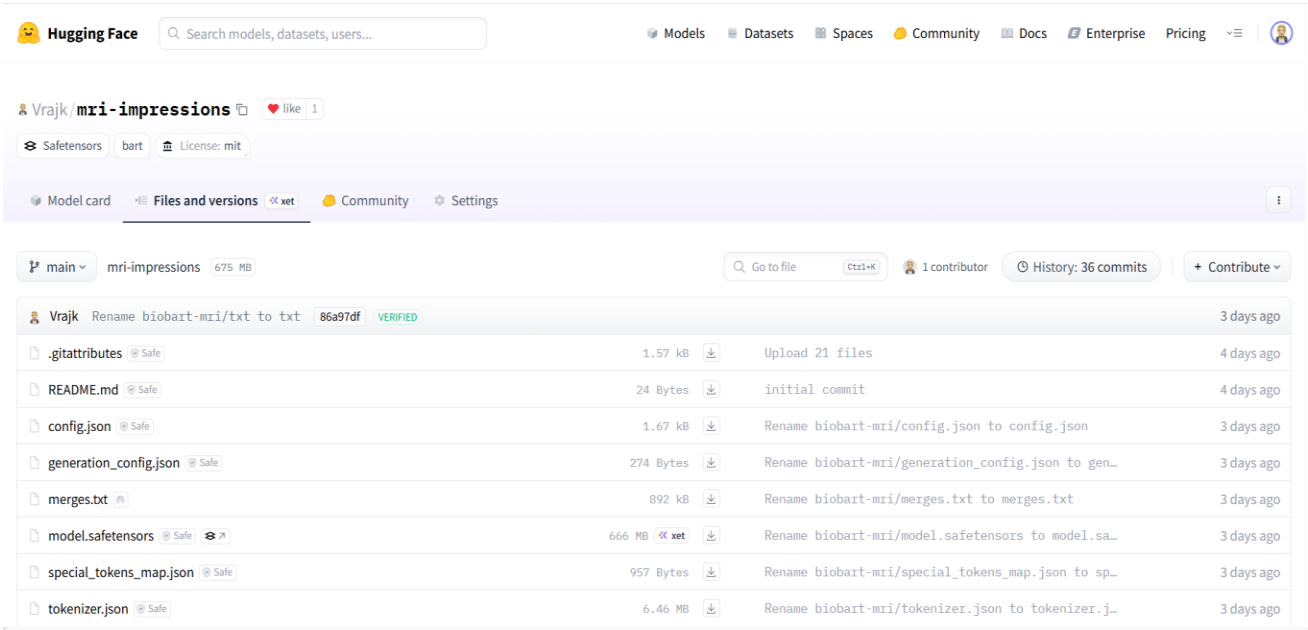
Deployment Architecture

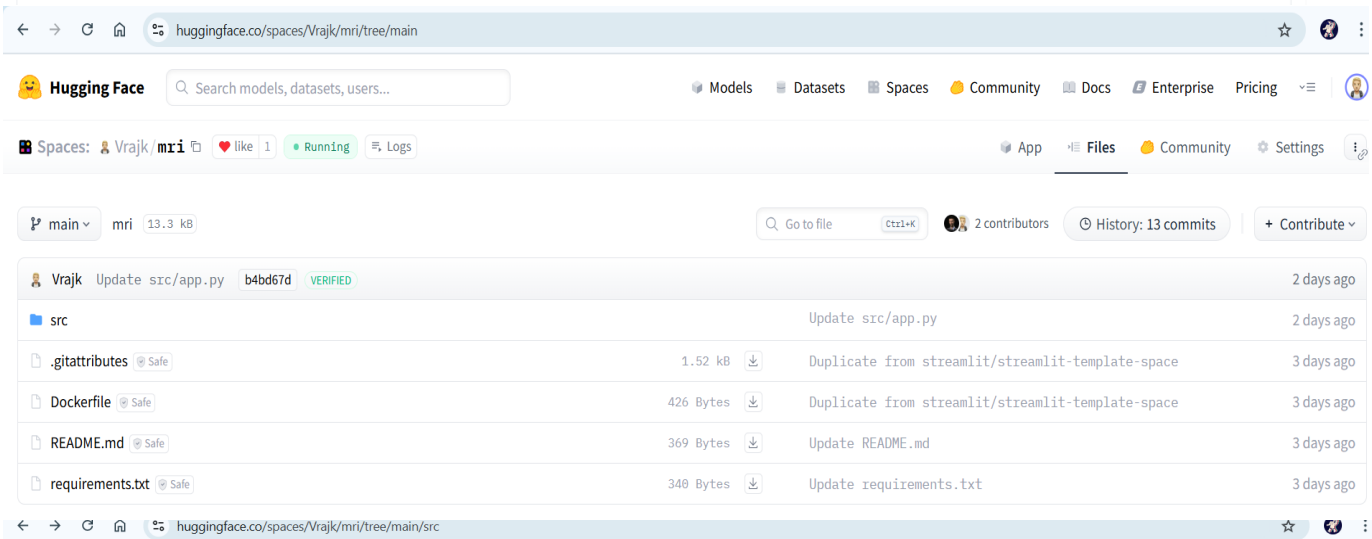
Our deployment strategy was built on the principle of separating the large model from the application code.

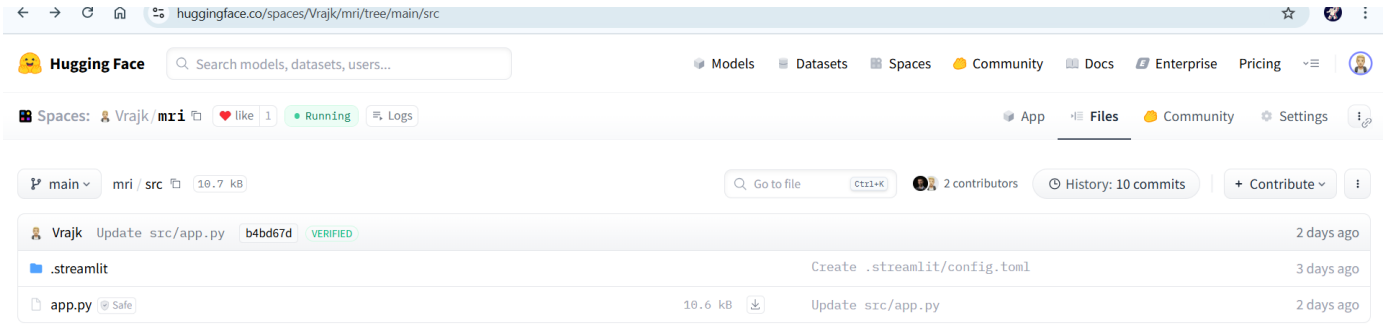
- Model : Our fine-tuned BioBART model, which is approximately 675 MB, is hosted on its own Hugging Face Model Repository. This keeps the model's files organized and version-controlled.
- Code : Our application code, including the app.py script and the requirements.txt file, is hosted in a separate Hugging Face Space.

When the application starts, it dynamically downloads the model from the Hub. This hybrid architecture ensures that our application is lightweight and that the large model is managed efficiently.

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**







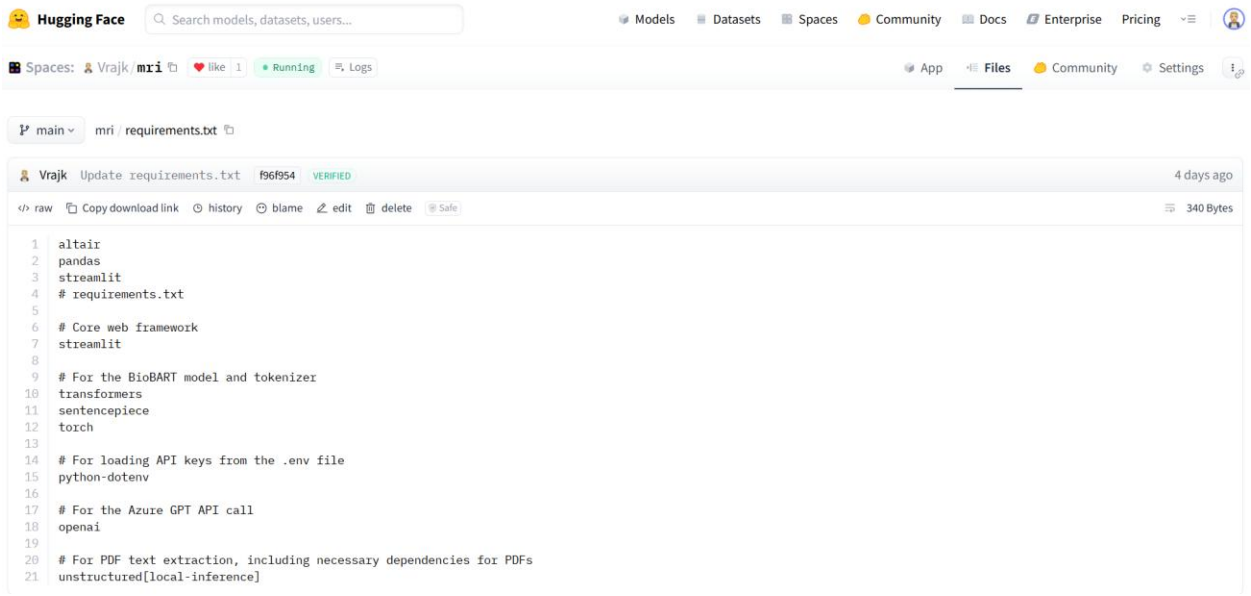
Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**

Configuration and Deployment Steps

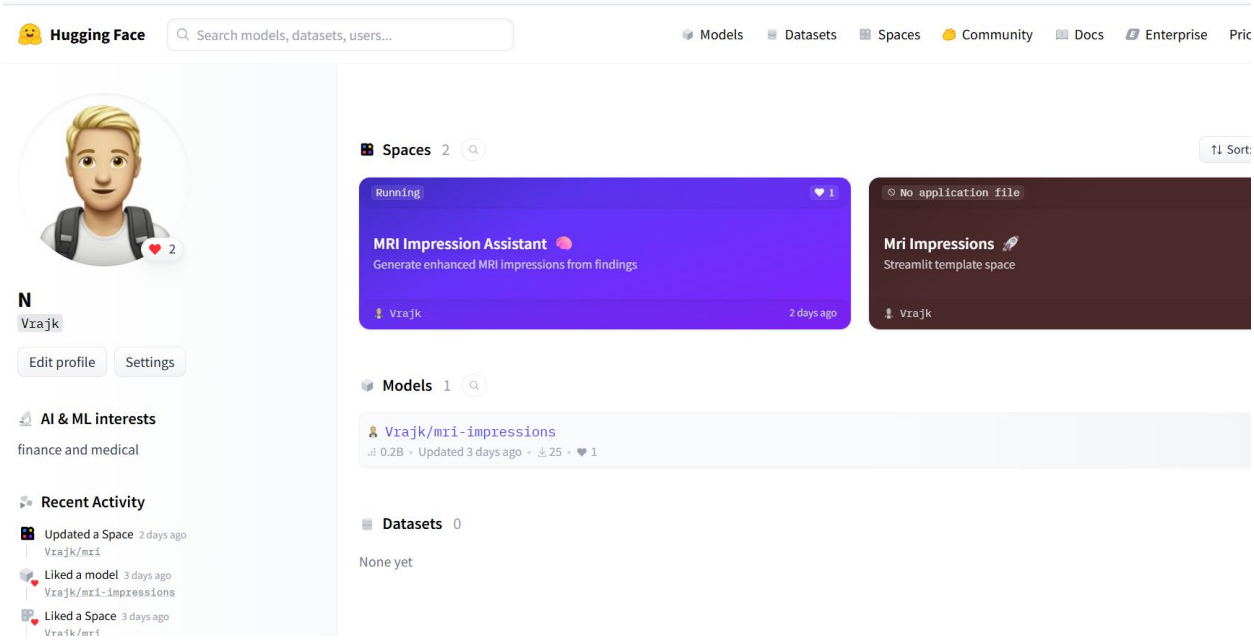
The deployment was an iterative process that involved several key steps to get the application running smoothly in the cloud environment:

1. Model Preparation: The first step was to upload the trained model from our local system to the Hugging Face Hub using Git and Git LFS (Large File Storage) to handle the large model.safetensors file.

2. Code Preparation: We created the final app.py script, ensuring it loaded the model from our Hugging Face repository (Vrajk/mri-impressions) instead of a local path. We also created a requirements.txt file to list all necessary libraries.

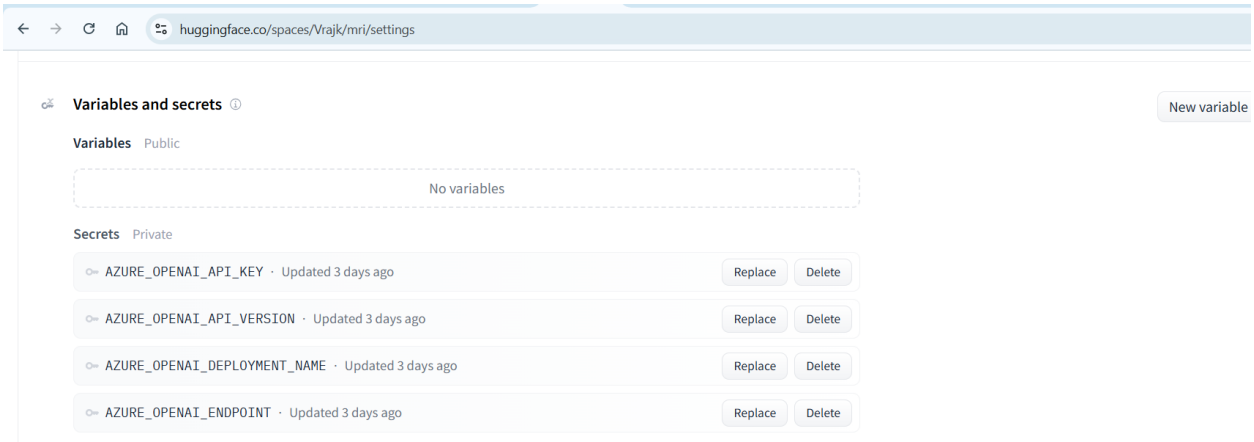


3. Space Creation: We created a new Hugging Face Space, making sure to select "Streamlit" as the SDK to ensure the correct environment was prepared for our app.



4. File Upload: We uploaded our app.py (renaming it from our local file), requirements.txt, and a special .streamlit/config.toml file to the Space.

5. Configuration & Secrets: We securely added our Azure OpenAI API keys to the "Repository secrets" section in the Space's settings. This keeps our keys safe and allows the deployed app to access them.

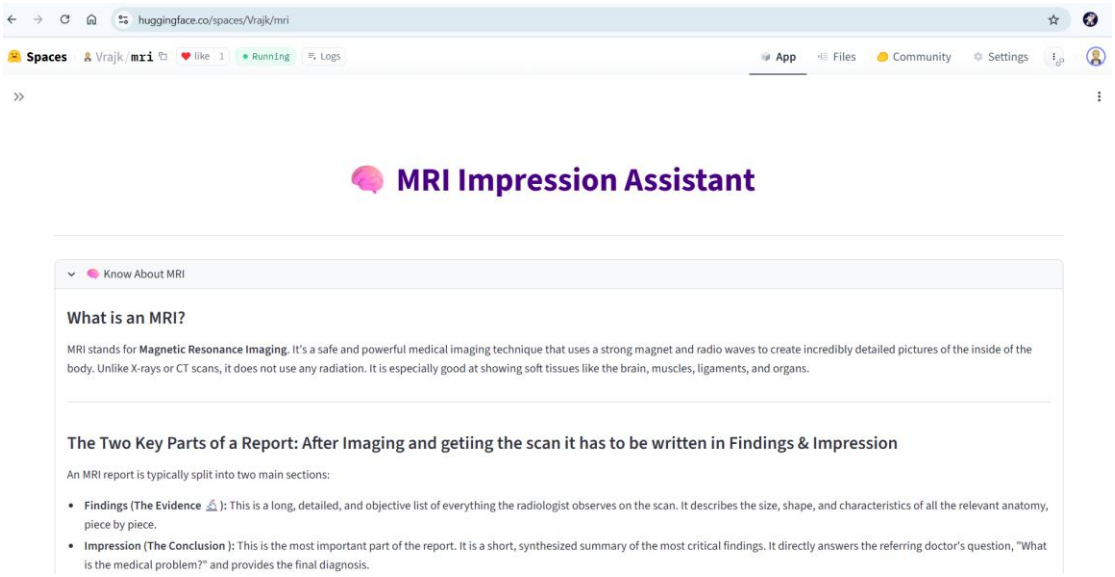


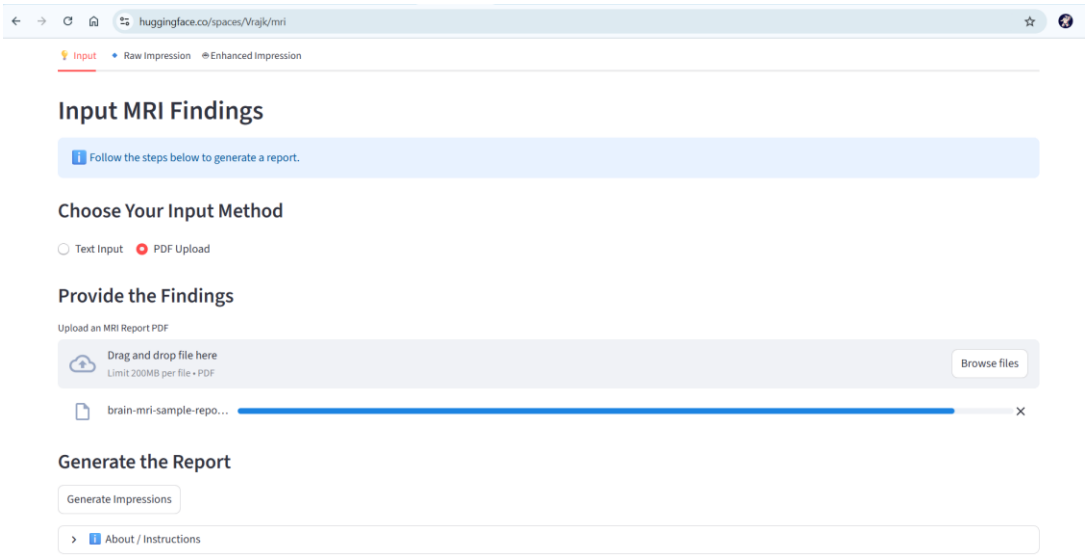
Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**

Live Deployment

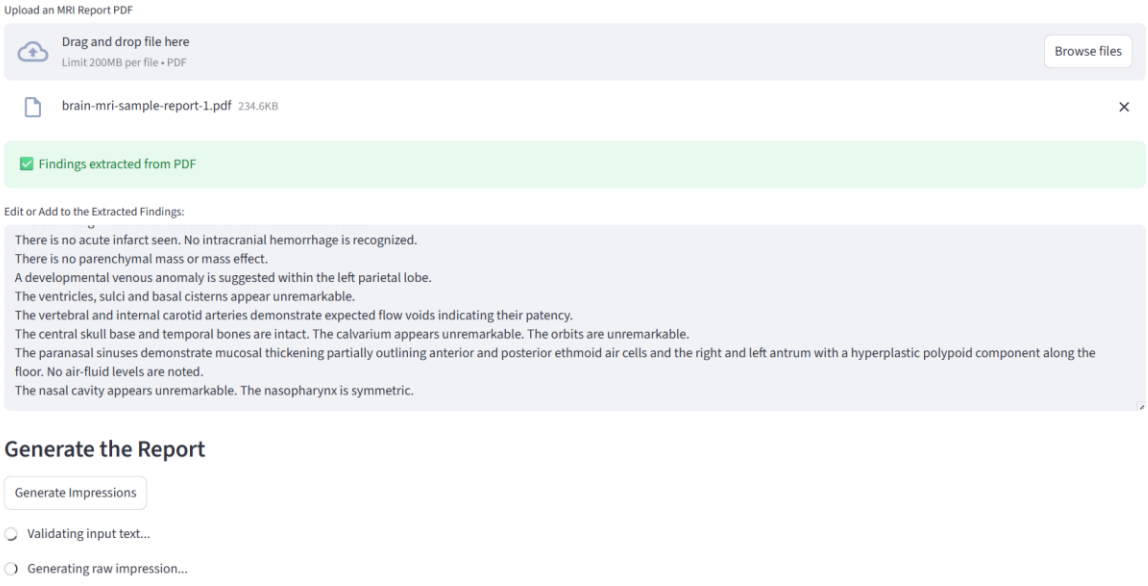
The MRI Impression Assistant is successfully deployed and publicly accessible.

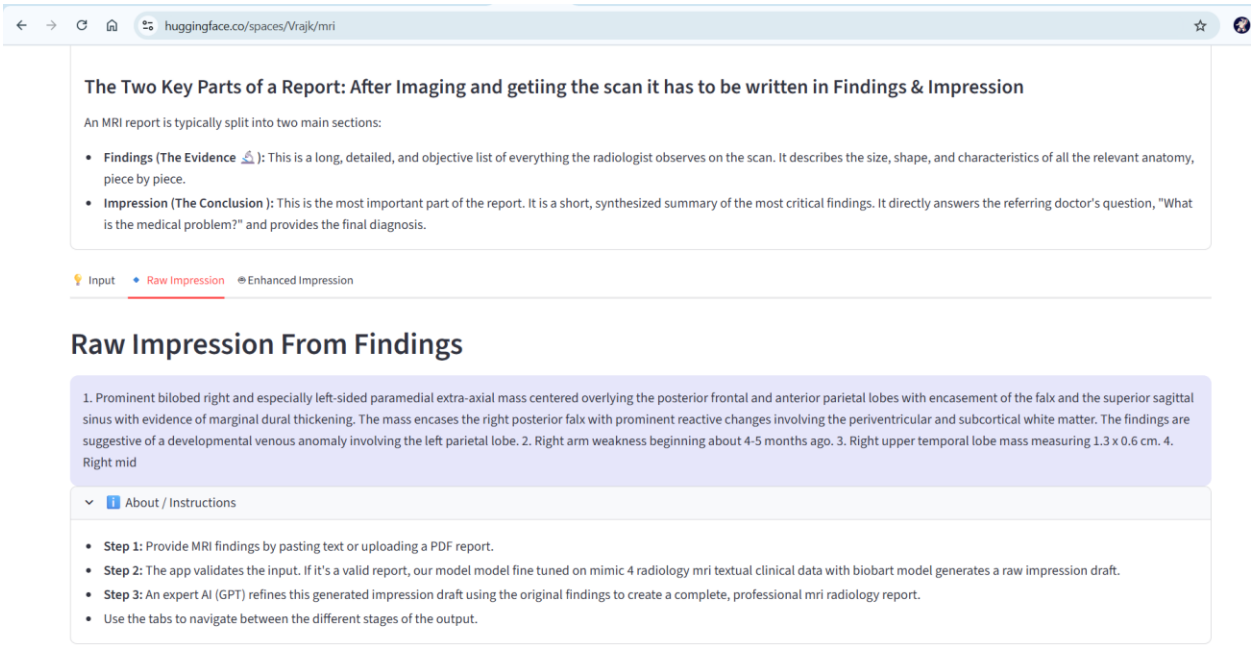
- Live URL: https://huggingface.co/spaces/Vrajk/mri
- Screenshots of Deployed App:



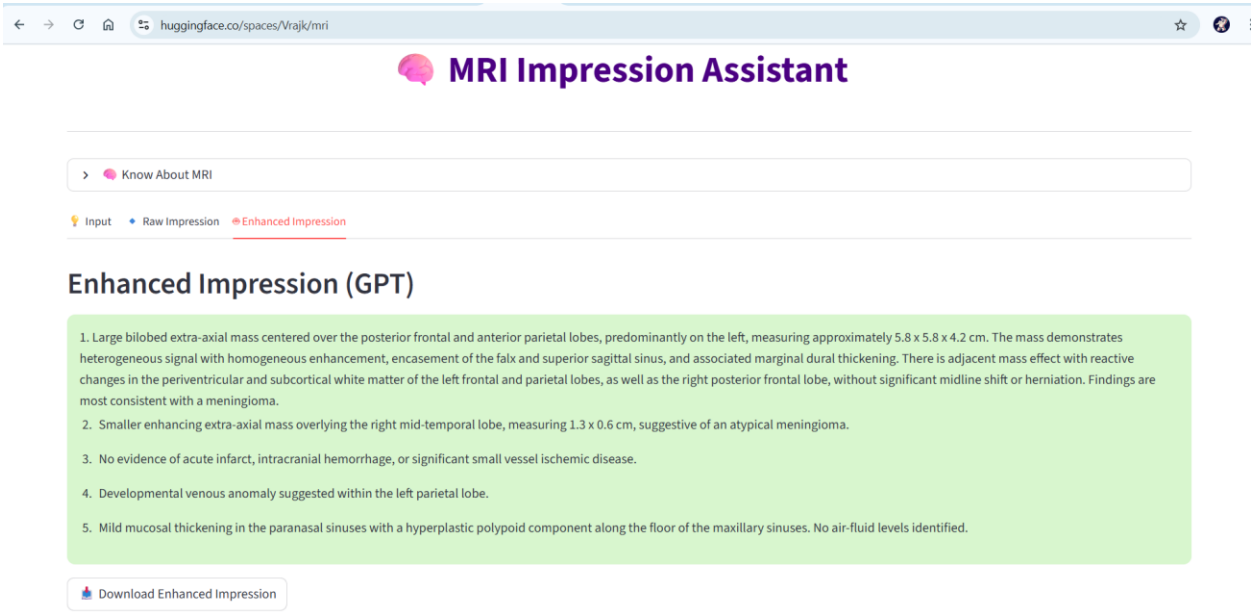


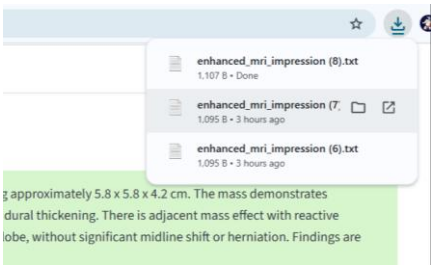
Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**

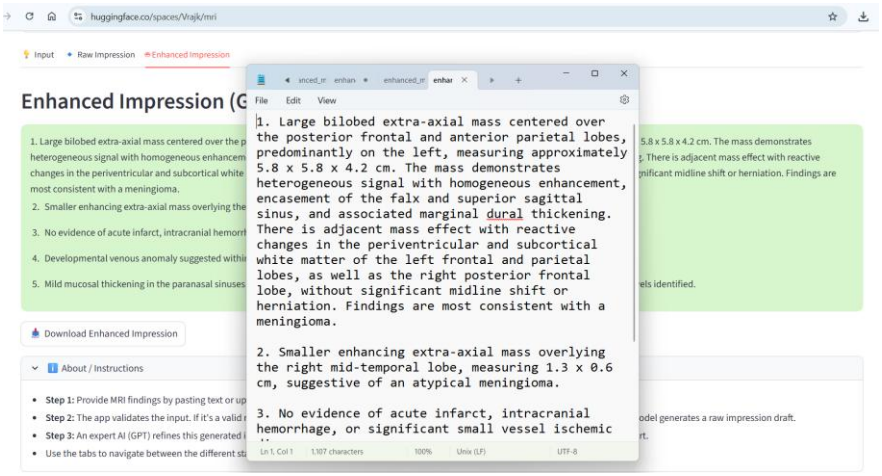




Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**







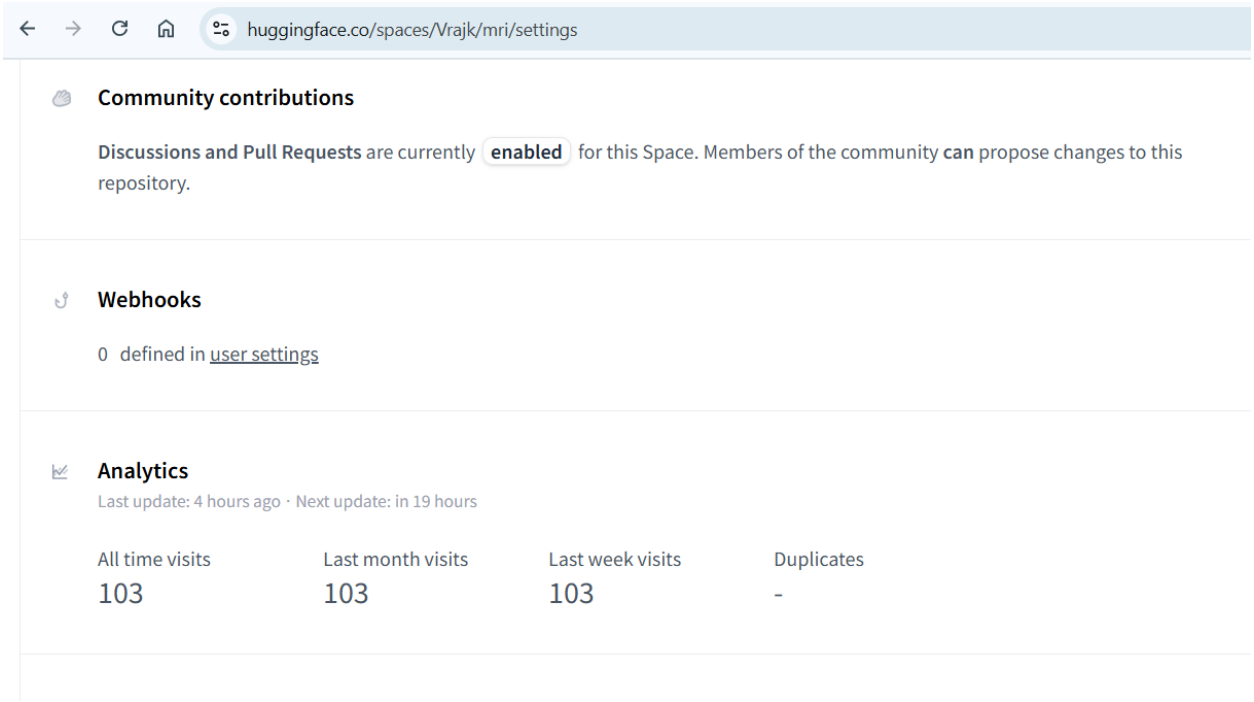
Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**

Monitoring Setup and Tools

To ensure the reliability and performance of the deployed MRI Impression Assistant, we implemented a monitoring strategy using the native tools provided by the Hugging Face Spaces platform. This approach allows for real-time insights without the need for external infrastructure.

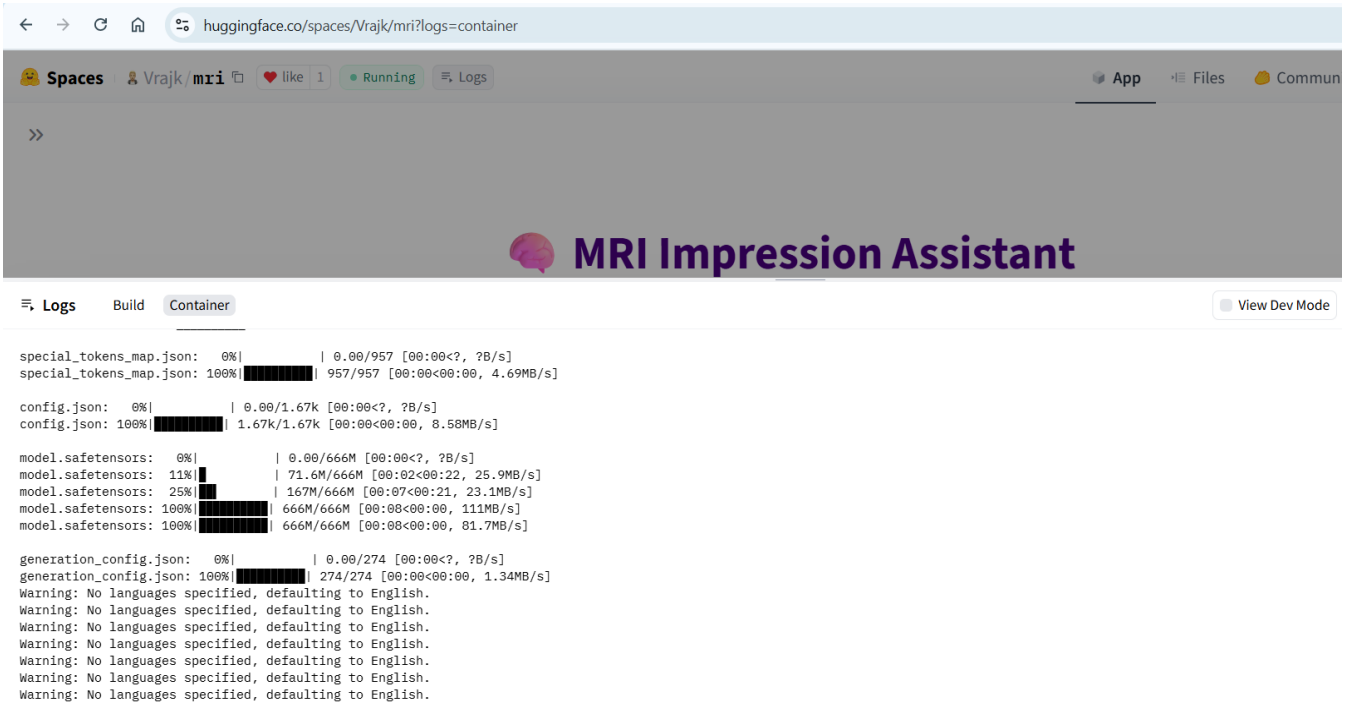
The two primary tools used are:

1. Hugging Face Spaces Analytics: A built-in dashboard that provides graphs and metrics for resource consumption.



2. Hugging Face Spaces Logs: A real-time stream of the application's console output, essential for tracking application health and debugging errors.

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**



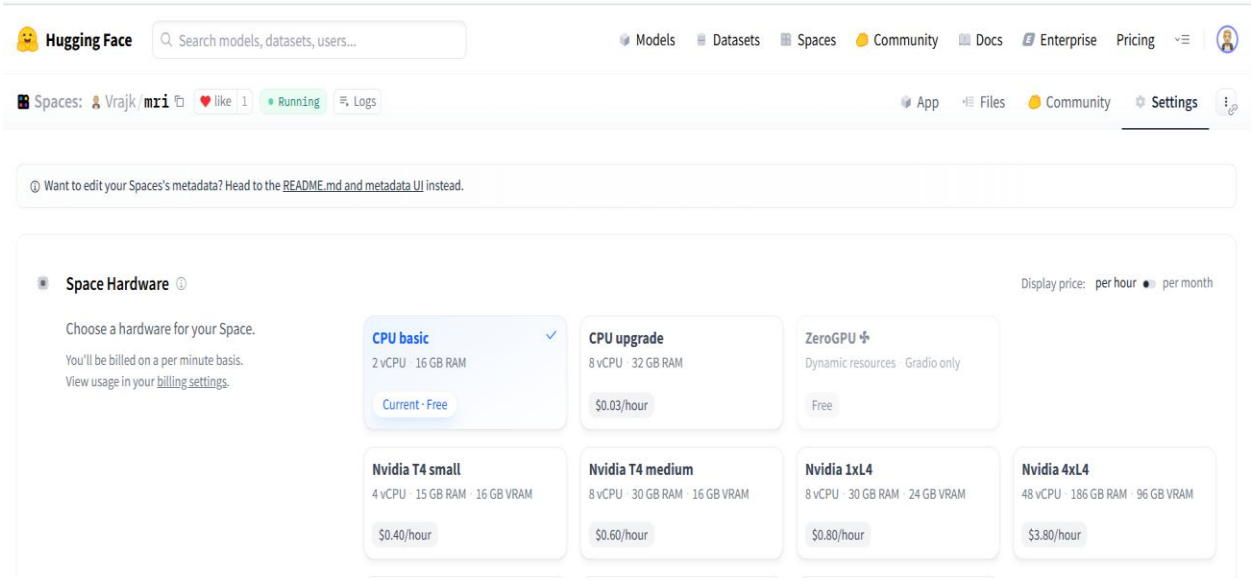
Key Performance Indicators (KPIs) & Monitoring

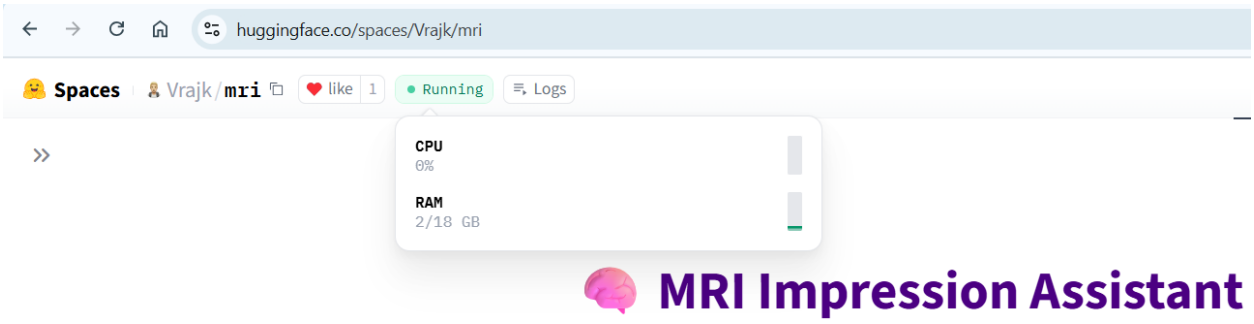
We identified three critical KPIs to monitor the operational performance and health of our application:

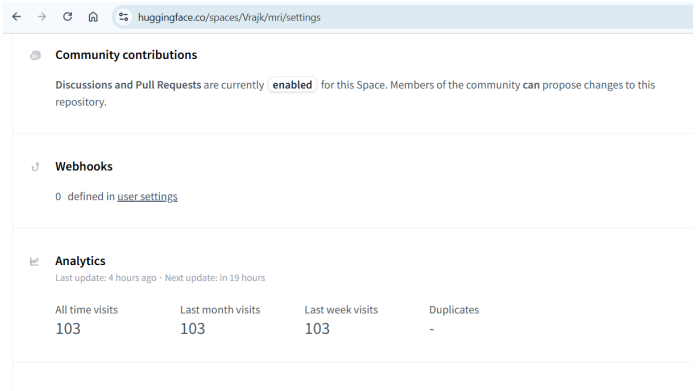
1. **Latency Monitoring (Response Time):** We measured inference latency manually during our testing phase by timing the duration from the button click to the final output.After doing the latency test on real time to generate first raw impression approx. 15 seconds it takes and after 3 seconds it gives of gpt as well , so it takes around 20 seconds , bu tit totally depends upon the users internet connectivity.

2. **CPU Utilization (%):** Monitored via the Analytics dashboard, this KPI shows how much processing power the application is using they give us free 16 gb to use and this application hardly requires 2 gb of ram out of 18 gb so this is easy to use for multiple interaction as well. And also the session handling is done by them toml file

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**







2 gb of ram is used out of 18 , one can easily give continuous input and process and generate the data

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project (01CT0715)	**Implementation- Continuous progress review**	
Implementation	**Date: 25/09/2025**	**Enrolment No: 92200133018**

Maintenance Plan

A maintenance plan is in place to ensure the long-term reliability, security, and performance of the MRI Impression Assistant. The plan is centered around three key areas:

- **Regular Updates:** Software dependencies listed in requirements.txt (like Streamlit and Transformers) will be updated on a regular schedule

  The core BioBART model will be updated as needed if retraining provides significant accuracy improvements.

- **Performance and Scalability:** We will monitor application performance (CPU and Memory usage) using the built-in Hugging Face Spaces analytics dashboard.

  If the application becomes slow due to high user traffic, the primary mitigation strategy is to upgrade the Space to more powerful hardware, which can be done directly in the settings.

- **Bug Fixes and Security:** Any bugs discovered in the application's code will be addressed by pushing fixes to the Space's repository, which triggers an automatic redeployment.

  All secret API keys are managed securely in the Space's encrypted settings and will be rotated periodically to maintain security.

This tool is deployed to a live cloud environment on **Hugging Face Spaces**, proving its viability as a real-world tool. With a clear plan for ongoing monitoring and maintenance, this project serves as a robust proof-of-concept for how Generative AI can be effectively utilized to enhance efficiency, consistency, and quality in the field of radiology.