In-class Assignment 2

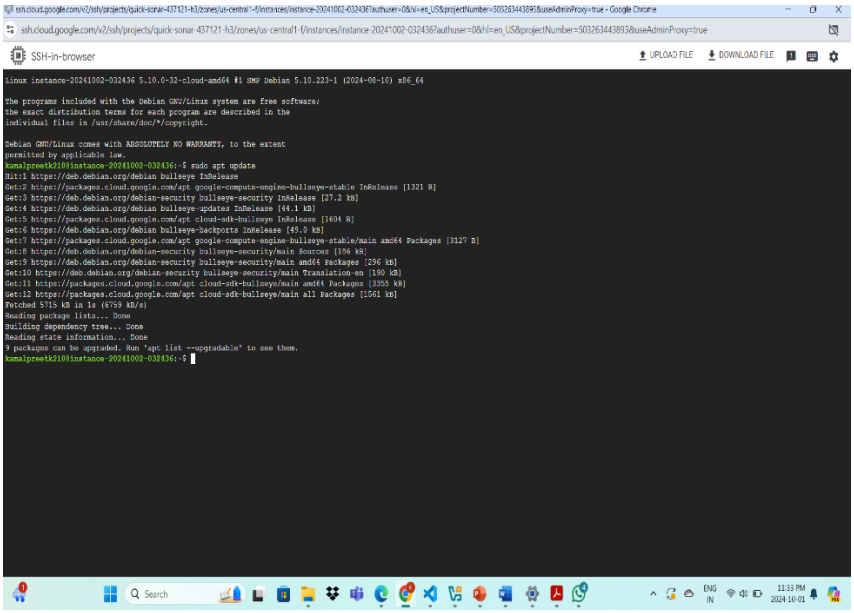
Instructor: Qasim A

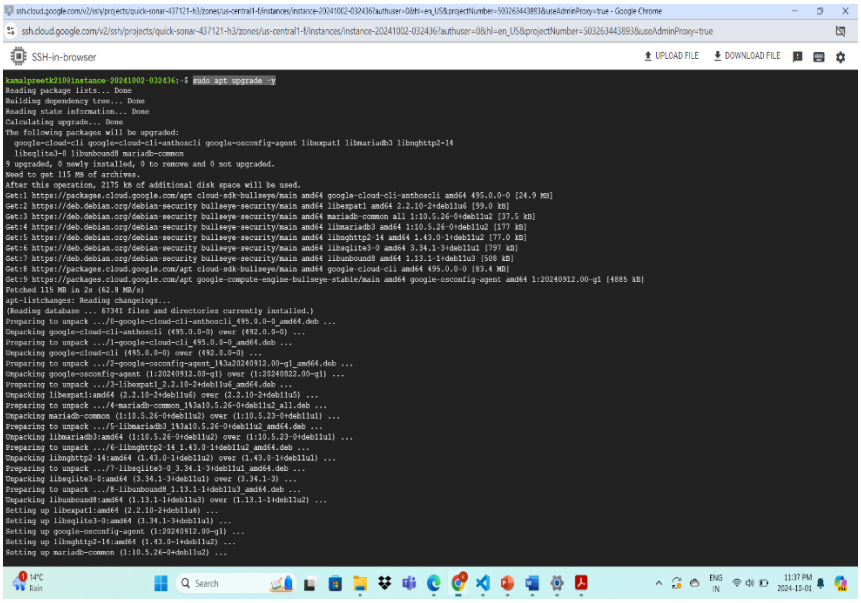
Develop and Deploy a Machine Learning Application using Docker

Name: Kamal Preet Kaur
Student ID: 500227884

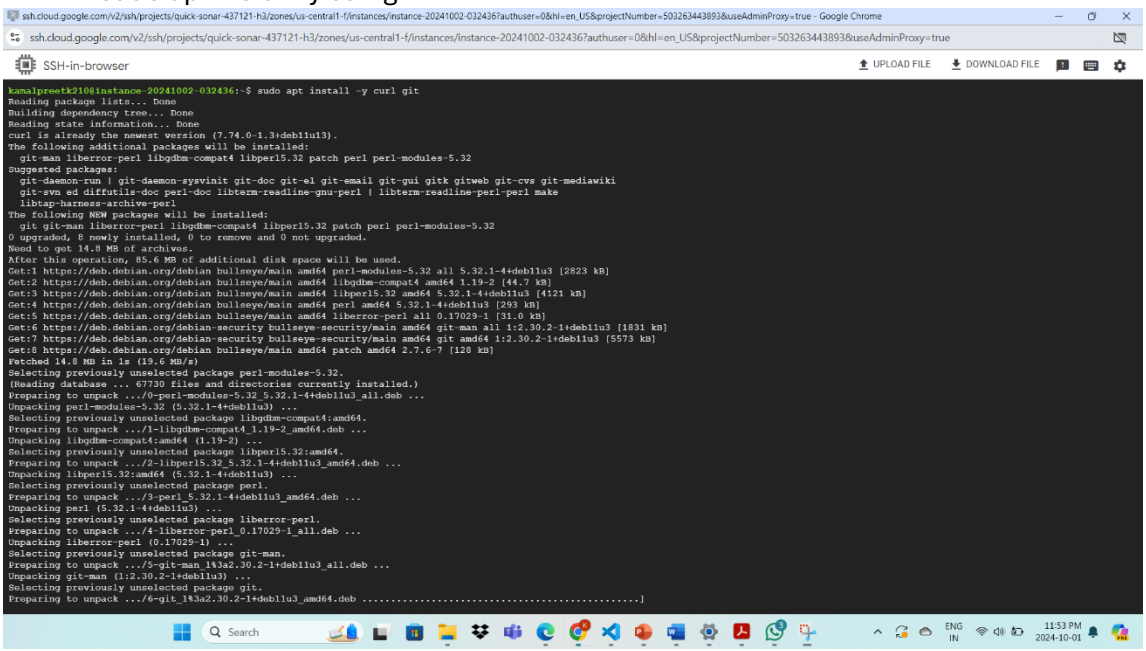
Step 1: Set Up the VM

1. **Update the System**
   sudo apt update sudo apt upgrade -y



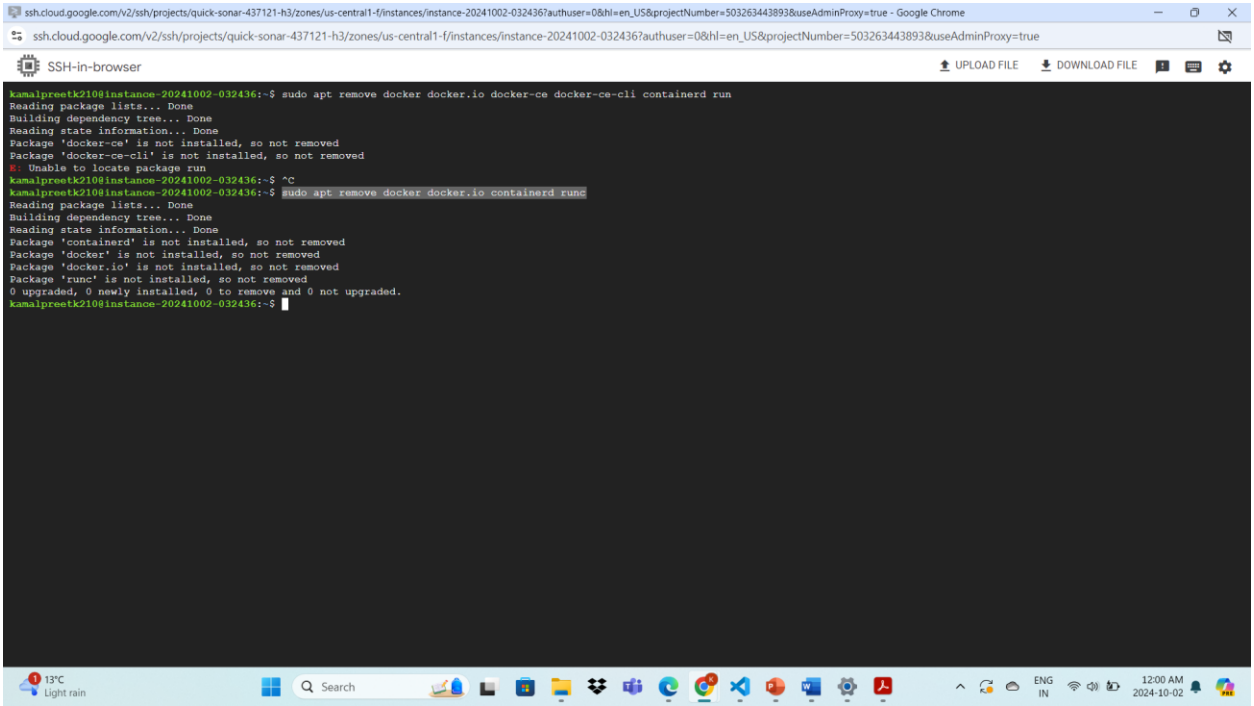


2. **Install Necessary Packages**
   sudo apt install -y curl git

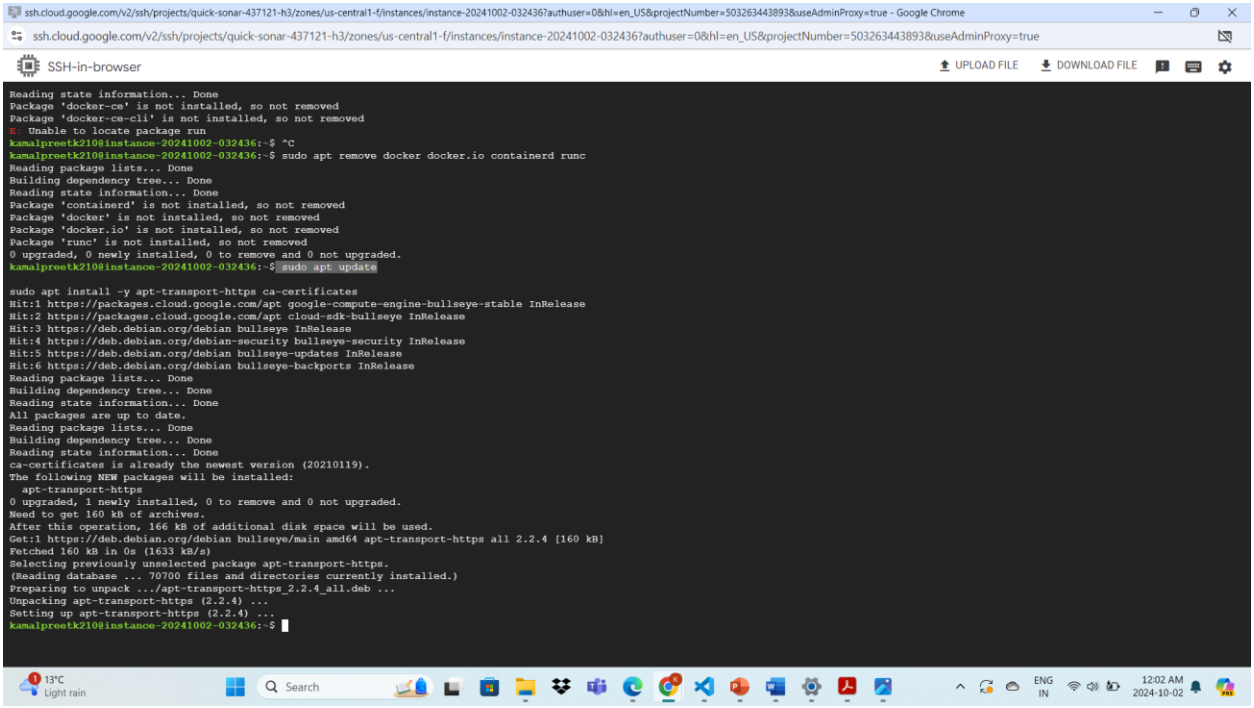


Step 2: Install Docker

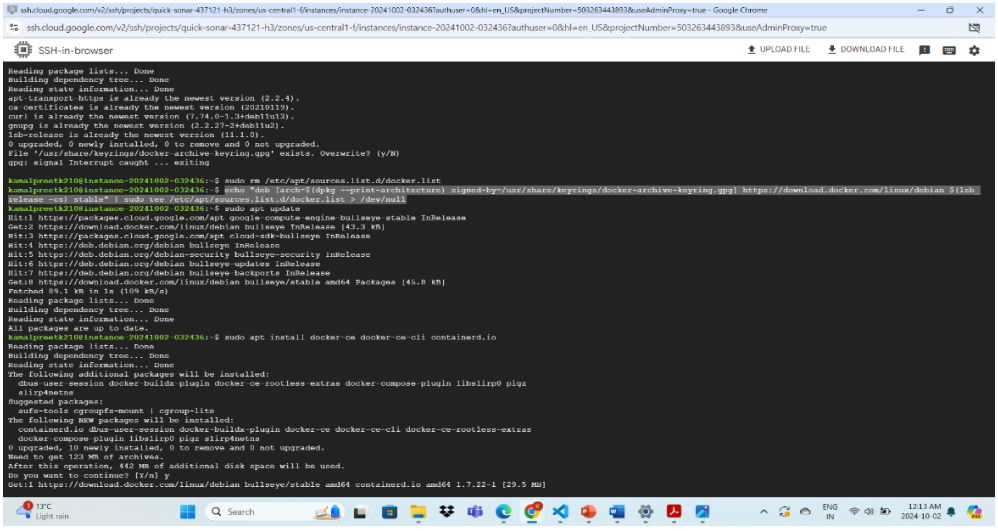
1. **Remove Old Versions:** sudo apt remove docker docker-engine docker.io containerd runc



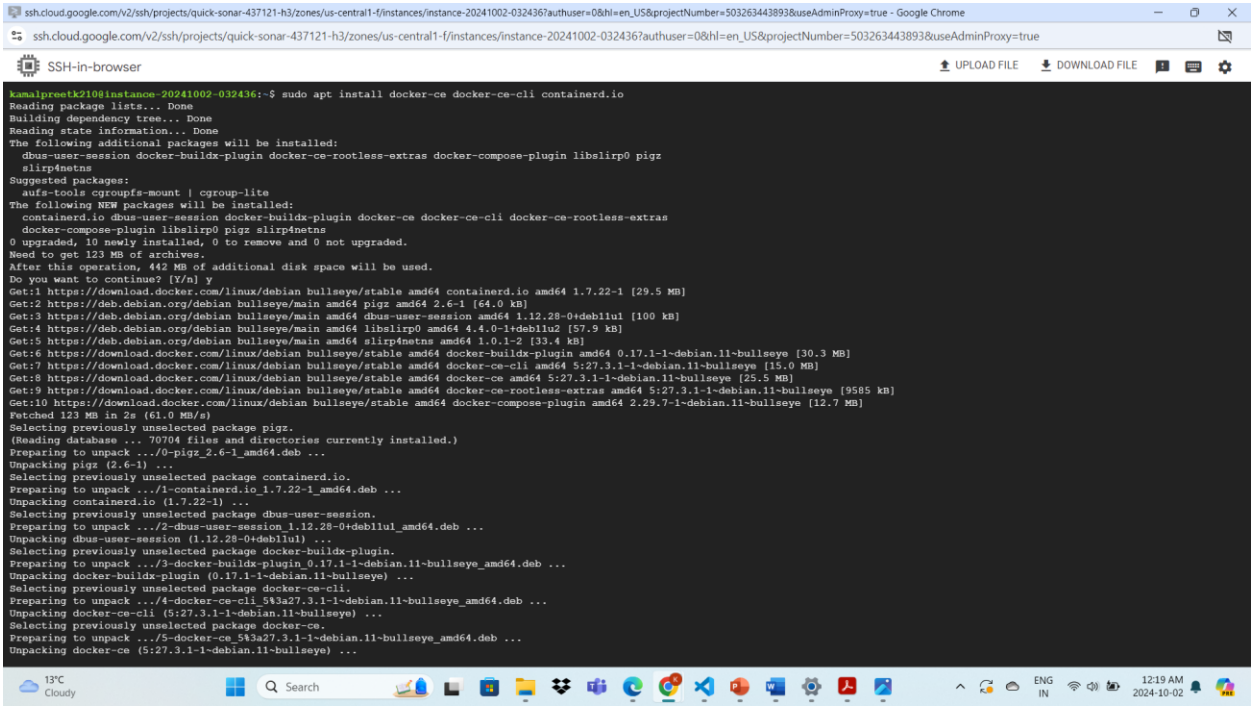
2. **Set Up the Docker Repository**
   sudo apt update
   sudo apt install -y apt-transport-https ca-certificates



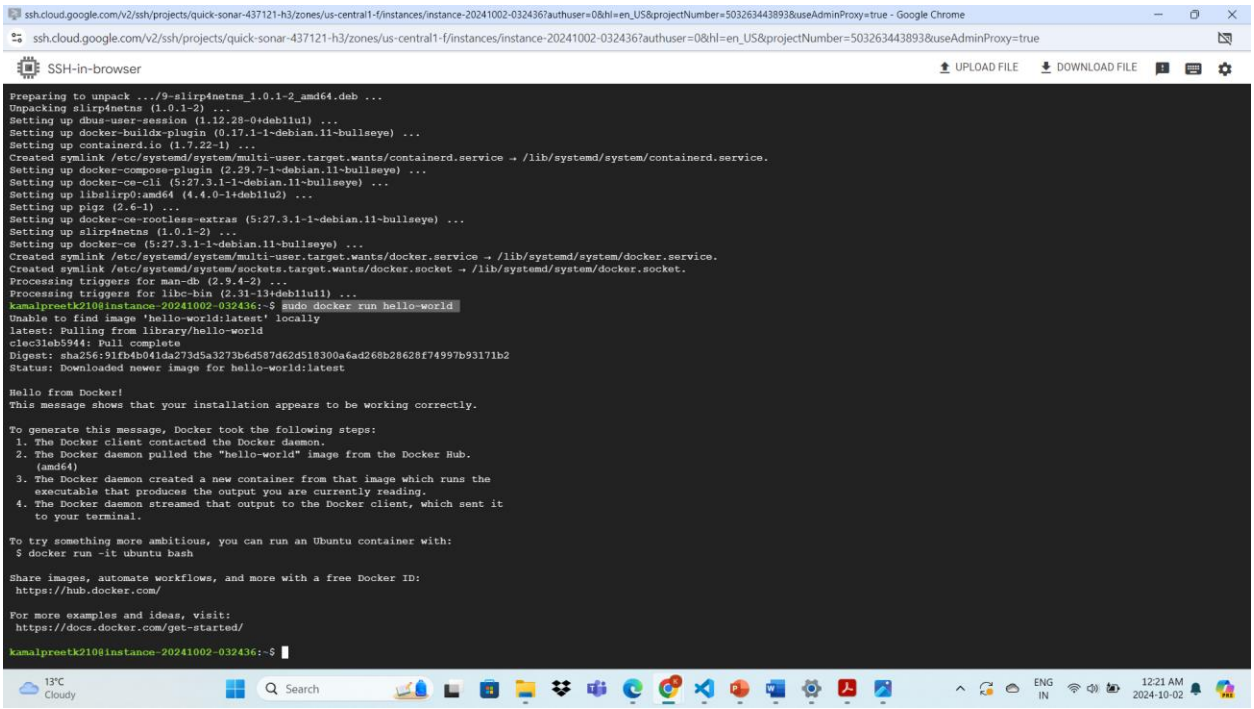
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee/etc/apt/sources.list.d/docker.list > /dev/null



3. **Install Docker Engine**
   sudo apt update sudo apt install -y docker-ce docker-ce-cli containerd.io

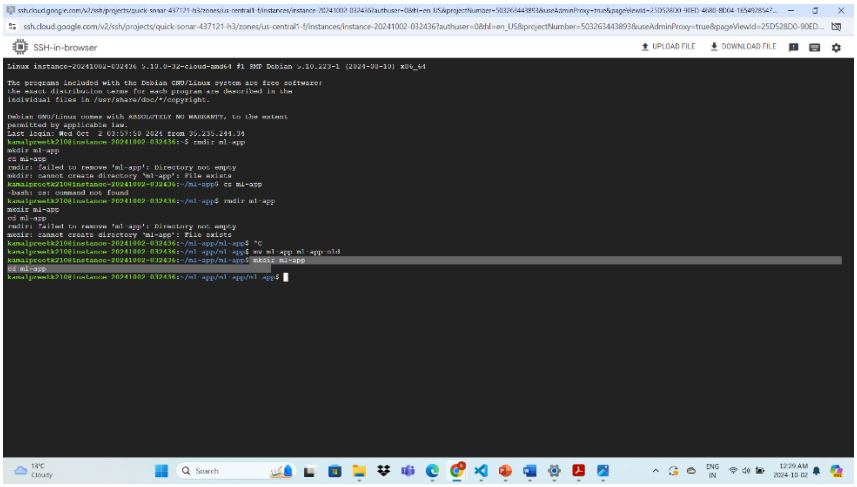


4. **Verify Docker Installation**
   sudo docker run hello-world

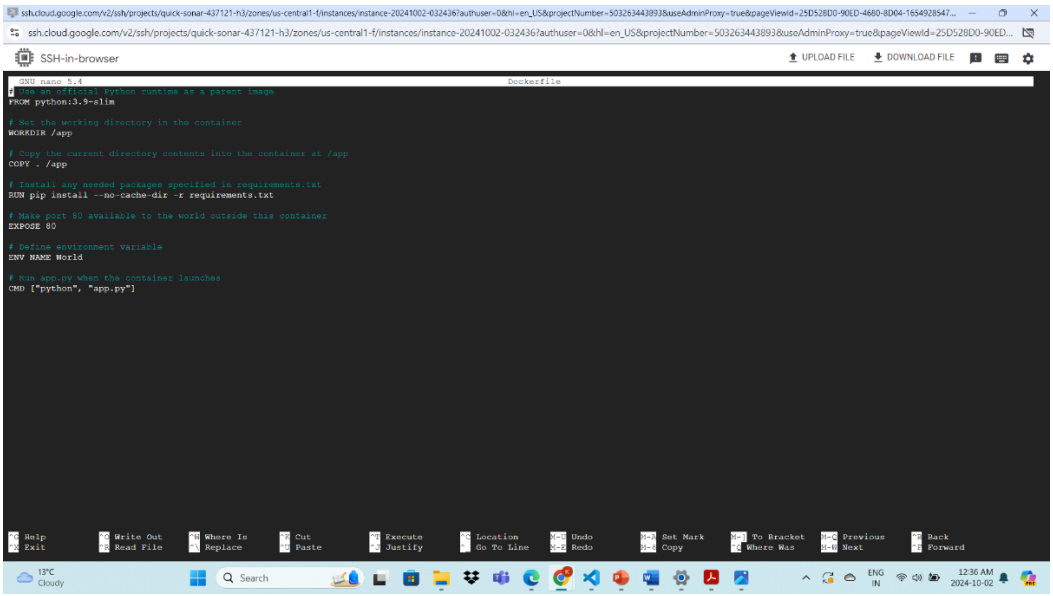


Step 3: Create a Dockerfile for the ML Application

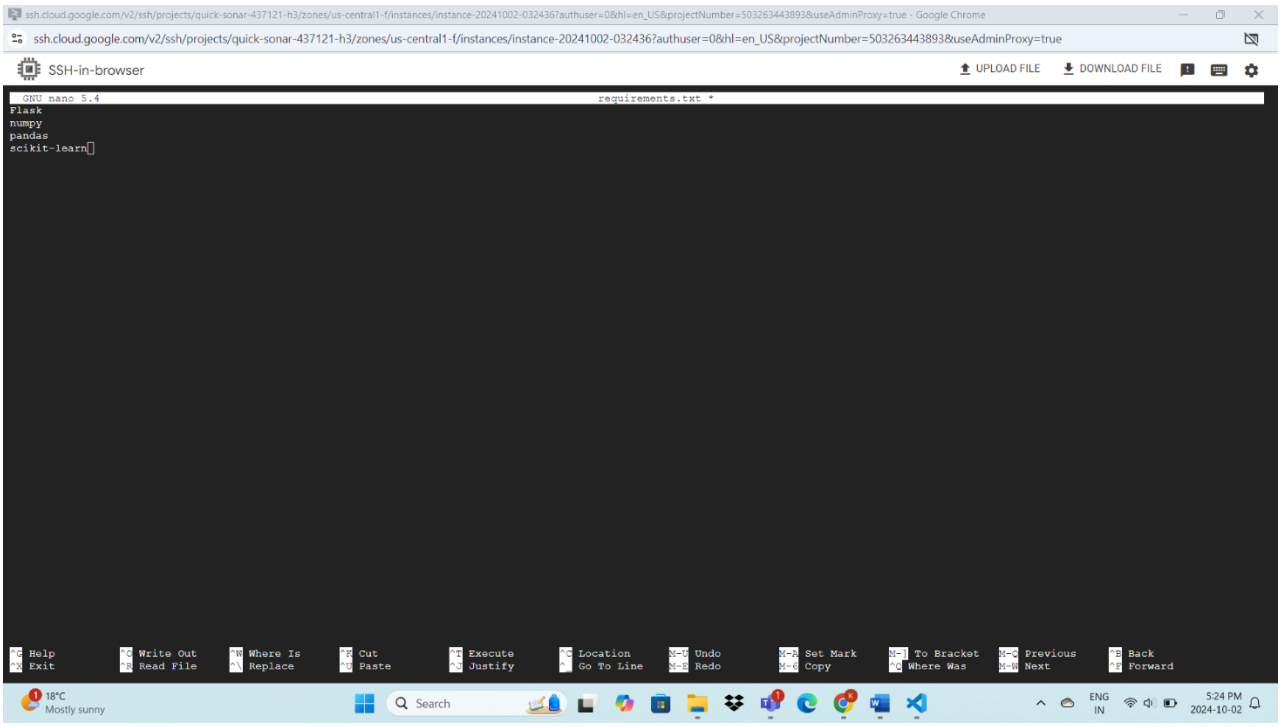
1. **Create Project Directory**
   mkdir ml-app cd ml-app



2. **Create a docker file**
   Create a Dockerfile with the following content:
   # Use an official Python runtime as a parent image
   FROM python:3.9-slim
   # Set the working directory
   WORKDIR /usr/src/app
   # Copy the current directory contents into the container at /usr/src/app
   # Install any needed packages specified in requirements.txt RUN pip install --no-cache-dir -r requirements.txt # Make port 80 available to the world outside this container EXPOSE 80
   Run app.py when the container launches CMD ["python", "app.py"]



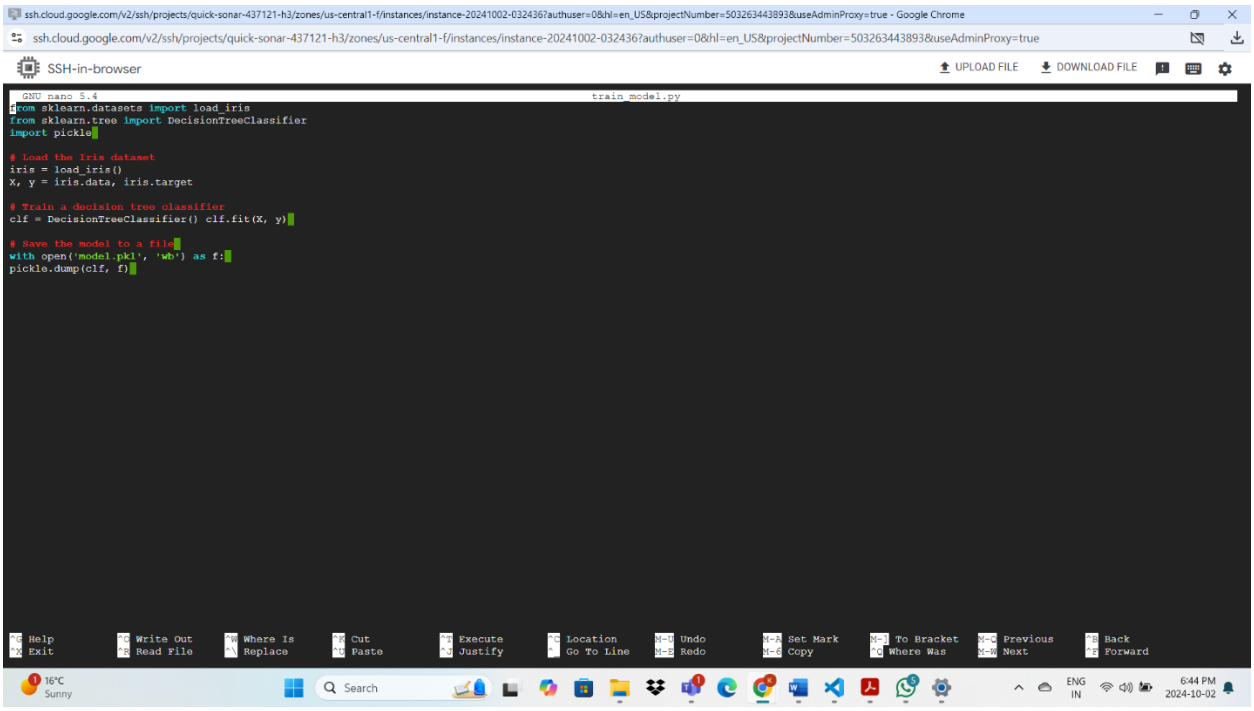
3. **Create requirements.txt File**



Step 4: Develop the Machine Learning Application

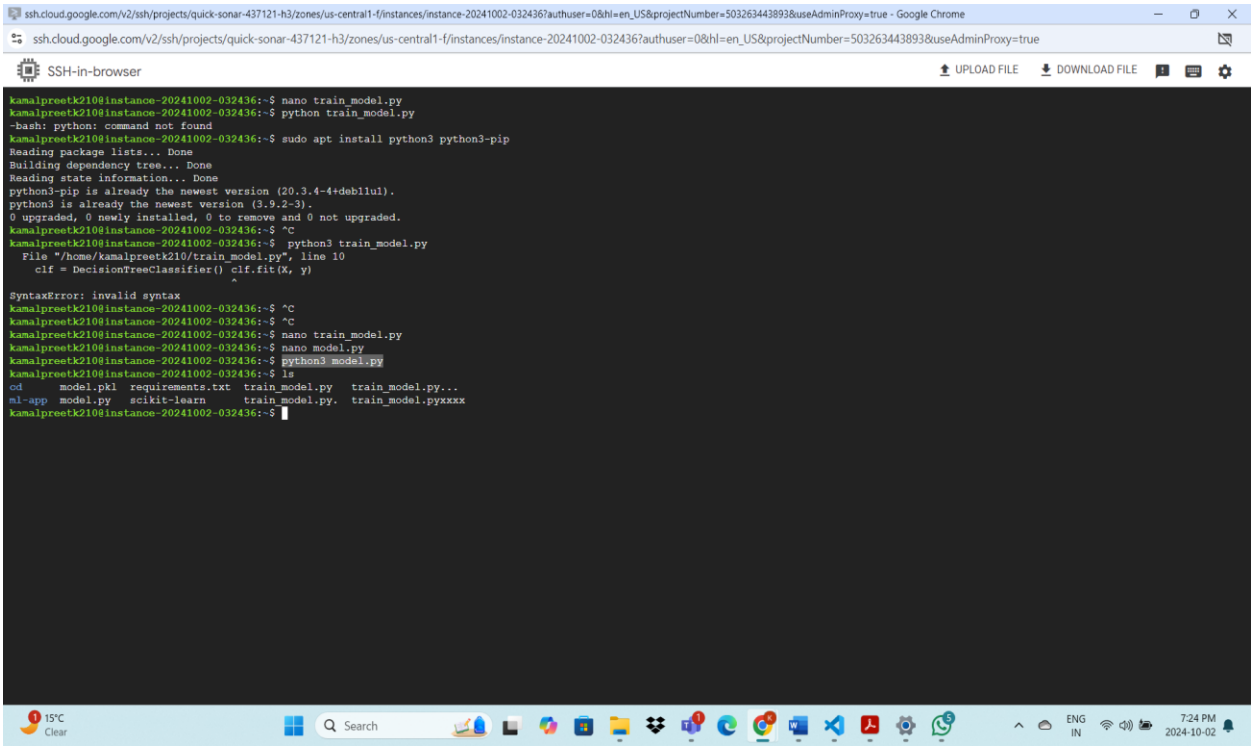
1. **Create a Simple ML Model**

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
import pickle

# Load the Iris dataset
iris = load_iris()
X, y = iris.data, iris.target

# Train a decision tree classifier
clf = DecisionTreeClassifier()
clf.fit(X, y)

# Save the model to a file
with open('model.pkl', 'wb') as f:
pickle.dump(clf, f)
```

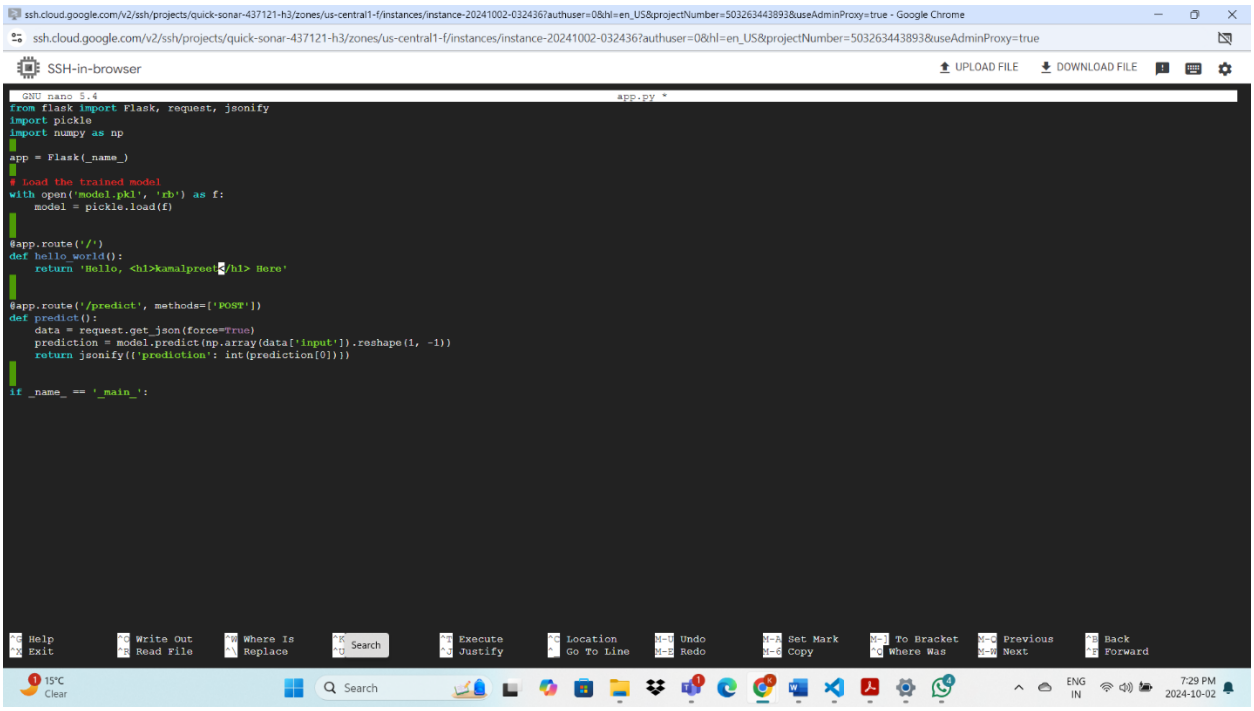


2. **Run the Model Training Script**
   python train_model.py

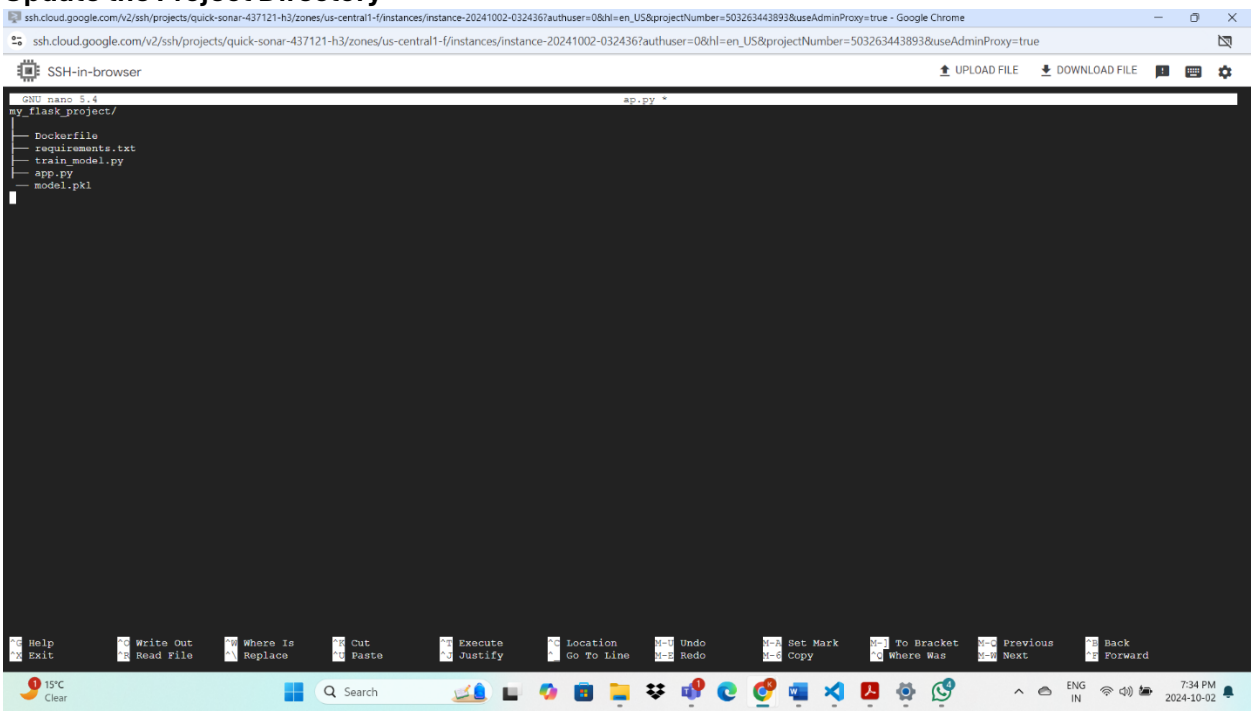
nano



3. **Integrate the Model into the Flask App**



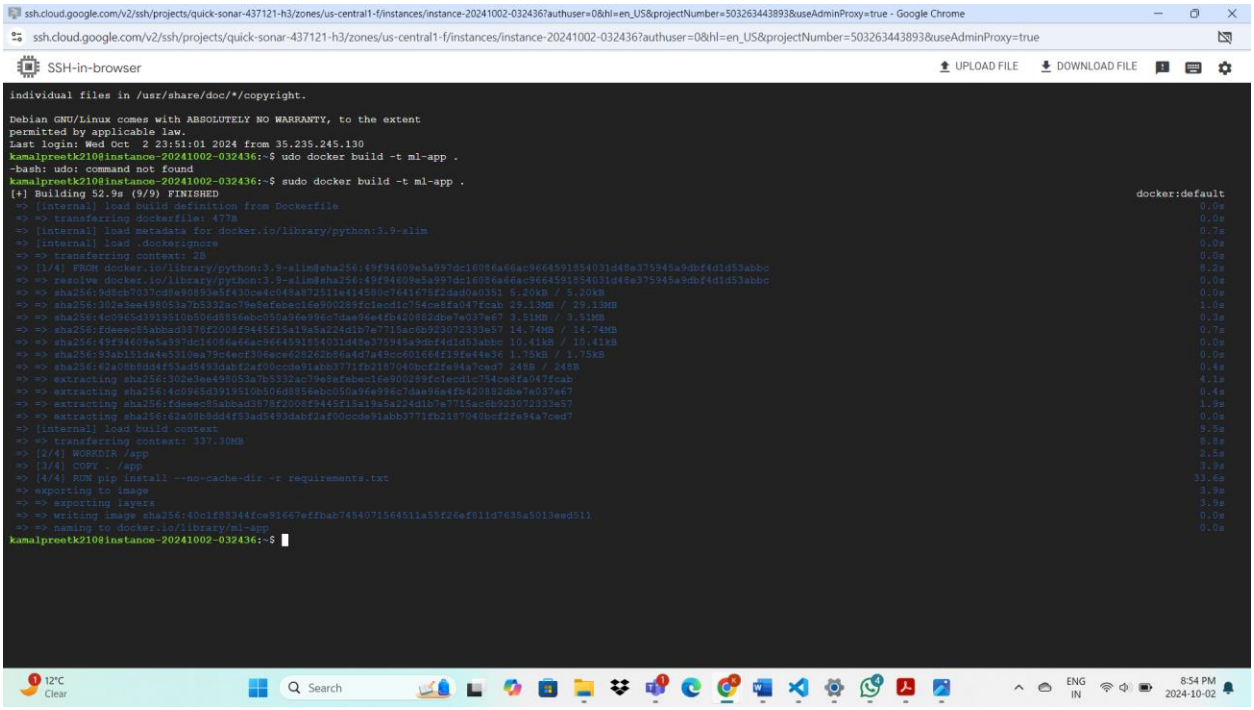
4. Update the Project Directory



Step 5: Build and run the Docker Container

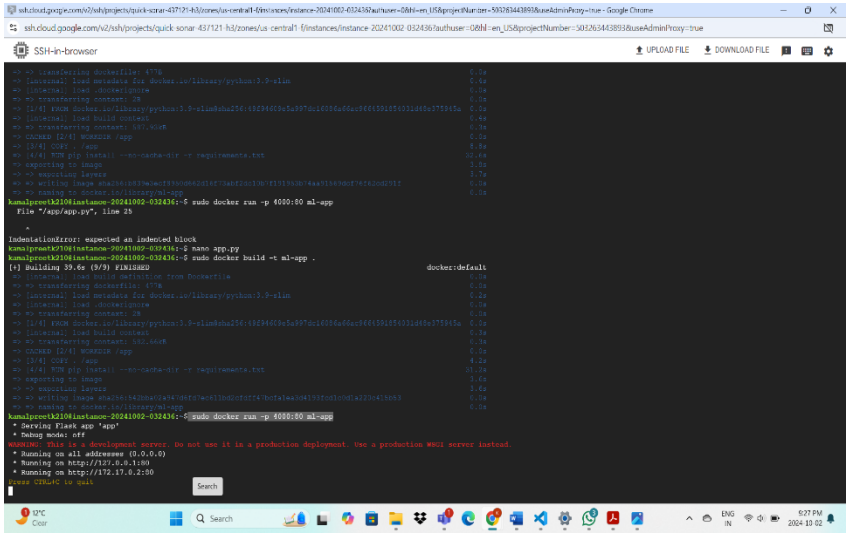
1.Build the Docker Image

sudo docker build -t ml-app .



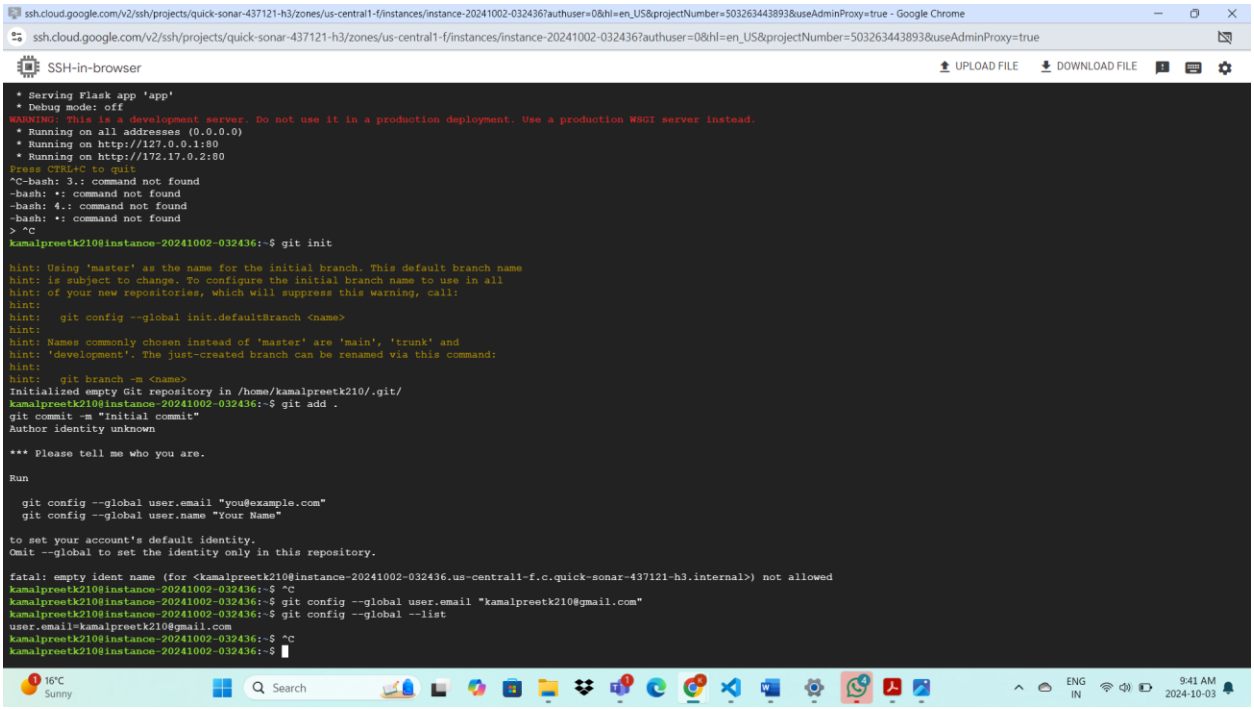
2. Run the Docker Container

sudo docker run -p 4000:80 ml-app

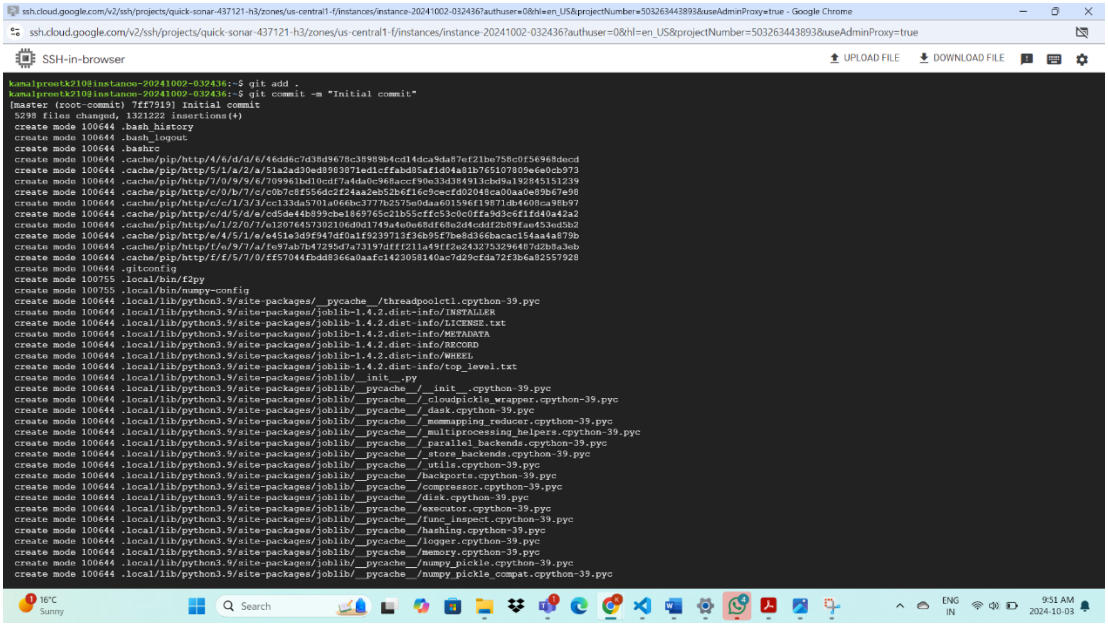


Step 6: Deploy the Application to GitHub

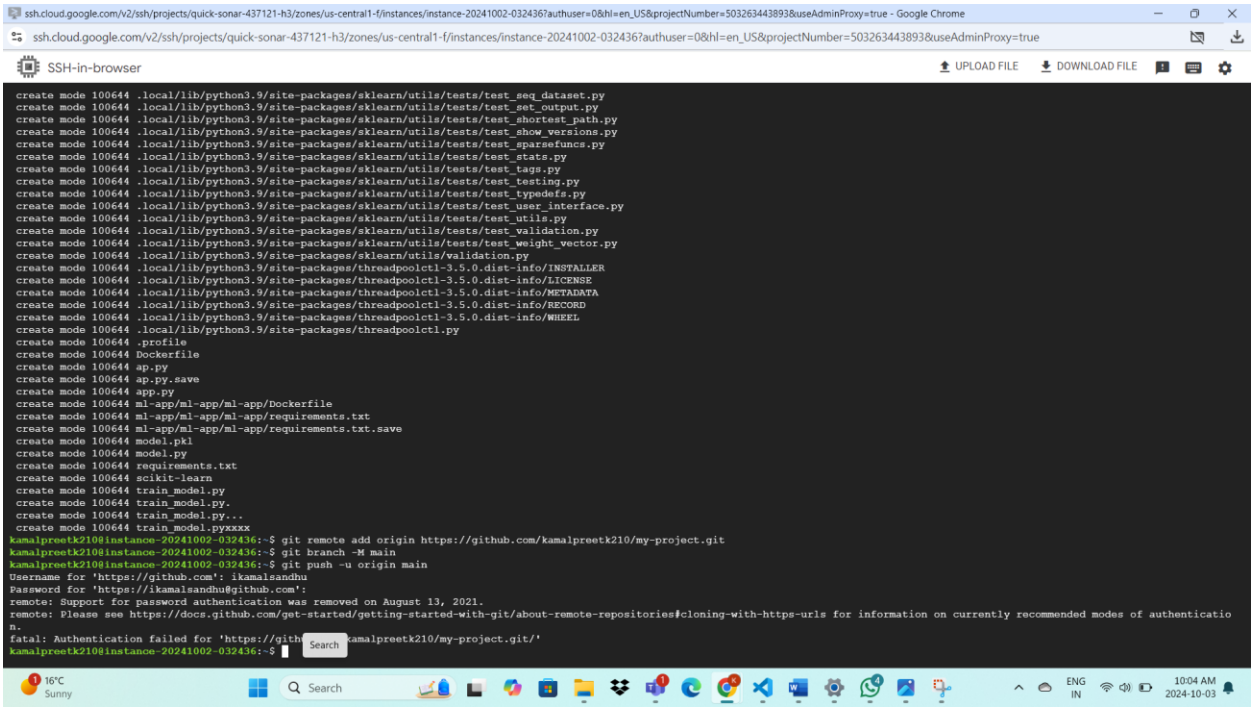
1. **Initialize a Git Repository**
   git init



2. **Add All Files and Commit**
   git add .
   git commit -m "Initial commit"



3. **Create a New Repository on GitHub**
   git remote add origin https://github.com/yourusername/your-repository.git
   git branch -M main
   git push -u origin main



Step 7: Document the Process

1. **Create a README.md File**

   - Document the process in a **README.md** file in your repository. Include the following:

     - Overview of the project
     - Instructions to build and run the Docker container