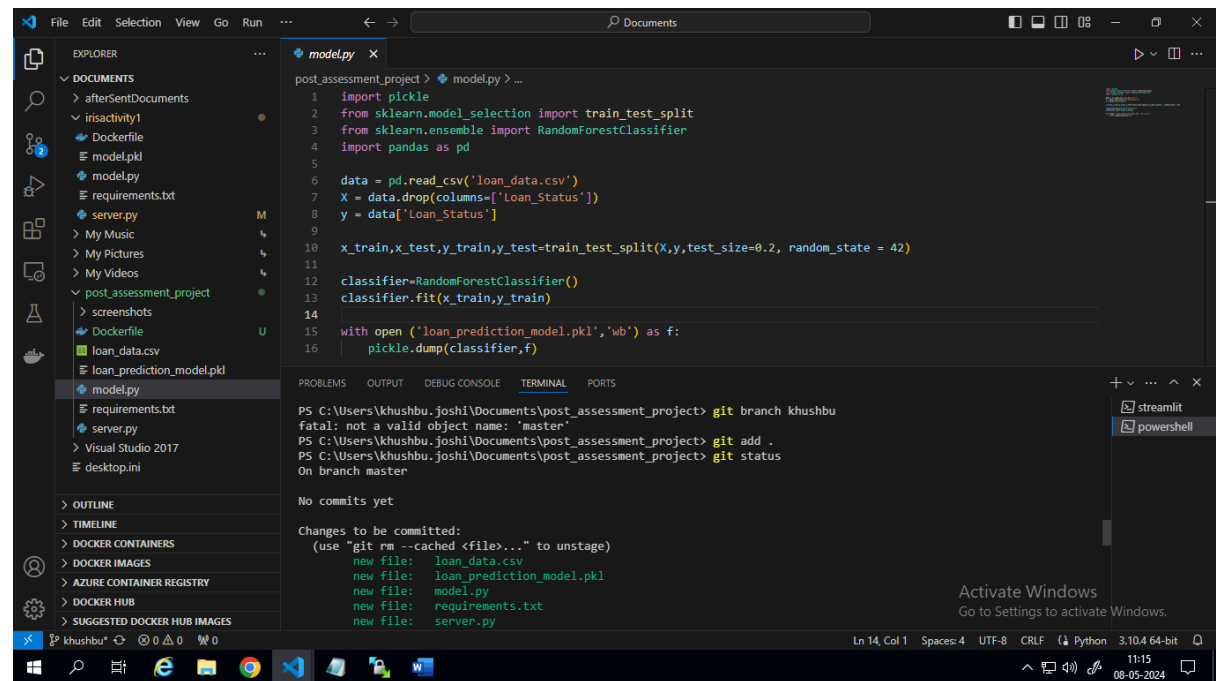


1. Creating ML Model with Python



The screenshot shows a Visual Studio Code editor with a file named `model.py` open. The file contains Python code for training a Random Forest classifier. The code imports `pickle`, `train_test_split` from `sklearn.model_selection`, `RandomForestClassifier` from `sklearn.ensemble`, and `pandas` as `pd`. It reads `loan_data.csv`, drops the `Loan_Status` column, and splits the data into training and testing sets. A `RandomForestClassifier` is trained on the training data, and the model is saved to `loan_prediction_model.pkl` using `pickle.dump`.

```
1 import pickle
2 from sklearn.model_selection import train_test_split
3 from sklearn.ensemble import RandomForestClassifier
4 import pandas as pd
5
6 data = pd.read_csv("loan_data.csv")
7 X = data.drop(columns=["Loan_Status"])
8 y = data["Loan_Status"]
9
10 x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.2, random_state = 42)
11
12 classifier=RandomForestClassifier()
13 classifier.fit(x_train,y_train)
14
15 with open ('loan_prediction_model.pkl','wb') as f:
16     pickle.dump(classifier,f)
```

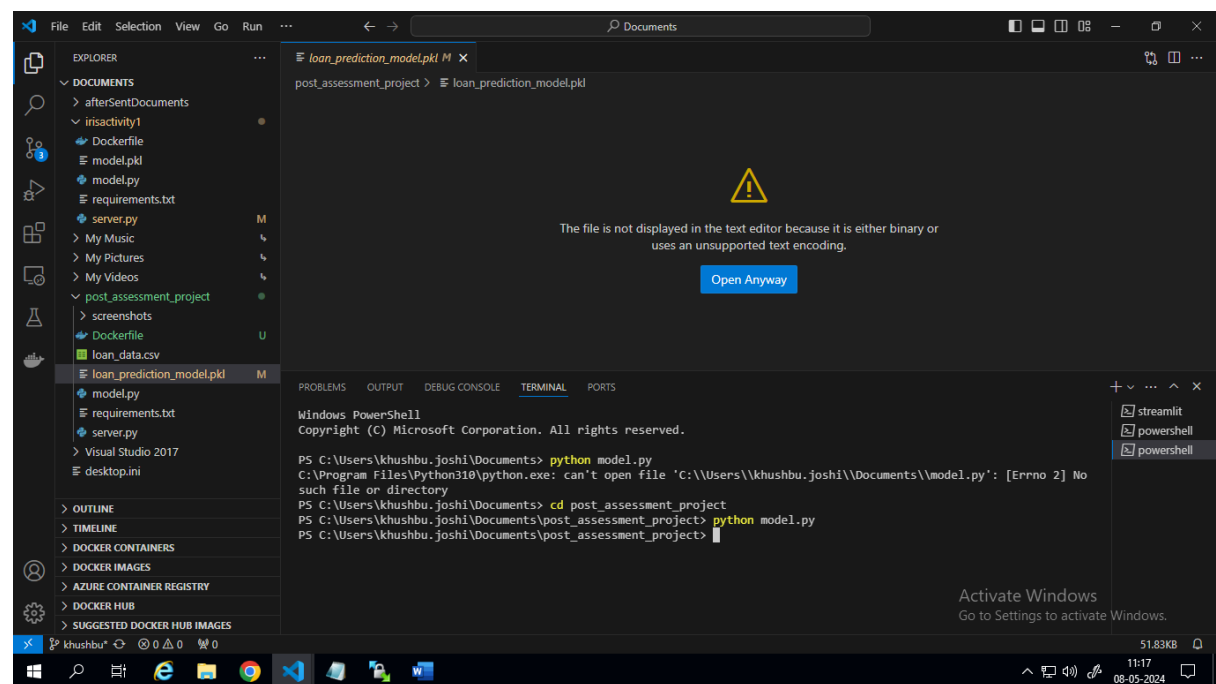
The terminal at the bottom shows the execution of `git` commands:

```
PS C:\Users\khushbu.joshi\Documents\post_assessment_project> git branch khushbu
fatal: not a valid object name: 'master'
PS C:\Users\khushbu.joshi\Documents\post_assessment_project> git add .
PS C:\Users\khushbu.joshi\Documents\post_assessment_project> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   loan_data.csv
        new file:   loan_prediction_model.pkl
        new file:   model.py
        new file:   requirements.txt
        new file:   server.py
```

2. Generating Pickle file



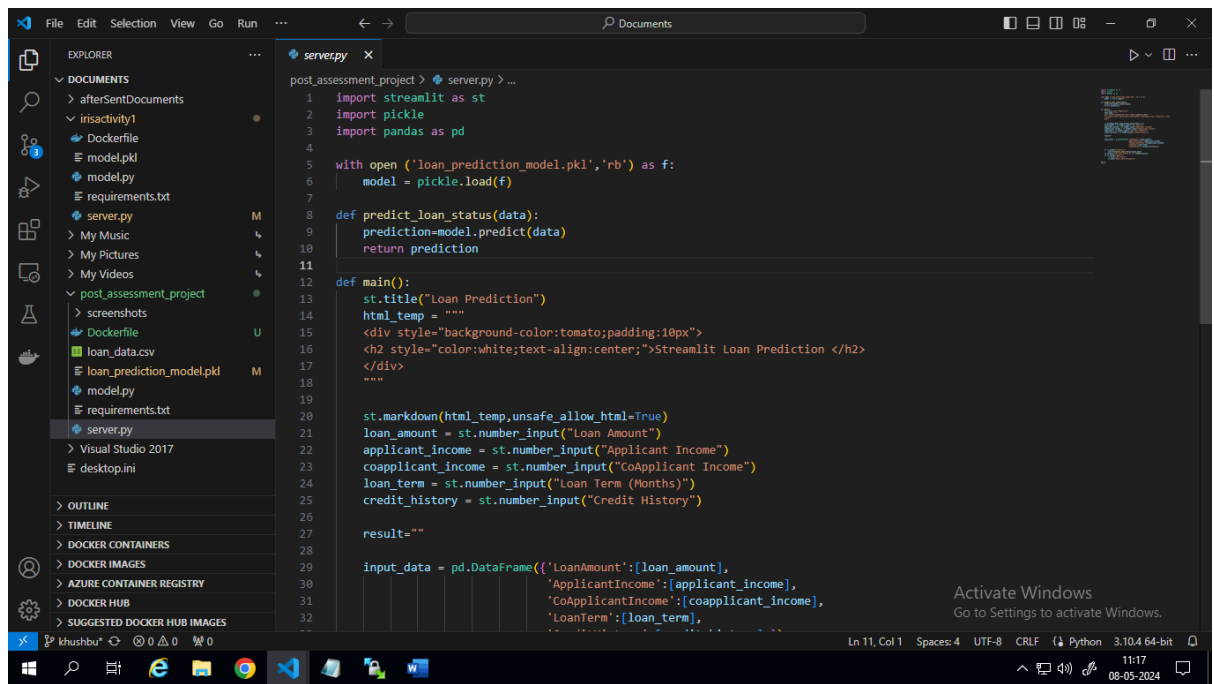
The screenshot shows a Visual Studio Code editor with a file named `loan_prediction_model.pkl` open. A warning message is displayed in the center of the editor, stating: "The file is not displayed in the text editor because it is either binary or uses an unsupported text encoding." Below the message is a button labeled "Open Anyway".

The terminal at the bottom shows the execution of `python model.py`:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\khushbu.joshi\Documents> python model.py
C:\Program Files\Python310\python.exe: can't open file 'C:\\Users\\khushbu.joshi\\Documents\\model.py': [Errno 2] No such file or directory
PS C:\Users\khushbu.joshi\Documents> cd post_assessment_project
PS C:\Users\khushbu.joshi\Documents\post_assessment_project> python model.py
PS C:\Users\khushbu.joshi\Documents\post_assessment_project>
```

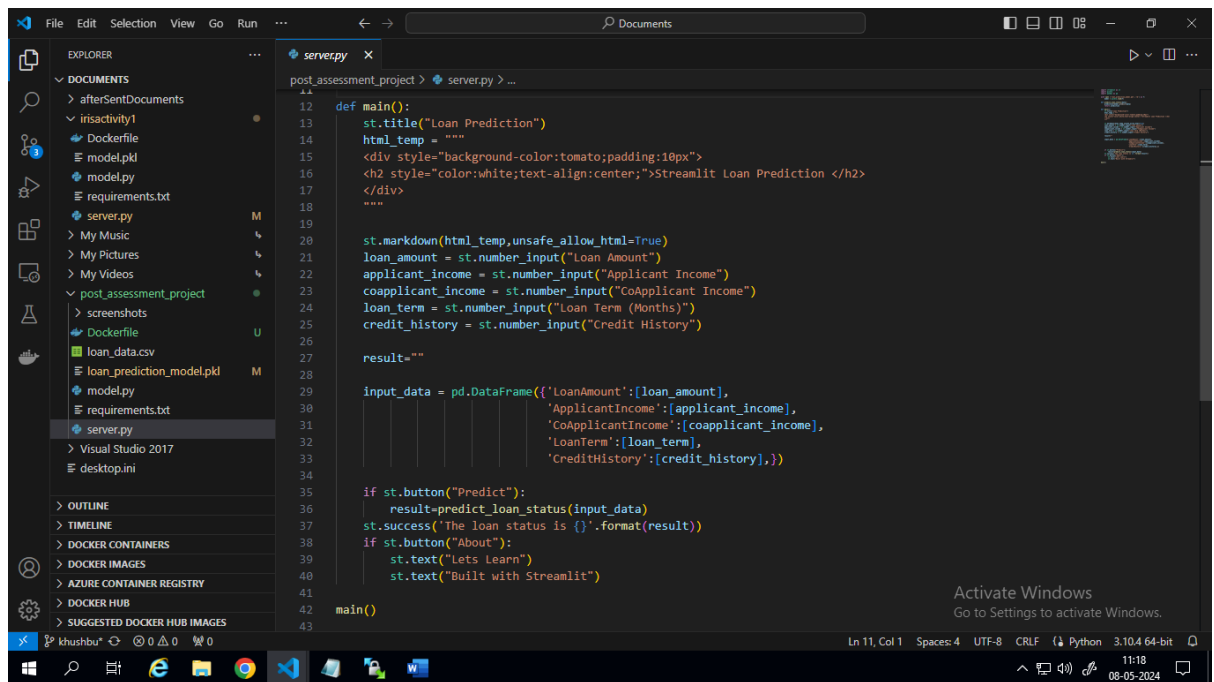
3. Web App with Streamlit



This screenshot shows the first part of the `server.py` file in VS Code. The Explorer sidebar on the left shows the project structure with files like `requirements.txt`, `model.pkl`, and `loan_data.csv`. The main editor displays the following code:

```
post_assessment_project > server.py > ...
1  import streamlit as st
2  import pickle
3  import pandas as pd
4
5  with open('loan_prediction_model.pkl','rb') as f:
6      model = pickle.load(f)
7
8  def predict_loan_status(data):
9      prediction=model.predict(data)
10     return prediction
11
12 def main():
13     st.title("Loan Prediction")
14     html_temp = """
15     <div style="background-color:tomato;padding:10px">
16     <h2 style="color:white;text-align:center;">Streamlit Loan Prediction </h2>
17     </div>
18     """
19
20     st.markdown(html_temp,unsafe_allow_html=True)
21     loan_amount = st.number_input("Loan Amount")
22     applicant_income = st.number_input("Applicant Income")
23     coapplicant_income = st.number_input("CoApplicant Income")
24     loan_term = st.number_input("Loan Term (Months)")
25     credit_history = st.number_input("Credit History")
26
27     result=""
28
29     input_data = pd.DataFrame({'LoanAmount':[loan_amount],
30                               'ApplicantIncome':[applicant_income],
31                               'CoApplicantIncome':[coapplicant_income],
32                               'LoanTerm':[loan_term],
33                               'CreditHistory':[credit_history],})
```

The status bar at the bottom indicates the file is at line 11, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.10.4 64-bit.

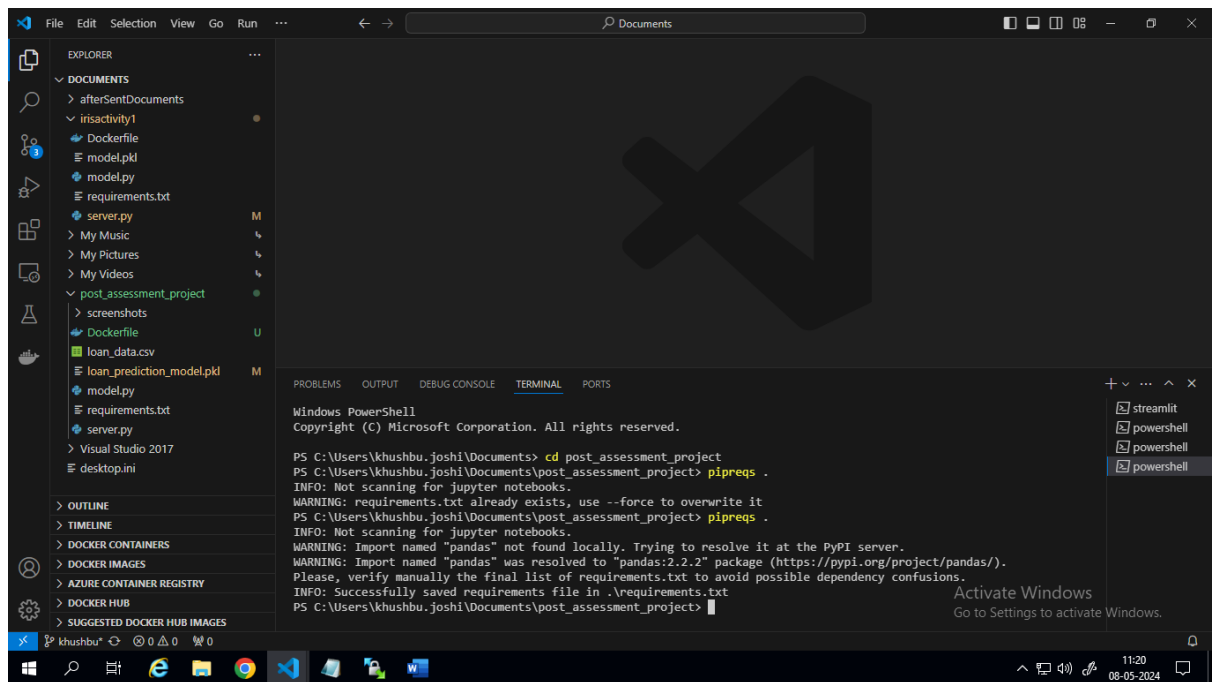


This screenshot shows the second part of the `server.py` file in VS Code. The Explorer sidebar on the left shows the project structure. The main editor displays the following code:

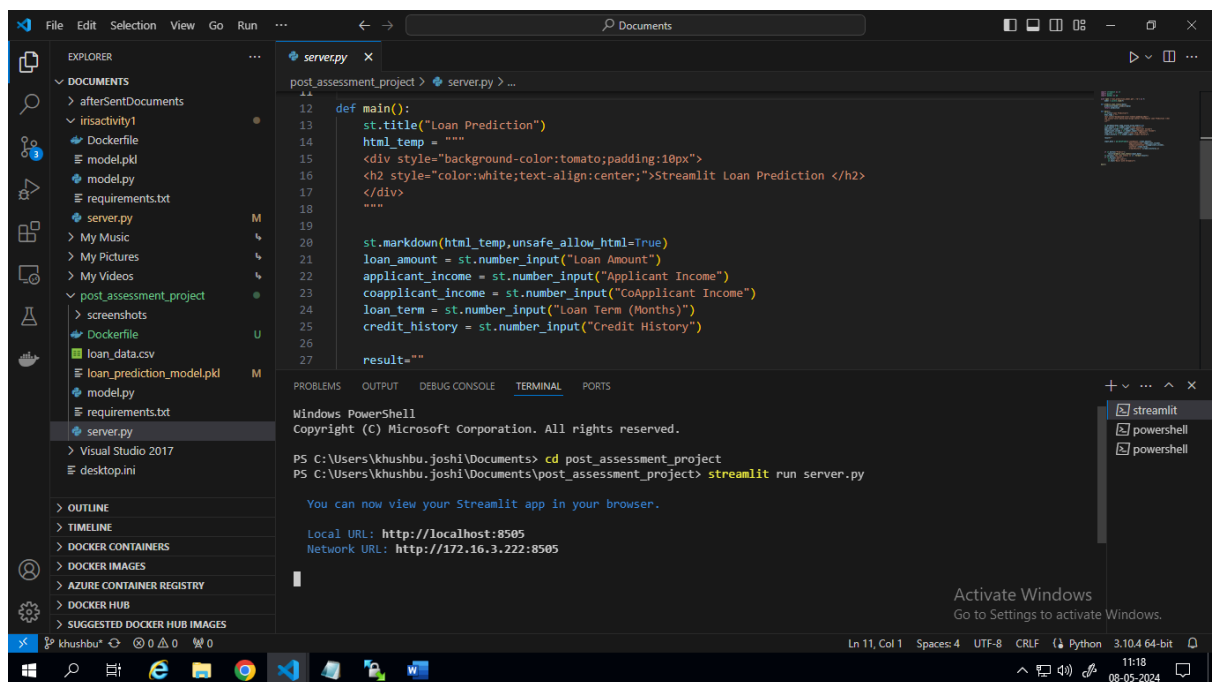
```
post_assessment_project > server.py > ...
11
12 def main():
13     st.title("Loan Prediction")
14     html_temp = """
15     <div style="background-color:tomato;padding:10px">
16     <h2 style="color:white;text-align:center;">Streamlit Loan Prediction </h2>
17     </div>
18     """
19
20     st.markdown(html_temp,unsafe_allow_html=True)
21     loan_amount = st.number_input("Loan Amount")
22     applicant_income = st.number_input("Applicant Income")
23     coapplicant_income = st.number_input("CoApplicant Income")
24     loan_term = st.number_input("Loan Term (Months)")
25     credit_history = st.number_input("Credit History")
26
27     result=""
28
29     input_data = pd.DataFrame({'LoanAmount':[loan_amount],
30                               'ApplicantIncome':[applicant_income],
31                               'CoApplicantIncome':[coapplicant_income],
32                               'LoanTerm':[loan_term],
33                               'CreditHistory':[credit_history],})
34
35     if st.button("Predict"):
36         result=predict_loan_status(input_data)
37         st.success('The loan status is {}'.format(result))
38     if st.button("About"):
39         st.text("Lets Leann")
40         st.text("Built with Streamlit")
41
42 main()
43
```

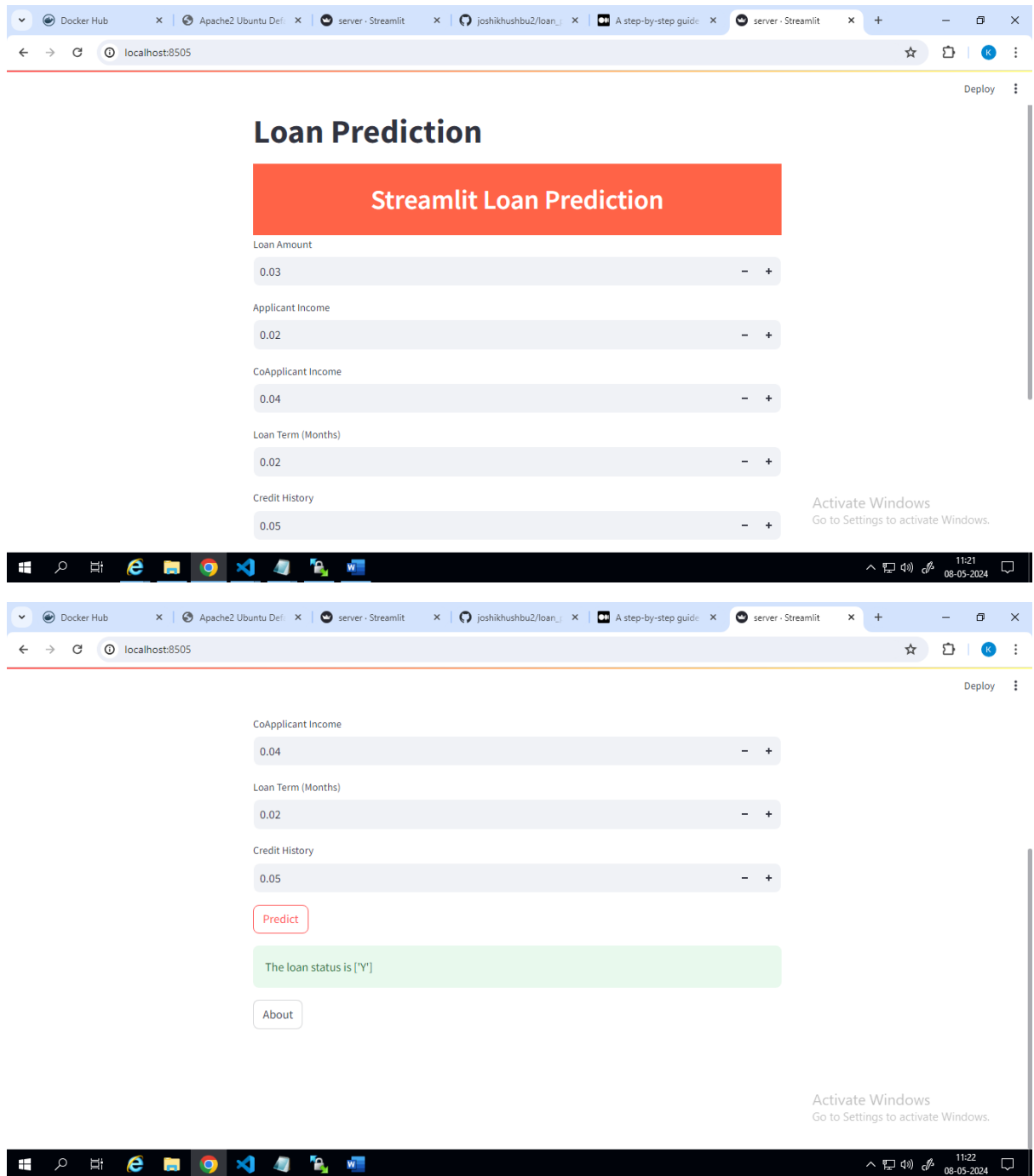
The status bar at the bottom indicates the file is at line 11, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.10.4 64-bit.

4. Creating requirements.txt

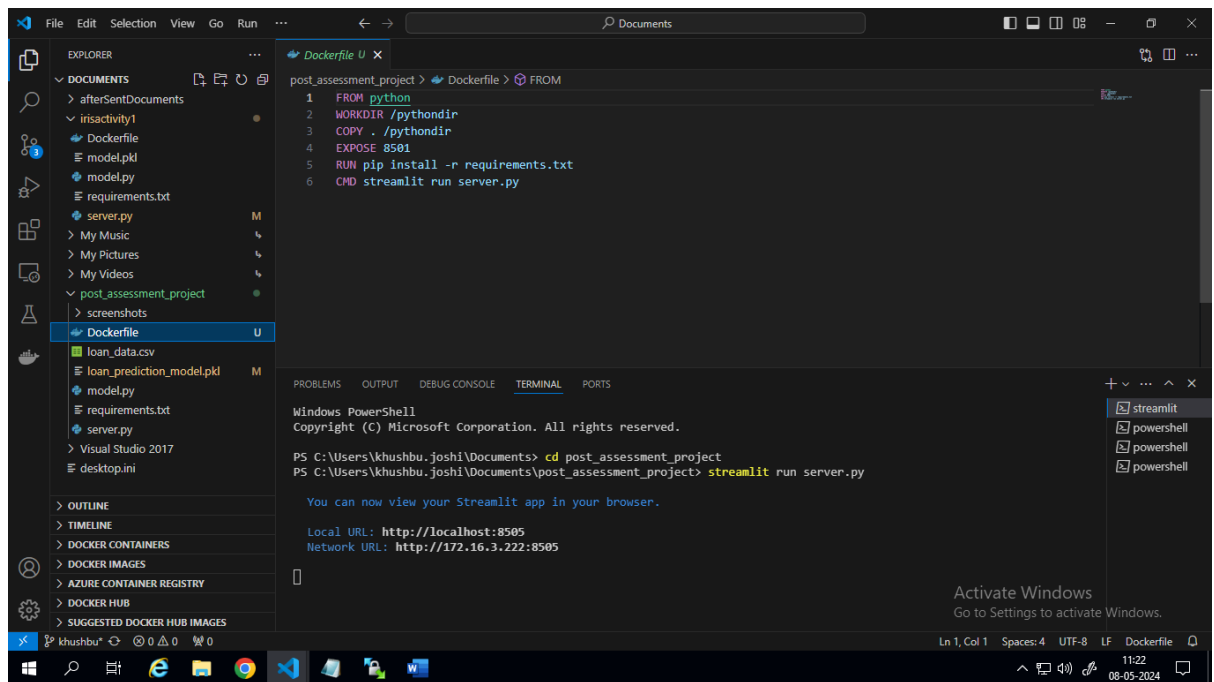


5. Executing files in rdp

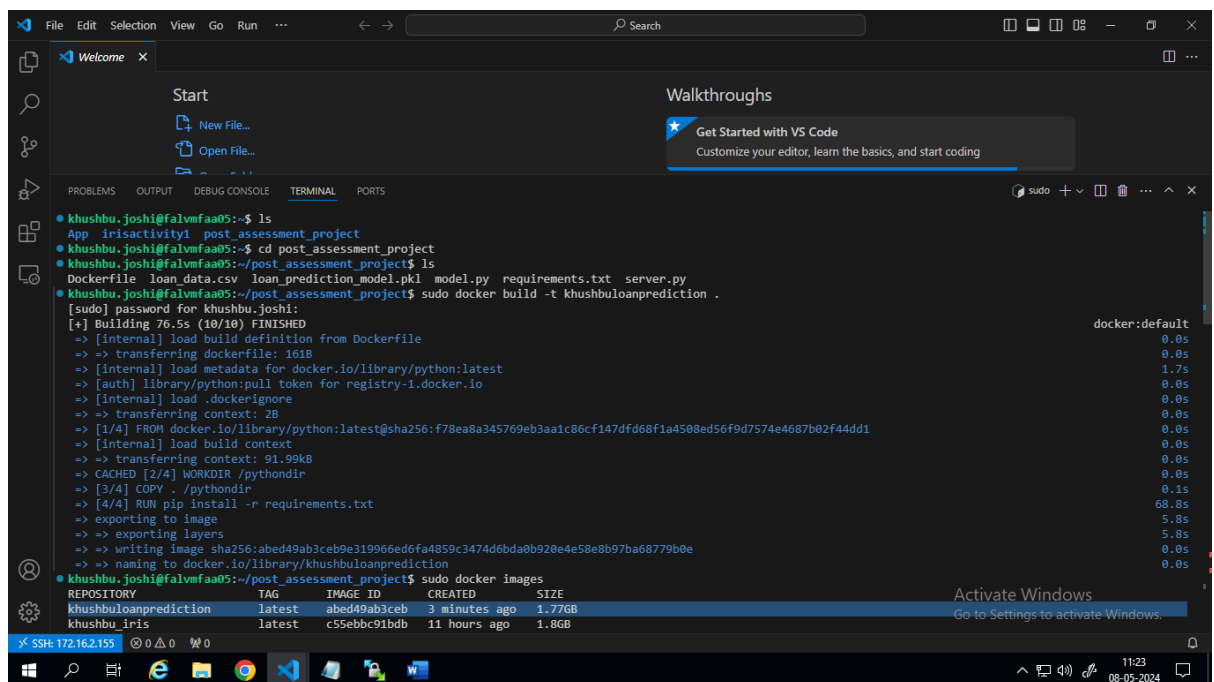




6. Creating Dockerfile



7. Building Docker image



8. Running Docker container

