Introduction To Problem Statement :-

- Developing a Handwriting Recognition Web Application : The web application should allow users to add data to their training dataset and also the recognition of letters.

- Starting to create an application or service has many problems but one of the main problems is which tool, language, stack or framework to build one's service or application on. As this is a deep convolutional neural network project so I have to use the libraries , frameworks and train dataset accordingly.

- Libraries used : numpy , bidict , flask, tensorflow , sklearn

- Tools used : Any OS, Jupyter lab , PyCharm

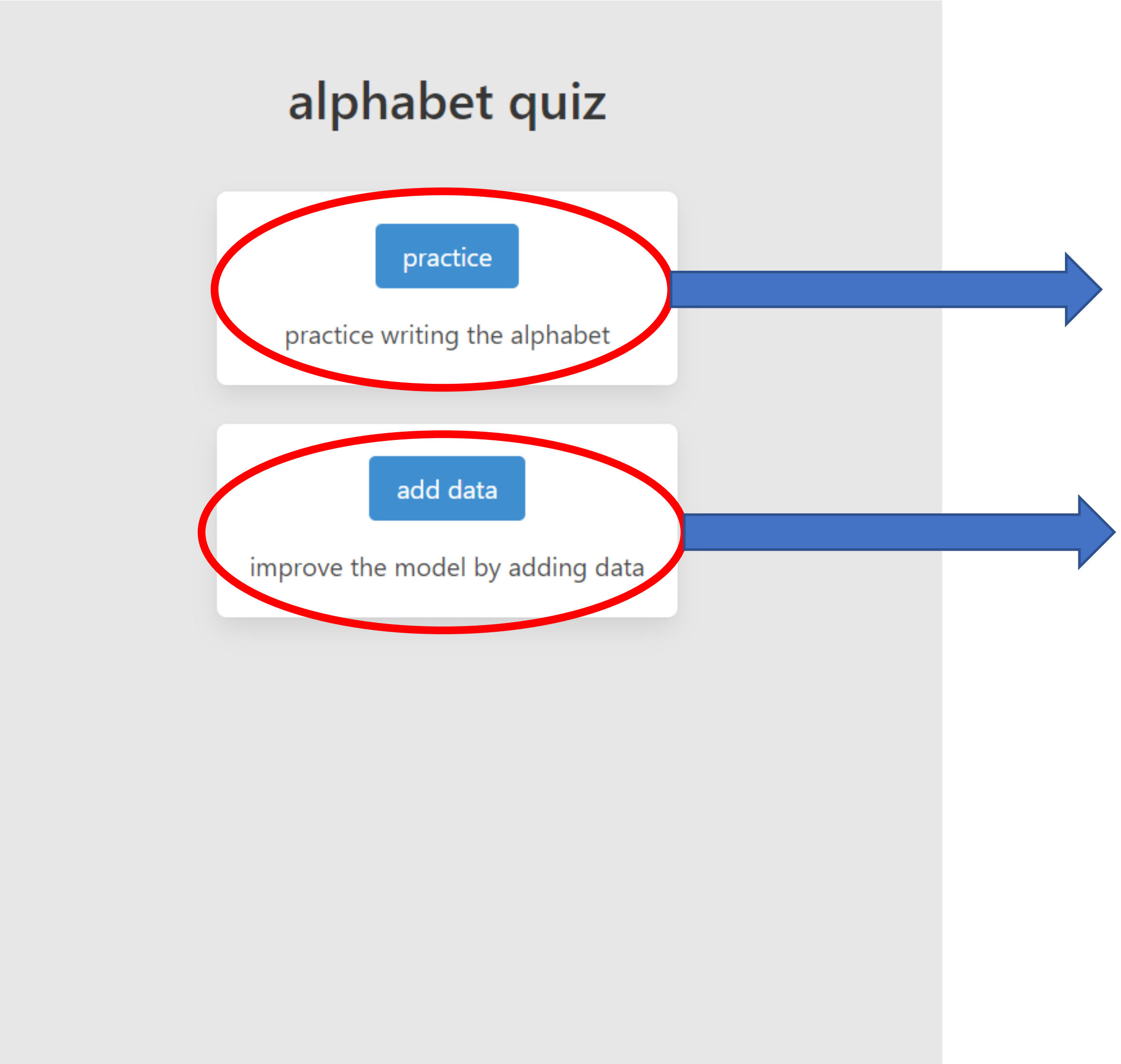
Basic Overview :-

- This web app is basically like a quiz which has two option in the landing page i.e. practice and add the data.

- Within Practice page one is given letter to write within a canvas and if a congratulatory message will be shown if he had written it correct else incorrect message.
- Within add data user is asked to write Upper case alphabets which are added to the training dataset and can to used further to train the model to increase its accuracy.

- It is basically hosted on a local host and if left idle for long time server will have timeout.

Landing Page (Home Page)



alphabet quiz

practice

practice writing the alphabet

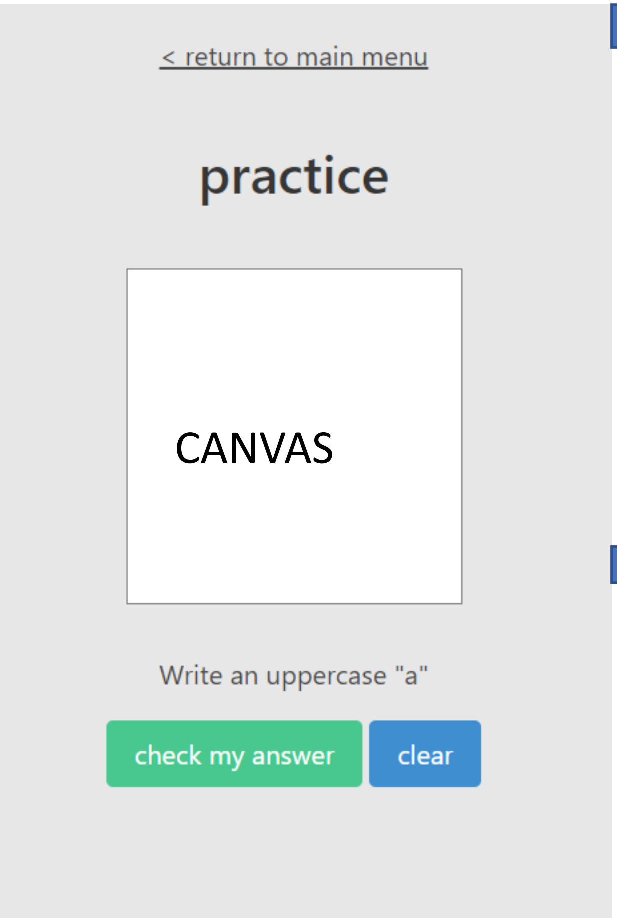
add data

improve the model by adding data

Practice Section

Add Data Section

PRACTICE PAGE :-

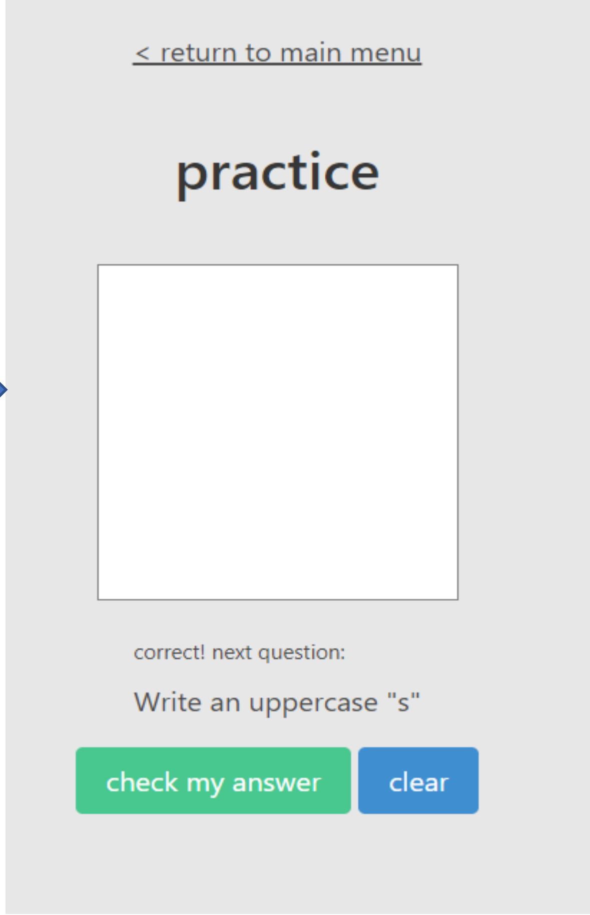

< return to main menu

practice

CANVAS

Write an uppercase "a"

TASK GIVEN

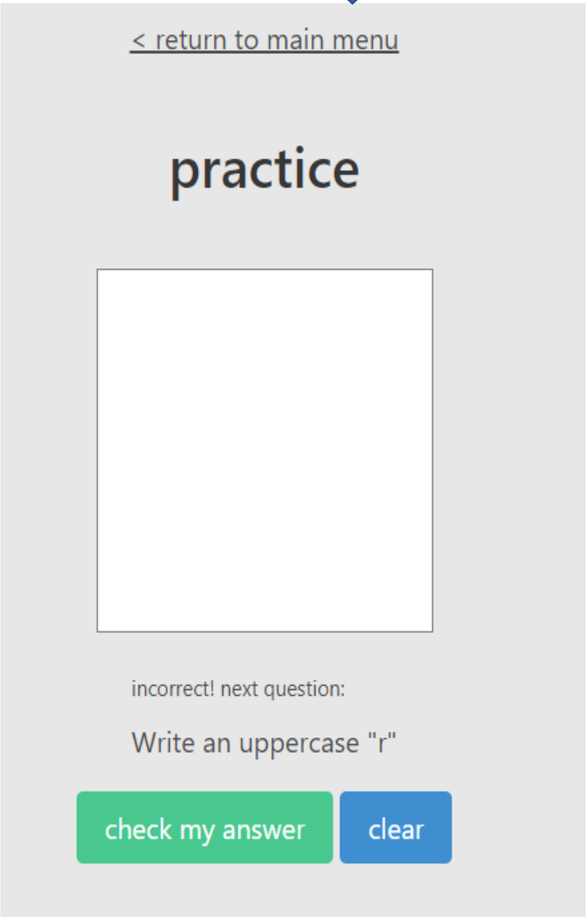

< return to main menu

practice

correct! next question:

Write an uppercase "s"

Completed Correctly


< return to main menu

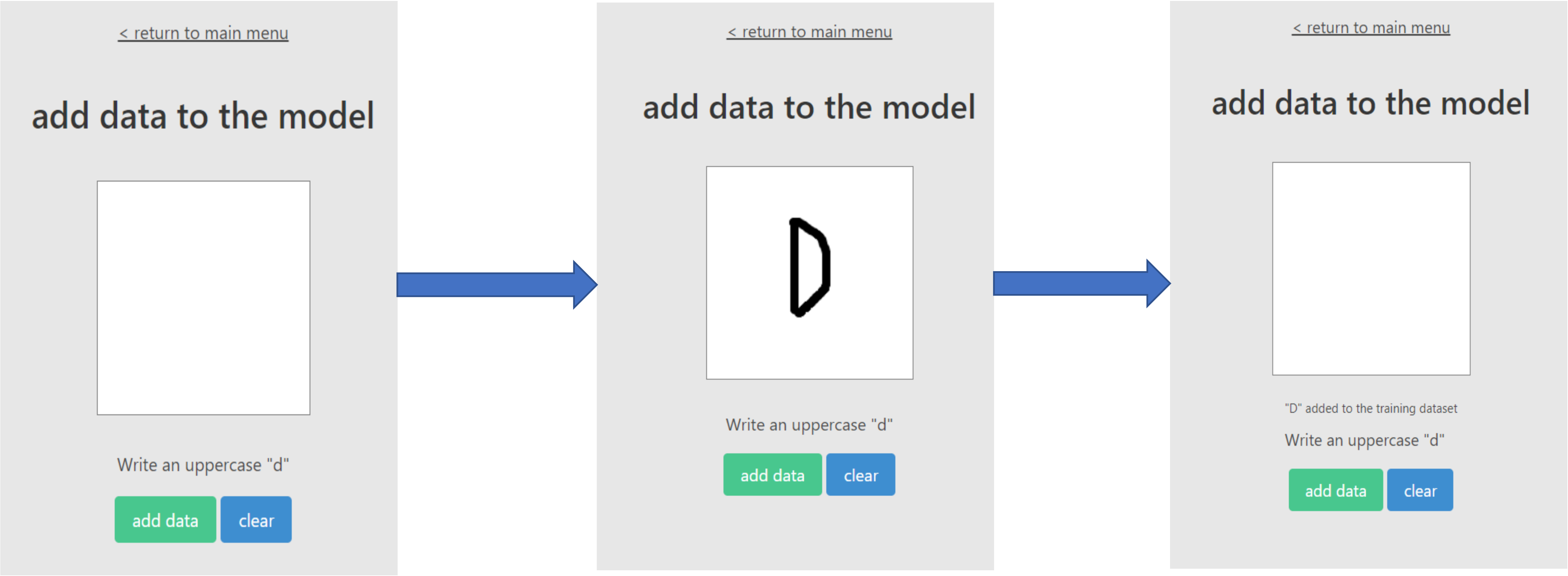
practice

incorrect! next question:

Write an uppercase "r"

Incorrect Answer

ADD DATA PAGE :-



< return to main menu

add data to the model

Write an uppercase "d"

< return to main menu

add data to the model

Write an uppercase "d"

< return to main menu

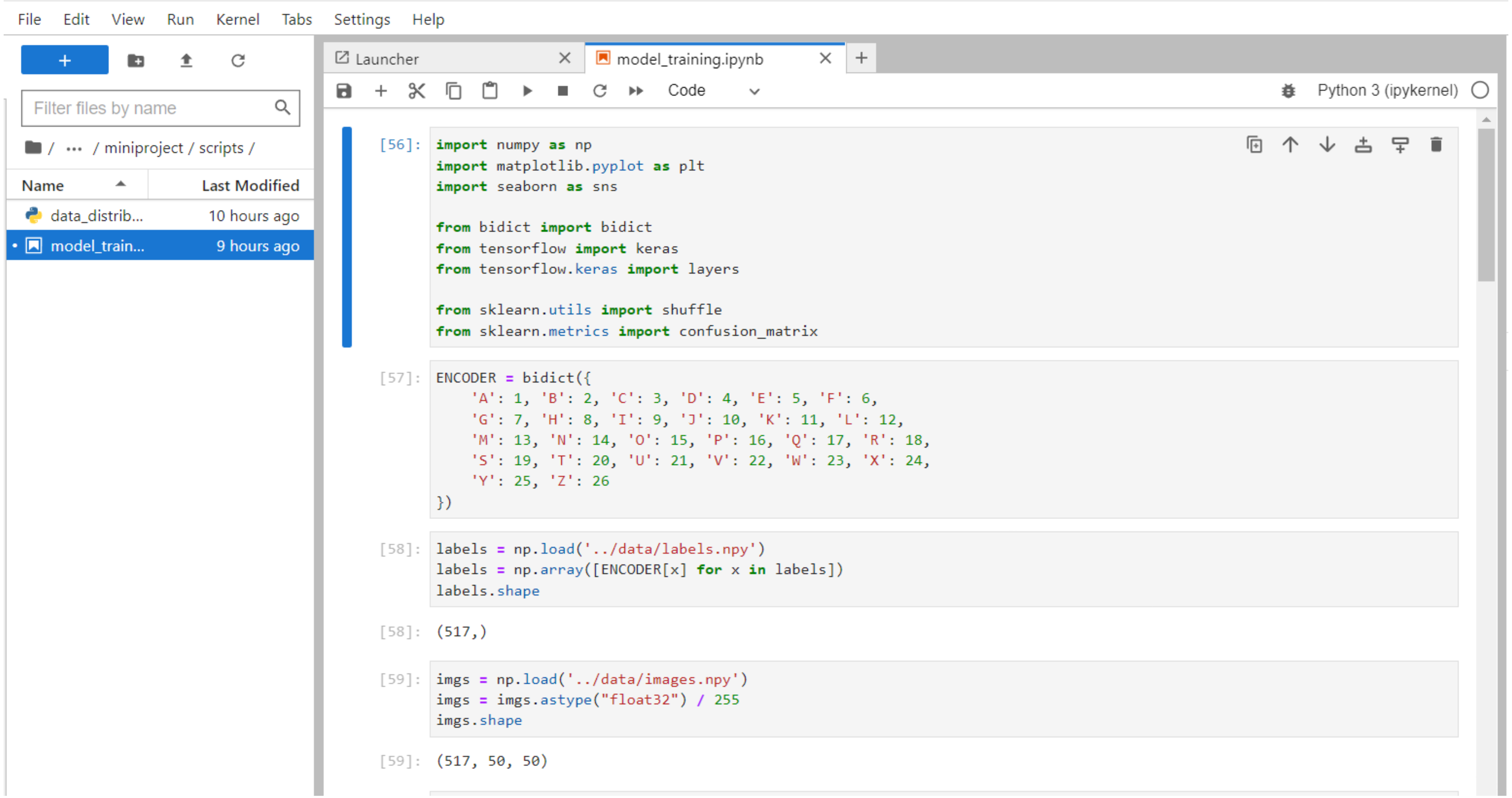
add data to the model

"D" added to the training dataset

Write an uppercase "d"

ALL STAGES OF ADD DATA PAGES

Letter Model :-



File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ ··· / miniproject / scripts /
Name
Last Modified
data_distrib... 10 hours ago
model_train... 9 hours ago
Launcher
model_training.ipynb
Code
Python 3 (ipykernel)

```
[56]: import numpy as np
      import matplotlib.pyplot as plt
      import seaborn as sns

      from bidict import bidict
      from tensorflow import keras
      from tensorflow.keras import layers

      from sklearn.utils import shuffle
      from sklearn.metrics import confusion_matrix
```

```
[57]: ENCODER = bidict({
          'A': 1, 'B': 2, 'C': 3, 'D': 4, 'E': 5, 'F': 6,
          'G': 7, 'H': 8, 'I': 9, 'J': 10, 'K': 11, 'L': 12,
          'M': 13, 'N': 14, 'O': 15, 'P': 16, 'Q': 17, 'R': 18,
          'S': 19, 'T': 20, 'U': 21, 'V': 22, 'W': 23, 'X': 24,
          'Y': 25, 'Z': 26
      })
```

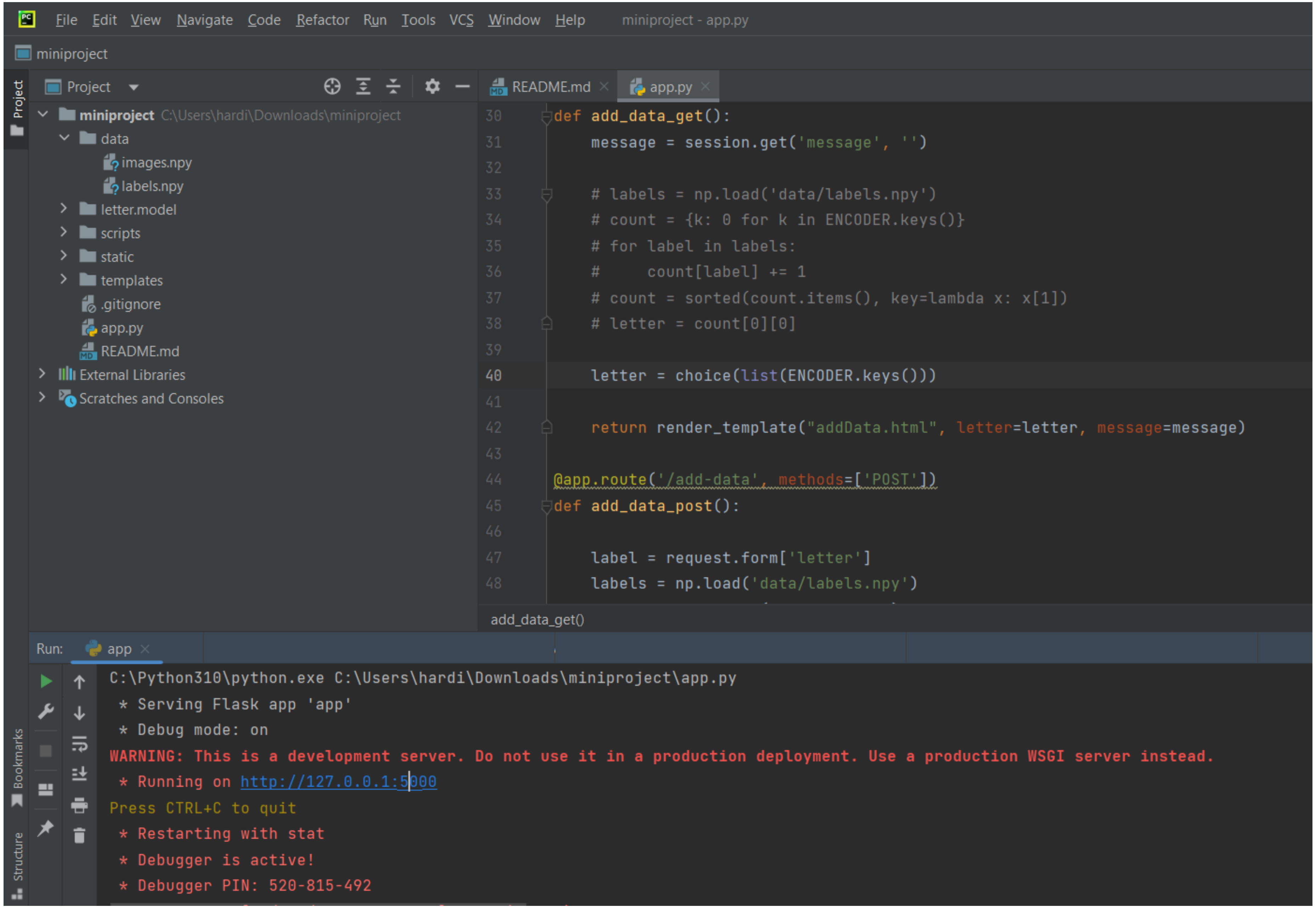
```
[58]: labels = np.load('../data/labels.npy')
      labels = np.array([ENCODER[x] for x in labels])
      labels.shape

[58]: (517,)
```

```
[59]: imgs = np.load('../data/images.npy')
      imgs = imgs.astype("float32") / 255
      imgs.shape

[59]: (517, 50, 50)
```

PyCharm Screen :-



```
def add_data_get():
    message = session.get('message', '')

    # labels = np.load('data/labels.npy')
    # count = {k: 0 for k in ENCODER.keys()}
    # for label in labels:
    #     count[label] += 1
    # count = sorted(count.items(), key=lambda x: x[1])
    # letter = count[0][0]

    letter = choice(list(ENCODER.keys()))

    return render_template("addData.html", letter=letter, message=message)

@app.route('/add-data', methods=['POST'])
def add_data_post():

    label = request.form['letter']
    labels = np.load('data/labels.npy')
```

```
C:\Python310\python.exe C:\Users\hardi\Downloads\miniproject\app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 520-815-492
```

AIM & Objectives :-

- To create user friendly
- Less complex
- Recognize letters correctly

Future Perspective :-

- Can be created for words also.
- Accuracy of model can be improved.
- UI can be enhanced.
- Can be integrated with digit recognition and after that for number plate detection.

THANK YOU