APPLICATION BUILDING

Create An HTML File

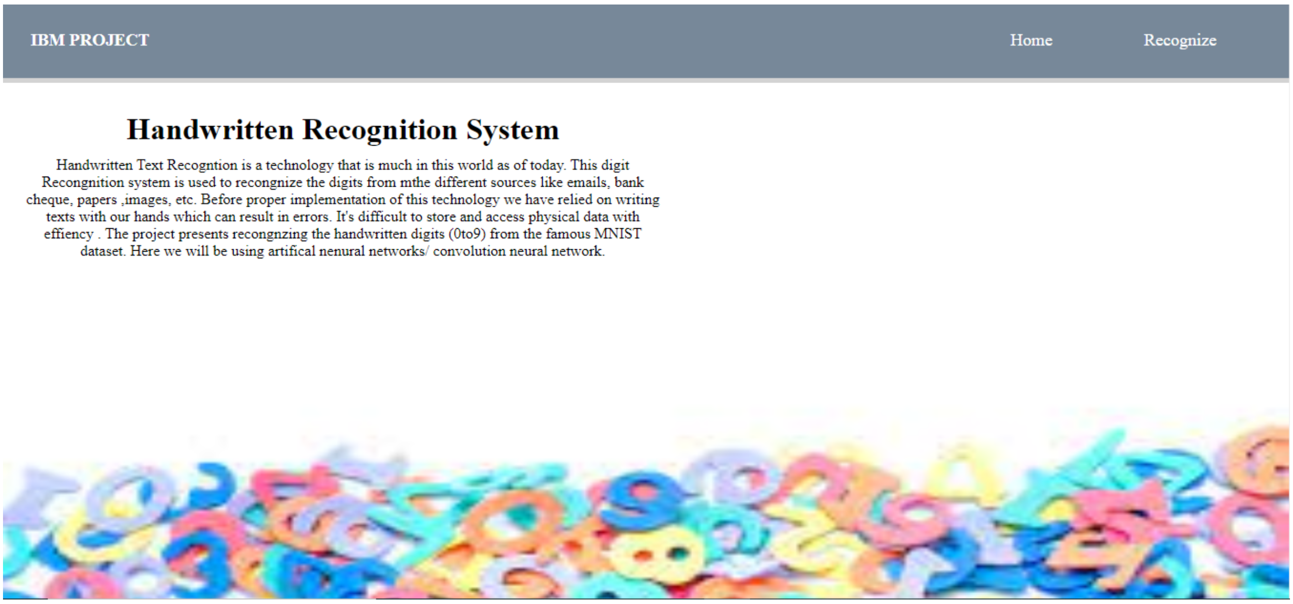
We use HTML to create the front end part of the web page.

Here, we created 2 html pages- index.html, web.html.

index.html displays home page.

web.html accepts the values from the input and displays the prediction.

Our index.html file looks like

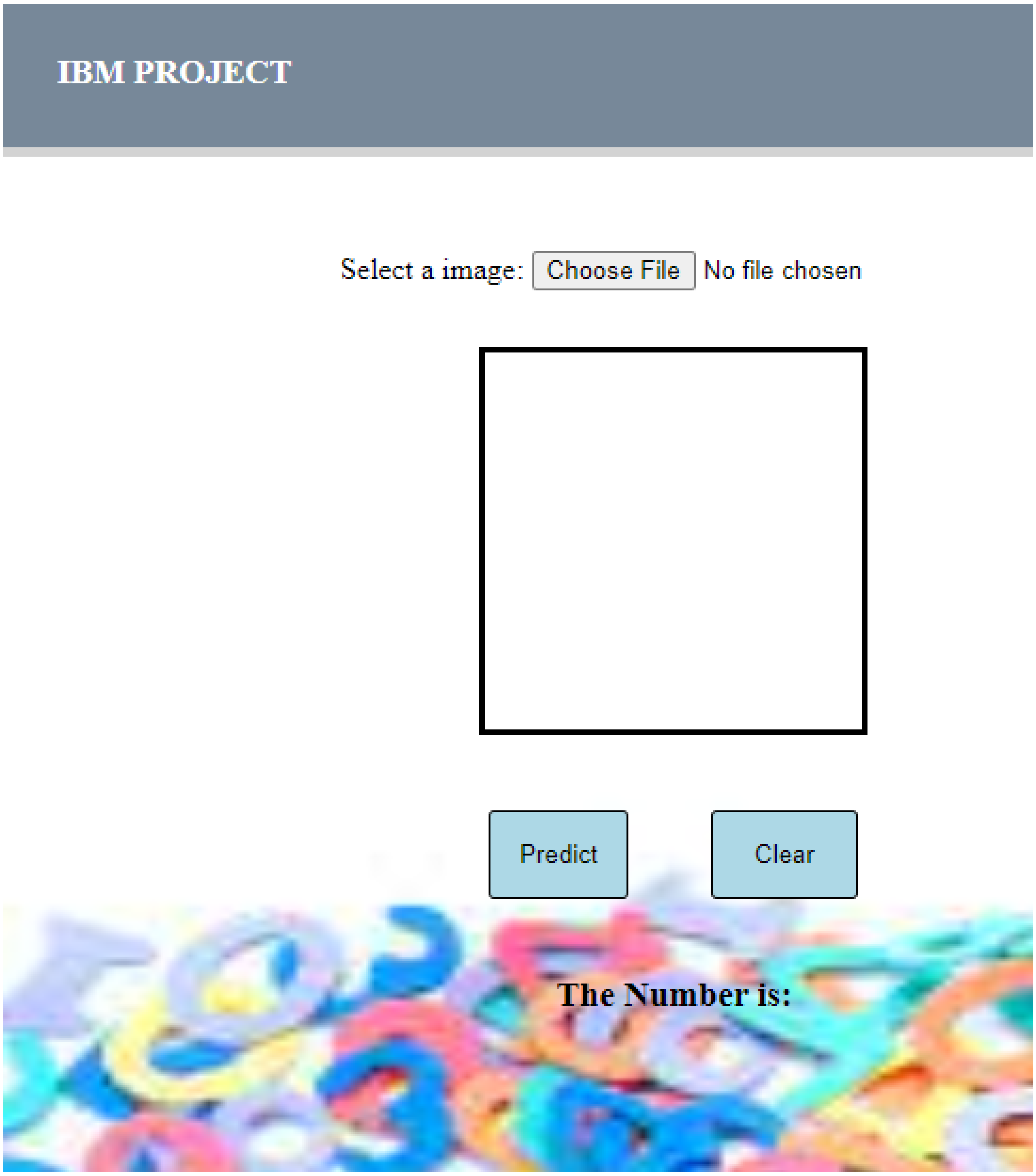


Handwritten Recognition System

Handwritten Text Recogntion is a technology that is much in this world as of today. This digit Recongnition system is used to recongnize the digits from mthe different sources like emails, bank cheque, papers ,images, etc. Before proper implementation of this technology we have relied on writing texts with our hands which can result in errors. It's difficult to store and access physical data with effiency . The project presents recongnzing the handwritten digits (0to9) from the famous MNIST dataset. Here we will be using artifical nenural networks/ convolution neural network.

This is the main page which describes about the project and summarizes it.

web.html accepts the values from the input and displays the prediction.



Build Python Code (Part 1)

flask file ‘app.py’ which is a web framework written in python for server-side scripting. step by step procedure for building the backend application.

App starts running when the “__name__” constructor is called in main.

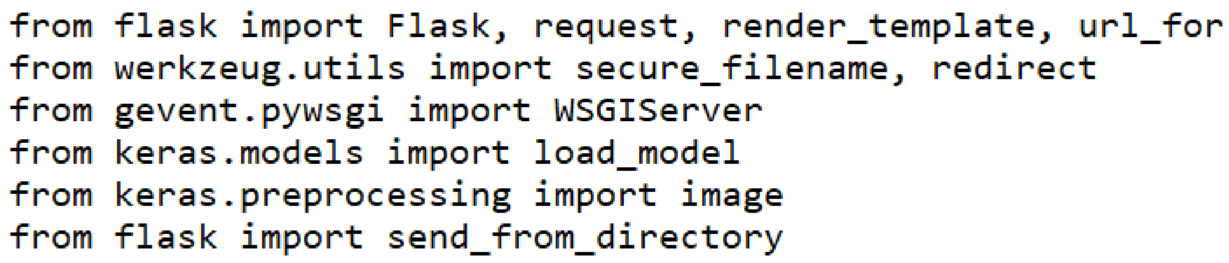
render_template is used to return HTML file.

“GET” method is used to take input from the user.

“POST” method is used to display the output to the user.

Import Libraries:

Libraries required for the app to run are to be imported.

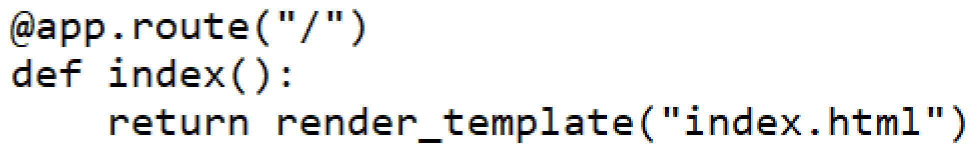


```
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory
```

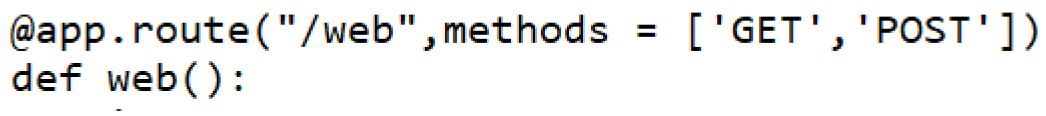
Routing to the html Page

We are routing the app to the HTML templates which we want to render.

Firstly we are rendering the main.html template and from there we are navigating to our prediction page that is index.html



```
@app.route("/")
def index():
    return render_template("index.html")
```



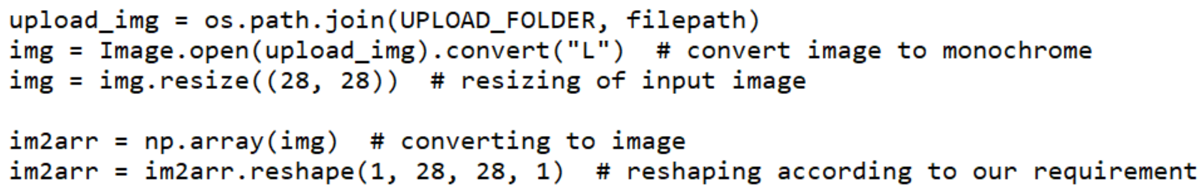
```
@app.route("/web",methods = ['GET','POST'])
def web():
```

Returning the prediction on UI:

Build Python Code (Part 2)

Here the route for prediction is given and necessary steps are performed in order to get the predicted output.

Necessary conditions are given according to the input classes and the app will be returning the templates according to that.

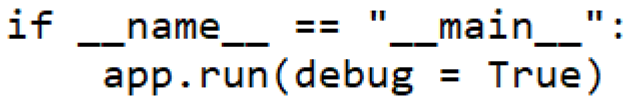


```
upload_img = os.path.join(UPLOAD_FOLDER, filepath)
img = Image.open(upload_img).convert("L")  # convert image to monochrome
img = img.resize((28, 28))  # resizing of input image

im2arr = np.array(img)  # converting to image
im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement
```

Main Function:

This function runs your app in a web browser

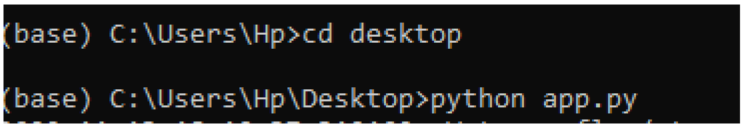


```
if __name__ == "__main__":
    app.run(debug = True)
```

.

Run The Application

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type “python app.py” command

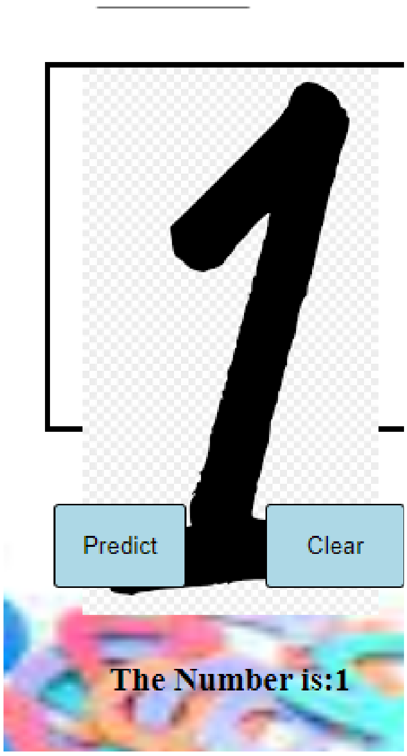


```
(base) C:\Users\Hp>cd desktop

(base) C:\Users\Hp\Desktop>python app.py
```

Navigate to the localhost where you can view your web page

Upload an image and see the predicted output on UI of your page and result looks like:



The Number is:1