Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Team ID: PNT2022TMID23190

Application Building

In this section, we have built a web application that is integrated to the model we built. A UI is provided to the user where he has uploaded the image. Based on the saved model, the uploaded image will be analyzed and prediction is showcased on the UI.

This section has the following tasks

Building HTML Pages

Building server-side script

- **Building Html Pages**

For this project create three HTML files namely

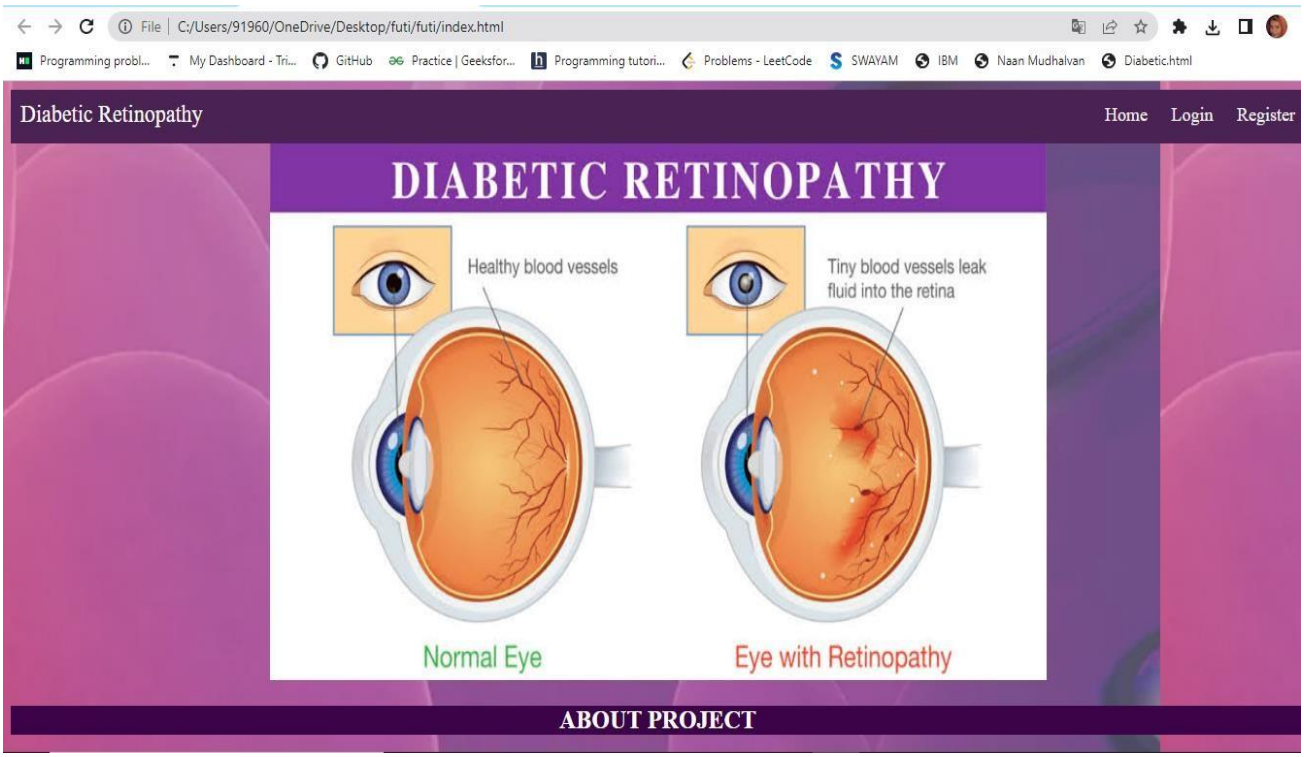
index.html

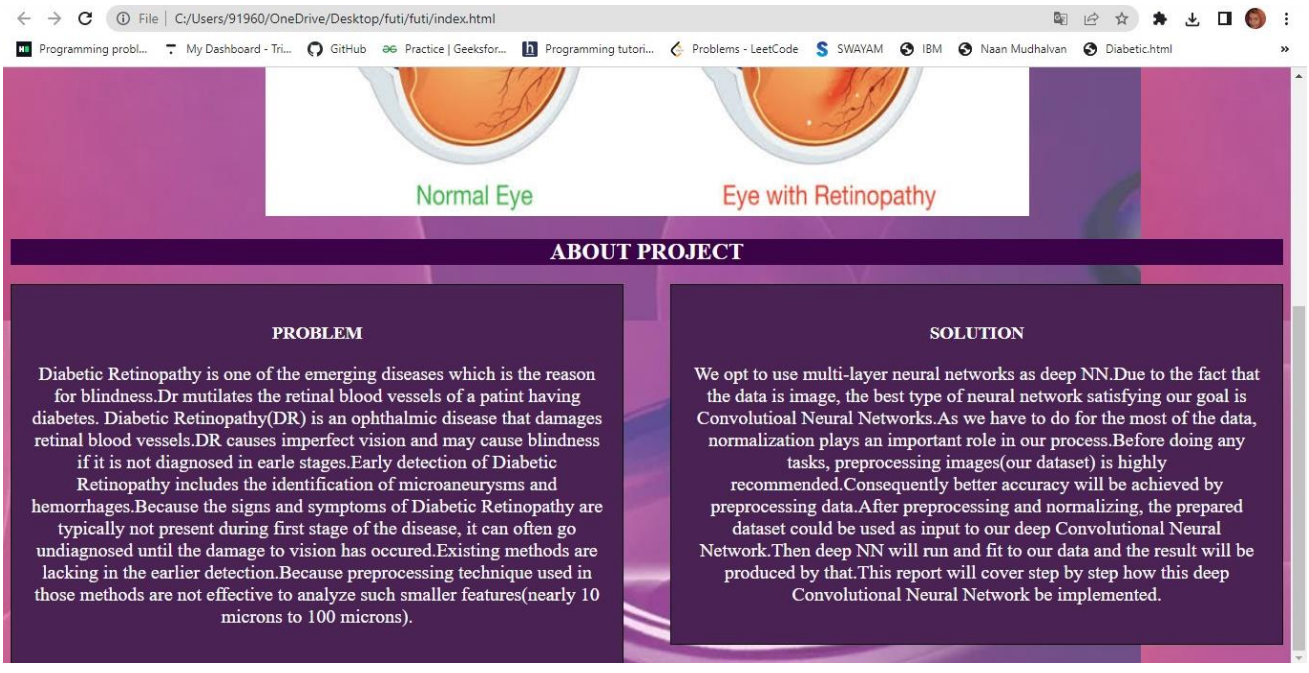
register.html

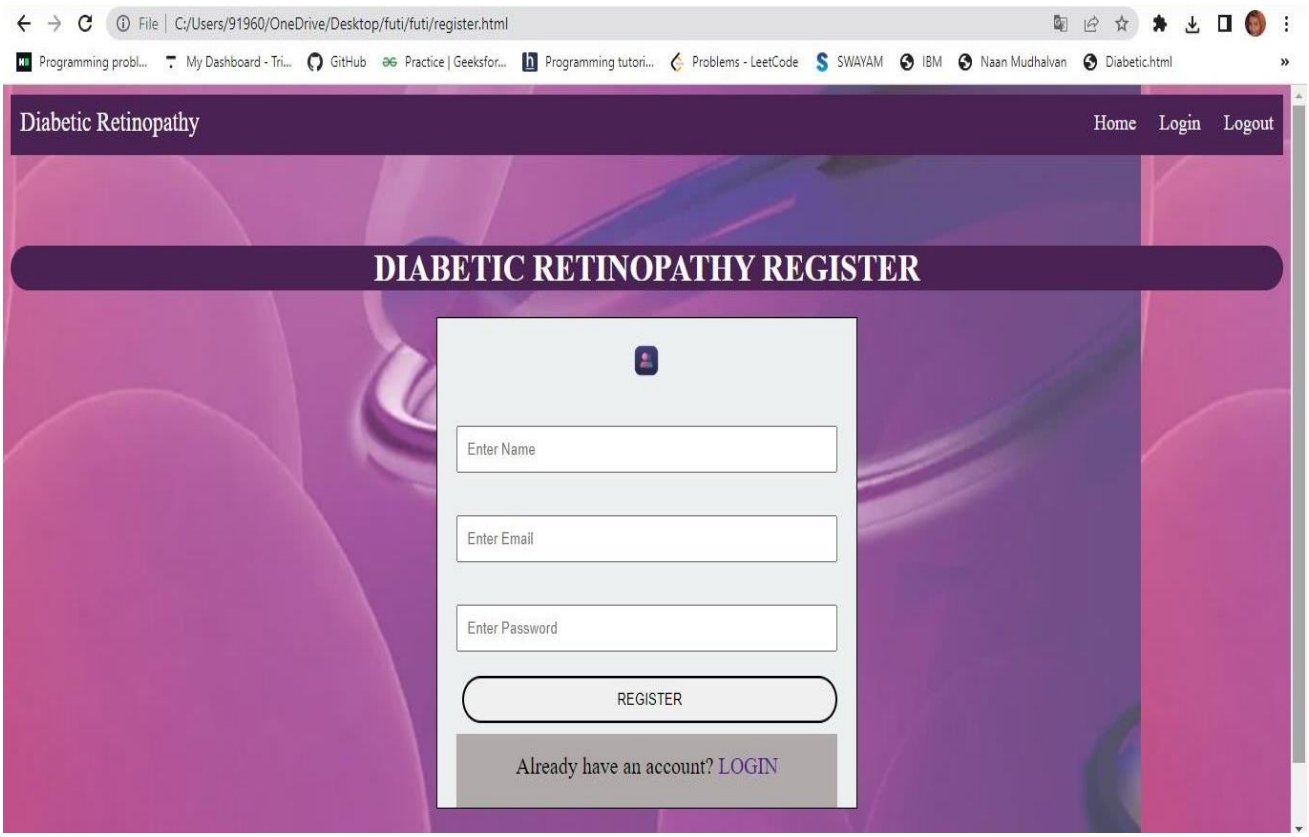
login.html

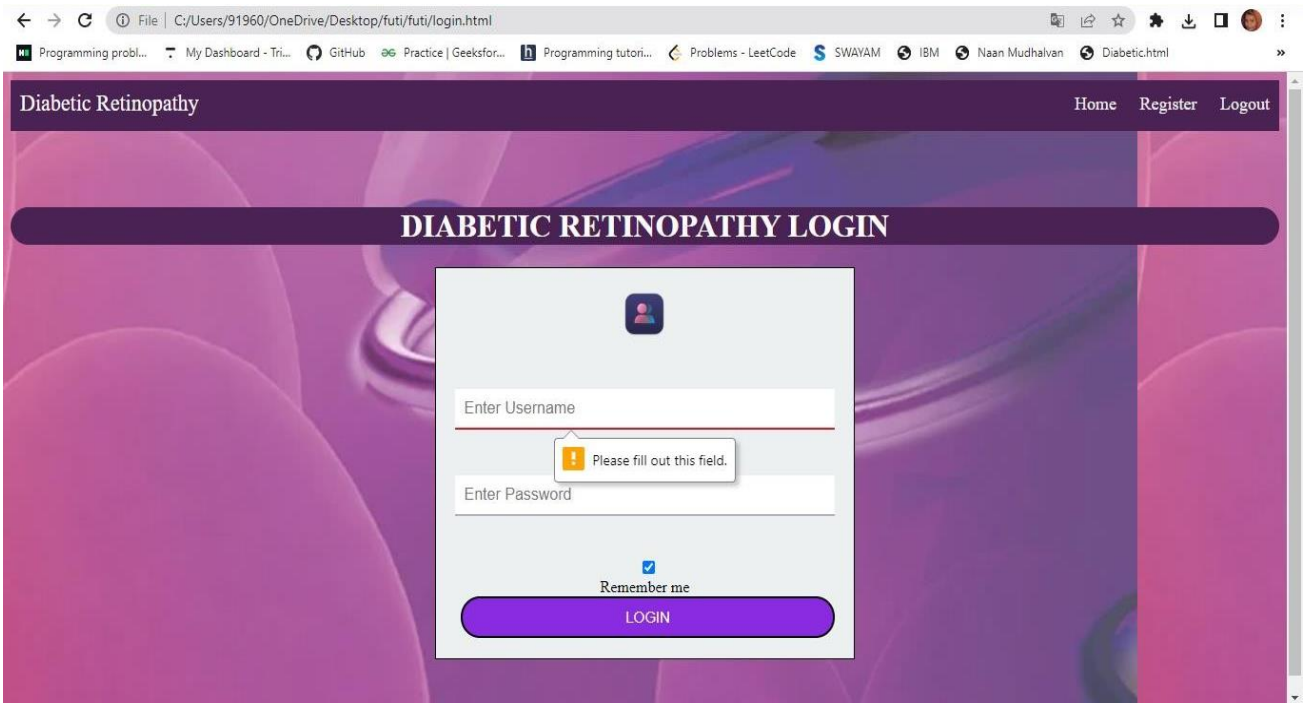
prediction.html

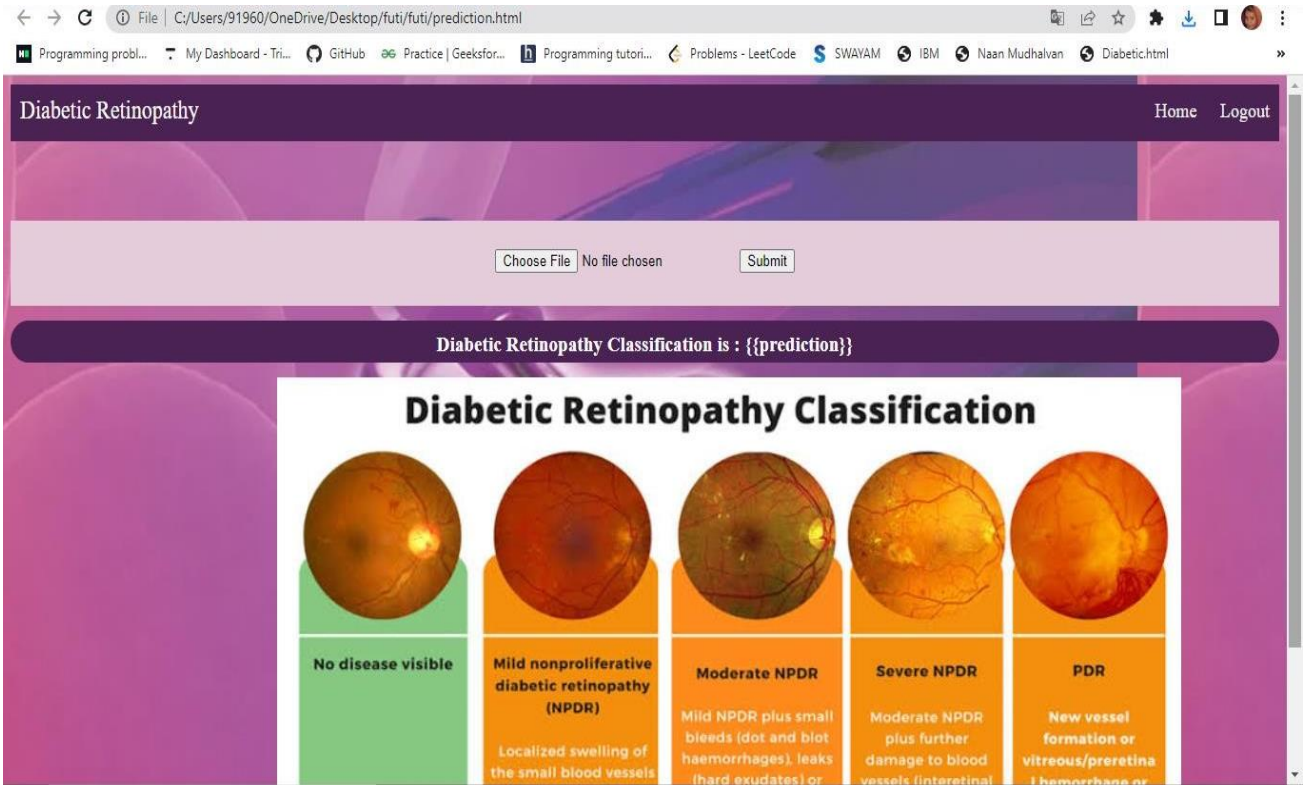
logout.html

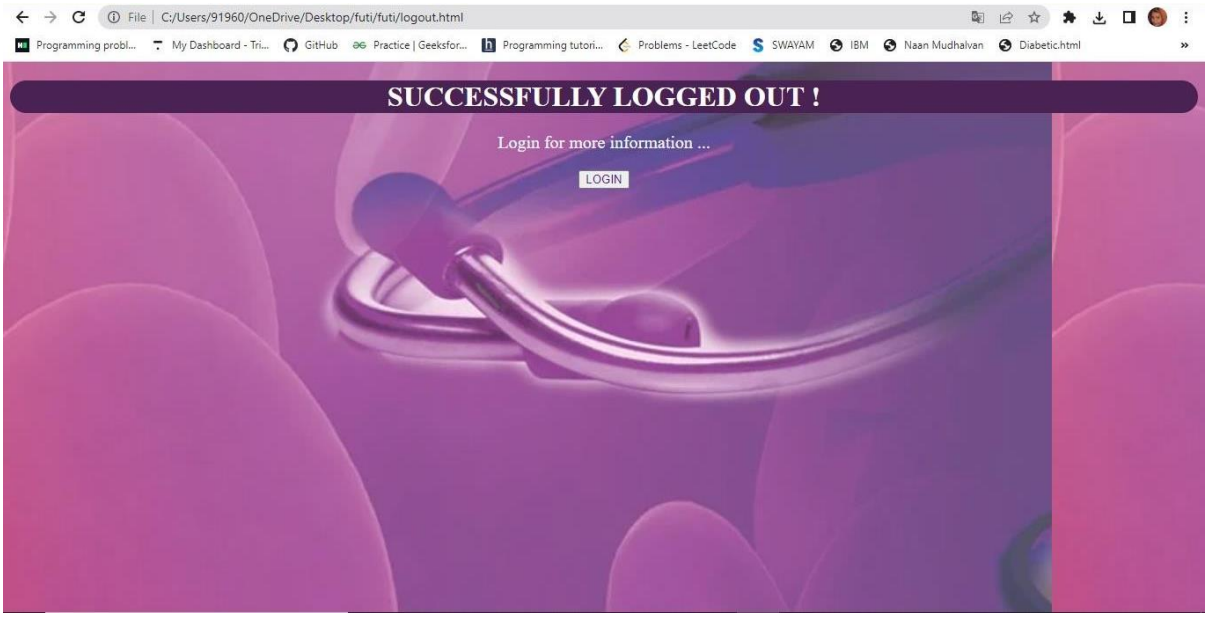

File | C:/Users/91960/OneDrive/Desktop/futi/futi/index.html
Programming probl...
My Dashboard - Tri...
GitHub
Practice | Geeksfor...
Programming tutori...
Problems - LeetCode
SWAYAM
IBM
Naan Mudhalvan
Diabetic.html
Diabetic Retinopathy
Home
Login
Register
DIABETIC RETINOPATHY
Healthy blood vessels
Tiny blood vessels leak fluid into the retina
Normal Eye
Eye with Retinopathy
ABOUT PROJECT


File | C:/Users/91960/OneDrive/Desktop/futi/futi/index.html
Programming probl...
My Dashboard - Tri...
GitHub
Practice | Geeksfor...
Programming tutori...
Problems - LeetCode
SWAYAM
IBM
Naan Mudhalvan
Diabetic.html
Normal Eye
Eye with Retinopathy
ABOUT PROJECT
PROBLEM
Diabetic Retinopathy is one of the emerging diseases which is the reason for blindness.Dr mutilates the retinal blood vessels of a patint having diabetes. Diabetic Retinopathy(DR) is an ophthalmic disease that damages retinal blood vessels.DR causes imperfect vision and may cause blindness if it is not diagnosed in earle stages.Early detection of Diabetic Retinopathy includes the identification of microaneurysms and hemorrhages.Because the signs and symptoms of Diabetic Retinopathy are typically not present during first stage of the disease, it can often go undiagnosed until the damage to vision has occured.Existing methods are lacking in the earlier detection.Because preprocessing technique used in those methods are not effective to analyze such smaller features(nearly 10 microns to 100 microns).
SOLUTION
We opt to use multi-layer neural networks as deep NN.Due to the fact that the data is image, the best type of neural network satisfying our goal is Convolutioal Neural Networks.As we have to do for the most of the data, normalization plays an important role in our process.Before doing any tasks, preprocessing images(our dataset) is highly recommended.Consequently better accuracy will be achieved by preprocessing data.After preprocessing and normalizing, the prepared dataset could be used as input to our deep Convolutional Neural Network.Then deep NN will run and fit to our data and the result will be produced by that.This report will cover step by step how this deep Convolutional Neural Network be implemented.


File | C:/Users/91960/OneDrive/Desktop/futi/futi/register.html
Programming probl...
My Dashboard - Tri...
GitHub
Practice | Geeksfor...
Programming tutori...
Problems - LeetCode
SWAYAM
IBM
Naan Mudhalvan
Diabetic.html
Diabetic Retinopathy
Home
Login
Logout
DIABETIC RETINOPATHY REGISTER
Enter Name
Enter Email
Enter Password
REGISTER
Already have an account? LOGIN


File | C:/Users/91960/OneDrive/Desktop/futi/futi/login.html
Programming probl...
My Dashboard - Tri...
GitHub
Practice | Geeksfor...
Programming tutori...
Problems - LeetCode
SWAYAM
IBM
Naan Mudhalvan
Diabetic.html
Diabetic Retinopathy
Home
Register
Logout
DIABETIC RETINOPATHY LOGIN
Enter Username
Please fill out this field.
Enter Password
Remember me
LOGIN

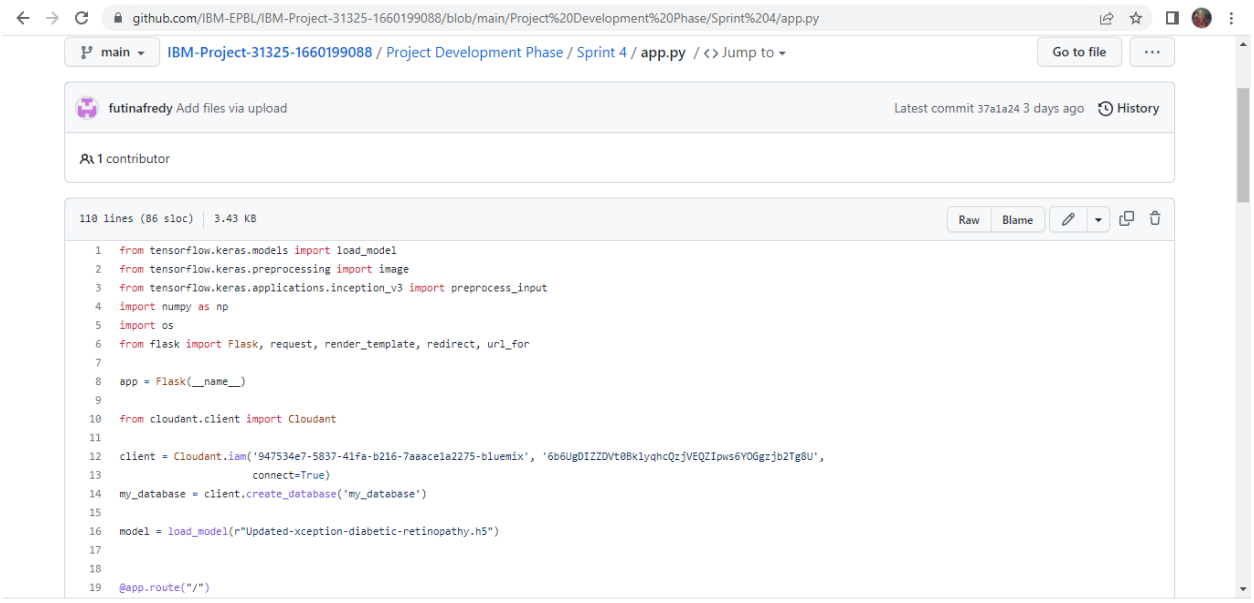




Build Python Code

- Import the libraries
- Render HTML page

- Configure the registration page
- Configure the login page
- Showcasing prediction on UI:
- Main Function:



github.com/IBM-EPBL/IBM-Project-31325-1660199088/blob/main/Project%20Development%20Phase/Sprint%204/app.py

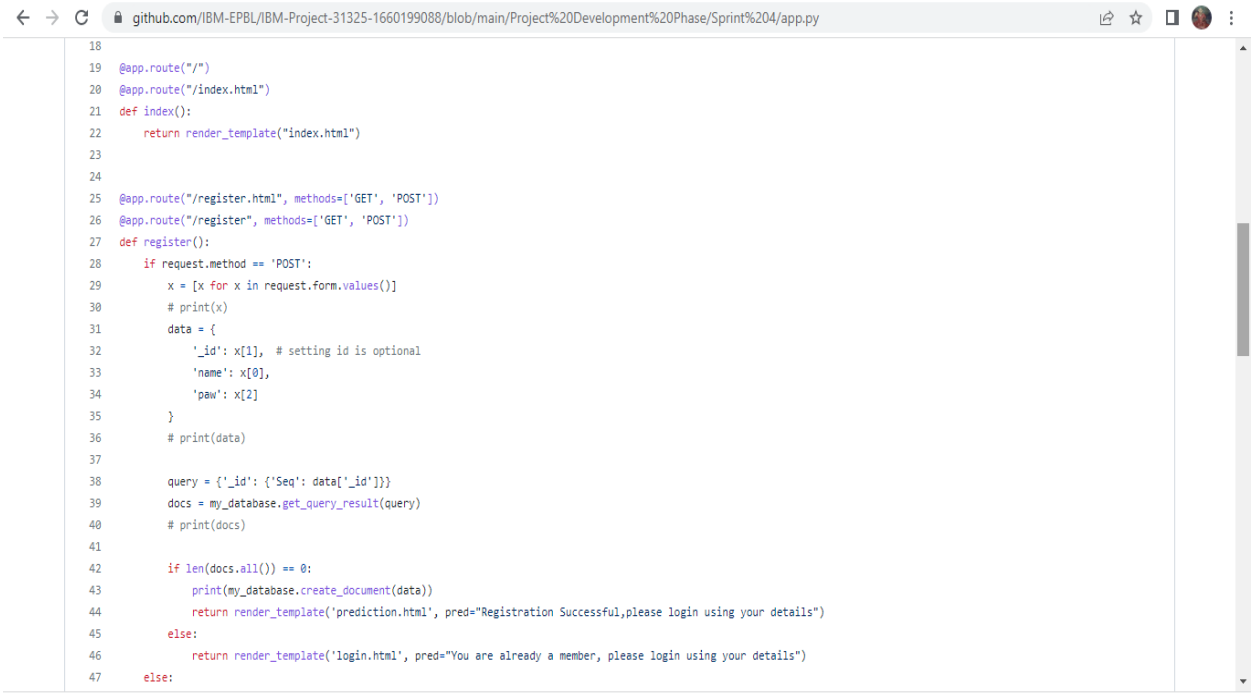
main · IBM-Project-31325-1660199088 / Project Development Phase / Sprint 4 / app.py / Jump to · Go to file

futinafredy Add files via upload — Latest commit 37a1a24 3 days ago · History

1 contributor

110 lines (86 sloc) | 3.43 KB · Raw · Blame

```python
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
import numpy as np
import os
from flask import Flask, request, render_template, redirect, url_for

app = Flask(__name__)

from cloudant.client import Cloudant

client = Cloudant.iam('947534e7-5837-41fa-b216-7aaace1a2275-bluemix', '6b6UgDIZZDVt0Bk1yqhcQzjVEQZIpws6YOGgzjb2Tg8U',
                      connect=True)
my_database = client.create_database('my_database')

model = load_model(r"Updated-xception-diabetic-retinopathy.h5")


@app.route("/")
```



github.com/IBM-EPBL/IBM-Project-31325-1660199088/blob/main/Project%20Development%20Phase/Sprint%204/app.py

```python

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


@app.route("/register.html", methods=['GET', 'POST'])
@app.route("/register", methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        x = [x for x in request.form.values()]
        # print(x)
        data = {
            '_id': x[1],  # setting id is optional
            'name': x[0],
            'paw': x[2]
        }
        # print(data)

        query = {'_id': {'Seq': data['_id']}}
        docs = my_database.get_query_result(query)
        # print(docs)

        if len(docs.all()) == 0:
            print(my_database.create_document(data))
            return render_template('prediction.html', pred="Registration Successful,please login using your details")
        else:
            return render_template('login.html', pred="You are already a member, please login using your details")
    else:
```

```
        return render_template("register.html")


# Login page
@app.route('/login', methods=['GET', 'POST'])
@app.route('/login.html', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        user = request.form['_id']
        passw = request.form['psw']
        print(user, passw)

        query = {'_id': {'$eq': user}}

        docs = my_database.get_query_result(query)
        print(docs.all())

        if len(docs.all()) == 0:
            return render_template('login.html', pred="The username is not found.")
        else:
            if user == docs[0][0]['_id'] and passw == docs[0][0]['paw']:
                return redirect(url_for('prediction'))
            else:
                print('Invalid User')

    else:
        return render_template('login.html')


# Logout
```

github.com/IBM-EPBL/IBM-Project-31325-1660199088/blob/main/Project%20Development%20Phase/Sprint%204/app.py

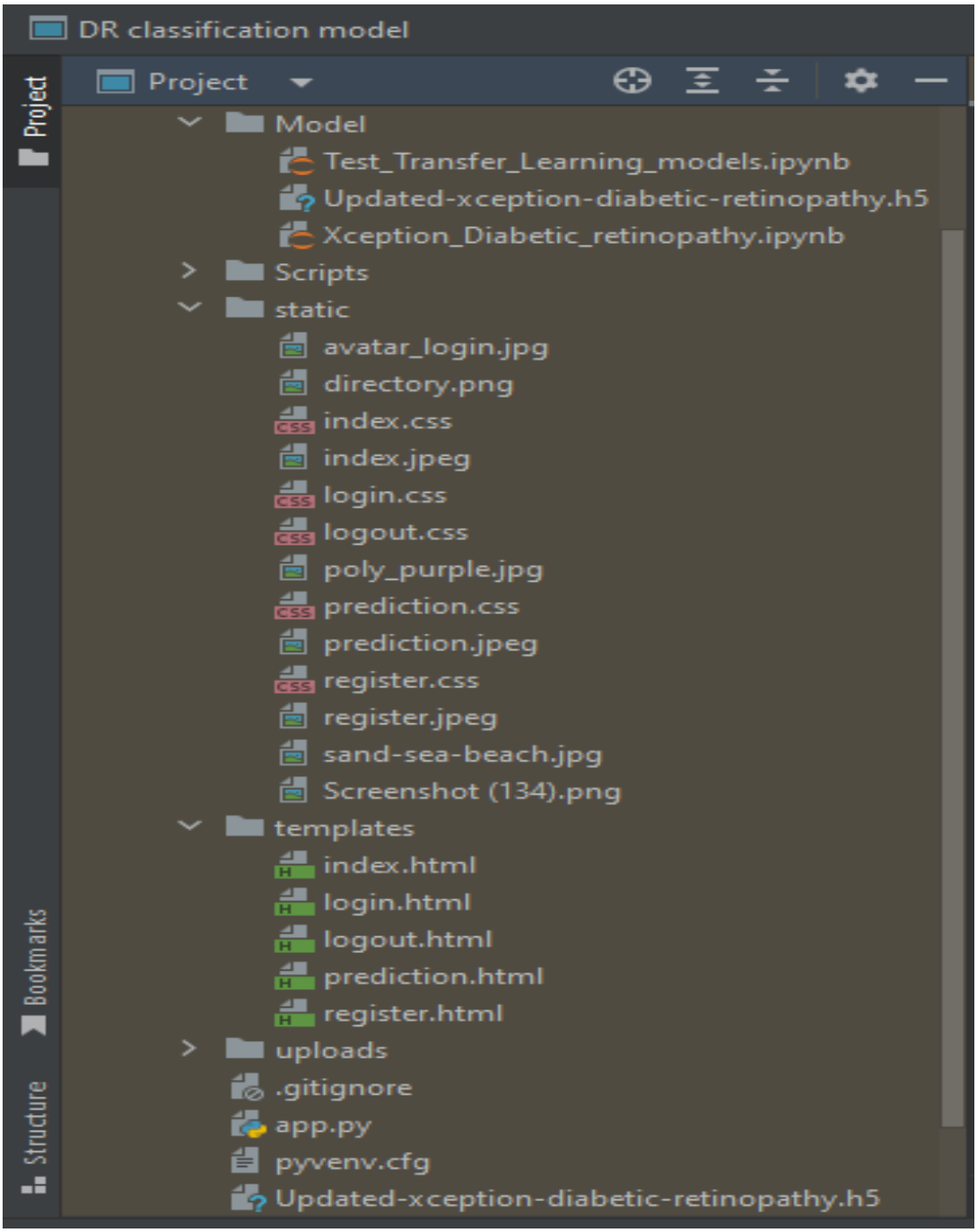
```
@app.route('/logout')
@app.route('/logout.html')
def logout():
    return render_template('logout.html')


# Prediction page
@app.route('/prediction', methods=["GET", "POST"])
def prediction():
    if request.method == 'POST':
        print(request.files.keys())
        f = request.files['image']
        basepath = os.path.dirname(__file__)
        filepath = os.path.join(basepath, 'uploads',
                                f.filename)
        f.save(filepath)
        img = image.load_img(filepath, target_size=(299, 299))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        img_data = preprocess_input(x)
        prediction = np.argmax(model.predict(img_data), axis=1)
        index = ['No Diabetic Retinopathy, Just Relax', 'Mild DR, Time for a basic checkup', 'Moderate DR, Condult a doctor', 'Severe DR,Check with your doctor immediately', '
        result = str(index[prediction[0]])
        return render_template('prediction.html', prediction=result)
    else:
        return render_template('prediction.html')


#
```

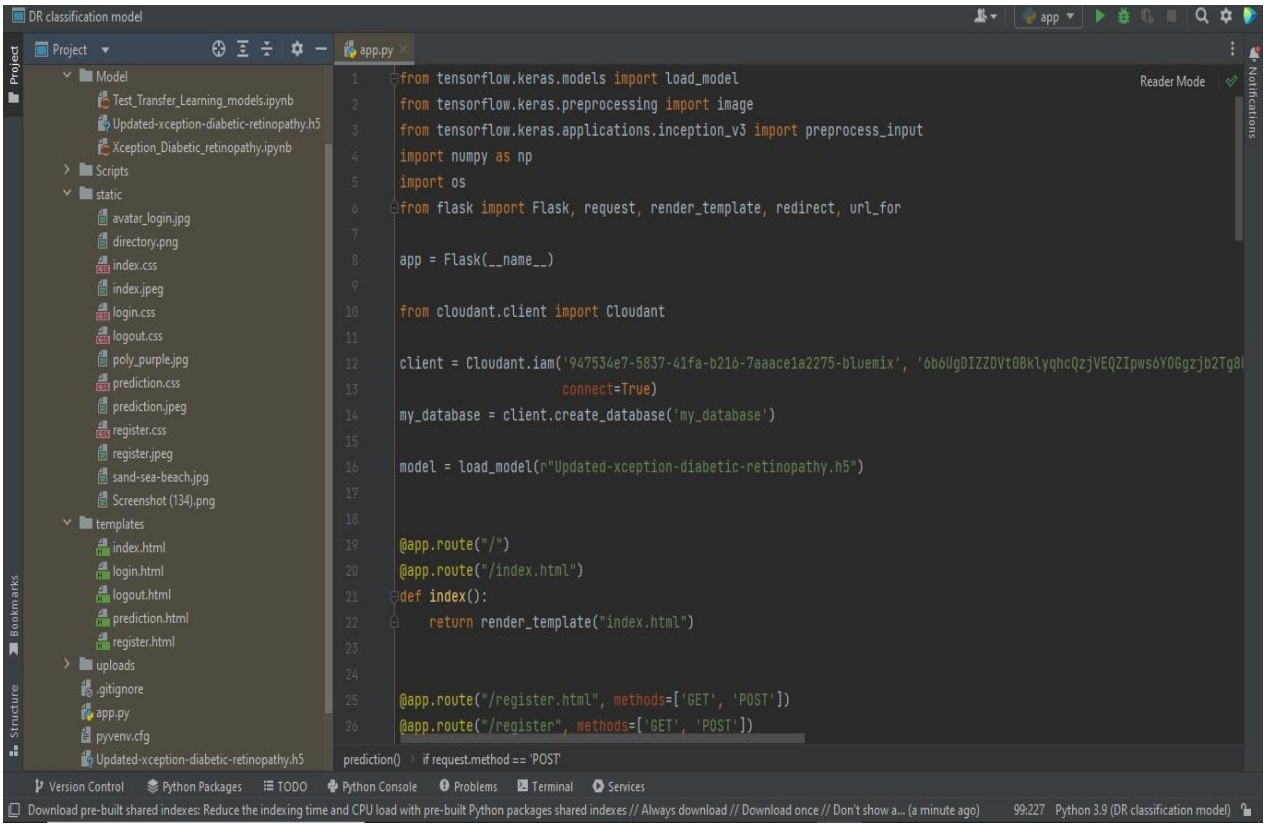
github.com/IBM-EPBL/IBM-Project-31325-1660199088/blob/main/Project%20Development%20Phase/Sprint%204/app.py

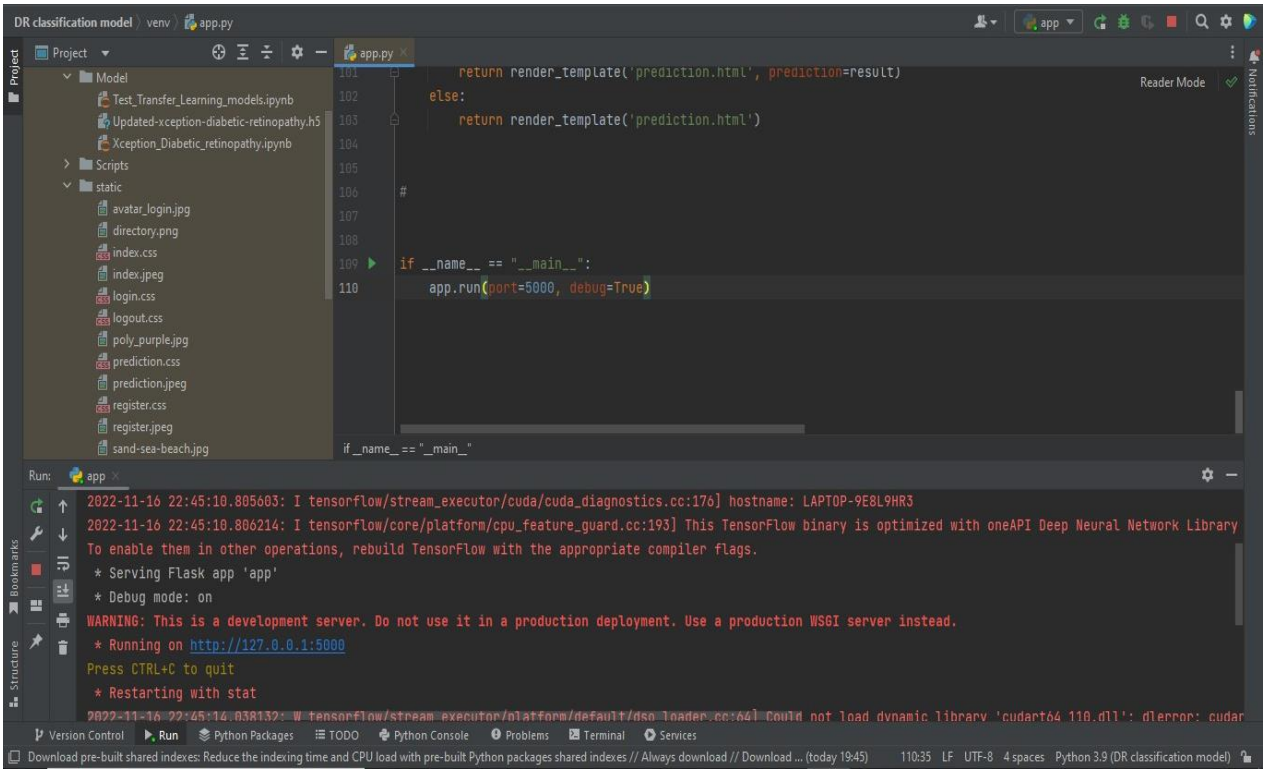
```
        basepath = os.path.dirname(__file__)
        filepath = os.path.join(basepath, 'uploads',
                                f.filename)
        f.save(filepath)
        img = image.load_img(filepath, target_size=(299, 299))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        img_data = preprocess_input(x)
        prediction = np.argmax(model.predict(img_data), axis=1)
        index = ['No Diabetic Retinopathy, Just Relax', 'Mild DR, Time for a basic checkup', 'Moderate DR, Condult a doctor', 'Severe DR,Check with your doctor immediately', '
        result = str(index[prediction[0]])
        return render_template('prediction.html', prediction=result)
    else:
        return render_template('prediction.html')


#


if __name__ == "__main__":
    app.run(port=5000, debug=True)
```

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

- Run The Application




DR classification model
app
Project
app.py
Reader Mode
Model
Test_Transfer_Learning_models.ipynb
Updated-xception-diabetic-retinopathy.h5
Xception_Diabetic_retinopathy.ipynb
Scripts
static
avatar_login.jpg
directory.png
index.css
index.jpeg
login.css
logout.css
poly_purple.jpg
prediction.css
prediction.jpeg
register.css
register.jpeg
sand-sea-beach.jpg
Screenshot (134).png
templates
index.html
login.html
logout.html
prediction.html
register.html
uploads
.gitignore
app.py
pyvenv.cfg
Updated-xception-diabetic-retinopathy.h5
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
import numpy as np
import os
from flask import Flask, request, render_template, redirect, url_for
app = Flask(__name__)
from cloudant.client import Cloudant
client = Cloudant.iam('947534e7-5837-41fa-b216-7aaace1a2275-bluemix', '6b6UgDIZZDVtOBklyqhcQzjVEQZIpws6YOGgzjb2Tg8'
connect=True)
my_database = client.create_database('my_database')
model = load_model(r"Updated-xception-diabetic-retinopathy.h5")
@app.route("/")
@app.route("/index.html")
def index():
return render_template("index.html")
@app.route("/register.html", methods=['GET', 'POST'])
@app.route("/register", methods=['GET', 'POST'])
prediction() > if request.method == 'POST'
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show a... (a minute ago)
99:227
Python 3.9 (DR classification model)


DR classification model
venv
app.py
app
Project
app.py
Reader Mode
Model
Test_Transfer_Learning_models.ipynb
Updated-xception-diabetic-retinopathy.h5
Xception_Diabetic_retinopathy.ipynb
Scripts
static
avatar_login.jpg
directory.png
index.css
index.jpeg
login.css
logout.css
poly_purple.jpg
prediction.css
prediction.jpeg
register.css
register.jpeg
sand-sea-beach.jpg
return render_template('prediction.html', prediction=result)
else:
return render_template('prediction.html')
#
if __name__ == "__main__":
app.run(port=5000, debug=True)
if __name__ == "__main__"
Run:
app
2022-11-16 22:45:10.805603: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-9E8L9HR3
2022-11-16 22:45:10.806214: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* 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
2022-11-16 22:45:14.038132: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudar
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download ... (today 19:45)
110:35
LF
UTF-8
4 spaces
Python 3.9 (DR classification model)