

App.py

```
import os
from os.path import join, dirname
from dotenv import load_dotenv
from functools import wraps
from http.client import HTTPException
import numpy as np
from flask import Flask, request, render_template, session,
url_for, redirect, flash
import pickle
import inputScript
import pymongo
from passlib.hash import pbkdf2_sha256
import json
import inputScript

app = Flask(__name__, template_folder='../Flask')
model = pickle.load(open('../Flask/Phishing_Website.pkl', 'rb'))

dotenv_path = join(dirname(__file__), '.env')
load_dotenv(dotenv_path)
MONGODB_URL = os.environ.get("MONGODB_URL")
SECRET_KEY = os.environ.get("SECRET_KEY")
```

```
mongoDB=pymongo.MongoClient(MONGODB_URL)
db=mongoDB['Web_Phishing_Detection']
account=db.account
app.secret_key= SECRET_KEY
```

```
carouselDataFile = open('./static/json/carouselData.json')
carouselData = json.load(carouselDataFile)
aboutDataFile = open('./static/json/aboutData.json')
aboutData = json.load(aboutDataFile)
```

```
def login_required(f):
    @wraps(f)
    def wrap(*args, **kwargs):
        if('logged_in' in session):
            return f(*args, **kwargs)
        else:
            return redirect('/')
    return wrap
```

```
def start_session(userInfo):
    if userInfo:
        userInfo['_id']=str(userInfo['_id'])
```

else:

raise HTTPException(status_code=404, detail=f"Unable to retrieve record")

del userInfo['password']

session['logged_in']=True

session['user']=userInfo

return redirect(url_for('index'))

@app.route('/login/',methods=['POST'])

def login():

if request.method=="POST":

email=request.form.get("email")

password=request.form.get("password")

if(account.find_one({"email":email})):

user=account.find_one({"email":email})

if(user and pbkdf2_sha256.verify(password,user['password'])):

return start_session(user)

else:

flash("Password is incorrect","loginError")

return redirect(url_for('index',loginError=True))

flash("Sorry, user with this email id does not exist","loginError")

return redirect(url_for('index',loginError=True))

```
@app.route('/signup/',methods=['POST'])
def signup():
    if request.method=="POST":
        userInfo={
            "fullName":request.form.get('fullName'),
            "email":request.form.get('email'),
            "phoneNumber":request.form.get('phoneNumber'),
            "password":request.form.get('password'),
        }

        userInfo['password']=pbkdf2_sha256.encrypt(userInfo['password'])
        if(account.find_one({"email":userInfo['email']})):
            flash("Sorry,user with this email already exist","signupError")
            return redirect(url_for('index',signupError=True))
        if(account.insert_one(userInfo)):
            return start_session(userInfo)
        flash("Signup failed","signupError")
        return redirect(url_for('index',signupError=True))
```

```
@app.route('/logout/',methods=["GET"])
def logout():
    if request.method=="GET":
        session.clear()
        return redirect(url_for('index'))
@app.route('/')
```

```
def index():

    if(session and '_flashes' in dict(session)):
        loginError=request.args.get('loginError')
        signupError=request.args.get('signupError')
        if(loginError):
            return
        render_template('./index.html',loginError=loginError,carousel_content=carouselData['carousel_content'])
        if(signupError):
            return
        render_template('./index.html',signupError=signupError,carousel_content=carouselData['carousel_content'])
    if(session and '_flashes' not in dict(session)):
        print(dict(session))
        if(session['logged_in']==True):
            return
        render_template('./index.html',userInfo=session['user'],carousel_content=carouselData['carousel_content'])
        else:
            return
        render_template('./index.html',carousel_content=carouselData['carousel_content'])
    else:
        return
        render_template('./index.html',carousel_content=carouselData['carousel_content'])
```

```

@app.route('/predict/', methods=['GET','POST'])
@login_required
def predict():
    if request.method == 'POST':
        url = request.form['url']
        checkprediction = inputScript.main(url)
        print(url)
        print(checkprediction)
        prediction = model.predict(checkprediction)
        print(prediction)
        output=prediction[0]
        if(output==1):
            pred="Safe,legitimate link"

        else:
            pred="Malicious URL alert!"
        if(session and session['logged_in']):
            if(session['logged_in']==True):
                return render_template('./templates/prediction-
result.html',userInfo=session['user'],pred=pred)
            # else:
            #     return render_template('./templates/prediction-
result.html',pred=pred)
        # else:

```

```
    # return render_template('./templates/prediction-  
result.html',pred=pred)
```

```
    elif request.method == 'GET':
```

```
        return render_template('./templates/predict-  
form.html',userInfo=session['user'])
```

```
@app.route('/about/')
```

```
def about():
```

```
    if(session and session['logged_in']):
```

```
        if(session['logged_in']==True):
```

```
            return
```

```
render_template('./templates/about.html',userInfo=session['user'],ab  
outContents=aboutData['aboutContents'])
```

```
    else:
```

```
        return
```

```
render_template('./templates/about.html',aboutContents=aboutData[  
'aboutContents'])
```

```
    else:
```

```
        return
```

```
render_template('./templates/about.html',aboutContents=aboutData[  
'aboutContents'])
```

```
@app.route('/contact/')
```

```
def contact():
```

```
    if(session and session['logged_in']):
```

```
        if(session['logged_in']==True):
```

```
        return
render_template('./templates/contact.html',userInfo=session['user'])
    else:
        return render_template('./templates/contact.html')
else:
    return render_template('./templates/contact.html')
```

```
# @app.route('/predict/', methods=['POST'])
# def y_predict():
#     url = request.form['URL']
#     checkprediction = inputScript.main(url)
#     prediction = model.predict(checkprediction)
#     print(prediction)
#     output=prediction[0]
#     if(output==1):
#         pred="Your are safe!! This is a Legitimate Website."

#     else:
#         pred="You are on the wrong site. Be cautious!"
#     #return render_template('final.html',
# prediction_text='{}'.format(pred),url=url)
#     flash(pred)

if __name__ == '__main__':
```



```
app.run(host='127.0.0.1', debug=True)
```