

```

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

```

```

    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'),
            "phone Number":request.form.get('phone Number')

```

```

        "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'])
            ,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)

```

```

126     prediction = model.predict(checkprediction)
127     print(prediction)
128     output=prediction[0]
129     if(output==1):
130         pred="Safe,legitimate link"
131
132     else:
133         pred="Malicious URL alert!"
134     if(session and session['logged_in']):
135         if(session['logged_in']==True):
136             return render_template('./templates/prediction-result
                .html',userInfo=session['user'],pred=pred)
137     # else:
138     #     return render_template('./templates/prediction-result
                .html',pred=pred)
139     # else:
140     #     return render_template('./templates/prediction-result.html'
                ,pred=pred)
141     elif request.method == 'GET':
142         return render_template('./templates/predict-form.html',userInfo
                =session['user'])
143
144 @app.route('/about/')
145 def about():
146     if(session and session['logged_in']):
147         if(session['logged_in']==True):
148             return render_template('./templates/about.html',userInfo
                =session['user'],aboutContents=aboutData['aboutContents']
                )
149     else:
150         return render_template('./templates/about.html',aboutContents
                =aboutData['aboutContents'])
151     else:
152         return render_template('./templates/about.html',aboutContents
                =aboutData['aboutContents'])
153
154 @app.route('/contact/')
155 def contact():
156     if(session and session['logged_in']):
157         if(session['logged_in']==True):
158             return render_template('./templates/contact.html'
                ,userInfo=session['user'])
159     else:
160         return render_template('./templates/contact.html')
161     else:
162         return render_template('./templates/contact.html')
163
164 # @app.route('/predict/', methods=['POST'])
165 # def y_predict():
166 #     url = request.form['URL']
167 #     checkprediction = inputScript.main(url)

```