```
import datetime
from flask import jsonify
from flask import Flask, render_template, request
from cloudant.client import Cloudant

client = Cloudant.iam('d9b401b4-6c0f-4740-9da7-4376a6dc8fdf-bluemix',
'TsO1xlMzArJKQ1vqNv08hxxraWpbSt9lOWNxtAHvYGv8',
                      connect=True)
my_database = client.create_database('my_database')
app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '"083458892a3c1ab6f18660a9cfeae6f5c'


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


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


@app.route("/addamount")
@app.route("/register")
def NewUser():
    return render_template('register.html')
```

```
@app.route("/login")
def user():
    return render_template('login.html')



@app.route("/newuse", methods=['GET', 'POST'])
def newuse():
    if request.method == 'POST':
        x = [x for x in request.form.values()]
        print(x)
        data = {
            '_id': x[1],
            'name': x[0],
            'psw': x[2]
        }
        print(data)
        query = {'_id': {'Seq': data['_id']}}
        docs = my_database.get_query_result(query)
        print(docs)
        print(len(docs.all()))
        if (len(docs.all()) == 0):
            url = my_database.create_document(data)
            return render_template('login.html', data="Register, please login using your details")
        else:
            return render_template('register.html', data="You are already a member, please login using your
details")
```

```
@app.route("/userlog", methods=['GET', 'POST'])
def userlog():
    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)
        print(len(docs.all()))
        if len(docs.all()) == 0:
            return render_template('goback.html', pred="The username is not found.")
        else:
            if user == docs[0][0]['_id'] and passw == docs[0][0]['psw']:
                return render_template("index.html")
            else:
                return render_template('goback.html', data="user name and password incorrect")



@app.route("/predict", methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        file = request.files['fileupload']
        DateTimeMilliSeconds = datetime.datetime.now().strftime("%Y%m%d_%H%M%S_%f")
        image_file_path = r'media/images/DamageImage_{}.jpg'.format(DateTimeMilliSeconds)
        file.save(image_file_path)

        import tensorflow as tf
        import numpy as np
```

```
import warnings

warnings.filterwarnings('ignore')

test_image = tf.keras.preprocessing.image.load_img(image_file_path, target_size=(200, 200))

# test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis=0)
# DAMAGE_COST MODEL
classifierLoad = tf.keras.models.load_model(r'model/body.h5')
result = classifierLoad.predict(test_image)
result1 = ''
if result[0][0] == 1:
   result1 = "front"
elif result[0][1] == 1:
   result1 = "rear"
elif result[0][2] == 1:
   result1 = "side"
print('[INFO!!]', result1)

##########################################
# file = request.files['fileupload1']
# DateTimeMilliSeconds = datetime.datetime.now().strftime("%Y%m%d_%H%M%S_%f")
# image_file_path = r'media/images/DamageType_{}.jpg'.format(DateTimeMilliSeconds)
# file.save(image_file_path)
# test_image = tf.keras.preprocessing.image.load_img(
#     r'C:\Users\Macro\Downloads\Car damage\level\validation\03-severe\0017.JPEG',
target_size=(200, 200))
#
```

```
# test_image = np.expand_dims(test_image, axis=0)
################################
# Damage_type Model
classifierLoad = tf.keras.models.load_model(r'model/level.h5')
result = classifierLoad.predict(test_image)
result2 = ''
if result[0][0] == 1:
   result2 = "minor"
elif result[0][1] == 1:
   result2 = "moderate"
elif result[0][2] == 1:
   result2 = "severe"
print('[INFO!!]', result2)
if result1 == "front" and result2 == "minor":
   value = "3000 - 5000 INR"
elif result1 == "front" and result2 == "moderate":
   value = "6000 - 8000 INR"
elif result1 == "front" and result2 == "severe":
   value = "9000 - 11000 INR"
elif result1 == "rear" and result2 == "minor":
   value = "4000 - 6000 INR"
elif result1 == "rear" and result2 == "moderate":
   value = "7000 - 9000 INR"
elif result1 == "rear" and result2 == "severe":
   value = "11000 - 13000 INR"
elif result1 == "side" and result2 == "minor":
   value = "6000 - 8000 INR"
elif result1 == "side" and result2 == "moderate":
   value = "9000 - 11000 INR"
```

```
        elif result1 == "side" and result2 == "severe":
            value = "12000 - 15000 INR"
        else:
            value = "16000 - 50000 INR"
        print('[INFO!!] Damage Cost Range: ', value)
        # Please comment this return and uncomment the 'render_template' in 147 line
        return jsonify({'Damage Cost Range': value, 'Damage_angle': result1, 'Damage_type': result2})
        # return render_template('userhome.html', prediction=value)



if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```