```
from flask
import Flask,
render_template,
flash,
request,session
                  from cloudant.client import  Cloudant

                  import cv2

                  client = Cloudant.iam("eb55a2b7-ae45-4df8-8d1c-69c5229ffdbe-
                  bluemix","YzG5FZg9Vs_HScOBZaWyVXm7PpNjbPrmPaPMfHx7w3X9",connect=True)
                  my_database = client.create_database("database-dharan")


                  app = Flask(__name__)
                  app.config.from_object(__name__)
                  app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'



                  @app.route("/")
                  def homepage():

                      return render_template('index.html')



                  @app.route("/userhome")
                  def userhome():

                      return render_template('userhome.html')
                  @app.route("/addamount")

                  @app.route("/NewUser")
                  def NewUser():

                      return render_template('NewUser.html')
```

```
@app.route("/user")
def user():

    return render_template('user.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('goback.html', data="Register, please
login using your details")
        else:
            return render_template('goback.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("userhome.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']
        file.save('static/Out/Test.jpg')

        import warnings
        warnings.filterwarnings('ignore')

        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('body.h5')

        import numpy as np
        from keras.preprocessing import image

        test_image = image.load_img('static/Out/Test.jpg',
target_size=(200, 200))
        img1 = cv2.imread('static/Out/Test.jpg')
        # test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
        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"



        file = request.files['fileupload1']
        file.save('static/Out/Test1.jpg')

        import warnings
        warnings.filterwarnings('ignore')

        import tensorflow as tf
        classifierLoad = tf.keras.models.load_model('level.h5')

        import numpy as np
        from keras.preprocessing import image

        test_image = image.load_img('static/Out/Test1.jpg',
target_size=(200, 200))
        img1 = cv2.imread('static/Out/Test1.jpg')
        # test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
        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"



        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"


        return render_template('userhome.html', prediction=value)



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