In [ ]:

```
#importing libraries
import pandas as pd
from flask import Flask, request, render_template, send_file
import pickle
import zipfile
from tensorflow.keras.models import load_model

app = Flask(__name__)

#loading preprocessed files
transformer = pickle.load(open('transform.pkl','rb'))
encoder = load_model('encoder.h5')
model = pickle.load(open('final_model.pkl','rb'))
columns = pickle.load(open('target_columns.pkl','rb'))

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

@app.route('/Predict', methods = ['GET','POST'])
def predict_moa():
    try:
        file = request.files['search_file']
        X = pd.read_csv(file, header=0)

        #encoding categorical features
        X.iloc[:,1] = X.iloc[:,1].map({'trt_cp':0, 'ctl_vehicle':1})
        X.iloc[:,2] = X.iloc[:,2].map({24:0, 48:1, 72:2})
        X.iloc[:,3] = X.iloc[:,3].map({'D1':0, 'D2':1})

        #normalizing gene and cell columns
        X.iloc[:,4:] = transformer.transform(X.iloc[:,4:])

        #getting more features using auto-encoder
        encoded_features = encoder.predict(X.iloc[:,1:])

        #adding encoded features with original features
        total_X = pd.concat([X.reset_index(drop=True),pd.DataFrame(encoded_features).reset_index(drop=True)], axis=1)

        #predictions
        pred = model.predict(total_X.iloc[:,1:])
        pred_data = pd.DataFrame(pred, columns = columns)
        pred_data.insert(loc=0,column='sig_id',value = X.index)
        pred_data.to_csv('pred_data.csv',index=False)

        pred_prob = model.predict_proba(total_X.iloc[:,1:])
        pred_prob_data = pd.DataFrame(pred_prob,columns=columns)
        pred_prob_data.insert(loc=0,column='sig_id',value = X.index)
        pred_prob_data.to_csv('pred_prob_data.csv',index=False)

        #zipping prediction and probability prediction
        zipf = zipfile.ZipFile('Predictions.zip','w', zipfile.ZIP_DEFLATED)
        zipf.write('pred_data.csv')
        zipf.write('pred_prob_data.csv')
        zipf.close()

        return send_file('Predictions.zip', mimetype='zip', as_attachment=True, attachment_filename='moa_predictions.zip')

    #if file is not same as in instruction then exception will be thrown
    except:
        err = "Please upload file in correct format as shown in instruction and try again"
        return render_template('index.html',error=err)

@app.route('/Important Instructions')
def imp_instruction():
    return render_template('instruction.html')

@app.route('/download')
def download_file():
    return send_file('sample_input.csv', as_attachment=True)

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