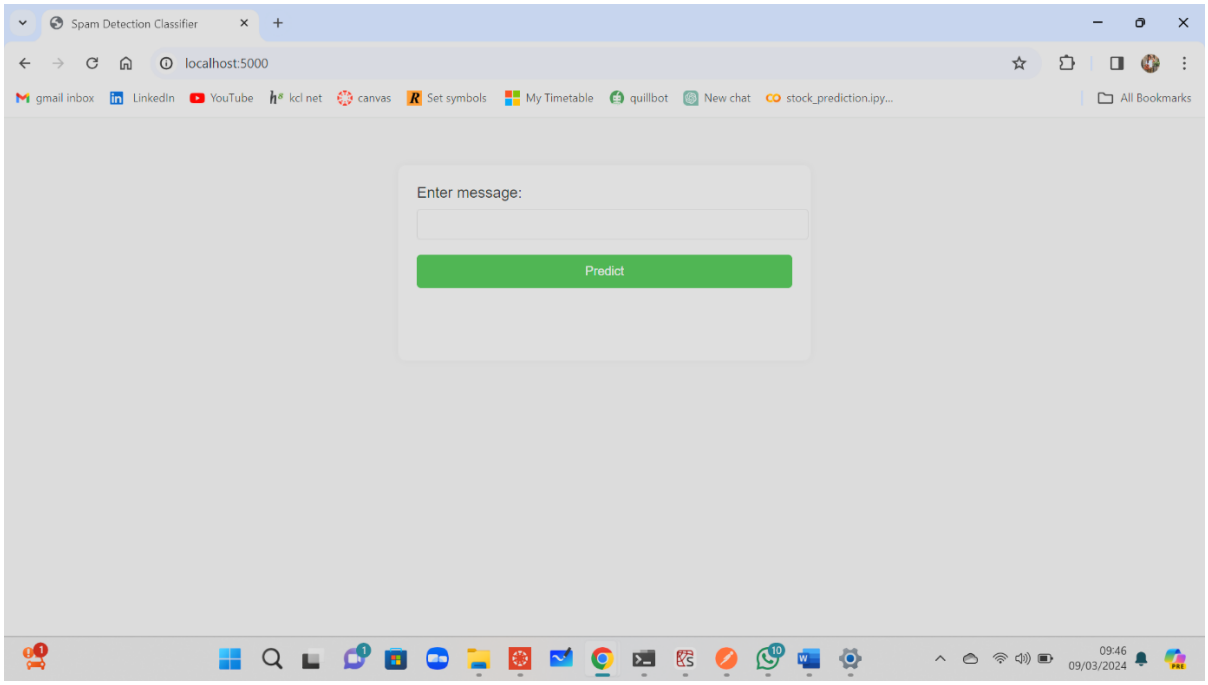
Name: Krishnaja Koonapra

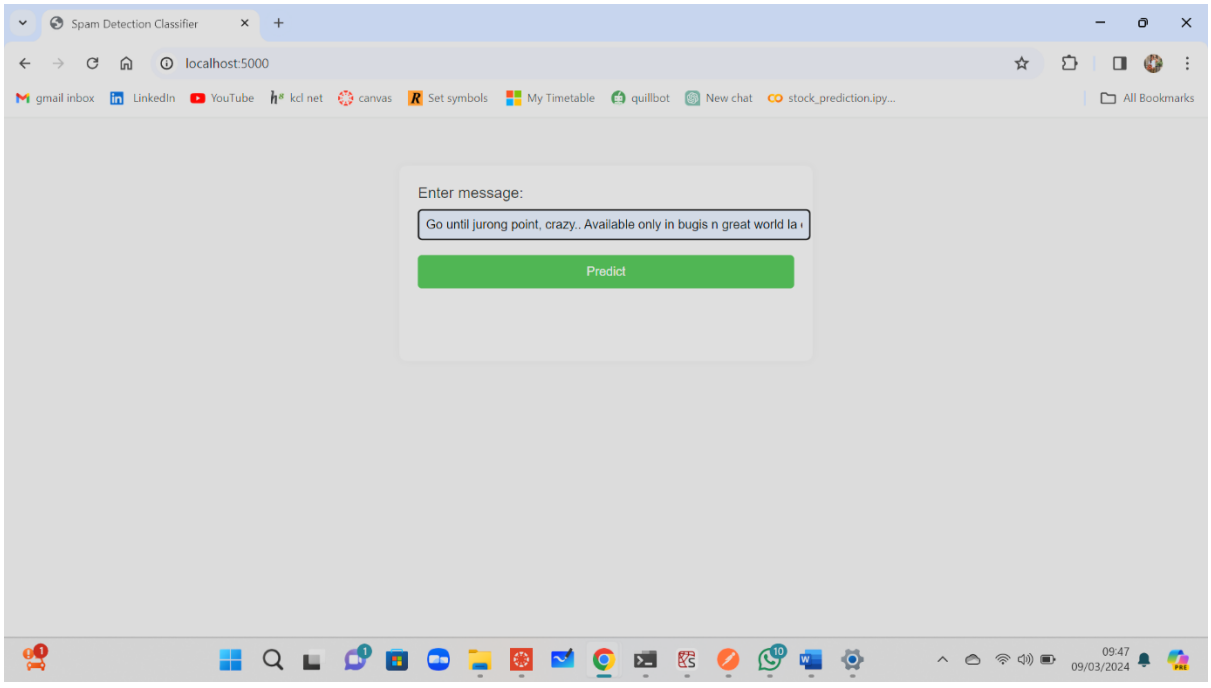
Batch Code: LISUM30

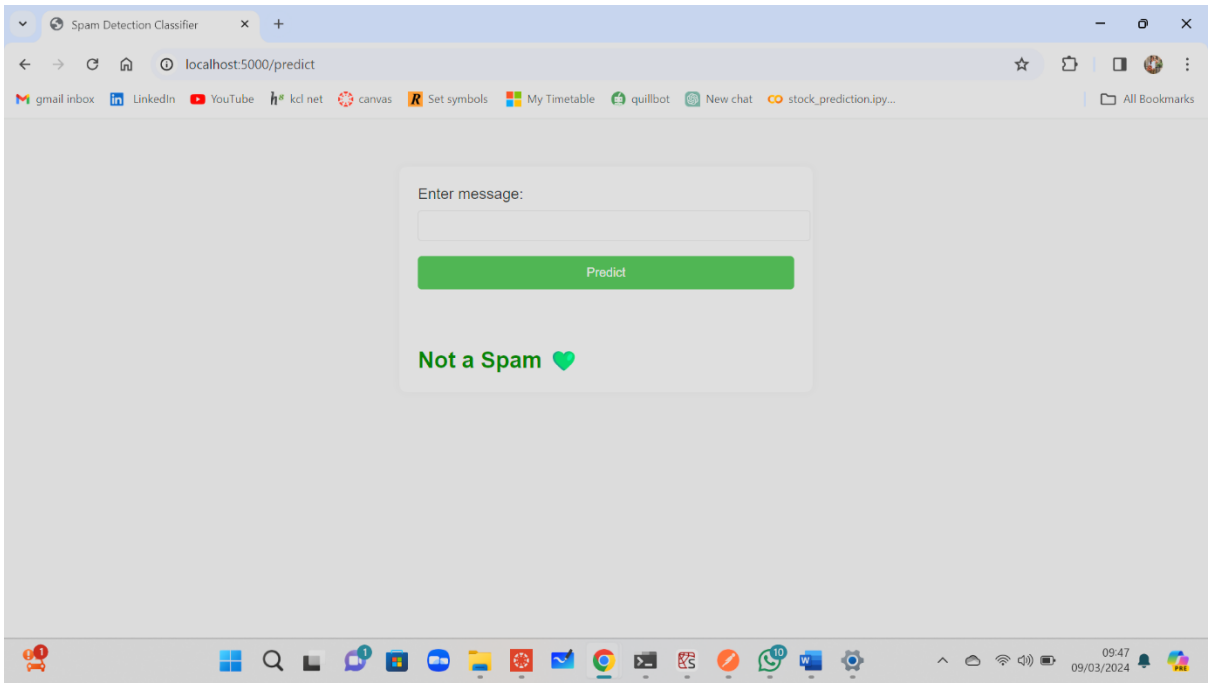
Submission Date: 28 Feb 2024

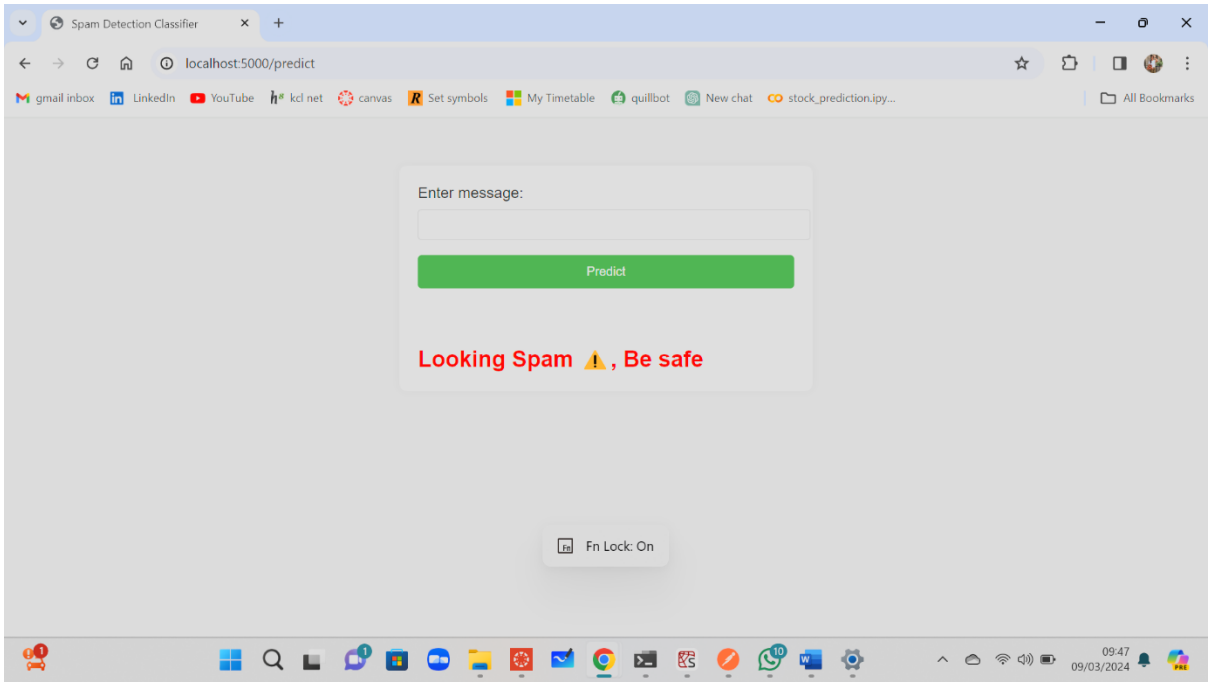
Submitted To: Data Glacier

# SPAM DETECTION CLASSIFIER




Spam Detection Classifier
localhost:5000/predict
gmail inbox
LinkedIn
YouTube
kcl net
canvas
Set symbols
My Timetable
quillbot
New chat
stock_prediction.ipy...
All Bookmarks
Enter message:
Predict
Not a Spam
09:47
09/03/2024


Spam Detection Classifier
localhost:5000/predict
gmail inbox
LinkedIn
YouTube
kcl net
canvas
Set symbols
My Timetable
quillbot
New chat
stock_prediction.ipy...
All Bookmarks
Enter message:
Predict
Looking Spam ⚠, Be safe
Fn Lock: On
09:47
09/03/2024

```python
#  Model building
cv = CountVectorizer()
tfidf = TfidfVectorizer(max_features = 3000)
X = tfidf.fit_transform(df['transformed_text']).toarray()
y = df['target'].values
X_train, X_test , y_train, y_test = train_test_split(X,y,test_size = 0.20,
print(X_train[1])
# Initialise the models
svc = SVC(kernel= "sigmoid", gamma  = 1.0)

svc.fit(X_train,y_train)
print(X_train)
print(X_train.shape)
print(X_test)
print(X_test.shape)

y_pred = svc.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
print(accuracy , precision)

# Saving model
pickle.dump(tfidf, open('vectoriser.pkl', 'wb'))
pickle.dump(svc, open('model.pkl', 'wb'))
```

```html
<div class="login">
    <form action="{{ url_for('predict') }}" method="post">
        <label for="message">Enter message:</label>
        <input type="text" id="message" name="message" />
        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

    <br>
    <br>

    <div class="results">
        {% if prediction_text == 1 %}
            <h2 class="color-red">Looking Spam ⚠️, Be safe</h2>
        {% elif prediction_text == 0 %}
            <h2 class="color-green"><b>Not a Spam 💚</b></h2>
        {% endif %}
    </div>
</div>
```

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
import nltk
from nltk.corpus import stopwords    # For stopwords
from nltk.stem.porter import PorterStemmer

# Importing the string module for handling special characters
import string

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
tfidf = pickle.load(open('vectoriser.pkl', 'rb'))
@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    message = request.form['message']


    # Creating an instance of the Porter Stemmer
    ps = PorterStemmer()
    def transform_text(text):
        # Transform the text to lowercase
        text = text.lower()
        # Tokenization using NLTK
        text = nltk.word_tokenize(text)
        # Removing special characters
```

```python
    y = []
    for i in text:
        if i.isalnum():
            y.append(i)

    # Removing stop words and punctuation
    text = y[:]
    y.clear()

    # Loop through the tokens and remove stopwords and punctuation
    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)

    # Stemming using Porter Stemmer
    text = y[:]
    y.clear()
    for i in text:
        y.append(ps.stem(i))

    # Join the processed tokens back into a single string
    return " ".join(y)
transformed_text = transform_text(message)
vector_input = tfidf.transform([transformed_text]).toarray()
prediction = model.predict(vector_input)


return render_template('index.html', prediction_text=prediction)
```