SMS SPAM DETECTION USING IBM WATSON

AN INDUSTRY ORIENTED MINI PROJECT REPORT
Submitted to
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE

AND ENGINEERING

Submitted by

SYED ASIMUDDIN	**19UK1A0517**
MATTAPALLY AMRUSHA	**19UK1A0508**
KULLA ALEKYA	**19UK1A0513**
DOMMETI SAIKUMAR	**19UK1A0540**

Under the esteemed guidance of

S. ANOOSHA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
(Affiliated to JNTUH, Hyderabad)
Bollikunta, Warangal – 506005
2019– 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005
2019 – 2023

<u>CERTIFICATE OF COMPLETION</u>

<u>INDUSTRY ORIENTED MINI PROJECT</u>

This is to certify that UG project phase-1 entitled **"SMS SPAM DETECTION USING IBM WATSON"** is being submitted by **SYEDASIMUDDIN(19UK1A0517)**,**MATTAPALLY AMRUSHA(19UK1A0508),KULLAALEKYA(19UK1A0513),DOMMETISAIKUMAR(19UK1A0 540)** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** during the academic year **2022-2023,** is a record of work carried out by them under the guidance and supervision.

Project Guide   **Head of the Department**

S. ANOOSHA   **Dr. R. Naveen kumar**

(Assistant professor)   (professor)

ACKNOWLEDGEMENT

SYED ASIMUDDIN	**19UK1A0517**
MATTAPALLY AMRUSHA	**19UK1A0508**
KULLA ALEKYA	**19UK1A0513**
DOMMETI SAIKUMAR	**19UK1A0540**

TABLE OF CONTENTS

REPORT OF SMS SPAM DETECTION USING WATSON

1.INTRODUCTION

1.1 Overview

In the modern world where digitization is everywhere, SMS has become one of the most vital forms of communications, unlike other chatting-based messaging systems like Facebook, WhatsApp etc, SMS does not require active inter n et connect ion a t all. As we all know that Hackers / Spammer tries to intrude in Mobile Computing Device, and SMS support for mobile devices had become vulnerable, as attacker tries to intrude to the system by sending unwanted link, with which on clicking those link the attacker can gain remote access over the mobile computing device. So, to identify those messages Authors have developed a system which will identify such malicious messages and will identify whether or not the message is SPAM or HAM (malicious or not malicious). Authors have created a dictionary using the TF-IDF Vectorizer algorithm, which will include all the features of words a SPAM SMS possess, based on content of message and referring to this dictionary the system will be classifying the SMS as spam or ham.

1.2 Purpose

To avoid such Spam SMS people use white and black list of numbers. But this technique is not adequate to completely avoid Spam SMS. To tackle this problem it is needful to use a smarter technique which correctly identifies Spam SMS. Natural language processing technique is useful for Spam SMS identification. It analyses text content and finds patterns which are used to identify Spam and Non-Spam SMS.

# 2. LITERATURE SURVEY

2.1 Existing problem

Over recent years, as the popularity of mobile phone devices has increased, Short Message Service (SMS) has grown into a multi-billion dollar industry. At the same time, reduction in the cost of messaging services has resulted in growth in unsolicited commercial advertisements (spams) being sent to mobile phones. Due to Spam SMS, Mobile service providers suffer from some sort of financial problems as well as it reduces calling time for users. Unfortunately, if the user accesses such Spam SMS they may face the problem of virus or malware. When SMS arrives at mobile it will disturb mobile user privacy and concentration. It may lead to frustration for the user. So Spam SMS is one of the major issues in the wireless communication world and it grows day by day.

2.2 Proposed solution

Natural language processing technique is useful for Spam SMS identification. It analyses text content and finds patterns which are used to identify Spam and Non-Spam SMS.

3.THEORITICAL ANALYSIS

3.1 Block diagram
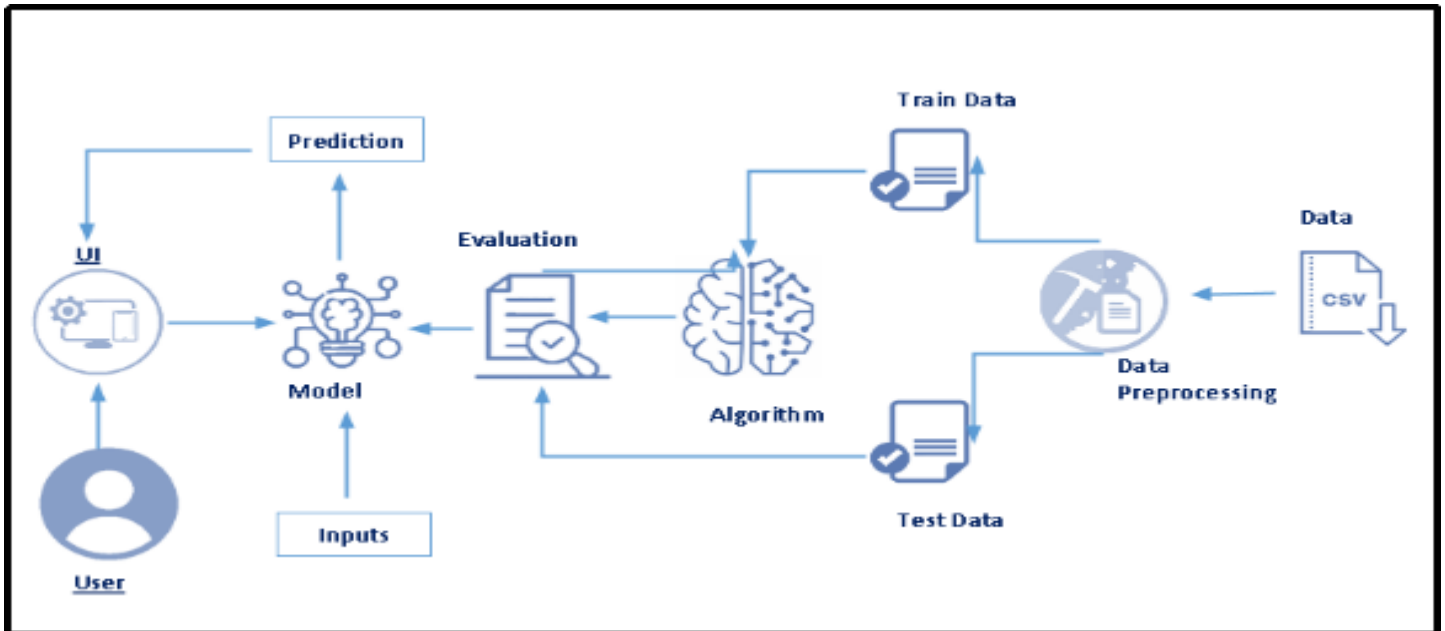
3.2 Hardware / Software designing

In order to develop this project we need to install the following software/packages.

Anaconda Navigator:

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.

To make a responsive python script you must require the following packages

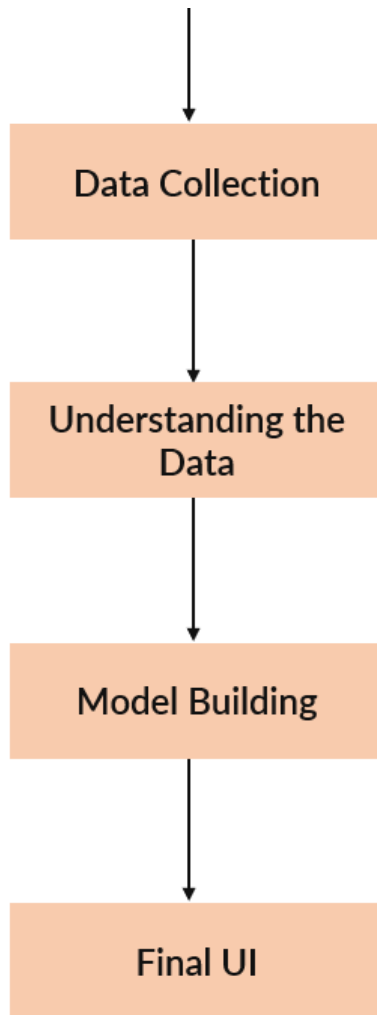**Requests**: This allows you to send HTTP requests using Python.

Flask: Web framework used for building Web applications

4.EXPERIMENTAL INVESTIGATIONS

1. Knowledge on Machine Learning Algorithms.

2. Knowledge on Python Language with Machine Learning

3. Knowledge on Statistics and Graphs and their relations

4. Knowledge on Natural Language Processing (NLP).

5. Real Time Analysis of Project

6. Building an ease of User Interface (UI)

7. Navigation of ideas towards other projects(creativeness)

8. Knowledge on building ML Model.

9. You will be able to know how to find the accuracy of the model.

10. How to Build web applications using the Flask framework
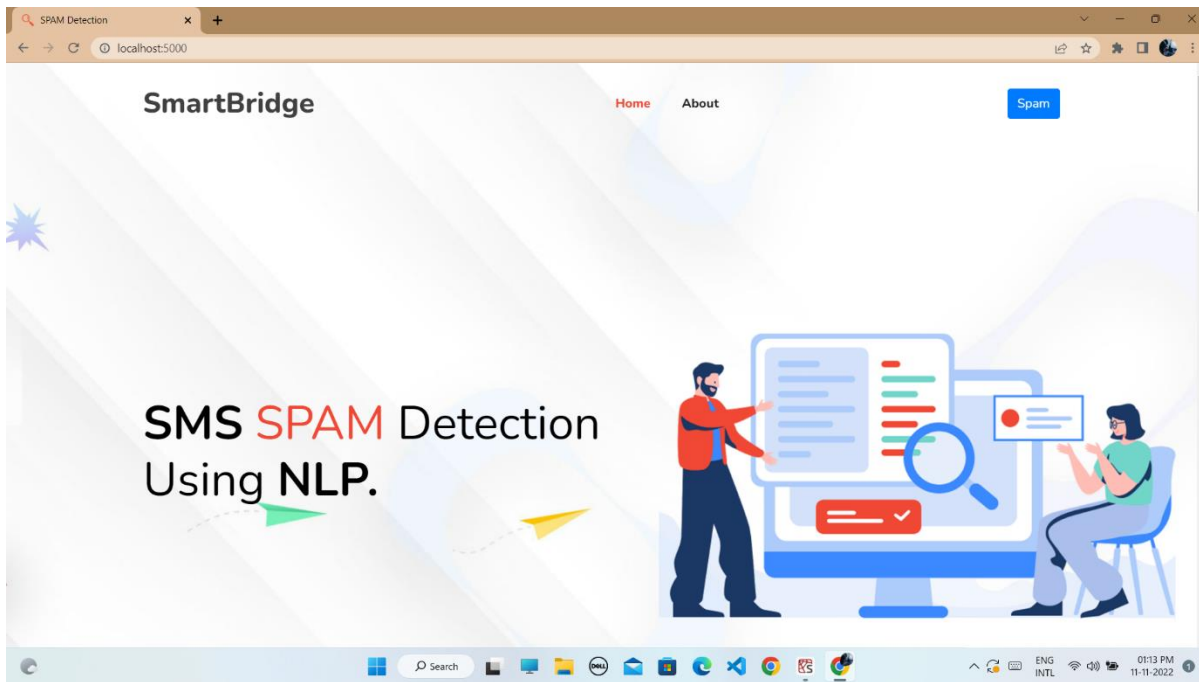
5.FLOWCHART

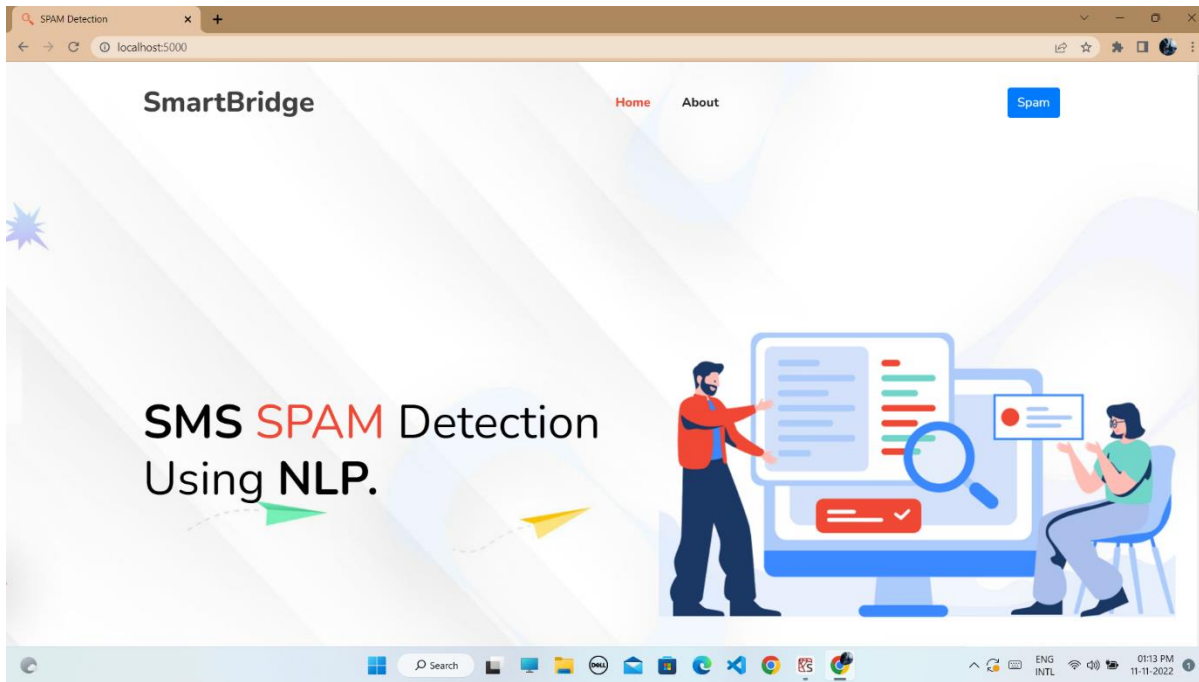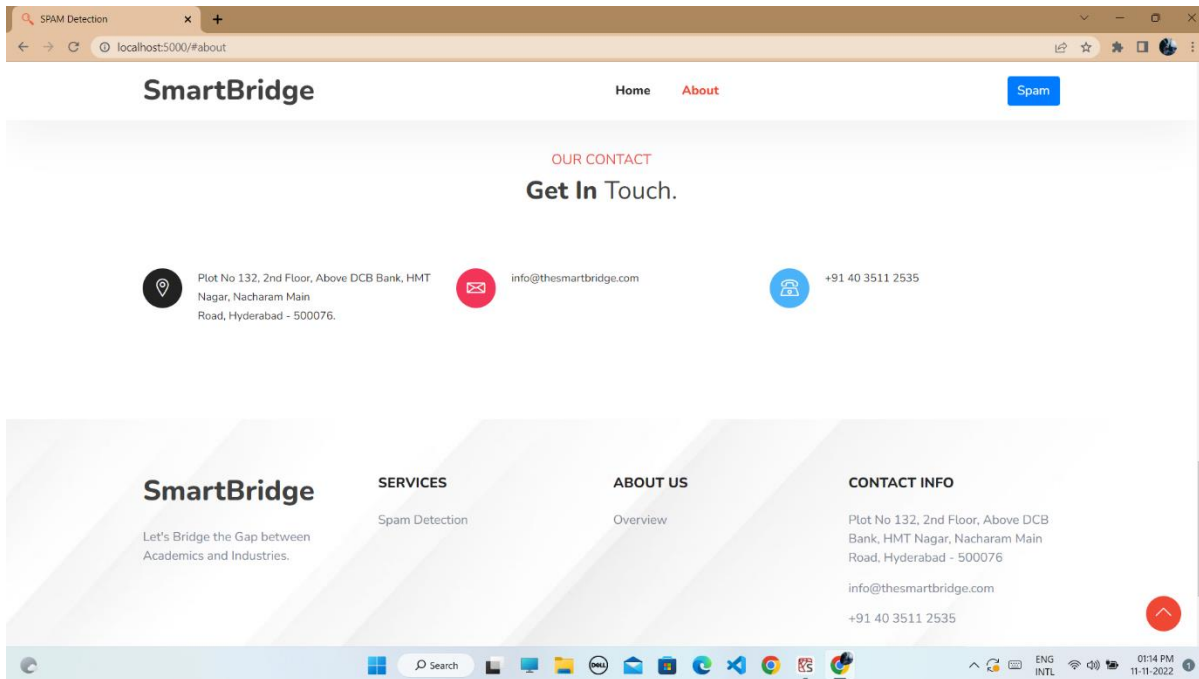Diagram showing the control flow of the solution

```
        │
        ▼
┌─────────────────┐
│ Data Collection │
└─────────────────┘
        │
        ▼
┌─────────────────┐
│ Understanding the│
│      Data        │
└─────────────────┘
        │
        ▼
┌─────────────────┐
│ Model Building  │
└─────────────────┘
        │
        ▼
┌─────────────────┐
│    Final UI     │
└─────────────────┘
```
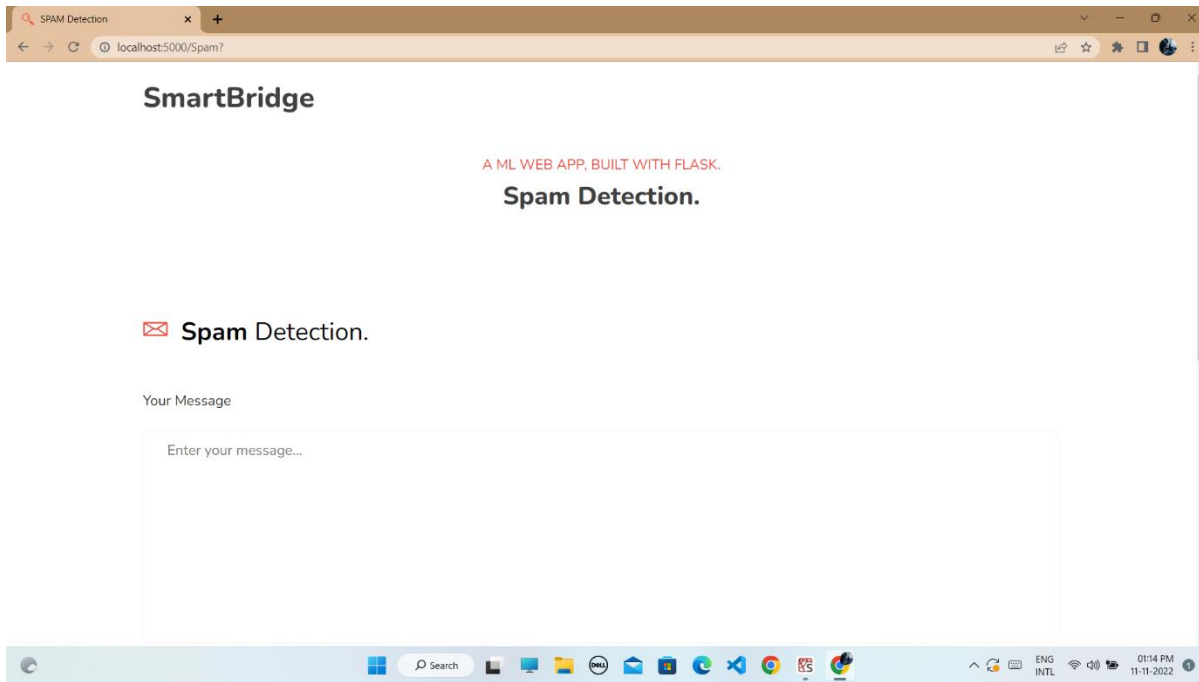
6.RESULT

OUTPUT SCREENSHOTS

HOME PAGE:

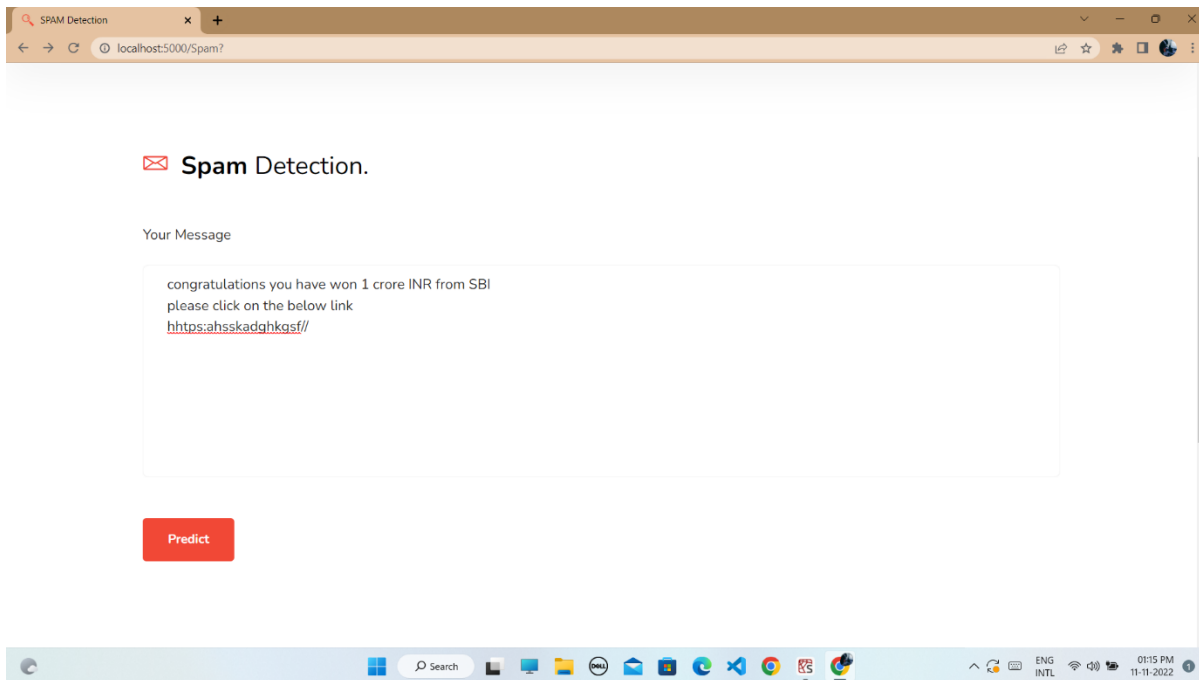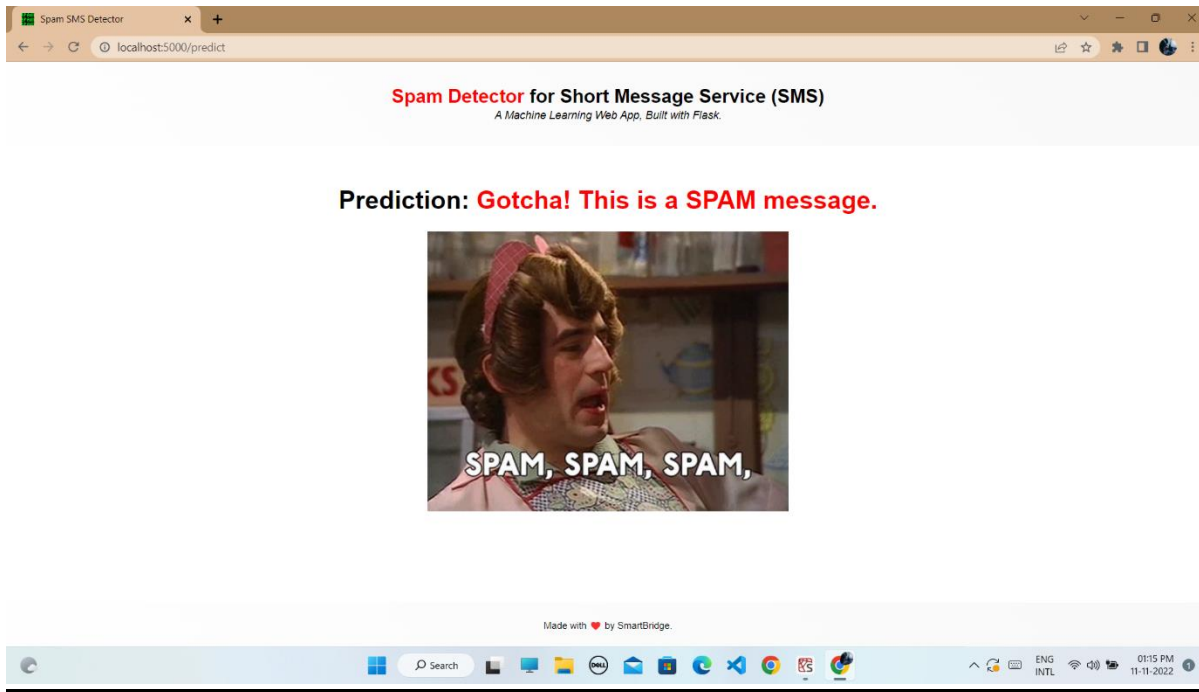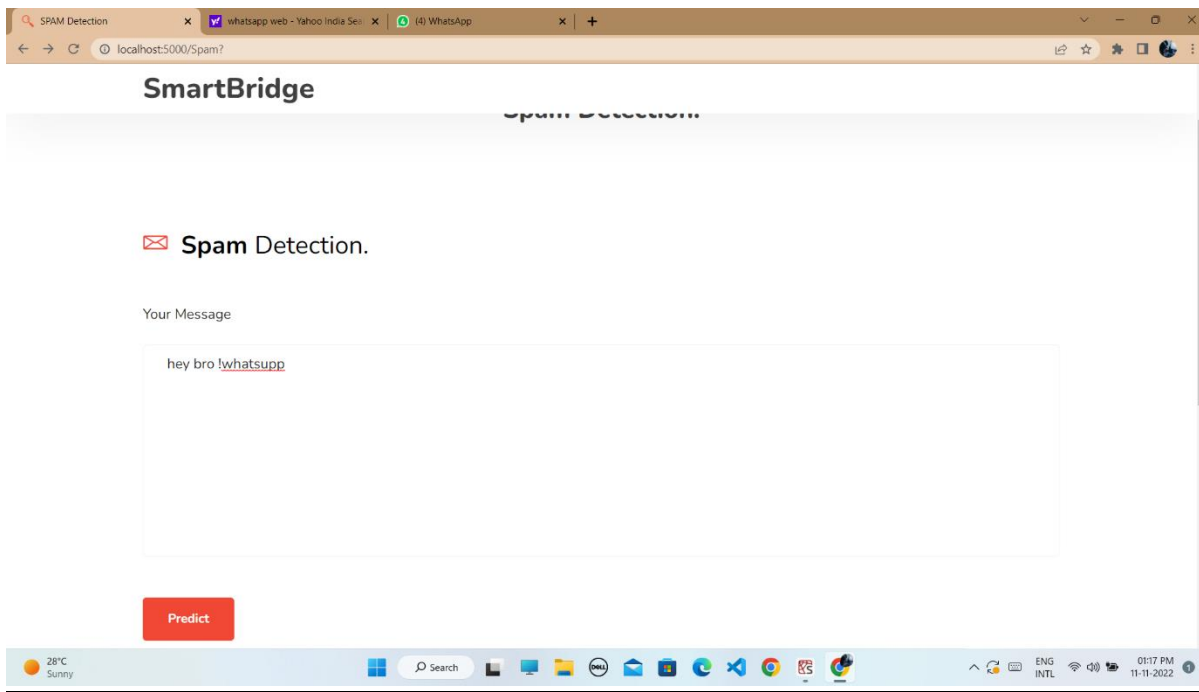# ABOUT PAGE:

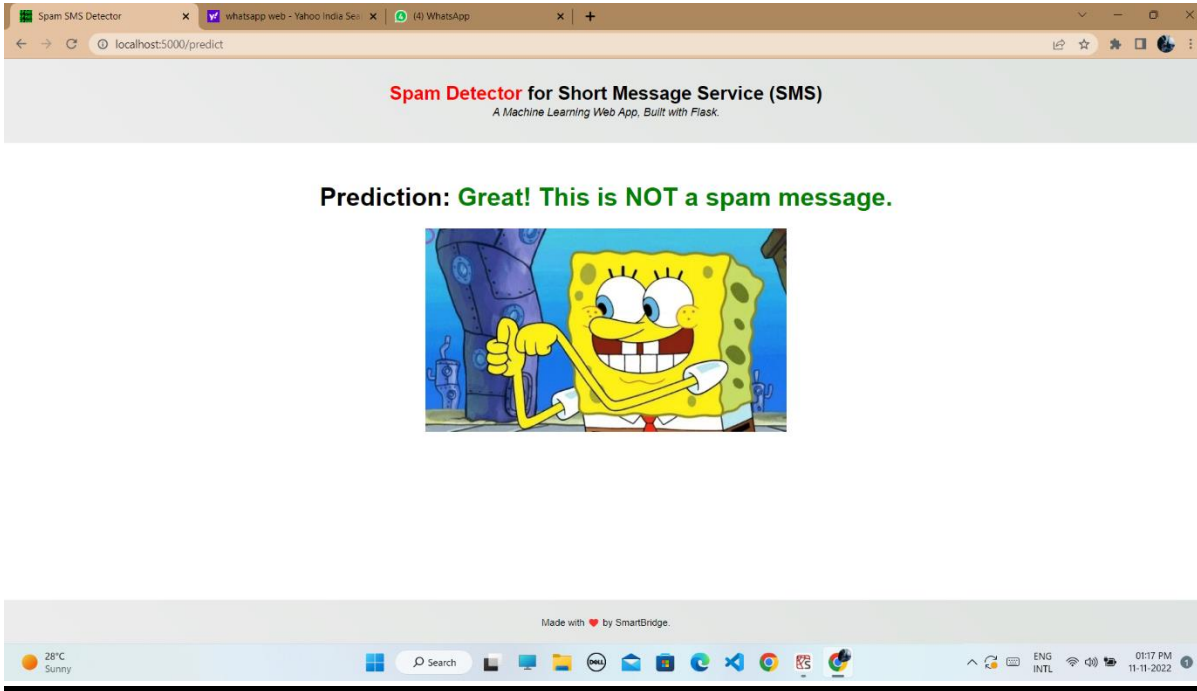# SMS SPAM DETECTION PAGE:

ENTERING A SPAM SMS TO CHECK FOR SOLUTION:

RESULT:

ENTERING A NORMAL SMS TO CHECK FOR SOLUTION:

RESULT:

7.ADVANTAGES & DISADVANTAGES

ADVANTAGES	DISADVANTAGES
Detect whether a SMS is spam or not spam	Inaccurate data may give wrong prediction
Easy to use and implement	Data privacy issues
It works on Real time and has low complexity	Results may be inaccurate for other languages

8.APPLICATIONS

An integrated framework in which Spam probability is just one component will enhance the model's usefulness by reducing the number of false positives (from only blocking after a certain amount of

warnings), not evaluating messages from known legitimate users and promptly block repeated spammers from using the app

9.<u>CONCLUSION</u>

Machine learning algorithms can play a vital role in finding the Spam SMS with natural language processing techniques.

10.<u>FUTURE SCOPE</u>

The proposed system of the project will effectively detect the spam SMS and the system will extract the spam SMS by using some machine learning algorithms and it gives the result with greater accuracy and with good performance. Also, this proposed system will optimize the data storage by blocking and deleting the spam SMS. And with the help of the Opinion Rank model, this proposed system will find trustworthiness of the SMS and it will carry the filtering of spam messages. This proposed system will save the user's time and it destroys the risk of spam SMS.

11.<u>BIBILOGRAPHY</u>

Model Building
 1. Dataset
 2. Jupiter Notebook
Application Building
 1. HTML file
 2. CSS file
 3. Flask
4. IBM Watson

12.<u>APPENDIX</u>

SOURCE CODE
```
# Importing essential libraries
from flask import Flask, render_template, request
import pickle

# Load the Multinomial Naive Bayes model and CountVectorizer object from disk
filename = 'spam-sms-mnb-model.pkl'
classifier = pickle.load(open(filename, 'rb'))
cv = pickle.load(open('cv-transform.pkl','rb'))
```

```python
app = Flask(__name__)

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

@app.route('/Spam',methods=['POST','GET'])
def prediction(): # route which will take you to the prediction page
    return render_template('spam.html')

@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        message = request.form['message']
        data = [message]
        vect = cv.transform(data).toarray()
        my_prediction = classifier.predict(vect)
        return render_template('result.html', prediction=my_prediction)

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