## EDITH:THE FAKE NEWS DETECTOR

**A MINI PROJECT REPORT**



|  |  |
| --- | --- |
| **Submitted** | **by** |
| **JITHEESWARAN.V** | **220701108** |
| **JEROME SATHIYAGU.A** | **220701106** |
| **JAYA PRAKASH.V** | **220701103** |

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS) THANDALAM

CHENNAI-602105

2024 - 25

**BONAFIDE CERTIFICATE**

Certified that this project report “**EDITH:THE FAKE NEWS DETECTOR**” is the bonafide work of **“JITHEESWARAN V (220701108) ,JEROME SANTHIYAGU.A(220701106),JAYA PRAKASH.V(220701103) ”**

who carried out the project work under my supervision.

**Submitted for the Practical Examination held on**

**SIGNATURE SIGNATURE**

**Dr.R.SABITHA Ms. DHARANI DEVI M.Tech.,PhD.,**

**Professor and II Year Academic Head Assistant Professor (SG), Computer Science and Engineering, Computer Science and Engineering,**

**Rajalakshmi Engineering College Rajalakshmi Engineering College, (Autonomous), (Autonomous),**

**Thandalam, Chennai - 602 105 Thandalam, Chennai - 602 105**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

#### ABSTRACT

Our database-driven fake news detection system utilizes historical news data to identify potentially misleading articles. By querying a database of known fake news, the system compares new articles, determining their credibility. Through headline matching and content analysis, it provides real-time feedback on article authenticity. This approach empowers users to make informed decisions about the news they consume, complementing NLP and ML techniques.

Efficient and scalable, our system promotes media literacy, critical thinking, and safeguards information integrity in the digital age, offering a promising solution to combat misinformation.

#### ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR.S.MEGANATHAN** and the chairperson **DR.M.THANGAM MEGANATHAN** for their timely support and encouragement. We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance. No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr.P.KUMAR M.E Ph.D.,** for being ever supporting force during our project work We also extend our sincere and hearty thanks to our internal guide Dr G **DHARANI DEVI M.Tech.,PhD.,** for her valuable guidance and motivation during the completion of this project. Our sincere thanks to our **STARK INDUSTRIES** team and other staff members of computer science engineering who worked with us in this mini project.

1. JITHEESWARAN.V
2. JEROME SATHIYAGU.A

### JAYAPRAKESH.V

#### TABLE OF CONTENTS

1. INTRODUCTION
   1. INTRODUCTION
   2. OBJECTIVES
   3. MODULES
2. SURVEY OF TECHNOLOGIES
   1. SOFTWARE DESCRIPTION
   2. LANGUAGES
      1. POSTGRE SQL
      2. HTML,CSS,JS
      3. PYTHON
3. REQUIREMENTS AND ANALYSIS
   1. REQUIREMENT SPECIFICATION
   2. HARDWARE AND SOFTWARE REQUIREMENTS
   3. ARCHITECTURE DIAGRAM
   4. ER DIAGRAM
   5. NORMALIZATION
4. PROGRAM CODE
5. RESULTS AND DISCUSSION
6. CONCLUSION
7. REFERENCE

## CHAPTER 1 INTRODUCTION

#### INTRODUCTION

In today's digital age, the proliferation of fake news presents significant challenges to information integrity and public trust. To combat this issue, fake news detection systems are developed to identify and flag misleading or false content across various platforms. These systems rely on sophisticated algorithms, natural language processing (NLP), and machine learning techniques to analyze and classify news articles. A critical component of these systems is the underlying database structure, which efficiently stores and manages the data involved in the detection process.

In a fake news detection system, the database typically includes entities such as articles and fact- check records. Articles contain essential information like the title, content, publication date, author, and source. Fact-check records document the verification process, including the results of fact-checking, the individual or organization conducting the check, and the date it was performed. By organizing this data effectively in a relational database such as PostgreSQL, the system can support robust analysis and retrieval processes necessary for accurate fake news detection.

#### OBJECTIVE

The primary objective of the fake news detector is to accurately identify and flag misinformation and disinformation within news articles. This system aims to enhance the integrity and reliability of information available to the public by leveraging advanced data analysis, natural language processing (NLP), and machine learning techniques.

#### MODULE

* + - ADD NEWS
    - NEWS CHECK FAKE OR REAL
    - REPOTING METHOD
    - SAVING THE FAKE NEWS IF IT IS NOT THERE IN REAL DATABASE

# Ch 2. SURVEY OF TECHNOLOGY

#### SOFTWARE DESCRIPTION

**Visual studio Code**

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

* 1. **LANGUAGES**

#### HTML

Hypertext: text (often with embeds such as images, too) that is organized in order to connect related items

Markup: a style guide for typesetting anything to be printed in hardcopy or soft copy format

Language: a language that a computer system understands and uses to interpret commands.

HTML determines the structure of web pages. This structure alone is not enough to make a web page look good and interactive. So you'll use assisted technologies such as CSS and JavaScript to make your HTML beautiful and add interactivity, respectively.

#### CSS

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

#### PYTHON

Python is pivotal in the fake news detector, offering simplicity and extensive libraries for data processing, natural language processing (NLP), and machine learning. Its robust frameworks, such as NLTK and scikit-learn, facilitate efficient algorithm development, making Python ideal for building and deploying effective fake news detection systems.

#### POSTGRESQL

"Postgre" typically refers to PostgreSQL, which is a powerful, open-source relational database management system (RDBMS). PostgreSQL is known for its robustness, extensibility, and standards compliance. It supports advanced data types and performance optimization features, making it ideal for handling large datasets and complex queries. In the context of a fake news detector, PostgreSQL is used to store and manage data related to news articles and fact- checking, ensuring efficient data retrieval and integrity for accurate and timely analysis

# Ch 3. REQUIREMENT AND ANALYSIS

#### REQUIREMENTS SPECIFICATION

**User Requirements**

The system requirement in library management focuses on the possibility of search for books by title, author or subject by the member.

#### System Requirements

There should be a database backup of the library management system. Operating system should be WindowsXP or a higher version of windows.

#### HARDWARE AND SOFTWARE REQUIREMENTS Software Requirements

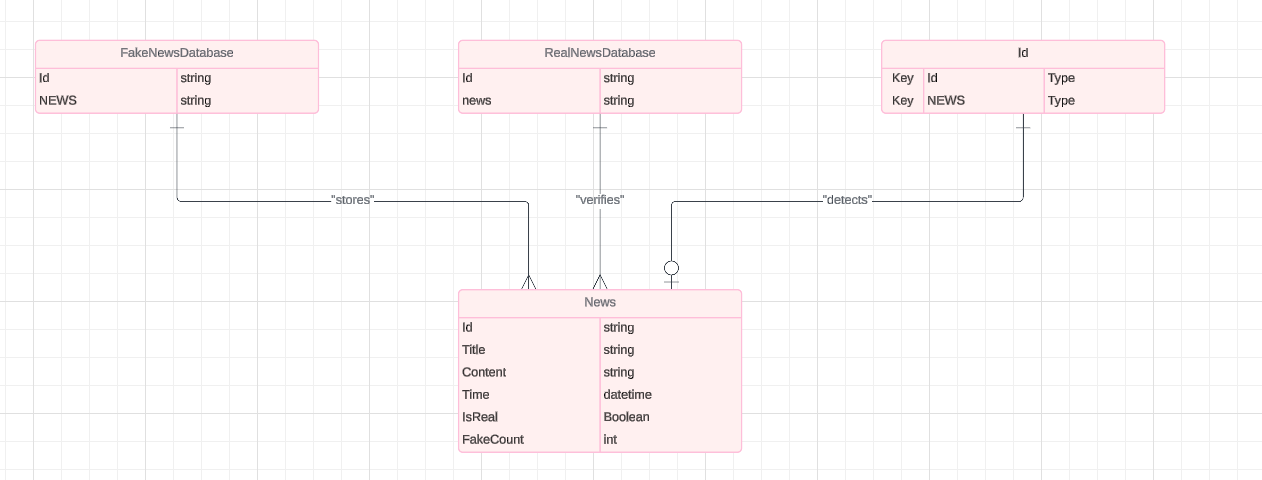
* Operating System Windows 10
* Front End HTML , CSS , javascript
* Back End PYTHON, Postgre SQL

#### Hardware Requirements

* Desktop PC or a Laptop
* Printer
* Operating System – Windows 10
* Intel® CoreTM i3-6006U CPU @ 2.00GHz ● 4.00 GB RAM
* 64-bit operating system, x64 based processor ● 1024 x 768 monitor resolution
* Keyboard and Mouse

#### ARCHITECTURE DIAGRAM

* 1. **ER-DIAGRAM**

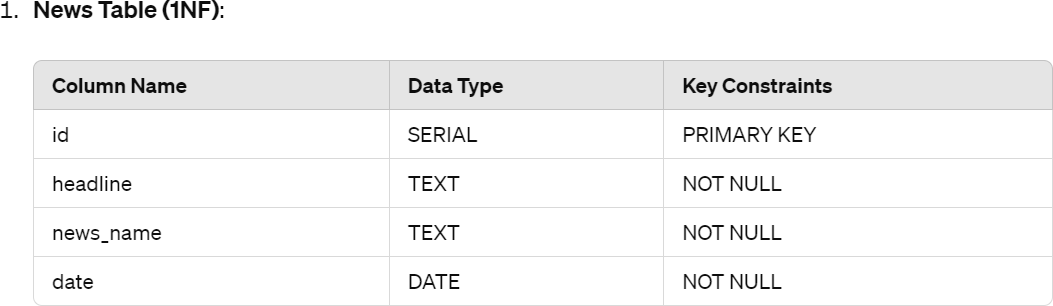


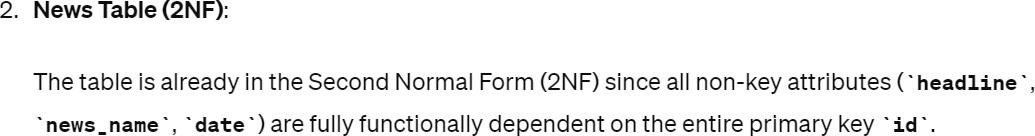
**3.4 NORMALIZATION**

Raw database

Raw database

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Key constraints |
| id | number | Primary Key NOT NULL |
| News headlines | varchar(50) | NOT NULL |
| News name | varchar(50) | NOT NULL |
| date | DATE | NOT NULL |





# Ch 4. PROGRAM CODE

### NEWS FEEDER

* + 1. HTML/CSS AND JS

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>News Submission</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body {

font-family: Arial, sans-serif; margin: 0;

padding: 0;

background-image: url("https://static1.srcdn.com/wordpress/wp-content/uploads/2019/05/Tony-Stark-from-Iron- Man.jpg");

background-size: cover; background-repeat: no-repeat;

}

h1 {

text-align: center;

}

form {

width: 50%;

margin: 0 auto; background-color: #fff; padding: 20px;

border-radius: 5px;

box-shadow: 0px 0px 10px 0px rgba(0,0,0,0.1);

}

input[type=text] { width: 100%; padding: 10px; margin: 5px 0 15px 0;

border: 1px solid #ccc; border-radius: 3px;

box-sizing: border-box;

}

input[type=submit] { width: 100%;

background-color: #4CAF50; color: white;

padding: 10px 0; margin-top: 10px; border: none;

border-radius: 3px; cursor: pointer;

}

input[type=submit]:hover { background-color: #45a049;

}

</style>

</head>

<body>

<div class="container mt-4">

<h1 class="mb-4">News feeder</h1>

{% with messages = get\_flashed\_messages(with\_categories=true) %}

{% if messages %}

{% for category, message in messages %}

<div class="alert alert-{{ category }} alert-dismissible fade show" role="alert">

{{ message }}

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

{% endfor %}

{% endif %}

{% endwith %}

<form action="{{ url\_for('submit\_news') }}" method="POST">

<div class="form-group">

<label for="headline">Headline:</label>

<input type="text" class="form-control" id="headline" name="headline">

</div>

<div class="form-group">

<label for="news\_name">News Name:</label>

<input type="text" class="form-control" id="news\_name" name="news\_name">

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="[https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js](https://cdn.jsdelivr.net/npm/%40popperjs/core%402.5.4/dist/umd/popper.min.js)"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

### PYTHON

from flask import Flask, render\_template, request, redirect from datetime import datetime

import psycopg2 import os

app = Flask( name )

# PostgreSQL connection details from environment variables db\_name = os.getenv('DB\_NAME', 'postgres')

db\_user = os.getenv('DB\_USER', 'postgres')

db\_password = os.getenv('DB\_PASSWORD', 'Vj@07150209') db\_host = os.getenv('DB\_HOST', 'localhost')

db\_port = os.getenv('DB\_PORT', '5433') # Specify the custom port here

def get\_db\_connection(): conn = psycopg2.connect(

dbname=db\_name, user=db\_user, password=db\_password, host=db\_host, port=db\_port

)

return conn

@app.route('/') def index():

return render\_template('index.html')

@app.route('/submit\_news', methods=['POST']) def submit\_news():

headline = request.form['headline'] news\_name = request.form['news\_name'] date = datetime.now().date()

if not headline or not news\_name:

return "Both headline and news name are required!", 400

try:

conn = get\_db\_connection() cursor = conn.cursor() cursor.execute(

"INSERT INTO news (headline, news\_name, date) VALUES (%s, %s, %s)", (headline, news\_name, date)

)

conn.commit() cursor.close() conn.close()

return "News submitted successfully!" except Exception as e:

return f'An error occurred: {e}', 500

if name == ' main ': app.run(debug=True)

### EDITH FAKE NEWS DETECTOR

* + 1. html,css and js

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Fake News Detection System</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {

background: rgba(0, 128, 0, 0.3);

background-image: url("https://static1.srcdn.com/wordpress/wp-content/uploads/2019/05/Tony-Stark-from-Iron- Man.jpg");

margin: 0;

padding: 0;

font-family: 'Open Sans Condensed', sans-serif;

}

.login {

text-align: center; padding: 5em;

}

input[type=text], select, textarea { width: 100%;

padding: 10px;

border: 3px solid #ccc; border-radius: 1px;

box-sizing: border-box; margin-top: 6px; margin-bottom: 16px; resize: vertical;

}

button {

background-color: #4CAF50; color: white;

padding: 14px 20px; margin: 8px 0; border: none; cursor: pointer; width: 100%;

}

button:hover { opacity: 0.8;

}

h1 {

text-align: center;

font-family: 'Pacifico', cursive;

}

p {

text-align: center;

}

.results {

margin-top: 20px; color: aliceblue;

}

.results h2 {

font-family: 'Arimo', sans-serif; font-size: 24px;

color: aliceblue;

}

.error {

color: red;

font-weight: bold;

}

.report {

margin-top: 20px; display: none;

}

.report h2 {

font-size: 24px; font-weight: bold;

margin-bottom: 10px;

}

.report label { font-size: 18px; display: block;

margin-bottom: 5px;

}

.report textarea { height: 100px; resize: vertical;

}

#reportMessage { margin-top: 10px; font-weight: bold;

}

</style>

</head>

<body>

<div class="login">

<h1>Fake News Detector</h1>

<p>EDITH("Even dead I'm the hero") Predict</p>

<form id="predictionForm" action="/predict" method="post">

<!-- Existing headline input -->

<input type="text" id="headlineInput" name="title" required="required" style="font-size: 18pt">

<br>

<button type="button" onclick="predictFakeNews()">Predict</button>

<div class="results">

<h2 id="predictionResult"></h2>

</div>

</form>

<!-- Report section -->

<div class="report" id="reportForm" style="display: none;">

<h2>Report Fake News</h2>

<form id="reportFormInner">

<label for="website">Website:</label>

<input type="text" id="websiteInput" name="website" required>

<br>

<label for="fake\_news">DESCRIPTION OF THE NEWS:</label>

<textarea id="fakeNewsInput" name="fake\_news" required></textarea>

<br>

<button type="button" onclick="submitReportForm()">Report</button>

</form>

</div>

</div>

<script>

function predictFakeNews() {

const headlineInput = document.getElementById("headlineInput").value;

// Send the user input to the server for prediction fetch('/predict', {

method: 'POST',

body: JSON.stringify({ title: headlineInput }), headers: { 'Content-Type': 'application/json' }

})

.then(response => response.json())

.then(data => {

const predictionResult = document.getElementById("predictionResult");

predictionResult.textContent = `Looking ${data.prediction === "FAKE" ? "Spam ı." : "Real"} News ] ⬛`;

// Show or hide the report form based on the prediction

const reportForm = document.getElementById("reportForm"); if (data.show\_report\_form) {

reportForm.style.display = "block";

} else {

reportForm.style.display = "none";

}

})

.catch(error => {

console.error('Error predicting news:', error);

const predictionResult = document.getElementById("predictionResult"); predictionResult.textContent = 'Error predicting news';

});

}

function submitReportForm() {

const websiteInput = document.getElementById("websiteInput").value; const fakeNewsInput = document.getElementById("fakeNewsInput").value;

// Send the report data to the server fetch('/report', {

method: 'POST',

body: JSON.stringify({ website: websiteInput, fake\_news: fakeNewsInput }), headers: { 'Content-Type': 'application/json' }

})

.then(response => response.json())

.then(data => {

const reportMessage = document.getElementById("reportMessage"); reportMessage.textContent = data.message;

})

.catch(error => {

console.error('Error submitting report:', error);

const reportMessage = document.getElementById("reportMessage"); reportMessage.textContent = 'Error submitting report';

});

}

</script>

</body>

</html>

### PYTHON

from flask import Flask, render\_template, request, jsonify import psycopg2

app = Flask( name ) def connect\_to\_database():

# Connect to PostgreSQL database

conn = psycopg2.connect( dbname="postgres", user="postgres", password="Vj@07150209", host="localhost", port="5433"

)

return conn

@app.route('/') def index():

return render\_template('ig.html')

@app.route('/predict', methods=['POST']) def predict():

user\_headline = request.json.get('title', '')

if not user\_headline:

return jsonify({'error': 'No headline provided'}) conn = connect\_to\_database()

# Query headlines from the database cursor = conn.cursor()

cursor.execute("SELECT headline FROM news") headlines = cursor.fetchall()

cursor.close()

# Check if the user's headline matches any headline from the database if user\_headline in [headline[0] for headline in headlines]:

prediction = "REAL" else:

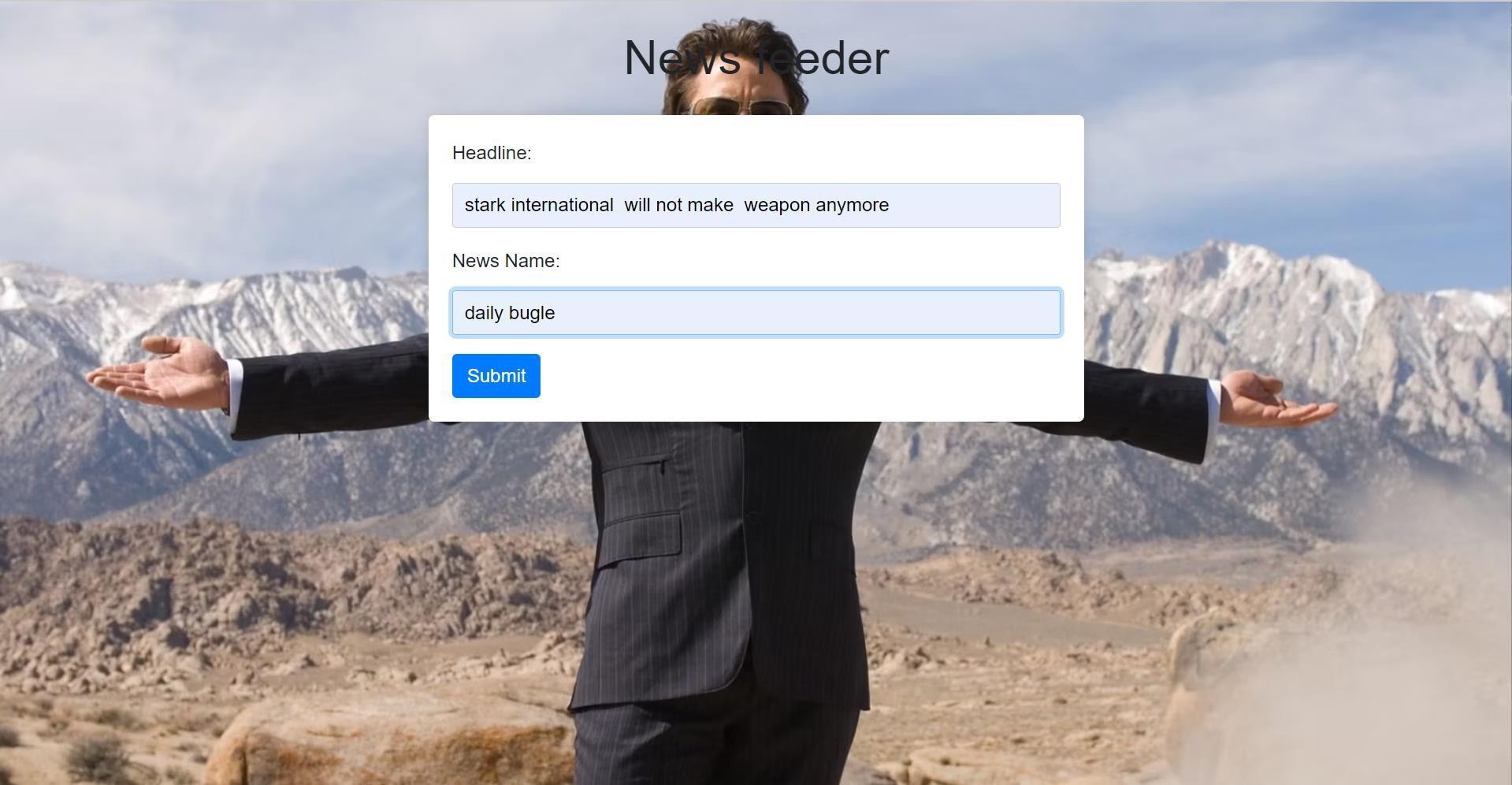
prediction = "FAKE" conn.close()

return jsonify({'prediction': prediction})

if name == ' main ': app.run(debug=True)

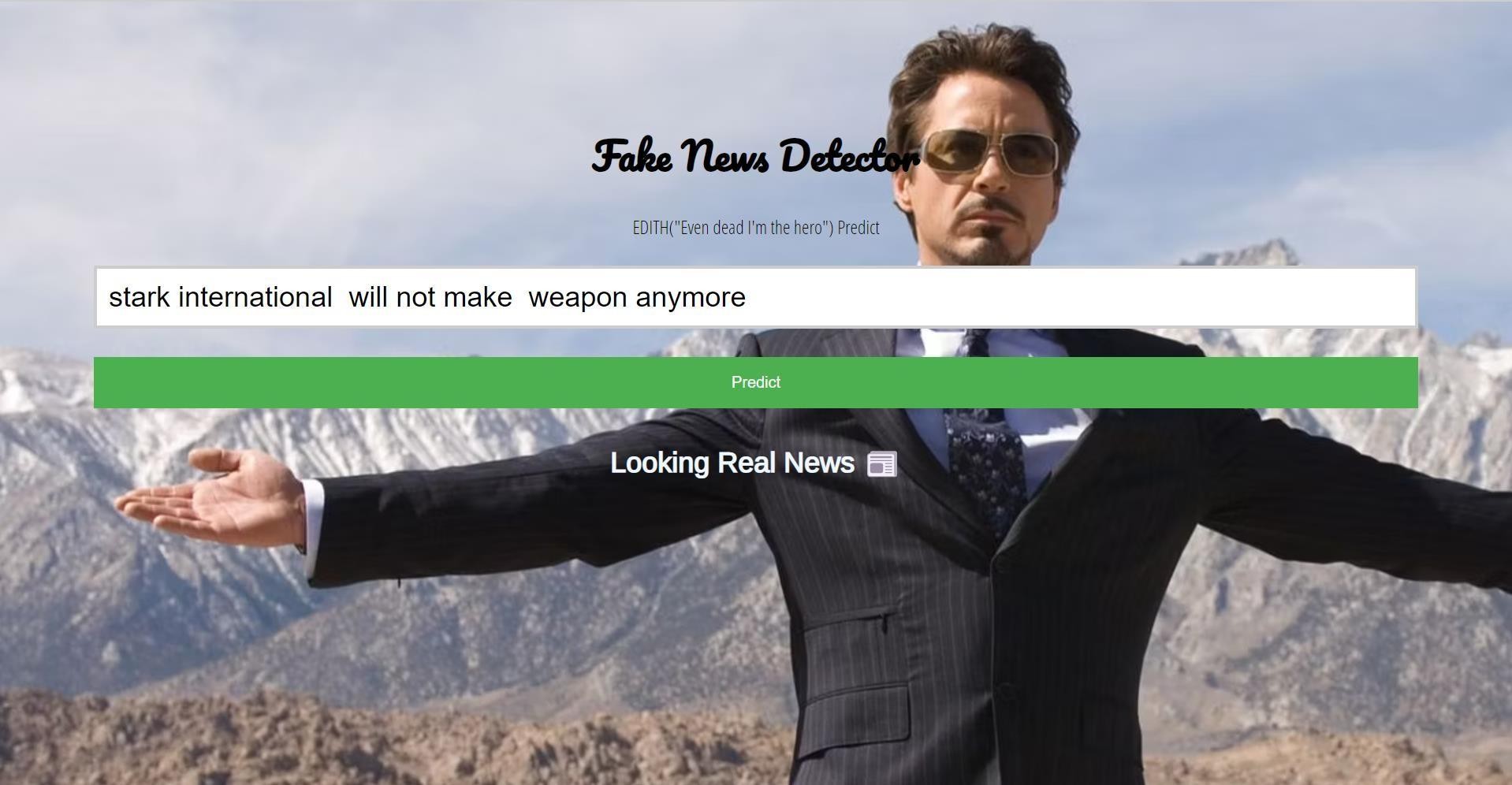
# Ch 5. RESULTS AND DISCUSSION

* 1. **USER DOCUMENTATION NEWS FEEDER**



# News feeder database

**Edith fake news predictor Real news**



# Fake news

* 1. **DISCUSSION**

1. Database Reliability: Discuss the reliability of using a database lookup approach for fake news detection. Highlight its strengths in quickly identifying known fake news articles based on historical data stored in the database.
2. Scalability: Explore how the system would scale as the size of the database grows. Consider the potential challenges and solutions for handling large volumes of news articles efficiently.
3. Real-time Detection vs. Historical Data: Compare the advantages and disadvantages of real-time detection methods, such as natural language processing (NLP) and machine learning (ML), with the database lookup approach. Discuss scenarios where each method might be more suitable.
4. Accuracy and False Positives: Discuss the potential for false positives and false negatives in the database lookup approach. Consider how to minimize these errors and improve the accuracy of the system.
5. Updating the Database: Explore strategies for updating the database with new information and removing outdated or incorrect data. Discuss the importance of maintaining a reliable and up-to-date database for effective fake news detection.
6. Integration with Other Systems: Discuss how the database lookup approach could be integrated with other systems or technologies to enhance fake news detection. Consider the potential for combining database lookup with NLP, ML, or user feedback mechanisms for improved accuracy.
7. Ethical Considerations: Consider the ethical implications of using historical data to classify news articles as real or fake. Discuss the importance of transparency, fairness, and accountability in designing and deploying fake news detection systems.
8. User Experience: Discuss the user experience of interacting with the fake news detection system. Consider how to communicate the system's predictions effectively to users and provide actionable information to help them make informed decisions.
9. Educational Outreach: Explore the role of educational outreach in combating fake news. Discuss how the fake news detection system could be used as a tool for educating users about the importance of critical thinking and media literacy.
10. Future Directions: Consider future directions for improving fake news detection technology. Discuss potential research areas, technological advancements, and collaborative efforts to address the evolving challenges of misinformation and disinformation online.

# Ch 6. CONCLUSION

In conclusion, the database-driven fake news detection system offers a promising solution to the pervasive issue of misinformation. By leveraging historical data and advanced database technology, the system efficiently identifies potentially misleading news articles in real-time. Its ability to complement existing techniques such as NLP and ML enhances its effectiveness in promoting media literacy and critical thinking. Moreover, the system's scalability and user-centric approach empower individuals to make informed decisions about the information they consume, thereby safeguarding the integrity of online discourse. As we continue to innovate in the fight against fake news, this system stands as a valuable tool in preserving the truth and fostering a more informed society.

# Ch 7. REFERENCE

* 1. Alakesh Bora aka @alokesh985. *Passive Aggressive Classifiers*. GeeksForGeeks, 17-July-2020.
  2. Giuseppe Bonaccorso aka @giuseppebonaccorso. *ML Algorithms addendum: Passive Aggressive Algo- rithms*. [www.bonaccorso.eu,](http://www.bonaccorso.eu/) 10 June 2017.
  3. Koby Crammer & al. *Online Passive-Aggressive Algo- rithms*. Journal of Machine Learning Research, March 2006.