A

Real Time Research Project

On

"FAKE NEWS DETECTION SYSTEM"

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

In

Computer Science and information technology

By

T.HASADHIKA - 23R21A3358
PRASAD GOUD - 23R21A3341
V.HAMSINI - 23R21A3364
N.ADITHYA - 23R21A3346

Under the guidance of

Mrs N.Thulasi Chitra
Assistant
Professor

Department of Computer Science & Information technology

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS
MLR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Affiliated to JNTUH,Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India
NBA NATIONAL BOARD ACCREDITATION
A NAAC

MAY 2025

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS
MLR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Affiliated to JNTUH,Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India
NBA NATIONAL BOARD OF ACCREDITATION
A NAAC

Department of Computer Science and Information Technology

CERTIFICATE

This is to certify that the project entitled "**FAKE NEWS DETECTION SYSTEM**" has been submitted by **T. Hasadhika (23R21A3358), M. Prasad Goud (23R21A3341), V. Hamsini (23R21A3364), N.Adithya (23R21A3346)** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Information Technology from MLR Institute of Technology, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Internal Guide **Head of the Department**

External Examinar

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS
MLR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Affiliated to JNTUH,Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India
NBA NATIONAL BOARD ACCREDITATION
A NAAC

Department of Computer Science and Information Technology

DECLARATION

We hereby declare that the project entitled "**FAKE NEWS DETECTION SYSTEM**" is the work done during the period from **February 2025 to May 2025** and is submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Information technology from MLR Institute of Technology, Hyderabad.

T.HASADHIKA - 23R21A3358
M.PRASAD GOUD - 23R21A3341
V.HAMSINI - 23R21A3364
N.ADITHYA - 23R21A3346

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS
MLR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Affiliated to JNTUH,Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India
NBA NATIONAL BOARD OF ACCREDITATION
A NAAC

Department of Computer Science and Information Technology

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I now have the opportunity to express my guidance for all of them.

First of all We would like to express our deep gratitude towards our internal guide **N.Thulasi Chitra**, Assistant Professor, Department of CSIT for her support in the completion of our dissertation. We wish to express our sincere thanks to **Dr. P. Subhashini**, HOD, Dept. of CSIT and also principal **Dr. K. SRINIVAS RAO** for providing the facilities to complete the dissertation.

We would like to thank all my faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

T.HASADHIKA - 23R21A3358
M.PRASAD GOUD - 23R21A3341
V.HAMSINI - 23R21A3364
N.ADITHYA - 23R21A3346

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS
MLR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Affiliated to JNTUH,Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India
NBA NATIONAL BOARD OF ACCREDITATION
A NAAC

Department of Computer Science and Information Technology

ABSTRACT

Misinformation has become a growing threat in today's information-driven world, especially due to the rapid and uncontrolled spread of fake news across social media and online platforms. This project presents the development of a web-based fake news detection system that utilizes machine learning to classify news articles as either true or fake. The core of the system is a Multinomial Naive Bayes classifier trained on a labeled dataset containing real and fake news articles, preprocessed using a TF-IDF vectorizer to extract meaningful patterns from text data. The application is built using Python's Flask web framework and offers an interactive, category-driven user interface where users can select a news type and submit content for verification. The system outputs real-time predictions and also logs user inputs and results for performance tracking. Additionally, it includes modular components for storage and reporting, enabling future scalability and enhancements such as trend visualization and admin-level analytics. This solution demonstrates how machine learning and natural language processing can be effectively integrated into a responsive web application to combat the spread of misinformation and promote responsible digital behaviour.

INDEX

LIST OF FIGURES

FIGURES PAGE NO.

1.INTRODUCTION

In today's digitally interconnected world, the rapid dissemination of information through online platforms has become both a strength and a vulnerability of modern communication. While digital media has democratized news distribution and made information widely accessible, it has also led to a surge in the spread of fake news—fabricated stories or misleading content presented as factual reporting. This phenomenon has had far-reaching consequences, influencing public opinion, fueling misinformation during crises, undermining democratic institutions, and damaging societal trust. Traditional methods of verifying news, such as manual fact-checking or relying on third-party validation, are slow, labor-intensive, and incapable of keeping up with the speed and volume of content generated and shared online. To address this growing challenge, the proposed system introduces a smart, web-based application for fake news detection that leverages machine learning and natural language processing (NLP). Built using Python and the Flask web framework, the application provides users with a simple interface where they can input any news article and instantly receive a classification result indicating whether the content is likely to be true or fake. At the core of the system is a Multinomial Naive Bayes classifier trained on a well-labeled dataset of real and fake news, utilizing a TF-IDF vectorizer to extract relevant textual features. The application is designed to be user-centric, featuring category-wise news selection, secure login, and dynamic styling that adapts to the type of content entered. It also incorporates modular components for storage and reporting, allowing prediction logs to be stored and summarized for analytical purposes. The system's lightweight design ensures smooth performance across devices, while its reliance on open-source technologies guarantees affordability and scalability. By providing an automated, real-time tool for verifying news credibility, this platform not only counters the spread of misinformation but also empowers users to develop critical thinking and digital literacy skills in an era where truth in media is increasingly difficult to discern..

1.1 Existing System

In the current landscape, the identification of fake news heavily relies on manual and semi-automated verification processes. Users are often left to verify the credibility of a news article using independent online fact-checking websites, browser plug-ins, or by cross-referencing with known media houses. This system is inefficient and prone to human error, especially as fake news becomes

more sophisticated in mimicking real journalism. While some online platforms offer basic tools to flag or report suspicious content, their effectiveness is limited, particularly in the early stages of news propagation. Additionally, most of the available tools are reactive rather than preventive; they act after the misinformation has already begun circulating. Although some artificial intelligence and machine learning-based tools exist, they are often part of premium services, lack accessibility for common users, or require complicated setups like browser extensions or integration with APIs. Many such models are also trained using social signals (such as likes, shares, or user interactions), which are not always available at the onset of news distribution. Furthermore, these tools are not adaptable to multilingual contexts or domain-specific news types, limiting their utility in diverse media environments. Hence, the need for a proactive, easy-to-use, and real-time fake news detection system remains largely unmet.

Disadvantages of Existing system

- Requires manual effort and user awareness to search and verify news content.
- Lacks support for early-stage detection as many rely on social media metrics.
- Many tools are browser-specific, limiting accessibility across all devices.
- Some services are paid or restricted, reducing their reach among common users.
- Absence of personalized experience or category-wise prediction results.
- Limited adaptability to multiple languages or non-English content.
- Inconsistent accuracy due to dependency on third-party databases and tools.

1.2 Proposed System

The proposed system introduces a user-friendly and intelligent web application that detects fake news using machine learning and natural language processing. The core objective is to provide a real-time platform where users can paste a news article and get an immediate verdict on whether the article is likely true or fake. Developed using Python and the Flask framework, the system is lightweight, fast, and easily deployable. It incorporates a Multinomial Naive Bayes classifier trained on a labeled dataset of real and fake news articles. To prepare textual data for the model, a TF-IDF vectorizer is employed to extract frequency-based features. The application is divided into three main modules: Prediction, Storage, and Reporting. Upon submission, the news text is processed, classified, and the

result is displayed with appropriate contextual visuals. At the same time, the prediction and user input are stored securely in local logs or a backend database. An additional reporting module is integrated to allow administrators or developers to generate usage statistics, accuracy tracking, and logs for evaluation. The UI is designed using HTML, CSS, and Jinja templating, with news-type-based themes (e.g., Politics, Sports, Health) for better contextual interaction. The system supports secure login functionality and is scalable for future upgrades like multilingual support, multimedia verification, or chatbot integration

.Advantages of Existing System

- Accessible platforms such as fact-checking websites help users verify content manually.
- Some tools offer real-time prediction features using social media propagation data.
- Several browser extensions allow users to assess news credibility while browsing.
- A few AI-driven platforms exist that attempt to flag suspicious or fake content.
- Centralized databases of fake news claims support comparative verification.
- Public awareness around fake news has increased due to such existing solutions.
- Encourages critical thinking by promoting manual cross-checking before sharing.

Disadvantages of Existing system

- Requires manual effort and user awareness to search and verify news content.
- Lacks support for early-stage detection as many rely on social media metrics.
- Many tools are browser-specific, limiting accessibility across all devices.
- Some services are paid or restricted, reducing their reach among common users.
- Absence of personalized experience or category-wise prediction results.
- Limited adaptability to multiple languages or non-English content.
- Inconsistent accuracy due to dependency on third-party databases and tools.

Advantages of Proposed System.

- The proposed system offers real-time fake news classification using machine learning.
- Provides a styled and intuitive user interface based on news categories.
- Employs supervised learning with Naive Bayes and TF-IDF for accurate predictions.
- Includes a secure login system with personalized access and session tracking.
- Modular architecture separates prediction, storage, and reporting components.

- Fully built using open-source technologies like Python and Flask.
- Promotes digital literacy by making news verification accessible and educational.

1.3System Requirements:

1.3.1 Software Requirements:

- **Front-end Development:**HTML, CSS, Bootstrap, and Jinja2 templates for styling and rendering.
- **Back-end Development:**Python, Flask web framework for routing, server control, and backend logic.
- **Libraries:** scikit-learn (ML modeling), pandas (data handling), NumPy (mathematical operations), joblib (model persistence).

1.3.2 Hardware Requirements:

- **Processor:** Dual-core Intel i3 or equivalent (minimum clock speed 2.4 GHz).
- **RAM:** Minimum 8 GB to support model loading, web hosting, and real-time response generation.
- **Storage:**At least 20 GB of available space for datasets, logs, model files, and local deployment.
- **Network Connectivity:**A stable internet connection (10 Mbps or higher) recommended for local hosting and testing.

2. LITERATURE SURVEY

S.No.	TITLE	ADVANTAGES	DISADVANTAGES
1	Multi-model Framework for Early Fake News Detection	It improves detection accuracy by combining both textual and early propagation data through the multimodal architecture, enhancing the ability to detect fake news in its early stages.	It requires high computational resources and complex integration of multiple models like BERT and GNN, making it less suitable for real-time or lightweight applications. .
2	Fake News Detection Using Sentiment Analysis Approach in Indonesian Language	It improves classification by using a heterogeneous attention network that captures global node relationships, addressing the limitations of local attention in traditional GNNs.	It depends heavily on well- structured graph representations, and performance may drop on short or poorly connected news articles with sparse sentiment data.
3	Regularized LSTM Method for Detecting Fake News Articles	It enhances the detection performance with high accuracy by using the LSTM with regularization and optimization techniques, making it robust for large and diverse datasets.	It suffers from slow training time and may require significant hardware resources due to the sequential nature of LSTM networks and multiple model iterations.
4	An Improved Ensemble-base Online Fake News Detection System	It improves performance using an ensemble of classifiers with weighted majority voting, achieving higher accuracy, precision, and recall compared to individual models.	It increases model complexity and maintenance effort, as combining multiple classifiers and managing online updates like with the NTA agent requires more system overhead.

5	An Analysis of Various Classification Algorithms for Fake News Detection	It improves understanding by evaluating multiple models under a unified framework and shows LSTM as a highly accurate solution, useful for selecting the best approach.	It does not provide an integrated deployment strategy and mainly focuses on comparative results, lacking a real-time implementation or web-based system demonstration.

3.SYSTEM DESIGN

3.1 PROPOSED SYSTEM ARCHITECTURE

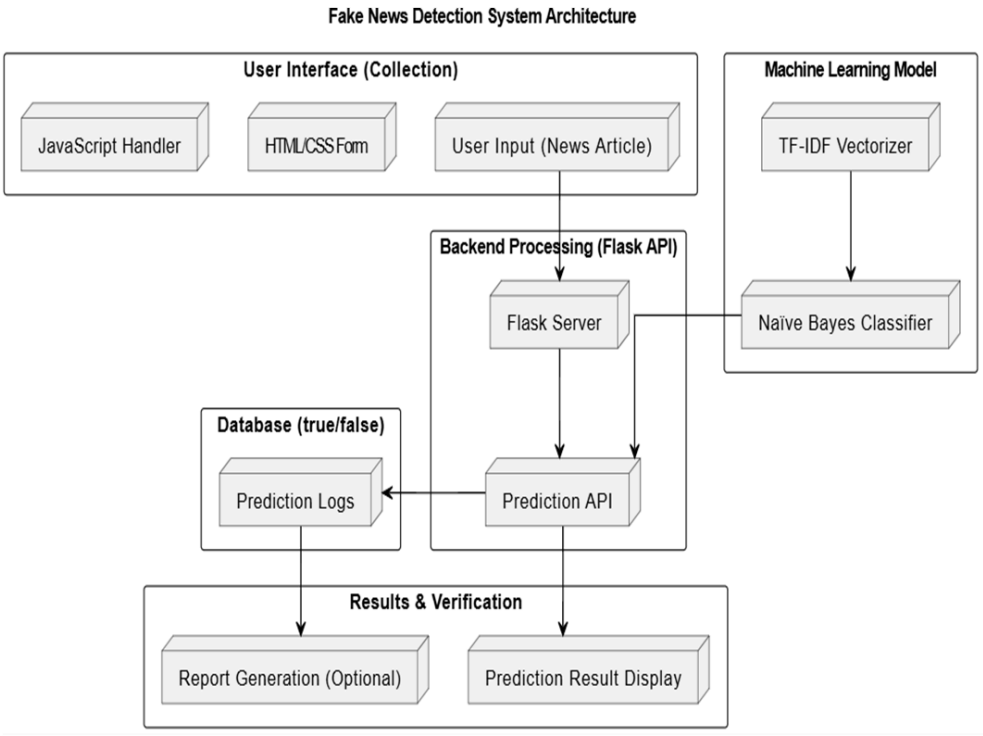

Fig 3.1.1 – Proposed System Architecture

The Fake News Detection Website operates through a modular client-server architecture, with a user-facing interface built using HTML, CSS, and Flask templates, and a backend built in Python using the Flask micro-framework. The system provides functionalities for news submission, category-based UI, user authentication, machine learning-based classification, and prediction logging. The classification model—Multinomial Naive Bayes—is trained on a dataset of real and

fake news articles, using TF-IDF for feature extraction. Upon submission, the input text is processed and passed to the model, which returns a binary prediction. The interface dynamically adjusts based on the selected category (e.g., Politics, Sports, Health) to enhance user context and experience. Logs of predictions are stored in a local database or CSV file, and an admin can access report summaries. The system is lightweight, scalable, and suited for real-time deployment on local or cloud servers.

3.2 UML DIAGRAM

The Fake News Detection Website is divided into two major modules: Admin Login and User Login, each with defined responsibilities and interaction flows to optimize prediction and report generation.

In the User Login module, users begin by registering or logging into the system. Once authenticated, they can select a news category, enter the article content, and submit it for prediction. The system processes the text input, classifies it using the trained model, and displays the result—either "True" or "Fake." Users also have access to their prediction history for review and learning.

In the Admin Login module, the administrator logs in to access logs of all user predictions, check system statistics (like accuracy or usage frequency), and manage datasets for retraining the model. This division ensures a clear role-based interface and improves data traceability and system maintainability.

3.3 Flow Chart

- User logs into the system using secure credentials.
- User selects a news type from the list (Politics, Sports, etc.).
- System dynamically loads category-specific UI.
- User pastes or writes the article text into the input box.
- System processes the text using TF-IDF and applies the Naive Bayes model.
- Prediction is generated and shown to the user in real time.
- User can view result and access past predictions.
- Logs are stored for admin-level reporting.
- Admin views summarized report statistics via dashboard.

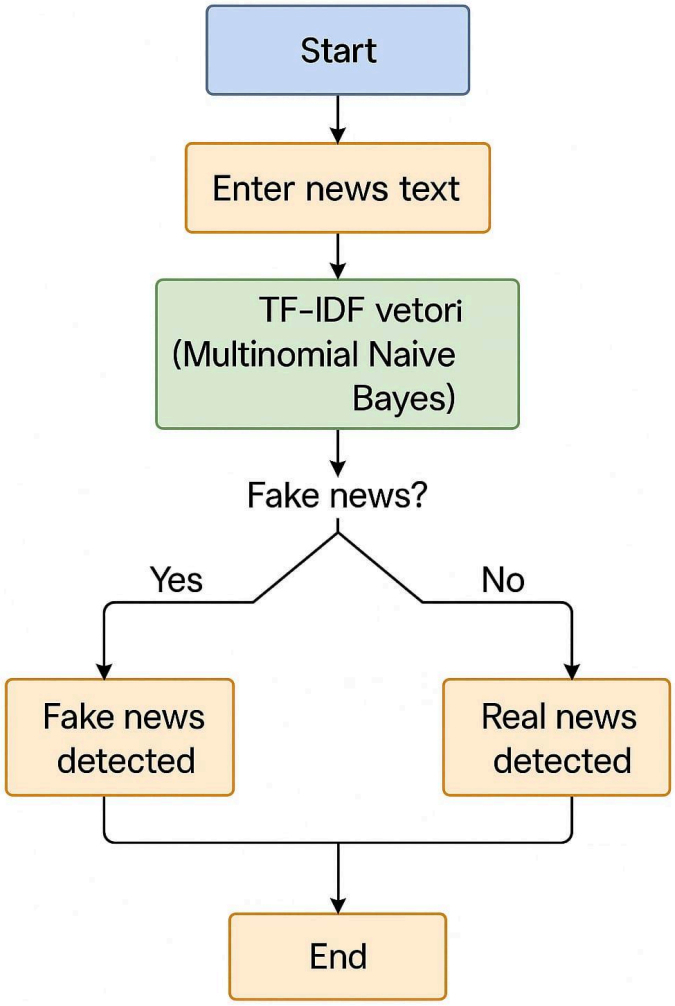

Fig 3.3.1 – Flow Chart

3.4 Sequence diagram

User Authentication: The user logs into the platform and the system authenticates the credentials.

News Submission: The user selects a category and submits the news article for review.

Model Invocation: The Flask backend processes the text using TF-IDF and calls the trained Naive Bayes model.

Prediction Generation: The model returns the prediction and the Flask server routes it back to the user interface.

Result Display: The result is displayed dynamically, and the input/prediction is stored.

Admin Reporting: The admin accesses prediction history and analytical summaries for system

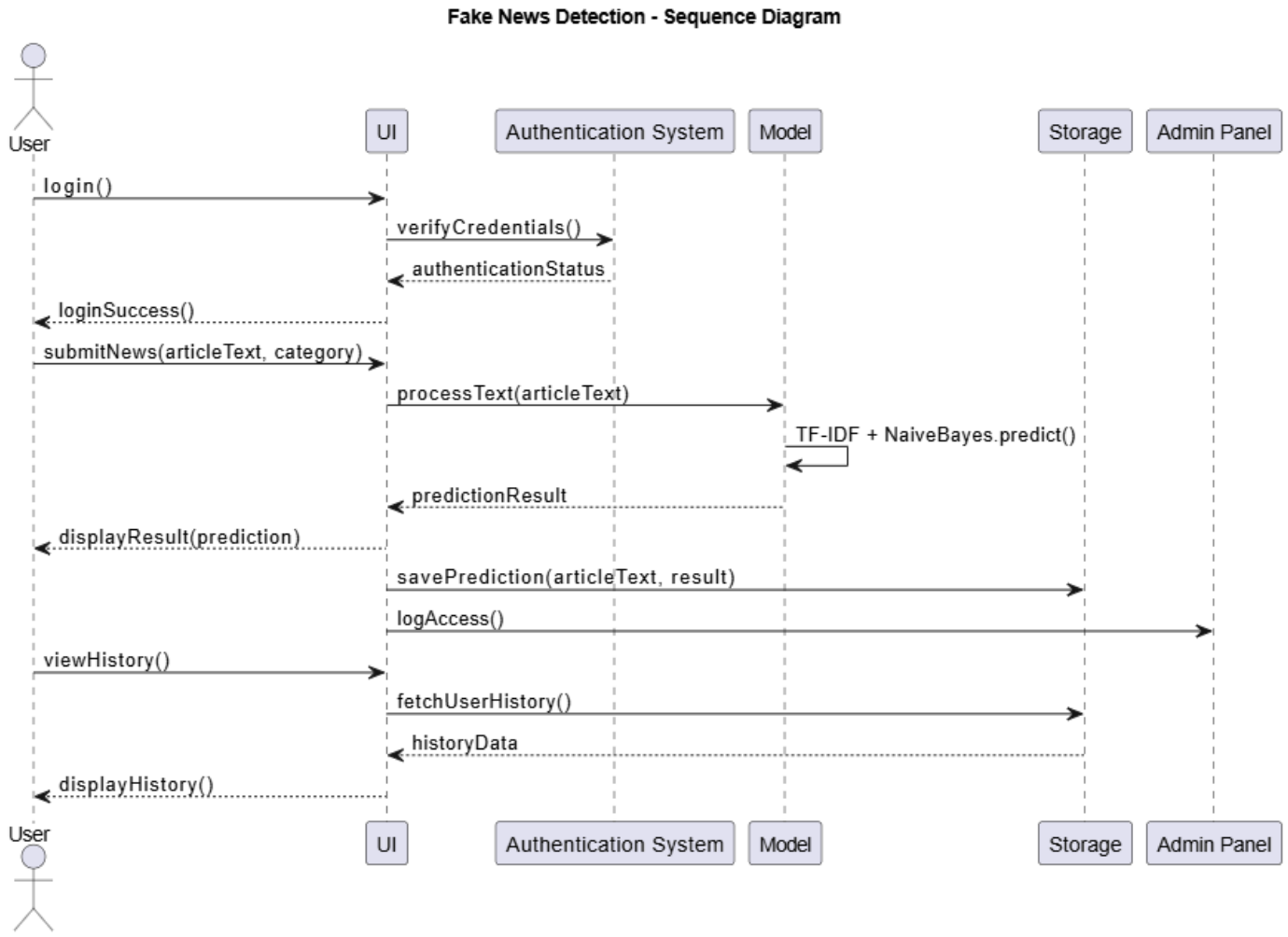

Fig 3.3.2 – Sequence diagram

3.5 Usecase Diagram

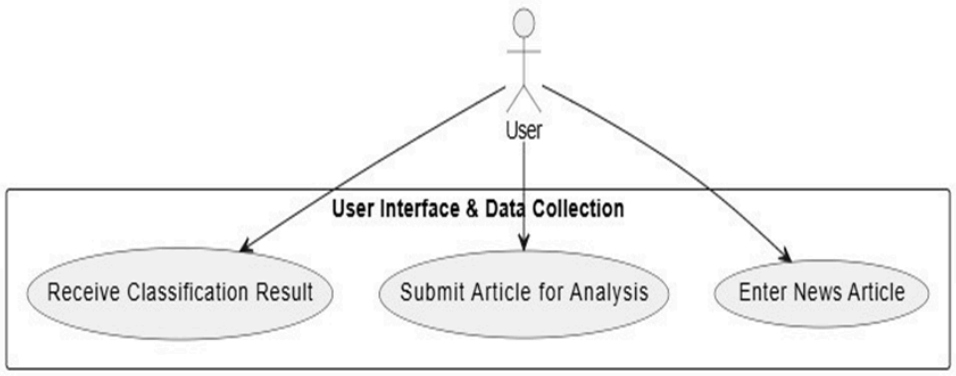

Fig 3.5.1 – Usecase Diagram

3.6 Class Diagram

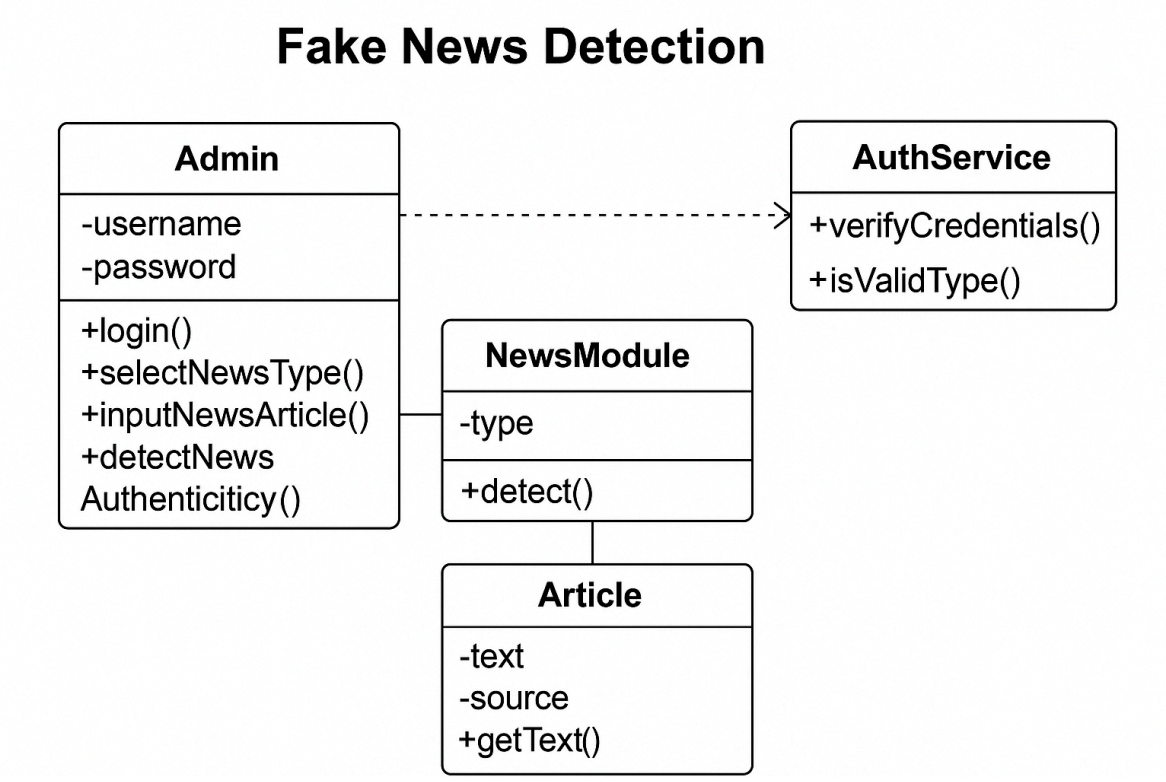

Fig 3.6.1 – Class Diagram

4.IMPLEMENTATION

Source Code

Code which we used to run the project

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB

from sklearn.pipeline import make_pipeline

from sklearn.metrics import accuracy_score, classification_report
true_df = pd.read_csv('True.csv')

true_df['label'] = 0 # Add a label column for true

# Load fake news dataset
fake_df = pd.read_csv('Fake.csv')
fake_df['label'] = 1 # Add a label column for fake
news # Combine the datasets
df = pd.concat([true_df, fake_df],
ignore_index=True) # Split the data into training
and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['label'], test_size=0.2, random_state=42)
model = make_pipeline(TfidfVectorizer(), MultinomialNB())
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)


# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
classification_report_result = classification_report(y_test,
predictions) # Display the result
```

```
print(f'Accuracy: {accuracy}')
print('Classification Report:\n', classification_report_result)
user_input = input("Enter a news article: ")

# Make a prediction
prediction = model.predict([user_input])
# Display the result if prediction[0] == 0:
    print("The news is likely to be true.") else:
    print("The news is likely to be fake.") 6.Testing
```

App.py

```
from flask import Flask, render_template_string, request, redirect, url_for, session import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline


app = Flask(_name_)
app.secret_key = 'supersecret' # Needed for session


# Hardcoded login credentials USERNAME = 'admin' PASSWORD = 'password'

# Load and train model
true_df = pd.read_csv('True.csv')
true_df['label'] = 0
fake_df = pd.read_csv('Fake.csv')
fake_df['label'] = 1
```

```
df = pd.concat([true_df, fake_df], ignore_index=True)

X_train, X_test, y_train, y_test = train_test_split(df['text'], df['label'],
    test_size=0.2, random_state=42)
model = make_pipeline(TfidfVectorizer(), MultinomialNB())
model.fit(X_train, y_train)
# HTML


Templates


login_page =


'''<!DOCTYPE


html>


<html


lang="en">


<head>
  <meta charset="UTF-8">
  <title>Login - Fake News Detector</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
  <style>
    body {background-color: #f0f2f5;}
    .login-container {max-width: 400px; margin: 100px auto; background: white; padding: 30px;
border-radius: 10px; box-shadow: 0px 0px 10px 0px #0000001a;}
    .logo {display: block; margin: 0 auto 20px; width: 80px;}
  </style>
</head>
<body>
  <div class="login-container">
    <img src="https://cdn-icons-png.flaticon.com/512/21/21601.png" alt="News
Logo" class="logo">
    <h3 class="text-center mb-4">Login</h3>
    <form method="POST" action="/login">
      <div class="form-group mb-3">
        <label>Username</label>
```

```
            <input type="text" class="form-control" name="username"
placeholder="Enter username" required>
        </div>
        <div class="form-group mb-3">
            <label>Password</label>

            <input type="password" class="form-control" name="password" placeholder="Enter password"
required>
        </div>
        <button type="submit" class="btn btn-primary w-100">Login</button>
    </form>
</d
iv>
</
bo
dy
>
</
ht
ml
>
'''

home_page = '''

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Select News Type</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        body {background-color: #f4f4f9;}
        .container {margin-top: 100px;}
        .selection {max-width: 400px; margin: auto;}
    </style>
</head>
<body>


<div class="container text-center">

    <h1 class="mb-5">Fake News Detection</h1>
```

```
    <div class="selection">

        <h4>Select the type of news</h4>
        <form action="/news-type" method="POST">
            <select name="news_type" class="form-select mt-3" required>
                <option value="Politics">Politics</option>
                <option value="Sports">Sports</option>
                <option value="Technology">Technology</option>
                <option value="Entertainment">Entertainment</option>
                <option value="Health">Health</option>
            </select>
            <button type="submit" class="btn btn-primary w-100 mt-4">Proceed</button>
        </form>
    </div>

</div>


</body>

</
ht
ml
>
'''


news_type_page = '''

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{{ news_type }} - Fake News Detection</title>
    <link                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        body {
            background-image:                                                    url('https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcT2iehWJ137UW9FD6ycN9wN377QC2naNT1-
3Q&s{{ news_type.lower() }}');
            background-size:    cover;
            background-position:
            center;
            background-repeat:
```

```
        no-repeat; height: 100vh;
        color: white;
    }
    .overlay {
        background: rgba(0, 0, 0, 0.6);
        height:
        100%;
        display:
        flex;
        justify-content:
        center;
        align-items:
        center;
        flex-direction:
        column;   padding:
        20px;

    }

        .form-card      {
            background:
            white;
            padding:
            30px;
            border-radiu
            s:          15px;
            width:
            100%;
        max-width:
        600px;  color:
        black;


        box-shadow: 0 4px 15px rgba(0,0,0,0.4);
    }
        textare
            a   {
            resi
            ze:
            non
            e;

    }

</style>
```

```
</head>
<body>


<div class="overlay">

   <div class="form-card">
      <h2 class="mb-4 text-center">Detect Fake News in {{ news_type }}</h2>
      <form action="/predict" method="POST">
         <div class="mb-3">
            <textarea name="news_article" class="form-control" placeholder="Enter news article here..." rows="6" required></textarea>
         </div>
         <input type="hidden" name="news_type" value="{{ news_type }}">
         <button type="submit" class="btn btn-success w-100">Check News</button>
      </form>


      {% if prediction_result %}

      <div class="alert alert-info mt-4 text-center">
         <h5>Prediction Result:</h5>
         <p>{{ prediction_result }}</p>
      </div>
      {% endif %}
   </div>
</div>


</body>
</
ht
ml
>
'''


# Flask Routes


@app.route(
'/')          def
main_login(
):
   return
```

```
render_template_string(login_page)

@app.route('/login', methods=['POST'])



def login():
    username = request.form['username']
    password = request.form['password']
        if username == USERNAME and password == PASSWORD:
            session['user'] = username
        return
    redirect('/home')
    else:

        return render_template_string(login_page, error="Invalid Credentials!")

@app.route('/ho
me') def home():
        if 'user' not in
            session: return
            redirect('/')

    return render_template_string(home_page)

@app.route('/news-type',
methods=['POST']) def news_type():
        if 'user' not in
            session: return
            redirect('/')
    news_type = request.form['news_type']
    return render_template_string(news_type_page, news_type=news_type)


@app.route('/predict', methods=['POST'])
def predict():
        if 'user' not in
            session: return
            redirect('/')
    user_input =
    request.form['news_article']
    news_type =
    request.form['news_type'] prediction =
    model.predict([user_input]) if
```

```
    prediction[0] == 0:
        result = " ✅ The news is likely to be TRUE."
    else:
        result = " ❌ The news is likely to be FAKE."
    return render_template_string(news_type_page, news_type=news_type, prediction_result=result)

if _name_____== '_main_': app.run(debug=True)
```

5.RESULTS

Fake News Detection System illustrates the interaction between the user and the system. The primary actor, the user, begins by logging into the platform, after which they can select the type of news (e.g., Politics, Health, Sports) and enter the text of a news article they wish to verify. The system, which incorporates a trained machine learning model using TF-IDF and Naive Bayes, processes the input and performs a fake news detection operation. Once the analysis is complete, the system returns a prediction result, indicating whether the article is likely to be true or fake. Optional use cases include feedback collection or logging user results for future analysis.

OUTPUTS

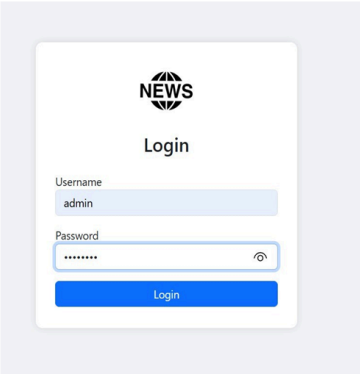

Fig 4.1 – login interface

Fig 4.2 – News selection

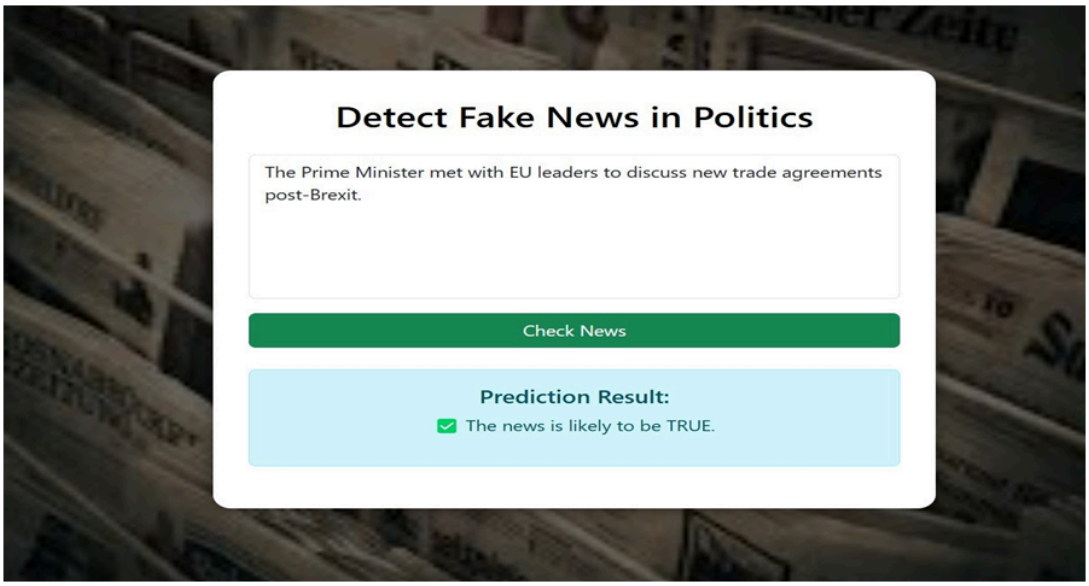

Fig 4.3–results prediction

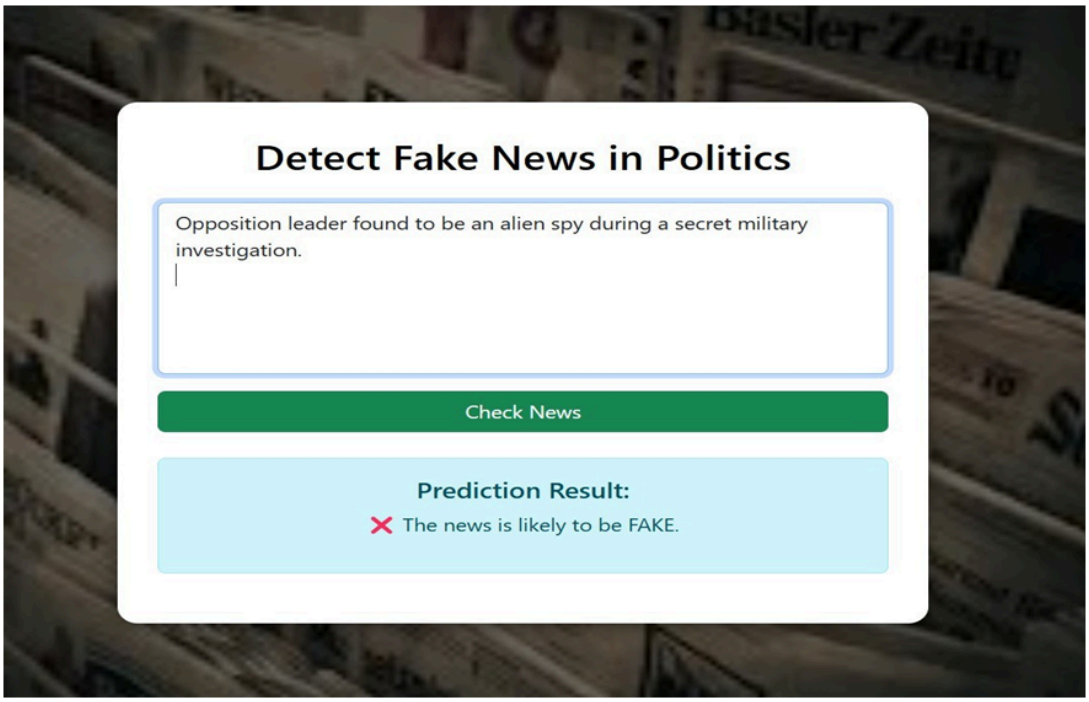

Fig 4.3–fake news prediction

6.CONCLUSION

In conclusion, The “Fake News Detection System” demonstrates the significant potential of machine learning and natural language processing in addressing one of today’s most pressing digital challenges: the rapid spread of misinformation. By employing a robust Multinomial Naive Bayes classifier in conjunction with TF-IDF for feature extraction, the system achieves high accuracy in identifying fake news based on textual content. The use of Python, coupled with Flask for web deployment, provides a powerful yet lightweight solution that ensures real-time usability for end users.This system significantly reduces the manual effort needed for verifying news articles, offering an accessible platform where users can instantly evaluate the authenticity of news headlines or articles. Preprocessing techniques such as stopword removal, stemming, and label encoding help optimize model performance and reduce noise. Furthermore, the intuitive user interface enhances usability, making the application suitable for users without technical expertise.The project reflects an

effective synthesis of backend machine learning models with a responsive frontend, leading to a practical and scalable application. It contributes to the broader societal goal of promoting digital literacy and responsible content consumption, setting a solid foundation for future improvements and real-world deployment.

7.FUTURE SCOPE

The Fake News Detection System, though currently effective with a Naive Bayes model and TF-IDF vectorization, holds immense potential for future enhancement and expansion. Integrating advanced deep learning models like LSTM, BERT, or other transformer-based architectures could significantly improve contextual understanding and classification accuracy, especially for more subtle or sophisticated misinformation. Expanding the system to support multiple languages would broaden its global usability, while real-time news scraping and classification capabilities could help flag misleading content as it emerges. Future developments could also include personalized user dashboards, query history tracking, and crowdsourced feedback features to improve interactivity and result accuracy. Moreover, adding image and video analysis for detecting fake multimedia content would transform the tool into a more comprehensive misinformation detection platform. On the technical side, cloud deployment and scalability options would make the system suitable for large-scale, enterprise-level use, particularly by media organizations and government agencies.

REFERENCES

[1]M. A. Idakwo, B. Salau Habeeb, O. Johnson and A. Busayo, "An Improved Ensemble-based Online Fake News Detection System," *2024 1st International Conference on Innovative Engineering Sciences and Technological Research (ICIESTR)*, Muscat, Oman, 2024, pp. 1-6, doi: 10.1109/ICIESTR60916.2024.10798365.

[2] X. Jose, S. D. M. Kumar and P. Chandran, "Characterization, Classification and Detection of Fake News in Online Social Media Networks," *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, Hassan, India, 2021, pp. 759-765, doi: 10.1109/MysuruCon52639.2021.9641517.

[3] S. El-Amrany, M. R. Brust, J. E. Pecero and P. Bouvry, "Tri-FusionDet: Leveraging User Engagement, Textual, and Visual Features for Enhanced Fake News Detection," *2024 28th International Computer Science and Engineering Conference (ICSEC)*, Khon Kaen, Thailand, 2024,

pp. 1-6, doi: 10.1109/ICSEC62781.2024.10770746.

[4] N. L. S. R. Krishna and M. Adimoolam, "Fake News Detection system using Decision Tree algorithm and compare textual property with Support Vector Machine algorithm," *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, Dubai, United Arab Emirates, 2022, pp. 1-6, doi: 10.1109/ICBATS54253.2022.9758999.

[5] S. Rastogi and D. Bansal, "Time is Important in Fake News Detection: a short review," *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, 2021, pp. 1441-1443, doi: 10.1109/CSCI54926.2021.00286.

[6] M. Jadhav, M. Patil, P. Giri and Y. Hande, "Advancing Fake News Detection: A Comparative Analysis of SVM and LR Models and Prospects for Dynamic Model Updating," *2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI)*, Wardha, India, 2024, pp. 1-5, doi: 10.1109/IDICAIEI61867.2024.10842732.

[7]C. Song, Y. Teng and B. Wu, "Dynamic Graph Neural Network for Fake News Detection," *2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, Xi'an, China, 2021, pp. 27-31, doi: 10.1109/CCIS53392.2021.9754681.

[8] A. Mewada, M. A. Ansari and S. K. Maurya, "From Misinformation to Truth: Fake News Detection with Transformer-Based Models," *2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT)*, Bhopal, India, 2025, pp. 1321-1326, doi: 10.1109/CSNT64827.2025.10967607.

[9] P. G, M. Germanaus Alex and S. John Peter, "Review of Fake News Detection in Social Media using Machine Learning Techniques," *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, Trichy, India, 2022, pp. 496-501, doi: 10.1109/ICAISS55157.2022.10010796.

[10] A. Jamshidi and M. Arif, "Revolutionizing Fake News Detection: A Quantum Machine Learning Paradigm," *2025 59th Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, USA, 2025, pp. 1-3, doi: 10.1109/CISS64860.2025.10944642.