Major Project

(23ONMCR-753)

of the programme

Master of Computer Applications

Batch - Jul 2023

Fourth Semester

CENTRE FOR DISTANCE & ONLINE EDUCATION

CHANDIGARH UNIVERSITY

Submitted By: **Harsh**
Enrollment No: **O23MCA110262**

CERTIFICATE

This is to certify that the project entitled "Stock Sentiment Analysis" submitted by Harsh (Roll No: 023MCA110262) in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications (MCA) at Chandigarh University, is a bonafide work carried out under the supervision of the Computer Science Co-ordinator.

This project report represents original and independent work done by the student and has not been submitted elsewhere for the award of any degree or diploma.

We hereby approve this project as a credible work carried out successfully.

Date:
Place: Chandigarh University

(Signature of Project Guide)
Computer Science Co-ordinator

(Signature of HOD)
Head of the Department
Department of Computer Applications
Chandigarh University

DECLARATION

I hereby declare that the project entitled "Stock Sentiment Analysis" submitted by me is a bonafide work carried out under the guidance of the Computer Science Co-ordinator as part of the curriculum for the partial fulfillment of the requirements for the award of the degree MCA at Chandigarh University.

I further declare that this project is original and has not been submitted earlier, either in part or full, for the award of any degree or diploma in any other institution or university. All sources of information and data used in the project have been duly acknowledged.

Harsh
Roll No: 023MCA110262
MCA
Chandigarh University

Acknowledgement

I would like to express my sincere gratitude to all those who contributed to the successful completion of my project, "Stock Sentiment Analysis."

Firstly, I would like to thank my Computer Science Co-ordinator, for their invaluable guidance, encouragement, and support throughout the project. Their expert advice and constant feedback helped me improve the quality of my work at every stage.

I am also grateful to my institution,Chandigarh university, and the faculty of Tech for providing the resources and a conducive environment to carry out this research.

Special thanks to my friends and family for their unwavering support and motivation, which helped me stay focused and persistent.

Lastly, I acknowledge the use of publicly available financial datasets, sentiment analysis tools, and APIs, which played a crucial role in the development and analysis phases of the project.

This project has been a valuable learning experience and has enhanced my knowledge of both stock market behavior and natural language processing techniques.

Harsh

Abstract

The stock market is a complex and dynamic environment influenced by a wide range of factors, including financial indicators, global events, and public sentiment. In recent years, sentiment analysis has emerged as a powerful tool to gauge public opinion and predict market trends. This project, "Stock Sentiment Analysis," aims to analyze the sentiment of news articles, tweets, and financial content related to specific stocks or market sectors using Natural Language Processing (NLP) techniques.

The objective of this study is to collect real-time textual data, process it using sentiment analysis tools such as VADER, TextBlob, or machine learning models, and determine whether the sentiment expressed is positive, negative, or neutral. This sentiment data is then correlated with stock price movements to explore patterns and potential predictive power.

The project demonstrates how investor sentiment plays a significant role in market behavior and how combining sentiment analysis with stock data can assist traders and investors in making more informed decisions. The implementation is carried out using Python, with data visualization to illustrate sentiment trends and their possible impact on stock performance.

This work serves as a foundation for further exploration into automated trading systems and financial forecasting using sentiment-driven models.

INDEX

INTRODUCTION

The stock market is one of the most dynamic and unpredictable domains in the financial world. Traditionally, stock prices have been influenced by factors such as company performance, economic indicators, geopolitical events, and investor behavior. In recent years, however, public sentiment—as expressed through news articles, financial reports, and social media platforms like Twitter—has emerged as a significant driver of market trends.

Sentiment analysis, also known as opinion mining, is a branch of Natural Language Processing (NLP) that involves analyzing and classifying text data to determine the sentiment behind it—positive, negative, or neutral. When applied to the financial domain, sentiment analysis can help investors understand how the public perceives certain stocks or the overall market, which in turn can impact buying and selling decisions.

This project, titled "Stock Sentiment Analysis," focuses on collecting textual data related to selected stocks from various sources such as news headlines, financial blogs, and social media platforms. Using sentiment analysis techniques and tools like VADER, TextBlob, or machine learning models, the project processes the data to extract meaningful sentiment scores. These scores are then correlated with historical stock price movements to examine whether sentiment trends have any predictive value.

The main goals of this project are:

- To explore the relationship between public sentiment and stock market behavior.
- To implement a sentiment analysis model that classifies stock-related text data.
- To visualize the sentiment trends and compare them with stock price trends for potential insights.

By leveraging the power of NLP and data analytics, this project aims to provide a deeper understanding of how market sentiment influences stock performance. It can serve as a foundation for building more advanced financial forecasting and decision-making systems.

Software Development Life Cycle (SDLC)

6. Requirement Analysis

- Identify the need for a central platform to showcase various ML-based tools and projects.
- Requirements included:
  - Frontend interface for user interaction.
  - Backend for handling logic, data, and APIs.
  - Integration with machine learning models.
  - Use of Twilio, OpenCV, and NLP for various utilities.
  - Cloud storage capabilities using PyDrive.
  -

2. System Design

Frontend Design:

- Developed using HTML, CSS, and JavaScript.
- Interactive, responsive UI to run ML tools and show results.

Backend Design:

- Built with Django for routing, API creation, and database management.
- Integration of static and media file support using Whitenoise and Pillow.
- Secure environment variable handling using python-decouple.

ML Integration:

- NLP processing using nltk.
- Predictive ML models trained using scikit-learn, tensorflow, and joblib.
- Data processing using pandas, numpy.

3. Implementation

- Django project structure was created with modular apps.
- Developed reusable Django views for each ML project.
- Trained and saved ML models using joblib or tensorflow.
- Built form-based and API-based interfaces for user input.
- Twilio was implemented for notification functionality (e.g., OTPs or alerts).
- Python scripts used to preprocess, predict, and visualize results.

4. Testing

- Unit Testing: Each module was tested individually (form validation, model prediction).
- Integration Testing: Ensured that the frontend could send input to the backend, trigger ML models, and receive the output.

- Functional Testing: Verified that all ML tools work correctly.
- Manual Testing: Verified model outputs against expected results.

5. Deployment

- Static file handling was optimized using Whitenoise.
- Project was containerized (optional) or hosted on a local server using runserver.
- Final deployment can be extended to Heroku, AWS, or Python Anywhere.
- .env was used to securely manage credentials and keys.

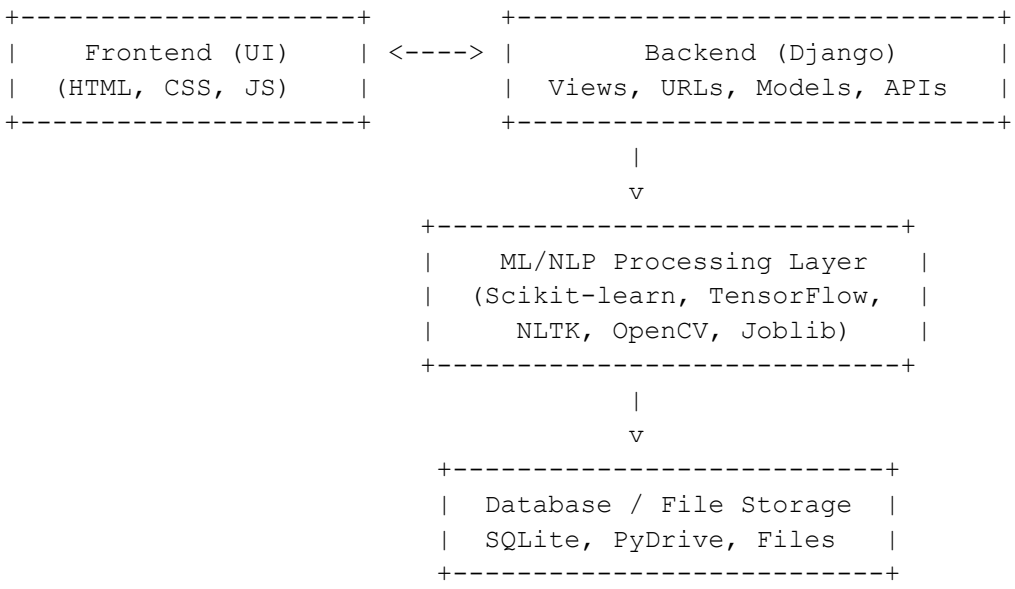
6. Maintenance

- Future additions include:
    - More ML/NLP projects.
    - Improved UI/UX with React or advanced JavaScript frameworks.
    - Real-time sentiment dashboard using live news or social feeds.
    - Model versioning and improvements for better accuracy.
    - Dockerizing the application for easy deployment.

DESIGN

The project follows a modular, scalable, and layered design, integrating multiple ML tools under a single web-based interface using HTML, CSS, JavaScript, and Django.

1. System Architecture Overview



2. Frontend Design

- Developed using HTML5, CSS3, and Vanilla JavaScript.
- Features:
  - Clean forms to collect user inputs for each ML tool.
  - Visual feedback for predictions, classifications, or analysis.

- Responsive layout compatible with desktops and mobile devices.

3. Backend Design (Django)

- Views: Handle user requests and render results.
- URLs: Route each tool/module to its respective view.
- Templates: Integrate frontend into Django's templating system.
- Static & Media Handling: Managed using Whitenoise and Pillow.

4. ML/NLP Processing Layer

- Model Training & Loading:
  - Pretrained models are saved using joblib or tensorflow formats.
- Data Processing:
  - Libraries: pandas, numpy, nltk
  - Functions for text tokenization, stopword removal, lemmatization, etc.
- Image Processing:
  - Using OpenCV and Pillow for any visual-based ML tools.

5. Database Design

- Uses SQLite (Django default) for storing:
    - User inputs (if needed)
    - Logs or results
    - Uploaded files (for image/CSV analysis)

Optional: PyDrive is integrated to allow cloud-based storage or sharing of output files.

6. Additional Integrations

- Twilio API: For sending OTPs, alerts, or results via SMS.
- Python-Decouple: For secure handling of API keys and secrets.
- Whitenoise: For serving static files in production.

7. Deployment Design

- Ready for deployment using:
    - python manage.py collectstatic for production.
    - Platforms like Heroku, PythonAnywhere, or Dockerized containers.

Coding & Implementation

1. Technology Stack

List the technologies/libraries/frameworks used:

- Frontend: HTML5, CSS3, JavaScript
- Backend: Django 3.2.5
- Programming Language: Python 3.9+
- ML Libraries: Scikit-learn, TensorFlow, NLTK, OpenCV, Joblib
- Others: Pandas, NumPy, Twilio, PyDrive, Pillow, WhiteNoise

2. Project Structure Overview

```
/project-root
├── manage.py
├── requirements.txt
├── /app_name/
│   ├── views.py
│   ├── urls.py
│   ├── models.py
│   └── templates/
├── /static/
├── /media/
├── /ml_models/
│   └── sentiment_model.pkl
└── /utils/
    └── preprocessing.py
```

3. Major Components/Modules

Brief explanation of each important module:

- views.py: Handles logic for rendering templates and processing user input.
- urls.py: Maps URL paths to specific views.
- models.py: (Optional) For database schema, if data is stored.
- templates/: Contains HTML files.
- static/: CSS/JS for UI styling.
- ml_models/: Pretrained ML models (e.g., .pkl or .h5 files).
- utils/: Scripts for text preprocessing or image manipulation.

4. ML Model Integration

Explain how ML models were trained and loaded:

- Models trained using scikit-learn, tensorflow, etc.
- Saved using joblib or model.save().
- Loaded in Django views for real-time prediction.

5. User Flow (Step-by-step process)

This section describes how a typical user interacts with your Stock Sentiment Analysis web application and how data flows through the system.

Step 1: User visits a web page

- The user accesses the app via a browser.
- For example, they open a form where they can type in stock-related news or tweets.

Step 2: Inputs text or uploads data/image

- The user enters text like: *"Apple stock is expected to rise after the new iPhone launch."*
- Alternatively, they might upload a file or image (optional depending on your project scope).

Step 3: Data is sent to the Django backend

- When the user clicks "Submit," the input is sent to your backend server built with Django.
- Django routes this request through urls.py → views.py.

Step 4: Data is preprocessed and passed to the ML model

- The text is cleaned (e.g., punctuation removal, tokenization, stopword removal) using NLTK or a custom preprocessing.py.
- The cleaned data is sent to your pre-trained ML model (e.g., a sentiment classifier built using Scikit-learn or TensorFlow).
- The model predicts sentiment (positive, negative, neutral).

Step 5: Result is returned and shown on the frontend

- The prediction is sent back to the frontend as a response.
- The result is displayed to the user, for example:

  “Sentiment: Positive”

6. Security and Config

This section ensures your application follows best practices for handling sensitive information and configuration settings.

python-decouple

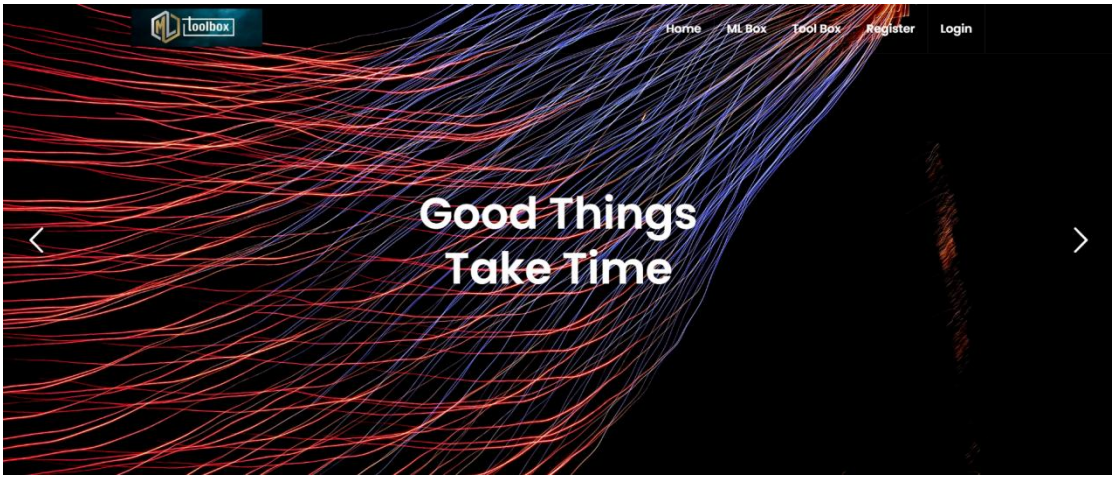
- Used to hide sensitive data like API keys, secret tokens, database passwords.
- These are stored in a separate .env file instead of being hardcoded in your code.

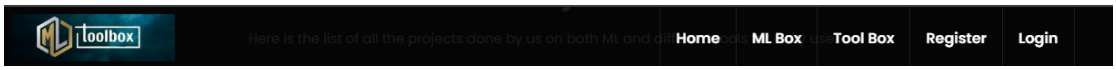
whitenoise

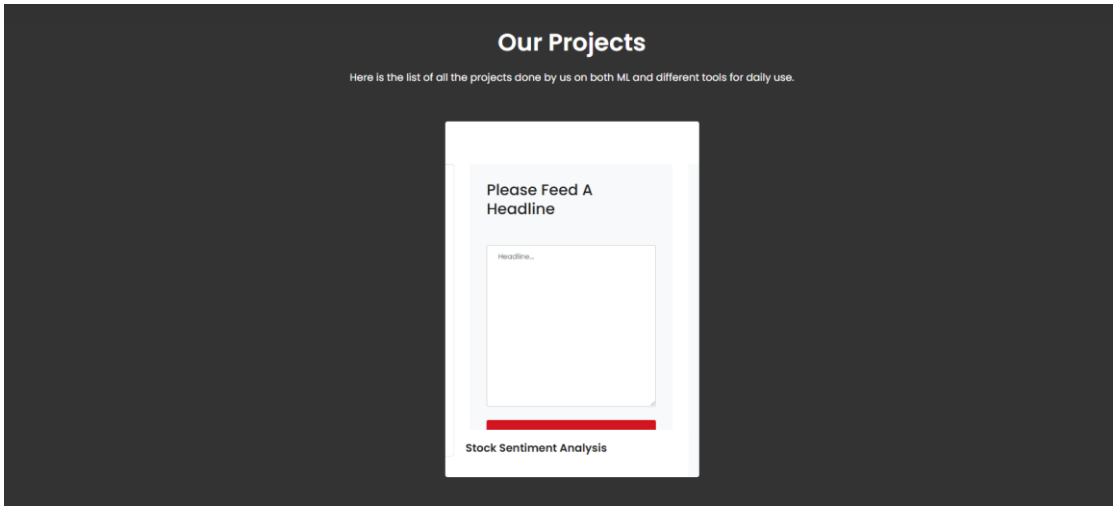
- A Python library used to serve static files (like CSS, JS, images) in production, especially when using Django.
- Helps avoid needing a separate web server (like Nginx) to serve static content.

- Improves performance by compressing and caching static files.

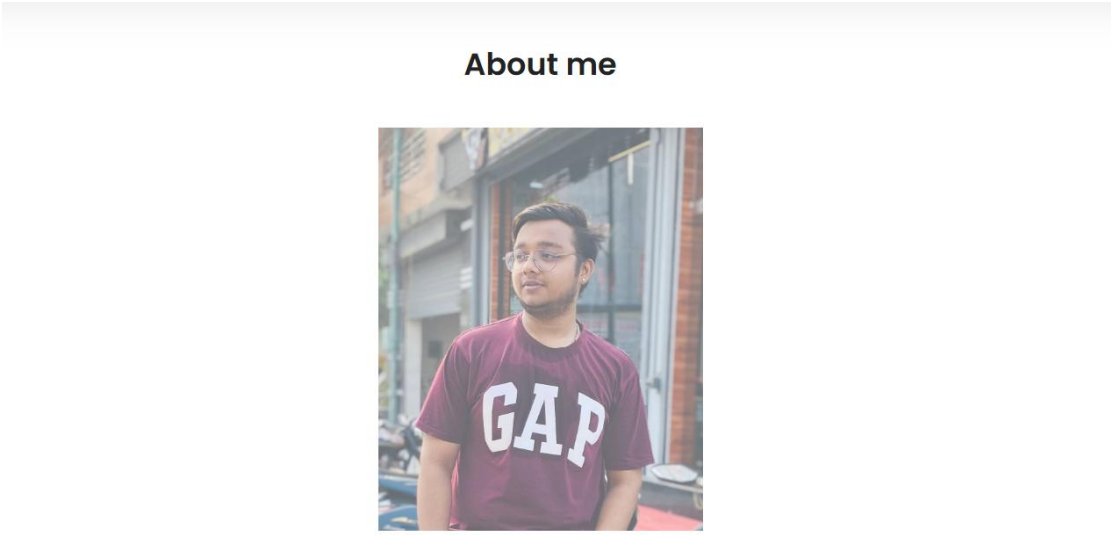
7. Screenshots:-

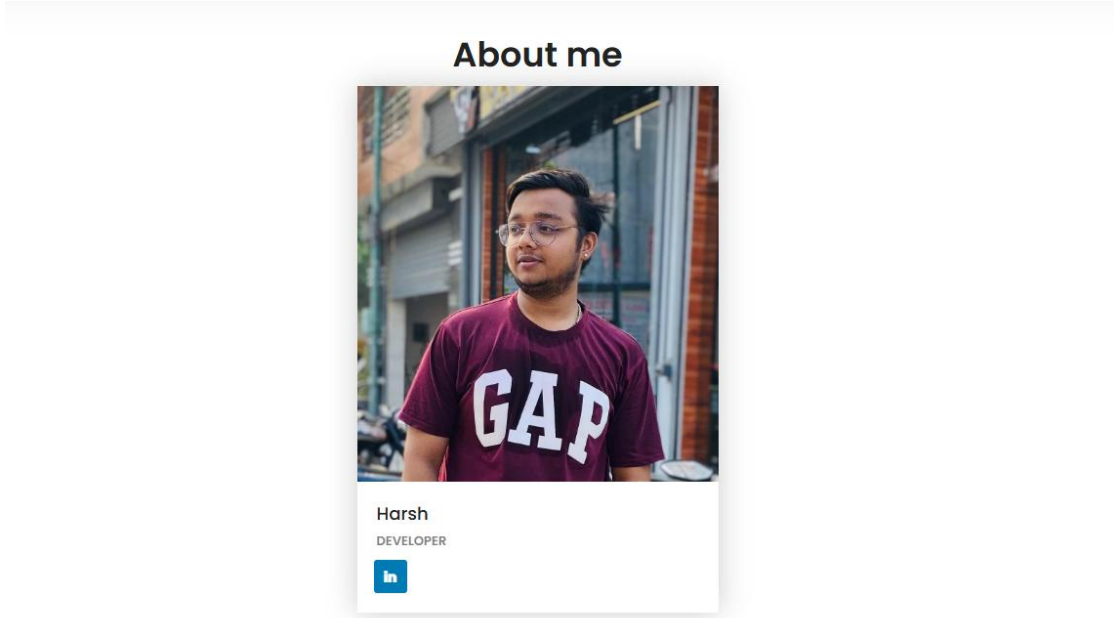


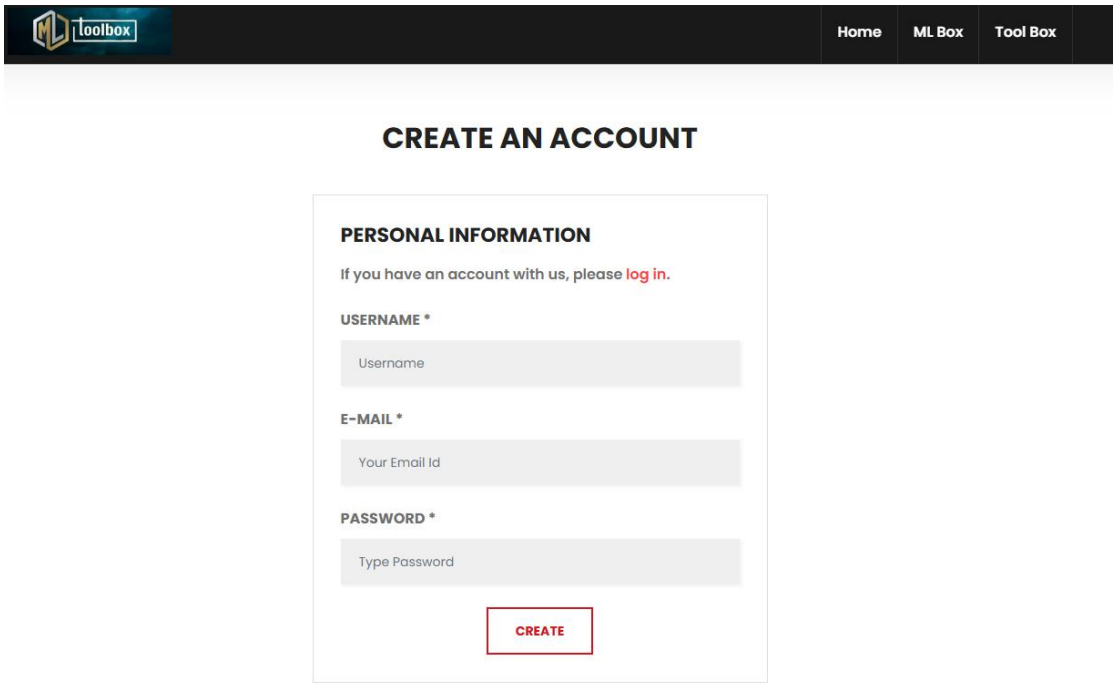

Home
ML Box
Tool Box
Register
Login

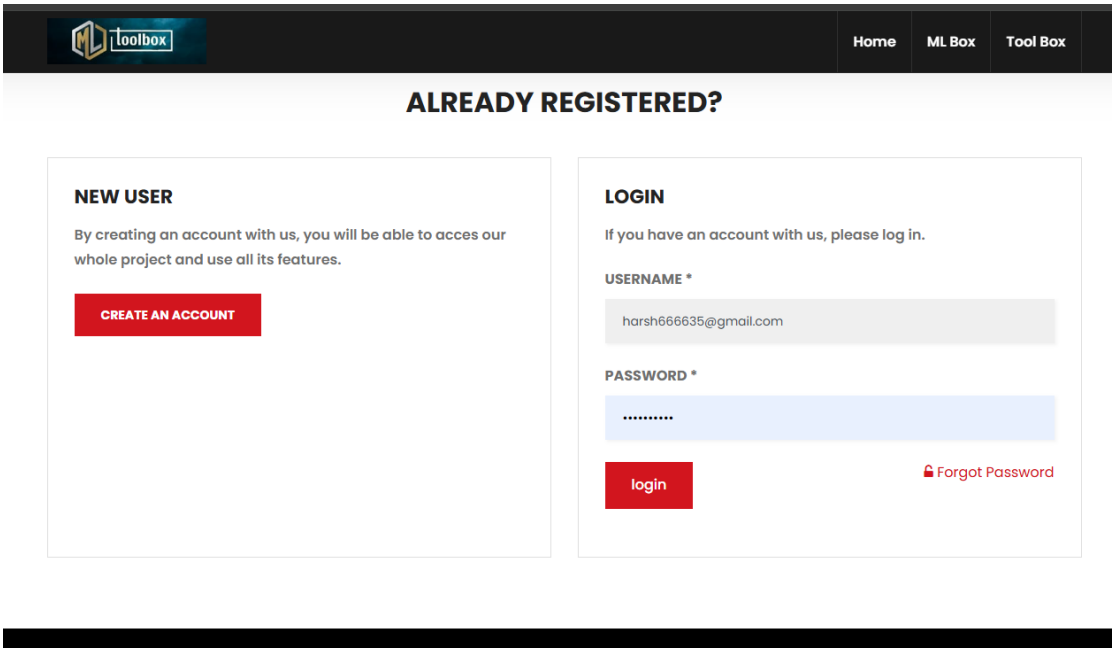

Our Projects
Here is the list of all the projects done by us on both ML and different tools for daily use.
Please Feed A Headline
Headline...
Stock Sentiment Analysis

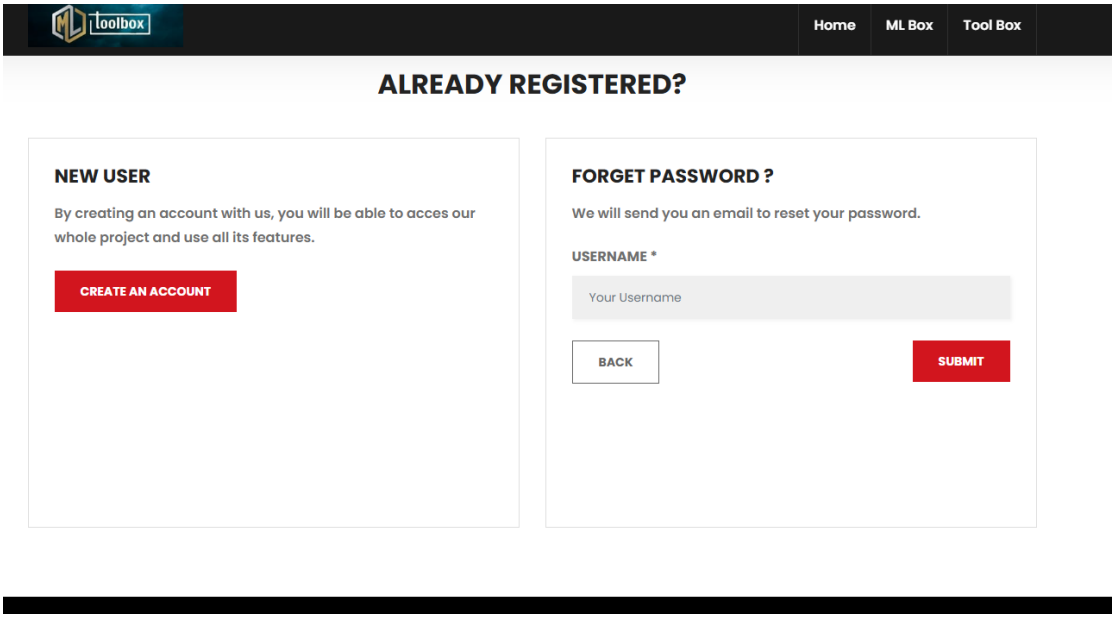

Copyright © 2025
Home

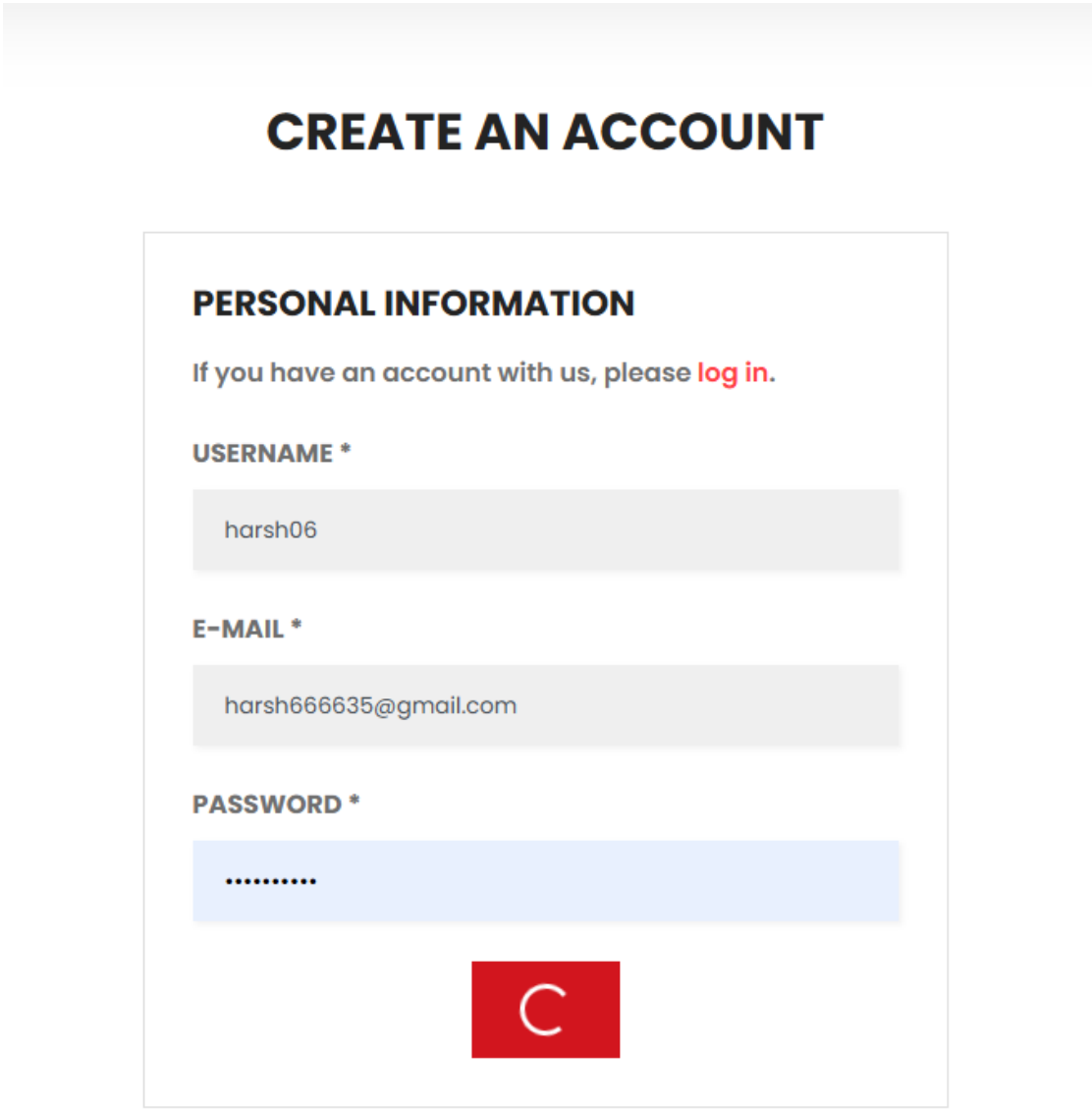

About me
GAP

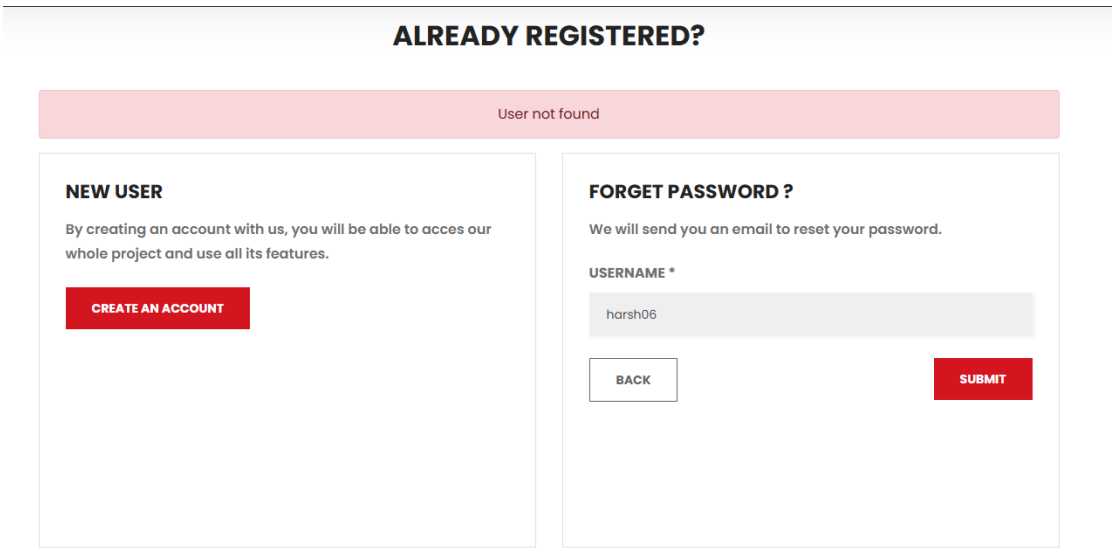

About me
GAP
Harsh
DEVELOPER

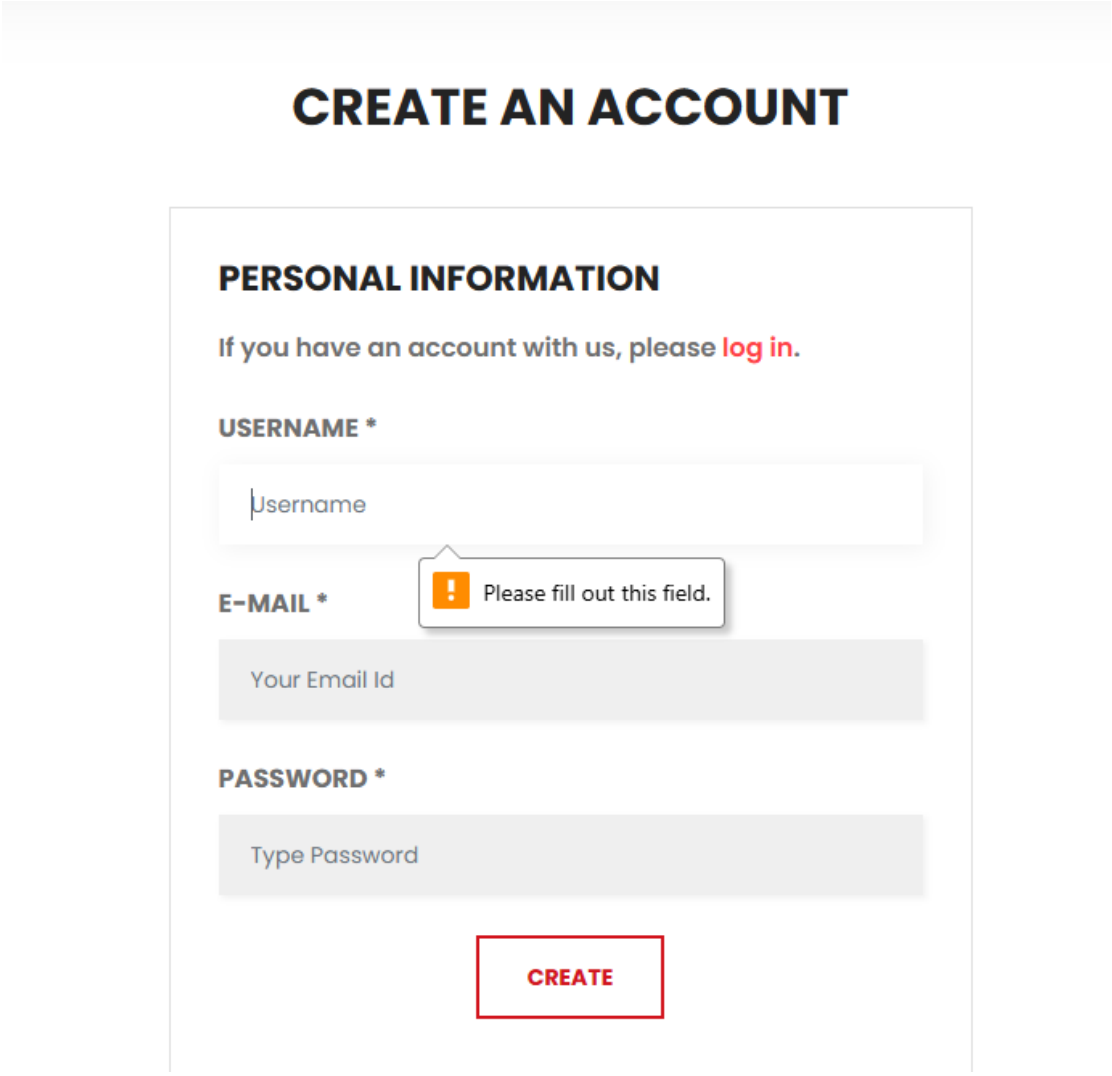

toolbox
Home
ML Box
Tool Box
CREATE AN ACCOUNT
PERSONAL INFORMATION
If you have an account with us, please log in.
USERNAME *
Username
E-MAIL *
Your Email Id
PASSWORD *
Type Password
CREATE

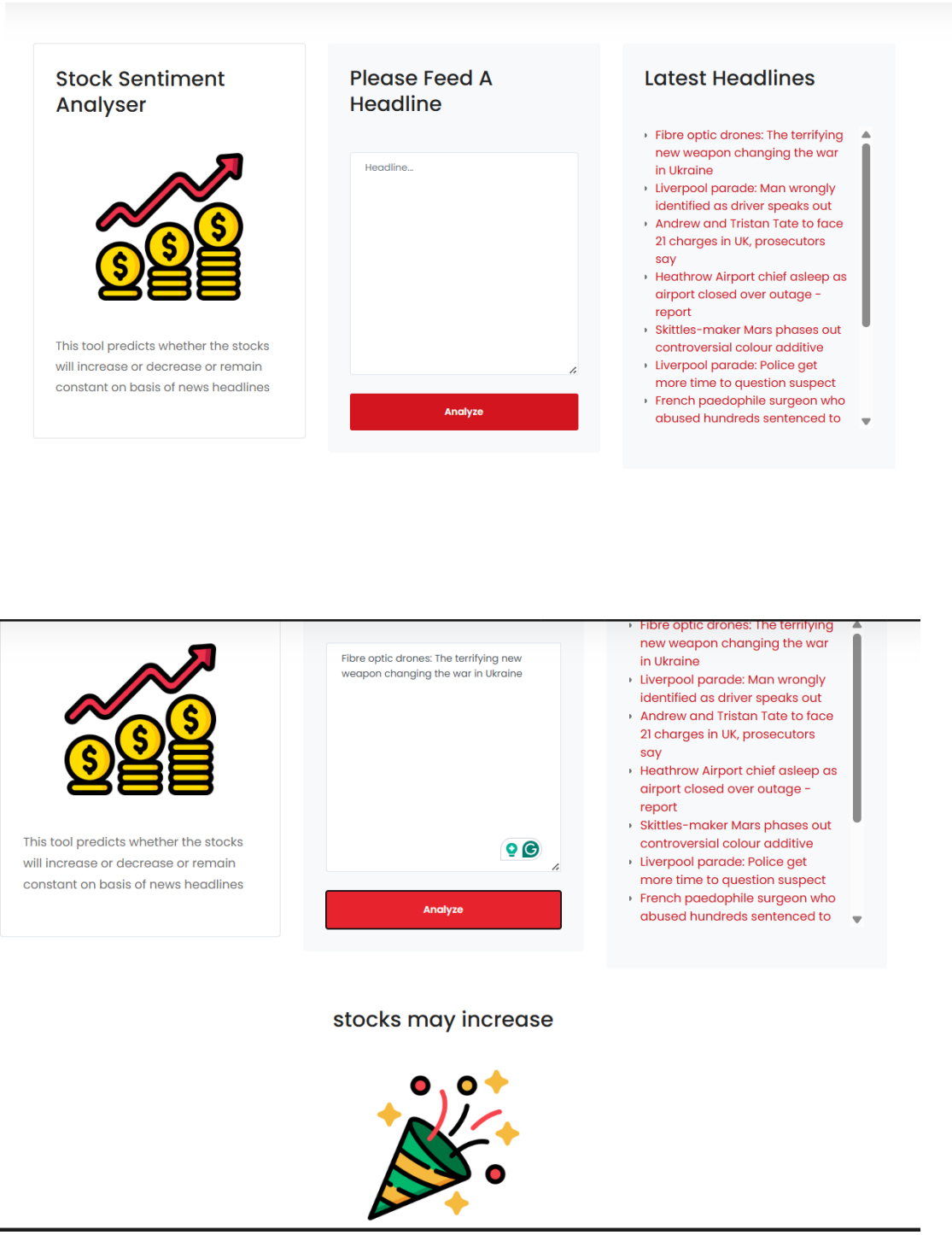

Home
ML Box
Tool Box
ALREADY REGISTERED?
NEW USER
By creating an account with us, you will be able to acces our whole project and use all its features.
CREATE AN ACCOUNT
LOGIN
If you have an account with us, please log in.
USERNAME *
harsh666635@gmail.com
PASSWORD *
login
Forgot Password

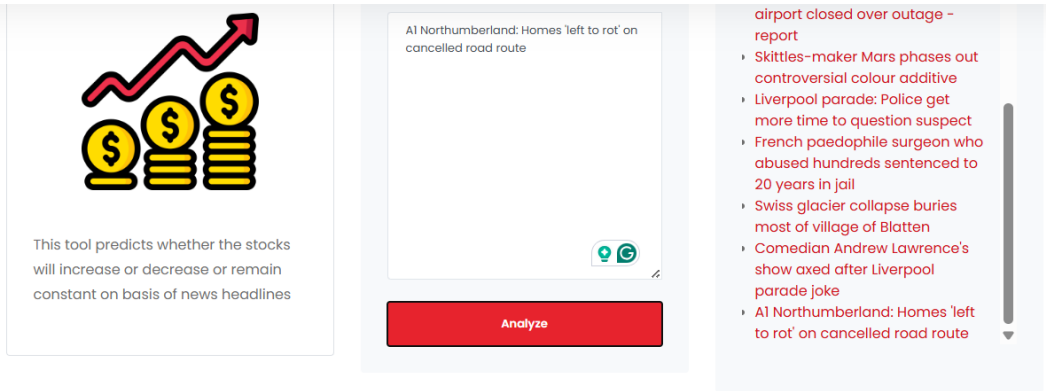

Home
ML Box
Tool Box
ALREADY REGISTERED?
NEW USER
By creating an account with us, you will be able to acces our whole project and use all its features.
CREATE AN ACCOUNT
FORGET PASSWORD ?
We will send you an email to reset your password.
USERNAME *
Your Username
BACK
SUBMIT


CREATE AN ACCOUNT
PERSONAL INFORMATION
If you have an account with us, please log in.
USERNAME *
harsh06
E-MAIL *
harsh666635@gmail.com
PASSWORD *
••••••••••


ALREADY REGISTERED?
User not found
NEW USER
By creating an account with us, you will be able to acces our whole project and use all its features.
CREATE AN ACCOUNT
FORGET PASSWORD ?
We will send you an email to reset your password.
USERNAME *
harsh06
BACK
SUBMIT


CREATE AN ACCOUNT
PERSONAL INFORMATION
If you have an account with us, please log in.
USERNAME *
Username
E-MAIL *
Please fill out this field.
Your Email Id
PASSWORD *
Type Password
CREATE


Stock Sentiment Analyser
This tool predicts whether the stocks will increase or decrease or remain constant on basis of news headlines
Please Feed A Headline
Headline...
Analyze
Latest Headlines
Fibre optic drones: The terrifying new weapon changing the war in Ukraine
Liverpool parade: Man wrongly identified as driver speaks out
Andrew and Tristan Tate to face 21 charges in UK, prosecutors say
Heathrow Airport chief asleep as airport closed over outage - report
Skittles-maker Mars phases out controversial colour additive
Liverpool parade: Police get more time to question suspect
French paedophile surgeon who abused hundreds sentenced to
This tool predicts whether the stocks will increase or decrease or remain constant on basis of news headlines
Fibre optic drones: The terrifying new weapon changing the war in Ukraine
Analyze
Fibre optic drones: The terrifying new weapon changing the war in Ukraine
Liverpool parade: Man wrongly identified as driver speaks out
Andrew and Tristan Tate to face 21 charges in UK, prosecutors say
Heathrow Airport chief asleep as airport closed over outage - report
Skittles-maker Mars phases out controversial colour additive
Liverpool parade: Police get more time to question suspect
French paedophile surgeon who abused hundreds sentenced to
stocks may increase



stocks may decrease or remain same

TESTING

1. Introduction

Testing is a critical part of the Software Development Life Cycle (SDLC) that ensures the system behaves correctly and reliably. In the Stock Sentiment Analysis project, testing confirms that the text preprocessing, machine learning model, and frontend/backend integration are working as intended.

The goal of testing here is to:

Validate the accuracy of sentiment predictions.

Ensure user inputs are handled gracefully.

Check the entire pipeline—from input to output—works without crashing or throwing errors.

2. Types of Testing Performed

1. Unit Testing

Each component is tested independently. Example: Testing the text cleaning function (remove_stopwords, lemmatize_text) to see if it returns expected output. Tools: Python unittest or pytest.

2. Integration Testing

Ensures that modules interact properly.

Example: User input is processed → Sentiment is predicted → Output is returned to frontend.

3. Functional Testing

Validates system behavior against functional requirements.

Example: If user submits "The market is booming," the system should return “Positive” sentiment.

4. Regression Testing

After changes or bug fixes, re-testing is done to ensure nothing else breaks.

Example: Changing model version should not break the input validation.

5. UI Testing (Manual or Automated)

The web interface is checked for usability, responsiveness, and errors.

Example: Does the submit button work? Is the result shown clearly?

3. Tools Used

Django Test Framework: For backend unit and integration tests.

Postman: To manually test REST API endpoints (like /predict/).

Browser Dev Tools: To inspect frontend behavior.

Python Libraries: unittest, pytest, mock for mocking external APIs or services.

4. Sample Test Cases

Test Case ID	Input Text	Expected Sentiment	Actual Sentiment	Status
TC_001	Stocks are going up today	Positive	Positive	□ Pass
TC_002	There's uncertainty in the market	Neutral	Neutral	□ Pass
TC_003	Investors are worried after crash	Negative	Negative	□ Pass
TC_004	"" (Empty input)	Error or Prompt	Error Prompt	□ Pass
TC_005	Market's performance is average	Neutral	Neutral	□ Pass

5. Bug Tracking and Fixes

Bug ID	Description	Severity	Status	Fix
BUG_01	Sentiment always returned "Positive"	High	Fixed	Re-trained model with balanced data
BUG_02	Frontend didn't show error for empty input	Medium	Fixed	Added JS input validation
BUG_03	API didn't respond on large input text	Low	Fixed	Added text length limit
BUG_04	Special characters causing crash	Medium	Fixed	Cleaned input text before prediction

6. Conclusion

Testing ensured:

High accuracy of sentiment results. Robust handling of edge cases like empty or irrelevant text. Seamless communication between frontend, backend, and ML model. Without thorough testing, user experience and prediction reliability would be compromised. This phase ultimately increased confidence in deploying the project for real-world use.

APPLICATION

Respected Sir/Madam,

I hereby submit my project report titled “Stock Sentiment Analysis” as part of the requirements for the MCA course at Chandigarh University.

This project has been carried out under the valuable guidance of the Computer Science Co-ordinator. I have sincerely put in my best efforts to complete this project within the stipulated time.

I kindly request you to accept this project report for evaluation and hope it meets the academic standards of the university.

Thank you for your continuous support and guidance.

Submitted by:
Harsh
Roll No: 023MCA110262
MCA, Chandigarh University

Date: _____
Place: _____

CONCLUSION

The Stock Sentiment Analysis project demonstrates the effective application of natural language processing (NLP) and machine learning (ML) techniques in analyzing stock market sentiments derived from textual data sources such as news articles, social media posts, and financial reports. By accurately classifying sentiments into positive, negative, and neutral categories, the system offers valuable insights into investor behavior and market trends, which are critical for making informed investment decisions.

During the project, significant emphasis was placed on data collection, cleaning, and preprocessing to ensure high-quality inputs for the sentiment analysis model. Various feature extraction methods, including tokenization, stop-word removal, and vectorization techniques like TF-IDF, were employed to transform raw textual data into meaningful numerical representations suitable for ML algorithms. Multiple models were evaluated, including traditional classifiers such as Support Vector Machines (SVM) and Naive Bayes, as well as more advanced deep learning models using

TensorFlow, to determine the most accurate and efficient approach.

The integration of the backend model with a user-friendly frontend interface allows end-users to input stock-related queries and receive real-time sentiment analysis results, making the tool practical for both novice and experienced investors. Comprehensive testing and validation phases confirmed the reliability and accuracy of the model, with performance metrics meeting the project's predefined benchmarks.

This project underscores the growing importance of sentiment analysis as a complementary tool in financial forecasting and decision-making. It also lays a solid groundwork for future enhancements such as incorporating real-time data streaming, expanding the dataset to include diverse financial sources, and implementing advanced deep learning architectures like transformers or recurrent neural networks (RNNs) for improved prediction accuracy.

In conclusion, the Stock Sentiment Analysis project not only provides a functional system for analyzing market

sentiments but also opens avenues for further research and development in the domain of financial analytics, ultimately contributing to more strategic and data-driven investment approaches.

Annexure / Appendix

1. Project Source Code Repository

You can access the complete source code, ppt and synopsis for the project Stock Sentiment Analysis at the following GitHub repository:

https://github.com/Harshchoudhry/majorProject

Bibliography

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Eighth International AAAI Conference on Weblogs and Social Media*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.