A MINI-PROJECT REPORT

ON

# “Trade Tavern”

BY

## Soham Mandavkar Pooja Mayekar Kamran Dhopaunkar Advait Jadhav

Under the guidance of

##### Internal Guide Prof. Dilip Dalgade



Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53 University of Mumbai

NOVEMBER- 2024



Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53

### C E R T I F I C A T E

Department of Computer Engineering

This is to certify that

1. Soham Mandavkar
2. Pooja Mayekar
3. Kamran Dhopaunkar
4. Advait Jadhav

Have satisfactorily completed this project entitled

##### “TradeTavern: ML-Driven Market Predictions”

Towards the partial fulfilment of the

**THIRD YEAR BACHELOR OF ENGINEERING IN**

**(COMPUTER ENGINEERING)**

**as laid by University of Mumbai**.

**Prof. Dilip Dalgade Prof. Sunil P. Khachane**

**Guide H.O.D.**

**Dr. Sanjay Bokade**

**Principal**

#### Project Report Approval for T. E.

This project report entitled **“Trade Tavern*”*** by ***Soham Mandavkar, Pooja Mayekar, Kamran Dhopaunkar, Advait Jadhav*** is approved for the degree of ***Third-Year Bachelor of Computer Engineering***.

##### Examiners:

1

2.

Date:

Place:

#### Declaration

We wish to state that the work embodied in this project titled **“Trade Tavern”** forms our own contribution to the work carried out under the guidance of “**Prof. Dilip Dalgade**” at the Rajiv Gandhi Institute of Technology.

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

|  |  |  |
| --- | --- | --- |
| **(Students Signatures)** |  | |
| Soham Mandavkar | (A-554) |  |
| Pooja Mayekar | (A-556) |  |
| Kamran Dhopaunkar | (A-561) |  |
| Advait Jadhav | (A-564) |  |

## Abstract

This project introduces TradeTravern, a comprehensive stock market trading application designed to empower users with advanced trading capabilities and accurate market predictions. The need for such an application arises from the challenges that traders frequently encounter, including a limited understanding of market dynamics, difficulties in making informed investment decisions, and a lack of real-time insights into stock performance.

To address these challenges, TradeTravern leverages machine learning algorithms, specifically the Long Short-Term Memory (LSTM) model, which excels in capturing temporal dependencies in sequential data. This enables the application to provide precise price predictions for various stocks, allowing users to anticipate market movements and optimize their trading strategies. Built using ReactJS for a dynamic and user-friendly interface and Firebase for robust backend support, TradeTravern facilitates real-time trading and notifications, ensuring users never miss critical investment opportunities.

In addition to its predictive capabilities, the application features dynamic visualizations, including line graphs and various charts, which enhance users' ability to analyze market trends effectively. The methodology involves data collection, preprocessing, and model training, aimed at delivering an intuitive platform that notifies users when to invest at the right time.

While significant progress has been made in developing TradeTravern, the project is still ongoing. Continuous efforts are focused on refining the user experience, enhancing predictive accuracy, and incorporating user feedback. Ultimately, TradeTravern aspires to be a valuable tool for both novice and experienced traders, fostering a deeper understanding of market dynamics and maximizing investment opportunities.

## Contents

|  |  |  |
| --- | --- | --- |
| 1 | **Introduction**   * 1. Introduction Description...……………………………………….   2. Organization of Report. …………………………………………. | 1  1 |
| 2 | **Literature Review**   * 1. Survey Existing system………………………………………….   2. Problem Statement………………………...……………………..   3. Objective………………………...……………………………….   4. Scope…………………………………………………………….. | 2  3  3  3 |
| 3 | **Proposed System**   * 1. Proposed System………………………………………………..      1. Existing Model…………………………………………….      2. Proposed Model……………………………………………      3. System Architecture……………………………………….   2. Details of Hardware & Software………………………………...      1. Software Requirement.…………………………………….      2. Hardware Requirement……………………………………...   3. Design Details……………………………………………………..      1. Activity Diagram………………...………………………….   4. Methodology/Procedures (Your methodology to solve problem).. | 4  4  5  6  6  7  8 |
| 4 | **Results**  4.1.1 Landing page………………………………………………………  4.1.2 Signup page………………………………………………………..  4.1.3 Login page………………………………………………………...  4.1.4 home page…………………………………………………………  4.1.5 Subscription page………………………………………………….  4.1.6 Database…………………………………………………………... | 9  9  10  10  11  11 |
| 5 | **Reference** | 13 |

### CHAPTER 1

##### Introduction Description

This project introduces TradeTravern, a comprehensive stock market trading application designed to empower users with advanced trading capabilities and precise market predictions. By merging powerful machine learning algorithms with a user-friendly interface, TradeTravern aims to provide both novice and seasoned traders with the tools necessary for informed decision-making in the stock market

At its core, TradeTravern leverages powerful machine learning models, specifically the Long Short-Term Memory (LSTM) algorithm, to deliver precise price predictions for a variety of stocks. This predictive capability allows users to anticipate market movements and optimize their trading strategies. By integrating real-time stock trading functionalities, TradeTravern ensures that users can act on these predictions immediately, enhancing their potential for profit.

The application is built using ReactJS, a popular JavaScript library known for its flexibility and performance, ensuring a seamless user experience. Additionally, Firebase is utilized for backend support, providing a reliable framework for data storage, user authentication, and real- time notifications. This integration allows TradeTravern to alert users when optimal investment opportunities arise, ensuring they never miss the right moment to buy or sell.

TradeTravern also features dynamic visualizations of stock data, including line graphs and various charts, which facilitate better understanding and analysis of market trends. By presenting data in an intuitive manner, users can quickly grasp the nuances of stock performance and make well-informed decisions.

Overall, TradeTravern represents a significant step forward in the realm of stock trading applications. By merging advanced technology with user-centric design, it offers a comprehensive platform that not only supports trading activities but also empowers users with the insights needed to navigate the complexities of the stock market effectively. As we embark on this project, our goal is to refine this application into a valuable tool for traders, fostering a deeper understanding of market dynamics while maximizing investment opportunities.

##### Organization of report

Describe every chapter (what every chapter contains)

* + - Ch.1 Introduction: This section covers the introduction of our project.
    - Ch.2 Literature Review: This sections covers the ideology of other papers
    - Ch.3 Proposed System: This sections covers the methodology of our project
    - Ch.4 Results: This section provides the Results of our project

### CHAPTER 2

##### Survey Existing system

This literature review synthesizes insights from key research papers that explore stock market prediction using machine learning techniques. The studies highlight various methodologies, algorithms, and their implications for improving prediction accuracy and trading strategies.

1. This study presents a comparative analysis of several machine learning models, including decision trees, random forests, and neural networks. The critical role of feature selection and data preprocessing is emphasized to achieve high predictive accuracy. By utilizing historical stock data to train models, the findings indicate that machine learning approaches significantly outperform traditional statistical methods. The study concludes that leveraging these techniques can provide investors with a reliable framework for making informed trading decisions, ultimately enhancing profitability.
2. This research investigates the application of algorithms such as linear regression, support vector machines (SVM), and k-nearest neighbors (KNN) for stock price forecasting. It addresses challenges in data collection and preprocessing, underscoring the importance of obtaining accurate historical data for effective model training. The findings suggest that machine learning models can produce promising results in predicting stock trends, contributing to the development of sophisticated trading strategies. This research highlights the need for continuous improvement in data handling and algorithm performance.
3. This review provides a comprehensive overview of machine learning techniques applied to stock market prediction. Methodologies are categorized into supervised and unsupervised learning, discussing the advantages and limitations of each. The significance of feature engineering, data normalization, and evaluation metrics in achieving high prediction accuracy is emphasized. This review serves as an essential resource for researchers and practitioners, summarizing the state of the art in stock market prediction and identifying future research directions.

Collectively, these studies underscore the transformative potential of machine learning in stock market prediction, emphasizing the importance of robust data preprocessing and feature selection for enhancing model accuracy. They demonstrate that different algorithms yield varying results, suggesting a hybrid approach could be beneficial for optimal predictions. The literature also highlights the growing interest in applying machine learning techniques to stock prediction, with future advancements relying on integrating real-time data analysis and advanced algorithms to improve predictive capabilities and support informed investment decisions.

##### Problem Statement

* + 1. To implement a stock market prediction system that provides users with both a fundamental understanding of the stock market and accurate price predictions.
    2. Existing challenges in stock market education and prediction: Limited understanding of market dynamics, difficulty in making informed predictions, and lack of real-time insights.

##### Objectives

* + 1. **Enable Real-Time Trading**: Ensures users can trade stocks in real-time with an intuitive interface.
    2. **Implement LSTM Predictions:** Provides advanced stock price predictions using LSTM algorithms.
    3. **Deliver Interactive Visualizations:** Offers dynamic visualizations to help users understand and analyze stock data.
    4. **Ensure Robust Performance**: Ensures the platform is scalable and reliable using React and Flask.

##### Scope

1. For the front-end:
   * Create an intuitive and visually appealing interface to enhance user engagement.
   * Ensure smooth navigation and real-time updates to provide an immersive and interactive experience.
2. For the back-end:
   * Establish a robust data management system to notice creation, user information, and results securely and efficiently.
   * Implement reliable performance tracking and analytics tools to provide colleges with comprehensive insights into student progress for some assignments.

**CHAPTER 3**

# 3.1Proposed System

##### C:\Users\USER\Downloads\TradeTavern - System Architecture (4).png3.1.1 Existing model

Fig. 3.1 .1Existing Model

The existing model for stock price prediction utilizes a Long Short-Term Memory (LSTM) neural network. The process begins by collecting historical stock data, which is then preprocessed to clean and format it appropriately for analysis. The data is split into training and test sets, using a 99:1 ratio, with the majority allocated to training.

The LSTM model is employed to learn from the training data by capturing the time-dependent patterns in the stock prices. After the model is trained, it makes predictions that are subsequently tested using the test set. Finally, the model's performance is evaluated to determine its prediction accuracy, completing the stock prediction pipeline.

##### C:\Users\USER\Downloads\TradeTavern - System Architecture (2).pngProposed Model

Fig. 3.1.2 Proposed Model

The proposed stock price prediction model leverages a Long Short-Term Memory (LSTM) neural network to forecast stock prices. It begins with the collection of historical stock data, which is then preprocessed through data cleaning, scaling, and transformation to ensure it is suitable for training. After preprocessing, the data is split into training and testing sets, typically in an 80:20 ratio.

The core of the model is the LSTM network, which is designed to capture the temporal dependencies in the stock data. It uses internal mechanisms like the Forget Gate, Input Gate, Cell Gate, and Output Gate to manage information flow and optimize learning over time. Throughout the training process, various optimizations are applied to improve the accuracy of the model.

Once the training is complete, predictions are made using the trained LSTM model, and the test set is used to evaluate the model's accuracy. The final output of the model is a set of predicted stock prices, which are compared with actual stock prices to assess the model’s performance.

##### C:\Users\USER\Downloads\TradeTavern - System Architecture (5).pngSystem Architecture

Fig. 3.1.3 System Architecture

The system architecture of this stock trading platform integrates React.js for the front-end, Vite as middleware, and Firebase for the backend. The front-end consists of pages like login/register, home, investment options, a dashboard, and individual stock templates. Real-time stock data is fetched through an API and passed via Vite, which facilitates smooth communication between the front-end and backend. Firebase manages user information and transactions, while a prediction model forecasts stock prices. A local server is employed to update the model as necessary, ensuring accurate predictions and real-time data flow across the platform.

##### Details of hardware and software

* + 1. **Software requirements:**
       - **Frontend**
* **React:** For building the user interface and handling client-side operations.
* **Chart.js / D3.js:** For rendering stock price visualizations (line graphs and charts).
  + - * **Backend:**
* **Flask:** For managing server-side operations, handling routing, and rendering HTML templates.
* **Flask-RESTful API:** For creating APIs to connect the frontend with the backend.
  + **Database**:
* **Firebase**: For storing user data and trading records.
  + **Machine Learning**:
* **TensorFlow Python**: For implementing the LSTM algorithm.
* **Keras**: For building and training the LSTM model.
  + **Development Tools**:
* **VS Code / PyCharm**: For coding and debugging
* **Git**: For version control.
* **Firebase/heroku** : (Optional) For containerizing the application for easier deployment.

##### Hardware requirements

* **Server Requirements:**
  + **Processor:** Intel Core i5 or equivalent (for development purposes).
  + **RAM: 16** GB (minimum) or more.
  + **Storage:** 256 GB SSD (for development and deployment).
  + **Network:** Stable internet connection for real-time a and model updates.
* **Model training Requirements:**Google colab account
  + **.**
* **User Requirements:**
  + **Processor:** Any modern processor.
  + **Storage: Storage:** Sufficient storage for application use and data.
  + **RAM:** 4 GB (minimum).
  + **Network:** Stable internet connection for real-time data and model updates.

##### Design Detail

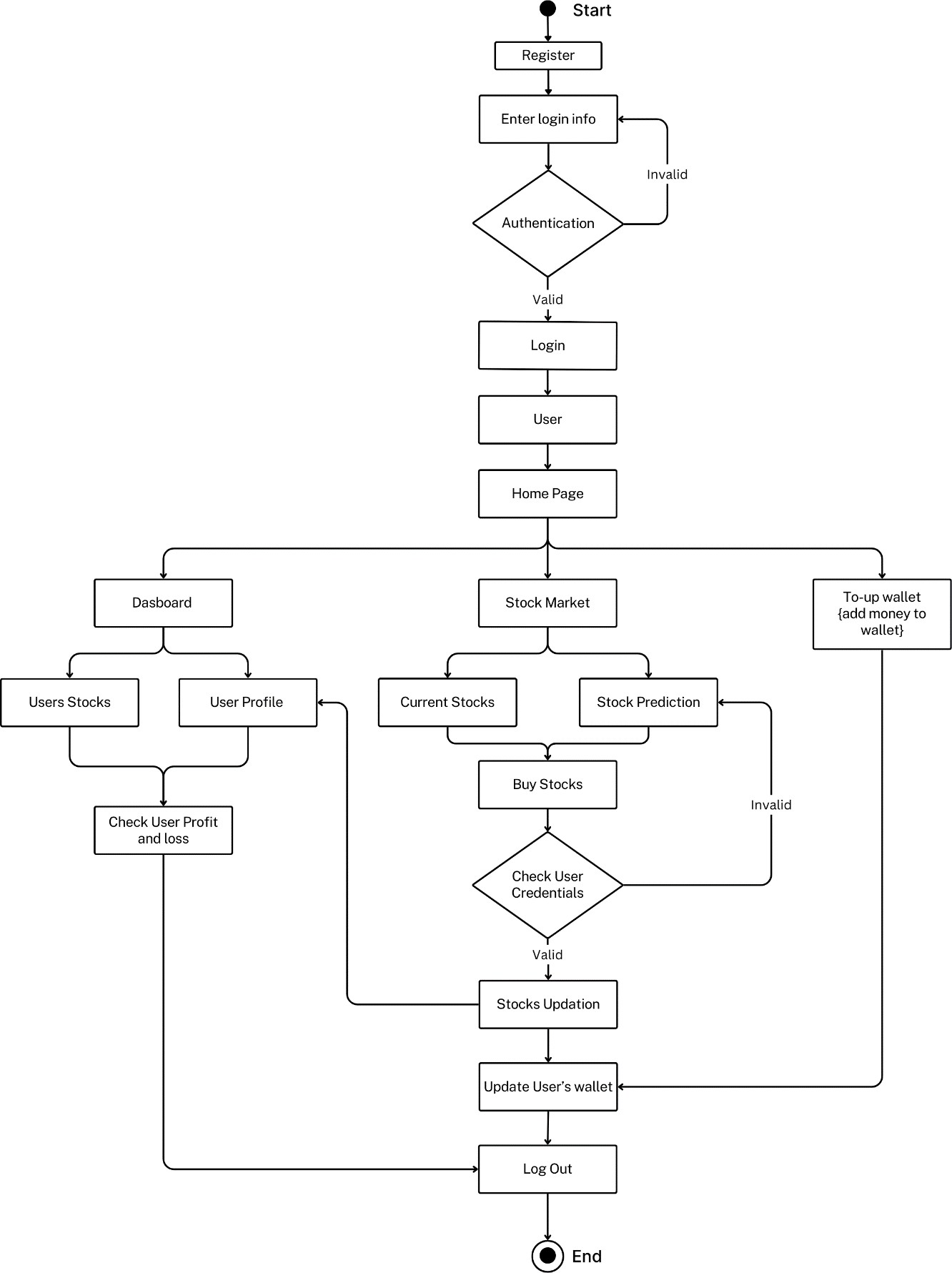


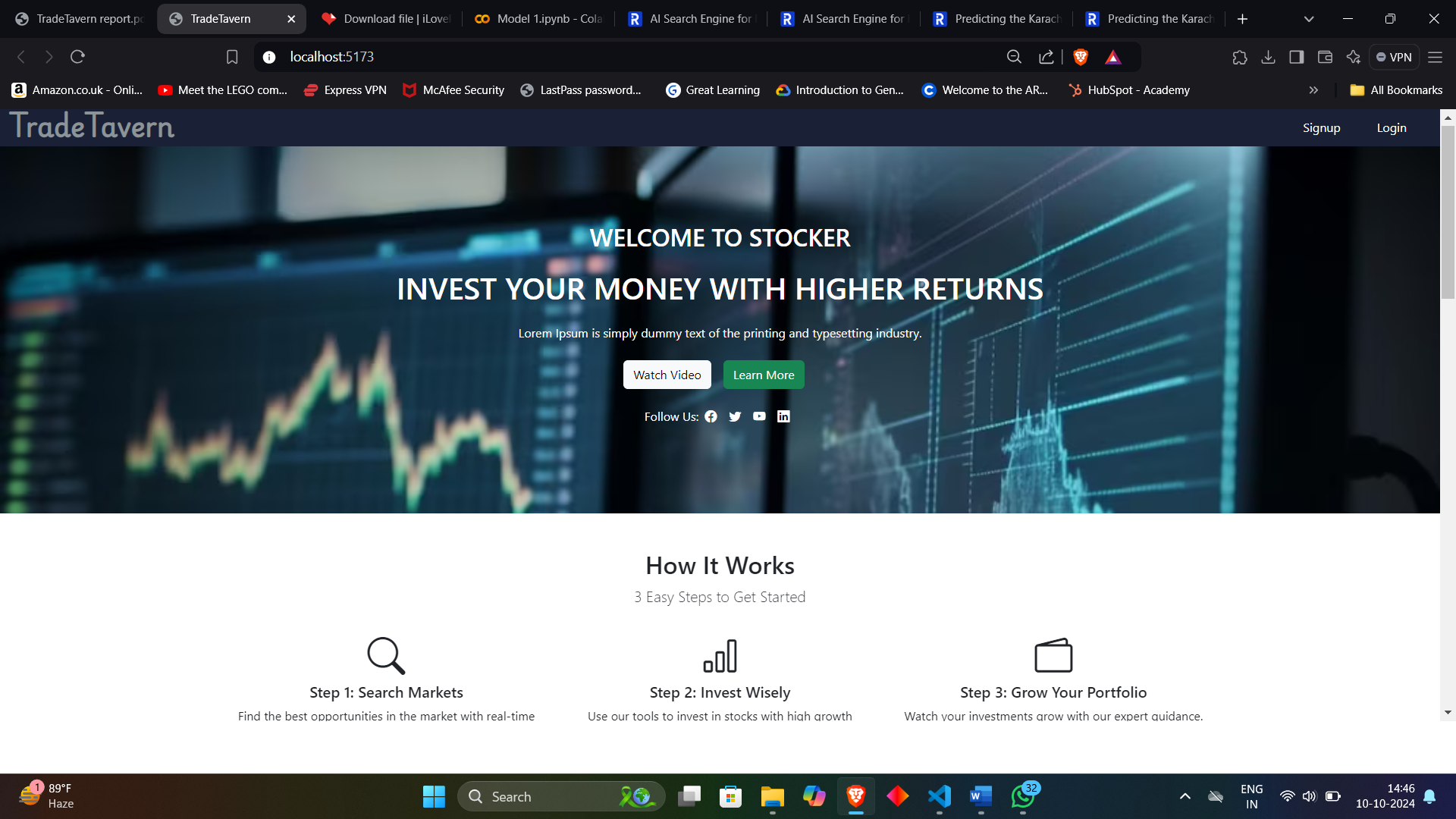
Fig. 3.1 Activity Diagram

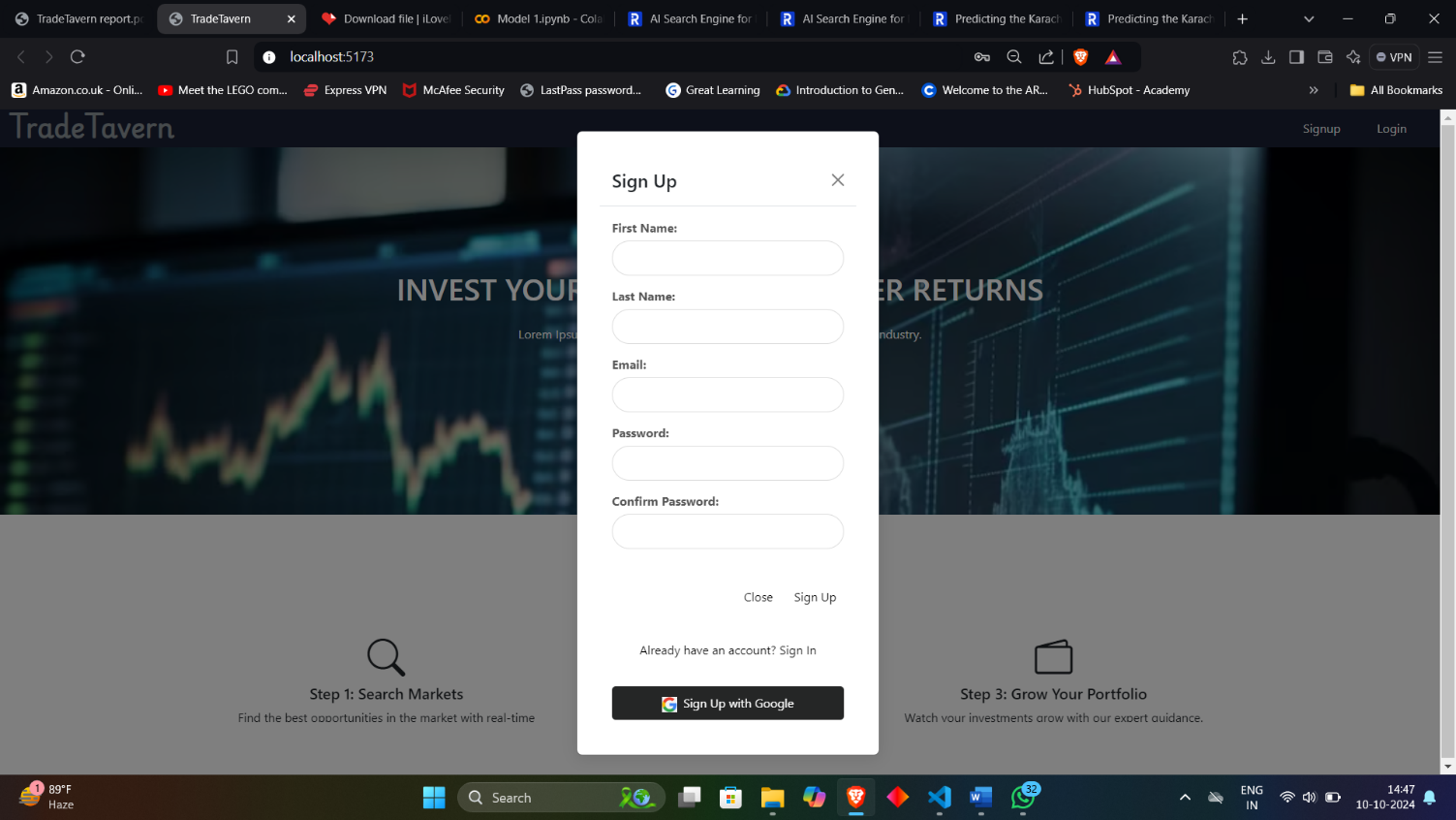
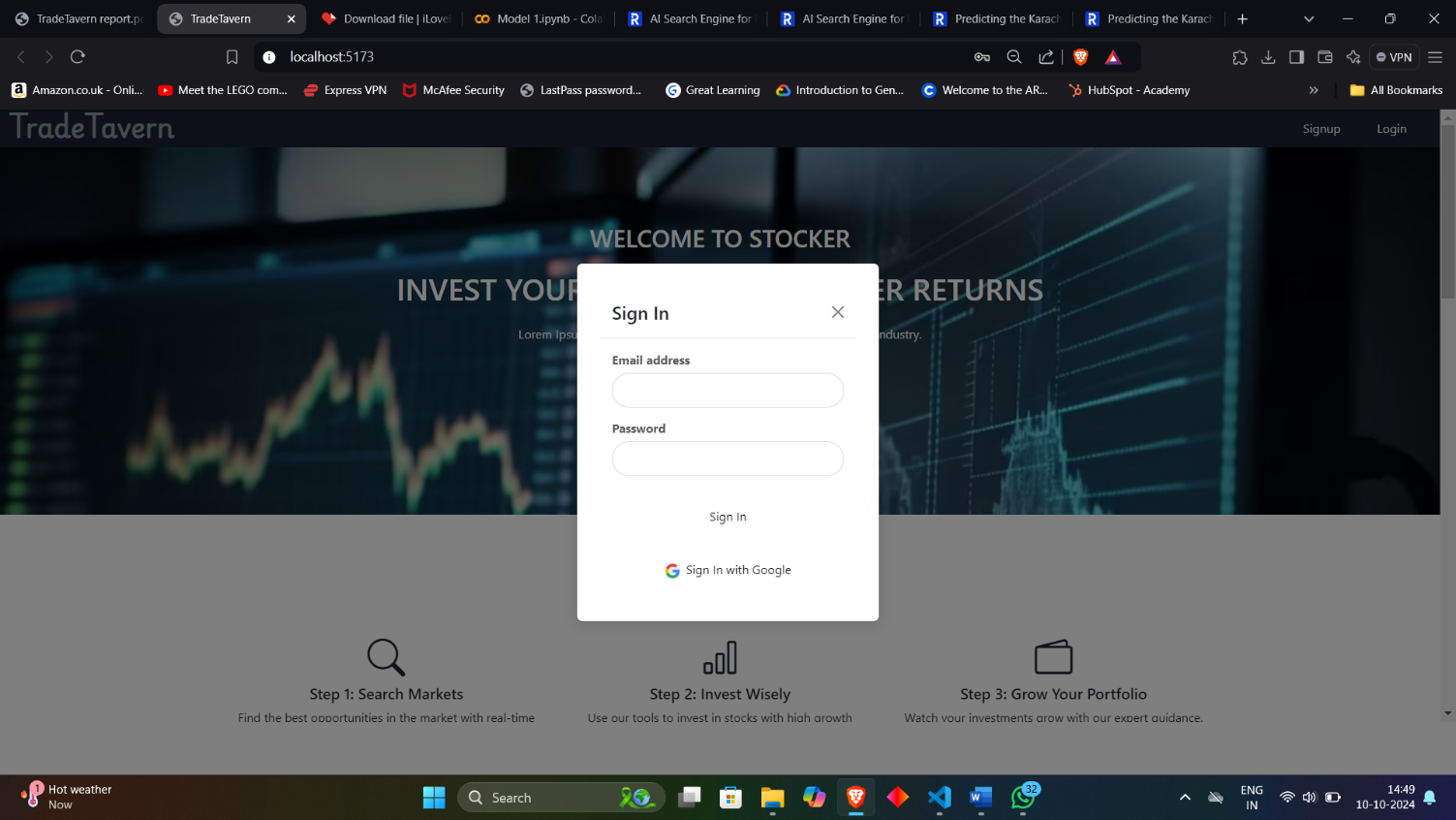
##### Methodology/Procedure

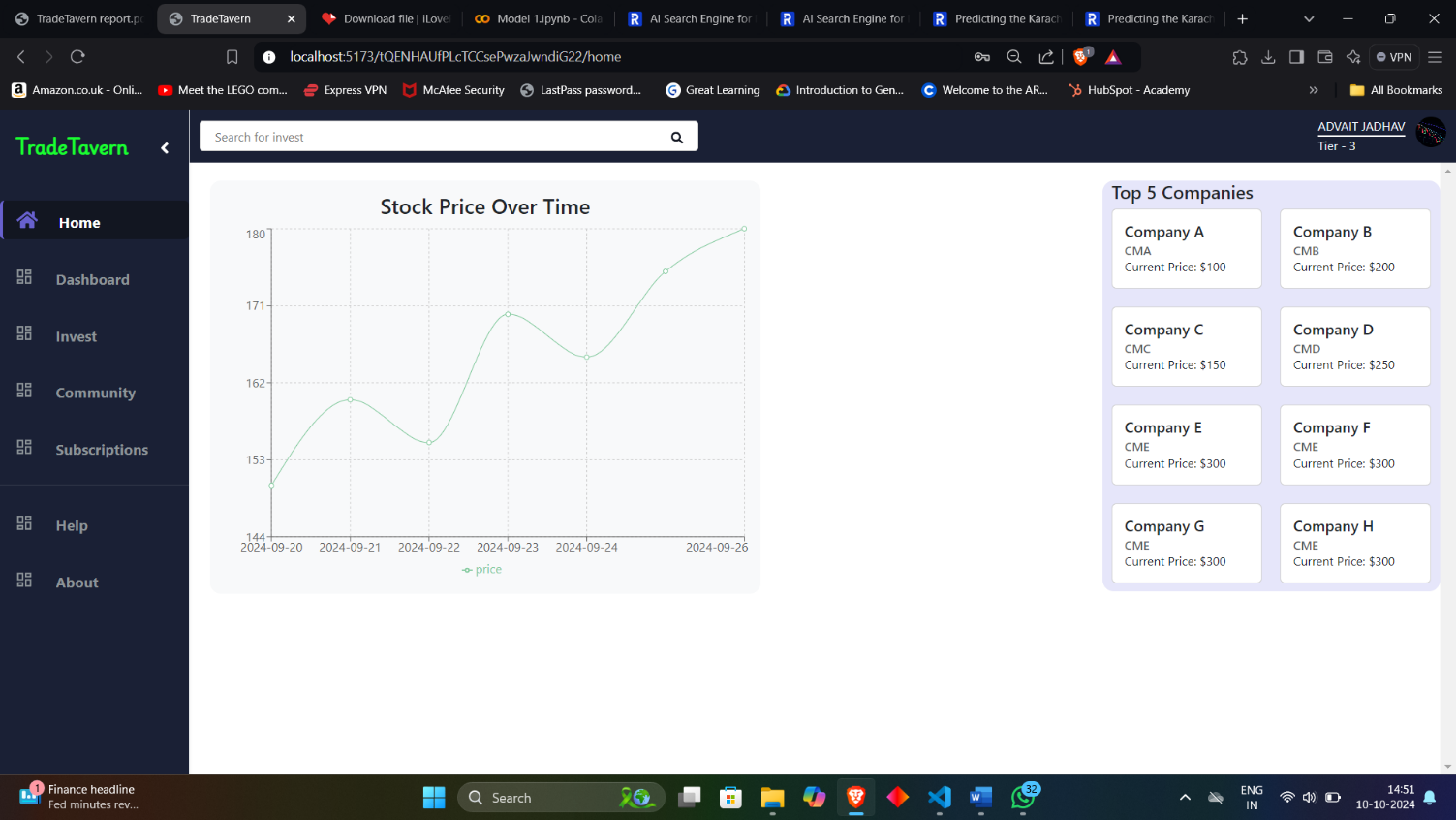
1. **Data Collection:** Historical stock data is gathered from various sources to create a dataset to run the LSTM algorithm on.
2. **Data Preprocessing:**
   1. **Data Cleaning:** The raw data is cleaned to remove any inconsistencies, missing values, or outliers that could affect the model's performance.
   2. **Scaling:** The cleaned data is normalized to ensure that all features are on a similar scale, which is crucial for the effective training of the LSTM model.
   3. **Transformation:** The data is transformed into a format suitable for input into the LSTM model, involving the creation of sequences that represent the temporal dependencies of stock prices.
3. **Data Splitting:** The preprocessed data is split into training and testing sets. The training set is used to teach the model, while the test set is reserved for evaluating the model's accuracy. This splitting will be **80%** of data on training set and **20%** in testing set.
4. **Model Building & Training:** The LSTM model is constructed with multiple layers, where each layer captures different aspects of the temporal patterns in the stock data. The model is then trained on the training set, learning to predict future stock prices based on historical trends.
5. **Prediction & Evaluation:** After training, the model is used to make predictions on the test set. These predictions are then evaluated against actual stock prices to assess the model's accuracy and performance. The evaluation metrics help in refining the model for better accuracy.
6. **Optimization:**
   1. **HyperParameter Tuning:** Hyperparameters are configuration settings used to control the learning process of a model. These include the learning rate, batch size, and the number of layers or units in a network(Hidden Layers). They are not learned during training, but has to be tuned manually or through automated.
   2. **Early Stopping:** It is a regularization technique used to prevent overfitting during training. It involves monitoring the model's performance on a validation set and halting training once the validation loss stops improving for a certain number of epochs, ensuring the model does not learn noise from the training data.
   3. **Adjusting Sliding Window:** It is the number of time steps in the input sequences. A smaller window size can reduce computational complexity, while a larger window may capture more long-term dependencies

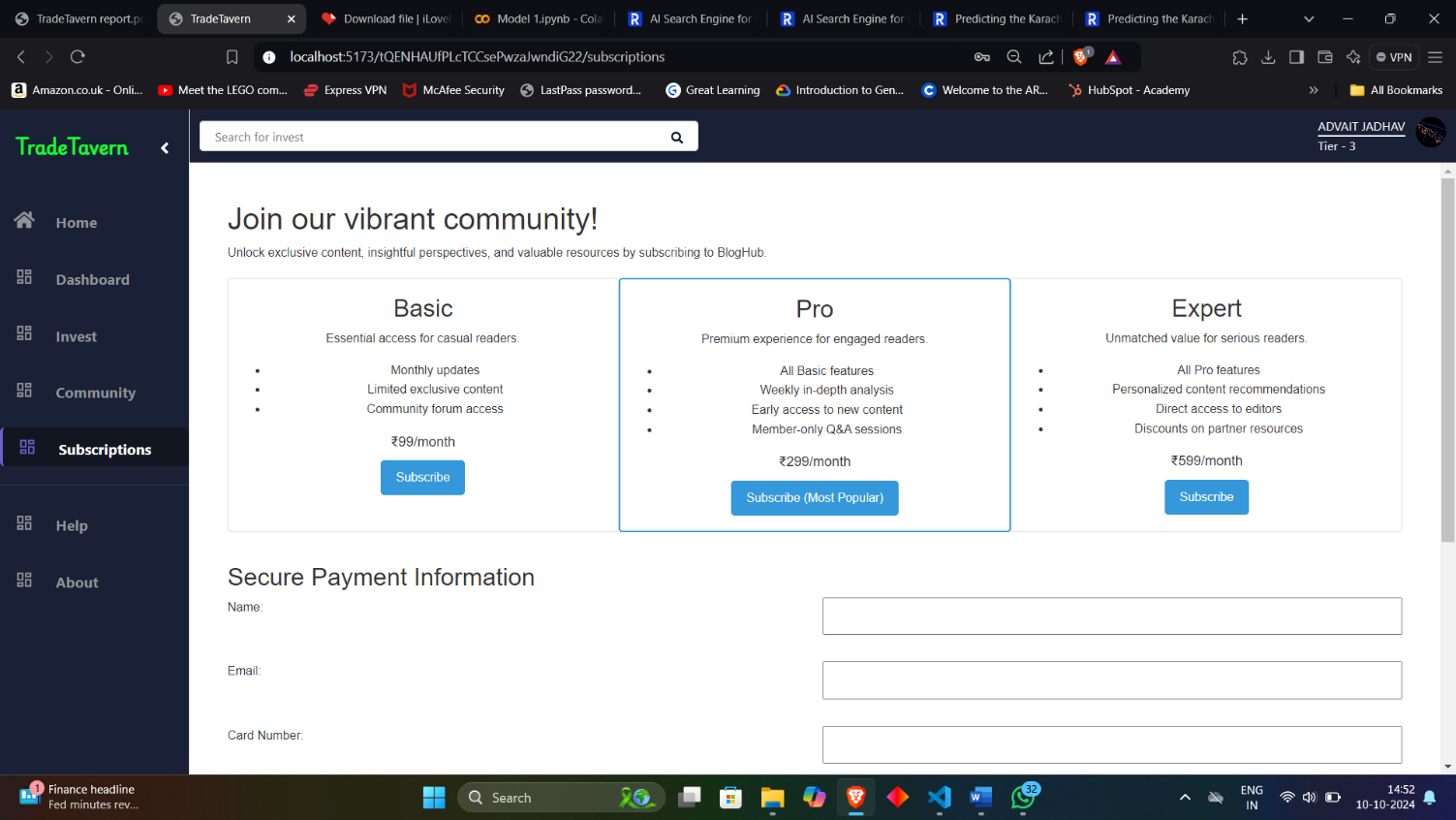
**CHAPTER 4**

**4.1 Result**

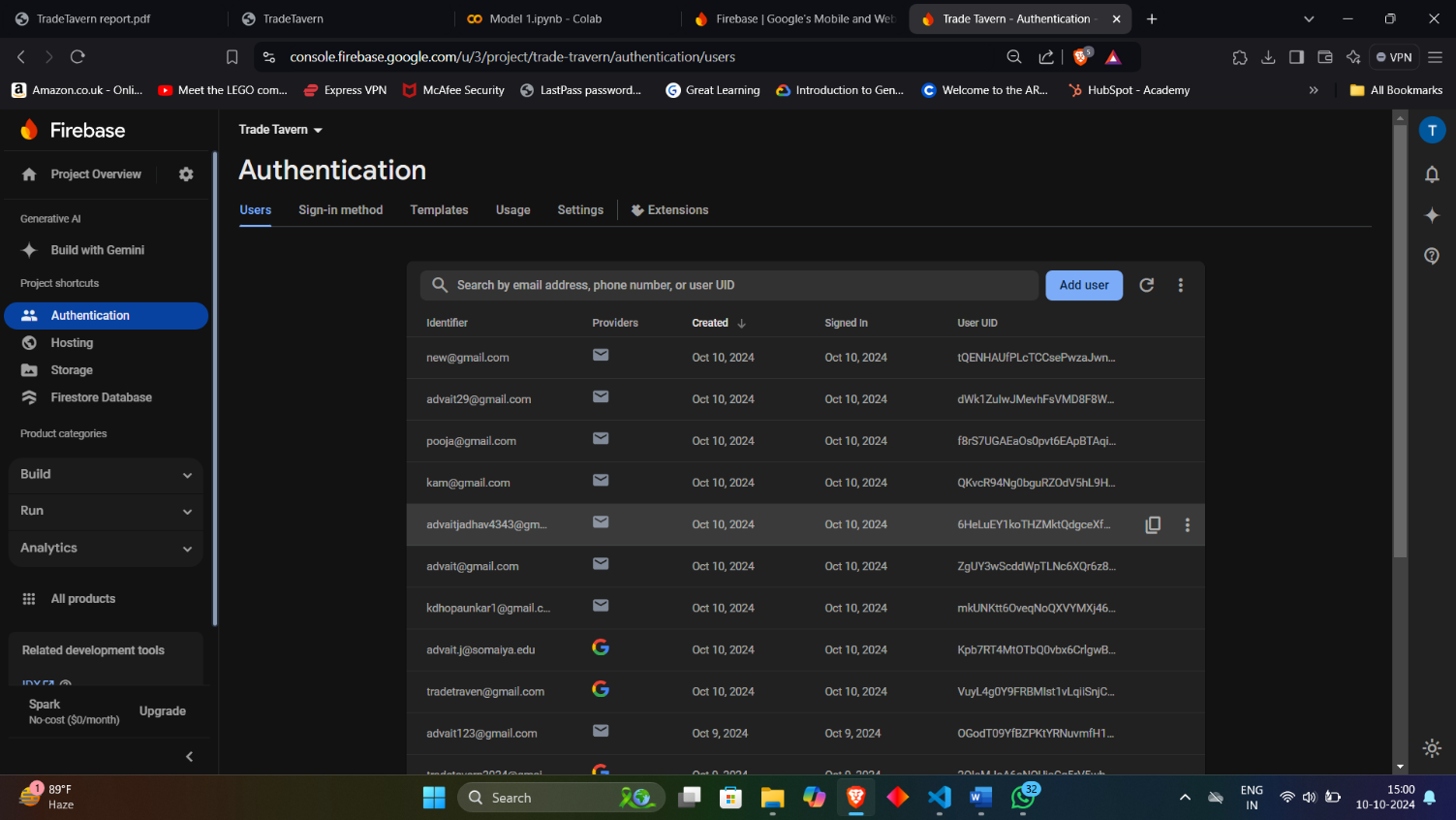
 **4.1.1 Landing Page:**

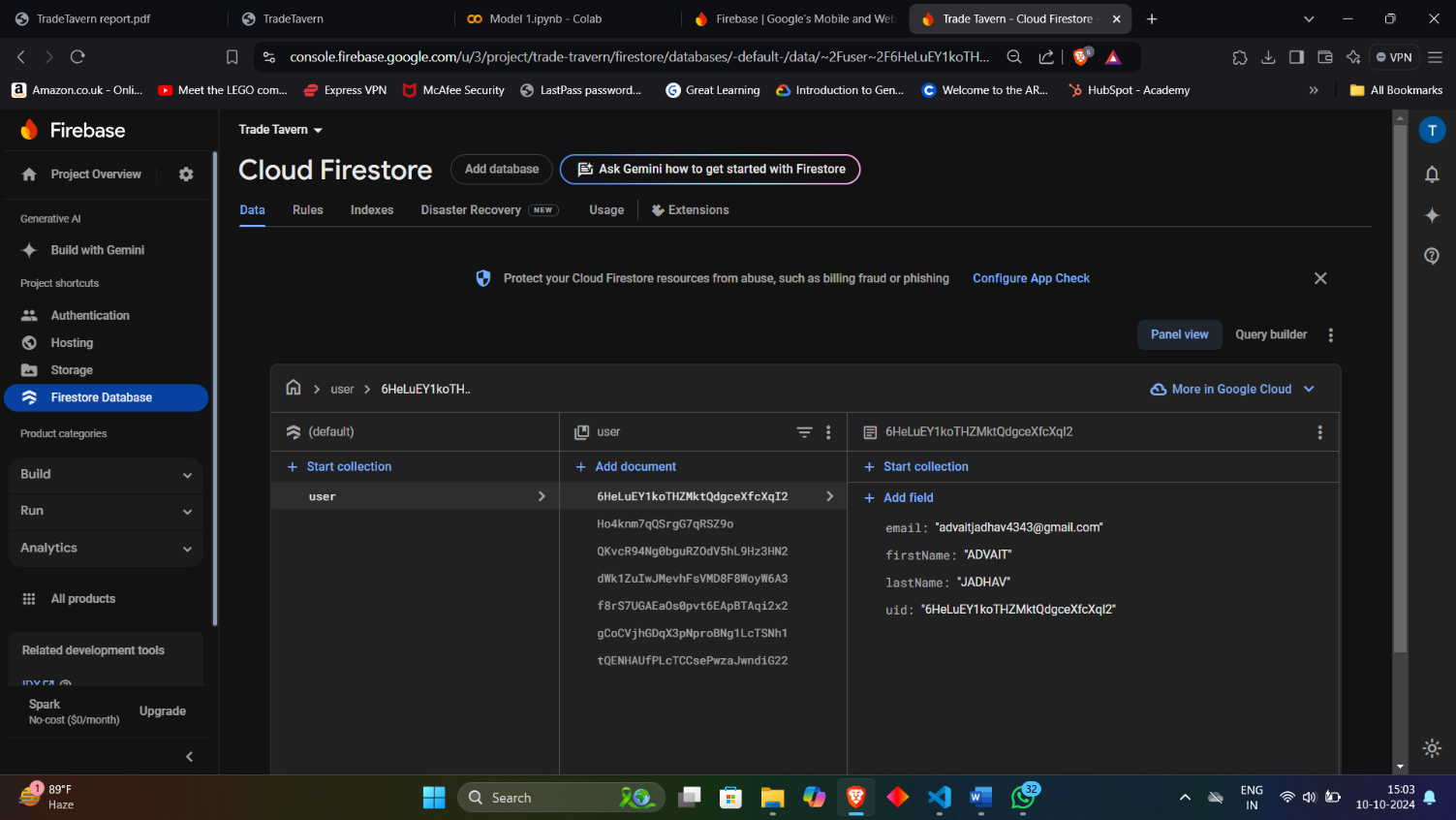
**4.1.1 Signup page :****4.1.3 login page:**

**4.1.4 Home page:**

**4.1.5 Subscription page :**

**4.1.6 Database :**

 **1. Authentication :**

1. **Users data ;**

**CHAPTER 5**

**REFERENCES**

1. Malti Bansal, Apoorva Goyal, Apoorva Choudhary,” Stock Market Prediction with High Accuracy using Machine Learning Techniques” , Procedia Computer Science 215 247–265, 2022.
2. V. Kranthi Sai Reddy,” Stock Market Prediction Using Machine Learning”, International Research Journal of Engineering and Technology (IRJET), October 2018.
3. Nirbhey Singh Pahwa, Neeha Khalfay, Vidhi Soni, Deepali Vora,” Stock Prediction using Machine Learning: A Review Paper”, International Journal of Computer Applications(0975 – 8887), April 2017
4. Sentiment Analysis: https://youtu.be/EeoCcjPuJwE?si=ddtsdu9p65dZ9Tok
5. Pandas: https://youtu.be/2uvysYbKdjM?si=2GgZ3S0Uh9wN-A\_G
6. React: https://youtu.be/CgkZ7MvWUAA?si=bgr0uUD05sNH6dt8
7. Flaskhttps://youtube.com/playlist?list=PLzMcBGfZo4n4vJJybUVV3Un\_NFS5EOgX&si=agss2I3UrrIgBHTT
8. MathplotLib: <https://youtu.be/OZOOLe2imFo?si=izAPtS4RedkwACs5>
9. Forntend Theme: <https://themewagon.github.io/stocker/>
10. Article:<https://www.sciencedirect.com/science/article/pii/S1877050920307924>
11. Article:<https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1435&context=jitim>
12. Survey:<https://www.researchgate.net/publication/341482418_A_Survey_on_Stock_Market_Prediction_Using_Machine_Learning_Techniques>
13. Mdpi :<https://www.mdpi.com/2079-9292/10/21/2717>
14. Zerodha :<https://zerodha.com/>
15. Grow :<https://groww.in/>
16. Stock analysis and screening tool : <https://www.screener.in/>
17. Tickertape :<https://www.tickertape.in/>
18. Investing.com :<https://in.investing.com/equities/indiaindia>
19. Finology :<https://ticker.finology.in>
20. Techical analysis tool :<https://www.angelone.in/knowledge-center/share-market/technical-analysis-tools-for-the-indian-stock-market>

#### Acknowledgement

We wish to express our sincere gratitude to **Dr. Sanjay U. Bokade, Principal** and **Prof.S. P. Khachane , H.O.D.** of Department Computer Engineering of Rajiv Gandhi Institute of Technology for providing us an opportunity to do our project work on **“(Trade Travern)”**.

This project bears on imprint of many peoples. We sincerely thank our project guide Dr/Prof**. (Prof. Dilip Dalgade)** for her guidance and encouragement in carrying out this synopsis work.

Finally, we would like to thank our colleagues and friends who helped us in completing project work successfully.

* 1. Soham Mandavkar
  2. Pooja Mayekar
  3. Kamran Dhopaunkar
  4. Advait Jadhav