COINSCRYPT: CRYPTOCURRENCY PRICE TRACKER AND PREDICTOR

A Report submitted

in partial fulfilment of the requirement

for the award of Degree of

Bachelor of Technology

In

Computer Science and Engineering

by

Gurpreet Singh (2020BCSE023)

Under the Guidance of

Dr. Lavanya Madhuri Bollipo

Department of Computer Science and Engineering

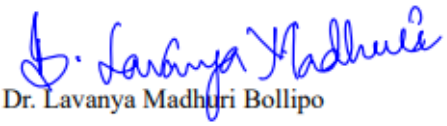
National Institute of Technology Srinagar,

Kashmir 190006, INDIA.

JUNE, 2024

CERTIFICATE

It is to certify that the contents of the report entitled "**CoinsCrypt: Cryptocurrency price tracking and price prediction**" is a bonafide work carried out by **Mr. Gurpreet Singh (2020BCSE023)** under my supervision and guidance in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. The contents of the report have not been submitted earlier for the award of any other degree or certificates and I hereby commend the work done by him in this connection.



Dr. Lavanya Madhuri Bollipo
Department of Computer Science and Engineering,
National Institute of Technology, Srinagar
Srinagar, J & K

CERTIFICATE OF APPROVAL

This project titled "**CoinsCrypt: Cryptocurrency price tracking and price prediction**" carried out by **Mr. Gurpreet Singh (2020BCSE023)** is hereby approved as the creditable study of technology in Computer Science and Engineering and is presented in a satisfactory manner. It warrants its acceptance as a prerequisite in fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at National Institute of Technology Srinagar, J&K.

Internal Examiner External Examiner

Dr. Mohammad Ahsan Chisti
Head
Department of Computer Science and Engineering
National Institute of Technology, Srinagar
Srinagar, J & K

STUDENT DECLARATION

I hereby declare that the report of the project work entitled **"CoinsCrypt: Cryptocurrency Price Tracker and Predictor App,"** is an authentic record of my own work carried out under the supervision of **Dr. Lavanya Madhuri Bollipo**, Department of Computer Science and Engineering, National Institute of Technology, Srinagar. The matter embodied in this project, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma. I also declare that the work submitted by me is entirely original, free from plagiarism, and has been diligently checked through Turnitin software to ensure its authenticity.

Gurpreet Singh
2020BCSE023

Srinagar, 190006
20 June 2024

ACKNOWLEDGEMENT

I would like to thank my supervisor **Dr. Lavanya Madhuri Bollipo**, National Institute of Technology Srinagar, whose valuable guidance has been the ones that helped me progress so far in this project. Her instruction throughout the process has been a blessing for me which provided me clarity and encouraged me to do the work. I also sincerely thank you for the time spent proofreading and correcting my many mistakes. I extend my sincere thanks to one and all of NIT Srinagar family for the completion of this document on the project report format guidelines.

Gurpreet Singh
2020BCSE023

ABSTRACT

Cryptocurrency, sometimes called crypto, is any form of currency that exists digitally or virtually and uses cryptography to secure transactions. Cryptocurrencies have no central reporting or regulatory authority; instead, a decentralized system is used to record transactions and issue new units. Cryptocurrencies operate on a public ledger called blockchain, which is a record of all new transactions and maintained by account holder.

The proposed web application CoinsCrypt is a digital currency tool which offers a set of applications such as, price tracking, historical data analysis, news aggregation, even price prediction etc. The proposed work uses React for frontend and Firebase for the backend — which makes the user experience smooth and ensures the application is responsive. The model allows users to create an account either through email or by their Google accounts: which makes the interaction more secure and personalized. With the help of CoinGecko API, real-time information on more than 100 cryptocurrencies is now available through CoinsCrypt — that can assist users in monitoring market trends effectively.

The proposed application CoinsCrypt uses machine learning algorithms such as LSTM to forecast future cryptocurrency prices. The proposed work uses data sourced from yahoo finance. The R2 score for the test data is 0.98 and for the train data is 0.99. The results demonstrates that the prediction loss of algorithm is very low and achieved acceptable prediction accuracy. Thus, the proposed tool CoinsCrypt can provide a better interface for digital currency transactions.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

In the age of evolving financial markets, cryptocurrencies represent an immensely important new asset class [2]. Bitcoin, Ethereum and hundreds of other cryptocurrencies have revolutionized the landscape for wealth creation and financial inclusion, while trade in crypto assets is evolving to approach its own natural equilibrium, shaped by the intrinsic volatility of cryptocurrencies and the very new nature of current market activity, which needs powerful tools for tracking, analysis, and prediction [3]. Cryptocurrency prices have never been more valuable for investors looking to understand this complex environment [2].

Given the peculiarities of the cryptocurrency market – 24/7 trading, high volatility, and multifactorial influences, arising from technological dynamism, regulatory changes, market sentiment, and macroeconomic trends – how do we apply traditional financial tools and applications to this new asset class [11]? A variety of applications have sprung up to offer data-tracking, analysis, and predictions specific to the cryptocurrency market [12].

The prediction of cryptocurrency prices is crucial due to the highly volatile nature of digital currencies [2]. Accurate forecasts enable investors to make informed decisions, optimizing their portfolios and mitigating risks associated with market fluctuations [11]. For traders, predictive models offer insights into potential price movements, enhancing strategic trading and maximizing profits. Additionally, reliable price predictions can stabilize markets by reducing uncertainty, fostering greater trust and participation in the cryptocurrency ecosystem [13]. Overall, these predictions are vital for the financial planning and strategic initiatives of various stakeholders in the cryptocurrency market.

1.1 Problem Statement

The cryptocurrency market is characterized by extreme volatility, a continuous trading cycle and numerous influencing factors such as technological advancements, regulatory changes and shifting market sentiments. These special attributes make it tough for those who would like to invest in the sector using critical analysis tools and for enthusiasts who want to be sure of their decisions.

One of the major challenges of cryptocurrency investors is that they do not have access to real-time tracking tools. In most cases, existing platforms fail dismally when it comes to provision of accurate up-to-the-minute cryptocurrencies prices making it hard for users to be aware of fast changing markets. This constraint prevents timely investment decision-making among investors thereby risking lost opportunities or losses in highly volatile markets.

In addition, many cryptocurrency tracking tools have inadequate historical data analysis. Expansive historical data remains important for researchers examining patterns that inform their investment strategies. Nonetheless, most of the extant platforms only provide limited history perspectives with no suitable visualizations thereby hindering in-depth analysis by users.

However, it is very important to note that even some of the currently available platforms are yet to contain personalization as one of their attributes. The tastes and strategies of investors differ, and this strategy cannot solve individual requirements properly. For users to be able to focus on things they want most and need regarding cryptocurrencies, they should create a favourite cryptocurrency watchlist that they can manage. Unfortunately, many platforms have limited personalization features which makes their investment management ineffectual.

Finally, there is a significant problem with information sources' fragmentation in terms of cryptocurrency traders. To make informed decisions, one must remain updated on relevant news and events that may impact the market. However, getting timely and comprehensive news articles from multiple sources can prove overwhelming and time-consuming at best. Many platforms do not effectively blend in appropriate news content leaving the user without critical information required for successful navigation of the market.

1.2 Objectives

The main goal of CoinsCrypt is to fill in this gap by developing a cutting-edge web application that combines current cryptocurrency price analysis, historical data evaluation and machine learning based price forecasting. For this platform to provide users with a full-fledged investment guide, it should have all these traits. The following are the project's specific objectives:

1 **Real-Time Price Tracking**: To provide users with up-to-the-minute data on the prices of over 100 cryptocurrencies, ensuring they have access to the latest market information.

2 **Historical Data Analysis**: To offer detailed historical data and visualization tools, enabling users to analyse past market trends and patterns.

3 **Personalized Watchlist**: To allow users to create and manage a personalized watchlist of their favourite cryptocurrencies, helping them keep track of assets they are most interested in.

4 **News Aggregation**: To integrate relevant news articles from reputable sources, keeping users informed about the latest developments and trends in the cryptocurrency market.

5 **Price Prediction**: To utilize advanced machine learning algorithms, specifically Long Short-Term Memory (LSTM) networks, to predict future prices of cryptocurrencies, providing users with valuable foresight into market movements.

1.3 Scope of the Project

CoinsCrypt is designed to cater to both novice and experienced cryptocurrency enthusiasts by offering a user-friendly interface combined with powerful analytical tools. The project encompasses the following key features:

1. **User Authentication**: Secure user authentication through email and Google sign-in options, allowing for personalized user experiences and data management.
2. **Real-Time Data Visualization**: Interactive charts and graphs powered by Chart.js to visualize real-time and historical price data.
3. **Watchlist Management**: A feature that lets users add or remove cryptocurrencies from their watchlist, stored in a Firebase Realtime Database.

4. **News Integration**: Aggregation of news articles using News API to provide users with up-to-date information on market trends and events.
5. **Machine Learning Predictions**: Implementation of an LSTM model to predict the next 30 days' prices of selected cryptocurrencies, displayed through an interactive Streamlit interface.

1.4 Relevance and Importance

The cryptocurrency market's volatility presents both opportunities and risks for investors. Accurate and timely information can significantly influence investment decisions and outcomes. CoinsCrypt aims to empower users by providing a platform that combines real-time data, historical insights, and predictive analytics, thereby enhancing their ability to navigate the market's complexities. The integration of machine learning for price prediction is particularly relevant, as it leverages advanced algorithms to forecast market trends, offering users a potential edge in their investment strategies.

1.5 Technical Challenges

Developing CoinsCrypt involves addressing several technical challenges, including:

1. **Data Integration**: Aggregating and synchronizing data from multiple sources, such as the CoinGecko API for cryptocurrency prices and News API for news articles, requires robust data handling and processing capabilities.
2. **Real-Time Updates**: Ensuring real-time data updates and maintaining the accuracy and responsiveness of the user interface.
3. **Scalability**: Designing a system that can handle a growing number of users and an expanding dataset of cryptocurrencies and historical price points.
4. **Machine Learning**: Implementing and fine-tuning the LSTM model to achieve accurate and reliable price predictions, along with integrating these predictions into the web interface seamlessly.
5. **User Experience**: Creating an intuitive and user-friendly interface that caters to both novice and experienced users, with clear visualizations and easy navigation.

1.6 Project Methodology

The development of CoinsCrypt follows an iterative approach, incorporating agile principles to ensure continuous improvement and adaptation. The project is divided into distinct phases, including:

1. **Requirement Analysis**: Defining the functional and non-functional requirements of the application based on user needs and market research.
2. **System Design**: Creating detailed architectural and design diagrams to outline the system's components and their interactions.
3. **Implementation**: Developing the application using modern web technologies such as React, Firebase, and Python, along with integrating APIs and machine learning models.
4. **Testing**: Conducting thorough testing to ensure the functionality, performance, and security of the application.
5. **Deployment**: Deploying the application on a cloud platform to ensure accessibility and scalability.
6. **Maintenance and Updates**: Providing ongoing support and updates to address user feedback and incorporate new features.

1.7 Summary

The chapter provides a comprehensive overview of the CoinsCrypt project, outlining the motivations behind its development and the problems it aims to solve in the volatile cryptocurrency market. The chapter details the primary objectives, which include providing real-time price tracking, historical data analysis, news integration, and future price predictions using machine learning. The scope of the project is defined, emphasizing the integration of multiple technologies such as React, Firebase, Chart.js, and LSTM networks to create a robust and user-friendly web application. This foundational chapter sets the stage for the detailed exploration of the system design and implementation in the subsequent chapters.

CHAPTER 2

LITERATURE SURVEY

S. No.	Source	References	Observation
1	Bitcoin: A Peer-to-Peer Electronic Cash System" by Satoshi Nakamoto	[1]	This seminal paper by the pseudonymous creator of Bitcoin, Satoshi Nakamoto, introduced the concept of a decentralized digital currency that operates without a central authority. The paper outlines the underlying technology, blockchain, which enables secure and transparent transactions. This foundational work has paved the way for the development of numerous cryptocurrencies and blockchain applications, forming the basis for the CoinsCrypt project's focus on cryptocurrency price tracking and prediction.
2	Cryptocurrency: How Bitcoin and Digital Money are Challenging the Global Economic Order" by Paul Vigna and Michael J. Casey	[2]	This book explores the broader implications of cryptocurrencies on the global economic system. Vigna and Casey discuss the disruptive potential of digital currencies and their ability to provide financial services to the unbanked. This contextual understanding of the cryptocurrency market helps in appreciating the significance of tools like CoinsCrypt, which aim to make cryptocurrency trading more accessible and informed.
3	Bitcoin and Cryptocurrency Technologies" by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder	[3]	This academic text offers a thorough examination of the technical aspects of Bitcoin and other cryptocurrencies. It covers cryptographic foundations, blockchain technology, and security issues. The technical depth provided in this book is critical for understanding the mechanisms behind cryptocurrency price movements and for developing robust predictive models in CoinsCrypt.

2.1 Summary

The literature survey for the CoinsCrypt project highlights the foundational principles and broader implications of cryptocurrencies, focusing on the technology and economic impact essential for the project's objectives. The survey underscores the importance of blockchain technology for secure, decentralized transactions, which is critical for accurate cryptocurrency price tracking and prediction. It also emphasizes the disruptive potential of digital currencies in providing financial services and making trading more accessible, aligning with CoinsCrypt's goal to democratize the cryptocurrency market. Additionally, the technical insights into cryptographic foundations and security mechanisms are crucial for developing robust predictive models within the application. These conclusions collectively inform the development and focus of the CoinsCrypt project.

CHAPTER 3
METHODOLOGY

3.1 Introduction

This chapter details the development process of the CoinsCrypt web application, focusing on the integration of frontend and backend technologies. The primary technologies used include React for the frontend, Firebase for the backend, and various APIs for fetching cryptocurrency data and news articles. The chapter will cover the project setup, the implementation of core functionalities, and guidelines on where to place project images for optimal organization and performance.

3.2 System Design

3.2.1 Architecture Diagram

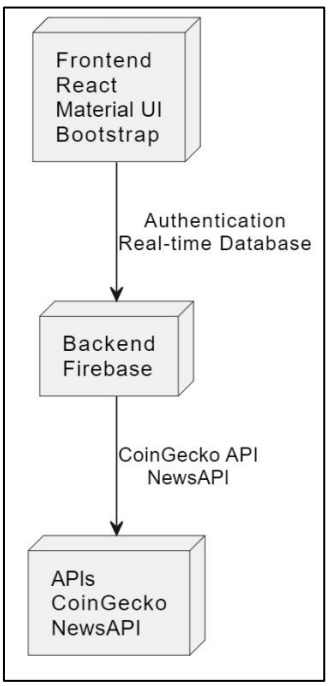


Figure 3.1 Architecture Diagram

The architecture of CoinsCrypt integrates several components to create a seamless user experience:

- **Frontend**: Built with React, styled using Material UI and Bootstrap.
- **Backend**: Managed with Firebase for authentication and real-time database services.
- **APIs**: CoinGecko API for cryptocurrency data, and News API for fetching relevant news articles.

3.2.2 Use Case Diagrams

- **Use Case: User Authentication**
  - **Actors**: User
  - **Description**: The user can sign up, log in, and log out using email/password or Google authentication.
  - **Flow**:
    1. User accesses the sign-up/login page.
    2. User enters credentials or uses Google authentication.
    3. System authenticates the user via Firebase.
    4. User is redirected to the dashboard upon successful authentication.

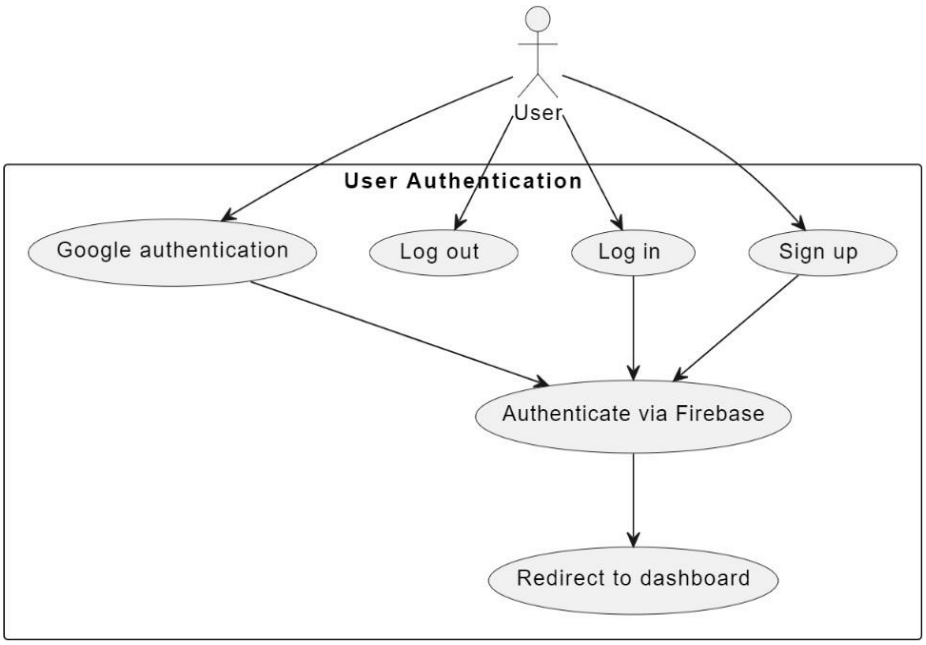


Figure 3.2 User Authentication Use Case

- **Use Case: Cryptocurrency Price Tracking**

  - **Actors**: User
  - **Description**: The user can view real-time and historical cryptocurrency prices.
  - **Flow**:
    1. User accesses the price tracking page.
    2. System fetches data from the CoinGecko API.
    3. Data is displayed in chart format using Chart.js.

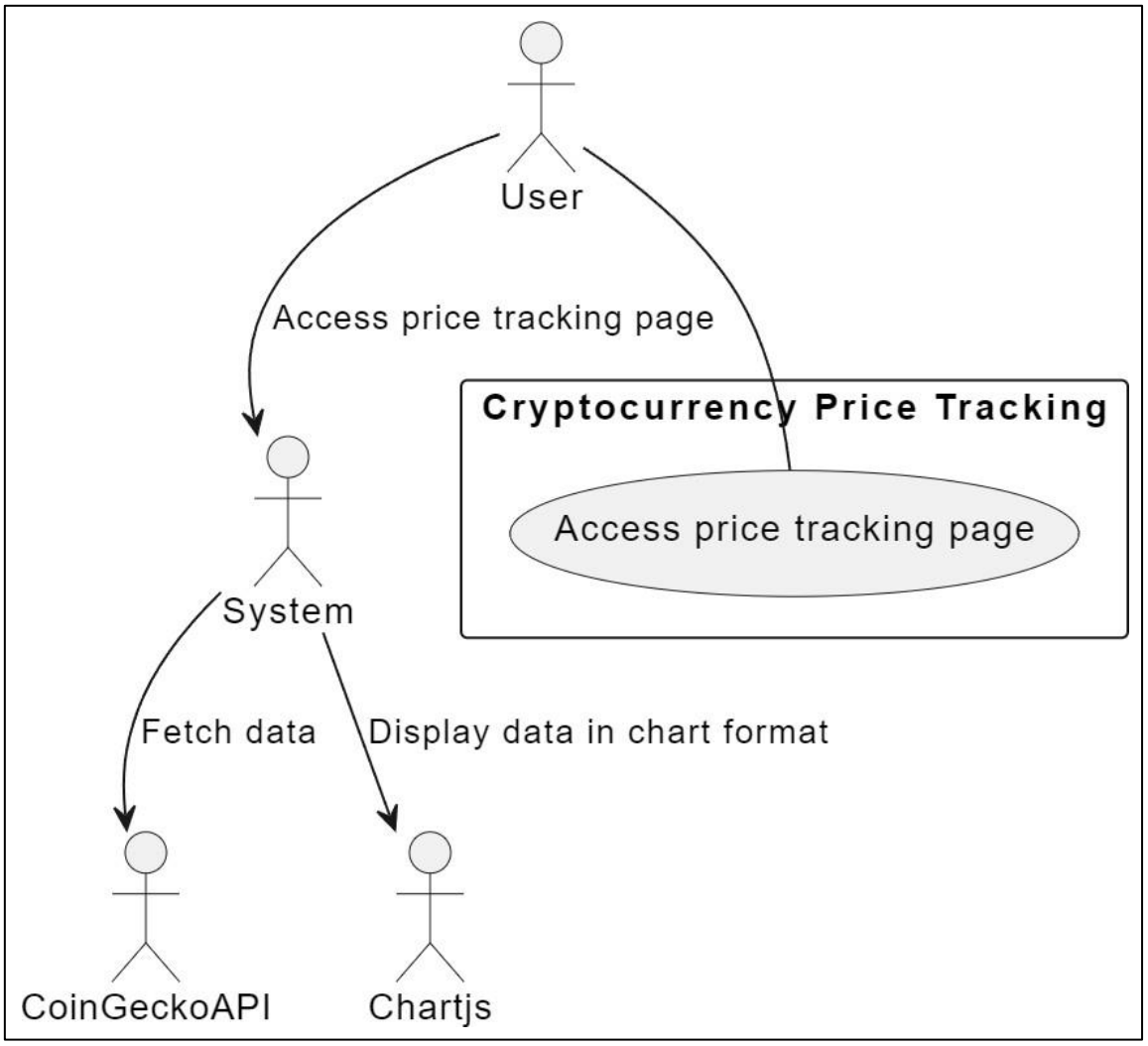


Figure 3.3 Price Tracking Use Case

- **Use Case: Watchlist Management**

  - **Actors**: User
  - **Description**: The user can add or remove cryptocurrencies from their watchlist.
  - **Flow**:
    1. User adds/removes a cryptocurrency to/from the watchlist.
    2. System updates the user's watchlist in Firebase Realtime Database.

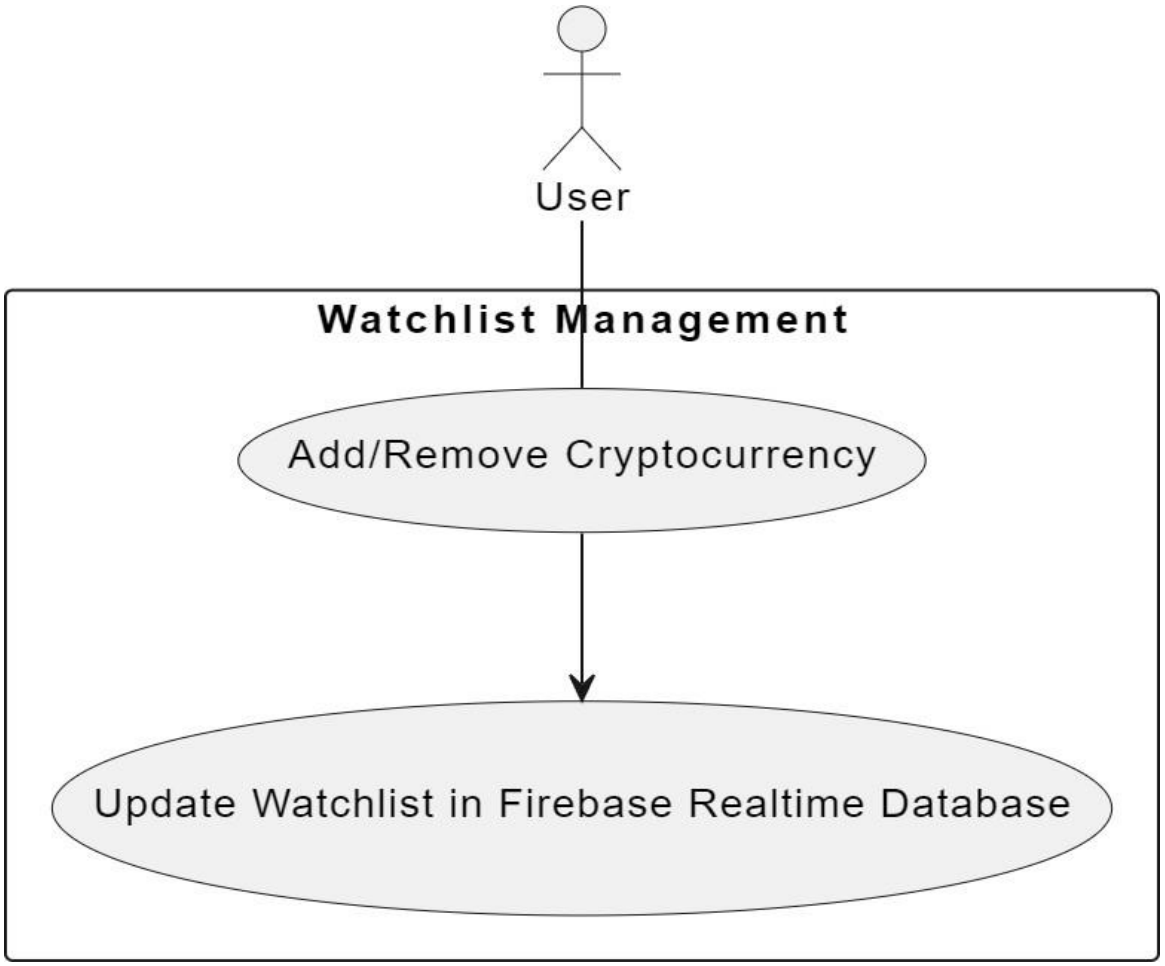


Figure 3.4 Watchlist Use Case

- **Use Case: News Integration**
  - **Actors**: User
  - **Description**: The user can view the latest news articles related to cryptocurrencies.
  - **Flow**:
    1. User accesses the news page.
    2. System fetches news articles from News API.
    3. Articles are displayed in a list format.

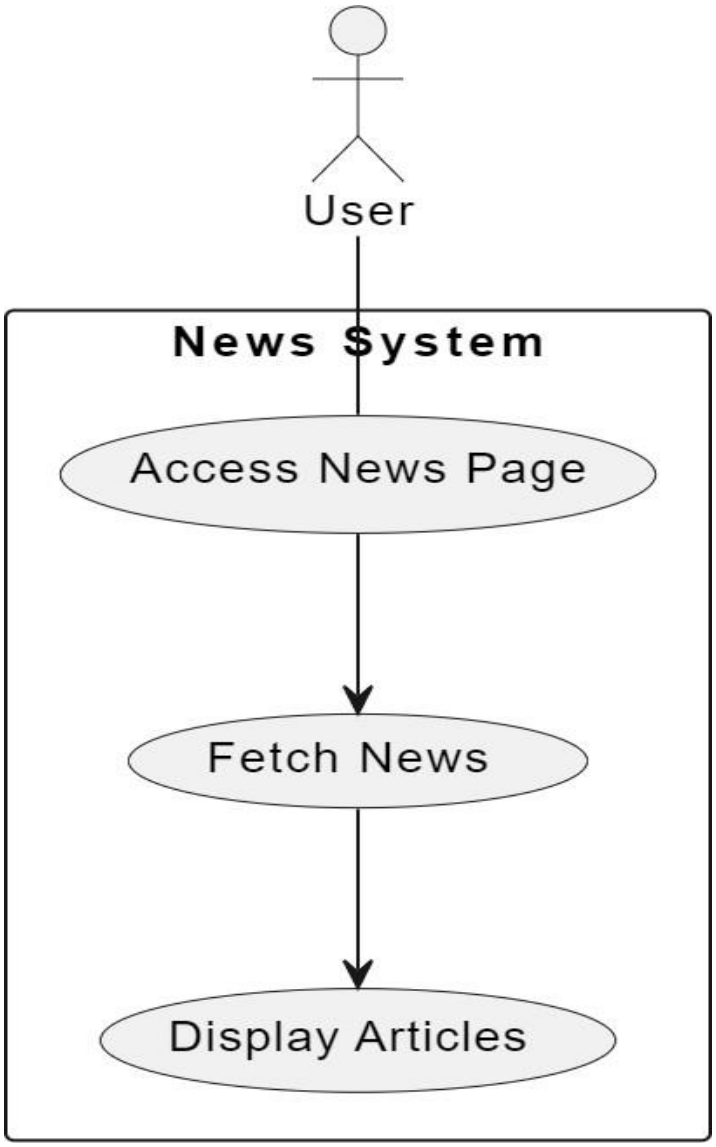


Figure 3.5 News Use Case Diagram

3.3 Technology Stack

3.3.1 Frontend

- **React:**
  React is a JavaScript library for creating user interfaces (UI) on the web. React is a de clarative component library that allows developers to create reusable UI components. It follows the virtual DOM (Data Type Model) and optimizes performance by reducing DOM updates. React is fast and works well with other tools and libraries. Here are some key points about ReactJS:

  - **Component-Based Architecture**: React applications are built using components, which are reusable, self-contained pieces of UI. Each component manages its own state and renders UI based on that state.

  - **Virtual DOM**: React uses a virtual DOM to optimize performance. Instead of directly manipulating the DOM, React creates a virtual representation of the UI and updates it in response to changes in the application's state. This approach minimizes direct DOM manipulations, resulting in better performance.

  - **JSX Syntax**: React uses JSX, a syntax extension for JavaScript that allows developers to write HTML-like code within JavaScript. JSX makes it easier to visualize the structure of the UI components.

  - **Declarative**: React adopts a declarative approach to programming. Developers describe what the UI should look like, and React takes care of updating the UI to reflect the current state of the application.

  - **Unidirectional Data Flow**: React enforces a unidirectional data flow, meaning data flows in one direction from parent to child components. This makes the application more predictable and easier to debug.

  - **Ecosystem and Community**: React has a rich ecosystem of tools, libraries, and extensions. Popular ones include Redux for state management, React Router for

routing, and many others. The community around React is large and active, providing extensive support, tutorials, and third-party libraries.

- **Server-Side Rendering**: React can be rendered on the server using frameworks like Next.js, which improves performance and SEO for web applications.

- **React Native**: React concepts can be used to build mobile applications with React Native, allowing developers to write mobile apps for iOS and Android using React.

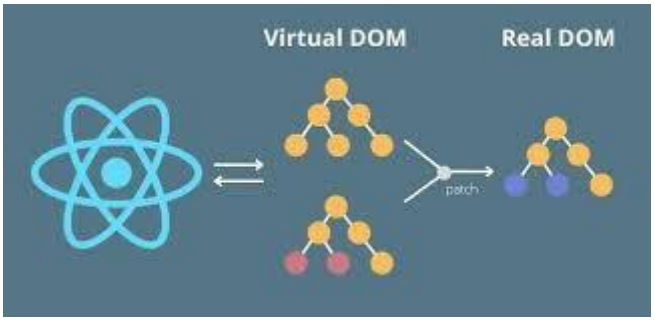


Figure 3.6 React Virtual DOM

- **Bootstrap & Material UI:**

  Libraries for responsive design and component styling.

  - **Bootstrap**:

    Bootstrap is a widely-used open-source front-end framework designed for creating responsive and mobile-first websites and web applications. Developed originally by Twitter, Bootstrap simplifies the development process with its responsive grid system, pre-styled components, and comprehensive CSS framework. The grid system allows for flexible layouts that automatically adjust to various screen sizes, ensuring a consistent user experience across desktops, tablets, and smartphones. Pre-styled components such as navigation bars, buttons, forms, modals, and carousels provide reusable building blocks that speed up development and maintain design consistency.

    In addition to its CSS capabilities, Bootstrap includes a range of JavaScript plugins that add interactive elements like tooltips, carousels, and modals

without requiring extensive custom code. The framework is highly customizable, allowing developers to override default styles with custom CSS or use Sass variables and mixins for more granular theming. This flexibility makes it easy to tailor Bootstrap to specific branding and design needs. Furthermore, Bootstrap ensures cross-browser compatibility, so web applications look and function consistently across different browsers and devices, enhancing both development efficiency and user experience.

- **Material UI**:

  Material-UI is a popular open-source library for building React applications using Google's Material Design principles. It provides a comprehensive set of pre-designed components that follow the Material Design guidelines, offering a cohesive and visually appealing user interface out of the box. Components such as buttons, forms, modals, and navigation bars are included, each customizable to meet specific design requirements. Material-UI's components are designed to be highly responsive, ensuring that applications look and function well on various devices and screen sizes.

  Beyond its rich component library, Material-UI offers powerful theming capabilities. Developers can easily customize the default Material Design theme or create their own themes using Material-UI's theme provider and style overrides. This flexibility allows for consistent branding and unique visual styles tailored to individual projects. Additionally, Material-UI integrates seamlessly with other libraries and tools in the React ecosystem, such as Redux for state management and React Router for navigation, making it a versatile choice for developing modern web applications.

  Material-UI also emphasizes ease of use and developer productivity. The library provides extensive documentation, including usage examples, customization guides, and a robust API reference, helping developers quickly get up to speed and efficiently build feature-rich applications. With its combination of pre-designed components, customization options, and integration capabilities, Material-UI is a powerful tool for creating high-quality, visually consistent, and responsive user interfaces in React applications.

- **Chart.JS:**

  Chart.js is a popular open-source JavaScript library for creating visually appealing and interactive charts on web applications. It supports a variety of chart types, including line, bar, radar, doughnut, pie, polar area, and bubble charts, providing flexibility for data visualization needs. The library is highly customizable, allowing developers to adjust colors, fonts, labels, and other aspects of the charts to match the design and functional requirements of their projects. Chart.js is easy to integrate into web applications, especially those built with frameworks like React, Angular, and Vue.

  One of the key features of Chart.js is its simplicity and ease of use. With a minimal amount of code, developers can create and manage complex charts that are responsive and adapt well to different screen sizes. The library also supports animations and tooltips, enhancing user interaction and making data insights more accessible. Additionally, Chart.js has comprehensive documentation and a supportive community, making it easier for developers to troubleshoot issues and implement advanced features.

3.3.2 Backend

- **Firebase:**

  Firebase is a comprehensive platform developed by Google for building mobile and web applications, offering a suite of tools and services such as real-time database, authentication, cloud storage, hosting, and analytics. It simplifies backend development by providing scalable infrastructure, enabling developers to focus on creating high-quality user experiences. Firebase's real-time database and Firestore allow for synchronized data across clients in real time, while its authentication service supports secure user sign-ins. Additionally, Firebase integrates seamlessly with other Google services and popular development frameworks, making it a versatile choice for developers looking to build robust, scalable applications quickly.

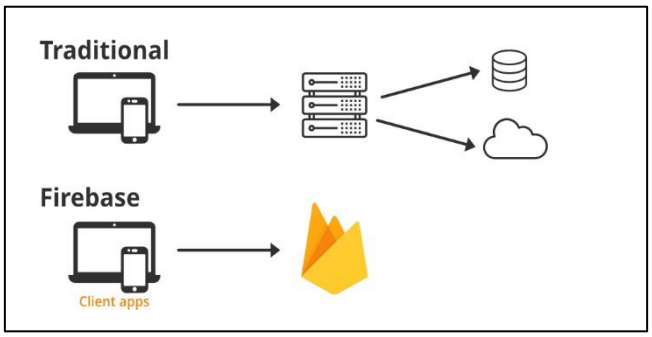


Figure 3.7 Firebase

3.3.3 APIs

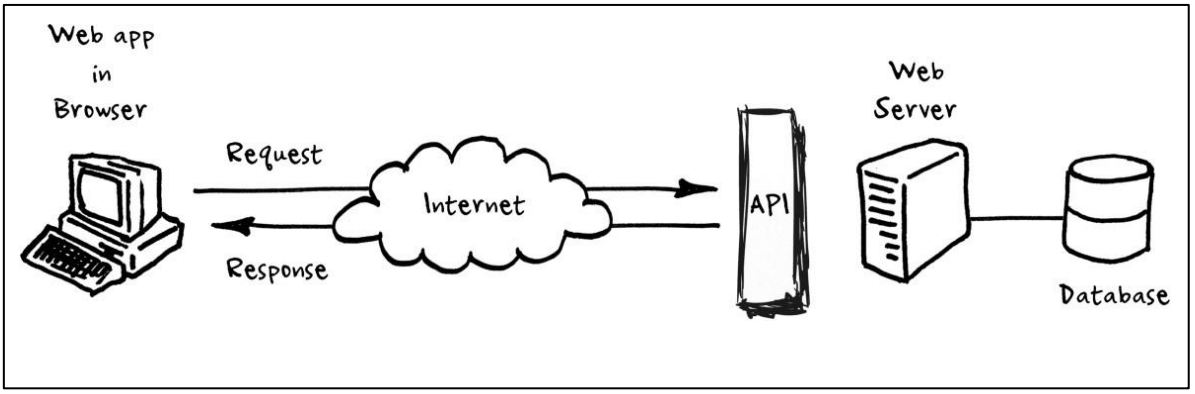


Figure 3.8 APIs Working

- **CoinGecko API:**

  The CoinGecko API is a comprehensive and user-friendly interface for accessing real-time and historical data on cryptocurrencies, including market prices, trading volumes, market capitalization, and more. It supports a wide range of endpoints that provide detailed information about thousands of cryptocurrencies, exchanges, and financial metrics. The API is widely used by developers and analysts to integrate cryptocurrency data into applications, perform market analysis, and develop trading algorithms. Its robust documentation and free access tier make it an accessible and valuable resource for anyone looking to leverage cryptocurrency data in their projects.

- **NewsData API:**

  The NewsData API is a powerful and versatile tool for accessing a vast array of news articles from various sources in real-time. It provides developers with endpoints to search for news by keywords, categories, languages, and geographical regions, offering detailed metadata such as publication dates, authors, and source credibility. The API is designed to be easy to integrate into applications, making it ideal for building news aggregation platforms, sentiment analysis tools, and media monitoring systems. Its extensive documentation and flexible query options allow for precise and efficient retrieval of news data, catering to a wide range of use cases.

3.4 Predictive Analysis

RNN stands for Recurrent Neural Network, which is a type of artificial neural network designed for sequential data processing. Unlike traditional feedforward neural networks, RNNs have connections that form a directed cycle, allowing them to exhibit temporal dynamic behavior and effectively model sequences of data.

Key features of RNNs include their ability to maintain a "memory" of previous inputs through hidden states, which enables them to process sequences of arbitrary length. This makes them particularly useful for tasks such as natural language processing (NLP), speech recognition, time series prediction, and handwriting recognition.

However, RNNs can suffer from issues like vanishing gradients, where gradients diminish as they propagate back through time, impacting their ability to capture long-term dependencies effectively. This limitation has led to the development of more advanced variants such as Long Short-Term Memory networks (LSTMs) and Gated Recurrent Units (GRUs), which alleviate the vanishing gradient problem and improve the learning and memory capabilities of RNNs.

Overall, RNNs and their variants play a crucial role in modeling sequential data and have found widespread application in various fields where understanding and predicting patterns over time are essential.

3.5 Types of RNN

RNNs generally have the characteristic of a one-to-one architecture: one input is associated with one output. However, you have the flexibility to change them to different configurations for specific purposes. Below are several types of RNNs.

- **One-to-many:** This type of RNN takes one input for multiple outputs, generating sentences from a single keyword, enabling language applications such as image captions.
- **Many-to-many:** The model uses multiple inputs to predict multiple versions. For example, you can use RNNs to create translators that analyse sentences and generate language patterns in different languages.
- **Many-to-one:** Multiple inputs are mapped to a single output. This is useful for applications such as sentiment analysis, where models can predict customer preferences such as positive, negative, and neutral based on recommendations.

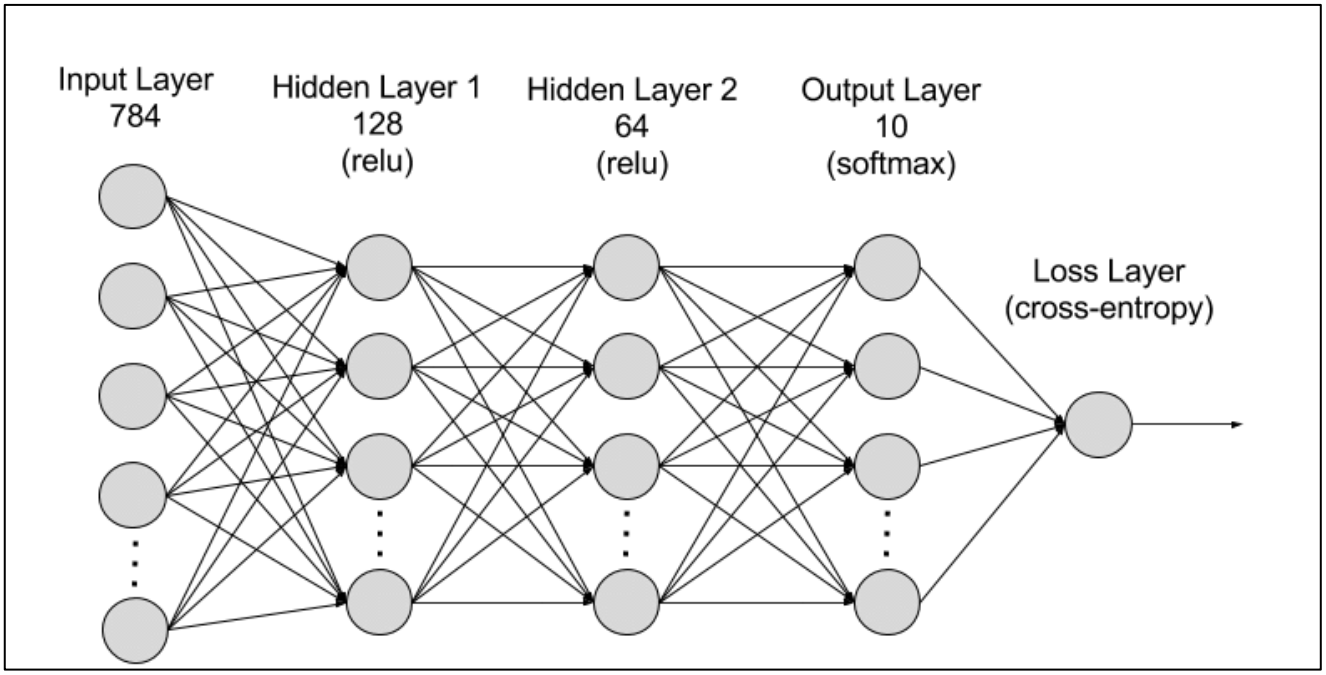


Figure 3.9 Diagram of RNN

3.6 Long Short-Term Memory

LSTM stands for Long Short-Term Memory, which is a type of recurrent neural network (RNN) architecture designed to overcome the limitations of traditional RNNs in learning and remembering long-term dependencies in sequential data. LSTMs are particularly effective in tasks where context and temporal dependencies are crucial, such as speech recognition, language modeling, machine translation, and time series forecasting.

The key innovation of LSTMs lies in their ability to maintain and control the flow of information through a cell state, which acts as a conveyor belt that can carry information across time steps. This is achieved through specialized mechanisms such as gates, including the forget gate, input gate, and output gate, which regulate the flow of information and prevent the vanishing gradient problem that commonly affects traditional RNNs.

Each LSTM unit processes input sequences step-by-step, updating its internal state based on the current input and its own previous state. This allows LSTMs to selectively remember or forget information over long durations, making them well-suited for capturing dependencies in sequential data that span across many time steps.

In practice, LSTMs have demonstrated superior performance in various applications were understanding context and capturing long-range dependencies are critical. Their architecture has inspired further developments in recurrent neural networks and continues to be a cornerstone in the field of deep learning for sequential data analysis.

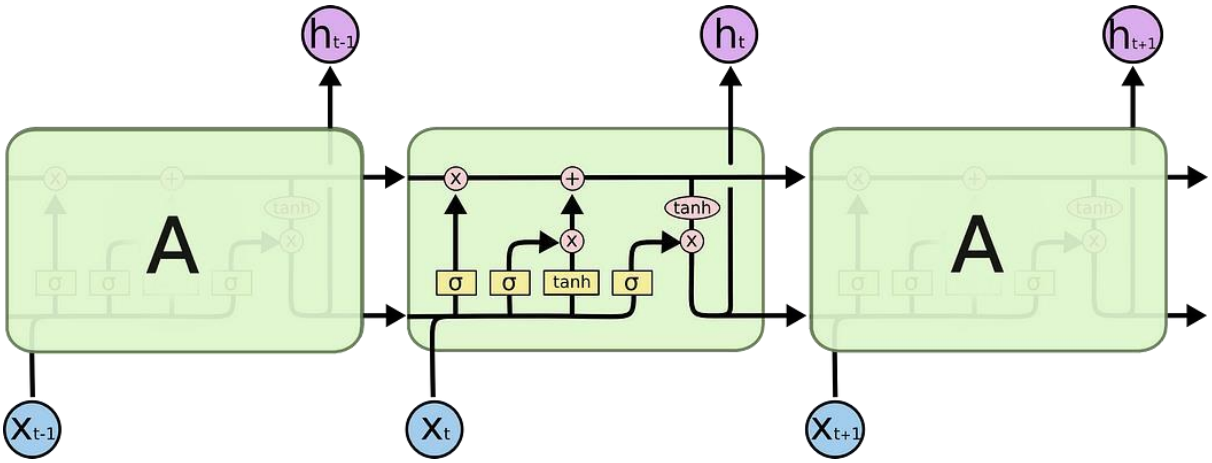


Figure 3.10 LSTM Block

3.6.1 Components of LSTM

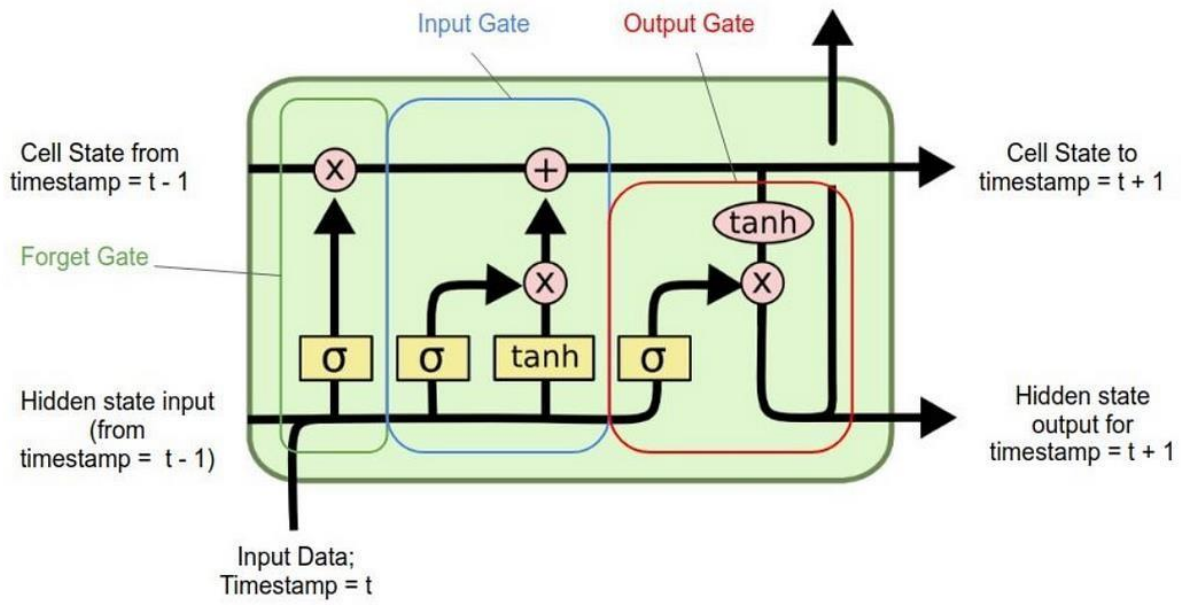


Figure 3.11 LSTM Components

- **Input Gate:** The input gate decides what new information to store in the cell state. First, a sigmoid layer, called the "input gate layer," decides which values of the candidate will be updated. Then, a tanh layer creates a vector of new candidate values, which are added to the state of the cell.

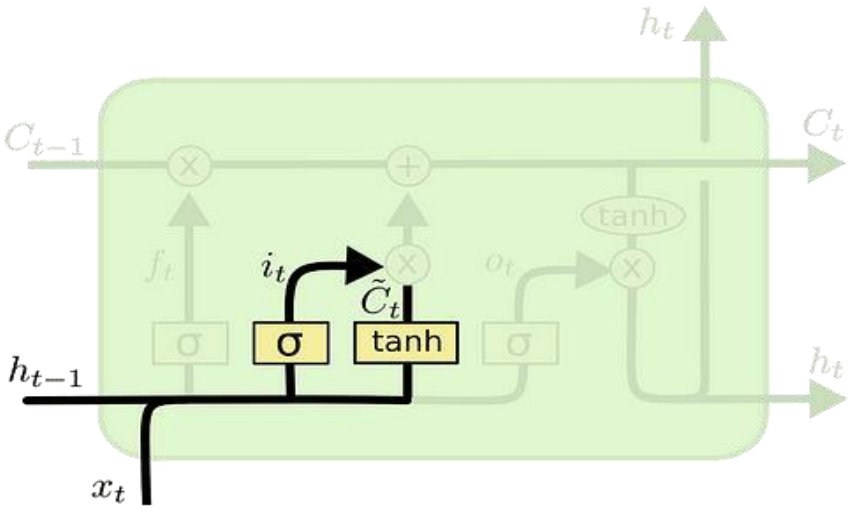


Figure 3.12 Input Gate

- **Forget Gate:** This gate decides what information from the cell state to forget or retain. It takes as input the previous hidden state (h_{t-1}) and current input (x_t), and outputs a number between 0 and 1 for each number in the cell state (C_{t-1}). A 1 represents "keep this," while a 0 represents "get rid of this."

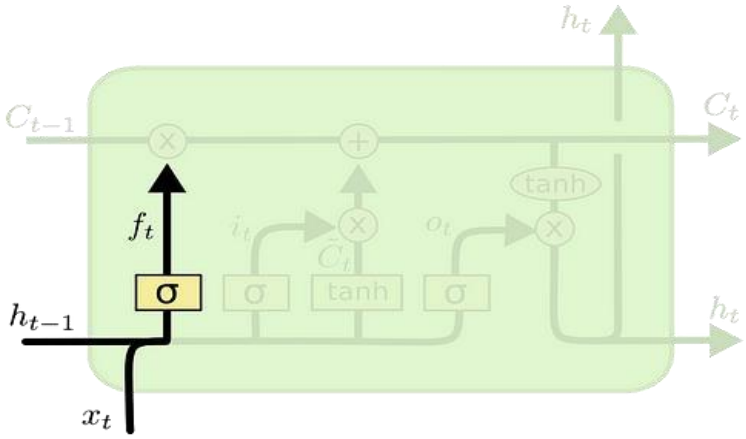


Figure 3.13 Forget Gate

- **Output Gate:** This gate decides what part of the cell state to output. The current input and previous hidden state are used as input to the Output Gate, which decides what part of the cell state to expose as the output at the current time.

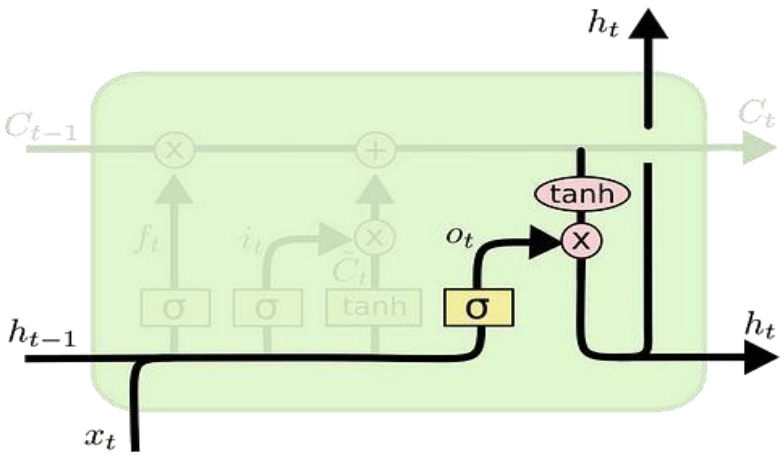


Figure 3.14 Output Gate

- **Cell State:** The cell state is the main conveyor belt that runs through the entire chain of LSTM units. It allows information to flow unchanged through many time steps, mitigating the vanishing gradient problem. The cell state is regulated by various gates to add or remove information as it passes through the network.

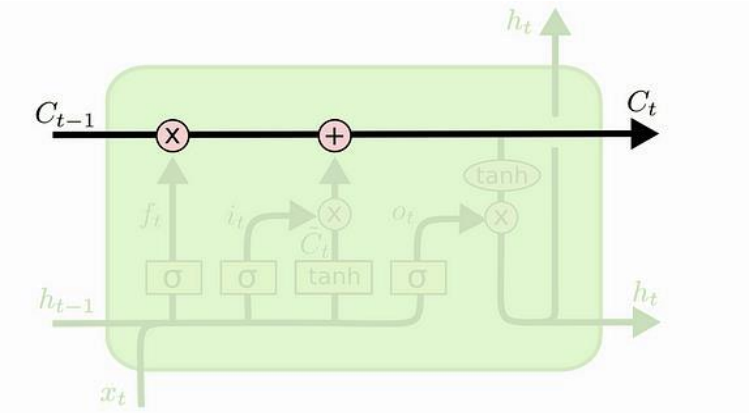


Figure 3.15 Cell State

- **Hidden State:** The concept of increasing number of layers in an LSTM network is rather straightforward. All time-steps get put through the first LSTM layer / cell to generate a whole set of hidden states (one per time-step). These hidden states are then used as inputs for the second LSTM layer / cell to generate another set of hidden states, and so on and so forth.

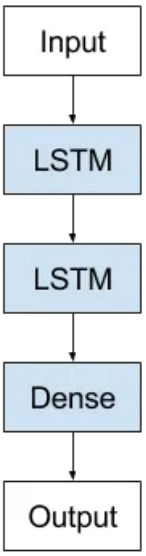


Figure 3.16 Hidden State

3.7 Building Model

3.7.1 Dataset

The CoinsCrypt project utilizes financial data from Yahoo Finance, a comprehensive source for real-time and historical market data. Yahoo Finance provides various features that are essential for analysing and predicting cryptocurrency prices. Below is a detailed explanation of the features provided by Yahoo Finance and how they are used in the application.

Data Features

Yahoo Finance provides several key features for each cryptocurrency. These features are crucial for understanding the market behaviour and trends over time. Here is a breakdown of each feature:

- **Date**
  - **Description**: The specific day on which the market data was recorded.
  - **Usage**: This feature is used to organize and sequence the data points in chronological order. It is essential for plotting time series charts and performing time-based analyses.

- **Open**
  - **Description**: The price of the cryptocurrency at the start of the trading day.
  - **Usage**: The opening price gives insight into the initial market sentiment and is often used in conjunction with other features to identify price movements and trends.

- **Close**
  - **Description**: The price of the cryptocurrency at the end of the trading day.
  - **Usage**: The closing price is a crucial indicator of the day's market performance. It is commonly used for historical comparisons, trend analysis, and in calculating returns.

- **High**
  - **Description**: The highest price at which the cryptocurrency traded during the day.
  - **Usage**: The high price indicates the peak value within a trading day, providing insights into the maximum potential value reached. It is often analysed to

understand volatility and market strength.

- **Low**
  - **Description**: The lowest price at which the cryptocurrency traded during the day.
  - **Usage**: The low price shows the minimum value within a trading day, which helps in assessing the market's lower bounds and price support levels.

- **Adjusted Close**
  - **Description**: The closing price adjusted for dividends, stock splits, and new stock offerings.
  - **Usage**: The adjusted close provides a more accurate reflection of the true value of a cryptocurrency by accounting for events that affect its price. It is particularly useful for long-term analysis and comparing historical performance.

- **Volume**
  - **Description**: The total number of cryptocurrency units traded during the day.
  - **Usage**: Trading volume measures market activity and liquidity. High volume often indicates strong interest and can signal potential price movements, while low volume might suggest limited trading interest or stability.

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	09-11-2017	308.644989	329.451996	307.056	320.884003	320.884003	893249984
3	10-11-2017	320.67099	324.717987	294.541992	299.252991	299.252991	885985984
4	11-11-2017	298.585999	319.453003	298.191986	314.681	314.681	842300992
5	12-11-2017	314.690002	319.153015	298.513	307.90799	307.90799	1613479936
6	13-11-2017	307.024994	328.415009	307.024994	316.716003	316.716003	1041889984
7	14-11-2017	316.763	340.177002	316.763	337.631012	337.631012	1069680000
8	15-11-2017	337.963989	340.911987	329.812988	333.356995	333.356995	722665984
9	16-11-2017	333.442993	336.158997	323.605988	330.924011	330.924011	797254016
10	17-11-2017	330.166992	334.963989	327.52301	332.394012	332.394012	621732992

Figure 3.7 Dataset View (Ethereum)

3.7.2 Necessary Libraries

- **Data Handling and Manipulation**
  - **os**: A standard library in Python that provides functions for interacting with the operating system, such as reading and writing files.
  - **pandas**: A powerful library for data manipulation and analysis, offering data structures like DataFrames to efficiently handle and analyse structured data.
  - **numpy**: A fundamental library for numerical computing in Python, providing support for arrays, mathematical functions, and linear algebra operations.
  - **math**: A standard library offering mathematical functions like exponential, logarithmic, and trigonometric operations.
  - **datetime**: A module for handling date and time operations, useful for timestamping and time series analysis.

- **Evaluation Metrics**
  - **sklearn.metrics**: A module from scikit-learn providing various metrics for evaluating the performance of machine learning models, including:
    - **mean_squared_error** and **mean_absolute_error**: Metrics for regression models to measure prediction accuracy.
    - **explained_variance_score** and **r2_score**: Metrics for assessing how well the model explains the variance in the data.
    - **mean_poisson_deviance** and **mean_gamma_deviance**: Specialized metrics for specific types of data distributions.
    - **accuracy_score**: Metric for evaluating the accuracy of classification models.
  - **sklearn.preprocessing.MinMaxScaler**: A tool for scaling and normalizing data, transforming features to a given range (e.g., between 0 and 1), which is crucial for optimizing the performance of certain machine learning algorithms.

- **Model Building**
  - **tensorflow**: An open-source platform for machine learning and deep learning developed by Google. It provides comprehensive tools and libraries for building and training neural networks.

- **tensorflow.keras**: A high-level API within TensorFlow for building and training deep learning models. Key components include:
    - **Sequential**: A linear stack of layers.
    - **Dense** and **Dropout**: Layers for fully connected neural networks and regularization.
    - **LSTM**: A specialized layer for building Long Short-Term Memory networks, ideal for time series prediction tasks.

- **Plotting and Visualization**
    - **matplotlib.pyplot**: A plotting library for creating static, interactive, and animated visualizations in Python. It is widely used for generating line plots, bar charts, histograms, and other types of graphs.
    - **itertools.cycle**: A tool for iterating over a collection in a cyclic manner, useful for repeating a set of items indefinitely.
    - **plotly.graph_objects**, **plotly.express**, and **plotly.subplots**: Part of the Plotly library for creating interactive and dynamic visualizations. Plotly offers extensive support for various plot types, including:
        - **graph_objects**: For creating complex and highly customizable visualizations.
        - **express**: For quickly generating common plot types with minimal code.
        - **subplots**: For creating multi-plot layouts, allowing for detailed and comparative visual analyses.

3.7.3 Data Preprocessing

- First Step is Preparing Data for Training and Testing.
- Here we are just considering 3 years data for training data.
- Since Bitcoin price has drastically fluctuated from 200 dollar in year 2014 to 15000 dollar in year 2018 to 3000 dollar in year 2019(theses values are approx.) so we will just consider 3 Years to avoid this type of fluctuation in the data.
- As we want to predict Close Price of the Bitcoin so we are just Considering Close and Date.

Figure 3.18 Close Price for last three years

3.7.3.1 Normalizing Data

- Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information.
- MinMaxScaler. For each value in a feature, MinMaxScaler subtracts the minimum value in the feature and then divides by the range. The range is the difference between the original maximum and original minimum. MinMaxScaler preserves the shape of the original distribution.
- Output after normalizing data, shape of the new dataframe:

```
(1095, 1)
```

3.7.3.2 Data Splitting

- We keep the training set as 60% and 40% testing set.

```
train_data:  (657, 1)
test_data:  (438, 1)
```

- Now we Transform the Close price based on Time-series-analysis forecasting requirement , here we will take 15-time steps.
  - **Time series forecasting** involves using historical data to predict future values in a series. It is widely used in various domains such as finance, weather prediction,

stock market analysis, and economics. The goal is to model the time series data so that accurate predictions can be made for future time points.

- When time_step = 15, the function will:
  - Create sequences of 15 consecutive data points from the dataset to use as input features (dataX).
  - The target value (dataY) will be the data point that immediately follows each sequence of 15 points.

```
X_train:  (641, 15)
y_train:  (641,)
X_test:  (422, 15)
y_test (422,)
```

- Reshaping input to be [samples, time steps, features] which is required for LSTM.

```
X_train:  (641, 15, 1)
X_test:  (422, 15, 1)
```

3.7.4 Actual Model Building

- Model Initialization
  1. Sequential Model:
     - **Definition**: The model is initialized using the Sequential class from the tensorflow.keras.models' module.
     - **Purpose**: The Sequential model is a linear stack of layers, where each layer has one input tensor and one output tensor, making it simple to build models layer by layer.

- Layers
  2. LSTM Layer:
     - **Layer Type**: Long Short-Term Memory (LSTM) layer, added using model.add(LSTM(...)).
     - **Units**: The parameter 10 specifies the number of LSTM units (neurons) in the layer.
     - **Input Shape**: input_shape=(None, 1) indicates the shape of the input data. Here, none represents the time steps (variable length), and 1 represents the number of features in the input.

- **Activation Function**: activation="relu" applies the Rectified Linear Unit (ReLU) activation function to the output of each LSTM unit. ReLU introduces non-linearity, allowing the model to learn more complex patterns.
- **Purpose**: The LSTM layer is crucial for capturing temporal dependencies and sequential patterns in time series data, making it suitable for forecasting tasks.

3. Dense Layer:
   - **Layer Type**: Fully connected (Dense) layer, added using model.add(Dense(1)).
   - **Units**: The parameter 1 specifies that the layer has a single output unit, which is suitable for a regression task where we predict a single continuous value (e.g., the next time step value).
   - **Purpose**: The Dense layer aggregates the outputs from the LSTM layer and produces the final prediction.

- Model Compilation
  4. Loss Function:
     - **Definition**: Specified using loss="mean_squared_error".
     - **Type**: Mean Squared Error (MSE) is used as the loss function.
     - **Purpose**: MSE measures the average squared difference between the predicted values and the actual values, providing a clear metric for how well the model is performing. It is suitable for regression tasks where the goal is to minimize the error between predicted and actual values.

  5. Optimizer:
     - **Definition**: Specified using optimizer="adam".
     - **Type**: Adam (Adaptive Moment Estimation) optimizer.
     - **Purpose**: The Adam optimizer combines the advantages of two other extensions of stochastic gradient descent, namely AdaGrad and RMSProp. It adapts the learning rate for each parameter, making it efficient and suitable for training deep learning models.

3.8 Summary

The chapter outlines the development of the CoinsCrypt web application, detailing the integration of frontend and backend technologies such as React, Firebase, and various APIs for cryptocurrency data and news. The system design includes architecture and use case diagrams illustrating user authentication, price tracking, watchlist management, and news integration. The technology stack encompasses React for the frontend, Bootstrap and Material UI for styling, Chart.js for data visualization, Firebase for backend services, and CoinGecko and Newsdata APIs for data fetching. Additionally, the chapter introduces predictive analysis using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, highlighting the data preprocessing steps, necessary libraries, and model building process essential for accurate cryptocurrency price predictions.

CHAPTER 4
EXPERIMENTAL RESULTS

4.1 Graphical User Interface (GUI)

4.1.1 Home Page

The homepage of the application serves as the central hub where users can access key features and information briefly. The homepage is made with React. At the top, there is a navbar that is made using the CSS framework Bootstrap, a fully responsive navbar with a hamburger menu for mobile that users can use to navigate to other pages. The homepage also provides some important information about crypto market statistics that users may be interested in. There is a carousel for the trending coins that shows some coins that are currently in high demand or high gainers. Users can see more information about them by clicking them and navigating to the single coin page.

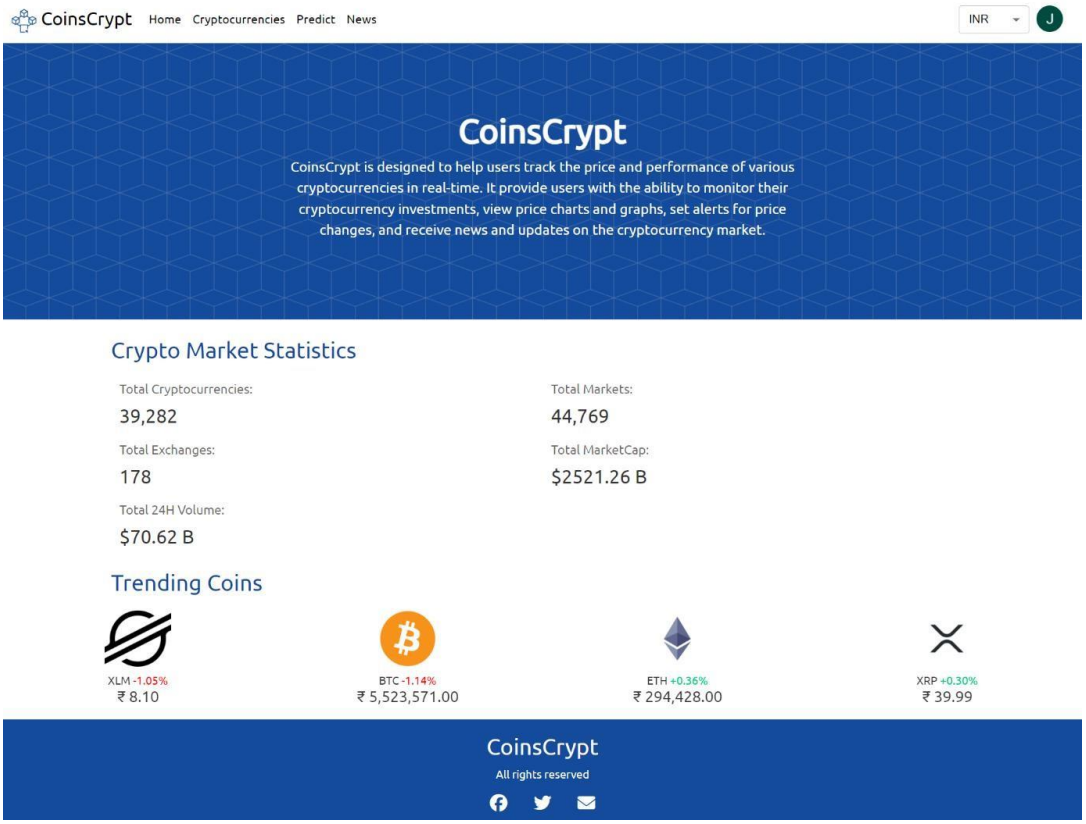


Figure 4.1 Home Page

4.1.2 Coins Table Page

The Coins Table Page has a list of 100 cryptocurrencies from which users can select their favourite coin and see more information about the coin by navigating to the single coin page. The list is made using the Material UI, and it has a pagination feature. Only 10 coins are featured per page. Users can select another page to see the nest coins. All the coins are sorted in descending order by market cap. There is a search bar at the top to search for any coin from the list if they really want to search for it.

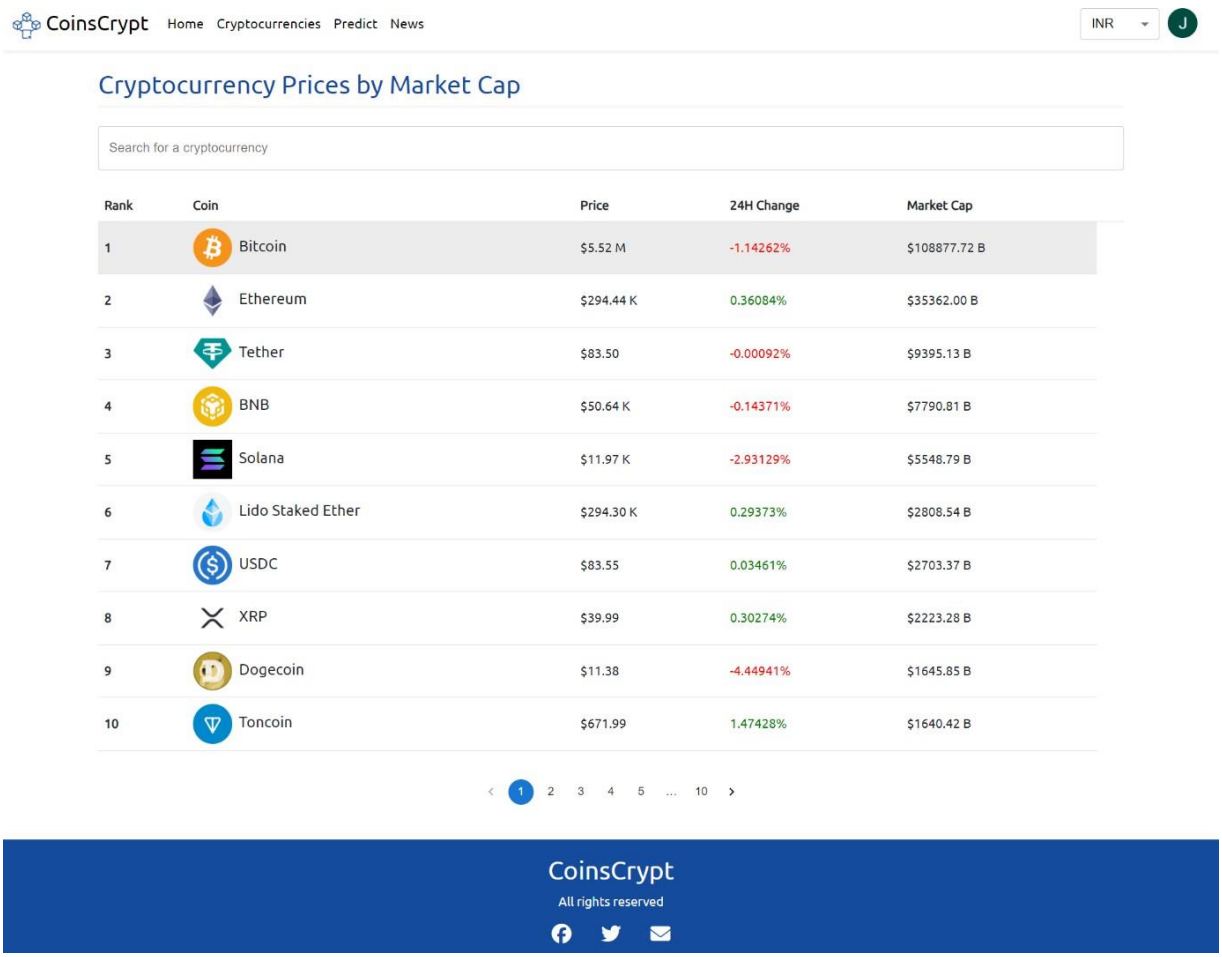


Figure 4.2 Coins Table Page

4.1.3 Prediction List Page

The Prediction List Page shows the list of all the coins for which users can see the predicted value of the coin for the next month. It shows the name of the coin along with the logo and ID of the coin. Users can click on the coin for which they want to see the prediction and navigate to the respected prediction page.

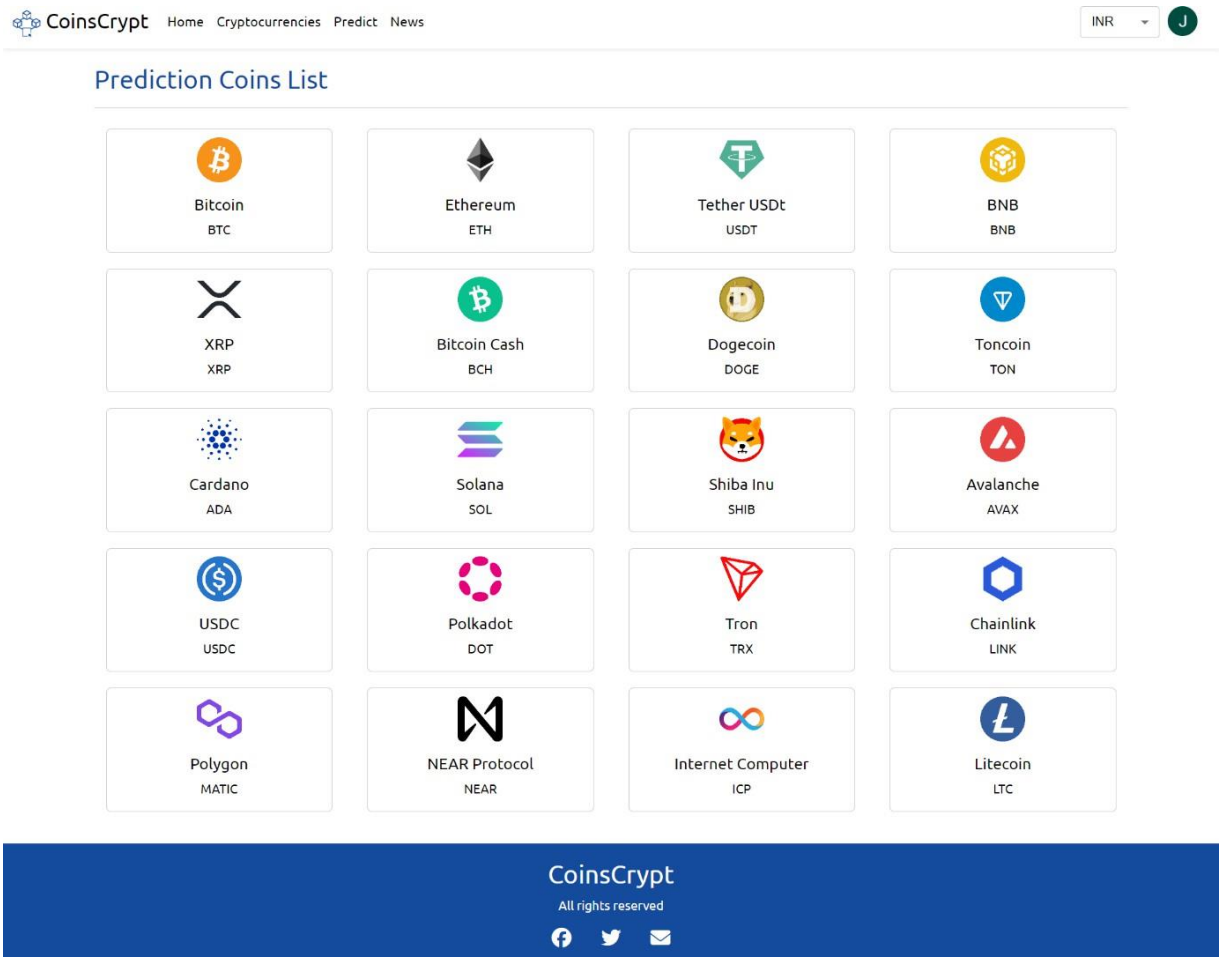


Figure 4.3 Prediction List Page

4.1.4 News Page

The news page shows the latest articles related to cryptocurrencies or the crypto market. Articles are shown in grid format. Each box shows the image related to the article and a small preview of the article. It also shows the author and the time at which the article was posted. For the news articles, the Newsdata API is used, which provides a list of the articles in JSON format, and then the articles are formatted to be viewed on the page.

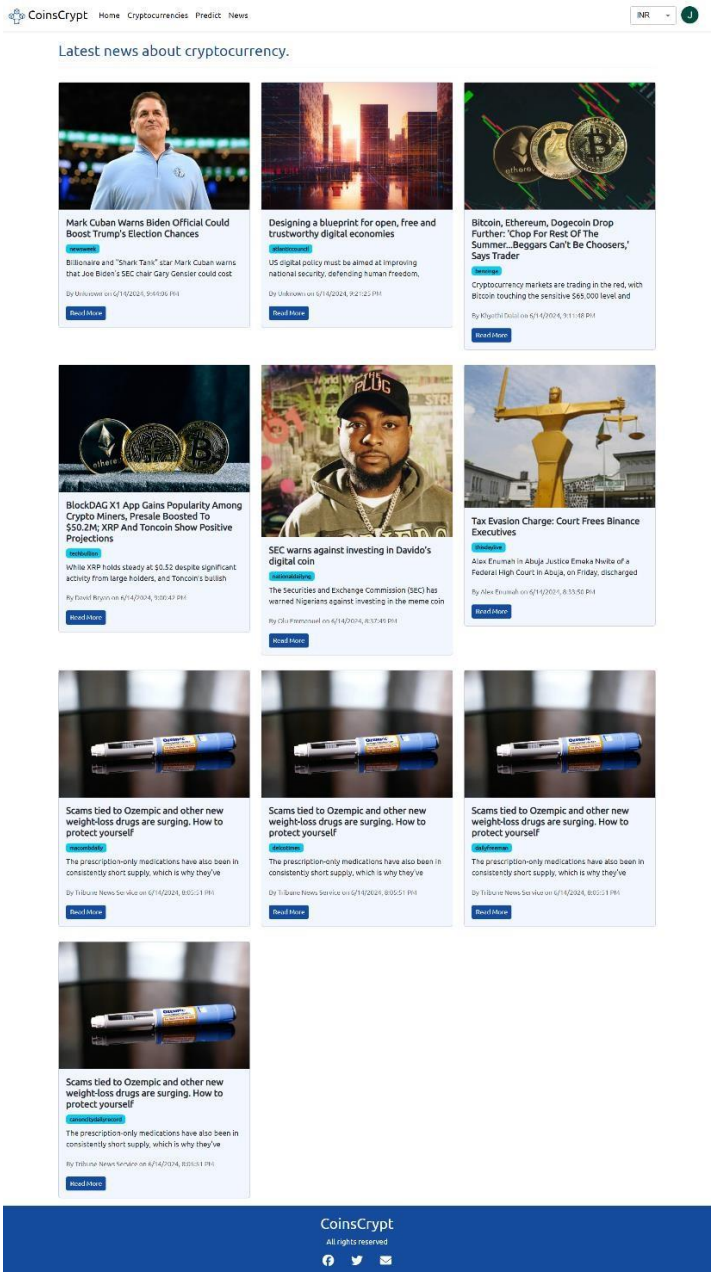


Figure 4.4 News Page

4.1.5 Single Coin Page

This page provides historical data about single coins. Users can navigate to this page either from the trending coin carousel on the homepage or from the coin table by clicking on the coin, and they will be redirected to this page of the coin. This page has information about the coin, like its rank, current price, market cap, and some other lines of information that new users might be interested in. Historical data on the coin is shown in the form of a graph, which is made using the Chart.js library. Users can select from either of the of the options provided below to change the time, and they can see the graphical data.

Figure 4.5 Single Coin Page (Bitcoin)

4.1.6 User Login/Signup

User login/signup is the most important part of the application. This is what makes it personalized for each user. The user can sign up either with their email or using their Google account. After signing up, users can add their favourite coins to the watchlist.

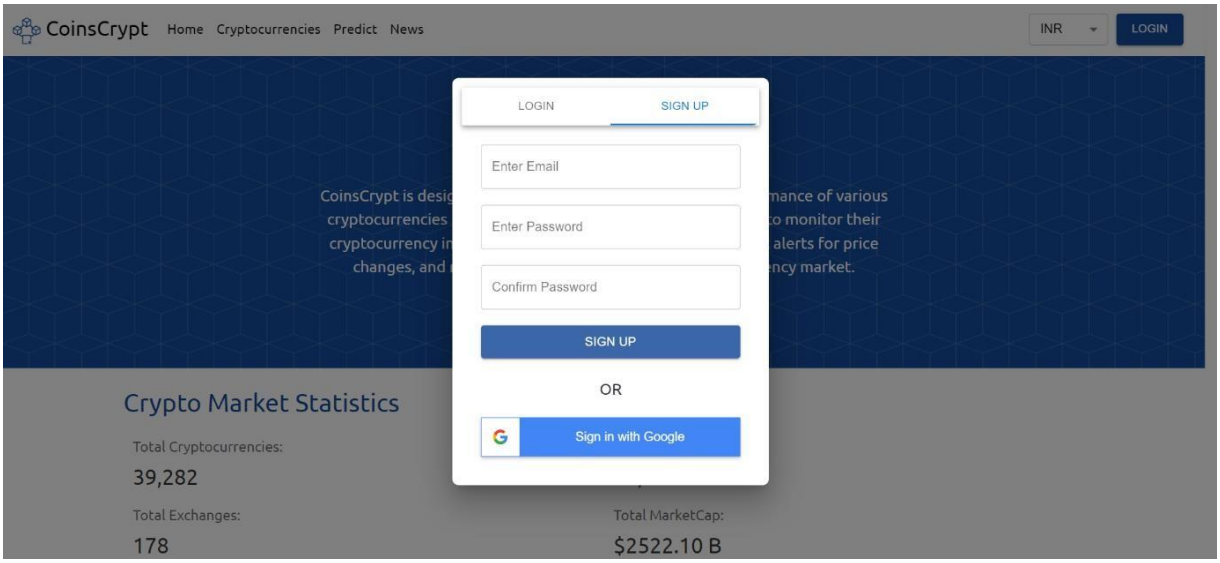


Figure 4.6 Signup Modal

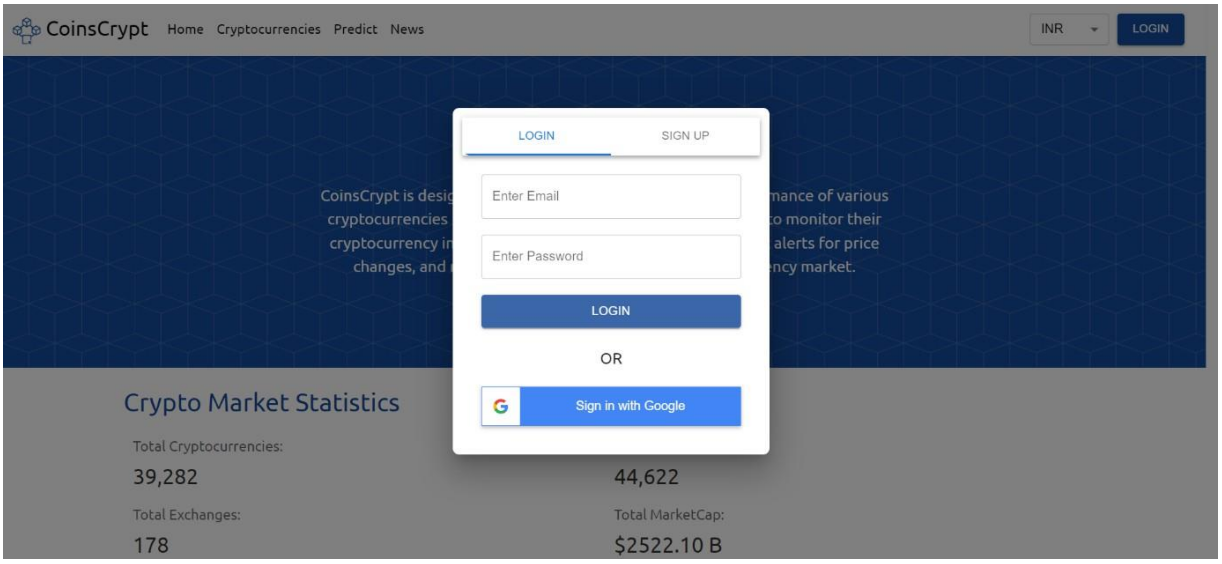


Figure 4.7 Login Modal

4.1.7 Watchlist

Watchlist is where users can add their favourite coins and check their performance at any time. Users can add coins to the watchlist by clicking on the Add to watchlist button on the single coins page, and the coin will be added to the watchlist, which they can see by clicking on the avatar on the right-hand side of the navbar, and then the watchlist modal will open with the watchlist, user confirmation, and the logout button. The user can use the delete bin next to the coin to delete it from the watchlist.

Figure 4.8 Watchlist Sidebar

4.2 Firebase – Backend-as-a-Service

4.2.1 Introduction

Firebase is a Google product that helps developers easily build, manage, and test their applica tions. It helps developers build applications faster and more securely. There is no need for an y programming on the Firebase side, making it easier to use its better features. Provides andro id, ios, web and integration services. He placed the cloud on the ground. It uses NoSQL as th e database to store information.

- **Authentication:** Firebase provides users with a secure and easy way to access their a pplications. Developers can use Firebase Authentication to support email and passwor d logins, Google logins, Facebook logins, and more.

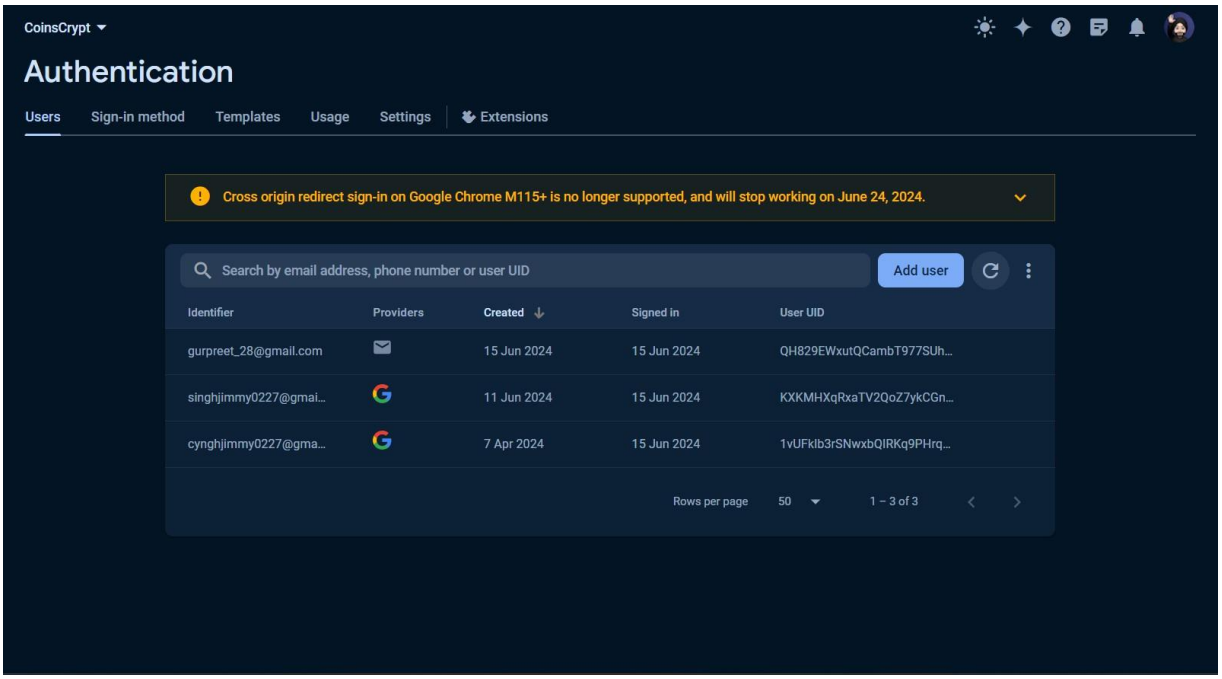


Figure 4.9 Users View

- **Realtime Database:** Firebase Realtime Database is a cloud-hosted NoSQL database that allows organizations to store and synchronize data in real-time across all client devices. This makes it easy to create applications that are always updated, even for offline users.

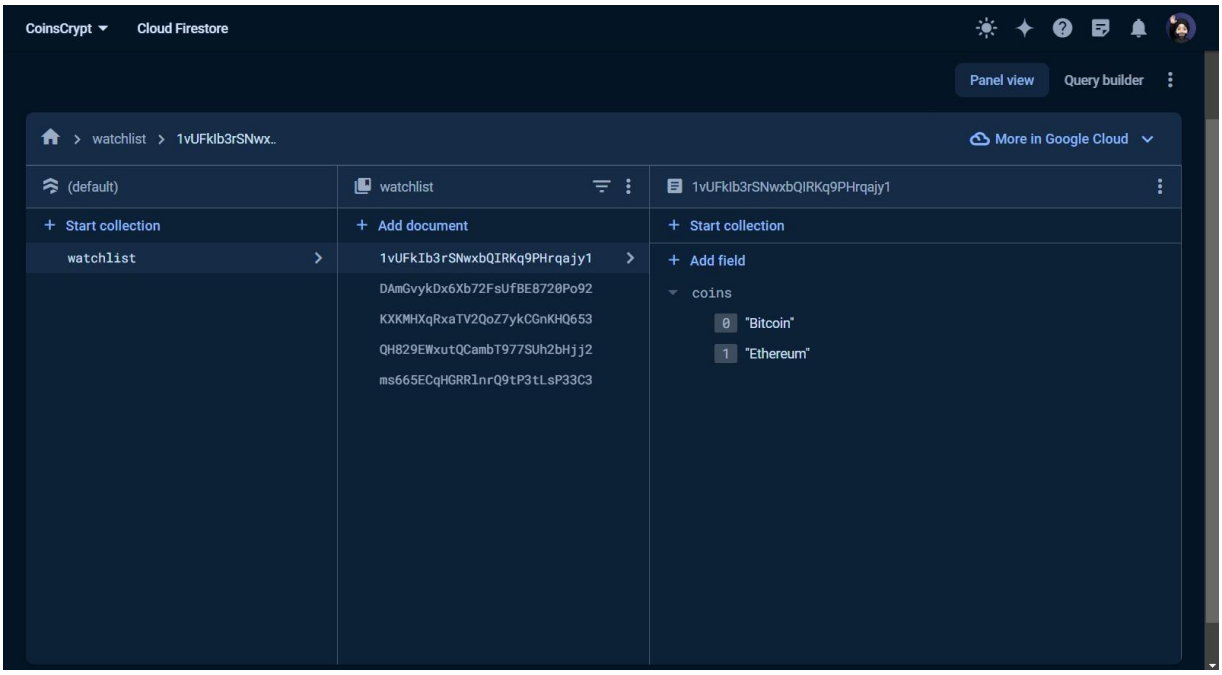


Figure 4.10 Watchlist Database

4.3 Model Performance Metrics

The performance of the LSTM model was evaluated using several metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) score. These metrics provide insight into the accuracy and reliability of the model's predictions.

Train data RMSE: 0.084

Train data MSE: 0.007

Train data MAE: 0.057

Test data RMSE: 0.035

Test data MSE: 0.001

Test data MAE: 0.024

Train data R2 score: 0.968

Test data R2 score: 0.952

4.4 Training Loss and Validation Loss

The training process was monitored by plotting the loss over epochs for both the training and validation datasets. This helps in understanding the model's learning curve and identifying any potential overfitting or underfitting issues.

The plot shows that the training loss decreased steadily, indicating that the model was learning effectively. The validation loss also decreased, suggesting that the model generalizes well to unseen data.

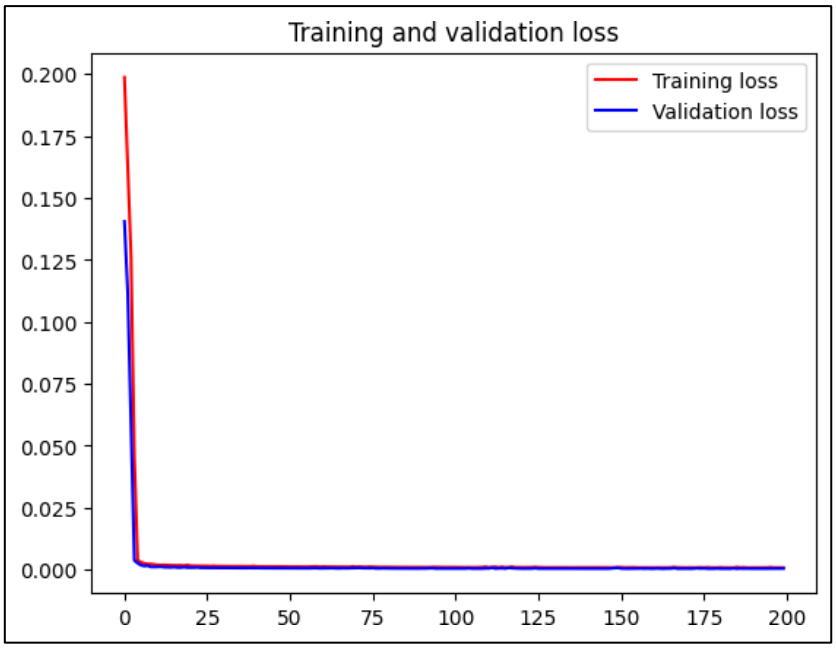


Figure 4.11 Training Loss vs Validation Loss

4.5 Predictions vs Actual Values

To evaluate the model's performance, we compared the predicted values with the actual values for a subset of the data. The following plot illustrates this comparison.

Figure 4.12 Prediction vs Actual Values

The plot demonstrates that the model's predictions closely follow the actual values, indicating that the model can capture the underlying patterns in the data.

4.6 Forecasting Future Prices

One of the primary objectives of the CoinsCrypt application is to forecast future cryptocurrency prices. The model was used to predict the prices for the next 30 days, and the results are plotted below.

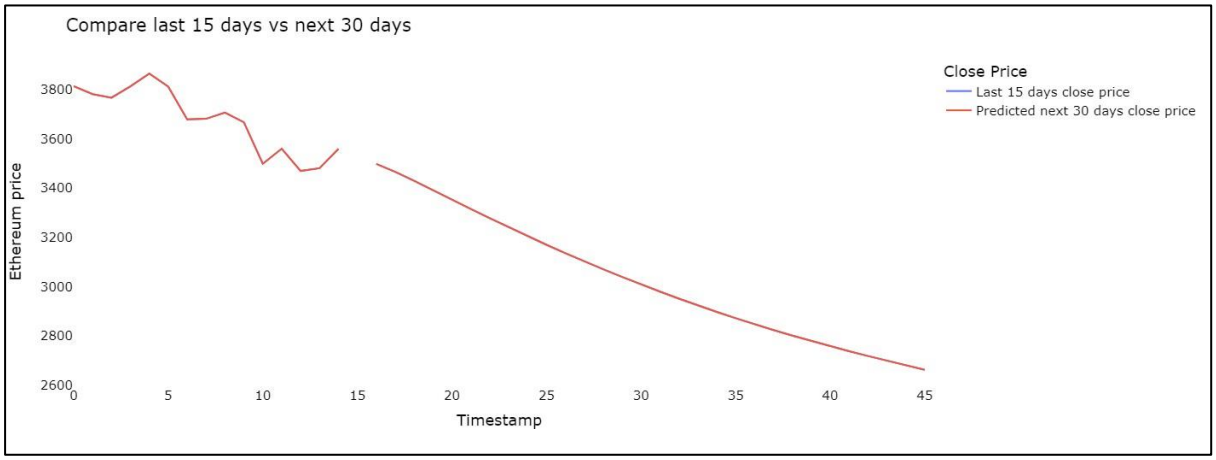


Figure 4.13 Predicted Value of Next 30 Days

The forecast plot shows the predicted prices for the next 30 days, providing users with valuable insights into potential future market trends. This information can be crucial for making informed investment decisions.

4.7 Results Plotting Using Streamlit

The visualization component of the CoinsCrypt project is crucial for presenting the forecasting results in an interactive and comprehensible manner. By using Streamlit, we enable users to visualize the predicted cryptocurrency prices through an engaging web interface. This allows users to easily interpret and make informed decisions based on the model's predictions.

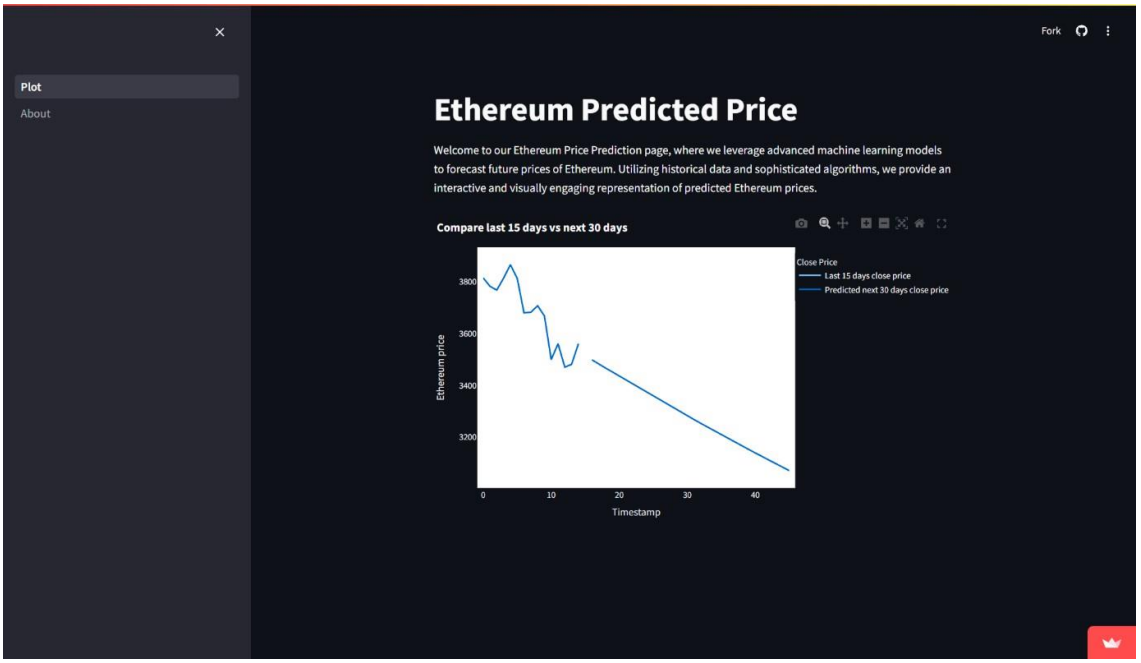


Figure 4.14 Predicted Values Plot on Web

4.8 Summary

The chapter describes the predictive analysis methodology for the CoinsCrypt project, utilizing financial data from Yahoo Finance to forecast cryptocurrency prices. It covers the data preprocessing steps, including normalization, data splitting, and transformation for time series analysis. The chapter elaborates on the use of Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, detailing their components and various configurations. It also outlines the implementation of the predictive model, from initializing a sequential model with LSTM and Dense layers to compiling it with mean squared error loss and the Adam optimizer, aiming to capture temporal dependencies and provide accurate price predictions.

CHAPTER 5
CONCLUSION AND FUTURE WORKS

5.1 Conclusion

The CoinsCrypt project successfully implements a comprehensive cryptocurrency price tracking and prediction web application. By leveraging advanced machine learning techniques, particularly Long Short-Term Memory (LSTM) networks, the application provides accurate short-term price forecasts for various cryptocurrencies. This enables users to make informed decisions based on both historical data and predicted future trends.

The integration of React for the frontend ensures a responsive and interactive user interface, enhancing user experience. Bootstrap and Material UI were instrumental in creating a visually appealing and cohesive design. Firebase effectively manages user authentication and stores user-specific data, such as watchlists, securely. The use of Chart.js for displaying cryptocurrency prices and historical data in chart format, and the implementation of Streamlit for displaying predictive results, adds significant value to the application by making complex data comprehensible and accessible.

Overall, CoinsCrypt stands as a robust tool for cryptocurrency enthusiasts and traders, providing essential features like real-time price tracking, historical data analysis, news updates, and future price predictions. The successful integration of various technologies showcases the potential of web applications in the financial technology domain.

5.2 Future Works

While the CoinsCrypt project achieves its primary objectives, there are several areas for potential enhancement and expansion to increase its utility and functionality:

- **Enhanced Prediction Models**
    - **Incorporating Other Machine Learning Models**: Future iterations of the project could explore other machine learning models such as ARIMA, Prophet, or even more advanced deep learning models like Transformer networks. Comparing the

performance of these models with the current LSTM model could provide deeper insights and potentially more accurate predictions.

- **Multi-feature Analysis**: Including additional features such as trading volume, social media sentiment, and macroeconomic indicators could improve the accuracy of the predictions by providing a more comprehensive understanding of the factors influencing cryptocurrency prices.

- **Real-time Data Integration**

  - **Live Data Feeds**: Integrating live data feeds from cryptocurrency exchanges can enhance the real-time tracking capability of the application, providing users with the most up-to-date information.
  - **Automated Alerts**: Implementing a system for automated alerts based on significant price changes or predefined conditions can help users stay informed about critical market movements.

- **Expanded User Features**

  - **Portfolio Management**: Adding features for users to manage their cryptocurrency portfolios, including tracking their holdings, calculating gains/losses, and providing personalized investment insights.
  - **Social Features**: Introducing social features where users can share their watchlists, discuss market trends, and follow other traders could create a more engaging and collaborative platform.

REFERENCES

1. Nakamoto, S. (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System." Retrieved from https://bitcoin.org/bitcoin.pdf

2. Cryptocurrency: How Bitcoin and Digital Money are Challenging the Global Economic Order" by Paul Vigna and Michael J. Casey

3. Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). "Bitcoin and Cryptocurrency Technologies." Princeton University Press. ISBN: 978-0691171692.

4. Brownlee, J. (2017). "Long Short-Term Memory Networks with Python: Develop Sequence Prediction Models with Deep Learning." Machine Learning Mastery. Retrieved from https://machinelearningmastery.com.

5. Agarwal, S., & Gupta, A. (2019). "Predicting Stock Prices Using LSTM." International Journal of Engineering and Advanced Technology (IJEAT), 8(3), 2249-8958.

6. Sito, M. (2020). "React.js Essentials." Packt Publishing. ISBN: 978-1783551620.

7. Bulma, M. (2018). "Bootstrap 4 Quick Start: A Beginner's Guide to Building Responsive Layouts with Bootstrap 4." Independently published. ISBN: 978-1983088559.

8. Sommer, S., & Christianson, E. (2020). "Mastering Firebase for Web Applications." Apress. ISBN: 978-1484262425.

9. Milani, S. (2017). "Mastering Full Stack React Web Development." Packt Publishing. ISBN: 978-1786461766.

10. Rodriguez, C. (2020). "Material UI Made Simple: A Step-by-Step Guide for Developing Modern Web Apps with Material UI." Independently published. ISBN: 979-8648524183.

11. Rao, A. R., & Rambhia, M. (2018). "Cryptocurrency Price Prediction Using Time Series Forecasting." International Journal of Engineering and Computer Science, 7(4), 23833-23837.

12. Agarwal, S., & Gupta, A. (2019). "Predicting Stock Prices Using LSTM." International Journal of Engineering and Advanced Technology (IJEAT), 8(3), 2249-8958.

13. Yan, L., & Ouyang, Z. (2018). "Application of LSTM Neural Networks in Stock Price Prediction." Engineering Letters, 26(2), 276-283.

Crypo currency

ORIGINALITY REPORT

18%	17%	7%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	medium.com Internet Source	3%
2	www.coursehero.com Internet Source	1%
3	5dok.net Internet Source	1%
4	dokumen.pub Internet Source	1%
5	Olushola, Akinbusola. "Enhanced BSTS and LSTM Model for Irregular and Mixed Frequency Time Series Data", Indiana University of Pennsylvania, 2024 Publication	1%
6	fastercapital.com Internet Source	<1%
7	libweb.kpfu.ru Internet Source	<1%
8	umpir.ump.edu.my Internet Source	<1%