A PROJECT REPORT

on

"AI POWERED CRYPTOCURRENCY PRICE PREDICTOR"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN

Computer Science and Engineering

SUBMITTED BY:

Aaryak Prasad	**2105171**
Shivli Singh	**2105237**
Abhipsha Das	**21052725**
Swayam Yadav	**21052956**
Neetu Dey	**21053298**
Aayushma Gautam	**21053475**

UNDER THE GUIDANCE OF:

Dr. Bhaswati Sahoo

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

November 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"AI POWERED CRYPTOCURRENCY PRICE PREDICTOR"

SUBMITTED BY:

Aaryak Prasad	2105171
Shivli Singh	2105237
Abhipsha Das	21052725
Swayam Yadav	21052956
Neetu Dey	21053298
Aayushma Gautam	21053475

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2023-2024, under our guidance.

Date: 2 4 / 1 1 / 2 4

(Dr. Bhaswati Sahoo)
Project Guide

ACKNOWLEDGEMENT

We are profoundly grateful to **Dr.Bhaswati Sahoo of KIIT University** for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

Aaryak Prasad

Shivli Singh

Abhipsha Das

Swayam Yadav

Neetu Dey

Aayushma Gautam

ABSTRACT

As the cryptocurrency market continues to evolve at a rapid pace, accurate and timely price predictions are essential for investors, traders, and enthusiasts to make informed decisions. This project introduces a live cryptocurrency prediction platform that leverages advanced analytics and cutting-edge technology to provide real-time data and price forecasts. Utilizing a Long Short-Term Memory (LSTM) model for price prediction, the platform forecasts cryptocurrency values for a variety of currencies, including BTC-USD, ETH-USD, and BTCK-USD, with customizable prediction durations of up to 100 days. The backend of the platform is built with Flask, ensuring smooth data processing and secure user authentication, while the frontend utilizes HTML, CSS, and JavaScript to deliver a responsive, intuitive, and visually engaging user interface. With a focus on seamless navigation and user experience across both desktop and mobile devices, the platform empowers users to track market trends and tailor forecasts to suit their individual trading strategies. By integrating sophisticated predictive models with an accessible interface, this project equips users—from beginners to experts—with the insights and tools needed to make confident decisions in the volatile cryptocurrency market.

Keywords: Cryptocurrency predictions, Bitcoin price prediction, LSTM model, BTC-USD, ETH-USD, BTCK-USD, forecasting, Flask backend, HTML, CSS, JavaScript, real-time data, responsive design.

Contents

1. INTRODUCTION

In today’s fast-paced cryptocurrency market, access to accurate, timely, and reliable predictions is vital for traders, investors, and enthusiasts of all experience levels. This project introduces a live cryptocurrency value prediction website that combines advanced analytics with an intuitive and visually appealing design, creating a platform that is both engaging and highly functional. Featuring a modern, user-friendly interface with smooth navigation and secure login and sign-up options, the website ensures personalized experiences for users. Users gain access to real-time cryptocurrency data through the “predict” section, where continuous updates keep them informed of the latest market trends. The platform’s standout feature is its advanced prediction page, allowing users to forecast cryptocurrency prices for customizable durations of up to 100 days, making it suitable for both short-term and long-term trading strategies.

Moreover, the prediction page supports a variety of popular currency pairs, including BTC-USD, ETH-USD, and BTCK-USD, enabling users to tailor forecasts to their specific interests and requirements. The responsive and engaging design ensures seamless performance across devices, providing an optimal experience whether users are accessing it from desktops, tablets, or smartphones. By integrating sophisticated prediction algorithms with an easy-to-navigate interface, this platform not only delivers real-time price tracking but also empowers users with actionable insights to make well-informed decisions in the dynamic world of cryptocurrency. Whether you're a seasoned investor or new to the market, this website serves as a powerful tool to navigate and thrive in the ever-evolving cryptocurrency landscape.

Key features include real-time price tracking, customizable forecasting durations, and a responsive, engaging design that ensures a seamless device experience. By combining advanced analytics with an intuitive and visually appealing interface, this project empowers users to gain actionable insights and make informed decisions in the fast-evolving cryptocurrency landscape.

2. LITERATURE REVIEW

1.Aggarwal et al. (2019) explored the potential of using gold prices to predict Bitcoin prices through three deep learning models: CNN, LSTM, and GRU. Their study concluded that a model relying solely on gold prices for Bitcoin prediction deviates significantly from actual values. Among the three algorithms, LSTM demonstrated the highest prediction accuracy. Similarly, Liu et al. (2021) broadened the scope of explanatory variables by incorporating indicators from cryptocurrency markets, macroeconomic factors (such as stock indices, oil prices, and exchange rates), and search engine trends. With 40 variables, they observed that the SDAE algorithm outperformed other methods, including BPNN, PCA-SVR, and SVR, in forecasting Bitcoin prices.

2. When selecting explanatory variables, many studies commonly include macroeconomic factors. Jagannath et al. (2021) emphasized Bitcoin blockchain-specific factors like user activity, miner behavior, and exchange data as core variables. Technical indicators have also been identified as valuable tools for Bitcoin price prediction in studies by Jaquart et al. (2021) and Mudassir et al. (2020). Although LSTM models enhanced with adaptive techniques have shown promising results, comparative experiments incorporating macroeconomic factors remain underexplored in the literature.

3. Monisha Mittal et al. (2022) highlighted Bitcoin as a decentralized cryptocurrency requiring no intermediary like banks, with robust security due to its blockchain foundation. Despite its rising value since 2010 as a form of "digital gold," Bitcoin's price volatility poses significant risks to investors. This study reviewed multiple machine learning regression-based algorithms for predicting Bitcoin prices, eventually focusing on a deep learning model using Gated Recurrent Units (GRU). The GRU model effectively predicted Bitcoin prices based on historical data, with Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) used as key performance indicators to measure its accuracy.

4. Grace L.K. Joshila et al. (2021) aimed to improve the analysis and forecasting of Bitcoin prices by considering various factors influencing its value. Their research utilized daily Bitcoin market data from previous years, analyzing parameters affecting price trends. Using the Support Vector Machine (SVM) algorithm, they achieved better accuracy compared to earlier methods. This study was particularly beneficial for investors, offering predictions on price direction and helping newcomers to navigate the volatile cryptocurrency market effectively.

3. PROBLEM STATEMENT

3.1 PROJECT PROBLEM STATEMENT

In today’s rapidly growing cryptocurrency market, knowing the current value of popular cryptocurrencies like Bitcoin and predicting their future prices is crucial for making smart investment decisions. However, many existing tools are challenging to use, featuring complex interfaces and limited visual aids that hinder users' ability to interpret data effectively. This project addresses these issues by creating a user-friendly platform that not only provides real-time price updates for Bitcoin but also predicts future values for various cryptocurrencies.

The platform is designed to be simple and visually engaging, ensuring accessibility for users of all experience levels. It features clear graphs and charts, enabling users to analyze trends effortlessly and make informed decisions. Supporting a range of currency pairs, including BTC-USD, ETH-USD, BTCK-USD, and many more, the platform offers a one-stop solution for cryptocurrency tracking and forecasting. With a strong focus on visual compatibility and intuitive navigation, the platform ensures a seamless experience across desktop and mobile devices. Whether you're a seasoned investor or a cryptocurrency novice, this project equips you with the tools to stay informed and confident in your investment decisions.

3.2 PROJECT PLANNING

This project aims to develop a visually appealing and easy-to-use website that uses machine learning to predict Bitcoin prices for durations selected by users. This platform is tailored to meet the needs of cryptocurrency enthusiasts, traders, and investors by offering them tools to make informed decisions through accurate and actionable data. By combining real-time price updates with customizable forecasting options, the platform delivers a personalized and engaging experience. Secure login features protect user data and ensure safe access, while a responsive design guarantees a smooth experience across all devices.

Step 1: Requirement Analysis
The first step involved identifying the needs of cryptocurrency users, focusing on features such as real-time Bitcoin price tracking, accurate future predictions, and an intuitive user interface. Technical requirements were established to include secure login functionalities, a responsive design compatible with multiple devices, and integration with machine learning for predictive analytics. Tools and technologies were finalized for the project: HTML, and CSS, for the front end; Python frameworks like Flask or Django for the back end; and LSTM neural networks for the predictive model. This step provided a clear understanding of user expectations and technical feasibility, forming a solid foundation for subsequent development.

Step 2: Designing the System Architecture
The system architecture was designed to seamlessly connect the user interface, back-end functionality, and prediction engine. Key modules were outlined, including a real-time price module for fetching and displaying live Bitcoin data dynamically, a prediction module that allowed users to customize forecasting durations with a default setting of 30 days, and a user dashboard for personalized interactions such as storing past forecasts and managing preferences. Secure login and sign-up functionalities were integrated into the design to protect user accounts and enhance data privacy. This step ensured that the system's structure was robust and aligned with the project's objectives.

Step 3: Front-end Development
The front-end development began with a strong focus on design, ensuring the website would be both visually appealing and easy to use. Initially, the user interface was conceptualized and designed using Canva and Figma, tools that allowed the creation of detailed mockups and wireframes. These designs laid the groundwork for a seamless user experience, with a layout that prioritized intuitive navigation and accessibility. Once the design was finalized, it was implemented using HTML and CSS. These technologies were employed to create a dynamic and interactive interface, featuring responsive components that adapt flawlessly to desktops, tablets, and smartphones. Special attention was given to interactive elements like buttons, forms, and data visualizations, which were built to enhance user engagement and functionality. The result was a polished, responsive front-end that set the stage for the platform's other features.

Step 4: Back-end Development
The back-end development focused on creating a secure and efficient framework to manage user interactions and data processing. Authentication mechanisms were implemented using frameworks like Flask. Password encryption, account recovery options, and role-based access controls were integrated to enhance the overall security of the platform. Real-time data fetching and processing capabilities were developed to retrieve accurate Bitcoin values from cryptocurrency APIs.
Database: MongoDB - The backend uses MongoDB to store user data, including usernames, emails, and hashed passwords.

API Routes:

Sign Up Route
- Endpoint: `/auth/signup`
- Method: `POST`
- Description: Registers a new user by accepting JSON data containing Username, email and password.It hashes the password and stores the user data in the MongoDB database.
- Response: Returns a JSON response with a redirect URL to the `prediction` page if successful, or an error message if the request is invalid.

Login Route
- Endpoint: `/auth/login`
- Method: `POST`
- Description: Authenticates a user by accepting JSON data containing username and password. It checks the credentials against the stored data in the MongoDB database.
- Response: Returns a JSON response with a redirect URL to the `dashboard` page if successful, or an error message if the credentials are invalid.

Prediction Page Route
- Endpoint: /prediction
- Method: GET
- Authentication: Requires user to be logged in (@auth.login_required)
- Description: Renders the prediction page where users can input data for stock price predictions.

Price Prediction Input Route
- Endpoint: /pricepredict
- Methods: GET, POST
- Description:

GET: Renders a form where users can input the stock ticker and the number of days for prediction.
POST: Processes the form data and redirects to the predict route with the necessary parameters.
This backend setup ensures secure user registration and authentication using hashed passwords and MongoDB for data storage. The API routes handle JSON requests and provide appropriate responses for client-side handling.

Step 5: Machine Learning Integration
The predictive aspect of the platform was built using LSTM (Long Short-Term Memory) networks, a specialized type of neural network well-suited for sequential data. The model was trained on historical Bitcoin data to accurately forecast future trends. Once trained, the model was fine-tuned to improve its reliability and integrated into the website's back-end. Users were given the flexibility to select the duration for predictions, with a default setting of 30 days. The prediction results were displayed through visually engaging graphs that compared historical and forecasted values, ensuring clarity and accessibility for users of all experience levels.

Step 6: Data Visualization
A significant emphasis was placed on data visualization to enhance the usability of the platform. Graphs and charts were developed to display historical price trends, live Bitcoin prices, and forecasted values side by side. These visual elements were interactive, allowing users to zoom in on specific timeframes, hover for detailed insights, and toggle data views for a more customized experience. The

visualizations were designed to be intuitive and user-friendly, ensuring that even beginners could easily interpret complex data. This step bridged the gap between technical data and user understanding, making the platform accessible to a broader audience.

Step 7: Testing and Optimization
To ensure the platform's reliability, thorough testing was conducted across all components, including the front-end, back-end, and machine learning model. Bugs and inconsistencies were identified and resolved to optimize performance. The website was tested for responsiveness on various devices and browsers to guarantee a consistent user experience. Additionally, the system's speed was fine-tuned to handle real-time updates without lag. User feedback from initial trials was incorporated to refine the interface and functionality further. This step ensured the platform's robustness and readiness for deployment.

Step 8: Deployment and Feedback
The final phase involved deploying the website on a secure and reliable hosting platform, making it accessible to users worldwide. Monitoring tools were set up to track user behavior and gather feedback for ongoing improvements. Regular updates to the prediction model and website features were planned to maintain accuracy and relevance in the fast-changing cryptocurrency market. User suggestions were actively considered to enhance the platform's functionality and user satisfaction, ensuring it remained a valuable tool for both novice and seasoned cryptocurrency enthusiasts.

3.3 SYSTEM DESIGN

3.3.1 Design Constraints
The design constraints establish the boundaries within which the system operates, focusing on balancing functionality, usability, and performance while addressing real-world technical limitations. These constraints ensure the platform is efficient, accessible, and reliable for all users.

Resource Constraints:
The project was developed with limited computational resources to maintain affordability and scalability. Efficient coding practices were adopted to reduce processing overhead, enabling the machine learning model and real-time data fetching modules to run smoothly even on mid-range hardware. Optimization techniques such as batching data requests and implementing caching mechanisms were utilized to minimize latency and resource consumption.

Scalability:
A key requirement was designing the system to accommodate an increasing number of users as the platform grows. To ensure scalability, modular code architecture was implemented, allowing for easy updates and extensions. The APIs used for fetching cryptocurrency data were selected based on their ability to handle

high-volume requests without degradation in performance. Additionally, server load balancing strategies were considered to maintain consistent performance during peak usage periods.

User Accessibility:
The platform was designed to provide a seamless experience for users with varying levels of expertise in cryptocurrency trading. Responsive design techniques ensured compatibility across devices, including desktops, tablets, and smartphones. The navigation structure was simplified to minimize learning curves, while features such as tooltips and brief tutorials were included to assist novice users in understanding Bitcoin trends and predictions. The platform's accessibility extends to support for users with disabilities, with considerations for screen readers and keyboard navigation.

Prediction Accuracy:
Accurate predictions were central to the platform's success. The Long Short-Term Memory (LSTM) neural network model was chosen for its superior ability to analyze sequential data like Bitcoin price trends. To maintain high prediction accuracy, the model was trained using extensive historical Bitcoin price datasets, cleaned and pre-processed to eliminate noise. Regular retraining and validation of the model were planned to adapt to evolving market patterns and ensure consistent accuracy. The prediction results were also accompanied by confidence intervals, providing users with insights into the reliability of the forecasts.
By addressing these design constraints thoughtfully, the platform achieved a balance between performance, security, and user-centric design, making it a valuable tool for cryptocurrency enthusiasts and investors.

3.3.2 System Architecture

The system architecture is designed to integrate the front-end interface, back-end logic, machine learning model, and data visualization tools to create a seamless and interactive user experience. It ensures that all components work together efficiently to provide real-time data, accurate predictions, and personalized services.

Front-End Layer:
The front end of the platform is built using HTML, CSS, and JavaScript, with initial designs created in Canva and Figma for visual planning. This layer includes interactive elements like real-time graphs, customizable prediction features, and a

user-friendly navigation system. The interface is designed to be responsive, ensuring compatibility across various devices, including desktops, tablets, and smartphones. The front end integrates seamlessly with the back end to display dynamic content and predictions, enhancing the overall user experience.

Back-End Layer:
The back end is built using frameworks such as Flask or Django, responsible for

handling user authentication, data processing, and serving prediction requests. It features secure login and sign-up functionality, with encrypted password storage and account recovery options. The backend communicates with the machine learning module and cryptocurrency APIs to fetch real-time Bitcoin price data and send prediction requests based on user input. This ensures smooth interaction between the user and the system, with data processed efficiently in real time.

Machine Learning Module:
The core of this prediction system is a Long Short-Term Memory (LSTM) neural network implemented using Keras. The model is trained on historical Bitcoin price data fetched via the yfinance library. It employs a scaled time-series approach to learn patterns in Bitcoin's price movements over a 100-day rolling window. Predictions are generated for unseen data, with the option to forecast future prices (e.g., 10 days ahead). The training pipeline includes data preprocessing using MinMaxScaler for normalization, feature extraction, and splitting into training and testing datasets to ensure model accuracy. Visualization of results is accomplished with Matplotlib, displaying both the actual and predicted Bitcoin prices for a clear comparison.

Database Layer:
Although the current code does not explicitly feature a database, the described architecture could integrate user-specific configurations and historical prediction data using a relational database. This layer could efficiently manage user preferences, forecast history, and other interactions to personalize the experience and allow retrieval of past predictions.

Data Visualization Layer:
Visualization is a key component powered by Matplotlib. Historical trends and moving averages (upto 100-day) are plotted to provide insights into the Bitcoin price's long-term and short-term dynamics. Predictions are plotted against actual values for comparison, and future price forecasts are presented interactively, enhancing decision-making and user engagement.

Key Components:

- Cryptocurrency API Integration: Provides real-time Bitcoin price data, fetching and updating it dynamically to ensure that users are always viewing the latest market information.
- Interactive Dashboard: Displays essential data such as current holdings (if linked to the wallet), prediction summaries, and alerts for significant price fluctuations, giving users quick access to key metrics.

- Prediction Engine: The LSTM-based prediction engine processes user-defined inputs, such as forecasting duration, to generate accurate predictions and ensure actionable insights for users.

- Secure User Authentication: Ensures that users can securely sign up, log in, and manage their accounts with features such as encrypted password storage, account recovery, and 2FA for enhanced security.

- This cohesive architecture ensures the platform is secure, scalable, and efficient, providing a high-quality user experience for cryptocurrency investors and enthusiasts.

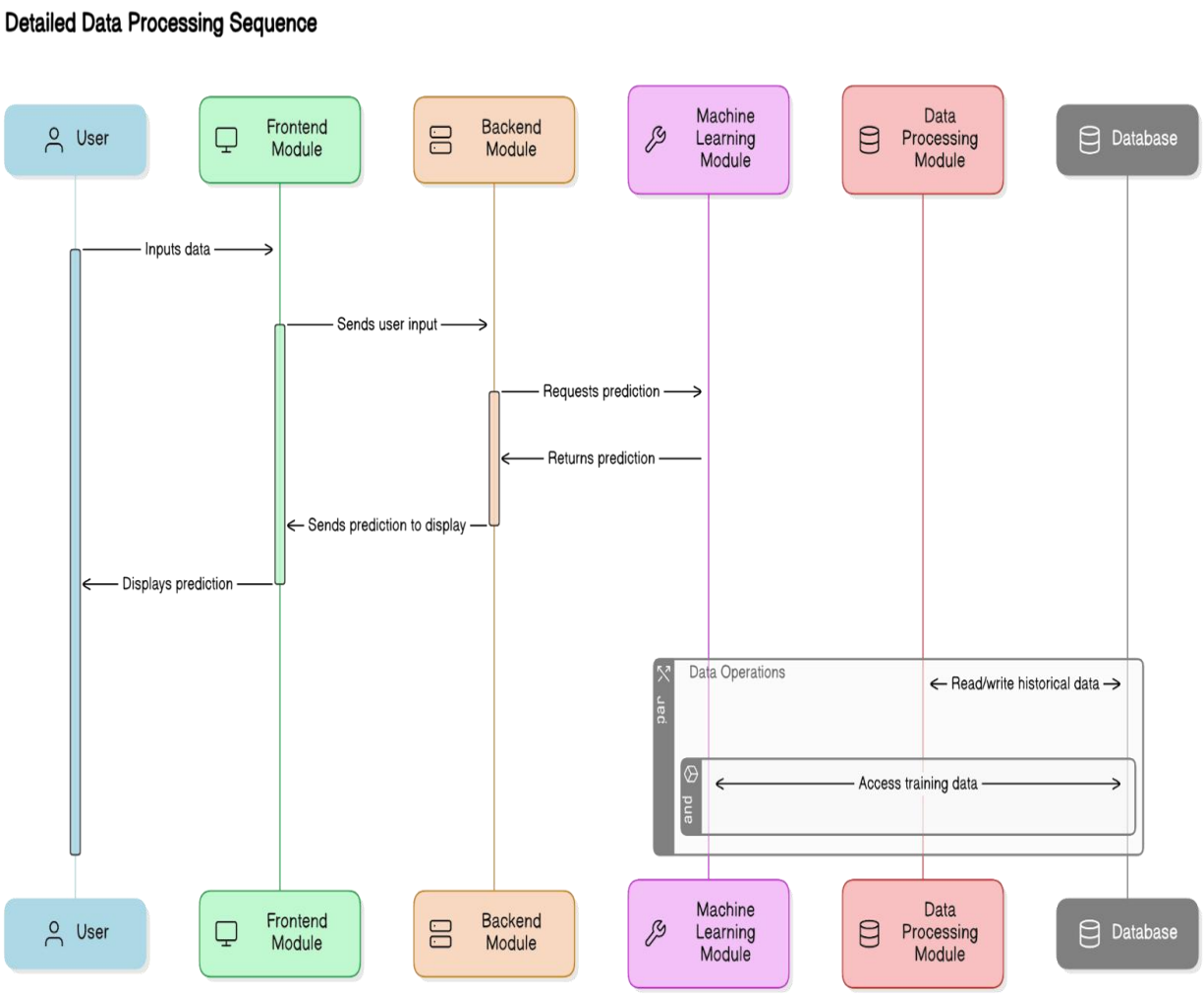


Fig : UML Diagram of Data Processing Sequence

4. IMPLEMENTATION

A4.1 METHODOLOGY

This project involved designing a Bitcoin price prediction system using a Long Short-Term Memory (LSTM) model. The implementation is divided into three primary components: front-end, back-end, and machine learning model integration. Historical price data was obtained using the yfinance library, while real-time price data was fetched from the CoinGecko API.

1. Front-End Development:

HTML, CSS, and JavaScript were used in the front end's implementation. The goal was to develop an intuitive user interface that enables users to view forecasted Bitcoin values, enter parameters, and obtain the actual price of Bitcoin. The following components make up the front-end,

Dashboard: Shows the current, real-time Bitcoin price as well as the anticipated price for a chosen date.Users can choose the parameters for analysis and the prediction window using the input form.

Styling: A responsive and user-friendly layout was created using CSS to ensure smooth interaction. Crucial components including buttons, text fields, and result displays were designed with readability and usability in mind.

2. Back-End Development:

The lightweight Python web framework Flask was used to create the back-end. It serves as a link between the LSTM model and the front-end. The model is triggered for predictions, user requests are processed, and real-time data is retrieved by the back-end.

The back-end's primary features include:

- Routes for APIs: /pricepredict:
  Returns the anticipated price after accepting user-defined inputs
  /auth/login: Logs in the user
  /auth/signup: Signs up the user

- Preprocessing Data: In the back-end, historical price data from 2015 to 2023 was preprocessed. This involved actions including managing missing values, restructuring the data to fit the input format needed by the LSTM model, and scaling the data using MinMaxScaler.

- Front-end integration: The API endpoints that the front-end may query to obtain results were hosted on Flask's server.

3. Extraction of Data

For this project, YFinance Library, Past Price Information has been taken from the 2015–2023 edition of the library.The open, high, low, close, and volume values in the data were essential for LSTM model training.To eliminate discrepancies, the data was cleansed and processed after extraction.

4. LSTM Model for Cryptocurrency Price Prediction

The LSTM model, which works well for time-series forecasting, is the project's central component. The following actions were taken during implementation:
Data Preparation: Training and testing sets of historical price data were separated (e.g., 80% for training, 20% for testing).To enhance model performance, features were scaled from 0 to 1 using MinMaxScaler.The data was prepared in sequences using a sliding window technique, with a predetermined number of prior time steps in each series used to forecast the subsequent value.

Model Architecture: TensorFlow/Keras was used to implement the LSTM model. It has several layers, including LSTM layers with dropout to avoid overfitting, dense layers to forecast the outcome.The Adam optimizer and mean squared error (MSE) as the loss function were used to compile the model.

Training:A batch size of 32 and 50 epochs of historical price data were used to train the model.

Prediction:After training, the model was used to forecast Bitcoin values using user-defined inputs and the testing set.The front-end received the expected values and displayed them.

4.1.1 Front-end Implementation:
Using HTML, CSS, and JavaScript, the front-end of our Bitcoin price prediction system was painstakingly created to offer a smooth and easy user experience. The user interface is made up of two CSS files for uniform application styling and several HTML files, each with a specific function.

1.Figma Design:
Figma was first used to build the Bitcoin Price Predictor's frontend in order to provide an aesthetically pleasing and intuitive prototype. The layout, color schemes, typography, and general structure of the interface could all be visualized with Figma. To guarantee clarity and functionality, important components like the navigation bar, input fields, interactive charts, and buttons like "Get Started" were positioned and designed with purpose. The end user's ease of use was maintained

while a contemporary and polished appearance was prioritized in the design.

2.HTML Structure:

index.html: This is the application's main page and point of entry. With a succinct system overview and navigation choices to investigate the prediction tool or user authentication functions, it provides an appealing user experience.

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF">
        <title>Batwa</title>
        <meta name="viewport",content="width-device-width,initial-scale=1.0">
        <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpP
        <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
    </head>
    <body style="background-color: rgb(239, 239, 239)">
        <nav class="navbar">
            <div class="batwa">
                <div class="logo">
                    <img src="{{ url_for('static', filename='images/batwa.png') }}"  alt="LOGO">
                </div>


                <div>
                    <ul class="name" style="position: absolute; right: 0;">
                        <p><a href="#">Wallet</a></p>
                        <p><a href="/prediction">Prediction</a></p>
                        <p><a href="#">About</a></p>
                        <p><a href="#">Contact</a></p>
                        <p><a href="#">Help</a></p>
                        <a href="#">
                            <div class="name2">
                                <a href="/signup">
                                    <button class="signin-button"> Sign-in</button>
                                </a>
                            </div>
                        </a>
                    </ul>
                </div>
            </div>
        </nav>
        <div class="content">
            <div class="second">
                <h1><b>Buy and Sell</b></h1>
                <h1>Crypto with Batwa</h1>
                <h2>Get started with Batwa today and start buying and selling crypto with ease.</h2>
                <p>Easily manage, store, and trade your digital assets with confidence. Batwa makes navigating the crypto landscape simpl
                <button type="button" class="btn btn-warning"><a href="/login">Get Started</a></button>

            </div>
            <div class="image-container">
                <img src="{{ url_for('static', filename='images/phone2.jpg') }}" alt="Rounded Image with Glow">
            </div>

        </div>
        <footer class="text-center py-4 mt-5">

            &copy;2024 Bitcoin Predictor | All Rights Reserved

        </footer>



    </body>
</html>
```

prediction.html: This page collects user data needed to forecast prices. The kind of currency value (USD, for example) and the number of days in the future that users wish to forecast the price of Bitcoin can be specified. This page's form guarantees quick data entry and clarity.

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF">
        <title>Batwa</title>
        <meta name="viewport",content="width-device-width,initial-scale=1.0">
        <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpP
        <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
    </head>
    <body style="background-color: rgb(239, 239, 239)">
        <nav class="navbar">
            <div class="batwa">
                <div class="logo">
                    <img src="{{ url_for('static', filename='images/batwa.png') }}" alt="LOGO">
                </div >


                <div>
                    <ul class="name" style="position: absolute; right: 0;">
                        <p><a href="/">Home</a></p>
                        <p><a href="#">About</a></p>
                        <p><a href="#">Contact</a></p>
                    </ul>
                </div>
            </div>

        </nav>
        <div class="content">

            <div class="card mx-auto" style="max-width: 600px">
            <div class="card-body">
                <h1 class="text-center card-title">Bitcoin Price Predictor</h1>
                <p class="text-center text-muted">
                Predict future Bitcoin prices with AI-powered analysis.
                </p>
                 <form method="post" action="/pricepredict" class="mt-4">
                 <div class="mb-3">
                     <label for="stock" class="form-label">Stock Ticker</label>
                     <input
                     type="text"
                     class="form-control"
                     id="stock"
                     name="stock"
                     placeholder="e.g., BTC-USD"
                     required
                     title="Enter the stock ticker for the cryptocurrency (e.g., BTC-USD)."
                     />
                 </div>
                 <div class="mb-3">
                     <label for="no_of_days" class="form-label"
                     >Number of Days to Predict</label
                     >
                     <input
                     type="number"
                     class="form-control"
                     id="no_of_days"
                     name="no_of_days"
                     min="1"
                     max="100"
                     value="10"
                     required
                     title="Choose the number of days you want to predict (1-100)."
                     />
                 </div>
                 <button type="submit" class="btn btn-primary w-100">Predict</button>
                 </form>
```

```
        </div>
    </div>
    <footer class="text-center py-4 mt-5">
        <div>
            <p>&copy; 2024 Bitcoin Predictor | All Rights Reserved</p>
        </div>
    </footer>



</body>
</html>
```

Login.html and signup.html: It help with user authentication. The login.html page allows registered users to log in securely, while the signup.html page provides a clear registration form for new users. These pages guarantee that system access is controlled and customized.

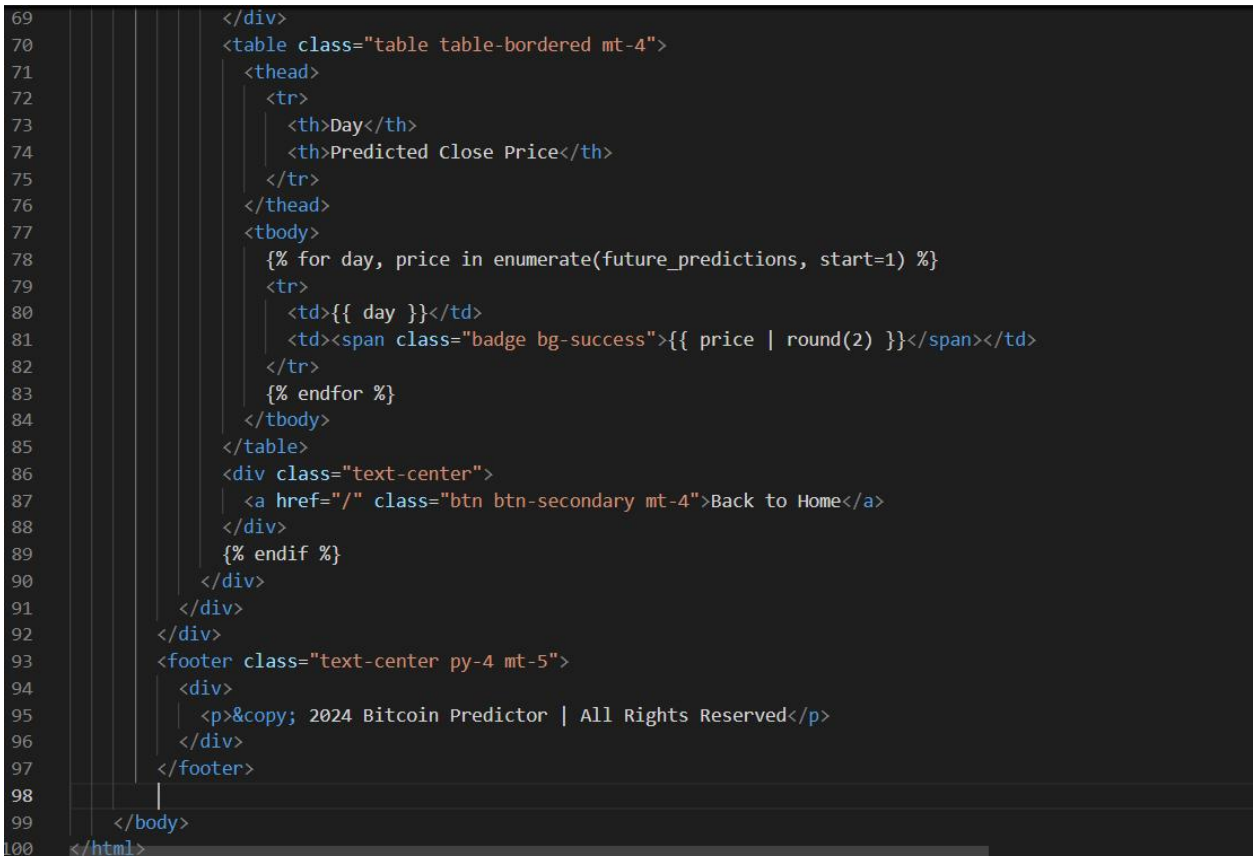
HTML CODE:

```
mplates > login.html > html > body > div.background-container > div.form-container > form > div.mb-3
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Batwa - Login</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style1.css') }}">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body style="background-color: rgb(239, 239, 239);">
    <!-- Navbar -->
    <nav class="navbar">
        <div class="batwa">
            <div class="logo">
                <img src="{{ url_for('static', filename='images/batwa.png') }}" alt="LOGO">
            </div>
            <div>
                <ul class="name" style="position: absolute; right: 0;">
                    <li><a href="/">Home</a></li>
                    <li><a href="/prediction">Features</a></li>
                    <li><a href=#>About</a></li>
                    <li><a href=#>Contact</a></li>
                    <li>
                        <a href="/signup">
                            <button class="signin-button">Sign Up</button>
                        </a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <!-- Login Section -->
    <div class="background-container">
        <div class="form-container">
            <h2>Login to Your Account</h2>
            <form action="/submit_login" method="POST">
                <div class="mb-3">
                    <label for="email" class="form-label">Email</label>
                    <input type="email" class="form-control" id="email" name="email" required>
                </div>
                <div class="mb-3">
                    <label for="password" class="form-label">Password</label>
                    <input type="password" class="form-control" id="password" name="password" required>
                </div>
                <button type="submit" class="btn btn-warning">Login</button>
                <p>Don't have an account? <a href="/signup">Sign up here</a></p>
            </form>
        </div>
    </div>
</body>
</html>


```

result.html: The Bitcoin price prediction's outcome is shown on this page. For comparison, it displays the anticipated price next to the past values. To help the user better comprehend the projections, the page also includes a graph that shows the trend of both the real and predicted Bitcoin prices.

base.html: This file provides reusable elements such as headers, footers, and navigation menus and forms the basis of the application's layout. It makes sure that every page has the same layout and design, which enhances the user experience.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF">
        <title>Batwa</title>
        <meta name="viewport",content="width-device-width,initial-scale=1.0">
        <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpP
        <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
    </head>
    <body style="background-color: rgb(239, 239, 239)">
        <nav class="navbar">
            <div class="batwa">
                <div class="logo">
                    <img src="{{ url_for('static', filename='images/batwa.png') }}" alt="LOGO">
                </div >


                <div>
                    <ul class="name" style="position: absolute; right: 0;">
                        <p><a href="/">Home</a></p>
                        <p><a href="#">About</a></p>
                        <p><a href="#">Contact</a></p>
                    </ul>
                </div>
            </div>

        </nav>
        <div class="content">
          <div class="card mx-auto" style="max-width: 800px">
            <div class="card-body">
              <h2 class="text-center">Results for {{ stock }}</h2>
              {% if error %}
              <div class="alert alert-danger text-center">{{ error }}</div>
              {% else %}
              <div id="plotsCarousel" class="carousel slide" data-bs-ride="carousel">
                <div class="carousel-inner">
                  <div class="carousel-item active">
                    <h3 class="text-center">Original Closing Prices</h3>
                    <img src="{{ original_plot }}" class="d-block w-100 img-fluid" />
                  </div>
                  <div class="carousel-item">
                    <h3 class="text-center">Original vs Predicted Test Data</h3>
                    <img src="{{ predicted_plot }}" class="d-block w-100 img-fluid" />
                  </div>
                  <div class="carousel-item">
                    <h3 class="text-center">Future Predictions</h3>
                    <img src="{{ future_plot }}" class="d-block w-100 img-fluid" />
                  </div>
                </div>
                <button
                  class="carousel-control-prev"
                  type="button"
                  data-bs-target="#plotsCarousel"
                  data-bs-slide="prev"
                >
                  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
                  <span class="visually-hidden">Previous</span>
                </button>
                <button
                  class="carousel-control-next"
                  type="button"
                  data-bs-target="#plotsCarousel"
                  data-bs-slide="next"
                >
                  <span class="carousel-control-next-icon" aria-hidden="true"></span>
                  <span class="visually-hidden">Next</span>
                </button>
```



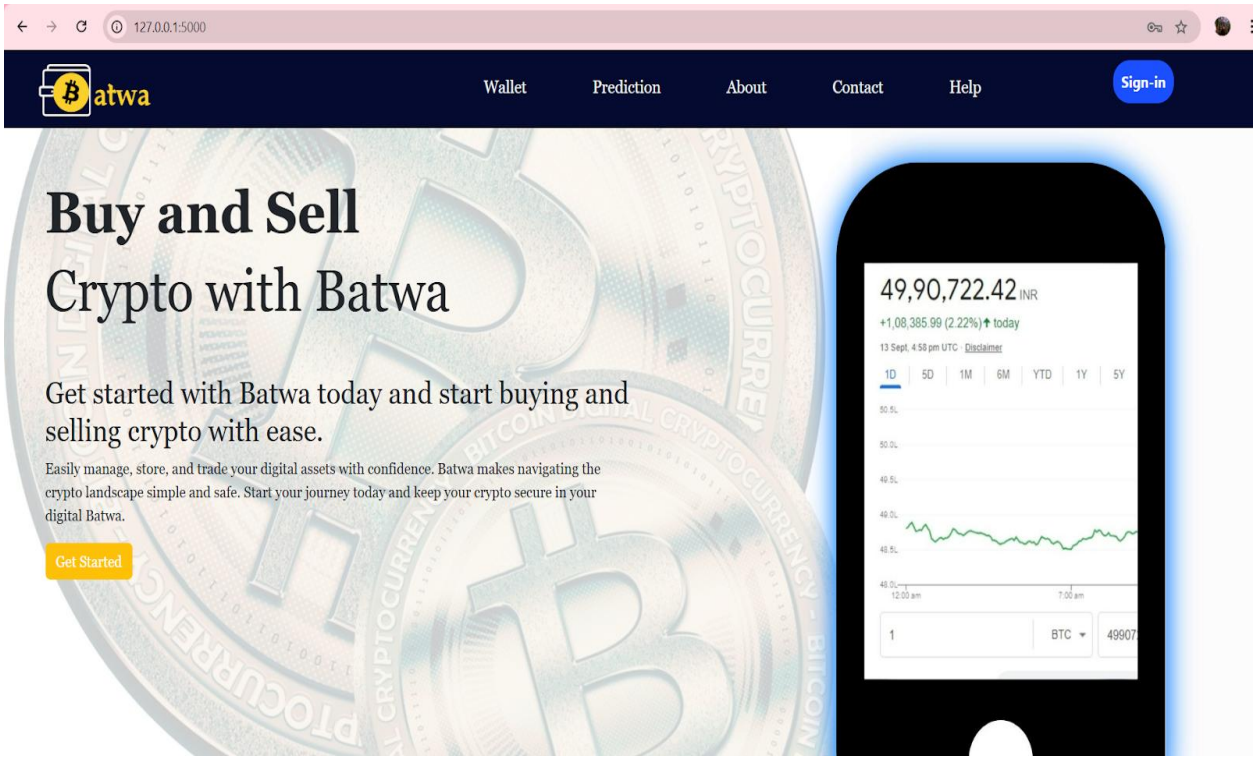
```
                </div>
                <table class="table table-bordered mt-4">
                  <thead>
                    <tr>
                      <th>Day</th>
                      <th>Predicted Close Price</th>
                    </tr>
                  </thead>
                  <tbody>
                    {% for day, price in enumerate(future_predictions, start=1) %}
                    <tr>
                      <td>{{ day }}</td>
                      <td><span class="badge bg-success">{{ price | round(2) }}</span></td>
                    </tr>
                    {% endfor %}
                  </tbody>
                </table>
                <div class="text-center">
                  <a href="/" class="btn btn-secondary mt-4">Back to Home</a>
                </div>
                {% endif %}
              </div>
            </div>
          </div>
          <footer class="text-center py-4 mt-5">
            <div>
              <p>&copy; 2024 Bitcoin Predictor | All Rights Reserved</p>
            </div>
          </footer>

    </body>
</html>
```

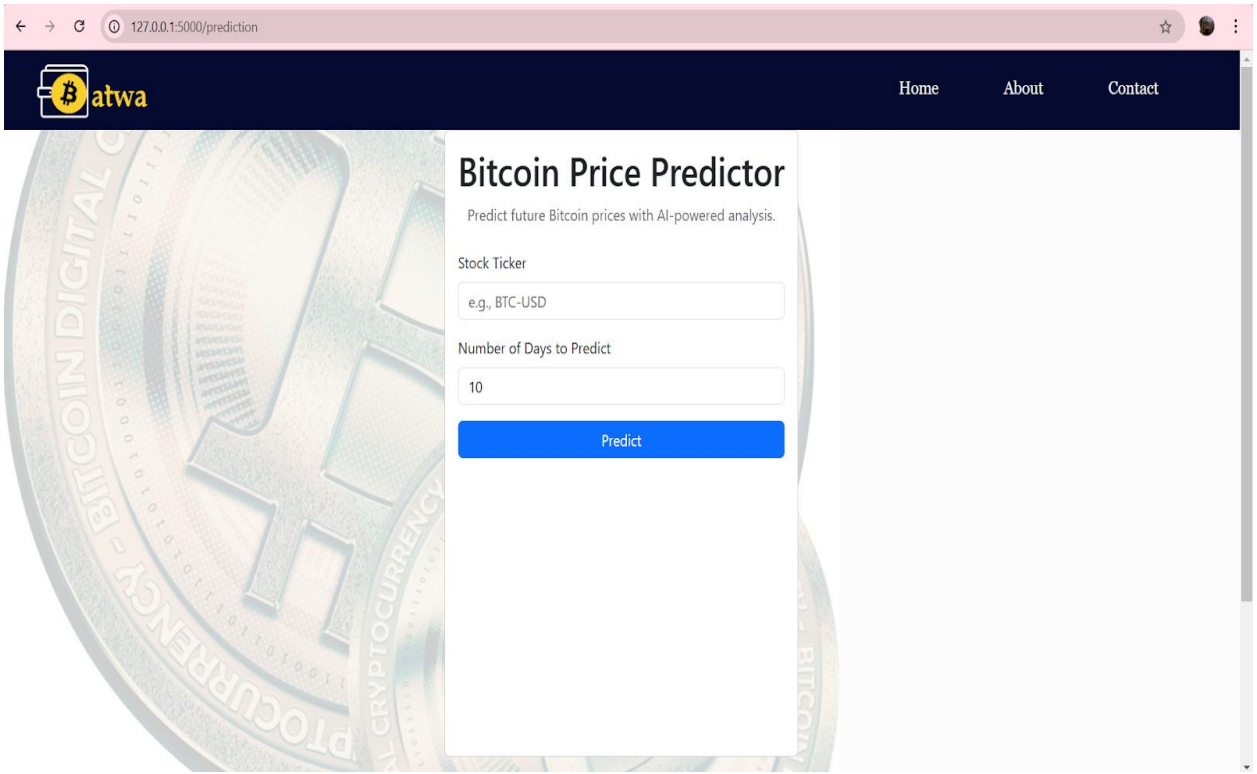
3.Styling with CSS:

The index.html, prediction.html, and result.html pages are linked to the style.css CSS file. It emphasizes on designing a responsive and aesthetically pleasing interface while paying close attention to layout, color palettes, and typography. Buttons, input fields, and charts are examples of elements that are stylized for both visual coherence and ease of use.
The login.html and signup.html pages are the focus of the style1.css CSS file. With clear form layouts and responsive components that improve usability across a range of devices, it guarantees that the user authentication pages have a distinctive yet polished appearance.

Screenshot of the User Interface:

Landing page:





atwa

Home About Contact

Results for BTC-USD

Original Closing Prices

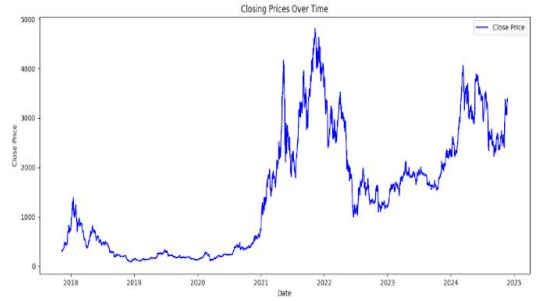
Day	Predicted Close Price
1	94544.21
2	92943.48
3	91599.42
4	90469.71

atwa

Home About Contact

Results for ETH-USD

Original Closing Prices



Day	Predicted Close Price
1	3324.61
2	3311.66
3	3301.52
4	3293.2

4.2.2 Backend Implementation:

Flask is a Python web framework that is used to manage server-side functions and enable communication between the prediction model and the front end in the backend implementation of the Bitcoin Price Predictor. To manage data processing, machine learning, visualization, and security, several libraries are connected. While yfinance is used to retrieve historical and real-time Bitcoin price data from Yahoo Finance, pandas and numpy are utilized for data manipulation and numerical calculations. A pre-trained LSTM model is loaded using Keras for predictions, and the input data is preprocessed and scaled for consistency using scikit-learn's MinMaxScaler.

Matplotlib creates graphs to show trends and predictions; these graphs are then converted into base64 format using the io and base64 packages, allowing for smooth frontend interaction. Flask-HTTPAuth is used to implement authentication, while Flask-Bcrypt and werkzeug.security handle password hashing and verification, ensuring security. Flask-PyMongo integrates MongoDB to offer a reliable database option for storing logs, prediction records, and user data. In order to ensure exact correlation between user actions and financial data, the datetime module is utilized for precise time monitoring. This set of tools guarantees that the backend processes data effectively, makes precise predictions, protects user interactions, and facilitates an engaging and dynamic frontend experience.

```
from flask import Flask, render_template, request, redirect, url_for
import pandas as pd
import numpy as np
import yfinance as yf
from keras.models import load_model
from sklearn.preprocessing import MinMaxScaler
import matplotlib
import matplotlib.pyplot as plt
import io
import base64
from datetime import datetime
from flask import Flask, request, jsonify
from flask_pymongo import PyMongo
from flask_bcrypt import Bcrypt
from flask_httpauth import HTTPBasicAuth
from werkzeug.security import generate_password_hash, check_password_hash

```

4.2.3 LSTM Model Implementation:

The LSTM (Long Short-Term Memory) model was chosen for predicting Bitcoin prices due to its effectiveness in handling sequential and time-series data. Bitcoin prices exhibit high volatility and depend on past trends and temporal patterns, making a model like LSTM ideal for this task. Unlike traditional machine learning models, which treat data as independent observations, LSTM captures long-term dependencies and learns patterns over time, allowing it to adapt to the non-linear and dynamic nature of cryptocurrency markets. Its ability to remember important information from past sequences while forgetting irrelevant data makes it particularly suitable for predicting prices in volatile markets like Bitcoin.

Data processing:

Data for training and testing the LSTM model was fetched using the yfinance library, which provided historical Bitcoin price data from Yahoo Finance. This data was essential for creating a reliable dataset containing price trends over time.

```
end = datetime.now()
start = datetime(end.year-15, end.month, end.day)
stock = 'BTC-USD'
stock_data = yf.download(stock, start=start, end=end)
```

```
[**********100%**********]  1 of 1 completed
```

```
stock_data.head()
```

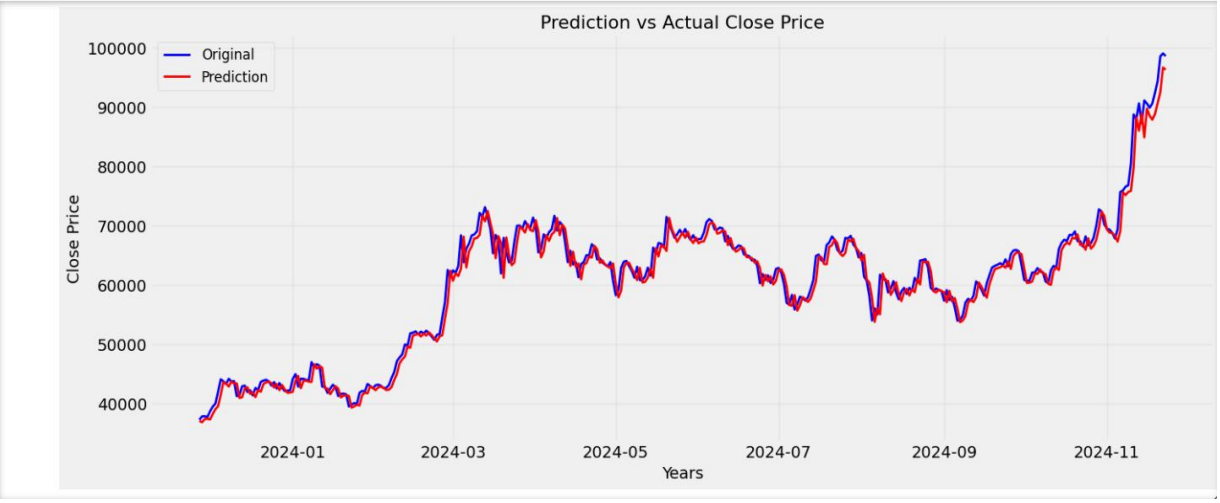
Price	Adj Close	Close	High	Low	Open	Volume
Ticker	BTC-USD	BTC-USD	BTC-USD	BTC-USD	BTC-USD	BTC-USD
Date						
2014-09-17 00:00:00+00:00	457.334015	457.334015	468.174011	452.421997	465.864014	21056800
2014-09-18 00:00:00+00:00	424.440002	424.440002	456.859985	413.104004	456.859985	34483200
2014-09-19 00:00:00+00:00	394.795990	394.795990	427.834991	384.532013	424.102997	37919700
2014-09-20 00:00:00+00:00	408.903992	408.903992	423.295990	389.882996	394.673004	36863600
2014-09-21 00:00:00+00:00	398.821014	398.821014	412.425995	393.181000	408.084991	26580100

Before feeding the data into the LSTM model, it was preprocessed using the MinMaxScaler from scikit-learn. This step scaled the data to a range between 0 and 1, ensuring that the model training was consistent and the influence of large numerical values was minimized. Proper scaling is critical for LSTM models to converge effectively and perform optimally.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(closing_price[['Close']].dropna())
```

Model training:

Once the data was preprocessed, it was divided into training and testing sets to evaluate the model's performance. The training set was used to teach the LSTM model the underlying patterns in Bitcoin price movements, while the testing set was reserved to assess how well the model could predict future prices based on unseen data. By implementing data from
yfinance and processing it with MinMaxScaler, the LSTM model was trained to deliver accurate and reliable predictions, making it a powerful tool for forecasting Bitcoin price trends.
Testing for accuracy:



This graph shows the comparison between the predicted and actual Bitcoin close prices over time. The blue line represents the actual close prices, while the red line shows the predicted values generated by the LSTM model. The close price of Bitcoin fluctuates throughout the year, and the model does a relatively good job of predicting these fluctuations, especially in the later months where the trends align more closely. The slight divergence between the two lines at certain points may be due to inherent volatility in the Bitcoin market, which is challenging to model with absolute precision. However, the overall performance of the model is evident in how well it captures the trends and movement of Bitcoin prices.

5. STANDARD ADOPTED

5.1 DESIGN STANDARD

User-Centric Design:

- The platform adheres to user-centered design principles, focusing on intuitive navigation, accessibility, and responsiveness.
- Wireframing and Prototyping: Initial designs were created using Canva and Figma, aligning with industry-standard practices for visual planning and user experience (UX) design.
- Responsive Design: The front end was built to comply with WCAG 2.1 guidelines, ensuring accessibility across devices and for users with disabilities, including keyboard navigation and compatibility with screen readers.

System Architecture:

- Modular architecture was implemented to support scalability and maintainability.
- Components such as front-end interfaces, back-end APIs, machine learning modules, and databases are clearly delineated, following the Model-View-Controller (MVC) paradigm.

Data Visualization:

- Graphical components adhere to standards like Edward Tufte's principles of data visualization, ensuring clarity and avoiding unnecessary clutter.
- Interactive charts were designed using Matplotlib to provide an intuitive user experience.

5.2 Coding Standards

Front-End Code:

- HTML/CSS: The front-end adheres to W3C Standards, ensuring compatibility and semantic correctness.
- JavaScript: Follows ECMAScript (ES6) standards for clean and modern scripting, including modular code organization for scalability.
- Version Control: Front-end assets are version-controlled using Git, following branching strategies like Gitflow for development and deployment.

Back-End Code:

- Frameworks: The use of Flask aligns with PEP 8 coding conventions for Python, ensuring readability and consistency.
- API Development: API endpoints were designed following RESTful architecture standards, using appropriate HTTP methods (GET, POST) and status codes.
- Security: Passwords are hashed using bcrypt, and authentication mechanisms follow OAuth 2.0 protocols for secure access control.

Machine Learning Module:

- Model Development: The LSTM neural network is implemented using TensorFlow/Keras, adhering to their official API guidelines.
- Data Preprocessing: Data cleaning and normalization are performed using scikit-learn, following standard practices for reproducibility and accuracy.
- Versioning: Models and training pipelines are version-controlled, with clear documentation of hyperparameters and training data.

5.3 Testing Standards

Front-End Testing:

- Cross-Browser Compatibility: Tested across multiple browsers (Chrome, Firefox, Safari) and devices (desktop, tablet, mobile) to ensure uniform performance.
- Responsiveness Testing: Conducted using tools like Chrome DevTools and frameworks like Selenium to verify adaptive layouts.

Back-End Testing:

- Unit Testing: API endpoints and backend logic were tested using pytest or unittest, ensuring all routes return the correct responses.
- Integration Testing: Comprehensive tests verified the seamless interaction between the front-end, back-end, and machine learning components.
- Security Testing: Authentication routes were tested for vulnerabilities using OWASP standards.

Machine Learning Testing:

- Model Evaluation: Accuracy and reliability were validated using cross-validation techniques and metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).
- Stress Testing: The model was tested for robustness with varying input sizes and data anomalies.

User Acceptance Testing (UAT):

- Feedback was gathered from a sample group of users, ensuring the platform met real-world expectations and provided a satisfactory experience.

6. CONCLUSION

This project demonstrates an innovative approach to cryptocurrency forecasting by seamlessly blending advanced predictive analytics with an intuitive user experience. The live cryptocurrency value predictor website caters to traders, investors, and enthusiasts, providing real-time price tracking and customizable forecasting. Its responsive design, secure login system, and user-friendly interface make it accessible to users of all skill levels, ensuring a smooth and engaging experience across devices.

By integrating cutting-edge machine learning algorithms with practical, actionable insights, this project equips users to navigate the volatile cryptocurrency market with confidence. The customizable prediction features and visually appealing interface position the platform as a valuable tool for making informed investment decisions in the dynamic world of digital assets.

7. FUTURE SCOPE

- Quantum-Resistant and Adaptive Security: The wallet's multi-layer encryption, designed to withstand quantum computing threats, can evolve to incorporate cutting-edge cryptographic techniques, setting a benchmark for quantum-safe financial systems across various digital assets.

- Enhanced AI and Privacy Features: The AI-powered threat detection system can be refined to tackle emerging cyber risks proactively, while privacy protocols like Zero-Knowledge Proofs and Ring Signatures can be optimized for broader adoption and enhanced confidentiality.

- Interoperability and Scalability: Expanding support for multiple cryptocurrencies and integrating with other blockchain platforms will make the wallet a versatile tool. Scalability enhancements can position it for enterprise-level applications, such as secure payment processing and institutional cryptocurrency management.

- Influence on Industry Standards and Emerging Tech Integration: By addressing current and future challenges, the project could shape industry practices. Future integrations with technologies like IoT and smart contracts would further expand its applicability, reinforcing its role in next-generation secure financial ecosystems.

REFERNCES

1. https://jfin-swufe.springeropen.com/articles/10.1186/s40854-024-00643-1
2. https://ieeexplore.ieee.org/document/9697202/
3. https://www.researchgate.net/publication/380481812_Forecasting_Bitcoin_Prices_A_Comparative_Study_of_Machine_Learning_and_Deep_Learning_Algorithms
4. https://www.sciencedirect.com/science/article/pii/S1057521923005719
5. https://www.sciencedirect.com/science/article/pii/S0952197624001490
6. https://arxiv.org/abs/2405.08089
7. https://www.mdpi.com/1911-8074/16/1/51
8. https://pmc.ncbi.nlm.nih.gov/articles/PMC10981097/
9. https://www.researchgate.net/publication/347155658_Factors_affecting_Cryptocurrency_Prices_Evidence_from_Ethereum
10. https://www.ijert.org/cryptocurrency-prediction-using-machine-learning

INDIVIDUAL CONTRIBUTION REPORT:

AI POWERED CRYPTO PRICE PREDICTOR

AARYAK PRASAD

2105171

Abstract: This project delivers a live cryptocurrency prediction platform offering accurate, real-time price forecasts with customizable durations up to 100 days. Built on an LSTM model, the platform ensures seamless data processing and secure user authentication via Flask, while a responsive HTML, CSS, and JavaScript frontend provides an intuitive user experience. Designed for accessibility, it empowers users to navigate the dynamic cryptocurrency market with confidence.

Individual contribution for project presentation and demonstration: I created visualizations for the cryptocurrency price predictions using Matplotlib, and collaborated on designing the backend authentication system using MongoDB to ensure secure data handling and presentation.

Full Signature of Supervisor: Full signature of the student:

…….. ……..

School of Computer Engineering, KIIT, BBSR

INDIVIDUAL CONTRIBUTION REPORT:

AI POWERED CRYPTO PRICE PREDICTOR

SHIVLI SINGH

2105237

Abstract: This project delivers a live cryptocurrency prediction platform offering accurate, real-time price forecasts with customizable durations up to 100 days. Built on an LSTM model, the platform ensures seamless data processing and secure user authentication via Flask, while a responsive HTML, CSS, and JavaScript frontend provides an intuitive user experience. Designed for accessibility, it empowers users to navigate the dynamic cryptocurrency market with confidence.

Individual contribution for project presentation and demonstration: I fetched live cryptocurrency data from Yahoo Finance using yfinance, cleaned and transformed it for accurate model input, and integrated the data pipeline with the machine learning model to ensure seamless data processing.

Full Signature of Supervisor: Full signature of the student:

…….. ……..

School of Computer Engineering, KIIT, BBSR

INDIVIDUAL CONTRIBUTION REPORT:

AI POWERED CRYPTO PRICE PREDICTOR

ABHIPSHA DAS

21052725

Abstract: This project delivers a live cryptocurrency prediction platform offering accurate, real-time price forecasts with customizable durations up to 100 days. Built on an LSTM model, the platform ensures seamless data processing and secure user authentication via Flask, while a responsive HTML, CSS, and JavaScript frontend provides an intuitive user experience. Designed for accessibility, it empowers users to navigate the dynamic cryptocurrency market with confidence.

Individual contribution for project presentation and demonstration: I designed the dashboard, and other landing pages using Figma, then translated these designs into responsive web pages using HTML, CSS, and JavaScript enhancing user experience through a functional interface.

Full Signature of Supervisor: Full signature of the student:

…….. ……..

School of Computer Engineering, KIIT, BBSR

INDIVIDUAL CONTRIBUTION REPORT:

AI POWERED CRYPTO PRICE PREDICTOR

SWAYAM YADAV

21052956

Abstract: This project delivers a live cryptocurrency prediction platform offering accurate, real-time price forecasts with customizable durations up to 100 days. Built on an LSTM model, the platform ensures seamless data processing and secure user authentication via Flask, while a responsive HTML, CSS, and JavaScript frontend provides an intuitive user experience. Designed for accessibility, it empowers users to navigate the dynamic cryptocurrency market with confidence.

Individual contribution for project presentation and demonstration: I developed and trained the LSTM model for cryptocurrency price prediction, fine-tuned the model parameters for optimal performance, and evaluated the model's accuracy and prediction results.

Full Signature of Supervisor: Full signature of the student:

……. ……..

School of Computer Engineering, KIIT, BBSR

INDIVIDUAL CONTRIBUTION REPORT:

AI POWERED CRYPTO PRICE PREDICTOR

NEETU DEY

21053298

Abstract: This project delivers a live cryptocurrency prediction platform offering accurate, real-time price forecasts with customizable durations up to 100 days. Built on an LSTM model, the platform ensures seamless data processing and secure user authentication via Flask, while a responsive HTML, CSS, and JavaScript frontend provides an intuitive user experience. Designed for accessibility, it empowers users to navigate the dynamic cryptocurrency market with confidence.

Individual contribution for project presentation and demonstration: I worked on backend integration using Flask to manage communication between the LSTM model and frontend, ensuring smooth data exchange and implementing efficient server-side logic for the platform.

Full Signature of Supervisor: Full signature of the student:

……. ……..

School of Computer Engineering, KIIT, BBSR

INDIVIDUAL CONTRIBUTION REPORT:

AI POWERED CRYPTO PRICE PREDICTOR

AAYUSHMA GAUTAM

21053475

Abstract: This project delivers a live cryptocurrency prediction platform offering accurate, real-time price forecasts with customizable durations up to 100 days. Built on an LSTM model, the platform ensures seamless data processing and secure user authentication via Flask, while a responsive HTML, CSS, and JavaScript frontend provides an intuitive user experience. Designed for accessibility, it empowers users to navigate the dynamic cryptocurrency market with confidence.

Individual contribution for project presentation and demonstration: I developed and designed the login and sign-up pages, collaborating on the backend authentication system using MongoDB to securely store user data, including usernames, emails, and hashed passwords.

Full Signature of Supervisor: Full signature of the student:

…….. ……..

School of Computer Engineering, KIIT, BBSR