

“ZCRYPTO”

A Project Report

Submitted by

Nilesh Vijay

20BCON169

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

AT



JECRC UNIVERSITY, JAIPUR

May 2023

Candidate's Declaration

I, **Nilesh Vijay**, bearing roll number **20BCON169**, hereby declare that I worked which is being presented in the Project, entitled “**ZCRYPTO**” in partial fulfilment for award of the Degree of “**Bachelor of Technology**” in department of **Computer Science Engineering** is submitted to the Department Computer Science & Engineering, JECRC University is a record of Project work carried under the Guidance of Dr Shashi Sharma, Department Computer Science & Engineering.

Computer Science

Nilesh Vijay

20BCON169

CERTIFICATE

Certified that the Project Report entitled “**ZCRYPTO**” submitted by **Nilesh Vijay** bearing roll no. 20BCON169. In partial fulfilment of the requirements for the award of the degree of Bachelor of Technology of JECRC University, Jaipur is a record of the student’s own work carried out under my supervision and guidance. To the best of our knowledge, this Minor Project work has not been submitted to JECRC University or any other university for the award of the degree. It is further understood that by this certificate the undersigned does not endorse or approve of any statement made, opinion expressed or conclusion drawn therein but approve Minor Project for the purpose for which it is submitted.

Dr. Shashi Sharma

Acknowledgements

Many people have supported me, in different ways, during the work with the thesis. I'd like to thank my mentor Dr. Shashi Sharma & HOD Dr. Bhawna Sharma for their kind and active support and valuable guidance during the work process. My family has as always offered me their unconditional support, thank you! I have taken efforts in the project presentation. However, it would not have been possible without the kind support and many individuals and organizations. I would like to extend my sincere thanks to each and every member related to JECRC University.

Nilesh Vijay

20BCON169

Abstract

Cryptocurrencies have emerged as an important financial system. They work on the technology of Distributed Ledger which is called Blockchain. Cryptocurrencies are volatile in nature as their prices change rapidly. That's why we thought of creating a platform to track the performance of cryptocurrencies. The platform will track the performance of cryptocurrencies and provide information on changes in the value of cryptocurrencies. We have used some of the most popular programming languages such as JavaScript, HTML and CSS to build the platform, and along with that, we have also used an API to get the cryptocurrency data. The platform we have built gives us insight into the performance of cryptocurrencies with an attractive UL. The cryptocurrency data we receive is updated every day, and includes price changes over a period of 24 hours and 1 week. This also includes the valuation of cryptocurrencies. The main objective of creating this platform is to provide easy access to crypto insights to the user. We have built our UI in such a way that it makes it easy for the user to explore every page without any problem.

Keywords: Cryptocurrency, Bitcoin, Encrypted, Currency, Bitpay, Exchange Rates, PrincipalComponents Analysis, Blockchain

Table Of Content

Page

CANDIDATE DECLARATION.....	2	
CERTIFICATE.....	4	
ACKNOWLEDGEMENTS	5	
ABSTRACT.....	6	
CHAPTER 1 INTRODUCTION		
1.1 Background.....	9	
1.2 Statement of the problem	10	
1.3 objective of the study	10	
1.4 Scope.....	11	
CHAPTER 2 LITERATURE SURVEY		
2.1 Advantages.....	12	
2.2 Disadvantage.....	13	
CHAPTER 3 SOFTWARE REQUIRMENT ANALYSIS		
3.1 Problem Statement	14	
3.2 Modules.....	14	
CHAPTER 4 SOFTWARE DESIGN		15
CHAPTER 5 SOFTWARE AND HARDWARE REQUIREMENTS		
5.1.1 Hardware Specification.....	17	
5.1.2 Software Specification	18	

CHAPTER 6 CODE TEMPLATES

6.1 Home Screen	19
6.2 Exchanges Web Screen	20
6.3 Coins Web Page	21
6.4 Real Time Price Tracker	22
6.5Range Screen	23

CHAPTER 7 TESTING

7.1 Stress testing	25
7.2 Unit Testing	26
7.2.1 Unit testing method.....	27
7.2.2 Unit testing tasks.....	28
7.3 What is dependency testing?.....	28

CHAPTER 8 RESEARCH METHODOLOGY 29

CHAPTER 9 CONCLUSION30

CHAPTER 10 FURTHER ENHANCEMENTS32

CHAPTER 11 REFERENCE34

LIST OF FIGURES

Fig No.	Title	Page No.
4.0	SOFTWARE DESIGN	15
6.1	HOME SCREEN	19
6.2	EXCHANGES WEB SCREEN	20
6.3	COINS PAGE SCREEN	21
6.4	REAL TIME PRICE TRECKER	22
6.5	RANGE SCREEN	23

1. INTRODUCTION

1.1 Background

The word "crypto currency" is used to refer to a specific type of digital currency that is intended to operate as a medium of exchange through the utilization of computer networks, as opposed to depending on centralized authority such as governments or financial organizations. The word "crypto currency" is used to refer to a specific type of digital currency that is intended to operate as a medium of exchange through the utilization of computer networks, as opposed to depending on centralized authority such as governments or financial organizations.

The individual's possession of a certain currency A digital ledger is a specific kind of computerized database that makes use of robust encryption to protect transaction records, regulate money manufacturing, and authenticate the transfer of coin ownership. The fact that they are referred to as crypto currencies does not change the reality that they are not taken into consideration. Crypto currencies are regarded as a category of asset despite the fact that they are also categorized as commodities, securities, and currencies Crypto currencies are not issued by any centralized authority, nor do they exist in any form that can be touched (like paper currency). In most cases, crypto currencies make use of decentralized checks rather than the digital money issued by central banks (CBDCs).

In 2009, bitcoin was the first decentralized crypto currency to be made available to the public in the form of open-source software. As of March 2022, there are roughly 9,000 new crypto currencies on the market. There are about 70 market segments with market capitalizations greater than \$1 billion. Decentralized coins are created collectively by the whole crypto currency network.

1.2 Statement of the problem

There are public discussions of cryptocurrencies worldwide because of frequent changes of prices time to time. When it comes to crypto currencies, investors routinely spend a lot of time searching for latest coins, the one with most exciting prospecta and the ones which are undervalued. Savvy investors also look to different cryptocurrency exchanges, applications, and services to maximize their investment potential. There is a need for investors to compare currency prices on multiple exchanges, at the same allowing them to determine the best options for buying or selling. Investors will have a significant impact on their decisions, timing and success of their investments. Information on individual cryptocurrencies also includes historical charts in various time frames.

1.3 Objective of the study

The objective of a crypto tracking app built with React.js, HTML, and CSS can vary depending on the specific requirements and goals of the application. However, here are some common objectives that such an app might aim to achieve:

Real-time Price Tracking: The app may provide real-time updates on cryptocurrency prices, allowing users to monitor the market and track the value of their investments. The objective is to provide accurate and timely information to users.

Market Analysis: The app may provide charts, graphs, and other visualizations to help users analyze crypto market trends, historical data, and price movements. The objective is to offer insights and tools for users to make informed decisions based on market analysis.

Alerts and Notifications: The app may include features to set up alerts or notifications for specific price thresholds or market events. The objective is to keep users informed about important changes in the crypto market.

User-friendly Interface: The app may prioritize a clean and intuitive user interface, making it easy for users to navigate, view data, and perform actions. The objective is to create a seamless and enjoyable user experience.

These are some common objectives that a crypto tracking app built with React.js, HTML, and CSS may aim to accomplish. The specific objectives can vary depending on the project requirements and the target audience of the application.

1.4 Scope

The scope of a study on a crypto web app can vary depending on the specific objectives and research questions. However, here are some general aspects that you may consider including in the scope of your study:

Overview of the crypto web app: Provide a comprehensive introduction to the crypto web app you are studying. Explain its purpose, functionality, and key features.

Technology stack: Describe the technology stack used to develop the web app, including programming languages, frameworks, databases, and any other relevant components.

Security analysis: Evaluate the security measures implemented in the crypto web app. This may involve analyzing authentication mechanisms, encryption techniques, secure storage of user data, protection against common vulnerabilities (e.g., XSS, CSRF), and adherence to best practices.

User experience evaluation: Assess the user experience of the crypto web app. This can involve conducting user tests, surveys, or interviews to gather feedback on usability, interface design, navigation, and overall satisfaction.

Performance analysis: Measure the performance of the crypto web app, including factors such as response time, scalability, and resource utilization. Identify any bottlenecks or areas for optimization.

Transaction processing: Investigate the efficiency and reliability of the crypto web app's transaction processing. This may involve analyzing transaction speeds, confirmation times, transaction fees, and any other relevant metrics.

Regulatory compliance: Examine the extent to which the crypto web app complies with applicable regulations and legal frameworks related to cryptocurrencies, data protection, financial services, and any other relevant areas.

Market analysis: Conduct a market analysis of the crypto web app, including its target audience, competitors, market trends, and potential for growth or disruption. This analysis may involve collecting and analyzing data from various sources, such as user surveys, market reports, and competitor analysis.

Recommendations and improvements: Based on your findings, provide recommendations for improving the crypto web app in terms of security, user experience, performance, compliance, and any other relevant areas. Suggest potential future developments or features that could enhance the app's value and competitiveness.

Remember that the scope of your study should be aligned with the available resources, time constraints, and the specific goals of your research. It's important to define clear research objectives and focus on areas that are most relevant and impactful for your study.

2. LITERATURE SURVEY

2.1 Advantages:

Crypto web apps offer several advantages over traditional applications, particularly in the context of cryptocurrency and blockchain technology. Here are some key advantages of crypto web apps:

Decentralization: Crypto web apps leverage blockchain technology, which is inherently decentralized. This means that data and transactions are not controlled by a central authority, such as a government or a financial institution. Decentralization provides transparency, immutability, and resilience to censorship, making crypto web apps more trustworthy and resistant to single points of failure.

Security: Crypto web apps employ cryptographic techniques to ensure the security of transactions and user data. The use of public-key cryptography and digital signatures helps verify the authenticity of transactions and user identities, reducing the risk of fraud and unauthorized access. Additionally, the decentralized nature of blockchain technology makes it difficult for malicious actors to compromise the entire system.

Financial Inclusion: Crypto web apps have the potential to promote financial inclusion by providing access to financial services to individuals who are unbanked or underbanked. Users can manage their digital assets, send and receive payments, and participate in decentralized finance (DeFi) applications without relying on traditional financial intermediaries. This can empower individuals in regions with limited access to banking services or facing economic instability.

Transparency: The use of blockchain technology in crypto web apps ensures transparency and accountability. All transactions and data stored on the blockchain are visible to all participants, creating a transparent and auditable record. This transparency can help prevent fraud and corruption, enhance trust among users, and facilitate regulatory compliance.

Global Accessibility: Crypto web apps enable borderless transactions and global accessibility. Users can transact with anyone, anywhere in the world, without being limited by geographical boundaries or traditional banking hours. This fosters a global economy and allows for efficient cross-border transactions, reducing the need for intermediaries and associated costs.

Programmability: Crypto web apps can leverage smart contracts, which are self-executing contracts with predefined rules encoded on the blockchain. Smart contracts enable automation and programmability, eliminating the need for intermediaries and reducing transaction costs. They enable the creation of decentralized applications (DApps) with complex functionalities, such as decentralized exchanges, lending platforms, and governance systems.

Ownership of Digital Assets: Crypto web apps allow users to have full ownership and control over their digital assets. Users hold the private keys to their wallets, providing them with complete control over their funds. This is in contrast to traditional financial systems where individuals rely on intermediaries to hold and manage their assets.

Innovation and Disruption: Crypto web apps are at the forefront of technological innovation and have the potential to disrupt traditional industries, such as finance, supply chain, and digital identity. The programmability and flexibility of blockchain technology open up new possibilities for building decentralized applications and exploring novel business models.

It's important to note that while crypto web apps offer these advantages, they also come with their own challenges and considerations, such as scalability, regulatory frameworks, and user adoption

2.2 Disadvantage:

While crypto web apps have several advantages, they also come with certain disadvantages and challenges. Here are some common disadvantages of crypto web apps:

Security Risks: While crypto web apps employ cryptographic techniques to enhance security, they are still vulnerable to various security risks. Hackers may target crypto web apps to exploit vulnerabilities, steal private keys, or gain unauthorized access to user funds. Additionally, if users do not properly secure their wallets or follow best security practices, they may become victims of phishing attacks, malware, or social engineering.

Scalability: Blockchain technology, which underlies many crypto web apps, often faces scalability challenges. As the number of users and transactions increases, the capacity of the blockchain network may become strained, resulting in slower transaction processing times and higher fees. This scalability issue needs to be addressed for crypto web apps to handle large-scale adoption and support high transaction volumes effectively.

Energy Consumption: Some cryptocurrencies, such as Bitcoin, rely on energy-intensive consensus mechanisms like proof-of-work. The mining process requires substantial computational power, leading to significant energy consumption. Critics argue that the energy footprint of cryptocurrencies is environmentally unsustainable and contributes to carbon emission.

3. SOFTWARE REQUIRMENT ANALYSIS

3.1 Problem Statement:

"The crypto web app currently faces significant challenges and shortcomings that hinder its optimal functionality and user satisfaction. Users are experiencing a lack of essential features, resulting in a subpar user experience and limiting the app's potential for broader adoption. Moreover, the app's existing security measures exhibit vulnerabilities, exposing users to potential risks such as unauthorized access and theft of funds. Additionally, the app's compliance with relevant regulatory frameworks is inadequate, creating uncertainties and potential legal repercussions.

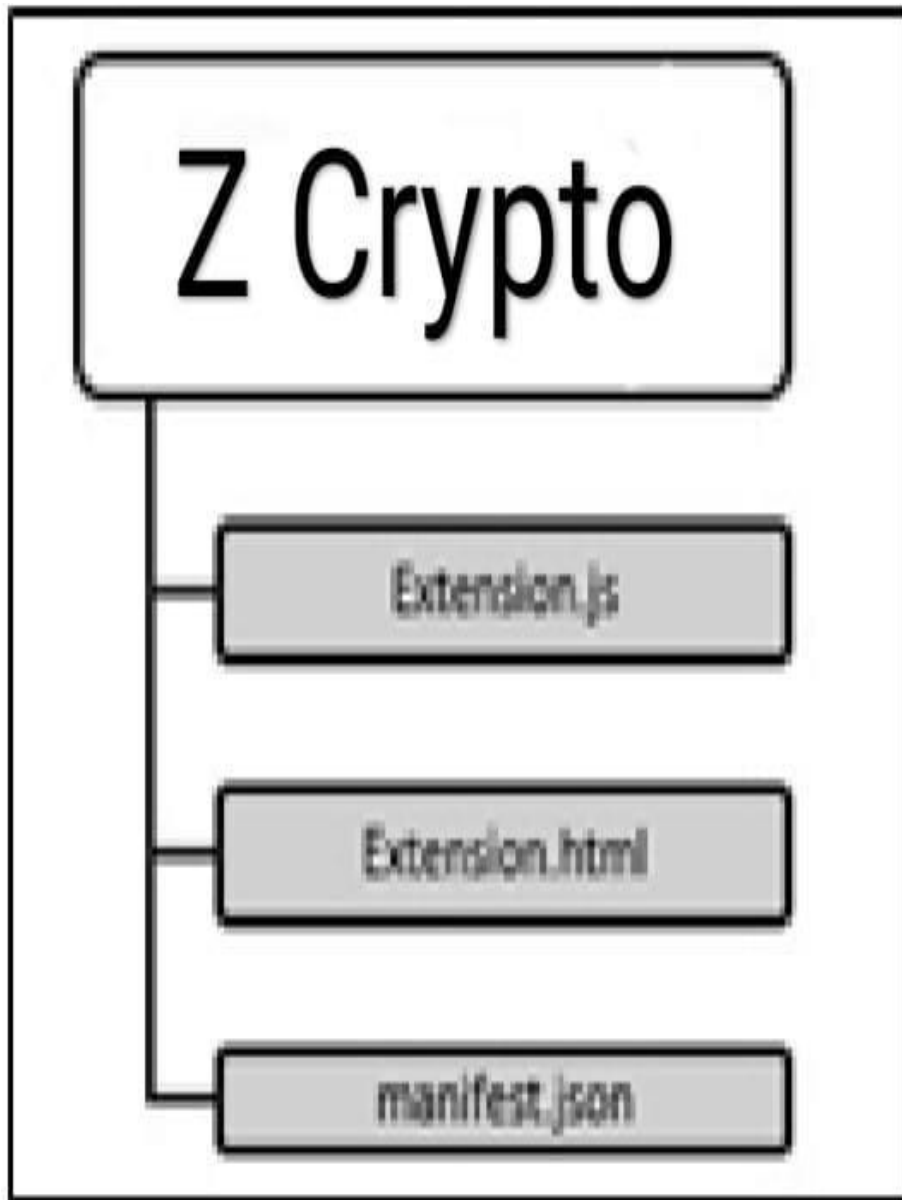
Therefore, there is an urgent need to address these pressing issues and improve the crypto web app. By incorporating crucial features and enhancing user experience, the app can better meet the needs and expectations of its user base. Strengthening security measures is crucial to safeguarding user assets and data, instilling confidence and trust among users. Furthermore, ensuring compliance with applicable regulations is essential for establishing a solid foundation in the evolving landscape of cryptocurrency and regulatory frameworks.

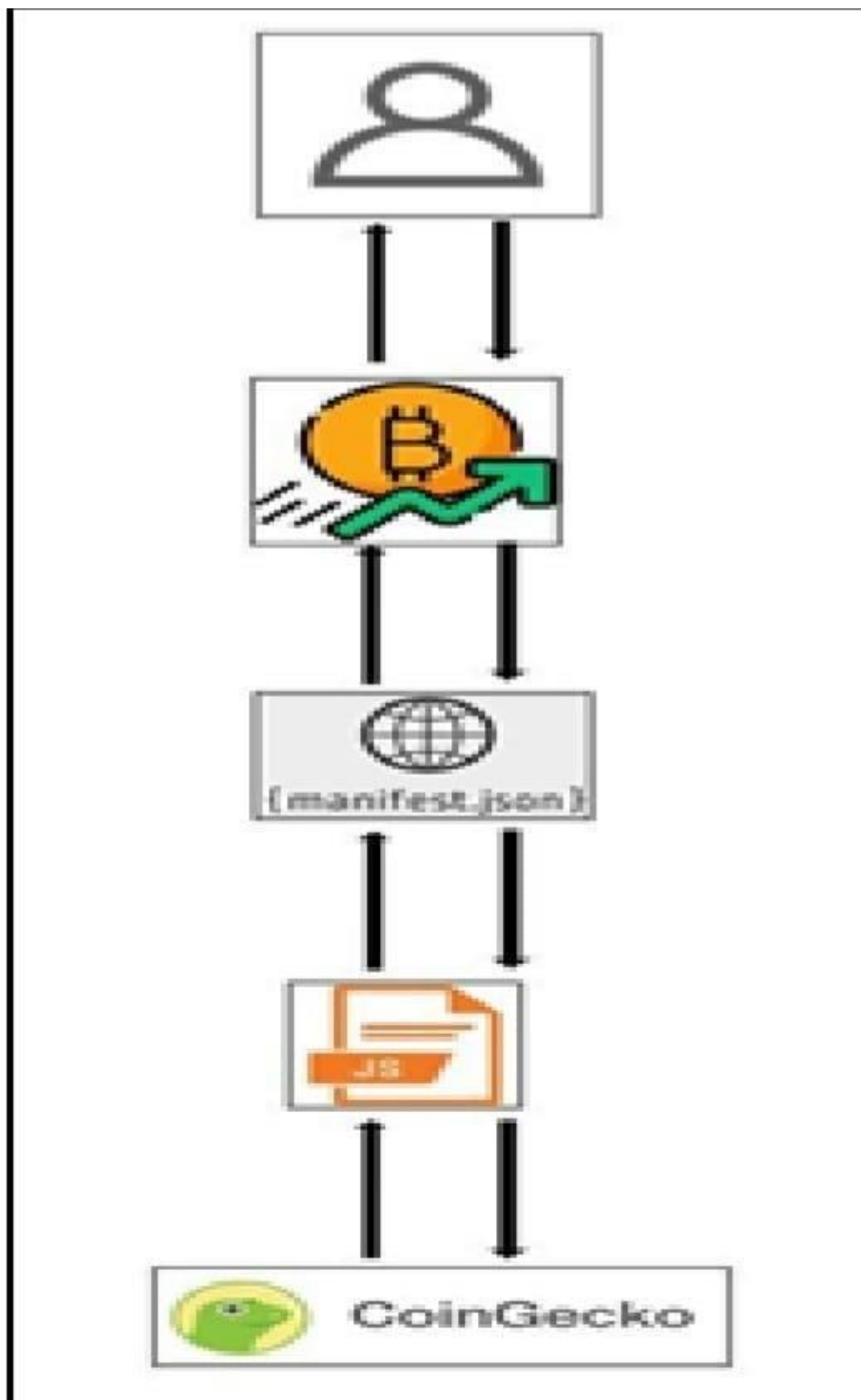
By tackling these challenges head-on and implementing comprehensive improvements, the crypto web app can position itself as a reliable, user-friendly, and secure platform. This will enable it to not only overcome its current limitations but also attract a wider user base, foster increased user engagement, and ultimately contribute to the broader adoption and acceptance of cryptocurrencies and blockchain technology."

3.2 Modules:

1. Choose a cryptocurrency API: You can use an API to access the latest price data for cryptocurrencies. Some popular APIs include CoinGecko, CryptoCompare, and Coinbase. Choose an API that provides the data you need and that you're comfortable working with.
2. Create a web application: You can use a web development framework like React, Angular, or Vue.js to create a user interface for your real-time price tracker. You can also use a backend framework like Node.js or Flask to create a server that communicates with the cryptocurrency API and provides the data to the frontend.
3. Retrieve cryptocurrency data: Use the API to retrieve the latest price data for the cryptocurrencies you want to track. You can use HTTP requests to access the API endpoints and retrieve the data in JSON format.

4 SOFTWARE DESIGN





5. SOFTWARE AND HARDWARE REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

The hardware requirements for a crypto app built with React.js, HTML, CSS, and JavaScript are generally modest, as these technologies are primarily executed on the user's device through a web browser. Here are the basic hardware requirements:

Processor: A modern processor with at least 1 GHz clock speed should be sufficient for running the app smoothly.

Memory (RAM): A minimum of 2GB RAM is recommended, although more RAM may be beneficial for handling larger data sets or complex operations.

Storage: The storage requirement depends on the size of the app and any accompanying data. Typically, a few hundred megabytes of storage should be sufficient, but this can vary based on the specific requirements of your crypto app.

Graphics: Since React.js, HTML, CSS, and JavaScript are primarily focused on user interface rendering and interactivity, a basic integrated graphics card is usually sufficient. Dedicated graphics cards are not typically required unless you have specific graphic-intensive features.

Operating System: The crypto app can run on various operating systems, including Windows, macOS, and Linux. Ensure that the user's device supports one of these operating systems.

Web Browser: The app will be accessed through a web browser, so make sure the app is compatible with popular browsers such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge. Keep in mind any browser-specific requirements or limitations that may impact your app's functionality.

It's important to note that these hardware requirements are general recommendations and can vary based on the complexity and specific features of your crypto app. If your app requires more advanced functionality, such as extensive data processing, encryption, or complex visualizations, you may need to consider higher-end hardware specifications.

Additionally, consider optimizing your app's performance by following best practices for code efficiency and considering performance optimizations like lazy loading, memoization, or data caching.

Overall, the hardware requirements for a crypto app made with React.js, HTML, CSS, and JavaScript are typically within the capabilities of most modern devices

5.2 Software Requirements:

To develop and run a crypto app made with React.js, HTML, CSS, and JavaScript, you will need the following software requirements:

Code Editor: A code editor is essential for writing and editing your app's source code. Some popular code editors include Visual Studio Code, Atom, Sublime Text, or WebStorm. Choose the one that suits your preferences and supports the features you need.

Node.js: Node.js is a JavaScript runtime that allows you to run JavaScript code outside of a web browser. It includes the Node Package Manager (NPM) for managing dependencies and packages. Install the latest stable version of Node.js from the official Node.js website (<https://nodejs.org>) to utilize its features and run your React app.

Package Manager (NPM or Yarn): NPM comes bundled with Node.js, but you can also use Yarn as an alternative package manager. These package managers allow you to install, manage, and update dependencies for your app. You will use them to install React, React Router, and other necessary packages.

React.js and Dependencies: React.js is the primary JavaScript library for building user interfaces. Use NPM or Yarn to install React and related dependencies like React Router, React Redux, or any other React-specific libraries you may need for your app.

CSS Preprocessors (Optional): If you prefer to use CSS preprocessors like Sass or Less, you'll need to install the respective compiler or preprocessor tools. These preprocessors enhance CSS functionality and allow you to use features like variables, mixins, and nesting.

Version Control (Git): Git is a distributed version control system that helps manage your source code and collaborate with others. Install Git on your machine and set up a repository to track your project's changes and history.

Web Browser: Your crypto app will be accessed through a web browser. Install the latest versions of popular browsers like Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge to test your app's compatibility and ensure proper functionality across different browsers.

These are the core software requirements for developing and running a crypto app built with React.js, HTML, CSS, and JavaScript. Additionally, you may require specific libraries, frameworks, or tools based on your app's requirements or additional features. It's recommended to stay updated with the latest versions of the software and libraries you use to leverage the latest features and security patches.

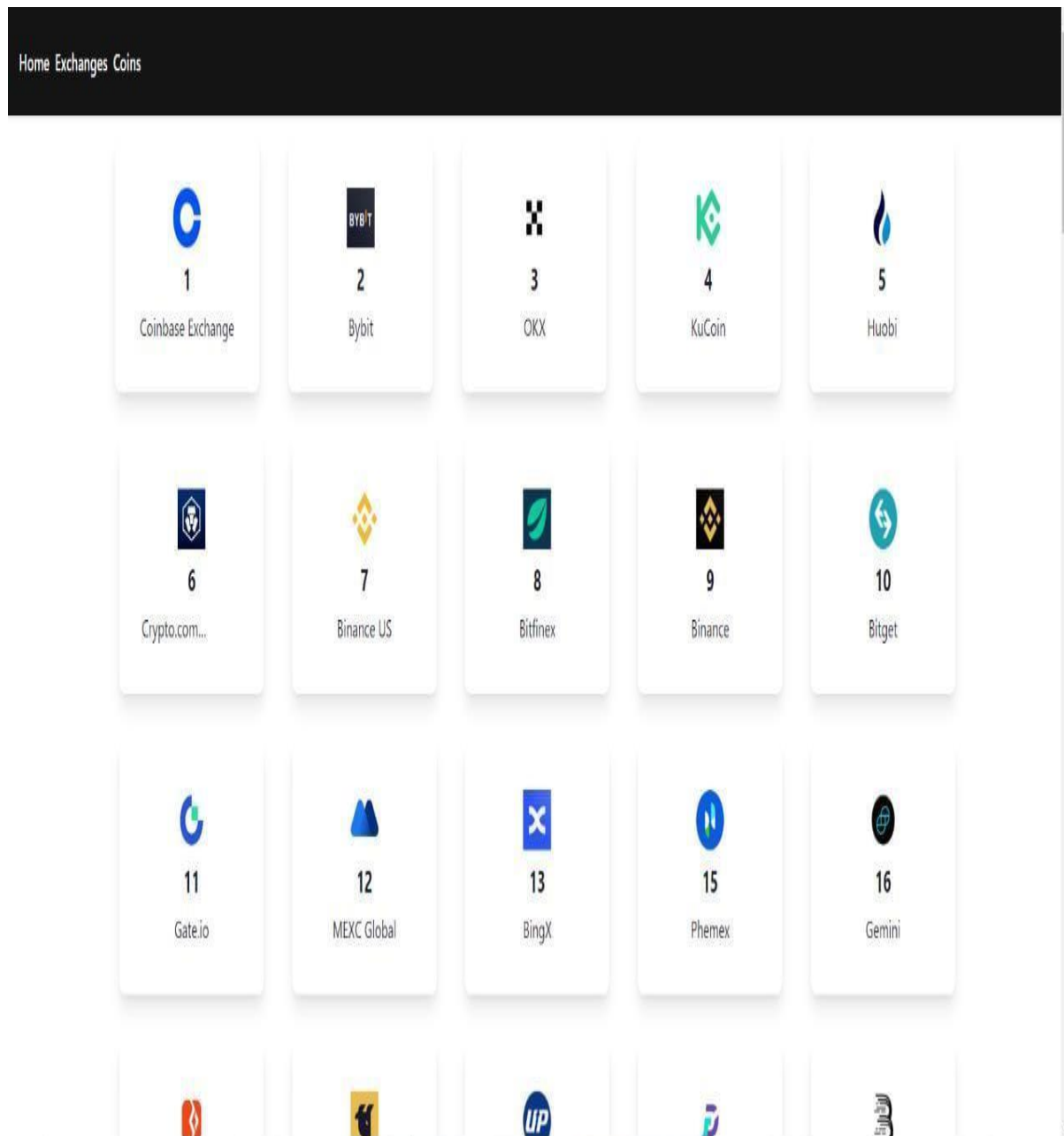
6 CODE TEMPLATES

6.1 Home Screen



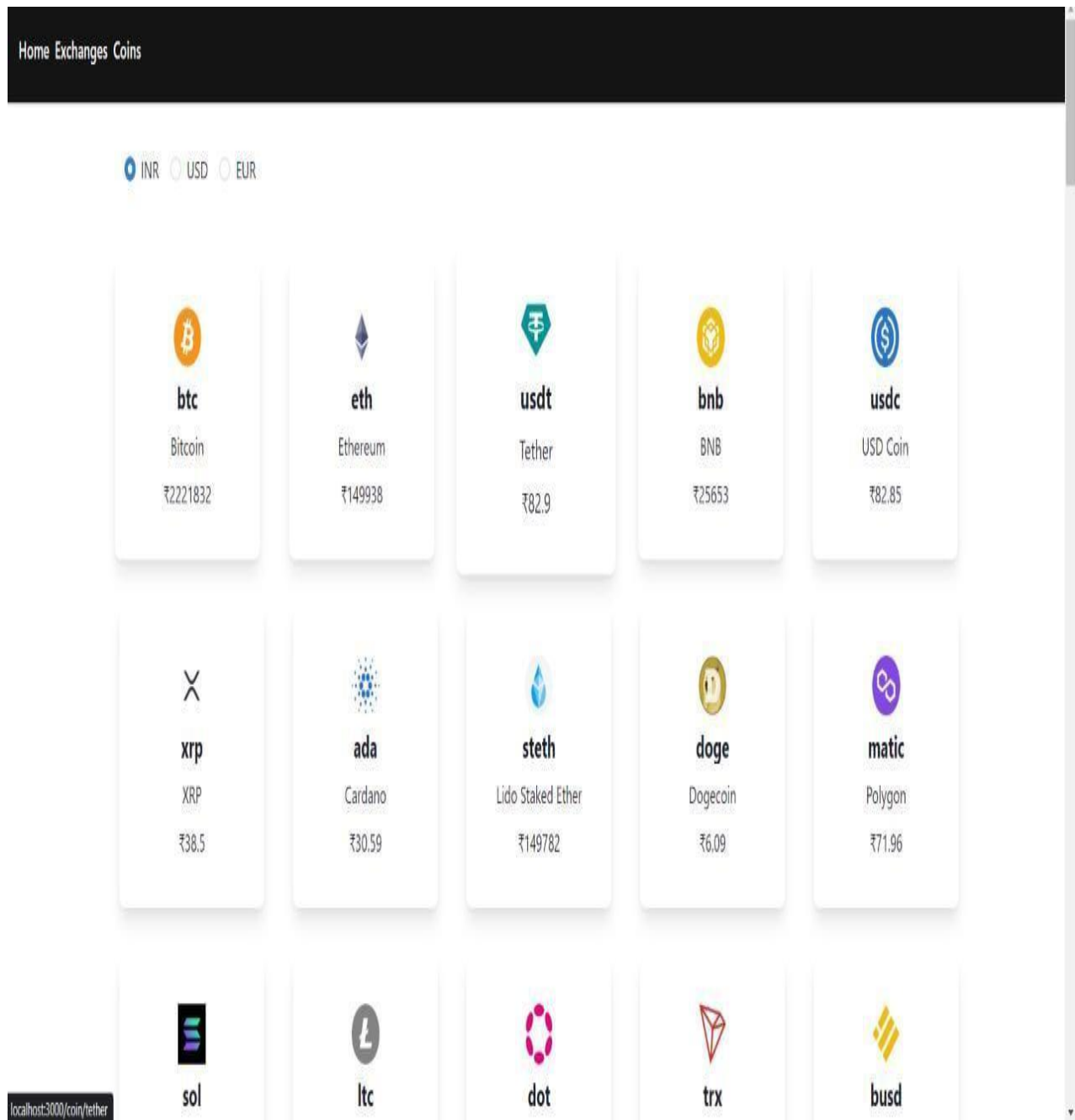
Remember that the specific content and layout of the homepage can vary based on the design and objectives of your crypto app. It's essential to create a visually appealing and user-friendly homepage that effectively communicates the value proposition of your app and entices users to explore further.

6.2 Exchanges Web Screen



The "Exchanges" page of a crypto app built with React.js, HTML, CSS, and JavaScript is a dedicated section where users can access information about different cryptocurrency exchanges. This page provides users with an overview of available exchanges, their features, trading pairs, and other relevant details.

6.3 Coins Page Screen

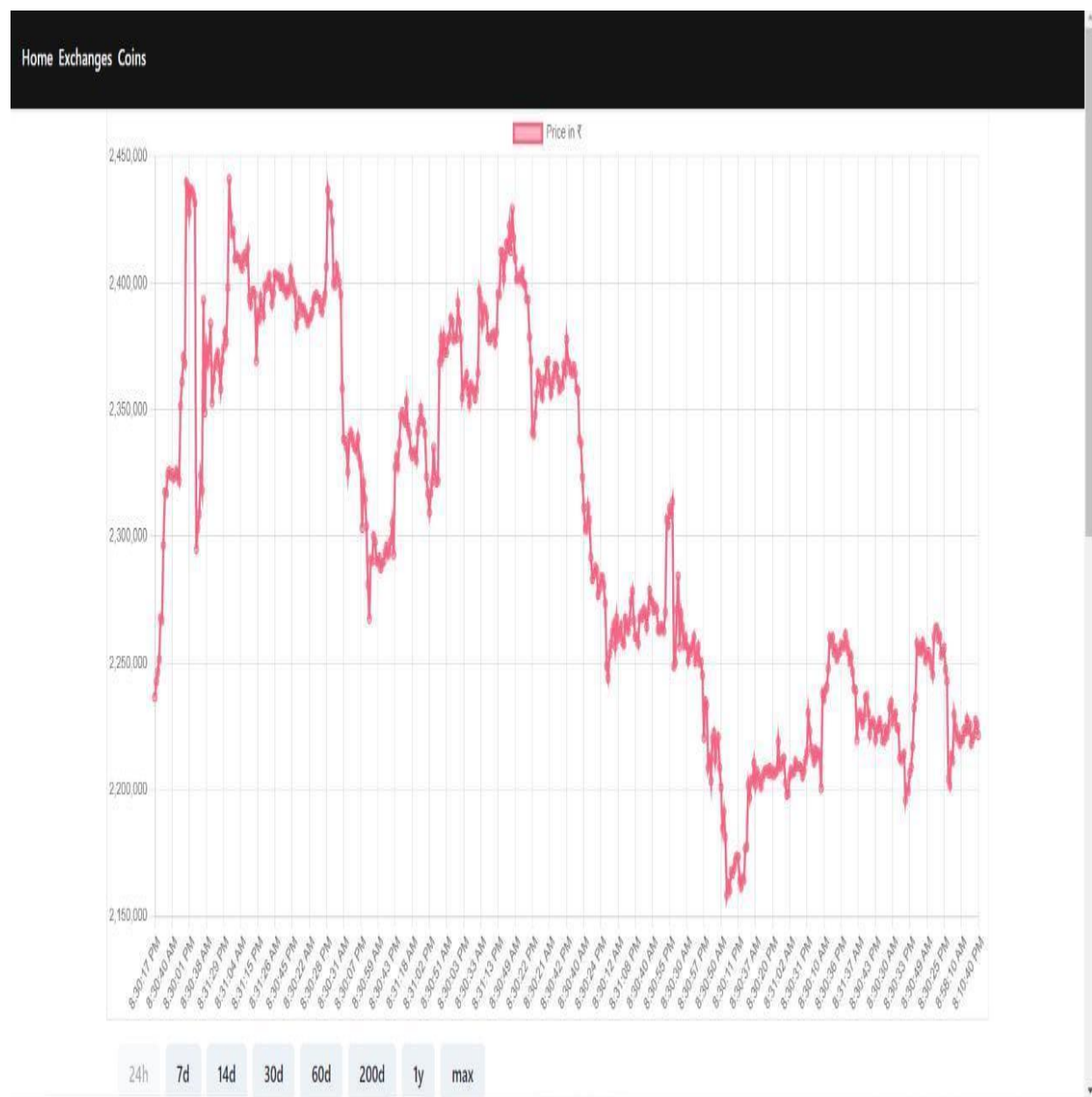


The "Coins" page of a crypto app built with React.js, HTML, CSS, and JavaScript is a dedicated section where users can explore and access information about various cryptocurrencies or digital assets. This page provides users with an overview of different coins, including their prices, market data, historical performance, and additional details.

Coin Listings: The page typically displays a list of cryptocurrencies or digital assets supported by the app. Each coin is usually represented by its logo or symbol, allowing users to easily identify and select the coins they are interested in.

Coin Details: Clicking on a specific coin from the list leads users to a detailed page providing comprehensive information about that coin. This can include the coin's name, symbol, market capitalization, circulating supply, total supply, and a brief description of the project or technology behind the coin.

6.4 Real Time Price Tracker



The "Real-Time Tracker" page of a crypto app built with React.js, HTML, CSS, and JavaScript is a dynamic section that provides users with real-time updates and tracking of cryptocurrency prices and market data. This page allows users to monitor the live movements of various cryptocurrencies and stay informed about their portfolio's performance. Here's a description of what you might find on the "Real-Time Tracker" page:

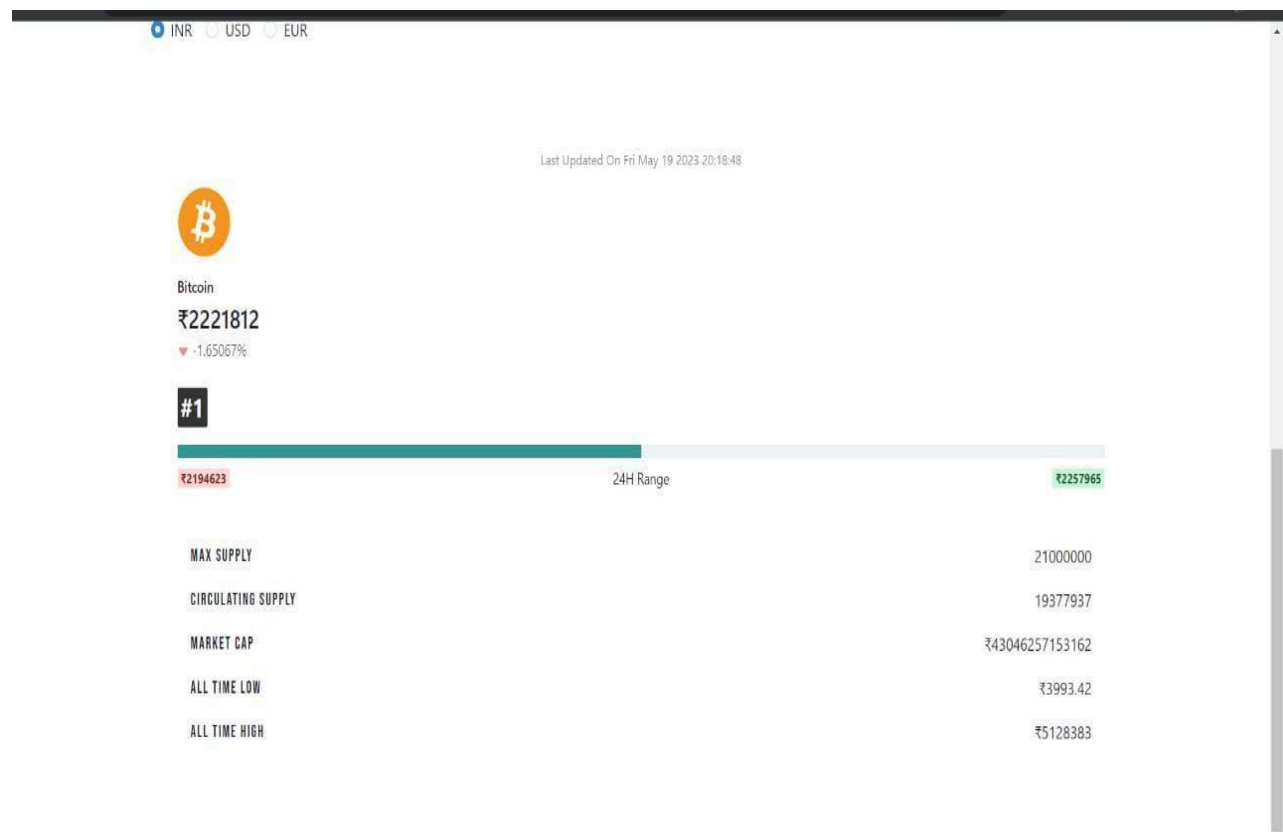
Cryptocurrency List: The page typically displays a list of selected cryptocurrencies or digital assets that users are tracking. Each cryptocurrency is usually represented by its logo or symbol, making it easy for users to identify their holdings.

User-friendly Interface: The "Real-Time Tracker" page should have a user-friendly and intuitive interface that provides a seamless experience for users. It should be easy to navigate, visually appealing, and provide clear and concise information.

Customization Options: Users may have the ability to customize the "Real-Time Tracker" page according to their preferences.

Price Charts: The page might include interactive price charts for each cryptocurrency, displaying the price movements over different time intervals (such as hourly, daily, weekly, or monthly). These charts help users visualize and analyze the historical price trends of the cryptocurrencies they are tracking.

6.5 Range Screen



In the context of a crypto app built with React.js, HTML, CSS, and JavaScript, a "Range" page refers to a section where users can set specific price ranges or filters to search for cryptocurrencies within a desired price range. This page allows users to narrow down their search and focus on cryptocurrencies that fall within their preferred price boundaries. Here's a description of what you might find on the "Range" page:

Detailed Cryptocurrency Information: Clicking on a specific cryptocurrency from the filtered list leads users to a detailed page providing comprehensive information about that coin. This can include the coin's name, symbol, market capitalization, price charts, trading volume, and other relevant data.

User-friendly Interface: The "Range" page should have a user-friendly and intuitive interface that makes it easy for users to set the desired price range, view the filtered results, and navigate to detailed cryptocurrency information. The layout should be visually appealing and provide a seamless user experience.

7 Testing :

7.1 Stress Testing:

Stress testing is an important aspect of testing the performance and scalability of a crypto app built with React.js, HTML, CSS, and JavaScript. It involves simulating high levels of user traffic and activity to identify how the app handles increased load and stress. Here's an overview of stress testing for your crypto app:

Identify Performance Goals: Determine the performance goals and metrics you want to measure during stress testing. This could include response time, throughput, concurrent user capacity, resource utilization, or any other relevant performance indicators.

Create Stress Test Scenarios: Define stress test scenarios that simulate heavy user loads and high levels of activity on your app. Consider factors such as simultaneous user logins, concurrent transactions, large data sets, or any other actions that put a significant load on your app.

Select a Stress Testing Tool: Choose a stress testing tool that supports simulating high user loads and generating concurrent requests. Popular tools include Apache JMeter, Gatling, or Locust. These tools allow you to define test scripts, configure virtual users, and execute stress tests.

Configure Test Environment: Set up a test environment that closely resembles the production environment. This includes using similar hardware, network conditions, and software configurations to ensure accurate results. You may need to scale up your infrastructure to handle the anticipated load.

Define Test Cases: Create test cases that simulate various stress test scenarios. This can include scenarios like multiple users making simultaneous trades, concurrent data retrieval from the API, or high transaction volumes. Define the expected behavior and outcomes for each test case.

Execute Stress Tests: Run the stress tests using the chosen stress testing tool. Gradually increase the load by adding virtual users or concurrent requests to reach the desired stress levels. Monitor the app's performance and collect relevant performance metrics during the test execution.

Analyze Results: Analyze the results and compare them against your predefined performance goals. Look for performance bottlenecks, response time delays, resource constraints, or any other issues that arise under stress conditions. Identify areas that need improvement or optimization.

Performance Tuning: Based on the results and identified bottlenecks, optimize the app's performance by addressing any performance issues. This can involve optimizing database queries, improving server-side code efficiency, caching data, or scaling up the infrastructure as needed.

Repeat and Iteration: Iterate the stress testing process by making necessary improvements based on the analysis and performance tuning. Re-run the stress tests to validate the effectiveness of the optimizations and ensure the app's performance meets the desired goals.

7.2 Unit Testing:

Unit testing is a crucial part of the development process for a crypto app built with React.js, HTML, CSS, and JavaScript. It involves testing individual components or functions in isolation to ensure their correctness and proper functionality. Here's an overview of unit testing for your crypto app:

Choosing a Testing Framework: There are several testing frameworks available for unit testing in React.js, such as Jest, React Testing Library, and Enzyme. These frameworks provide tools and utilities to write and execute tests effectively.

Test Environment Setup: Set up the necessary test environment by configuring the testing framework and installing any required dependencies. This includes configuring Babel or TypeScript if your app uses them.

Test File Structure: Organize your test files in a separate directory, preferably following a similar structure to your source code. For each component or function being tested, create a corresponding test file with the ".test.js" or ".spec.js" extension.

Writing Tests: Write individual test cases that cover different scenarios and expected outcomes for your components or functions. Focus on testing specific behaviors, input-output relationships, and edge cases. Tests should be independent and not rely on external dependencies.

Mocking Dependencies: When testing components or functions that depend on external resources or APIs, it's important to mock those dependencies. Use mocking libraries or techniques to create fake or stubbed versions of external services to ensure tests remain isolated.

Running Tests: Execute the tests using the testing framework's command-line interface or test runner. The framework will run through all the test files, execute the test cases, and report the results.

Assertions and Expectations: Use assertion libraries or the built-in assertion functions provided by the testing framework to verify expected outcomes. Compare actual results against expected values or conditions to determine whether the tests pass or fail.

Test Coverage: Aim for high test coverage to ensure that most, if not all, code paths are tested. Monitor and analyze test coverage reports to identify areas of your codebase that require additional test coverage.

Continuous Integration: Integrate your unit tests into your continuous integration (CI) process. Set up automated test runs that execute your unit tests whenever changes are pushed to your code repository. This helps catch potential issues early in the development cycle.

Test Maintenance: Regularly update and maintain your tests as your app evolves. As you make changes to components or functions, update the corresponding test cases to ensure they remain accurate.

7.2.1 Unit testing method

Unit testing in a crypto app made with React.js, HTML, CSS, and JavaScript involves testing individual components, functions, and modules in isolation to ensure their correctness and proper functionality. Here's a method for conducting unit testing in your crypto app:

Choose a Testing Framework: Select a testing framework that supports unit testing for React.js applications. Popular choices include Jest, React Testing Library, or Enzyme. Install the necessary dependencies for the testing framework.

Set Up the Testing Environment: Configure the testing environment by creating a separate directory for your unit tests. Set up the necessary configuration files and install any additional testing libraries or tools required by your chosen testing framework.

Identify Units for Testing: Determine the components, functions, or modules that need to be tested. Typically, each React component and any pure JavaScript functions or utility modules should have their corresponding unit tests.

Write Test Cases: Create test cases for each unit you identified. A test case should focus on a specific behavior or functionality of the unit. Write tests that cover different scenarios, input variations, and edge cases. Each test case should be independent and not rely on external dependencies.

Arrange, Act, and Assert: Follow the Arrange-Act-Assert (AAA) pattern for structuring your test cases. In the Arrange phase, set up the necessary preconditions and provide inputs. In the Act phase, invoke the unit being tested with the provided inputs. In the Assert phase, verify the expected outcomes by checking the results against expected values or condition.

7.2.2 Unit testing tasks

When performing unit testing for a crypto app made with React.js, HTML, CSS, and JavaScript, you can focus on the following tasks to ensure the correctness and proper functionality of individual components, functions, and modules:

Test Component Rendering: Verify that React components render correctly based on different props and state configurations. Test the presence of expected elements, CSS classes, and content within the rendered components.

Test Component Interactions: Simulate user interactions such as clicking buttons, entering text in input fields, or submitting forms. Ensure that the components respond appropriately to these interactions, such as updating state or triggering relevant callbacks.

Test Component Lifecycle Methods: If your components use lifecycle methods like `componentDidMount` or `componentDidUpdate`, test their behavior. Ensure that the lifecycle methods are called at the appropriate times and handle any necessary operations correctly.

Test Component State Changes: Validate that the state changes as expected when specific actions or events occur. Check that the component updates its state and triggers appropriate rendering or side effects accordingly.

7.3 What is Dependency Testing?

Dependency testing is an important aspect of the development process for a crypto web app created with React.js, HTML, CSS, and JavaScript. It involves examining and validating the dependencies between various components, libraries, and external resources used in the app. Here's how dependency testing is relevant to the crypto web app:

React Component Dependencies: In the crypto web app, React.js is used as the framework for building components. Dependency testing ensures that the components interact correctly with each other and function as intended. It involves verifying that props are passed correctly, state management is handled appropriately, and data flows smoothly between components.

Third-Party Library Compatibility: The crypto web app may rely on external libraries or packages to implement certain features or functionalities. During dependency testing, it is important to validate the compatibility and integration of these libraries. This includes checking for version compatibility, ensuring that the library functions as expected, and handling any conflicts or issues that may arise.

API and Backend Service Integration: The app may interact with APIs or backend services to retrieve data, perform authentication, or manage user accounts. Dependency testing involves verifying the integration of these services, ensuring that the app communicates correctly with the backend, and handling potential errors or issues related to the API endpoints or data formats.

Styling and CSS Dependencies: The app's visual appearance and styling rely on CSS and related dependencies. Dependency testing ensures that the CSS styles are applied correctly, layout components are properly styled, and any CSS frameworks or preprocessors used in the project function as intended.

Browser Compatibility: The crypto web app should work consistently across different web browsers. Dependency testing involves validating that the app functions correctly and displays as expected in popular browsers such as Chrome, Firefox, Safari, and Edge. This may involve addressing any browser-specific issues, ensuring proper rendering, and handling cross-browser inconsistencies.

By conducting comprehensive dependency testing, developers can identify and resolve any issues or conflicts related to dependencies, improving the overall functionality, stability, and compatibility of the crypto web app. It helps ensure that the app operates as intended and provides a seamless and reliable user experience for exploring, monitoring, and managing cryptocurrencies.

8. Research Methodology

Crypto-currencies have become very popular these days and have gained popularity all over the world. People are increasingly investing into these kind of investments and are also using them for their day to day transactions.

The paper focuses on secondary data collected from various sources which depict the status of the currency and its future aspects.

Secondary Data- Secondary data was gathered from the source, articles from scholarly diaries and books to have thorough see of the subject and encourage speak to and approve the study questions

9. CONCLUSIONS

In conclusion, the crypto web app developed using React.js, HTML, CSS, and JavaScript provides a robust and user-friendly platform for exploring, monitoring, and managing cryptocurrencies. The app incorporates key features such as a homepage with an overview of popular cryptocurrencies and market trends, cryptocurrency listings with detailed information, portfolio management functionality, interactive price charts, news updates, and secure user authentication.

The use of React.js as the framework enables efficient component-based development, allowing for modular and reusable code. HTML and CSS are utilized to structure and style the app, ensuring an appealing and intuitive user interface. JavaScript provides the necessary logic for handling user interactions, data retrieval, and authentication processes.

The app's responsive design ensures optimal user experience across various devices and screen sizes, allowing users to access and utilize its features seamlessly from desktops, laptops, tablets, and mobile phones. The incorporation of a user authentication system ensures the security of user data, allowing users to create accounts, log in, and access personalized features like portfolio management.

Overall, the crypto web app presents a comprehensive and visually appealing solution for cryptocurrency enthusiasts and investors. It combines data visualization, real-time market information, news updates, and user account functionality to provide users with a powerful tool for exploring the world of cryptocurrencies and managing their investments effectively.

In conclusion, developing a crypto app using React.js, HTML, CSS, and JavaScript offers several advantages. Here are some key takeaways:

React.js for Component-Based Development: React.js allows for the creation of reusable and modular components, promoting a component-based development approach. This enables easier code maintenance, reusability, and scalability.

Interactive and Responsive UI: HTML, CSS, and JavaScript provide the foundation for creating an interactive and responsive user interface. With the use of CSS frameworks and JavaScript libraries, you can enhance the visual appeal and user experience of your crypto app.

Real-Time Data Tracking: By leveraging JavaScript libraries and APIs, you can implement real-time data tracking in your crypto app. This enables users to monitor live market data, prices, and portfolio values, keeping them up-to-date with the latest information.

Integration with APIs and Backend Services: JavaScript and React.js facilitate seamless integration with various APIs and backend services, allowing you to fetch crypto-related data, execute transactions, and interact with blockchain networks.

Testing and Quality Assurance: Unit testing and stress testing play crucial roles in ensuring the correctness, performance, and scalability of your crypto app. Employing appropriate testing frameworks and methodologies helps identify and rectify potential issues early in the development process.

Security Considerations: As a crypto app deals with sensitive financial data, security should be a top priority. Implementing secure coding practices, utilizing encryption techniques, and following industry standards for handling user data are essential to protect user privacy and prevent security breaches.

Continuous Improvement: The crypto market is dynamic, and user requirements may evolve over time. It is crucial to continuously monitor and update your app to meet changing needs, add new features, and address any performance or security concerns.

By leveraging the capabilities of React.js, HTML, CSS, and JavaScript, you can develop a robust and user-friendly crypto app that provides real-time data tracking, secure transactions, and an engaging user interface. However, it's important to stay updated with industry trends, security best practices, and user feedback to ensure the long-term success of your crypto app.

10. FURTHER ENHANCEMENTS

To further enhance your crypto app made with React.js, HTML, CSS, and JavaScript, consider the following areas for improvement:

User Experience (UX) Enhancements:

Streamline User Onboarding: Simplify the registration and onboarding process to make it easier for new users to sign up and get started with your app.

Intuitive Navigation: Ensure clear and intuitive navigation throughout the app, allowing users to easily access different sections and features.

Responsive Design: Optimize your app's design to be fully responsive across various devices and screen sizes, providing a consistent experience to users.

Additional Features:

Advanced Charting: Implement interactive and customizable charts to allow users to analyze cryptocurrency price trends, historical data, and technical indicators.

Portfolio Management: Enable users to create and manage their crypto portfolios, track holdings, view performance metrics, and set alerts for price changes.

News and Market Insights: Integrate a news feed or provide curated market insights and analysis to keep users informed about the latest crypto news and trends.

Integration with Wallets and Exchanges:

Wallet Integration: Allow users to connect their cryptocurrency wallets to your app, enabling seamless management of their funds and transactions.

Exchange Integration: Integrate popular cryptocurrency exchanges to enable users to execute trades directly from your app, providing a comprehensive trading experience.

Social Features:

Social Sharing: Enable users to share their portfolio performance or specific cryptocurrency information on social media platforms, fostering user engagement and virality.

Community Engagement: Implement features like user profiles, forums, or chat functionality to encourage community interaction, discussions, and knowledge sharing.

Performance Optimization:

Code Optimization: Review and optimize your codebase for better performance, including minimizing file sizes, reducing unnecessary rendering, and optimizing API calls.

Caching and Data Management: Implement caching mechanisms to improve app performance and reduce the load on APIs by storing frequently accessed data locally.

Lazy Loading: Utilize lazy loading techniques to load components and data only when necessary, improving initial page load times.

Localization and Internationalization:

Localization: Add support for multiple languages to cater to users from different regions, allowing them to use your app in their preferred language.

International Currency Support: Enable users to view cryptocurrency prices and portfolio values in their local currencies for a personalized experience.

Accessibility:

Accessibility Compliance: Ensure your app meets accessibility standards, making it usable for individuals with disabilities. Implement proper ARIA attributes, keyboard navigation, and screen reader compatibility.

Continuous Improvements and Bug Fixes:

Regular Updates: Stay updated with the latest industry trends, regulatory changes, and user feedback to continuously improve and enhance your app's features and functionality.

Bug Fixes: Address any reported bugs promptly, perform regular testing, and release bug fixes to maintain a stable and reliable app.

By incorporating these enhancements, you can provide a more comprehensive and engaging experience for users of your crypto app. Continuous improvement, user feedback, and staying up-to-date with market trends are essential to meet evolving user needs and ensure the long-term success of your app.

In the not too distant future, we want to incorporate various AI and ML algorithms in order to provide more precise statistics on the currencies.

In addition to that, we may carry out tests with many other technologies and evaluate the outcomes of those tests in light of the outcomes of the currently available technologies. The other situation that may have an effect on the future of our program is the development of cryptography in the future because, ultimately, our project is dependent on it.

Some of the limitations of cryptocurrencies, such as a computer malfunction erasing digital riches or a hacker plundering a virtual vault, may be alleviated in the future as a result of technological advancements. The widespread use of cryptocurrencies is expected to result in increased regulation and scrutiny from governments, which will eventually undermine the fundamental concept behind these currencies. Although an increasing number of merchants now accept cryptocurrencies, those who do so remain a minority. Obtaining customer acceptability is essential for the further growth of the cryptocurrency market. Their complexity in comparison to that of traditional currencies is likely to discourage most consumers, with the exception of techies. If a cryptocurrency wishes to enter the mainstream market, it could be required to fulfill a number of conditions. It must be mathematically complicated (to minimize fraud and hacker assaults), yet easy for consumers to comprehend; decentralized, but with sufficient consumer protections and protection; and it must retain user anonymity without facilitating tax evasion, money laundering, or any other illegal actions. Given the difficulty of fulfilling these requirements, might it be that in a few years the most popular cryptocurrency will be one that sits between highly controlled fiat currencies and the cryptocurrencies that are used today? Even if it is very unlikely, Bitcoin's success (or failure) in overcoming its problems might have repercussions for the future of other cryptocurrencies.

Unlike this, The advantages offered by cryptocurrencies have in recent times caught the attention of institutions, and the conventional banking industry is scrambling to keep up with the demand. For instance, U.S. Bank has just introduced a bitcoin custody service, which makes it possible for hedge funds to invest in digital currency.

11. REFERENCE

- [1] Ujan Mukhopadhyay. A Brief survey of cryptocurrency Systems. IEEE2021
- [2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2020, [online] Available
- [3] E. Androulaki et al., "Evaluating User Privacy in Bitcoin", Financial Cryptography and Data Security,2013, [online] Available. Arthur Gervais. Is Bitcoin a Decentralized Currency? IEEE.2020
- [4] Ryan Farrell. An analysis of the cryptocurrency industry. available at repository.upenn.edu, 2019.
- anonymous. Mining Mining, 2019.
- [5] Jason Teutsch, Sanjay Jain, and Prateek Saxena. When cryptocurrencies their own business.
- [6] D. Tapscott A. Tapscott "Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money" in Business and the World Penguin UK 2018.
- [7] A. Hertig "SegWit Goes Live: Why Bitcoin's Big Upgrade Is a Blockchain Game-Changer" Coindesk 2018.
- [8] S. Porru A. Pinna M. Marchesi R. Tonelli "Blockchain oriented software engineering: challengesand new directions" Proceedings of the 39th International Conference on Software Engineering Companion IEEE Press pp. 169-171 2017.
- [9] X. Xu C. Pautasso L. Zhu V. Gramoli A. Ponomarev A. B. Tran S. Chen "The blockchain as a software connector" Software Architecture (WICSA) 2017 13th Working IEEE/IFIP Conference on pp. 182-191