CRYPTO EXCHANGE APP

A dissertation submitted to the Jawaharlal Nehru Technological University, Hyderabad in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

M. Ritvik Reddy(20B81A05M2)

P. Sowmith Reddy (20B81A05P0)

V. Sujith Reddy(20PQ1A0567)

Under the Guidance of
Mr. Syed Muqthadar Ali
Sr.Assistant Professor , CSE Department

Department of Computer Science and Engineering

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH, Accredited by NBA, and NAAC)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Ranga Reddy (Dist.) - 501510, Telangana State

2023-24

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH, Accredited by NBA, and NAAC)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Ranga Reddy (Dist.) - 501510, Telangana State.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"Crypto Exchange App"** is being submitted by **M. Ritvik Reddy(20B81A05M2), P. Sowmith Reddy (20B81A05P0), and V. Sujith Reddy(20PQ1A0567)**in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering,** during the academic year 2023-2024.

Project Guide
Mr. Syed Muqthadar Ali
Sr.Assistant Professor , CSE Department

Professor-in-charge projects
Dr. S. Suguna Mallika

External Examiner

Professor and Head, CSE
(Dr. A. Vani Vasthala)

DECLARATION

We hereby declare that this project report titled "**Crypto Exchange App**" submitted to the Department of Computer Science and Engineering, CVR College of Engineering, is a record of original work done by us under the guidance of **Mr. Syed Muqthadar Ali, Sr. Assistant Professor**. The information and data given in the report is authentic to the best of our knowledge. This project report is not submitted to any other university or institution for the award of any degree or diploma or published at any time before.

M. RITVIK REDDY (20B81A05M2)

P. SOWMITH REDDY (20B81A05P0)

V. SUJITH REDDY (20PQ1A0567)

Date: 1st April, 2024

Place: **Hyderabad**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is a great pleasure to convey out profound sense of gratitude to our principal **Dr. K. Ramamohan Reddy**, Vice Principal Prof. **L. C. Siva Reddy, Dr. A. Vani Vathsala**, Head of CSE Department, CVR College of Engineering, for having been kind enough to arrange necessary facilities for executing the project in the college.

We deem it a pleasure to acknowledge our gratitude to our project guide **Mr. Syed Muqthadar Ali,** Sr. Assistant Professor, CSE Department under whom we have carried out the project work. Her incisive and objective guidance timely encouraged us with constant flow of energy to continue the work. Our project coordinator, **Mr. K. Giri Babu**, Senior Assistant Professor, CSE Department, provided us with all the necessary information, timelines, and inputs when and where required for the project's successful completion.

We convey our heartfelt thanks to management for providing excellent lab facilities and tools. Finally, we thank all those guidance helpful to us in this regard.

ABSTRACT

A cryptocurrency exchange app that uses swapping of two different valued tokens named EVT and DVT, and how it is implemented using Flutter frontend, Infura for smart contract integration, and Solidity for writing smart contracts. The cryptocurrency exchange app is designed to provide users with the ability to trade or swap tokens on the Ethereum blockchain. In this app, two tokens are supported: EVT and DVT. EVT and DVT are ERC-20 tokens, which means they are built on the Ethereum blockchain and follow the ERC-20 standard for token implementation.

The frontend of the app is developed using Flutter, a popular cross-platform framework for building mobile applications. Flutter allows for the creation of a seamless user interface that enables users to interact with the app's functionalities easily. The app's frontend communicates with the Ethereum blockchain using Infura.

Infura is a service that provides access to the Ethereum network without the need for users to run their own Ethereum node. It acts as a bridge between the app's frontend and the Ethereum blockchain. By connecting to Infura, the app can interact with the smart contracts deployed on the Ethereum network.

The smart contracts that power the swapping functionality of the app are written in Solidity. Solidity is a programming language specifically designed for writing smart contracts on the Ethereum platform. The smart contracts define the rules and logic for token swapping, including price calculation, token balances, and transfer functions.

TABLE OF CONTENTS

LIST OF FIGURES

1.INTRODUCTION

In the fast-evolving landscape of cryptocurrency, the need for efficient and user-friendly platforms for token exchange has never been greater. With this in mind, we present a groundbreaking project: a cryptocurrency exchange app tailored to the swapping of two distinct tokens, EVT and DVT, built on the Ethereum blockchain.

EVT and DVT represent the next generation of digital assets, adhering to the widely recognized ERC-20 standard on the Ethereum network. Our app is designed to empower users with the seamless ability to trade or swap these tokens, offering a streamlined and secure avenue for managing their crypto portfolios.

At the forefront of our app's architecture lies Flutter, a versatile cross-platform framework renowned for its ability to craft intuitive user interfaces. Through Flutter, we ensure that users can effortlessly navigate and interact with the app's functionalities, enhancing their overall trading experience.

Facilitating the integration between our app's frontend and the Ethereum blockchain is Infura, a service that provides seamless access to the Ethereum network without the complexities of running individual nodes. By leveraging Infura as a bridge, our app establishes a direct link to the Ethereum blockchain, enabling real-time interaction with deployed smart contracts.

Central to the functionality of our exchange app are the smart contracts written in Solidity, Ethereum's native programming language for smart contract development. These contracts serve as the backbone of the swapping mechanism, defining rules and logic for price calculation, token balances, and transfer operations, ensuring transparency and security throughout the trading process.

Through the convergence of innovative technologies and robust infrastructure, our cryptocurrency exchange app sets a new standard for accessibility and efficiency in digital asset trading. Join us on this journey as we redefine the future of decentralized finance, one swap at a time.

1.1 MOTIVATION

Smart contracts written in Solidity are revolutionizing the landscape of digital transactions by ensuring a high level of security and transparency. The choice of Solidity as the programming language for crafting these smart contracts is pivotal, as it offers robustness and reliability. These contracts, executed on the Ethereum blockchain, establish a trustless environment where transactions are conducted without the need for intermediaries. This eliminates the reliance on centralized authorities, thereby enhancing security and reducing the potential for fraud or manipulation.

One of the key applications of smart contracts is in decentralized exchanges (DEX), where users can directly swap tokens on the blockchain. By enabling users to exchange EVT and DVT tokens directly on the Ethereum blockchain, the app embraces the ethos of decentralization. DEX platforms offer numerous benefits, including enhanced security, increased privacy, and reduced transaction costs. Furthermore, they empower users by giving them full control over their assets without the intervention of third-party intermediaries. This not only fosters financial sovereignty but also promotes a more equitable and inclusive financial ecosystem.

The development of a cryptocurrency exchange app using innovative technologies such as Flutter for the frontend, Infura for blockchain interaction, and Solidity for smart contract deployment presents an invaluable educational opportunity. It provides developers and enthusiasts with hands-on experience in building decentralized applications (DApps) and navigating the intricacies of blockchain technology. By delving into the nuances of blockchain development, individuals gain a deeper understanding of decentralized finance (DeFi) principles and contribute to the wider adoption of blockchain technology.

Moreover, educational initiatives focused on blockchain development foster a community of proficient developers and enthusiasts who are well-versed in the intricacies of decentralized systems. This not only enriches the talent pool within the blockchain ecosystem but also catalyzes innovation and adoption across various industries. As individuals acquire proficiency in blockchain development, they become instrumental in

driving the evolution of decentralized finance and spearheading the development of groundbreaking applications.

In essence, the convergence of Solidity-based smart contracts, decentralized exchanges, and educational endeavors in blockchain development creates a symbiotic relationship that propels the industry forward. By leveraging cutting-edge technologies and embracing decentralized principles, developers contribute to the growth of a resilient and inclusive financial ecosystem. This not only enhances the security and efficiency of digital transactions but also empowers individuals to take control of their financial destinies in a trustless and transparent manner.

1.2 PROBLEM STATEMENT

The current landscape of cryptocurrency exchanges presents significant challenges in terms of accessibility, user experience, and decentralized functionality. While decentralized exchanges (DEX) hold the promise of increased financial inclusion and user control, existing implementations often fall short in providing a seamless and user-friendly interface.

One major obstacle is the lack of intuitive interfaces on decentralized exchanges. Users, especially newcomers to the cryptocurrency space, encounter difficulties navigating and executing trades on these platforms. The complexity of decentralized systems and the technical knowledge required to interact with them effectively hinder the adoption of DEX among mainstream users.

Moreover, the absence of efficient cross-platform solutions tailored for mobile devices exacerbates the issue. Many existing DEX platforms offer limited or cumbersome mobile experiences, further deterring users who rely on smartphones and tablets for their trading activities.

Additionally, interacting with blockchain networks directly adds another layer of complexity to decentralized trading. Users often face challenges such as high transaction fees, slow confirmation times, and the risk of errors when interacting with smart contracts and decentralized protocols.

As a result, the widespread adoption of decentralized trading remains hindered by these usability and accessibility issues. Addressing these challenges is crucial for realizing the full potential of decentralized exchanges and advancing the democratization of finance through cryptocurrency.

1.3 PROJECT OBJECTIVES

1.To Define the Scope:

Clearly outline the boundaries and parameters of the project to ensure a focused approach.

2.To Identify Stakeholders:

Identify and engage with key stakeholders who will be impacted by or have an interest in the project outcomes.

3.To Establish Clear Goals:

Define specific, measurable, achievable, relevant, and time-bound (SMART) goals that the project aims to achieve.

4.To Develop a Project Plan:

Create a detailed project plan outlining tasks, milestones, timelines, and resources required for successful project execution.

5.To Allocate Resources:

Identify and allocate the necessary human, financial, and material resources to support the project objectives.

6.To Conduct Research:

Gather relevant data, literature, and information to inform decision-making and project implementation.

7.To Execute the Project:

Implement the project plan according to the defined scope, goals, and timeline, while monitoring progress and addressing any issues that arise.

8.To Manage Risks:

Identify potential risks and develop mitigation strategies to minimize their impact on project outcomes.

9.To Ensure Quality:

Implement quality assurance processes to ensure that project deliverables meet the desired standards and requirements.

10.To Evaluate and Monitor Progress:

Regularly assess project progress against established milestones and objectives, making adjustments as necessary to ensure successful completion.

11.To Communicate Effectively:

Maintain open and transparent communication with stakeholders throughout the project lifecycle, providing updates on progress, challenges, and achievements.

12.To Achieve Project Deliverables:

Successfully deliver project outputs, outcomes, or products that meet the defined scope, goals, and quality standards.

13.To Document and Share Learnings:

Document project processes, outcomes, and lessons learned to facilitate knowledge sharing and continuous improvement for future projects.

1.4 PROJECT REPORT ORGANIZATION

Chapter 1: Introduction

The introduction of the "Crypto Exchange App" project encapsulates the motivation behind the endeavor, highlighting the growing significance of decentralized finance (DeFi) and the need for user-friendly platforms for token swapping. It delves into the challenges faced by centralized exchanges and the demand for secure and transparent alternatives. Furthermore, it articulates the problem statement, emphasizing the importance of developing a decentralized exchange app using innovative technologies like Flutter, Infura, and Solidity.

Chapter 2: Literature Review

The literature review provides a comprehensive examination of existing research in the fields of cryptocurrency exchanges, blockchain technology, and decentralized finance. It synthesizes prior studies and projects, offering insights into the current state of knowledge in the domain. Additionally, it critically assesses the limitations of centralized exchanges and highlights the benefits of decentralized exchanges in enhancing security, reducing transaction costs, and fostering financial sovereignty.

Chapter 3: Requirement Analysis

This chapter focuses on the analysis of software, hardware, and user requirements essential for the development and implementation of the crypto exchange app. It outlines the tools, technologies, and resources necessary to support the project's objectives, ensuring a clear understanding of the project's logistical requirements and constraints.

Chapter 4: System Design

The system design chapter elaborates on the proposed architecture, methods, and algorithms employed in the development of the crypto exchange app. It provides readers with an in-depth understanding of the app's structure, functionality, and technical components through the use of diagrams and descriptions of datasets and technology stacks utilized.

Chapter 5: Implementation

In this chapter, the implementation of the crypto exchange app is detailed. It includes screenshots showcasing the user interface design, descriptions of individual modules, and discussions on outcomes, user feedback, and future development directions.

Chapter 6: Conclusion

The conclusion section synthesizes the findings and outcomes of the project, offering a succinct summary of the key insights gleaned from the research and implementation process. It also outlines potential avenues for future research and development, providing guidance on areas for further exploration and refinement in the realm of decentralized exchanges and blockchain technology.

Chapter 7: References

The references section provides a comprehensive list of all sources cited throughout the report, ensuring transparency and academic integrity by acknowledging the contributions of prior research and scholarship in the field of cryptocurrency exchanges and blockchain technology.

2. LITERATURE REVIEW

The literature review section serves as a critical component of any project report, offering a thorough examination of existing research, theories, and scholarly works pertinent to the project's topic. Beginning with an introduction, this section provides essential context by defining key terms and concepts, thereby ensuring clarity for the reader.

As the review progresses, it delves into the historical context of the topic, tracing its evolution and development over time. This historical perspective offers valuable insights into the trajectory of research and helps situate the project within a broader academic framework. Additionally, it sets the stage for understanding the current state of knowledge in the field.

A comprehensive literature review entails a thorough analysis of previous research studies, articles, and academic papers related to the project's focus. By summarizing key findings, methodologies, and conclusions from these studies, the review synthesizes existing knowledge and identifies gaps or limitations that the project aims to address. This critical analysis serves as a foundation for framing the project's research questions and objectives.

Moreover, the literature review highlights emerging trends, innovations, and developments in the field, shedding light on potential areas for further exploration. By examining different methodological approaches and research methods employed in previous studies, the review also provides insights into the strengths and limitations of various research methodologies, informing the project's own methodological approach.

Ultimately, the literature review serves as a roadmap for the project, guiding its direction, and informing its methodology and objectives. By critically analyzing and synthesizing existing research, the review lays the groundwork for the project's contributions to the field and underscores its significance within the broader academic discourse.

2.1 EXISTING WORK

Decentralized Exchange Models:

In the landscape of decentralized exchanges (DEX), a multitude of models have emerged, each presenting unique approaches to facilitate peer-to-peer token trading. These models vary in their implementation, with some platforms leveraging on-chain order books and smart contracts to match and execute trades, thereby ensuring a trustless environment (Buterin, 2013). Conversely, others explore off-chain order matching mechanisms, aiming to optimize speed and scalability (Lamport et al., 2018). The choice of DEX model plays a pivotal role in shaping various factors such as latency, liquidity, and user control. Understanding the nuances and implications of these different models is essential for designing and implementing efficient and effective decentralized exchange platforms.

User Interface Innovations:

Within the realm of decentralized exchange applications, there has been a notable emphasis on innovating user interfaces to attract a broader user base. Research indicates that improving the user experience is crucial for fostering the mass adoption of decentralized financial tools (Luther, 2020). As such, there has been a concerted effort to develop intuitive designs, streamline onboarding processes, and provide real-time transaction feedback to enhance user engagement and satisfaction. These user interface innovations not only contribute to a more seamless trading experience but also serve as catalysts for broader adoption and acceptance of decentralized exchange platforms among mainstream users.

Smart Contract Implementations:

Smart contracts, written in languages like Solidity, serve as the foundational building blocks of decentralized exchanges. Researchers have explored various smart contract architectures to govern token swapping, ensuring transparency and security in the exchange process. These smart contracts are integrated into decentralized exchange platforms to enhance trading functionalities, providing mechanisms for order matching, fund custody, and dispute resolution (Swan, 2015). By leveraging smart contract technology,

decentralized exchanges can offer users greater transparency, efficiency, and autonomy in their trading activities.

In summary, the existing work in the field of decentralized exchanges encompasses diverse areas of research and development, ranging from exploring different exchange models to innovating user interfaces and implementing smart contract solutions. By understanding and building upon these existing frameworks and advancements, researchers and developers can continue to push the boundaries of decentralized finance, driving towards greater accessibility, usability, and trust in the cryptocurrency ecosystem.

2.2 LIMITATIONS OF EXISTING WORK

Scalability Challenges:

Despite significant advancements, scalability remains a pervasive limitation across many decentralized exchanges (DEX). As user adoption continues to grow, certain DEX models encounter difficulties in efficiently handling increased transaction volumes. This scalability challenge underscores the inherent trade-off between decentralization and scalability, where maintaining decentralized principles often comes at the expense of transaction throughput and speed (Zohar, 2015).

Liquidity Concerns:

Maintaining adequate liquidity poses a persistent challenge for decentralized exchanges. Limited liquidity can lead to higher slippage rates and reduced trading efficiency for users, ultimately impacting the overall user experience. Research indicates that incentivizing liquidity providers through mechanisms such as yield farming presents a potential solution to address liquidity concerns (Hendrik, 2021). By offering incentives for liquidity provision, decentralized exchanges can attract more market participants, thereby enhancing liquidity pools and improving trading conditions for users.

Regulatory Uncertainty:

The regulatory landscape surrounding decentralized exchanges remains fraught with uncertainty in many jurisdictions. Compliance with local regulations presents significant challenges for DEX operators, as the lack of standardized regulatory frameworks leaves room for ambiguity and interpretation (Haque, 2019). Regulatory uncertainty not only introduces legal and compliance risks but also inhibits the growth and adoption of decentralized finance (DeFi) platforms. Clearer regulatory guidelines and frameworks are needed to provide DEX operators and users with greater clarity and confidence in navigating the regulatory landscape. By establishing transparent and consistent regulatory frameworks, policymakers can foster trust and legitimacy within the DeFi ecosystem, thereby facilitating broader adoption and innovation in decentralized finance.

3.REQUIREMENT ANALYSIS

3.1 SOFTWARE REQUIREMENTS

1. Operating System - Windows, OS

2. Programming Language's – Flutter, Infura and Solidity

3. Execution Environment – Android Studio, Visual Studio Code, RemixIDE

4. Metamask Wallet: For testing and deploying smart contracts during development.

3.2 HARDWARE REQUIREMENTS

1. RAM - 8 GB or above

2. Processor - Intel core i5 or above

3. Storage - 256 GB or above

4. Stable network connection

5. Smartphone Emulators/Devices

3.3 USER REQUIREMENTS

1. Network Browser such as Chrome, Edge or Firefox

2. Android Mobile Phone

4. SYSTEM DESIGN

4.1 PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture for our project entails a scalable and modular design aimed at accommodating the dynamic requirements of a decentralized exchange platform. At its core, the architecture comprises several key components, including but not limited to a front-end user interface, a back-end trading engine, smart contract infrastructure, and external integrations with blockchain networks. The front-end interface provides users with an intuitive and seamless trading experience, featuring real-time market data, order book visualization, and transaction execution capabilities. Behind the scenes, the back-end trading engine orchestrates order matching, trade execution, and portfolio management functionalities, leveraging advanced algorithms to optimize trading efficiency and liquidity. Smart contract infrastructure facilitates secure and transparent token swapping, custody, and settlement, ensuring trustless transactions on the blockchain. Additionally, external integrations enable interoperability with various blockchain networks, allowing for cross-chain asset transfers and enhanced liquidity sourcing.

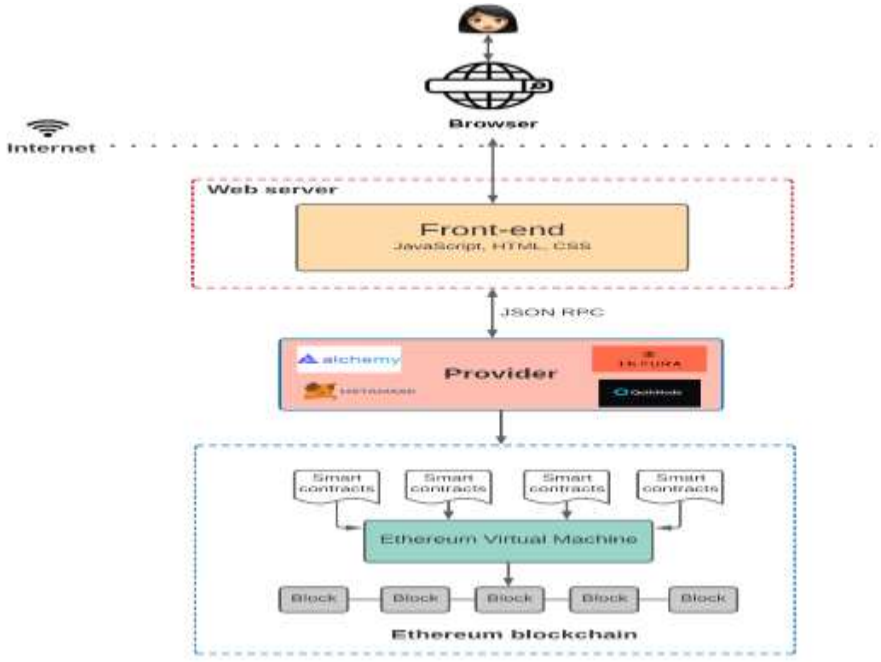

Figure 1 System Architecture

4.2 PROPOSED METHOD

This research aims to address these challenges by proposing the development of a cryptocurrency exchange app using Flutter for the frontend, Infura for Ethereum blockchain interaction, and Solidity for smart contract implementation. The focus will be on facilitating the swapping of two ERC-20 tokens, EVT and DVT, to provide users with a decentralized and secure platform for token trading. The research will explore how the chosen technologies enhance the overall user experience, contribute to financial inclusion, and support the principles of decentralized finance (DeFi). The goal is to bridge the gap between traditional centralized exchanges and decentralized alternatives, creating a user-friendly, cross-platform application that empowers users to engage in secure and transparent token swapping on the Ethereum blockchain.

4.3 UML DIAGRAMS

1.Use Case Diagram

1. **Admin and User Interaction:**

The flowchart illustrates the interactions between two main entities: Admin and User.

Both Admin and User have the capability to connect with a wallet.

2. **User Functionality:**

The User can perform the following actions:

Connect with Wallet: This step allows the User to link their wallet to the Crypto Exchange platform.

Swap Tokens: The User can exchange one cryptocurrency for another. The flowchart includes two smaller ovals labeled “Account 1 balance” and “Account 2 balance”, which likely represent the balances of different tokens.

The User interacts directly with the Ethereum rectangles, indicating that they can swap tokens involving these cryptocurrencies.

3. **Admin Functionality:**

The Admin has additional responsibilities:

Manage Users: Admins oversee user accounts, ensuring smooth operations and resolving any issues.

View Real-Time Prices: Admins can monitor the real-time prices ofEthereum.

4. System Overview:

The Crypto Exchange system facilitates seamless token swapping between users.

It provides a user-friendly interface for managing transactions and viewing cryptocurrency prices.

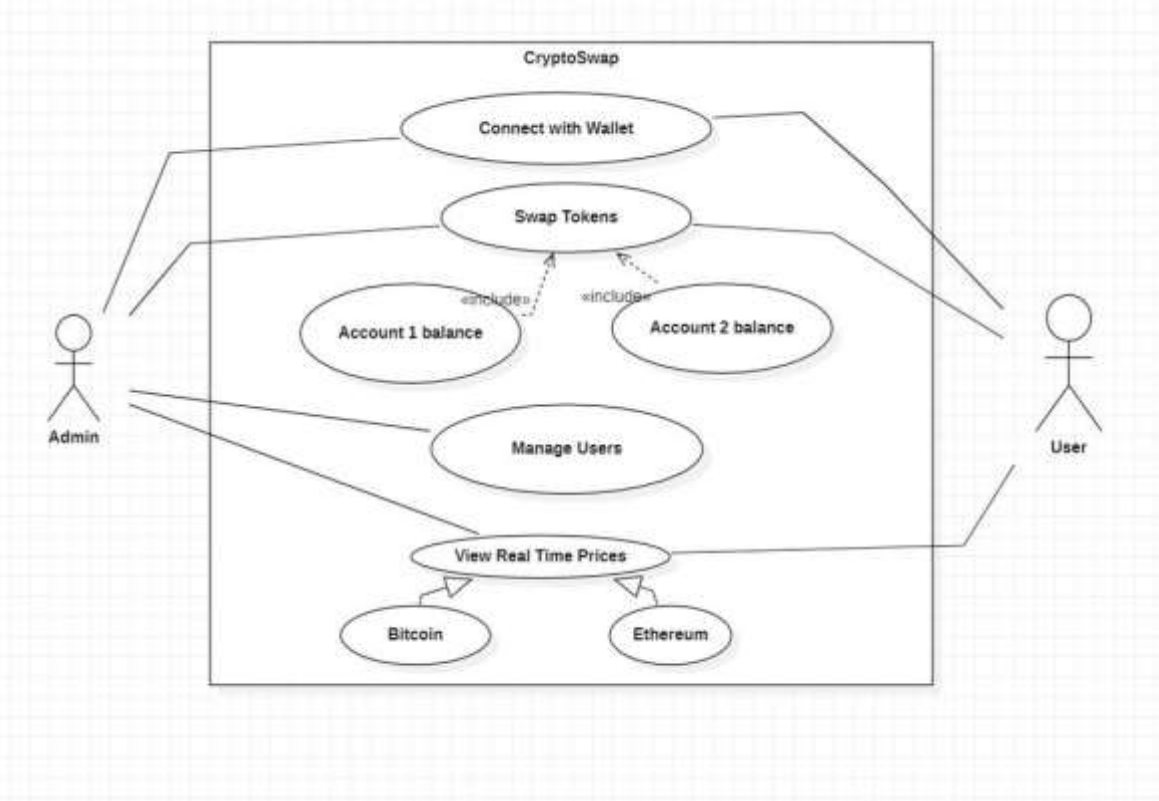

Figure 2 Use Case Diagram

2.Activity Diagram

1.**Connect to Wallet:**

This step involves verifying the user's credentials to connect their wallet to the Crypto Exchange platform.

The user likely provides authentication details (such as private keys or wallet addresses) to establish the connection.

2.**Credentials Verified:**

After the user submits their credentials, the system verifies their authenticity.

If the credentials are valid, the process proceeds; otherwise, an error message may be displayed.

3.**Check Balance:**

Once connected, the system accesses the user's account details.

It retrieves the balance associated with the connected wallet.

The balance is then displayed to the user.

4.**Token Swap Initiation:**

If the user intends to swap tokens (for example, exchanging Bitcoin for Ethereum), they initiate the process.

The user provides relevant details for the swap, such as the amount and type of tokens involved.

5.**Transaction Execution:**

If the user has sufficient balance for the token swap, the transaction is executed.

The system processes the swap based on the user's input.

If successful, the new token balance is reflected in the user's account.

6.**Insufficient Balance:**

If the user's balance is insufficient for the requested swap, an error message is displayed.

The transaction fails, and the user is informed about the reason (insufficient balance).

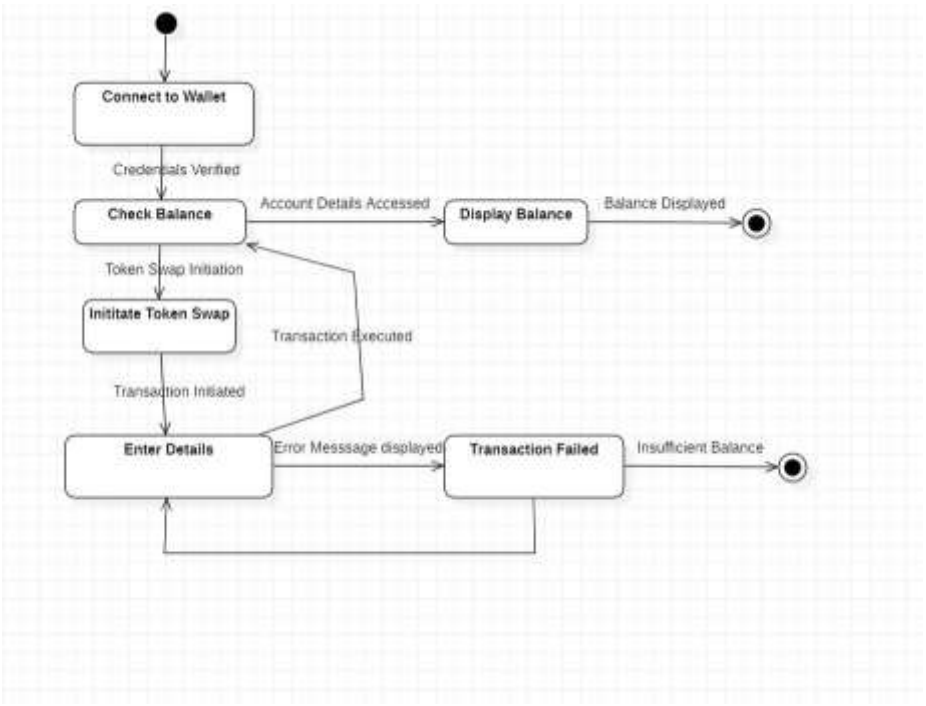

Figure 3 Activity Diagram

3.Sequence Diagram

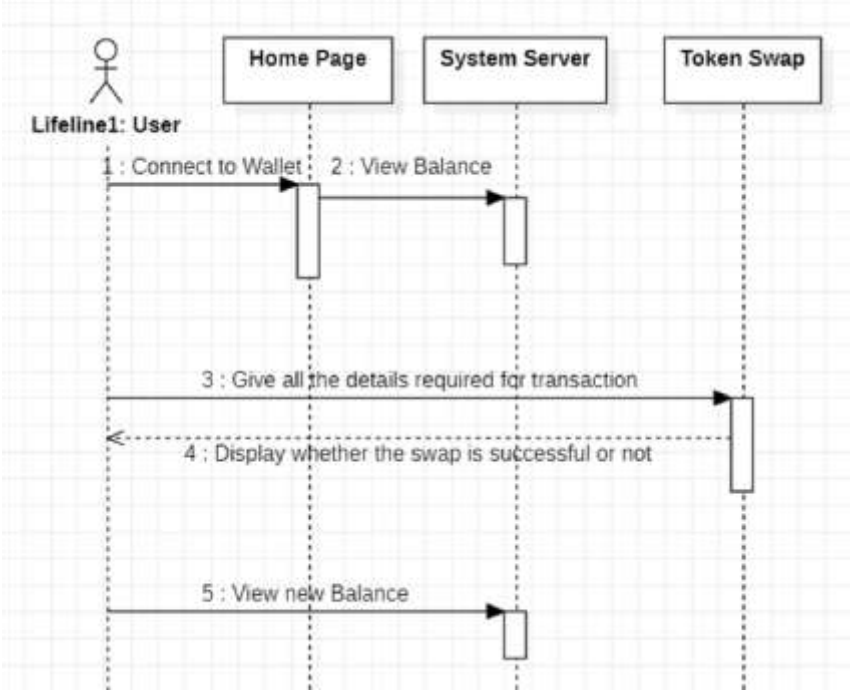

Figure 4 Sequence Diagram

4.4 TECHNOLOGY STACK

1. Market Data: Information about digital asset prices, trading volumes, order book depths, and historical price movements across different trading pairs.

2. Order Data: Details of placed orders, including order types, order sizes, order statuses (open, filled, cancelled), and timestamps.

3. User Data: User profiles, account information, transaction histories, portfolio holdings, and user preferences.

4. Liquidity Data: Data on liquidity pools, liquidity provider activities, liquidity ratios, and liquidity incentives.

5. Platform Data: Operational data such as platform settings, configurations, performance metrics, and system logs.

Technology Stack:

The technology stack for building a decentralized exchange platform can vary depending on specific requirements, preferences, and technological considerations. However, a typical technology stack may include:

1. Blockchain Protocol: Choose a suitable blockchain protocol for building the decentralized exchange platform, such as Ethereum, Binance Smart Chain, or Polkadot, depending on factors like security, scalability, and interoperability requirements.

2. Smart Contracts: Develop smart contracts using blockchain-specific programming languages like Solidity (for Ethereum) or Rust (for Polkadot) to implement core functionalities such as order matching, trade settlement, and asset custody.

3. Front-End Framework: Utilize front-end frameworks like Flutter , React.js, Angular, or Vue.js to build the user interface for the decentralized exchange platform, providing traders with an intuitive and responsive trading interface.

4. Back-End Framework: Choose a back-end framework such as Node.js, Django, or Flask to handle server-side logic, API integrations, and data processing tasks required for platform operations.

5. Database: Select a suitable database solution for storing and managing data related to users, orders, transactions, and other platform entities. Options include relational databases like PostgreSQL or MySQL, or NoSQL databases like MongoDB or Cassandra.

6. Web3 Libraries: Integrate Web3 libraries like Web3.js (for Ethereum) or Polkadot.js (for Polkadot) to interact with the blockchain network, enabling functionalities such as account management, transaction signing, and smart contract interactions.

7. Security Tools: Implement security measures such as HTTPS encryption, secure authentication mechanisms, and auditing tools to enhance the platform's security posture and protect user assets from potential threats and vulnerabilities.

5. IMPLEMENTATION

The cryptocurrency exchange app, built upon Flutter for the frontend, Infura for blockchain integration, and Solidity for smart contract development, enables users to seamlessly trade or swap EVT and DVT tokens on the Ethereum blockchain. Utilizing Flutter's cross-platform capabilities, the app boasts a user-friendly interface, allowing easy interaction with functionalities such as viewing token prices and executing swaps. Through Infura's service, the app establishes a direct connection to the Ethereum network without requiring users to run their own nodes, facilitating real-time interaction with smart contracts deployed on the blockchain. Smart contracts, written in Solidity, define the rules and logic for token swapping, ensuring transparency and security throughout the process, including price calculation, token balances, and transfer functions. This integrated approach provides users with a streamlined and efficient platform for managing their digital assets, redefining the landscape of decentralized finance.

Tokens.dart :

```
import 'package:flutter/material.dart';
import 'package:my_app/pages/dvttokens.dart';
class tokens extends StatefulWidget{
const tokens({Key? key}):super(key:key);
@override
State<tokens>createState()=>_tokens();
}
class _tokens extends State<tokens>{
  String address="";

@override
Widget build(BuildContext context){
var controller=TextEditingController();

    return Scaffold(
appBar: AppBar(
          title: Text("Token Details"),
backgroundColor: Color.fromARGB(255, 138, 246, 197),
foregroundColor: Colors.black54,

),

body: Center(
            child: Column(
mainAxisAlignment: MainAxisAlignment.center, // Align the children in
```

```
the center
children:[
SizedBox(height: 30,),
TextField(
                    controller: controller,
decoration:
InputDecoration(
border:OutlineInputBorder(
borderRadius: BorderRadius.circular(19),
),
hintText: "Enter your Metamask address",
contentPadding: EdgeInsets.symmetric(horizontal: 75.0),

),
),
ElevatedButton(
onPressed: (){
setState(() {
address=controller.text;
});
Navigator.push(context, MaterialPageRoute(builder:
(context)=>Tokeninfo(address:address)));
},
child: const Text("Get Balance"),
style:ButtonStyle(
backgroundColor: MaterialStateProperty.all(Color.fromARGB(255, 249, 211,
147)),
foregroundColor: MaterialStateProperty.all(Colors.black45),

shape:MaterialStateProperty.all<RoundedRectangleBorder>(
RoundedRectangleBorder(
borderRadius: BorderRadius.circular(18.0),

)
                          )
// shadowColor: MaterialState,
)

                  )
                ]
            )
        )
    );
}

}
```

RealSwap.dart :

```
var controller=TextEditingController();
    var controller2=TextEditingController();
    return Scaffold(
appBar: AppBar(
          title: Text("BlockchainATM"),
// backgroundColor: Color.fromARGB(255, 42, 147, 239),
backgroundColor: Color.fromARGB(255, 138, 246, 197),
foregroundColor: Colors.black54,
),
body:(
Center(
child:Column(
                    children:[
SizedBox(height: 110,),
TextField(
controller:controller,
decoration: InputDecoration(
border:OutlineInputBorder(
gapPadding: 50,
borderRadius: BorderRadius.circular(21),
),
hintText: "Enter Amount of DVT Tokens"
),
),
SizedBox(height: 20,),
TextField(
                        controller:controller2,
decoration: InputDecoration(
border:OutlineInputBorder(
gapPadding: 50,
borderRadius: BorderRadius.circular(21),
),
hintText: "Enter Amount of EVT Tokens"
),
),
SizedBox(
                           width: 100,
child:ElevatedButton(onPressed: ()async{
setState(() {
amount = BigInt.parse(controller.text) * BigInt.from(10).pow(18);
});
setState(() {
amount1 = BigInt.parse(controller2.text) * BigInt.from(10).pow(18);
});

ans=await swapfunction("swap", [amount,amount1]);
store=ans;

}, child: const Text("Swap"),style: ButtonStyle(
backgroundColor: MaterialStateProperty.all(Color.fromARGB(255, 237, 209,
107)),
```

```
//  backgroundColor: MaterialStateProperty.all(Color.fromARGB(255, 249,
211, 147)),
foregroundColor: MaterialStateProperty.all(Colors.black45),

shape:MaterialStateProperty.all<RoundedRectangleBorder>(
RoundedRectangleBorder(

borderRadius: BorderRadius.circular(14.0)
                                    )
                                ),
),)),
Text(store,style: const TextStyle(color: Colors.blue),)
                    ])))
    );




}
}
```

main.dart :

```
import 'package:flutter/material.dart';
import 'package:my_app/utils/routes.dart';
import 'package:my_app/pages/login_page.dart';

void main(List<String>args) {
runApp(const MyApp());
}

class MyAppextends StatelessWidget {
const MyApp({Key? key}) : super(key: key);

@override
Widget build(BuildContext context) {
return MaterialApp(
debugShowCheckedModeBanner: false,
// theme:ThemeData{
      //   Brightness:Brightness
      // }
initialRoute: MyRoutes.loginRoute,
routes: {
MyRoutes.loginRoute: (context) =>const LoginPage(),
},
);
}
}
```

DVT.sol :

```
// SPDX-License-Identifier: GPL-3.0

pragmasolidity^0.8.0;

import"@openzeppelin/contracts/token/ERC20/ERC20.sol";

contractDevTokenis ERC20{
    constructor() ERC20("DevToken","DVT"){
        _mint(msg.sender,1000*10**18);
    }
}
```

EVT.sol :

```
// SPDX-License-Identifier: GPL-3.0

pragmasolidity^0.8.0;

import"@openzeppelin/contracts/token/ERC20/ERC20.sol";

contractDevTokenis ERC20{
    constructor() ERC20("DevToken","EVT"){
        _mint(msg.sender,1000*10**18);
    }
}
```

TokenSwap.sol :

```
// SPDX-License-Identifier: MIT
pragmasolidity^0.8.17;

import"https://github.com/OpenZeppelin/openzeppelin-
contracts/blob/v4.0.0/contracts/token/ERC20/IERC20.sol";


contractTokenSwap{
    IERC20 public token1;
    addresspublic owner1;
    uintpublic amount1;
    IERC20 public token2;
```

```
    addresspublic owner2;
    uintpublic amount2;

    constructor(
        address _token1,
        address _owner1,
        uint _amount1,
        address _token2,
        address _owner2,
        uint _amount2
    ){
        token1 = IERC20(_token1);
        owner1 = _owner1;
        amount1 = _amount1;
        token2 = IERC20(_token2);
        owner2 = _owner2;
        amount2 = _amount2;
    }

    function swap()public{
        require(msg.sender== owner1 ||msg.sender== owner2,"Not
authorized");
        require(
            token1.allowance(owner1,address(this))>= amount1,
            "Token 1 allowance too low"
        );
        require(
            token2.allowance(owner2,address(this))>= amount2,
            "Token 2 allowance too low"
        );

        _safeTransferFrom(token1, owner1, owner2, amount1);
        _safeTransferFrom(token2, owner2, owner1, amount2);
    }

    function _safeTransferFrom(
        IERC20 token,
        address sender,
        address recipient,
        uint amount
    )private{
        bool sent =token.transferFrom(sender, recipient, amount);
        require(sent,"Token transfer failed");
    }
}
```

5.1 FRONT PAGE SCREENSHOT

1. **Main Page Overview:**

i. The home page appears to be for a cryptocurrency application or website.

ii. It features a light green background.

iii. At the top, the title “Main Page” is displayed in black text.

iv. Below the title, there is an icon of stacked coins representing Ethereum.

2**. Ethereum Price**:

A green button labeled “Ethereum Price” likely leads to a section where users can check the current price of Ethereum.

3. **Connect with Metamask:**

A white button labeled “Connect with Metamask” suggests functionality for linking the app with a Metamask wallet. Metamask is a popular crypto wallet used for managing Ethereum and other tokens.

4. **Swap Tokens:**

A yellow button at the bottom labeled “Swap Tokens” indicates a feature that allows users to exchange different types of cryptocurrency tokens

Figure 5 Home Page

5.2 RESULTS

Figure 6Token details before Transaction

Figure 7Swap Token Page

Figure 8Token details after Transaction

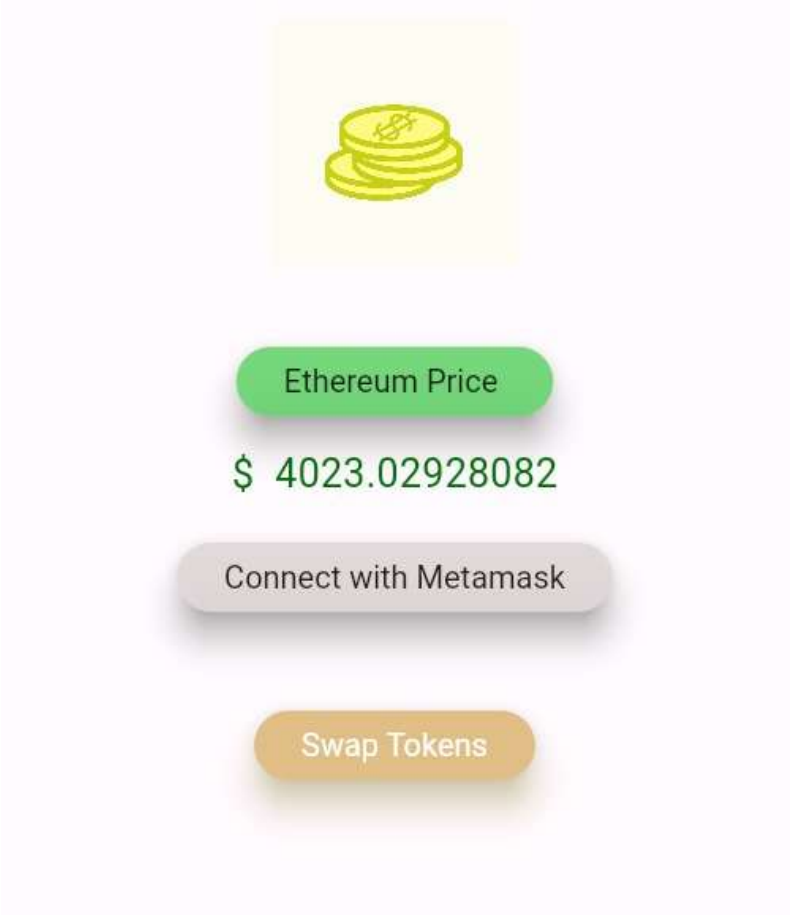

Figure 9Current Ethereum price

Figure 10 Get Balance

Figure 11Main Page

Figure 12Load Swap Contract Page

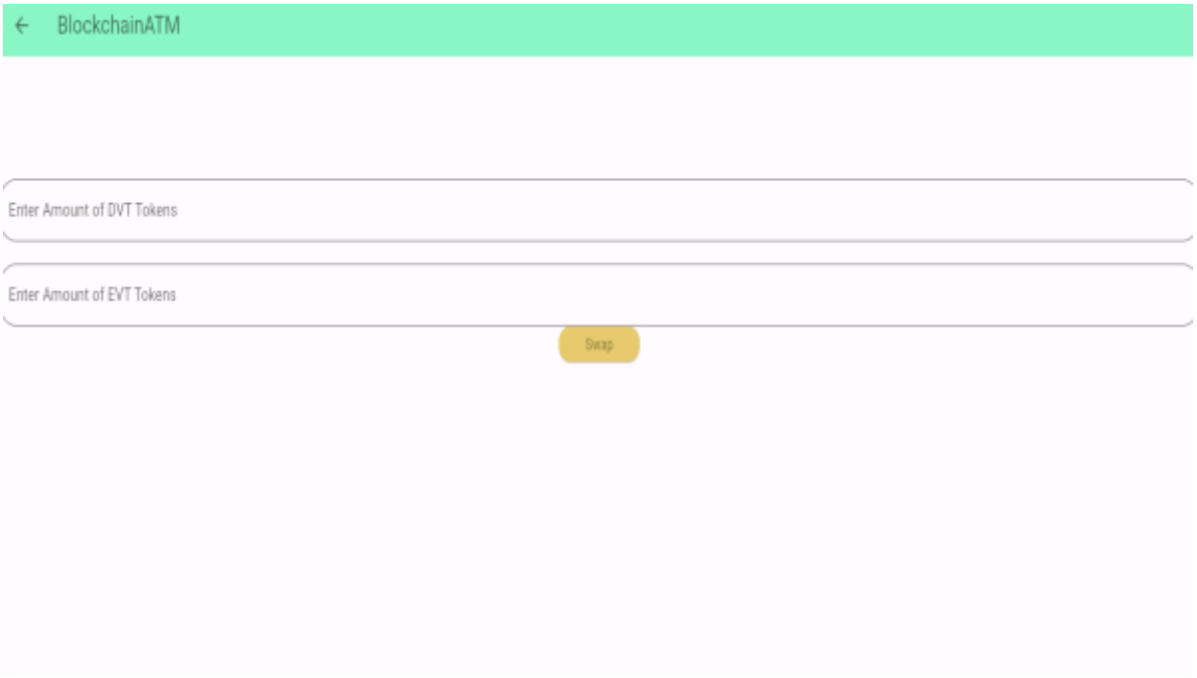

Figure 13Token Exchange Page

5.3 TESTING

Testing is a critical aspect of the development process for a decentralized exchange platform, ensuring that the platform functions reliably, securely, and in accordance with user requirements. The testing process encompasses various types of testing methodologies, each serving a distinct purpose in evaluating different aspects of the platform's functionality and performance. Unit testing involves testing individual components or modules of the platform to verify their functionality in isolation, while integration testing focuses on testing the interactions between different components to ensure seamless integration and compatibility. End-to-end testing evaluates the platform's functionality across multiple components, simulating real-world user scenarios to validate its overall performance and user experience. Security testing is crucial for identifying and mitigating potential security vulnerabilities and threats, while performance testing assesses the platform's performance under different conditions to identify bottlenecks and scalability limitations. Usability testing gathers feedback from real users to evaluate the platform's user interface and experience, while regression testing ensures that recent code changes do not introduce new bugs or regressions. Additionally, compliance testing verifies that the platform complies with relevant legal and regulatory requirements. By conducting thorough testing across these different areas, developers can ensure that the decentralized.

5.4 VALIDATION

Validation is a pivotal stage in the development of a decentralized exchange platform, ensuring that the platform effectively meets the needs and expectations of its users and stakeholders. Functional validation encompasses rigorous testing of the platform's features and functionalities, including order placement, matching, and trade execution, to confirm adherence to specified requirements. Concurrently, user validation entails gathering feedback directly from users through surveys, interviews, and usability testing to assess satisfaction, usability, and overall experience, uncovering any usability issues or areas for improvement. Market validation assesses the platform's competitiveness, viability, and market fit within the cryptocurrency ecosystem, analyzing factors such as user adoption, trading volumes, and competitive positioning. Lastly, security validation involves comprehensive audits and testing to identify and mitigate potential security vulnerabilities, ensuring adherence to best practices and compliance standards. Through these validation processes, decentralized exchange platforms can validate their functionality, usability, market fit, and security, ultimately delivering a robust and reliable trading environment for users while fostering trust and confidence in the platform's capabilities.

6. CONCLUSION

6.1 CONCLUSION

In conclusion, the development of a decentralized exchange platform represents a significant step forward in the evolution of cryptocurrency trading, offering users a secure, efficient, and transparent alternative to traditional centralized exchanges. Through rigorous validation processes encompassing functional, user, market, and security validation, the platform demonstrates its ability to meet the diverse needs of users, maintain competitiveness in the market, and adhere to stringent security standards. By prioritizing user experience, market viability, and regulatory compliance, the platform establishes itself as a trusted and reliable trading venue, empowering users to participate in the decentralized finance ecosystem with confidence. As the cryptocurrency landscape continues to evolve, the decentralized exchange platform stands poised to play a pivotal role in facilitating global financial inclusion, innovation, and democratization of access to financial services. With a commitment to continuous improvement and innovation, the platform aims to set new standards for decentralized trading, driving forward the adoption of blockchain technology and reshaping the future of finance.

6.2 FUTURE SCOPE

In the realm of decentralized exchange platforms, the future holds immense potential for innovation and growth. Scalability enhancements represent a crucial avenue for exploration, with solutions such as layer 2 scaling and sharding offering promise in addressing the challenges of increasing user demand and transaction volumes. Cross-chain compatibility stands out as another area ripe for development, as interoperability with multiple blockchain networks can significantly expand the platform's reach and liquidity pools. Additionally, the introduction of advanced trading features, including algorithmic trading and decentralized lending protocols, has the potential to enhance the trading experience and attract a broader user base. Moreover, the implementation of decentralized autonomous organization (DAO) governance models can further enhance decentralization and community-driven decision-making. Strengthening security measures and ensuring regulatory compliance remain paramount considerations, necessitating ongoing investment in robust security protocols and compliance frameworks. Furthermore, initiatives focused on user education, community engagement, and strategic partnerships can foster growth and sustainability, while continuous research and development efforts ensure that the platform remains at the forefront of technological innovation. By embracing these avenues of future scope, decentralized exchange platforms can continue to evolve as vital pillars of the decentralized finance ecosystem, empowering users and driving the adoption of blockchain technology on a global scale.

REFERENCES

[1] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., & Felten, E. W. (2022). SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In Security and Privacy (SP), 2015 IEEE Symposium on.

[2] Ahmad, F., Malik, M. I., & Mushtaq, Q. (2021). Flutter: A New Evolution in Mobile App Development. In 2021 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET).

[3] Griffiths, I., & Hinchcliffe, R. (2020). Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2. Packt Publishing Ltd

[4] Swan, M. (2019). Blockchain: blueprint for a new economy. O'Reilly Media, Inc.

[5] Wüst, K., & Gervais, A. (2018). Do you need a blockchain? In 2018 Crypto Valley Conference on Blockchain Technology (CVCBT).

[6] Atzei, N., Bartoletti, M., &Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts (SoK). In Principles of Security and Trust (POST).