សាកលវិទ្យាល័យជាតិគ្រប់គ្រង

NATIONAL UNIVERSITY OF MANAGEMENT

THESIS

USING DECENTRALIZED APPLICATION (DAPP) AND API TO SOLVE SECURITY AND INTEROPERABILITY PROBLEMS OF DIGITAL IDENTITY

BY

NGY SAMLENG

Phnom Penh
2023

FACULTY OF
DIGITAL ECONOMY

MINISTRY OF EDUCATION, YOUTH AND SPORT

NATIONAL UNIVERSITY OF MANAGEMENT

FACULTY OF DIGITAL ECONOMY

Using Decentralized Application (DaPP) and API to solve security and interoperability problems of digital identity

BY

NGY SAMLENG

Project Thesis Submitted in Partial Fulfillment of the Requirements for
the Bachelor of Financial Technology
(English-Based Program)

SPECIALIZATION IN
DIGITAL ECONOMY

Phnom Penh, Cambodia

October 2023

MINISTRY OF EDUCATION, YOUTH AND SPORT

NATIONAL UNIVERSITY OF MANAGEMENT

FACULTY OF DIGITAL ECONOMY

Using Decentralized Application (DaPP) and API to solve security and interoperability problems of digital identity

BY

NGY SAMLENG

Project Thesis Submitted in Partial Fulfillment of the Requirements for
the Bachelor of Financial Technology
(English-Based Program)

Supervised by:

____

Dr. Samreth Sovannroeun

Associate Professor at the Graduate School of Humanities and Social Science, Saitama University, Japan

Phnom Penh, Cambodia

October 2023

COMMITTEE APPROVAL

The members of the committee approved the thesis of **Ngy Samleng** defended on October 20th, 2023

Mr. KLEUNG Sinet (Chairman)

Dr. Samreth Sovannroeun (Committee Member)

Asst. Professor CHAY Sengtha (Committee Member)

DECLARATION

I declare that all the work in this thesis was conducted by my own and has not been used for any possible purpose before. All the information, words and numbers included in this thesis retrieved from published and unpublished work of others have been appreciated and listed down in the given reference.

Ngy Samleng

National University of Management

Phnom Penh, Cambodia

October 2023

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, Dr. Samreth Sovannroeun, for their invaluable guidance, unwavering support, and insightful feedback throughout the entire process of crafting this thesis. Their expertise and encouragement have been instrumental in shaping this work. I am immensely thankful to the members of my thesis committee, Professor Phim Runsinarith, and Mr. Kleung Sinet, for their efficient and supportive management of the thesis process and for their recommendations, suggestions, and expertise that greatly enriched the content of this thesis. I extend my thanks to my family for their patience, understanding, and constant motivation. Their unwavering support has been a pillar of strength throughout this academic journey. Finally, I would like to thank all those individuals whose names might not appear here but who, in various ways, contributed to this thesis.

ABSTRACT

In an era marked by digital transformation and the proliferation of online services, securing and managing digital identities is of paramount importance. Traditional centralized identity solutions have raised significant security and interoperability concerns. This thesis delves into the development and implementation of a Decentralized Application and Application Programming Interface as innovative solutions to address the multifaceted challenges facing digital identity. The study begins with a comprehensive literature review, revealing the vulnerabilities and limitations of current digital identity systems and emphasizing the potential of blockchain technology. Leveraging a robust theoretical framework, encompassing blockchain, decentralized identity standards, smart contracts, and APIs, this research establishes the conceptual framework foundation for the proposed solutions. The DaPP and API are designed to not only bolster the security of digital identities but also enhance interoperability between disparate systems and platforms. The DaPP utilizes blockchain technology, smart contracts, and decentralized identity standards, while the API facilitates seamless communication between various components. These solutions are informed by an extensive exploration of cryptographic principles and blockchain security measures.

Keywords: Blockchain; Smart Contract; API; DaPP; Digital Identity

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

1.1 Background

NID is at the forefront of the WEB3 revolution revolutionizing how people manage and secure their identities. In todays interconnected world, where personal information is spread across platforms and databases NID offers a game changing solution. It provides a platform that consolidates and protects all aspects of an individual's identity credentials. At its core NID acts as a repository for essential identifiers. This innovative platform brings together data such as ID cards, banking details, utility bills, phone numbers, email addresses and important identification documents. By centralizing all this information in an ecosystem, NID simplifies access and management for its users. It offers convenience and efficiency.

Security is paramount in NIDs architecture. By using encryption protocols and leveraging technologies immutability NID ensures robust protection for user's sensitive personal data. Every piece of information stored in the NID ecosystem undergoes rigorous encryption to maintain confidentiality and prevent access. The decentralized nature of blockchain also adds a layer of security by reducing the risk of points of failure and enhancing resilience, against potential cyber threats. The versatility of NID goes beyond keeping your information secure. It allows you to smoothly use it across platforms and systems. With NID you can securely access all your identity details from anywhere at any time. This makes it easier for you to interact with applications and services while ensuring that everything is authenticated properly. This interoperability feature also simplifies processes, like verifying accounts submitting documents and confirming identities across industries. Additionally, NID puts an emphasis on user control and consent. You have ownership and authority over your data giving permission for its usage in different situations. This approach not gives you autonomy but also ensures transparency and accountability when it comes to using your data. It aligns with the changing data privacy regulations and ethical standards. In essence NID revolutionizes how digital identity management works. By combining security, interoperability and user centricity it creates an ecosystem where individuals can confidently manage their identities in the digital world. This sets benchmarks, for privacy, convenience and trust in the evolving landscape of online identity management.

1.2 Problem Statement

There are a number of challenges associated with NID digital identity. One challenge is that NID trying to solved is digital identities are often made up of multiple pieces of information that are stored in different places. This can make it difficult to manage and protect our digital identities. Another challenge is that our digital identities are not always secure. Hackers can steal our digital identities and use them to commit fraud or identity theft. As a result, the purpose of this research is to fill a knowledge vacuum by evaluating the relevance of decentralized apps and API to the interoperability of current digital identities.

1.3 Research Objectives

The research in this thesis is guided by the following key objectives: To develop a Decentralized Application that utilizes blockchain technology, smart contracts, and decentralized identity standards to enhance the security and privacy of digital identities and to design an Application Programming Interface that enables seamless communication and data exchange between various digital identity systems, fostering interoperability.

1.4 Research Questions

This thesis will address the following research questions: 1) How can a Decentralized Application be developed to enhance the security and privacy of digital identities in a decentralized manner? 2) How can an Application Programming Interface facilitate seamless communication and interoperability among disparate digital identity systems, and what are the security considerations involved?

1.5 Scope of the Thesis

This research is delimited to the development and practical implementation of the DApp and API, focusing on their roles in enhancing the security and interoperability of digital identity systems. While acknowledging the vastness of the digital identity landscape, this research provides a focused exploration into innovative solutions in line with the defined objectives.

CHAPTER TWO: LITERATURE REVIEW

2.1 Blockchain Technology and Security

There have been attempts, at defining blockchain. They often tend to be intricate and hard to understand. The purpose of this section is to familiarize you with technology explain its workings and provide some historical background. We will briefly explore the components of blockchain highlighting their characteristics and unique features. The main objective of blockchain technology is to store digital information and enable its distribution while ensuring that the data remains unchanged. Stuart Haber and W. Scott Stornetta introduced the concept of blockchain in their article "How to timestamp a document" back, in 1991. Their aim was to establish a mechanism that would prevent tampering with document timestamps. However, it wasn't until 2009 when an individual known as Nakamoto Santoshi launched the blockchain based platform called Bitcoin that their proposal was put into action. According to Nakamoto (2008) Bitcoin is a payment system that relies on mechanisms instead of central authorities, such as banks to control transactions. This allows for peer to peer transactions without the need for a trusted party, like a central bank.

According to a study of (Haiyan Kang, Xiameng Si, and Boyu Liu, 2022) blockchain have 3 main feature it includes Trust and Transparency: The blockchain technology operates in a manner creating a network where peers can interact with each other. In this system all nodes have rights to send and receive messages and every transaction within the network is visible to all nodes ensuring transparency. Immutable and traceability: The concept of immutability plays a role in technology. It ensures that there is no possibility of data or records. Immutability is achieved through two aspects of the structure and mechanism. As we have discussed earlier if any data within a block is modified it alters the information contained in that block. Consequently, this change renders the following block to Recognize it ultimately leading to the deletion of the blockchain. Privacy and Security: The blockchain system is a decentralized system that can accomplish user privacy and security without relying on third-party protection.

2.2 Decentralized Application

A decentralized application is an application built on a decentralized network that combines a smart contract and a front-end user interface (Ethereum white paper 2023). According to ethereum white paper a DAPP has its backend code running on a decentralized peer-to-peer network. Contrast this with an app where the backend code is running on centralized servers. A Dapp can have frontend code and user interfaces written in any language (just like an app) to make calls to its backend. Furthermore, its frontend can get hosted on decentralized storage such as decentralized - Dapp operate on Ethereum, an open public decentralized platform where no one person or group has control Deterministic - DaPP perform the same function irrespective of the environment in which they get executed, Turing complete - Dapp can perform any action given the required resources Isolated - Dapp are executed in a virtual environment known as Ethereum Virtual Machine so that if the smart contract has a bug, it won't hamper the normal functioning of the blockchain network.

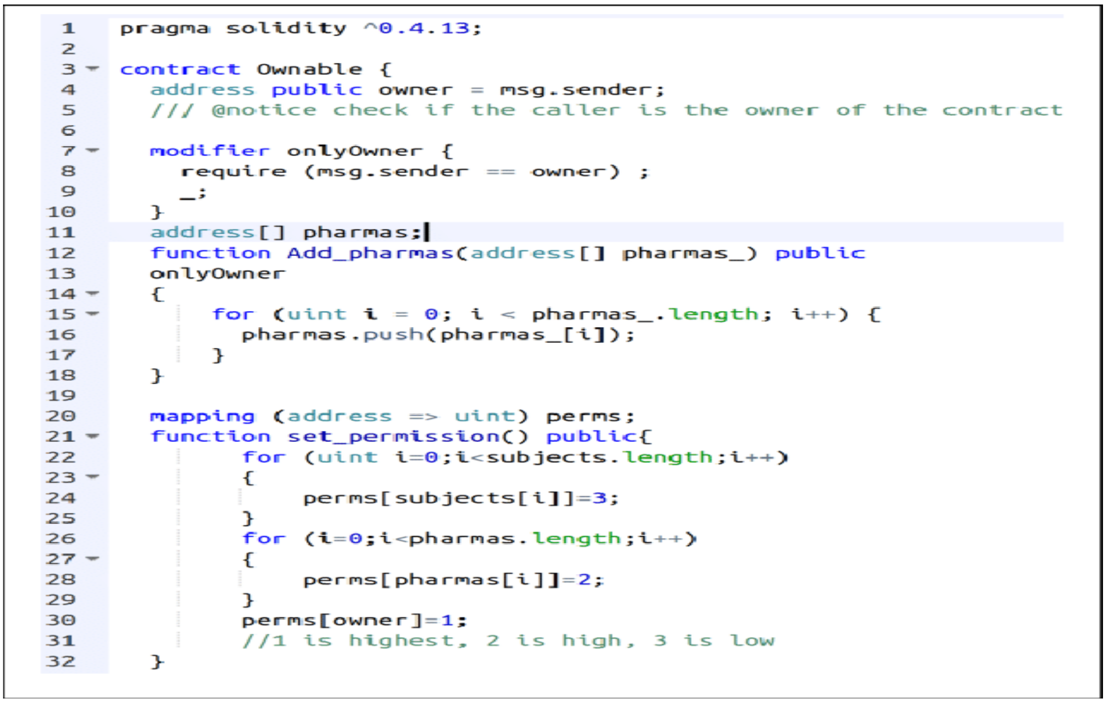
2.3 Smart Contract

A smart contract is a computerized transaction protocol that executes the terms of a contract (Szabo, 1996). It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain. There are similarities, between contracts and smart contracts. In contracts, people engage with each other. Document their agreements using computers or pen and paper. On the hand, smart contracts are entirely written in code. They capture all the terms of the agreement in code form. Automatically carry them out. Smart contracts rely on a blockchain network, for deployment. Unlike contracts that involve one or more party's smart contracts often serve as a means to record and enforce the rules and commitments made by parties.

Once certain conditions are met these smart contracts execute automatically. Smart contracts operate on blockchains and are decentralized, eliminating the need for intermediaries such as banks (Ma et al., 2019). Smart contracts, on the other hand, differ. The major difference between the Bitcoin protocol and the Ethereum protocol, for example, is that the former lacks a full tour. As a comprehensive application, Ethereum contains smart contracts with a full range of functionalities (Collins, 2021).

However, unlike Ethereum, Bitcoin has smart contracts. Bitcoin developers purposefully made this change because they see the Bitcoin network as an asset.

Figure 1: *Example of Smart Contract*



```
pragma solidity ^0.4.13;

contract Ownable {
  address public owner = msg.sender;
  /// @notice check if the caller is the owner of the contract

  modifier onlyOwner {
    require (msg.sender == owner) ;
    _;
  }
  address[] pharmas;
  function Add_pharmas(address[] pharmas_) public
  onlyOwner
  {
      for (uint i = 0; i < pharmas_.length; i++) {
        pharmas.push(pharmas_[i]);
      }
  }

  mapping (address => uint) perms;
  function set_permission() public{
        for (uint i=0;i<subjects.length;i++)
        {
            perms[subjects[i]]=3;
        }
        for (i=0;i<pharmas.length;i++)
        {
            perms[pharmas[i]]=2;
        }
        perms[owner]=1;
        //1 is highest, 2 is high, 3 is low
  }
```

Source: https://www.researchgate.net/figure/A-Smart-Contract-example-demonstrating- ownership-and-permission-levels-of-different-roles_fig1_331439661

2.4 Introduction to API

APIs, or application programming interfaces, are essential tools for developers, allowing them to access and utilize the services of other software. They play a crucial role in coordinating collaborative software development, serving as contracts, boundaries, and communication mechanisms (Souza, 2009). However, they can be challenging to learn and use due to their complexity and the need for clear documentation (Scaffidi, 2006). Despite these challenges, APIs are crucial for creating innovative applications that enhance user experience (De, 2017). They define the vocabulary and calling conventions for requesting services from libraries and operating systems (Tollerud, 2016).

2.5 The Definition of API

These papers offer insights into the definition and various aspects surrounding Application Programming Interfaces (APIs). According to Tollerud (2016) an API is an interface that enables applications to request services from libraries and operating systems. Rauf (2019) focuses on methods for evaluating the usability of APIs emphasizing the importance of assessing their ease of use. Shishmano (2021) delves into the significance of API strategy in developing ecosystems and introduces the concept of API economy. Macvean (2016) explores how API usability can be evaluated at a scale discussing techniques such as log file analysis and customer satisfaction surveys to gauge user sentiment towards APIs. In summary these papers collectively define APIs, underscore the significance of evaluating their usability, explore API strategy within ecosystems and delve into methodologies for assessing user satisfaction, with APIs.

CHAPTER THREE: RESEARCH METHODOLOGY

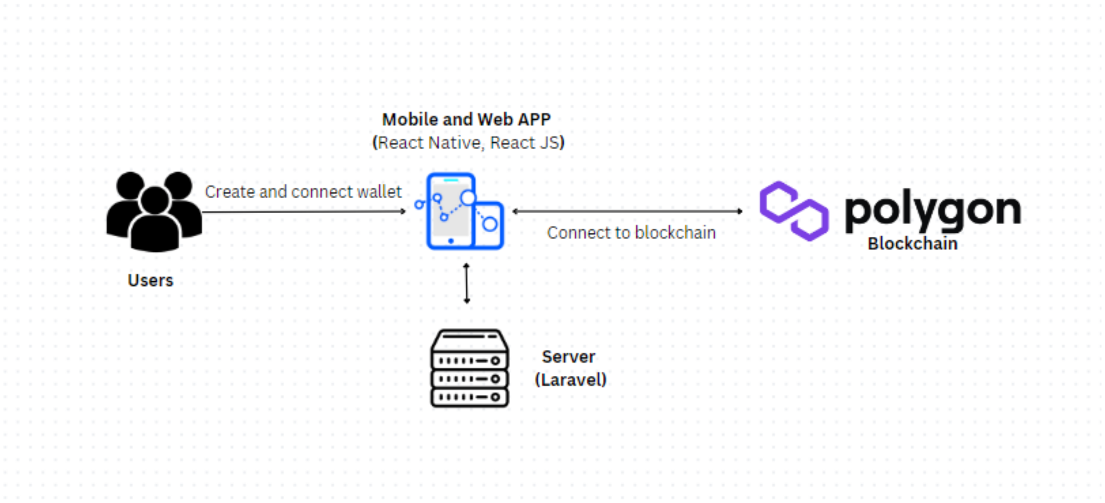
3.1 RESEARCH STRATEGY

In this paper, the researchers will employ methodology by developing a decentralized application (DaPP) aimed at effectively addressing and resolving the research question at hand. The proposed methodology involves a comprehensive approach that integrates innovative technologies implements, and evaluates the DaPP's functionality and performance in providing solutions pertinent to the research.

3.2 Overview of NID Platforms

NID (NFT-based Identity) is a Web3 platform that allows users to create a digital ID that stores a variety of IDs in one place, including phone numbers, email addresses, bank numbers, national IDs, utility bill IDs, and many others.

Figure 2: NID DaPP Framework



Creating a wallet within the NID app suggests that users might have the ability to manage their digital assets, possibly including cryptocurrencies or tokens on the Polygon blockchain. React Native is a popular framework for building mobile applications, known for its ability to create cross-platform apps. Laravel, on the other hand, is a PHP-based backend framework known for its elegant syntax and developer-friendly features. When an app interacts with a blockchain like Polygon, it likely uses specific protocols or APIs to facilitate transactions, data retrieval, or smart contract interactions.

3.3 NID Features

NID introduces a way to manage identities by utilizing technology, ensuring secure and decentralized storage. This groundbreaking approach eliminates the vulnerabilities associated with data storage, making it more resistant to hacking and breaches. With NID, users have a platform that allows them to easily handle digital identities such as phone numbers, email addresses, bank accounts, national IDs, and more. This consolidated system simplifies identity management by eliminating the need for logins and passwords. One of the aspects of NID is that it puts users in control of their identities. Users have ownership and authority over their data access permissions and usage, which enhances privacy protection and enables decision-making about their digital presence. The identities created through NID are designed to be seamlessly interoperable across platforms and applications, eliminating the hassle of verification. Moreover, NID offers privacy-enhancing features like zero-knowledge proofs that allow identity validation without compromising information. Transparency is also emphasized, as all transactions and data access within NID are meticulously recorded on the blockchain. This ensures auditability while enabling users to track their data access history.

In addition, NID embraces community-centric values through governance mechanisms that encourage user participation in shaping the platform's growth. NID's dedication to open-source infrastructure and protocols is particularly noteworthy. This approach promotes innovation. encourages the development of solutions driven by the community. The platform serves as a shining example of a user-focused and community-led identity management solution that has the potential to bring about changes in the digital realm. These features make NID a promising solution for secure, decentralized, and user-centric identity management in the Web3 era.

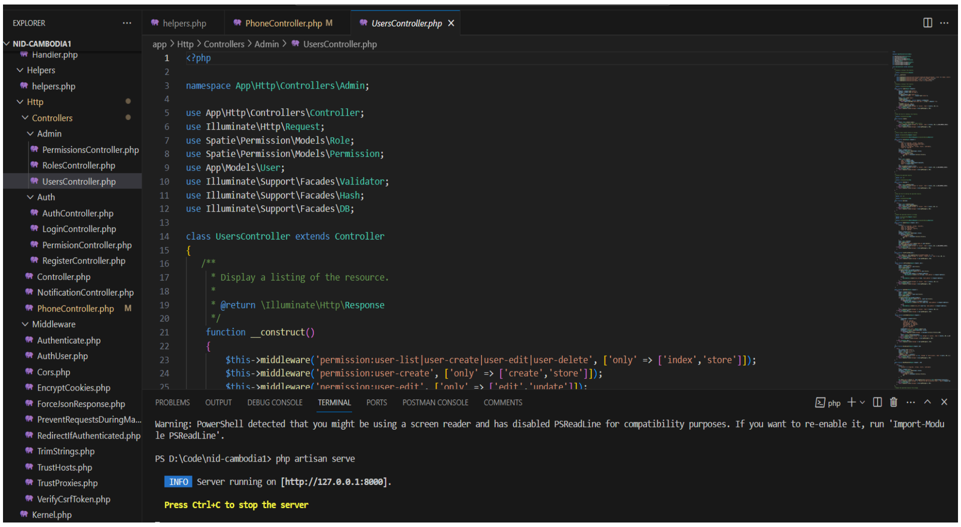
3.4 Frontend Framework

The react native framework will be used for building applications. React Native, an open source UI software framework developed by Meta Platforms, Inc. allows developers to create applications for platforms such as Android, Android TV, iOS, macOS, tvOS, Web, Windows and UWP. It combines the power of the React framework, with native platform features to facilitate application development. The app will let us create the wallet, and generate the digital identity in the application, one we have that id, that is one id can connect your every id into one place. The front end of a Dapp utilizes technologies, like HTML, CSS and JavaScript to render webpages or mobile applications. However, of communicating with a server it interacts with a network and smart contracts in the case of smart contract networks. Most DaPPs still have centralized user interfaces. This aligns with the decentralization philosophy since components are stored on the blockchain. Nevertheless, some dApps are beginning to adopt storage protocols like the Inter Planetary File System (IPFS) for storing front end files (Voshmgir, 2020). Additionally, the front end also operates an application known as a "wallet." This wallet manages the connection to the contract. Maintains records of public private key pairs and blockchain addresses. This ensures that network nodes have identities for interactions, with the network.

3.5 Backend Technology

Laravel is the one of the backend technologies that will be used to create the project. Laravel is a backend framework that offers a wide range of features, for building contemporary web applications. It provides functionalities such as routing, validation, caching, queues, file storage and more, however it is crucial to offer developers a full stack experience that includes methods for designing the frontend of their applications. We use laravel framework for building api, the api that we build for any organization or institution that want integrate with our app.

Figure 3: *Backend NID source-code*



```php
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Spatie\Permission\Models\Role;
use Spatie\Permission\Models\Permission;
use App\Models\User;
use Illuminate\Support\Facades\Validator;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\DB;

class UsersController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    function __construct()
    {
        $this->middleware('permission:user-list|user-create|user-edit|user-delete', ['only' => ['index','store']]);
        $this->middleware('permission:user-create', ['only' => ['create','store']]);
        $this->middleware('permission:user-edit', ['only' => ['edit','update']]);
```

```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Modu
le PSReadLine'.

PS D:\Code\nid-cambodia1> php artisan serve

  INFO  Server running on [http://127.0.0.1:8000].

  Press Ctrl+C to stop the server
```

3.6 LIBRARY AND TOOL

NID is employing ethers.js in our DaPP development by providing a reliable and efficient toolkit tailored specifically for Ethereum. Its intuitive interface and well-documented functionalities simplify complex interactions with smart contracts, enabling swift deployment, seamless function calls, and precise event management within the Ethereum network. Remix for deploy ERC-20 token, we also using alchemy APIs to interact with various blockchain networks, including Ethereum. It offers a range of services that simplify blockchain development, such as reliable and scalable infrastructure, developer tools, APIs, and analytics. Developers use Alchemy to access and interact with blockchain networks without managing their own nodes or infrastructure. OpenZeppelin for building secure smart contracts. OpenZeppelin provides a complete suite of security products and audit services to build, manage, and inspect all aspects of software development and operations for decentralized applications.

CHAPTER FOUR: DISCUSSION AND FINDING

4.1 Technology

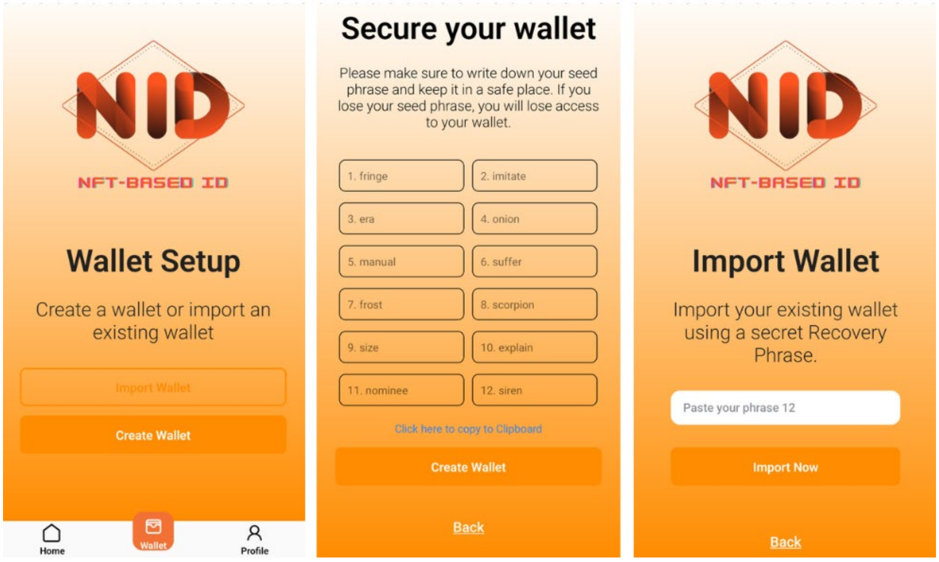
NID application uses the Polygon blockchain technology because of its security, interoperability, and scalability. Polygon is a side chain network with its native token (Matic) and validation mechanism (Proof-of-Stake), which means that the security is separate from the L1 network. Polygon is pegged to the Ethereum blockchain system, and users can transfer tokens from Polygon to Ethereum and vice versa using a bridge (Thibault et al. (2022). However, this technology secure because it has wallets address, which manage digital identities by controlling access to assets and interactions with the blockchain, contribute to security by safeguarding private keys.

4.2. Result and Discussion

In this section I am going to discuss about function and the feature of security and interoperability in NID, this include wallet address, NID token, NID NFT and NID metadata.

Wallet Address: When a user creates a wallet address, they are essentially creating a unique identifier for their cryptocurrency wallet. This address is used to send and receive cryptocurrency transactions. The process of creating a wallet address is relatively simple and can be done in a few steps. They can import existing wallet or create a new wallet.

Figure 4: *NID Wallet Setup*



Wallet Setup

Create a wallet or import an existing wallet

Secure your wallet

Please make sure to write down your seed phrase and keep it in a safe place. If you lose your seed phrase, you will lose access to your wallet.

1. fringe
2. imitate
3. era
4. onion
5. manual
6. suffer
7. frost
8. scorpion
9. size
10. explain
11. nominee
12. siren

Import Wallet

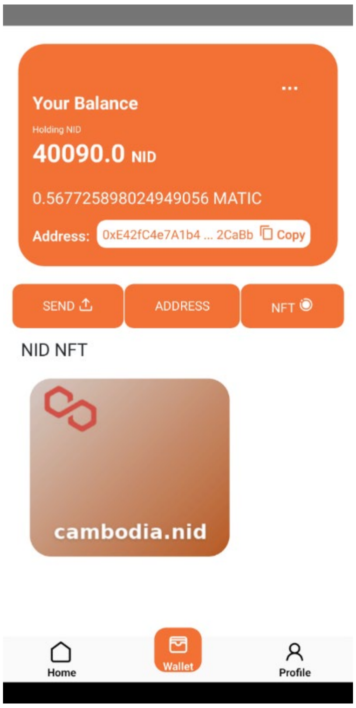
Import your existing wallet using a secret Recovery Phrase.

Paste your phrase 12

NID provide non-custodial wallet which mean that user can control and own their digital identity without rely on any third party. The wallets give users control over their funds and private keys which helps minimize the risk of hacking. By storing keys offline or on hardware devices they enhance protection against access. Moreover, decentralization reduces dependence on an entity ensuring greater security. While users are responsible, for safeguarding their keys they gain control and security over their assets. As a result, non-custodial wallets are highly recommended for individuals who prioritize security in the space.

NID NFT: The NID NFT is a unique digital token issued on the blockchain following the ERC-721 standard. This token represents an individual's digital identity on the NID platform. It contains a range of identification information, such as national ID card details, bank account information, utility bill IDs, phone numbers, email addresses, and potentially more personal data. Each NID token is distinct and serves as proof of ownership and authenticity for the associated digital identity. It can store various IDs and documents securely, allowing users to manage and control their personal information within a single digital token. By utilizing NFT technology, the NID token offers a level of security, immutability, and uniqueness that traditional digital identifiers may lack. It can also facilitate trade or transfer of digital identities within the platform's ecosystem, offering users flexibility in managing their personal information.

Figure 5: *NID NFT*



Your Balance

Holding NID

40090.0 NID

0.567725898024949056 MATIC

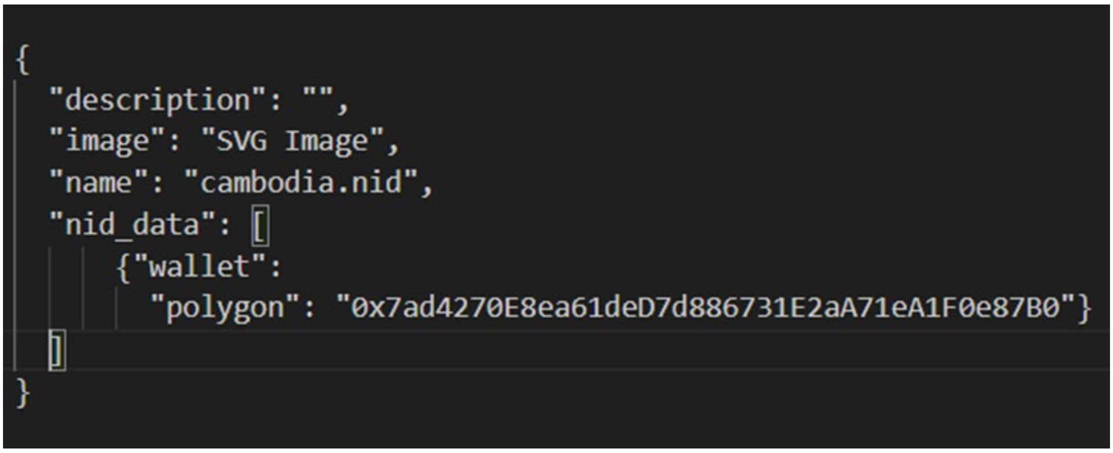
Address: 0xE42fC4e7A1b4 ... 2CaBb

NID NFT

cambodia.nid

NID Metadata: The NID platform has a system, in place for managing identity tokens. It ensures that user's information is securely stored by linking metadata and wallet addresses. Each NID token represents an identity and has a metadata link that provides additional details, descriptions, images, name, NID data, related to the token. This metadata is stored off chain. Offers insights into the represented digital identity. At the time users NID tokens are safely stored in their wallets with each wallet having a unique address on the blockchain. These wallet addresses act as gateways to access and manage the associated NID tokens ensuring the security and integrity of user's digital identities, within the NID ecosystem. By synchronizing metadata links and wallet addresses the NID platform creates a secure environment for users to effectively manage and utilize their identities.

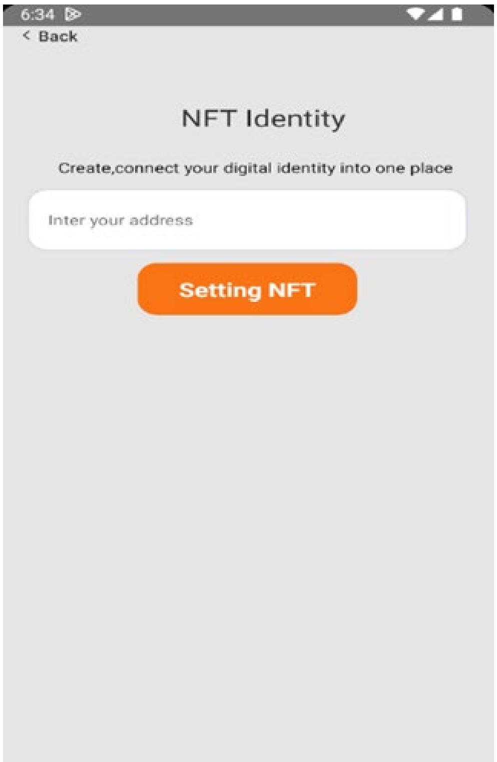
Figure 6: *NID Sample Metadata*



```
{
  "description": "",
  "image": "SVG Image",
  "name": "cambodia.nid",
  "nid_data": [
      {"wallet":
        "polygon": "0x7ad4270E8ea61deD7d886731E2aA71eA1F0e87B0"}
  ]
}
```

Description: This field appears to be empty in the sample, but user can add a description or additional information related to the ENS name or address. Image: This field expects a URL to an SVG image that get from ipfs or. Replace "Put SVG image URL here" with the actual URL. Name: This field contains the name associated with the NID as specified by NID NFT. NID data: This is an array that contains nested wallet information. It seems designed to hold data for different types of wallets associated with the address. In the provided example, it contains a wallet for Polygon. Within NID_data: Wallet: Contains information related to a specific blockchain or network. In this case, it holds information for the Polygon network. Polygon: This field is meant to store the Polygon wallet address. It's expecting input from a form or user interaction.

***Figure 7:** Setting Wallet Address*



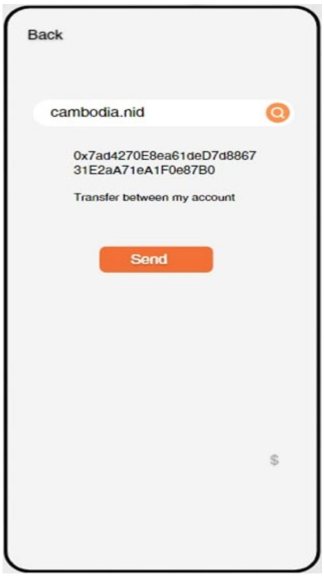
NFT Identity

Create,connect your digital identity into one place

Once the user enters the address the system will establish a connection, to NIDs metadata using achamys API. This connection enables us to access and retrieve information stored in IPFS that is associated with that address. This seamless integration between the provided address and NIDs metadata, through achamys API simplifies the process of accessing and retrieving data from the IPFS network.

After establishing the NID NFT and linking it to the corresponding NID metadata through a connection to the blockchain API, the resulting output is the unique identifier associated with the NFT. This ID encapsulates and represents the authenticated and verifiable information present within the NID's metadata, providing a secure and interoperability reference point within the blockchain ecosystem.

Figure 8: *Linking ID*



cambodia.nid

0x7ad4270E8ea61deD7d8867
31E2aA71eA1F0e87B0

Transfer between my account

In summary The NID wallet address and metadata work together to ensure both security and compatibility. The wallet address functions, as an identifier for owning and accessing NID assets while the metadata holds information connected to the NID. By incorporating encryption and blockchain technology these components enable storage of data and smooth interaction across platforms and systems. This integration guarantees protection for information while facilitating effective communication and seamless compatibility, among various applications and networks.

CHAPTER FIVE: RECOMMENDATION AND CONCLUSION

In light of the comprehensive evaluation conducted on various Decentralized Applications and APIs aimed at rectifying the security and interoperability challenges within digital identity systems, several key recommendations emerge. Firstly, it is imperative for developers and stakeholders to prioritize collaborative efforts in refining existing DaPPs and APIs, integrating more robust encryption protocols and standardized interoperable frameworks. Additionally, a concerted focus on user-centric design principles will enhance usability and trust among end-users. Future research and development endeavors should gravitate toward exploring novel consensus mechanisms and encryption methodologies to fortify the security layers embedded within these solutions. Moreover, fostering industry-wide standards and protocols will facilitate seamless interoperability among diverse digital identity platforms, ensuring a cohesive ecosystem. These recommendations collectively aim to propel the evolution of DaPPs and APIs, fostering a more secure, interconnected, and user-friendly digital identity landscape.

In conclusion, this study underscores the pivotal role of Decentralized Applications and APIs in mitigating the persistent security and interoperability challenges within digital identity systems. Through a systematic evaluation utilizing results-based monitoring and evaluation techniques, it becomes evident that while existing solutions exhibit promising advancements, there remains ample room for improvement. The analyzed DaPPs and APIs showcase commendable strides in enhancing security measures and fostering interoperability, yet their efficacy varies significantly. Nonetheless, the findings highlight a fundamental shift towards decentralized frameworks as a viable avenue for addressing intricate digital identity issues. As we navigate the ever-evolving landscape of digital identities, it is imperative to heed the recommendations proposed herein, fostering innovation, collaboration, and standardization to forge a more resilient, user-centric, and harmonized digital identity ecosystem.

REFERENCES

CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

Javaid a, a, b, c, d, & Abstract Financial service providers find blockchain technology useful to enhance authenticity. (2022, October 22). A review of Blockchain technology applications for financial services. Bench Council Transactions on Benchmarks, Standards and Evaluations. https://www.sciencedirect.com/science/article/pii/S2772485922000606

Faber, B., Michelet, G., Weidmann, N., Mukkamala, R. R., & Vatrapu, R. (2019). BPDIMS: A blockchain-based personal data and identity management system. In The 52nd Hawaii International Conference on System Sciences. HISS 2019: HISS 2019 (pp. 6855-6864). Hawaii International Conference on System Sciences (HICSS).

Liu, Y., He, D., Obaidat, M. S., Kumar, N., Khan, M. K., & Choo, K. K. R. (2020). Blockchain-based identity management systems: A review. Journal of network and computer applications, 166, 102731.

W. Zou et al., "Smart Contract Development: Challenges and Opportunities," in IEEE Transactions on Software Engineering, vol. 47, no. 10, pp. 2084-2106, 1 Oct. 2021, doi: 10.1109/TSE.2019.2942301.

Szabo N (1994) Smart contracts. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/

S. Nakamoto, et al., Bitcoin: A peer-to-peer electronic cash system, 2008, https: //git.dhimmel.com/bitcoin-whitepaper/.

Szabo, N. (1996). Smart Contracts: Building Blocks for Digital Markets http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOT winterschool2006/szabo.best.vwh.net/smart_contracts_2.html [https://perma.cc/TQ5SNUWL].

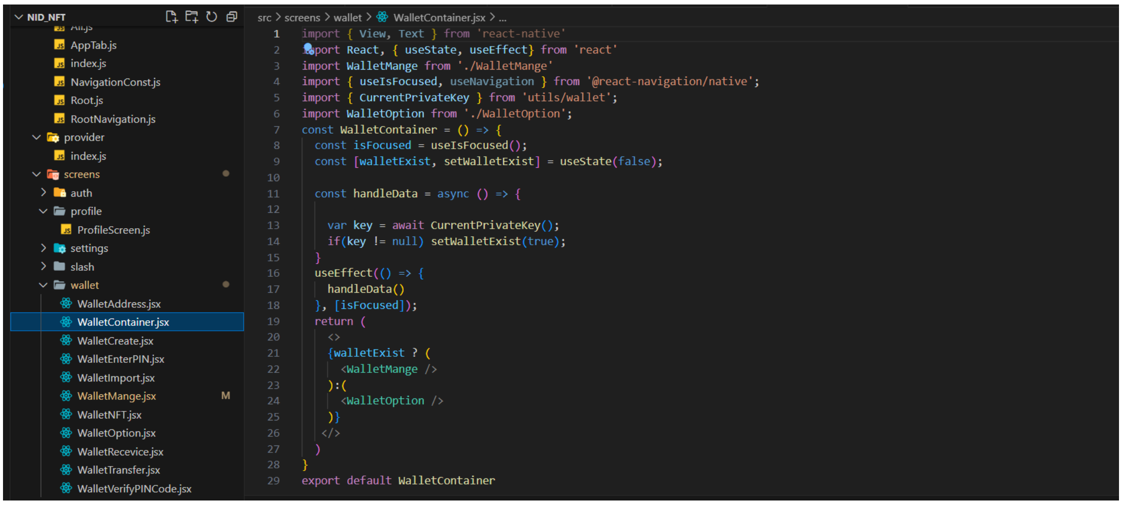
Osborn, G., & Alan, N. (2023). Web3 Disruption and the Domain Name System: Understanding the Trends of Blockchain Domain Names and the Policy Implications. Available at SSRN 4498160.

John Wiley & Sons, Ltd. (2006, April 4). (PDF) how do apis evolve? A story of refactoring (2006): Danny Dig: 254 citations. SciSpace - Paper. https://typeset.io/papers/how-do-apis-evolve-a-story-of-refactoring-1w18hp7rih

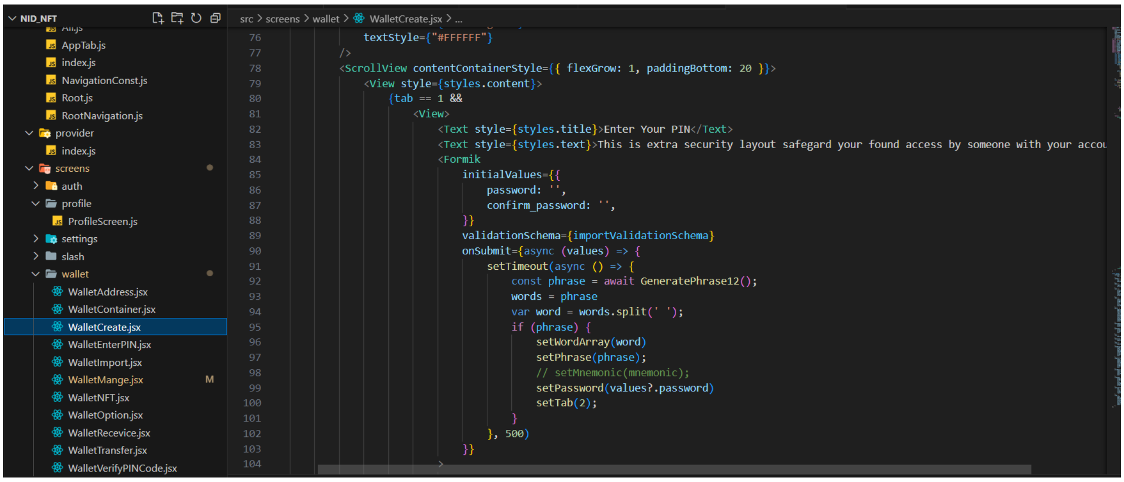
Liu, B., Si, X., &amp; Kang, H. (2022, November 16). A literature review of blockchain-based applications in Supply Chain. MDPI. https://www.mdpi.com/2071-1050/14/22/15210

APPENDIX

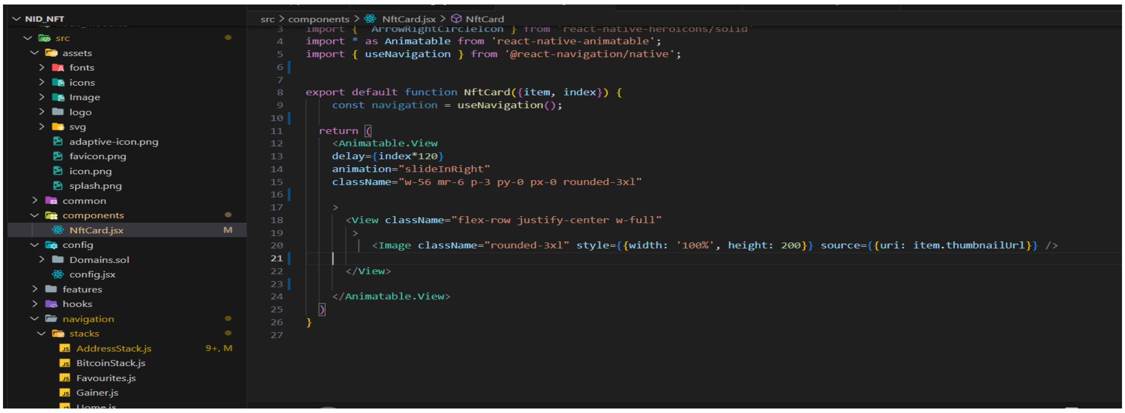
Appendix 1: DaPP source code



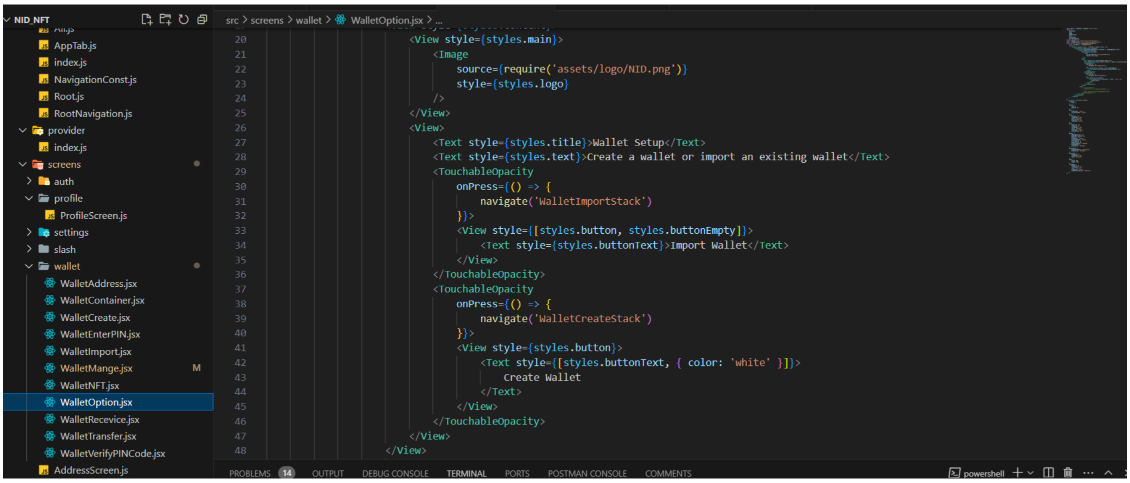
```
import { View, Text } from 'react-native'
import React, { useState, useEffect} from 'react'
import WalletMange from './WalletMange'
import { useIsFocused, useNavigation } from '@react-navigation/native';
import { CurrentPrivateKey } from 'utils/wallet';
import WalletOption from './WalletOption';
const WalletContainer = () => {
  const isFocused = useIsFocused();
  const [walletExist, setWalletExist] = useState(false);

  const handleData = async () => {

    var key = await CurrentPrivateKey();
    if(key != null) setWalletExist(true);
  }
  useEffect(() => {
    handleData()
  }, [isFocused]);
  return (
    <>
    {walletExist ? (
      <WalletMange />
    ):(
      <WalletOption />
    )}
   </>
  )
}
export default WalletContainer
```



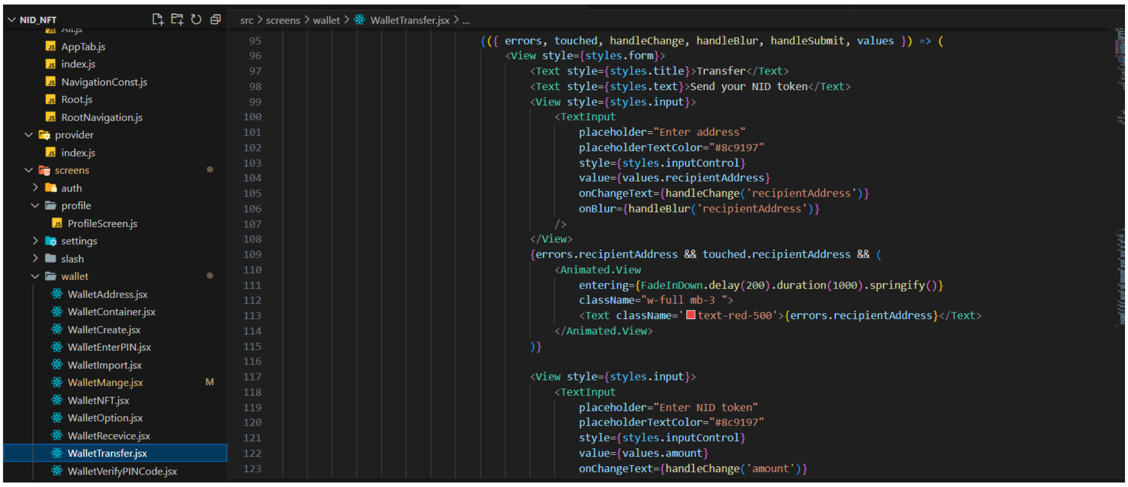
```
            textStyle={"#FFFFFF"}
          />
          <ScrollView contentContainerStyle={{ flexGrow: 1, paddingBottom: 20 }}>
              <View style={styles.content}>
                  {tab == 1 &&
                      <View>
                          <Text style={styles.title}>Enter Your PIN</Text>
                          <Text style={styles.text}>This is extra security layout safegard your found access by someone with your accou
                          <Formik
                              initialValues={{
                                  password: '',
                                  confirm_password: '',
                              }}
                              validationSchema={importValidationSchema}
                              onSubmit={async (values) => {
                                  setTimeout(async () => {
                                      const phrase = await GeneratePhrase12();
                                      words = phrase
                                      var word = words.split(' ');
                                      if (phrase) {
                                          setWordArray(word)
                                          setPhrase(phrase);
                                          // setMnemonic(mnemonic);
                                          setPassword(values?.password)
                                          setTab(2);
                                      }
                                  }, 500)
                              }}
                          >
```



```
import {  ArrowRightCircleIcon } from 'react-native-heroicons/solid'
import * as Animatable from 'react-native-animatable';
import { useNavigation } from '@react-navigation/native';


export default function NftCard({item, index}) {
    const navigation = useNavigation();

  return (
    <Animatable.View
    delay={index*120}
    animation="slideInRight"
    className="w-56 mr-6 p-3 py-0 px-0 rounded-3xl"

    >
      <View className="flex-row justify-center w-full"
       >
        <Image className="rounded-3xl" style={{width: '100%', height: 200}} source={{uri: item.thumbnailUrl}} />

      </View>

    </Animatable.View>
  )
}

```

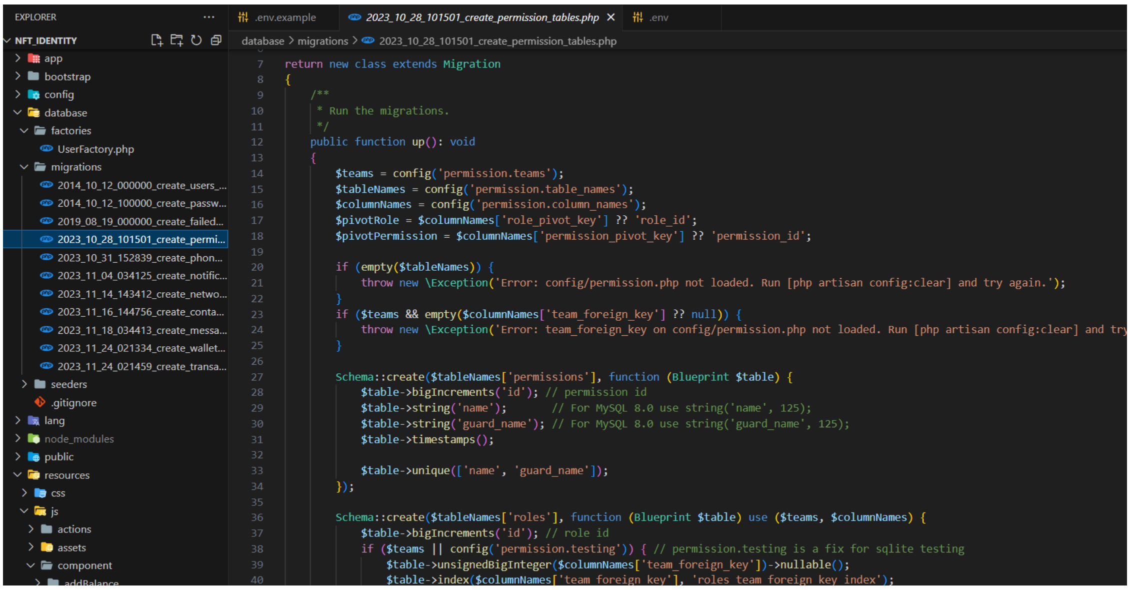


```
        <View style={styles.main}>
            <Image
                source={require('assets/logo/NID.png')}
                style={styles.logo}
            />
        </View>
        <View>
            <Text style={styles.title}>Wallet Setup</Text>
            <Text style={styles.text}>Create a wallet or import an existing wallet</Text>
            <TouchableOpacity
                onPress={() => {
                    navigate('WalletImportStack')
                }}>
                <View style={[styles.button, styles.buttonEmpty]}>
                    <Text style={styles.buttonText}>Import Wallet</Text>
                </View>
            </TouchableOpacity>
            <TouchableOpacity
                onPress={() => {
                    navigate('WalletCreateStack')
                }}>
                <View style={styles.button}>
                    <Text style={[styles.buttonText, { color: 'white' }]}>
                        Create Wallet
                    </Text>
                </View>
            </TouchableOpacity>
        </View>
    </View>
```



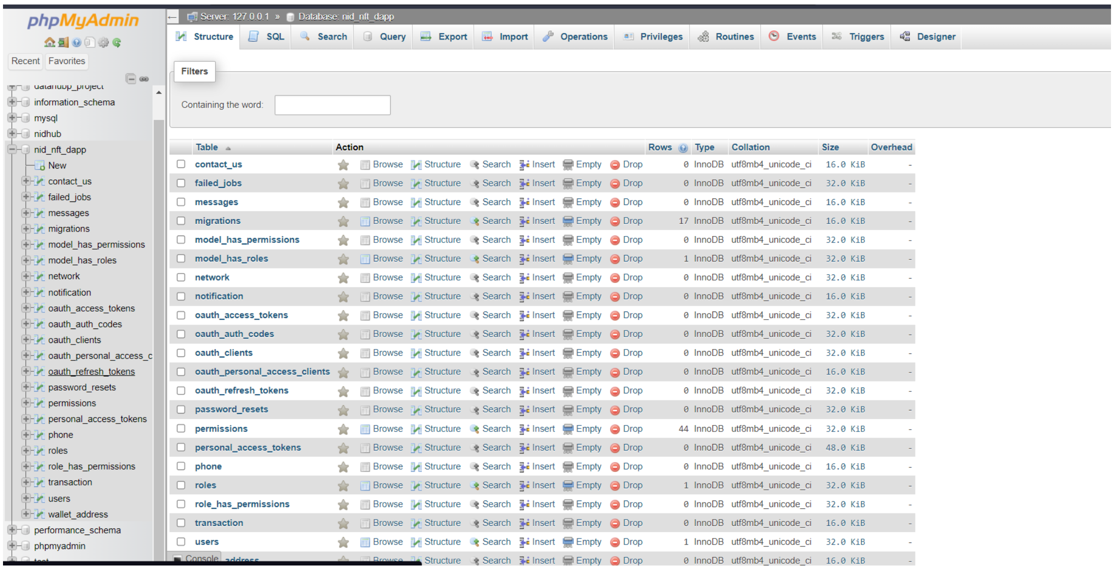
```
{({ errors, touched, handleChange, handleBlur, handleSubmit, values }) => (
    <View style={styles.form}>
        <Text style={styles.title}>Transfer</Text>
        <Text style={styles.text}>Send your NID token</Text>
        <View style={styles.input}>
            <TextInput
                placeholder="Enter address"
                placeholderTextColor="#8c9197"
                style={styles.inputControl}
                value={values.recipientAddress}
                onChangeText={handleChange('recipientAddress')}
                onBlur={handleBlur('recipientAddress')}
            />
        </View>
        {errors.recipientAddress && touched.recipientAddress && (
            <Animated.View
                entering={FadeInDown.delay(200).duration(1000).springify()}
                className="w-full mb-3 ">
                <Text className='🟥text-red-500'>{errors.recipientAddress}</Text>
            </Animated.View>
        )}

        <View style={styles.input}>
            <TextInput
                placeholder="Enter NID token"
                placeholderTextColor="#8c9197"
                style={styles.inputControl}
                value={values.amount}
                onChangeText={handleChange('amount')}
```

Backend Source code:


EXPLORER
NFT_IDENTITY
app
bootstrap
config
database
factories
UserFactory.php
migrations
2014_10_12_000000_create_users_...
2014_10_12_100000_create_passw...
2019_08_19_000000_create_failed...
2023_10_28_101501_create_permi...
2023_10_31_152839_create_phon...
2023_11_04_034125_create_notific...
2023_11_14_143412_create_netwo...
2023_11_16_144756_create_conta...
2023_11_18_034413_create_messa...
2023_11_24_021334_create_wallet...
2023_11_24_021459_create_transa...
seeders
.gitignore
lang
node_modules
public
resources
css
js
actions
assets
component
addBalance
.env.example
2023_10_28_101501_create_permission_tables.php
.env
database > migrations > 2023_10_28_101501_create_permission_tables.php

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        $teams = config('permission.teams');
        $tableNames = config('permission.table_names');
        $columnNames = config('permission.column_names');
        $pivotRole = $columnNames['role_pivot_key'] ?? 'role_id';
        $pivotPermission = $columnNames['permission_pivot_key'] ?? 'permission_id';

        if (empty($tableNames)) {
            throw new \Exception('Error: config/permission.php not loaded. Run [php artisan config:clear] and try again.');
        }
        if ($teams && empty($columnNames['team_foreign_key'] ?? null)) {
            throw new \Exception('Error: team_foreign_key on config/permission.php not loaded. Run [php artisan config:clear] and try
        }

        Schema::create($tableNames['permissions'], function (Blueprint $table) {
            $table->bigIncrements('id'); // permission id
            $table->string('name');       // For MySQL 8.0 use string('name', 125);
            $table->string('guard_name'); // For MySQL 8.0 use string('guard_name', 125);
            $table->timestamps();

            $table->unique(['name', 'guard_name']);
        });

        Schema::create($tableNames['roles'], function (Blueprint $table) use ($teams, $columnNames) {
            $table->bigIncrements('id'); // role id
            if ($teams || config('permission.testing')) { // permission.testing is a fix for sqlite testing
                $table->unsignedBigInteger($columnNames['team_foreign_key'])->nullable();
                $table->index($columnNames['team_foreign_key'], 'roles_team_foreign_key_index');
```

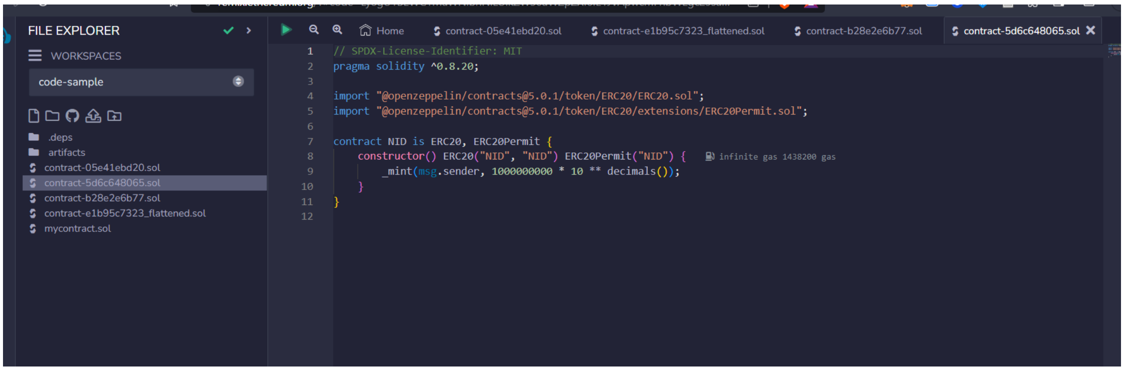
Local Database



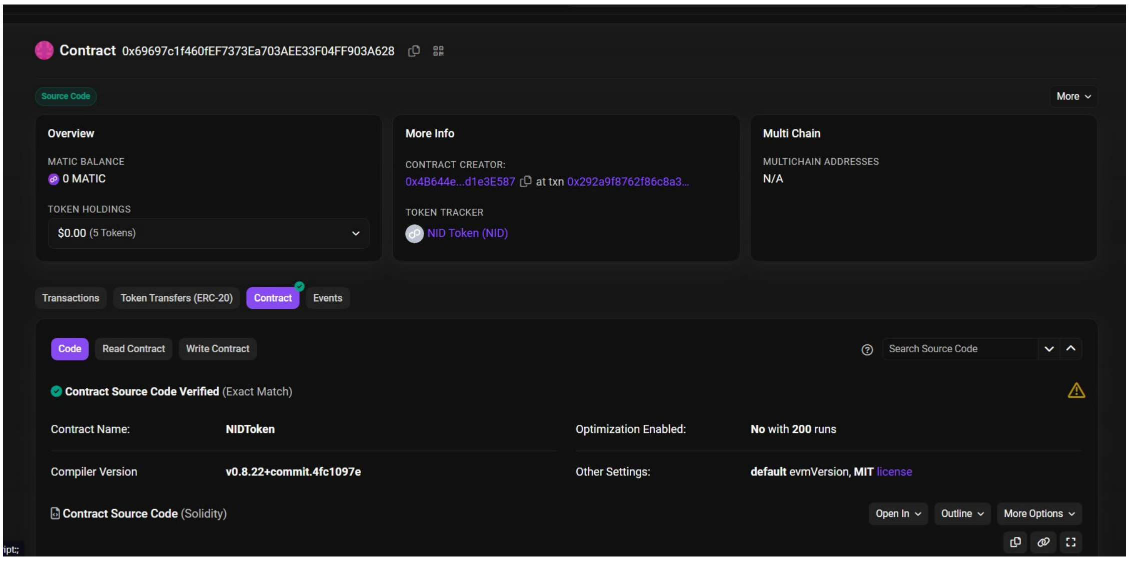
Containing the word:

	Table	Action							Rows	Type	Collation	Size	Overhead
□	contact_us	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
□	failed_jobs	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	messages	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
□	migrations	★	Browse	Structure	Search	Insert	Empty	Drop	17	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
□	model_has_permissions	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	model_has_roles	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	network	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	notification	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
□	oauth_access_tokens	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	oauth_auth_codes	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	oauth_clients	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	oauth_personal_access_clients	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
□	oauth_refresh_tokens	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	password_resets	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	permissions	★	Browse	Structure	Search	Insert	Empty	Drop	44	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	personal_access_tokens	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
□	phone	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
□	roles	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	role_has_permissions	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
□	transaction	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
□	users	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
	address	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-

Appendix 2: Smart contract



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

import "@openzeppelin/contracts@5.0.1/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts@5.0.1/token/ERC20/extensions/ERC20Permit.sol";

contract NID is ERC20, ERC20Permit {
    constructor() ERC20("NID", "NID") ERC20Permit("NID") {
        _mint(msg.sender, 1000000000 * 10 ** decimals());
    }
}
```



Contract 0x69697c1f460fEF7373Ea703AEE33F04FF903A628

Overview

MATIC BALANCE
0 MATIC

TOKEN HOLDINGS
$0.00 (5 Tokens)

More Info

CONTRACT CREATOR:
0x4B644e...d1e3E587 at txn 0x292a9f8762f86c8a3...

TOKEN TRACKER
NID Token (NID)

Multi Chain

MULTICHAIN ADDRESSES
N/A

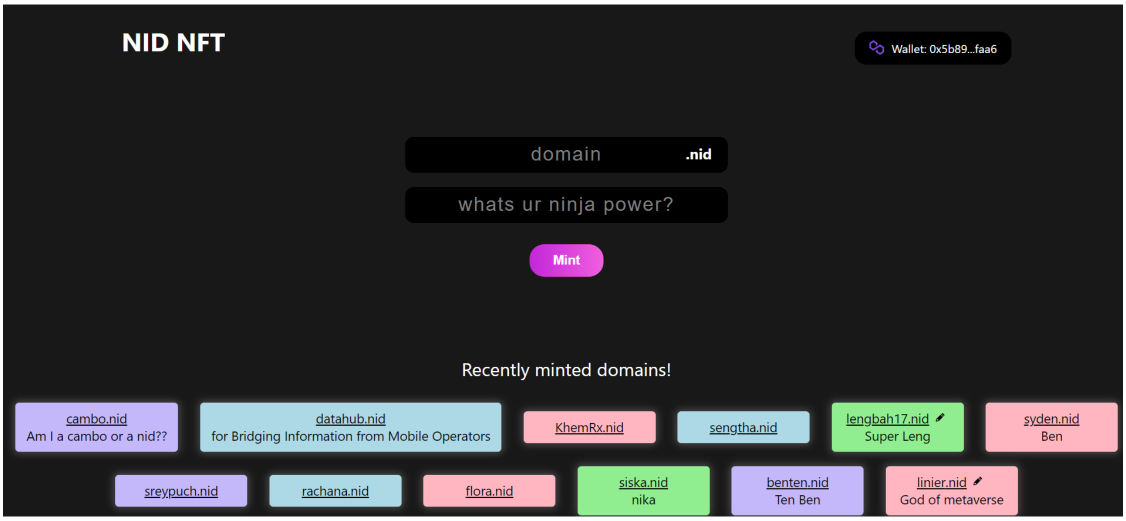
Contract Source Code Verified (Exact Match)

Contract Name:	NIDToken	Optimization Enabled:	No with 200 runs
Compiler Version	v0.8.22+commit.4fc1097e	Other Settings:	default evmVersion, MIT license

Contract Source Code (Solidity)

Appendix 3: Minting NFT

Mint NFT



Recently minted domains!

cambo.nid
Am I a cambo or a nid??

datahub.nid
for Bridging Information from Mobile Operators

KhemRx.nid

sengtha.nid

lengbah17.nid ✎
Super Leng

syden.nid
Ben

sreypuch.nid

rachana.nid

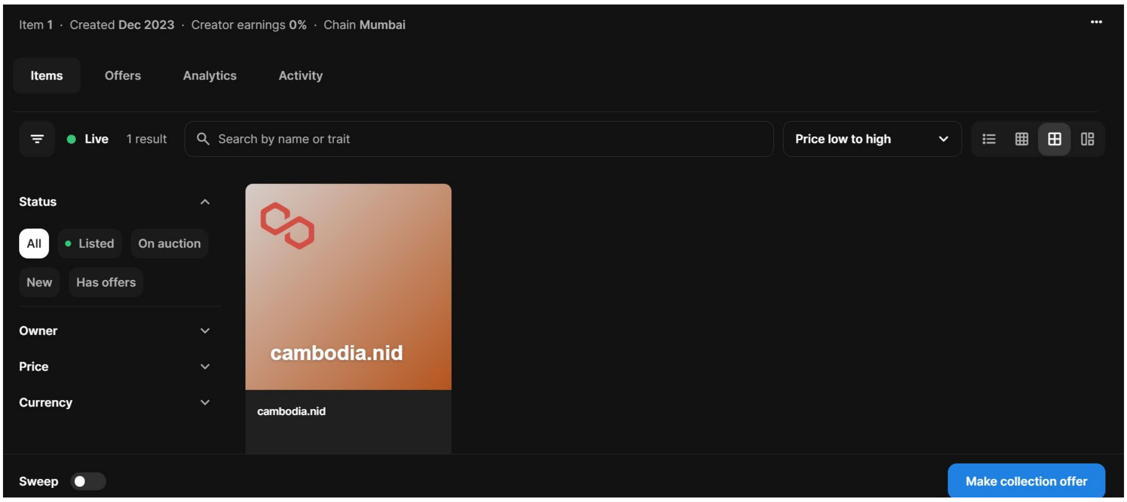
flora.nid

siska.nid
nika

benten.nid
Ten Ben

linier.nid ✎
God of metaverse

Minted NID NFT in Opensea


Item 1 · Created Dec 2023 · Creator earnings 0% · Chain Mumbai