Report On

Decentralized Portfolio

Submitted in partial fulfillment of the requirements of the Course project in Semester VII of Fourth Year Computer Engineering

By
Shubham Nakashe(Roll No. 63)
Sanika Patil (Roll No. 64)
Rishabh Tripathi (Roll No. 68)

Supervisor
Dr. Dinesh Patil

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled "Decentralized Portfolio" is a bonafide work of "Shubham Nakashe (Roll No. 63), Sanika Patil (Roll No. 64), Rishabh Tripathi (Roll No. 68) submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester VII of Fourth Year Computer Engineering.

.

Supervisor

Dr. Dinesh Patil

Dr. Megha Trivedi
Head of Department

Abstract

Personal portfolios are important tools for demonstrating one’s skills, experience, and achievements to potential employers, clients, or collaborators. However, traditional portfolios are often centralized, vulnerable to tampering, and dependent on third-party platforms. This project proposes a novel solution: Decentralize Portfolio, a DApp that leverages the Ethereum blockchain to enable users to create and manage their own digital portfolios in a secure, transparent, and censorship-resistant way. The project uses smart contracts to store and verify portfolio data, IPFS to store portfolio assets, and web3.js to interact with the blockchain. The project also implements a rating system that allows users to receive feedback and endorsements from other users. The project demonstrates the feasibility and benefits of using blockchain technology for personal portfolio management and showcases the potential of DApps for empowering individuals in the digital age.

Contents **Page No.**

1 Introduction

1.1 Introduction

In an era defined by digitalization, the need for secure and transparent tools to showcase one's skills, experience, and achievements has become increasingly essential. The Decentralized Portfolio is a revolutionary decentralized application (DApp) that harnesses the power of blockchain technology to provide individuals with a cutting-edge platform for managing and sharing their personal portfolios.

Built on the Ethereum blockchain, the Decentralized Portfolio leverages the inherent security and immutability of blockchain to empower users in their pursuit of personal and professional growth. This platform allows individuals to take control of their digital identity by crafting comprehensive and censorship-resistant portfolios that highlight their unique skills, experiences, and accomplishments.

Key features of the Decentralized Portfolio DApp include user-friendly portfolio creation tools, advanced authentication mechanisms, and the ability to verify and timestamp entries for authenticity. By utilizing the decentralized nature of the Ethereum blockchain, users can rest assured that their portfolios are resistant to tampering and offer a level of transparency and trust that traditional portfolio platforms cannot match.

As we delve deeper into the world of decentralized applications and blockchain technology, the Decentralized Portfolio stands as an exemplar of innovation, empowerment, and self-sovereignty in the digital age.

1.2 Problem Statement

The problem statement revolves around the limitations of current portfolio management systems in the rapidly evolving digital asset landscape. These systems need more transparency, making it easier for investors to verify asset authenticity. Security and custody concerns arise due to the reliance on third-party custodians, introducing theft and fraud risks. Managing diverse asset classes, like cryptocurrencies, stocks, and real estate, across various platforms is complex. Regulatory compliance is cumbersome and varies across jurisdictions, hindering investors' efforts to ensure they are compliant. Accessibility to portfolio management tools is restricted to institutional investors, excluding a broader audience.

Interoperability issues between traditional and blockchain platforms hinder seamless portfolio management. Data integrity is compromised within centralized systems due to data manipulation risks. High fees and administrative costs in traditional systems reduce investor returns. As the digital asset landscape continues to grow, scalability is essential. The project aims to develop a blockchain-based decentralized portfolio management system to address these issues, offering transparency, security, accessibility, and cost-efficiency. It empowers a diverse range of investors, ensuring regulatory compliance, data integrity, and scalability in portfolio management.

1.3 Project Scope

The scope of the Decentralized Portfolio project extends to encompass a comprehensive ecosystem that facilitates the creation, management, and sharing of personal portfolios using the Ethereum blockchain. This includes the development of a user-friendly DApp interface with intuitive portfolio creation tools, smart contract integration for ensuring the authenticity and immutability of portfolio entries, and robust user authentication mechanisms. The project also encompasses the implementation of a decentralized storage solution to securely host portfolio data. Furthermore, it involves the development of a verification system to validate and timestamp portfolio entries, enhancing their credibility. In addition, the project will explore potential integrations with other decentralized applications and services, aiming to create a vibrant and interconnected network for personal and professional growth. Ultimately, the Decentralized Portfolio seeks to empower individuals to control their digital identities and provide a trusted, censorship-resistant platform for showcasing their skills, experiences, and achievements.

2. Requirement Analysis

2.1 Hardware Requirements

Hardware Required:

Recommended:

- 16 GB RAM

Minimum:

- 8 GB RAM

2.2 Software Requirements:

- Visual Studio Code
- Pinata Cloud
- React Js & Node JS
- Metamask Wallet
- Solidity

2.3 Functional Requirements

User Registration and Authentication:

- Users should be able to create accounts securely using a username and password, or with a cryptocurrency wallet address.
- Implement multi-factor authentication for enhanced security.
- Users must be able to log in and log out of their accounts.

Portfolio Creation:

- Users should be able to create and customize their digital portfolios.
- Include fields for personal details, work experience, skills, and educational background.
- Allow users to upload images, documents, and links to showcase their work.

Blockchain Integration:

- Utilize Ethereum smart contracts to store and manage portfolio data.
- Enable users to timestamp and verify portfolio entries on the blockchain for authenticity.

Portfolio Management:

- Users should have the ability to edit, update, and delete portfolio entries.
- Implement version control for portfolio changes to track the evolution of a user's profile.

2.4 Nonfunctional Requirements

Performance: The DApp should be able to handle a large number of transactions and users simultaneously without any significant degradation in response time.

Security: Given the sensitive nature of the data, the DApp should ensure that all data is encrypted and secure from unauthorized access. It should also be resistant to common web vulnerabilities.

Reliability: The DApp should be available for use at all times, with minimal downtime for maintenance or updates.

Usability: The user interface should be intuitive and easy to use, even for individuals who are not familiar with blockchain technology.

Scalability: The DApp should be able to scale and accommodate an increasing number of users and transactions over time.

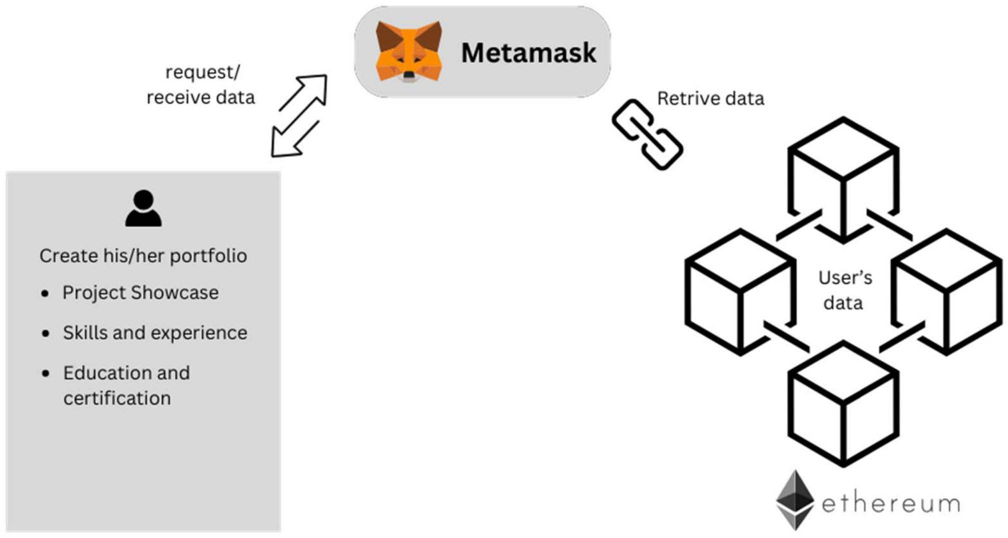
Maintainability: The codebase should be well-documented and modular to allow for easy updates and bug fixes.

3. System Design

3.1 System Design

The Decentralized Portfolio works by allowing users to create and showcase their skills and achievements on a secure blockchain. Users can add details about their projects, skills, education, and experience. This information is stored on the blockchain, making it tamper-proof and transparent. Users can also use a tool like Metamask to interact with the system and manage their digital portfolios securely.

3.2 Diagram



3.2 Module Description

The Decentralized Portfolio website is a decentralized application (DApp) developed on the Ethereum blockchain, designed to empower users in creating, managing, and showcasing their digital portfolios. This module plays a crucial role in facilitating the core features of the Decentralized Portfolio DApp, ensuring a secure, transparent, and censorship-resistant platform for individuals to exhibit their professional accomplishments.

1) Change description:

This module uses SMART CONTRACT which is deployed in user remix ide to collect the user's portfolio description and then publish it to our portfolio website. This includes the “description” that is displayed on the front page.

2) ChangeExperience/ InsertExperience:

This module of the smart contract is used to actively add, edit, and update details of user experience history being displayed on the portfolio. This contains the “date”, “post”, “skills learned”, and “company”

3) ChangeProject/ InsertProject:

This function of the smart contract is used to actively update the project details of the user that is being portrayed on the portfolio. This contains the “project name”, “skills learned”, and “date”.

4) ChangeEducation/ InsertEducation:

Incorporating a section for users to input details about their educational background. This module ensures that users can emphasize their academic and professional qualifications, enhancing the credibility of their portfolios. This includes “year”, “course”, “college” and “knowledge gained”.

4. Implementation

4.1 Methodology

The methodology for developing the Decentralized Portfolio project involves a structured and iterative approach that encompasses the following stages:

Project Inception and Planning:

Define the project scope, objectives, and success criteria.

Identify key stakeholders and establish communication channels.

Create a detailed project plan outlining timelines, resource allocation, and milestones.

Requirement Analysis:

- Conduct a thorough analysis of the functional and non-functional requirements, ensuring a clear understanding of user needs.
- Develop detailed use cases and user stories to describe how users will interact with the DApp.
-

Blockchain Technology Selection:

- Select Ethereum as the blockchain platform for building the DApp.
- Determine the specific Ethereum network (mainnet, testnet, or a private network) for development and testing.

Architecture and Design:

- Design the system architecture, including front-end and back-end components.
- Create a database schema for portfolio data storage on the Ethereum blockchain using smart contracts.
- Plan the integration of Metamask for wallet management.

Development and Implementation:

- Implement the front end of the DApp using React for the user interface.
- Develop the smart contracts using Solidity for portfolio data management.
- Integrate Ethereum nodes to interact with the blockchain.

- Implement user authentication and authorization mechanisms.
- Develop a decentralized storage solution (e.g., IPFS) for hosting portfolio media files.

Metamask Integration:

- Integrate Metamask into the DApp for wallet management and transaction signing.
- Test the Metamask integration across multiple browsers and platforms.

Quality Assurance and Testing:

- Conduct comprehensive testing, including unit testing, integration testing, security testing, and user acceptance testing.
- Address and rectify any identified issues and bugs.

4.2 Sample Modules

Portfolio Creation and Management:

This module allows users to create and manage their digital portfolios. Users can input details about their projects, skills, education, and experience. They can add, edit, or remove entries, ensuring their portfolios are up-to-date.

Project Showcase:

Within the portfolio, users can showcase their projects. This module enables users to provide detailed descriptions, upload images or media files, add links to project repositories or websites, and include relevant information to highlight their work.

Skills and Experience Highlight:

Users can emphasize their skills and experience within their portfolios. This module allows users to list their skills, rate their proficiency, and add descriptions. Additionally, users can detail their work experience, including job roles, companies, and descriptions of their roles and accomplishments.

Education and Certification Entries:

This module permits users to include details about their educational background and certifications. Users can add information about their academic history, including institutions, degrees, and graduation years. Furthermore, they can list certifications they've obtained, along with their descriptions and issue dates.

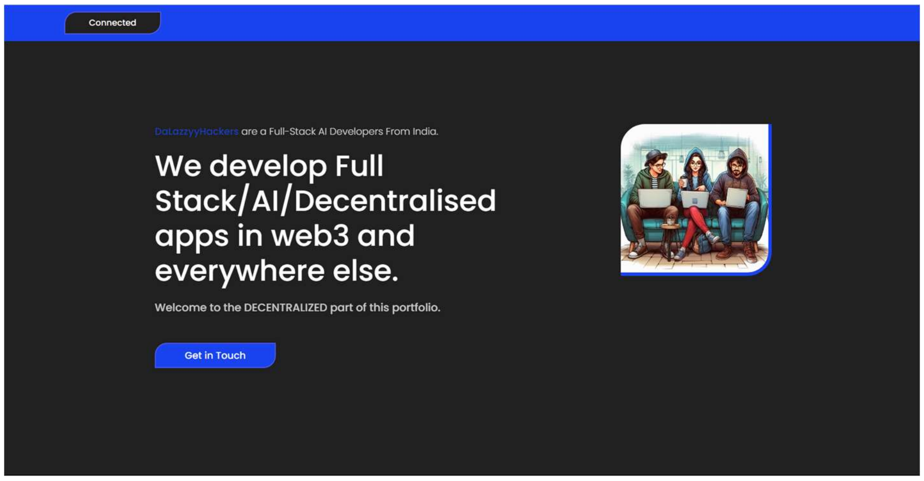
4.3 Code

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

contract Portfolio{

  struct Project{
      uint id;
      string name;
      string description;
      string image;
      string githubLink;
  }

  struct Education{
      uint id;
      string date;
      string degree;
      string knowledgeAcquired;
      string instutionName;
  }

  struct Experience{
      uint id;
      string date;
      string post;
      string knowledgeAcquired;
      string compamyName;
  }
  Project[3] public projects;
  Education[3] public educationDetails;
  Experience[3] public experienceDetails;

  string                                                                    public
imageLink="QmZVJjgQ2h379kGqKBqBs8PTozy5MtfWYe9KsGKdJTrgRY";
  string public description="";
  string                                                                    public
resumeLink="QmPVxpUgNwAXDuWnnQYRz6V2Rfv6WHLfdaHPCCJTCWfo
aY";
  uint projectCount;
  uint educationCount;
  uint experienceCount;
  address public manager;

  constructor(){
      manager=msg.sender;
  }

  modifier onlyManager(){
      require(manager==msg.sender,"You are not the manager");
      _;
  }

  //Project
  function insertProject(string calldata _name,string calldata _description,string
calldata _image,string calldata _githubLink) external{
      require(projectCount<3,"Only 3 projects allowed");
      projects[projectCount]                                                          =
Project(projectCount,_name,_description,_image,_githubLink);
      projectCount++;
  }

  function changeProject(string calldata _name,string calldata _description,string
calldata _image,string calldata _githubLink,uint _projectCount) external{
      require(_projectCount>=0 && _projectCount<3,"Only 3 projects allowed");_
     projects[projectCount] =
Project(_projectCount,_name,_description,_image,_githubLink);
  }
    function allProjects() external view returns(Project[3] memory){
       return projects;
    }

//Education
   function insertEducation(string calldata _date,string calldata _degree,string
calldata _knowledgeAcquired,string calldata _instutionName) external
onlyManager{
     require(educationCount<3,"Only 3 education details allowed");

educationDetails[educationCount]=Education(educationCount,_date,_degree,_kn
owledgeAcquired,_instutionName);
     educationCount++;
    }

  function changeEducation(string calldata _date,string calldata _degree,string
calldata _knowledgeAcquired,string calldata _instutionName,uint
_educationDetailCount) external onlyManager{
     require(_educationDetailCount>=0 && _educationDetailCount<3,"Invalid
educationCount");

educationDetails[_educationDetailCount]=Education(_educationDetailCount,_dat
e,_degree,_knowledgeAcquired,_instutionName);
  }

    function allEducationDetails() external view returns(Education[3] memory){
      return educationDetails;
  }
//Experience

    function insertExperience(string calldata _date,string calldata _post,string
calldata _knowledgeAcquired,string calldata _companyName) external
onlyManager{
     require(experienceCount<3,"Only 3 education details allowed");

experienceDetails[experienceCount]=Experience(experienceCount,_date,_post,_k
nowledgeAcquired,_companyName);
     experienceCount++;
    }

    function changeExperience(string calldata _date,string calldata _post,string
calldata _knowledgeAcquired,string calldata _companyName,uint
_experienceDetailCount) external  onlyManager{
     require(_experienceDetailCount>=0 && _experienceDetailCount<3,"Invalid
experienceCount");

experienceDetails[_experienceDetailCount]=Experience(_experienceDetailCount,
_date,_post,_knowledgeAcquired,_companyName);
    }

    function allExperienceDetails() external view returns(Experience[3] memory){
      return experienceDetails;
    }
  function changeDescription(string calldata _description) external{
      description=_description;
  }

    function changeResumeLink(string calldata _resumeLink) external
onlyManager{
      resumeLink=_resumeLink;
  }
     function changeImageLink(string calldata _imageLink) external onlyManager{
      imageLink=_imageLink;
  }

  function donate() public payable{
      payable(manager).transfer(msg.value);
  }
}
```

5. Results:

5.1 Results:

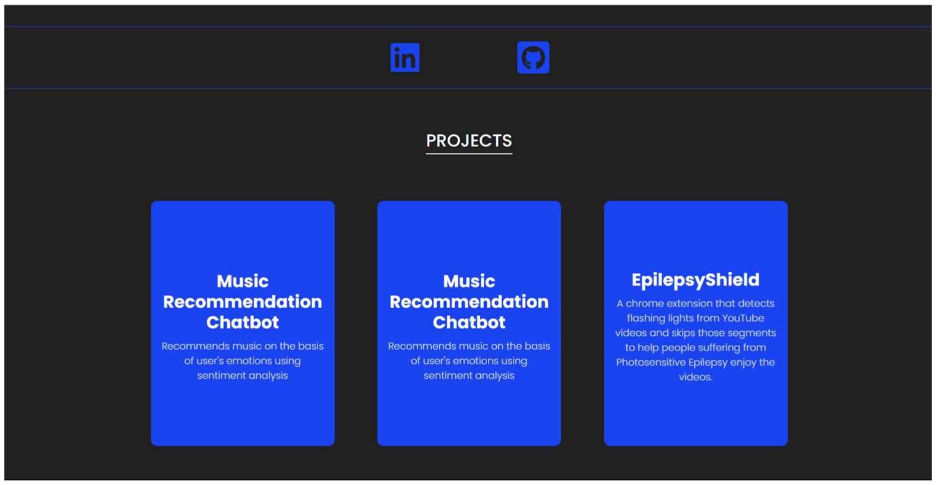
The development and implementation of the Decentralized Portfolio DApp on the Ethereum blockchain have resulted in successful user adoption of its key features, including Portfolio Creation, Project Showcase, Skills and Experience, Education and Certifications sections, and Decentralized Storage. These features collectively provide a secure and efficient platform for individuals to manage their professional portfolios while benefiting from the Ethereum blockchain decentralized storage, ensuring data integrity, immutability, and resistance to censorship. Additionally, the seamless integration of Metamask enhances user accessibility and interaction on both desktop and mobile devices, further solidifying the DApp's trustworthiness and usability.



DaLazzyyHackers are a Full-Stack AI Developers From India.

We develop Full Stack/AI/Decentralised apps in web3 and everywhere else.

Welcome to the DECENTRALIZED part of this portfolio.



Music Recommendation Chatbot

Recommends music on the basis of user's emotions using sentiment analysis

Music Recommendation Chatbot

Recommends music on the basis of user's emotions using sentiment analysis

EpilepsyShield

A chrome extension that detects flashing lights from YouTube videos and skips those segments to help people suffering from Photosensitive Epilepsy enjoy the videos.

5.2 Conclusion:

In conclusion, the Decentralized Portfolio DApp has successfully fulfilled its mission of creating a secure, transparent, and censorship-resistant platform for users to showcase their professional achievements. This DApp addresses a crucial need in the digital landscape by empowering users to efficiently manage and exhibit their portfolios while leveraging the Ethereum blockchain decentralized storage for data integrity and security. The seamless integration of Metamask has further enhanced user accessibility, emphasizing the DApp's dedication to user convenience and privacy. This project serves as a testament to the potential of blockchain technology in solving longstanding challenges and introduces a user-friendly, censorship-resistant solution for portfolio management. As the digital landscape continues to evolve, the DApp is poised to play a pivotal role in individuals' professional lives and the broader adoption of blockchain technology for practical, real-world applications.

References:

[1] R. A. Canessane, N. Srinivasan, A. Beuria, A. Singh and B. M. Kumar, "Decentralised Applications Using Ethereum Blockchain," 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2019, pp. 75-79, doi: 10.1109/ICONSTEM.2019.8918887.

[2] https://docs.metamask.io/

[3] Lee, Wei-Meng. (2019). Using the MetaMask Chrome Extension. 10.1007/978-1-4842-5086-0_5.

[4] Coutinho, Paulo & Tabak, Benjamin. (2001). Decentralized Portfolio Management. Revista Brasileira de Finanças. 1.

[5] Atzori, Marcella. (2017). Blockchain technology and decentralized governance: Is the state still necessary? Journal of Governance and Regulation. 6. 10.22495/jgr_v6_i1_p5.