Submission Information

Author Name	T.J. Lakshmi
Title	G63_1GA20CS149
Paper/Submission ID	1878436
Submitted by	kiran.p@gat.ac.in
Submission Date	2024-05-27 12:21:20
Total Pages, Total Words	30, 6279
Document type	Project Work

Result Information

Similarity **9 %**

1   10   20   30   40   50   60   70   80   90

Sources Type

- Student Paper 0.97%
- Journal/Publication 3.01%
- Internet 5.02%

Report Content

- Quotes 0.08%
- Words < 14, 6.43%
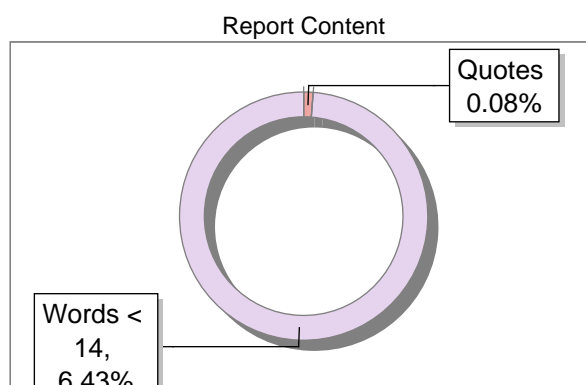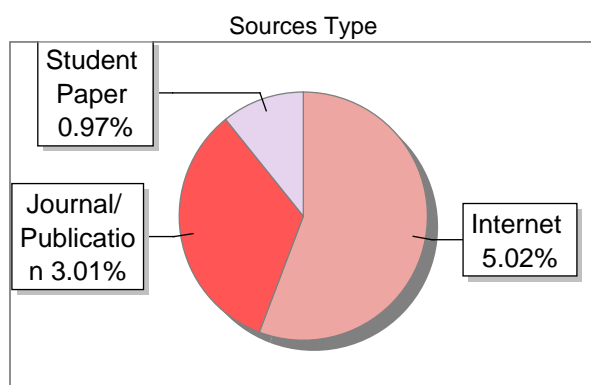
Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

15	www.ncbi.nlm.nih.gov	<1	Internet Data
16	auspaynetsummit.com.au	<1	Internet Data
17	gcelt.gov.in	<1	Internet Data
18	springeropen.com	<1	Internet Data
19	inba.info	<1	Internet Data
20	Progressive communication by -1982	<1	Publication
21	repository.arizona.edu	<1	Internet Data
22	www.dx.doi.org	<1	Publication
23	coek.info	<1	Internet Data
24	Collectivist and Individualist Influences on Website Design in South Korea and t by Heema-2009	<1	Publication
25	IEEE 2011 Information Security for South Africa (ISSA) - Johannesbur	<1	Publication
26	slate.com	<1	Internet Data
27	www.captainticket.com	<1	Internet Data
28	www.hireitpeople.com	<1	Internet Data
29	www.osha.gov	<1	Internet Data
30	americanaffairsjournal.org	<1	Internet Data
31	aprenderly.com	<1	Internet Data
32	A Privacy-preserving Mobile and Fog Computing Framework to Trace and Prevent COV by Whaiduzzaman-2020	<1	Publication
33	bibbase.org	<1	Internet Data

CHAPTER 1 INTRODUCTION 1.1 Introduction to Project The InnoVault project is a ground-breaking initiative at the interface of intellectual property protection and the technology of blockchain. In a world of rapid innovation and technological development, traditional intellectual asset safeguarding techniques have proven inefficient, frequently falling to centralization and security issues. In response, InnoVault leverages the irreversible and transparent features of blockchain technology, together with the decentralized framework of distributed applications (dApps), to create an innovative solution. InnoVault intends to empower creators and innovators by providing a secure, tamper-proof, and readily accessible platform that will transform how intellectual property is secured, shared, and monetized in the digital era. This project represents a key step in establishing a more fair and open innovation ecosystem, prepared to unleash the full potential of human creativity on a global scale. 1.2 Problem Definition The challenge at hand is the difficulties and challenges present with conventional intellectual property management techniques. The centralized systems can lack transparency, making inventions susceptible to tampering and unauthorized access. In addition, these systems rely mostly on middlemen, introducing inefficiencies and difficulties to the process. This fragmented ecosystem hinders development and restricts development throughout industries. As a result, a secure, transparent, and decentralized solution is needed to streamline intellectual property management, permitting seamless collaboration while respecting innovators' rights. The InnoVault project solves this important issue by combining blockchain technology and decentralized apps (dApps) to build an integrated platform that will completely change how intellectual property is maintained and shared in the digital age.

1.3 Existing System The existing system for intellectual property protection relies primarily on centrally located databases and authorities. Developers and creators file their ideas and creations with centralized platform for licensing and security. These platforms tend to suffer from various drawbacks. Poor Transparency: Centralized systems may be incomprehensible, making it difficult to verify the validity and possession of intellectual property. Tampering Risk: unauthorized individuals can tamper with or modify centralized databases, compromising the confidentiality of property data. Reliance on Middlemen: The method of establishing and handling the intellectual property may include mediators such as law firms or government agencies, resulting in delays, higher expenses, and administrative obstacles. Restricted Accessibility: Access to centralized intellectual property databases may be restricted, hindering collaboration as well as creativity. Administrative Enforcement: Understanding the regulatory and legal structures for safeguarding intellectual property can be challenging and time-consuming, especially in many nations. **1.4 Proposed System** The proposed InnoVault system offers a distributed approach to intellectual property management, integrating blockchain technology and decentralized apps (dApps) to get past the limits present in centralized solutions. The primary features of the proposed system include: Blockchain Infrastructure: Using a blockchain network to produce an irreversible and open ledger for registering and managing intellectual property rights. Smart Contracts: Using smart contracts to manage and uphold the rules and contracts that regulate the use of intellectual property ownership, authorization, and exchange. Distributed Storage: Using decentralized storage technologies such as IPFS to safely store intellectual property assets while preserving confidentiality and availability without depending on central servers for storage. InnoVault 2023-2024

Dept of CSE, GAT, Bengaluru-98 Page | 9 Peer-to-Peer collaboration: Facilitating peer-to-peer cooperation and transactions, enabling creators to interact directly with one another without the use of middlemen. 1.5 Objectives of the Project Work Implement User Authentication: Build an effective authentication system to restrict access to the platform, making sure only those with authorization may upload and see files. This could include using authentication methods. Secure File Uploads: Implement methods such as authenticated requests to verify your identity and accept file uploads. Validate file sizes and types to avoid unauthorised or harmful uploads, which enhances platform safety. Choose Storage alternatives: Evaluate and select appropriate on-chain and off-chain storage for data options. Options like IPFS enable decentralized and immutable storage, making them ideal for on-chain storage solutions. If required, traditional servers can be used to store data on the blockchain. Consider utilizing other databases for off-chain data, such as user information. Integrate IPFS or Conventional Servers: Integrate your choice of storage methods into the platform to guarantee seamless file uploads and access. For IPFS, this means using IPFS APIs or libraries to store and retrieve files. Safeguard user data through the installation of secure file storage methods on typical servers. Store Off-chain Data: Connect to other databases like MongoDB or Azure to securely store off-chain data, such as user information. Implement appropriate security measures, such as encryption and access control, to protect sensitive user data from unwanted access or breaches. 1.6 Scope of the Project Work The project scope comprises the development of the InnoVault platform, having a focus on user interfaces for authorization, file uploads, and access monitoring. It involves developing an effective user identification system to regulate platform access, as well as secure file upload abilities that verifies file sizes, types, and permissions. Storage solutions, including IPFS for on-chain storage and outside databases like MongoDB for off-chain data, must be assessed and integrated thoroughly. The project scope additionally involves smooth solution integration, thorough testing, InnoVault 2023-2024

document preparation, and deployment planning. Ongoing maintenance and support assure the platform's functionality, safety, and scalability after implementation. With these efforts, the project aims to provide an extensive way for creators to securely store and share their intellectual property assets. 1.7 Project Report Outline Chapter 1: Introduction Provides an overview of the project, highlighting its significance, objectives, and its scope, and providing the context for the subsequent chapters. Chapter 2: Literature Survey Evaluates existing research and literature concerning the project, giving perspectives on current trends, the internet, and methods in the field. Chapter 3: System needs and Specification Descriptions the system's functional and non-functional requirements, containing specific features, goals for performance, and limitations that must be dealt with. Chapter 4: System Design Describes the system's design, elements, and interactions, transforming requirements into a thorough implementation strategy. Chapter 5: Implementation This section discusses the true process of development that involves coding, component integration, and setting up the system in line with standards for design. Chapter 6: Testing Examines the system's functionality, efficiency, and reliability using various testing approaches to make sure it fulfils the defined requirements InnoVault 2023-2024

Dept of CSE, GAT, Bengaluru-98 Page | 11 Chapter 7: Results The results are given, emphasizing the results of the process of execution, such as efficiency advantages, security improvements, and any other important findings, and giving empirical proof to support the project's aims. Chapter 8: Conclusion The conclusion remarks on the results, highlighting the project's appreciate while offering insights into future implications and potential opportunities for blockchain study and development. InnoVault 2023-2024

CHAPTER 2 LITERATURE SURVEY 2.1 System Study The system study evaluates the features present in modern file management and storage systems, emphasizing safety, capacity, and privacy. This involves reviewing a variety of publications, including academic studies, literature surveys, and research papers, in order to gain insight into the current state of file storage solutions and what obstacles they face. The study will focus on data integrity challenges, vulnerabilities in security, and scalability restricts associated with centralized storage. Blockchain technology could help solve these challenges by offering a decentralized, immutable ledger for file data storage and guaranteeing data integrity that is resistant to modification. Will also look at how adding blockchain technology could impact user experience, system performance, and legal compliance while taking transaction productivity, delay, and cost-effectiveness into account. The system study will inform the design and execution of Innovault through this scientific evaluation, leading the project toward the creation of a reliable, private, and extensible file storage solution that utilizes blockchain technology. 2.2 Review of Literature The review of literature phase is a vital part of any research project, involving a thorough examination and integration of existing research, papers, and discussions in academia relevant to the topic's domain. This comprehensive review goes into a wide range of substances, including but not limited to academic works, books, journals, papers from conferences, and other online sites. Researchers attempt to clarify vital concepts, theories, methods, and empirical results through strict review of the literature, leading to a comprehensive understanding of the field's current state of knowledge, creating trends, and common best practices. This comprehensive synthesis not only puts the research try within the wider academic discourse, but it also helps to recognize gaps, discrepancies, and areas ripe for additional investigation or innovation.

Dept of CSE, GAT, Bengaluru-98 Page | 13 2.3 Comparison of Literature Table 2.1 Literature Survey Title of the paper Methodology Year Disadvantages Research on digital copyright protection based on the hyper ledger fabric blockchain network technology. A protection technique, automatic management of the complete digital rights life cycle on the blockchain using fabric's smart contract technology. September 2021 A lot of calculations for using public Blockchain. Blockchain Technology and Intellectual Property Rights Blockchain technology for various e-governance schemes initiated by governments. Vol.24(1-2) January-March 2019 Restrictions in terms of security, network size, speed, growth and transaction rates. Blockchainbased reliable image copyright protection Design having Blockchainbased reliable image copyright protection system named BBRICP using Hyperledger Fabric. March 2023 Adoption of Fabric mitigates the limitations of other blockchain platforms. A Comparative Study: Blockchain Technology Utilization Benefits, Challenges and Functionalities Aims to explore the benefits, challenges and functionalities that affect blockchain applications in different sectors. Volume 9, pages(12730 - 12749) 08 January 2021 Lack of empirical evidence and a systematic review of the current blockchain-enabled state-of-the-art. IP dLedger - Decentralized ledger for intellectual property administration IP decentralized ledger to significantly reduce IP administration costs and improve IP use. Volume 186, Part A, 2023 Many companies have not moved away from conventional centralized systems. Blockchain-based digital rights management systems: Design principles for the music industry To protect intellectual property by enabling transparent music licensing structures. April 2023 Design principles are high level as a result prescriptive power is limited InnoVault 2023-2024

Digital Art as 'Monetized Graphics': Enforcing Intellectual Property on the Blockchain To create proprietary digital art markets March 2018 Hinge on the dynamism of the digital as easily Copyable and replicable. A Comprehensive Survey on Blockchain- Based Decentralized Storage Networks Overview of blockchain- based storage systems and how they work, comparing with cloud- based storage networks and surveying of various decentralized storage networks like SIA, File coin, and Storj available on the market. Volume 11, pages(10995 - 11015), 27 January 2023 One of era's most crucial and controversial issues is the storage and retrieval of data in cloud storage. Solutions to Scalability of Blockchain: A Survey Covering the existing scaling solutions for blockchain and classify them by level. Volume 8, pages(16440 - 16455), 17 January 2020 Huge amount of blockchain data that need to be compressed, inefficient cross- shard transaction, unfinished protocols Blockchain Meets Metaverse and Digital Asset Management: A Comprehensive Survey How blockchain can enable the metaverse from different perspectives, ranging from user applications to virtual services and the blockchain-enabled economic system. Volume 11, pages(26258 - 26288), 14 March 20 Integration of blockchain into the metaverse is crucial to ensure the decentralization, security, and privacy of this virtual world. InnoVault 2023-2024

CHAPTER 3 SYSTEM REQUIREMENTS SPECIFICATION 3.1 Functional Requirements Functional requirements provide the required features and capabilities for a system to meet the needs of those who use it and achieve its objectives. These requirements usually specify the system's behaviour, efficiency, and interactions with users and other systems. The functional specifications for a project like InnoVault may include: User Authentication: To use the system, users must be able to securely register, log in, and handle their accounts. File Upload: The system ought to enable users to safely effectively upload any kind of files, like documents, photos, and videos. Access Control: Various user roles (such as administrators, contributors, and viewers) should have various degrees of access to the platform's files and features. File Management: Users should be able to organize, search, look around, alter, and delete uploaded files on the site. Blockchain Integration: Using blockchain technology to offer a transparent and permanent record of intellectual property rights and transaction. Integration with decentralized storage technologies such as IPFS ensures file storage is secure and durable. 3.2 Non-Functional Requirements Non-functional requirements, as compared to functional requirements, detect the characteristics or attributes that indicate how a system should behave. Performance: The system should be quick and able to handle a large number of simultaneous users and file uploads with a low delay. Scalability: The platform should be able to extend either vertically or horizontally to meet growing user and data requirements over time. Reliability: The system should be highly accessible and resilient to failure, with minimal downtime or disruptions to user access. Security: Encrypt sensitive information, use secure authentication methodologies, and protect user data.

Dept of CSE, GAT, Bengaluru-98 Page | 16 3.3 Hardware Requirements The hardware requirements for the project are as follows: 1. Server Infrastructure: Innovault needs server infrastructure with hardware parameters like CPU, RAM, and storage that can host and operate the program. 2. Network Connectivity: Stable internet access with enough bandwidth to facilitate file uploads, downloads, and blockchain transactions in addition to user interactions. 3. Backup and Redundancy: In order to lessen the chance of data loss or system failures, includes strategies such as redundant storage options, backup servers, and disaster recovery plans. 3.4 Software Requirements The software requirements encompass libraries and frameworks essential for the project's functionality: 1. Operating System: The server's operating system, such as Ubuntu or Linux distributions, allows it to host web applications. 2. Web Server: For Innovault to provide the application users online, a web server program such as Apache HTTP Server is needed. 3. Database Management system: User data, file metadata, and other application-specific data are stored in NoSQL databases like MongoDB. 4. Blockchain Platform: Innovault's fundamental functions require integration with a blockchain platform such as Dfinity (an internet computer) and IPFS (InterPlanetary File System) for decentralized storage. 5. Programming Languages and Frameworks: Programming languages like JavaScript, web development frameworks and libraries (such as Node.js, Express.js, and React.js), smart-contracts development language called Motoko is used to create the application. 6. Additional Software Dependencies: Authentication and encryption, and third-party APIs for incorporating other features or services into the Innovault application. These elements serve as the basis for the development of Innovault, assuring dependability, safety, and expandability. By following these guidelines, Innovault may satisfy performance requirements and meet with the regulations while providing consumers with a blockchain- powered, secure, and decentralized file storage service InnoVault 2023-2024

CHAPTER 4 4.1 Design Overview System Design Innovault focuses on implementing blockchain and smart contracts on the Internet Computer platform to result in a decentralized intellectual property (IP) management system. Motoko, the smart contract language, provides steady storage of ownership information on the blockchain. Every digital asset is provided with a unique identifier on the InterPlanetary File System (IPFS) so that the content is stored in a decentralized manner. With the help of efficient verification through smart contracts and robust access controls, Innovault can make IP management secure and transparent for users. 4.2 System Architecture The system architecture of Innovault consists of the basic and mostly used layers in every other application the only peculiar one in our case is the blockchain integration layer. Descriptions of every layer are given below. Fig 4.2.1 Architecture of Innovault

1. User Interface Layer: This is the layer where the user interacts with the entire application. It consists of options like User registration and User login to enter the application and make use of its functionalities. Innovault has a dashboard so that users can view their accounts, log out, and also search for digital assets. Users can upload their digital assets using the upload options shown in the dashboard of the Innovault. 2. Application Layer: This is the layer that is behind the entire business logic of Innovault. This layer performs the process of IP management by interacting with the user interface to get the digital assets and then the smart contracts are executed after a successful execution it gets the transaction details from the blockchain network and stores them in the database layer. Whenever the user wants to fetch data this application layer interacts with the database layer to fetch them. 3. Blockchain Integration Layer: This layer is the implementation of Internet Computer (IC) technology in Innovault. This enables the development and deployment of smart contracts and applications. The IC platform utilizes blockchain principles to achieve features like transparency, security, and decentralized storage. 4. Security Layer: This layer is responsible for implementing security principles and functionalities to the entire Innovault project. Hashing algorithms like SHA-256 are incorporated into the Internet Computer (IC) platform and related technologies, such as the InterPlanetary File System (IPFS). Bcrypt algorithm is used while storing non- blockchain related data like user details, passwords, and usernames into the database. To conclude, Innovault integrates a user-friendly interface for registration, login, and asset uploads. The Application Layer manages IP processes, executes smart contracts, and interacts with the database. Utilizing Internet Computer (IC) technology, the Blockchain Integration Layer ensures transparency and decentralized storage. A robust Security Layer employs SHA- 256 for blockchain data and Bcrypt for non-blockchain information, fortifying user credentials and maintaining data security. Innovault stands as a comprehensive and secure platform for intellectual property management. InnoVault 2023-2024

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 19 4.3 Data Flow Diagram Fig 4.3.1 Dataflow of Innovault The above diagram represents the flow of data in Innovault. The above process can be broken down into the following steps: 1. First the users register themselves into the application by providing their usernames, passwords, and other details. 2. All these user details are retrieved by the application layer and stored in the user profile database. 3. Then the users get access to the dashboard and can upload their digital assets during this process ownership is verified with the help of smart contract execution and after the contract is validated the data is stored on the blockchain network. 4. When a user wants to retrieve the digital assets of another user then a transaction process takes place. The transaction facilitates the validation of the user's request and ensures that the user has the necessary permissions to access the specified digital asset. 5. After the transaction is validated, the transaction history related to the download of digital assets is updated on the IC blockchain. The Internet Computer blockchain, being a decentralized and secure ledger, records and maintains the details of transactions, including downloads.

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 20 CHAPTER 5 IMPLEMENTATION 5.1 Steps for Implementation 1. Installation of Hardware and Software utilities: By launching PowerShell as an Administrator and setting up the required parameters to support the Windows Subsystem for Linux, WSL was enabled on a Windows computer. Ubuntu was installed via the Microsoft Store after WSL was configured, choosing the most recent LTS version for stability. After installation, Ubuntu was started from the Start menu, and the process of configuring a user account—which included setting up a username and password—was finished. Visual Studio Code's Remote - WSL plugin was installed in order to set up the Dfinity platform and Node.js. This addon makes it easier to develop in a Linux environment straight from Windows. Following the extension's installation, Visual Studio Code was launched, and the remote indication located in the bottom-left corner was used to create a connection to the WSL Ubuntu environment. Through this connection, the editor could communicate directly with the Ubuntu environment. Once the connection was made, the Ubuntu command line was accessible through a terminal that opened in Visual Studio Code. The Dfinity SDK and other essential components were installed, along with all necessary dependencies and tools, by following the Dfinity platform setup instructions. In order to facilitate JavaScript development and runtime, Node.js was also installed. By utilizing the strength and adaptability of the WSL-enabled Ubuntu system, this thorough setup made sure the development environment was ready for creating, testing, and deploying Dfinity applications and Node.js projects straight from Visual Studio Code. Fig 5.1 Dfinity Setup

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 21 Fig 5.2 Node.js Setup 2. Implementation steps for InnoVault : 1. Node.js Server Setup: Among other essential tasks, Node.js was utilized in the server setup to manage EJS files, render dashboards, and manage authentication. Express.js, a lightweight web framework for Node.js, was used in the server's construction and contributed to its effective route handling. Login credentials were strictly verified, and sessions were managed to ensure a seamless user experience and maintain program state. 2. Frontend Development: JavaScript was essential to the frontend development process since it scripted dynamic behaviors, improved user interactions, and guaranteed responsiveness. The foundation for organizing and presenting web content was made up of HTML, CSS, and Bootstrap for simplified styling. The dashboard's user interface was painstakingly created to give consumers a clear, visually appealing, and intuitive experience. Fig 5.3 Frontend Development

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 22 3. MongoDB Integration: Integration with MongoDB was essential for organizing and effectively storing off-chain data, especially user login credentials. The beautiful MongoDB object modeling tool, Mongoose, was made for Node.js and allowed for easy database interaction. User credentials were safely saved and retrieved during the authentication process by establishing schemas and utilizing Mongoose's robust querying capabilities, guaranteeing data integrity and secrecy. Fig 5.4 MongoDB integration 4. IPFS Storage: Files uploaded, including PDFs and photos, were safely kept by using the decentralized storage network offered by IPFS. The application included an easy-to-use "Contribute" button that made it simple for users to upload files. Files were uploaded, automatically stored on IPFS, and each user was given a different passcode. In order to guarantee that only authorized users could safely access and download their data, this passcode acted as a crucial authorization method. Fig 5.5 IPFS Storage

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 23 Fig 5.6 Uploading to IPFS 5. Dfinity Integration: An important project milestone was reached with the integration with Dfinity Internet Computer, which used its decentralized infrastructure to host the full web application. The native programming language of Dfinity, Motoko, was essential to securing digital assets and guaranteeing the resilience, scalability, and security of the application. Through the use of Dfinity, Innovault becomes a state-of-the-art decentralized application that has the potential to completely transform the protection of digital assets. Fig 5.7 Smart Contracts

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 24 6. Working of Innovault : 1. User authentication: Users must provide their login information in order to access Innovault. The Node.js server meticulously confirms the legitimacy of the user by comparing these credentials to the data that is kept in MongoDB. Users are automatically taken to the dashboard after completing the authentication process, where they can examine the features of the program. 2. Dashboard Navigation: With its unambiguous design and simple navigation options, the dashboard acts as users' main hub. "Explore" and "Contribute" are the two buttons that users will notice here. Users can interact with the site in a way that suits their requirements and tastes thanks to these possibilities. Fig 5.8: User Authentication and Dashboard Navigation 3. File Contribution: Contributing files is one of Innovault's primary features. Through the web application, users can easily contribute files by clicking the "Contribute" button. They are asked for necessary information, including the file name and extension. After uploading, the file is safely kept on IPFS, and the user is given a special passcode. To improve protection and control, this passcode serves as a key to guarantee that only authorized users can access and download the content in the future.

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 25 Fig 5.9 File Contribution Form 4. Exploring Files: Users can view a comprehensive page with all uploaded files in Innovault by using the "Explore" option. They can peruse the assortment of digital resources that different users have supplied here. However, by requiring users to provide the correct passcode, access to download these files is restricted. This strict security feature preserves the platform's integrity and confidentiality by guaranteeing that only authorized users can access the digital assets that have been saved. Fig 5.10 Exploring and downloading files

5.2 Implementation issues: During the project development, several challenges arose. Initially, setting up Ubuntu using WSL proved to be tricky due to configuration intricacies. Subsequently, configuring the Dfinity platform within VS Code using WSL presented its own set of challenges, requiring careful navigation of dependencies. Compatibility issues with different versions of Node.js caused discrepancies in functionality while managing webpack configuration file versions leading to frequent compilation errors. Implementing server-side rendering posed challenges in ensuring the correct pages were displayed based on user interactions, and developing a robust logic for passcode verification added complexity to the application's security mechanism. Designing smart contracts in Motoko for the Dfinity platform proved to be the most difficult task, with persistent errors requiring thorough debugging. Additionally, integrating styling and CSS into the UI proved to be challenging, demanding meticulous attention to detail to achieve the desired aesthetic appeal while maintaining functionality. Despite these obstacles, perseverance, collaboration, and meticulous problem-solving efforts ultimately allowed us to overcome these challenges and successfully complete the project. 5.3 Algorithms 1. MongoDB (User Details): In MongoDB, we make sure your personal information stays safe by using strong encryption methods like AES (Advanced Encryption Standard). This means your data is encrypted before it's even stored in the database, keeping it secure from prying eyes. Plus, MongoDB supports authentication methods like SCRAM (Salted Challenge Response Authentication Mechanism), so only you can access your account with your secure login details. With these measures in place, you can trust that your user details are kept confidential and protected from unauthorized access. 2. IPFS (Files): When it comes to storing your files securely, IPFS has got you covered. We use advanced cryptographic hashing algorithms to create unique addresses for your files, ensuring their integrity and preventing tampering. And for an extra layer of protection, we offer encryption options for your files before they're uploaded to the network. This means that only you, or those you authorize, can access and decrypt your files, keeping your data safe and private.

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 27 3. Motoko (Smart Contracts on Dfinity): With Motoko smart contracts on Dfinity, we prioritize security to ensure your transactions are safe and reliable. Motoko comes packed with features like type safety and memory safety, reducing the risk of errors or vulnerabilities in your smart contracts. Plus, with built-in access control and permission management, you have full control over who can interact with your contracts and what actions they can perform. This means you can trust that your smart contracts are executed securely, protecting your assets and ensuring smooth operations. In implementing the Innovault project, we encountered various challenges and employed a diverse range of technologies to ensure the security and functionality of the platform. Setting up Ubuntu with WSL and configuring the Dfinity platform using VS Code presented initial hurdles, while managing compatibility issues with Node.js versions and webpack configuration files proved challenging. Additionally, implementing server-side rendering and designing secure passcode verification logic required careful consideration and problem-solving. Despite these obstacles, MongoDB was utilized for storing user details securely, while IPFS provided decentralized storage for files, ensuring data integrity and confidentiality. Smart contracts were developed using Motoko on Dfinity, prioritizing security with built-in features like type safety and access control. Throughout the development process, user-centric design principles guided the creation of an intuitive dashboard interface, empowering users to explore and contribute to the platform effortlessly. By leveraging these technologies and addressing challenges with perseverance and collaboration, Innovault emerged as a robust decentralized application, safeguarding digital assets while providing a seamless user experience.

CHAPTER 6 TESTING In the testing phase of the Innovault project, our focus shifted to ensuring the reliability, security, and performance of the application. Testing plays a crucial role in validating the functionality of each component and ensuring that it meets the specified requirements. Through a combination of unit testing, integration testing, and end-to-end testing, we aim to identify and rectify any issues or vulnerabilities before deployment. This phase involves rigorous testing of user authentication, file upload and storage, smart contract functionality, and overall system behavior. By employing various testing methodologies and tools, we strive to deliver a robust and resilient application that meets the expectations of our users while maintaining the highest standards of quality and security. 6.1 Testing Environment In our testing environment for Innovault, we focused on verifying the functionality and reliability of the application without relying on third-party tools. We conducted unit testing for the Node.js server-side code and Motoko smart contracts by manually verifying individual components' behavior and functionality. Integration testing was performed by directly testing the interaction between different modules within the application. For end-to-end testing, we meticulously tested user flows and system behavior to ensure seamless functionality from user interaction to backend processes. Additionally, we ensured that MongoDB and IPFS configurations in the testing environment mirrored those in production to accurately assess the application's behavior. By prioritizing thorough code review and manual testing, we aimed to identify and address any issues or vulnerabilities, ensuring the quality and reliability of Innovault across its entire tech stack. 6.2 Unit Testing of Modules 6.2.1 Module 1 Steps Test Data Expected Results Observed Results Remarks Step 1: Test Authentication Module User credentials Successful login, logout and session management Authentications function as expected Pass Step 2: Test Dashboard Rendering User session date Accurate rendering of dashboard components and UI elements Accurate rendering of dashboard components and UI elements Pass

Step 3: Test File Upload module File data, user credentials Proper file upload handling and secure passcode generation Proper file upload handling and secure passcode generation Pass Step 4: Test Database Interaction Database queries Reliable data retrieval, storage, and manipulation Reliable data retrieval, storage, and manipulation Pass 6.2.2 Module 2 Steps Test Data Expected Results Observed Results Remarks Step 1: Validate Passcode Verification Contract Passcode data Robust passcode generation and verification Robust passcode generation and verification Pass Step 2: Confirm Authorization Contract Authorization requests Proper implementation of user authorization and access control Not implementing contract authorization on user's authentication only through passcode verification Fail Step 3: Validate File storage Contract File data Correct storage and retrieval of files on IPFS Correct storage and retrieval of files on IPFS Pass 6.2.3 Module 3 Steps Test Data Expected Results Observed Results Remarks Step 1: Integration Testing- API Endpoints Various API requests Seamless functionality between API endpoints and components Seamless functionality between API endpoints and components Pass Step 2: Integration Testing- Node.js & MongoDB Database interaction scenarios Smooth data retrieval and storage operations Smooth data retrieval and storage operations Pass Step 3: Integration Testing- IPFS File storage/retrieval requests Successful interaction with IPFS Successful interaction with IPFS Pass 6.2.4 Module 4 Steps Test Data Expected Results Observed Results Remarks Step 1: End- to-End Testing Full user workflows Seamless integration and functionality across all components Seamless integration and functionality across all components Pass Table 6.1 Node.js Server-side Test cases Table 6.2 Motoko Smart Contracts Test cases Table 6.Integration Testing Test cases Table 6.4 End-to-End Testing Test cases

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 30 CHAPTER 7 RESULTS 1. Home Page: Here we can see that users will be able to register or log in using this particular homepage designed for Innovault. Fig 7.1: Homepage On the homepage, we have provided two buttons for the users to register and log in, once any of them are clicked the users are navigated to the corresponding pages. 2. Register Page and Login Page: This page is provided so that the users can register themselves using their email and also they must provide a password for the complete registration. We have also provided users a way to log in through their Google accounts. This is implemented using the Google OAuth functionality on the server side. The registered details are stored on the MongoDB cloud.

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 31 Fig 7.2 Register Page The login page is also provided for the users who have already registered. Users who have not registered are not allowed to login through the login page, registration is mandatory. Fig 7.3 Login Page Fig 7.4 User Data in MongoDB Cloud

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 32 3. Dashboard Page: Fig 7.5 Dashboard Page In the dashboard page provided the user can view all the files uploaded by various users with their file names. There are mainly three main functionalities provided in this dashboard, first one is to navigate to the page which will allow the users to upload their own digital assets or files, and the second is to explore or view all the files uploaded by all users through our Innovault web application. The third one is the simple user functionality to logout. Additionally, we have also provided a link to our institution and another feature to delete the account. 4. Upload Page: Fig 7.6 Upload Page

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 33 Fig 7.7 Uploading a PDF File Fig 7.8 Providing File name When the user navigates to the upload page when they want to upload any file they will come across a form where they have to choose the file to upload from their device which can be of the format PDF or any image. Then they have to provide the file name with the extension before hitting the upload file button. After hitting the upload file button, the file is uploaded and the user is provided with the file's CID and passcode for maintaining security. The users can share their passcode with others, hence only authorized users can download this file by providing the passcode.

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 34 Fig 7.9 File details 5. Explore Page: When the users land on this page they can see all the files which are uploaded till then. This particular page provides the user the functionality to download the files displayed on this explore page. But again, this can only be done if the user provides the right passcode to the right file, if the passcode is not correct or if it does not match to the corresponding file then downloading is not successful. Fig 7.10 Explore Page

6. Download Page: If the users click the download button for any particular file through the explore page then they are navigated to this particular page, here they have to submit the passcode for that particular file in order to begin the download process. Fig 7.11: Download Page If the passcode of the file provided is correct then after clicking the submit button the download process must begin and in few seconds the file will be downloaded and displayed to the user. If the passcode is not correct an error message is displayed saying invalid passcode. File 7.12 Downloaded File

InnoVault 2023-2024 Dept of CSE, GAT, Bengaluru-98 Page | 36 CHAPTER 8 CONCLUSION In conclusion, Innovault efficiently integrates the Internet Computer (IC) platform through blockchain technology, making sure that the intellectual property is secured and undergoes decentralized management. The user interface lays out a user-friendly experience, permitting the users to log in, register, and interact on the dashboard. Tasks like IP management, interfacing with blockchain, and executing smart contracts are handled by the application layer. Security is dominant, with popular hashing algorithms such as SHA-256 and Bcrypt for ensuring the safety of the user data. Innovault's creative idea leverages blockchain principles for decentralized storage, transparency, and security, marking a significant stride in intellectual property protection and management. Major Contributions: A number of cutting-edge technologies have been skillfully combined by the Innovault project to create a decentralized, user-friendly, and safe digital asset protection program. Using Node.js for server-side processing, MongoDB for user data storage, and IPFS for decentralized file storage, we ensured that efficiency and security were fulfilled. The Dfinity platform saw the implementation of Motoko smart contracts, which added an extra degree of security and decentralization while making use of the infrastructure of the Internet Computer to enable scalable and resilient operations. The comprehensive testing, which included unit, integration, and end-to-end testing, ensured the dependability and robustness of the application. Together, these efforts made it easier for users to submit, save, and retrieve content with confidence on a safe and user-friendly platform. Future Enhancements: Innovault has the potential to undergo a number of fascinating improvements in the future. By supporting more kinds of digital assets than only PDFs and photos, the application's usefulness may be increased. Sophisticated user authentication techniques like multi-factor authentication or biometric verification could be implemented to improve security even more. The entire user experience may be enhanced by adding more intuitive design components to the user interface and taking user feedback into account. Furthermore, incorporating sophisticated analytics to monitor usage trends and spot possible security risks might support preserving the integrity and functionality of the platform. Innovault's usefulness and reach could be increased by investigating cross-chain interoperability with other blockchain networks, making it a more useful tool for users all over the world.