VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi , Karnataka, INDIA

A Project Report
on

Innovault using Blockchain Technology

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted By

TJ LAKSHMI	**1GA20CS149**
SONUPRIYA KJ	**1GA20CS140**
SAAKSHI P	**1GA20CS120**

Under the Guidance of
RESHMA D’SOUZA
Assistant Professor

Department of Computer Science and Engineering
Accredited by NBA(2022-2025)
GLOBAL ACADEMY OF TECHNOLOGY
(Autonomous Institute, Affliated to VTU, Belagavi)
Rajarajeshwarinagar, Bengaluru - 560 098
2023 – 2024

GLOBAL ACADEMY OF TECHNOLOGY

(Autonomous Institute, Affliated to VTU, Belagavi)
Rajarajeshwarinagar, Bengaluru - 560 098

Department of Computer Science and Engineering

Accredited by NBA (2022-2025)

CERTIFICATE

Certified that the Project Entitled **"Innovault using Blockchain Technology"** carried out by **TJ LAKSHMI**, bearing **USN 1GA20CS149, SONUPRIYA KJ**, bearing **USN 1GA20CS140, SAAKSHI P**, bearing **USN 1GA20CS120,** Bonafede students of Global Academy of Technology, in partial fulfillment for the award of the **BACHELOR OF ENGINEERING** in **Computer Science and Engineering** from Visvesvaraya Technological University, Belagavi during the year 2023-2024. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the Report submitted to the department. The Partial Project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said Degree.

Reshma D'Souza Assistant Professor Dept. of CSE GAT, Bengaluru.	Dr. Kumaraswamy S Professor & Head Dept. of CSE GAT, Bengaluru.	Dr. N. Ranapratap Reddy Principal GAT, Bengaluru.

External Viva

Name of the Examiners	Signature with date
1	
2.	

GLOBAL ACADEMY OF TECHNOLOGY

(Autonomous Institute, Affliated to VTU, Belagavi)
Rajarajeshwarinagar, Bengaluru - 560 098

DECLARATION

We, **TJ LAKSHMI**, bearing **USN 1GA20CS149, SONUPRIYA KJ**, bearing **USN 1GA20CS140, SAAKSHI P**, bearing **USN 1GA20CS120,** students of Eighth Semester B.E, Department of Computer Science and Engineering, Global Academy of Technology, Rajarajeshwari Nagar Bengaluru, declare that the Project Work entitled "**Innovault using Blockchain Technology**" has been carried out by us and submitted in partial fulfillment of the course requirements forthe award of degree in **Bachelor of Engineering** in **Computer Science and Engineering** from **Visvesvaraya Technological University, Belagavi** during the academic year **2023- 2024**.

1.TJ LAKSHMI	1GA20CS149
2.SONUPRIYA KJ	1GA20CS140
3.SAAKSHI P	1GA20CS120

Place: Bengaluru

Date:

ABSTRACT

The InnoVault project improves intellectual property management by deploying blockchain and Decentralized applications (DApps), which address transparency and security limitations in centralized solutions. Its methodology includes comprehensive research, developing smart contracts, the incorporation of decentralized storage, and the design of a user interface that is straightforward. Key findings include the successful deployment of smart contracts for asset management, the integration of decentralized storage systems for increased data security, and the user interface's efficiency. In summary, InnoVault illustrates blockchain's transformational capacity for enhancing intellectual property protection and distribution. In a quickly changing digital ecosystem, it empowers creators, stimulates collaboration, and enables the seamless sharing of ideas by providing a safe, transparent, and decentralized.

ACKNOWLEDGEMENT

The satisfaction and the euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. The constant guidance of these people and the encouragement provided crowned our efforts with success and glory. Although it is not possible to thank all the members who helped with the completion of the phase - 1 project individually, we take this opportunity to express our gratitude to one and all.

We are grateful to the management and our institute, the **GLOBAL ACADEMY OF TECHNOLOGY,** for its very ideals and inspiration for having provided us with the facilities that made this phase-1 project a success.

We express our sincere gratitude to **Dr. N. Rana Pratap Reddy**, Principal, Global Academy of Technology for the support and encouragement.

We wish to place on record our thanks to **Dr. Kumaraswamy S**, Professor and HOD, Department of CSE, Global Academy of Technology, for the constant encouragement provided to us. We are indebted with a deep sense of gratitude for the constant inspiration, encouragement, timely guidance, and valid suggestions given to us by our guide **Reshma D'Souza**, Assistant Professor, Department of CSE, Global Academy of Technology.

We are thankful to all the staff members of the department for providing relevant information and helping in different capacities in carrying out this phase -1 project.

Last, but not least, we owe our debts to our parents, friends, and those who directly or indirectly have helped us to make the phase-1 project work a success.

TJ LAKSHMI	**1GA20CS149**
SONUPRIYA KJ	**1GA20CS140**
SAAKSHI P	**1GA20CS120**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

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.

- **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, document preparation, and deployment planning.

Ongoing maintenance and support ensure 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

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.

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.

2.3 Comparison of Literature

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.
Blockchain-based reliable image copyright protection	Design having Blockchain-based reliable image copyright protection system named BB-RICP 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 - Decentralizedledger 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

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.

Table 2.1 Literature Survey

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.

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, encryption, and third-partyAPIs 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.

CHAPTER 4
SYSTEM DESIGN

4.1 Design Overview

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.

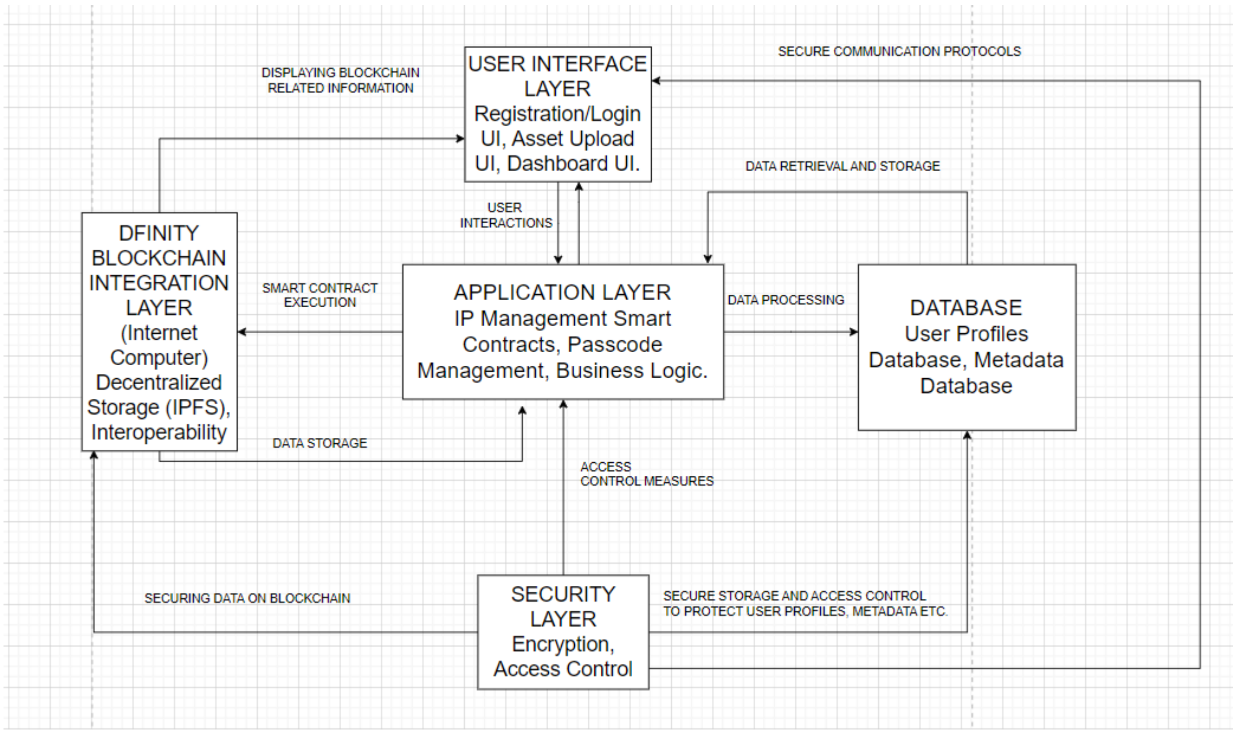

SECURE COMMUNICATION PROTOCOLS
DISPLAYING BLOCKCHAIN RELATED INFORMATION
USER INTERFACE LAYER Registration/Login UI, Asset Upload UI, Dashboard UI.
DATA RETRIEVAL AND STORAGE
USER INTERACTIONS
DFINITY BLOCKCHAIN INTEGRATION LAYER (Internet Computer) Decentralized Storage (IPFS), Interoperability
SMART CONTRACT EXECUTION
APPLICATION LAYER IP Management Smart Contracts, Passcode Management, Business Logic.
DATA PROCESSING
DATABASE User Profiles Database, Metadata Database
DATA STORAGE
ACCESS CONTROL MEASURES
SECURING DATA ON BLOCKCHAIN
SECURITY LAYER Encryption, Access Control
SECURE STORAGE AND ACCESS CONTROL TO PROTECT USER PROFILES, METADATA ETC.

Figure 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 users can view their accounts, log out, and explore 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 if any user wants to download any data, this is done by executing smart contract for passcode management. Whenever the user wants to fetch data this application layer interacts with the IPFS layer to fetch them.
3. Blockchain Integration Layer: This layer implements 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 standsas a comprehensive and secure platform for intellectual property management.

4.1 Data Flow Diagram

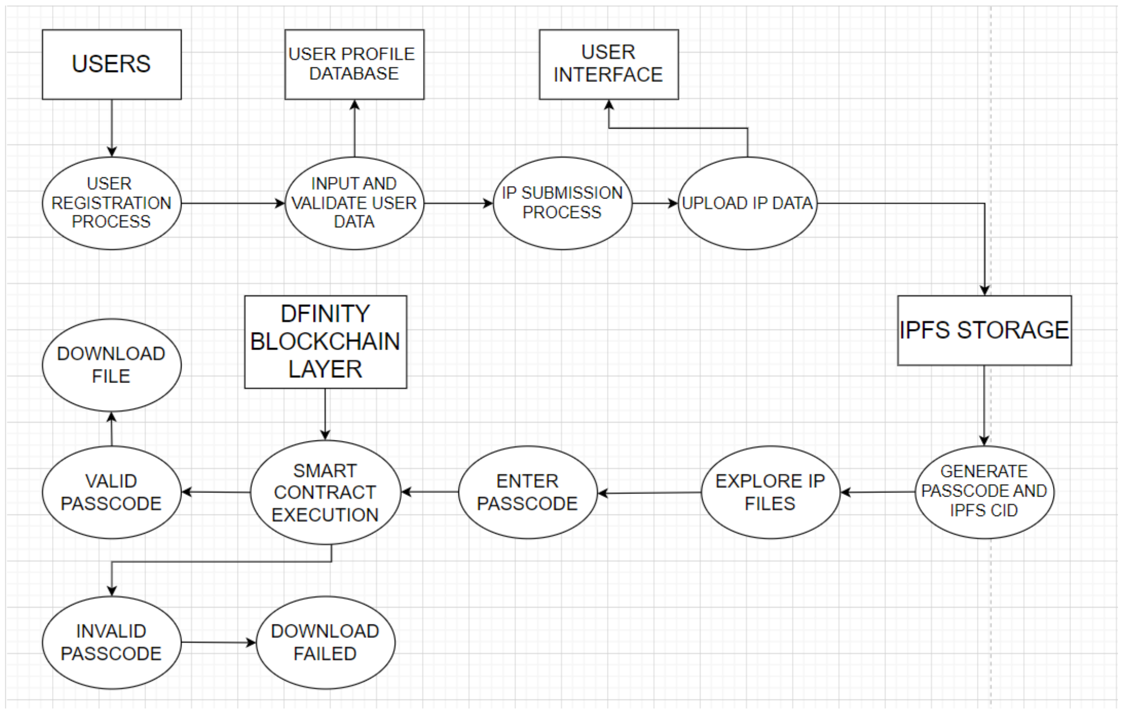


Figure 4.3.1: Dataflow of Innovault

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. The files uploaded are stored in IPFS storage. The user receives a unique passcode and CID (Content Identifier).
4. When a user wants to retrieve another user's digital assets, passcode-checking occurs. This process facilitates the validation of the user's request and ensures the user has the correct passcode to access the specified digital asset.
5. After the passcode is validated the download process of the file is complicated, if the passcode is incorrect the download process is failed and an error message is displayed to the user.

CHAPTER 5

IMPLEMENTATION

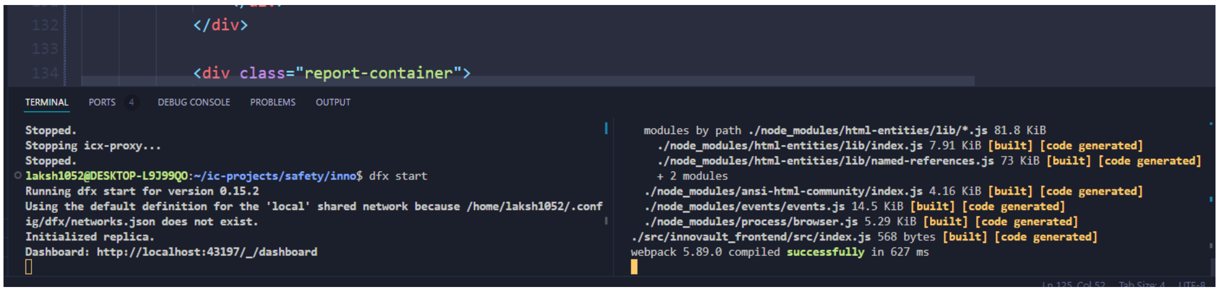
5.1 Steps for Implementation

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.

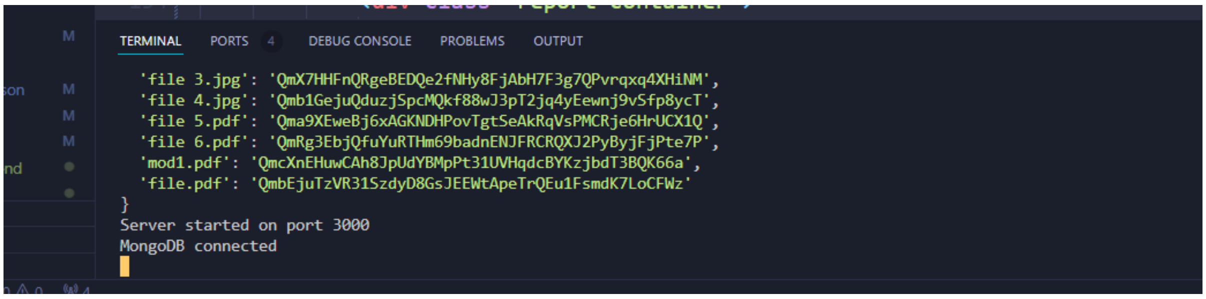


```
            </div>

            <div class="report-container">
```

```
Stopped.
Stopping icx-proxy...
Stopped.
laksh1052@DESKTOP-L9J99QO:~/ic-projects/safety/inno$ dfx start
Running dfx start for version 0.15.2
Using the default definition for the 'local' shared network because /home/laksh1052/.conf
ig/dfx/networks.json does not exist.
Initialized replica.
Dashboard: http://localhost:43197/_/dashboard
```

```
    modules by path ./node_modules/html-entities/lib/*.js 81.8 KiB
      ./node_modules/html-entities/lib/index.js 7.91 KiB [built] [code generated]
      ./node_modules/html-entities/lib/named-references.js 73 KiB [built] [code generated]
      + 2 modules
    ./node_modules/ansi-html-community/index.js 4.16 KiB [built] [code generated]
    ./node_modules/events/events.js 14.5 KiB [built] [code generated]
    ./node_modules/process/browser.js 5.29 KiB [built] [code generated]
  ./src/innovault_frontend/src/index.js 568 bytes [built] [code generated]
  webpack 5.89.0 compiled successfully in 627 ms
```

Fig 5.1: Dfinity Setup



```
  'file 3.jpg': 'QmX7HHFnQRgeBEDQe2fNHy8FjAbH7F3g7QPvrqxq4XHiNM',
  'file 4.jpg': 'Qmb1GejuQduzjSpcMQkf88wJ3pT2jq4yEewnj9vSfp8ycT',
  'file 5.pdf': 'Qma9XEweBj6xAGKNDHPovTgtSeAkRqVsPMCRje6HrUCX1Q',
  'file 6.pdf': 'QmRg3EbjQfuYuRTHm69badnENJFRCRQXJ2PyByjFjPte7P',
  'mod1.pdf': 'QmcXnEHuwCAh8JpUdYBMpPt31UVHqdcBYKzjbdT3BQK66a',
  'file.pdf': 'QmbEjuTzVR31SzdyD8GsJEEWtApeTrQEu1FsmdK7LoCFWz'
}
Server started on port 3000
MongoDB connected
```

Figure 5.2: Node.js Setup

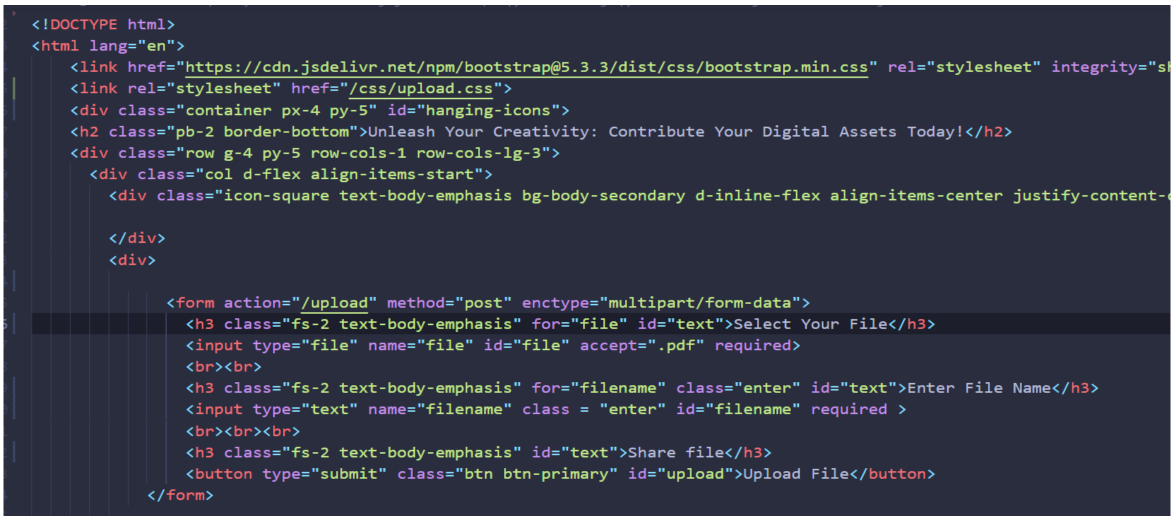
2. Implementation steps for InnoVault :

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.

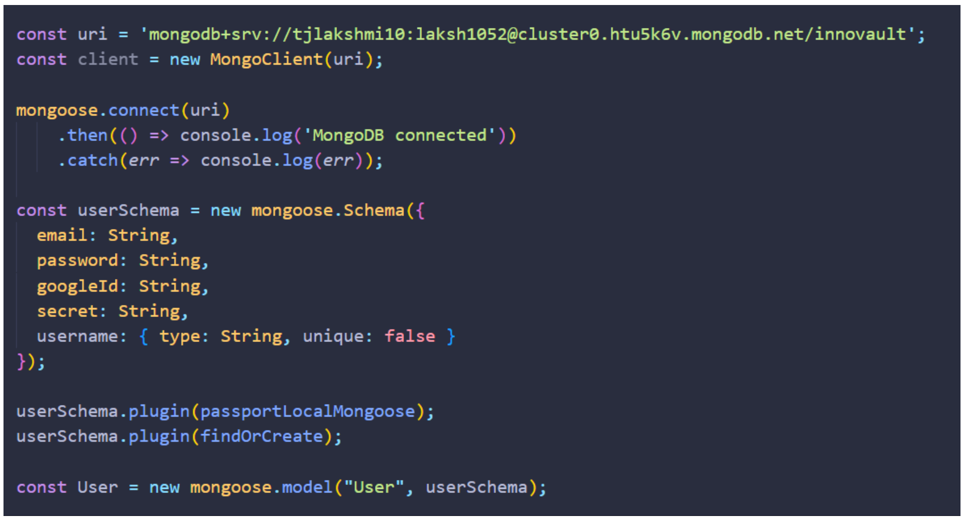


```html
<!DOCTYPE html>
<html lang="en">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sh
    <link rel="stylesheet" href="/css/upload.css">
    <div class="container px-4 py-5" id="hanging-icons">
    <h2 class="pb-2 border-bottom">Unleash Your Creativity: Contribute Your Digital Assets Today!</h2>
    <div class="row g-4 py-5 row-cols-1 row-cols-lg-3">
      <div class="col d-flex align-items-start">
        <div class="icon-square text-body-emphasis bg-body-secondary d-inline-flex align-items-center justify-content-

        </div>
        <div>

            <form action="/upload" method="post" enctype="multipart/form-data">
              <h3 class="fs-2 text-body-emphasis" for="file" id="text">Select Your File</h3>
              <input type="file" name="file" id="file" accept=".pdf" required>
              <br><br>
              <h3 class="fs-2 text-body-emphasis" for="filename" class="enter" id="text">Enter File Name</h3>
              <input type="text" name="filename" class = "enter" id="filename" required >
              <br><br><br>
              <h3 class="fs-2 text-body-emphasis" id="text">Share file</h3>
              <button type="submit" class="btn btn-primary" id="upload">Upload File</button>
          </form>
```

Figure 5.3: Frontend Development

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.

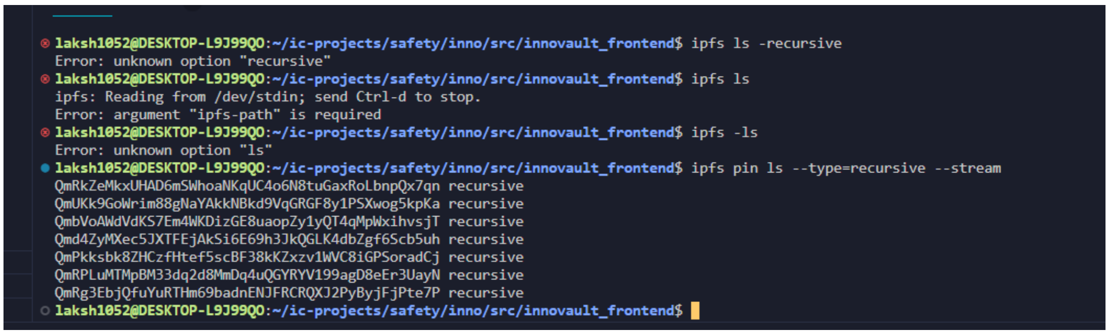


```
const uri = 'mongodb+srv://tjlakshmi10:laksh1052@cluster0.htu5k6v.mongodb.net/innovault';
const client = new MongoClient(uri);

mongoose.connect(uri)
    .then(() => console.log('MongoDB connected'))
    .catch(err => console.log(err));

const userSchema = new mongoose.Schema({
  email: String,
  password: String,
  googleId: String,
  secret: String,
  username: { type: String, unique: false }
});

userSchema.plugin(passportLocalMongoose);
userSchema.plugin(findOrCreate);

const User = new mongoose.model("User", userSchema);
```

Figure 5.4: MongoDB integration

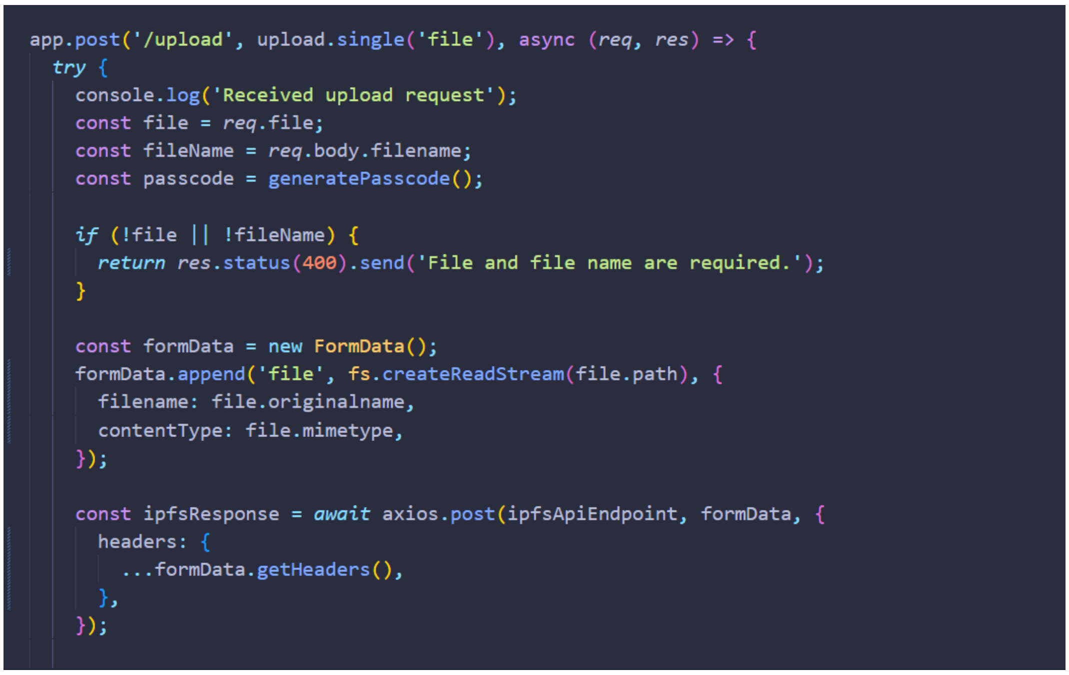
4. IPFS Storage:

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, and 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.



```
laksh1052@DESKTOP-L9J99QO:~/ic-projects/safety/inno/src/innovault_frontend$ ipfs ls -recursive
Error: unknown option "recursive"
laksh1052@DESKTOP-L9J99QO:~/ic-projects/safety/inno/src/innovault_frontend$ ipfs ls
ipfs: Reading from /dev/stdin; send Ctrl-d to stop.
Error: argument "ipfs-path" is required
laksh1052@DESKTOP-L9J99QO:~/ic-projects/safety/inno/src/innovault_frontend$ ipfs -ls
Error: unknown option "ls"
laksh1052@DESKTOP-L9J99QO:~/ic-projects/safety/inno/src/innovault_frontend$ ipfs pin ls --type=recursive --stream
QmRkZeMkxUHAD6mSWhoaNKqUC4o6N8tuGaxRoLbnpQx7qn recursive
QmUKk9GoWrim88gNaYAkkNBkd9VqGRGF8y1PSXwog5kpKa recursive
QmbVoAWdVdKS7Em4WKDizGE8uaopZy1yQT4qMpWxihvsjT recursive
Qmd4ZyMXec5JXTFEjAkSi6E69h3JkQGLK4dbZgf6Scb5uh recursive
QmPkksbk8ZHCzfHtef5scBF38kKZxzv1WVC8iGPSoradCj recursive
QmRPLuMTMpBM33dq2d8MmDq4uQGYRYV199agD8eEr3UayN recursive
QmRg3EbjQfuYuRTHm69badnENJFRCRQXJ2PyByjFjPte7P recursive
laksh1052@DESKTOP-L9J99QO:~/ic-projects/safety/inno/src/innovault_frontend$ 
```

Figure 5.5: IPFS Storage

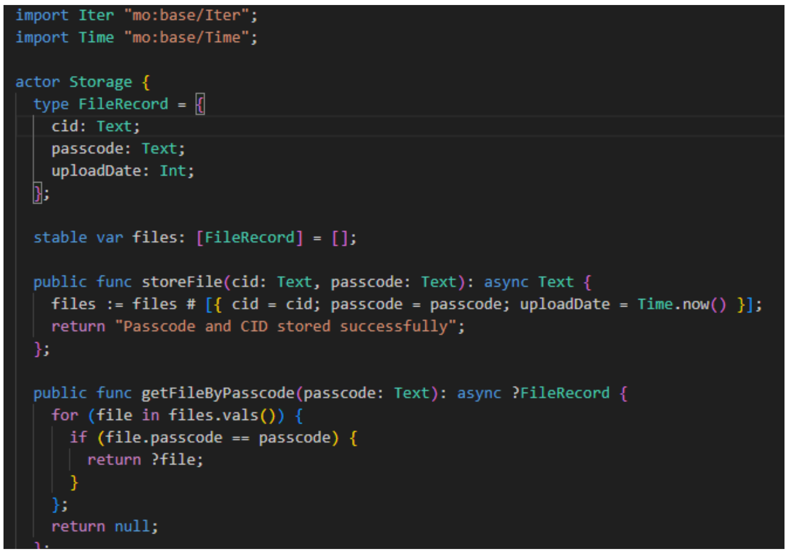


```
app.post('/upload', upload.single('file'), async (req, res) => {
  try {
    console.log('Received upload request');
    const file = req.file;
    const fileName = req.body.filename;
    const passcode = generatePasscode();

    if (!file || !fileName) {
      return res.status(400).send('File and file name are required.');
    }

    const formData = new FormData();
    formData.append('file', fs.createReadStream(file.path), {
      filename: file.originalname,
      contentType: file.mimetype,
    });

    const ipfsResponse = await axios.post(ipfsApiEndpoint, formData, {
      headers: {
        ...formData.getHeaders(),
      },
    });
```

Figure 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.



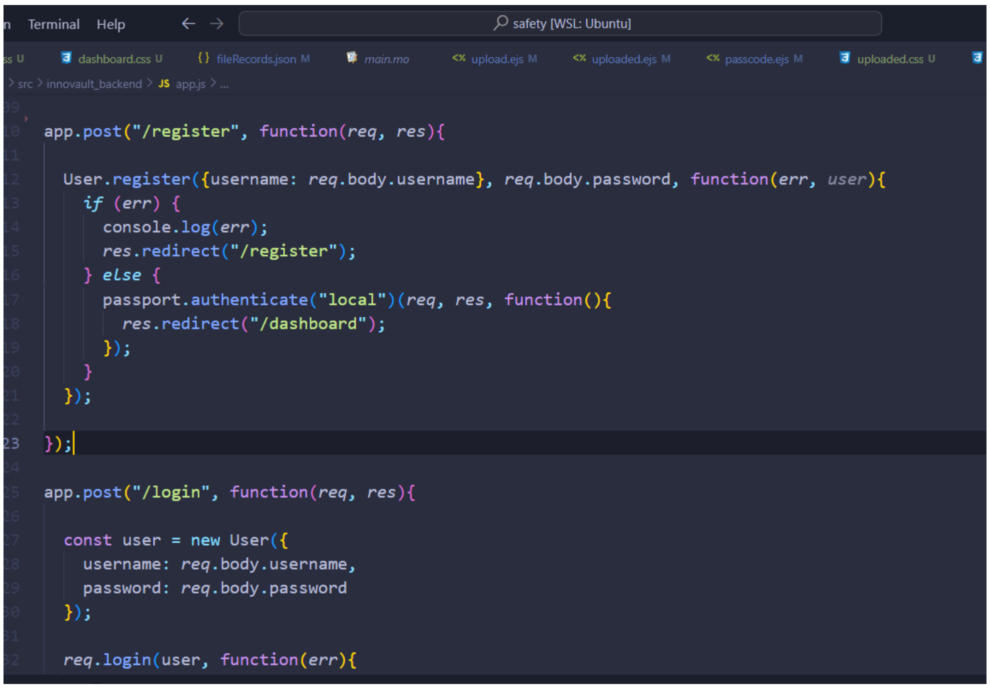
```
import Iter "mo:base/Iter";
import Time "mo:base/Time";

actor Storage {
  type FileRecord = {
    cid: Text;
    passcode: Text;
    uploadDate: Int;
  };

  stable var files: [FileRecord] = [];

  public func storeFile(cid: Text, passcode: Text): async Text {
    files := files # [{ cid = cid; passcode = passcode; uploadDate = Time.now() }];
    return "Passcode and CID stored successfully";
  };

  public func getFileByPasscode(passcode: Text): async ?FileRecord {
    for (file in files.vals()) {
      if (file.passcode == passcode) {
        return ?file;
      }
    };
    return null;
```

Figure 5.7: Smart Contracts

6. Working of Innovault :

1. User authentication: Users must provide their login information 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 program's features.

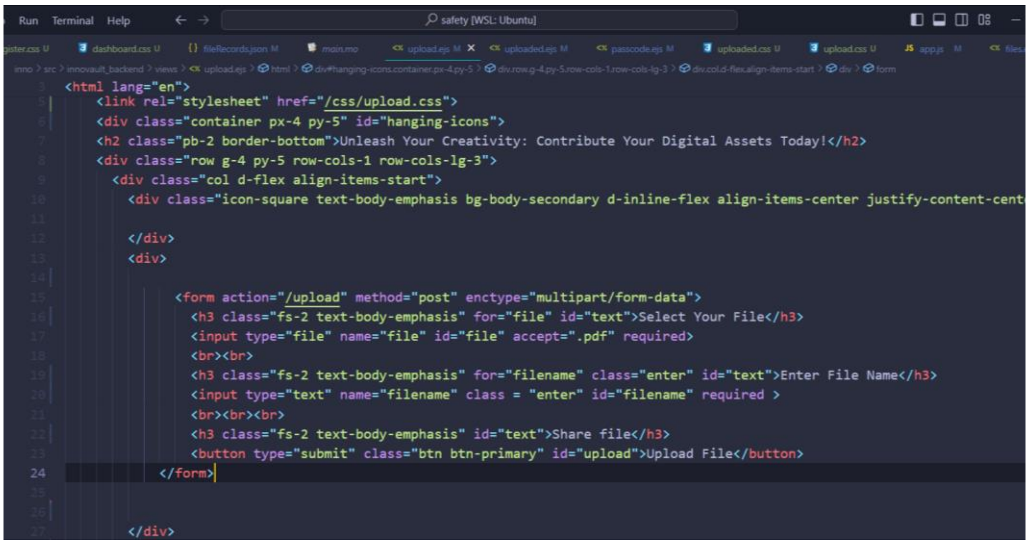
2. Dashboard Navigation: The dashboard acts as users' main hub with its unambiguous design and simple navigation options. "Explore" and "Contribute" are the two buttons that users will notice here. Users can interact with the site in a way that suits their requirements and tastes thanks to these possibilities.



```
app.post("/register", function(req, res){

  User.register({username: req.body.username}, req.body.password, function(err, user){
    if (err) {
      console.log(err);
      res.redirect("/register");
    } else {
      passport.authenticate("local")(req, res, function(){
        res.redirect("/dashboard");
      });
    }
  });

});

app.post("/login", function(req, res){

  const user = new User({
    username: req.body.username,
    password: req.body.password
  });

  req.login(user, function(err){
```

Figure 5.8: User Authentication and Dashboard Navigation

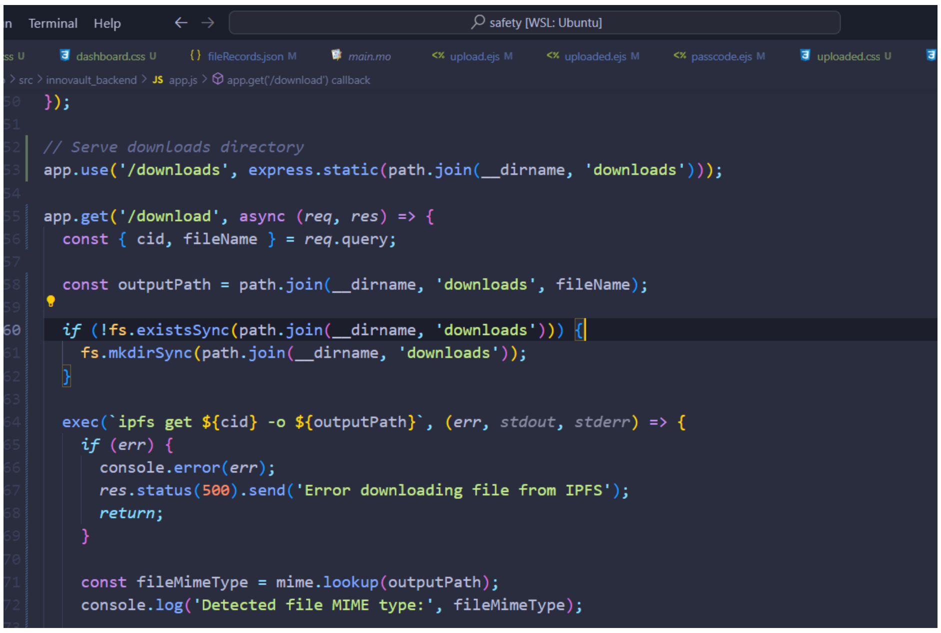
3. File Contribution: Contributing files is one of Innovault's primary features. Users can easily contribute files through the web application by clicking the "Contribute" button. They are asked for necessary information, including the file name and extension. After uploading, the file is safely kept 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.



```
<html lang="en">
    <link rel="stylesheet" href="/css/upload.css">
    <div class="container px-4 py-5" id="hanging-icons">
    <h2 class="pb-2 border-bottom">Unleash Your Creativity: Contribute Your Digital Assets Today!</h2>
    <div class="row g-4 py-5 row-cols-1 row-cols-lg-3">
      <div class="col d-flex align-items-start">
        <div class="icon-square text-body-emphasis bg-body-secondary d-inline-flex align-items-center justify-content-cent

        </div>
        <div>

              <form action="/upload" method="post" enctype="multipart/form-data">
                <h3 class="fs-2 text-body-emphasis" for="file" id="text">Select Your File</h3>
                <input type="file" name="file" id="file" accept=".pdf" required>
                <br><br>
                <h3 class="fs-2 text-body-emphasis" for="filename" class="enter" id="text">Enter File Name</h3>
                <input type="text" name="filename" class = "enter" id="filename" required >
                <br><br><br>
                <h3 class="fs-2 text-body-emphasis" id="text">Share file</h3>
                <button type="submit" class="btn btn-primary" id="upload">Upload File</button>
            </form>


        </div>
```

Figure 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.



```
});

// Serve downloads directory
app.use('/downloads', express.static(path.join(__dirname, 'downloads')));

app.get('/download', async (req, res) => {
  const { cid, fileName } = req.query;

  const outputPath = path.join(__dirname, 'downloads', fileName);

  if (!fs.existsSync(path.join(__dirname, 'downloads'))) {
    fs.mkdirSync(path.join(__dirname, 'downloads'));
  }

  exec(`ipfs get ${cid} -o ${outputPath}`, (err, stdout, stderr) => {
    if (err) {
      console.error(err);
      res.status(500).send('Error downloading file from IPFS');
      return;
    }

    const fileMimeType = mime.lookup(outputPath);
    console.log('Detected file MIME type:', fileMimeType);
```

Figure 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.

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. We aim to identify and rectify any issues or vulnerabilities before deployment through a combination of unit testing, integration testing, and end-to-end testing. 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 data, user	Proper file upload	Proper file upload	Pass

File Upload module	credentials	handling and secure passcode generation	handling and secure passcode generation	
Step 4: Test Database Interaction	Database queries	Reliable data retrieval, storage, and manipulation	Reliable data retrieval, storage, and manipulation	Pass

Table 6.1 Node.js Server-side Test cases

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

Table 6.2 Motoko Smart Contracts Test cases

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

Table 6.Integration Testing Test cases

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 6.4 End-to-End Testing Test cases

CHAPTER 7

RESULTS

1. **Home Page:**

Here we can see that users can register or log in using this particular homepage designed for Innovault.

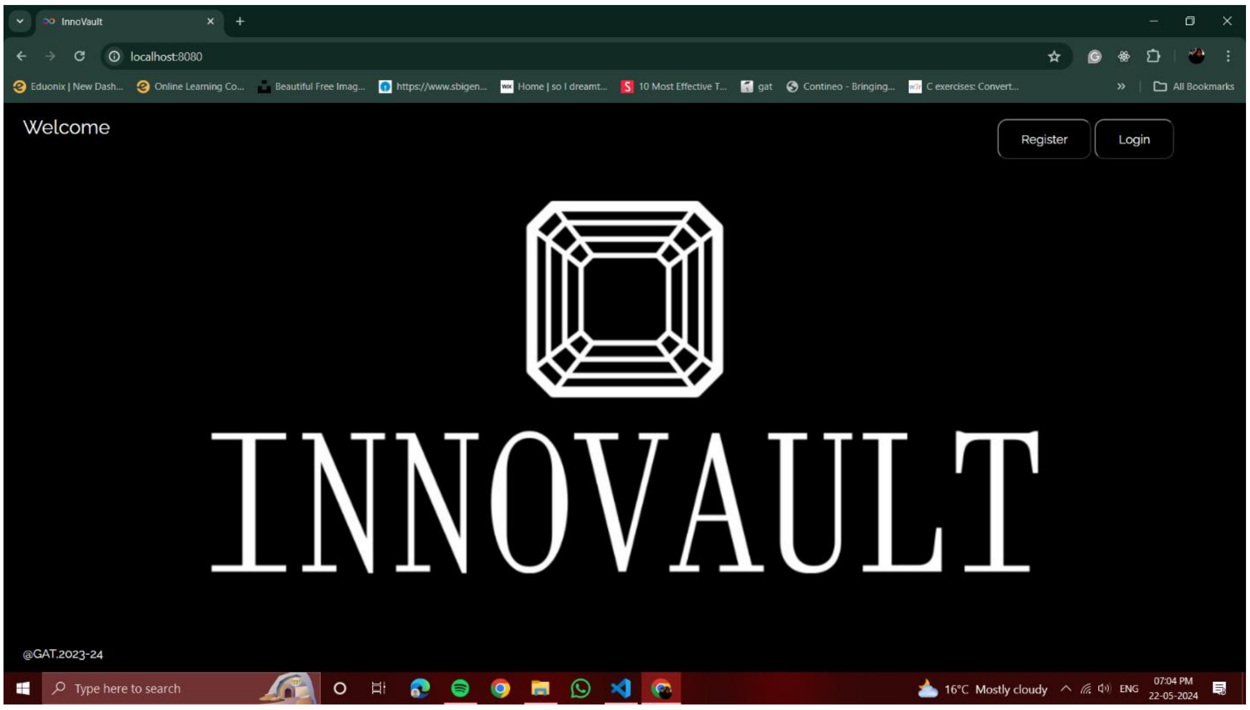


Figure 7.1: Homepage

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.

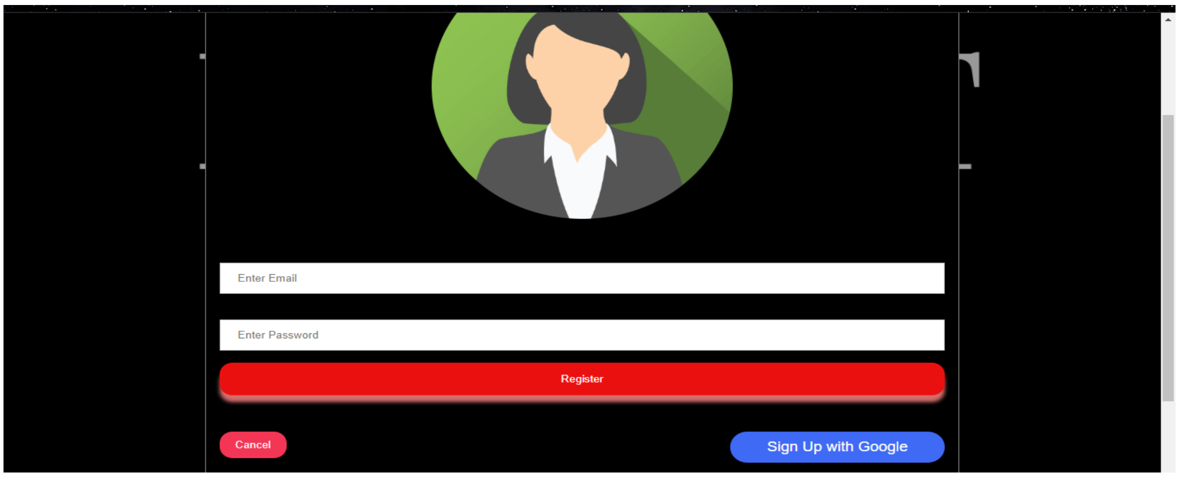


Figure 7.2: Register Page

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.

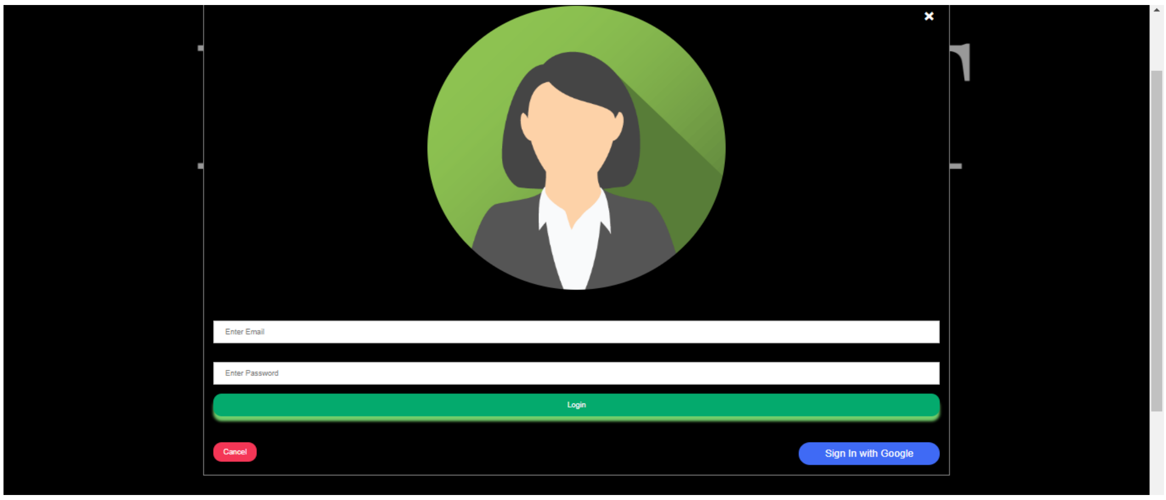
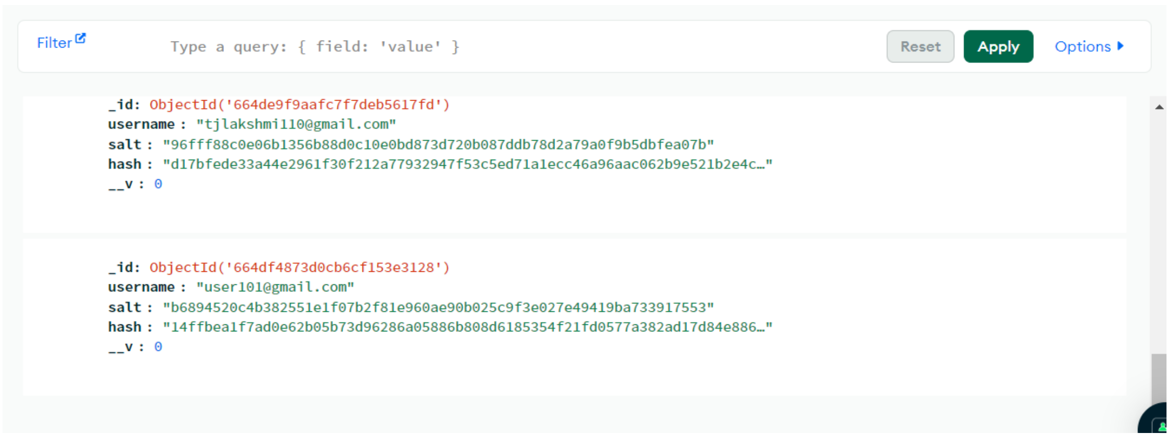


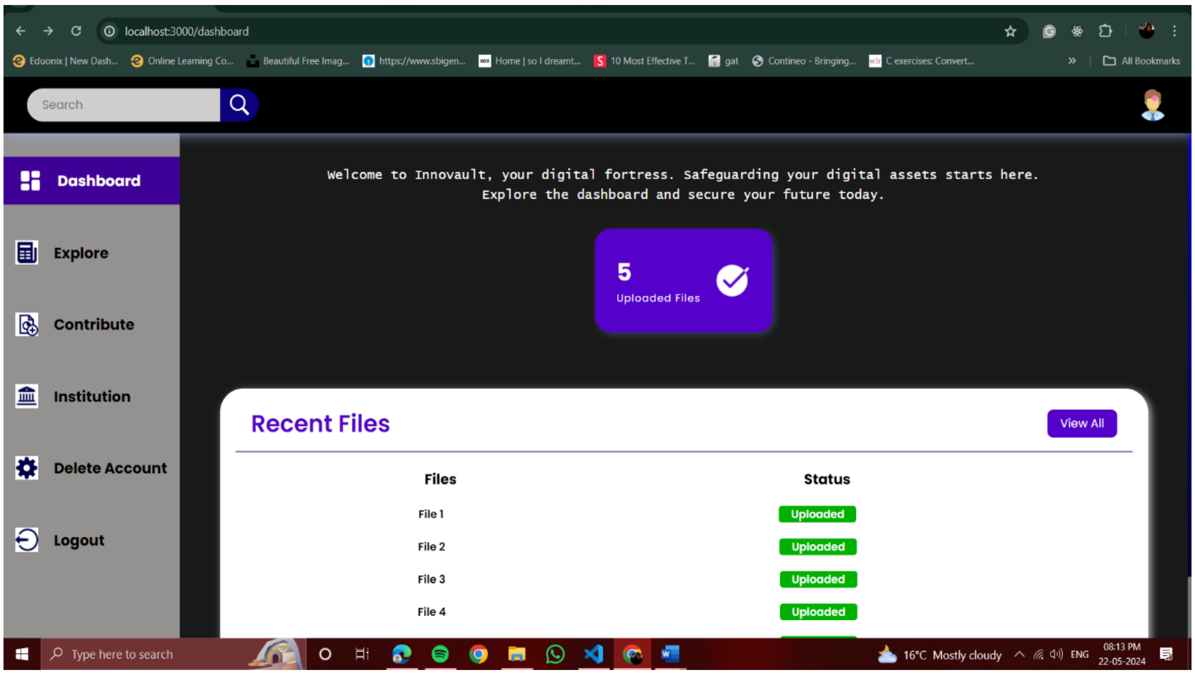
Figure 7.3: Login Page



```
Type a query: { field: 'value' }

_id: ObjectId('664de9f9aafc7f7deb5617fd')
username : "tjlakshmi110@gmail.com"
salt : "96fff88c0e06b1356b88d0c10e0bd873d720b087ddb78d2a79a0f9b5dbfea07b"
hash : "d17bfede33a44e2961f30f212a77932947f53c5ed71a1ecc46a96aac062b9e521b2e4c…"
__v : 0

_id: ObjectId('664df4873d0cb6cf153e3128')
username : "user101@gmail.com"
salt : "b6894520c4b382551e1f07b2f81e960ae90b025c9f3e027e49419ba733917553"
hash : "14ffbea1f7ad0e62b05b73d96286a05886b808d6185354f21fd0577a382ad17d84e886…"
__v : 0
```

Figure 7.4: User Data in MongoDB Cloud

3. Dashboard Page:



Welcome to Innovault, your digital fortress. Safeguarding your digital assets starts here.
Explore the dashboard and secure your future today.

5
Uploaded Files

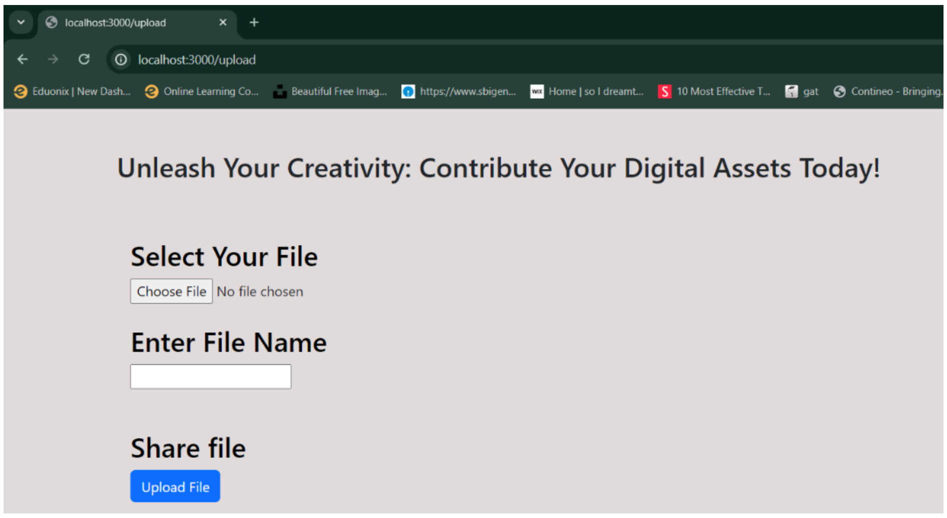
Recent Files

Files	Status
File 1	Uploaded
File 2	Uploaded
File 3	Uploaded
File 4	Uploaded

Figure 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:



Unleash Your Creativity: Contribute Your Digital Assets Today!

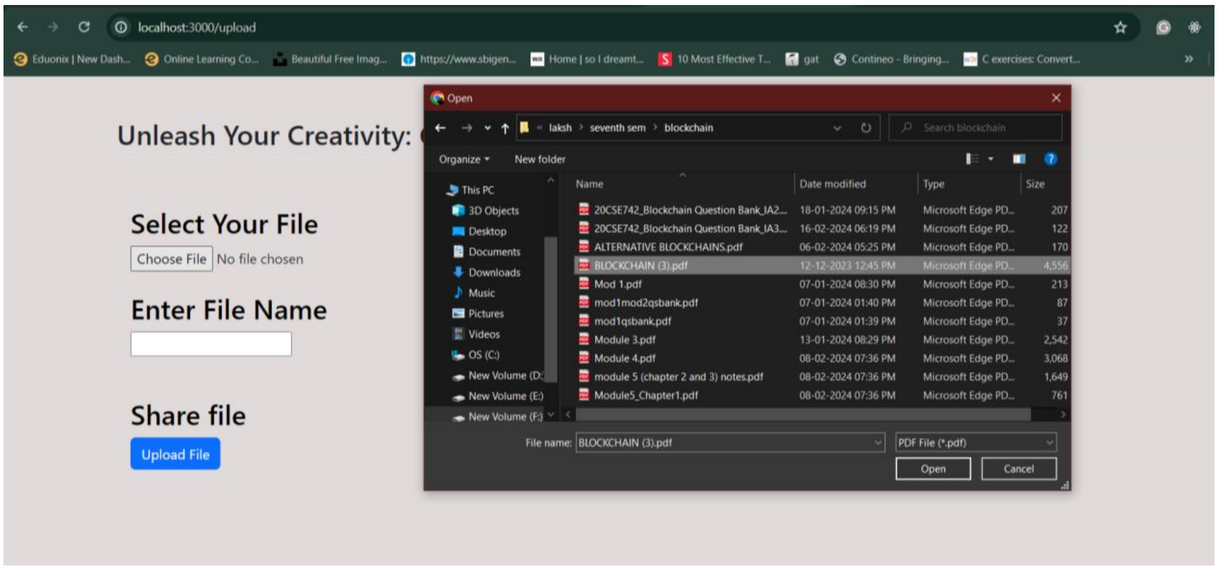
Select Your File

No file chosen

Enter File Name

Share file

Figure 7.6: Upload Page



localhost:3000/upload

Unleash Your Creativity:

Select Your File

Choose File No file chosen

Enter File Name

Share file

Upload File

Open

laksh › seventh sem › blockchain

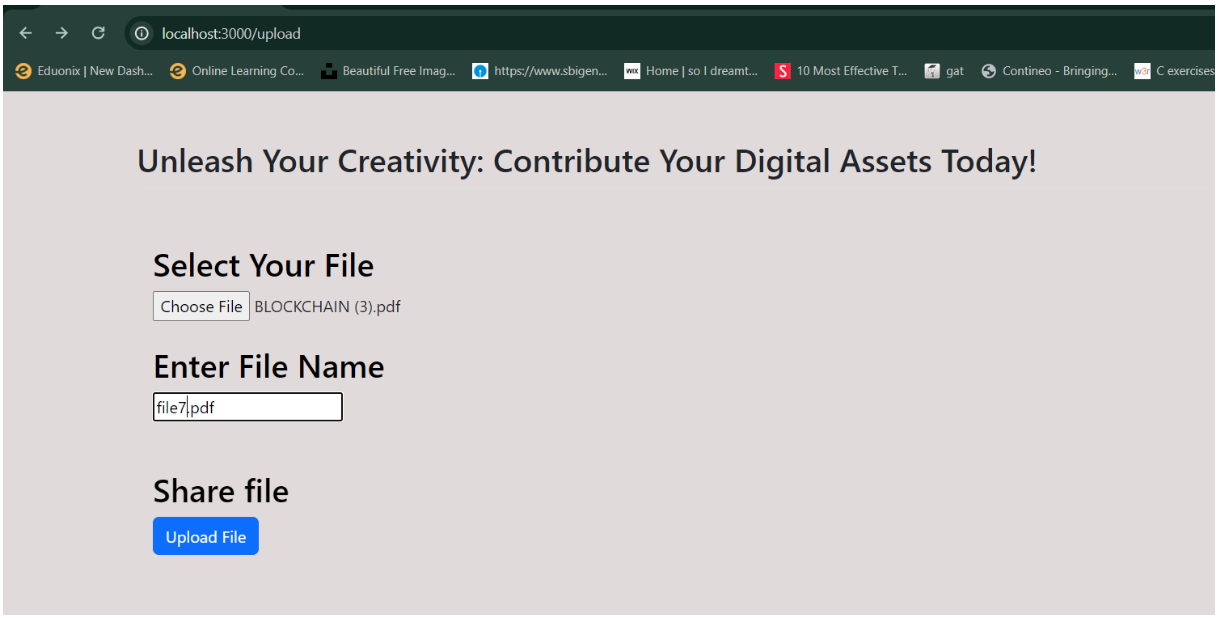
Name	Date modified	Type	Size
20CSE742_Blockchain Question Bank_IA2....	18-01-2024 09:15 PM	Microsoft Edge PD...	207
20CSE742_Blockchain Question Bank_IA3....	16-02-2024 06:19 PM	Microsoft Edge PD...	122
ALTERNATIVE BLOCKCHAINS.pdf	06-02-2024 05:25 PM	Microsoft Edge PD...	170
BLOCKCHAIN (3).pdf	12-12-2023 12:45 PM	Microsoft Edge PD...	4,556
Mod 1.pdf	07-01-2024 08:30 PM	Microsoft Edge PD...	213
mod1mod2qsbank.pdf	07-01-2024 01:40 PM	Microsoft Edge PD...	87
mod1qsbank.pdf	07-01-2024 01:39 PM	Microsoft Edge PD...	37
Module 3.pdf	13-01-2024 08:29 PM	Microsoft Edge PD...	2,542
Module 4.pdf	08-02-2024 07:36 PM	Microsoft Edge PD...	3,068
module 5 (chapter 2 and 3) notes.pdf	08-02-2024 07:36 PM	Microsoft Edge PD...	1,649
Module5_Chapter1.pdf	08-02-2024 07:36 PM	Microsoft Edge PD...	761

File name: BLOCKCHAIN (3).pdf

PDF File (*.pdf)

Open Cancel

Figure 7.7: Uploading a PDF File



localhost:3000/upload

Unleash Your Creativity: Contribute Your Digital Assets Today!

Select Your File

Choose File BLOCKCHAIN (3).pdf

Enter File Name

file7.pdf

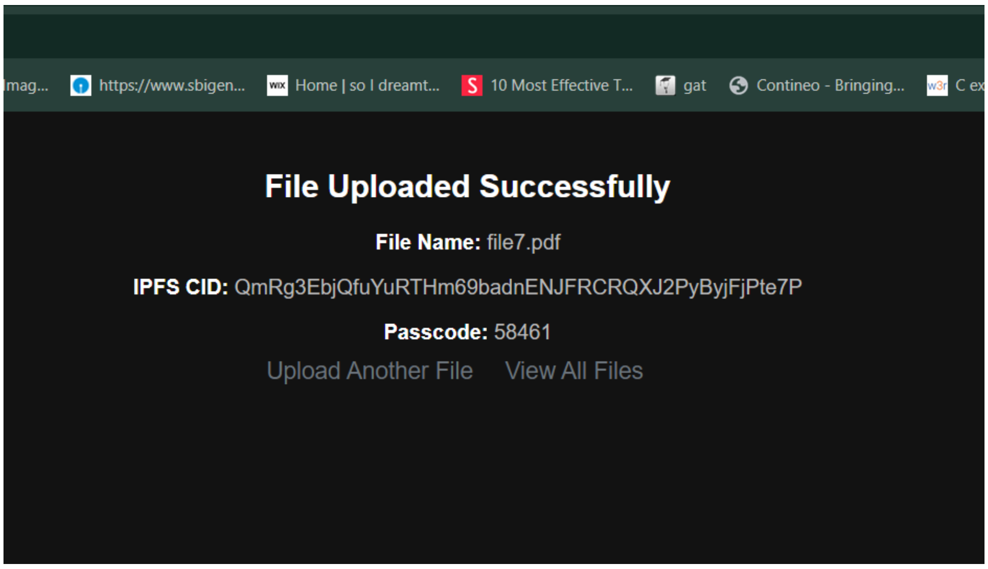
Share file

Upload File

Figure 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.



File Uploaded Successfully

File Name: file7.pdf

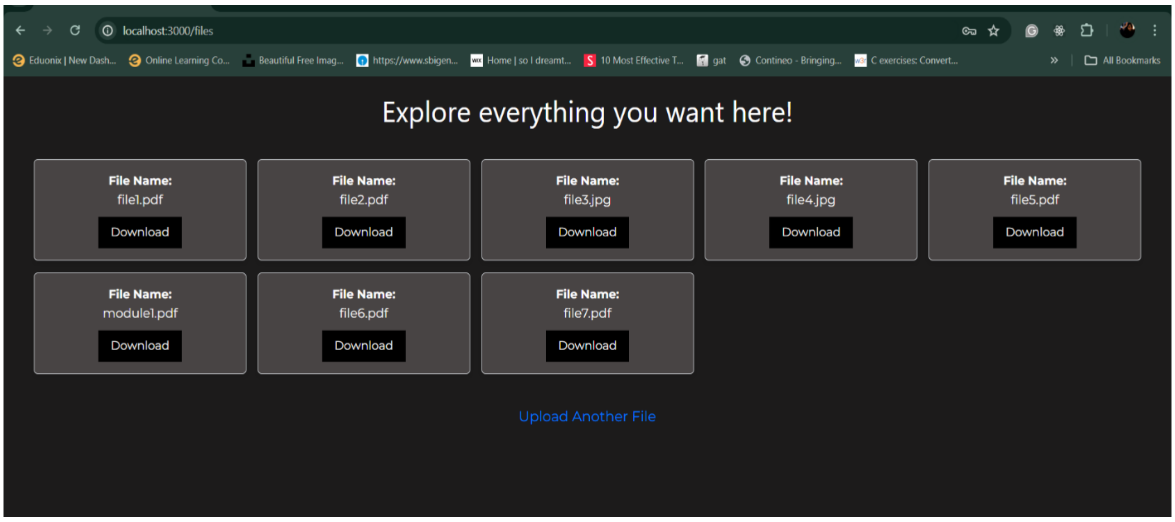
IPFS CID: QmRg3EbjQfuYuRTHm69badnENJFRCRQXJ2PyByjFjPte7P

Passcode: 58461

Figure 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.



localhost:3000/files

Explore everything you want here!

File Name: file1.pdf
File Name: file2.pdf
File Name: file3.jpg
File Name: file4.jpg
File Name: file5.pdf
File Name: module1.pdf
File Name: file6.pdf
File Name: file7.pdf

Figure 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.

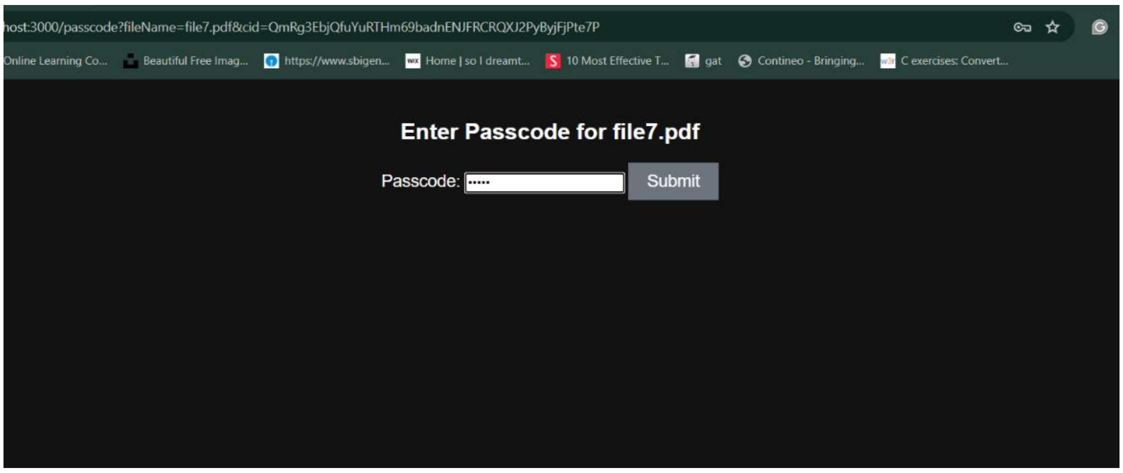
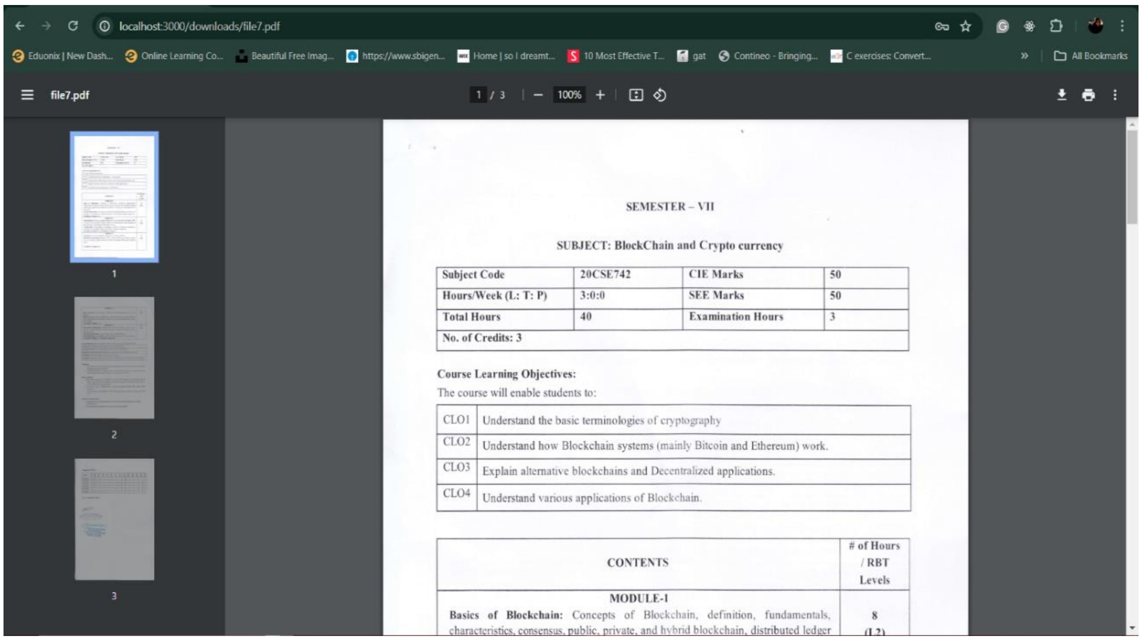

host:3000/passcode?fileName=file7.pdf&cid=QmRg3EbjQfuYuRTHm69badnENJFRCRQXJ2PyByjFjPte7P
Enter Passcode for file7.pdf
Passcode:
Submit

Figure 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.



SEMESTER – VII

SUBJECT: BlockChain and Crypto currency

Subject Code	20CSE742	CIE Marks	50
Hours/Week (L: T: P)	3:0:0	SEE Marks	50
Total Hours	40	Examination Hours	3
No. of Credits: 3			

Course Learning Objectives:

The course will enable students to:

CLO1	Understand the basic terminologies of cryptography
CLO2	Understand how Blockchain systems (mainly Bitcoin and Ethereum) work.
CLO3	Explain alternative blockchains and Decentralized applications.
CLO4	Understand various applications of Blockchain.

CONTENTS	# of Hours / RBT Levels
MODULE-1 Basics of Blockchain: Concepts of Blockchain, definition, fundamentals, characteristics, consensus, public, private, and hybrid blockchain, distributed ledger	8 (L2)

Figure 7.12: Downloaded File

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 several 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.

REFERENCES

[1] Singh, B P Tripathi, Anand Kumar: Blockchain Technology and Intellectual Property Rights. JIPR Vol.24(1-2) [January-March 2019].

[2] Chandratre, Atharv and Pathak, Abhinav, Blockchain Based Intellectual Property Management (December 10, 2019). Available atSSRN: https://ssrn.com/abstract=3800734 or http://dx.doi.org/10.2139/ssrn.3800734.

[3] Yanhui liu, Zianbiao Zhang, Shupei Wu: Research on digital copyright protection based on the hyperledger fabric blockchain network technology. September 2021, PeerJ Computer Science 7(1):e709, DOI:10.7717/peerj-cs.709

[4] Raffaele Fabio Ciriello, Alexandra Cecilie Gjøl Torbensen, Magnus Rotvit Perlt Hansen, Christoph Mueller-Bloch: Blockchain-based digital rights management systems: Design principles for the music industry. April 2023, Electronic Markets 33(1), DOI:10.1007/s12525-023-00628-5

[5] Martin Zeilinger: Digital Art as 'Monetised Graphics': Enforcing Intellectual Property on the Blockchain. March 2018, Philosophy & Technology 31(1), DOI:10.1007/s13347-016-0243-1

[6] Xiangli Xiao, Xiaotong He, Yushu Zhang, Xuewen Dong: Blockchain-based reliable image copyright protection. March 2023, IET Blockchain 3(4), DOI:10.1049/blc2.12027

[7] M. I. Khalid et al., "A Comprehensive Survey on Blockchain-Based Decentralized Storage Networks," in IEEE Access, vol. 11, pp. 10995-11015, 2023, doi: 10.1109/ACCESS.2023.3240237.

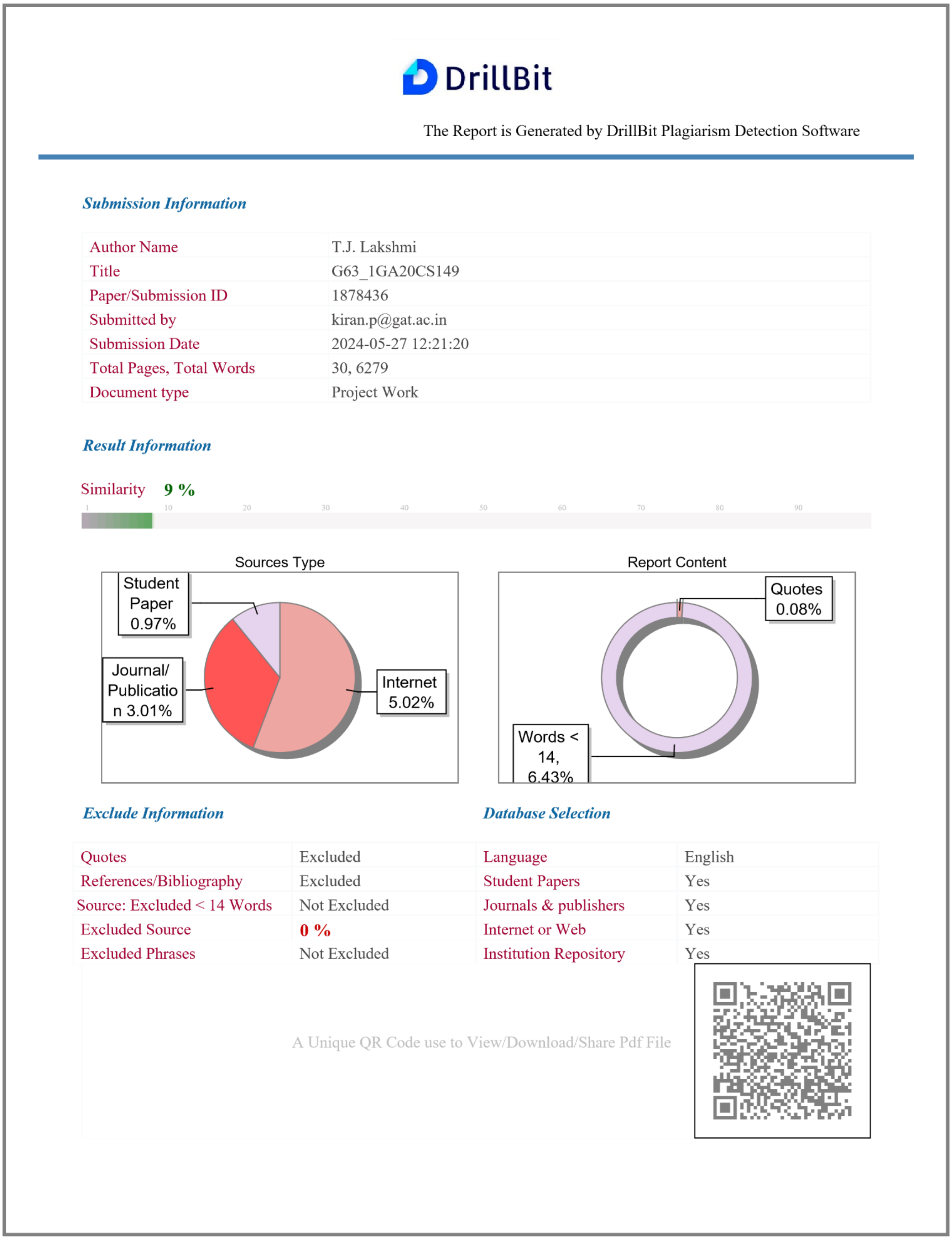
[8] Q. Zhou, H. Huang, Z. Zheng and J. Bian, "Solutions to Scalability of Blockchain: A Survey," in IEEE Access, vol. 8, pp. 16440-16455, 2020, doi: 10.1109/ACCESS.2020.2967218.

[9] V. T. Truong, L. Le and D. Niyato, "Blockchain Meets Metaverse and Digital Asset Management: A Comprehensive Survey," in IEEE Access, vol. 11, pp. 26258-26288, 2023, doi: 10.1109/ACCESS.2023.3257029.

[10] O. Ali, A. Jaradat, A. Kulakli and A. Abuhalimeh, "A Comparative Study: Blockchain Technology Utilization Benefits, Challenges and Functionalities," in IEEEAccess, vol. 9, pp. 12730-12749, 2021, doi: 10.1109/ACCESS.2021.3050241.

APPENDIX A

PLAGIARISM REPORT



DrillBit

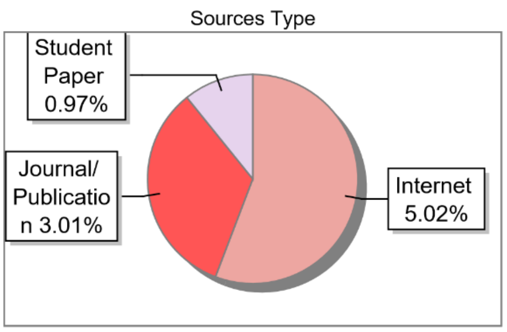
The Report is Generated by DrillBit Plagiarism Detection Software

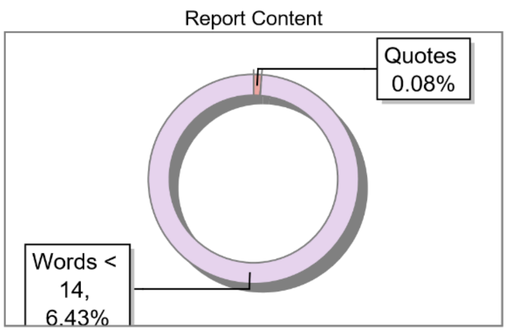
Submission Information

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 %**





Exclude Information

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

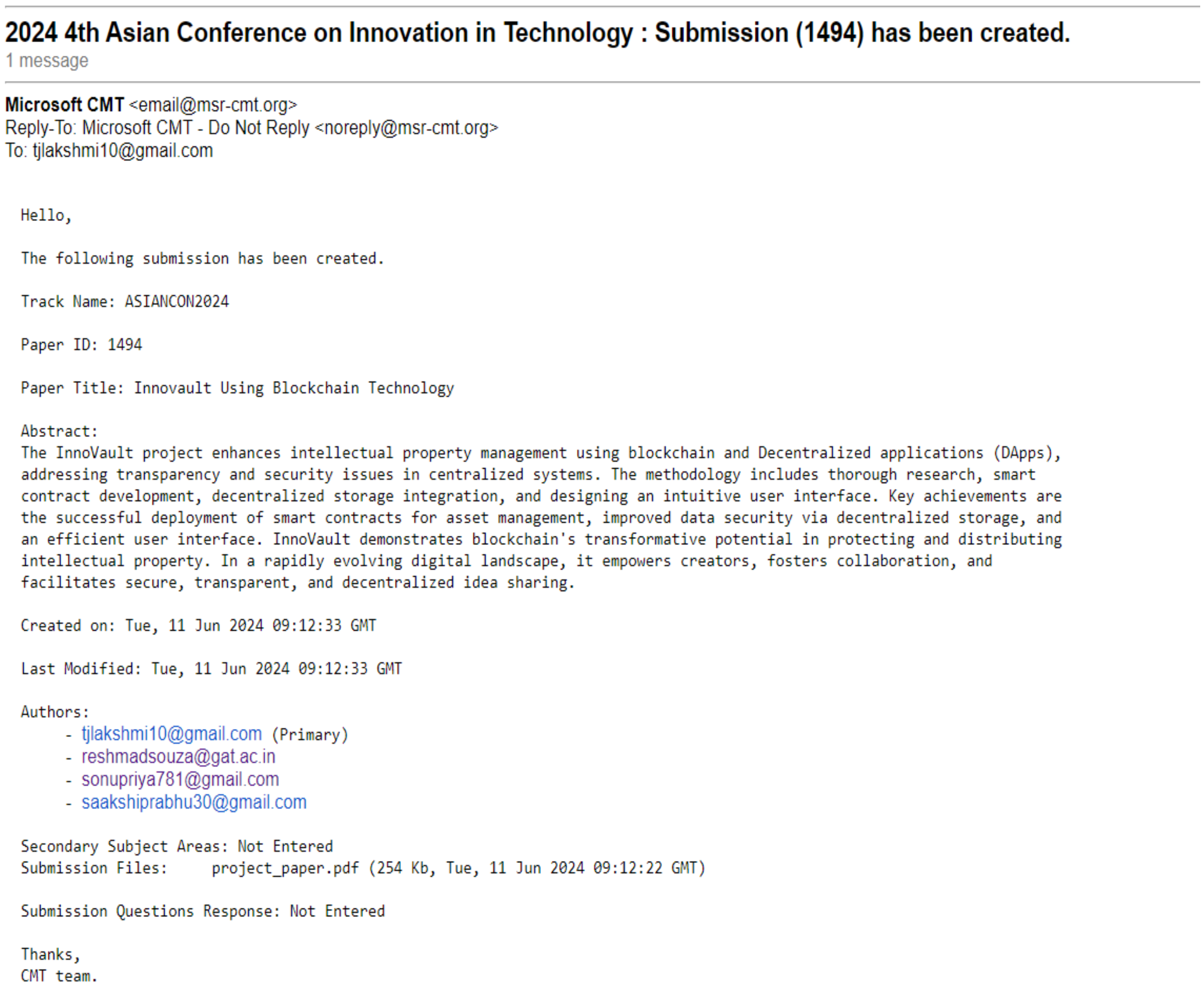
Database Selection

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

APPENDIX B

PAPER SUBMISSION



2024 4th Asian Conference on Innovation in Technology : Submission (1494) has been created.
1 message

Microsoft CMT <email@msr-cmt.org>
Reply-To: Microsoft CMT - Do Not Reply <noreply@msr-cmt.org>
To: tjlakshmi10@gmail.com

Hello,

The following submission has been created.

Track Name: ASIANCON2024

Paper ID: 1494

Paper Title: Innovault Using Blockchain Technology

Abstract:
The InnoVault project enhances intellectual property management using blockchain and Decentralized applications (DApps), addressing transparency and security issues in centralized systems. The methodology includes thorough research, smart contract development, decentralized storage integration, and designing an intuitive user interface. Key achievements are the successful deployment of smart contracts for asset management, improved data security via decentralized storage, and an efficient user interface. InnoVault demonstrates blockchain's transformative potential in protecting and distributing intellectual property. In a rapidly evolving digital landscape, it empowers creators, fosters collaboration, and facilitates secure, transparent, and decentralized idea sharing.

Created on: Tue, 11 Jun 2024 09:12:33 GMT

Last Modified: Tue, 11 Jun 2024 09:12:33 GMT

Authors:
- tjlakshmi10@gmail.com (Primary)
- reshmadsouza@gat.ac.in
- sonupriya781@gmail.com
- saakshiprabhu30@gmail.com

Secondary Subject Areas: Not Entered
Submission Files: project_paper.pdf (254 Kb, Tue, 11 Jun 2024 09:12:22 GMT)

Submission Questions Response: Not Entered

Thanks,
CMT team.