INSTITUTE ANNUAL REPORT MANAGEMENT SYSTEM USING BLOCK CHAIN

PRODUCT DEVELOPMENT LAB-IV

Submitted by

SATHIYAMOORTHI K (421123108047)
&
SARAVANAN R (421123108044)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

IFET COLLEGE OF ENGINEERING

(An Autonomous Institution)

Approved by AICTE, New Delhi and Accredited by NAAC & NBA

Affiliated to Anna University, Chennai-25

Gangarampalayam, Villupuram – 605 108

MAY 2025

IFET COLLEGE OF ENGINEERING
(An Autonomous Institution)

BONAFIDE CERTIFICATE

Certified that this report titled **INSTITUTE ANNUAL REPORT MANAGEMENT SYSTEM USING BLOCK CHAIN is** the Bona-fide work **SATHIYAMOORTHI K (421123108047) & SARAVANAN R (421123108044)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Mr. M. ARUNKUMAR M.E.,
SUPERVISOR,
ASSISTANT PROFESSOR,
DEPARTMENT OF IT,
IFET COLLEGE OF ENGINEERING,

SIGNATURE
Dr. R. THENDRAL,
HEAD OF THE DEPARTMENT,
SENIOR ASSISTANT PROFESSOR,
DEPARTMENT OF IT,
IFET COLLEGE OF ENGINEERING,

CERTIFICATE OF EVALUATION

College name	:	IFET College of Engineering, Villupuram.
Branch	:	B.Tech - IT
Month &Year	:	May 2025
Sub. Code & Name	:	23PL4004 - PRODUCT DEVELOPMENT LAB-IV

Name of the Student	Register Number	Title of the Project	Name of the Supervisor with Designation
SATHIYAMOORTHI K **SARAVANAN R**	**421123108047** **421123108044**	**INSTITUTE ANNUAL REPORT MANAGEMENT SYSTEM USING BLOCK CHAIN**	**Mr. M. ARUNKUMAR** Assistant Professor

The report for the Product Development Lab - IV submitted for the fulfillment of the award of the degree of Bachelor of Technology in Information Technology of IFET College of Engineering (Autonomous), permanently affiliated to Anna University was evaluated and confirmed to be the work done by the above student.

SUPERVISOR HEAD OF THE DEPARTMENT

Submitted for the End Semester examination held on ____________________

ACKNOWLEDGEMENT

We thank the almighty, for the blessings that have been showered upon me to bring forth the success of the project. I would like to express my sincere gratitude to our Chairman **Mr. K. V. RAJA**, our Secretary **Mr. K. SHIVRAM ALVA** and our Treasurer **Mr. R. VIMAL** for providing us with an excellent infrastructure and necessary resources to carry out this project and we extend our gratitude to our Principal **Dr. G. MAHENDRAN,** for his constant support to our work.

I also take this opportunity to express my sincere thanks to our Dean Academics **Dr. P. KANIMOZHI** who has provided all the needful help in executing the project successfully.

I wish to express our thanks to our Head of the Department, **Dr.R.THENDRAL** Senior Assistant Professor for her persistent encouragement and support to complete this project. We express our heartfelt gratitude to our guide **Mr. M. ARUNKUMAR,** Assistant Professor for his priceless guidance and motivation which helped us to bring this project to a perfect shape.

And I thank our Batch Coordinator **Mr. M. ARUNKUMAR,** Assistant Professor, Department of Information Technology who encouraged us in each and every step of this project to complete it successfully.

I also thank our lab technicians and all the staff members of our department for their support and assistance.

Last but not the least, I whole heartedly thanks to our lovely parents and friends for their moral support in tough times and their constructive criticism which made us to succeed in my work.

ABSTRACT

The Smart Data Integration System for the Annual Report Portal is a blockchain driven solution designed to revolutionize institutional reporting through secure, transparent and automated data management. Leveraging a hybrid blockchain architecture, the system ensures that all submitted data is verifiable, tamper-proof and permanently recorded, thereby enhancing trust and integrity. Smart contracts are employed to automate the validation and verification of reports, significantly reducing manual effort and ensuring compliance with institutional standards. A robust frontend and backend infrastructure enables seamless user interaction, efficient data submission, and dynamic report generation. The platform also implements Role-Based Access Control (RBAC) to safeguard sensitive information and enforce user-specific permissions, ensuring that only authorized personnel can access or modify particular datasets. Immutable audit trails provide complete traceability and accountability for all actions performed within the system. By integrating blockchain with modern web technologies, the project addresses common challenges in traditional reporting systems such as data tampering, unauthorized access, and manual errors. This project aims to streamline the annual report compilation process, increase operational efficiency and foster data-driven decision-making within institutions. The system supports real-time analytics, enabling institutions to derive actionable insights for strategic planning. Its modular design facilitates scalability, supporting future enhancements like automated notifications. On Addition, robust error-handling ensures reliable performance during peak reporting periods.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

RBAC	**-**	**R**ole-**B**ased **A**ccess **C**ontrol
AMD	-	**A**dvanced **M**icro **D**evices
RAM	-	**R**andom **A**ccess **M**emory
SSD	-	**S**olid-**S**tate **D**rive
API	-	**A**pplication **P**rogramming **I**nterface
UI	**-**	**U**ser **I**nterface
SHA-256	-	**S**ecure Hash **A**lgorithm **256**
JSON	-	**J**avaScript **O**bject **N**otation
HTML	-	**H**yper **T**ext **M**arkup **L**anguage
CSS	-	**C**ascading **S**tyle **S**heets

CHAPTER-1

INTRODUCTION

1.1 General

In the digital age, institutions increasingly rely on accurate and timely data for effective decision-making and compliance. Annual reporting plays a crucial role in showcasing an institution's achievements, performance metrics and strategic goals. Traditional methods of compiling annual reports often suffer from issues like data inconsistencies, unauthorized access, manual errors and lack of transparency. To address these challenges, the Smart Data Integration System for the Annual Report Portal introduces a modern, technology-driven approach that ensures secure, efficient and verifiable data management. The system leverages blockchain technology, specifically a hybrid blockchain architecture, to store and validate data in a tamper-proof and transparent manner. Through the use of smart contracts, it automates the validation process and enforces data submission rules, significantly reducing manual oversight. Role-based access control mechanism ensures that users interact with the system according to their designated permissions, while immutable audit trails maintain traceability and accountability of all activities.

The integrated system streamlines the institutional reporting process, strengthens data governance and builds trust in the authenticity and accuracy of the generated reports. The system incorporates a user-friendly interface, enabling seamless interaction and dynamic report generation across various institutional roles. It also supports scalability, allowing future integration of advanced features like predictive analytics to enhance strategic decision-making. The system ensures real-time data synchronization, enabling institutions to access up-to-date reports for timely decision-making. Its robust error-handling mechanisms further enhance reliability.

1.2 Domain Review

The project falls under the intersection of Blockchain Technology, Data Management Systems and Institutional Reporting Automation. Traditional data integration systems used for compiling institutional reports often face issues like inconsistent data formats, lack of transparency, risk of tampering and inefficient validation processes. This project addresses these issues by adopting cutting-edge solutions rooted in the blockchain domain. Blockchain, known for its decentralized, transparent and immutable nature, provides a secure foundation for data storage and validation. The hybrid blockchain architecture used in this system combines the benefits of both public and private blockchains, ensuring scalability, privacy and data integrity. Smart contracts enable the automated execution of report verification and guideline enforcement, minimizing human error and manual interventions. By integrating role-based access control (RBAC), the system further secures access to sensitive information and ensures only authorized users can perform critical operations. The audit trail functionality ensures that every action on the platform is traceable, thus enhancing accountability.

The domain leverages advanced principles from cybersecurity, distributed ledger technologies and automation to deliver a reliable solution for institutional data integration and report management. The system incorporates modern web technologies to facilitate seamless user interactions and dynamic report generation. Its modular architecture supports future integration with emerging technologies like artificial intelligence for advanced data analytics. This approach not only streamlines reporting processes but also aligns with global standards for secure and efficient data management. The project sets a precedent for leveraging blockchain in institutional settings, paving the way for broader adoption in data-driven governance.

CHAPTER-2

LITERATURE SURVEY

2.1 REVIEW OF LITERATURE

2.1.1 Wang, X., Li, Y., & Zhang, Z., "Blockchain-Based Data Integrity Verification for Secure Applications", 2023

This paper proposes a hybrid blockchain model to ensure secure data integration and integrity verification for industrial applications. It focuses on real-time data validation using blockchain's immutable ledger properties. The model enhances security by detecting tampering attempts instantly. However, it also highlights challenges like high computational costs and latency in real-time processing. The approach is suitable for scenarios where data accuracy and trustworthiness are critical. This model aligns with the Smart Data Integration System's goals, providing a robust framework for tamper-proof institutional reporting.

2.1.2 Sharma, A., & Jain, P. K., "Role-Based Access Control with Blockchain for Secured Data Sharing," 2022

This study integrates Role-Based Access Control (RBAC) with blockchain technology to provide secure data sharing mechanisms. It restricts unauthorized access through granular permission settings recorded immutably on the blockchain. The approach strengthens security and transparency in multi-user environments. Challenges include managing complex role hierarchies and permissions efficiently. This framework is ideal for environments demanding strict access control and auditability. The study also demonstrates improved performance in access control enforcement through blockchain's decentralized architecture. Its findings are particularly applicable to institutional systems like the Smart Data Integration System, enhancing secure report management.

2.1.3 Patel, S., Shah, K., & Thakkar, R., "Hybrid Blockchain for Secure and Scalable Data Management," 2020.

The authors introduce a hybrid blockchain architecture that balances security with scalability in data management systems. It combines private and public blockchains to optimize transaction efficiency and secure data storage. The model aims to mitigate limitations of conventional blockchains like slow processing and high costs. Despite its benefits, the implementation complexity and maintenance pose significant challenges. The solution targets organizations needing secure yet scalable data solutions.

2.1.4 Chen, J., Xu, L., & Zhang, Q., "Blockchain-Based Audit Transparent Data Management," 2020.

This paper discusses methods to improve data integration in universities by leveraging smart technologies. It highlights how combining data from multiple departments enhances reporting accuracy and decision-making. The authors advocate for automation in data collection and validation processes to reduce errors. The study also emphasizes user-friendly interfaces to facilitate seamless data interactions. This approach supports institutions in maintaining comprehensive, reliable annual reports.

2.1.4 Johnson, R., & Kim, L. J., "Smart Data Integration for Universities," 2019.

The research focuses on using blockchain to maintain secure and immutable audit trails for transparent data management. It ensures that every action is traceable and tamper-proof, improving trustworthiness in data handling. The system addresses transparency requirements in regulatory and institutional reporting contexts. Scalability remains a concern due to the growing volume of audit data. The framework is suitable for environments where accountability and traceability are paramount.

2.2 EXISTING SYSTEM

Manual report submission and review systems often suffer from errors, delays and risks of data tampering. Traditional centralized database systems store reports in a single location, making them vulnerable to unauthorized modifications. File-based storage with version control attempts to manage changes but lacks automated data validation and consistent version tracking. Cloud-based report portals offer accessibility but do not fully guarantee data integrity and are susceptible to insider threats. Digital report management platforms with basic security features have limited auditability and fail to provide complete traceability of records.

2.3 DISADVANTAGES OF EXISTING SYSTEM

- **Error-Prone and Slow,** Manual processes introduce human errors and slow down report submission and review cycles.
- **Centralized Vulnerabilities,** Centralized databases increase risk by creating single points of failure and easy targets for tampering.
- **Inadequate Validation,** Absence of automated validation results in inconsistent and unreliable data across versions.
- **Security Gaps,** Insider threats and unauthorized modifications remain significant risks due to insufficient protection measures.
- **Poor Auditability and Traceability,** Limited or no comprehensive audit trails hinder accountability and make it difficult to track data history accurately.
- **Complex Role Management,** managing user permissions and roles in conventional systems is often complicated and error-prone, increasing the risk of unauthorized access.

CHAPTER-3

PROPOSED SYSTEM

3.1 SYSTEM REQUIREMENT

3.1.1 Software Requirements

- Operating System (Windows/Linux/macOS)
- Python 2.9
- Flask Framework (Flask == 2.0.1, Flask-Login == 0.5.0)
- Node.js and npm
- Gunicorn == 20.1.0, Werkzeug == 2.0.3

3.1.2 Hardware Requirements

- Processor: Intel i5 or equivalent AMD processor
- RAM: Minimum 8 GB (Recommended 16 GB)
- Storage: At least 256 GB SSD
- Operating System: Windows 10/11 or Linux (Ubuntu 18.04 or later)

3.2 SYSTEM ARCHITECTURE

The Smart Data Integration Portal follows a layered architecture to ensure secure, transparent academic record management. The Frontend UI Module offers role-based dashboards for users to interact with the system through templates like homepage.html. The Backend API Gateway (app.py) handles routing, authentication and role-based access control. The Smart Contracts Layer (utils.py) performs automated validation of submitted data, which is then immutably stored in the Hybrid Blockchain Network (blockchain.py) using files like annual_reports.json. The Audit and Logging Module (audit_log.py) captures all user actions in audit_logs.json, ensuring traceability, integrity and accountability.

3.2.1 SYSTEM ARCHITECTURE

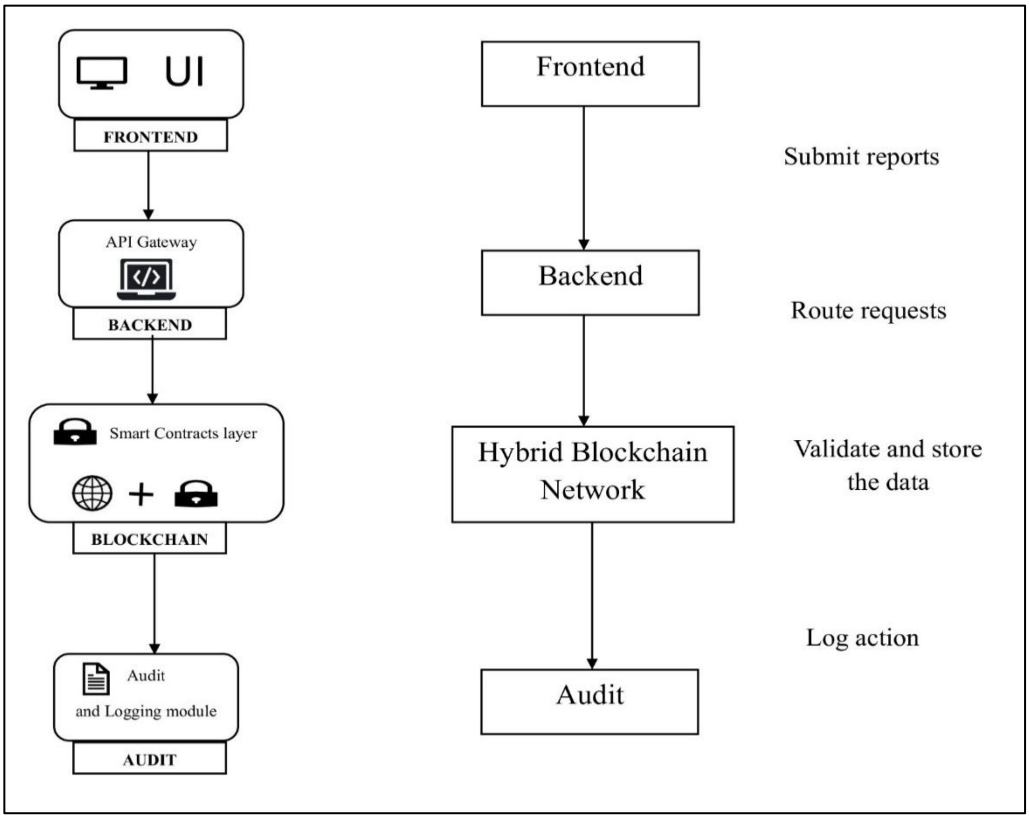


Fig.3.2. System Architecture

COMPONENTS OF THE ARCHITECTURE

1. Frontend (UI Module)

The Frontend of the Smart Data Integration System comprises templates such as homepage.html, dashboard_student.html, dashboard_faculty.html, dashboard_admin.html, dashboard_developer.html and styles.css, rendering role specific interfaces to capture user inputs like login credentials and report submissions, while displaying data such as annual reports and blockchain records through intuitive dashboards. This module provides an accessible and responsive platform, ensuring a seamless user experience across diverse roles including Students, Faculty, Admins and Developers. The frontend incorporates real-time feedback mechanisms, such as form validation alerts, to enhance user interaction efficiency. It leverages modern web design principles, including a blue-grey color scheme, to ensure visual consistency and accessibility across devices. The module supports dynamic content updates, enabling users to view the latest report data without requiring page reloads.

2. Backend (API Gateway)

The Backend (API Gateway) module of the Smart Data Integration System, consisting of app.py and users.json, processes API requests from the frontend, manages user authentication through Flask-Login and enforces access control using Role-Based Access Control (RBAC), ensuring that only authorized users can perform operations like report submission. It serves as a secure bridge between the frontend and other modules, facilitating seamless and authorized data transfer across the system. The backend implements caching mechanisms to optimize response times for frequently accessed data, such as dashboard reports. It also includes robust error-handling capabilities to maintain system reliability during high-traffic periods, such as annual report submission deadlines.

3. Hybrid Blockchain Network

The Hybrid Blockchain Network module of the Smart Data Integration System, comprising blockchain.py, annual_reports.json, attack_results.json and snapshots.json, stores validated academic reports in an immutable ledger using a SHA-256 hash chain, while offering tools like validate_chain for validation and simulate_attack for tamper analysis. This module ensures data integrity, tamper resistance and auditability, maintaining a secure and verifiable record of academic reports for institutional use. It supports periodic snapshots of the blockchain state, enabling efficient historical data analysis and compliance checks.

4. Audit and Logging Module

The Audit and Logging Module of the Smart Data Integration System, consisting of audit_log.py and audit_logs.json, records system activities such as logins and report submissions with timestamps and user metadata, securely storing them in a file-based log. This module ensures transparency and accountability by maintaining immutable audit trails, enabling comprehensive monitoring and traceability of all user actions within the platform.

3.3 MODULES

1. User Interface (Frontend)
2. Backend API & Smart Contract
3. Hybrid Blockchain & Validation
4. Audit & Logging Module

Module 1: User Interface (Frontend)

The User Interface (Frontend) Module of the Smart Data Integration System aims to provide a user-friendly interface for login, report submission and specific dashboards for each role, utilizing components such as base.html, homepage.html, dashboard_roles.html and styles.css, with a workflow where users first access homepage.html via the '/' route to submit login credentials, then proceed to the dashboard route which renders the appropriate role specific dashboard, such as the Faculty dashboard displaying annual reports, allowing users to interact with forms for submitting reports and viewing data like blockchain records.

Module 2: Backend API & Smart Contract

The Backend API & Smart Contract Module of the Smart Data Integration System is designed to handle user authentication and validate and process reports using smart contract logic, incorporating components such as app.py, utils.py, users.json and annual_reports.json, with a workflow where a user submits a request, such as logging in via the /login route or submitting a report through /submit_annual_report, followed by the backend validating inputs using mechanisms like validate_annual_report in utils.py to check report data for compliance before processing and storing it in annual_reports.json

Module 3: Hybrid Blockchain & Validation

The Hybrid Blockchain & Validation Module of the Smart Data Integration System is mainly designed to store submitted reports as tamper-proof blocks and ensure integrity using blockchain principles, utilizing components such as blockchain.py, in-memory chains, annual_reports.json, attack_results.json and snapshots.json, with a workflow where a validated report from the Smart Contract is added as a block using add_annual_report, followed by validate_chain checking hashes to ensure no tampering, while Developer tools like simulate_attack and analyze_snapshot analyze the chain and store results in attack_results.json and snapshots.json for further integrity verification.

Module 4: Audit & Logging Module

The Audit & Logging Module of the Smart Data Integration System comprises the detailed logs of user actions for traceability and security auditing, utilizing components such as audit logging scripts and audit log storage, with a workflow where a user action, such as a login or report submission, triggers the creation of a log entry containing details like a timestamp, user identifier (e.g., "faculty1") and action type (e.g., "login"), which is then saved to the audit log storage, allowing Admin and Developer dashboards to retrieve and display these logs for monitoring purposes.

Overall System Objectives:

The Smart Data Integration System aims to revolutionize institutional reporting by achieving secure record storage through a hybrid blockchain that ensures academic records are tamper-proof and verifiable, providing role-based access with tailored dashboards for Students, Faculty, Admins and Developers to facilitate user specific interactions, ensuring data integrity through smart contract validation that enforces compliance and accuracy and enabling action tracking by logging all user actions to maintain transparency and auditability across the platform

3.4 ADVANTAGES OF PROPOSED SYSTEM

1. **Enhanced Security and Integrity**

The **Smart Data Integration System** enhances security and integrity using hashing (SHA256) to link blockchain blocks, detecting tampering through broken hash chains, with validate_chain() verifying integrity and simulate_attack() testing resilience, ensuring trustworthy academic records.

2. **Controlled and Secure Access**

The system ensures controlled access via Role-Based Access Control (RBAC), limiting actions to authorized roles like Faculty and Admins, with Flask-Login securing authentication and restricted Student access preventing misuse, safeguarding sensitive data.

3. Automated and Reliable Data Validation

The system automates data validation with smart contracts enforcing rules on data length and format, rejecting invalid inputs like incorrect dates and filtering only valid data for blockchain storage, ensuring high-quality academic reports.

4. Transparency and Accountability

The system fosters transparency by logging all activities with timestamps in secure audit storage, allowing Admins to track user actions, with immutable audit trails ensuring trustworthy records and accountability.

5. Efficient User Experience

The system offers an efficient user experience with a responsive, mobile friendly UI, seamless backend integration for instant feedback on actions like report submissions and role-specific dashboards enhancing productivity.

6. Scalable and Future-Proof Design

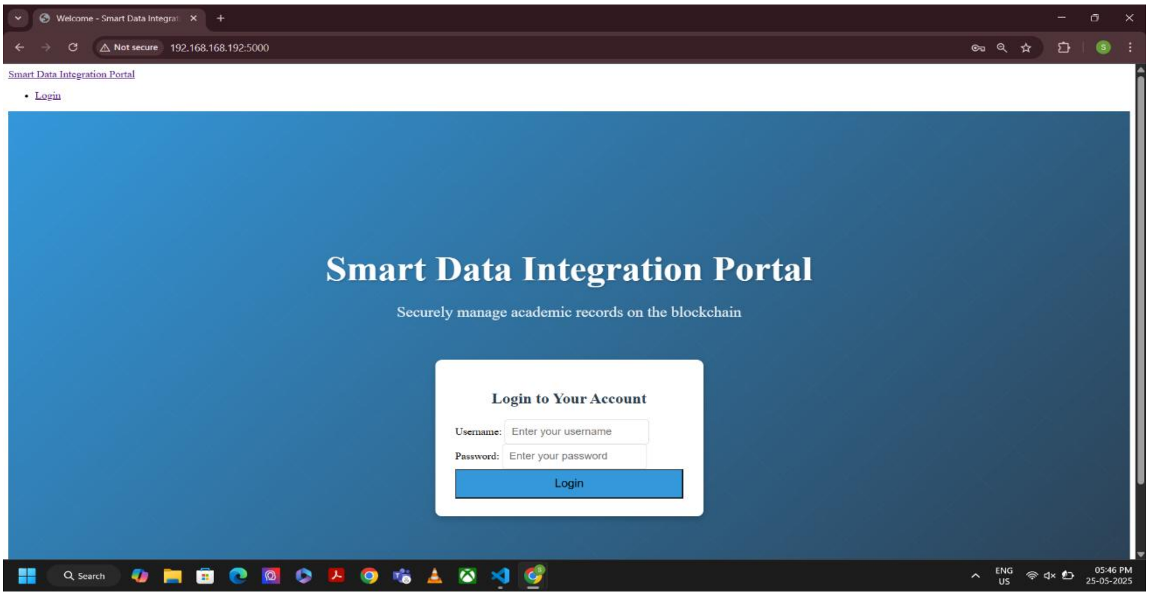
The system features a scalable and future-proof design with a modular architecture supporting easy integration of new features like analytics, efficient data handling through file-based storage expandable to databases and compatibility with emerging technologies, ensuring long-term adaptability for institutional needs.

7. Real-Time Data Insights

The system enables real-time data insights by processing analytics on dashboard displays, supporting instant report queries for timely decisions and leveraging blockchain data for rapid access, empowering institutions with actionable intelligence.

CHAPTER -4

RESULT AND DISCUSSIONS



Smart Data Integration Portal

- Login

Smart Data Integration Portal

Securely manage academic records on the blockchain

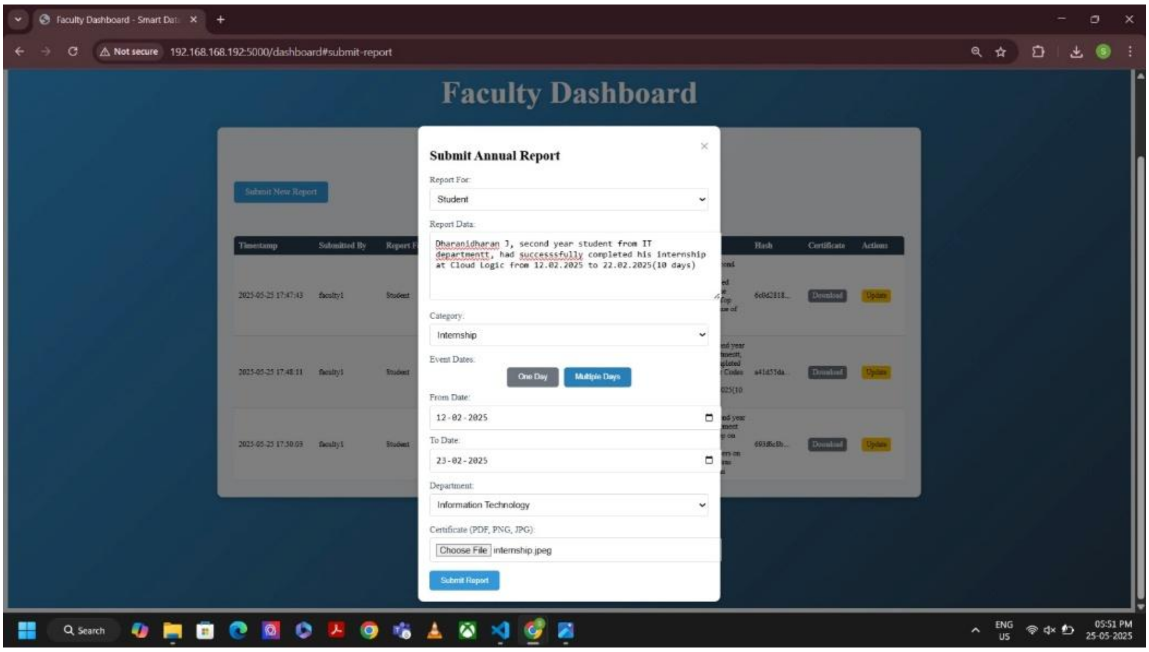
Login to Your Account

Username: Enter your username

Password: Enter your password

Login

Fig.4.1. Login Page



Faculty Dashboard - Smart Dat
Not secure
192.168.168.192:5000/dashboard#submit-report
05:51 PM
25-05-2025

Faculty Dashboard

Submit New Report

Timestamp	Submitted By	Report F	Hash	Certificate	Actions
2025-05-25 17:47:43	faculty1	Student	6c0d2818...	Download	Update
2025-05-25 17:48:11	faculty1	Student	a41d55da...	Download	Update
2025-05-25 17:50:03	faculty1	Student	693d6c8b...	Download	Update

Submit Annual Report

Report For: Student

Report Data:

```
Dharanidharan J, second year student from IT
departmentt, had successsfully completed his internship
at Cloud Logic from 12.02.2025 to 22.02.2025(10 days)
```

Category: Internship

Event Dates: One Day ● Multiple Days

From Date: 12-02-2025

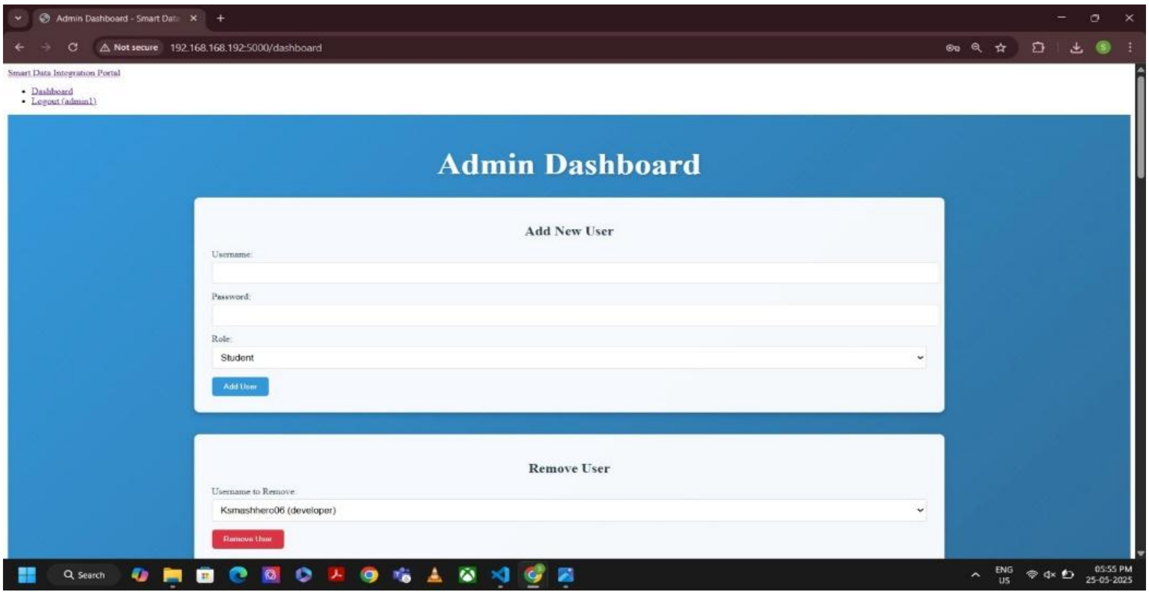
To Date: 23-02-2025

Department: Information Technology

Certificate (PDF, PNG, JPG): Choose File internship.jpeg

Submit Report

Fig.4.2. Annual Report Submission



Admin Dashboard - Smart Data

Not secure 192.168.168.192:5000/dashboard

Smart Data Integration Portal

- Dashboard
- Logout (admin1)

Admin Dashboard

Add New User

Username:

Password:

Role: Student

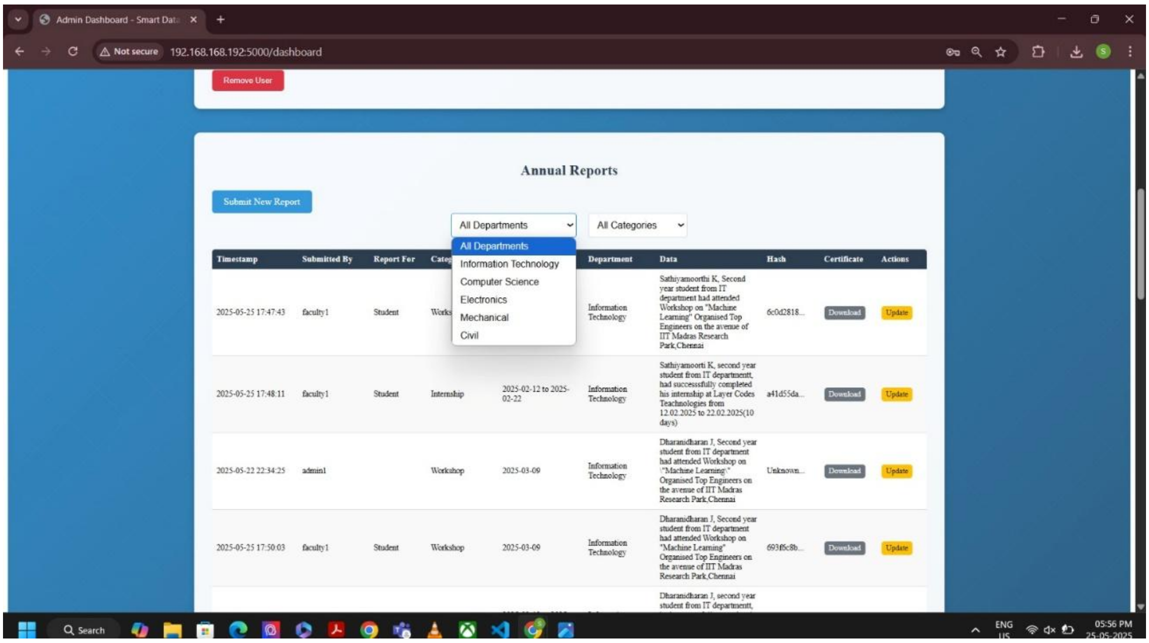
Add User

Remove User

Username to Remove: Ksmashhero06 (developer)

Remove User

Fig.4.3. Admin Dashboard



Admin Dashboard - Smart Data

Not secure 192.168.168.192:5000/dashboard

Remove User

Annual Reports

Submit New Report

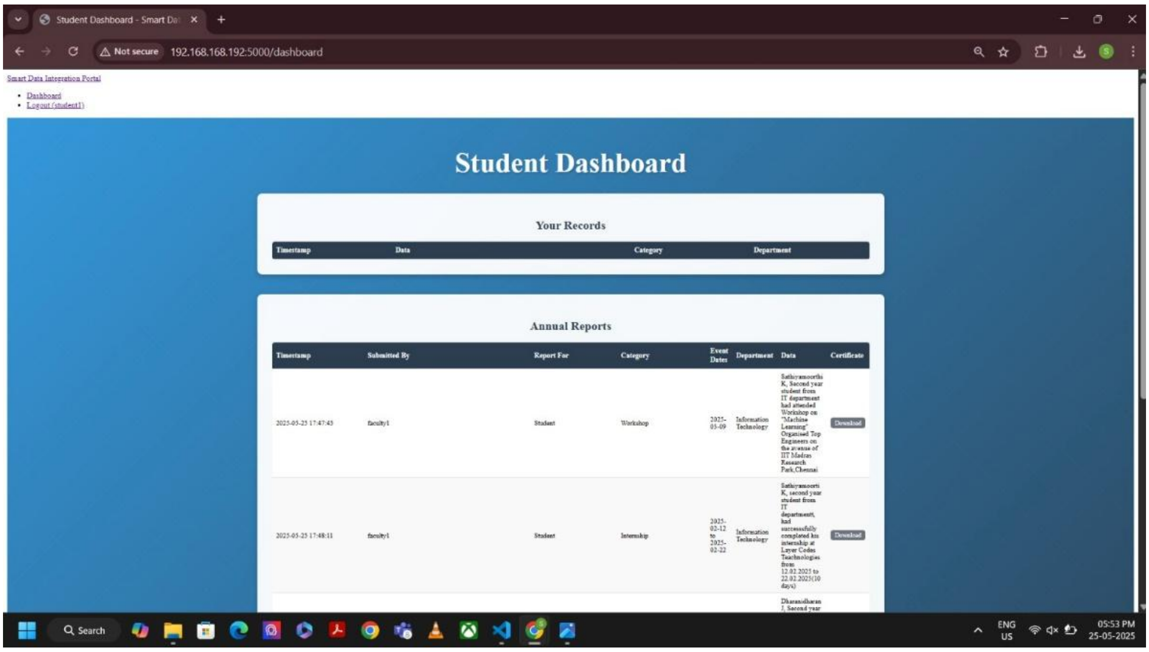
All Departments

- All Departments
- Information Technology
- Computer Science
- Electronics
- Mechanical
- Civil

All Categories

Timestamp	Submitted By	Report For	Categ		Department	Data	Hash	Certificate	Actions
2025-05-25 17:47:43	faculty1	Student	Works		Information Technology	Sathiyamoorthi K, Second year student from IT department had attended Workshop on "Machine Learning" Organised Top Engineers on the avenue of IIT Madras Research Park,Chennai	6c0d2818...	Download	Update
2025-05-25 17:48:11	faculty1	Student	Internship	2025-02-12 to 2025-02-22	Information Technology	Sathiyamoorti K, second year student from IT departmentt, had successsfully completed his internship at Layer Codes Teachnologies from 12.02.2025 to 22.02.2025(10 days)	a41d55da...	Download	Update
2025-05-22 22:34:25	admin1		Workshop	2025-03-09	Information Technology	Dharanidharan J, Second year student from IT department had attended Workshop on \"Machine Learning\" Organised Top Engineers on the avenue of IIT Madras Research Park,Chennai	Unknown...	Download	Update
2025-05-25 17:50:03	faculty1	Student	Workshop	2025-03-09	Information Technology	Dharanidharan J, Second year student from IT department had attended Workshop on "Machine Learning" Organised Top Engineers on the avenue of IIT Madras Research Park,Chennai	693f6c8b...	Download	Update
						Dharanidharan J, second year student from IT departmentt,			

Fig.4.4. Annual Reports



Smart Data Integration Portal

- Dashboard
- Logout (student1)

Student Dashboard

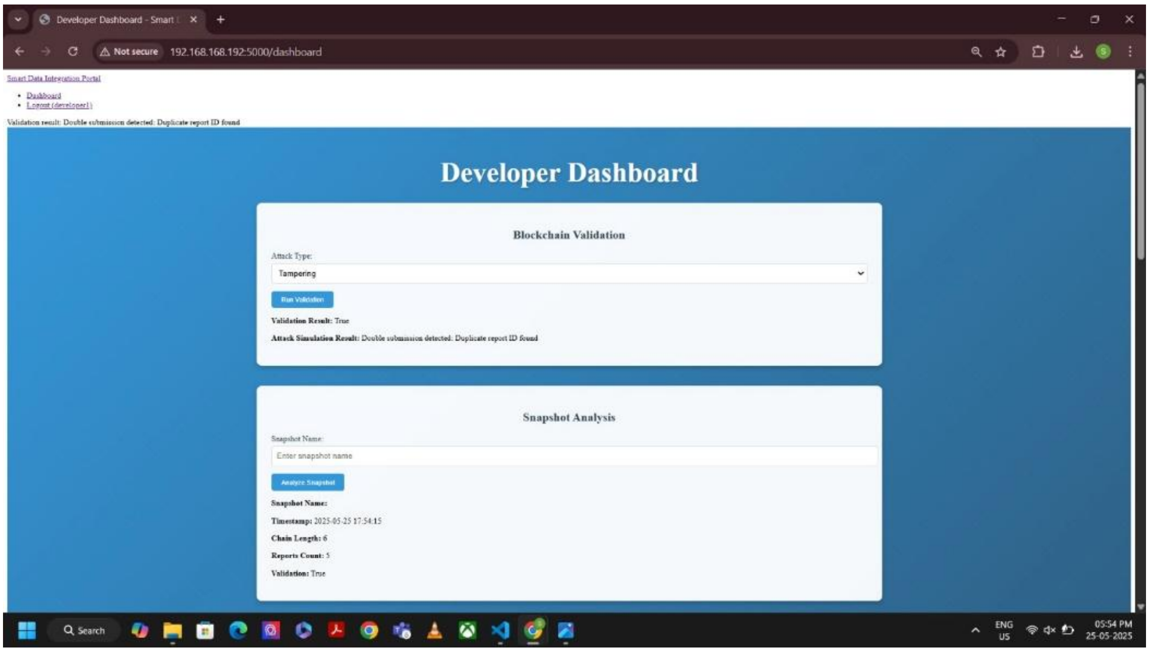
Your Records

Timestamp	Data	Category	Department

Annual Reports

Timestamp	Submitted By	Report For	Category	Event Dates	Department	Data	Certificate
2025-05-25 17:47:45	faculty1	Student	Workshop	2025-05-09	Information Technology	Sathiyamoorthi K, Second year student from IT department had attended Workshop on "Machine Learning" Organised Top Engineers on the avenue of IIT Madras Research Park,Chennai	Download
2025-05-25 17:48:11	faculty1	Student	Internship	2025-02-12 to 2025-02-22	Information Technology	Sathiyamoorti K, second year student from IT department, had successfully completed his internship at Layer Codes Teachnologies from 12.02.2025 to 22.02.2025(10 days)	Download
						Dharanidharan J, Second year	

Fig.4.5. Student Dashboard



Smart Data Integration Portal

- Dashboard
- Logout (developer1)

Validation result: Double submission detected: Duplicate report ID found

Developer Dashboard

Blockchain Validation

Attack Type: Tampering

Validation Result: True

Attack Simulation Result: Double submission detected: Duplicate report ID found

Snapshot Analysis

Snapshot Name: Enter snapshot name

Snapshot Name:

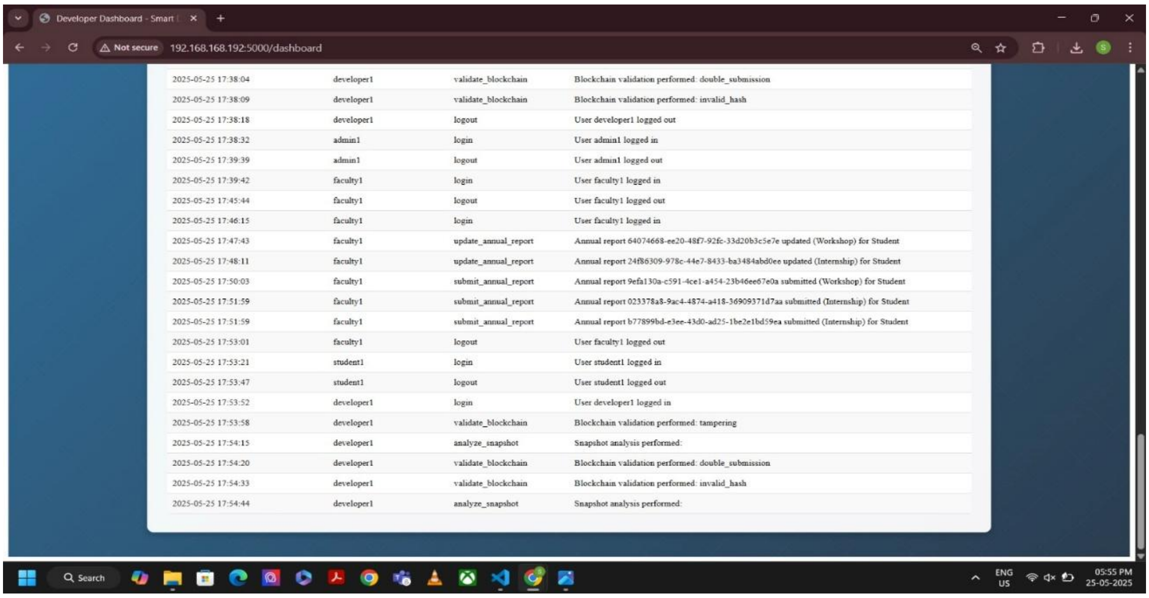
Timestamp: 2025-05-25 17:54:15

Chain Length: 6

Reports Count: 5

Validation: True

Fig.4.6. Developer Dashboard



2025-05-25 17:38:04	developer1	validate_blockchain	Blockchain validation performed: double_submission
2025-05-25 17:38:09	developer1	validate_blockchain	Blockchain validation performed: invalid_hash
2025-05-25 17:38:18	developer1	logout	User developer1 logged out
2025-05-25 17:38:32	admin1	login	User admin1 logged in
2025-05-25 17:39:39	admin1	logout	User admin1 logged out
2025-05-25 17:39:42	faculty1	login	User faculty1 logged in
2025-05-25 17:45:44	faculty1	logout	User faculty1 logged out
2025-05-25 17:46:15	faculty1	login	User faculty1 logged in
2025-05-25 17:47:43	faculty1	update_annual_report	Annual report 64074668-ee20-48f7-92fc-33d20b3c5e7e updated (Workshop) for Student
2025-05-25 17:48:11	faculty1	update_annual_report	Annual report 24f86309-978c-44e7-8433-ba3484abd0ee updated (Internship) for Student
2025-05-25 17:50:03	faculty1	submit_annual_report	Annual report 9efa130a-c591-4ce1-a454-23b46ee67e0a submitted (Workshop) for Student
2025-05-25 17:51:59	faculty1	submit_annual_report	Annual report 023378a8-9ac4-4874-a418-36909371d7aa submitted (Internship) for Student
2025-05-25 17:51:59	faculty1	submit_annual_report	Annual report b77899bd-e3ee-43d0-ad25-1be2e1bd59ea submitted (Internship) for Student
2025-05-25 17:53:01	faculty1	logout	User faculty1 logged out
2025-05-25 17:53:21	student1	login	User student1 logged in
2025-05-25 17:53:47	student1	logout	User student1 logged out
2025-05-25 17:53:52	developer1	login	User developer1 logged in
2025-05-25 17:53:58	developer1	validate_blockchain	Blockchain validation performed: tampering
2025-05-25 17:54:15	developer1	analyze_snapshot	Snapshot analysis performed:
2025-05-25 17:54:20	developer1	validate_blockchain	Blockchain validation performed: double_submission
2025-05-25 17:54:33	developer1	validate_blockchain	Blockchain validation performed: invalid_hash
2025-05-25 17:54:44	developer1	analyze_snapshot	Snapshot analysis performed:

Fig.4.7. Audit Logs (Attack Simulation Included)

The system, built with Flask and blockchain technology, ensures secure, tamper proof storage and management of annual reports. Screenshots showcase key interfaces: the Login Page for secure user authentication; the Report Submission Page for faculty/admin to submit reports with certificate uploads; the Admin Dashboard featuring add/remove user and report submission functionalities; the Annual Reports Page displaying all reports with filters; the Student Dashboard showing relevant reports and records; the Developer Dashboard with blockchain validation and snapshot analysis tools; and the Audit Log Page tracking user actions for transparency. These interfaces demonstrate role-based access, blockchain integrity and user friendly design, validated through successful report submissions and attack simulations (e.g., tampering detected).

4. CONCLUSION & FUTURE WORK

4.1 CONCLUSION

The system’s The Smart Data Integration Portal, provides a secure and efficient platform for academic record management. Its Hybrid Blockchain Architecture ensures tamper-proof and verifiable report storage, while Role Based Access Control (RBAC) restricts access to authorized users, enhancing security. Smart Contracts automate data validation, enforcing guidelines and minimizing errors and Immutable Audit Trails maintain transparency through traceable action logs.

The system’s Seamless Frontend and Backend Integration delivers an intuitive user experience, enabling smooth data submission and report generation. With a modular design User Interface, Backend API, Blockchain and Audit Logging modules which ensures scalability and maintainability. The portal effectively addresses academic record management challenges, offering a reliable, blockchain-based solution. It lays a strong foundation for future enhancements, such as advanced features or deployment improvements.

4.2 FUTURE WORK

Advanced filtering on dashboards to allow users to sort reports by category or department. Adding a search functionality for blockchain records on the Student dashboard can enhance accessibility. Incorporating pagination for audit logs on the Admin dashboard will improve performance with large datasets. Lastly, deploying the system on a cloud platform with automated backups can ensure scalability and data redundancy.

REFERENCES

[1] Wang, X., Li, Y., & Zhang, Z., "Blockchain-Based Data Integrity Verification for Secure Applications", 2023

[2] Sharma, A., & Jain, P. K., "Role-Based Access Control with Blockchain for Secured Data Sharing", 2022

[3] Patel, S., Shah, K., & Thakkar, R., "Hybrid Blockchain for Secure and Scalable Data Management", 2020

[4] Chen, J., Xu, L., & Zhang, Q., "Blockchain-Based Audit Transparent Data Management", 2020

[5] Kim, Laura J., "Smart Data Integration for Universities", 2019

[6] Gupta, H., Yadav, M., & Choudhury, T., "Smart Contract-Based Access Control for Decentralized Data Sharing", 2023

[7] Rahman, M. A., & Sarker, I. H., "Blockchain for Education Data Security: A Survey and Roadmap", 2022

[8] Liu, Y., Zhang, C., & Wang, F., "A Decentralized Framework for Institutional Report Verification using Blockchain", 2021

[9] Verma, S., & Bhatia, P., "Security and Privacy in Educational Record Systems using Blockchain", 2021

[10] Das, R., & Roy, A., "Improving Trust in Academic Records with Blockchain and Smart Contracts", 2020

[11] Nguyen, T. H., Pham, L. V., & Tran, D. K., "Blockchain-Enabled Real Time Analytics for Institutional Reporting", 2024

[12] Kaur, M., Singh, R., & Sharma, V., "Scalable Hybrid Blockchain Architectures for Educational Data Management", 2023

[13] Zhou, H., Li, X., & Yang, Q., "Smart Contracts for Automated Compliance in Academic Reporting", 2024

6.2 APPENDIX

Dashboard_student.html

```
{% extends "base.html" %}
{% block title %}Student Dashboard{% endblock %}
{% block content %}
<div class="dashboard-wrapper">
  <!-- Hero Section -->
  <div class="hero-section">
    <h1 class="hero-title">Student Dashboard</h1>
  </div>
  <!-- Your Records Section -->
  <div class="section-card">
    <h2>Your Records</h2>
    <div class="table-container">
      <table class="table">
        <thead>
          <tr>
            <th>Timestamp</th>
            <th>Data</th>
            <th>Category</th>
            <th>Department</th>
          </tr>
        </thead>
        <tbody>
          {% for record in records %}
            <tr>
              <td>{{ record.timestamp | datetime }}</td>
              <td>{{ record.data }}</td>
              <td>{{ record.category }}</td>
              <td>{{ record.department }}</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
  <!-- Annual Reports Section -->
  <div class="section-card">
    <h2>Annual Reports</h2>
    <div class="table-container">
      <table class="table">
        <thead>
          <tr>
            <th>Timestamp</th>
            <th>Submitted By</th>
            <th>Report For</th>
            <th>Category</th>
            <th>Event Dates</th>
```

```
                    <th>Department</th>
                    <th>Data</th>
                    <th>Certificate</th>
                </tr>
            </thead>
            <tbody>
                {% for report_id, report in annual_reports.items() %}
                    <tr>
                        <td>{{ report.timestamp | datetime }}</td>
                        <td>{{ report.author }}</td>
                        <td>{{ report.target }}</td>
                        <td>{{ report.category }}</td>
                        <td>
                            {% if report.from_date == report.to_date %}
                                {{ report.from_date }}
                            {% else %}
                                {{ report.from_date }} to {{ report.to_date }}
                            {% endif %}
                        </td>
                        <td>{{ report.department }}</td>
                        <td>{{ report.data }}</td>
                        <td>
                            {% if report.certificate %}
                                <a href="{{ url_for('download_certificate', report_id=report_id) }}" class="btn btn-sm btn-
secondary">Download</a>
                            {% else %}
                                No Certificate
                            {% endif %}
                        </td>
                    </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
  </div>
</div>
```

App.py

```
from flask import Flask, render_template, request, redirect, url_for, flash, send_file
from flask_login import LoginManager, UserMixin, login_user, logout_user, login_required, current_user
from blockchain import Blockchain
from audit_log import AuditLog
import uuid, json, os, base64
from datetime import datetime
from io import BytesIO
app = Flask(__name__)
app.secret_key = 'super_secret_key_123'
login_manager = LoginManager(app)
login_manager.login_view = 'login'
BASE_DIR = '/app' if os.environ.get('RENDER') else os.path.dirname(os.path.abspath(__file__))
```

```
USER_FILE = os.path.join(BASE_DIR, 'users.json')
ATTACK_RESULT_FILE = os.path.join(BASE_DIR, 'attack_results.json')
SNAPSHOT_FILE = os.path.join(BASE_DIR, 'snapshots.json')
ANNUAL_REPORT_FILE = os.path.join(BASE_DIR, 'annual_reports.json')
def load_users():
    default_users = {
        'student1': {'password': 'pass123', 'role': 'student', 'id': str(uuid.uuid4())},
        'faculty1': {'password': 'pass456', 'role': 'faculty', 'id': str(uuid.uuid4())},
        'admin1': {'password': 'pass789', 'role': 'admin', 'id': str(uuid.uuid4())},
        'developer1': {'password': 'pass101', 'role': 'developer', 'id': str(uuid.uuid4())}
    }
    try:
        if os.path.exists(USER_FILE):
            with open(USER_FILE, 'r') as f:
                users = json.load(f)
                for data in users.values():
                    data.setdefault('id', str(uuid.uuid4()))
                return users
    except (json.JSONDecodeError, IOError):
        pass
    with open(USER_FILE, 'w') as f:
        json.dump(default_users, f, indent=4)
    return default_users
def load_annual_reports():
    try:
        if os.path.exists(ANNUAL_REPORT_FILE):
            with open(ANNUAL_REPORT_FILE, 'r') as f:
                return json.load(f)
    except (json.JSONDecodeError, IOError):
        pass
    return {}
users = load_users()
blockchain = Blockchain()
audit_log = AuditLog()
annual_reports = load_annual_reports()
class User(UserMixin):
    def __init__(self, user_id, username, role):
        self.id = user_id
        self.username = username
        self.role = role
@login_manager.user_loader
def load_user(user_id):
    return next((User(data['id'], username, data['role']) for username, data in users.items() if data['id'] == user_id), None)
@app.template_filter('datetime')
def format_datetime(timestamp):
    return datetime.fromtimestamp(timestamp).strftime('%Y-%m-%d %H:%M:%S') if isinstance(timestamp, (int, float)) else
"Invalid Timestamp"
@app.route('/')
def index():
    return redirect(url_for('dashboard')) if current_user.is_authenticated else render_template('homepage.html')
```

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username, password = request.form.get('username'), request.form.get('password')
        if username in users and users[username]['password'] == password:
            login_user(User(users[username]['id'], username, users[username]['role']))
            audit_log.log_action(username, 'login', f'User {username} logged in')
            return redirect(url_for('dashboard'))
        flash('Invalid credentials', 'error')
    return render_template('homepage.html')
@app.route('/logout')
@login_required
def logout():
    audit_log.log_action(current_user.username, 'logout', f'User {current_user.username} logged out')
    logout_user()
    flash('Logged out successfully', 'success')
    return redirect(url_for('index'))
@app.route('/dashboard')
@login_required
@app.route('/submit_annual_report', methods=['POST'])
@app.route('/update_annual_report/<report_id>', methods=['POST'])
@login_required
def handle_annual_report(report_id=None):
    if current_user.role not in ['faculty', 'admin'] or (report_id and (report_id not in annual_reports or
annual_reports[report_id]['author'] != current_user.username)):
        return 'Unauthorized', 403
    try:
        data = {k: request.form.get(k) for k in ('report_data', 'category', 'target', 'event_date', 'from_date', 'to_date', 'department')}
        certificate = request.files.get('certificate')
        if not all(v for k, v in data.items() if k != 'event_date'):
            flash('All fields except certificate required', 'error')
            return redirect(url_for('dashboard'))
        data['from_date'] = data['event_date'] or data['from_date']
        data['to_date'] = data['event_date'] or data['to_date']
        if not (data['from_date'] and data['to_date']):
            flash('Valid event dates required', 'error')
            return redirect(url_for('dashboard'))
        certificate_data, certificate_filename = (annual_reports[report_id].get('certificate'),
annual_reports[report_id].get('certificate_filename')) if report_id else (None, None)
        if certificate and certificate.filename.endswith(('.pdf', '.png', '.jpg', '.jpeg')):
            certificate_data, certificate_filename = base64.b64encode(certificate.read()).decode('utf-8'), certificate.filename
        elif certificate:
            flash('Certificate must be PDF, PNG, JPG, or JPEG', 'error')
            return redirect(url_for('dashboard'))
        validation = blockchain.validate_annual_report(data['report_data'], data['category'], data['from_date'], data['to_date'],
data['department'], current_user.username)
        if validation != 'valid':
            flash(f'Report validation failed: {validation}', 'error')
            return redirect(url_for('dashboard'))
        report_id = report_id or str(uuid.uuid4())
        block = blockchain.add_annual_report(report_id, **data, author=current_user.username)
        annual_reports[report_id] = {**data, 'author': current_user.username, 'timestamp': block['timestamp'], 'certificate': certificate_data, 'certificate_filename': certificate_filename, 'hash': block['hash']}
        with open(ANNUAL_REPORT_FILE, 'w') as f:
            json.dump(annual_reports, f, indent=4)
        audit_log.log_action(current_user.username, 'submit_annual_report' if not report_id else 'update_annual_report',
                             f'Report {report_id} {"updated" if report_id else "submitted"} ({data["category"]}) for {data["target"]}')
        flash(f'Report {"updated" if report_id else "submitted"} successfully')
    except Exception as e:
        flash(f'Error: {str(e)}', 'error')
    return redirect(url_for('dashboard'))
@app.route('/download_certificate/<report_id>')
@login_required
def download_certificate(report_id):
    if current_user.role not in ['faculty', 'admin', 'student', 'developer'] or report_id not in annual_reports or not annual_reports[report_id].get('certificate'):
        flash('No certificate available', 'error')
        return redirect(url_for('dashboard'))
    return send_file(BytesIO(base64.b64decode(annual_reports[report_id]['certificate'])),
                     download_name=annual_reports[report_id].get('certificate_filename', 'certificate.pdf'), as_attachment=True)
@app.route('/admin_add_user', methods=['POST'])
@app.route('/admin_remove_user', methods=['POST'])
@login_required
@app.route('/developer_validate_blockchain', methods=['POST'])
@app.route('/developer_analyze_snapshot', methods=['POST'])
@login_required
def developer_tools():
    if current_user.role != 'developer':
        return 'Unauthorized', 403
    try:
        action = 'validate_blockchain' if 'attack_type' in request.form else 'analyze_snapshot'
        if action == 'validate_blockchain':
            result = blockchain.simulate_attack(request.form.get('attack_type', 'tampering'))
            with open(ATTACK_RESULT_FILE, 'w') as f:
                json.dump(result, f, indent=4)
            audit_log.log_action(current_user.username, action, f'Blockchain validation: {request.form.get("attack_type", "tampering")}')
            flash(f'Validation result: {result["result"]}')
        else:
            snapshot_name = request.form.get('snapshot_name', f'Snapshot_{int(time.time())}')
            result = blockchain.analyze_snapshot(snapshot_name)
            with open(SNAPSHOT_FILE, 'w') as f:
                json.dump(result, f, indent=4)
            audit_log.log_action(current_user.username, action, f'Snapshot analysis: {snapshot_name}')
            flash('Snapshot analysis completed')
    except Exception as e:
        flash(f'Error: {str(e)}', 'error')
    return redirect(url_for('dashboard'))
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```