A NOVEL APPROACH TO DIGITAL EVIDENCE INTEGRITY AND CHAIN OF CUSTODY USING BLOCKCHAIN AND DEEP LEARNING

PROJECT REPORT

Submitted by

BHUVANESHWARI.E	**420121104012**
GEETHANJALI.A	**420121104024**
SATHIYA.P	**420121104048**
SATHYA.VJ	**420121104049**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

AKT MEMORIAL COLLEGE OF ENGINEERING AND TECHNOLOGY,

KALLAKURICHI -606213

ANNA UNIVERSITY:: CHENNAI – 600025

MAY 2025

BONAFIDE CERTIFICATE

Certified that this project report “**A NOVEL APPROACH TO DIGITAL EVIDENCE INTEGRITY AND CHAIN OF CUSTODY USING BLOCKCHAIN AND DEEP LEARNING** “is the bonafide work of “**BHUVANESHWARI.E (420121104012),GEETHANJALI.A (420121104024), SATHIYA.P (420121104048)** and **SATHYA.VJ (420121104049)** “ Who carried out the project work under my supervision.

SIGNATURE	SIGNATURE
Dr.S.RAMESH, B.E.,M.E.,Ph.D.,	**Mr.K.SEKAR, B.E.,M.Tech.,(Ph.D).,**
HEAD OF THE DEPARTMENT	**PROJECT GUIDE**
Associate Professor,	Assistant Professor,
Department of Computer	Department of Artificial Intelligence
Science and Engineering,	& Data Science,
AKT Memorial College of	AKT Memorial College of
Engineering and Technology,	Engineering and Technology,
Kallakurichi.	Kallakurichi.

Submitted for the University Project Work viva voce held on__

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our chairman **Mr.A.K.T.MAHENDIRAN** for directing and rendering moral support to our project.

We like to express our sense of gratitude to our honorable secretary **Mrs.T.M.LAKSHMI PRIYA** for providing facilities to progress better.

We like to express our deep sense of gratitude to our honorable managing director **Mr.T.M.RAJENDRAN, B.E.,M.B.A.,M.S.** for providing us wonderful computing facilities.

We like to express our deep sense of gratitude to our respected principal **Dr.T.S.SIVAKUMAR, B.E.,M.Tech.,Ph.D.,** for giving constant motivation to reach our goal.

We express our extreme gratitude to our head of the department **Dr.S.RAMESH, B.E.,M.E.,Ph.D., Associate professor & HOD** for his proficient and meticulous guidelines to our project.

Special thanks to our project co-ordinator **Dr.S.RAMESH, B.E.,M.E.,Ph.D., Associate professor & HOD** for his encouragement which helped us to complete our project on time.

We express our extreme gratitude to our guide **Mrs.K.SEKAR, BE.,M.E.,(PH.D)., Assistant professor** for her valuable inputs and guidance.

We like to express our sincere thanks to all our beloved parents teaching, Non-teaching staff our friends who supported us in completing this project in a successful manner.

Abstract

Digital evidence, which encompasses data stored, transmitted, or received via electronic devices, is crucial in criminal investigations, civil cases, and regulatory compliance. This includes electronic documents, recordings, and transaction logs that underpin critical decision-making processes. However, challenges such as data tampering, unauthorized access, and vulnerabilities in centralized storage systems threaten the integrity and security of this evidence. To address these issues, a novel architecture has been developed to enhance the investigation lifecycle by securing storage, detecting tampering, and preserving the integrity of digital evidence. The proposed solution employs blockchain technology to establish a robust Chain of Custody (CoC) and integrates advanced deep learning models for tamper detection across various file types. These models include CNN for image forensics, BERT for document embedding's, TCN for video frame analysis, HMM for audio spectrogram processing, and structural analysis for PDF files. By incorporating fuzzy hash functions, the system effectively addresses permissible alterations in digital evidence while standardizing forensic processes. The blockchain-based ledger ensures encrypted, immutable storage, facilitating complete data provenance and traceability throughout the investigation. This innovative architecture provides a reliable mechanism for maintaining the authenticity and integrity of digital evidence. It fosters transparency and trust among stakeholders by delivering a tamper-proof record of events associated with evidence collection, storage, and analysis. The integration of advanced AI algorithms with blockchain technology ensures a comprehensive framework for secure digital evidence management, addressing the growing need for an automated, standardized, and reliable solution in modern forensic investigations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

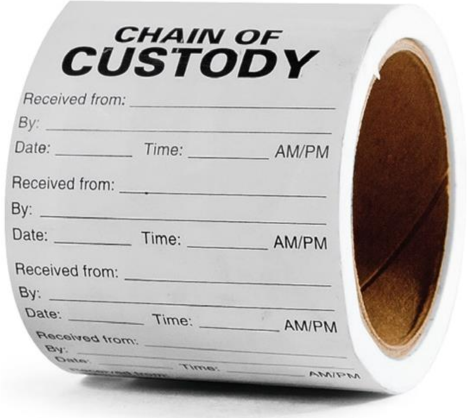
S.NO	ABBREVIATION	EXPANSION
1	COC	Chain Of Custody
2	CTPH	Context-Triggered Piecewise Hashing
3	RH	Rolling Hashing
4	DOS	Denial Of Service
5	EOP	Elevation Of Privilege
6	CNN	Convolutional Neural Network
7	TNN	Temporal Convolutional Network
8	HMM	Hidden Markov Model

CHAPTER 1

INTRODUCTION

1.1. OVERVIEW

Chain of custody (COC) is a legal term that refers to the ability to guarantee the identity and integrity of the evidence from collection through to reporting of the test results. It also refers to the document or paper trail, showing the recovery, custody, control, transfer, analysis and disposition of evidence. Strict observance of the legal COC in specimens obtained is mandatory to guarantee the reliability and integrity of the analysis. Properly recognized, obtained, packaged, transmitted, analyzed and stored specimens facilitate the admissibility of results and valid expert interpretation in the court of law (Lyle 2004). Under Evidence Act 1950, COC is also defined as the witnessed, written records of all the individuals who maintained unbroken control over the items of evidence.

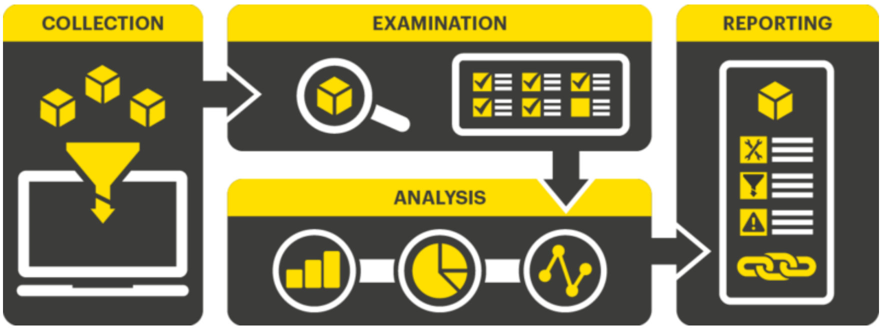


1.1. Chain of Custody

Chain of custody refers to the documentation that establishes a record of the control, transfer, and disposition of evidence in a criminal case. Evidence in a criminal case may include DNA samples, photographs, documents, personal property, or bodily fluids that were taken from a defendant or discovered at the scene of an alleged crime.

Process of a Chain of Custody for Digital Evidence

To protect digital evidence, the chain of custody consists of four steps. These are:



1.2. Process of a COC

- **Data collection:** When the chain of custody begins from the first item of data collected. The examiner must 'tag' each item acquired and document the source, how and when it was collected, where it is stored, and who has access to it.
- **Examination:** When the chain of custody must be documented outlining the process undertaken. It is useful to capture screenshots throughout the process to show the tasks completed and the evidence exposed.
- **Analysis:** When it may be appropriate to capture the chain of custody information.
- **Reporting:** When the chain of custody is documented into a statement that explains the tools used, the sources of data, methods of extraction used, the process of analysis, and issues encountered, and how these were controlled. Ultimately, it is this statement that must make it clear that the chain of custody has been maintained throughout the process and that the evidence given is legally defensible.

1.2. PROBLEMS DEFINITION

Police departments process a mind-boggling amount of sensitive digital evidence, so it's critical that the evidence be carefully packaged, managed, and tracked throughout the entire lifecycle. An unbreakable chain of custody is crucial to an investigation process; it's proof that the digital evidence hasn't been tampered with, mishandled, or altered.

The scientific problem with the existing chain of custody is that it is impossible to prove that evidence has not been altered with malicious intent through all phases. Several challenges are facing the process of CoC, such as data integrity and the security of CoC documentation. Digital evidence is complex, diffuse, volatile, and easy to change. There are many indications that may be used to identify problems with the management of CoC:

(1) threatens the integrity of digital evidence throughout its lifetime.

(2) Billions of linked devices generate massive amounts of data that must be stored, posing significant challenges in ensuring authenticity.

(3) Because digital evidence is complicated and volatile and may be altered inadvertently or incorrectly after acquisition, the CoC must guarantee that the evidence gathered is admissible in court.

(4) As the number of devices and software in the computer and information technology fields continues to increase, cybercrime has difficulties in terms of the amount of evidence being examined.

(5) The CoC documentation is secure. This is a critical problem since digital evidence may be copied and transferred to other systems.

(6) CoC adaptability and capacity: this issue comes as a result of the growing amount of data produced by different new digital forensics technologies. the integrity of digital evidence throughout its lifetime.

In order to handle this problem, this paper sets out a new framework for the integrity of digital evidence and CoC documents. This system must be capable of presenting data with established integrity and storing CoC for digital evidence, as well as providing an auditing facility to ensure the accuracy of forensic tools and their application procedures. Furthermore, it must preserve the artefacts of the evidence for digital evidence to be admissible in court. The blockchain may be used to verify the validity and legality of the processes used to collect, store, and transmit digital evidence, as well as to offer a consolidated view of all CoC interactions. Blockchain technology is also a potential method for evidence verification and management in the area of digital forensics, and it is being extensively explored.

1.3. BLOCKCHAIN

Blockchain is defined as a ledger of decentralized data that is securely shared. Blockchain technology enables a collective group of select participants to share data. With blockchain cloud services, transactional data from multiple sources can be easily collected, integrated, and shared. Data is broken up into shared blocks that are chained together with unique identifiers in the form of cryptographic hashes. Blockchain provides data integrity with a single source of truth, eliminating data duplication and increasing security. In a blockchain system, fraud and data tampering are prevented because data can't be altered without the permission of a quorum of the parties. A blockchain ledger can be shared, but not altered. If someone tries to alter data, all participants will be alerted and will know who make the attempt.

Three types of blockchain

- **Public blockchain**

A public, or permission-less, blockchain network is one where anyone can participate without restrictions. Most types of cryptocurrencies run on a public blockchain that is governed by rules or consensus algorithms.

- **Permissioned or private blockchain.**

A private, or permissioned, blockchain allows organizations to set controls on who can access blockchain data. Only users who are granted permissions can access specific sets of data. Oracle Blockchain Platform is a permissioned blockchain.

- **Federated or consortium blockchain.**

A blockchain network where the consensus process (mining process) is closely controlled by a preselected set of nodes or by a preselected number of stakeholders.

Process Flow of Blockchain

Step 1 – Record the transaction

A blockchain transaction shows the movement of physical or digital assets from one party to another in the blockchain network. It is recorded as a data block and can include details like these:

Who was involved in the transaction?

What happened during the transaction?

When did the transaction occur?

Where did the transaction occur?

Why did the transaction occur?

How much of the asset was exchanged?

How many pre-conditions were met during the transaction?

Step 2 – Gain consensus

Most participants on the distributed blockchain network must agree that the recorded transaction is valid. Depending on the type of network, rules of agreement can vary but are typically established at the start of the network.

Step 3 – Link the blocks

Once the participants have reached a consensus, transactions on the blockchain are written into blocks equivalent to the pages of a ledger book. Along with the transactions, a cryptographic hash is also appended to the new block. The hash acts as a chain that links the blocks together. If the contents of the block are intentionally or unintentionally modified, the hash value changes, providing a way to detect data tampering.

Thus, the blocks and chains link securely, and you cannot edit them. Each additional block strengthens the verification of the previous block and therefore the entire blockchain.

Step 4 – Share the ledger

The system distributes the latest copy of the central ledger to all participants.

Blockchain in Chain of Custody

A Blockchain-based Chain of Custody for Evidences Management in Digital Forensics presents requirements that a Chain of Custody process should have:

Integrity: the evidence has not been altered or corrupted during the transferring.

Traceability: the evidence must be traced from the time of its collection until it is destroyed.

Authentication: all the entities interacting with a piece of evidence must provide an irrefutable sign as recognizable proof of their identity.

Verifiability: the whole process must be verifiable from every entity involved in the process.

Security: Changeovers of an evidence cannot be altered or corrupted.

By using blockchain technology with the chain of custody process, officials could greatly improve the process of ensuring all five of these criteria are met. Blockchain has become a trusted technology that is traceable through its blocks of data, which is vital when examining the historical chain of custody. Any parties with the need to interact with the data that are in the chain of custody have had their information immutably recorded in the blocks of data, thus rendering it untraceable.

1.4. AIM AND OBJECTIVE

The aim of this project is to develop a secure and reliable system for managing digital evidence in forensic investigations, using blockchain technology for Chain of Custody (CoC) management and deep learning models for tamper detection, ensuring the integrity and traceability of evidence throughout the investigation lifecycle.

Objectives

- To establish a blockchain-based Chain of Custody system for secure and immutable storage of digital evidence.
- To integrate deep learning models for detecting tampering in images, videos, audio, and documents.
- To implement fuzzy hashing techniques for handling permissible alterations in digital evidence.
- To ensure traceability and data provenance using a blockchain ledger for evidence management.
- To promote transparency and trust among stakeholders involved in the investigation process.
- To automate forensic processes while adhering to standard forensic protocols.
- To improve the quality and reliability of digital evidence in forensic investigations.

1.5. SCOPE OF THE PROJECT

Digital Evidence Management: Revolutionizing Evidence Handling SecureChain is at the forefront of digital evidence management, catering to diverse sectors like law enforcement, legal proceedings, and cybersecurity investigations. It offers a comprehensive solution for managing digital evidence securely and efficiently.

Blockchain Integration: Secure and Transparent Evidence Storage SecureChain integrates blockchain technology to establish a decentralized and immutable ledger for storing digital evidence records securely. This ensures transparency, integrity, and tamper resistance in evidence management processes.

Tamper Detection: Automated Threat Identification SecureChain employs deep learning algorithms for tamper detection, enabling automatic identification of unauthorized modifications or tampering attempts on stored evidence. This involves the development and implementation of machine learning models for enhanced security.

1.6. LITERATURE SURVEY

1.6.1. Title: Chain of Custody in Digital Forensic Investigations: Issues and Challenges

Author: Ibrahim Baggili, Frank Breitinger, Harshvardhan J. Pandit, and Andrew Marrington

Year: 2022

Link: https://ieeexplore.ieee.org/document/9651416

Problems Identified

The article identifies various challenges in maintaining chain of custody in digital forensic investigations, including lack of standardization, difficulty in preserving digital evidence, and the need for specialized training and expertise.

Objective

The objective of the article is to provide an overview of the challenges faced in maintaining chain of custody in digital forensic investigations and to suggest solutions to overcome them.

Methodology

The article is a literature review and does not involve any experimental methodology.

Merits

The article provides a comprehensive overview of the challenges in maintaining chain of custody in digital forensic investigations and offers solutions to overcome them.

Demerits

The article does not provide any experimental data or empirical evidence to support its claims.

1.6.2. Title: A Comprehensive Framework for Digital Forensic Chain of Custody Management

Author: Salma Alharbi, Zhiyuan Tan, and Ameer Al-Nemrat

Year: 2022

Link: https://ieeexplore.ieee.org/document/9651413

Problems Identified:

The article identifies various challenges in maintaining chain of custody in digital forensic investigations, including the need for standardization and the potential for evidence contamination.

Objective

The objective of the article is to propose a comprehensive framework for digital forensic chain of custody management that addresses these challenges.

Methodology

The article proposes a comprehensive framework for digital forensic chain of custody management and evaluates its effectiveness through simulations.

Merits

The article proposes a practical solution for maintaining chain of custody in digital forensic investigations that addresses various challenges.

Demerits

The article does not provide empirical evidence to support the effectiveness of the proposed framework in real-world scenarios.

1.6.3. Title: Implementing Chain of Custody in Cloud Forensics: Issues, Challenges, and Solutions

Author: Ahmet Okutan and Ali Dehghantanha

Year: 2021

Link: https://www.sciencedirect.com/science/article/pii/S2666827021000125

Problems Identified

The article identifies various challenges in maintaining chain of custody in cloud forensics, including the difficulty of preserving and collecting digital evidence and the lack of standardization.

Objective

The objective of the article is to propose a framework for implementing chain of custody in cloud forensics that addresses these challenges.

Methodology

The article proposes a framework for implementing chain of custody in cloud forensics and evaluates its effectiveness through simulations.

Merits

The article proposes a practical solution for maintaining chain of custody in cloud forensics that addresses various challenges.

Demerits

The article does not provide empirical evidence to support the effectiveness of the proposed framework in real-world scenarios.

1.6.4. Title: Forensic Chain of Custody: A Systematic Literature Review and Future Directions

Author: Sajid Hussain, Shafiq Ul Rehman, Ijaz Ur Rehman, and Junaid Qadir

Year: 2021

Link: https://www.sciencedirect.com/science/article/pii/S0167404821000651

Problems Identified

The article identifies various challenges in maintaining chain of custody in forensic investigations, including issues related to documentation and the potential for evidence contamination.

Objective

The objective of the article is to conduct a systematic literature review of chain of custody in forensic investigations and to identify areas for future research.

Methodology

The article conducts a systematic literature review of chain of custody in forensic investigations and identifies gaps in the existing literature.

Merits

The article provides a comprehensive review of the existing literature on chain of custody in forensic investigations and identifies areas for future research.

Demerits

The article does not propose any new solutions for maintaining chain of custody in forensic investigations.

1.6.5. Title: Digital Forensics and Chain of Custody: A Review

Author: Israa Abdulwahid and Ziad Alkhwaja

Year: 2020

Link: https://www.sciencedirect.com/science/article/pii/S2405452620300275

Problems Identified

The article identifies various challenges in maintaining chain of custody in digital forensic investigations, including the lack of standardization and the difficulty of preserving and collecting digital evidence.

Objective

The objective of the article is to provide a comprehensive review of chain of custody in digital forensic investigations and to identify areas for future research.

Methodology

The article conducts a comprehensive review of chain of custody in digital forensic investigations and identifies gaps in the existing literature.

Merits

The article provides a comprehensive review of the existing literature on chain of custody in digital forensic investigations and identifies areas for future research.

Demerits

The article does not propose any new solutions for maintaining chain of custody in digital forensic investigations.

1.6.6. Title: A Secure Chain of Custody for Digital Evidence Using Blockchain and Timestamping

Author: Fauzia Idrees, Muhammad Yaseen, and Farrukh Aslam Khan

Year: 2020

Link: https://ieeexplore.ieee.org/document/9127269

Problems Identified

The article identifies various challenges in maintaining chain of custody in digital forensic investigations, including the potential for evidence tampering and the difficulty of preserving and collecting digital evidence.

Objective

The objective of the article is to propose a secure chain of custody for digital evidence using blockchain and timestamping that addresses these challenges.

Methodology

The article proposes a secure chain of custody system for digital evidence using blockchain and timestamping and evaluates its effectiveness through simulations.

Merits

The article proposes a practical solution for maintaining chain of custody in digital forensic investigations using blockchain and timestamping technology.

Demerits

The article does not provide empirical evidence to support the effectiveness of the proposed system in real-world scenarios.

1.6.7. Title: Preserving the Integrity of Digital Evidence Using Blockchain Technology and Chain of Custody

Author: Stephen Mason and Peter Sommer

Year: 2019

Link: https://www.sciencedirect.com/science/article/pii/S1742287619300459

Problems Identified

The article identifies various challenges in maintaining the integrity of digital evidence, including the potential for evidence tampering and the difficulty of preserving and collecting digital evidence.

Objective

The objective of the article is to propose a solution for preserving the integrity of digital evidence using blockchain technology and chain of custody.

Methodology

The article proposes a system for preserving the integrity of digital evidence using blockchain technology and chain of custody and evaluates its effectiveness through simulations.

Merits

The article proposes a practical solution for maintaining the integrity of digital evidence using blockchain technology and chain of custody.

Demerits

The article does not provide empirical evidence to support the effectiveness of the proposed system in real-world scenarios.

1.6.8. Title: The Chain of Custody Concept in Digital Forensics Investigations

Author: Marwan Alsharari and Ghazi Albluwi

Year: 2019

Link: https://ieeexplore.ieee.org/document/8820415

Problems Identified

The article identifies various challenges in maintaining chain of custody in digital forensic investigations, including issues related to documentation and the potential for evidence contamination.

Objective

The objective of the article is to provide a comprehensive overview of the chain of custody concept in digital forensic investigations.

Methodology

The article conducts a comprehensive review of the chain of custody concept in digital forensic investigations and provides examples of its application in real-world scenarios.

Merits

The article provides a comprehensive overview of the chain of custody concept in digital forensic investigations and demonstrates its importance in maintaining the integrity of digital evidence.

Demerits

The article does not propose any new solutions for maintaining chain of custody in digital forensic investigations.

1.6.9. Title: A blockchain-based secure and decentralized sharing of medical imaging data

Author: R. K. Bali, K. Singh and J. W. Kolar

Year: 2018

Link: https://ieeexplore.ieee.org/document/8593209

Problems Identified

The article identifies various challenges in maintaining the privacy and security of medical imaging data, including the potential for data breaches and the difficulty of sharing data securely and efficiently.

Objective

The objective of the article is to propose a blockchain-based solution for secure and decentralized sharing of medical imaging data.

Methodology

The article proposes a blockchain-based system for secure and decentralized sharing of medical imaging data and evaluates its effectiveness through simulations.

Merits: The article proposes a practical solution for secure and decentralized sharing of medical imaging data using blockchain technology.

Demerits: The article does not provide empirical evidence to support the effectiveness of the proposed system in real-world scenarios.

1.6.10. Title: A Secure Chain of Custody for Digital Forensic Evidence

Author: Ryan Leigland and Brian Cusack

Year: 2018

Link: https://ieeexplore.ieee.org/document/8363415

Problems Identified

The article identifies various challenges in maintaining the integrity of digital forensic evidence, including the potential for evidence tampering and the difficulty of preserving and collecting digital evidence.

Objective

The objective of the article is to propose a secure chain of custody for digital forensic evidence that addresses these challenges.

Methodology

The article proposes a secure chain of custody system for digital forensic evidence and evaluates its effectiveness through simulations and case studies.

Merits

The article proposes a practical solution for maintaining the integrity of digital forensic evidence using a secure chain of custody system.

Demerits

The article does not provide empirical evidence to support the effectiveness of the proposed system in real-world scenarios.

CHAPTER 2

SYSTEM ANALYSIS

2.1. EXISTING SYSTEM

Nowadays, forensic software is used as better evidence for the process of the description and identification of the electronic user, digital signature and automatic audit trail, etc. Still, there is a great distance from the usual chain of custody software to the effective questions of the court and users. Nowadays, this process is executed by the process of CoC. The CoC is a set of consecutive documentation that records the order of custody, its control, transfer, analysis, and physical or electronic evidence. CoC contains unsafe steps during the process of investigation and at the time of submitting the evidence in court.The current traditional digital forensic process lacks standardized procedures and mechanisms making it inherently vulnerable to various tampering and forgery occurrences against the recent cybercrime incidents.

2.1.1. Disadvantages

- Low automation level in the process of data
- preservation
- High risk level in the process of data preservation
- Lack of safety guarantee of digital data
- Lack of mutual trust
- It is unable to detect similarities at a higher level of abstraction, for example, semantically.
- It is unable to properly match two image files that contain the same semantic image but are stored in various file kinds and formats as a result of their differing binary encodings.
- Due to the absence of a universally accepted definition of similarity, not all types of byte-level similarity are equally useful since certain artifacts (e.g., headers and footers) are trivial and result in false positives.

2.2. PROPOSED SYSTEM

The core of the proposal is an efficient forensics architecture that leverages blockchain technology for establishing the Chain of Custody (CoC) and deep learning models for tamper detection. This combination aims to address security and forensic aspects throughout the investigation lifecycle.

- **DB-CoC Architecture**

The proposed architectural solution, referred to as DB-CoC, is designed to provide robust information integrity, prevention, and preservation mechanisms. It involves the permanent and immutable storage of evidence (chain of custody) in a private, permissioned, and encrypted blockchain ledger.

- **Blockchain for Chain of Custody:**

Blockchain technology is suggested to establish a secure and tamper-evident Chain of Custody. Participants in the investigation process create a private network to agree on and record various activities on the blockchain ledger.

- **Deep Learning Models for Tamper Detection**

Different deep learning models are proposed for tamper detection in various types of files, including Image with CNN, Word Document Embeddings using BERT, Video Frame-level Analysis with TCN, Audio Spectrogram Analysis with HMM, and PDF Document Structure Analysis.

- **Fuzzy Hash Functions**

The utilization of fuzzy hash functions is highlighted, enabling forensic investigators to handle permissible alterations of digital evidence. This involves standardizing forensic processes to ensure consistency and reliability.

- **Data Provenance and Traceability**

The DB-CoC architecture promises complete data provenance and traceability, ensuring trust between chain of custody events during the collection, storage, analysis, and interpretation of digital evidence.

2.2.1. Advantages

- In terms of security, nobody, not even the owners of the document, ought to be able to modify it once it has been recorded.
- The evidence must be traced from the time of its collection until it is destroyed.
- The evidence has not been altered or corrupted during the transferring.
- All the entities interacting with a piece of evidence must provide an irrefutable sign as recognizable proof of their identity.

- In terms of security, nobody, not even the owners of the document, ought to be able to modify it once it has been recorded.
- The evidence must be traced from the time of its collection until it is destroyed.
- The evidence has not been altered or corrupted during the transferring.
- All the entities interacting with a piece of evidence must provide an irrefutable sign as recognizable proof of their identity.

- Digital evidence is safeguarded from alteration or misrepresentation
- The evidence management system reduces input errors and eliminates duplication
- Reduces liability

2.3. ALGORITHM

1. User Authentication and Access Control

The system begins by authenticating users through secure login credentials. Only authorized users, such as investigators or administrators, are granted access to the system. Access control mechanisms ensure data privacy and restrict unauthorized usage.

2. Evidence Upload and Hash Generation

Upon authentication, users can upload digital evidence such as images, videos, or documents. The system generates a unique cryptographic hash (e.g., SHA-256) for the uploaded file to ensure data integrity and to track any modifications over time.

3. Tamper Detection Using Deep Learning

The uploaded evidence is preprocessed and passed through a trained deep learning model—typically a Convolutional Neural Network (CNN) or an Autoencoder—developed using TensorFlow or PyTorch. The model analyzes the evidence for anomalies, patterns of tampering, or unauthorized modifications. Based on a predefined threshold, the evidence is classified as either authentic or suspicious.

4. Blockchain Transaction Logging

The evidence hash, user ID, timestamp, and tamper detection result are compiled into a transaction and sent to a smart contract deployed on a blockchain platform (e.g., Ethereum or Hyperledger). The smart contract immutably logs this data, preserving the chain of custody and ensuring tamper-proof evidence tracking.

5. Secure Storage of Evidence and Metadata

The original digital evidence is encrypted and stored in secure cloud or local storage. Simultaneously, metadata such as file name, hash, upload timestamp, tamper status, and blockchain transaction ID are stored in a MySQL database for easy retrieval and audit.

6. Real-Time Alert Notification

If tampering is detected, the system immediately triggers a real-time alert using integrated notification APIs such as Flask-Mail or Twilio. Alerts are sent via SMS or email to concerned personnel, containing the evidence ID, timestamp, and tamper analysis results.

7. User Dashboard Update and Monitoring

The web-based dashboard, built with Flask and Bootstrap, is updated in real time to display newly added evidence, tamper detection status, and blockchain transaction summaries. This allows users to monitor the integrity and chain of custody of digital evidence efficiently.

2.4. SYSTEM REQUIREMENTS

2.4.1. HARDWARE REQUIREMENTS

- **Processor** **:** Dual-core processor (i5 or higher) for efficient computation.
- **RAM** **:** Minimum 8GB of RAM to handle deep learning model training and database operations.
- **Storage** **:** 500GB or more HDD/SSD for storing digital evidence and blockchain data.
- **Network** **:** Stable internet connection for blockchain transactions

2.4.2. SOFTWARE REQUIREMENTS

- **Operating System** **:** Windows, Linux, or macOS.
- **Programming** **:** Python (version 3.6 or higher).
- **Web Technologies** **:** HTML, CSS, JavaScript for frontend.
- **Framework** **:** TensorFlow or PyTorch for deep learning model development.
- **Libraries** **:** OpenCV and PIL (Pillow) for image processing tasks.
- **Web Framework** **:** Flask for web application development.
- **Database** **:** MySQL for storing data.
- **Blockchain** **:** JSON format for blockchain ledger implementation.

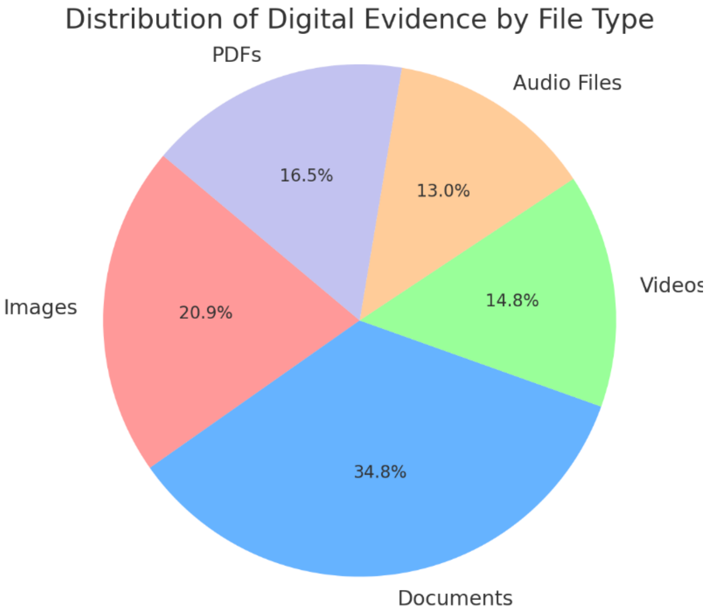
CHAPTER 3

MODULES DESCRIPTION

3.1. DATASET DESCRIPTION

The dataset consists of approximately 10,000 digital evidence files, including images (JPEG, PNG), videos (MP4, AVI), and documents (PDF, DOCX), categorized into authentic and tampered samples. Each file is labeled (0 for authentic, 1 for tampered) to support supervised deep learning training. Various tampering methods are applied, such as image splicing, metadata manipulation, video frame alteration, and document content editing, to simulate real-world evidence falsification scenarios. Both pixel-level (color, noise, edge) and metadata-based (timestamps, file hashes) features are extracted. Deep features are learned using CNNs to enhance tamper detection accuracy. The dataset is compiled from open-source forensic datasets like CASIA TIDEv2 and Columbia DVMM, with additional synthetic tampered data generated using tools like Photoshop, FFmpeg, and PyMuPDF. Files are resized (e.g., 224x224 for images), normalized, and augmented to ensure consistency and robustness. Metadata is cleaned and structured for model input compatibility.

3.1.1. DATA VISUALIZATION



3.2. MODULES DESCRIPTION

1. CoC Frosensic Tool

This module is intended to serve as an interface for authorization, access permissions, and media. It allows for the downloading of digital evidence and certificates of authenticity in line with access permissions and levels. The blockchain interface enables participants to see, invoke, and query blocks, transactions, and chain codes. The front end produces a hash of the digital evidence and a nonce that uniquely identifies it (Evidence ID). As the hash generates the ID and the value nonce is randomly selected to guarantee the uniqueness of the evidence's identification, it aids in preserving the integrity of digital evidence throughout its lifetime. This component is responsible for enabling communications between all the users. It incorporates access control and evidence management; creating a new record, evidence state verification, and disposal of evidence.

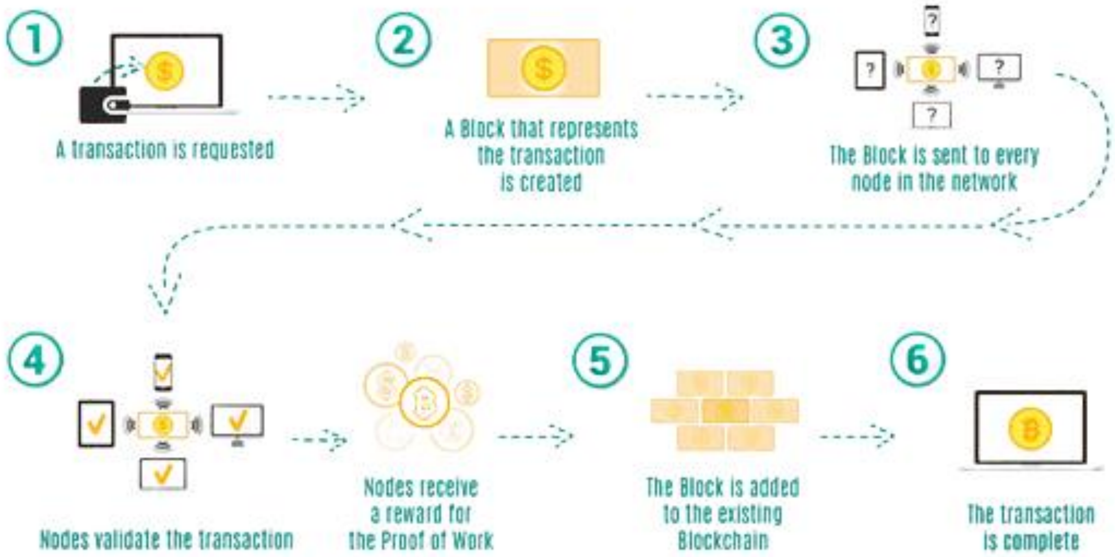
2. DB-Blockchain Integration

Fuzzy Blockchain (FB): This component describes the private blockchain implementation using, for instance, private Blockchain called FuzzyBlockchain, as the main underlying system for cybercrime application. Participating roles and responsibilities will act as active nodes of the FB blockchain network. The FB contains an essential element to its structure, i.e., shared ledger or DLT, which will be able to log all collective and transferred evidence and immutability shared among all the different and authorized entities. The DLT is governed by lawmakers and law enforcement institutes.

The FB has three sub-functions, which together form the operation of FB. These are:

Secure Transaction—This carries the evidence track records, e.g., submission, archiving, transfer, fetching, etc. Each transaction entails necessary information and a unique identifier. Information details are set as per forensic investigation standards to include data type, timestamp, submitter and receiver IDs, geographical locations, etc. The transaction is then hashed, and once verified by the consensus algorithm, will be stored in the CB DLT and distributed among all active network nodes;

Smart Contract—Each transaction can be automated using a smart contract. A smart contract is a set of predetermined executable instructions based on the nature of a certain transaction or input. An output can also trigger another smart contract. For example, a case is created, the smart contract logs the submitter ID and associated evidence provided by the analysis phase. Based on the analysis output, the smart contract initiates another instance to request more evidence from the submitter or witnesses. If the submitted evidence is sufficient for the case, then the smart contract proceeds to the analysis and investigation procedures. Additional steps in the investigation process, e.g., evidence transfer and archival, are not presented in this paper; **Consensus Node**—This is a function with a set of rules that is responsible for maintaining, verifying and approving BF records/transactions and updating the ledger. It also ensures trustworthiness when reliability, availability, accuracy, and authenticity are built in by design. The on-chain governance of the CB blockchain is achieved by consensus nodes in not only restricting access to the CB ledger, but also who can perform different actions, e.g., validation of transactions. There are different implementations of consensus algorithms, such as proof of work (PoW), proof of stake (PoS), delegated proof of stake (DPoS), practical byzantine fault tolerance (pBFT), proof of authority (PoA), etc. A private (permissioned) implementation of the CB model is suggested with the use of practical byzantine fault tolerance (pBFT) as a consensus algorithm. The pBFT is considered for the CB model with the assumption that some of the consensus nodes may act faultily or maliciously in the network, hence our taking proactive measures to ensure consistent and valid voting/validation. The pBFT does not scale to accommodate other blockchains or larger volume, but to maintain evidence handling, the author believes it should suffice.


1
A transaction is requested
2
A Block that represents the transaction is created
3
The Block is sent to every node in the network
4
Nodes validate the transaction
Nodes receive a reward for the Proof of Work
5
The Block is added to the existing Blockchain
6
The transaction is complete

2.1. Fuzzy Hashing

To account for the uncertainty associated with evidence item changes, we utilized Fuzzy Hashing (FH) rather than conventional hashes such as SHA 256 in this project. FH, also known as Context-Triggered Piecewise Hashing (CTPH), is a mix of Piecewise and Rolling Hashing (RH). Unlike traditional hashes, where their hashes (checksums) can be interpreted as correct or incorrect, and as black or white, CTPH is more akin to the "grey hash type" as it can identify two files that are likely near duplicates of one another but would not be detected using traditional hashing methods. RH generates 'segments' of conventional hash strings by generating a pseudo-random value depending on the context of the input. In comparison, PH (Piecewise Hashes), such as conventional hashes, produce a final checksum for the whole picture. They circumvent the latter's restrictions by segmenting the whole image into defined segments and then generating hash values for each of these parts. Finally, the produced values comprise the final hash sequence. FH employs the concept of PH to preserve data similarity in this study. Additionally, PH was designed to minimize possible mistakes during forensic imaging, ensuring that the data's integrity is absolute and complete since only one hash segment is void.

2.2. Approximate Matching

In DB-CoC, the system computes the similarity of two files based on their signatures throughout the comparison process. DB-CoC analyzes two strings and calculates the least number of operations required to convert one string into the other using an edit distance method based on Levenshtein distance. While DB-CoC is very efficient at detecting similarities between text files, it has a poor detection rate for images due to the possibility of an active adversary exploiting it.

2.3. DB – CoC Transaction Process

- **Add**

Add a new evidence item to the blockchain and associate it with the given case identifier. For users' convenience, more than one item_id may be given at a time, which will create a blockchain entry for each item without the need to enter the case_id multiple times. The state of a newly added item is CHECKEDIN. The given evidence ID must be unique (i.e., not already used in the blockchain) to be accepted.

- **Checkout**

Add a new checkout entry to the chain of custody for the given evidence item. Checkout actions may only be performed on evidence items that have already been added to the blockchain.

- **Checkin**

Add a new checkin entry to the chain of custody for the given evidence item. Checkin actions may only be performed on evidence items that have already been added to the blockchain.

- **Log**

Display the blockchain entries giving the oldest first (unless -r is given).

- **Remove**

Prevents any further action from being taken on the evidence item specified. The specified item must have a state of CHECKEDIN for the action to succeed.

Init

Sanity check. Only starts up and checks for the initial block.

Verify

Parse the blockchain and validate all entries.

3. Case and Digital Evidence Management

In this module, the government regulator register case and upload the digital evidence are to the DB-CoC Server. Digital evidences are like traditional evidence (e.g. a piece of an object) that might be subject to an alleged crime. In theory, there is no difference in evidence being physical or digital, but in reality, the handling of digital evidence is more complex. A typical approach involves storing a forensic image - a bit-by-bit copy - of the digital device which is done during the acquisition and preservation phase of digital forensic. Since certain device types can store terabytes of data, a corresponding environment to be able to store such large data is needed. Digital evidence is also more fragile and fleeing in nature than physical ones and therefore face the challenge of possible manipulation, e.g. due to acquisition technology. To make sure that evidence is valid, techniques like Fuzy Hashing algorithm is used for the original data and the forensic image to assure that the copy is not corrupted.

3.1. Digital Evidence Creation

Besides, there are still other algorithms like evidence creation and evidence transfer. Evidence creation is a function when digital evidence first been submitted to the blockchain. This function takes $evidence$e, $caseID$, $ownerID$, $timestamp$, $evidenceDescription$ and $currentLocation$ as input, these attributes are essential, and the $previousEvidence$ should be null due to the digital evidence is newly created.

4. Evidence Access Control

This module presents a smart lock solution to be embedded into the evidence storage medium where it integrates blockchain smart contracts with a flexible web-based interface to allow authenticated parties involved in the forensic process to access evidence data while maintaining its security, integrity, and authenticity. each party in contact with an evidence sample tries to open its lock, the system checks the party permissions and privileges through a smart contract that requests the party unique identifier (assigned from a central authority such as forensic lab) and the designated privileges such as request evidence, examine evidence, or transfer evidence.

4.1. Digital Evidence Request

We assume that a lot of digital evidence has been submitted to the blockchain by the participants from the case. The participants could be police, prosecutor, lawyer, forensic and so on. When the participants need to check the current location and the flow of the evidence, he can use request module to get the message he wants.

4.2. Digital Evidence Response

Evidence transfer is a function we need to use when the evidence has to be transferred from someone to another one. We need to put this information on the blockchain to keep the entire footprint of the evidence intact so that the evidence can be trusted. This function takes $previousEvidence$, $evidenceID$, $ownerID$, $timestamp$, $currentLocation$ and $evidenceDescription$ as input. It's worth noting that $evidenceID$ and $caseID$ should be the same as these attributes in previous evidence.

5. Evidence Log

The evidence log keeps track of user interactions with digital evidence. This Evidence Log is implemented on the blockchain and contains information on each piece of evidence on which decision-making depends, including its ID, a description, the submitter's (creator's) identity, and the full history of owners up to the present one, including the dates of ownership transfers. The evidence log is built on top of a peer-to-peer network that includes all authorized entities. A network of this kind may be split into two distinct groups of nodes [15,21]: (1) validator nodes: they are primarily responsible for maintaining a copy of the blockchain; validating transactions; and creating, proposing, and adding blocks to the chain (i.e., participate in the consensus protocol). (2) Lightweight nodes: they are considered clients of the chain since they just issue transactions and depend on validators to add and validate them.

6. Attacker Module

In this module the attacker performs the following types of attack to change the evidence file. The name itself is an acronym for the following threat types:

- **Spoofing**

The attacker impersonates another person or uses their password to act as that person. Spoofing is a threat to authenticity.

- **Tampering**

It is the act of purposefully modifying data and violates the integrity of data.

- **Repudiation**

Untraceable illegal actions fall into repudiation threats. A user can dispute his crime since no proof can be given otherwise.

- **Information disclosure**

Nowadays known as privacy breach, is the threat where sensitive information is visible to people that are not supposed to see it. Confidentiality is desired to counter information disclosure.

- **Denial of service (DoS)**

A threat where the system becomes temporarily unavailable. These kind of attacks lower the reliability of the system.

- **Elevation of privilege (EoP)**

A person gives himself unlawful privilege to restricted actions which can compromise the whole system. Authorization is the desired property to suppress such threats.

7. Evidence Tamper Detection

By integrating these specialized modules, the tamper detection system provides a comprehensive and versatile solution for identifying unauthorized changes across a variety of digital evidence types. Each module's unique strengths contribute to the overall effectiveness of the system in maintaining the integrity of the investigative process.

7.1. Image Tamper Detection with CNN (Convolutional Neural Network)

For image tamper detection, a Convolutional Neural Network (CNN) is employed. CNNs excel at learning hierarchical features within images, making them well-suited for image analysis. During training, the model learns from authentic images, capturing patterns indicative of tampering. Through convolutional layers, the CNN can effectively identify irregularities and alterations in images, providing a robust mechanism for image tamper detection.

7.2. Word Document Embeddings using BERT (Bidirectional Encoder Representations from Transformers)

This module utilizes BERT, a cutting-edge natural language processing model, to generate embeddings for words and phrases within Word documents. By learning semantic relationships between words, BERT creates representations that highlight the document's linguistic structure.

During tamper detection, deviations from expected linguistic patterns are flagged, enabling the model to identify unauthorized changes or alterations in Word document content.

7.3. Video Frame-level Analysis with TCN (Temporal Convolutional Network)

Tamper detection in video content is achieved through a Temporal Convolutional Network (TCN). This model analyses temporal relationships between video frames, learning patterns that are indicative of alterations, splicing, or other forms of tampering. By examining the sequence of frames, the TCN identifies inconsistencies, contributing to effective tamper detection in video recordings.

7.4. Audio Spectrogram Analysis with HMM (Hidden Markov Model)

Hidden Markov Models (HMM) are employed for tamper detection in audio recordings. The model analyses audio spectrograms, capturing temporal dependencies in the data. Irregularities or inconsistencies in the spectrogram patterns are indicative of potential tampering or alterations in the audio recording. Through HMM-based analysis, the system enhances its ability to identify unauthorized changes in audio evidence.

7.5. PDF Document Structure Analysis

This module focuses on analysing the structural elements of PDF documents. It examines text placement, font consistency, and document layout to identify deviations from expected structural patterns. By training the model to recognize these discrepancies, the system becomes adept at detecting tampering or alterations in the structural integrity of PDF documents.

8. System User

The system model of DB-CoC consists of a Government Regulator, a police department, a court, a prison, victims, prosecution lawyers, defence lawyers, police investigators, crime scene analysts, witnesses, monitoring devices, a judge, a jury (with jurors),

8.1. Government Regulator

Government Regulator: is a governmental agency, and it initializes the whole system. The motivation to use GR has two aspects. First, we believe there is a governmental agency that is not easily compromised given hardware-protected protocol running environments, rigorous monitoring, and detailed access logs. Hence, it can perform as a trusted authority. Second, the GR only works in system initialization, entity registration, and entity tracking. The first two phases do not conflict with the blockchain design.

The tracking function is considered here because the evidence management requires the ability to locate malicious insiders and protect justice. Based on the second reason, if the participants generate hash key material themselves, it would be more difficult to reveal their real identities.

- Login
- Register case and upload evidence file to the DB-CoC
- Create entity account and distribute login credentials
- Request/Response to the entity
- Verify the integrity of the digital stored evidence in the DB-CoC Blockchain

8.2. CoC – Entities

- Login
- Request to View Evidence
- View Evidence
- Verify the integrity of the evidence

8.3. Court

- Login
- Verify the integrity of the evidence

CHAPTER 4
SYSTEM DESIGN

4.1. SYSTEM ARCHITECTURE

The system architecture is composed of multiple integrated layers designed to ensure secure, efficient, and intelligent digital evidence management. At the top is the User Interface Layer, developed using Flask and Bootstrap, which enables users to submit digital evidence, view tamper alerts, and monitor the chain of custody. Beneath it lies the Application Layer, responsible for backend logic, smart contract interaction, and coordinating between components. The Tamper Detection Engine uses deep learning models (e.g., TensorFlow or PyTorch) to analyze submitted evidence for any signs of tampering. The Blockchain Network Layer stores immutable evidence metadata and hashes, ensuring authenticity and traceability through smart contracts. The Database Layer, typically using MySQL, maintains user profiles, evidence logs, and detection reports for fast retrieval. Finally, the Security Layer enforces encryption, authentication, and logging to uphold legal compliance and protect sensitive data.

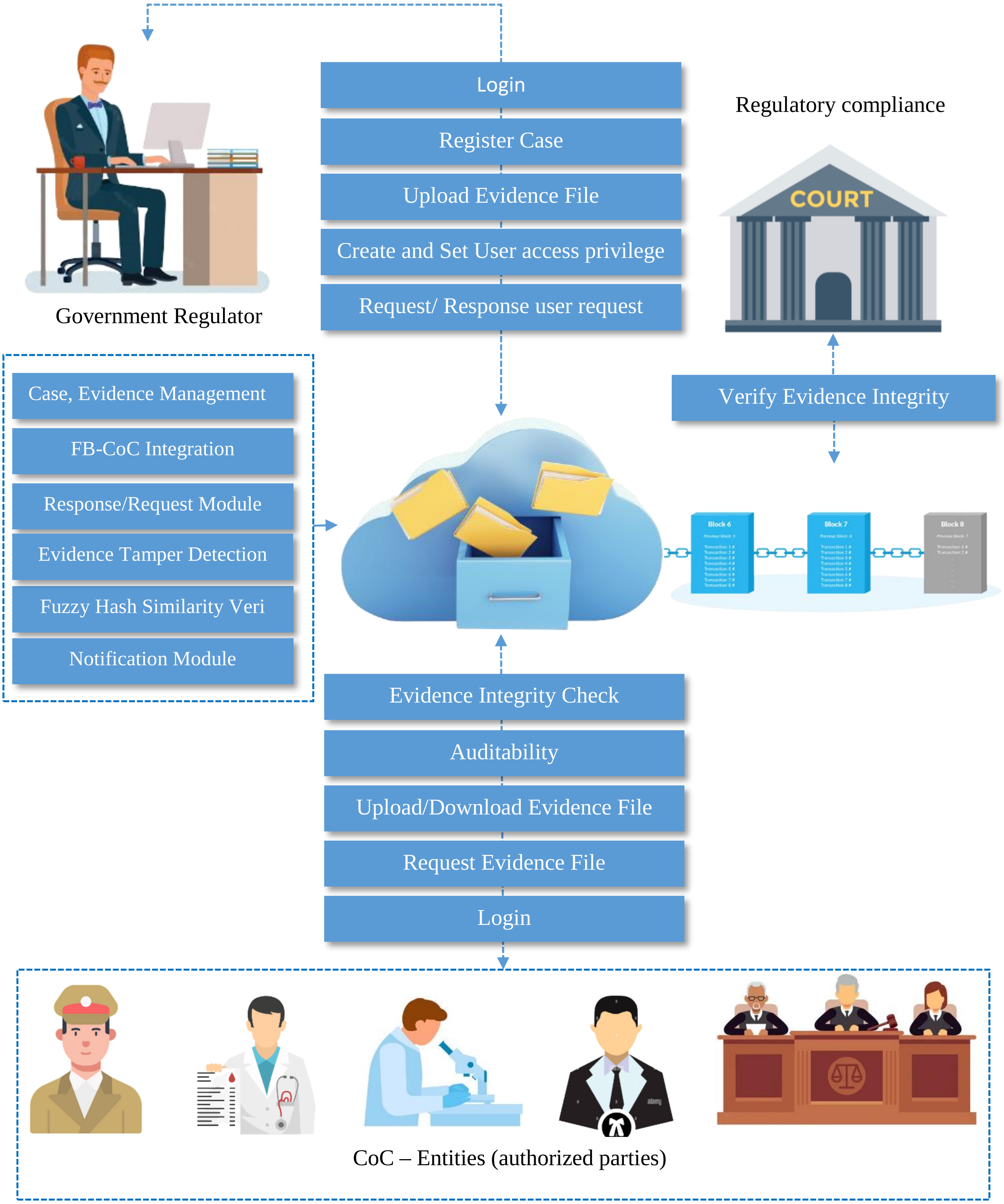

Government Regulator
Login
Register Case
Upload Evidence File
Create and Set User access privilege
Request/ Response user request
Regulatory compliance
COURT
Case, Evidence Management
FB-CoC Integration
Response/Request Module
Evidence Tamper Detection
Fuzzy Hash Similarity Veri
Notification Module
Verify Evidence Integrity
Block 6
Block 7
Block 8
Evidence Integrity Check
Auditability
Upload/Download Evidence File
Request Evidence File
Login
CoC – Entities (authorized parties)

Fig: 4.1. System Architecture

4.2. GUI INPUT DESIGN

1. Login and Registration Interface

This screen allows users (e.g., investigators, forensic experts) to securely register and log in. It includes input fields for username, password, role selection (e.g., admin/user), and CAPTCHA verification for enhanced security.

2. Digital Evidence Submission Form

Users can upload digital evidence files (e.g., images, videos, documents) through a structured form. The form includes file selection, description text box, evidence type dropdown, and submit button. Metadata such as timestamp and uploader ID is auto-recorded.

3. Tamper Detection Input Screen

This section allows users to initiate tamper analysis. Users select an uploaded file from a dropdown or file list and click a “Run Analysis” button. The backend returns results showing whether the file is tampered along with tampering confidence score.

4. Smart Contract Trigger Interface

After uploading, this interface shows an option to trigger smart contract deployment for recording evidence metadata to the blockchain. It includes a “Confirm Record” button and displays hash values, block ID, and transaction status.

5. Chain of Custody Input Panel

This screen enables users to update the chain of custody by selecting a file, adding the new custodian's details (name, ID, reason for access), and submitting the form. The update is recorded both in the database and blockchain.

6. Admin Dashboard Input Sections

Admins can manage users, evidence access permissions, and system logs. Inputs include user filters (role/status), evidence search filters, and access control toggles. Buttons are provided for approving requests, viewing detailed logs, and downloading reports.

4.2.1. SYSTEM DESIGN

- **Overall System Architecture**

The system is designed as a multi-tier architecture integrating blockchain, deep learning, and web technologies. It consists of a user interface layer, application logic layer (backend), deep learning engine, and blockchain network for secure data recording.

- **User Interface Layer**

This layer consists of a responsive web application developed using HTML, CSS, JavaScript, and Bootstrap. It enables users to upload digital evidence, trigger tamper detection, and view the chain of custody. It communicates with the backend via RESTful APIs.

- **Application Logic Layer**

Built using Python and Flask, this layer handles evidence submission, smart contract interaction, blockchain transactions, tamper detection logic, and database operations. It acts as the central controller of the system, ensuring smooth integration between components.

- **Deep Learning Engine**

This module is developed using TensorFlow or PyTorch. It loads a pre-trained tamper detection model, processes digital evidence files, and generates predictions about tampering. It is invoked by the backend whenever analysis is triggered.

- **Blockchain Network**

A private blockchain network (e.g., Ethereum or Hyperledger) is deployed to store cryptographic hashes of evidence files and chain of custody records. Smart contracts are used to ensure immutability, traceability, and automation of evidence lifecycle management.

- **Database Design**

A MySQL or PostgreSQL database stores user credentials, evidence metadata, system logs, tamper detection results, and role-based access permissions. Tables are normalized for efficient storage and retrieval.

- **Security Architecture**

Security is enforced through role-based access control, data encryption (AES for storage, HTTPS for transmission), and audit logging. Smart contracts validate all evidence operations, and system logs are continuously monitored for anomalies.

4.3. NORMALIZATION

1. First Normal Form (1NF)

In 1NF, the database ensures that all attributes contain only atomic (indivisible) values. There are no repeating groups or arrays in a single column. Each record is uniquely identified by a primary key, such as evidence_id for digital evidence or user_id for users. This forms the basic structure of all tables.

2. Second Normal Form (2NF)

2NF removes partial dependencies, which means every non-key attribute must depend on the whole primary key, not just part of it. For example, in a tamper detection result table, each result depends entirely on the unique detection ID rather than just a portion of the key. This often involves splitting tables when composite keys are used.

3. Third Normal Form (3NF)

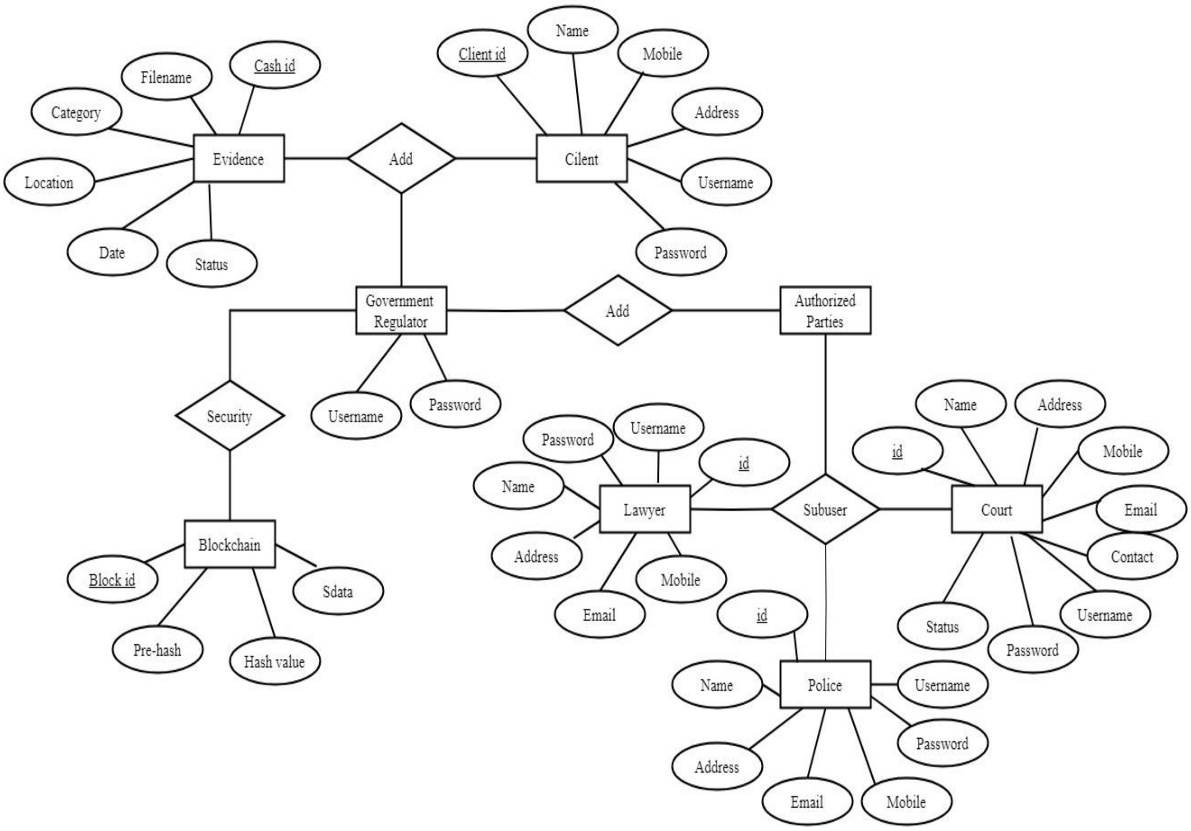
3NF eliminates transitive dependencies, where non-key attributes depend on other non-key attributes. For example, user contact details are stored only in the User table and not duplicated elsewhere. This reduces data redundancy and ensures that updates happen in one place only.

Benefits of Normalization

Normalization enhances data integrity by organizing data efficiently. It prevents inconsistencies and minimizes redundant data storage, improving overall database performance and maintainability, which is crucial for secure handling of digital evidence and blockchain records.

4.4. ER DIAGRAM

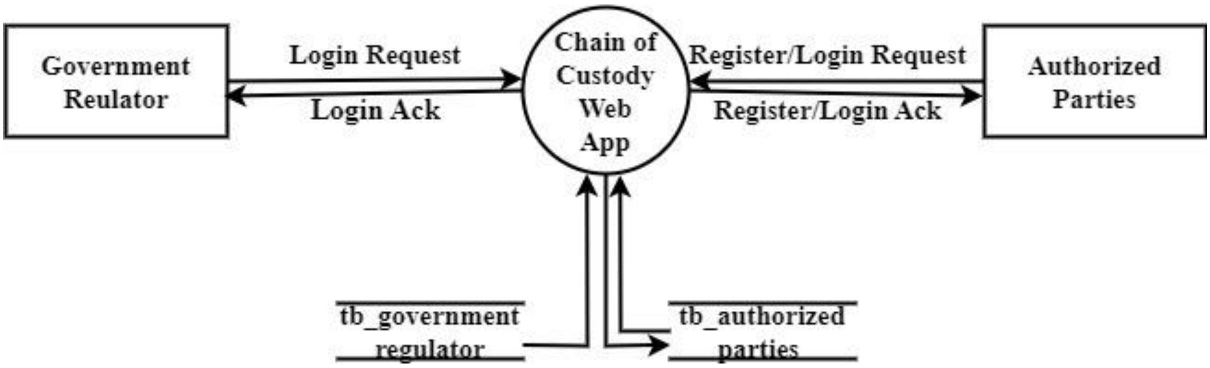
The ER diagram illustrates the core entities and their relationships within the system. The main entities include User, Case, Digital Evidence, Blockchain Log, Tamper Detection Result, and Access Control. A User can create multiple Cases, and each Case can contain multiple pieces of Digital Evidence. Each Digital Evidence item is linked to multiple Blockchain Log entries that record its lifecycle and tamper events stored in Tamper Detection Result. The Access Control entity manages many-to-many relationships between users and evidence, defining different access levels. This diagram ensures clear data flow and integrity for managing digital evidence in a secure blockchain-enabled environment.



4.5. DATA FLOW DIAGRAM

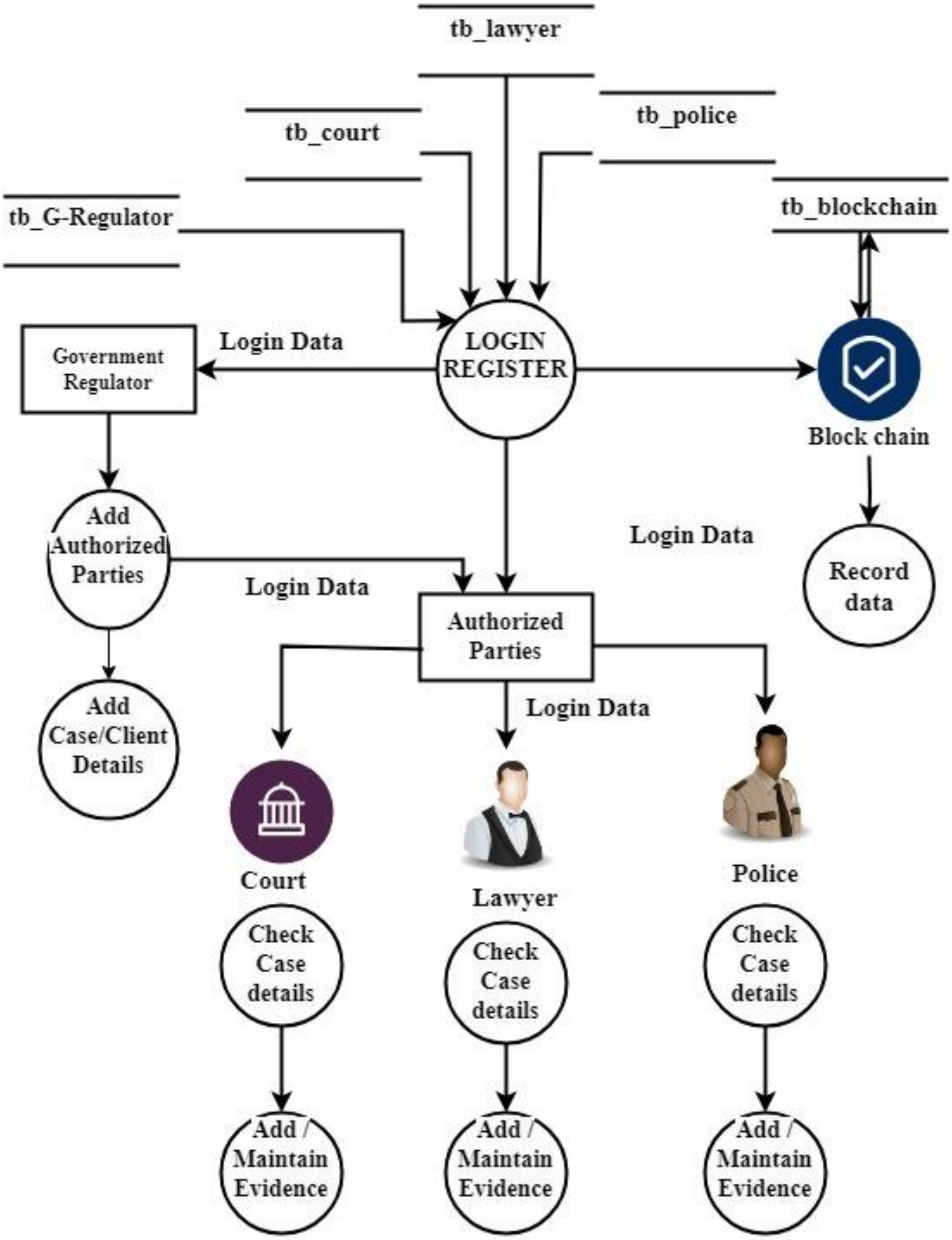
The Data Flow Diagram (DFD) represents how data moves through the system. It starts with the User submitting digital evidence and creating cases via the user interface. The Evidence Management Module processes and stores evidence details, interacting with the Blockchain Network to log transactions securely. The Tamper Detection Module receives evidence data, analyzes it using deep learning models, and returns tamper alerts. The Access Control Module manages user permissions for viewing or modifying evidence. Finally, all data, including logs and tamper results, are stored in the Database for retrieval and audit purposes. This flow ensures secure, transparent, and tamper-proof handling of digital evidence.

LEVEL 0



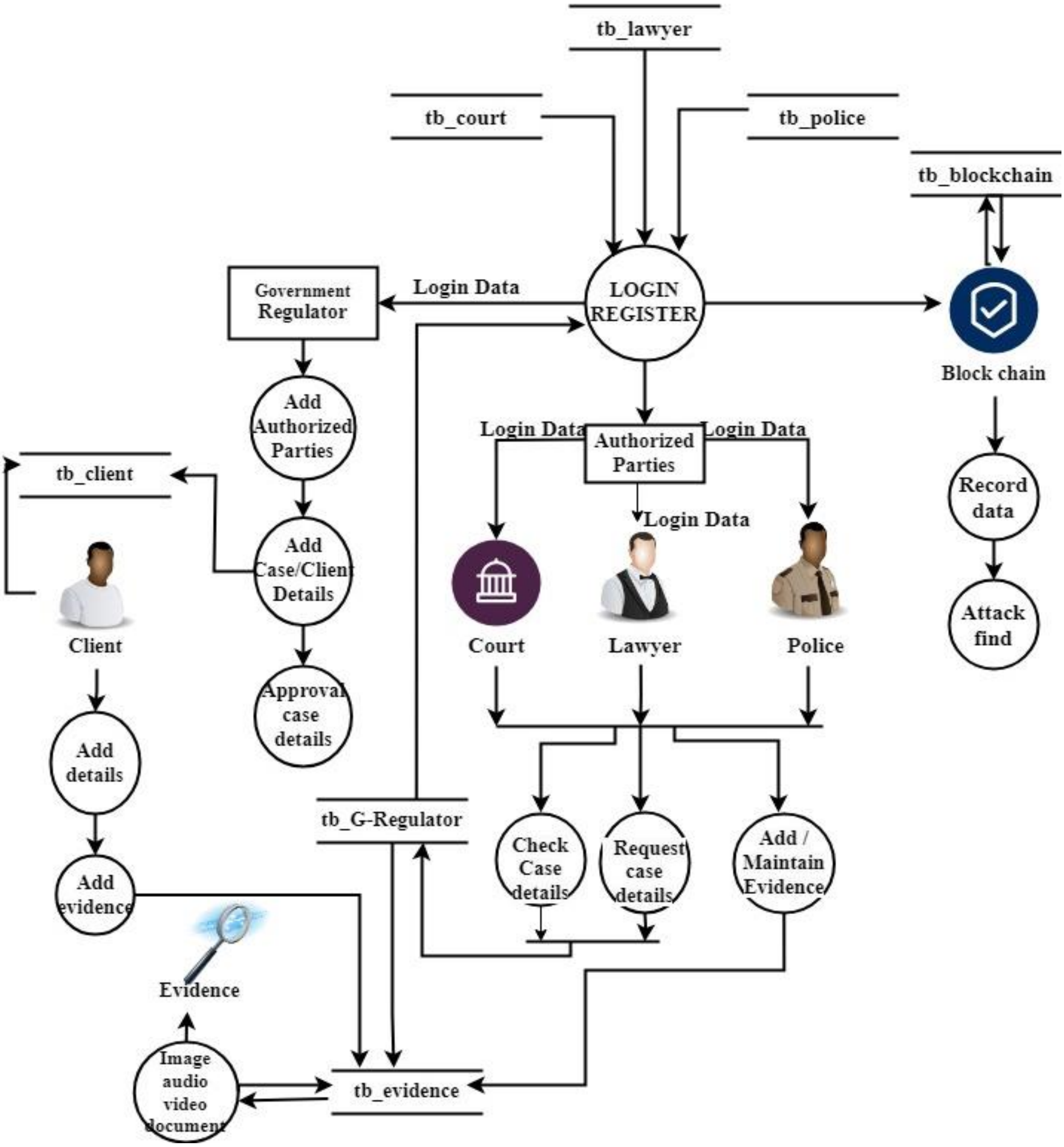
4.5.1. Data flow Diagram level 0

LEVEL 1



4.5.2. Data flow Diagram level 1

LEVEL 2



4.5.3. Data flow Diagram level 2

4.6. UML DIAGRAM

The UML diagrams provide a visual blueprint of the system’s structure and behavior, illustrating the interactions and relationships between users, components, and processes within the digital evidence management platform integrated with blockchain and deep learning tamper detection.

4.6.1 USE CASE DIAGRAM

The Use Case Diagram captures the main actors such as Users, Administrators, and Blockchain Nodes interacting with the system. It highlights key functionalities like evidence submission, tamper detection, blockchain logging, access control, and report generation, showing how users engage with different system features.

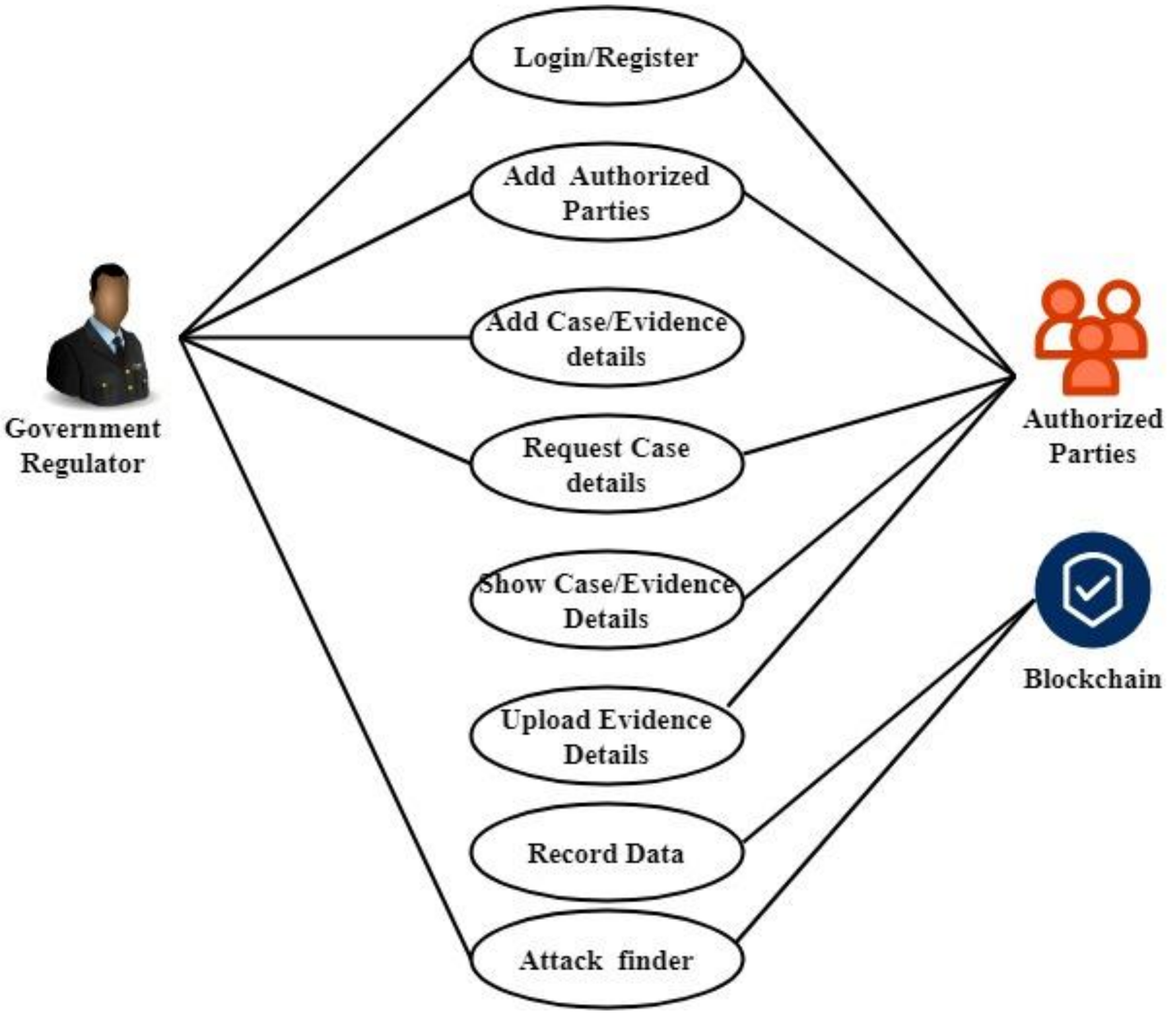


Fig: 4.6.1. USE CASE DIAGRAM

4.6.2 CLASS DIAGRAM

The Class Diagram defines the system’s static structure, detailing classes such as User, Case, DigitalEvidence, BlockchainLog, TamperDetectionResult, and AccessControl. It illustrates attributes, methods, and relationships like inheritance and associations, providing a comprehensive model of system data and behavior.

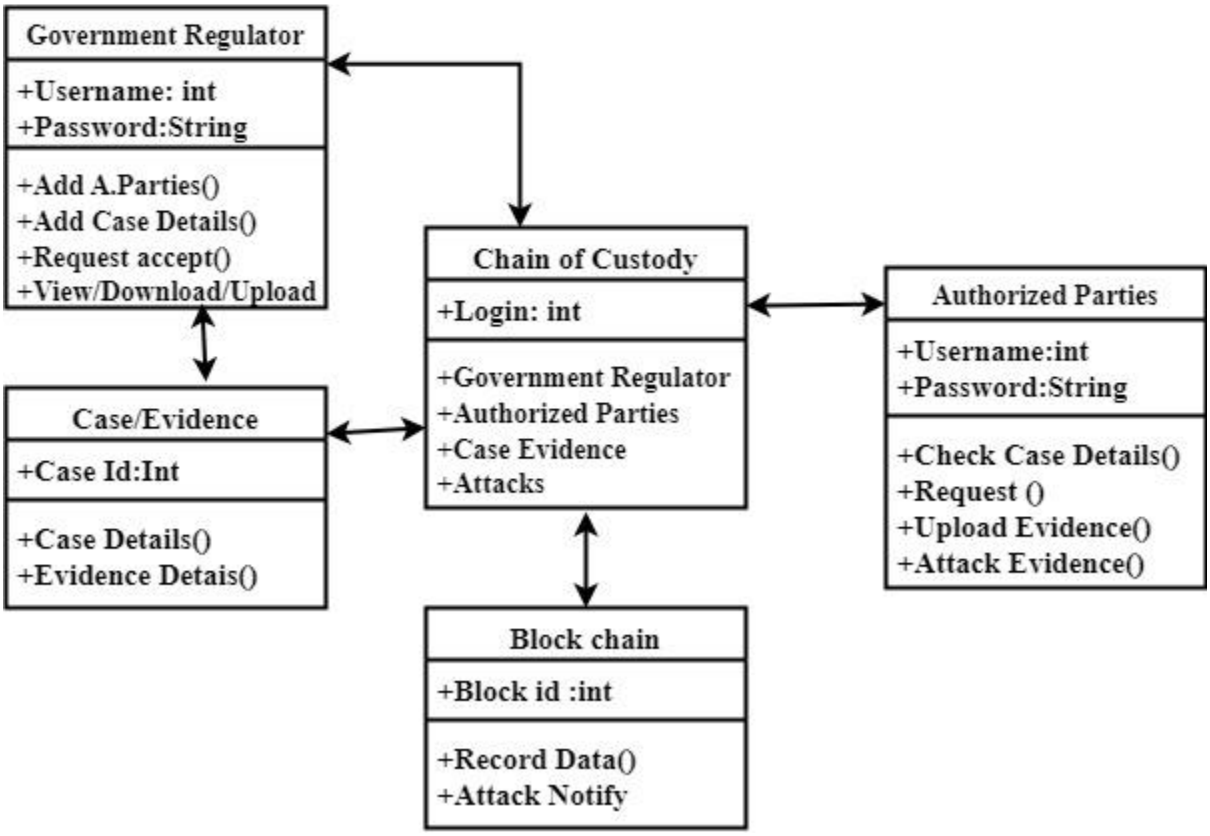


Fig: 4.6.2. CLASS DIAGRAM

4.6.3 SEQUENCE DIAGRAM

The Sequence Diagram shows the time-ordered flow of messages between objects during critical processes, such as evidence submission and tamper detection. It maps how user actions trigger system components like blockchain logging and deep learning analysis, ensuring secure and efficient processing.

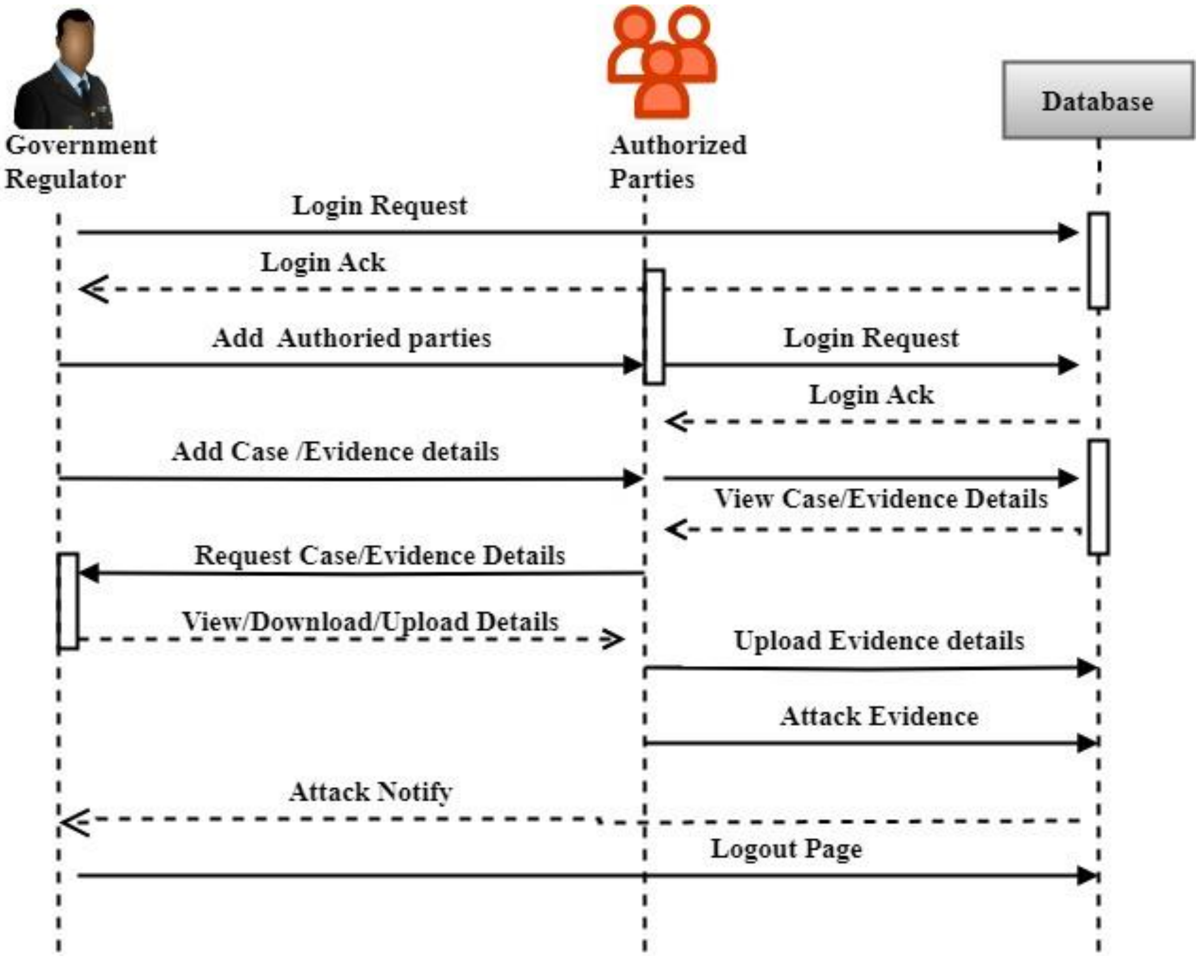


Fig: 4.6.3. SEQUENCE DIAGRAM

4.6.4 ACTIVITY DIAGRAM

The Activity Diagram represents the workflow of major system processes, including steps from evidence upload, verification, tamper analysis, blockchain transaction recording, to access authorization. It visually details decision points and parallel activities in the system’s operation.

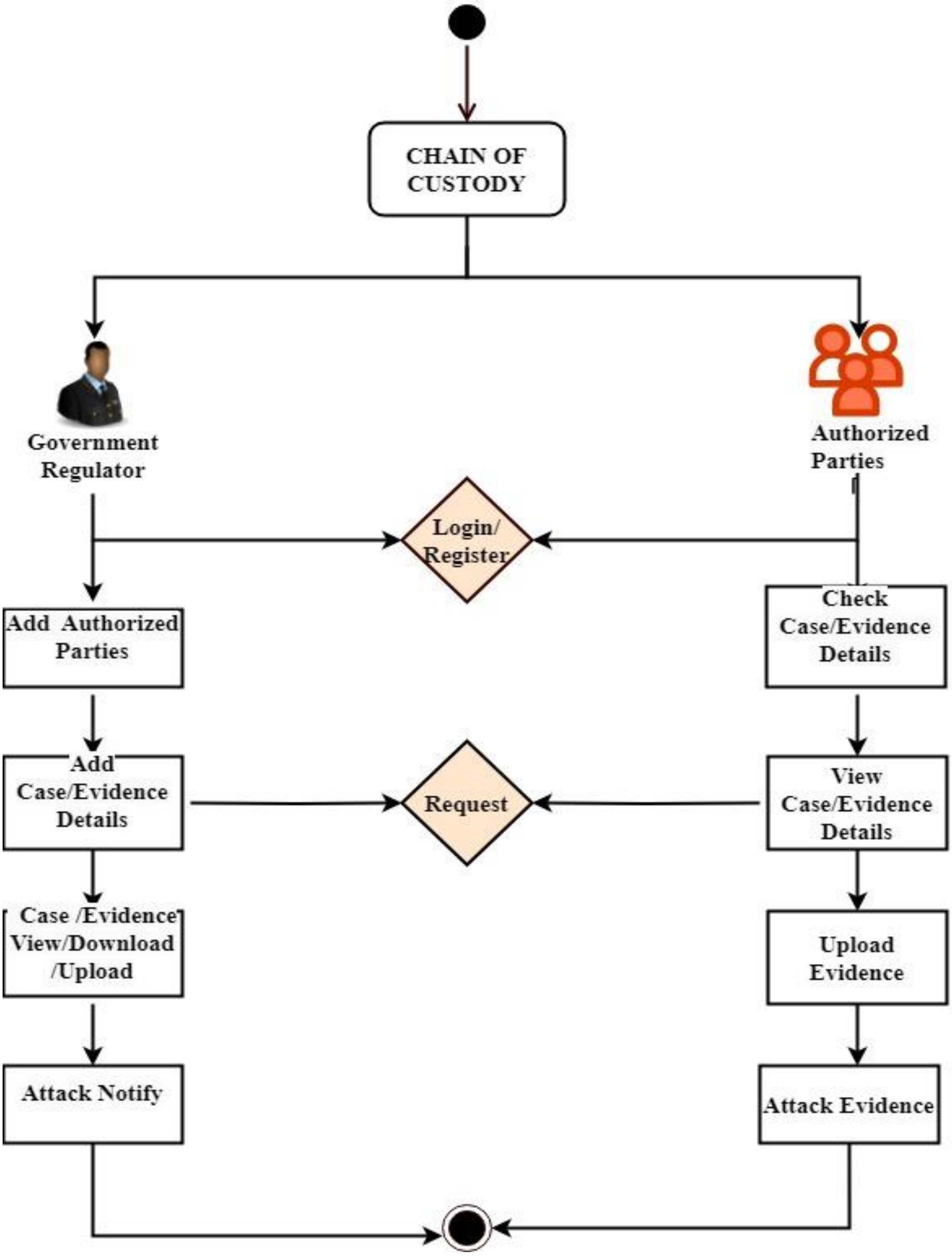


Fig: 4.6.4. ACTIVITY DIAGRAM

4.6.5 COLLABORATION DIAGRAM

The Collaboration Diagram highlights the interactions and relationships between system objects working together to perform functions like tamper detection and evidence tracking. It emphasizes message exchanges within object groups to fulfill use cases effectively.

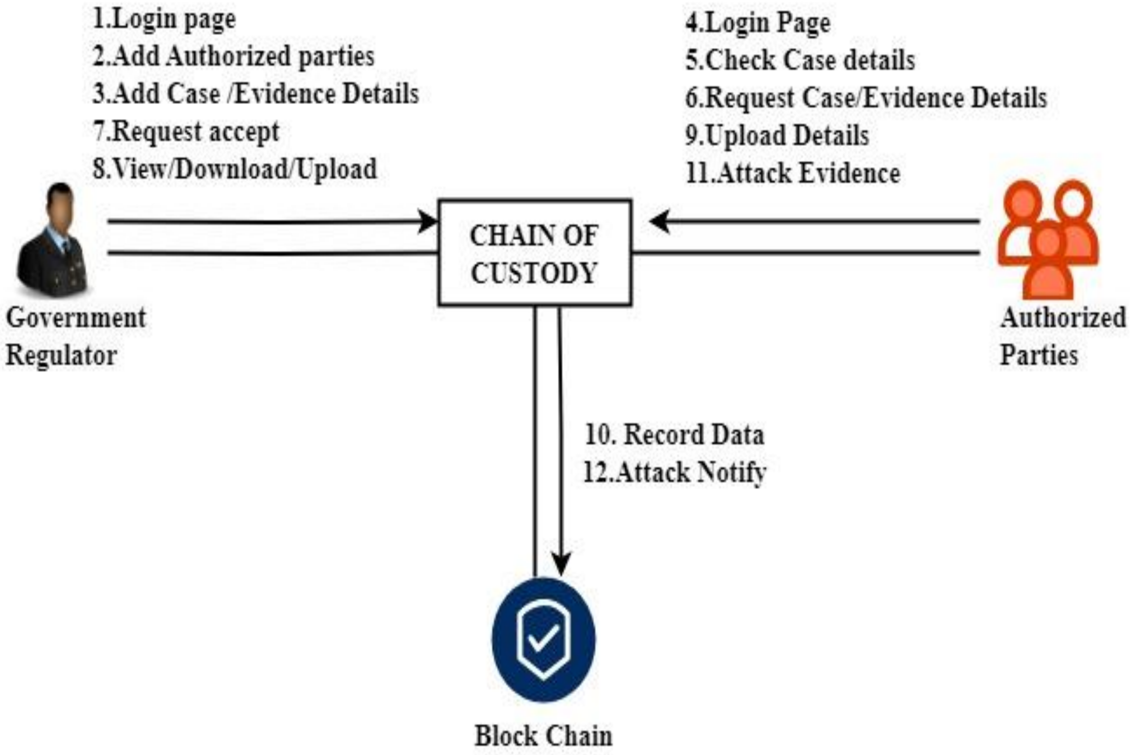


Fig: 4.6.5. COLLABORATION DIAGRAM

4.6.6 DEPLOYMENT DIAGRAM

The Deployment Diagram depicts the high-level organization of software components such as User Interface, Blockchain Module, Deep Learning Tamper Detection Module, Database, and Backend Services. It shows dependencies and how components integrate to form the complete system.

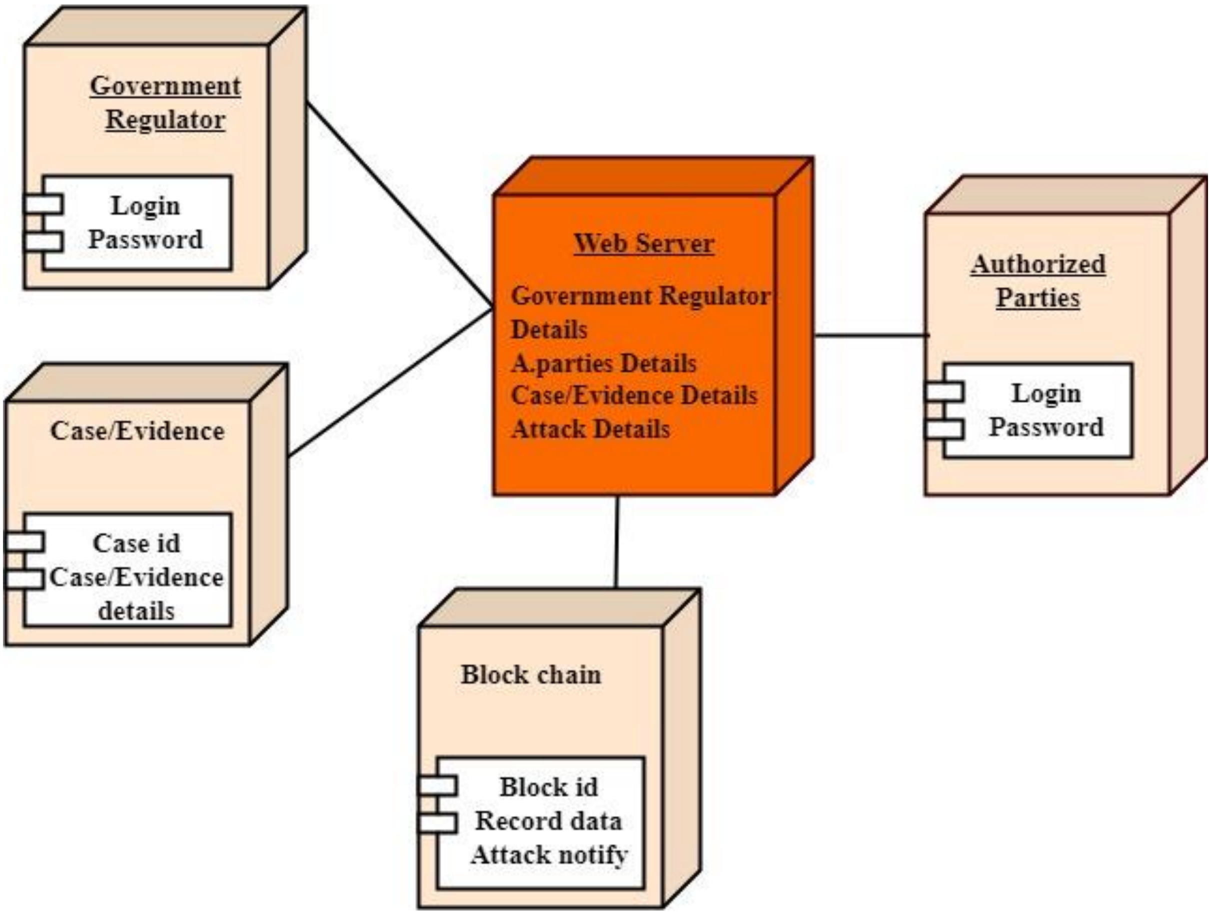


Fig: 4.6.6. DEPLOYMENT DIAGRAM

CHAPTER 5

SOFTWARE DESCRIPTION

5.1. PYTHON 3.7.4

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard library which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia
- Scientific computing
- Text processing and many more.

Pandas

pandas are a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. pandas are a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

Pandas is mainly used for data analysis and associated manipulation of tabular data in Data frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The development of pandas introduced into Python many comparable features of working with Data frames that were established in the R programming language. The panda's library is built upon another library NumPy, which is oriented to efficiently working with arrays instead of the features of working on Data frames.

NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

Scikit Learn

scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

Scikit-learn (formerly scikits. learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

5.2 MYSQL

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company. MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database.

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Ess Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

5.3 WAMPSERVER

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database.

WAMPServer is a reliable web development software program that lets you create web apps with MYSQL database and PHP Apache2. With an intuitive interface, the application features numerous functionalities and makes it the preferred choice of developers from around the world. The software is free to use and doesn't require a payment or subscription.

5.4. BOOTSTRAP 4

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap

Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops

Mobile-first approach: In Bootstrap, mobile-first styles are part of the core framework

Browser compatibility: Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera)

5.5. FLASK

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have formed a validation support. Instead, Flask supports the extensions to add such functionality to the application. Although Flask is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers. Let’s take a closer look into Flask, so-called “micro” framework for Python. Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins.

5.6. JSON

JSON, or JavaScript Object Notation, is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application, as an alternative to XML. Squarespace uses JSON to store and organize site content created with the CMS. **JSON** (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures. JSON is a lightweight format that enables you to share, store, and work with data. As a format, JSON has been experiencing increased support in APIs, including the Twitter API. JSON is also a natural format to use in JavaScript and has many implementations available for use in various popular programming languages. You can read the full language support on the “Introducing JSON” site.

CHAPTER 6

SYSTEM TESTING

6.1. SOFTWARE TESTING

A blockchain is a chain of blocks that contains information. Its technology was originally intended to timestamp digital documents so that it's not possible to backdate or tamper with them. A blockchain technology or platform can be used to secure, store, and manage data in a decentralized and cryptic format. This addresses the current challenges of trust or data breach between B2B, B2C, and C2B entities. It was adopted by Satoshi Nakamoto in 2009 to create Bitcoin – a digital cryptocurrency. Blockchain technology has since then revolutionized the way businesses are conducted. It is at the core of digital currencies and utility tokens that have gone mainstream.

Blockchain Testing

Blockchain is being widely accepted in the industry. With the rise of popularity, we need to be ready to adapt existing testing strategies to blockchain technology. But the lack of best practices, the creation of suitable test data, and dealing with scale, security and performance are some of the key testing challenges in the blockchain. Blockchain testing assists in enabling smart records and ensures fraud security. Data in the blockchain are stored in blocks. Any change in the block will invalidate the subsequent blocks.This makes it important that whenever a new block is added, it is added in the right way. Since it is complex to exploit a blockchain, the testing of blockchain becomes even more complex. Since large transactions go through processes like encryption and decryption, it becomes necessary that these processes go smoothly.

Types of Blockchain Testing

1. **Functional Testing-** Functional testing is the basic testing of components, systems and their functionality, like the addition of a block in the blockchain, block size, chain size, etc. Every new block is added to the chain once the transaction's validity is authenticated.
2. **Integration Testing-** Tests the integration or interfaces between components and different parts of the system. As there are multiple components involved in the blockchain application, integration tests should be done properly and frequently, to test that all the components are properly integrated.
3. **Security Testing-** Security testing is essential for blockchain application debugging, as blockchain is used in highly secure financial, government, or regulatory environments.

4. **Performance Testing-** One of the most important criteria of blockchain applications is speed. the performance is based on the size of the network, and transactions are tested in this type of testing.
5. **Node Testing-** A blockchain’s strength is maintained through consensus across all nodes on the order in which the transactions are added to the network. This consensus protocol needs to be tested to ensure transactions are stored in the proper sequence.
6. **Smart Contracts Testing-** Smart contracts are software modules on the blockchain that automatically execute transactions. SC testing involves making sure that the parties involved in transactions are adhering to the rules.
7. **API Testing-** Based on the application, blockchain can trigger events or external applications. API plays a key role here, and API testing needs to consider the interaction of applications in and out of the blockchain system.

6.2. TEST CASES

User Authentication

- Test Case 1: Verify that users can log in with valid credentials.
- Test Case 2: Verify that users cannot log in with invalid credentials.
- Test Case 3: Verify that users are redirected to the login page when accessing restricted features without authentication.

Evidence Submission

- Test Case 4: Verify that users can submit digital evidence files to the system.
- Test Case 5: Verify that the system accepts various file formats for evidence submission.
- Test Case 6: Verify that evidence submission triggers a transaction on the blockchain ledger.

Tamper Detection

- Test Case 7: Verify that deep learning tamper detection accurately identifies unauthorized modifications or tampering attempts on stored evidence.
- Test Case 8: Verify that users receive alerts or notifications for suspicious activities detected by tamper detection.
- Test Case 9: Verify that tamper detection results are logged and recorded for audit purposes.

Evidence Retrieval

- Test Case 10: Verify that users can search and retrieve digital evidence records based on various criteria such as file name, category, and date of submission.
- Test Case 11: Verify that retrieved evidence files are intact and have not been tampered with.

Chain of Custody Management

- Test Case 12: Verify that the system maintains a verifiable and auditable record of the chain of custody for each piece of digital evidence.
- Test Case 13: Verify that chain of custody transactions are recorded on the blockchain ledger in a tamper-resistant manner.

User Management

- Test Case 14: Verify that administrators can create new user accounts and assign roles and permissions.

- Test Case 15: Verify that users can update their profile information and change their passwords.
- Test Case 16: Verify that inactive user accounts are disabled and cannot access the system.

Integration and Interoperability

- Test Case 17: Verify that Secure Chain can integrate with external systems and technologies for data exchange and interoperability.
- Test Case 18: Verify that data integrity is maintained when exchanging digital evidence between Secure Chain and external systems.

Security and Compliance

- Test Case 19: Verify that data encryption is applied to protect digital evidence stored in the system.
- Test Case 20: Verify that access control mechanisms prevent unauthorized users from accessing sensitive data and features.
- Test Case 21: Verify that Secure Chain complies with relevant regulations and standards governing digital evidence management.

User Interface

- Test Case 22: Verify that the user interface is intuitive and user-friendly, allowing users to navigate through different modules and perform actions efficiently.
- Test Case 23: Verify that the user interface is responsive and compatible with different devices and screen sizes

6.3. TEST REPORT

The report presents the testing outcomes of project integrated with deep learning tamper detection capabilities. The project aims to provide a secure and reliable platform for managing digital evidence across various sectors, including law enforcement, legal proceedings, and cybersecurity investigations.

Test Objective

The objective of this testing is to evaluate the functionality, security, and reliability of project, ensuring that it effectively manages digital evidence, integrates blockchain technology securely, and accurately detects tampering attempts using deep learning algorithms.

Test Scope

- Digital evidence management
- Blockchain integration
- Tamper detection using deep learning
- System architecture
- Security and reliability
- Usability and accessibility
- Scalability and performance
- Integration and interoperability
- Regulatory compliance

Test Environment

The testing environment includes:

- Operating System: Windows 10
- Browser: Google Chrome
- Python: 3.7
- TensorFlow: 2.4.0
- MySQL: 8.0.23
- Blockchain framework

Test Result

Digital evidence management: Successfully managed digital evidence across different sectors.

- **Blockchain integration:** Integrated blockchain technology securely to establish a decentralized and immutable ledger.
- **Tamper detection:** Implemented deep learning algorithms for automatic detection of unauthorized modifications.

- **System architecture:** Designed and developed a robust system architecture for seamless integration.
- **Security and reliability:** Ensured data integrity, authentication, and access control measures.
- **Usability and accessibility:** Designed a user-friendly interface accessible to various stakeholders.
- **Scalability and performance:** Addressed scalability requirements and optimized system performance.
- **Integration and interoperability:** Successfully integrated with existing systems and ensured interoperability.
- **Regulatory compliance:** Adhered to relevant legal and regulatory requirements governing digital evidence management.

Test Conclusion

The project demonstrated satisfactory performance across all tested aspects, providing a secure, reliable, and user-friendly platform for digital evidence management. The system effectively integrates blockchain technology and deep learning tamper detection mechanisms while adhering to regulatory standards. Further refinement and continuous testing may be necessary to enhance features and address any potential issues.

CHAPTER 7

CONCLUSION

7.1. CONCLUSION

In today's ever-growing digital world, we are facing huge challenges in securing our digital infrastructures against different types of cybersecurity incidents. The goal of digital forensics is to perform a structured investigation and maintain a documented chain of evidence to find out exactly what happened on a digital infrastructure network or computing devices involved and who was responsible for it to mitigate and halt such cyber incidents. In conclusion, this project developed a FB – CoC model and a platform to secure Multimedia Forensic Digital Evidence (MFDE) and to ensure the forensic soundness of the stored evidence. The purpose of this DB-CoC is to determine the efficacy of fuzzy hashing algorithms inside blockchain technology, as opposed to conventional cryptographic hash algorithms, in preserving the integrity of digital evidence in image forensics. According to the performance evaluation, fuzzy hash-based blockchains proved to be an effective support for the chain of custody process due to their ability to sustain a realistic workload with a manageable overhead in terms of memory used to store the chain and their ability to handle the chain of custody-related uncertainty. With DB-CoC an investigator does not need to be concerned about verification and authenticity of evidence when performing a digital investigation.

7.2. FUTURE ENHANCEMENT

Future work will consider extending the platform to provide lossless compression and storage optimization, developing novel methods for data ingestion, and well as develop novel methods for MFDE relevance categorization.

- **Enhanced User Experience and Accessibility:**
  Focus on improving the user experience and accessibility of Secure Chain by developing user-friendly interfaces, mobile applications, and browser extensions, and incorporating features such as voice recognition or natural language processing for seamless interaction.
- **Cross-Platform Compatibility:**
  Ensure cross-platform compatibility and support for Secure Chain across different operating systems, devices, and browsers, enabling users to access and utilize the system effectively from anywhere, anytime.
- **Community Engagement and Collaboration:**
  Foster a vibrant user community and establish partnerships with industry stakeholders, law enforcement agencies, legal professionals, academia, and cybersecurity experts to gather feedback, share best practices, and drive innovation in digital evidence management and forensic analysis.

APPENDIX - 1

SOURCE CODEING

Package

```
from flask import Flask, render_template, Response, redirect, request, session, abort, url_for
import os
import base64
from datetime import date
import shutil
import hashlib
import cv2
import imagehash
import mysql.connector
```

Login

```
def login():
cnt=0
act=""
msg=""
if request.method == 'POST':
username1 = request.form['uname']
password1 = request.form['pass']
mycursor = mydb.cursor()
mycursor.execute("SELECT count(*) FROM coc_login where username=%s &&
password=%s",(username1,password1))
myresult = mycursor.fetchone()[0]
if myresult>0:
session['username'] = username1
#result=" You are Logged in sucessfully**"
return redirect(url_for('admin'))
else:
msg="You are logged in fail!!!"
```

Create Authorized Party

```
if request.method=='POST':
```

```
name=request.form['name']
designation=request.form['designation']
mobile=request.form['mobile']
email=request.form['email']
aadhar=request.form['aadhar']
location=request.form['location']
city=request.form['city']
mycursor.execute("SELECT count(*) FROM coc_register where aadhar=%s",(aadhar,))
myresult = mycursor.fetchone()[0]
if myresult==0:
mycursor.execute("SELECT max(id)+1 FROM coc_register")
maxid = mycursor.fetchone()[0]
if maxid is None:
maxid=1
uname="AT"+str(maxid)
p1=randint(1000,9999)
pass1="123456"
now = date.today() #datetime.datetime.now()
rdate=now.strftime("%d-%m-%Y")
sql = "INSERT INTO
coc_register(id,name,designation,mobile,email,aadhar,location,city,uname,pass,status)
VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
val = (maxid,name,designation,mobile,email,aadhar,location,city,uname,pass1,'1')
mycursor.execute(sql, val)
mydb.commit()
print(mycursor.rowcount, "Registered Success")
msg="success"
act="1"
mess="Dear "+name+", Authorized Party - User ID: "+uname+", Password: "+pass1
mycursor.execute('SELECT * FROM coc_register WHERE id=%s', (maxid,))
dd = mycursor.fetchone()
dtime=str(dd[11])
bdata="ID:"+str(maxid)+", User ID:"+uname+", Status:Authorized User Created,
Aadhar:"+aadhar+", Date: "+dtime
```

Case Register

```
if request.method=='POST':
district=request.form['district']
station=request.form['station']
title=request.form['title']
cdate=request.form['cdate']
details=request.form['details']
suspect=request.form['suspect']
name=request.form['name']
fname=request.form['fname']
gender=request.form['gender']
dob=request.form['dob']
address=request.form['address']
district2=request.form['district2']
pincode=request.form['pincode']
mobile=request.form['mobile']
email=request.form['email']
aadhar=request.form['aadhar']
mycursor.execute("SELECT max(id)+1 FROM coc_case")
maxid = mycursor.fetchone()[0]
if maxid is None:
maxid=1
now = date.today() #datetime.datetime.now()
rdate=now.strftime("%d-%m-%Y")
mm=now.strftime("%m")
yy=now.strftime("%Y")
case_id="C"+mm+yy+str(maxid)
sql = "INSERT INTO
coc_case(id,case_id,district,station,title,cdate,details,suspect,name,fname,gender,dob,address,
district2,pincode,mobile,email,aadhar,status) VALUES
(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
val =
(maxid,case_id,district,station,title,cdate,details,suspect,name,fname,gender,dob,address,district2,pincode,mobile,email,aadhar,'0')
```

```
mycursor.execute(sql, val)
mydb.commit()
print(mycursor.rowcount, "Registered Success")
msg="success"
```

Upload Evidence

```
if request.method=='POST':
details=request.form['details']
file=request.files['file']
mycursor.execute("SELECT max(id)+1 FROM coc_evidence")
maxid = mycursor.fetchone()[0]
if maxid is None:
maxid=1
now = date.today() #datetime.datetime.now()
rdate=now.strftime("%d-%m-%Y")
if file:
fname = file.filename
filename = secure_filename(fname)
efile="E"+str(maxid)+filename
file.save(os.path.join("static/upload1", efile))
with open("static/upload1/"+efile, "rb") as image2string:
converted_string = base64.b64encode(image2string.read())
print(converted_string)
bfile1="E"+str(maxid)+".hash"
with open('static/upload/'+bfile1, "wb") as file:
file.write(converted_string)
mm=now.strftime("%m")
yy=now.strftime("%Y")
sql = "INSERT INTO coc_evidence(id,case_id,details,filename,upload_by) VALUES
(%s,%s,%s,%s,%s)"
val = (maxid,case_id,details,efile,'admin')
mycursor.execute(sql, val)
mydb.commit()
print(mycursor.rowcount, "Registered Success")
msg="success"
```

```
mycursor.execute('SELECT * FROM coc_evidence WHERE id=%s', (maxid,))
dd = mycursor.fetchone()
dtime=str(dd[4])
bdata="Evidence ID:"+str(maxid)+", Case ID:"+case_id+", Status: Evidence File: "+efile+",
Upload by admin, Date: "+dtime
```

Allow Permission

```
if request.method=='POST':
user=request.form['user']
ch=request.form.getlist('ch[]')
print(ch)
l=len(ch)
if l==2:
s1="1"
s2="1"
q="View and Upload"
elif l==1:
if ch[0]=="1":
s1="1"
s2="0"
q="View"
else:
s2="1"
s1="1"
q="View and Upload"
mycursor.execute('SELECT count(*) FROM coc_allow WHERE uname=%s &&
case_id=%s', (user,case_id))
c1 = mycursor.fetchone()[0]
if c1==0:
mycursor.execute("SELECT max(id)+1 FROM coc_allow")
maxid = mycursor.fetchone()[0]
if maxid is None:
maxid=1
sql = "INSERT INTO coc_allow(id,uname,case_id,view_st,upload_st) VALUES
(%s,%s,%s,%s,%s)"
```

```
val = (maxid,user,case_id,s1,s2)
mycursor.execute(sql, val)
mydb.commit()
mycursor.execute('SELECT * FROM coc_allow WHERE id=%s', (maxid,))
dd = mycursor.fetchone()
dtime=str(dd[5])
bdata="Allow ID:"+str(maxid)+", Case ID:"+case_id+", Status:Allowed for "+q+",
User:"+user+", Date: "+dtime
```

Send Request

```
mycursor.execute("SELECT count(*) FROM coc_request where cname=%s order by id
desc",(uname,))
cnt2 = mycursor.fetchone()[0]
if cnt2>0:
st2="1"
mycursor.execute("SELECT * FROM coc_request where cname=%s order by id
desc",(uname,))
data3 = mycursor.fetchall()
if request.method == 'POST':
message = request.form['message']
mycursor.execute("SELECT max(id)+1 FROM coc_request")
maxid = mycursor.fetchone()[0]
if maxid is None:
maxid=1
sql = "INSERT INTO coc_request(id,uname,message,reply,status) VALUES
(%s,%s,%s,%s,%s)"
val = (maxid,uname,message,",'0')
mycursor.execute(sql, val)
mydb.commit()
mycursor.execute('SELECT * FROM coc_request WHERE id=%s', (maxid,))
dd = mycursor.fetchone()
dtime=str(dd[5])
bdata="Request ID:"+str(maxid)+", User ID:"+uname+", Status:Request, Date: "+dtime
msg="send"
```

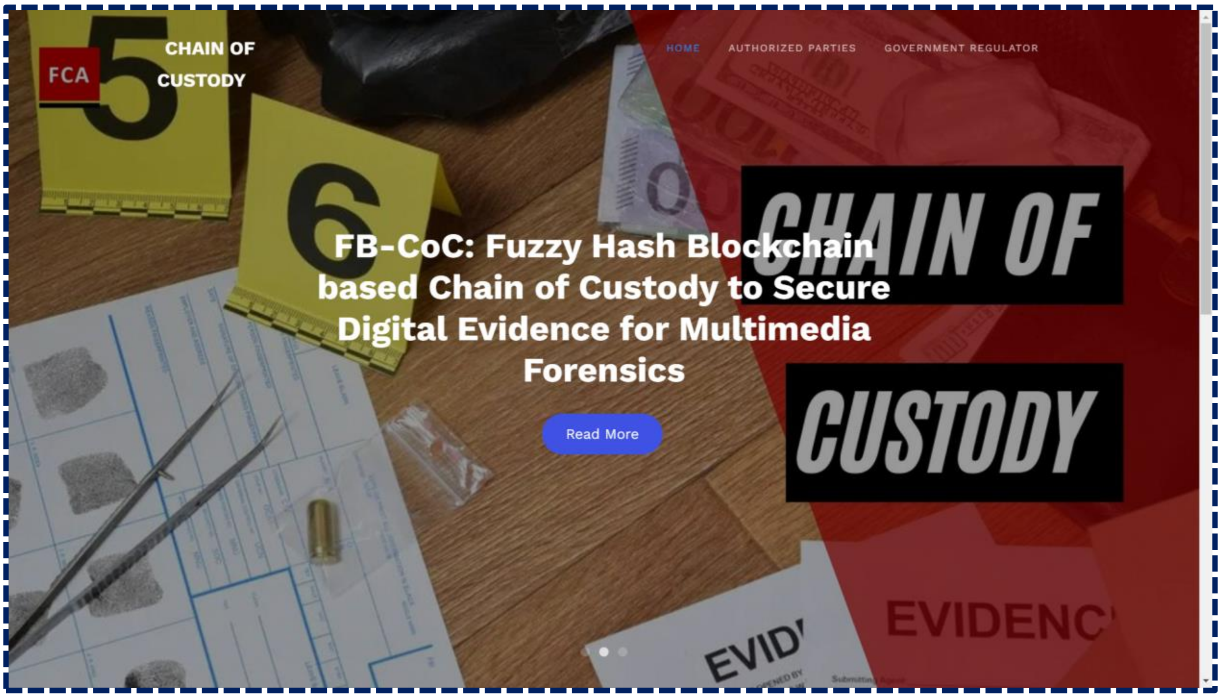
Verification

```
####Fuzzy hash similarity verification##
mycursor.execute('SELECT * FROM coc_evidence WHERE id=%s', (maxid,))
dt = mycursor.fetchall()
cutoff=10
for rr in dt:
hash0 = imagehash.average_hash(Image.open("static/upload1/"+rr[3]))
hash1 = imagehash.average_hash(Image.open("static/upload1/"+efile))
cc1=hash0 - hash1
print("cc="+str(cc1))
if cc1<=cutoff:
ss="ok"
pre_id=str(rr[0])
break
else:
ss="no"
if ss=="ok":
mycursor.execute('SELECT * FROM coc_evidence where id=%s',(maxid,))
sp3 = mycursor.fetchone()
dtime=str(sp3[4])
mycursor.execute('SELECT * FROM coc_evidence where id=%s',(pre_id,))
sp1 = mycursor.fetchone()
pre_user=sp1[6]
mycursor.execute('SELECT * FROM coc_register where uname=%s',(pre_user,))
sp2 = mycursor.fetchone()
pre_vid=sp2[0]
bdata1="ID:"+str(pre_vid)+", Case ID:"+sp3[1]+", Status:Attack Found, Similar Evidence
uploaded by "+uname+", Evidence ID:"+str(maxid)+", File: "+sp3[3]+" (Previous
ID:"+str(pre_id)+"), Date:"+dtime
msg1="attack"
##Download
def down():
eid=request.args.get('eid')
mycursor = mydb.cursor()
mycursor.execute("SELECT * FROM coc_evidence where id=%s",(eid,))
```

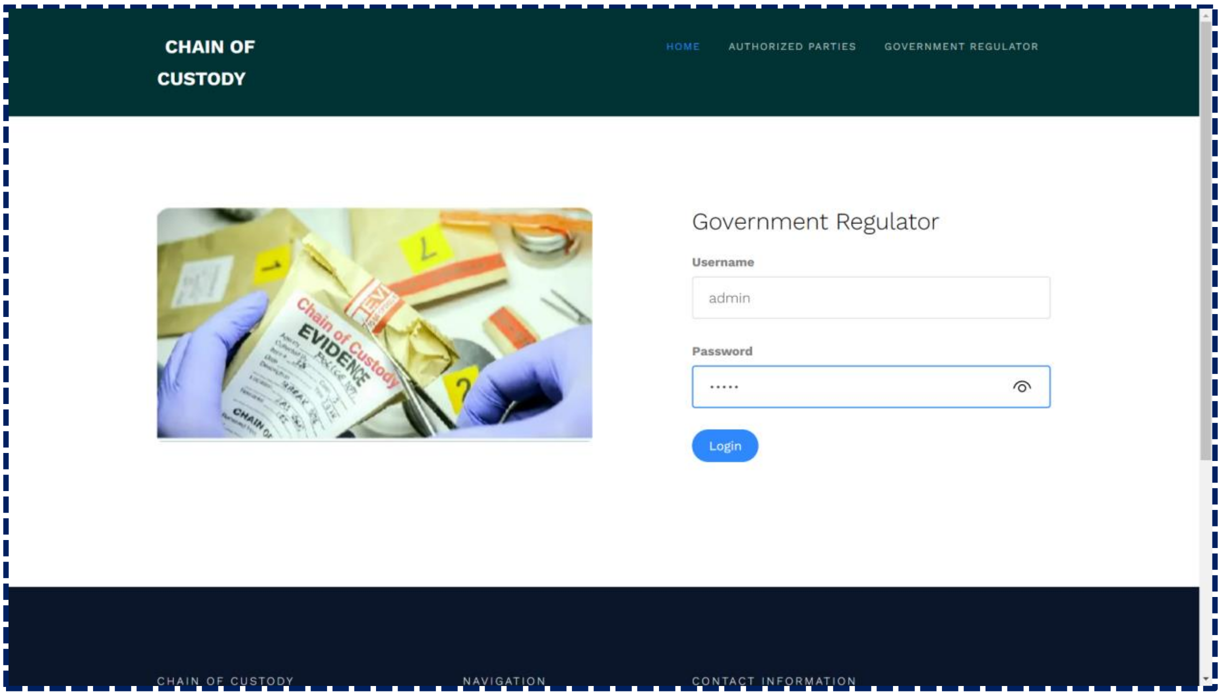
```
data = mycursor.fetchone()
fn=data[3]
ff="E"+eid+".hash"
file = open('static/upload/'+ff, 'rb')
byte = file.read()
file.close()
decodeit = open('static/down/'+fn, 'wb')
decodeit.write(base64.b64decode((byte)))
decodeit.close()
path="static/down/"+fn
return send_file(path, as_attachment=True)
```

APPENDIX -2

SCREEN SHOT



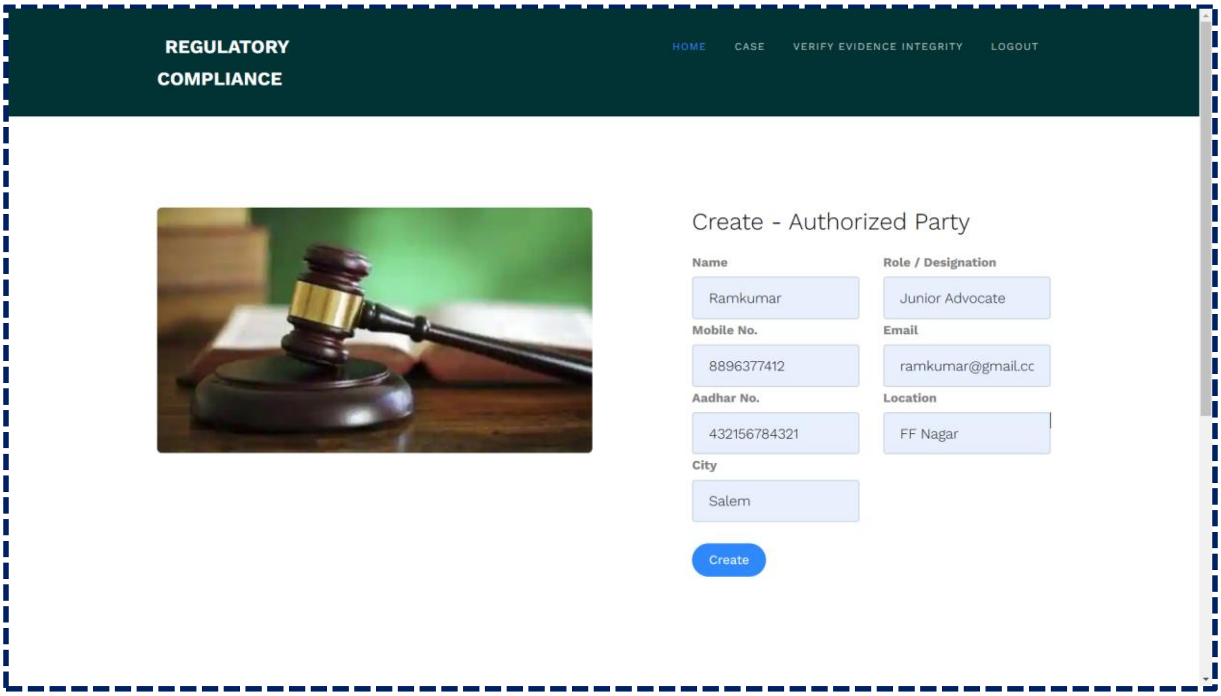
FB-CoC: Fuzzy Hash Blockchain based Chain of Custody to Secure Digital Evidence for Multimedia Forensics



Government Regulator

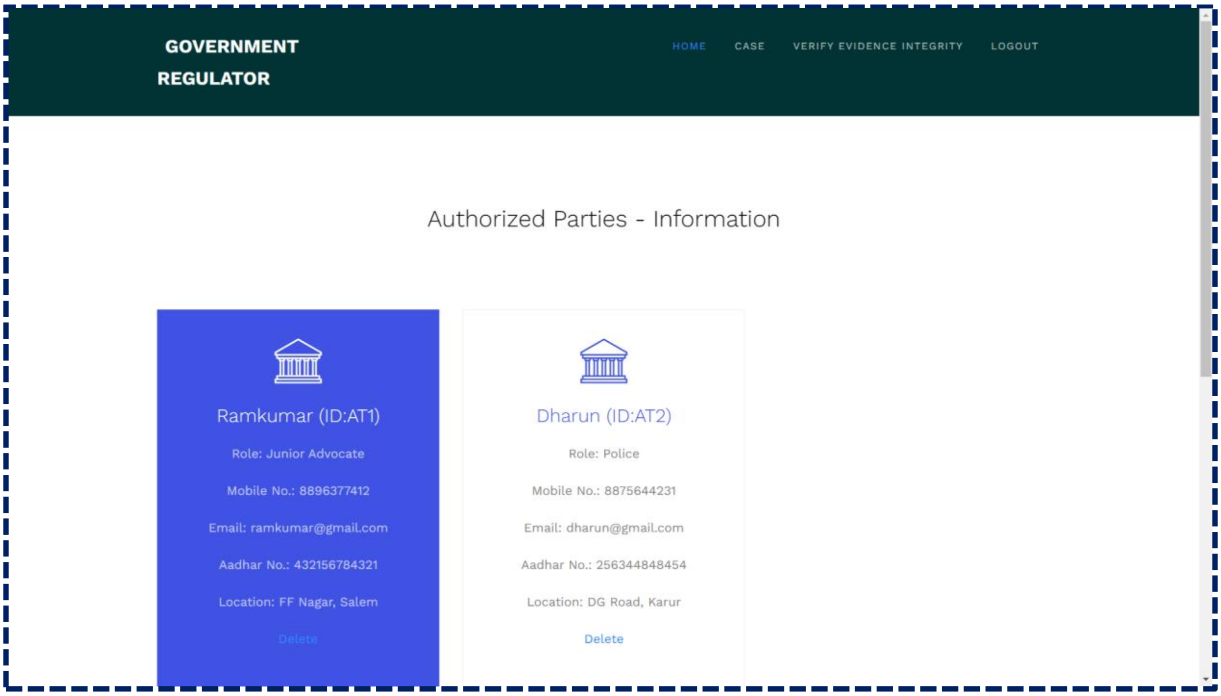
Username: admin

Password: •••••



Create - Authorized Party

Name Ramkumar	Role / Designation Junior Advocate
Mobile No. 8896377412	Email ramkumar@gmail.cc
Aadhar No. 432156784321	Location FF Nagar
City Salem	



Authorized Parties - Information

Ramkumar (ID:AT1)

Role: Junior Advocate

Mobile No.: 8896377412

Email: ramkumar@gmail.com

Aadhar No.: 432156784321

Location: FF Nagar, Salem

Dharun (ID:AT2)

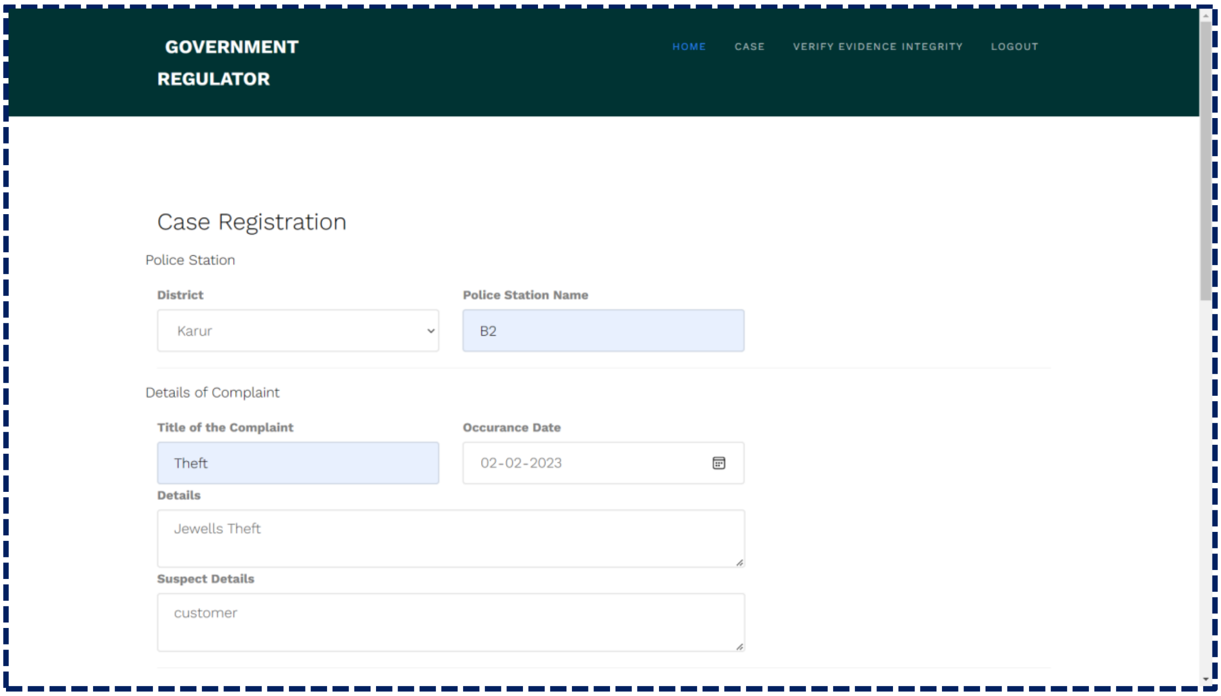
Role: Police

Mobile No.: 8875644231

Email: dharun@gmail.com

Aadhar No.: 256344848454

Location: DG Road, Karur



Case Registration

Police Station

District: Karur

Police Station Name: B2

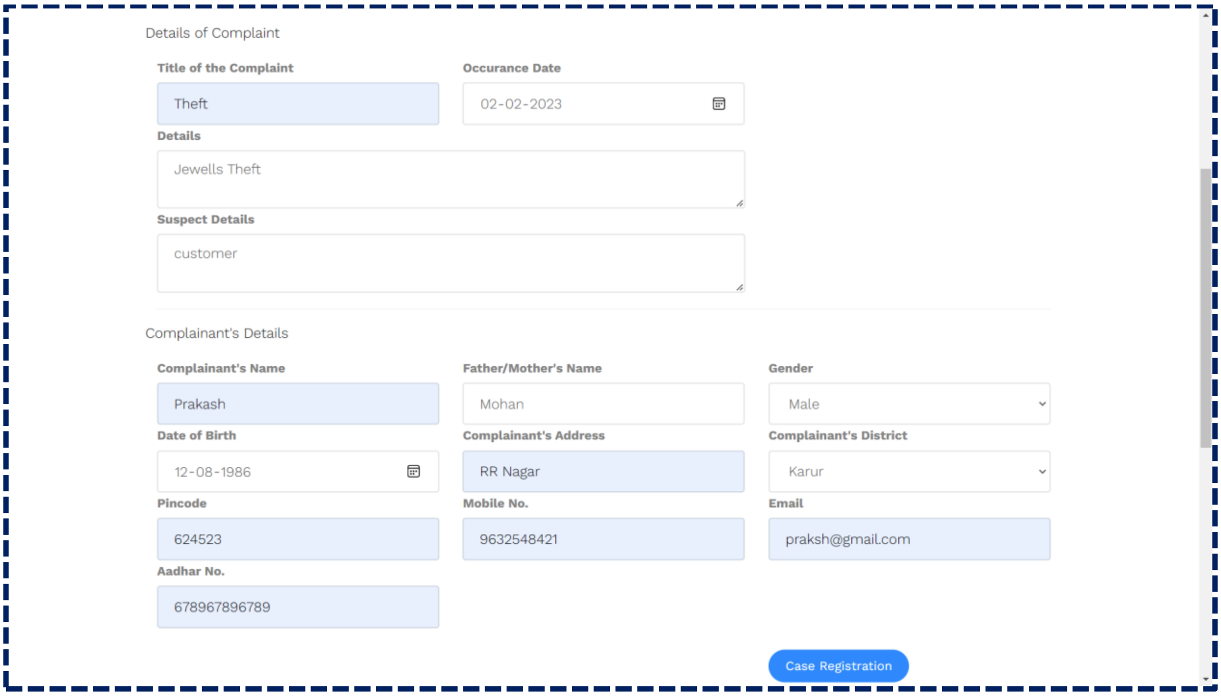
Details of Complaint

Title of the Complaint: Theft

Occurance Date: 02-02-2023

Details: Jewells Theft

Suspect Details: customer



Details of Complaint

Title of the Complaint: Theft

Occurance Date: 02-02-2023

Details: Jewells Theft

Suspect Details: customer

Complainant's Details

Complainant's Name: Prakash

Father/Mother's Name: Mohan

Gender: Male

Date of Birth: 12-08-1986

Complainant's Address: RR Nagar

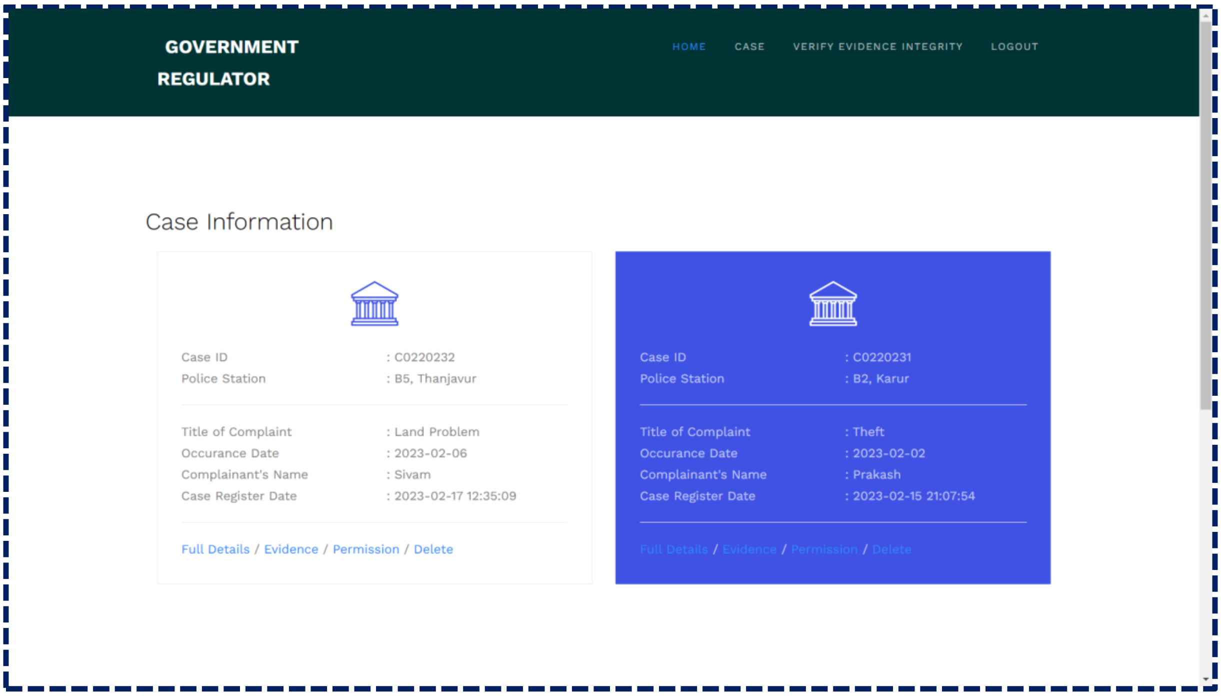
Complainant's District: Karur

Pincode: 624523

Mobile No.: 9632548421

Email: praksh@gmail.com

Aadhar No.: 678967896789


GOVERNMENT
REGULATOR
HOME
CASE
VERIFY EVIDENCE INTEGRITY
LOGOUT

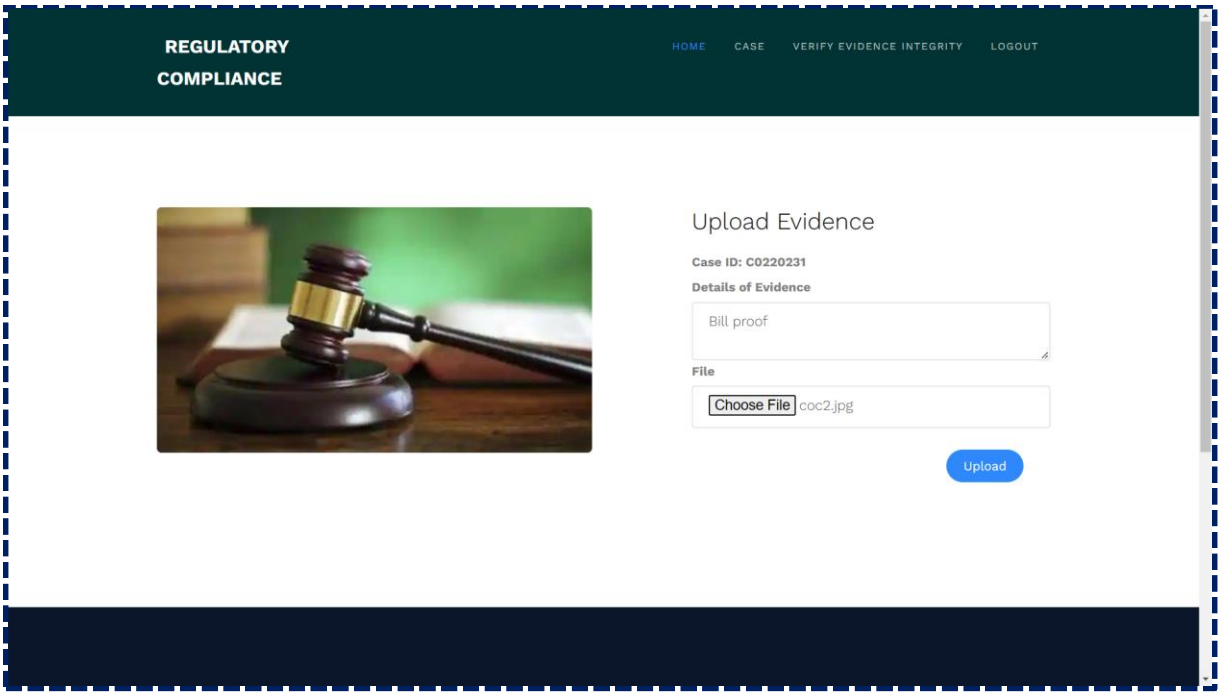
Case Information

Case ID	: C0220232
Police Station	: B5, Thanjavur
Title of Complaint	: Land Problem
Occurance Date	: 2023-02-06
Complainant's Name	: Sivam
Case Register Date	: 2023-02-17 12:35:09

Full Details / Evidence / Permission / Delete

Case ID	: C0220231
Police Station	: B2, Karur
Title of Complaint	: Theft
Occurance Date	: 2023-02-02
Complainant's Name	: Prakash
Case Register Date	: 2023-02-15 21:07:54

Full Details / Evidence / Permission / Delete



Upload Evidence

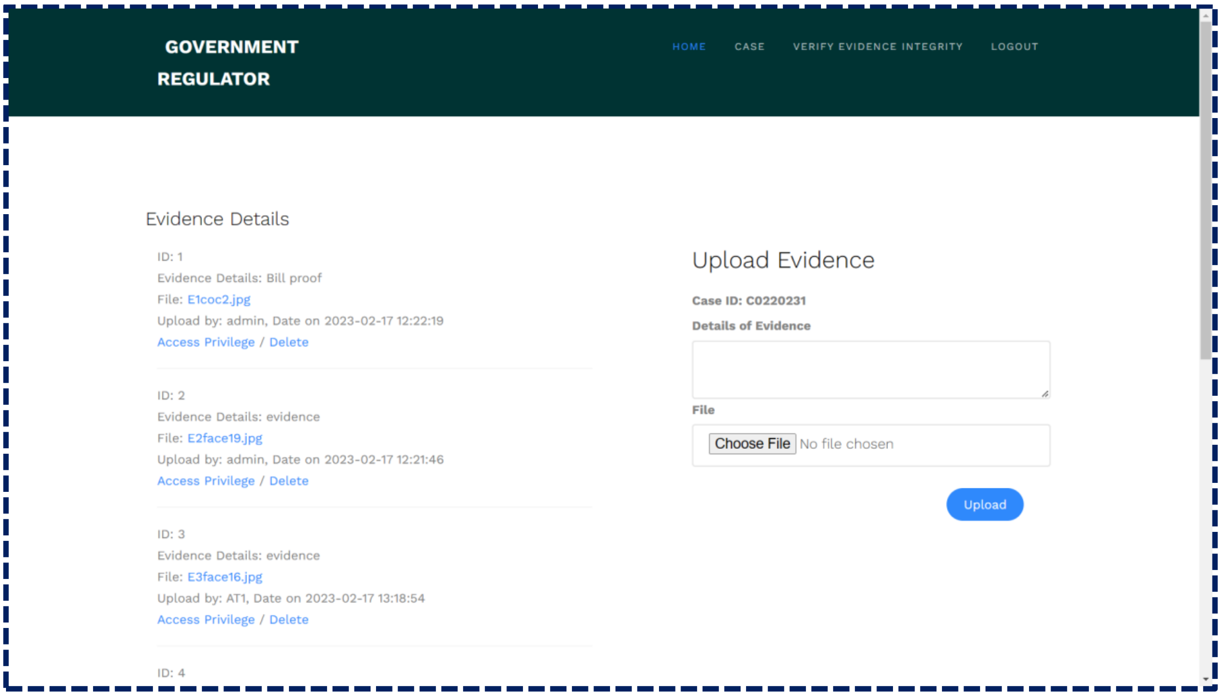
Case ID: C0220231

Details of Evidence

Bill proof

File

Choose File coc2.jpg



Evidence Details

ID: 1
Evidence Details: Bill proof
File: E1coc2.jpg
Upload by: admin, Date on 2023-02-17 12:22:19
Access Privilege / Delete

ID: 2
Evidence Details: evidence
File: E2face19.jpg
Upload by: admin, Date on 2023-02-17 12:21:46
Access Privilege / Delete

ID: 3
Evidence Details: evidence
File: E3face16.jpg
Upload by: AT1, Date on 2023-02-17 13:18:54
Access Privilege / Delete

ID: 4

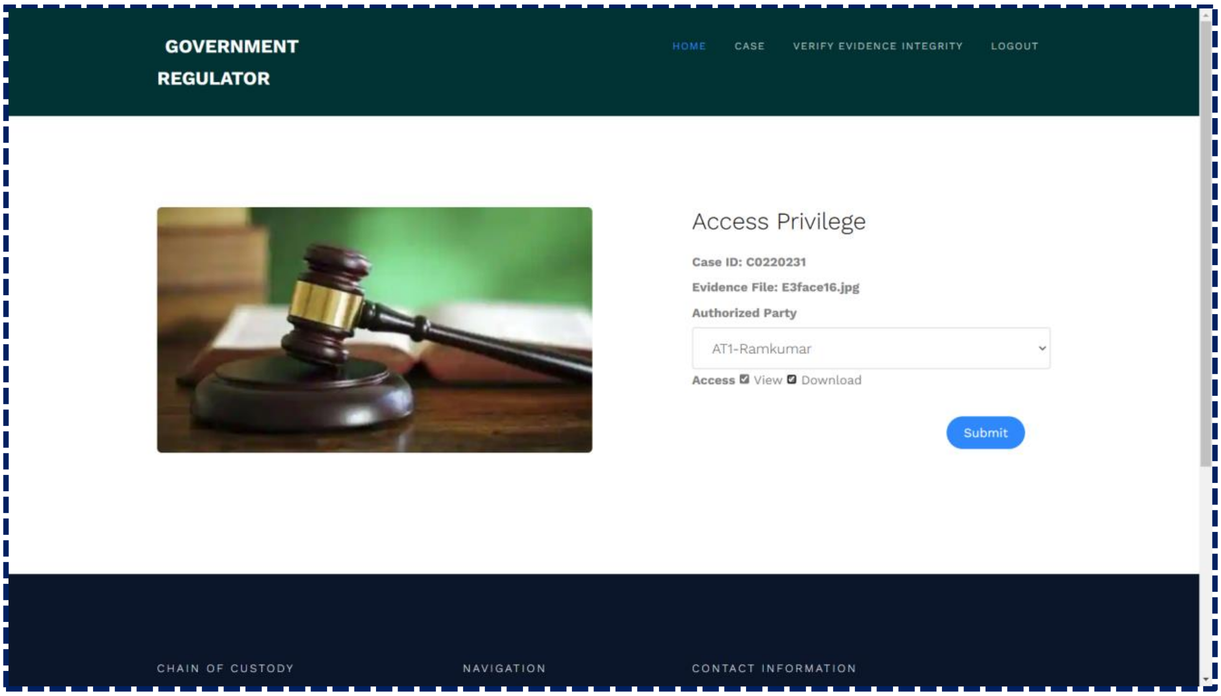
Upload Evidence

Case ID: C0220231

Details of Evidence

File

No file chosen



Access Privilege

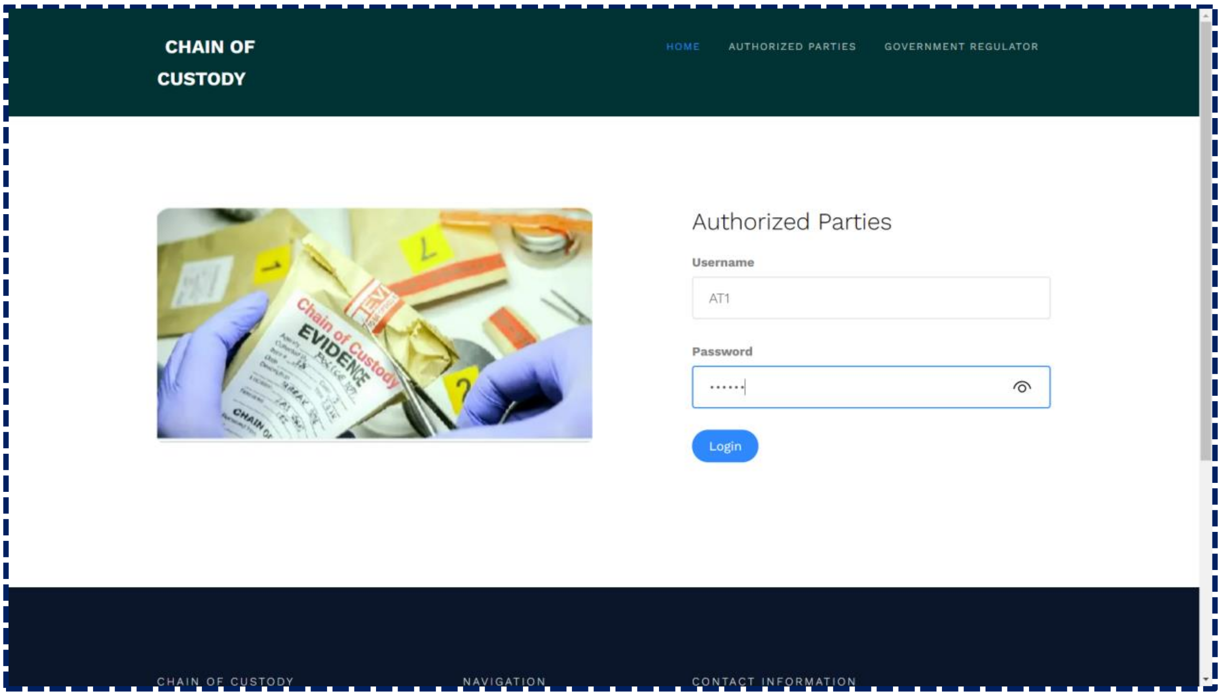
Case ID: C0220231

Evidence File: E3face16.jpg

Authorized Party

AT1-Ramkumar

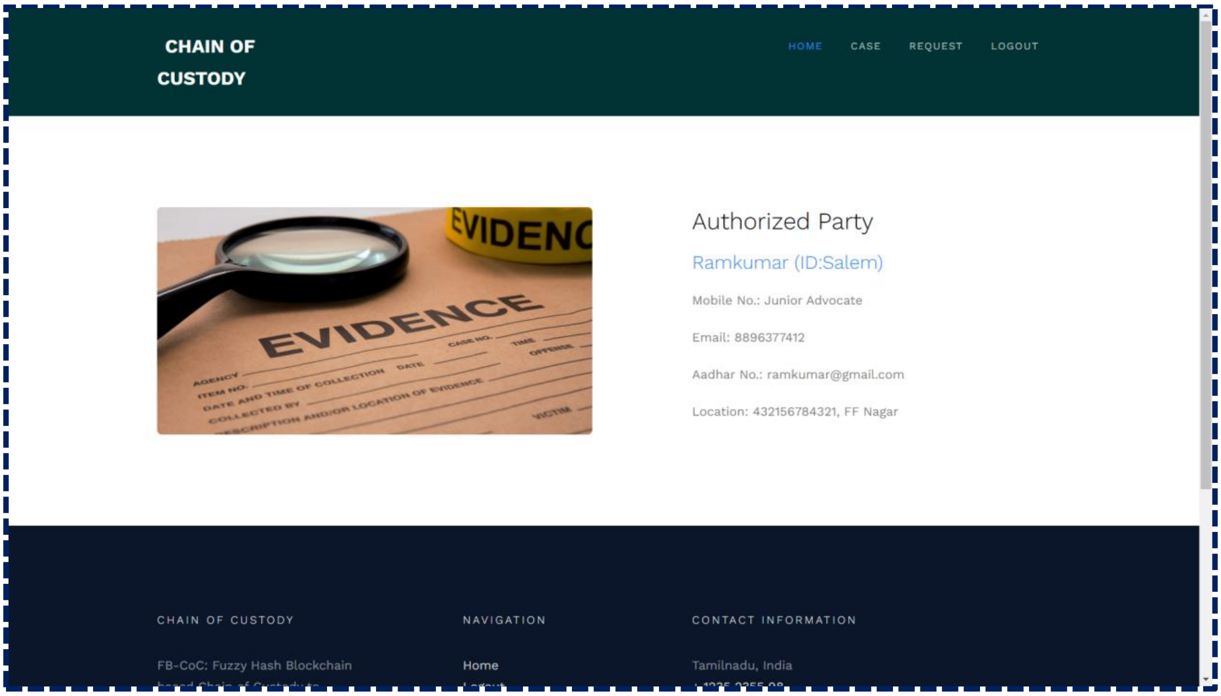
Access ☑ View ☑ Download



Authorized Parties

Username: AT1

Password: ••••••



Authorized Party

Ramkumar (ID:Salem)

Mobile No.: Junior Advocate

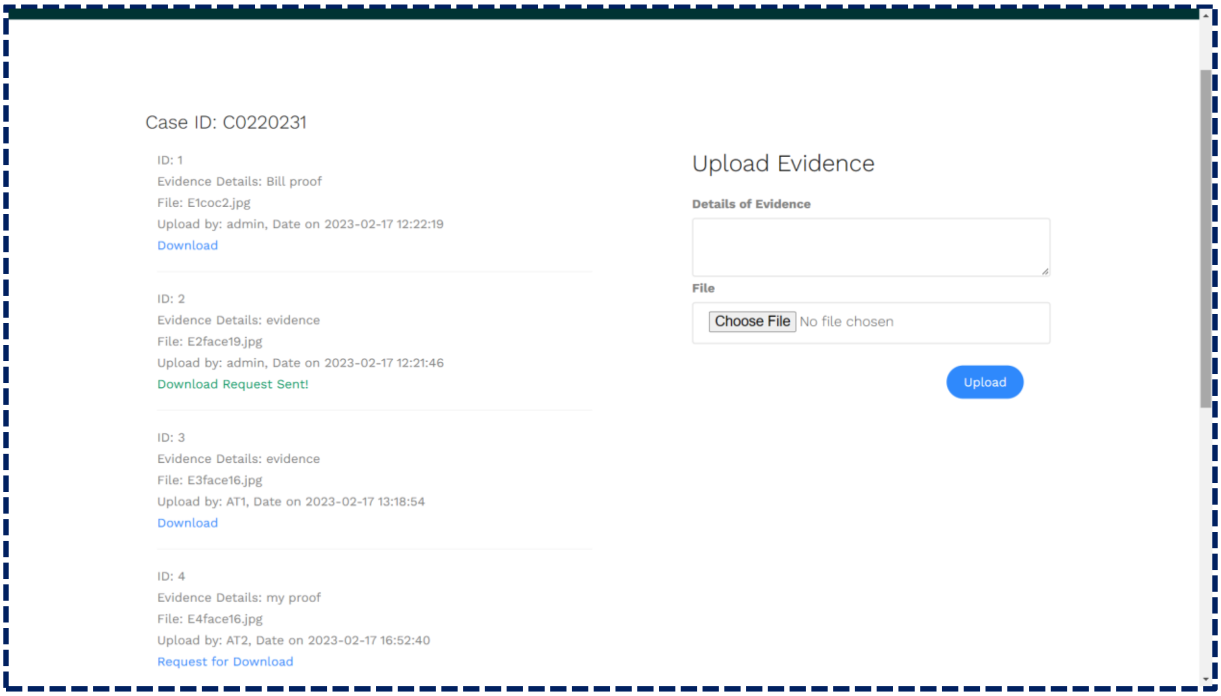
Email: 8896377412

Aadhar No.: ramkumar@gmail.com

Location: 432156784321, FF Nagar

FB-CoC: Fuzzy Hash Blockchain

Tamilnadu, India



Case ID: C0220231

ID: 1
Evidence Details: Bill proof
File: E1coc2.jpg
Upload by: admin, Date on 2023-02-17 12:22:19
Download

ID: 2
Evidence Details: evidence
File: E2face19.jpg
Upload by: admin, Date on 2023-02-17 12:21:46
Download Request Sent!

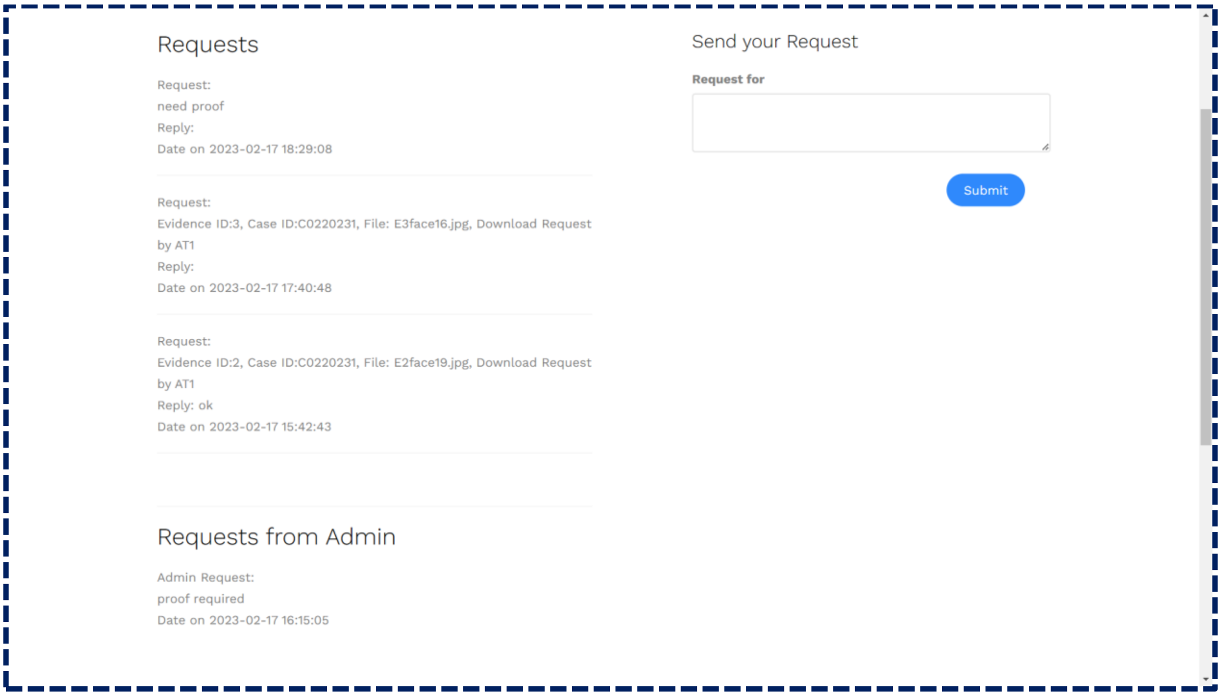
ID: 3
Evidence Details: evidence
File: E3face16.jpg
Upload by: AT1, Date on 2023-02-17 13:18:54
Download

ID: 4
Evidence Details: my proof
File: E4face16.jpg
Upload by: AT2, Date on 2023-02-17 16:52:40
Request for Download

Upload Evidence

Details of Evidence

File



Requests

Request:
need proof
Reply:
Date on 2023-02-17 18:29:08

Request:
Evidence ID:3, Case ID:C0220231, File: E3face16.jpg, Download Request by AT1
Reply:
Date on 2023-02-17 17:40:48

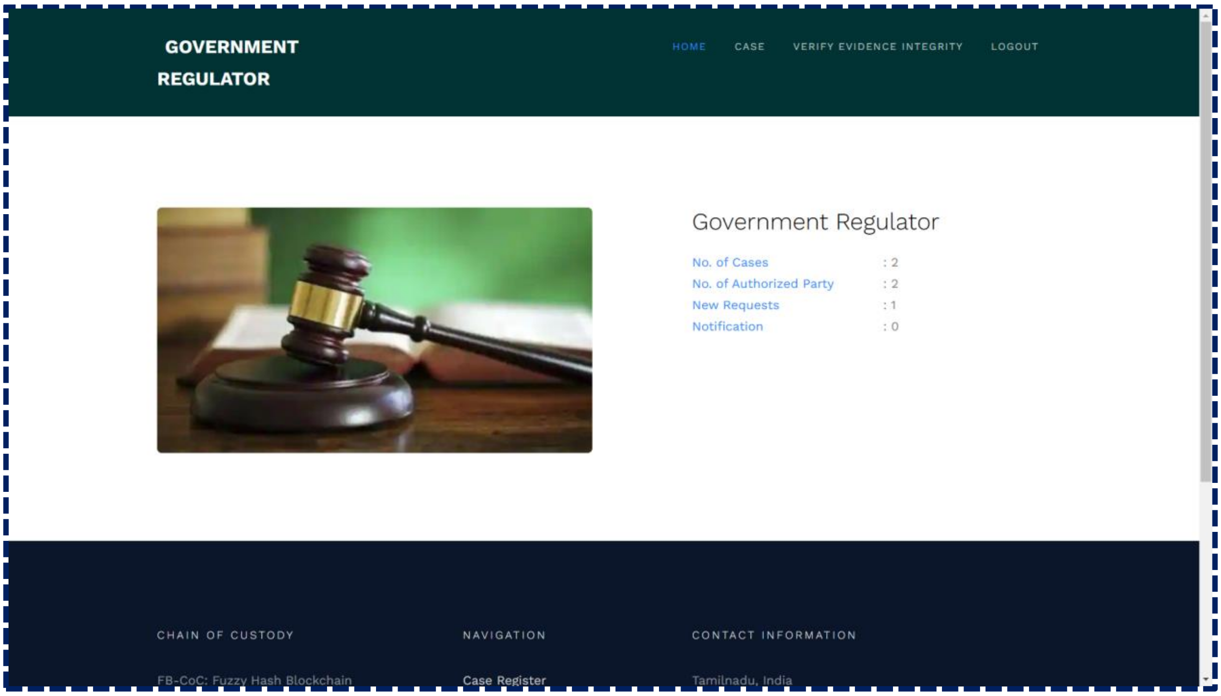
Request:
Evidence ID:2, Case ID:C0220231, File: E2face19.jpg, Download Request by AT1
Reply: ok
Date on 2023-02-17 15:42:43

Requests from Admin

Admin Request:
proof required
Date on 2023-02-17 16:15:05

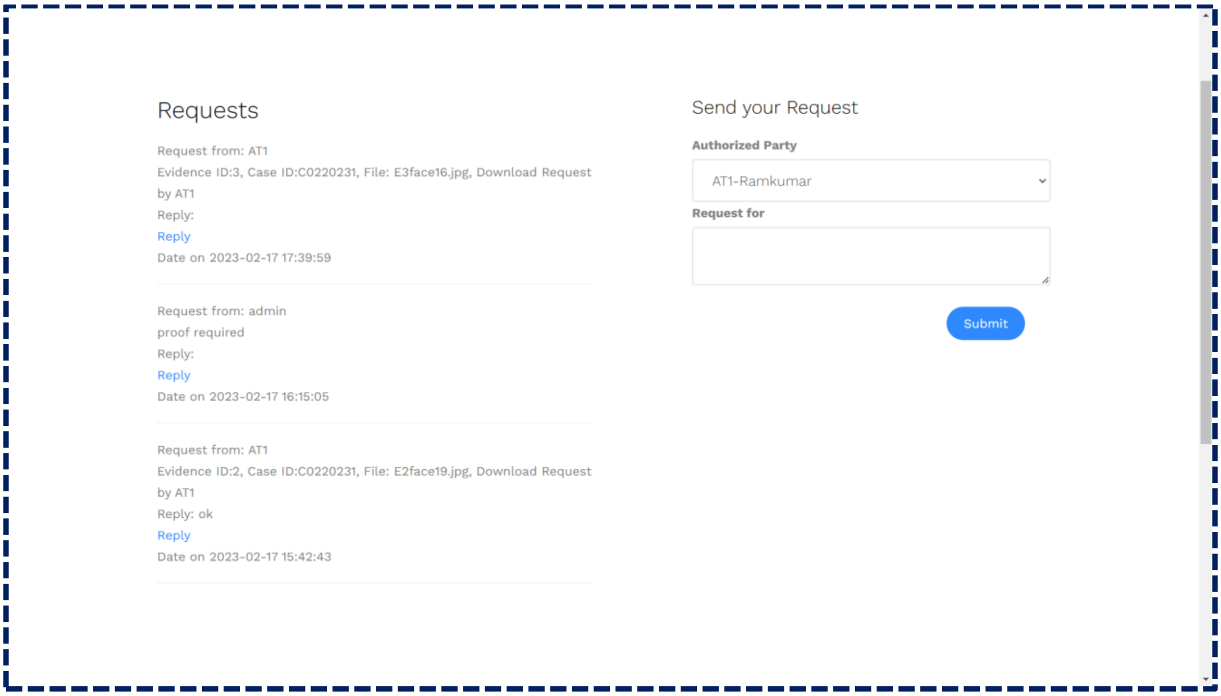
Send your Request

Request for



Government Regulator

No. of Cases	: 2
No. of Authorized Party	: 2
New Requests	: 1
Notification	: 0



Requests

Request from: AT1
Evidence ID:3, Case ID:C0220231, File: E3face16.jpg, Download Request by AT1
Reply:

Date on 2023-02-17 17:39:59

Request from: admin
proof required
Reply:

Date on 2023-02-17 16:15:05

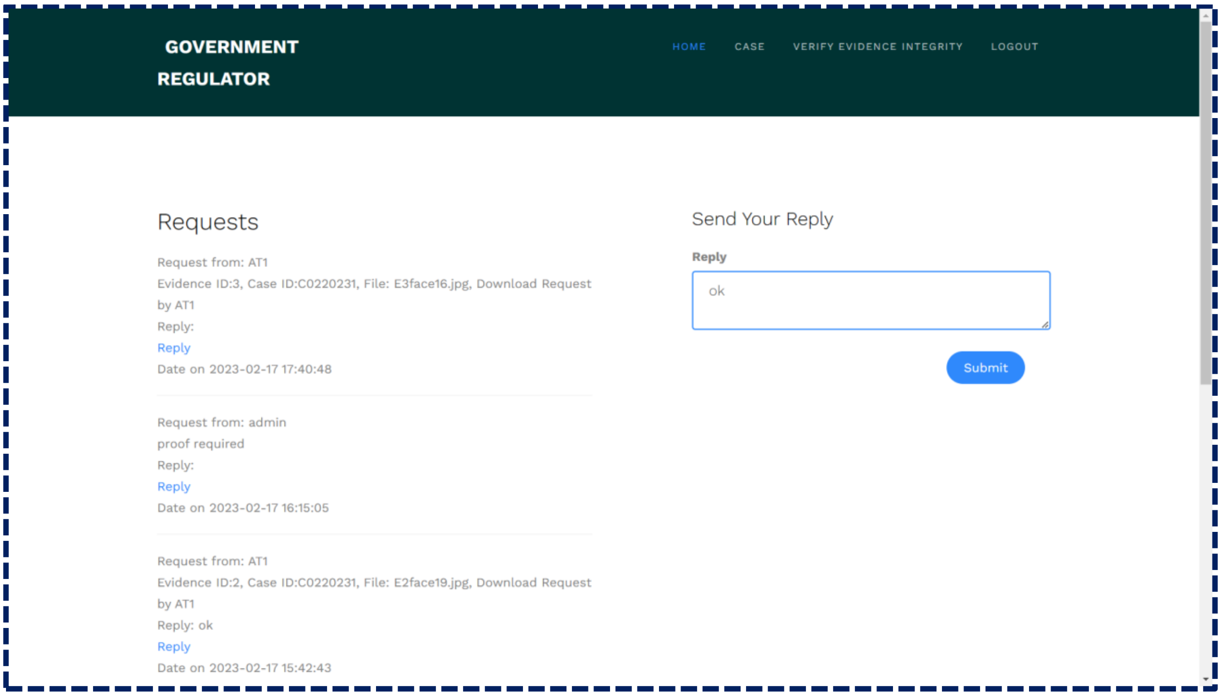
Request from: AT1
Evidence ID:2, Case ID:C0220231, File: E2face19.jpg, Download Request by AT1
Reply: ok

Date on 2023-02-17 15:42:43

Send your Request

Authorized Party: AT1-Ramkumar

Request for:



Requests

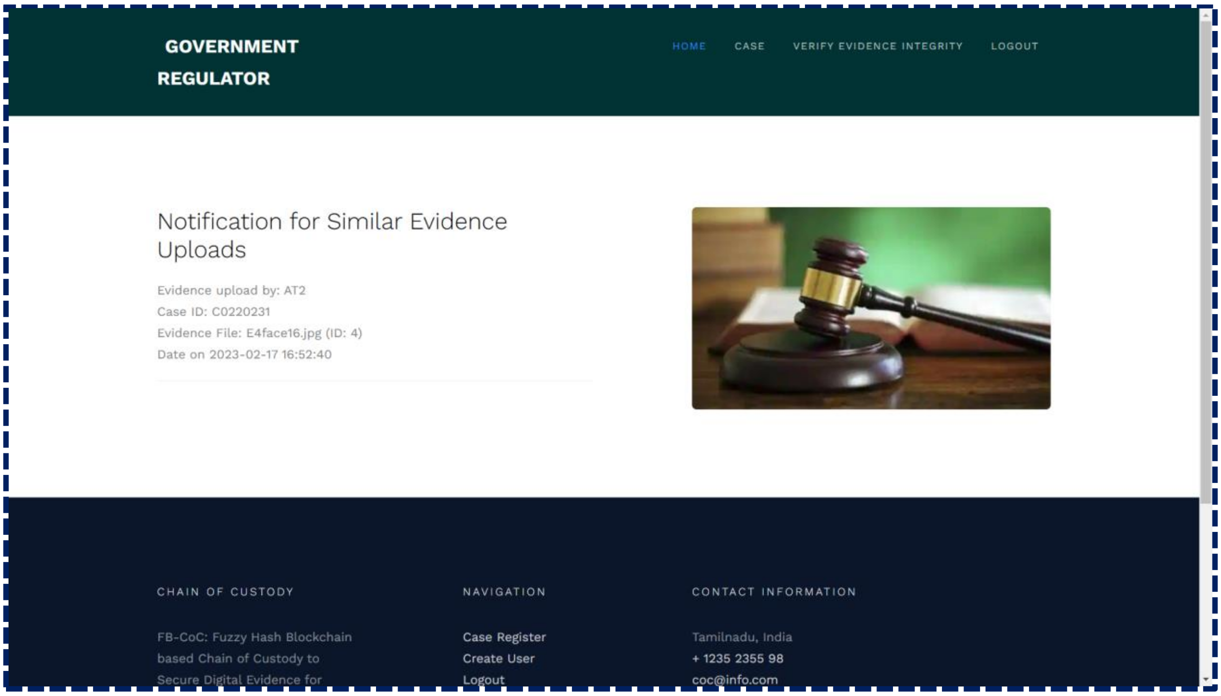
Request from: AT1
Evidence ID:3, Case ID:C0220231, File: E3face16.jpg, Download Request by AT1
Reply:
Reply
Date on 2023-02-17 17:40:48

Request from: admin
proof required
Reply:
Reply
Date on 2023-02-17 16:15:05

Request from: AT1
Evidence ID:2, Case ID:C0220231, File: E2face19.jpg, Download Request by AT1
Reply: ok
Reply
Date on 2023-02-17 15:42:43

Send Your Reply

Reply: ok



Notification for Similar Evidence Uploads

Evidence upload by: AT2
Case ID: C0220231
Evidence File: E4face16.jpg (ID: 4)
Date on 2023-02-17 16:52:40

CHAIN OF CUSTODY

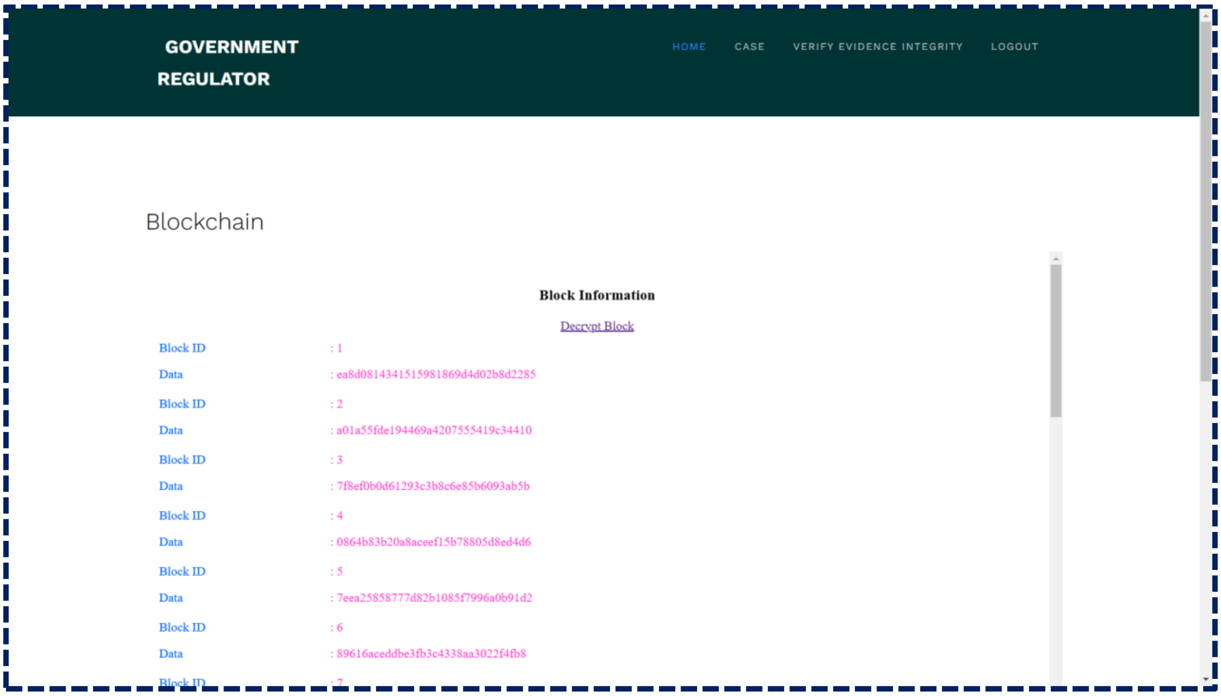
FB-CoC: Fuzzy Hash Blockchain based Chain of Custody to Secure Digital Evidence for

NAVIGATION

Case Register
Create User
Logout

CONTACT INFORMATION

Tamilnadu, India
+ 1235 2355 98
coc@info.com

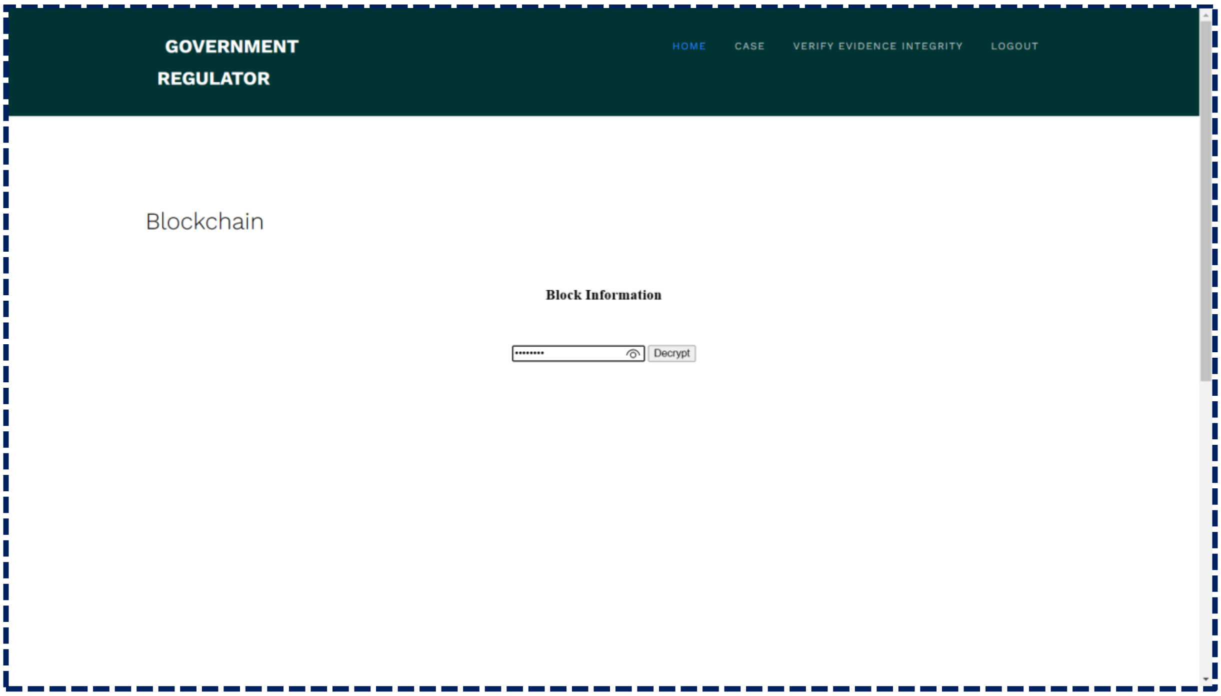


Blockchain

Block Information

Decrypt Block

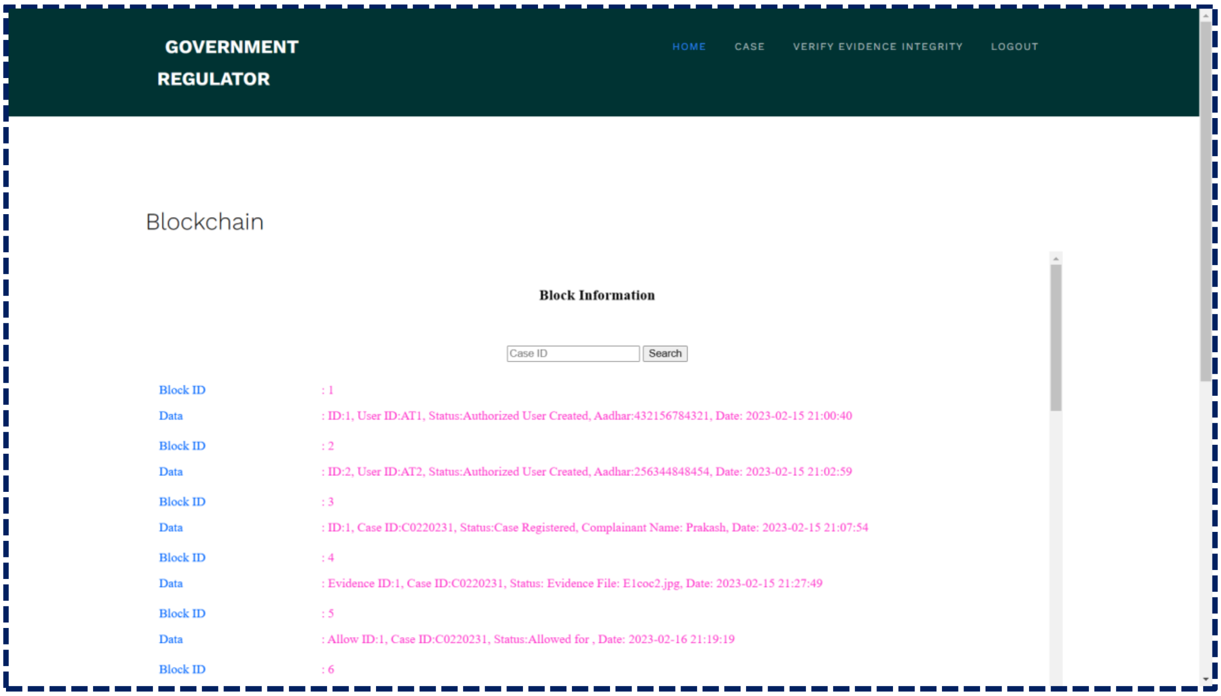
Block ID	: 1
Data	: ea8d0814341515981869d4d02b8d2285
Block ID	: 2
Data	: a01a55fde194469a4207555419c34410
Block ID	: 3
Data	: 7f8ef0b0d61293c3b8c6e85b6093ab5b
Block ID	: 4
Data	: 0864b83b20a8aceef15b78805d8ed4d6
Block ID	: 5
Data	: 7eea25858777d82b1085f7996a0b91d2
Block ID	: 6
Data	: 89616aceddbe3fb3c4338aa3022f4fb8
Block ID	: 7



Blockchain

Block Information

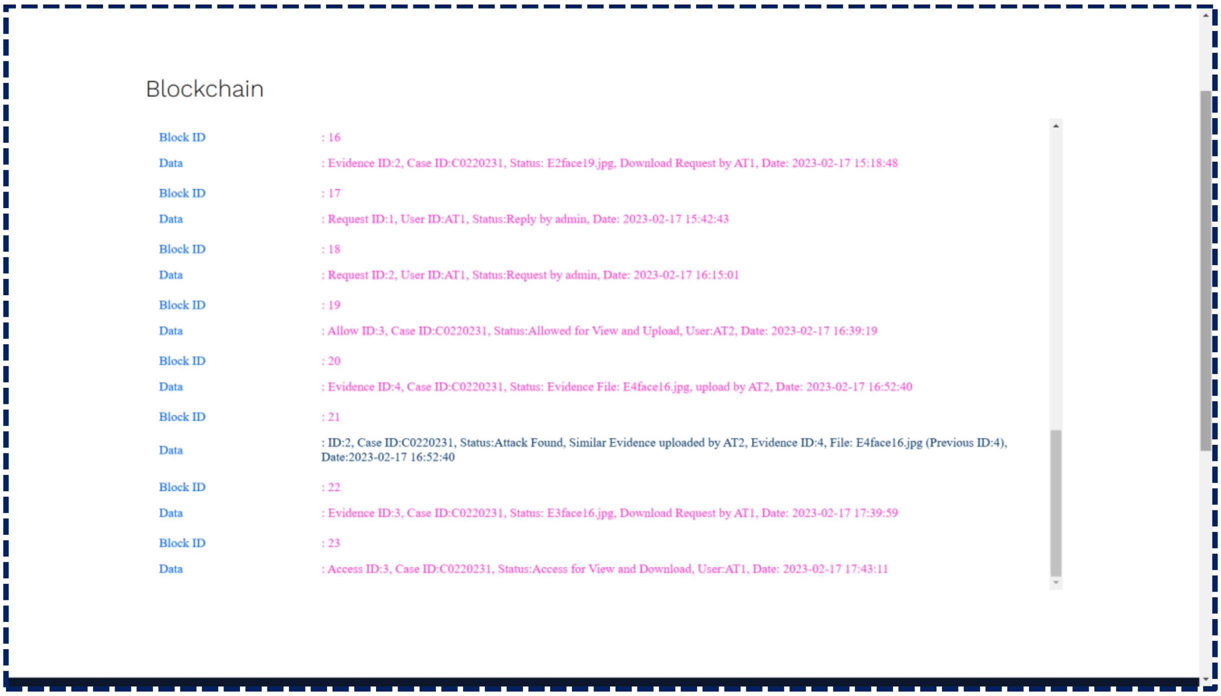
••••••••



Blockchain

Block Information

Block ID	: 1
Data	: ID:1, User ID:AT1, Status:Authorized User Created, Aadhar:432156784321, Date: 2023-02-15 21:00:40
Block ID	: 2
Data	: ID:2, User ID:AT2, Status:Authorized User Created, Aadhar:256344848454, Date: 2023-02-15 21:02:59
Block ID	: 3
Data	: ID:1, Case ID:C0220231, Status:Case Registered, Complainant Name: Prakash, Date: 2023-02-15 21:07:54
Block ID	: 4
Data	: Evidence ID:1, Case ID:C0220231, Status: Evidence File: E1coc2.jpg, Date: 2023-02-15 21:27:49
Block ID	: 5
Data	: Allow ID:1, Case ID:C0220231, Status:Allowed for , Date: 2023-02-16 21:19:19
Block ID	: 6



Blockchain

Block ID	: 16
Data	: Evidence ID:2, Case ID:C0220231, Status: E2face19.jpg, Download Request by AT1, Date: 2023-02-17 15:18:48
Block ID	: 17
Data	: Request ID:1, User ID:AT1, Status:Reply by admin, Date: 2023-02-17 15:42:43
Block ID	: 18
Data	: Request ID:2, User ID:AT1, Status:Request by admin, Date: 2023-02-17 16:15:01
Block ID	: 19
Data	: Allow ID:3, Case ID:C0220231, Status:Allowed for View and Upload, User:AT2, Date: 2023-02-17 16:39:19
Block ID	: 20
Data	: Evidence ID:4, Case ID:C0220231, Status: Evidence File: E4face16.jpg, upload by AT2, Date: 2023-02-17 16:52:40
Block ID	: 21
Data	: ID:2, Case ID:C0220231, Status:Attack Found, Similar Evidence uploaded by AT2, Evidence ID:4, File: E4face16.jpg (Previous ID:4), Date:2023-02-17 16:52:40
Block ID	: 22
Data	: Evidence ID:3, Case ID:C0220231, Status: E3face16.jpg, Download Request by AT1, Date: 2023-02-17 17:39:59
Block ID	: 23
Data	: Access ID:3, Case ID:C0220231, Status:Access for View and Download, User:AT1, Date: 2023-02-17 17:43:11

CHAPTER 8

REFERENCE

1. D. Li, W. Liu, L. Deng, and B. Qin, ``Design of multimedia blockchain privacy protection system based on distributed trusted communication,'' Trans. Emerg. Telecommun. Technol., vol. 32, no. 2, p. e3938, Feb. 2021.
2. M. Uddin, ``Blockchain medledger: Hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry,'' Int. J. Pharmaceutics, vol. 597, Mar. 2021, Art. no. 120235.
3. A. A. Khan, M. Uddin, A. A. Shaikh, A. A. Laghari, and A. E. Rajput, ``MF-ledger: Blockchain hyperledger sawtooth-enabled novel and secure multimedia chain of custody forensic investigation architecture,'' IEEE Access, vol. 9, pp. 103637-103650, 2021.
4. M. Li, C. Lal, M. Conti, and D. Hu, ``LEChain: A blockchain-based lawful evidence management scheme for digital forensics,'' Future Gener. Comput. Syst., vol. 115, pp. 406-420, Feb. 2021.
5. M. R. Kumar and N. Bhalaji, ``Blockchain based chameleon hashing technique for privacy preservation in E-governance system,''Wireless Pers. Commun., vol. 117, no. 2, pp. 1-20, 2020.
6. M. Lusetti, L. Salsi, and A. Dallatana, ``A blockchain based solution for the custody of digital files in forensic medicine,'' Forensic Sci. Int., Digit. Invest., vol. 35, Dec. 2020, Art. no. 301017.
7. J. Jeong, D. Kim, B. Lee, and Y. Son, ``Design and implementation of a digital evidence management model based on Hyperledger fabric,'' J. Inf. Process. Syst., vol. 16, no. 4, pp. 760-773, 2020.
8. L. Zarpala and F. Casino, ``A blockchain-based forensic model for financial crime investigation: The embezzlement scenario,'' 2020, arXiv:2008.07958. [Online]. Available: https://arxiv.org/abs/2008.07958.
9. H. R. Hasan, K. Salah, R. Jayaraman, M. Omar, I. Yaqoob, S. Pesic, T. Taylor, and D. Boscovic, ``A blockchain-based approach for the creation of digital twins,'' IEEE Access, vol. 8, pp. 34113-34126, 2020.
10. A. H. Lone and R. N. Mir, ``Forensic-chain: Blockchain based digitalnforensics chain of custody with PoC in hyperledger composer,'' Digit. Invest., vol. 28, pp. 44-55, Jan. 2019.
11. Z. Tian, M. Li, M. Qiu, Y. Sun, and S. Su, ``Block-DEF: A secure digital evidence framework using blockchain,'' Inf. Sci., vol. 491, pp. 151-165, Apr. 2019.

12. E. Yunianto, Y. Prayudi, and B. Sugiantoro, ``B-DEC: Digital evidence cabinet based on blockchain for evidence management," Int. J. Comput. Appl., vol. 181, no. 45, pp. 22-29, Mar. 2019.
13. M. Takemiya and B. Vanieiev, ``Sora identity: Secure, digital identity on the blockchain," in Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC), Jul. 2018, pp. 582-587.
14. K. Widatama, Y. Prayudi, and B. Sugiantoro, ``Application of RC4 cryptography method to support XML security on digital chain of custody data storage," Int. J. Cyber-Secur. Digit. Forensics, vol. 7, no. 3, pp. 230-237, 2018.
15. M. Shah, S. Saleem, and R. Zulqarnain, ``Protecting digital evidence integrity and preserving chain of custody," J. Digit. Forensics, Secur. Law, vol. 12, no. 2, pp. 121-130, Jun. 2017.

8.1. BOOK REFERENCES

1. Python and Flask:
    - "Flask Web Development" by Miguel Grinberg - This book focuses on web development using Flask, a popular Python web framework.
2. MySQL:
    - "Learning MySQL" by Bill Scott, Paul Dubois, Michael McLaughlin - A good resource to learn MySQL, which is a popular relational database management system.
3. Bootstrap:
    - "Bootstrap in Practice" by Alex Libby - This book covers the practical usage of Bootstrap for building responsive and visually appealing web pages.
4. WampServer:
    - As WampServer is a web development environment that includes Apache, MySQL, and PHP, resources on Apache, MySQL, and PHP would be relevant.
    - Online documentation and tutorials provided by the WampServer official website.
5. General Web Development (HTML, CSS, JavaScript):
    - "Head First HTML and CSS" by Elisabeth Robson, Eric Freeman - A beginner-friendly book for learning HTML and CSS.
    - "JavaScript and JQuery: The Missing Manual" by David Sawyer McFarland - A comprehensive guide to JavaScript and JQuery.

8.2. WEB REFERENCES

1. Python and Flask:
    - Flask Official Documentation: https://flask.palletsprojects.com/
    - "FlaskWeb " by Miguel Grinberg: https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world
2. MySQL:
    - MySQL Official Documentation: https://dev.mysql.com/doc/
    - W3Schools MySQL Tutorial: https://www.w3schools.com/sql/
3. Bootstrap:
    - Bootstrap Official Documentation: https://getbootstrap.com/docs/4.6/getting-started/introduction/
    - "BootstrapinPractice"byAlexLibby: https://www.manning.com/books/bootstrap-in-practice
4. WampServer:
    - WampServer Official Website: https://www.wampserver.com/
5. General Web Development (HTML, CSS, JavaScript):
    - MozillaDeveloperNetwork(MDN)Web Docs: https://developer.mozilla.org/en-US/docs/Web
    - "Head First HTML and CSS" by Elisabeth Robson, Eric Freeman: https://www.oreilly.com/library/view/head-first-html/9780596159917/
    - "JavaScript and JQuery: The Missing Manual" by David Sawyer McFarland: https://www.oreilly.com/library/view/javascript-jquery-the/9781491947074/