

CRIME ANALYSIS AND SECURITY ENHANCEMENT OF FORENSIC EVIDENCES USING BLOCKCHAIN

A Main Project Report

submitted by

ARCHANA A P (MES20MCA-2010)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications



Department of Computer Applications

MES College of Engineering
Kuttipuram, Malappuram - 679 582

July 2022

DECLARATION

I undersigned hereby declare that the project report **CRIME ANALYSIS AND SECURITY ENHANCEMENT OF FORENSIC EVIDENCES USING BLOCKCHAIN** , submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under supervision of Prof.Hyderali K, Associate Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: KUTTIPPURAM

ARCHANA A P (MES20MCA-2010)

Date: 08-07-2022

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM



CERTIFICATE

This is to certify that the report entitled **CRIME ANALYSIS AND SECURITY ENHANCEMENT OF FORENSIC EVIDENCES USING BLOCKCHAIN** is a bona fide record of the Main Project work carried out by **ARCHANA A P (MES20MCA-2010)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s)

External Supervisor(s)

Head Of The Department

Acknowledgements

At the very outset I would like to thank the almighty's mercy towards me over the years.I wish to express my sincere thanks to my project coordinator Ms.Priya J.D Assistant Professor, Dept. of Computer Applications who guided me for the successful completeness of this project. I also thank her for valuable suggestions, guidance, constant encouragement, boundless corporation, constructive comments and motivation extended to me for completion of this project work.I would express my sincere thanks to my guide Prof.Hyderali K ,Associate Professor for his immense guidance to complete the project successfully.I would like to express my sincere thanks to all the staff members of Master of Computer Applications department for their support and valuable suggestion for doing the project work.Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

ARCHANA A P (MES20MCA-2010)

Abstract

Crime Analysis in assisting police and helps in reduction and prevention of crimes and crime disorders using data mining tools. I collect the data from police department and try to get each and every detail, like the person's name, address , age, sex and photo details. Once I get the information and start to process the data. The use of information mining methods helps in resolving most complicated criminal cases. Then I calculate the crime rate based on the previous data using data mining techniques. I can identity the highest risk crime zones with the help of data mining techniques. The storage and processing on data with security is the need of each and every application field. Rapid increase in cyber crime, attackers behave maliciously to alter those data. But it is having great impact on forensic evidences which is required for provenance. To build the transparent system with immutability of forensic evidences, blockchain technology is more suitable. Blockchain technology provides the transfer of assets or evidence reports in transparent environment without central authority. The security enhancement of forensic evidences is achieved through implementation on Ethereum platform with high integrity, traceability and immutability.

Contents

Declaration	i
Certificate	ii
Acknowledgements	iii
Abstract	iv
Contents	v
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Background	1
1.1.1 Motivation	2
1.2 Objective	2
1.3 Report Organization	2
2 Literature Survey	3
3 Methodology	4
3.1 Introduction	4
3.2 Modules	5
3.3 Developing Environment	6
3.4 Work Flow	6
3.5 Crime pattern	7
3.6 BlockChain	7
3.6.1 Configuration of BockChain	8
3.7 Agile Methodology	8
3.7.1 User Story	9
3.7.2 Product Backlog	11
3.7.3 Project Plan	12
3.7.4 Sprint Plan	13

CONTENTS

vi

3.7.5 Sprint Actuals	14
4 Results and Discussions	18
4.1 Results	18
4.2 Discussions	27
4.3 Datasets	27
5 Conclusions	28
References	29
Appendix	30
Source Code	30

List of Figures

4.1	User Interface 1	18
4.2	User Interface 2	19
4.3	User Interface 3	19
4.4	User Interface 4	20
4.5	User Interface 5	20
4.6	User Interface 6	21
4.7	User Interface 7	21
4.8	User Interface 8	22
4.9	User Interface 9	22
4.10	User Interface 10	23
4.11	User Interface 11	23
4.12	User Interface 12	24
4.13	User Interface 13	24
4.14	User Interface 14	25
4.15	User Interface 15	25
4.16	User Interface 16	26
A.1	Dataflow Diagram	43
A.2	Dataflow Diagram	43
A.3	Dataflow Diagram	44
A.4	Dataflow Diagram	44
A.5	Dataflow Diagram	45

List of Tables

3.1	User Story 1	9
3.2	User Story 2	10
3.3	Product Backlog	11
3.4	Project Plan	12
3.5	Sprint Plan	13
3.6	Sprint 1 Actual	14
3.7	Sprint 2 Actual	15
3.8	Sprint 3 Actual	16
3.9	Sprint 4 Actual	17
A.1	Login	38
A.2	Signup	38
A.3	Crime type	39
A.4	Allocation	39
A.5	Dataset	39
A.6	Criminal list	40
A.7	Complaint	40
A.8	Police station	41
A.9	Report criminal	41
A.10	Evidence	42
A.11	Report	42
A.12	Request	42

Chapter 1

Introduction

1.1 Background

Crime prevention and detection become an important trend in crime and a very challenging to solve crimes. Several studies have discovered various techniques to solve the crimes that used to many applications. Such studies can help speed up the process of solving crime and help the computerized systems detect the criminals automatically. In addition, the rapidly advancing technologies can help address such issues. However, the crime patterns are always changing and growing. The crime data previously stored from various sources have a tendency to increase steadily. As a consequence, the management and analysis with huge data are very difficult and complex. To solve the problems previously mentioned, data mining techniques employ many learning algorithms to extract hidden knowledge from huge volume of data. Data mining is data analyzing techniques to find patterns and trends in crimes. It can help solve the crimes more speedily and also can help alert the criminal detection automatically.

The Blockchain technology is decentralized and distributed system in nature. Due to its distributed nature it has replaced traditional platforms. Blockchain makes use of a distributed ledger where information is arranged into blocks. Each block of the ledger contains huge number of signed transactions. The Blockchain stores and manages all kinds of transactions in a transparent way on a distributed ledger which replaces intermediary databases, employs a consensus mechanism; achieves decentralization, trust and data immutability which is useful in many use cases. Data immutability means it is not possible to manipulate or update any

transaction. Decentralization in Blockchain helps to replace intermediaries. Trust is assured distributed ledger and consensus mechanism which leads to security.

1.1.1 Motivation

Crime analysis with crime mapping helps in understanding the concepts and practices of crime analysis in assisting police and helps in the reduction and prevention of crimes and crime disorders. For public safety purposes, the crime mapping is an essential research area to concentrate on. We can identify the highest risk crime zones with the help of data mining techniques. And add the forensic department for investigation the uploaded evidences are stored in blockchain.

1.2 Objective

To identify the frequent crime pattern in a particular area for assisting police and helps in reduction and prevention of crimes by providing patrol in hotspot areas. To store details of crimes and criminal, that make efficient retrieval whenever needed. For more security use the Block Chain technology. In this block chain based secure system for forensic evidences are proposed. All the uploaded evidences and the uploaded reports are stored in block chain.

1.3 Report Organization

The project report is divided into four sections. Section 1, Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

As essential apparatus in crime analysis, crime mapping and Geographical Information Systems (GIS) are being progressively more accepted by police agencies. Development in technology and the accessibility of geographic data sources make it feasible for police departments to use GIS and crime mapping. GIS and crime mapping can be utilized as devices to discover reasons contributing to crime, and hence let law enforcement agencies proactively take action against the crime problems before they become challenging. The purpose of this study is to conduct a literature review of Geographical Information System and Crime Mapping in Crime Analysis and to propose policy recommendations regarding to implementation of crime mapping and GIS. To achieve this purpose, first a historical evaluation of GIS and crime mapping will be rendered and then the importance of place will be explained in terms of assessing crime problems accurately. Blockchain is one of the emerging technologies. It solves problem of mutability by storing the data in blocks and making chain by keeping previous block hash value in next block. It also solves the problem of centralized authority by establishing peer to peer network. The first application of blockchain was bitcoin by Satoshi Nakamoto, which introduced the concepts of block mining and consensus algorithm. The literature shows blockchain based solution will certainly enhance security and trust in forensic evidences.

Chapter 3

Methodology

3.1 Introduction

Crime Analysis in assisting police and helps in reduction and prevention of crimes and crime disorders using data mining tools. We collect the data from police department and try to get each and every detail, like the person's name, address , age, sex and photo details. Once we get the information, we start to process the data. The use of information mining methods helps in resolving most complicated criminal cases. We calculate the crime rate based on the previous data using data mining techniques. We can identify the highest risk crime zones with the help of data mining techniques.

Crime prevention and detection become an important trend in crime and a very challenging to solve crimes. Several studies have discovered various techniques to solve the crimes that used to many applications. Such studies can help speed up the process of solving crime and help the computerized systems detect the criminals automatically. In addition, the rapidly advancing technologies can help address such issues. However, the crime patterns are always changing and growing. The crime data previously stored from various sources have a tendency to increase steadily. As a consequence, the management and analysis with huge data are very difficult and complex. To solve the problems previously mentioned, data mining techniques employ many learning algorithms to extract hidden knowledge from huge volume of data. Data mining is data analyzing techniques to find patterns and trends in crimes. It can help solve the crimes more speedily and also can help alert the criminal detection automatically.

The Blockchain technology is decentralized and distributed system in nature. Due to its distributed nature it has replaced traditional platforms. Blockchain makes use of a distributed ledger where information is arranged into blocks. Each block of the ledger contains huge number of signed transactions. Blockchain technology provides the transfer of assets or evidence reports in transparent environment without central authority. The security enhancement of forensic evidences is achieved through implementation on Ethereum platform with high integrity, traceability and immutability. For more security use the Block Chain technology. In this block chain based secure system for forensic evidences are proposed. All the uploaded evidences and the uploaded reports are stored in block chain.

This paper gives the brief reviews of researches on various implementation of data mining and the guidelines to solve the crimes by using data mining techniques. It also discusses research gaps and challenges in the area of crime data mining. . And add the forensic department for investigation the uploaded evidences are stored in blockchain.

3.2 Modules

The project is divided into 3 functional modules. They are,

1. Admin

- Login
- Add and manage police and forensic
- View criminal list
- View complaints
- Allocate complaint to police

2. User

- Register
- Login
- View crime types
- View crimes
- View criminals
- Make a complaint
- Report a criminal found

- Upload evidence
- 3. Police
 - Login
 - Manage crime types
 - Add and manage criminal list
 - View crime pattern
 - View allocated case
 - Update status
 - Upload evidence
 - View reported criminals
- 4. Forensic
 - Login
 - View request and update status
 - Report upload
 - View history

3.3 Developing Environment

- OPERATING SYSTEM: WINDOWS 10
 - FRONT END: HTML, CSS, JAVASCRIPT
 - BACK END: Mysql
 - IDE USED: Jetbrains Pycharm, Android studio
 - TECHNOLOGY USED: PYTHON JAVA
 - FRAME WORK USED: Flask

3.4 Work Flow

Apriori is an algorithm for frequent itemset mining and association rule learning over relational databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as those item sets appear sufficiently often in the database. Apriori algorithm is used to help prune the candidates explored during frequent

item-set generation to reduce the processing time. Apriori algorithm needs to scan all item-sets. So, it uses a long period of time as well. proposed the improved Apriori algorithm by using the compressed database algorithm for association rule mining to reduce the amount of time needed to read data from the database. Improved Apriori algorithm to find the effective association rule and to reduce the amount of processing time.

3.5 Crime pattern

The issues of crime pattern are concerning with finding and predicting the hidden crime. Nowadays, the crime rate is increasing continuously and the crime patterns are always changing. As a consequence, the behaviours in crime are difficult to be explained and predicted. The research interests on crime prevention and detection are concerning with finding and conducting the crime model to detect crimes. The challenge is modeling the crime attack behaviours that support crime detection although the crime patterns are changing. The predictive and statistic methods may be useful to find and conduct the crime model. The crime model should be able to predict and detect the criminal behaviours

3.6 BlockChain

All the evidence information's are stored in block chain. Block chain is a record-keeping technology designed to make it impossible to hack the system or forget the data stored on it, thereby making it secure and immutable. It is a type of distributed ledger technology (DLT), a digital system for recording transactions and related data in multiple places at the same time. Once the information stored in block chain it is not possible to manipulate the stored information . It consists of an expanding list of transactions or records stored in the blocks and uses peer to peer networks. The blocks in the block chain are connected as a chain with the use of hashing algorithms. Blocks are stored in a decentralized network where all the blocks are present in multiple nodes. As data is decentralized the chances of data tampering and data loss is less which makes block chain more secure and transparent. Each block of the block chain consists of the previous block's hash value, nonce, a timestamp, the records of the block and the hash of the current block. The main advantages of using block chain are decentralization,

security , transparency, and immutability.

3.6.1 Configuration of BockChain

Truffle:

Truffle is the most popular development tooling for Ethereum programmers. Easily deploy smart contracts and communicate with their underlying state without heavy client side programming. An especially useful library for the testing and iteration of Ethereum smart contracts. It is used to create configuration files and compile block chain. First install node to create files for block chain automatically. Through this create contract that contain sol files. Sol files contain the information that we want to pass into the block chain. This concept is called smart contracting.

Ganache:

Ganache is a high-end development tool used to run your own local block chain for both Ethereum and Corda App development. It act as a server to see the info that pass to the block chain.

3.7 Agile Methodology

This project was developed using Agile Development Model. The entire project was divided into four sprints. In the first sprint, the characters for the password was developed. The designing of front-end and development of back-end was done in the second , third and fourth sprint respectively.

3.7.1 User Story

User StoryID	As a <type of user>	I want to	So that I can
1	Police	Login	Login successful with correct username and password
2	Police	Manage crime types	Add and manage crime types
3	Police	Manage crimes	Add and manage crimes
4	Police	Manage criminals	Add and manage criminal information
5	Police	View complaints	View complaints from user
6	Police	View allocated case	View allocated case from admin
7	User	Registration	User registration by personal information
8	User	Login	Login successful with correct username and password
9	User	View crime types	View registered crime types
10	User	View crimes and criminals	View registered crimes

Table 3.1: User Story 1

User StoryID	As a <type of user>	I want to	So that I can
11	User	Upload evidence	Send evidences
12	User	Make a complaint	Send a new complaint complaint
13	Admin	Login	Login successful with correct username and password
14	Admin	Add police and forensic	Add and manage police and forensic
15	Admin	View criminal list	View criminal list
16	Admin	View complaint	View complaint from user
17	Admin	Allocate complaint	Allocate complaint to police
18	Forensic	Login	Login successful with correct username and password
19	Forensic	View request and update status	View request from police and update status
20	Forensic	Upload report	Send report
21	Forensic	View history	View history

Table 3.2: User Story 2

3.7.2 Product Backlog

User Story ID	Priority <High/ Medium/ Low>	Size (Hours)	Sprint <#>	Status <Planned/ In progress/ Completed	Release Date	Release Goal
1	Medium	8	1	Completed	01/05/2022	Forensic department form designing
2	High	10		Completed	12/05/2022	Code for forensic
3	Medium	6	2	Completed	25/05/2022	Form designing for police and admin
4	High	5		Completed	29/05/2022	Add and manage police and admin
5	High	3	3	Completed	02/06/2022	Block chain management , create block chain, truffle management
6	High	2		Completed	05/06/2022	Contract creation blockchain implementation
7	High	11	4	Completed	20/06/2022	Add and management blocks to block chain
8	Medium	8		Completed	25/06/2022	Block chain implements to forensic and police

Table 3.3: Product Backlog

3.7.3 Project Plan

User StoryID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	20/04/2022	01/05/2022	10	Completed
2		04/05/2022	12/05/2022	8	Completed
3	Sprint 2	15/05/2022	25/05/2022	6	Completed
4		26/05/2022	29/05/2022	5	Completed
5	Sprint 3	30/05/2022	02/06/2022	3	Completed
6		03/06/2022	05/06/2022	2	Completed
7	Sprint 4	06/06/2022	20/06/2022	9	Completed
8		24/06/2022	25/06/2022	9	Completed

Table 3.4: Project Plan

3.7.4 Sprint Plan

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	0	1	1	0	1	0	0	0	0	0
Form Design	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Coding	25/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	1	0
UserStory #4,#5																
Add and manage operations of police and forensic	29/05/2022	5	1	1	1	0	0	1	0	0	0	1	0	0	0	0
UserStory #6,#7																
Blockchain management	02/06/2022	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Blockchain creation and implementation	05/06/2022	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0
UserStory #8,#9,#10																
Add and manage blocks to blockchain	20/06/2022	11	3	0	0	2	1	0	3	2	0	0	0	0	0	0
UserStory #11,#12,#13, #14																
Blockchain implements to police and forensic	25/06/2022	8	1	1	1	0	0	2	1	2	0	0	0	0	0	0
Testing & Output generation	4/07/2022	3	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Total		54	10	5	5	5	3	4	6	5	2	2	2	2	2	1

Table 3.5: Sprint Plan

3.7.5 Sprint Actuals

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	0	1	1	0	1	0	0	0	0	0
Form desihn	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Coding	25/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	1	0
UserStory #4,#5																
Add and manage operations of police and forensic																
UserStory #6,#7																
Blockchain management																
Blockchain creation and implementation																
UserStory #8,#9,#10																
Add and manage blocks to blockchain																
UserStory #11,#12,#13, #14																
Blockchain implements to police and forensic																
Testing & Output generation																
Total		22	3	2	1	3	2	1	2	2	1	2	1	1	1	0

Table 3.6: Sprint 1 Actual

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	0	1	1	0	1	0	0	0	0	0
Form desihn	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Coding	25/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	1	0
UserStory #4,#5																
Add and manage operations of police and forensic	29/05/2022	5	1	1	1	0	0	1	0	0	0	1	0	0	0	0
UserStory #6,#7																
Blockchain management																
Blockchain creation and implementation																
UserStory #8,#9,#10																
Add and manage blocks to blockchain																
UserStory #11,#12,#13, #14																
Blockchain implements to police and forensic																
Testing & Output generation																
Total		27	4	3	2	3	2	2	2	1	2	2	2	1	1	0

Table 3.7: Sprint 2 Actual

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	0	1	1	0	1	0	0	0	0	0
Form desihn	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Coding	25/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	1	0
UserStory #4,#5																
Add and manage operations of police and forensic	29/05/2022	5	1	1	1	0	0	1	0	0	0	1	0	0	0	0
UserStory #6,#7																
Blockchain management	02/06/2022	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Blockchain creation and implementation	05/06/2022	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0
UserStory #8,#9,#10																
Add and manage blocks to blockchain																
UserStory #11,#12,#13, #14																
Blockchain implements to police and forensic																
Testing & Output generation																
Total		32	6	4	4	3	2	2	2	1	2	2	2	1	1	0

Table 3.8: Sprint 3 Actual

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	0	1	1	0	1	0	0	0	0	0
Form Design	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Coding	25/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	0	0
UserStory #4,#5																
Add and manage operations of police and forensic	29/05/2022	5	1	1	1	0	0	1	0	0	0	1	0	0	0	0
UserStory #6,#7																
Blockchain management	02/06/2022	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Blockchain creation and implementation	05/06/2022	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0
UserStory #8,#9,#10																
Add and manage blocks to blockchain	20/06/2022	11	3	0	0	2	1	0	3	2	0	0	0	0	0	0
UserStory #11,#12,#13, #14																
Blockchain implements to police and forensic	25/06/2022	8	1	1	1	0	0	2	1	2	0	0	0	0	0	0
Testing & Output generation	4/07/2022	3	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Total		54	10	5	5	5	3	4	6	5	2	2	2	2	2	1

Table 3.9: Sprint 4 Actual

Chapter 4

Results and Discussions

4.1 Results

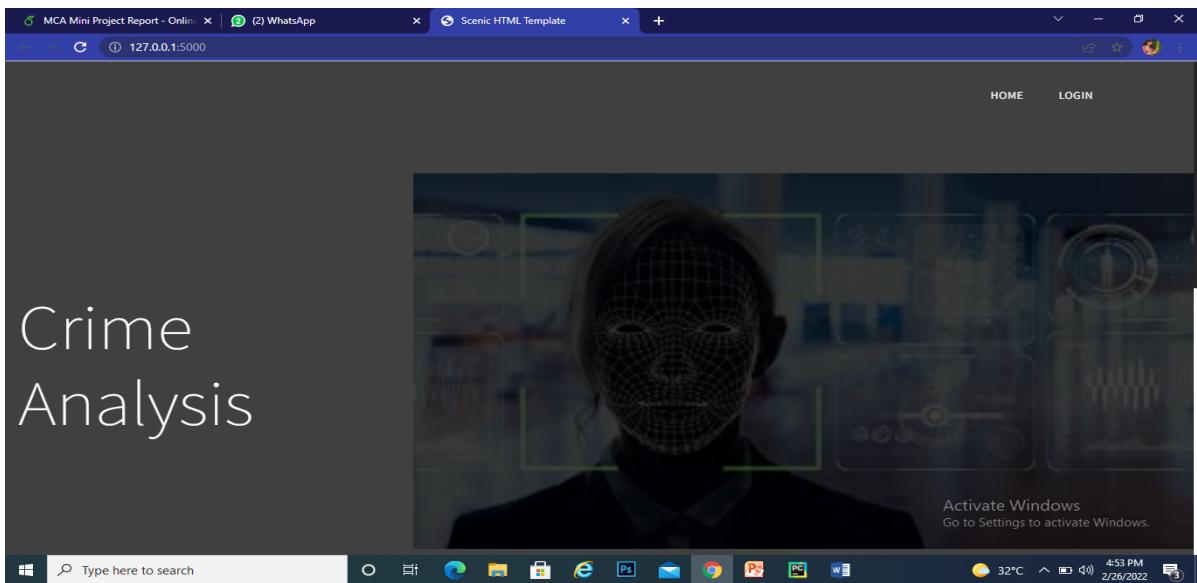


Figure 4.1: User Interface 1

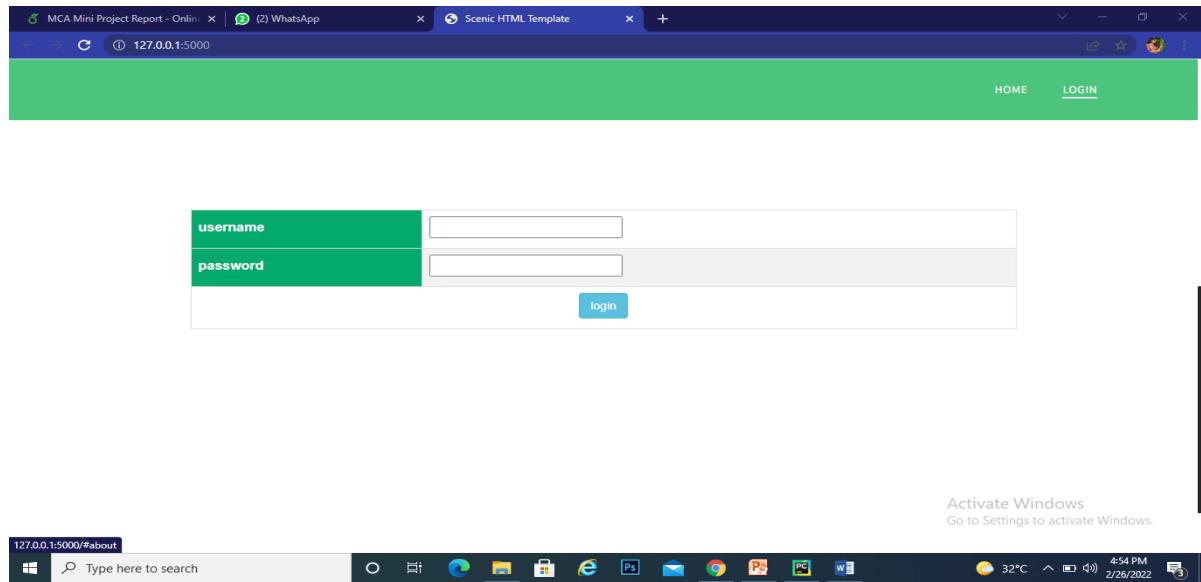


Figure 4.2: User Interface 2

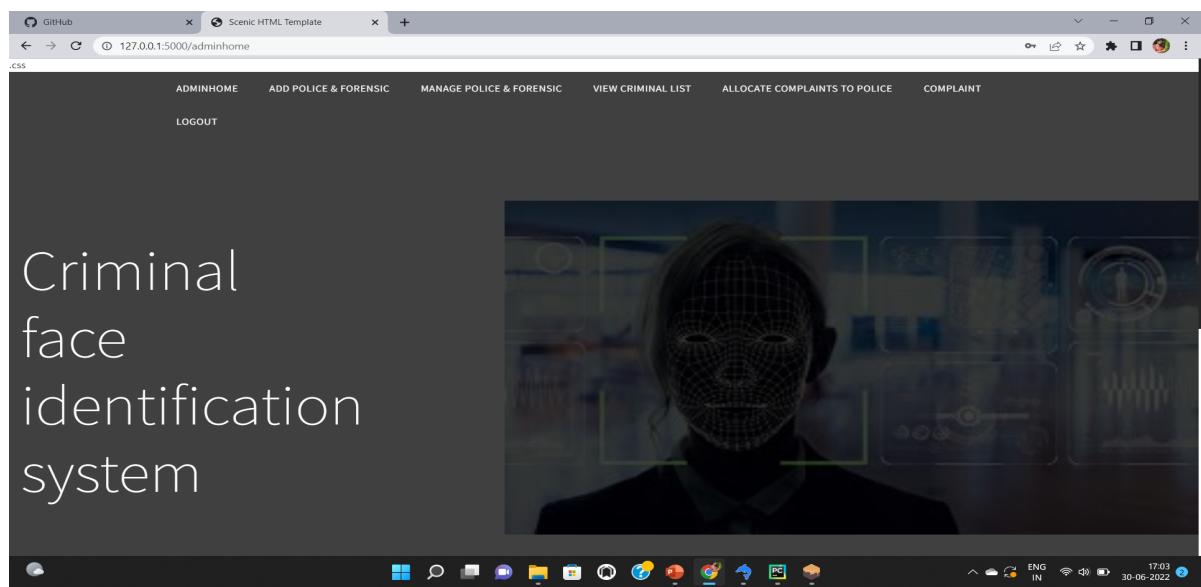


Figure 4.3: User Interface 3

The screenshot shows a web application interface titled "Manage Police & Forensic". The top navigation bar includes links for ADMINHOME, ADD POLICE & FORENSIC, MANAGE POLICE & FORENSIC, VIEW CRIMINAL LIST, ALLOCATE COMPLAINTS TO POLICE, and COMPLAINT. A "LOGOUT" link is also present. The main content area displays a table with the following data:

Type	police station				
<input type="button" value="Search"/>					
name	email	phone	place	pin code	
police	police@gmail.com	8786765764	vadakara	678451	edit delete
Biju	biju@gmail.com	8907654321	ponnani	676102	edit delete
joy	joy@gmail.com	7865438900	calicut	673456	edit delete
Kumar	kumar@gmail.com	8906543678	Tirur	673602	edit delete

The status bar at the bottom shows "127.0.0.1:5000/managepolicestation#about", the date "06-07-2022", and system icons.

Figure 4.4: User Interface 4

The screenshot shows a web application interface titled "COMPLAINT". The top navigation bar includes links for ADMINHOME, ADD POLICE & FORENSIC, MANAGE POLICE & FORENSIC, VIEW CRIMINAL LIST, ALLOCATE COMPLAINTS TO POLICE, and COMPLAINT. A "LOGOUT" link is also present. The main content area displays a table with the following data:

DATE	USER NAME	PLACE	COMPLAINT	
2022-06-21	Anu srwe	vadakara	plz help	Allocate complaint
2022-06-21	Anu srwe	vadakara	helpme	Allocate complaint
2022-06-21	Anu srwe	vadakara	please help	Allocate complaint
2022-06-21	Anu srwe	vadakara	gfghhhddd	Allocate complaint
2022-06-21	Anu srwe	vadakara	gfghhhddd	Allocate complaint

The status bar at the bottom shows "127.0.0.1:5000/COMPLAINT#about", the date "06-07-2022", and system icons.

Figure 4.5: User Interface 5

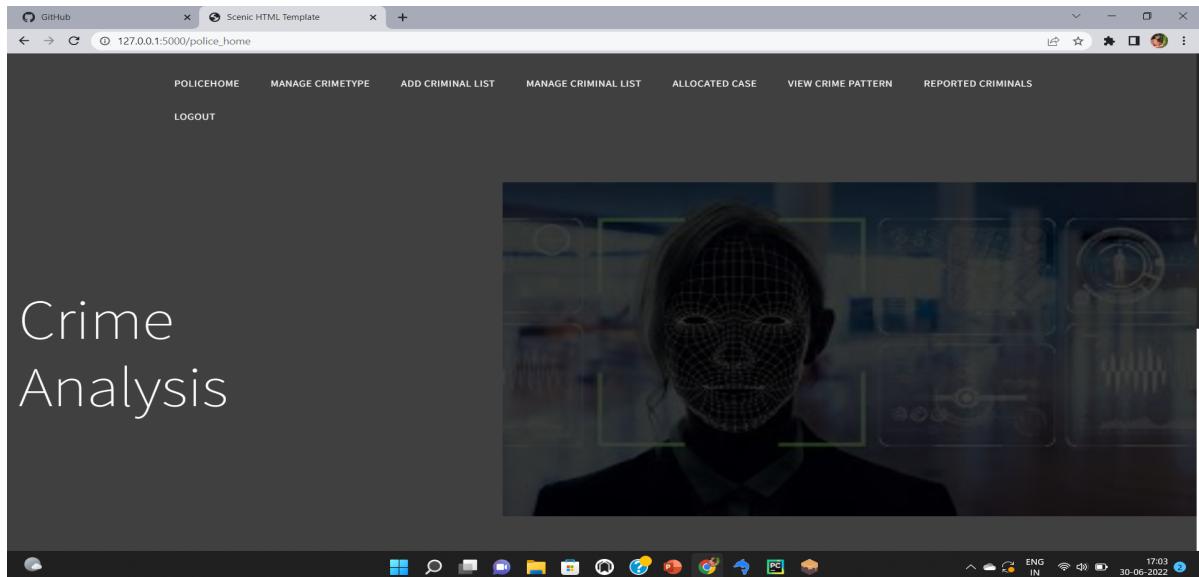


Figure 4.6: User Interface 6

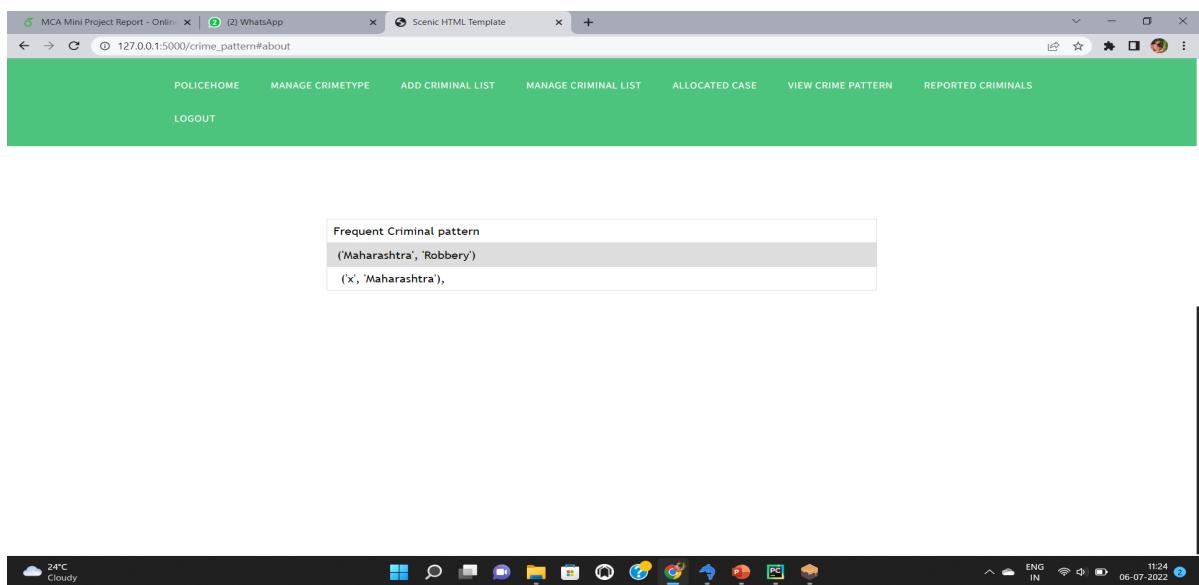


Figure 4.7: User Interface 7

The screenshot shows a web application interface titled "Scenic HTML Template". The top navigation bar includes links for POLICEHOME, MANAGE CRIMETYPE (which is currently selected), ADD CRIMINAL LIST, MANAGE CRIMINAL LIST, ALLOCATED CASE, VIEW CRIME PATTERN, and REPORTED CRIMINALS. There is also a LOGOUT link. Below the navigation bar is a table listing various crime types:

type	description	
Murder	more done	delete
Robbery	most areas	delete
Theft of vehicle	many times	delete
kidnapping	complicated	delete
Rape	help	delete
Rioting	complicated	delete
add		

Figure 4.8: User Interface 8

The screenshot shows a web application interface titled "Scenic HTML Template". The top navigation bar includes links for FORENSIC HOME (which is currently selected), VIEW REQUEST, REPORT UPLOAD, VIEW HISTORY, and LOGOUT. The main content area features a large image of a person's face with a wireframe overlay, suggesting a facial recognition or analysis process. To the left of the image, the text "Forensic Department" is displayed. The bottom of the screen shows a standard Windows taskbar with various icons.

Figure 4.9: User Interface 9

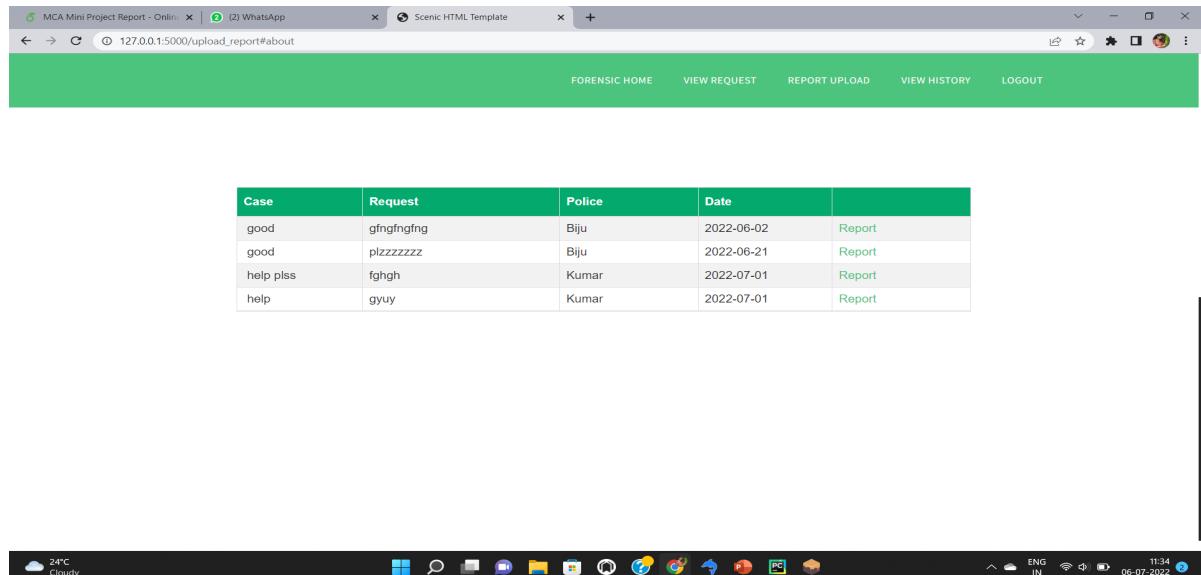


Figure 4.10: User Interface 10

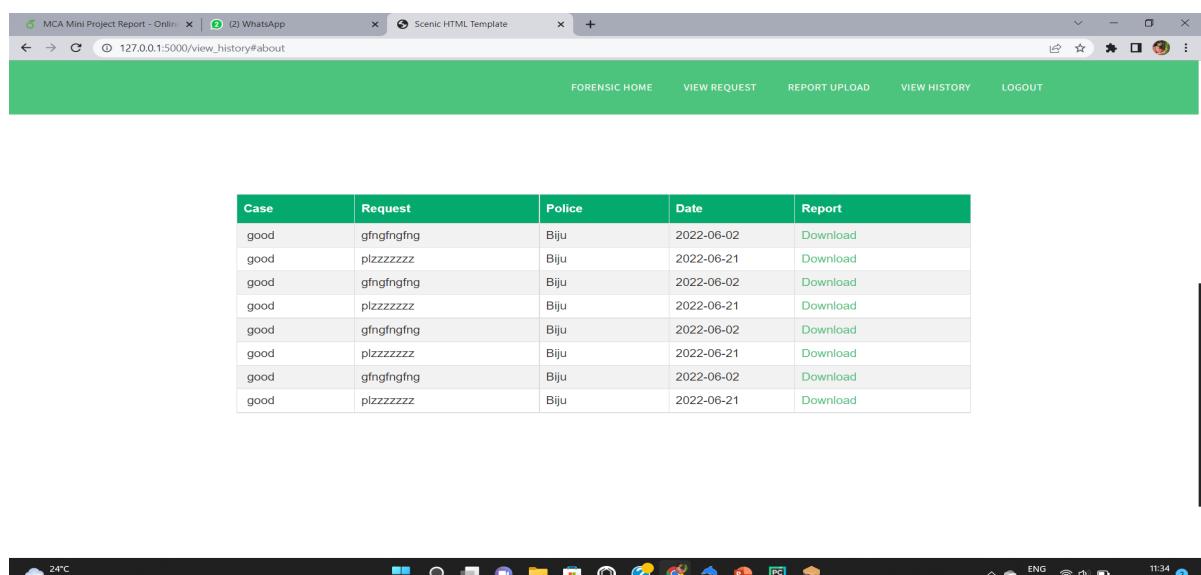


Figure 4.11: User Interface 11

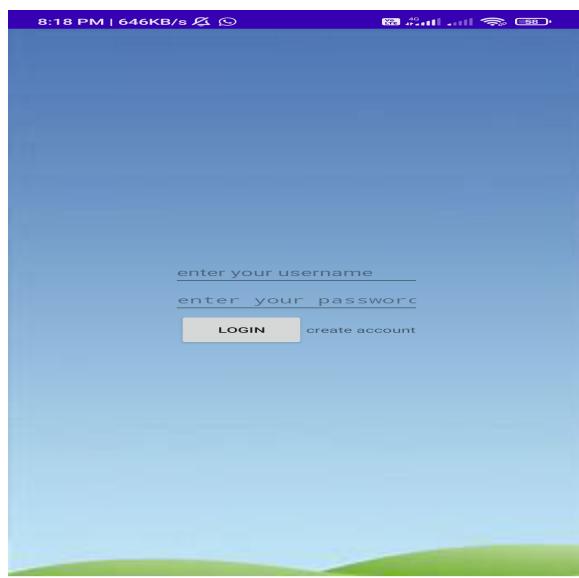


Figure 4.12: User Interface 12

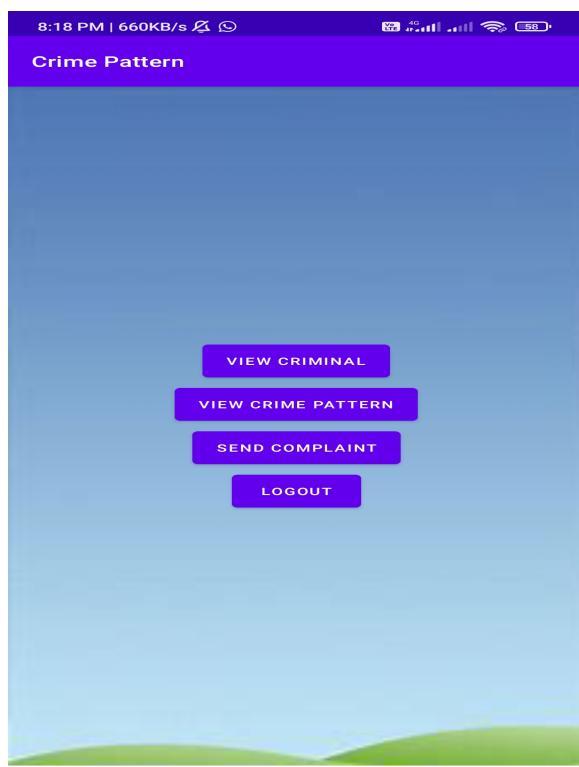


Figure 4.13: User Interface 13

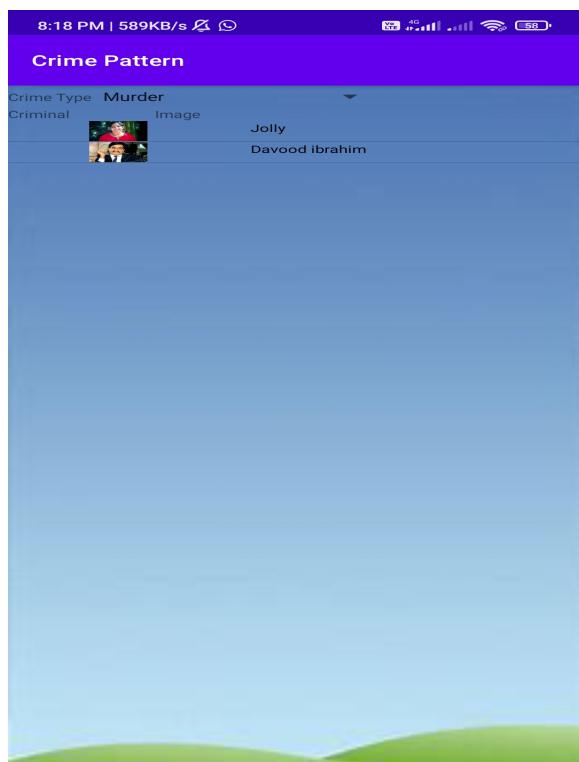


Figure 4.14: User Interface 14

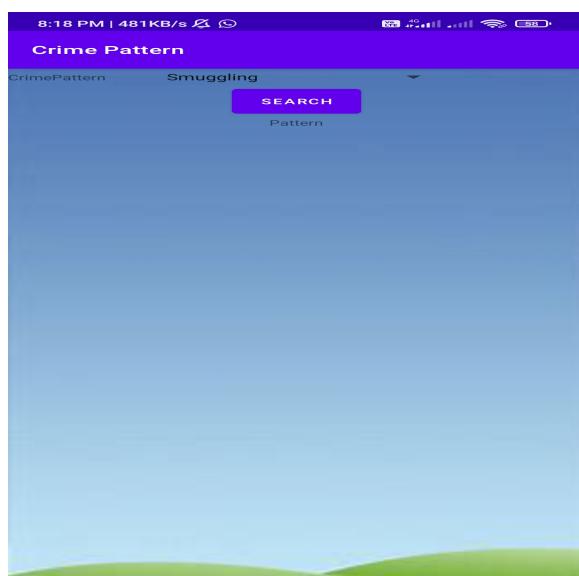


Figure 4.15: User Interface 15

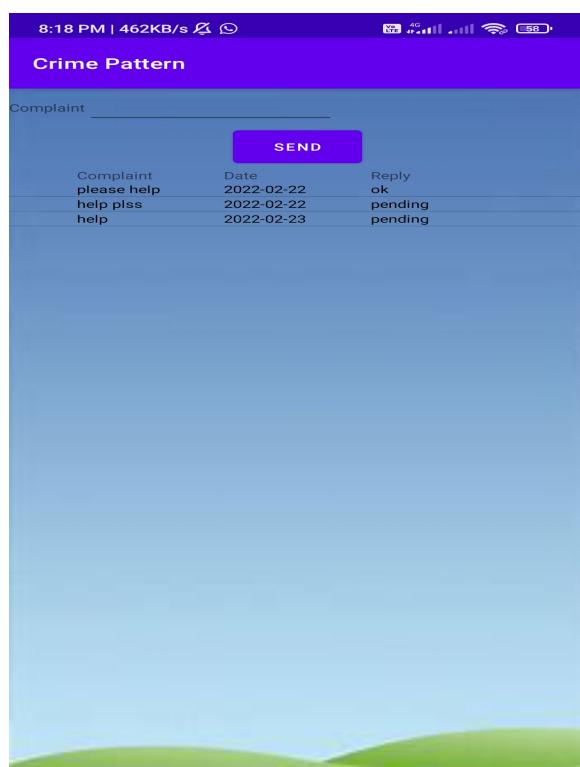


Figure 4.16: User Interface 16

4.2 Discussions

It gives the brief reviews of researches on various implementation of data mining and the guidelines to solve the crimes by using data mining techniques. It also discusses research gaps and challenges in the area of crime data mining. The crime data previously stored from various sources have a tendency to increase steadily. As a consequence, the management and analysis with huge data are very difficult and complex. To solve the problems previously mentioned, data mining techniques employ many learning algorithms to extract hidden knowledge from huge volume of data. Data mining is data analyzing techniques to find patterns and trends in crimes. It can help solve the crimes more speedily and also can help alert the criminal detection automatically. The blockchain based system is implemented for securing forensic reports. As soon as report is added into blocks, if any other node tries to change the report then it will get reflected in complete chain. This provides the immutability.

4.3 Datasets

Case History dataset from Kaggle website is used.

ideal for data items that are used frequently and in an unpredictable order. The term data set refers to a file that contains one or more records. The record is the basic unit of information used by a program running on OS. Any named group of records is called a data set. Data sets can hold information such as medical records or insurance records, to be used by a program running on the system. Data sets are also used to store information needed by applications or the operating system itself, such as source programs, macro libraries, or system variables or parameters. For data sets that contain readable text, you can print them or display them on a console (many data sets contain load modules or other binary data that is not really printable). Data sets can be cataloged, which permits the data set to be referred to by name without specifying where it is stored. The records in a data set can be organized in various ways, depending on how we plan to access the information. If you write an application program that processes things like personnel data, for example, your program can define a record format for each person's data.

Chapter 5

Conclusions

Crime are characterized which change over time and increase continuously. The changing and increasing of crime lead to the issues of understanding the crime behaviour, crime predicting, precise detection, and managing large volumes of data obtained from various sources. Research interests have tried to solve these issues. However, these researches are still gaps in the crime detection accuracy. This leads to the challenges in the field of crime detection. The challenges include modeling of crimes for finding suitable algorithms to detect the crime, precise detection, data preparation and transformation, and processing time.

The secure forensic evidence system has been proposed to achieve optimization by creating chain of limited users responsible in forensic investigation. The private Ethereum platform is used to build blockchain technology. The proposed system is used to fulfill the accurate tracking of police complaint with increased level of belief and decreased level of conflict. Whenever the new complaint is raised, a new block will get added to the blockchain. After creation of block if any change is there the tracing of that particular block is possible. Such blocks will be categorized into invalid blocks. So very rare chance of immutability. Overall, the security enhancement of forensic evidences is achieved through implementation on Ethereum platform with high integrity, traceability and immutability.

References

- [1] **Y. Zhao and G. Karypis** “Evaluation of hierarchical clustering algorithms for document datasets,” [Online]. Available: <http://doi.acm.org/10.1145/584792.584877>.
- [2] **T. Pang-Ning, S. Michael, and K. Vipin** Introduction to Data Mining, 1st ed. Pearson, 5 2005
- [3] **N. Sasaki, R. Nishimura, and Y. Suzuki**, (2006) “Audiotwatermarking based on association analysis,” ” *in Signal Processing*.
- [4] **Zarpala, L., Casino, F.** (2020),”A blockchain-based Forensic Model for Financial Crime Investigation: The Embezzlement Scenario,” ArXiv, abs/2008.07958.

Appendix

Source Code

```

import os
from flask import*
import pymysql
from werkzeug.utils import secure_filename
from src.new import imgprocessing
from src.apri import aprioris

app=Flask(__name__)
from datetime import *
con=pymysql.connect(host="localhost",user="root",password="",port=3306,db="criminal face identification")
cmd=con.cursor()
app.secret_key="qwer"
path=r"./static/criminalpic"
from web3 import Web3, HTTPProvider

blockchain_address = 'HTTP://127.0.0.1:7545'
# Client instance to interact with the blockchain
web3 = Web3(HTTPProvider(blockchain_address))
# Set the default account (so we don't need to set the "from" for every transaction call)
web3.eth.defaultAccount = web3.eth.accounts[0]

compiled_contract_path = r'D:\criminal face identification\criminal face
identification\src\node_modules\.bin\build\contracts\MigrationsCase.json'
# Deployed contract address (see 'migrate' command output: 'contract address')
deployed_contract_address = '0xb737139F6EBacC43C55d3ff436708Bc37eFA7aa8'

import functools
def login_required(func):
    @functools.wraps(func)
    def secure_function():
        if "lid" not in session:
            return redirect ("/")
        return func()
    return secure_function

@app.route('/')
def main():
    return render_template("login.html")

```

```

@app.route('/logout')
def logout():
    session.clear()
    return render_template("login.html")

@app.route('/login',methods=['get','post'])
def login():
    username=request.form['textfield']
    password=request.form['textfield2']
    cmd.execute("select * from login where username='"+username+"' and password='"+password+"'")
    s=cmd.fetchone()
    if s is None:
        return """<script>alert("invalid");window.location="/</script>"""
    elif s[3]=="admin":
        session['lid']= s[0]
        return """<script>alert("success");window.location="/adminhome"</script>"""
    elif s[3] == "forensic":
        session['lid']=s[0]
        return """<script>alert("success");window.location="/forensic_home"</script>"""

    elif s[3] == "police":
        session['lid']= s[0]
        return """<script>alert("success");window.location="/police_home"</script>"""
    else:
        return """<script>alert("login success");window.location="/</script>"""

@app.route('/police_station_home',methods=['get','post'])
def police_station_home():
    return render_template("home.html")

@app.route('/forensic_home',methods=['get','post'])
def forensic_home():
    return render_template("home.html")

@app.route('/police_home',methods=['get','post'])
def police_home():
    return render_template("police home.html")

@app.route('/editpolice_statiion',methods=['get','post'])
def editpolice_statiion():
    id=request.args.get('id')
    session['sid']=id
    cmd.execute("select * from police_statiion where p_id='"+str(id)+"'")
    s=cmd.fetchone()
    return render_template("editpolice_station.html",val=s)

@app.route('/updatestation',methods=['get','post'])
def updatestation():
    name=request.form['textfield']
    email=request.form['textfield2']
    phone=request.form['textfield3']
    place=request.form['textfield4']
    pincode=request.form['textfield5']
    cmd.execute("update police_statiion set
                name='"+name+"',phone='"+phone+"',email='"+email+"',place='"+place+"',pincode='"+pincode+"'
                where
                p_id='"+str(session['sid'])+"'")
    con.commit()
    return """<script>alert("updated");window.location="/managepolicestation"</script>"""

@app.route('/deletestation',methods=['get','post'])
def deletestation():
    sid=request.args.get('id')
    cmd.execute("delete from police_statiion where p_id='"+str(sid)+"'")
```

```

con.commit()
return '''<script>alert("delete");window.location="/managepolicestation"</script>''

@app.route('/police_station_reg',methods=['get','post'])
def police_station_reg():
    try:
        name=request.form['textfield']
        email=request.form['textfield2']
        phone=request.form['textfield3']
        place=request.form['textfield4']
        pincode=request.form['textfield5']
        username=request.form['textfield6']
        password=request.form['textfield7']
        type=request.form['select']
        cmd.execute("insert into login values(null,'" +username+"','"+password+"','"+type+"')")
        id=con.insert_id()
        cmd.execute("insert into police_statiion
                    values(null,'" +name+"','"+phone+"','"+email+"','"+place+"','"+pincode+"','"+str(id)+"')")
        con.commit()
        return '''<script>alert("registration success");window.location="/adminhome"</script>''
    except Exception as e:
        return '''<script>alert("duplicate entry");window.location="/adminhome"</script>''

@app.route('/policestationreg')
@login_required

def policestationreg():
    return render_template("police station reg.html")
@app.route('/regpolice')
def regpolice():
    return render_template("reg police.html")

@app.route('/addpolice',methods=['post'])
def addpolice():
    fname=request.form['textfield2']
    lastnamr = request.form['textfield3']
    gender = request.form['radiobutton']
    email = request.form['textfield4']
    phone = request.form['textfield5']
    DOB = request.form['textfield6']
    rank = request.form['textfield7']
    username = request.form['textfield8']
    password = request.form['textarea']
    cmd.execute("insert into login values(null,'" +username+"','"+password+"','police')")
    id=con.insert_id()
    cmd.execute("insert into police_statiion
                values(null,'" +fname+"','"+lastnamr+"','"+gender+"','"+email+"','"+phone+"','"+DOB+"','"+rank+"','"+str(session['lid'])+"','"+str(id)+"')")
    con.commit()
    return '''<script>alert("registration success");window.location="/police_station_home"</script>''

@app.route('/registration')
def registration():
    return render_template("registration.html")
@app.route('/sendcomplaint')
def sendcomplaint():
    return render_template("send_complaint.html")

@app.route('/sendrportpolice')

```



```

def sendrportpolice():
    return render_template("send rport police.html")
@app.route('/viewandmanagecriminallist')
@login_required
def viewandmanagecriminallist():
    cmd.execute("SELECT `criminal_list`.* FROM `criminal_list` ")
    s=cmd.fetchall()
    print(s)
    return render_template("view and manage criminal list.html",val=s)

@app.route('/deletecriminalist',methods=['get','post'])
def deletecriminalist():
    sid=request.args.get('id')
    cmd.execute("delete from criminal_list where criminal_id='"+str(sid)+"' ")
    con.commit()
    return '''<script>alert("delete");window.location="/viewandmanagecriminalist"</script>'''

@app.route('/viewcriminalist',methods=['get','post'])
@login_required
def viewcriminalist():
    cmd.execute("select * from criminal_list")
    s = cmd.fetchall()
    print(s)
    return render_template("view criminal list.html", val=s)

@app.route('/viewrecord')
def viewrecord():
    cmd.execute("SELECT `criminal_record`.*,`criminal_list`.* FROM `criminal_list` JOIN `criminal_record` ON
               `criminal_list`.`criminal_id`='criminal_record`.`cid`")
    s = cmd.fetchall()
    return render_template("view record.html",val=s)
@app.route('/viewtaskstatus')
def viewtaskstatus():
    cmd.execute("SELECT `register_police`.*,`view_task_status`.* FROM `view_task_status` JOIN `register_police` ON
               `view_task_status`.p_id='register_police'.p_id")
    s=cmd.fetchall()
    return render_template("view task status.html",val=s)
@app.route('/viewtask')
def viewtask():
    p_id=session['lid']
    cmd.execute("SELECT*FROM view_task_status WHERE p_id='"+str(p_id)+"' AND `status`='pending' ")
    s=cmd.fetchall()
    return render_template("view tasks.html",val=s)

@app.route('/updatetestatus',methods=['get','post'])
def updatetestatus():
    id=request.args.get('id')
    print(id)
    session['sid']=id
    return render_template("updatetestatus.html")

@app.route('/updatetestatus1',methods=['get','post'])
def updatetestatus1():

    ssid=session['sid']
    status=request.form['textfield']
    cmd.execute("update view_task_status set status='"+status+"' where w_id='"+str(ssid)+"' ")

```



```

con.commit()
return '<script>alert("update success");window.location="/police_home"</script>'

@app.route('/crimepattern')
@login_required
def crimepattern():
    cmd.execute("SELECT DISTINCT pattern FROM `criminal_list` union SELECT DISTINCT `crime_type` FROM `dataset`")
    s = cmd.fetchall()
    return render_template('crimepattern.html', p=s)

@app.route('/crime_pattern', methods=['GET', 'POST'])
def crime_pattern():
    pattern= request.form['select']
    cmd.execute("SELECT `crime_type`.`crime_type`, `criminal_list`.`criminal_name`, `criminal_list`.`place` FROM `criminal_list` JOIN `crime_type` ON `crime_type`.`id`='criminal_list`.`crime` WHERE `pattern`='"+pattern+"'")
    results = cmd.fetchall()

    cmd.execute("SELECT * FROM `dataset` WHERE `crime_type`='"+pattern+"'")
    results1 = cmd.fetchall()

    roww=[]
    for i in results:
        roww.append(i)
        print(i)
    for i in results1:
        rr = [i[2], i[3], i[1]]
        print(rr,i[4])
        for jj in range(int(i[4])):
            rr=[i[2],i[3],i[1]]
            roww.append(rr)
    print(len(roww))
    ap = aprioris()
    accident = ap.aprioril(roww)

    ss = list(accident.split("frozenset"))
    print("-----")

    print(ss)
    li = []
    i = 0
    for s in ss:
        sss = s.split(',')
        if len(sss) > 2:
            s1 = s.replace('[', '')
            li.append(s1)
        i = i + 1
    print(li)
    lii = str(li)
    ha = []
    ha = lii.replace("'", '')
    ha1 = []
    ha1.replace("'", '')

    ha2 = []
    ha2 = ha1.replace('{', '')

    ha3 = ha2.replace('[', '')

    ha4 = ha3.replace(']', '')
    print("ha4444", ha4)

```

```

hh = []
hh = ha4.split(',')
print("hhhh", hh)

return render_template('result2.html', val=hh)

@app.route('/add_crime_type',methods=['get','post'])
def add_crime_type():
    return render_template("add_crimetype.html")

@app.route('/manage_crimetype',methods=['get','post'])
@login_required
@login_required
def manage_crimetype():
    cmd.execute("SELECT * FROM `crime_type`")
    s=cmd.fetchall()

    return render_template("manage_crimetype.html",val=s)

@app.route('/dltype',methods=['get','post'])
def dltype():
    id=request.args.get('id')
    cmd.execute("DELETE FROM `crime_type` WHERE `id`='"+str(id)+"'")
    con.commit()
    return '''<script>alert("success");window.location="/manage_crimetype"</script>'''

@app.route('/view_reported_criminal',methods=['get','post'])
@login_required
def view_reported_criminal():

    cmd.execute("SELECT `signup`.* , `criminal_list`.* FROM `criminal_list` JOIN `report_criminal` ON
        `report_criminal`.`criminalid`='criminal_list`.`criminal_id` JOIN `signup` ON
        `signup`.`login_id`='report_criminal`.`userid`"
    s=cmd.fetchall()
    return render_template("view_reported_criminals.html",val=s)

@app.route('/viewcases')
def viewcases():
    cmd.execute("SELECT complaint . * , `police_statiion` . NAME , `request` . `request` , `request` . `date` , request.id FROM
        `complaint` JOIN allocation ON complaint.complaint_id=allocation.complaint_id JOIN `police_statiion` ON
        `police_statiion`.login_id=allocation.police_id JOIN login ON `police_statiion`.login_id=login.id JOIN request ON
        `allocation`.`complaint_id`='request'.cid WHERE `request`.fid='"+str(session['lid'])+"'"
    s=cmd.fetchall()
    return render_template('case.html',v=s)

@app.route('/send_request_update',methods=['get','post'])
def send_request_update():
    status=request.form['textfield']
    cmd.execute("UPDATE `request` SET `status`='"+status+"' WHERE `id`='"+str(session['rid'])+"'")
    con.commit()
    return '''<script>alert("success");window.location="/viewcases"</script>''

@app.route('/statusupdate')
def statusupdate():
    id=request.args.get('id')
    session['rid']=id
    return render_template('status.html')

@app.route('/home')
def home():
    return render_template('home.html')

```



```

@app.route('/reportupload')
def reportupload():
    return render_template('uploadreport.html')
@app.route('/reports')
def reports():
    return render_template('report.html')
@app.route('/historyview')
def historyview():
    return render_template('viewhistory.html')
@app.route('/allocatepolice')
def allocatepolice():
    con = pymysql.connect(host="localhost", user="root", password="", port=3306, db="criminal face identification")
    cmd = con.cursor()
    id=request.args.get('id')
    session['cid']=id
    cmd.execute("select * from police_statitoin join login on login.id=police_statitoin.login_id where login.type='police'")
    s=cmd.fetchall()
    return render_template('allocate police.html',val=s)

@app.route('/add_allocate_police',methods=['get','post'])
def add_allocate_police():
    Police=request.form['select']
    cmd.execute("INSERT INTO `allocation` VALUES(NULL,'" +session['cid']+ "','" +Police+ "','pending',curdate())")
    con.commit()
    cmd.execute("update COMPLAINT set reply='assigned' where complaint_id='"+str(session['cid'])+"'")
    con.commit()
    return """<script>alert("success");window.location="/COMPLAINT"</script>"""

@app.route('/request_to_forensic',methods=['get','post'])
@login_required
def request_to_forensic():
    forensic=request.form['select']
    req=request.form['textarea']
    cmd.execute("insert into request
        values(null,'" +str(session['cid'])+ "','" +str(forensic)+ "','" +req+"',curdate(),'pending')")
    con.commit()
    cmd.execute("UPDATE `complaint` SET `reply`='assigned' WHERE `complaint_id`='"+str(session['cid'])+"'")
    con.commit()
    return """<script>alert("Assigned");window.location="/view_allocated_case"</script>"""

@app.route('/upld_evidence')
@login_required
def upld_evidence():
    id=request.args.get('id')
    session['cid']=id
    return render_template('police evidence upload.html')
@app.route('/upld_evidence1',methods=['get','post'])
@login_required
def upld_evidence1():
    evidence=request.files['file']
    fn=secure_filename(evidence.filename)
    path=r'./static/evidence'
    evidence.save(os.path.join(path,fn))
    des=request.form['textarea']
    cmd.execute("INSERT INTO `evidence` VALUES(NULL,'" +str(session['cid'])+ "','" +fn+"','"+des+"',CURDATE())")
    cid=con.insert_id()
    typee="police evidence"
    report=fn
    con.commit()

```

```

with open(compiled_contract_path) as file:
    contract_json = json.load(file) # load contract info as JSON
    contract_abi = contract_json['abi'] # fetch contract's abi - necessary to call its functions
    date = datetime.now().strftime("%Y-%m-%d")
    contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi)
    blocknumber = web3.eth.get_block_number()
    message2 = contract.functions.report_info(blocknumber + 1, int(cid), typee, report, date).transact()
    print(blocknumber,"+++++++=+=====")
    print(message2)

return '''<script>alert("Evidence Uploaded");window.location="/view_allocated_case"</script>''

@app.route('/upload_report')
def upload_report():
    cmd.execute(
        "SELECT
            `complaint`.'complaint','request'.'request','police_statiion'.'name','request'.'date','request'.'status','complaint'.'complaint_id'
        FROM `complaint` JOIN `request` ON `request`.'cid'='complaint'.'complaint_id' JOIN `allocation` ON
        `allocation`.'complaint_id'='complaint'.'complaint_id' JOIN `police_statiion` ON
        `police_statiion`.'login_id'='allocation'.'police_id'")
    res = cmd.fetchall()
    print(res)
    return render_template("uploadreport.html", val=res)

@app.route('/uploadreport')
def uploadreport():
    id=request.args.get('id')
    session['compid']=id
    return render_template("upload_report.html")

@app.route('/uploadreport1',methods=['post'])
def uploadreport1():
    report=request.files['file']
    imgnm=secure_filename(report.filename)
    report.save(os.path.join('static/report',imgnm))
    cmd.execute("INSERT INTO `report` VALUES(NULL,'" +session['compid']+ "','" +imgnm+"',CURDATE())")
    cid=con.insert_id()
    typee="forensic report"
    report=imgnm
    con.commit()

    with open(compiled_contract_path) as file:
        contract_json = json.load(file) # load contract info as JSON
        contract_abi = contract_json['abi'] # fetch contract's abi - necessary to call its functions
        date = datetime.now().strftime("%Y-%m-%d")
        contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi)
        blocknumber = web3.eth.get_block_number()
        message2 = contract.functions.report_info(blocknumber + 1, int(cid), typee, report, date).transact()
        print(blocknumber,"+++++++=+=====")
        print(message2)

    return '''<script>alert("uploaded");window.location="/upload_report"</script>''

if (__name__=="__main__"):
    app.run(debug=True)

```

Database Design

Attribute Name	Datatype	Length	Description
id	Integer	11	Primary Key
username	Varchar	20	
password	Varchar	20	
type	Varchar	20	

Table A.1: Login

Attribute Name	Datatype	Length	Description
id	Integer	10	Primary Key
login-id	integer	10	
fname	varchar	50	
gender	varchar	50	
place	varchar	50	
post	varchar	50	
pin	Integer	10	
email	varchar	50	
phone-no	bigint	20	
lname	varchar	60	

Table A.2: Signup

Attribute Name	Datatype	Length	Description
type-id	Integer	11	Primary Key
crime-type	Varchar	50	
description	Varchar	50	

Table A.3: Crime type

Attribute Name	Datatype	Length	Description
id	Integer	11	Primary Key
complaint-id	Integer	11	
police-id	Integer	11	
status	Varchar	20	
date	date		

Table A.4: Allocation

Attribute Name	Datatype	Length	Description
id	Integer	11	Primary Key
place	Varchar	100	
crime-type	Varchar	100	
name	Varchar	30	
count	Integer	11	

Table A.5: Dataset

Attribute Name	Datatype	Length	Description
criminal _i d	Integer	11	Primary Key
criminal _n ame	Varchar	50	
phone	bigint	20	
place	Varchar	50	
photo	Varchar	50	
crime	Varchar	100	
date _o f _r egistration	Varchar	15	
pattern	Varchar	100	

Table A.6: Criminal list

Attribute Name	Datatype	Length	Description
complaint _i d	Integer	11	Primary Key
u _i d	Varchar	50	Primary Key
date	Varchar	90	
complaint	tinytext		
reply	Varchar	200	

Table A.7: Complaint

Attribute Name	Datatype	Length	Description
$p_i d$	Integer	11	Primary Key
name	Varchar	50	
phone	bigint	100	
email	Varchar	50	
place	Varchar	50	
pincode	Varchar	25	Primary Key
$login_i d$	Integer	11	

Table A.8: Police station

Attribute Name	Datatype	Length	Description
rid	Integer	11	Primary Key
criminalid	Integer	11	
userid	Integer	11	
date	date		

Table A.9: Report criminal

Attribute Name	Datatype	Length	Description
id	Integer	11	Primary Key
cid	Integer	11	
evidence	Varchar	400	
description	text		
date	date		

Table A.10: Evidence

Attribute Name	Datatype	Length	Description
id	Integer	11	Primary Key
compid	Integer	11	
report	Varchar	76	
date	date		

Table A.11: Report

Attribute Name	Datatype	Length	Description
id	Integer	11	Primary Key
cid	Integer	11	
fid	Integer	11	
request	Varchar	100	
date	date		
status	Varchar	200	

Table A.12: Request

Dataflow Diagram

Level 0

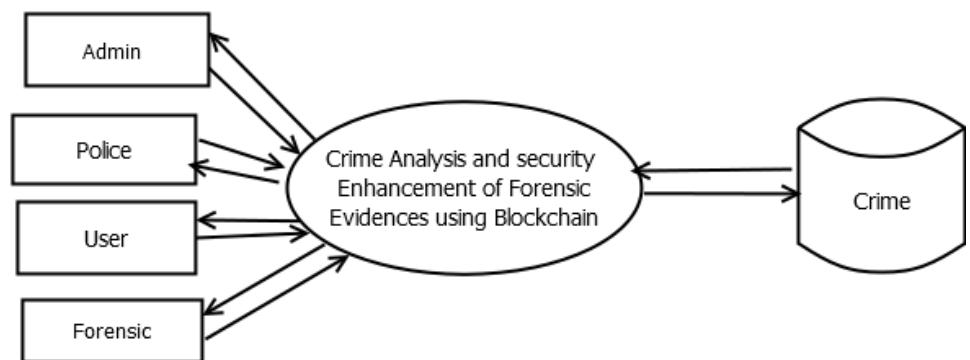


Figure A.1: Dataflow Diagram

Level 1.1

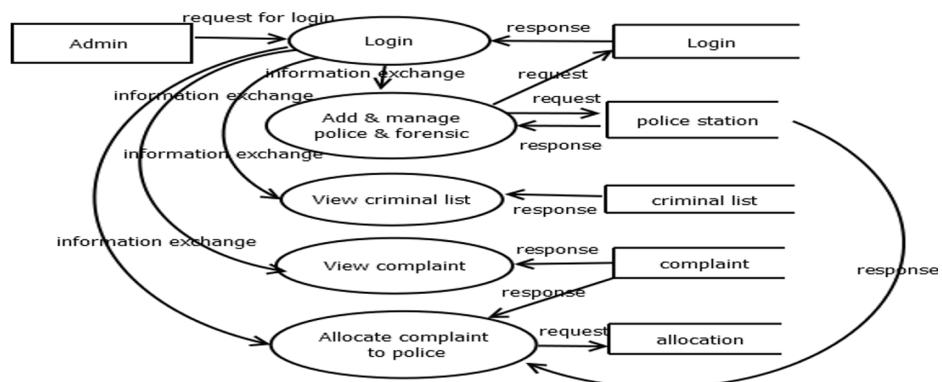


Figure A.2: Dataflow Diagram

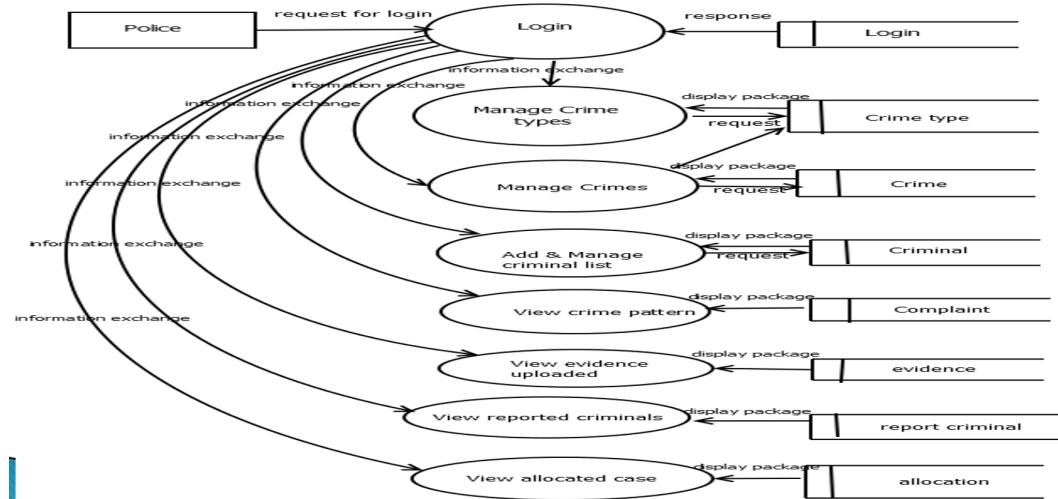
Level 1.2

Figure A.3: Dataflow Diagram

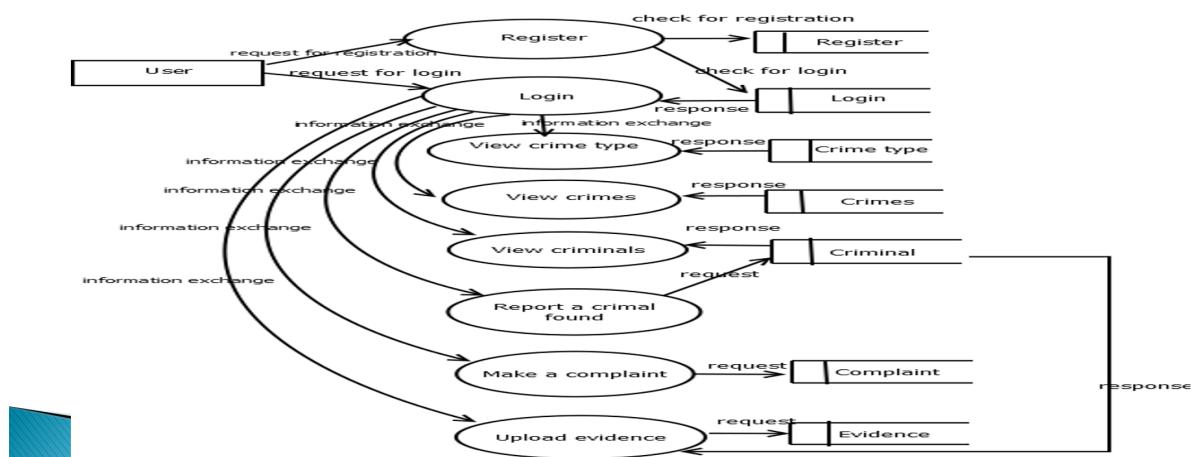
Level 1.3

Figure A.4: Dataflow Diagram

Level 1.4

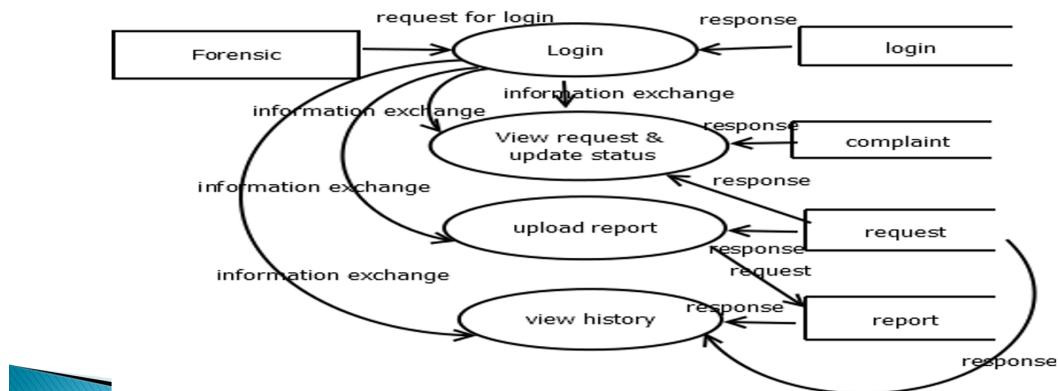


Figure A.5: Dataflow Diagram