Visvesvaraya Technological University
Belagavi-590 018 Karnataka

A Mini Project Report on

"Forensic Investigation Management System"

Mini Project Report submitted in partial fulfilment of the requirement for the DBMS Laboratory with Mini Project
[BCS403]

Bachelor of Engineering
In
Artificial Intelligence and Machine Learning

Submitted by

Jyothish S	**1JT22AI016**
Khushi S Sorathia	**1JT22AI018**
Prabhava R Bhat	**1JT22AI034**
Monisha Bharadwaj M H	**1JT22AI062**

Under the guidance of

Mrs. Ushashri Gunti
Assistant Professor,
AIML Dept.

Mr. S Vinod Kumar
Assistant Professor
AIML Dept

Department of Artificial Intelligence and Machine Learning
Jyothy Institute of Technology
Tataguni,Bengaluru-560082

Jyothy Institute of Technology

Tataguni, Bengaluru-560082

Department of Computer Science and Engineering

CERTIFICATE

Certified that the mini project work entitled **“Forensic Investigation Managemant System”** carried out by **Jyothish S [1JT22AI016], Khushi S Sorathia [1JT22AI018], Prabhava R Bhat[1JT22AI034] and Monisha Bharadwaj M H [1JT22AI062]** bonafide students of Jyothy Institute Technology, in partial fulfilment for the award of **Bachelor of Engineering** in **Artificial Intelligence and Machine Learning** department of the **Visvesvaraya Technological University, Belagavi** during the year **2023-2024**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Mrs. Ushasri Gunti	**Mr. S Vinod Kumar**
Assistant Professor	**Assistant Professor**
Dept of –AI&ML	**Dept of –AI&ML**
JIT, Bengaluru	**JIT, Bengaluru**
External Examiner	Signature with Date:

1.

2

ACKNOWLEDGEMENT

Firstly, we are very grateful to this esteemed institution **"Jyothy Institute of Technology**" for providing us an opportunity to complete our project.

We express our sincere thanks to our **Principal Dr. Gopalakrishna K** for providing us with adequate facilities to undertake this project.

We would like to thank **Dr. Madhu B R, Professor and Head of the Artificial Intelligence and Machine Learning Department** for providing his valuable support.

We would like to thank our guide **Dr. Ushasri Gunti, Assistant Professor** for her keen interest and guidance in completing this project.

Finally, we would like to thank all our friends who have helped us directly or indirectly in this project.

Jyothish S [1JT22AI016]

Khushi S Sorathia [1JT22AI018]

Prabhava R Bhat [1JT22AI034]

Monisha Bharadwaj M H [1JT22AI062]

ABSTRACT

Database Forensic Investigation (DBFI) is a specialized area within Digital Forensics (DF) dedicated to the meticulous examination of database contents. The core objective of DBFI is to identify, collect, preserve, reconstruct, analyze, and report database-related incidents. This field is crucial in understanding and mitigating the impact of unauthorized activities within databases, often involving intricate forensic processes to uncover and interpret complex data interactions and transactions.

In the context of our project, DBFI is applied to the investigation of murder cases spanning various time periods. The database is designed to capture and document the essential details of each case, along with comprehensive investigative information. This structured approach ensures that every aspect of a case is meticulously recorded, facilitating a thorough analysis and reconstruction of events.

The database schema includes several key entities: employees, victims, suspects, and police officers. Each entity is characterized by specific attributes that provide detailed information relevant to the investigation. For instance, the 'Employee' entity includes attributes such as employee ID, first name, last name, age, and sex, which are crucial for identifying and linking individuals involved in the investigative process. Similarly, the 'Victim' entity captures vital details like victim ID, first name, last name, age, sex, and circumstances of death, including the date and details of the incident.

A central aspect of the DBFI project is the relationship between these entities, which is pivotal in reconstructing a comprehensive and chronological timeline of events. For example, the 'Victim' entity is connected to the 'Murderer' entity, which details the killer's information, including the weapon used, and the victim ID to establish the linkage between the victim and the perpetrator. The 'Evidence' entity plays a critical role in the investigation, encompassing various attributes such as DNA, eye colour, hair colour, skin colour, and blood group, which help in identifying and confirming suspect involvement.

The 'Suspects' entity includes suspect ID, first name, last name, age, sex, and suspicion details, forming a crucial part of the investigative process. Additionally, the 'Police' entity, which includes police ID, first name, last name, age, and sex, represents the law enforcement officers responsible for investigating the cases. Relationships such as 'investigates' and 'logs in' between these entities ensure a coherent flow of information, aiding in the systematic and logical progression of the investigation.

By leveraging DBFI, investigators can construct a detailed and accurate narrative of intruder activities, leading to the resolution of complex forensic cases. This approach not only enhances the efficiency and accuracy of forensic investigations but also ensures that all critical aspects of a case are thoroughly examined and documented.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Introduction to DBMS

A database is simply an organized collection of related data, typically stored on disk, and accessible by many concurrent users, it is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

Databases are managed by a Database Management System (DBMS) which is a collection of programs that enables user to create and maintain a database.

Advantages of DBMS:

- Redundancy is controlled.
- Unauthorized access is restricted.
- Providing multiple user interfaces.
- Enforcing integrity constraints.
- Providing backup and recovery.

1.2 Introduction to SQL

Structured Query Language (SQL), is a language used to request data from a database which includes database creation, deletion, and retrieval of required tables and even manipulation of data held in a relational database management system.

SQL is considered as a Non-Procedural or a High-level language in which the expected result or operation is given without the specific details about how to accomplish the task. So, SQL is a declarative language.

Therefore, SQL is designed at a higher conceptual level of operation than procedural languages as procedural languages include only the information about opening and closing tables, loading and searching indexes, or flushing buffers and writing data to file systems, but the lower level logical and physical operations are not specified in SQL.

CHAPTER 2
DESIGN

2.1 Theory of ER Diagram

The Entity–Relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**

An **Entity Relationship Diagram (ERD)** shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data.

An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of database. ER diagrams are used to sketch out the design of a database.

2.2 ENTITIES

An entity is an 'object' in the real world with an independent existence and an entity type defines a collection (or set) of entities that have the same attributes. Each entity type in the database is described by its name and attributes.

An entity type is represented in ER diagrams as a rectangular box enclosing the entity type name.

2.3 RELATIONSHIPS

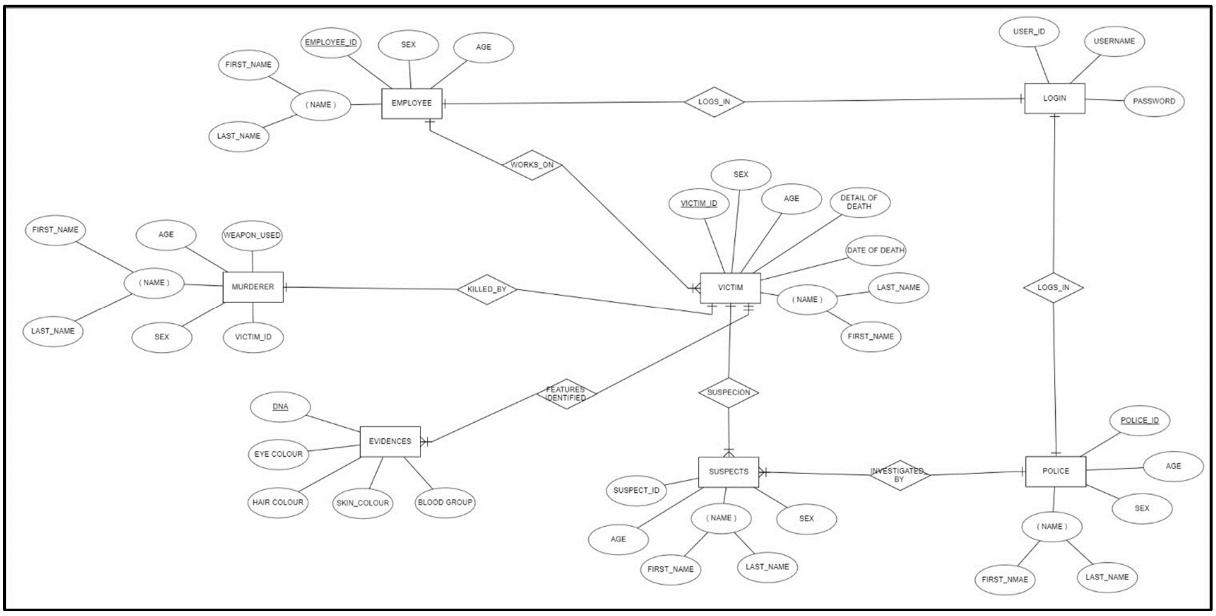
A relational database collects different types of datasets that use tables, records, and columns. It is used to create a well defined relationship between database tables so that relational database can be easily stored. For example say we need to have a connection between the two entities such as staff and customer we can connect them using the relationship say staff serves customer where serves is the relation that exists between them.

2.4 ATTRIBUTES

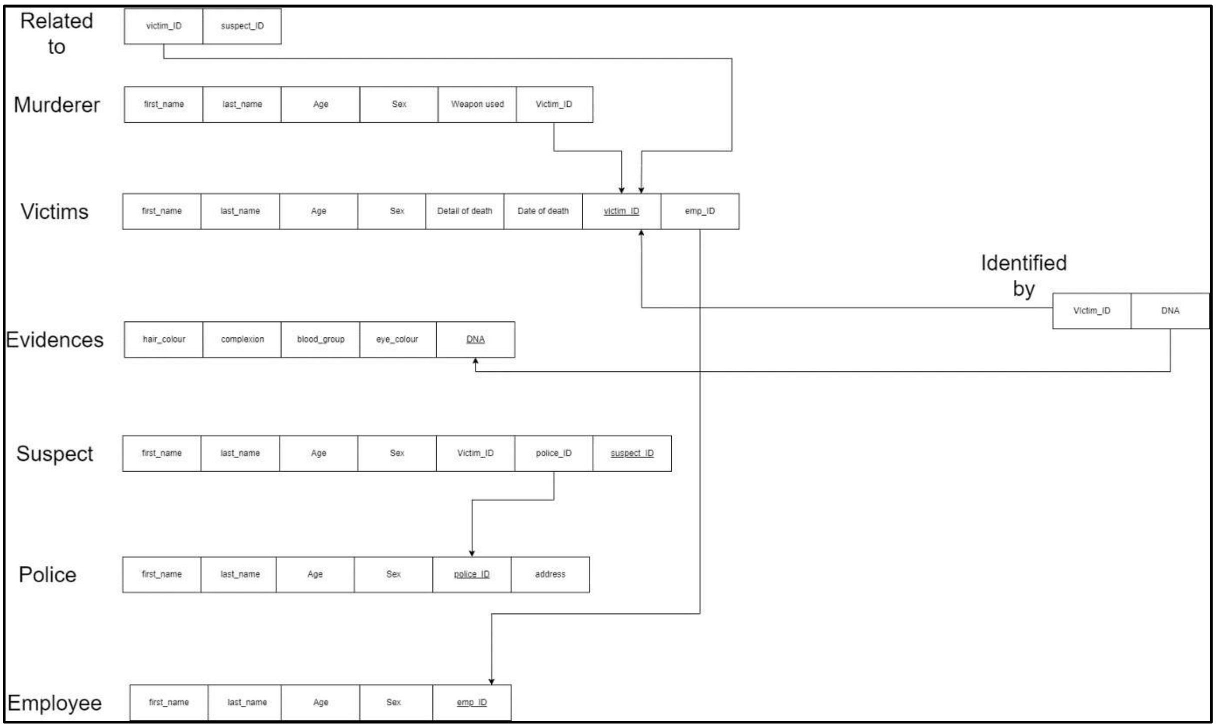
An attribute represents some property of interest that further describes an entity and he column header of the table shows the attributes. Each attribute in a table has a certain domain which allows it to accept a certain 'set of values' only.

The attribute values, of each entity, will define its characteristics in the table and is represented by oval in the ER diagram

ER Diagram:



Relational schema:



Related to

victim_ID	suspect_ID

Murderer

first_name	last_name	Age	Sex	Weapon used	Victim_ID

Victims

first_name	last_name	Age	Sex	Detail of death	Date of death	victim_ID	emp_ID

Identified by

Victim_ID	DNA

Evidences

hair_colour	complexion	blood_group	eye_colour	DNA

Suspect

first_name	last_name	Age	Sex	Victim_ID	police_ID	suspect_ID

Police

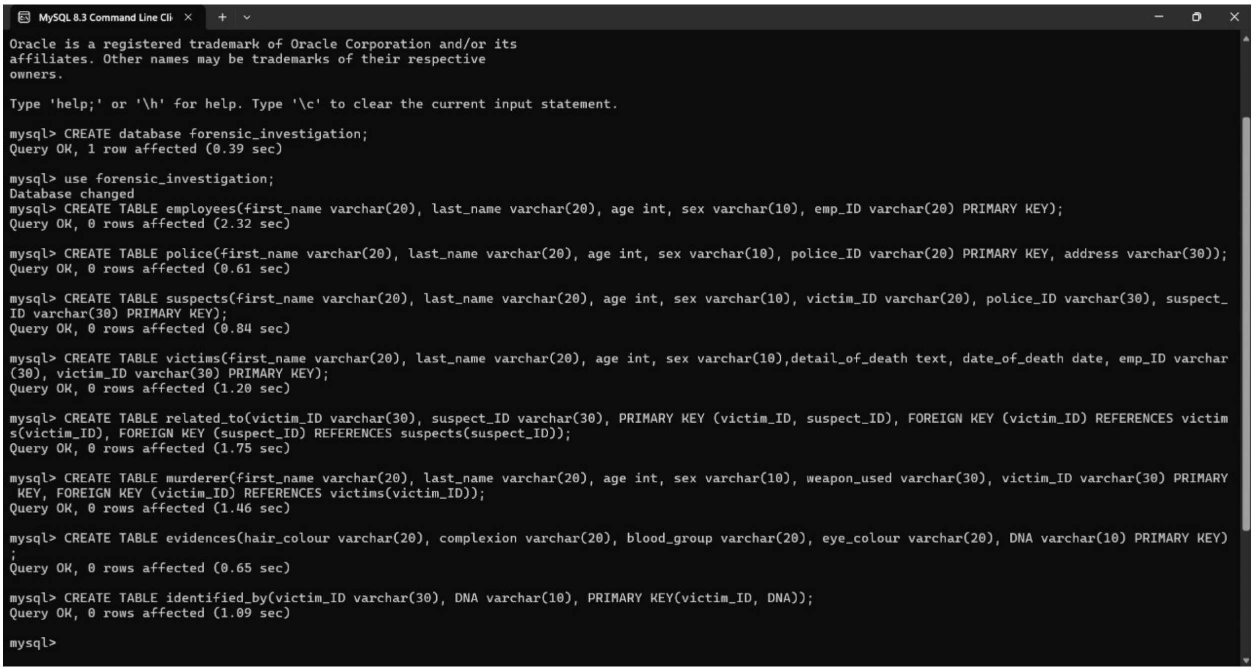
first_name	last_name	Age	Sex	police_ID	address

Employee

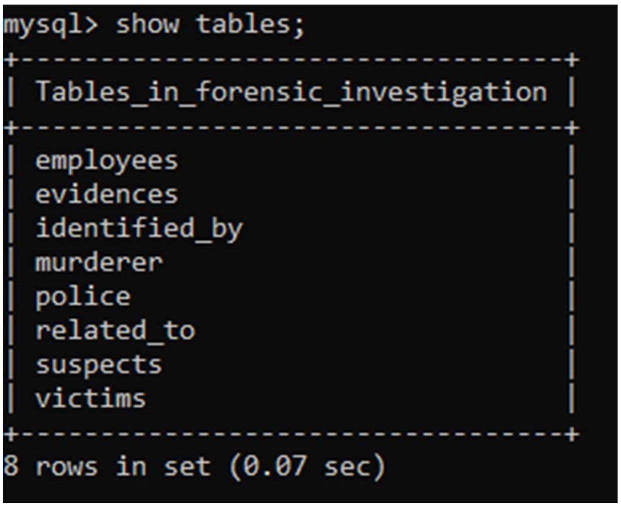
first_name	last_name	Age	Sex	emp_ID

CHAPTER 3
IMPLEMENTATION

3.1 TABLE CREATION

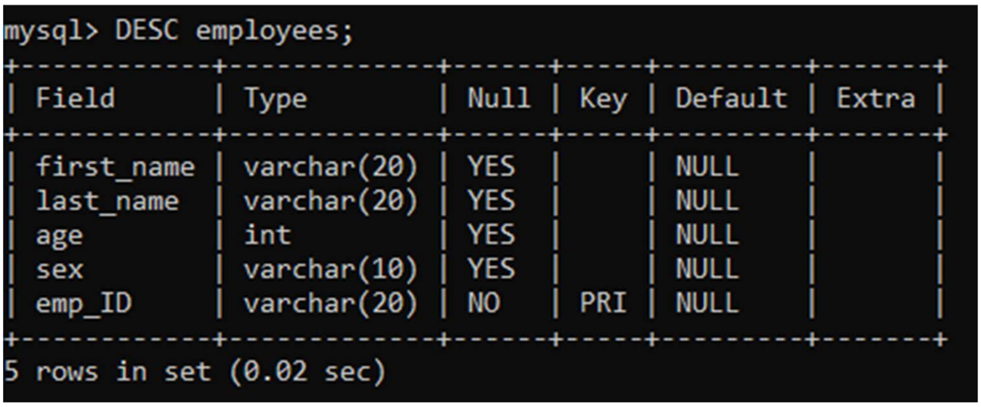


```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE database forensic_investigation;
Query OK, 1 row affected (0.39 sec)

mysql> use forensic_investigation;
Database changed
mysql> CREATE TABLE employees(first_name varchar(20), last_name varchar(20), age int, sex varchar(10), emp_ID varchar(20) PRIMARY KEY);
Query OK, 0 rows affected (2.32 sec)

mysql> CREATE TABLE police(first_name varchar(20), last_name varchar(20), age int, sex varchar(10), police_ID varchar(20) PRIMARY KEY, address varchar(30));
Query OK, 0 rows affected (0.61 sec)

mysql> CREATE TABLE suspects(first_name varchar(20), last_name varchar(20), age int, sex varchar(10), victim_ID varchar(20), police_ID varchar(30), suspect_
ID varchar(30) PRIMARY KEY);
Query OK, 0 rows affected (0.84 sec)

mysql> CREATE TABLE victims(first_name varchar(20), last_name varchar(20), age int, sex varchar(10),detail_of_death text, date_of_death date, emp_ID varchar
(30), victim_ID varchar(30) PRIMARY KEY);
Query OK, 0 rows affected (1.20 sec)

mysql> CREATE TABLE related_to(victim_ID varchar(30), suspect_ID varchar(30), PRIMARY KEY (victim_ID, suspect_ID), FOREIGN KEY (victim_ID) REFERENCES victim
s(victim_ID), FOREIGN KEY (suspect_ID) REFERENCES suspects(suspect_ID));
Query OK, 0 rows affected (1.75 sec)

mysql> CREATE TABLE murderer(first_name varchar(20), last_name varchar(20), age int, sex varchar(10), weapon_used varchar(30), victim_ID varchar(30) PRIMARY
 KEY, FOREIGN KEY (victim_ID) REFERENCES victims(victim_ID));
Query OK, 0 rows affected (1.46 sec)

mysql> CREATE TABLE evidences(hair_colour varchar(20), complexion varchar(20), blood_group varchar(20), eye_colour varchar(20), DNA varchar(10) PRIMARY KEY)
;
Query OK, 0 rows affected (0.65 sec)

mysql> CREATE TABLE identified_by(victim_ID varchar(30), DNA varchar(10), PRIMARY KEY(victim_ID, DNA));
Query OK, 0 rows affected (1.09 sec)

mysql>
```

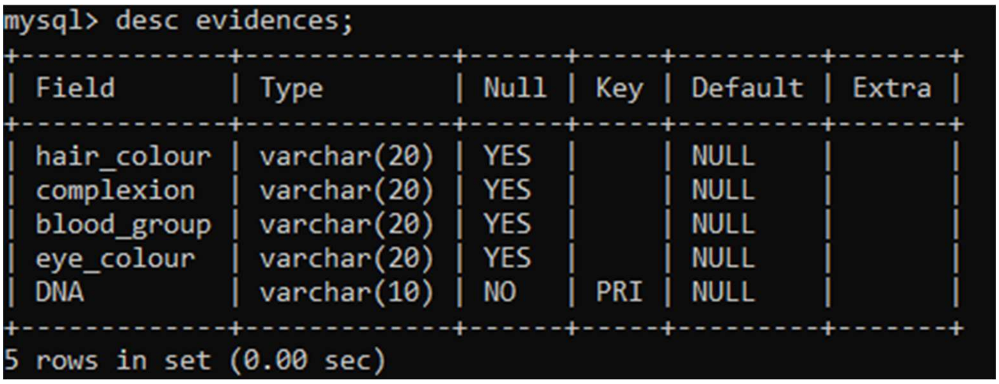


```
mysql> show tables;
+--------------------------------+
| Tables_in_forensic_investigation |
+--------------------------------+
| employees                        |
| evidences                        |
| identified_by                    |
| murderer                         |
| police                           |
| related_to                       |
| suspects                         |
| victims                          |
+--------------------------------+
8 rows in set (0.07 sec)
```

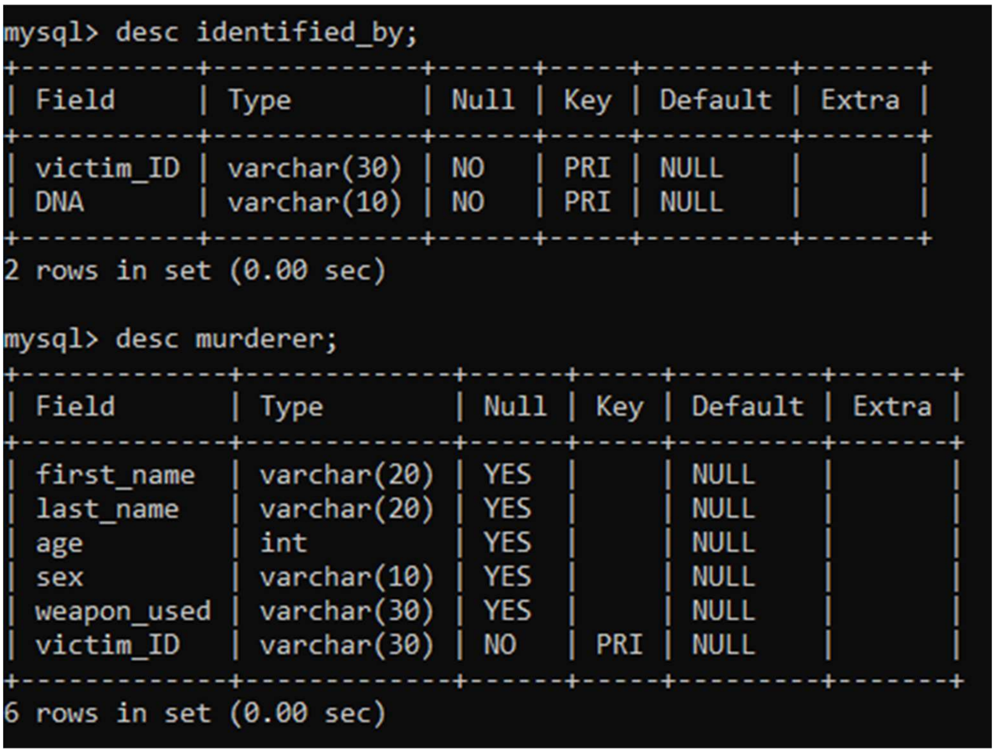
3.1 TABLE DESCPTION



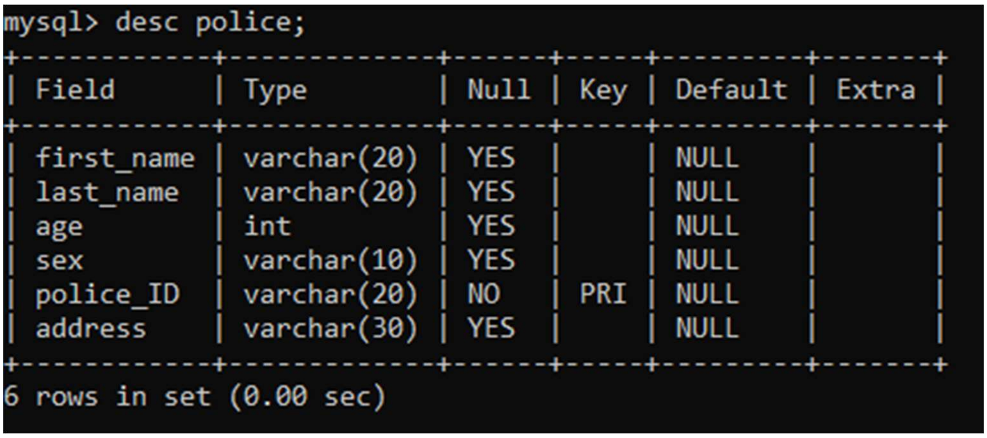
```
mysql> DESC employees;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| first_name | varchar(20) | YES  |     | NULL    |       |
| last_name  | varchar(20) | YES  |     | NULL    |       |
| age        | int         | YES  |     | NULL    |       |
| sex        | varchar(10) | YES  |     | NULL    |       |
| emp_ID     | varchar(20) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.02 sec)
```



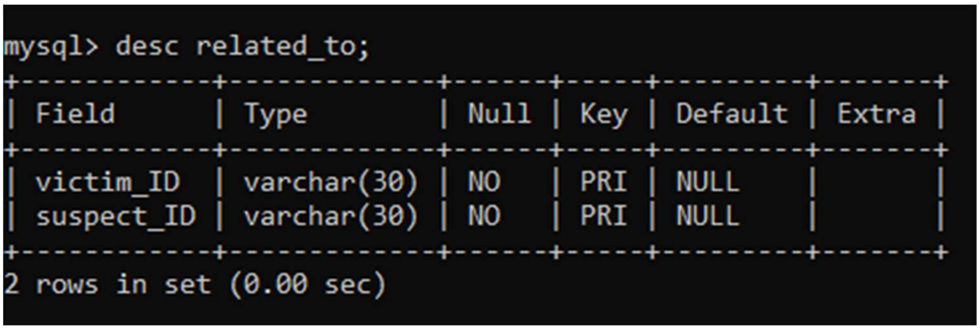
```
mysql> desc evidences;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| hair_colour | varchar(20) | YES  |     | NULL    |       |
| complexion  | varchar(20) | YES  |     | NULL    |       |
| blood_group | varchar(20) | YES  |     | NULL    |       |
| eye_colour  | varchar(20) | YES  |     | NULL    |       |
| DNA         | varchar(10) | NO   | PRI | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```



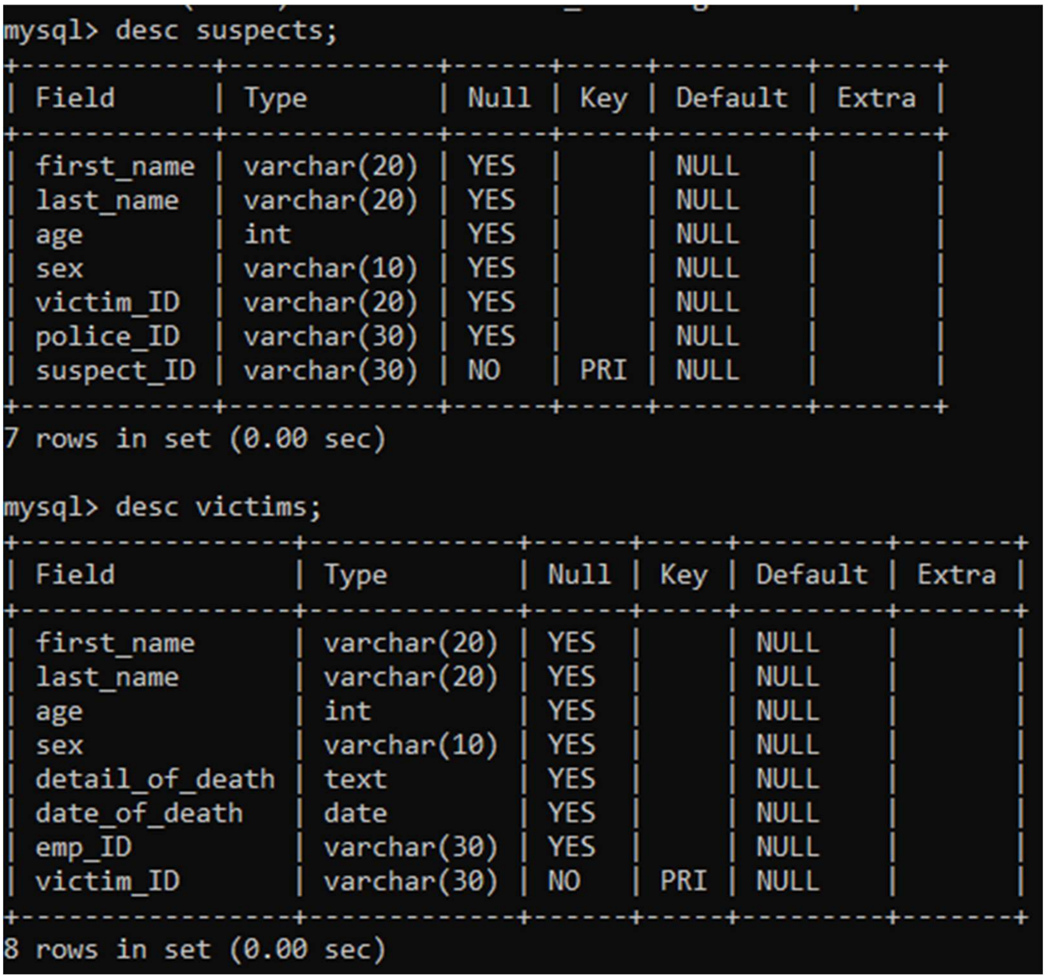
```
mysql> desc identified_by;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| victim_ID | varchar(30) | NO   | PRI | NULL    |       |
| DNA       | varchar(10) | NO   | PRI | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> desc murderer;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| first_name  | varchar(20) | YES  |     | NULL    |       |
| last_name   | varchar(20) | YES  |     | NULL    |       |
| age         | int         | YES  |     | NULL    |       |
| sex         | varchar(10) | YES  |     | NULL    |       |
| weapon_used | varchar(30) | YES  |     | NULL    |       |
| victim_ID   | varchar(30) | NO   | PRI | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```



```
mysql> desc police;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| first_name | varchar(20) | YES  |     | NULL    |       |
| last_name  | varchar(20) | YES  |     | NULL    |       |
| age        | int         | YES  |     | NULL    |       |
| sex        | varchar(10) | YES  |     | NULL    |       |
| police_ID  | varchar(20) | NO   | PRI | NULL    |       |
| address    | varchar(30) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```



```
mysql> desc related_to;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| victim_ID  | varchar(30) | NO   | PRI | NULL    |       |
| suspect_ID | varchar(30) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

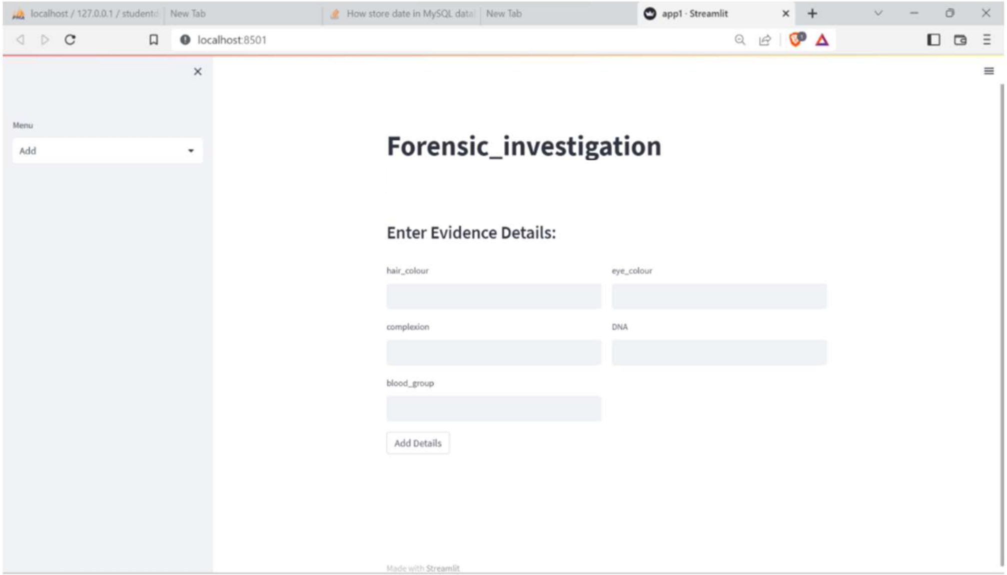


```
mysql> desc suspects;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| first_name | varchar(20) | YES  |     | NULL    |       |
| last_name  | varchar(20) | YES  |     | NULL    |       |
| age        | int         | YES  |     | NULL    |       |
| sex        | varchar(10) | YES  |     | NULL    |       |
| victim_ID  | varchar(20) | YES  |     | NULL    |       |
| police_ID  | varchar(30) | YES  |     | NULL    |       |
| suspect_ID | varchar(30) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

mysql> desc victims;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| first_name      | varchar(20) | YES  |     | NULL    |       |
| last_name       | varchar(20) | YES  |     | NULL    |       |
| age             | int         | YES  |     | NULL    |       |
| sex             | varchar(10) | YES  |     | NULL    |       |
| detail_of_death | text        | YES  |     | NULL    |       |
| date_of_death   | date        | YES  |     | NULL    |       |
| emp_ID          | varchar(30) | YES  |     | NULL    |       |
| victim_ID       | varchar(30) | NO   | PRI | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

CHAPTER 4
RESULT AND SNAPSHOTS

Frontend:

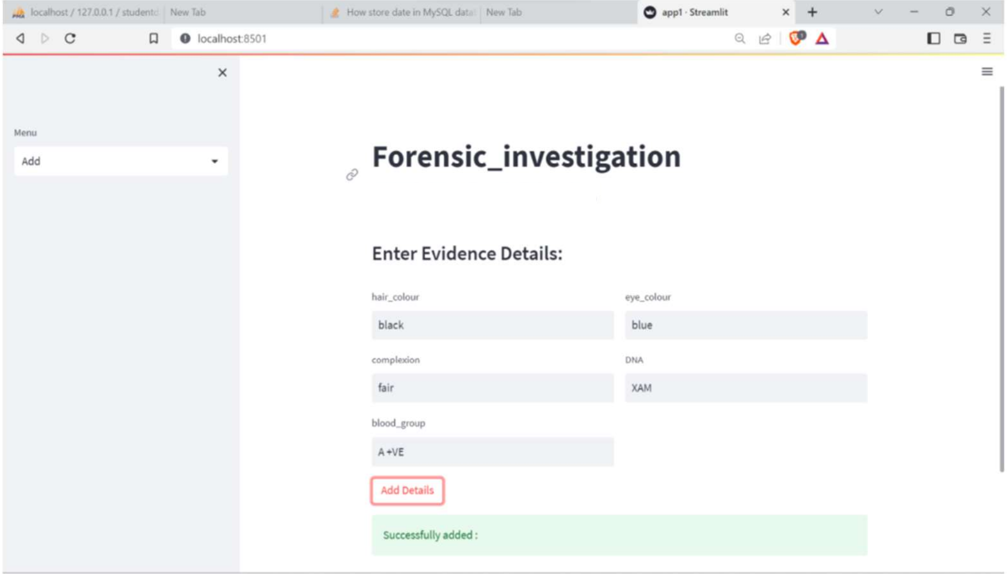
1. Addition, Modification and Deletion of records from any chosen table



Forensic_investigation

Enter Evidence Details:

A frontend with which we can add table details.



Forensic_investigation

Enter Evidence Details:

hair_colour: black

eye_colour: blue

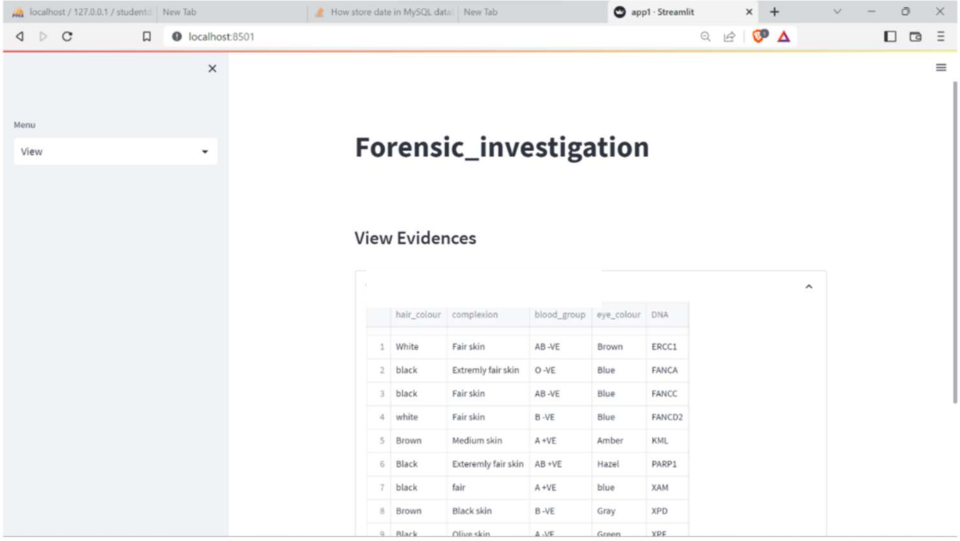
complexion: fair

DNA: XAM

blood_group: A +VE

Successfully added :

A frontend to view the attributes of the table:



Menu

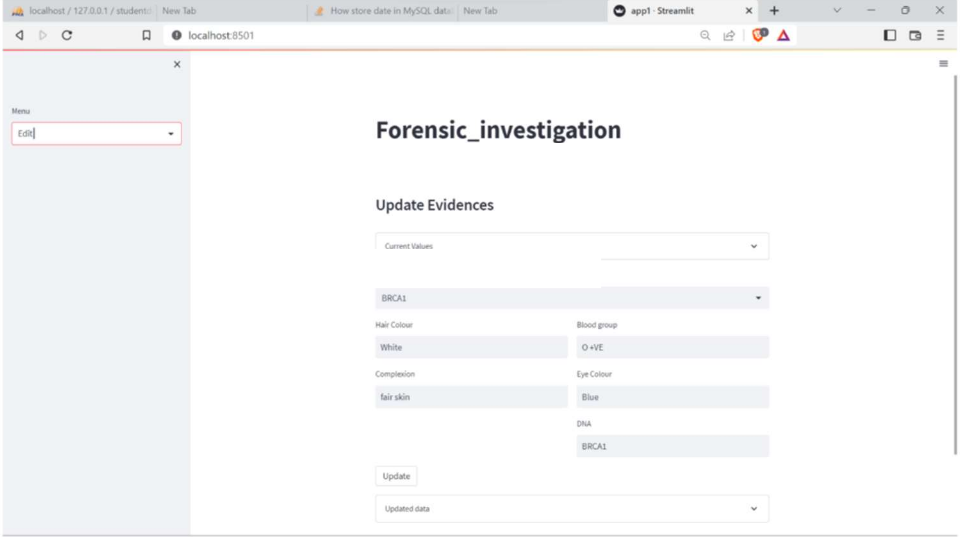
View

Forensic_investigation

View Evidences

	hair_colour	complexion	blood_group	eye_colour	DNA
1	White	Fair skin	AB -VE	Brown	ERCC1
2	black	Extremly fair skin	O -VE	Blue	FANCA
3	black	Fair skin	AB -VE	Blue	FANCC
4	white	Fair skin	B -VE	Blue	FANCD2
5	Brown	Medium skin	A +VE	Amber	KML
6	Black	Exteremly fair skin	AB +VE	Hazel	PARP1
7	black	fair	A +VE	blue	XAM
8	Brown	Black skin	B -VE	Gray	XPD
9	Black	Olive skin	A -VE	Green	XPF

A frontend with which we can update the table details:



Menu

Edit

Forensic_investigation

Update Evidences

Current Values

BRCA1

Hair Colour: White

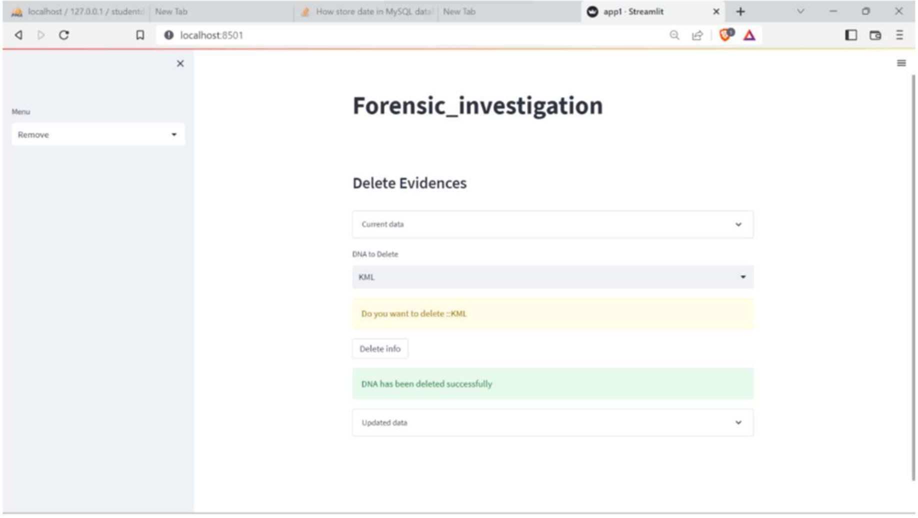
Blood group: O +VE

Complexion: fair skin

Eye Colour: Blue

DNA: BRCA1

A frontend with which we can delete rows from the table:



Menu

Remove

Forensic_investigation

Delete Evidences

Current data

DNA to Delete

KML

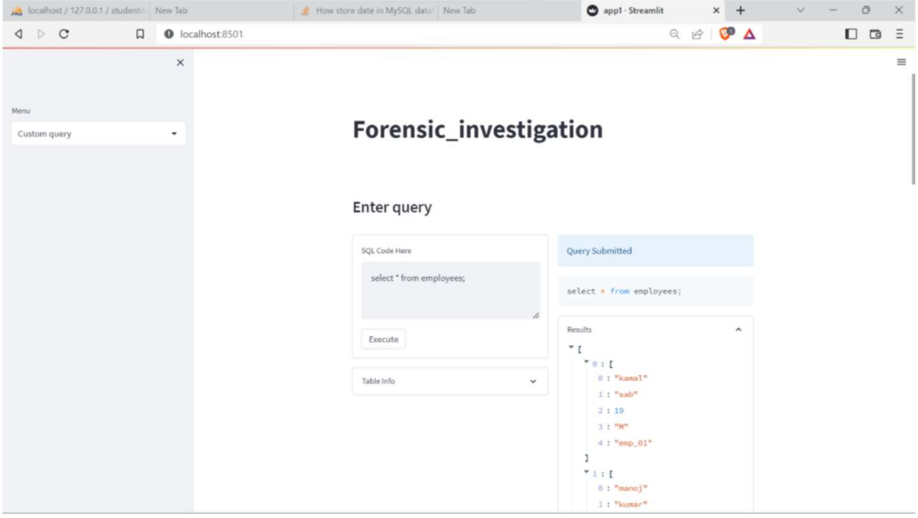
Do you want to delete ::KML

Delete info

DNA has been deleted successfully

Updated data

A window which accepts custom query and displays the result:



Menu

Custom query

Forensic_investigation

Enter query

SQL Code Here

```
select * from employees;
```

Execute

Table Info

Query Submitted

```
select * from employees;
```

Results

```
[
  0 : [
    0 : "kamal"
    1 : "sab"
    2 : 19
    3 : "M"
    4 : "emp_01"
  ]
  1 : [
    0 : "manoj"
    1 : "kumar"
```

CHAPTER 5
CONCLUSION

CONCLUSION

While developing the system a conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources-that would generate a proper system for cases.

While making the system, an eye has been kept on making it as user-friendly. As such one may hope that the system will be acceptable to any user and will adequately meet needs. As in case of any system development process where there are a number of short comings, there have been some short comings in the development of this system also.

5.1 Features

User can sign up and then login. User can add blogs, view all blogs, add and view categories, view published blogs, edit or delete his own blogs.

Admin has full control of the system, he/she can add blogs, view all blogs, add and view categories, view published blogs, edit all published blog, choose featured blog, add and delete user id.

The other main feature is that it also contains an admin area from where he/she can check latest site updates and can maintain all site. While adding new blog the admin has to provide a title, select category, content, photos and date.

From the admin panel, he/she can easily customize the entire website. A responsive dashboard is provided in the admin panel for the easy management of the site.

The admin has complete control over the database and blog, the user can control only his posts. A new category, post, user and admin can be created by admin. A featured post on homepage can be added.

Search bar helps in searching posts. A special feature is added to display posted date and time with username.

Supports multiple media queries. Posts can be viewed category wise.

5.2 References

1. https://github.com/Khushi-Sorathia/Forensics-investigation
2. https://share.streamlit.io/
3. https://www.youtube.com/watch?v=zSRBVxvhriA
4. https://www.youtube.com/results?search_query=streamlit+python+tutorial+mysql