



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

ONLINE CRIME MANAGEMENT

A Project Report Capstone of Project-2

Submitted by

KM. SHAGUFTA FAREED

(16SCSE101091)

*in partial fulfillment for the award of the degree
of*

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Dr. KAVITA, Assoc. Prof.

APRIL / MAY- 2020



**SCHOOL OF COMPUTING AND SCIENCE AND
ENGINEERING**

BONAFIDE CERTIFICATE

Certified that this project report “**Online Crime Management**” is the bonafide work of “**K.M. Shagufta Fareed**” who carried out the project work under my supervision

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean,
**School of Computing Science &
Engineering**

SIGNATURE OF SUPERVISOR

Dr. KAVITA, ASSO.
PROF.
Professor
**School of Computing Science &
Engineering**

ABSTRACT

The proposed system applies to all Police stations across the country and specifically looks into the subject of Crime Management. It is well understood that Crime Prevention, Detection and Conviction of criminals depend on a highly responsive backbone of Information Management. The efficiency of the police function and the effectiveness with which it tackles crime depend on what quality of information it can derive from its existing records and how fast it can have access to it. Online Crime Management System:

The purpose of Online Crime Management System is to automate the existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Online Crime Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	iii
	LIST OF CONTENTS	iv
	LIST OF FIGURES	vi
1	INTRODUCTION	1
	1.1 Overview	1
	1.2 Purpose	2
2	LITERATURE REVIEW	3
3	PROBLEM STATEMENT	4
4	PROPOSED MODEL	5
	4.1 DESIGN	5
	4.1.1 Data Flow Diagrams	5
	4.1.1.1 DFD Symbols	6
	4.1.1.2 Constructing a DFD	7
	4.1.1.3 Salient Feature of DFD's	7
	4.1.1.4 Data Flow:	7
	4.1.2 ER-Diagrams	10
	4.1.3 Unified Modeling Language Diagrams	12
	4.1.4 Use Case Diagrams	13
	4.1.4.1 Use case Description	14
	4.1.5 Sequence Diagram:	17
	4.1.6 Testing Methodologies:	19
	4.1.6.1 Test Planning:	19
	4.1.6.2 Types of Testing:	19

	4.1.7 Technologies used:	20
5	IMPLEMENTATIONS	22
6	RESULTS	31
	CONCLUSIONS /PROJECT SUMMARY	45
	FUTURE IMPROVEMENT	46
	REFERENCES:	47

TABLE OF CONTENTS

FIGURE NAME	PAGE
	NO
Fig 4.1.1.4.1 Zero-level DFD for Online Crime management.	8
Fig 4.1.1.4.2 DFD for User/Complainer registration.	9
Fig 4.1.1.4.3 DFD for a Police Officer.	9
Fig 4.1.2.1 Employee Information.	10
Fig 4.1.2.2 Police Station Master.	11
Fig 4.1.2.3 Victims FIR Master.	11
Fig 4.1.2.4 Login Master.	11
Fig 4.1.4.1 Use-case diagram.	14
Fig 4.1.5.1 Administration Sequence diagram.	17
Fig 4.1.5.2 Investigator Sequence diagram.	18
Fig 4.1.5.3 Writer Sequence diagram.	18
Fig 5.1 Initial Design.	22
Fig 5.2 “crime_portal” Database in phpMyAdmin.	22
Fig.5.3 Structure of “complaint table”	23
Fig.5.4 Structure of “head” table.	23
Fig.5.5 Structure of “police” table.	23
Fig.5.6 Structure of “police_station” table.	24
Fig.5.7 Structure of “update_case” table.	24
Fig.5.8 Structure of “user” table.	24
Fig5.9 Code snippet of home page.	25
Fig.5.10 User-flow.	25
Fig5.11 Code snippet of home page.	26
Fig5.12 Code snippet that implements complaint filing part.	27
Fig5.13 Official login page user flow.	28
Fig5.14 Code snippet of “incharge_login.php”.	28

Fig5.15 Code snippet of “police_login.php”.	29
Fig5.16 Code snippet of “head_login.php”.	30
Fig 6.1 Xampp Control Panel Showing Apache and MySQL server running.	31
Fig 6.2 “crime_portal” Database in phpMyAdmin.	31
Fig 6.3 “complaint” table pre-filled with dummy values.	32
Fig 6.4 “head” table pre-filled with dummy values.	32
Fig 6.5 “police” table pre-filled with dummy values.	32
Fig 6.6 “police_station” table with dummy values.	32
Fig 6.7 “update_case” table with dummy values.	33
Fig 6.8 “user” table with dummy values.	33
Fig 6.1.1 Front page of the “Crime Portal” website.	33
Fig 6.1.2 “Sign Up” page filled with dummy values for demonstration purpose.	34
Fig 6.1.3 Successful Registration.	35
Fig 6.1.4 Successful addition of user represented by black rectangular box.	35
Fig 6.1.5 “User Login” page.	35
Fig 6.1.6 Successful Login.	36
Fig 6.1.7 Logging complaint.	37
Fig 6.1.8 Complaint Logged Successfully.	37
Fig 6.1.9 Successful addition of complaint represented by black rectangular box.	38
Fig 6.1.10 Complaint History Page.	38
Fig 6.1.11 “Official Login” page.	39
Fig 6.1.12 “Incharge Login” page.	40
Fig 6.1.13 “Incharge Complain” page.	40
Fig 6.1.14 “Police Login” page.	41

Fig 6.1.15 “Police Pending Complaint” page.	41
Fig 6.1.16 “Police Complaint Details” Page.	42
Fig 6.1.17 Status updated to “Criminal accepted Crime”.	42
Fig 6.1.18 Successful updating of complaint represented by black rectangular box.	43
Fig 6.1.19 “Head Quarter Login” page.	43
Fig 6.1.20 “Head Quarter Home” page.	44
Fig 6.1.21 “Case Details” page.	44

Chapter 1

INTRODUCTION

1.1 Overview

The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specifications have been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations. The authentication and authorization were crosschecked at all the relevant stages. The user level accessibility has been restricted into two zones namely. The administrative zone and the normal user zone.

Why new system?

- The system at any point of time can provide the details of the police station and the employees.
- The system at any point of time can provide the details of victims and the registered FIR's
- The system at any point of time can provide the details of evidence and their sequence
- The system at any point of time can provide the details of existing charge sheets and their statuses.

The "Online Crime Management System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it

is user-friendly. Online Crime Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

1.2. Purpose

The main objective of the Project on Online Crime Management System is to manage the details of Crime, Criminal, Police Department, Public, Complaint. It manages all the information about Crime, Solutions, Complaint, Crime. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Crime, Criminal, Solutions, Police Department. It tracks all the details about the Police Department, Public, Complaint.

Functionalities provided by Online Crime Management System are as follows:

- Provides the searching facilities based on various factors. Such as Crime, Police Department, Public, Complaint
- Online Crime Management System also manage the Solutions details online for Public details, Complaint details, Crime.
- It tracks all the information of Criminal, Solutions, Public etc.
- Manage the information of Criminal
- Shows the information and description of the Crime, Police Department
- To increase efficiency of managing the Crime, Criminal
- It deals with monitoring the information and transactions of Public.
- Manage the information of Crime.

Chapter 2

LITERATURE SURVEY

Behavioral analysis of crime against women using a graph Based clustering approach.2017 International Conference on Computer Communication and Information (ICCI): Crime against women is increasing at an alarming rate in almost all parts of India and women in the Indian society have been victims of humiliation, torture and exploitation. It has even existed in the past but only in the recent years the issues have been brought to the open for concern. According to the latest data released by the National Crime Records Bureau (NCRB), crimes against women have increased more than doubled over the past ten years. While a number of analyses have been done in the field of crime pattern detection, none have done an extensive study on the crime against women in India. The present paper describes a behavioral analysis of crime against women in India from the year 2001 to 2014. The study evaluates the efficacy of Info map clustering algorithm for detecting communities of states and union territories in India based on crimes. As it is a graph-based clustering approach, all the states of India along with the union territories have been considered as nodes of the graph and similarity among the nodes have been measured based on different types of crimes. Each community is a group of states or union territories which are similar based on crime trends. Initially, the method finds the communities based on current year crime data, subsequently at the end of a year when new crime data for Special Issue Published in Int. Jnl. Of Advanced Networking & Applications (IJANA), the graph is modified and new communities are formed.

Chapter 3

PROBLEM STATEMENT

Problem statement System of Online Crime Management System:

In the problem statement system, the managements are done only manually but in proposed system we have to computerize the management using this application. In the existing system we face problems like:

- Lack of security of data.
- More man power.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials.

Chapter 4

PROPOSED MODEL

4.1.DESIGN

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement has been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

4.1.1 Data Flow Diagrams

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical

data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this led to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

4.1.1.1 DFD Symbols:

In the DFD, there are four symbols

- A square defines a source(originator) or destination of system data
- An arrow identifies data flow. It is the pipeline through which the information flows
- A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.

- An open rectangle is a data store, data at rest or a temporary repository of data

4.1.1.2 Constructing a DFD:

Several rules of thumb are used in drawing DFD'S:

Process should be named and numbered for an easy reference. Each name should be representative of the process.

The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.

When a process is exploded into lower level details, they are numbered.

The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized. A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out. Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

4.1.1.3 Salient Feature of DFD's

- The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
- The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
- The sequence of events is not brought out on the DFD.

4.1.1.4 Data Flow:

- A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated however by two separate arrows since these happen at different type.

- A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
- A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
- A Data flow to a data store means update (delete or change).
- A data Flow from a data store means retrieve or use. A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

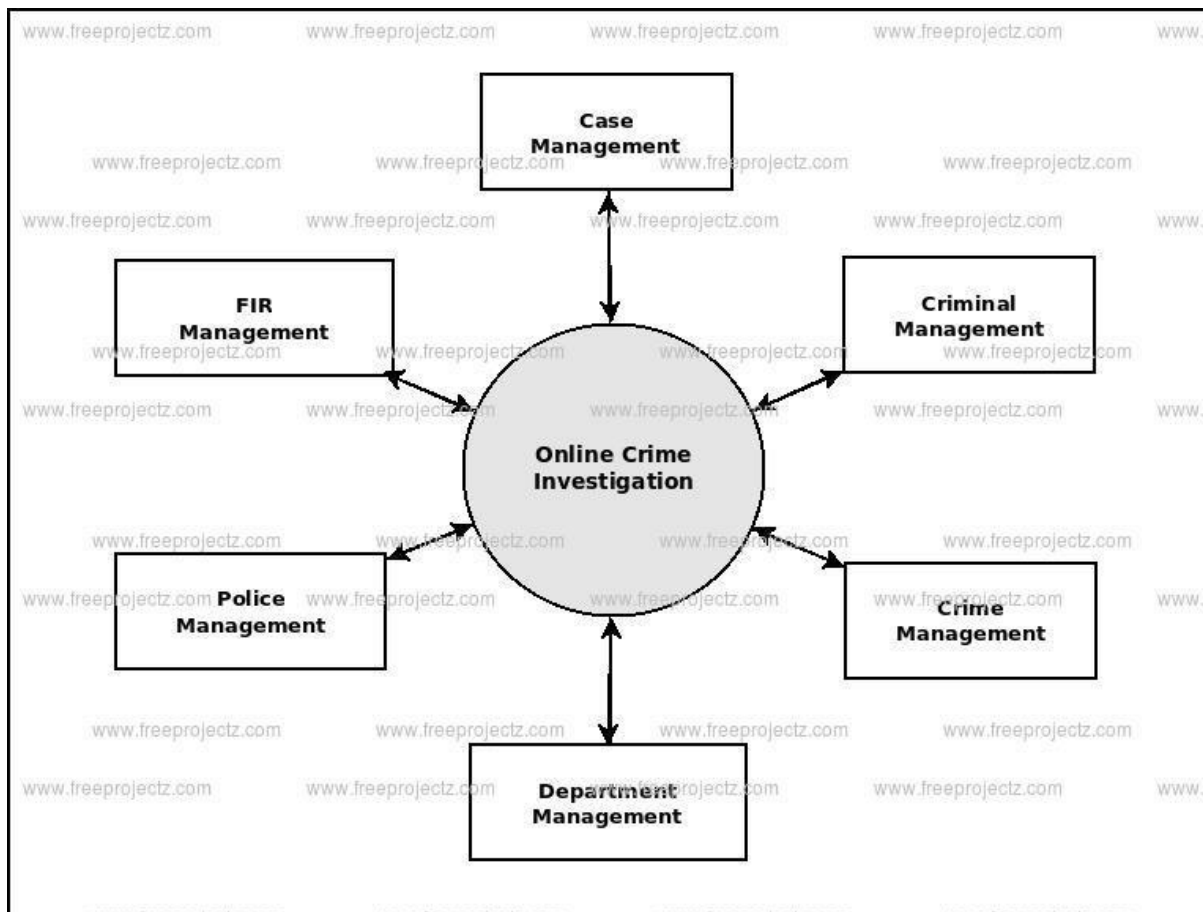


Fig 4.1.1.4.1 Zero-level DFD for Online Crime management.

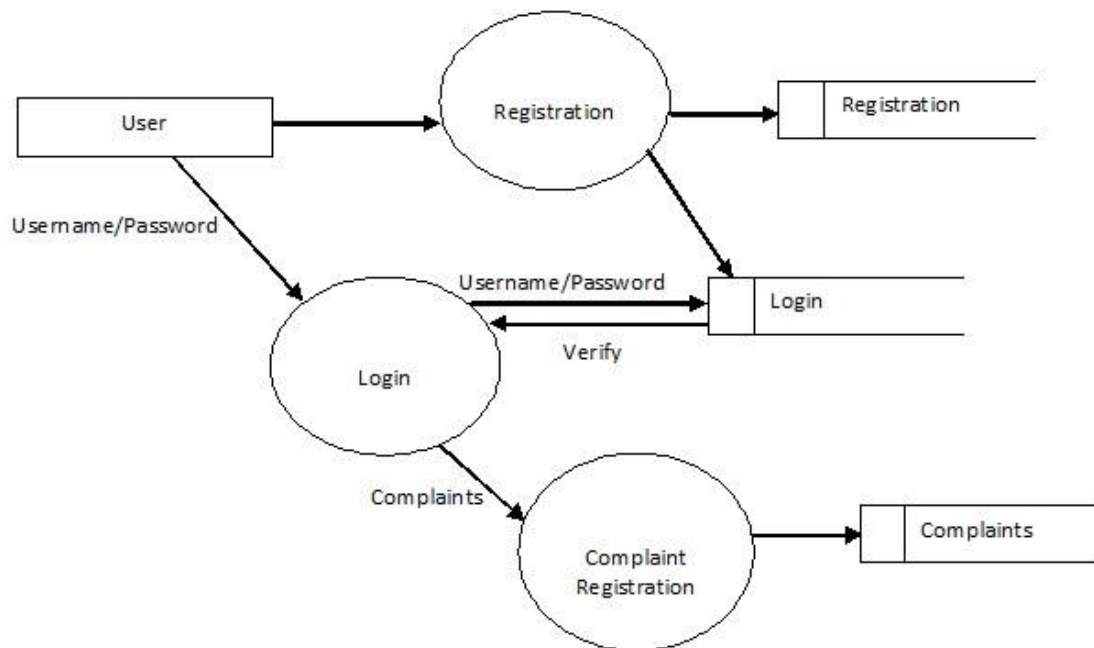


Fig 4.1.1.4.2 DFD for User/Complainer registration.

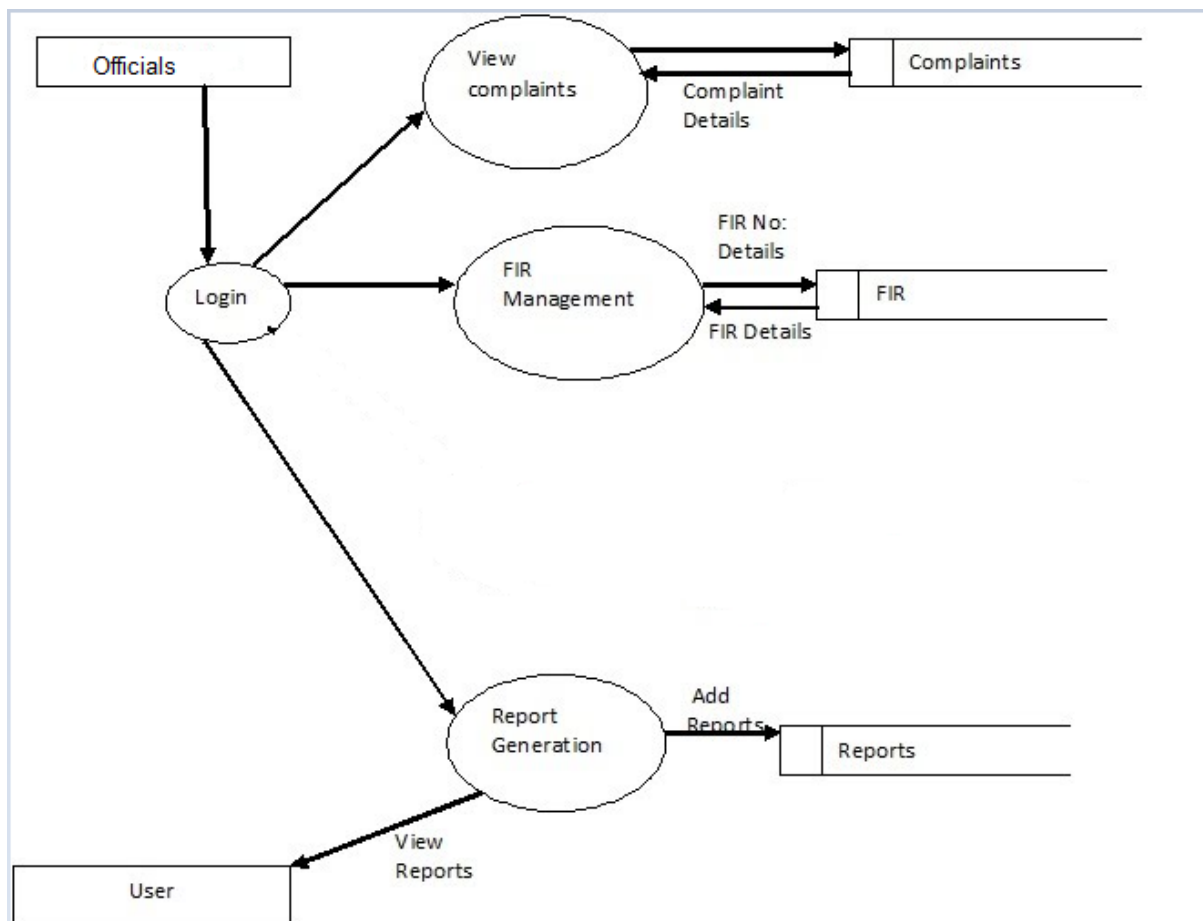


Fig 4.1.1.4.3 DFD for a Police Officer.

4.1.2 ER-Diagrams

The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the data modeling activity the attributes of each data object noted in the ERD can be described using a data object description.

- The set of primary components that are identified by the ERD are
- Data object
- Relationships
- Attributes
- Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.

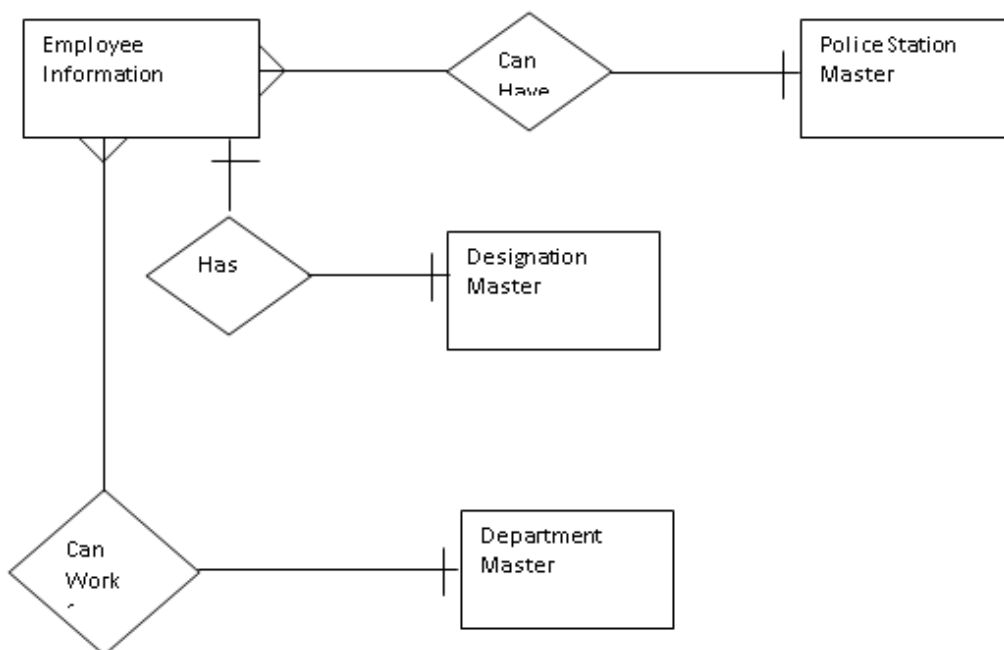


Fig 4.1.2.1 Employee Information.

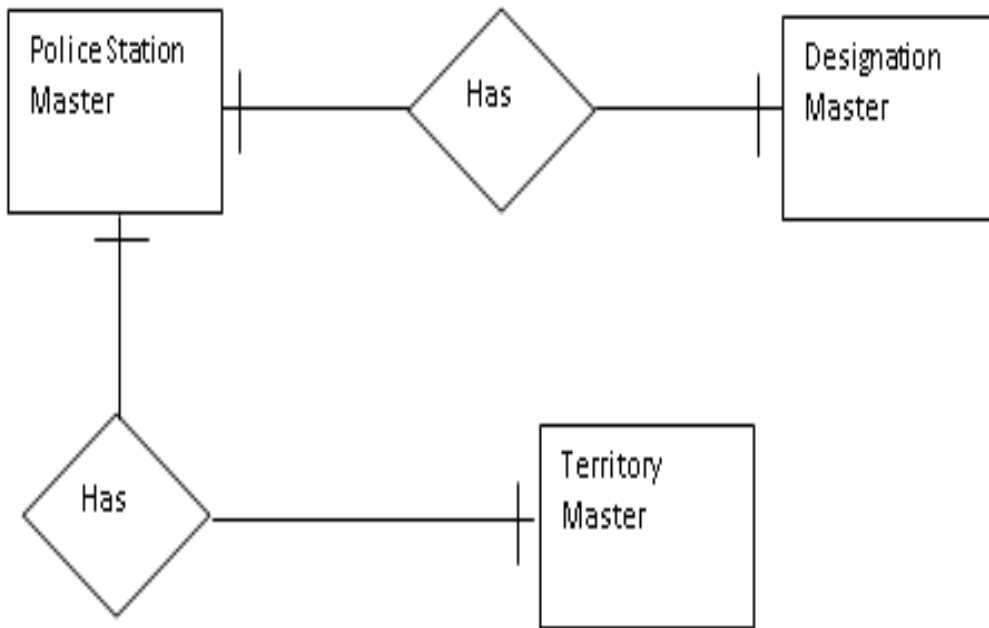


Fig 4.1.2.2 Police Station Master.

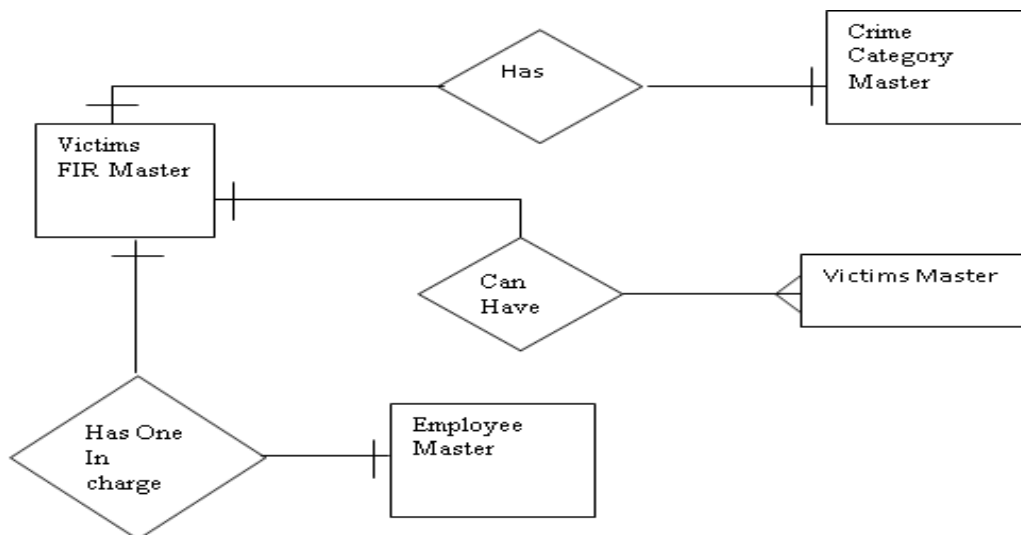


Fig 4.1.2.3 Victims FIR Master.

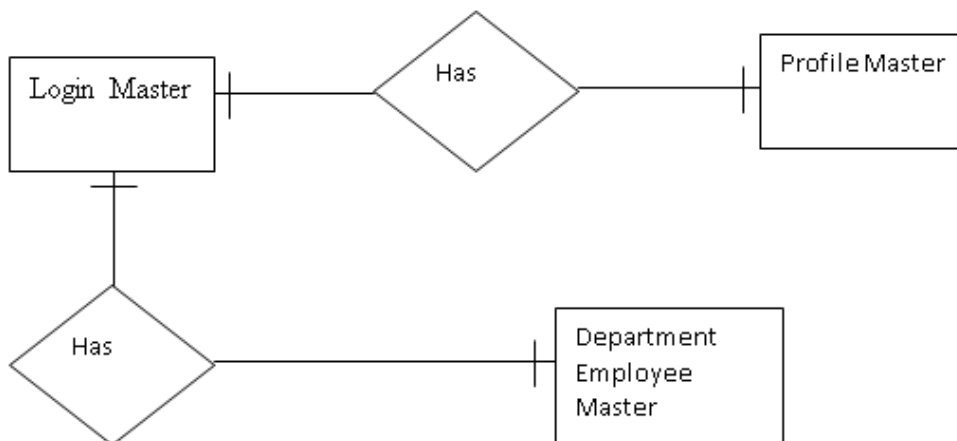


Fig 4.1.2.4 Login Master.

4.1.3 Unified Modeling Language Diagrams

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

User Model View:

- This view represents the system from the user's perspective.
- The analysis representation describes a usage scenario from the end-user's perspective.

Structural model view:

- In this model the data and functionality are arrived from inside the system.
- This model view models the static structures

Behavioral Model View:

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View:

In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental Model View:

In this the structural and behavioral aspect of the environment in which the system is to be implemented are represented. UML is specifically constructed through two different domains they are,

- UML Analysis modeling, which focuses on the user model and structural

model views of the system.

- UML design modeling, which focuses on the behavioral modeling implementation modeling and environmental model views

4.1.4 Use Case Diagrams

The actors who have been identified in the system are as follows:

- Administrator
- Complainer
- In-charge officer
- Police officer
- Head Quarter

Administrator: Administrator is the one who is responsible for management of information. This includes, managing user information, managing FIR, managing police information (police station, police officers), managing updates on FIR.

Complainer: Complainer is the person who logs FIR.

In-charge: This person is responsible for assigning the police officer to the cases, and also has the power to assign or remove the police officer from the case and from the database.

Police: This person is responsible for updating of case status and generating report.

Head Quarter: Head-Quarter is the supreme power who can see everything. It can see all the police stations that come under its area. And also see the cases under those police stations by searching.

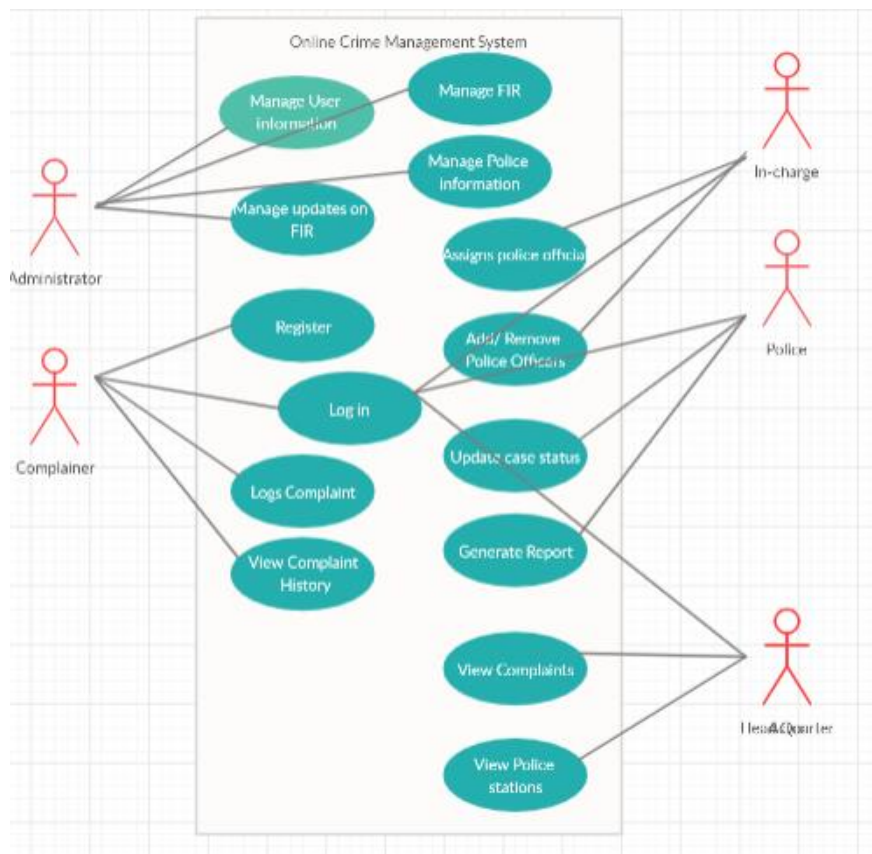


Fig 4.1.4.1 Use-case diagram.

4.1.4.1 Use case Description:

“Login” use-case:

Use case Login Information
name

Participating actors Complainer, In-charge, Police, Head Quarter

Flow of events Provides username and password

Entry Condition Users must know the username and password

Exit condition User successfully logged into the system

Quality Requirements Should provide proper error messages while login into the system.

“Complainer Register” use-case

Use case Register Victims

name

Participating Complainer

actors

Flow of User will enter his/her information

events

Entry User should know the his/her details

Condition

Exit Details are successfully inserted into the system.

condition

Quality Display proper error messages while insertion.

Requirements

“Log Fir” use-case

Use case Log FIR

name

Participating Complainer

actors

Flow of User will register the FIR

events

Entry User should know the details of the FIR

Condition

Exit FIR details are successfully inserted into the system.

condition

Quality Display proper error messages while insertion.

Requirements

“Assign police officials” use-case

Use case Assign police officials

name

Participating In charge

actors

Flow of User will search the case and assign the police official

events

Entry User should know the case number

Condition

Exit Assigned details should be successfully inserted into the
condition system.

Quality Display proper error messages while insertion.

Requirements

“Update case status” use case

Use case Update case status

name

Participating Police officer

actors

Flow of User will search the case number and update the case status

events

Entry User should know the details.

Condition

Exit Details are successfully inserted into the system.

condition

Quality Display proper error messages while insertion.

Requirements

“View Complaint and View police station” use-cases

<i>Use case name</i>	View Complaint and View police station
<i>Participating actors</i>	Head Quarter
<i>Flow of events</i>	User can view all the crime details, police stations under head quarter
<i>Entry Condition</i>	Display the details of crime, and police stations.
<i>Exit condition</i>	Details are successfully displayed.
<i>Quality Requirements</i>	N/A

4.1.5 Sequence Diagram:

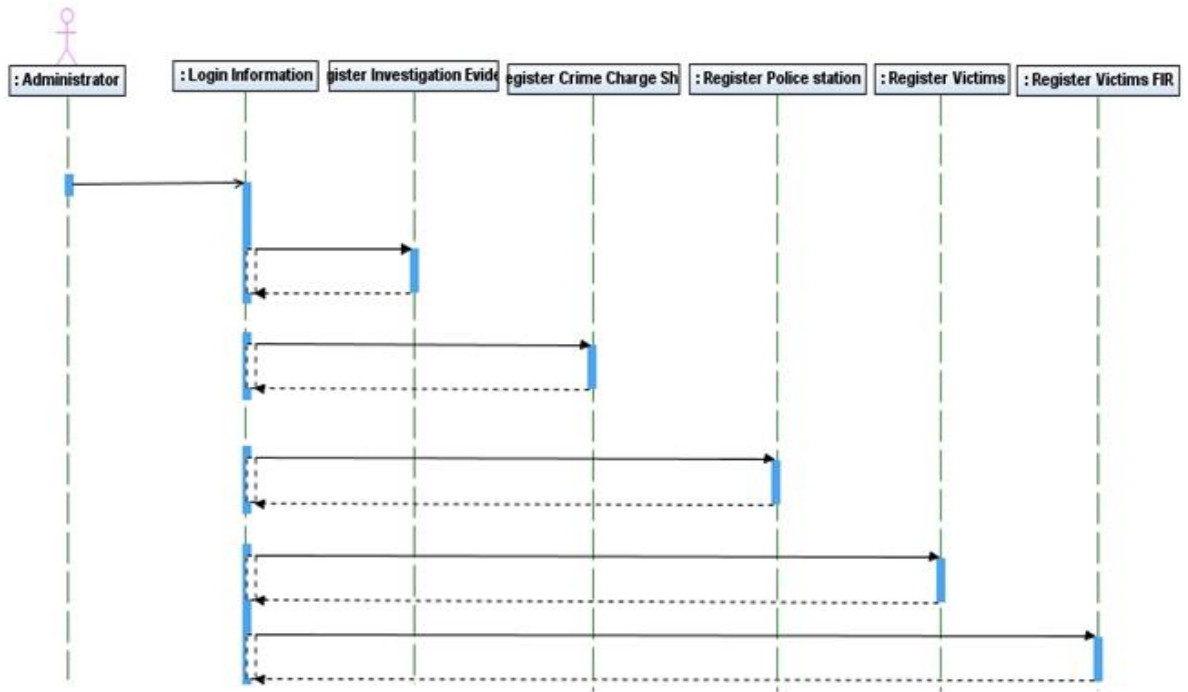


Fig 4.1.5.1 Administration Sequence diagram.

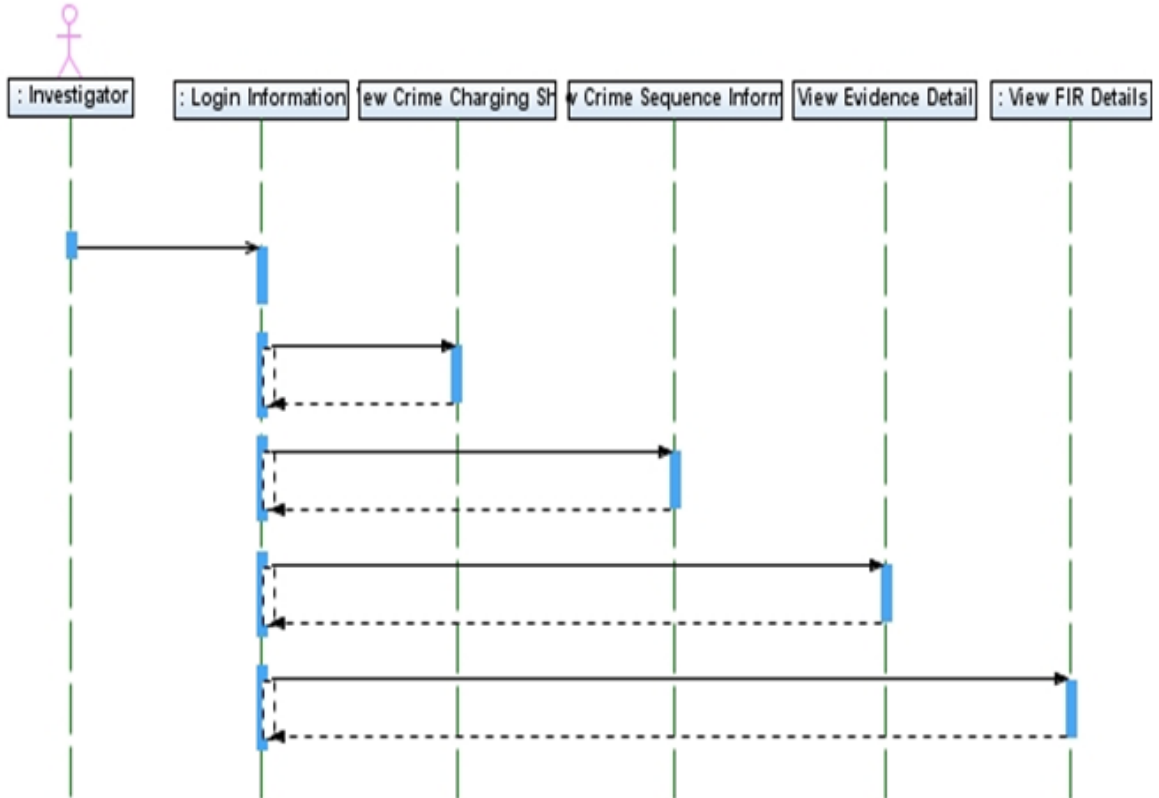


Fig 4.1.5.2 Investigator Sequence diagram.

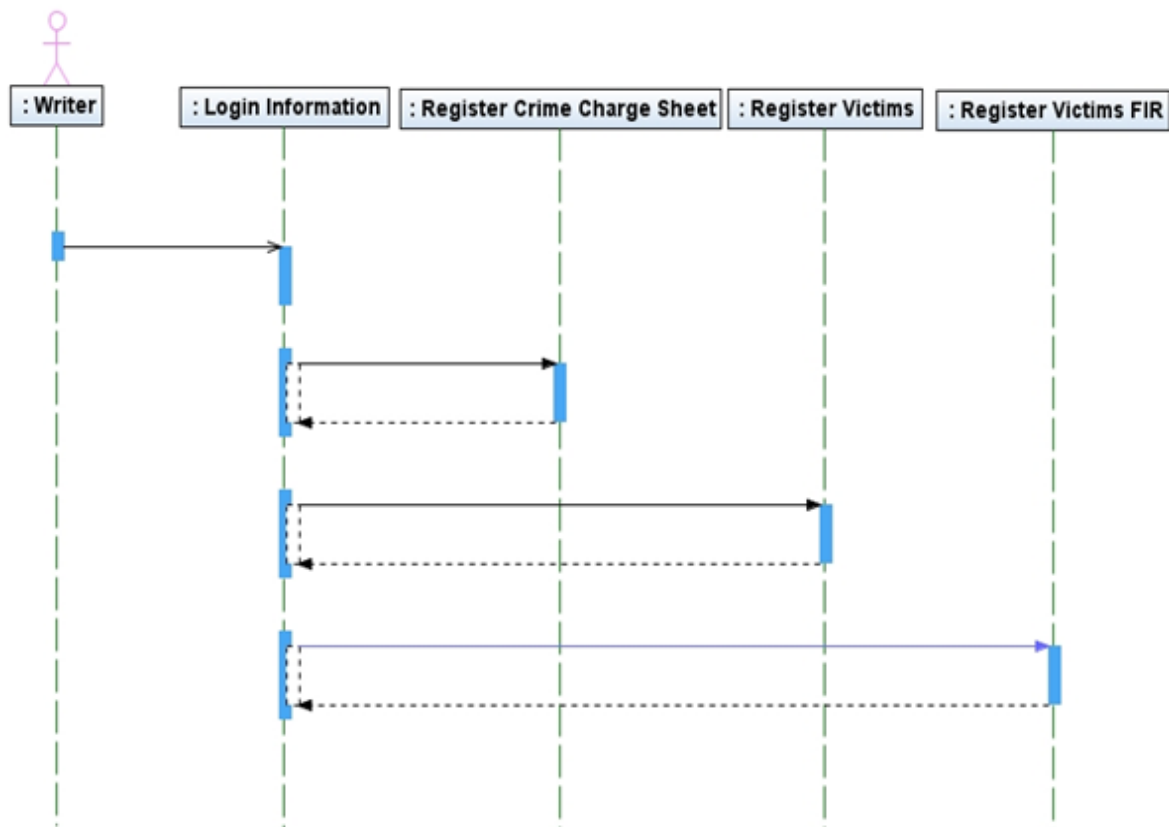


Fig 4.1.5.3 Writer Sequence diagram.

4.1.6 Testing Methodologies:

- **Black box Testing:** is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.
- **White box Testing:** is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually The Developers are involved in white box testing.
- **Gray Box Testing:** is the process in which the combination of black box and white box tonics are used.

4.1.6.1 Test Planning:

- Test Plan is defined as a strategic document which describes the procedure how to perform various testing on the total application in the most efficient way.
- This document involves the scope of testing,
- Objective of testing.
- Areas that need to be tested.
- Areas that should not be tested.
- Scheduling Resource Planning.

4.1.6.2 Types of Testing:

- Regression Testing: is one of the best and important testing. Regression testing is the process in which the functionality, which is already tested before, is once again tested whenever some new change is added in order to check whether the existing functionality remains same.
- Re-Testing: is the process in which testing is performed on some functionality which is already tested before to make sure that the defects are reproducible and to rule out the environment's issues if at all any defects are there.
- Static Testing: is the testing, which is performed on an application when it is not been executed: GUI, Document Testing
- Dynamic Testing: is the testing which is performed on an application when it is being executed: Functional testing.
- Alpha Testing: it is a type of user acceptance testing, which is conducted on an application when it is just before released to the customer.
- Beta-Testing: it is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.

- **Installation Testing:** it is the process of testing in which the tester try to install or try to deploy the module into the corresponding environment by following the guidelines produced in the deployment document and check whether the installation is successful or not.

4.1.7 Technologies used:

Front end: HTML, CSS, JavaScript

- **HTML:** HTML is used to create and save web document. E.g. Notepad/Notepad++
- **CSS:** (Cascading Style Sheets) Create attractive Layout
- **JavaScript:** it is a programming language, commonly use with web browsers.

Back end: PHP, MySQL

- **PHP:** Hypertext Pre-processor (PHP) is a technology that allows software developers to create dynamically generated web pages, in HTML, XML, or other document types, as per client request. PHP is open source software.
- **MySQL:** MySQL is a database, widely used for accessing querying, updating, and managing data in databases.

Server used:

XAMPP server: XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

Chapter 5

IMPLEMENTATION

5.1 Initial Design

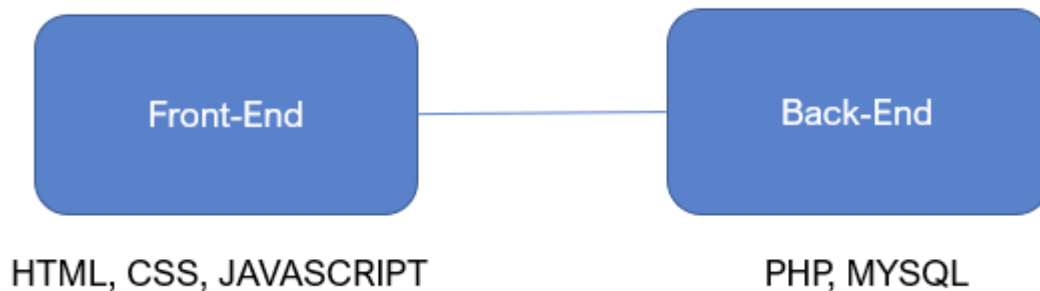


Fig 5.1 Initial Design.

Initial Design consisted of the two blocks namely: Front-end and Back-end

- Front-End comprised of the UI components through which the user interacts.
- Back-end comprises of the Database and the connector.

Front-end languages used are HTML, CSS, JavaScript. Using these languages, we form the GUI (Graphical User interface). While PHP is the language that enables communication between the front end and backend.

Step-1: Forming of Database

First task was to form the database that consisted of the tables where our data will be stored. “crime_portal” is the name of the database of our project.

Following tables were created:

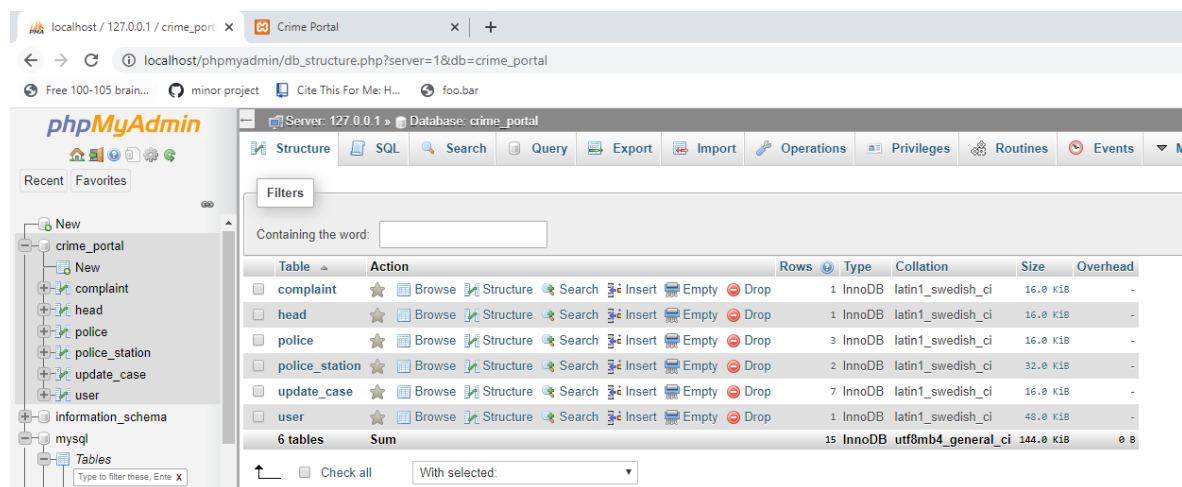


Fig 5.2 “crime_portal” Database in phpMyAdmin.

Server: 127.0.0.1 » Database: crime_portal » Table: complaint

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	c_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	a_no	bigint(12)			No	None			Change Drop More
3	location	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	type_crime	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
5	d_o_c	date			No	None			Change Drop More
6	description	varchar(7000)	latin1_swedish_ci		No	None			Change Drop More
7	inc_status	varchar(50)	latin1_swedish_ci		Yes	Unassigned			Change Drop More
8	pol_status	varchar(50)	latin1_swedish_ci		Yes	null			Change Drop More
9	p_id	varchar(50)	latin1_swedish_ci		Yes	Null			Change Drop More

Fig.5.3 Structure of “complaint table”.

Server: 127.0.0.1 » Database: crime_portal » Table: head

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	h_id	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
2	h_pass	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext
Remove from central columns

Fig.5.4 Structure of “head” table.

Server: 127.0.0.1 » Database: crime_portal » Table: police

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	p_name	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
2	p_id	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
3	spec	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	location	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
5	p_pass	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Fig.5.5 Structure of “police” table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	i_id	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
2	i_name	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
3	location	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	i_pass	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Fig.5.6 Structure of “police_station” table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	c_id	int(11)			No	None			Change Drop More
2	d_o_u	timestamp			No	current_timestamp()			Change Drop More
3	case_update	varchar(200)	latin1_swedish_ci		No	None			Change Drop More

Fig.5.7 Structure of “update_case” table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	u_name	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
2	u_id	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
3	u_pass	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	u_addr	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
5	a_no	bigint(12)			No	None			Change Drop More
6	gen	varchar(15)	latin1_swedish_ci		No	None			Change Drop More
7	mob	bigint(10)			No	None			Change Drop More

Fig.5.8 Structure of “user” table.

Step-2: Implementing front-end and forming connection with the back-end

Front-end consisted of implementing various UI pages through which the client would interact. This comprises of the pages as described below,

- “home.php”: This would be the first page where the client would land. From here, on the officer would go to “official_login.php”. Client who is new will go to “registration.php”. If the complainer wants to log a

complaint, then he/she will go to “userlogin.php” page. Below, is the code snippet of home.php

```

home.php - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
<head>

    <title>Crime Portal</title>
    <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
    <link href="http://fonts.googleapis.com/css?family=Lato:300,400,700,300italic,400italic,700italic" rel="stylesheet" type="text/
    <link rel="stylesheet" type="text/css" href="home.css">
</head>
<body>
    <div class="content">
        <div class="row">
            <div class="col-md-12">
                <div class="text-align: center;">
                    <h1>Crime Portal</h1>
                </div>
            </div>
        </div>
    </div>
</body>
</html>

```

Fig5.9 Code snippet of home page.

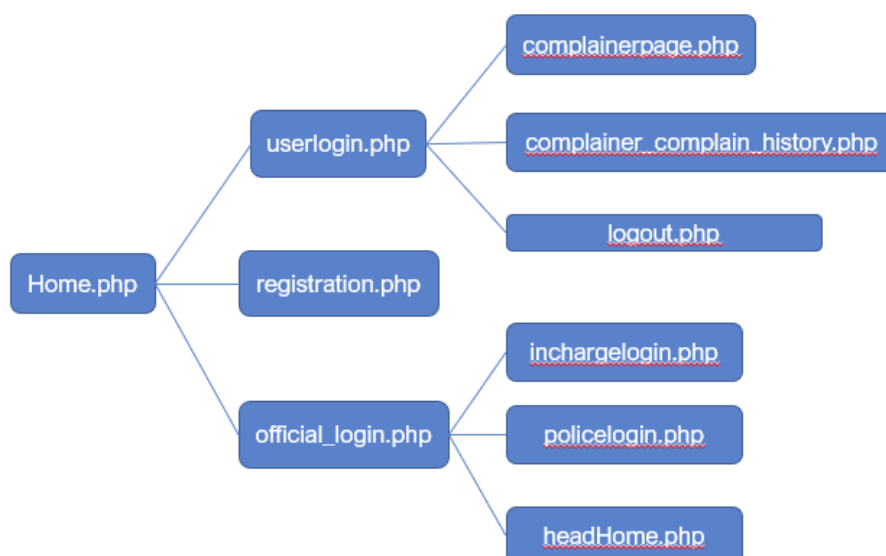


Fig.5.10 User-flow.

The lines in above means user can move to-and-fro on these pages.

- “registration.php”: In the home page when the user clicks on “Sign Up” button. The user is directed to this page. This page is for registration for a user who is new to the portal and wants to log the complaint. On this page the user must provide the following credentials for successful registration. The details provided here are stored in the backend database table named “user”. Below, is the code snippet of the “registration.php”.

```
        echo "<script type='text/javascript'>alert('$message');</script>";
    }
}
?>

<script>
function f1()
{
    var sta=document.getElementById("name1").value;
    var sta1=document.getElementById("email1").value;
    var sta2=document.getElementById("pass").value;
    var sta3=document.getElementById("addr").value;
    var sta4=document.getElementById("aadh").value;
    var sta5=document.getElementById("mobno").value;

    var x=sta.trim();
    var x1=sta1.indexOf(' ');
    var x2=sta2.indexOf(' ');
    var x3=sta3.trim();
    var x4=sta4.indexOf(' ');
    var x5=sta5.indexOf(' ');
    if(sta!="" && x!=""){
        document.getElementById("name1").value="";
        document.getElementById("name1").focus();
        alert("Space Not Allowed");
    }

    else if(sta1!="" && x1>=0){
        document.getElementById("email1").value="";
        document.getElementById("email1").focus();
        alert("Space Not Allowed");
    }
    else if(sta2!="" && x2>=0){
        document.getElementById("pass").value="";

```

Fig5.9 Code snippet of home page.

In the above image, the function1() is the part of the code is responsible for storing the data of registrar.

- “Userlogin.php”: This page is used when the complainer wants to log a new complain, view the status of an old complain. The user logs in with the credentials that he/she has provided at the time of registration. If the user credentials are correct the user is taken to the “complainer_page.php” where the user is given option to log a new complaint or view the history

of complaint and Logout. Below, is the code of “complainer_page.php” that implements log a new complaint

```
if($_SERVER["REQUEST_METHOD"]=="POST")
{

    $location=$_POST['location'];
    $type_crime=$_POST['type_crime'];
    $d_o_c=$_POST['d_o_c'];
    $description=$_POST['description'];

    $var=strtotime(date("Ymd"))-strtotime($d_o_c);

if($var>=0)
{

    $comp="INSERT into complaint(a_no,location,type_crime,d_o_c,description) values('$a_no','$location','$type_crime','$d_o_c','$description')";
    mysqli_select_db($conn,"crime_portal");
    $res=mysqli_query($conn,$comp);

    if(!$res)
    {
        $message1 = "Complaint already filed";
        echo "<script type='text/javascript'>alert('$message1');</script>";
    }
    else
    {
        $message = "Complaint Registered Successfully";
        echo "<script type='text/javascript'>alert('$message');</script>";
    }
}
}
```

Fig5.9 Code snippet that implements complaint filing part.

In the above image, we can see that there is an INSERT query that inserts the complaint data in to the backend database table “complaint”.

- “complainer_complaint_history”: This page is use to check the history of the complaints logged. This page fetches the complaint data that the complainer might have logged in the past. The table from which the client fetches the data is “complaint” table.
- “Officer_login.php”: On the home page, in the navigation bar, there is a button “Officer login”. This takes the user to “officer_login.php”. This page is the login page for police officials. We have divided the police into three categories deciding on which the police officer is allowed to login through one of the three portals. These are namely,
 - Police login
 - In-charge Login
 - Headquarter Login

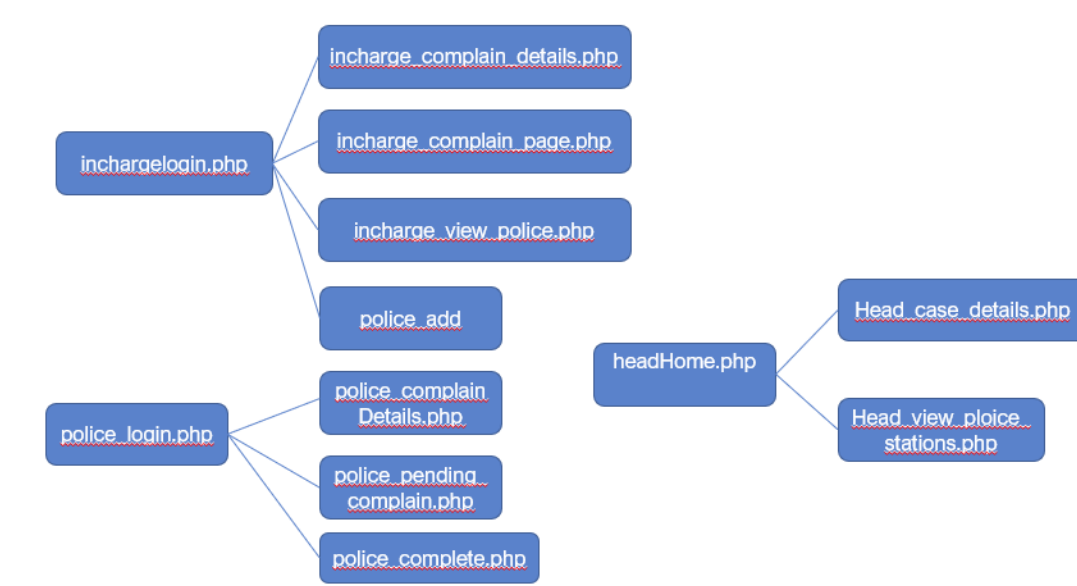


Fig.5.11. Official login page user flow.

- “Incharge_login.php”: When the complainer files a complaint, for example in city “A”, then the in-charge of that city will login through this portal to assign the police pertaining to that case. The in-charge is responsible for assigning police officer to the case. Further, he/she has the power to add the police and remove the police from the case. Based on the selection the in-charge can move to different pages.

Above is the user-flow of “incharge_login.php”. Below is the code snippet of the “incharge_login.php”.

```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="bootstrap.css">
  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
  <link href="http://fonts.googleapis.com/css?family=Lato:300,400,700,300italic,400italic,700italic" rel="stylesheet" type="text/css">
  <title>Incharge Login</title>
</head>
<?php
if(isset($_POST['s']))
{
  session_start();
  $_SESSION['x']=1;
  $conn=mysqli_connect("localhost","root","","crime_portal");
  if(!$conn)
  {
    die("could not connect".mysqli_error());
  }
  mysqli_select_db("crime_portal",$conn);

  if($_SERVER["REQUEST_METHOD"]=="POST")
  {
    $name=$_POST['email'];
    $pass=$_POST['password'];
    $result=mysqli_query($conn,"SELECT i_id,i_pass FROM police_station where i_id='$name' and i_pass='$pass' ");

    $_SESSION['email']=$name;
    if(!$result || mysqli_num_rows($result)==0)
    {

```

Fig5.9 Code snippet of “incharge_login.php”.

- “Police_login.php”: When the incharge assigns the police officer to the case. The police officer can view in his account that a case has been assigned to him. Now the police officer can update the case from time to time. These updates are saved in the back-end in the “update_case” table. So, the complainer who had initially logged the complaint, can see the case updates in the complaint history. In other words, the police officer updates the case, this is stored in the “update_case” table. This table is also linked to the complainer complaint history, so the complainer can also see the case status. Below is the code snippet of the “police.php”.

```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="bootstrap.css">
  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
  <link href="http://fonts.googleapis.com/css?family=Lato:300,400,700,300italic,400italic,700italic" rel="stylesheet" type="text/css">
  <title>Police Login</title>
</head>
<?php
if(isset($_POST['s']))
{
  session_start();
  $_SESSION['x']=1;
  $conn=mysqli_connect("localhost","root","","crime_portal");
  if(!$conn)
  {
    die("could not connect".mysqli_error());
  }
  mysqli_select_db("crime_portal",$conn);

  if($_SERVER["REQUEST_METHOD"]=="POST")
  {
    $name=$_POST['email'];
    $pass=$_POST['password'];
    $result=mysqli_query($conn,"SELECT p_id,p_pass FROM police where p_id='$name' and p_pass='$pass' ");
    $_SESSION['pol']=$name;
    if(!$result || mysqli_num_rows($result)==0)
    {
      $message = "Id or Password not Matched.";
      echo "<script type='text/javascript'>alert('$message');</script>";
    }
  }
}

```

Fig5.9 Code snippet of “police_login.php”.

- “Head_home.php”: This is the headquarter login page. From this page the headquarter officials can login and see all the details pertaining to a case, what is the status of the case and How many stations come under the headquarter. Based on the details, the officials can keep a track of any case. Below is the code snippet of “head_home.php”.

```

<!DOCTYPE html>
<html>
<head>

<?php
session_start();
if(!isset($_SESSION['x']))
    header("location:headlogin.php");

$conn=mysqli_connect("localhost","root","","crime_portal");
if(!$conn)
{
    die("could not connect".mysqli_error());
}
mysqli_select_db($conn,"crime_portal");

if(isset($_POST['s1']))
{
if($_SERVER["REQUEST_METHOD"]=="POST")
{
    $cid=$_POST['cid'];
    $_SESSION['cid']=$cid;
    header("location:head_case_details.php");
}
}

if(isset($_POST['s2']))
{
if($_SERVER["REQUEST_METHOD"]=="POST")
{
    $loc=$_POST['loc'];
    $_SESSION['loc']=$loc;
    header("location:headHome1.php");
}
}
}

```

Fig5.9 Code snippet of “head_login.php”.

Chapter 6

RESULTS

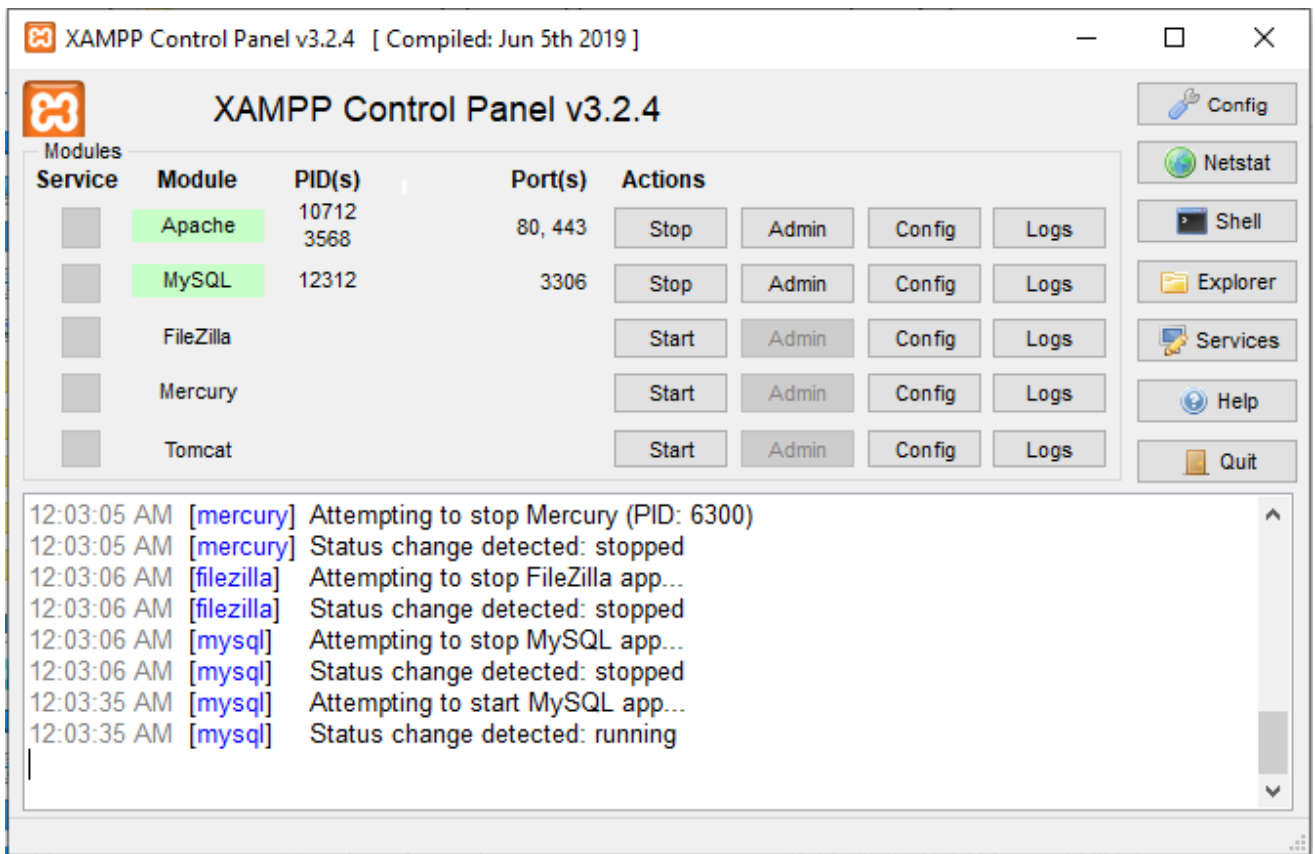


Fig 6.1 Xampp Control Panel Showing Apache and MySQL server running.

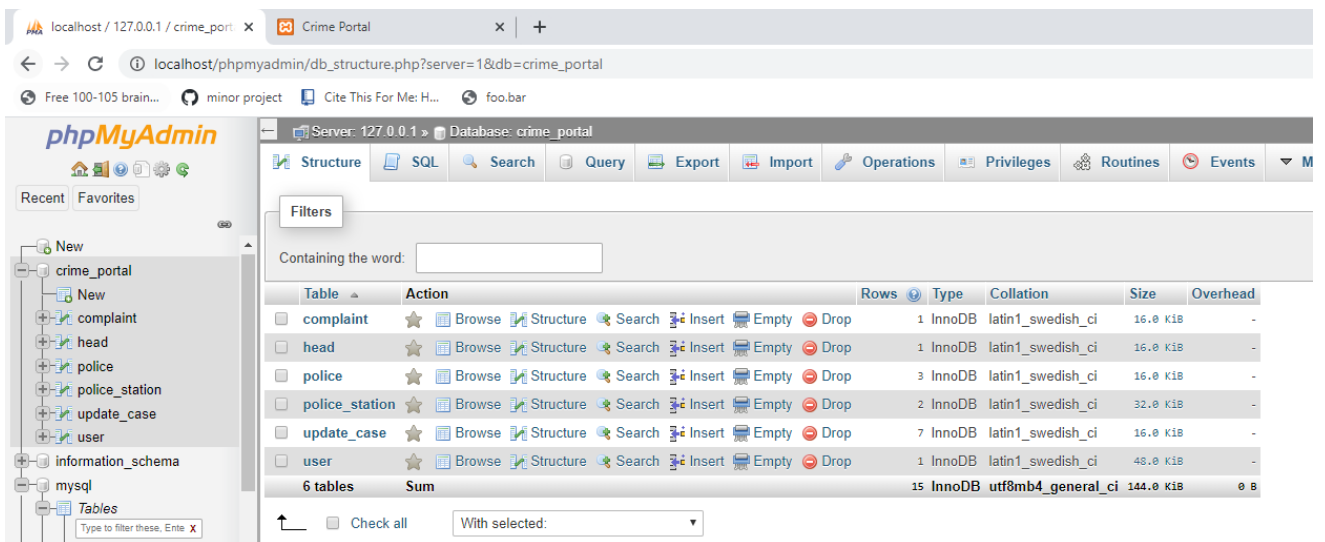


Fig 6.2 “crime_portal” Database in phpMyAdmin.

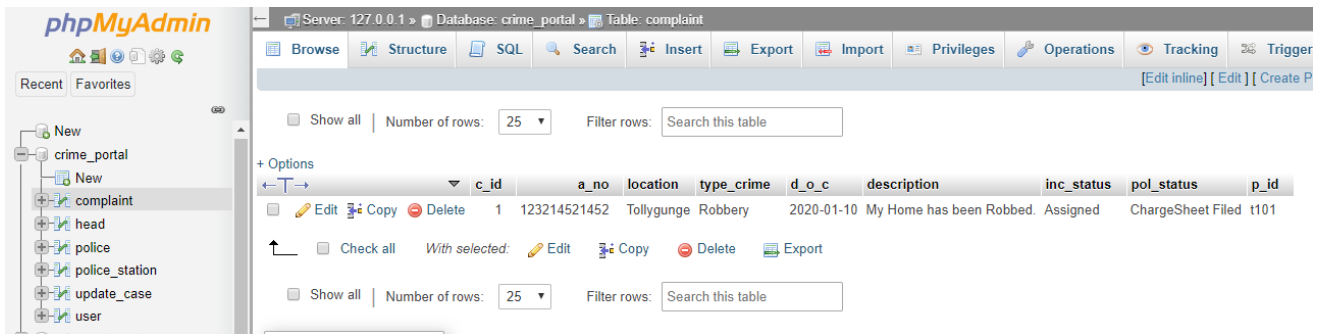


Fig 6.3 “complaint” table pre-filled with dummy values.

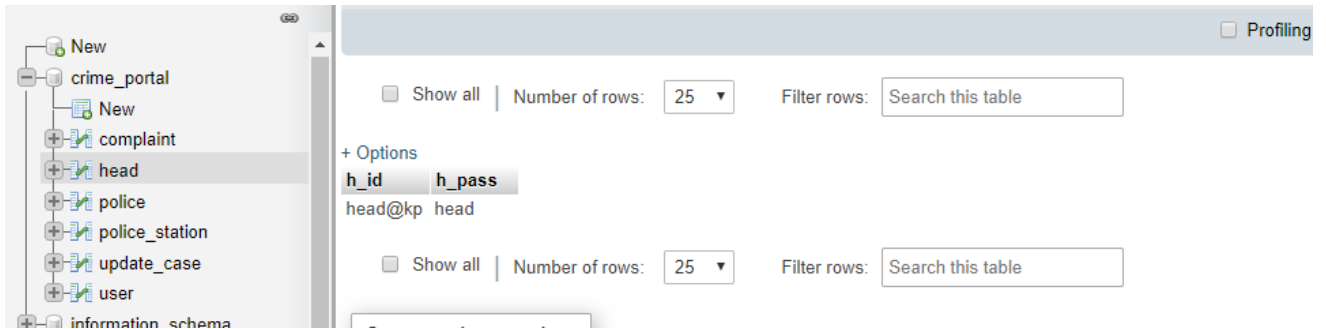


Fig 6.4 “head” table pre-filled with dummy values.

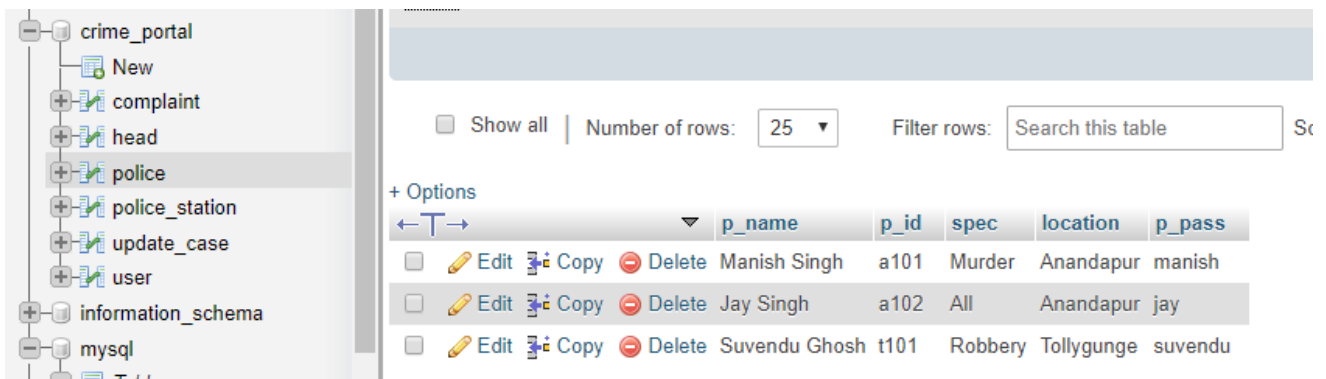


Fig 6.5 “police” table pre-filled with dummy values.

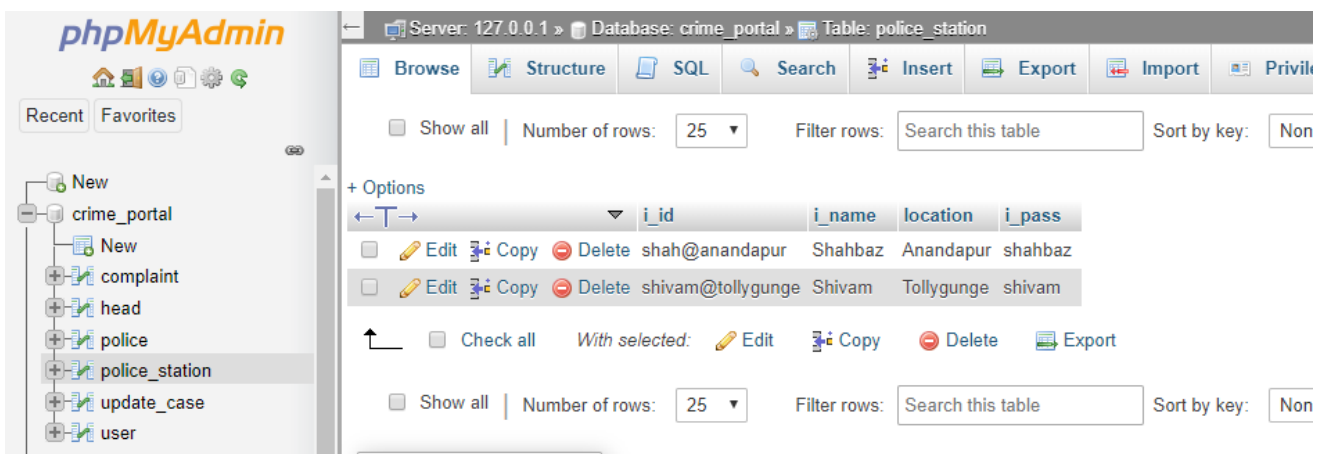


Fig 6.6 “police_station” table with dummy values.

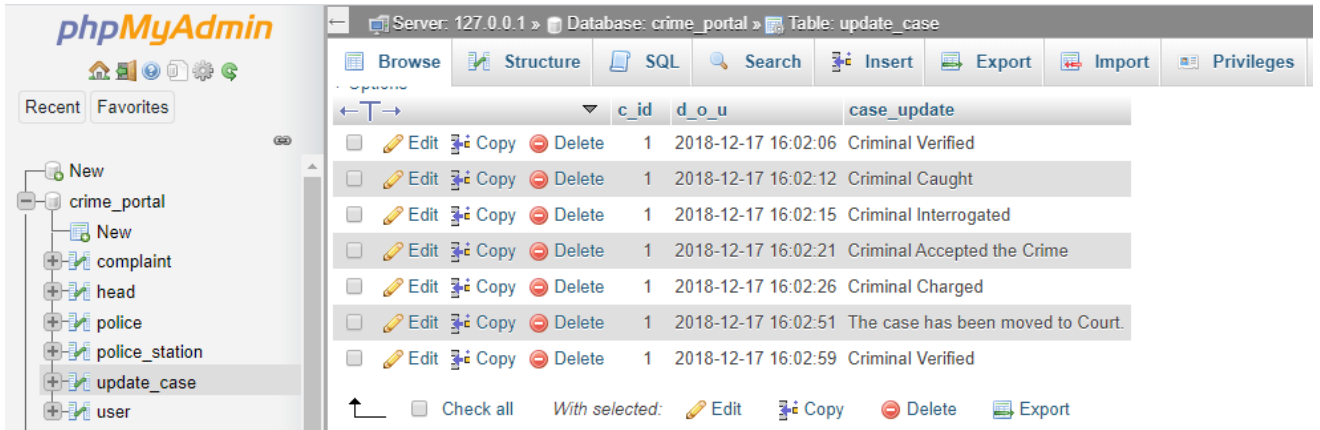


Fig 6.7“update_case” table with dummy values.

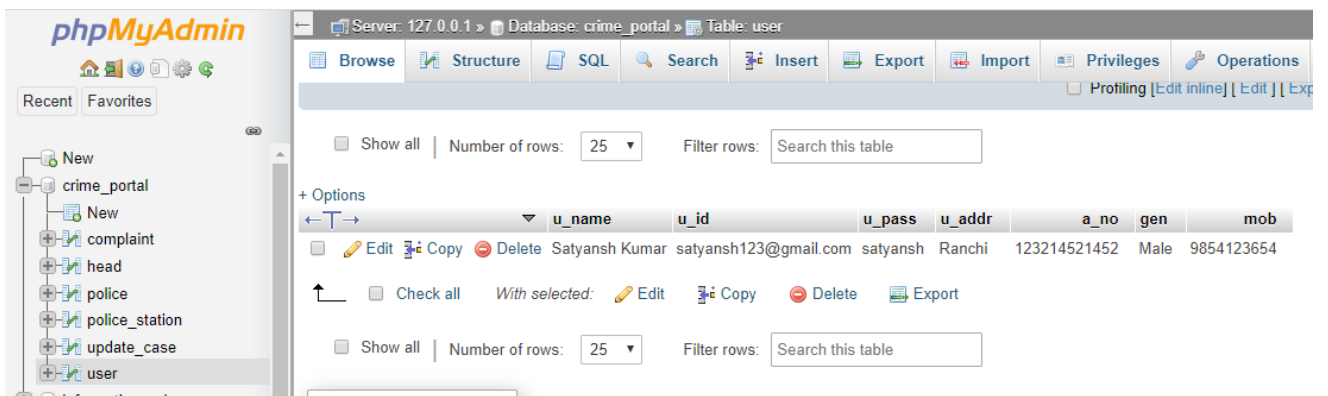


Fig 6.8 “user” table with dummy values.

6.1 Step-by-Step view of the website

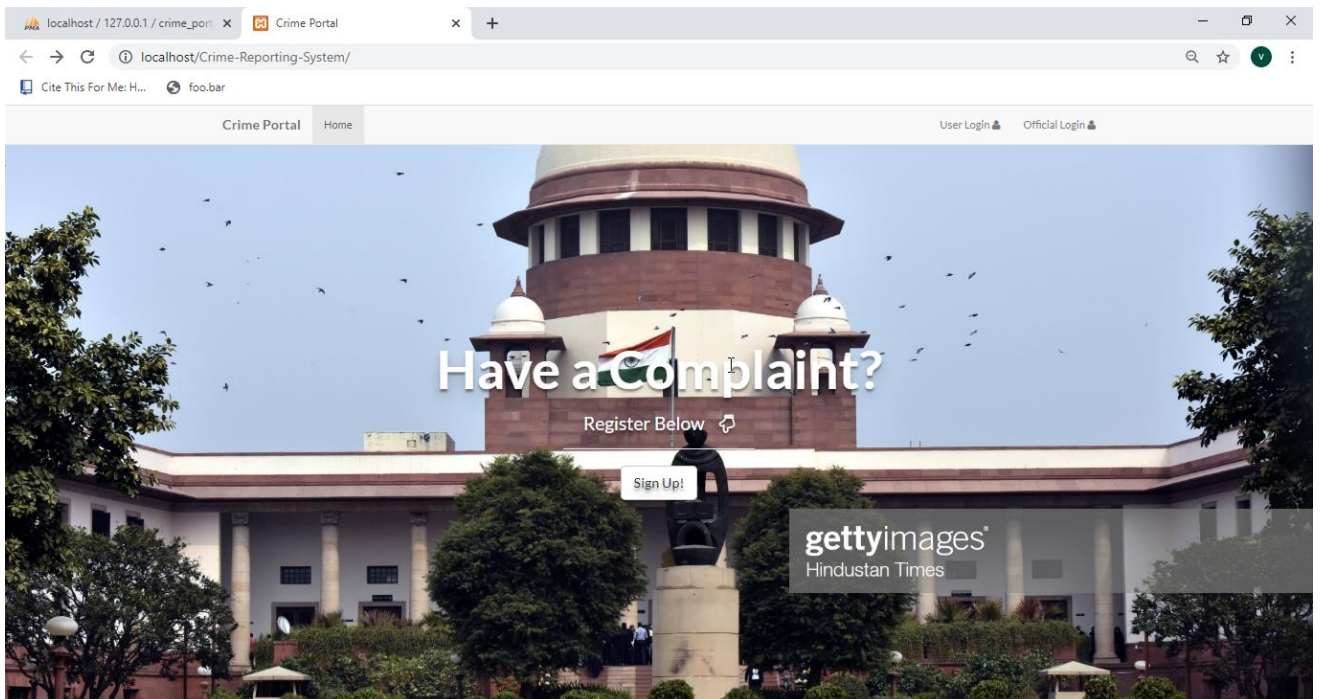
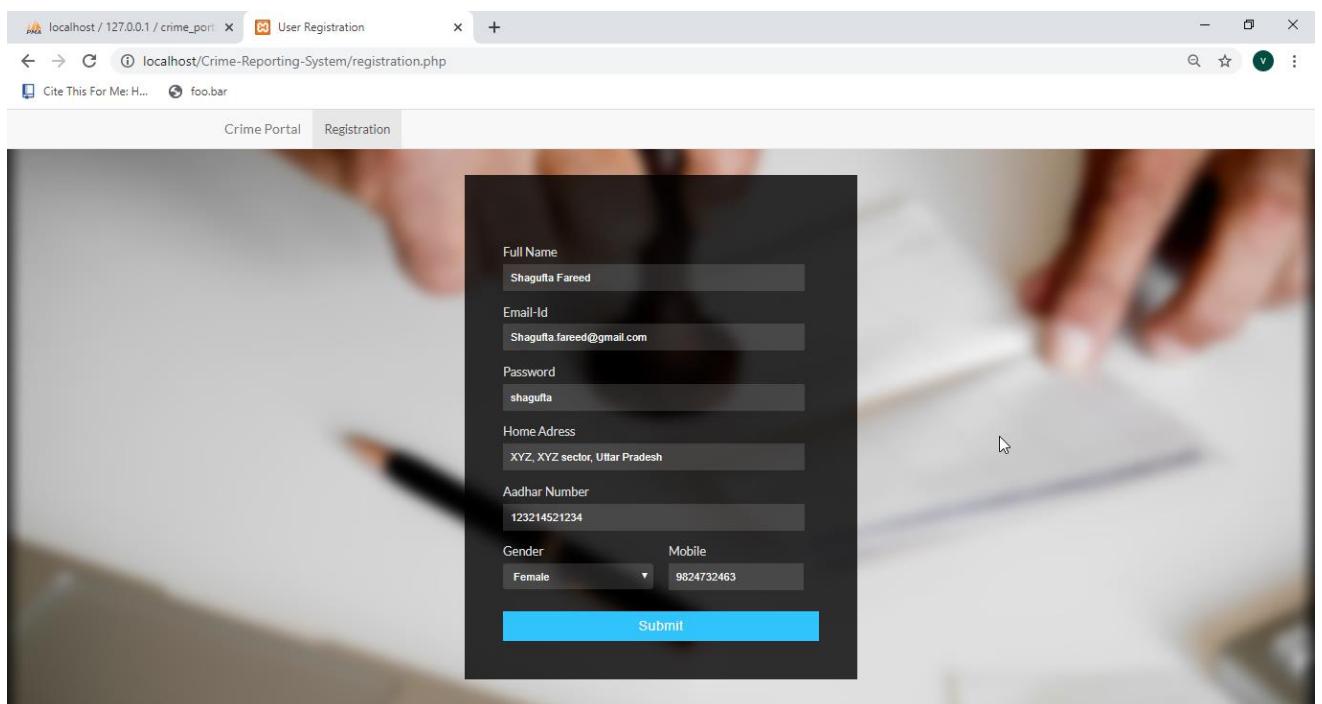


Fig 6.1.1 Front page of the “Crime Portal” website.

This is the page which will be common for all the viewers. From here on there multi-options for users to login depending on their type, below is their specification:

- If the person is a complainer, who wants to report a complaint, then complainant will have two options:
 - If he/she has never lodged a complaint before then he/she should “Sign Up”.
 - If the complainer has lodged a complaint earlier and want to either get an update on it then he/she can login to his/her account through “User Login” button at the navigation bar.
- If the person is a police official then he/she has to login through the “Official Login” button at the navigation bar.



The screenshot shows a web browser window with two tabs: 'localhost / 127.0.0.1 / crime_por...' and 'User Registration'. The address bar shows 'localhost/Crime-Reporting-System/registration.php'. The page content includes a navigation bar with 'Crime Portal' and 'Registration' tabs. The main content is a registration form with the following fields and values:

Full Name	Shagufa Fareed
Email-Id	Shagufa.fareed@gmail.com
Password	shagufa
Home Address	XYZ, XYZ sector, Uttar Pradesh
Aadhar Number	123214521234
Gender	Female
Mobile	9824732463

A blue 'Submit' button is located at the bottom of the form.

Fig 6.1.2 “Sign Up” page filled with dummy values for demonstration purpose.

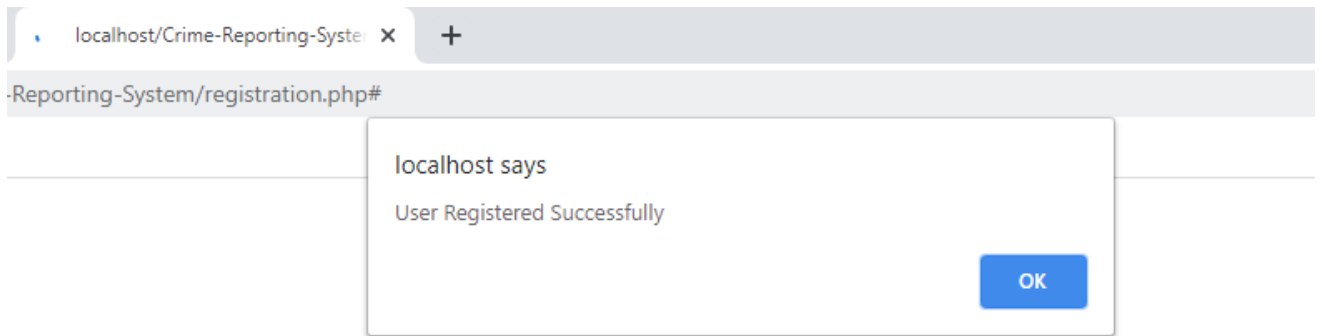


Fig 6.1.3 Successful Registration.

This should be reflected in the database. So below, is the database affirmation.

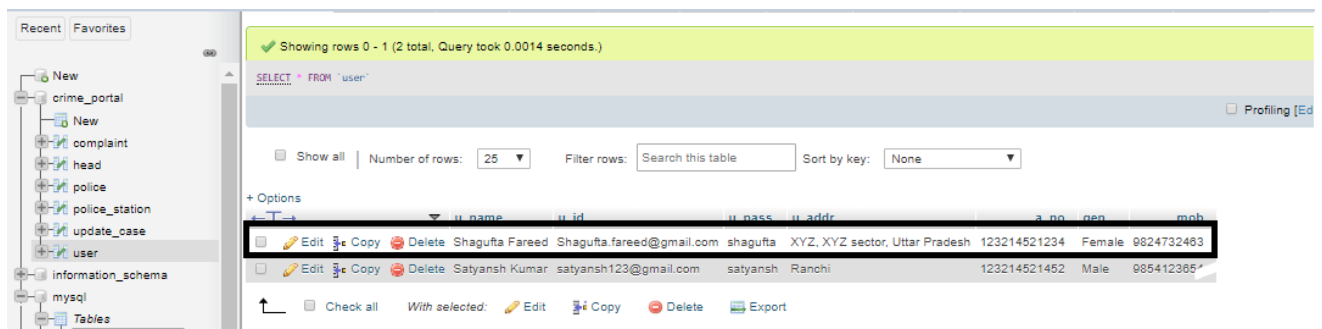


Fig 6.1.4 Successful addition of user represented by black rectangular box.

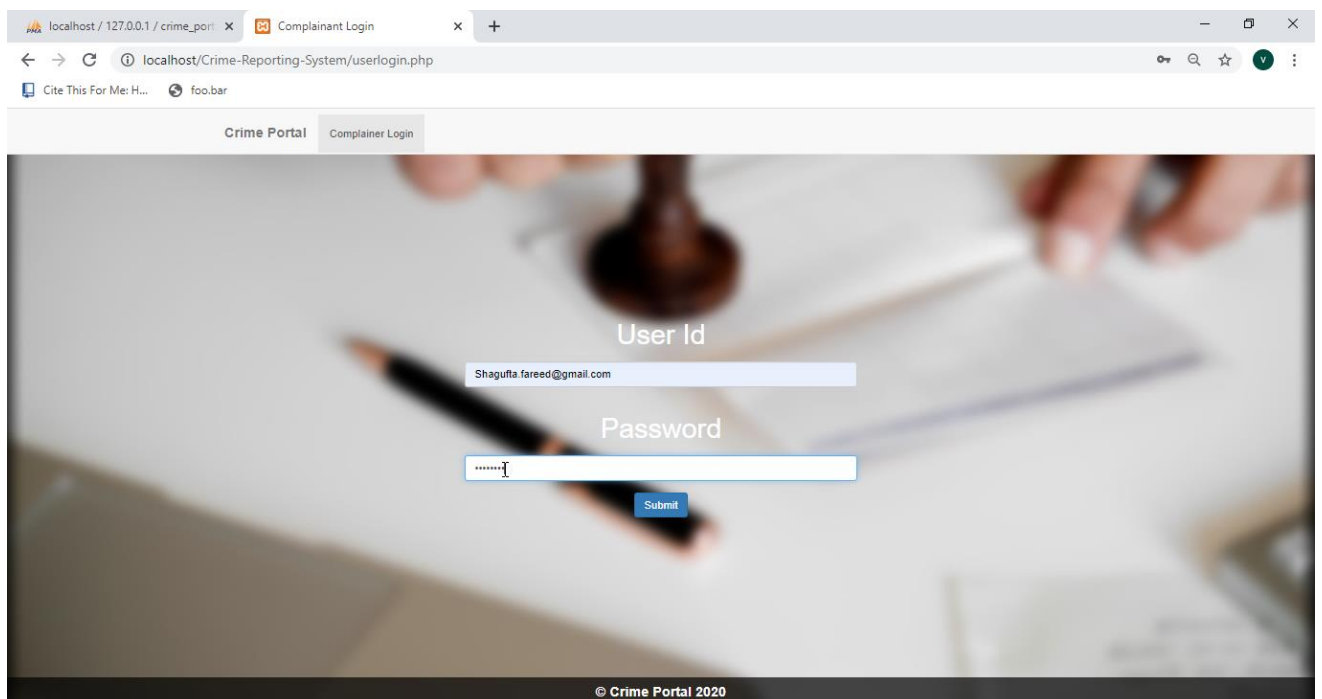


Fig 6.1.5 “User Login” page.

This is the “User Login” page. Here the user will login with credentials that he/she has filled during registration (“Sign Up”) process.

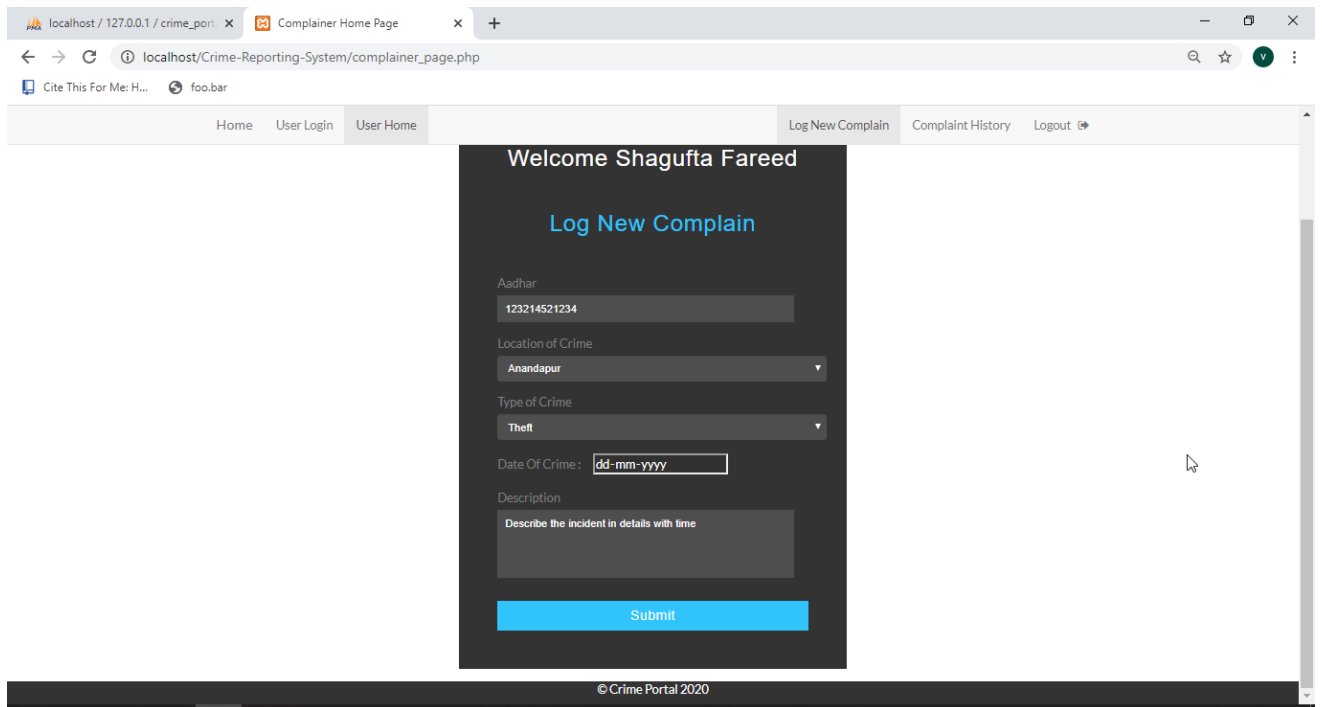


Fig 6.1.6 Successful Login.

Once the complainer has successfully registered himself/herself and logs in with the credentials from the “User Login” page. Then, the complainer lands up at the “complainer_page”. Here, the complainer can do following activities:

- Log new Complaint
- Check the History of complaint from the “Complaint History” button at the “navigation bar”.
- After everything is done the complainer can Logout by clicking on “Logout” Button.

Aadhar
123214521234

Location of Crime
Anandapur

Type of Crime
Robbery

Date Of Crime : 05-03-2020

Description
The incident took place yesterday where the jewellery and cash was taken

Submit

© Crime Portal 2020

Fig 6.1.7 Logging complaint.

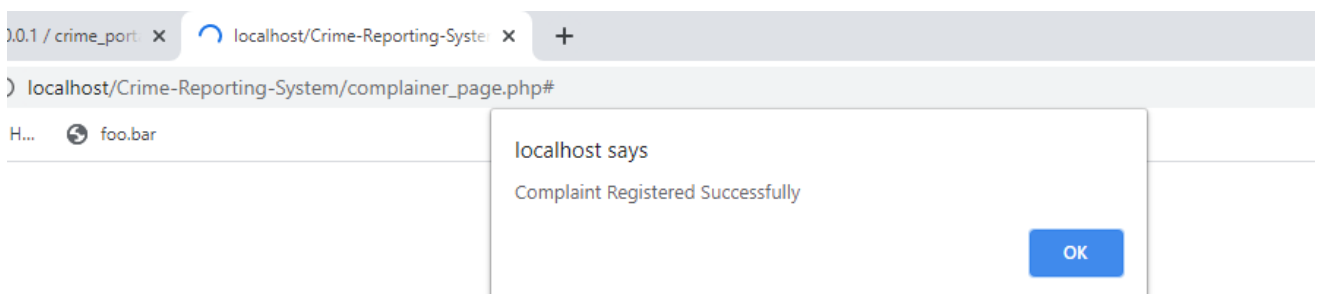


Fig 6.1.8 Complaint Logged Successfully.

This should be reflected in the database.

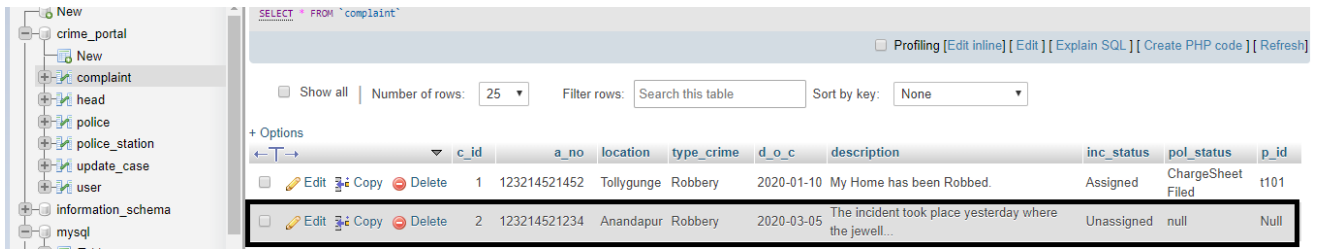


Fig 6.1.9 Successful addition of complaint represented by black rectangular box.

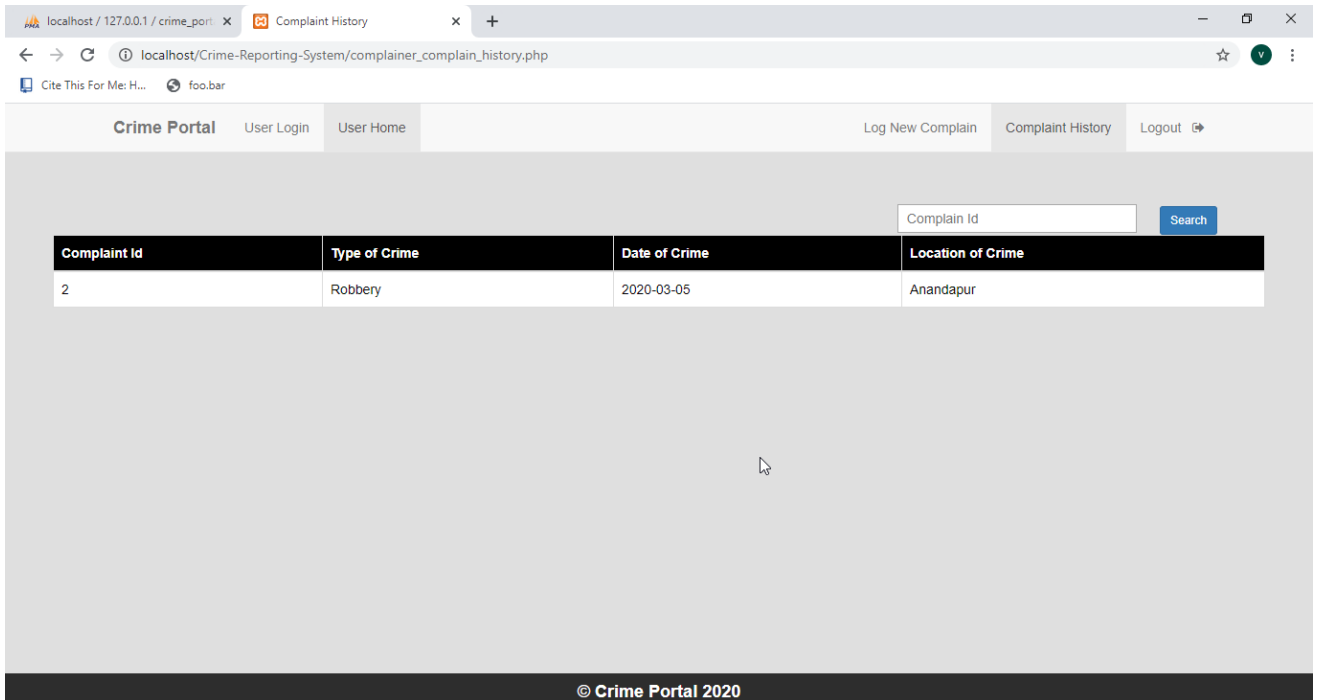


Fig 6.1.10 Complaint History Page.

Here, the complainer can see his/her recent logged complaints, and history of any previously filed complaints.

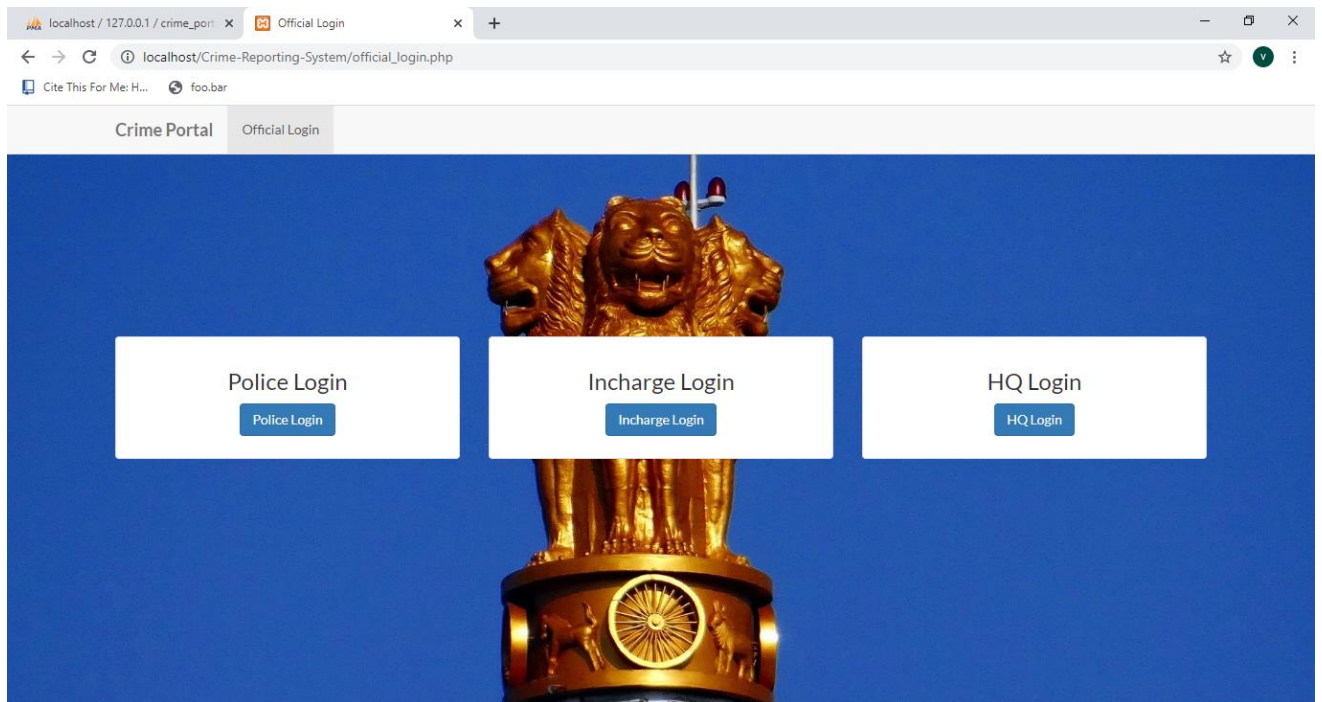


Fig 6.1.11 “Official Login” page.

Here, as we can see that there are three ways to login in the “Official Login” page.

Below is the description of the each:

- **Police Login:** This is login portal for police officials. Here the police officials can see the cases they have assigned to and can update the cases. Only the police officials declared who are shown in the “police” table in the database can login.
- **In charge Login:** For every are there is an in charge who keeps track of all the cases. The person who is in charge logs-in from here. In charge is responsible for assigning the case to the respective police officers. Only the police officials declared who are shown in the “police_station” table in the database can login.
- **HQ Login:** Head Quarter login is for a person like Commissioner. The HQ login can see all the things, which complaint has been logged from where, how many cases are solved many things.

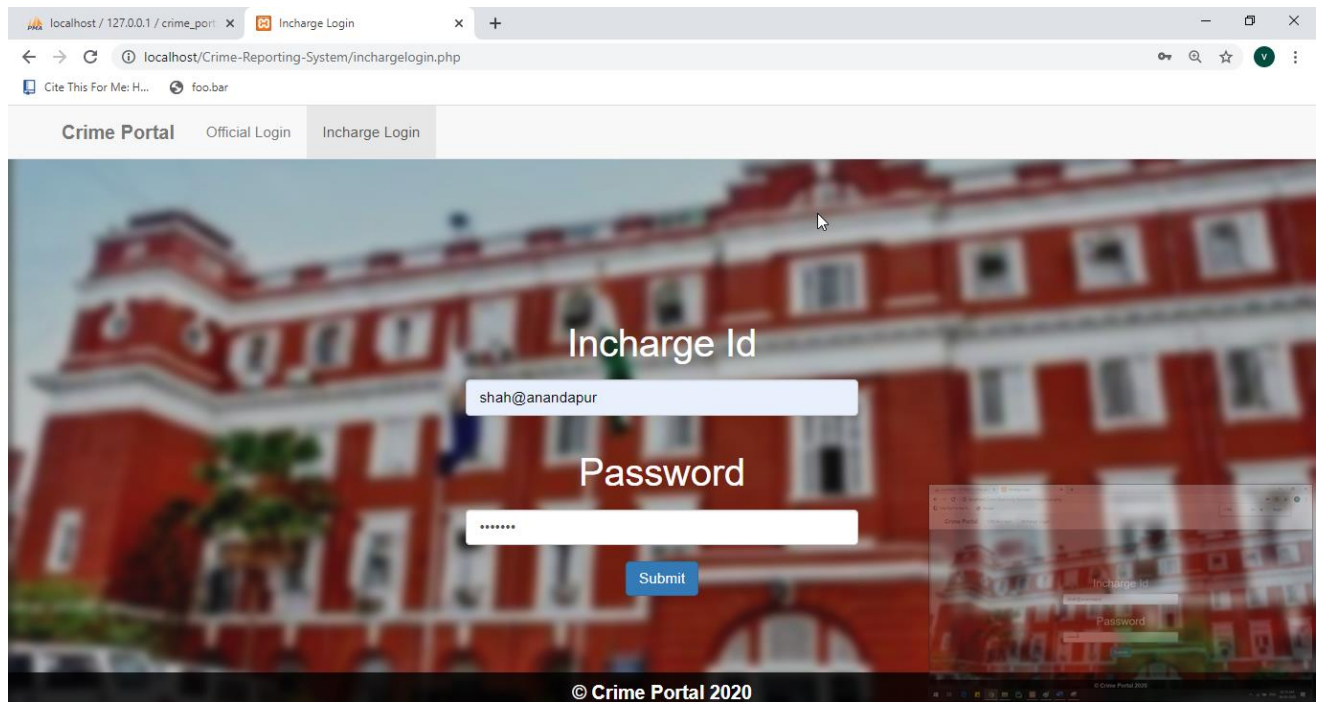


Fig 6.1.12 “Incharge Login” page.

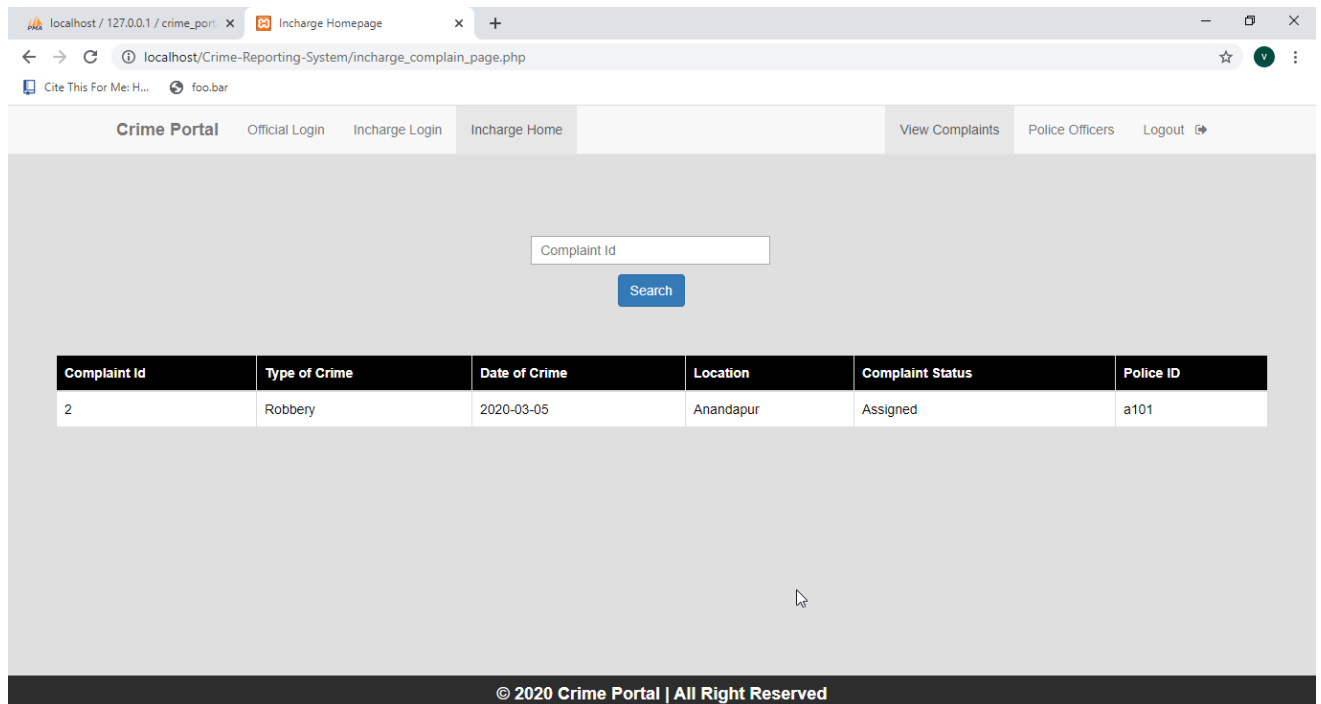


Fig 6.1.13 “Incharge Complain” page.

Here, in charge searches the complaint and assigns the police officials.

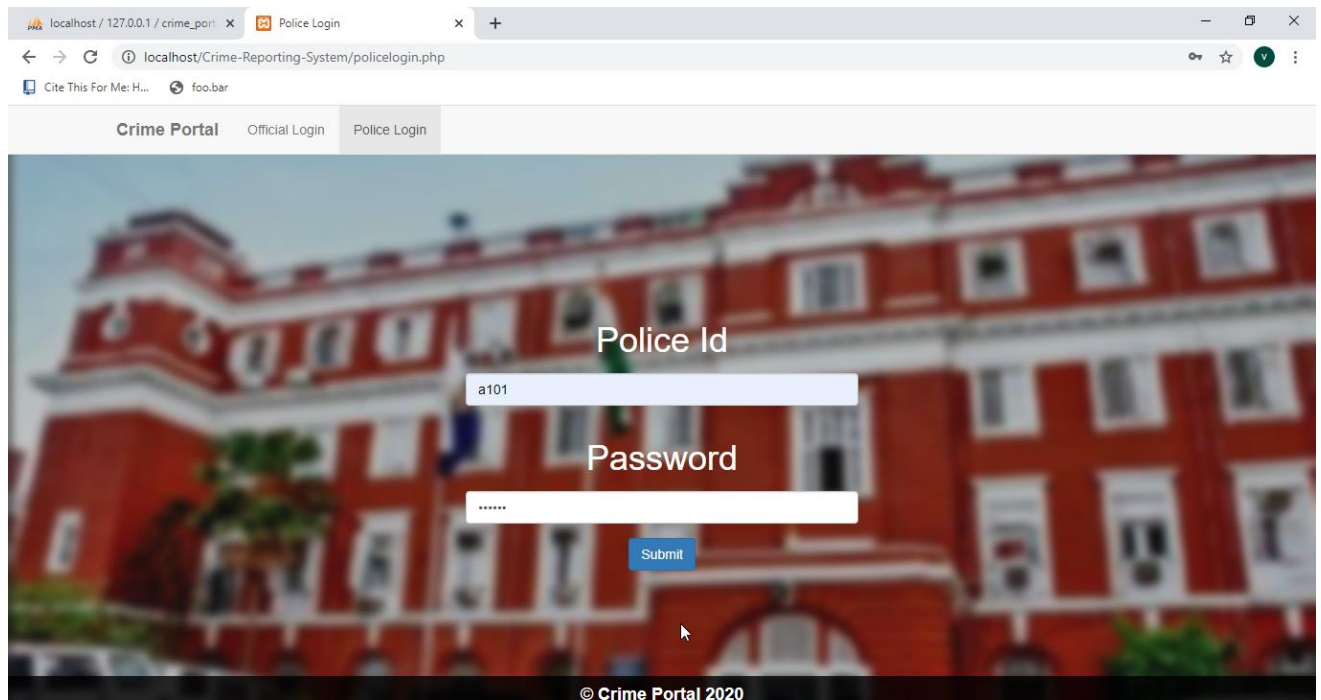


Fig 6.1.14 “Police Login” page.

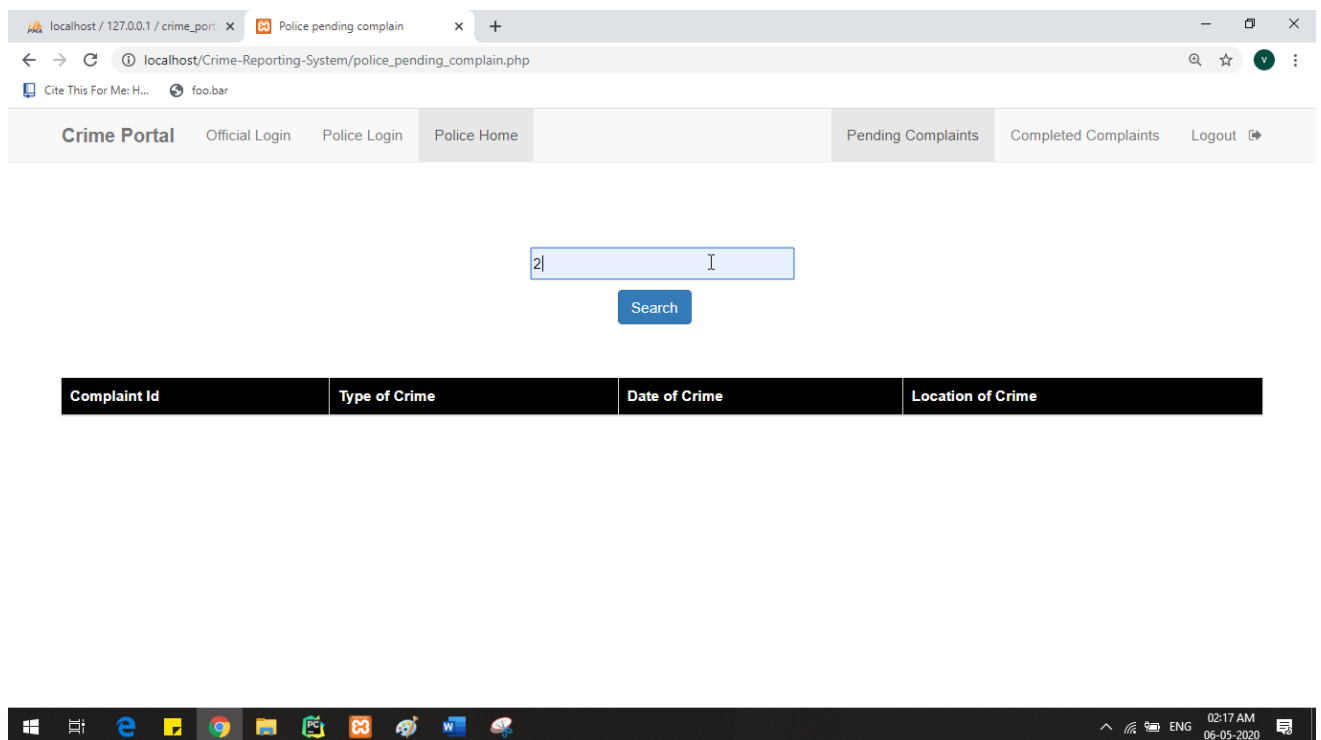


Fig 6.1.15 “Police Pending Complaint” page.

Here, the police officer can do two things:

- First, he/she can update the status of the case assigned to him/her.
- Second, the officer can check the “Completed Complaints”.

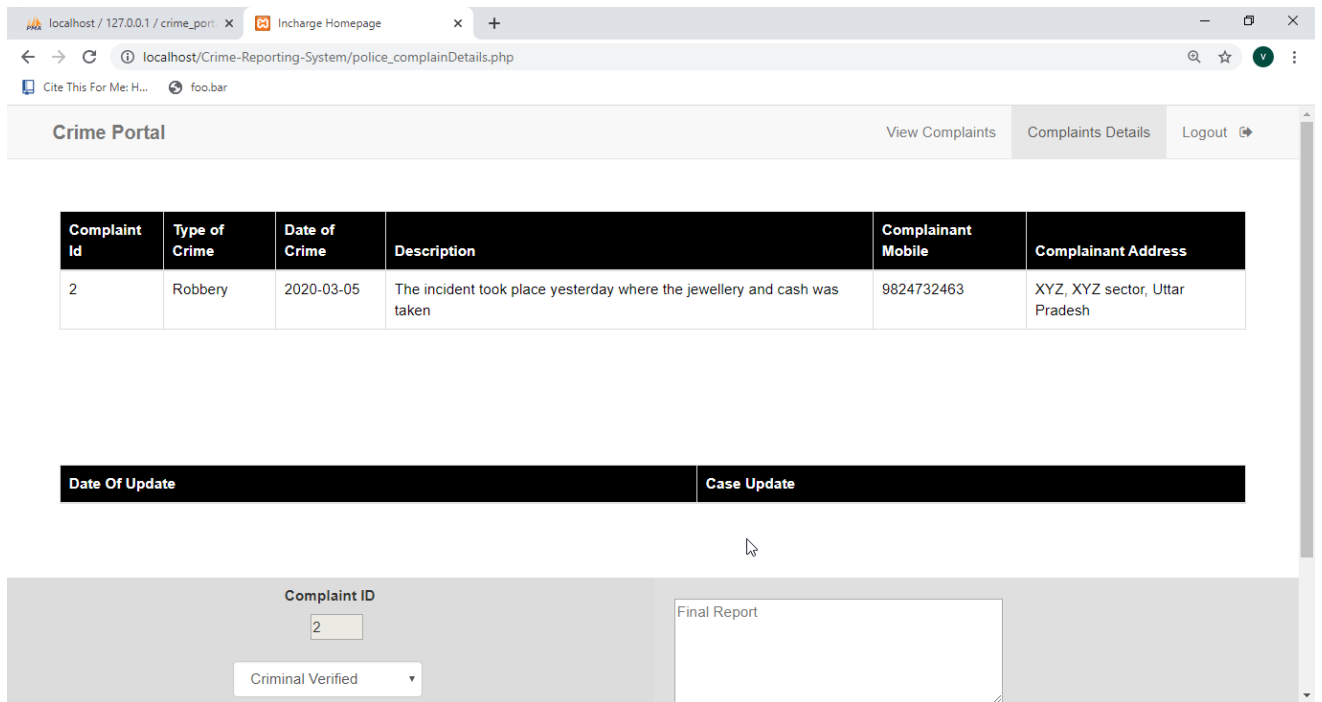


Fig 6.1.16 “Police Complaint Details” Page.

Here, the police official can update the status of the case and close the case once the case is solved.

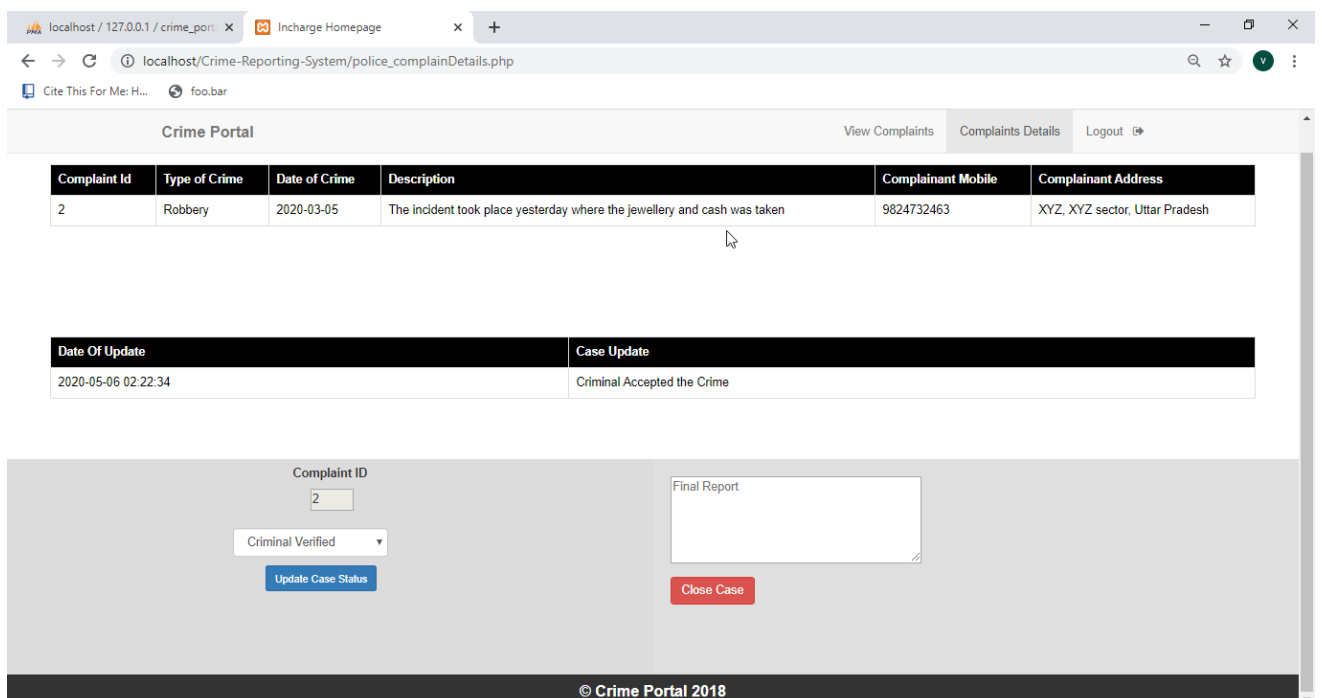


Fig 6.1.17 Status updated to “Criminal accepted Crime”.

This will be reflected in the database also.

	c_id	d_o_u	case_update
<input type="checkbox"/>	1	2018-12-17 16:02:06	Criminal Verified
<input type="checkbox"/>	1	2018-12-17 16:02:12	Criminal Caught
<input type="checkbox"/>	1	2018-12-17 16:02:15	Criminal Interrogated
<input type="checkbox"/>	1	2018-12-17 16:02:21	Criminal Accepted the Crime
<input type="checkbox"/>	1	2018-12-17 16:02:26	Criminal Charged
<input type="checkbox"/>	1	2018-12-17 16:02:51	The case has been moved to Court.
<input type="checkbox"/>	1	2018-12-17 16:02:59	Criminal Verified
<input type="checkbox"/>	2	2020-05-06 02:22:34	Criminal Accepted the Crime

Fig 6.1.18 Successful updating of complaint represented by black rectangular box.

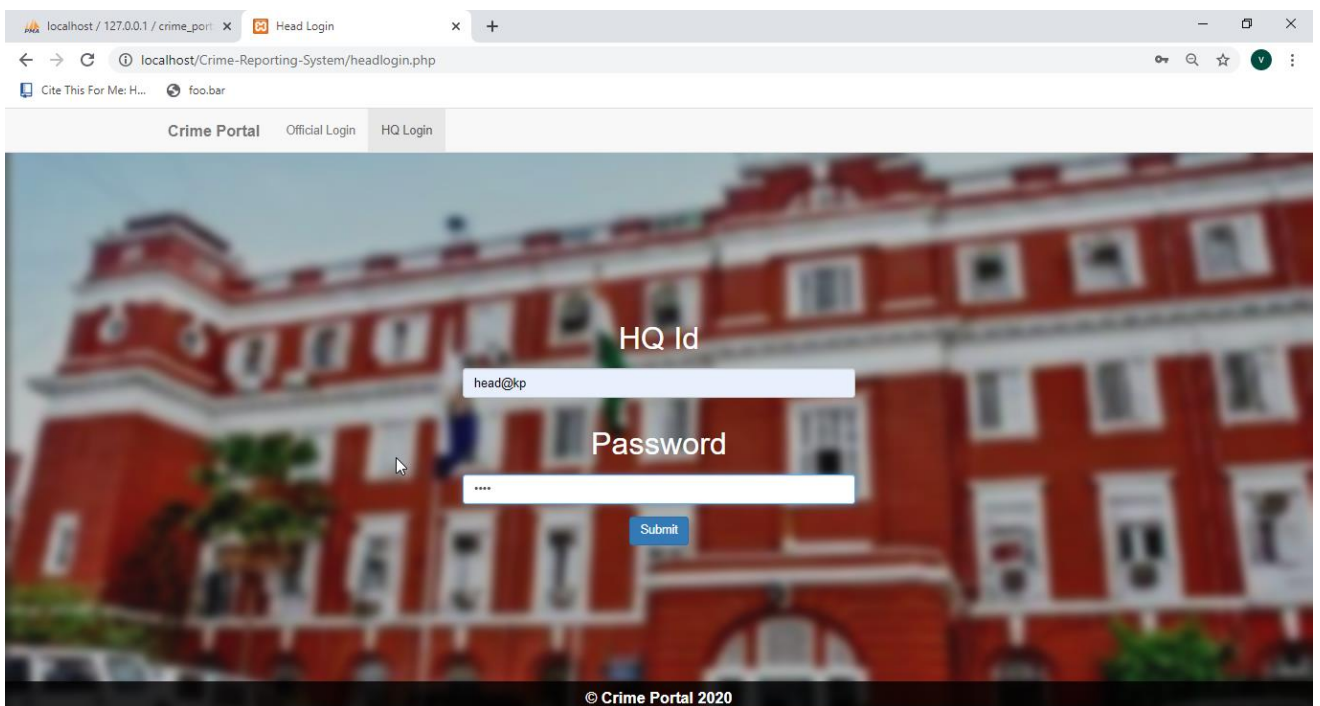


Fig 6.1.19 “Head Quarter Login” page.

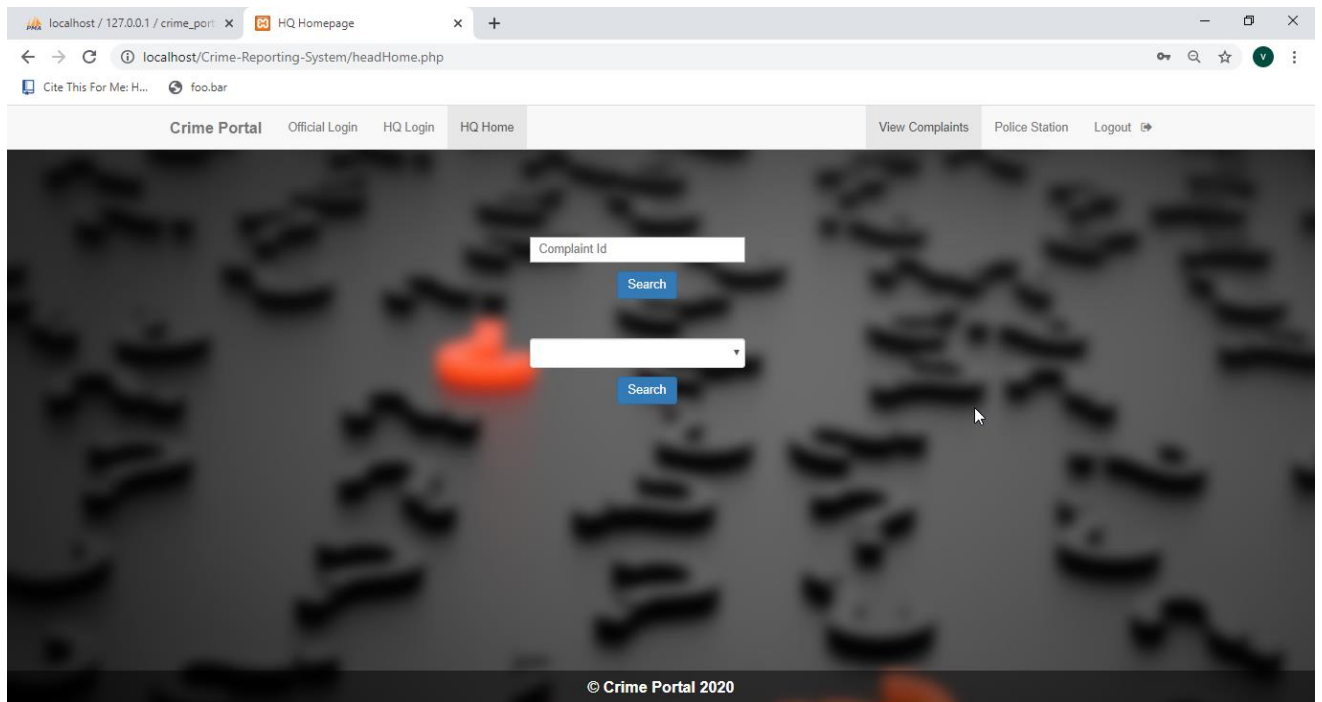


Fig 6.1.20 “Head Quarter Home” page.

From, here the head official(Commissioner) can keep a track of any case at any branch.

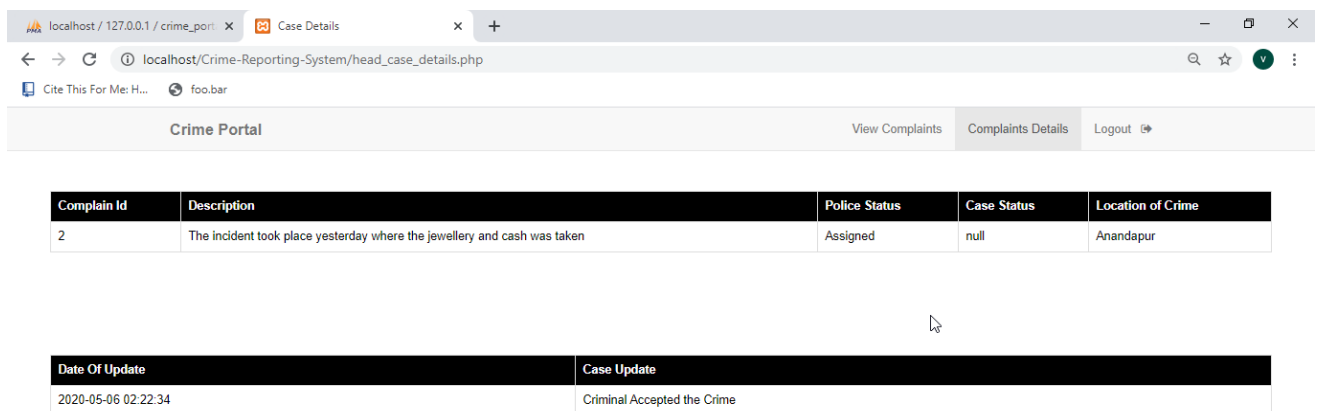


Fig 6.1.21 “Case Details” page.

CONCLUSIONS /PROJECT SUMMARY

The **ONLINE CRIME MANAGEMENT** is a web-based application for primarily providing training to the employees who provide customized solutions to meet organizational needs.

This application software has been computed successfully and was also tested successfully by taking “test cases”. It is user friendly, and has required options, which can be utilized by the user to perform the desired operations.

The software is developed using Java as front end and Oracle as back end in Windows environment. The goals that are achieved by the software are:

- ✓ Instant access.
- ✓ Improved productivity.
- ✓ Optimum utilization of resources.
- ✓ Efficient management of records.
- ✓ Simplification of the operations.
- ✓ Less processing time and getting required information.
- ✓ User friendly.

FUTURE IMPROVEMENT

- This System being web-based and an undertaking of Cyber Security Division, needs to be thoroughly tested to find out any security gaps.
- A console for the data center may be made available to allow the personnel to monitor on the sites which were cleared for hosting during a particular period.
- Moreover, it is just a beginning; further the system may be utilized in various other types of auditing operation viz. Network auditing or similar process/workflow-based applications...

REFERENCES:

- [1]"MySQL Tutorial - Tutorialspoint", *Tutorialspoint.com*, 2020. [Online]. Available: <http://www.tutorialspoint.com/mysql/>.
- [2]2020. [Online]. Available: <https://www.apachefriends.org/index.html>.
- [3]"HTML Tutorial", *W3schools.com*, 2020. [Online]. Available: <https://www.w3schools.com/html/>.
- [4]"CSS Tutorial", *W3schools.com*, 2020. [Online]. Available: <https://www.w3schools.com/css/>.
- [5]"JavaScript Tutorial", *W3schools.com*, 2020. [Online]. Available: <https://www.w3schools.com/js/>.
- [6]"Learn JavaScript - Free Interactive JavaScript Tutorial", *Learn-js.org*, 2020. [Online]. Available: <https://www.learn-js.org/>.
- [7]"PHP Tutorial", *W3schools.com*, 2020. [Online]. Available: <https://www.w3schools.com/php/>.