**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction**

The project titled as **Crime Reporting System** is a web-based application. This Software provide facility for reporting online FIR’s, view missing persons, show most wanted person details, stolen vehicles, etc. Any number of clients can connect to the server. Each user first makes their login to server to show their availability. The **Crime Reporting System** project is to provide all crime management solutions which are easily accessible to everyone. The Crime application starts with the common people who want to log a complaint through the website so it can be very useful for police department to find out the problem in the society without people are coming to the Police station every time.

**PURPOSE:**

* To reduce the trouble of going to Police station frequently.
* To take a quick action against crime.
* To get the information about the crime easily.
* Reduce the man power.
* Reduces the time.
* Provides the current status of the crime rate.
* To aware people about crime.

**1.2 DBMS (DATABASE MANAGEMENT SYSTEM)**

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.

A DBMS relieves users of framing programs for data maintenance. Fourth-generation query languages, such as SQL, are used along with the DBMS package to interact with a database.

Some other DBMS examples include:

* MySQL
* SQL Server
* Oracle
* dBase
* FoxPro

**1.3 PHP (HYPERTEXT PREPROCESSOR)**

PHP is the most popular and widely used server-side scripting language for web development. It is used to make the Dynamic pages in websites. Rasmus Lerdorf was the creator of PHP in 1995. PHP codes are embedding in HTML source codes for making the page dynamic. PHP can deal with most of the requirements in web development like Database, File handling, String operations, Arrays, Graphics, File Uploads, Data processing etc. PHP can be used in any operating system with a web server Supports PHP.

Apache web server is one of the popular web servers dealing with PHP + MySQL. Moreover, PHP is absolutely free to use.

**1.4 PROBLEM STATEMENT**

We all know that nowadays crime rate is frequently increasing day by day in our society. From this we get an idea to develop the system. Problem was that people got tired by going here and there for getting justice. So, our application is capable of registering FIR online, shows investigation update, deliver news about crime etc. So, it is an application which provide solution to the problems faced during taking action against crime.

**1.5 OBJECTIVES**

* We know users are of many categories, like users from who know working with computers very well and users who didn’t know about computers. So, all the categories can use the system. So, it should be user friendly
* The software is made to work efficiently and effectively. It results in regular and timely action against crime reported. If can be observed that the information can be obtained easily and accurately. Better communication, Reducing crime and the Entire functioning less time consuming.
* The main aim of this project is to secure and make privacy on crime related data over manually data storage. This website base project is made for providing the information and awareness about the crimes.

**1.6 EXISTING SYSTEM**

In the existing system only, we can see the details of particular information about the Police stations in a particular state, the existing system has more workload for the authorized person.

**1.7 PROPOSED SYSTEM**

In the proposed system, the any person who have witnessed crime can file an online complaint in no of crime taken place. Easily the user can register in the online site and admins can identify the crime faster. Faster the crime is noticed the action taken will be much more enhanced and quicker.

**CHAPTER 2**

**REQIUREMENT SPECIFICATION**

**2.1 HARDWARE REQUIREMENTS:**

The hardware required for the development of this project is:

* Processor - Intel Core i5
* Processor speed - 1.7 GHz
* RAM - 2 GB RAM
* System Type - 64-Bit Operating System

**2.2 SOFTWARE REQUIREMENTS**

The software required for the development of this project is:

* Software - XAMPP
* Operating System - Windows 7 (and another higher version)
* Front End - HTML, CSS
* Programming Language - PHP
* Data Base Environment - MySQL and PhpMyAdmin
* Server - APACHE

**CHAPTER 3**

**DESIGNS**

**3.1 ENTITY - RELATIONSHIP DIAGRAM**

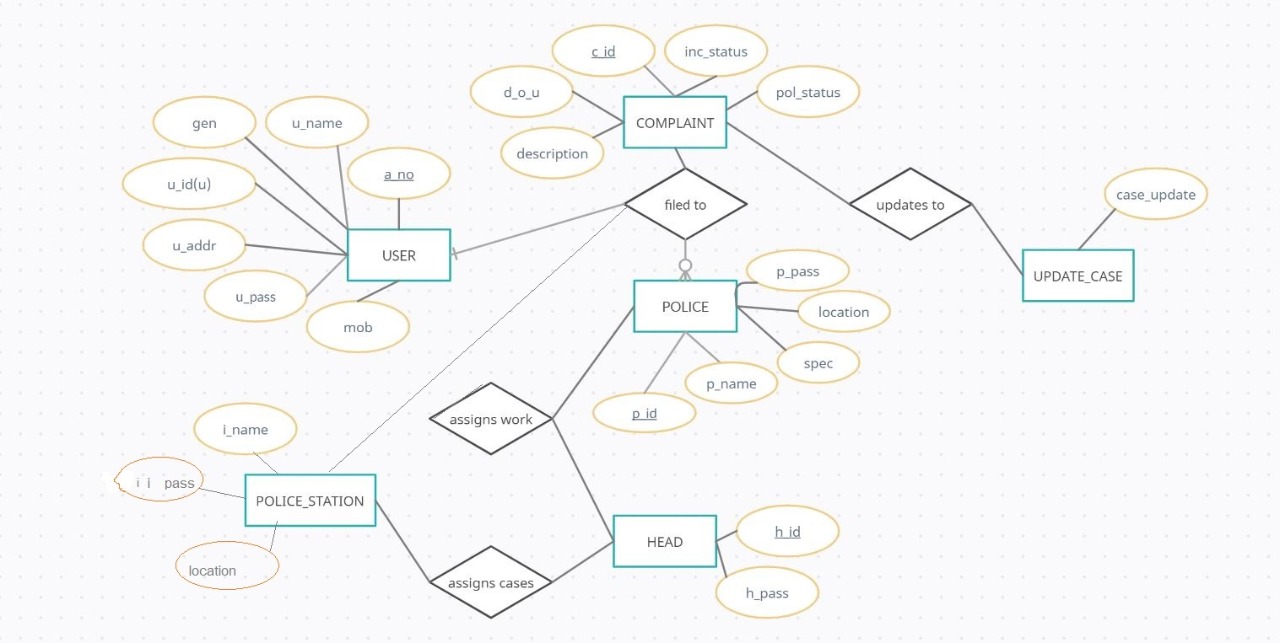


Fig 3.1: Entity Relationship Diagram of Crime Reporting system

Entity

Attribute

Relationship Type

Primary Key

1

Cardinality ratio

N

**3.2 DESCRIPTION:**

The ER Model figure shows the proposed system. It defines the conceptual view of the database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases. So, let’s see each entity

**crime\_portal**

This entity stores the information of the complete database of the crime reporting system. Attributes are complaint, head, police, police\_station, update\_case, user.

**user**

This entity stores the information about the candidate who stand files the complaint. Attributes are u\_name, u\_id, u\_pass, u\_addr, a\_no, gender, mob.

**complaint**

This entity stores the information about the complaint filed. Attributes are c\_id, a\_no, location, Type\_crime, d\_o\_c, description, inc\_status, pol\_status, p\_id.

**head**

This entity stores information about the higher official’s login page. Attributes are h\_id, h\_pass.

**police**

This entity stores the details of police officers. Attributes p\_name, p\_id, spec, location, p\_pass.

**police\_station**

This entity stores the details about all the police stations. Attributes i\_id, i\_name, location, i\_pass.

**update\_case**

This entity stored the details about all the updates of a case that is filed by user. Attributes c\_id,

d\_o\_u, case\_update.

**3.3 SCHEMA DIAGRAM**

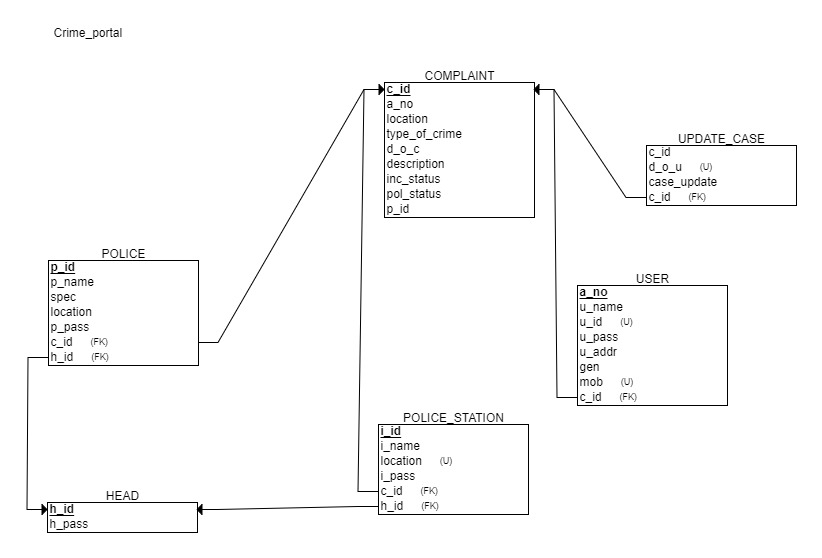


Fig 3.3 Schema Diagram Online Crime Reporting System.

**3.4 SEVEN STEPS FOR ER TO SCHEMA CONVERSION**

**Step 1: Mapping of Regular Entity Types.**

For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E. Include only the simple component attributes of a composite attribute. Choose one of the key attributes of E as the primary key for R. If the chosen key of E is a composite, then the set of simple attributes that form it will together form the primary key of R. If multiple keys were identified for E during the conceptual design, the information describing the attributes that form each additional key is kept in order to specify secondary (unique) keys of relation R. Knowledge about keys is also kept for indexing purposes and other types of analyses.

**Step 2: Mapping of Weak Entity Types.**

For each weak entity type W in the ER schema with owner entity type E, create a relation

R and include all simple attributes (or simple components of composite attributes) of was attributes of R. In addition, include as foreign key attributes of R, the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s); this takes care of mapping the identifying relationship type of W. The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any. If there is a weak entity type E2 whose owner is also a weak entity type E1, then E1 should be mapped before E2 to determine its primary key first.

**Step 3: Mapping of Binary 1:1 Relationship Types.**

For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R. There are three possible approaches:

1. The foreign key approach.

2. The merged relationship approach, and

The first approach is the most useful and should be followed unless special conditions exist, as we discuss below.

**1. Foreign key approach*:***

Choose one of the relations—S, say—and include as a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S. Include all the simple attributes (or simple components of composite attributes) of the 1:1 relationship type R as attributes of S.

**2. Merged relation approach:**

An alternative mapping of a 1:1 relationship type is to merge the two entity types and the relationship into a single relation. This is possible when both participations are total, as this would indicate that the two tables will have the exact same number of tuples at all times.

**3. Cross-reference or relationship relation approach:**

The third option is to set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types. As we will see, this approach is required for binary M:N relationships. The relation R is called a relationship relation (or sometimes a lookup table), because each tuple in R represents a relationship instance that relates one tuple from S with one tuple from T. The relation R will include the primary key attributes of S and T as foreign keys to S and T. The primary key of R will be one of the two foreign keys, and the other foreign key will be a unique key of R. The drawback is having an extra relation, and requiring an extra join operation when combining related tuples from the tables.

**Step 4: Mapping of Binary 1:N Relationship Types.**

For each regular binary 1:N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R; we do this because each entity instance on the N-side is related to at most one entity instance on the 1-side of the relationship type. Include any simple attributes (or simple components of composite attributes) of the 1:N relationship type as attributes of S.

**Step 5: Mapping of Binary M:N Relationship Types.**

For each binary M:N relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

Notice that we cannot represent an M:N relationship type by a single foreign key attribute in one of the participating relations (as we did for 1:1 or 1:N relationship types) because of the M:N cardinality ratio; we must create a separate relationship relation S.

**Step 6: Mapping of Multivalued Attributes.**

For each multivalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as a multivalued attribute.

The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

**Step 7: Mapping of N-array Relationship Types.**

For each n-array relationship type R, where n > 2, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types. Also include any simple attributes of the n-array relationship type (or simple components of composite attributes) as attributes of S. The primary key of S is usually a combination of all the foreign keys that reference the relations representing the participating entity types. However, if the cardinality constraints on any of the entity types E participating in R is 1, then the primary key of S should not include the foreign key attribute that references the relation E ‘corresponding to E.

**3.5 DATABASEDESCRIPTION** :

crime\_portal**:**

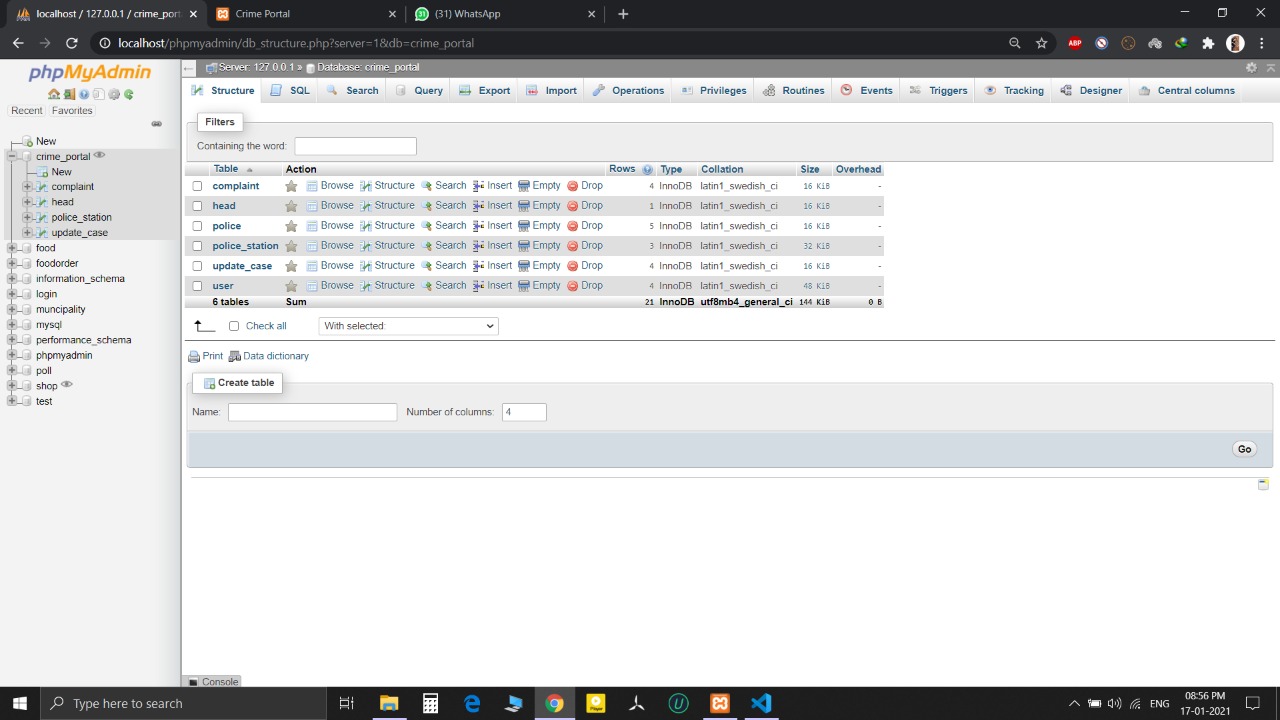


Table 3.5.1: Description of project database

**user:**

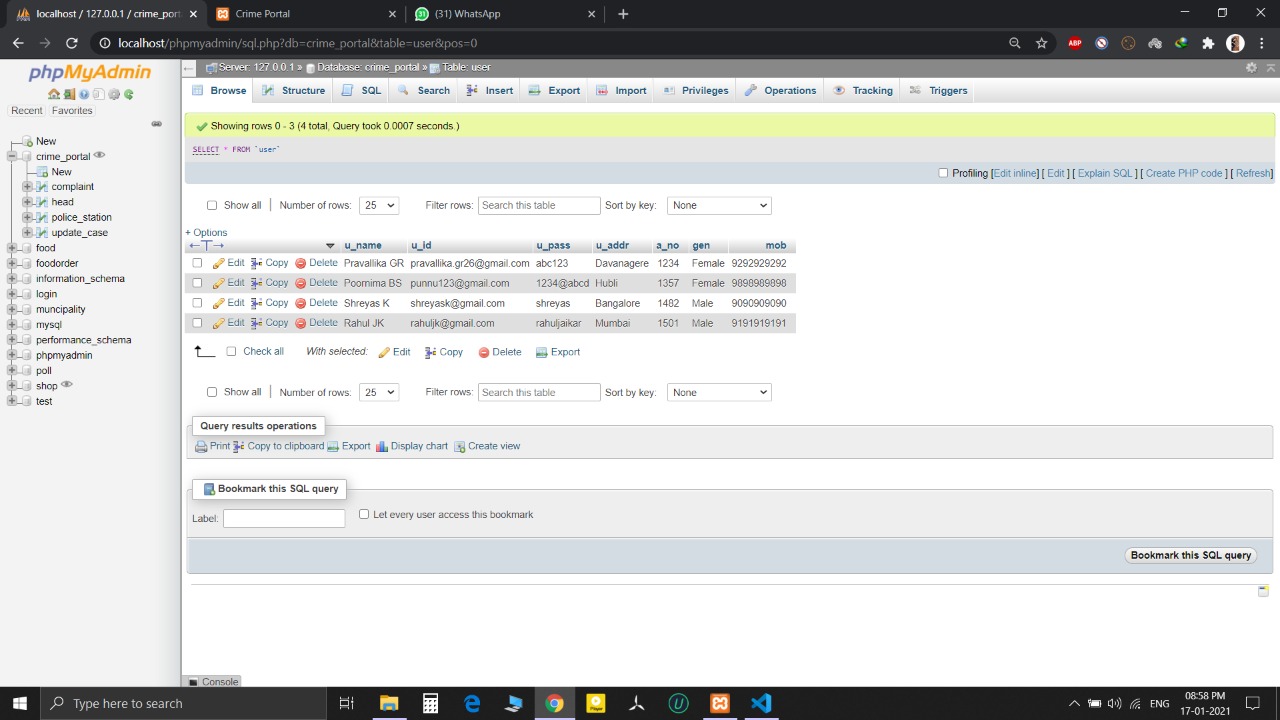


Table 3.5.2: Description of registered user.

**complaint:**

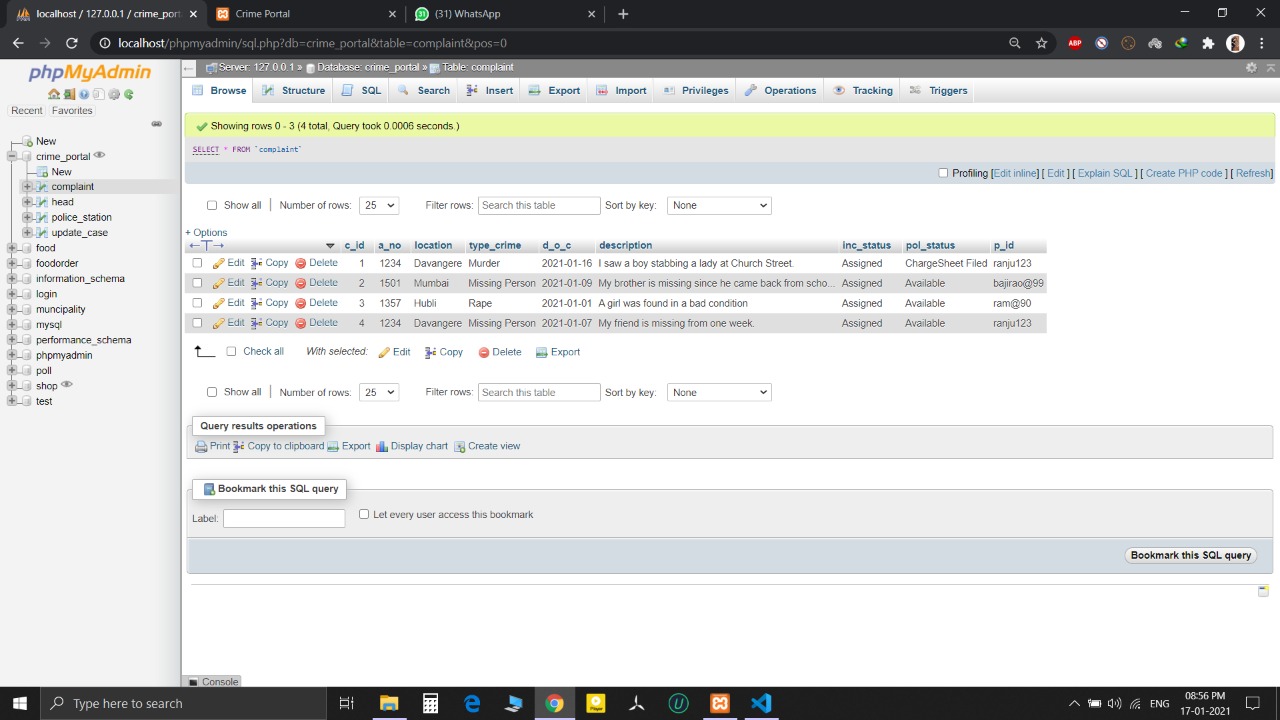


Table 3.5.3: Table description for registered complaints.

**head:**

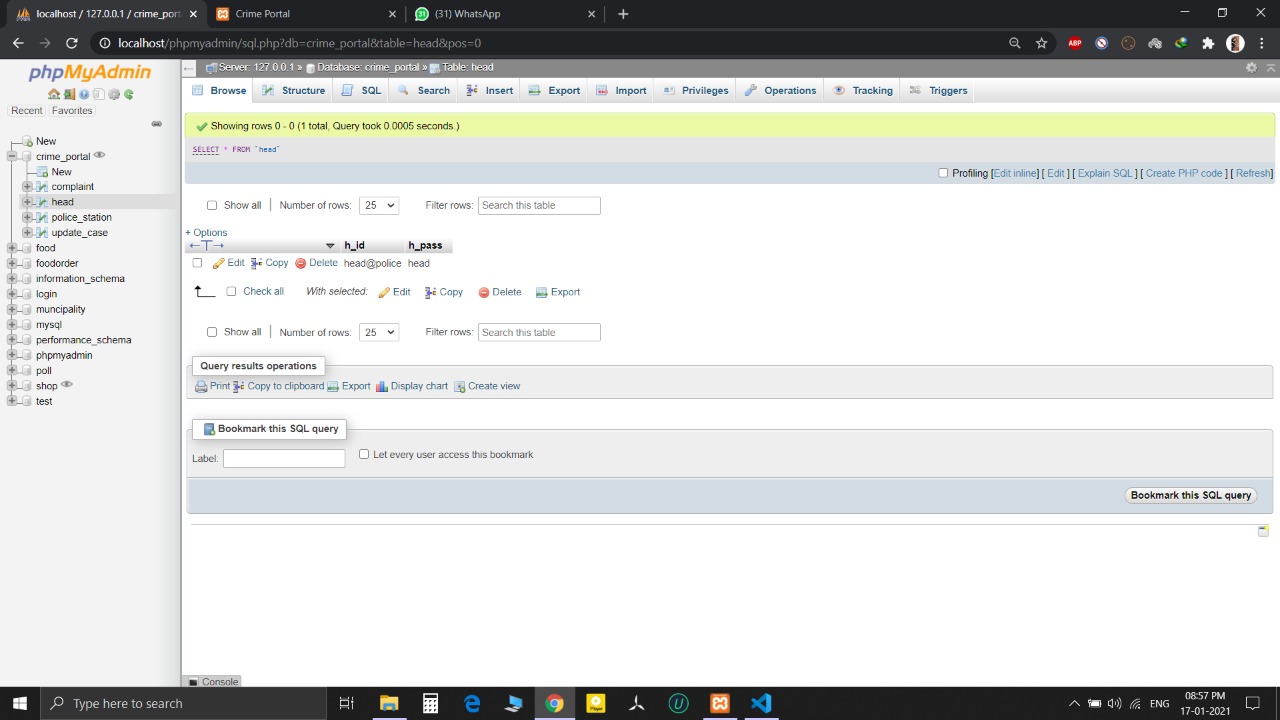


Table 3.5.4: Table description for Head Quarters login.

**police:**

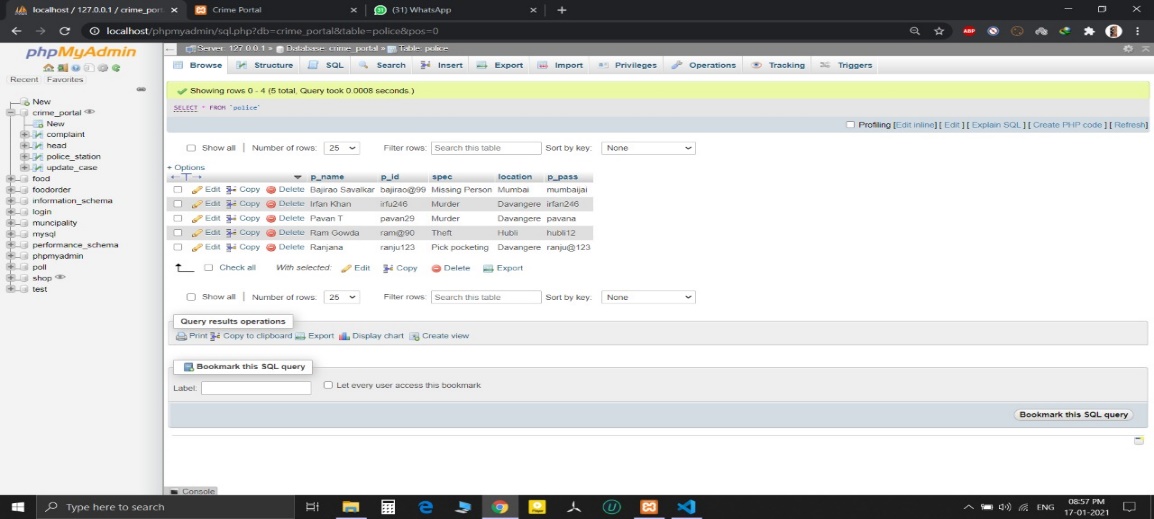


Table 3.5.5: Table description about Police.

**police\_station:**

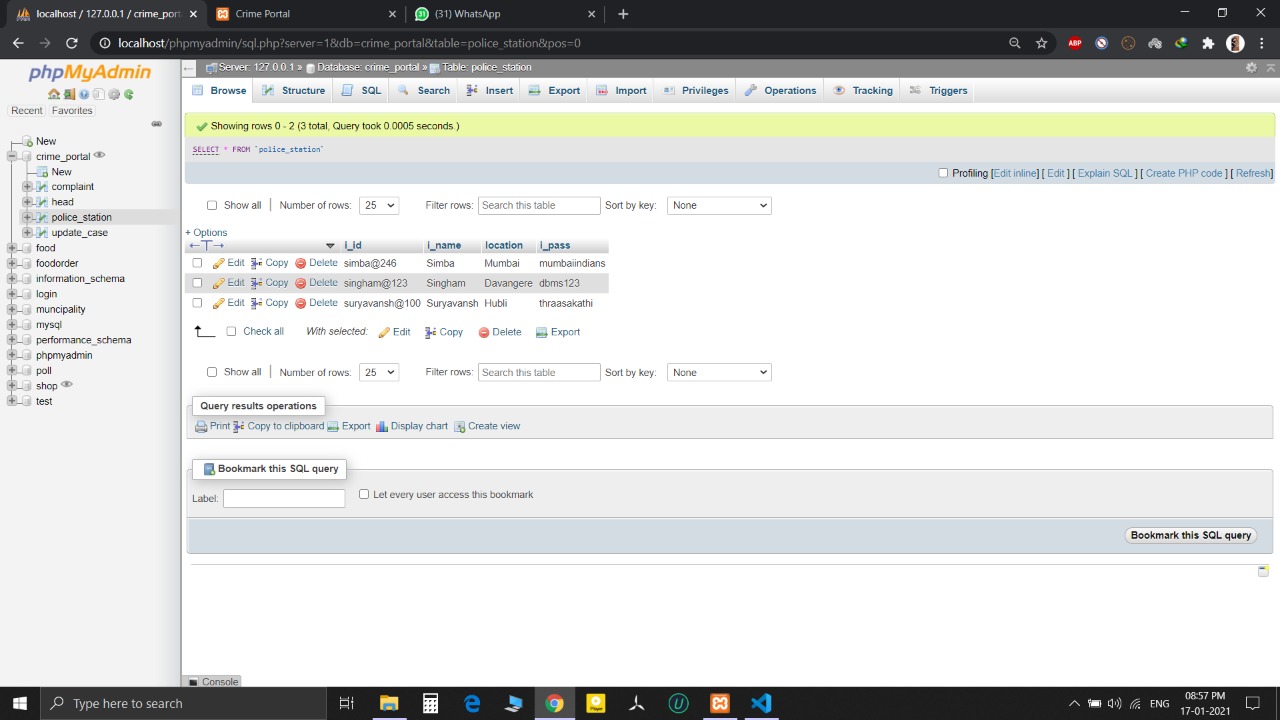


Table 3.5.6: Description of Police station.

**update\_case:**

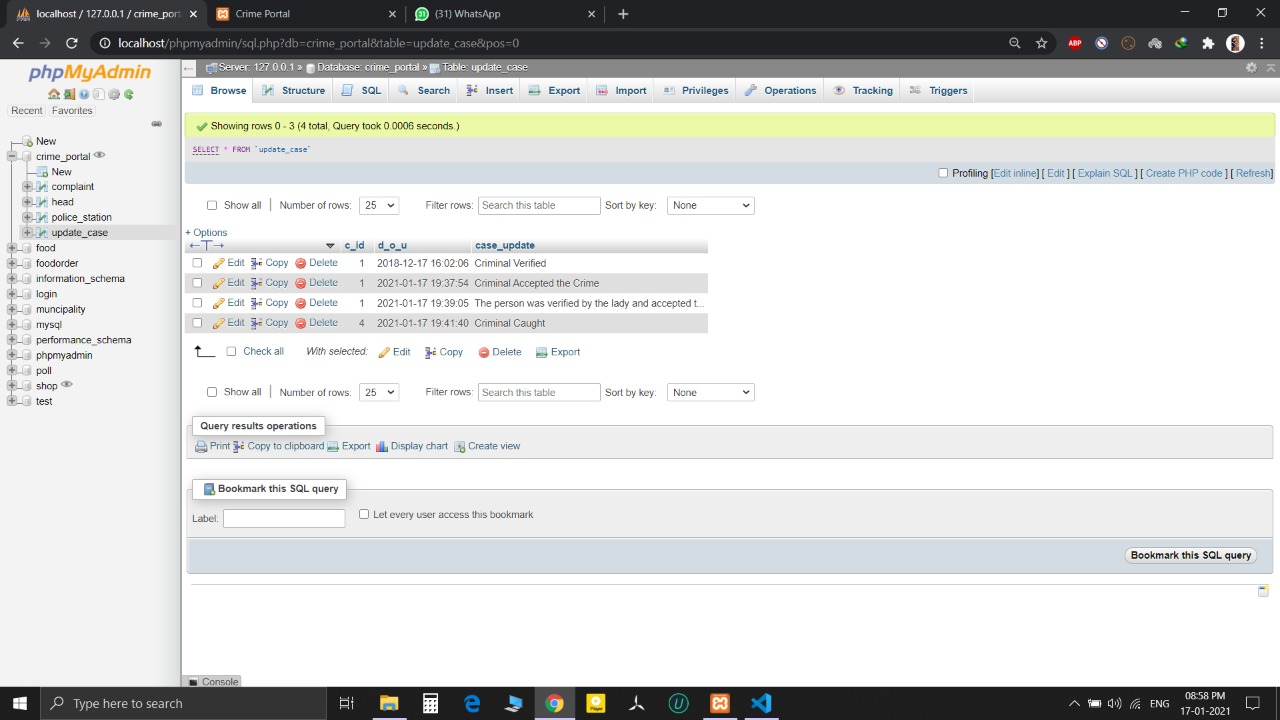


Table 3.5.7: Table description of Update case.

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 CODE**

**4.1.1 CONNECTION CODE FOR FRONT END TO BACK END**

<?php

error\_reporting(1);

$con=mysqli\_connect('localhost', 'root', '','poll') or die(mysqli\_error());

?>

**4.1.2 CODE FOR INSERT:**

insert(registration as a user):

<?php

if(isset($\_POST['s'])){

$con=mysqli\_connect('localhost','root','','crime\_portal');

if(!$con)

{

die('could not connect: '.mysqli\_error());

}

if($\_SERVER["REQUEST\_METHOD"]=="POST"){

$u\_name=$\_POST['name'];

$u\_id=$\_POST['email'];

$u\_pass=$\_POST['password'];

$u\_addr=$\_POST['adress'];

$a\_no=$\_POST['aadhar\_number'];

$gen=$\_POST['gender'];

$mob=$\_POST['mobile\_number'];

$reg="insert into user values('$u\_name','$u\_id','$u\_pass','$u\_addr','$a\_no','$gen','$mob')";

mysqli\_select\_db("crime\_portal");

$res=mysqli\_query($con,$reg);

if ($conn->query($sql) === TRUE) {

echo "New record created successfully";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

**4.1.3 CODE FOR DELETE:**

delete(deletion of police):

<?php

session\_start();

if(!isset($\_SESSION['x']))

header("location:inchargelogin.php");

$conn=mysqli\_connect("localhost","root","","crime\_portal");

if(!$conn)

{

die("could not connect".mysqli\_error());

}

mysqli\_select\_db("crime\_portal",$conn);

$i\_id=$\_SESSION['email'];

$result1=mysqli\_query($conn,"SELECT location FROM police\_station where i\_id='$i\_id'");

$q2=mysqli\_fetch\_assoc($result1);

$location=$q2['location'];

if(isset($\_POST['s2']))

{

if($\_SERVER["REQUEST\_METHOD"]=="POST")

{

$pid=$\_POST['pid'];

$q1=mysqli\_query($conn,"delete from police where p\_id='$pid'");

$q3=mysqli\_query($conn,"update complaint set pol\_status='null',inc\_status='Unassigned',p\_id='Null' where p\_id='$pid'");

}

}$result=mysqli\_query($conn,"select p\_id,p\_name,spec,location from police where location='$location'");

?>

**4.1.4 CODE FOR UPDATE:**

trigger(update of case and date):

<div style="padding:50px; margin-top:8px;">

<table class="table table-bordered">

<thead class="thead-dark" style="background-color: black; color: white;">

<tr>

<th scope="col">Date Of Update</th>

<th scope="col">Case Update</th>

</tr>

</thead>

<?php

while($rows1=mysqli\_fetch\_assoc($res2)){

?>

<tbody style="background-color: white; color: black;">

<tr>

<td><?php echo $rows1['d\_o\_u']; ?></td>

<td><?php echo $rows1['case\_update']; ?></td>

</tr>

</tbody>

<?php

}

?>

</table>

</div>

**CHAPTER 5**

**SNAPSHOTS.**

**User Register:**

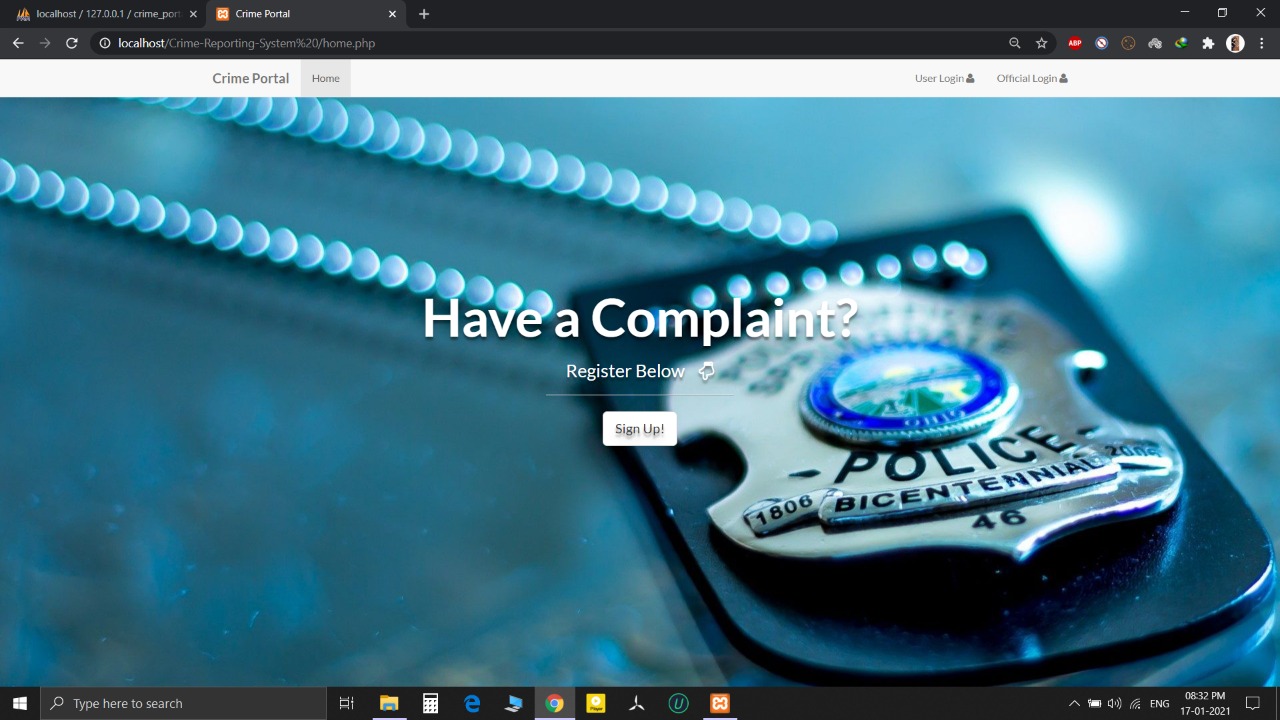


Image 5.1: Click On Sign-in To Register.

**User Details:**

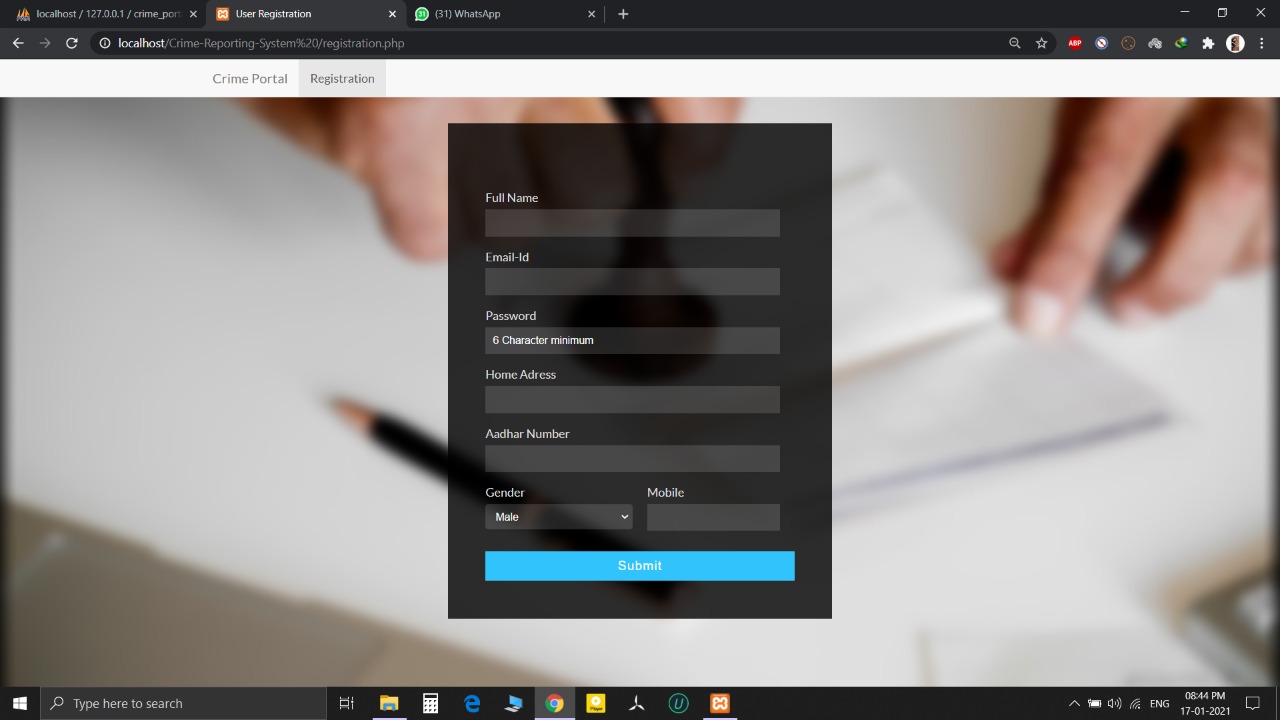


Image 5.2: User Details To Sign-in.

**Complainer Login:**

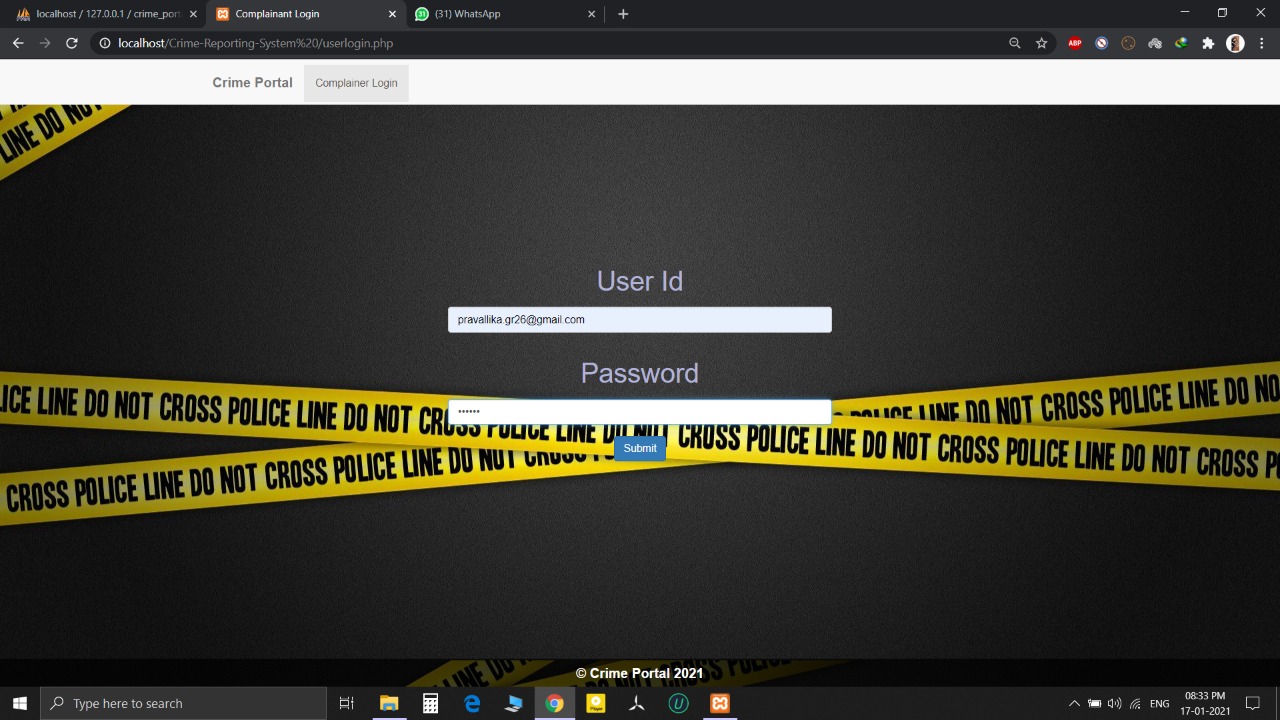


Image 5.3: User Login For Registering Complaints.

**Official Login:**

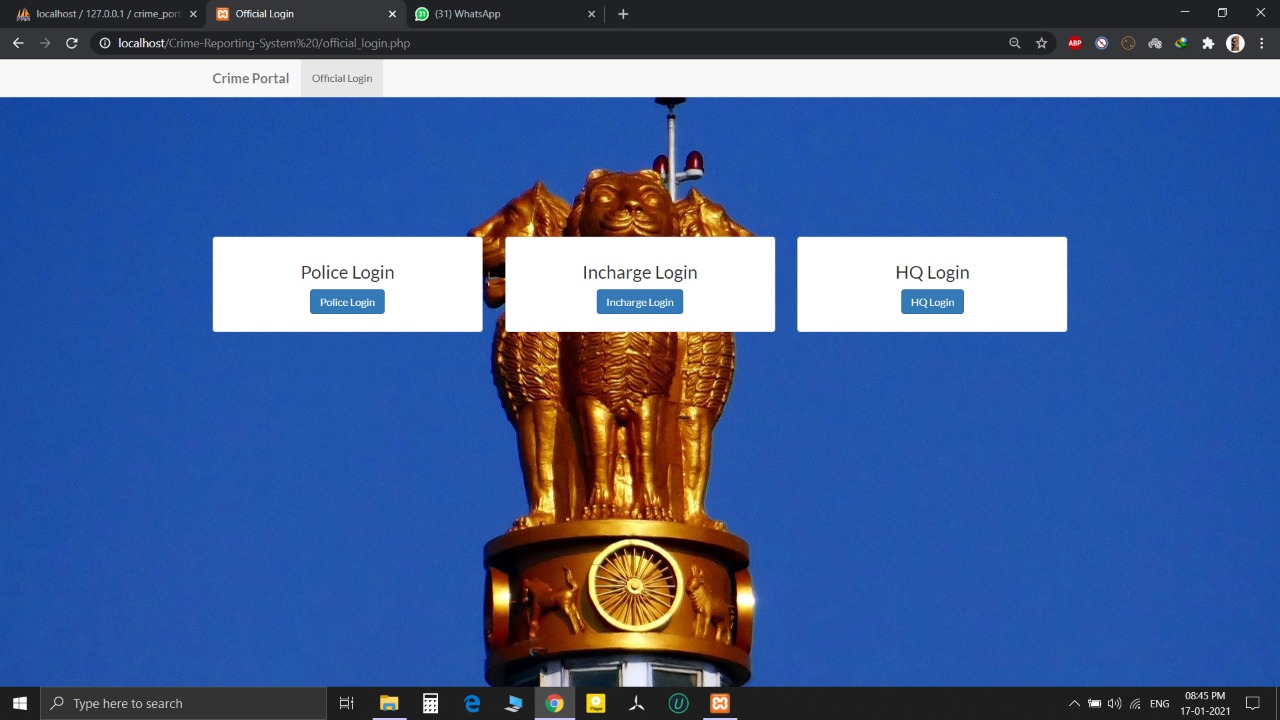


Image 5.4: Official Login Home Page.

**Head Quarters:**

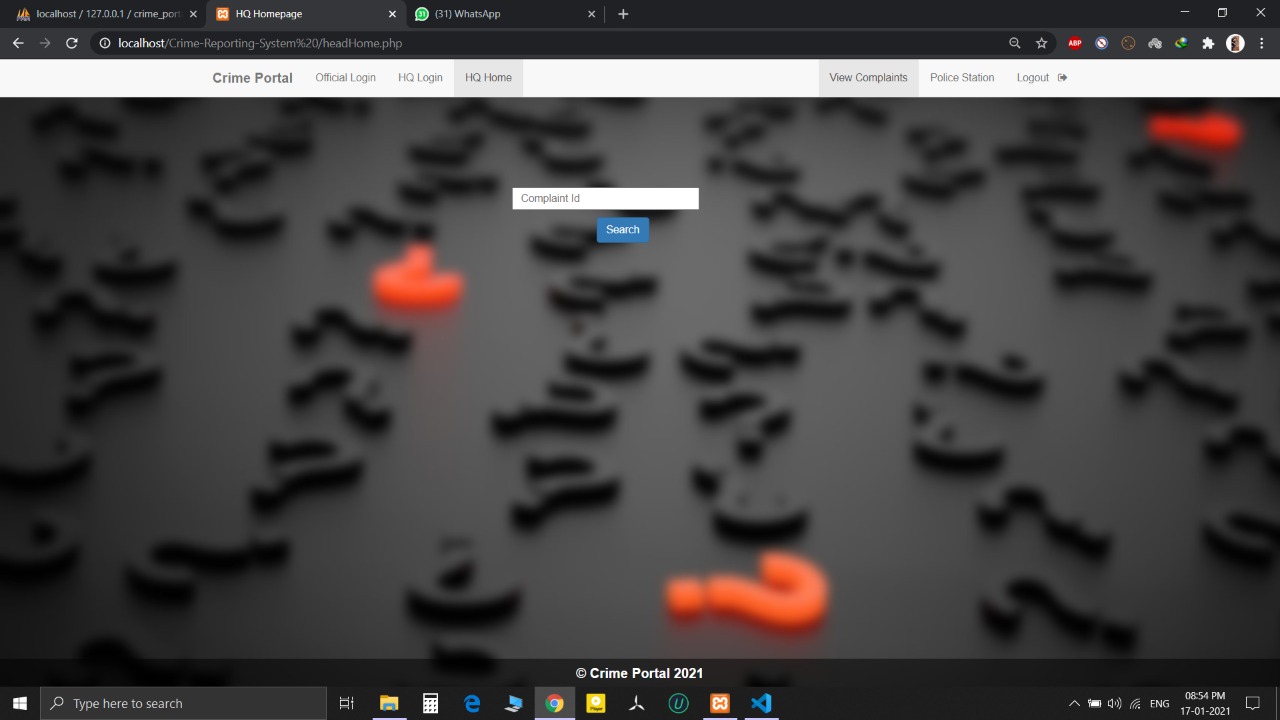


Image 5.5: Login Page Of Head Quarters.

**Police Login:**

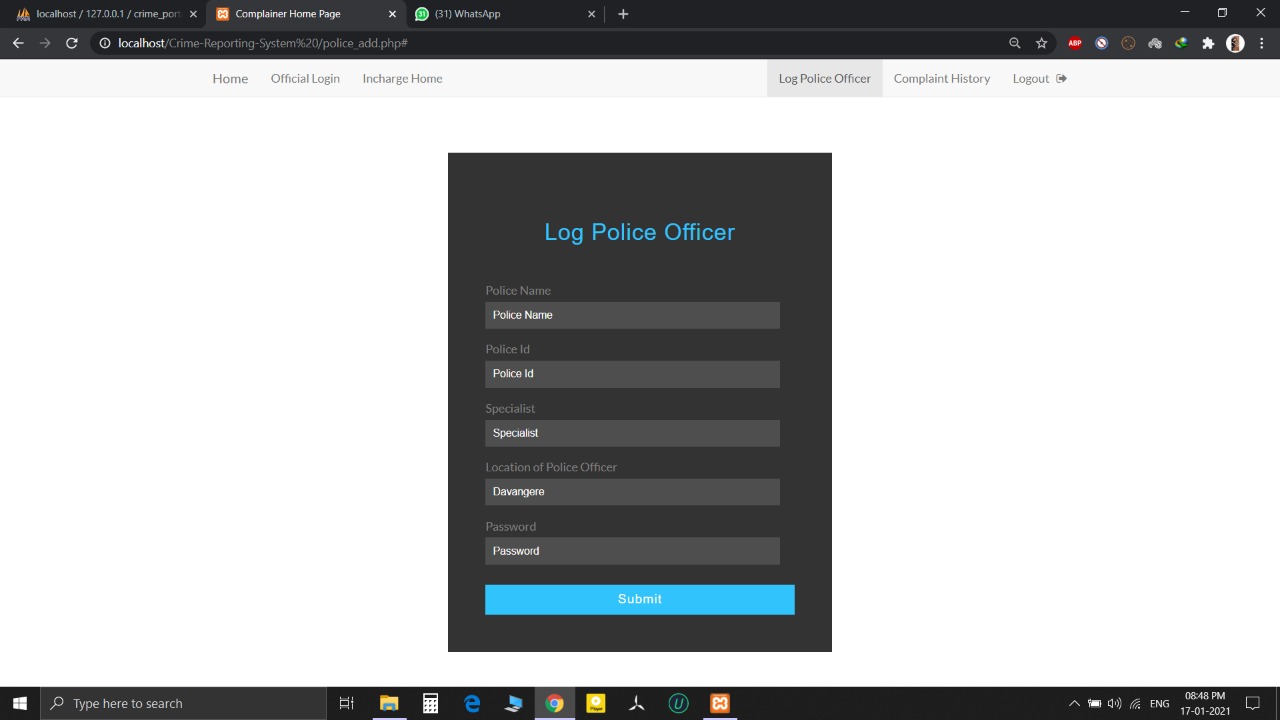


Image 5.6: Police Login.

**Incharge Login:**

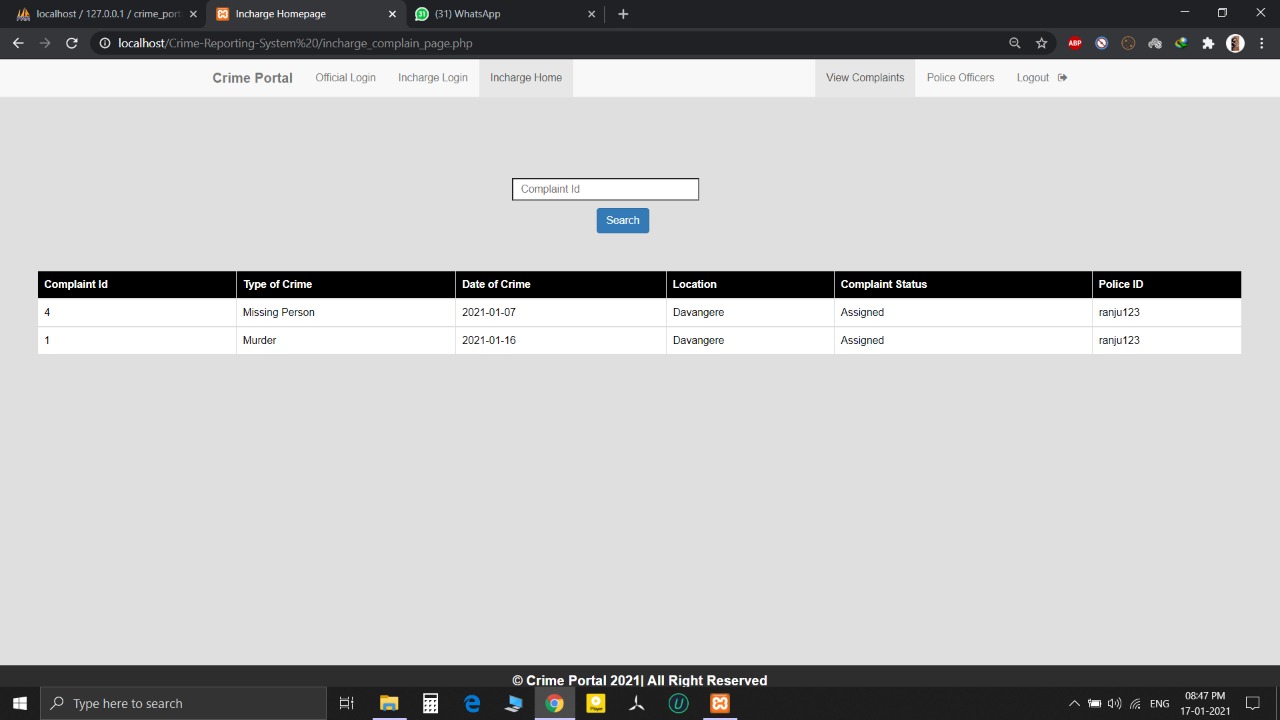


Image 5.7: Incharge Login.

**Update:**

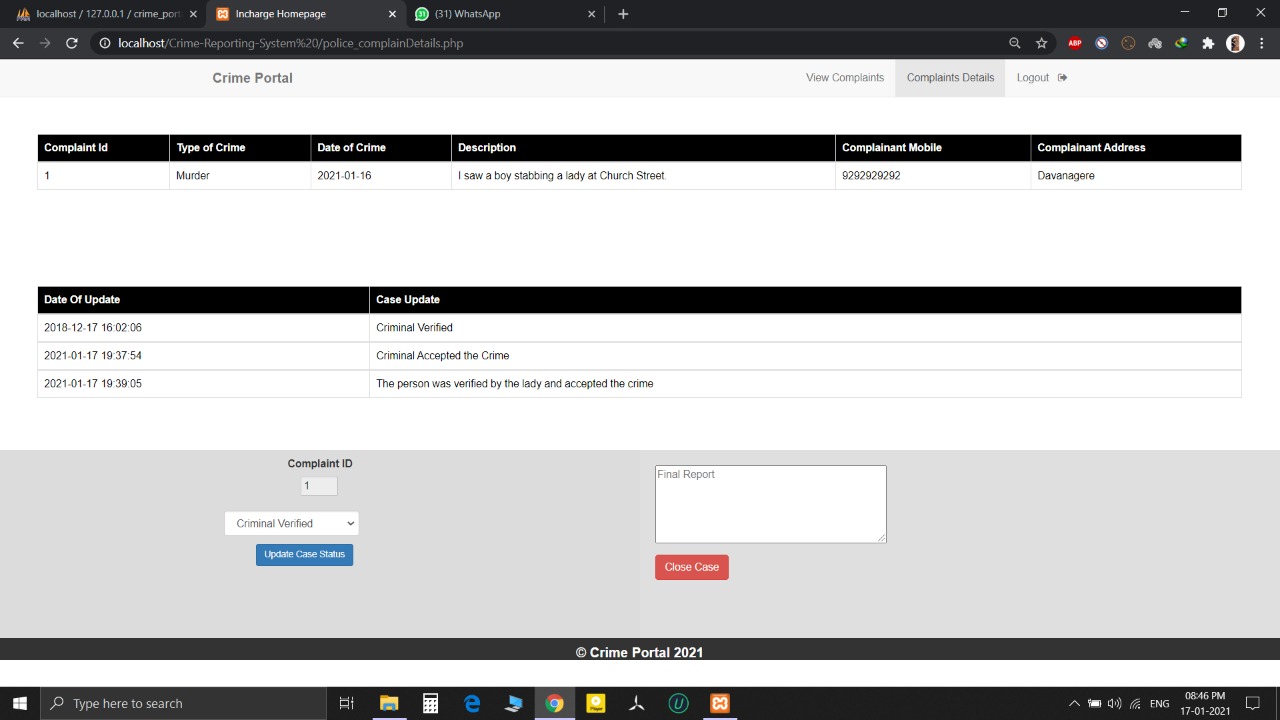


Image 5.8: Update Of The Crime.

**Complain Details:**

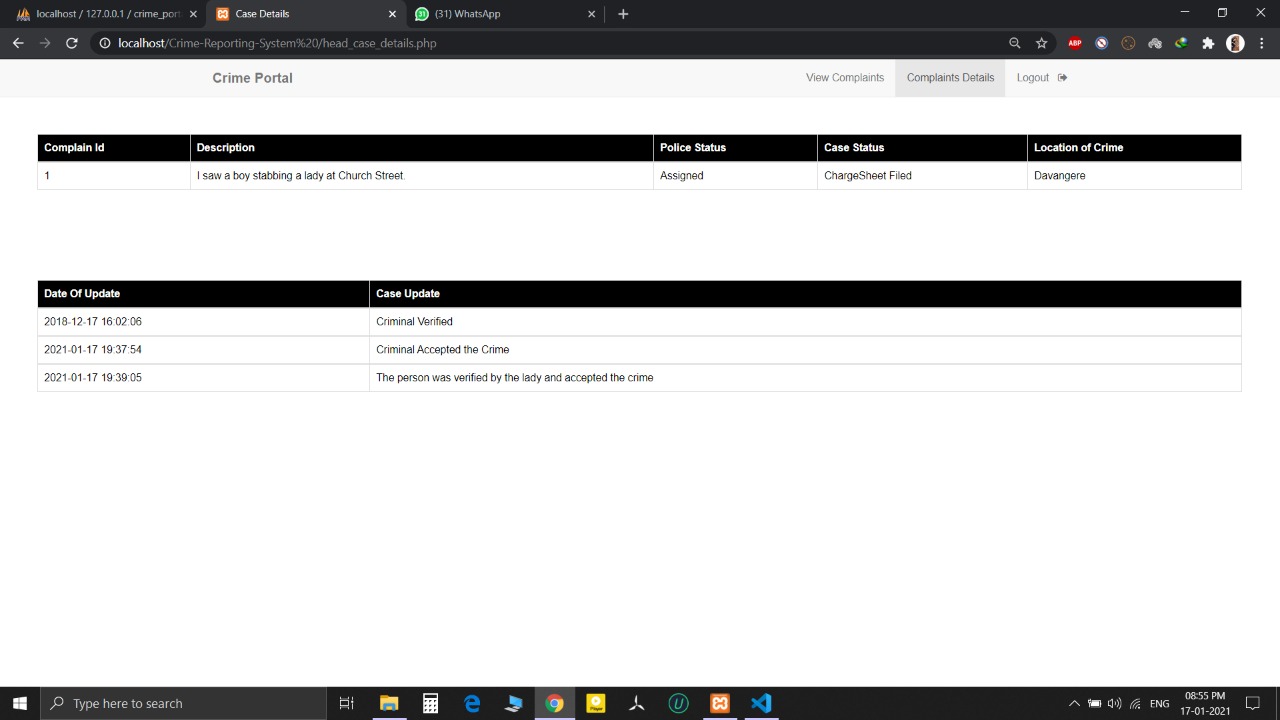
****

Image 5.9: Complain Details.

**CONCLUSION**

In this modern world, the use of computers and mobile phones is becoming rampant. As a result, the Crime Reporting system needs to embrace new technologies. This report has presented a simple, convenient, cost-effective, but efficient online crime Reporting System with a user-friendly, sensitive and intelligible web interface. Whereby it can be accessed at any time provided there is internet connection.

**FUTURE ENHANCEMENTS**

* We look forward to working with the Government in implementing

the recommendations and seeing an improvement in the effectiveness of the organization.

* The method of video conferencing can be added to make the project livelier.
* An intercom facility will add a little more flexible communication between the

Master and Branch in Charge, Master and Members etc.

**REFERENCE**

**BOOKS:**

* Fundamentals of database systems,Ramez Elmasri and S B Navathe, 7th Edition, 2017 , Pearson.
* Database management systems, Ramakrishnan, and Gehrke, 3rd Edition , 2014 , McGraw Hill.
* Coronel, Morris , and Rob,Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.
* Silberschatz Korth and Sudharshan, Database System Concepts , 6th Edition ,

McGraw Hill, 2013.

**LIST OF WEBSITES:**

* <https://en.wikipedia.org/wiki/Database/>
* <https://www.w3schools.com/>