# ONLINE FIRST INFORMATION REPORT FILING SYSTEM

**An industry oriented project submitted to Jawaharlal Nehru Technological University, Kakinada in partial fulfillment of requirements for the award of Degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

| | |
|---|---|
| **Y.SNEHITHA** | **N.MOUNIKA** |
| **16AJ1A05B1** | **16AJ1A0578** |
| **M.KARUNA** | **S.SUNILPRASAD** |
| **16AJ1A0571** | **16AJ1A0594** |

**Under the supervision of**

**Mrs. A.DURGA BHAVANI, M.Tech**

**Assistant Professor**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**AMRITA SAI INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(AUTONOMOUS)**

Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada
ISO 9001:2015 Certified Institution, Accredited by NAAC with ‘A’ grade
Recognized by UGC under 2(f) and 12(B) of 1956 Act
Amrita Sai Nagar, Paritala, Krishna District, Andhra Pradesh-521180
**www.amritasai.org.in,0866-2428399**

**2016-2020**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Amrita Sai Institute of Science and Technology**
Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada
ISO 9001:2015 Certified Institution, Accredited by NAAC with ‘A’ grade
Recognized by UGC under 2(f) and 12(B) of 1956 Act
Amrita Sai Nagar, Paritala, Krishna District, Andhra Pradesh-521180
**www.amritasai.org.in,0866-2428399**

# CERTIFICATE

This is to Certify that, the project work entitled **“ONLINE FIRST INFORMATION REPORT FILING SYSTEM”** done by **Y.SNEHITHA (16AJ1A05B1), N.MOUNIKA (16AJ1A0578), M.KARUNA (16AJ1A0571), S.SUNILPRASAD (16AJ1A0594)**, in partial fulfillment of requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** to **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY-KAKINADA, KAKINADA** is a record of bonafied work carried out under my guidance supervision.

The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

**Project Guide** **Head of the Department**

**EXTERNAL EXAMINER**

# DECLARATION

The project entitled **"ONLINE FIRST INFORMATION REPORT FILING SYSTEM",** which is submitted by me in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** at **AMRITA SAI INSTITUTE OF SCIENCE AND TECHNOLOGY**, **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY-KAKINADA, KAKINADA** comprises only my original work and due acknowledgement has been made in the text to all other material used.

We also declare that this project work is the result of our sincere efforts and that it has not been submitted to any other university for the award of the degree or any diploma

**Project Associates**

**Y.SNEHITHA (16AJ1A05B1)**
**N.MOUNIKA (16AJ1A0578)**
**M.KARUNA (16AJ1A0571)**
**S.SUNILPRASAD (16AJ1A0594)**

# ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without greeting those who made it possible and whose guidance and encouragement made the effort take a success.

We extend our heartfelt gratitude to the almighty for giving us strength in proceeding with this project work titled "**online first information report filing system**".

We wish to take this opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during our project work. It is our pleasure and responsibility to acknowledge the help of all those individuals.

We express our deepest gratitude to The Management of Amrita Sai Institute of Science And Technology, Paritala for their support and encouragement in completing our project work and providing us necessary facilities.

We express our deepest gratitude to **Dr. K.Sai Manoj, CEO, Dr. M.Sasidhar, principal and Dr. P.chiranjeevi, Director of Academics** Amrita Sai Institute of Science and Technology, Paritala for their valuable suggestions during preparation of draft this document.

We are grateful to Head of the Department and our Supervisor, Department of Computer Science and Engineering for his assistance and encouragement in all respects in carrying throughout our project work.

We sincerely thank all the faculty members and staff of the Department of Computer Science and Engineering for their valuable advices, suggestions, and constant encouragement which played a vital role in carrying out this project work.

Finally, we thank one and all who directly or indirectly helped us for successful completion of our project work.

With Sincerely,
Y.Snehitha (16AJ1A05B1)
N.Mounika (16AJ1A0578)
M. Karuna (16AJ1A0571)
S.Sunilprasad (16AJ1A0594)

# ABSTRACT

The aim of this project and implementation is to develop interactive online first information report system which is easily accessible to the public. Crime is a part of illegal activities in human life. The rate of crimes is increasing day by day in all societies across the world, but we do believe that there is a lot which can be done by both the governments and the individuals to reduce the crimes in communities. There are many current crime management systems which faces several difficulties, as there is no means to report crime instantly other than phone calls, messaging or face-to-face compliant filing. Project has proposed an online FIR system which allows the user to file complaints or missing reports and keep a track of it.The back end and front end of the crime reporting system is done using: HTML, CSS, PHP, MY SQL the system is designed.

## INDEX

# 1. INTRODUCTION

This synopsys documents the process of designing, building and testing this web based application system to be used for marketing. The “Online First Information Report Filing System” is a web based application .This application provides facility for reporting, crimes, complaints, show most wanted person details online. Any number of clients can connect to the server. Each user first makes their login to server to show their availability. There are 4 types of users in our system.

1. Super administrator. This post is allotted for higher authority of ministry.
2. It is allocated for officer in charge of any police station.
3. It is allocated for chief justice of any district court.
4. Final one is allotted for civilian to acknowledge them about crime activities.

The duty of the super administrator is to authenticate police station and courts. He can also search for any information regarding this system.2nd user i.e. The police stations can maintain the local area’s citizen information database and he can generate fir gd.3rd user is the court it can also update the final decision of court and the 4th user i.e. Citizen able to search for criminal activities. In present manual system the total process of criminal report generate is not so fast and friendly for the officials to solve this kind of problem we have proposed for better and automated system i.e. ONLINE FIRST INFORMATION REPORT FILING SYSTEM(OFIR). The server can be any web server. An SMTP server must be maintain for temporary storage of emails; the application produced by this project exceeded expectations and would be suitable to be used in industry.

## 1.1 INTRODUCTION ABOUT THE PROJECT

The project titled as “Online First Information Report Filing System “is a web based application. This software provides facility for reporting online crimes, complaints, missing persons, show most wanted person detail. Any Number of clients can connect to the server. Each user first makes their login to sever to show their availability.

Modules

- Users Registration
- Administrator

- Team

The registered user module includes

- Login, Registration and Forgot Password.
- Add Complaint: This module help the user to report online complaints.
- Add Crime report: This module help the user to report online crimes.
- Add Missing person: This module help the user to report online missing persons details also we have to add photos of missing person using heterogeneous database.
- View Missing persons: This module help the user to view all the missing person details.
- View Mostwanted persons: This module help the user to view all mostwanted persons which can be given by the administrator.
- Edit Complaint: This module help the user to edit his complaint details.
- Edit Account: This module help the user to update his or her profile.
- View complaint status: This module allows us to view the status of all complaint that you have posted earlier.
- View crime status: This module allows us to view the status of the all crimes that you have posted earlier.
- Add and View feedback: This module helps the user to add and view feedbacks
- Mail: This module helps the user to send mail to the administrator.
- Upload and Download Evidence.

The administrator module includes

- View and delete user’s feedback: This module helps the admin to add and delete user’s feedback
- Admin can create investigating teams. And view Teams.
- Admin can see all the crime record registered by users.
- Admin can allot word to teams for investigate. And review time to time.
- Admin can registered FIR and View it.
- Add and view Criminal registration: This module helps the amin to add and view criminal registrations
- Add and view history sheet: This module helps the admin to add and view history reports

- View and delete all complaint reply: This module helps the admin to view and delete complaint reply.
- Add and view prisoner report: This module helps the admin to add and prisoner reports
- Change password: This module helps the amin to update his or her password
- Chat: This module help the admin to chat with the administrator or with other registered users

The Team module includes

- Team can login their account by team name and team id.
- Team can see their allotted word and start investigation.
- Team can upload and download evidences.

## 1.2 PURPOSE OF THE SYSTEM

This web based publication is for automation in managing the criminal record. The four types of users in our system will monitor this word processor. 1st one is super administrator. This post is allotted for higher authority of ministry. 2nd one is allotted for officer incharge of any police station. And the 3rd one is allotted for chief justice of any district court. And the final one is allotted for civilian to acknowledge them about criminal activities. In present manual system the total process of criminal report generation is not so fast and friendly for the officials. To solve this kind of problems we have proposed for better and automated system i.e. ONLINE FIRST INFORMATION REPORT FILING SYSTEM (OFIR). In present manual system there are three types of terms 1st one is GD (general dairy). 2nd one is FIR (first information report). 3rd one is charge sheet. GD taken when for stored preliminary information about any crime. FIR is launched when murder, rape, vandalism like crime is happened. And the final charge sheet is produced after investigation and submitted in court but in our system to be proposed to be take GD for any kind of preliminary criminal activities. And FIR will be launched after investigation and it will directly submit to court to complete the rest of the process. The server can be any web server. An SMTP server must be maintained for temporary storage of emails;
The application produced by this project exceeded expections and would be suitable to the used in industry.

# 2. SYSTEM ANALYSIS

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Here in the Email to Fax server project, a detailed study of existing system is carried along with all the steps in system analysis. An idea for creating a better project was carried and the next steps were followed.

## 2.1 EXISTING SYSTEM

In the existing system only we can see the details of particular information about the police stations in our state, the existing system has more workload for the authorized person, but in the case of Proposed System, the user can registered in our site and send the crime report and complaint about a particular city or person.

### 2.1.1 Drawbacks of Existing System

- More manpower.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials.
- Damage of machines due to lack of attention.

To avoid all these limitations and make the working more accurately the system needs to be computerized.

## 2.2 PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The proposed system helps the user to work user friendly and he can easily do his jobs without time lagging.

### 2.2.1 Expected Advantages of Proposed System

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

- Ensure data accuracy’s.
- Proper control of the higher officials.

- Reduce the damages of the machines.
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required.

## 2.3 SOFTWARE REQUIREMENT SPECIFICATION

| | | |
|---|---|---|
| Operating System | : | Windows 2000/xp |
| Languages | : | PHP |
| Front End | : | HTML, JavaScript |
| Platform | : | Xammp |
| Web Servers | : | Localhost |
| Backend | : | My SQL |
| Browser Program | : | Internet explorer/Mozilla Fireworks |

# 3. SYSTEM DESIGN

## 3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

## 3.2 TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

### 3.3 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

### 3.4 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

## 4. SOFTWARE REQUIREMENT SPECIFICATION

### 4.1 FUNCTIONAL REQUIREMENT

Requirement no 1: Register for the criminal or prisoner

- Input: Criminal personal details via-Criminal No, Criminal Name, Age, Occupation, Crime Type, Address, Most Wanted.

- Prisoner details: Prisoner No, Change Sheet No, Nickname, Crime Type, Family Member, Identity Mark, Height, Weight, Color.

- Processing: Validation checks are performed on the input data by the user/administrator like

  1. Mandatory fields should not be kept blank.
  2. Prisoner family member should be check.
  3. All the entries in database should be correct.

Requirement no 2: FIR management

- Input: Fir no, COMPLAINT NO, FIR Data, FIR Time, Information Type, Place of crime, Act for crime, District of Fir, Address of victim, Passport No, Name of police, Received Time, Information received.
- Processing: After entering the required information the various validation checks are performed on submission of the information, the Updation are made to the database.
  1. Administrator must enter the correct user name and password.
- Output: If the specified database is stored then the whole database is display in the tabular form the data displayed is the updated and current information of the user personal and academic details. It also displays the status of the prisoner and criminals.

### 4.2 NON FUNCTIONAL REQUIREMENT

- Secure access of confidential data.SSL can be used.
- 24*7 availability
- Better component design to get better performance at peak time.
- Flexible service based architecture will be highly desirable for future extension.

# 5. DESIGN

## 5.1 INTRODUCTION TO UML

UML is a method for describing the system architecture in detail using the blue print. UML represents a collection of best engineering practice that has proven successful in the modeling of large and complex systems. The UML is very important parts of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the helps UML helps project teams communicate explore potential designs and validate the architectural design of the software.

## 5.2 USE CASE DIAGRAM

Use case diagram represents the functionality of the system. Use case focus on the behavior of the system from external point of view. Actors are external entities that interact with the system.

**Use cases**

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

**Actors**

An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

**System boundary boxes** (optional)

A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not.

Four relationships among use cases are used often in practice.

**Include**

In one form of interaction, a given use case may include another. "Include is a Directed Relationship between two use cases, implying that the behavior of the included use case is inserted into the behavior of the including use case.

The first use case often depends on the outcome of the included use case. This is useful for extracting truly common behaviors from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "«include»".There are no parameters or return values. To specify the location in a flow of events in which the base use case includes the behavior of another, you simply write include followed by the name of use case you want to include, as in the following flow for track order.

**Extend**

In another form of interaction, a given use case (the extension) may extend another. This relationship indicates that the behavior of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label "«extend»". Modelers use the «extend» relationship to indicate use cases that are "optional" to the base use case.

**Generalization**

In the third form of relationship among use cases, a generalization/ specialization relationship exists. A given use case may have common behaviors, requirements, constraints, and assumptions with a more general use case. In this case, describe them once, and deal with it in the same way, describing any differences in the specialized cases. The notation is a solid line ending in a hollow triangle drawn from the specialized to the more general use case (following the standard generalization notation

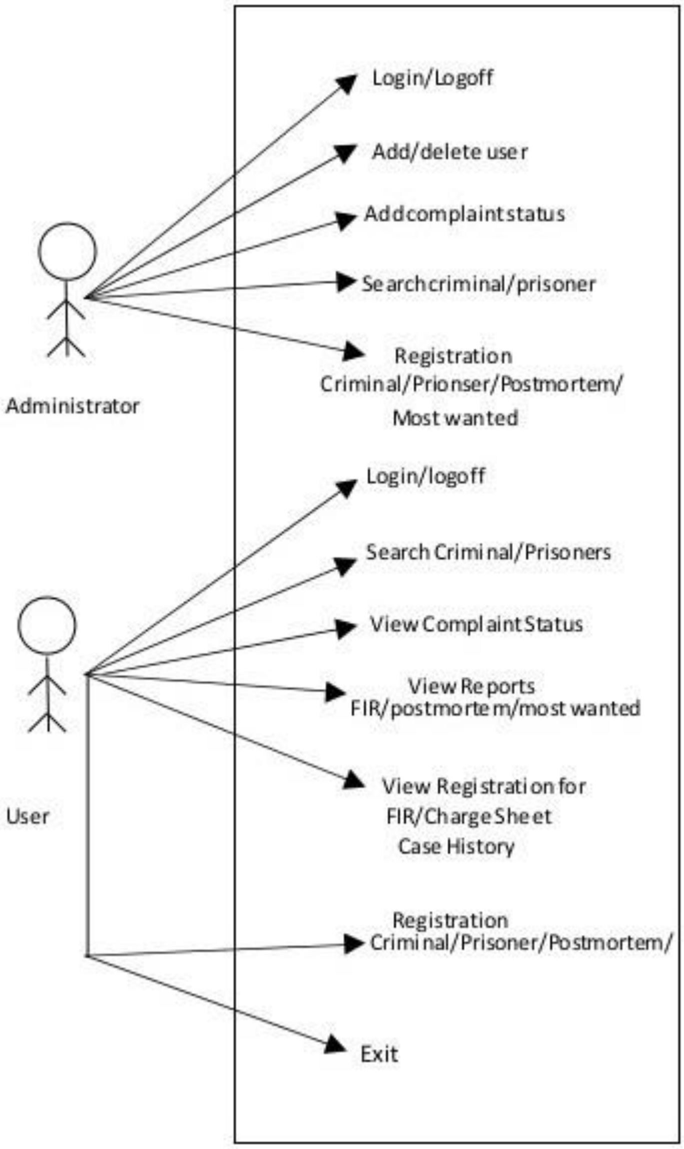
**Associations**

Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case.

**Identified Use Cases**

The "user model view"encompasses a problem and solution from the preservative of those individuals whose problem the solution addresses.

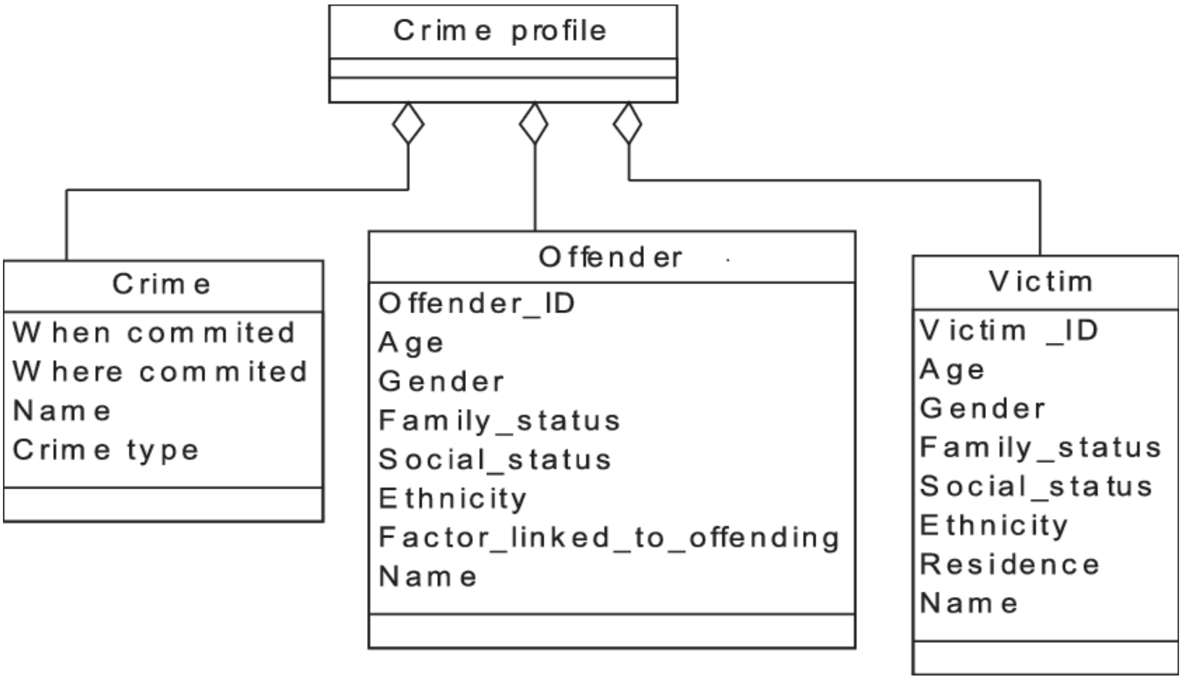
The view presents the goals and objectives of the problem owners and their requirements of the solution. This   view is composed of “use case diagrams”. These diagrams describe the functionality provided by a system to external integrators.  These diagrams contain actors, use cases, and their relationships.



### 5.3 Class Diagram

Class-based Modeling, or more commonly class-orientation, refers to the style of object-oriented programming in which inheritance is achieved by defining classes of objects; as opposed to the objects themselves (compare Prototype-based programming).

The most popular and developed model of OOP is a class-based model, as opposed to an object-based model. In this model, objects are entities that combine state (i.e., data), behavior (i.e., procedures, or methods) and identity (unique existence among all other objects). The structure and behavior of an object are defined by a class, which is a definition, or blueprint, of all objects of a specific type. An object must be explicitly created based on a class and an object thus created is considered to be an instance of that class. An object is similar to a structure, with the addition of method pointers, member access control, and an implicit data member which locates instances of the class (i.e. actual objects of that class) in the class hierarchy (essential for runtime inheritance features).



## 5.4 Sequence Diagram

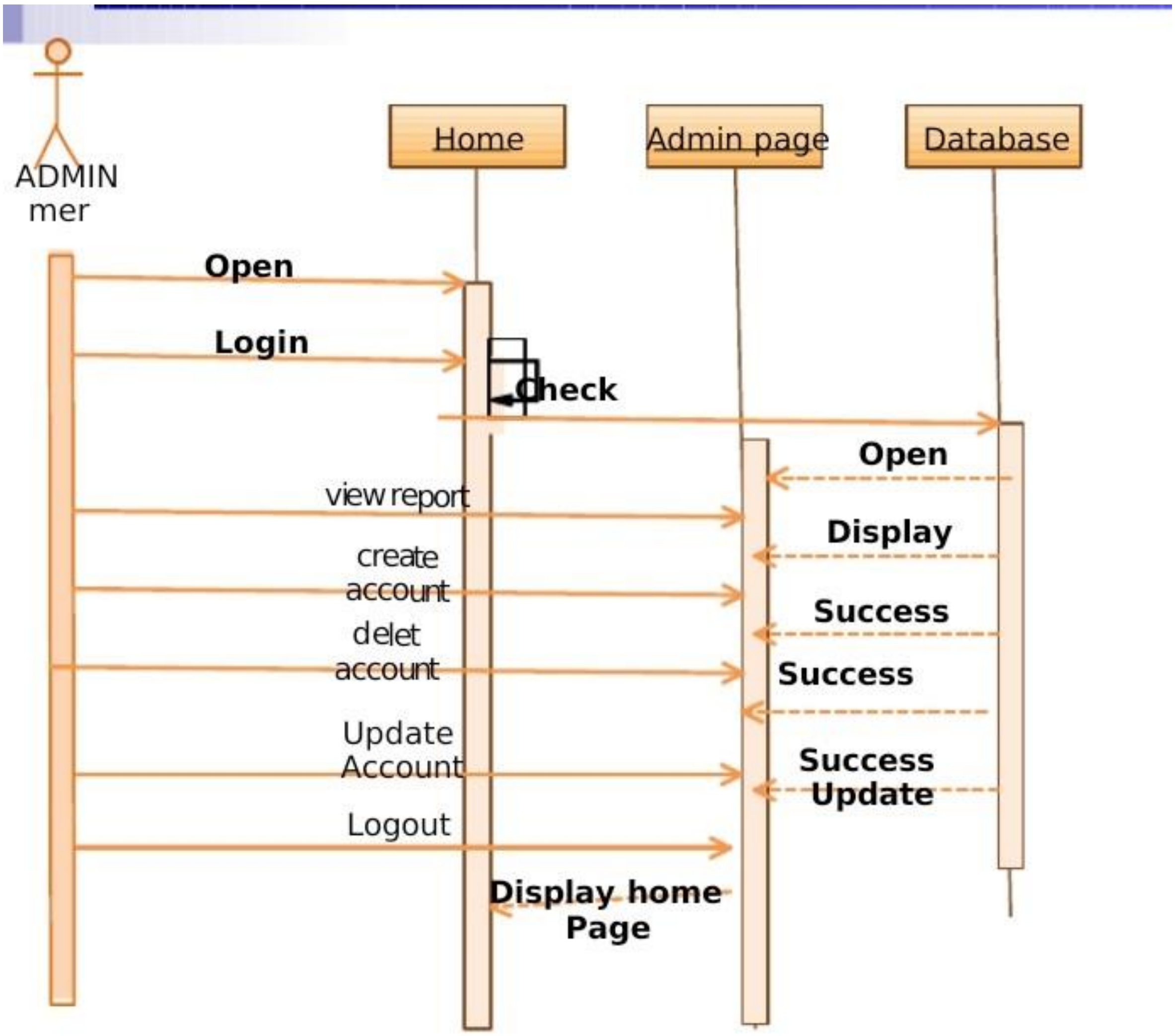
**A sequence diagram** in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. If the lifeline is that of an object, it demonstrates a role. Note that leaving the instance name blank can represent anonymous and unnamed instances. In order to display interaction, messages are used. These are horizontal arrows with the message name written above them. Solid arrows with full heads are synchronous calls, solid arrows with stick heads are asynchronous calls and dashed arrows with stick heads are return messages. This definition is true as of UML 2, considerably different from UML 1.x.

Activation boxes, or method-call boxes, are opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message (Execution Specifications in UML).

Objects calling methods on themselves use messages and add new activation boxes on top of any others to indicate a further level of processing. When an object is destroyed (removed from memory), an X is drawn on top of the lifeline, and the dashed line ceases to be drawn below it (this is not the case in the first example though). It should be the result of a message, either from the object itself, or another.

A message sent from outside the diagram can be represented by a message originating from a filled-in circle (found message in UML) or from a border of sequence diagram (gate in UML)



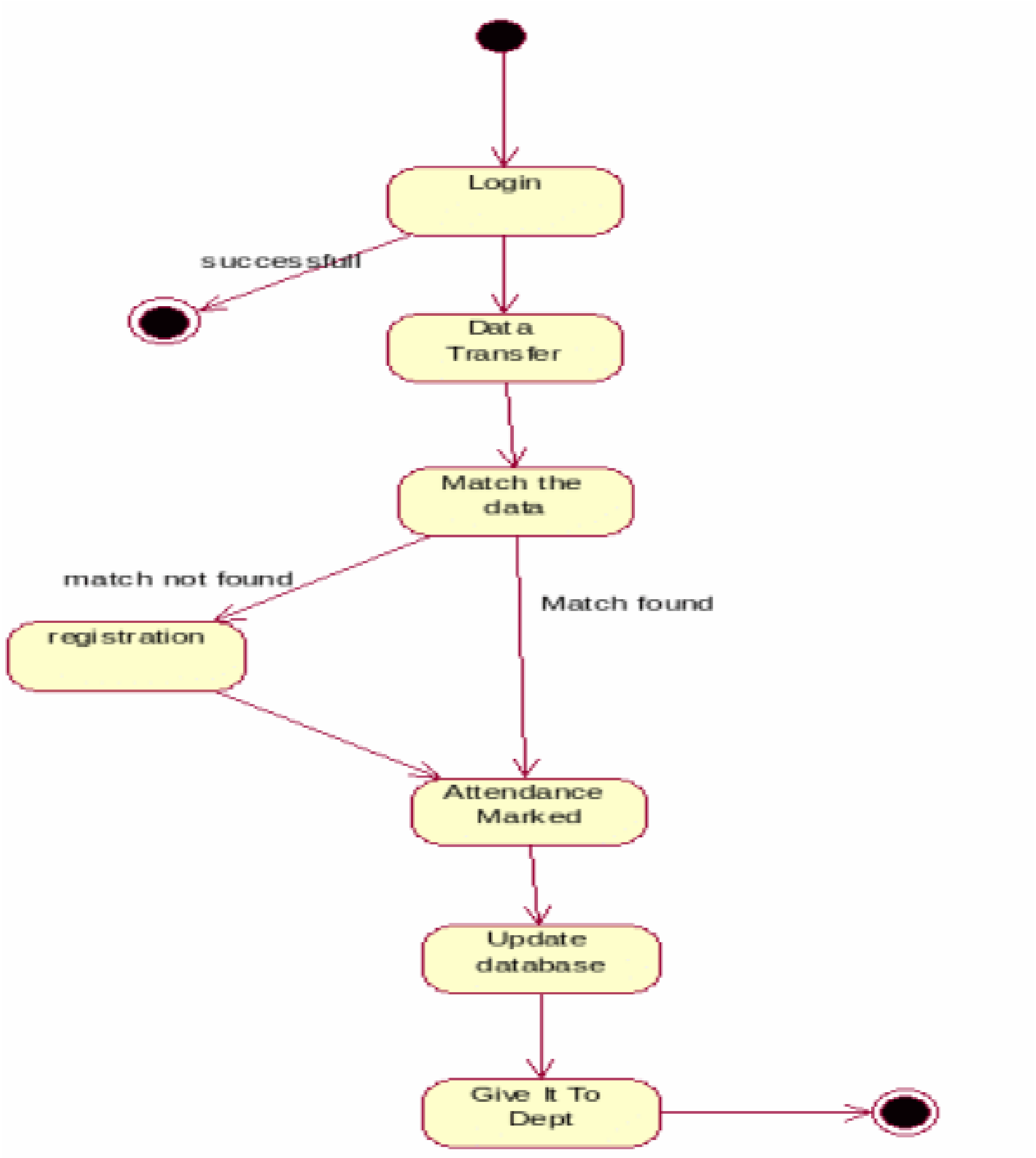
## 5.5 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Activity diagrams are constructed from a limited repertoire of shapes, connected with arrows. The most important shape types:

- rounded rectangles represent activities;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial state) of the workflow;
- An encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with the decisions or loops.

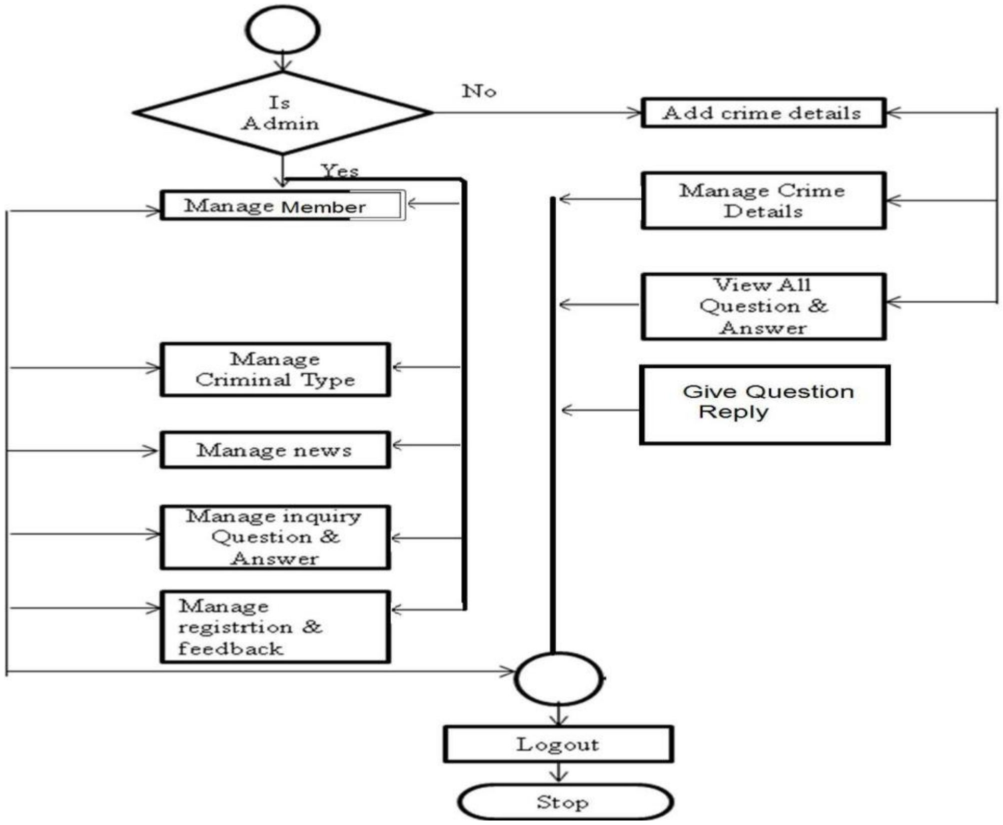


### 5.6 State Chart Diagram

Objects have behaviors and states. The state of an object depends on its current activity or condition. A state chart diagram shows the possible states of the object and the transitions that cause a change in state. A state diagram, also called a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language. A state diagram resembles a flowchart in which the initial state is represented by a large black dot and subsequent states are portrayed as boxes with rounded corners.

**State Chart Diagram**

There may be one or two horizontal lines through a box, dividing it into stacked sections. In that case, the upper section contains the name of the state, the middle section (if any) contains the state variables and the lower section contains the actions performed in that state. If there are no horizontal lines through a box, only the name of the state is written inside it. External straight lines, each with an arrow at one end, connect various pairs of boxes. These lines define the transitions between states. The final state is portrayed as a large black dot with a circle around it. Historical states are denoted as circles with the letter H inside.



## 5.7 Component Diagram

## COMPONENT LEVEL CLASS DESIGN

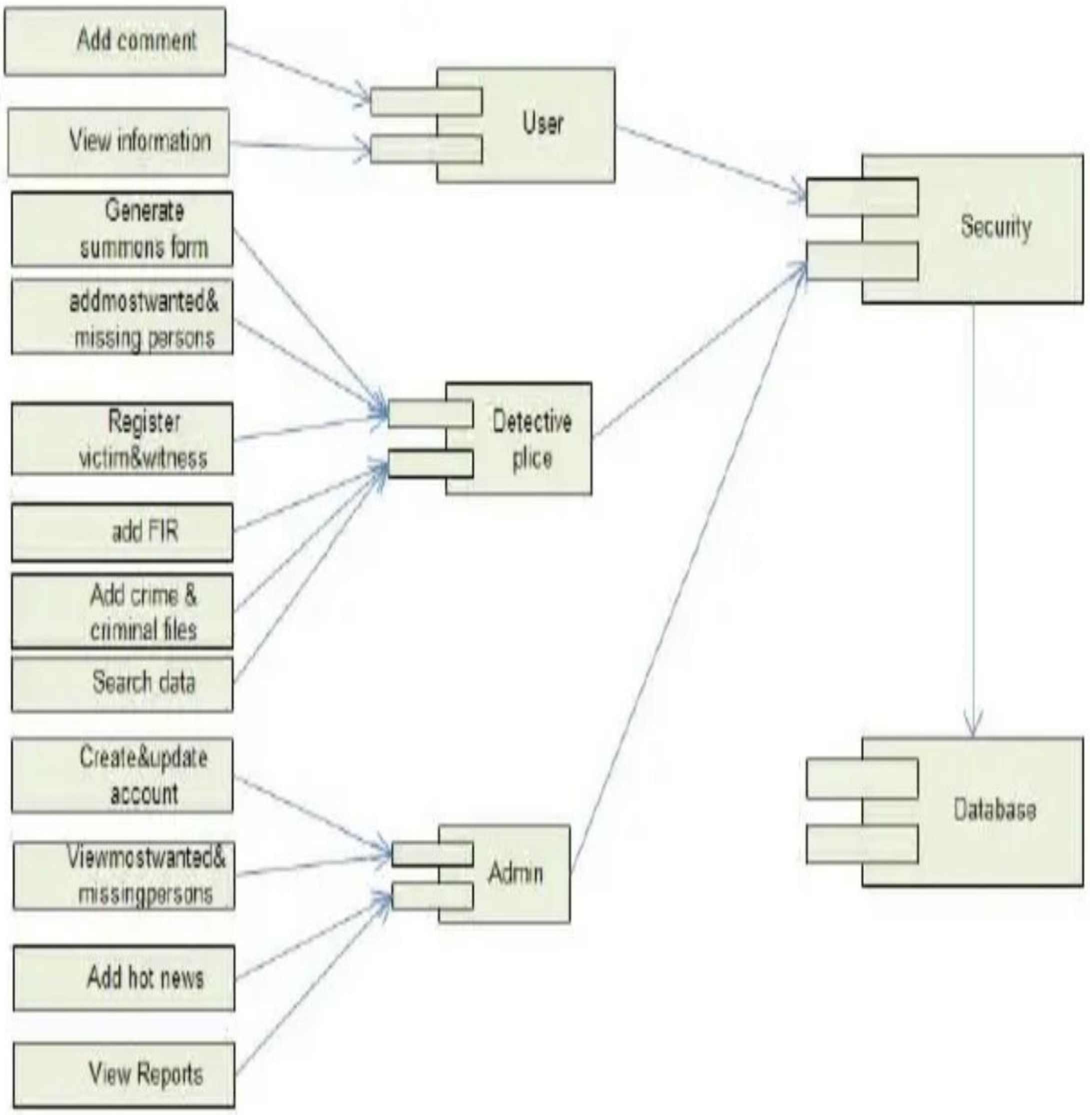
This chapter discusses the portion of the software development process where the design is elaborated and the individual data elements and operations are designed in detail. First, different views of a “component” are introduced. Guidelines for the design of object-oriented and traditional (conventional) program components are presented.

### What is a Component?

This section defines the term component and discusses the differences between object oriented, traditional, and process related views of component level design. Object Management Group OMG UML defines a component as “… a modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces.”

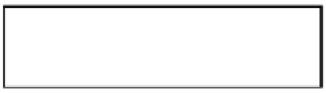
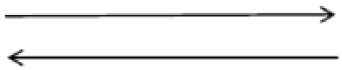
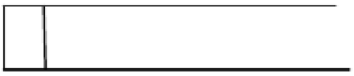
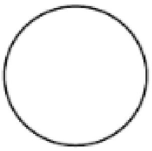
### An Object Oriented View

A component contains a set of collaborating classes. Each class within a component has been fully elaborated to include all attributes and operations that are relevant to its implementation. As part of the design elaboration, all interfaces (messages) that enable the classes to communicate and collaborate with other design classes must also be defined. To accomplish this, the designer begins with the analysis model and elaborates analysis classes (for components that relate to the problem domain) and infrastructure classes (or components that provide support services for the problem domain).

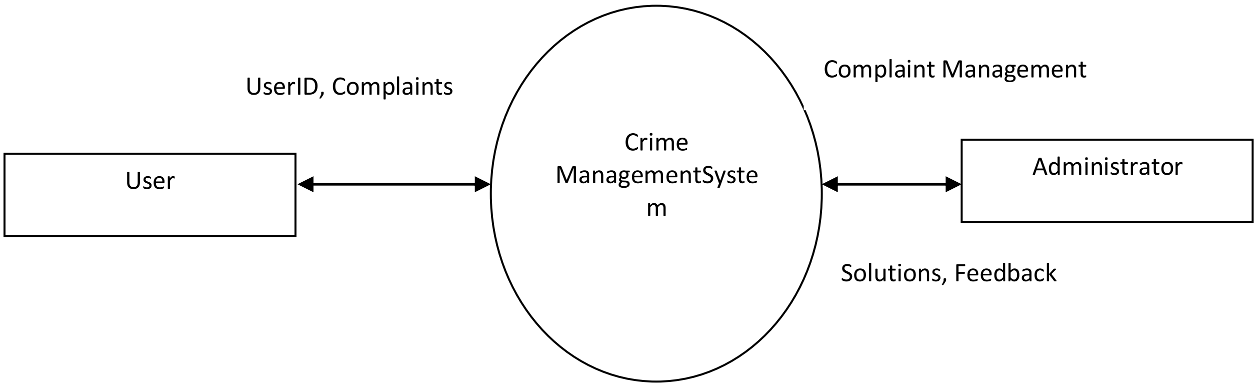


### 5.8 Data Flow Diagram

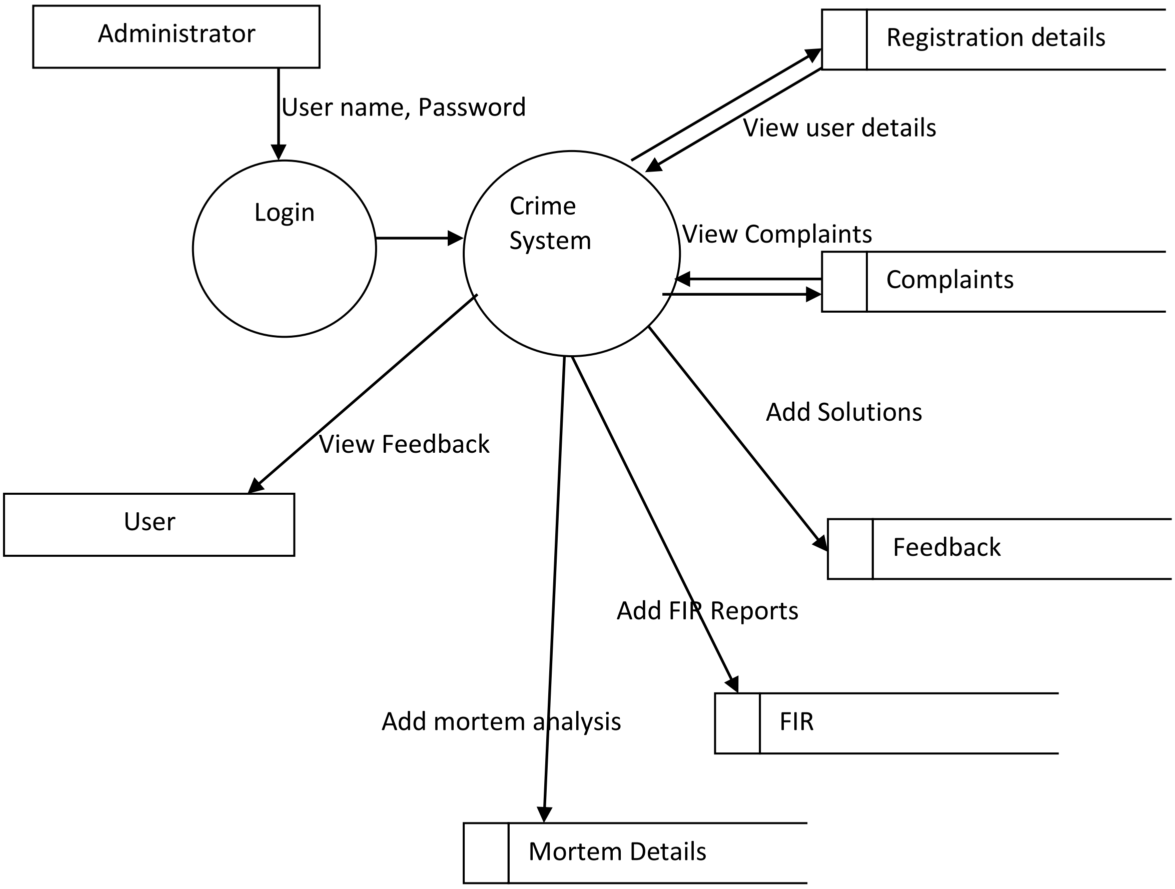
**DFD NOTATIONS**

| NAME | NOTATION | DESCRIPTION |
|---|---|---|
| EXTERNAL ENTITY |  | Represents the source or destination of data within the system. |
| DATAFLOW |  | Represents the movement of data from its source to destination within the system. |
| DATA STORE |  | Indicate the place for storing information within the system. |
| PROCESS |  | Shows a transformation or manipulation of data within the system. Also known as bubble. |

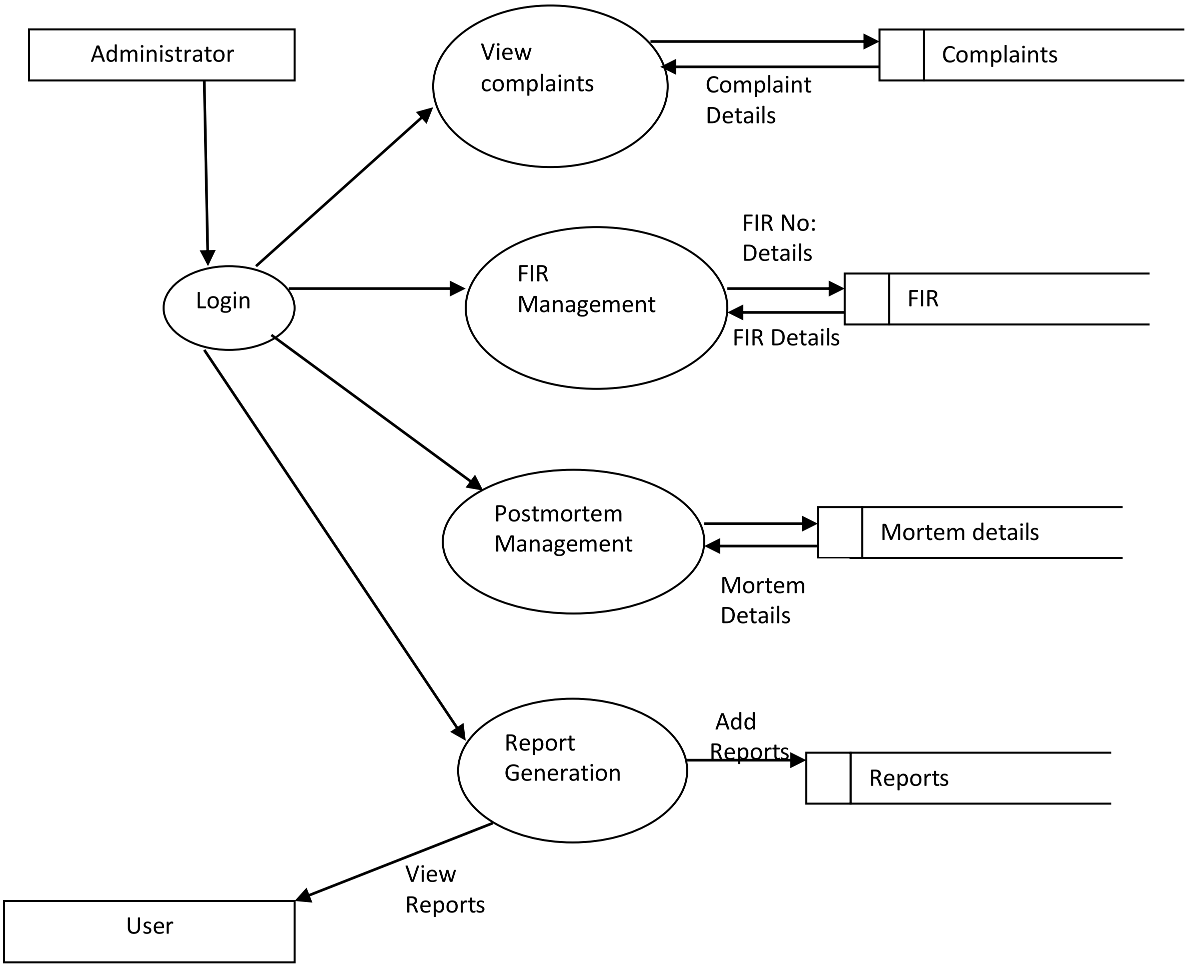
**Context flow diagram**



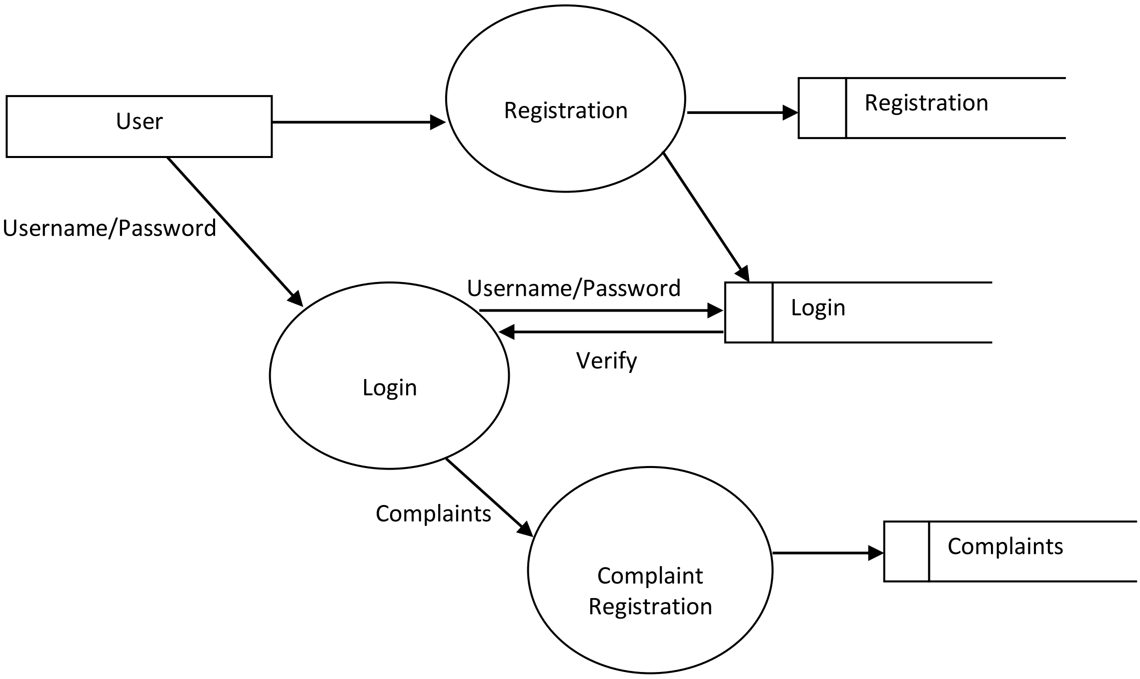
**Level 1 DFD- Administrator**


Administrator
User name, Password
Login
Crime
System
Registration details
View user details
View Complaints
Complaints
Add Solutions
View Feedback
User
Feedback
Add FIR Reports
FIR
Add mortem analysis
Mortem Details

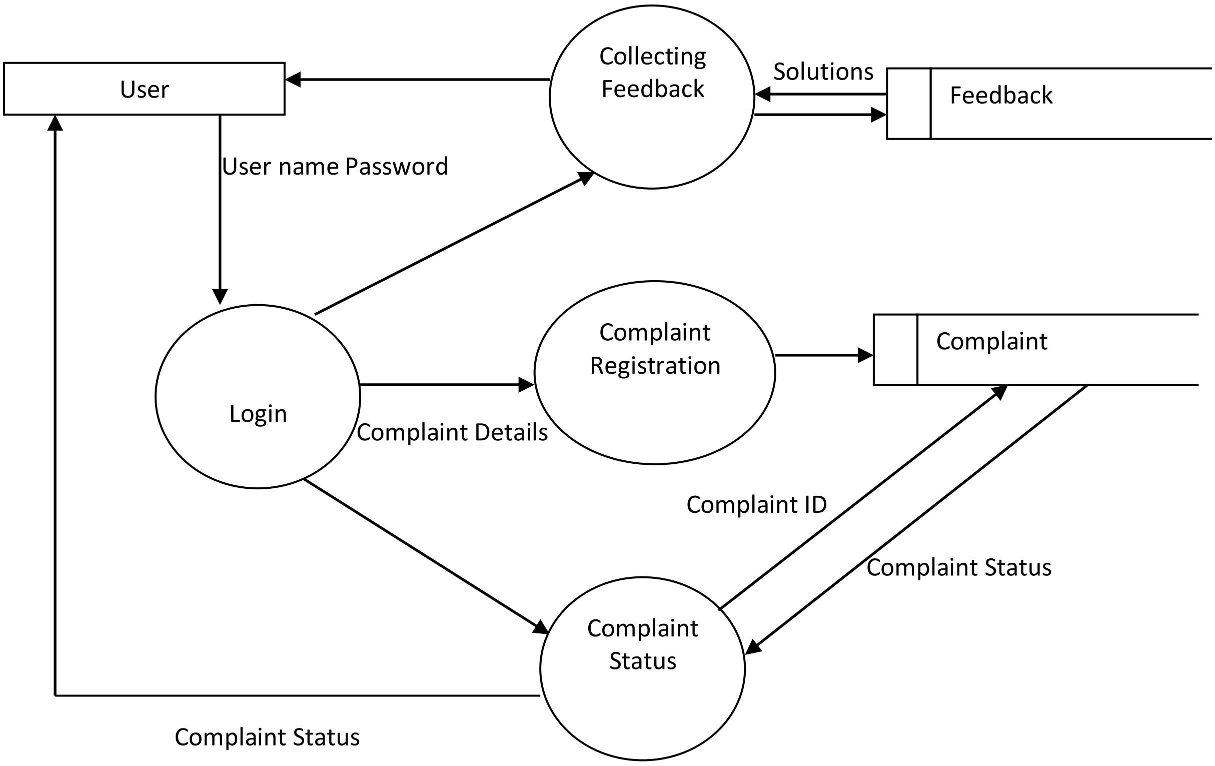
## Level 2 DFD- Administrator



## Level 1 DFD- User

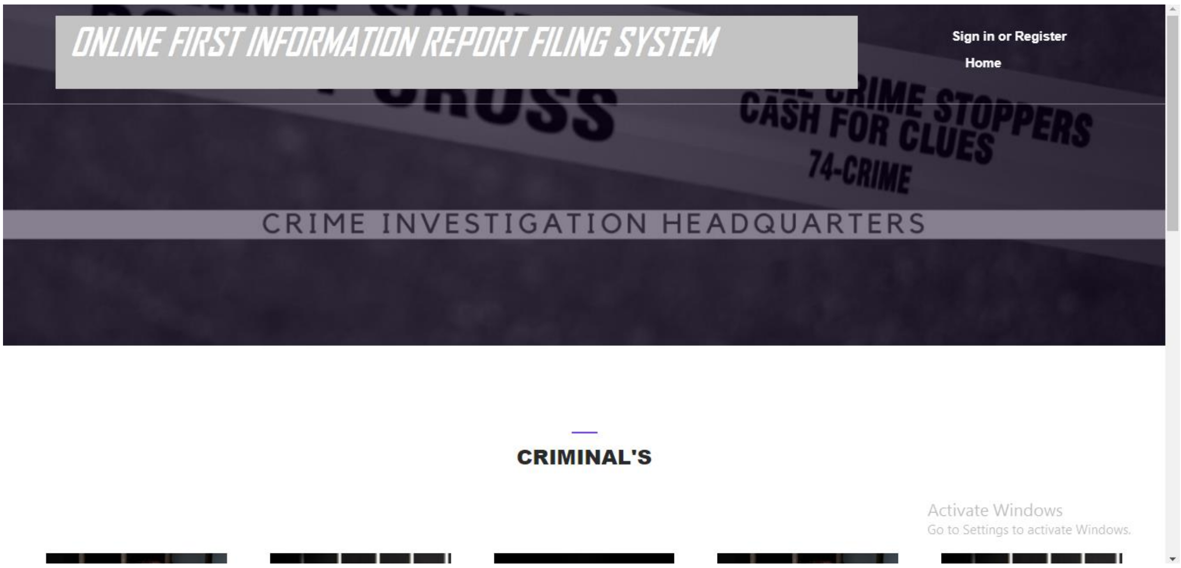


## Level 2 DFD- User

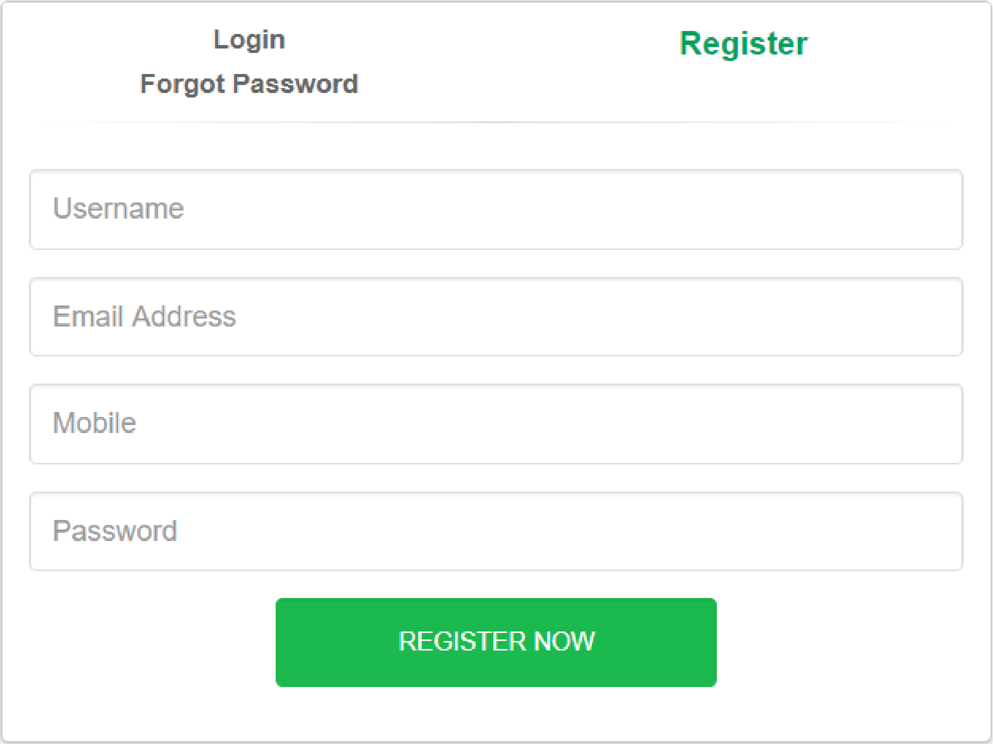


# 6. OUTPUT SCREENS

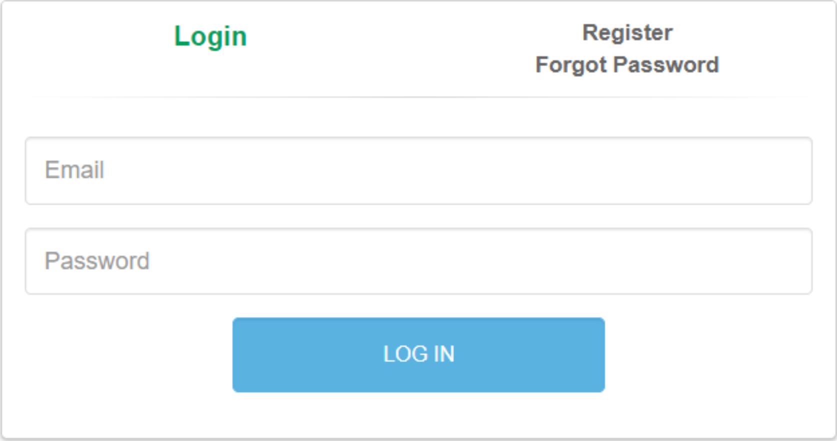
### User Homepage


ONLINE FIRST INFORMATION REPORT FILING SYSTEM
Sign in or Register
Home
CRIME STOPPERS
CASH FOR CLUES
74-CRIME
CRIME INVESTIGATION HEADQUARTERS
CRIMINAL'S
Activate Windows
Go to Settings to activate Windows.

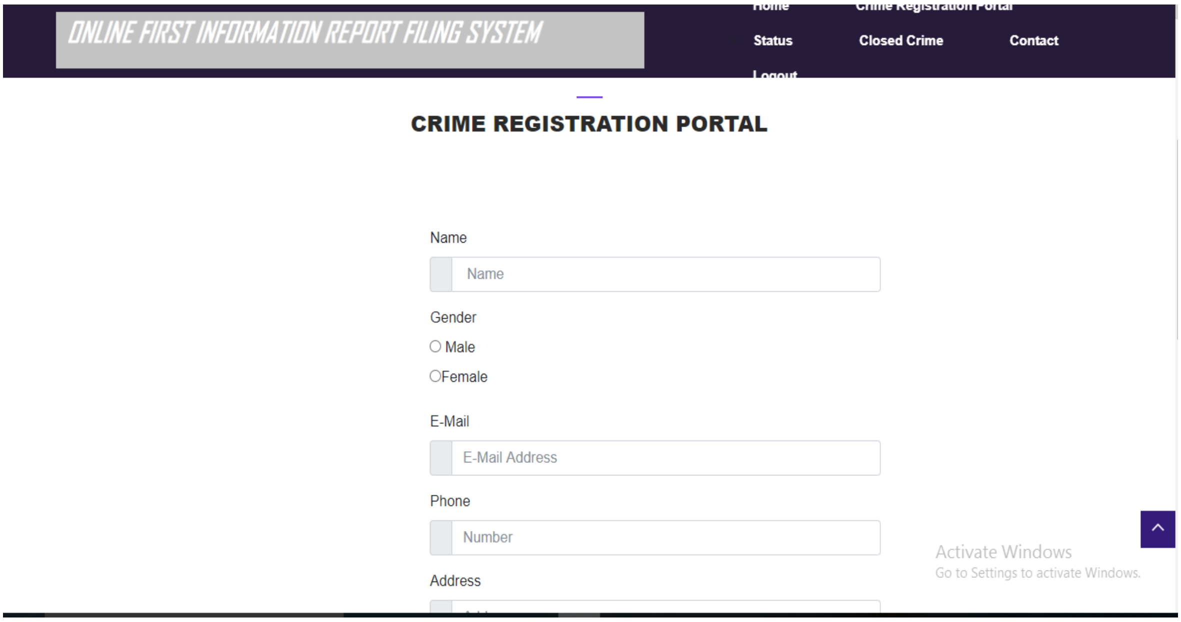
### User Register



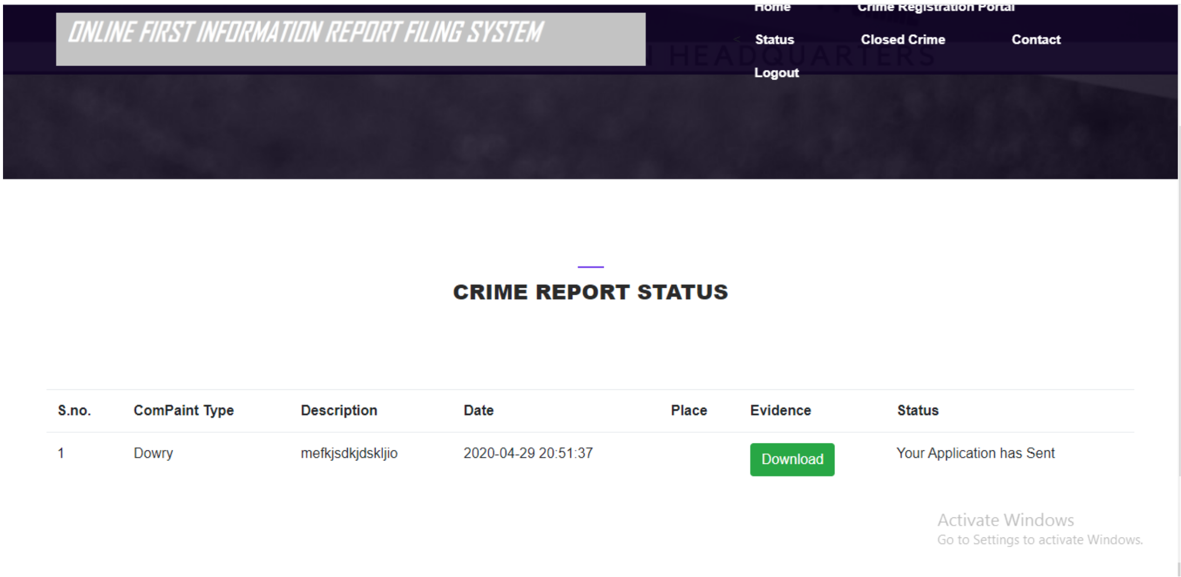
## Login



## Crime Registration


ONLINE FIRST INFORMATION REPORT FILING SYSTEM
Home
Crime Registration Portal
Status
Closed Crime
Contact
Logout
CRIME REGISTRATION PORTAL
Name
Name
Gender
○ Male
○Female
E-Mail
E-Mail Address
Phone
Number
Address
Activate Windows
Go to Settings to activate Windows.

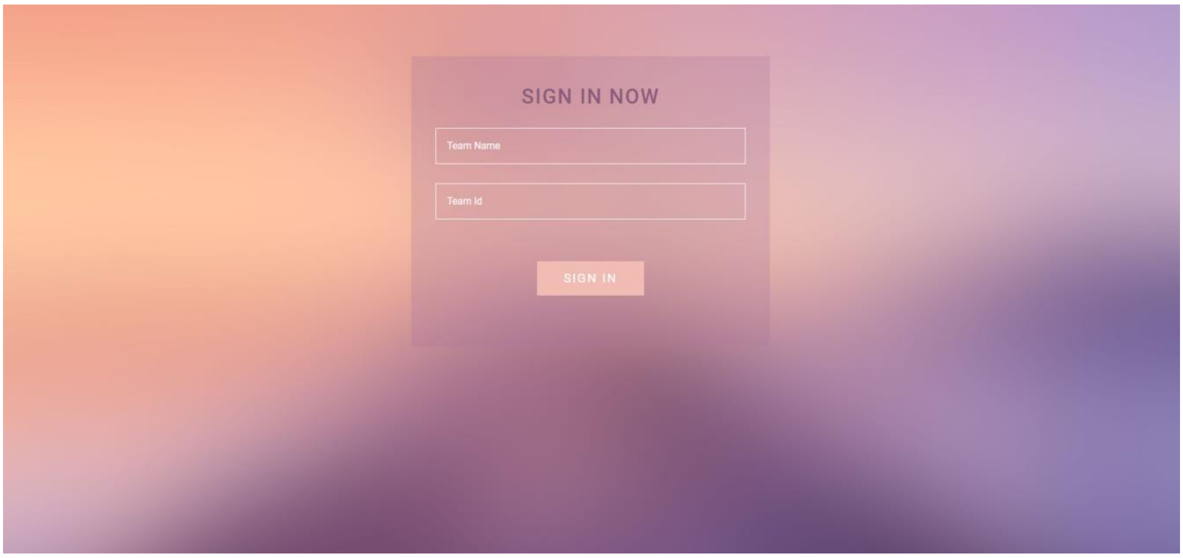
## Status


ONLINE FIRST INFORMATION REPORT FILING SYSTEM
Home
Crime Registration Portal
Status
Closed Crime
Contact
Logout
Activate Windows
Go to Settings to activate Windows.

CRIME REPORT STATUS

| S.no. | ComPaint Type | Description | Date | Place | Evidence | Status |
|---|---|---|---|---|---|---|
| 1 | Dowry | mefkjsdkjdskljio | 2020-04-29 20:51:37 | | Download | Your Application has Sent |

## Team Login



# 7. TECHNOLOGIES

## 7.1 INTRODUCTION TO PHP

PHP is one of the most widely used server side scripting language for web development. Popular websites like Facebook, Yahoo, Wikipedia etc., and our very own Study tonight, are developed using PHP.

PHP is so popular because it's very simple to learn, code and deploy on server, hence it has been the first choice for beginners since decades.

In this tutorial series we will be covering all the important concepts of Php language from basics to advanced and will also share some ready-to-use, useful code sinppets for beginners to kick start their web development project.

### What is PHP?

PHP stands for Hypertext Pre-Processor. PHP is a scripting language used to develop static and dynamic webpages and web applications. Here are a few important things you must know about PHP:

1. PHP is an Interpreted language, hence it doesn't need a compiler.
2. To run and execute PHP code, we need a Web server on which PHP must be installed.
3. PHP is a server side scripting language, which means that PHP is executed on the server and the result is sent to the browser in plain HTML.
4. PHP is open source and free.

### Prerequisite for starting with PHP

Although we will cover all the concepts related to PHP in this tutorial series, but you must have a basic understanding of HTML before starting with this tutorial.

You can learn HTML using our HTML Interactive Course, which is also available in Hindi Language.

### Is PHP the right language?

If you are still confused about whether you should learn PHP or is PHP the right language for your web project, then here we have listed down some of the features and use cases of PHP language, which will help you understand how simple yet powerful PHP scripting language is and why you should learn it.

1. PHP is open source and free, hence you can freely download, install and start developing using it.

2. PHP has a very simple and easy to understand syntax, hence the learning curve is smaller as compared to other scripting languages like JSP, ASP etc.
3. PHP is cross platform, hence you can easily develop and move/deploy your PHP code/project to almost all the major operating systems like Windows, Linux, and Mac OSX etc.
4. All the popular web hosting services support PHP. Also the web hosting plans for PHP are generally the amongst the cheapest plans because of its popularity.
5. Popular Content Management Systems like Joomla, Drupal etc. are developed in PHP and if you want to start your own website like Study tonight, you can easily do that with PHP.
6. With PHP, you can create static and dynamic webpages, perform file handling operations, send emails, access and modify browser cookies, and almost everything else that you might want to implement in your web project.
7. PHP is fast as compared to other scripting languages like JSP and ASP.
8. PHP has in-built support for MySQL, which is one of the most widely used Database management system.

These are some of the main features of PHP, while as you will learn the language you will realize that apart from these features,

**Uses of PHP**

To further fortify your trust in PHP, here are a few applications of this amazing scripting language:

1. It can be used to create Web applications like Social Networks (Facebook, Dig), Blogs (Word press, Joomla), ecommerce websites (Open Cart, Magneto etc.) etc.
2. Comman Line Scripting. You can write PHP scripts to perform different operations on any machine, all you need is a PHP parser for this.
3. Create Facebook applications and easily integrate Facebook plugins in your website, using Facebook's PHP SDK. Check this link for more information.
4. Sending Emails or building email applications because PHP provides with a robust email sending function.

5. Wordpress is one of the most used blogging (CMS) platform in the World, and if you know PHP, you can try a hand in Wordpress plugin development.

**Install PHP on your Local Machine**

To run PHP scripts on any machine (server, computer etc.) we need PHP installed on it. In this tutorial, we will learn how to install PHP on our computer/laptop and setup the development environment

**Requirements for PHP**

To run PHP scripts, we need the following services:

1. PHP Parser: To execute PHP scripts, PHP installation is required.
2. Web Server: Because PHP is mostly used to develop websites, hence most of its implementations comes bundled with Apache Web Server, which is required to host the application developed in PHP over HTTP.
3. Database: Any one database management system, which is generally MySQL, as PHP comes with a native support for MySQL

Now, you can install all the 3 services separately yourself, or you can simply download and install software's which automatically installs all the above services.

The most popular such software package is XAMPP.

**What is XAMPP?**

XAMPP stands for:

X: Cross Platform, as it supports all the modern operating systems like Windows, Mac OSX, and Linux etc.

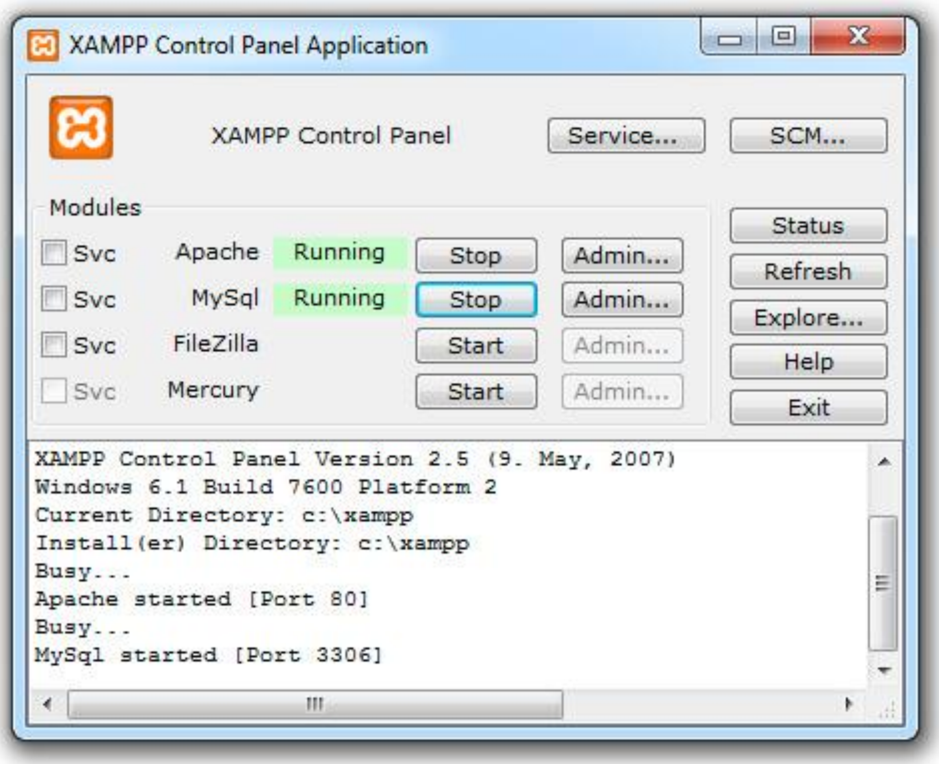
A: Apache Web Server

M: MySQL database management system.

P: PHP installation

P: Perl scripting language

You can easily download and install XAMPP from this link. The installation process is simple and once installed you will see a small window like this, showing the status of various services which are running.


XAMPP Control Panel Application
XAMPP Control Panel
Service...
SCM...
Modules
Svc Apache Running Stop Admin...
Svc MySql Running Stop Admin...
Svc FileZilla Start Admin...
Svc Mercury Start Admin...
Status
Refresh
Explore...
Help
Exit

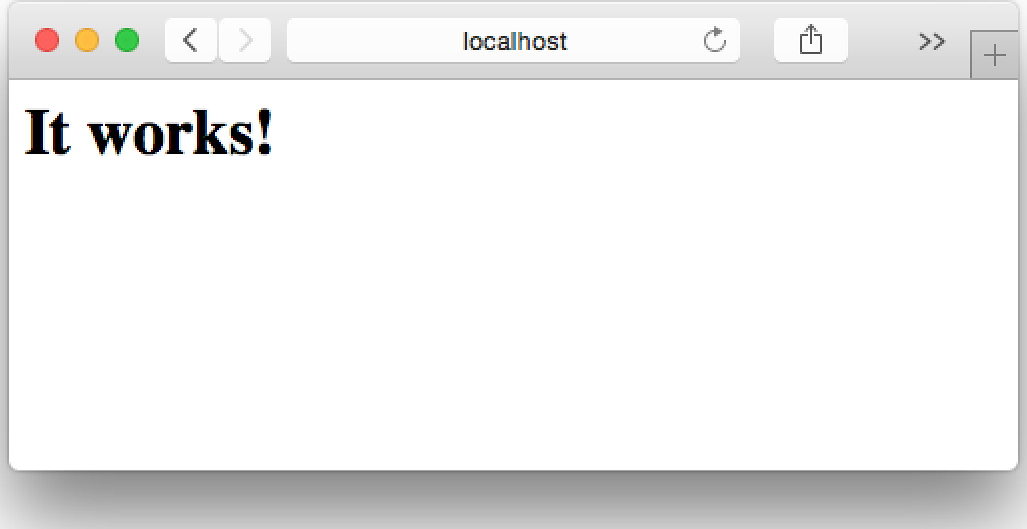
```
XAMPP Control Panel Version 2.5 (9. May, 2007)
Windows 6.1 Build 7600 Platform 2
Current Directory: c:\xampp
Install(er) Directory: c:\xampp
Busy...
Apache started [Port 80]
Busy...
MySql started [Port 3306]
```

You can easily control, stop and restart, various services using the XAMPP Control Panel.

Upon successful installation, a folder with name xampp will be created in the C drive (by default). In the folder xampp there are many sub-folders like apache, cgi-bin, FileZillaFTP etc, but the most important sub-folders are:

1. htdocs: This is the folder in which we will keep all our PHP files.
2. MySQL: This folder contains all the files for the MySQL database. By default the My SQL database runs on port number 3306.
3. Php: This folder holds all the installation files for PHP. All the configurations for the current PHP installation is saved in php.ini file which is stored in this folder.

If everything seems fine and the apache server is running (check in the XAMPP control panel), open your Web browser and enter local host in the address bar and hit enter. You will see the default page of Apache web server.



It works!

## 7.2 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons.

- To structure the data so that there is no repetition of data , this helps in saving.

- To permit simple retrieval of data in response to query and report request.
- To simplify the maintenance of the data through updates, insertions, Deletions.
- To reduce the need to restructure or reorganize data which new application Requirements arise.

### 7.2.1 RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

### RELATIONS, DOMAINS & ATTRIBUTES

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

### RELATIONSHIPS

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.

- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.
- Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity.

## NORMALIZATION

As the name implies, it denoted putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These includes:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

### First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values.

The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each monatomic attribute or nested relation. This eliminated repeating groups of data.

A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

### Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.

In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on apart of the key.

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

**Third Normal Form**

According to Third Normal Form, Relation should not have a nonkey attribute functionally determined by another nonkey attribute or by a set of nonkey attributes. That is, there should be no transitive dependency on the primary key.

In this we decompose and set up relation that includes the nonkey attributes that functionally determines other nonkey attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key.

A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

## 7.3 INTRODUCTION TO MySQL

The MySQL database has become the world's most popular open source database because of its high performance, high reliability and ease of use. It is also the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) Many of the world's largest and fastest-growing organizations including Facebook, Google, Adobe, Alcatel Lucent and Zappos rely on MySQL to save time and money powering their high-volume Web sites, business-critical systems and packaged software.

MySQL runs on more than 20 platforms including Linux, Windows, Mac OS, Solaris, IBM AIX, giving you the kind of flexibility that puts you in control. Whether you're new to database technology or an experienced developer or DBA, MySQL offers a comprehensive range of database tools, support, training and consulting services to make you successful.

For many years, large corporations have enjoyed the ability to deploy relational database management systems (RDBMSs) across their enterprise. Companies have used these systems to collect vast amounts of data that serve as the “fuel” for numerous applications that create useful business information. Until recently, RDBMS technology has been out of reach for small businesses and individuals. Widely used RDBMS systems such as Oracle and DB2 require complex, expensive hardware. License fees for these systems are in the tens to hundreds of thousands of dollars for each installation. Businesses must also hire and retain staff with specialized skill sets to maintain and develop these systems. Smaller enterprises have relied on systems like Microsoft Access and FoxPro to maintain their corporate data.

Early on, during the explosive growth of the Internet, open source database systems like MySQL, Postgres (now PostgreSQL), and MySQL became available for use. Over a relatively short amount of time, the developers of these systems have provided a large subset of the functionality provided by the expensive commercial database systems. These open source database systems also run on less-expensive commodity hardware, and have proven in many cases to be easier to develop for and maintain than their commercial counterparts.

Finally, smaller businesses and individuals have access to the same powerful level of software tools that large corporations have had access to for over a decade.

Why Use an RDBMS?

Almost every piece of software that has been developed needs to persist or store data. Once data has been persisted, it is natural to assume that this data needs to be retrieved, changed, searched, and analyzed. You have many options for data persistence in your software, from rolling your own code, to creating libraries that access flat files, to using full-blown RDBMS systems. Factors to consider when choosing a persistence strategy include whether you need multiuser access, how you will manage storage requirements, whether you need transactional integrity, and whether the users of your software need ad hoc query capability. RDBMSs offer all of this functionality.

Multiuser Access

Many programs use flat files to store data. Flat files are simple to create and change. The files can be used by many tools, especially if they are in comma- or tab-delimited formats. A large selection of built-in and third-party libraries is available for dealing with flat files in Java. The java.util.Properties class included with the Java Development Kit is one example. Flat file systems can quickly become untenable when multiple users require simultaneous access to the data. To prevent corrupting the data in your file, you must lock the file during changes, and perhaps even during reads. While a file is locked, it cannot be accessed by other users. When the file becomes larger and

the number of users increases, this leads to a large bottleneck because the file remains locked most of the time—your users are forced to wait until they can have exclusive access to the data. RDBMSs avoid this situation by employing a number of locking strategies at varying granularities. Rather than using a single lock, the database system can lock an individual table, an individual page (a unit of storage in the database, usually covering more than one row), or an individual row. This increases throughput when multiple users are attempting to access your data, which is a common requirement in Web-based or enterprise-wide applications.

Storage Transparency

If you use flat files in your software, you are also responsible for managing their storage on disk. You have to figure out where and how to store the data, and every time the location or layout of the files changes, you are required to change your software. Once the datasets your software is storing become numerous or large, the storage management process becomes cumbersome. Using a database system gives you “storage transparency.” Your software does not care where and how the data is stored. The data can even be stored on some other computer and accessed via networking protocols. Transactions when you have more than one user accessing and changing your data, you want to make these changes transactional. Transactions group operations on your data into units of work that meet the ACID test. The ACID test concept is best illustrated with a commonly used example from the banking industry. Jack and Jill share a joint checking account with a balance of $1000. They are both performing various operations, such as deposits, withdrawals, and transfers, on the account. Let’s see how the four aspects of the ACID test come into play:

■■ Atomicity: All changes made during a transaction are made successfully, or in the case of failure, none are made. If any operation fails during the transaction, then the entire transaction is rolled back, leaving your data in the state it was before the transaction was started. For example, suppose Jack is making a transfer of $500 from his checking account to a savings account. Sometime between the withdrawal of the $500 from the checking account and the deposit of $500 to the savings account, the software running the banking system crashes. Jack's $500 has disappeared! With atomicity, either the entire transfer would have happened, or none of it would have happened, leaving Jack a much happier customer than he is now.

■■ Consistency: All operations transform the database from one consistent state to another consistent state. Consistency is defined by how the database schema is designed and whether integrity constraints such as foreign keys are used. The database management system is responsible for ensuring that transactions do not violate the database schema or integrity constraints. For example, the bank's database developers have declared in the database schema that the balance of an account cannot be empty, or "null." If any transaction attempts to set the balance to an empty value, the transaction will be aborted and any changes rolled back.

■■ Isolation: A transaction's changes are not made visible to other transactions until they are committed under the atomicity rule described earlier. This is best demonstrated by what happens when month-end reports are generated. Let's say that Jack is performing the transfer transaction outlined in the atomicity example, and at the same time you are generating his monthly statement. Without isolation, the monthly statement might show the withdrawal from the checking account but not the deposit into the savings account. This discrepancy would make it impossible for Jack or the bank to balance their books.

■■ Durability: Once completed, a transaction's changes are never lost through system or hardware crashes. If Jill has paid for $50 worth of groceries with her debit card at the grocery store and the transaction succeeds, even if the database software crashes immediately after the transaction competes, it won't forget that her checking account balance is $50 lower.

Until recently, MySQL did not comply with all components of the ACID test. However, with the new BDB and InnoDB table types (supported in MySQL 3.23 and MySQL 4.0), MySQL can now pass the ACID test. Not all software requires the robustness (or the associated overhead) of transaction semantics. MySQL is one of the only databases that enable you to decide what level of robustness you need on a table-by-table basis. This becomes important when you are trying to maximize performance, especially when much of the data is read-only (such as in a product catalog).

### Searching, Modifying, and Analyzing Data

Any time you store a significant amount of data with your software, your users want to search, modify, and analyze the data you have stored. If you are using flat files, you most likely have to develop this functionality yourself. As your data stored in flat files takes up more and more space, it takes longer and longer to search. A common solution to this problem is to create an index for your data. Indexes are basically shortcuts to finding a particular piece of data, usually using some sort of key. If you need to develop indexing functionality yourself, you have to learn about data structures, such as hashes and Btrees, and how to store these indexes alongside your data. In addition, you must learn how to implement the index in your software. If you use an RDBMS, you can tell the database system what data you think people will search on, and it does all of the fancy indexing for you. Users of your software also want to retrieve, modify, and analyze the data you have stored. They expect that your system knows how to compute such values as sums, averages, minimums, and maximums to be used for updating related data or analyzing existing data. They expect that your software will be able to

sort the data or group the data by similar attributes. All of this functionality requires you to implement numerous functions and algorithms. If you use an RDMBS, all of these features are built in.

### Ad Hoc Queries

It is likely that your software will need to retrieve stored data using arbitrary parameters, otherwise known as ad hoc queries. This becomes difficult with flat files because they are not self-describing, and every file layout is different. You also need to consider how you are going to read the data for these queries from your persistent storage mechanism. Many RDBMSs use SQL (Structured Query Language) for manipulating data. SQL is a declarative language in that you declare what data you want, not the procedure for how to get it. SQL is also an accepted and widely used standard, so a large set of tools are available (JDBC and Enterprise Java Beans, among them) to help you work with it. After outlining all of the benefits of an RDBMS, I hope you are ready to consider using one for your software projects. The next question to ask is “Why choose My SQL?”

Why Choose MySQL?

As was the case with many other open source projects, MySQL was first created by someone who needed a better tool to get a specific job done. Monty Widenius and David Axmark started out with another open source project (MSQL), but found that it lacked some features that they needed. They decided to develop their own database system that met their specific requirements. They started building MySQL by using some low-level database storage code they had already developed for other projects and layered a multithreaded server, SQL parser, and client-server protocol on top. They also structured the API for MySQL to appear very similar to MSQL in order to make it easier for developers to port their MSQL-based software to MySQL. MySQL was eventually released in source-code form, under a proprietary license. Eventually, this license was changed to the GNU General Public License (GPL), which in most cases allows the software to be used without license cost. However, in certain situations you must purchase a commercial license.

The exact terms of the license are available in the documentation that ships with MySQL or on the Web at www.mysql.com. Commercial support is also available for those who need it from MySQL-AB, the company that was created by Monty and David to support the continued development of the MySQL software. The requirements that Monty and David originally had for MySQL were that it be as fast as possible, while still being stable, simple to use, and able to meet the needs of the majority of database developers. Even today, feature requests for future MySQL development are weighed carefully against these original requirements, and are implemented only when and if the original requirements can be met as much as possible. Over the years, MySQL has evolved into an RDBMS that has the following core

**Features**

■■ Portability: MySQL runs on almost every flavor of UNIX, as well as Windows and MacOS X. You can obtain binaries or source code for the MySQL server as well as the tools that access it. More ports of the software become available every day. It is almost a given that MySQL will run on whatever operating system you have available.

■■ Speed: Using techniques such as efficient indexing mechanisms, in memory temporary tables, and highly optimized join algorithms, MySQL executes most queries much faster than most other database systems.

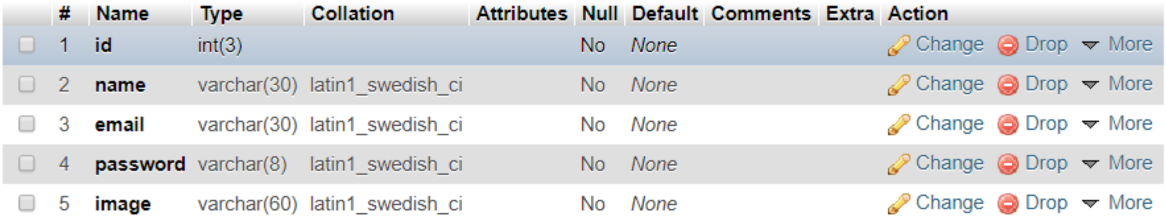
■■ Scalability: Because of its modularity and its flexibility in configuration, MySQL can run in systems varying in size from embedded systems to large multiprocessor UNIX servers hosting databases with tens of millions of records. This scalability also allows you to run a copy of MySQL on a developer-class machine, and later use the same database system on a larger machine in production. Because it is multithreaded, MySQL efficiently utilizes resources for multiple users, compared to other database servers that start full-fledged processes for each user. It is not uncommon to hear of MySQL installations supporting thousands of concurrent users.

■■ Flexibility: MySQL lets you choose the table types you need to meet your software's requirements, ranging from in-memory heap tables, fast on-disk MyISAM tables, merge tables that group together other sets of tables to form larger "virtual" tables, and transaction-safe tables such as InnoDB. MySQL is also very tunable and includes many parameters that can be changed to increase performance for a given solution. However, MySQL comes with sensible defaults for these parameters, and many users never have to tune MySQL to reach a performance they are happy with.

■■ Ease of use: MySQL is easy to install and administer. While other database systems require special knowledge and training, not to mention special operating system configurations, MySQL can be installed in less than 10 minutes if you've done it before. Even if you are a newcomer, you should be able to install MySQL in under an hour. Once it's installed, MySQL requires little maintenance and administration other than adding or changing user permissions and creating or removing databases.

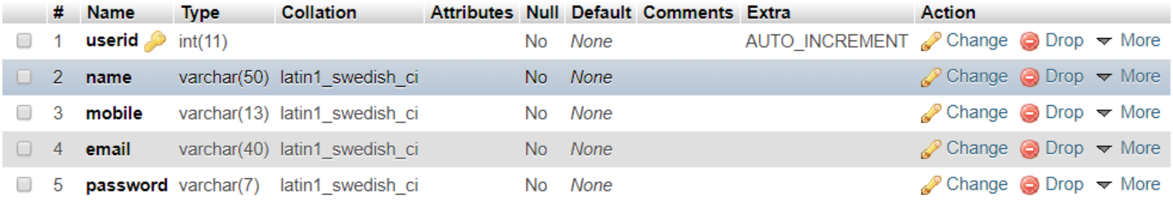
■■ Fine-grained security model: You can restrict users' rights from an entire database down to the column level based on login name, password, and the hostname that users are connecting from. This allows you to create secure systems by partitioning responsibilities and capabilities of different users and applications to prevent unauthorized modification or retrieval of data. Access from other languages/systems: There are libraries and APIs for connecting to MySQL from Java (the focus of this book), C/C++, Perl, PHP, ODBC (Microsoft Windows applications), TCL, Eiffel, and Lisp. Because of this, a whole set of tools has appeared surrounding the use of MySQL from these languages and systems.

## TABLES STRUCTURE



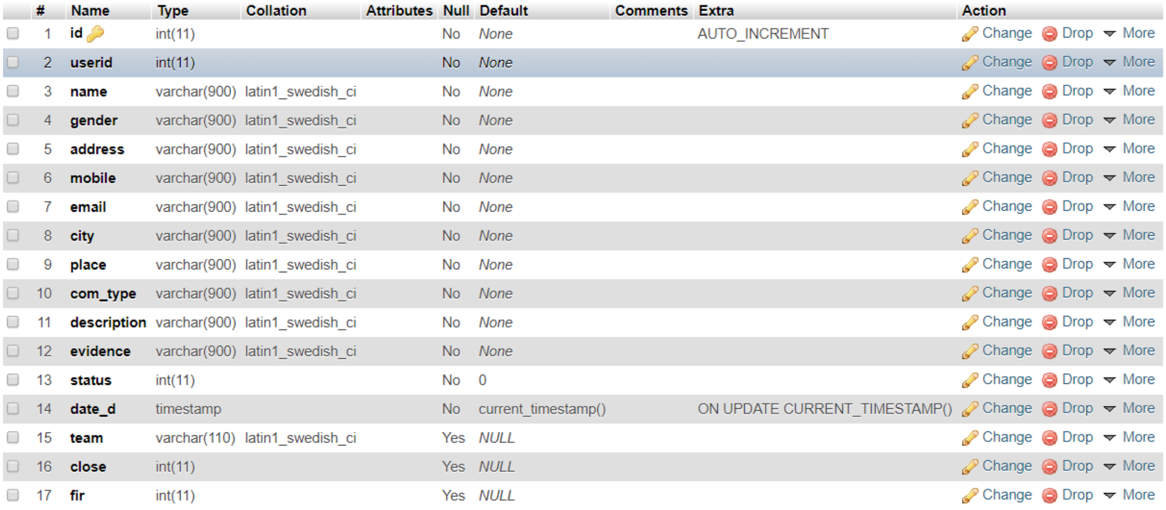
|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **id** | int(3) |  |  | No | *None* |  |  | Change Drop More |
| □ | 2 | **name** | varchar(30) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 3 | **email** | varchar(30) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 4 | **password** | varchar(8) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 5 | **image** | varchar(60) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |

### 1.1 Table Name: Login



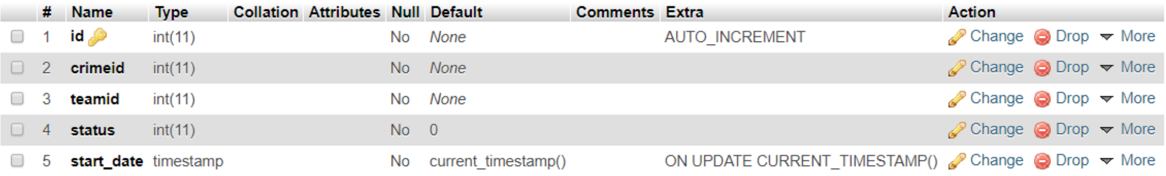
|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **userid** 🔑 | int(11) |  |  | No | *None* |  | AUTO_INCREMENT | Change Drop More |
| □ | 2 | **name** | varchar(50) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 3 | **mobile** | varchar(13) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 4 | **email** | varchar(40) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 5 | **password** | varchar(7) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |

### 1.2 Table Name: User Registration



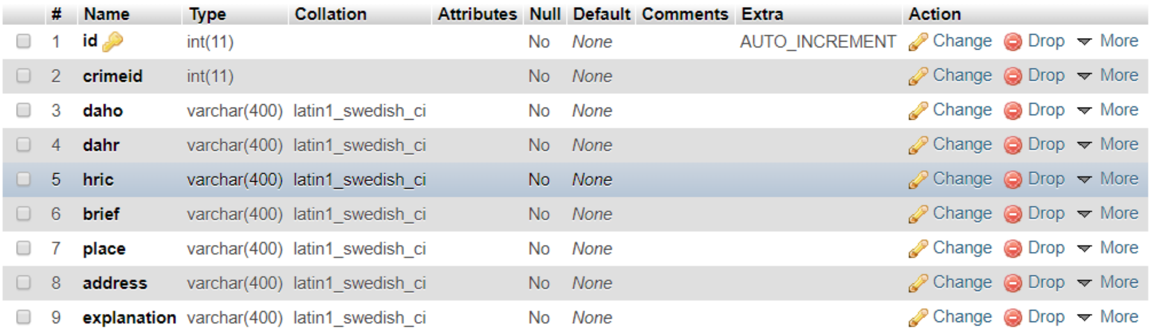
|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **id** 🔑 | int(11) |  |  | No | *None* |  | AUTO_INCREMENT | Change Drop More |
| □ | 2 | **userid** | int(11) |  |  | No | *None* |  |  | Change Drop More |
| □ | 3 | **name** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 4 | **gender** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 5 | **address** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 6 | **mobile** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 7 | **email** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 8 | **city** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 9 | **place** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 10 | **com_type** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 11 | **description** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 12 | **evidence** | varchar(900) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 13 | **status** | int(11) |  |  | No | 0 |  |  | Change Drop More |
| □ | 14 | **date_d** | timestamp |  |  | No | current_timestamp() |  | ON UPDATE CURRENT_TIMESTAMP() | Change Drop More |
| □ | 15 | **team** | varchar(110) | latin1_swedish_ci |  | Yes | *NULL* |  |  | Change Drop More |
| □ | 16 | **close** | int(11) |  |  | Yes | *NULL* |  |  | Change Drop More |
| □ | 17 | **fir** | int(11) |  |  | Yes | *NULL* |  |  | Change Drop More |

### 1.3 Table Name: Crime Report Registration



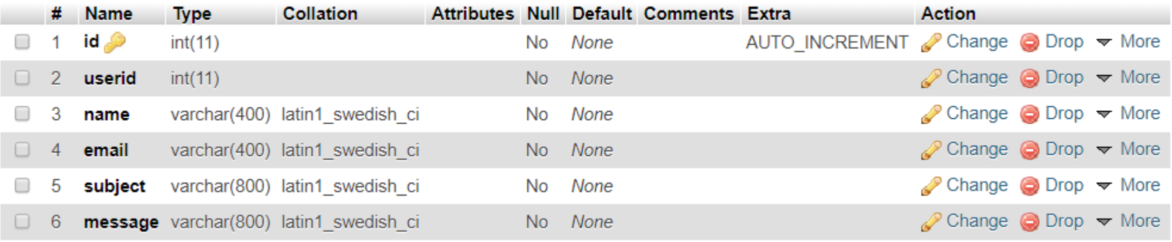
|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **id** 🔑 | int(11) |  |  | No | *None* |  | AUTO_INCREMENT | Change Drop More |
| □ | 2 | **crimeid** | int(11) |  |  | No | *None* |  |  | Change Drop More |
| □ | 3 | **teamid** | int(11) |  |  | No | *None* |  |  | Change Drop More |
| □ | 4 | **status** | int(11) |  |  | No | 0 |  |  | Change Drop More |
| □ | 5 | **start_date** | timestamp |  |  | No | current_timestamp() |  | ON UPDATE CURRENT_TIMESTAMP() | Change Drop More |

## 1.4 Table Name: Team Allot



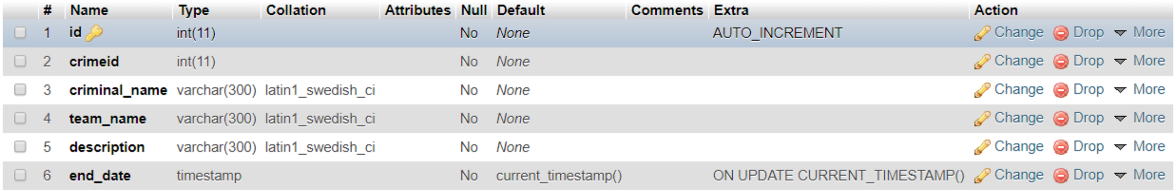
|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **id** 🔑 | int(11) |  |  | No | *None* |  | AUTO_INCREMENT | Change Drop More |
| □ | 2 | **crimeid** | int(11) |  |  | No | *None* |  |  | Change Drop More |
| □ | 3 | **daho** | varchar(400) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 4 | **dahr** | varchar(400) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 5 | **hric** | varchar(400) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 6 | **brief** | varchar(400) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 7 | **place** | varchar(400) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 8 | **address** | varchar(400) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 9 | **explanation** | varchar(400) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |

## 1.5 Table Name: FIR



|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **id** 🔑 | int(11) |  |  | No | *None* |  | AUTO_INCREMENT | Change Drop More |
| □ | 2 | **userid** | int(11) |  |  | No | *None* |  |  | Change Drop More |
| □ | 3 | **name** | varchar(400) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 4 | **email** | varchar(400) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 5 | **subject** | varchar(800) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |
| □ | 6 | **message** | varchar(800) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop More |

## 1.6 Table Name: Feedback



|  | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **id** 🔑 | int(11) |  |  | No | *None* |  | AUTO_INCREMENT | Change Drop ▾ More |
| □ | 2 | **crimeid** | int(11) |  |  | No | *None* |  |  | Change Drop ▾ More |
| □ | 3 | **criminal_name** | varchar(300) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop ▾ More |
| □ | 4 | **team_name** | varchar(300) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop ▾ More |
| □ | 5 | **description** | varchar(300) | latin1_swedish_ci |  | No | *None* |  |  | Change Drop ▾ More |
| □ | 6 | **end_date** | timestamp |  |  | No | current_timestamp() |  | ON UPDATE CURRENT_TIMESTAMP() | Change Drop ▾ More |

## 1.7 Table Name: Missing person

# 8. TESTING

After the implementation phase the testing part begins. Testing of the application determines the correctness, completeness and the quality of the application. Being a developer, the main objective of us is to detect the bug as well as the other errors in an application. Error prevents to produce the correct output/desired output. There are mainly three kinds of error in a system.

- Requirement error
- Design error
- Programming error

Once the testing is done, the application is ready to be performed and delivered the correct output.

✓ There are six kind of testing is done in our project:

- Unit testing
- Performance testing
- Beta testing
- Security testing
- Load testing
- Acceptance testing

## 8.1 UNIT TESTING

Unit testing is performed to test individual units of an application. Since, the application come promises the various unit and modules, detecting errors in this units is simple and consumes less time, as they are small in size. However it is possible that the output produced by one unit becomes the input for another unit. Hence if the incorrect output produced by the one unit is provided as input to the second unit then it also produces wrong output. If this process is not corrected, the entire application may produce unexpected outputs.to avoid this all the units in the application are tested independently using unit testing.

The unit testing is not just performed once during the application development but is repeated whenever the application is modified or used in a new environment. Some other points that kept in mind are

- Each unit is tested separately regardless of other unit of the application.
- The developers themselves performs this testing.
- The method of white box testing is done in this testing.
- In addition unit testing performs the following functions.

➢ It ensures that all statements in the unit have be executed at least once.

➢ It tests data structures that represents relationships among individual elements.

➢ It ensures that the data entered in the variables is of the same data type as defined in the unit.

➢ It checks all arithmetic calculation present in the unit with all possible combinations of the input values.

In our project, we test each unit or module. Hence, administrator is the first module of our project so the entire project is based on the output of the first unit. Suppose for an example we generating a random sequence that is used as username, so if there is any error exist in our application then the total system may be produce expected outputs and errors. After completion of unit testing we successfully avoid this types of problem.

## 8.2 SECURITY TESTING

System with sensitive information are generally the target of improper or illegal use therefore protection mechanism are required to restrict unauthorized access to the system to avoid any kind of improper usage, security testing is performed which identifies and removes the flaws from the application that can be exploited by the intruders and thus results in security violation. Security testing focuses on the following areas of security:
Application security:
To check whether the user can access only those data and functions for which the system developer or user of the system has given permission this kind of security is referred to authorization.
System security:
To check whether only the users who have permissions to access the system are accessing it this security is referred to as authentication.
Security is the main objective of any project. The importance of security testing is to stop unauthorized usages of data, means to ensure that the users only can access the data that is permitted system developer. In our system we uses encryption, decryption and session for security purposes.

## 8.3 LOAD TESTING

Load is the process of putting demand on a system or device and measuring its response. Load testing is performed to determine a systems behavior under both normal and anticipated peak load conditions. It helps to identify the maximum operating capacity of an application as well as any bottle necks and determine which element is causing degradation. When the load placed on the system is raised beyond normal usage patterns, in order to test the systems response at unusually high or peak loads it is known as stress testing. The load is usually so great that error conditions are the expected result, although no clear boundary exist when an activity ceases to be a load test and becomes a stress test.
The term load testing is used in different ways in the professional application testing community. Load testing generally refers to the practice of modelling the expected usage of application program by simulating multiple users accessing the program concurrently. As such, this testing is most relevant for multi user systems; often one built using a client/server model, such as web servers. However, other types of application systems can also be load tested. For example, a word processor or graphics editor can be forced to read an externally large document; are financial package can be forced to generate report based on several years' worth of data. The most accurate load testing simulates actual use, as opposed to testing using theoretical or analytical modelling.
Hence our project is an online web based multi user application so load testing is so imported to state that our application can work smoothly in any critical condition. The

server hit of our system is high because of in each and every step our application compare the input with the attribute of the database that resides in the server side. But at the time of testing we observed that the load testing is successfully passed by our application.

### 8.4 ACCEPTANCE TESTING

In engineering and its various sub disciplines, acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. It may involve chemical tests, physical tests, or performance tests.

In system engineering it may involve black box testing performed on a system (for example: a piece of application, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery.

Application developers often distinguish acceptance testing by the system provider from accepting transfer of ownership. In the case of application, acceptance testing by the customer is known as user acceptance testing (UAT), end-user testing, site (acceptance) testing, or field (acceptance) testing.

### 8.5 BETA TESTING

Application testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Application testing can also provide an objective, independent view of the application to allow the business to appreciate to appreciate and understand the risks of application implementation. Test technique include but are not limited to, the process of executing a program or application with the intent of finding application bug (errors or other defects).

Application testing can be stated as the process of validating and verifying that a computer program/ application/ product:

Meets the requirements that guided its design and development,

- Works as accepted,
- Can be implemented with the same characteristics,
- And satisfies the needs of stakeholders.

## 9. Conclusion

The project titled as “Crime Management System” is a web based application. This software provides facility for reporting online crimes, complaints, missing persons, show mostwanted person details mailing as well as chatting. This software is developed with scalability in mind. Additional modules can be easily added when necessary. The software is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The project has been completed successfully with the maximum satisfaction of the organization. The constraints are met and overcome successfully. The system is designed as like it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. This software has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

# 10. Scope for Future Enhancement

In future we can use Image recognization instead of using heterogeneous database more over High speed, accuracy and non-redundant data are the main advantages of the proposed system. In the proposed system the user is provided with a choice of data screen, which are similar in formats to the source documents. Data entry errors can be minimized through validity checks. After the verification only the data are placed the permanent database. The software can be developed further to include a lot of modules because the proposed system is developed on the view of future, for example we should develop the system as a database independent using JDBC so we can connect it to any other database, Now the proposed system is based on PC and intranet but in the future if we need to convert it into internet then we need to change the front end only because we are developing this on the basis of OOP technology and most of the business logic's are bounded in the class files and module like reusable components.

# 11. SOURCE CODE

## 11.1 Register Form

```
<link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<link href="css/ragister.css" rel="stylesheet">
<script>
$(function() {

   $('#login-form-link').click(function(e) {
                $("#login-form").delay(100).fadeIn(100);
                $("#register-form").fadeOut(100);
                $("#forgot-form").fadeOut(100);

                $('#register-form-link').removeClass('active');
                $('#forgot-form-link').removeClass('active');
                $(this).addClass('active');
                e.preventDefault();
        });
        $('#register-form-link').click(function(e) {
                $("#register-form").delay(100).fadeIn(100);
                $("#login-form").fadeOut(100);
                $("#forgot-form").fadeOut(100);
                $('#login-form-link').removeClass('active');
                $(this).addClass('active');
                e.preventDefault();
        });
        $('#forgot-form-link').click(function(e) {
                $("#forgot-form").delay(100).fadeIn(100);
                $("#register-form").fadeOut(100);
                $("#login-form").fadeOut(100);
                $('#register-form-link').removeClass('active');
                $('#login-form-link').removeClass('active');
                $(this).addClass('active');
                e.preventDefault();
        });

});

</script>
<!------ Include the above in your HEAD tag ---------->
<?php
require_once"dbconfig.php";
?>
<div class="container">
        <div class="row">
                        <div class="col-md-6 col-md-offset-3">
                <div class="panel panel-login">
                    <div class="panel-heading">
                        <div class="row">
                            <div class="col-xs-6">
                                <a href="#" class="active"
id="login-form-link">Login</a>
                            </div>
                            <div class="col-xs-6">
                                <a href="#" id="register-form-
link">Register</a>
                            </div><div class="col-xs-6">
                                <a href="#" id="forgot-form-
link">Forgot Password</a>
                            </div>
                        </div>
                        <hr>
                    </div>
                    <div class="panel-body">
                        <div class="row">
                            <div class="col-lg-12">
                                <form id="login-form"
method="post" role="form" style="display: block;">
                                    <div class="form-
group">
                                        <input
type="email" name="email" required id="username" tabindex="1" class="form-control"
placeholder="Email" value="">
                                    </div>
                                    <div class="form-
group">
                                        <input
type="password" name="password" required id="password" tabindex="2" class="form-
control" placeholder="Password">
                                    </div>

                                    <div class="form-
group">
                                        <div
class="row">
                                            <div
class="col-sm-6 col-sm-offset-3">

    <input type="submit" name="login" id="login-submit" tabindex="4" class="form-
control btn btn-login" value="Log In">
                                            </div>
                                        </div>
                                    </div>

                                </form>
                                <form id="forgot-form"
method="post" role="form" style="display: none;">
                                    <div class="form-
group">
                                        <input
type="Mobile" name="mobile" required id="username" tabindex="1" class="form-control"
placeholder="Mobile" value="">
                                    </div>
                                    <div class="form-
group">
                                        <input
type="email" name="email" required id="username" tabindex="1" class="form-control"
placeholder="Email" value="">
                                    </div>
                                    <div class="form-
group">
                                        <input
type="password" name="password" required id="password" tabindex="2" class="form-
control" placeholder="New Password">
                                    </div>

                                    <div class="form-
group">
                                        <div
class="row">
                                            <div
class="col-sm-6 col-sm-offset-3">

    <input type="submit" name="forgot" id="login-submit" tabindex="4" class="form-
control btn btn-login" value="Forgot">
                                            </div>
                                        </div>
                                    </div>

                                </form>
                                <form id="register-form"
action="" method="post" role="form" style="display: none;">
                                    <div class="form-
group">
                                        <input
type="text" name="name" required  id="username" tabindex="1" class="form-control"
placeholder="Username" value="">
                                    </div>
                                    <div class="form-
group">
                                        <input
type="email" name="email" required  id="email" tabindex="1" class="form-control"
placeholder="Email Address" value="">
                                    </div>
```

```
                                                                    <div class="form-
group">
                                                                            <input
type="text" name="mobile" required  id="password" tabindex="2" class="form-control"
placeholder="Mobile">
                                                                    </div>
                                                                    <div class="form-
group">
                                                                            <input
type="password" name="password" required  id="confirm-password" tabindex="2"
class="form-control" placeholder="Password">
                                                                    </div>
                                                                    <div class="form-
group">
                                                                            <div
class="row">
                                                                                    <div
class="col-sm-6 col-sm-offset-3">

        <input type="submit" name="register" id="register-submit" tabindex="4"
class="form-control btn btn-register" value="Register Now">
                                                                                    </div>
                                                                            </div>
                                                                    </div>
                                                            </form>

<?php
if(isset($_REQUEST['register']))
        {
                extract($_REQUEST);
         $query="INSERT INTO `user`(`name`, `mobile`, `email`, `password`)
                values('$name','$mobile','$email','$password')";
                                $n=iud($query);
         if($n==1)
        {
                echo"<script>alert('Register successfull');
                 window.location='ragister.php';
                 </script>";
        }
        else
        {
        echo"<script>alert('Something Wrong');
                 window.location='ragister.php';
                 </script>";

        }
        }
        if(isset($_REQUEST['forgot']))
        {
                extract($_REQUEST);
```

```
            $query="update `user` set  `password`='$password' where mobile='$mobile' and
email='$email'";
                        $n=iud($query);
            if($n==1)
            {
                        echo"<script>alert(' successful');

                         </script>";
            }
            else
            {
            echo"<script>alert('Something Wrong');

                         </script>";

            }
            }

            if(isset($_REQUEST['login']))
            {
                        extract($_REQUEST);
                        $query="select * from user where email='$email' and
password='$password'";
                        $login_data=select($query);
            $n=mysqli_num_rows($login_data);
            if($n==1)
            {
                        while($data=mysqli_fetch_array($login_data))
                        {
                        extract($data);

                        }

                        $_SESSION['userid']=$userid;
                        $_SESSION['login']="yes";
                        echo"<script>alert(' successful');
                         window.location='index.php';
                         </script>";
                        //header("location:index.php");
            }
            else
            {
                        echo"email or password is incorrect";
            }

            }
            ?>
                                                                  </div>
                                                      </div>
                                          </div>
```

```
                              </div>
                         </div>
                    </div>
          </div>
```

## 11.2 Change password

```
<?php require_once"dbconfig.php";
if(isset($_SESSION['login']))
{


}
else
{
          header("location:login.php");
}

?>
<!DOCTYPE HTML>
<html>
<?php include"head.php";?>
<body>
<div class="page-container">
<div class="left-content">
<div class="inner-content">
<?php //include"header.php";?>
<div class="outter-wp">
<div class="sub-heard-part">
<ol class="breadcrumb m-b-0">
<li><a href="index.html">Home</a></li>
<li class="active">Blank Page</li>
</ol>
</div>
<div class="graph-visual tables-main">
<h2 class="inner-tittle">Change Password</h2>
<div class="graph">
<div class="block-page">
<p>
<h3 class="inner-tittle two">change password </h3>
<div class="grid-1">
<div class="form-body">
<form class="form-horizontal" action="myphp.php" method="post">

<div class="form-group">
<label for="inputPassword3" id="oldpassworderror" class="col-sm-2 control-label">Old
Password</label>
```

```
<div class="col-sm-9">
<input type="password" class="form-control" id="oldpassword" name='oldpassword'
placeholder="old Password">
</div>
</div>
<div class="form-group">
<label for="inputPassword3" id="newpassworderror" class="col-sm-2 control-label">New
Password</label>
<div class="col-sm-9">
<input type="password" class="form-control" id="newpassword" name='newpassword'
placeholder="NewPassword">
</div>
</div>
<div class="form-group">
<label for="inputEmail3" id="cpassworderror" class="col-sm-2 control-label">Confirm-
password</label>
<div class="col-sm-9">
<input type="password" class="form-control" id="cpassword" name="cpassword"
placeholder="Confirm-password">
</div>
</div>

<div class="col-sm-offset-2">
<input type="submit" class="btn btn-default" name="change_password"
id="change_password" value="Change Password"> </div> </form>
</div>

</div>
</p>
</div>

</div>

</div>
</div>
<?php include"footer.php";?>

</div>
</div>
<?php include"side_bar.php";?>
</div>
<?php include"footer_script.php";?>
<script>

$(document).ready(function(){
$("#change_password").click(function(){

var valid=true;
var oldpassword=$.trim($("#oldpassword").val());
var newpassword=$.trim($("#newpassword").val());
var cpassword=$.trim($("#cpassword").val());

if(oldpassword.length<6)
{
$("#oldpassworderror").html('Invalid Old Password');
$("#oldpassworderror").css("color","red");
$("#oldpassword").css("border-color","red");
valid=false;
}
else
{
$("#oldpassworderror").html('Old Password');
$("#oldpassworderror").css("color","black");
$("#oldpassword").css("border-color","#ddd");
}
if(newpassword.length<6)
{
$("#newpassworderror").html('Invalid New Password');
$("#newpassworderror").css("color","red");
$("#newpassword").css("border-color","red");
valid=false;
}
else
{
$("#newpassworderror").html('New Password');
$("#newpassworderror").css("color","black");
$("#newpassword").css("border-color","#ddd");
}

if(cpassword!=newpassword)
{
$("#cpassworderror").html('Invalid Confirm Password');
$("#cpassworderror").css("color","red");
$("#cpassword").css("border-color","red");
valid=false;
}
else
{
$("#cpassworderror").html('Confirm Password');
$("#cpassworderror").css("color","black");
$("#cpassword").css("border-color","#ddd");
}
var mymethod="post";
var myurl="myphp.php";
var
mydata="oldpassword="+oldpassword+"&newpassword="+newpassword+"&cpassword="
+cpassword+"&change=yes";

$.ajax({
```

```
            method:mymethod,
            url:myurl,
            data:mydata,
            success:function(result)
            {
                    if(result==1)
                    {
                alert("Password Changed Successfully");
                            $("#oldpassword").val("");
                            $("#newpassword").val("");
                            $("#cpassword").val("");
                    }
                    else
                    {
                            alert(result);
                    }
}
});
return false;
});
});
</script>
</body>
</html>
```

## 11.3 Login

```
<!DOCTYPE html>
<head>
<title>Team Login</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="keywords" content="ADMIN Responsive web template,
Bootstrap Web Templates, Flat Web Templates, Android Compatible web
template,
Smartphone Compatible web template, free webdesigns for Nokia, Samsung,
LG, SonyEricsson, Motorola web design" />
<script type="application/x-javascript"> addEventListener("load", function()
{ setTimeout(hideURLbar, 0); }, false); function hideURLbar(){
window.scrollTo(0,1); } </script>
<!-- bootstrap-css -->
<link rel="stylesheet" href="css/bootstrap.min.css" >
<!-- //bootstrap-css -->
<!-- Custom CSS -->
<link href="css/style.css" rel='stylesheet' type='text/css' />
<link href="css/style-responsive.css" rel="stylesheet"/>
```

```
<!-- font CSS -->
<link
href='//fonts.googleapis.com/css?family=Roboto:400,100,100italic,300,300ita
lic,400italic,500,500italic,700,700italic,900,900italic' rel='stylesheet'
type='text/css'>
<!-- font-awesome icons -->
<link rel="stylesheet" href="css/font.css" type="text/css"/>
<link href="css/font-awesome.css" rel="stylesheet">
<!-- //font-awesome icons -->
<script src="js/jquery2.0.3.min.js"></script>
</head>
<body>
<div class="log-w3">
<div class="w3layouts-main">
        <h2>Sign In Now</h2>
                <form action="myphp.php" method="post">
                        <input type="text" class="ggg" name="name"
placeholder="Team Name" required="">
                        <input type="text" class="ggg" name="id"
placeholder="Team Id" required="">
                                <div class="clearfix"></div>
                                <input type="submit" value="Sign In"
name="login">
                </form>
                <p></p>
</div>
</div>
<script src="js/bootstrap.js"></script>
<script src="js/jquery.dcjqaccordion.2.7.js"></script>
<script src="js/scripts.js"></script>
<script src="js/jquery.slimscroll.js"></script>
<script src="js/jquery.nicescroll.js"></script>
<!--[if lte IE 8]><script language="javascript" type="text/javascript"
src="js/flot-chart/excanvas.min.js"></script><![endif]-->
<script src="js/jquery.scrollTo.js"></script>
</body>
</html>
```

Logout

```
<?php
```

```
session_start();
session_destroy();
header("location:index.php");
?>
```

## 11.4 Status

```
<?php
require_once"dbconfig.php";

?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="description" content="">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
     <title>Crime Management System</title>


    <link rel="icon" href="img/core-img/favicon.ico">

     <link href="style.css" rel="stylesheet">

    <link href="css/responsive/responsive.css" rel="stylesheet">
<script src="jquery.min.js"></script>
           <script>
 $(document).ready(function(){
      $('#country').keyup(function(){
           var query = $(this).val();
aler(query);
           if(query != '')
           {
                $.ajax({
                     url:"searchinput.php",
                     method:"POST",
                     data:{query:query},
                     success:function(data)
                     {
                          $('#countryList').fadeIn();
                          $('#countryList').html(data);

                     }
                });
           }
      });
      $(document).on('click', 'li', function(){
          $('#country').val($(this).text());
          $('#countryList').fadeOut();
      });

$('.navbar-light .dmenu').hover(function () {
        $(this).find('.sm-menu').first().stop(true, true).slideDown(150);
    }, function () {
        $(this).find('.sm-menu').first().stop(true, true).slideUp(105)
    });
 });
 </script>
</head>

<body>
    <div id="preloader">
        <div class="dorne-load"></div>
    </div>

   <header class="header_area" id="header">
        <div class="container-fluid h-100">
            <div class="row h-100">
                <div class="col-12 h-100">
                    <nav class="h-100 navbar navbar-expand-lg">
                        <a class="navbar-brand" href="index.php"><img src="img/logo.png"></a>

                        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#dorneNav" aria-controls="dorneNav" aria-expanded="false" aria-label="Toggle
navigation"><span class="fa fa-bars"></span></button>
                        <!-- Nav -->
                        <div class="collapse navbar-collapse" id="dorneNav">
                            <ul class="navbar-nav mr-auto" id="dorneMenu">
                                <li class="nav-item active">
                                    <a class="nav-link" href="index.php"><span class="sr-
only">(current)</span></a>
                                </li>
                                <li class="nav-item dropdown">
                                    <
                                </li>

                            </ul>
                             <?php include"nav.php";?>


                        </div>
                    </nav>
                </div>
            </div>
        </div>
    </header>
```

```
     <section class="dorne-welcome-area bg-img bg-overlay" style="background-image:
url(img/bg-img/hero-1.jpg);">
        <div class="container h-100">
            <div class="row h-100 align-items-center justify-content-center">
                <div class="col-12 col-md-6"></br></br>
                    <div class="hero-content"></br></br></br></br></br>



                    </div>

<div class="container-fluid">
<div class="row">
<div class="col-lg-4">


</div>
</div>
<div id="countryList" style="background-color:yellow"></div>
                </div>
            </div>
        </div>
        <!-- Hero Social Btn -->

    </section>


    <!-- ***** Features Destinations Area Start ***** -->
    <section class="dorne-features-destinations-area">
        <div class="container-fluid">
            <div class="row">
                <div class="col-12">
                    <div class="section-heading dark text-center">
                        <span></span>
                        <h4>CRIME REPORT STATUS</h4>

                    </div>
                </div>
            </div>

            <div class="row">
                <div class="col-12">
<?php
$t=select("SELECT * FROM `crime` LEFT JOIN team on crime.team=team.id
 where crime.userid='".$_SESSION['userid']."' and crime.close is null");
$y=mysqli_num_rows($t);
if($y>=1)
{
?>
                                                <table class="table">
				<tr style="font-weight:bold">
				<td>S.no. </td>
				<td>ComPaint Type</td>
				<td>Description</td>
				<td>Date</td>
				<td>Place</td>
				<td>Evidence</td>
				<td>Status</td>
				</tr>
				<?php
				$p=1;
				while($r=mysqli_fetch_array($t))
				{extract($r);
				?>
				<tr>
				<td><?=$p?></td>
			    <td><?=$com_type?></td>
				<td><?=$description?></td>
				<td><?=$date_d?></td>
				<td><?=$place?></td>
				<td><a href="img/<?=$evidence?>"><button class="btn btn-success">Download</button></a></td>
				<td><?php
				if(!empty($close))
			{
				echo"CLOSED";
			}else
			{
				if(!empty($team))
			{echo ucwords($team_name)." Is Handling Your Case";}
			}



			if($r[12]==0)
			{
				echo"Your Application has  Sent";
			}
			if($r[12]==1)
			{
				echo"Your Application has been Accepted";
			}
			if($r[12]==2)
			{
				echo"Your Application has been Rejected";
			}

				?></td>
				</tr>
				<?php
```

```
                                    $p++;
                                    }
                                    ?>
              </table>
                                          <?php
}
else
{
      ?>
<h1 style="font-weight:bold;text-align:center">Status Is'nt Available</h1>
      <?php
}
                                          ?>
        </div>
      </div></br></br>







    </div>
  </section>


  <!-- ***** Clients Area Start ***** -->


  <!-- ****** Footer Area Start ****** -->
  <footer class="dorne-footer-area">
    <div class="container-fluid">
      <div class="row">
        <div class="col-12 d-md-flex align-items-center justify-content-between">
          <div class="footer-text">

          </div>

        </div>
      </div>
    </div>
  </footer>
  <!-- ****** Footer Area End ****** -->

  <!-- jQuery-2.2.4 js -->
  <script src="js/jquery/jquery-2.2.4.min.js"></script>
  <!-- Popper js -->
  <script src="js/bootstrap/popper.min.js"></script>
  <!-- Bootstrap-4 js -->
  <script src="js/bootstrap/bootstrap.min.js"></script>
  <!-- All Plugins js -->
```

```
    <script src="js/others/plugins.js"></script>
    <!-- Active JS -->
    <script src="js/active.js"></script>
</body>

</html>
```

## 11.5 Closed Form

```
<?php require_once"dbconfig.php";
if(isset($_SESSION['login']))
{

}
else
{
        header("location:login.php");
}

?>
<!DOCTYPE html>
<head>
<title>Team</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="keywords" content="ADMIN Responsive web template, Bootstrap Web
Templates, Flat Web Templates, Android Compatible web template,
Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG,
SonyEricsson, Motorola web design" />
<script type="application/x-javascript"> addEventListener("load", function() {
setTimeout(hideURLbar, 0); }, false); function hideURLbar(){ window.scrollTo(0,1); }
</script>
<!-- bootstrap-css -->
<link rel="stylesheet" href="css/bootstrap.min.css" >
<!-- //bootstrap-css -->
<!-- Custom CSS -->
<link href="css/style.css" rel='stylesheet' type='text/css' />
<link href="css/style-responsive.css" rel="stylesheet"/>
<!-- font CSS -->
<link
href='//fonts.googleapis.com/css?family=Roboto:400,100,100italic,300,300italic,400italic,
500,500italic,700,700italic,900,900italic' rel='stylesheet' type='text/css'>
<!-- font-awesome icons -->
<link rel="stylesheet" href="css/font.css" type="text/css"/>
<link href="css/font-awesome.css" rel="stylesheet">
<!-- //font-awesome icons -->
<script src="js/jquery2.0.3.min.js"></script>
</head>
<body>
<section id="container">
```

```
<!--header start-->
<header class="header fixed-top clearfix">
<!--logo start-->
<div class="brand">

    <a href="index.php" class="logo">
        Team
    </a>
    <div class="sidebar-toggle-box">
        <div class="fa fa-bars"></div>
    </div>
</div>
<!--logo end-->

<?php include"nav_top.php";?>

</header>
<!--header end-->
<!--sidebar start-->
<?php include"sidebar.php";?>


<!--header end-->
<!--sidebar start-->

<!--sidebar end-->
<!--main content start-->
<section id="main-content">
            <section class="wrapper">
                        <div class="table-agile-info">
 <div class="panel panel-default">
    <div class="panel-heading">
     close form
    </div>
    <div>
      <div class="form-body">

<fieldset>


<form method="post">
<div class="form-group">
  <label class="col-md-8 "> Criminal Name</label>
  <div class="col-md-8 inputGroupContainer">
  <div class="input-group">
  <span class="input-group-addon"><i class="glyphicon glyphicon-user"></i></span>
  <input  name="name" placeholder=" Name" class="form-control"  type="text">
    </div>
  </div>
</div>
```

```
<!-- Select Basic -->

<div class="form-group">
  <label class="col-md-8 ">Complaint Details</label>
    <div class="col-md-8 inputGroupContainer">
    <div class="input-group">
        <span class="input-group-addon"><i class="glyphicon glyphicon-home"></i></span>
  <input name="des" placeholder="Description" class="form-control"  type="text">
    </div>
  </div>
</div>


  <input name="close" value="close" class="form-control btn btn-warning col-lg-3"
type="submit">
    </div>

</div>
<!-- Text input-->



<!-- Text input-->

<!-- radio checks -->


<!-- Text area -->


</form>

<?php
if(isset($_REQUEST['close']))
{
	extract($_REQUEST);
	$crimeid=$_REQUEST['id'];
	$team=$_SESSION['Name'];
	$n=iud("INSERT INTO `close`( `crimeid`, `criminal_name`, `team_name`,
`description`) VALUES ('$crimeid','$name','$team','$des')");
	if($n==1)
	{
		$p=iud("update crime set close='1' where id='$crimeid'");
		if($p==1)
		{
			echo"<script>alert('successfully');
		 window.location='view_life_info.php';
		 </script>";
		}
		else
```

```
                    {
                              echo"<script>alert('Something Wrong');
                     window.location='view_life_info.php';
                     </script>";
                    }
          }
          else
          {
                    echo"<script>alert('Something Wrong');
                     window.location='view_life_info.php';
                     </script>";
          }

}
?>
   </div>
  </div>
</div>

</section>
 <!-- footer -->
                     <div class="footer">
                              <div class="wthree-copyright">
                                <p>©  All rights reserved </p>
                              </div>
                     </div>
  <!-- / footer -->
</section>

<!--main content end-->
</section>
<script src="js/bootstrap.js"></script>
<script src="js/jquery.dcjqaccordion.2.7.js"></script>
<script src="js/scripts.js"></script>
<script src="js/jquery.slimscroll.js"></script>
<script src="js/jquery.nicescroll.js"></script>
<!--[if lte IE 8]><script language="javascript" type="text/javascript" src="js/flot-chart/excanvas.min.js"></script><![endif]-->
<script src="js/jquery.scrollTo.js"></script>
</body>
</html>
```

## 11. 6 Image Upload

```
 <?php
            include"dbconfig.php";
            if(isset($_REQUEST['upload']))
            {
                     extract($_REQUEST);
                    $error=$_FILES["myfile"]["error"];
```

```
$name=$_FILES["myfile"]["name"];
$type=$_FILES["myfile"]["type"];
$size=$_FILES["myfile"]["size"];
$tmp_name=$_FILES["myfile"]["tmp_name"];

          if(move_uploaded_file($tmp_name,"img/$name"))
          {
                    echo"<script>alert('uploaded');

                    </script>";

header("location:preview.php?back=$background_image&image=$name&id=$id");
          }
            else
            {
                      echo"<script>alert('something wrong')</script>";
            }




            }
            ###################################################################
            if(isset($_REQUEST['upload_canvas']))
            {
                     extract($_REQUEST);
                    $error=$_FILES["myfile"]["error"];

$name=$_FILES["myfile"]["name"];
$type=$_FILES["myfile"]["type"];
$size=$_FILES["myfile"]["size"];
$tmp_name=$_FILES["myfile"]["tmp_name"];

          if(move_uploaded_file($tmp_name,"img/$name"))
          {
                    echo"<script>alert('uploaded');

                    </script>";

header("location:preview_canvas.php?back=$background_image&image=$name&id=$id")
;
          }
            else
            {
                      echo"<script>alert('something wrong')</script>";
            }
```

```
              }

#########################################################################
############
              if(isset($_REQUEST['upload_collage']))
              {
                   extract($_REQUEST);
                  $error=$_FILES["myfile"]["error"];

$name=$_FILES["myfile"]["name"];
$type=$_FILES["myfile"]["type"];
$size=$_FILES["myfile"]["size"];
$tmp_name=$_FILES["myfile"]["tmp_name"];

             if(move_uploaded_file($tmp_name,"img/$name"))
             {
                  echo"<script>alert('uploaded');

                  </script>";

header("location:preview_collage.php?back=$background_image&image=$name&id=$id"
);
             }
              else
              {
                    echo"<script>alert('something wrong')</script>";
              }




              }
              ?>
```

## 11.7 DataBase Config

```
<?php

session_start();
define("server","localhost",true);
      define("user","root",true);
      define("password","",true);
      define("database","criminal",true);
      function iud($query)
      {
```

```
                $cid=mysqli_connect(server,user,password,database) or
die("connection error");
        $result=mysqli_query($cid,$query);
        $n=mysqli_affected_rows($cid);
        mysqli_close($cid);
        return $n;
        }

function select($query)
{
        $cid=mysqli_connect(server,user,password,database) or
die("connection error");
        $result=mysqli_query($cid,$query);
        mysqli_close($cid);
        return $result;
}
?>
```

11.8 VIEW TEAM

```
<?php require_once"dbconfig.php";
if(isset($_SESSION['login']))
{


}
else
{
        header("location:login.php");
}

?>
<!DOCTYPE HTML>
<html>
<head>
<?php include"head.php";?>
<script type="text/javascript" src="js/nicEdit-latest.js"></script>
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.19/css/jquery.dataTables.css">

<script type="text/javascript" charset="utf8"
src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.js"></script>


  <script type="text/javascript">
  $(document).ready(function () {
$('#dtBasicExample').DataTable();
$('.dataTables_length').addClass('bs-select');
});

  </script>
  <style>
  table.dataTable thead>tr>td.sorting,
table.dataTable thead>tr>td.sorting_asc,
table.dataTable thead>tr>td.sorting_desc,
table.dataTable thead>tr>th.sorting,
table.dataTable thead>tr>th.sorting_asc,
table.dataTable thead>tr>th.sorting_desc {
  padding-right: 30px
}

table.dataTable thead .sorting,
table.dataTable thead .sorting_asc,
table.dataTable thead .sorting_asc_disabled,
table.dataTable thead .sorting_desc,
table.dataTable thead .sorting_desc_disabled {
  cursor: pointer;
  position: relative
}

table.dataTable thead .sorting:after,
table.dataTable thead .sorting:before,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_asc_disabled:after,
table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc:after,
table.dataTable thead .sorting_desc:before,
table.dataTable thead .sorting_desc_disabled:after,
table.dataTable thead .sorting_desc_disabled:before {
  position: absolute;
  bottom: .9em;
  display: block;
  opacity: .3
}

table.dataTable thead .sorting:before,
table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc:before,
table.dataTable thead .sorting_desc_disabled:before {
  right: 1em;
  content: "\f0de";
  font-family: FontAwesome;
  font-size: 1rem
}

table.dataTable thead .sorting:after,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_asc_disabled:after,
table.dataTable thead .sorting_desc:after,
table.dataTable thead .sorting_desc_disabled:after {
  content: "\f0dd";
  font-family: FontAwesome;
  right: 16px;
  font-size: 1rem
}

table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_desc:after {
  opacity: 1
}

table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc_disabled:after {
  opacity: 0
}
  </style>
  </head>
<body>
<div class="page-container">
<div class="left-content">
<div class="inner-content">
<?php //include"header.php";?>
<div class="outter-wp">
<div class="sub-heard-part">
<ol class="breadcrumb m-b-0">
<li><a href="index.html">Home</a></li>
<li class="active">Team List</li>
</ol>
</div>
<div class="graph-visual tables-main">

<div class="graph">
<div class="block-page">
<p>
<h3 class="inner-tittle two">Team List</h3>
<a href="additem.php"> <button class="btn btn-pill btn-primary">Add New Team
</button></a>
<div class="form-body">
<table id="dtBasicExample" class="table table-striped table-bordered" cellspacing="0"
width="100%">
  <thead>
    <tr>
            <th class="th-sm">S.no.
      </th>
      <th class="th-sm">team Name
      </th>
      <th class="th-sm">Team ID
      </th>


    </tr>
  </thead>
  <tbody>
  <?php
  $result=select("select * from team");
  $n=1;
  while($r=mysqli_fetch_array($result))
  {  extract($r);
  ?>
    <tr>
      <td><?=$n?></td>
      <td><?=ucwords($team_name)?></td>
      <td><?=$teamid?></td>


    </tr>

    <?php
          $n++;
  }
          ?>


    </tbody>

</table>
</div>


</p>
</div>

</div>

</div>
</div>
<?php include"footer.php";?>

</div>
</div>
<?php include"side_bar.php";?>
</div>
<?php include"footer_script.php";?>

</body>
</html>
```

11.9 VIEW FEEDBACK

```
<?php require_once"dbconfig.php";
if(isset($_SESSION['login']))
{


}
else
{
	header("location:login.php");
}

?>
<!DOCTYPE HTML>
<html>
<head>
<?php include"head.php";?>
<script type="text/javascript" src="js/nicEdit-latest.js"></script>
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.19/css/jquery.dataTables.css">

<script type="text/javascript" charset="utf8"
src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.js"></script>


  <script type="text/javascript">
  $(document).ready(function () {
$('#dtBasicExample').DataTable();
$('.dataTables_length').addClass('bs-select');
});

  </script>
  <style>
  table.dataTable thead>tr>td.sorting,
table.dataTable thead>tr>td.sorting_asc,
table.dataTable thead>tr>td.sorting_desc,
table.dataTable thead>tr>th.sorting,
table.dataTable thead>tr>th.sorting_asc,
table.dataTable thead>tr>th.sorting_desc {
  padding-right: 30px
}

table.dataTable thead .sorting,
table.dataTable thead .sorting_asc,
table.dataTable thead .sorting_asc_disabled,
table.dataTable thead .sorting_desc,
table.dataTable thead .sorting_desc_disabled {
  cursor: pointer;
  position: relative
}

table.dataTable thead .sorting:after,
table.dataTable thead .sorting:before,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_asc_disabled:after,
table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc:after,
table.dataTable thead .sorting_desc:before,
table.dataTable thead .sorting_desc_disabled:after,
table.dataTable thead .sorting_desc_disabled:before {
  position: absolute;
  bottom: .9em;
  display: block;
  opacity: .3
}

table.dataTable thead .sorting:before,
table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc:before,
table.dataTable thead .sorting_desc_disabled:before {
  right: 1em;
  content: "\f0de";
  font-family: FontAwesome;
  font-size: 1rem
}

table.dataTable thead .sorting:after,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_asc_disabled:after,
table.dataTable thead .sorting_desc:after,
table.dataTable thead .sorting_desc_disabled:after {
  content: "\f0dd";
  font-family: FontAwesome;
  right: 16px;
  font-size: 1rem
}

table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_desc:after {
  opacity: 1
}

table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc_disabled:after {
  opacity: 0
}
  </style>
```

```
 </head>
<body>
<div class="page-container">
<div class="left-content">
<div class="inner-content">
<?php //include"header.php";?>
<div class="outter-wp">
<div class="sub-heard-part">
<ol class="breadcrumb m-b-0">
<li><a href="index.html">Home</a></li>
<li class="active">Feedback</li>
</ol>
</div>
<div class="graph-visual tables-main">

<div class="graph">
<div class="block-page">
<p>
<h3 class="inner-tittle two">Feedback</h3>
<div class="form-body">
<table id="dtBasicExample" class="table table-striped table-bordered" cellspacing="0"
width="100%">
  <thead>
    <tr>
            <th class="th-sm">S.no.
      </th>
      <th class="th-sm">Name
      </th>
      <th class="th-sm">Email</th>
      <th class="th-sm">Subject</th>
      <th class="th-sm">message</th>


    </tr>
  </thead>
  <tbody>
  <?php
  $result=select("select * from feedback");
  $n=1;
  while($r=mysqli_fetch_array($result))
  {  extract($r);
  ?>
    <tr>
      <td><?=$n?></td>
      <td><?=ucwords($name)?></td>
      <td><?=$email?></td>
      <td><?=$subject?></td>
      <td><?=$message?></td>
    </tr>

    <?php
            $n++;
  }
            ?>


    </tbody>

</table>
</div>


</p>
</div>

</div>

</div>
</div>
<?php include"footer.php";?>

</div>
</div>
<?php include"side_bar.php";?>
</div>
<?php include"footer_script.php";?>

</body>
</html>
```

## 11.10 View all crimes

```
<?php require_once"dbconfig.php";
if(isset($_SESSION['login']))
{


}
else
{
            header("location:login.php");
}

?>
<!DOCTYPE HTML>
<html>
<head>
<?php include"head.php";?>
```

```
<script type="text/javascript" src="js/nicEdit-latest.js"></script>
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.19/css/jquery.dataTables.css">

<script type="text/javascript" charset="utf8"
src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.js"></script>


  <script type="text/javascript">
  $(document).ready(function () {
$('#dtBasicExample').DataTable();
$('.dataTables_length').addClass('bs-select');
});

  </script>
  <style>
  table.dataTable thead>tr>td.sorting,
table.dataTable thead>tr>td.sorting_asc,
table.dataTable thead>tr>td.sorting_desc,
table.dataTable thead>tr>th.sorting,
table.dataTable thead>tr>th.sorting_asc,
table.dataTable thead>tr>th.sorting_desc {
  padding-right: 30px
}

table.dataTable thead .sorting,
table.dataTable thead .sorting_asc,
table.dataTable thead .sorting_asc_disabled,
table.dataTable thead .sorting_desc,
table.dataTable thead .sorting_desc_disabled {
  cursor: pointer;
  position: relative
}

table.dataTable thead .sorting:after,
table.dataTable thead .sorting:before,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_asc_disabled:after,
table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc:after,
table.dataTable thead .sorting_desc:before,
table.dataTable thead .sorting_desc_disabled:after,
table.dataTable thead .sorting_desc_disabled:before {
  position: absolute;
  bottom: .9em;
  display: block;
  opacity: .3
}
```

```
table.dataTable thead .sorting:before,
table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc:before,
table.dataTable thead .sorting_desc_disabled:before {
  right: 1em;
  content: "\f0de";
  font-family: FontAwesome;
  font-size: 1rem
}

table.dataTable thead .sorting:after,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_asc_disabled:after,
table.dataTable thead .sorting_desc:after,
table.dataTable thead .sorting_desc_disabled:after {
  content: "\f0dd";
  font-family: FontAwesome;
  right: 16px;
  font-size: 1rem
}

table.dataTable thead .sorting_asc:before,
table.dataTable thead .sorting_desc:after {
  opacity: 1
}

table.dataTable thead .sorting_asc_disabled:before,
table.dataTable thead .sorting_desc_disabled:after {
  opacity: 0
}
  </style>
  </head>
<body>
<div class="page-container">
<div class="left-content">
<div class="inner-content">
<?php //include"header.php";?>
<div class="outter-wp">
<div class="sub-heard-part">
<ol class="breadcrumb m-b-0">
<li><a href="index.html">Home</a></li>
<li class="active">Crime List</li>
</ol>
</div>
<div class="graph-visual tables-main">

<div class="graph">
<div class="block-page">
<p>
```

```
<h3 class="inner-tittle two">All Crime List</h3>
<div class="form-body">
<table id="dtBasicExample" class="table table-striped table-bordered" cellspacing="0"
width="100%">
  <thead>
    <tr>
            <th class="th-sm">S.no.
      </th>
      <th class="th-sm"> Name
      </th>
      <th class="th-sm">City</th>
      <th class="th-sm">Mobile</th>
      <th class="th-sm">Complaint Type</th>
      <th class="th-sm">Date</th>
      <th class="th-sm">Evidence</th>
      <th class="th-sm"> View All Detail</th>


    </tr>
  </thead>
  <tbody>
  <?php
  $result=select("SELECT * FROM `crime` ");
  $n=1;
  while($r=mysqli_fetch_array($result))
  {  extract($r);
  ?>
    <tr>
      <td><?=$n?></td>
      <td><?=ucwords($name)?></td>
      <td><?=$city?></td>

      <td><?=$mobile?></td>
      <td><?=$com_type?></td>
      <td><?=$date_d?></td>
      <td><a href="../img/<?=$evidence?>"><button class="btn btn-
warning">Download</button></a></td>
      <td><a href="view_crime_form.php?id=<?=$id?>"><button class="btn btn-
success">View</button></td>



    </tr>

    <?php
            $n++;
  }
            ?>
    </tbody>

</table>
</div>


</p>
</div>

</div>

</div>
</div>
<?php include"footer.php";?>

</div>
</div>
<?php include"side_bar.php";?>
</div>
<?php include"footer_script.php";?>

</body>
</html>
```

## 11.11 Index

```
<?php require_once"dbconfig.php";
if(isset($_SESSION['login']))
{

}
else
{
        header("location:login.php");
}

?>
<!DOCTYPE HTML>
<html>
<?php include"head.php";?>
<body>
<div class="page-container">
<div class="left-content">
<div class="inner-content">
<?php //include"header.php";?>
<div class="outter-wp">
<div class="sub-heard-part">
<ol class="breadcrumb m-b-0">
<li><a href="index.html">Home</a></li>
<li class="active">Blank Page</li>
```

```
</ol>
</div>
<div class="graph-visual tables-main">
<h2 class="inner-tittle">Blank Page</h2>
<div class="graph">
<div class="block-page">
<p>Welcome
<?php
if(isset($_SESSION['name']))
{
        $name=$_SESSION['name'];
        echo"$name";
}

?>

</p>
</div>

</div>

</div>
</div>
<?php include"footer.php"?>
</div>
</div>
<?php include"side_bar.php";?>
</div>
<?php include"footer_script.php";?>
</body>
</html>
```

## 11.12 FIR from view

```
<?php require_once"dbconfig.php";
if(isset($_SESSION['login']))
{


}
else
{
        header("location:login.php");
}

?>
<!DOCTYPE HTML>
<html>
<head>
<?php include"head.php";?>
<script type="text/javascript" src="js/nicEdit-latest.js"></script>
```

```
</head>
<body>
<div class="page-container">
<div class="left-content">
<div class="inner-content">
<?php //include"header.php";?>
<div class="outter-wp">
<div class="sub-heard-part">
<ol class="breadcrumb m-b-0">
<li><a href="index.html">Home</a></li>
<li class="active">add Team</li>
</ol>
</div>
<div class="graph-visual tables-main">
<h2 class="inner-tittle">FIRST INFORMATION REPORT (FIR)</h2>
<div class="graph" style="height:1000px">
<div class="block-page">
<p>
<h3 class="inner-tittle two" style="font-weight:bold">F I R</h3>
<div class="form-body">
<?php
$res=select("select * from fir where crimeid='".$_REQUEST['id']."'");
while($r=mysqli_fetch_array($res))
{extract($r);
?>
<form class="form-horizontal"  method="post" enctype="multipart/form-data">

<div class="form-group"><label for="inputPassword3" id="errormetakey" class="col-sm-2
control-label"></label>

<div class="col-xs-6">
Date And Hour Of Occurance
<input type="text" class="form-control" value="<?=$daho?>" id="title" name='daho' >
</div>
</div>


<div class="form-group"> <label for="inputPassword3" id="errormetakey" class="col-sm-
2 control-label"></label>

<div class="col-xs-6">
Date And Hour Of Reported<input type="text" class="form-control"  value="<?=$dahr?>"
name="dahr">
</div>
</div><div class="form-group">
<label for="inputPassword3" id="errormetakey" class="col-sm-2 control-label"></label>
<div class="col-xs-6">
Name And Residance Of Informer And Complain<input type="text" class="form-control"
value="<?=$hric?>"   name="nric">
```

```
</div>
</div>
<div class="form-group">
<label for="inputPassword3" id="errormetakey" class="col-sm-2 control-label"></label>
<div class="col-xs-6">
Brief Description Of Offence And Of Property Carried Off ,If Any<input type="text"
value="<?=$brief?>"  class="form-control"   name="brief">
</div>
</div>
<div class="form-group">
<label for="inputPassword3" id="errormetakey" class="col-sm-2 control-label"></label>
<div class="col-xs-6">
Place Of Occurance<input type="text" class="form-control"   value="<?=$place?>"
name="place">
</div>
</div>
<div class="form-group">
<label for="inputPassword3" id="errormetakey" class="col-sm-2 control-label"></label>
<div class="col-xs-6">
Name And Address Of Criminal<input type="text" class="form-control"
value="<?=$address?>"   name="address">
</div>
</div>
<div class="form-group">
<label for="inputPassword3" id="errormetakey" class="col-sm-2 control-label"></label>
<div class="col-xs-6">
Steps Taken Regarding Investigation Explanation Of Delay In Regarding
Information<input type="text" class="form-control"   name="explanation"
value="<?=$explanation?>"  >
</div>
</div>


</div>
<div class="col-xs-6">
<center><input type="submit" class="btn btn-info" name="submit" id="elec_submit"
value="Update"></center> </div> </form>
<?php
}
?>
<?php
if(isset($_REQUEST['submit']))
{
extract($_REQUEST);

        $n=iud("UPDATE  `fir` set  `daho`='$daho', `dahr`='$dahr', `hric`='$nric',
`brief`='$brief',
        `place`='$place', `address`='$address', `explanation`='$explanation' where
crimeid='".$_REQUEST['id']."'");
         if($n==1)
```

```
           {
                      echo"<script>alert(' Added successfully');
                      window.location='view_crime_fir.php';
                      </script>";
           }


          else
          {
                      echo"<script>alert(' Something Went Wrong');
                      window.location='view_crime_fir.php';
                      </script>";
          }




}
?>


</div>

</p>
</div>

</div>

</div>
</div>
<?php include"footer.php";?>

</div>
</div>
<?php include"side_bar.php";?>
</div>
<?php include"footer_script.php";?>
<!--<script>

$(document).ready(function(){

$("#project_submit").click(function(){

var valid=true;
var title=$.trim($("#title").val());
var metakey=$.trim($("#metakey").val());
var metadis=$.trim($("#metadis").val());

if(title.length<6)
```

```
{
$("#errortitle").html('Invalid title');
$("#errortitle").css("color","red");
$("#title").css("border-color","red");
valid=false;
}
else
{
$("#errortitle").html('Title');
$("#errortitle").css("color","black");
$("#title").css("border-color","#ddd");
}
if(metakey.length<6)
{
$("#newpassworderror").html('Invalid New Password');
$("#newpassworderror").css("color","red");
$("#newpassword").css("border-color","red");
valid=false;
}
else
{
$("#newpassworderror").html('New Password');
$("#newpassworderror").css("color","black");
$("#newpassword").css("border-color","#ddd");
}
if(metadis.length<6)
{
$("#newpassworderror").html('Invalid New Password');
$("#newpassworderror").css("color","red");
$("#newpassword").css("border-color","red");
valid=false;
}
else
{
$("#newpassworderror").html('New Password');
$("#newpassworderror").css("color","black");
$("#newpassword").css("border-color","#ddd");
}



var mymethod="post";
var myurl="myphp.php";
var
mydata="oldpassword="+oldpassword+"&newpassword="+newpassword+"&cpassword="
+cpassword+"&change=yes";

$.ajax({

        method:mymethod,
          url:myurl,
          data:mydata,
          success:function(result)
          {
                    if(result==1)
                    {
                alert("Password Changed Successfully");
                              $("#oldpassword").val("");
                              $("#newpassword").val("");
                              $("#cpassword").val("");
                    }
                    else
                    {
                              alert(result);
                    }


          }
});
return false;
});
});

</script>-->
</body>
</html>
```

# 12. BIBLIOGRAPHY

- Allen (2002), “Meaning of Crimes”, Crime Management
- Lyon (2005), “Global Positioning System” Location Tracker
- Black (1984), “Crime as social control”, in Toward a General Theory of Social Control, vol. 2, Selected Problems, ed.
- Robert D. Crutchfield, George S. Bridges, and Susan R.Pitchford, “Analytical and Aggregation Biases in Analyses of Imprisonment: Reconciling Discrepancies in Studies of Racial Disparity,”Journal of Research in Crime
- Skogan (1984), “Reporting crimes to the police: The status of world research”, Journal of Research in Crime and Delinquency, vol. 21, pp. 113-37.
- The Nigeria Dailies, (2009), “The Historical Background of Nigerian Police”
- Zender (2003) 'Too Much Security?', International Journal of the Sociology of Law, 31 (3): 155-184.