GRIEVANCES REGISTRATION AND MANAGEMENT SYSTEM

IET REPORT

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF SCIENCE

IN

COMPUTER TECHNOLOGY

Submitted by

HAFIL A

22BCT017

Under the Guidance of

Ms. S SRISOWMIYA M.Sc.,

Assistant Professor

Department of Computer Technology and Data Science

SRI KRISHNA ARTS AND SCIENCE COLLEGE

(An Autonomous Institutions)

Accredited By NAAC with 'A' Grade

Coimbatore - 641008

JULY 2024

DECLARATION

SRI KRISHNA ARTS AND SCIENCE COLLEGE

Accredited By NAAC with 'A' Grade

Kuniamuthur, Coimbatore - 641008

DECLARATION

I hereby declare that the IET report entitled **"GRIEVANCES REGISTRATION AND MANAGEMENT SYSTEM"** submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Science in Computer Technology** is an original work submitted and it has not been previously formed the basis for the award of any other Degree, Diploma, Associate ship, Fellowship or similar titles to any other university or body during the period of my study.

Place: Coimbatore

Date: 24 /07/2024

[illegible]

SIGNATURE OF THE STUDENT

CERTIFICATE

SRI KRISHNA ARTS AND SCIENCE COLLEGE

Accredited By NAAC with 'A' Grade

Kuniamuthur, Coimbatore - 641008

CERTIFICATE

This is to certify that the IET report entitled **"GRIEVANCES REGISTRATION AND MANAGEMENT SYSTEM"** in partial fulfilment of requirements for the award of the degree of Bachelor of Science in Computer Technology is a record of bonafide work carried out by **HAFIL A (22BCT017)** and that no part of this has been submitted for the award of any other degree or diploma and the work has not been published in popular journal or magazine.

S. Sri Sowmiya [illegible] 28/2/24 [illegible] 28/2/24

GUIDE **HoD** **DEAN**

This Industrial Exposure Training Report is submitted for the viva voce conducted on 27/07/2024 at Sri Krishna Arts and Science College.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I am ineffably indebted to **Dr. K. Sundararaman, M.Com., M.Phil., Ph.D., Chief Executive Officer**, Sri Krishna Institutions, Coimbatore.

I convey my profound gratitude to **Dr. R. Jagajeevan, MBA., M.Phil., Ph.D., Principal**, Sri Krishna Arts and Science College for giving permission for the fulfilment of the venture.

It is my prime to solemnly express my sense of gratitude to **Dr. K. S. Jeen Marseline, MCA., M.Phil., Ph.D., Dean, Computer Science and Mathematics**, Sri Krishna Arts and Science College.

It is my prime to solemnly express my sense of gratitude to **Dr. B. Radha, MCA., Ph.D., Associate Professor & Head, Department of Computer Technology and Data Science**, Sri Krishna Arts and Science College.

I would like to extend my thanks and unbound sense for the timely help and assistance given by **Ms. S Srisowmiya, M.Sc., Assistant Professor, Department of Computer Technology and Data Science**, Sri Krishna Arts and Science College in completing the IET Project work. Her remarkable guidance at every stage of my work was coupled with suggestion and motivation.

I owe my gratitude and thanks to Mr.Eswaran, Live Stream Technologies for permitting me to carry out the training.

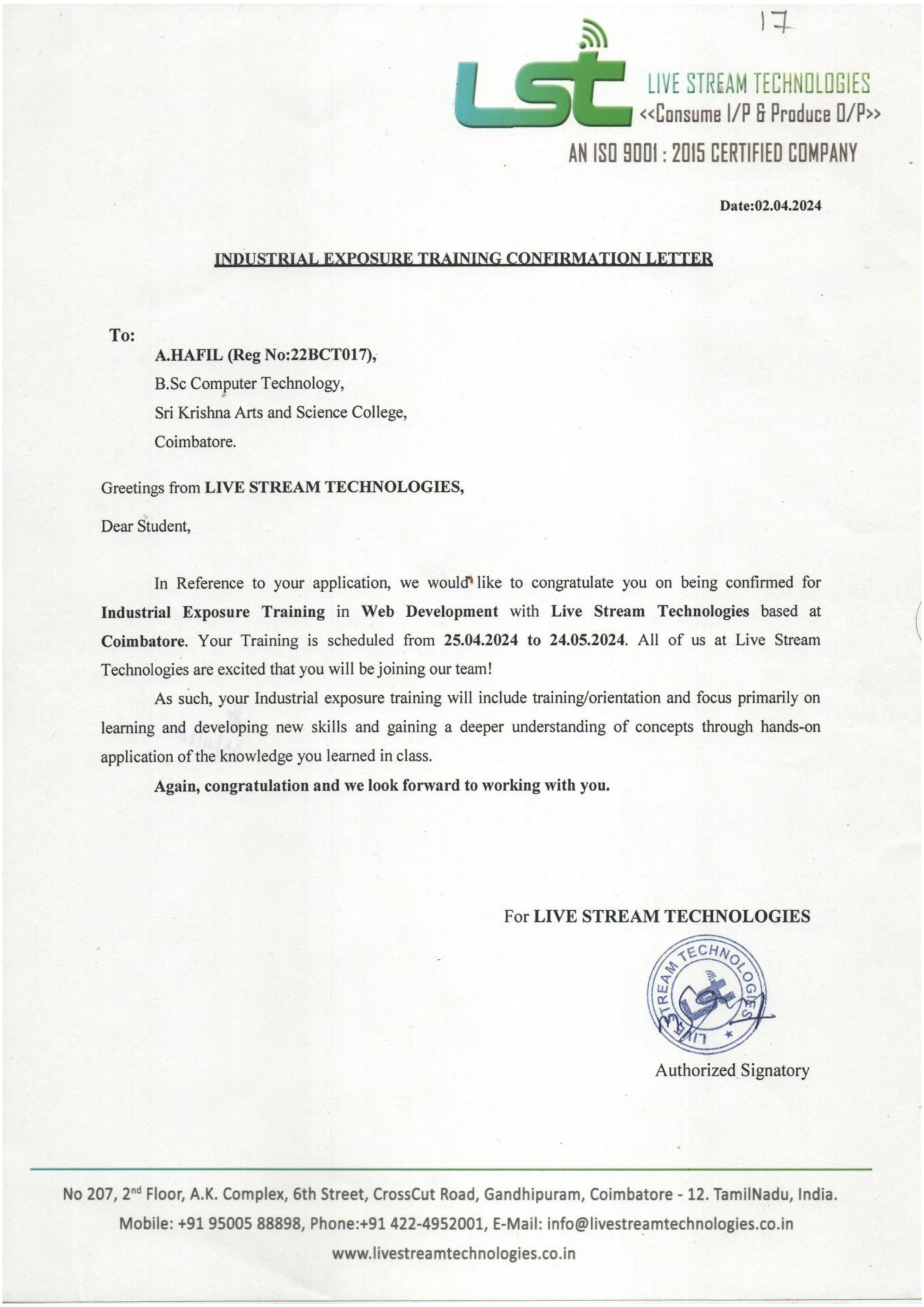
I take this opportunity to thank my parents and friends for their constant support and encouragement throughout this training.

HAFIL A

22BCT017

IET ACCEPTANCE LETTER

IET ACCEPTANCE LETTER



17

LIVE STREAM TECHNOLOGIES
<<Consume I/P & Produce O/P>>
AN ISO 9001 : 2015 CERTIFIED COMPANY

Date:02.04.2024

INDUSTRIAL EXPOSURE TRAINING CONFIRMATION LETTER

To:

A.HAFIL (Reg No:22BCT017),
B.Sc Computer Technology,
Sri Krishna Arts and Science College,
Coimbatore.

Greetings from **LIVE STREAM TECHNOLOGIES,**

Dear Student,

In Reference to your application, we would like to congratulate you on being confirmed for **Industrial Exposure Training** in **Web Development** with **Live Stream Technologies** based at **Coimbatore**. Your Training is scheduled from **25.04.2024 to 24.05.2024**. All of us at Live Stream Technologies are excited that you will be joining our team!

As such, your Industrial exposure training will include training/orientation and focus primarily on learning and developing new skills and gaining a deeper understanding of concepts through hands-on application of the knowledge you learned in class.

Again, congratulation and we look forward to working with you.

For **LIVE STREAM TECHNOLOGIES**

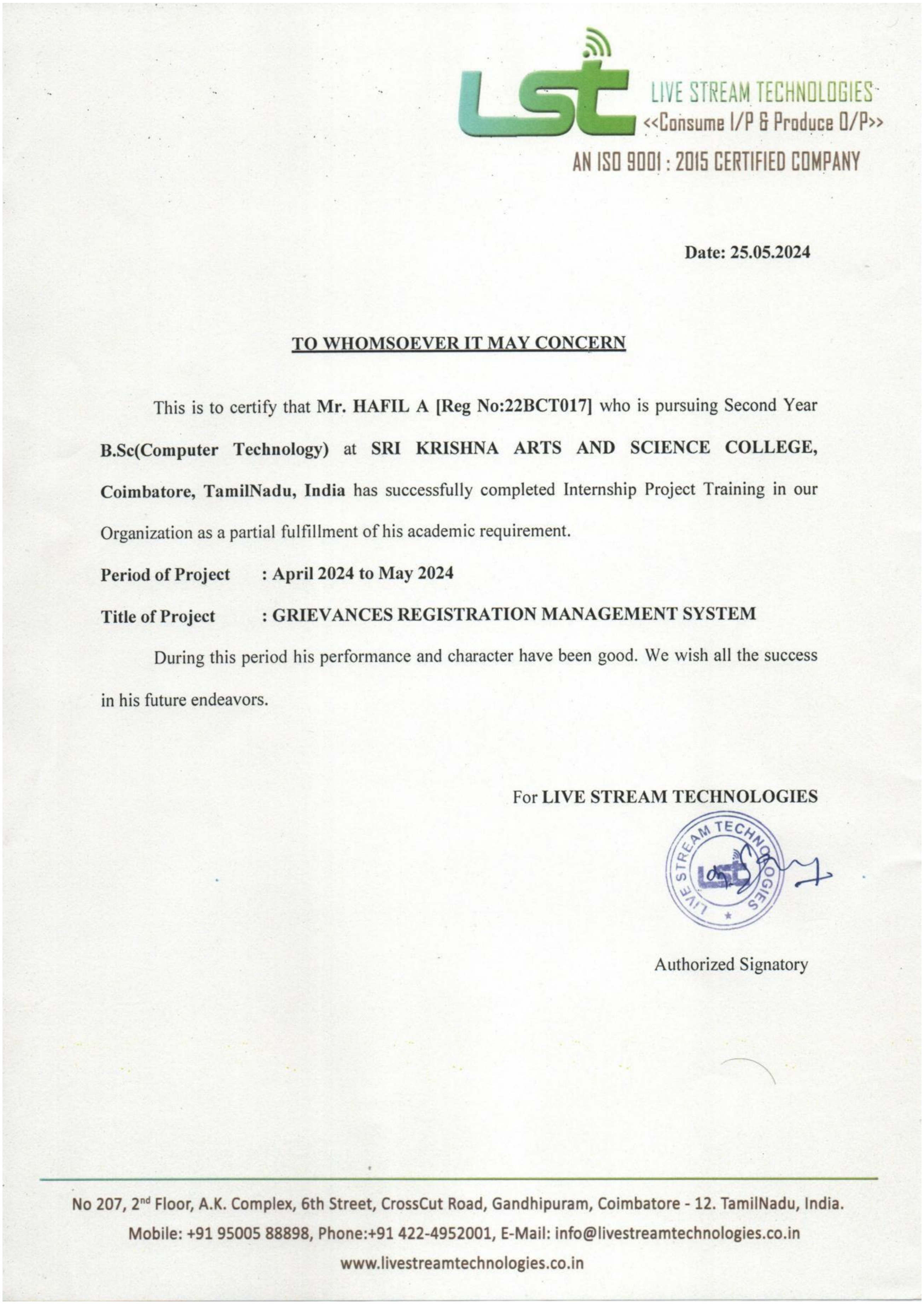
LIVE STREAM TECHNOLOGIES ★

Authorized Signatory

No 207, 2nd Floor, A.K. Complex, 6th Street, CrossCut Road, Gandhipuram, Coimbatore - 12. TamilNadu, India.
Mobile: +91 95005 88898, Phone:+91 422-4952001, E-Mail: info@livestreamtechnologies.co.in
www.livestreamtechnologies.co.in

PROJECT COMPLETION CERTIFICATE

PROJECT COMPLETION CERTIFICATE



LIVE STREAM TECHNOLOGIES
<<Consume I/P & Produce O/P>>
AN ISO 9001 : 2015 CERTIFIED COMPANY

Date: 25.05.2024

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. HAFIL A [Reg No:22BCT017]** who is pursuing Second Year **B.Sc(Computer Technology)** at **SRI KRISHNA ARTS AND SCIENCE COLLEGE, Coimbatore, TamilNadu, India** has successfully completed Internship Project Training in our Organization as a partial fulfillment of his academic requirement.

Period of Project	**: April 2024 to May 2024**
Title of Project	**: GRIEVANCES REGISTRATION MANAGEMENT SYSTEM**

During this period his performance and character have been good. We wish all the success in his future endeavors.

For **LIVE STREAM TECHNOLOGIES**

LIVE STREAM TECHNOLOGIES

[illegible]

Authorized Signatory

No 207, 2nd Floor, A.K. Complex, 6th Street, CrossCut Road, Gandhipuram, Coimbatore - 12. TamilNadu, India.
Mobile: +91 95005 88898, Phone:+91 422-4952001, E-Mail: info@livestreamtechnologies.co.in
www.livestreamtechnologies.co.in

TABLE OF CONTENTS

TABLE OF CONTENTS

ORGANISATION PROFILE

ORGANIZATION PROFILE

LIVE STREAM TECHNOLOGIES
<<Consume I/P & Produce O/P>>

Live Stream Technologies (LST) is a software development company headquartered in Coimbatore, India. The cornerstone of the company's outsourcing strategy is a balancing combination of onsite presence and service delivery.

Live Stream Technologies is a leading software development company, that is into customized software applications, and website development services based on a range of platforms and technologies.

Live Stream Technologies provides Web designing, Application Development, and Maintenance Outsourcing services that lead to business process improvement. This allows for the reduction of costs and enables business growth. **Live Stream's** Application Development and Maintenance Services is a part of its IT Services Group.

Vision:

Integrity - Honesty in how we deal with our clients, each other, and with the world Candor - Be open and upfront in all our conversations. Keep clients updated on the real situation. Deal with situations early; avoid last-minute surprises Service - Seek to empower and enable our clients. Consider ourselves successful not when we deliver our client's final product but when the product is launched and meets success Kindness - Go the extra mile. Speak the truth with grace. Deliver more than is expected or promised Competence - Benchmark with the best in the business. Try new and better things. Never rest on laurels. Move out of your comfort zone. Keep suggesting new things. Seek to know more.

Mission:

LIVE STREAM Technologies' mission is “To assist the company’s progression and mechanize their in sequence stream” and to reach out to the global markets and create innovative, world-class software solutions that match to International quality standards.

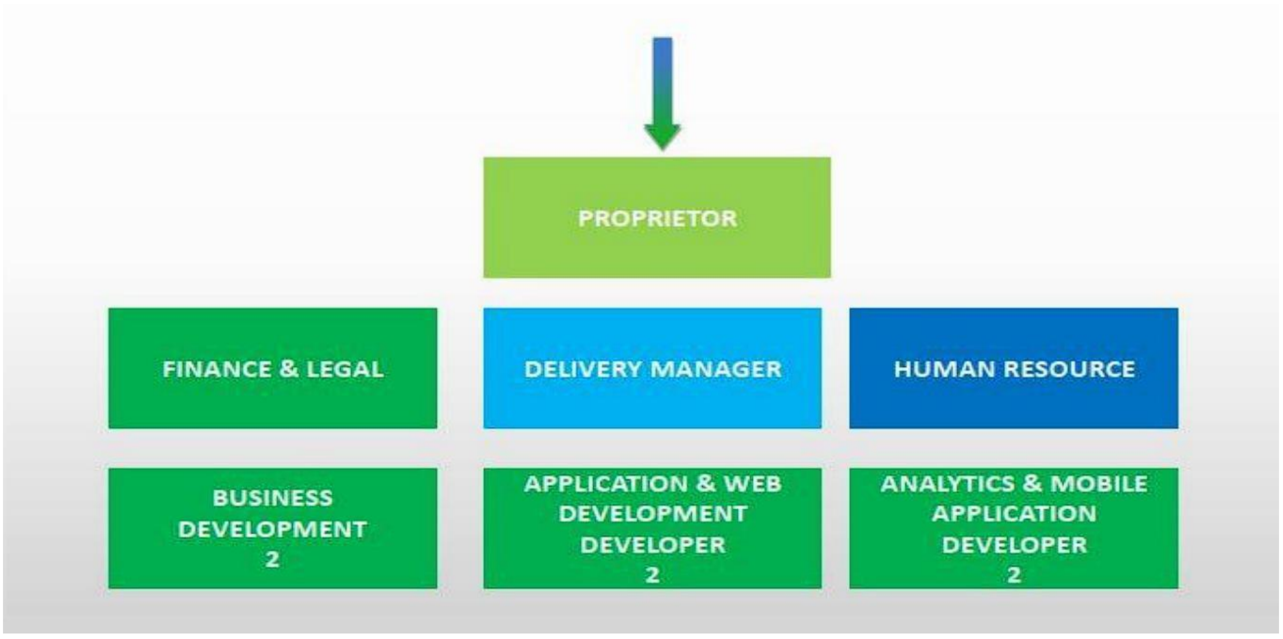
LST Team:

LST is proud of its company team. LST believes that people are the key to successfully providing high-level offshore software development services and more importantly, an ongoing relationship with our clients. LST has a clear policy of employing only the very best in their field. The team is composed of developers who have significant experience in the Education, Healthcare, Transportation, Hospitality, Manufacturing, Real estate, Retail, and Technology sectors. “Initially launched as a pure offshore software development concern, LST has tremendously grown over the years with the changing global marketplace”.

Visionary & Dynamic Management:

LST Management is passionate about its operations & team. Short years have led to a long track record and ever-green relationships with its clients. The company’s management has experience in delivering projects to several companies. They have managed the team’s people effectively delivering projects on time and to stipulated norms.

Organization Structure:


PROPRIETOR
FINANCE & LEGAL
DELIVERY MANAGER
HUMAN RESOURCE
BUSINESS DEVELOPMENT 2
APPLICATION & WEB DEVELOPMENT DEVELOPER 2
ANALYTICS & MOBILE APPLICATION DEVELOPER 2

ABSTRACT

ABSTRACT

This project aims to empower the public by providing detailed locality information and an efficient online platform to resolve civic issues without frequent visits to government offices. The system, accessible via Android and web applications, focuses on registering, tracking, and resolving complaints, thereby promoting transparency and reducing corruption. It addresses common issues like street light malfunctions, water pipe leaks, rainwater drainage problems, road reconstruction, and garbage management.

Currently, addressing these issues involves cumbersome manual processes, requiring citizens to visit multiple offices and follow up repeatedly. This consumes significant time and effort and opens the door to potential corruption. By automating the complaint management system, the project enhances efficiency and responsiveness. Users can register complaints through a simple form, attach relevant photos or documents, and receive a unique tracking number for real-time updates.

The user-friendly interface includes GPS integration for location detection, push notifications for updates, and a feedback system to rate the resolution process. Multilingual support ensures accessibility for India's diverse population. For government authorities, the backend system allows efficient complaint management and resource allocation, generating analytical reports to identify recurring issues and plan long-term solutions.

Residents of housing schemes across India can report concerns and track complaints, improving public service delivery and citizen satisfaction. The application reduces the need for physical office visits, saving time and reducing corruption opportunities. Additionally, it fosters community engagement by allowing users to view and support neighbour's complaints and disseminate important information.

In summary, this project modernizes public service delivery, promotes good governance, and builds trust between citizens and authorities, making cities and towns across India cleaner, safer, and more reliable.

INTRODUCTION

CHAPTER – 1

INTRODUCTION

The Grievance Registration and Management System (GMS) is a web-based application designed to provide an efficient and effective way to address and resolve issues faced by the public and organizations. This application aims to streamline the process of recording, resolving, and responding to grievances, ensuring that users receive timely and appropriate responses to their concerns.

The primary goal of developing GMS is to simplify the coordination, monitoring, tracking, and resolution of grievances. By prioritizing and addressing problem areas, the system helps organizations enhance their operations and improve user satisfaction.

One of the significant advantages of GMS is the transition from manual record-keeping to a digital platform, which increases accuracy and efficiency while reducing the time constraints typically associated with manual processes.

GMS offers multiple channels for users to file complaints, making it easier for them to report issues. The web application allows users to submit grievances conveniently and provides a platform for progress reporting. This feature ensures that users can stay informed about the status of their complaints and receive updates in real time.

Additionally, the application classifies complaints and directs them to the appropriate department, streamlining the resolution process. This automated system reduces the need for extensive operational staff and minimizes the time required to address grievances. Employees can manage and resolve complaints through the web application, eliminating the need for paper-based forms and manual handling.

Overall, GMS enhances the current grievance management system by leveraging web technology to provide a more efficient, accurate, and user-friendly solution. It enables organizations to better manage grievances, improve their response times, and ultimately deliver a higher level of service to their users.

SYSTEM STUDY

CHAPTER – 2

SYSTEM STUDY

2.1 EXISTING SYSTEM

Traditional grievance management systems, being manual, face numerous inefficiencies. They are notably time-consuming, relying heavily on extensive paperwork and manual processes at every stage, which delays grievance resolution and complicates tracking and monitoring.

Such systems demand more resources, including significant manpower, to manage and resolve complaints, escalating operational costs and straining organizational resources. Human intervention at every step also introduces a higher risk of errors and oversight.A critical issue with manual systems is the potential for user problems to go unresolved. Complaints can be misplaced, overlooked, or not addressed promptly, leading to user dissatisfaction and issue escalation. The absence of a structured and transparent process makes efficient grievance handling challenging.

Additionally, management may forget or overlook user complaints without a centralized, automated tracking mechanism, making it difficult to monitor grievance statuses and ensure timely follow-up. This can result in unresolved issues accumulating and users feeling neglected.

The Grievance Registration and Management System (GMS) overcomes these shortcomings by digitizing the entire process. It automates the recording, tracking, and resolution of grievances, ensuring systematic management and timely resolution. By reducing reliance on manual intervention, GMS enhances accuracy, efficiency, and user satisfaction while minimizing resource requirements and the risk of oversight.

2.2 PROPOSED SYSTEM

We want to replace the existing manual Grievances Registration and Management System with an application that will transform the way we handle society's complaints, proving beneficial by improving efficiency and saving time.

Using this application, people can register their complaints easily and in a proper format. They will also be well aware of the progress of their complaint. Additionally, they can provide feedback on the progress of their complaints, indicating whether they are satisfied or not.

Furthermore, users can post their requirements through this system and receive the needed items from the admin within a couple of hours, depending on the item. Users can also track the status of their requirements. These user complaints and requirements are managed by the admin, who can view and address the feedback posted by users on the GMS system.

An additional point is that the application can generate detailed reports and analytics on complaint trends, response times, and user satisfaction levels. This data can help the organization identify common issues, measure performance, and implement improvements to enhance service quality continuously.

2.3 SOFTWARE REQUIREMENTS

Hardware Configuration

Processor	: Core i3
RAM	: 2GB
Speed	: 1.3GHz or above
Hard Disk	: 250 GB
I/O Devices	: Standard Keyboard & Logitech Mouse

Software Configuration

Front-End	: HTML
Client-Side	: JAVA SCRIPT
Server-Side	: PYTHON
Back-End	: MYSQL
Browser	: IE 6.0 or Later, Firefox, Chrome

SOFTWARES USED:

APACHE SERVER 2.2
PYTHON 3.0 AND ABOVE
MYSQL 5.1

EXTERNAL MODULE INTERFACED:

PY MySQL

Front End:

HTML

Hypertext Mark-up Language is a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.

Client-Side Scripting:

JAVA SCRIPT

Java Script is a programming language commonly used in web development. It was originally developed by Netscape to add dynamic and interactive elements to websites. ... Like server-side scripting languages, such as PYTHON, PHP, and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage.

Server-Side Scripting:

PYTHON

Python is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. Python is one of the top 10 popular programming languages of 2017. The simple syntax rules of the programming language further make it easier for you to keep the code base readable and the application maintainable. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Since most modern OS is written in C, compilers/interpreters for modern high-level languages are also written in C. Python is not an exception - its most popular/"traditional" implementation is called C Python and is written in C. Python is a general-purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. All general-purpose languages can be used at both the front end and back end. Java, C#, C, C++, etc. are general-purpose languages. It can be used both as a back end and a front end. API's using Django/Flask can be made as back end. Python is a general-purpose language, which means it can be used to build just about anything, which will be made easy with the right tools/libraries. Professionally, Python is great for backend web development, data analysis, artificial intelligence, and scientific computing.

Applications of Python is used to simplifying the complex software development process as it is a general-purpose programming language. It is used for developing the complex application like scientific and numeric application, and for both desktop and web applications. You can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high-level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks.

BENEFITS OF PYTHON

- Interactive
- Interpreted
- Modular
- Dynamic
- Object-oriented
- Portable
- High level
- Extensible in C++ & C

FEATURES OF PYTHON

1) **Readable and Maintainable Code:** While writing a software application, you must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes code readability and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting in extra time and effort.
2) **Multiple Programming Paradigms:** Like other modern programming languages, Python also supports several programming paradigms. It supports object-oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.
3) **Compatible with Major Platforms and Systems:** At present, Python supports many operating systems. You can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted programming language.

It allows you to run the same code on multiple platforms without recompilation. Hence, you are not required to recompile the code after making any alterations. You can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to make changes to the code without increasing development time.

4) **Robust Standard Library:** Its large and robust standard library makes Python score over other programming languages. The standard library allows you to choose from a wide range of modules according to your precise needs. Each module further enables you to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, you can use specific modules to implement web services, perform string operations, manage operating system interfaces or work with internet protocols. You can even gather information about various modules by browsing through the Python Standard Library documentation.

5) **Many Open Source Frameworks and Tools:** As an open-source programming language, Python helps you to curtail software development costs significantly. You can even use several open-source Python frameworks, libraries, and development tools to curtail development time without increasing development costs. You even have the option to choose from a wide range of open-source Python frameworks and development tools according to your precise needs. For instance, you can simplify and speed up web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle, and Cherrypy. Likewise, you can accelerate desktop GUI application development using Python GUI frameworks and toolkits like PyQT, PyJ s, PyGUI, Kivy, PyGTK, and WxPython.

6) **Simplify Complex Software Development:** Python is a general-purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. You can take advantage of the data analysis features of Python to create custom big data solutions without putting in extra time and effort. At the same time, the data visualization libraries and APIs provided by Python help you to visualize and present data in a more appealing and effective way. Many Python developers even use Python to accomplish artificial intelligence (AI) and natural language processing tasks.

7) **Adopt Test-Driven Development**: You can use Python to create a prototype of the software application rapidly. Also, you can build the software application directly from the prototype simply by refactoring the Python code. Python even makes it easier for you to perform coding and testing simultaneously by adopting test test-driven development (TDD) approach. You can easily write the required tests before writing code and use the tests to assess the application code continuously. The tests can also be used to check if the application meets predefined requirements based on its source code.

ADVANTAGES OF PYTHON

1) **Extensive Support Libraries:** It provides large standard libraries that include areas like string operations, Internet, web service tools, operating system interfaces, and protocols. Most of the highly used programming tasks are already scripted into it which limits the length of the codes to be written in Python.

2) **Integration Feature:** Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Jython. Python also processes XML and other mark-up languages as it can run on all modern operating systems through the same byte code.

3) **Improved Programmer's Productivity:** The language has extensive support libraries and clean object-oriented designs that increase two to tenfold of the programmer's productivity while using languages like Java, VB, Perl, C, C++, and C#.

4) **Productivity:** With its strong process integration features, unit testing framework and enhanced control capabilities contribute towards the increased speed for most applications and productivity of applications. It is a great option for building scalable multi-protocol network applications.

Back End:

MySQL

MySQL is a powerful database management system that allows users to create applications with little or no programming.

It supports GUI features and an entire programming language, PhpMyAdmin, for developing richer and more robust applications.There are several reasons to choose MySQL. Firstly, MySQL is feature-rich and capable of handling any database-related task. You can create places to store your data, build tools to read and modify database contents easily and query your data effectively.

MySQL is a relational database, meaning it stores information about related objects. In MySQL, this translates to a collection of tables that hold data. It also stores other related objects such as queries, forms, and reports, collectively enabling effective functionality. As a back-end database for PHP as a front-end, MySQL supports powerful database management functions. Beginners can create their own databases with simple mouse clicks.

Another compelling reason to use MySQL is its status as a component of popular open-source software. MySQL is an open-source database management system used by many of the most frequently visited websites, including Flickr, Nokia.com, YouTube, Wikipedia, Google, Facebook, and Twitter.

MySQL is the world's most used open-source relational database management system, running as a server providing multi-user access to multiple databases. Free-software open-source projects often use MySQL, while several paid editions are available for commercial use, offering additional functionality.

Applications using MySQL databases include TYPO3, Joomla, WordPress, phpBB, MyBB, Drupal, and other software built on the LAMP stack. MySQL is integral to many high-profile, large-scale web products, including Wikipedia and Google.

An additional point is that MySQL offers excellent performance and scalability, making it suitable for both small applications and large, complex databases. Its ability to handle high volumes of data and transactions efficiently makes it a preferred choice for many businesses and developers.

Moreover, MySQL provides robust security features, ensuring that data is protected through various mechanisms such as user authentication, encryption, and access control. This makes it a reliable choice for handling sensitive information.

Finally, MySQL has a strong and active community, which means abundant resources, forums, and documentation are available for support. This community-driven approach ensures continuous improvement and quick resolution of issues, benefiting all users.

SYSTEM DESIGN

CHAPTER – 3

DESIGN

3.1 DATAFLOW DIAGRAM

The data flow diagram (DFD) is one of the most important tools used by system analysts. Data flow diagrams are made up of a number of symbols, which represent system components. Most data flow modeling methods use four kinds of symbols. These symbols are used to represent four kinds of system components. Processes, data stores, data flows, and external entities. Processes are represented by circles in DFD. Data Flow is represented by a thin line in the DFD and each data store has a unique name and a square or rectangle represents external entities.
Unlike detailed flowcharts, Data Flow Diagrams do not supply detailed descriptions of the modules but graphically describe a system's data and how the data interacts with the system.

To construct a Data Flow Diagram, we use,

- Arrow
- Circles
- Open End Box
- Squares

An arrow identifies the data flow in motion. It is a pipeline through which information is flown like the rectangle in the flowchart. A circle stands for a process that converts data into information. An open-ended box represents a data store, data at rest, or a temporary repository of data. A square defines a source or destination of system data.

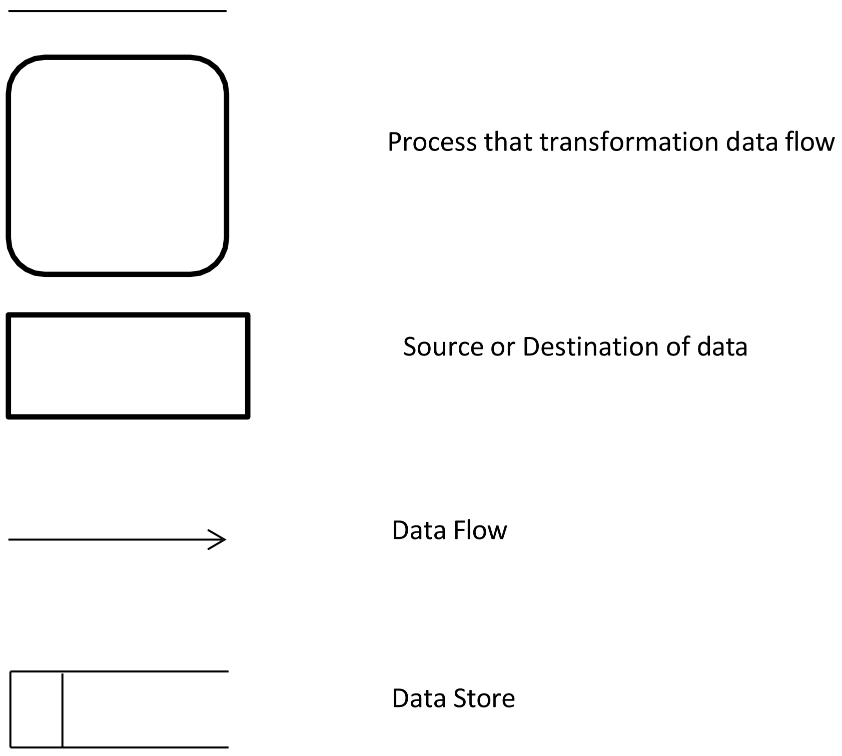
Five rules for constructing a Data Flow Diagram

- Arrows should not cross each other.
- Squares, circles, and files must bear names.
- Decomposed data flow squares and circles can have the same names.
- Choose meaningful names for data flow.
- Draw all data flows around the outside of the diagram.

DFD Symbols

In the DFD, there are four Symbols

- A square defines a source or destination system data
- An arrow identified data flow. It is the pipeline through which the information flow
- A circle or a bubble represents a process that transforms
- Incoming data flow into outgoing data flows
- An open rectangle is a data store, data at rest, or a temporary data


Process that transformation data flow
Source or Destination of data
Data Flow
Data Store

Level-0-DFD

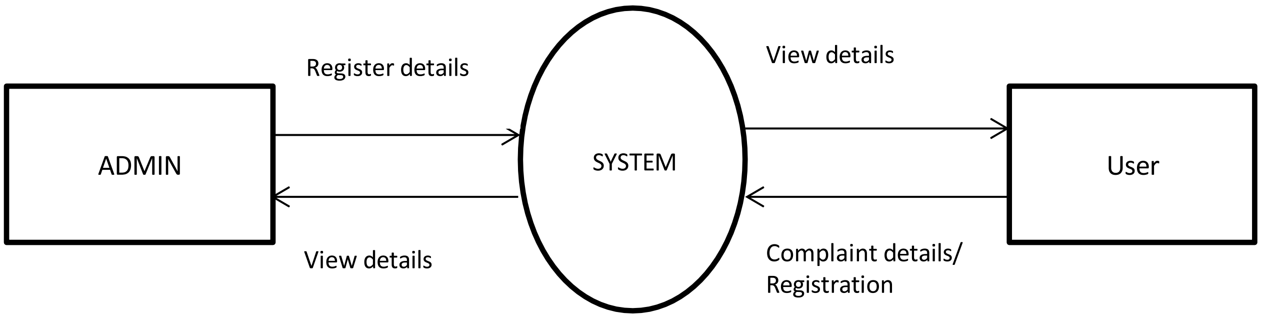


Fig: 3.1.1

Level-1-DFD (ADMIN)

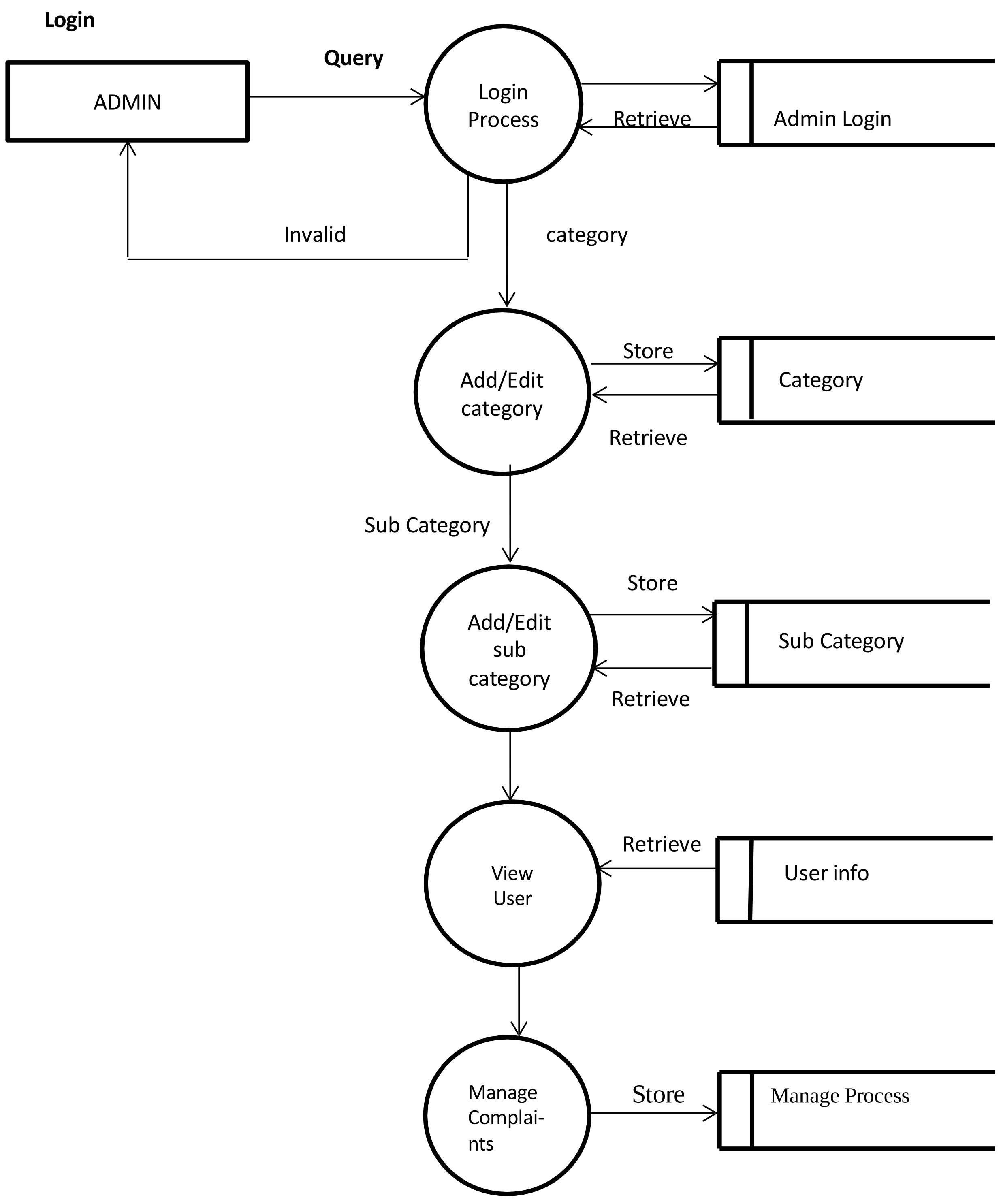


Fig: 3.1.2

Level-1-DFD (USER)

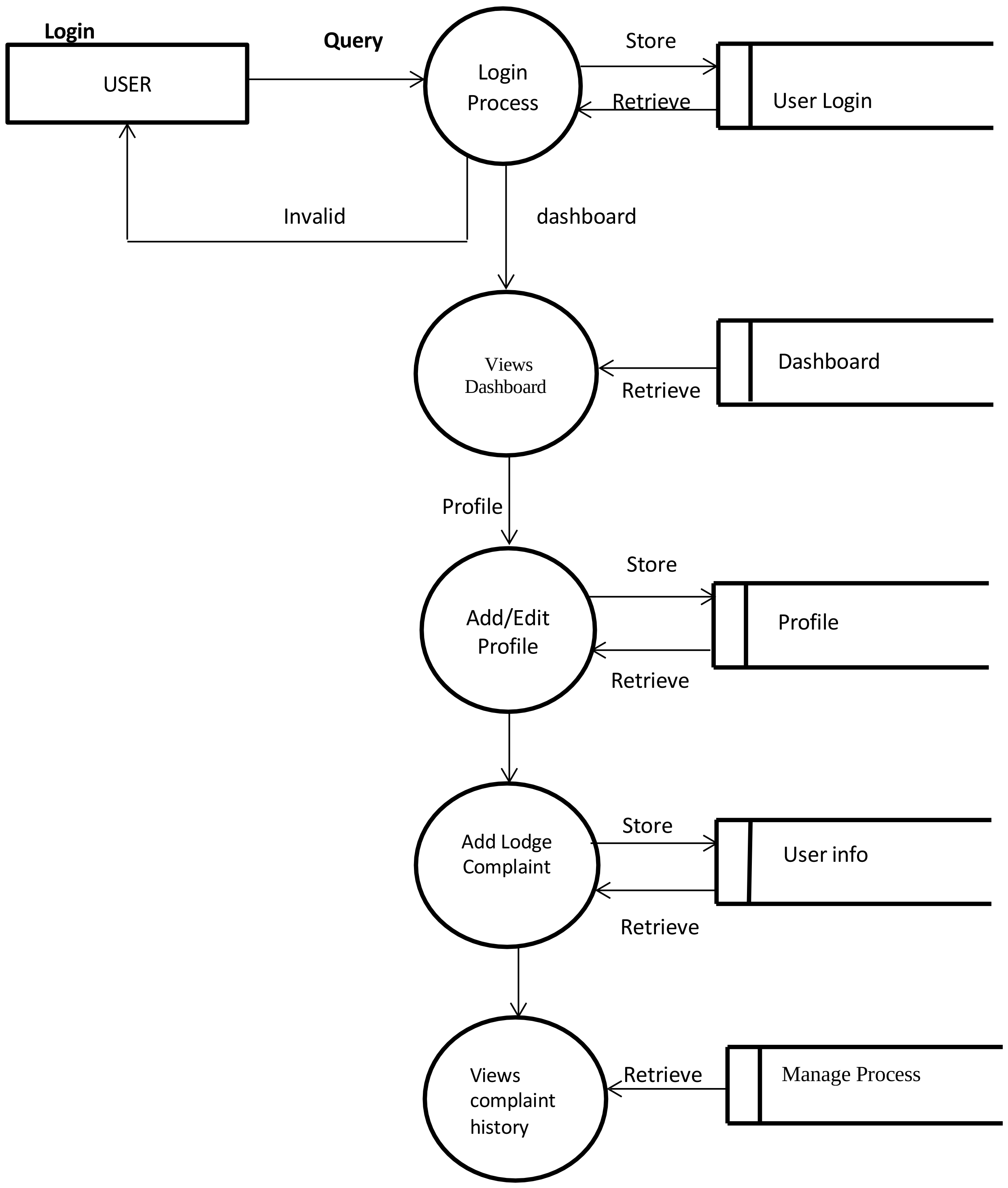


Fig: 3.1.3

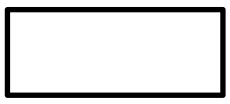
3.2 ENTITY RELATIONSHIP DIAGRAM

An Entity Relationship Diagram is a graphical tool to express the overall structure of a database. It is based on a perception of the real world which consists of a set of basic objects.

- An entity is a person, place, thing, or event of interest to the organization and about which data are captured, stored, or processed.
- The attributes are various kinds of data that describe an entity.
- An association of several entities in an Entity-Relationship model is called a relationship.

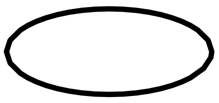
An ERD consists of the following major components:

Rectangles:



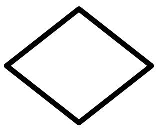
Used for representing entity types

Ellipses:



Used for representing attributes

Diamond:



Used for representing relationship types

Lines:

Used for linking attributes to entity type

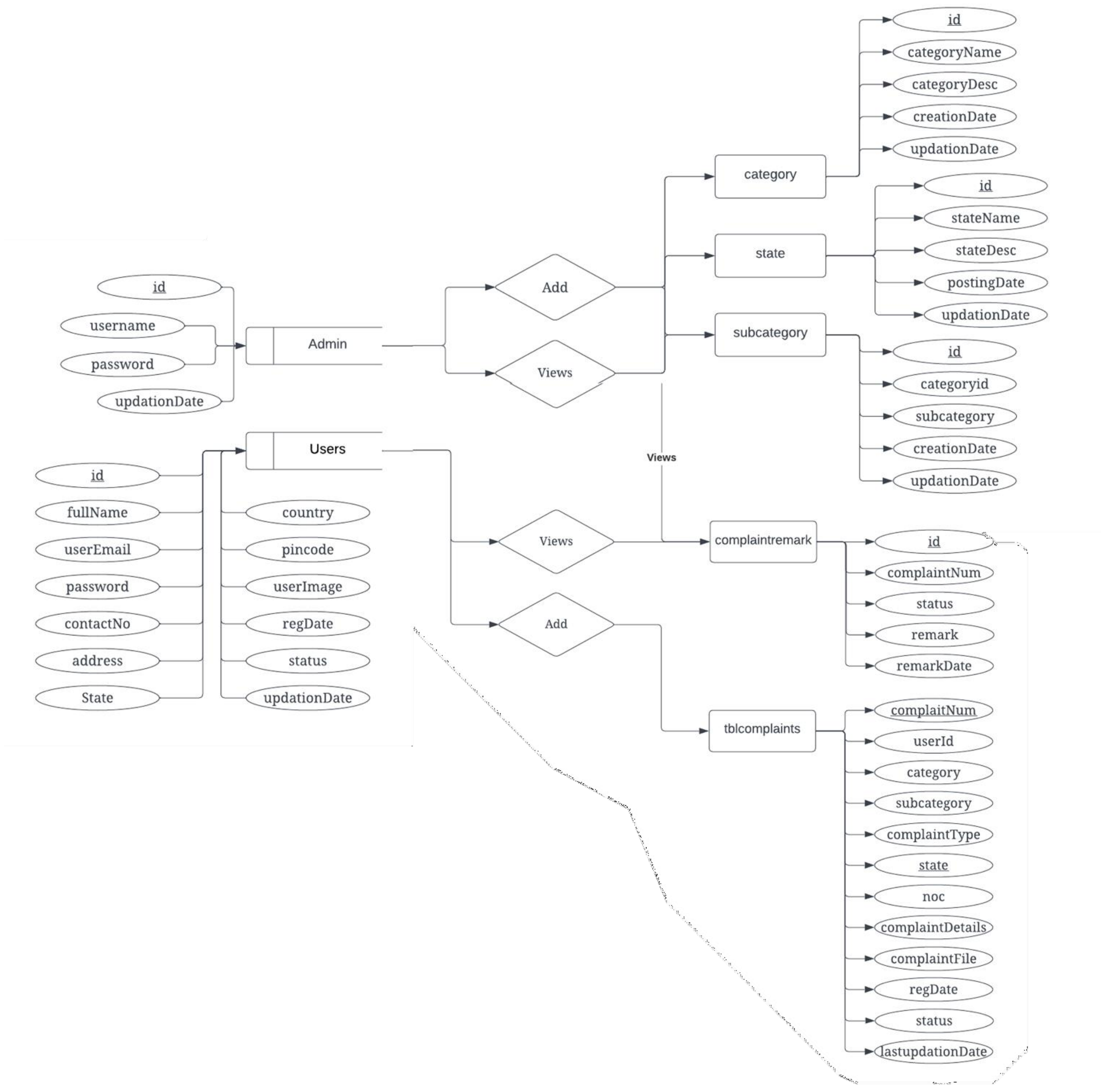


Fig: 3.2.1(ER DIAGRAM)

3.3 SYSTEM DESIGN

INPUT DESIGN

Input design is the process of converting the user-oriented input to a computer-based format. The goal of the input design is to make the data entry easier, logical, and free of error. Errors in the input data are controlled by the input design. The quality of the input determines the quality of the system output.

The entire data entry screen is interactive in nature so that the user can directly enter data according to the prompted messages. The user also can directly enter data according to the prompted messages. The users are also provided with the option of selecting an appropriate input from a list of values. This will reduce the number of errors, which are otherwise likely to arise if they were to be entered by the user itself.

Input design is one of the most important phases of the system design. Input design is the process where the input received in the system is planned and designed, so as to get necessary information from the user, eliminating the information that is not required. The aim of the input design is to ensure the maximum possible levels of accuracy and also ensure that the input is accessible and understood by the user.

The input design is the part of the overall system design, which requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify the errors.

Input design features can ensure the reliability of the system and produce results from accurate data or they can result in the production of erroneous information.

OUTPUT DESIGN

Output design is a very important concept in the computerized system. Without reliable output, the user may feel the entire system is unnecessary and avoid using it. Proper output design is crucial in any system and facilitates effective decision-making.

Computer output is the most important and direct source of information for the user. Efficient, intelligible output design should improve the system’s relationships with the user and help in decision-making. A major form of output is the hard copy from the printer.

Output requirements are designed during system analysis. A good starting point for the output design is the data flow diagram. Human factors reduce issues for design involved addressing internal controls to ensure readability.

An application is successful only when it can provide efficient and effective reports. Reports are actually a presentable form of the data. The report generation should be useful to the management for future reference. The report is the main source of information for users, operators, and management.

Reports generated are a permanent record of the transaction that occurred. After any valid transactions, the report of the same is generated and filed for future reference. Great care has been taken when designing the report as it plays an important role in decision-making.

Another key aspect of output design is ensuring the outputs are customizable and flexible to meet the varying needs of different users. Customizable reports allow users to filter, sort, and format data according to their specific requirements, enhancing usability and relevance.

DATABASE DESIGN

E-R modeling: It is an object-based model and is based on a perception of the real world that is made up of a collection of objects or activities and relationships among these. E-R modeling generally uses a top-down approach for new systems.

Normalization: It is a process of simplifying the relationship between data elements in a record. It is the transformation of complex data stores to a set of smaller, stable data structures.

Purpose of Normalization: Normalization is carried out due to the following reasons:

- Structuring the data so that there is no repetition of data, helps in saving space.
- To permit, simple retrieval of data in response to query and report requests.
- To simplify the maintenance of the data through updates, insertions, and deletions.
- To reduce the need to restructure or reorganize data when application requests arise.

Steps of Normalization:

System analysts should be familiar with the steps in Normalization. Since the process can improve the design of an application.

Starting with a data store developed for a data dictionary the analyst normalized a structure in three steps.

- 1NF: Any multi-valued attribute has been removed, so there is a single value at the insertion of each row & column of the table.
- 2NF: Any partial functional dependency has been removed.
- 3NF: Any transitive dependency has been removed.

Table structure for table `admin`:

```
CREATE TABLE `admin` (
  `Id` I nt (11) NOT NULL,
  `Username` varchar (250) NOT NULL,
  `Password` varchar (250) NOT NULL,
  `Updating Date` varchar (255) NOT NULL
);
```

Indexes for table `admin`:

```
ALTER TABLE `admin`
  ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT for table `admin`:

```
ALTER TABLE `admin`
  MODIFY `id` I nt (11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
```

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
id	Int(11)	NO	PRI	NULL	auto increment
username	Varchar(250)	NO		NULL	
password	Varchar(250)	NO		NULL	
UpdatingDate	Varchar(250)	NO		NULL	

Table structure for table `category`:

```
CREATE TABLE `category` (
  `Id` I nt (11) NOT NULL,
  `Category Name` varchar (255) NOT NULL,
  `Category Description` long text NOT NULL,
  `Creation Date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`UpdatingDate` varchar (255) NOT NULL

);
```

Indexes for table `category`:

```
ALTER TABLE `category`
ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT for table `category`:

```
ALTER TABLE `category`
  MODIFY `id` I nt (11) NOT NULL AUTO_INCREMENT;
```

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
id	I nt(11)	NO	PRI	NULL	auto _ increment
category Name	varchar(255)	NO		NULL	
category Description	long text	NO		NULL	
creation Date	timestamp	NO		CURRENT_TIMESTAMP	
UpdatingDate	varchar(255)	NO		NULL	

Table structure for table `complaint remark`:

```
CREATE TABLE `complaint remark` (
  `Id` I nt (11) NOT NULL,
`Complaint Number` I nt (11) NOT NULL,
  `Status` varchar (255) NOT NULL,
 `Remark` medium text NOT NULL,
 `Remark Date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

Indexes for table `complaint remark`:

```
ALTER TABLE `complaint remark`
ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT for table `complaint remark`:

```
ALTER TABLE `complaint remark`
 MODIFY `id` I nt (11) NOT NULL AUTO_INCREMENT;
```

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
Id	Int(11)	No	PRI	NULL	Auto increment
complaint Number	Int(11)	No		NULL	
Status	Varchar(255)	No		NULL	
Remark	Medium text	No		NULL	
remark date	Timestamp	No		CURRENT_TIMESTAMP	

Table structure for table `state`

```
CREATE TABLE `state` (
  `Id` int (11) NOT NULL,
  `State Name` varchar (255) NOT NULL,
  `State Description` tiny text NOT NULL,
  `Posting Date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `UpdatingDate` varchar (255) NOT NULL
);
```

Indexes for table `state`:

```
ALTER TABLE `state`
  ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT for table `state`:

```
ALTER TABLE `state`
  MODIFY `id` int (11) NOT NULL AUTO_INCREMENT;
```

Field	Type	Null	Key	Default	Extra
id	Int(11)	NO	PRI	NULL	Auto _ increment
complaint Number	Int(11)	NO		NULL	
status	varchar(255)	NO		NULL	
remark	medium text	NO		NULL	
remark Date	timestamp	NO		CURRENT_TIMESTAMP	

Table structure for table `subcategory`

```
CREATE TABLE `subcategory` (
  `Id` int (11) NOT NULL,
  `Category ID` int (11) NOT NULL,
  `Subcategory` varchar (255) NOT NULL,
  `Creation Date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `UpdatingDate` varchar (255) NOT NULL
);
```

Indexes for table `subcategory`:

```
ALTER TABLE `subcategory`
ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT for table `subcategory`:

```
ALTER TABLE `subcategory`
  MODIFY `id` int (11) NOT NULL AUTO_INCREMENT;
```

Field	Type	Null	Key	Default	Extra
id	Int(11)	NO	PRI	NULL	Auto _ increment
category ID	Int(11)	NO		NULL	
subcategory	Varchar(255)	NO		NULL	
Creation Date	timestamp	NO		CURRENT_TIMESTAMP	
UpdatingDate	varchar(255)	NO		NULL	

Table structure for table `tablecomplaints`:

```
CREATE TABLE `tablecomplaints` (
        `Complaint Number` int (11) NOT NULL,
        `UserId` int (11) NOT NULL,
        `Category` int (11) NOT NULL,
        `Subcategory` varchar (255) NOT NULL,
        `Complaint Type` varchar (255) NOT NULL,
        `State` varchar (255) NOT NULL,
        `Noc` varchar (255) NOT NULL,
        `Complaint Details` medium text NOT NULL,
        `Complaint File` varchar (255) DEFAULT NULL,
        `RegDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
        `Status` varchar (50) DEFAULT NULL,
        `LastUpdatingDate` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON
    UPDATE CURRENT_TIMESTAMP
       );
```

Indexes for table 'table complaints':

```
       ALTER TABLE `tablecomplaints`
        ADD PRIMARY KEY (`complaint Number`);
```

AUTO_INCREMENT for table `tablecomplaints`:

```
       ALTER TABLE `tablecomplaints`
        MODIFY `complaint Number` int (11) NOT NULL AUTO_INCREMENT;
```

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
complaint Number	Int(11)	NO	PRI	NULL	Auto increment
UserId	Int(11)	NO		NULL	
category	Int(11)	NO		NULL	

subcategory	varchar(255)	NO		NULL	
complaint Type	varchar(255)	NO		NULL	
state	varchar(255)	NO		NULL	
Noc	varchar(255)	NO		NULL	
complaint Details	medium text	NO		NULL	
complaint File	varchar(255)	YES		NULL	
RegDate	timestamp	NO		CURRENT_TIMESTAMP	
status	varchar(50)	YES			
LastUpdatingDate	timestamp	NO		0000-00-00 00:00:00	

Table structure for table `user log`:

```
CREATE TABLE `user log` (
  `Id` int (11) NOT NULL,
  `Uid` int (11) NOT NULL,
  `Username` varchar (255) NOT NULL,
  `User I p` binary (16) NOT NULL,
  `Login Time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `Logout` varchar (255) NOT NULL,
  `Status` int (11) NOT NULL);
```

Indexes for table `user log`:

```
ALTER TABLE `user log`
  ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT for table `user log`:

```
ALTER TABLE `user log`
MODIFY `id` int (11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
```

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
id	Int(11)	NO	PRI	NULL	auto increment
Uid	Int(11)	NO		NULL	
username	Varchar(255)	NO		NULL	

User I p	binary(16)	NO		NULL	
login Time	timestamp	NO		CURRENT_TIMESTAMP	
logout	varchar(255)	YES		NULL	
status	Int(11)	NO		NULL	

Table structure for table `users`:

```
CREATE TABLE `users` (
  `Id` I nt (11) NOT NULL,
  `Full Name` varchar (255) DEFAULT NULL,
  `User Email` varchar (255) DEFAULT NULL,
  `Password` varchar (255) DEFAULT NULL,
  `Contact No` big I nt (11) DEFAULT NULL,
  `Address` tiny text,
  `State` varchar (255) DEFAULT NULL,
  `Country` varchar (255) DEFAULT NULL,
  `Pin code` I nt (6) DEFAULT NULL,
  `User Image` varchar (255) DEFAULT NULL,
  `Reg Date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `UpdatingDate` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON UPDATE
CURRENT_TIMESTAMP,
  `Status` I nt (1) NOT NULL
);
```

Indexes for table `users`:

```
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);
```

AUTO_INCREMENT for table `users`:

```
ALTER TABLE `users`
  MODIFY `id` I nt (11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
COMMIT;
```

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
id	Int(11)	NO	PRI	NULL	auto increment
full Name	varchar(255)	YES		NULL	
user Email	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
contact No	Big I nt(11)	YES		NULL	
address	tiny text	YES		NULL	
State	varchar(255)	YES		NULL	
country	varchar(255)	YES		NULL	
pin code	I nt(6)	YES		NULL	
user Image	varchar(255)	YES		NULL	
RegDate	timestamp	NO		CURRENT_TIMESTAMP	
UpdatingDate	timestamp	NO		0000-00-00 00:00:00	
status	I nt(1)	NO		NULL	

MODULE DESCRIPTION

USER

- Login
- Register
- Post Complaint
- View complaint status
- Feedback
- Get Admin Contact details

ADMIN

- Login
- View Complaint
- Update status
- Update proof of work
- The administrator has the full-fledged rights over the GMS.
- Can view the accounts.
- Can change the password.
- Can hide any kind of features from both users.
- Insert the information available on GMS.

Login Module

The main activity in the application is the user login page for the user. The other modules are followed by this login page. This module records only the user and password of the user.

Registration Module

Another main function of our proposed system is the registration module. To register, users need to provide unique application details such as name, password, email, place, and time. This information ensures that each user has a unique profile within the system, allowing for personalized interaction and secure access.

USER

- **Post Complaint**

  Users can post complaints through the application, describing issues such as irregular water supply in their area.
- **Location Mark in Google Maps**

  Users mark their location on Google Maps for precise complaint registration. This feature aids the administration in identifying and locating the problem area accurately.
- **View Status**

  Users can track the status of their complaints from registration to resolution. This transparency keeps users informed and reassured about the progress of their issues.
- **Feedback**

  Users provide feedback on the actions taken regarding their complaints. This input helps the administration improve service quality and responsiveness.
- **Get Admin Contact Details**

  Users can access admin contact details, allowing for direct communication if needed.

ADMIN

- **View Complaint**

  Admins view detailed complaints, including descriptions and images.
- **Update Status**

  Admins update the status of complaints in real-time, keeping users informed about the progress and current state of their issues.
- **Update Completed Proof**

  Admins upload photo proof of completed work, ensuring accountability and transparency. Users can visually confirm that their complaints have been addressed.
- **Prioritize Complaints**

  Admins can prioritize complaints based on their severity and urgency, ensuring critical issues are resolved promptly and improving overall service efficiency.
- **Generate Reports**

  Admins generate detailed reports on complaint statistics, response times, and resolution effectiveness. These reports help assess performance, identify recurring issues, and support data-driven decision-making for service improvements.
- **Communicate with Users**

  Admins send messages and notifications to users regarding their complaints

TESTING

CHAPTER – 4

TESTING

Testing is a process used to help identify the correctness, completeness, and quality of developed computer software. With that in mind, testing can never completely establish the correctness of computer software. The quality of the application can and normally does vary widely from system to system but some of the common quality attributes include reliability, stability, portability, maintainability, and usability. Refer to the ISO standard ISO 9126 for a more complete list of attributes and criteria. Testing helps in verifying and Validating if the Software is working as it is intended to be working. This involves using Static and Dynamic methodologies to Test the application.

TESTING OBJECTIVES INCLUDE

1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a high probability of finding an as-yet-undiscovered error.
3. A successful test is one that uncovers an as-yet-undiscovered error.

Testing should systematically uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. A secondary benefit of testing is that it demonstrates that the software appears to be working as stated in the specifications.

TESTING START PROCESS

WHEN TESTING SHOULD START:

Testing early in the life cycle reduces the errors. Test deliverables are associated with every phase of development. The goal of a Software Tester is to find bugs, find them as early as possible, and make sure they are fixed. The number one cause of Software bugs is the Specification. There are several not communicated well to the entire team. Planning software is vitally important. If it's not done reasons specifications are the largest bug producer. In many instances, a Spec simply isn't written. Other reasons may be that the spec isn't thorough enough, it's constantly changing, or if it's correct bugs will be created.

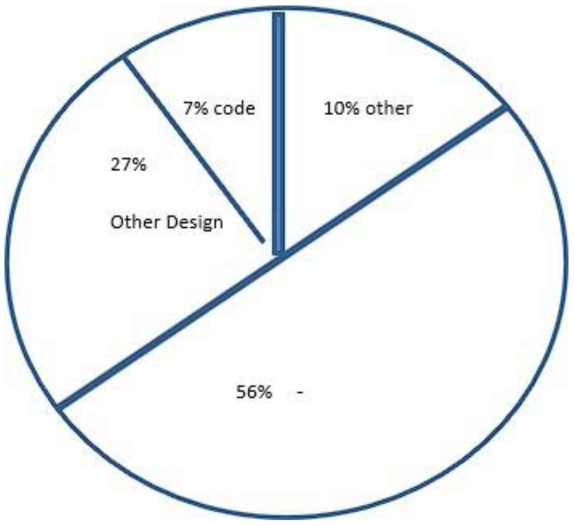


Fig: 5.1.1

Coding errors may be more familiar to you if you are a programmer. Typically these can be traced to the Software complexity, poor documentation, schedule pressure, or just plain mistakes. It's important to note that many bugs that appear on the surface to be programming errors can really be traced to specification. It's quite common to hear a programmer say, "Oh, so that's what it's supposed to do.

If someone had told me that I wouldn't have written the code that way. The other category is the catch-all for what is left. Some bugs can blamed for false positives, conditions that were thought to be bugs but really weren't. There may be duplicate bugs, multiple ones that resulted from the square root cause. Some bugs can be traced to Testing errors.

Costs: The costs are logarithmic- that is, they increased tenfold as time increased. A bug found and fixed during the early stages when the specification is being written might cost next to nothing, or 10 cents in our example. The same bug, if not found until the software is coded and tested, might cost $1 to $10. If a customer finds it, the cost would easily top $100.

TESTING STOP PROCESS

WHEN TO STOP TESTING

This can be difficult to determine. Many modern software applications are so complex and run in such an interdependent environment, that complete testing can never be done. "When to stop testing" is one of the most difficult questions for a test engineer. Common factors in deciding when to stop are:

- Deadlines (release deadlines, testing deadlines.)
- Test cases completed with certain percentages passed
- Test budget depleted
- Coverage of code/functionality/requirements reaches a specified point
- The rate at which Bugs can be found is too small
- Beta or Alpha Testing period ends
- The risk in the project is under an acceptable limit.

Practically, we feel that the decision to stop testing is based on the level of risk acceptable to the management. As testing is a never-ending process we can never assume that 100 % testing has been done, we can only minimize the risk of shipping the product to the client with X testing done. The risk can be measured by Risk analysis but for small duration / low budget / low resources projects, risk can be deduced by simply: -

- Measuring Test Coverage.
- Number of test cycles.
- Number of high-priority bugs.

TEST PLAN

TEST PLAN CONSISTS OF THE FOLLOWING PARTS:

TITLE OF THE PROJECT:

Title "Grievances Registration and Management System".

Objective of the document: - In this test plan we are covering the activities and functionality of different modules and their sub-modules. In this document, we cover what kind of test cases should described.

Scope of the document: In this document, in each phase, what are going to do and how are going to do it? In this section, we are mentioning which requirements are tested.

For the Login Module, the requirements are:

- User name and password should not be empty
- If it is empty then the alert message will be prompted.

The above requirements are only for the login module and other functional requirements can declared in this section. The requirements which are declared in this section it may depend on other requirements in different modules.

The Objective of Testing: the main objective of testing in this application is to chances of preventing defects on the client environment.

Critical Functionalities: - in this section, we are discussing Key roles & Causes for the success of the application.

Test Data Requirements and Collection:-

In this section, we are collecting the requirements for the application. From the different resources. Such as

- Collecting from the client.
- Refereeing the existing applications which are similar to the current ones

Training Requirements:-

Training requirements are focused on 2 areas

a) Technology – in this, we are discussing the new technology in the market and the scope of that technology in the future.

b) Domain- in this we are discussing the Existing personal Training knowledge.

Resource Requirements: - In this, we discuss the required resources for the application such

- Employees
- Software Licenses
- Bridge Number

Scheduling: - In this section, we will be discussing about starting date and End Date of the application.

Input Criteria: - there are different criteria conditions i.e. Unit Testing, R e lease Note & Installation.

Exit Criteria: - In this only evaluation documents only accepted.

Risk Analysis: - In this, we are conversing about Risk Analysis such as

a) Risk on Resources.

b) Risk on TimeLine. Risks are identified by Preparing a Solution Plan.

TYPES OF TEST CARRIED:-

In our application, we are performing the Gary Box testing (white and black).

4.1 WHITE BOX TESTING

White Box Testing mainly focuses on the internal performance of the Complaints. Here apart will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors.

Also, construct a loop in such a way that the part will be tested within a range. That means the part is executed at its boundary values and within bounds for the purpose of testing.

White Box Testing in this Project: I tested stepwise every piece of code, taking care that every statement in the code is executed at least once.

I have generated a list of test cases, and sample data, which is used to check all possible combinations of execution paths through the code at every module level.

4.2 BLACK BOX TESTING

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather than getting into details at the statement level. Here the module will be treated as a block box that will take some input and generate output. Output for a given set of input combinations is forwarded to other modules.

Black Box testing in this Project: I tested each and every module by considering each module as a unit. I have prepared some sets of input combinations and checked the outputs for those inputs. Also, I tested whether the communication between one module and to other module is performing well or not.

4.3 VALIDATION TESTING:

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled.

Validation helps in building the right complaints as per the user's requirement and helps in satisfying their needs. Validation is basically done by the testers during the testing. In case of erroneous input corresponding error messages are displayed.

TEST CASE

Test No.	Form Name	Inputs to different fields	Expected Result	Actual Result	Status	Remarks
1	Login Form	Username, password empty	Alert messages should be displayed	Alert message displayed	Test Success	Pass
2	Login Form	Username, password provided is incorrect	Should stay on the same page	It is on the same page	Test Success	Pass
3	Login Form	Username, password provided is correct	Should navigate to the main page	Redirected to the main page	Test Success	Pass
4	User Registration Form	Enter All details like name, password ,mobile, email id etc.,	Should Insert the value to dB and redirect to index page	Inserted, redirected to index page	Test Success	Pass
5	Admin Login Form	Username, password provided is correct	Should navigate to the dashboard	Redirected to dashboard	Test Success	Pass
6	Category Form	Category name , category Description should be provided	Description should be stored in file system and other details into dB	Description Stored in file system and others in dB	Test Success	Pass
7	Sub Category Form	Category name , Sub category name should be provided	Category name , Sub category name should be stored in dB	Details Stored in dB	Test Success	Pass
8.	Complaint Form	Category , sub category and Complaint and Complaint image should be provided	The given data's should be stored in the database	Details Stored in the database	Test Success	Pass

CONCLUSION AND FUTURE SCOPE

CHAPTER – 5

CONCLUSION & FUTURE SCOPE

CONCLUSION

The **Grievances Registration and Management System has** been successfully developed with the needs and requirements of the company. The system is tested with intensive care for all possibilities of errors. The database in this system is maintained in a controlled manner to avoid all possible errors.

The system developed is very user friendly and there is no need for any special training with the software. All the data involved in this system can be viewed in the form of a report at any time. The reports are produced in such a way that satisfies the management to make certain decisions over the company management.

FUTURE SCOPE

The "Grievances Registration and Management System" can be further developed into a separate, automated system with the following enhancements:

More future work and knowledge are needed to further improve the performance of the opinion spam analysis. There is a huge need in the industry, in day-to-day life for such applications because every company wants to know how consumers really feel about their products and services and those of their competitors by analyzing true reviews, not spam reviews. This research proposes an opinion spam analyzer that automatically classifies input text data into either spam or non-spam categories.

A direction for future research is to implement the system and check performance by applying the proposed approach to various benchmark data sets. Comparing the performance of different classification methods to find the best one for our proposed opinion spam classification method could be another future research direction. However, there exist other kinds of review or reviewer-related features that are likely to make a contribution to the prediction task. In the future, we will further investigate different kinds of features to make more accurate predictions

BIBLIOGRAPHY

CHAPTER - 6

BIBLIOGRAPHY

6.1 BOOKS

1. "Python Programming: A Complete Guide for Beginners to Master, Python Programming Language" by Brian Draper
2. "Beginning Programming with Python for Dummies" by John Paul Mueller
3. "Python: Programming For Beginners: Learn The Fundamentals of Python in 7 Days" by Michael Knapp and Python Programming
4. "Python Programming for Beginners: Python Programming Language Tutorial" by Joseph Joyner
5. "Python Programming: Using Problem Solving Approach" by Reema Thareja

6.2 WEBSITES

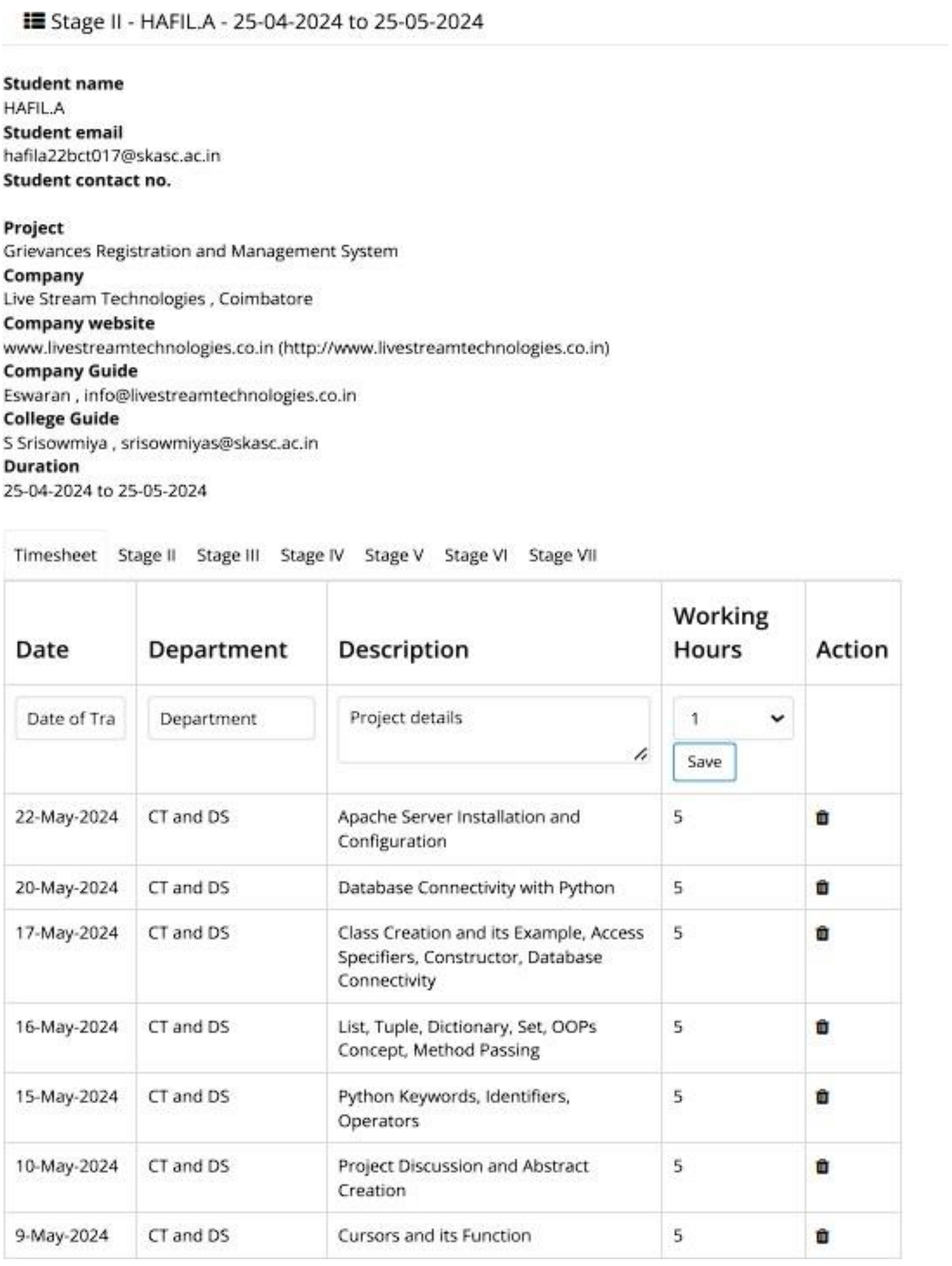
1. For JQuery, http://jquery.com/
2. For CSS, http://www.w3schools.com/CSS/default.asp
3. For JavaScript, http://www.w3schools.com/js/default.asp
4. For MySQL, http://www.w3schools.com/php/php_mysql_intro.asp
5. https://howtothink.readthedocs.io/en/latest/
6. https://python.swaroopch.com/
7. https://learnpythonthehardway.org/book/
8. https://linux.die.net/diveintopython/html/toc/index.html
9. https://anandology.com/python-practice-book/
10. https://developers.google.com/edu/python/

APPENDIX

CHAPTER – 7

APPENDIX

7.1 TIME SHEET



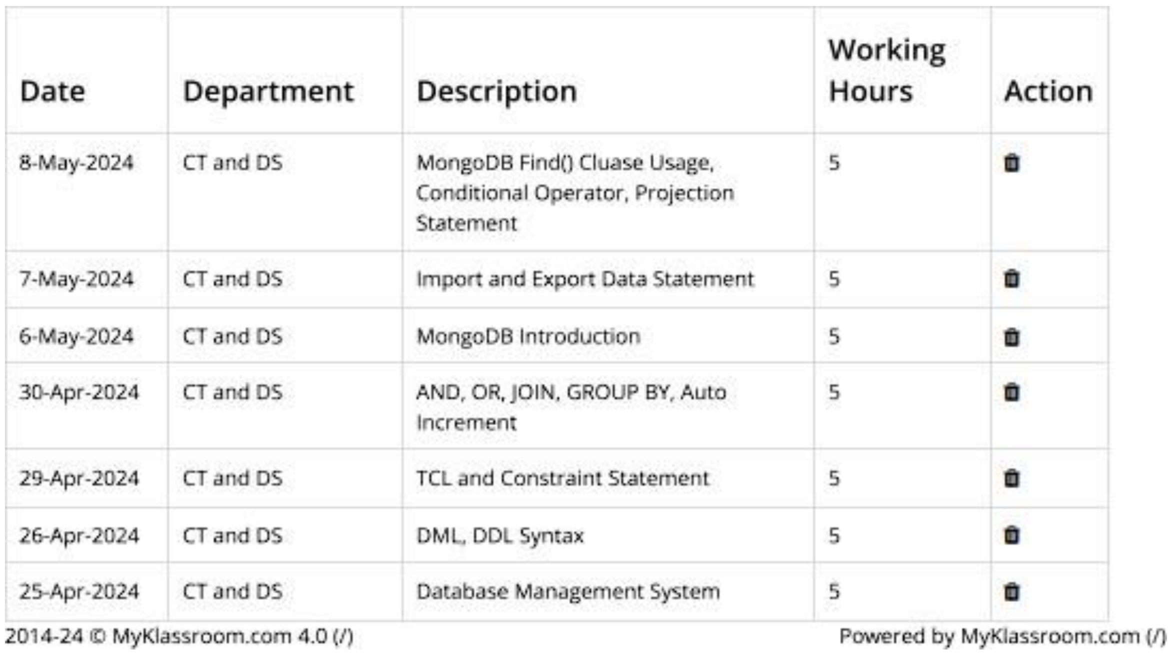
Stage II - HAFIL.A - 25-04-2024 to 25-05-2024

Student name
HAFIL.A
Student email
hafila22bct017@skasc.ac.in
Student contact no.

Project
Grievances Registration and Management System
Company
Live Stream Technologies , Coimbatore
Company website
www.livestreamtechnologies.co.in (http://www.livestreamtechnologies.co.in)
Company Guide
Eswaran , info@livestreamtechnologies.co.in
College Guide
S Srisowmiya , srisowmiyas@skasc.ac.in
Duration
25-04-2024 to 25-05-2024

Date	Department	Description	Working Hours	Action
Date of Tra	Department	Project details	1 Save	
22-May-2024	CT and DS	Apache Server Installation and Configuration	5	🗑
20-May-2024	CT and DS	Database Connectivity with Python	5	🗑
17-May-2024	CT and DS	Class Creation and its Example, Access Specifiers, Constructor, Database Connectivity	5	🗑
16-May-2024	CT and DS	List, Tuple, Dictionary, Set, OOPs Concept, Method Passing	5	🗑
15-May-2024	CT and DS	Python Keywords, Identifiers, Operators	5	🗑
10-May-2024	CT and DS	Project Discussion and Abstract Creation	5	🗑
9-May-2024	CT and DS	Cursors and its Function	5	🗑

Fig: 7.1.1



Date	Department	Description	Working Hours	Action
8-May-2024	CT and DS	MongoDB Find() Cluase Usage, Conditional Operator, Projection Statement	5	🗑
7-May-2024	CT and DS	Import and Export Data Statement	5	🗑
6-May-2024	CT and DS	MongoDB Introduction	5	🗑
30-Apr-2024	CT and DS	AND, OR, JOIN, GROUP BY, Auto Increment	5	🗑
29-Apr-2024	CT and DS	TCL and Constraint Statement	5	🗑
26-Apr-2024	CT and DS	DML, DDL Syntax	5	🗑
25-Apr-2024	CT and DS	Database Management System	5	🗑

2014-24 © MyKlassroom.com 4.0 (/)   Powered by MyKlassroom.com (/)

Fig: 7.1.2

7.2 SAMPLE CODE

index.html

```
<!DOCTYPE html>
<html lang=”en”>
<head>
   <meta charset=”utf-8”>
   <meta http-equiv=”X-UA-Compatible” content=”IE=edge”>
   <meta name=”viewport” content=”width=device-width, initial-scale=1”>
   <meta name=”description” content=””>
   <meta name=”author” content=””>

   <title>Complaint Management System</title>
   <link href=”css/bootstrap.min.css” rel=”stylesheet”>
   <link href=”css/half-slider.css” rel=”stylesheet”>
</head>
<body>
   <!—Navigation →
   <nav class=”navbar navbar-inverse navbar-fixed-top” role=”navigation”>
      <div class=”container”>
         <!—Brand and toggle get grouped for better mobile display →
         <div class=”navbar-header”>
            <button type=”button” class=”navbar-toggle” data-toggle=”collapse” data-target=”#bs-example-
navbar-collapse-1”>
               <span class=”sr-only”>Toggle navigation</span>
               <span class=”icon-bar”></span>
               <span class=”icon-bar”></span>
               <span class=”icon-bar”></span>
            </button>
            <a class=”navbar-brand” href=”#”>COMPLAINT MANAGEMENT </a>
         </div>
         <!—Collect the nav links, forms, and other content for toggling →
         <div class=”collapse navbar-collapse” id=”bs-example-navbar-collapse-1”>
            <ul class=”nav navbar-nav”>
               <li>
                  <a href=”http://localhost/Complaint Management System/users/”>User Login</a>
               </li>
               <li>
                  <a href=”http://localhost/Complaint Management System/users/registration.php”>User
Registration</a>
               </li>
               <li>
                  <a href=”http://localhost/Complaint Management System/admin/”>admin</a>
               </li>
            </ul>
         </div>
         <!-- /.navbar-collapse →
      </div>
      <!-- /.container →
   </nav>
   <!—Half Page Image Background Carousel Header →
   <header id=”myCarousel” class=”carousel slide”>
```

```
        <!—Indicators →
        <ol class=”carousel-indicators”>


            <li data-target=”#myCarousel” data-slide-to=”0” class=”active”></li>
            <li data-target=”#myCarousel” data-slide-to=”1”></li>
            <li data-target=”#myCarousel” data-slide-to=”2”></li>
        </ol>

        <!—Wrapper for Slides →
        <div class=”carousel-inner”>

            <div class=”item active”>
                <!—Set the second background image using inline CSS below. →
                <div class=”fill” style=”background-
image:url(‘http://localhost/Complaint%20Management%20System/img/c2.jpg’);”></div>
                <div class=”carousel-caption”>

                </div>
            </div>
            <div class=”item”>
                <!—Set the third background image using inline CSS below. →
                <div class=”fill” style=”background-
image:url(‘http://localhost/Complaint%20Management%20System/img/c10.jpg’);”></div>
                <div class=”carousel-caption”>

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

        <!—Controls →
        <a class=”left carousel-control” href=”#myCarousel” data-slide=”prev”>
            <span class=”icon-prev”></span>
        </a>
        <a class=”right carousel-control” href=”#myCarousel” data-slide=”next”>
            <span class=”icon-next”></span>
        </a>
    </header>

    <!—Page Content →
    <div class=”container”>

        <div class=”row”>
            <div class=”col-lg-12”>
<div class=”com”><h1><b>WELCOME TO COMPLAINT MANAGEMENT SYSTEM</b></h1>
                <p><h2>The Easiest Way To Complain Online </h2><br> <b><i>Tired off getting ripped off?
Fight back and file a customer complaint now.Sign Up Today And You Can Leave Complaints on various
topics/company.</b> </i></p></div>
            </div>
        </div>
<style>.com{text-align: center;} </style>
        <hr>

        <!—Footer →
        <footer>
```

```
          <div class="row">
              <div class="col-lg-12">
                  <p>Copyright &copy; 2024 CMS</p>
              </div>
          </div>



          <!-- /.row →
      </footer>
      <style> .col-lg-12{text-align: center; border: 1px solid black;}</style>

  </div>
  <!-- /.container →

  <!—jQuery →
  <script src="js/jquery.js"></script>

  <!—Bootstrap Core JavaScript →
  <script src="js/bootstrap.min.js"></script>

  <!—Script to Activate the Carousel →
  <script>
  $('.carousel').carousel({
      interval: 5000 //changes the speed
  })
  </script>
</body>
</html>
```

index.py

```
#!C:\Users\hafil\AppData\Local\Programs\Python\Python312/python.exe
print("Content-Type:text/html\n\r")
print("""
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">


    <title>Grievances Registeration and Management System</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/half-slider.css" rel="stylesheet">
</head>

<body>
    <!—Navigation →
    <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
        <div class="container">
```

```
          <!—Brand and toggle get grouped for better mobile display →
          <div  class=”navbar-header”>
             <button type=”button” class=”navbar-toggle” data-toggle=”collapse” data-target=”#bs-example-
navbar-collapse-1”>
                <span class=”sr-only”>Toggle navigation</span>
                <span class=”icon-bar”></span>
                <span class=”icon-bar”></span>
                <span class=”icon-bar”></span>
             </button>



             <a class=”navbar-brand” href=”#”>GRIEVANCES MANAGEMENT </a>
          </div>
          <!—Collect the nav links, forms, and other content for toggling →
          <div class=”collapse navbar-collapse” id=”bs-example-navbar-collapse-1”>
             <ul class=”nav navbar-nav”>
                <li>
                   <a href=”http://localhost/Complaint Management System/users/”>User Login</a>
                </li>
                <li>
                   <a href=”http://localhost/Complaint Management System/users/registration.py”>User
Registration</a>
                </li>
                <li>
                   <a href=”http://localhost/Complaint Management System/admin/”>admin</a>
                </li>
             </ul>
          </div>
          <!-- /.navbar-collapse →
       </div>
       <!-- /.container →
    </nav>

    <!—Half Page Image Background Carousel Header →
    <header id=”myCarousel” class=”carousel slide”>
       <!—Indicators →
       <ol  class=”carousel-indicators”>
          <li data-target=”#myCarousel” data-slide-to=”0” class=”active”></li>
          <li data-target=”#myCarousel” data-slide-to=”1”></li>
          <li data-target=”#myCarousel” data-slide-to=”2”></li>
       </ol>

       <!—Wrapper for Slides →
       <div  class=”carousel-inner”>
          <div class=”item active”>
             <!—Set the second background image using inline CSS below. →
             <div class=”fill” style=”background-
image:url(‘http://localhost/Complaint%20Management%20System/img/c2.jpg’);”></div>
             <div  class=”carousel-caption”>

             </div>
          </div>
          <div class=”item”>
             <!—Set the third background image using inline CSS below. →
```

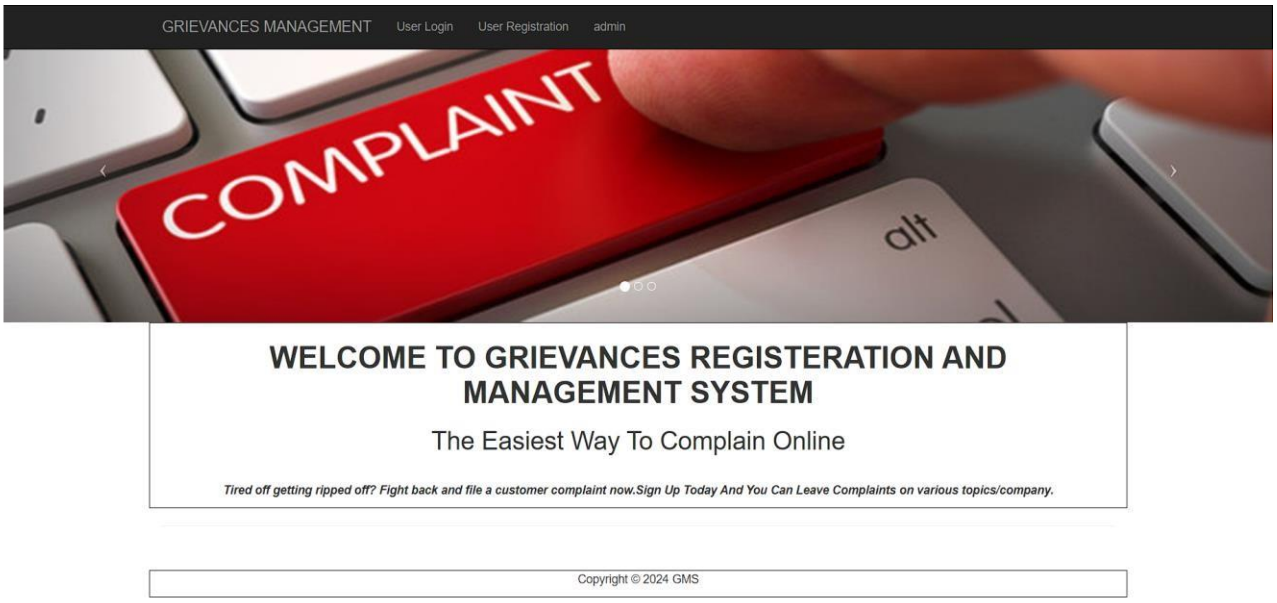
```
                <div class="fill" style="background-
image:url('http://localhost/Complaint%20Management%20System/img/c10.jpg');"></div>
                <div  class="carousel-caption">
                </div>
            </div>
        </div>
        <!—Controls →
        <a class="left carousel-control" href="#myCarousel" data-slide="prev">
            <span class="icon-prev"></span>
        </a>
        <a class="right carousel-control" href="#myCarousel" data-slide="next">
            <span class="icon-next"></span>
        </a>
    </header>
    <!—Page Content →
    <div class="container">
        <div class="row">
            <div  class="col-lg-12">
                <div class="com"><h1><b>WELCOME TO GRIEVANCES REGISTERATION AND
MANAGEMENT SYSTEM</b></h1>
                <p><h2>The Easiest Way To Complain Online </h2><br> <b><i>Tired off getting ripped off?
Fight back and file a customer complaint now.Sign Up Today And You Can Leave Complaints on various
topics/company.</b> </i></p></div>
            </div>
        </div>
<style>.com{text-align: center;} </style>
        <hr>
        <!—Footer →
        <footer>
            <div class="row">
                <div  class="col-lg-12">
                    <p>Copyright &copy; 2024 GMS</p>
                </div>
            </div>
            <!-- /.row →
        </footer>
        <style> .col-lg-12{text-align: center; border: 1px solid black;}</style>
    </div>
    <!-- /.container →
    <!—jQuery →
    <script src="js/jquery.js"></script>
    <!—Bootstrap Core JavaScript →
    <script src="js/bootstrap.min.js"></script>
    <!—Script to Activate the Carousel →
    <script>
    $('.carousel').carousel({
        interval: 5000 //changes the speed
    })
    </script>

</body>

</html>
""")
```

7.3 SCREENSHOTS

HOME PAGE:



WELCOME TO GRIEVANCES REGISTERATION AND MANAGEMENT SYSTEM

The Easiest Way To Complain Online

Tired off getting ripped off? Fight back and file a customer complaint now.Sign Up Today And You Can Leave Complaints on various topics/company.

Copyright © 2024 GMS

Fig: 7.3.1(Home page)

USER LOGIN PAGE:

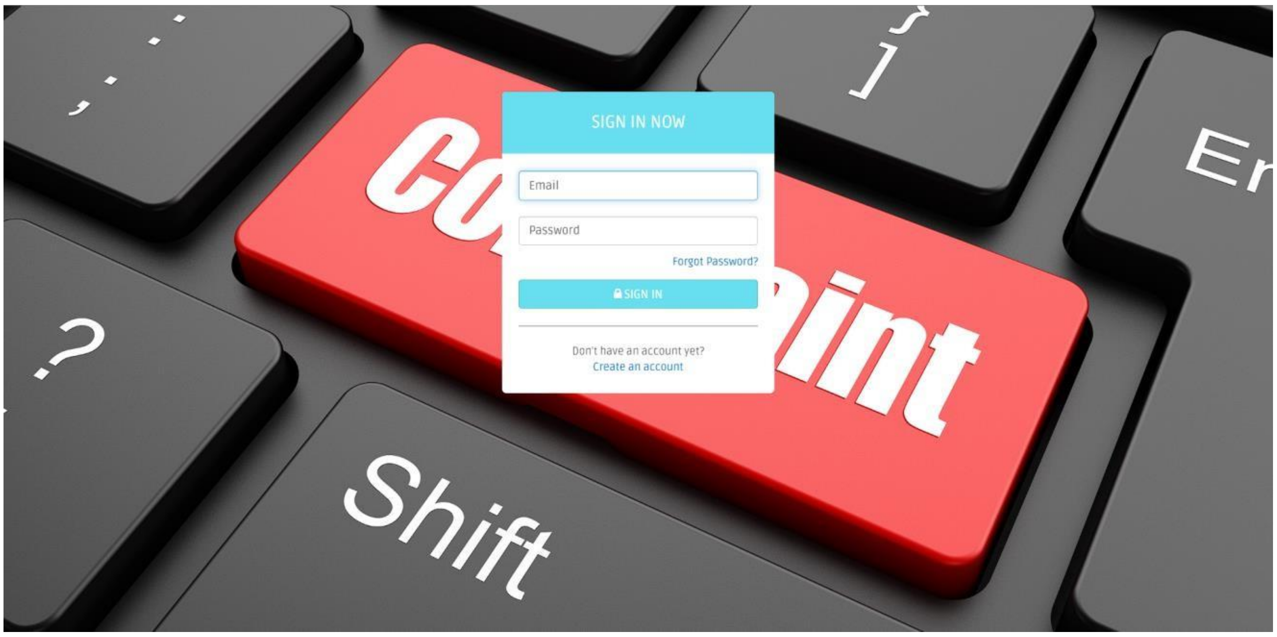


Fig: 7.3.2(User login page)

USER DASHBOARD PAGE:

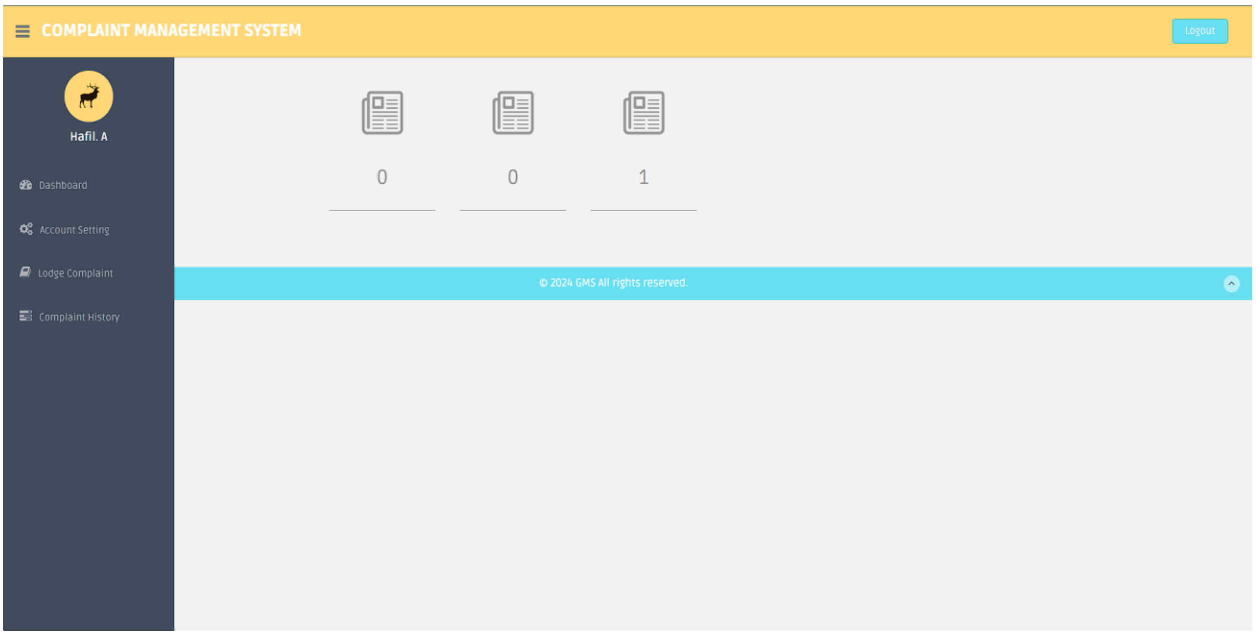
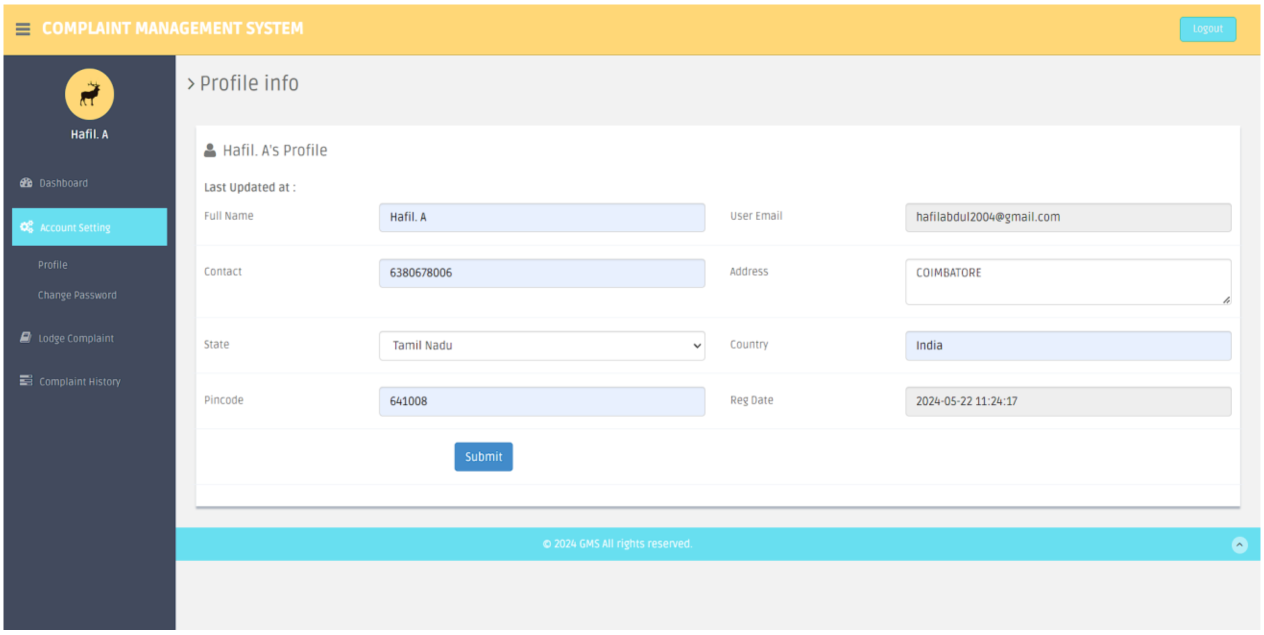


Fig: 7.3.3(User dashboard page)

USER PROFILE PAGE:



> Profile info

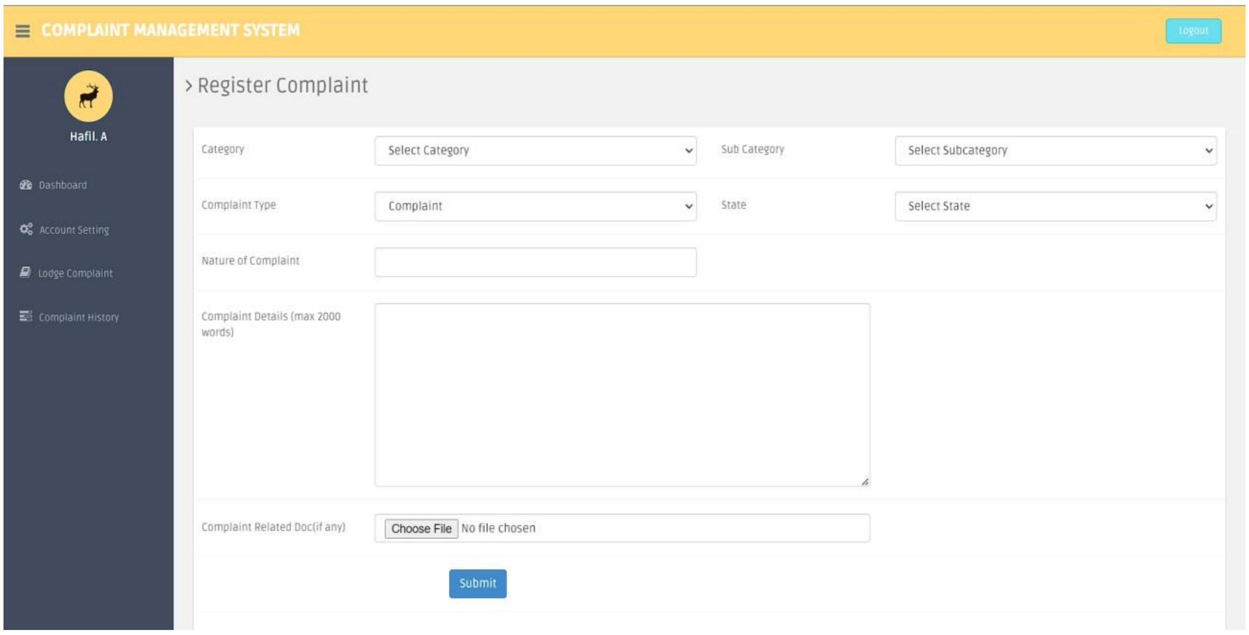
Hafil. A's Profile

Last Updated at :

Full Name	Hafil. A	User Email	hafilabdul2004@gmail.com
Contact	6380678006	Address	COIMBATORE
State	Tamil Nadu	Country	India
Pincode	641008	Reg Date	2024-05-22 11:24:17

Fig: 7.3.4(User profile page)

LODGE COMPLAINT PAGE:

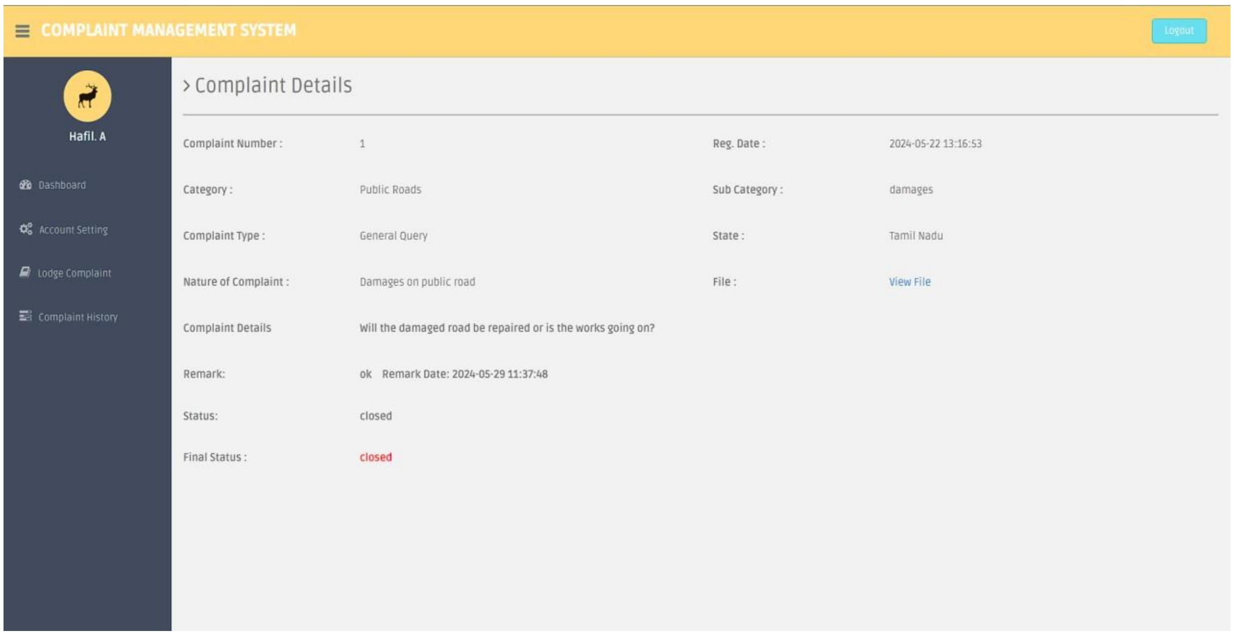


> Register Complaint

Category	Select Category	Sub Category	Select Subcategory
Complaint Type	Complaint	State	Select State
Nature of Complaint			
Complaint Details (max 2000 words)			
Complaint Related Doc(if any)	No file chosen		

Fig: 7.3.5(Lodge complaint page)

COMPLAINT HISTORY PAGE:



> Complaint Details

Complaint Number :	1	Reg. Date :	2024-05-22 13:16:53
Category :	Public Roads	Sub Category :	damages
Complaint Type :	General Query	State :	Tamil Nadu
Nature of Complaint :	Damages on public road	File :	View File
Complaint Details	Will the damaged road be repaired or is the works going on?		
Remark:	ok Remark Date: 2024-05-29 11:37:48		
Status:	closed		
Final Status :	closed		

Fig: 7.3.6(Complaint history page)

ADMIN LOGIN PAGE:

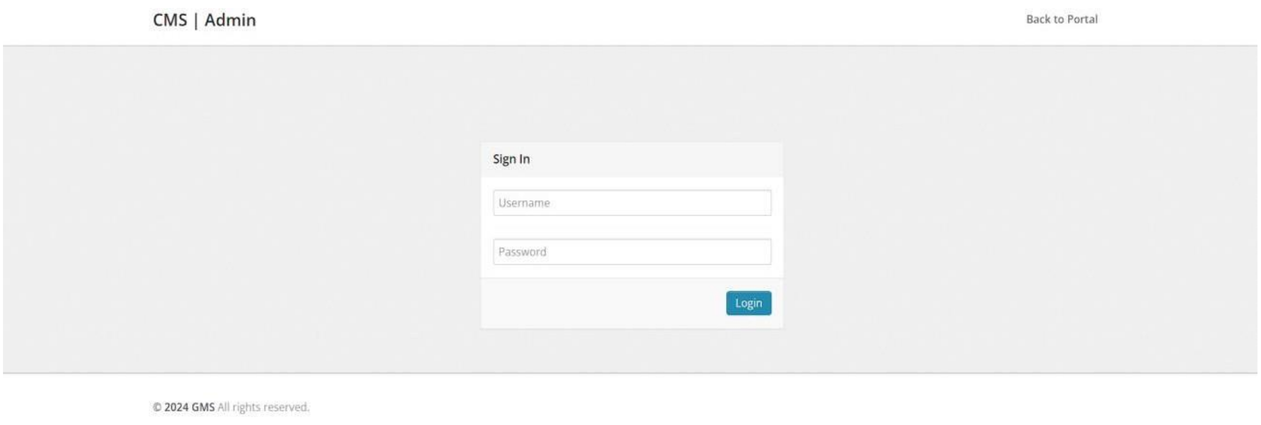
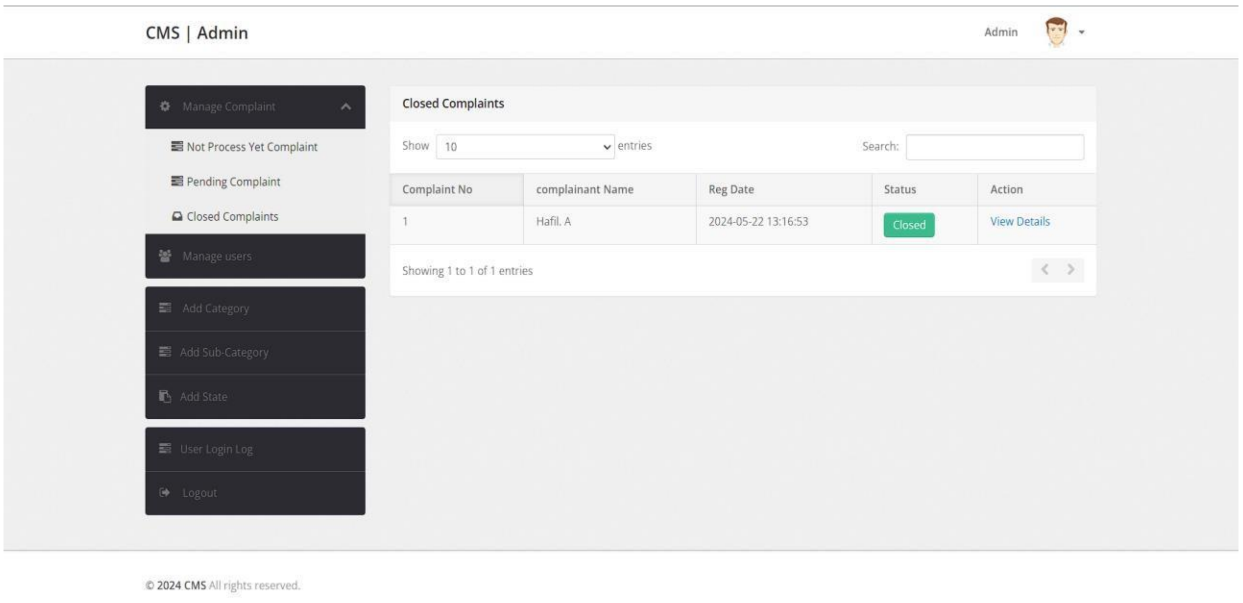


Fig: 7.3.7(Admin login page)

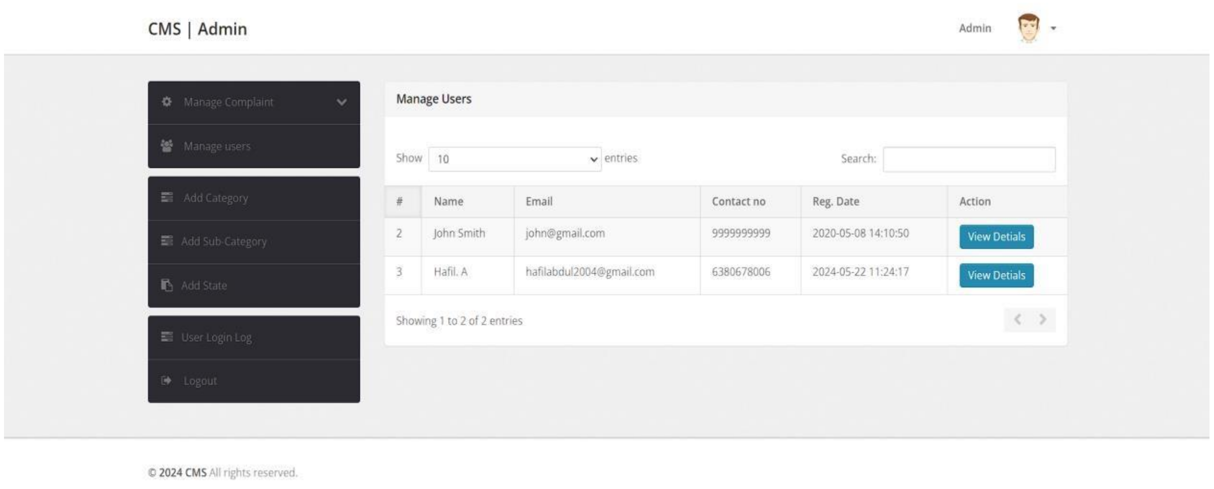
ADMIN DASHBOARD PAGE:



Complaint No	complainant Name	Reg Date	Status	Action
1	Hafil. A	2024-05-22 13:16:53	Closed	View Details

Fig: 7.3.8(Admin dashboard page)

ADMIN MANAGE USERS PAGE:



Manage Users

Show 10 entries Search:

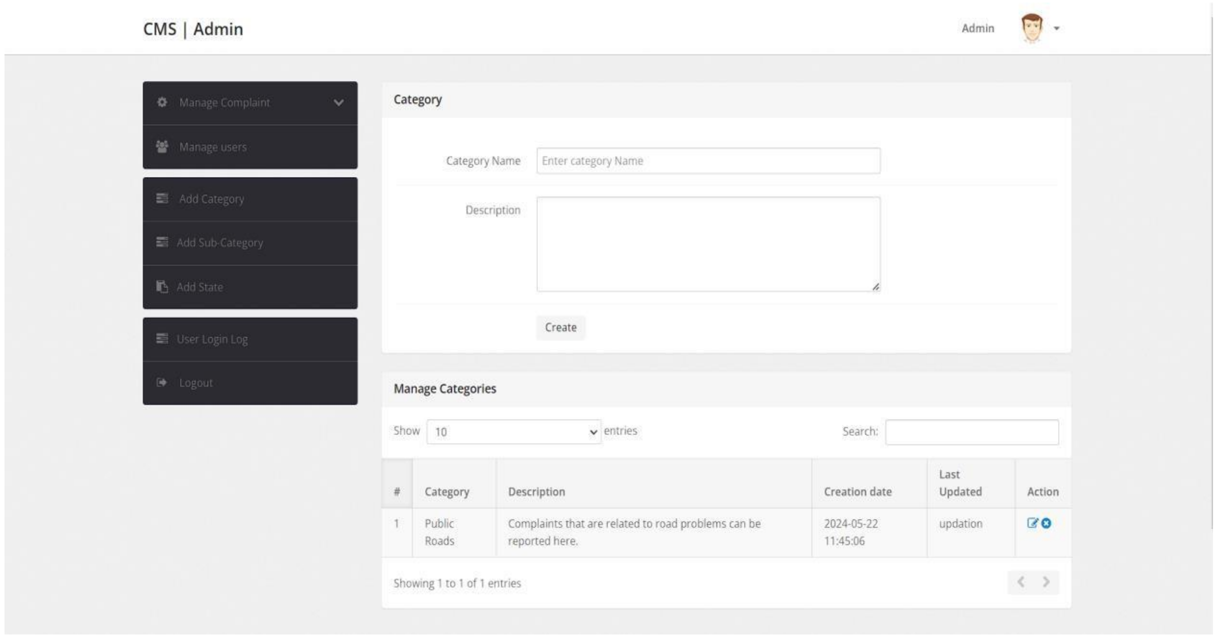
#	Name	Email	Contact no	Reg. Date	Action
2	John Smith	john@gmail.com	9999999999	2020-05-08 14:10:50	View Detials
3	Hafil. A	hafilabdul2004@gmail.com	6380678006	2024-05-22 11:24:17	View Detials

Showing 1 to 2 of 2 entries

© 2024 CMS All rights reserved.

Fig: 7.3.9(Admin manage user page)

ADMIN ADD CATEGORY PAGE:



Category

Category Name: Enter category Name

Description:

Create

Manage Categories

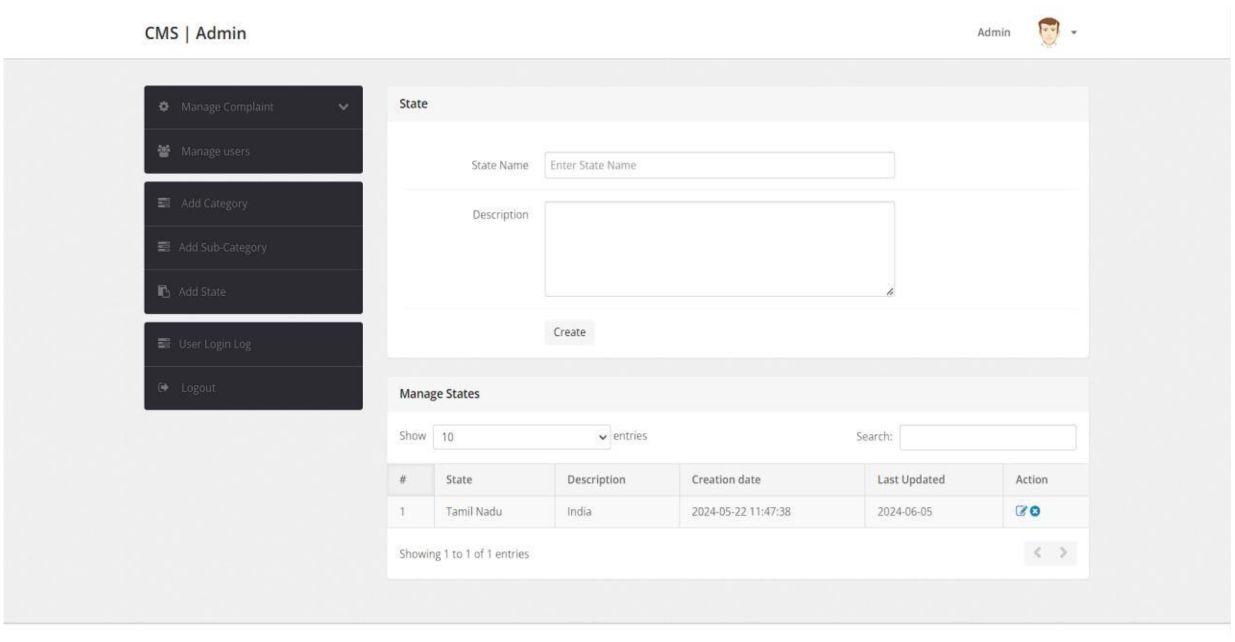
Show 10 entries Search:

#	Category	Description	Creation date	Last Updated	Action
1	Public Roads	Complaints that are related to road problems can be reported here.	2024-05-22 11:45:06	updation	

Showing 1 to 1 of 1 entries

Fig: 7.3.10(Admin add category page)

ADMIN ADD STATE PAGE:



State Name: Enter State Name

Description:

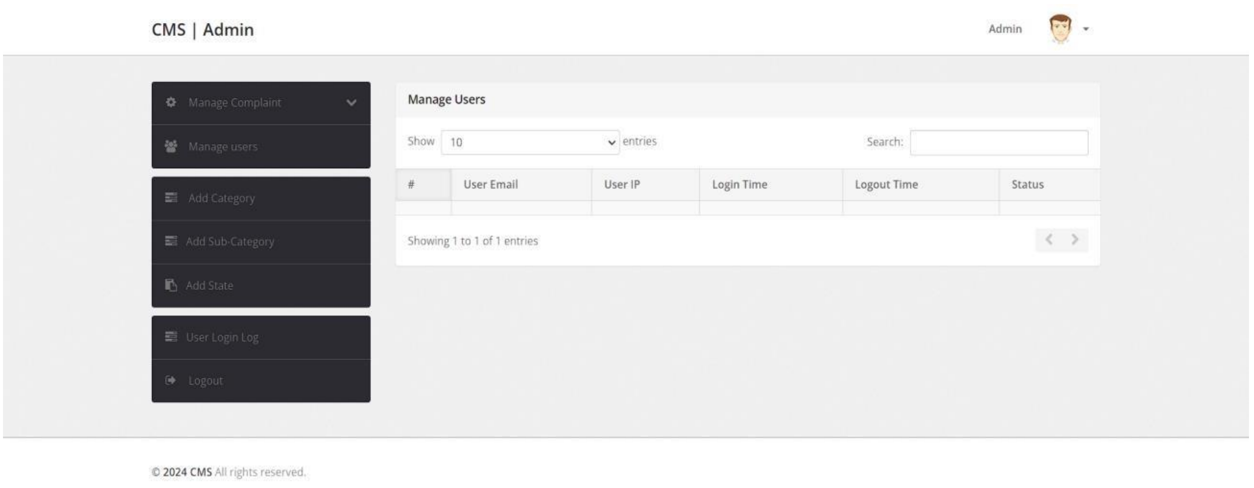
Show 10 entries Search:

#	State	Description	Creation date	Last Updated	Action
1	Tamil Nadu	India	2024-05-22 11:47:38	2024-06-05	

Showing 1 to 1 of 1 entries

Fig: 7.3.11(Admin add state page)

ADMIN USER LOGIN LOG PAGE:



Show 10 entries Search:

#	User Email	User IP	Login Time	Logout Time	Status

Showing 1 to 1 of 1 entries

© 2024 CMS All rights reserved.

Fig: 7.3.12(Admin user login log page)