**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANA SANGAMA, Belagavi - 590 018.**



**2022- 2023**

**DBMS** **Mini Project Report on**

**“IT SERVICE MANAGEMENT SYSTEM”**

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

**BACHELOR OF ENGINEERING**

in

**INFORMATION SCIENCE & ENGINEERING**

**Submitted by**

**PARVEEN SIDDIQA (1AT20IS057)**

**SHAHISTA ANJUM (1AT20IS084)**

**SHARANYA AVADHANI M.U (1AT20IS085)**

**Under the guidance of**

**Dr. T N Anitha**

Professor

Dept. of ISE, ATRIA I. T.

**Ms. Malini R**

Assistant Professor

Dept. of ISE, ATRIA I. T.

**Mrs. Asma Begum Altaf**

Assistant Professor

Dept. of ISE, ATRIA I. T.



**ATRIA INSTITUTE OF TECHNOLOGY**

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

Anandnagar, Bengaluru-560 024

**ATRIA INSTITUTE OF TECHNOLOGY**

**(Affiliated to Visvesvaraya Technological University)**

**ASKB Campus, Anandnagar,**

**Bengaluru–560024**

**Department of Information Science and Engineering**



**CERTIFICATE**

Certified that the project work entitled **“IT SERVICE MANAGEMENT SYSTEM”** carried out by **PARVEEN SIDDIQA (1AT20IS057), SHAHISTA ANJUM (1AT20IS084), SHARANYA AVADHANI M.U (1AT20IS085)** are bonafide students of **ATRIA INSTITUTE OF TECHNOLOGY**, Bengaluru, in partial fulfillment for the award of Degree of **Bachelor of Engineering** in **Information Science & Engineering** of **Visvesvaraya Technological University, Belagavi,** during the academic year **2022-23**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

|  |  |
| --- | --- |
| **Guide** | **HOD** |
| **Dr. T N Anitha**  (Professor., ISE)    **Ms. MALINI R** | **Dr. Shanthi Mahesh** |
| (Asst. Prof., ISE)  **Mrs. ASMA BEGUM**  (Asst. Prof., ISE) | (Dept. of ISE) |
|  | **External Viva** |
| **Name of Examiners**  **1.**  **2.** | **Signature with date** |

**DECLARATION**

We, **PARVEEN SIDDIQA (1AT20IS057), SHAHISTA ANJUM (1AT20IS084), SHARANYA AVADHANI M.U (1AT20IS085)** students of **5th semester Bachelor of Engineering, Department of Information Science and Engineering, Atria Institute of Technology, Bengaluru,** would hereby declare that the project entitled **“IT Service Management System”** has been carried out by us at **Atria Institute of Technology, Bengaluru,** and submitted in partial fulfillment of the course requirement for the award of degree of **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi,** during the academic year **2022-23.**

We further declare that the work embodied in this report has not been submitted to any other university or institution for the award of any other degree.

Place: Bengaluru Signature of the students

Date:

**PARVEEN SIDDIQA**

**(1AT20IS057)**

**SHAHISTA ANJUM**

**(1AT20IS084)**

**SHARANYA AVADHANI M.U (1AT20IS085)**

**ABSTRACT**

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

IT System, as described above, can lead to error free, secure, reliable and fast management system. The aim is to automate its existing manual system by the help of computerized equipment’s and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients.

# ACKNOWLEDGEMENT

We are grateful to our institution, **Atria Institute of Technology**, for having provided us with the facilities to successfully complete this mini project on IT SERVICE MANAGEMENT SYSTEM.

We thank **Dr. T N Srinivasa, principal** and **Dr. Shanthi Mahesh, HOD, ISE** for providing us all the necessary facilities for the successful completion of our mini-project.

Deadlines play a very important role in the successful completion of the academic project on time, efficiently and effectively. We take this opportunity to express our deep sense of gratitude to our guide and coordinators **Dr. T N Anitha, Professor, Mrs. Asma Begum and Ms. Malini R, Assistant Professor, Department of ISE** for their valuable guidance and help throughout the course of the academic mini-project. They have always been patient with us and helped immensely in completing the task on hand. We also thank them for their immense support, guidance, specifications & ideas without which seminar would have been completed without full merit.

Last but not least from the Department of Information Science and Engineering, teaching and non-teaching staffs for their constant encouragement, support, patience, and endurance shown during the preparation of this report were remarkable. We also thank the management.

Finally, we thank our parents and friends for their motivation, morale and material support.

**PARVEEN SIDDIQA**

**(1AT20IS057)**

**SHAHISTA ANJUM**

**(1AT20IS084)**

**SHARANYA AVADHANI M.U (1AT20IS085)**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
|  | **Declaration**  **Abstract**  **Acknowledgement** |  |
| **1** | **Introduction to database** | 1 |
|  | 1.1 Database environment system | 4 |
|  | 1.2 Advantages of using DBMS approach | 5 |
|  | 1.3 Architecture of Database | 6 |
| **2** | **Introduction to project** | 10 |
|  | 2.1 Overview of project | 10 |
|  | 2.1.1 Introduction | 10 |
|  | 2.1.2 Problem statement | 10 |
|  | 2.1.3 Reasons for project | 10 |
|  |  |  |
|  | 2.2 System services | 12 |
|  | 2.2.1 Working of services | 12 |
|  | 2.2.2 Benefits | 12 |
|  | 2.3 Table description | 13 |
|  | 2.3.1 Category | 13 |
|  | 2.3.2 Users | 13 |
|  | 2.3.3 TaskStatus | 14 |
|  | 2.3.4 Task | 14 |
|  | 2.3.5 TicketHistory | 15 |
|  |
|  |

|  |  |  |
| --- | --- | --- |
|  | **Design** | 16 |
|  | 3.1 System architecture | 16 |
| **3** | 3.2 Entity relationship diagram | 17 |
|  | **Requirements** | 19 |
|  | 4.1 Functional and non-functional requirements | 19 |
|  | 4.1.1 Functional requirements | 19 |
| **4** | 4.1.2 Nonfunctional requirements | 20 |
|  | 4.2 Hardware requirements | 21 |
|  | 4.3 Software requirements | 21 |
|  | 4.3.1 Server side | 21 |
|  | 4.3.1.1 Node js | 21 |
|  | 4.3.1.2 Web server | 21 |
|  | 4.3.1.3 HTML | 21 |
|  | 4.3.1.4 JavaScript | 22 |
|  | 4.4 Security Requirements | 22 |
|  | **CONCLUSION** | 23 |
|  | **REFERENCES** | 24 |
|  |  |  |
|  |  |  |
|  |  |  |

**APPENDIX**

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Title** | **Page No.** |
| A | SCREENSHOTS | 25 |
| A.1 | Home window | 25 |
| A.2 | Admin side login page | 25 |
| A.3 | Admin panel | 26 |
| A.4 | User Modelling Page | 26 |
| A.5 | Created users | 27 |
| A.6 | Task Category | 27 |
| A.7 | Task Status | 28 |
| A.8 | Task Assigned | 28 |
| A.9 | Task Raised | 29 |
| A.10 | Category Admin Panel | 29 |
| A.11 | Task Assignment | 30 |
| A.12 | User Side Login Page | 30 |
| A.13 | User Panel | 31 |
| A.14 | Task Panel | 31 |
| A.15 | Assigned Task | 32 |
| A.16 | Assigned Task Details | 32 |
| B | CODE SNIPPETS | 33 |
| B.1 | Database connection | 33 |
| B.2 | Insert Query | 33 |
| B.3 | Delete query | 34 |
| B.4 | Update Query | 34 |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Title** | **Page No.** |
| 1.1 | Simplified database environment system | 5 |
| 1.2 | Architecture of DBMS | 9 |
| 3.1 | System architecture of CRMS | 16 |
| 3.2 | E R of CRMS | 18 |
| A.1 | Home window | 25 |
| A.2 | Admin side login page | 25 |
| A.3 | Admin panel | 26 |
| A.4 | User Modelling Page | 26 |
| A.5 | Created users | 27 |
| A.6 | Task Category | 27 |
| A.7 | Task Status | 28 |
| A.8 | Task Assigned | 28 |
| A.9 | Task Raised | 29 |
| A.10 | Category Admin Panel | 29 |
| A.11 | Task Assignment | 30 |
| A.12 | User Side Login Page | 30 |
| A.13 | User Panel | 31 |
| A.14 | Task Panel | 31 |
| A.15 | Assigned Task | 32 |
| A.16 | Assigned Task Details | 32 |
| B.1 | Database connection | 33 |
| B.2 | Insert Query | 33 |
| B.3 | Delete query | 34 |
| B.4 | Update Query | 34 |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Title** | **Page No.** |
| 2.3.1 | Category | 13 |
| 2.3.2 | Users | 13 |
| 2.3.3 | TaskStatus | 14 |
| 2.3.4 | Task | 14 |
| 2.3.5 | TaskHistory | 15 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**CHAPTER 1**

**INTRODUCTION TO DATABASE**

Database and database technology have a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, engineering, medicine, genetics, law, education, and library science. The word database is so commonly used that we must begin by defining what the database is.

Our initial definition is quite general. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using personal computers and software such as Microsoft excel. This collection of related data with an implicit meaning is a database.

The preceding definition of a database is quite general, for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted. A database has the following properties:

* A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse. The changes to the mini world are reflected in the database.
* A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
* A database is designed, built and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents. The end-users of the database may perform business transactions (for example a customer buys a camera) or events may happen that may cause the information in the database to change. In order for a database to be accurate and reliable at all times, it must be a true reflection of the mini world that it represents; therefore changes must be reflected in the database as soon as possible.

A database can be of any size and complexity. A database may be generated and maintained manually or computerized. For example, a library card catalog is a database that may be created and maintained manually. A computerized database may be created and maintained either by a group of application programs written specifically for that task or by a database management system.

A database is a collection of data, typically describing the activities of one or more related organizations. For example, a university database might contain information about the following:

* Entities such as students, faculty, courses, and classrooms.
* Relationships between entities, such as student’s enrolment in courses, faculty teaching courses, and the use of rooms for courses.

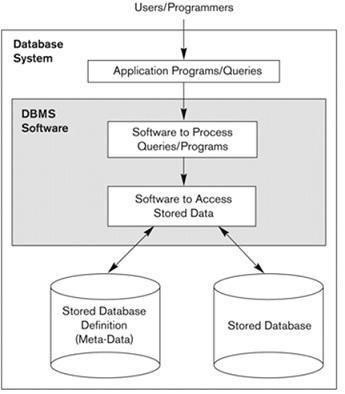
A database management system, or DBMS, is software designed to assist in maintaining and utilizing a large collection of data. The need for such systems as well as their use is growing rapidly. The alternative to using a DBMS is to store the data in files and write application-specific code to manage it.

**File system versus DBMS**

To understand the need for a DBMS, let us consider a motivating scenario: a company has a large collection (say 500 GB) of data on employees, departments, products, sales, and so on. This data is accessed concurrently by several employees. Questions about the data must be answered quickly, changes made to the data by different users must be applied consistently and access to certain parts of the data must be restricted. We can try to manage the data by storing it in operating system files.

This approach has many drawbacks, including the following: -

* We probably do not have 500GB of main memory to hold all the data. We must, therefore, store data in a storage device such as a disk or tape and bring relevant parts into the main memory for processing as needed.
* Even if we have 500 GB of main memory, on computer systems with 32 bit addressing, we cannot refer directly to more than about 4 GB of data. We have to program some method of identifying all data items
* We have to write special programs to answer each question a user may want to ask about the data. These programs are likely to be complex because of the large volume of data to be searched.
* We must protect the data from the inconsistent changes made by different users accessing the data concurrently. If applications must address the details of such concurrent access, this adds greatly to their complexity.
* We must ensure that the data is restored to a consistent state if the system crashes while changes are being made.
* Operating systems provide only a password mechanism for security. This is not sufficiently flexible to enforce security policies in which different users have permission to access different subsets of the data.
* DBMS is a piece of software designed to make the preceding tasks easier. By storing data in DBMS rather than as a collection of operating system files, we can use the DBMS’s features to manage the data in a robust and efficient manner. As the volume of data and the number of users grow hundreds of gigabytes of data and thousands of users are common in current corporate database DBMS support becomes indispensable.

* 1.  **DATABASE ENVIRONMENT SYSTEM**

**Fig 1.1: Simplified database environment system**

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilities the processes of defining, constructing, manipulating and sharing databases among various users and applications. Defining a database involves specifying the data types, structures and constraints of the data to be stored in the database.

The database definition or description information is also stored by the DBMS in the form of a database catalog or dictionary, it is called Metadata. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

An application program accesses the database by sending queries or requests for data to DBMS. A query typically causes some data to be retrieved; a transaction may cause some data to be read and some data to be written into the database.

Other important functions provided by DBMS include protecting the database and maintaining it over a long period of time, protection includes system protection against hardware or software malfunction and security protection against unauthorized or malicious access. A typical large database may have a life cycle of many years, so the DBMS must be able to maintain the database system by allowing the system to evolve as requirements change over time.

It is not absolutely necessary to use general-purpose DBMS software to implement a computerized database. We could write our own set of programs to create and maintain the database, in effect creating our own special purpose DBMS software. In either case, whether we use a general-purpose DBMS or not we usually have deployed a considerable amount of complex software. In fact, most DBMSs are very complex software systems. Fig 1.1 shows a simplified database environment system.

* 1. **ADVANTAGES OF USING DBMS APPROACH**

Using a DBMS to manage data has many advantages: -

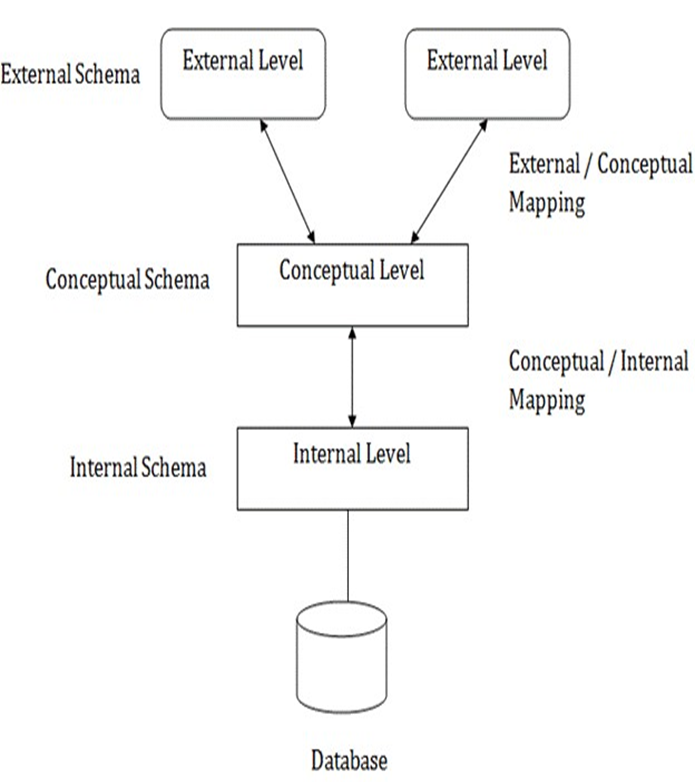
* **Data Independence:** application program should not, ideally, be expected to detailsof data representation and storage, the DBMS provides an abstract view of the data that hides such details.
* **Efficient Data Access:** A DBMS utilizes a variety of sophisticated techniques to storeand retrieve data efficiently. This feature is especially important if the data is to be stored on an external device.
* **Data Integrity and Security:** if data is always accessed through DBMS, the DBMScan enforce integrity constraints. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, it can enforce access controls that govern what data is visible to different classes of users.
* **Data Administration:** when several users share data, centralizing the administrationof data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, it can be responsible for organizing the data representation to minimize redundancy and for finetuning the storage of the data to make retrieval efficient.
* **Concurrent Access and Crash Recovery:** A DBMS schedules concurrent accessesto the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures
* **Reduced Application Development Time:** clearly, the DBMS supports importantfunctions that are common to many applications accessing data in the DBMS. This, in conjunction with the high-level interface to data, facilities quick application development. DBMS applications are also likely to be more robust than a similar standalone application because many important tasks are handled by the DBMS.

**1.3 ARCHITECTURE OF DATABASE**

**The Three-Schema Architecture**: -

The goal of the three-schema architecture illustrated in the figure is to separate the user application from the physical database. In this architecture, schemas can be defined at the following three levels:

* The internal level has an internal schema, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.
* The conceptual level has a conceptual schema, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. Usually, a representational data model is used to describe the conceptual schema when a database system is implemented. This implementation conceptual schema is often based on a conceptual schema design in a high-level data model.
* The external or view level includes a number of external schemas or user views. Each external schema describes the part of a database that a particular user group is interested in and hides the rest of the database from that user group. As in the previous level, each external schema is typically implemented using a representational data model, possibly based on external schema design in a high-level data model.

****

**Fig 1.2: Architecture of DBMS**

**CHAPTER 2**

**INTRODUCTION TO PROJECT**

**2.1 Overview of the project**

**2.1.1 Introduction**

Service Desk is a communication center that provides a Single Point of Contact (SPOC). The purpose of a service desk is to ensure that users receive appropriate help in a timely manner. This project aims at digitizing the problem reporting and solving domain. This aims at handling the events(problems) that are faced by the users in a company in their day to day environment. There are many tools available in the market which use the ITIL technology. The ITIL technology has many modules and is complicated. Small and medium sized companies cannot afford those tools. This is a simple tool with minimum number of modules which aims at solving the problems of the small and medium sized companies.

**2.1.2 Problem statement**

There is a need for a simple tool which digitizes the ticketing services of the industries. It enables the user to report his/her problem to the admin of the corresponding sector and enables the admin to assign those tasks to another user who is expected to handle it. A database is designed to maintain all these transactions.

**2.1.3 Reason for the Project**

The advancement in Information Technology and internet penetration has greatly enhanced various business processes and communication between companies (services provider) and their customers of which issues is not left out. This IT System is developed to provide the following services:

* Fixing things that are broken: Some things wear out and break over time, others are flawed from day-1. Personal and business environments are full of things, activities, interactions and processes that are broken or not operating in the way they are desired to work. Problem-solving gives us a mechanism for identifying these things, figuring out why they are broken and determining a course of action to fix them.
* Addressing risk: Humans have learned to identify trends and developed an awareness of cause-and-effect relationships in their environment. These skills not only enable us to fix things when they break but also anticipate what may happen in the future (based on past-experience and current events). Problem-solving can be applied to the anticipated future events and used to enable action in the present to influence the likelihood of the event occurring and/or alter the impact if the event does occur.
* Improving Performance: Individuals and organizations do not exist in isolation in the environment. There is a complex and ever-changing web of relationships that exist and as a result, the actions of one person will often have either a direct impact on others or an indirect impact by changing the environment dynamics. These interdependencies enable humans to work together to solve more complex problems but they also create a force that requires everyone to continuously improve performance to adapt to improvements by others. Problem-solving helps us understand relationships and implement the changes and improvements needed to compete and survive in a continually changing environment.
* Seizing Opportunity : Issues Resolving isn’t just about responding to (and fixing) the environment that exists today. It is also about innovating, creating new things and changing the environment to be more desirable. IT Service Management enables us to identify and exploit opportunities in the environment and exert (some level of) control over the future.

**2.2 Services Provided**

**2.2.1 How the Services Work:**

Whenever there is an issue encountered by the user, the user can raise a ticket regarding the issue. The database is responsible for collecting these tickets and making it feasible for the tasks to be allocated by the admin to the respective users responsible for the issue. The system also keeps track of how, when the task is completed and allows comments by the user.

**2.2.2 Benefits of IT Service Management System:**

1. Digitizes the ticketing services of the industries.

2. Simple and affordable to the small and medium size companies.

3. Gives access to all of your incidents in one place.

4. Tracks the requests received until they are resolved.

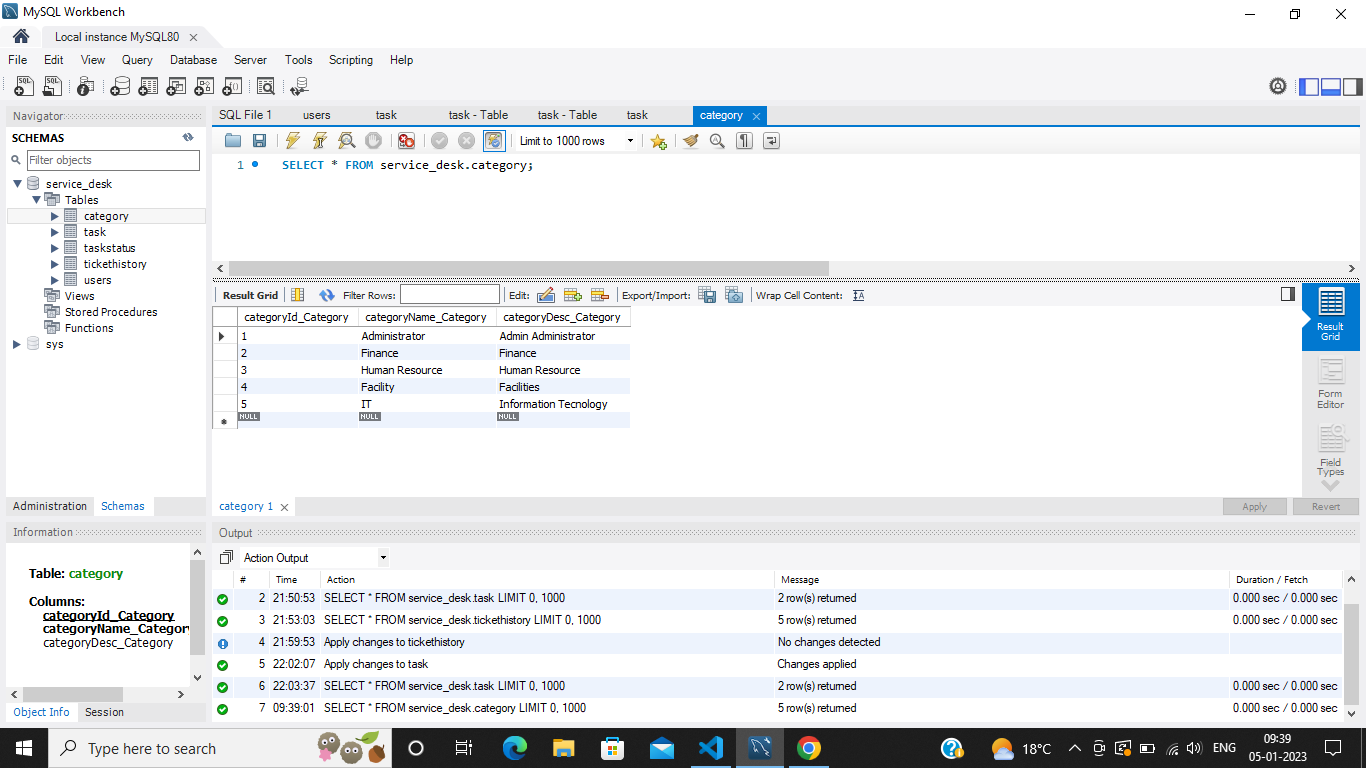
5. It helps the admins to resolve the issues faced by the users in a timely manner.

**2.3 TABLE DESCRIPTION**

**2.3.1 TABLE CATEGORY**

Category table has the details of all the categories and it has attributes id, name, and description.

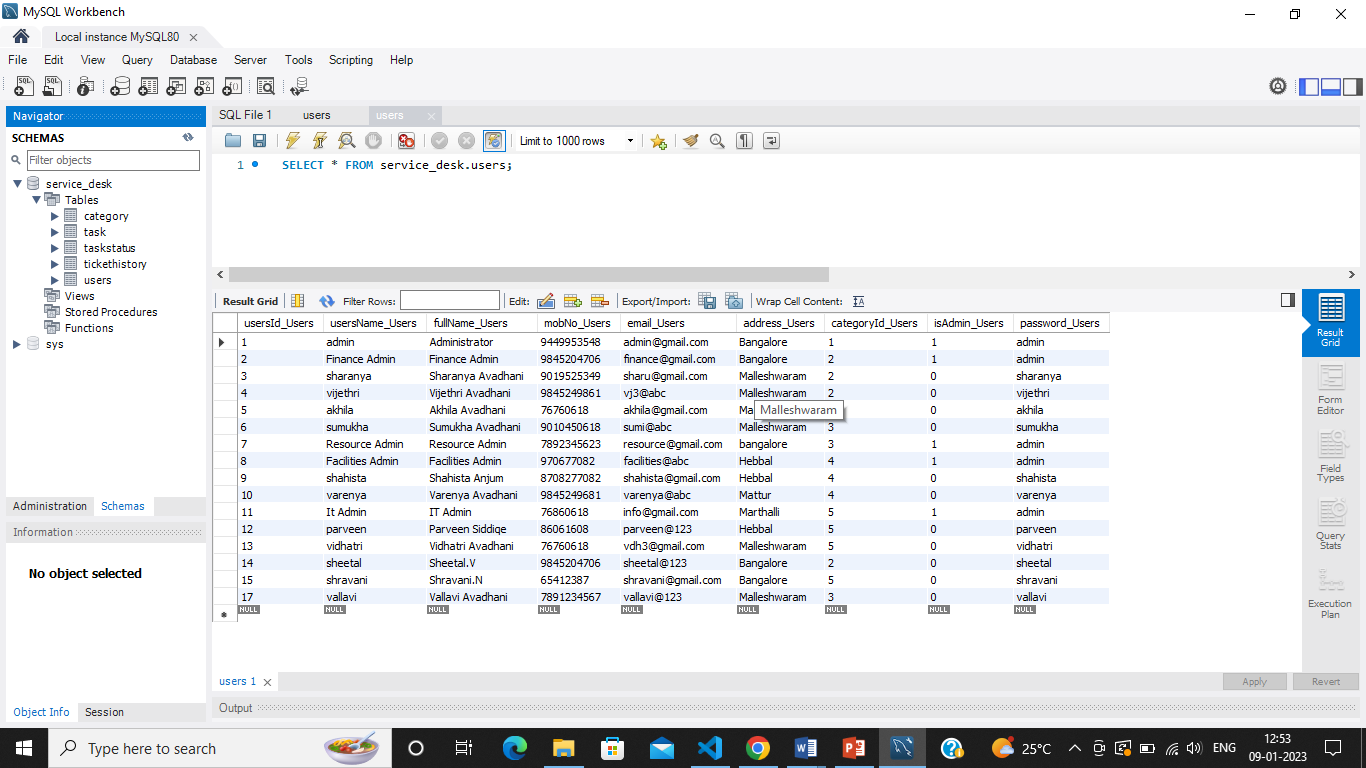
**Table 2.3.1 Structure of Category**



**2.3.2 TABLE USERS**

User table has the details of all the users and it has attributes id, name, fullname, mobile number, email, address, and passwords.

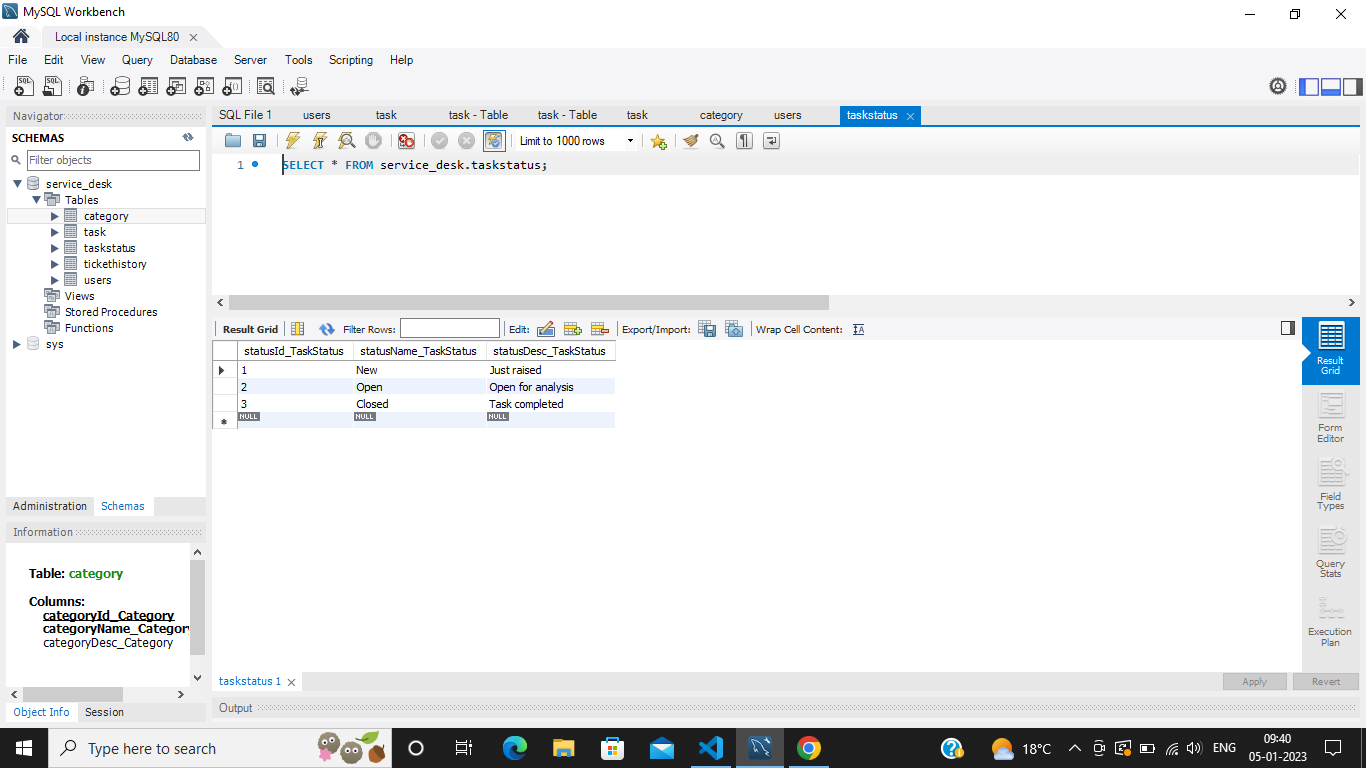
**Table 2.3.2 Structure of Users**



**2.3.3 TABLE TASKSTATUS**

TaskStatus table has the attributes id, name and description.

**Table 2.3.3 Structure of TaskStatus**



**2.3.4 TABLE TASK**

Task table has the attributes id, name, description, category\_id, admin task id, from user\_id, to user\_id, start date, end date, actual end date, status\_id.

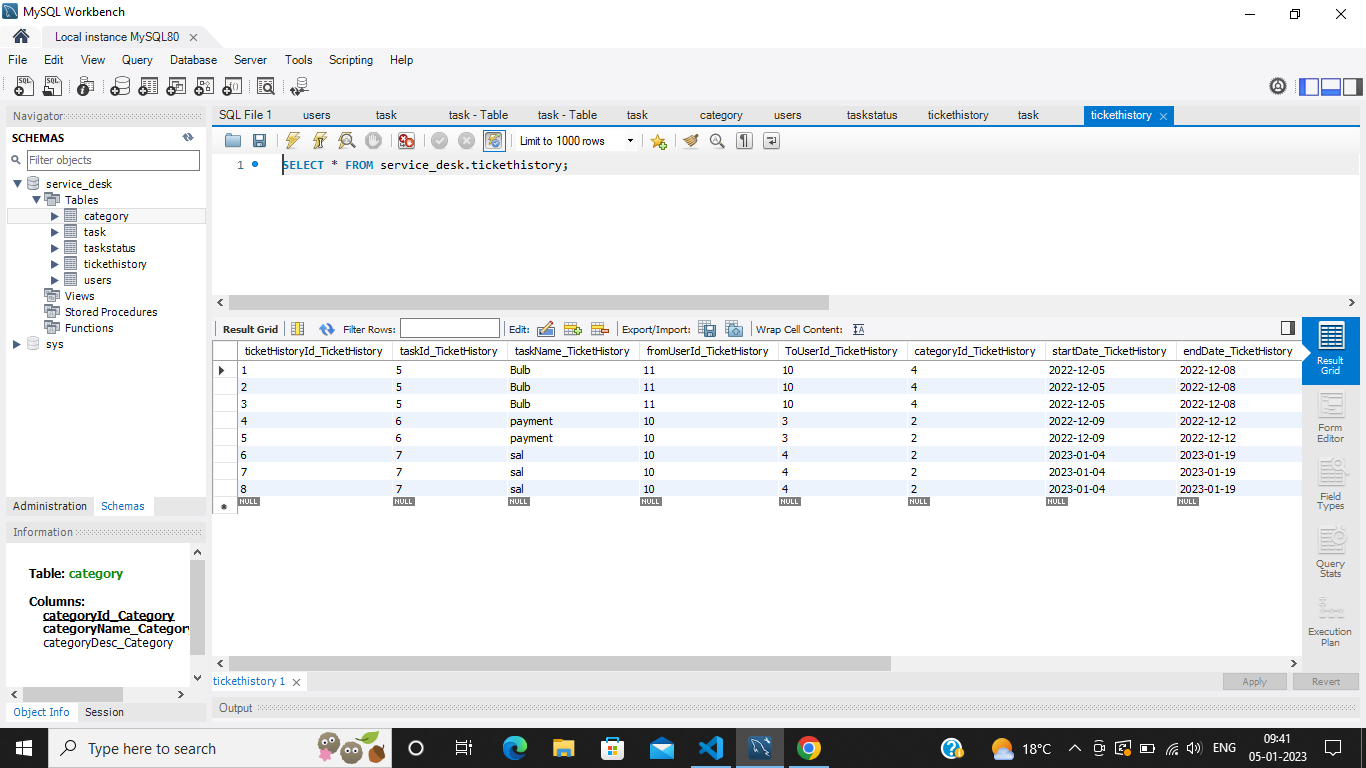
**Table 2.3.4** **Structure of Task**



**2.3.5 TABLE TICKETHISTORY**

TicketHistory table has the attributes id, brand name, creation date, updating date.

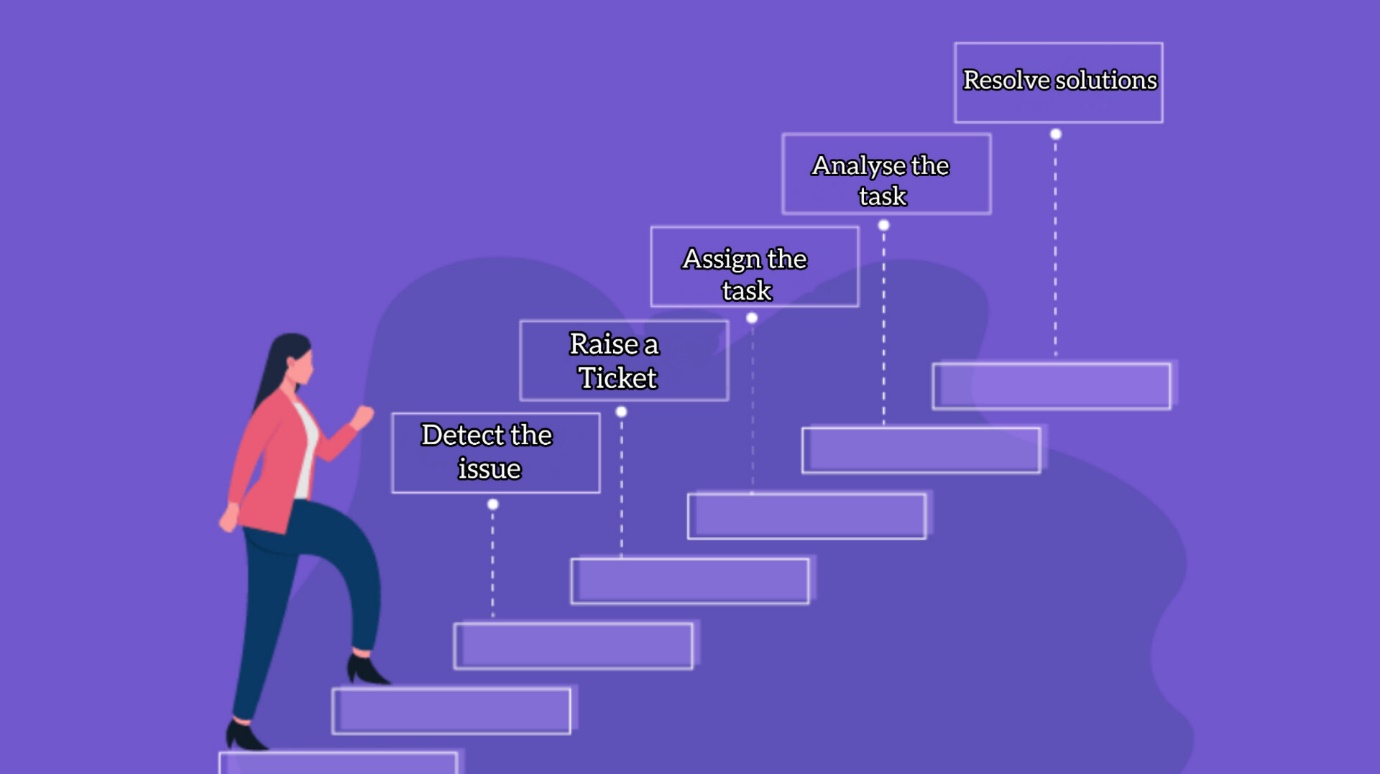
**Table 2.3.5 Structure of TicketHistory**



**CHAPTER 3**

**DESIGN**

**3.1 SYSTEM ARCHITECTURE**



The above figure describes the system architecture of IT Service Management System. It describes how a user uses a system to accomplish a particular goal. System Architecture helps to ensure that the correct system is developed by capturing the requirements from the user's point of view.

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

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

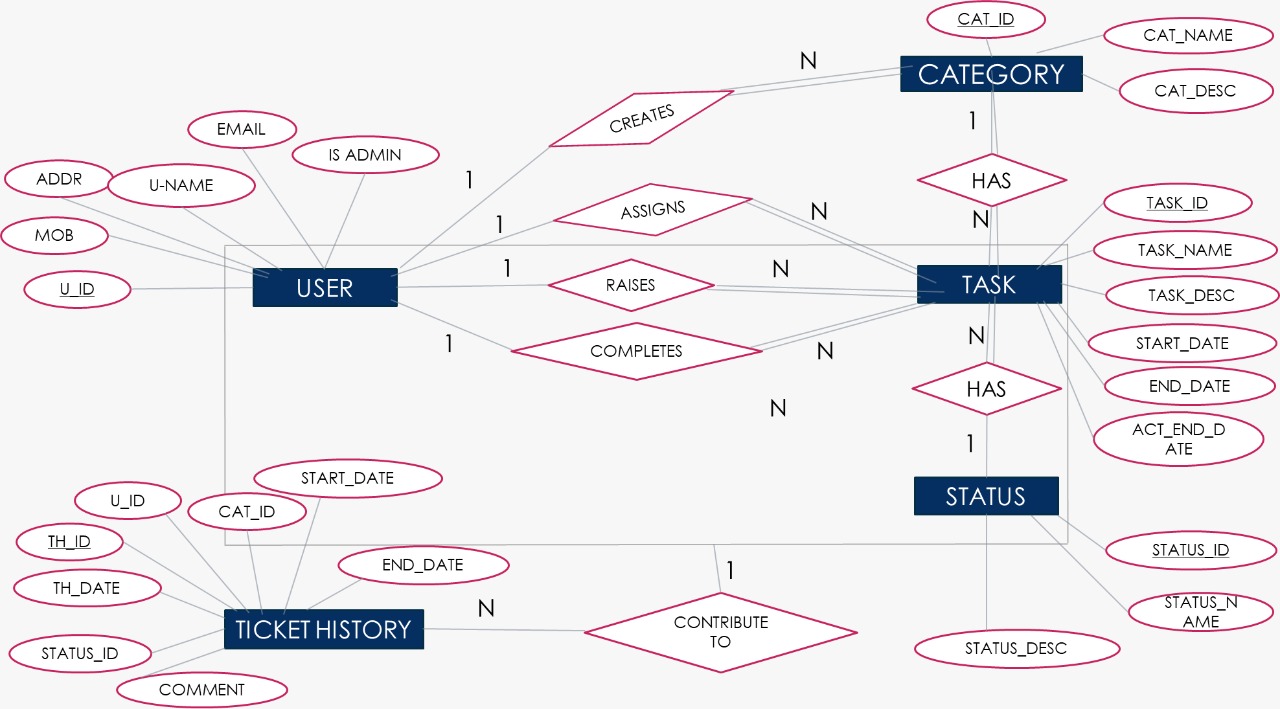
IT Service Management System is developed to produce a web-based system that allow employees to raise a ticket for an issue they are facing, for the company to effectively manage their hygiene and technical need to ease employee’s task whenever they need to raise an issue.

A System Architecture is a standardized set of concepts, practices, and criteria for dealing with a common type of problem, which can be used as a reference to help us approach and resolve new problems of a similar nature.

The aim of System Architecture is to provide a common structure so that developers don’t have to redo it from scratch and can reuse the code provided. In this way, frameworks allow us to cut out much of the work and save a lot of time.

**3.2 ENTITY RELATIONSHIP DIAGRAM**

An entity-relationship model describes inter-related things of interest in specific domain of knowledge. An ER module is composed of entity types and specifies relationships that can exist between instances of those entity types. It is a data modeling technique that graphically illustrates an information systems entity and the relationship between those entities.



**Fig 3.2 ER DIAGRAM-IT SERVICE MANAGEMENT SYSTEM**

**CHAPTER 4**

**REQUIREMENTS**

**4.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

**4.1.1 Functional Requirements Requirement**

Functional Requirements Requirement analysis is a software engineering technique that is composed of the various tasks that determine the needs or conditions that are to be met for a new or altered product, taking into consideration the possible conflicting requirements of the various users.

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data should the system holds and the interfaces with the user.

The functional requirements identified are:

* Raising Ticket: The system should allow the users to raise the ticket for the particular issue they are facing.
* Tasks Assignment: The database should display from which user the task is asked for and to which user it is assigned.
* Automatic update to database once tickets raised: Whenever there’s new ticket, the system should be able update the database without any additional efforts from the admin.
* Admin Duties: The admin of the particular category is responsible for assigning tasks to the respective employees of that category.
* Feedbacks to users: It should provide means for users to leave feedback.

**4.1.2 Non-Functional Requirements**

It describes aspects of the system that are concerned with how the system provides the functional requirements.

They are:

* Security: The subsystem should provide a high level of security and integrity of the data held by the system, only authorized personnel of the company can gain access to the company’s secured page on the system; and only users with valid password and username can login to view user’s page.
* Performance and Response time: The system should have high performance rate when executing user’s input and should be able to provide feedback or response within a short time span usually 50 seconds for highly complicated task and 20 to 25 seconds for less complicated task.
* Error handling: Error should be considerably minimized and an appropriate error message that guides the user to recover from an error should be provided. Validation of user’s input is highly essential. Also, the standard time taken to recover from an error should be 15 to 20 seconds.
* Availability: This system should always be available for access at 24 hours, 7 days a week. Also, in the occurrence of any major system malfunctioning, the system should be available in 1 to 2 working days, so that the business process is not severely affected.
* Ease of use: Considered the level of knowledge possessed by the users of this system, a simple but quality user interface should be developed to make it easy to understand and required less training.

**4.2 HARDWARE REQUIREMENTS**

A desktop or laptop with a proper internet connection with a hard disk of size 250GB or 60GB with a RAM of 3.4GB 8GB is required. The system can have

Windows 7 or 8 or 10 Operating system.

**4.3 SOFTWARE REQUIREMENTS**

**4.3.1 SERVER SIDE**

The programming language used on the server side is JAVASCRIPT, with the incorporating web server NODE JS and database used is MYSQL 8.0 CE.

**4.3.2 CLIENT SIDE**

The programming language used are JAVASCRIPT, HTML, CSS with an operating system of windows7/8/10.

**4.3.1.3 HTML**

HTML is an acronym that stands for Hypertext Markup Language.

* Hypertext: Hypertext simply means "Text within Text". A text has a link within it, is a hypertext. Every time you click on a word that brings you to a new webpage, you have clicked on a hypertext.
* Markup language: A markup language is a programming language that is used to make text more interactive and dynamic. It can turn a text into images, tables, links, etc. An HTML document is made of many HTML tags and each HTML tag contains different content.

**4.3.1.4 JAVASCRIPT**

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

**4.4 SECURITY REQUIREMENTS**

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below.

Ascertain functions to different modules, restrict communication between areas of the program, check data integrity for critical variables, user/admin authentication process are some of the security function embedded in our project.

**CONCLUSION**

IT Service Management System plays a critical role in supporting the IT company functions and satisfying employee/user requirements. As all companies and disciplines move toward a service orientation, service desk provides direction in that move for basic operations. A company as a whole can use the service desk to resolve issues and support employees for efficient working and control of tasks and problems. The main goal of our system is to manage the IT infrastructure while better aligning IT with organizational objectives.

**REFERENCES**

**Books:**

* Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe 7th edition,2017, Pearson.

• Software Engineering - R.S. Pressman

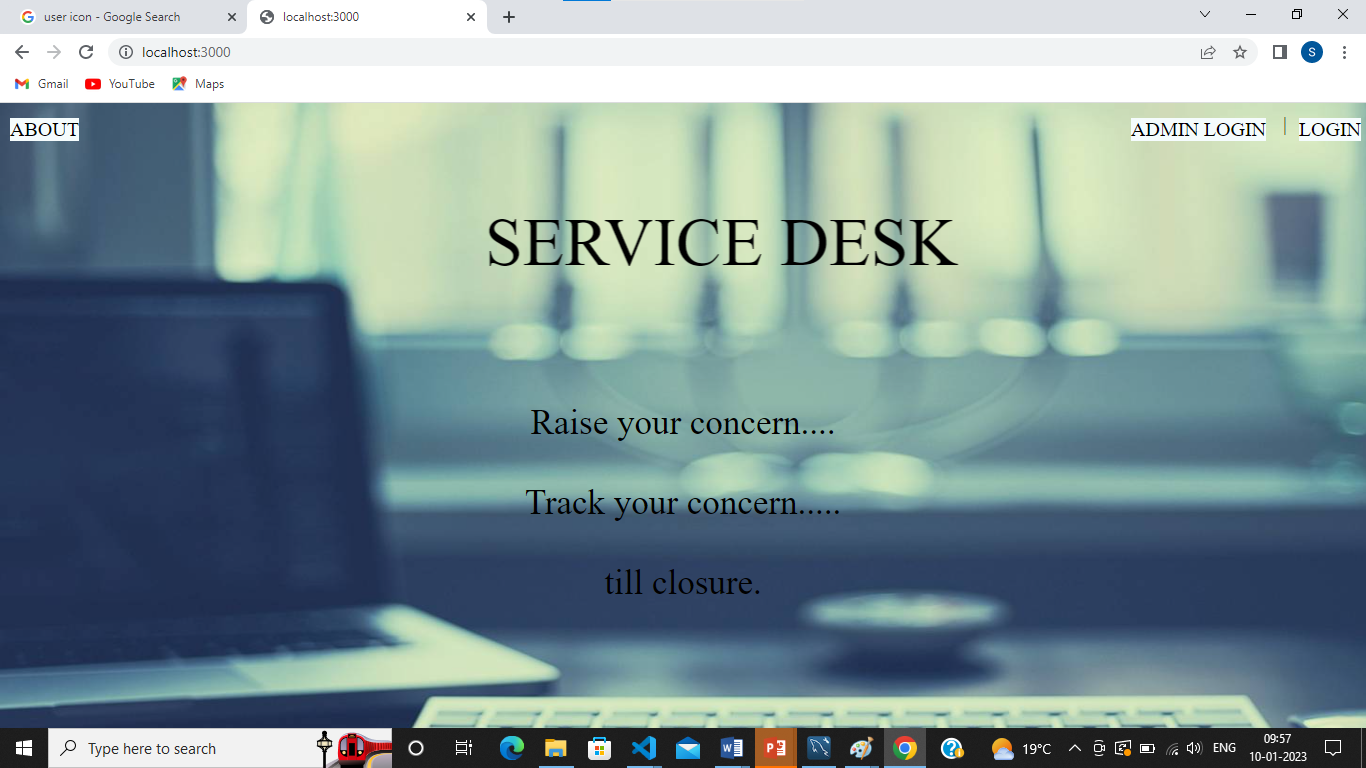
**Links:**

* <https://kepner-tregoe.com/>
* fayllor.org
* <https://www.w3schools.com/html/>
* <https://www.javatpoint.com/html-tutorial>
* https://html.com/

**APPENDIX ‘A’– SCREENSHOTS**

**A.1 Home Window**

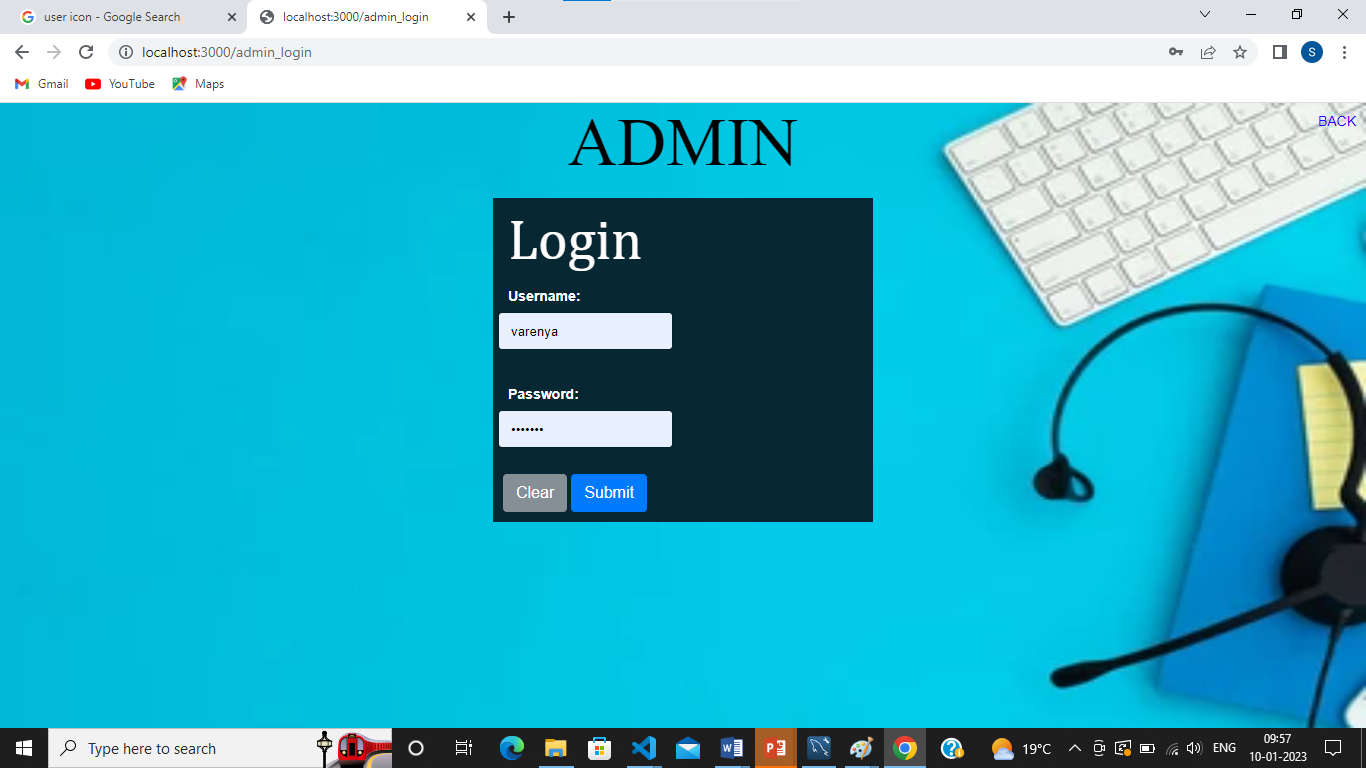
This is the first window when application is executed.

****

**Fig A.1 Home page**

**A.2 Admin side login page**

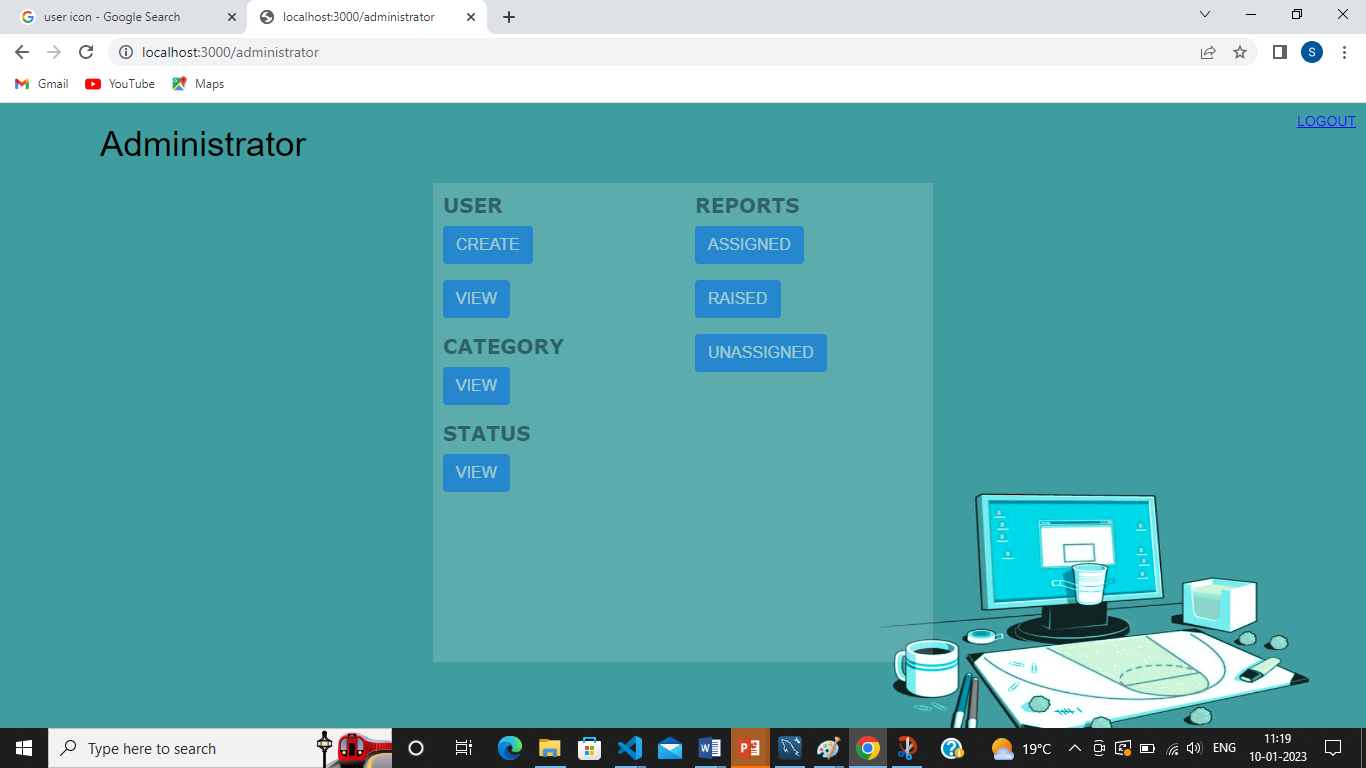
Admin side of the login page.



**Fig B.2 Admin side login page**

**A.3 Admin panel**

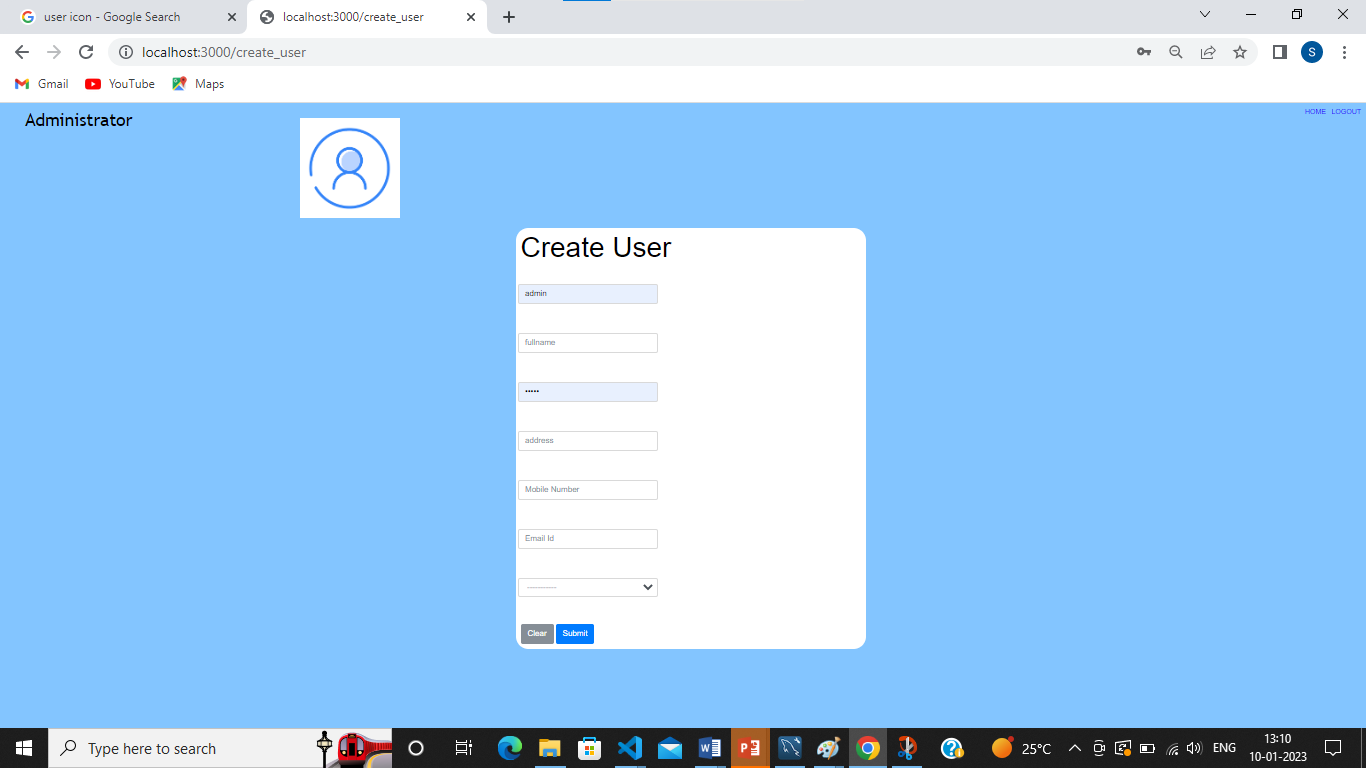
Overall managing like create, view and assigned details of the entries and queries taking place in the system is given.

****

**Fig A.3 Admin panel**

**A.4 User Modelling Page**

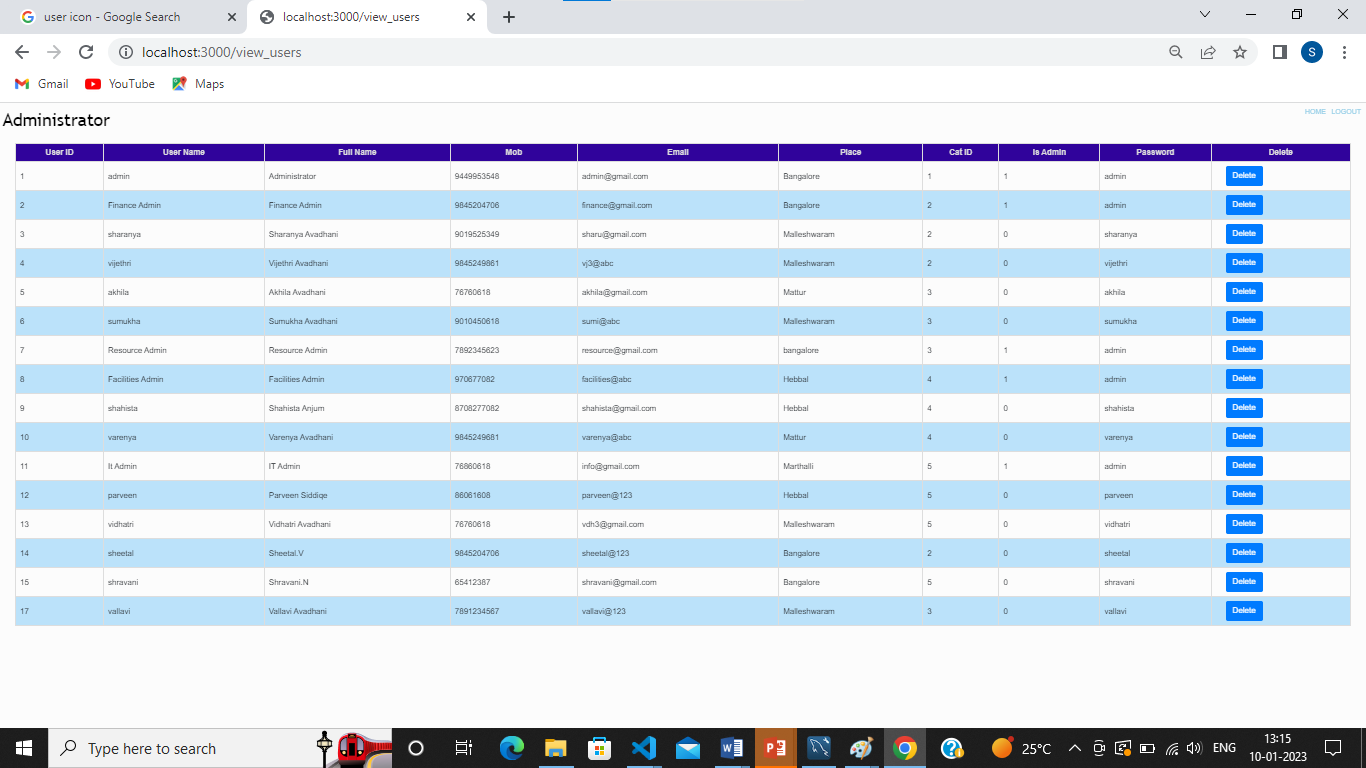
This page is to create users.



**Fig B.2** **User Creation**

**A.5 Created users**

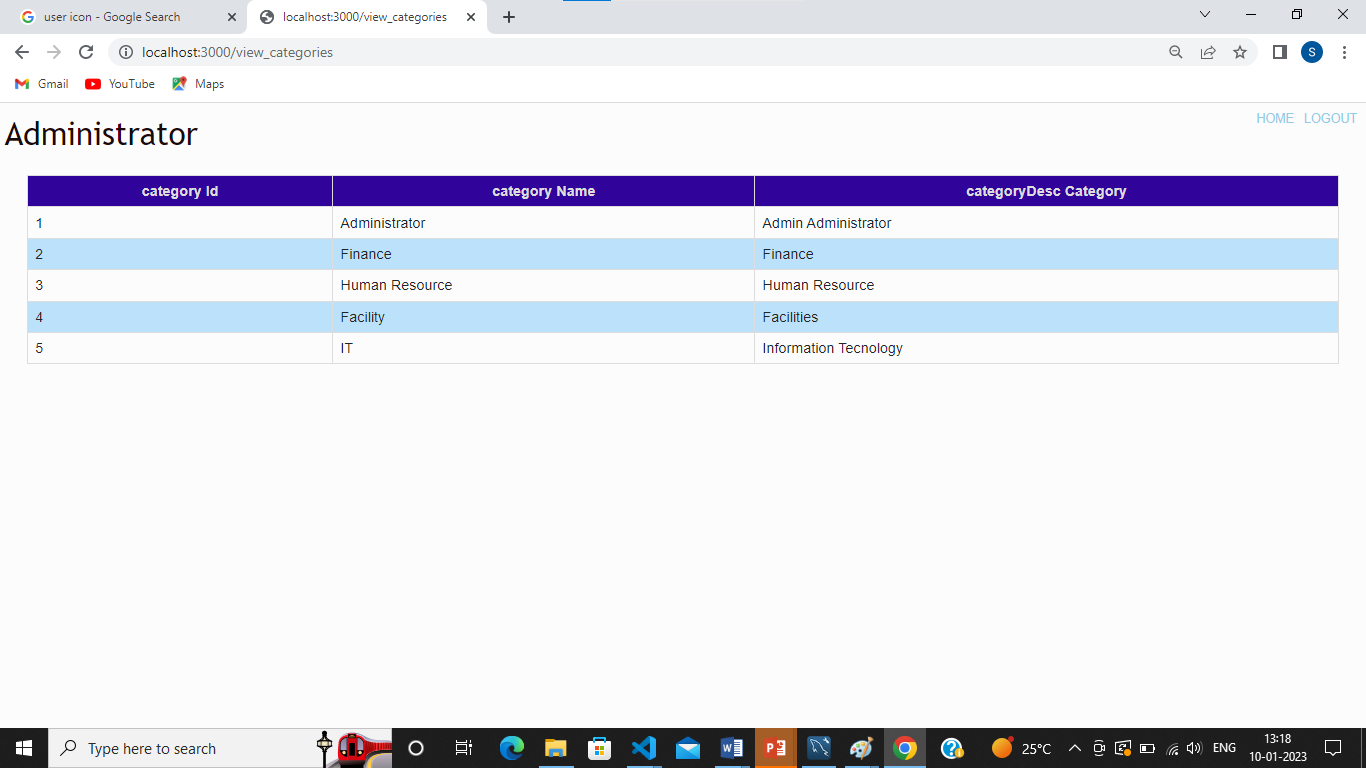
When users are created, details are asked to store in the database.



**Fig A.5 Created users**

**A.6 Task Category**

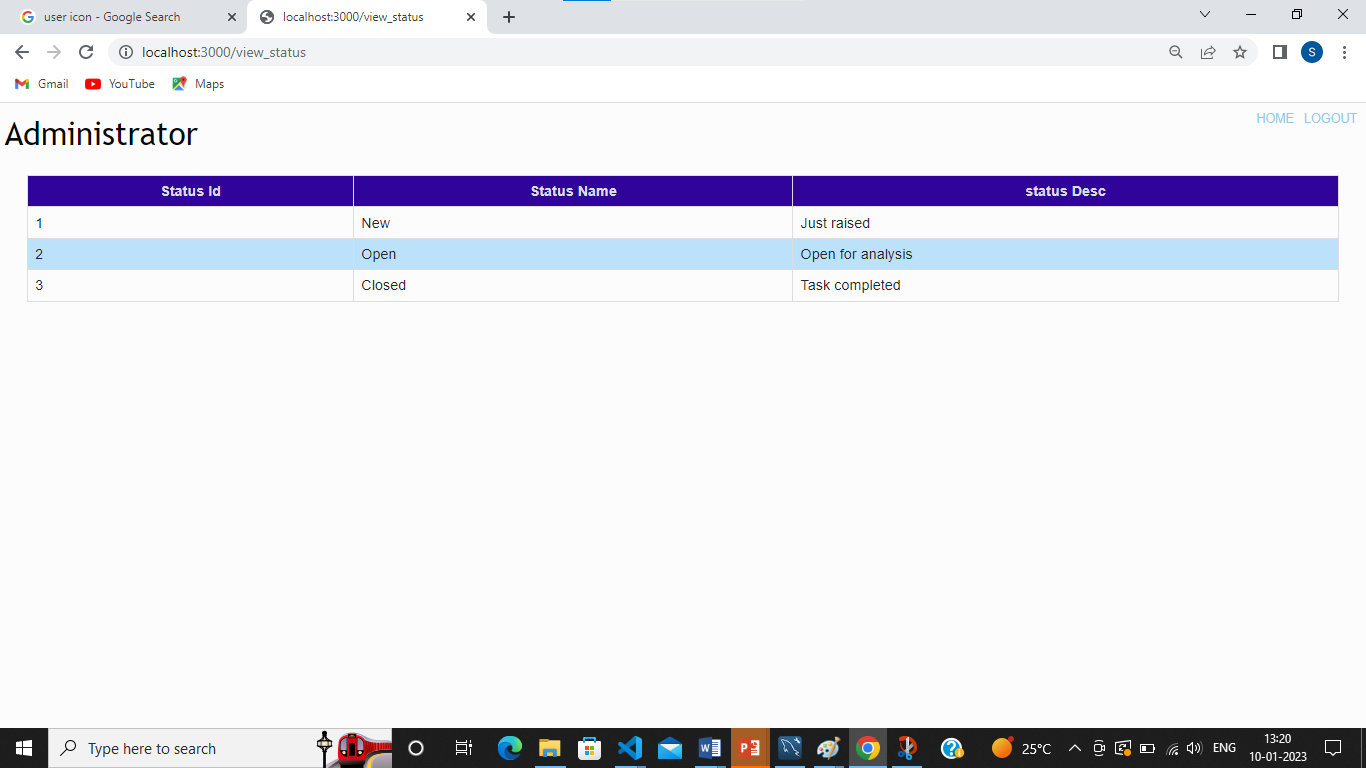
These are the categories under which different tasks are created.

****

**Fig A.6 Categories**

**A.7 Task Status**

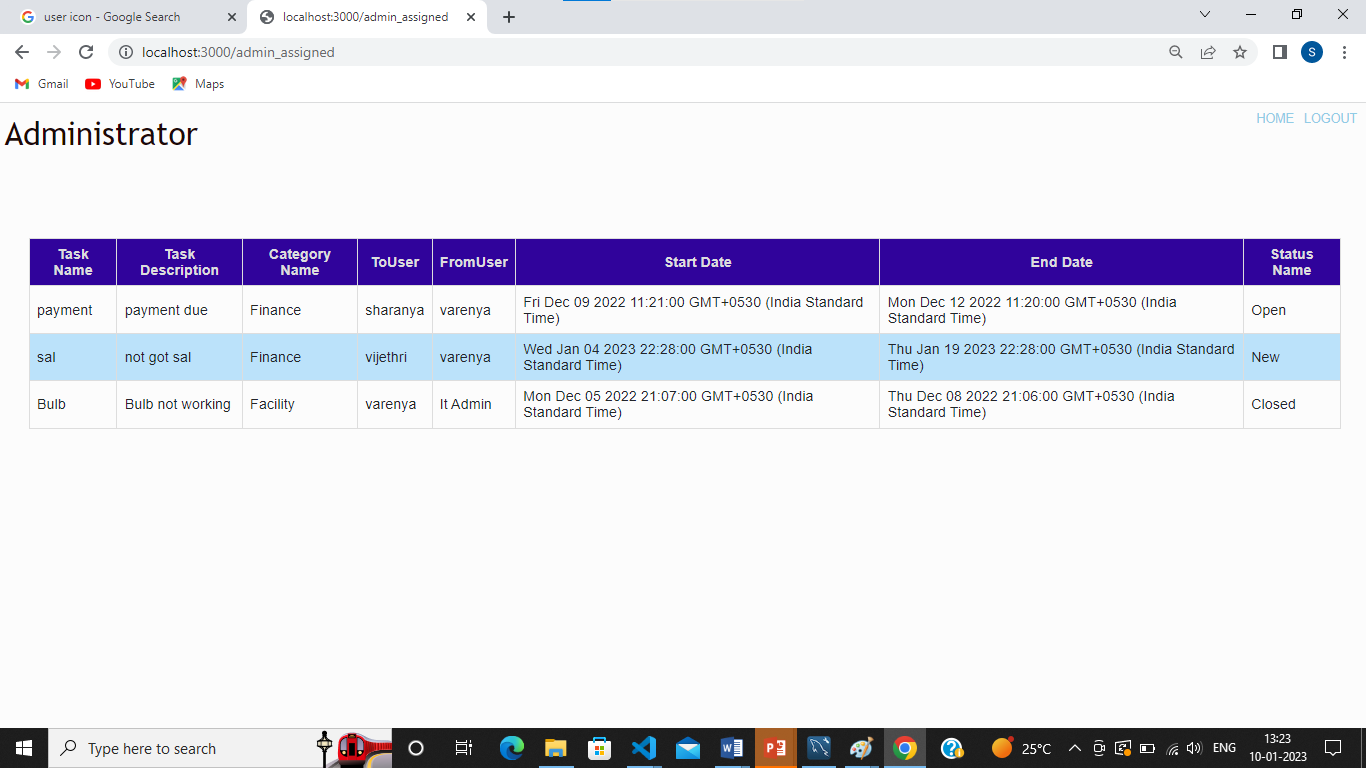
These are the different status a task can exist in.



**Fig A.7 Status**

**A.8 Task Assigned**

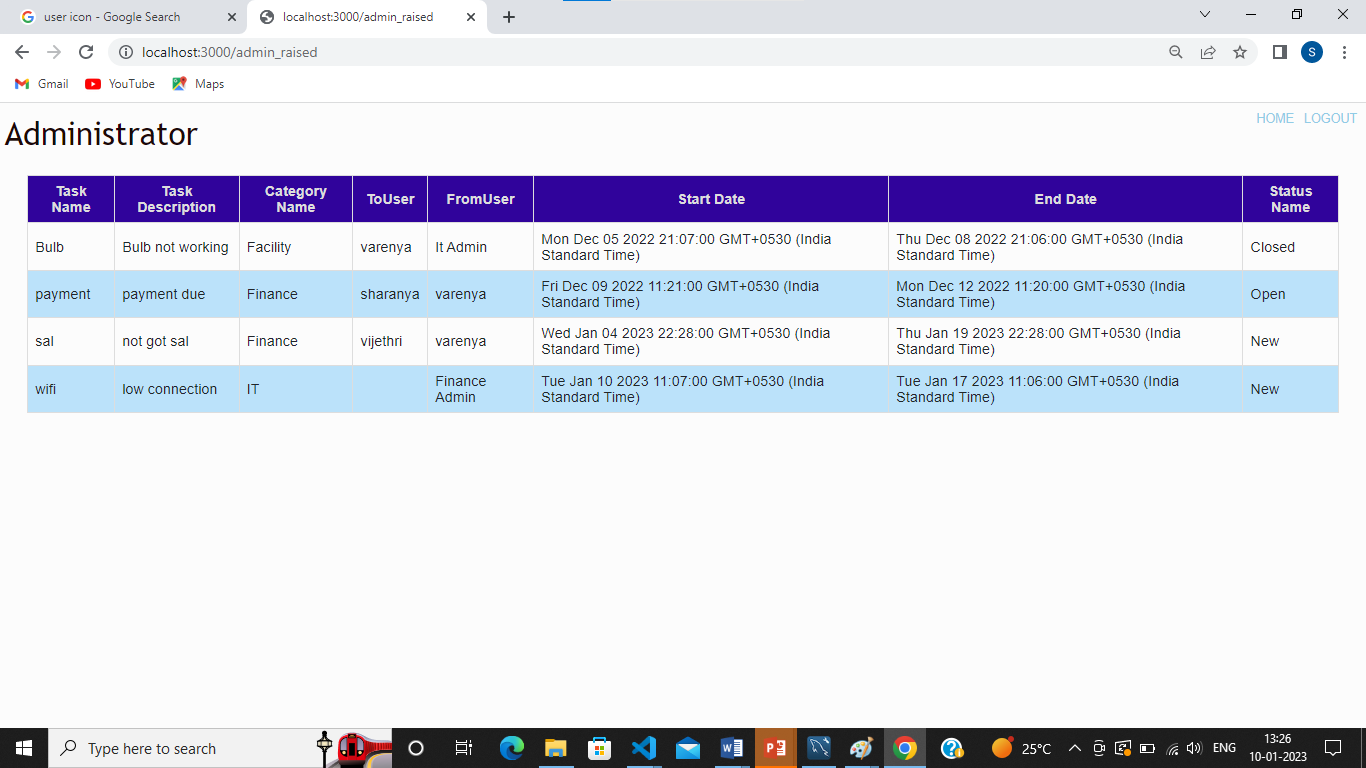
These are the different tasks that are currently assigned.



**Fig A.8 Assigned**

**A.9 Task Raised**

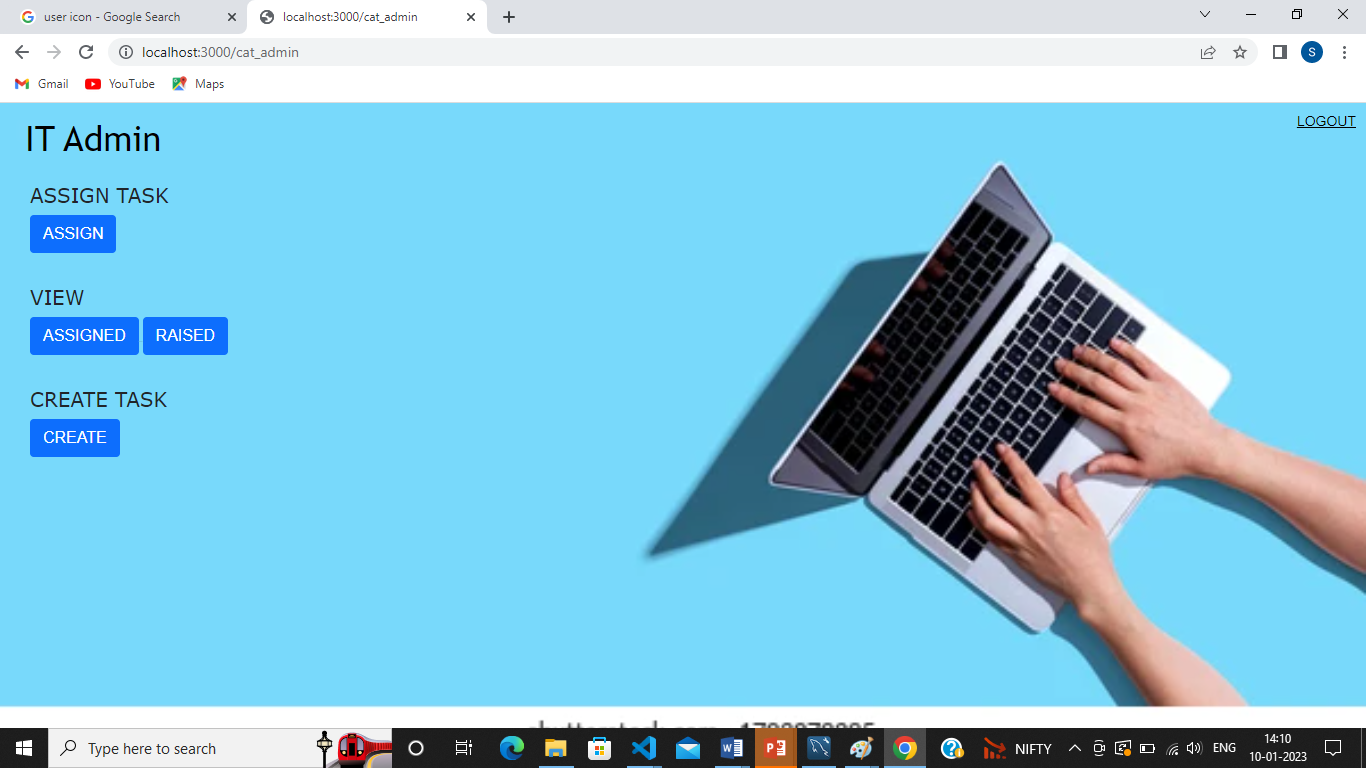
These are the different tasks that are currently raised but not assigned.



**Fig A.9 Raised**

**A.10 Category Admin Panel**

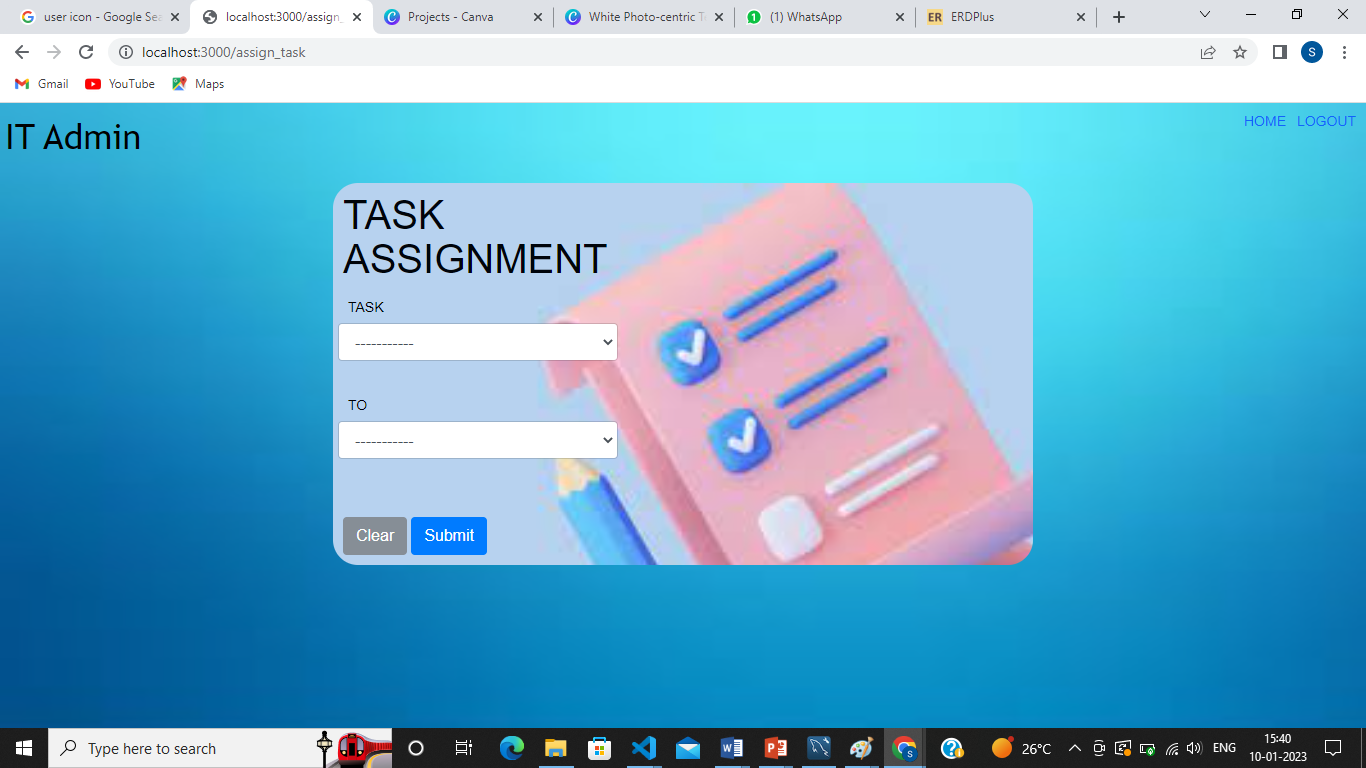
The panels help the admins to deal with their respective tasks of their department.



**Fig A.10 Category Admin Panel**

**A.11 Task Assignment**

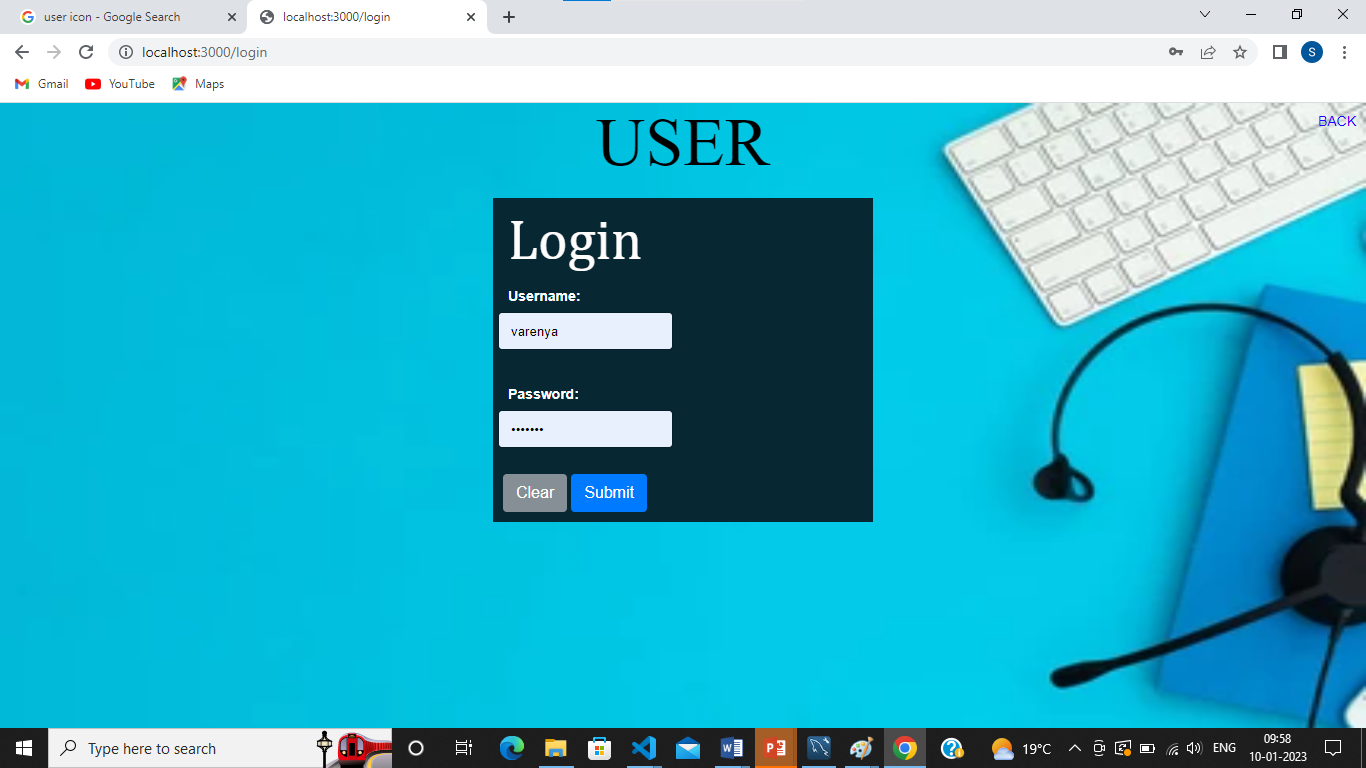
The raised tasks are assigned to the employee’s in this panel.

****

**Fig A.11 Task Assignment**

**A.12 User Side Login Page**

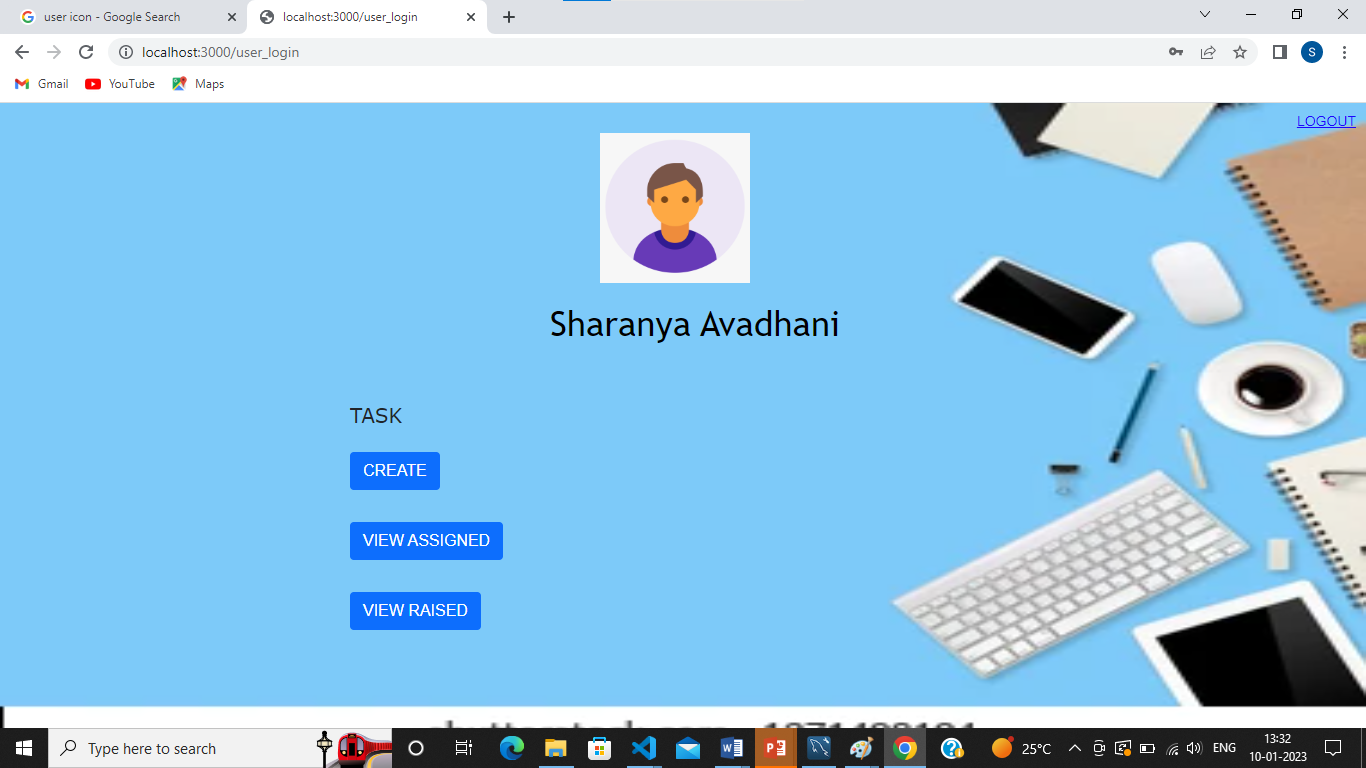
User side of the login page.

****

**Fig A.12 User side login page**

**A.13 User Panel**

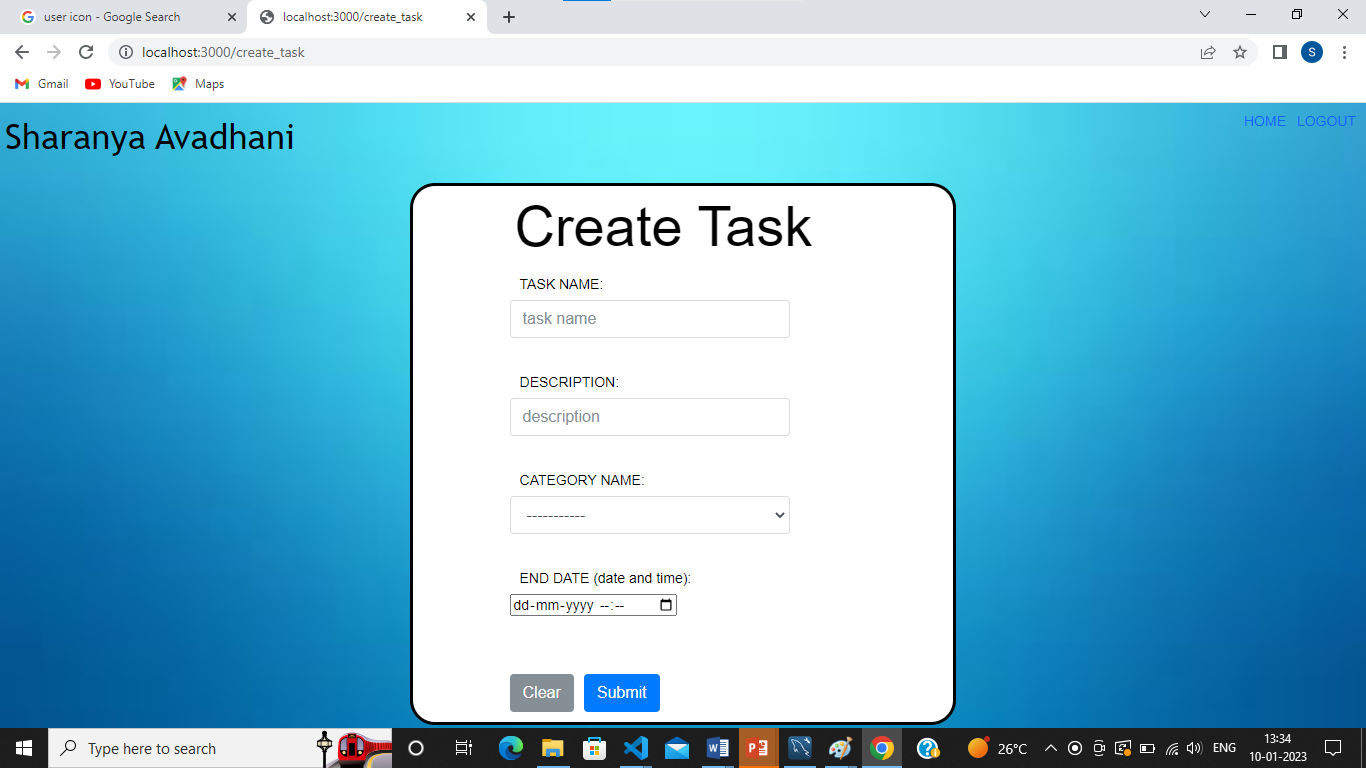
Overall functions like create, assigned and raised details of the entries and queries taking place in the system is given.

****

**Fig A.13 User Panel**

**A.14 Task Panel**

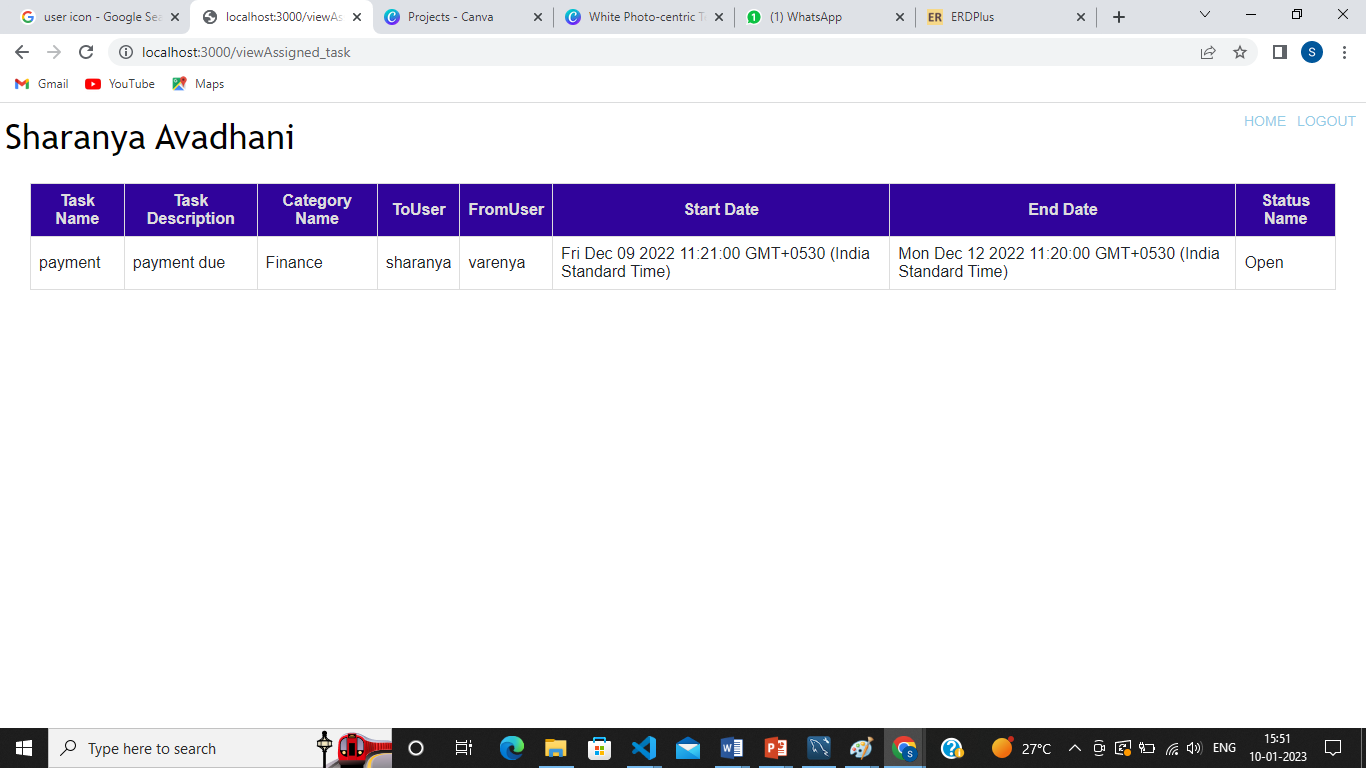
Creation of task.

****

**Fig A.14 Creation of task**

**A.15 Assigned Task**

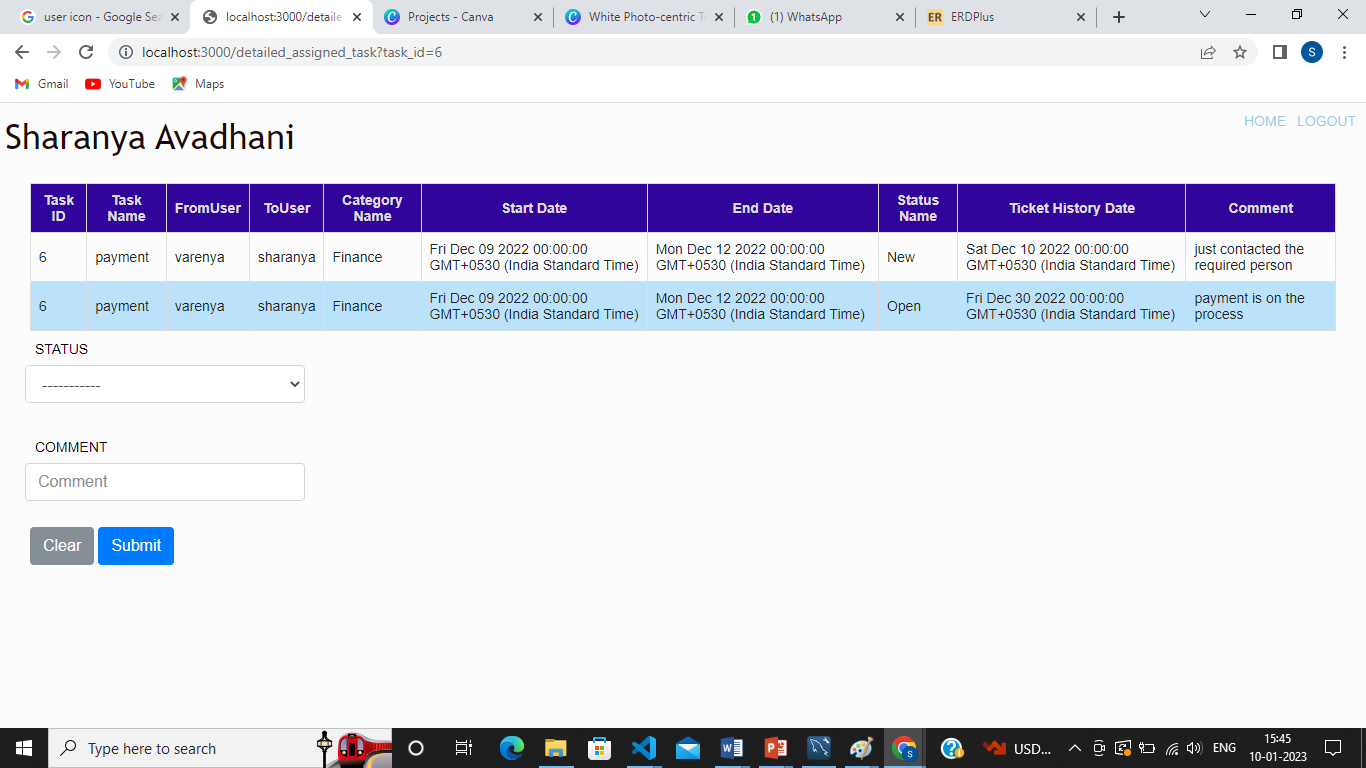
This panel tells the user what tasks are assigned.

****

**Fig A.15 assigned task**

**A.16 Assigned Task Details**

This panel makes the user resolve the task assigned to it and give time to time updated about the task.

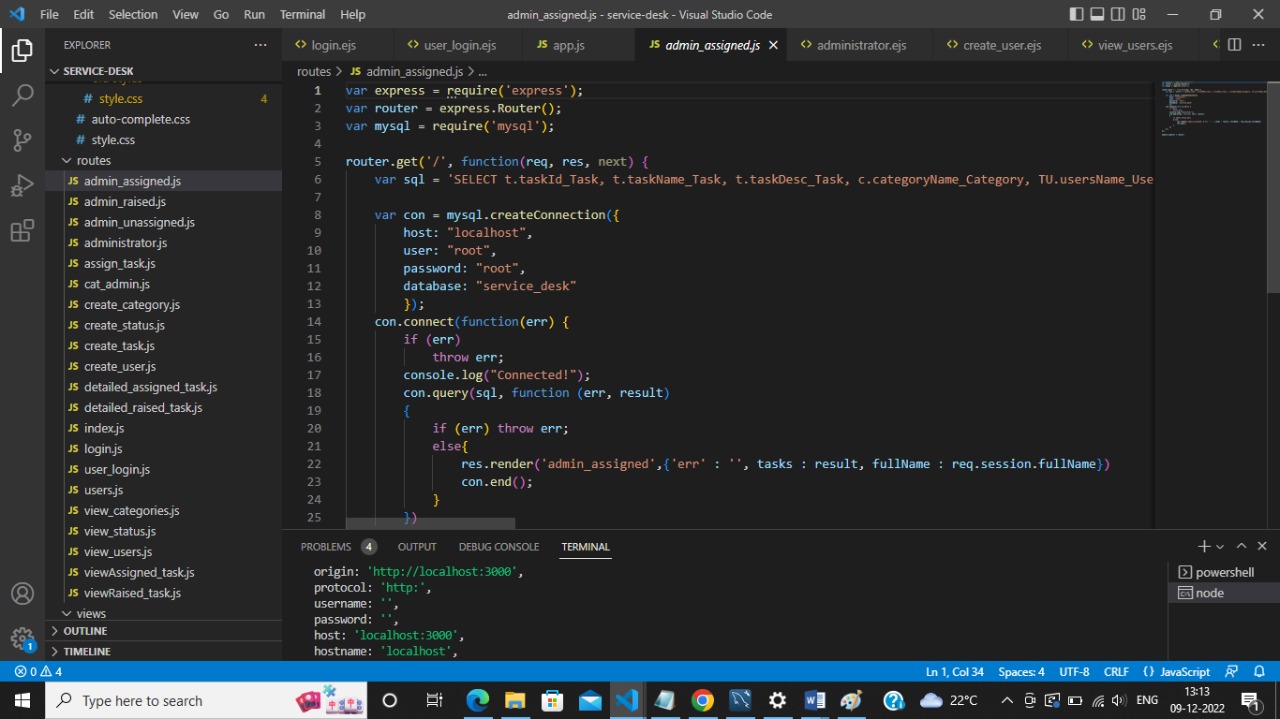


**Fig A.16 Detailed assigned task**

**APPENDIX ‘B’- CODE SNIPPETS**

**B.1 DATABASE CONNECTION**

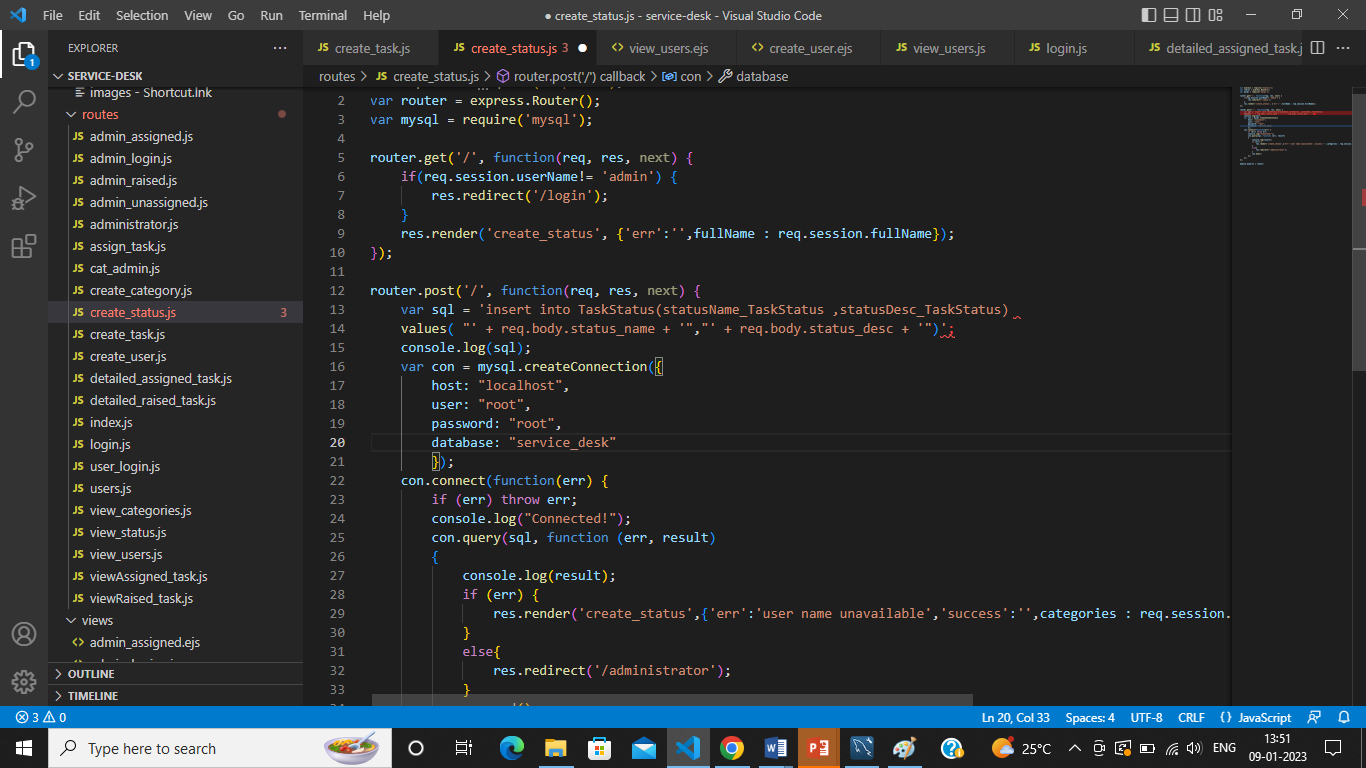
This is for database connection.

S

**Fig B.1 Database Connection**

**B.2 INSERT QUERY**

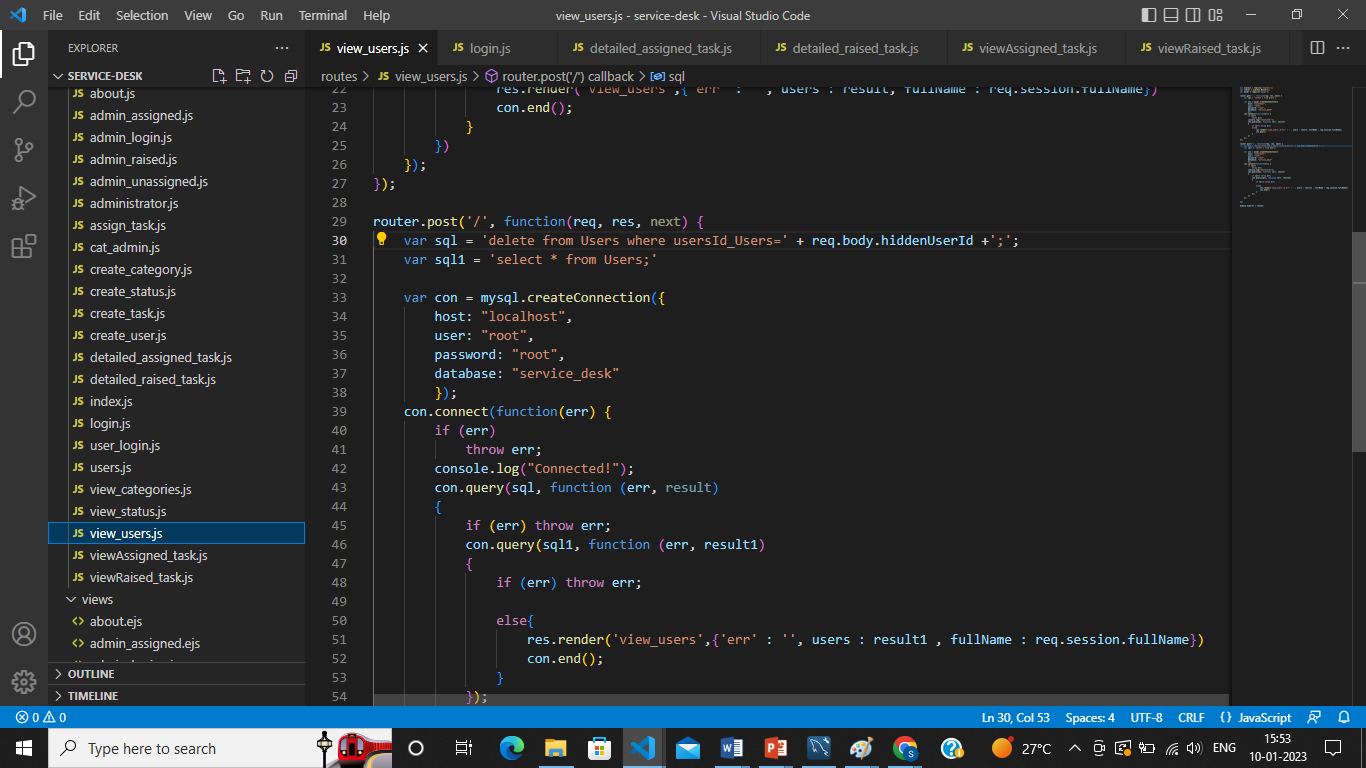
This query is used to insert a new.



**Fig B.2 Insert Query**

**B.3 DELETE QUERY**

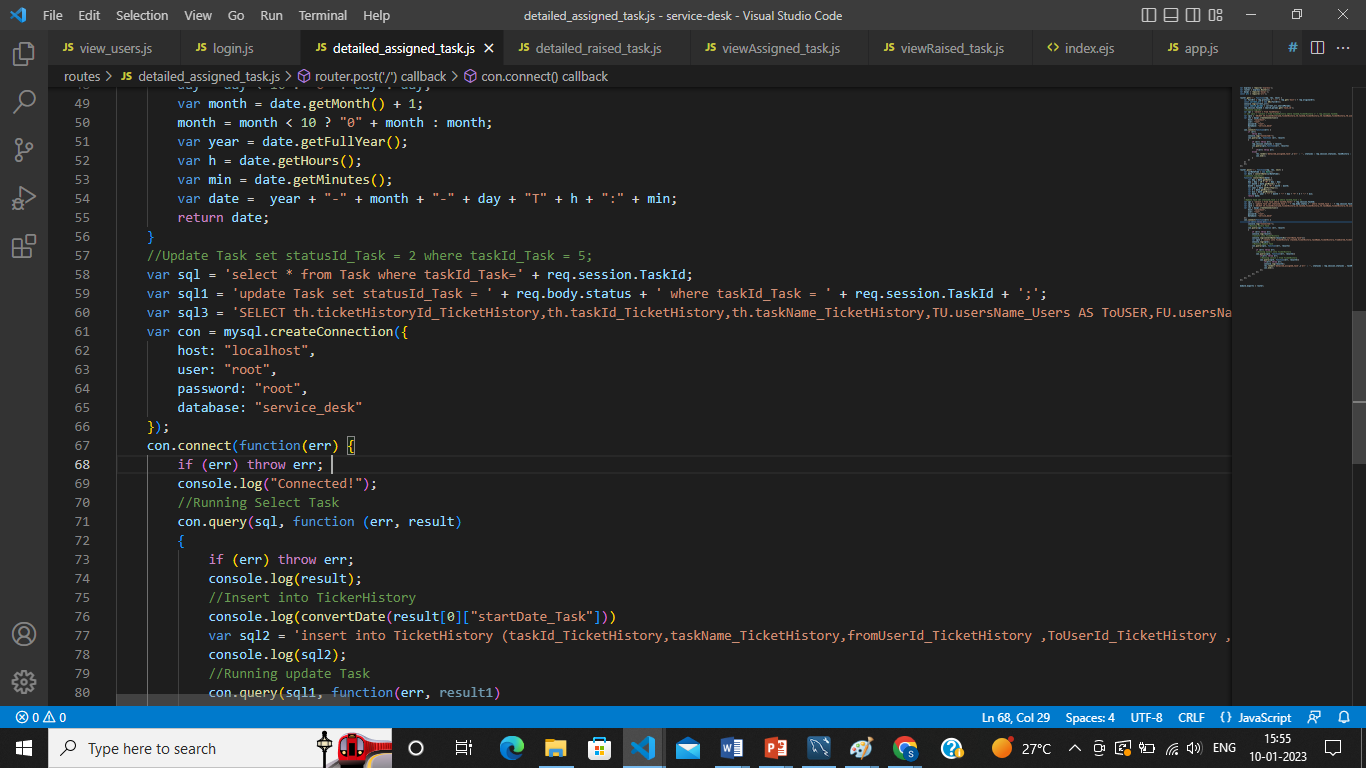
This query is to delete the users.

****

**Fig B.3 Delete Query**

**B.4 UPDATE QUERY**

This query is to update the task info.



**Fig B.4 Update Query**