**CHAPTER 1**

**INTRODUCTION**

* 1. **OBJECTIVES**:
* The main objective of the project is to design and develop a user friendly-system.
* Easy to use and an efficient computerized system.
* To develop an accurate and flexible system, it will eliminate data redundancy.
* To study the functioning of Road Quality Monitoring System.
* To make a software fast in processing.
* To make software with good user interface so that it should be used for a long time without bugs and maintenance.
* To provide centralized Road Quality database.
* Computerization can be helpful as a means of saving time and money.
* To provide better Graphical User Interface (GUI).
* Less chances of information leakage.
* Provides Security to the data by using login and password method.
* To provide immediate storage and retrieval of data and information.
* Improving arrangements for better Quality Roads.
* Reducing paperwork.
  1. **LIMITATIONS**:
* Time consumption in data entry as the records are to be manually maintained a lot of time.
* Lot of paper work is involved as the records are maintained in the files and registers.
* Less Reliable use of papers for storing valuable data information is not at all reliable.
* Google maps are not connected , Once its connected we can get the location directly.

**CHAPTER 2**

**STUDY OF EXISTING SYSTEM**

**2.1 CASE STUDY**

* Smartphones and tablets have become an important aspect of modern society. In recent years,the smartphone market has seen substantial increases in the number of shipments all over the world . Furthermore, mobile internet use has also seen steady growth in the last few years and has recently surpassed desktop internet usage for the first time worldwide .
* The popularity of mobile devices, coupled with their increasing computing power, means that smartphones are performing an increasingly wide range of functions. Applications on mobile platforms such as Google Android and Apple iOS, can also take advantage of devices’ internal sensors, which opens up new ways to utilize these smart devices. Sensors can be leveraged for different tasks, such as detecting the orientation of the device or for more complex problems, such as human activity detection .
* Road infrastructure is another important part of modern society. A large number of people frequently use roads, whether it be in a personal car, on a bike, on foot, or in public transport. Many public services and businesses also depend on the road network in one way or another. As the motorization rate continues to grow , a well-maintained and safe road network is beneficial for everybody involved .
* As this is the mini project our main aim to have better data base to know the quality of roads. This is easy-to-use, integrated database application is geared towards reducing time spent on administrative tasks. The system is intended to accept Accelerometer data and generate report accurately. Any user can access the system at any point in time and enter the details where potholes are present provided internet facility is available
* The system is also intended to provide better services to users, provide meaningful, consistent, and timely data and information and finally promotes efficiency by converting paper processes to electronic form.
* The system was developed using technologies such as, HTML, CSS ,JS and MySQL. PYTHON- FLASK, HTML and CSS are used to build the user interface and database was built using MySQL.
* The system is free of errors and very efficient and less time consuming due to the care taken to develop it. All the phases of software development cycle are employed and it is worthwhile to state that the system is very robust. Provision is made for future development in the system.

**2.2 MOTIVATION AND PROBLEM STATEMENTS**

* Roads require frequent oversight to find and fix anomalies as soon as possible to keep them from deteriorating further.
* In addition to normal wear and tear, factors such as poor construction quality, heavy traffic, poor drainage, weak subgrade and large variations in temperature can 1 contribute to the creation of potholes, cracks and other anomalies, which significantly lower the overall road quality and ride comfort of asphalt pavement roads .
* Poor road quality also causes mechanical strain for vehicles, increasing the need for repairs.
* Furthermore, it can distract drivers, as they might try to dodge road anomalies, which can lead to dangerous situations in traffic. However, building new roads and keeping existing roads in satisfactory shape can be very expensive .
* Inspecting roads manually or by using special dedicated hardware can also be time-consuming. Therefore, developing a more accessible and efficient road quality detection solution, that can highlight problematic areas, would be beneficial to both the road users and the administrators.

**CHAPTER 3**

**DATABASE DESIGN**

**3.1 REQUIREMENTS SPECIFICATION**

**3.1.1 SOFTWARE REQUIREMENTS**

Frontend- HTML, CSS, Java Script, Bootstrap

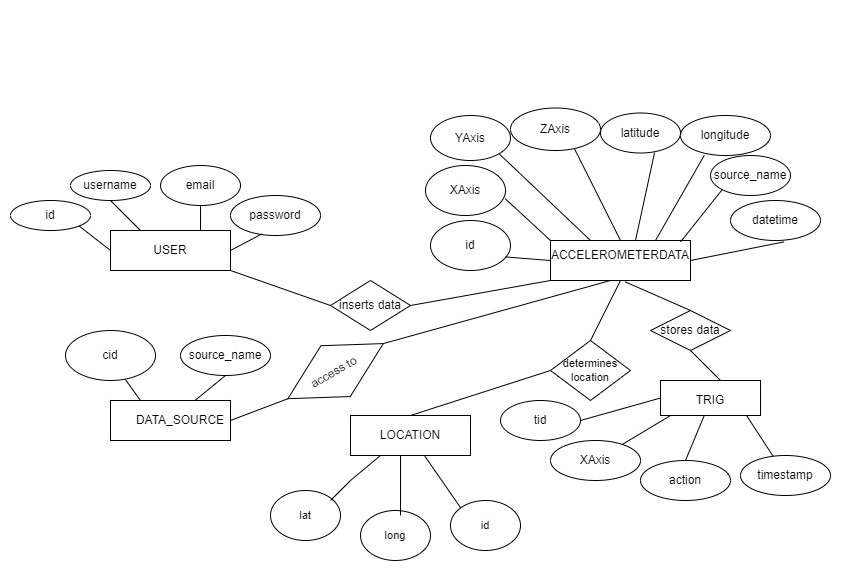
Backend-Python flask (Latest version) , SQLAlchemy

* Operating System: Windows 11
* Google Chrome/Internet Explorer
* XAMPP (Latest Version)
* Python main editor (user interface): PyCharm Community
* workspace editor: Sublime text 3

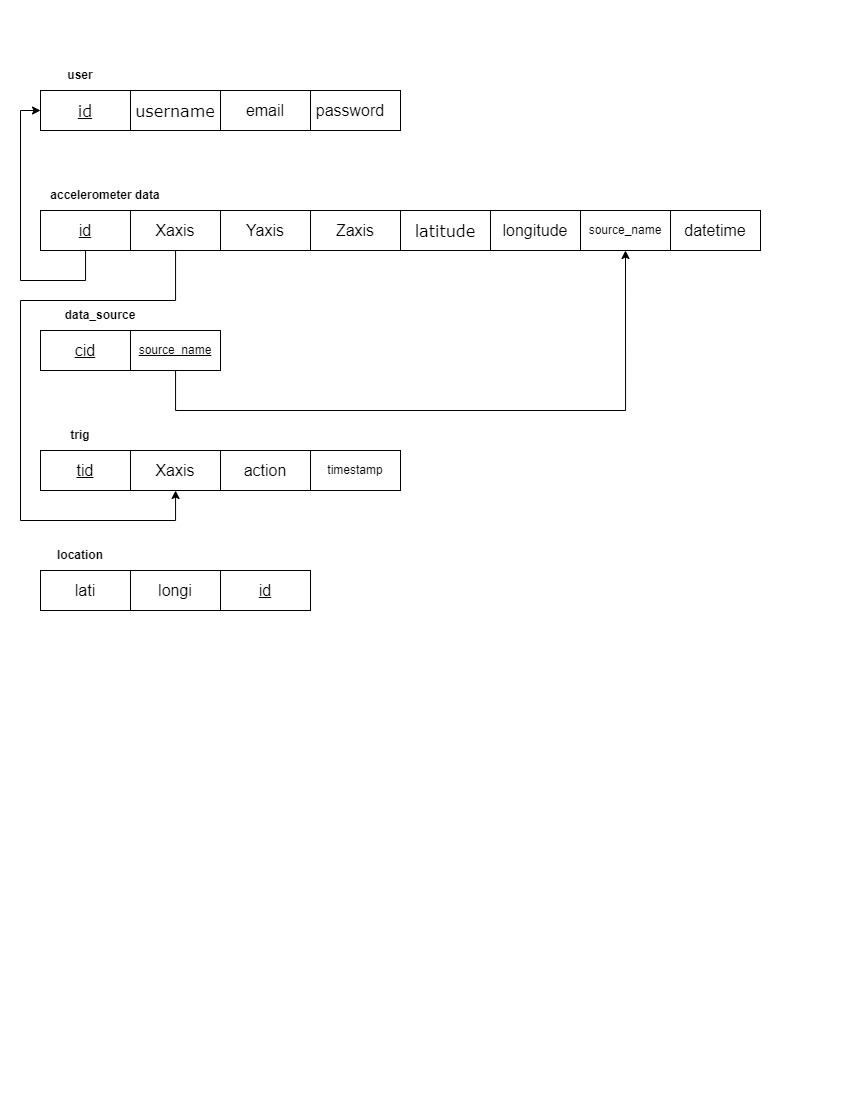
**3.1.2 HARDWARE REQUIREMENTS:**

* Computer with a 1.1 GHz or faster processor
* Minimum 2GB of RAM or more
* 2.5 GB of available hard-disk space or SSD
* 5400 RPM hard drive
* 1366 × 768 or higher-resolution display
  1. **CONCEPTUAL DESIGN:**

**3.2.1 E-R DIAGRAM**

****

* + 1. **SCHEMA DIAGRAM**



**3.3 IMPLEMENTATION:**

An "implementation" of Python should be taken to mean a program or environment which provides support for the execution of programs written in the Python language, as represented by the [CPython r](https://wiki.python.org/moin/CPython)eference implementation.

There have been and are several distinct software packages providing of what we all recognize as Python, although some of those are more like distributions or variants of some existing implementation than a completely new implementation of the language.

**Back end (My sql) Database:**

A Database Management System (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:

* A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.
* The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.
* Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory).A database query language and report writer to allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.
* Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called sub schemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and student data.
* If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.

A transaction mechanism, that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control), and faults (fault tolerance).

* + It also maintains the integrity of the data in the database.
  + The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database. See ACID properties for more information (Redundancy avoidance).

When a DBMS is used, information systems can be changed much more

easily as the organization's information requirements change. to the

Organizations may use one kind of DBMS for daily transaction processing and

then move the detail onto another computer that uses another DBMS better

suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

**SQL:**

Structured Query Language (SQL) is the language used to manipulate relational databases.

SQL is tied very closely with the relational model.

* In the relational model, data is stored in structures called relations or tables*.*

SQL statements are issued for the purpose of:

* Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes) .

## 3.3.3 Stored Procedure

Routine name: proc

Type: procedure

Definition: Select \* from register;

## 3.3.4 Triggers:

## It is the special kind of stored procedure that automatically executes when an event occurs in the database.

Triggers used :

1: Trigger name: on insert

Table: register

Time: after

Event: insert

INSERT INTO trig VALUES(null,NEW.rid,'Latitude INSERTED',NOW())

2: Trigger name: on delete

Table: register

Time: after

Event: delete

Definition: INSERT INTO trig VALUES(null,OLD.rid,'Latitude DELETED',NOW())

3: Trigger name: on update

Table: register

Time: after

Event: update

Definition: INSERT INTO trig VALUES(null,NEW.rid,'Latitude UPDATED',NOW())

**3.3.5 TABLE DESCRIPTION:**

A table in a database is a collection of related data stored in a structured format, with rows and columns. Each row represents a single record, and each column represents a specific field within that record. Tables are used to store data in a relational database management system (DBMS) and are the foundation of relational databases. The structure of a table, including the names and data types of its columns, is known as its schema. In a DBMS, tables are used to store data and can be related to other tables through relationships, such as one-to-one, one-to-many, and many-to-many. Additionally, tables can be indexed for faster searching and sorting of data.

A Table 2.1 **AccelerometerData** **Table** is a database table that stores information about Accelerometer and location data. It typically includes fields such as ID, Xaxis, Yaxis , Zaxis ,latitude, longitude, source name.

**Table 2.1: AccelerometerData Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Data Type** | **Constraints** | **Description** |
| Id | Int | Primary key | Unique identification for Data. |
| Xaxis | Float |  | Accelerometer sensor data. |
| Yaxis | Float |  | Accelerometer sensor data. |
| Zaxis | Float |  | Accelerometer sensor data. |
| Latitude | Float |  | Location data. |
| ­­­­­­­Longitude | Float |  | Location data. |
| Source name | Varchar | Foreign key | Source through which data is Entered |

A Table 2.2 **Datasource Table** is a database table that stores information about Source through which Accelerometer is entered . It includes fields such as ID and source name.

**Table 2.2: Source name Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Data Type** | **Constraints** | **Description** |
| cid | Int | Primary key | Unique identification for Data. |
| Source\_name | varchar |  | Accelerometer sensor data. |

A Table 2.3 **Trigger Table** is a database table that stores information about time stamp and values respectively . It includes fields such as tid, Xaxis, action and timestamp

**Table 2.3: Trigger Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Data Type** | **Constraints** | **Description** |
| tid | Int | Primary key | Unique identification for Data. |
| Xaxis | Float |  | Accelerometer sensor data. |
| Action | Varchar |  | Accelerometer sensor data. |
| timestamp | datetime |  | Accelerometer sensor data. |

A Table 2.4 **User** **Table** is a database table that stores the login credentials for the users of a website or application. It typically includes fields such as the id,username, password and other related information like email.

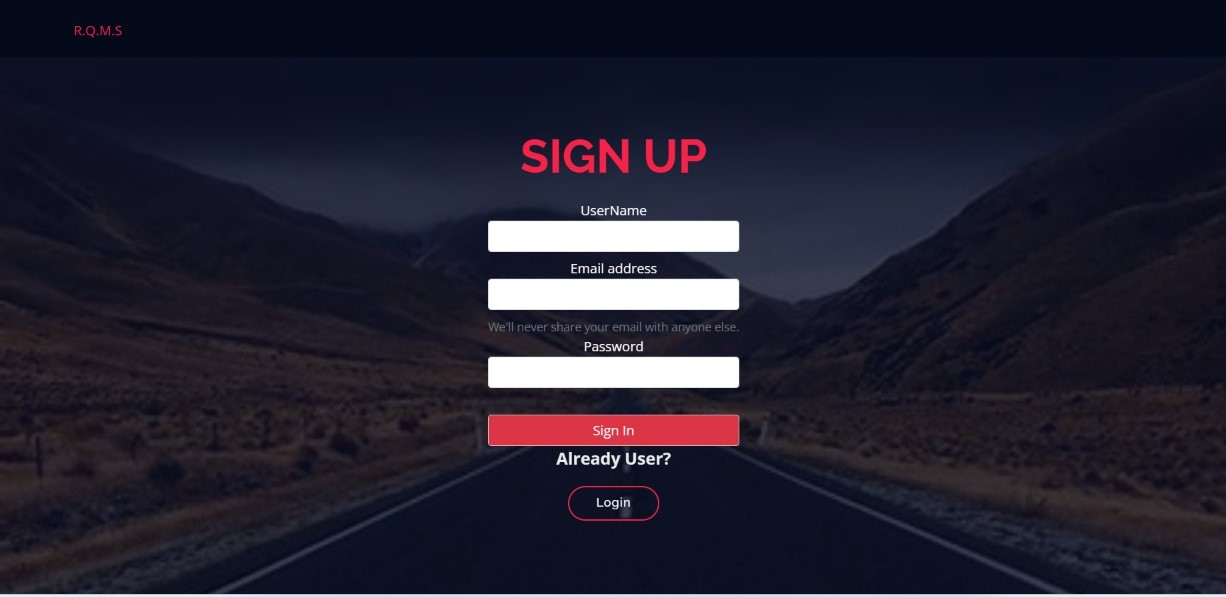
**Table 2.3: User Table**

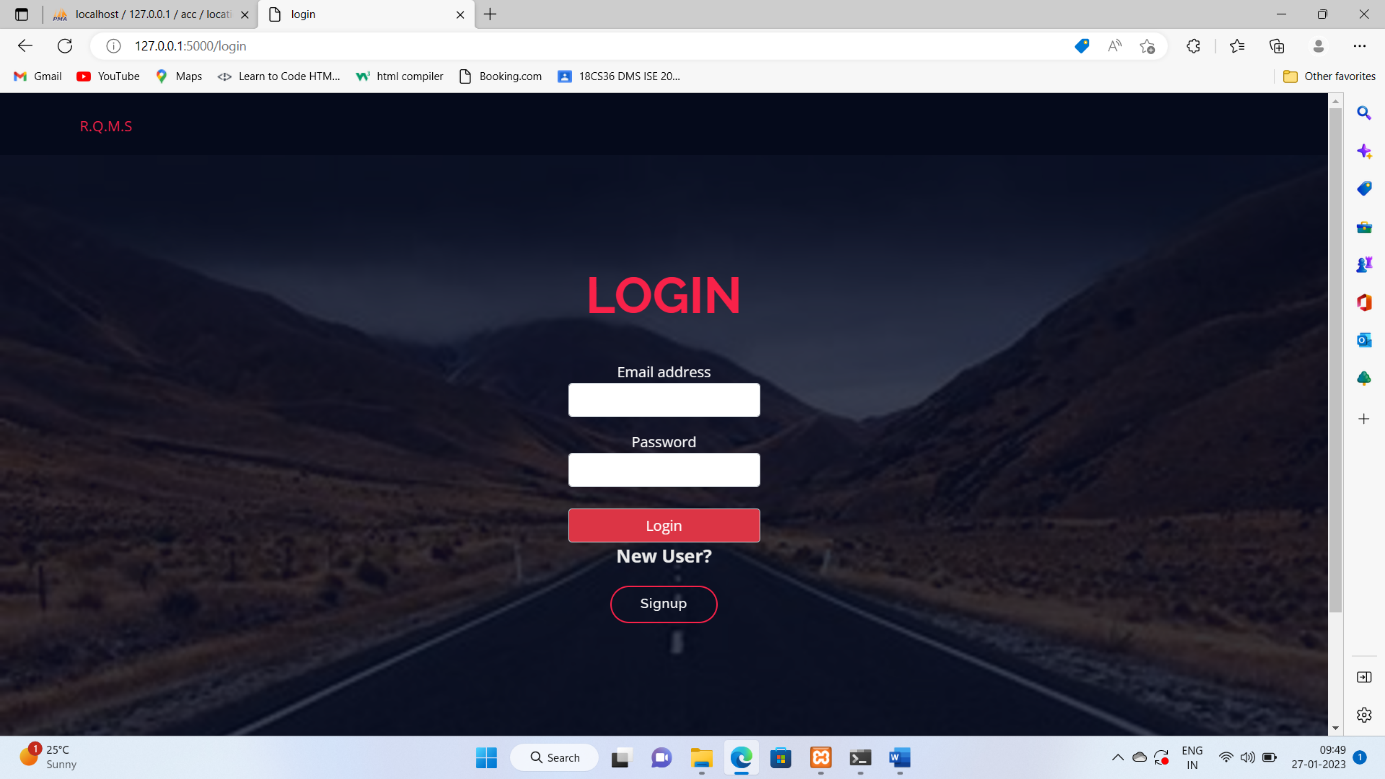
|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Data Type** | **Constraints** | **Description** |
| id | Int | Primary key | Unique identification for Data. |
| Username | varchar |  | Unique identification name. |
| Email | Varchar |  | Electronic mail identification |
| Password | Varchar |  | Account access key phrase.. |

**CHAPTER 4:**

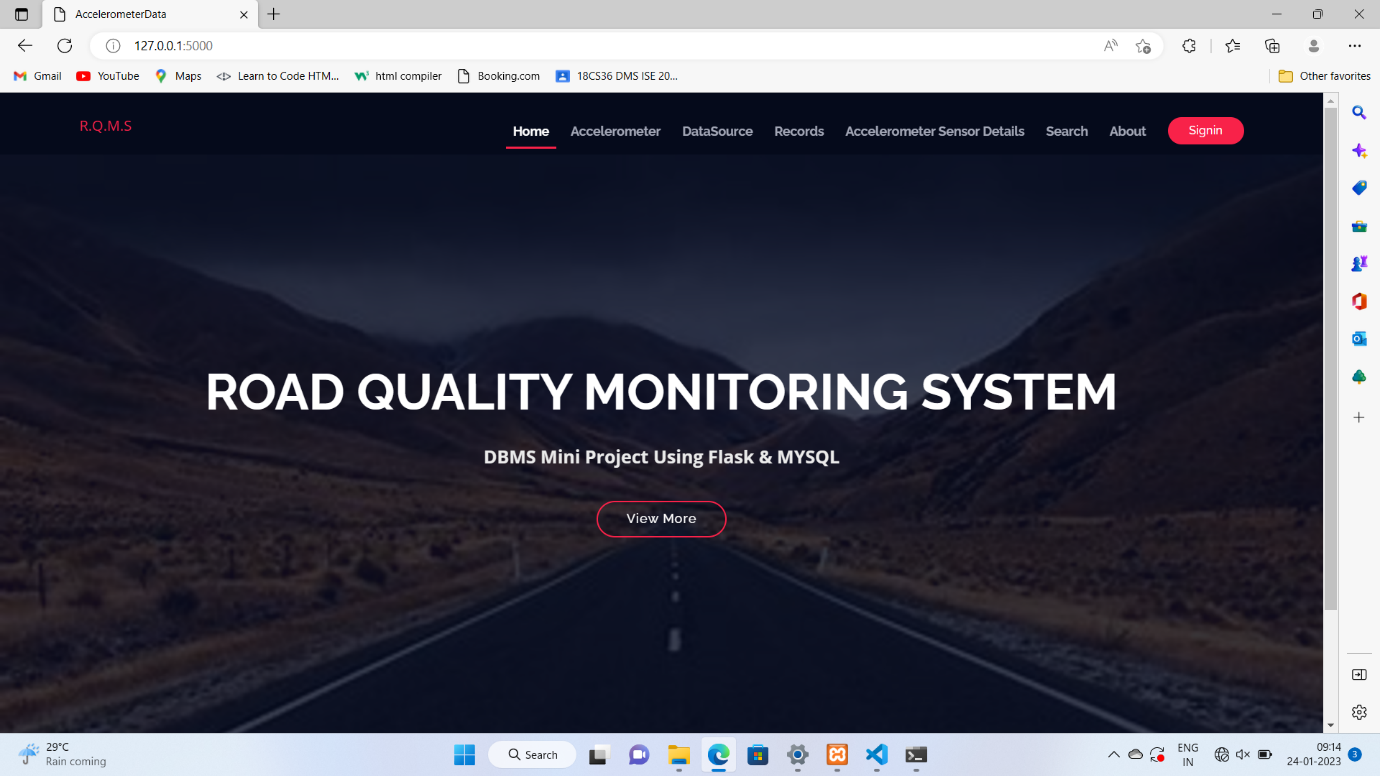
**4.1 USER INTERFACE SNAPSHOTS**

**Login page**

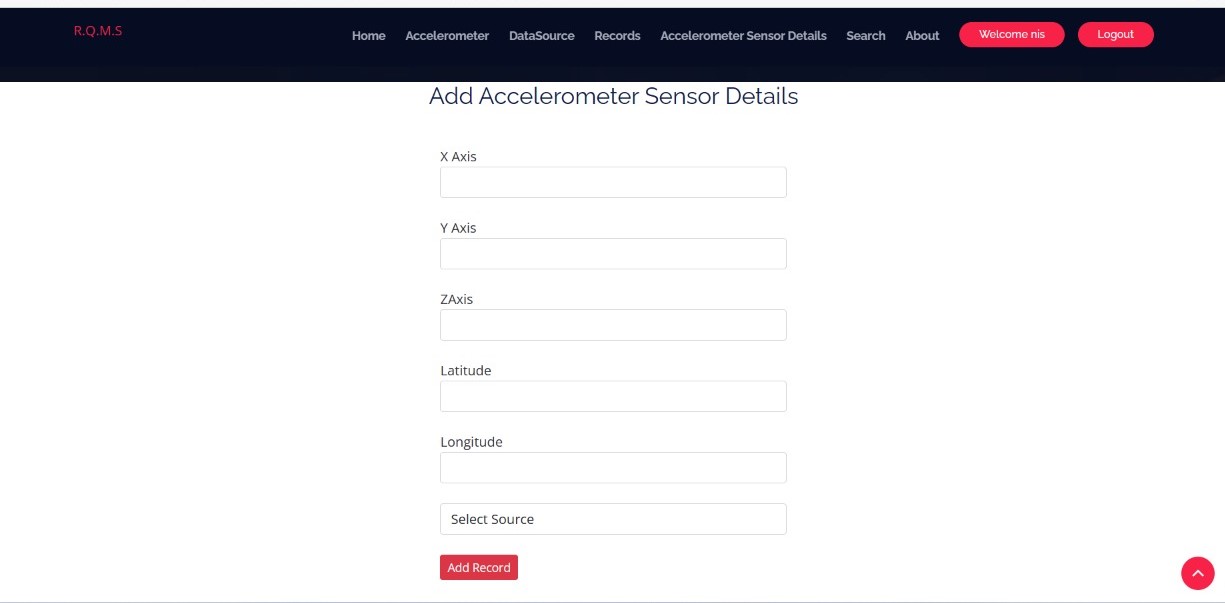
****

****

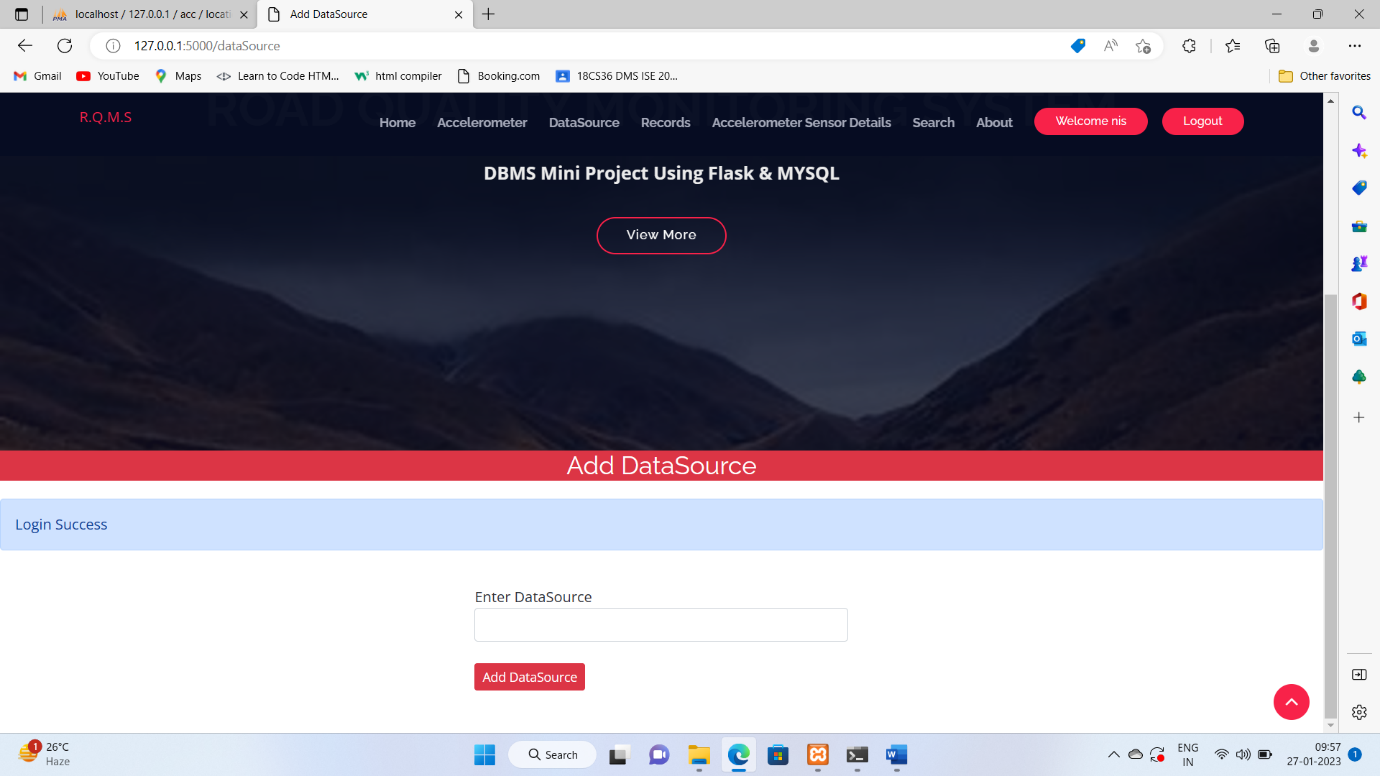
**HOME PAGE**

****

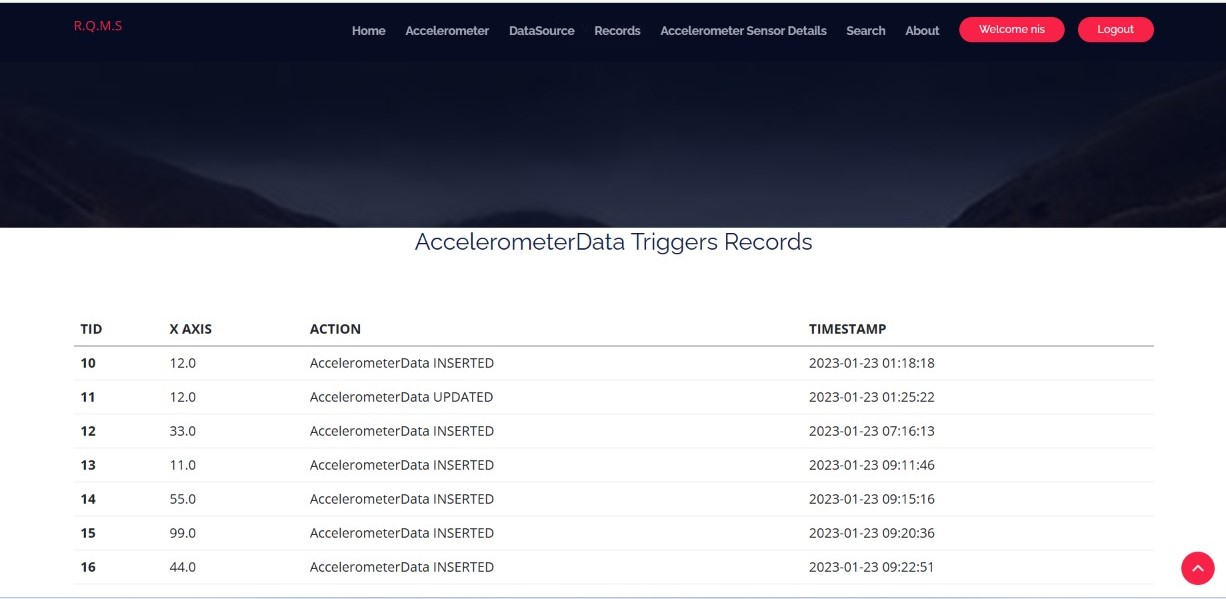
**ADD ACCELEROMETER SENSOR DETAILS**

****

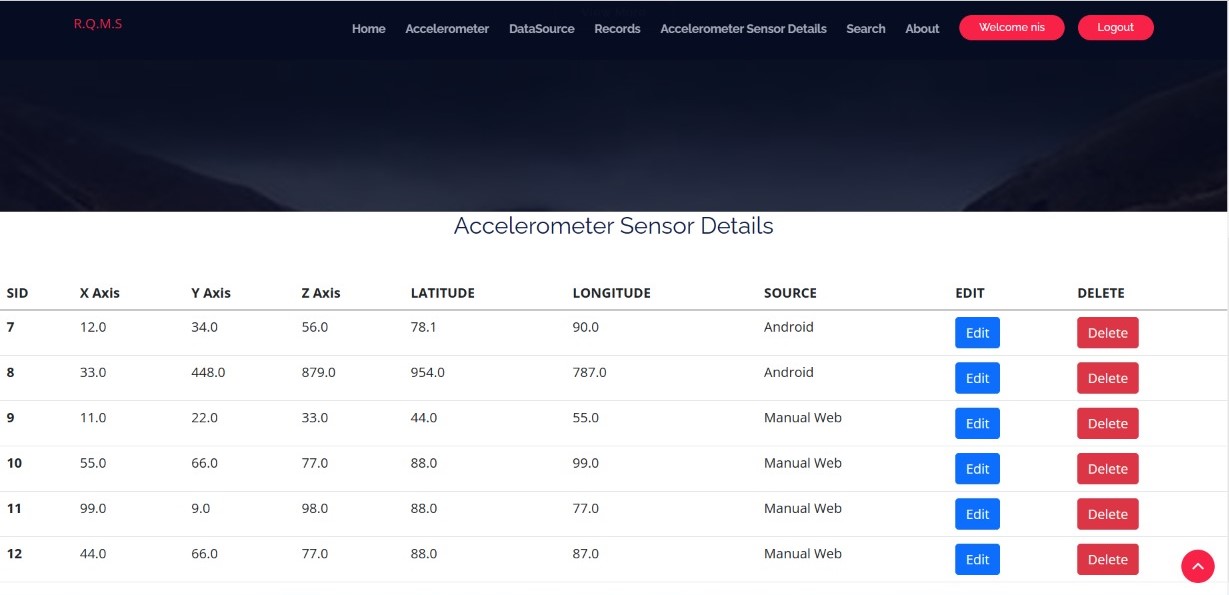
**ADD DATA SOURCE**

****

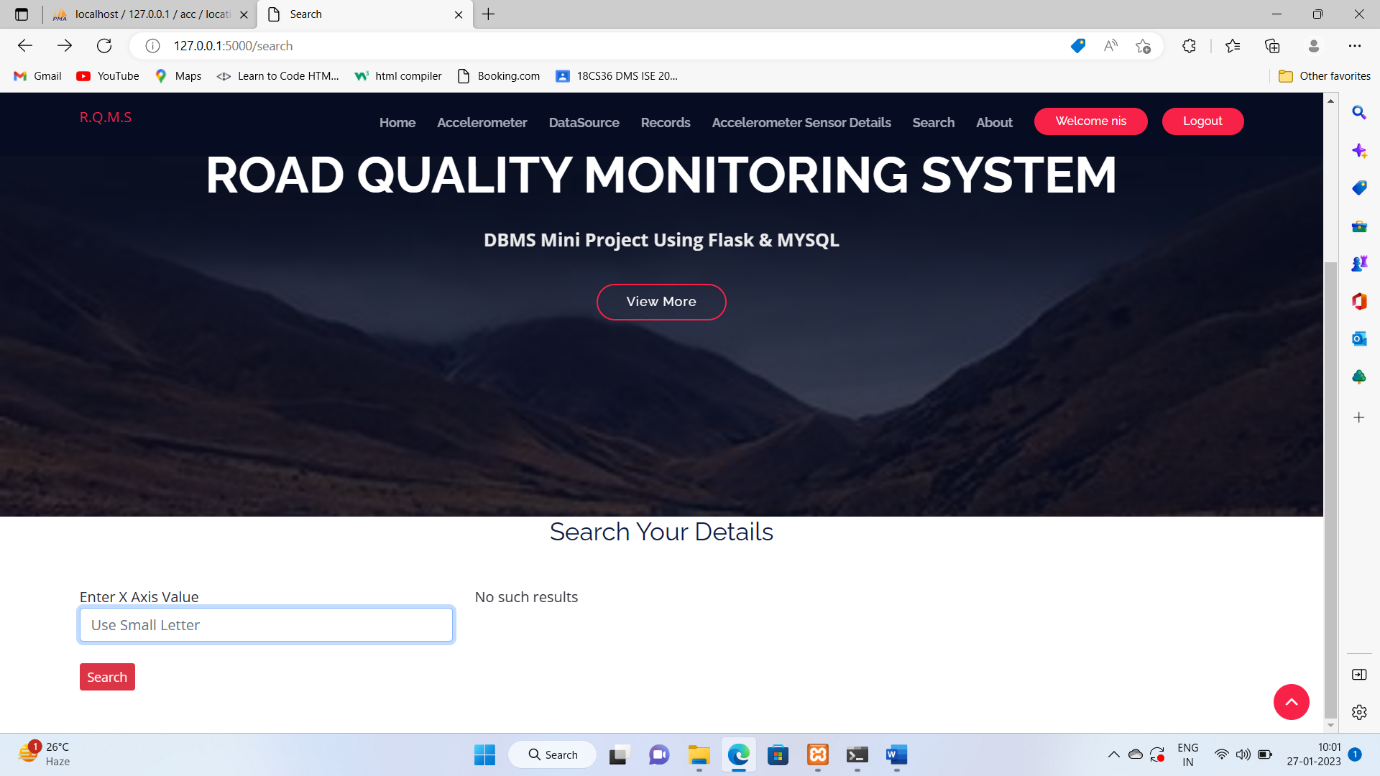
**TRIGGERS RECORDS**

****

**ACCELEROMETER SENSOR DETAIS**

****

**SEARCH VALUES**

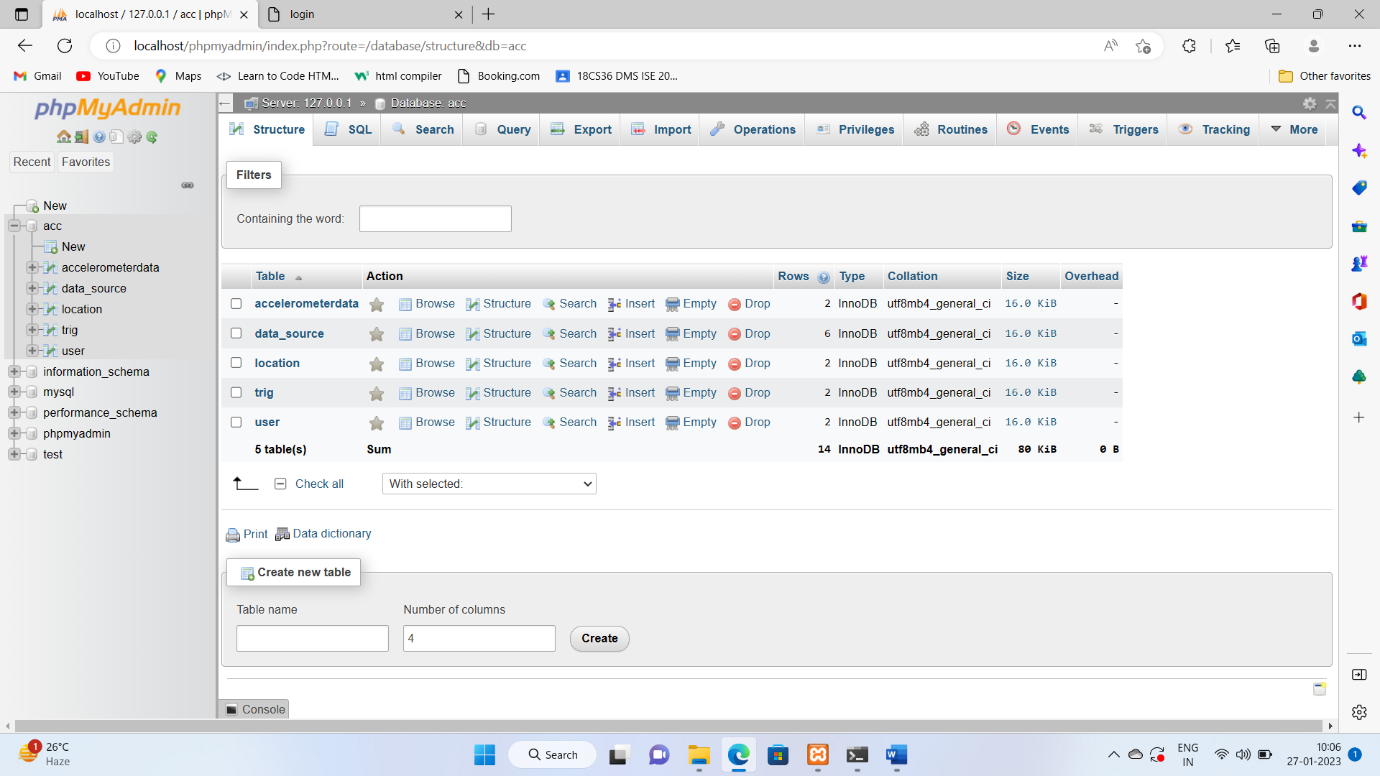
****

**LOGOUT PAGE**

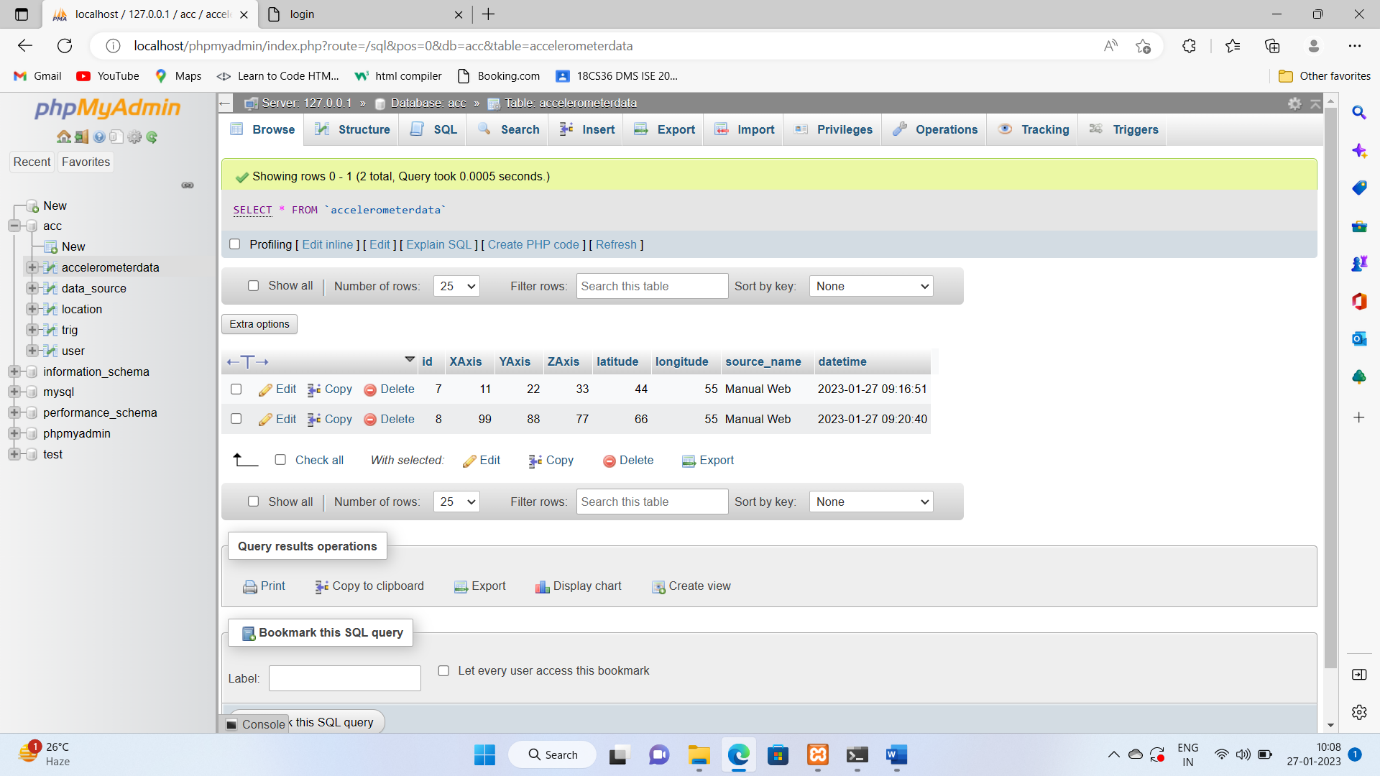
****

**4.2 DATA BASE SCREENSHOTS**

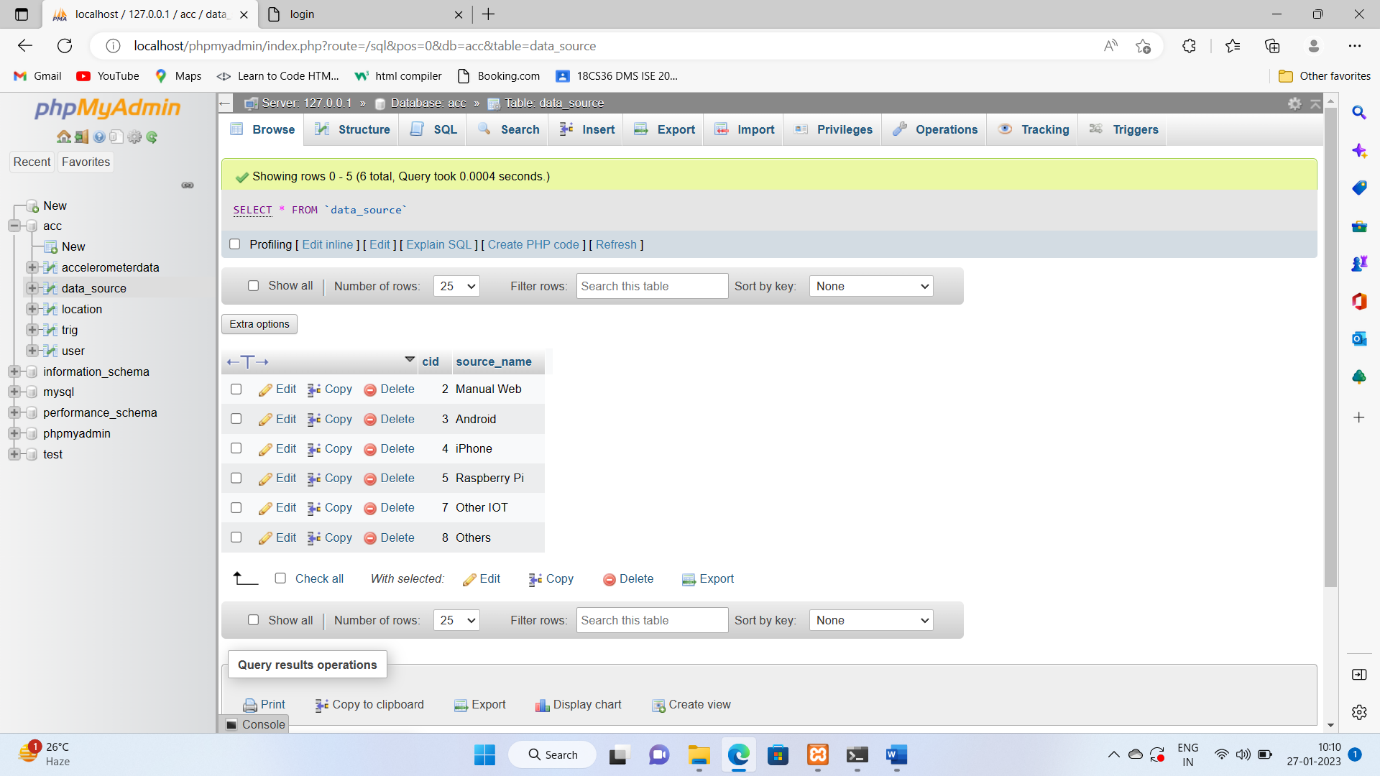
**BASE**

****

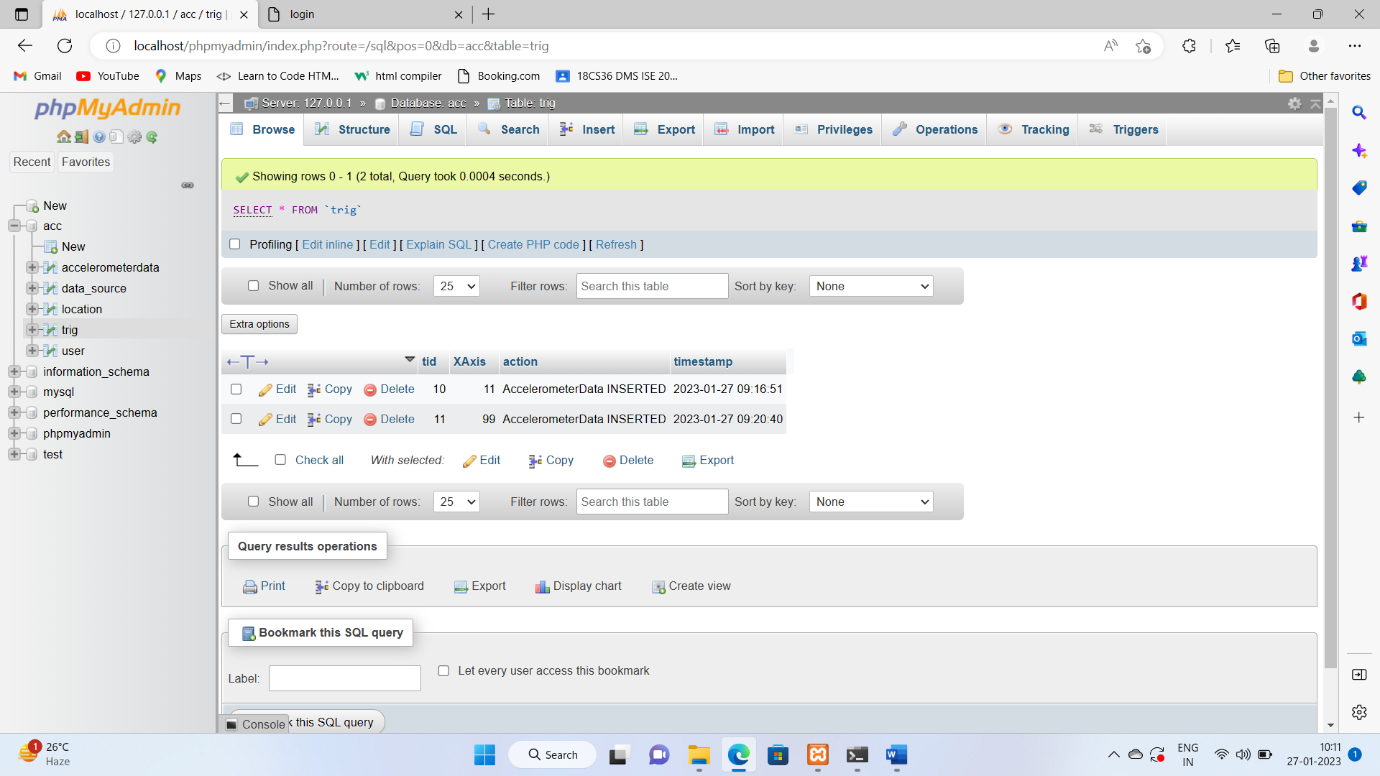
**ACCELEROMETER DATA TABLE**

****

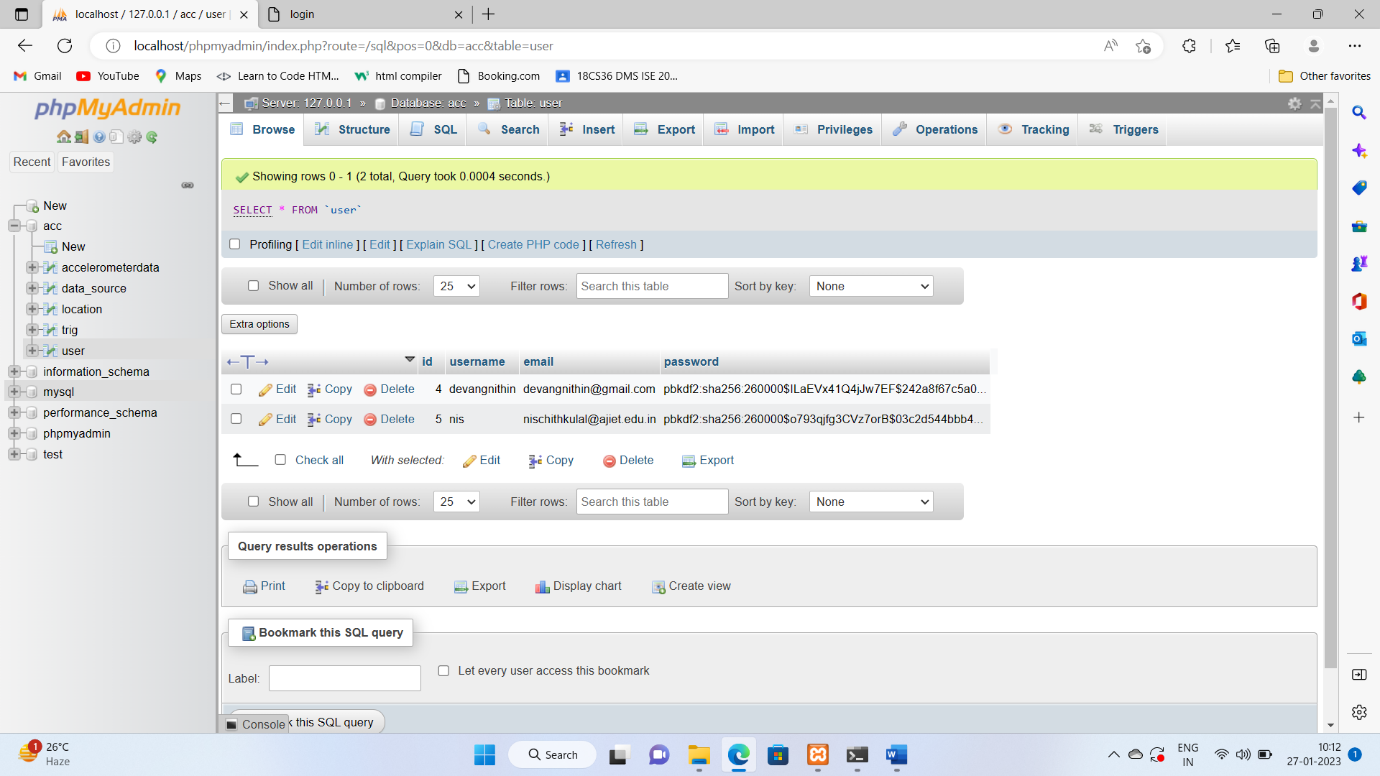
**DATA SOURCE TABLE**

****

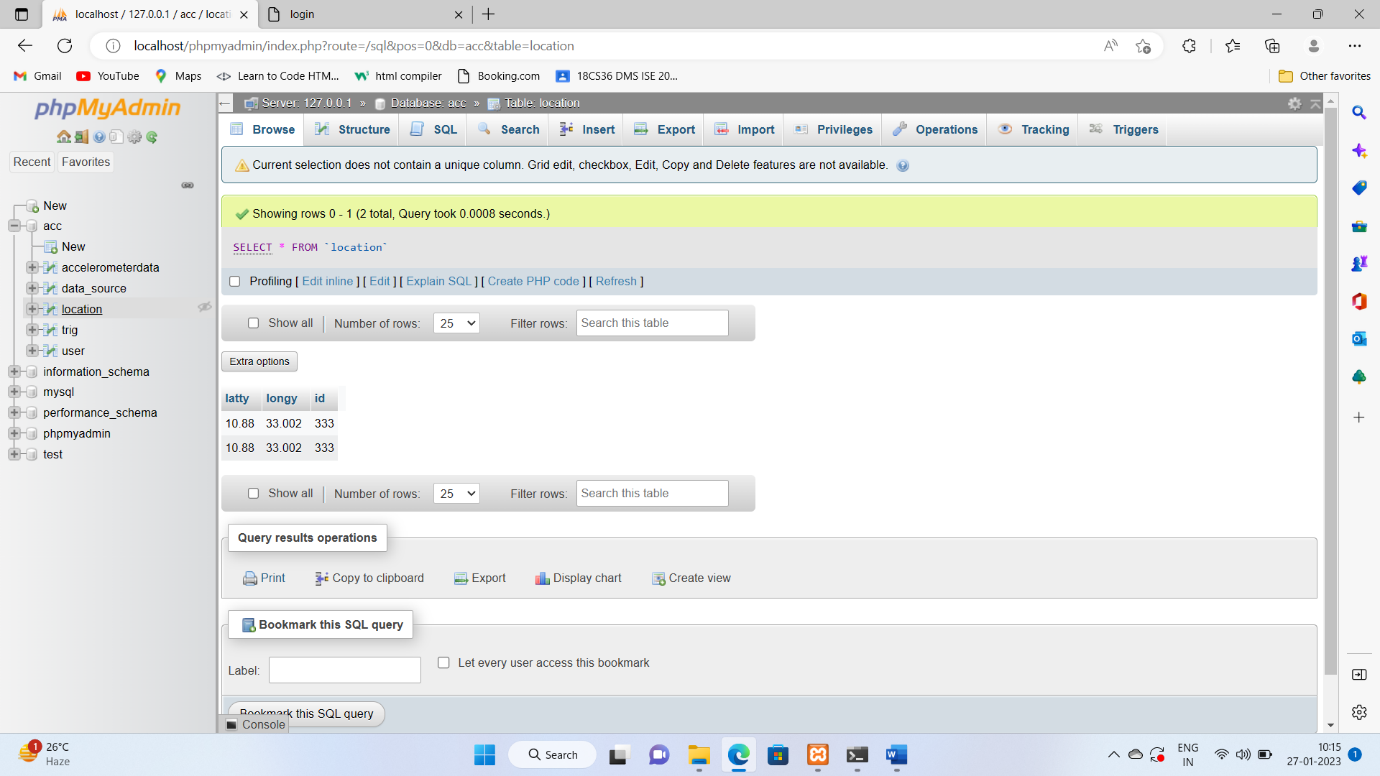
**TRIGGER TABLE**

****

**USER TABLE**

****

**LOCATION TABLE**

****

# CHAPTER 5

# CONCLUSION

1. ROAD QUALITY MONITORING SYSTEM successfully implemented based on online data filling which helps us in administrating the data user for managing the tasks to be performed for better roads. .
2. The project successfully used various functionalities of Xampp and python flask and also create the fully functional database management system for online portals
3. Using MySQL as the database is highly beneficial as it is free to download, popular and can be easily customized.
4. The data stored in the MySQL database can easily be retrieved and manipulated according to the requirements with basic knowledge of SQL.
5. With the theoretical inclination of our syllabus it becomes very essential to take the atmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project “Road Quality Monitoring System” was one of these opportunities.
6. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer.
7. The project from a personal point of view also helped us in understanding the following aspects of project development:

The planning that goes into implementing a project.

* The importance of proper planning and an organized methodology.
* The key element of team spirit and co-ordination in a successful project.

**FUTURE ENHANCEMENT**

* Enhanced database storage facility
* Enhanced user friendly GUI
* More advanced location plotting systems
* online feedbacks forms
* Application Development of fetching the accelerometer data from mobile devices

**REFERENCES**

* https://www.youtube.com
* <https://www.google.com>
* <https://www.geeksforgeeks.org>
* <https://javapoint.com>