**BASAVARAJESWARI GROUP OF INSTITUTIONS**

**BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT**

****NACC Accredited Institution\*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to**

**Visvesvaraya Technological University, Belagavi)**

**"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur,**

**Ballar1-583 104 (Karnataka) (India)**

**Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197**

**DEPARTMENT OF CSE(DATA SCIENCE)**

**A Mini-Project Report**

**On**

**“BIKE SHOWROOM MANAGEMENT ”**

**A report submitted in partial fulfillment of the requirements for the**

**Data Base Management System LAB(22CD43)**

**Submitted By**

**R ROHIT SAI USN: 3BR22CD048**

**Under the Guidance of**

**Mrs. Parvathi Asst Prof.**

**Mrs. Umadevi B E Asst. Prof.**



[**Visvesvaraya Technological University**](http://www.vtu.ac.in/)

**Belagavi, Karnataka 2021-2022**

**BASAVARAJESWARI GROUP OF INSTITUTIONS**

**BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT**

NACC Accredited Institution\*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to**

**Visvesvaraya Technological University, Belagavi)**

**"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur,**

**Ballari-583 104 (Karnataka) (India)**

**Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197**

****

**DEPARTMENT OF CSE(DATA SCIENCE)**

**CERTIFICATE**

This is to certify that the OPEN ENDED PROJECT of DBMS LAB title **“**BIKESHOWROOMMANAGEMENT**”** has been successfully presented by R ROHIT SAIbearing USN **3BR22CD048** students of semester B.E for the partial fulfillment of the requirements for the award of **Bachelor Degree in CSE(DS)** of the BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT, BALLARI during the academic year 2023-2024.

Signature of guideSignature of HOD

**Mrs. Parvathi Dr. D Aradhana**

**External viva**

**Name of the examiners Signature with date**

**1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ACKNOWLEDGEMENT**

The satisfactions that accompany the successful completion of our mini project on **“BIKE SHOWROOM MANAGEMENT ”** would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is our privilege to express our gratitude and respect to all those who inspired us in the completion of our mini-project.

We are extremely grateful to our Guides **Mrs. Parvathi and Mrs. Umadevi B E** for their noble gesture, support co-ordination and valuable suggestions given in completing the mini-project. We also thank **Dr. D Aradhana,** H.O.D. Department of CSE(DS), for his co-ordination and valuable suggestions given in completing the mini-project. We also thank Principal, Management and non-teaching staff for their co-ordination and valuable suggestions given to us in completing the Open ended project.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Name | USN |
|  |  | R ROHIT SAI | 3BR22CD048 |
|  |  |  |  |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| Ch No | Chapter Name | Page |
| 1 | Introduction   * 1. About the project   1.2 Language used | 1 |
| 2 | Scope of the project   * 1. Problem statement   2.2 Objectives | 2 |
| 3 | System requirements   * 1. Functional Requirements   2. Non-Functional Requirements   3. Hardware Requirements   3.4 Software Requirements | 3-5 |
| 4 | Methodology   * 1. E R diagram   4.2 Schema Diagram | 6-7 |
| 5 | Implementation   * 1. Table Description   2. Source Code   5.3 Connections | 8-13 |
| 6 | Result | 14-21 |
|  | Conclusion | 22 |

**Abstract**

The Bike Showroom Management System is a comprehensive software application designed to streamline the operations of a bike showroom. It encompasses a range of functionalities including inventory management, sales tracking, customer relationship management, and reporting. The system aims to enhance efficiency, reduce manual errors, and provide real-time data for informed decision-making. Developed using modern programming languages and frameworks, the system is designed to be user-friendly, scalable, and secure.

**Chapter-1**

**Introduction**

**1.1 About the Project**

The Bike Showroom Management System is a digital solution tailored for bike showrooms to manage their day-to-day operations efficiently. Traditional methods of managing a showroom, which often involve extensive paperwork and manual processes, can lead to inefficiencies, errors, and delays. This project aims to digitize and automate these processes, offering a more organized, accessible, and accurate system for managing bikes, sales, customer details, and more.

**1.2 Language used**

This project is developed using Python for the application logic and MySQL for database management. Python is utilized to build the user interface and handle database operations, while MySQL is employed to store and manage the showroom data efficiently. Together, these technologies enable effective management and organization of showroom details

**Chapter-2**

**Scope of the Project**

### 2.1 Problem Statement

Bike showrooms often face challenges in managing their inventory, sales, and customer data efficiently. The lack of an integrated system can lead to errors, mismanagement, and loss of potential sales opportunities. Manual processes are time-consuming and prone to errors, making it difficult for showroom managers to keep track of operations and make informed decisions..

**2.2 Objectives**

* To automate the management of inventory, sales, and customer data to reduce manual errors and increase efficiency.
* To provide a user-friendly interface that can be easily used by employees with varying levels of technical expertise.
* To generate accurate and timely reports that assist in decision-making and help in tracking the performance of the showroom.
* To enhance customer satisfaction by maintaining detailed customer records and providing personalized services.
* To ensure data security and integrity by implementing robust authentication and authorization mechanisms

## Chapter-3

## System Requirements

### Functional Requirements

### Inventory Management:

### Add, update, and remove bike details.

### Track stock levels and availability.

### Generate alerts for low stock levels.

### Sales Management :

### Record sales transactions.

### Generate invoices and receipts.

### Track payment statuses.

### Customer Management:

### Store customer details and purchase history

### .Generate customer reports

### .Provide personalized recommendations based on purchase history.

### User Management :

### Create and manage user accounts.

### Assign roles and permissions

### .Track user activities.

### Reporting:

### Generate sales, inventory, and customer reports.

### Export reports in various formats (PDF, Excel).

### Schedule automatic report generation.

### 3.2 Non-Functional Requirements

### Performance:

### The system should be responsive with minimal load times.

### It should handle simultaneous access by multiple users without performance degradation.

### Usability:

### The interface should be intuitive and easy to navigate.

### It should accommodate users with varying levels of technical expertise.

### Scalability:

### The system should be scalable to accommodate future growth in the number of users, inventory items, and transactions.

### Security :

### Implement authentication and role-based access control.

### Encrypt sensitive data and ensure secure data transmission.

### Reliability :

### The system should be highly reliable with minimal downtime.

### Regular backups should be performed to prevent data loss

### 3.3 Hardware Requirements

1. Server:

* Processor: Intel Xeon or equivalent
* RAM: 16 GB or higher
* Storage: 500 GB SSD
* Network: High-speed internet connection

1. Client Machines:

* Processor: Intel Core i5 or higher
* RAM: 8 GB or higher
* Storage: 256 GB SSD
* Display: 1080p resolution or higher

### 3.4 Software Requirements

* Operating System: Windows Server 2019 or Linux (for server), Windows 10 or Ubuntu (for clients)
* Database: MySQL 8.0 or higher
* Backend: Java 11, Spring Boot
* Frontend: HTML5, CSS3, JavaScript, Bootstrap
* Web Server: Apache Tomcat 9 or higher
* IDE: IntelliJ IDEA or Eclipse for development
* Version Control: Git
* Browser: Google Chrome, Mozilla Firefox

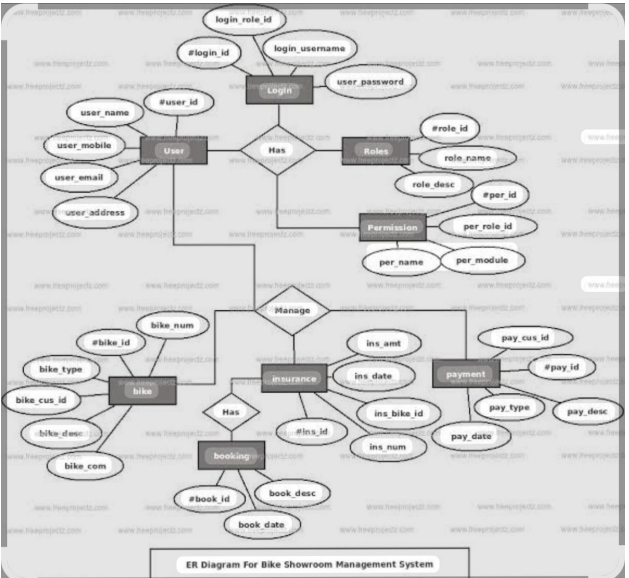
**Chapter-4**

**Methodology**

**4.1 Schema Diagram**



**4.2 ER Diagram**

 4.2Fig:Bike showroom management

**Chapter-5**

**Implementation**

**5.1 Table Description**

Bike Showroom Management Table:

Table structure: Bike\_id (INT):

This column is used to store a unique identifier for each medication. It is the primary key of the table, which means each value in this column must be unique and not null. It is used to uniquely identify each record in the table. Bike\_name (VARCHAR (50)): This column stores the name of the medication. It can hold up to 50 characters. This field is used to provide a descriptive name for each medication. Bike\_model (VARCHAR (30)): This column holds the type or category of the medication. It can store up to 30 characters. This field helps classify medications into different types. Address (VARCHAR (30)): This column stores the price of the medication. The DECIMAL (10, 2) type allows for up to 10 digits in total, with 2 digits after the decimal point. This ensures accurate representation of monetary values.

* 1. **Source Code**

import mysql.connector as mycon

# Connect to the MySQL database

con = mycon.connect(

host='localhost',

user='root',

password='MySQL$123#',

database='BIKE\_SHOWROOM' # Use the database name here

)

cur = con.cursor()

# Create the table if it does not exist

cur.execute("""

CREATE TABLE IF NOT EXISTS BIKE\_SHOWROOM(

BIKE\_ID INT PRIMARY KEY,

BIKE\_NAME VARCHAR(50),

BIKE\_MODEL VARCHAR(30),

ADRESSS VARCHAR(30)

)

""")

con.commit()

# Main program loop

while True:

print("1. ADD RECORD")

print("2. DISPLAY RECORD")

print("3. UPDATE RECORD")

print("4. DELETE RECORD")

print("0. EXIT")

choice = int(input("Enter Choice: "))

if choice == 1:

bike\_id = int(input("Enter BIKE ID: "))

bike\_name = input("Enter BIKE NAME: ")

bike\_model = input("Enter BIKE MODEL: ")

address=input("Enter ADDRESS:")

query = "INSERT INTO BIKE\_SHOWROOM(BIKE\_ID, BIKE\_NAME, BIKE\_MODEL, ADRESSS) VALUES (%s, %s, %s, %s)"

cur.execute(query, (bike\_id, bike\_name, bike\_model, address))

con.commit()

print("## Data Saved ##")

elif choice == 2:

query = "SELECT \* FROM BIKE\_SHOWROOM"

cur.execute(query)

result = cur.fetchall()

print("{:<10} {:<30} {:<20} {:<10}".format("BIKE\_ID", "BIKE\_NAME", "BIKE\_MODEL", "ADRESSS"))

for row in result:

print("{:<10} {:<30} {:<20} {:<10}".format(row[0], row[1], row[2], row[3]))

elif choice == 3:

bike\_id = int(input("Enter BIKE ID: "))

bike\_name = input("Enter BIKE NAME: ")

bike\_model = input("Enter BIKE MODEL: ")

address = input("Enter ADDRESS: ")

query = """

UPDATE BIKE\_SHOWROOM

SET BIKE\_NAME = %s,

BIKE\_MODEL = %s,ADRESSS = %s

WHERE BIKE\_ID = %s

"""

cur.execute(query, (bike\_name, bike\_model, address, bike\_id))

con.commit()

if cur.rowcount > 0:

print("## Data Updated ##")

else:

print("## BIKE\_SHOWROOM ID not found ##")

elif choice == 4:

bike\_id = int(input("Enter BIKE\_SHOWROOM ID to delete: "))

query = "DELETE FROM BIKE\_SHOWROOM WHERE BIKE\_ID = %s"

cur.execute(query, (bike\_id,))

con.commit()

if cur.rowcount > 0:

print("## Data Deleted ##")

else:

print("## BIKE\_SHOWROOM ID not found ##")

elif choice == 0:

con.close()

print("## Bye!! ##")

Break

else:

print("## INVALID CHOICE ##")

**5.3 Connections**

The database connection is established using the following parameters:

con = mycon.connect(host='localhost', user='root', password="user")

Connection details:

* Host: localhost
* User: root
* Password: user
* Database: music (created if not exists)

The connection is initiated at the start of the program and closed when the user chooses to exit.

Additional implementation details:

1. The program uses prepared statements for database operations to prevent SQL injection.
2. Date input is validated and converted to the appropriate format before insertion.
3. The display function formats the output into a table-like structure for easy reading.
4. The program allows for multiple insertions and deletions in a single session.
5. Update operations are field-specific, allowing users to choose which attribute to modify.
6. The program handles potential errors, such as duplicate primary keys or invalid date formats, providing user-friendly error messages.

This implementation provides a robust, user-friendly interface for managing music data, with built-in error handling and data validation to ensure data integrity.

**Chapter-6**

**Result**

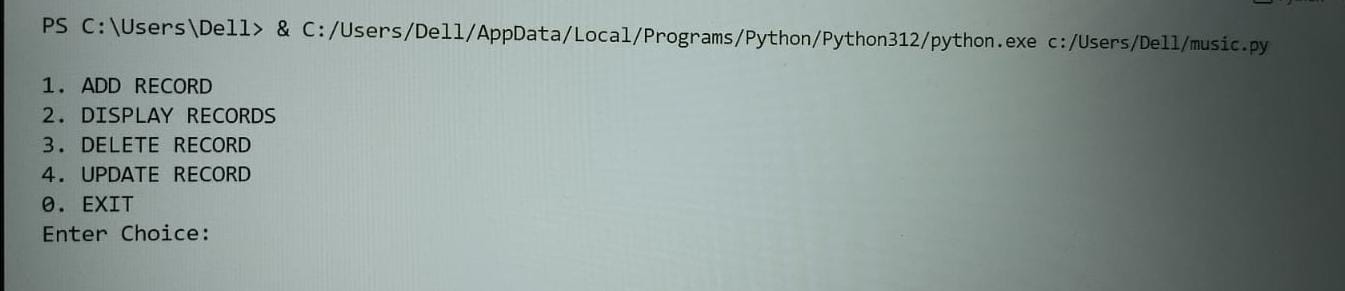
Outcome:A streamlined bike showroom management system that efficiently manages bike\_id, Bike\_name, Bike\_model and Address, offering a secure and user-friendly interface for accurate data hadling. 

Fig 6.1: Menu of the program

--------------------------------------Insertion operation--------------------------------------------

1. ADD RECORD

2. DISPLAY RECORDS

3. UPDATE RECORD

4. DELETE RECORD

0. EXIT

Enter Choice: 1

Enter Bike\_ID:

Enter Bike\_Name:

Enter Bike\_Model:

Enter Address:

## Data Saved ##

1. ADD RECORD

2. DISPLAY RECORDS

3. UPDATE RECORD

4. DELETE RECORD

0. EXIT

Enter Choice: 1

Enter Bike\_ID:

Enter Bike\_Name:

Enter Bike\_Model:

Enter Address:

## Data Saved ##

1. ADD RECORD

2. DISPLAY RECORDS

3. UPDATE RECORD

4. DELETE RECORD

0. EXIT

Enter Choice: 1

Enter Bike\_ID:

Enter Bike\_Name:

Enter Bike\_Model:

Enter Address:

## Data Saved ##

1. ADD RECORD

2. DISPLAY RECORDS

3. UPDATE RECORD

4. DELETE RECORD

0. EXIT

Enter Bike\_ID:

Enter Bike\_Name:

Enter Bike\_Model:

Enter Address:

## Data Saved ##

1. ADD RECORD

2. DISPLAY RECORDS

3. UPDATE RECORD

4. DELETE RECORD

0. EXIT

Enter Choice: 2

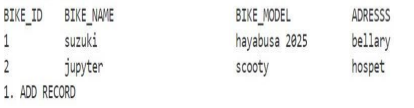
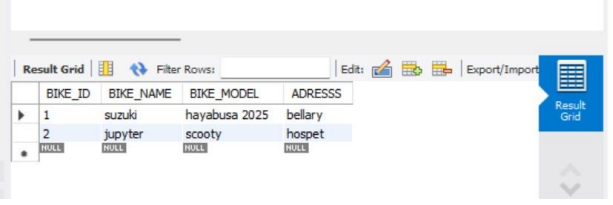


Fig 6.2:Display operation vs code and workbench



------------------------------------------Update operation-----------------------------------------------

1. ADD RECORD

2. DISPLAY RECORDS

3. UPDATE RECORD

4. DELETE RECORD

0. EXIT

Enter Choice: 3

Enter Bike\_ID to Update: 1

Enter new Bike\_Name:

Enter new Bike\_Model:

Enter new Address:

## Record Updated ##

1. ADD RECORD

2. DISPLAY RECORDS

3. UPDATE RECORD

4. DELETE RECORD

0. EXIT

Enter Choice: 2

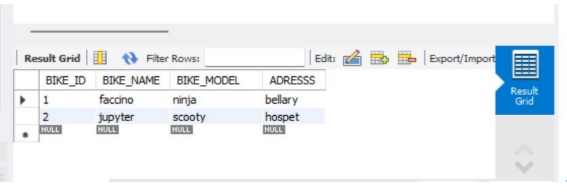


Fig 6.3:Updated output

----------------------------------------------Delete operation----------------------------------------------

1. ADD RECORD

2. DISPLAY RECORDS

3. UPDATE RECORD

4. DELETE RECORD

0. EXIT

Enter Choice: 4

Enter Bike\_ID to delete: 2

## Record Deleted ##

1. ADD RECORD

2. DISPLAY RECORDS

3. UPDATE RECORD

4. DELETE RECORD

0. EXIT

Enter Choice: 2

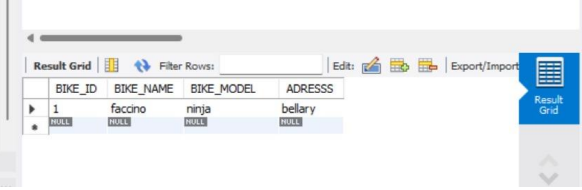


Fig 6.4: Deletion Operation

1. ADD RECORD

2. DISPLAY RECORDS

3. DELETE RECORD

4. UPDATE RECORD

0. EXIT

Enter Choice: 0

## Bye!! ##

------------------------------------------------Exit--------------------------------------------------------

**Conclusion**

The Bike Showroom Management System is an essential tool for modernizing and optimizing the operations of bike showrooms. By automating key processes such as inventory management, sales tracking, and customer relationship management, the system reduces the dependency on manual tasks, thereby minimizing errors and improving efficiency.The use of robust technologies like Java, Spring Boot, and MySQL ensures that the system is both scalable and secure, capable of meeting the demands of growing businesses. The project also emphasizes user-friendliness, ensuring that the system can be easily used by individuals with varying levels of technical expertise.Through this project, we address the core challenges faced by bike showrooms—such as inefficient inventory tracking, manual sales processing, and inadequate customer data management—by providing a comprehensive, integrated solution. The successful implementation of this system will lead to enhanced operational efficiency, better decision-making, and ultimately, improved customer satisfaction.Overall, the Bike Showroom Management System stands as a significant step towards digital transformation, empowering bike showrooms to thrive in a competitive market through the use of cutting-edge technology