



UNIVERSITY GYM MANAGEMENT SYSTEM

NAVYA PANT 2022UEE6605

SHITIZ SOLANKI 2022UEE4634

AAYUSHI 2022UEE4617



CHAPTER 1: INTRODUCTION

1.1 Introduction of Problem Area

University gym management is a critical aspect of campus life, ensuring the effective and safe utilization of fitness facilities. From resource allocation to safety protocols and community engagement, proficient management plays a pivotal role in fostering a healthy and inclusive environment for all members of the university community. This introduction highlights the importance of addressing key challenges in gym management to enhance the overall well-being and satisfaction of students, faculty, and staff.

1.2 : Problem Statement

The aim of implementing a Gym Management System is to optimize operational efficiency, enhance member experience, and ensure effective resource utilization within the gym facility. By streamlining administrative tasks, improving safety measures, and leveraging data for informed decision-making, the system aims to create a dynamic and member-centric environment conducive to health and fitness goals.

1.3 : Objective of the project

1. Manual data management: The entries of students are done manually which gives rise to discrepancies. It also makes it nearly impossible to verify the identity of the student who is making the entry. This makes it difficult to hold someone responsible for the damages caused, if any.

2. Crowd management: There is no cap on how many students are allowed to access the gym at a time. Similarly, there is no time constraint provided to the students for gym usage. Even if such constraints are imposed, there is no way to maintain order.

3. Absence of trainer: Human bodies are fragile. Incorrect exercising patterns or over-exercising can often prove to be threatening. Since our gym is free of cost, it is accessible to anybody and everybody which includes gentry that may not be educated enough in this sphere.

4. Inventory management: Our gym has facilities to provide indoor games such as chess, carrom board, etc. It is also equipped with state-of-the-art cardio and strength equipment. There is no way to ensure its management.

5. Students bunking/ skipping classes: There is no way to ensure whether or not a student is accessing the gym in the middle of their class time.

6. Employee duty management: There are guards/employees on duty in the gym at all times. The duty management of these individuals is difficult.

Photograph of the gym of Netaji Subhas University of Technology





CHAPTER 2: REQUIREMENTS

PEPTOX

2.1 : Hardware requirement

Hardware Requirements for Project Implementation:

1.Computer/Desktop:

- Operating System: Compatible with Windows/MAC OS / Linux
- Processor: Intel Pentium Duo
- RAM: Recommended 8GB
- Storage: Minimum 1GB

2. Mobile Phone:

- Platform: Android/iOS
 - Operating System: Compatible with Android version 7.4 or higher
 - RAM: Recommended 4GB
- **Storage:** Minimum 100MB These refined details ensure a seamless integration of the project on both computer and mobile platforms.

2.2 : Software Tools

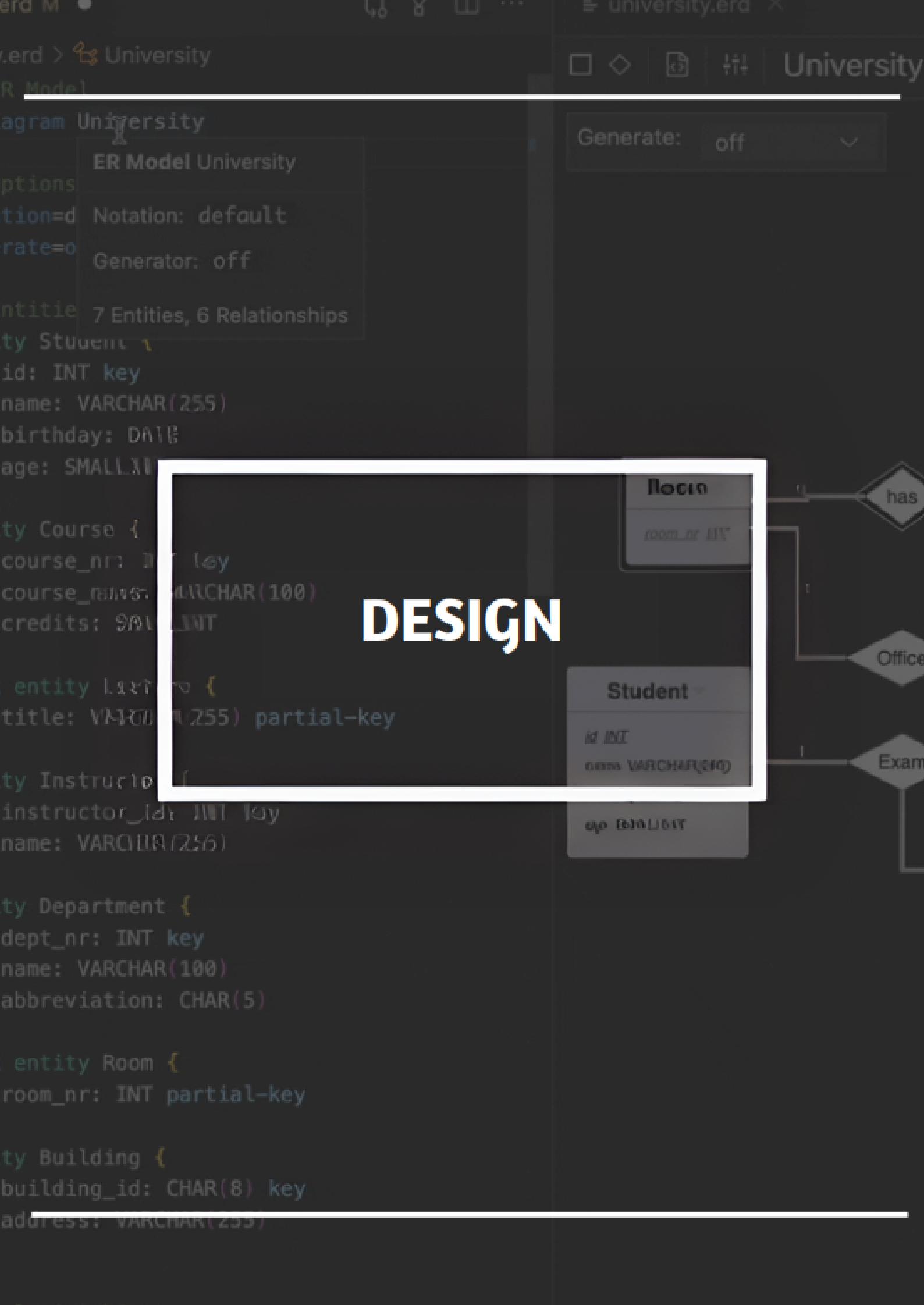
Software/Tech Stacks/Languages Used:

XAMPP:

- Description: XAMPP is a comprehensive software package that includes an RDBMS (Relational Database Management System) for creating databases and executing SQL queries.
- Purpose: Used for database management and interaction with the project's data.

APACHE NETBEANS IDE 18:

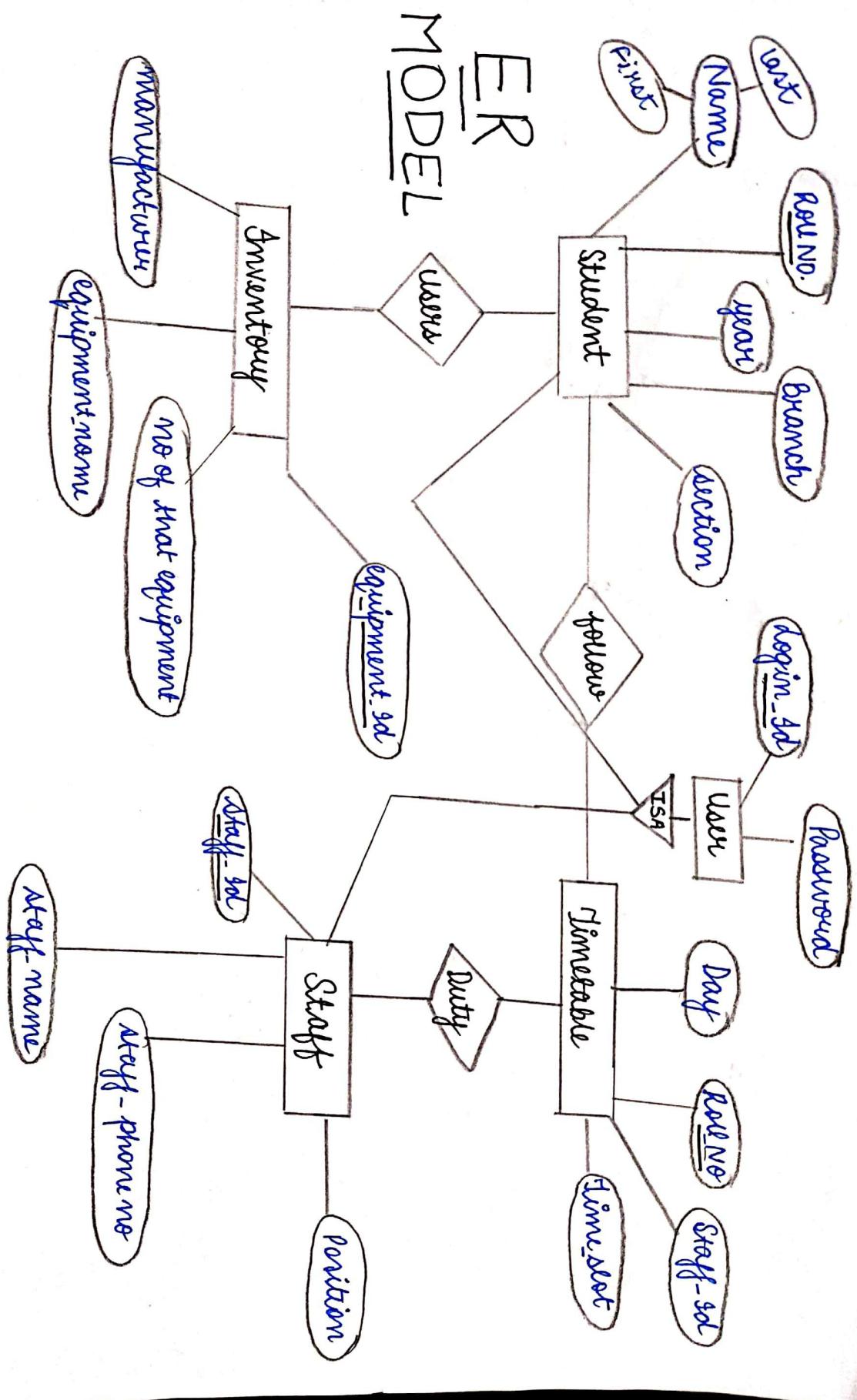
- NetBeans IDE, with XAMPP, supports database tasks via JDBC integration, SQL editing, project management, version control, testing, debugging, and deployment.
- It complements XAMPP's DBMS capabilities for efficient development.



3.1 ER MODEL



Scanned with
CamScanner



ER to Schema

1. Student (name, roll no, year, branch, section)
2. Timetable (day, roll no, staff Id, time slot)
3. Staff (staff Id, staff-Phone no., staff name, position)
4. Inventory (Manufacturer, equipment name, no. of that equipment), equipment Id)
5. User (Login Id, Password)

Primary Keys

1. Student : Roll No
2. Timetable : Roll No
3. Staff : Staff - Id
4. Inventory : equipment Id
5. User : login Id

NORMALISATION (B.C.N.F.)

BCNF Form:

Rule: LHS should be candidate key or super key

1. Student table:

FD: Roll no. \rightarrow name, year, branch, section

Closure of [Roll no.] = name, year, branch, section

Candidate key: {Roll no.}

Prime attribute: {Roll no.}

Roll no. \rightarrow name, year, branch, section (T)

Hence, in BCNF form

2. Timetable table:

FD: Roll no. \rightarrow Day, Time-slot

Closure of [Roll no.] = Day, time-slot

Candidate key: {Roll no.}

Prime attribute: {Roll no.}

Roll no. \rightarrow Day, time-slot (T)

Hence, in BCNF form

3. Staff table:

FD: staff-id \rightarrow staff-name, staff-phone no., position

Closure of [staff-id] = staff-name, staff-phone no.,
position

Candidate key : {staff-id}

Prime attribute: {staff-id}

staff-id \rightarrow staff-name, staff-phone no., position (T)

Hence, in BCNF form

4. Inventory table:

FD: Equipment-id \rightarrow manufacturer, equipment-name,
no. of that equipment
closure of [Equipment-id] = manufacturer, equipment-
name, no.of that equipment

Candidate key : {Equipment-id}

Prime attribute: {Equipment-id}

Equipment-id \rightarrow manufacturer, equipment-name,
no.of that equipment (T)

Hence, in BCNF form

5. User table:

FD: Login-id \rightarrow Password

closure of [Login-id] = Password

Candidate key : {Login-id}

Prime attribute: {Login-id}

Login-id \rightarrow Password (T)

Hence, in BCNF form

Functional Dependencies

1. Student: Roll no, \rightarrow name, \rightarrow year, branch, section
2. Timetable: Roll no \rightarrow Day, staff-id, time slot
3. Staff: Staff id \rightarrow staff name, staff phone no, position
4. Inventory:- EquipmentId \rightarrow manufacturer, equipment-
name, equip no of that
equipment
5. User : loginId \rightarrow password

CODE AND QUERIES

```
package jdbc;
import java.sql.*;
import java.util.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class JDBC {
    public static void main(String[] args) {
        // creates Connection object
        Connection conn1 = null;
        try {
            // connect way #1
            String url1 = "jdbc:mysql://localhost:3306/university gym management";
            String user = "root";
            String password = "";
            conn1 = DriverManager.getConnection(url1, user, password);
            if (conn1 != null) {
                System.out.println("Connected to the database university gym management");
            }
        }
        catch (SQLException ex) {
            System.out.println("An error occurred. Maybe user/password is invalid");
            ex.printStackTrace();
        }
        finally {
            if (conn1 != null) {
                try {
                    conn1.close();
                }
                catch(Exception ex) {
                    ex.printStackTrace();
                }
            }
        }
    }
}
```

```
package jdbc;
import java.sql.*;
import java.util.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.HashMap;

public class JDBC {
    static final String DB_URL = "jdbc:mysql://localhost/university gym management";
    static final String USER = "root";
    static final String PASS = "";

    public static void main(String[] args) {
        // Open a connection
        try(Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
            Statement stmt = conn.createStatement();
        ) {
            // Execute a query
            System.out.println("Inserting records into the table...");

            String sql = "INSERT INTO student VALUES ('2022UIT2940',
'UTKARSH',2026,'IT', 2,1200,4000)";

            stmt.executeUpdate(sql);

            System.out.println("Inserted records into the table...");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

The screenshot shows the NetBeans IDE interface with the following details:

- Projects Tab:** Contains a project named "JDBC". Inside "Source Packages", there is a package named "jdbc" with a file named "JDBC.java". Other packages like "Test Packages" and "Libraries" are also listed.
- Start Page Tab:** Shows the code for "JDBC.java". The code imports various Java.sql classes and defines a static final String for the database URL. It then attempts to connect to the database and insert records into a table.
- Output Tab:** Displays the run output. It shows the application inserting records into a table and confirming successful insertion. The build log indicates a total time of 0 seconds.
- System Tray:** Shows the date and time as 11:16 PM, and the system language as ENG.

```
1 package jdbc;
2 import java.sql.*;
3 import java.util.*;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6 import java.sql.SQLException;
7 import java.util.Properties;
8 import java.sql.ResultSet;
9 import java.util.ArrayList;
10 import java.util.HashMap;
11
12 public class JDBC {
13     static final String DB_URL = "jdbc:mysql://localhost/university_gym_management";
14     static final String USER = "root";
15     static final String PASS = "";
16
17     public static void main(String[] args) {
18         // Open a connection
19         try(Connection conn = DriverManager.getConnection(url:DB_URL, user:USER, password:PASS);
20             Statement stmt = conn.createStatement();
21
22             run:
23             Inserting records into the table...
24             Inserted records into the table...
25             BUILD SUCCESSFUL (total time: 0 seconds)
26
27             System.out.println("Inserted records into the table...");
```

The screenshot shows the NetBeans IDE interface with the following details:

- Projects Tab:** Contains a project named "JDBC". Inside "Source Packages", there is a package named "jdbc" with a file named "JDBC.java". Other packages like "Test Packages" and "Libraries" are also listed.
- Start Page Tab:** Shows the code for "JDBC.java". The code imports various Java.sql classes and defines a static final String for the database URL. It then attempts to connect to the database and insert records into a table.
- Output Tab:** Displays the run output. It shows the application inserting records into a table and confirming successful insertion. The build log indicates a total time of 0 seconds.
- System Tray:** Shows the date and time as 11:16 PM, and the system language as ENG.

```
17     public static void main(String[] args) {
18         // Open a connection
19         try(Connection conn = DriverManager.getConnection(url:DB_URL, user:USER, password:PASS);
20             Statement stmt = conn.createStatement();
21
22             run:
23             Inserting records into the table...
24             Inserted records into the table...
25             BUILD SUCCESSFUL (total time: 0 seconds)
26
27             System.out.println("Inserted records into the table...");
```

The screenshot shows the NetBeans IDE interface with the following details:

- Projects View:** Shows a project named "JDBC" containing "Source Packages" (with "JDBC" selected), "Test Packages", "Libraries" (containing "C:\Users\navya\Desktop\mysql-connector-j-8.3.0.jar"), and "Test Libraries".
- Code Editor:** Displays the "JDBC.java" file with the following code:

```
1  static final String PASS = "";
2
3  public static void main(String[] args) {
4      // Open a connection
5      try(Connection conn = DriverManager.getConnection(url:DB_URL, user:USER, password:PASS)) {
6          Statement stmt = conn.createStatement();
7      }
8      // Execute a query
9      System.out.println("Inserting records into the table...");
10
11     String sql = "INSERT INTO inventory VALUES ('74PDS', 'BENCHPRESS', 2, 'Fitline')";
12
13     stmt.executeUpdate(string: sql);
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32 }
```

- Output Window:** Shows the results of the run:

```
run:
Inserting records into the table...
Inserted records into the table...
BUILD SUCCESSFUL (total time: 0 seconds)
```

The screenshot shows the NetBeans IDE interface with the following details:

- Projects View:** Shows a project named "JDBC" containing "Source Packages" (with "JDBC" selected), "Test Packages", "Services", and "Files".
- Code Editor:** Displays the "JDBC.java" file with the following code:

```
1  package jdbc;
2
3  import java.sql.*;
4  import java.util.*;
5  import java.sql.Connection;
6  import java.sql.DriverManager;
7  import java.sql.SQLException;
8  import java.util.Properties;
9  import java.sql.ResultSet;
10 import java.util.ArrayList;
11 import java.util.HashMap;
12
13 public class JDBC {
14     static final String DB_URL = "jdbc:mysql://localhost/university_gym_management";
15     static final String USER = "root";
16     static final String PASS = "";
17
18     public static void main(String[] args) {
19         // Open a connection
20         try(Connection conn = DriverManager.getConnection(url:DB_URL, user:USER, password:PASS)) {
21             Statement stmt = conn.createStatement();
22         }
23         // Execute a query
24         System.out.println("Inserting records into the table...");
25
26         String sql = "INSERT INTO staff VALUES ('EXIT119', 'ken', 7897394, 'Security')";
27
28         stmt.executeUpdate(string: sql);
29
30
31
32 }
```

- Output Window:** Shows the results of the run:

```
run:
Inserting records into the table...
Inserted records into the table...
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
package jdbc;
import java.sql.*;
import java.util.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.HashMap;

public class JDBC {
    static final String DB_URL = "jdbc:mysql://localhost/university gym management";
    static final String USER = "root";
    static final String PASS = "";
    static final String QUERY = "SELECT Equipment_id, Equipment_name, no_of_that_equipment,
Manufacturer FROM inventory";

    public static void main(String[] args) {
        // Open a connection
        try( Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
            Statement stmt = conn.createStatement();
        ) {

            ResultSet rs = stmt.executeQuery(QUERY);
            while(rs.next()){
                //Display values
                System.out.print("Equipment_id: " + rs.getString("Equipment_id"));
                System.out.print(", Equipment_name: " + rs.getString("Equipment_name"));
                System.out.print(", no_of_that_equipment: " + rs.getInt("no_of_that_equipment"));
                System.out.println(", Manufacturer: " + rs.getString("Manufacturer"));
            }
            rs.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

The screenshot shows a Java code editor with the following code:

```
package jdbc;
import java.sql.*;
import java.util.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.HashMap;

public class JDBC {
    static final String DB_URL = "jdbc:mysql://localhost/university gym management";
    static final String USER = "root";
    static final String PASS = "";
    static final String QUERY = "SELECT Equipment_id, Equipment_name, no_of_that_equipment, Manufacturer FROM inventory";

    public static void main(String[] args) {
        // Open a connection
        try(Connection conn = DriverManager.getConnection(url:DB_URL, user:USER, password:PASS);{
            Statement stmt = conn.createStatement();
        ) {

            ResultSet rs = stmt.executeQuery(string: QUERY);
            while(rs.next()){
                //Display values
                System.out.print("Equipment_id: " + rs.getString(string: "Equipment_id"));
                System.out.print(", Equipment_name: " + rs.getString(string: "Equipment_name"));
                System.out.print(", no_of_that_equipment: " + rs.getInt(string: "no_of_that_equipment"));
                System.out.println(", Manufacturer: " + rs.getString(string: "Manufacturer"));
            }
            rs.close();
        }
    }
}
```

The code is for a JDBC application that connects to a MySQL database named "university gym management". It uses a static final string for the database URL, and static final strings for the user ("root") and password (""). The query selects four columns from the "inventory" table: Equipment_id, Equipment_name, no_of_that_equipment, and Manufacturer. The results are printed to the console.

Output

The screenshot shows the "Output - JDBC (run)" window with the following output:

```
Output - JDBC (run)
run:
Equipment_id: 0SPD03, Equipment_name: PECTORAL FLY REAR DELTOID, no_of_that_equipment: 1, Manufacturer: Fitline
Equipment_id: 210s, Equipment_name: BENCHPRESS, no_of_that_equipment: 2, Manufacturer: Fitline
Equipment_id: 489HU, Equipment_name: TREADMILL, no_of_that_equipment: 3, Manufacturer: Body Strong
Equipment_id: 67GH, Equipment_name: SHOULDER PRESS MACHINE, no_of_that_equipment: 2, Manufacturer: Fitline
Equipment_id: 74PDS, Equipment_name: BENCHPRESS, no_of_that_equipment: 2, Manufacturer: Fitline
Equipment_id: OS_LD_04, Equipment_name: PULL_DOWN_MACHINE, no_of_that_equipment: 2, Manufacturer: FIT_LINE
BUILD SUCCESSFUL (total time: 0 seconds)
```

The output displays the results of the SQL query executed by the JDBC code. It lists six pieces of equipment with their respective details: Equipment_id, Equipment_name, no_of_that_equipment, and Manufacturer. The output concludes with a "BUILD SUCCESSFUL" message.

```
package jdbc;
import java.sql.*;
import java.util.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.HashMap;

import com.mysql.jdbc.Statement;

public class JDBC {
    public static void main(String[] args) {
        Connection con = null;
        Statement statement = null;
        try {
            HashMap hm = new HashMap<>();
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/university gym management",
"root", "");
            statement = (Statement) con.createStatement();
            String sql;
            sql = "select * from student";
            ResultSet resultSet = statement.executeQuery(sql);
            while (resultSet.next()) {
                hm.put(resultSet.getString("roll_number"), resultSet.getString("name"));
            }
            System.out.println(hm);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

The screenshot shows the Apache NetBeans IDE interface. The title bar reads "JDBC - Apache NetBeans IDE TO". The main window displays a Java code editor with the following content:

```
package jdbc;
import java.sql.*;
import java.util.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.HashMap;

import com.mysql.jdbc.Statement;

public class JDBC {
    public static void main(String[] args) {
        Connection con = null;
        Statement statement = null;
        try {
            HashMap hm = new HashMap<>();
            con = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/university_gym_management", user:"root", password:"");
            statement = (Statement) con.createStatement();
            String sql;
            sql = "select * from student";
            ResultSet resultSet = statement.executeQuery(string: sql);
            while (resultSet.next()) {
                hm.put(key:resultSet.getString(string: "roll_number"), value: resultSet.getString(string: "name"));
            }
            System.out.println(: hm);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

The code uses JDBC to connect to a MySQL database named "university_gym_management" running on localhost at port 3306. It selects all columns from the "student" table and stores the results in a HashMap where the key is the roll number and the value is the name. The output is then printed to the console.

Output

The screenshot shows the "Output" window titled "Output - JDBC (run)". The window displays the following log entries:

```
run:
{2022ME4678=ARYAN MODI, 2022UIC3207=HARSH, 2022UEE4633=Ananya, 2022UIT2940=UTKARSH, 2022UBT1906=ANSHITA, 2022UEC3786=AMAN, 2022UEC3746=CHINMAY}
BUILD SUCCESSFUL (total time: 1 second)
```

The output shows the names of the students from the database, indicating that the program ran successfully and printed the expected results.

TABLES

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=university+gym+management&table=student

```
SELECT * FROM `student`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	roll_number	name	student_year	student_branch	section	check_in	check_out
<input type="checkbox"/>	2022ME4678	ARYAN MODI	2026	ME	2	00:01:00	00:04:00
<input type="checkbox"/>	2022UBT1906	ANSHITA	2026	BT	1	00:04:30	00:05:00
<input type="checkbox"/>	2022UEC3746	CHINMAY	2025	ECE	1	00:01:00	00:05:00
<input type="checkbox"/>	2022UEC3786	AMAN	2026	ECE	3	00:11:00	00:03:00
<input type="checkbox"/>	2022UEE4633	Ananya	2026	EE	2	00:10:10	00:12:12
<input type="checkbox"/>	2022UIC3207	HARSH	2026	ICE	1	00:09:09	00:11:01
<input type="checkbox"/>	2022UIT2940	UTKARSH	2026	IT	2	00:12:00	00:04:00

Check all With selected: Edit Copy Delete Export

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=university+gym+management&table=staff&pos=0

```
SELECT * FROM `staff`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	staff_id	staff_name	staff_phone_no	position
<input type="checkbox"/>	EXIT114	RAHUL	98765432	TRAINER
<input type="checkbox"/>	EXIT115	ROHAN	8595852	TRAINER
<input type="checkbox"/>	EXIT118	CHITRANSHU	8287709	Receptionist
<input type="checkbox"/>	EXIT119	ken	7897394	Security

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

ema
schema
management

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

Equipment_id Equipment_name no_of_that_equipment Manufacturer

0SPD03	PECTORAL FLY REAR DELTOID	1	Fitline
21Os	BENCHPRESS	2	Fitline
489HU	TREADMILL	3	Body Strong
67GH	SHOULDER PRESS MACHINE	2	Fitline
74PDS	BENCHPRESS	2	Fitline
OS_LD_04	PULL_DOWN_MACHINE	2	FIT_LINE

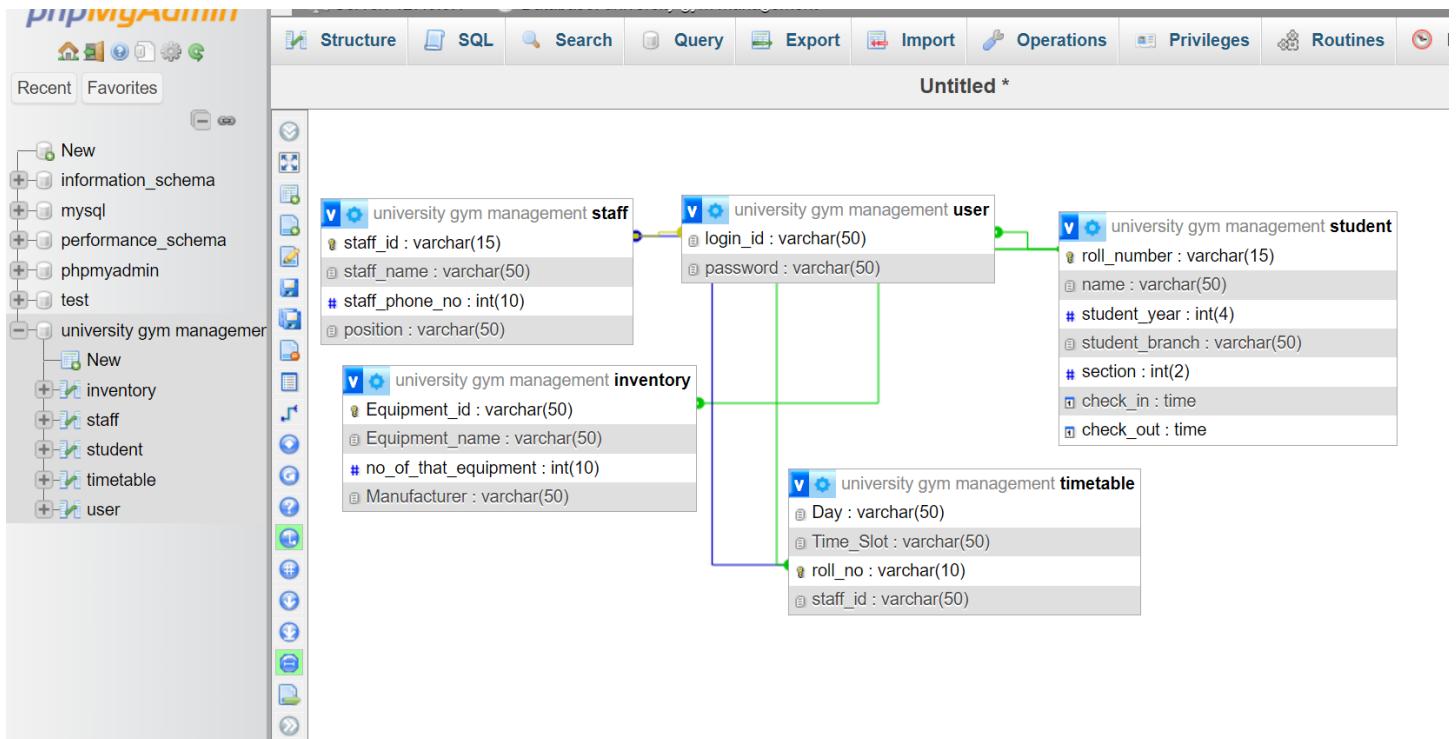
Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view Console

DESIGN



REFERENCES

- 1) NSUT Administration Department for data
- 2) Dr. Anand Gupta , Associate Professor(DBMS)