**PROJECT REPORT**

**ON**

**TASK MANAGEMENT SYSTEM**

**Developed By**

**Abhishek D Borisaniya Roll No: MCA004**

**Nikhil R Lathiya** **Roll No: MCA024**

**Submitted To**



**Master of Computer Application**

**Faculty of Management and Information Science**

**DHARMSINH DESAI UNIVERSITY**

**APRIL — 2024**

**CODE :-**

import java.sql.\*;

import java.util.Scanner;

public class termwork {

private static final String URL = "jdbc:mysql://localhost:3306/task\_management\_system";

private static final String USER = "root";

private static final String PASSWORD = "123456";

public static void main(String[] args) {

try {

Connection connection = DriverManager.getConnection(URL, USER, PASSWORD);

Statement statement = connection.createStatement();

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("=======================================================");

System.out.println("\t\tTask Management System");

System.out.println("=======================================================\n");

System.out.println("1.-> Add Task");

System.out.println("2.-> View All Tasks");

System.out.println("3.-> View Task by ID");

System.out.println("4.-> Update Task");

System.out.println("5.-> Update Task Progress");

System.out.println("6.-> Delete Task");

System.out.println("7.-> Exit");

System.out.print("\nChoose an option: ");

int choice = scanner.nextInt();

scanner.nextLine();

switch (choice) {

case 1:

addTask(connection, scanner);

break;

case 2:

viewAllTasks(statement);

break;

case 3:

viewTaskById(connection, scanner);

break;

case 4:

updateTask(connection, scanner);

break;

case 5:

updateTaskProgress(connection, scanner);

break;

case 6:

deleteTask(connection, scanner);

break;

case 7:

System.out.println("Exiting...");

connection.close();

scanner.close();

return;

default:

System.out.println("Invalid choice. Please try again.");

}

}

} catch (SQLException e) {

e.printStackTrace();

}

}

private static void addTask(Connection connection, Scanner scanner) throws SQLException {

System.out.println("\n\n\t\tAdd Task");

System.out.println("---------------------------------------------\n");

System.out.print("Enter task name: ");

String name = scanner.nextLine();

System.out.print("Enter task description: ");

String description = scanner.nextLine();

System.out.print("Enter task priority: ");

int priority = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter due date (YYYY-MM-DD): ");

String dueDate = scanner.next();

System.out.print("Enter assigned to: ");

String assignedTo = scanner.next();

System.out.print("Enter task status: ");

String status = scanner.next();

String insertQuery = "INSERT INTO tasks (name, description, priority, due\_date, assigned\_to, status, progress) "

+

"VALUES (?, ?, ?, ?, ?, ?, 0)";

PreparedStatement preparedStatement = connection.prepareStatement(insertQuery);

preparedStatement.setString(1, name);

preparedStatement.setString(2, description);

preparedStatement.setInt(3, priority);

preparedStatement.setString(4, dueDate);

preparedStatement.setString(5, assignedTo);

preparedStatement.setString(6, status);

preparedStatement.executeUpdate();

System.out.println("========================================================");

System.out.println("\tTask added successfully.");

System.out.println("========================================================");

System.out.println("\n\n\n");

}

private static void viewAllTasks(Statement statement) throws SQLException {

System.out.println("\n\n\t\tAll Tasks");

System.out.println("---------------------------------------------\n");

ResultSet resultSet = statement.executeQuery("SELECT \* FROM tasks");

boolean tasksExist = false; // Flag to track if any tasks exist

while (resultSet.next()) {

tasksExist = true;

System.out.println("-------------------------------------------------------");

System.out.println("Task ID: " + resultSet.getInt("id"));

System.out.println("Name: " + resultSet.getString("name"));

System.out.println("Description: " + resultSet.getString("description"));

System.out.println("Priority: " + resultSet.getInt("priority"));

System.out.println("Due Date: " + resultSet.getDate("due\_date"));

System.out.println("Assigned To: " + resultSet.getString("assigned\_to"));

System.out.println("Status: " + resultSet.getString("status"));

System.out.println("Progress: " + resultSet.getInt("progress") + "%");

System.out.println("-------------------------------------------------------");

}

if (!tasksExist) {

System.out.println("========================================================");

System.out.println("\tNo tasks available.");

System.out.println("========================================================");

System.out.println("\n\n\n");

}

}

private static void viewTaskById(Connection connection, Scanner scanner) throws SQLException {

System.out.println("\n\n\t\tView Task by ID");

System.out.println("---------------------------------------------\n");

System.out.print("Enter task ID: ");

int taskId = scanner.nextInt();

scanner.nextLine();

PreparedStatement selectStatement = connection.prepareStatement("SELECT \* FROM tasks WHERE id = ?");

selectStatement.setInt(1, taskId);

ResultSet resultSet = selectStatement.executeQuery();

if (resultSet.next()) {

System.out.println("-------------------------------------------------------");

System.out.println("Task ID: " + resultSet.getInt("id"));

System.out.println("Name: " + resultSet.getString("name"));

System.out.println("Description: " + resultSet.getString("description"));

System.out.println("Priority: " + resultSet.getInt("priority"));

System.out.println("Due Date: " + resultSet.getDate("due\_date"));

System.out.println("Assigned To: " + resultSet.getString("assigned\_to"));

System.out.println("Status: " + resultSet.getString("status"));

System.out.println("Progress: " + resultSet.getInt("progress") + "%");

System.out.println("-------------------------------------------------------");

} else {

System.out.println("========================================================\n");

System.out.println("\tTask with ID " + taskId + " not found.");

System.out.println("========================================================");

System.out.println("\n\n\n");

}

}

private static void updateTask(Connection connection, Scanner scanner) throws SQLException {

System.out.println("\n\n\t\tUpdate Task");

System.out.println("---------------------------------------------\n");

System.out.print("Enter task ID to update: ");

int taskId = scanner.nextInt();

scanner.nextLine();

PreparedStatement selectStatement = connection.prepareStatement("SELECT \* FROM tasks WHERE id = ?");

selectStatement.setInt(1, taskId);

ResultSet resultSet = selectStatement.executeQuery();

if (resultSet.next()) {

System.out.println("Current Task Details:");

System.out.println("Name: " + resultSet.getString("name"));

System.out.println("Description: " + resultSet.getString("description"));

System.out.println("Priority: " + resultSet.getInt("priority"));

System.out.println("Due Date: " + resultSet.getDate("due\_date"));

System.out.println("Assigned To: " + resultSet.getString("assigned\_to"));

System.out.println("Status: " + resultSet.getString("status"));

System.out.println("Progress: " + resultSet.getInt("progress") + "%");

System.out.println("Enter new task details (excluding progress):");

System.out.print("Enter task name: ");

String name = scanner.nextLine();

System.out.print("Enter task description: ");

String description = scanner.nextLine();

System.out.print("Enter task priority: ");

int priority = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter due date (YYYY-MM-DD): ");

String dueDate = scanner.next();

System.out.print("Enter assigned to: ");

String assignedTo = scanner.next();

System.out.print("Enter task status: ");

String status = scanner.next();

PreparedStatement updateStatement = connection.prepareStatement(

"UPDATE tasks SET name = ?, description = ?, priority = ?, due\_date = ?, assigned\_to = ?, status = ? WHERE id = ?");

updateStatement.setString(1, name);

updateStatement.setString(2, description);

updateStatement.setInt(3, priority);

updateStatement.setString(4, dueDate);

updateStatement.setString(5, assignedTo);

updateStatement.setString(6, status);

updateStatement.setInt(7, taskId);

updateStatement.executeUpdate();

System.out.println("========================================================\n");

System.out.println("\tTask details updated successfully.");

System.out.println("========================================================");

System.out.println("\n\n\n");

} else {

System.out.println("========================================================\n");

System.out.println("\tTask with ID " + taskId + " not found.");

System.out.println("========================================================");

System.out.println("\n\n\n");

}

}

private static void updateTaskProgress(Connection connection, Scanner scanner) throws SQLException {

System.out.println("\n\n\t\tUpdate Task Progress");

System.out.println("---------------------------------------------\n");

System.out.print("Enter task ID to update progress: ");

int taskId = scanner.nextInt();

scanner.nextLine();

PreparedStatement selectStatement = connection.prepareStatement("SELECT progress FROM tasks WHERE id = ?");

selectStatement.setInt(1, taskId);

ResultSet resultSet = selectStatement.executeQuery();

if (resultSet.next()) {

int currentProgress = resultSet.getInt("progress");

System.out.println("Current Progress: " + currentProgress + "%");

int newProgress;

do {

System.out.print("Enter new progress percentage (0-100): ");

newProgress = scanner.nextInt();

scanner.nextLine();

if (newProgress < 0 || newProgress > 100) {

System.out.println("Please enter a valid progress percentage (0-100).");

}

} while (newProgress < 0 || newProgress > 100);

PreparedStatement updateStatement = connection.prepareStatement(

"UPDATE tasks SET progress = ? WHERE id = ?");

updateStatement.setInt(1, newProgress);

updateStatement.setInt(2, taskId);

updateStatement.executeUpdate();

System.out.println("========================================================\n");

System.out.println("\tProgress updated successfully.");

System.out.println("========================================================");

System.out.println("\n\n\n");

} else {

System.out.println("========================================================\n");

System.out.println("\tTask with ID " + taskId + " not found.");

System.out.println("========================================================");

System.out.println("\n\n\n");

}

}

private static void deleteTask(Connection connection, Scanner scanner) throws SQLException {

System.out.println("\n\n\t\tDelete Task");

System.out.println("---------------------------------------------\n");

System.out.print("Enter task ID to delete: ");

int taskId = scanner.nextInt();

scanner.nextLine();

PreparedStatement deleteStatement = connection.prepareStatement("DELETE FROM tasks WHERE id = ?");

deleteStatement.setInt(1, taskId);

int rowsAffected = deleteStatement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("========================================================\n");

System.out.println("\tTask with ID " + taskId + " deleted successfully.");

System.out.println("========================================================");

System.out.println("\n\n\n");

} else {

System.out.println("========================================================\n");

System.out.println("\tTask with ID " + taskId + " not found.");

System.out.println("========================================================");

System.out.println("\n\n\n");

}

}

}

**Command :-**

javac termwork.java

java -cp C:\Users\nikhi\OneDrive\Desktop\Termwork\mysql-connector-j-8.3.0.jar termwork.java

**A screenshot of a computer

Description automatically generatedOUTPUT :-**

**Add Task :-**

A screenshot of a computer program

Description automatically generated

**View All Task :-**

A screenshot of a computer program

Description automatically generated

**View Task By Id :-**

**A screenshot of a computer program

Description automatically generated**

**Update Task :-**

**A screenshot of a computer program

Description automatically generated**

**Update Task Process :-**

A screenshot of a computer program

Description automatically generated

**Delete Task :-**

A screenshot of a computer program

Description automatically generated

**Exit** :-

**A screenshot of a computer

Description automatically generated**