package propertymangement;

import java.sql.\*;

import java.util.Scanner;

public class PropertyMangement {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/RentalProperties";

private static final String DB\_USER = "root";

private static final String DB\_PASSWORD = "password";

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

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

String name = scanner.nextLine();

System.out.print("Enter a username: ");

String username = scanner.nextLine();

System.out.print("Enter a password: ");

String password = scanner.nextLine();

String sql = "INSERT INTO users (name, username, password, role) VALUES (?, ?, ?, 'manager')";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setString(1, name);

statement.setString(2, username);

statement.setString(3, password);

statement.executeUpdate();

System.out.println("Manager registered successfully.");

}

}

public static void login(Connection connection, Scanner scanner) throws SQLException {

System.out.print("Enter username: ");

String username = scanner.nextLine();

System.out.print("Enter password: ");

String password = scanner.nextLine();

String sql = "SELECT role FROM users WHERE username = ? AND password = ?";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setString(1, username);

statement.setString(2, password);

ResultSet resultSet = statement.executeQuery();

if (resultSet.next()) {

String role = resultSet.getString("role");

if (null == role) {

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

} else switch (role) {

case "manager" -> managerMenu(connection, scanner);

case "tenant" -> tenantMenu(connection, scanner, username);

default -> System.out.println("Unknown role. Exiting.");

}

} else {

System.out.println("Invalid username or password.");

}

}

}

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

while (true) {

System.out.println("\nManager Menu:");

System.out.println("1. View Properties\n2. View Tenants\n3. View Payments\n4. View Lease Agreements\n5. Approve/Decline Tenant Applications\n6. Evict Tenant\n7. Logout");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

scanner.nextLine();

switch (choice) {

case 1 -> viewProperties(connection);

case 2 -> viewTenants(connection);

case 3 -> viewPayments(connection);

case 4 -> viewLeaseAgreements(connection);

case 5 -> approveOrDeclineTenants(connection, scanner);

case 6 -> evictTenant(connection, scanner);

case 7 -> {

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

return;

}

default -> System.out.println("Invalid choice. Try again.");

}

}

}

private static void tenantMenu(Connection connection, Scanner scanner, String username) throws SQLException {

while (true) {

System.out.println("\nTenant Menu:");

System.out.println("1. View Available Properties\n2. View Payments\n3. Request Maintenance\n4. Send Vacation Notice\n5. Logout");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1 -> viewProperties(connection);

case 2 -> viewTenantPayments(connection, username);

case 3 -> requestMaintenance(connection, scanner, username);

case 4 -> sendVacationNotice(connection, scanner, username);

case 5 -> {

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

return;

}

default -> System.out.println("Invalid choice. Try again.");

}

}

}

private static void viewProperties(Connection connection) throws SQLException {

String sql = "SELECT \* FROM properties";

try (Statement statement = connection.createStatement(); ResultSet resultSet = statement.executeQuery(sql)) {

System.out.println("\nAvailable Properties:");

while (resultSet.next()) {

System.out.printf("ID: %d, Address: %s, Rent: %.2f\n",

resultSet.getInt("id"),

resultSet.getString("address"),

resultSet.getDouble("rent"));

}

}

}

private static void viewTenants(Connection connection) throws SQLException {

String sql = "SELECT \* FROM tenants";

try (Statement statement = connection.createStatement(); ResultSet resultSet = statement.executeQuery(sql)) {

System.out.println("\nTenants:");

while (resultSet.next()) {

System.out.printf("ID: %d, Name: %s, Username: %s\n",

resultSet.getInt("id"),

resultSet.getString("name"),

resultSet.getString("username"));

}

}

}

private static void viewPayments(Connection connection) throws SQLException {

String sql = "SELECT \* FROM payments";

try (Statement statement = connection.createStatement(); ResultSet resultSet = statement.executeQuery(sql)) {

System.out.println("\nPayments:");

while (resultSet.next()) {

System.out.printf("Payment ID: %d, Tenant ID: %d, Amount: %.2f, Date: %s\n",

resultSet.getInt("id"),

resultSet.getInt("tenant\_id"),

resultSet.getDouble("amount"),

resultSet.getDate("date"));

}

}

}

private static void viewLeaseAgreements(Connection connection) throws SQLException {

String sql = "SELECT \* FROM lease\_agreements";

try (Statement statement = connection.createStatement(); ResultSet resultSet = statement.executeQuery(sql)) {

System.out.println("\nLease Agreements:");

while (resultSet.next()) {

System.out.printf("Lease ID: %d, Tenant ID: %d, Property ID: %d, Start Date: %s, End Date: %s\n",

resultSet.getInt("id"),

resultSet.getInt("tenant\_id"),

resultSet.getInt("property\_id"),

resultSet.getDate("start\_date"),

resultSet.getDate("end\_date"));

}

}

}

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

String sql = "SELECT \* FROM pending\_approvals";

try (Statement statement = connection.createStatement(); ResultSet resultSet = statement.executeQuery(sql)) {

System.out.println("\nPending Tenant Applications:");

while (resultSet.next()) {

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

String username = resultSet.getString("username");

String password = resultSet.getString("password");

System.out.printf("ID: %d, Name: %s, Username: %s\n", id, name, username);

System.out.print("Approve this application? (yes/no): ");

String decision = scanner.nextLine();

if ("yes".equalsIgnoreCase(decision)) {

String approveSql = "INSERT INTO tenants (name, username, password) VALUES (?, ?, ?)";

try (PreparedStatement approveStatement = connection.prepareStatement(approveSql)) {

approveStatement.setString(1, name);

approveStatement.setString(2, username);

approveStatement.setString(3, password);

approveStatement.executeUpdate();

}

String deleteSql = "DELETE FROM pending\_approvals WHERE id = ?";

try (PreparedStatement deleteStatement = connection.prepareStatement(deleteSql)) {

deleteStatement.setInt(1, id);

deleteStatement.executeUpdate();

}

System.out.println("Application approved and tenant added.");

} else {

String deleteSql = "DELETE FROM pending\_approvals WHERE id = ?";

try (PreparedStatement deleteStatement = connection.prepareStatement(deleteSql)) {

deleteStatement.setInt(1, id);

deleteStatement.executeUpdate();

}

System.out.println("Application declined and removed.");

}

}

}

}

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

System.out.print("Enter Tenant ID to evict: ");

int tenantId = scanner.nextInt();

scanner.nextLine(); // Consume newline

String sql = "DELETE FROM tenants WHERE id = ?";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setInt(1, tenantId);

int rowsAffected = statement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Tenant evicted successfully.");

} else {

System.out.println("Tenant ID not found.");

}

}

}

private static void viewTenantPayments(Connection connection, String username) throws SQLException {

String sql = "SELECT \* FROM payments WHERE tenant\_id = (SELECT id FROM tenants WHERE username = ?)";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setString(1, username);

try (ResultSet resultSet = statement.executeQuery()) {

System.out.println("\nYour Payments:");

while (resultSet.next()) {

System.out.printf("Payment ID: %d, Amount: %.2f, Date: %s\n",

resultSet.getInt("id"),

resultSet.getDouble("amount"),

resultSet.getDate("date"));

}

}

}

}

private static void requestMaintenance(Connection connection, Scanner scanner, String username) throws SQLException {

System.out.print("Enter your maintenance request: ");

String request = scanner.nextLine();

String sql = "INSERT INTO maintenance\_requests (tenant\_id, request) VALUES ((SELECT id FROM tenants WHERE username = ?), ?)";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setString(1, username);

statement.setString(2, request);

statement.executeUpdate();

System.out.println("Maintenance request submitted.");

}

}

private static void sendVacationNotice(Connection connection, Scanner scanner, String username) throws SQLException {

System.out.print("Enter your vacation notice message: ");

String notice = scanner.nextLine();

String sql = "INSERT INTO vacation\_notices (tenant\_id, notice) VALUES ((SELECT id FROM tenants WHERE username = ?), ?)";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setString(1, username);

statement.setString(2, notice);

statement.executeUpdate();

System.out.println("Vacation notice submitted.");

}

}

public static void main(String[] args) {

try (Connection connection = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

Scanner scanner = new Scanner(System.in)) {

System.out.println("Welcome to Rental Properties Management System");

System.out.println("1. Register as Manager\n2. Login\nChoose an option: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (choice != 1) if (choice == 2) {

login(connection, scanner);

} else {

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

} else {

registerManager(connection, scanner);

}

} catch (SQLException e) {

}

}

}