**8. a) ROLE BASED ACCESS CONTROL SYSTEM**

**Code :**

package Experiment8;

import java.util.\*;

class User

{

public String username;

User(String name)

{

username = name;

}

public void accessSystem()

{

System.out.println("Access Denied: Insufficient permission");

}

}

class Admin extends User

{

Admin(String name)

{

super(name);

}

@Override

public void accessSystem()

{

System.out.println(username+" : Admin access -> Access to everything");

}

}

class Manager extends User

{

Manager(String name)

{

super(name);

}

@Override

public void accessSystem()

{

System.out.println(username+" : Manager access -> Access to management resources granted");

}

}

class Employee extends User

{

Employee(String name)

{

super(name);

}

@Override

public void accessSystem()

{

System.out.println(username+" : Employee access-> Access to general resources granted");

}

}

public class RoleBasedAccessControl {

public static void main(String[] args) {

String role;

Scanner sc = new Scanner(System.in);

String name;

User u;

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

name = sc.nextLine();

System.out.println("Please select the role from the following : ");

System.out.println("1. Admin\n2. Manager\n3. Employee\n4. Other\n");

System.out.print("Provide your role : ");

role = sc.nextLine();

if(role.equalsIgnoreCase("admin"))

u = new Admin(name);

else if(role.equalsIgnoreCase("manager"))

u = new Manager(name);

else if(role.equalsIgnoreCase("employee"))

u = new Employee(name);

else

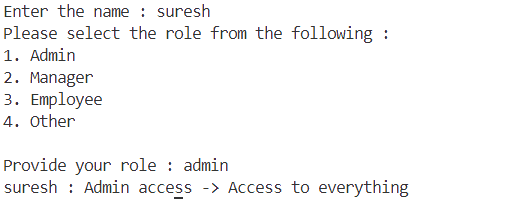
u = new User(name);

u.accessSystem();

}

}

**Output :**

****

**8. b) MULTI-FACTOR AUTHENTICATION SYSTEM**

**Code :**

import java.util.Random;

import java.util.Scanner;

abstract class UserAuthentication {

abstract boolean authenticate();

}

class PasswordLogin extends UserAuthentication {

private final String storedUsername;

private final String storedPassword;

public PasswordLogin(String username, String password) {

this.storedUsername = username;

this.storedPassword = password;

System.out.println("User registered with Username and Password.");

}

@Override

public boolean authenticate() {

Scanner sc = new Scanner(System.in);

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

String username = sc.nextLine();

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

String password = sc.nextLine();

if (storedUsername.equals(username) && storedPassword.equals(password)) {

System.out.println(" Login Successful: Valid Username & Password");

return true;

} else {

System.out.println(" Login Failed: Invalid Username or Password");

return false;

}

}

}

class OTPLogin extends UserAuthentication {

private final String email;

private final String otp;

public OTPLogin(String email) {

this.email = email;

this.otp = generateOTP();

System.out.println("User registered with Email: " + this.email);

System.out.println(" OTP sent: " + this.otp );

}

private String generateOTP() {

Random rand = new Random();

return String.format("%06d", rand.nextInt(1000000));

}

@Override

public boolean authenticate() {

Scanner sc = new Scanner(System.in);

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

String enteredOTP = sc.nextLine();

if (this.otp.equals(enteredOTP)) {

System.out.println(" Login Successful: Valid Email & OTP");

return true;

} else {

System.out.println(" Login Failed: Invalid OTP");

return false;

}

}

}

class BiometricLogin extends UserAuthentication {

private final boolean isBiometricEnabled;

public BiometricLogin(boolean isBiometricEnabled) {

this.isBiometricEnabled = isBiometricEnabled;

System.out.println("User registered with Biometric Authentication.");

}

public boolean authenticate() {

if (isBiometricEnabled) {

System.out.println(" Login Successful: Biometric Authentication Passed");

return true;

} else {

System.out.println(" Login Failed: Biometric Authentication Failed");

return false;

}

}

}

public class MultiFactorAuthenticationSystem {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int choice;

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

System.out.println("1. Login using Username & Password");

System.out.println("2. Login using Email & OTP");

System.out.println("3. Login using Biometric");

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

choice = sc.nextInt();

sc.nextLine();

UserAuthentication authMethod;

switch (choice) {

case 1:

authMethod = new PasswordLogin("solai@123", "12345");

authMethod.authenticate();

break;

case 2:

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

String email = sc.nextLine();

authMethod = new OTPLogin(email);

authMethod.authenticate();

break;

case 3:

authMethod = new BiometricLogin(true);

authMethod.authenticate();

break;

default:

System.out.println(" Invalid choice! Please select a valid authentication method.");

break;

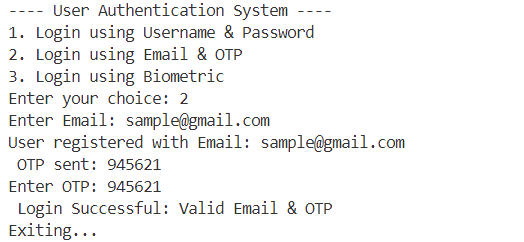
}

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

}

}

**Output :**

****