Міністерство освіти та науки України

Черкаський політехнічний фаховий коледж

ЗВІТ

з виконання лаборадорної роботи

Виконав

студент групи ПС 4-1

Карпенко Іван

Перевірив

викладач

Сметанін О. Г.

м.Черкаси, 2024

**Тема:** «Роботи з серверами БД засобами java»

**Завдання**

До розроблених в лабораторній роботі No1 форм fxml графічних інтерфейсів створити:

1) авторизаційні та реєстраційні форми. Таким чином, в разі успішної авторизації, користувач бачить форму із зображенням, що запрошує його скористатися додатком, інакше – виводиться графічне xml-вікно з попередженням про неавтентичність даного користувача.

2) В якості графічних ресурсів для xml-форм використати графічні ресурси проекту.

3) Розробити архітектури відповідної БД в MySQL.

4) Для роботи з базами даних в Java вкиористати переносних незалежний промисловий стандарт взаємодії Java-додатків з різними СУБД (JDBC).

**Лістинг Register.fxml**

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.PasswordField?>

<?import javafx.scene.control.RadioButton?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.control.ToggleGroup?>

<?import javafx.scene.layout.HBox?>

<?import javafx.scene.layout.VBox?>

<VBox alignment="CENTER" prefHeight="410.0" prefWidth="333.0" spacing="15.0" style="-fx-padding: 20; -fx-background-color: #f4f4f4;" xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.karpenko.RegisterController">

   <children>

      <HBox alignment="CENTER" prefHeight="100.0" prefWidth="200.0">

         <children>

            <!-- Header Label -->

            <Label style="-fx-font-size: 18px; -fx-font-weight: bold;" text="Registration Form">

               <HBox.margin>

                  <Insets right="25.0" />

               </HBox.margin>

            </Label>

            <Button fx:id="registerButton1" onAction="#switchToSecondary" style="-fx-font-size: 14px; -fx-background-color: #007bff; -fx-text-fill: white; -fx-background-radius: 5;" text="Login" textAlignment="CENTER" />

         </children>

      </HBox>

      <!-- Name Field -->

      <TextField fx:id="nameField" promptText="Enter your first name" style="-fx-font-size: 14px; -fx-background-radius: 5;" />

      <!-- Last Name Field -->

      <TextField fx:id="lastNameField" promptText="Enter your last name" style="-fx-font-size: 14px; -fx-background-radius: 5;" />

      <!-- City Field -->

      <TextField fx:id="cityField" promptText="Enter your city" style="-fx-font-size: 14px; -fx-background-radius: 5;" />

      <!-- Login Field -->

      <TextField fx:id="loginField" promptText="Choose a login" style="-fx-font-size: 14px; -fx-background-radius: 5;" />

      <!-- Password Field -->

      <PasswordField fx:id="passwordField" promptText="Choose a password" style="-fx-font-size: 14px; -fx-background-radius: 5;" />

      <RadioButton mnemonicParsing="false" text="Male">

         <toggleGroup>

            <ToggleGroup fx:id="gender" />

         </toggleGroup>

      </RadioButton>

      <RadioButton mnemonicParsing="false" text="Female" toggleGroup="$gender" />

      <RadioButton mnemonicParsing="false" text="Helicopter" toggleGroup="$gender" />

      <!-- Register Button -->

      <Button fx:id="registerButton" onAction="#handleRegister" style="-fx-font-size: 14px; -fx-background-color: #007bff; -fx-text-fill: white; -fx-background-radius: 5;" text="Register" />

   </children>

</VBox>

**Лістинг Login.fxml**

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.PasswordField?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.HBox?>

<?import javafx.scene.layout.VBox?>

<VBox alignment="TOP\_CENTER" spacing="15.0" style="-fx-padding: 20; -fx-background-color: #f4f4f4;" xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.karpenko.LoginController">

    <children>

      <HBox alignment="CENTER" prefHeight="50.0" prefWidth="200.0" style="-fx-background-color: #333;">

         <children>

                <Label style="-fx-text-fill: white; -fx-font-size: 16px; -fx-font-weight: bold;" text="Authorization" />

            <Button fx:id="RegisterButt" alignment="CENTER\_RIGHT" onAction="#switchToPrimary" style="-fx-font-size: 10px; -fx-background-color: #007bff; -fx-text-fill: white; -fx-background-radius: 5;" text="Register">

               <HBox.margin>

                  <Insets left="25.0" />

               </HBox.margin>

            </Button>

         </children>

      </HBox>

        <!-- Login Field -->

        <TextField fx:id="LoginText" promptText="Enter Login" style="-fx-font-size: 14px; -fx-background-radius: 5;">

            <VBox.margin>

                <Insets left="20.0" right="20.0" />

            </VBox.margin>

        </TextField>

        <!-- Password Field -->

        <PasswordField fx:id="PasswordText" promptText="Enter Password" style="-fx-font-size: 14px; -fx-background-radius: 5;">

            <VBox.margin>

                <Insets left="20.0" right="20.0" />

            </VBox.margin>

        </PasswordField>

        <!-- Login Button -->

        <Button fx:id="LogimButt" onAction="#Login" style="-fx-font-size: 14px; -fx-background-color: #007bff; -fx-text-fill: white; -fx-background-radius: 5;" text="Login">

            <VBox.margin>

                <Insets bottom="20.0" top="10.0" />

            </VBox.margin>

        </Button>

    </children>

</VBox>

**Лістинг RegisterController.java**

package com.karpenko;

import java.io.IOException;

import java.nio.charset.StandardCharsets;

import java.security.MessageDigest;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.xml.bind.DatatypeConverter;

import javafx.fxml.FXML;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Button;

import javafx.scene.control.ButtonType;

import javafx.scene.control.PasswordField;

import javafx.scene.control.RadioButton;

import javafx.scene.control.TextField;

import javafx.scene.control.ToggleGroup;

public class RegisterController {

    private final String url = "jdbc:sqlite:db.db";

    @FXML

    private TextField nameField;

    @FXML

    private TextField lastNameField;

    @FXML

    private TextField cityField;

    @FXML

    private TextField loginField;

    @FXML

    private PasswordField passwordField;

    @FXML

    private Button registerButton;

    @FXML

    private ToggleGroup gender;

    @FXML

    private void switchToSecondary() throws IOException {

        App.setRoot("Login");

    }

    @FXML

    private void handleRegister() throws IOException {

        String error = "";

        if (nameField.getText().isEmpty()) {

            System.err.println("sdsad");

            error += "name, ";

        }

        if (lastNameField.getText().isEmpty()) {

            System.err.println("sdsad");

            error += "lastName, ";

        }

        if (cityField.getText().isEmpty()) {

            System.err.println("sdsad");

            error += "city, ";

        }

        if (loginField.getText().isEmpty()) {

            System.err.println("sdsad");

            error += "login, ";

        }

        if (passwordField.getText().isEmpty()) {

            System.err.println("sdsad");

        }

        RadioButton butt = (RadioButton) gender.getSelectedToggle();

        if (butt == null) {

            error += "gender, ";

        }

        if (error != "") {

            Alert alert = new Alert(AlertType.ERROR, "erors in folowing sections: " + error, ButtonType.OK);

            alert.show();

            return;

        }

        try (Connection connection = DriverManager.getConnection(url);

                PreparedStatement statement = connection.prepareStatement(

                        "SELECT  COUNT(\*) FROM Users WHERE login = ?; ");) {

            statement.setString(1, loginField.getText());

            ResultSet result = statement.executeQuery();

            result.next();

            if (result.getInt(1) != 0) {

                Alert alert = new Alert(AlertType.ERROR, "Login taken", ButtonType.OK);

                alert.show();

                return;

            }

        } catch (Exception e) {

            e.printStackTrace();

        }

        try (Connection connection = DriverManager.getConnection(url);

                PreparedStatement statement = connection.prepareStatement(

                        "SELECT  login FROM Users WHERE password = ?; ");) {

            MessageDigest digest = MessageDigest.getInstance("SHA-256");

            byte[] hash = digest.digest(passwordField.getText().getBytes(StandardCharsets.UTF\_8));

            String hex = DatatypeConverter.printHexBinary(hash);

            System.out.println(hex);

            statement.setString(1, hex);

            ResultSet result = statement.executeQuery();

            if (result.isBeforeFirst()) {

                result.next();

                Alert alert = new Alert(AlertType.ERROR,

                        "that pasword is taken by login \"" + result.getString("login") + "\"", ButtonType.OK);

                alert.show();

                return;

            }

        } catch (Exception e) {

            e.printStackTrace();

        }

        try (Connection connection = DriverManager.getConnection(url);

                PreparedStatement statement = connection.prepareStatement(

                        "INSERT INTO Users ( name, lastName, city, login, password, genda) VALUES(?,?,?,?,?,?);");) {

            statement.setString(1, nameField.getText());

            statement.setString(2, lastNameField.getText());

            statement.setString(3, cityField.getText());

            statement.setString(4, loginField.getText());

            MessageDigest digest = MessageDigest.getInstance("SHA-256");

            byte[] hash = digest.digest(passwordField.getText().getBytes(StandardCharsets.UTF\_8));

            String hex = DatatypeConverter.printHexBinary(hash);

            System.out.println(hex);

            statement.setString(5, hex);

            String gender = butt.getText().toString();

            statement.setString(6, gender);

            statement.executeUpdate();

            System.err.println("created");

            Alert alert = new Alert(AlertType.INFORMATION, "Created" + error, ButtonType.OK);

            alert.show();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

**Лістинг LoginController.java**

package com.karpenko;

import java.io.IOException;

import java.nio.charset.StandardCharsets;

import java.security.MessageDigest;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.xml.bind.DatatypeConverter;

import javafx.fxml.FXML;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.Button;

import javafx.scene.control.ButtonType;

import javafx.scene.control.PasswordField;

import javafx.scene.control.TextField;

public class LoginController {

    private final String url = "jdbc:sqlite:db.db";

    @FXML private TextField LoginText;

    @FXML private PasswordField PasswordText;

    @FXML private Button LogimButt;

    @FXML

    private void switchToPrimary() throws IOException {

        App.setRoot("Register");

    }

    @FXML

    private void Login() throws IOException {

        String error = "";

        if (LoginText.getText().isEmpty()) {

            System.err.println("sdsad");

            error += "Login, ";

        }

        if (PasswordText.getText().isEmpty()) {

            System.err.println("sdsad");

            error += "Password, ";

        }

         if (error != "") {

            Alert alert = new Alert(AlertType.ERROR, "erors in folowing sections: " + error, ButtonType.OK);

            alert.show();

            return;

        }

        try (Connection connection = DriverManager.getConnection(url);

                PreparedStatement statement = connection.prepareStatement(

                        "SELECT  name, lastName FROM Users WHERE login = ? AND password = ?;");) {

            MessageDigest digest = MessageDigest.getInstance("SHA-256");

            byte[] hash = digest.digest(PasswordText.getText().getBytes(StandardCharsets.UTF\_8));

            String hex = DatatypeConverter.printHexBinary(hash);

            System.out.println(hex);

            statement.setString(1, LoginText.getText().toString());

            statement.setString(2, hex);

            ResultSet result = statement.executeQuery();

            if(!result.isBeforeFirst()){

                Alert alert = new Alert(AlertType.ERROR, "THERE NO SHUCH MAN", ButtonType.OK);

                alert.show();

                return;

            }

            result.next();

            Alert alert = new Alert(AlertType.INFORMATION, "Whelcome " + result.getString("name")+" " + result.getString("lastName"), ButtonType.OK);

            alert.show();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

**Лістинг App.java**

package com.karpenko;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.stage.Stage;

import java.io.IOException;

/\*\*

 \* JavaFX App

 \*/

public class App extends Application {

    private static Scene scene;

    @Override

    public void start(Stage stage) throws IOException {

        scene = new Scene(loadFXML("Login"), 640, 480);

        stage.setScene(scene);

        stage.show();

    }

    static void setRoot(String fxml) throws IOException {

        scene.setRoot(loadFXML(fxml));

    }

    private static Parent loadFXML(String fxml) throws IOException {

        FXMLLoader fxmlLoader = new FXMLLoader(App.class.getResource(fxml + ".fxml"));

        return fxmlLoader.load();

    }

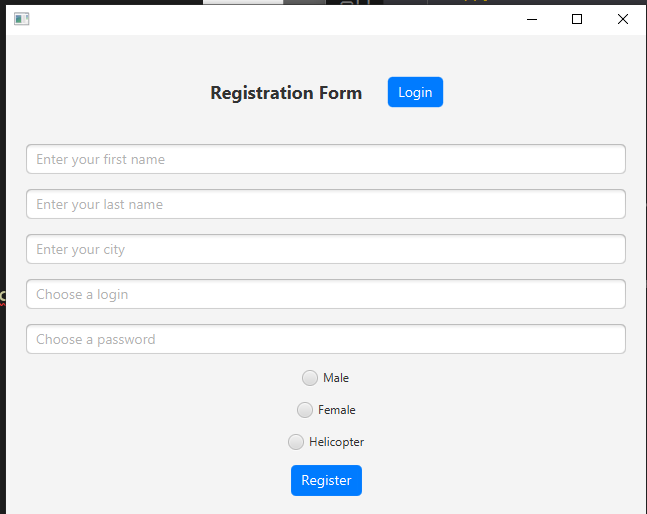
    public static void main(String[] args) {

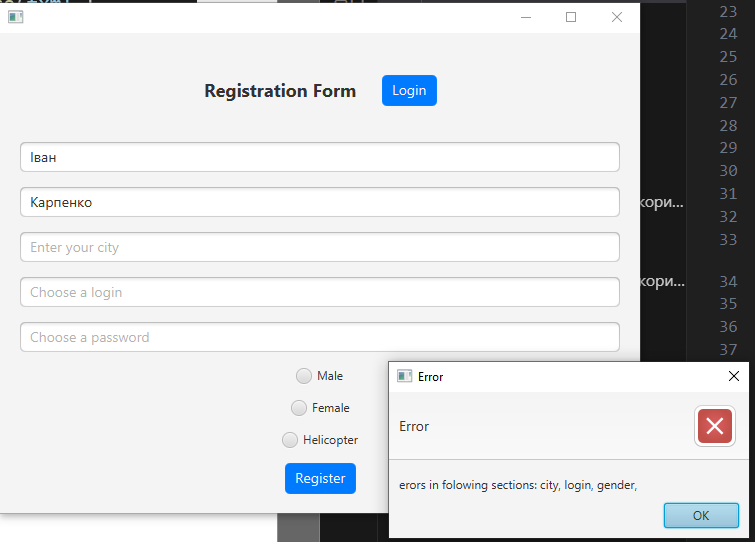
        launch();

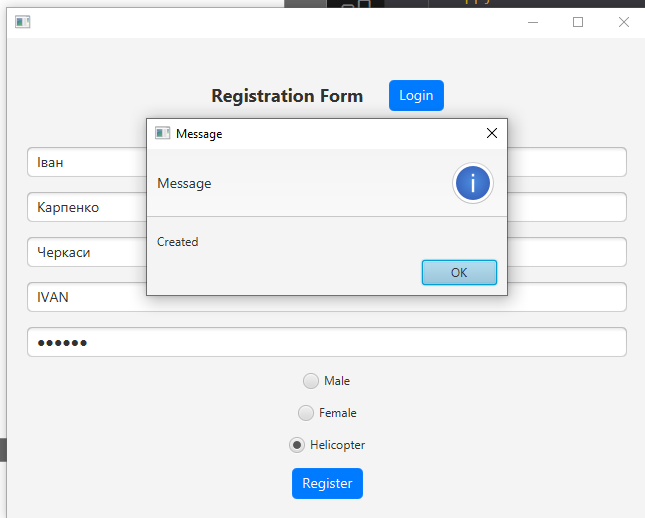
    }

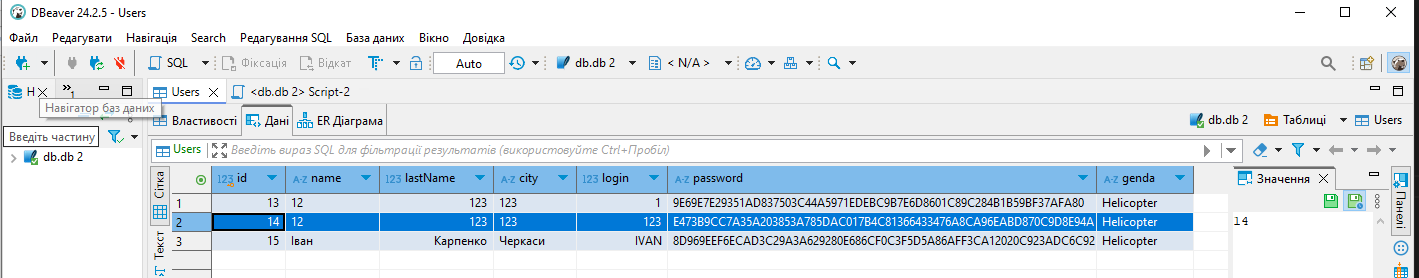
}

**Результат роботи**

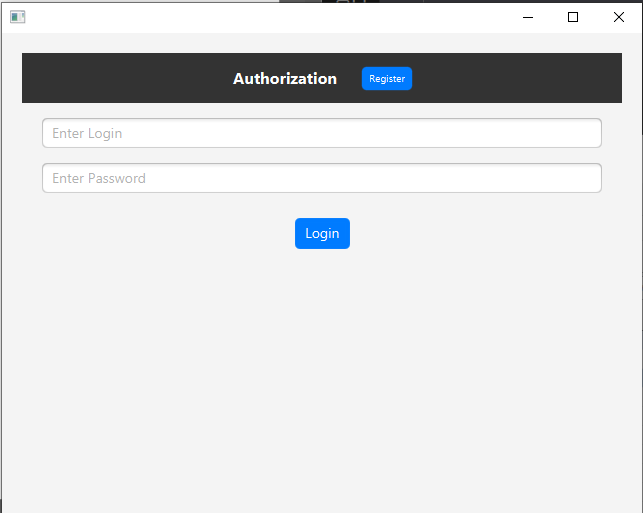
****

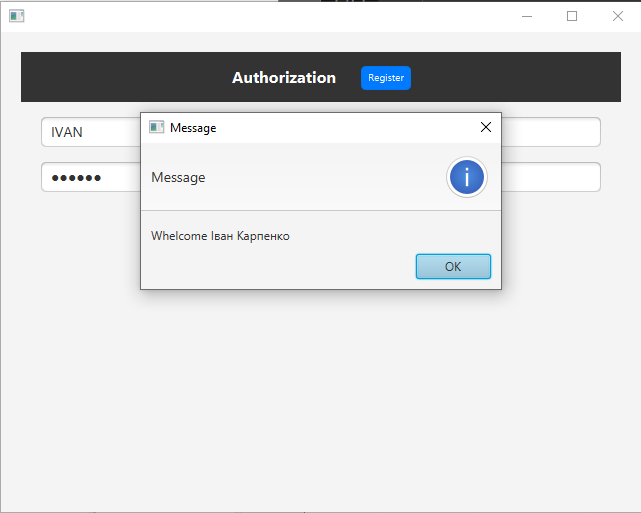
****

****

****

Паролі зберігаються в зашифрованому вигляді

****

****