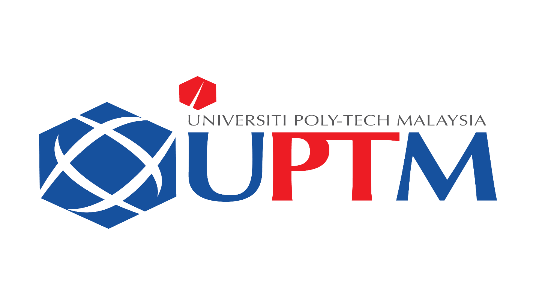
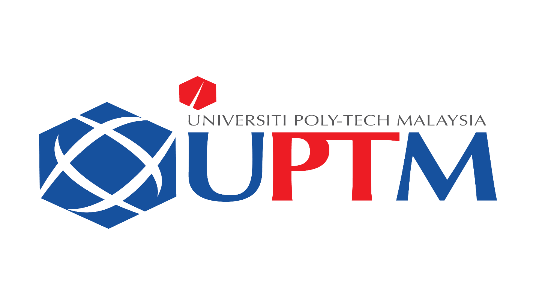
|  |  |  |  |
| --- | --- | --- | --- |
| **Name:**  NURULAIN NAJWA BINTI HAMDAN | | **Section : 01** | |
| **ID Number:**  AM2211012866 | |
| **Lecturer:**  WAN NOR ASNIDA BTE WAN JUSOH | | **Lab group / Tutorial group / Tutor (if applicable):** |
| **Course and Course Code:**  CT204 SWC4243 | | **Submission Date:** |
| **Assignment No. / Title:**  LABTASK | | **Extension & Late submission:**  **Disallowed** |
| **Assignment Type:** | **% of Assignment Mark** | **Returning Date:** | |
| **Penalties:**   1. 10% of the original mark will be deducted for every one week period after the submission date. 2. No work will be accepted after two weeks of the deadline. 3. If you were unable to submit the coursework on time due to extenuating circumstances you may be eligible for an extension. 4. Extension will not exceed one week. | | |
| **Declaration**: I/we the undersigned confirm that I/we have read and agree to abide by these regulations on plagiarism and cheating. I/we confirm that this piece of work is my/our own. I/we consent to appropriate storage of our work for checking to ensure that there is no plagiarism/ academic cheating.  **Signature(s): Nurulain Najwa**  **Full Name(s):**  NURULAIN NAJWA BINTI HAMDAN | | |
| **This section may be used for feedback or other information** | | |





**Table of Contents**

[**FXMLDocument.fxml** 3](#_Toc182434054)

[**Student.java** 8](#_Toc182434055)

[**FXMLDocumentController.java** 9](#_Toc182434056)

[**DatabaseConnection.java** 16](#_Toc182434057)

[**AinLabTask.Java** 22](#_Toc182434058)

[**Output** 24](#_Toc182434059)

# **FXMLDocument.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.TableColumn?>

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

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

<?import javafx.scene.image.Image?>

<?import javafx.scene.image.ImageView?>

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

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

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

<?import javafx.scene.text.Font?>

<AnchorPane id="AnchorPane" prefHeight="565.0" prefWidth="835.0" style="-fx-background-color: FEF3E2;" xmlns:fx="http://javafx.com/fxml/1" xmlns="http://javafx.com/javafx/23.0.1" fx:controller="ainlabtask.FXMLDocumentController">

<children>

<HBox layoutX="305.0" layoutY="14.0" prefHeight="100.0" prefWidth="476.0">

<children>

<Label alignment="CENTER" contentDisplay="CENTER" prefHeight="100.0" prefWidth="478.0" text="Genius Kindergarten" textFill="#ff8b65">

<font>

<Font name="Rockwell Condensed" size="55.0" />

</font>

</Label>

</children>

</HBox>

<HBox layoutX="240.0" layoutY="114.0" prefHeight="317.0" prefWidth="621.0">

<children>

<TableView fx:id="tvStudent" prefHeight="317.0" prefWidth="808.0">

<columns>

<TableColumn fx:id="colID" prefWidth="47.0" text="ID" />

<TableColumn fx:id="colName" prefWidth="133.0" text="Name" />

<TableColumn fx:id="colBirth" prefWidth="81.0" text="Birth Date" />

<TableColumn fx:id="colAge" prefWidth="58.0" text="Age" />

<TableColumn fx:id="colAddress" prefWidth="194.0" text="Address" />

<TableColumn fx:id="colAllergic" prefWidth="107.0" text="Allergies" />

</columns>

</TableView>

</children>

</HBox>

<VBox alignment="CENTER" layoutX="42.0" layoutY="448.0" prefHeight="100.0" prefWidth="79.0">

<children>

<Button fx:id="btnAdd" alignment="CENTER" mnemonicParsing="false" onAction="#handleButtonAction" style="css: #9FE2BF; -fx-text-fill: White; -fx-font-size: 14px; -fx-background-color: FC8F54; -fx-border-width: 3px; -fx-background-radius: 5px;" text="Add">

<VBox.margin>

<Insets bottom="10.0" />

</VBox.margin>

</Button>

<Button fx:id="btnUpdate" mnemonicParsing="false" onAction="#handleButtonAction" style="css: -fx-background-color: #ff6f61; -fx-text-fill: white; -fx-font-size: 14px; -fx-background-color: FC8F54;" text="Update" />

</children>

</VBox>

<VBox alignment="CENTER" layoutX="121.0" layoutY="448.0" prefHeight="100.0" prefWidth="79.0">

<children>

<Button fx:id="btnDelete" alignment="CENTER" mnemonicParsing="false" onAction="#handleButtonAction" style="css: #9FE2BF; -fx-text-fill: white; -fx-font-size: 14px; -fx-background-color: FC8F54; -fx-border-width: 3px;" text="Delete">

<VBox.margin>

<Insets bottom="10.0" />

</VBox.margin>

</Button>

<Button fx:id="btnReset" mnemonicParsing="false" onAction="#handleButtonAction" style="css: -fx-background-color: #8B5DFF; -fx-text-fill: white; -fx-font-size: 14px; -fx-background-color: FC8F54;" text="Reset" />

</children>

</VBox>

<VBox alignment="CENTER" layoutX="14.0" layoutY="177.0" prefHeight="48.0" prefWidth="213.0">

<children>

<Label text="Name" />

<TextField fx:id="tfName" />

</children>

</VBox>

<VBox alignment="CENTER" layoutX="14.0" layoutY="225.0" prefHeight="48.0" prefWidth="213.0">

<children>

<Label text="Birth Date" />

<TextField fx:id="tfBirthDate" />

</children>

</VBox>

<VBox alignment="CENTER" layoutX="14.0" layoutY="273.0" prefHeight="48.0" prefWidth="213.0">

<children>

<Label text="Age" />

<TextField fx:id="tfAge" />

</children>

</VBox>

<VBox alignment="CENTER" layoutX="15.0" layoutY="328.0" prefHeight="48.0" prefWidth="213.0">

<children>

<Label text="Address" />

<TextField fx:id="tfAddress" />

</children>

</VBox>

<VBox alignment="CENTER" layoutX="14.0" layoutY="383.0" prefHeight="48.0" prefWidth="213.0">

<children>

<Label text="Allergies" />

<TextField fx:id="tfAllergic" />

</children>

</VBox>

<HBox layoutX="543.0" layoutY="520.0" prefHeight="25.0" prefWidth="53.0" />

<HBox layoutX="21.0" layoutY="20.0" prefHeight="100.0" prefWidth="200.0">

<children>

<ImageView fitHeight="151.0" fitWidth="205.0" pickOnBounds="true">

<image>

<Image url="@logo.jpg" />

</image>

</ImageView>

</children>

</HBox>

<TextField fx:id="tfSearch" layoutX="395.0" layoutY="454.0" prefHeight="25.0" prefWidth="329.0" />

<HBox alignment="CENTER" layoutX="424.0" layoutY="503.0" prefHeight="33.0" prefWidth="271.0">

<children>

<Button fx:id="btnSearch" mnemonicParsing="false" onAction="#handleButtonAction" style="css: -fx-background-color: #8B5DFF; -fx-text-fill: white; -fx-font-size: 14px; -fx-background-color: FC8F54;" text="Search" />

</children>

</HBox>

</children>

</AnchorPane>

# 

# **Student.java**

package ainlabtask;

public class Student {

private int ID;

private String name;

private String birth\_date;

private String age;

private String address;

private String allergies;

public Student(int ID, String name, String birth\_date, String age, String address, String allergies) {

this.ID = ID;

this.name = name;

this.birth\_date = birth\_date;

this.age = age;

this.address = address;

this.allergies = allergies;

}

public int getID() { return this.ID; }

public void setID(int ID) { this.ID = ID; }

public String getName() { return this.name; }

public void setName(String name) { this.name = name; }

public String getBirth\_date() { return this.birth\_date; }

public void setBirth\_date(String birth\_date) { this.birth\_date = birth\_date; }

public String getAge() { return this.age; }

public void setAge(String age) { this.age = age; }

public String getAddress() { return this.address; }

public void setAddress(String address) { this.address = address; }

public String getAllergies() { return this.allergies; }

public void setAllergies(String allergies) { this.allergies = allergies; }

}

# **FXMLDocumentController.java**

package ainlabtask;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Button;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

public class FXMLDocumentController implements Initializable {

@FXML

private TextField tfID;

@FXML

private TextField tfName;

@FXML

private TextField tfBirthDate;

@FXML

private TextField tfAge;

@FXML

private TextField tfAddress;

@FXML

private TextField tfAllergic;

@FXML

private TextField tfSearch;

@FXML

private TableView<Student> tvStudent;

@FXML

private TableColumn<Student, Integer> colID;

@FXML

private TableColumn<Student, String> colName;

@FXML

private TableColumn<Student, String> colBirth;

@FXML

private TableColumn<Student, String> colAge;

@FXML

private TableColumn<Student, String> colAddress;

@FXML

private TableColumn<Student, String> colAllergic;

@FXML

private Button btnAdd;

@FXML

private Button btnUpdate;

@FXML

private Button btnDelete;

@FXML

private Button btnReset;

@FXML

private Button btnSearch;

public FXMLDocumentController() {

}

@FXML

private void handleButtonAction(ActionEvent event) {

if (event.getSource() == btnAdd) {

insertRecord();

} else if (event.getSource() == btnUpdate) {

updateRecord();

} else if (event.getSource() == btnDelete) {

deleteRecord();

} else if (event.getSource() == btnReset) {

resetRecord();

} else if (event.getSource() == btnSearch) {

String query = tfSearch.getText().trim().toLowerCase();

if (!query.isEmpty()) {

search(query);

} else {

showStudent();

}

}

}

@FXML

void handleMouseAction(MouseEvent event) {

Student stud = tvStudent.getSelectionModel().getSelectedItem();

if (stud != null) {

tfID.setText(String.valueOf(stud.getID()));

tfName.setText(stud.getName());

tfBirthDate.setText(stud.getBirth\_date());

tfAge.setText(stud.getAge());

tfAddress.setText(stud.getAddress());

tfAllergic.setText(stud.getAllergies());

}

}

@Override

public void initialize(URL url, ResourceBundle rb) {

showStudent();

}

private void insertRecord() {

DatabaseConnection db = new DatabaseConnection();

db.addStudent(

tfName.getText(),

tfBirthDate.getText(),

tfAge.getText(),

tfAddress.getText(),

tfAllergic.getText()

);

showStudent();

}

private void updateRecord() {

Student selectedStudent = tvStudent.getSelectionModel().getSelectedItem();

if (selectedStudent != null) {

// Update the selected student object with the new values from text fields

selectedStudent.setName(tfName.getText());

selectedStudent.setBirth\_date(tfBirthDate.getText());

selectedStudent.setAge(tfAge.getText());

selectedStudent.setAddress(tfAddress.getText());

selectedStudent.setAllergies(tfAllergic.getText());

// Use the updated student object for the update operation

DatabaseConnection db = new DatabaseConnection();

db.updateStudent(selectedStudent);

showStudent(); // Refresh the student list after updating

} else {

System.out.println("No student selected for update.");

}

}

private void deleteRecord() {

Student selectedStudent = tvStudent.getSelectionModel().getSelectedItem();

if (selectedStudent != null) {

// Perform delete operation only if a student is selected

DatabaseConnection db = new DatabaseConnection();

db.deleteStudent(selectedStudent.getID());

showStudent(); // Refresh the list after deletion

} else {

// Show an alert if no student is selected

System.out.println("No student selected for deletion.");

}

}

private void resetRecord() {

tfID.clear();

tfName.clear();

tfBirthDate.clear();

tfAge.clear();

tfAddress.clear();

tfAllergic.clear();

}

private void search(String query) {

ObservableList<Student> filteredList = FXCollections.observableArrayList();

for (Student student : tvStudent.getItems()) {

if (student.getName().toLowerCase().contains(query) ||

student.getBirth\_date().toLowerCase().contains(query) ||

student.getAge().toLowerCase().contains(query) ||

student.getAddress().toLowerCase().contains(query) ||

student.getAllergies().toLowerCase().contains(query) ||

String.valueOf(student.getID()).equals(query)) {

filteredList.add(student);

}

}

tvStudent.setItems(filteredList);

}

public void showStudent() {

ObservableList<Student> studentList = getStudentList();

colID.setCellValueFactory(new PropertyValueFactory<>("ID"));

colName.setCellValueFactory(new PropertyValueFactory<>("name"));

colBirth.setCellValueFactory(new PropertyValueFactory<>("birth\_date"));

colAge.setCellValueFactory(new PropertyValueFactory<>("age"));

colAddress.setCellValueFactory(new PropertyValueFactory<>("address"));

colAllergic.setCellValueFactory(new PropertyValueFactory<>("allergies"));

tvStudent.setItems(studentList);

}

public ObservableList<Student> getStudentList() {

ObservableList<Student> studList = FXCollections.observableArrayList();

DatabaseConnection db = new DatabaseConnection();

studList.addAll(db.getAllStudents());

return studList;

}

}

# **DatabaseConnection.java**

package ainlabtask;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

import java.text.ParseException;

import java.text.SimpleDateFormat;

public class DatabaseConnection {

private static final String URL = "jdbc:mysql://127.0.0.1:3306/studentdb";

private static final String USER = "root";

private static final String PASSWORD = "root";

public static Connection connect() throws SQLException {

return DriverManager.getConnection(URL, USER, PASSWORD);

}

// Retrieve all students

public List<Student> getAllStudents() {

List<Student> students = new ArrayList<>();

String query = "SELECT \* FROM student";

try (Connection conn = connect();

PreparedStatement stmt = conn.prepareStatement(query);

ResultSet rs = stmt.executeQuery()) {

while (rs.next()) {

students.add(new Student(

rs.getInt("id"),

rs.getString("name"),

rs.getString("birth\_date"),

rs.getString("age"),

rs.getString("address"),

rs.getString("allergies")

));

}

} catch (SQLException e) {

e.printStackTrace();

}

return students;

}

// Add a new student

public void addStudent(String name, String birthDate, String age, String address, String allergies) {

String query = "INSERT INTO student (name, birth\_date, age, address, allergies) VALUES (?, ?, ?, ?, ?)";

try (Connection conn = connect();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, name);

// Convert the date string to java.sql.Date format

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

java.util.Date parsedDate = format.parse(birthDate);

stmt.setDate(2, new java.sql.Date(parsedDate.getTime()));

stmt.setString(3, age);

stmt.setString(4, address);

stmt.setString(5, allergies);

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

} catch (ParseException e) {

System.out.println("Invalid date format. Please use 'yyyy-MM-dd'");

}

}

// Update an existing student

public void updateStudent(Student student) {

String query = "UPDATE student SET name = ?, birth\_date = ?, age = ?, address = ?, allergies = ? WHERE id = ?";

try (Connection conn = connect();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, student.getName());

stmt.setString(2, student.getBirth\_date());

stmt.setString(3, student.getAge());

stmt.setString(4, student.getAddress());

stmt.setString(5, student.getAllergies());

stmt.setInt(6, student.getID()); // Use the student's ID to locate the record to update

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

//delete student

public void deleteStudent(int studentId) {

String query = "DELETE FROM student WHERE id = ?";

try (Connection conn = connect();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setInt(1, studentId);

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

// Search for students by criteria

public List<Student> searchStudents(String name, String birthDate, String age, String address, String allergies) {

List<Student> students = new ArrayList<>();

StringBuilder query = new StringBuilder("SELECT \* FROM student WHERE 1=1");

if (!name.isEmpty()) {

query.append(" AND name LIKE ? ");

}

if (!birthDate.isEmpty()) {

query.append(" AND birth\_date = ? ");

}

if (!age.isEmpty()) {

query.append(" AND age = ? ");

}

if (!address.isEmpty()) {

query.append(" AND address LIKE ? ");

}

if (!allergies.isEmpty()) {

query.append(" AND allergies LIKE ? ");

}

try (Connection conn = connect();

PreparedStatement stmt = conn.prepareStatement(query.toString())) {

int index = 1;

if (!name.isEmpty()) stmt.setString(index++, "%" + name + "%");

if (!birthDate.isEmpty()) stmt.setString(index++, birthDate);

if (!age.isEmpty()) stmt.setString(index++, age);

if (!address.isEmpty()) stmt.setString(index++, "%" + address + "%");

if (!allergies.isEmpty()) stmt.setString(index++, "%" + allergies + "%");

try (ResultSet rs = stmt.executeQuery()) {

while (rs.next()) {

students.add(new Student(

rs.getInt("id"),

rs.getString("name"),

rs.getString("birth\_date"),

rs.getString("age"),

rs.getString("address"),

rs.getString("allergies")

));

}

}

} catch (SQLException e) {

e.printStackTrace();

}

return students;

}

}

# **AinLabTask.Java**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/javafx/FXML.java to edit this template

\*/

package ainlabtask;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.stage.Stage;

/\*\*

\*

\* @author nor96

\*/

public class AinLabTask extends Application {

@Override

public void start(Stage stage) throws Exception {

Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

Scene scene = new Scene(root);

stage.setScene(scene);

stage.show();

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

launch(args);

}

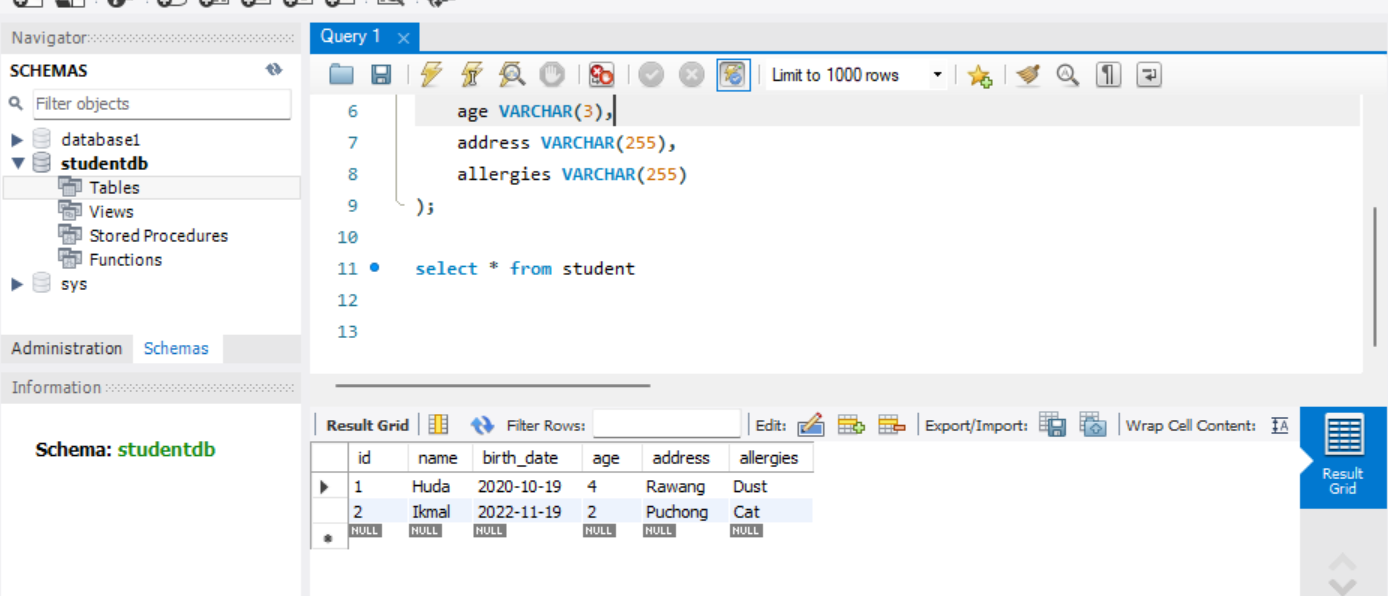
}

# **Output**



**Figure 1.0**

Figure 1.0, show how user can add student data. The data that add will appear in Table view and will be stored inside database(Figure 2.0)



**Figure 2.0**



**Figure 3.0**

Figure 3.0 show, how user can update Student data, by clicking the data inside table.  
Once user click on it, it will update the data inside tableview and also database.(Figure 4.0)



**Figure 4.0**



**Figure 5.0**

Figure 5.0 show,user can search for student data, just using desired keyword and it will filter accordingly.