**JavaFX CRUD Application - Documentation**

# Section A: Project Overview

This JavaFX application is a basic CRUD (Create, Read, Update, Delete) system that allows users to manage student records. It includes a login page for authentication and a secondary interface for CRUD operations. The project utilizes JavaFX for UI, MySQL for backend data storage, and JDBC for database connectivity.

# Section B: Project Setup

- JavaFX SDK installed and configured in the project build path.

- MySQL database named `database\_lab2\_b` with a `students` table.

- Maven used for dependency management.

- Required JDBC driver (MySQL Connector/J) added to the `/lib` directory and referenced in `pom.xml`.

# Section C: Data Models and ORM (120 Marks)

The `Student` class serves as the data model representing rows from the `students` table. It uses JavaFX `Property` classes for binding with UI elements in the TableView. Object-Relational Mapping is handled manually using JDBC queries.

# Section D: Open JDBC Jar (120 Marks)

The MySQL JDBC driver (`mysql-connector-j-8.0.32.jar`) is placed in the `lib` directory and referenced in `pom.xml` with the `<systemPath>` property for integration with Maven.

# Section E: Load Data in Table View (120 Marks)

Upon initialization, the application connects to the database and retrieves student records using SQL queries. These records are loaded into an `ObservableList` and displayed in a TableView.

# Section F: CRUD Operations

- Add: Inserts a new student record into the database.

- Update: Modifies the selected student record.

- Delete: Removes the selected student from the database.

- Clear: Resets the input fields and table selection.

# Section G: Scene Switching and Login

The `PrimaryController` handles login logic. Upon successful authentication using either `login.txt` or the `users` database table, the user is redirected to the CRUD interface defined in `secondary.fxml`.

# Section H: Disclaimer and References

Some components of this application were built with assistance from ChatGPT and W3Schools for resolving errors and improving structure.

GitHub Repository: https://github.com/SaddamHussainSafi/JavaFX\_Application\_LAB\_2B

Section I: Screenshots

Include the following screenshots with appropriate labels:

- Login page interface

A screenshot of a computer

AI-generated content may be incorrect.

- CRUD interface with student data

A screenshot of a computer

AI-generated content may be incorrect.

- MySQL database table structure

A screenshot of a computer

AI-generated content may be incorrect.

- Maven project structure in VS Code

A screenshot of a computer

AI-generated content may be incorrect.

- Terminal output during build/run

A screenshot of a computer program

AI-generated content may be incorrect.

# Disclaimer

I have taken help from ChatGPT and StackOverflow forums for solving errors and understanding concepts during the development of this project. The following StackOverflow discussions were referenced:  
  
• https://stackoverflow.com/questions/37200845/how-to-switch-scenes-in-javafx  
• https://stackoverflow.com/questions/23729277/javafx-fxml-load-file-issues-with-setting-root  
• https://stackoverflow.com/questions/218384/what-is-a-nullpointerexception-and-how-do-i-fix-it  
  
Additionally, ChatGPT was used for syntax validation, JDBC integration assistance, FXML layout corrections, and to overcome runtime errors.