

**WORLDSKILLS TP**

**QUALIFYING ROUND TEST PROJECT  
TEMASEK POLYTECHNIC**

---

**IT SOFTWARE SOLUTIONS FOR BUSINESS**

**SESSION 1**

WS\_TP\_2023\_QR\_S1

**Written by: Temasek Polytechnic**

# 1. CONTENTS

Session 1 of this Test Project consists of the following files:

1. WS\_TP\_2023\_QR\_S1.pdf (Session 1 instructions)
2. WS\_TP\_2023\_QR\_S1\_db\_create.sql - this script will create the necessary tables for user registration and login.
3. WS\_TP\_2023\_QR\_S1\_db\_import.sql - this script will allow you to insert test datasets into your tables.

# 2. INTRODUCTION

In this session, you will start developing the TP Skills 2023 application. Due to the complexity and unique characteristics, an Education Tech company has hired you to develop an in-house customised system to support their various business processes and application development. The designer has provided you with system documentation to build it according to the client's needs. Take some time to review what has been provided and what is required carefully. Prepare your test data to help you to test the application. Document your test data, assumptions and critical information in a Readme.txt file and save it with the other files you submit.

### 3. INSTRUCTIONS FOR THE COMPETITOR

In this session, you will develop a desktop application. While developing this application, please conform to the following basic instructions:

- You should consistently follow the provided style guide throughout the application.
- Time management is critical to the success of any project, so by the end of this session, you should submit the deliverables listed in the next section so that the TP Skills 2023 application will be finished on time. Any deliverables that are not submitted on time will not be evaluated.
- Make sure you follow the provided style guide throughout all system parts. Make sure that you follow the general layout and flow of the screens as outlined in this document and the storyboard.
- Make sure that you provide appropriate validation and error messages throughout all parts of the system.
- Ensure all relevant buttons/links are working at the end of the session. Make sure you use appropriate naming conventions for all parts of the system.
- Where applicable, include comments in your code to make it easier for evaluators and clients to understand your code.
- Do note that you are building the entire system progressively, so some functions may only be added in subsequent sessions.

## 4. WORKING WITH THE DATABASE

Create a database named “Session1”. This will be the only database that you use in this session. Save this database in your main project folder. Two SQL Scripts are provided for you. These scripts consist of the database structure and data required to complete the tasks in this session. As instructed by the designers, the database structure for this session cannot be altered (i.e. no adding or removing tables, fields in the tables or data types).

For the database, you will need to create tables to store the user information (e.g. first name, last name, email address, password, and user type) and write SQL queries to handle the registration and login processes. The scripts to create the necessary tables are given in Session 1 folder. These are:

1. WS\_TP\_2023\_QR\_S1\_db\_create.sql - this script will create the necessary tables for user registration and login.
2. WS\_TP\_2023\_QR\_S1\_db\_import.sql - this script will allow you to insert test datasets into your tables.

## **5. DELIVERABLES**

### **5.1. ALERT MODAL BOX**

You will have to use an alert (modal style) dialogue box to confirm if the user discards or cancels any changes in any of the screens. For example, throughout the screens below, you might find situations where the user can cancel the changes and go to the previous screen. Although it is not inherently stated, you must assume that you must show an alert box in such situations. The alert box should also show the user when to notify of any invalid input. Depending on your need, you will need to customise the message displayed on this alert box.

### **5.2. MAIN SCREEN**

The main screen allows an entry point to the application. Use the wireframe outlined in the Wireframes document under the heading MAIN SCREEN.

### **5.3. REGISTRATION SCREEN**

To register users, you should create a registration screen. The registration screen should have a form with the following fields:

1. First name
2. Last name
3. Email address
4. Password
5. Confirm password
6. User type (Tutor or Student)

The form should also include a **Register** button to submit the form. Do note that the form should validate the input fields to ensure that all required fields are filled out and the email address and password meet the necessary criteria (e.g. valid email format, password meets the minimum length and complexity requirements). Upon successful registration, the application should redirect to the login screen. In the event of an error (e.g. invalid input, email address already in use), the user should be notified and allowed to correct the mistakes.

#### 5.4. LOGIN SCREEN

The login screen will allow registered users to log in to the application. The login screen should have a form with the following fields:

1. Email address
2. Password

The form should also include a **Log In** button to submit the form and allow the user to log in, provided the information keyed in is correct. The form should validate the input fields to ensure that the email address and password are filled out. Upon successful login, the application should redirect the user to their respective dashboard (**Tutor Dashboard Screen** or **Student Dashboard Screen**). In the event of an error (e.g. invalid email address or password), the user should be notified and allowed to correct the error.

#### 5.5. TUTOR DASHBOARD SCREEN

The tutor dashboard screen comprises a text welcoming the user on the top of the screen. Below is a button called **Create Subject**. Once a user clicks on this button, they are routed to the **Subject Creation Screen**. It also displays a list view of all the subjects created by the tutor. Each item in the list view is displayed as a Subject Card.

The subject card should display the Subject Code, Subject Name, Subject Description (max 100 characters if the number of characters is more than 100), and a button called **Open** to open this card and go to the **Subject Detail Screen**.

## 5.6. STUDENT DASHBOARD SCREEN

The student dashboard screen displays the welcome message.

## 5.7. SUBJECT CREATION SCREEN

This screen should allow Tutors to create new Subjects by entering the following information:

1. Subject name
2. Subject description
3. Subject code (format AlphabetAlphabetNumberNumberNumber, for example, CS102, AI501, EN100, etc.)

The screen should have a **Cancel** button that will discard the information keyed in by the user and bring the user back to the Tutor Dashboard Screen. The screen should also include a **Create** button to submit the form. The form should validate the input fields to ensure that all required fields are filled out, and the subject code meets the necessary criteria (e.g. unique, follows a specific format). The Tutor should be redirected to the Subject details screen after successfully creating the Subject. In the event of an error (e.g. invalid input, subject code already in use), the Tutor should be notified and allowed to correct the mistakes.

## 5.8. SUBJECT DETAILS SCREEN

This screen should display the following information about the Subject:

1. Subject name
2. Subject description
3. Subject code

The screen should allow the Tutor to edit the Subject information by clicking an **Edit** button and opening the Subject Edit Screen. There should also be a **Back** button that will bring the user to the Tutor Dashboard Screen.

## 5.9. SUBJECT EDIT SCREEN

This screen should allow the Tutor to edit the Subject information (subject name, description, and code) and submit the changes by clicking a **Save** button. Besides, validate the input fields to ensure that all required fields are filled out and the subject code meets the necessary criteria (e.g. unique, follows a specific format). The Tutor should be redirected to the Subject Details Screen after successfully updating the Subject information. The tutor should also have a Cancel button to discard the changes and return them to the **Subject Detail Screen**. In the event of an error (e.g. invalid input, subject code already in use), the Tutor should be notified and allowed to correct the mistakes.