PE\_PRN222\_SU25\_TrialTest

SUMMER 2025  
Subject: PRN222  
Duration: 85 minutes

**INSTRUCTIONS**

**Please read the instructions carefully before doing the questions.**

* You are **NOT allowed** to use any other materials. You are **NOT allowed** to use any device to share data with others.
* You must use **Visual Studio 2019 or above, MSSQL Server 2008 or above** for your development tools.

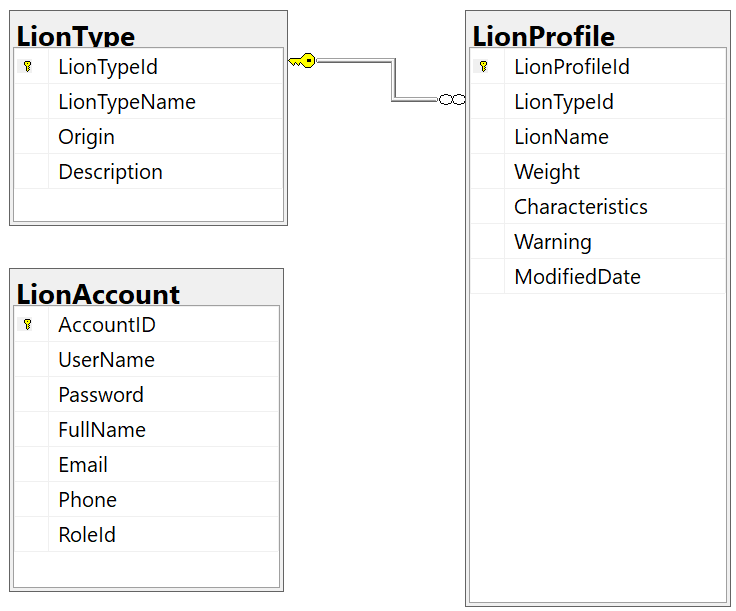
**IMPORTANT – before you start doing your solution, MUST do the following steps:**

1. Create Solution in Visual Studio named **PE\_PRN222\_SU25\_TrialTest\_StudentName**. Inside your Solution, the Project ASP.NET Core Razor Page must be named: **LionPetManagement\_StudentName.**
2. To do your program, you must use **ASP.NET Core Web App (Razor Pages)**, apply 3-Layer architecture (Presentation Layer, Business Logic Layer, Data Access Layer), there are at least 2 Projects fortheSolution. *The database connection string must get from the appsettings.json file.*

***In the case your code connects direct to the database from ASP.NET Core Web App (Razor Pages) or hard coded the connection string, you will get 0 point.***

1. Create your MS SQL database named **SU25LionDB** by running code in script **SU25LionDB.sql.**
2. Set the default user interface for your project as a **Login** page.
3. ***If there are syntax errors or compilation errors in your PE program, you will not pass the PE requirements, the mark will be 0.***
4. ***Your work will be considered invalid (0 point) if your code inserts stuff that is unrelated to the test.***

You are required to build an application using ASP.NET Core Razor Pages. The application must connect to the following SQL Server database named **SPU5LionDB**



*Store account role: Manager = 2; Staff = 3;*

|  |  |  |
| --- | --- | --- |
| ***Role***  ***Function/URL*** | ***Manager*** | ***Staff*** |
| ***Create*** | x |  |
| ***Update*** | x |  |
| ***Delete*** | x |  |
| ***View list of items*** | x | x |
| ***View detail of an item*** | x | x |
| ***Search*** | x | x |

*Table 1. Permissions to implement functions in the test*

|  |  |
| --- | --- |
| ***Function/URL*** |  |
| ***Create*** | *http://localhost:XXXX/LionProfile/Create* |
| ***Update*** | *http://localhost:XXXX/LionProfile/Update* |
| ***Delete*** | *http://localhost:XXXX/LionProfile/Delete* |
| ***View list of items*** | *http://localhost:XXXX/LionProfile/Index* |
| ***View detail of an item*** | *http://localhost:XXXX/LionProfile/Detail* |
| ***Search*** | *http://localhost:XXXX/LionProfile/Search* |
| ***Login*** | *http://localhost:XXXX/Login* |

*Table 2. Mapping between functions and URLs*

**Important Notes:**

* The roles and permissions are defined in Table 1.
* Your application must use Razor Page routes as Table 2. **If any feature does not follow the required URL structure, the score for that feature will be zero**.
* **If authentication or authorization is implemented incorrectly, the entire exam will receive 0 points**.
* Authentication must be done using email and password.
* Login page must appear first when running the application.
* If the login fails (wrong email or password), display: "Invalid Email or Password!"
* If a user tries to access a feature they don’t have permission for, show: "You have no permission to access this function!"

**Question 1 (1.0 point)**

Implement the Login page using email and password. Upon successful login, redirect to the LionProfile list page in Question 2.

**Question 2 (1.5 points)**

Implement the View List function for LionProfile with the following:

* List all of the items in the LionProfile table. Display each record with its associated LionTypeName.
* Paginate the list to show 3 records per page.

**Question 3 (2.5 points)**

Implement the Create function to add new LionProfile into the database with the following validations:

* All fields are required.
* LionName: Minimum 4 characters, Each word starts with a capital letter. No special characters (#, @, &, (, )).
* Weight must be greater than 30.
* LionTypeId must be selected from a dropdown sourced from the LionType table
* The newly added item should appear **at the top of the list**.

**Question 4 (2.0 points)**

Implement the Update function using the same validations as the Create function.

**Question 5 (1.5 points)**

Implement the Search function with the following:

* Allow filtering by **Weight** or **LionTypeName** (OR logic). The UI must include these two input fields.
* Use **partial match** for LionTypeName.

**Question 6 (1.5 points)**

Implement the Delete function using SignalR for real-time updates with the following:

* Confirm deletion before proceeding.
* After deletion, **all connected clients** should receive the updated list automatically.