**CS 504 – Software Engineering**

**HOP07 – Blazor Application - Connecting to the Database**

11/16/2020 Developed by Kim Nguyen & Yu-Che Liu

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

Icon

Description automatically generated

**Before You Start**

* Version numbers may not match with the most current version at the time of writing. If given the option to choose between stable release (long-term support) or most recent, please choose the stable release rather than beta-testing version.
* This tutorial targets Windows users and MacOS users.
* There might be subtle discrepancies along the steps. Please use your best judgement while going through this cookbook style tutorial to complete each step.
* For your working directory, use your course number. This tutorial may use a different course number as an example.
* The directory path shown in screenshots may be different from yours.
* If you are not sure what to do or confused with any steps:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Continue customizing Blazor Application
* Connect the application to the SQLite3 database.

**Resources:**

* Microsoft | ASP .NET - <https://dotnet.microsoft.com/apps/aspnet>

**Blazor App – Connecting to the Database**

Since we want to have users for our app, we will need a database.

**What is Database and why use it?**

A database is a data structure that stores organized information.

We use Databases because it can store very large numbers of records efficiently (they take up little space). It is very quick and easy to find information. It is easy to add new data and to edit or delete old data.

**What is SQLite Database?**

There are different types of Database. In this course, we will use SQLite.

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private.

**Please move the Blazor App folder from “Module 6” folder to “Module 7”.**

(Under the Hands-on Practice folder)

First, let’s add SQLite database to our app. An explanatory video will be provided in the end of the tutorial.

1. Open the Blazor App project in VSCode. Open the Terminal in VSCode by hitting the control + ~ key. Type the following command to add SQLite to the app:

dotnet add package System.Data.SQLite.Core

1. Then, under the project folder. Under Data folder, create a new file and name it SqliteService.cs

Graphical user interface

Description automatically generated

1. Add the following code to the SqliteService.cs

Text

Description automatically generated

Text

Description automatically generated

1. Under Data folder, go to LoginService.cs file, add the following block of code:

Text

Description automatically generated

1. Under Data folder, go to MemberData.cs file, add the following code:

Graphical user interface, text, application

Description automatically generated

1. Under Pages folder, go to Login.razor file, add the following code:

Text

Description automatically generated

1. Under the root folder, in Startup.cs file, add the following line of code:

Text

Description automatically generated

1. Run the app:

dotnet watch run

Go to Login page, click Login button, you should see:

Graphical user interface, application

Description automatically generated

1. **Watch the following video to see the explanations of all the code we just created this week:**

<https://youtu.be/6SKtBERYGwI>

Next week, we will add Sign up function and look into authentication.

1. **Q&A: *(Save your answer into PDF version and save the file under “Module 7” folder.)***

**What are the pros and cons of using SQLite database?**

Apparently sqlite is written completely with the C library.

And interestingly, sqlite is not supposed to run in a server, it is supposed to be embedded into the project. I guess we can inject the sqlite into the project and then upload the project to the server. But this embedding also allows us to embed the database to a project that we will send to the customer. No need for us or the customer to worry with database downloading, installing, configuring, connecting, etc.

Sqlite is lighweight (a term used for apps that don’t require much resources both cpu and memory wise).

No installation needed.

Limited in size to 2 GB.

**Push your work to GitHub**

Open the terminal from the VSCode by hitting the control + ~ key, make sure you are in the right path, for example: /Desktop/CS504/HOP07-YourGitHubUsername/Module 7

Type the following command:

git add . (to copy all changes you have made)

git commit -m “Submission for Module 7 – Your Name” (To add a message to your submission)

git push origin master (to upload your work to Github)