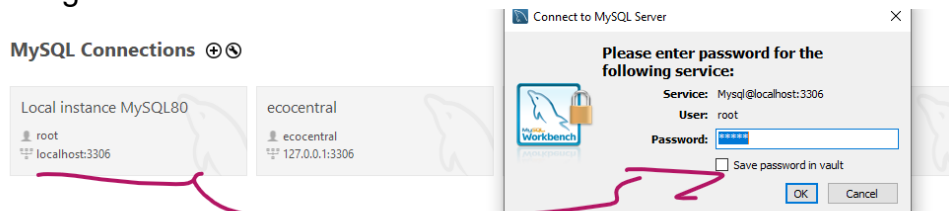# Project Guide for EDP ඩඩඩඩඩඩඩ
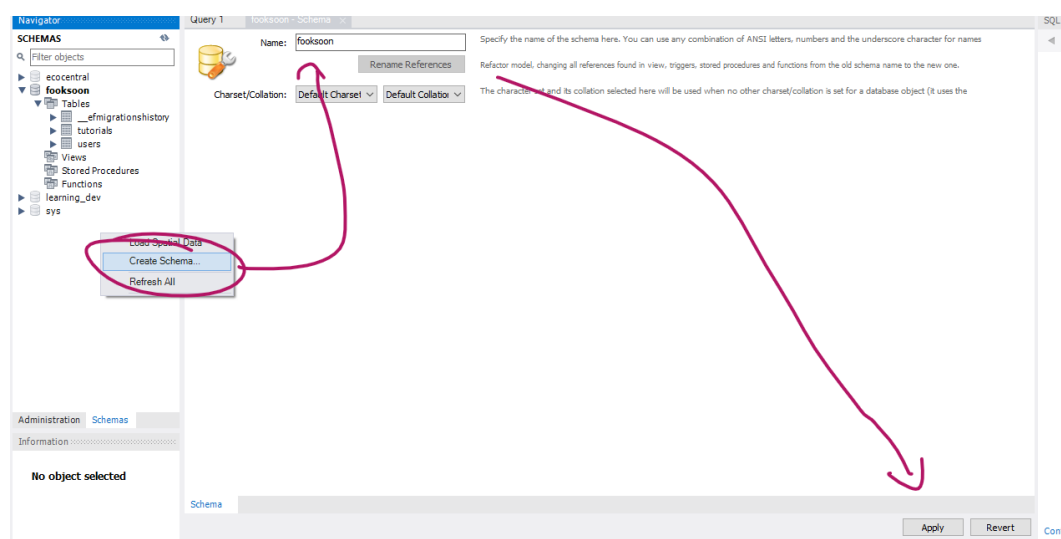
First time setup:

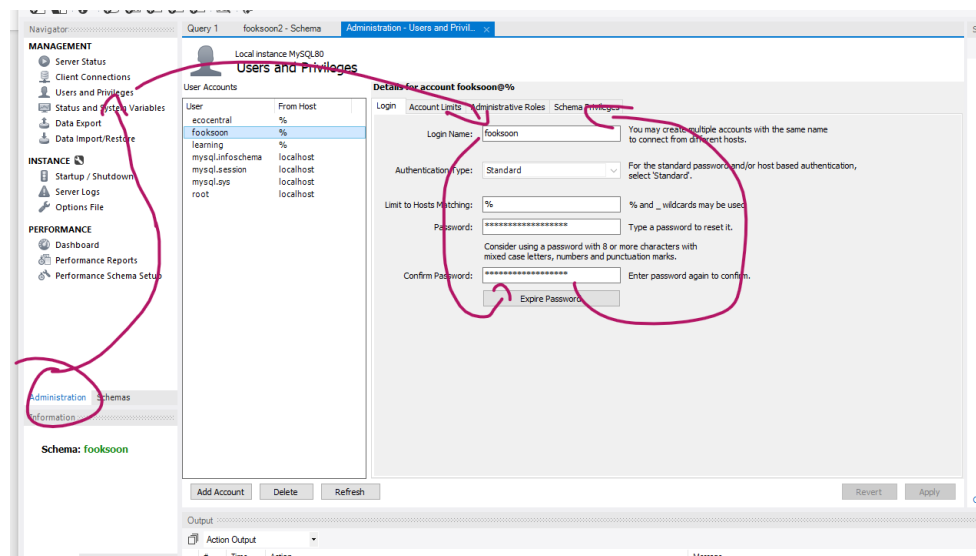Setup new SQL Workbench

## 1. Login as local instance



## 2. Create new schema and name it as fooksoon



## 3. Make admin account for fooksoon schema with all privileges.

Username: fooksoon | Password: mysql

Local instance MySQL80
**Users and Privileges**

**Details for account fooksoon@%**

| Login | Account Limits | Administrative Roles | Schema Privileges |

| Schema | Privileges |
|--------|-----------|
| fooksoon | ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, |

From Host
%
%
%
schema    localhost
sion       localhost
            localhost
            localhost

Schema and Host fields may use % and _ wildcards.
The server will match specific entries before wildcarded ones.

Revoke All Privileges    Delete Entry    Add Entry...

**New Schema Privilege Definition**

Select the Schema for which the user 'fooksoon' will have the privileges you want to define.

**Schema**
- ○ All Schema (%)            This rule will apply to any schema name.
- ○ Schemas matching pattern:    This rule will apply to schemas that match the given name or pattern.
                                  You may use _ and % as wildcards in a pattern.
                                  Escape these characters with \ in case you want their literal value.
- ● Selected schema:    fooksoon    Select a specific schema name for the rule to apply to.

"ALL

---

| fooksoon | ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, |

Schema and Host fields may use % and _ wildcards.
The server will match specific entries before wildcarded ones.

Revoke All Privileges    Delete Entry    Add Entry...

The user 'fooksoon'@'%' will have the following access rights to the schema 'fooksoon':

**Object Rights**
- ☑ SELECT
- ☑ INSERT
- ☑ UPDATE
- ☑ DELETE
- ☑ EXECUTE
- ☑ SHOW VIEW

**DDL Rights**
- ☑ CREATE
- ☑ ALTER
- ☑ REFERENCES
- ☑ INDEX
- ☑ CREATE VIEW
- ☑ CREATE ROUTINE
- ☑ ALTER ROUTINE
- ☑ EVENT
- ☑ DROP
- ☑ TRIGGER

**Other Rights**
- ☐ GRANT OPTION
- ☑ CREATE TEMPORARY TABLES
- ☑ LOCK TABLES

Unselect All    Select "ALL"

Revert    Apply

## 5. Make new connection

**Welcome to MySQL Workbench**

**Setup New Connection**

| Connection Name: | fooksoon | | Type a name for the connection |
| Connection Method: | Standard (TCP/IP) | | Method to use to connect to the RDBMS |

| Parameters | SSL | Advanced |

| Hostname: | 127.0.0.1 | Port: | 3306 | Name or IP address of the server host - and TCP/IP port. |
| Username: | root | | | Name of the user to connect with. |
| Password: | Store in Vault ... | Clear | | The user's password. Will be requested later if it's not set. |
| Default Schema: | fooksoon | | | The schema to use as default schema. Leave blank to select it later. |

Br                                                                    the Forums >

**MySQL Connections** ⊕ ⊙

Local instance MySQL80
👤 root
🖵 localhost:3306

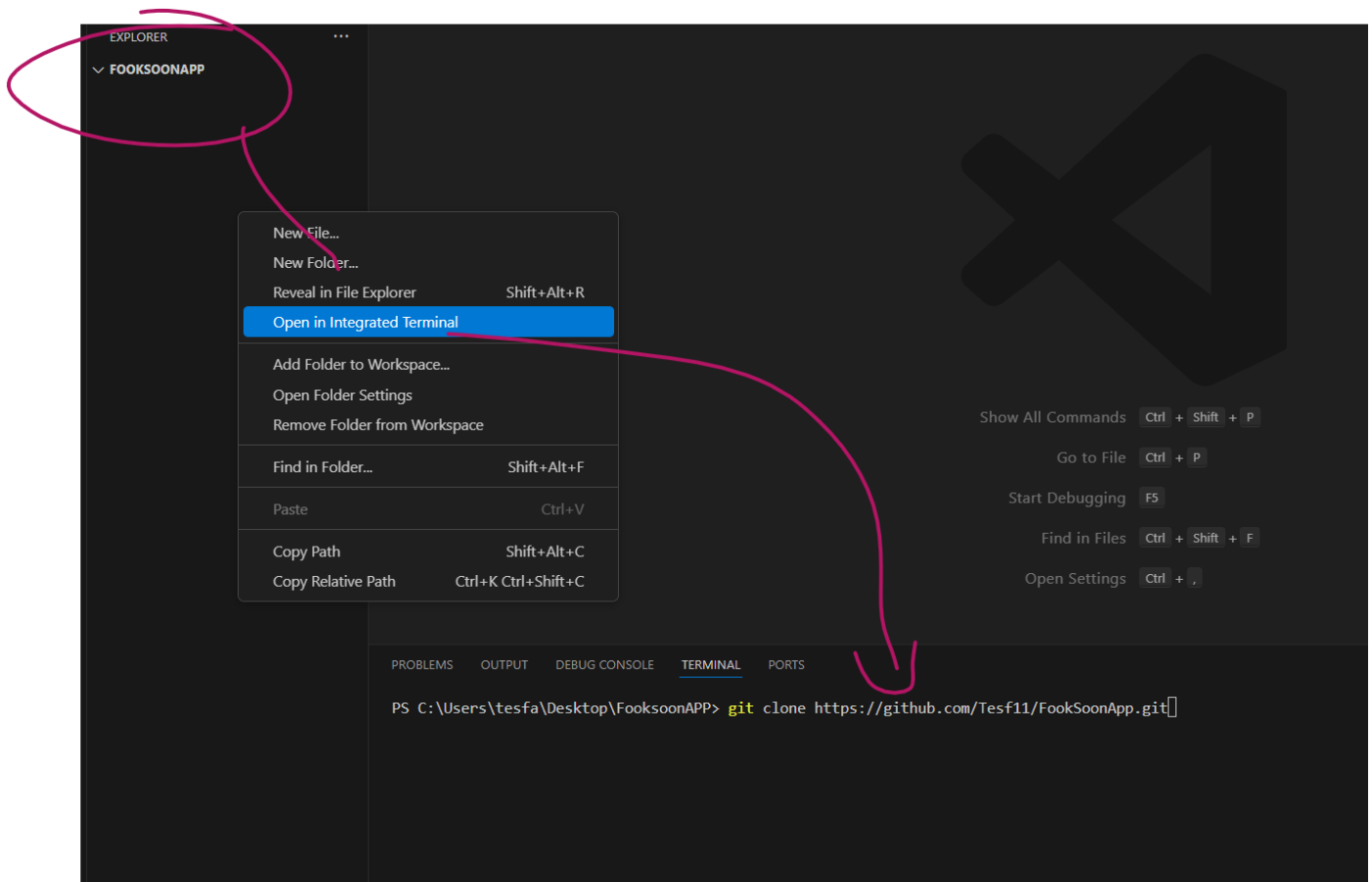Configure Server Management...    Test Connection    Cancel    OK

Setting up github repo

1.Make a dedicated folder (Name it like EDP folder or something) in somewhere safe and sound
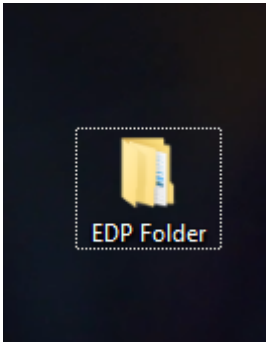


2. Open it in VS Code and start integrated terminal and type

git clone https://github.com/Tesf11/FookSoonApp.git



Project Structure:

Frontend is using React, similar to Full stack development

1. Do "npm i" for first time setup

2. Do "npm start" to start local server"



You're suppose to use Visual studio 2022 for the backend but for convenience sake, we will be using VS Code. Bonus: Jialong can code on his mac as well.

To start backend:

1. Right click and start integrated terminal (ALERT: MAKE SURE THE TARGET DIRECTORY IS FOOKSOONAPI, NOT BACKEND)
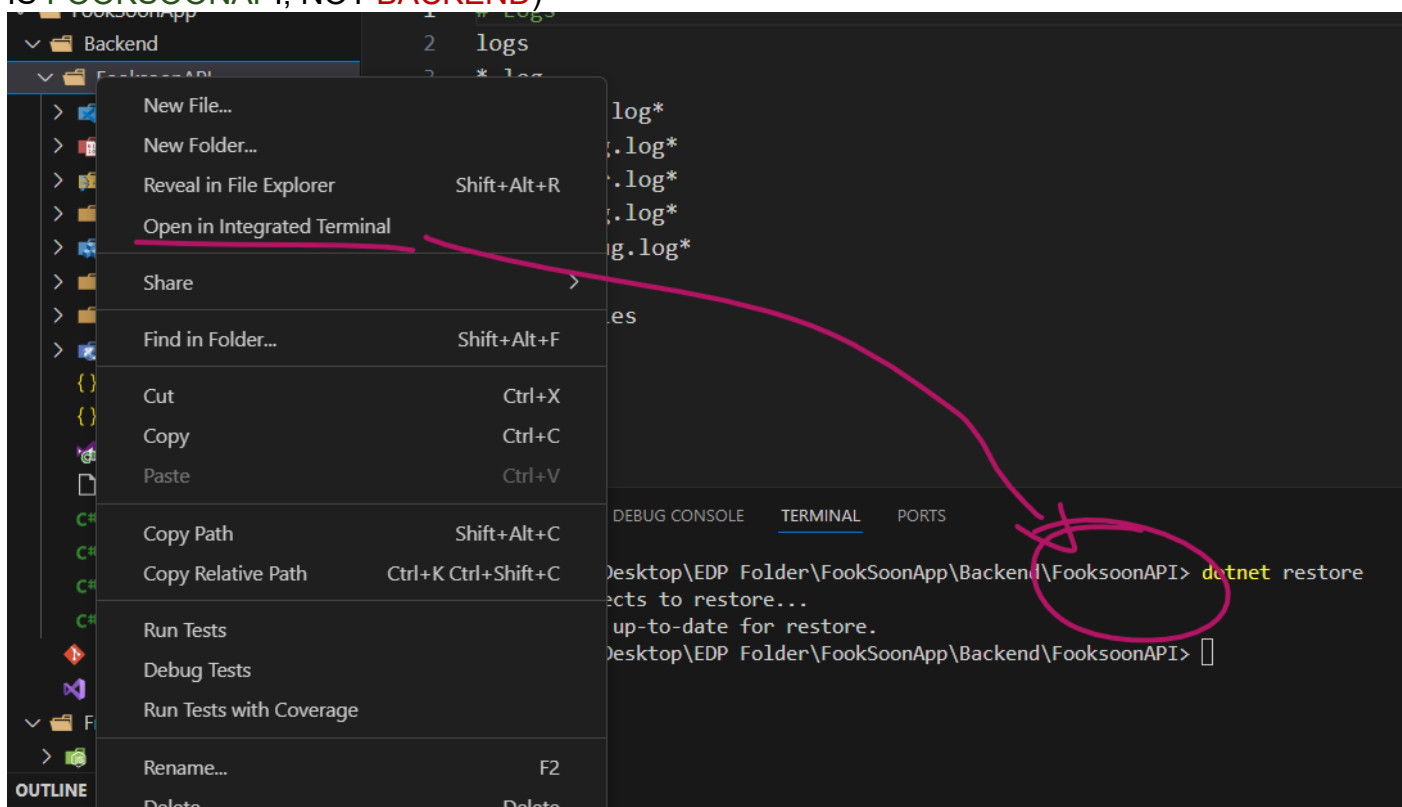
2.  (First time setup)
    Type:
    dotnet restore

    dotnet dev-certs https --trust

    dotnet ef database update (Must update database everytime you make a model change like
    a new User table)



3.  To run the program, type in the following (this command will restart and run the backend
    upon any changes)
    dotnet watch run --urls "https://localhost:7004;http://localhost:5082"

Scroll down

The usual stuff:

**Github Link:**

git clone https://github.com/Tesf11/FookSoonApp.git

**To make changes:**

**Note | Do it under this directory:**

```
PS C:\Users\tesfa\Desktop\EDP Folder\FookSoonApp\Backend\FooksoonAPI> cd ..
PS C:\Users\tesfa\Desktop\EDP Folder\FookSoonApp\Backend> cd ..
PS C:\Users\tesfa\Desktop\EDP Folder\FookSoonApp>
```

*↑ Do Git there*

git add .

git commit -m "Type your Changes here" e.g. Include change messages

git push origin main

**To Update and Pull changes:**

git pull origin main

**Run backend**

dotnet watch run --urls https://localhost:7004;http://localhost:5082

**Run Frontend**

npm start

**Update database:**

dotnet ef database update (do it under backend/FooksoonAPI directory)

**REVERTING A COMMIT (BE VERY CAREFUL)**

git reset —hard (Branch Code from Github)

git push origin main —force

**To update the git repo link:**

git remote set-url origin https://github.com/Tesf11/FookSoonApp.git