

# Databases - Exercise

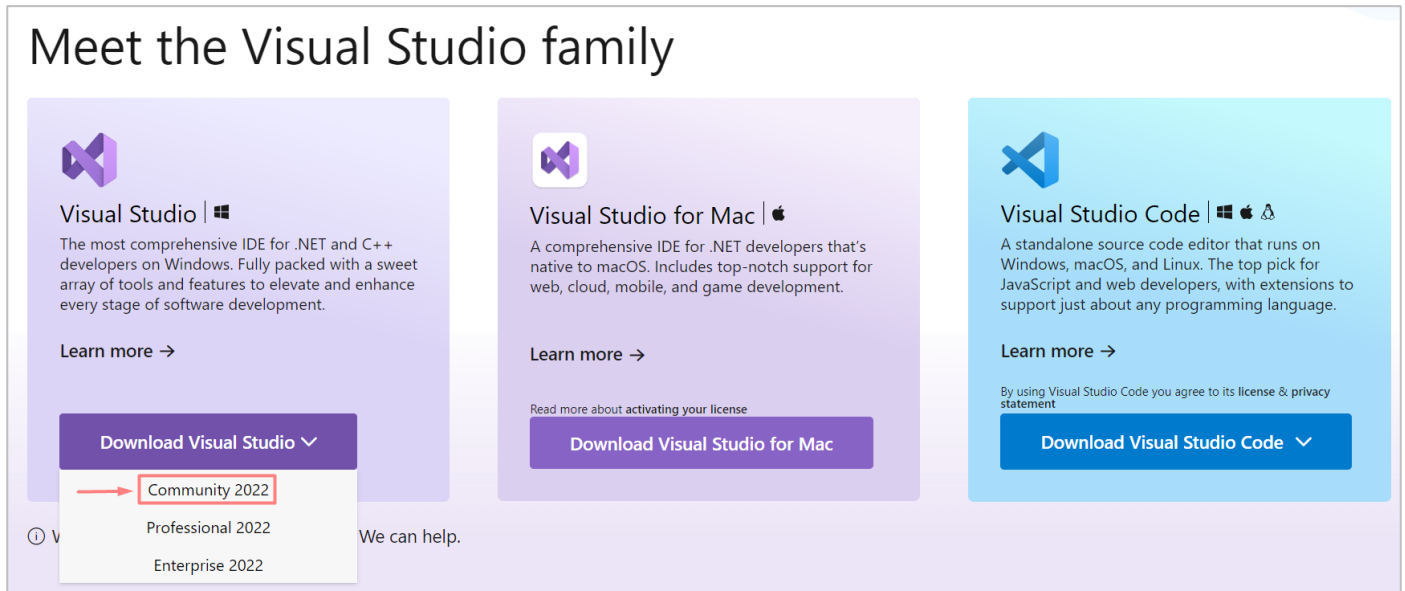
## 1. Visual Studio Setup

*Steps for Installation and Setup of Visual Studio for C#*

### Step 1:

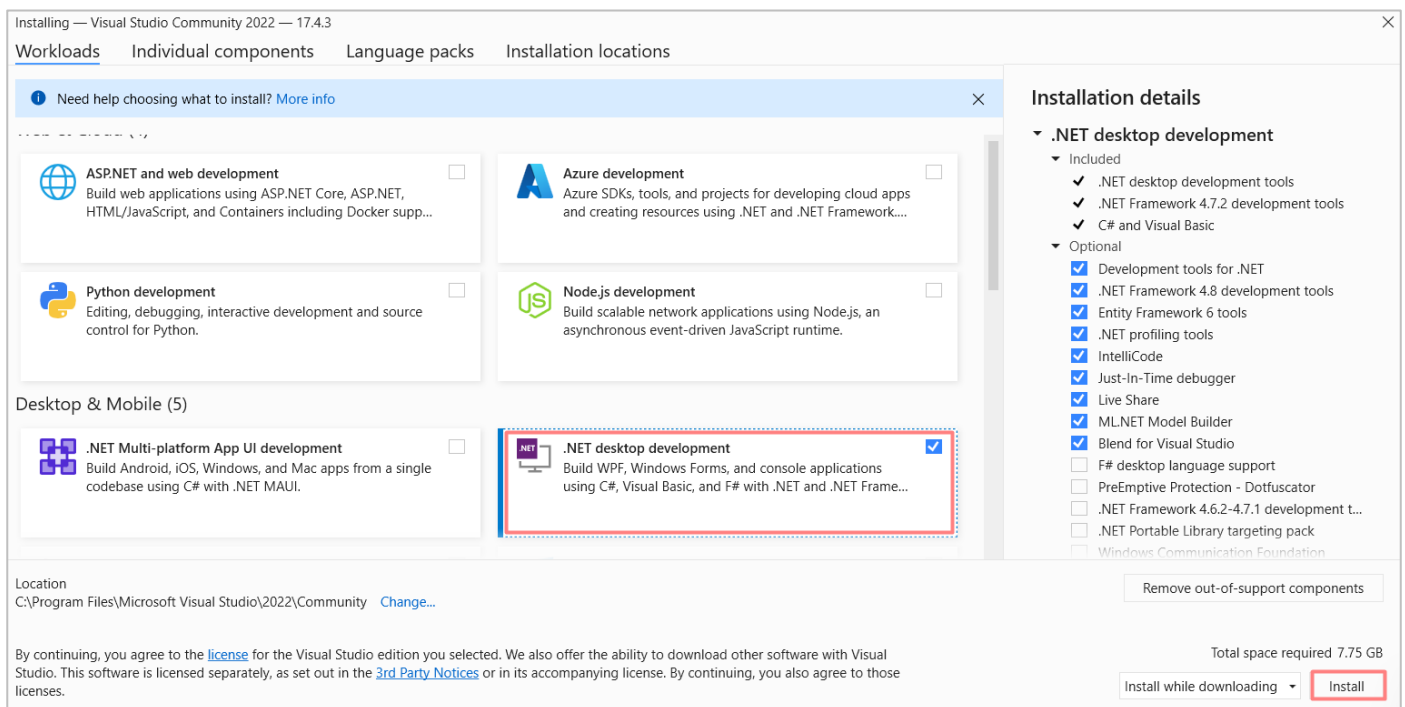
First, there is a need to download Visual Studio. Open <https://visualstudio.microsoft.com>.

Download the **Visual Studio Community** installer (the free Community edition).



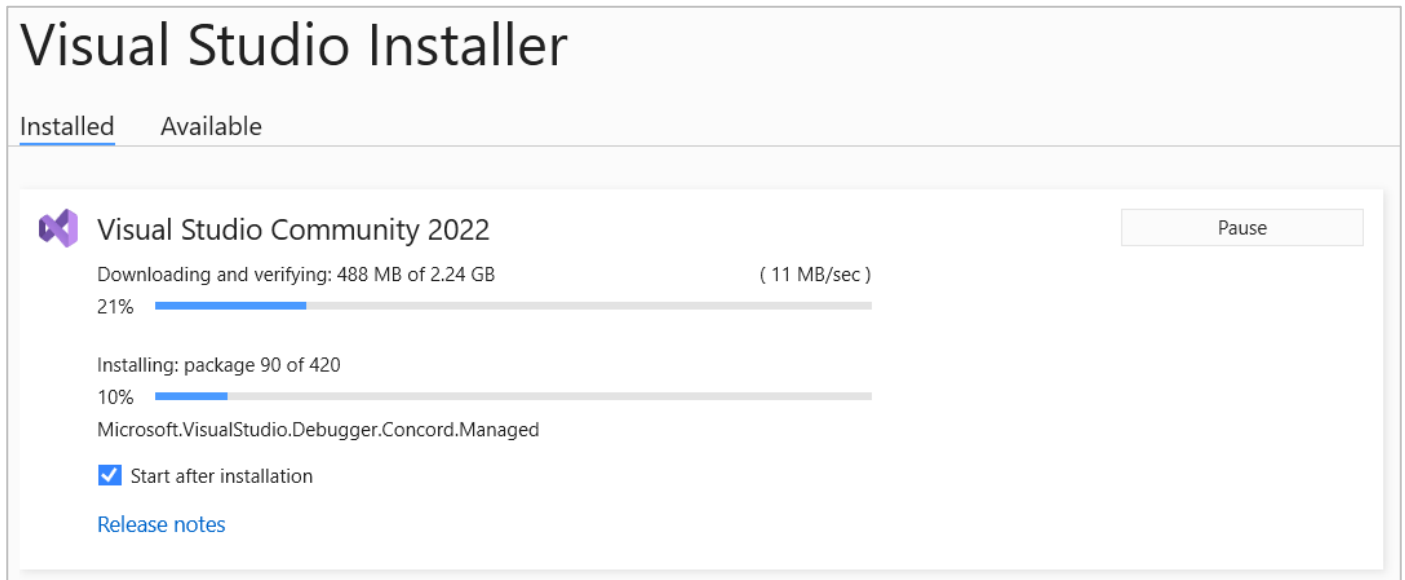
### Step 2:

After downloading the Visual Studio Installer, select the **.NET desktop development** option and then click on **Install**.



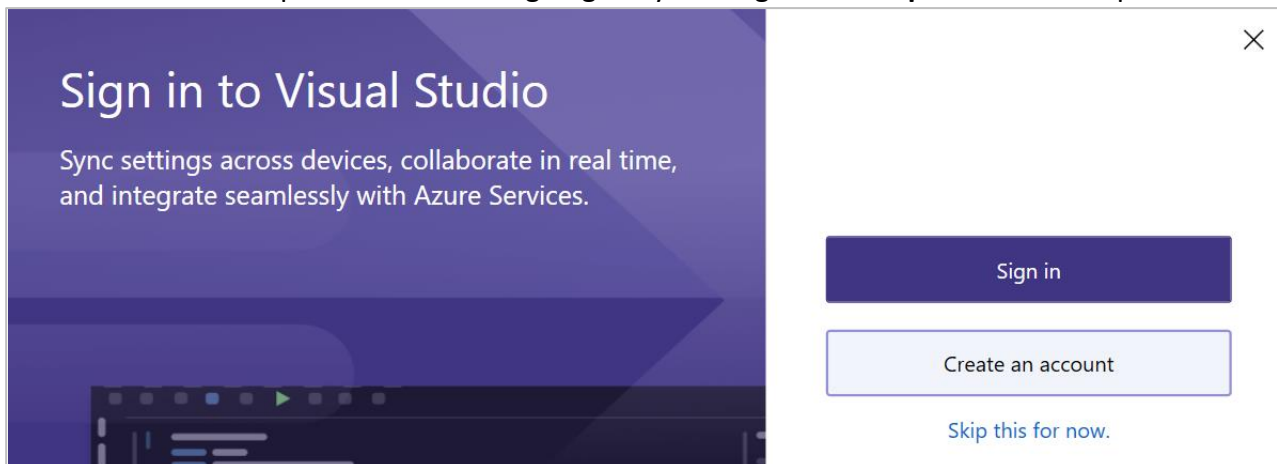
### Step 3:

Now let the **Visual Studio Installer** download the packages and perform the installation.



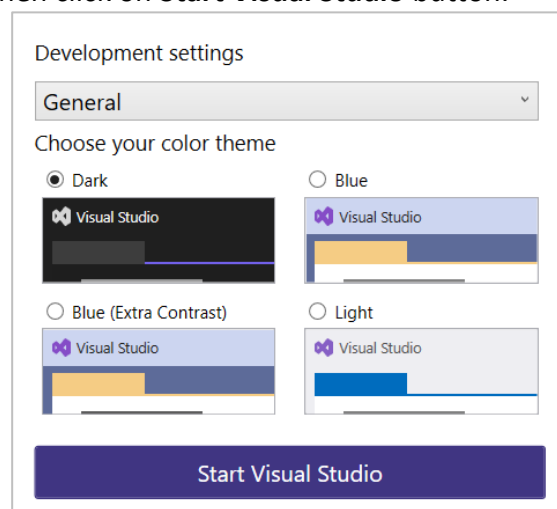
### Step 4:

After the installation is over, begin with the Visual Studio by **Signing In** to your Microsoft account or **Create an account!** You can also proceed without Signing In by clicking on the **Skip this for now** option.



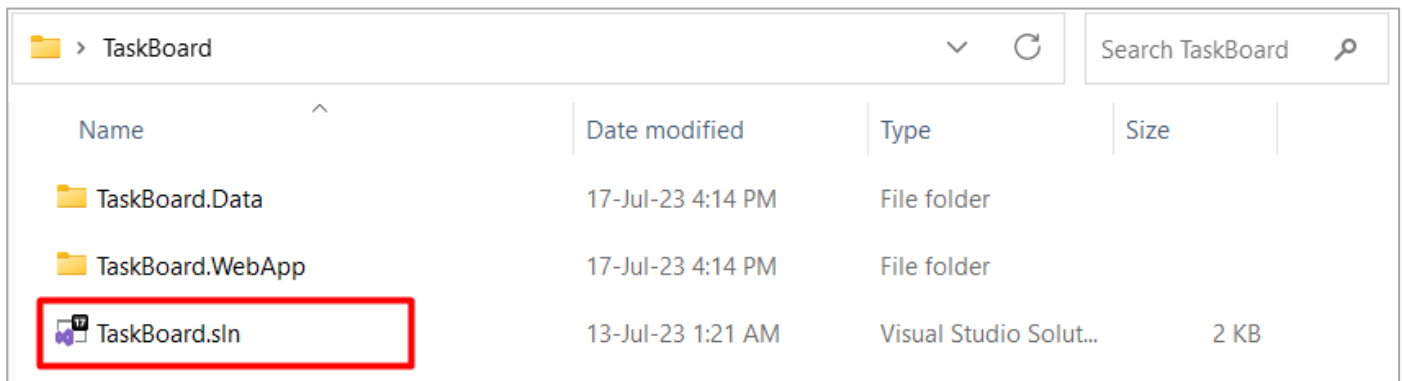
### Step 5:

Choose your color theme and then click on **Start Visual Studio** button.

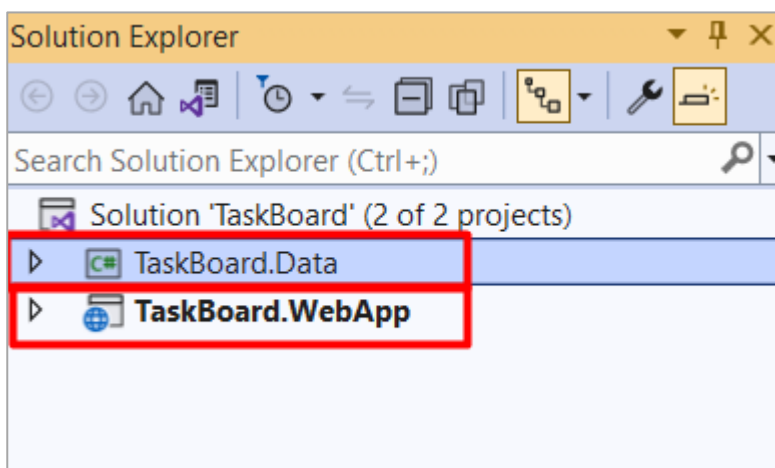
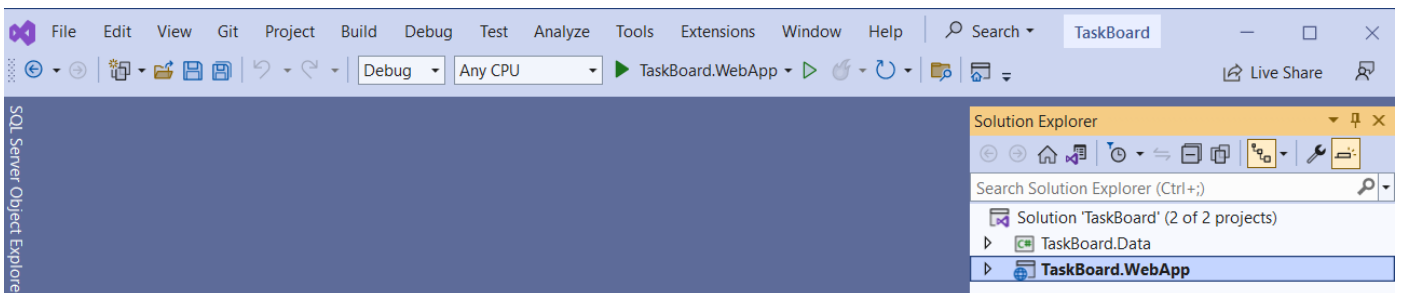


## 2. Running The "TaskBoard" App

1. Navigate to the folder when you extracted the "TaskBoard" from the provided .zip archive and find the TaskBoard.sln. Double click it.



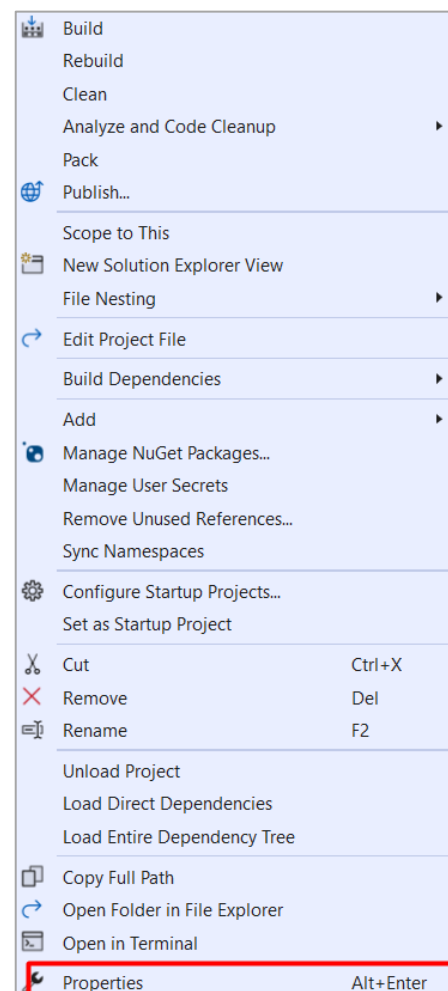
2. This will open TaskBoard solution in your Visual Studio. As you can see, on the right side, in the Solution Explorer there are two projects in the solution: TaskBoard.Data and TaskBoard.WebApp

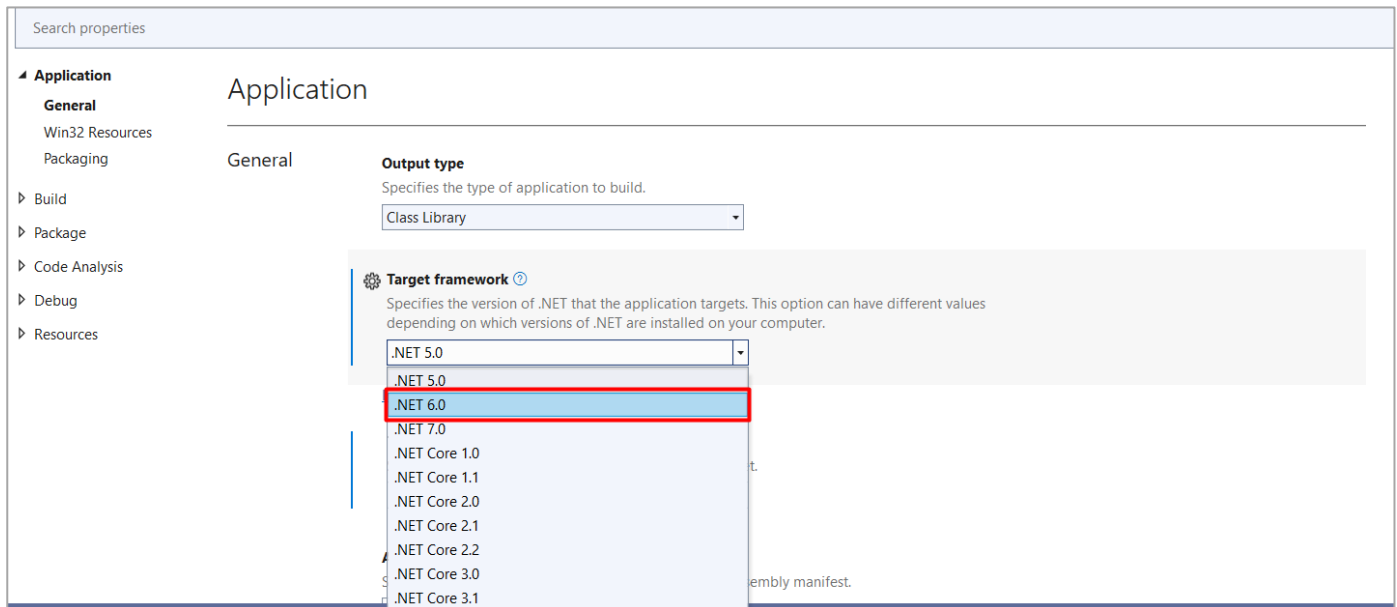


3. You will have to right click on the TaskBoard.Data, which will open the context menu and choose "Properties" option.

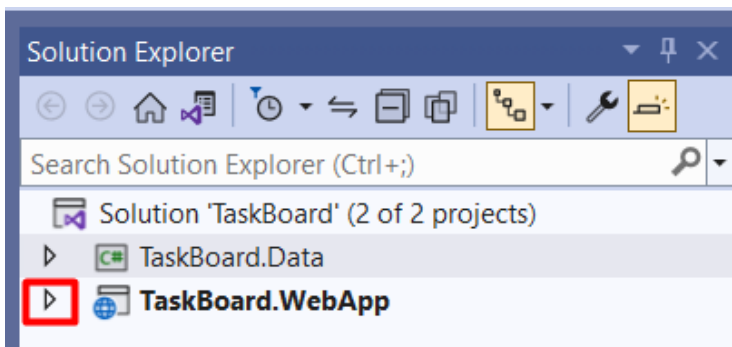
Since the app targets .NET5 Framework, which soon will be out of Support, it would be wise to change it to .NET6 as shown below.

4. Now do exactly the same for TaskBoard.WebApp. Change the Framework.

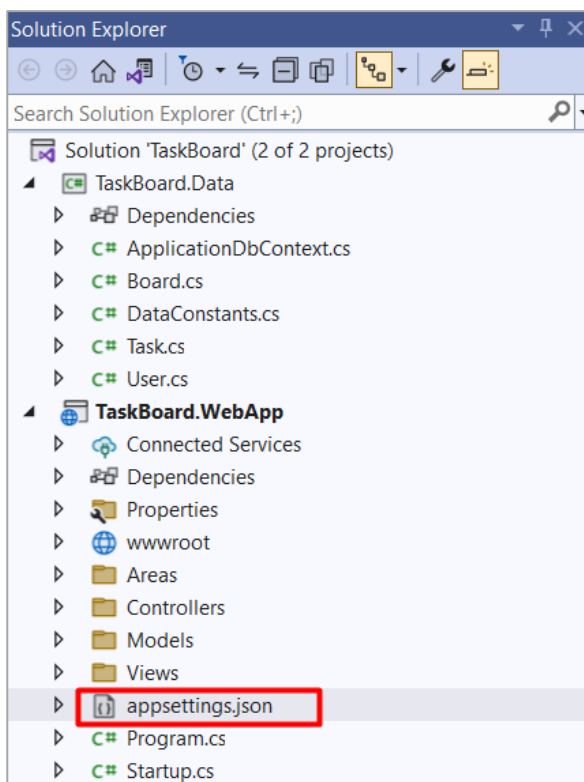




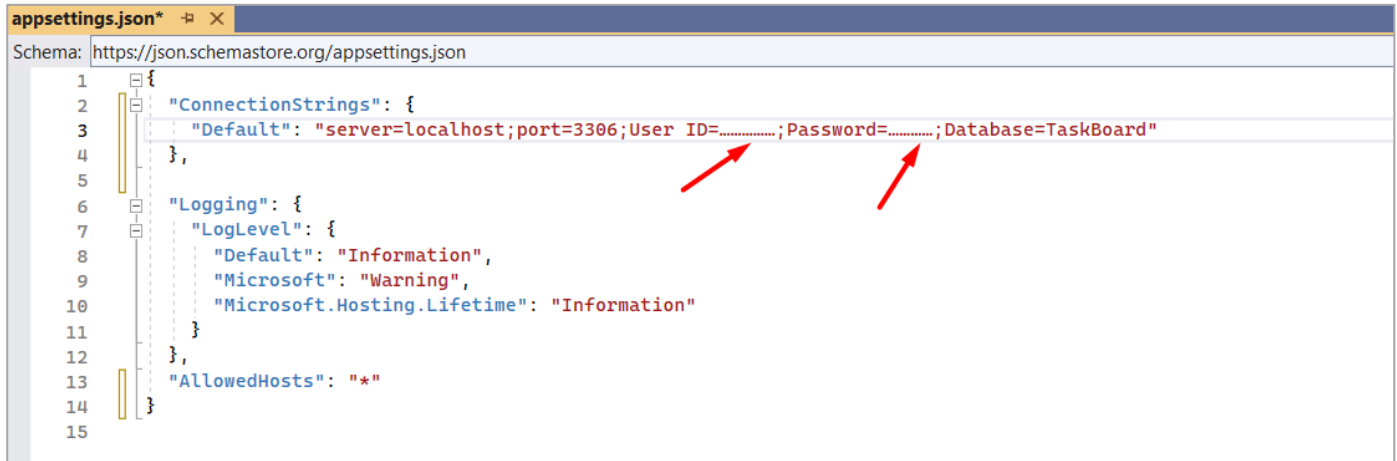
5. When you're done changing the Framework. Click (normal click) on the small triangle in front of the TaskBoard.WebApp project in the Solution Explorer.



6. This will open all the files that belong to the TaskBoard.WebApp Project. You will need to find the **appsettings.json** file. Double click to open it.



7. When opened the json file will look like the picture below. Your task is to change the dots in User ID and password, with those that you created for your MYSQL server.

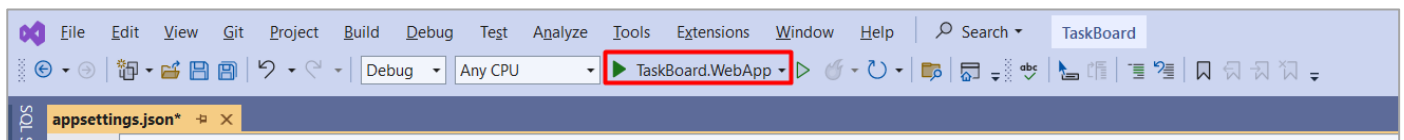


The screenshot shows the 'appsettings.json' file in a code editor. The file contains the following JSON structure:

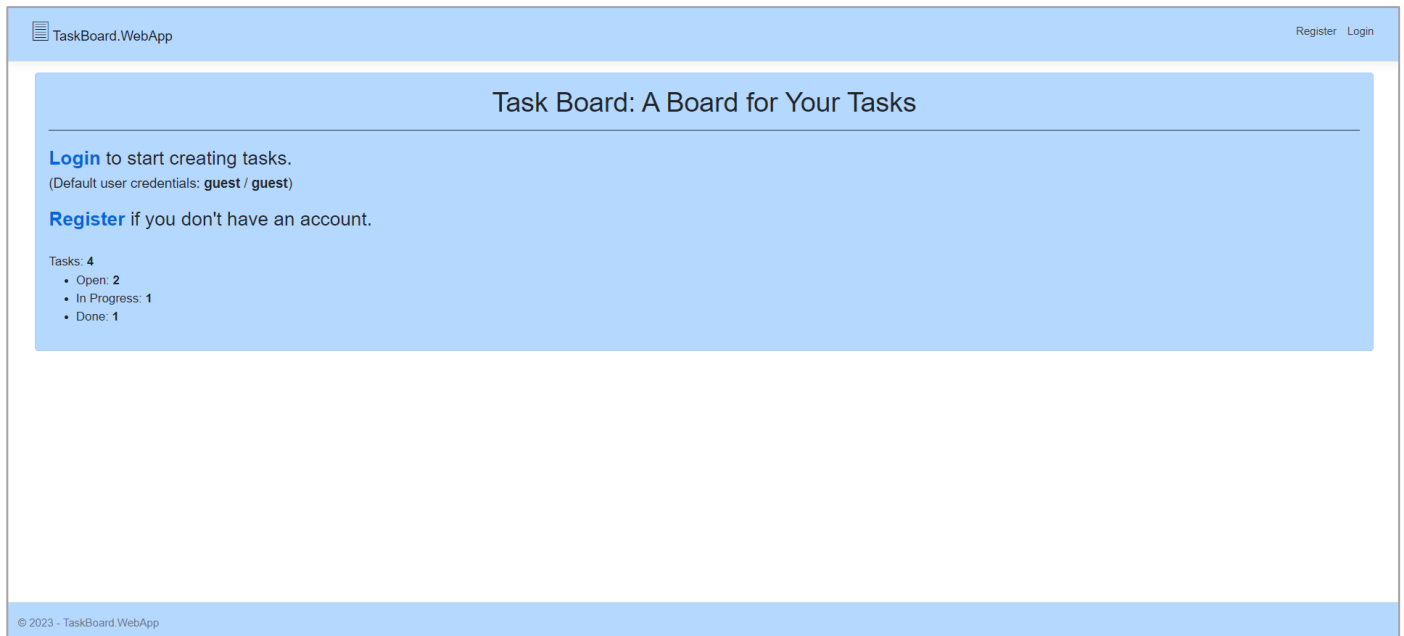
```
1 {
2   "ConnectionStrings": {
3     "Default": "server=localhost;port=3306;User ID=.....;Password=.....;Database=TaskBoard"
4   },
5
6   "Logging": {
7     "LogLevel": {
8       "Default": "Information",
9       "Microsoft": "Warning",
10      "Microsoft.Hosting.Lifetime": "Information"
11     }
12   },
13   "AllowedHosts": "*"
14 }
15
```

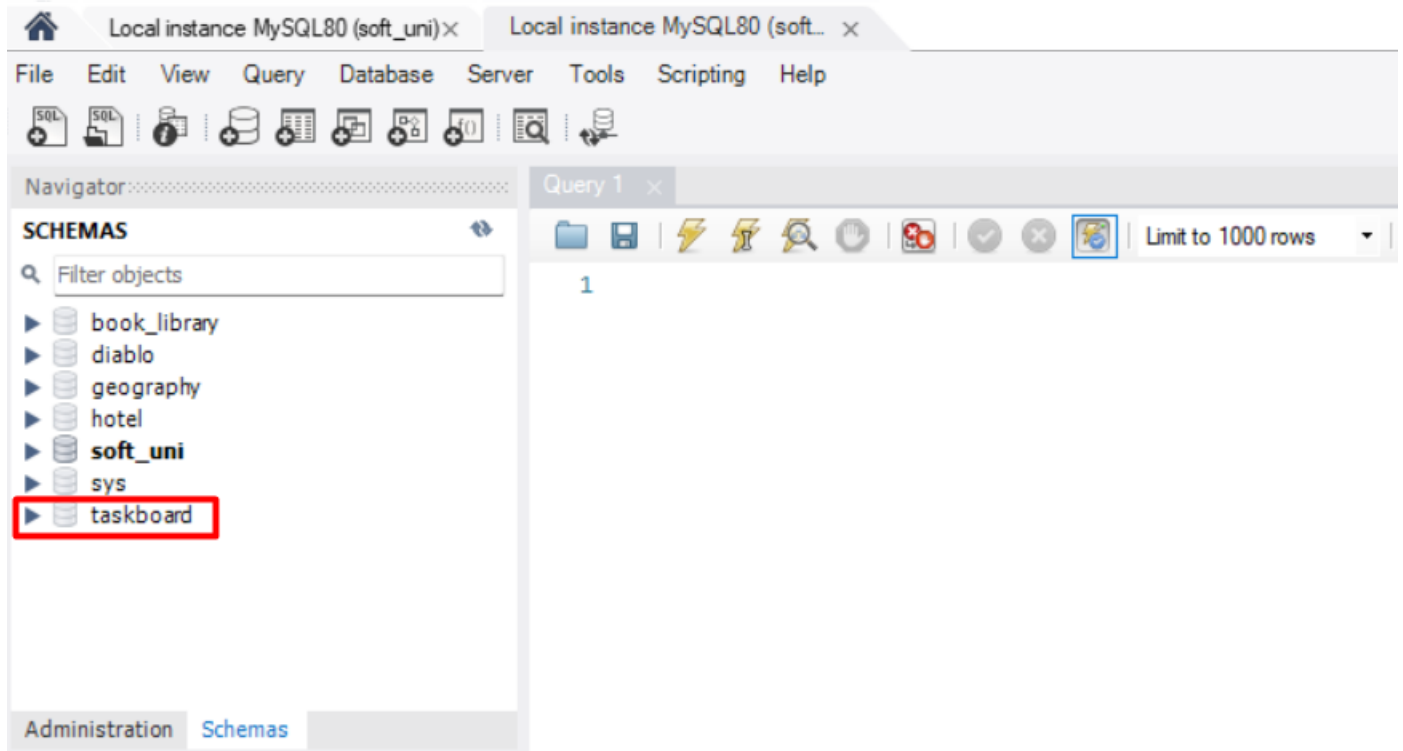
Two red arrows point to the 'User ID=.....' and 'Password=.....' fields in the 'Default' connection string, indicating where the user should replace the dots with their MySQL credentials.

8. When done, locate the play button and run the app



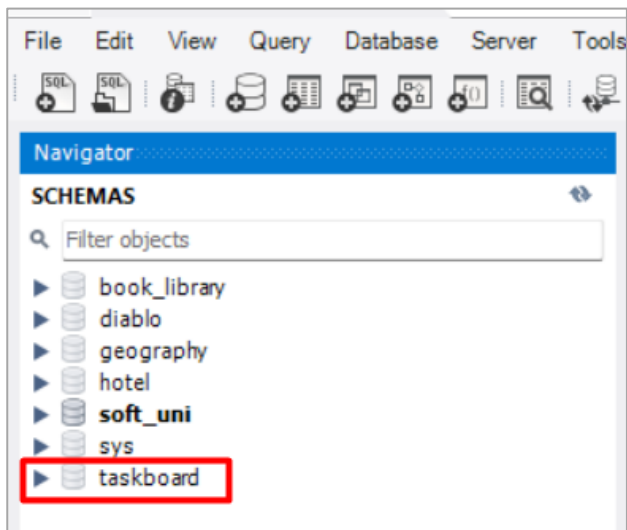
9. This will start the "good old" TaskBoard, locally on your machine.





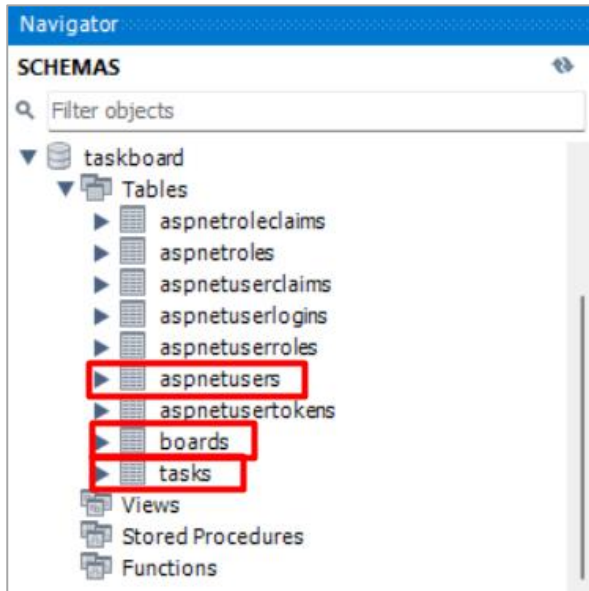
### 3. Browse the "taskboard" database

While the TaskBoard is running, open MySQL Workbench, connect to Server and observe the Databases (Schemas) that you have. You will notice that the taskboard schema is already created.



### 3.1. Explore the tables in the Schema

Mostly, we will use aspnetusers, boards, and tasks tables, but feel free to explore the other tables as well.

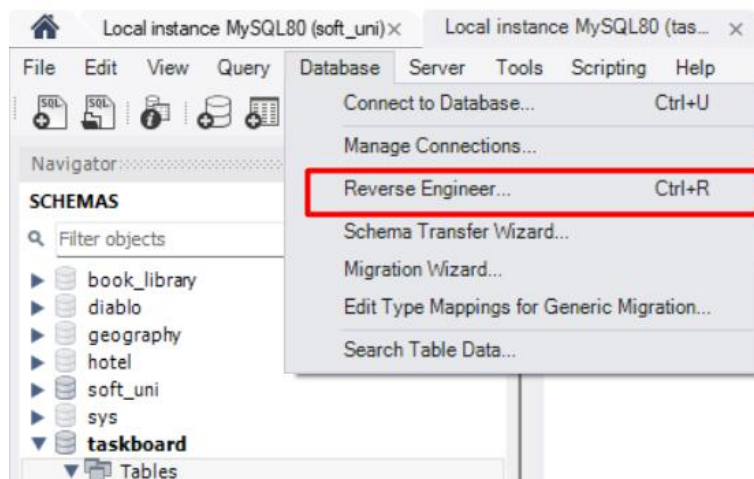


Try Schema Inspector (right click on the schema and find it in the context menu)

Try Alter Table (right click on some table and find it in the context menu) to see how the table is structured, what columns it has, what constraints.

### 3.2. Try the Reverse Engineering Function

From the main menu, click Database → Reverse Engineering and follow the steps




**Set Parameters for Connecting to a DBMS**

→ NEXT

**Connect to DBMS and Fetch Information**

→ NEXT

**Select Schemas to Reverse Engineer**

 **Select the schemas you want to include:**


- ☐ book\_library
- ☐ diablo
- ☐ geography
- ☐ hotel
- ☐ soft\_uni
- ☒ taskboard

→ NEXT

**Retrieve and Reverse Engineer Schema Objects**

→ NEXT

**Select Objects to Reverse Engineer**

 ☒ Import MySQL Table Objects Show Filter

9 Total Objects, 9 Selected

☒ Place imported objects on a diagram

Back Execute > Cancel

→ Execute >

**Reverse Engineering Progress**

→ NEXT

**Reverse Engineering Results**

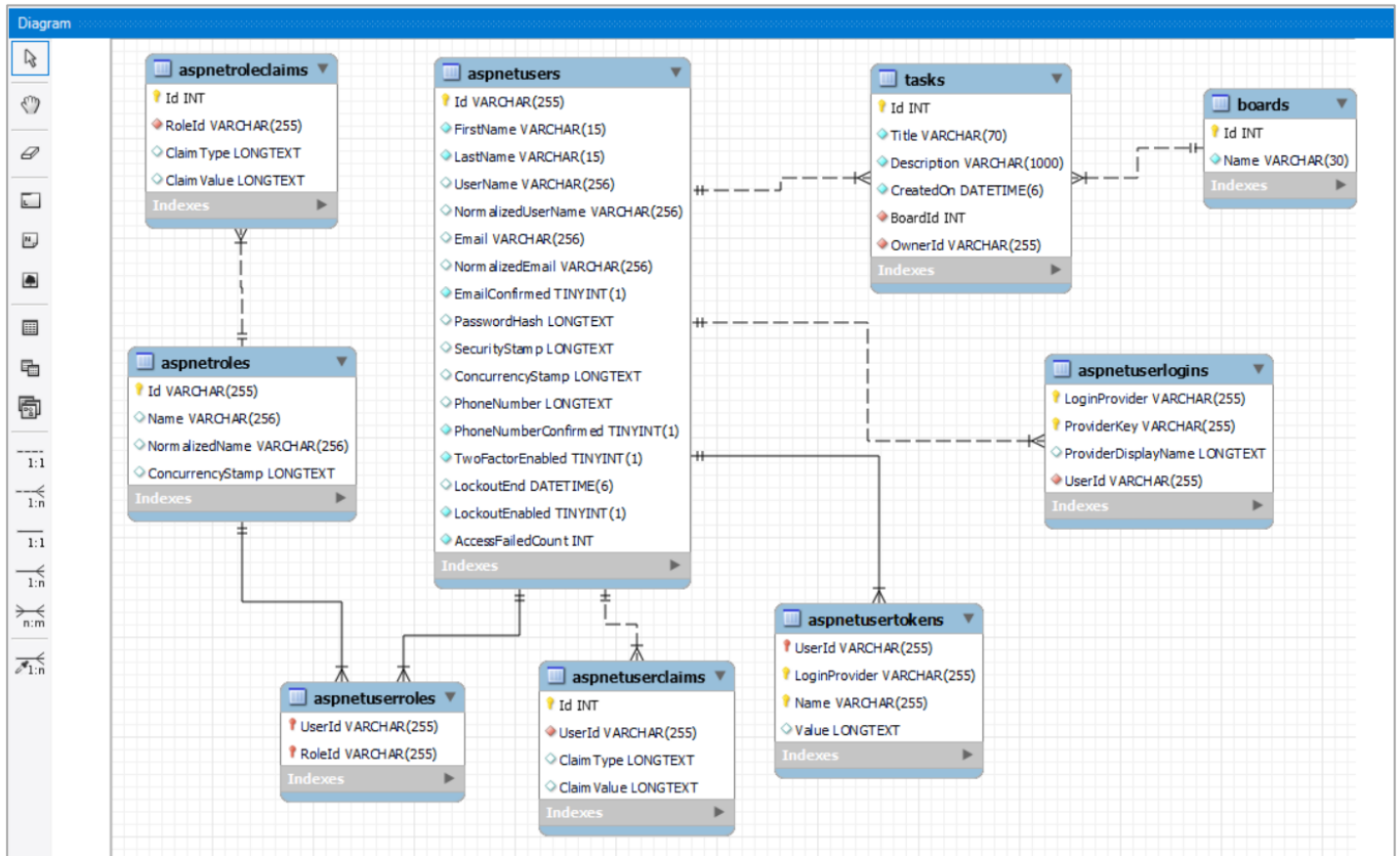
Summary of Reverse Engineered Objects:

- 9 tables from schema 'taskboard'

→ Finish

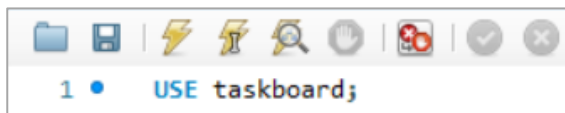
Don't worry if your diagram doesn't look quite like the picture, you can arrange each segment by dragging it.





## 4. Working with Taskboard Database

1. Create a new SQL tab for executing queries.
2. Execute the USE statement to use the taskboard db.



3. Execute SELECT all from aspnetusers to see how many users are registered.



By default, there is only one user.

Id	FirstName	LastName	UserName	NormalizedUserName	Email	NormalizedEmail
0627cf57-df1f-46e0-9f96-de4da31adce1	Guest	User	guest	GUEST	guest@mail.com	GUEST@MAIL.COM

4. Register a new user via the Web App

TaskBoard.WebApp

## Register

Create a new account.

Username

JuniorQA

Email

juniorqa@test.com

Password

\*\*\*\*\*

Confirm password

\*\*\*\*\*

First Name

Test

Last Name

Testov

Register

5. Go to the database and run the SELECT all statement again. See if the new user is present in the table.

Id	FirstName	LastName	UserName	NormalizedUserName	Email	NormalizedEmail
0627cf57-df1f-46e0-9f96-de4da31adce1	Guest	User	guest	GUEST	guest@mail.com	GUEST@MAIL.COM
e973279e-7209-496e-9b22-169931669e00	Test	Testov	JuniorQA	JUNIORQA	juniorqa@test.com	JUNIORQA@TEST.COM
NULL	NULL	NULL	NULL	NULL	NULL	NULL

See the Id number of the created user, copy it and paste it somewhere to use it a bit later.

6. Now, create a new task via the UI.

### Create Task

Title

TEST TASK

Description

TEST TASK Description

Board

In Progress

Create

7. Execute SELECT all on the task table to see if the new task is successfully created.

Id	Title	Description	CreatedOn	BoardId	OwnerId
1	Improve CSS styles	Implement better styling for all public pages	2023-01-05 14:26:31.340910	1	0627cf57-df1f-46e0-9f96-de4da31adce1
2	Android Client App	Create Android client app for the TaskBoard RE...	2023-02-24 14:26:31.341024	1	0627cf57-df1f-46e0-9f96-de4da31adce1
3	Desktop Client App	Create Windows Forms desktop app client for t...	2023-06-24 14:26:31.341031	2	0627cf57-df1f-46e0-9f96-de4da31adce1
4	Create Tasks	Implement [Create Task] page for adding new t...	2022-07-24 14:26:31.341032	3	0627cf57-df1f-46e0-9f96-de4da31adce1
11	TEST TASK	TEST TASK Description	2023-07-24 18:32:13.410780	2	e973279e-7209-496e-9b22-169931669e00
NULL	NULL	NULL	NULL	NULL	NULL

8. No let's try the reverse, to create task in the database and see if it will appear in the UI. In order to do that you will use the INSERT statement, and you will have to populate the following fields – Title, Description, CreatedOn, BoardId, OwnerId.

For CreatedOn – you can use NOW()

For BoardId – you can run the SELECT all statement on boards table, to find the id of the boards, or just easy guess that Boards Ids are as follows:

	Id	Name
▶	1	Open
	2	In Progress
	3	Done
★	NULL	NULL

For OwnerId you have to use the long Id, that we advised you to store somewhere for later. 😊

**Actually, let's create three tasks at the same time!**

```
INSERT INTO tasks (Title, `Description`, BoardId, CreatedOn, OwnerId)
VALUES ('TEST TASK 2', 'TEST TASK Description', '2', NOW(), 'e973279e-7209-496e-9b22-169931669e00'),
('TEST TASK 3', 'TEST TASK Description', '3', NOW(), 'e973279e-7209-496e-9b22-169931669e00'),
('TEST TASK 4', 'TEST TASK Description', '1', NOW(), 'e973279e-7209-496e-9b22-169931669e00');
```

9. Refresh the Web App to check in the UI if the tasks are created.

Task Board					
Open		In Progress		Done	
Title	Improve CSS styles	Title	Desktop Client App	Title	Create Tasks
Description	Implement better styling for all public pages	Description	Create Windows Forms desktop app client for the TaskBoard RESTful API	Description	Implement [Create Task] page for adding new tasks
Actions	View	Actions	View	Actions	View
Title	Android Client App	Title	TEST TASK	Title	TEST TASK 3
Description	Create Android client app for the TaskBoard RESTful API	Description	TEST TASK Description	Description	TEST TASK Description
Actions	View	Actions	View Edit Delete	Actions	View Edit Delete
Title	TEST TASK 4	Title	TEST TASK 2		
Description	TEST TASK Description	Description	TEST TASK Description		
Actions	View Edit Delete	Actions	View Edit Delete		

10. What will happen if we try to create a new board? We don't have that option from the UI. Let's try that. Execute again INSERT INTO on boards. You only need the name of the board.

```
INSERT INTO boards (`name`) values ('On Hold');
```

11. Refresh the app and Voila! The new board was created.

Open		In Progress		Done	
Title	Improve CSS styles	Title	Desktop Client App	Title	Create Tasks
Description	Implement better styling for all public pages	Description	Create Windows Forms desktop app client for the TaskBoard RESTful API	Description	Implement [Create Task] page for adding new tasks
Actions	<button>View</button>	Actions	<button>View</button>	Actions	<button>View</button>
Title	Android Client App	Title	TEST TASK	Title	TEST TASK 3
Description	Create Android client app for the TaskBoard RESTful API	Description	TEST TASK Description	Description	TEST TASK Description
Actions	<button>View</button>	Actions	<button>View</button> <button>Edit</button> <button>Delete</button>	Actions	<button>View</button> <button>Edit</button> <button>Delete</button>
Title	TEST TASK 4	Title	TEST TASK 2		
Description	TEST TASK Description	Description	TEST TASK Description		
Actions	<button>View</button> <button>Edit</button> <button>Delete</button>	Actions	<button>View</button> <button>Edit</button> <button>Delete</button>		
On Hold					

12. Let's add some tasks to it.

```
INSERT INTO tasks (Title, `Description`, BoardId, CreatedOn, OwnerId)
VALUES ('TASK ON NEW BOARD 1', 'Description 1', '4', NOW(), 'e973279e-7209-496e-9b22-169931669e00'),
('TASK ON NEW BOARD 2', 'Description 2', '4', NOW(), 'e973279e-7209-496e-9b22-169931669e00'),
('TASK ON NEW BOARD 3', 'Description 3', '4', NOW(), 'e973279e-7209-496e-9b22-169931669e00');
```

13. Refresh and here they are.

On Hold	
Title	TASK ON NEW BOARD 1
Description	Description 1
Actions	<button>View</button> <button>Edit</button> <button>Delete</button>
Title	TASK ON NEW BOARD 2
Description	Description 2
Actions	<button>View</button> <button>Edit</button> <button>Delete</button>
Title	TASK ON NEW BOARD 3
Description	Description 3
Actions	<button>View</button> <button>Edit</button> <button>Delete</button>

14. What will happen if we try to delete a task. Let's try to delete the last task that we created by name.

```
DELETE FROM tasks where Title = 'TASK ON NEW BOARD 3';
```

Even thou the query is correct; we get an error:

Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column. To disable safe mode, toggle the option in Preferences -> SQL Editor and reconnect.

This means that MySQL server is running in "safe update mode," which restricts certain types of dangerous or accidental DELETE or UPDATE statements without a WHERE clause that uses a KEY column to identify specific rows. In order to delete the task, we have to use a KEY column, in our case the exact Id, or Where clause with at least two conditions pointing to the exact record. Let's try both.

15. First, we have to find the Id of the task that we want to delete. Let's find it using the task's Title.

```
SELECT Id from tasks where Title = 'TASK ON NEW BOARD 3';
```

16. The returned Id in with us is 20, but it might differentiate with you, depending on which task you are searching for. Let's try to delete it again.

```
DELETE FROM tasks where Id = '20';
```

17. When we execute SELECT all, as you can see, there is no Task with Id 20.

Id	Title	Description	CreatedOn	BoardId	OwnerId
1	Improve CSS styles	Implement better styling for all public pages	2023-01-05 14:26:31.340910	1	0627cf57-df1f-46e0-9f96-de4da31adce1
2	Android Client App	Create Android client app for the TaskBoard RE...	2023-02-24 14:26:31.341024	1	0627cf57-df1f-46e0-9f96-de4da31adce1
3	Desktop Client App	Create Windows Forms desktop app client for t...	2023-06-24 14:26:31.341031	2	0627cf57-df1f-46e0-9f96-de4da31adce1
4	Create Tasks	Implement [Create Task] page for adding new t...	2022-07-24 14:26:31.341032	3	0627cf57-df1f-46e0-9f96-de4da31adce1
11	TEST TASK	TEST TASK Description	2023-07-24 18:32:13.410780	2	e973279e-7209-496e-9b22-169931669e00
15	TEST TASK 2	TEST TASK Description	2023-07-24 18:57:31.000000	2	e973279e-7209-496e-9b22-169931669e00
16	TEST TASK 3	TEST TASK Description	2023-07-24 18:57:31.000000	3	e973279e-7209-496e-9b22-169931669e00
17	TEST TASK 4	TEST TASK Description	2023-07-24 18:57:31.000000	1	e973279e-7209-496e-9b22-169931669e00
18	TASK ON NEW BOARD 1	Description 1	2023-07-24 19:23:09.000000	4	e973279e-7209-496e-9b22-169931669e00
19	TASK ON NEW BOARD 2	Description 2	2023-07-24 19:23:09.000000	4	e973279e-7209-496e-9b22-169931669e00
NULL	NULL	NULL	NULL	NULL	NULL

18. Let's try to delete another task, but not by Id, but using the other method with WHERE clause with two conditions, pointing to exact record. Let's delete TEST TASK 3.

```
DELETE FROM tasks WHERE Title = 'TEST TASK 3' AND BoardId = 3;
```

And it's gone!

Id	Title	Description	CreatedOn	BoardId	OwnerId
1	Improve CSS styles	Implement better styling for all public pages	2023-01-05 14:26:31.340910	1	0627cf57-df1f-46e0-9f96-de4da31adce1
2	Android Client App	Create Android client app for the TaskBoard RE...	2023-02-24 14:26:31.341024	1	0627cf57-df1f-46e0-9f96-de4da31adce1
3	Desktop Client App	Create Windows Forms desktop app client for t...	2023-06-24 14:26:31.341031	2	0627cf57-df1f-46e0-9f96-de4da31adce1
4	Create Tasks	Implement [Create Task] page for adding new t...	2022-07-24 14:26:31.341032	3	0627cf57-df1f-46e0-9f96-de4da31adce1
11	TEST TASK	TEST TASK Description	2023-07-24 18:32:13.410780	2	e973279e-7209-496e-9b22-169931669e00
15	TEST TASK 2	TEST TASK Description	2023-07-24 18:57:31.000000	2	e973279e-7209-496e-9b22-169931669e00
17	TEST TASK 4	TEST TASK Description	2023-07-24 18:57:31.000000	1	e973279e-7209-496e-9b22-169931669e00
18	TASK ON NEW BOARD 1	Description 1	2023-07-24 19:23:09.000000	4	e973279e-7209-496e-9b22-169931669e00
19	TASK ON NEW BOARD 2	Description 2	2023-07-24 19:23:09.000000	4	e973279e-7209-496e-9b22-169931669e00
NULL	NULL	NULL	NULL	NULL	NULL

19. Try a few more:

- Try to filter only Tasks that has the word "task" in the name.
- Compare the result with the same search made in the Web App.
- Try to delete the user and see what happens (hint: you will need his id). See if the tasks created by this user still exist.

- Try to update the name of TEST TASK 4 (hint: The UPDATE statement works similar to DELETE, e.g. you will need the Id of the task or at least two conditions pointing directly to the task you want to delete).