Exercises - WIMTACH

Action Items: Check List

Wacht and follow:
https://www.youtube.com/playlist?list=PLJAUzIc5d2XXWcE596lUMHrDcbUcoP4qK

If you dont know something Google It.

- Write down what is new for you and write it here.

  - I did not work with AspnetBoilerPlate before. With this exercise I got to learn about it now.
  - Docker is not new to me as I have used docker for some of my project, but did not work that much with docker.

Follow On the backend and the frontend instructions.

- For example: one but not limited: create your **postgreSQL** container. Docker install, Connect to the db, Start the Swagger, In the example I created the **Student.**

  - Followed all the instructions

Additional to the video instructions, you have to Create you own:

- **College** is the class: (Id, Name, Address, Description, GPS latitude, longitude, Contact Email, IsFoysalSleep).
- You have to do everything is was made for the student. It means that you will have an endpoint for update, create, delete, college in swagger.
- Update the student to have one **collegeId.**
- Database Migration Entityframework.
- Demonstrate how migrations works.
- IMPORTANT: create your own Repo. Use GitHub to commit your work. For one commit:

```
# What are we doing?

1. Start the PostgreSQL container
2. Quick Overview of the Backend
3. Download the ASP.NET Boilerplate
4. Update the project to use PostgreSQL
5. Migrate the PostgreSQL database
6. Disable Background Jobs feature

---

# What are we doing? - Continue...

```

```
7. Disable SignalR feature
8. Disable Multi-Tenancy feature
9. Create a new Entity and Add it to DBcontext
10. Add the Application Service
11. Check Swagger Endpoint
12. Test the CRUD functionality
13. BONUS - Show Audit logs and Generic Repository.
```

- I want to see the commits, it means, commit that shows the changes. One commit for steps.

  - **All the specific tasks are committed individually as instructed.**

Probably questions:
What is interface, abstract class, how is the Repositry working, IRepositry<>,

Interface: Interface is a way to define or declare a set of methods and properties and it hides the implementation of these methods from the user and only exposes the functionality.

Abstract Class: Abstract class the blueprint for other classes and instances can not be created of an abstract class. It can contain both **-** the declaration of methods(Abstract methods) and The concrete methods (with implementation). It can also contain fields and properties

Repository:

Code repository works for version control, collaboration and we can keep all the changes history by using core repository (Ex. Github and Gitlab are the popular site for storing Git Repository and Project management)

Data Repository: It is Centralized storage location for Entity and Data., Data repository manage structured and unstructured data. Data operations and access control also managed by Data Repository

IRepository/ Generic Repository:
Generic repository actually manages the commonly used data operations of Entities and It increases usability and user don't have to repeat the same methods for all Entities

IMPORTANT: For the Frontend and the Backend.
Preference: Backend