**DatingApp**

**BackEnd**

1. SET UP
   1. Using the CLI to create the App

* Commands :

PS C:\MyGitRepos\DatingApp> dotnet --info

new -h (give all info of what can be done with the “new” command)

new list (give list of available templates to create projects)

new sln (create a solution file and if we don’t give the name it will chose the same as the folder name)

ls (check what’s in the folder)

new webapi -n API (create a new project with name API)

sln add API/ (add the project in the API folder)

dotnet run -lp https (to run the app on https server)

dotnet restore (reload the app after adding or removing packages)

dotnet watch run (run the app)

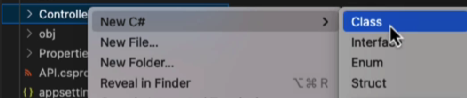
dotnet watch -h (check available commands, can stop the hot reload if it’s not working well)

* 1. Extensions to install In VS Code
* Material Icon theme (icons looking better)
* C# (if you want to work with C# on VS code)

Go to extension settings -> enable Async completion

Enable Import completion

Organize imports on format

* C# Extensions from JosCreativ
* 
* NuGet gallery (to be able to install NuGet packages)
* In the NuGet gallery find and install
  + Microsoft.EntityFrameworkCore.Design
  + Microsoft.EntityFrameworkCore.Sqlite

1. Install Entity Framework

Check with “dotnet tool list -g” if **EF** is installed and if not In the browser : nuget.org/packages/dotnet-ef -> copy the command “dotnet tool install --global dotnet-ef --version 7.0.1” and use it in terminal to install EF

* 1. VS Code set up

1. File -> Auto Save (enable auto save)
2. TO REFRESH VS CODE : Ctrl+Shift+P -> Ctrl+R (Reload Window)
3. TO Exclude unused folders from solution explorer : bin and obj => File->Preferences->Settings-> exclude-> Add pattern “ \*\*/bin ” ->Ok (same for \*\*/obj)

**Context, Controllers**

1. First Migration

dotnet ef migrations add InitialCreate -o Data/Migrations

*TO REMOVE LAST MIGRATION :* ***dotnet ef migrations remove***

1. DataBase

dotnet ef database update

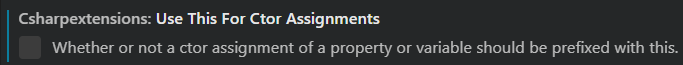
* DB is created, but file can’t be read, to manage that , in extensions -> SQLite (install the first one) -> Ctrl+Shift+P -> SQLite: Open Database

SQLite Explorer -> Users -> New Query [Insert] (to fill the table with data) -> run the query

1. Create Users Controller

* If you want to put \_ when inject by default in the constructor -> In the extension “C# Extensions” -> Extension Settings ->

Text

Description automatically generated 

**Source control GIT**

* Has to be in the main folder and in the terminal tap “ git init “ command
* “dotnet new list” to check available templates -> dotnet new gitignore (to create the gitignore file) .
* Add the appsettings.json file in gitignore file, we don’t want to add this file In Git, because will have secret info in it. To do so hover over the file -> Add file to .gitignore
* We want to create the global.json file : dotnet new globaljson
* Stage changes and make first commit
* Create new Repo in GitHub

**FrontEnd**

1. Commands

* **Check version of node.js**

**node** –version

**npm** –version

* **Install Angular CLI and check version**

**npm** install -g @angular/cli@14 **ng** version

* **Create the new Angular app and run it**

**ng** new client **ng** serve

1. Extensions to install

* Angular Language Service

1. VS code set up

* Productivity features

File -> Preferences -> Settings -> write “bracket”

Text

Description automatically generated 

Text

Description automatically generated Graphical user interface, text

Description automatically generated with medium confidence

1. Making Http request in angular

* Add  
  Text

  Description automatically generated

Check commit [Add Http request to API](https://github.com/ZackoYo/DatingApp/commit/4f9f9d7580b4a690c1e3530b54ccf03a2c650377) = 4f9f9d7580b4a690c1e3530b54ccf03a2c650377

1. Solve CORS issue

In Program add  above 

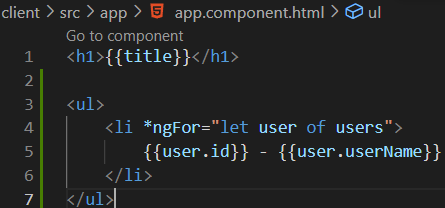
Then under  add



Check commit [Fix CORS policy issue](https://github.com/ZackoYo/DatingApp/commit/9b1f78c7d15160fa70c37b3678ac9ff6e155aeb0)

1. Displaying the fetch users in the browser

Have to be careful it’s case sensitive



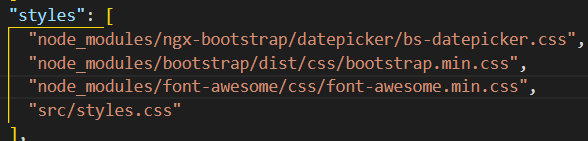
1. Install ngx-bootstrap and bootstrap5

**npm** install [ngx-bootstrap@9.0.0](mailto:ngx-bootstrap@9.0.0) **npm** install bootstrap@5

1. Install font-awesome for fancy icons

**npm** install font-awesome

* **Have to add** styles files in angular.json and remember that the last file overrides those before!!!



1. Using Https in Angular

Go to C:\MyGitRepos\DatingApp\Resources\StudentAssets\StudentAssets\generateTrustedSSL

Install the certificate and copy the certificate and key files, then go in VS code , on client -> reveal in File Explorer, create folder ssl and paste the files. Then to make Angular use these files go into angular.json file and add this Text

Description automatically generated

Have to change “http” to “https” in C:\MyGitRepos\DatingApp\API\Program.cs



Check commit [add ssl certificate and using https](https://github.com/ZackoYo/DatingApp/commit/15b50782e7b968b1bbcb87b6297c7b27f1c84c7d)

1. Password hash

* Add two properties to AppUser class : PasswordHash and PasswordSalt
* Add new migration : use command : dotnet ef migrations add UserPasswordAdded
* Update DB : dotnet ef database update

1. Add base API Controller : check commit [Add base API controller](https://github.com/ZackoYo/DatingApp/commit/26b946ff3cadc57de4b30a548b5ecc5f469fb4d0)
2. Add AccountController : commit [Add AccountController and register endpoint](https://github.com/ZackoYo/DatingApp/commit/353eadf98ede70fb512669e79d44ad1cd96f1db7)
3. Debbuging the API with VSCode

* In folder .vscode check for launch.json (if not there in command palette “Ctrl+Shift+P” write .Net and choose “Generate Assets for build and Debug” and that will generate this file)
* For debugging are two options ,
* **first** *.Net Core Launch (web),* debbuger is launched when app is runned
* **second** *.Net Core Attach , have to run the API using this command*

PS C:\MyGitRepos\DatingApp\API> dotnet watch --no-hot-reload , then in debugger choose .Net Core Attach, a pop up will show, search for API and choose API.exe , then send request from Postman and the program will stop on the first break point.

1. Commit : [Add RegisterDto and UserExists method](https://github.com/ZackoYo/DatingApp/commit/679aee8f7b0c4cfddc76089e103e825493888ac0)
2. Validation if the fields are empty : when using the attribute [ApiController] , we just need to put attribute [Required] over Username and Password in the RegisterDto
3. Add login endpoint and DTO : commit [Add login endpoint](https://github.com/ZackoYo/DatingApp/commit/838455e975b93c8dcddaf4639c31b6b765c34b28)
4. JWT : JSON Web Token – Industry standard for tokens (RFC 7519) – Self-contained and can contain credentials, claims, other info. It’s a long string of three parts, first is header of the token and this contains the algorithm used to encrypt the signature in the third (the key is kept on the server) part of the token and the token what it is, second part is the payload where the data is, like credentials, claims.

|  |
| --- |
|  |
|  |

1. Add token service and Create Token logic : commit [Add Token Service and Create Token logic](https://github.com/ZackoYo/DatingApp/commit/e9c37a21c665acd6de28a9be368c8f547e5bdfda)

It’s a Http service and when we register it in the container the best scope to choose is Scoped.

* In NuGet Gallery have to download System.IdentityModel.Tokens.Jwt

1. Commit [Add UserDto and return the token](https://github.com/ZackoYo/DatingApp/commit/b98ccd31fd513417831bb146a7b7aa91692ac32c)
2. Add authentication middleware

Have to install Microsoft.AspNetCore.Authentication.JwtBearer

Positioning of

app.UseAuthentication();

app.UseAuthorization();

is verry important , always after app.UseCors and before app.MapControllers();