**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)

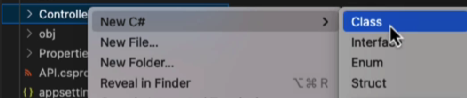
dotnet watch run (run the app)

* 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

1. DataBase

dotnet ef database update

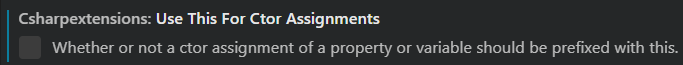
* DB is created, but file can’t be read, to manage that , in extensions -> SQLite (install the first one) -> Ctrl+Shift+P -> SQLite: Open Database

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