**HabitsTracker**

Application to track your daily habits

Content:

1. Intro
2. Application setup
3. Used technologies
4. Instruction manual
5. Challenges during development
6. Final thoughts
7. **Intro**

Habits Tracker is an application made using ASP.NET Core MVC architecture. Application is designed with simplicity in mind and for everyday usage of users habits with focus on easeage of use with the mindset of “less is more”.

1. **Application setup**

Download the original sourcecode/files from the repo and to run it directly from the development environment. In the case the user has to have the following programs installed on their computer:

* Visual Studio 2022 Community Edition
* .NET 8.0 SDK
* Microsoft SQLExpress server setup (LocalDB)

Click on green “Code” button on Github, select “Download .zip”, unzip it, run “HabitsTracker.sln” from the folder and wait for the solution to open in Visual Studio 2022. Upon everything being loaded, all there’s left to do is to run the following commands:

*Add-Migration FirstMigration*

After successfully ran, the following command also needs to be ran:  
  
*Update-Database*

And you’re good to go. That’s the more complex way of running it.

The easier way of running it is to download .exe

1. **Used technologies**

For backend part of the application, following technologies were used:

- ASP.NET Core 8.0

- EntityFrameworkCore

- SQL server

For frontend part of the application, following technologies were used:

* HTML/CSS
* Tailwind CSS
* Razor Views
* Bootstrap

1. **Usage instructions**

Upon running the application you are greeted with a welcome screen. Press “Register”. After registering, use the login credentials to Login.

After logging in you’re greeted with a simple habits UI. Enter the name of habit you want to be tracking daily. Optionally, you can type in hour/minute of the day, in the case the habit is put on top of the screen above the habits which don’t have time of the day associated with them and is ordered in a descending order with other habits that have time attached to them.

Pressing the radial button next to the habit marks it as “done” for the day. Upon the day ending (date changing), all of the habits are “reset” and they’re stored in a calendar you can access by clicking the name of the habit. If the habit didn’t exist on certain dates the calendar will do nothing, if the habit existed and was done that day, the date will glow with green color, if the habit existed and wasn’t done that day, the date will glow with red color. Pressing the habit again will collapse the calendar and hide it.

1. **Challenges during development**

Editing pages that were originally supplied with the project was easy and it was super fast to set up a functional project that looks visually distinctive from the basic projects offered by the templates. Controllers also make development flow rather easy and make it make “sense” when you’re stuck somewhere.

What was super annoying was to implement the strikethrough function. I had the brilliant idea to use JavaScript and couldn’t get it to work, then ended up using ordinary HTML form submission+page refresh that most of the tutorials for MVC suggest. I had a similar experience with calendar function however it initially worked with JavaScript but I remembered we shouldn’t do that for this project and should focus more on the backend instead of solving problems on the frontend so I went back to a simpler approach after google-fuing and furiously scavenging code for a while. I couldn’t get month/year to show up on calender for a while and it drove me nuts.

I wish I had more time to explore other options, such as implementing a statistics page or to add habit categories or allow the specific comments or maybe even explored making it accessible from a mobile app, but I’m not nearly as versed in .NET yet as I should be so this was a big step for start. Once I get more time I will continue working on this project for myself and perhaps even publish a fully polished version.

1. **Final thoughts**

Working in .NET and following all the rules is definitely more intensive than using standard technology stacks for which there are numerous tutorials and everything seems to be “easier” and more streamlined, but I can definitely see why .NET is considered as powerful as it is and is something I’m looking forward to learn more of.

I had fun making this project even though it was done in the nick of time and I look forward expanding upon in it even further.