**说明**

从这里开始我真正体会到了.Net的学习曲线是多么的陡峭。

使用工具/依赖项：VS2022 Community，SQL Server Express，SSMS

项目文件：Blazorapp2.sln

所有代码都是自己写的（除了数据库迁移那一部分）

网站界面上相当简陋，但是对直到购物车功能的完成度都还不错。

生成订单后的结账物流等功能未实现。

数据库里残留一些测试数据，应该不影响。

测试之前务必完成数据库的迁移！具体事项请打开sql文件文件夹

进入网站将强制导航至signup界面，可以很方便的注册或登录。（注意字符限制）

然后可以进入其他页面。

具体特性（不是bug是特性：））可以自己体验。

一点点感想：

看到要做网页其实心已经凉了一半了（x）

还记得当时和老师抱怨说webform基本废掉，以后必须用别的方法来做，必须学习html然后老师说：那个学习成本很高啊！实在是没想到这天来的如此之快。

在vs的asp.net界面踌躇一会之后选择了自夸很方便的blazor：）

新建blazor项目便是一个示例，一开始整个人都是蒙的。

跟了ms网站上的几个示例，依然感觉不明觉厉。

于是决定从基础的开始学。

以前我对网页的概念仅限于html，于是便从html开始学习。学了学html便转向css，然后才意识到这仅仅是学了一点前端设计罢了☹

花了好几天之后觉得差不多会用一点html和css就差不多了，开始正式转向razor标记语言的学习。逐渐进入后端之后越来越感觉自己要学的东西浩如烟海，内心相当绝望：（

终于稍微会设计一点razor页面，差不多明白了razor标记语言是怎么一回事，会写一点页面逻辑了，我意识到绕不开数据库的使用，便开始研究怎么连接数据库。

两个备选方案：.net EF 和ADO.Net 毫不犹豫的选择了看起来封装程度更高的前者。

但是光是搞明白.Net EF也花了我好几天。一开始想使用内存数据库省的装别的数据库软件，后来感觉不行，就安排上了sql server全家桶。这时候终于发现了靠谱的教程，差不多明白了如何使用。建表迁移一气呵成：）

然后就是疯狂的码：）又花了我相当长时间才终于完成这个看起来还是比较简陋的版本。中间有着大量的debug，实验，以及搜索和学习，甚至还包括数据库结构的修改。有些奇怪的bug使我了解了许多奇怪的特性，比如数据库迁移需要项目能够build：（

比如一个小小的输入结构，便花了我不少时间。一开始试图使用html自带的原始的input，但是研究了不少时间也搞不明白它的逻辑。这时候发现了ms提供的好东西：EditForm。虽然学习这个又花了我不少时间，还第一次了解了attribute的使用，但是总感觉是值得的。

又如page上方的三个inject，每一个背后都是不少时间的学习。光是有点明白inject是个什么概念就花了我不少时间。为此还被迫学习了async，总感觉自己摸到了线程世界的一角。

无论如何，做完这样一个项目成就感还是很强的。在一个自己完全不熟悉甚至几乎是毫无了解的领域学习了许久。

光是从语言上涉及的就包括：HTML,CSS,C#,RAZOR,SQL,Javascript

我觉得这个项目的难点全在学习曲线上了：）也算是收获颇丰吧！

看完有啥建议或者意见可以来git哦!网站界面右上角点击github即可到达。或者用下面的链接：<https://github.com/inoriace/firsttry>