НП “Обучение за ИТ Умения и кариера“

Име на проекта: ,,Restaurant System”

Екип: Александър Девинизов, Христо Костадинов

Хасково 2023г.

1. Цел на проекта:

Целта на проекта е да се създаде програма, която да наподобява на програмите за управление на ресторанти. Използвайки проекта, човек може да добавя, изтрива, взема и да актуализира информация за служители, работна позиция, поръчки, маси и продукти в базата данни.

2. Разпределение на ролите:

Работата и разпределението на задачите е 60% (Александър) на 40% (Христо) заради проблем с неуточнение.

3. Основни етапи в реализирането на проекта:

Основните етапи през които преминахме са 4 на брой:

Първи етап - първите стъпки бяха измисляне на тема, създаване на проекта и създаване на Repository в GitHub.

Втори етап – имахме проблем с неуточнение, но се разпределихме кой на какво ще работи по проекта. Към края се сблъскахме с проблем при добавянето на на данни за продукти и поръчки, но го оправихме чрез преглеждане на връзките по между им и внимателно добавяне на данни.

Трети етап – в този етап тествахме програмата, добавихме XML коментари за методите и редактирахме текстове и символи да изглежда по-добре. Опитахме да направим компоненти тестове за методите, но правеше промени по базата от данни и затова решихме да не ги правим.

Четвърти етап – последният етап на проекта беше създаване на ASP.NET проект.

4. Реализация:

- Използвани технологии:

EntityFrameworkCore

EntityFrameworkCore.SqlServer

EntityFrameworkCore.Design

EntityFrameworkCore.Tools

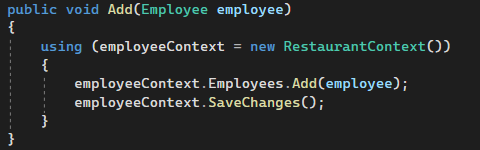
- Описание на методите и скрийшоти на приложението

Бизнес слой:

Всички бизнес класове съдържат еднакви методи, които отговарят за класа си.

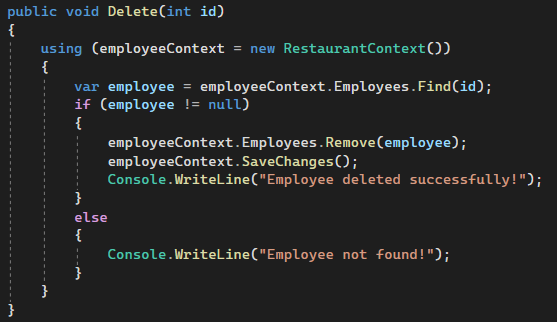
+ void Add(Employee/Job/Order/Table/Product parameter)

Зададеният параметър (от типа Employee, Job, Order, Table или Product) се добавя в базата от данни



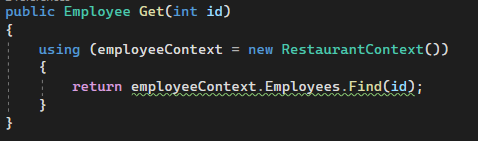
+ void Delete(int id)

Метода намира параметъра id и го изтрива ако обекта е намерен. Ако обектът не същесвува не прави нищо.



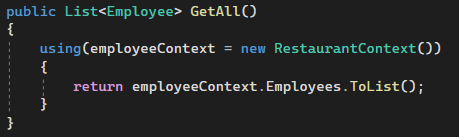
+ Employee/Job/Order/Table/Product Get(int id)

Зададения параметър бива открит и връщан от метода



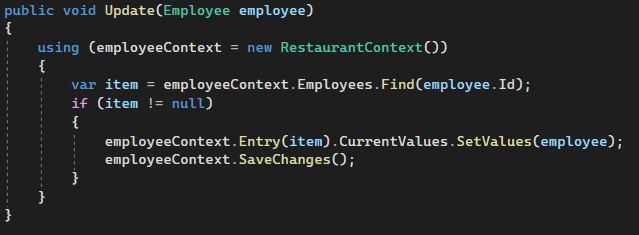
+ List<Employee/Job/Order/Table/Product> GetAll()

Метод, който връща всички обекти, запазени в базата от данни



+ void Update(Employee/Job/Order/Table/Product parameter)

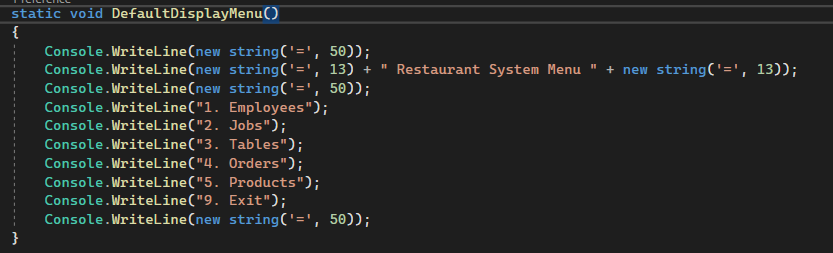
Метода намира същесвуващия параметър и променя свойствата му като накрая ги запазва



-Презентационнен слой:

Всеки клас има собствените си методи, за да изглежда по-подреден и да може да се намери лесно. След това методите биват обединени в главен дисплей.

Главния дисплей включва главното меню на програмата от където се навигира към всички останали менюта:



Главния дисплей включва безкраен цикъл, който води до друг цикъл от дугите класове, извикващ своя метод:



Всеки друг презентационнен клас съдържа цикъл, извикващ свой собствен метод.

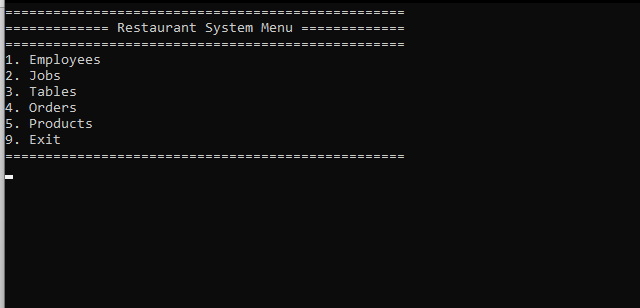
Пример EmployeesDisplay:

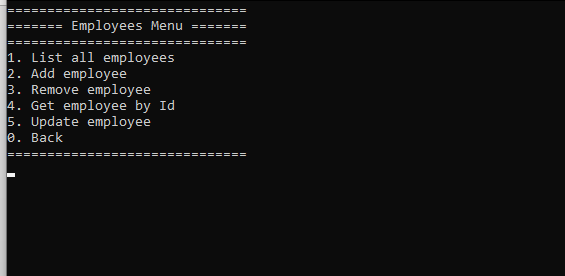


Менютата в другите класове са еднакви, но се отнасят за съответните класове.

Методите са подобни като се различават за променливите за класовете им.

Скрийншоти на приложението:

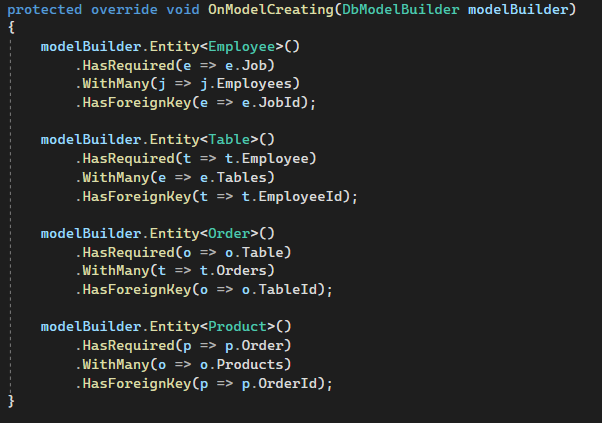




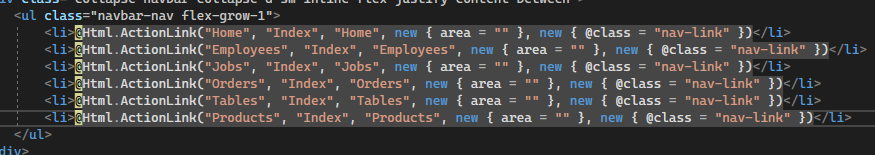
4.1 Реализация на ASP.NET проект

- След създаване не проектите RestaurantSystemASPNET, Data и Business на .NET Framework и добавяне на референции между тях, копирахме полетата за Employee, Job, Order, Table и Product от оригиналния проект и ги прибихме в проекта Data. Добавяме RestaurantSystemContext класа като използваме метода:

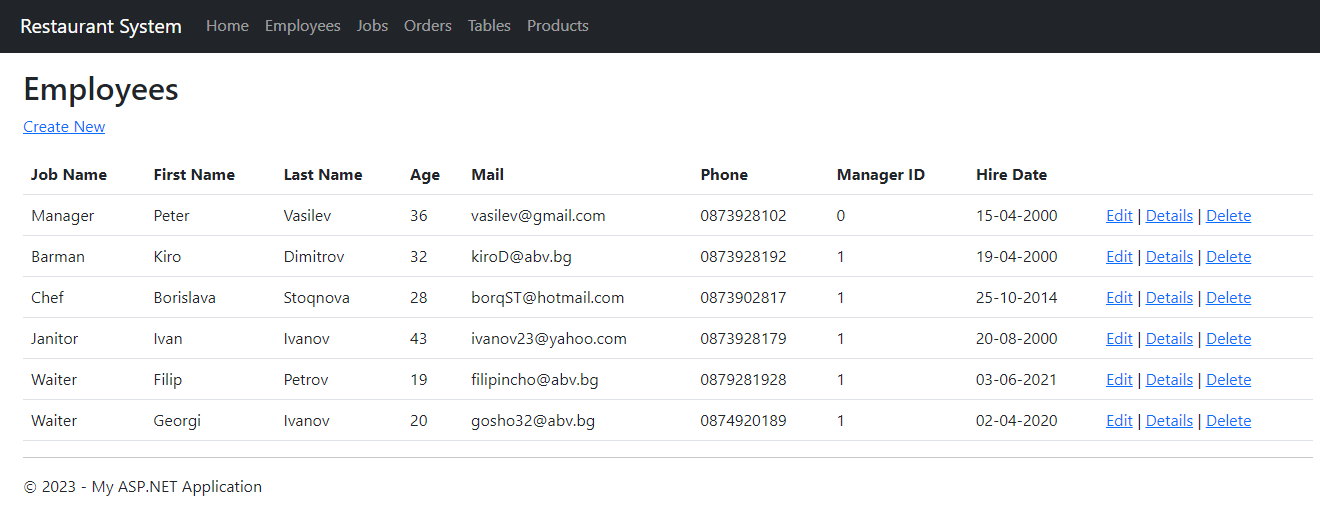
protected override void OnModelCreating(DbModelBuilder modelBuilder), за да може да направим връзките между таблиците.

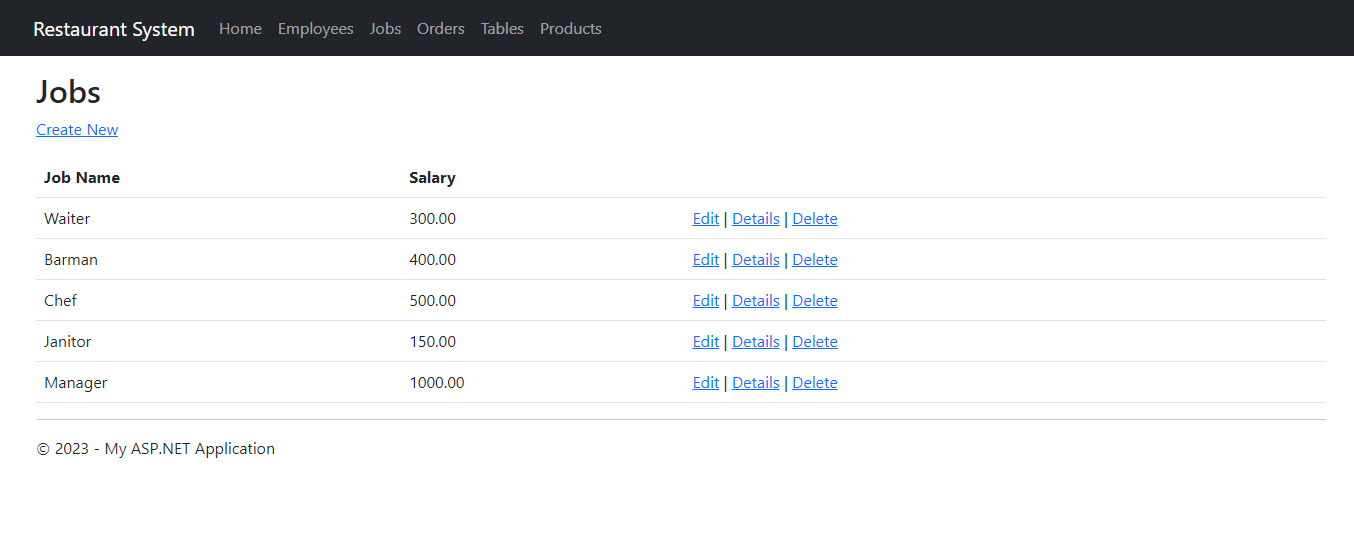


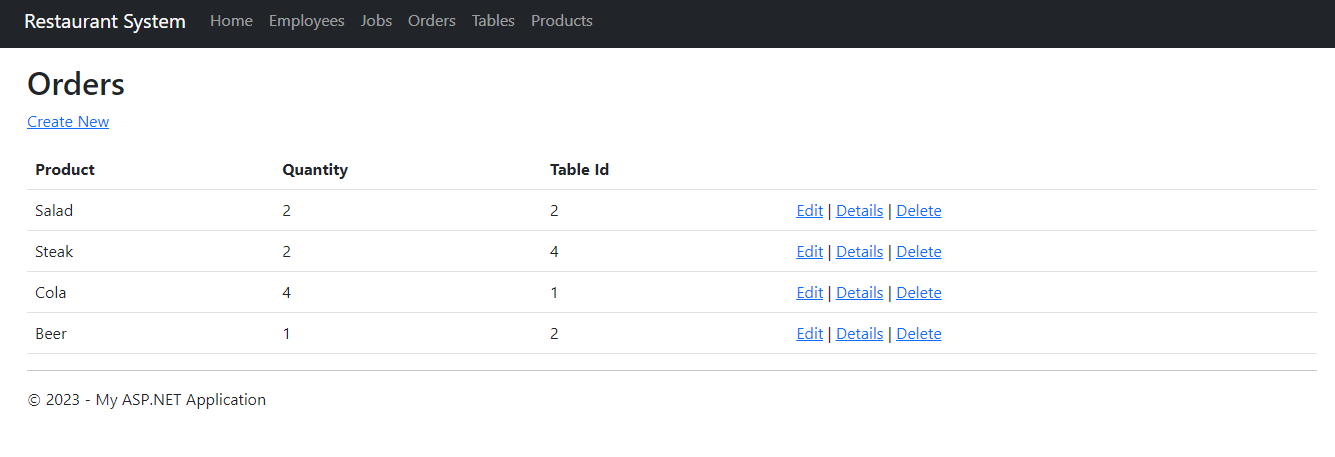
След това добавихме connectionString към web.config файла, за да може да се свържем към сървъра. Най-накрая добавихме контролери за всяка таблица и добавяме навигационни полета:

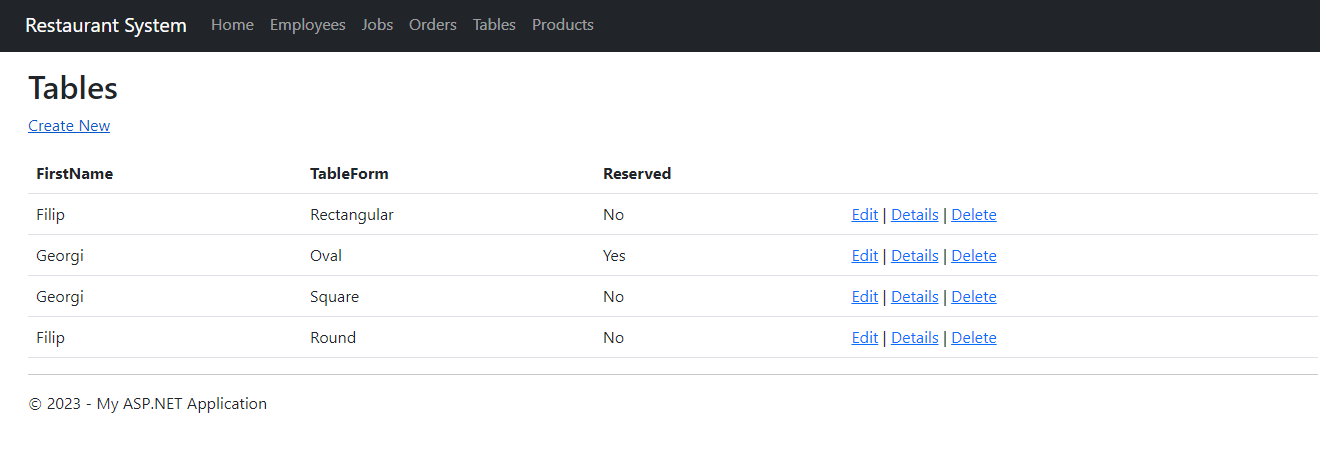


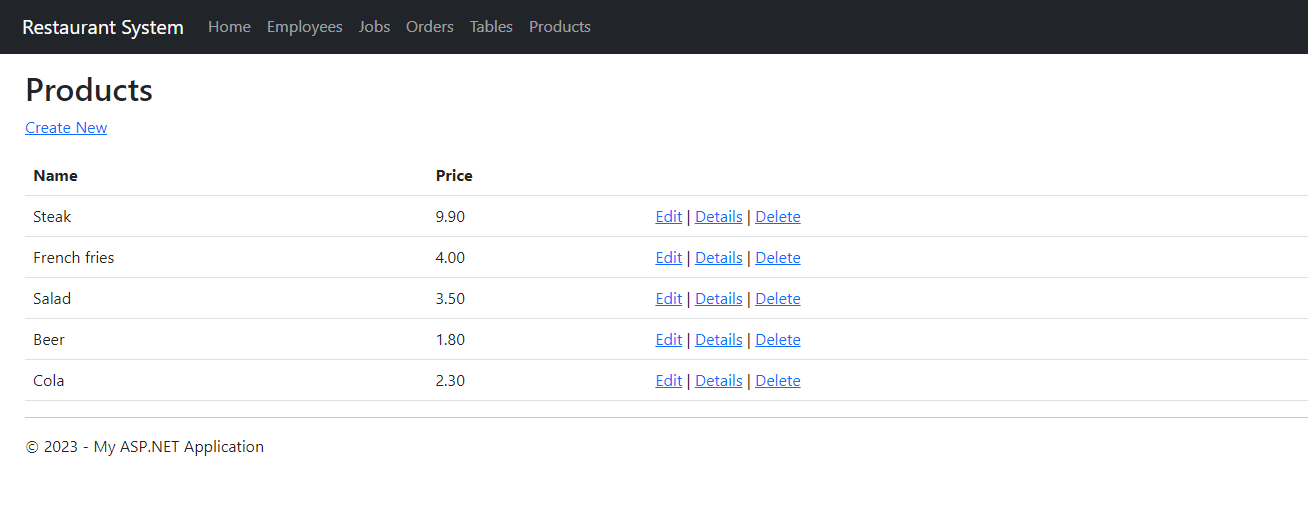
Скрийншоти за ASP.NET











Използвана литература

StackOverflow.com

learn.microsoft.com

W3schools.com