پروژه پایانی نرم افزار

تهیه کننده : رضا صلحی

رشته تحصیلی : نرم افزار کامپیوتر

استاد : جناب آقای مهندس .....

عنوان پروژه

پروژه پورتال کلینیک دندان سازی

سال تحصیلی 1403-1404

مقدمه

در این بخش به طور خلاصه توضیح می‌دهید که پروژه چه کاری انجام می‌دهد، هدف آن چیست و برای چه کسانی طراحی شده است. مثلا:

پروژه دندان سازی صلحی یک وب‌اپلیکیشن برای مدیریت خدمات دندان‌سازی و ثبت سفارشات دندان‌پزشکان است. این سیستم به دندان‌پزشکان امکان می‌دهد که سفارشات دندان‌سازی خود را ثبت کنند و اطلاعاتی درباره خدمات و نمونه‌کارهای دندان‌پزشکی به عموم نمایش داده می‌شود. همچنین، مدیر سیستم می‌تواند سفارشات را مشاهده کرده و به آنها رسیدگی کند.

معماری پروژه

در این بخش، معماری پروژه و لایه‌های مختلف آن را شرح دهید. چون شما از معماری چند لایه استفاده کرده‌اید، هر لایه را توضیح دهید.

- **WebUI**: لایه‌ای که مسئول نمایش صفحات HTML و تعامل با کاربر است.
- **Domain**: لایه‌ای که مدل‌ها و منطق تجاری (Business Logic) پروژه در آن قرار دارد.
- **Infrastructure**: لایه‌ای که شامل کدهایی برای تعامل با دیتابیس و دیگر سرویس‌ها است. در اینجا از **EF Core** برای ارتباط با دیتابیس استفاده شده است.

نحوه کارکرد پروژه

در این بخش توضیح دهید که پروژه چگونه عمل می‌کند. برای هر بخش کلیدی پروژه، یک توضیح بدهید:

- **صفحه اصلی (Home)**: شامل اطلاعاتی درباره دندان‌سازی و خدمات آن.
- **صفحه سفارشات (Order)**: دندان‌پزشکان می‌توانند سفارش‌های خود را ثبت کنند.
- **صفحه لیست سفارشات (OrderList)**: این صفحه فقط برای مدیر است و لیست تمام سفارشات را نمایش می‌دهد.
- **صفحه تماس با ما (ContactUs)** و **درباره ما (AboutUs)**: شامل فرم‌های مربوطه برای ارسال اطلاعات.
- **سیستم ورود و مدیریت دسترسی‌ها**: سیستم ورود برای مدیر (Admin) جهت مشاهده سفارشات و دسترسی به صفحات خاص.

لایه‌های پروژه

در این بخش جزئیات فنی و نحوه ساختار لایه‌های مختلف پروژه را شرح دهید:

لایهWebUI

این لایه مسئول نمایش رابط کاربری پروژه است. در اینجا از **Razor Pages**برای ساخت صفحات استفاده شده است. برخی از صفحات مهم شامل:

- صفحه :**Home**که اطلاعات عمومی درباره دندان‌سازی و خدمات آن نمایش داده می‌شود.
- صفحه :**Order**فرم ثبت سفارش برای دندان‌پزشکان.
- صفحه :**OrderList**تنها برای مدیر نمایش داده می‌شود و لیست سفارشات ثبت شده را نشان می‌دهد.
- صفحات **ContactUs**و :**AboutUs**فرم‌هایی برای تماس با کلینیک و نمایش اطلاعات مربوطه.

لایهDomain

این لایه شامل مدل‌ها و منطق تجاری پروژه است. در اینجا مدل‌های مختلف مانند ,**Order**, **User** **ContactInfo**و غیره تعریف شده‌اند. در این بخش از **AutoMapper**برای مپ کردن داده‌ها بین مدل‌ها وDTO ها استفاده شده است.

- :**Order**مدل سفارش که شامل اطلاعاتی مانند نام دندان‌پزشک، نوع خدمات، تاریخ و وضعیت سفارش است.
- :**User**مدل کاربران که شامل اطلاعات کاربران (مدیر و دندان‌پزشک) می‌شود.
- :**ContactInfo**مدل اطلاعات تماس که شامل آدرس، شماره تلفن و دیگر اطلاعات کلینیک است.

لایهInfrastructure

این لایه مسئول تعامل با منابع خارجی است، مانند دیتابیس وAPI های دیگر. در این لایه از **EF Core**برای ارتباط با دیتابیس استفاده شده است.

- :**EF Core**برای ایجاد و مدیریت دیتابیس و عملیات CRUD استفاده می‌شود.
- :**Repository Pattern**برای مدیریت دسترسی به داده‌ها از این الگو استفاده می‌شود.

AutoMapper

در این پروژه از **AutoMapper**برای نقشه‌برداری داده‌ها استفاده شده است. به‌طور مثال، هنگام ثبت یا ویرایش سفارشات، داده‌ها از مدل‌های DTO به مدل‌های دامنه (Domain) مپ می‌شوند تا به دیتابیس ارسال شوند.

پایگاه داده

در این بخش ساختار دیتابیس و جداول موجود را توضیح دهید.

- **جدول** :**Users** برای ذخیره اطلاعات کاربران (دندان‌پزشکان و مدیر).
- **جدول** :**Orders** برای ذخیره اطلاعات سفارشات دندان‌پزشکان.
- **جدول** :**ContactInfo** برای ذخیره اطلاعات تماس و آدرس کلینیک.
- **استفاده از** :**EF Core** توضیحاتی درباره نحوه استفاده از EF Core برای تعامل با دیتابیس و انجام عملیات.CRUD

امنیت و احراز هویت

در این بخش به پیاده‌سازی سیستم ورود و امنیت پروژه اشاره کنید.

- **سیستم ورود** :**(Login)** با استفاده از **ASP.NET Core Identity**برای مدیریت ورود و ثبت‌نام کاربران.

نحوه نصب و راه‌اندازی پروژه

- **پیش‌نیازها** :نسخه‌های مورد نیاز **.NET Core**, **Visual Studio**, **SQL Server**و غیره.
- **نصب و راه‌اندازی** :مراحل راه‌اندازی پروژه، مانند تنظیم دیتابیس، نصب پکیج‌های NuGet و تنظیمات مربوطه.

###نحوه استفاده از EF Core و:AutoMapper
در این پروژه، برای انجام عملیات CRUD از **EF Core** استفاده شده است و داده‌ها با استفاده از **AutoMapper** بین مدل‌ها وDTO ها نقشه‌برداری می‌شوند.

توضیحات درباره.NET Core

NET Core.یک فریم‌ورک متن‌باز، مدرن و چندمنظوره است که توسط شرکت مایکروسافت توسعه داده شده است. این فریم‌ورک برای ساخت اپلیکیشن‌های وب، دسکتاپ، موبایل، سرویس‌های ابری و سایر انواع نرم‌افزارها مناسب است. از ویژگی‌های اصلی و مزایای استفاده از NET Core. می‌توان به موارد زیر اشاره کرد:

ویژگی‌های اصلی:NET Core.

1. **چند پلتفرمی بودن** NET Core. **(Cross-Platform)** این امکان را فراهم می‌کند که اپلیکیشن‌ها روی سیستم‌عامل‌های مختلف از جمله **Windows**, **Linux**و **macOS**اجرا شوند. این ویژگی باعث می‌شود که توسعه‌دهندگان بتوانند اپلیکیشن‌هایی بسازند که روی پلتفرم‌های مختلف بدون نیاز به تغییرات اساسی در کد، اجرا شوند.
2. **متن‌باز بودن** NET Core. **(Open-Source)** متن‌باز است و کد آن در GitHub قابل دسترس است. این ویژگی باعث می‌شود که توسعه‌دهندگان بتوانند به راحتی به فریم‌ورک دسترسی داشته باشند و آن را مطابق نیازهای خود توسعه دهند یا آن را بررسی کنند.
3. **عملکرد بالا** NET Core. **(High Performance)** به طور قابل توجهی از نظر عملکرد بهتر از نسخه‌های قبلی NET. عمل می‌کند. این فریم‌ورک برای ساخت اپلیکیشن‌هایی با عملکرد بالا و مصرف کم منابع بسیار مناسب است.
4. **پشتیبانی از کانتینرها و** NET Core. **Kubernetes** به طور طبیعی از فناوری‌هایی مانند **Docker**و **Kubernetes**پشتیبانی می‌کند. این ویژگی به توسعه‌دهندگان این امکان را می‌دهد که اپلیکیشن‌های خود را در کانتینرها اجرا کنند و آن‌ها را در محیط‌های ابری مانند **Microsoft Azure**یا **AWS**به راحتی مقیاس‌پذیر کنند.
5. **پشتیبانی ازAPI ها و میکروسرویس‌ها** NET Core.برای ساخت **APIهای RESTful** و سیستم‌های میکروسرویس (Microservices) بسیار مناسب است. این فریم‌ورک از تکنیک‌های مدرن معماری نرم‌افزار پشتیبانی می‌کند که باعث می‌شود توسعه‌دهندگان بتوانند اپلیکیشن‌هایی مقیاس‌پذیر و انعطاف‌پذیر بسازند.
6. **پشتیبانی از** NET Core. **Dependency Injection (DI)** به طور داخلی از **Dependency Injection**پشتیبانی می‌کند. این ویژگی باعث می‌شود که کدهایی که نوشته می‌شود تمیزتر، قابل تست‌تر و به راحتی قابل نگهداری باشد.
7. **سرعت و کارایی در توسعه** NET Core.دارای ابزارهای توسعه قدرتمندی است که سرعت توسعه را افزایش می‌دهند. به عنوان مثال، **Entity Framework Core**برای تعامل با دیتابیس‌ها، **ASP.NET Core**برای توسعه وب وRESTful API ، و ابزارهای **CLI (Command Line Interface)**برای مدیریت پروژه‌ها و توسعه سریع‌تر موجود هستند.

ویژگی‌های استفاده شده در این پروژه باNET Core.

در پروژه "**دندان سازی صلحی** "از ویژگی‌های مختلف NET Core. برای ساخت و توسعه یک اپلیکیشن وب مدرن استفاده شده است. در اینجا ویژگی‌های اصلی و مزایای استفاده از NET Core. که در این پروژه پیاده‌سازی شده‌اند، آورده شده است:

1. **ASP.NET Coreبرای توسعه وب:**
2. **Entity Framework Coreبرای ارتباط با دیتابیس:**
   - در این پروژه از **Entity Framework Core (EF Core)**برای مدیریت ارتباط با دیتابیس استفاده شده است EF Core .یک ORM (Object-Relational Mapper) است که عملیات CRUD (Create, Read, Update, Delete) را به راحتی و با کمترین کد نویسی انجام می‌دهد.
   - با استفاده ازEF Core ، ارتباط با دیتابیس به شکل خودکار انجام می‌شود و نیاز به نوشتن SQL دستی کاهش می‌یابد. همچنین، از قابلیت **Migrations**برای مدیریت تغییرات دیتابیس استفاده شده است.
3. **AutoMapperبرای مپ کردن داده‌ها:**
   - برای تبدیل داده‌ها بین مدل‌ها وDTO ها (Data Transfer Objects) از **AutoMapper**استفاده شده است. این ابزار به طور خودکار داده‌ها را از یک مدل به مدل دیگر مپ می‌کند و کدنویسی اضافی را کاهش می‌دهد.
   - به عنوان مثال، هنگام ثبت یا ویرایش سفارش‌ها، داده‌ها از مدل‌های DTO به مدل‌های دامنه مپ می‌شوند.
4. **Middlewareو مدیریت درخواست‌ها:**
5. **احراز هویت و مجوزها:(Authentication & Authorization)**
   - برای مدیریت ورود و امنیت کاربران از **ASP.NET Core Identity**استفاده شده است. این سیستم شامل تمام امکانات برای مدیریت کاربران، احراز هویت و مدیریت دسترسی است. در این پروژه کاربران به دو نقش اصلی **مدیر** و **دندان‌پزشک** تقسیم می‌شوند و دسترسی به صفحات مختلف بر اساس نقش‌ها کنترل می‌شود.
6. **Dependency Injection (DI):**
   - NET Core.به طور پیش‌فرض از **Dependency Injection**پشتیبانی می‌کند. در این پروژه، از DI برای مدیریت وابستگی‌ها بین اجزای مختلف سیستم مانند سرویس‌ها و مخزن‌ها (Repositories) استفاده شده است. این کار باعث می‌شود که کد به صورت تمیزتر و قابل تست‌تر باشد.
7. **ترکیب Bootstrap و TailwindCssو طراحی ریسپانسیو:**

- برای طراحی رابط کاربری و ایجاد صفحات وب ریسپانسیو از **ترکیب Bootstrap و TailwindCss** استفاده شده است. این فریم‌ورک طراحی برای ساخت صفحات وب زیبا و پاسخگو (Responsive) با استفاده از CSS و JavaScript بسیار مفید است.

چرا از NET Core. برای این پروژه استفاده شده است؟

استفاده از **NET Core.**در این پروژه به دلایل زیر انتخاب شده است:

- **چند پلتفرمی بودن** :با استفاده از NET Core. می‌توان پروژه را روی پلتفرم‌های مختلف (Windows, Linux, macOS)اجرا کرد. این ویژگی به ما این امکان را می‌دهد که در آینده این اپلیکیشن را در محیط‌های مختلف میزبانی کنیم.
- **عملکرد بالا** :با توجه به اینکه این پروژه ممکن است به مرور زمان به تعداد سفارشات زیاد و درخواست‌های همزمان نیاز داشته باشد، استفاده از NET Core. به دلیل عملکرد بالا و مصرف پایین منابع بسیار مناسب است.
- **امنیت و احراز هویت** :سیستم‌های امنیتی موجود در **ASP.NET Core Identity**به ما این امکان را می‌دهد که به راحتی سیستم ورود، مدیریت کاربران، و محدودیت‌های دسترسی را پیاده‌سازی کنیم.
- **مقیاس‌پذیری** :این فریم‌ورک به راحتی مقیاس‌پذیر است و می‌توان آن را برای نیازهای آینده پروژه گسترش داد.