ASP.Net Core 3.1 React Project Basic Implementation Guide

1. Application Structure

Clean Architecture [(What is clean architecture?)](https://pusher.com/tutorials/clean-architecture-introduction)

1. **Domain Layer (Portal.Domain)**

It is the center part of the architecture. It holds all application domain objects. If an application is developed with ORM entity framework then this layer holds POCO classes (Code First) or Edmx (Database First) with entities. These domain entities don't have any dependencies and not any piece of code which talks to database or has any other business functions.

1. **Infrastructure Layer (Portal.Infrastructure)**

Maps to the layers that hold the Database and Gateway concerns. In here, we define database access (typically in the shape of repositories), integrations with other network services, caches, etc. This project/layer contains the physical implementation of the interfaces defined in our domain layer.

1. **Application Core Layer (Portal.ApplicationCore)**

Maps to the layers that hold the Use Case and Entity concerns and is also where our External Interfaces get defined. These innermost layers contain our domain objects and business rules. The code in this layer is mostly pure C# - no network connections, databases, etc. allowed. Interfaces represent those dependencies, and their implementations get injected into our use cases.

1. **Web Layer (Portal.Web)**

Maps to the layers that hold the Web, UI and Presenter concerns. Means it accepts input in the form of http requests over the network (e.g., GET/POST/etc.) and returns its output as content formatted as JSON/HTML/XML,JSX etc. The Presenters contain .NET framework-specific references and types, so they also live here as per The Dependency Rule we don't want to pass any of them inward.

References –

<https://fullstackmark.com/post/18/building-aspnet-core-web-apis-with-clean-architecture>