

Layered Architecture Pattern

Database (DB)

Stores the actual data in tables/collections. The persistent data layer.



Model (Entity/Domain Model)

Represents database tables as classes. The blueprint of your data structure that maps directly to database schema.



Repository (Repo)

Abtracts database operations. Handles CRUD operations and queries using Models, separating data access logic from business logic.



Service

Contains business logic and rules. Orchestrates operations, validates data, and coordinates between repositories.



Controller

Handles HTTP requests/responses. Routes incoming requests to appropriate services and returns responses to the client.



DTO (Data Transfer Object)

Simple objects that carry data between layers. Used to transfer only necessary data, often for API requests/responses.



ViewModel

Shapes data specifically for the view/UI. Contains only what the presentation layer needs, often combining data from multiple sources.



View

The UI layer that displays data to users. HTML, React components, or mobile screens.

⚡ Supporting Components

🛠 Helper

Utility functions and common operations used across the application (e.g., date formatting, string manipulation).

⚙ Configuration

Settings and setup for the application (database connections, API keys, environment-specific settings).

🏗 Mapping Profile

Defines how to convert between objects (Model \leftrightarrow DTO \leftrightarrow ViewModel), typically using tools like AutoMapper.