3 layer architecture

### What is the Three layer architecture

* The three-layer architecture organizes applications into three logical layers:
  + Presentation layer: responsible for making the bridge between the user and our application, it receives and validate user input, provides responses...
  + Business logic layer: apply business logic and transform the input data to entity, transport the response from the database to the presentation layer
  + Data access layer: manipulating and managing data

### Benefits

* **Separate responsibilities**: Each layer has a specific responsibility
* **Manage dependencies**: A higher layer can use services in a lower layer, but not the other way around
* **Maintainability**: codebase much cleaner, more understandable, and more maintainable
* **Better security**: The client does not have direct access to the database and obtain unauthorized data
* **Integrity**: BLL ensure that only validated information is allowed to be updated in the database
* **Faster development**: each team/member can work on each layer independently

### Challenges

* For small projects, it's likely end up with a BLL just for passing CRUD requests, adding unnecessary latency
* In a monolithic app, all functions are implemented using the same technology stack. That limits the ability to take advantage of other languages, frameworks, or resources
* BLL holds the important logic but it is dependent on data access implementation details -> hard to test business logic, requires mock db
* To scale any part requires scaling the entire app, which often leads to a costly waste of compute and infrastructure resources

### Workflow

* The Presentation layer directly interact with client. Mainly receives requests and data from user and passes them to BLL for further processing
* BLL does business logic on the data. It retrieve and/or update some data to the database by making requests to Data access layer
* DAL builds queries to the database to handle these requests and sends the result back to BLL
* BLL completes the process and sends result to PL
* PL sends responses to client

### 3 layer architecture in ASP.Net

* To reduce complexity, we create an optional layer - Domain/Entity layer - contains classes and common helper functions used for all layer
* We also rename layers to fit our application better
  + Presentation - API
  + Business logic - Service
  + Data access - Infrastructure