**Clean Architecture: Where Software Comes to Life**

- Also known as Onion, Hexagonal, or Ports and Adapters Architecture

- Focuses on reducing complexities and putting core logic first

- Perfect for projects that value solid business rules and rigorous testing

**What Makes Clean Architecture Stand Out**

- Core domain logic takes the center stage

- Domain talks to the web part, while infrastructure handles databases and services

**The Core Project: Heart of the System**

- Holds vital business elements and rules together

- Acts as the central hub for dependencies

- Internal projects define interfaces, external ones implement them

**Infrastructure: The Backbone**

- Manages repositories, databases, and APIs

- Infrastructure project handles the nitty-gritty details

**Web Project: Gateway to the App**

- API endpoints, controllers, views, and more

- Web project is where the app comes to life

**Shared Kernel Area: Common Ground**

- Houses essentials like entities and logging mechanisms

- No ties to infrastructure components

**The Clean Architecture Advantage**

- Clears up responsibilities, reducing confusion

- Cuts down on complex links between parts

- Follows SOLID principles for robust design

- Testing becomes a breeze with better separation

**Remember the Trade-off**

- Clean Architecture brings benefits but takes time to learn

- Simpler styles like All-in-One or Layered are quicker to set up