 **Part 01: Views**

* **Objective**: Understand how to create and work with views in ASP.NET MVC.
* **Key Concepts**: Creating .cshtml files, passing data to views, using ViewBag, and ViewData.
* **Action Points**: Practice rendering views with dynamic data and explore how to use Razor syntax to integrate C# into HTML.

 **Part 02: Serving Static Files**

* **Objective**: Learn how to manage and serve static resources like images, CSS, and JS files.
* **Key Concepts**: Configuring the wwwroot folder, linking static files to your views, and ensuring proper paths are used.
* **Action Points**: Set up a sample project where static files like Bootstrap, custom CSS, and images are properly served.

 **Part 03: Layout and \_ViewStart**

* **Objective**: Implement consistent site-wide layouts using shared \_Layout and \_ViewStart files.
* **Key Concepts**: Creating a layout template, including headers, footers, and navigation. Using \_ViewStart to apply layout globally.
* **Action Points**: Create a layout page with a responsive navigation bar and ensure all views inherit from it using \_ViewStart.

 **Part 04: Tag Helpers and \_ViewImport**

* **Objective**: Simplify HTML elements using Tag Helpers and organize imports with \_ViewImport.
* **Key Concepts**: Applying Tag Helpers for form elements, links, and inputs. Managing namespaces and imports for views using \_ViewImport.
* **Action Points**: Create a form using Tag Helpers and ensure \_ViewImport is properly set up to reduce repetitive namespace declarations.

 **Part 05: Architecture Pattern**

* **Objective**: Explore different architectural patterns used in MVC projects.
* **Key Concepts**: Understanding the MVC pattern, possibly touching on other patterns like Repository and Service layers.
* **Action Points**: Draft a diagram of your current project's architecture and evaluate if a pattern like Repository or Unit of Work could enhance your design.

 **Part 06: Create MVC Project**

* **Objective**: Step-by-step guide to setting up a new MVC project in Visual Studio.
* **Key Concepts**: Project creation, folder structure, adding controllers, models, and views.
* **Action Points**: Create a sample MVC project and familiarize yourself with the default folders and how they interact (Models, Views, Controllers).

 **Part 07: Data Access Layer**

* **Objective**: Learn how to implement the Data Access Layer (DAL) for database operations.
* **Key Concepts**: Setting up Entity Framework, writing queries, creating repositories for data handling.