ASP.NET MVC Web Development Roadmap

A detailed guide to building web applications with ASP.NET MVC.

1. Overview of ASP.NET MVC

- **Introduction to ASP.NET MVC**: Learn about the Model-View-Controller (MVC) architecture and its benefits in web development.
- **Controller - View - Model**: Understand the roles and responsibilities of each component: Model (data), View (UI), and Controller (processing logic).
- **Introduction to Routing**: Learn how ASP.NET MVC maps URLs to actions in a Controller.
- **Building a Web MVC App with a Database**: Practice creating a basic MVC project to connect with a database.

2. Database Interaction Languages & Architectures

- **Entity Framework (EF)**: Learn about .NET's powerful ORM (Object-Relational Mapping), its architecture, and programming models.
- **Database First Model**: Learn how to create models from an existing database using the EF API.
- **Code First Model**: Learn to define models (Entities) in code first, then EF will automatically create the database. Includes: building Entities, creating the Database, initializing data.
- **LINQ to SQL**: Learn the architecture of LINQ (Language-Integrated Query) and how to use it to query data from a database natively.

3. Data Control – Controller

- **Parameter Binding**: Learn different ways for a Controller to receive data from the user: via Request, FormCollection, Action arguments, and model binding.
- **Action Results**: Learn about the different types of results an Action can return, such as text, a View, a redirect to another Action, a URL, or a file.
- **Action Selectors**: Use attributes to help the routing system select the correct Action to execute, e.g., [HttpGet], [HttpPost].
- **Action Filters**: Learn how to use filters to add logic before or after an Action is executed, such as for authentication or logging.

4. Website Organization & Navigation

- **Layouts**: Use Layouts to create a common UI template and @RenderBody(), @RenderSection() to define changeable content areas.
- **Bundles**: Learn how to group and optimize CSS and JavaScript files to improve page load performance.
- **Partial Views**: Learn how to create reusable Views using @Html.Action() and @Html.Partial().
- **Multi-language**: Learn techniques to build websites that support multiple languages.
- **Areas**: Organize a large project into smaller, independent functional sections.

5. UI Generation & Data Sharing

- **Razor**: Master the Razor syntax to smoothly embed C# code within View files.
- **MVC Helpers**: Use built-in Helpers (@Html) to generate HTML elements like Forms, TextBoxes, CheckBoxes, DropdownLists.
- **Custom Helpers**: Learn how to create your own Helpers to reuse UI generation code.
- **Data & State Sharing**: Learn the ways to pass data from a Controller to a View: ViewBag, Model, Session, Application, Cookie.
- **Global.asax**: Understand the application lifecycle and how to handle global events like Application_Start, Session_Start.

6. Validation & Security

- **MVC Validation**: Use Data Annotations ([Required], [StringLength], [Range],...) to automatically validate Model data.
- **Manual Validation**: Add custom validation logic in the Controller when Data Annotations are not sufficient.
- **Website Attacks - XSS, CSRF**: Learn about and prevent common attack types like XSS and CSRF (using Antiforgery Tokens).
- **Authentication & Authorization**: Implement mechanisms for authentication (who you are) and authorization (what you are allowed to do).

7. Website Deployment

- **Register Web Hosting**: Learn how to sign up for a hosting service to host your website.
- **Manage Hosting Server**: Get familiar with hosting management tools like cPanel or Plesk to configure your website.
- **Deploy ASP.NET MVC Website**: Learn the steps to publish and upload your ASP.NET MVC application to the hosting server.