Topics that we will Cover:

* Asp.net MVC building blocks,
* CRUD operation with Entity FrameWork Core,
* Build RestFul Services,
* Asp.net Identity,
* Paypal integration
* Azure Development
* Model Binding

By the end of this course I will learn how to use asp.net frame work and use it in a personal and commercial webapplication

Step by step what we’ll be learning and doing :

1. Whatis MVC ? (Learn about how to set up the views model and controllers)
2. DBContext file which is used to interact with database
3. CRUD data using EFCore
4. Services ( to work with the data)
5. ViewComponents ( To add and edit the shopping cart )
6. Authentication and Authorization.

Other topic include:

* PayPal SDK for online payments,
* Dependency Injections,
* Model Binding,
* Routing
* Model Validations
* Tag helpers

**MVC ( Model View Controller)**

A model is just a C# class that represent server in the sql format:

* It servers as a data blue print,
* And also help to find the data relation.

Eg : Authors, main characters and anything related to to database we need to create the tables and to create the tables we need to create the models,

**View**

is what user get to see it is a file that represents user interface

* It have cshtml ( cshtml is razor meaning the combination of html and c#)
* It is used to trigger the event.

**Controller:**

* It receives the event,
* Returns the data to the view,]
* A controller is just a C# class that inherits from the Controller base class,
* Through this course we are going to create, author controller, main character controller, manga controller.

**Web Applications**

How does it work ?

A computer screen with arrows pointing to a server

Description automatically generated

From the above picture

1. First the browser sends the request to the server,
2. Server then look the request in the Database and get the item that it wants,
3. DataBase then returns the item to the server,
4. Server then send the item to the browser

**MVC**

**A screen shot of a computer

Description automatically generated**

From above picture:

1. View first sends the request to the controller,
2. Controller then sends the request to the Model which in this case is a database
3. Model then analyze the request and sends the data that needed to be send to the controller,
4. Controller and then send data to the view, which is later on displayed to the client.