3 types of applications in .NET Core:

1. Desktop Applications - WPF (with GUI) / Console appliations (with out GUI)

2. Web Based Applications - Razor pages(websites), MVC (web applications), API ( Web application)

3. Mobile Applications - Xamarin in VS

Web site is a collection of web pages

Web application - web site + business logic

Application server to host the web applications/ websites - IIS in .NET

Controller is a class which is a collection of Action methods.

Every controller will have a folder in the Views/Controllername

URL : https://localhost:portnumber/controllername/actionmethod

http://localhost:7040/Home/Privacy

HTTP://localhost:portnumber/Student/studentlist

Model folder

Student.cs - class

namespace FirstMVCPrj.Models

{

public class Student

{

public int Sid { get; set; }

public string Name { get; set; }

public Student(int id, string name)

{

Sid = id;

Name = name;

}

}

}

Controller folder

Add controller - studentController

public IActionResult Studentlist()

{

List<Student> students = new List<Student>();

students.Add(new Student(1,"Ram"));

students.Add(new Student(2,"Vishal"));

students.Add(new Student(3,"Vandhana"));

return View(students);

}

Right click within the Studentlist

- Add view - Razor view - template - list

Model - Student

Add

Website

Web application

Web server

Application server

Hosting/Deployment

What is MVC

API

Difference

Json format

Model

View

Controller

API:

1. API returns the data in the form of Json -[{"name":"value"},{},{}]

2. API uses HTTP protocol only

HTTP verbs: - CRUD

1. Create - adding a new record - HTTPPOST - [annotations]

2. Read - select \* from - HTTPGET

3. Update -HTTPPUT

4. Delete - HTTPDELETE

swagger is a testing cum documentation tool

Entity Framework Core:(ADO.NET as an underlying architecture) - used to connect to the backend

ORM tool - Object Relation Mapping tool

Table - Class in the front end - Database first approach

Class - Table in the back end - Code first approach

3 Nuget packages -

1. Microsoft.entityframeworkcore

2. Microsoft.entityframeworkcore.tools

3. Microsoft.entityframeworkcore.sqlserver

Context- front end database

collection of entity classes is called context

Server=your\_server\_address;Database=your\_database\_name;User ID=your\_username;Password=your\_password;

Create an API for Food ordering system.

1. products - GetallProducts, AddProduct, GetProductByCategory

2. Orders - AddOrder, CancelOrder,ListAllOrders

3. Customers - AddCustomer, GetCustomerById, UpdateCustomerDetails