15/10/24

* Web:

→ Client/Server Architecture

→ Port: 80

→ Basic Language: HTML

* HTML:

* Heading: <h>

Paragraph <p>

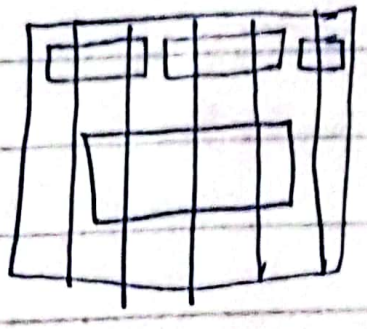
Division

→ Web Page not only contains HTML.

→ Inline Element (In same line)
→ Block Element (Step by step elements (In different line).
→ (Block level element) (Divide a page)
→ (Inline element) (Container) (On same line)
→ <img>, <h>, <ul>, <ol> (ordered list), <p>
→ To choose our own place of elements we use CSS (Visual appearance) (Position) (cascading style sheet)
→ For behavior in page, we use javascript.
→ Page → HTML, CSS, JavaScript
→ Form tag which take input from User. It needs to be saved to Server side DB. (Using ASP.NET)
→ Razor Pages, MVC pattern, Single Page Application BLazor (These can be used by page. to make

web application dynamic).

* Bootstrap will be used.
* FlexBox and Grid style is used for positioning of elements. (Main, Header, Footer). It can also be done using simple CSS.
* Bootstrap divides the page into 12 columns.



* **MVC** (Model View Controller):

→ One application is divided into 3 parts:

(i) View (HTML) etc.... (Includes Interfaces)

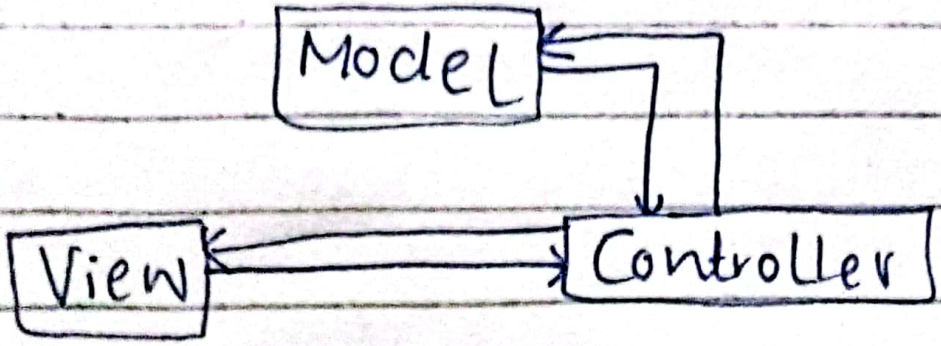
(ii) Controller (Control the flow of application) (Brain)

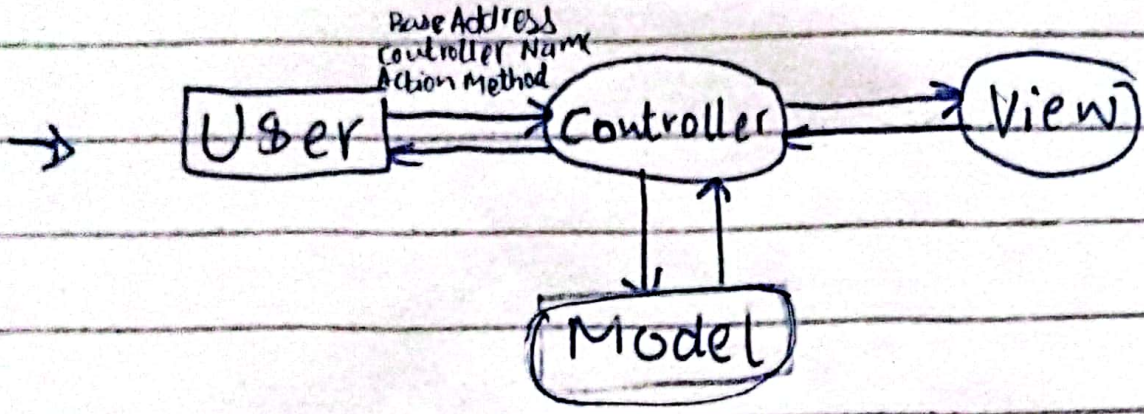
(iii) Model (Those classes which represent Business) (CMS (Course, Student, Faculty etc....))

; Work is done according to use-case

: Problem contains Entities

* MVC Diagram:





* Actions in Controller : Add, Update, Delete for student

* MVC using : (2) Action Method (1) Controller Name

* Multiple controller can be used in web. It is made based on entities. Action Method are defined in Controller.

* Search template : (i) ASP NET core App (Razor Pages)

(ii) Model View Controller. (ASP.NET core web App) (Model View Controller)

(iii) Blazor web App

* Visual Studio Installer (ASP.NET AND web Development)

(Model View Use)

* My First MVC Project

(In solution Explorer) www root folder (css, js, lib)

(site.css) (site.js) (Bootstrap JQuery) (Libraries can be added/removed).

* Add -> New Folder -> Images
* Controllers -> HomeController.cs
  (Any Name)
  (This naming convention would be followed) (Student Controller).
* Starts with "I", its interfaces.
* Class Represents real world object.
  (Entities, Functions, Properties)
* Interface:
  -> Abstract-level thing which defines what this entity will do.
  -> Function Definition.
  -> Logger (Application behavior Log)
  (For crashing of App) (Real Deployment contains log)
* public string Index()

```
{
    return "My First MVC Application";
}
```

(Controller) (Action)
-> Home / Index
-> MapController

→ Inspect → Network Tab (Enter)

Response ← Click on Action Button

→ public ViewResult About()

New Action method

{

return View();

}

→ Now we will create View

(View Folder in Solution Explorer)

(Home → Add → View → Razor

About ← View Empty)

→ <html>

<body>

<h1> my first MVC Application

View </h1>

<body>

</html>

→ BaseAddress / Home / About

(Controller)

→ Model → Add → Class → Product

(Id, Name, Desc, Price) Properties ←

→ New Controller → Add → Controller

(Product Controller ← MVC Empty ←

: add Project Name;
using MyFirstMVC.Models;
(In Namespace MyFirstMVC)

: cshtml (csharp html) (csharp can be embeded)

-> public IAction Result Index()
{
Product product = ...
product.Id = 1;
product.Name = "Mobile"
product.Description = "This is a Mobile"
product.Price = 10;
return View(product); : object model
}

=> New view -> Folder (Product)
Index <- Add view
↳ HTML (simple)

(To embed C#) @
<h1> Current Time : @System.DateTime.Now </h1>

: (Browser only receives HTML) (Not Csharp)
: View Page Source (contains HTML only)
: All Processing on Service side.

 @ Model.Name
" " " .Price "
" " " .Description "

: Save changes By red Button.