**Blazor Course**

1.

Data Binding

Components and routing

Parameters and cascading Parameters

Javascript in blazor

2. Blazor Server  
 Admin Management Portal

Authentication And Authorization

Quill JS

Seed Database

3. .Net Api

Endpoint for Client

Bearer Token

CORS

Data Transfer Object

Using EF Core

Repository Pattern

4. Blazor WASM

Customer Facing Website

HttpClient Api

Stripe Payment

Validations and Email

Two Types of hosting model

Server hosting model -> Blazor Server

Client hosting model -> Blazor WASM

**One Way Data Binding**

In blazor is the process of rendering view as per the values inside the model that you see

<h2 class="bg-light border p-2">

First Room

</h2>

Room : @Room.RoomName

<br />

Price : @Room.Price

@code{

BlazorRoom Room = new BlazorRoom()

{

Id = 1,

RoomName = "Villa Suite",

IsActive = true,

Price = 2000

};

}

**Two Way Data Binding**

If you change value inside the textbox it will automatically update in model or object

@bind-value -> attribute for two way binding

* It is on change event attribute

<h2 class="bg-light border p-2">

First Room

</h2>

Room : @Room.RoomName

<br />

Price : @Room.Price

<br />

<input type="number" @bind-value="@Room.Price" @bind-value:event ="oninput"/>

@code{

BlazorRoom Room = new BlazorRoom()

{

Id = 1,

RoomName = "Villa Suite",

IsActive = true,

Price = 2000

};

}

**Blazor Lifecycle**

**Oninitialized()**

**OninitializedAsync()**

Executed when the component is completely loaded.

You can use this method to load data from service because each control in the UI is loaded after this mehod.

**OnParameterSet()**

**OnParameterSetAsync()**

* This method call when a component is first initialized after that they are called each time a new or updated parameter are received from parent in the render tree.

**OnAfterRender(bool firstRender)**

**OnAfterRenderAsync(bool firstRender)**

* Called after each render of the component.

**ShouldRender()**

* this method returns a Boolean value, if true, it refresh the ui, otherwise changes are not sent to ui.

**StateHasChanged()**

* it is called after any lifecycle method has been called.
* this method notifies the component that its state has changed.
* It can be invoked manually to trigger a ui Rerender. If some at some point the ui is not refreshing and you want to refresh manually you can use this method.

[Parameter]

public EventCallback<string> OnAmenitySelect { get; set; }

* Passed into chilled component as parameter to call parent component method in chilled component.
* EventCallback provides a simple API for communication between child and parent components.
* When you want to call parent component method into chilled component

**We can directly call parent component method in chilled component for that create a eventcallback parameter in chilled component and pass into button as onclick method and then in parent component create a method and pass in chilled component as parameter**

We can pass multiple renderfragment, defining multiple fragment parameter

**Attribute Splatting**

<div>

<h4 class="text-primary pt-3">Another chilled component</h4>

@\* <input id="RoomName" placeholder="@Placeholder" maxlength="@MaxLength" required="@Required" class="form-control"/>\*@

<input id="RoomName" @attributes="InputAttributes" class="form-control" />

</div>

@code {

// [Parameter]

// public string Placeholder { get; set; }

// [Parameter]

// public string Required { get; set; } = "required";

// [Parameter]

// public string MaxLength { get; set; } = "5";

// Attrigute Splattig

[Parameter]

public Dictionary<string, object> InputAttributes { get; set; } = new Dictionary<string, object>()

{

{"required", "required"},

{"placeholder", "Initial Text"},

{"maxlength", 5}

};

}

* Mostly it is used to capture and render additional parameters that have passed on top of the default parameter.

**We can pass multiple url in one page**

@page "/LearnRouting"

@page "/NewRouting"

@page "/LearnRouting/{parameter}/{parameter2}"

<h3>LearnRouting</h3>

<br />

<h4>Parameter - @parameter</h4>

<h4>Parameter2 - @parameter2</h4>

@code {

[Parameter]

public string parameter { get; set; }

[Parameter]

public string parameter2 { get; set; }

}

IJSRunTime

* Has two types of method one to call those function that returns something and other that does not return anything.

ToastR

* In blazor server side add js and css in \_host.cshtml

**Referencing Component :**

<**ChildComponent** @ref="childComponent" **OnClickBtnMethod**="ShowMessage" **Title**="Welcome!">

@\* render fragment \*@

<**ChildContent**> HI Bro!</**ChildContent**>

@\* <NewChildContent> How are you?</NewChildContent> \*@

</**ChildComponent**>

private ChildComponent childComponent;  
  
<div class="row">

<button class="btn btn-success" @onclick="(() => childComponent.ShowToastrSuccess())">Show Alert</button>

</div>

* It is used to call chilled component method in parent component

DbInitializer :

To create dynamic roles and assign admin role to created user.