1. Create a ASP.NET Core MVC Project and perform following task:

a. Create a controller Controller1 with 3 action methods:

i. Index – this should return string

ii. Details – pass some value and show this in its corresponding view page.

iii. Show - use of ViewData and ViewBag

# 1Creating a Web App & Run

 From the Visual Studio, select Create a new project.

 Select ASP.NET Core Web Application and then select Next.

 Name the project as you like or Lab2

 Choose the location path to save your project.

 Click Create

 Select Web Application(Model-View-Controller), and then select

Create.

 Now, To run the App,

◦ Select Ctrl-F5 to run the app in non-debug mode, Or

◦ Select IIS Express Button

## aAdd a controller

 In Solution Explorer, right-click Controllers > Add > Controller

 In the Add Scaffold dialog box, select Controller Class – Empty

 In the Add Empty MVC Controller dialog, enter Controller1Controller and

select Add.

### i) Index - this should return string

public string Index() {

return "This is my default action...";

}

ii) Details – pass some value and show this in its corresponding view page.

Action method in controller

public IActionResult Details(string detail)

{

ViewData["detail"] = detail;

return View();

}

View of details

@{

ViewData["Title"] = "Details";

}

<h1>@ViewData["detail"]</h1>

Url: <https://localhost:44394/Controller1/Details?detail=Hello>

### iii. Show - use of ViewData and ViewBag

Action method in controller

public IActionResult Show(string name)

{

ViewData["name "] = name;

return View();

}

View of details

@{

ViewData["Title"] = "Show";

}

<h1>@ViewData["name "]</h1>

Url: https://localhost:44394/Controller1/Show?name=bijay&&user\_address=kusunti

# 2. In your project, add new Controller Controller2 with add one action method to send id numeric value and string value to your view page. Then show using razor syntax to display the string value to id number of times.

public class Controller2Controller : Controller

{

public IActionResult Index()

{

return View();

}

public IActionResult Display(int n, string greet)

{

ViewData["n"] = n;

ViewBag.Greet = greet;

return View();

}

}

View/Controller2/Display.cshtml

@{

ViewData["Title"] = "Display";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h1>Display</h1>

<ul>

@for (int i = 0; i < (int)ViewData["n"]; i++)

{

<li>@ViewBag.greet</li>

}

</ul>

Url: https://localhost:44394/Controller2/Display?n=2&&greet=hello

# 3. Now Add model Model1 in your project with these property members – FullName, Email, Salary

## a. Add Data Annotation on Model

## b. Add necessary Model Validation on each property member

public class Model1

{

[Display(Name = "Full Name")]

[Required]

[StringLength(100)]

public string FullName { get; set; }

[Display(Name = "Email")]

[EmailAddress]

public string Email { get; set; }

[Display(Name = "Salary")]

[Required]

public decimal Salary { get; set; }

}

# 4. On any of the above controller, create action methods with HttpGet and HttpPost:

## a. show how you can bind the Model,

## b. check the Model State is Valid or not and

## c. show validation message in your view page.