

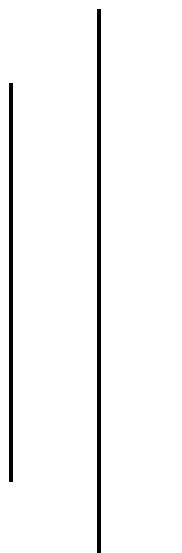


**(Affiliated to Tribhuvan University)**

A LAB REPORT OF NET CENTRIC COMPUTING

**Lab Report on: BEAUTY SALON APPOINTMENT SYSTEM**

**Lab Report no:02**



**SUBMITTED BY:**

Name: Manisha kc

Faculty: Bsc.CSIT ,6<sup>th</sup> sem

Symbol NO:20809/075

**SUBMITTED TO:**

Dipendra KM

(Department of CSIT)

Signature:

## **Contents**

1.Introduction To Template Inheritance

2.Introduction To Controller

3.Lesson Learned

4.Razor Page Code

- ✓ Auth/Signup.cshtml
- ✓ Auth/Login.cshtml
- ✓ Auth/AboutUs.cshtml
- ✓ Auth/ AppointmentDetails.cshtml
- ✓ Auth/AddService.cshtml
- ✓ Auth/Gallery.cshtml

5.Controller Action Code

- ✓ Controller/AuthController

6.Output

- ✓ Home Page
- ✓ Signup Page
- ✓ Login Page
- ✓ AboutUs Page
- ✓ AppointmentDetails Page
- ✓ AddService
- ✓ Gallery

7.Conclusion

## 1.Introduction To Template Inheritance

A template is a file that serves as a starting point for a new document. When we open a template, it is pre-formatted in some way. So, ASP.NET MVC template is a project starter template for ASP.NET MVC based solutions (mainly for multi-paged enterprise solutions, which could change in the future if there will be need for it). We can find and download many templates. In object-oriented programming, we can use inheritance to facilitate code reuse that may be template too. The derived class inherits the fields and methods of the downloaded template. This helps with the code reusability in C#.

## 2.Introduction to Controller

A controller is responsible for controlling the way that a user interacts with an MVC application. A controller contains the flow control logic for an ASP.NET MVC application. A controller determines what response to send back to a user when a user makes a browser request.

Suppose we are entering the following URL in the browser:

`http://localhost:7099/Sample`

It means that the Controller is called HomeController. The browser is making a request to the controller so now the HomeController is responsible for providing the response to the browser. It might be possible that the controller returns a specific view back to the browser or the controller might redirect the user to another controller.

## 3.Lesson Learned

ASP.NET is a free web framework for building websites and web applications on .NET Framework using HTML, CSS, and JavaScript. ASP.NET MVC is a web framework based on Model-View-Controller (MVC) architecture. Developers can build dynamic web applications using ASP.NET MVC framework that enables a clean separation of concerns, fast development, and TDD friendly.

The Controller in MVC architecture handles any incoming URL request. The Controller is a class, derived from the base class `System.Web.Mvc.Controller`. Controller class contains public methods called **Action** methods. Controller and its action method handles incoming browser requests, retrieves necessary model data and returns appropriate responses.`System.Web.Mvc.Controller`.

All the public methods of the Controller class are called Action methods. They are like any other normal methods with the following restrictions:

1. Action method must be public. It cannot be private or protected
2. Action method cannot be overloaded
3. Action method cannot be a static method.

With the release of new ASP.NET Core 2 framework, Microsoft and its community has provided us with a brand-new alternative for the MVC (Model-View-Controller) approach. Microsoft has named it Razor Pages, and while it's a little bit different approach, but it's still similar to MVC in some ways.

A Razor Page is very similar to ASP.NET MVC's view component. It has basically same syntax and functionality as MVC.

The key difference is that the model and controller code is also included within the Razor Page itself. It is more an MVVM (Model-View-ViewModel) framework. It enables two-way data binding and a simpler development experience with isolated concerns.

## 4. Razor page code

### Auth/Signup.cshtml

```
@{
    ViewData["Title"] = "Signup";
}
<div class="row justify-content-md-center">
    <div class="col-6">
        <div class="card">
            <div class="card-body">
                <h3>Signup</h3>
                <form method="post">
                    <div class="mb-3">
                        <label for="exampleFormControlInput1" class="form-label">Email
address</label>
                        <input type="email" class="form-control" id="exampleFormControlInput1"
placeholder="name@example.com">
                    </div>
                    <div class="mb-3">
                        <label for="fullname" class="form-label">Full Name</label>
                        <input type="text" class="form-control" id="fullname" placeholder="name">
                    </div>
                    <div class="mb-3">
                        <label for="pass1" class="form-label">Password</label>
                        <input type="password" class="form-control" id="pass1">
                    </div>
                    <div class="mb-3">
                        <label for="pass2" class="form-label">Password Confirmation</label>
                        <input type="password" class="form-control" id="pass2">
                    </div>
                    <hr/>
                    <button type="submit" class="btn btn-success">Signup</button>
                </form>
            </div>
        </div>
    </div>
</div>
```

### Auth/Login.cshtml

```
@{
    ViewData["Title"] = "Login";
}
<div class="row justify-content-md-center">
    <div class="col-6">
        <div class="card">
            <div class="card-body">
                <h3>Login</h3>
                <div class="mb-3">
                    <label for="exampleFormControlInput1" class="form-label">Email
address</label>
                    <input type="email" class="form-control" id="exampleFormControlInput1"
placeholder="name@example.com">
                </div>

                <div class="mb-3">
                    <label for="pass1" class="form-label">Password</label>
                    <input type="password" class="form-control" id="pass1">
                </div>

                <hr/>
                <button type="submit" class="btn btn-success">Login</button>

            </div>
        </div>
    </div>
</div>
```

### Auth/AboutUs.cshtml

```
@{
    ViewData["Title"] = "About Us";
}
<div class="card" style="width:1200px">

    <div class="card-img-overlay">

        <h4 class="card-title" align="center"><b>About Us</b></h4>

        <p class="card-text">Electronic Bidding management System will facilitate the
old bidding processes through electronic means basically by using online forms
through the internet. An electronic Bidding Management System is an important
process for any business that helps that corporation to search for a reliable
and affordable contractor according to its need.

        Suppose take an example of a building and construction company and
clients which can invites proposals from different vendors to propose an
estimated cost of a contract. This proposal will include detailed costs of the
contract with a time frame. In such a way, the company can choose the best
suited vendor on the result of the proposals. Such as we are Implementing an
Electronic Bidding management System by using basic Web services for the
automation processes.
    </p>
    </div>
</div>
```

## Auth/ AppointmentDetails.cshtml

```
@{
    ViewData["Title"] = "Appontment Details";
}
<div class="row">
    </div>
    <div class="col-8">
        <form method="post">
            <div class="mb-3">
                <label for="month" class="form-label">Month</label>
                <input type="month" class="form-control" id="month">
            </div>
            <div class="mb-3">
                <label for="week" class="form-label">Week</label>
                <input type="week" class="form-control" id="week">
            </div>
            <div class="mb-3">
                <label for="datedetail" class="form-label">Appointment Date</label>
                <input type="date" class="form-control" >
            </div>
            <div class="mb-3">
                <label for="timedetail" class="form-label"> Appointment Time</label>
                <input type="time" class="form-control">
            </div>
            <div class="mb-3">
                <label for="phone" class="form-label">Phome Number</label>
                <input type="number" class="form-control" id="number">
            </div>
            <div class="mb-3">
                <label for="email" class="form-label">Email</label>
                <input type="email" class="form-control" id="email"
placeholder="example@gmail.com">
            </div>
            <div class="mb-3">
                <label for="extradetails" class="form-label">Extra Details</label>
                <textarea class="form-control" id="details" rows="3"></textarea>
            </div>
            <hr/>
            <button type="submit" class="btn btn-success">Signup</button>
        </form>

    </div>
</div>
```

## Auth/AddService.cshtml

```
@{
    ViewData["Title"] = "Add Service";
}
    <div class="row">
        <div class="col-8">
            <h3>Add Service</h3>
            <hr/>
            <form method="post">
                <div class="mb-3">
                    <label for="datetime" class="form-label">Date Time</label>
                    <input type="datetime-local" class="form-control">
                </div>
            </form>
        </div>
    </div>
```

```

<div class="mb-3">
  <label for="fullname" class="form-label">Hair Service</label>
  <select class="form-control" name="type">
    <option value="Haircut">Haircut</option>
    <option value="HairTreatment">HairTreatment</option>
    <option value="Straight">Straight</option>
    <option value="Curls">Curls</option>
    <option value="KeratinStraight">KeratinStraight</option>
    <option value="BridalHairStyle">BridalHairStyle</option>
  </select>
</div>

<div class="mb-3">
  <label for="fullname" class="form-label">Makeup</label>
  <select class="form-control" name="type">
    <option value="PartyMakeup">PartyMakeup</option>
    <option value="BridalMakeup">BridalMakeup</option>
    <option value="EngagementMakeup">EngagementMakeup</option>
    <option value="NormalMakeup">NormalMakeup</option>
  </select>
</div>

<div class="mb-3">
  <label for="fullname" class="form-label">HairRemove</label>
  <select class="form-control" name="type">
    <option value="Threading">Threading</option>
    <option value="Waxing">Waxing</option>
    <option value="VeetCream">VeetCream</option>
    <option value="LazorTreatment">LazorTreatment</option>
    <option value="Razor">Razor</option>
  </select>
</div>

<div class="mb-3">
  <label for="fullname" class="form-label">Others</label>
  <select class="form-control" name="type">
    <option value="Spa">Spa</option>
    <option value="Menicure">Menicure</option>
    <option value="Pedicure">Pedicure</option>
    <option value="NailExtension">NailExtension</option>
  </select>
</div>
<hr />
<button type="submit" class="btn btn-success">Signup</button>
  </form>
</div>
</div>
</div>

```

### Auth/Gallery.cshtml

```

@{
  ViewData["Title"] = "Gallery";
}

<html>
<head>
  <style>
    div.gallery {
      margin: 5px;
      border: 1px solid #ccc;
      float: left;
      width: 200px;
    }
  </style>

```

```

        div.gallery:hover {
            border: 1px solid #777;
        }

        div.gallery img {
            width: 100%;
            height: 79%;
        }
    </style>
</head>
<body>

    <div class="gallery">
        <a target="_blank" href="image/1.jpeg">
            
        </a>
    </div>

    <div class="gallery">
        <a target="_blank" href="~/image/11.jpeg">
            
        </a>
    </div>

    <div class="gallery">
        <a target="_blank" href="image/8.jpeg">
            
        </a>
    </div>

    <div class="gallery">
        <a target="_blank" href="image/4.jpeg">
            
        </a>
    </div>

    <div class="gallery">
        <a target="_blank" href="image/5.jpeg">
            
        </a>
    </div>

    <div class="gallery">
        <a target="_blank" href="~/image/12.jpeg">
            
        </a>
    </div>

    <div class="gallery">
        <a target="_blank" href="~/image/13.jpeg">
            
        </a>
    </div>

    <div class="gallery">
        <a target="_blank" href="~/image/14.jpeg">
            
        </a>
    </div>

    <div class="gallery">
        <a target="_blank" href="~/image/9.jpeg">
            

```



```

        </a>
    </div>
    <div class="gallery">
        <a target="_blank" href="~/image/10.jpeg">
            
        </a>
    </div>
</body>
</html>

```

## 5.Controller Action Code

### Controllers/AuthController.cs

```

using Microsoft.AspNetCore.Mvc;

namespace BeautySalon.Controllers
{
    public class AuthController : Controller
    {
        public IActionResult Signup()
        {
            return View();
        }
        public IActionResult Login()
        {
            return View();
        }
        public IActionResult AboutUs()
        {
            return View();
        }
        public IActionResult AddService()
        {
            return View();
        }
        public IActionResult AppointmentDetails()
        {
            return View();
        }
        public IActionResult Gallery()
        {
            return View();
        }
    }
}

```

### Layout.cshtml

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - BeautySalon</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css"
/>
    <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
    <link rel="stylesheet" href="~/BeautySalon.styles.css" asp-append-
version="true" />
</head>

```

```

<body>
  <header>
    <nav class="navbar navbar-expand-sm navbar-togglerable-sm navbar-light
bg-white border-bottom box-shadow mb-3">
      <div class="container-fluid">
        <a class="navbar-brand" asp-area="" asp-controller="Home"
asp-action="Index">BeautySalon</a>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
          aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="navbar-collapse collapse d-sm-inline-flex
justify-content-between">
          <ul class="navbar-nav flex-grow-1">
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-
controller="Home" asp-action="Index">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-
controller="Auth" asp-action="AboutUs">AboutUs</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-
controller="Auth" asp-action="Gallery">Gallery</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-
controller="Auth" asp-action="AppointmentDetails">AppontmentDetails</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-dark" asp-area="" asp-
controller="Auth" asp-action="AddService">AddService</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" asp-area="" asp-
controller="Auth" asp-action="Login">Login</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" asp-area="" asp-
controller="Auth" asp-action="Signup">Signup</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>
  <div class="container">
    <main role="main" class="pb-3">
      @RenderBody()
    </main>
  </div>
  <script src="~/lib/jquery/dist/jquery.min.js"></script>
  <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
  <script src="~/js/site.js" asp-append-version="true"></script>
  @await RenderSectionAsync("Scripts", required: false)
</body>
</html>

```

## 6.OUTPUT

# Home Page

BeautySalon   Home   AboutUs   Gallery   AppontmentDetails   AddService   Login   Signup

## Welcome to our website.

We aim at elevating our guests inside and out. Be your best self, live your best life.



# Signup Page

BeautySalon   Home   AboutUs   Gallery   AppontmentDetails   AddService   Login   Signup

## Signup

Email address

name@example.com

Full Name

name

Password

Password Confirmation

Signup

# Login Page

BeautySalon   Home   AboutUs   Gallery   AppontmentDetails   AddService   Login   Signup

## Login

Email address

name@example.com

Password

Login

## AboutUs Page

[BeautySalon](#) [Home](#) [AboutUs](#) [Gallery](#) [AppontmentDetails](#) [AddService](#) [Login](#) [Signup](#)

### About Us

Salon system is that important detail that helps your beauty shop to stand out among competitors. Nowadays, if a beauty salon doesn't have online booking, it loses clients. Often if a person is ready to book a service, there is a possibility that he or she won't be comfortable with calling, arranging time, etc. Sometimes people don't even want to talk to the administrator. Plus, customers would like to book a service after their work and out of your work hours (no one will call to book a manicure at midnight but to book it online... why not). As the system works 24/7 you won't lose any minute and any client. When a client is not able to book a service at the desired moment, there is a huge possibility that he'll change the salon. Why would you wait to book something, if the nearest beauty salon offers booking right now? The final is pathetic – a beauty shop without salon scheduling software loses. But don't worry, you can always change things and select the software that suits your beauty salon best. Let's speak about a salon booking system – a handy tool that allows online booking 24/7. Clients will find your services even when you sleep.



## AppointmentDetails Page

[BeautySalon](#) [Home](#) [AboutUs](#) [Gallery](#) [AppontmentDetails](#) [AddService](#) [Login](#) [Signup](#)

Month	<input type="text" value="-----"/>	📅
Week	<input type="text" value="Week --, ----"/>	📅
Appointment Date	<input type="text" value="dd/mm/yyyy"/>	📅
Appointment Time	<input type="text" value="--:--"/>	🕒
Phone Number	<input type="text"/>	
Email	<input type="text" value="example@gmail.com"/>	
Extra Details	<input type="text"/>	

## AddService

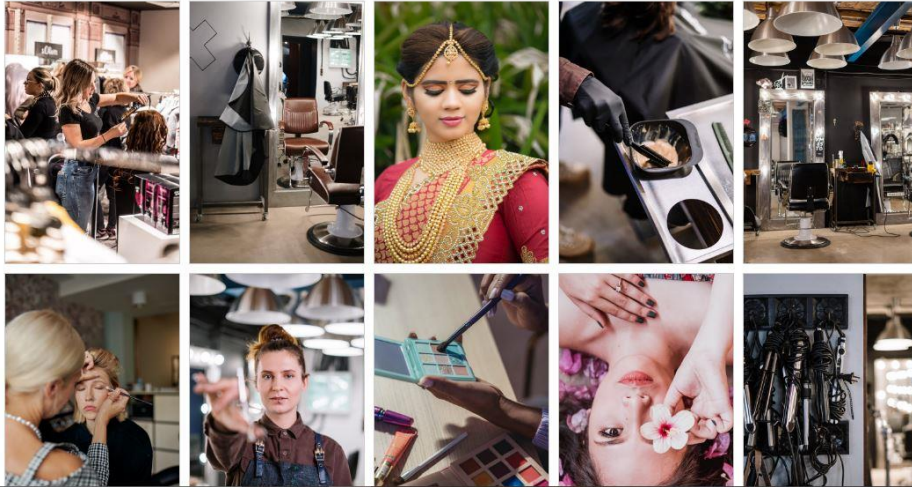
[BeautySalon](#) [Home](#) [AboutUs](#) [Gallery](#) [AppontmentDetails](#) [AddService](#) [Login](#) [Signup](#)

### Add Service

Date Time	<input type="text" value="dd/mm/yyyy --:--"/>	📅
Hair Service	<input type="text" value="Haircut"/>	
Makeup	<input type="text" value="PartyMakeup"/>	
HairRemove	<input type="text" value="Threading"/>	
Others	<input type="text" value="Spa"/>	
<input type="button" value="Signup"/>		

## Gallery

[BeautySalon](#) [Home](#) [AboutUs](#) [Gallery](#) [AppontmentDetails](#) [AddService](#) [Login](#) [Signup](#)



## 7. CONCLUSION

Hence, in this way, In this lab session, we learnt about routing Model-View-Controller patterns, Razor pages, template inheritance, and so on. We also started a small project implementation of these concepts. This lab session focused primarily on MVC patterns, layouts, template inheritance, using views and rendering views using the Razor view. This lab session also marked the start of a project using ASP.NET core.