**PRACTICAL 14**

**Aim: Develop ASP.NET Core MVC application to demonstrate model validation with implicit validation and explicit validation.**

**Code:**

**HomeController.cs :** (Controllers / HomeController.cs)

using Microsoft.AspNetCore.Mvc;

using PRACTICAL14.Models;

namespace ValidationDemoApp.Controllers {

public class HomeController : Controller {

[HttpGet]

public IActionResult Index() {

return View(new Person());

}

[HttpPost]

public IActionResult Index(Person person) {

if (ModelState.IsValid) {

return RedirectToAction("Result", person);

}

return View(person);

}

public IActionResult Result(Person person) {

return View(person);

}

}

}

**Person.cs :** (Models / Person.cs)

using System.ComponentModel.DataAnnotations;

namespace PRACTICAL14.Models {

public class Person {

[Required(ErrorMessage = "Name is Required.")]

[RegularExpression(@"^[a-zA-Z\s]{1,26}$", ErrorMessage = "Name Must be up to 26 Alphabets and Spaces Only.")]

public string Name { get; set; } = string.Empty;

[Required(ErrorMessage = "Email is required.")]

[EmailAddress(ErrorMessage = "Invalid Email Format.")]

[RegularExpression(@"^[^@\s]+@(icloud|gmail|yahoo|rediffmail)\.com$", ErrorMessage = "Email must be from icloud, gmail, yahoo, or rediffmail.")]

public string Email { get; set; } = string.Empty;

[Required(ErrorMessage = "Age is required.")]

[Range(18, 100, ErrorMessage = "Age Must be Between 18 & 100.")]

public int Age { get; set; }

[Required(ErrorMessage = "Phone Number is Required.")]

[RegularExpression(@"^\+?[1-9]\d{1,14}$", ErrorMessage = "Invalid Phone Number Format.")]

public string Phone { get; set; } = string.Empty;

}

}

**Index.cshtml :** (Views / Home / Index.cshtml)

@model PRACTICAL14.Models.Person

@{

ViewData["Title"] = "Model Validation";

}

<style>

.form-control-custom {

width: 200px;

}

</style>

<h2>Model Validation Form</h2>

<form method="post" asp-action="Index">

<div class="form-group">

<label for="Name">Name:</label>

<input type="text" id="Name" name="Name" class="form-control form-control-sm form-control-custom" value="@Model.Name" />

<span asp-validation-for="Name" class="text-danger"></span>

</div><br>

<div class="form-group">

<label for="Email">Email:</label>

<input type="email" id="Email" name="Email" class="form-control form-control-sm form-control-custom" value="@Model.Email" />

<span asp-validation-for="Email" class="text-danger"></span>

</div><br>

<div class="form-group">

<label for="Age">Age:</label>

<input type="number" id="Age" name="Age" class="form-control form-control-sm form-control-custom" value="@Model.Age" />

<span asp-validation-for="Age" class="text-danger"></span>

</div><br>

<div class="form-group">

<label for="Phone">Phone Number:</label>

<input type="tel" id="Phone" name="Phone" class="form-control form-control-sm form-control-custom" value="@Model.Phone" />

<span asp-validation-for="Phone" class="text-danger"></span>

</div><br>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

@if (!ViewData.ModelState.IsValid) {

<div class="text-danger">

<h3>Validation Errors:</h3>

<ul>

@foreach (var error in ViewData.ModelState.Values.SelectMany(v => v.Errors)) {

<li>@error.ErrorMessage</li>

}

</ul>

</div>

}

**Result.cshtml :** (Views / Home / Result.cshtml)

@model PRACTICAL14.Models.Person

@{

ViewData["Title"] = "Submission Result";

}

<h1>Submission Result</h1>

<p>Name: @Model.Name</p>

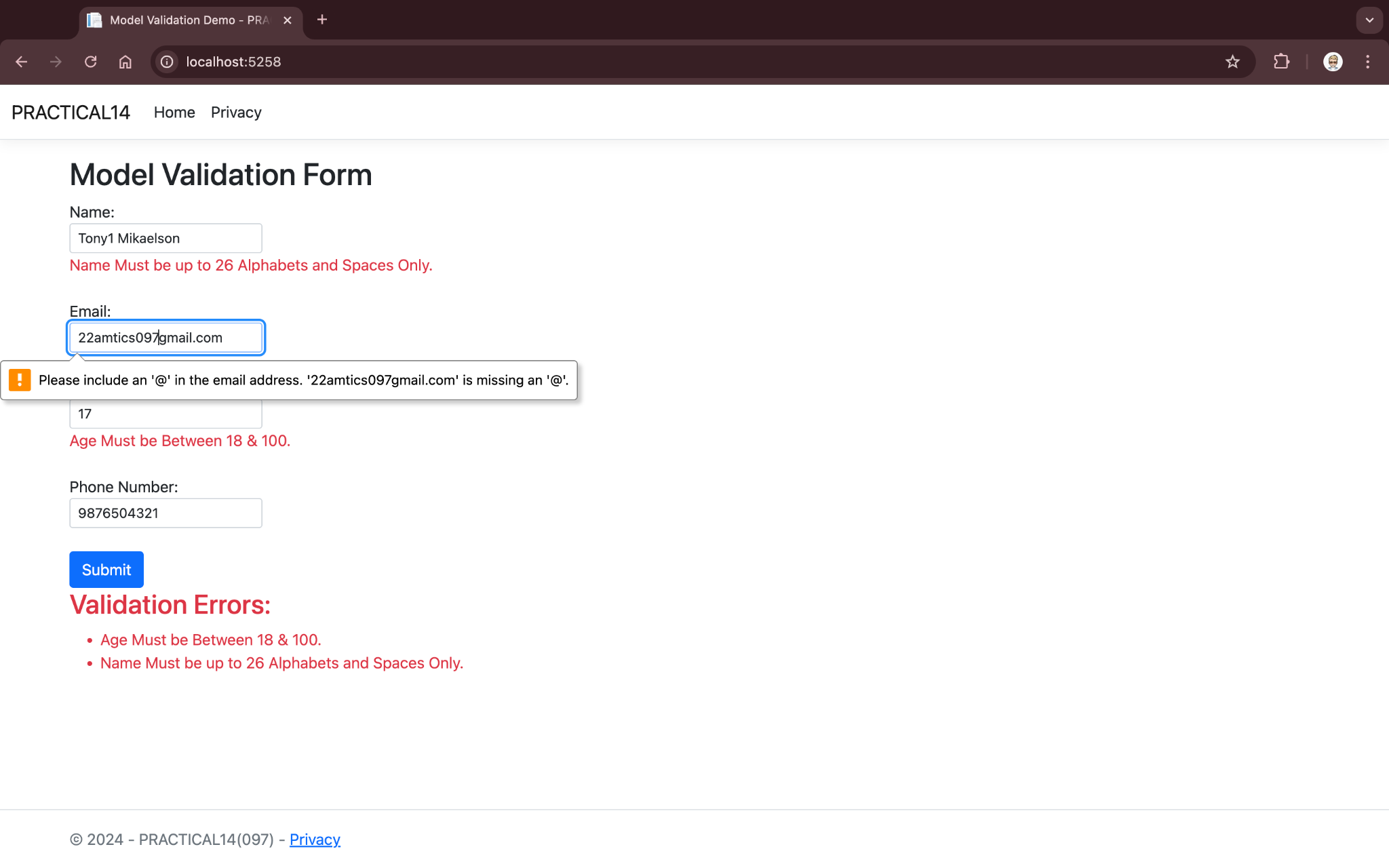
<p>Email: @Model.Email</p>

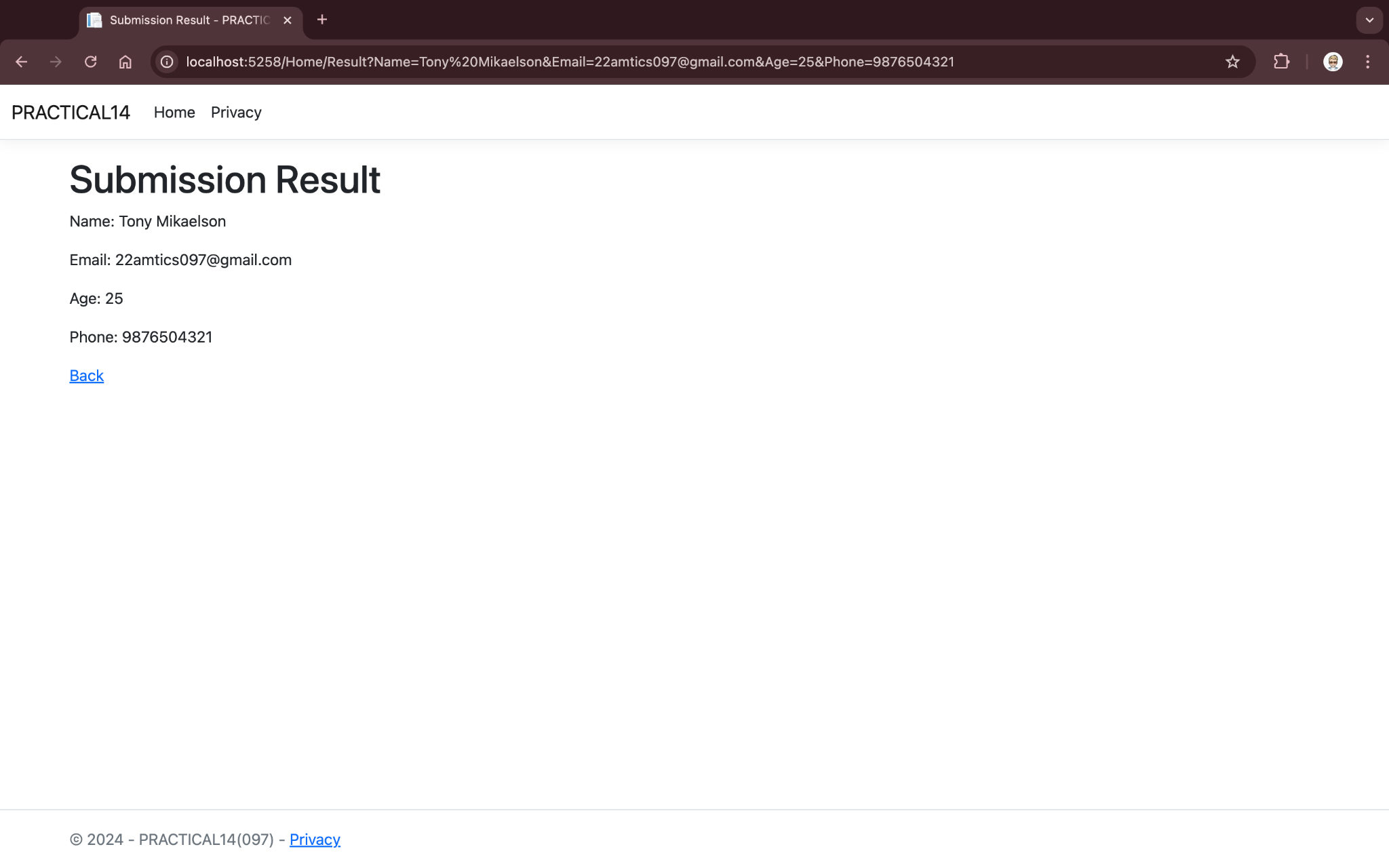
<p>Age: @Model.Age</p>

<p>Phone: @Model.Phone</p>

<a asp-controller="Home" asp-action="Index">Back</a>

**Output:**

****

****