**Program.cs**

using FirstWebApplication.Contexts;

using FirstWebApplication.Interfaces;

using FirstWebApplication.Models;

using FirstWebApplication.Repositories;

using FirstWebApplication.Services;

using Microsoft.EntityFrameworkCore;

namespace FirstWebApplication

{

public class Program

{

public static void Main(string[] args)

{

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddRazorPages();

builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<ShoppingContext>(opts =>

{

opts.UseSqlServer(builder.Configuration.GetConnectionString("conn"));

});

builder.Services.AddScoped<IRepository<int, Product>,ProductRepositoryDb > ();

builder.Services.AddScoped<IProductService, ProductService>();

builder.Services.AddScoped<IRepository<string, Customer>, CustomerRepository>();

builder.Services.AddScoped<ICustomerService, CustomerService>();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Error");

}

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapRazorPages();

app.MapDefaultControllerRoute();

app.Run();

}

}

}

**ShoppingContext.cs**

using FirstWebApplication.Models;

using Microsoft.EntityFrameworkCore;

namespace FirstWebApplication.Contexts

{

public class ShoppingContext : DbContext

{

public ShoppingContext(DbContextOptions options) : base(options)

{

}

public DbSet<Product> Products { get; set; }

public DbSet<Category> Categories { get; set; }

public DbSet<Customer> Customers { get; set; }

}

}

**CustomerController.cs**

using FirstWebApplication.Interfaces;

using FirstWebApplication.Models;

using Microsoft.AspNetCore.Mvc;

namespace FirstWebApplication.Controllers

{

public class CustomerController : Controller

{

private readonly ICustomerService \_customerService;

public CustomerController(ICustomerService customerService)

{

\_customerService = customerService;

}

public IActionResult Index()

{

List<Customer> customers = \_customerService.GetAllCustomers();

return View(customers);

}

public IActionResult Create()

{

return View();

}

[HttpPost]

public IActionResult Create(Customer customer)

{

if (ModelState.IsValid)

{

\_customerService.CreateCustomer(customer);

ViewBag.Message = "Registration successful!";

return RedirectToAction("Login");

}

else

{

ViewBag.Message = "Registration failed. Please enter all fields correctly.";

return View(customer);

}

}

public IActionResult Login()

{

return View();

}

[HttpPost]

public IActionResult Login(string email, string password)

{

if (\_customerService.ValidateCredentials(email, password))

{

return RedirectToAction("Index", "Customer");

}

else

{

ModelState.AddModelError(string.Empty, "Incorrect username or password");

return View();

}

}

}

}

**ICustomerService.cs**

using FirstWebApplication.Models;

namespace FirstWebApplication.Interfaces

{

public interface ICustomerService

{

Customer GetCustomer(string email);

List<Customer> GetAllCustomers();

Customer CreateCustomer(Customer customer);

Customer UpdateCustomer(Customer customer);

void DeleteCustomer(string email);

bool ValidateCredentials(string email, string password);

}

}

**Customer.cs**

using System.ComponentModel.DataAnnotations;

namespace FirstWebApplication.Models

{

public class Customer

{

[Key]

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

[Required]

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

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

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

public int Age { get; set; }

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

}

}

**CustomerRepository.cs**

using FirstWebApplication.Contexts;

using FirstWebApplication.Interfaces;

using FirstWebApplication.Models;

using Microsoft.EntityFrameworkCore;

namespace FirstWebApplication.Repositories

{

public class CustomerRepository : IRepository<string, Customer>

{

private readonly ShoppingContext \_context;

public CustomerRepository(ShoppingContext context)

{

\_context = context;

}

public Customer Add(Customer item)

{

\_context.Customers.Add(item);

\_context.SaveChanges();

return item;

}

public Customer Delete(string key)

{

var customer = Get(key);

\_context.Customers.Remove(customer);

\_context.SaveChanges();

return customer;

}

public Customer Get(string key)

{

return \_context.Customers.FirstOrDefault(c => c.Email == key);

}

public IList<Customer> GetAll()

{

return \_context.Customers.ToList();

}

public Customer Update(Customer item)

{

\_context.Entry(item).State = EntityState.Modified;

\_context.SaveChanges();

return item;

}

}

}

**CustomerService.cs**

using FirstWebApplication.Interfaces;

using FirstWebApplication.Models;

public class CustomerService : ICustomerService

{

private readonly IRepository<string,Customer> \_customerRepository;

public CustomerService(IRepository<string, Customer> customerRepository)

{

\_customerRepository = customerRepository;

}

public Customer GetCustomer(string email)

{

return \_customerRepository.Get(email);

}

public List<Customer> GetAllCustomers()

{

return \_customerRepository.GetAll().ToList();

}

public Customer CreateCustomer(Customer customer)

{

return \_customerRepository.Add(customer);

}

public Customer UpdateCustomer(Customer customer)

{

return \_customerRepository.Update(customer);

}

public void DeleteCustomer(string email)

{

\_customerRepository.Delete(email);

}

public bool ValidateCredentials(string email, string password)

{

var customer = GetCustomer(email);

return customer != null && customer.Password == password;

}

}

**Views**

**Create.cshtml**

@model FirstWebApplication.Models.Customer

<h1>Register</h1>

<**form** **asp-action**="Create" method="post">

<div class="form-group">

<**label** **asp-for**="Email">Email</**label**>

<**input** **asp-for**="Email" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="Password">Password</**label**>

<**input** **asp-for**="Password" **type**="password" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="FirstName">First Name</**label**>

<**input** **asp-for**="FirstName" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="LastName">Last Name</**label**>

<**input** **asp-for**="LastName" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="Age">Age</**label**>

<**input** **asp-for**="Age" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="Phone">Phone</**label**>

<**input** **asp-for**="Phone" class="form-control" />

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

</div>

<!-- Include other form fields and validation messages -->

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

</**form**>

Index.cshtml

@model List<FirstWebApplication.Models.Customer>

<!DOCTYPE html>

<html>

<head>

<title>Customer List</title>

</head>

<body>

<h2>Customer Details</h2>

<table border="1">

<tr>

<th>Email</th>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

<th>Phone</th>

</tr>

@foreach (var customer in Model)

{

<tr>

<td>@customer.Email</td>

<td>@customer.FirstName</td>

<td>@customer.LastName</td>

<td>@customer.Age</td>

<td>@customer.Phone</td>

</tr>

}

</table>

</body>

</html>

Login.cshtml

@model FirstWebApplication.Models.Customer

<h1>Login</h1>

<**form** **asp-action**="Login" method="post">

<div class="form-group">

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

<input type="text" class="form-control" name="email" id="email" required>

</div>

<div class="form-group">

<label for="password">Password</label>

<input type="password" class="form-control" name="password" id="password" required>

</div>

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

</**form**>A

**Program.cs**

using FirstWebApplication.Contexts;

using FirstWebApplication.Interfaces;

using FirstWebApplication.Models;

using FirstWebApplication.Repositories;

using FirstWebApplication.Services;

using Microsoft.EntityFrameworkCore;

namespace FirstWebApplication

{

public class Program

{

public static void Main(string[] args)

{

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddRazorPages();

builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<ShoppingContext>(opts =>

{

opts.UseSqlServer(builder.Configuration.GetConnectionString("conn"));

});

builder.Services.AddScoped<IRepository<int, Product>,ProductRepositoryDb > ();

builder.Services.AddScoped<IProductService, ProductService>();

builder.Services.AddScoped<IRepository<string, Customer>, CustomerRepository>();

builder.Services.AddScoped<ICustomerService, CustomerService>();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Error");

}

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapRazorPages();

app.MapDefaultControllerRoute();

app.Run();

}

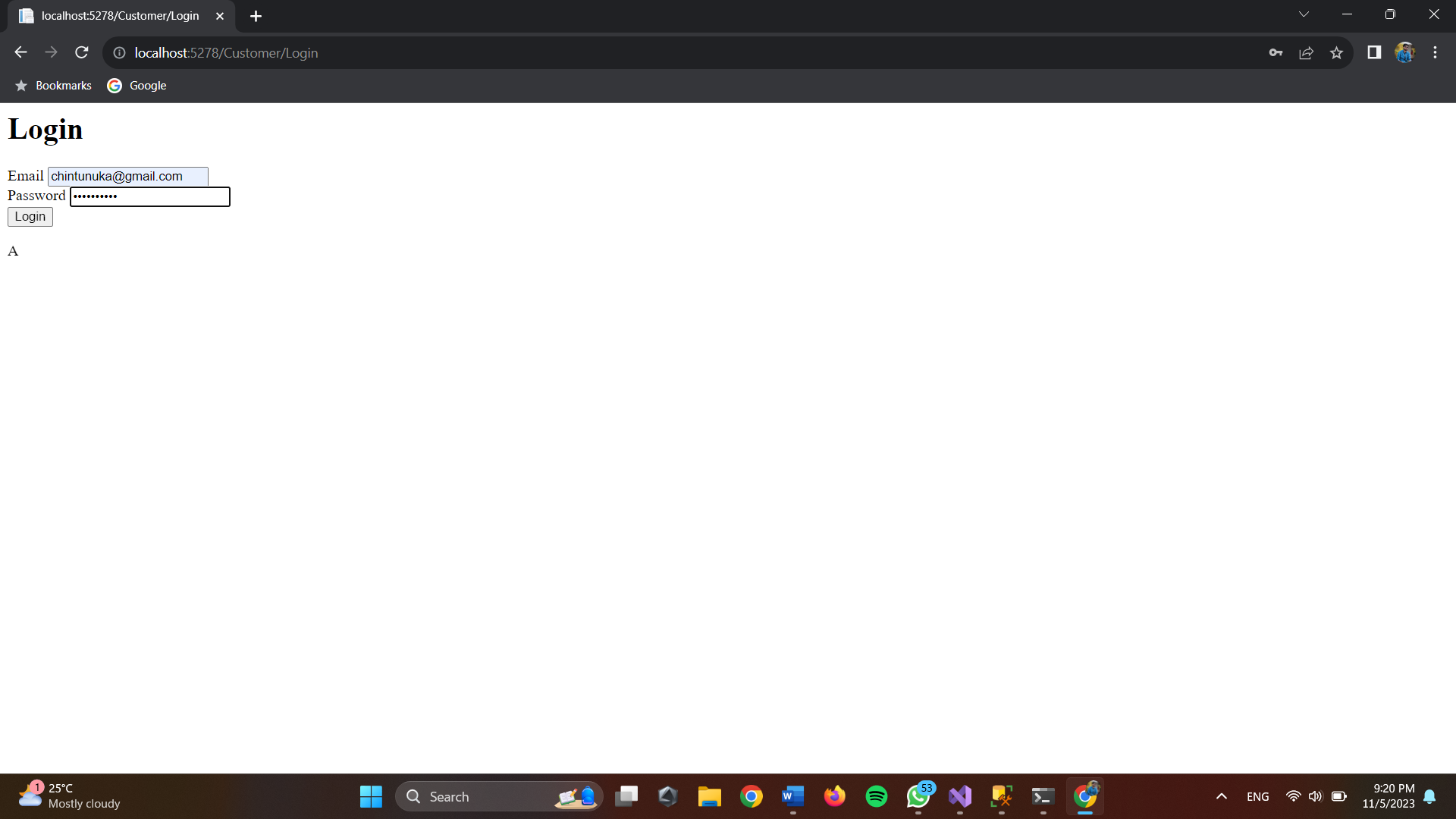
}

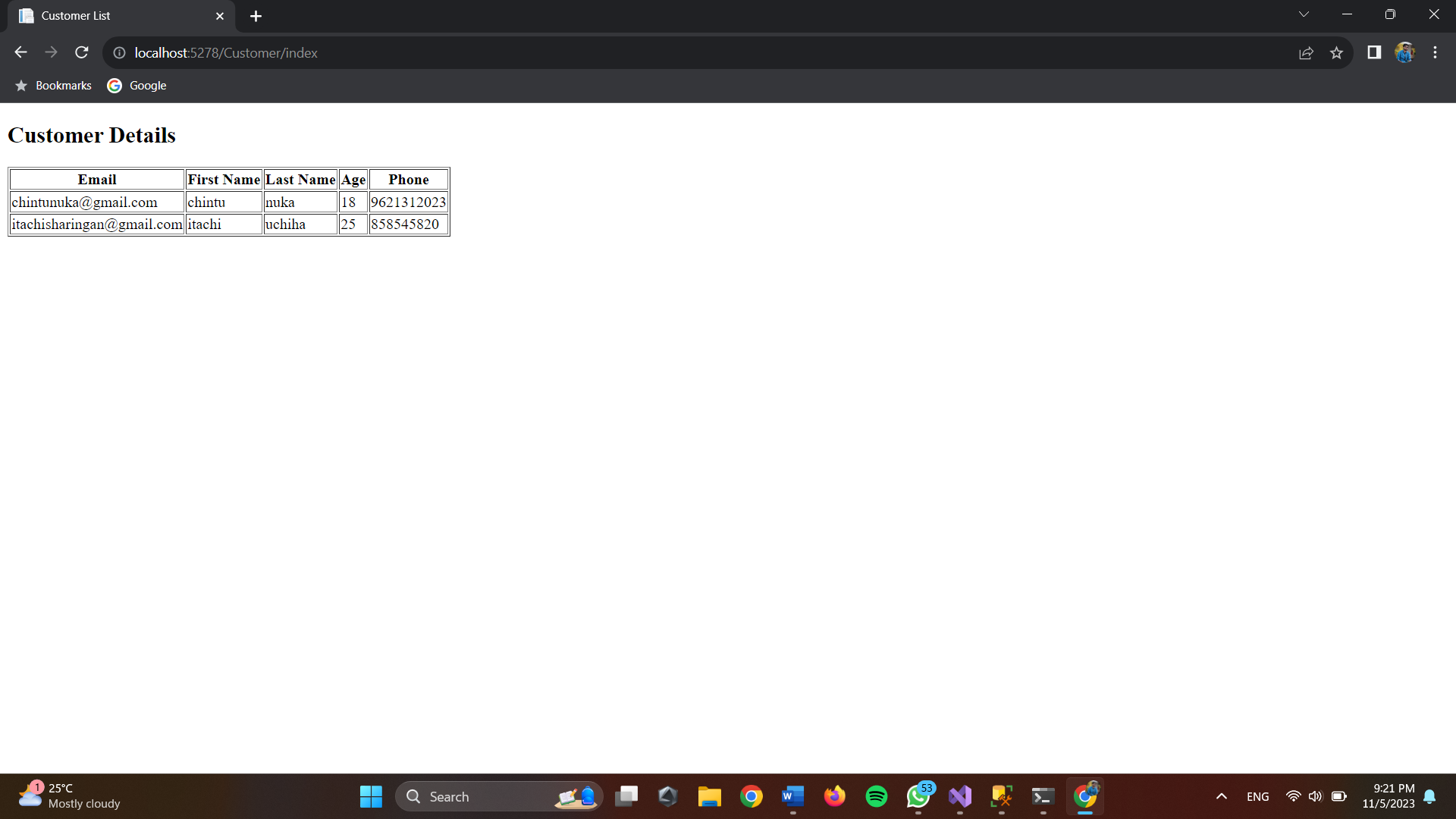
}

**Output**

A screenshot of a computer

Description automatically generated





A computer screen with a white screen

Description automatically generated