

Customer Model

```
using System.ComponentModel.DataAnnotations;

namespace FirstWebApplication.Models
{
    public class Customer
    {
        [Key]
        public string Email { get; set; } = string.Empty;
        public int Age { get; set; } = 0;
        private string password;
        public string Password
        {
            get
            {
                return GetMaskedPassword();
            }
            set
            {
                password = value;
            }
        }
        public string Phone { get; set; } = string.Empty;

        public Customer()
        {
        }

        public Customer(string email, int age, string Password, string phone)
        {
            Email = email;
            Age = age;
            Password = password;
            Phone = phone;
        }

        public bool ComparePassword(string userPassword)
        {
            return (password == userPassword) ? true : false; ;
        }

        string GetMaskedPassword()
        {
            var len = password.Length;
            string maskedPass = password.Substring(0, 2);
            for (int i = 2; i < len; i++)
            {
                maskedPass += "*";
            }
            return maskedPass;
        }

        public override string ToString()
        {
            string maskedPass = GetMaskedPassword();
            return $"Email : {Email}\nAge : {Age}\nPhone : {Phone}\nPassword : {maskedPass}";
        }
    }
}
```

Customer Controller

```
using FirstWebApplication.Interfaces;
using FirstWebApplication.Models;
using Microsoft.AspNetCore.Mvc;

namespace FirstWebApplication.Controllers
{
    public class CustomerController : Controller
    {
        ICustomerService _customerService;
        public CustomerController(ICustomerService customerService)
        {
            _customerService = customerService;
        }
        public IActionResult Index()
        {
            var customer = _customerService.GetCustomers();
            return View(customer);
        }
        public IActionResult Create()
        {
            return View();
        }
        [HttpPost]
        public IActionResult Create(Customer customer)
        {
            var result = _customerService.Register(customer);
            if (result != null)
            {
                return RedirectToAction("Index");
            }
            return View();
        }
    }
}
```

Customer Repository

```
using FirstWebApplication.Contexts;
using FirstWebApplication.Interfaces;
using FirstWebApplication.Models;

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)
        {
            var customer = _context.Customers.SingleOrDefault();
            return customer;
        }

        public IList<Customer> GetAll()
        {
            var customer = _context.Customers.ToList();
            return customer;
        }

        public Customer Update(Customer item)
        {
            var customer = Get(item.Email);
            if (customer != null)
            {
                _context.Entry<Customer>(item).State =
Microsoft.EntityFrameworkCore.EntityState.Modified;
                _context.SaveChanges();
                return customer;
            }
            return null;
        }
    }
}
```

ICustomer Service

```
using FirstWebApplication.Models;

namespace FirstWebApplication.Interfaces
{
    public interface ICustomerService
    {
        public Customer Register(Customer customer);

        public List<Customer> GetCustomers();
    }
}
```

Customer Service

```
using FirstWebApplication.Interfaces;
using FirstWebApplication.Models;
using FirstWebApplication.Repositories;

namespace FirstWebApplication.Services
{
    public class CustomerService : ICustomerService
    {
        IRepository<string, Customer> repository;
        public CustomerService(IRepository<string, Customer> repo)
        {
            repository = repo;
        }
        public Customer GetCustomers(string email)
        {
            var result = repository.Get(email);
            return result == null ? throw new Exception() : result;
        }

        public List<Customer> GetCustomers()
        {
            var customers = repository.GetAll();
            if (customers.Count != 0)
                return customers.ToList();
            throw new Exception();
        }

        public Customer Register(Customer customer)
        {
            var result = repository.Add(customer);
            if (result != null)
                return result;
            throw new Exception();
        }
    }
}
```

Views

Register / Login

@model FirstWebApplication.Models.Customer

```
@{
    ViewData["Title"] = "Create";
}
< head >
    < link href = "~/lib/bootstrap/dist/css/bootstrap.css" rel = "stylesheet" />
    < h1 > Create </ h1 >
    < h4 > Customer </ h4 >
    < hr />
    < style >
        head {
padding: 15px;
}
body {
        padding: 30px;
    }
    </ style >
</ head >
< body >
    < div class= "row" >
        < div class= "col-md-4" >
            < form asp - action = "Create" >
                < div asp - validation - summary = "ModelOnly" class= "text-danger" ></ div >
                < div class= "form-group mb-3" >
                    < label asp -for= "Email" class= "control-label" > Email </ label >
                    < input asp -for= "Email" type = "email" class= "form-control" />
                    < span asp - validation -for= "Email" class= "text-danger" ></ span >
                </ div >
                < div class= "form-group mb-3" >
                    < label asp -for= "Age" class= "control-label" ></ label >
                    < input asp -for= "Age" type = "number" class= "form-control" />
                    < span asp - validation -for= "Age" class= "text-danger" ></ span >
                </ div >
                < div class= "form-group mb-3" >
                    < label asp -for= "Password" class= "control-label" ></ label >
                    < input asp -for= "Password" type = "password" class= "form-control" />
                    < span asp - validation -for= "Password" class= "text-danger" ></ span >
                </ div >
                < div class= "form-group mb-3" >
                    < label asp -for= "Phone" class= "control-label" ></ label >
                    < input asp -for= "Phone" type = "number" class= "form-control" />
                    < span asp - validation -for= "Phone" class= "text-danger" ></ span >
                </ div >
                < div class= "form-group" >
                    < input type = "submit" value = "Create" class= "btn btn-primary" />
                </ div >
            </ form >
        </ div >
    </ div >

    < div >
        < a asp - action = "Index" > Back to List</a>
    </div>
</body>

@section Scripts {
    @{
        await Html.RenderPartialAsync("_ValidationScriptsPartial");
    }
}
```

Index

```
@*Strongly type the view *@
@model IEnumerable<FirstWebApplication.Models.Customer>
<html>
<head>
    <link href="~/lib/bootstrap/dist/css/bootstrap.css" rel="stylesheet" />
    <style>
        .prod {
            height: fit - content;
float: left;
margin: 10px;
        }
    </ style >
</ head >
< body >
    @foreach(var item in Model)
    {
        < div >
            < div class= "card prod" style = "width: fit-content;" >
                < div class= "card-body" >
                    < h5 class= "card-title" > Email: @item.Email </ h5 >
                    < p class= "card-text" > Phone: @item.Phone </ p >
                    < p class= "card-text" > Password: @item.Password </ p >
                    < p class= "card-text" > Age: @item.Age </ p >
                </ div >
            </ div >
        </ div >
    }
</ body >
</ html >
```