

1.

Create the model

```
public class Customer
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Email { get; set; }
    public string PhoneNumber { get; set; }
}
```

2.

Migrate to create table

```
dotnet ef migrations add InitialCreate
```

```
dotnet ef database update
```

3.

Ensure DI

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<ApplicationDbContext>(options =>
        options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

    services.AddScoped<ICustomerRepository, CustomerRepository>();
}
```

4.

Repository pattern that does the CRUD operation

```
public interface ICustomerRepository
{
    IEnumerable<Customer> GetAll();
    Customer GetById(int id);
    void Add(Customer customer);
    void Update(Customer customer);
    void Delete(int id);
}
```

```
public class CustomerRepository : ICustomerRepository
{
    private readonly ApplicationDbContext _context;

    public CustomerRepository(ApplicationDbContext context)
    {
        _context = context;
    }

    public IEnumerable<Customer> GetAll()
    {
        return _context.Customers.ToList();
    }

    public Customer GetById(int id)
    {
        return _context.Customers.FirstOrDefault(c => c.Id == id);
    }
}
```

```
public void Add(Customer customer)
```

```
{  
    _context.Customers.Add(customer);  
    _context.SaveChanges();  
}
```

```
public void Update(Customer customer)
```

```
{  
    _context.Customers.Update(customer);  
    _context.SaveChanges();  
}
```

```
public void Delete(int id)
```

```
{  
    var customer = _context.Customers.Find(id);  
    if (customer != null)  
    {  
        _context.Customers.Remove(customer);  
        _context.SaveChanges();  
    }  
}
```

5.

View for Add and displaying Customers

```
public class CustomerController : Controller
{
    private readonly ICustomerRepository _customerRepository;

    public CustomerController(ICustomerRepository customerRepository)
    {
        _customerRepository = customerRepository;
    }

    public IActionResult Index()
    {
        var customers = _customerRepository.GetAll();
        return View(customers);
    }

    public IActionResult Add()
    {
        return View();
    }

    [HttpPost]
    public IActionResult Add(Customer customer)
    {
        if (ModelState.IsValid)
        {
            _customerRepository.Add(customer);
            return RedirectToAction("Index");
        }
    }
}
```

```
return View(customer);
```

```
}
```

```
}
```