[**1. SECTION – 1 : Cart-Service-Setup & Integration-Using-API-Gateway** 1](#_Toc173098118)

[**2. SECTION – 2 : Payment-Service-Setup & Integration-Using-API-Gateway** 6](#_Toc173098119)

[**3. SECTION – 3 : Call Cart and Payment Service API by UI** 7](#_Toc173098120)

# **1. SECTION – 1 : Cart-Service-Setup & Integration-Using-API-Gateway**

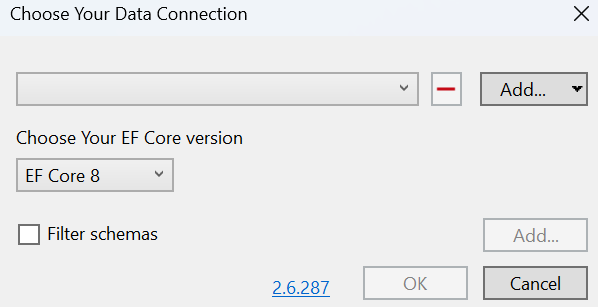
**Folder and File Structure:**

**Get Products from Database.**

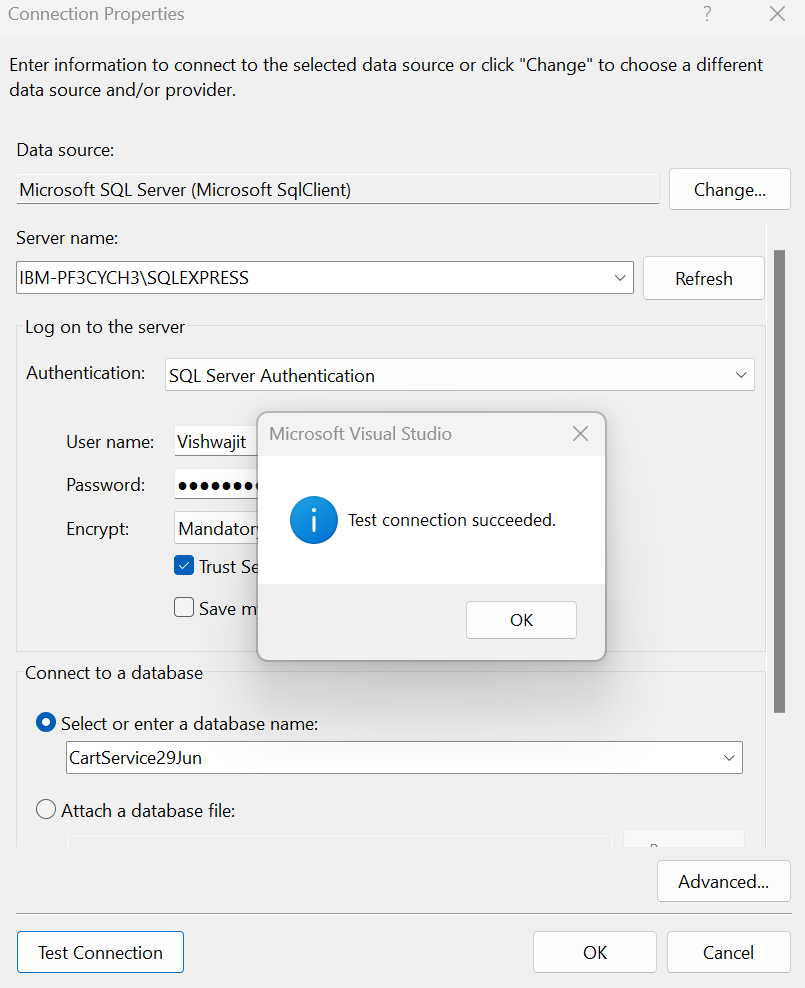
1. Add CatalogService under FrontendServices > WebApp > HttpClients and ProductModel.cs, AddressModel.cs, CartItemModel.cs, CartModel.cs, PaymentConfirmModel.cs, PaymentModel.cs, RazorPayOrder.cs, TransactionModel.cs under Models Folder.
2. Add CatalogService service to the container in Program.cs file.
3. Home Controller call the CatalogService for Index action method to fetch products. Add a respective list view for the Index action method.
4. Add cart.js file under wwwroot > js folder. Add css under wwwroot > css > site.css.
5. Except SupportApp set all projects as start up projects.

**Add Cart to Database.**

1. Add CartService project (ASP.NET Core Web API) under BackendServices Folder.
2. Add Cart database name as **CartService29Jun** in database.
3. Create Database folder for AppDbContext under CartService project and right click > EF Core Power Tools > Reverse Engineer >



Add > Add Database Connection.

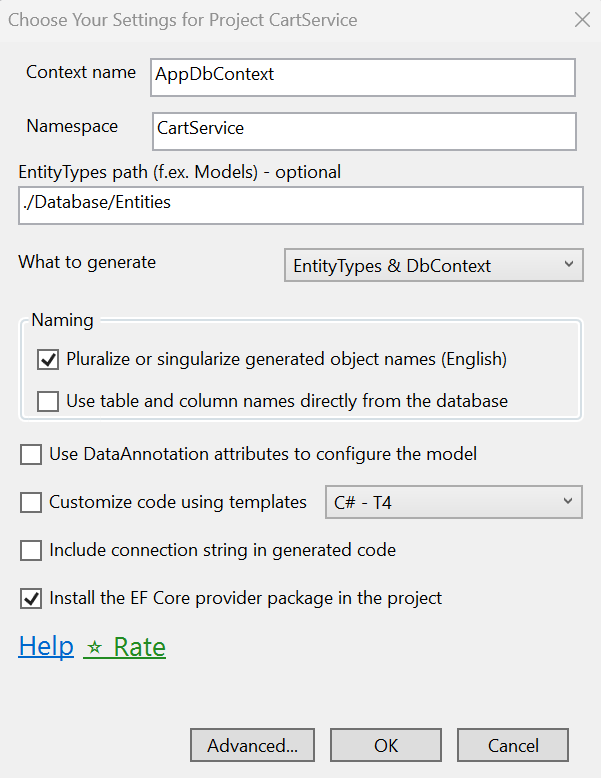


Ok >

A screenshot of a computer

Description automatically generated

Ok >



Advanced.. >

A screenshot of a computer

Description automatically generated

Ok

1. Add Models and Services (Repository) folder for Cart.
2. Add CatalogService under CartService Project > HttpClients
3. Define the connection string in appsettings.json file and add the Services to the container for database connection and serialization in Program.cs file.
4. Add a CartController in CartService Project and add API Gateway for CartService in ocelot.json. (Refer CartService https profile port – 7207. Same port refer in ocelot.json file for CartService gateway.

# **2. SECTION – 2 : Payment-Service-Setup & Integration-Using-API-Gateway**

1. Create PaymentService project (.NET Core Web API template).
2. Create a Database named as **PaymentService29Jun** and add table inside it. (Refer Script inside Database > Schema > Schema.sql )
3. Create AppDbContext using EF Core Power Tool as we did in the CartService. (But hereuse **PaymentService29Jun** database)
4. Add the below NuGet Package and Adding the services (Repository) and Models.

A screenshot of a computer

Description automatically generated

1. Add the connection string and RazorPay Key and Secret configuration in appsetting.json file

A white background with red text

Description automatically generated

1. Add Services to the container for connection string,Scoped services and serialization in Program.cs file of PaymentService project.
2. Add a PaymentController.
3. Include a PaymentGateway in ocelot.json file

# **3. SECTION – 3 : Call Cart and Payment Service API by UI**

1. Create the PaymentService and CartService(http client) in FrontendServices > HttpClients to call the PaymentService , CartService and register the services in service to the container.
2. Add TempDataExtension in WebApp > Helpers folder and BaseController, CartController in WebApp > Controller folder.
3. Add a List page for Index action method of CartController.
4. Add a returnUrl parameter and its condition in Login Post action method of AccountController. Add the returnUrl parameter in Views > Account>Login.cshtml page.



1. Add the cart.js reference in Views > Shared > \_Layout.cshtml file.
2. Add a view for CartController (CheckOut action method) of WebApp project.

A screenshot of a computer program

Description automatically generated

1. Except support App project, set all projects as startup project.
2. Add a PaymentController in WebApp project.
3. Ensure all the namespaces (WebApp.Models) imported in \_ViewImports.cshtml file
4. Create View for PaymentController Index and Status action method.
5. Checkout the below links for Razorpay Web Integration Step for PaymentController Index view.

[Steps to integrate the Standard Checkout form on your website. | Razorpay Docs](https://razorpay.com/docs/payments/payment-gateway/web-integration/standard/integration-steps/)

1. Create View for PaymentController Receipt action method.
2. Add OrderController inside WebApp > Areas > User > Controllers. Add a View for its Index Action method.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a phone number

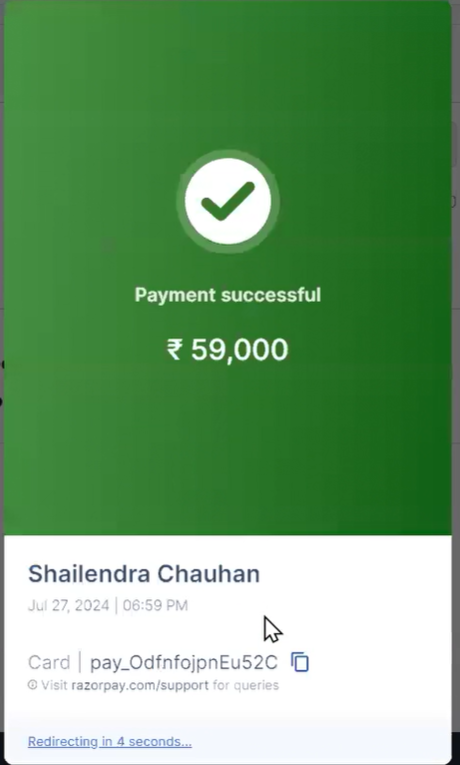
Description automatically generated

A screenshot of a phone

Description automatically generated

A screenshot of a credit card

Description automatically generated



A close up of a computer screen

Description automatically generated