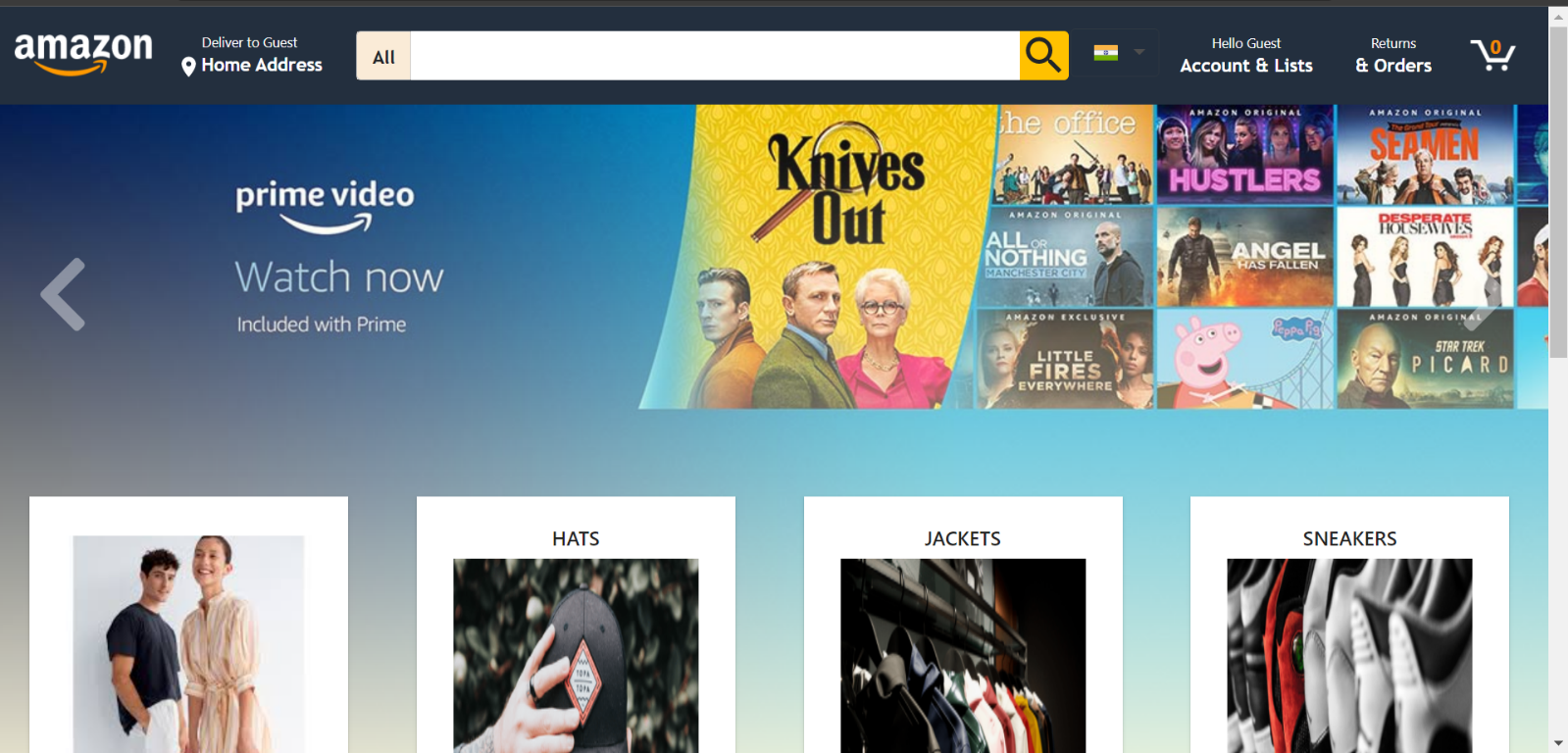
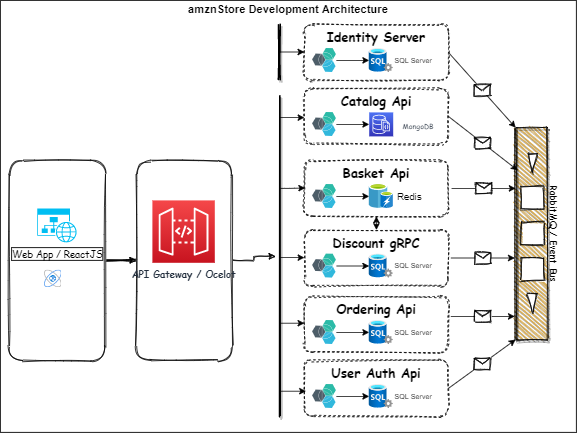
**amznStore Specification**



**System Design**



**Project Architecture**

The solution is built using ASP.NET core as backend and ReactJS as frontend. The system is design based on Microservice Architecture. Other tool stack include Docker, RabbitMQ, Ocelot API Gateway, Atlas MongoDB, Redis, SQL Server.

Frontend application built using ReactJS State Management using Redux, Thunk, Saga, Reselect, Hooks

Local Storage using Persist

API Calls using Axios

Logging using Logger

**Common Library**

All your files and folders are presented as a tree in the file explorer. You can switch from one to another by clicking a file in the tree.

**Common -> Core**

Install-Package Microsoft.AspNetCore.Http

**Common -> Infrastructure**

Install-Package Microsoft.EntityFrameworkCore

Install-Package MailKit

Install-Package Microsoft.AspNetCore.Mvc.Versioning

Install-Package Microsoft.AspNetCore.Mvc.Versioning.ApiExplorer

Install-Package Microsoft.AspNetCore.Mvc.ApiExplorer

Install-Package Swashbuckle.AspNetCore

Install-Package Serilog

**ApiGateway**

**ApiGateway.Web**

Install-Package Ocelot

**User Authentication Service**

The User Database is built on SQL Server. You can either use local SQL Server or download SQL Server on Docker. I have use online SQL Server for the project. You can use following sites for the SQL Server Setup.

[VPS/VDS Hosting. ASP.Net MVC Hosting. Free ASP.Net Hosting (somee.com)](https://somee.com/default.aspx) or

<https://www.gearhost.com/>

The connection string is then stored in secrets.json

**User**

**POST​/api​/v1​/User​/register**

**GET​/api​/v1​/User​/verify-email**

**POST​/api​/v1​/User​/authenticate**

**POST​/api​/v1​/User​/addrole**

**POST​/api​/v1​/User​/refresh-token**

**POST​/api​/v1​/User​/revoke-token**

**POST​/api​/v1​/User​/validate-reset-token**

**POST​/api​/v1​/User​/forgot-password**

**POST​/api​/v1​/User​/reset-password**

**POST​/api​/v1​/User​/change-password**

**GET​/api​/v1​/User**

**GET​/api​/v1​/User​/{id}**

**PUT​/api​/v1​/User​/{id}**

**DELETE​/api​/v1​/User​/{id}**

**HEALTH​/api/health**

**Services -> UserAuthentication -> UserAuthentication.Core**

Install-Package AutoMapper

Install-Package AutoMapper.Extensions.Microsoft.DependencyInjection

Install-Package Microsoft.AspNetCore.Identity

Install-Package Microsoft.AspNetCore.Identity.EntityFrameworkCore

Install-Package Microsoft.AspNetCore.Mvc.Core

**Services -> UserAuthentication -> UserAuthentication.Infrastructure**

Install-Package Microsoft.AspNetCore.Identity.EntityFrameworkCore

Install-Package Microsoft.EntityFrameworkCore

Install-Package Microsoft.EntityFrameworkCore.Design

Install-Package Microsoft.EntityFrameworkCore.SqlServer

Install-Package Microsoft.EntityFrameworkCore.Tools

Install-Package System.IdentityModel.Tokens.Jwt

**Services -> UserAuthentication -> UserAuthentication.Api**

Install-Package Microsoft.AspNetCore.Authentication.JwtBearer

Install-Package Microsoft.EntityFrameworkCore

Install-Package Microsoft.EntityFrameworkCore.Design

Install-Package Microsoft.EntityFrameworkCore.SqlServer

Install-Package Microsoft.EntityFrameworkCore.Tools

Install-Package Microsoft.AspNetCore.Mvc.NewtonsoftJson

Install-Package Serilog.AspNetCore

Install-Package Serilog.Enrichers.Environment

Install-Package Serilog.Enrichers.Process

Install-Package Serilog.Enrichers.Thread

Install-Package Serilog.Settings.Configuration

Install-Package Serilog.Sinks.MSSqlServer

Install-Package AspNetCore.HealthChecks.SqlServer

Install-Package Microsoft.Extensions.Diagnostics.HealthChecks.EntityFrameworkCore

Install-Package AspNetCore.HealthChecks.UI.Client

**Product Catalog Service**

MongoDB is used as the Catalog Database. The database can be setup on <https://cloud.mongodb.com/>

The connection string needs to be stored on secrets.json

**Services -> Catalog -> Catalog.Core**

Install-Package MongoDB.Driver

Install-Package MongoDB.Bson

**Services -> Catalog -> Catalog.Infrastructure**

Install-Package ServiceStack

Install-Package Microsoft.Extensions.Configuration

Install-Package MongoDB.Driver

**Services -> Catalog -> Catalog.Application**

Install-Package AutoMapper

Install-Package Microsoft.Extensions.DependencyInjection

Install-Package AutoMapper.Extensions.Microsoft.DependencyInjection

**Services -> Catalog -> Catalog.Api**

Install-Package AspNetCore.HealthChecks.MongoDb

Install-Package Swashbuckle.AspNetCore

Install-Package Serilog

Install-Package Serilog.Extensions.Logging

Install-Package Serilog.Extensions.Hosting

Install-Package Serilog.Settings.Configuration

Install-Package Newtonsoft.Json

Install-Package Microsoft.AspNetCore.Mvc.NewtonsoftJson

Install-Package Microsoft.AspNetCore.Authentication.JwtBearer

**Services -> Catalog -> Catalog.Test**

Install-Package FluentAssertions

Install-Package Microsoft.AspNetCore.Mvc.Testing

Install-Package RESTFulSense

**Services -> Catalog -> Catalog.Test**

Install-Package FluentAssertions

Install-Package Microsoft.AspNetCore.Mvc.Testing

Install-Package RESTFulSense

**Basket Services**

**Services -> Basket -> Basket.Core**

Install-Package StackExchange.Redis

**Services -> Basket -> Basket.Infrastructure**

Install-Package Microsoft.EntityFrameworkCore

Install-Package Microsoft.Extensions.Configuration

Install-Package Newtonsoft.Json

Install-Package StackExchange.Redis

**Services -> Basket -> Basket.Application**

Install-Package AutoMapper.Extensions.Microsoft.DependencyInjection

**Services -> Basket -> Basket.API**

Install-Package Microsoft.Extensions.Caching.StackExchangeRedis

Install-Package Serilog

Install-Package Serilog.Extensions.Logging

Install-Package Serilog.Extensions.Hosting

Install-Package Serilog.Settings.Configuration

Install-Package Newtonsoft.Json

Install-Package Microsoft.AspNetCore.Mvc.NewtonsoftJson

Install-Package Microsoft.AspNetCore.Authentication.JwtBearer

Install-Package Microsoft.AspNetCore.Mvc.Versioning

Install-Package AutoMapper

Install-Package AutoMapper.Extensions.Microsoft.DependencyInjection

Install-Package Microsoft.AspNetCore.Authentication.OpenIdConnect

**Discount Services**

**Services -> Discount -> Discount.Core**

Install-Package Microsoft.Extensions.Configuration

Install-Package Npgsql

Install-Package Dapper

**Services -> Discount -> Discount.Api**

Install-Package Microsoft.AspNetCore.Mvc.Versioning

Install-Package Serilog

Install-Package Serilog.Extensions.Logging

Install-Package Serilog.Extensions.Hosting

Install-Package Serilog.Settings.Configuration

Install-Package Newtonsoft.Json

Install-Package Npgsql

Install-Package Dapper

**Services -> Discount -> Discount.Grpc**

Install-Package Serilog

Install-Package Newtonsoft.Json

Install-Package Npgsql

Install-Package Dapper

Install-Package Polly

Install-Package AutoMapper

Install-Package AutoMapper.Extensions.Microsoft.DependencyInjection

Install-Package Grpc.AspNetCore

**Ordering Services**

**Services -> Ordering -> Ordering.Infrastructure**

Install-Package Microsoft.EntityFrameworkCore

Install-Package Microsoft.EntityFrameworkCore.SqlServer

Install-Package Microsoft.EntityFrameworkCore.Tools

Install-Package Microsoft.Extensions.Configuration

Install-Package Microsoft.Extensions.Options

Install-Package Newtonsoft.Json

**Services -> Ordering -> Ordering.Application**

Install-Package AutoMapper

Install-Package AutoMapper.Extensions.Microsoft.DependencyInjection

Install-Package FluentValidation

Install-Package FluentValidation.DependencyInjectionExtensions

Install-Package MediatR

Install-Package MediatR.Extensions.Microsoft.DependencyInjection

Install-Package Microsoft.Extensions.Options

**UI Layer**

**Client -> WebStore**

npm install @material-ui/core --save

npm install @material-ui/lab --save

npm install @material-ui/icons --save

npm install react-flags-select --save

npm install react-icons --save

npm install redux react-redux redux-logger redux-thunk --save

npm install reselect --save

npm install redux-persist --save

npm install redux-saga --save

npm install axios --save

npm install formik --save

npm install yup --save

npm install rxjs --save

npm install react-currency-format --save