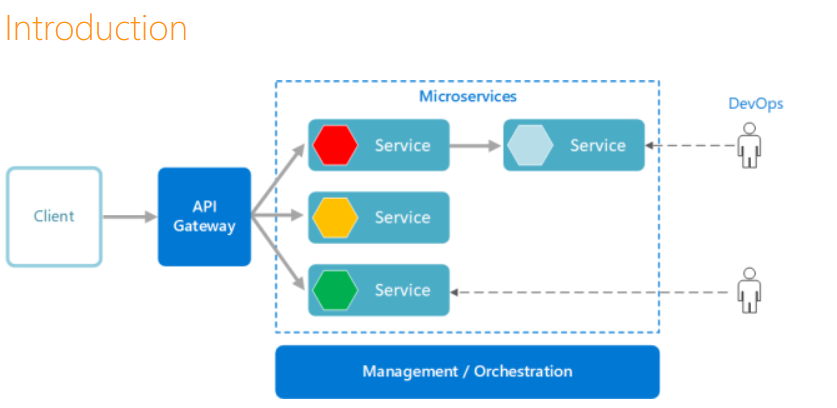
**Micro services using ASP.NET Core, Ocelot, MongoDB and JWT**

**Notes:-**

**Introduction:-**



**1-A micro services architecture consists of a collection of small, independent, and loosely coupled services. Each service is self-contained, implements a single business capability, is responsible for persisting its own data, is a separate codebase, and can be deployed independently.**

**2-API gateways are entry points for clients. Instead of calling services directly, clients call the API gateway, which forwards the call to the appropriate services.API gateways are entry points for clients. Instead of calling services directly, clients call the API gateway, which forwards the call to the appropriate services.**

**3-There are multiple advantages using micro services architecture:**

**Developers can better understand the functionality of a service.**

**Failure in one service does not impact other services.**

**It's easier to manage bug fixes and feature releases.**

**Services can be deployed in multiple servers to enhance performance.**

**Services are easy to change and test.**

**Services are easy and fast to deploy.**

**Allows to choose technology that is suited for a particular functionality.**

**4-Before choosing microservices architecture, here are some challenges to consider:**

**Services are simple but the entire system as a whole is more complex.**

**Communication between services can be complex.**

**More services equals more resources.**

**Global testing can be difficult.**

**Debugging can be harder.**

**Development Environment**

**Visual Studio 2019**

**.NET Core 3.1**

**MongoDB**

**Postman**

**Technologies**

**C#**

**ASP.NET Core**

**Ocelot**

**Swashbuckle**

**Serilog**

**JWT**

**MongoDB**

**xUnit**

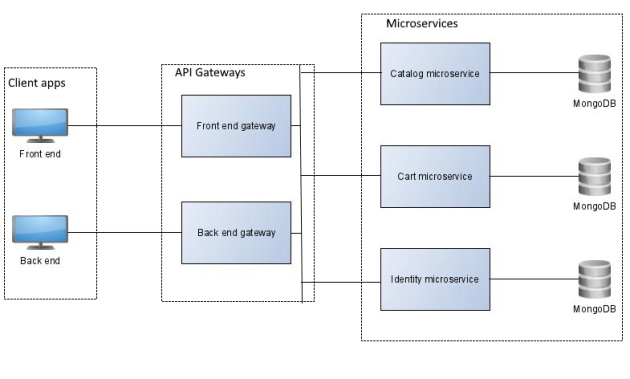
**Moq**

**HTML**

**CSS**

**JavaScript**

**Architecture**



**There are three micro services:**

**Catalog micro service: allows to manage the catalog.**

**Cart micro service: allows to manage the cart.**

**Identity micro service: allows to manage users.**

**Each micro service implements a single business capability and has its own MongoDB database.**

**There are two API gateways, one for the front end and one for the back end.**

**Below is the front end API gateway:**

**GET /catalog: retrieves catalog items.**

**GET /catalog/{id}: retrieves a catalog item.**

**GET /cart: retrieves cart items.**

**POST /cart: adds a cart item.**

**PUT /cart: updates a cart item.**

**DELETE /cart: deletes a cart item.**

**POST /identity/login: performs a login.**

**POST /identity/register: registers a user.**

**GET /identity/validate: validates a JWT token.**

**Below is the back end API gateway:**

**GET /catalog: retrieves catalog items.**

**GET /catalog/{id}: retrieves a catalog item.**

**POST /catalog: creates a catalog item.**

**PUT /catalog: updates a catalog item.**

**DELETE /catalog/{id}: deletes a catalog item.**

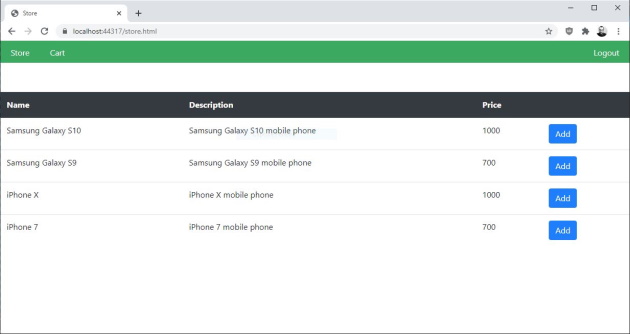
**POST /identity/login: performs a login.**

**GET /identity/validate: validates a JWT token.**

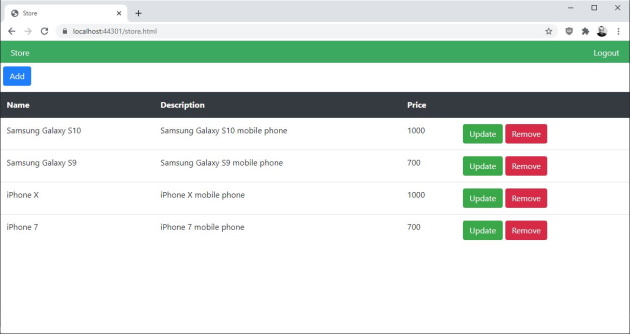
**Finally, there are two client apps. A front end for accessing the store and a back end for managing the store.**

**The front end allows registered users to see the available catalog items, allows to add catalog items to the cart, and allows to remove catalog items from the cart.**

**Here is a screenshot of the store page in the front end**



**The back end allows admin users to see the available catalog items, allows to add new catalog items, allows to update catalog items, and allows to remove catalog items.**



**Microservice Projects:-**

**CatalogMicroservice project contains the source code of the microservice managing the catalog.**

**CartMicroservice project contains the source code of the microservice managing the cart.**

**IdentityMicroservice project contains the source code of the microservice managing users.**

**Middleware project contains the source code of common functionalities used by microservices.**

**FrontendGateway project contains the source code of the front end API gateway.**

**BackendGateway project contains the source code of the back end API gateway.**

**Frontend project contains the source code of the front end client app.**

**Backend project contains the source code of the back end client app.**

**test solution folder contains unit tests of all microservices.**