# Општи информации

Проектот е направен во Visual Studio 2022 .Net C# технологија со MS SQL база на податоци.

Се состои од 2 дела Front End ( Blazor Server Application) и Back End (ASP .Net Core Web API) - Framework .Net 6.0.

Во самиот проект е прикажано и видео (Video\_ClientsApp.gif ) кое е поставено на GitHub со едно сценарио како е направено и замислено решението.

Моделот на базата се состои од 3 табели прикажани на сликата: шифрарник со Клиенти, шифрарник со Адреси и табела ClientsAddress каде што се запишуваат бројт на клиентот (IdClient) и типот на адресата (IdAddress) како foreign key и самиот назив на адресата во текстуално поле AddressData.

A screenshot of a computer

Description automatically generated

Целосна скрипта од сите табели и процедури кои се потребни за да функционира решението се наоѓа во фајлот DatabaseScript.sql кој е поставен на GitHub.

За секој Client се чуваат општи податоци ( број на клиент, име и датум на раѓање). Еден клиент може да има повеќе адреси типизирани во табелата Address, но истиот тој клиент за еден тип на адреса може да има само еден податок, пример е прикажан на следнава слика

A screenshot of a computer

Description automatically generated

# Архитектура на решението

Архитектурата на системот се состои од следниве делови прикажани на сликата

A screenshot of a computer program

Description automatically generated

**Clients.Server - Клиентска страна** FrontEnd– овде се наоѓаат сите .razor Pages каде што е прикажан UI на корисникот. Направено е во Blazor технологија. Во истиот тој Solution се и Services коишто служат за повик на API EndPoint.

Во фајлот appsettings.json е дефинирана конекцијата со API "BaseAPIUrl": <https://localhost:7200>

Во фолдерот Services се наоѓаат сервисите кои служат за повик на Client.API делот (во Controllers)

**Clients.API Серверска страна** BackEnd – овде се креирани API EndPoints, постојат 2 типа на requests (Get и Post) и истите може да се тестираат преку swagger.

**Class Library** – додадени се 2 такви солуции (DataAcess i Models) што всушност претставува комуникација со база и дефинирање на моделите кои се користат.

# Имплементација

1. Приказ на сите клиенти - дефинирана е прецодура [dbo].[p\_ClientsGet] која ги враќа сите клиенти и истите се прикажани во табела која може да се сортира по секоја колона и податоците може да се експортираат во json.
2. Upload на xml фајл – овде целокупниот фајл како влезен параметар со сите податоци се користи во процедурата за внес [dbo].[p\_ClientsInsertFromXML]. Внатре во самата процедура постои логика за тоа кој податок во која табела треба да се запише.
3. Запис на еден клиент –поединечен внес на клиент овде е направен само API делот а имплементацијата на клиентската страна со валидација на податоците моментално не е имплементирано.

A screenshot of a computer

Description automatically generated