**Documentatie proiect**

Proiectul meu si-a dorit a fi site-ul web al unui producator de miere naturala.

Am inceput prin a-mi alege cum sa imi fac baza de date. Pentru acest lucru am ales sa folosesc Azure, deoarece am gandit ca imi este mai usor sa imi creez baza de date in cloud. Mi-am creeat baza de date care are numele “Laura” si server name-ul MIERE.database.windows.net.

A screenshot of a computer

Description automatically generated

In sectiunea de querry am adaugat querry uri pentru creearea tabelelor si inserarea inregistrarilo. Querry-urile sunt de forma aceasta:

A white screen with black text

Description automatically generated

A white text on a white background

Description automatically generated

Putem vedea structura bazei de date in imaginea de mai jos, si de asemenea si tabele create:

A screenshot of a computer

Description automatically generated

Am trecut apoi la lucrul propriu zis din Visual studio si am inceput prin a-mi crea Modele si Controllere pentru fiecare table in parte, de ex:

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Controllere si modelele au rol essential pentru functionalitatea si organizarea apliccatiei. Modleel defines structura bazei de date si sunt folosite la maparea datelor intre baza de date si aplicatie. Modelel mai specifica de asemenea si relatiile dintre tabele si valideaza datele. Controllerele sunt puncta de intrare pentru cererile HTTP catre aplicatia mea. Ele gestioneaza operatii CRUD(create, delete read, update) pentru datele din modele. Ex controller:

namespace AlbinutaDulce.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class ProductsApiController : ControllerBase

{

private readonly AppDbContext \_context;

public ProductsApiController(AppDbContext context)

{

\_context = context;

}

// GET: api/ProductsApi

[HttpGet]

public IActionResult GetProducts()

{

var products = \_context.Products.ToList();

return Ok(products); // Returnează datele în format JSON

}

// GET: api/ProductsApi/{id}

[HttpGet("{id}")]

public IActionResult GetProduct(int id)

{

var product = \_context.Products.FirstOrDefault(p => p.ProductId == id);

if (product == null)

{

return NotFound();

}

return Ok(product);

}

// POST: api/ProductsApi

[HttpPost]

public IActionResult CreateProduct([FromBody] Product product)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

\_context.Products.Add(product);

\_context.SaveChanges();

return CreatedAtAction(nameof(GetProduct), new { id = product.ProductId }, product);

}

// PUT: api/ProductsApi/{id}

[HttpPut("{id}")]

public IActionResult UpdateProduct(int id, [FromBody] Product product)

{

if (id != product.ProductId || !ModelState.IsValid)

{

return BadRequest();

}

var existingProduct = \_context.Products.FirstOrDefault(p => p.ProductId == id);

if (existingProduct == null)

{

return NotFound();

}

existingProduct.Name = product.Name;

existingProduct.Description = product.Description;

existingProduct.Price = product.Price;

existingProduct.Stock = product.Stock;

\_context.SaveChanges();

return NoContent();

}

// DELETE: api/ProductsApi/{id}

[HttpDelete("{id}")]

public IActionResult DeleteProduct(int id)

{

var product = \_context.Products.FirstOrDefault(p => p.ProductId == id);

if (product == null)

{

return NotFound();

}

\_context.Products.Remove(product);

\_context.SaveChanges();

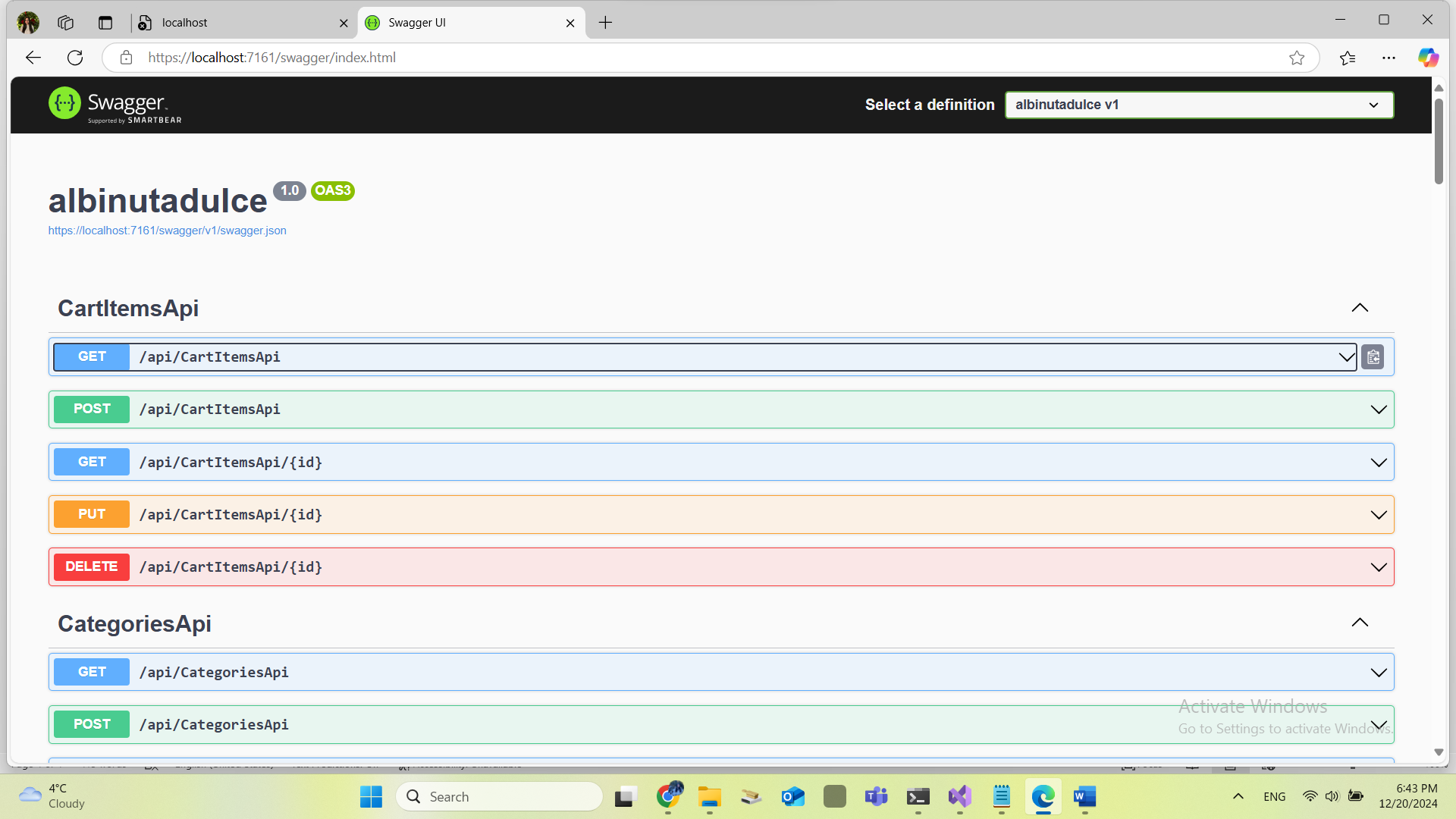
return NoContent();

}

}

}

Astefl am reusit sa construiesc doar backendul aplicatiei mele cu swagger. Swagger este un set de specificatii si instrumente standard pentru documentarea si testarea API-urilor RESTful. Mai jos sunt prezentate functionalitatile Get, Post, Delete, Update.



Se selecteeaza un table si se apara pe try it out, apoi execute iar api ul returneaza ca response priodusele din baza mea de date.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Se pot incerca si testa si celelalte variante. Conexiunea cu baza mea de date s-a realizat in AppSettings.json:

{

"ConnectionStrings": {

"DefaultConnection": "Server=MIERE.database.windows.net,1433;Initial Catalog=Laura;Persist Security Info=False;User ID=citul;Password=La13anI@;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;"

},

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*"

}