Pastikan Anda menginstal paket yang diperlukan seperti Microsoft.EntityFrameworkCore.SqlServer dan Microsoft.EntityFrameworkCore.Tools.

Buka terminal dan jalankan perintah berikut untuk menginstal paket yang diperlukan:

dotnet add package Microsoft.EntityFrameworkCore.SqlServer

dotnet add package Microsoft.EntityFrameworkCore.Tools

2.2. Buat Model dan DbContext

Buat model untuk produk dan DbContext untuk menghubungkan aplikasi ke database.

Product.cs (Model):

```
public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public decimal Price { get; set; }
    public int Quantity { get; set; }
}
```

ApplicationDbContext.cs (DbContext):

```
using Microsoft.EntityFrameworkCore;

public class ApplicationDbContext : DbContext
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) :
base(options) { }

    // DbSet untuk produk
    public DbSet<Product> Products { get; set; }
}
```

2.3. Konfigurasi Koneksi Database di Program.cs

Tambahkan koneksi ke SQL Server di Program.cs.

```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

builder.Services.AddControllers();
builder.Services.AddCors(options =>
{
    options.AddPolicy("AllowAngularApp", policy =>
        policy.WithOrigins("http://localhost:4200")  // URL Angular Anda
              .AllowAnyMethod()
              .AllowAnyHeader());
});

var app = builder.Build();

// Enable CORS
app.UseCors("AllowAngularApp");

app.MapControllers();

app.Run();
```

2.4. Tambahkan Connection String di appsettings.json

Tambahkan connection string di appsettings.json untuk menghubungkan aplikasi dengan SQL Server.

```
{
  "ConnectionStrings": {
```

```
  "DefaultConnection":
"Server=localhost;Database=InventoryDB;Trusted_Connection=True;"
 }
}
```

2.5. Buat Controller untuk Mengakses Stored Procedure

Buat controller untuk mengakses **Stored Procedure** dan menampilkan data produk.

ProductsController.cs:

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System.Data.SqlClient;

[Route("api/[controller]")]
[ApiController]
public class ProductsController : ControllerBase
{
  private readonly ApplicationDbContext _context;

  public ProductsController(ApplicationDbContext context)
  {
    _context = context;
  }

  // GET: api/products
  [HttpGet]
  public async Task<IActionResult> GetAllProducts()
  {
    var products = await _context.Products.FromSqlRaw("EXEC
GetAllProducts").ToListAsync();
    return Ok(products);
  }
}
```

3. Buat Frontend dengan Angular 18

3.1. Buat Proyek Angular

Jika Anda belum memiliki proyek Angular, buatlah proyek baru dengan perintah berikut:

```
ng new angular-app
cd angular-app
ng serve
```

3.2. Buat Service di Angular untuk Mengakses API

Di dalam proyek Angular, buat service untuk berkomunikasi dengan API ASP.NET Core.

api.service.ts:

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class ApiService {
  private apiUrl = 'http://localhost:5000/api/products'; // URL API ASP.NET Core

  constructor(private http: HttpClient) { }

  getProducts(): Observable<any> {
    return this.http.get<any>(this.apiUrl);
  }
}
```

3.3. Tampilkan Data di Komponen Angular

Di dalam komponen Angular, panggil service untuk mengambil data dari API dan tampilkan data produk.

app.component.ts:

```
import { Component, OnInit } from '@angular/core';
import { ApiService } from './api.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  products: any[] = [];

  constructor(private apiService: ApiService) {}

  ngOnInit() {
    this.apiService.getProducts().subscribe(data => {
      this.products = data;
    });
  }
}
```

app.component.html:

```
<div>
  <h1>List of Products</h1>
  <ul>
    <li *ngFor="let product of products">
      {{ product.name }} - ${{ product.price }} (Stock: {{ product.quantity }})
    </li>
  </ul>
</div>
```

3.4. Pastikan HttpClientModule Terinstal

Pastikan Anda mengimpor HttpClientModule di app.module.ts untuk mendukung HTTP requests.

app.module.ts:

```
import { HttpClientModule } from '@angular/common/http';


@NgModule({
  declarations: [AppComponent],
  imports: [HttpClientModule],
  bootstrap: [AppComponent]
})
export class AppModule {}
```