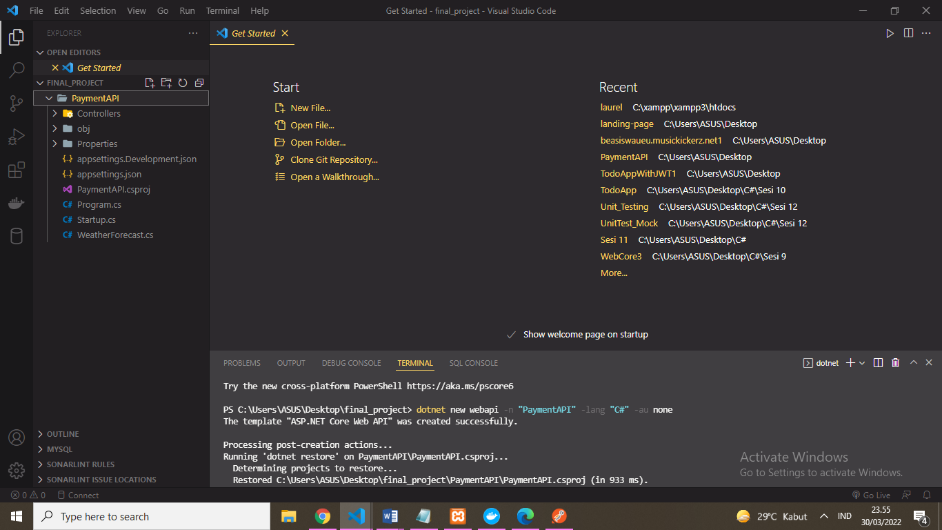
Nama : Hafizah Safira Kaurani

No. Peserta : FSDO003ONL001

**Panduan Pembuatan Final Project (Payment API)**

A. Pembuatan Project API

1. Buat folder baru
2. Buka visual studio code, kemudian buka folder yang sudah dibuat
3. Inisiasi project dotnet di dalam folder tersebut :

>> dotnet new webapi -n "PaymentAPI" -lang "C#" -au none

1. Install package. Beberapa package yang dibutuhkan yaitu :

>> dotnet add package NuGet.CommandLine.XPlat --version 5.11.0

>> dotnet add package Microsoft.EntityFrameworkCore.Tools --version 5.0.11

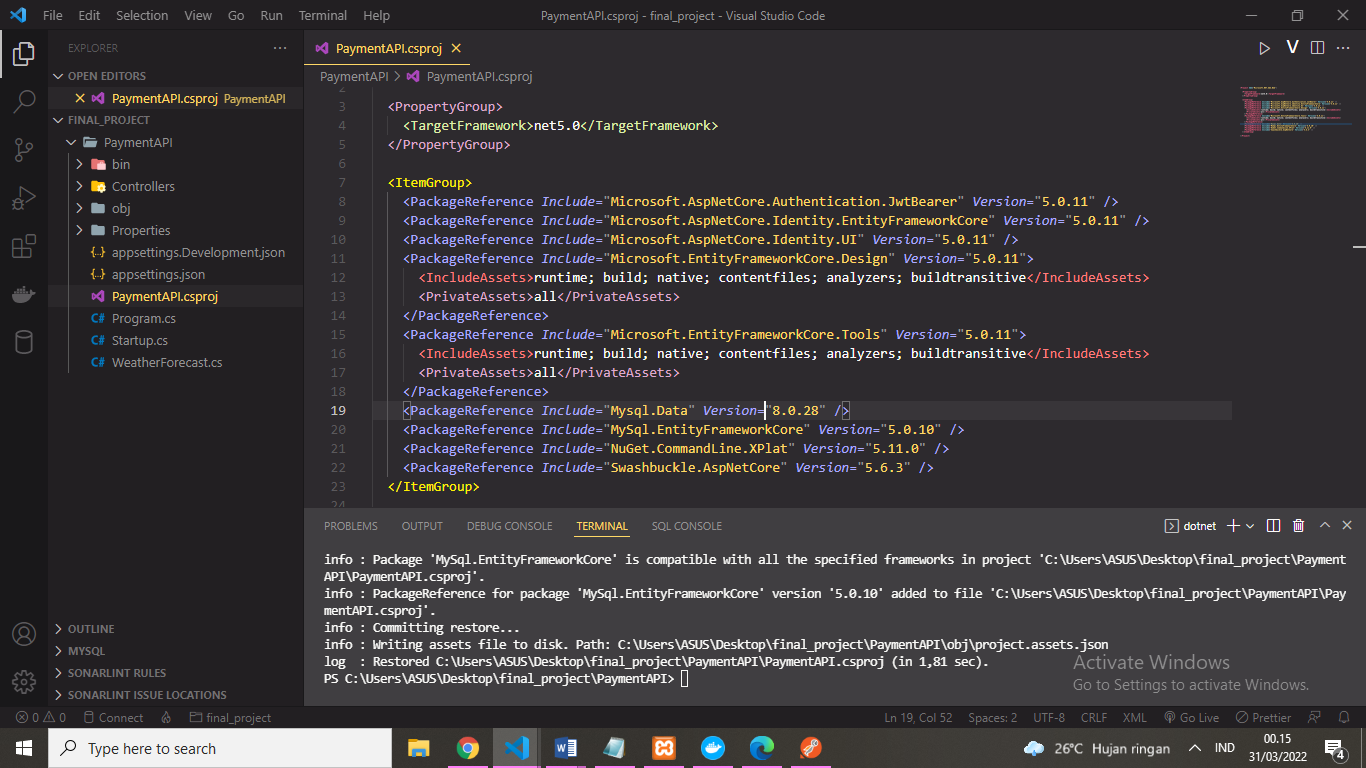
>> dotnet add package Microsoft.EntityFrameworkCore.Design --version 5.0.11

>> dotnet add package Microsoft.AspNetCore.Authentication.JwtBearer --version 5.0.11

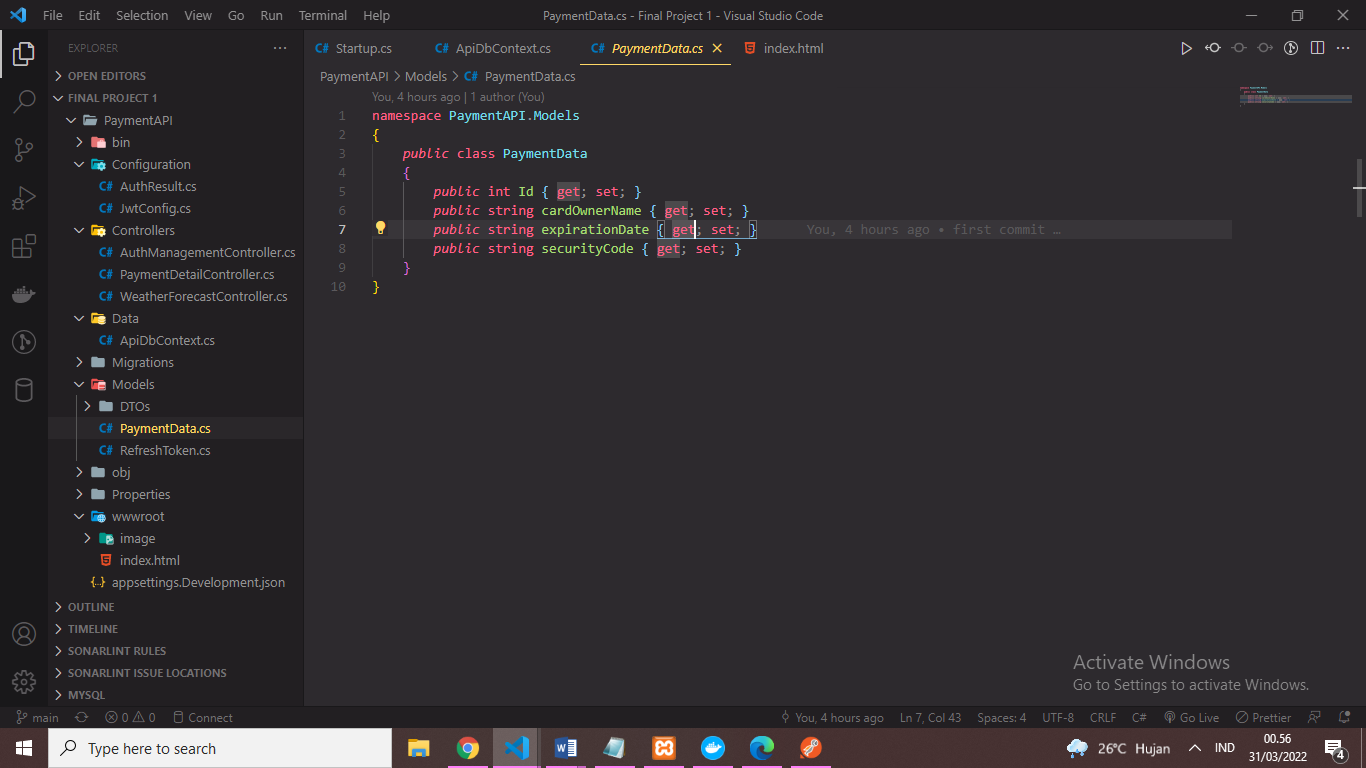
>> dotnet add package Microsoft.AspNetCore.Identity.EntityFrameworkCore --version 5.0.11

>> dotnet add package Microsoft.AspNetCore.Identity.UI --version 5.0.11

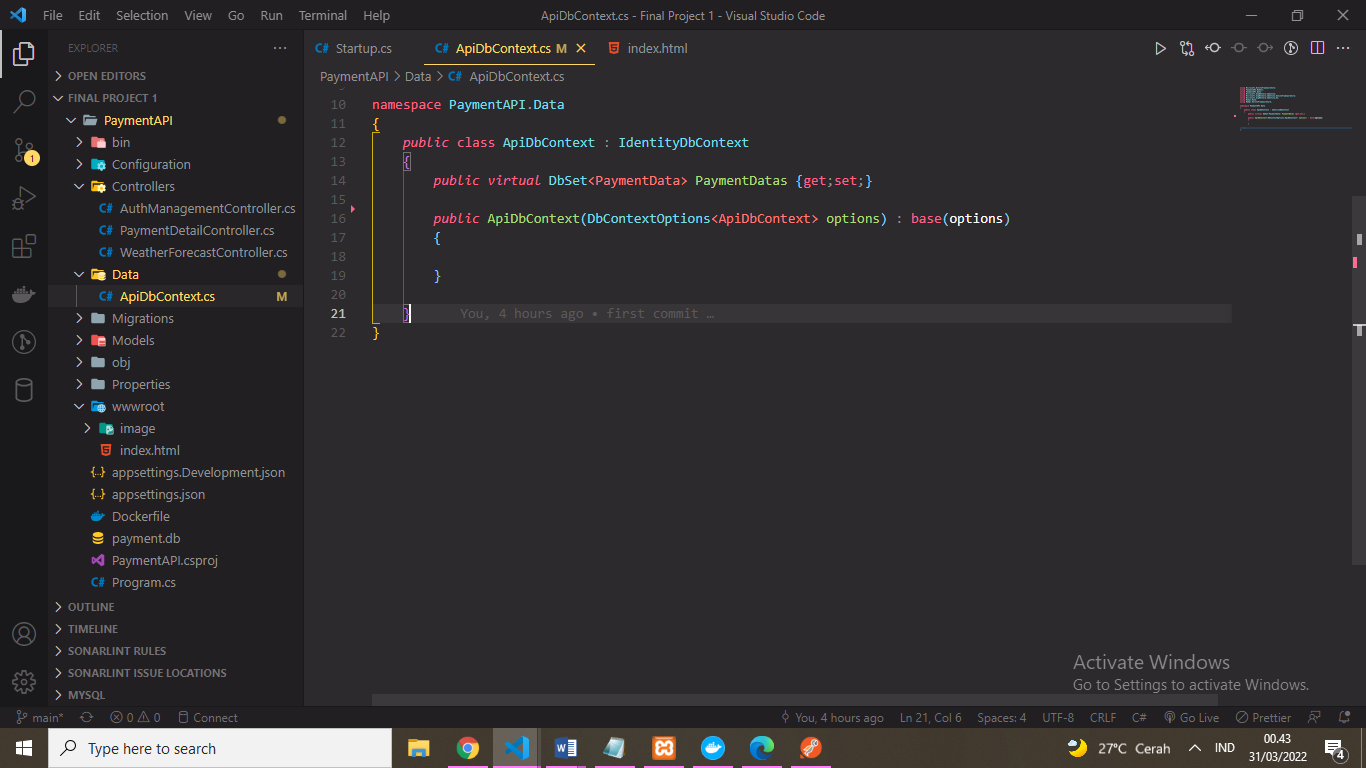
>> dotnet add package Microsoft.Mysql.Data --version 8.0.28

>> dotnet add package Microsoft.Mysql.EntityFrameworkCore --version 5.0.10

1. Buat folder Models, lalu tambahkan file PaymentData.cs di dalam folder tersebut

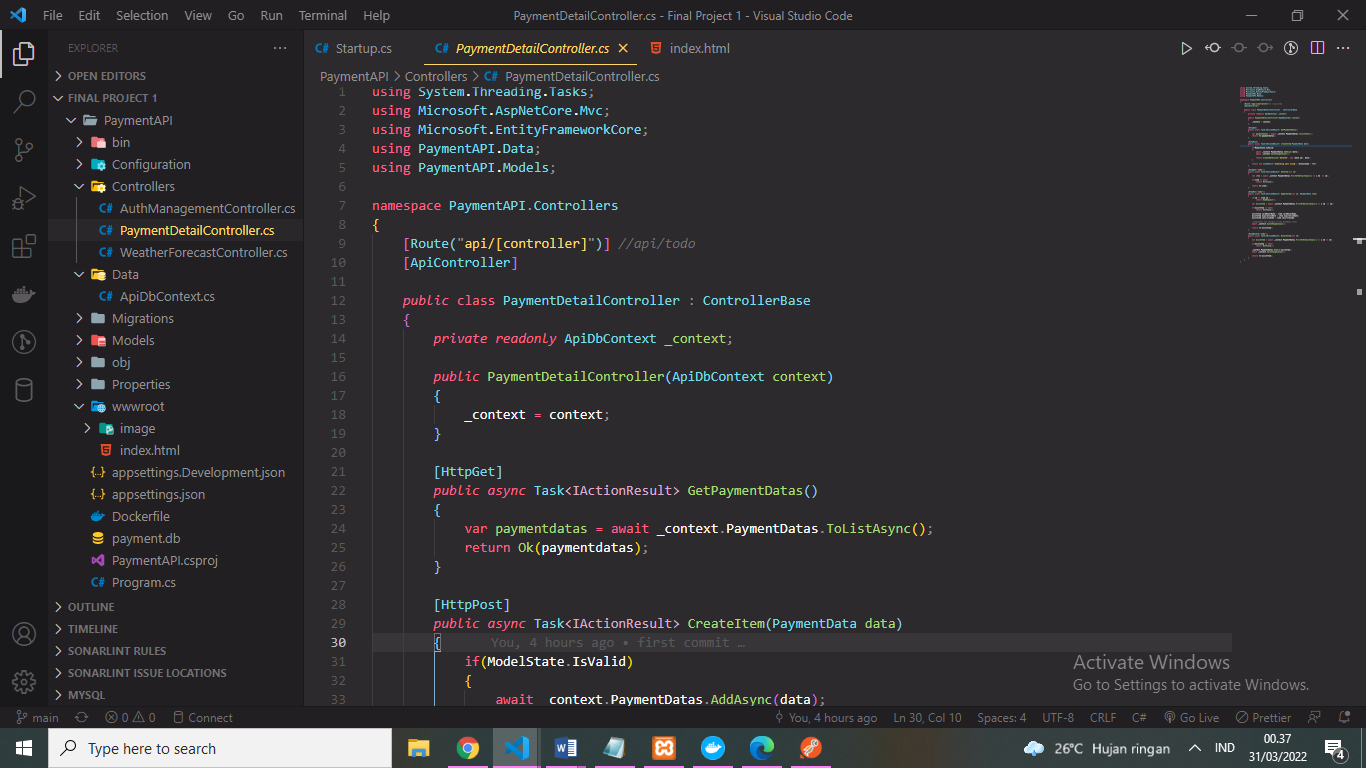
*Location : PaymentAPI/Models/PaymentData.cs*

1. Buat folder Data, lalu tambahkan file ApiDBContext

*Location : PaymentAPI/Data/ApiDbContext.cs*

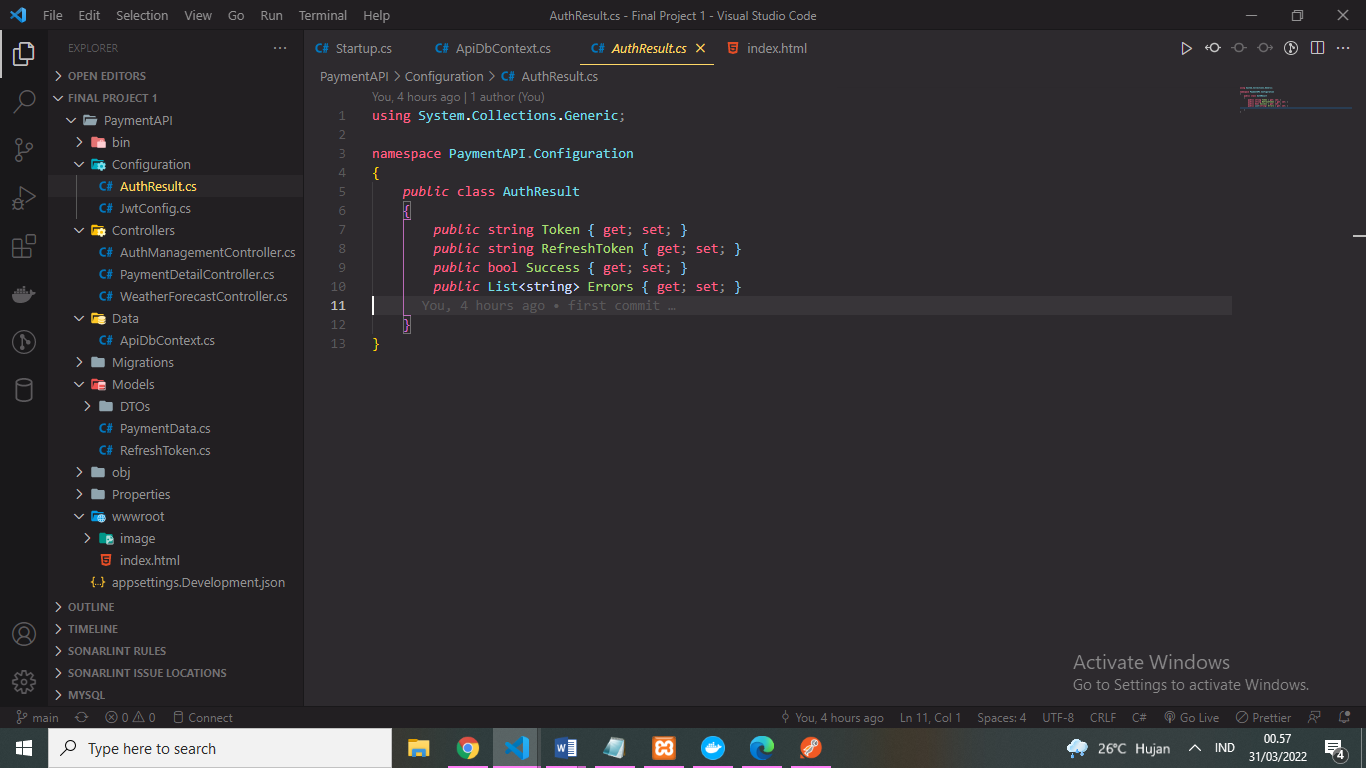
1. Buat folder Controllers, lalu lanjutkan build PaymentDetailController untuk method post, get, put, delete

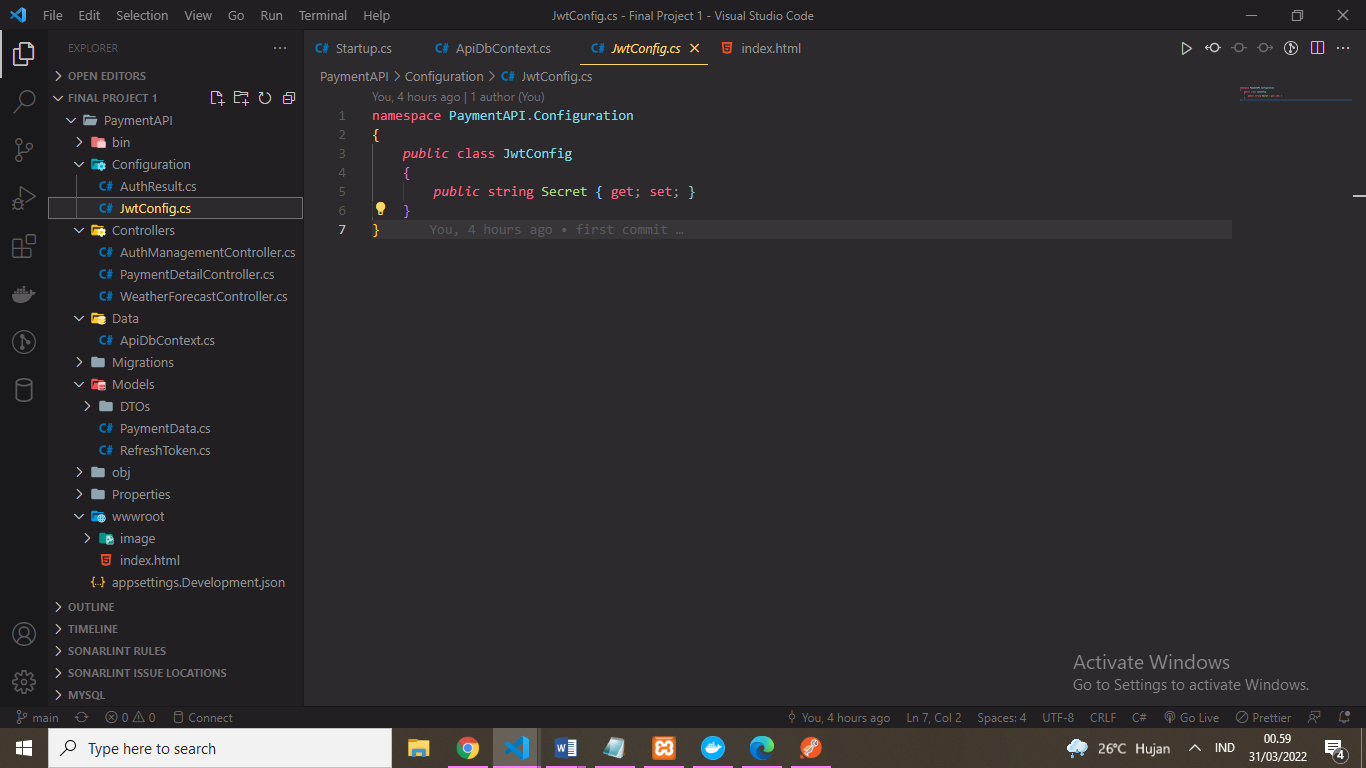
*Location : PaymentAPI/Controllers/PaymentDetailController.cs*



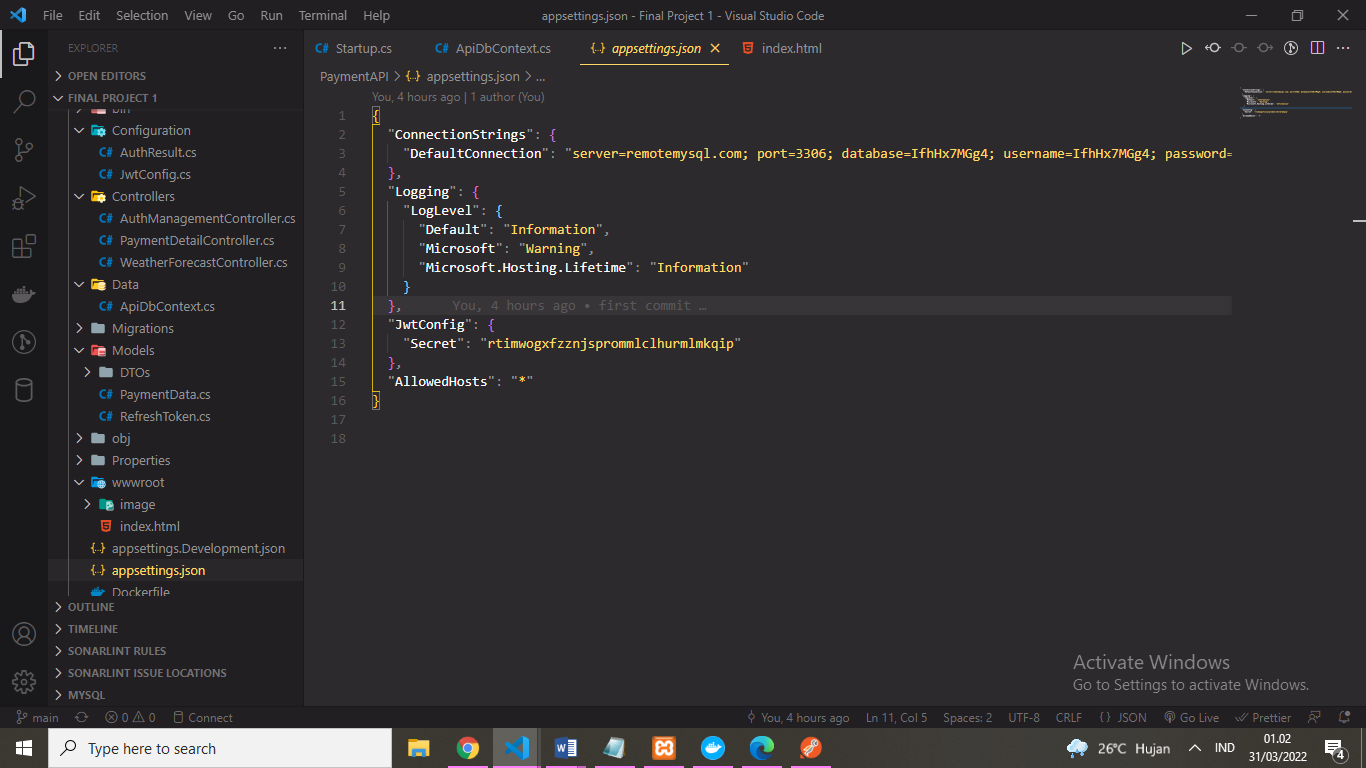
B. Penambahan JWT Token dan Refresh Token

1. Buat folder Configuration, lalu tambahkan file AuthResult.cs dan JwtConfig.cs:

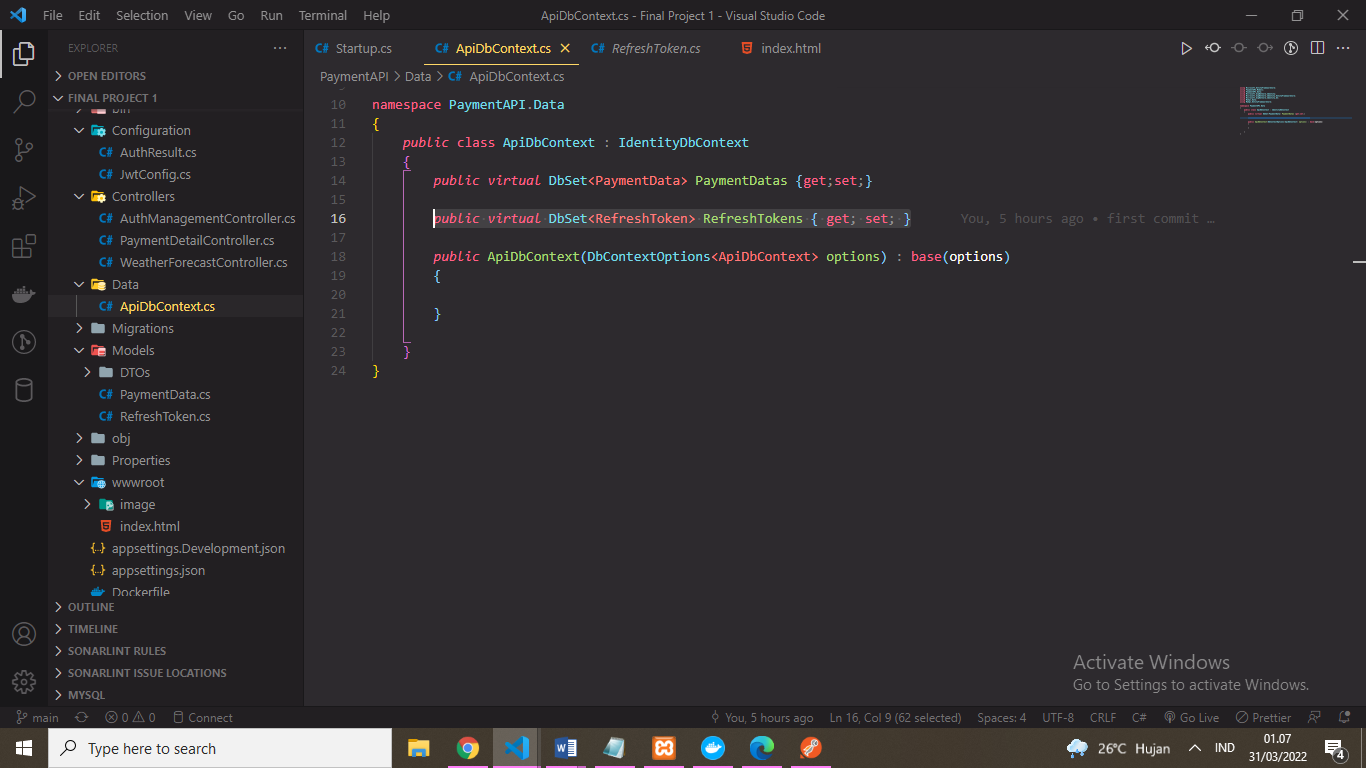
*Location : PaymentAPI/Configuration/AuthResult.cs*

*Location : PaymentAPI/Configuration/JwtConfig.cs*

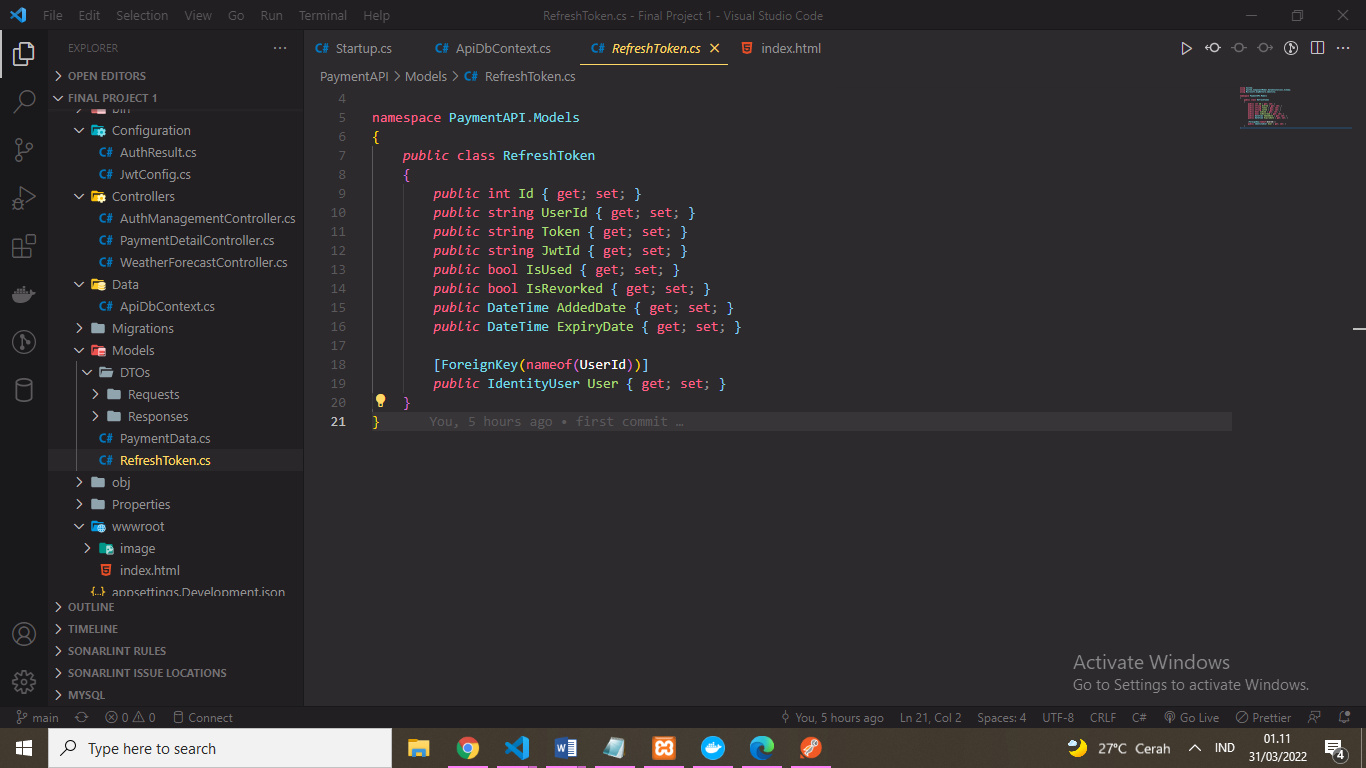
1. Pada file appsettings.json tambahkan JwtConfig.Secrets

*Location : PaymentAPI/appsettings.json*

1. Tambahkan DbSet untuk RefreshToken pada file ApiDbContext.cs

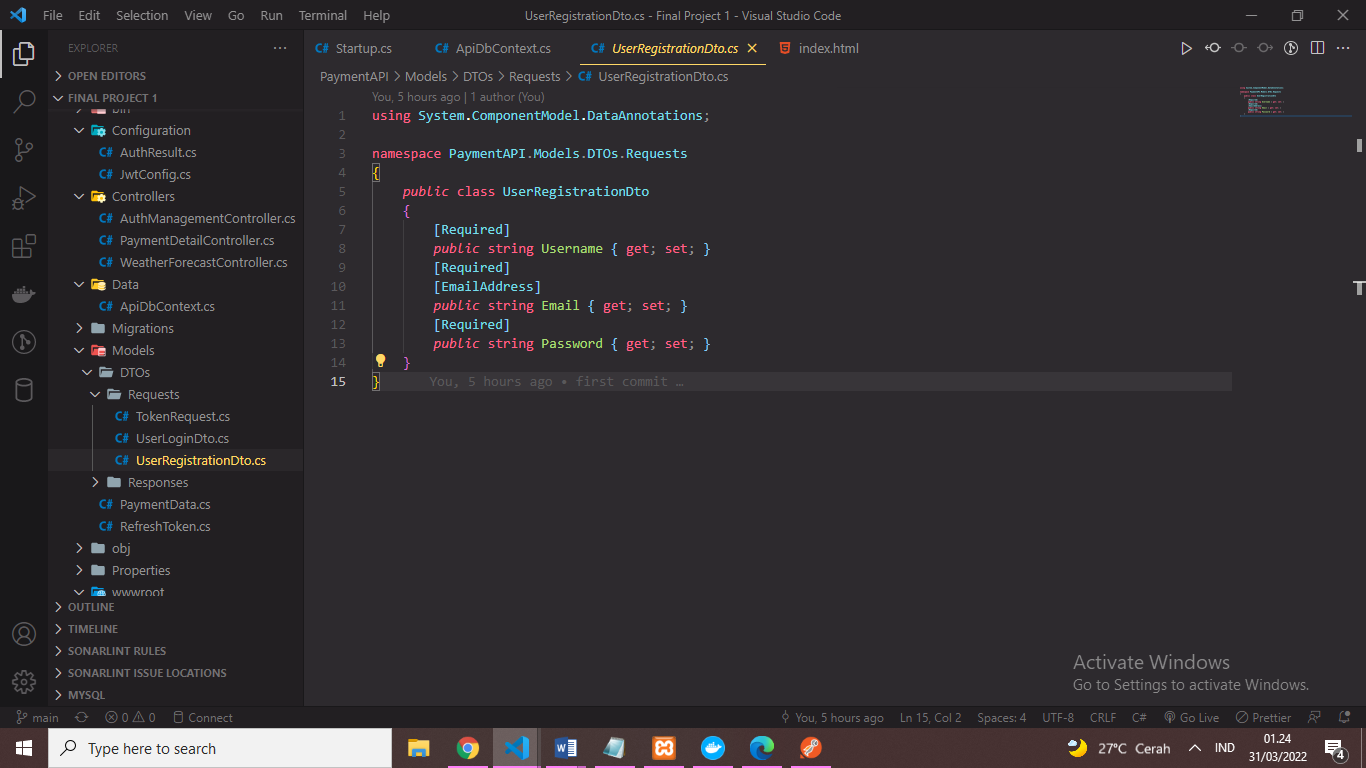
*Location : PaymentAPI/Data/ApiDbContext.cs*

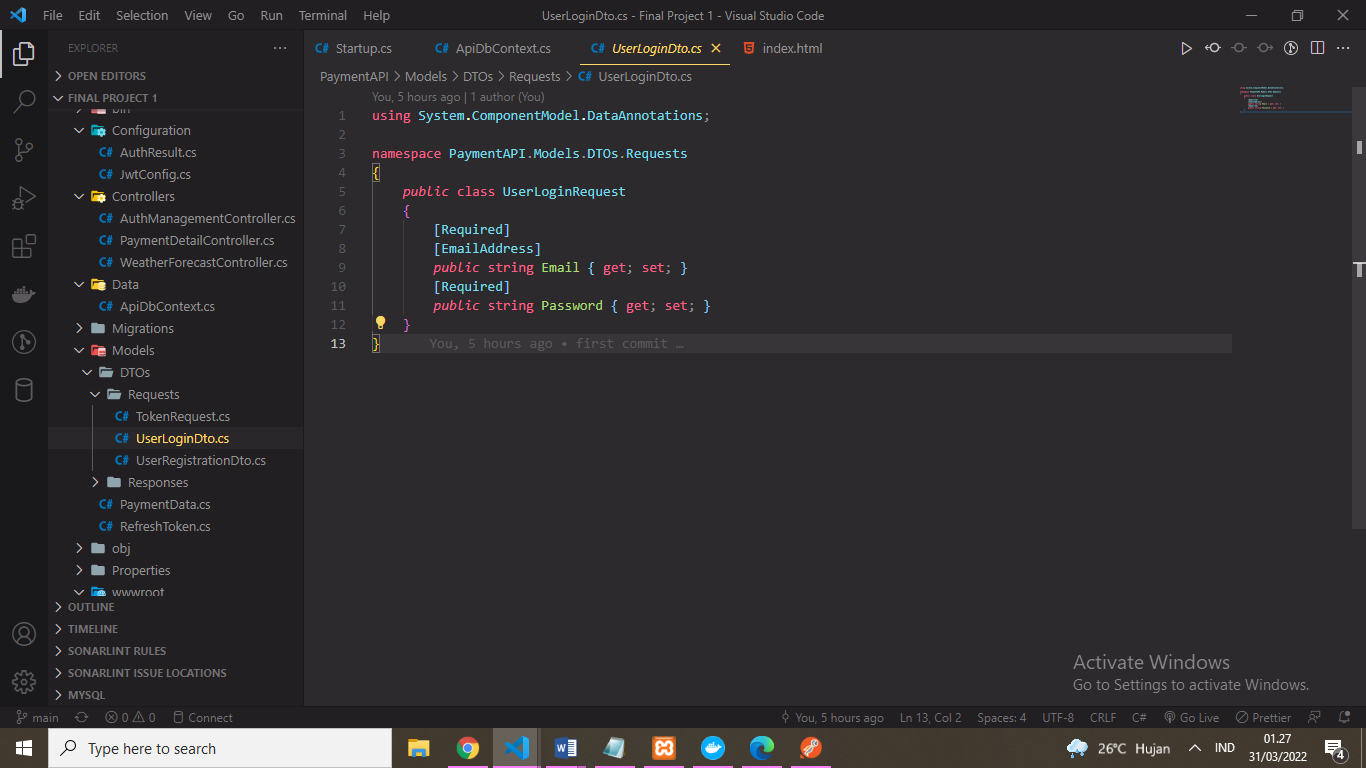
1. Buat file baru RefreshToken.cs yang terdapat pada folder Models

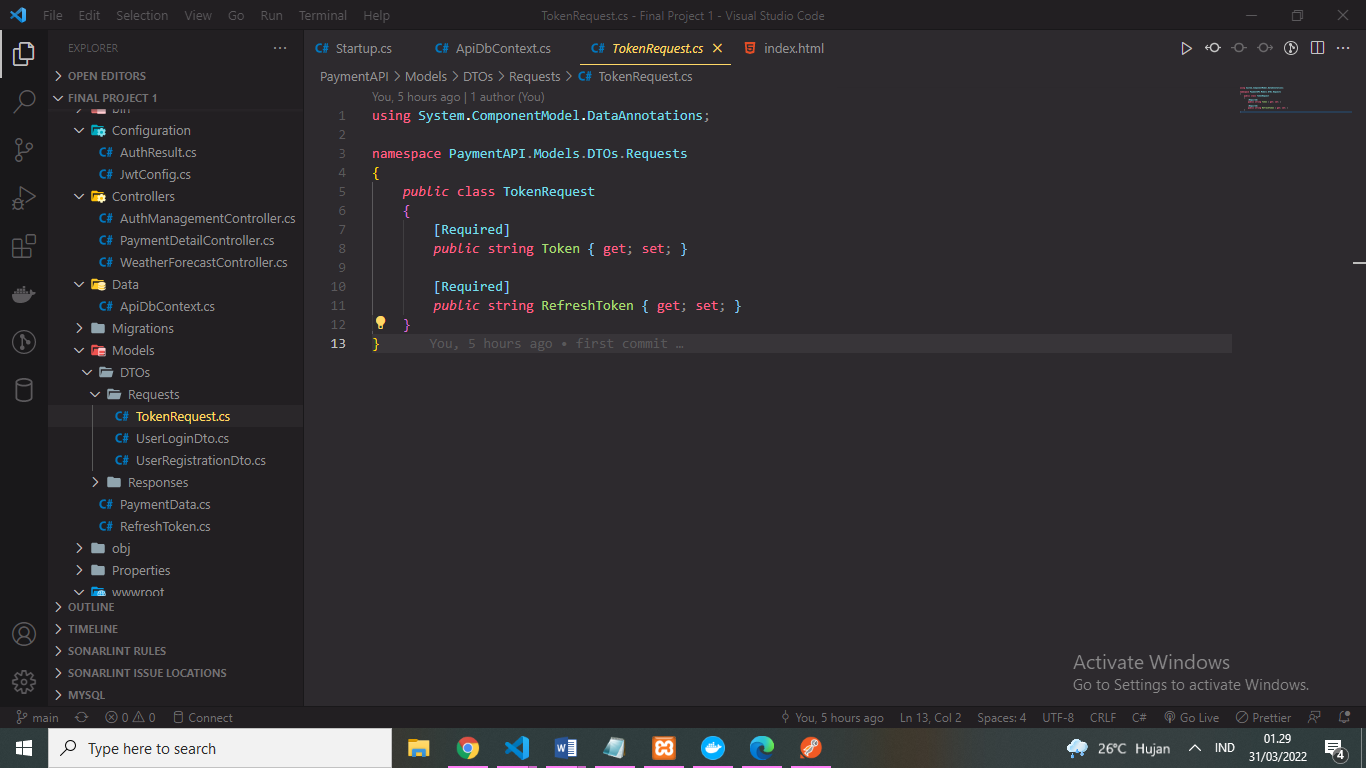
*Location : PaymentAPI/Models/RefreshToken.cs*

1. Buat folder baru bernama DTOs yang di dalamnya terdapat juga folder Request & Response

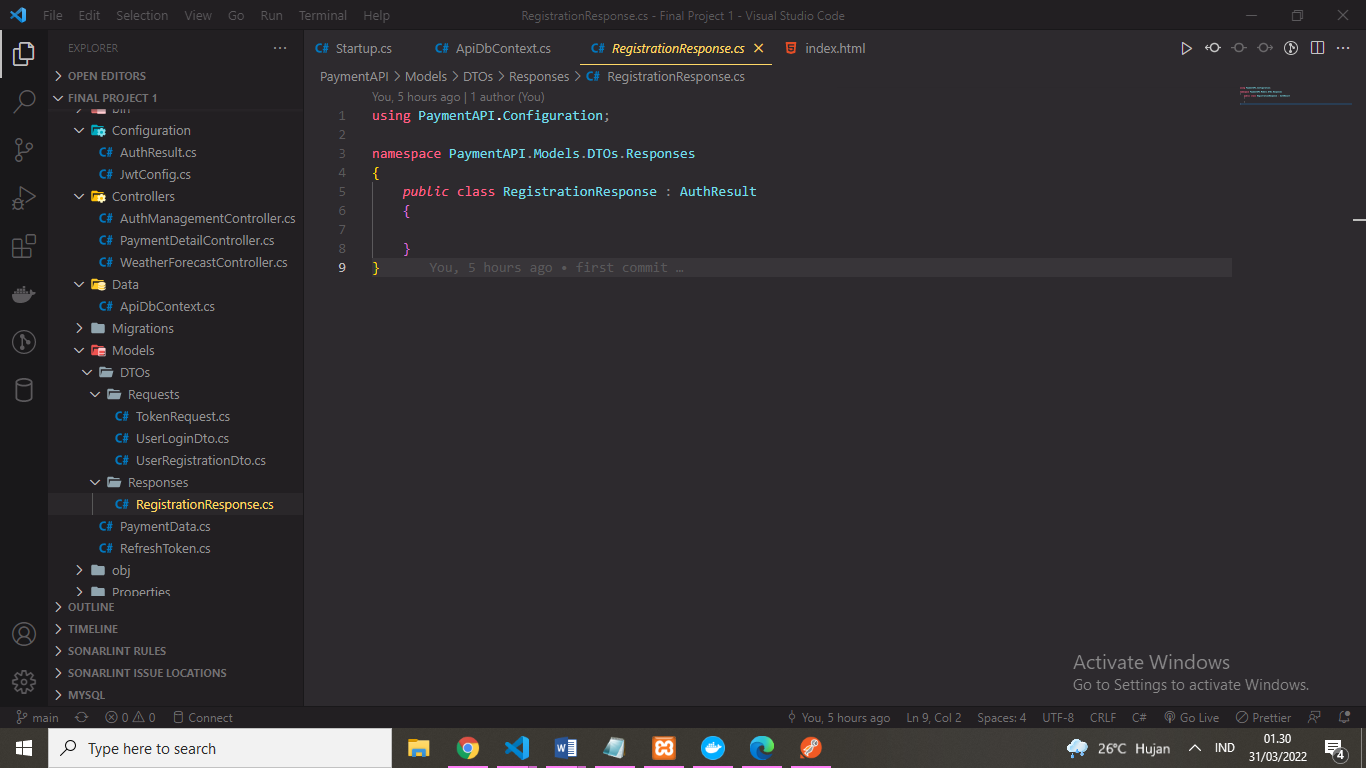
* **Folder Request**

*Location : PaymentAPI/Models/DTOs/Requests/UserRegistrationDto.cs*

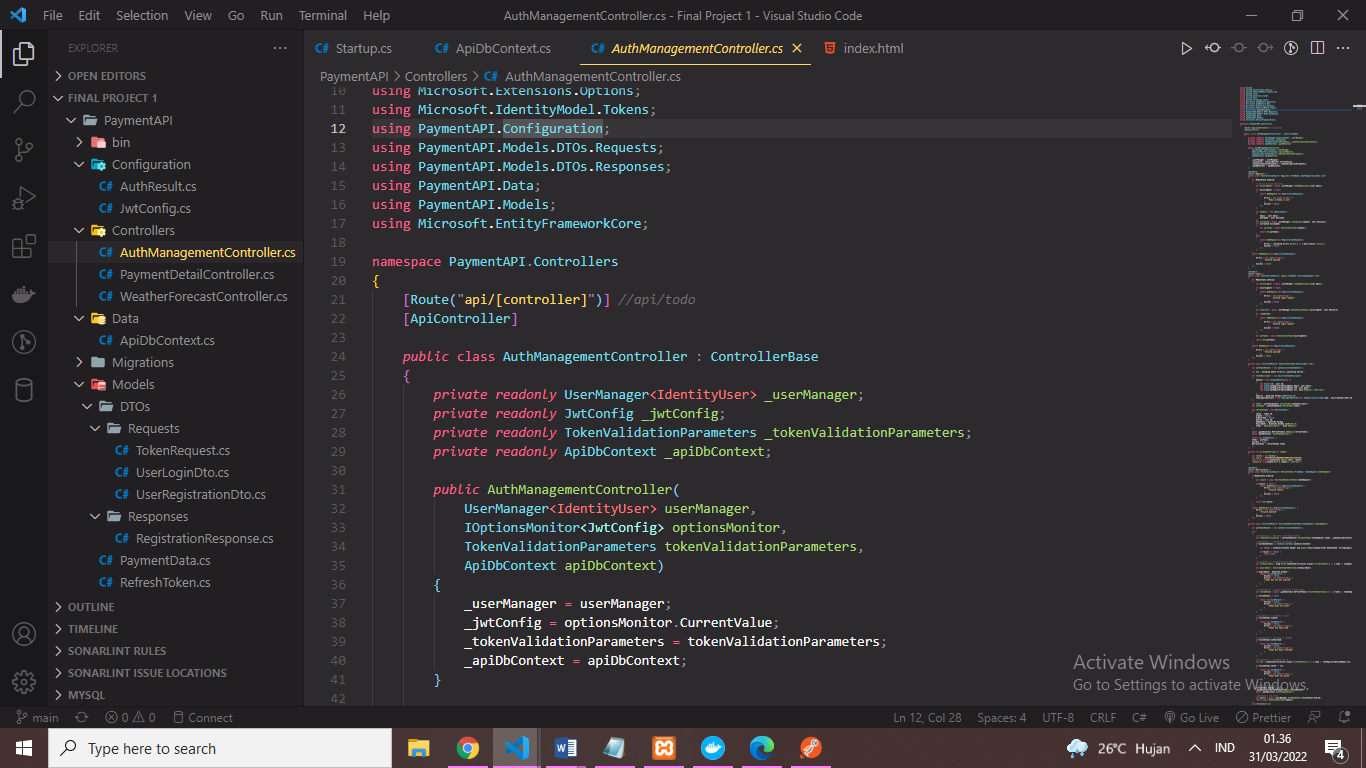
*Location : PaymentAPI/Models/DTOs/Requests/UserLoginDTOs.cs*

*Location : PaymentAPI/Models/DTOs/Requests/TokenRequest.cs*

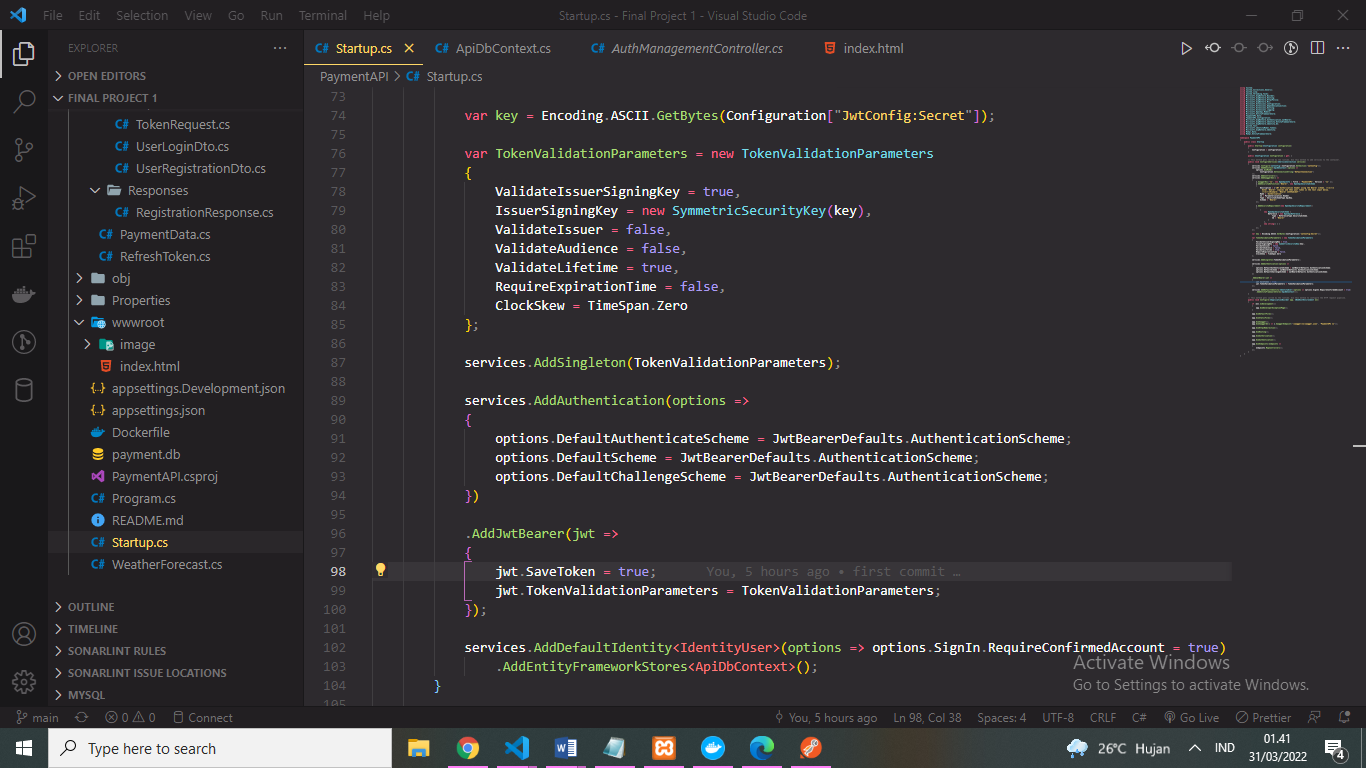
* **Folder Response**

*Location : PaymentAPI/Models/DTOs/Response/RegistrationResponse.cs*

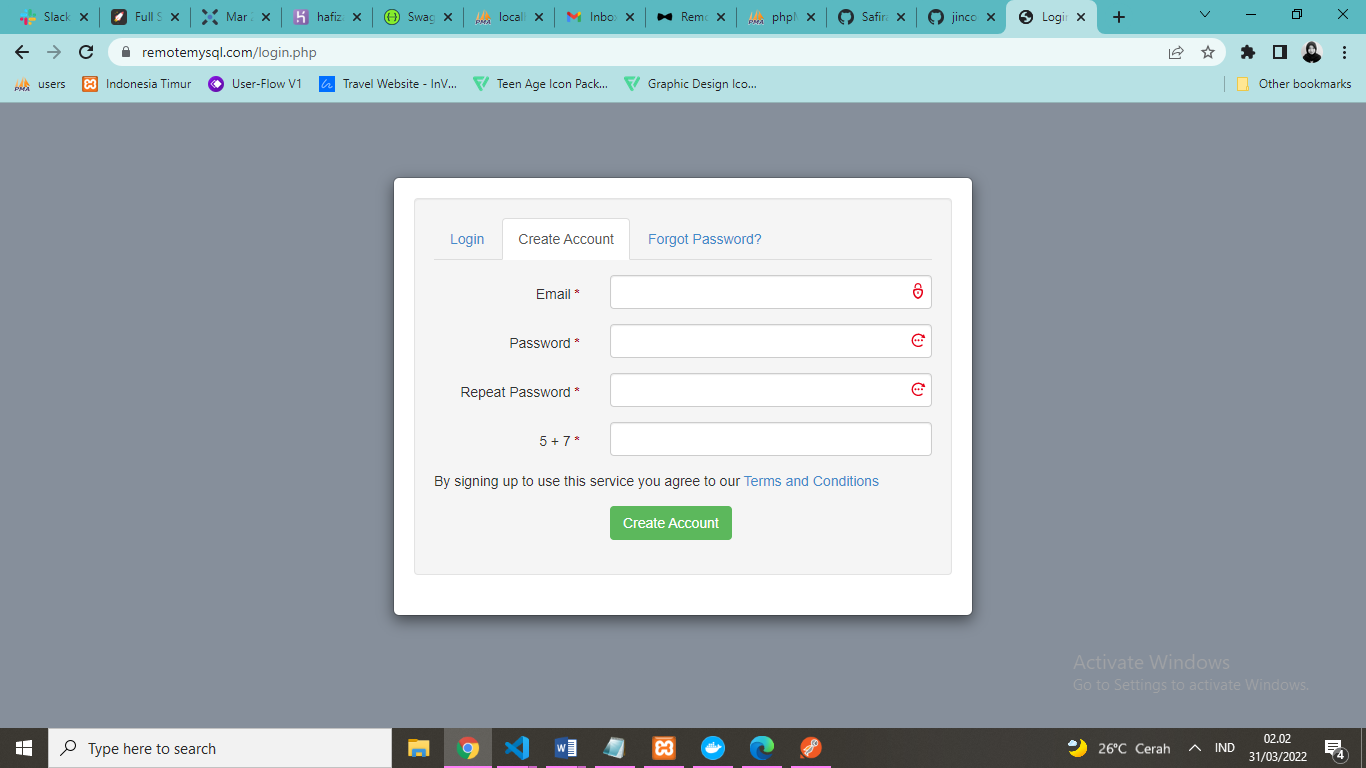
1. Tambahkan file AuthManagement pada folder Controllers

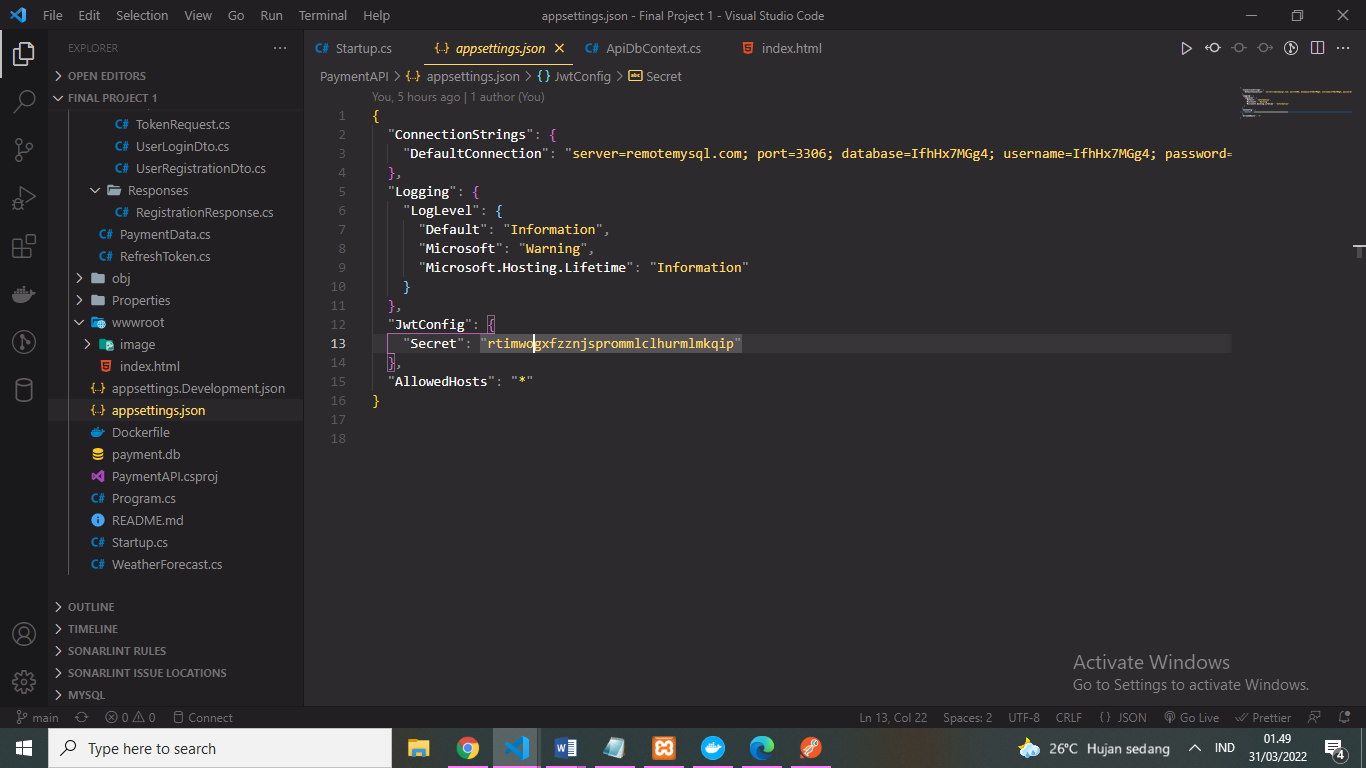
*Location : PaymentAPI/Controllers/Auth/ManagementController.cs*

1. Update startup.cs, di dalam method ConfigureServices tambahkan baris kode berikut :

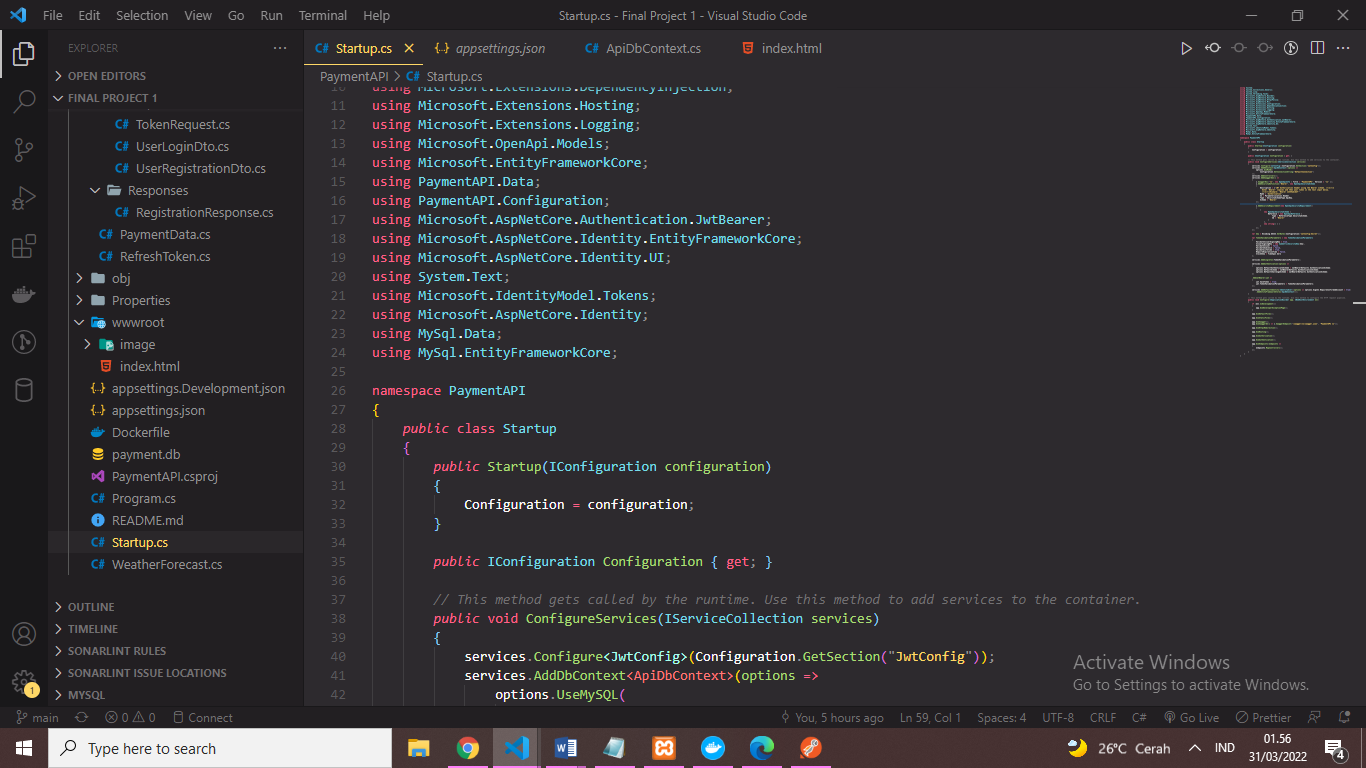
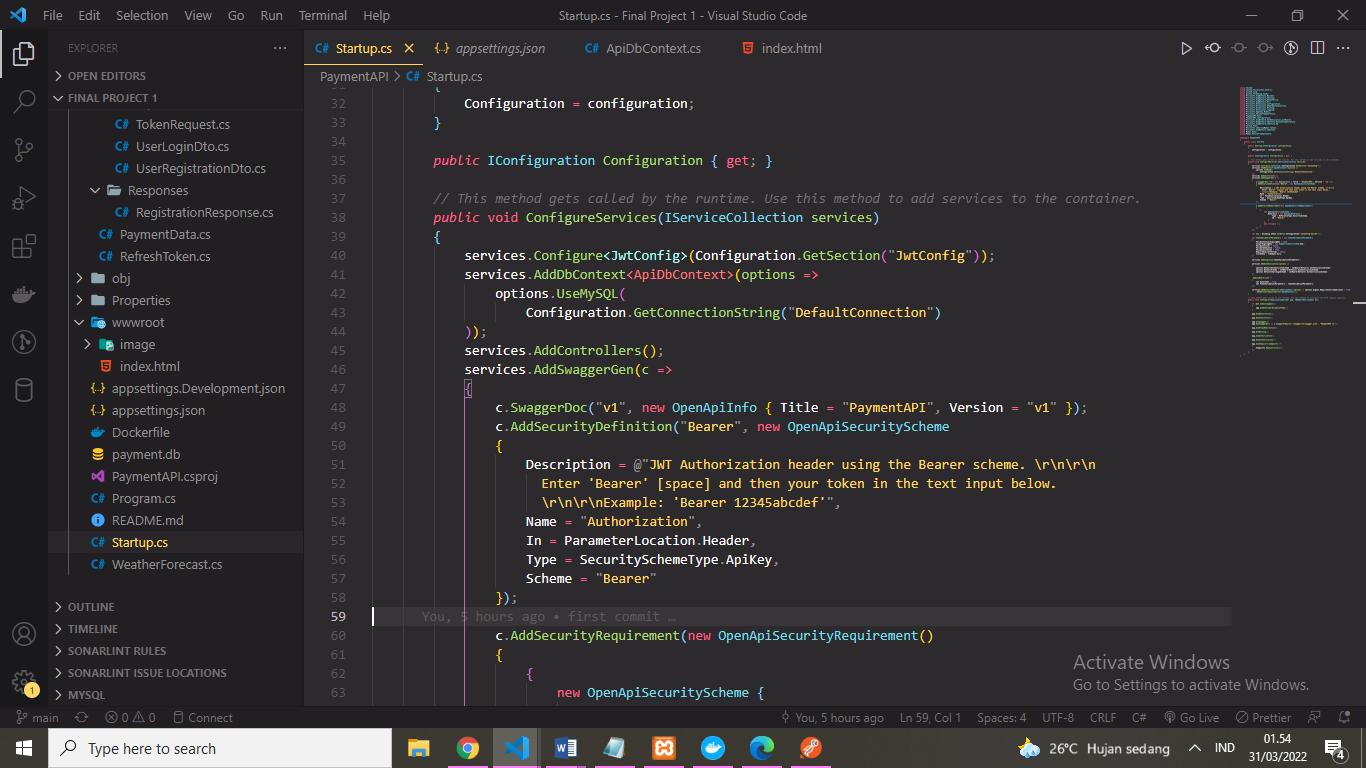
*Location : PaymentAPI/Startup.cs*

C. Initial Migrasi dan Update Database

1. Buat akun baru atau login di <https://remotemysql.com/>
2. Pada file appsettings.json tambahkan baris kode berikut untuk koneksi ke database server. Masukkan nama server, database, username, dan password sesuai dengan yang diberikan dari <https://remotemysql.com/>

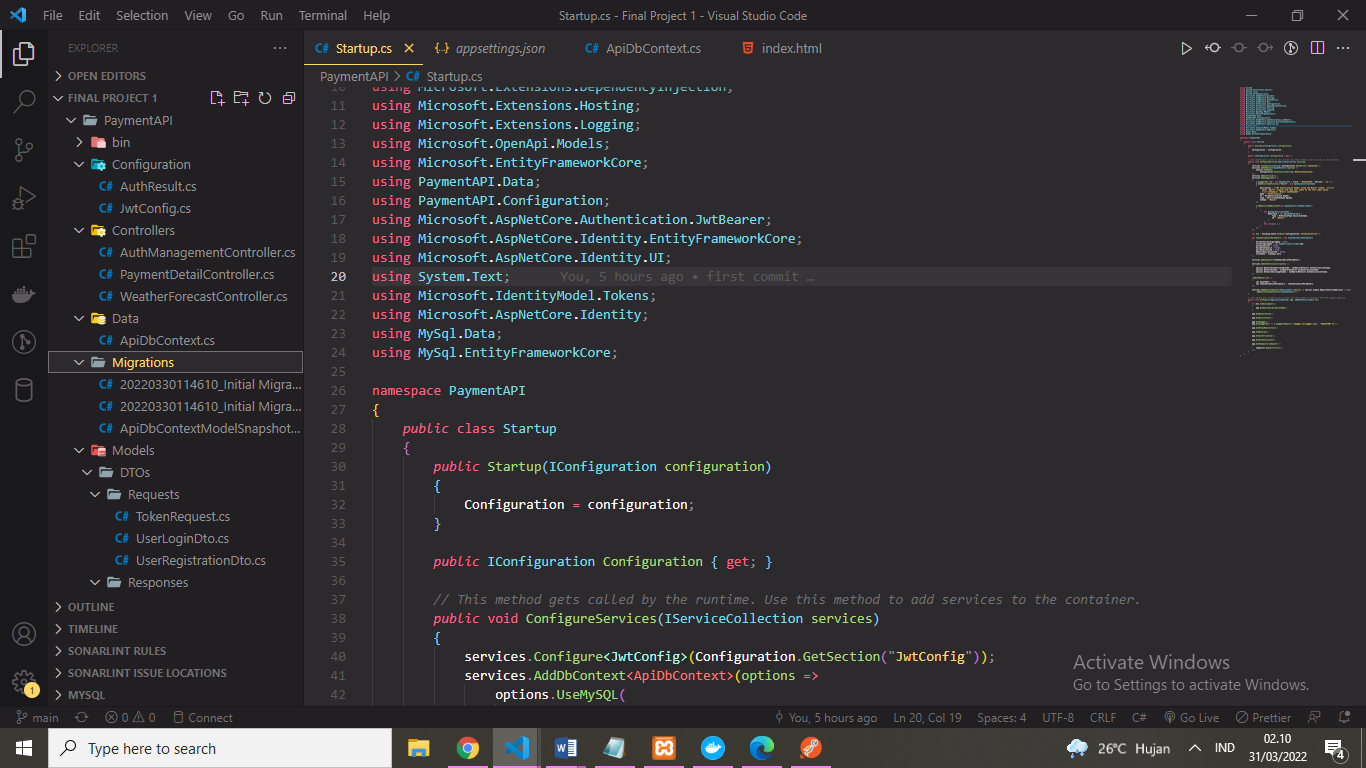
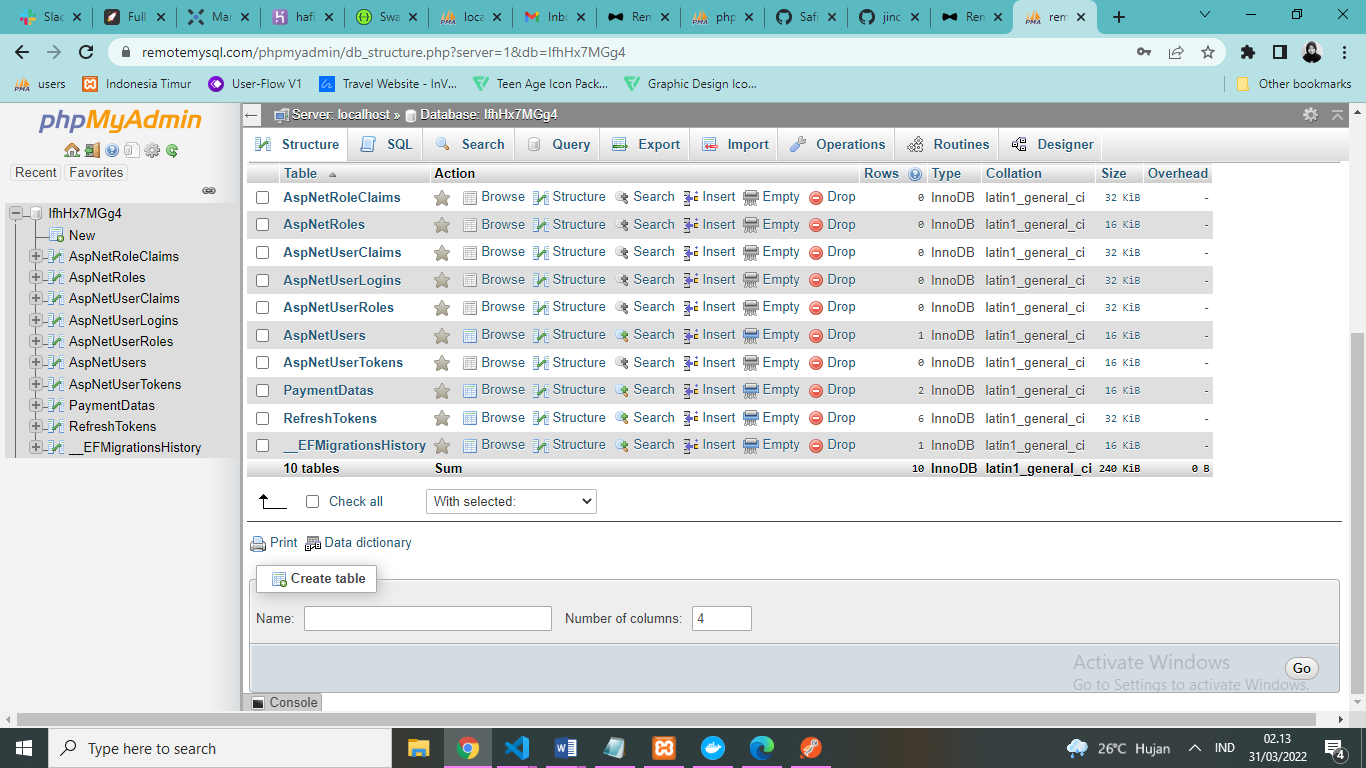
*Location : PaymentAPI/appsettings.json*

1. Untuk koneksi database nya menggunakan MySql, maka ada yang perlu ditambahkan pada file Startup.cs dan library nya

*Location : PaymentAPI/Startup.cs*

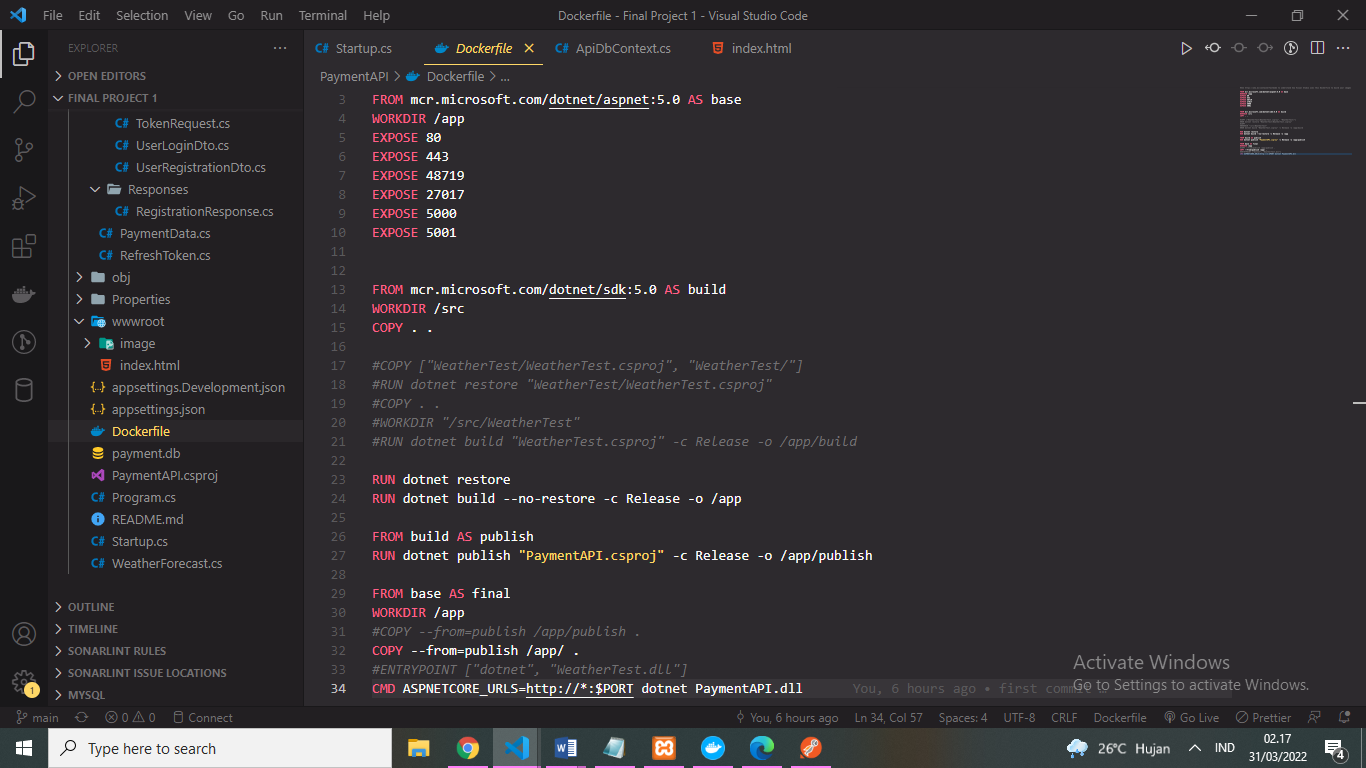
1. Initial migrasi dengan menjalankannya pada terminal :

>> dotnet ef migrations add "Initial Migrations"

>> dotnet ef database update

D. Pembuatan Docker

1. Buat Dockerfile di dalam folder PaymentAPI

*Location : PaymentAPI/Dockerfile*

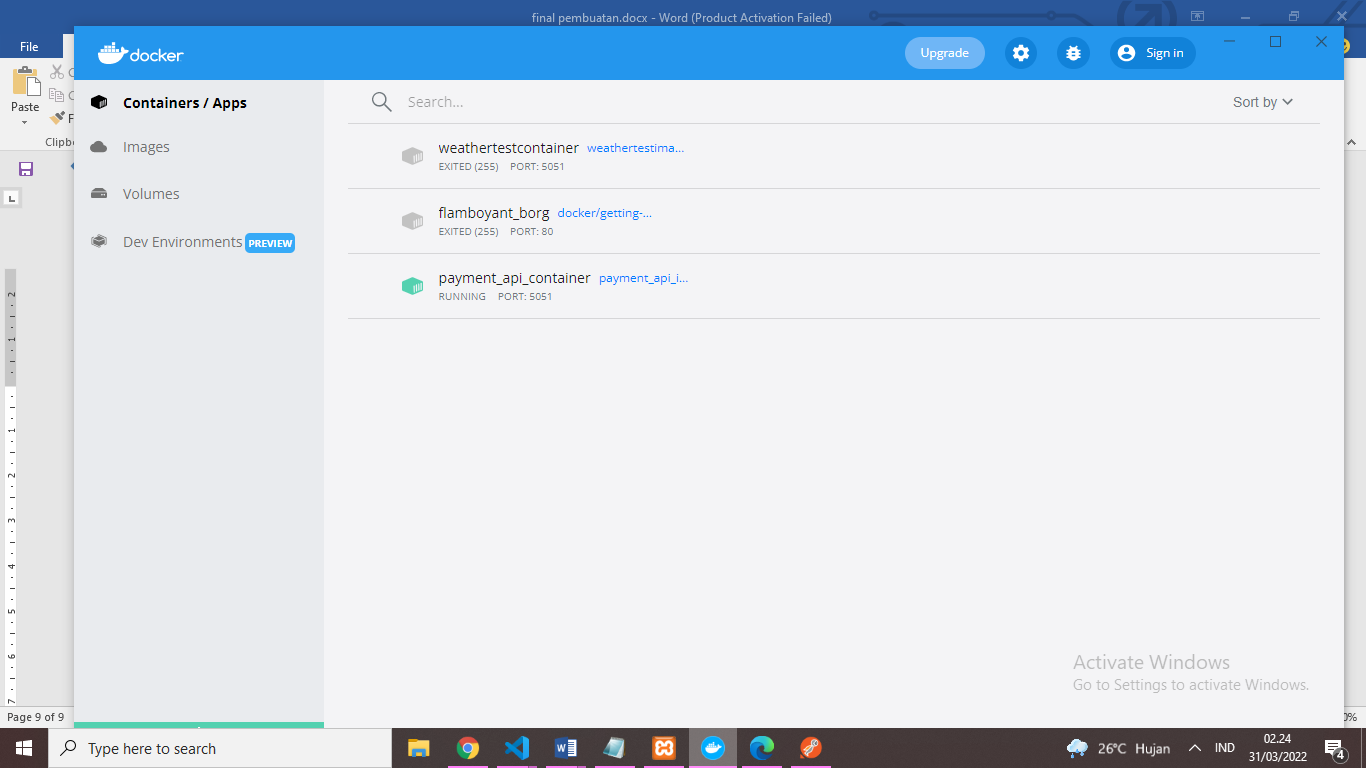
1. Jalankan kode berikut pada terminal :

>> dotnet build

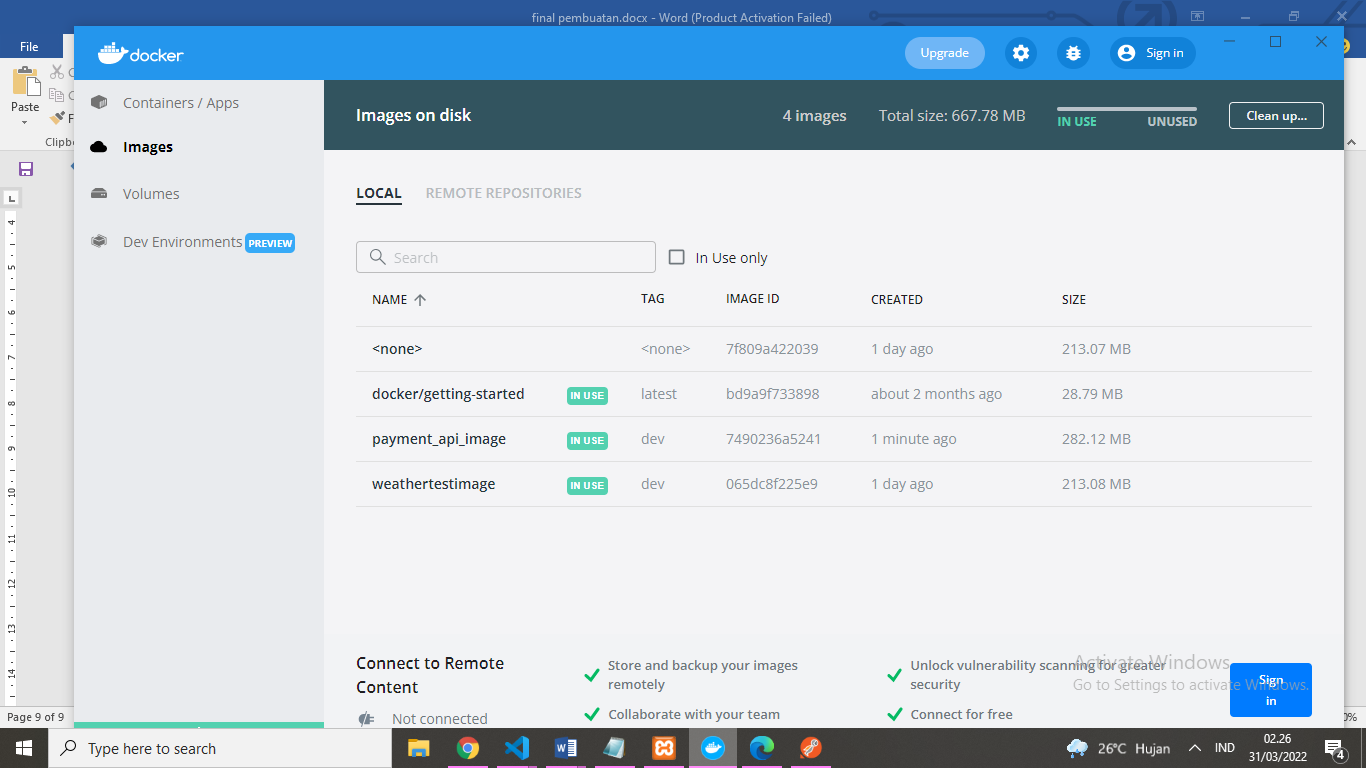
>> dotnet publish -c Release

>> docker build -t payment\_api\_image:dev .

>> docker docker run -d -p 5051:80 --name payment\_api\_container payment\_api\_image:dev

Sehingga project yang telah dibuat akan dimuat ke dalam aplikasi docker

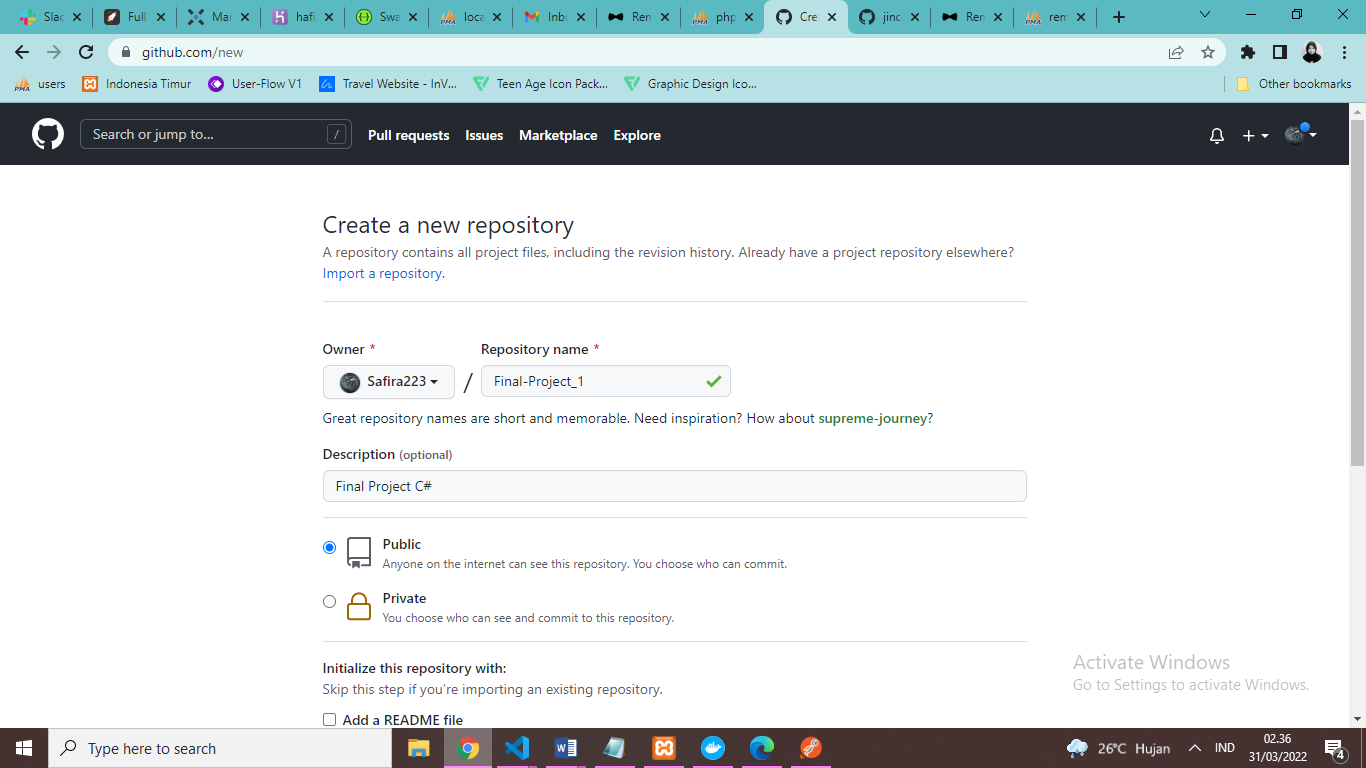
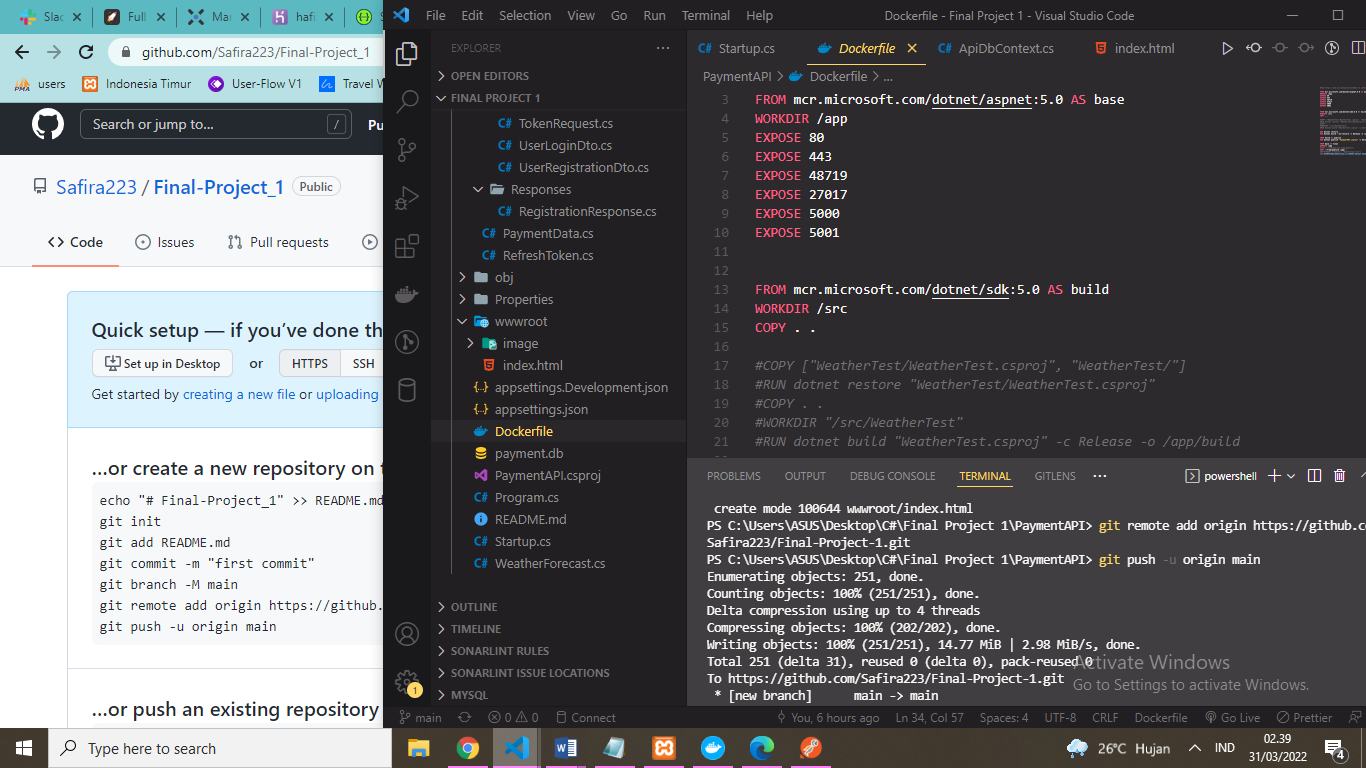
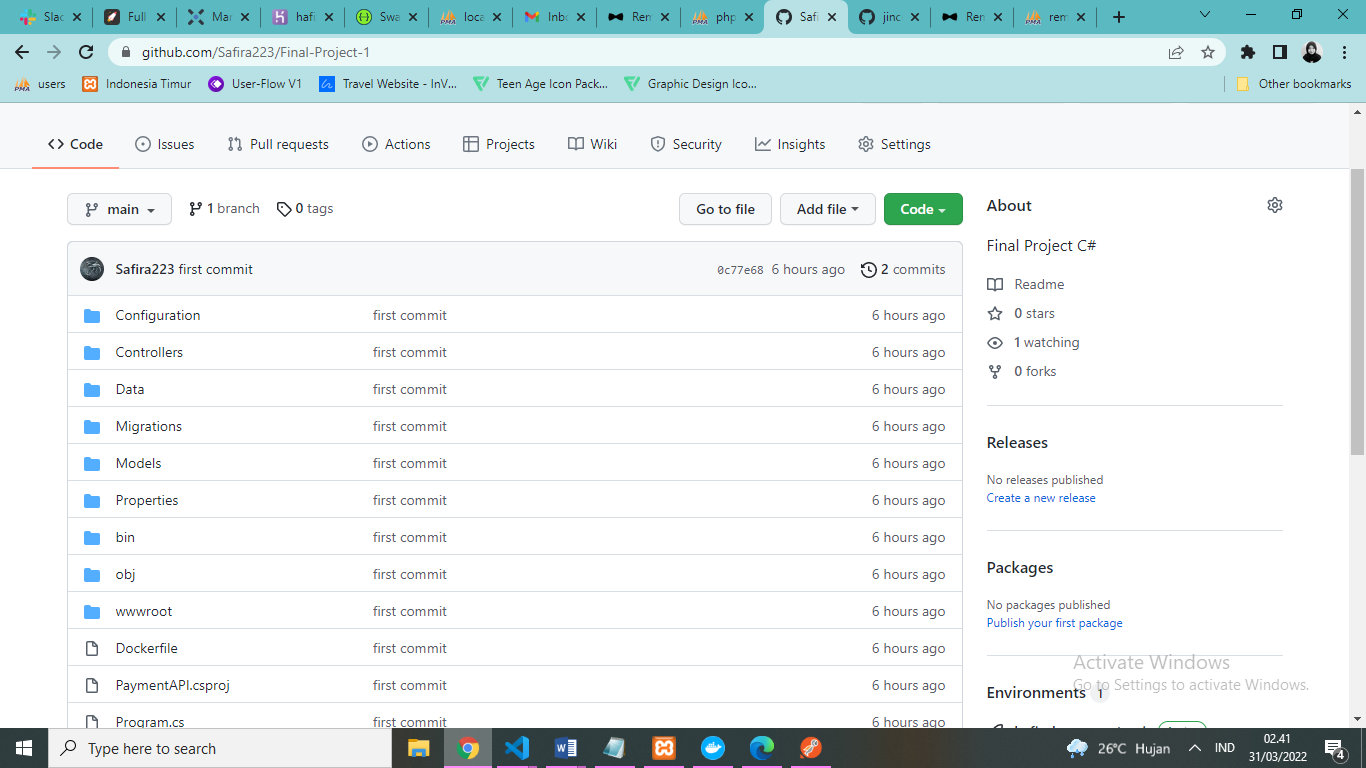
*payment\_api\_container*



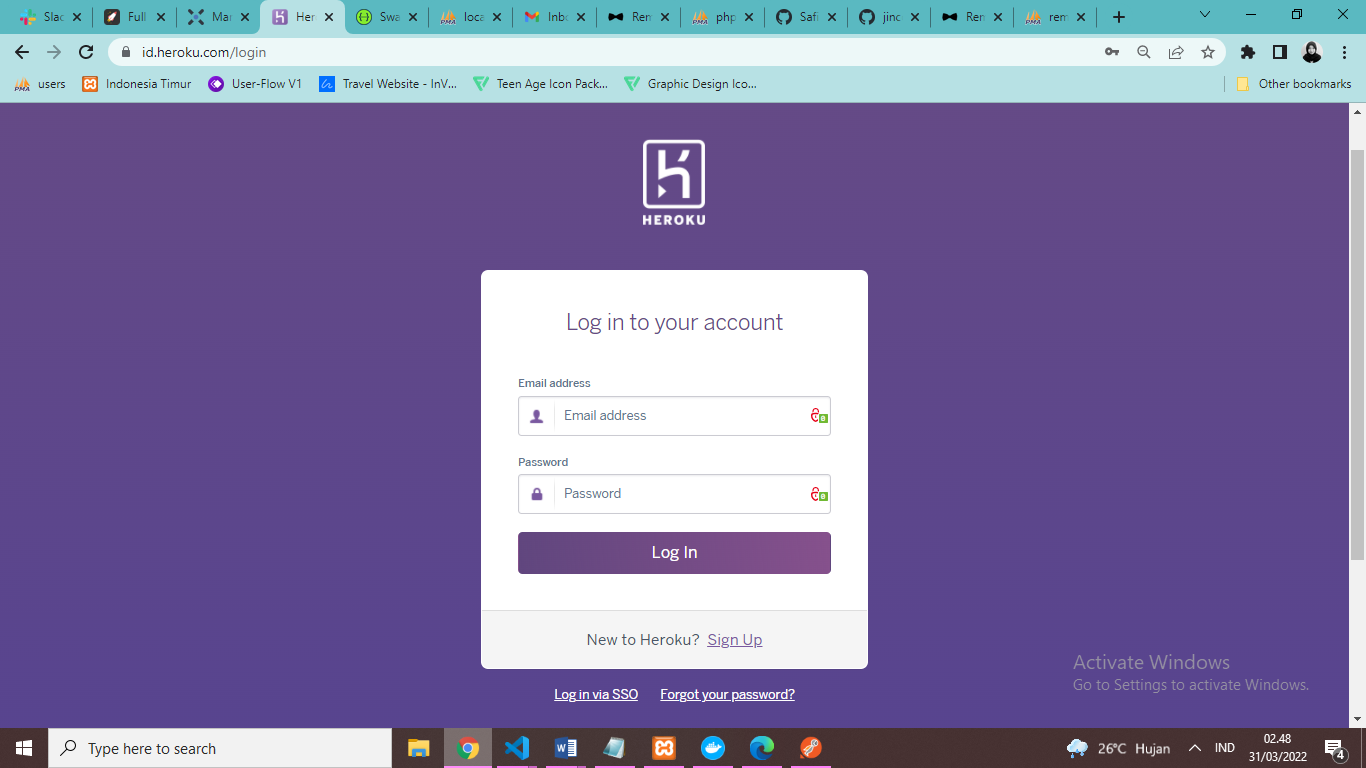
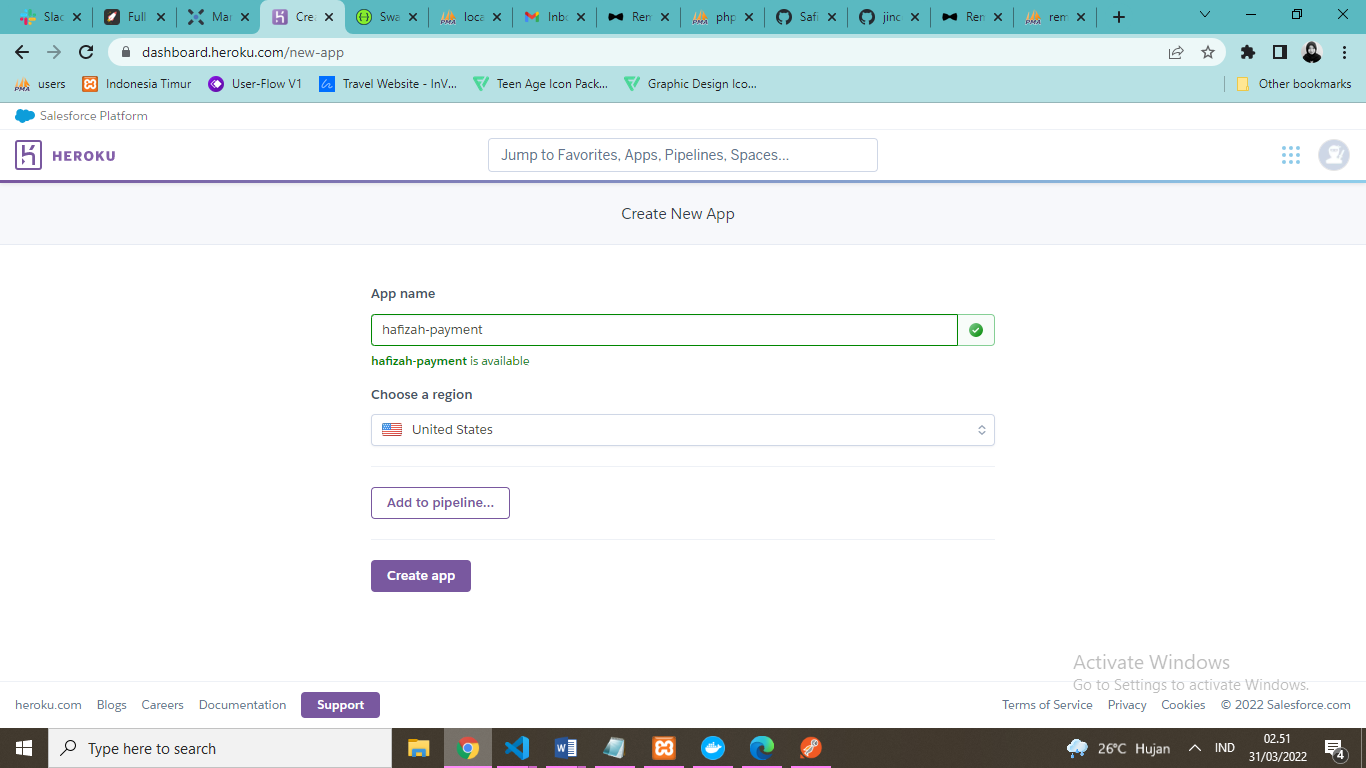
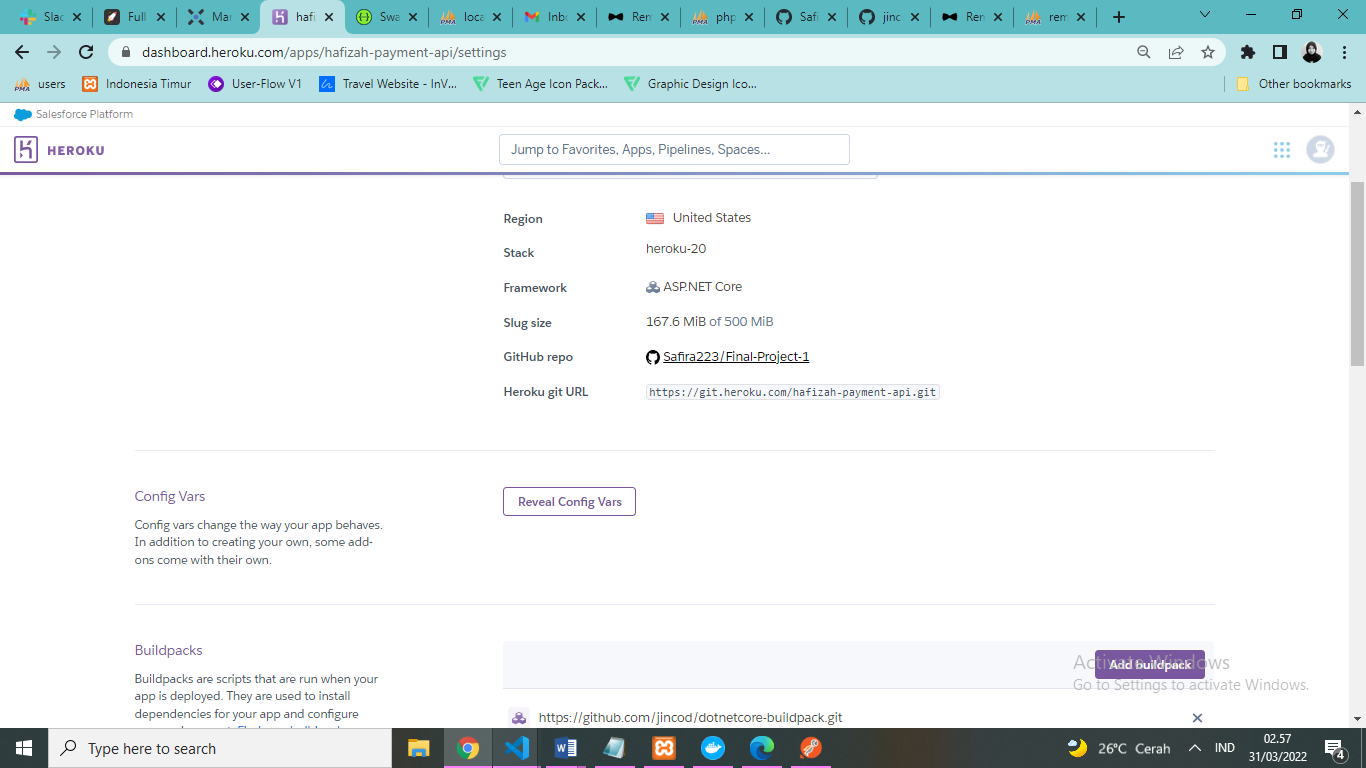
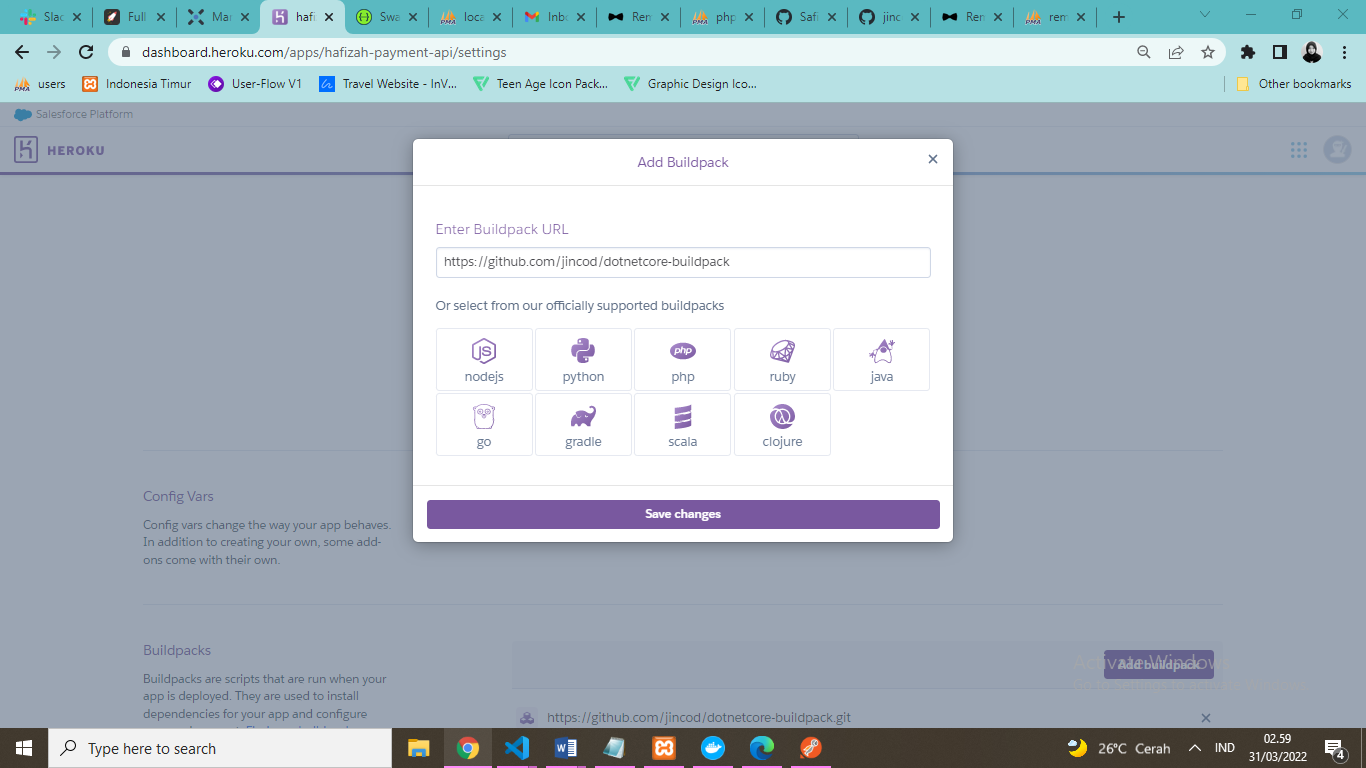
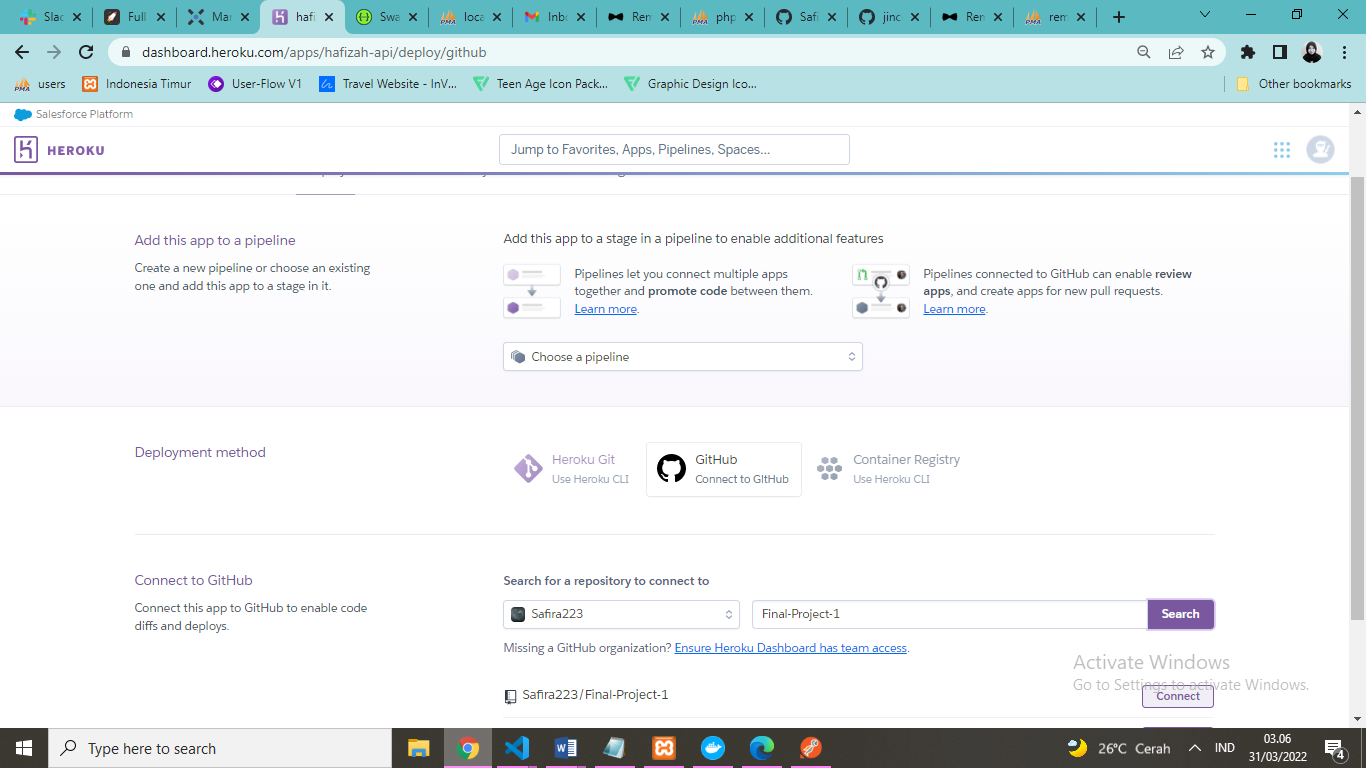
payment\_api\_image

E. Deployment (Github dan Heroku)

* Push Github

1. Login akun github <https://github.com>/
2. Buat repository baru di github
3. Push project ke dalam github melalui terminal
4. Refresh halaman pada github di browser. Apabila folder project sudah muncul di github, maka dapat dinyatakan berhasil tersimpan ke dalam github

* Deploy Heroku

1. Login akun heroku <https://www.heroku.com/>
2. Create new app
3. Klik settings pada navbar, lalu klik tombol Add buildpack
4. Masukkan link berikut, kemudian klik save changes
5. Pada menu Deploy, pilih Connect to Github, lalu masukkan nama repository github dari project yang telah di push. Jika ada maka akan muncul repositornya, lalu klik tombol ‘Connect’.
6. Setelah itu scroll ke bawah lalu klik tombol ‘Deploy Branch’. Tunggu hingga proses selesai. Jika prosesnya sudah selesai, maka link heroku yang berisi final project dapat digunakan di browser. Berikut adalah link untuk final project yang sudah di deploy menggunakan heroku <https://hafizah-payment-api.herokuapp.com/>