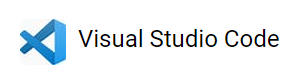
Final Projek

Rest API Online Payment

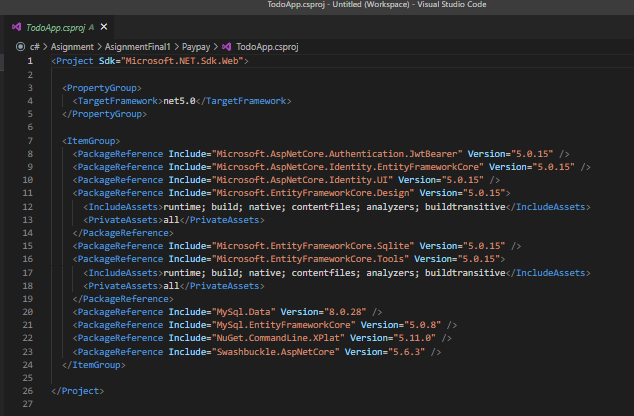
<https://github.com/apeloeza/PaymentAPI.git>

<https://arif-payment-api.herokuapp.com/swagger/index.html>

Muhammad Arif Rahman - FSDO003ONL004 – AsigmentFinal1



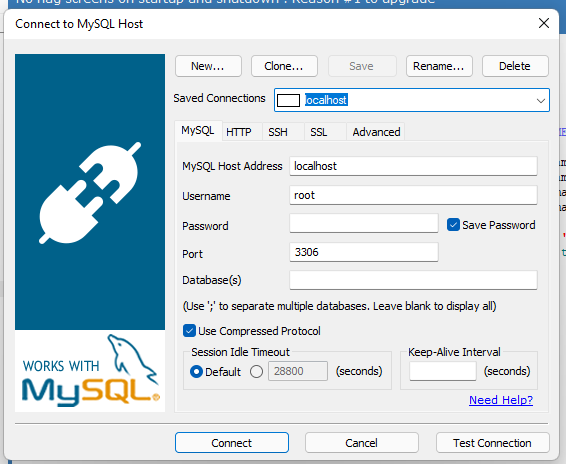
1. Aplikasi persiapan
   1. Visual Code
   2. Browser
   3. Xampp
   4. Docker
   5. VS Code
   6. SQLyog
   7. Postman
2. Membuat projek
   1. Library

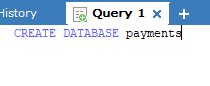


Untuk projek ini kita melanjuti dari tugas sesi 11 dimana untuk library ini kita menambahkan Mysql.Data dan MySql.EntityFrameworkCore

* 1. Membuat Database

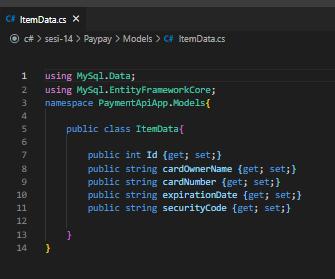
Open SqlYog



Koneksikan terlebih dahulu untuk membuat database local

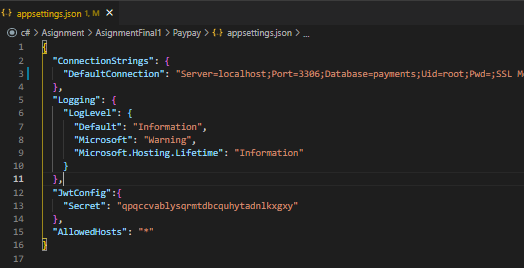
Lalu buat database dengan ketikan “ CREATE DATABASE payments “

* 1. Melakukan perubahan pada ItemData



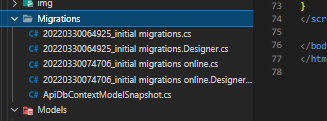
Projek disini melanjutkan dari sesi 11 dimana kita merubah input table Itemdata pada database

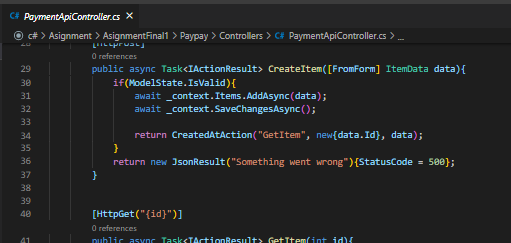
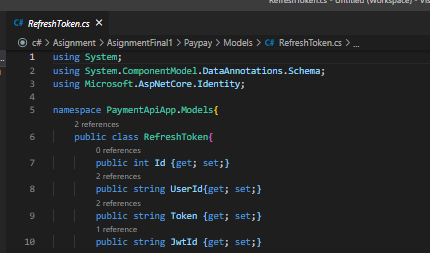
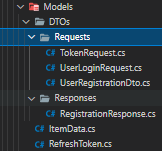
* 1. Membuat konfigurasi database di dalam appsetting.json

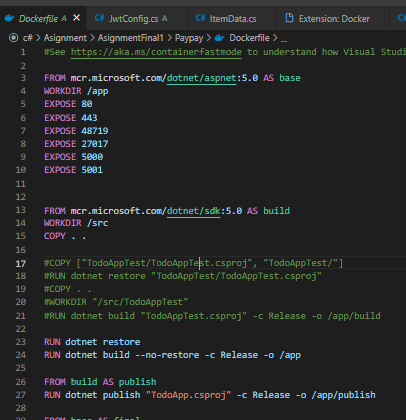


Disini kita mencoba untuk memakai localhost terlebih dahulu untuk pengetesan pada database.

Dotnet ef migration add untuk create database dan migration. Untuk meregenerate table ke databse payments



* 1. Untuk final projek ini kita menggnakan MySQL untuk database yang dipakai terlihat pada gambar library yang dipakai
  2. Payment Conttroller
  3. RefreshToken untuk merefresh token
  4. Kita buat DTOs pada models seperti gambar dibawah ini 

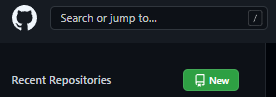
1. Mencoba Local server
   1. Menggunakan Docker dengan membuat file “DockerFile” lalu kita inputkan seperti yang dicontohkan share.io/csharpbatch3 
   2. Jalankan paymenAPI dengan terminal, Selanjutnya jalankan “dotnet build” 
   3. Selanjutnya jalankan syntax “dotnet publish -c Release”
   4. Selanjutnya jalankan syntax “docker build -t weatherimage:dev . “



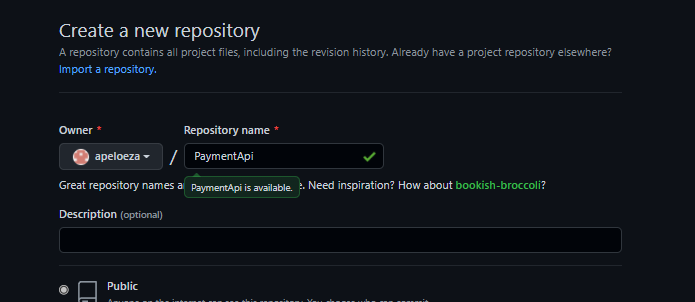
* 1. Selanjutnya jalankan syntax “docker run -d -p 5051:80 --name weathercontainer weatherimage:dev”
  2. Selanjutnya jalankan dengan syntax “docker run”

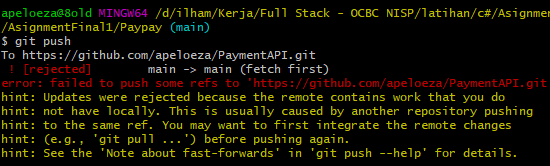
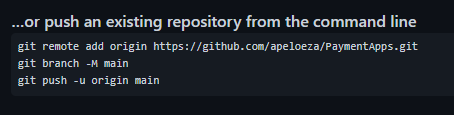
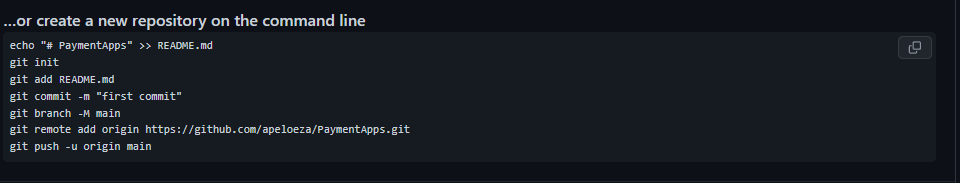


1. Push Directory ke Github
   1. Buat directory baru



lalu input repository name.

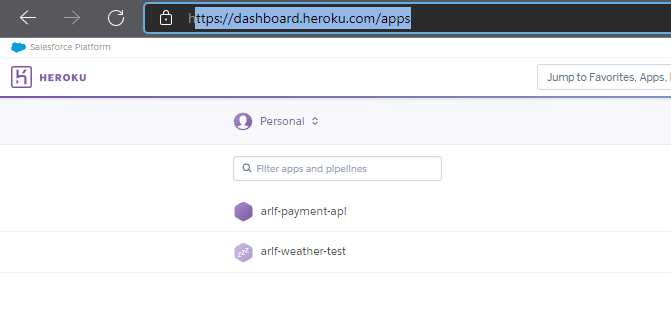
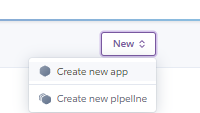
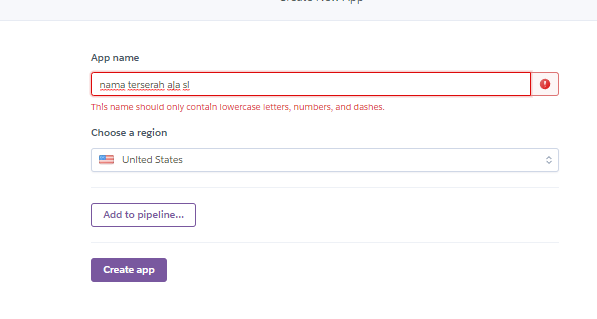
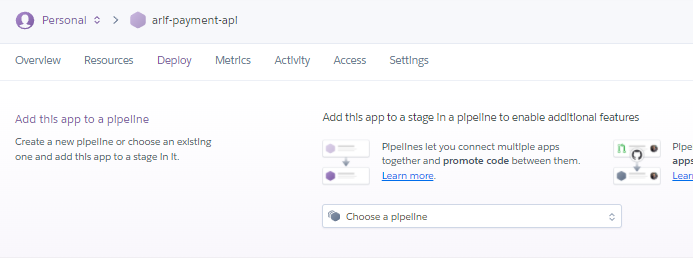
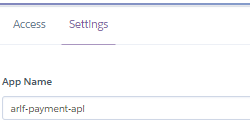
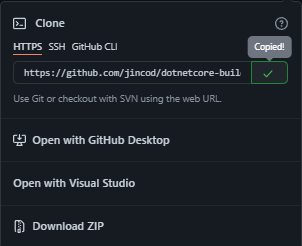
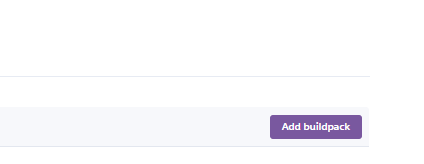
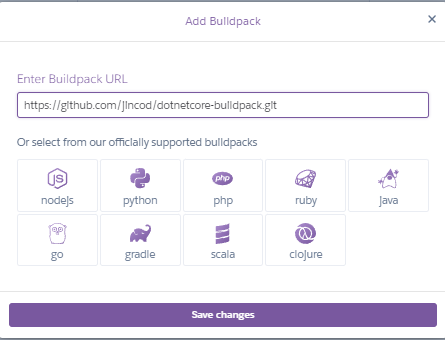
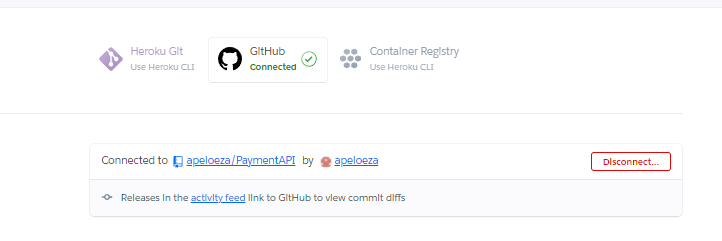
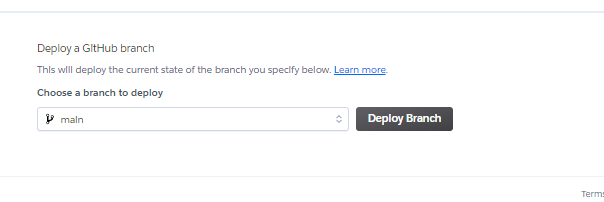
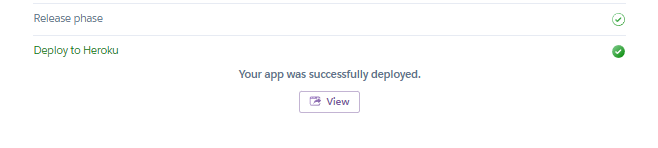
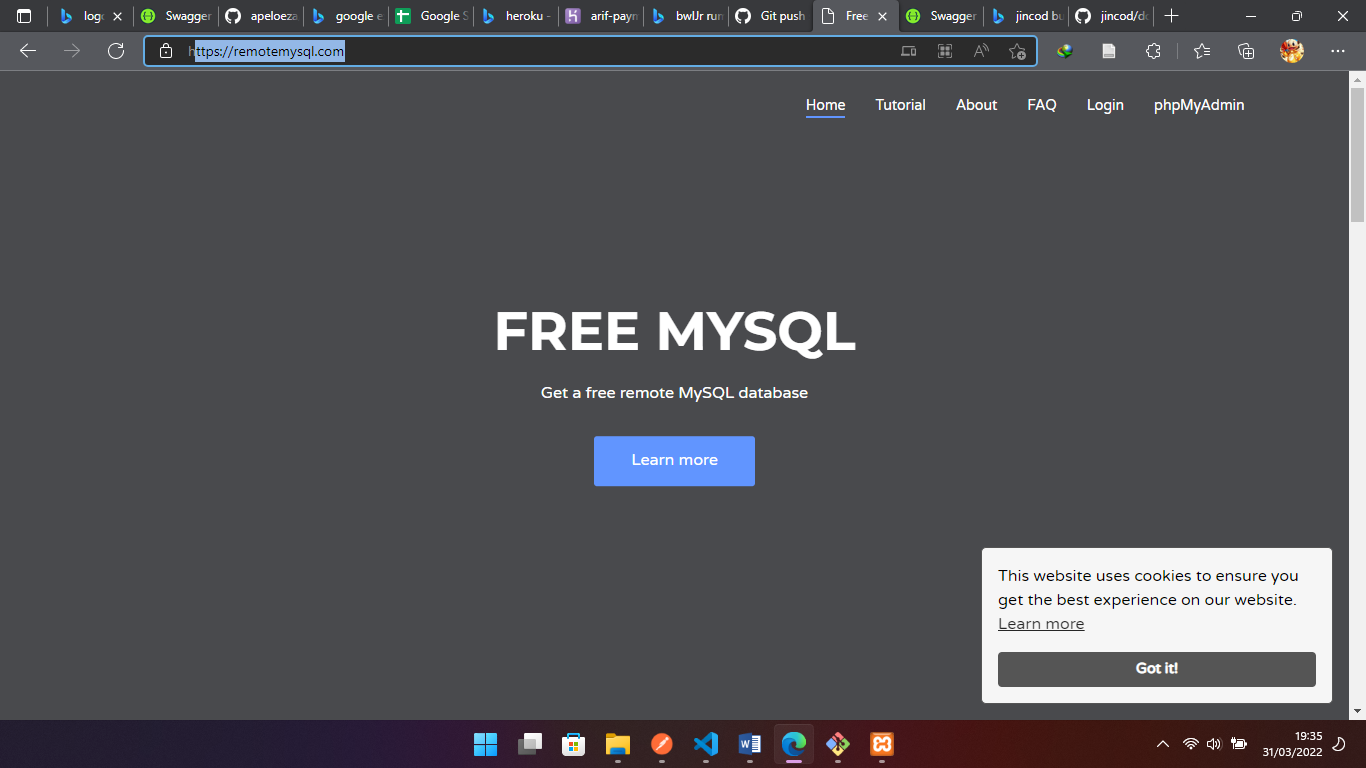
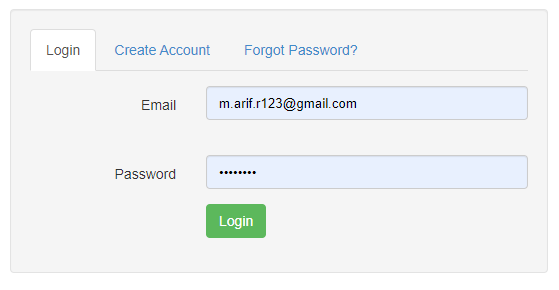
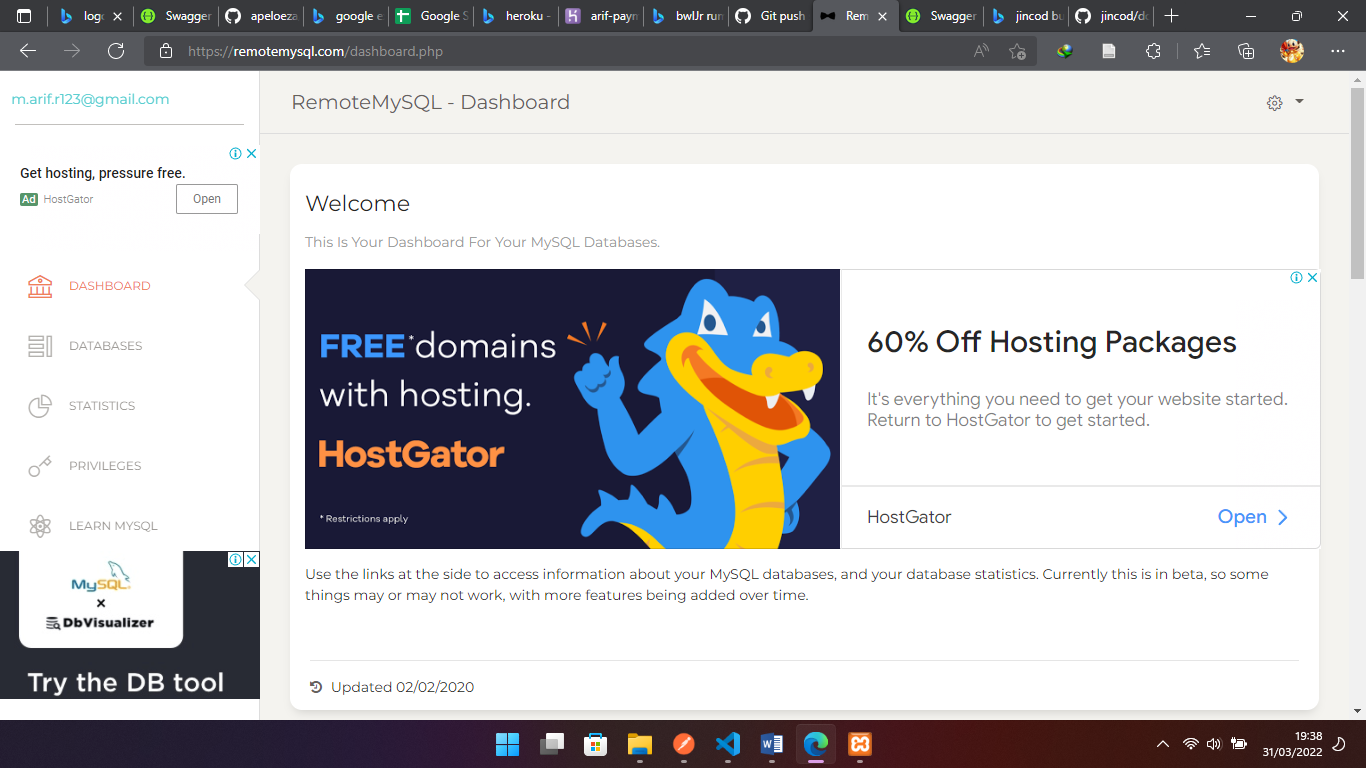
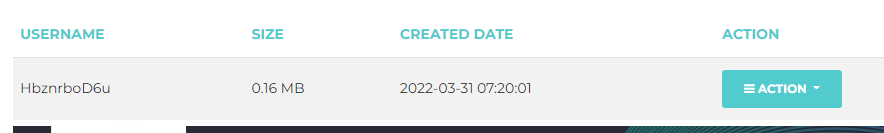
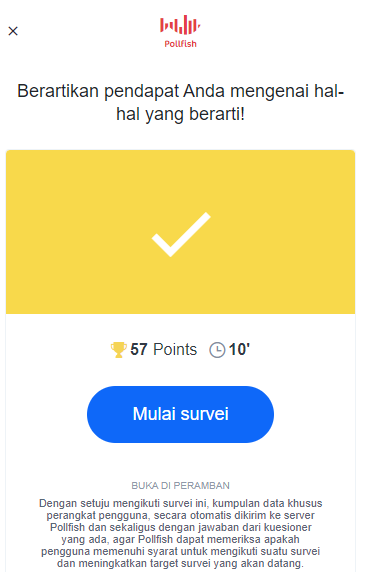


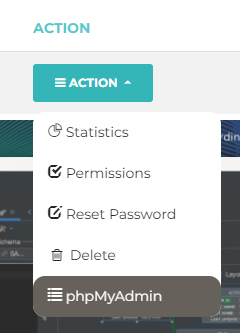
Selanjutnya ikuti petunjuk repository dengan menggunakan Git BASH

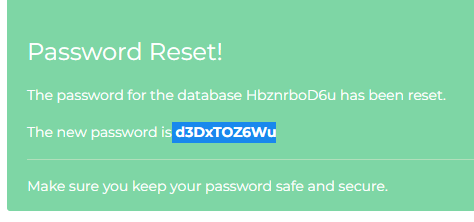
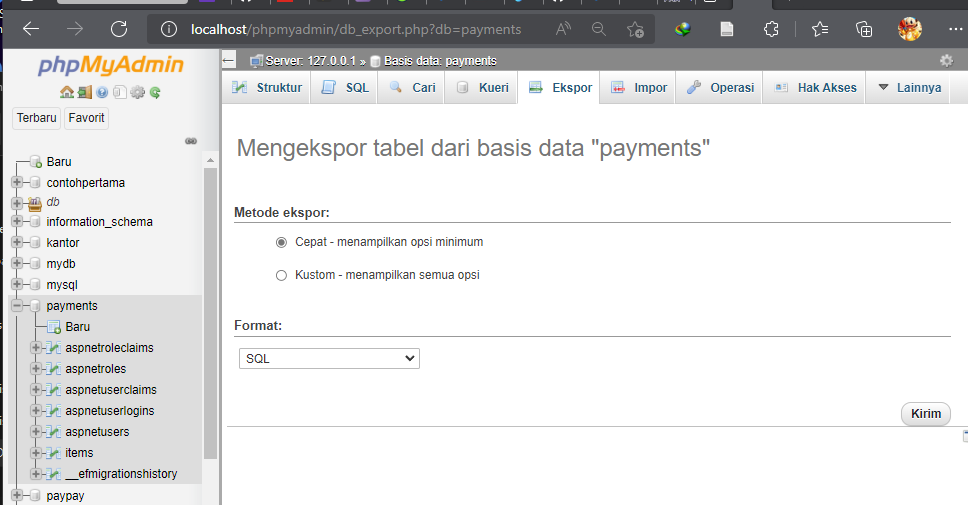
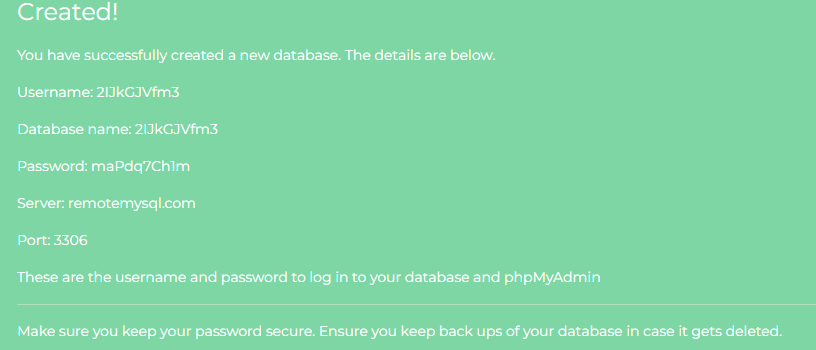
Jika ada eror seperti gambar dibawah ini

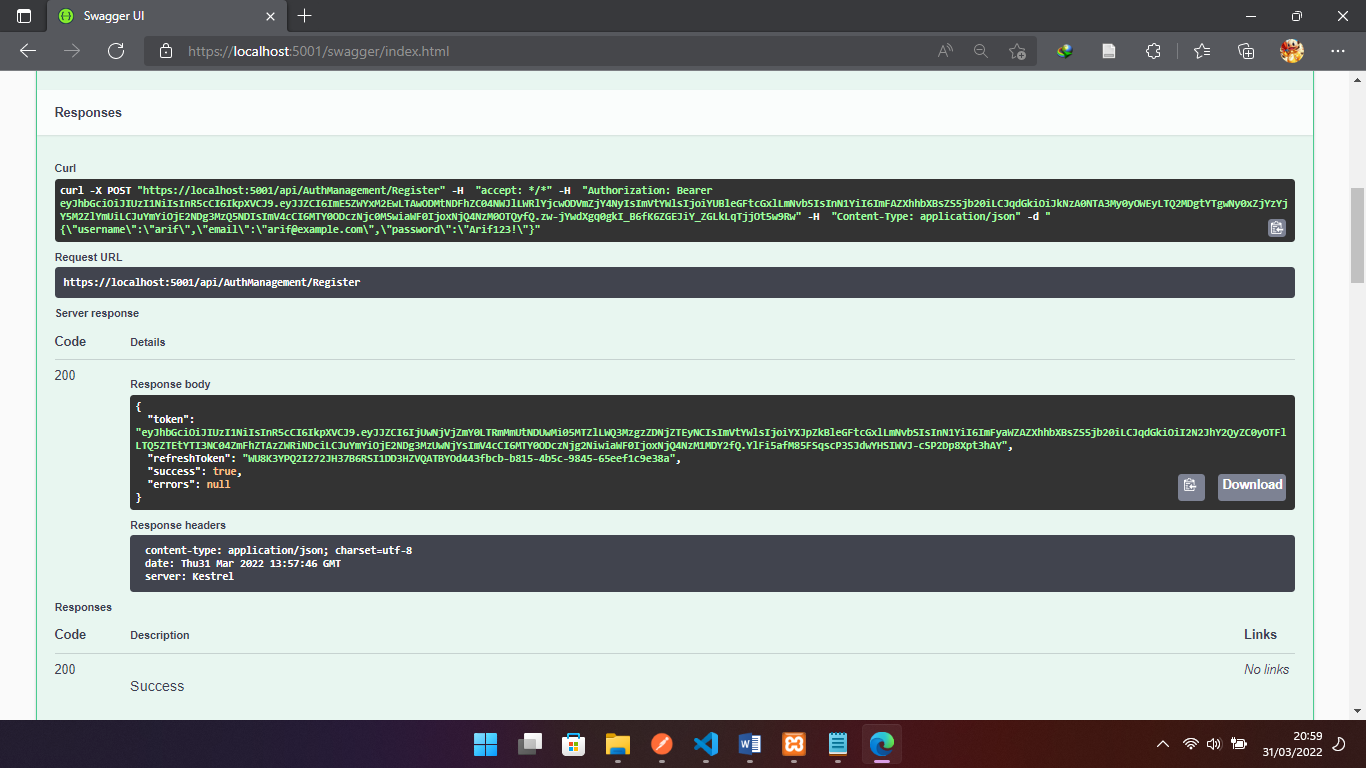
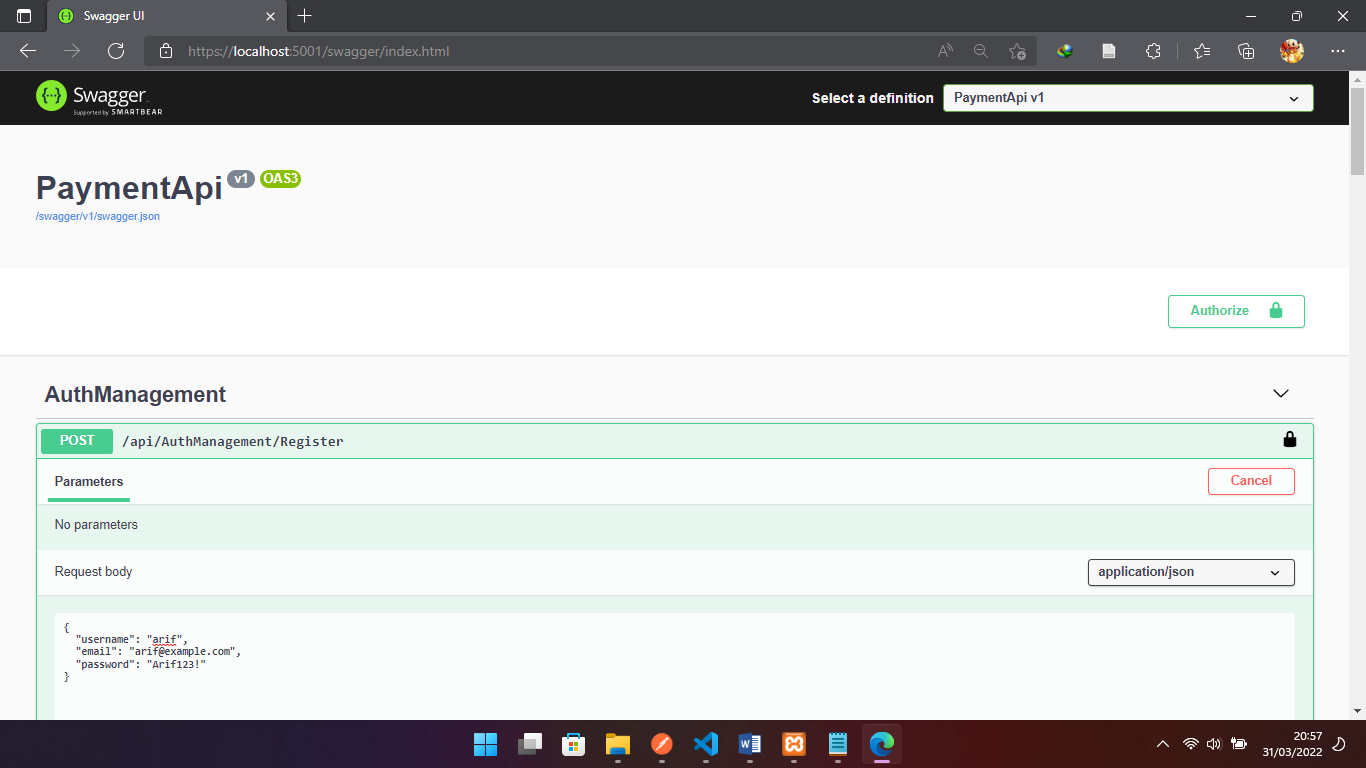
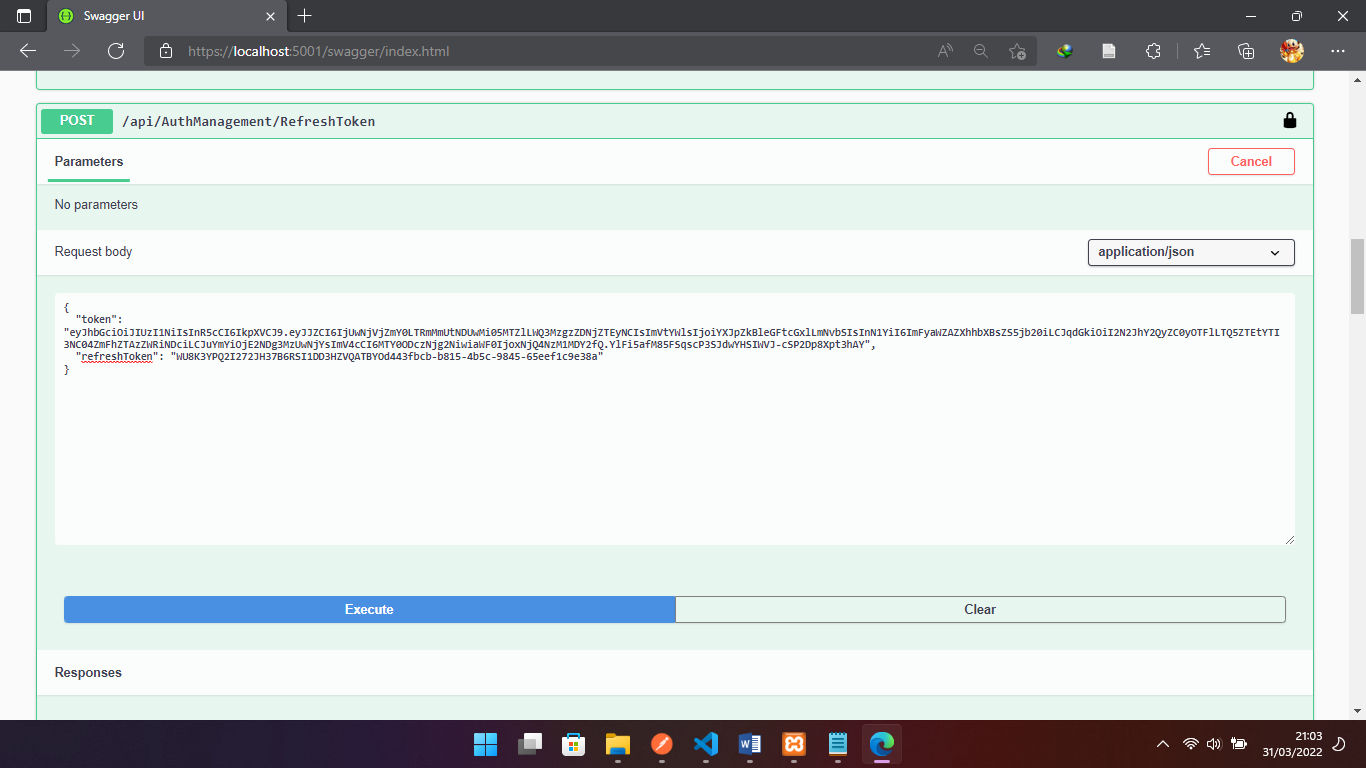
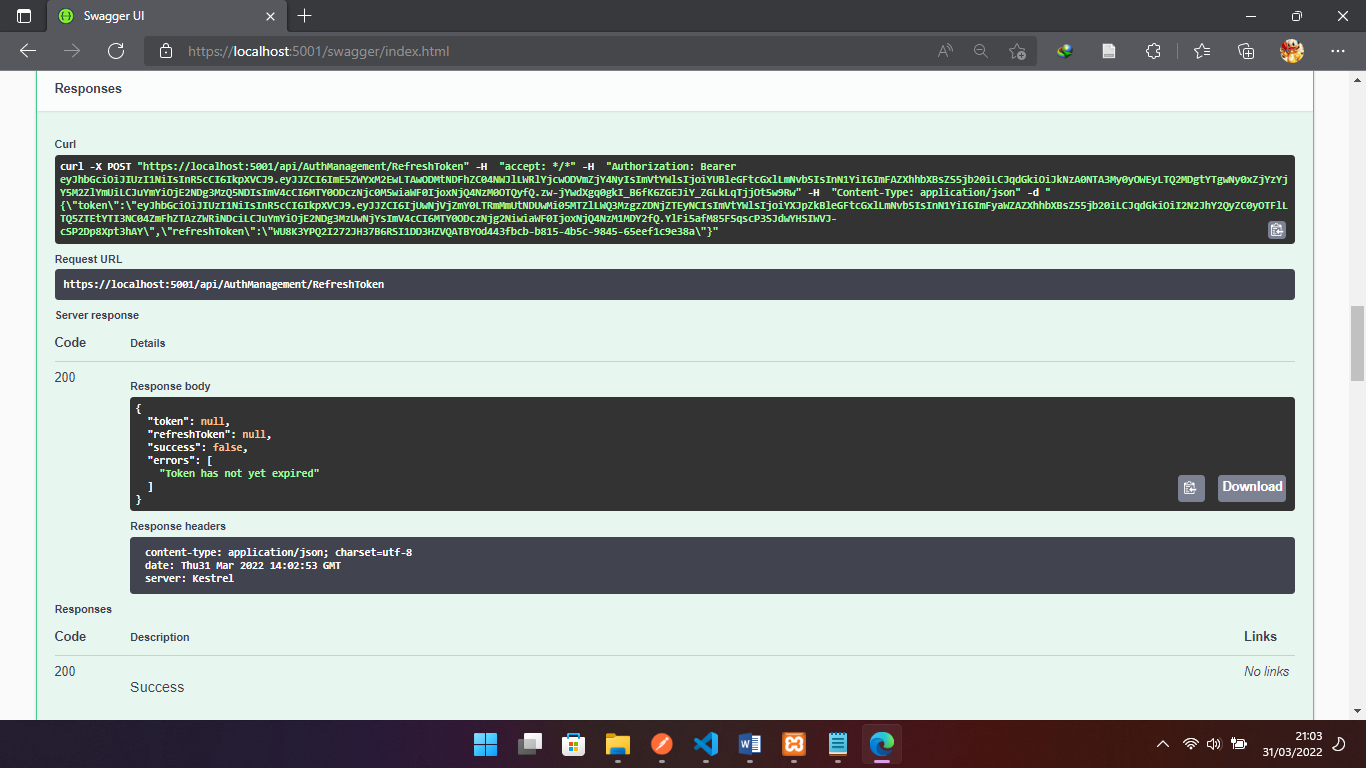
Ikuti step dibawah ini.

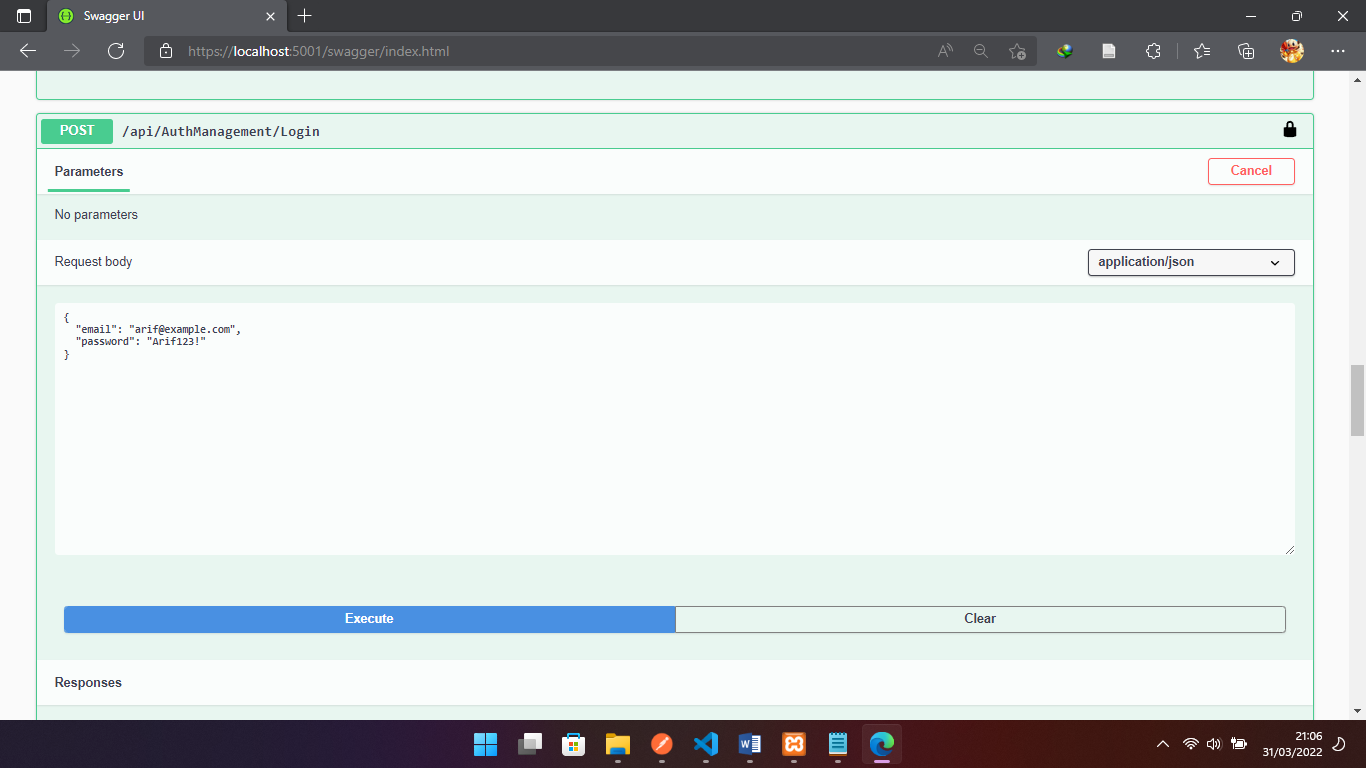
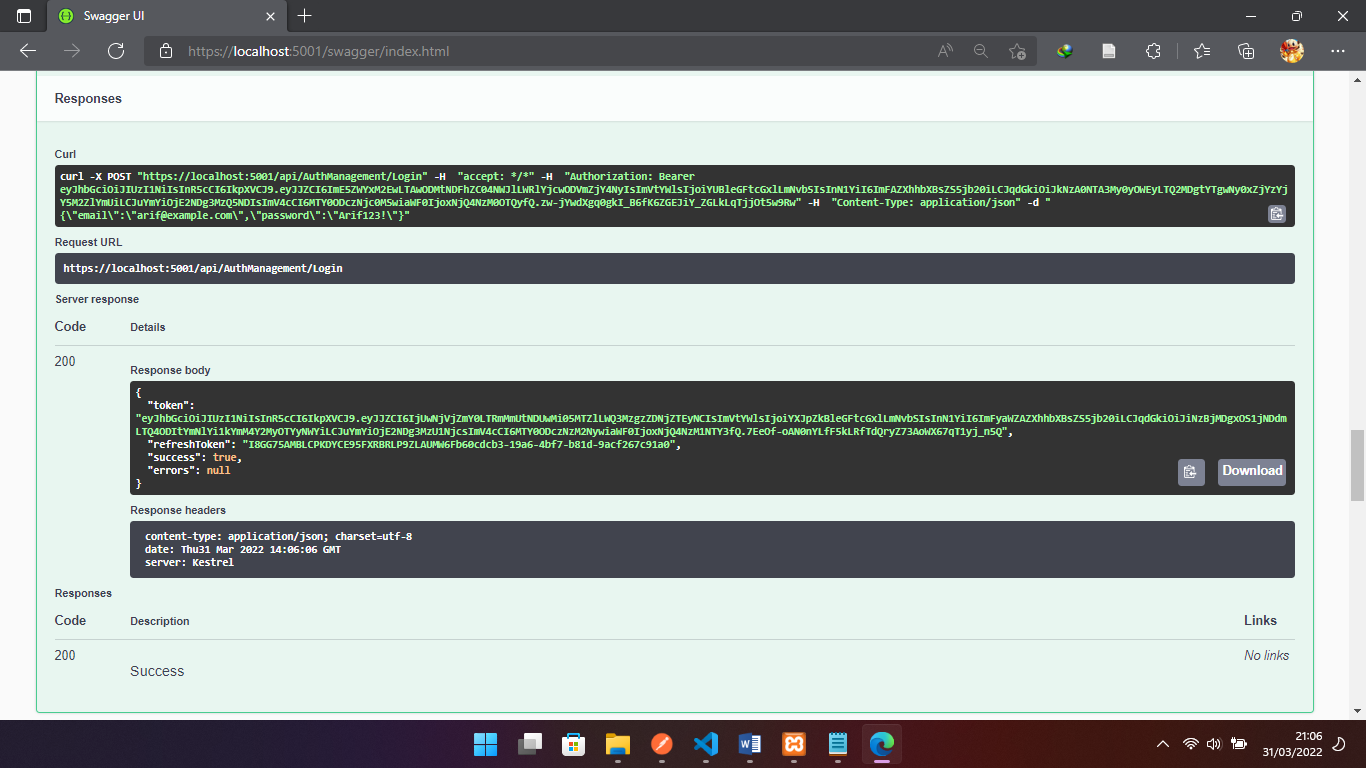
<https://github.com/git-lfs/git-lfs/issues/4320?msclkid=8302b842b0b011ecaff56be4eb9c740f>

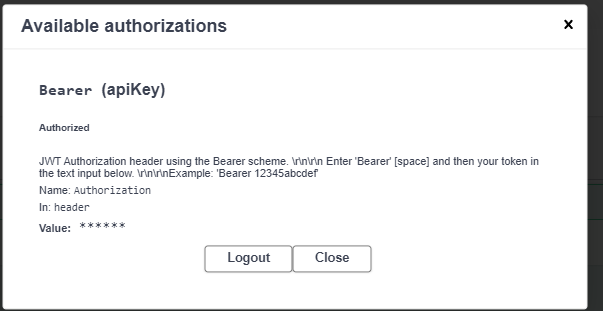
1. Penggunaan heroku sebagai hosting projek github
   1. Kunjungi heroku “ <https://dashboard.heroku.com/apps>” 
   2. Selanjutnya buat app baru pilih fitur create new app 
   3. Selanjutnya beri nama pada aplikasi yang ingin dibuat 
   4. Selanjutnya akan tamil tampilan awal setelah create app 
   5. Pilih setting lalu pilih add buildpack dan kita input dengan jincood github 🡺🡺
   6. Selanjutnya pilih deploy lalu hubungkan dengan github lalu untuk connected to github search nama directory github yang telah direpository. 
   7. Jangan gunakan enable automatic deploy lalu pilih deploy branch 
   8. Setelah selesai view projek dan projek akan menjadi online
   9. Asdasd
2. Menghubungkan database dengan remote MySQL
   1. Buat akun di <https://remotemysql.com>
   2. Pilih login lalu login dengan akun yang dimiliki jika belum segera create acount. 
   3. Selanjutnya akan tampil tampilan seperti di bawah ini. Lalu pilih “”
   4. Ikuti sufvey dan akan mendapatkan seperti gambar dibawah ini
   5. Pilihb action lalu pilih phpmyadmin

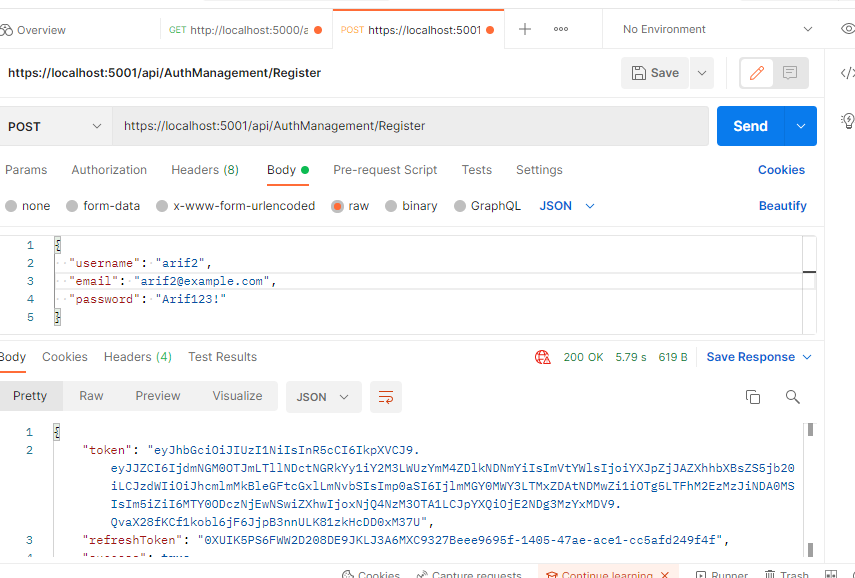
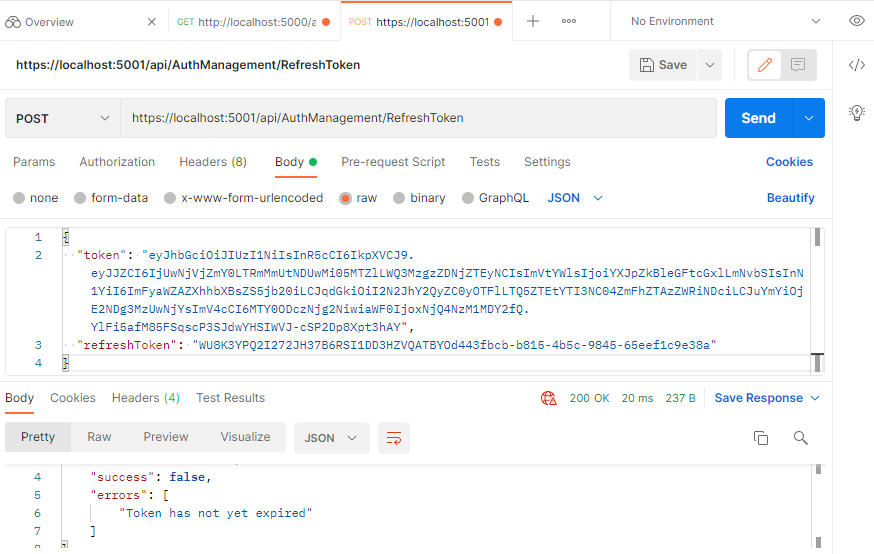
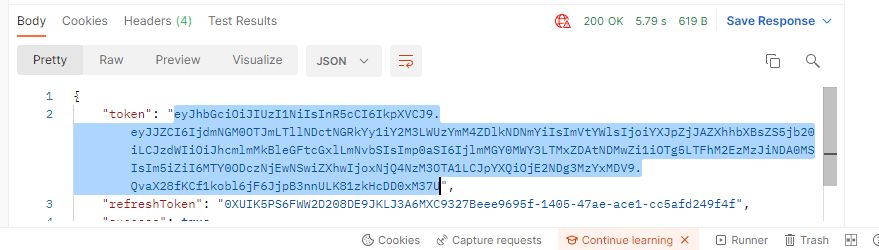
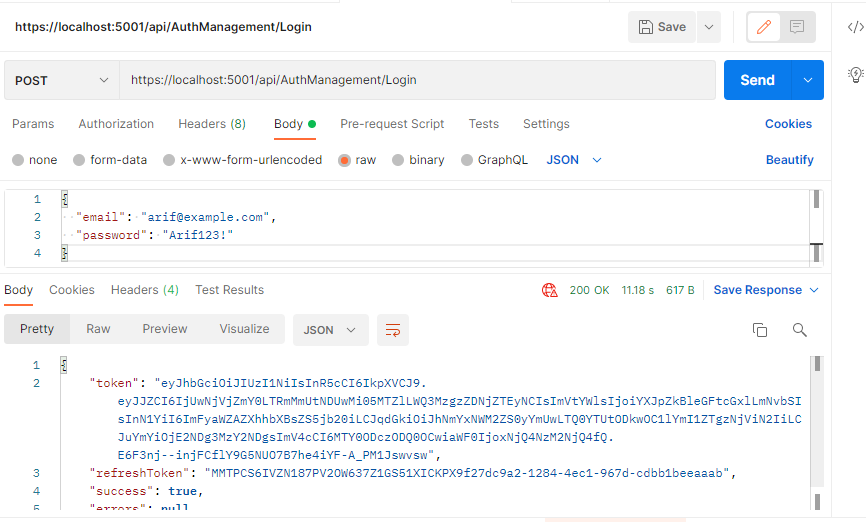


* 1. Jika lupa password gunakan reset password maka akan keluar 
  2. Selanjutnya setelah tampilam MySQL lalu inputkan nama Server, Port, Database, Uid dan Pasword. 
  3. Asdasdasdasdasdasdasdas

1. Ujicoba PaymentApi
   1. Percobaan AuthManagement ini kita mencoba untuk register dan berhasil dimana memberikan token dan refresh token selanjutnya kita lanjut input token
   2. Input token akan berhasil dimana terlihat keterangan “token has not yet expired” dikarenakan untuk waktu expired token kita berikan 30 menit.
   3. Selanjutnya kita melakukan login, ketika login berhasil lalu akan kelaur token seperti pada tampilan dibawah ini

* 1. Kita akan membuka authorization dengan copy token yang muncul pada token ketika login dengan akun yang sudah ada dengan menginputkan “Baerer <token>”
  2. Asdasd

1. Pencobaan Postman
   1. Kita coba untuk “register” dimana kita gunakan “post” lalu pilih “body” serta “json” selanjutnya pilih fitur “send” maka akan tampil respon dengan pilih “body” keterangan token yang kita akan copy kepercobaan selanjutnya. 
   2. Kita coba akses “RefreshToken” dengan menginputkan token dan refrest token dan hasil ynag didapat tampilan response “token has not yet expired” 
   3. Kita coba untuk “login” dimana kita gunakan “post” lalu pilih “body” serta “json” selanjutnya pilih fitur “send” maka akan tampil respon dengan pilih “body” keterangan token yang kita akan copy ke Bearer dengan imputan Bearer <newTokenLogin>. 
   4. Untuk terakhir kita mencoba untuk input Bearer <tokenLogin>

