**Email OTP Module**

To test the module, following are the **prerequisites** to be configured:

* Change the values in appsettings.json of OTP.Console project

**OTPModule\OTP.Console\appsettings.json**



Provide the connection string to DB in “**DBConnectionString**” property :

Ex: "DBConnectionString": "data source=(localdb)\\MSSQLLocalDB;initial catalog = AuthenticationDB;integrated security= True;MultipleActiveResultSets= True;App=EntityFramework"

To test out email feature, give values in “**Password**” and “**From**” properties of an email from which the email can be sent.

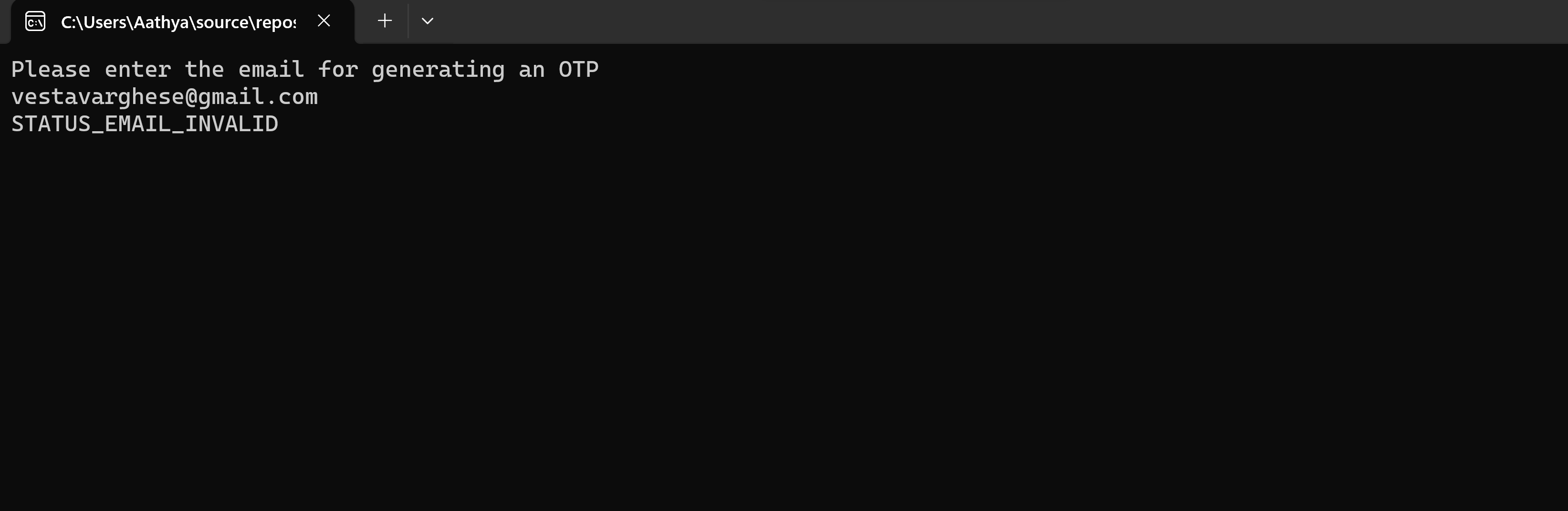
* Run migration script in the DB – MigrationScript.sql (Attached in the mail as well as repository). Entity framework migrations has been used to generate the script.
* Set startup project as OTP.Console

**Use Cases :**

**1)When the user types in an email which is not from a valid domain**

I/P : [vestavarghese@gmail.com](mailto:vestavarghese@gmail.com)

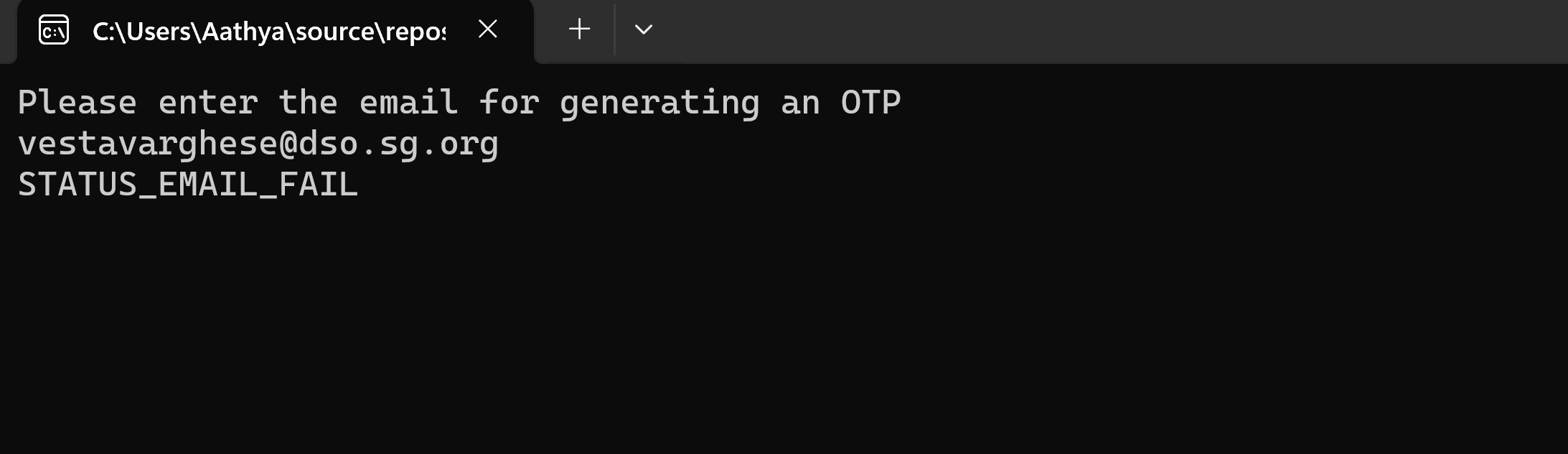
O/P: STATUS\_EMAIL\_INVALID



**2)When the user types in an email which is of correct format, but sending email is a failure.**

I/P : vestavarghese@dso.sg.org

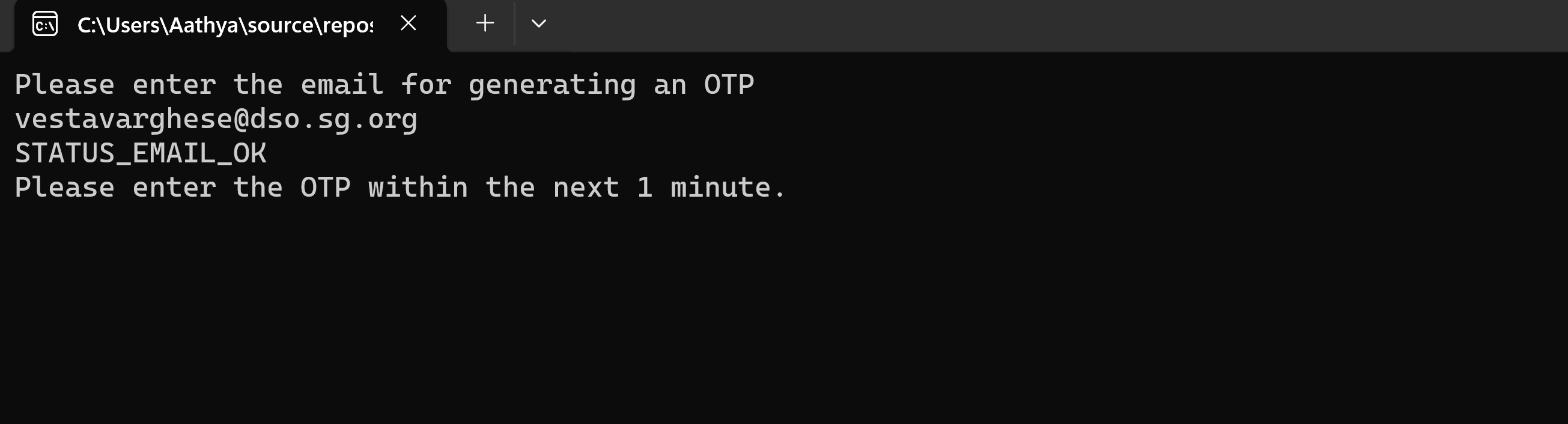
O/P: STATUS\_EMAIL\_FAIL



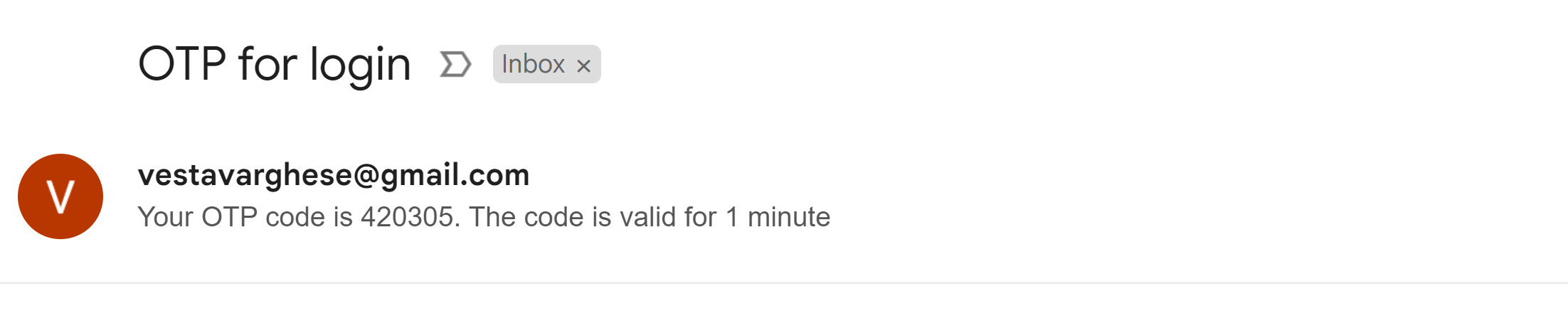
**3)When the user types in an email which is of correct domain, and sending email is a success.**

I/P : vestavarghese@dso.sg.org

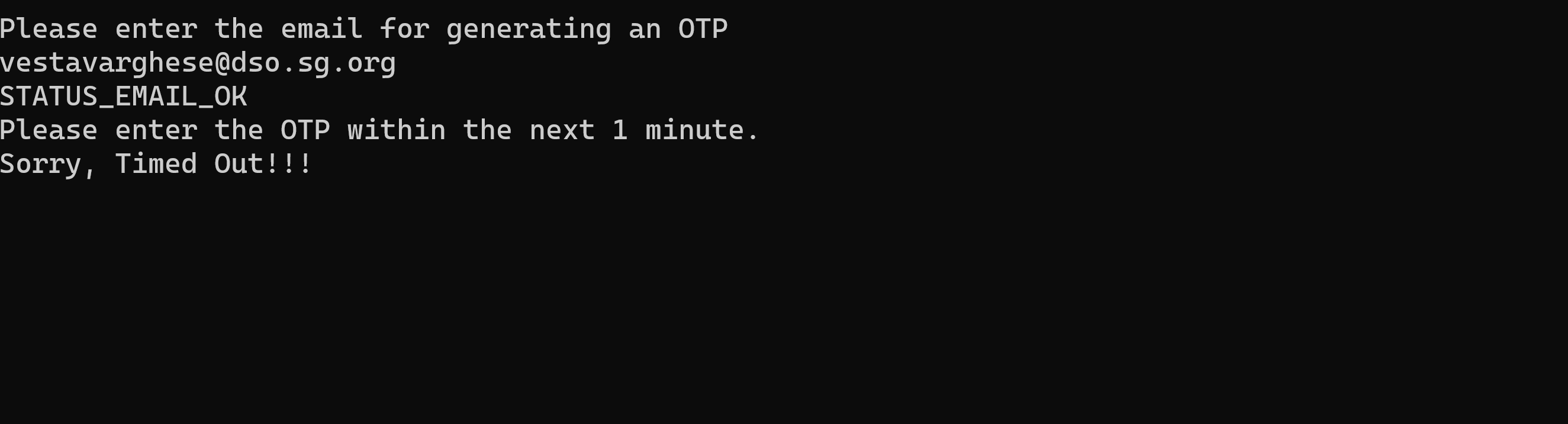
O/P: STATUS\_EMAIL\_OK



**Screenshot of email received for OTP**



**4)When email is sent successfully, but validation times out.**



5) **When the email is sent successfully, but validation of OTP fails**

I/P: 123456

O/P: STATUS\_OTP\_FAIL

**When a wrong OTP is given the 10th time,**

I/P: 124353

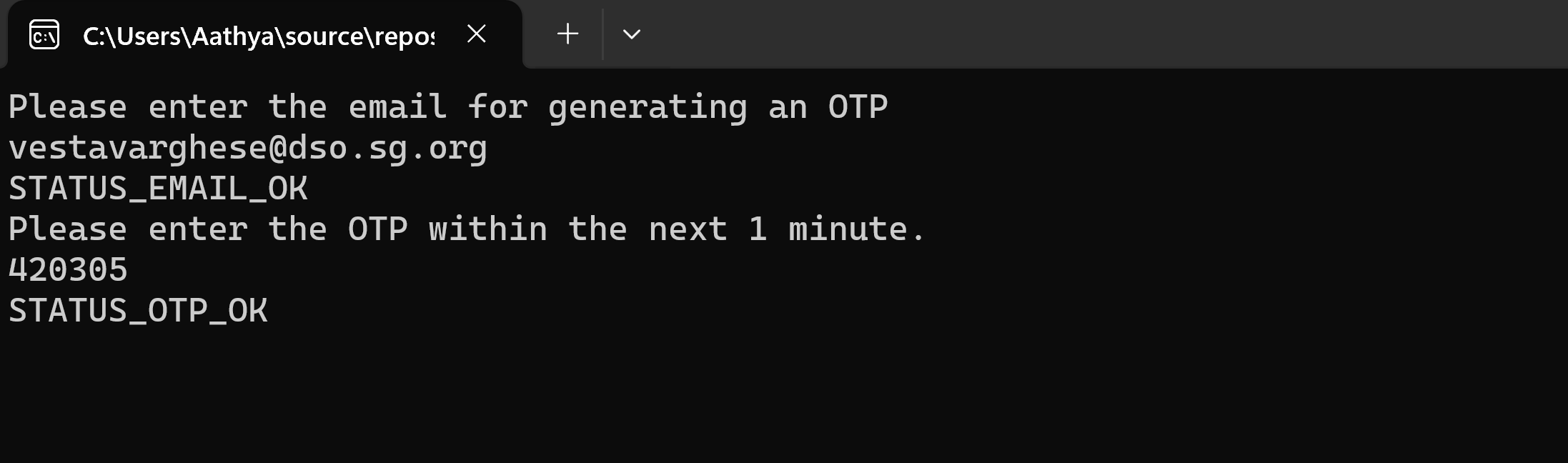
O/P: STATUS\_OTP\_MaxRetryCountReached



6) **When the email is sent successfully and validation of OTP succeeds**

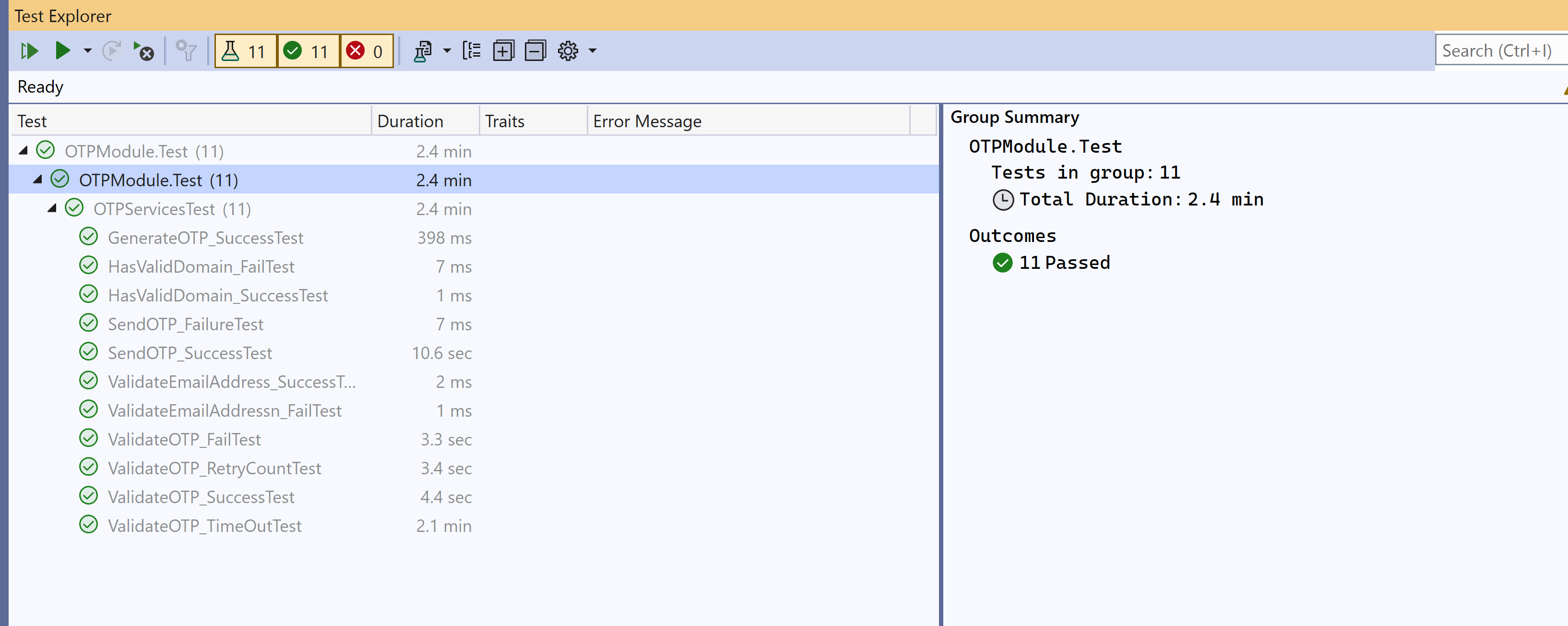
I/P: 420305

O/P: STATUS\_OTP\_OK



**Unit Testing**

Unit Testing has been implemented for all the use cases mentioned above using NUnit in the project OTPModule.Test. We can run the tests to confirm that the code works as expected.



**Project Structure Explained**

* **OTP.Console**

An application to read user inputs and show outputs to users

* **OTPModule.DataAccess**

A project for Entity Framework implementation. Code first approach is followed. And migrations are implemented to sync the code with DB.

* **OTPModule.Service**

A project which holds the logic layer.

* **OTPModule.Core**

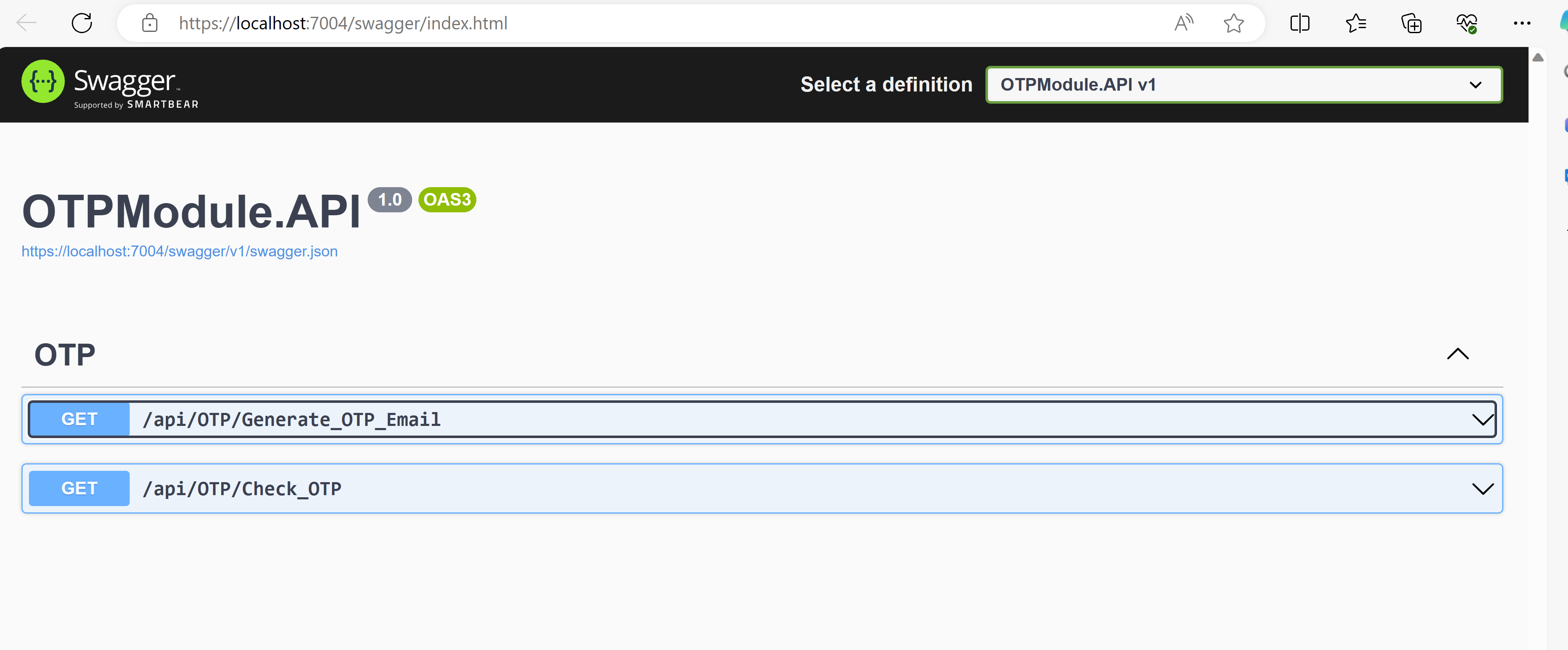
A project to define models, enums and interfaces. Separating out these in a separate project will help us in sharing it across multiple projects

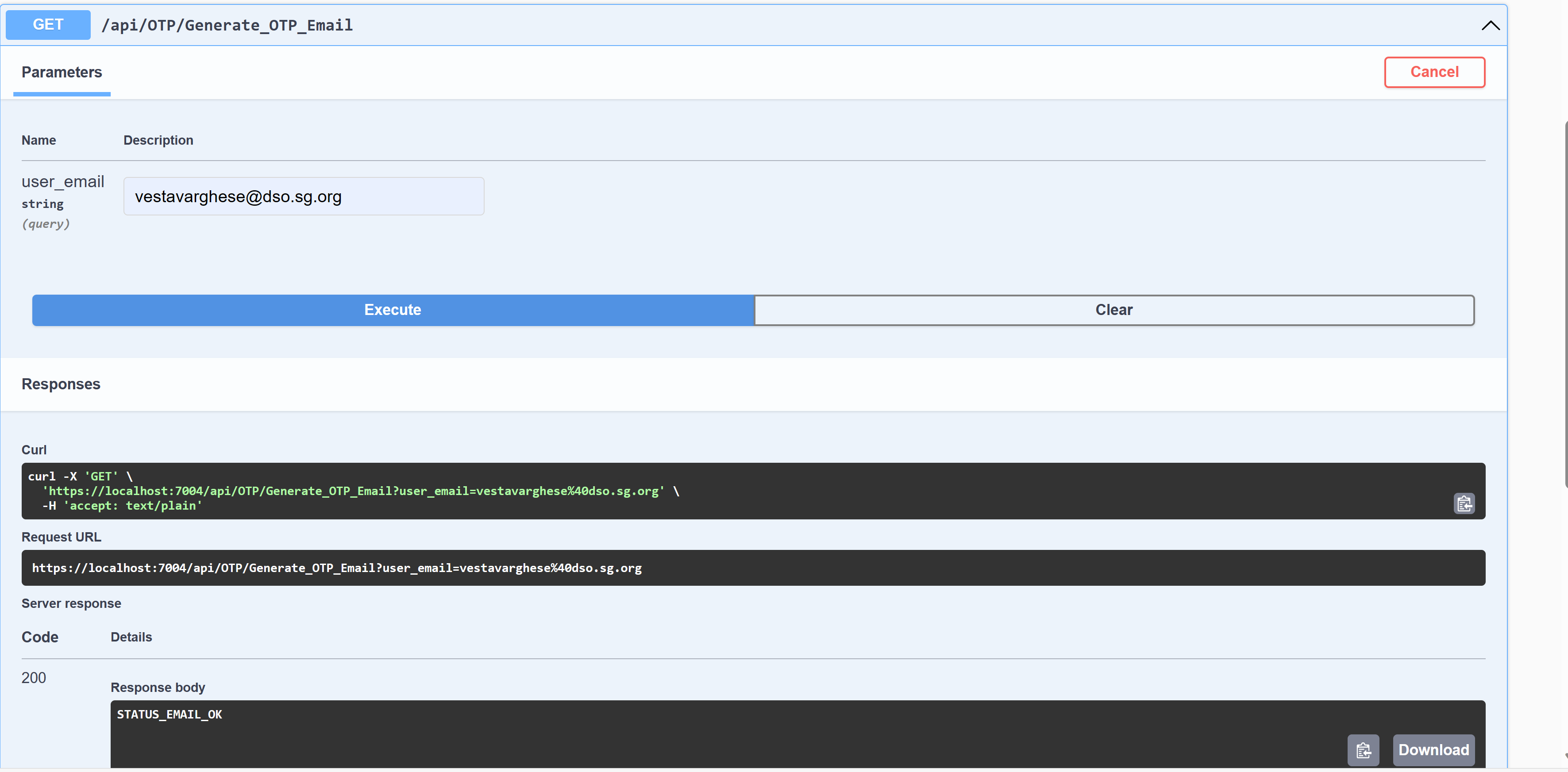
* **OTPModule.Test**

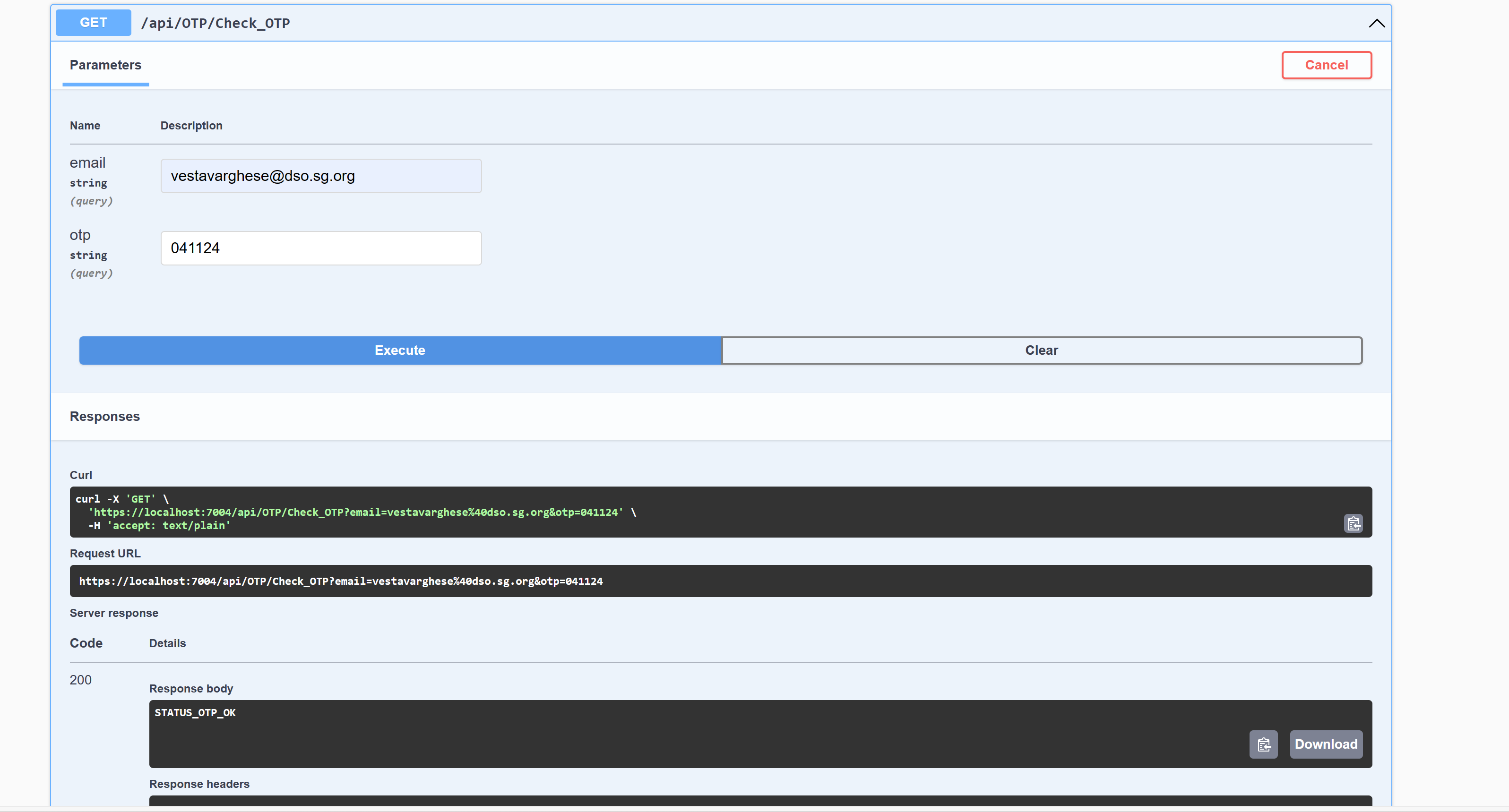
A unit test project. Test cases been implemented using NUnit.

* **OTPModule.API**

An alternative to OTP.Console. A swagger project to test out the APIs.





****

XML Documentation added for all methods in the project explaining the responsibility of the method.

**Database Diagram**

|  |  |
| --- | --- |
| **[dbo].[OTP]** | |
| **Column Name** | **Remarks** |
| Id | Primary Key |
| EmailAddress | Email address for which the OTP is generated |
| OTPCode | 6 digit OTP generated |
| RetryCount | No of times the user has tried inputting the OTP |
| CreatedDate | Date and Time at which the OTP has been generated |