**Python – Flask RESTful API with Mongo DB.**



Page Break

**Review History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Doc. Version** | **Summary of Changes** | **Initial Version** | **Review Comments** |
| **11/09/2020** | **1.0** | **Initial Authoring** | **Gopinath Udayasuriyan** |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Page Break

Python Step by Step Guidelines:

Prerequisites:

IDE: - Visual Studio Code / Atom

Database: - Mongo DB

API Tool: POSTMAN

Installing & Configuring Python

Installing & Configuring IDE  
Installing & Configuring Database

Installing & Configuring POSTMAN

Sample Test Project

Case Study:

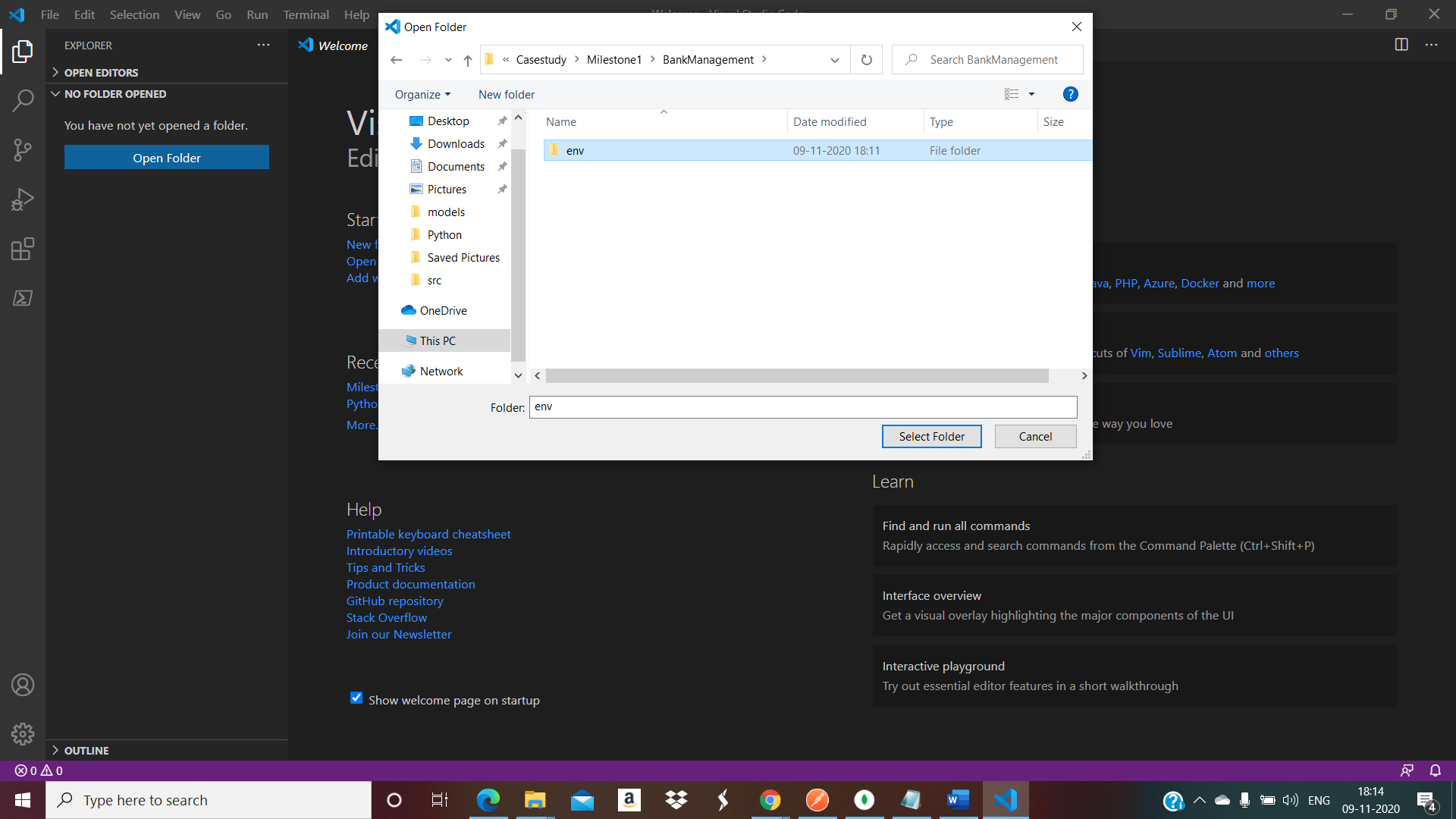
Objective

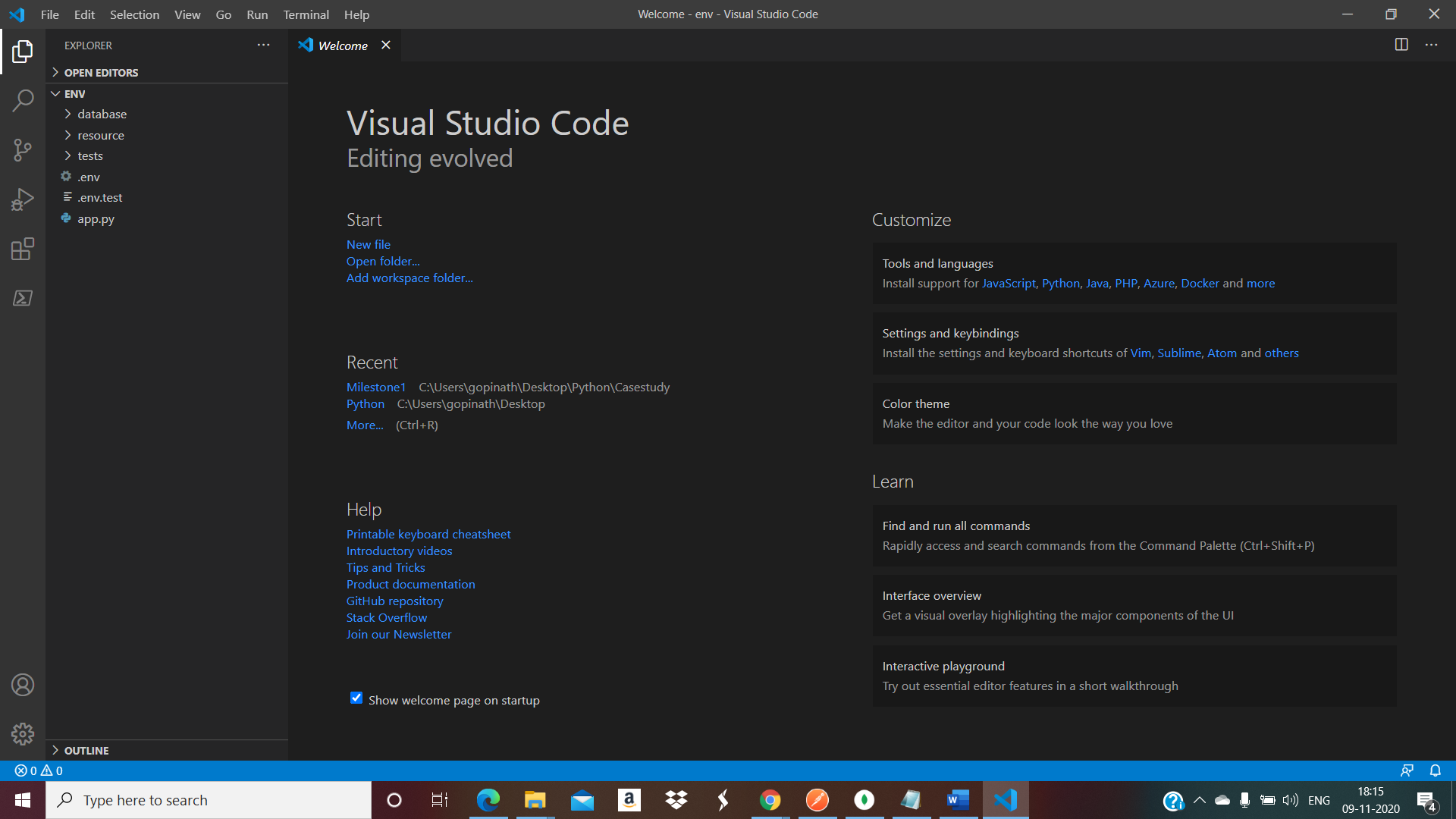
…

Configuring & Executing Case study solution

Step by Step Guidelines:

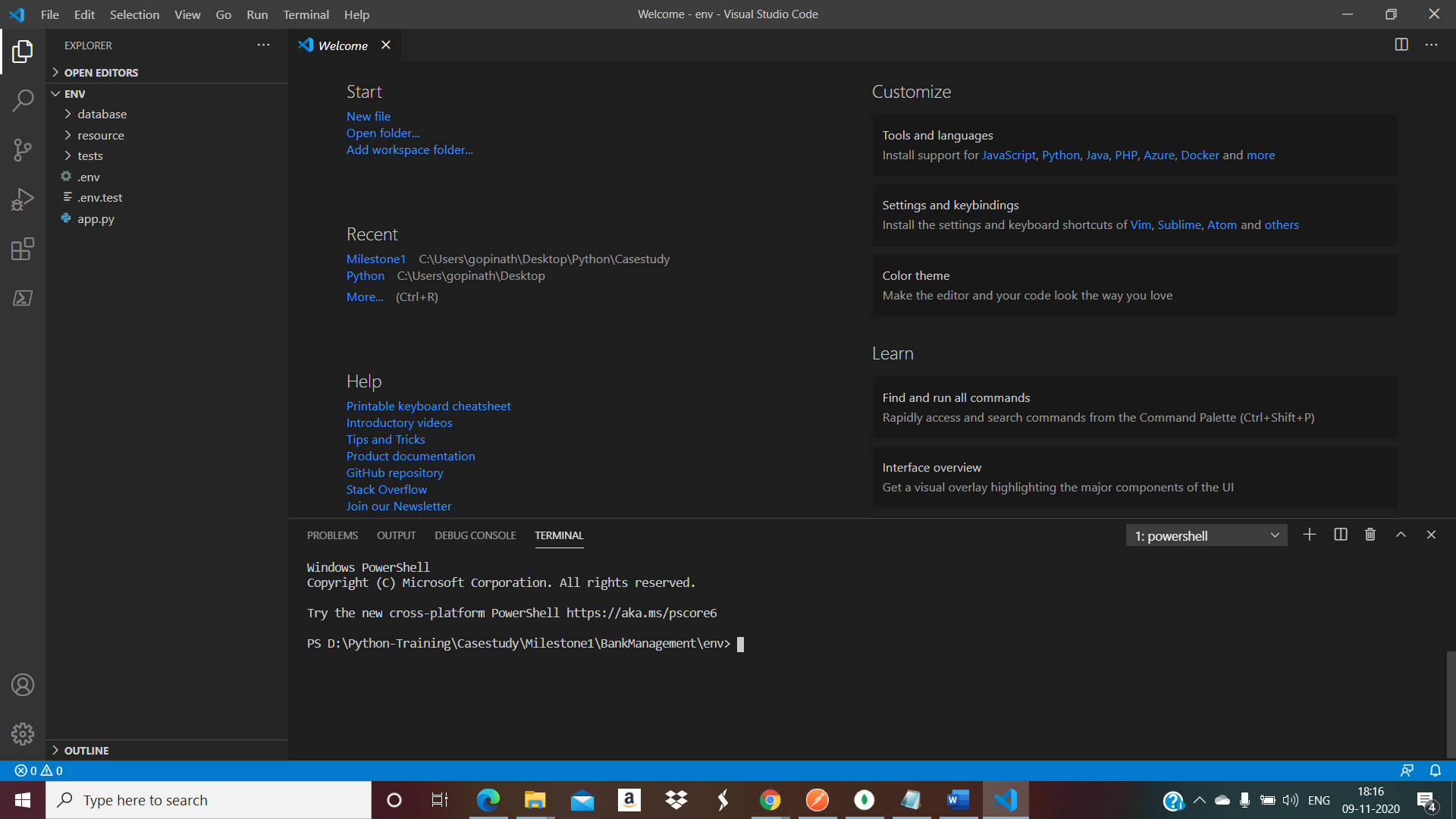
1. Open the IDE
2. Select the solution folder





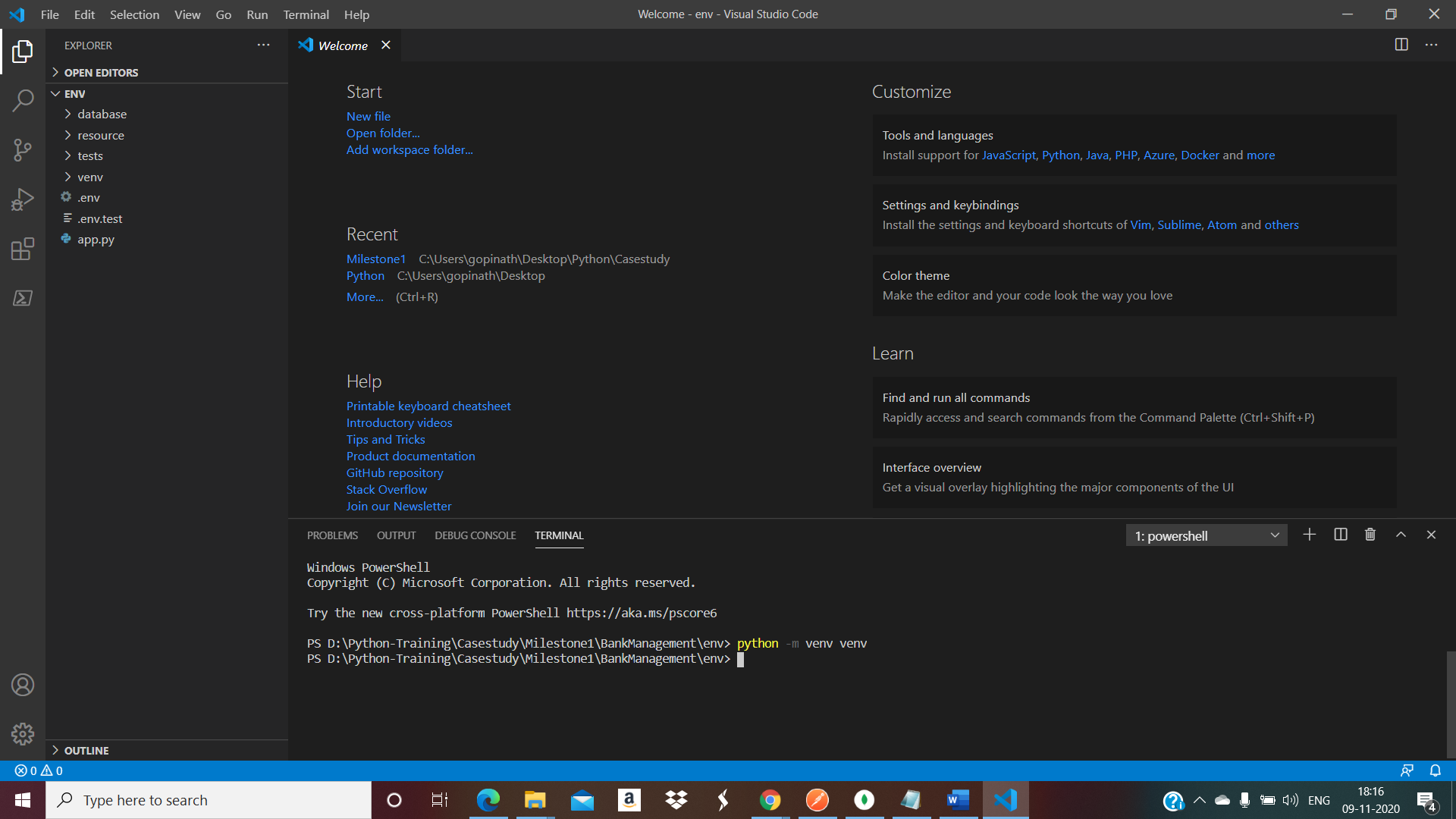
1. Commands to Configure the python dev environment

Open the Terminal to execute the commands



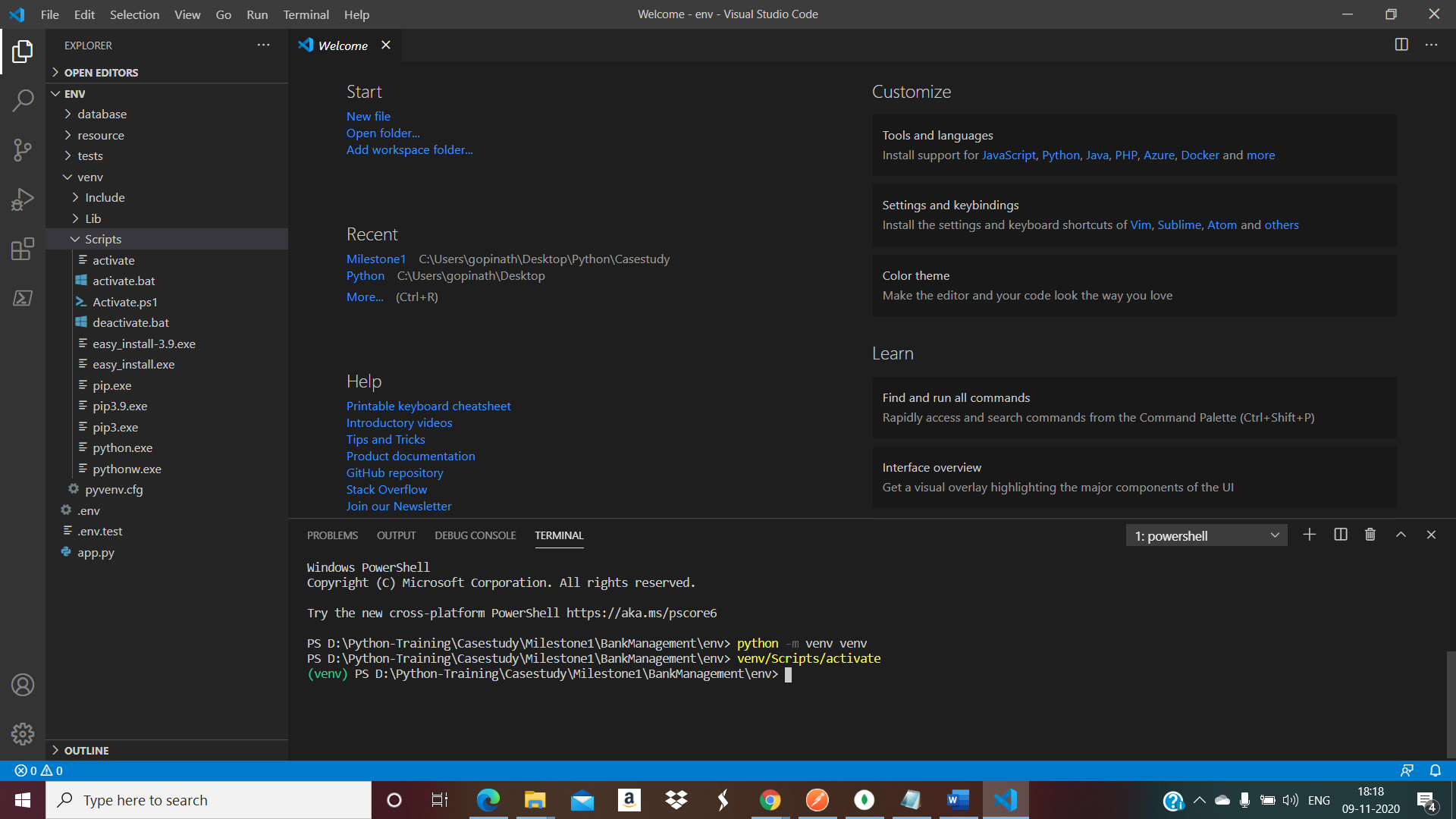
1. Configure virtual dev environment

**python -m venv venv**



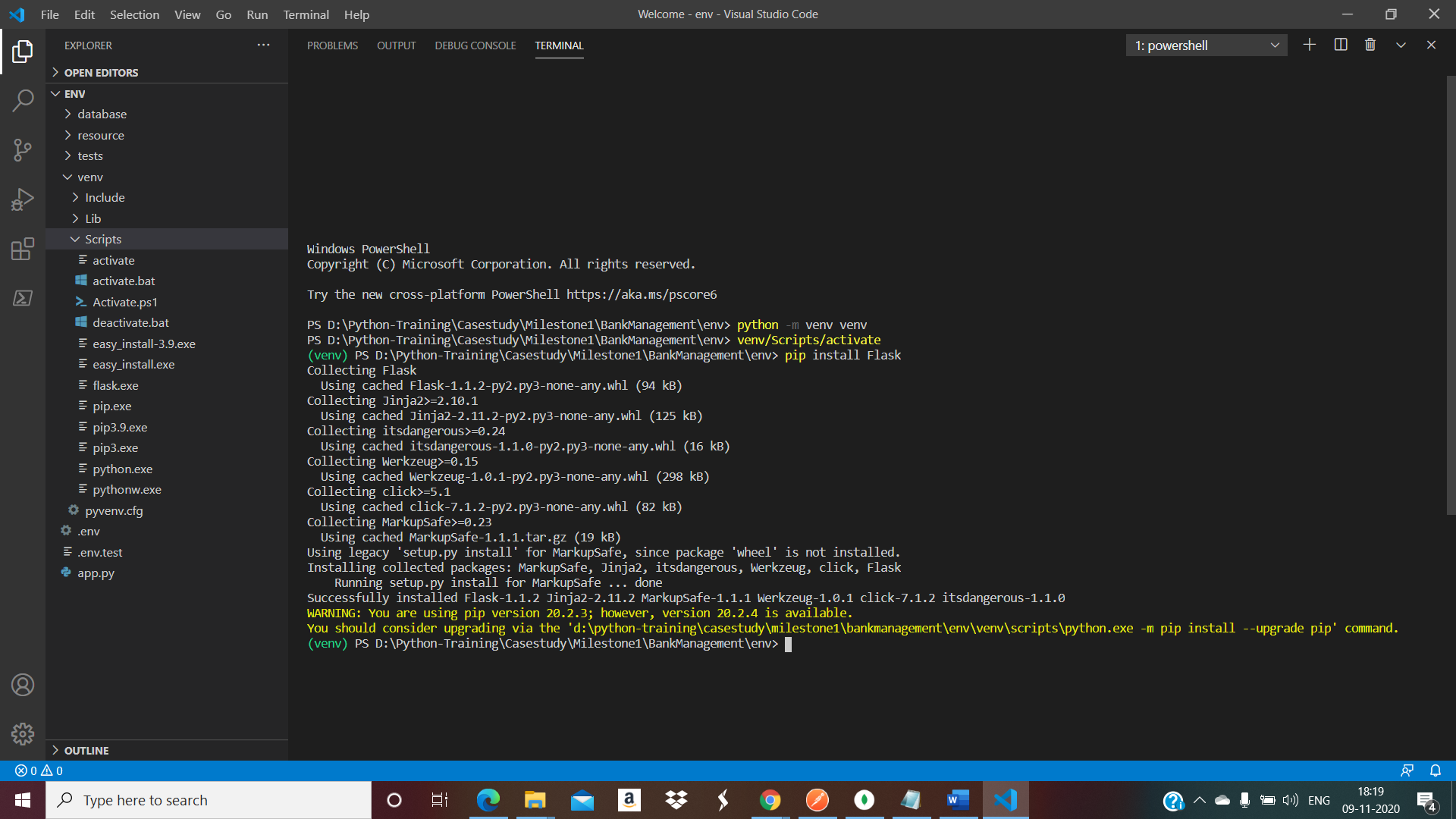
1. Activating the scripts

**venv/Scripts/activate**



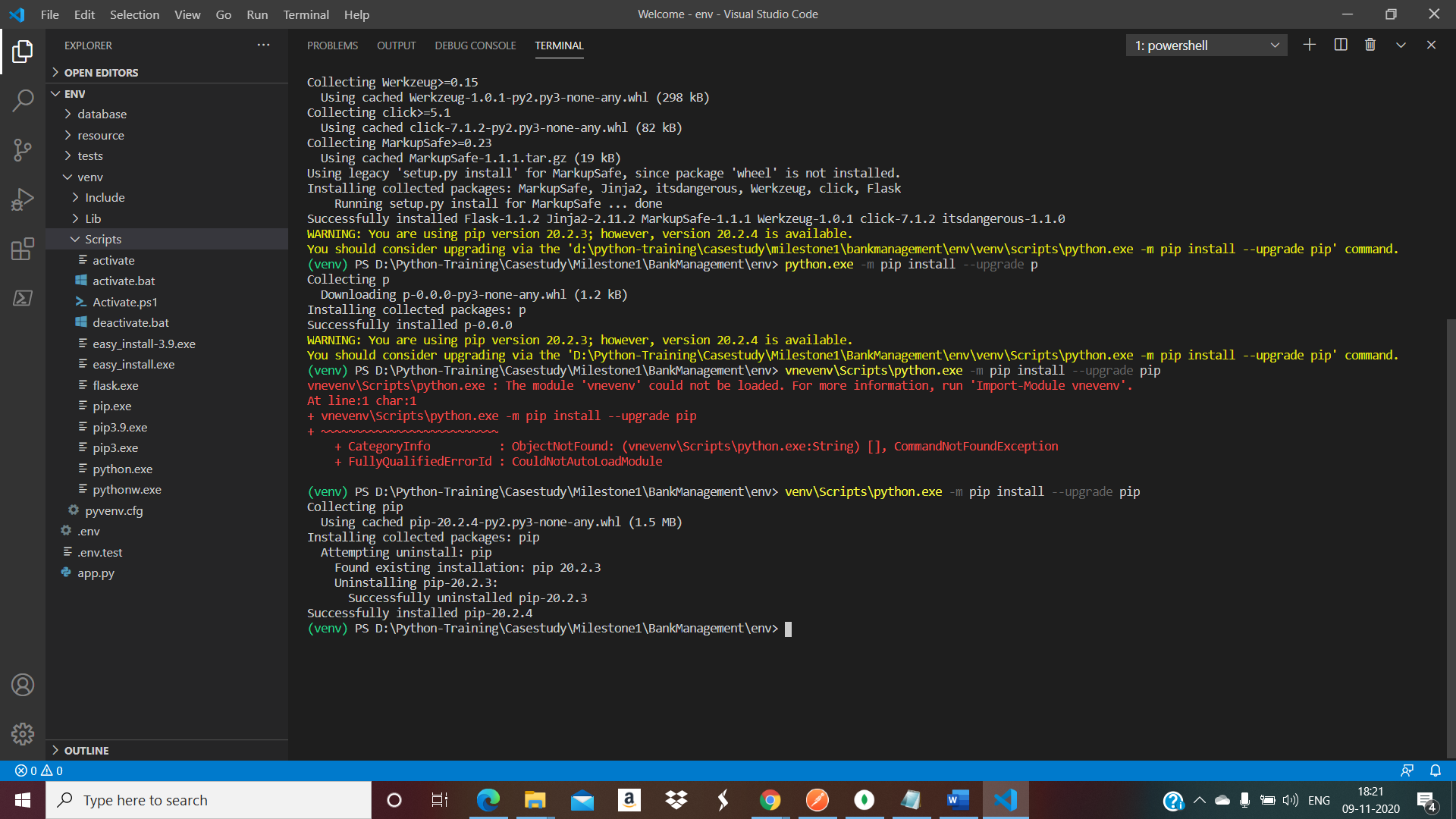
1. Installing pip flask

**pip install Flask**



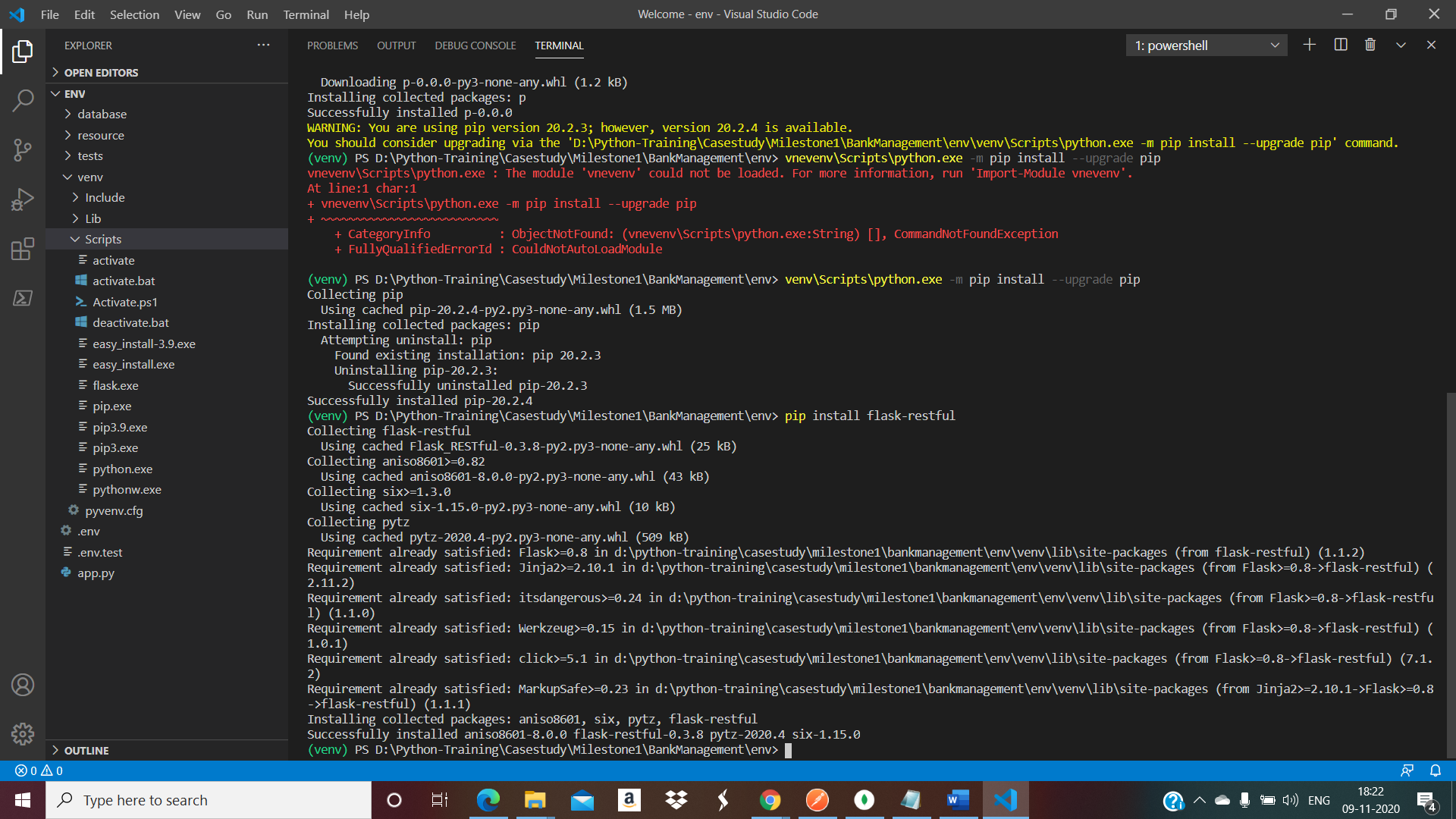
1. Upgrade the python

**python.exe -m pip install --upgrade pip**



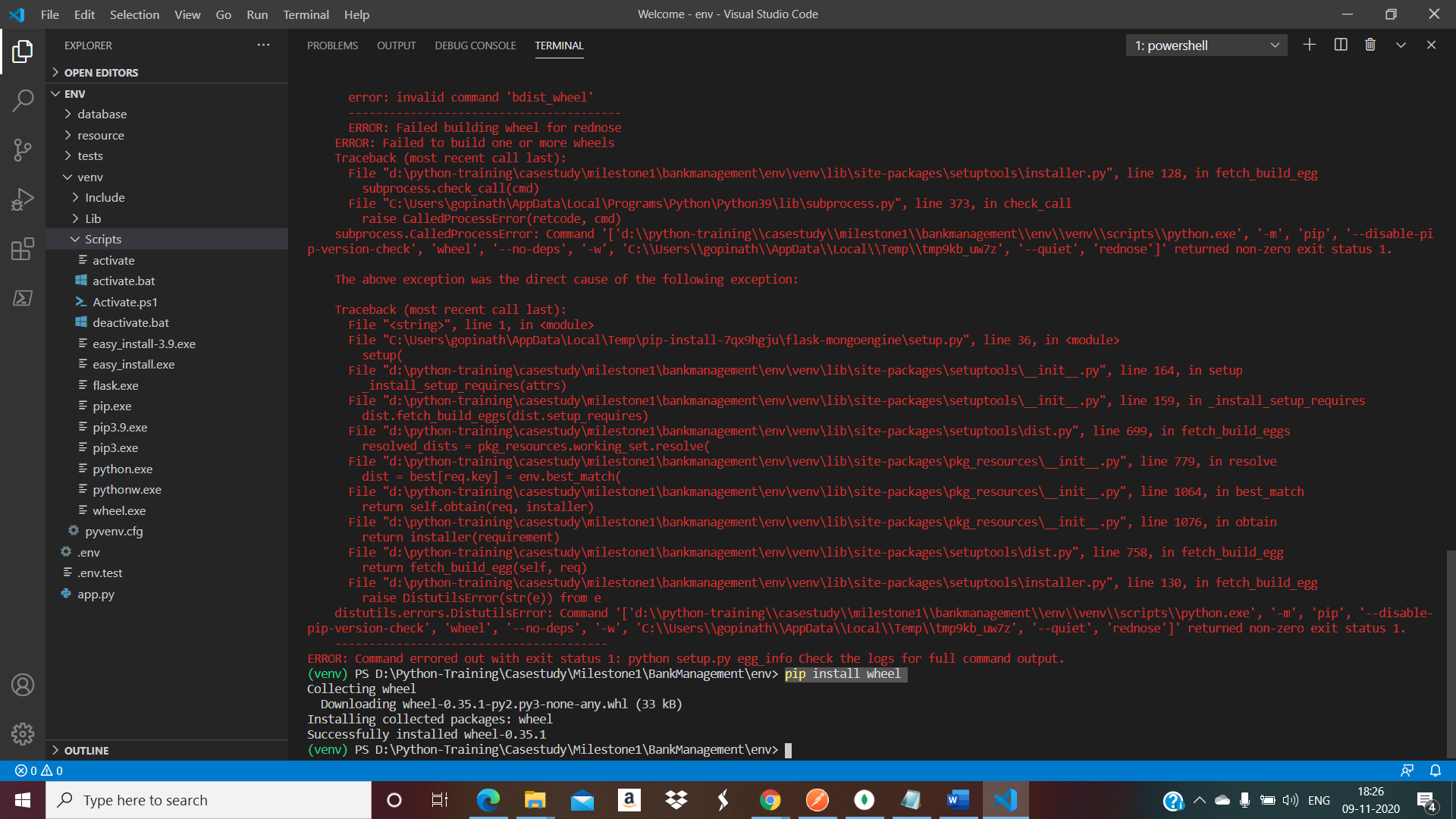
1. Installing pip flask RESTful

**pip install flask-restful**



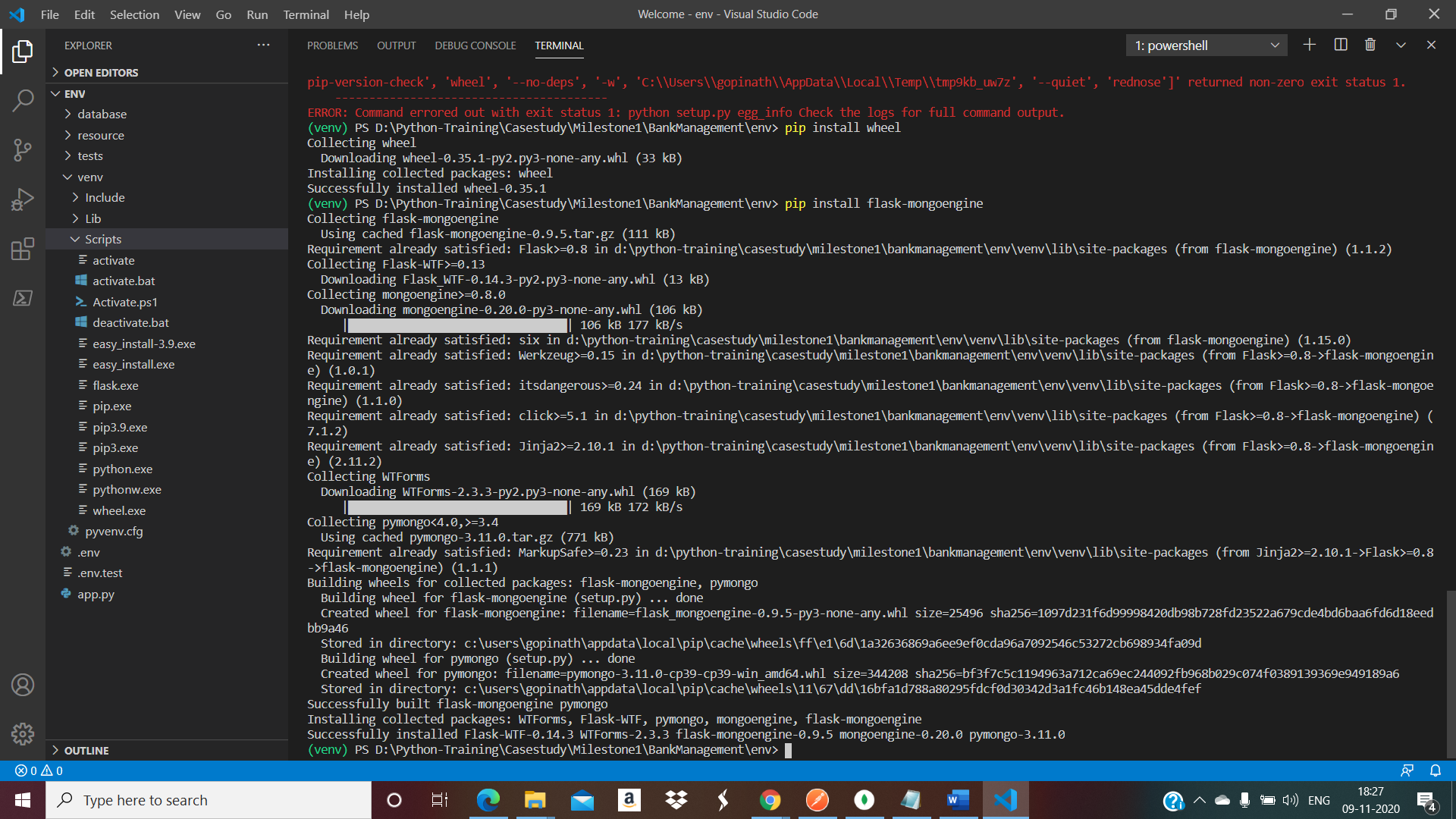
1. Installing pip wheel

**pip install wheel**



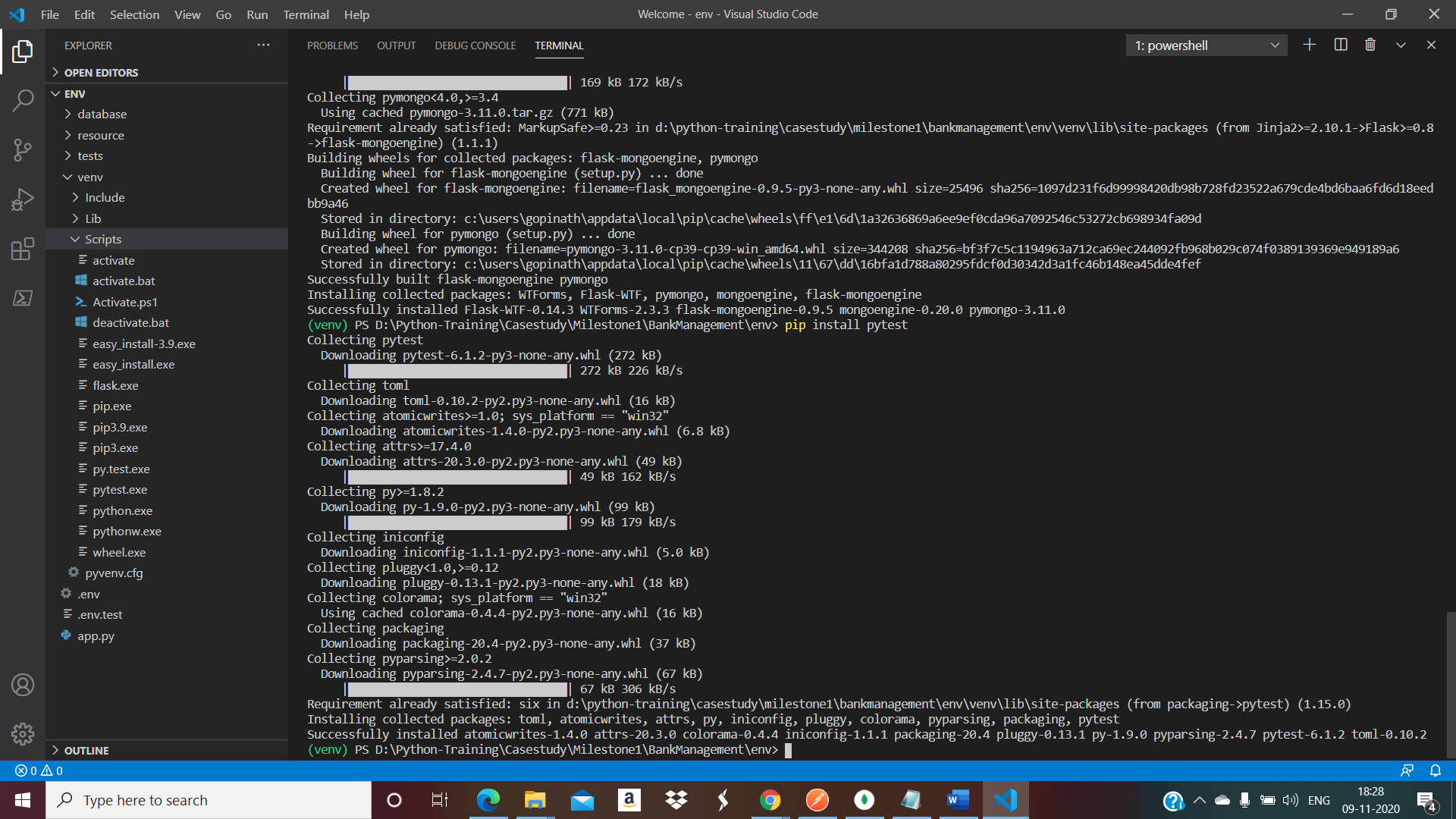
1. Installing pip flask mongoengine

**pip install flask-mongoengine**



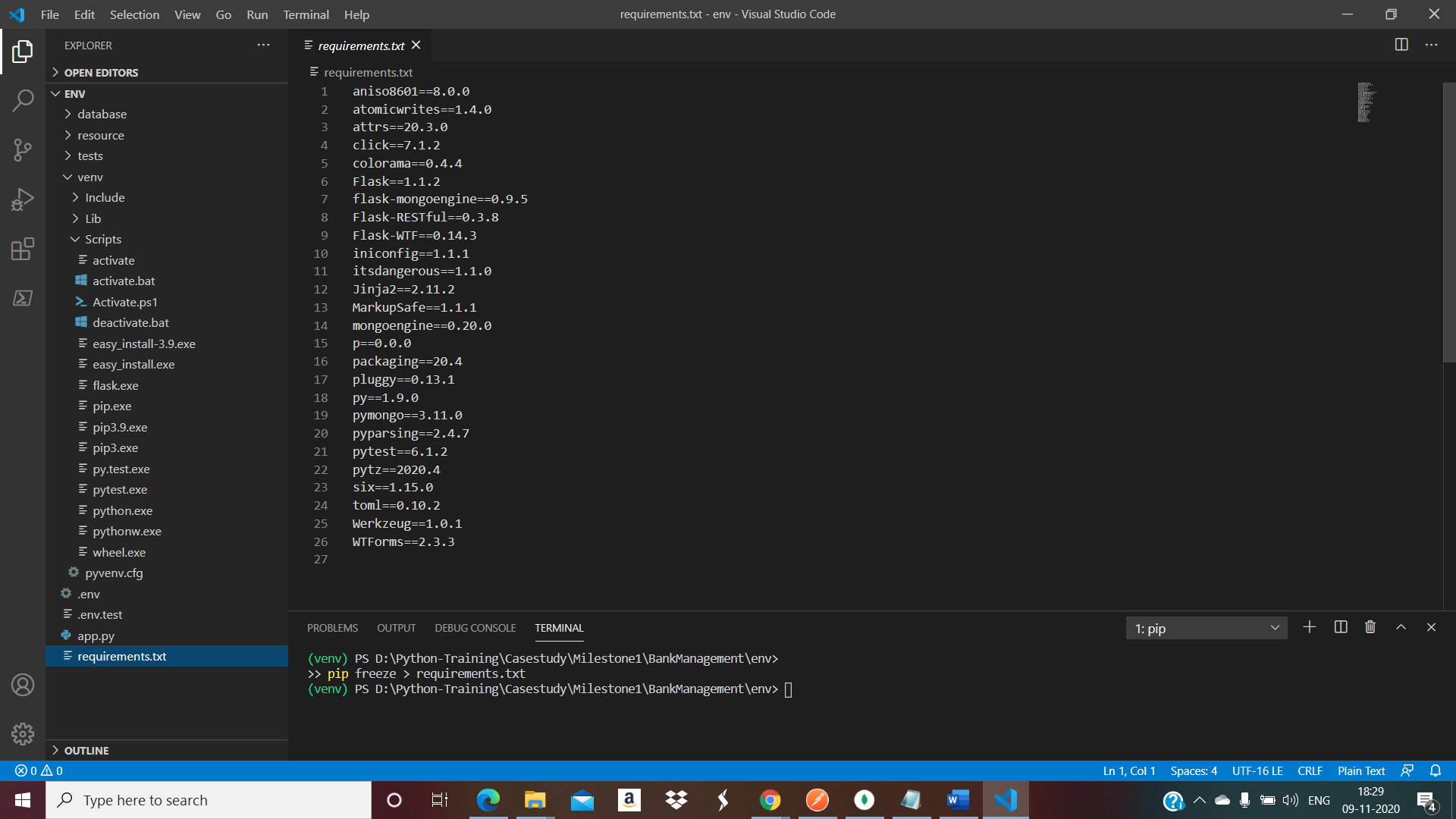
1. Installing pytest

**pip install pytest**



1. Execute Freeze Requirement to verify

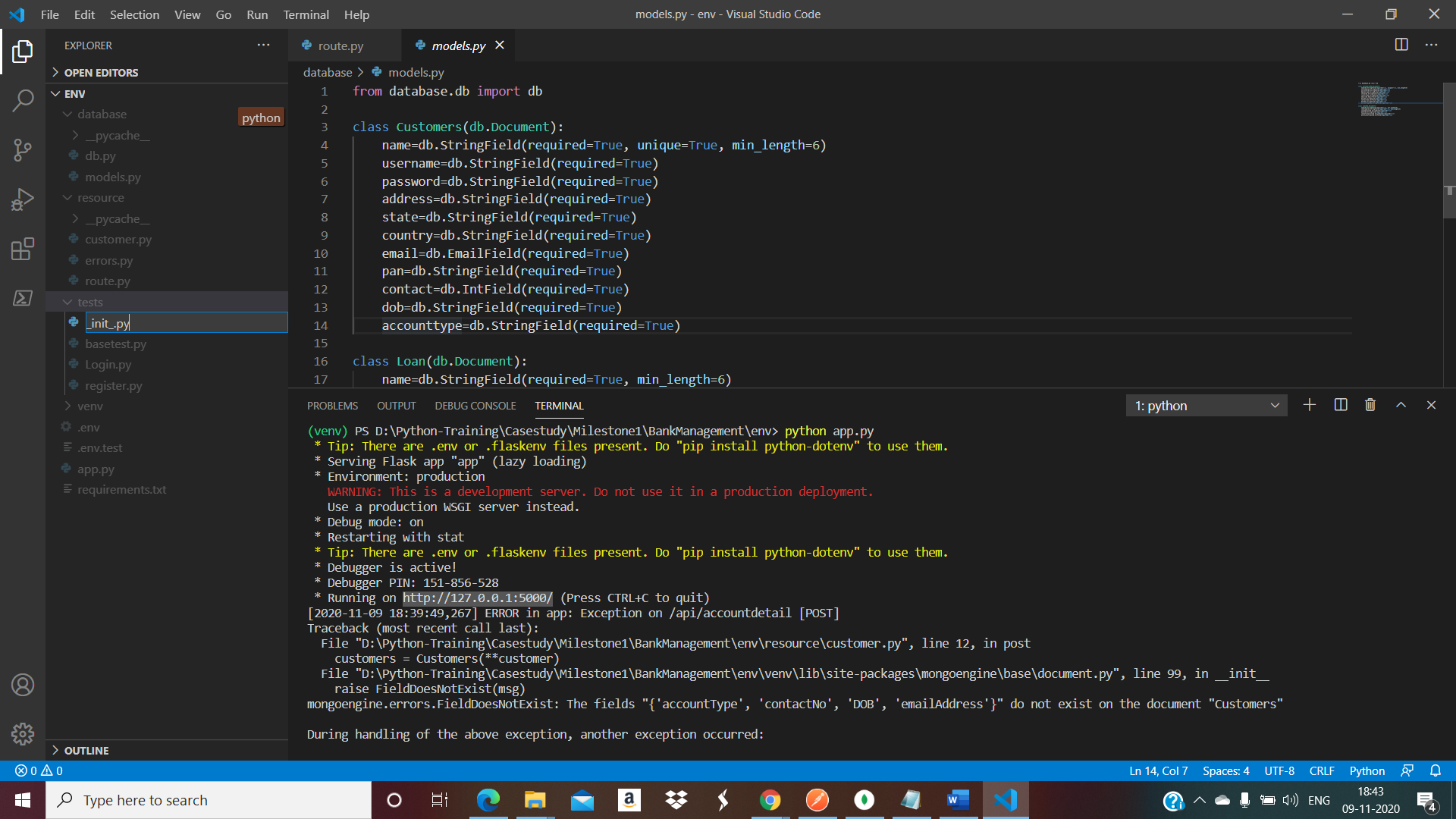
**pip freeze > requirements.txt**



Adding initiation python files:

Add \_init\_.py file to the child folders available. Here we need to add in the below folder

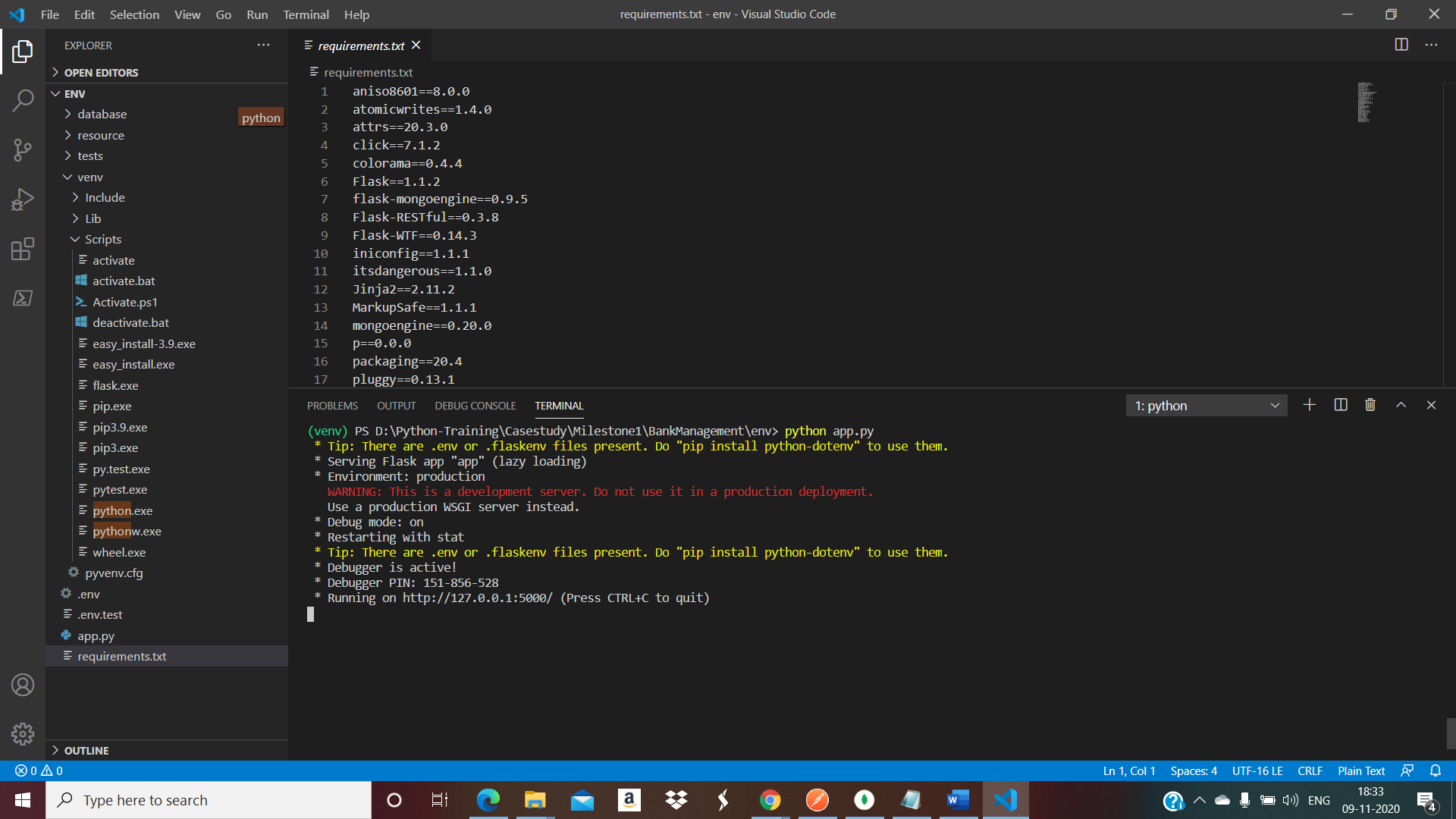
1. resource
2. test
3. database



Run the solution:

In terminal, point to the solution root folder. Execute the application start-up python file. Below the app.py is the start-up file for the solution.

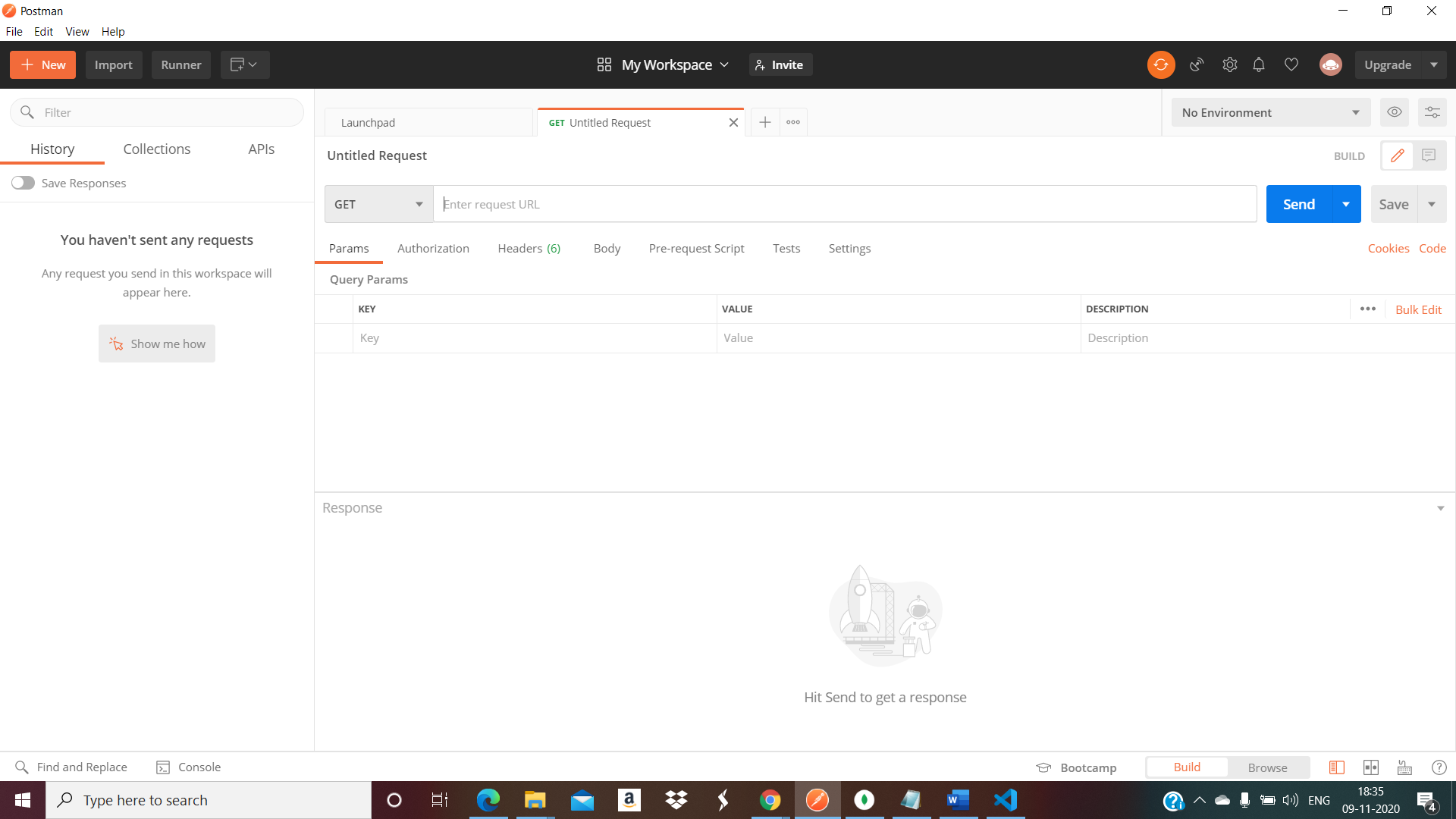
**python app.py**



RESTApi is built successfully, up and running on the[**http://127.0.0.1:5000/**](http://127.0.0.1:5000/)

**Test the API endpoints using POSTMAN.**

1. **Open the POSTMAN.**



**Testing POST methods.**

1. **Create Customer Account Details**

**Api Url:** <http://127.0.0.1:5000/api/accountdetail>

**Method Type:** POST

**Input JSON Payload:**

{

    "name":"daniel",

    "username":"daniel",

    "password":"daniel",

    "address":"address",

    "state":"tn",

    "country":"india",

    "email":"daniel@gmail.com",

    "pan":"7834287",

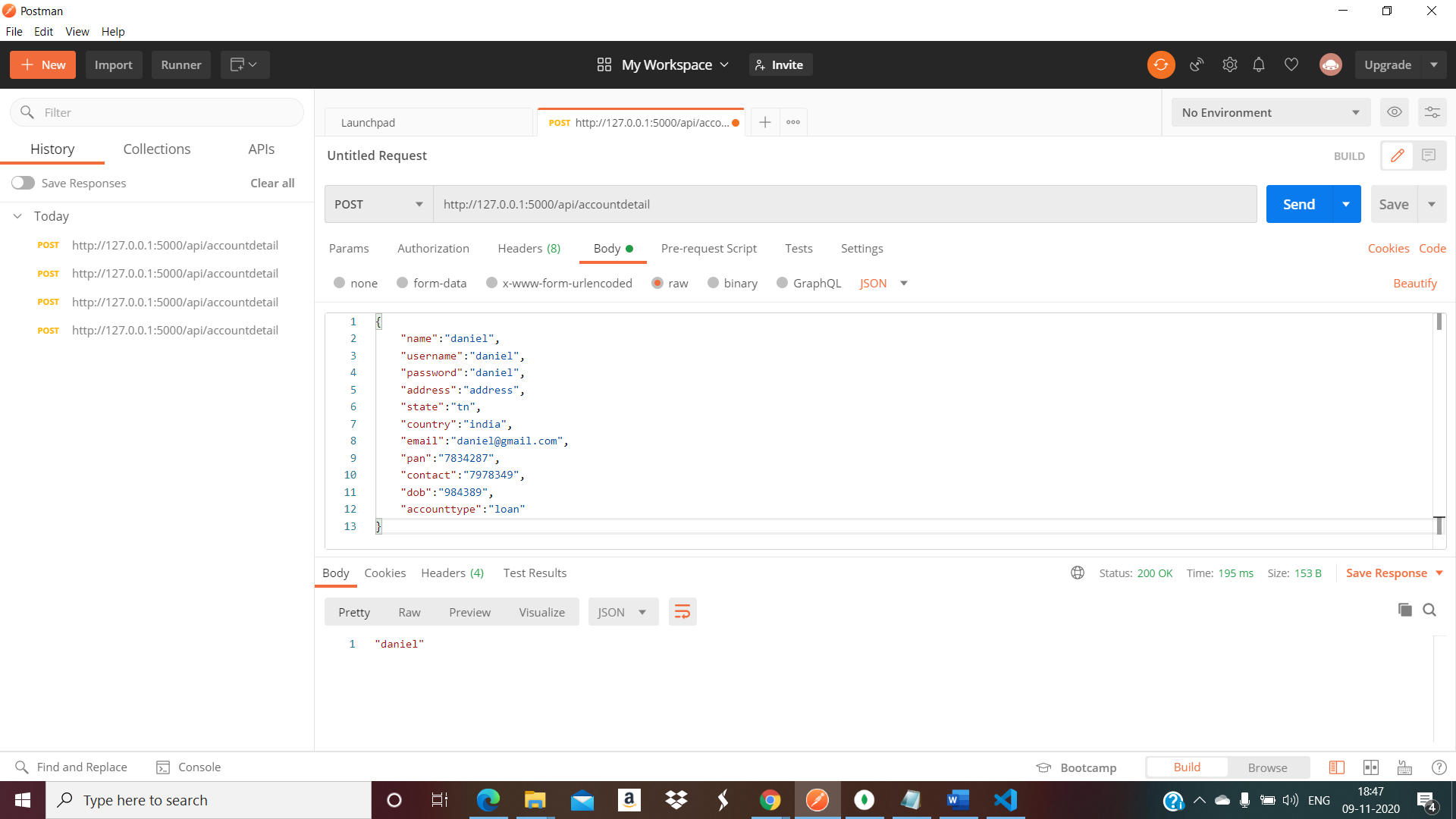
    "contact":"7978349",

    "dob":"984389",

    "accounttype":"loan"

}

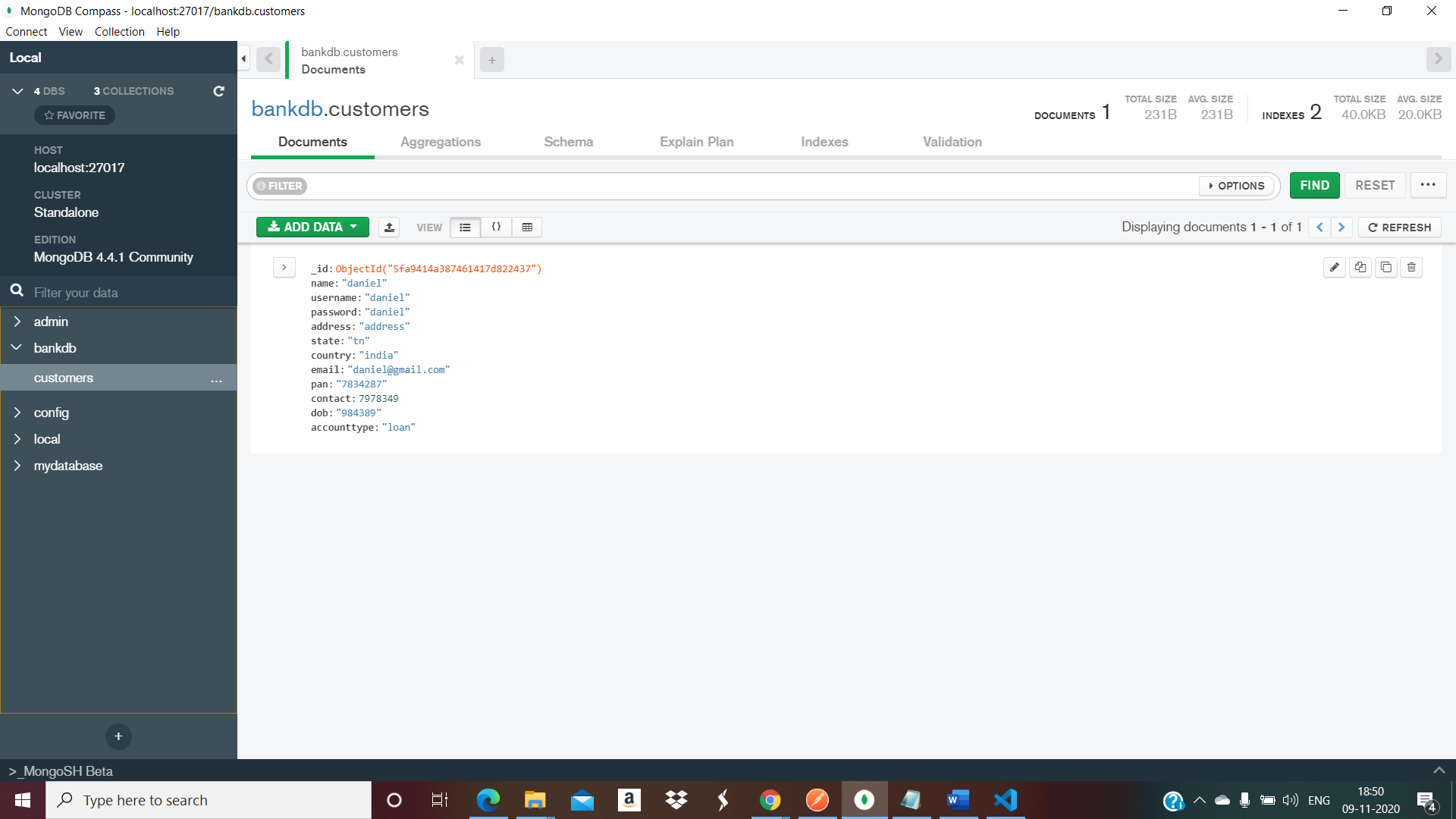
**Output:**



**New Database Create & Collection Created on Mongo DB:**

1. **Database: bankdb**

**Collection : customers**



1. **Create/Apply Loan for Customer**

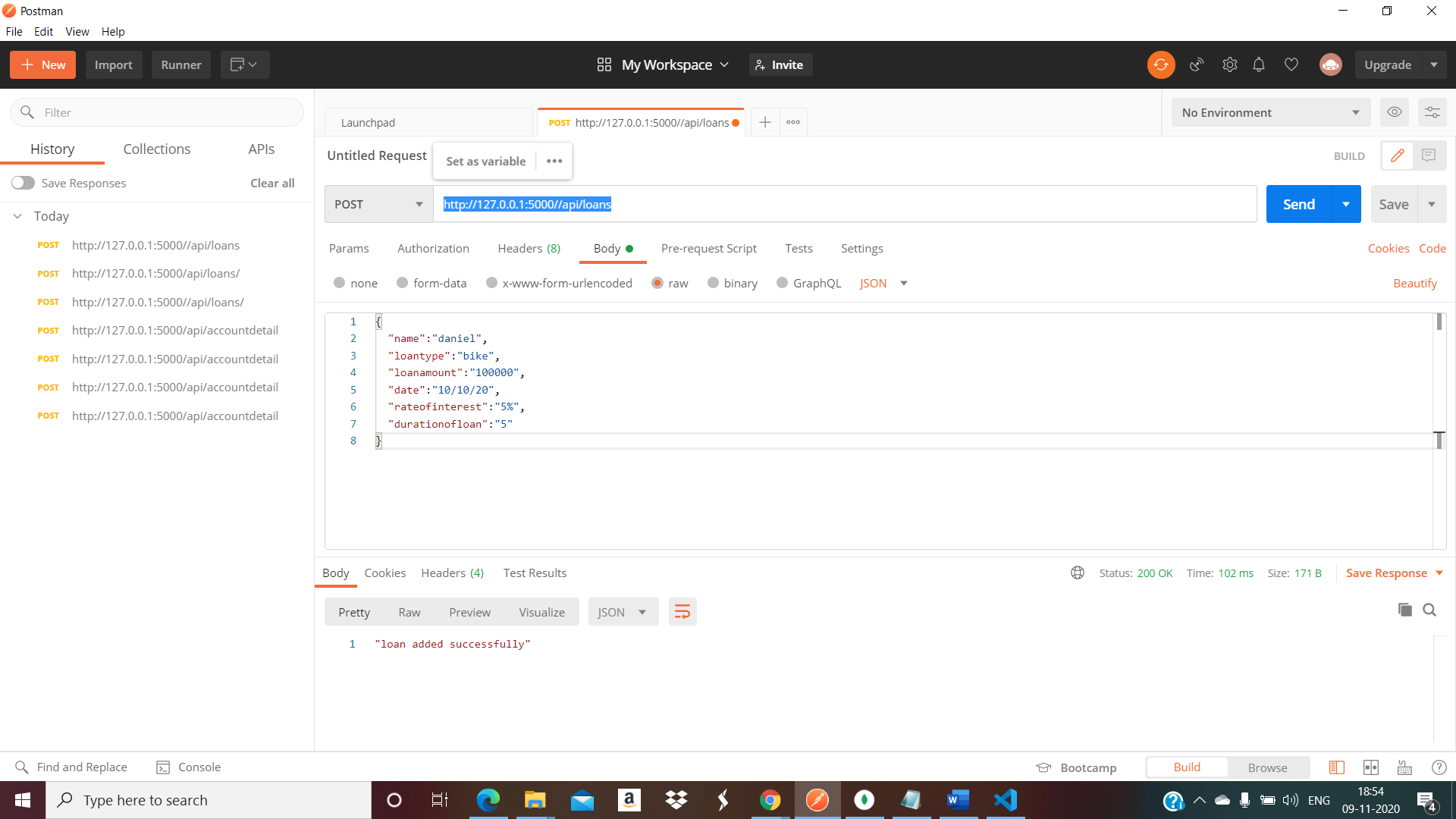
**Api Url:** <http://127.0.0.1:5000//api/loans>

**Method Type:** POST

**Input JSON Payload:**

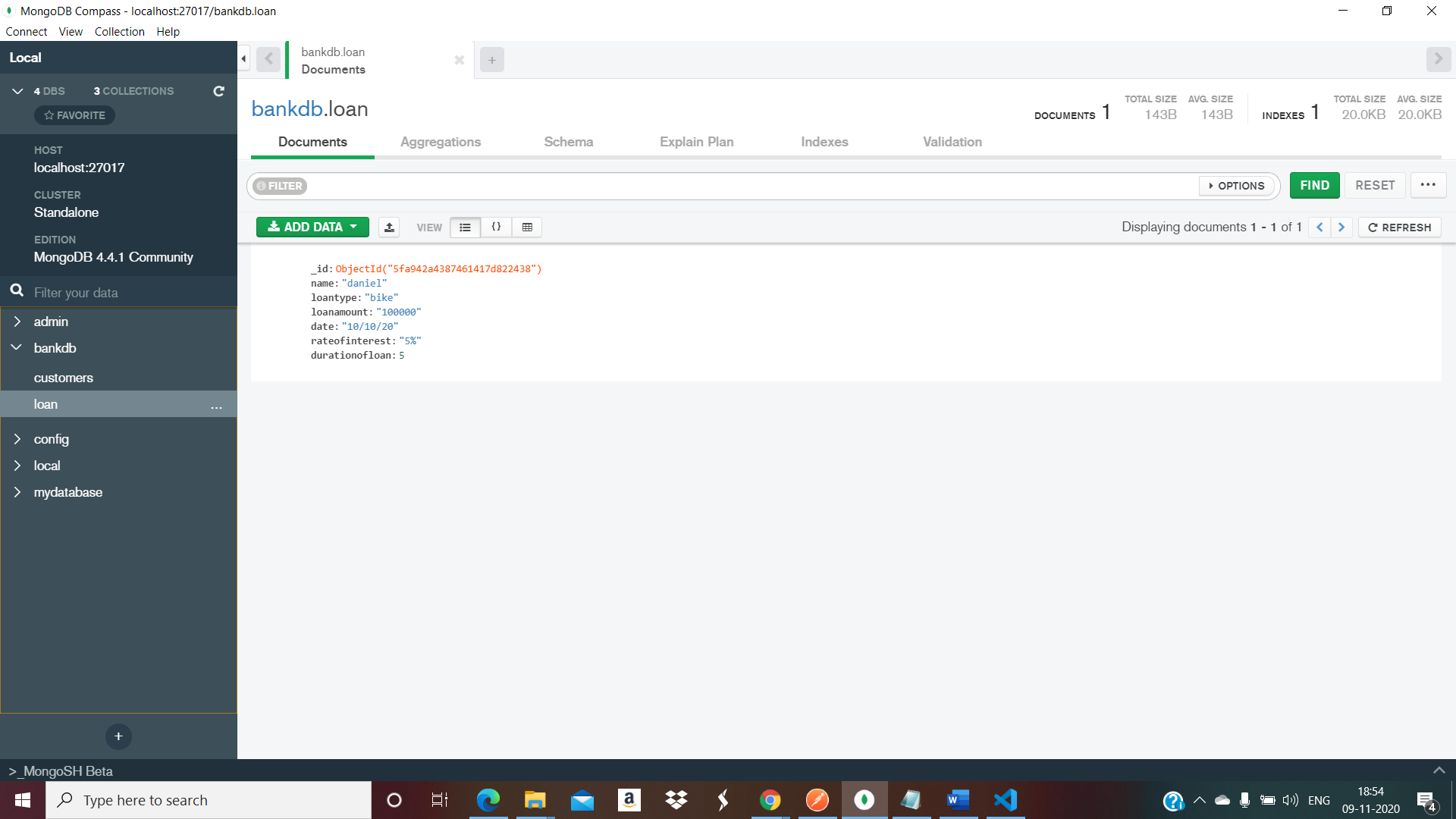
|  |
| --- |
| { |
|  | "name":"daniel", |
|  | "loantype":"bike", |
|  | "loanamount":"100000", |
|  | "date":"10/10/20", |
|  | "rateofinterest":"5%", |
|  | "durationofloan":"5" |
|  | } |

**Output:**



**New Collection Created to the existing Database on Mongo DB:**

**Collection: loan**



**Updating the Collection information**

**Update Customer Details**

**API url:** <http://127.0.0.1:5000/api/accountdetail/daniel>

**Method Type: PUT**

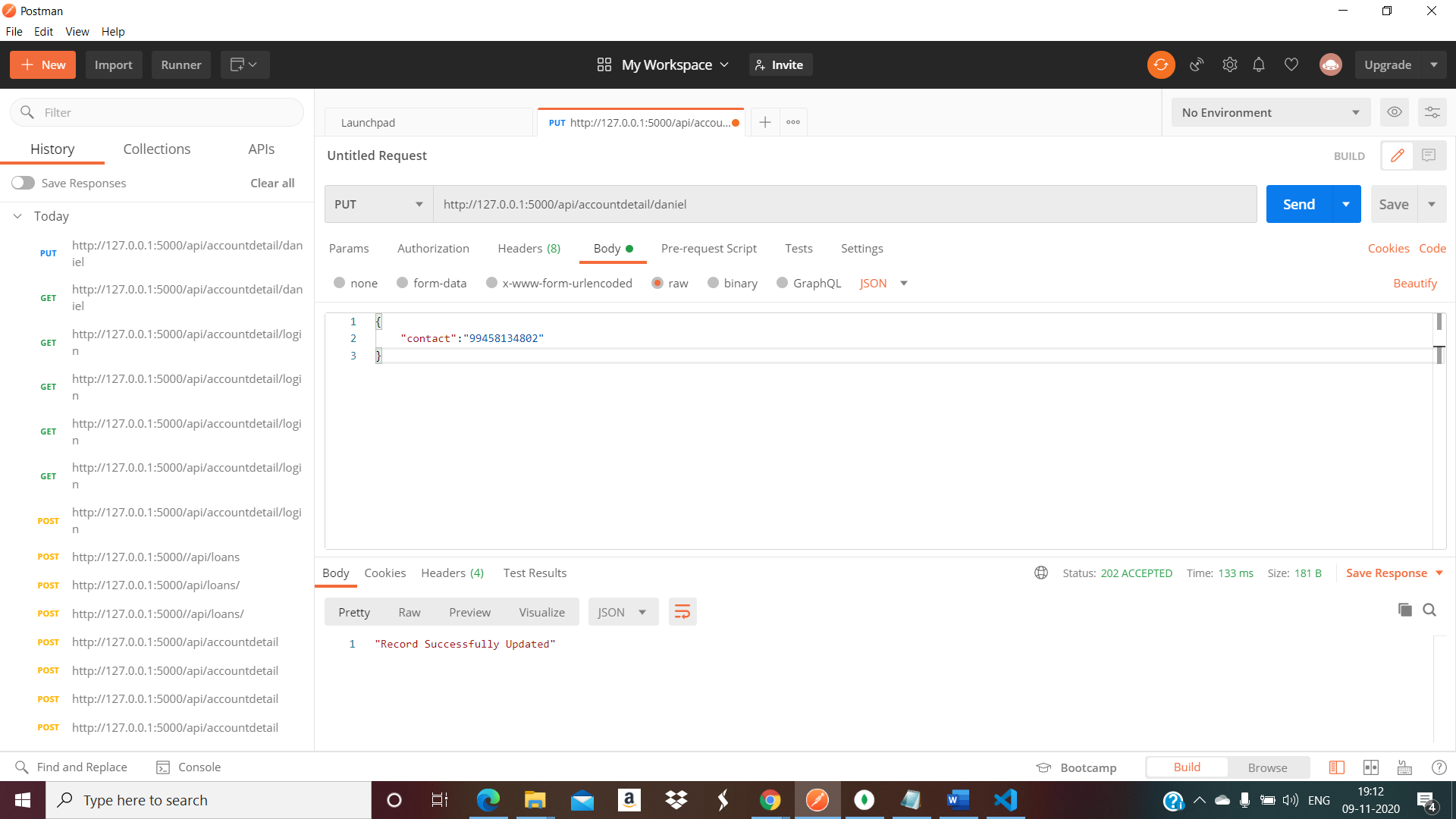
**Input Payload:**

{

    "contact":"99458134802"

}

**Output**



**Retrieving the Collection details**

**Get the Account details for selected customer**

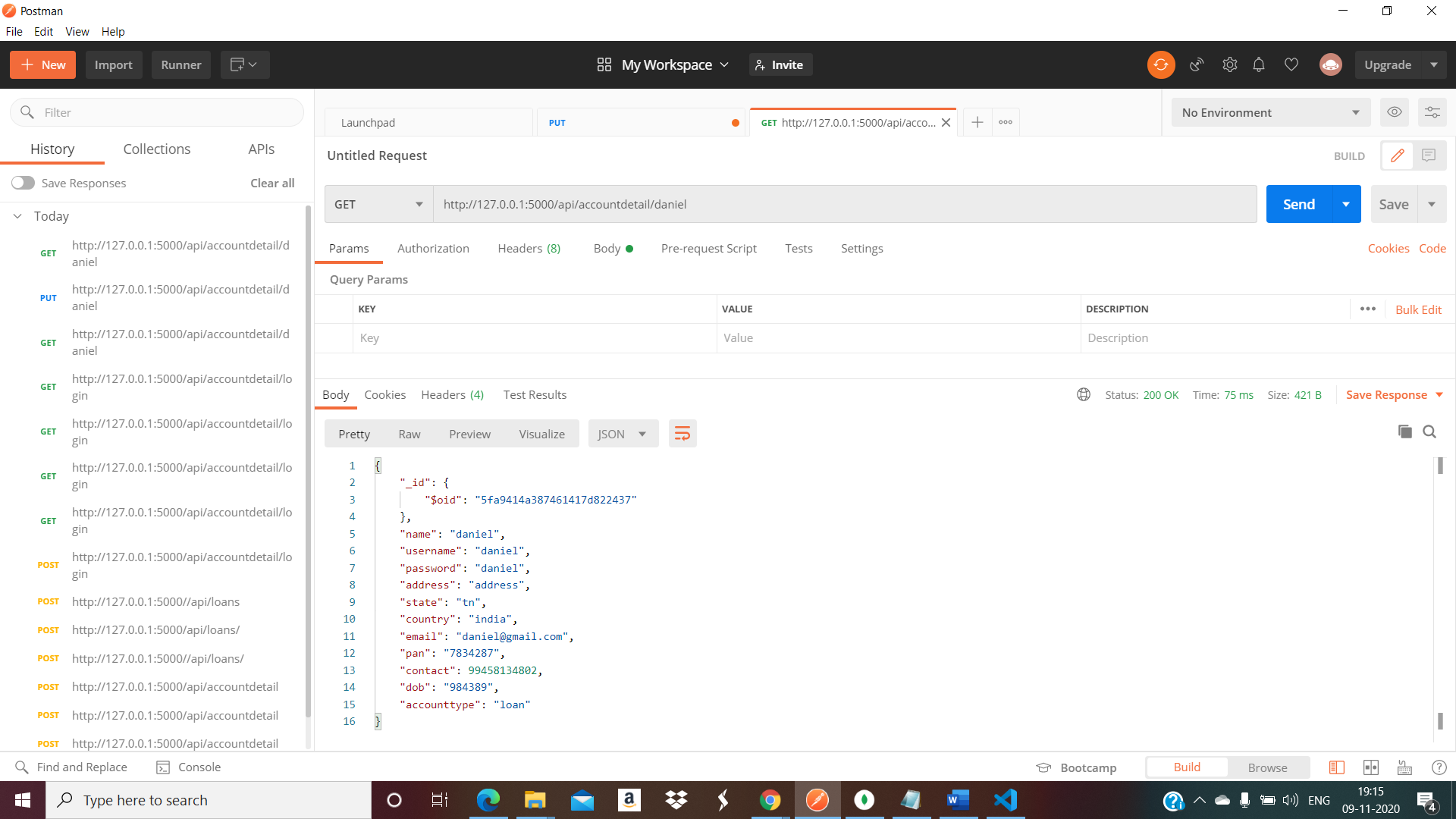
**API url:** <http://127.0.0.1:5000/api/accountdetail/daniel>

**Method Type: GET**

**Input: url parameter**

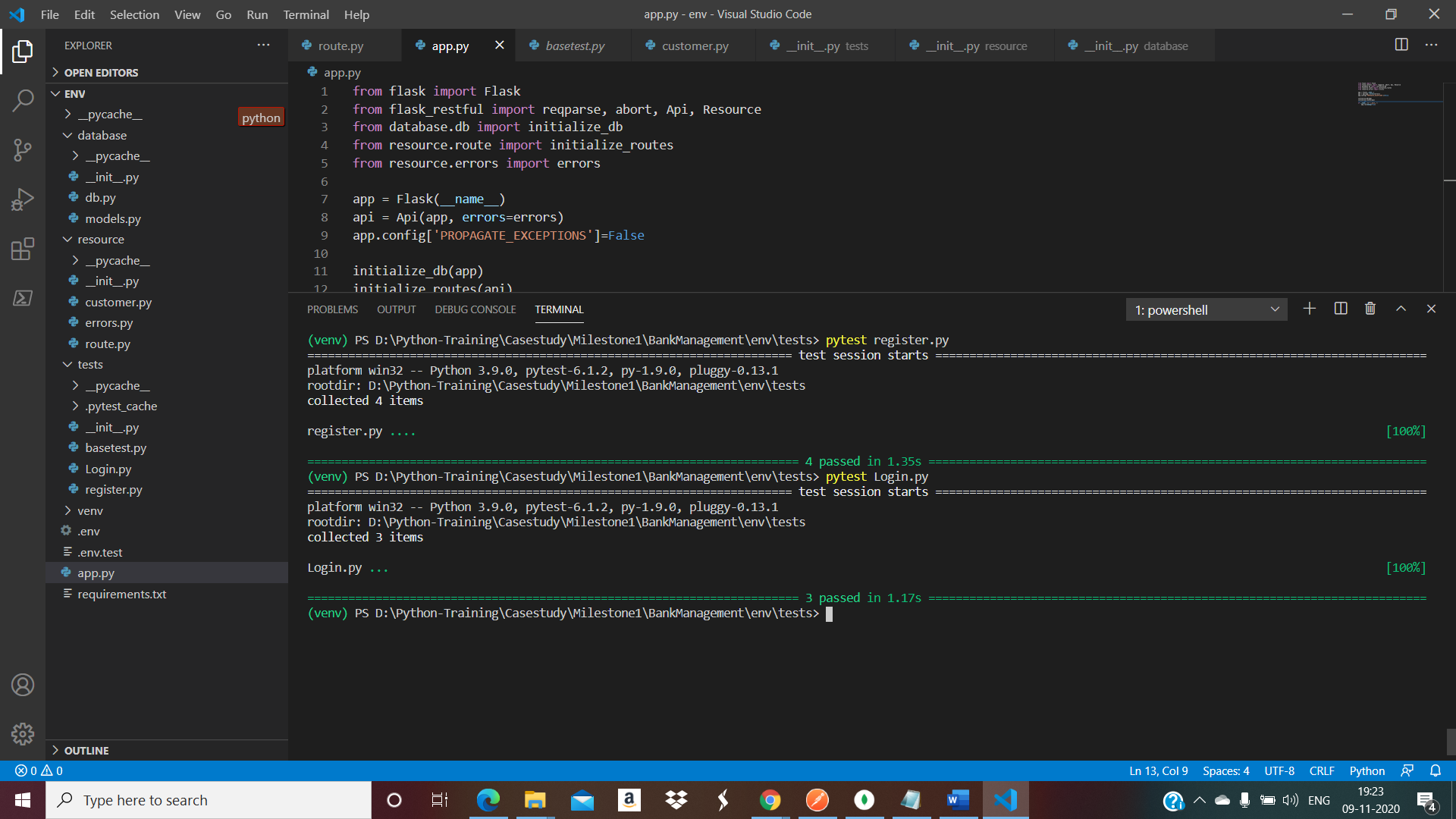
daniel

**Output:**



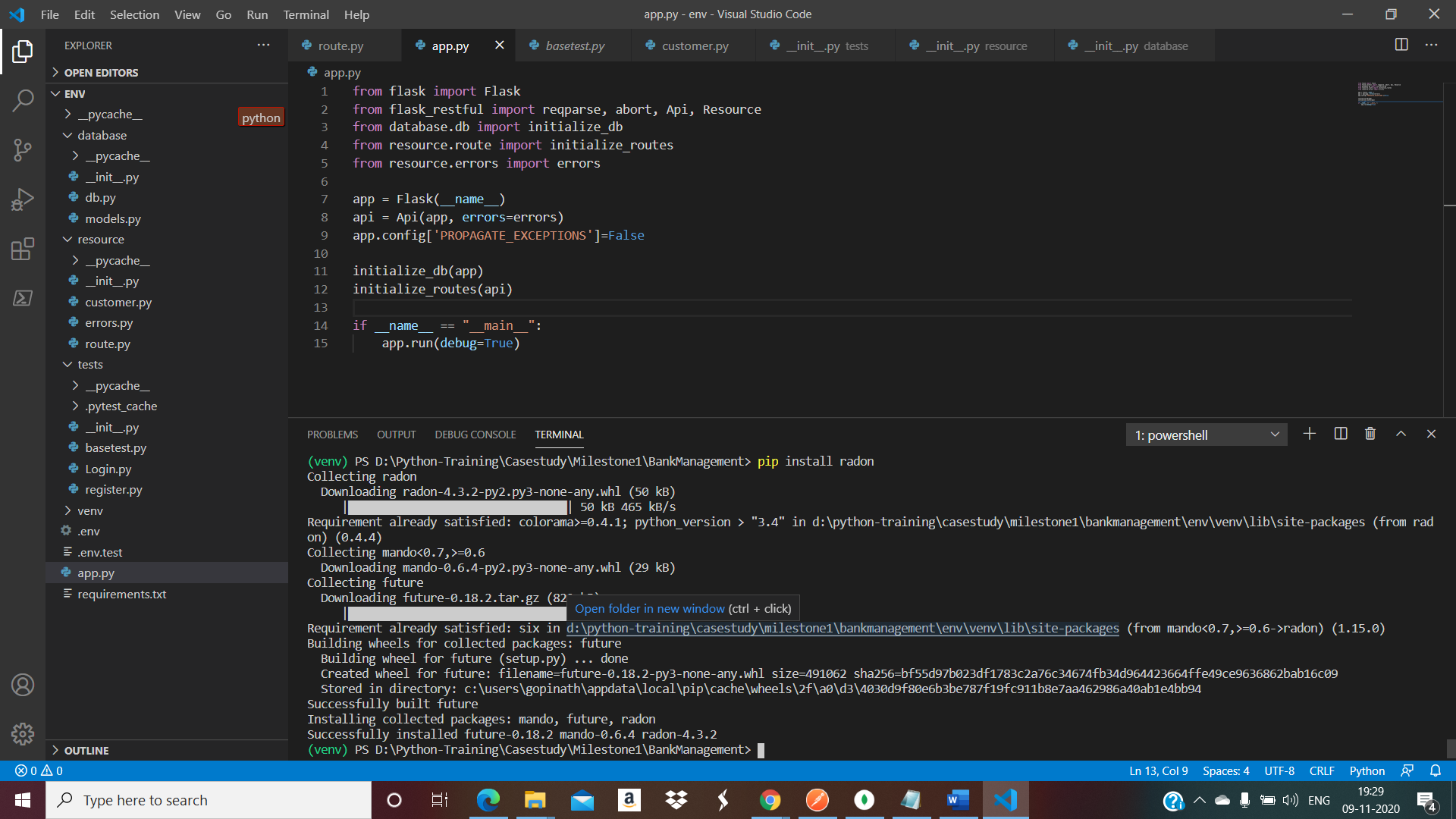
**Test Case Execution:**

1. **Point to the test folder**
2. **Remove the reference of the database from app.py**
3. **Execute the Login.py test cases & register.py test cases.**



**Code Coverage & Code Quality Analysis:**

**Install pip radon**



**Complexity Details of customer.py class.**

