To run the application, navigate to the folder and in the terminal run python application.py

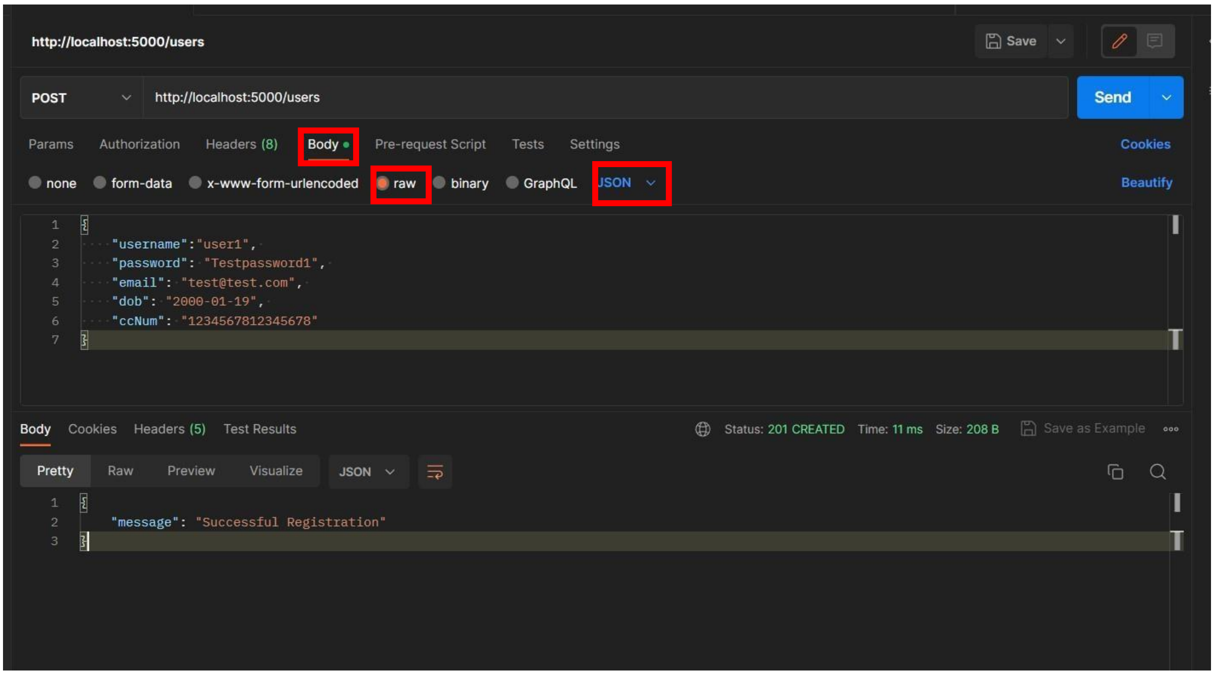
To run the test files, navigate to the folder and in the terminal run pytest applicationTest.py

The app does not need to be running to run the tests as a local app is created in the test file.

#You may need to install Flask, pytest, json and other modules in pip.

You can use Postman or curl, to do it in Postman create a new request and in the URL section and type http://localhost:5000/users assuming you are running on that port.

Make sure that the options marked in red are as shown.



POST http://localhost:5000/users

Body ● raw JSON

```
{
    "username":"user1",
    "password": "Testpassword1",
    "email": "test@test.com",
    "dob": "2000-01-19",
    "ccNum": "1234567812345678"
}
```

Body — Status: 201 CREATED Time: 11 ms Size: 208 B — JSON

```
{
    "message": "Successful Registration"
}
```

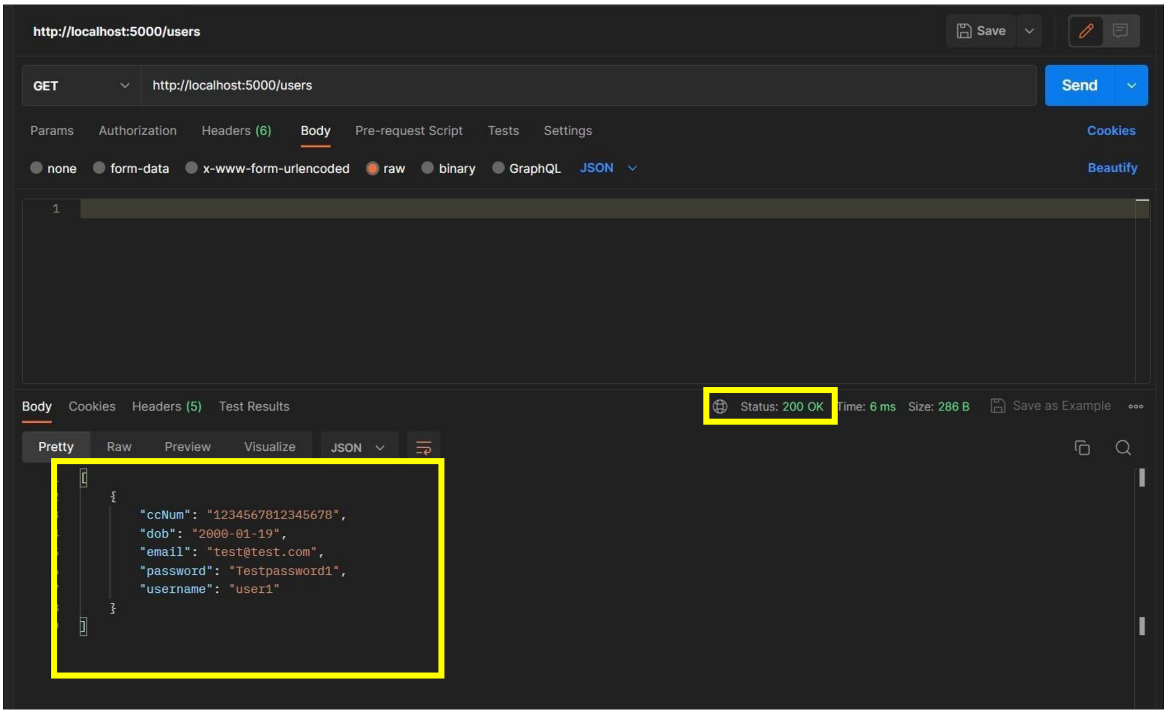
In the text box type some JSON in that you wish to add to the application, making sure it follows that formatting.

```
{
        “username” : “example”,
        “password”: “Example1”,
        “email” : example@example.com,
        “dob” : “2000-01-19”,
        “ccNum” : “1234567812345678”
}
```

Once done press send, in the body section at the bottom you will see the results depending on if the data you entered is valid or not.

--

To view the users you have added, change the method to GET and clear the JSON, press send and then you'll see the results at the bottom.



GET http://localhost:5000/users

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

Status: 200 OK Time: 6 ms Size: 286 B

JSON

```
[
    {
        "ccNum": "1234567812345678",
        "dob": "2000-01-19",
        "email": "test@test.com",
        "password": "Testpassword1",
        "username": "user1"
    }
]
```

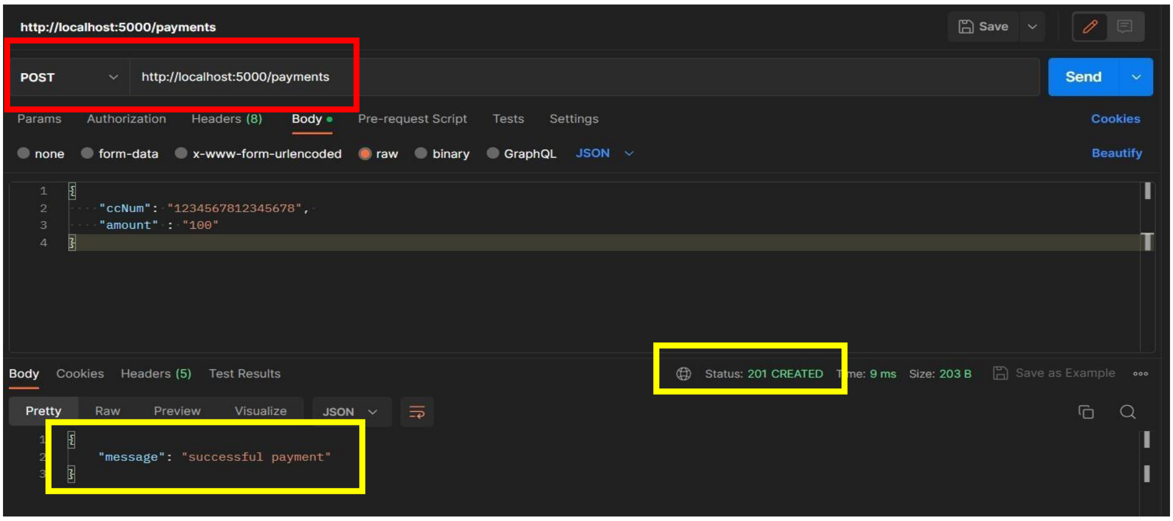
The results are at the bottom and the status code is displayed (highlighted in the yellow box).

To search for users with and without credit cards in the url add ?CreditCard=Yes or ?CreditCard=No onto the end of /users depending on what search option you wish to use.

/users?CreditCard=No or /users?CreditCard=Yes

--

To use the /payment functionality of the application in the URL change the /user to /payments to access the /payments function (red box) and make sure to change the method to POST (red box) and then in the JSON text box add the payment details you wish ensuring they follow the same format as shown.



```
{
    "ccNum": "1234567812345678",
    "amount" : "100"
}
```

```
{
    "message": "successful payment"
}
```

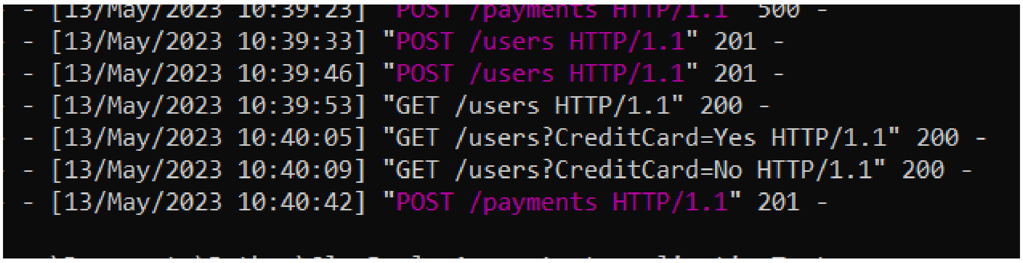
The formatting for the JSON is

```
{
        "ccNum" : "1234567812345678",
        "amount" : "100"
}
```

The results of that request will be shown at the bottom as well as the HTTP code (yellow boxes)

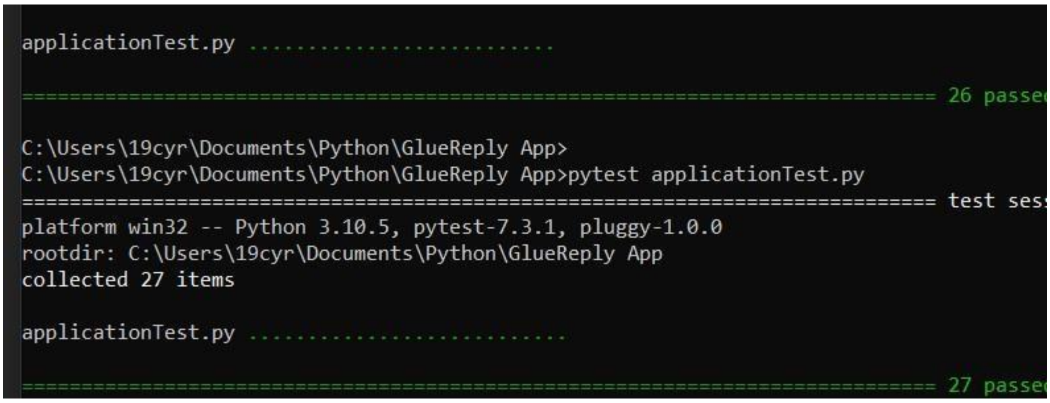
--

Terminal logs for the application showing the status codes for successful operation.



```
- [13/May/2023 10:39:23] "POST /payments HTTP/1.1" 500 -
- [13/May/2023 10:39:33] "POST /users HTTP/1.1" 201 -
- [13/May/2023 10:39:46] "POST /users HTTP/1.1" 201 -
- [13/May/2023 10:39:53] "GET /users HTTP/1.1" 200 -
- [13/May/2023 10:40:05] "GET /users?CreditCard=Yes HTTP/1.1" 200 -
- [13/May/2023 10:40:09] "GET /users?CreditCard=No HTTP/1.1" 200 -
- [13/May/2023 10:40:42] "POST /payments HTTP/1.1" 201 -
```

Proof that all 27 tests run and pass. (each green dot is a passed test, fails are F)



```
applicationTest.py ..........................

============================================================================ 26 passe

C:\Users\19cyr\Documents\Python\GlueReply App>
C:\Users\19cyr\Documents\Python\GlueReply App>pytest applicationTest.py
============================================================================ test ses
platform win32 -- Python 3.10.5, pytest-7.3.1, pluggy-1.0.0
rootdir: C:\Users\19cyr\Documents\Python\GlueReply App
collected 27 items

applicationTest.py ...........................

============================================================================ 27 passe
```