Testing the API’s for TunedIn on Postman

1.For Signup

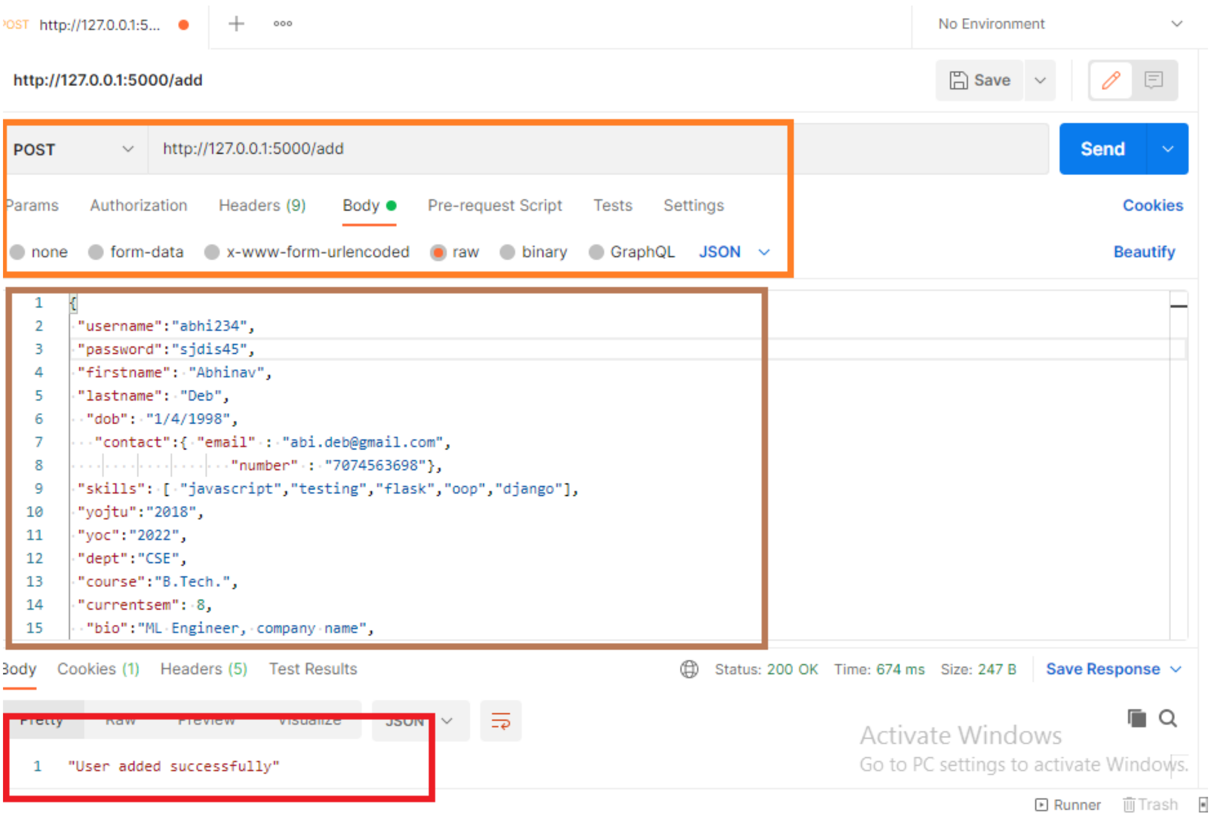
Method:POST

API endpoint: http://127.0.0.1/add

The Body tab in Postman allows us to specify the data we need to send with a request.If we use raw mode for our body data, Postman will set a header based on the type we select (such as text or json).Here we select json.

The json document should look like:

```
{
   "username":"abc234",
   "password":"sjdis34",
   "firstname":"Abhinav",
   "lastname":"Deb",
   "dob":"1/4/1998",
   "contact":{
      "email":"abi.deb@gmail.com",
      "number":"7074563698"
   },
   "skills":[
      "javascript",
      "testing",
      "flask",
      "oop",
      "django"
   ],
   "yojtu":"2018",
   "yoc":"2022",
   "dept":"CSE",
   "course":"B.Tech.",
   "currentsem":8,
   "bio":"ML Engineer, company name",
   "residence":"Jorhat",
   "hostel":"CMH",
   "education":{
      "class10":{
         "marks":{
            "cgpa":9,
            "percentage":0
         },
         "nameofschool":"Bidya Mandir",
         "place":{
            "city":"Jorhat",
            "district":"Jorhat",
            "state":"Assam"
         },
         "year":2015,
         "subjects":[
            "maths",
            "science",
            "english",
            "hindi",
            "soft skills"
         ],
         "board":"cbse"
      },
      "class12":{
         "marks":{
            "cgpa":0,
            "percentage":79
         },
         "nameofschool":"VKV",
         "place":{
            "city":"Sivsagar",
            "district":"Sivsagar",
            "state":"Assam"
         },
         "year":2017,
         "subjects":[
            "chemistry",
            "physics",
            "mathematics",
            "english"
         ],
         "board":"cbse"
      }
   },
   "internships":{
      "id1":{
         "company":"Airtel Industries",
         "title":" Devops Intern",
         "startdate":"23-04-2021",
         "enddate":"23-07-2021",
         "place":{
            "city":"Mumbai",
            "district":"Navi Mumbai",
            "state":"Maharashtra"
         },
            "description":"Worked on.. under the lead of..using tech..
"
         },
         "id2":{
            "company":"Jio Foundation",
            "title":"Django Developer",
            "startdate":"23-01-2021",
            "enddate":"23-03-2021",
            "place":{
               "city":"dibrugarh",
               "district":"dibrugarh",
               "state":"Assam"
            },
            "description":"Worked on.. under the lead of..using tech..
"
         }
      }
   }
```



http://127.0.0.1:5000/add

POST http://127.0.0.1:5000/add

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

```
{
 "username":"abhi234",
 "password":"sjdis45",
 "firstname": "Abhinav",
 "lastname": "Deb",
  "dob": "1/4/1998",
   "contact":{ "email" : "abi.deb@gmail.com",
                    "number" : "7074563698"},
 "skills": [ "javascript","testing","flask","oop","django"],
 "yojtu":"2018",
 "yoc":"2022",
 "dept":"CSE",
 "course":"B.Tech.",
 "currentsem": 8,
  "bio":"ML Engineer, company name",
```

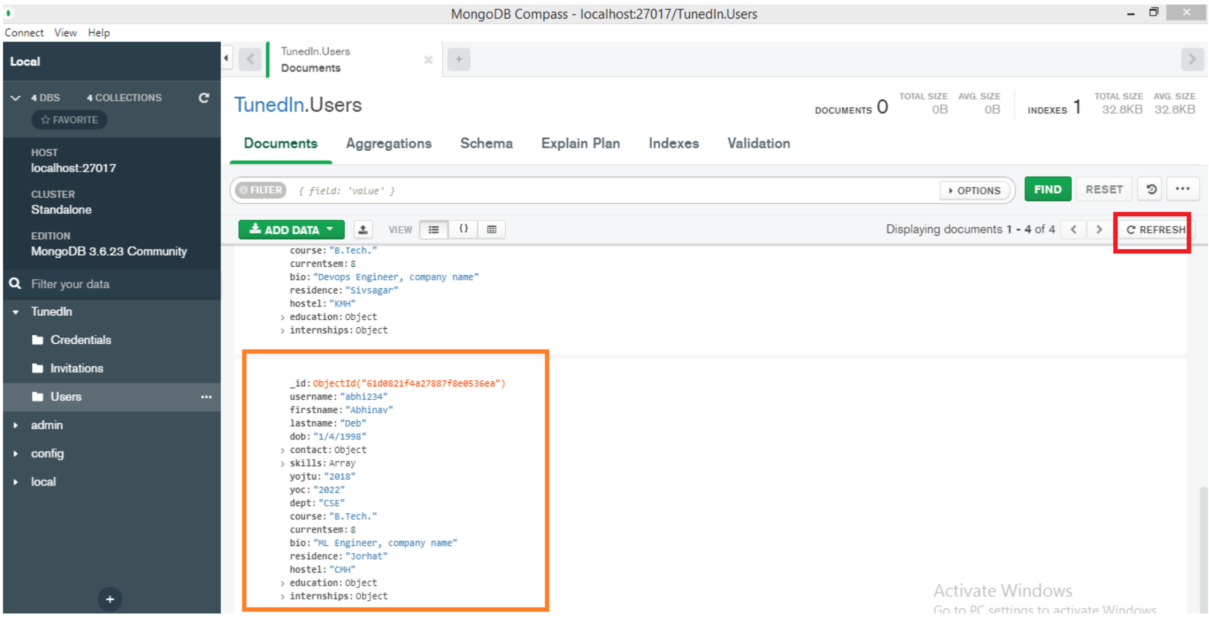
Status: 200 OK Time: 674 ms Size: 247 B

JSON

```
"User added successfully"
```

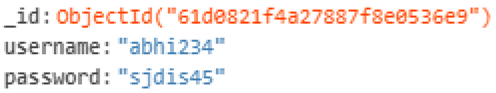
First, the username is checked. If there is already a user with the same username then the user is asked to change the username.On successful

accepting the username the username and password is extracted and inserted into the “Credentials” collection and other data excluding the password into the “Users” collection.
On refreshing the database we will be able to see our new document on Users and Credentials collection



```
course: "B.Tech."
currentsem: 8
bio: "Devops Engineer, company name"
residence: "Sivsagar"
hostel: "KMH"
> education: Object
> internships: Object
```

```
_id: ObjectId("61d0821f4a27887f8e0536ea")
username: "abhi234"
firstname: "Abhinav"
lastname: "Deb"
dob: "1/4/1998"
> contact: Object
> skills: Array
yojtu: "2018"
yoc: "2022"
dept: "CSE"
course: "B.Tech."
currentsem: 8
bio: "ML Engineer, company name"
residence: "Jorhat"
hostel: "CMH"
> education: Object
> internships: Object
```



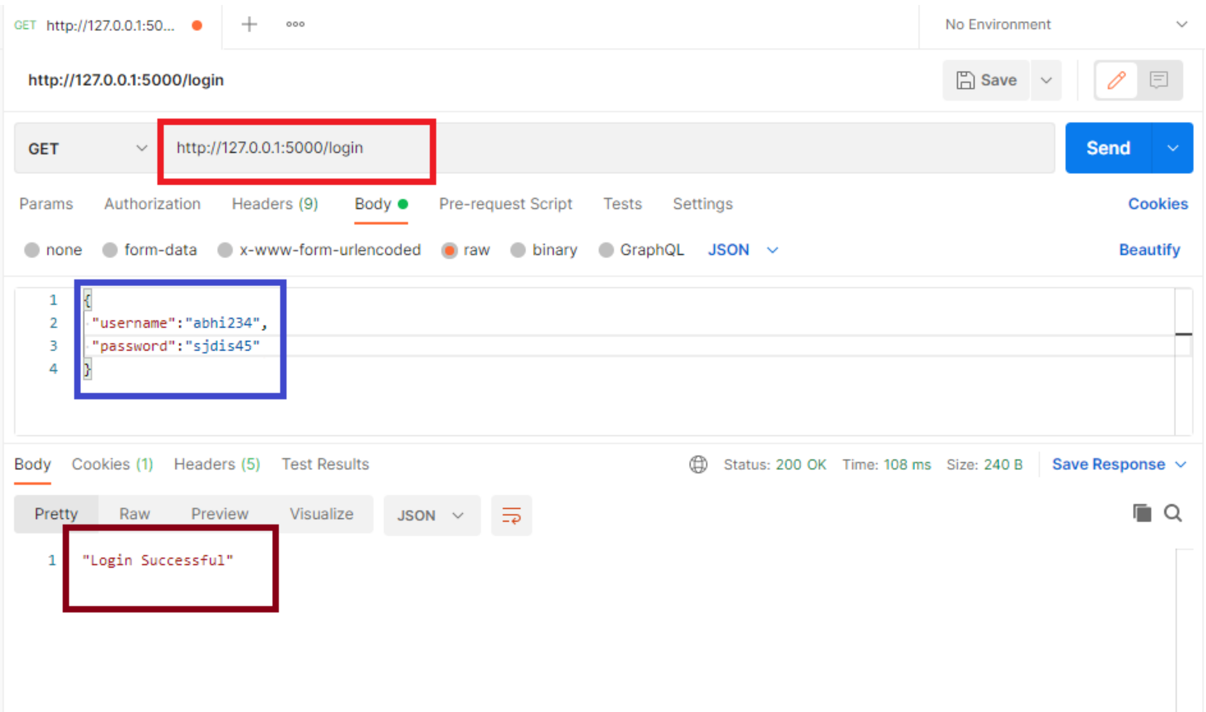
```
_id: ObjectId("61d0821f4a27887f8e0536e9")
username: "abhi234"
password: "sjdis45"
```

2.Login

The user login with username and password. These are validated from the “Credentials” collection and then a flask-session is created.Flask-Session is an extension for Flask that support server-side session to our application.Each client will have their own session there own data will be store in their session.In our case we store the username.
If the username and password does not match then 401 error code is flagged.
API endpoint: http://127.0.0.1/login
Json:

```
{
  "username":"abhi234",
  "password":"sjdis45"
}
```

Assuming the above user exists in DB



GET http://127.0.0.1:5000/login

```
{
"username":"abhi234",
"password":"sjdis45"
}
```

Status: 200 OK Time: 108 ms Size: 240 B

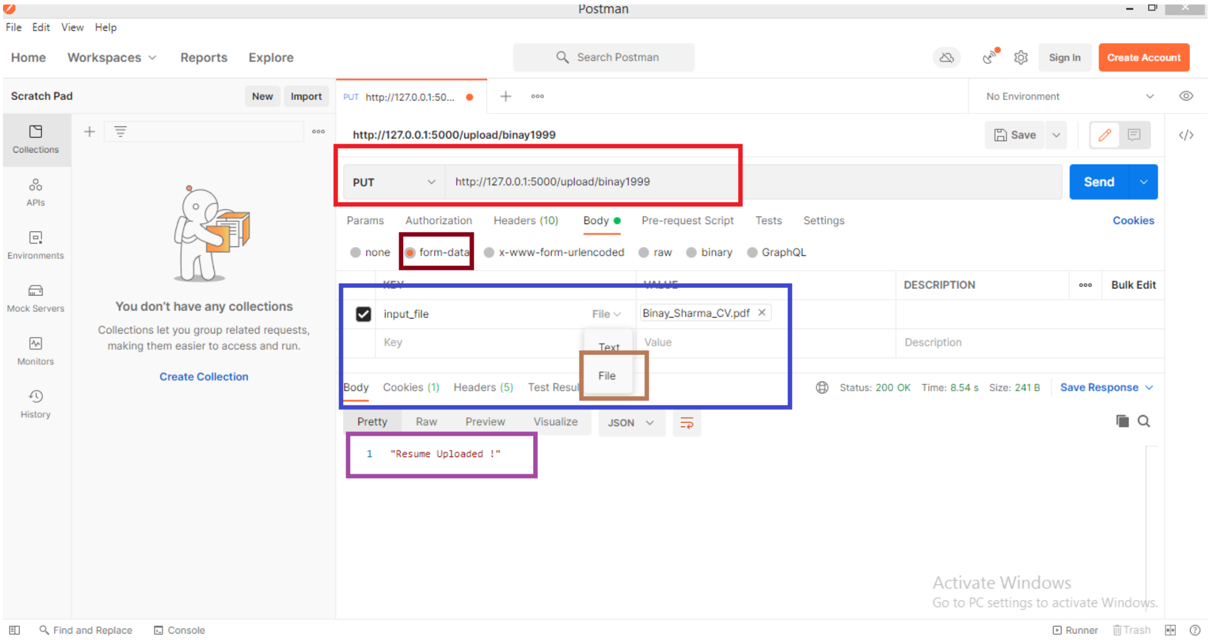
```
"Login Successful"
```

3.Upload Resume

API endpoint: http://127.0.0.1:5000/upload/<username>

Method: PUT since it updates the Users db

We need to select the form data and enter the key as “input_file” and from dropdown select file. This gives the option for selecting the file from our local computer.



PUT http://127.0.0.1:5000/upload/binay1999

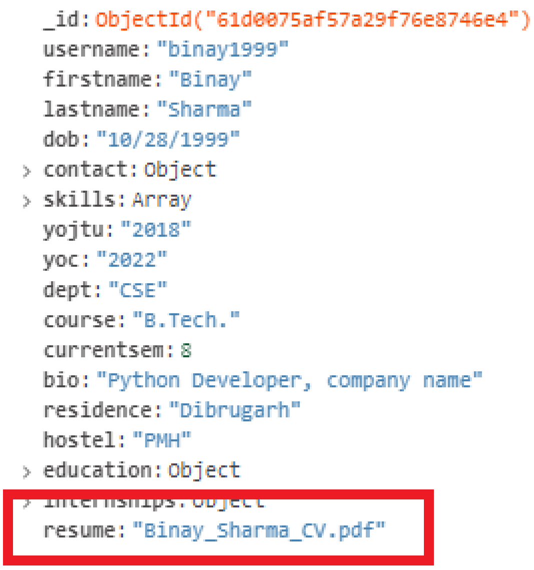
Body: form-data

KEY	VALUE	DESCRIPTION
☑ input_file (File)	Binay_Sharma_CV.pdf	
Key	Value	Description

Status: 200 OK Time: 8.54 s Size: 241 B

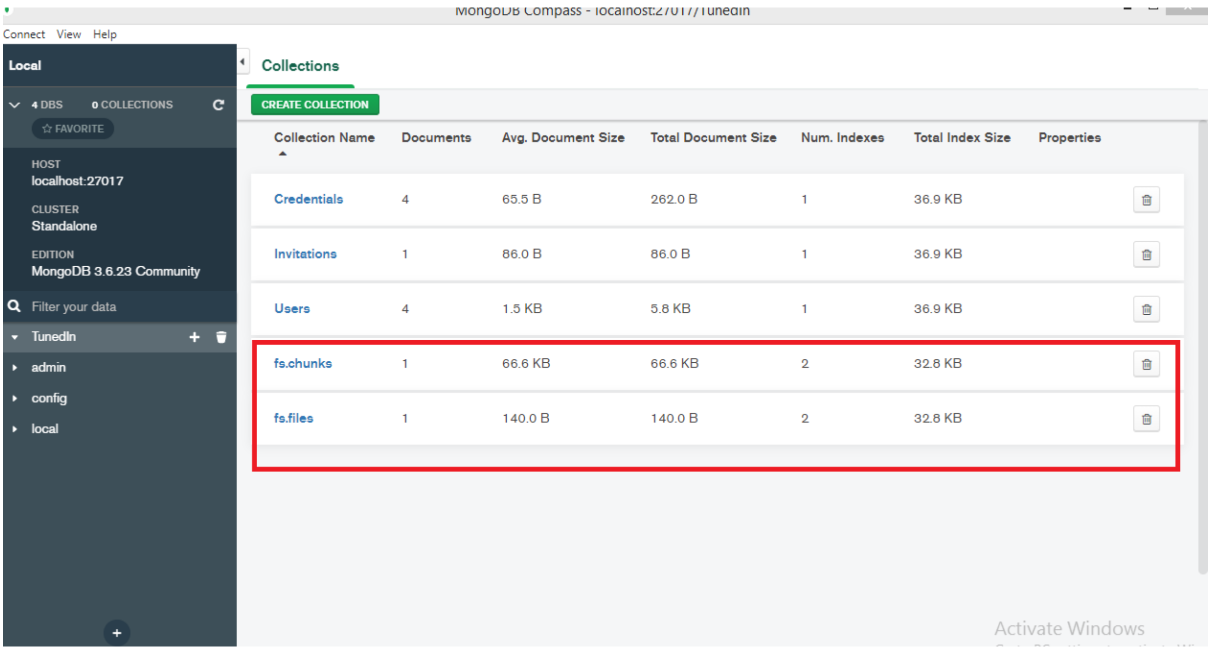
```
"Resume Uploaded !"
```

On refreshing the database we will be able to see the resume field being added



```
  _id: ObjectId("61d0075af57a29f76e8746e4")
  username: "binay1999"
  firstname: "Binay"
  lastname: "Sharma"
  dob: "10/28/1999"
> contact: Object
> skills: Array
  yojtu: "2018"
  yoc: "2022"
  dept: "CSE"
  course: "B.Tech."
  currentsem: 8
  bio: "Python Developer, company name"
  residence: "Dibrugarh"
  hostel: "PMH"
> education: Object
> internships: Object
  resume: "Binay_Sharma_CV.pdf"
```

Apart from the existing collections two new collections also get added when we upload the pdf

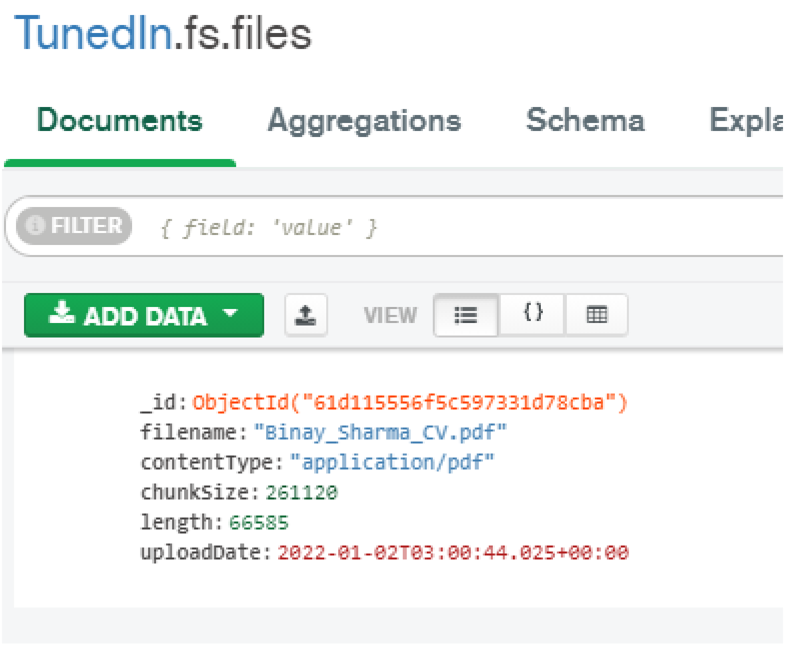


HOST
localhost:27017

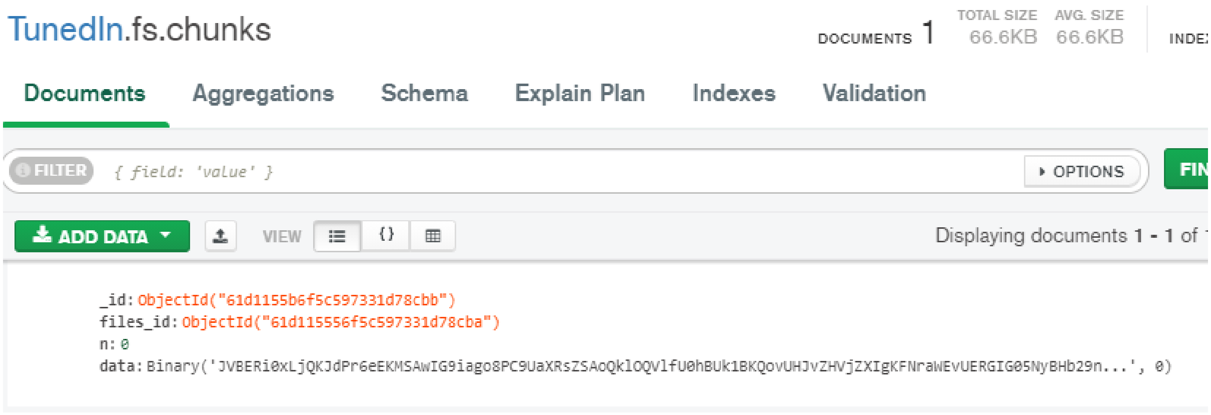
CLUSTER
Standalone

EDITION
MongoDB 3.6.23 Community

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
Credentials	4	65.5 B	262.0 B	1	36.9 KB	
Invitations	1	86.0 B	86.0 B	1	36.9 KB	
Users	4	1.5 KB	5.8 KB	1	36.9 KB	
fs.chunks	1	66.6 KB	66.6 KB	2	32.8 KB	
fs.files	1	140.0 B	140.0 B	2	32.8 KB	



```
_id: ObjectId("61d115556f5c597331d78cba")
filename: "Binay_Sharma_CV.pdf"
contentType: "application/pdf"
chunkSize: 261120
length: 66585
uploadDate: 2022-01-02T03:00:44.025+00:00
```



```
_id: ObjectId("61d1155b6f5c597331d78cbb")
files_id: ObjectId("61d115556f5c597331d78cba")
n: 0
data: Binary('JVBERi0xLjQKJdPr6eEKMSAwIG9iago8PC9UaXRsZSAoQklOQVlfU0hBUk1BKQovUHJvZHVjZXIgKFNraWEvUERGIG05NyBHb29n...', 0)
```

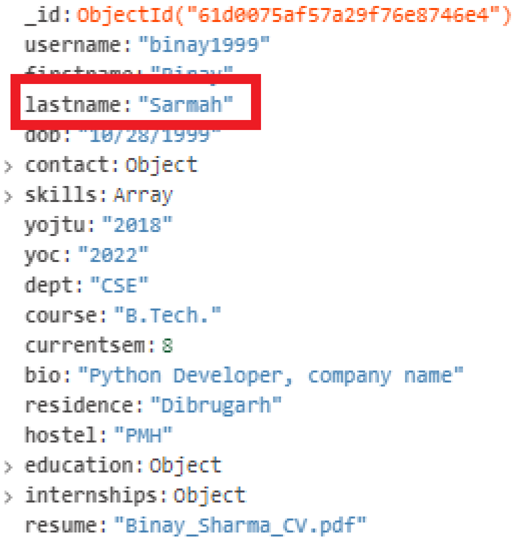
4.Update details

API endpoint:http://127.0.0.1:5000/update/<username>

Method:PUT

A signed in user can update details.The body contains json for which needs to be updated.The json looks like :

```
{
   "username":"binay1999",
   "lastname":"Sarmah"
}
```



```
_id: ObjectId("61d0075af57a29f76e8746e4")
username: "binay1999"
firstname: "Binay"
lastname: "Sarmah"
dob: "10/28/1999"
> contact: Object
> skills: Array
yojtu: "2018"
yoc: "2022"
dept: "CSE"
course: "B.Tech."
currentsem: 8
bio: "Python Developer, company name"
residence: "Dibrugarh"
hostel: "PMH"
> education: Object
> internships: Object
resume: "Binay_Sharma_CV.pdf"
```

On refreshing the db we can see that the details have been updated .Here lastname got updated from “Sharma” to “Sarmah”

5.Search with autocomplete feature

API:

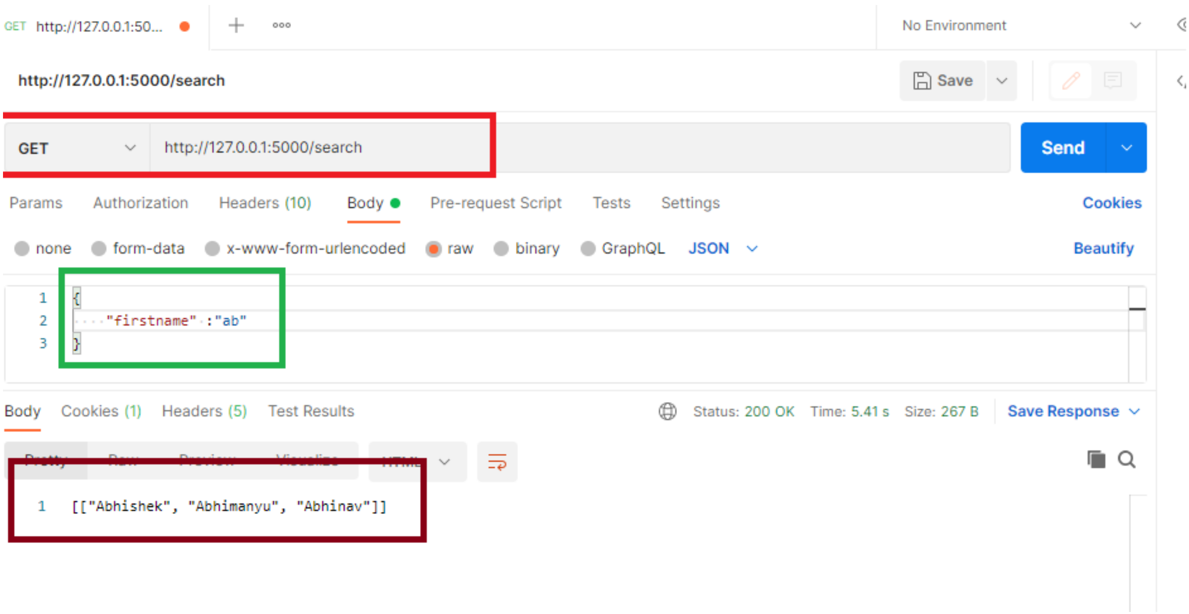
Method: GET

The json body contains the fields to be searched.If all conditions match then the document is returned.If the name does not match then a. regex search is made which returns a list containing the autocomplete names of the student.If we search by skill then if any of the skill mentioned in the query is matched with any students skill set,then that students document is returned.

5.1 Search by firstname prefix. The output is a list containing a list of all the possible names starting with the prefix. This is case insensitive.

The json is:

```
{
  "firstname":"ab"
}
```



http://127.0.0.1:5000/search

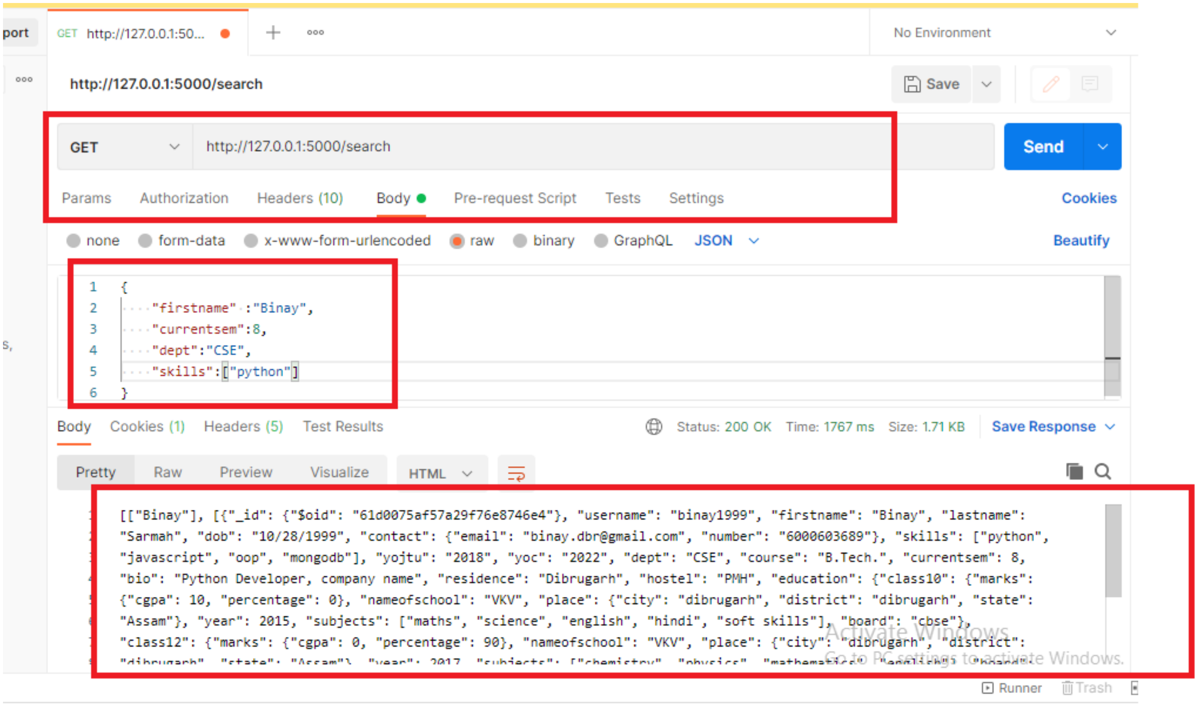
GET http://127.0.0.1:5000/search

```
{
    "firstname" :"ab"
}
```

Status: 200 OK Time: 5.41 s Size: 267 B

```
[["Abhishek", "Abhimanyu", "Abhinav"]]
```

5.2



http://127.0.0.1:5000/search

GET http://127.0.0.1:5000/search

```
{
    "firstname" :"Binay",
    "currentsem":8,
    "dept":"CSE",
    "skills":["python"]
}
```

Status: 200 OK Time: 1767 ms Size: 1.71 KB

```
[["Binay"], [{"_id": {"$oid": "61d0075af57a29f76e8746e4"}, "username": "binay1999", "firstname": "Binay", "lastname":
"Sarmah", "dob": "10/28/1999", "contact": {"email": "binay.dbr@gmail.com", "number": "6000603689"}, "skills": ["python",
"javascript", "oop", "mongodb"], "yojtu": "2018", "yoc": "2022", "dept": "CSE", "course": "B.Tech.", "currentsem": 8,
"bio": "Python Developer, company name", "residence": "Dibrugarh", "hostel": "PMH", "education": {"class10": {"marks":
{"cgpa": 10, "percentage": 0}, "nameofschool": "VKV", "place": {"city": "dibrugarh", "district": "dibrugarh", "state":
"Assam"}, "year": 2015, "subjects": ["maths", "science", "english", "hindi", "soft skills"], "board": "cbse"},
"class12": {"marks": {"cgpa": 0, "percentage": 90}, "nameofschool": "VKV", "place": {"city": "dibrugarh", "district":
"dibrugarh", "state": "Assam"}, "year": 2017, "subjects": ["chemistry", "physics", "mathematics", "english",
```

Here we have searched with firstname,current semester ,department and skill. The result is a list with names recommended and the second list is the required document.We can see that even if we have mentioned only one skill which matches the skill set of the user, the document is returned. If there are multiple skills then, all will be compared and the result will be shown only if all the required skills are present.

6.Delete account

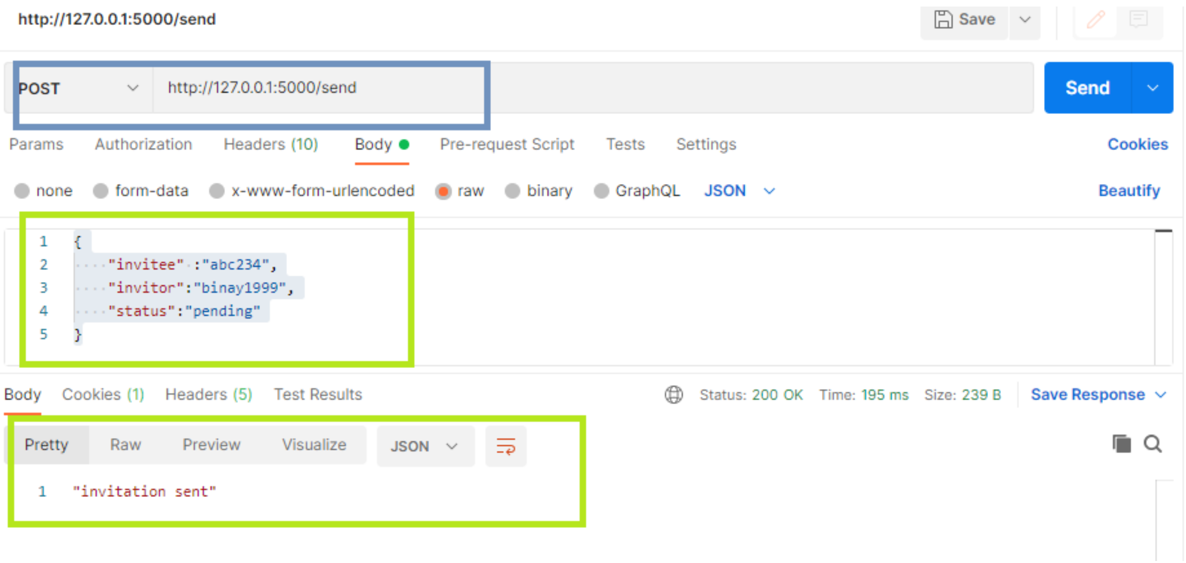
API :

Method : DELETE
A signed in user can delete the account.The user needs to pass the password in the json body for verification.
On refreshing the db, we can see the entry is deleted from the collections

7.Send connection request

API:http://127.0.0.1:5000/send
Method: POST
If a user sends a connection request, the the code first checks if the invitee (user to whom the connection request is sent) exists or not and is there any existing connection between the invitee and the invitor (user who sends the request). The invitor must be logged in to send the request. The json look like

```
{
  "invitee":"abc234",
  "invitor":"binay1999",
  "status":"pending"
}
```



http://127.0.0.1:5000/send

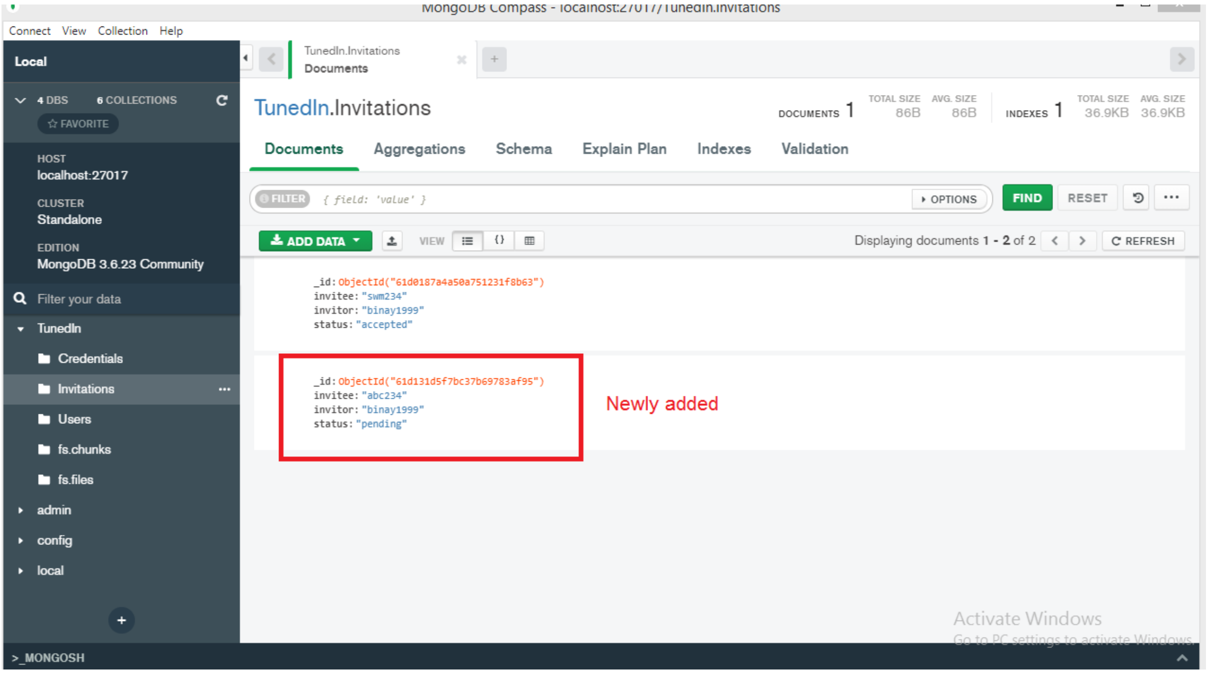
POST http://127.0.0.1:5000/send

```
{
    "invitee" :"abc234",
    "invitor":"binay1999",
    "status":"pending"
}
```

Status: 200 OK Time: 195 ms Size: 239 B

```
"invitation sent"
```

This new document is inserted in the **Invitations** db.



```
_id: ObjectId("61d0187a4a50a751231f8b63")
invitee: "swm234"
invitor: "binay1999"
status: "accepted"
```

```
_id: ObjectId("61d131d5f7bc37b69783af95")
invitee: "abc234"
invitor: "binay1999"
status: "pending"
```

Newly added

8.Accept connection request

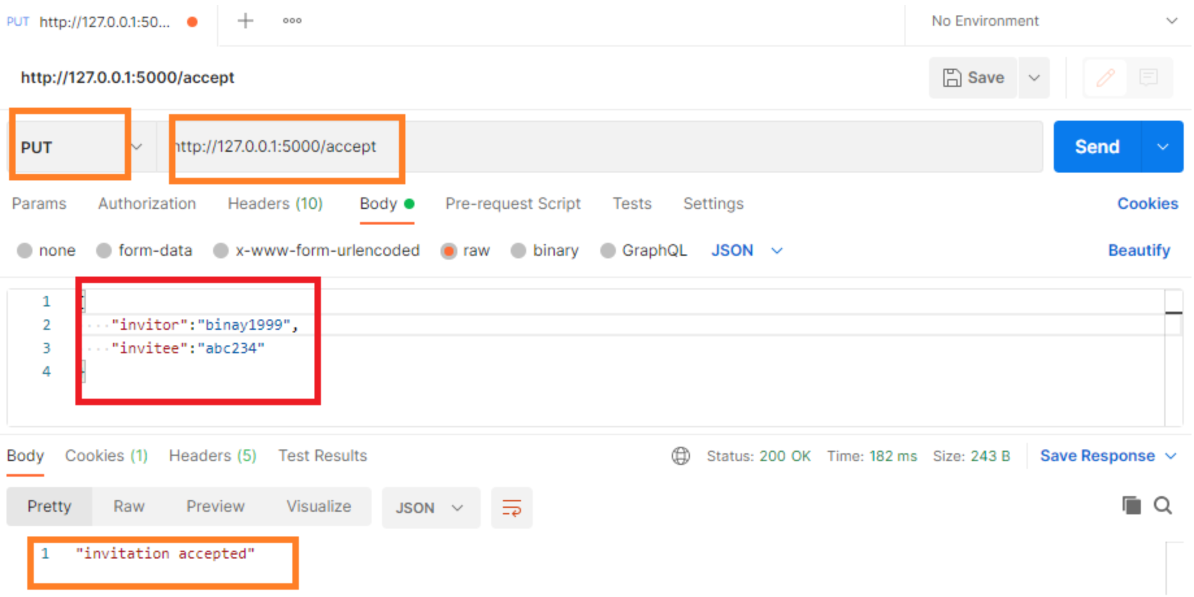
API:http://127.0.0.1:5000/accept

Method: PUT

A user can accept a connection request if he/she is in the invitee position in the invitations database. In this case the invitor must be logged in. The code checks if there is any pending request from the invitor and is the connection already accepted .If all checks are correct then an update command is performed and status is set to "accepted'.

The json looks like

```
{
   "invitor":"binay1999",
   "invitee":"abc234"
}
```



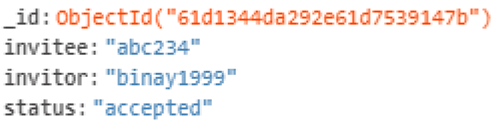
http://127.0.0.1:5000/accept

```
{
    "invitor":"binay1999",
    "invitee":"abc234"
}
```

Status: 200 OK Time: 182 ms Size: 243 B

```
"invitation accepted"
```

On refreshing the db, we can see the changes.



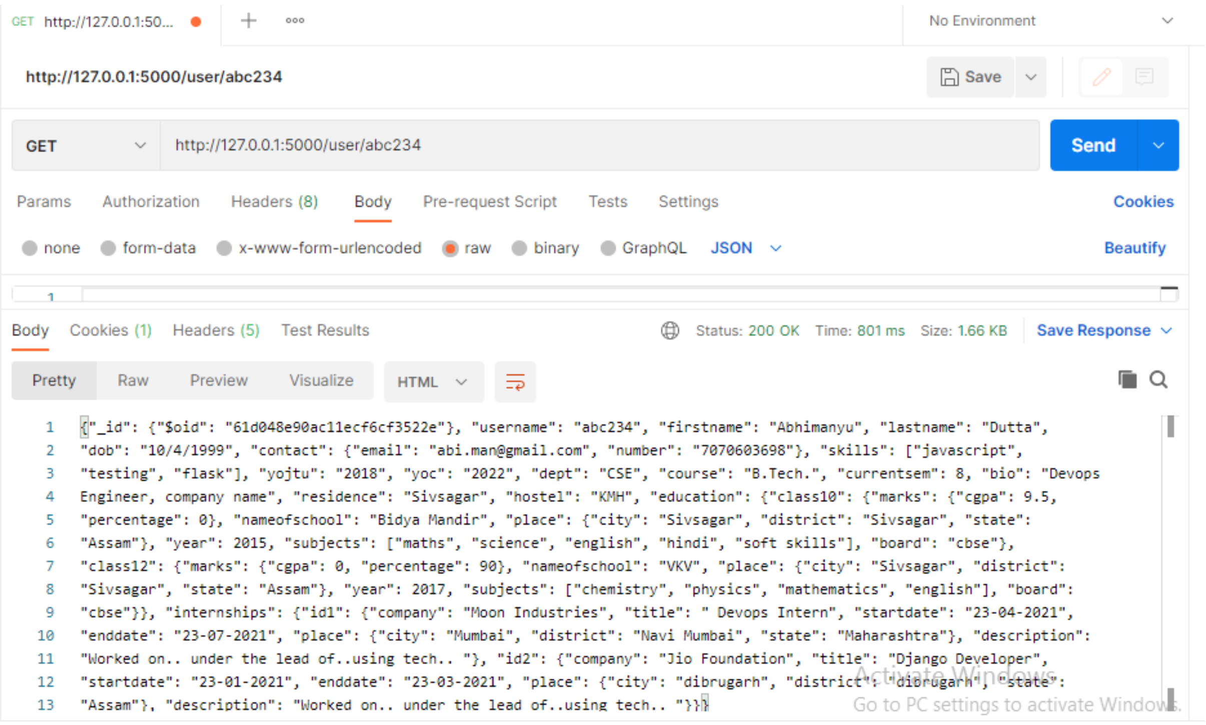
```
_id: ObjectId("61d1344da292e61d7539147b")
invitee: "abc234"
invitor: "binay1999"
status: "accepted"
```

9.Get all details

API:http://127.0.0.1:5000/user/<username>

Method: GET

A user can get all his/her details.



http://127.0.0.1:5000/user/abc234

Status: 200 OK Time: 801 ms Size: 1.66 KB

```
{"_id": {"$oid": "61d048e90ac11ecf6cf3522e"}, "username": "abc234", "firstname": "Abhimanyu", "lastname": "Dutta",
"dob": "10/4/1999", "contact": {"email": "abi.man@gmail.com", "number": "7070603698"}, "skills": ["javascript",
"testing", "flask"], "yojtu": "2018", "yoc": "2022", "dept": "CSE", "course": "B.Tech.", "currentsem": 8, "bio": "Devops
Engineer, company name", "residence": "Sivsagar", "hostel": "KMH", "education": {"class10": {"marks": {"cgpa": 9.5,
"percentage": 0}, "nameofschool": "Bidya Mandir", "place": {"city": "Sivsagar", "district": "Sivsagar", "state":
"Assam"}, "year": 2015, "subjects": ["maths", "science", "english", "hindi", "soft skills"], "board": "cbse"},
"class12": {"marks": {"cgpa": 0, "percentage": 90}, "nameofschool": "VKV", "place": {"city": "Sivsagar", "district":
"Sivsagar", "state": "Assam"}, "year": 2017, "subjects": ["chemistry", "physics", "mathematics", "english"], "board":
"cbse"}}, "internships": {"id1": {"company": "Moon Industries", "title": " Devops Intern", "startdate": "23-04-2021",
"enddate": "23-07-2021", "place": {"city": "Mumbai", "district": "Navi Mumbai", "state": "Maharashtra"}, "description":
"Worked on.. under the lead of..using tech.. "}, "id2": {"company": "Jio Foundation", "title": "Django Developer",
"startdate": "23-01-2021", "enddate": "23-03-2021", "place": {"city": "dibrugarh", "district": "dibrugarh", "state":
"Assam"}, "description": "Worked on.. under the lead of..using tech.. "}}}
```