Demand Fulfilment System DFS

(Can watch demo video of project in repo)

Problem Statement:

DFS is a system used by management to fulfill real-time project demands.
When there is a requirement opening in a project, the project team will raise the demand request for the required position with the skills, experience, level in need.
The DFS mgmt will be able to verify all demands raised by different project teams.
Also there will be membersList added to DFS and maintained by DFS mgmt.
When member is added to the DFS system - the member status will be set as AVAILABLE.
If the demand requirement meets with any of the available members, preferred member will be assigned to the demand and the demand will be closed by setting the demand status as CLOSED and member status as ASSIGNED.
If there is no matching member - the DFS mgmt. will close the demand by setting the demand status as NOT_FULFILLED

Object Model:

member

- id
- eid
- firstName
- lastName
- doj
- level
- location
- overallExp
- status
- skills

Note:
i) skills will be map contains Skill as a key and relevant experience in the technology as an Integer
eg : map <String , Integer >
ii) Member status will be **Assigned, Available**

demand

- id
- projectName
- mgrName
- level
- city
- skills
- status
- duration
- startDate

Note:
i) demand status will be OPEN, CLOSED, NOT_FULFILLED

Instructions:

1. Create a Rest API which accepts demand Request and persist the demand in datastore
2. Create a Rest API to retrieve demand by id or with the any one or all of the following *demand* fields
    a. member level in demand

    b. city
    c. manager or project Name
    d. status
    e. skills level
3. Create a REST API to retrieve all members for demand requested from the project team for the following input
    a. eid
    b. name
    c. level
    d. location
    e. *status*
    f. skills

Note:
i) Sort the list of members by Date of Joining, name and City
ii) If no status is set in the request, retrieve all member status

4. Create a Rest API to update the demand fulfilment which assigns the members to the demand submitted.

==> to accept demand req id
==> to save demand-member mapping in dataStore

Table Structure:

Member_Info

Id	Eid	FirstName	LastName	DOJ	Level	Location	OverallExp	Status	Skills
1	sumutha	Subramanian	Muthaiah	22-Jul-23	P01	Chennai	1	Available	Java-12,Springboot-12,Maven-12
2	mohanr	Mohan	Ranjith	23-Jul-23	P01	Chennai	1	Assigned	Java-12,Springboot-12,Maven-13

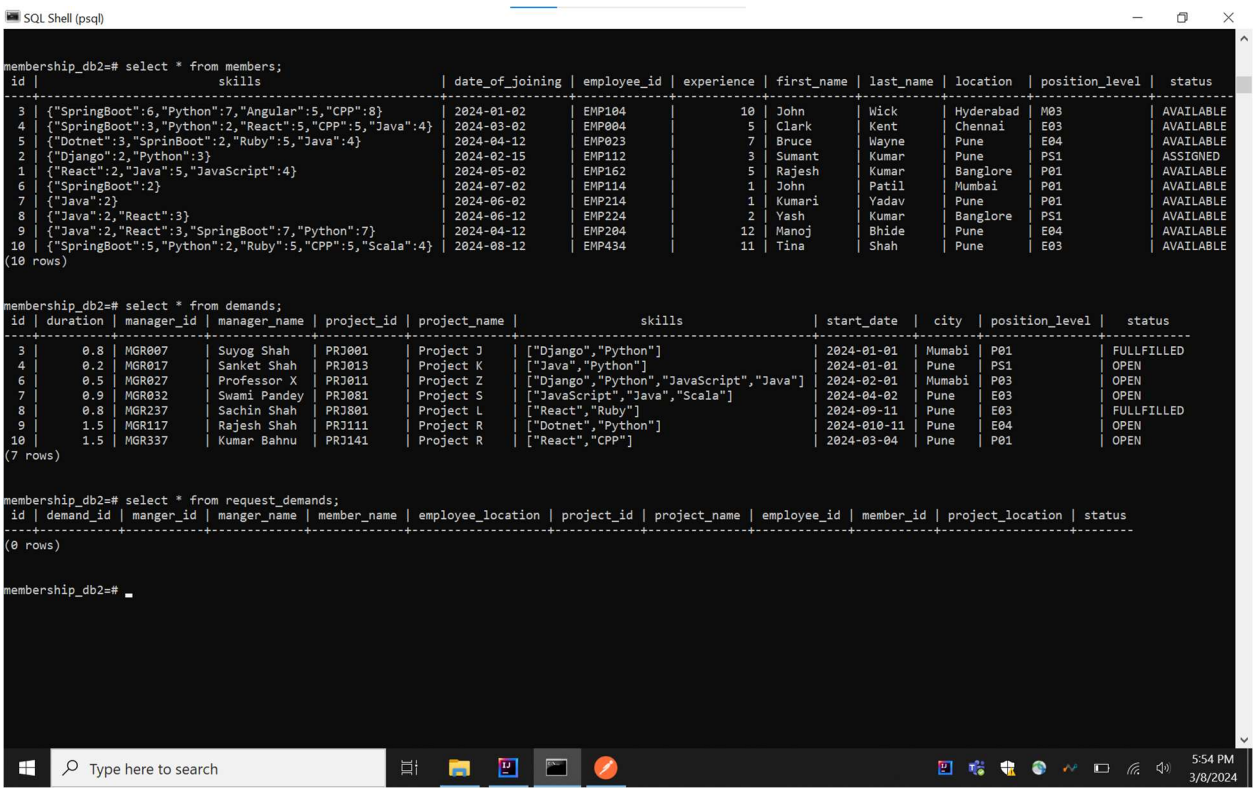
Demand_Info

Id	ProjectName	ManagerName	Level	City	Skills	Status	Duration	StartDate
1	Project1	Name1	P01	Chennai	Java,Springboot,Maven	Closed	0.3	30-Mar-24
2	Project2	Name2	P01	Pune	Angular,NodeJs	Open	1.2	1-Mar-24

In this project there are three table :

- **Members** – for storing data of members
- **Demands** – for storing data of demands make by Requestor for searching in members table and with project details
- **Request_demands** – for storing request made by requestor for a member and for which demand member is requested

Tables :



```
SQL Shell (psql)

membership_db2=# select * from members;
 id |                        skills                        | date_of_joining | employee_id | experience | first_name | last_name | location  | position_level |  status
----+------------------------------------------------------+-----------------+-------------+------------+------------+-----------+-----------+----------------+----------
  3 | {"SpringBoot":6,"Python":7,"Angular":5,"CPP":8}      | 2024-01-02      | EMP104      |         10 | John       | Wick      | Hyderabad | M03            | AVAILABLE
  4 | {"SpringBoot":3,"Python":2,"React":5,"CPP":5,"Java":4} | 2024-03-02      | EMP004      |          5 | Clark      | Kent      | Chennai   | E03            | AVAILABLE
  5 | {"Dotnet":3,"SprinBoot":2,"Ruby":5,"Java":4}         | 2024-04-12      | EMP023      |          7 | Bruce      | Wayne     | Pune      | E04            | AVAILABLE
  2 | {"Django":2,"Python":3}                              | 2024-02-15      | EMP112      |          3 | Sumant     | Kumar     | Pune      | PS1            | ASSIGNED
  1 | {"React":2,"Java":5,"JavaScript":4}                  | 2024-05-02      | EMP162      |          5 | Rajesh     | Kumar     | Banglore  | P01            | AVAILABLE
  6 | {"SpringBoot":2}                                     | 2024-07-02      | EMP114      |          1 | John       | Patil     | Mumbai    | P01            | AVAILABLE
  7 | {"Java":2}                                           | 2024-06-02      | EMP214      |          1 | Kumari     | Yadav     | Pune      | P01            | AVAILABLE
  8 | {"Java":2,"React":3}                                 | 2024-06-12      | EMP224      |          2 | Yash       | Kumar     | Banglore  | PS1            | AVAILABLE
  9 | {"Java":2,"React":3,"SpringBoot":7,"Python":7}       | 2024-04-12      | EMP204      |         12 | Manoj      | Bhide     | Pune      | E04            | AVAILABLE
 10 | {"SpringBoot":5,"Python":2,"Ruby":5,"CPP":5,"Scala":4} | 2024-08-12      | EMP434      |         11 | Tina       | Shah      | Pune      | E03            | AVAILABLE
(10 rows)

membership_db2=# select * from demands;
 id | duration | manager_id | manager_name | project_id | project_name |                skills                | start_date |  city  | position_level |   status
----+----------+------------+--------------+------------+--------------+--------------------------------------+------------+--------+----------------+-----------
  3 |      0.8 | MGR007     | Suyog Shah   | PRJ001     | Project J    | ["Django","Python"]                  | 2024-01-01 | Mumabi | P01            | FULLFILLED
  4 |      0.2 | MGR017     | Sanket Shah  | PRJ013     | Project K    | ["Java","Python"]                    | 2024-01-01 | Pune   | PS1            | OPEN
  6 |      0.5 | MGR027     | Professor X  | PRJ011     | Project Z    | ["Django","Python","JavaScript","Java"] | 2024-02-01 | Mumabi | P03            | OPEN
  7 |      0.9 | MGR032     | Swami Pandey | PRJ081     | Project S    | ["JavaScript","Java","Scala"]        | 2024-04-02 | Pune   | E03            | OPEN
  8 |      0.8 | MGR237     | Sachin Shah  | PRJ801     | Project L    | ["React","Ruby"]                     | 2024-09-11 | Pune   | E03            | FULLFILLED
  9 |      1.5 | MGR117     | Rajesh Shah  | PRJ111     | Project R    | ["Dotnet","Python"]                  | 2024-010-11 | Pune   | E04            | OPEN
 10 |      1.5 | MGR337     | Kumar Bahnu  | PRJ141     | Project R    | ["React","CPP"]                      | 2024-03-04 | Pune   | P01            | OPEN
(7 rows)

membership_db2=# select * from request_demands;
 id | demand_id | manger_id | manger_name | member_name | employee_location | project_id | project_name | employee_id | member_id | project_location | status
----+-----------+-----------+-------------+-------------+-------------------+------------+--------------+-------------+-----------+------------------+--------
(0 rows)

membership_db2=#
```

In this project there are 2 entities :

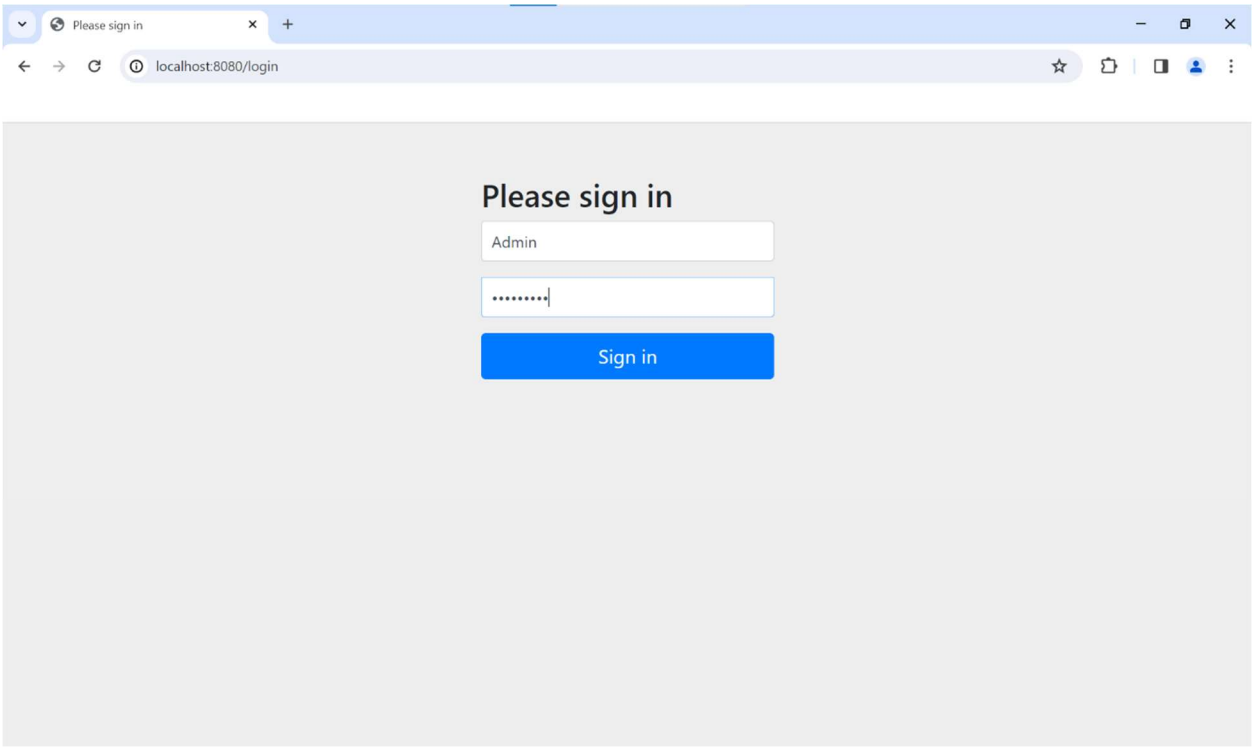
- **Admin** :
  - Create entry in members table
  - Get members from member table completely or by member_id
  - Edit a entry in members table
  - Get members from member table by filtering on various attributes
  - Get Requests from request_demand table

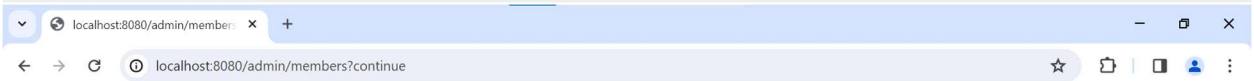
    - Approve the request in request table and it will be reflected on both demand and member table
- **Requestor**
    - Create entry as demands in demand table
    - Get members from member table completely or by member_id
    - Get demand from demand table completely or by demand_id
    - Get demand from demand table by filtering on various attributes
    - Create a request for a member to admin which is stored in request_demand table
    - Get member from member table based on demand table demands

Authentication :

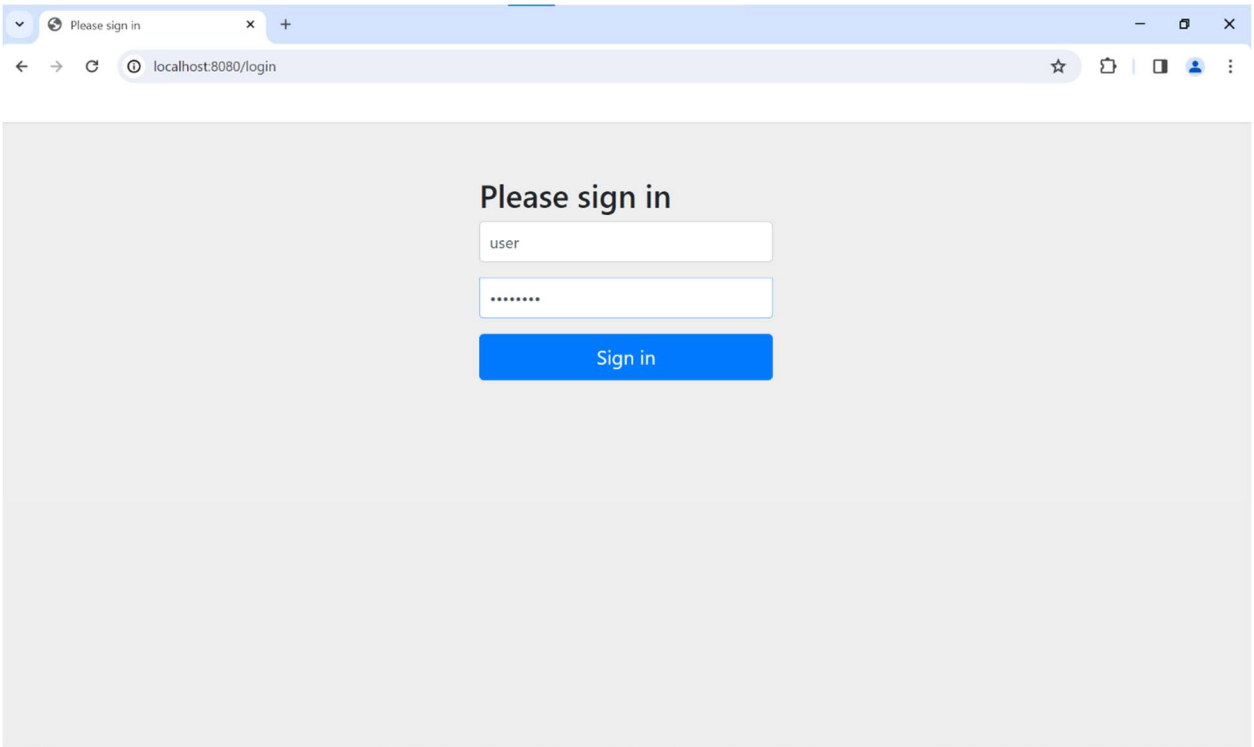
There is authentication for admin and requestor.(*due to some issue post request are not working , but working no it)

Admin and Requestor Authentication and get members and demands request :





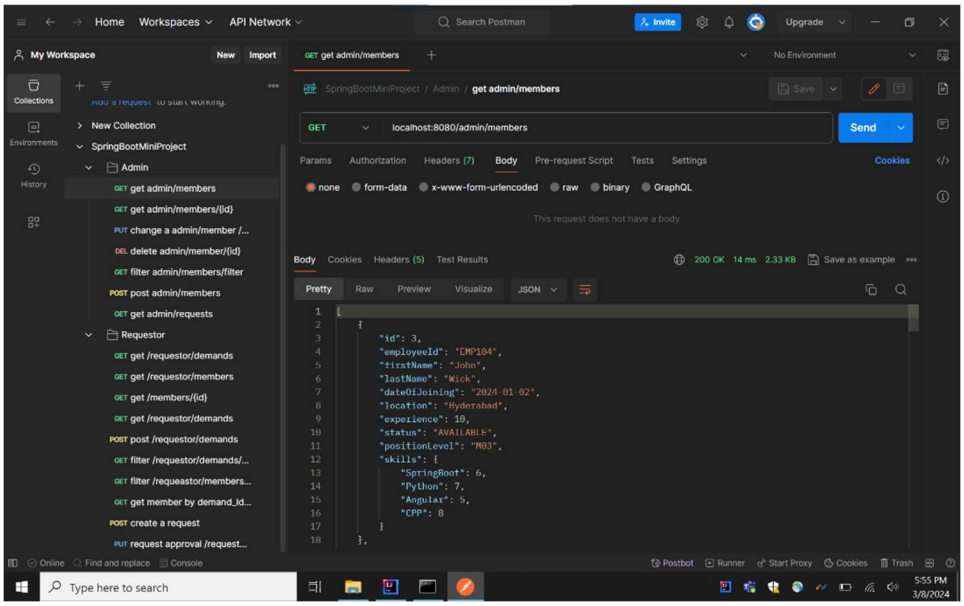

```
[{"id":3,"employeeId":"EMP104","firstName":"John","lastName":"Wick","dateOfJoining":"2024-01-
02","location":"Hyderabad","experience":10,"status":"AVAILABLE","positionLevel":"M03","skills":{"SpringBoot":6,"Python":7,"Angular":5,"CPP":8}},
{"id":4,"employeeId":"EMP004","firstName":"Clark","lastName":"Kent","dateOfJoining":"2024-03-
02","location":"Chennai","experience":5,"status":"AVAILABLE","positionLevel":"E03","skills":{"SpringBoot":3,"Python":2,"React":5,"CPP":5,"Java":4}},
{"id":5,"employeeId":"EMP023","firstName":"Bruce","lastName":"Wayne","dateOfJoining":"2024-04-
12","location":"Pune","experience":7,"status":"AVAILABLE","positionLevel":"E04","skills":{"Dotnet":3,"SprinBoot":2,"Ruby":5,"Java":4}},
{"id":2,"employeeId":"EMP112","firstName":"Sumant","lastName":"Kumar","dateOfJoining":"2024-02-
15","location":"Pune","experience":3,"status":"ASSIGNED","positionLevel":"PS1","skills":{"Django":2,"Python":3}},
{"id":1,"employeeId":"EMP162","firstName":"Rajesh","lastName":"Kumar","dateOfJoining":"2024-05-
02","location":"Banglore","experience":5,"status":"AVAILABLE","positionLevel":"P01","skills":{"React":2,"Java":5,"JavaScript":4}},
{"id":6,"employeeId":"EMP114","firstName":"John","lastName":"Patil","dateOfJoining":"2024-07-
02","location":"Mumbai","experience":1,"status":"AVAILABLE","positionLevel":"P01","skills":{"SpringBoot":2}},
{"id":7,"employeeId":"EMP214","firstName":"Kumari","lastName":"Yadav","dateOfJoining":"2024-06-
02","location":"Pune","experience":1,"status":"AVAILABLE","positionLevel":"P01","skills":{"Java":2}},
{"id":8,"employeeId":"EMP224","firstName":"Yash","lastName":"Kumar","dateOfJoining":"2024-06-
12","location":"Banglore","experience":2,"status":"AVAILABLE","positionLevel":"PS1","skills":{"Java":2,"React":3}},
{"id":9,"employeeId":"EMP204","firstName":"Manoj","lastName":"Bhide","dateOfJoining":"2024-04-
12","location":"Pune","experience":12,"status":"AVAILABLE","positionLevel":"E04","skills":{"Java":2,"React":3,"SpringBoot":7,"Python":7}},
{"id":10,"employeeId":"EMP434","firstName":"Tina","lastName":"Shah","dateOfJoining":"2024-08-
12","location":"Pune","experience":11,"status":"AVAILABLE","positionLevel":"E03","skills":{"SpringBoot":5,"Python":2,"Ruby":5,"CPP":5,"Scala":4}}]
```



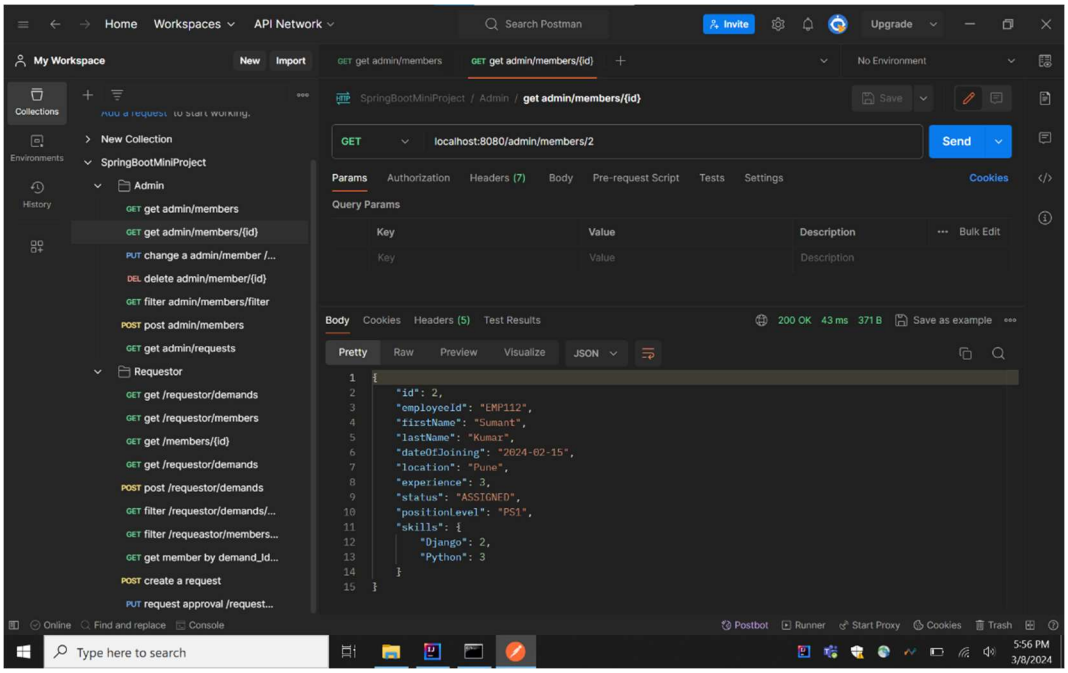
[{"id":3,"positionLevel":"P01","city":"Mumabi","status":"FULLFILLED","duration":0.8,"skills":["Django","Python"],"projectName":"Project J","mangerName":"Suyog Shah","manager_Id":"MGR007","startDate":"2024-01-01","projectId":"PRJ001"},{"id":4,"positionLevel":"PS1","city":"Pune","status":"OPEN","duration":0.2,"skills":["Java","Python"],"projectName":"Project K","mangerName":"Sanket Shah","manager_Id":"MGR017","startDate":"2024-01-01","projectId":"PRJ013"},{"id":6,"positionLevel":"P03","city":"Mumabi","status":"OPEN","duration":0.5,"skills":["Django","Python","JavaScript","Java"],"projectName":"Project Z","mangerName":"Professor X","manager_Id":"MGR027","startDate":"2024-02-01","projectId":"PRJ011"},{"id":7,"positionLevel":"E03","city":"Pune","status":"OPEN","duration":0.9,"skills":["JavaScript","Java","Scala"],"projectName":"Project S","mangerName":"Swami Pandey","manager_Id":"MGR032","startDate":"2024-04-02","projectId":"PRJ081"},{"id":8,"positionLevel":"E03","city":"Pune","status":"FULLFILLED","duration":0.8,"skills":["React","Ruby"],"projectName":"Project L","mangerName":"Sachin Shah","manager_Id":"MGR237","startDate":"2024-09-11","projectId":"PRJ801"},{"id":9,"positionLevel":"E04","city":"Pune","status":"OPEN","duration":1.5,"skills":["Dotnet","Python"],"projectName":"Project R","mangerName":"Rajesh Shah","manager_Id":"MGR117","startDate":"2024-010-11","projectId":"PRJ111"},{"id":10,"positionLevel":"P01","city":"Pune","status":"OPEN","duration":1.5,"skills":["React","CPP"],"projectName":"Project R","mangerName":"Kumar Bahnu","manager_Id":"MGR337","startDate":"2024-03-04","projectId":"PRJ141"}]

Admin :

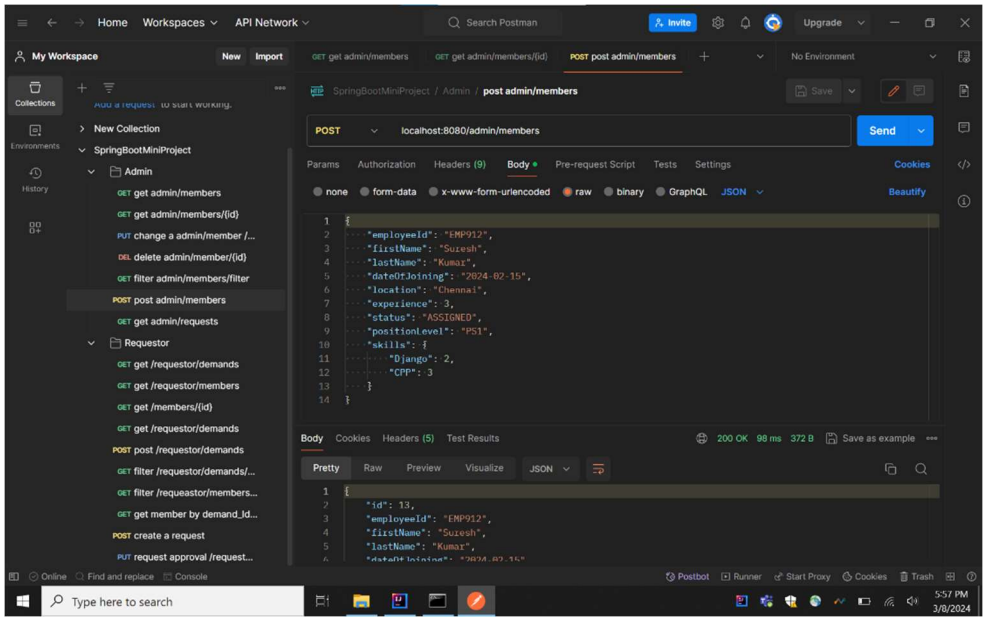
- Get All Members:



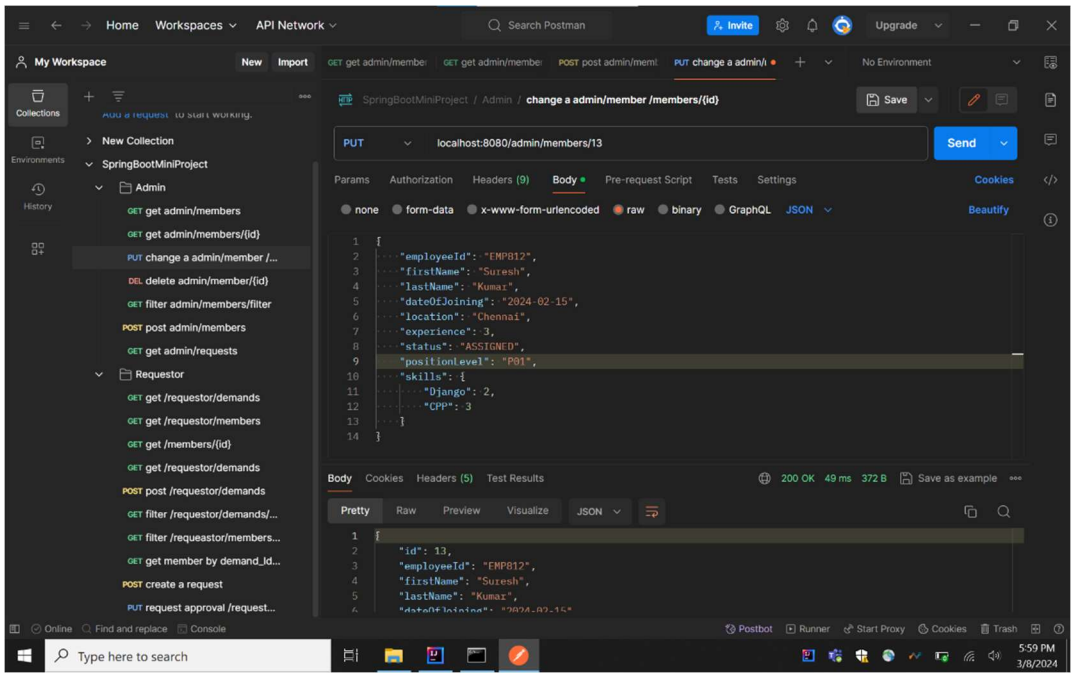
- Get Member by ID



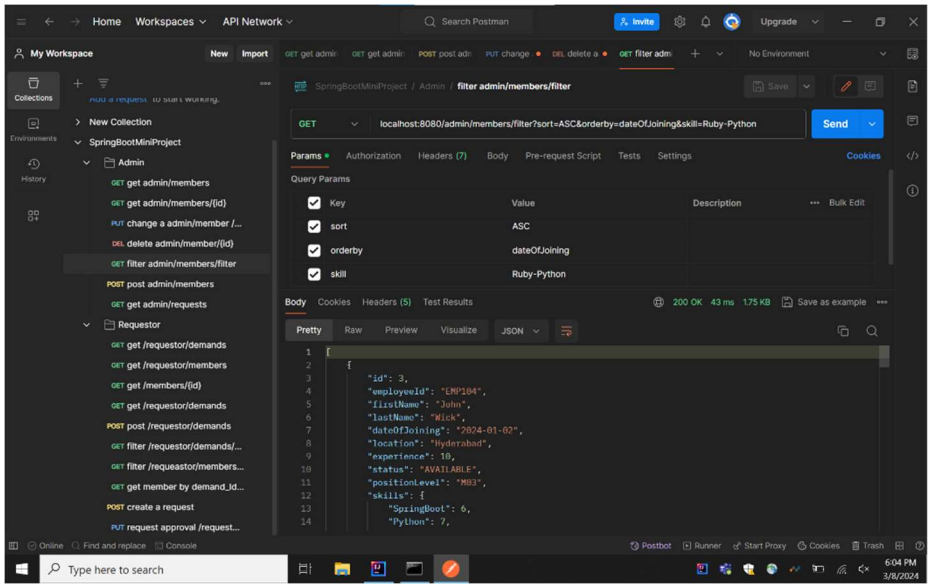
- Post a member record in member table :



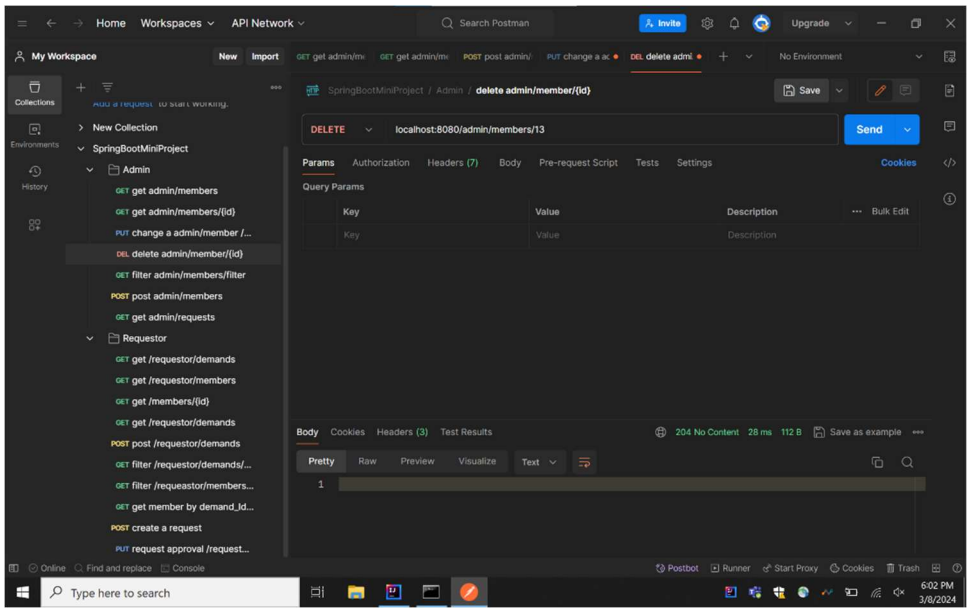
- Change a member :



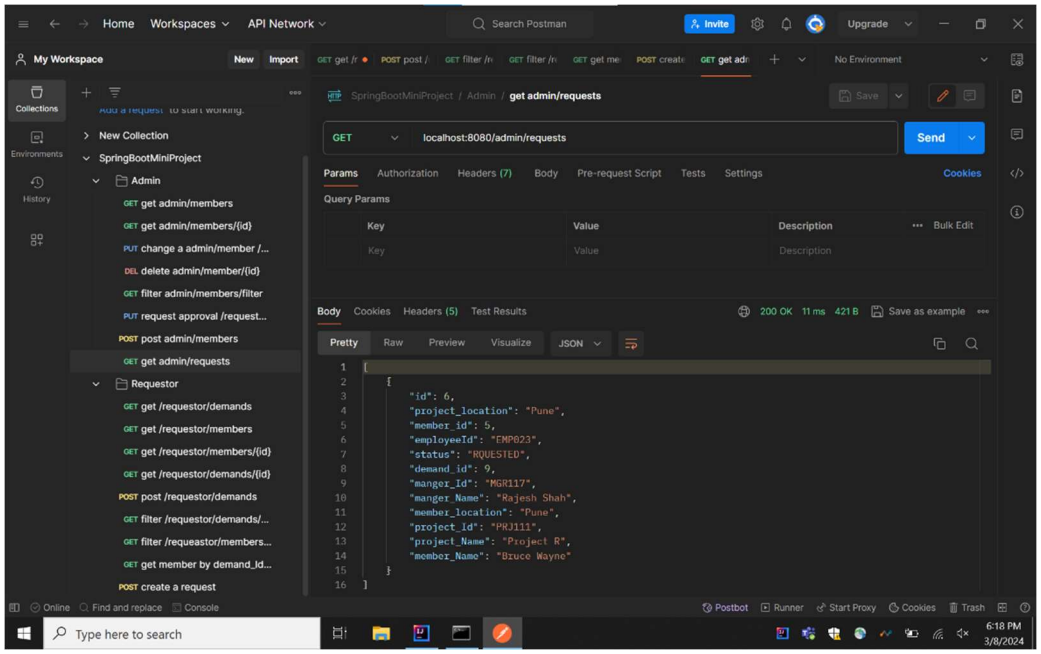
- Filter members :



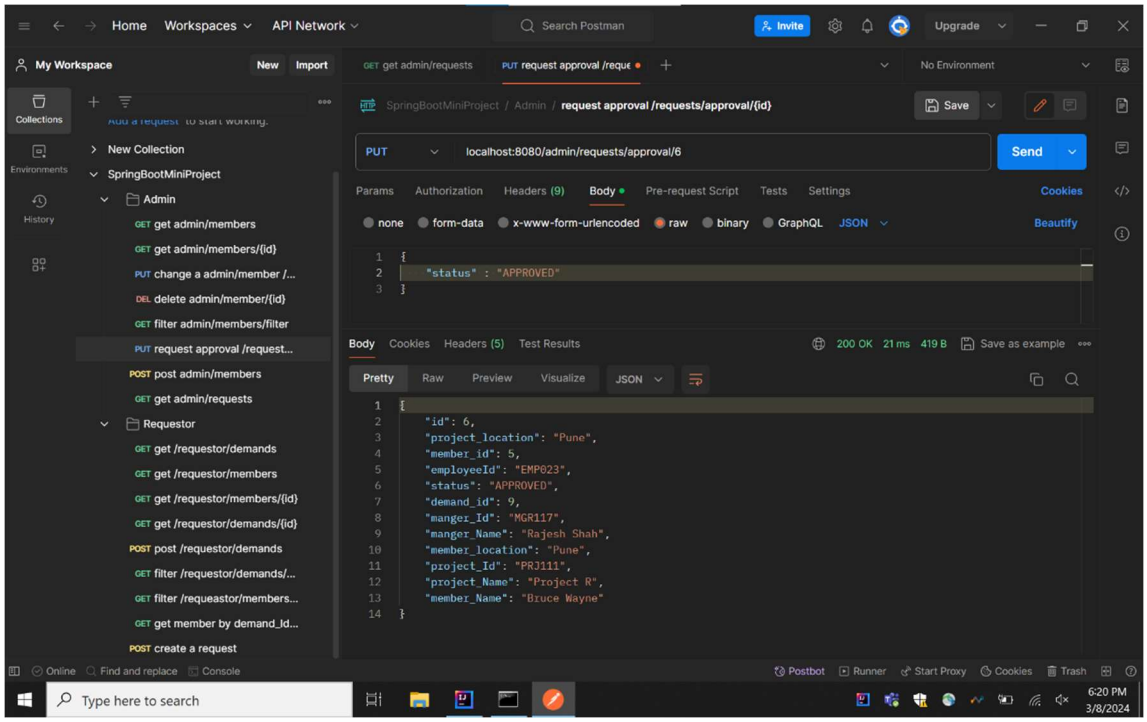
- Delete a member by id :



- Get requests :



- Approving the Request :

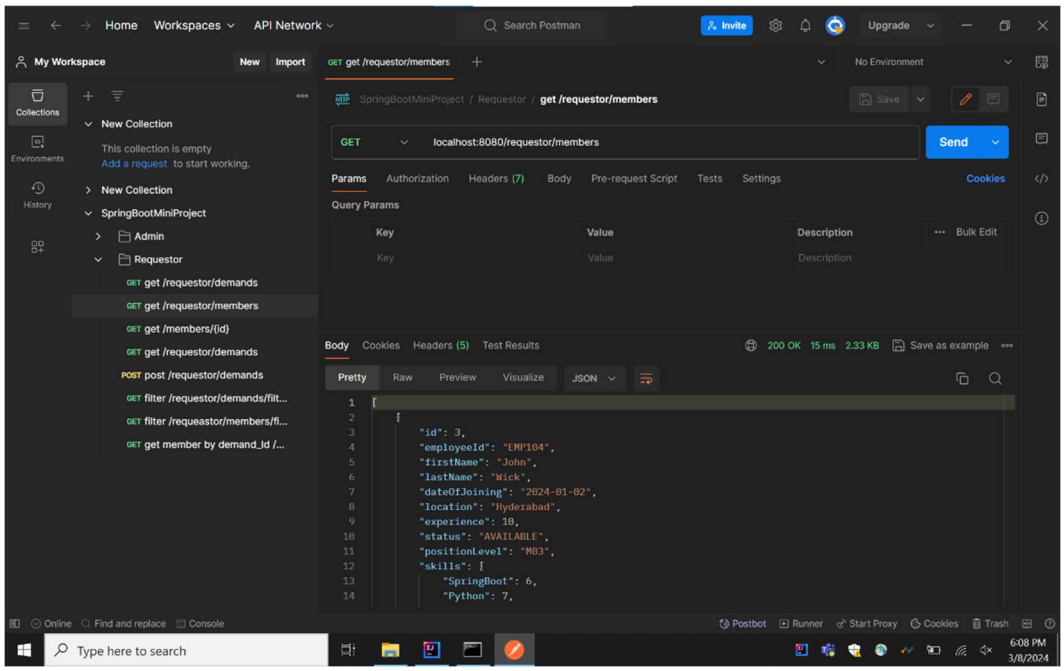


After approval changes reflected in database :

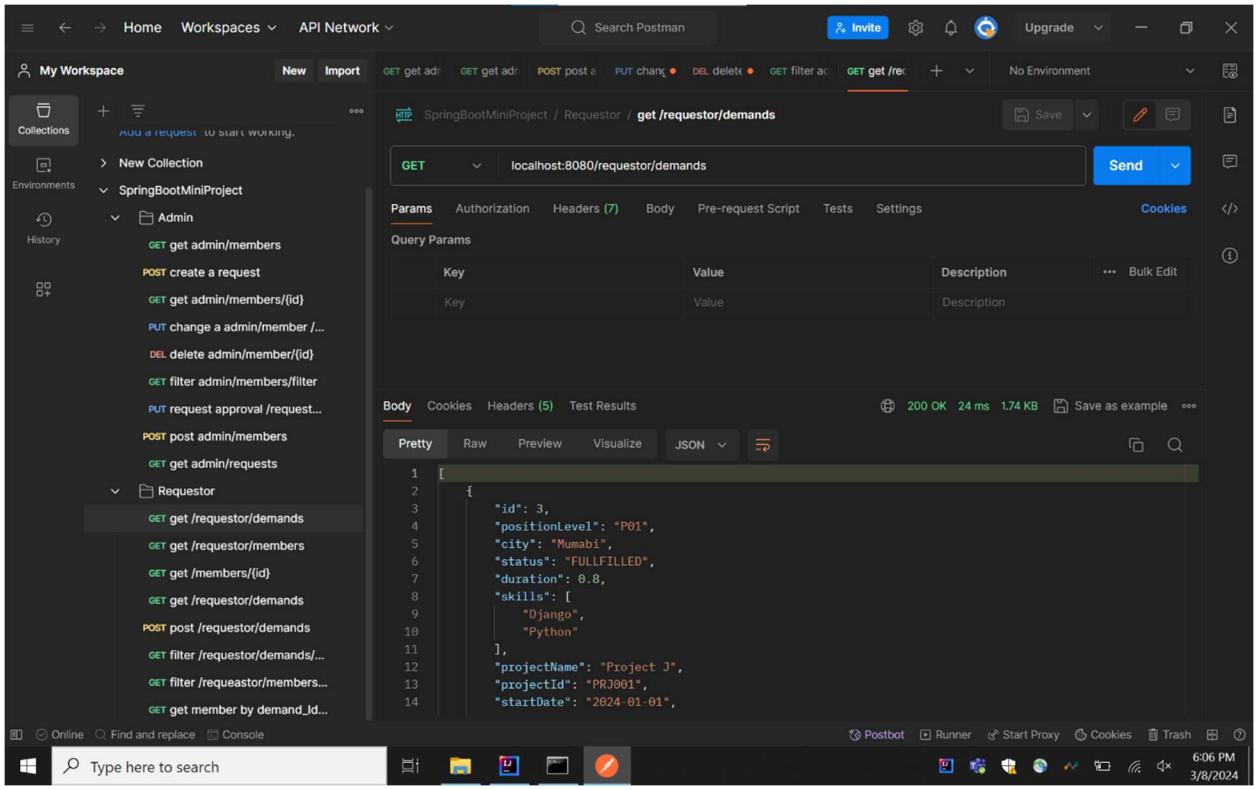
```
membership_db2=# select * from request_demands;
 id | demand_id | manger_id | manger_name | member_name | employee_location | project_id | project_name | employee_id | member_id | project_location |  status
----+----------+-----------+-------------+-------------+-------------------+------------+-------------+-------------+-----------+------------------+----------
  6 |         9 | MGR117    | Rajesh Shah | Bruce Wayne | Pune              | PRJ111     | Project R   | EMP023      |         5 | Pune             | APPROVED
(1 row)


membership_db2=# select * from demands;
 id | duration | manager_id | manager_name | project_id | project_name |                 skills                 | start_date |  city  | position_level |   status
----+---------+-----------+-------------+-----------+-------------+----------------------------------------+-----------+--------+---------------+----------
  3 |      0.8 | MGR007    | Suyog Shah   | PRJ001     | Project J    | ["Django","Python"]                    | 2024-01-01 | Mumabi | P01           | FULLFILLED
  4 |      0.2 | MGR017    | Sanket Shah  | PRJ013     | Project K    | ["Java","Python"]                      | 2024-01-01 | Pune   | PS1           | OPEN
  6 |      0.5 | MGR027    | Professor X  | PRJ011     | Project Z    | ["Django","Python","JavaScript","Java"] | 2024-02-01 | Mumabi | P03           | OPEN
  7 |      0.9 | MGR032    | Swami Pandey | PRJ081     | Project S    | ["JavaScript","Java","Scala"]          | 2024-04-02 | Pune   | E03           | OPEN
  8 |      0.8 | MGR237    | Sachin Shah  | PRJ801     | Project L    | ["React","Ruby"]                       | 2024-09-11 | Pune   | E03           | FULLFILLED
 10 |      1.5 | MGR337    | Kumar Bahnu  | PRJ141     | Project R    | ["React","CPP"]                        | 2024-03-04 | Pune   | P01           | OPEN
  9 |      1.5 | MGR117    | Rajesh Shah  | PRJ111     | Project R    | ["Dotnet","Python"]                    | 2024-010-11 | Pune   | E04           | FULLFILLED
(7 rows)


membership_db2=# select * from members;
 id |                        skills                        | date_of_joining | employee_id | experience | first_name | last_name | location  | position_level |  status
----+-----------------------------------------------------+----------------+-----------+----------+----------+----------+----------+---------------+----------
  3 | {"SpringBoot":6,"Python":7,"Angular":5,"CPP":8}      | 2024-01-02     | EMP104    |       10 | John     | Wick     | Hyderabad | M03           | AVAILABLE
  4 | {"SpringBoot":3,"Python":2,"React":5,"CPP":5,"Java":4} | 2024-03-02     | EMP004    |        5 | Clark    | Kent     | Chennai   | E03           | AVAILABLE
  2 | {"Django":2,"Python":3}                              | 2024-02-15     | EMP112    |        3 | Sumant   | Kumar    | Pune      | PS1           | ASSIGNED
  1 | {"React":2,"Java":5,"JavaScript":4}                  | 2024-05-02     | EMP162    |        5 | Rajesh   | Kumar    | Banglore  | P01           | AVAILABLE
  6 | {"SpringBoot":2}                                     | 2024-07-02     | EMP114    |        1 | John     | Patil    | Mumbai    | P01           | AVAILABLE
  7 | {"Java":2}                                           | 2024-06-02     | EMP214    |        1 | Kumari   | Yadav    | Pune      | P01           | AVAILABLE
  8 | {"Java":2,"React":3}                                 | 2024-06-12     | EMP224    |        2 | Yash     | Kumar    | Banglore  | PS1           | AVAILABLE
  9 | {"Java":2,"React":3,"SpringBoot":7,"Python":7}       | 2024-04-12     | EMP204    |       12 | Manoj    | Bhide    | Pune      | E04           | AVAILABLE
 10 | {"SpringBoot":5,"Python":2,"Ruby":5,"CPP":5,"Scala":4} | 2024-08-12     | EMP434    |       11 | Tina     | Shah     | Pune      | E03           | AVAILABLE
  5 | {"Dotnet":3,"SprinBoot":2,"Ruby":5,"Java":4}         | 2024-04-12     | EMP023    |        7 | Bruce    | Wayne    | Pune      | E04           | ASSIGNED
(10 rows)


membership_db2=#
```

Requestor:

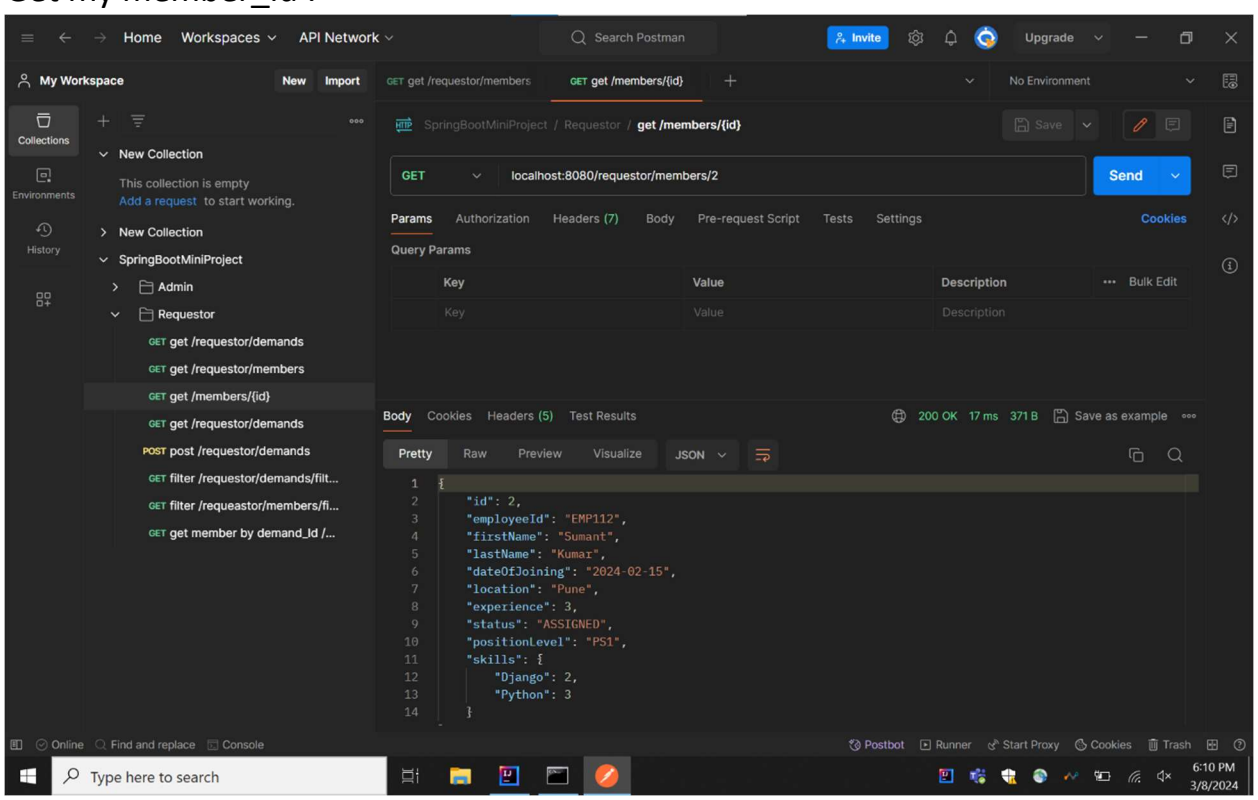
- Get members



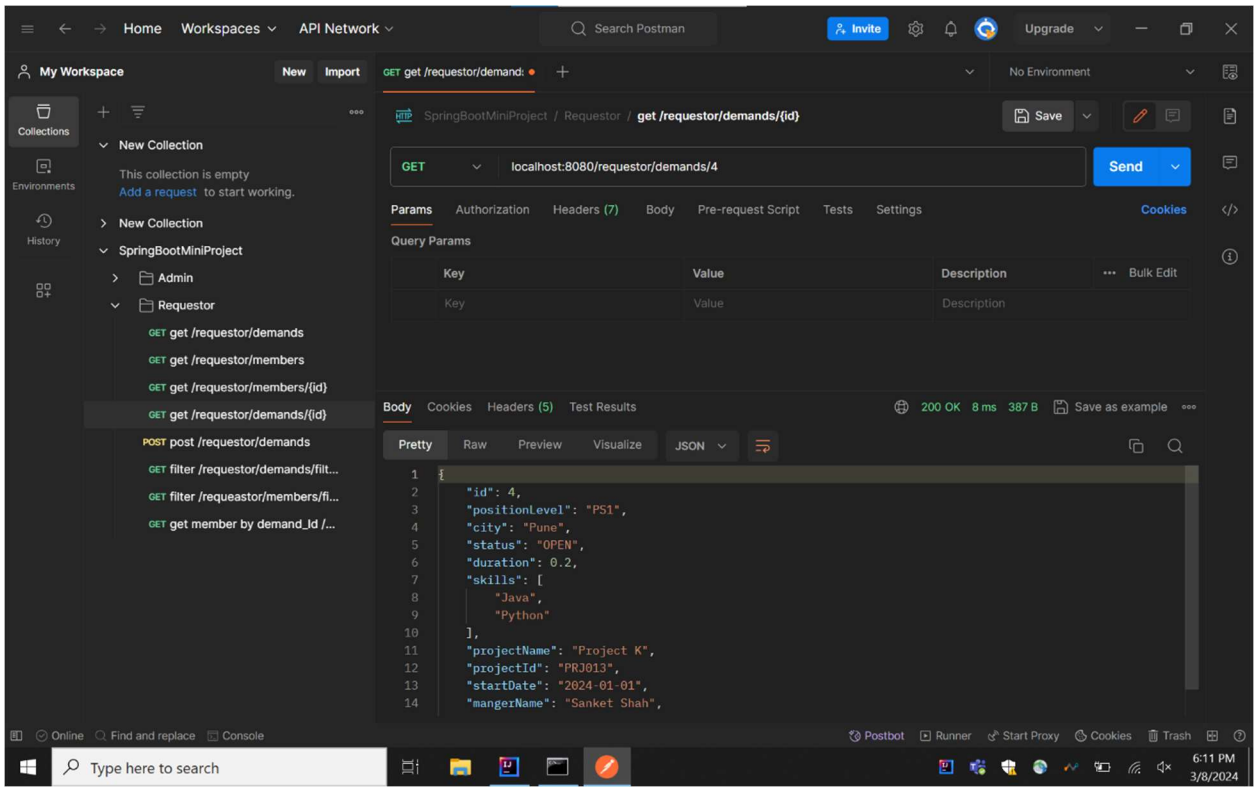
- Get Deamnds:



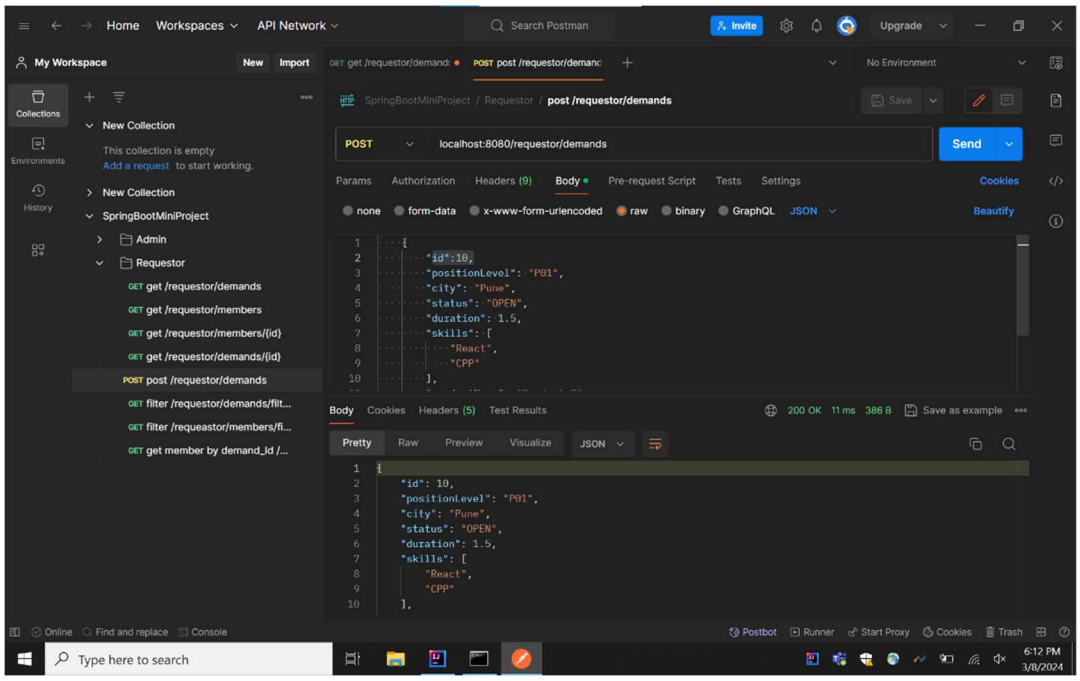
- Get my member_id :



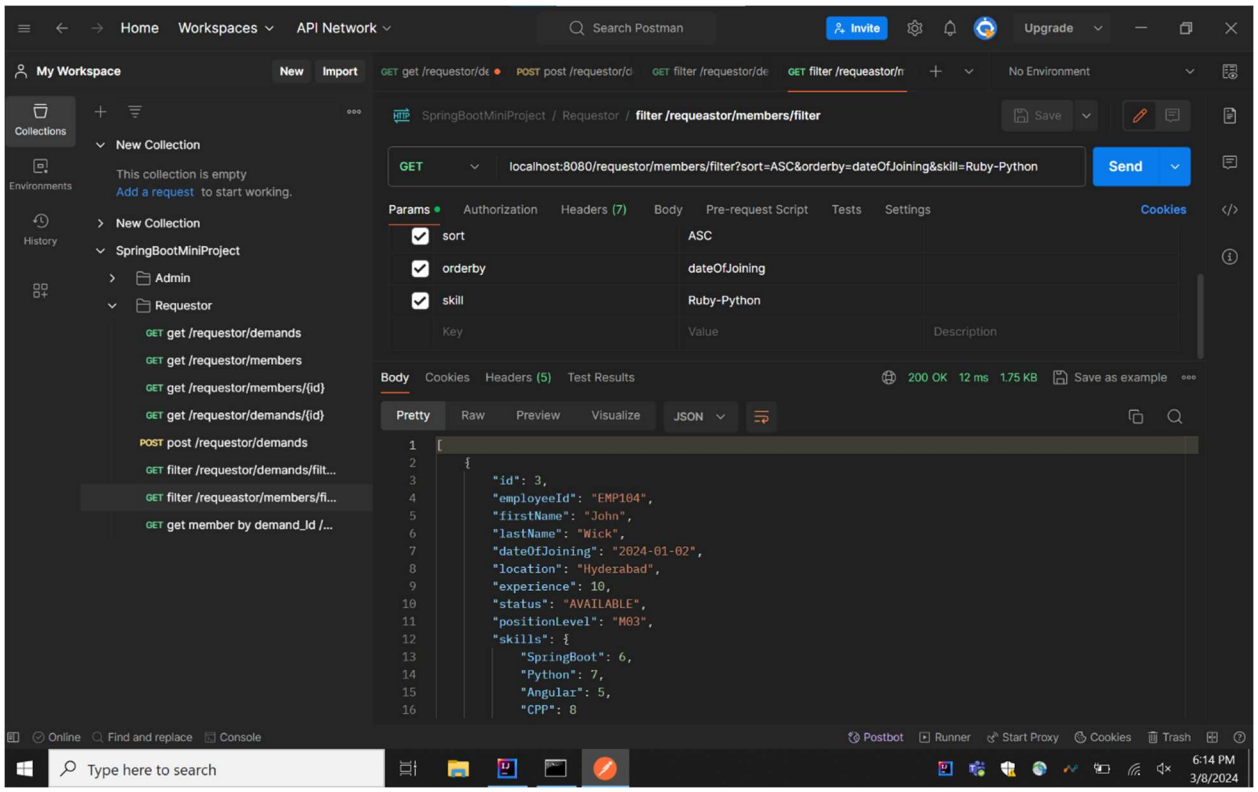
- Get my demand id:



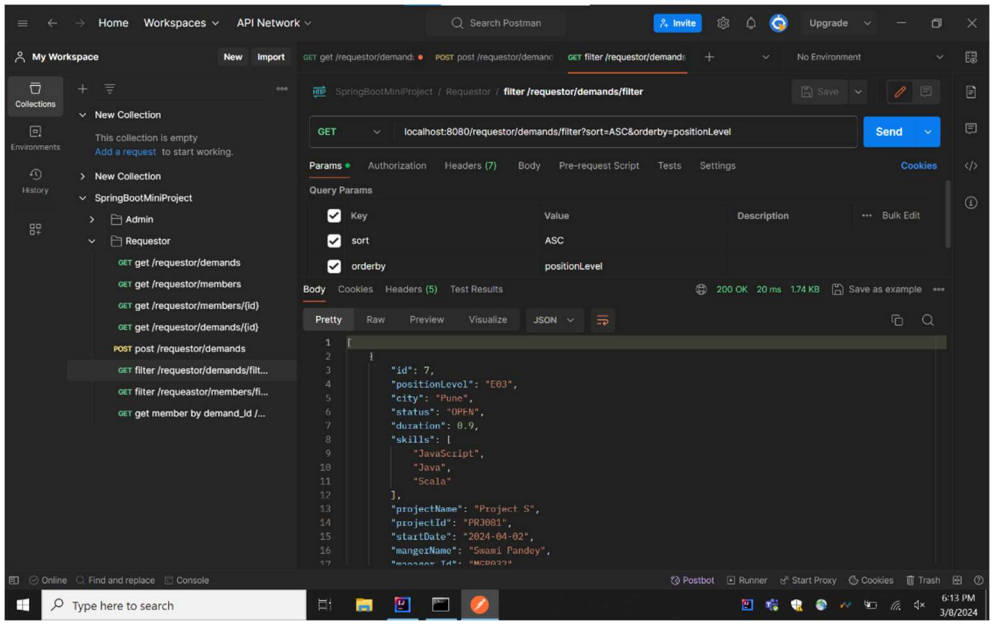
- Post a demand :



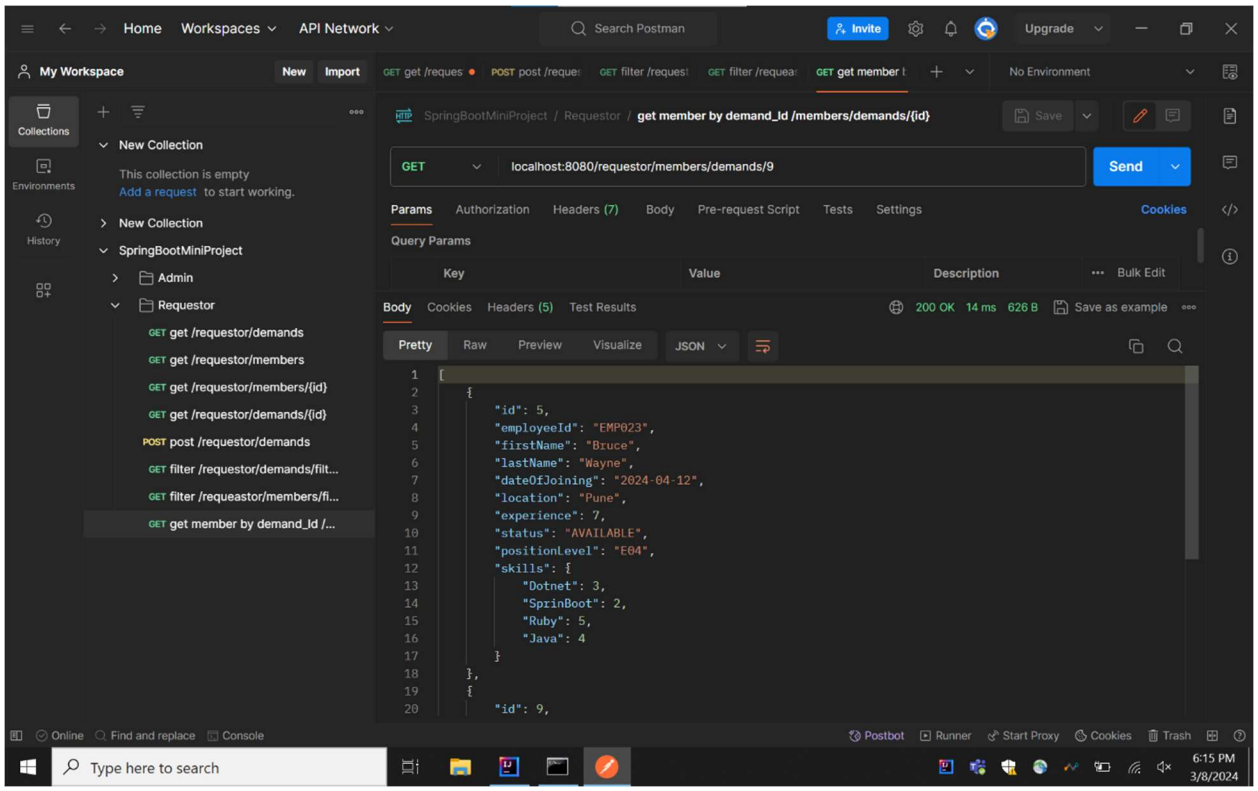
- Filter members :



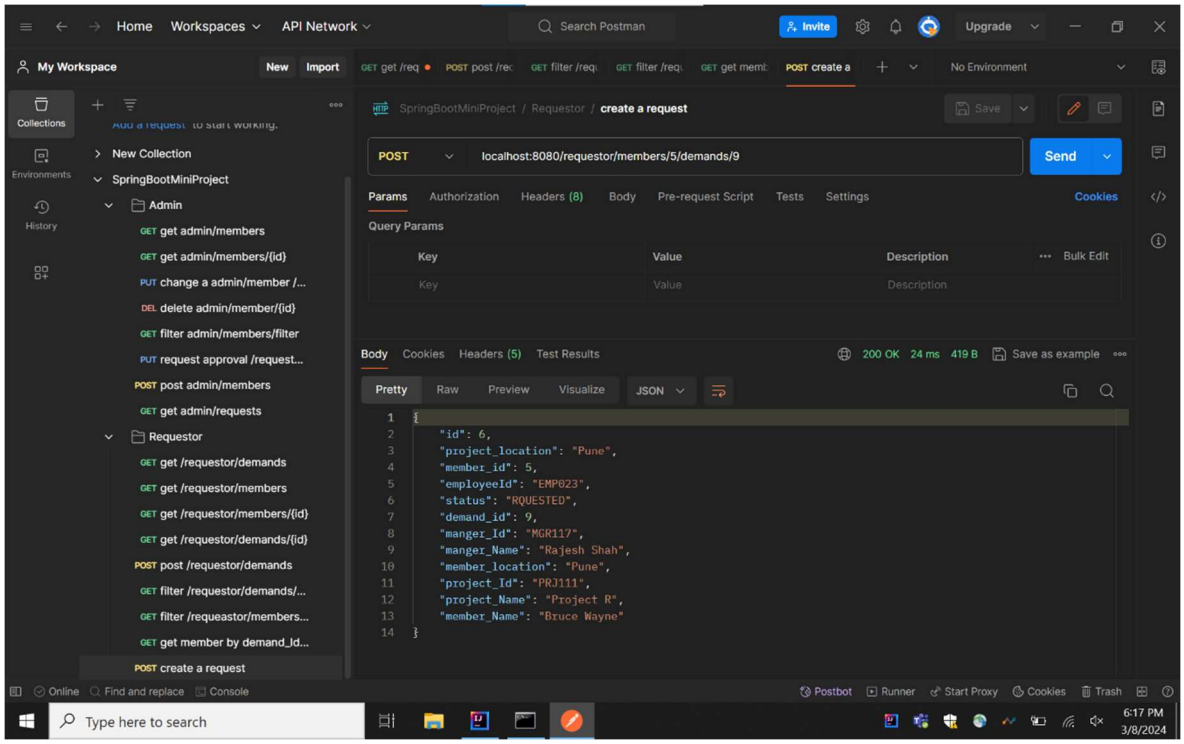
- Filter demands :



- Get member from members table by using demands from demand table :



- Create a Request :



Request in request_demand table :

```
membership_db2=# select * from request_demands;
 id | demand_id | manger_id | manger_name | member_name | employee_location | project_id | project_name | employee_id | member_id | project_location |  status
----+-----------+-----------+-------------+-------------+-------------------+------------+--------------+-------------+-----------+------------------+----------
  6 |         9 | MGR117    | Rajesh Shah | Bruce Wayne | Pune              | PRJ111     | Project R    | EMP023      |         5 | Pune             | RQUESTED
(1 row)
```