Question-

You are required to create an Employee Management Rest API based Web application, where you will be developing CRUD (Create, Read, Update and Delete) functionality along with Sorting and some concepts of security.

Your Rest API should be secure. And should have different endpoints for different operations-

- Get Role Details
- Add a new Role
- Get User Details
- Add a new User
- Save a new Employee
- Read Employee List
- Fetch Employee Data BY Id
- Update an existing Employee
- Delete an existing Employee
- Search Employee BY First Name
- Sort Employee BY First Name IN ASC / DESC Order

Test Scenarios:

Rest API endpoints for different operations:

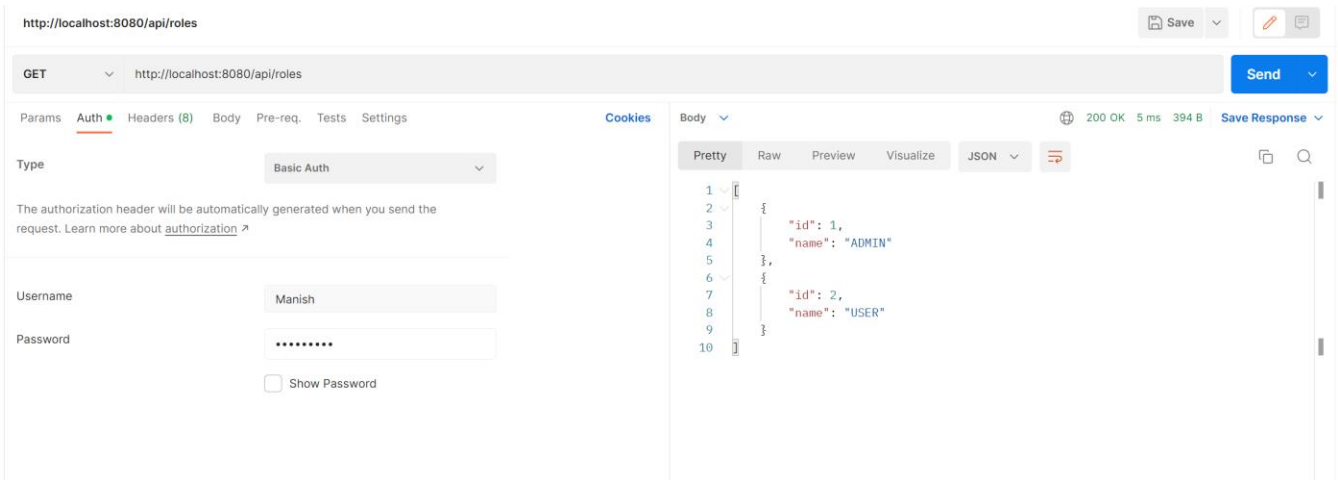
S. NO	Operation	Endpoints	Method	Access
1	Get Role Details	/api/roles	GET	ADMIN
2	Add a new Role	/api/add/role	POST	ADMIN
3	Get User Details	/api/users	GET	ADMIN
4	Add a new User	/api/add/user	POST	ADMIN
5	Save a new Employee	/api/employee	POST	ADMIN
6	Read Employee List	/api/employees	GET	USER, ADMIN
7	Fetch Employee Data BY Id	/api/employees/{employeeId}	GET	USER, ADMIN
8	Update an existing Employee	/api/employee	PUT	ADMIN
9	Delete an existing Employee	/api/employee/{employeeId}	DELETE	ADMIN
10	Search Employee BY First Name	/api/employees/search/{employeeFirstName}	GET	USER, ADMIN
11	Sort Employee BY First Name IN ASC / DESC Order	/api/employees/sort?order=(**ASC/DESC**)	GET	USER, ADMIN

Please use below test credentials for Admin and User roles:

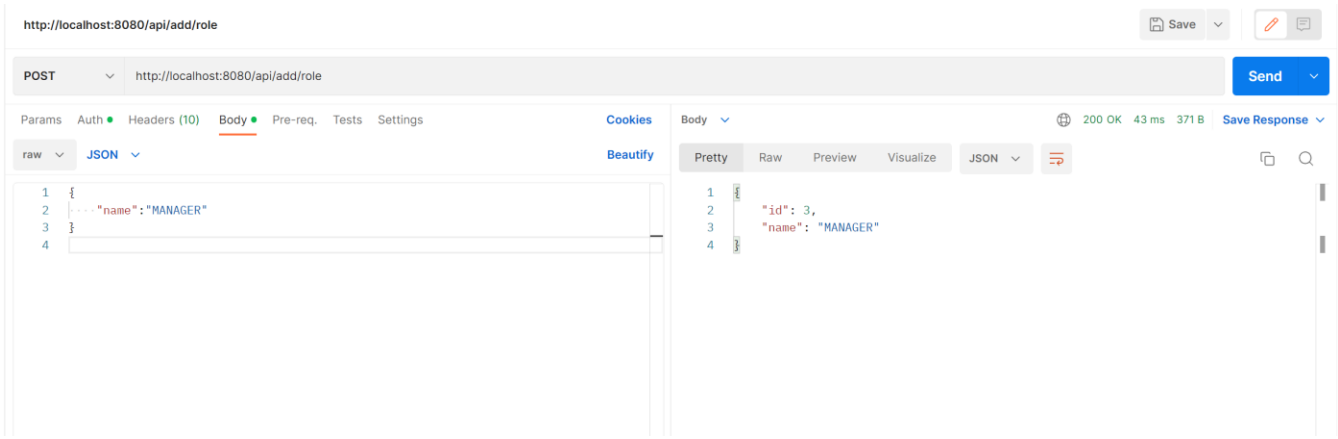
Admin:
Username: Manish
Password: admin@123

Test User:
Username: Shyam
Password: user@123

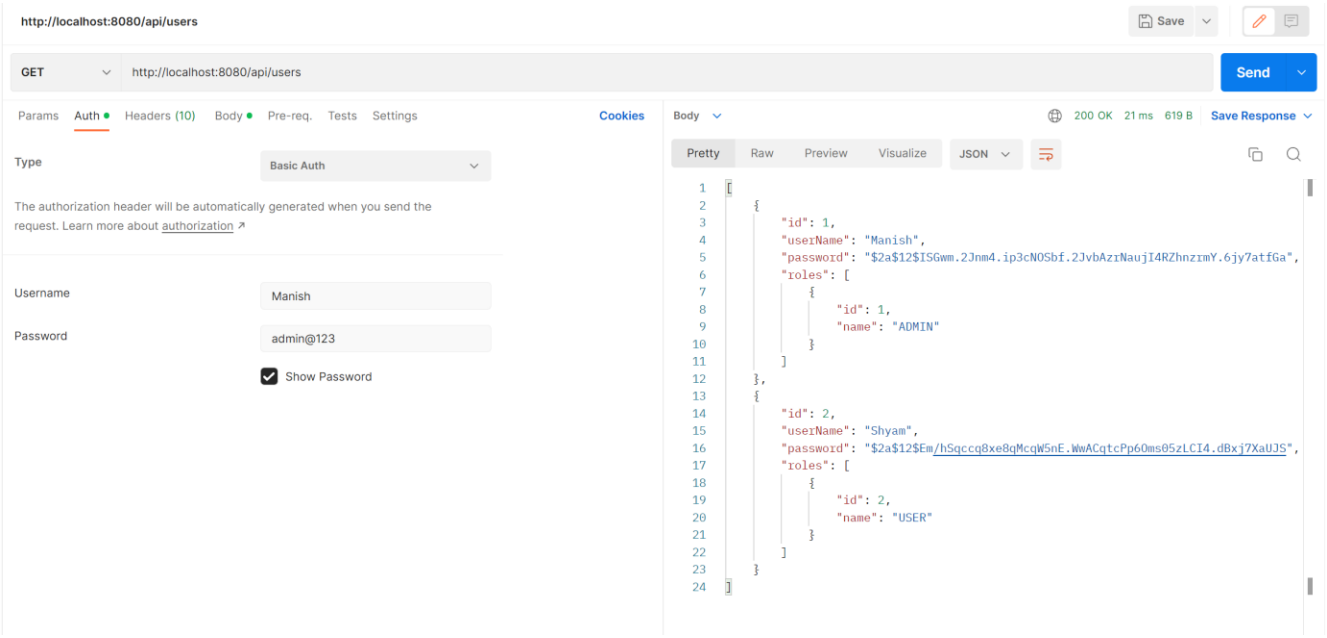
Test Scenario #1: Get Role Details - /api/roles



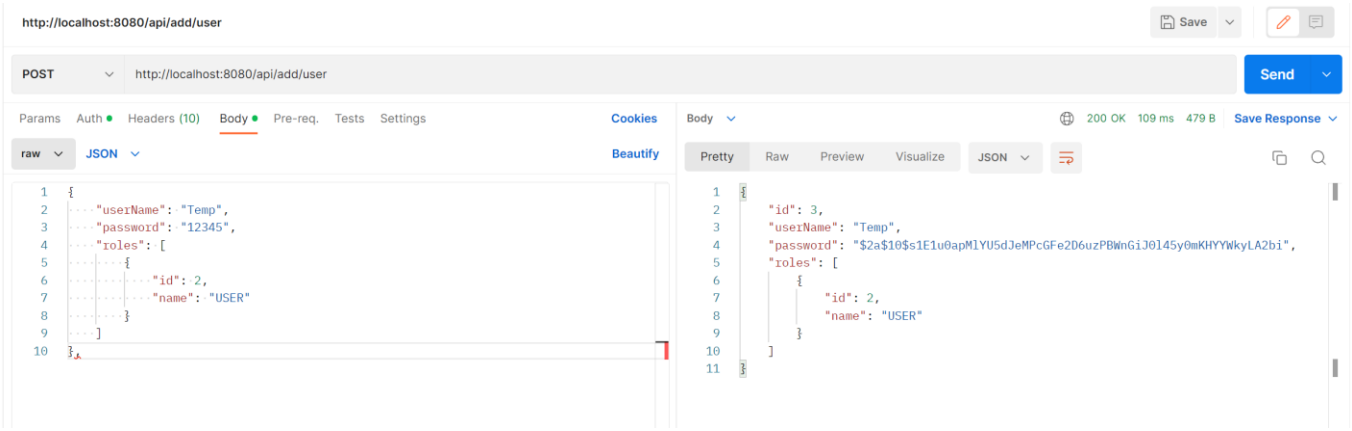
Test Scenario #2: Add a new Role - /api/add/role



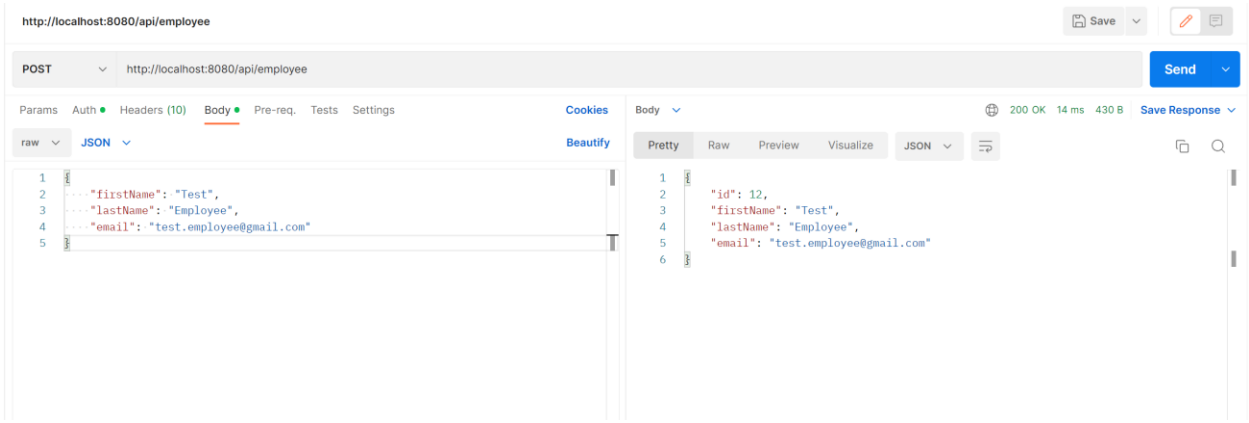
Test Scenario #3: Get User Details - /api/users



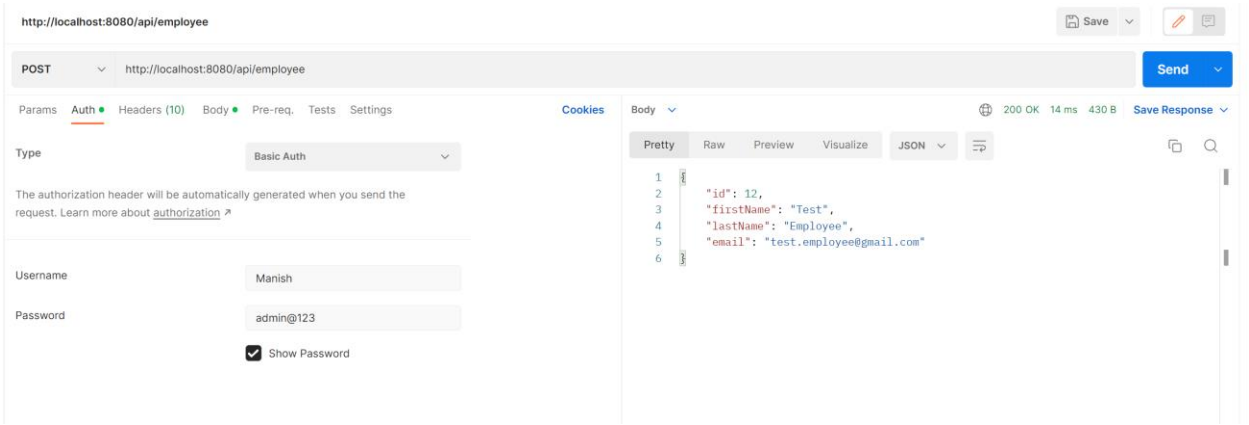
Test Scenario #4: Add a new User - /api/add/user



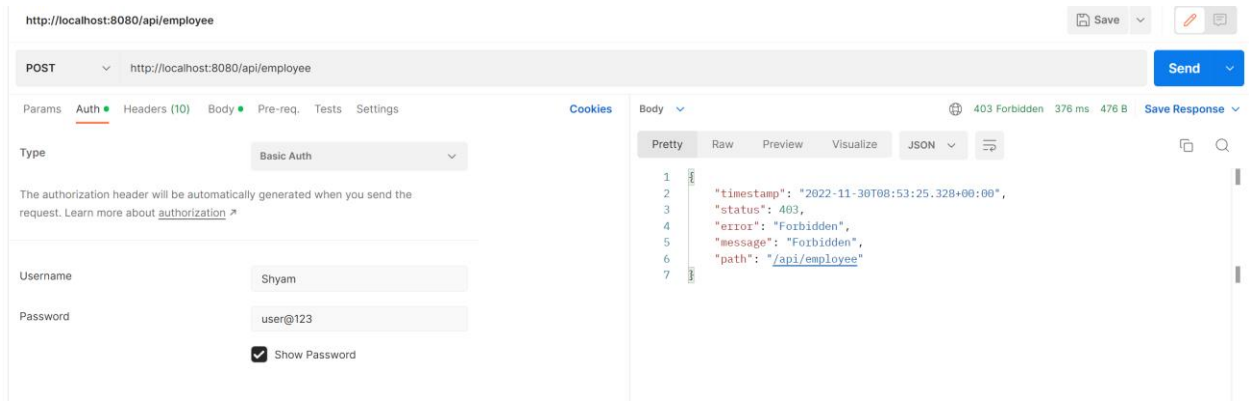
Test Scenario #5: Save a new Employee - /api/employee



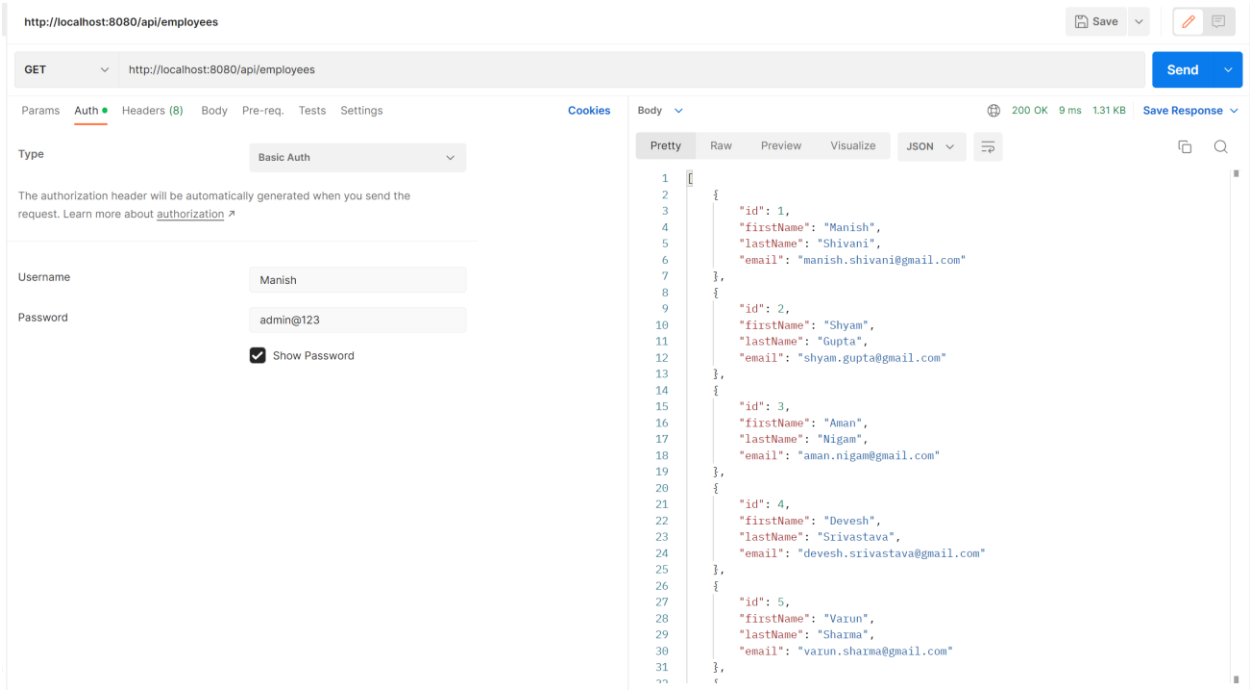
Added above new employee – “Test Employee” using Admin Credentials

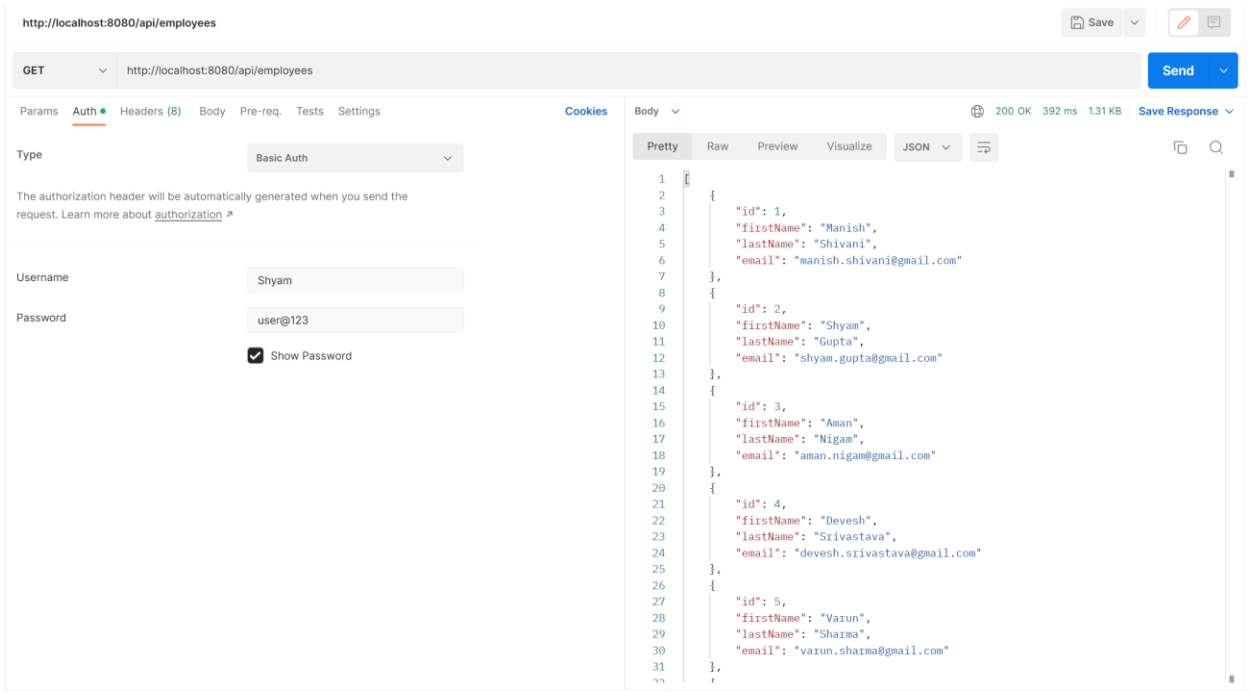


Try adding new employee – “Test1234 Employee” using User Credentials, we receive Status 403 “Forbidden”



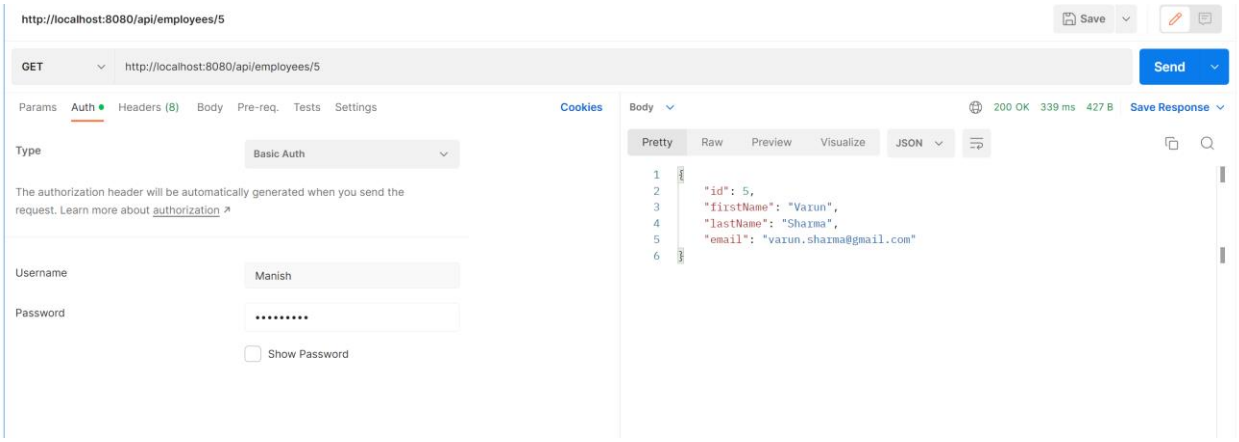
Test Scenario #6: Read Employee List - /api/employees





Test Scenario #7: Fetch Employee Data BY Id - /api/employees/{employeeId}

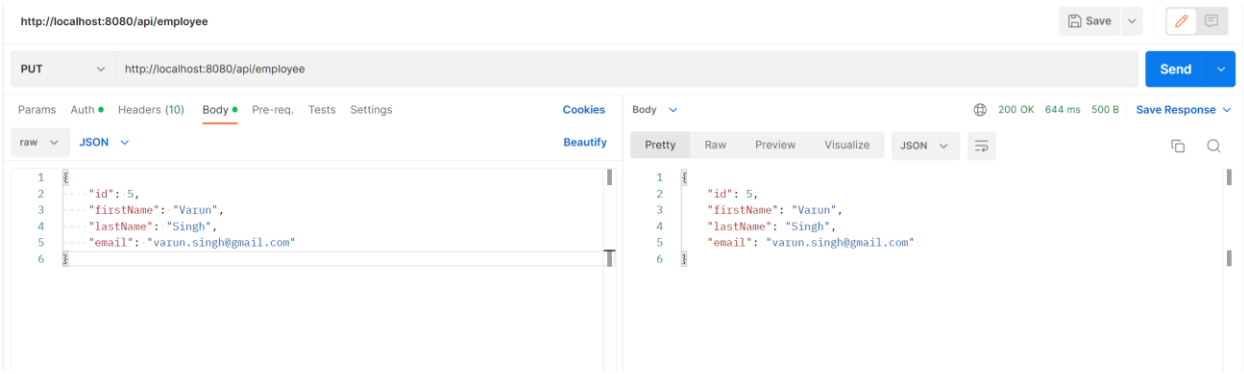
Fetch Employee ID 5:



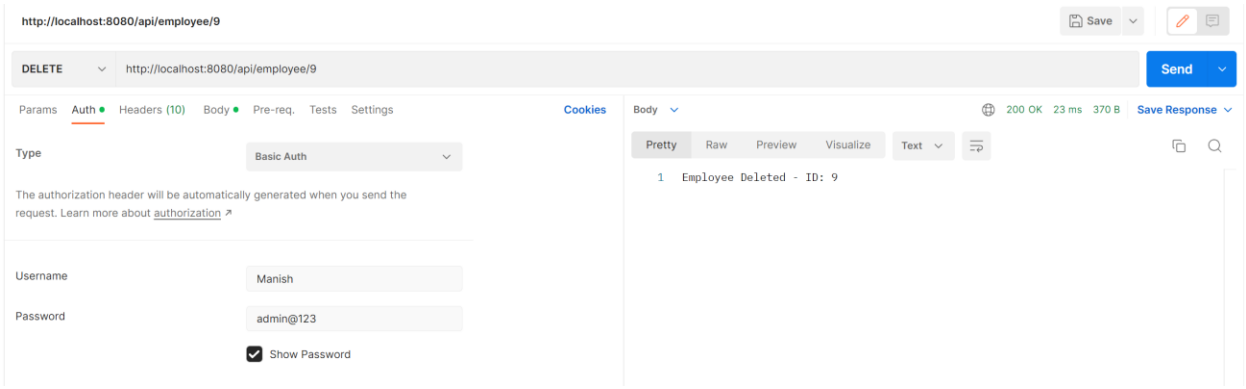
Test Scenario #8: Update an existing Employee - /api/employee

For the employee id #5, fetched in Test Scenario 7:

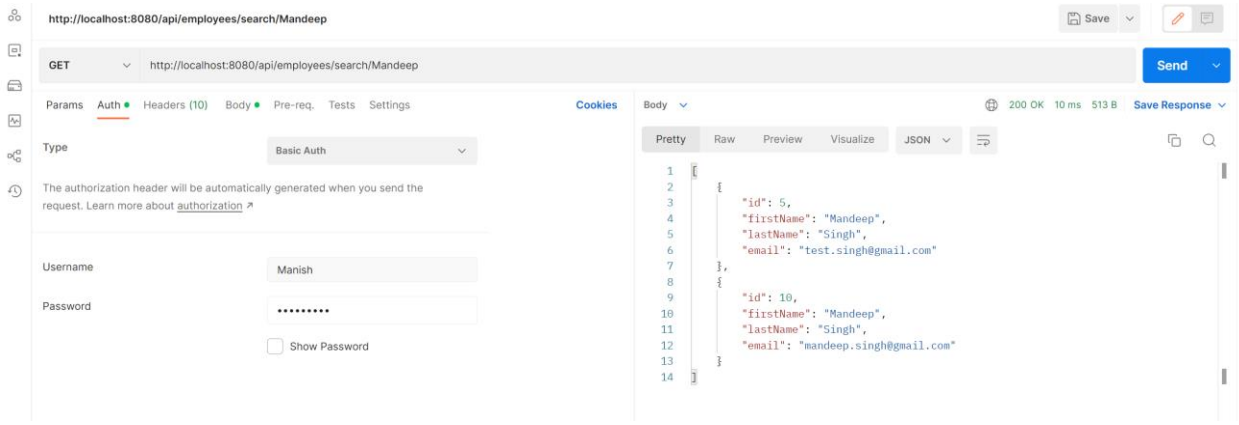
- Update last name from “Sharma” to “Singh”
- Update email from `"varun.sharma@gmail.com"` to `"varun.singh@gmail.com"`



Test Scenario #9: Delete an existing Employee - /api/employee/{employeeId}

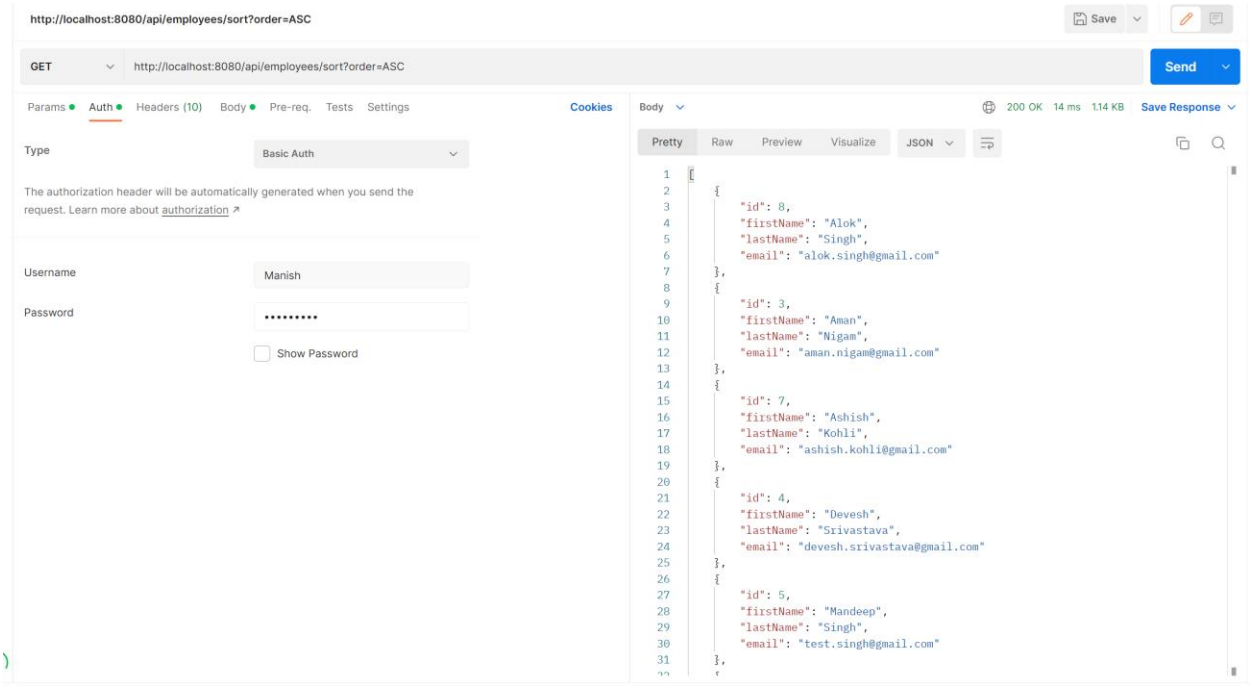


Test Scenario #10: Search Emp BY First Name - /api/employees/search/{employeeFirstName}



Test Scenario #11: Sort Employee BY First Name IN ASC / DESC Order
/api/employees/sort?order=(ASC/DESC)

ASC:



DESC: