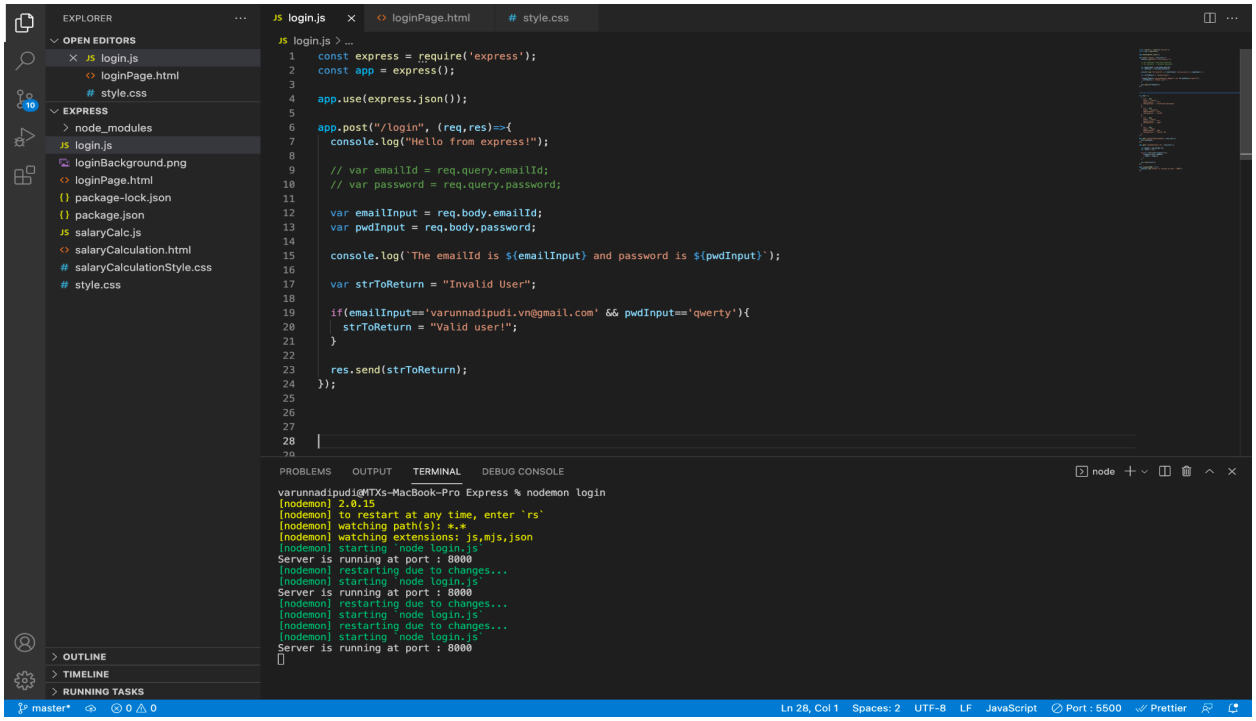
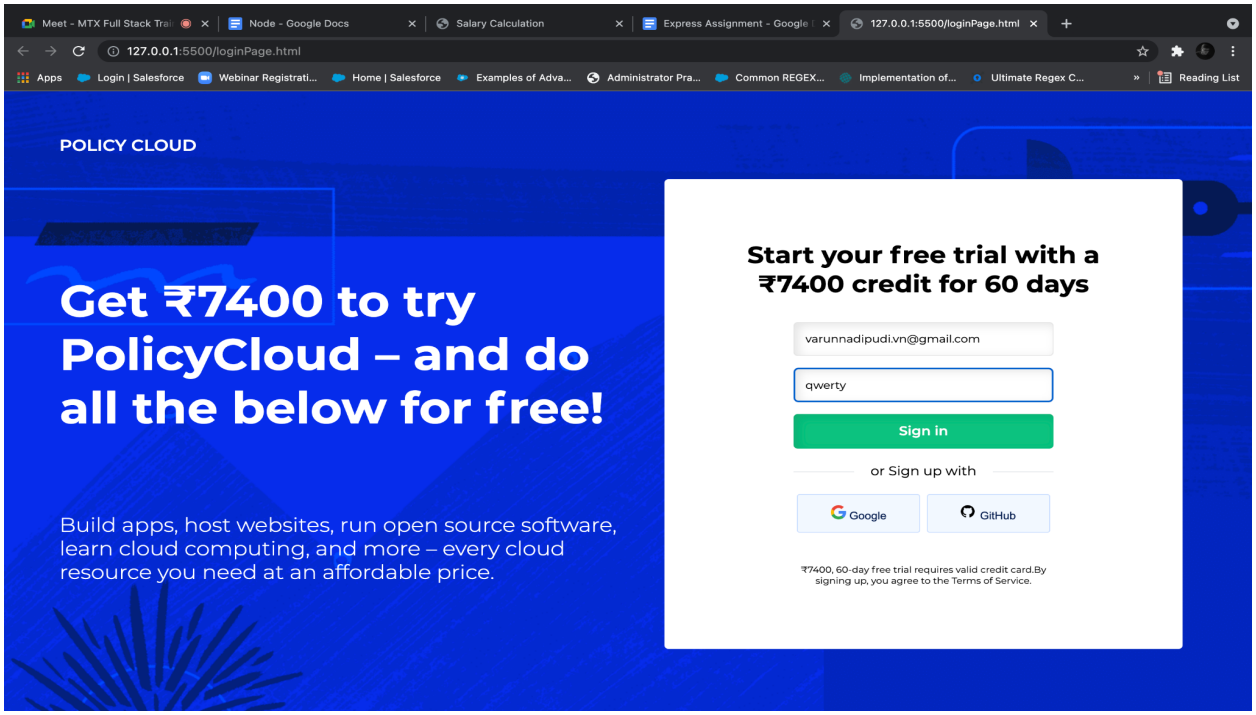
1. Write the Login.html which takes 2 inputs as user id and password. At the express side, validate it against your name for user id and password as admin. If this criterion matches then return as valid user else returns as an invalid user.

The Login.js file contains the method where the request 'http://localhost:8000/login' is answered.

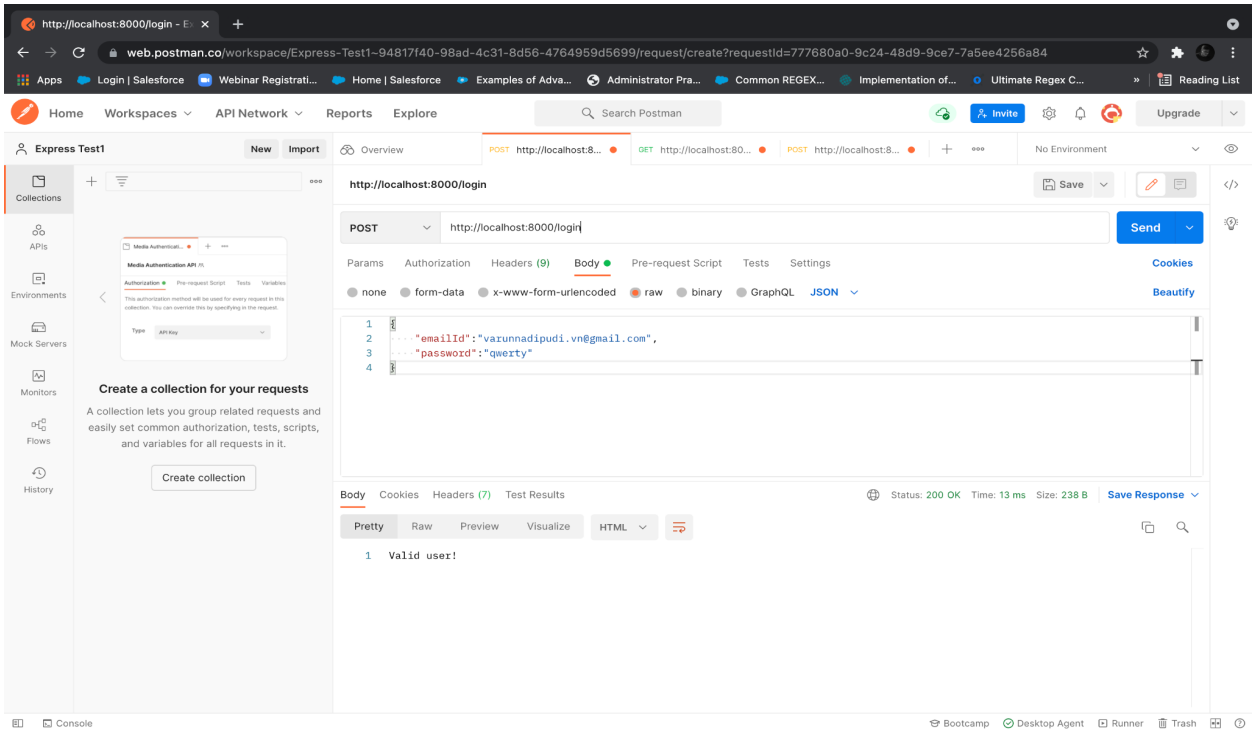


Browser login details

(Since we are using the post method the browser doesn't support it, so using Postman to show the output after clicking Sign in **)**

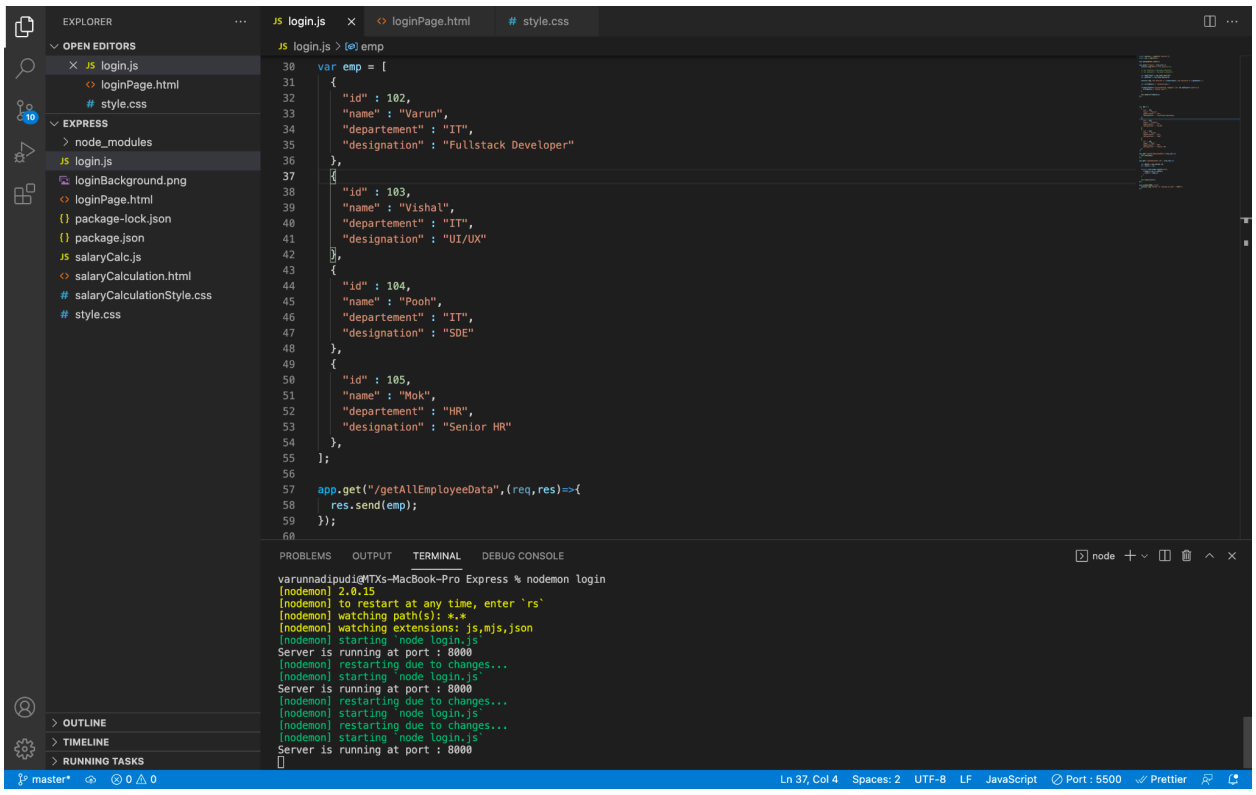


Postman output

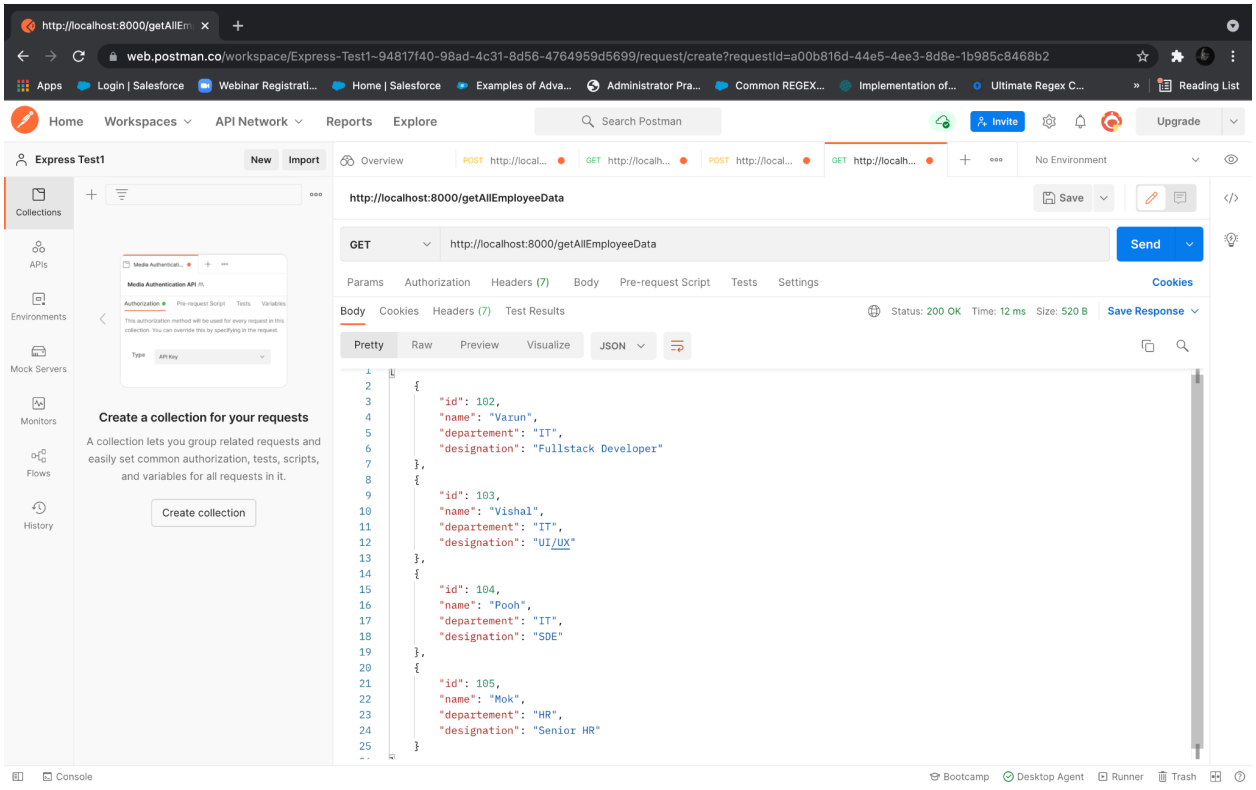


2. Write an express script to take the URL as **'/getAllEmployeeData**' and when the browser gives the URL 'http://localhost:8000/getAllEmployeeData' an array of Employee objects should be returned from express and it should be displayed on the browser.
Employee object should have the fields: Id, name, dept, and designation.

This is the method in login.js file which will answer the **'/getAllEmployeeData**' request.

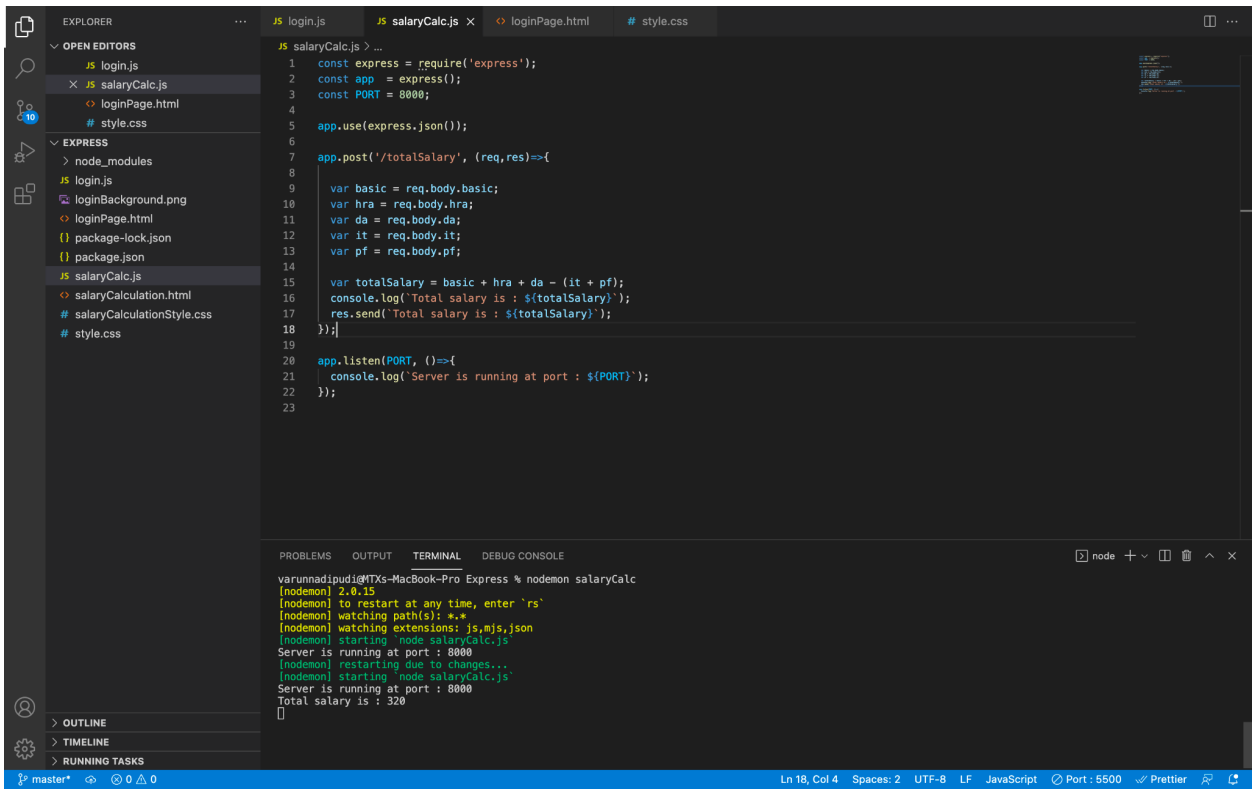


Postman output



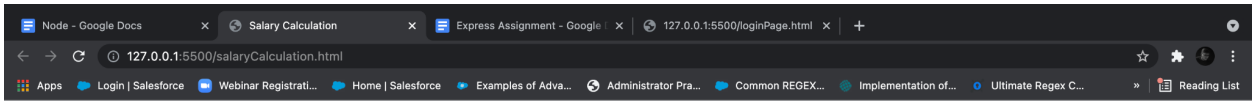
3. Create a post method in express which takes these parameters Basic, HRA, DA, IT, and PF and this function calculates the total salary as total_salary = Basic + HRA + DA - (IT + PF) and displays the total_salary as the result of this function.

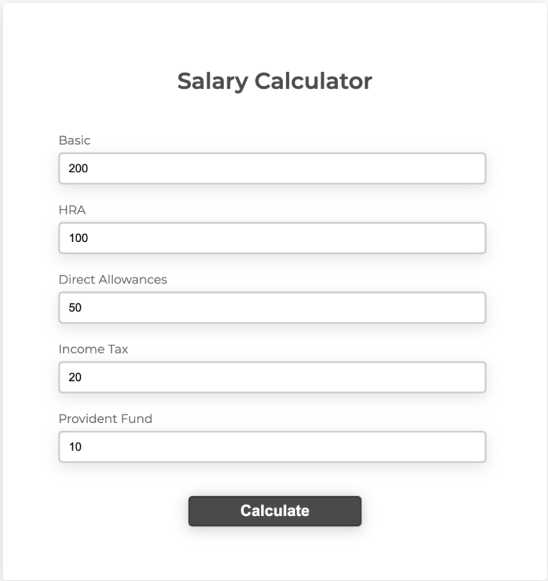
The salaryCalc.js method answers the 'http://localhost:8000/totalSalary' request



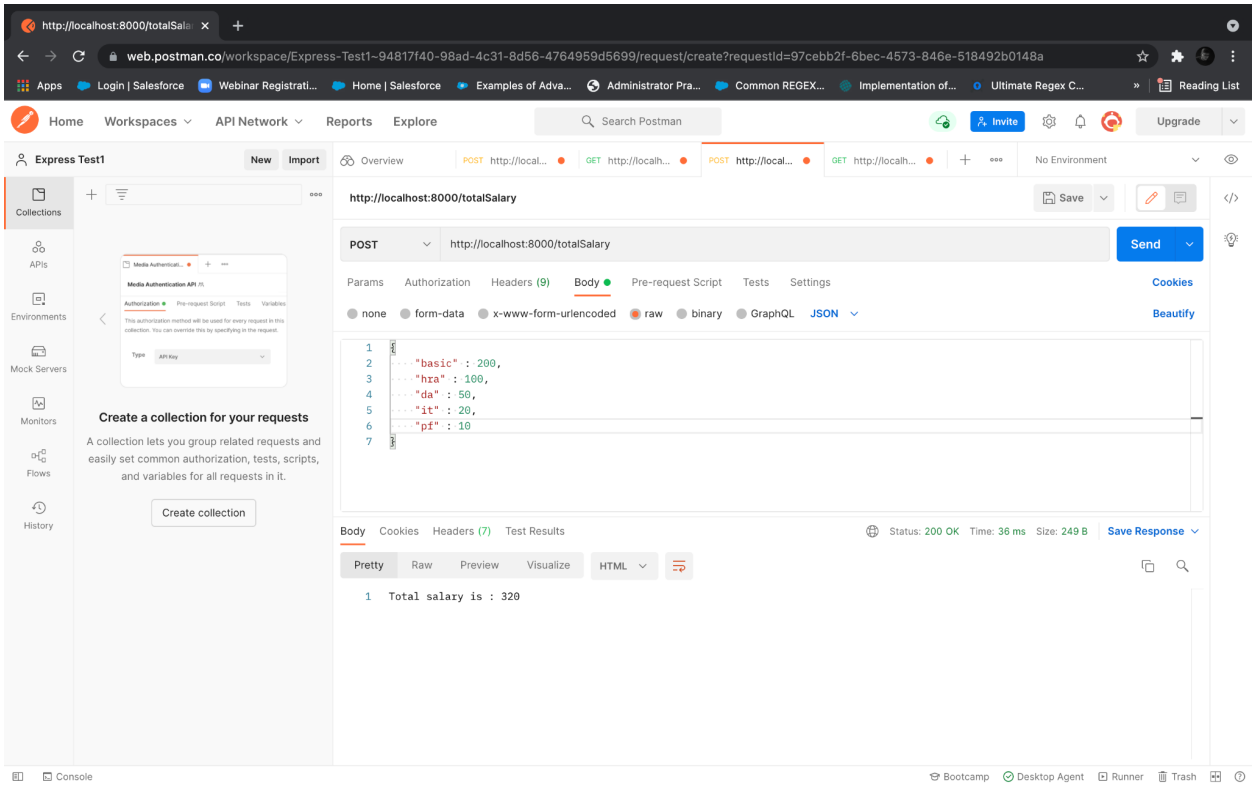
Browser input details

(Since we are using the post method the browser doesn't support it, so using Postman to show the output after clicking calculate **)**





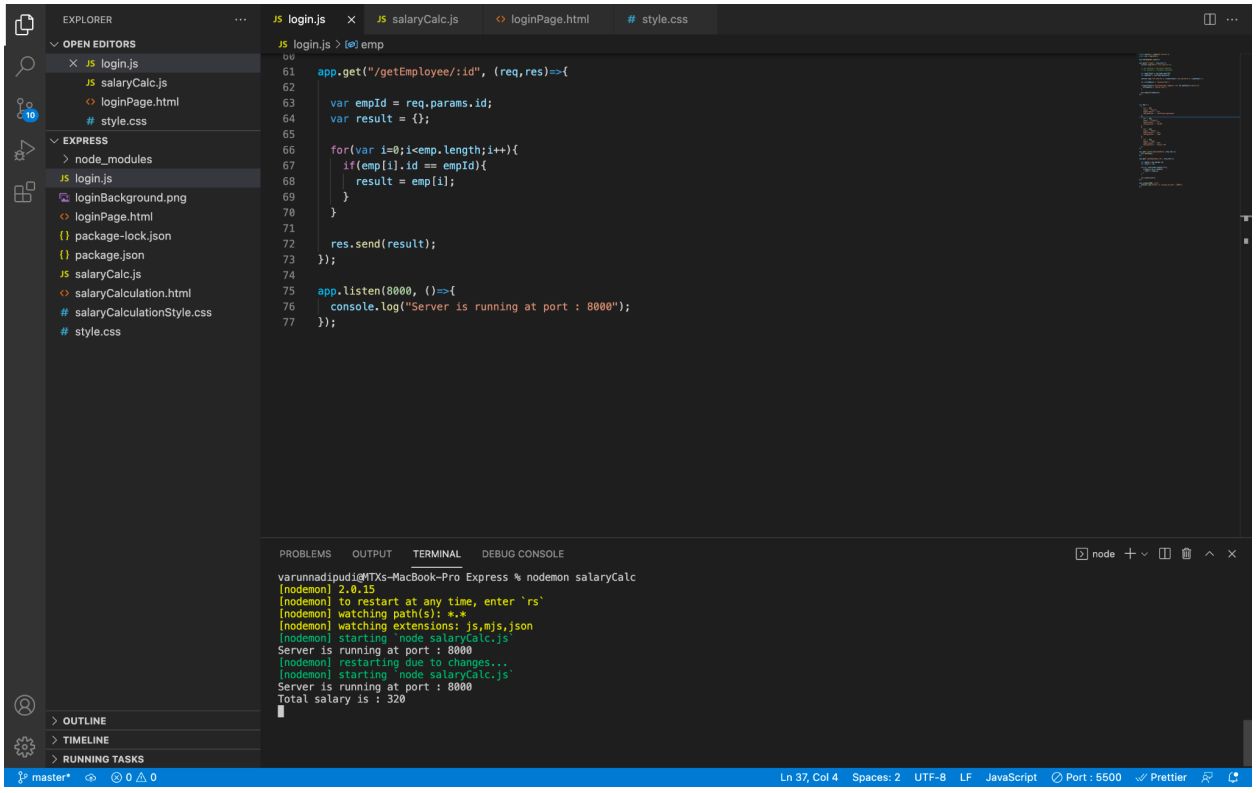
Postman Output

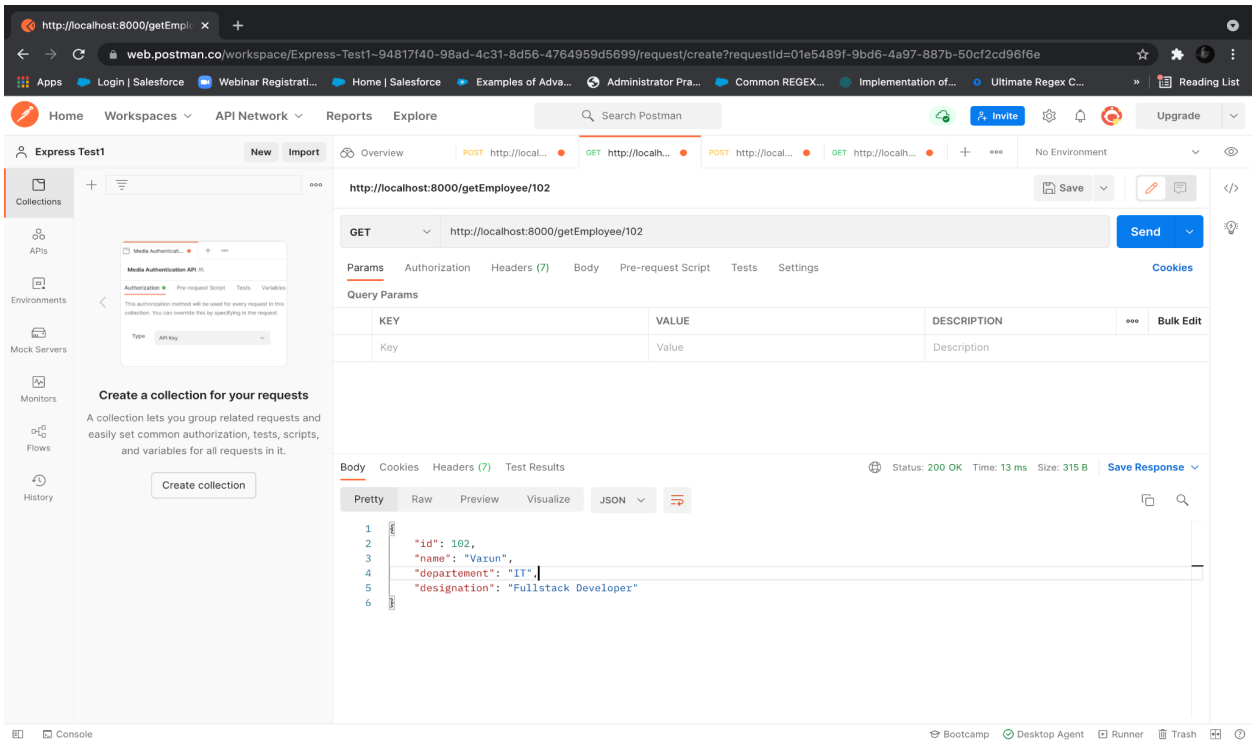


4. Create an array of objects of employees with the fields id, name, dept, and designation and create below-given functions in the Express and perform the actions as specified.

a. get - getAllEmployees() -- Returns all the employees of the array of objects.
**********Already done in question 2.**********

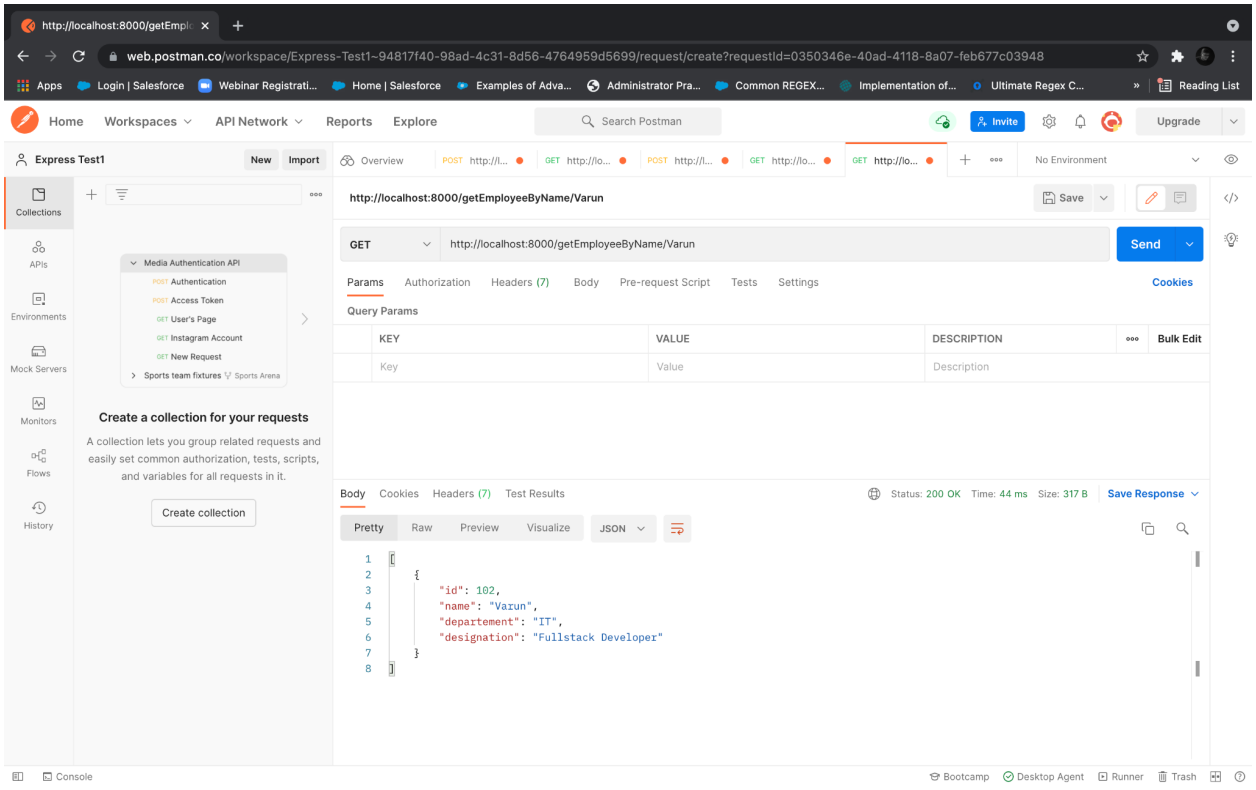
b.get - getEmployeeById() -- Takes the empid as the path param and displays employee record based on given Id.





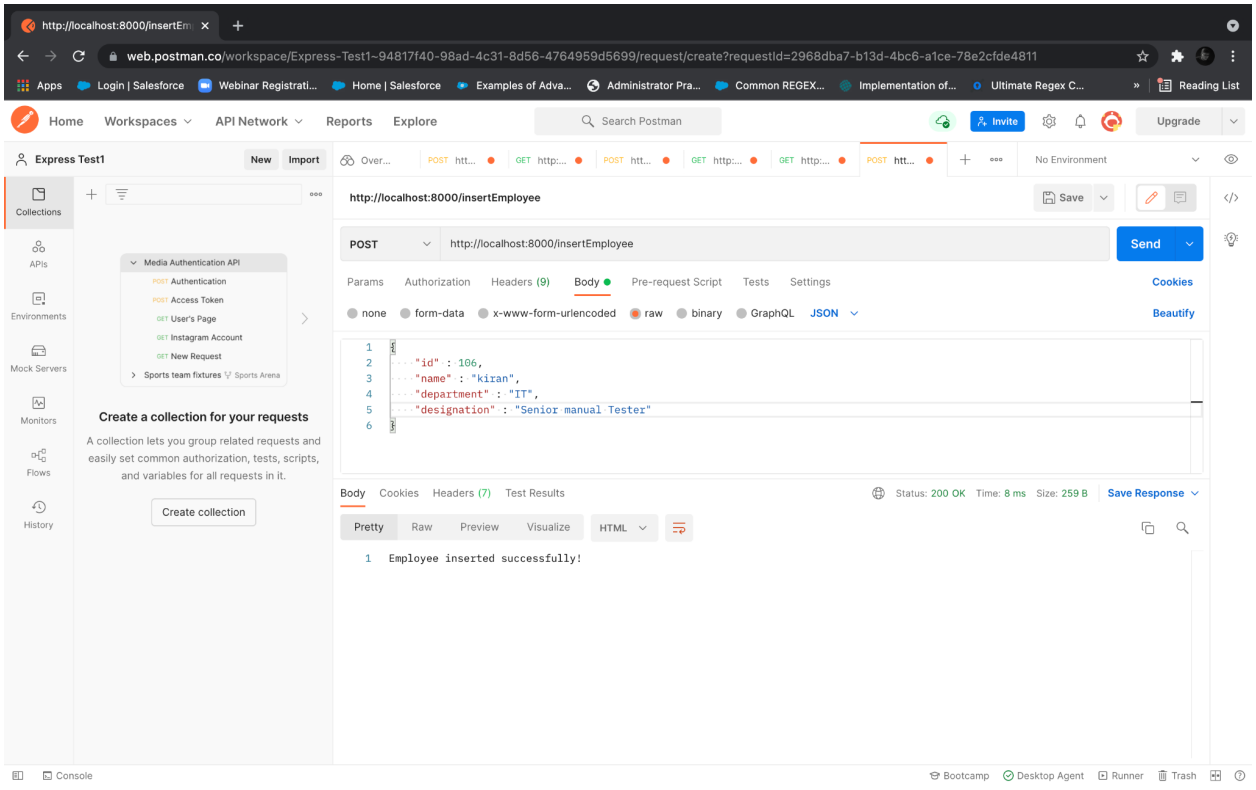
c. get - getEmployeeByName() -- Returns all the matching records with the given name.

```
app.get("/getEmployeeByName/:name", (req,res)=>{

  var employee = emp.filter( e => e.name == req.params.name);

  res.send(employee);
});
```

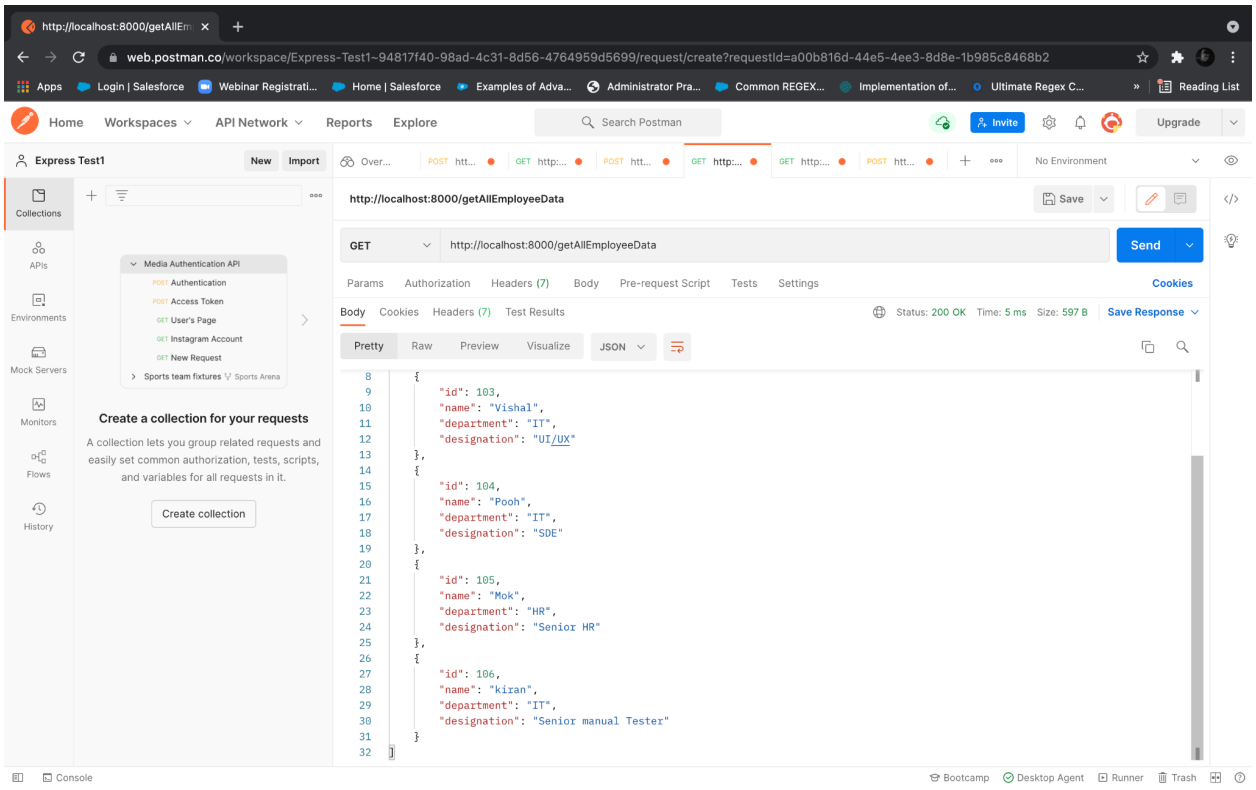


d. post - insertEmployeeData() -- Takes an object from input and inserts into the array of objects and returns the string "Record inserted successfully" or "Unable to insert the record". After this operation, verify in postman that the given record is inserted by giving 'http://localhost:8001/getAllEmployees'

```
//inserting Employee
app.post("/insertEmployee", (req,res)=>{

  var employee = {
    "id" : req.body.id,
    "name" : req.body.name,
    "department" : req.body.department,
    "designation" : req.body.designation
  };
  emp.push(employee);

  res.send("Employee inserted successfully!");

});
```

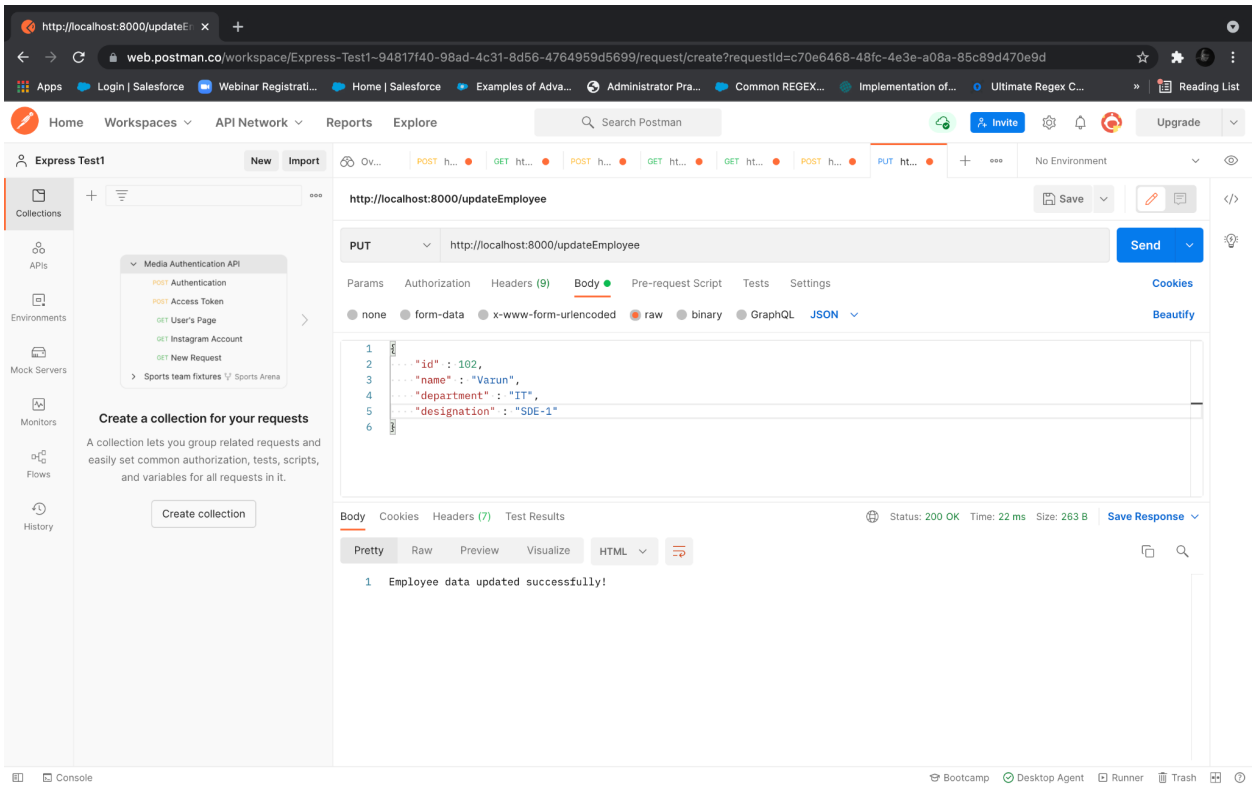


Verifying whether the data is added into emp object array using 'http://localhost:8000/getAllEmployeeData' request

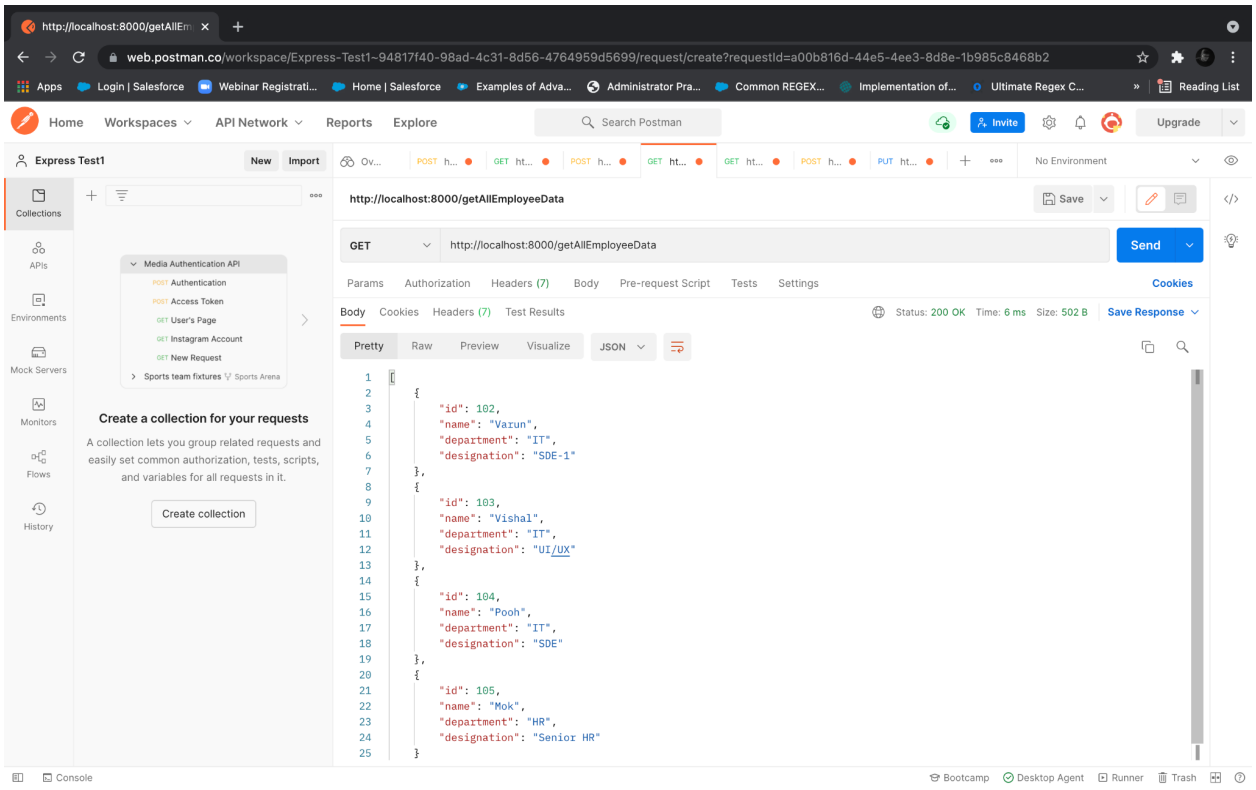


e. put - updateEmployeeData() -- Takes an object and updates that record in an array of objects. After this operation, verify in postman that the given record is updated by giving 'http://localhost:8000/getAllEmployeeData' request

```
//updating the Employee data with data from put request to updateEmployee
app.put('/updateEmployee', (req,res)=>{
  var employeeId = req.body.id;
  for(var i=0;i<emp.length;i++){
    if(emp[i].id == employeeId){
      emp[i].id = req.body.id;
      emp[i].name = req.body.name;
      emp[i].department = req.body.department;
      emp[i].designation = req.body.designation;
    }
  }

  res.send("Employee data updated successfully!");

});
```

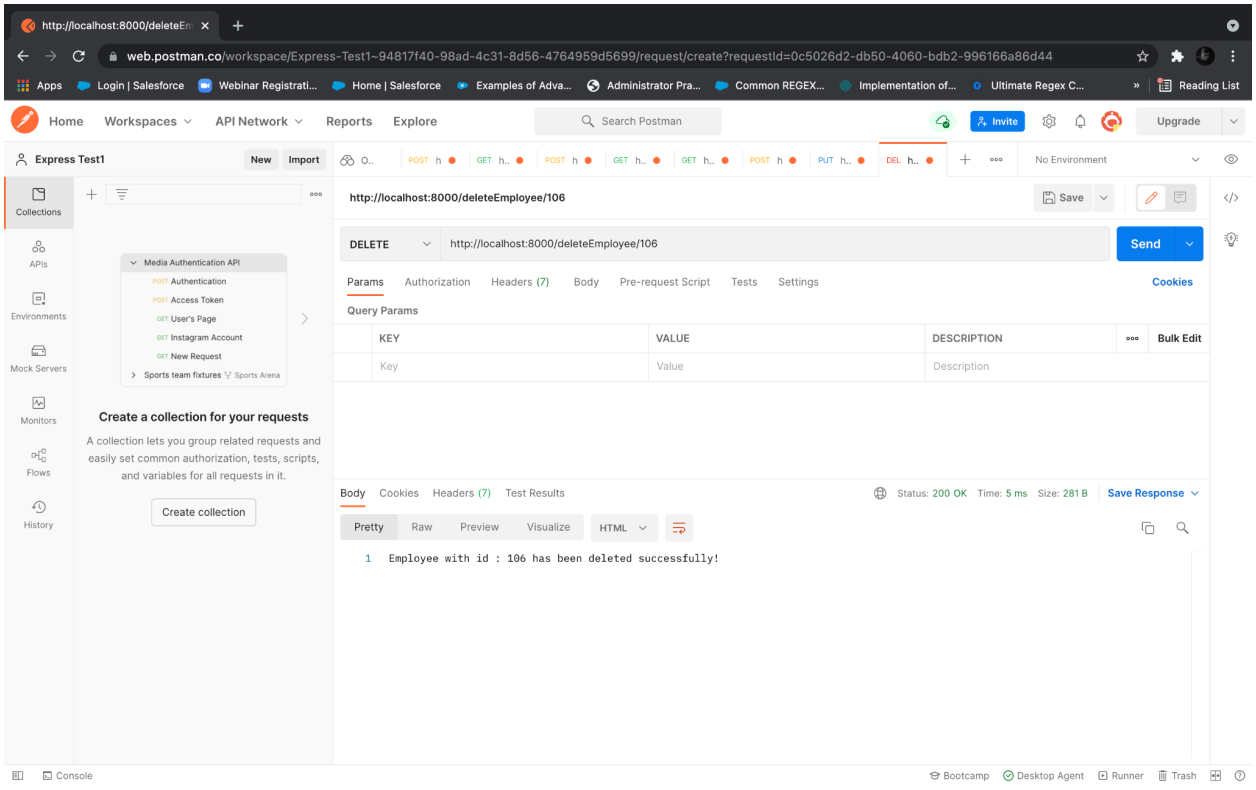


Verifying whether the data is added into emp object array using 'http://localhost:8000/getAllEmployeeData' request



f. delete - deleteRecord() -- Takes the id as the path parameter and removes the record on given id and returns the "Record deleted successfully" or "Unable to delete record".After this operation, verify in postman that the given record is deleted by giving 'http://localhost:8000/getAllEmployeeData' request

```
//deleting the employee with the given id
app.delete('/deleteEmployee/:id', (req,res)=>{
  var employeeId = req.params.id;
  var start = 0;
  for(var i=0;i<emp.length;i++){
    if(emp[i].id == employeeId){
      start = i;
      break;
    }
  }

  emp.splice(start,1);

  res.send(`Employee with id : ${employeeId} has been deleted successfully!`);
});
```



Verifying whether the data is added into emp object array using 'http://localhost:8000/getAllEmployeeData' request