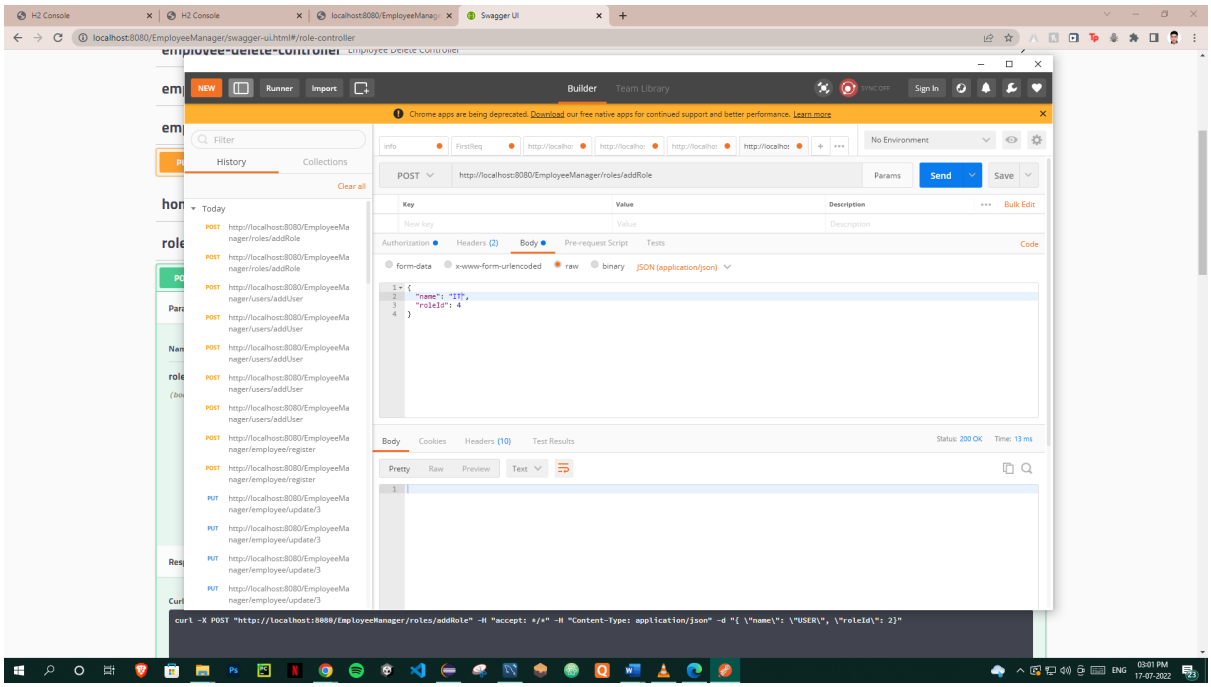
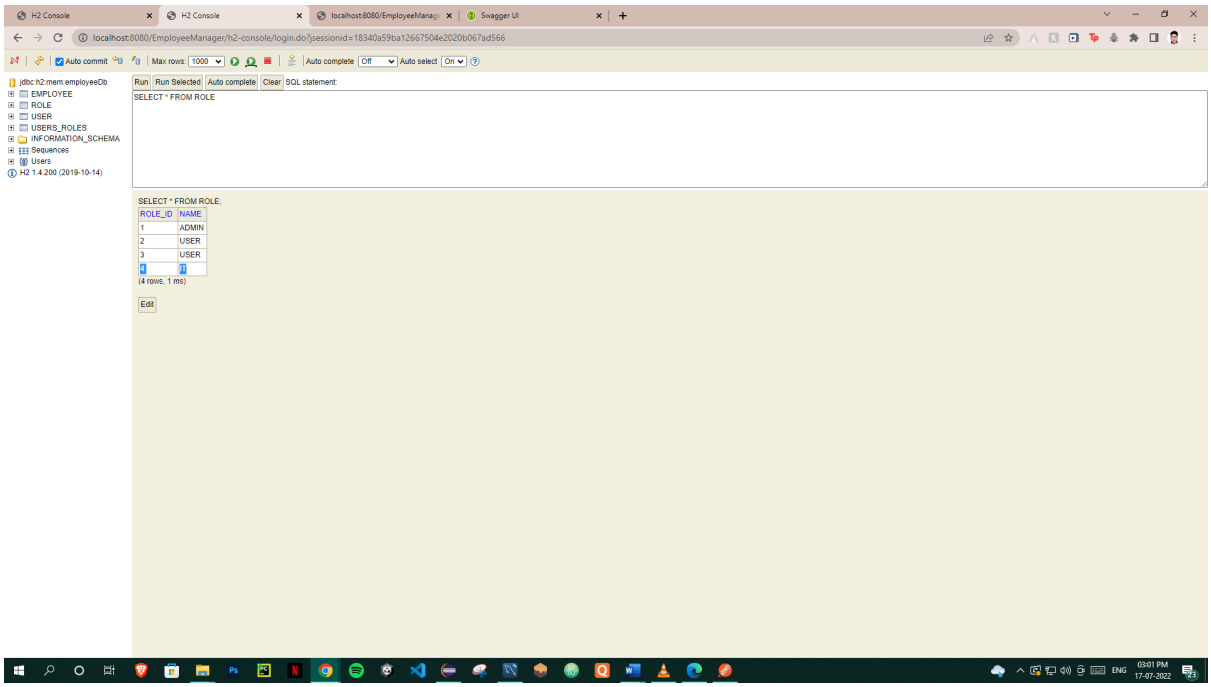
# 1. CRUD

The CRUD functionality as well as different endpoints required in the application - as per the software requirement specification document - have been shown below along with screenshots depicting complete functionality have been provided for reference :

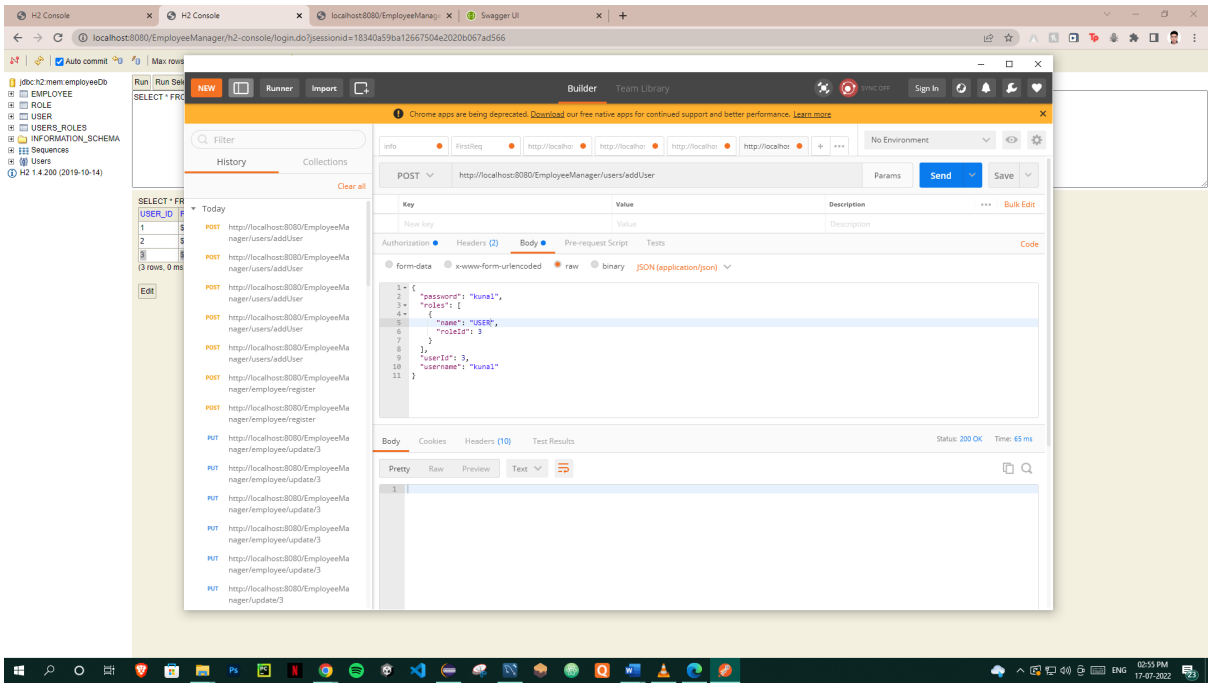
1. Adding new role dynamically in the DB.
   (**p.s:** *application should be able to add roles in the database dynamically in the db*)



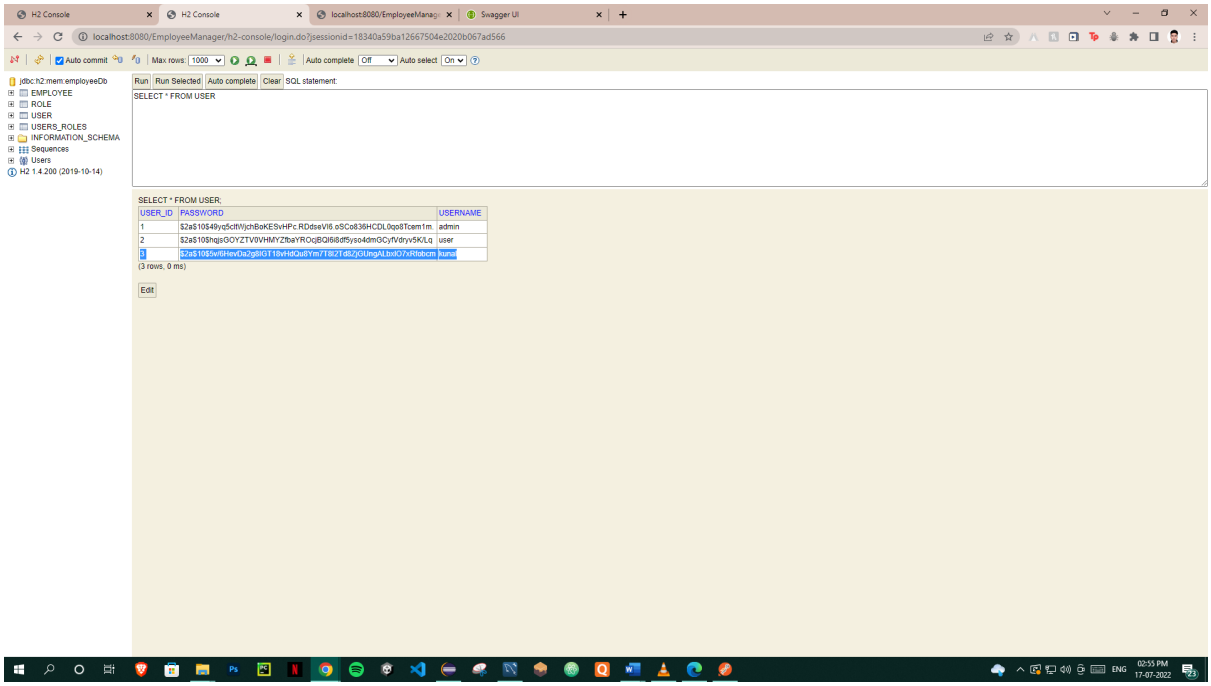

⇨ Verifying addition of new role to DB using h2-console.



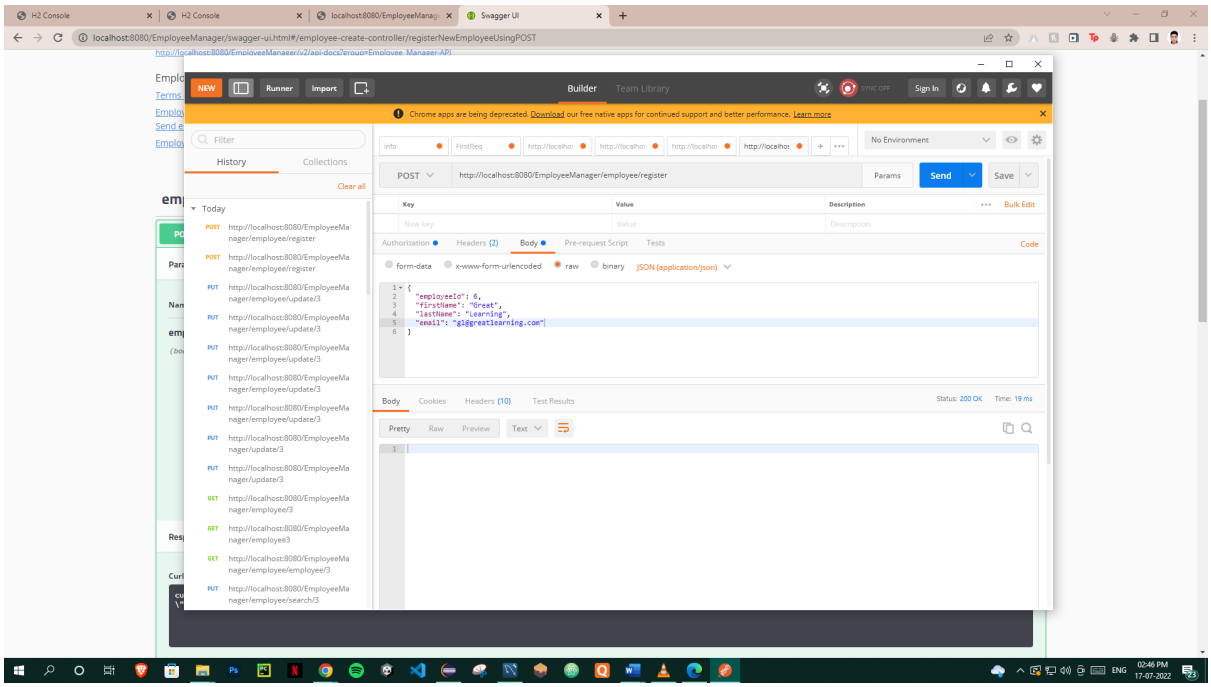
2. Adding new user ‘kunal’ in the DB with authentication role USER (**p.s:** *application should be able to add Users in the db which can be used for authentication purposes*)



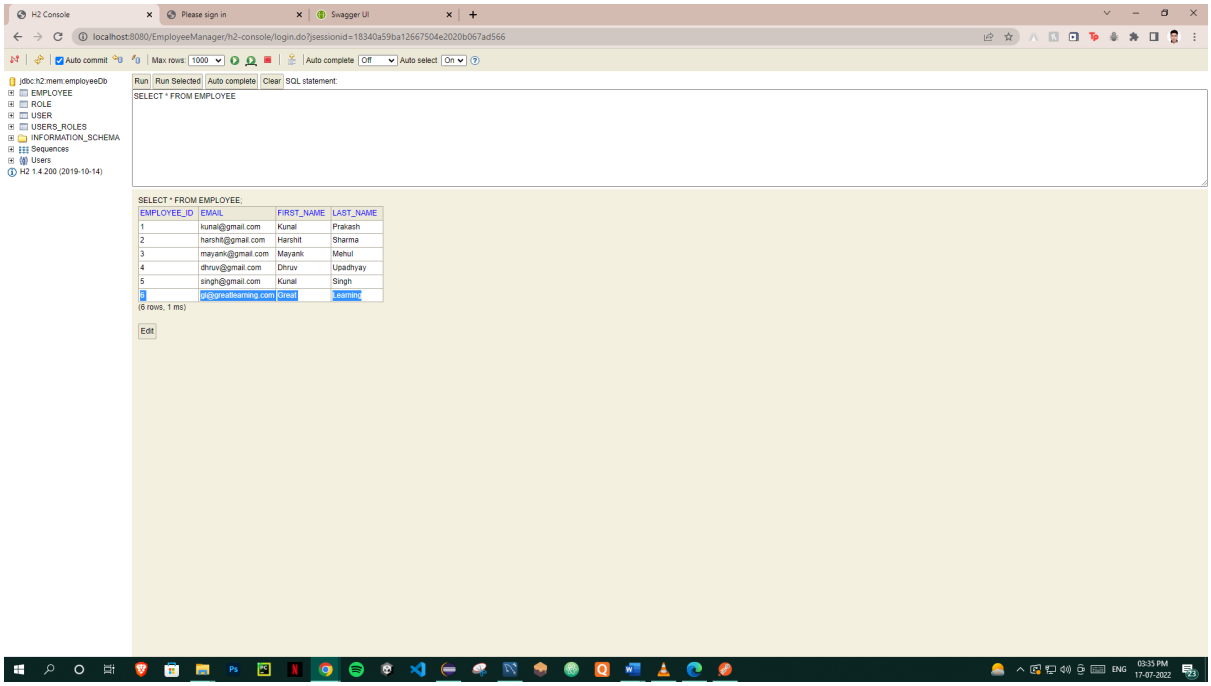

⇨ Verifying addition of new user ‘kunal’ with role USER added in h2-console.



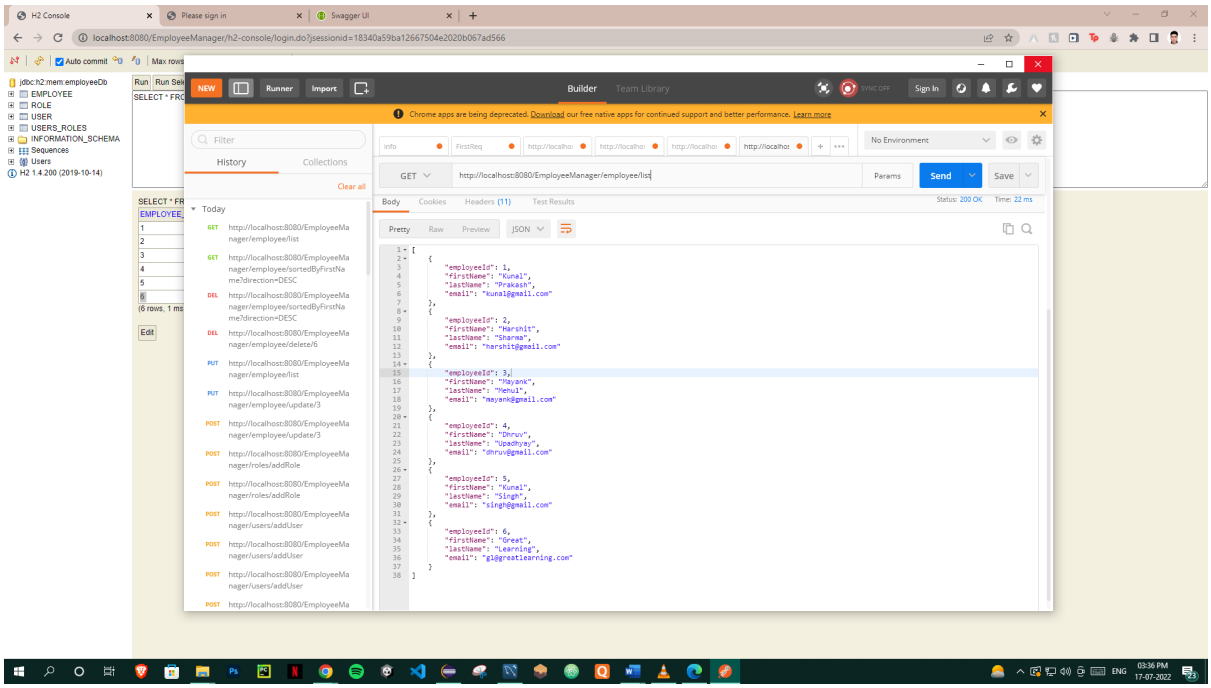
3. Adding new employee in DB while the authenticated user is ADMIN (**p.s:** *application should be able to add employees data in the db if and only if the authenticated user is ADMIN*)



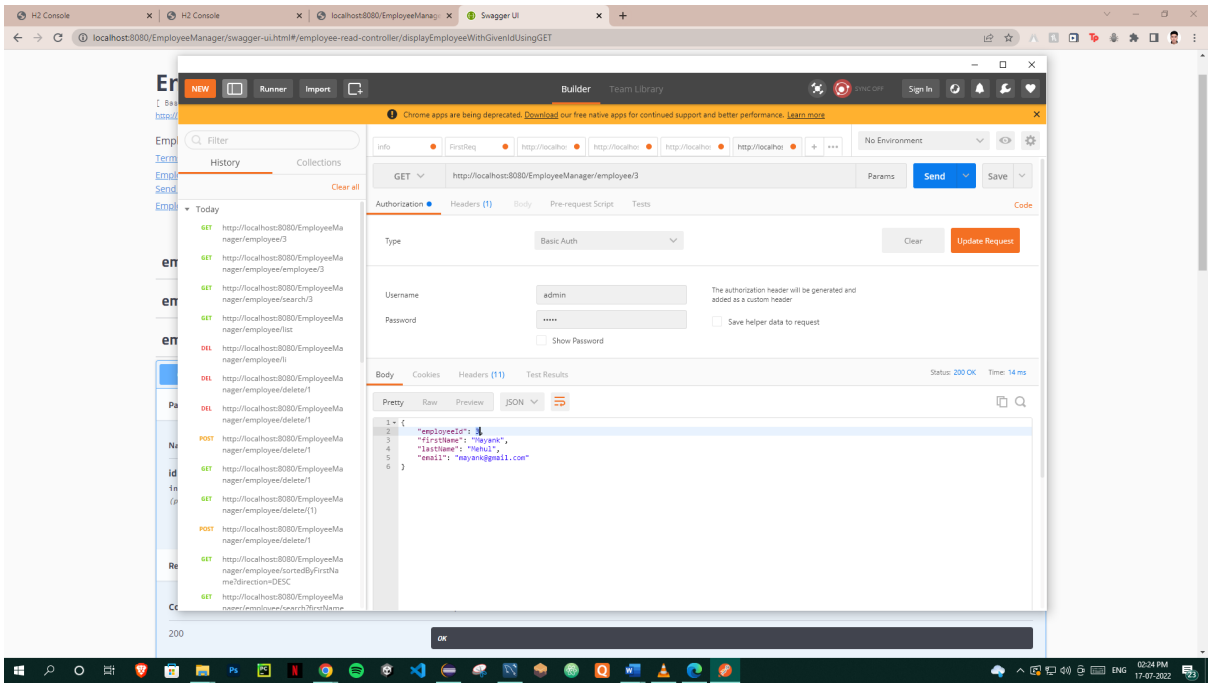
⇨ Verifying if new user was added in h2-console.



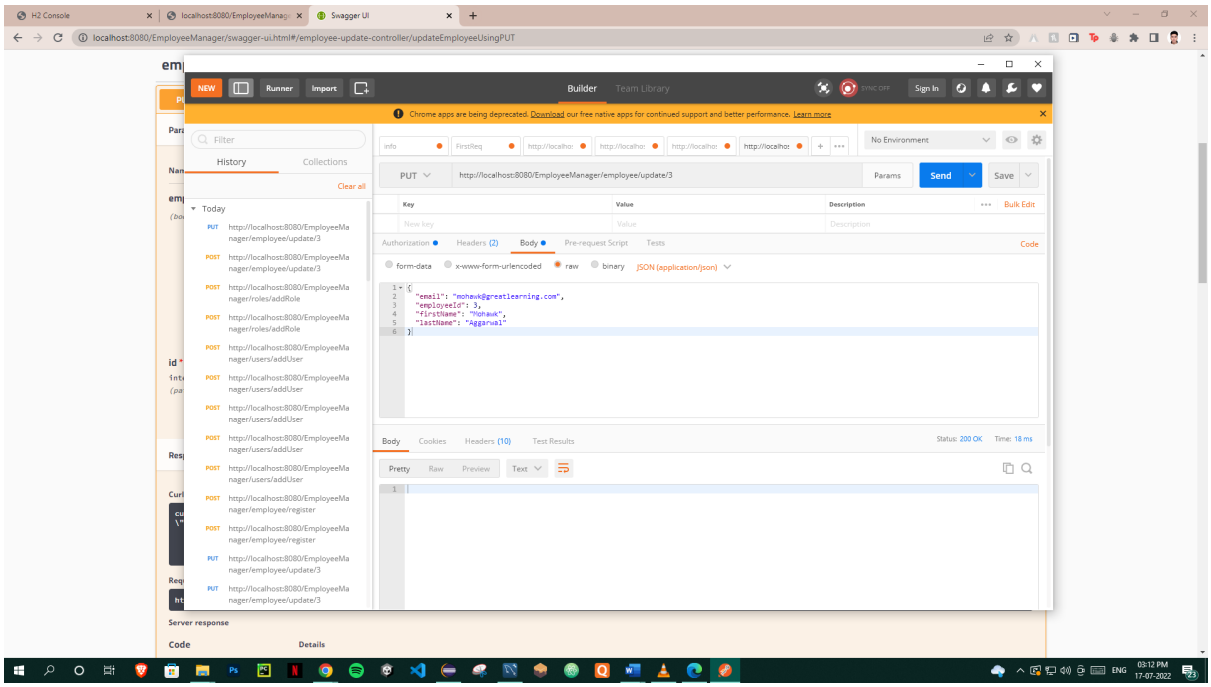
4. Listing all employees stored in database
   (**p.s:** *endpoint to list all the employees stored in the database*)



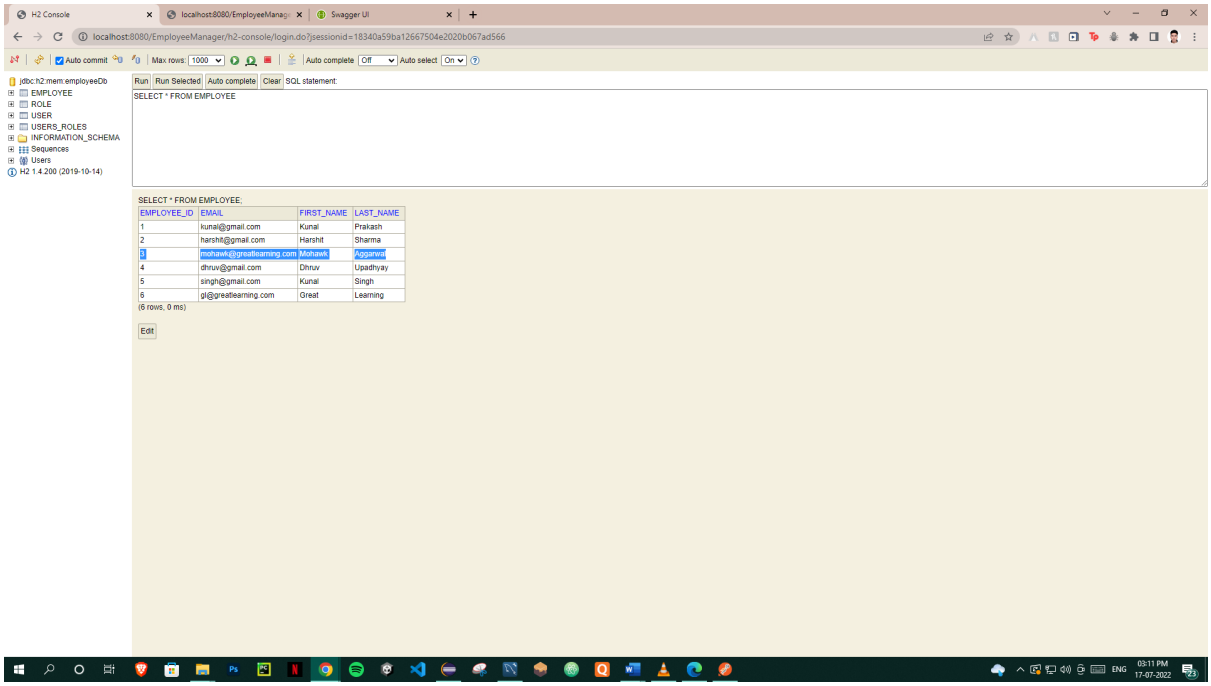

5. Fetching an employee record based on ‘employee_id’ of the employee. Here, we are fetching an employee with ‘employee_id’ : 3
   (**p.s:** *endpoint to fetch or get an employee record specifically based on the id of that employee*)



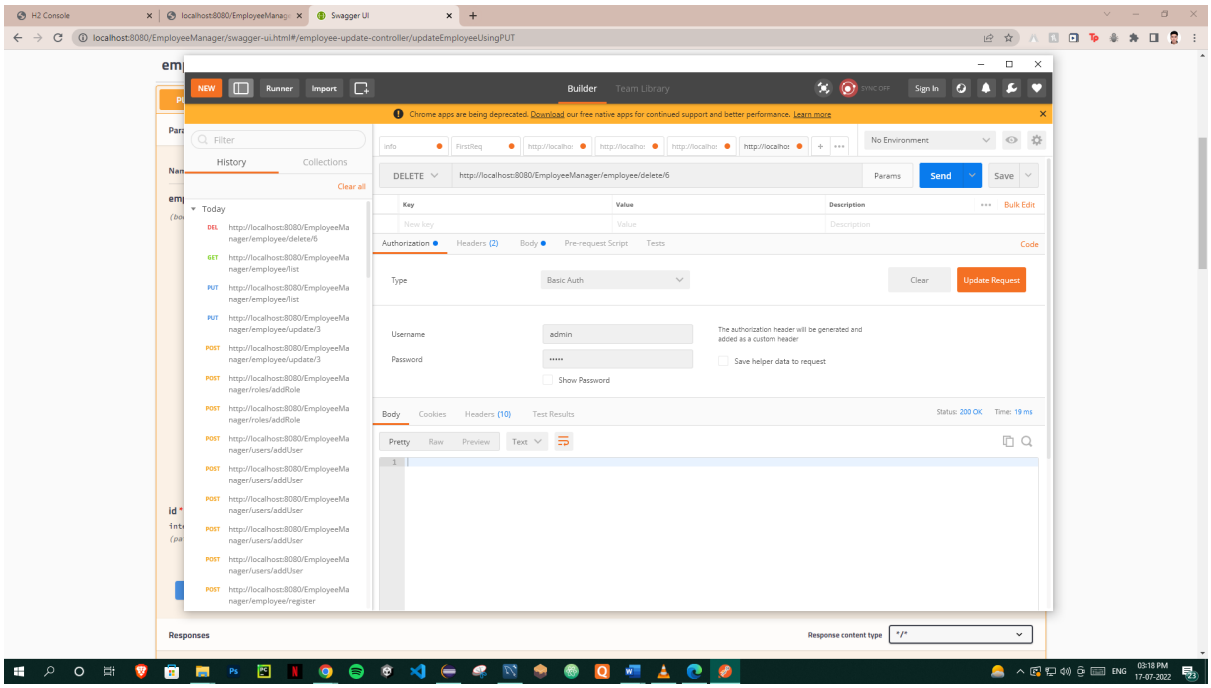
6. Performing update operation on employee with ‘employee_id’ : 3
(**p.s:** *endpoint to update an existing employee record with the given updated json object* )



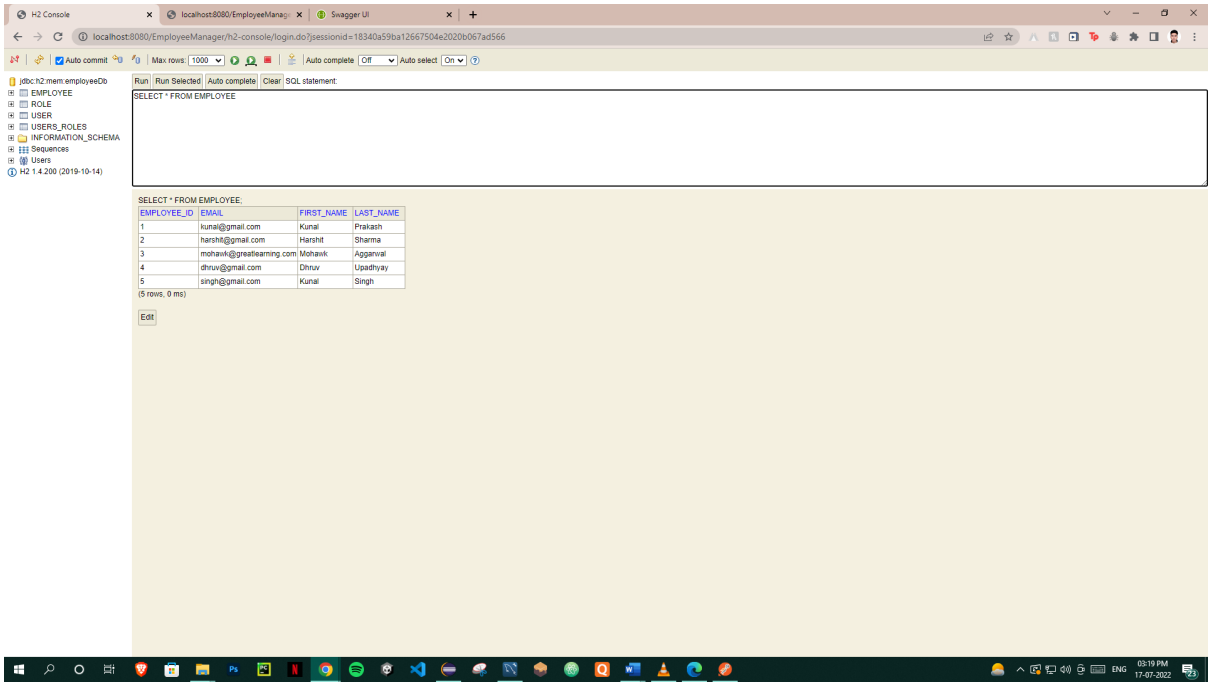

⇨ Verifying update operation on employee with ‘employee_id’ : 3 in h2-console



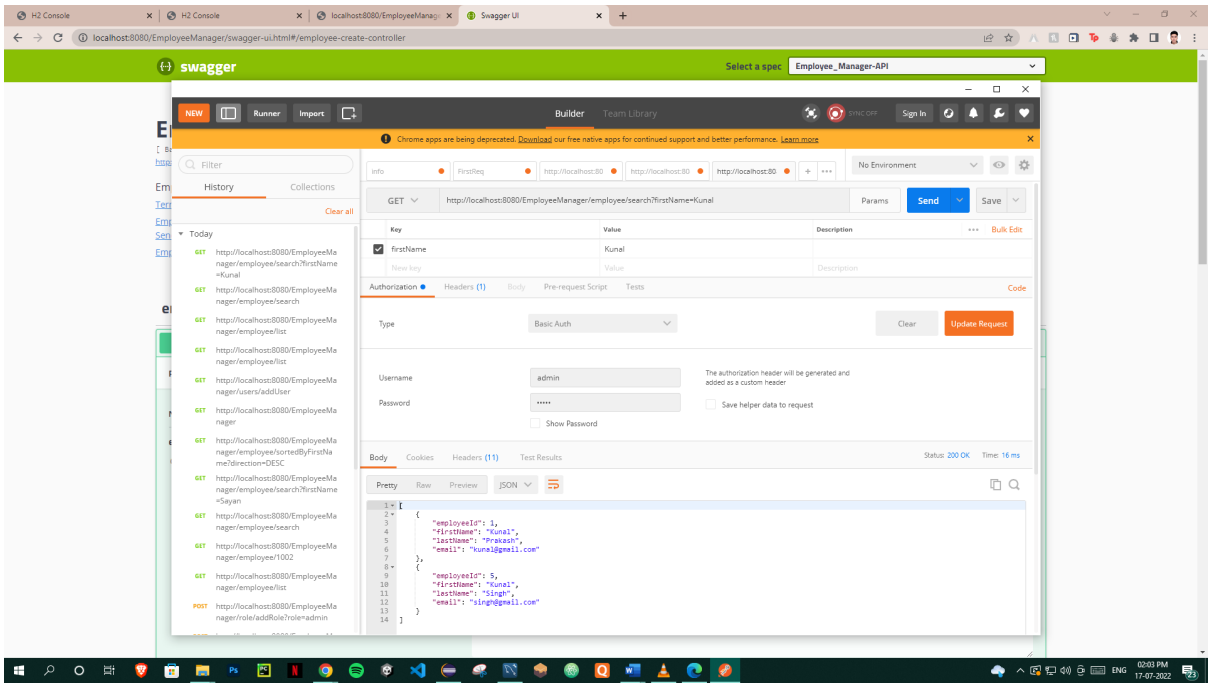
7. Deleting employee record based on ‘employee_id’. Deleting employee with id = 6. (**p.s:** *endpoint to delete an existing employee record based on the id of the employee*)



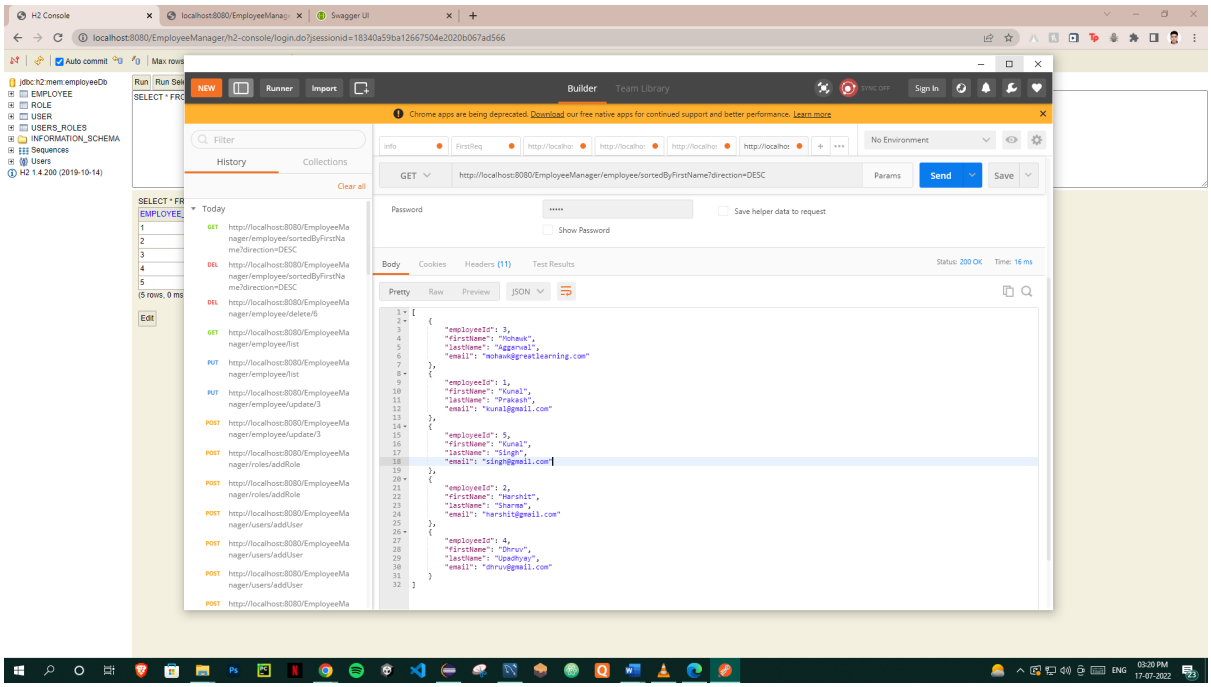

⇨ Verifying deleted employee with ‘employee_id’ : 6 using h2-console.



8. Searching an employee by firstName : Kunal , displays two records found.
(**p.s** : *endpoint to fetch an employee by his/her first name and if found more than one record then list them all* )

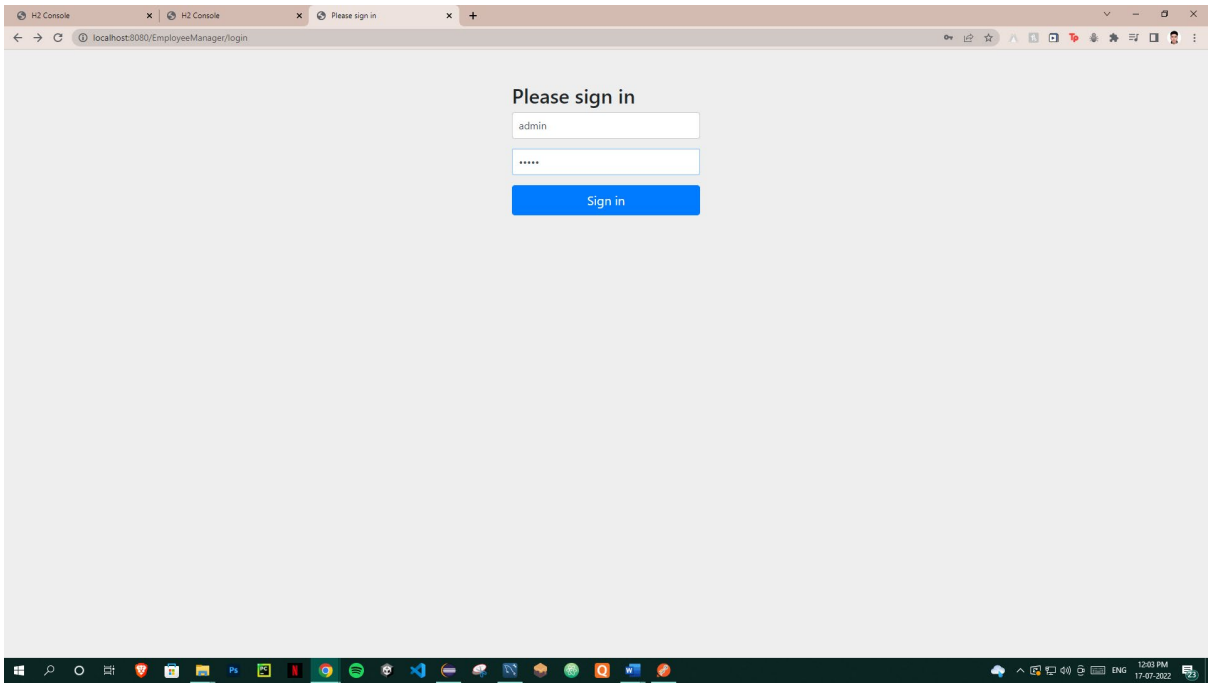



9. Listing employees sorted by their ‘firstName’ in ‘DESC’ order.
(**p.s:** *endpoint to list all employee records sorted on their first name in either ascending order or descending order*)

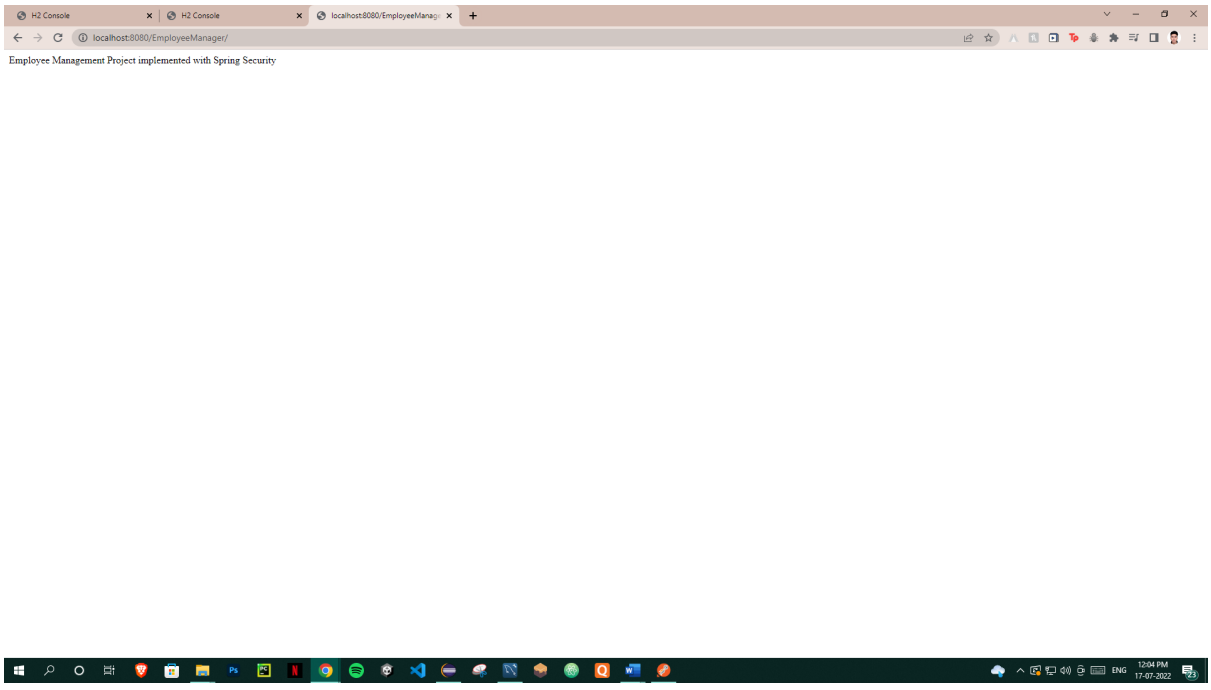


# 2. SECURITY

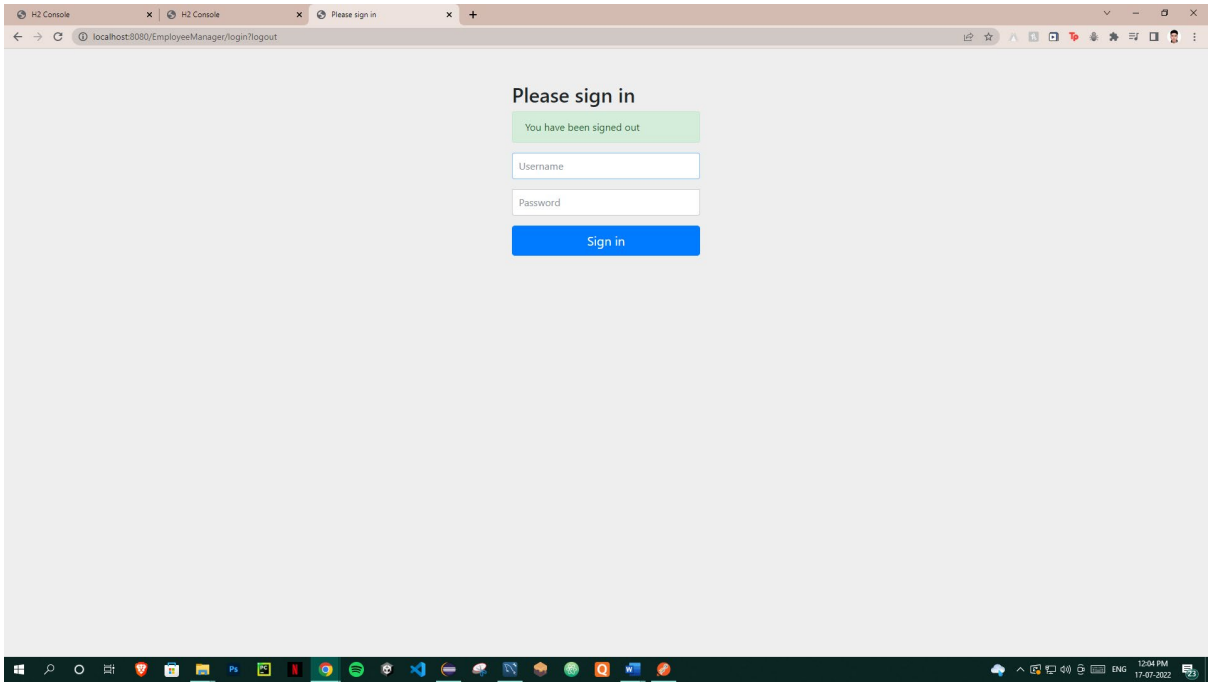
1. Logging in using credentials for ‘admin’ role.



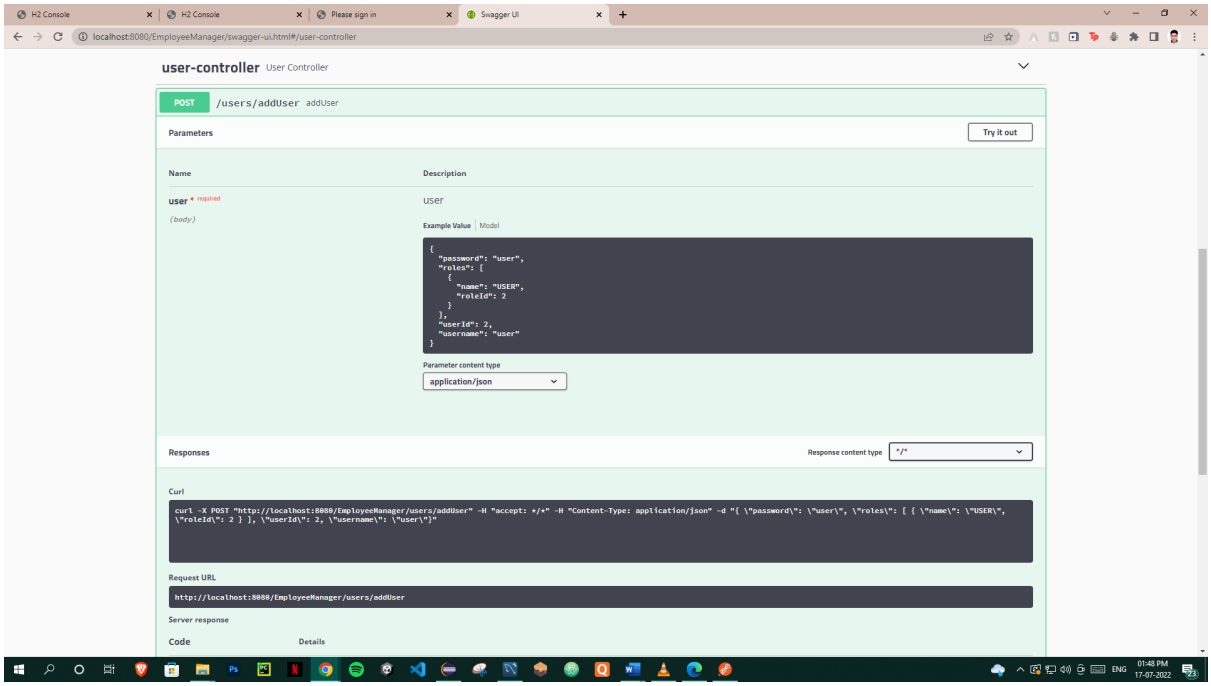

2. Displaying home page on successful login.



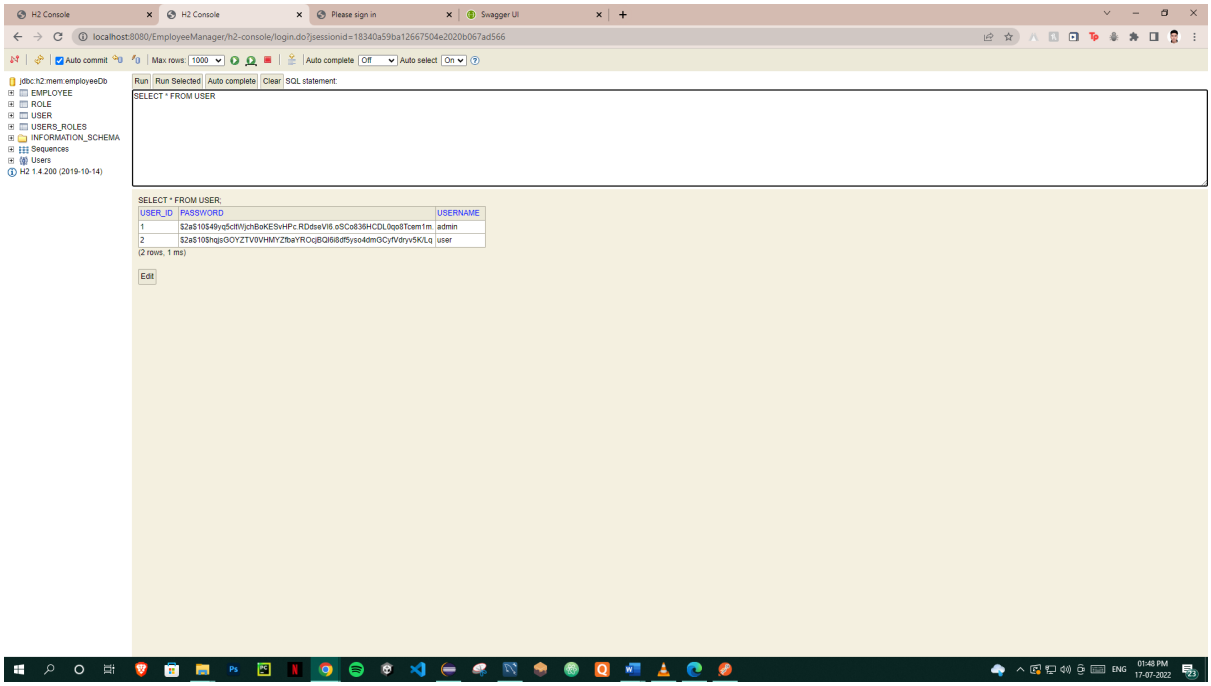
3. Logout page after successful logout.



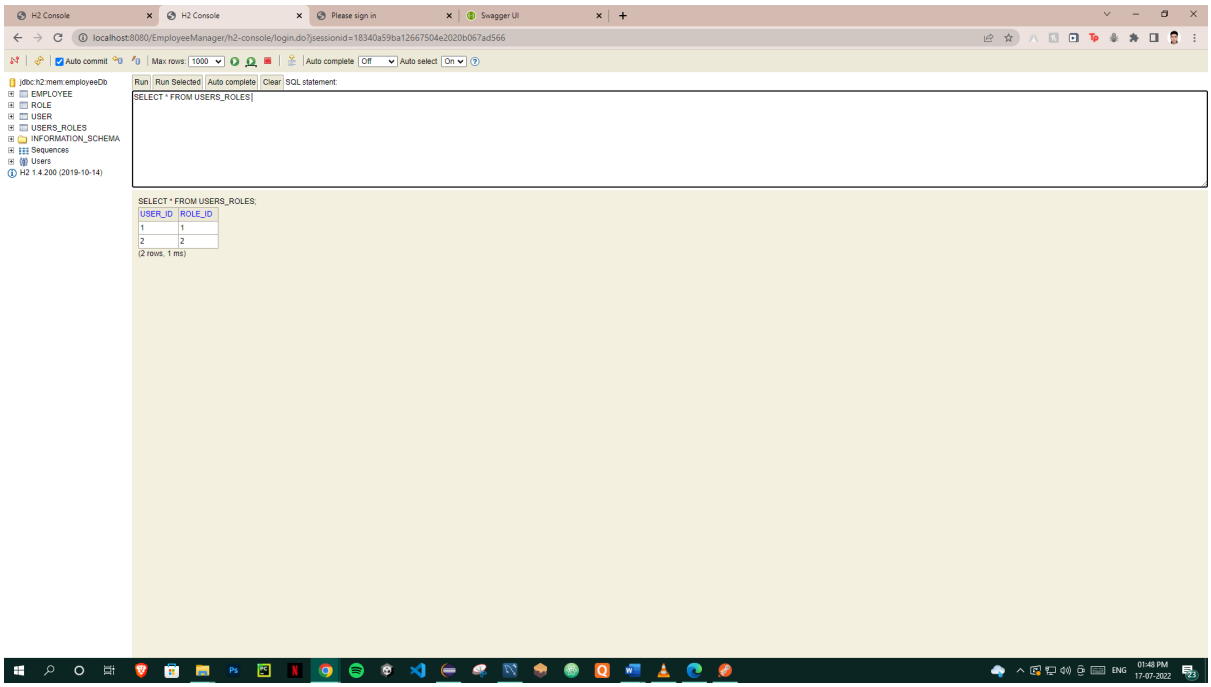

4. Adding user role



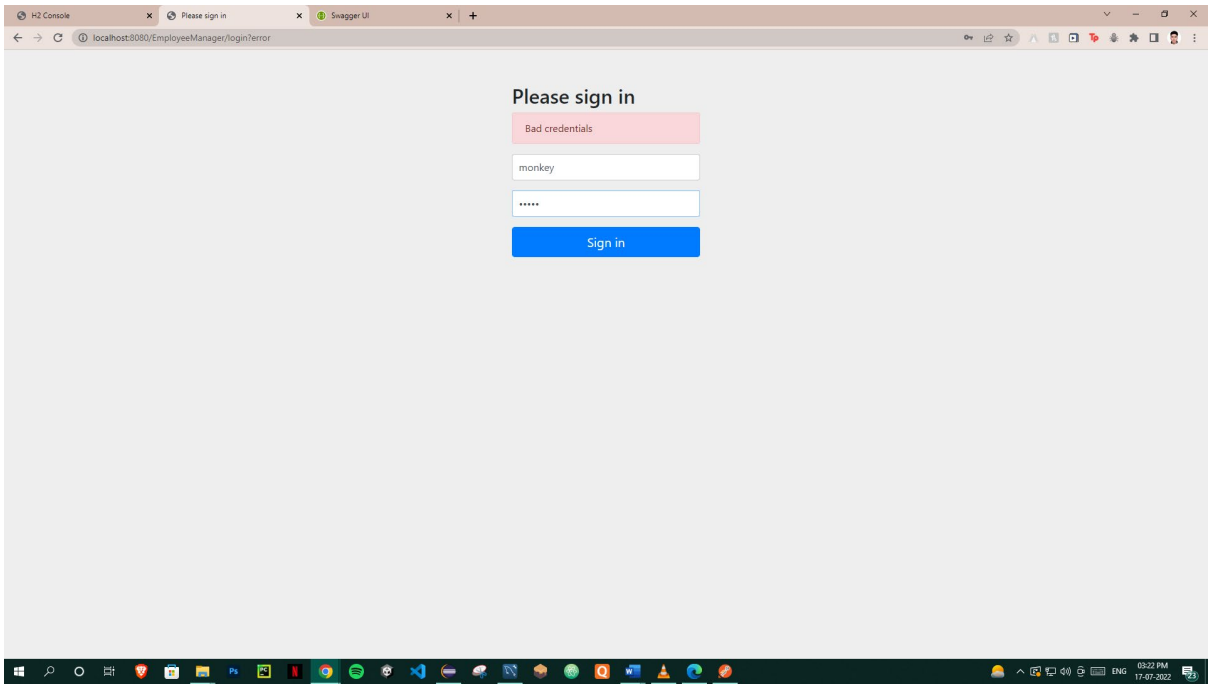
5. ‘user’ role added in USER table



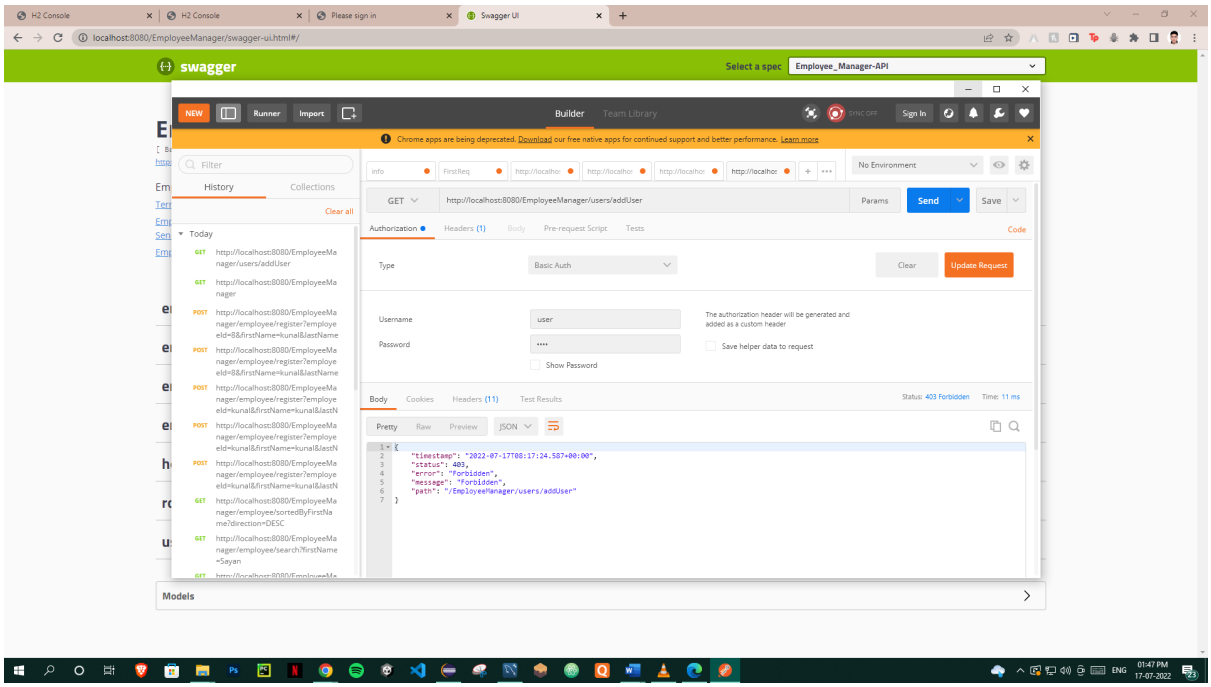

6. ‘user’ role added in user_roles table



7. Bad credentials' error since 'monkey' is not a registered role. (p.s: at this point we had stored only two roles i.e. 'admin' , 'user')




8. Access of 'user' role is restricted when trying to perform 'addUser' operation



9. Access of ‘user’ role is also restricted when trying to list employees