**Project - Salary Slip Generator**Project Documentation

**Project Requirements -**   
**Node version** - 18.12.0

**The database used** - MySQL

**Administration tool used for MySQL** - PHPMyAdmin

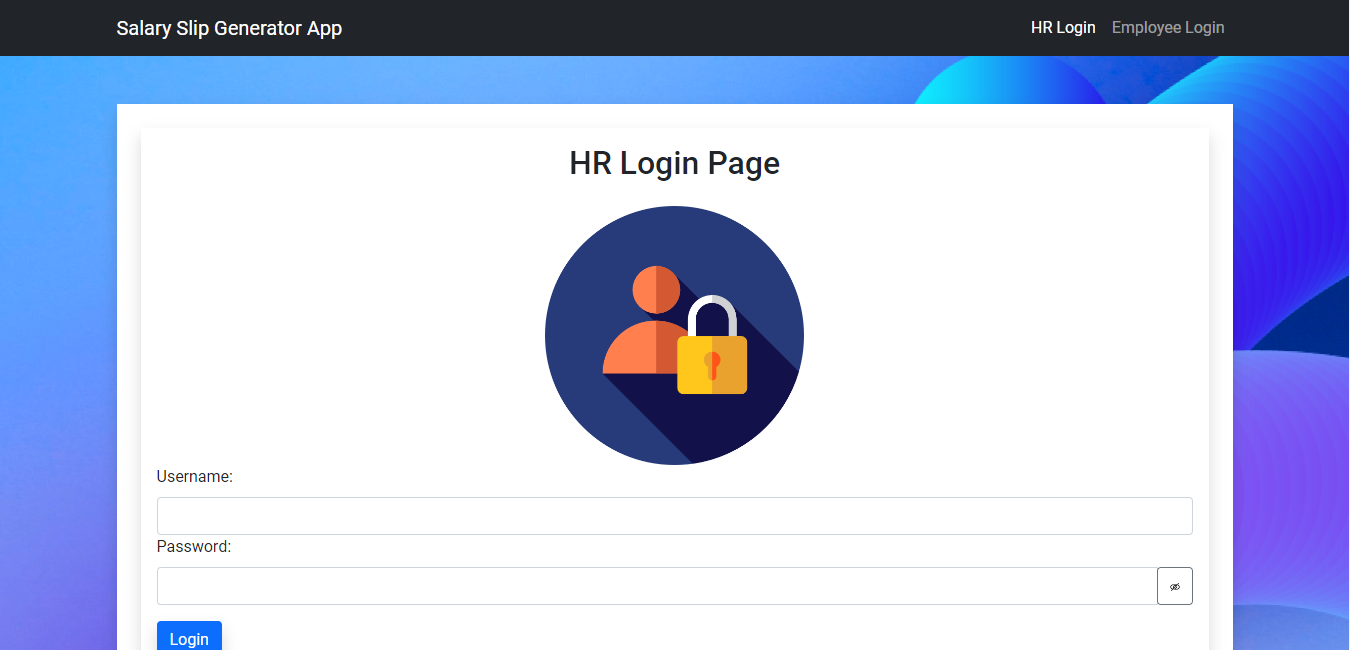
**API testing tool** - Postman

**Front-end** - Angular

**Backend** - Node

**Package manager command used** - npm

**Different views of the project -**

1. **http://localhost:4200/** **- HR Login Page / Homepage**  
 

**Angular component** - hr-login.component.ts

**Main features** -

1. **Hr Login**  
   **API call** **after entering the correct username and password**- When the user enters the correct credentials and clicks on the login button, the API call is -

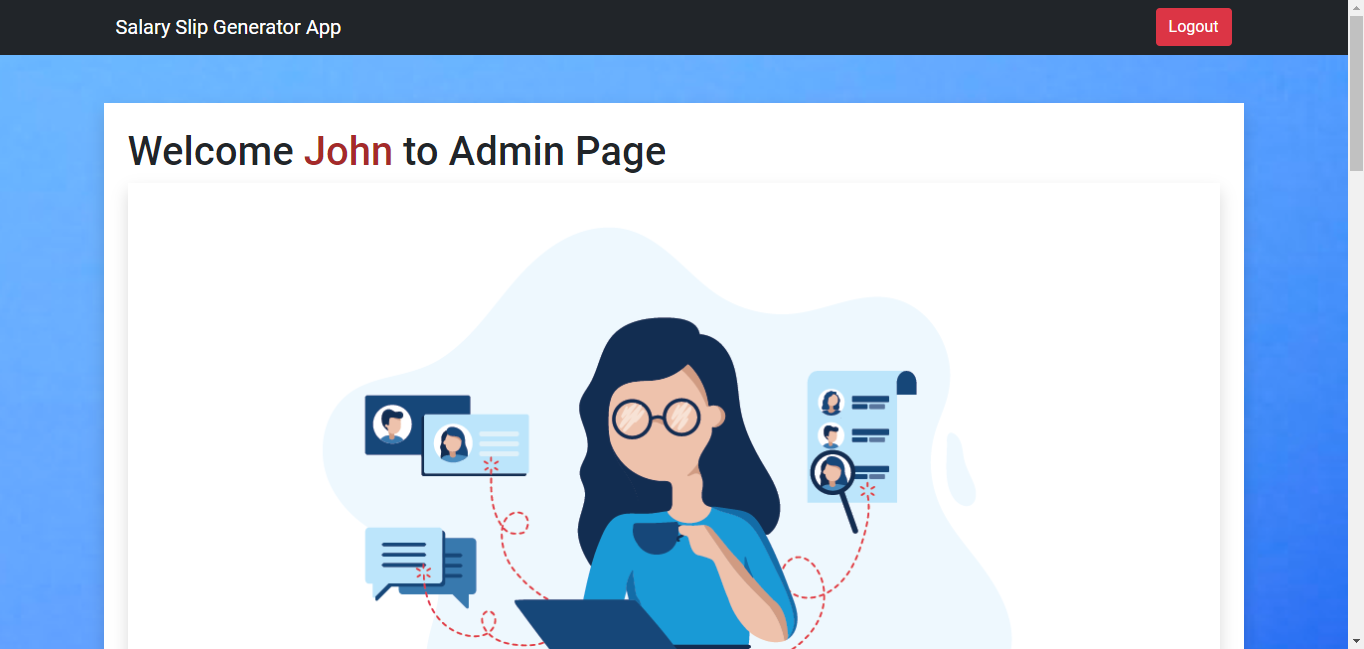
API with GET method - [https://localhost:3000/api/admin\_login](https://localhost:4200/api/admin_login)

**Route to another component** - Yes (to

[http://localhost:](http://localhost:4200/admin-page)[3000](https://localhost:4200/api/admin_login)[/admin-page](http://localhost:4200/admin-page)

**Data fetched from database -** Yes (from DB- “salary\_slip\_generator” by table name - “admin\_login”)

**2. http://localhost:4200/admin-page - Admin Page**



**Angular component** - hr-protected-page.component.ts

**Main feature -**

1. **Show all Employees Details -**

**The API call to fetch all employee details–** API with GET method ->

[https://localhost:](https://localhost:4200/api/fetchAllEmployeesDeatils)[3000](https://localhost:4200/api/admin_login)[/api/fetchAllEmployeesDetails](https://localhost:4200/api/fetchAllEmployeesDeatils)

**Route to another component** - No  
**Service.ts file used** - auth.service.ts

**Data fetched from the database -** Yes (from DB- “salary\_slip\_generator” by table name - “employees\_info”)

1. **Add new Employees -**

**The API call to register new employees -** API with POST method ->

[https://localhost:](https://localhost:4200/api/register)[3000](https://localhost:4200/api/admin_login)[/api/register](https://localhost:4200/api/register)

**Route to another component** - No  
**Service.ts file used** - auth.service.ts

**Data saved in the database -** Yes (in DB- “salary\_slip\_generator” by table name - “employees\_info”)

1. **Fetch the full name of the admin from the database -**

**The API call to fetch the full name of the admin from the database -** API with GET method ->

[https://localhost:](https://localhost:4200/api/getFullNameOfAdmin)[3000](https://localhost:4200/api/admin_login)[/api/getFullNameOfAdmin](https://localhost:4200/api/getFullNameOfAdmin)

**Service.ts file used** - auth.service.ts

**Data fetched from the database -** Yes (from DB- “salary\_slip\_generator” by table name - “admin\_login”)

1. **Update Employee details -**

**The API call to fetch the previous data values of the employee from the database-** API with GET method ->

[https://localhost:](https://localhost:4200/api/fetchAllDetailsOfEmployee)[3000](https://localhost:4200/api/admin_login)[/api/fetchAllDetailsOfEmployee](https://localhost:4200/api/fetchAllDetailsOfEmployee)

**Service.ts file used** - auth.service.ts

**Data fetched from the database -** Yes (from DB- “salary\_slip\_generator” by table name - “employees\_info”)

**The API call to update the data of the employee in the database -** API with POST method ->

<https://localhost:3000/api/updateUser>

**Route to another component** - No  
**Service.ts file used** - auth.service.ts

**Data saved in the database -** Yes (in DB- “salary\_slip\_generator” by table name - “employees\_info”)

1. **Generate Payslip -**

**The API call to fetch the previous data values of the employee from the database-** API with GET method ->

[https://localhost:](https://localhost:4200/api/fetchAllDetailsOfEmployee)[3000](https://localhost:4200/api/admin_login)[/api/fetchAllDetailsOfEmployee](https://localhost:4200/api/fetchAllDetailsOfEmployee)

**Service.ts file used** - auth.service.ts

**Data fetched from the database -** Yes (from DB- “salary\_slip\_generator” by table name - “employees\_info”)

1. **Delete Employee record -**

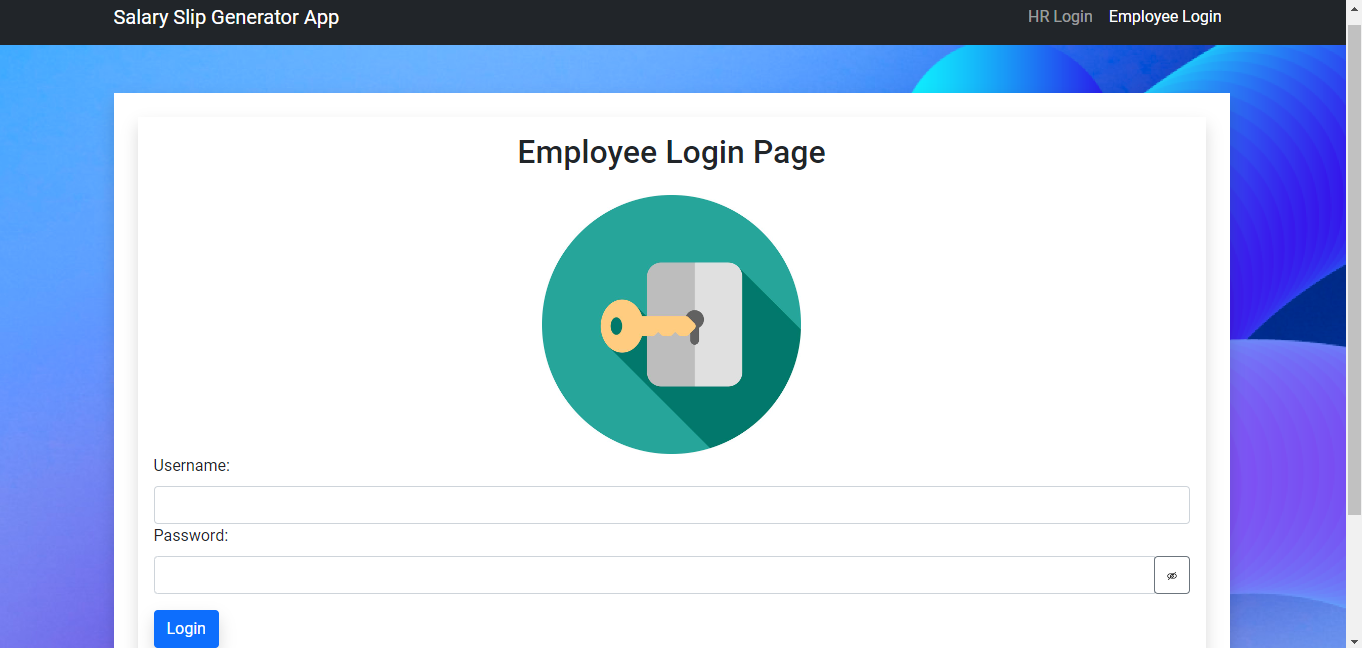
**The API call to delete the particular Employee record from the database -** API with POST method ->

[https://localhost:3000/api/deleteEmployee](https://localhost:4200/api/deleteEmployee)

**Route to another component** - No  
**Service.ts file used** - auth.service.ts

**Data saved in the database -** Yes (in DB- “salary\_slip\_generator” by table name - “employees\_info”)

**3. http://localhost:4200/emp\_login - Employee Login Page**



**Angular component** - login-page.component.ts

**Main features -**

1. **Employee Login**  
   **API call** **after entering the correct username and password**- When the user enters the correct credentials and clicks on the login button, the API call is -

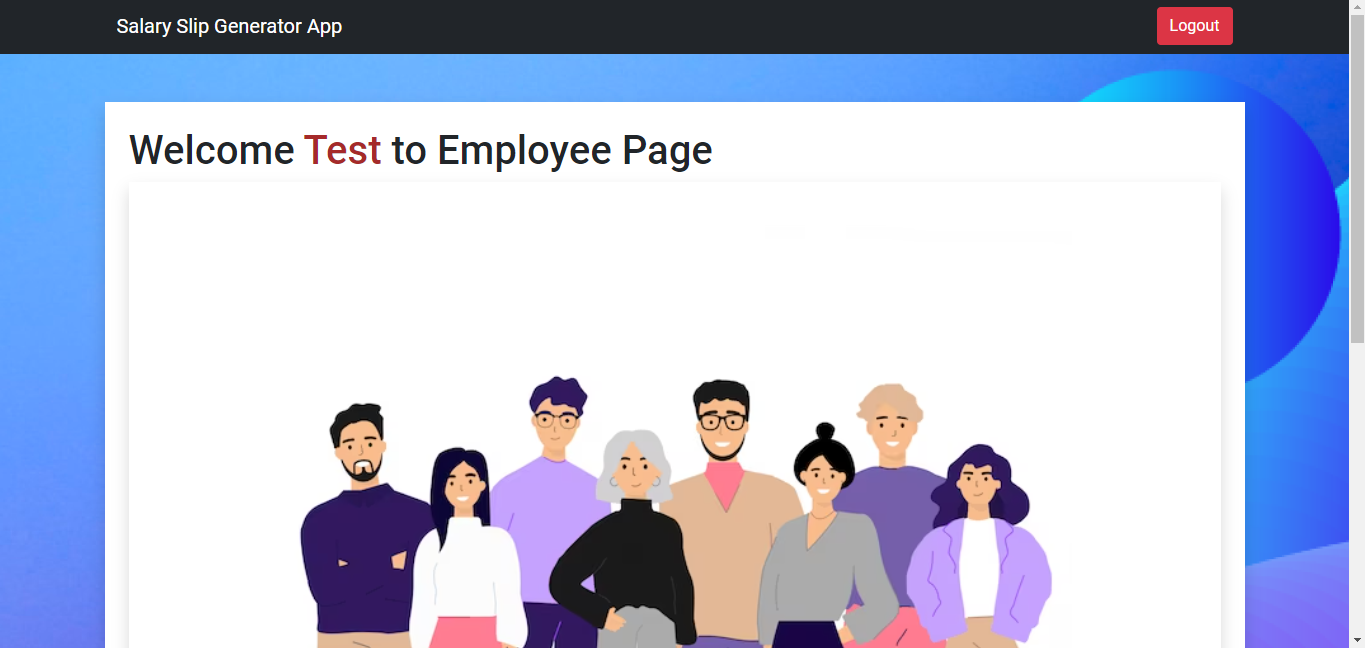
API with GET method - <https://localhost:3000/api/login>

**Route to another component** - Yes (to

<http://localhost:4200/protected-page>

**Data fetched from the database -** Yes (from DB- “salary\_slip\_generator” by table name - “employees\_info”)

**4. http://localhost:4200/protected-page - Employee Page**

****

**Angular component** - protected-page.component.ts

**Main features -**

1. **Show Employee Details -**

**The API call to fetch employee details–** API with GET method ->

<https://localhost:3000/api/fetchAllDetailsOfEmployee>

**Route to another component** - No  
**Service.ts file used** - auth.service.ts

**Data fetched from the database -** Yes (from DB- “salary\_slip\_generator” by table name - “employees\_info”)

1. **Fetch the full name of the employee from the database -**

**The API call to fetch the full name of the employee from the database -** API with GET method ->

[https://localhost:](https://localhost:4200/api/getFullNameOfAdmin)[3000](https://localhost:4200/api/admin_login)[/api/getFullName](https://localhost:4200/api/getFullNameOfAdmin)

**Service.ts file used** - auth.service.ts

**Data fetched from the database -** Yes (from DB- “salary\_slip\_generator” by table name - “employees\_info”)

**Note: -** Use the command - **“ng-serve”** to run the frontend server

Use the command - **“node app.js”** to run the backend server

The database name is - **“salary\_slip\_generator”**