Internship Project:
Employee Log Manager

Introduction

Employee log manager is a web application which is used to handle the daily entry and exit logs of an employee, calculate the working hours, excluding the break and is used to track monthly salary logs.

Whereas, the admin can check entry and salary logs of all the employees, and has the right to create, update and delete users. Also, he can calculate company's manpower investment.

Objectives

1. Calculate total hours spent by each employee of the organization (per month)
2. Calculate the total salary earned by each employee of the organization (per month)
3. Keep track of hours spent by all the employees at the organization (per month)
4. The total investment is done by the organization on manpower (per month)

Specifications

The basic technology stack for the project is mentioned below:

1. **Django Rest Framework**: It is a Framework used to create APIs according to the functionality. They act as an interface between the tables at the backend and data displayed at the front end. All the requests are sent to the API, which processes the request, taking data from the backend and as per the requirement, this web service fetches the results according to the business logic layer and passes it to the front-end. Serializers convert the response into JSON Format. We can make different requests like GET, POST, PUT, DELETE, basically the CRUD Operations.

2. **PostgreSQL:** The database management system to store information, which can be retrieved accordingly, when required. In the tables, we will have primary and foreign keys and we can have different relations among the tables. We firstly designed a Schema and accordingly planned the different fields of the database.

3. **HTML5, CSS, jQuery, JavaScript, Bootstrap:** Responsible for the front-end design and also it communicates with the business layer and accordingly display the results in the web application. Bootstrap is used to make attractive buttons and tables. jQuery is used to enable the filtering feature on the tables. JavaScript is used for basic functionality required on the front end.

4. **Python:** It is the Programming language for the project. Different modules and libraries of Python are used for different purposes.

Description

There would be a login page in the beginning where the users can enter their details, i.e. username and password to login. If the username and password match with those in the database, the user will be logged in, and will have an access token of API stored in the session variable. With the token, the user can access the API and accordingly make GET, POST and PUT requests.

Once he presses the logout button, the session variable is flushed and the token is vanished and hence he is redirected to the login page and cannot revisit his profile page, until and unless he logs in again.

If an employee forgets his/her password, he can reset a new password by providing his email address.

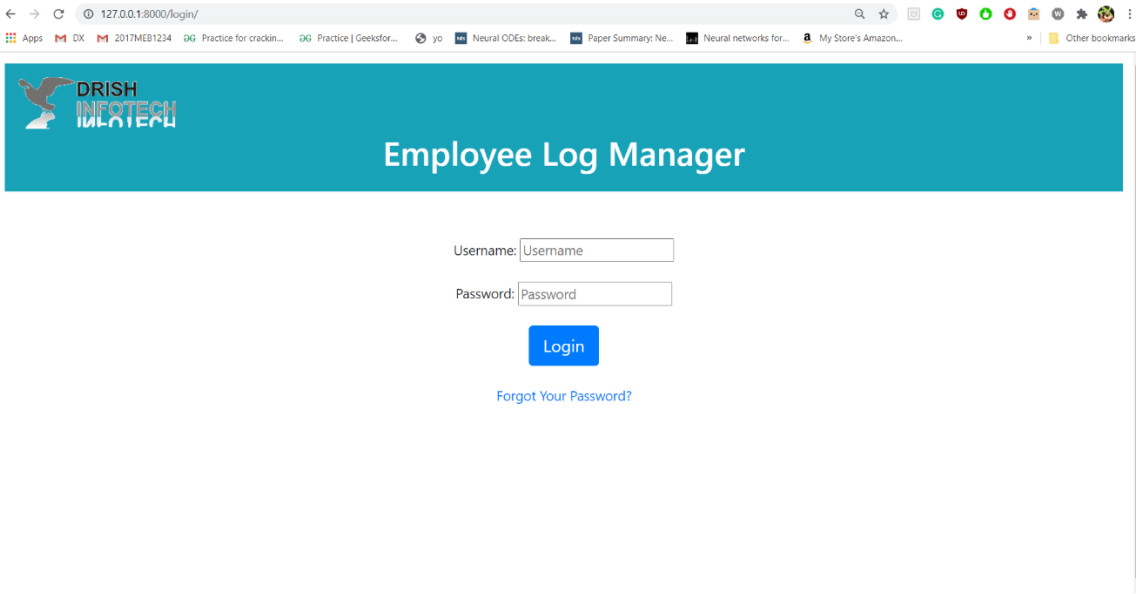

Figure 1: Login Page

There would be two types of users associated.

1. **Admin:** A superuser would have all the rights to perform any operation in the Application. He can check the timings of all the employees and able to check the salary of each of them for any month. He has all the rights to perform CRUD operation on anything.

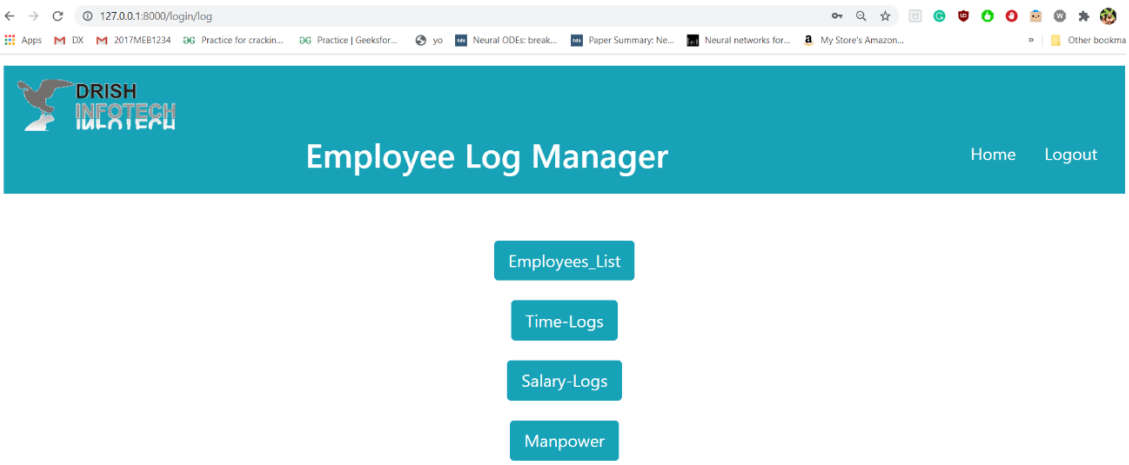

Figure 2: Admin Login

He can view the list of employees and has the right to create new employees in the database. The username would be unique and they will also be allotted a unique employee code.

Accordingly, he may also update or delete a pre-existing user.

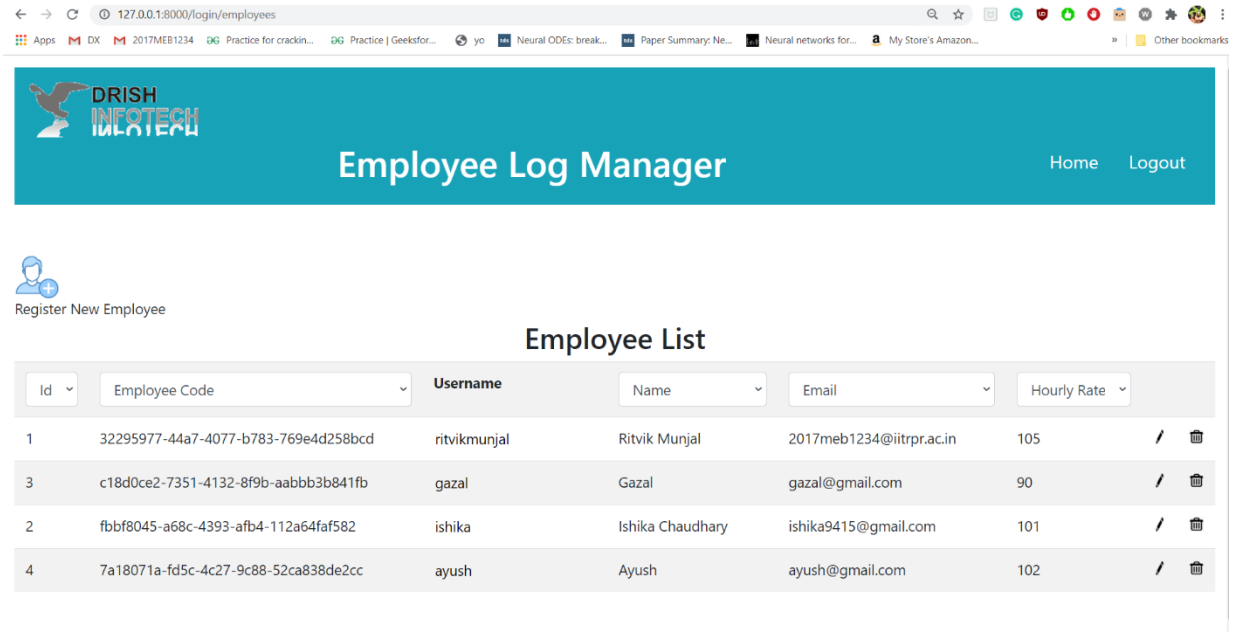

Figure 3: List of Employees with CRUD enabled

When a user is created, automatically an entry in the employees table is done.

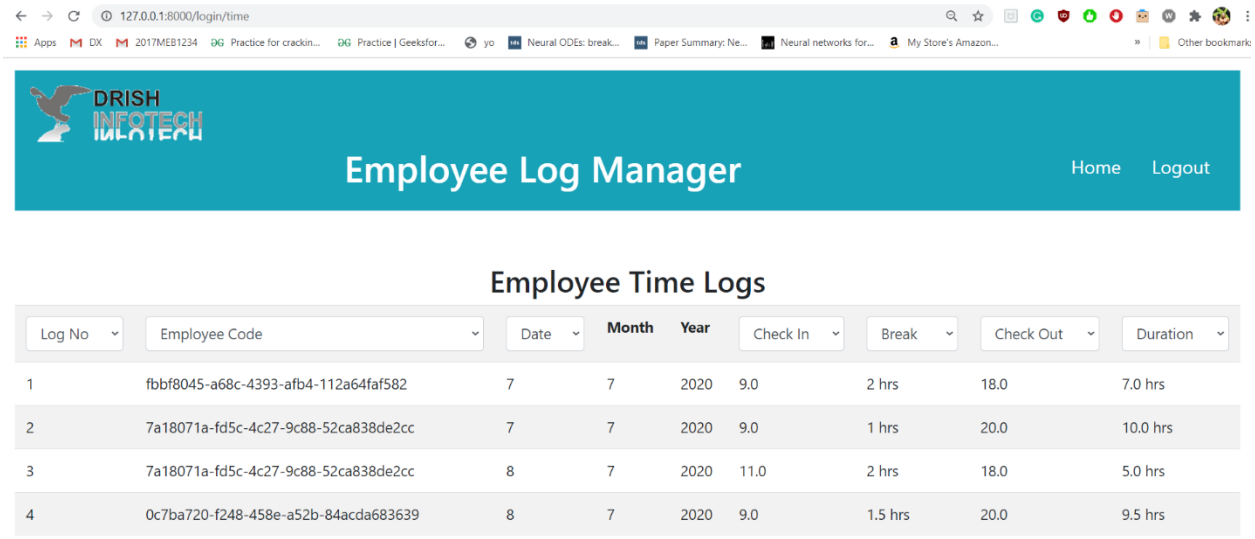

Figure 4: Time Logs of employees

The salary of an employee would be calculated on the following basis:

- **a.** For 192 hours a month, employee(s) would be paid at an hourly rate
- **b.** Above 192 hours and below 250 hours, employee(s) would be paid 1.5x of an hourly rate.
- **c.** Above 250, the hourly rate would get doubled for each employee.

E.g. If an employee has working hours of 225 and hourly rate of 50, so his salary would be:

Salary = 50*192 + 1.5*(225-192) *50 = Rs.12075

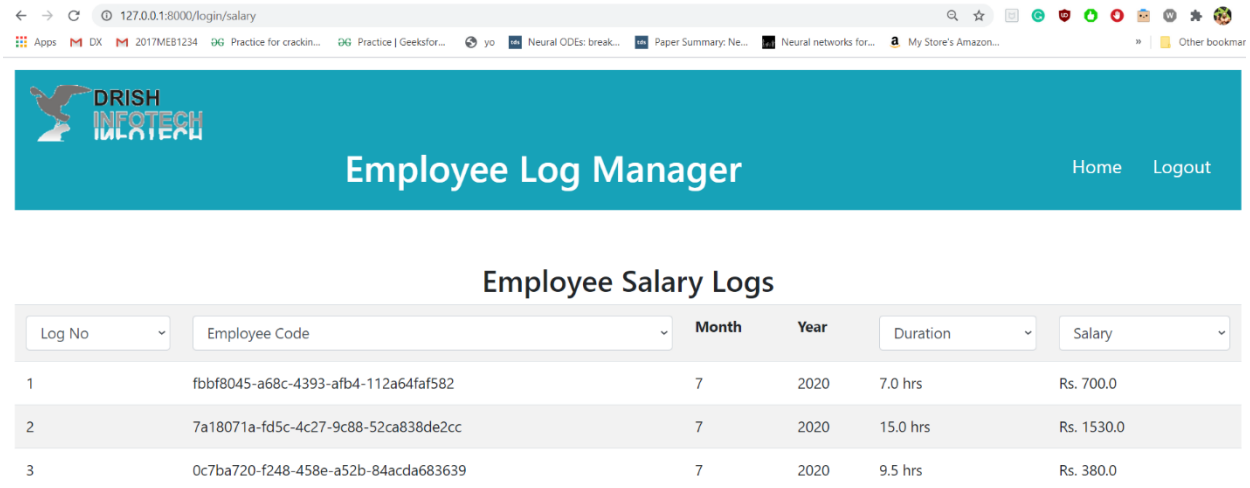

Figure 5: Salary Logs

Manpower is updated/created parallelly as a salary log is created or updated and shows how much a company has invested.

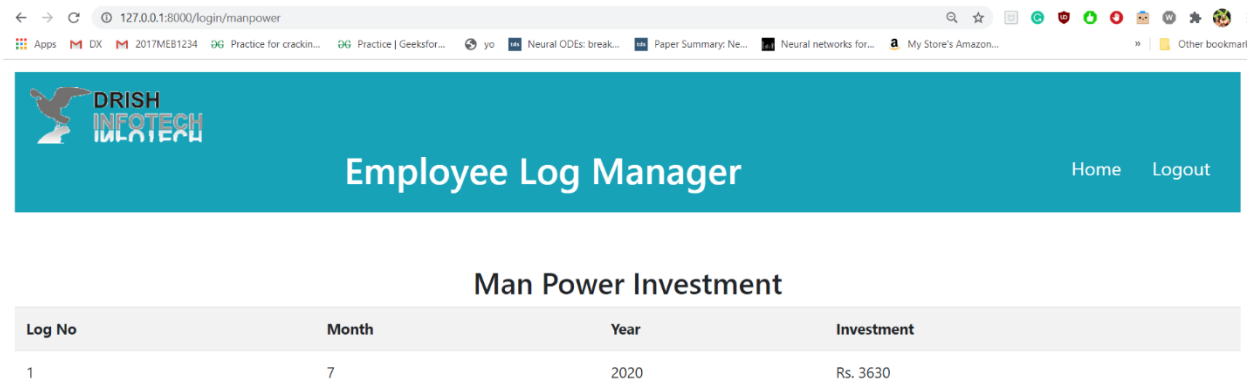

Figure 6: Manpower Investment

2. **Employees**: are restricted to check their timings and log them every day. Employee(s) can check their own timings entered previously. There would be three parameters to enter the time:
    i. Check-In: When employees had started working
    ii. Break: Total number of hours or minutes they took for the break(s)
    iii. Check Out: When the employee is done for the day.

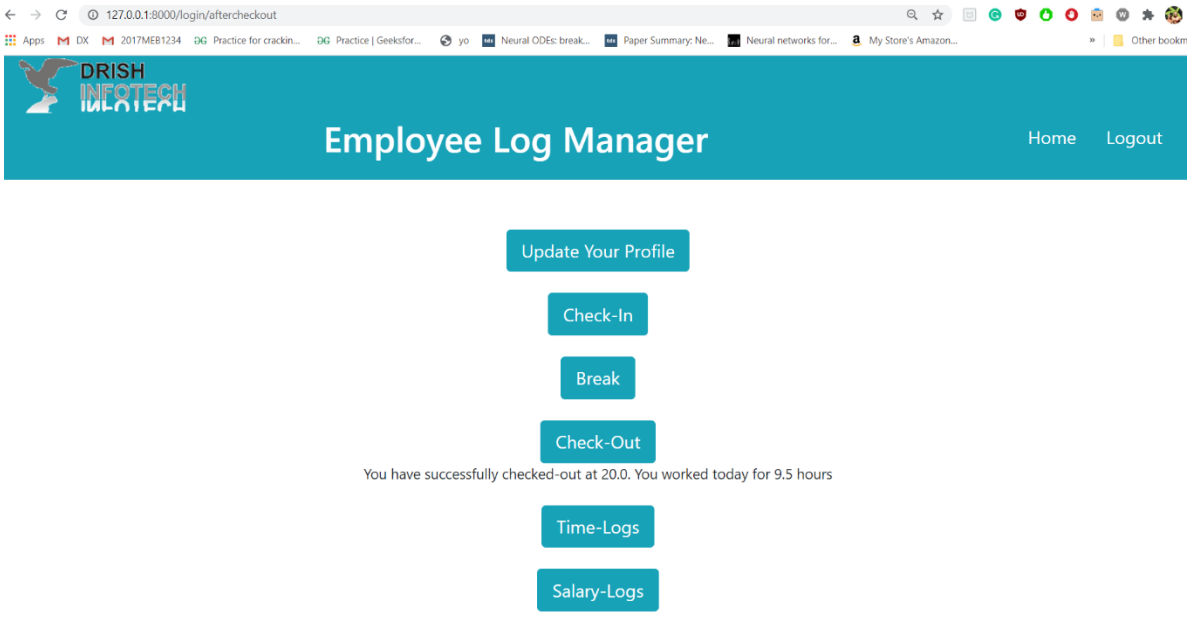

Figure 7: Employee Login

Whenever the user checks out, a new time log is created and salary log and manpower are updated.

API Models

Figure 8: API Models

These are the backend tables which store the data and they have the following fields. The Id is the default primary key in every model.

1. **Employees:**
    a. Username (from User Table)
    b. Employee Code
    c. Name
    d. Email
    e. Hourly Rate

2. **Entry Logs**
    a. Check-In
    b. Break
    c. Check-Out
    d. Date
    e. Month
    f. Year
    g. Employee Code (Foreign key reference from Employees Table)
    h. Duration

3. **Salary**
    a. Employee Code (Foreign key reference from Employees Table)
    b. Month
    c. Year
    d. Duration
    e. Salary

4. **Manpower Investment**
    a. Month
    b. Year
    c. Investment

Similarly, we have the following APIs with CRUD enabled.

1. **Employee Module and Authentication API:**
   It includes the User Management System for employees including all the validations.
2. **Time Log API**
   Logging time by employees, fetching, modifying, or deleting time logs
3. **Salary Calculator API**
   Calculate the Salary of each employee
4. **MPI Calculator API**
   Total investment is done by the company on its employees.

Conclusion

This project can be utilized by companies to track their employees. Also, we can extend this project by adding graphical representation of Manpower vs Month, Salary vs Month etc. which would provide us with better comparisons regarding monthly investments , employee performance and other parameters, and accordingly by analysing the data, the company can take important steps.

Project made by:

Ritvik Munjal
IIT Ropar
Intern Time Period: 4 May, 2020 – 8 July, 2020

Mentor:

Ripundeep Singh Gill
Senior Programmer Analyst (Python)
Drish Infotech Limited
Sector 34 A, Chandigarh, 160022