Drish Infotech Ltd
Top Floor, SCO 104-106, Sector 34 A
Chandigarh, India, 160022
(+91) 0172 266 5653

Employee Log Manager

26th May 2020

OVERVIEW

Employee log manager is a web application to keep track of time spent by employees and calculate their salary. It would interact through the web services in order to store the required and produced information. This would contain 4 modules:

1. User handling and Authentication
2. Time log of employee(s)
3. Salary Calculator.
4. MPI (Man Power Investment) Calculator.

GOALS

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 -** Framework to create web services and business layer
2. **PostgreSQL -** Database Management System to store information.
3. **HTML5, CSS, JQuery -** Front end design and communication with the business layer.
4. **CSV** || **Pandas** || **Matplotlib** || **Numpy** || **Math -** Any of the package to handle CSV files, calculations or graphical representation, if required.
5. **Python 3.6.5 -** Programming Language

DESCRIPTION

1. There would be two types of users:
   a. **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. One has all the rights to perform CRUD operation on anything.
   b. **Employee:** An employee would be subjected to enter their office timings only. 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.

2. 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.
   d. Aggregation of the salary

3. Register the employees with the following information:
   a. Employee Code (Unique)
   b. Name of the employee
   c. Email of the employee (Unique)
   d. Username of the employee (Unique)
   e. The password of the employee
   f. Hourly Rate (10, 100, 500, etc.) [In Rupees]

4. Create a Django superuser in order to create an Administrator.
5. There would be following panels in the front end:
   a. User Management System
      i. Registration
      ii. Password Change
      iii. Login Form

b. Employee Dashboard
   i. Time logging
   ii. Time logged in the previous months
   iii. Salary earned in the previous months
   iv. Periodic graphical representation of salary and time.
c. Admin Dashboard
   i. The time and salary of each employee logged so far.
   ii. Employees listing with CRUD enabled
   iii. Salary Calculator
   iv. Man Power Investment Calculator
   v. Graphical Representation of the time logged, salary earned by employees (Each and All included).
   vi. Graphical representation of Man Power Investment

MILESTONES

Planning of Project

It will contain the plotting of the complete project including web services and designing and development of the application.

Schema Design

The overall design of the tables which are going to be used in the project.

Front End Design

Design of all the HTML pages going to be used as front end design using CSS and applying JavaScript/Jquery on front end design wherever required.

Business Layer

Integrate the front end with Django's view and both should be able to communicate with each other.

Initial Environment Setup and Administration.

The initial environment setup includes the installation of dependencies as well as skeletal of the project.

Employee Module and Authentication API [CRUD || Dependencies]

It includes the User Management System for employees including all the validations.

Time Log API [CRUD || Dependencies]

Logging time by employees, fetching, modifying, or deleting time logs.

Salary Calculator API [CRUD || Dependencies]

Calculate the Salary of each employee.

MPI Calculator API [CRUD || Dependencies]

Total Investment is done by the company on its employees. Graphical representation would be appreciated.

Unit Test Cases of API's

Unit test case for each function or class developed for desired functionality.

Integrate API with the business layer

Integrate API with the business layer so that it can act as middleware between the API and the frontend.

Key Points

1. More changes may be subjected to the project while development. Suggestions from the intern are most welcome.
2. The intern is subjected to complete the whole project by the end of the Internship
3. Any delay that occurred due to the absenteeism/leaves would be clear responsibility of the Intern.
4. The intern has entitled to give a seminar at the end of the project.

SYNOPSIS MADE BY:

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