ONLINE HUMAN RESOURCE MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

MANOJKUMAR K.B (713816104049)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRI RAMAKRISHNA INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2020

ACKNOWLEDGEMENT

The satisfaction and euphoria of completion of any task would be incomplete without mentioning the people who made it possible.

Primarily, I would like to express my gratitude to our beloved Principal **Dr.M.Paulraj M.E., Phd.,** for providing an opportunity and necessary facilities for the accomplishment of the project.

I would like to express my special thanks and gratitude to our Head of the Department, **Dr. R.M.S Parvathi M.E., Ph.D.,** Department of Computer Science and Engineering, for imparting her knowledge and experiences.

I would like to convey my sincere thanks to my Project Supervisor **Ms. D. Betteena Sheryl Fernando M.E.,** Assistant Professor, Department of Computer Science and Engineering, for her valuable suggestions and guidance for the completion of the project. She helped me understand the intricate issues involved in the project making besides effectively presenting it.

I take immense pleasure to express my heartfelt thanks to the faculty members of the Department of Computer Science and Engineering for their constant support and guidance at all stages of completion of the project.

I would like to thank all lab assistants and my student colleagues without whom this project work would not have been completed successfully.

Finally, I would like to express my gratitude to my parents, whose love and encouragement have a great influence throughout my studies.

ABSTRACT

Human Resource Management system is a distributed application, developed to maintain the details of employee of any organization. This Application is to maintain a set of records about every employees in the company. This application stores all the provided personal details of the employees along with the Leave records, Performance analysis and the company provided benefits for every individual employees in the company. These records are secured and only the authorized persons can access the records. The Performance analysis is calculated based on the working hours of the employee, the number of tasks completed on time, efficiency and quality of the work and success rate of the work done. The performance will be evaluated and displayed in the dashboard. This application reduces the time consumption for the written process and the manual work and it also improves the refresh rate compared to other applications. This application can be developed using React Js, CSS, Html as a front end and Python, Django as a backend.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**i**
	LIST OF FIGURES	**v**
	LIST OF ABBREVIATIONS	**vi**
1	**INTRODUCTION**	**1**
	1.1 WEB APPLICATION	1
2	**LITERATURE SURVEY**	**6**
3	**SYSTEM ANALYSIS**	**11**
	3.1 EXISTING SYSTEM	11
	3.2 PROPOSED SYSTEM	11
4	**SYSTEM SPECIFICATION**	**13**
	4.1 HARDWARE REQUIREMENTS	13
	4.2 SOFTWARE REQUIREMENTS	13
5	**SOFTWARE DESCRIPTION**	**14**
	5.1 WEB BROWSER	14
	5.2 REACT JS	14
	5.2.1 Class component	14
	5.2.2 Functional Component	14
	5.2.3 React Router	15
	5.2.4 Life cycle Hooks	15
	5.4 HTML	17
	5.5 CSS	18
	5.6 JAVASCRIPT	19
	5.7 ANT DESIGN	20
	5.8 AXIOS	21
	5.9 MYSQL SERVER	22

5.10 DJANGO WEB DEVELOPMENT FRAMEWORK 30

6 PROJECT DESCRIPTION 32

6.1 PROBLEM DEFINITION 32

6.2 INTRODUCTION TO PROPOSED SYSTEM 32

6.3 MODULE DESCRIPTION 33

6.3.1 Module 1 - Employee's Goal Tracking 33

6.3.2 Module 2 - Database creation of Employee's 35

6.3.3 Module 3 - Leave Record 36

7 SYSTEM IMPLEMENTATION 40

7.1 SYSTEM ARCHITECTURE 40

8 RESULTS AND DISCUSSION 43

8.1 RESULTS AND CONCLUSION

8.2 SNAPSHOTS 44

9 CONCLUSION AND FUTURE ENHANCEMENTS 50

9.1 CONCLUSION 50

9.2 FUTURE ENHANCEMENTS 50

APPENDIX 51

SOURCE CODE 51

10 REFERENCES 65

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NUMBER
6.1	EMPLOYEE GOAL TRACKING	33
6.2	DATABASE CREATION FOR EMPLOYEE	35
6.3	LEAVE RECORD	36
7.1	MODEL VIEW CONTROL	41

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
API	Application Programming Interface
CSV	Comma Separated Values
CSS	Cascading Style Sheet
DBMS	Database Management System
GPL	General Public License
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JS	JavaScript
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
MVT	Modern-View-Template
MVC	Model-View-Controller
ORM	Object-Relational Mapping
SQL	Structure Programming Language
W3C	World Wide Web Consortium
XHTML	Extensible Hypertext Markup Language

CHAPTER 1

INTRODUCTION

A human resources management system (HRMS) is a form of human resources (HR) software that combines a number of systems and processes to ensure the easy management of human resources, business processes and data. Human resources software is used by businesses to combine a number of necessary HR functions, such as storing employee data, benefits administration, and keeping track of leave records. It ensures everyday human resources processes are manageable and easy to access. It merges human resources as a discipline and, in particular, its basic HR activities and processes with the information technology field, whereas the programming of data processing systems evolved into standardized routines and packages of enterprise resource planning (ERP) software. On the whole, these ERP systems have their origin from software that integrates information from different applications into one universal database. The linkage of its financial and human resource modules through one database is the most important distinction to the individually and proprietarily developed predecessors, which makes this software application both rigid and flexible.

Human resource information systems provide a means of acquiring, storing, analyzing and distributing information to various stakeholders. HRMS enable improvement in traditional processes and enhance strategic decision-making. The wave of technological advancement has revolutionized each and every space of life today, and HR in its entirety was not left untouched. Early systems were narrow in scope, typically focused on a single task, such as improving the payroll process or tracking employees' work hours. Today's systems cover the full spectrum of tasks associated with human resources departments, including tracking and improving process efficiency, managing organizational hierarchy, and simplifying financial transactions of all types. In

short, as the role of human resources departments expanded in complexity, HR technology systems evolved to fit these needs.

1.1 Web Application

A web application (or "web app" for short) is any computer program that performs a specific function by using a web browser as its client. The application can be as simple as a message board or a contact form on a website or as complex as a word processor or a multi- player mobile gaming app.

Most web applications are based on the client-server architecture where the client enters information while the server stores and retrieves information. Internet mail is an example of this, with companies like Google's Gmail and Microsoft's Outlook offering web- based email clients.

Web applications can provide the same functionality and gain the benefit of working across multiple platforms. For example, a web application can act as a word processor, storing information in the cloud and allowing you to 'download' the document onto your personal hard drive.

Fundamental steps in Web application

There are numerous steps in the web site design and development process. From gathering initial information, to the creation of your web site, and finally to maintenance to keep your web site up to date and current.

The exact process will vary slightly from designer to designer, but the basics are the same.

- Requirements Gathering
- Analysis
- Design
- Development
- Testing and Delivery
- Maintenance

Requirements Gathering

The first step in designing a successful web site is to gather information. Many things need to be taken into consideration when the look and feel of the site is created.

This first step is actually the most important one, as it involves a solid understanding of the company it is created for. It involves a good understanding of the business goals and dreams are, and how the web can be utilized to help to achieve those goals.

Analysis

Using the information gathered from phase one, it is time to put together a plan for the web site. This is the point where a site map is developed.

The site map is a list of all main topic areas of the site, as well as sub-topics, if applicable. This serves as a guide as to what content will be on the site, and is essential to developing a consistent, easy to understand navigational system. The end-user of the web site– aka customer – must be kept in mind when designing the site. These are, after all, the people who will be learning about service or buying the product. A good user interface creates an easy to navigate web site, and is the basis for this.

During the analysis phase, the web designer will also help to decide what technologies should be implemented. Elements such as what CMS (Content Management System) such as word press to incorporate, will any contact forms be needed, etc. are discussed when planning to develop web site.

Design

Drawing from the information gathered up to this point, it’s time to determine the look and feel of the website.

Target audience is one of the key factors taken into consideration. A site aimed at teenagers, for example, will look much different than one meant for a financial institution. As part of the design phase, it is also important to incorporate elements such as the company logo or colors to help strengthen the identity of the company on the web site.

Development

The developmental stage is the point where the web site itself is created. At this time, the web designer will take all of the interior pages. The shell serves as a template for the content pages of the site, as it contains the main navigational structure for the web site. Once the shell has been created, the designer will take the content and distribute it throughout the individual graphic elements from the prototype and use them to create the actual, functional site. This is typically done by first developing the home page, followed by a "shell" for site, in the appropriate areas.

Elements such as the CMS (Content Management System) like word press, interactive contact forms, or ecommerce shopping carts are implemented and made functional during this phase, as well.

Testing and Delivery

At this point, the web designer will attend to the final details and test the web site. They will test things such as the complete functionality of forms or other scripts, as well last testing for last minute compatibility issues (viewing differences between different web browsers), ensuring that the web site is optimized to be viewed properly in the most recent browser versions.

A good web designer is one who is well versed in current standards for web site design and development. The basic technologies currently used are HTML and CSS (Cascading Style Sheets). As part of testing, the designer should check to be sure that all of the code written for the web site validates. Valid code means that the site meets the current web development standards – this is helpful when checking for issues such as cross-browser compatibility as mentioned above.

Maintenance

The development of the web site is not necessarily over, though. One way to bring repeat visitors to the site is to offer new content or products on a regular basis. Most web designers will be more than happy to continue working together, to update the information on web site. Many designers offer maintenance packages at reduced rates, based on how often anticipate making changes or additions to the web site.

A web site driven by a CMS gives the ability to edit the content areas. The user also given access to a back-end administrative area, where the user can use an online text editor (similar to a mini version of Microsoft Word).

CHAPTER 2

LITERATURE SURVEY

Xiaobing Zhang , Xiaobing Zhang "The Effects on Human Resource Management System and Organizational Performance: A Role of Absorption" International Conference on Wireless Communications, Networking and Mobile Computing, 2008.

It constructs a research model integrating human resource management system (HRMS) and organizational performance (OP) from the perspective of absorptive capacity. Based on a theory development, we propose hypotheses between HRMS, absorptive capacity and organizational performance. We obtain 221 effective samples in a snowball sampling manner and testify the sample data with SEM, part of the hypotheses being supported. Both HRMS and organizational performance integration research model based on absorptive capacity have been partially supported empirically. The research direction for future theory development is discussed.

Suraj Francis Noronha , Paul G Aquinas , Aruna Doreen Manezes "Implementing Employee Management System" IEEE FORUM International Conference, 2016.

The implementation of any system requires meticulous planning and a careful execution. When it comes to human resource, the system implementation becomes even more challenging. In this paper an effort has been made to identify the various parameters that need to be taken into account while implementing a performance management system (PMS) and to list the various causes for the failure of PMS in an organization during its implementation. A scoping review method of the literature has been used to study and classify the various causes for the failures. The limitation of this paper is that it uses only the literature which is published and available from the

online database journals. The practical implication of the study is that the managers who are involved with the implementation of the PMS can use these as a checklist, to be better prepared and avoid making the same mistakes again. This paper is a compilation of the scattered literature on the problems faced in the implementation of the PMS. It may be used as a guide to avoid any pitfalls while implementing new systems or while modifying the existing system.

Vivin.S, Swathy.S, Sruthi.C "Employment Management System" International Research Journal of Engineering and Technology (IRJET), 2016

The primary epitome of the system is to allow the up-gradation and combination od smart card system and smart monitoring system together that can monitor the position of a particular person. This combination can be achieved via the use of radio frequencies and its applications. This system has developed into a drastic advancement in the field of radio frequencies. The two important parameters are RF-ID trans-receiver and tags. The employee will have wear an identity card equipped with RF-ID. The trans-receiver form the bridge between the database and the employee the active RF-ID trans-receiver transmits radio frequencies up to a certain distance and creates an electro magnet area depending upon the wavelength of the trans-receiver. The RF-ID tag is a passive device that takes up power from the electromagnetic charge and transmits the identification serial number to the transrceiver. The latter then transfers the identification serial number along with the unique identification number if he trans-receiver to the master system containing the database. Thus the trans- receiver is programmed with Structured Query Language (SQL) to avoid collision of identification numbers. This can be implemented via a monitoring system.

Anand Kotasthane, Akshay Bahade, Kaustubh Chinchalkar, Prasad Gawande, Ms. P K Karmore "Intranet Automation of Human Resource Management System" IOSR Journal of Computer Engineering (IOSR-JCE), 2013.

The module enhances good communication facilities between employees and HR Administrator. To allow the HR of an organization to update the employee details whenever there is a change in the employee profile pertaining to that organization. The objective of the project is to create a tool for HR department which helps it in giving confirmation to a recruited employee. HR department will have a CV and record of an employee's Interview in its database. Confirmation will be made on the basis of this data and a process known as SWOT.

All communication between the Employee phone and the admin is done through web network technology. This proposed application is quite user-friendly as it contributes in giving accurate digits in managing employees of the company by saving time, reducing manager efforts; avoids the unnecessary use of company phones which provides to the Employee for their official work during working hours. The proposed android application connects the centralized server with employee phones. The main idea of our paper is to provide an aid to Managers to navigate their all company .

Bandar khalaf Alharthey, Amran Rasli "The use of Human Resource Management Systems in the Saudi market" Asian Journal of Business Ethics,

The goal of the study was to investigate the current situation with Human Resource Management System in the Saudi market on the basics of survey conducted among 100 organization . Their HR and IT experts were to fill out a questionnaire that allowed receiving their experts opinon and make conclusion .

Considering the HR system usage in the country it the course of study eight hypothysys were investigated and proved the number of companies uses HRMS system does not exceed a half of the staff companies uses system development by their specialists for Hrms . Companies have high development potential despite independent . Their result have shown that the Saudi market is ready to implement of the new technology while the level of satisfication by the current solution is comparatively low and the new development preferably on the basics of Saas model

Xing Xu,Hao Hu,Na Hu,Lin Xiao,Weiqin Ying "Human Resource Management System Based on Factory Method Design Pattern" International Conference on Information Computing and Applications, 2015.

Analyzing the human resource management system many suitable difference in the user interface design reuse in the large number of repetitative works for developers . In order to avoid such situation design pattern as an important software reusable technology is broadly applied to many kinds of information management platform for effecitively development costs .Their paper introduced to kind of factory method design pattern as essential aid of the system design . The factory method is imported in HRMS analysis and design the UI .It shows the better reusability and scalablity and provide the strong support for meeting the growing needs of different business in HRMS so that people can more simply and conveniently reuse successful design and architecture and provides further details of its application in the UI of HRMS

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

The present system using a very basic excel sheet and primitive technology of webpage. The decision for appraisal of assigning next task are not properly projected.

Disadvantage of Existing System

- Need for extra effort.
- Data maintenance may not be accurate.
- Longer time to find suitable employee.
- Danger of losing files in some cases.
- It is not User Friendly.
- Increasing time consumption.
- Low refresh rate of page switching.

3.2 Proposed System

HR Management is to simplify the overall process of managing the details of the company. This is also a web application which will be hosted separately but will be linked on the career section of the company website.

The following are HR Management key features :

- HR can add/ modify/ delete Employee basic details by adding new employee or modify the existing employee details or delete the employee details

- List of employee with their details which presented in table with row wise its easy to maintain the record

- HR can maintain the leave records of Employee and list of approved, rejected and pending leave records of employee on table. And HR can view how many employees are present/leave in the company

- HR can view total no of goals and milestone of employees and track them. Its easy to absorb the individual performance of the employee

- HR can view how many goals and milestone are completed and pending. By this number of man power and effort can be managed

- HR can view completed and pending goals & milestones in department wise ReactJs induces modularity and the features of re-usability, this leads to higher refresh rate and secure method to store data .

- Hence, Reactjs is used to create the user interface with different module.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 Hardware Requirements

Processor	:	Intel i3 or higher
RAM	:	4 GB or higher

4.2 Software Requirements

Operating system	:	Windows 7 or higher
Coding Language	:	REACT JS, Python, Django, HTML, CSS,
ANT Design Data Base	:	MYSQL

CHAPTER 5 SOFTWARE

DESCRIPTION

5.1 WEB BROWSER

A web browser (commonly referred to as a browser) is a software application for retrieving, presenting and traversing information resources on the World Wide Web. An information resource is identified by a Uniform Resource Identifier (URI/URL) that may be a web page, image, video or other piece of content. Hyperlinks present in resources enable users easily to navigate their browsers to related resources.

Although browsers are primarily intended to use the World Wide Web, they can also be used to access information provided by web servers in private networks or files in file systems. The most popular web browsers are Chrome, Edge (preceded by Internet Explorer), Safari, Opera and Firefox.

5.2 REACT JS

ReactJS is a component based library (not a framework!!)which is used to develop interactive UI's. Currently it is the most popular front-end JavaScript library. It deals with View in the MVC(Model - View - Controller). So, if you are dealing with the applications where data keeps on changing in real time, you should go for React. As in React, the application is composed of different components. Each time any data is added , it will automatically update the specific component whose state has actually changed.

These components are independent building blocks which are integrated together to build the entire application. To understand this better, imagine UI as a tree. Here the starting component becomes the root and each of the independent pieces becomes branches, which are

further divided into sub-branches. This keeps our UI organised and moreover, it allows the data and state changes to logically flow from the root to branches and then to sub- branches. One component can refer to other components as well.

Moreover, React is much faster as compared to the JavaScript frameworks. It makes use of the virtual DOM which minimizes the memory consumption and the DOM manipulation expenses.

5.2.1 Class Component

A class component is a more featured way to define a React component. It also acts like a function that receives props, but that function also considers a private internal state as additional input that controls the returned JSX.

5.2.2 Functional Component

The function component receives an object of properties which is usually named props. It returns what looks like HTML, but is really a special JavaScript syntax called JSX.

5.2.3 React Router

React Router is the standard routing library for React. React Router keeps your UI in sync with the URL. It has a simple API with powerful features like lazy code loading, dynamic route matching, and location transition handling built right in.

5.2.4 Life Cycle Hooks

- **componentWillMount** is executed before rendering, on both the server and the client side.
- **componentDidMount** is executed after the first render only on the client side. This is where AJAX requests and DOM or state updates should occur. This method is also used for integration with other JavaScript frameworks and any functions with delayed execution such as setTimeout or setInterval. We are using it to update the state so we can trigger the other lifecycle methods.
- **componentWillReceiveProps** is invoked as soon as the props are updated before another render is called. We triggered it from setNewNumber when we updated the state.
- **shouldComponentUpdate** should return true or false value. This will determine if the component will be updated or not. This is set to true by default. If you are sure that the component doesn't need to render after state or props are updated, you can return false value.
- **componentWillUpdate** is called just before rendering.
- **componentDidUpdate** is called just after rendering.
- **componentWillUnmount** is called after the component is unmounted from the dom. We are unmounting our component in main.js.

5.3 HTML

HTML is the standard markup language for creating web pages and web applications. With CSS and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as ... surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

5.4 CSS

CSS is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most website to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications. CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader),and on Braille-based tactile devices

. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified. Changes to the graphic design of a document(or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents. The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

5.5 JavaScript (JS)

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype- based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web.[citation needed] It is used to make dynamic web pages interactive and provide online programs, including video games. The majority of websites employ it [citation needed], and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. Each of the many JavaScript engines represent a different implementation of JavaScript, all based on the ECMAScript specification, with some engines not supporting the spec fully, and with many engines supporting additional features beyond ECMA. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype- based) programming styles. It has an API for working with text, arrays, dates, regular expressions,

and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded. Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server- side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets. Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

5.6 ANT Design

In the process of internal desktop applications development, many different design specs and implementations would be involved, which might cause designers and developers difficulties and duplication and reduce the efficiency of development. After massive project practice and summaries, Ant Design, a design language for background applications, is refined by Ant UED Team, which aims to uniform the user interface specs for internal background projects, lower the unnecessary cost of design differences and implementation and liberate the resources of design and front-end development.

Ant Design which is specially created for internal desktop applications, is committed to improving the experience of users and product designers. User interface designers and user experience designers, collectively, are considered as product designers, and the boundaries of product managers, interaction designers, visual designers, front-end developers and develop engineers are blurred. Taking advantage of unitary specifications, Ant Design makes design and prototype more simple and accessible for all project members, which

comprehensively promotes experience and development efficiency of background applications and products.

5.6.1 DatePicker

This property provide an additional time selection. When showTime is an Object, its properties will be passed on to built-in TimePicker .

When To Use

By clicking the input box, you can select a date from a popup calendar.

Syntax:import {DatePicker } from 'antd'; const { RangePicker } = DatePicker;

5.6.3 Table

A table displays rows of data.

When To Use

- To display a collection of structured data.
- To sort, search, paginate, filter data.

Syntax

```
import {Table} from 'antd';
<Table dataSource={dataSource} columns={columns} />
```

5.6.4 Input

A basic widget for getting the user input is a text field. Keyboard and mouse can be used for providing or changing data.

When To Use

- A user input in a form field is needed.
- A search input is required.

Syntax

```
import {Input } from 'antd';
<Input />
```

5.6.5 Message

Display global messages as feedback in response to user operations.

When To Use

- To provide feedback such as success, warning, error etc.
- A message is displayed at top and center and will be dismissed automatically, as a non-interrupting light-weighted prompt.

Syntax

```
import {message } from
'antd'; message.();
```

5.6.6 Form

Form is used to collect, validate, and submit the user input, usually contains various form items including checkbox, radio, input, select, and etc.

Syntax

```
import { Form } from 'antd';
<Form>
…….
</Form>
```

5.6.7 Select

Select component to select value from options.

When To Use

- A dropdown menu for displaying choices - an elegant alternative to the native <select> element.
- Utilizing Radio is recommended when there are fewer total options (less than 5).

Syntax

```
import { Select } from 'antd';
const Option = Select.Option;
<Select >
    <Option >..... </Option>
</Select>
```

5.7 AXIOS

Axios Component for React with child function callback. This is intended to allow in render async requests.

Features

- Same great features found in Axios
- Component driven
- Child function callback (error, response, isLoading, makeRequest, axios) => { }
- Auto cancel previous requests
- Debounce to prevent rapid calls.
- Request only invoked on prop change and isReady state.
- Callback props for onSuccess, onError, and onLoading
- Supports custom axios instances through props or a <AxiosProvider ... >
- Create your own request components wrapped using the withAxios({options})(ComponentToBeWrapped)HoC

Methods

- **Axios.get()**

  To fetch the value from Database.

- **Axios.Put()**

  Modifying the existing record in the database.

- **Axios.Post()**

  Insert the new record into the database.

- **Axios.Delete()**

  Delete the Existing records in the database.

- **Axios.patch()**

  Modifying the required fields in the database.

5.8 MySql Server

MySQL is a free, open-source database management system (DBMS for short). A DBMS is a system that manages databases and connects them to software. For example, a MySQL database can be used to run a website, to run the database of an ERP or any other software. MySQL is a powerful, free open-source database management system that has been around for years. It is very stable and has a big community that helps maintain, debug and upgrade it. MySQL might not be as popular for larger systems that will mostly run on Microsoft SQL Server or Oracle. These proprietary DBMS are more scalable, have more resources available on the market and have more advanced features that MySQL.

Components In MySql

These are the main MySQL components:

- MySQL Server. It is the database server. It processes all queries and manipulates databases and tables.
- MySQL Clients. These are programs that communicate with the server.
- Data Directory. This is where MySQL stores the databases.
- Storage Engine. Manages how the information is organized, stored and accessed. MySQL uses a pluggable storage engine architecture. Example storage engines are MyISAM, InnoDB, Memory or Csv.

MySQL Server

The MySQL Server manages the changes in the data directory, which contains the databases and tables. It also accepts connections from clients and manages their access to the databases.

These are the most important MySQL Server programs:

- mysqld: the database server
- mysqld_safe: a script to start mysqld that adds some safety features. It

is especially useful when server starts with errors.

- mysql.server: a Unix script that starts mysqld_safe

mysqld_multi: Manages multiple mysqld servers in the same machine

MySQL Clients

- Mysql is an interactive program where you write SQL statements and execute in the server. It then presents the results in a tabular format. It can also run as a batch script.
- mysqladmin is an administrative program that can configure, monitor or shut down the server
- mysqldump is a tool to back up the database
- mysqlcheck and myisamchk are tools to check the integrity of tables and help repair them.

5.9 Django Web Development Framework

Django:

Django is a free and open-source web framework, written in Python, which follows the model-view-template (MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, less code, low coupling,

Rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include the Public Broadcasting Service, Instagram, Mozilla, The Washington Times, Disqus, Bitbucket, and Nextdoor. It was used on Pinterest, but later the site moved to a framework built over Flask.

Features of Django

Following are some of the important features that make Django the first choice of Python web developers.

Components : Despite having its own nomenclature, such as naming the callable objects generating the HTTP responses "views" ,the core Django framework can be seen as an MVC architecture. It consists of an object-relational mapper (ORM) that mediates between data models (defined as Python classes) and a relational database("Model"), a system for processing HTTP requests with a web templating system ("View"), and a regular-expression-based URL dispatcher ("Controller").

Also included in the core framework are:

- A lightweight and standalone web server for development and testing.
- A form serialization and validation system that can translate between HTML forms and values suitable for storage in the database.
- A template system that utilizes the concept of inheritance borrowed from object- oriented programming.
- A caching framework that can use any of several cache methods support for middleware classes that can intervene at various stages of request processing and carry out custom functions. An internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals. An internationalization system, including translations of Django's own components into a variety of languages. A serialization system that can produce and read XML and/or JSON representations of Django model instances. A system for extending the capabilities of the template engine an interface to Python's built-in unit

Bundled applications : The main Django distribution also bundles a number of

applications in its "contrib" package, including:

- An extensible authentication system.
- The dynamic administrative interface.
- Tools for generating RSS and Atom syndication feeds.
- A site's framework that allows one Django installation to run multiple websites, each with their own content and applications. Tools for generating Google Sitemaps.
- Built-in mitigation for cross-site request forgery, cross-site scripting, SQL injection, password cracking and other typical web attacks, most of them turned on by default.
- A framework for creating GIS applications.

Extensibility : Django's configuration system allows third party code to be plugged into a regular project, provided that it follows the reusable app conventions. More than 2500 packages are available to extend the framework's original behavior, providing solutions to issues the original tool didn't tackle: registration, search, API provision and consumption, CMS, etc. This extensibility is, however, mitigated by internal components dependencies. While the Django philosophy implies loose coupling, the template filters and tags assume one engine implementation, and both the auth and admin bundled applications require the use of the internal ORM. None of these filters or bundled apps are mandatory to run a Django project, but reusable apps tend to depend on them, encouraging developers to keep using the official stack in order to benefit fully from the apps ecosystem.

- Server arrangements : Django can be run in conjunction with Apache, Nginx using WSGI, Gunicorn, or Cherokee using flup (a Python module). Django also includes the ability to launch a FastCGI server, enabling use behind any web server which supports FastCGI, such as Lighttpd or Hiawatha. It is also possible to use other WSGI-compliant web servers. Django officially supports four database backends: PostgreSQL, MySQL, SQLite, and Oracle. Microsoft SQL Server can be used with django-mssql on Microsoft operating systems, while similarly external backends exist for IBM Db2, SQL Anywhere and Firebird. There is a fork named django-nonrel, which supports NoSQL databases, such as MongoDB and Google App Engine's Datastore.
- Django may also be run in conjunction with jython on any Java EE application server such as GlassFish or JBoss. In this case django-jython must be installed in order to provide JDBC drivers for database connectivity, which also can provide functionality to compile Django into a .war suitable for deployment. Google App Engine includes support for Django version 1.x.x as one of the bundled frameworks.

Development tools with Django support

For developing a Django project, no special tools are necessary, since the source code can be edited with any conventional text editor. Nevertheless, editors specialized on computer programming can help increase the productivity of development, e.g., with features such as syntax highlighting. Since Django is written in Python, text editors which are aware of Python syntax are beneficial in this regard.

Integrated development environments (IDE) add further functionality, such as debugging, refactoring, and unit testing. As with plain editors, IDEs with support for Python can be beneficial. Some IDEs that are specialized on Python additionally have integrated support for Django projects, so that using

such an IDE when developing a Django project can help further increase productivity.

Understanding Major Advantages of Django

- **Written in Python**

  Django is one of the web frameworks which are written in Python programming language. Hence, it becomes easier for programmers to build web applications with clean, readable, and maintainable code by taking advantage of syntax rules of Python. Also, the developers can easily curtail the development time by building custom web applications without writing additional code.

- **Accelerates custom web application development**

  Django is one of the most mature web frameworks for Python. Its design rules focus extensively on reducing web application development time. The features provided by Django enable developers to build custom web applications rapidly according to varying business requirements. A large percentage of Python programmers even opt for Django when they have to meet both goals and deadlines.

- **Designed as a batteries-included web framework**

  Django is one of the web frameworks that adopt the batteries-included approach. While developing a custom web application, Django provides the resources required by developers out of the box. It provides code for common operations like database manipulation, HTML templating, URL routing, session management, and

security. The batteries included approach help developers to curtail web application development time significantly.

- **Supports MVC programming paradigm**

  Django, like other modern web frameworks, supports model-view-controller (MVC) design rule. The MVC programming paradigm allows programmers to keep a web application's user interface (UI) and business logic layers separated. The approach further helps programmers to simplify and speed up development of large web applications by separating their user interface and business logic layers. Django further allows programmers to reuse the same business logic across multiple projects.

- **Compatible with major operating systems and databases**

  Nowadays, users access web applications on various devices and platforms. Django enhances the accessibility of web applications by supporting major operating systems like Windows, Linux and MacOS. At the same time, the ORM system provided by Django makes it easier for programmers to work with several widely used databases. They can even use the ORM system to perform common database operations and migrate from one database to another without writing additional code.

- **Provides robust security features**

  The built-in security features provided by Django help developers to protect the web applications from a variety of targeted security attacks – cross-site scripting, SQL injection and cross-site request forgery. At the same time, the web framework enhances the security of web applications by preventing common security mistakes related to Python coding.

- **Easy to extend and scale**

  Django has been evolving consistently to enable programmers to build better and modern web applications. At the same time, the Django

developers can easily customize, scale, and extend the web framework by making changes to its decoupled components. They even have option to unplug or replace these decoupled components according to precise requirements of individual projects. Likewise, they can accelerate development of large and complex web applications by wiring up a wide range of components.

CHAPTER 6

PROJECT DESCRIPTION

6.1 Problem Definition

Manual handling of employee information faces a number of challenges. This is clearly seen in procedures such as leave management where an employee is required to fill in a form which may take several weeks or months to be approved. The use of paper work in handling some of these processes could lead to human error. This system will maintain employee information in a database by fully privacy and authority access. The project is aimed at setting up employee information system about the status of the employee, their goals, milestones and the work experience in order to monitor the performance and achievements of the employee through a password protected system.

6.2 Introduction to Proposed System

This system is developed for maintaining each and every employee details in the company. This system consist of three modules.

- **Module 1- Employee's Goal Tracking**
  It consist of total no of employee, goals and milestones.
- **Module 2 - Database Creation of Employee**
  It consist of employee details in table form and HR can add the details of employee and can also delete and edit the employee details.
- **Module 3 – Leave Record**
  It consist of leave record of all employee and can add the details of employee leaves which is approved, pending and rejected.

6.3 Module Description

6.3.1 Module 1 - Employee's Goal Tracking

Figure 6.1 - Employee goal tracking

The above figure 6.1 flow diagram explain the module 1 description.In this module the HR manager will be able to assign tasks and projects to employees, assign a project team and keep track of the progress. HR and admin roles can search all the employees information in the database. Search feature works on specific keywords showing

employee's characteristics, skills and features. The database is created for each and every employee with following fields:

- Employee Id
- Employee Name
- Assessment Year

By using these field values, a goal is created for each and every employee

- Target Value
- Target Completion Date
- Target Achieved Value

Goals in department wise are calculated and represented in the form of bar chart. Another database is created for employee those who have goals for finding the total number of milestone with following fields:

- Employee Id
- Employee Name
- Assessment Year

By using these field values, a milestone is created for employee those who have goals with following fields:

- Target Value
- Target Completion Date
- Target Achieved Value

Now, with these field values and calculate the total number of milestones and find the number of milestones in the completed and pending stages. The completed and pending milestones in department wise are calculated and represented in the form of bar chart.

6.3.2 Module 2 - Database Creation for Employees

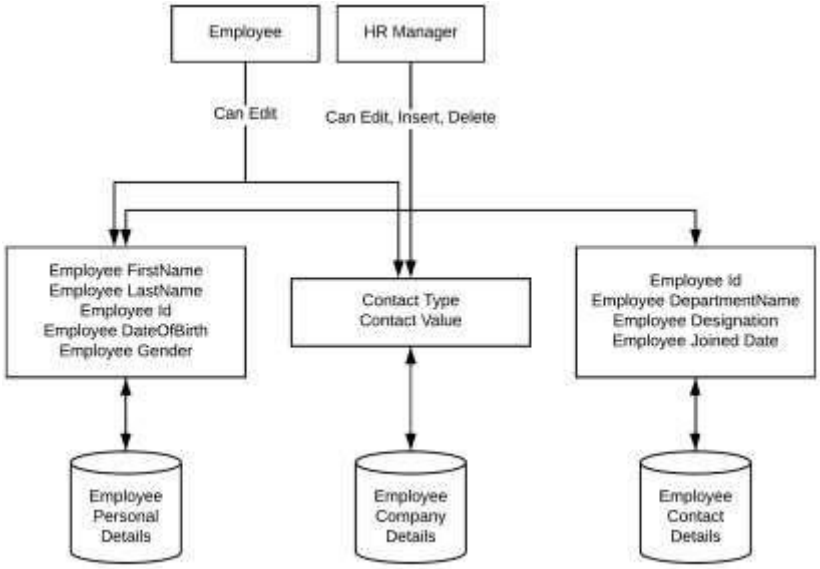


Figure 6.2 - Database creation for Employees

The above figure 6.2 flow diagram explain the module 2 description. In this module, the database is created for employee basic details in which employees will have access to their personal profiles and will be able to edit their details with following fields:

- Employee FirstName
- Employee LastName
- Employee Id
- Employee DateOfBirth

- Employee Gender

Another database is created for employment details with following fields:

- Employee Id
- Employee Department Name
- Employee Designation
- Employee Employment Start Date

Another database is created for employee contact details with following fields:

- Contact Type
- Contact Value

Here, It includes multiple contact type such as mobile number, fax, landline and email and it displayed in table form and it can edit and delete the employee details.

6.3.3 Module 3 - Leave Record

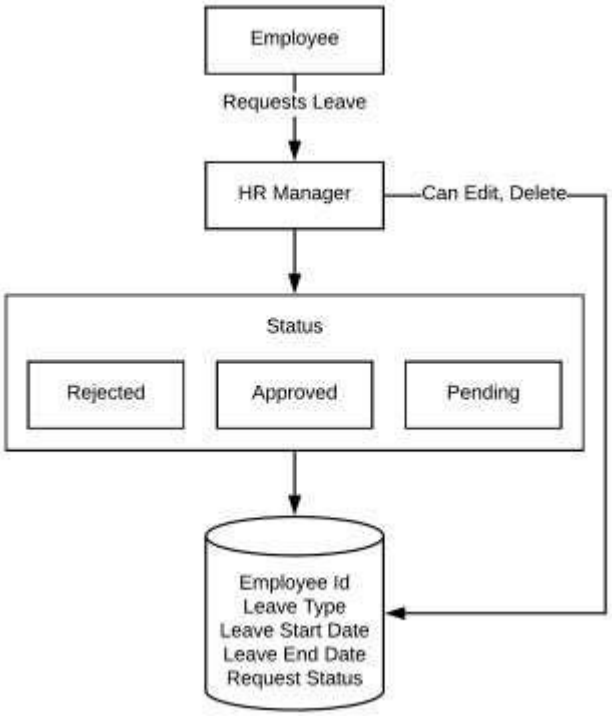


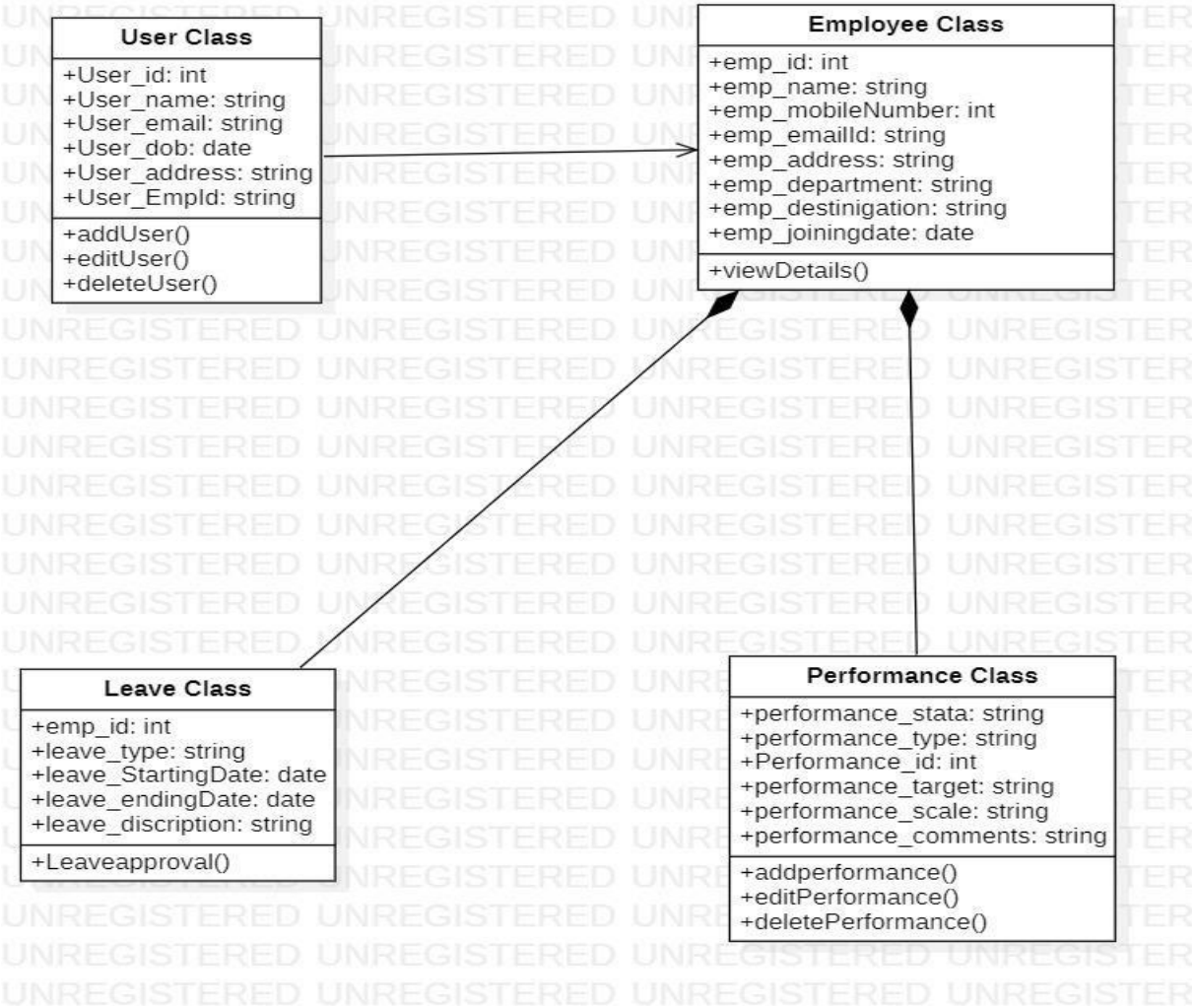
Figure 6.3 - Leave Record

The above figure 6.3 flow diagram explain the module 3 description. The database is created for maintaining employee leave record. It specifies if leave

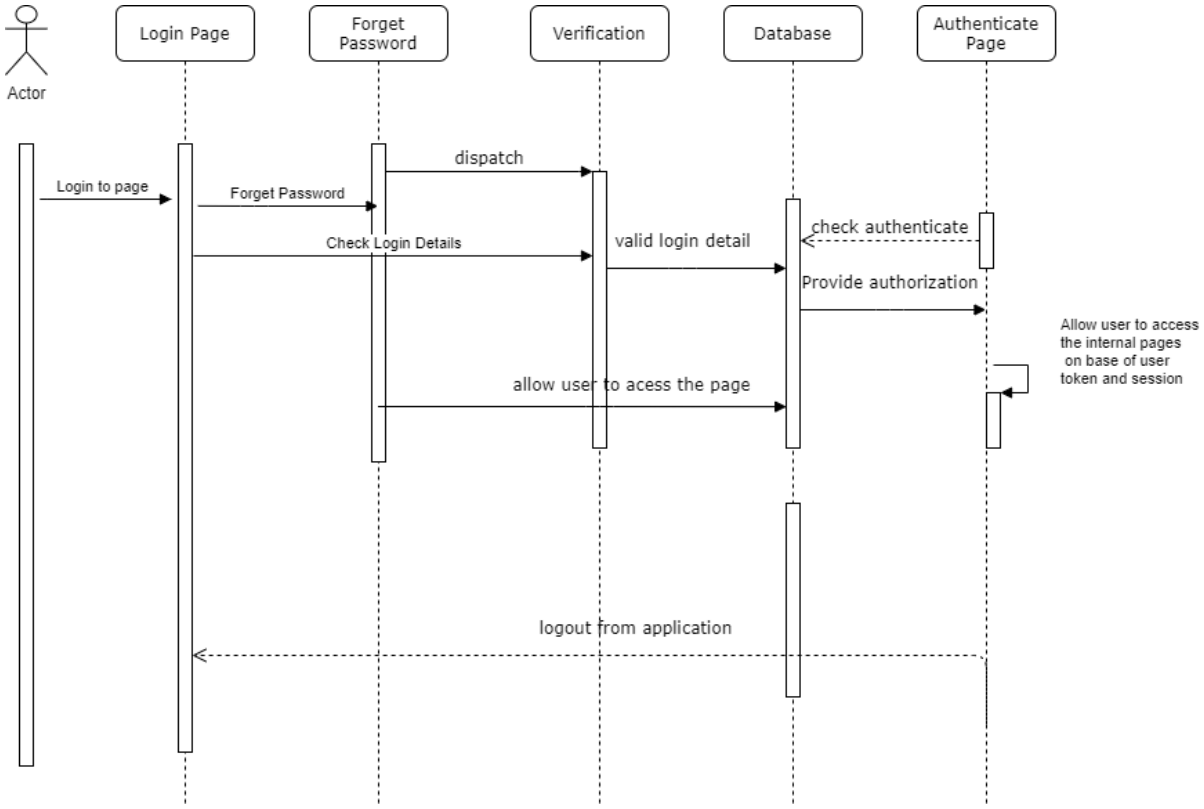
request is approved, pending or rejected. The required following fields are

- Employee Id
- Leave Type
- Leave start Date
- Leave End Date
- Request status

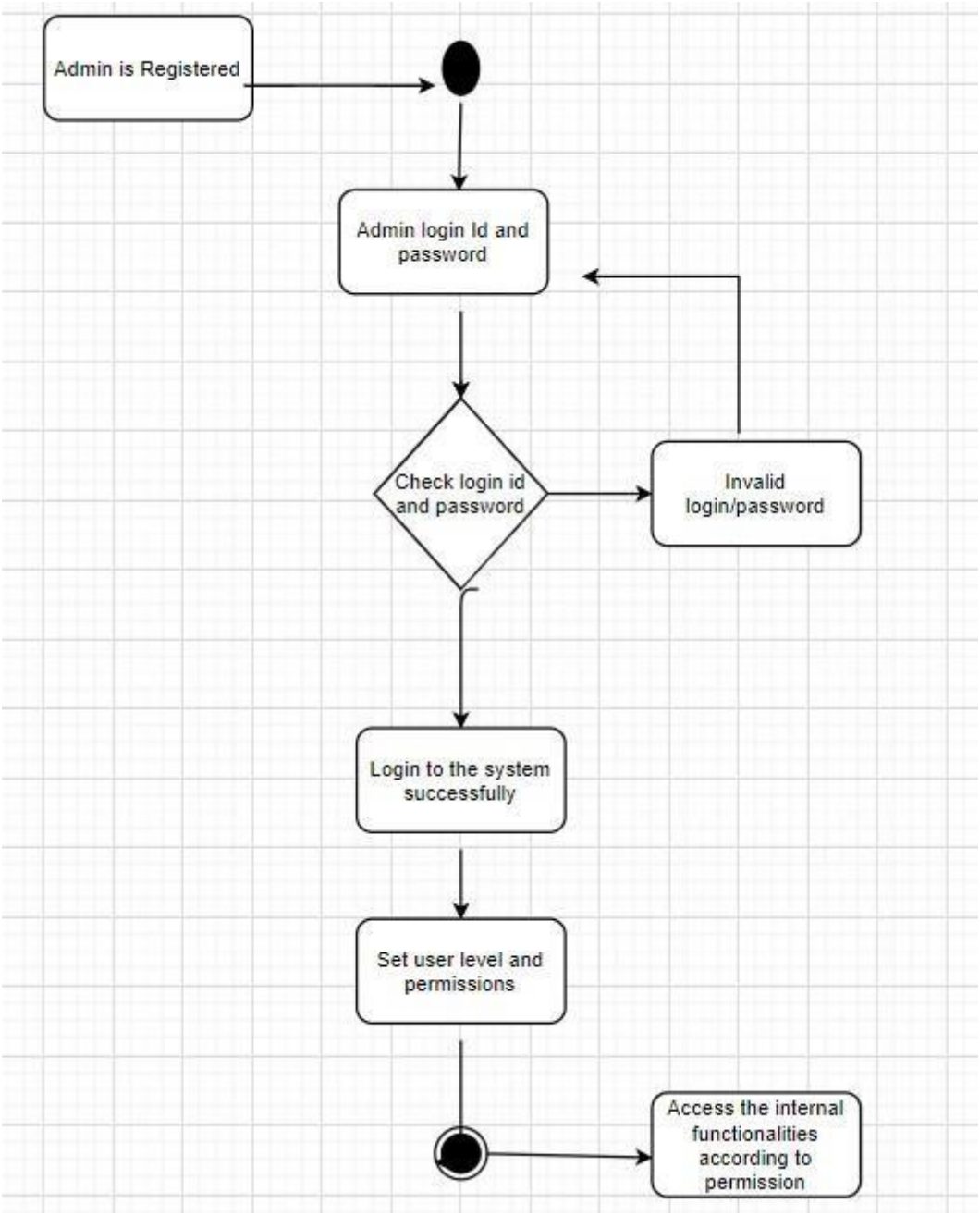
Then, It displayed in the table form and can also edit and delete the leave records.



CLASS-DIAGRAM



SEQUENCE-DIAGRAM



ACTIVITY-DIAGRAM

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 System Architecture

MODEL VIEW CONTROL (MVC)

Model: The central component of the pattern. It is the application's dynamic data structure, independent of the user interface. It directly manages the data, logic and rules of the application.

View: Any representation of information such as a chart, diagram or table. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.

Control: Accepts input and converts it to commands for the model or view.

GOALS OF MVC

- **Simultaneous development:** Work in parallel on different components without impacting or blocking one another
- **Code reuse:** Reuse components quickly and easily in other applications.
- **High cohesion:** MVC enables logical grouping of related actions on a controller together. The views for a specific model are also grouped together.
- **Low coupling:** The very nature of the MVC framework is such that there is low coupling among models, views or controllers
- **Ease of modification:** Because of the separation of responsibilities, future development or modification is easier
- **Multiple views for a model:** Models can have multiple views
- **Code navigability:** The framework navigation can be complex because it introduces new layers of abstraction and requires users to adapt to the decomposition criteria of MVC.
- **Multi-artefact consistency:** Decomposing a feature into three artifacts causes scattering. Thus, requiring developers to maintain the consistency of multiple representations at once.

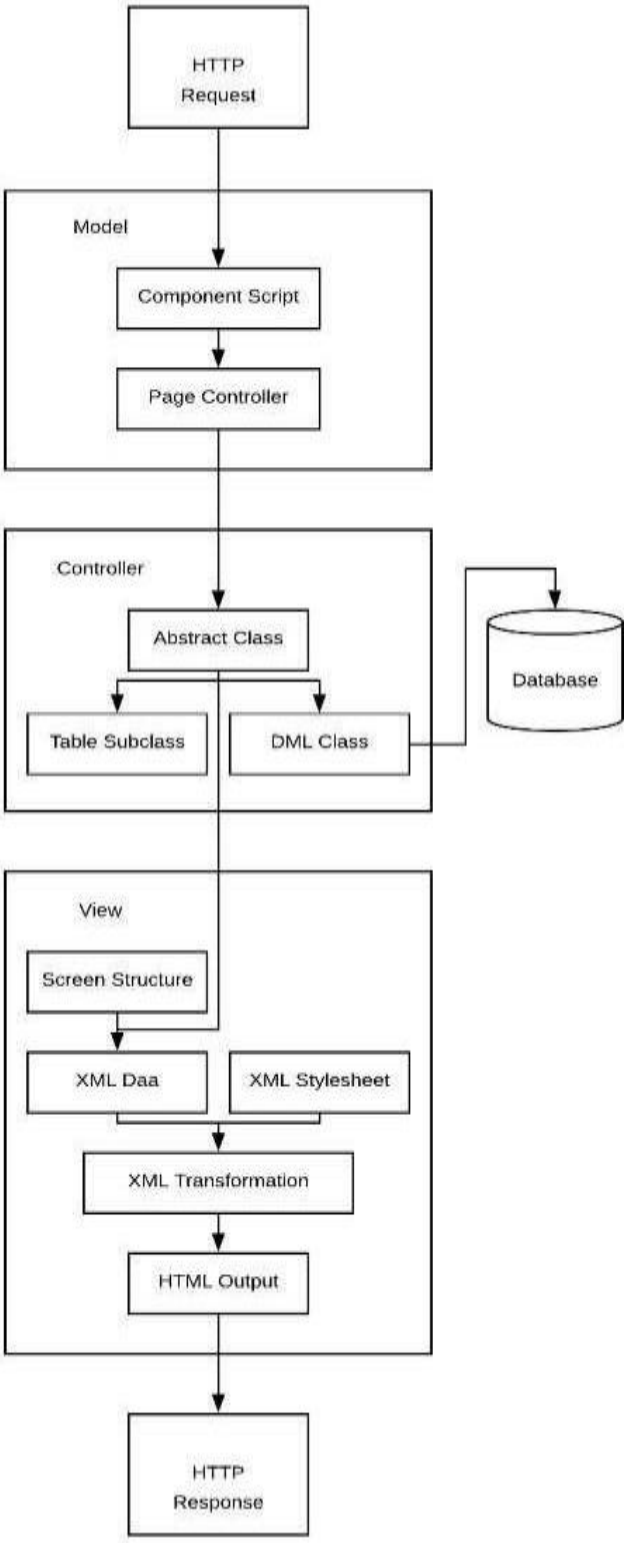


Figure 7.1 Model View Control

The basic diagrammatic representation of the MVC is shown in the above figure 7.1

ACCESS CONTROL LOGIC

Administrators have the privilege of controlling the columns. Columns are nothing but the date, status etc. Usually the access is given by the delivery or the technical team so that it prevents opportunities for a malicious person to use accounts as part of an attack. Privileges are only assigned to an operator with a validated authorization. Access to other system functions is disabled. User and administrator privileges are controlled in a way that allows all privileges to be tailored to individual needs. Accounts are granted with various levels of privileges that are necessary and additional access is given based on the requirements. Access control logic provides the permission to various users at different levels of entry into the application.

CHAPTER 8

RESULTS AND DISCUSSION

8.1 Results and Discussion

Thus the Human Resource Management System of manual work has been completely shifted to the computerized process and this has enabled the company to carry out its operation more quickly. The new system has reduced the paperwork. Accessing and getting data can be done at a single click. Data manipulation has become simpler and the cost factor has been reduced. It is faster and more efficient processing of data. It is less time consuming, Operations are more transparency.

React and Angular offer completely diverse approaches to web application development for startup, small and midmarket businesses. Both technologies are powerful and flexible, while none of them is worse or better, than the other. Depending upon custom app goals and particular system constraints, developers can run from ng2 to React, and back. Opting for Angular, it usually assumes the creation of core skeleton for a front-end app, whereas React.js can be applied to improve its specific parts. Moreover, it can be integrated with other frameworks, like Backbone or even well-known Angular.

8.2 SNAPSHOTS

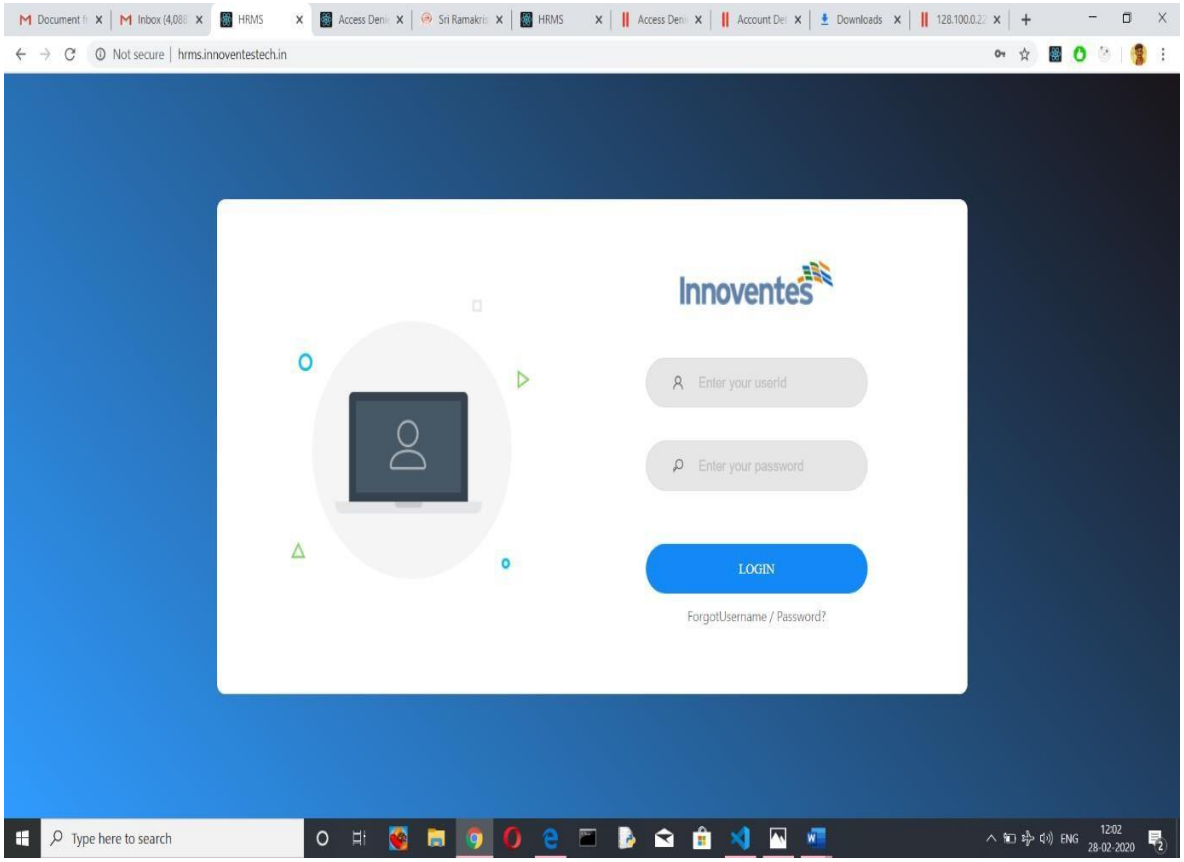


Figure 8.1 Login Page

Login Page where the HR need to login with their user name and password

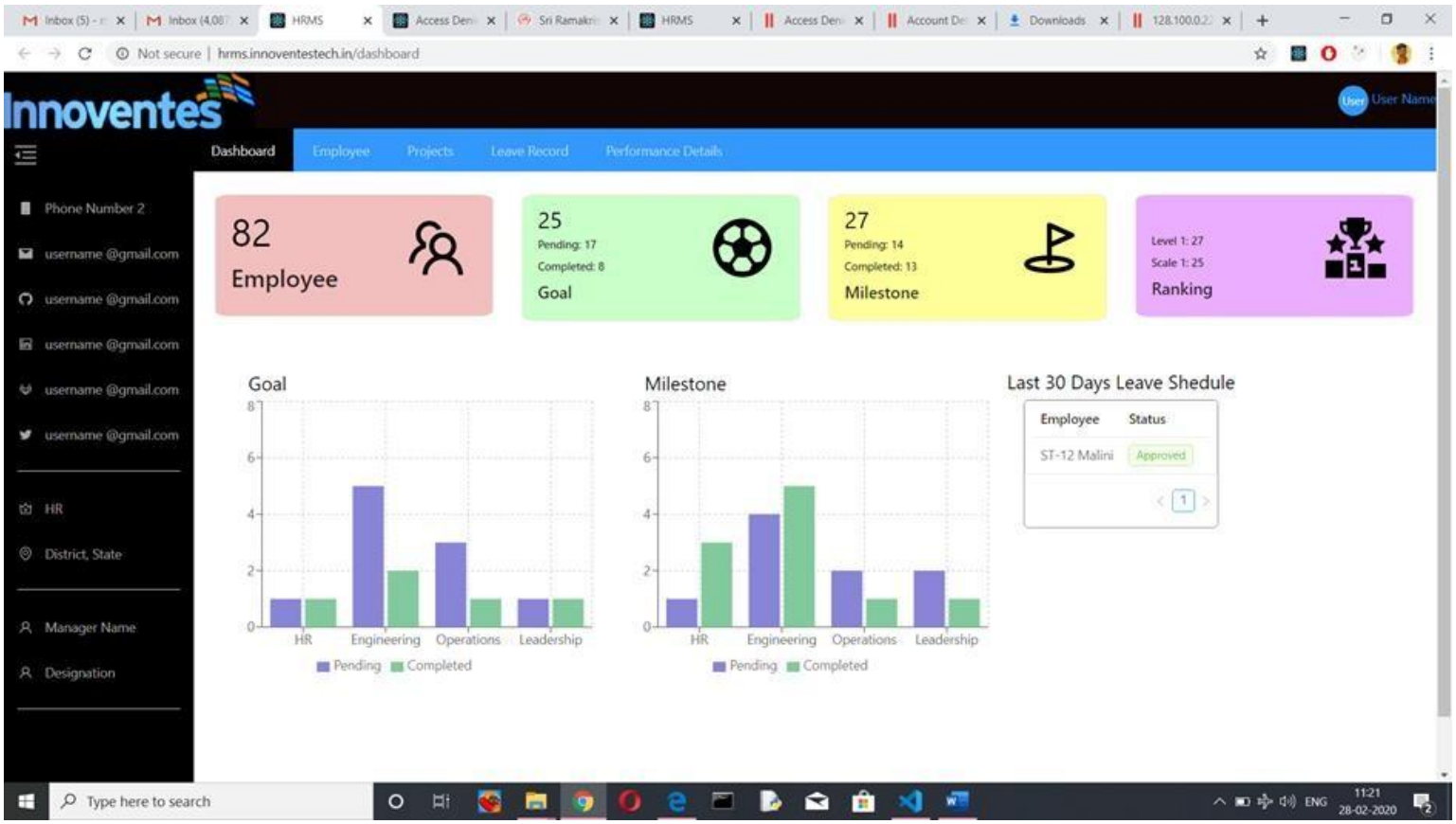


FIGURE 8.2

Where in the dashboard the HR can view brief about the number of Employee and number of work (both pending and completed),milestone and employee ranking

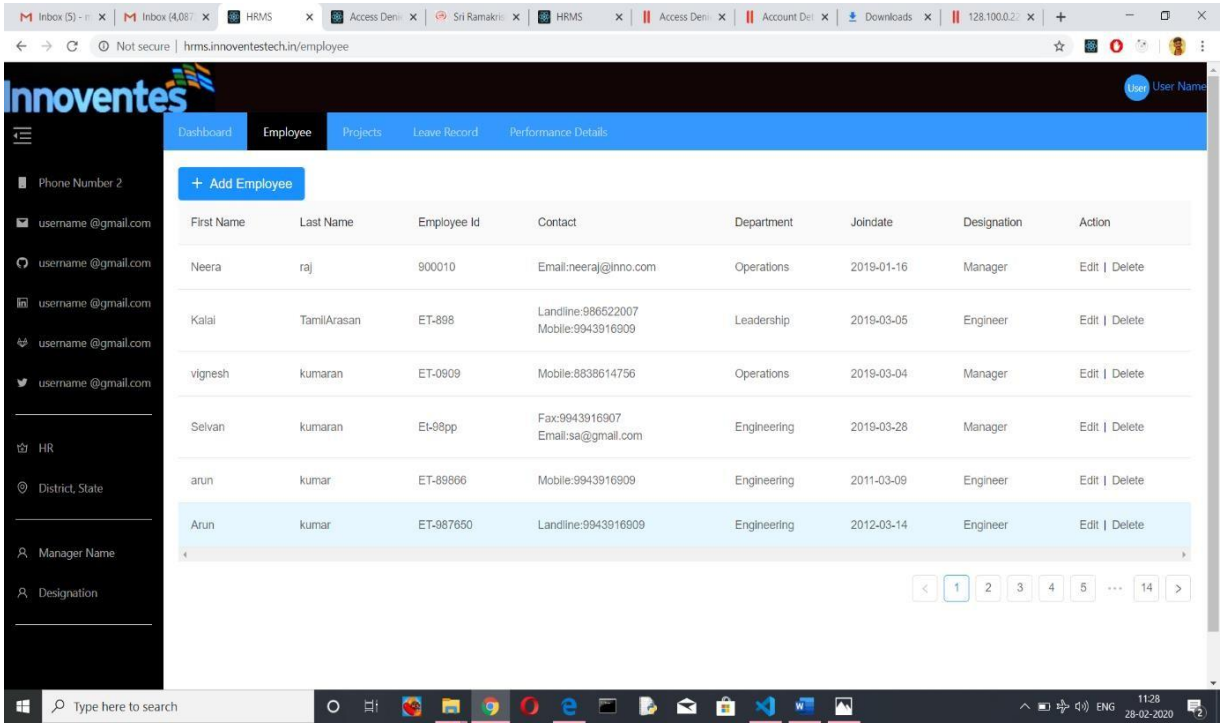


FIGURE 8.3 Employee details

Where in Employee details page HR can View the employee details and here HR can add the new employee details and edit the details or delete the employee

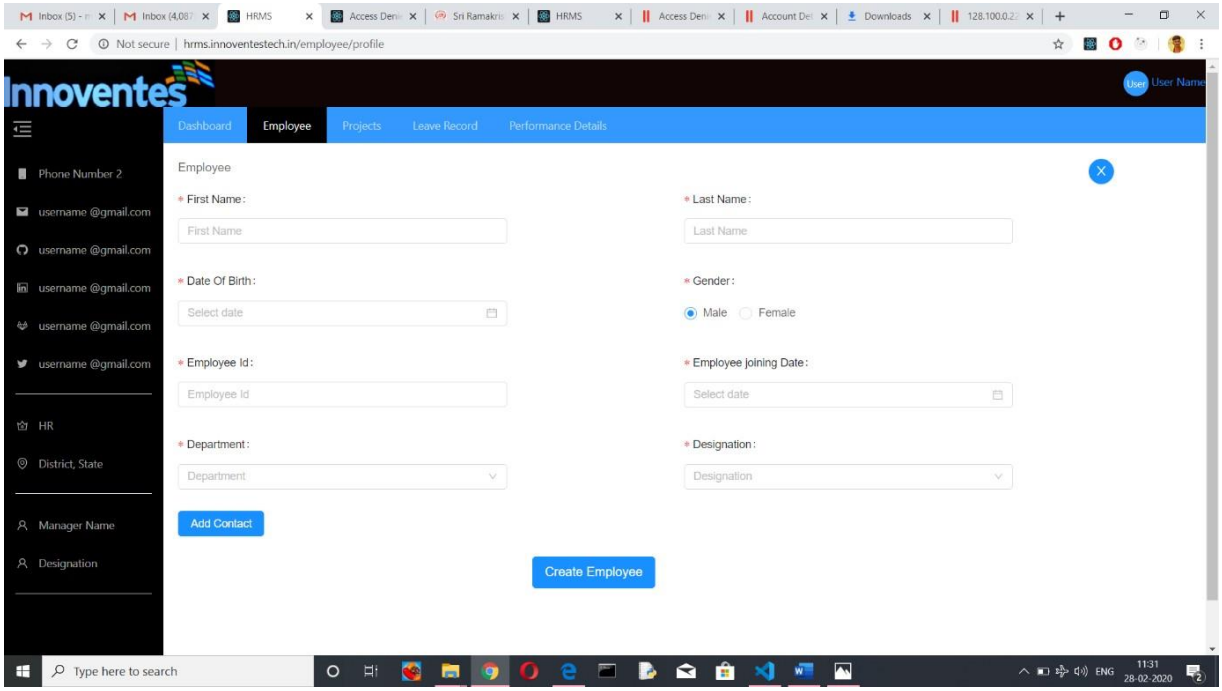


FIGURE 8.4 CREATE EMPLOYEE

Form to create the new employee by filling the details in the input fields

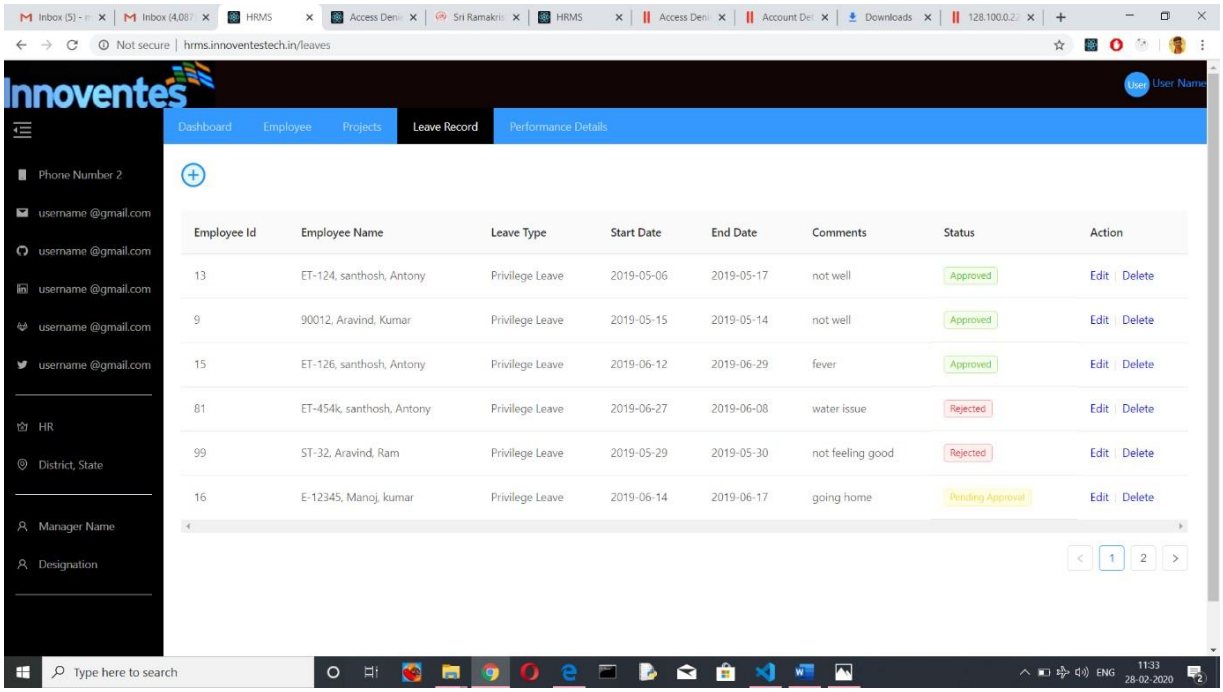


FIGURE 8.5 EMPLOYEE LEAVE RECORD

Leave record of employee consist of number of leave taken and HR can approval or delete the leave for the employee

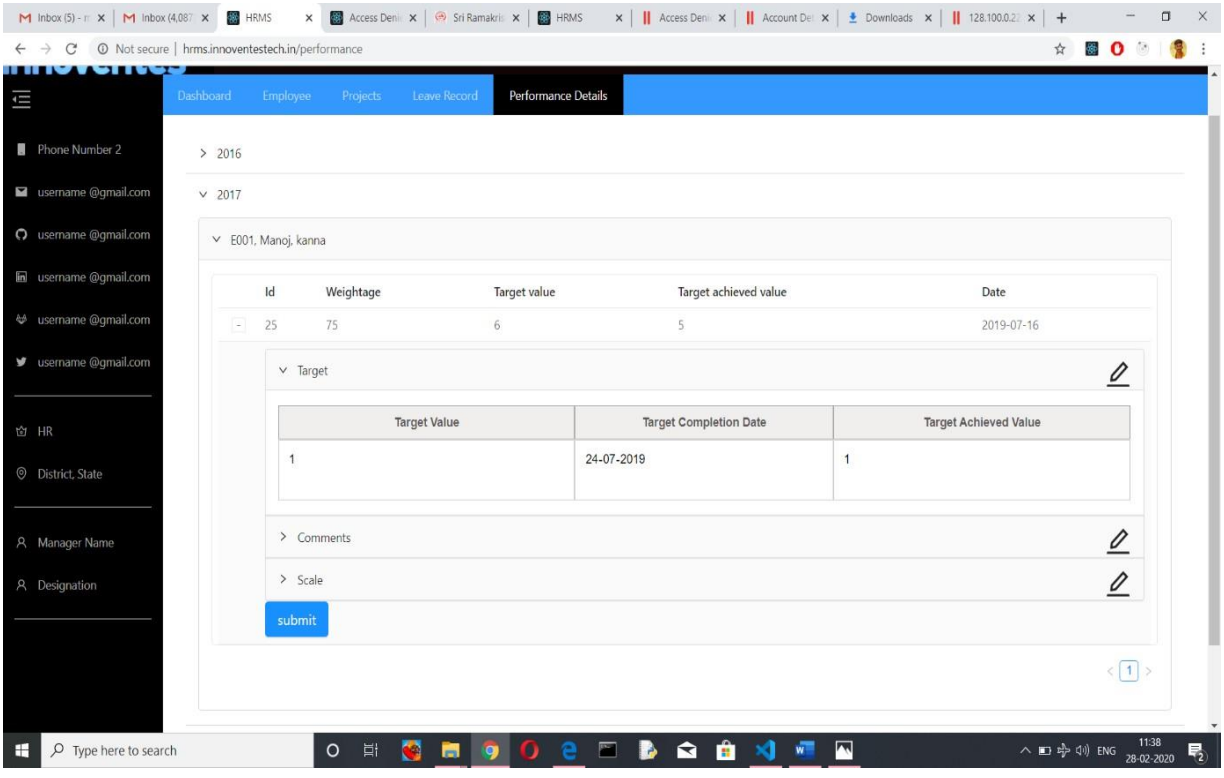


FIGURE 8.6 PERFORMANCE DETAIL

In the performance form can we see the performance for particular employees weightage, target value and target achieved value

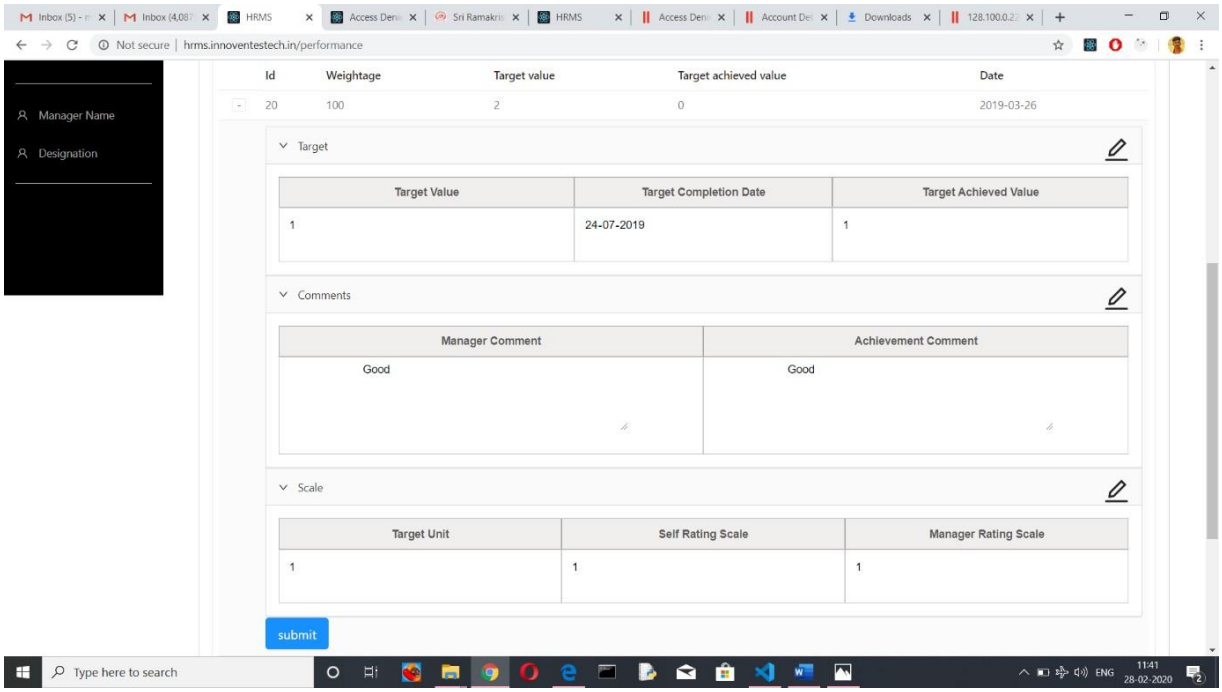


FIGURE 8.7 PERFORMANCE IN DETAIL

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

9.1 Conclusion

Thus the Human Resource Management System application contains the personal details of the employees, their goals and milestones. This can be tracked by the HR manager. It also includes the features like adding new employees details and other project details, editing the stored details and displaying the details in the table form. This system also have leave management module, with that employees can apply the leave request and get the approval from the manager.

9.2 Future Enhancement

This application can be extended to various other modules based on the organization’s need. Apart from employee information, leave management, Analytical dashboard, a login page for Employee and Admin can be created with the same UI but with different access according to priority . The analysis and the productivity of the employee can be monitored and tracked and the manual work of maintaining the records can be reduced dramatically in this method.

APPENDIX

1. Source Code

Module 1 – Employee's Goal Tracking

Dashboard.js

```
import React, { Component } from "react";
import {retrieveEmployeeDetails,retrieveGoalList, retrieveMilestoneList} from
"../API/dashboard";
import { Card, Row, Col, Icon } from "antd";
import "../styles/home.css";
import football from "../image/football.svg";
import milestone from "../image/milestone.svg";
import ranking from "../image/ranking.svg";
import {BarChart, Bar, XAxis, YAxis, CartesianGrid, Tooltip, Legend,
} from 'recharts';
class Home extends Component {
constructor(props) {
super(props);
this.state = {
employeeCount: 0,
goalCount: 0,
goalCompletedCount: 0,
goalPendingCount: 0,
milestoneCount: 0,
milestoneCompletedCount: 0,
milestonePendingCount: 0,
scaleCount: 0,
levelCount: 0,
};
}
componentDidMount() {
retrieveEmployeeDetails().then(Response=>{
this.setState({employeeCount: Response.data.count})
```

```
});
retrieveGoalList().then(Response=>{
this.countGoal(Response.data.results);
this.setState({goalCount: Response.data.count})
})
retrieveMilestoneList().then(Response=>{
this.countMilestone(Response.data.results);
this.setState({milestoneCount: Response.data.results.length})
})
}
countGoal = (goal) => {
let i=0;
var goalLenght = goal.length,completed = 0,pending = 0,scale = 0;
while(i<goalLenght){
if(goal[i].target_achieved_value === goal[i].target_value) {
completed = completed + 1;
} else {
pending = pending + 1;
}
if(goal[i].manager_rating_scale_name === "Scale 1") {
scale = scale + 1;
}
i++;
}
this.setState({
scaleCount:scale,
goalCompletedCount:completed,
goalPendingCount:pending
})
}
countMilestone = goal => {
let i=0;
var goalLenght=goal.length,completed = 0,pending = 0,fields,level = 0;
while(i<goalLenght){
```

```
if(goal[i].achieved_target_value === goal[i].target_value) {
completed=completed + 1;
} else {
pending = pending + 1;
}
fields = goal[i].goal_name.split(',');
if(fields[4] === " Level 1") {
level = level + 1;
}
i++;
}
this.setState({
levelCount:level,
milestoneCompletedCount:completed,
milestonePendingCount:pending
})
}
render() {
const data = [
{
name: 'Human Resource', Completed: 4, Pending: 24,
},
{
name: 'Leader Ship', Completed: 3, Pending: 13,
},
{
name: 'Operation', Completed: 2, Pending: 98,
},
{
name: 'Engineering', Completed: 27, Pending: 39,
},
];
return (
<div className="main-content">
```

```
<Row type="flex" gutter={24} >
<Col xs={22} sm={18} md={12} lg={10} xl={6} >
<Card hoverable className="all-employee">
<Row type="flex">
<Col span={5}>
<label className="employee-count">{this.state.employeeCount}</label>
<h5 className="employee-card-header ">Employee</h5>
</Col>
<Col span={1} offset={12}>
<Icon type="team" className="team" />
</Col>
</Row>
</Card>
</Col>
<Col xs={22} sm={18} md={12} lg={10} xl={6}>
<Card hoverable className="all-goals">
<Row type="flex">
<Col span={12}>
<label className="count">{this.state.goalCount}</label>
<h5 className="card-body">Pending: {this.state.goalPendingCount}</h5>
<h5 className="card-body">Completed: {this.state.goalCompletedCount}</h5>
<h5 className="card-header">Goal</h5>
</Col>
<Col span={1} offset={5}>
<img src={football} alt="football" className="football" />
</Col>
</Row>
</Card>
</Col>
<Col xs={22} sm={18} md={12} lg={10} xl={6} >
<Card hoverable className="all-milestone">
<Row type="flex">
<Col span={11}>
<label className="count">{this.state.milestoneCount}</label>
```

```
<h5 className="card-body">Pending: {this.state.milestonePendingCount}</h5>
<h5 className="card-body">Completed: {this.state.milestoneCompletedCount}</h5>
<h5 className="card-header">Milestone</h5>
</Col>
<Col span={1} offset={6}>
<img src={milestone} alt="Milestone" className="football" />
</Col>
</Row>
</Card>
</Col>
<Col xs={22} sm={18} md={12} lg={10} xl={6}>
<Card hoverable className="all-rank">
<Row type="flex">
<Col span={11} className="rank-card">
<h5 className="card-body">Level 1: {this.state.levelCount}</h5>
<h5 className="card-body">Scale 1: {this.state.scaleCount}</h5>
<h5 className="card-header">Ranking</h5>
</Col>
<Col span={1} offset={6}>
<img src={ranking} alt="Rank" className="rank" />
</Col>
</Row>
</Card>
</Col>
<Row gutter={10}>
<Col xs={23} sm={16} md={12} lg={24} xl={6}>
<Card title="Goal" bordered={false} className="goal-chart">
<BarChart data={data} height={300} width={420}>
<CartesianGrid strokeDasharray="3 3" />
<XAxis dataKey="name" />
<YAxis />
<Tooltip />
 <Legend />
 <Bar dataKey="Pending" fill="#8884d8" />
```

```
<Bar dataKey="Completed" fill="#82ca9d" />
</BarChart>
</Card>
</Col>
<Col xs={24} sm={24} md={12} lg={24} xl={6} >
<Card title="Milestone" bordered={false}>
<BarChart data={data} height={300} width={420}>
<CartesianGrid strokeDasharray="3 3" />
<XAxis dataKey="name" />
<YAxis />
<Tooltip />
<Legend />
<Bar dataKey="Pending" fill="#8884d8" />
<Bar dataKey="Completed" fill="#82ca9d" />
</BarChart>
</Card>
</Col>
</Row>
</Row>
</div>);
}}
export default Home;
```

Module 2 – Employee Details

employeeDetails.js

```
import { Button, Col, Divider, Icon, Row, Table, message, Popconfirm,Spin } from 'antd';
import React, { Component } from 'react';
import '../styles/employee.css';
import {retrieveEmployeeDetails,changeDeleteStatus, retrieveParticularEmployeeDetails}
from '../API/employeeApi';

var key=1;
class EmployeeDetails extends Component {
constructor (props) {
```

```
super(props)
this.state = {
data: [],
temp:        [],
loading: false
}
}

handleAddEvent = () => {
this.props.history.push(`${this.props.location.pathname}/profile`);
}
handleEdit = (row) => {
retrieveParticularEmployeeDetails(row.id).then(response=>{
this.props.history.push(`${this.props.location.pathname}/edit`,{data:response.data})
})

}
handleRowDel = (row, details) => {
var index =details.indexOf(row);
retrieveParticularEmployeeDetails(row.id).then(values =>{
values.data.deleted_status = true;
changeDeleteStatus(values.data,row.id).then(data =>{
details.splice(index,1);
this.setState({data:details});
}).catch(error => {
message.error("Unable to change Employee Delete Status");
});
}).catch(error => {
message.error("Unable to retrieve Employee Details");
});
}
componentWillMount() {
this.getTableData();
}
```

```
getTableData = () => {
retrieveEmployeeDetails().then(response =>{
this.setState({ data: response.data.results,
loading:true})
}).catch(error => {
message.error("Unable to Retrieve Employee Details");
});
}
render () {
const columns = [
{
title: "Name",
dataIndex: "first_name",
key: "first_name",
},
{
title: "Employee Id",
dataIndex: "employee_id",
key: "employee_id",
},
{
title: "Contact",
dataIndex: "contact_list",
key: "id",
render:(text)=>(text.map(item=>(<div key
={key++}>{item.contact_type_name}:{item.contact_value}</div>)))
},
{
title: "Department",
dataIndex: "employment_details_list",
key: "employment_details_list",
render:(text,row)=>(text.map((item)=><div key =
{row.id}>{item.department_name}</div>))
},
```

```
{
title: "Joindate",
dataIndex: "employment_details_list",
key: 'employement_start_date',
render:(text,row)=>(text.map((item)=><div key =
{row.id}>{item.employment_start_date}</div>))

},
{
title: "Designation",
dataIndex: "employment_details_list",
key: "designation",
render:(text,row)=>(text.map((item)=><div key =
{row.id}>{item.designation_name}</div>))
},
{
title: "Action",
dataIndex: "action",
key: "action",
render: (text,row) => (

<span>
<label onClick = {this.handleEdit.bind(this,row)}>Edit</label>
<Divider className="divider" type="vertical" />
<Popconfirm title="Sure to delete?"
onConfirm={this.handleRowDel.bind(this,row,this.state.data)}><label>Delete</label>
</Popconfirm>
</span>
)
}
]
return (
<div className="container">
{ this.state.loading === false ? <Row type="flex" align = "middle" justify="center">
```

```
<Spin tip= "Loading"/>
</Row> :
<div>
<Row>
<Col span={2}>
<Button type="primary" size={"large"} onClick={this.handleAddEvent}>
<Icon type="plus" /> Add Employee
</Button>
</Col>
</Row>
<Row>
<Col >
<Table columns={columns}
dataSource={this.state.data}
rowKey = "id"
/>
</Col>
</Row>
</div>
}
</div>
)
}
}
export default EmployeeDetails
```

Module 3 – Leave Record Management

leaveRecord.js

```
import React, { Component } from "react";
import { Tag, Table, Icon, Popconfirm, message, Divider } from "antd";
import { getLeaveRecords, deleteLeaveRequest } from "../API/LeavesApi";
import "../styles/LeaveRecord.css";
class LeaveRecord extends Component {
constructor(props) {
```

```
super(props)
;  this.state  =
{ data: []
};
}
addLeave = () => {
this.props.history.push(`${this.props.location.pathname}/add`);
};
componentWillMount() {
getLeaveRecords().then(leaveList => {
this.setState({ data: leaveList.data.results });
});
}
handleDelete = text => {
let row = this.state.data.indexOf(text);
let leaveData = [...this.state.data];
leaveData[row].deleted_status = "true";
deleteLeaveRequest(leaveData[row].id, leaveData[row]).then(Response => {
if (Response.status === 200) {
leaveData.splice(row, 1);
this.setState({ data: leaveData });
} else {
message.error("Unable to delete");
}
});
};
handleEdit = text => {
this.props.history.push(`${this.props.location.pathname}/add`,{data : text});
};
render() {
const columns = [
{
title: "Employee Id",
dataIndex: "employee",
```

```
key: "employee"
},
{
title: "Employee Name",
dataIndex: "employee_name",
key: "employee_name"
},
{
title: "Leave Type",
dataIndex: "leave_type_name",
key: "leave_type_name"
},
{
title: "Start Date",
dataIndex: "leave_start_date",
key: "leave_start_date"
},
{
title: "End Date",
dataIndex: "leave_end_date",
key: "leave_end_date"
},
{
title: "Comments",
dataIndex: "comments",
key: "comments"
},
{
title: "Status",
key: "request_status_name",
dataIndex: "request_status_name",
render: request_status_name => {
var color;
if (request_status_name === "Pending Approval") color = "yellow";
```

```
else if (request_status_name === "Approved") color = "green";
else if (request_status_name === "Rejected") color = "red";
return <Tag color={color}>{request_status_name}</Tag>;
}
},
{
title: "Action",
key: "action",
className: "action",
render: (text, record) => (
<Popconfirm
title="Sure to delete"
onConfirm={() => this.handleDelete(text)}
>
<label
className="delete-lable"
onClick={() => this.handleEdit(record)}
>
Edit
</label>
<Divider type="vertical"/>
<label className="delete-lable">Delete</label>
</Popconfirm>
)
},

];
return (
<div className="main-content">
<Icon
type="plus-circle"
theme="twoTone"
className="add_leave_icon"
onClick={this.addLeave}
```

```
/>
<Table
rowKey="id"
columns={columns}
dataSource={this.state.data}
className="leave_record_table"
/>
</div>
);
}
}
export default LeaveRecord;
```

CHAPTER 10

REFERENCES

[1] Xiaobing Zhang , Xiaobing Zhang, "The Effects on Human Resource Management System and Organizational Performance: A Role of Absorption", International Conference on Wireless Communications, Networking and Mobile Computing, Volume: 03 Issue: 06, pp 112-125, 2008.

[2] Suraj Francis Noronha, Paul G Aquinas, Aruna Doreen Manezes, "Implementing Employee Management System", IEEE FORUM International Conference, , Volume: 05 Issue: 08, pp 156-170, 2016.

[3] Vivin.S, Swathy.S, Sruthi.C, "Employee Management System", International Research Journal of Engineering and Technology (IRJET), Volume: 03 Issue: 08, pp 234-247, 2016.

[4] Anand Kotasthane, Akshay Bahade, Kaustubh Chinchalkar, Prasad Gawande and Ms. P K Karmor ,"Intranet Automation of Human Resource Management System", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 10 Issue: 05, pp 432-457, 2013.

[5] Prashant Dhotre, anesh Rasal, Mamta Jha, Nihal Pawar, " Employee Monitoring and Management System using GPS and Android ", International Research Journal of Engineering and Technology(IRJET), Volume: 04 Issue: 03, pp 637-642, 2017.

[6] Bandar khalaf Alharthey, Amran Rasli, "The use of Human Resource Management Systems in the Saudi market", Asian Journal of Business Ethics, Volume: 01 Issue: 02, pp 163-176, 2012.

[7] Xing Xu, Hao Hu, Na Hu, Lin Xiao, Weiqin Ying "Human Resource Management System Based on Factory Method Design Pattern", International Conference on Information Computing and Applications, Volume: 10 Issue: 03, pp 437-444, 2015.