CERTIFICATE

This is to certify that the project entitled

"Leave Management System"

Has been satisfactorily completed by

1) Anmol Chaudhary
2) Anubha Singh
3) Gangasagar Piraji Pikle
4) Utkarsh Manikrao Garad
5) Vaibhav Bharat Imade

Under supervision and guidance for partial fulfilment of the
PG Diploma in Advanced Computing (DAC) (24 weeks FullTime) Course

of

Centre for Development of Advanced Computing (C-DAC), Pune.

at

Academy of Information Technology

(YCP) Nariman Point, Mumbai –400 021.

Faculty **Course Co-ordinator**

Batch: PG DAC March 2024

BatchDate: 19th Aug 2024.

PREFACE

In today's fast-paced corporate environment, efficient management of employee leave is crucial for maintaining organizational productivity and employee satisfaction. The Leave Management System is a meticulously designed web-based platform that addresses the complexities of managing employee leaves with precision and ease.

This system empowers employees by providing a straightforward process to log in, select the type of leave they wish to apply for, specify the duration, and submit a reason for their request. A comprehensive employee profile page securely stores all necessary personal and professional information, including name, employee ID, contact details, date of birth, address, and company email, ensuring that the process is both personalized and accurate.

For administrators, the platform offers a robust and intuitive dashboard that delivers a clear, real-time overview of the organization's leave-related data. This includes detailed metrics on the various types of leave available, the number of active employees, department statistics, and the status of pending, declined, and approved leave applications. The dashboard's "recently" section is particularly useful, enabling administrators to quickly access and review the most recent leave approvals or declines.

Moreover, the system's functionality extends beyond basic leave management. It includes a department management section that allows administrators to easily add new departments or remove existing ones as organizational needs evolve. The leave type management section offers similar flexibility, enabling the addition or deletion of leave categories as required.

The employee management section provides a comprehensive view of all leave requests, categorized by status—pending, approved, or total—along with options to add new employees or review detailed employee profiles, including their leave request histories. Lastly, the leave management section ensures that all leave operations are handled efficiently and transparently, promoting smooth workflows and clear communication within the organization.

In essence, the Leave Management System is not just a tool for processing leave requests; it is an integral part of organizational management that enhances efficiency, transparency, and employee satisfaction across the board.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to everyone who has played a role in the development and success of the Leave Management System. Our dedicated team of developers, designers, and HR experts has invested countless hours of effort, creativity, and expertise into creating a platform that redefines the way organizations manage employee leaves. Their unwavering commitment to excellence has been the driving force behind bringing this project to fruition.

We are immensely thankful to our users for their invaluable feedback and contributions, which have been crucial in shaping and enhancing the Leave Management System. Their active engagement and input have inspired us to continuously innovate and refine the platform to better meet their needs.

We also acknowledge the support and collaboration of our partners and stakeholders. Their guidance, resources, and expertise have been instrumental in the development and promotion of the Leave Management System. Together, we have worked diligently to create a platform that not only streamlines the leave management process but also fosters a more efficient and transparent workplace environment.

As we look ahead, we are excited about the future possibilities for the Leave Management System. We remain committed to our mission of revolutionizing leave management and providing organizations with a more connected and proactive approach to handling employee absences. We thank everyone who has been a part of this incredible journey. Your dedication and support have been key in making the Leave Management System a reality, and we are deeply appreciative of your contributions.

Lastly, we express our deepest gratitude to our families and friends for their unwavering support, patience, and understanding throughout the course of this project. Your love, encouragement, and belief in us have been a constant source of strength and inspiration.

To everyone who has played a role, big or small, in making this project a reality – thank you from the bottom of our hearts.

Team Members;

- Anmol Chaudhary
- Anubha Singh
- Gangasagar Piraji Pikle
- Utkarsh Manikrao Garad
- Vaibhav Bharat Imade

INDEX

INTRODUCTION

In an era where efficient organizational management is more critical than ever, employee satisfaction and operational transparency are key to a productive workplace. The Leave Management System emerges as a cutting-edge solution, transforming the traditional methods of managing employee leave into a streamlined, digital experience. Imagine a platform that not only simplifies the process of applying for leave but also ensures that it's handled with precision and transparency – that's what the Leave Management System is all about.

The Leave Management System is more than just a tool for processing leave requests; it's an essential part of an organization's operational backbone. This platform offers a comprehensive suite of features designed to make the leave management process seamless for both employees and administrators. From detailed employee profiles to accurate leave tracking and efficient request management, the system ensures that every aspect of leave administration is handled with clarity and ease.

One of the standout features of the Leave Management System is its robust admin dashboard, which provides a holistic view of the organization's leave data. Administrators can easily manage departments, leave types, and employee records, creating a dynamic and responsive environment where decisions are informed by real-time data. The inclusion of a "recently" section allows admins to stay updated on the latest leave actions, adding an extra layer of oversight and control.

The Leave Management System transcends the limitations of traditional leave processes, turning them into proactive, tech-driven experiences that benefit the entire organization. It empowers employees with an easy-to-use interface for managing their leave while providing administrators with the tools they need to maintain organizational efficiency and transparency.

Join us on this journey as we redefine the future of leave management, creating a more connected, transparent, and efficient workplace. Welcome to the Leave Management System, where we are shaping the future of organizational management, one leave request at a time.

Software Requirements :

- Eclipse IDE
- MySQL
- Visual Studio Code
- Browser

Hardware Requirements :

- PC / Laptop
- I3 Processor minimum
- 4 GB Ram

Tools Requirements :

- Postman
- Git
- GitHub
- Jira

ER Diagram

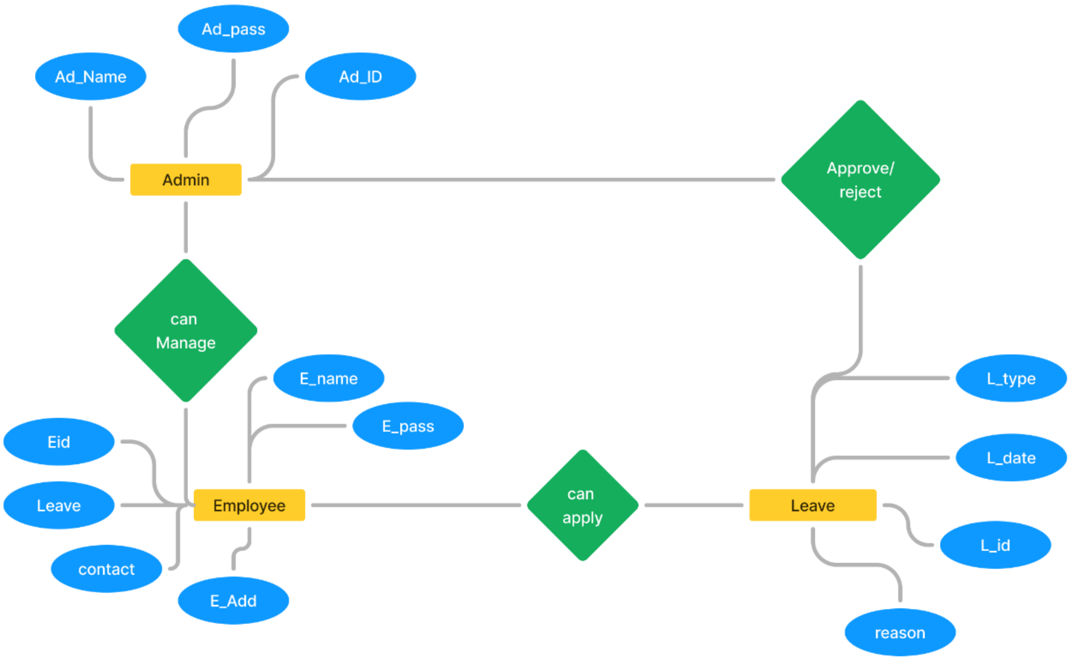


Diagram 1- ER Diagram

Data Flow Diagram

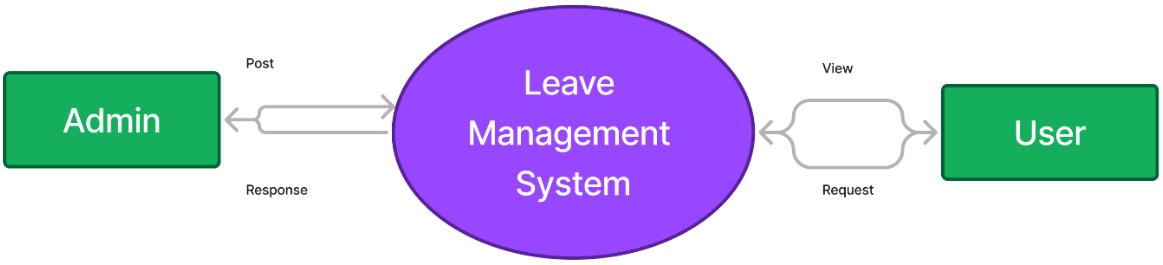


Diagram 2- Data Flow Diagram

Architecture Diagram

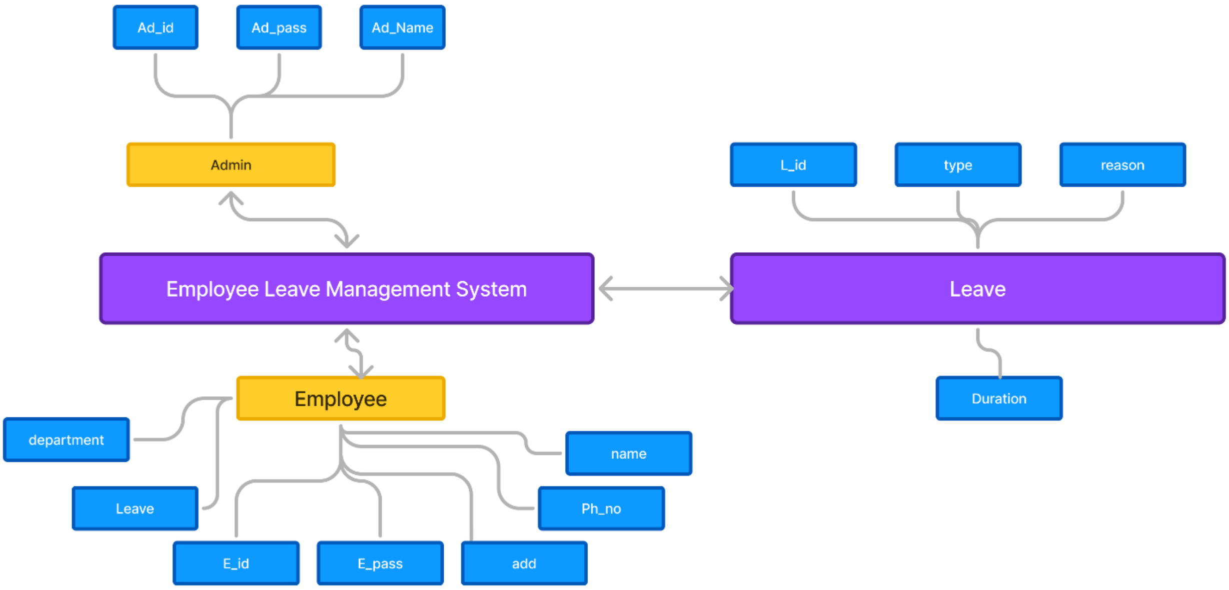


Diagram 3-Architecture Diagram

BACKEND DETAILS

What technology used?

- Spring Boot (Framework)
- MySQL (Database)
- REST API

Dependencies:

- Spring web
- Spring-data-Jpa
- Spring Lombok
- MySQL-connector
- Spring-security-core

Api End Points:

- Admin Login Functionality
- Employee Login Functionality
- Leave Request Functionality

FRONTEND DETAILS

React.js, commonly referred to as React, is an open-source JavaScript library for building user interfaces (UIs) or user interface components. It was developed and is maintained by Facebook and a community of individual developers and companies. React is often used for building single-page applications and mobile applications.

What libraries are used in React?

- React-router-Dom
- Axios

Admin Login:

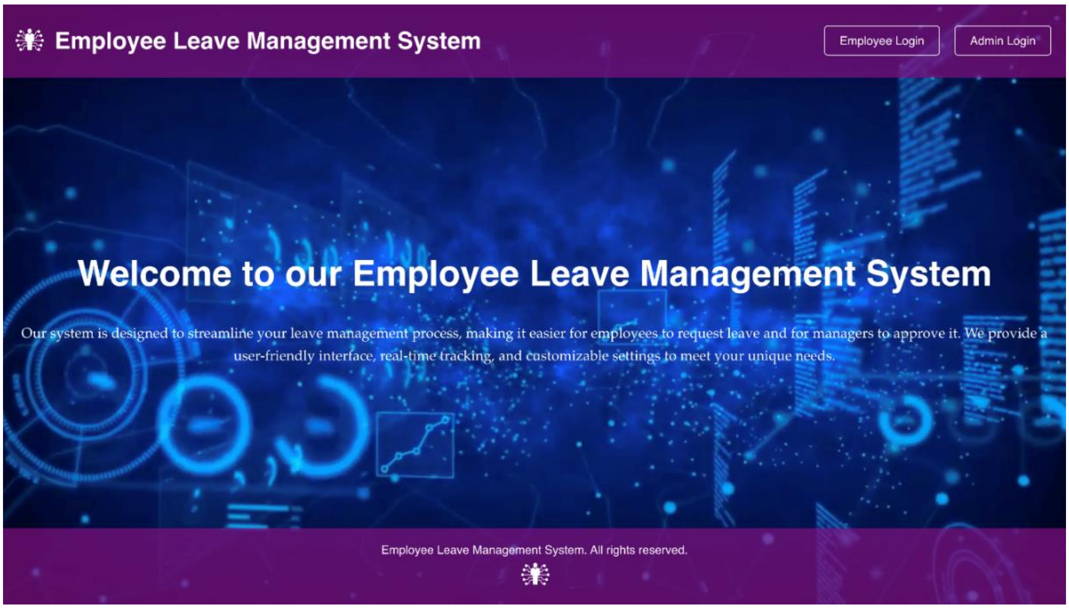
The Admin Login React component is a crucial part of the authentication system for administrators in the web application. It handles login attempts, Super Password, and navigation.

Employee Login:

The Employee Login React component is a crucial part of the authentication system for administrators in the web application. It handles login attempts, Username and password.

Interface

Home Page :



Employee Leave Management System

Welcome to our Employee Leave Management System

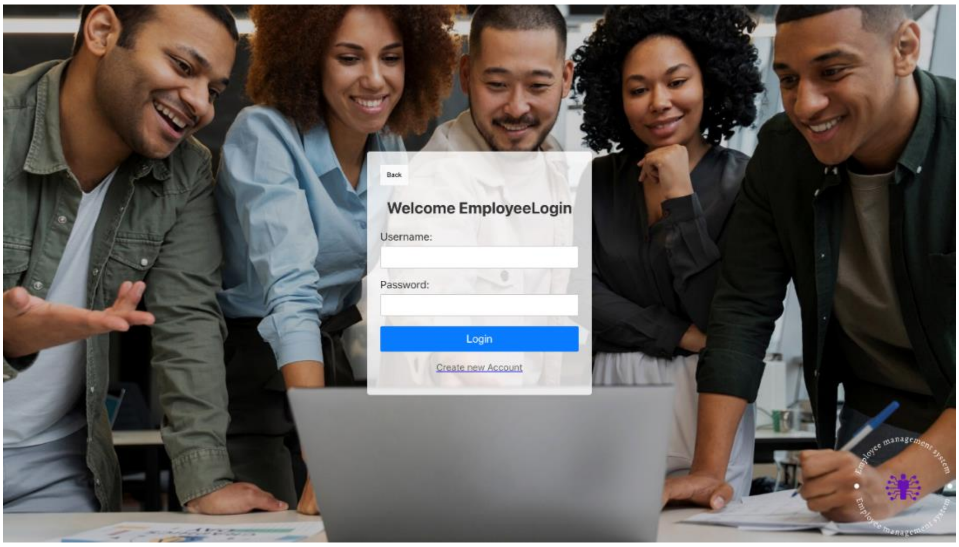
Our system is designed to streamline your leave management process, making it easier for employees to request leave and for managers to approve it. We provide a user-friendly interface, real-time tracking, and customizable settings to meet your unique needs.

Employee Leave Management System. All rights reserved.

Screenshot-1

This is our Home Page you can see there is option for selection Employee Login and Admin Login. When you click on given navbar you can redirect to the Particular page.

Employee Login Page :



Welcome EmployeeLogin

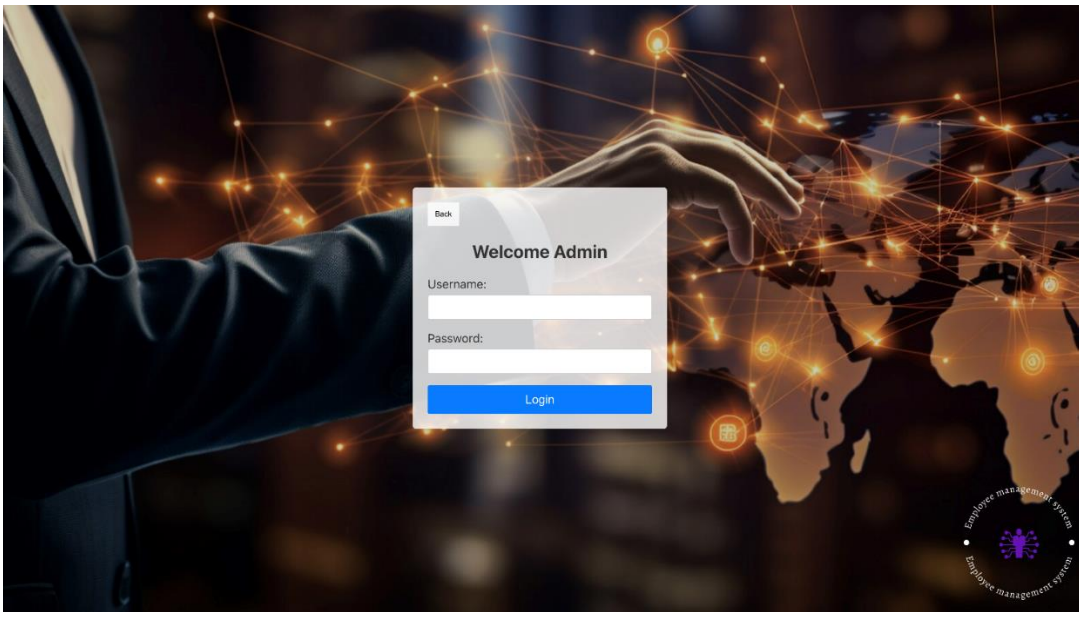
Username:

Password:

Screenshot-2

This is an employee login page where Employee can enter their username and password to access the system.

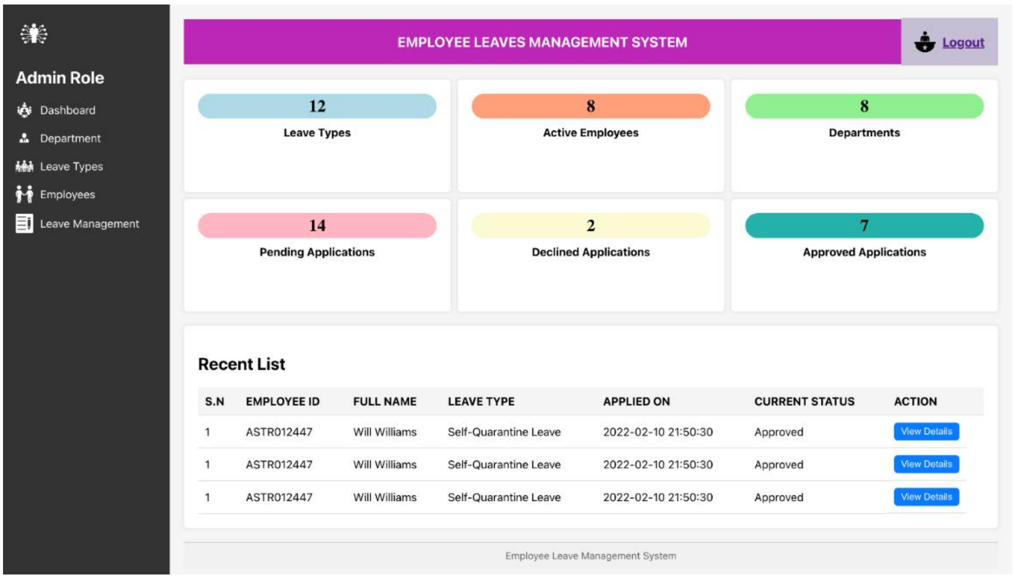
Admin Login Page :



Screenshot-3

This is an Admin login page where Admin can enter their username and password to access the system.

Admin Dashboard Page :



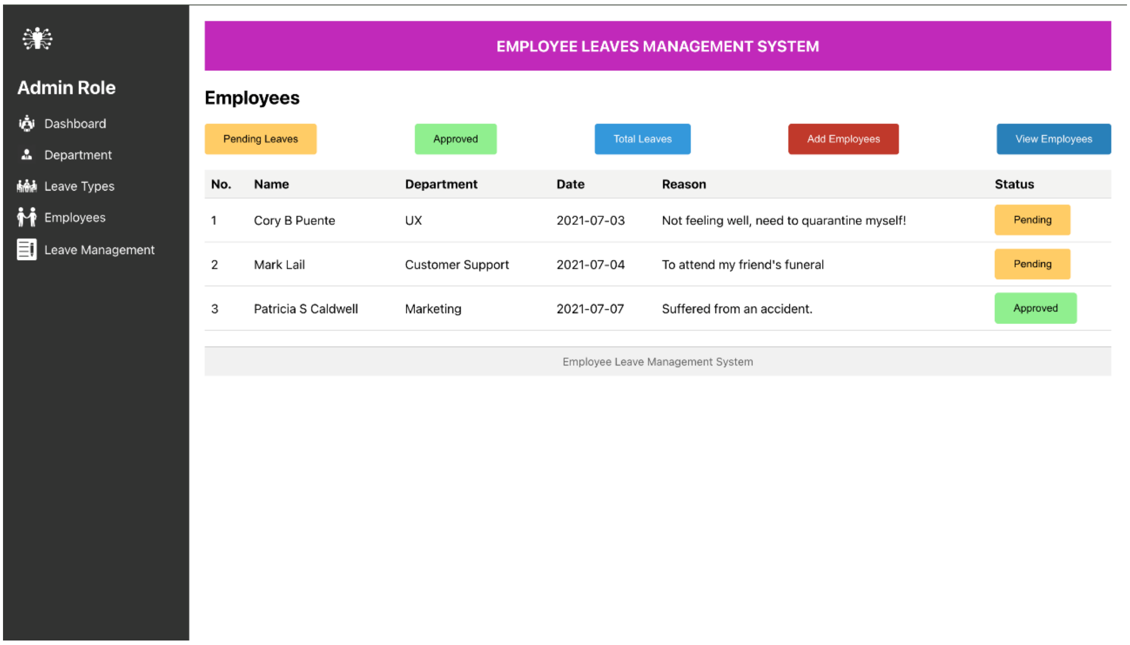
Recent List

S.N	EMPLOYEE ID	FULL NAME	LEAVE TYPE	APPLIED ON	CURRENT STATUS	ACTION
1	ASTR012447	Will Williams	Self-Quarantine Leave	2022-02-10 21:50:30	Approved	View Details
1	ASTR012447	Will Williams	Self-Quarantine Leave	2022-02-10 21:50:30	Approved	View Details
1	ASTR012447	Will Williams	Self-Quarantine Leave	2022-02-10 21:50:30	Approved	View Details

Screenshot-4

Admin dashboard for managing employee leave requests, with sections for monitoring and decision-making across the organization.

Employees Page :



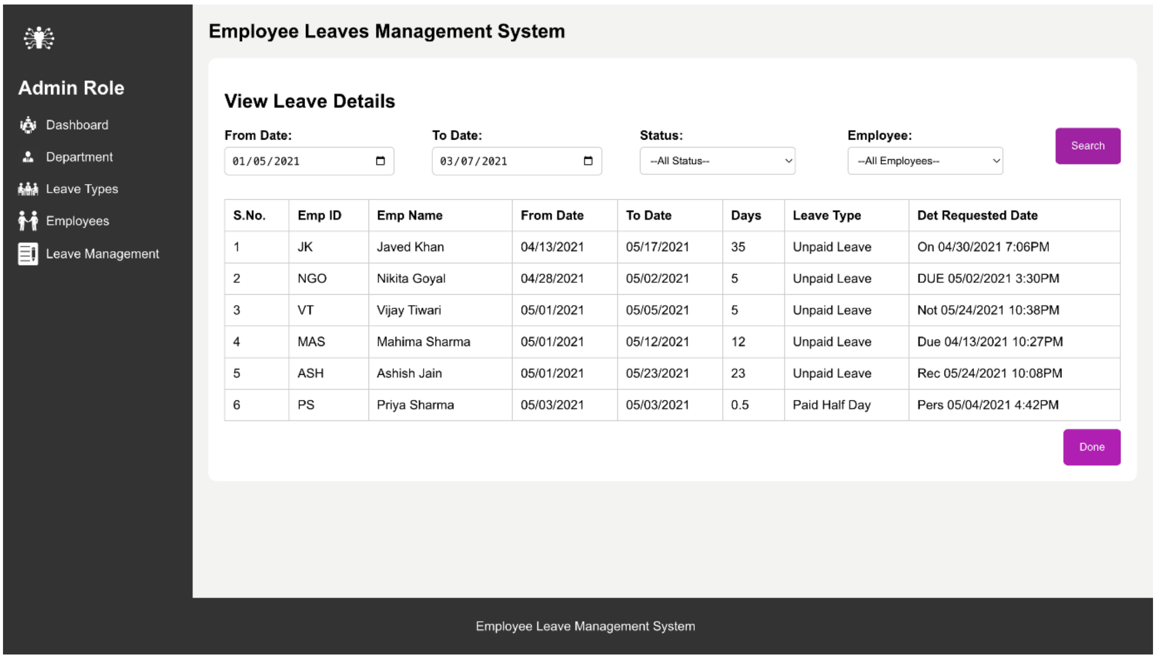
Employees

No.	Name	Department	Date	Reason	Status
1	Cory B Puente	UX	2021-07-03	Not feeling well, need to quarantine myself!	Pending
2	Mark Lail	Customer Support	2021-07-04	To attend my friend's funeral	Pending
3	Patricia S Caldwell	Marketing	2021-07-07	Suffered from an accident.	Approved

Employee Leave Management System

Screenshot-5

View Leave Details Page :



Employee Leaves Management System

View Leave Details

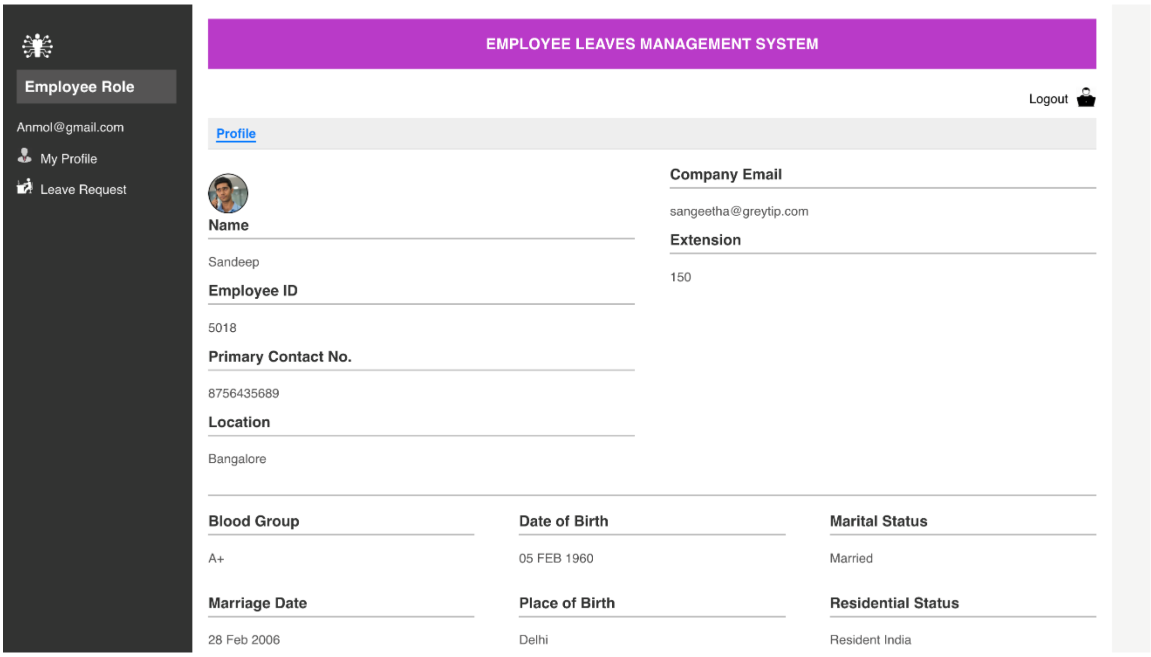
From Date: 01/05/2021
To Date: 03/07/2021
Status: --All Status--
Employee: --All Employees--

S.No.	Emp ID	Emp Name	From Date	To Date	Days	Leave Type	Det Requested Date
1	JK	Javed Khan	04/13/2021	05/17/2021	35	Unpaid Leave	On 04/30/2021 7:06PM
2	NGO	Nikita Goyal	04/28/2021	05/02/2021	5	Unpaid Leave	DUE 05/02/2021 3:30PM
3	VT	Vijay Tiwari	05/01/2021	05/05/2021	5	Unpaid Leave	Not 05/24/2021 10:38PM
4	MAS	Mahima Sharma	05/01/2021	05/12/2021	12	Unpaid Leave	Due 04/13/2021 10:27PM
5	ASH	Ashish Jain	05/01/2021	05/23/2021	23	Unpaid Leave	Rec 05/24/2021 10:08PM
6	PS	Priya Sharma	05/03/2021	05/03/2021	0.5	Paid Half Day	Pers 05/04/2021 4:42PM

Employee Leave Management System

Screenshot-6

Employee Profile Page :



Name

Sandeep

Employee ID

5018

Primary Contact No.

8756435689

Location

Bangalore

Company Email

sangeetha@greytip.com

Extension

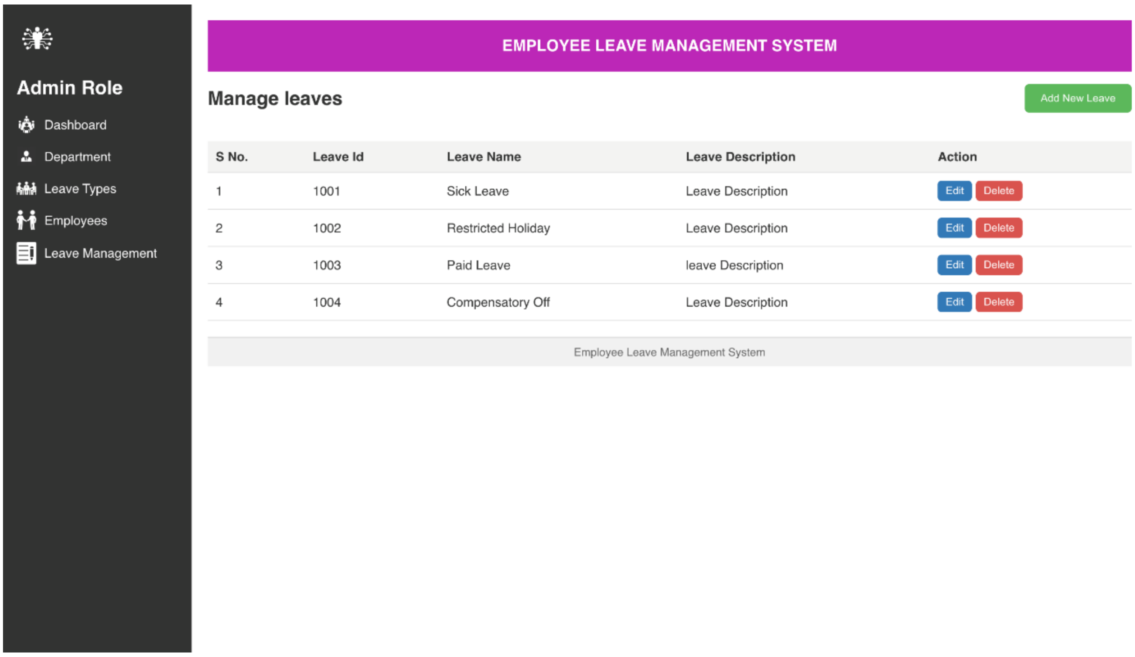
150

Blood Group	Date of Birth	Marital Status
A+	05 FEB 1960	Married

Marriage Date	Place of Birth	Residential Status
28 Feb 2006	Delhi	Resident India

Screenshot-7

Manage Leaves Page :



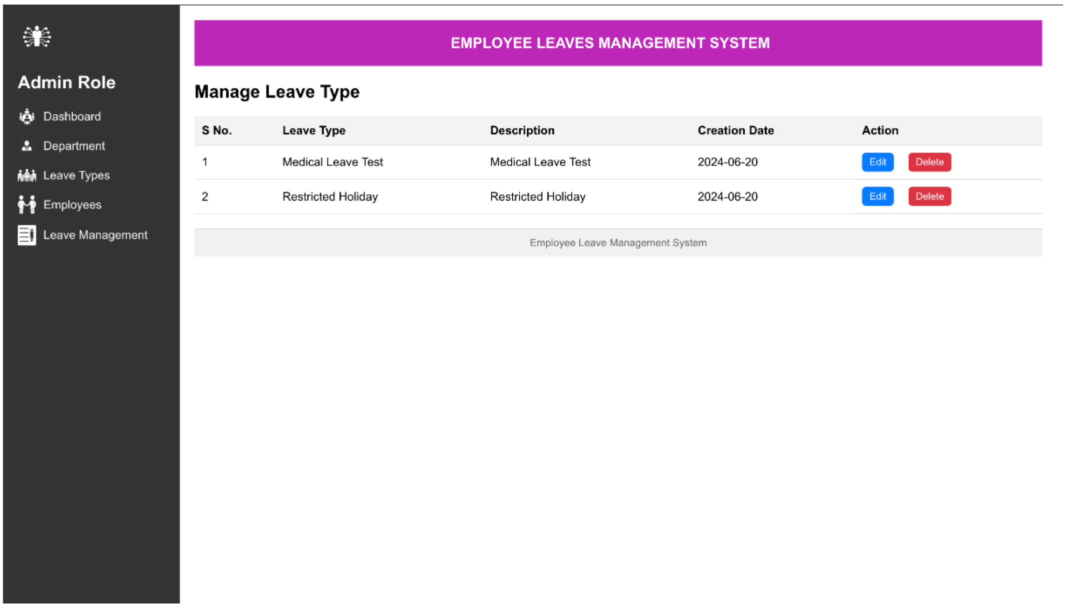
Manage leaves

S No.	Leave Id	Leave Name	Leave Description	Action
1	1001	Sick Leave	Leave Description	Edit Delete
2	1002	Restricted Holiday	Leave Description	Edit Delete
3	1003	Paid Leave	leave Description	Edit Delete
4	1004	Compensatory Off	Leave Description	Edit Delete

Employee Leave Management System

Screenshot-8

Manage Leaves Type Page :



EMPLOYEE LEAVES MANAGEMENT SYSTEM

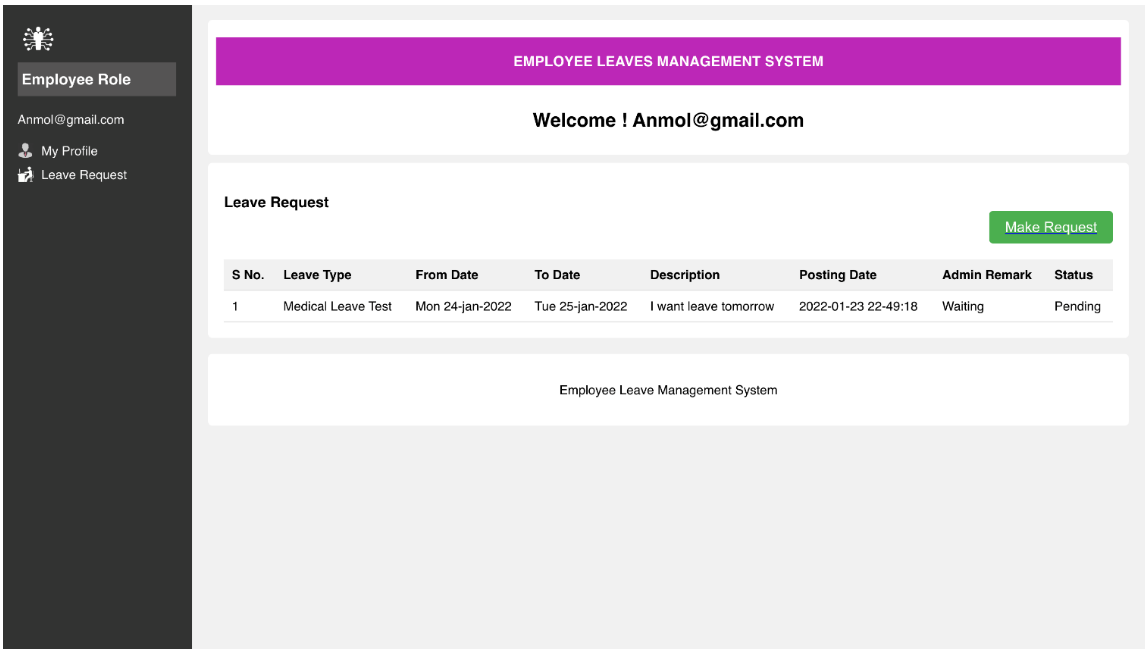
Manage Leave Type

S No.	Leave Type	Description	Creation Date	Action
1	Medical Leave Test	Medical Leave Test	2024-06-20	Edit Delete
2	Restricted Holiday	Restricted Holiday	2024-06-20	Edit Delete

Employee Leave Management System

Screenshot-9

Leave Request Page :



EMPLOYEE LEAVES MANAGEMENT SYSTEM

Welcome ! Anmol@gmail.com

Leave Request

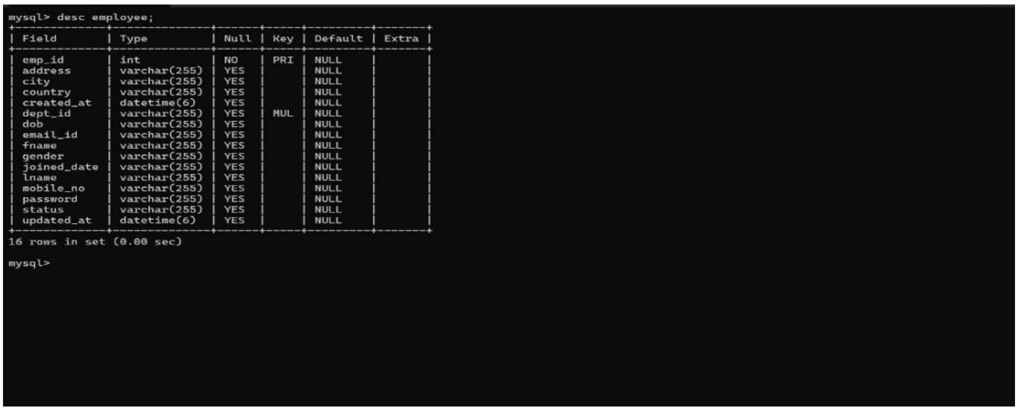
S No.	Leave Type	From Date	To Date	Description	Posting Date	Admin Remark	Status
1	Medical Leave Test	Mon 24-jan-2022	Tue 25-jan-2022	I want leave tomorrow	2022-01-23 22-49:18	Waiting	Pending

Employee Leave Management System

Screenshot-10

Database :

Employee :

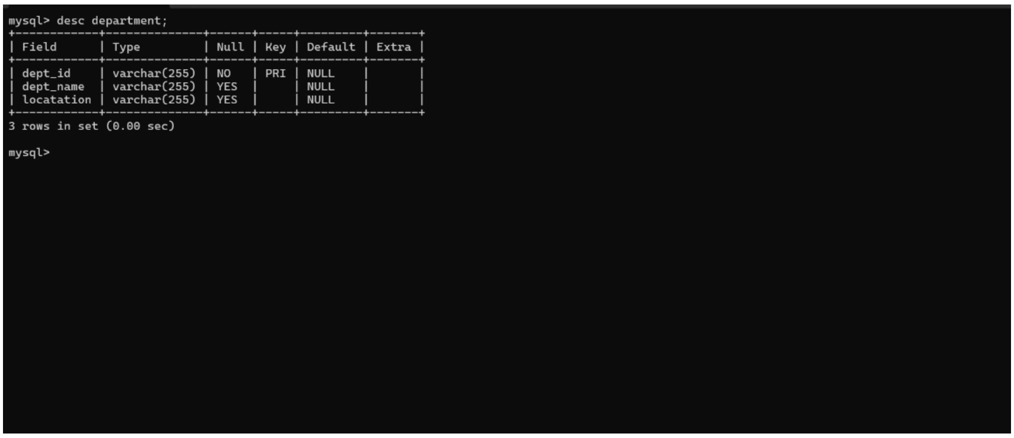


```
mysql> desc employee;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| emp_id      | int          | NO   | PRI | NULL    |       |
| address     | varchar(255) | YES  |     | NULL    |       |
| city        | varchar(255) | YES  |     | NULL    |       |
| country     | varchar(255) | YES  |     | NULL    |       |
| created_at  | datetime(6)  | YES  |     | NULL    |       |
| dept_id     | varchar(255) | YES  | MUL | NULL    |       |
| dob         | varchar(255) | YES  |     | NULL    |       |
| email_id    | varchar(255) | YES  |     | NULL    |       |
| fname       | varchar(255) | YES  |     | NULL    |       |
| gender      | varchar(255) | YES  |     | NULL    |       |
| joined_date | varchar(255) | YES  |     | NULL    |       |
| lname       | varchar(255) | YES  |     | NULL    |       |
| mobile_no   | varchar(255) | YES  |     | NULL    |       |
| password    | varchar(255) | YES  |     | NULL    |       |
| status      | varchar(255) | YES  |     | NULL    |       |
| updated_at  | datetime(6)  | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
16 rows in set (0.00 sec)

mysql>
```

Screenshot-5

Department:

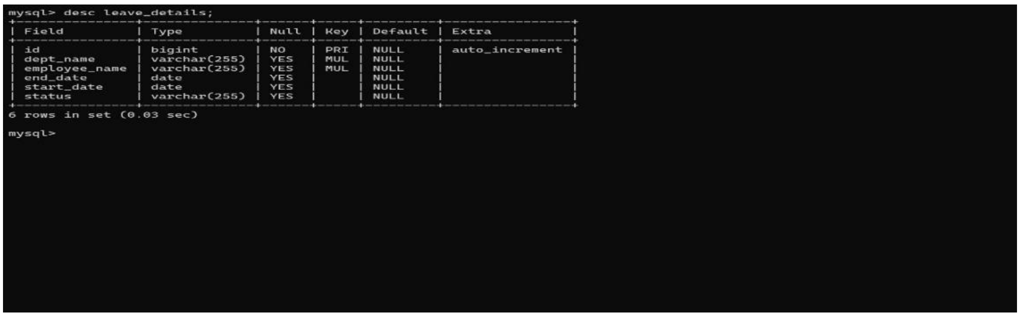


```
mysql> desc department;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| dept_id    | varchar(255) | NO   | PRI | NULL    |       |
| dept_name  | varchar(255) | YES  |     | NULL    |       |
| locatation | varchar(255) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql>
```

Screenshot-6

Leave Details :



```
mysql> desc leave_details;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | bigint       | NO   | PRI | NULL    | auto_increment |
| dept_name     | varchar(255) | YES  | MUL | NULL    |                |
| employee_name | varchar(255) | YES  | MUL | NULL    |                |
| end_date      | date         | YES  |     | NULL    |                |
| start_date    | date         | YES  |     | NULL    |                |
| status        | varchar(255) | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
6 rows in set (0.03 sec)

mysql>
```

Screenshot-7

Coding :

Functions:

The Leave Management System is designed using the Model-View-Controller (MVC) architectural pattern to efficiently manage employee leave requests and streamline administrative processes. The system architecture consists of three interconnected layers:

1. Model:

➢ **Representation**: The Model layer encapsulates the core data and business logic related to employee leave management.

➢ **Entities**: Includes entities such as Employee, Leave Request, Department, and Leave Type.

➢ **Business Logic**: Manages operations such as leave request processing, validation, employee profile management (including name, employee ID, contact details, DOB, address, and company email), and department management.

2. View:

➢ **Responsibility**: The View layer handles the presentation of user interfaces to both employees and administrators.

➢ **Components**: Includes HTML templates, CSS stylesheets, and React components for rendering pages like the leave application form, employee profile page, and the admin dashboard.

➢ **UI Elements**: User interfaces include leave application forms, profile management, detailed dashboards for admins, and sections for managing employees and departments.

3. Controller:

➢ **Function**: The Controller layer serves as the intermediary between the Model and View layers, processing user inputs and managing data flow.

➢ **Operations**: Includes controllers that handle user requests, invoke business logic in the Model layer, and render the appropriate views. It manages navigation, form submissions, and the integration between the user interface and backend operations.

➢ **Admin Dashboard Management**: Facilitates interactions with the admin dashboard, including managing leave types, employee records, department changes, and viewing recent leave actions.

Main Component of Backend:

Entry Point :

API Gateway :

- **Entity Class:**

```
package com.LeaveManagement.Entity;

import java.sql.Timestamp;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity
@Table(name="Employee")
public class Employee
{
	@Id
	@GeneratedValue(strategy = GenerationType.AUTO)
	@Column(name = "emp_id")
	private int emp_id;

	@Column(name = "fname") // Maps to the 'fname' column in the database
	private String fname;

	@Column(name = "lname") // Maps to the 'lname' column in the database
	private String lname;

	@Column(name = "email_id") // Maps to the 'email_id' column in the database
	private String email_id;

	@Column(name = "password") // Maps to the 'password' column in the database
	private String password;

	@Column(name = "gender") // Maps to the 'gender' column in the database
	private String gender;

	@Column(name = "dept_id") // Maps to the 'dept_id' column in the database
	private String dept_id;

	@Column(name = "joined_date") // Maps to the 'joined_date' column in the database
	private String joined_date;

	@Column(name = "mobile_no") // Maps to the 'mobile_no' column in the database
	private String mobile_no;

	@Column(name = "dob") // Maps to the 'dob' column in the database
```

```
    private String dob;

    @Column(name = "address") // Maps to the 'address' column in the database
    private String address;

    @Column(name = "city") // Maps to the 'city' column in the database
    private String city;

    @Column(name = "country") // Maps to the 'country' column in the database
    private String country;

    @Column(name = "status") // Maps to the 'status' column in the database
    private String status;

    @Column(name = "created_at") // Maps to the 'created_at' column in the database
    private Timestamp created_at = new Timestamp(System.currentTimeMillis());

    @Column(name = "updated_at") // Maps to the 'updated_at' column in the database
    private Timestamp updated_at = new Timestamp(System.currentTimeMillis());


    public Employee() {

    }

    public Employee(int emp_id, String fname, String lname, String email_id, String password, String
gender,
            String department, String joined_date, String mobile_no, String dob, String address,
String city,
            String country, String status, Timestamp created_at, Timestamp updated_at) {
        super();
        this.emp_id = emp_id;
        this.fname = fname;
        this.lname = lname;
        this.email_id = email_id;
        this.password = password;
        this.gender = gender;
        this.department = department;
        this.joined_date = joined_date;
        this.mobile_no = mobile_no;
        this.dob = dob;
        this.address = address;
        this.city = city;
        this.country = country;
        this.status = status;
        this.created_at = created_at;
        this.updated_at = updated_at;
    }

    public int getEmp_id() {
        return emp_id;
    }
    public void setEmp_id(int emp_id) {
        this.emp_id = emp_id;
    }
    public String getFname() {
```

```
		return fname;
	}
	public void setFname(String fname) {
		this.fname = fname;
	}
	public String getLname() {
		return lname;
	}
	public void setLname(String lname) {
		this.lname = lname;
	}
	public String getEmail_id() {
		return email_id;
	}
	public void setEmail_id(String email_id) {
		this.email_id = email_id;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
	public String getGender() {
		return gender;
	}
	public void setGender(String gender) {
		this.gender = gender;
	}
	public String getDepartment() {
		return department;
	}
	public void setDepartment(String department) {
		this.department = department;
	}
	public String getJoined_date() {
		return joined_date;
	}
	public void setJoined_date(String joined_date) {
		this.joined_date = joined_date;
	}
	public String getMobile_no() {
		return mobile_no;
	}
	public void setMobile_no(String mobile_no) {
		this.mobile_no = mobile_no;
	}
	public String getDob() {
		return dob;
	}
	public void setDob(String dob) {
		this.dob = dob;
	}
	public String getAddress() {
		return address;
	}
	public void setAddress(String address) {
		this.address = address;
	}
	public String getCity() {
		return city;
	}
	public void setCity(String city) {
		this.city = city;
	}
	public String getCountry() {
		return country;
	}
	public void setCountry(String country) {
		this.country = country;
	}
	public String getStatus() {
		return status;
	}
	public void setStatus(String status) {
		this.status = status;
	}
	public Timestamp getCreated_at() {
		return created_at;
	}
	public void setCreated_at(Timestamp created_at) {
		this.created_at = created_at;
	}
	public Timestamp getUpdated_at() {
		return updated_at;
	}
	public void setUpdated_at(Timestamp updated_at) {
		this.updated_at = updated_at;
	}

	@Override
	public String toString() {
		return "Employee [emp_id=" + emp_id + ", fname=" + fname + ", lname=" + lname + ", 
email_id=" + email_id
				+ ", password=" + password + ", gender=" + gender + ", department=" + 
department + ", joined_date="
				+ joined_date + ", mobile_no=" + mobile_no + ", dob=" + dob + ", address=" 
+ address + ", city=" + city
				+ ", country=" + country + ", status=" + status + ", created_at=" + created_at + 
", updated_at="
				+ updated_at + "]";
	}

}
```

- **Controller:**

```
package com.LeaveManagement.Controllers;
import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import com.LeaveManagement.Entity.Employee;
import com.LeaveManagement.Services.EmployeeService;

@RestController
public class EmployeeController {

    @Autowired
    private EmployeeService employeeService;

    @GetMapping(value = "/employees")
    public ResponseEntity<List<Employee>> getEmployees()
    {
            List<Employee> list = employeeService.getAllEmployees();
            if(list.size()<=0)
            {
                    return ResponseEntity.status(HttpStatus.NOT_FOUND).build();
            }
            return ResponseEntity.of(Optional.of(list)) ;
    }


    @GetMapping("/employees/{id}")
    public ResponseEntity<Employee> getEmployee(@PathVariable("id") int id)
    {
            Employee emp = employeeService.getEmployeeById(id);
            if(emp==null)
                    return ResponseEntity.status(HttpStatus.NOT_FOUND).build();
            return ResponseEntity.of(Optional.of(emp));
    }

    @PostMapping("/employee")
    public ResponseEntity<Optional<Employee>> addEmployee(@RequestBody Employee emp)
    {
            Employee e = null;
            try
            {
                    e = this.employeeService.addEmployee(emp);
                    return ResponseEntity.ok(Optional.of(e));
            }
            catch(Exception ex)
            {
                    ex.printStackTrace();
                    return
ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).build();
            }
    }

    @DeleteMapping("/employees/{emp_id}")
    public ResponseEntity<Void> deleteEmployee(@PathVariable("emp_id") int emp_id)
    {
        try
        {
            this.employeeService.deleteEmployee(emp_id);
            return ResponseEntity.status(HttpStatus.NO_CONTENT).build();
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
            return
ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).build();
        }

    }

    @PutMapping("/employees/{emp_id}")
    public ResponseEntity<Employee> updateEmployee(@RequestBody Employee emp,
@PathVariable("emp_id") int emp_id)
    {
        try
        {
            this.employeeService.updateEmployee(emp, emp_id);
            return ResponseEntity.ok().body(emp);
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
            return
ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).build();
        }

    }
}
```

- **Repository:**

```
package com.LeaveManagement.Repositories;
import java.util.List;

import org.springframework.data.repository.CrudRepository;
import com.LeaveManagement.Entity.Employee;

public interface EmployeeRepository extends CrudRepository<Employee, Integer>
{

 public Employee findById(int id);
}
```

- **Services:**

```
package com.LeaveManagement.Services;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import com.LeaveManagement.Entity.Employee;
import com.LeaveManagement.Repositories.EmployeeRepository;

@Component
public class EmployeeService
{
	@Autowired
	private EmployeeRepository employeeRepository;

	//Get all employees
	public List<Employee> getAllEmployees()
	{
		List<Employee> list = (List<Employee>) this.employeeRepository.findAll();
		return list ;
	}

	//Get single employee by id
	public Employee getEmployeeById(int id)
	{
		Employee emp = null;
		try
		{
			emp = this.employeeRepository.findById(id);
		}
		catch(Exception ex)
		{
			ex.printStackTrace();
		}
		return emp;
	}

	//Add new employee
	public Employee addEmployee(Employee emp)
	{
		Employee result = employeeRepository.save(emp);
		return result;
	}

	//Delete employee
	public void deleteEmployee(int id)
	{
		employeeRepository.deleteById(id);
	}

	//Update employee
	public void updateEmployee(Employee emp, int emp_id)
	{
		emp.setEmp_id(emp_id);
		employeeRepository.save(emp); }
```

FUTURE SCOPE

In the future, we plan to advance the Leave Management System by integrating sophisticated analytics tools to enhance the efficiency of leave processing and resource allocation. We aim to introduce real-time tracking and reporting features to provide immediate updates on leave requests and approvals. Additionally, we will incorporate tools to facilitate better employee engagement and communication regarding leave policies and procedures. Improving accessibility and expanding features to support various types of leave and flexible work arrangements will be key priorities. These enhancements will help us create a more interactive, efficient, and inclusive leave management experience for employees and administrators alike.

CONCLUSION

The Leave Management System has redefined traditional leave management practices, offering a comprehensive platform that simplifies the process of handling employee leave requests and streamlines administrative tasks. With features like a detailed employee profile page, real-time leave tracking, and a robust admin dashboard, the system transforms the management of employee leaves into a more efficient and effective experience.

Looking ahead, the integration of advanced analytics, real-time reporting, and enhanced employee engagement tools holds significant promise for further improving the platform’s capabilities. These advancements will allow the Leave Management System to better cater to the diverse needs of organizations, providing more responsive and tailored leave management solutions.

Additionally, expanding the system’s functionalities, improving accessibility, and focusing on broader outreach efforts will ensure that the Leave Management System serves a wider range of users and remains inclusive and user-friendly for all.

In conclusion, the Leave Management System has set a new benchmark in leave management technology, paving the way for a more streamlined, efficient, and inclusive approach to handling employee leave requests. As we continue to innovate and enhance the platform, the Leave Management System remains dedicated to shaping the future of leave management, one request at a time.

Bibliography:

We kept window open for further additional development.

- https://react.dev/

- https://spring.io/projects/spring-boot

- https://www.w3schools.com/