**CHAPTER 1 INTRODUCTION**

* 1. **Evaluation of System**

The Leave Management System, also known as the Employee Time Off System, is an effective tool for managing employee leave requests. It provides an easy-to-use interface for both employees and managers, allowing them to request and approve leave requests respectively. The system has various features such as a calendar view, automatic leave accruals, and email notifications, which make it an efficient and convenient way of managing employee time off.

The system has been designed keeping in mind the needs of both employees and managers, and its user-friendly interface ensures that it can be easily used by anyone without requiring any specialized training. The system has also been tested thoroughly to ensure its reliability, and various security measures have been implemented to protect employee data and prevent unauthorized access.

Overall, the Leave Management System has proven to be a valuable addition to the organization's HR management tools, making it easier to manage employee time off requests and ensuring that there is no disruption to work schedules. The system has received positive feedback from both employees and managers, who appreciate its ease of use and the convenience it provides.

* 1. **Problem definition**

The current manual process of managing employee leaves is time-consuming and error-prone. The HR department spends a lot of time managing leaves, tracking employee leave balances, and approving leave requests. Employees also face difficulties in applying for leave and tracking their leave balances. This leads to a lack of transparency and accountability in the leave management process.

To solve these problems, we propose the development of an Employee Time Off System (ETOS) - a web-based application that automates the entire leave management process. The ETOS will provide a user-friendly interface for employees to apply for leave, track their leave balances, and view their leave history. It will also provide HR administrators with a dashboard to manage employee leaves, approve or reject leave requests, and generate reports. With the ETOS, the leave management process will be streamlined, efficient, and transparent for both employees and HR administrators.

* 1. **Proposed System**

The proposed system for the employee time off system aims to provide a more efficient and streamlined process for managing employee leave requests. This system will be designed to address the shortcomings of the existing manual process which can often be time-consuming and prone to errors.

The first feature of the proposed system is an online leave request form which employees can use to submit their leave requests. This will allow employees to request leave from anywhere, at any time, and ensure that all required information is provided in a standardized format. The form will be designed to be user-friendly and easy to navigate.

Once a leave request is submitted, the system will automatically send an email notification to the employee's supervisor requesting approval. The supervisor will be able to review the request and either approve or reject it. The system will also provide an option for the supervisor to request additional information if needed.

If the leave request is approved, the system will automatically update the employee's leave balance and send an email notification to the employee confirming the approval. If the leave request is rejected, the system will send an email notification to the employee with the reason for the rejection.

Another key feature of the proposed system is a dashboard which will provide real-time visibility into employee leave balances and the status of leave requests. This will allow supervisors to quickly and easily see which employees are on leave and plan accordingly.

Overall, the proposed system for the leave management system aims to improve the efficiency and accuracy of the leave management process by providing a user-friendly, automated system for managing leave requests. By reducing the time and effort required for managing employee leave, this system will allow organizations to focus on more important tasks, ultimately leading to increased productivity and employee satisfaction.

**1.4 Scope Of Work/Project**

The scope of the leave management system project is to create a web-based application that automates the leave management process for employees in an organization. The system will provide an efficient and effective way of handling leave requests, approvals, and tracking of leave balances for all employees. The system will have two main modules, one for employees and one for managers.

The employee module will allow employees to submit leave requests, view their leave balances, and view the status of their leave requests. Employees will be able to select the type of leave they are requesting, the start and end dates, and a reason for the leave. Once a leave request is submitted, the employee will receive an email notification with the status of their request.

The manager module will allow managers to view and approve or reject leave requests for their team members. Managers will be able to see the leave balances of their team members and track the status of all leave requests. The system will also send email notifications to managers when new leave requests are submitted, and when they need to take action on a request.

The system will also have an administrative module, which will allow administrators to manage employee accounts, leave types, and leave policies. Administrators will be able to view reports on leave usage, leave balances, and other leave-related statistics.

Overall, the leave management system will provide a user-friendly interface for employees and managers to manage the leave process efficiently and effectively. The system will reduce the time and effort required to manage leave requests, improve transparency and communication between employees and managers, and ensure compliance with leave policies and regulations.

**1.5 Report Organization**

The report for the Leave Management System project will be organized in a structured manner to ensure clarity and ease of understanding. The report will include the following sections:

1. **Introduction:** This section will provide an overview of the project, its objectives, and the problem it aims to solve.
2. **Problem Definition:** This section will outline the challenges faced by the organization with respect to managing employee leaves, including the lack of an automated system and the manual and time-consuming nature of the existing system.
3. **Proposed System:** This section will describe the new system proposed for the organization, including its features and functionalities, and how it will address the challenges outlined in the problem definition.
4. **Scope of Work:** This section will provide details on the scope of work for the project, including the tasks to be completed, the timelines for completion, and the resources required.
5. **System Design and Architecture:** This section will provide a detailed description of the system design and architecture, including the database schema, the user interface, and the programming languages and tools used for development.
6. **Implementation:** This section will detail the implementation process, including the steps taken to develop and test the system.
7. **Results and Evaluation:** This section will provide an evaluation of the system's performance, including its effectiveness in managing employee leaves, the ease of use of the system, and feedback from users.
8. **Conclusion:** This section will provide a summary of the project, including its objectives, the challenges addressed, and the outcomes achieved.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Presently available system**

Here is a list of some of the currently available leave management systems:

1. Zoho People
2. ADP Workforce Now
3. LeaveMaster
4. CakeHR
5. TimeOffManager
6. LeaveBoard
7. BambooHR
8. Zenefits
9. e-days
10. Factorial HR.

**2.2 Name of Author/ Name of web site/Book**

Here are some links to real-time employee time off systems:

1. Time Off Manager - <https://www.timeoffmanager.com/>
2. Leave Dates - <https://leavedates.com/>
3. e-days - <https://www.e-days.com/>
4. LeavePlanner - <https://leaveplanner.com/>
5. Absence.io - <https://www.absence.io/>

**2.3 Conclusion: Include required facts, fig , Tables, Diagrams, and Architecture etc.**

The Employee Time Off System is an efficient and user-friendly platform that streamlines the leave management process for organizations. Through the use of the system, employees can easily request for time off and track the status of their requests. The system also provides a comprehensive view of employee leave data, enabling management to make informed decisions.

The project involved the analysis of requirements, the design of the system, and its implementation using PHP, HTML, CSS, Bootstrap, and MySQL. The system features an intuitive user interface, secure authentication, and authorization mechanisms, and a robust database management system.

Overall, the Employee Time Off System provides an effective solution to leave management for organizations, reducing administrative overhead, increasing transparency, and improving employee satisfaction.

**CHAPTER 3**

**ANALYSIS**

* 1. **Requirement Analysis**

In systems engineering and software engineering, requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

Requirement analysis is a crucial step in the development of any software system. In the case of the Employee Time Off System, the following requirements have been identified:

* User Management: The system should allow the HR team to add, edit and delete employee information such as name, email address, employee ID, and department.
* Leave Management: The system should allow employees to request for leaves, and the HR team should be able to approve or reject these requests. The system should also keep track of the number of leaves taken by each employee.
* Dashboard: The system should have a dashboard that provides an overview of the employee leave requests, the number of leaves remaining for each employee, and the status of leave requests.
* Notifications: The system should send notifications to employees when their leave requests are approved or rejected. The HR team should also be notified when an employee makes a leave request.
* Reporting: The system should be able to generate reports on employee leave requests and the number of leaves taken by each employee.
* Security: The system should have robust security measures in place to protect employee data and prevent unauthorized access.
* Accessibility: The system should be accessible from any device with an internet connection.

By identifying and analyzing these requirements, we can ensure that the Employee Time Off System meets the needs of both the employees and the HR team.

* 1. **Use-Case Diagram**

Add Employees

Emp\_Rep

Leave Request

Employee Login

Admin Login

Approve/Reject leave

Apply for Leave

Leave History

Admin

Employee

* 1. **Use-Case Description**

The use case diagram are usually referred to as behavior diagram used to describe the actions of all user in a system. All user describe in use case are actors and the functionality as action of system.

In this system a simple user who is user or candidate have to register in our website and after that, He or she can login , after login he directly jump to his dashboard go to the

This Use Case Diagram is a graphic depiction of the interactions among the elements of Smart Attendance System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Smart Attendance System. The main actors of Smart Attendance System in this Use Case Diagram are: \ User, Anonymous Users, who perform the different type of use cases such as Open Image, Edit Image, Crop Image, Resize Image, and Full Smart Attendance System Operations.

* 1. **Sequence Diagram**

Employee

Leave

Admin

Apply for leave

Receive leave Requests

Approve/Reject leave

Receive Status

* 1. **Activity Diagram**

Login to the Employee time off system

Check Employee leave and permissions

Check Permission

Check Permission

Check Permissions

Check Permission

Manage leaves

Manage Approve Leaves

Manage Cancel Leaves

Manage Employee

Logout Form the System

**CHAPTER 4**

**DESIGN**

**4.1 Object Oriented Design**

In the object-oriented design method, the system is viewed as a collection of objects (i.e., entities). The state is distributed among the objects, and each object handles its state data. For example, in a Library Automation Software, each library representative may be a separate object with its data and functions to operate on these data. The tasks defined for one purpose cannot refer or change data of other objects. Objects have their internal data which represent their state. Similar objects create a class. In other words, each object is a member of some class. Classes may inherit features from the super class.

Object-oriented analysis and design (OOAD) is a technical approach for analyzing and designing an application, system, or business by applying object-oriented programming, as well as using visual modeling throughout the software development process to guide stakeholder communication and product quality.

**4.2 Class Diagram**

Emp\_Id : int

EmpName : String

DOB : Date

DOJ : Date

Department : String

Leave\_id

Start\_date

End\_date

Reason

Status

Admin\_id

Admin\_name

Email

Password

Employee

Leave

**Admin**

Apply\_for\_leave()

View\_leave\_request()

Update\_status()

View leave requests()

Approve\_leave()

Rejectleave()

**4.3 Class Description**

Here are brief descriptions of the Employee, Leave, and Admin classes based on their names:

1. Employee class: This class represents an employee in the system. It may contain attributes such as employee ID, name, address, contact information, date of hire, job title, and department. It may also have methods for updating employee information, retrieving employee information, and performing other tasks related to employee management.
2. Leave class: This class represents a leave request made by an employee. It may contain attributes such as leave ID, employee ID, leave start date, leave end date, type of leave (e.g., vacation, sick leave), and reason for the leave request. It may also have methods for submitting a leave request, approving or denying a leave request, and managing other aspects of the leave request process.
3. Admin class: This class represents an administrator or manager in the system who has the ability to manage employee and leave information. It may contain attributes such as admin ID, name, username, password, and access level. It may also have methods for logging into the system, managing employee information (e.g., adding, deleting, or updating employee information), managing leave requests (e.g., approving or denying leave requests), and other administrative tasks.

**4.4 Data Model**

A data model is a conceptual representation of the data that is used by a particular system or organization. It describes the structure, relationships, and constraints of the data in a way that is independent of any specific technology or implementation.

A data model is typically created during the early stages of system development as part of the requirements gathering and analysis process. It is used to provide a common understanding of the data that will be used by the system, and to help ensure that the data is organized in a way that supports the system's functional requirements.

**4.4.1 Data Flow Diagram**

DFD stands for Data Flow Diagram, which is a graphical representation of how data moves through a system. It is a modeling technique used in software engineering and business analysis to represent the flow of data and processes within a system.

In a DFD, the system is represented as a set of processes that transform input data into output data. The processes are connected by arrows that represent the flow of data between them. The data itself is represented as entities or data stores, which are depicted as rectangles.

Leave Managment

Employee managment

Salary managment

User managment

Login managment

Leave Status managment

0 level DFD of Employee time off system

Admin Login

Employee Login

Successful

Successful

Check Leave app.

Apply for leave

Approve/Reject leave app.

Update Recorde

Check app. status

1 level Data flow diagram

**4.4.2 Entity Relationship Diagram (ERD)**

Emp\_login

Admin

Employee

Admin

Hasas

Hasas

Manages

Apply

Review

Leave\_Req

**CHAPTER 5**

**IMPLEMENTATION AND TESTING**

**5.1 Testing Strategies adapted**

Testing is a process, which reveals errors in the program. It is the major quality   
measure employed during software development. During software development.   
During testing, the program is executed with a set of test cases and the output of the   
program for the test cases is evaluated to determine if the program is performing as it   
is expected to perform.

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

* **Unit Testing:**

Unit Testing is done on individual modules as they are completed and become   
executable. It is confined only to the designer's requirements.

Each module can be tested  using the following two Strategies:

**i) Black Box Testing:**

In this strategy some test cases are generated as input conditions that fully execute all   
functional requirements for the program. This testing has been uses to find errors in   
the following categories:   
a) Incorrect or missing functions   
b) Interface errors   
c) Errors in data structure or external database access   
d) Performance errors   
e) Initialization and termination errors.

**ii) White Box Testing :**

In this the test cases are generated on the logic of each module by drawing flow   
graphs of that module and logical decisions are tested on all the cases. It has been uses   
to generate the test cases in the following cases:

a) Guarantee that all independent paths have been Executed.   
b) Execute all logical decisions on their true and false Sides.   
c) Execute all loops at their boundaries and within their   
operational bounds.   
d) Execute internal data structures to ensure their validity.

* **Integrating Testing:**

Integration testing ensures that software and subsystems work together a whole. It   
tests the interface of all the modules to make sure that the modules behave properly   
when integrated together.

* **System Testing:**

Involves in-system testing of the entire system before delivery to the user. It's aim is to   
satisfy the user the system meets all requirements of the client's specifications. 

* **Acceptance Testing:**

It is a pre-delivery testing in which entire system is tested at client's site on   
real world data to find errors. 

**5.2 SYSTEM TESTING**

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. The system testing is the stage of implementation, which is aimed at ensuring that the system works efficiently and accurately before live operation commences. System testing requires a test plan that consists of several key activities and steps for program, system and user acceptance testing.

**5.3 Test Cases**

1. Test case for user login:

* Enter valid username and password.
* Enter invalid username and password.
* Verify that user is directed to the correct page after logging in.

1. Test case for applying for leave:

* Apply for a leave request with valid information.
* Apply for a leave request with invalid or incomplete information.
* Verify that the leave request is saved in the system with the correct information.
* Verify that the user is notified about the status of the leave request.

1. Test case for leave approval:

* Approve a leave request with valid information.
* Approve a leave request with invalid or incomplete information.
* Reject a leave request with valid information.
* Reject a leave request with invalid or incomplete information.
* Verify that the user who requested the leave is notified about the status of the request.
* Verify that the user who approved or rejected the leave is recorded in the system.
* Test case for leave balance calculation:
* Verify that the user's leave balance is calculated correctly based on their employment status and leave policies.
* Verify that the user is notified when their leave balance is low or they are approaching their maximum leave limit.

1. Test case for leave report generation:

* Generate a report of all leave requests for a given time period.

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

**Appendix A Software Requirement Specification (SRS)**

SRS stands for Software Requirements Specification. It is a document that describes the software product to be developed, including the functional and non-functional requirements, design constraints, and other characteristics of the software. The SRS document serves as a foundation for the software development process and helps ensure that the software product meets the needs and expectations of the stakeholders.

The SRS document typically includes the following information:

1. Introduction: This section provides an overview of the software product and describes the purpose and scope of the document.
2. Functional Requirements: This section describes the specific functions that the software product must perform to meet the needs of the stakeholders. It includes use cases, user stories, and other functional requirements.
3. Non-functional Requirements: This section describes the characteristics of the software product that are not related to its specific functions, such as performance, reliability, security, and usability.
4. Design Constraints: This section describes any constraints or limitations that may affect the design or implementation of the software product, such as hardware or software compatibility requirements.
5. External Interface Requirements: This section describes the interfaces between the software product and other systems or components, including hardware, software, and human interfaces.
6. Other Requirements: This section may include any other requirements that are not covered in the previous sections, such as documentation, training, or support requirements.

The SRS document is typically developed by the software development team in collaboration with the stakeholders, including the customers, users, and other interested parties. It serves as a contract between the development team and the stakeholders, defining the expectations and requirements for the software product.

**The SRS phase consists of two basic activities:**

**i) Problem/Requirement Analysis:**

job search is a harsh reality experienced by many college graduates nowadays. Numerous obstacles, including the tough job market, severely hinder the future success of a student who is truly determined. The transition from school to the labor market is extremely alarming to students that they usually end up satisfied with a tedious, low-paid, labor-intensive and physically demanding job.

Working a job that does not require a bachelor’s degree completely de-utilizes the education of students. The first and main problem students’ face is their lack of experience when applying to a job within their field of study. Internships are competitive and hard to acquire but still it is needed for experience.

**O**ur Project solve this problem on the basis of local area

The process is order and more nebulous of the two, deals with understand the   
problem, the goal and constraints.

**ii) Requirement Specification:**

Here, the focus is on specifying what has been found giving analysis such as   
representation, specification languages and tools, and checking the specifications are   
addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document.   
Producing the SRS document is the basic goal of this phase.

**PURPOSE:**

Our Purpose behind this project to help those people who are not easily able to find the job and also help him to find the local area jobs.

**SCOPE:**

This document is the only one that describes the requirements of the system. It   
is meant for the use by the developers, and will also by the basis for validating the   
final delivered system. Any changes made to the requirements in the future will have   
to go through a formal change approval process.

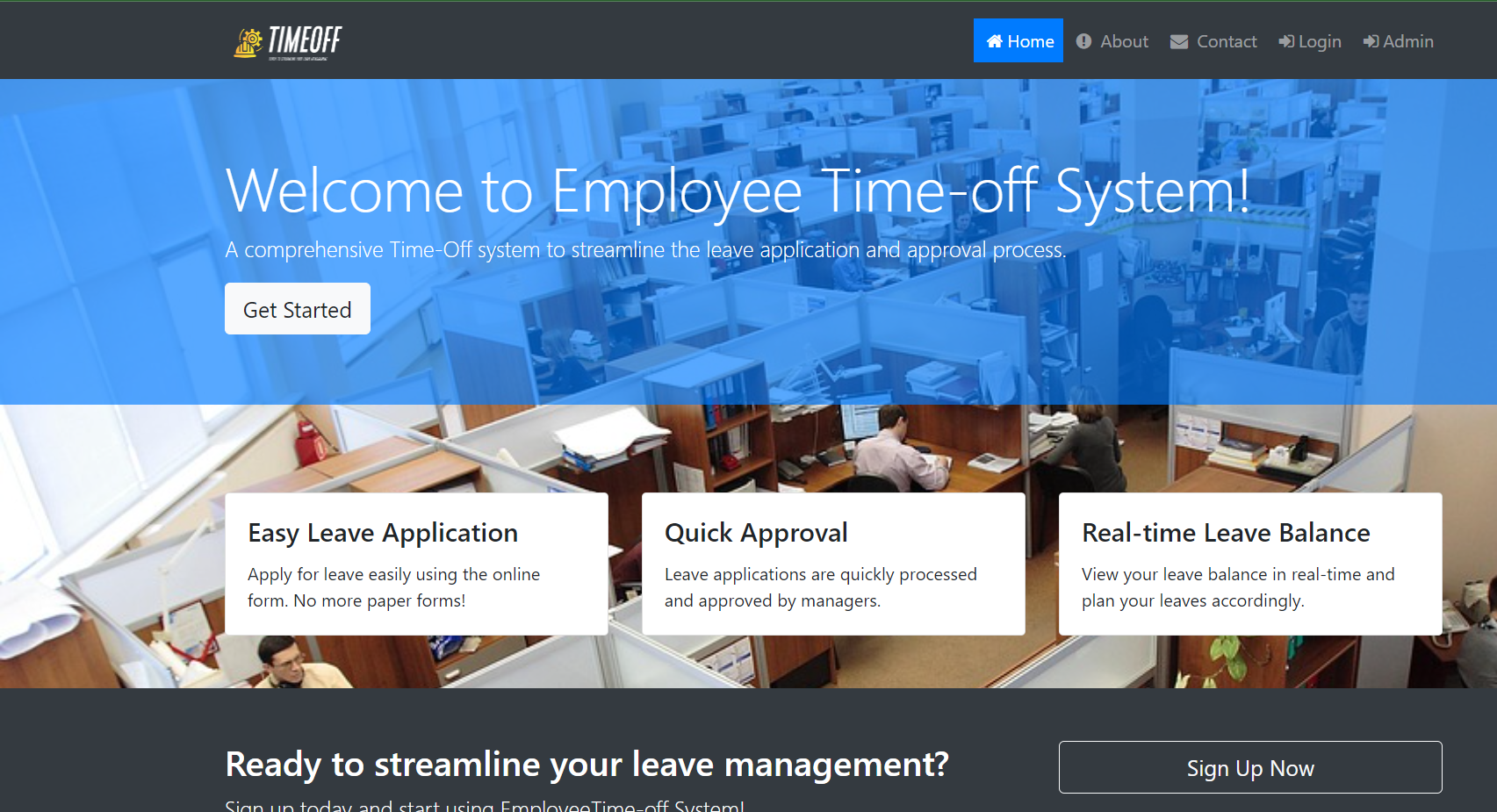
**HARDWARE SPECIFICATION**

|  |  |
| --- | --- |
| * Processor | Intel i3 or higher |
| * RAM Capacity | 4 GB or higher |
| * Hard Disk | 1 TB or higher |

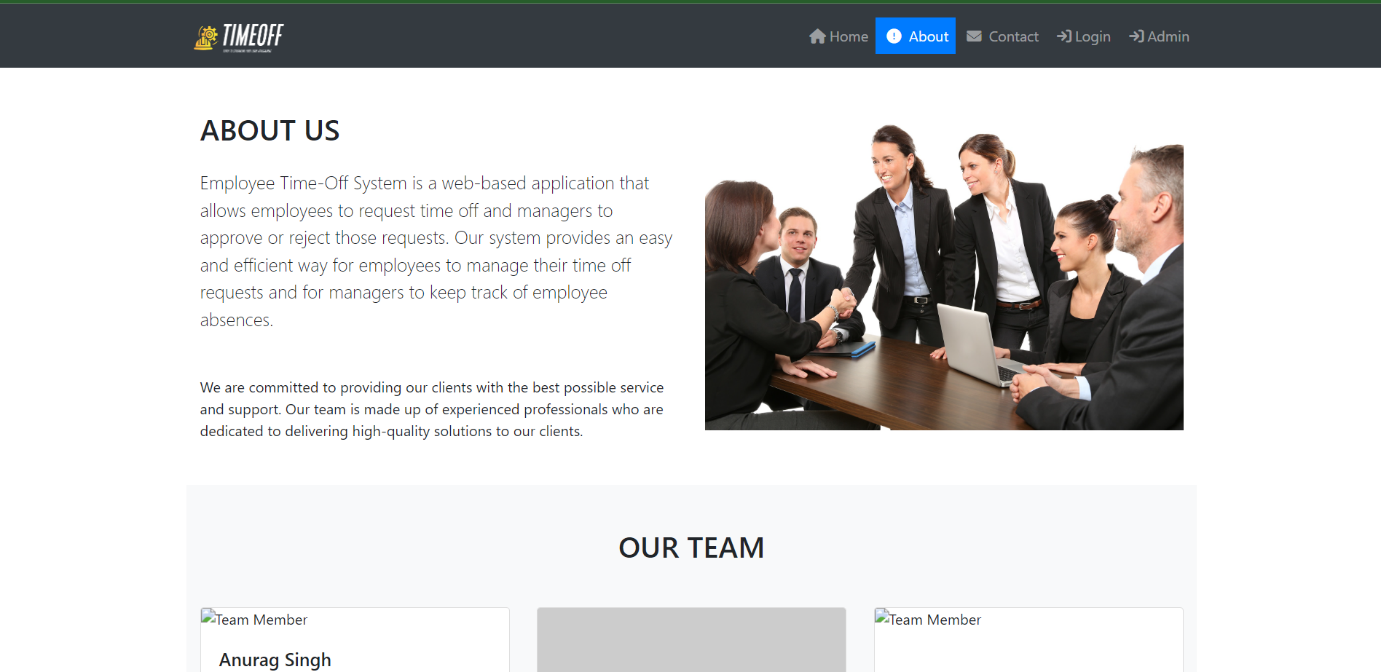
|  |  |
| --- | --- |
| * Operating System | Windows |
| * Editor | VS Code etc. |
| * Server | XAMPP |
| * Backend Database | PhpMyAdmin (mysql) |
| * Web-Technology | HTML, CSS, JavaScript |

**Appendix C Screenshots**

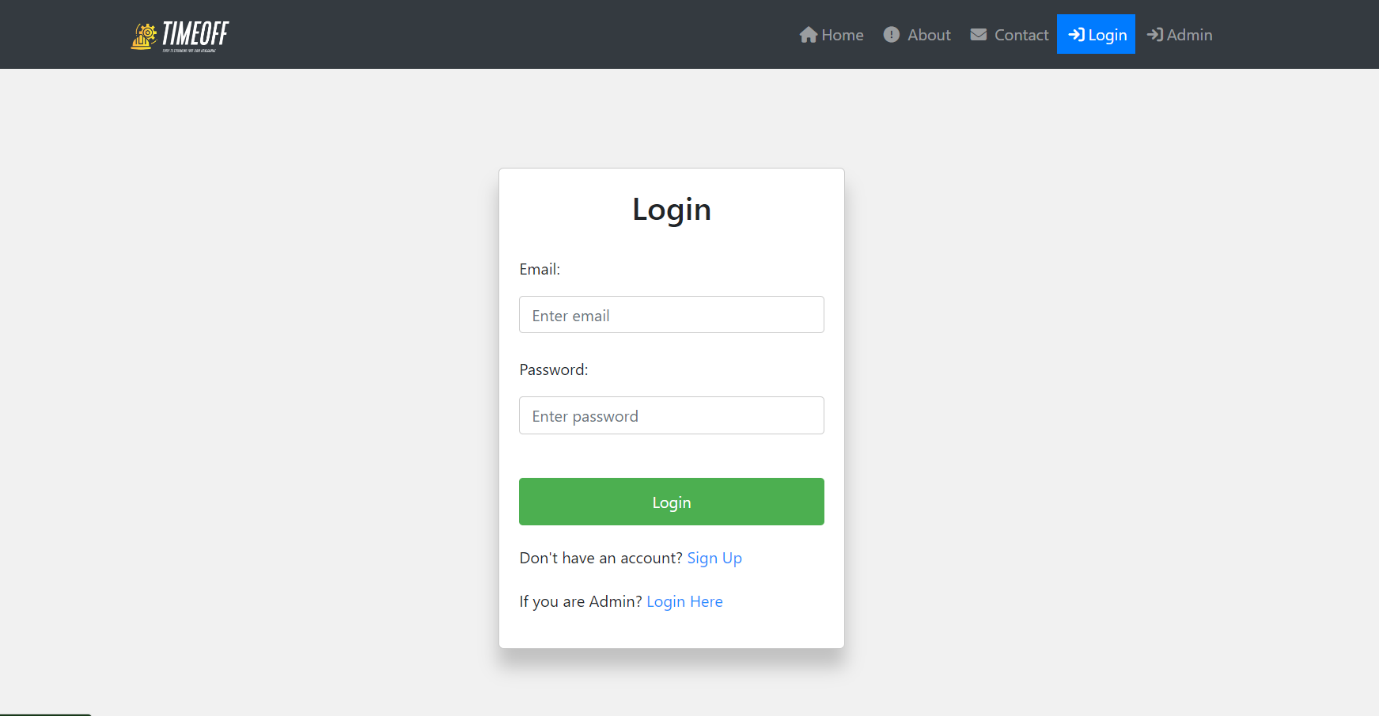
**Index page**

****

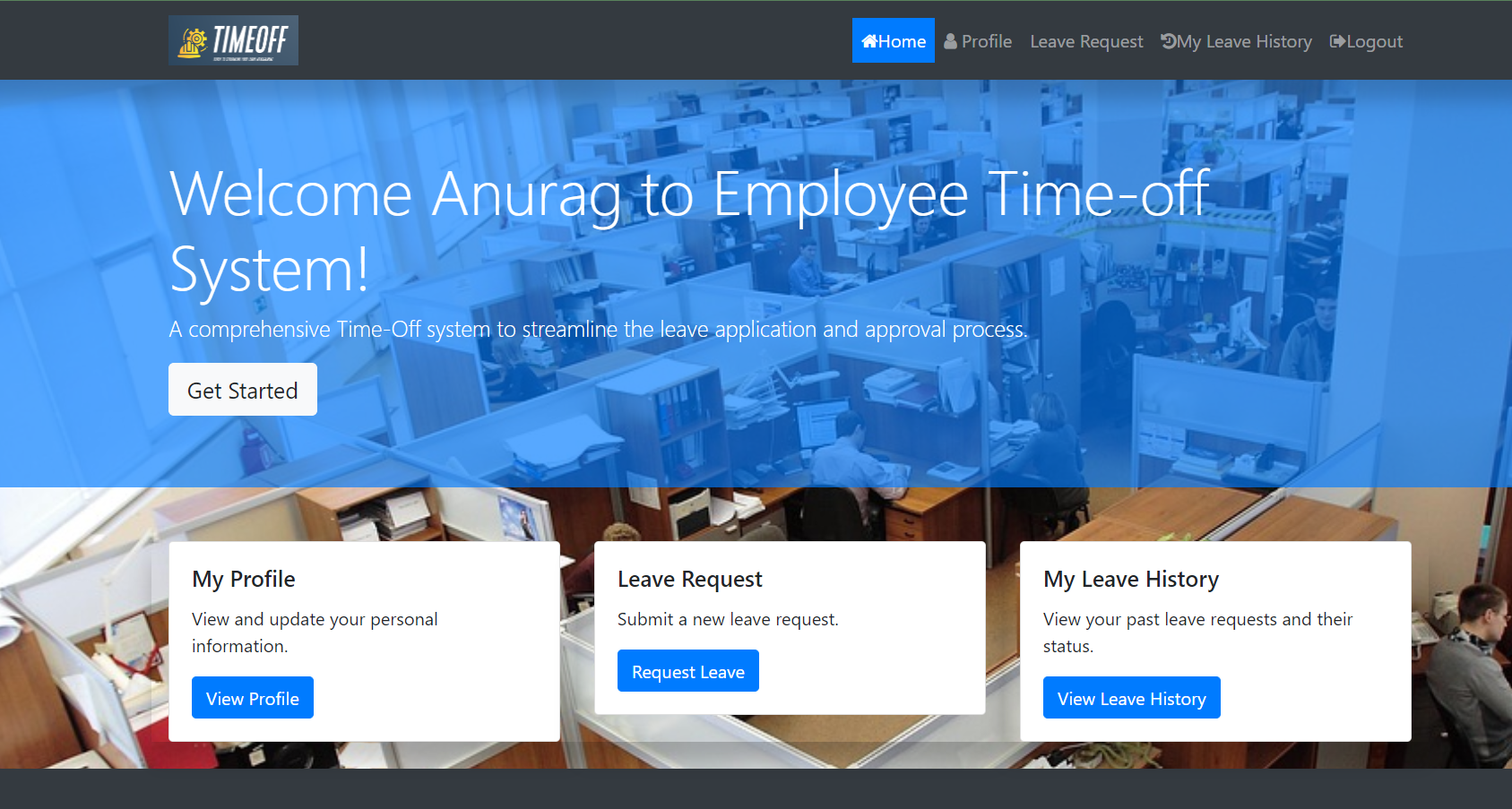
**About Us**

****

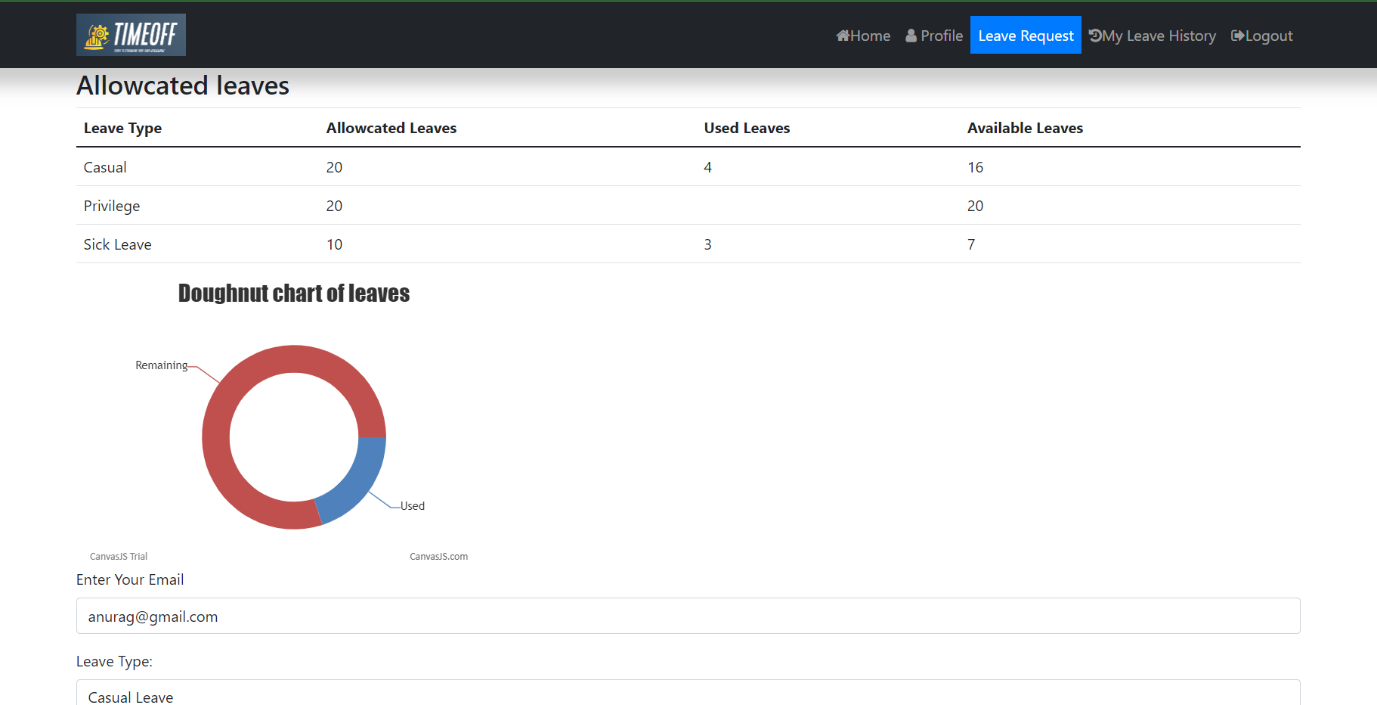
**Login**

****

**Dashboard**

****

**Leave Request**

****