"Leave Management System"

A PROJECT REPORT SUBMITTED TO

THE NATIONAL INSTITUTE OF ENGINEERING, MYSURU

(An Autonomous Institute under VTU, Belagavi)

In partial fulfilment of the requirements for Project work (Database Laboratory CS5L02), fifth semester

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by

Pratham Vijay Vargiya (4NI20CS068)

Vigneshwar Ravindran(4NI20CS121)

Naman Arbind (4NI20CS058)

Under the Guidance of

Narender M
Assistant Professor

Padmini M S
Assistant Professor

Priyanka R V
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2022-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THE NATIONAL INSTITUTE OF ENGINEERING

CERTIFICATE

This is to certify that the project work entitled "**Leave Management System**" is a work carried out by **Pratham Vijay Vargiya (4NI20CS068), Vigneshwar Ravindran (4NI20CS121) and Naman Arbind (4NI20CS058)** in partial fulfilment for the project work (Database Laboratory – CS5L02), fifth semester, Computer Science & Engineering, The National Institute of Engineering **(**Autonomous Institution under Visvesvaraya Technological University, Belagavi) during the academic year 2022-2023. It is certified that all corrections and suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the department library. The project work report has been approved in partial fulfilment as per academic regulations of The National Institute of Engineering, Mysuru.

Signature of the Internal Guides **Signature of the HoD**

_____ _____

Dr. Yuvaraju B N
Professor and Head
Dept. of CS&E
NIE, Mysuru

Signature of the Examiners with date

_____ _____

Name:
Designation:

Name:
Designation:

Acknowledgements

We are happy to state that the project of designing and developing the report on **"Leave Management System"** has successfully completed under the given constraints and with the expected outcome.

We express our profound thanks to **Dr. Rohini Nagapadma**, Principal, The National Institute of Engineering (South Campus), Mysuru for her much-needed moral support and encouragement.

We are highly obliged to our H.O.D **Dr. Yuvaraju B N**, Professor and Head Department of Computer Science and Engineering NIE, Mysuru for providing us timely help and support throughout the project**.**

A special gratitude we give to our internal guides**, Mr. Narender M,** Assistant Professor Department of Computer Science and Engineering and **Ms. Padmini M S,** Assistant Professor Department of Computer Science and Engineering and **Ms. Priyanka R V,** Assistant Professor Department of Computer Science and Engineering whose contribution in stimulating suggestions and encouragement, guidance through the project.

Also, we would like to thank the non- teaching staff of the Computer Science and Engineering Department and friends for their co-operation and help.

-Pratham Vijay Vargiya

-Vigneshwar Ravindran

-Naman Arbind

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Leave Management System, in simple words, is managing the leave system of an organization. Organizations often require an expert who can plan, organize, analyse, interpret and make decisions regarding employee leave policies. Moreover, the HR specialist usually always performs these roles. However, the industry and market are beginning to witness a gradual change from manual leave management to digital leave management system adoption. This project shows up the simpler execution of how a large-scale software for digital leave management can be implemented.

1.2 Problem Definition

The manual tracking of leaves and allocation of leaves are possible up-to a certain limit but when an organization is expanding and many employees are hired in it which are in different departments then the manual tracking of the leaves can be a very hectic job and an organisation do not want to allocate 3-4 persons just to manage the leave. All of this becomes extremely time taking as the employee has to first write an application stating the reason along with number of days and then the senior needs to look into it. Also, the employees have to remember their count of different kinds of leaves remaining. Therefore the “Leave Management System” project was taken up by us in order make lives easy for both the employees and the organisation authority.

Some of the advantages of using a leave management system for an organization are:

1. It helps in streamlining leave procedures.

2. Employee leave information are managed centrally.

3. And consequently, applications and approvals are controlled effectively online, thereby reducing its dependency on paperwork.

1.3 Objectives and Scope

1) To automate the existing leave management system in the college.
2) To decrease the paperwork and enable the process with efficient reliable records maintenance by using centralized database, thereby reducing chances of data loss.
3) This project is very useful because it is automated and itself generate the status of the registered employee when approved and denied by the higher authority and thus reduce the responsibility' of manually taking attendance of each and every employee. Faculties and Hod can also check their remaining leaves Left.

1.4 Methodologies

Project management methodologies are a set of guiding principles and processes used plan, manage, and execute projects. The project management methodology you choose determines how work is prioritized and completed.

A web-based application is software package that can be accessed through the web browser. The software and database reside on a central server rather than being installed in desktop system and is accessed over the network. In computing, web application or web app is a client-server software application which the client (or user interface) runs in a web browser.

Therefore, the aim of this project is to build a web-based application that can be used in all devices. The proposed application has been simple yet effective for users that enhances their leave management process. Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser a client to update and maintain web applications. The users can readily access the number of leaves they can take and also the type of leave which is left and apply for the leave and a higher authority can accept or decline right away which makes the process efficient.

Chapter 2

System Analysis

2.1 Existing System and Its Limitations:

1) The existing system involves with faculties and HODs having to write out physical applications to apply for leaves. This process involves in spending a lot of time and can become tedious.

2) Also, the Faculties and HODs need to remember the numbers of their different kinds of leaves remaining.

3) HODs and Principal have to maintain a certain physical record to either reject or approve a leave request.

4) The dependency of physical appearance, of the one seeking leave and the one who is authorised to either approve or reject the request, might not be fullfiled at some times.

2.2 Proposed System and Its Advantages:

1) We propose a completely online system that tackles all the difficulties faced by faculties and HOD in the paperwork . There are 4 modules of the project: 1.Admin, 2.Faculty, 3.HOD, 4.Principal.

2) The Admin can register new Faculty, Hod and Principal.

3) A Faculty can check their remaining leaves by logging in to their account, Edit their profile, Apply for a leave from HOD of their department, Track the leave request status that they have applied for, and can simply logout.

4) The HOD can check their remaining leaves by logging in to their account, Edit their profile, Apply for a leave from the Principal, check the new leave requests from Faculties of their department and approve or reject them, can track the leave request status that they have applied for, and can simply logout.

5) The Principal can simply edit their profile, check the new leave request from HODs of any department and either accept or reject them, and can simply logout.

6) A Faculty do not need to visit the HOD in person to apply or check the status of their leave and same in case of HOD with principal.

7) The cumbersome job of managing all the paperwork become completely automated thus saving all the hassle.

2.3 System Requirements:

A Software Requirements Specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete Concept.

2.4 External Requirements:

System Requirements:

- Front End — HTML, CSS
- Back End — MongoDB, Node-JS,EJS,Express,Passport-JS

Software Requirements:

- Operating System — Windows Xp – Sp2 / 7 / 8.1 Above
- Browsers — Internet Explorer, Google Chrome

Hardware Requirements:

- Processor — Dual Core (2.0) Ghz Or Above
- RAM — 2 GB and Above
- HDD — 120GB and Above

Chapter 3

System Design

3.1 System Architecture:

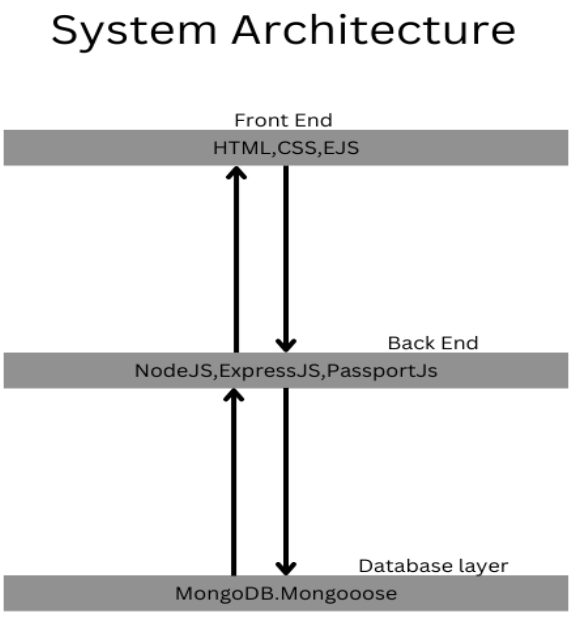
Architecture focuses on looking at a system as a combination of many different components, and how they interact with each other to produce the desired result. It involves the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a web application.

3.2 Architecture Diagram:

The above-mentioned Architecture Diagram consists of 3 Layers with Node-JS being the Server-Side Scripting tool that connects the Front End and Back End Layer.

- Front End Layer
- Back End Layer
- Database Layer

Fig 3.1: System architecture



1.Front End Layer

The front end of the application is developed using the HTML, CSS, EJS as shown in the fig3.1. These are the fundamentals behind the frontend part of our project. The UI is very user friendly so that every user of the application can work with the application easily. The login method is where the user manually inserts their username and password. The model first checks if the user is already authenticated using passport middleware. If not, then these details are verified by checking if the username exists in the database and if it exists it encrypts the entered password with bcrypt and checks whether this matches with the encrypted password already stored in the database.

In our Project there are Admin and 3 main types of users :

- HOD
- Principal
- Faculties

2. Back End Layer

The back end layer is managed effectively with the use of Node.js, Passport.js and Express.js as shown in fig 3.1. The Node-JS is used to interact between the user and the database. So that the data is stored effectively in the database and helps them to easily fetch and display the information from the database. Node modules are used to achieve some of the functionalities in our project which we availed using the Node Package Manager (NPM). Express-validator module is used to validate the entered details during registration of a new user. Passport.js is used so as to authenticate the users when required.

3. Database Layer

The MongoDB database is used for data storing purposes. The effective analysis is done to create the database design and to store the data. Mongoose is used to make connection with MongoDB database. The data is retrieved time to time as per the requirements as like when it is required to display the profile details of users or to store the leave request, as when applied by user and to keep track of accurate different types of leaves count, and also to change the status of the leave request when the higher authority either approve or deny the same.

3.3 ER DIAGRAM:

An entity-relationship diagram is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

In software Engineering, an Entity-relationship model(ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as relational database.

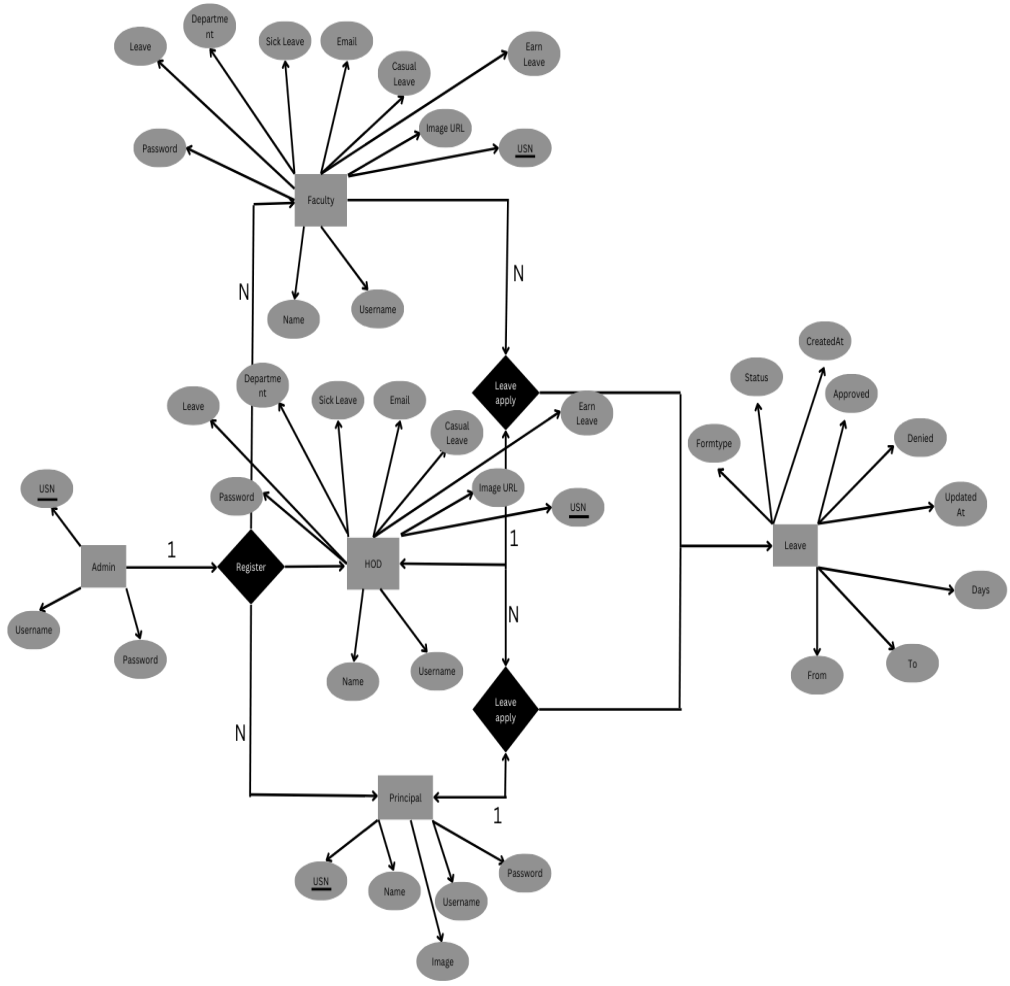


Fig 3.2:E-R Diagram

Chapter 4

System Implementation

4.1 Introduction

A crucial phase in the system lifecycle is the successful implementation of the new system design. Implementation simply means converting a new system design into operation. Coding is a stage in which the design is converted into working system. Implementation is the process of bringing the developed system into operational use and turning it to the user. This stage is considered to be the most crucial stage in the development of a successful system since a new system is developed and the users are given the confidence of its effectiveness.

Implementation Phases are as follows:

- First phase includes table design for Database module.
- Second phase includes coding for GUI modules
- Third phase includes the integration of modules
- Fourth phase includes connection establishment between front-end and back end
- Fifth phase includes error handling and message generator.

4.2 Implementation Approaches

The front end of the project was implemented using HTML, CSS and EJS. For the backend of our project we have used Node.js, Express.js, Passport.js. The proposed system is implemented with the help of MongoDB Database Management System along with its Object Data Modelling library, Mongoose. All the required tables and columns, considering the system design are executed and the basic structure is developed which serves as the back-end to the project. With the help of localhost server through app.js, guided by Node.js to provide for a website platform for Leave Management System is used.

4.3 Tools and its Explanation:

Front End:

HTML

HTML Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (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.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate css file, and reduce complexity and repetition in the structural content.

EJS

Embedded JavaScript (EJS) is a set of two open source libraries providing in-browser client side templates for web development and as a template system for node.js (including client-side template functionality). EJS uses <% %> or [% %] tags, executing any JavaScript within the tags. Adding an equals sign (<%= %>) causes the enclosed JavaScript to be evaluated, and the toString representation to be appended to the document.
The original client-side EJS has been subsumed into JavaScriptMVC, official support is offered at their forum (the old Google Groups group is no longer active).
Install via npm: npm install ejs.

Back End:

NODE-JS

Node.js (Node) is an open source, cross-platform runtime environment for executing JavaScript code. Node is used extensively for server-side programming, making it possible for developers to use JavaScript for client-side and server-side code without needing to learn an additional language. Node is sometimes referred to as a programming language or software development framework, but neither is true; it is strictly a JavaScript runtime.

Node incorporates the V8 JavaScript engine, the same one used in Google Chrome and other browsers. It is written in C++ and can run on macOS, Linux, Windows and other systems. The engine parses and executes JavaScript code. It can operate independently of a browser environment, either embedded in a C++ application or implemented as a standalone program.

Express-JS

Express.js is a small framework that works on top of Node.js web server functionality to simplify its APIs and add helpful new features. It makes it easier to organize your application's functionality with middleware and routing. It adds helpful utilities to Node.js HTTP objects and facilitates the rendering of dynamic HTTP objects.

Key Features:

- Develops Node.js web applications quickly and easily.
- It's simple to set up and personalise.
- Allows you to define application routes using HTTP methods and URLs.

Passport-JS

Passport is Express-compatible authentication middleware for Node.js.

Passport's sole purpose is to authenticate requests, which it does through an extensible set of plugins known as strategies. Passport does not mount routes or assume any particular database schema, which maximizes flexibility and allows application-level decisions to be made by the developer. The API is simple: you provide Passport a request to authenticate, and Passport provides hooks for controlling what occurs when authentication succeeds or fails.

Database: MongoDB

MongoDB, the most popular NoSQL database, is an open-source document-oriented database. The term 'NoSQL' means 'non-relational'. It means that MongoDB isn't based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data. This format of storage is called BSON (similar to JSON format).

SQL databases store data in tabular format. This data is stored in a predefined data model which is not very much flexible for today's real-world highly growing applications. Modern applications are more networked, social and interactive than ever. Applications are storing more and more data and are accessing it at higher rates.

Relational Database Management System (RDBMS) is not the correct choice when it comes to handling big data by the virtue of their design since they are not horizontally scalable. If the database runs on a single server, then it will reach a scaling limit. NoSQL databases are more scalable and provide superior performance. MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.

Chapter 5

System Testing

5.1 Introduction

Testing is the major process involved in software quality assurance (QA). It is iterative process. Here test data is prepared and is used to test the modules individually. System testing makes sure that all components of the system function properly as a unit by actually forcing the system to fail. Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. Thus, the system testing is a confirmation that all is correct and an opportunity to show the user that the system works. The final step involves Validation testing, which determines whether the software function as the user expected.

5.2 Purpose of Testing

Testing is done to analyse whether the application developed is according to the requirements. The main course of testing is to check for the existence of defects or errors in a program or project or product, based up on some predefined instructions or conditions. Following are some of important factors for which Testing for an application is required:

Different levels of testing are done like unit testing, integration testing in module are,

HOD:

- To guarantee the module ready to display the profile details of HOD and the leave requested by Faculties and the status of leave applied by HOD.
- To guarantee the module ready to apply for the respective leaves such as casual, sick and earn leave.
- Direct execute Home page.
- To guarantee that the module able to update the current leave status.
- To guarantee to obtain proper status of faculties leaves.

- To ensure that Hod can easily approve or deny the leave of the faculties.
- To be able to update their profile details.

Principal:

- To guarantee the module ready to view the leave request of the Hod.
- To guarantee the module ready to approve and deny the leave of Hod.
- To be able to update their profile details.

Faculty:

- To guarantee the module ready to apply for the leave.
- To guarantee the module can track the current status of the leave
- To guarantee the module can display the profile details.
- To guarantee the module can edit the profile details.
- To guarantee the module keeps the accurate count of remaining leaves.

Steps involved in system testing:

1. **Test Environment Setup**: Create testing environment for the better-quality testing.
2. **Create Test Case**: Generate test case for the testing process.
3. **Create Test Data**: Generate the data that is to be tested.
4. **Execute Test Case**: After the generation of the test case and the test data, test cases are executed.
5. **Defect Reporting**: Defects in the system are detected.
6. **Regression Testing**: It is carried out to test the side effects of the testing process.
7. **Log Defects**: Defects are fixed in this step.
8. **Retest**: If the test is not successful then again test is performed.

Sample Test Cases:

Test Case ID	Test Cases	Expected Output	Actual Output	Result
TC1	Execute / Run the application	Application should run without any interrupts and errors	Application executed properly	Pass
TC2	Home page loading	Home page should load properly and all contents should be displayed.	Home page and its contents loaded properly	Pass
TC3	Admin Login page loading	Login page of admin should be loaded properly	Admin login page executed properly	Pass
TC4	Faculty Login Page loading	Login page of faculty should be loaded properly	Faculty login page executed properly	Pass
TC5	HOD Login Page loading	Login page of HOD should be loaded properly	HOD login page executed properly	Pass
TC6	Admin login	Case 1: Admin entered credentials matches with database – Loading of Admin Home page (Registration page) Case 2: Admin entered credentials does not match with database – Credentials does not match pop up	Case 1: Admin Home page (Registration page) executed properly Case 2: Error message popped up for wrong credentials	Pass
TC7	Faculty login	Case 1: Faculty entered credentials matches with database – Loading of faculty home page Case 2: Faculty entered credentials does not match with database – Credentials does not match pop up	Case 1: Faculty Home page executed properly Case 2: Error message popped up for wrong credentials	Pass
TC8	HOD login	Case 1: HOD entered credentials matches with database – Loading of hod home page Case 2: HOD entered credentials does not match with database – Credentials does not match pop up	Case 1: HOD Home page executed properly Case 2: Error message popped up for wrong credentials	Pass

TC9	Principal login	Case 1: Principal entered credentials matches with database – Loading of principal home page Case 2: Principal entered credentials does not match with database – Credentials does not match pop up	Case 1: Principal Home page executed properly Case 2: Error message popped up for wrong credentials	Pass
T10	Faculty Profile Edit	Faculty should be able to edit their profile basic details	Basic profile details edited successfully	Pass
T11	HOD Profile Edit	HOD should be able to edit their profile basic details	Basic profile details edited successfully	Pass
T12	Principal Profile Edit	Principal should be able to edit their profile basic details	Basic profile details edited successfully	Pass
T13	Faculty Leave Apply	Faculty should be able to apply for a leave to HOD of their department	Leave is applied for the specified type successfully	Pass
T14	HOD Leave Apply	HOD should be able to apply for a leave to the Principal	Leave is applied for the specified type successfully	Pass
T15	Tracking of Faculty applied leave request	Faculty should be able to track the applied leave status	The status, pending or approved or denied, is visible	Pass
T16	Tracking of HOD applied leave request	HOD should be able to track the applied leave status	The status, pending or approved or denied, is visible	Pass
T17	Approving or denying of Faculty leave request by HOD	HOD should be able to either approve or deny the leave requested by Faculty, by checking the details of leave.	The Approval or Denial is reflected successfully to the faculty.	Pass
T18	Approving or denying of HOD leave request by Principal	Principal should be able to either approve or deny the leave requested by HOD, by checking the details of leave.	The Approval or Denial is reflected successfully to the HOD.	Pass
T19	Logout	Faculty, HOD and Principal all should be able to successfully logout without any errors.	Faculty, HOD and Principal all are able to logout from their accounts successfully.	Pass

Chapter 6

Results

6.1 Screenshots

Fig 6.1 About us

Fig 6.2 Leave Portal Home Page

Fig 6.3 Admin Login Page

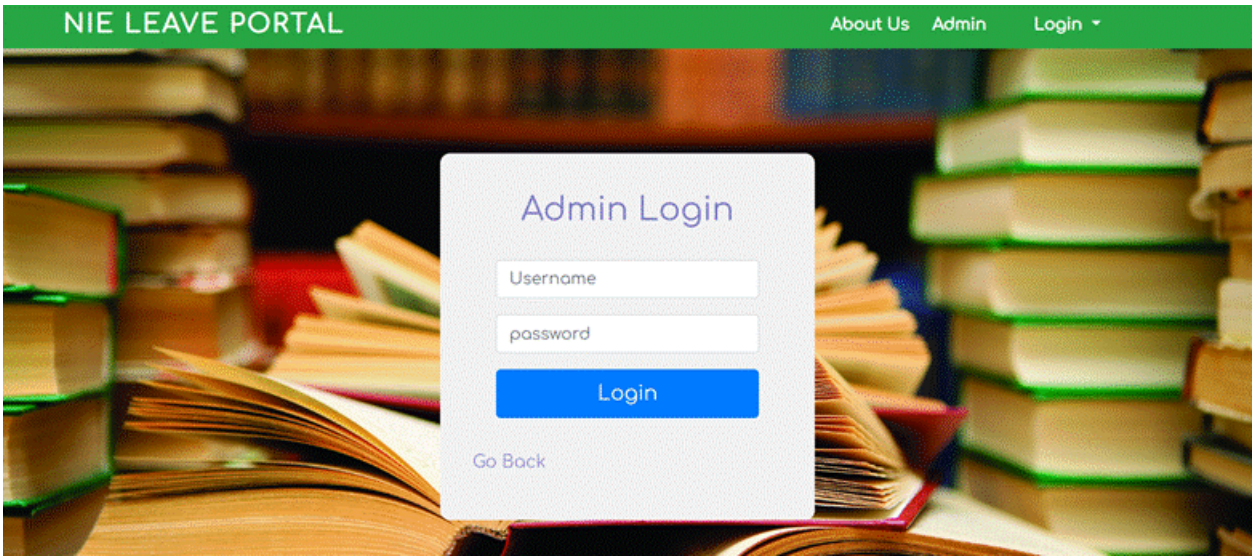


Fig 6.4 Registration page for new member

Fig 6.5 Registration page details Page require for Hod and Faculty

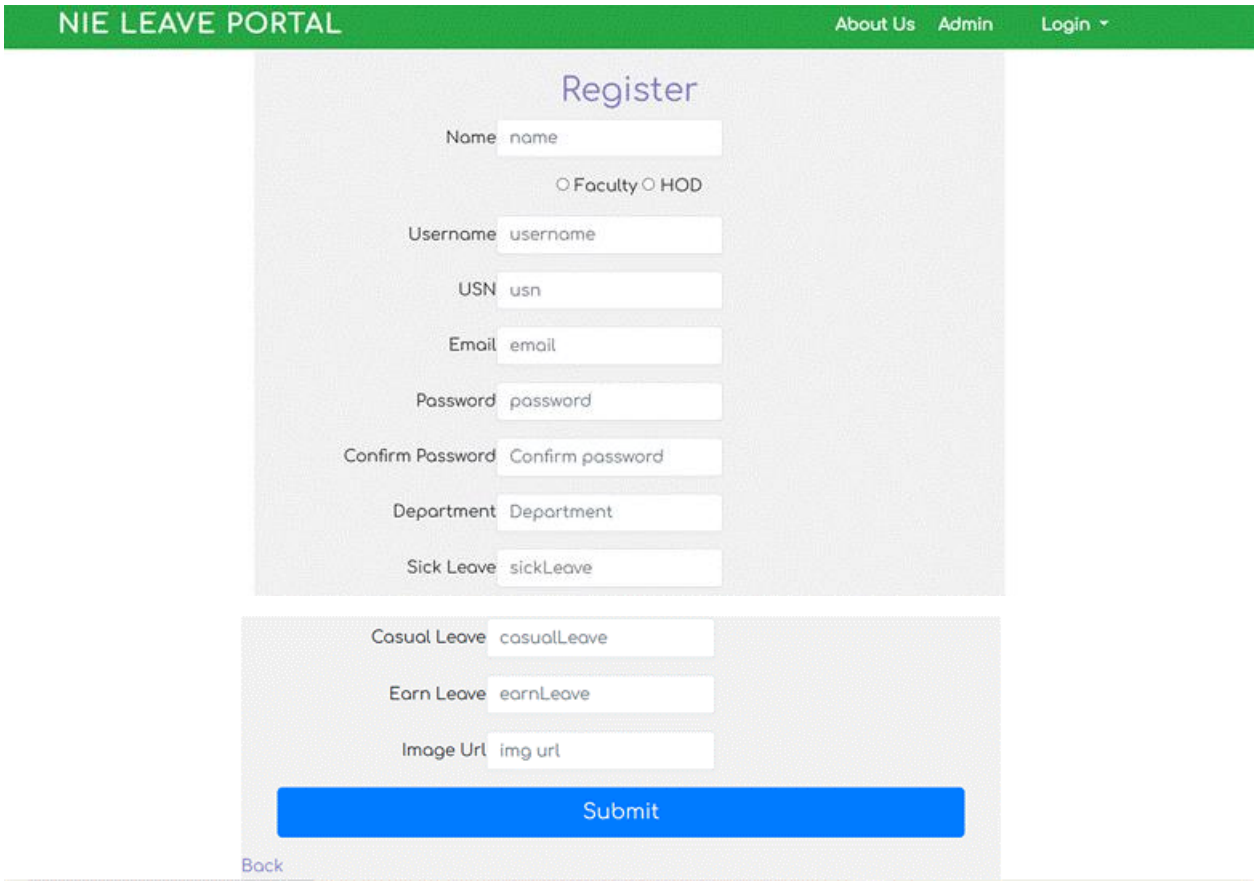


Fig 6.6 Principal Registration Page Details

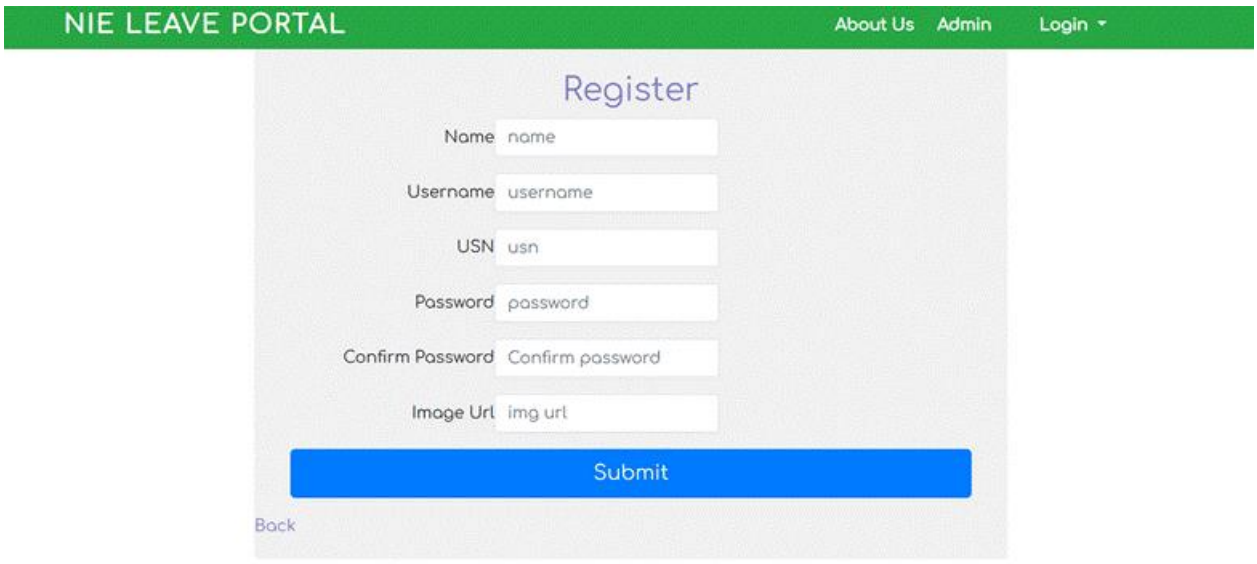


Fig 6.7 Faculty login page

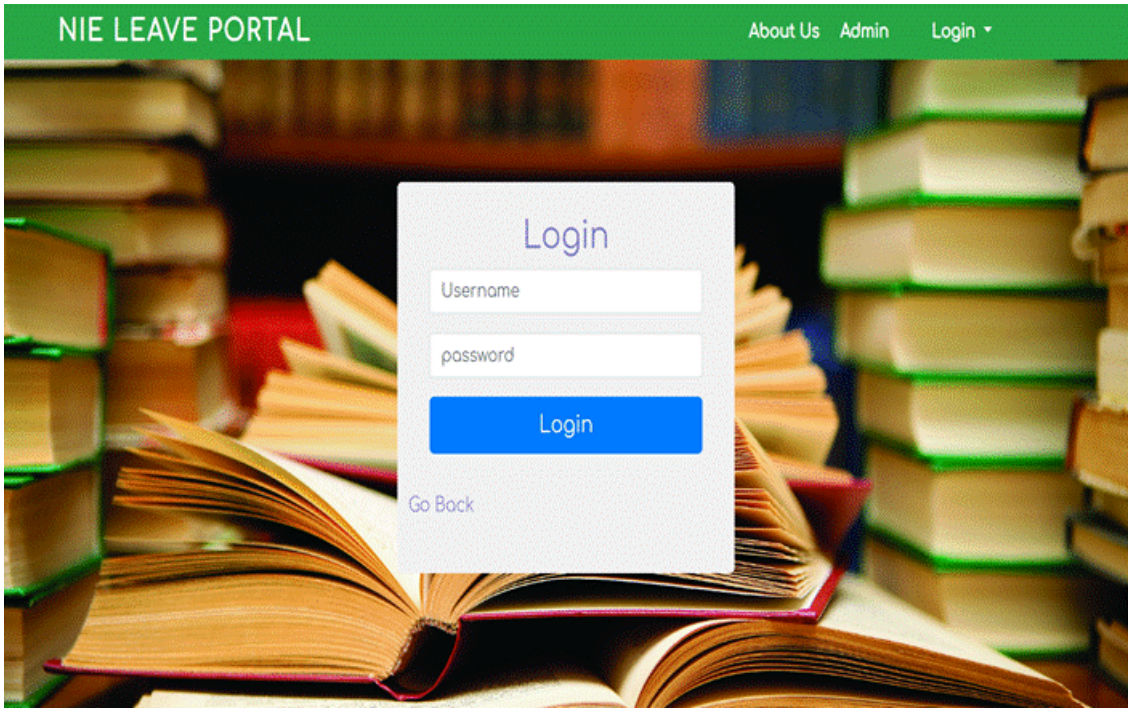


Fig 6.8 Faculty Home Page

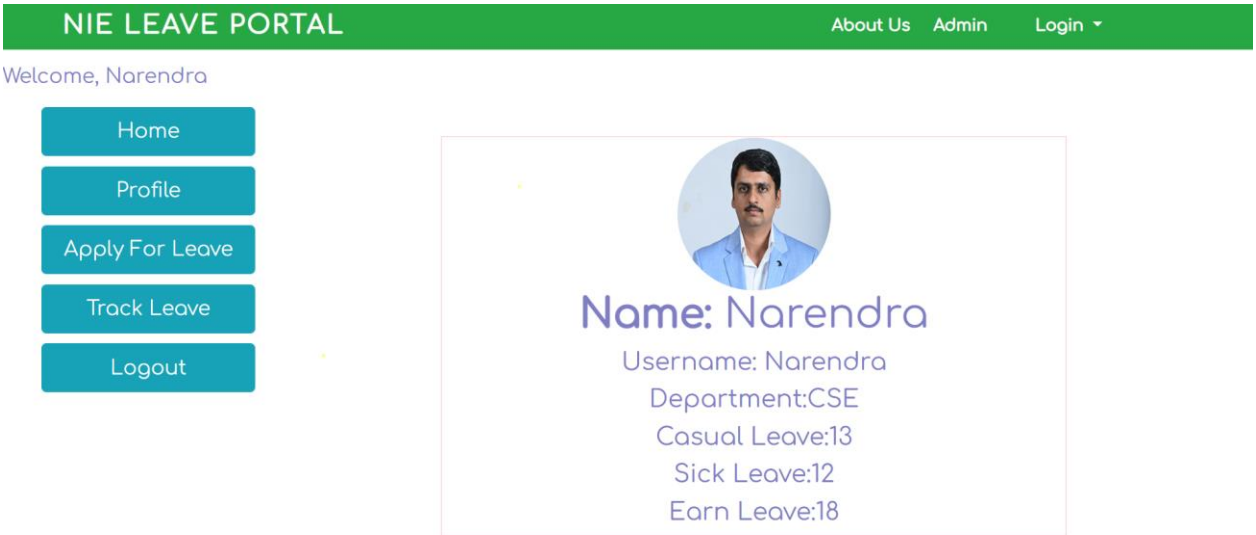


Fig 6.9 Faculty Profile

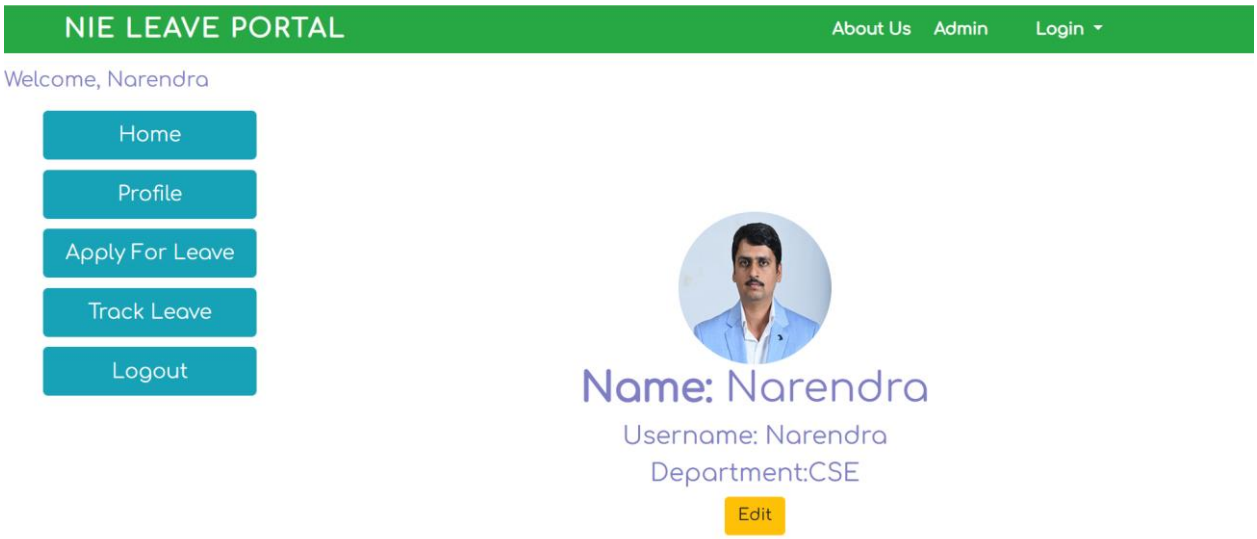


Fig 6.10 Faculty Apply for leave page

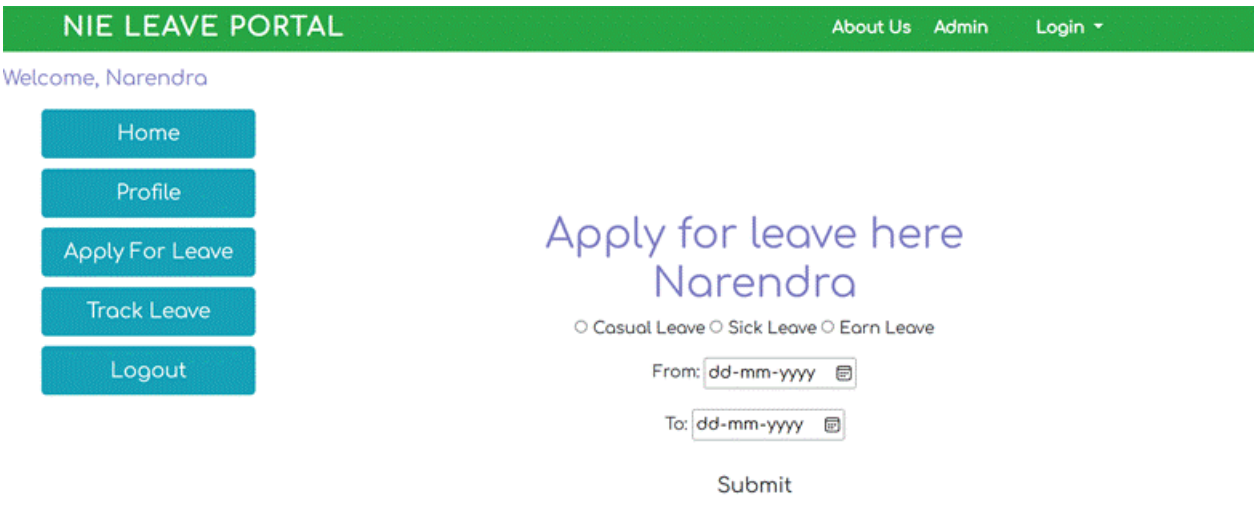


Fig 6.11 Track Leave page for Faculty

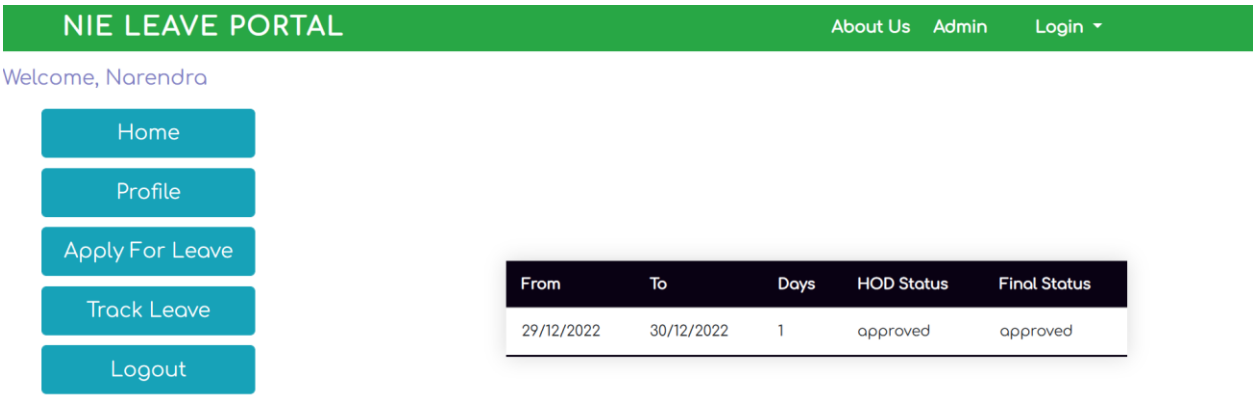


Fig 6.12 HOD login Page

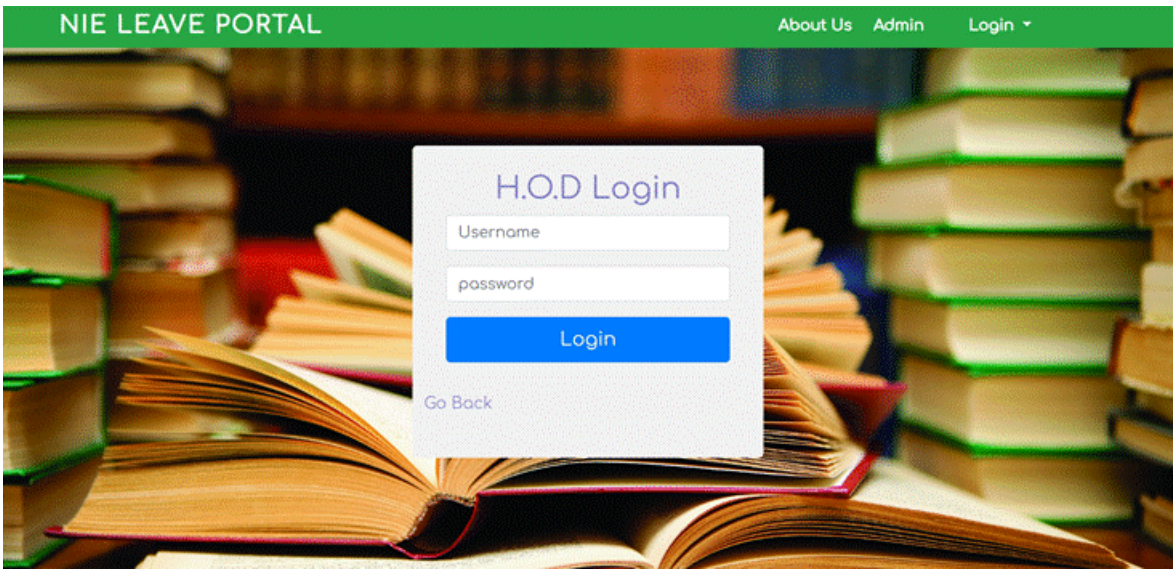


Fig 6.13 Hod Home page

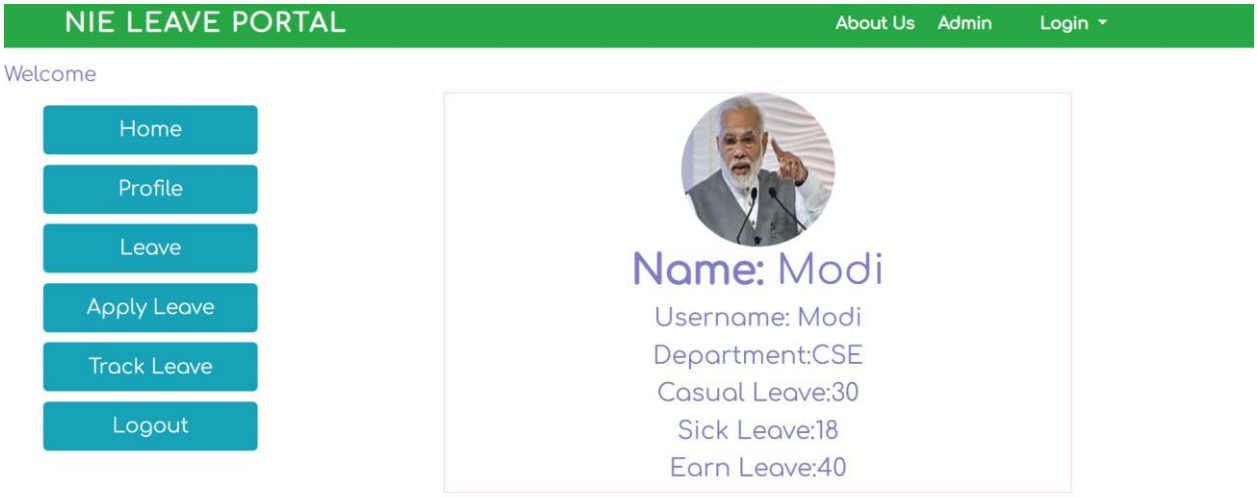


Fig 6.14 Hod Profile

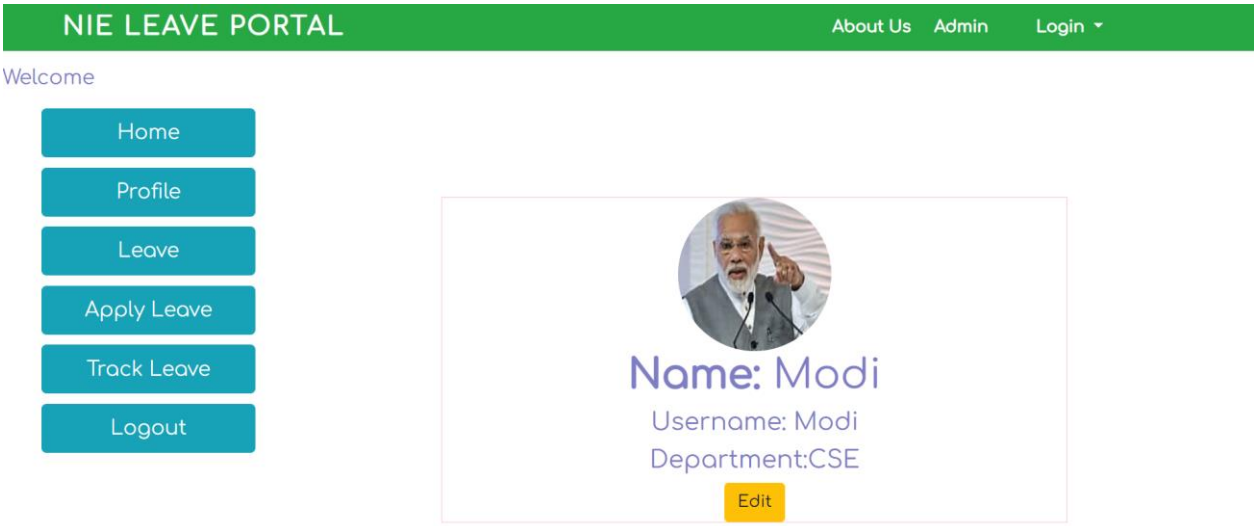


Fig 6.15 Hod Leave Page where faculties request are tracked

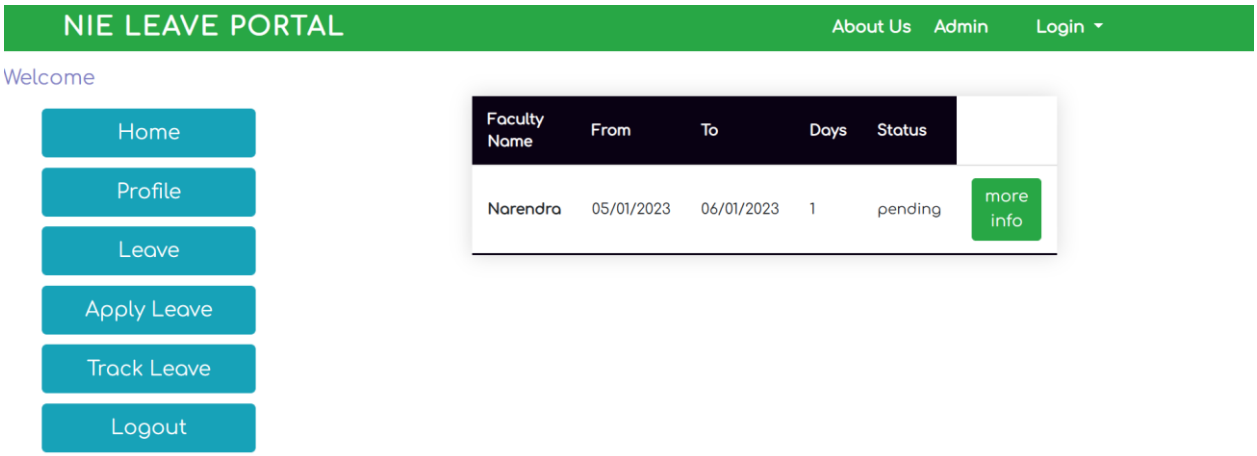


Fig 6.16 Hod Apply leave page

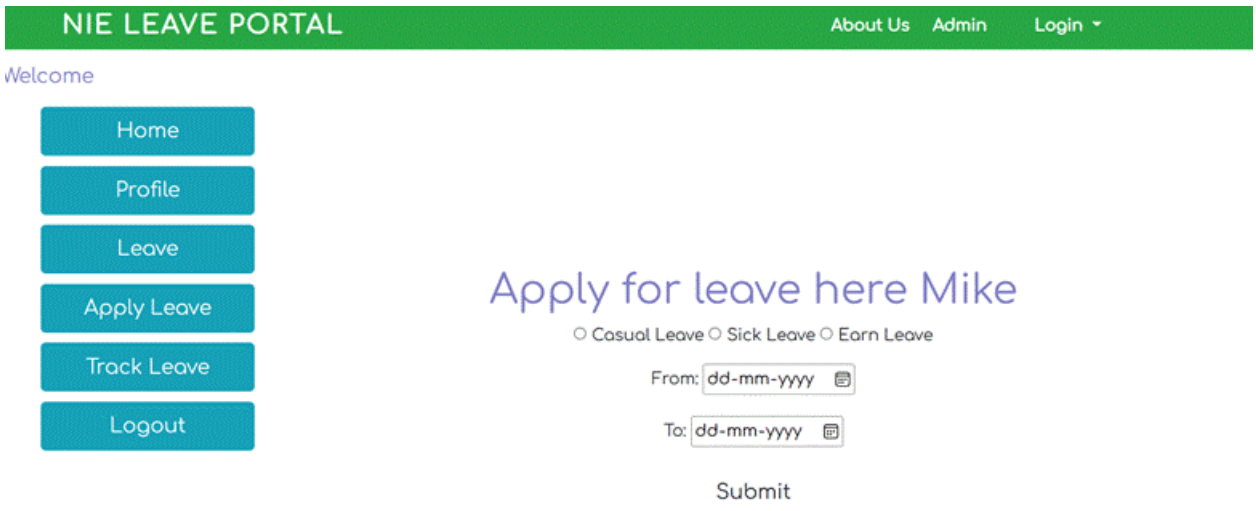


Fig 6.17 Track leave Page where Hod can track his/her leave

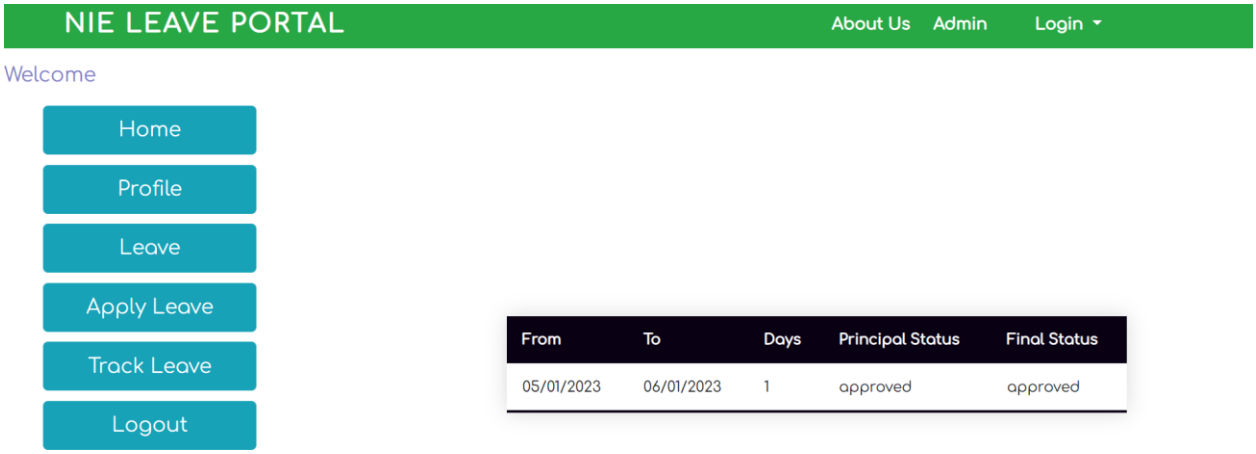


Fig 6.18 Principal Login Page

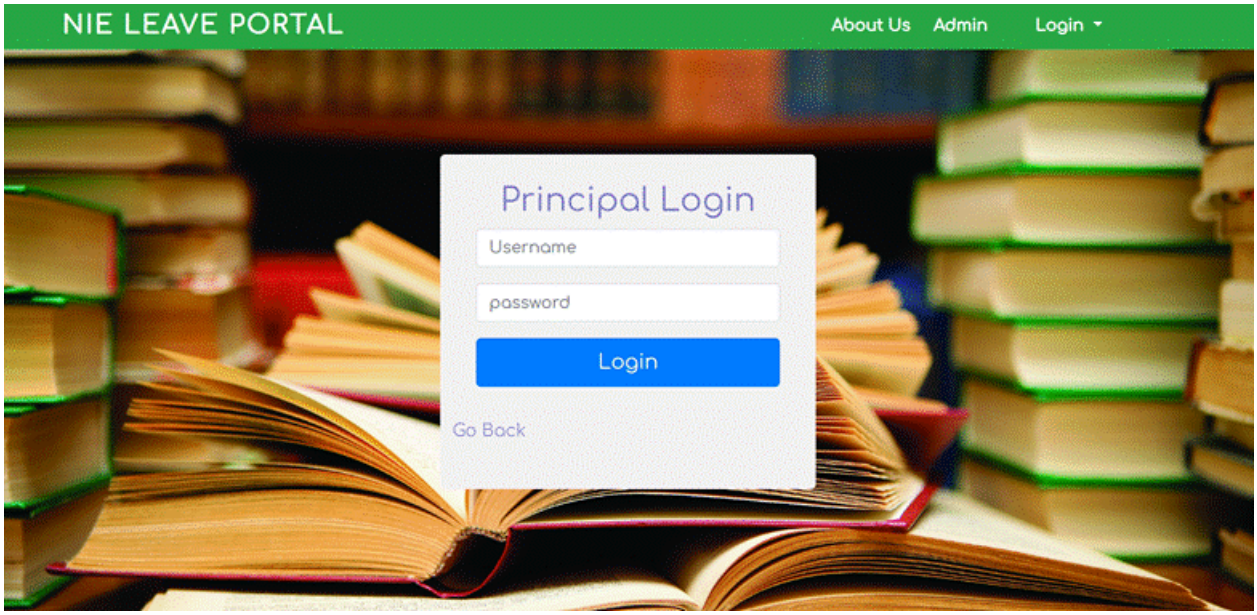


Fig 6.19 Principal Home Page

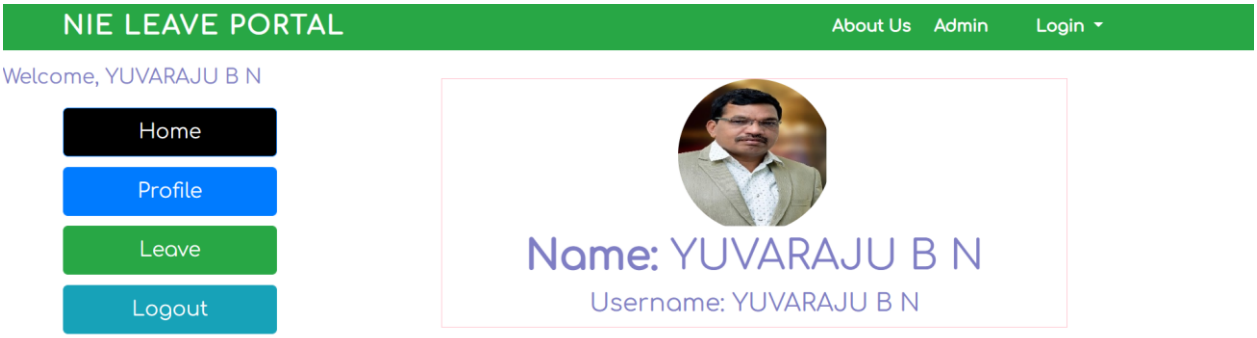


Fig 6.20 Principal Profile

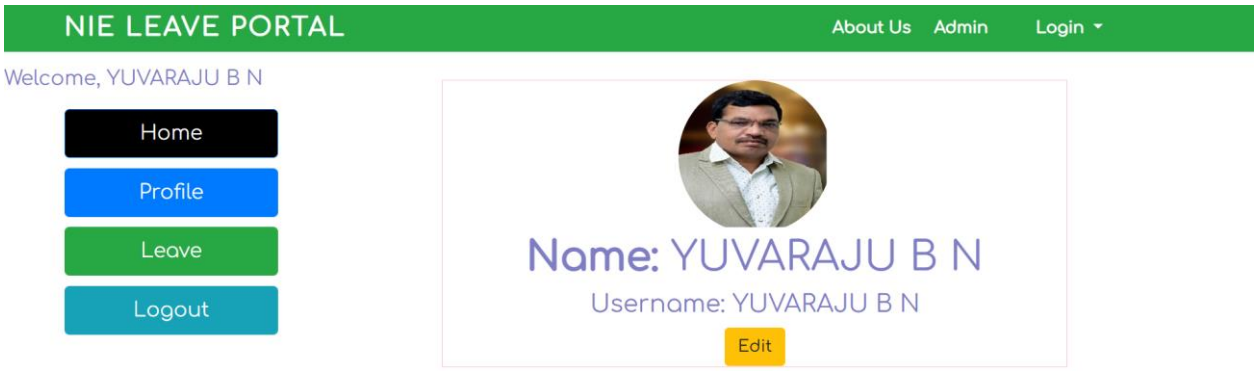
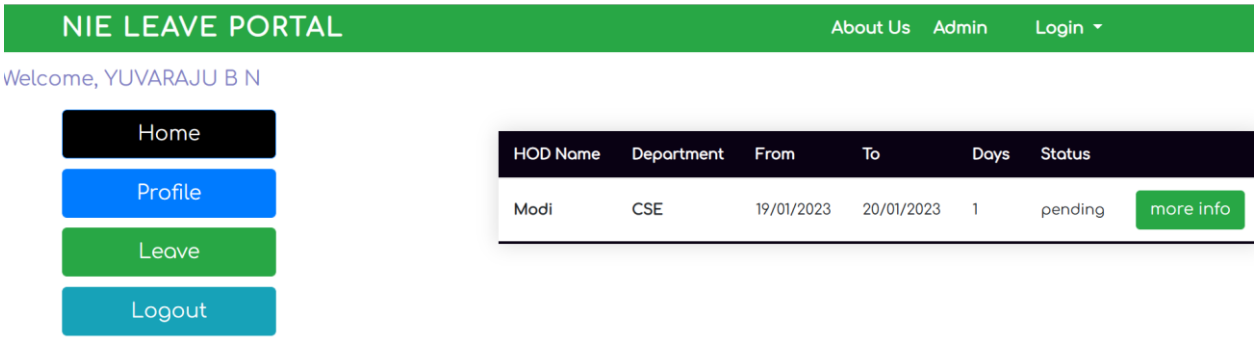


Fig 6.21 Principal Leave Page for tracking Hod Leave status



6.2 Conclusion

The “Leave Management System” web application provides the ease gateway through which the faculties and hod can apply for leaves to the concerned authority with ease and the concerned authority can approve/deny the leaves and also all the faculties and hod can obtain the necessary information about the leaves and type of the leave left. A web-based application is software package that can be accessed through the web browser. The software and database reside on a central server rather than being installed in desktop system and is accessed over the network. The application can also be used for the management of the remaining leaves and plan them to the best of their comfort.

Therefore, the aim of this project is to build a web-based application that can be used in all devices. The proposed application has been simple yet effective for faculties and hods that enhances their leave management experiences. The faculties and higher authority both can be benefitted from these features as they can access all the necessary information without much hassle and can apply for leaves from anywhere rather than physically visiting the higher authority’s cabin for the approval.

Further Enhancements:

- The timetable of the given day of leave seeking faculties can be added with the request of their leaves so that the HOD or principal can take the decision for the best benefits of faculties and students as well.

- Taking the platform to the next level by adding the support to host leave management systems of multiple organisations so that the application can be checked at a large scale of users and end problems could be dealt with.

- A mobile application can develope enhancing the website making it even simpler and easily accessible to students.

6.3 References

- HTML and CSS Concepts : https://developer.mozilla.org/en-US/docs/Web
- Node-JS Concepts : https://www.geeksforgeeks.org/nodejs/
- Express-validator module: https://www.npmjs.com/package/express-validator
- Authentication middleware module: https://www.npmjs.com/package/passport
- MongoDB Concepts : https://www.geeksforgeeks.org/mongodb-tutorial/
- EJS concepts: https://www.geeksforgeeks.org/use-ejs-as-template-engine-in-node-js/