

# **USER DATA PLATFORM**

*A*

*Project Report*

*Submitted in partial fulfillment of the  
Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**By**

**Hemanth Sherla, Pradeep Kumar Reddy Peddi**

**1602-18-737-072, 1602-18-737-088**

*Under the guidance of*

**Dr.B.Kezia Rani**

**Associate Professor**



**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**ACCREDITED BY NAAC WITH 'A++' GRADE**

**(Affiliated to Osmania University)**

**Ibrahim Bagh, Hyderabad-31**

**2022**

**Vasavi College of Engineering**  
**(Autonomous)**  
**ACCREDITED BY NAAC WITH 'A++' GRADE**  
**(Affiliated to Osmania University)**  
**Hyderabad-500 031**



### **DECLARATION BY THE CANDIDATE**

I, **Hemanth Sherla, Pradeep Kumar Reddy Peddi** bearing hall ticket number, **1602-18-737-072, 1602-18-737-088**, hereby declare that the project report entitled **User Data Platform** under the guidance of **Dr.B. Kezia Rani**, Associate Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering in Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**Hemanth Sherla**  
**1602-18-737-072,**

**Pradeep Kumar Reddy Peddi**  
**1602-18-737-088**

**Vasavi College of Engineering  
(Autonomous)**

**ACCREDITED BY NAAC WITH 'A++' GRADE**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**



**BONAFIDE CERTIFICATE**

This is to certify that the project entitled **User Data Platform** is being submitted by **Hemanth Sherla, Pradeep Kumar Reddy Peddi** bearing **1602-18-737-072, 1602-18-737-088** in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by him/her under my guidance.

**Dr. B.Kezia Rani**

**Associate Professor**

**Internal Guide**

**Dr. K. RamMohan Rao**

**HOD, IT**

**External examiner**

## **ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them. We would like to take the opportunity to express our humble gratitude to **Dr.B. Kezia Rani**, Associate Professor under whom we executed this project. We are grateful to her guidance, and constructive suggestions that helped us in the preparation of this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

I would like to thank all faculty members and staff of the Department of Information Technology and our external examiner for their generous help in various ways for the completion of this project.

Finally, yet importantly, we would like to express our heartfelt thanks to our HOD **Dr. K. Ram Mohan Rao**.



**Offer for Employment**

Date: 12-01-2022

To,  
Mr/Ms. Hemanth Sherla

Dear Hemanth,

Sub: Letter of Offer for Employment

We are pleased to make you an offer as Intern in our organization, Darwinbox Digital Solutions Private Limited (a private limited company incorporated as per Company's Act, 2013 with CIN:U74900TG2015PTC101793) with effect from 14-02-2022 and Hyderabad, Telangana, India as work location.

Your employment with us will be governed by the Terms & Conditions as detailed in Annexure –B as well as Annexure C: PROPRIETARY INFORMATION AND INTELLECTUAL PROPERTY ASSIGNMENT AGREEMENT. Your offer has been made based on information furnished by you. However, if there is a discrepancy in the copies of documents or certificates given by you as proof of the above we retain the right to review our offer of employment. Employment as per this offer is subject to your being medically fit.

We congratulate you on your appointment and wish you a long and successful career with us. We are confident that your contribution will take us further in our journey towards becoming world leaders. We assure you of our support for your professional development and growth.

Yours truly,  
For Darwinbox Digital Solutions Private Limited  
*PSN Chaitanya*  
Chaitanya Peddi  
Director.

With the signature below, I accept this offer for employment.

Name: Hemanth Sherla  
Date:



Offer for Employment

Date: 08-02-2022

To,  
Mr/Ms. Pradeep Peddi

Dear Pradeep,

Sub: Letter of Offer for Employment

We are pleased to make you an offer as Intern in our organization, Darwinbox Digital Solutions Private Limited (a private limited company incorporated as per Company's Act, 2013 with CIN:U74900TG2015PTC101793) with effect from 02-03-2022 and Hyderabad, Telangana, India as work location.

Your employment with us will be governed by the Terms & Conditions as detailed in Annexure -B as well as Annexure C: PROPRIETARY INFORMATION AND INTELLECTUAL PROPERTY ASSIGNMENT AGREEMENT. Your offer has been made based on information furnished by you. However, if there is a discrepancy in the copies of documents or certificates given by you as proof of the above we retain the right to review our offer of employment. Employment as per this offer is subject to your being medically fit.

We congratulate you on your appointment and wish you a long and successful career with us. We are confident that your contribution will take us further in our journey towards becoming world leaders. We assure you of our support for your professional development and growth.

Yours truly,  
For Darwinbox Digital Solutions Private Limited

*PSN Chaitanya*

Chaitanya Peddi  
Director.

With the signature below, I accept this offer for employment.

## **ABSTRACT**

Human Resource management is the one of the most important part of any organization. It uses few strategic approaches to the effective and efficient management of people in a company or organization such that they help their business gain a competitive advantage.

To provide a better and value-added service or outcome to customer or client of the organization. That is to make a human resource department more effective and efficient new technologies are now being introduced on a regular basis so make things much simpler and easier. One of the latest human resource technologies Darwin box [2] which is a new-age, enterprise-ready, HCM platform that enable's enterprises to automate day-to-day HR processes, simplifies human interactions, delivers actionable insights to build better workplaces.

As an organization grows the employee data also grows exponentially, this will be a challenge for storing, processing, and analyzing a large volume of data.so we use User Data Analytics Platform. Since it has lots of data, various searches have to be performed and run analytics along with the needs, to see how employees are performing from various metrics, which require data management. It plays a vital role, the plan is to leverage performance of MongoDB and Express.User Data Analytics Platform is a system designed to help provide information used in HR decision making such as employee details, payroll, recruiting, training, performance analysis. The platform will enable the employee and organization to track employee details, attendance, salary, leave management, and other data. It is a user interaction platform. Overall, it manages all the employees in an organization.

## TABLE OF CONTENTS

<b>ABSTRACT.....</b>	<b>i</b>
<b>LIST OF FIGURES.....</b>	<b>iii</b>
<b>1.INTRODUCTION.....</b>	<b>1</b>
1.1 OVERVIEW.....	1
1.2 OBJECTIVE.....	2
1.3 PROPOSED METHOD.....	3
<b>2.LITERATURE SURVEY.....</b>	<b>4</b>
<b>3.SYSTEM REQUIREMENTS SPECIFICATION.....</b>	<b>5</b>
3.1.SOFTWARE REQUIREMENTS.....	5
3.1.1.NODE JS.....	5
3.1.2.JAVA SCRIPT.....	6
3.1.3.HTML, CSS & BOOTSTRAP.....	7
3.1.4.EXPRESS JS.....	7
3.1.5.MONGODB DATABASE.....	8
3.1.6.AMAZON WEB SERVICES.....	8
3.1.7.NPMPACKAGES.....	9
3.2.HARDWARE REQUIREMENTS.....	9
<b>4. METHODOLOGY.....</b>	<b>10</b>
4.1. PROJECTFLOW EXPLANATION.....	10
4.1.1.USECASE DIAGRAM.....	11
4.2. FRONTEND.....	12
4.3.BACKEND.....	12
<b>5. RESULTS AND DISCUSSIONS.....</b>	<b>13</b>
<b>6. CONCLUSION AND FUTURE SCOPE.....</b>	<b>28</b>
<b>REFERENCES.....</b>	<b>29</b>
<b>APPENDIX.....</b>	<b>30</b>

## LIST OF FIGURES

<b>Figure 1:</b> Architecture & Project flow.....	10
<b>Figure 2:</b> Use Case Diagram.....	11
<b>Figure 3:</b> Entity Relationship model.....	12&13
<b>Figure 4:</b> Login page.....	13
<b>Figure 4(a):</b> Register page.....	14
<b>Figure 5:</b> Dashboard .....	15
<b>Figure 6:</b> About tab.....	15
<b>Figure 7:</b> Analytics tab.....	16
<b>Figure 8:</b> employee details.....	17
<b>Figure 9:</b> Performance tab.....	17
<b>Figure 10:</b> Managers tab.....	18
<b>Figure 11:</b> Departments tab.....	18
<b>Figure 12:</b> 1 vs 1 .....	19
<b>Figure 13:</b> Analysis with tfjs.....	20
<b>Figure 14:</b> Apply leave.....	21
<b>Figure 15:</b> Attendance.....	21
<b>Figure 16:</b> Favorites tab showing favorite videos.....	22
<b>Figure 17:</b> Employee leave.....	23
<b>Figure 18:</b> Clock tab.....	23
<b>Figure 19:</b> Employee Attendance.....	24
<b>Figure 20:</b> Change password.....	25
<b>Figure 21:</b> Amazon web services s3 storage .....	25
<b>Figure 22:</b> Mongodb atlas/compass.....	25

# **1. INTRODUCTION**

User Data Platform is a system designed to help provide information used in HR decision making such as employee details, payroll, recruiting, training, performance analysis. The platform will enable the employee and organisation to track employee details, attendance, salary, leave management, and other data. It is a user interaction platform. Overall it manages all the employees data in an organization, for this Database management, the plan is to leverage performance of MongoDB and Express and small part of tensorflowjs module.

## **1.1 Overview**

Node js Environment, Express js framework and Mongodb database is used for our website.

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop.

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications.

MongoDB is a cross-platform, document-oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

MongoDB is based on a NoSQL database that is used for storing data in a key-value pair. Its working is based on the concept of document and collection. It is also an open-source, a document-oriented, cross-platform database system that is written using C++.

Document-Oriented data storage, i.e., data, is stored in a JSON style format, which increases the readability of data as well. Replication and high availability of data. MongoDB provides Auto-sharding. Ad hoc queries are supported by MongoDB, which helps in searching by range queries, field, or using regex terms. Indexing of values can be used to create and improve the overall search performance in MongoDB. MongoDB allows any field to be indexed within a document.

MongoDB has a rich collection of queries. Updating of data can be done at a faster pace. It can be integrated with other popular programming languages also to handle structured as well as unstructured data within various types of applications.

MongoDB offers both local and cloud-hosted deployment options like compass and atlas we used both. Mongo dB Atlas is a hosted MongoDB Enterprise service option in the cloud which requires no installation overhead and offers a free tier to get started

## **1.2 Objective**

- The aim of the project is to help provide information used in HR decision making such as employee details, payroll, recruiting, training, performance analysis. The platform will enable the employee and organisation to track employee details, attendance, salary, leave management, and other data. It is a user interaction platform. Overall it manages all the employees in an organisation.
- Admin can be able to manage the employees and he is able to analyse the employees performance and managers and etc.
- As an organisation grows the employee data also grows exponentially, this will be a challenge for storing, processing, and analysing a large volume of data.so we use User Data Analytics Platform.
- Automate day-to-day manual efforts and help the companies and people
- Contribute to existing data analytics platforms and present various graphs and analysis about the employees and to gain knowledge about them

### **1.3 Proposed Method**

The website has two personas, i.e., Admin and employee, a different login mechanism is in place for each of them. The admin has all the access to every route and he has the ability to do all the tasks and to analyse the employee.

The employee can login and can only check about themselves and they can perform few actions on their account such as clock in clock out where he can provide a proof of work. And he can apply for the leave and can manage the attendance and can change the information about him.

While the admin has the abilities of the employee and has access to the analysis page he can analyse the employee and can also view the managers and their employees performance. With the help of the employee data we can analyse the data with mongodb aggregations and tensorflow model present in the npm modules.

Our User Data Analytics Platform is a system designed to help provide information used in HR decision making such as employee details, payroll, recruiting, training, performance analysis. The platform will enable the employee and organisation to track employee details, attendance, salary, leave management, and other data. It is a user interaction platform. Overall, it manages all the employees in an organisation.

## 2. LITERATURE SURVEY

This survey tells us about the previous HRMS software's and their drawbacks and the technologies they used in the previous software, by this we can get a gist of the previous HRMS software.

HRMS [1] software is a platform that integrates, manages, and automates many of the routine HR processes in an organization. It can perform a variety of functions including organizing and managing employee details and organizational information of a business, making it easily accessible.

As popularity for HRMS software has grown, there are many different systems on the market including Human Capital Management (HCM) and Human Resource Information System (HRIS), each offering varying features, and functionalities, at different price points.

We had a previous model of this HRMS software in php, yii framework 1.1, with mongo compass and local storage for data and image transfer, also used gii for model, crud and controller generator.

It is a bit slow than Node JS and it waits until the before step is fully executed as it is not an asynchronous run time environment.

The Ajax features are not well-drafted and require the help of JavaScript whenever needed. It does not provide AR queries or building up of multiple relations. The dependency on static methods is what make the developers to go against the framework.

Some of the limitations of HRMS are recent origin, lack of support of top management, improper actualisation, inadequate development program and inadequate information.

## **3.SYSTEM REQUIREMENTS SPECIFICATION**

A System Requirements Specification is a structured collection of information that embodies the requirements of a project. This section explains about both Software and Hardware requirements which are required for the completion of the project.

### **3.1 Software requirements**

Software requirements establish the agreement between your team and the customer on what the application is supposed to do. Without a description of what features will be included and details on how the features will work, the users of the software can't determine if the software will meet their needs. The key software requirements required for the project are:

- NODE JS
- JAVA SCRIPT
- HTML, CSS, BOOTSTRAP
- EXPRESS JS
- MONGODB
- AWS
- NODE PACKAGE MANAGERS MODULES

#### **3.1.1 NODE JS**

Node.js [3] is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

NodeJS is not a framework and it's not a programming language. Most people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as PayPal, Uber, Netflix, Walmart, and so on.

### 3.1.2 Javascript

JavaScript (JS) is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on user's devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

AJAX calls, which stand for Asynchronous JavaScript And XML, are becoming more popular as web applications look to bring in personalized

items dynamically. AJAX calls are one method to load personalized content separately from the rest of the HTML document, which allows for the full HTML document to be cached, improving back end load time. AJAX calls have a wide range of applications but are particularly useful for websites who have a large amount of personal information which can prevent full HTML document caching. Ecommerce websites are among those that can use AJAX calls to load cart contents, account information, and product recommendations without embedding that information directly into the HTML of a page. Platforms such as Magento have embraced AJAX calls, with Magento 2 utilizing AJAX for all personalized information. At Section we regularly use AJAX calls combined with Varnish Cache to allow for the caching of dynamic content without sharing personalized information between users.

### **3.1.3 HTML, CSS, and Bootstrap**

The Hypertext Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

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. CSS is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. Bootstrap is a free and open-source CSS framework directed at responsive, mobile- first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

### **3.1.4 EXPRESS JS**

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It

facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework –

- Allows to set up middleware's to respond to HTTP Requests.
- Defines a routing table which is used to perform different actions based on HTTP Method and URL.
- Allows to dynamically render HTML Pages based on passing arguments to templates.

### **3.1.5 MONGODB (NoSql Database)**

No NoSQL systems are also sometimes called Not only SQL to emphasize that they may support SQL-like query languages or sit alongside SQL databases in polyglot-persistent architectures. SQL (originally referring to "non-SQL" or "non-relational") database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

MongoDB [4] is a cross-platform, document-oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

### **3.1.6 AWS**

**AWS Storage Services:** AWS[5] offers a wide range of storage services that can be provisioned depending on your project requirements and use case. AWS storage services have different provisions for highly confidential data, frequently accessed data, and the not so frequently accessed data. You can choose from various storage types namely, *object storage*, *file storage*, *block storage services*, *backups*, and *data migration* options. All of which fall under the AWS Storage Services list.

**AWS Simple Storage Service (S3):** From the aforementioned list, S3, is the object storage service provided by AWS. It is probably the most commonly used, go-to storage service for AWS users given the features like extremely high availability, security, and simple connection to other AWS Services. AWS S3 can be used by people with all kinds of use cases like mobile/web applications, big data, machine learning and many more.

### 3.1.7 NPM MODULES

Npm[6] is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently.

npm is the world's largest software registry. Open source developers from every continent use npm to share and borrow packages, and many organizations use npm to manage private development as well.

npm consists of three distinct components:

- the website
- the Command Line Interface (CLI)
- the registry

It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs. We have used many modules like Multer, TensorFlow's, node-mailer, express-generator, bcrypt, alert, etc

### 3.2 Hardware requirements:

1. AWS and Atlas Cloud based Hardware Setup and Configuration
2. 4gb RAM and above
3. Intel i3 and above
4. Linux or Mac-Os or Windows

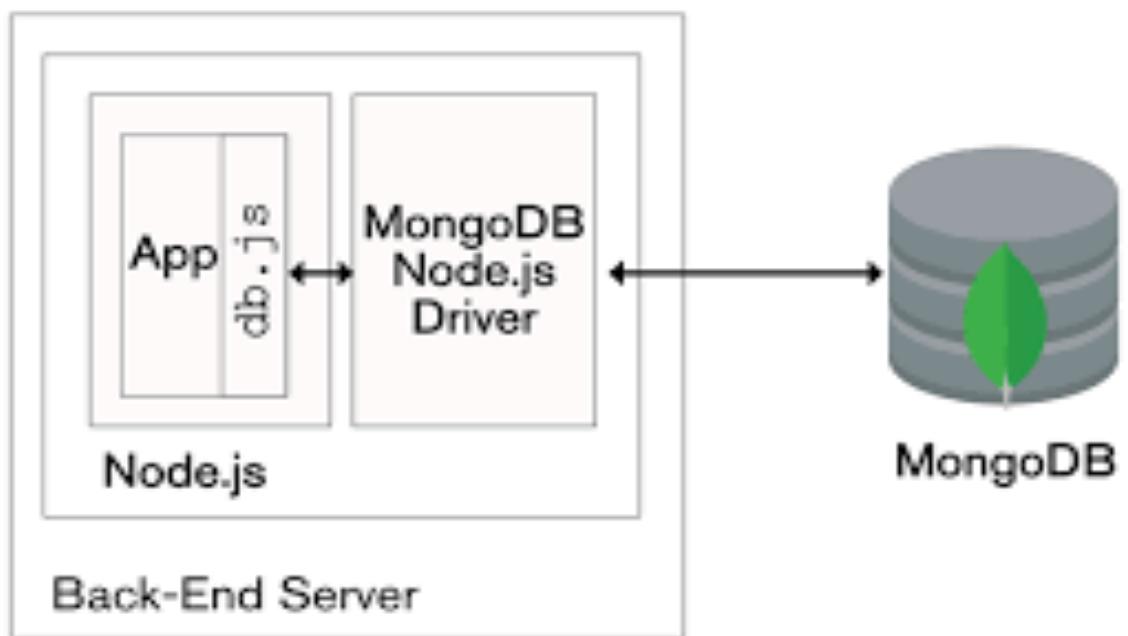
## 4.METHODOLOGY

User data platform is divided into modules like Leave management, Attendance, Analysis, Employee personal info etc.

### 4.1 Project Flow Explanation

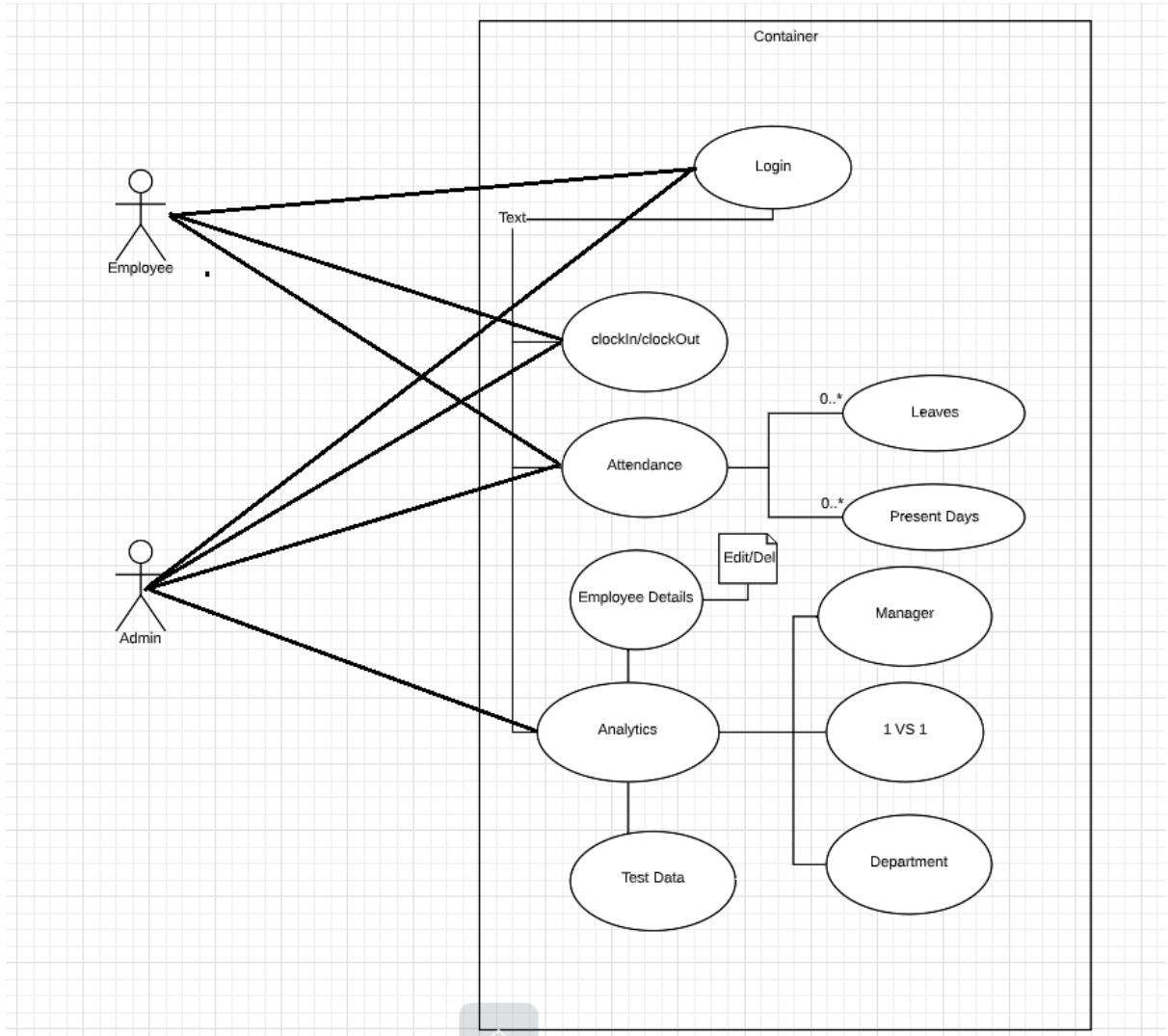
The clients (laptop, desktop, mobile, etc.) as shown in Fig 1, access the web server through the Internet and Nodejs is the one which accepts the requests to ajax and receives the responses.

AJAX calls, which stand for Asynchronous JavaScript and XML, are becoming more popular as web applications look to bring in personalized items dynamically. AJAX calls are one method to load personalized content separately from the rest of the HTML document, which allows for the full HTML document to be cached, improving back end load time. Ajax calls are in turn made to the database for the data manipulation.



**Fig 1:** Architecture & Project flow

#### 4.1.1 Use Case Diagram:



**Fig 2:** Use Case Diagram

The website has two persons (as shown in fig 2), i.e., Admin and Employee, a different login mechanism is in place for each of them.

1. Employee can login into their individual account and access Dashboard relevant to the information provided by them. They can clock In or clock Out, check attendance and make a leave request.
2. Admin can login into their individual account and have special privileges, create, delete update employee data in employee tab, they can also check the performance of each employee, manager, department wise and they can also check the probability of future joining employee based on male, female, married, single.

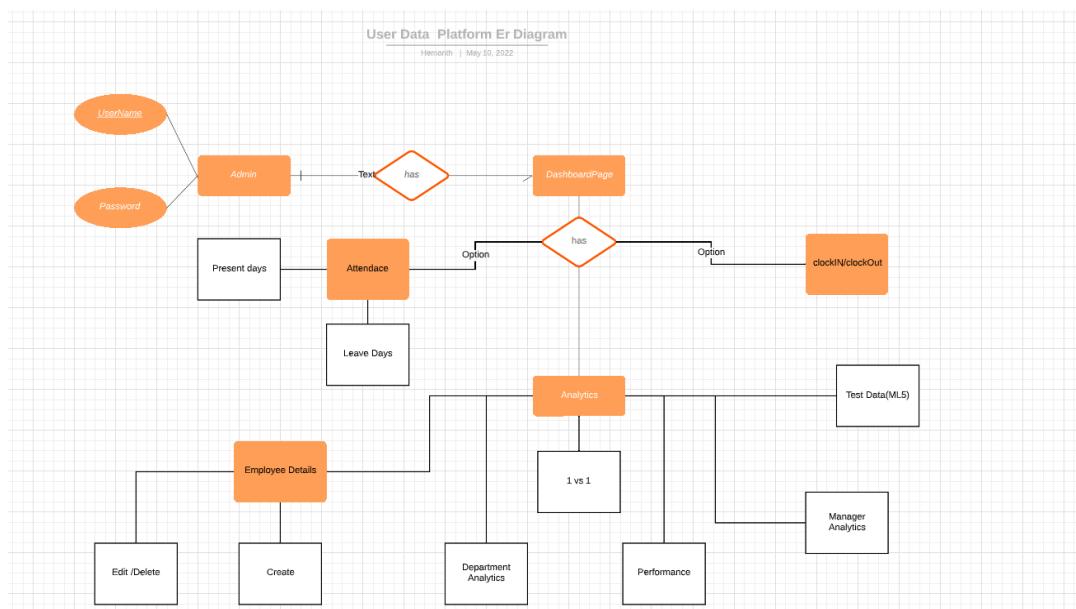
## 4.2 The Front-end:

The front-end of the application is implemented using HTML, CSS, and Bootstrap. The scripting language used for the back end is Node JS and Express JS.

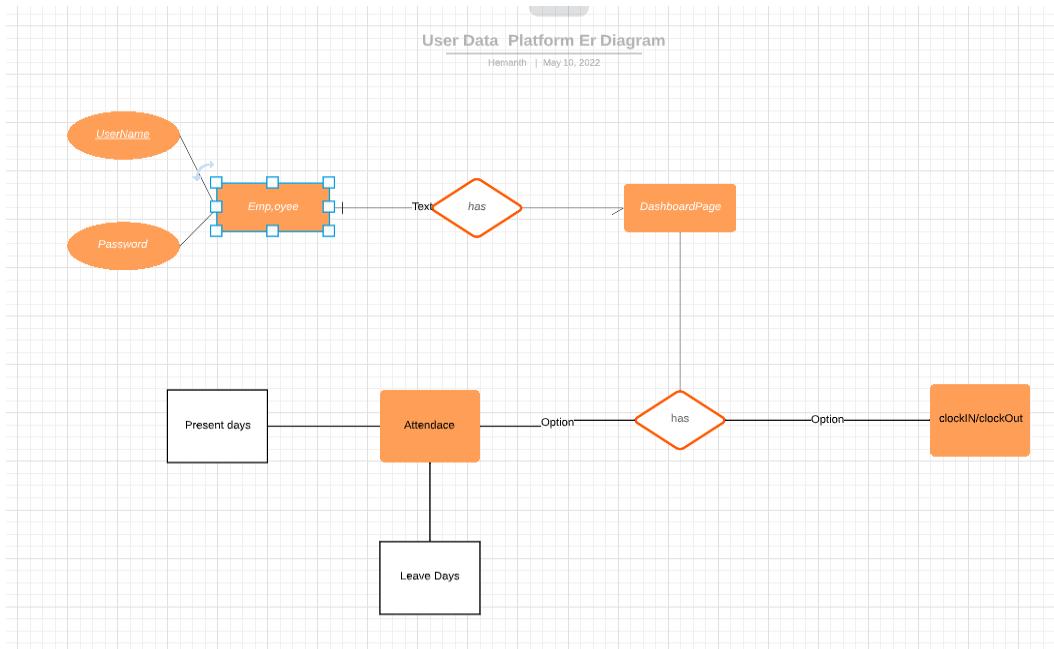
- The Home page is designed using **HTML**, **CSS**, and **Bootstrap**.
- The home page provides the user with two options channel or student registration and login form
- The form is designed using **HTML** and **CSS**.

## 4.3 The Back end:

- On submit, an Email is sent to the user for verification. Once the user clicks submit with correct credentials, the user can login successfully.
- In version 1 we used Mongo DB in the backend to store the data of both Admin and Employee. Data we store is like details of each entity shown.
- The menu has an on-click scroll feature that directs the webpage to a desired location when clicked.
- Every web page is designed using bootstrap components like header, container, grid, card, menu, and footer.



**Fig 3a:** Entity Relationship model (using MongoDb db.)



**Fig 3b:** flow charts

## 5. RESULTS AND DISCUSSIONS

These are the outputs of our user data platform here are tha all the modules in the software that we developed by using Node.js, Express, MongoDb, AWS S3, NPM modules

### 5.1 The Login page:

- Login & Register tabs are shown when any user visits the website (Fig 4).
- In login, email, password, is asked. Once the user enters correct credentials and submits, they will be redirected to their respective web pages according to the user type.

The screenshot shows a 'Login' page with a blue header bar. Below it, there are two input fields: 'Email address' containing 'pradeepkumarreddypeddi739@gmail.com' and 'Password' containing several dots. At the bottom left is a blue 'Submit' button, and at the bottom center is a link 'If you are not a user, [Sign up Here](#)'.

**Fig 4:** Login page

### 5.2 The Register page:

- In this page if the User is new to the website he will be asked to register with his credentials like Full Name, email, Password and confirm password .
- If he is already a user he can go to the login page from register page itself.

**CREATE AN ACCOUNT**

**Full Name \***

**Email \***

**Username already Exists**

**Password \***

**Confirm Password \***

**Register**

Have already an account? [Login here](#)

The image shows a registration form titled "CREATE AN ACCOUNT". It has fields for "Full Name \*", "Email \*", "Password \*", and "Confirm Password \*". The "Email" field contains "pradeepkumarreddypeddi739@gmail.com" and has a red error message "Username already Exists". A large blue "Register" button is at the bottom. Below the form is a link "Have already an account? [Login here](#)".

**Fig 4.1:** Register page

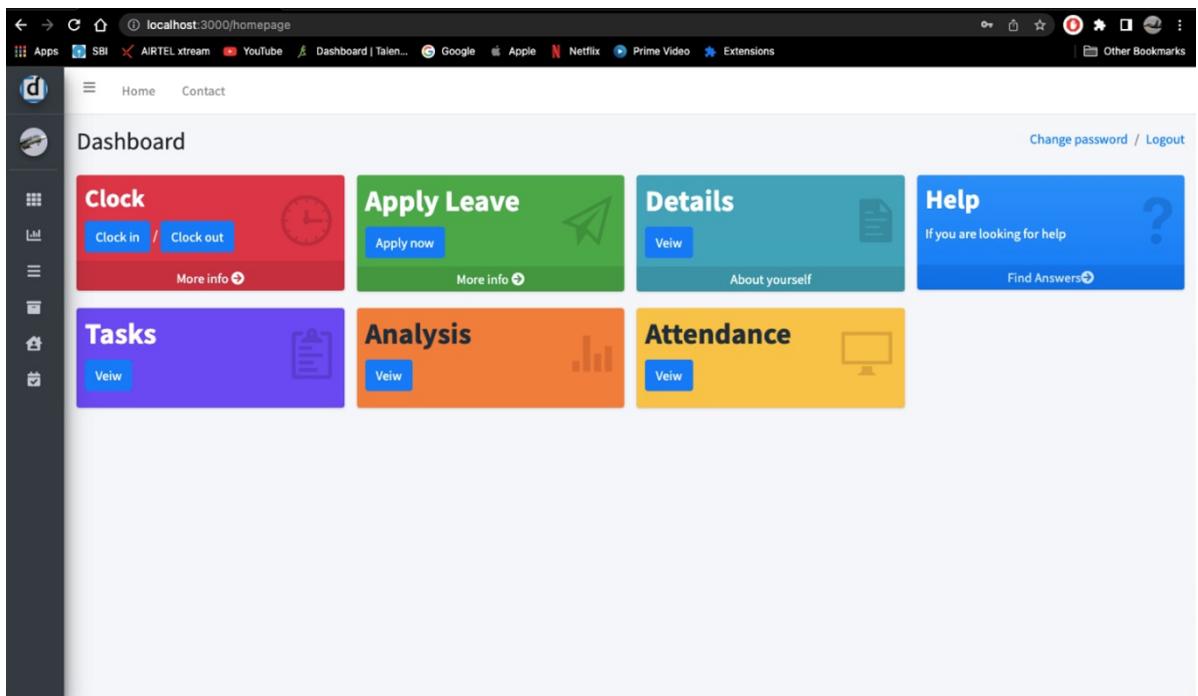
### **5.3 Admin user:**

If the user is of Admin type, five tabs are displayed on the dashboard and other options like changing password, and logging out of the website.

#### **5.3.1 Dashboard:**

On clicking the clock button it will prompt to clock in or clock out .[Fig 5].

- On clicking the attendance tab you will get redirect to the attendance page where you can view your attendance.
- On clicking the Tasks tab you can view your tasks shown in Fig 5.
- On clicking the leave tab you can apply leave.
- Each option has a different menu under itself.



**Fig 5:** Dashboard tab

### 5.3.2 About tab:

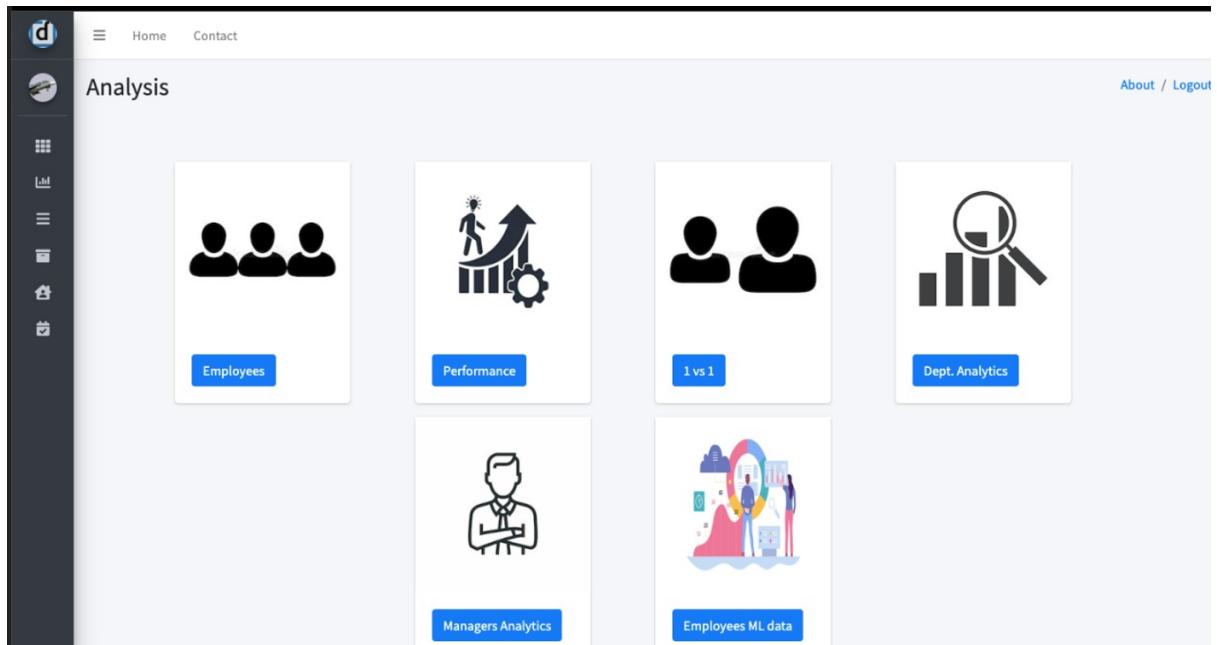
- Various details of the user asked are profile photo, Username, Email.
- The details can be updated at any time.

A screenshot of a registration form titled 'About'. The form has an orange header bar with the text 'Registration page'. Below it are four input fields: 'Username' (containing 'pradeep'), 'Email' (containing 'pradeepkumarreddy739@gmail.com'), 'Photo' (with a placeholder 'Choose file No file chosen'), and an 'Update' button at the bottom.

**Fig 6:** About tab

### 5.3.3 Analytics tab:

- Different graphs are displayed showing various aspects like Employee1 vs Employee2 Performance, Performance of the employee, Managers vs Managers, Department wise analytics and Employee ML data.



**Fig 7:** Analytics tab

### 5.4.1 Employee tab:

- In this tab we can view all the employee data in a form of a table and admin can edit , create or delete the employee details.
- We can see the details of employee

EmployeeID	Employee Name	Designation	Manager Name	Joining Date	Department	Customization
10001	Candie, Calvindroney	Production Manager	Amy Dunn	1/28/2010	Production	/
10002	Anderson, Linda lo	Production Technician I	Amy Dunn	1/9/2012	Production	/
10003	Billis, Helensky	Production Technician I	Brannon Miller	7/7/2014	Production	/
10004	Lynch, Lindsay	Production Technician I	Webster Butler	11/7/2011	Production	/
10005	Patronick, Lucas	Software Engineer	Alex Sweetwater	11/7/2011	Software Engineering	/
10006	Driver, Elle	Area Sales Manager	Lynn Daneault	1/10/2011	Sales	/
10007	Harrison, Kara	Production Technician I	Amy Dunn	5/12/2014	Production	/
10008	Lindsay, Leonara	IT Support	Eric Dougall	1/21/2011	IT/IS	/

**Fig 8a:** employee details

**Employee Form**

**Employee\_Name :** Candie, Calvindroney

**EmpID :** 10001

**MarriedID :** 0

**MaritalStatusID :** 1

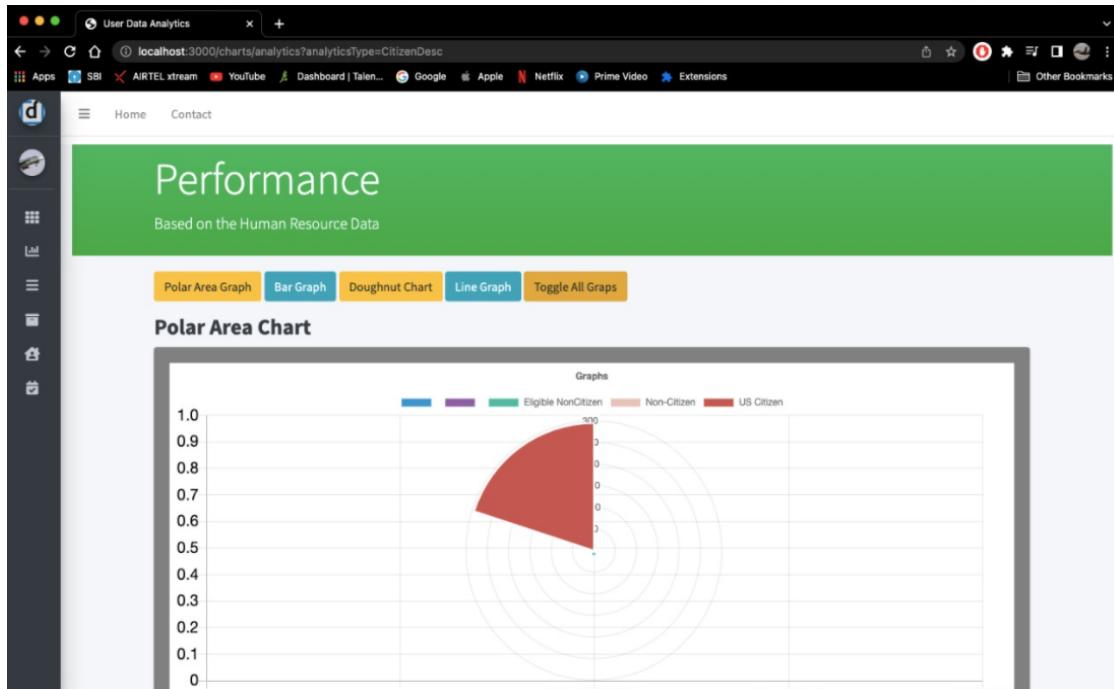
**GenderID :** 0

**EmpStatusID :** 1

**DeptID :**

**Fig 8b:** employee details

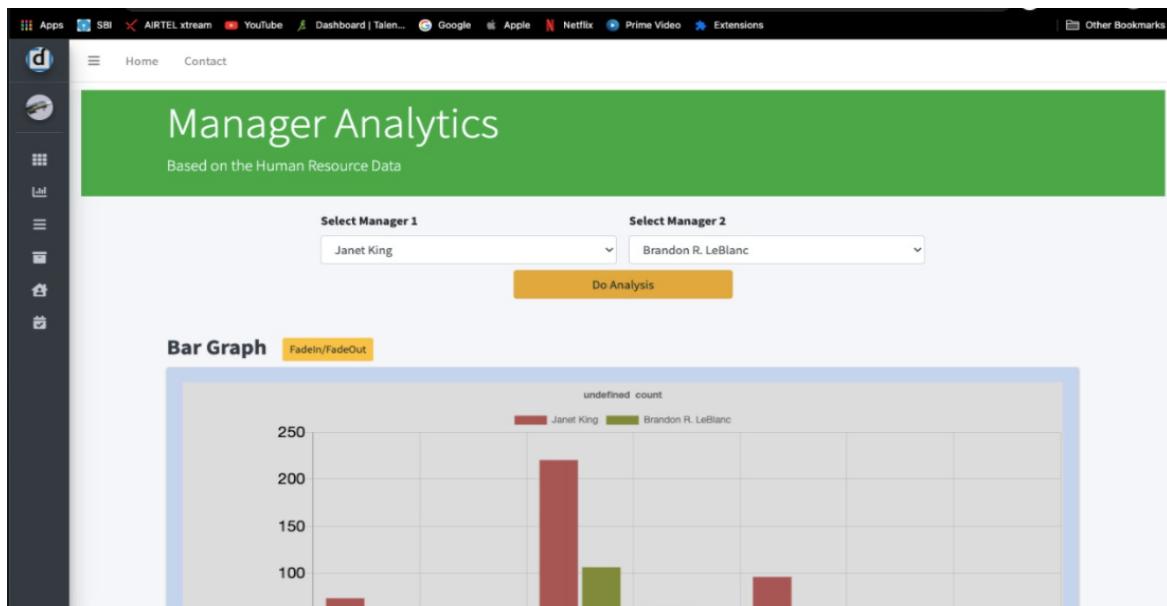
### 5.4.2 Performance analysis tab:



**Fig 9:** performance analysis

### 5.4.3 Manager tab:

- » In this tab you can see the performance of each manager
- » Manager 1 vs Manager 2



**Fig 10:** Manager tab

#### 5.4.4 Departments tab:

- On clicking the Department tab, admin can view all the performance department wise

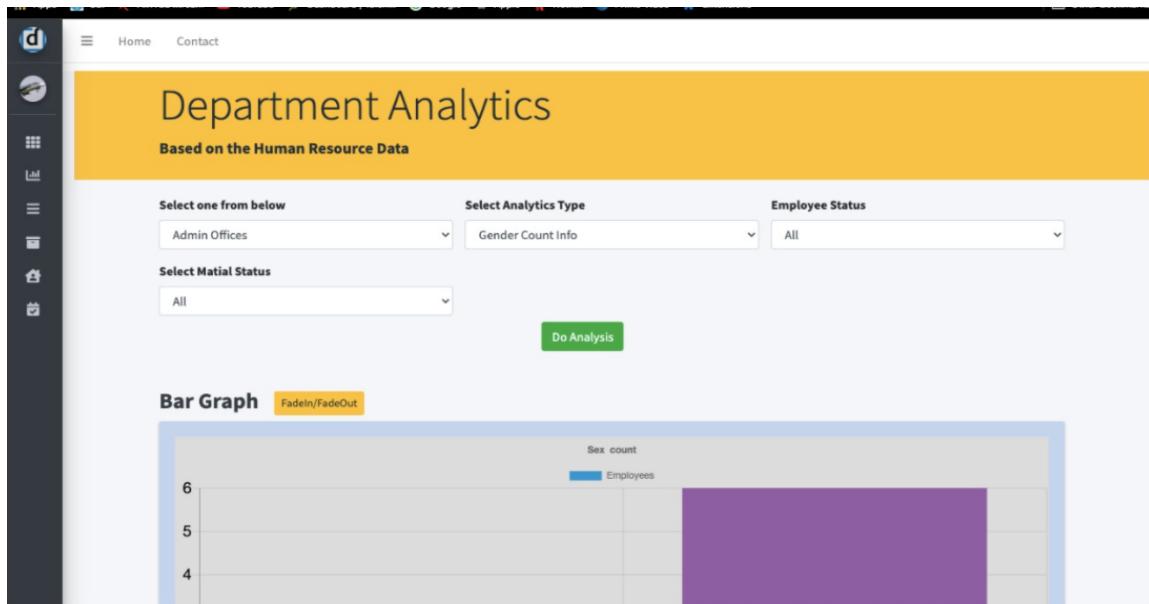


Fig 11: Department tab

#### 5.4.5 Employees 1vs1 tab:

- In this tab admin can view all the performances of each employee by comparing to employee to employee

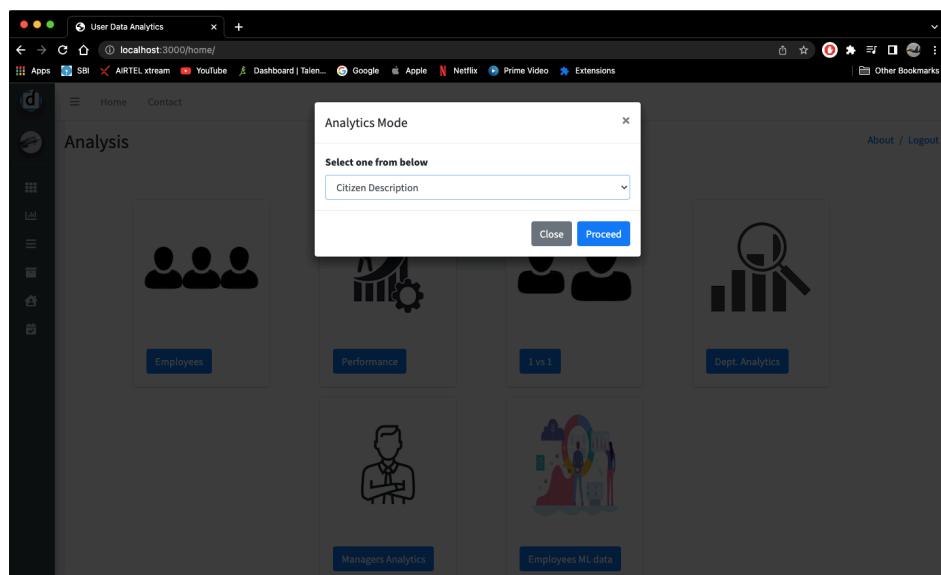
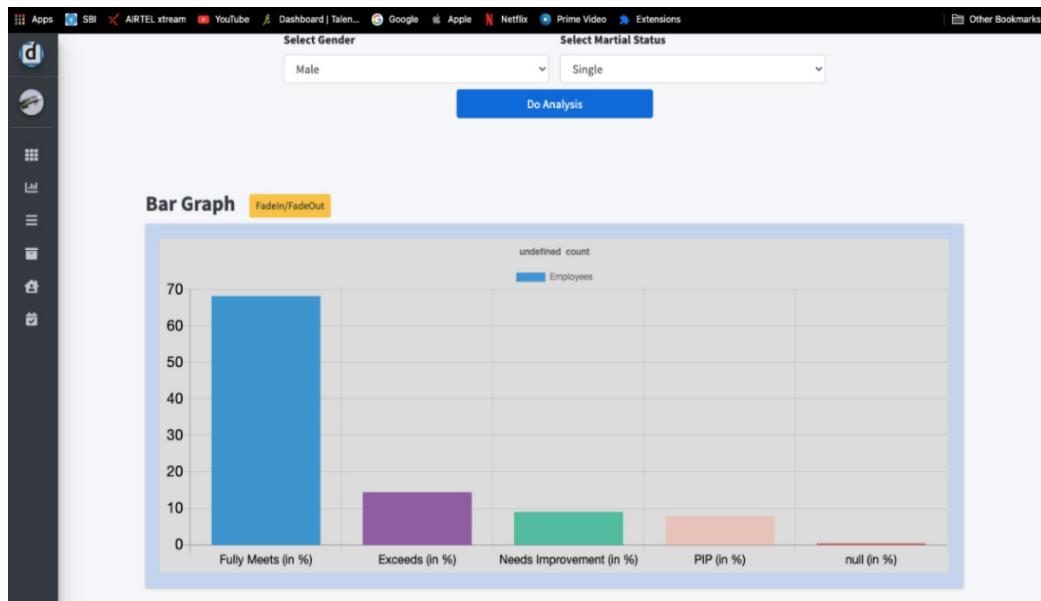


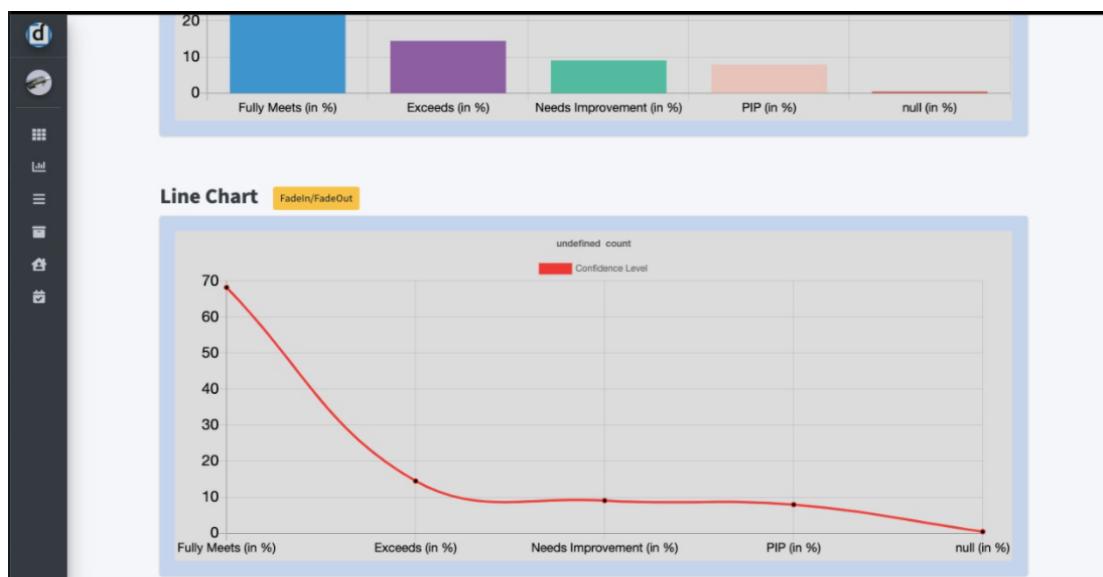
Fig 12: 1vs 1

#### 5.4.6 Analysis tab:

- In this tab admin can view the probability of the upcoming/ newly recruited employee performance based on the previous data of the all other employees.
- It is decided based on the gender, marital status etc.,



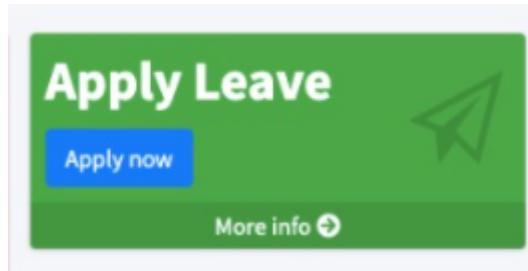
**Fig 13a:** analysis with tfjs



**Fig 13b :** analysis with tfjs

## 5.5 Leave:

- Admin can click on the apply leave tab and can view and apply leaves.



**Fig 14(a):** Apply leave tab

A screenshot of a web-based leave application form. The header includes a logo, "Home", and "Contact". The main title is "Leave application Form". A red bar at the top contains the text "Leave application page". The form fields include:

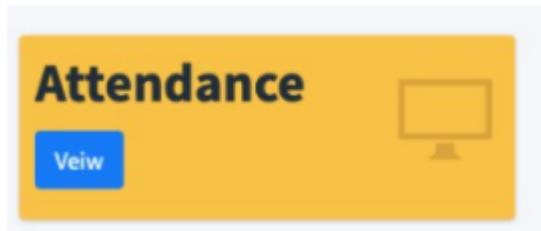
- Username: pradeep
- Type of leave: Please select
- From: dd/mm/yyyy
- To: dd/mm/yyyy
- Reason: Reason

A blue "Apply Leave" button is located at the bottom left of the form area.

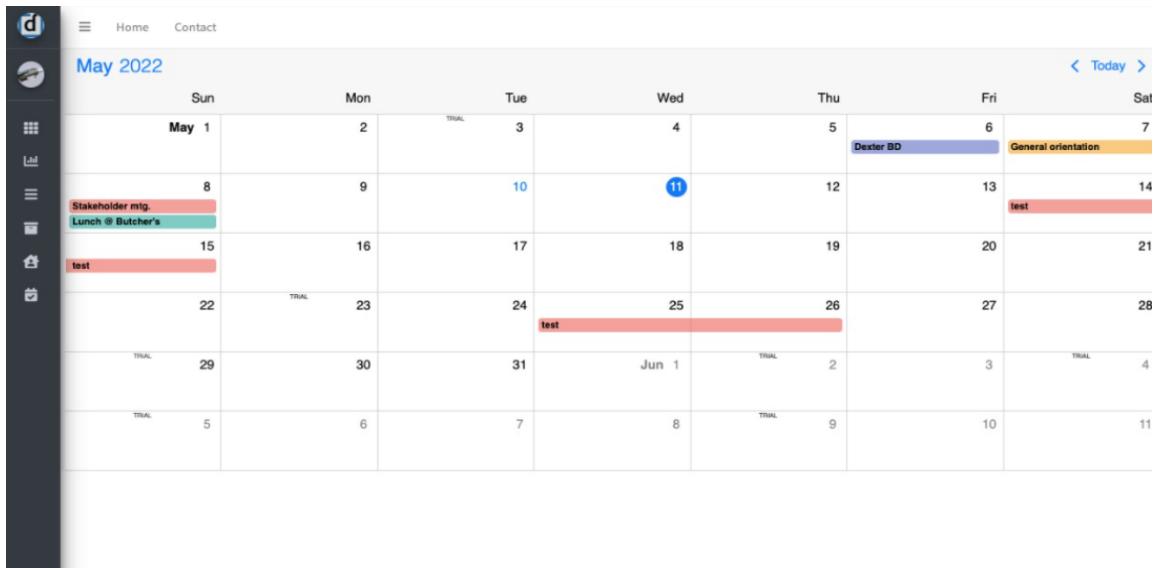
**Fig 14(b):** Leave page

### 5.5.1 Attendance:

- Admin can view his/her attendance.



**Fig 15(a):** Attendance tab



**Fig 15(b): Attendance page**

## 5.6 Employee user:

If any user logs in as a Employee, then four tabs will be displayed – Employee Profile tab,

Employee leave, clock, Attendance

### 5.6.1 Employee Profile tab:

- When this tab is clicked, all the details of the employee are shown name , email, upload profile pic.

Registration page

**Username**  
pradeep

**Email**  
pradeepkumarreddypeddi739@gmail.com

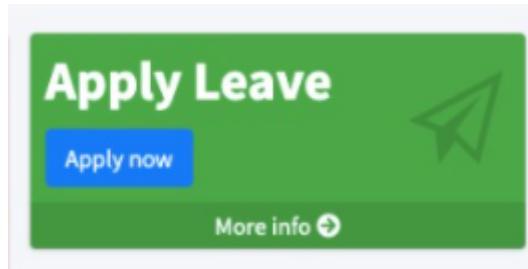
**Photo**  
 No file chosen

**Update**

**Fig 16: Employee Profile Tab**

### 5.6.2 Employee Leave:

- Employee can click on the apply leave tab and can view and apply leaves.



**Fig 17(a):** Apply leave tab

Leave application Form

Leave application page

Username  
pradeep

Type of leave: Please select ▾

From  
dd/mm/yyyy

To  
dd/mm/yyyy

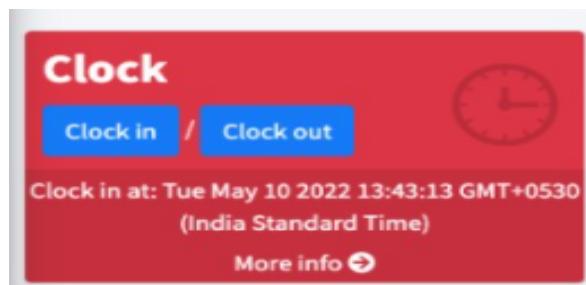
Reason  
Reason

Apply Leave

**Fig 17(b):** Leave page

### 5.6.3 Clock In Clock Out Tab:

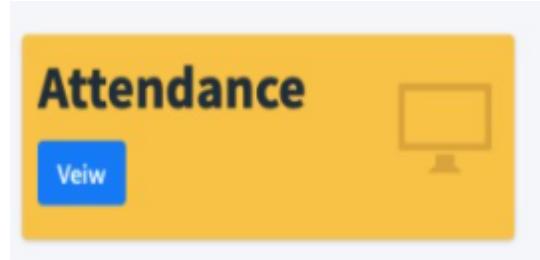
- Employee can clock into the system and clock out and can view his attendance.



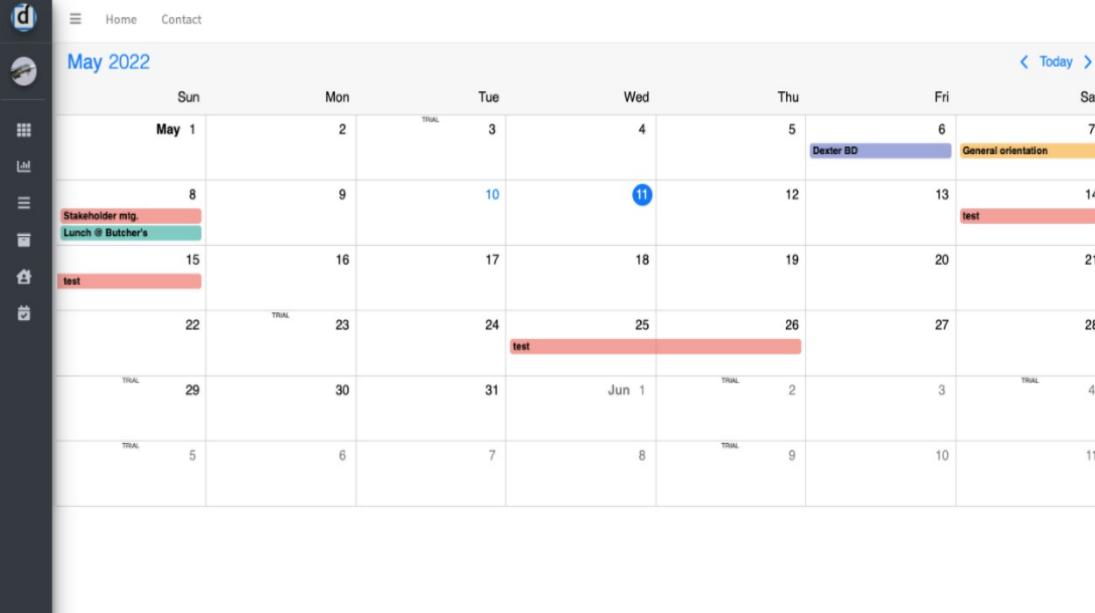
**Fig 18:** Clock tab

#### 5.6.4 Attendance Tab:

- Employee can view his/her attendance.



**Fig 19:** Attendance tab



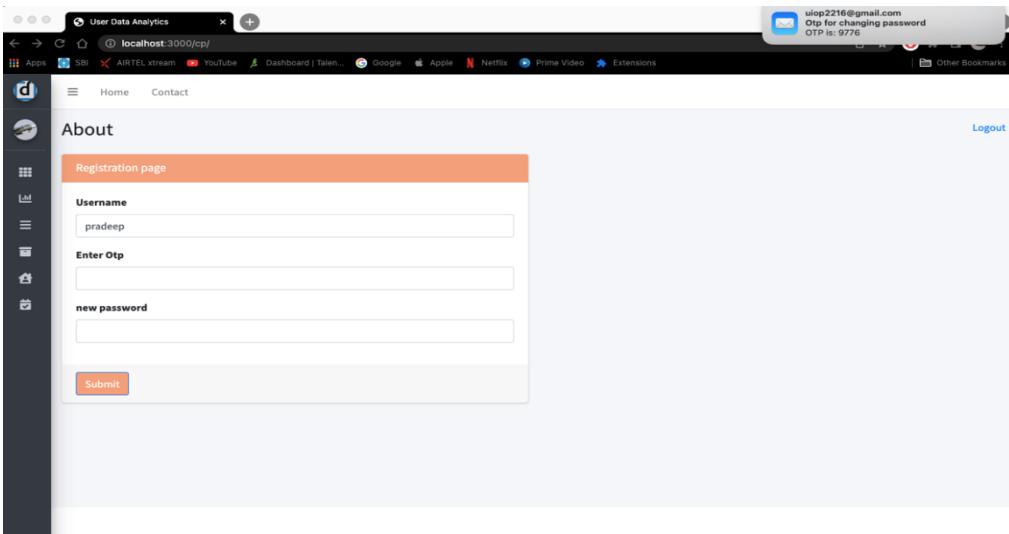
A screenshot of a calendar application for May 2022. The calendar shows days from Sunday to Saturday. Several events are marked with colored boxes: a red box for "Stakeholder mtg.", a green box for "Lunch @ Butcher's", and a pink box for "test". A blue box highlights "Dexter BD" on Friday the 6th. A yellow box highlights "General orientation" on Saturday the 7th. A red box covers the dates from May 24 to May 26, labeled "test". The month of June is partially visible at the bottom. The left sidebar contains icons for Home, Contact, and other functions.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
May 1	2	3	4	5	6	7
8	9	10	11	12	13	14
Stakeholder mtg. Lunch @ Butcher's					test	
15	16	17	18	19	20	21
test						
22	23	24	25	26	27	28
29	30	31	Jun 1	2	3	4
5	6	7	8	9	10	11

**Fig 19(b):** Attendance page

#### 5.7 Change Password tab:

- employees as well as admin can change their password by using this tab located in the home page
- with otp verification through email after that we can change the password



**Fig 20:** Change password page

## 5.8 AWS s3:

- Amazon provides us with the s3 storage cloud space for file storage
- Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance
- It stores in buckets and key value pairs

Name	AWS Region	Access	Creation date
invoiceduserimages	Asia Pacific (Mumbai) ap-south-1	Public	May 2, 2022, 16:23:03 (UTC+05:30)
userdataimages	Asia Pacific (Mumbai) ap-south-1	Public	May 8, 2022, 11:19:46 (UTC+05:30)

**Fig 21:** Aws storage

## 5.9 Mongodb atlas/compass:

- Mongo dB provides us with the NoSQL cloud and local storage to store the data in the place of document object model (DOM)
- It has many functions like aggregation, Indexing etc.

The image displays two screenshots of MongoDB management interfaces. The top screenshot shows the MongoDB Atlas 'Database Deployments' page for a project named 'STUDENT'S ORG - 2022-03-20'. It details a deployment named 'Invoicedb' with metrics: R 0, W 0, Connections 0, In 0.0 B/s, Out 0.0 B/s, Data Size 221.5 KB, and a status of 'Last 6 hours'. The bottom screenshot shows the MongoDB Compass interface for a database named 'FullAssignment'. It lists several collections: 'DepartmentsDB', 'EmployeeDetails', 'leave', 'ManagersDB', 'Positions DB', each showing their storage size, number of documents, average document size, indexes, and total index size. The left sidebar of Compass shows the database structure with '7 DBs' and '23 COLLECTIONS'.

Fig 22(a) & (b) : Mongodbs atlas& compass

## **5. CONCLUSION AND FUTURE SCOPE**

A good HRMS software automates mundane administrative tasks and allows HR leaders to focus on strategic functions, regardless of an organization's size. By implementing an HR management system business can boost employee engagement and reduce turnover.

We developed this software with help of Nodejs, mongo dB, express js, aws s3, ajax and node modules. As it is asynchronous software it helps us to speed up the time and quickly retrieve the data. We have created different modules like leave management, user data, login details,

The scope of HRM is extensive and far-reaching. Therefore, it is very difficult to define it concisely.

In the technological world, organizations are upgrading themselves according to the developments taking place. The future of HRMS in Indian business is shining brightly with the continuing progress in HR technology.

Any manager or employer would want to invest in software that maintains a balance between its employee-centric and company-centric nature, automates most of his manual operations, and would help him in time-management.

In future we can use other npm modules and Heroku for deployment. We can use other improved frame works such as sail.js or vue.js for better MVC structure, by these we can make it readable . By using above features we can make it become cost-effective, easily understandable and efficient.

## **REFERENCES**

- [1][https://en.wikipedia.org/wiki/Human\\_resource\\_managementsystem](https://en.wikipedia.org/wiki/Human_resource_managementsystem)
- [2] <https://darwininbox.com/>
- [3] <https://nodejs.org/en/>
- [4] <https://www.mongodb.com/>
- [5] <https://aws.amazon.com/>
- [6]<https://docs.npmjs.com/>

## APPENDIX

Code:

App:

```
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');
var session = require('express-session');
const { v4: uuidv4 } = require('uuid');
//var brain = require('./routes/ml')

var chartsJS = require('./routes/chartsJS')
var database = require('./routes/database')
var displayWholeData = require('./routes/displayWholeData')
var login = require('./routes/login')
var register = require('./routes/register')
var operationsOnData = require('./routes/operationsOnData')
var tasks = require('./routes/tasks')
var clock = require('./routes/clock')
var leave = require('./routes/leave')
var about = require('./routes/about')
var usertasks = require('./routes/usertasks')
var changepass = require('./routes/changepass')
var app = express();

app.use(logger('tiny'));
app.use(express.json());
app.use(express.urlencoded({ extended: false })); //to use the req.body variable
when the html form is submitted
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
app.use("/home", express.static(path.join(__dirname, 'public')));
app.set('view engine', 'ejs')
app.set('trust proxy', 1) // trust first proxy
app.use(session({
  genid: function (req) {
    return uuidv4() // use UUIDs for session IDs
  },
  secret: 'keyboard cat',
  resave: false,
  saveUninitialized: true,
  cookie: { maxAge: 3000000 },
  store: database.MongoStore.create({
    mongoUrl: database.url,
  }),
}))
```

```

}))
```

```

app.use(async (req, res, next) => {
  try {
    await database.connectDB();
  } catch (error) {
    console.log(error)
  }
  next()
})
var abt;
```

```

app.use('/login', login)
app.use('/register', register)
app.use(async (req, res, next) => {
  try {
    abt=await
    database.getDB().collection("users").findOne( {'name':req.session.name})
  } catch (error) {
    console.log(error)
  }
  next();
})
app.use('/charts', chartsJS)
app.use('/displayData', displayWholeData)
app.use('/operations', operationsOnData)
app.use('/tasks', tasks)
app.use('/clock', clock)
app.use('/leave', leave)
app.use('/about', about)
app.use('/taskview', usertasks)
app.use('/cp', changepass)
```

```

app.get('/homepage',async(req,res)=>{
  un=req.session.name
  var                                     abt=await
  database.getDB().collection("users").findOne( {'name':req.session.name})
  var obj={"username":un,"name":un,"log":"login","ct":"","clock":"","obj":abt}
  res.render('homepage',obj)
})
app.get('/att',async(req,res)=>{
  un=req.session.name
  var att=await database.getDB().collection('leave').find({uid:un}).toArray()
  console.log(att)
  var
  obj={"username":un,"name":un,"log":"login","ct":"","clock":"","leave":att,"obj
  ":abt}
  res.render('calender',obj)
```

```

        })
        app.post('/validateEmail/:id', (req, res) => {

            if (req.body.email.toLowerCase().match(/[a-z0-9]+@[a-z]+\.[a-z]{2,3}/) == null) {
                res.json(
                    {
                        msg: "Not in a valid email format"
                    }
                )
            } else {
                database.getDB().collection('users').findOne({ email: req.body.email }, (err, res1) => {
                    if (err) { throw err }
                    if (res1 == null) {
                        res.json({
                            msg: (req.params.id === "login") ? "Invalid Email" : ""
                        })
                    } else {
                        res.json({
                            msg: (req.params.id === "login") ? "" : "Username already Exists"
                        })
                    }
                })
            }
        })

        app.post("/validateIDs", (req, res) => {
            var empPromise = new Promise((resolve, reject) => {
                database.getDB().collection('EmployeeDetails').findOne({ EmpID: parseInt(req.body.empID) }, (err, res1) => {
                    if (err) {
                        reject(err)
                    } else {
                        resolve(res1)
                    }
                })
            })
            empPromise.then((val) => {
                if (val != null) {
                    res.json(
                        {
                            errMsg: "",
                        }
                    )
                } if (val == null) {

```

```

        res.json(
        {
            errMsg: "Employee_ID doesn't exists"
        }
    )
}

app.get("/", (req, res) => {
    if (req.session.auth) {
        res.redirect('/homepage')
    }
    else {
        res.render('login')
    }
})

app.get('/home', (req, res) => {
    if (req.session.auth) {
        res.render('home', { name: req.session.name , "obj":abt});
    }
    else {
        res.render('unauthorized')
    }
})

app.get('/ml_page', async(req,res)=>{
    let data;
    try {
        data=await
        database.getDB().collection('EmployeeDetails').find({},{projection:{_id:0,Sex:1,MaritalDesc:1,PerformanceScore:1}}).toArray()
    }
    catch (error) {
        console.log(error)
    }
    res.render('analysis',{data:data,name:req.session.name,"obj":abt})
})

app.get('/Managers_Analytics_Page', async (req, res) => {
    var managersDetails = await
    database.getDB().collection('ManagersDB').find({}).toArray()
    res.render('managersAnalysis', { name: req.session.name, managers: managersDetails,"obj":abt })
})

```

```

app.get('/department_Analytics_Page', (req, res) => {
  database.getDB().collection('DepartmentsDB').find({}).toArray((err, res1) => {
    res.render('deptAnalytics', { name: req.session.name, depts: res1 ,"obj":abt})
  })
})

app.get('/trash', async (req, res) => {
  var trashbinData;
  try {
    trashbinData = await database.getDB().collection('trashbin').find({}).toArray();
  } catch (error) {
    alert(error)
  }
  res.render('index', { tableArray: trashbinData, name: req.session.name, mode: "trashbin","obj":abt })
})

app.get('/form', async (req, res) => {
  if (req.session.auth == true) {

    var deptDetails = await database.getDB().collection('DepartmentsDB').find({}).toArray()
    var positionDetails = await database.getDB().collection('PositionsDB').find({}).toArray()
    var managersDetails = await database.getDB().collection('ManagersDB').find({}).toArray()
    console.log(managersDetails)
    var lastEmpID = await database.getDB().collection('EmployeeDetails').find({}).sort({ EmpID: -1 }).limit(1).toArray()
    res.render('form', { name: req.session.name, EmpID: lastEmpID[0].EmpID + 1, mode: "newForm", depts: deptDetails, positions: positionDetails, managers: managersDetails,"obj":abt })

  }
  else {
    res.render('unauthorized')
  }
})
}

app.get('/data', (req, res) => {
  if (req.session.auth == true) {
    database.getDB().collection('EmployeeDetails').find({}).toArray((err, res1) => {
      if (err) throw err;
      res.render('index', { tableArray: res1, name: req.session.name, mode: "employees","obj":abt });
    })
  }
})

```

```

        }
    else {
        res.render('unauthorized')
    }
})

app.post('/unauthorized', (req, res) => {
    res.render('unauthorized')
})

app.get('/logout', (req, res) => {
    req.session.destroy(function () {
        res.redirect('/')
    })
})

module.exports = app;

```

### **Home.ejs:**

```

<html lang="en" style="height: auto;">

    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Dbox | Dashboard</title>
        <%- include("partials/header") -%>
        <link rel="stylesheet"
            href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i
            ,700&amp;display=fallback">

        <link rel="stylesheet" href=".//stylesheets/all.min.css">
        <link rel="stylesheet" href=".//plugins/fontawesome-free/css/all.min.css">
        <link href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css" rel="stylesheet">

        <link rel="stylesheet" href="/stylesheets/tempusdominus-bootstrap-4.min.css">
        <link rel="stylesheet" href="/stylesheets/icheck-bootstrap.min.css">
        <link rel="stylesheet" href="/stylesheets/jqvmap.min.css">
        <link rel="stylesheet" href="/stylesheets/adminlte.min.css">

```

```

<link rel="stylesheet" href="/stylesheets/OverlayScrollbars.min.css">

<link rel="stylesheet" href="/stylesheets/daterangepicker.css">

<link rel="stylesheet" href="/stylesheets/summernote-bs4.min.css">
<style type="text/css">
/* Chart.js */
@keyframes chartjs-render-animation {
    from {
        opacity: .99
    }
    to {
        opacity: 1
    }
}

.chartjs-render-monitor {
    animation: chartjs-render-animation 1ms
}

.chartjs-size-monitor,
.chartjs-size-monitor-expand,
.chartjs-size-monitor-shrink {
    position: absolute;
    direction: ltr;
    left: 0;
    top: 0;
    right: 0;
    bottom: 0;
    overflow: hidden;
    pointer-events: none;
    visibility: hidden;
    z-index: -1
}

.chartjs-size-monitor-expand>div {
    position: absolute;
    width: 1000000px;
    height: 1000000px;
    left: 0;
    top: 0
}

.chartjs-size-monitor-shrink>div {
    position: absolute;
    width: 200%;
    height: 200%;
    left: 0;
}

```

```

        top: 0
    }
</style>
<link rel="stylesheet" href="plugins/datatables-
bs4/css/dataTables.bootstrap4.min.css">
<link rel="stylesheet" href="plugins/datatables-
responsive/css/responsive.bootstrap4.min.css">
<link rel="stylesheet" href="plugins/datatables-
buttons/css/buttons.bootstrap4.min.css">
</head>

<aside class="main-sidebar sidebar-dark-primary elevation-4">

    <a href="index3.html" class="brand-link">
        
        <span class="brand-text font-weight-light">Dbox</span>
    </a>

    <div
        class="sidebar os-host os-theme-light os-host-overflow os-host-
overflow-y os-host-resize-disabled os-host-scrollbar-horizontal-hidden os-host-
transition">
        <div class="os-resize-observer-host observed">
            <div class="os-resize-observer" style="left: 0px; right: auto;"></div>
        </div>
        <div class="os-size-auto-observer observed" style="height:
calc(100% + 1px); float: left;">
            <div class="os-resize-observer"></div>
        </div>
        <div class="os-content-glue" style="margin: 0px -8px; width: 249px;
height: 674px;"></div>
        <div class="os-padding">
            <div class="os-viewport os-viewport-native-scrollbars-invisible"
style="overflow-y: scroll;">
                <div class="os-content" style="padding: 0px 8px; height: 100%;
width: 100%;">
                    <div class="user-panel mt-3 pb-3 mb-3 d-flex">
                        <div class="image">
                            <% if(username=='pradeep') { %>
                                
                            <%} else if(username=='abd') { %>

```

```

        
            <%} else if(username=='ram') {%
                
                    <%} else {%
                        
                            <%}%
                    </div>
                <div class="info">
                    <a href="#" class="d-block"><%=username%></a>
                </div>
            </div>
            <%- include('partials/sidebar.ejs'); %>
        </div>
    </div>
<div class="os-scrollbar os-scrollbar-horizontal os-scrollbar-unusable
os-scrollbar-auto-hidden">
    <div class="os-scrollbar-track">
        <div class="os-scrollbar-handle" style="width: 100%; transform:
translate(0px, 0px);"></div>
    </div>
    </div>
    <div    class="os-scrollbar    os-scrollbar-vertical    os-scrollbar-auto-
hidden">
        <div class="os-scrollbar-track">
            <div    class="os-scrollbar-handle"    style="height: 30.8783%;
transform: translate(0px, 0px);">
                </div>
            </div>
        </div>
        <div class="os-scrollbar-corner"></div>
    </div>
</div>

</aside>

<div class="content-wrapper" style="min-height: 618px;">

    <div class="content-header">
        <div class="container-fluid">
            <div class="row mb-2">
                <div class="col-sm-6">
                    <h1 class="m-0">Dashboard</h1>
                </div>
                <div class="col-sm-6">
                    <ol class="breadcrumb float-sm-right">
                        <% if(log=='login'){%
                            <%} else {%

```

```

        <li class="breadcrumb-item"><a href="/login">Login</a></li>
        <% } %>
        <% if(log=='login') { %>
        <li class="breadcrumb-item active"><a href="/cp">Change
password</a></li>
        <li class="breadcrumb-item" active"><a href="/logout">Logout</a></li>
        <% } else { %>
        <li class="breadcrumb-item active"><a href="/reg">Sign
In</a></li>
        <% } %>
        </ol>
        </div>
        </div>
        </div>
</div>

<section class="content">
<div class="container-fluid">

<% if(log=='login') { %>

<div class="row">

<div class="col-lg-3 col-6">
<div class="small-box bg-danger">
<div class="inner">
<% if(log=='login') { %>
<h3 style="font-size: 1.85em">Clock</h3>
<a href="/clock/cin"><button type="button" class="btn
btn-primary">Clock in</button></a> / <a href="/clock/cout"><button
type="button" class="btn btn-primary">Clock out</button></a>
<% } else { %>
<h4 style="font-size: 1.6em"> Please login for clock In
</h4>
<p></p>
<% } %>
</div>
<div class="icon">
<i class="ion ion-clock"></i>
</div>
<a href="#" class="small-box-footer"><%=clock%>
<%=ct%><i
class=""></i></a>
<a href="#" class="small-box-footer">More info <i
class="fas fa-arrow-circle-right"></i></a>
</div>

```

```

</div>

<div class="col-lg-3 col-6">
    <div onclick="window.location='/leave'" class="small-box
bg-success">
        <div class="inner">
            <% if(log=='login')%>
                <h3>Apply Leave </h3>
                <a href="/leave"><button type="button" class="btn
btn-primary">Apply now</button></a>
            <% } else {%
                <h4></h4>
                <h4>Login for leave application
                </h4>
                <p></p>
            <% } %>
        </div>
        <div class="icon">
            <i class="ion ion-paper-airplane"></i>
        </div>
        <a href="#" class="small-box-footer">More info <i
            class="fas fa-arrow-circle-right"></i></a>
    </div>
</div>

<div class="col-lg-3 col-6">
    <div onclick="window.location='/about'" class="small-box
bg-info">
        <div class="inner">
            <% if(log=='login')%>
                <h3>Details</h3>
                <a href="/about"><button type="button" class="btn
btn-primary">View</button></a>
            <% } else {%
                <h4></h4>
                <h4>
                </h4>
                <p></p>
            <% } %>
        </div>
        <div class="icon">
            <i class="ion ion-document-text"></i>
        </div>
        <a href="#" class="small-box-footer">About yourself<i
            class="fas fa-arrow-circle-right"></i></a>
    </div>
</div>

<div class="col-lg-3 col-6">

```

```

<div class="small-box bg-gradient-primary">
  <div class="inner">
    <h3 style="font-size: 2.0em">Help</h3>
    <p>If you are looking for help</p>
  </div>
  <div class="icon">
    <i class="ion ion-help"></i>
  </div>
  <a href="https://darwinbox.com/" onclick="return help()">
    Find Answers<i class="fas fa-arrow-circle-right"></i></a>
  </div>
</div>
<!-- next row -->
<div class="row">

  <div class="col-lg-3 col-6">
    <div onclick="window.location='/taskview'" class="small-box bg-indigo">
      <div class="inner">
        <h3>Tasks</h3>
        <a href="/taskview"><button type="button" class="btn btn-primary" style="align-content: center;">View</button></a>
        <h3></h3>
      </div>
      <div class="icon">
        <i class="ion ion-clipboard"></i>
      </div>
    </div>
  </div>
</div>

<% if(username=='pradeep' ||username=='hemanth') {%>
<div class="col-lg-3 col-6">
  <div onclick="window.location='/home'" class="small-box bg-orange">
    <div class="inner">
      <h3>Analysis</h3>
      <a href="/home"><button type="button" class="btn btn-primary" style="align-content: center;">View</button></a>
      <h3></h3>
    </div>
    <div class="icon">
      <i class="ion ion-stats-bars"></i>
    </div>
  </div>
</div>

```

```

<%}else {} %>

<div class="col-lg-3 col-6">
<div onclick="window.location='/att'" class="small-box bg-
yellow">
<div class="inner">
<h3>Attendance</h3>
<a href="/att"><button type="button" class="btn btn-
primary" style="align-content: center;">View</button></a>
<h3></h3>
</div>
<div class="icon">
<i class="ion ion-monitor"></i>
</div>

</div>
</div>
</div>

</div>

<% } else {}%>

<div class="row">

<div class="col-lg-3 col-6">
<div class="small-box bg-warning">
<div class="inner">
<% if(log=='login'){}%>
<h3>Register user </h3>
<a href="/reg"><button type="button" class="btn btn-
primary">User Registration</button></a>
<% } else {}%>
<h4></h4>
<a href="/reg"><button type="button" class="btn btn-
primary">User Registration</button></a>
<h4></h4>
<% } %>

</div>
<div class="icon">
<i class="ion ion-person-add"></i>
</div>
<a href="" class="small-box-footer">T&C applicable <i
class="fas fa-arrow-circle-right"></i></a>
</div>
</div>

```

```

<div class="col-lg-3 col-6">
    <div class="small-box bg-info">
        <div class="inner">
            <h3>Help</h3>
            </div>
            <div class="icon">
                <i class="ion ion-help"></i>
            </div>
            <a href="/help" onclick="return help()" class="small-box-footer">Find Answers<i
                class="fas fa-arrow-circle-right"></i></a>
            </div>
        </div>
        </div>
    <%}%>

    </div>
</section>

</div>

<div id="sidebar-overlay"></div>
</div>

<script src="/javascripts/jquery.min.js"></script>

<script src="/javascripts/jquery-ui.min.js"></script>

<script>
    $.widget.bridge('uibutton', $.ui.button)
</script>

<script src="/plugins/jqvmap/maps/jquery.vmap.usa.js"></script>

<script src="/plugins/tempusdominus-bootstrap-4/js/tempusdominus-bootstrap-4.min.js"></script>

<script src="/plugins/summernote/summernote-bs4.min.js"></script>

<script src="/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js"></script>

<script src="/javascripts/adminlte.js?v=3.2.0"></script>

<script src="/dist/js/pages/dashboard.js"></script>

<script>
    function help(){

```

```

        if(window.confirm("if you want to continue to external page")){
            return true;
        }
        else{
            return false;
        }
    }
</script>

</body>

</html>

{
    "name": "assignexpress",
    "version": "0.0.0",
    "private": true,
    "scripts": {
        "start": "node ./bin/www"
    },
    "dependencies": {
        "alert": "^5.0.10",
        "attendance-calendar": "^1.0.2",
        "aws-sdk": "^2.1130.0",
        "bcryptjs": "^2.4.3",
        "brain.js": "^2.0.0-beta.15",
        "chart.js": "^3.7.1",
        "connect-mongo": "^4.6.0",
        "cookie-parser": "~1.4.4",
        "debug": "~2.6.9",
        "ejs": "^3.1.6",
        "express": "~4.16.1",
        "express-session": "^1.17.2",
        "gpu.js": "^2.15.2",
        "jsonwebtoken": "^8.5.1",
        "mongodb": "^4.4.1",
        "morgan": "~1.9.1",
        "multer": "^1.4.4",
        "multer-s3": "^2.10.0",
        "nodemailer": "^6.7.5",
        "path": "^0.12.7",
        "timestamp": "^0.0.1",
        "uuid": "^8.3.2"
    }
}

```