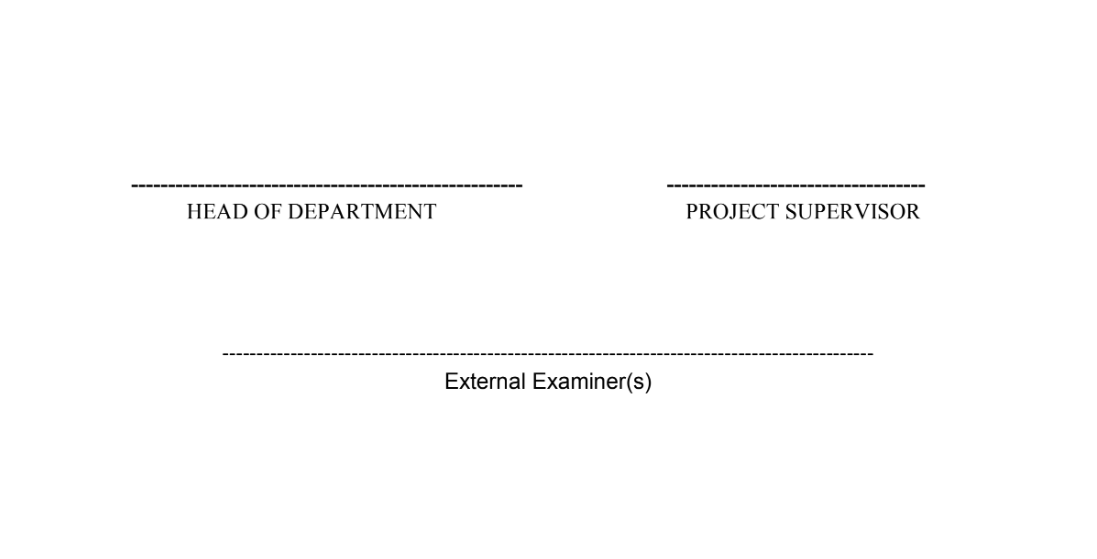


This is to certify that the project synopsis entitled **“Employee Leave Management System”** submitted for partial  fulfillment of the requirements of 6th Semester of B.Sc. in Computer Science under CBCS  Syllabus; has been carried out by Mr. **Tushar das (Roll No.: 213017-21-0071 Registration No.: 017-1111-0615-21)** under the supervision of **Mr. Sanjib Halder,  Department of Computer Science, The Bhawanipur Education Society College,  University of Calcutta.**



**ACKNOWLEDGEMENT**

The development of this Online Employee Leave Management System would not have been possible without the invaluable support and guidance of my esteemed project guide, **Mr. Sanjib Halder Associate Professor Department of Computer Science, The Bhawanipur Education Society College,  University of Calcutta**. I am incredibly grateful for his patience, insightful feedback, and encouragement throughout the project. His expertise in computer science was instrumental in helping me navigate the technical challenges and ensuring the successful completion of this project.

Furthermore, I would like to express my appreciation to the faculty members of the Computer Science department of my college for providing a stimulating learning environment and fostering my interest in the field.

I would also like to extend my thanks to my friends and class mates and especially to my cousin Rittika Addhya for her valuable assistance with some of the design aspects of this application. Her creative input helped shape the user interface and enhance the overall user experience.

This project has been a rewarding experience, allowing me to apply the knowledge and skills acquired during my B.Sc. (Hons.) in Computer Science program. I am grateful for the opportunity to have learned and grown through the development process.

THANK YOU

Date: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mr. Tushar Das**

|  |  |
| --- | --- |
|  | **Page No.** |
| Abstract | 6 |
| Introduction | 6 - 8 |
| Background and Review of Related Work | 9 |
| Methodology | 10 - 13 |
| Implementation | 14 – 91 |
| Results and Discussion | 92 – 103 |
| Conclusion | 104 |
| References | 105 |

**TABLE OF CONTENT**

**Abstract:**

* **Objective:** Develop a web application to automate and streamline employee leave management.
* **Technologies:** HTML, CSS, JavaScript, PHP, MySQL.
* **Functionalities:**
  + Employee leave request submission (electronic, with attachments).
  + Leave balance tracking.
  + Mobile-friendly, user-friendly interface.
  + Secure, role-based access control.
  + Manager review with additional employee information (single-manager approval currently).

**Introduction:**

**Domain Description:**

This project addresses the common challenge of managing leave applications within organizations. Traditionally, this process involves manual paperwork, often leading to inefficiencies, delays, and potential errors.

**Motivation:**

The motivation behind this project is to automate the leave application process, streamlining it for both employees and managers. This web application aims to:

* Reduce manual workload and paperwork associated with leave requests.
* Improve efficiency and processing time for leave applications.
* Enhance data security and record-keeping for leave management.

**Scope of Work:**

This web application is designed and developed from scratch, utilizing **HTML**, **CSS**, **JavaScript**, **PHP**, and **MySQL**. It offers a comprehensive suite of functionalities, including:

* **Leave Application Submission:**

Employees can submit leave requests electronically, specifying details like leave type, duration, reason, and attaching supporting documents.

* **Leave Balance Tracking:**

An integrated system tracks remaining leave balances for each employee, providing transparency and encouraging responsible leave planning.

* **Manager Review with Enhanced Information:**

While reviewing leave requests, managers can access additional employee details like rank, contact information, and current leave balance to make informed decisions.

* **Hierarchical Approval System:**

Leave requests follow a predefined approval chain, with managers at different levels providing consent.

* **Role-Based Access Control:**

A secure login system grants access based on user roles (employee, manager, HR).

* + Employees can submit requests, upload documents, and track their leave balance.
  + Managers have approval authority within their hierarchy, can access additional employee details, and make informed decisions.
  + HR manages user accounts and leave rules, with the ability to add, remove, and update them.
* **Elaborate Error and Input Handling:**

A robust system prevents invalid data entry, ensures data safety, and provides informative error messages to enhance user experience.

* **Intuitive User Interface:**

A highly user-friendly interface makes navigating the application effortless for all user roles.

**Tools used:**

**Front-End Functionality:**

* **HTML:**

This forms the foundation of this web application, defining the structure and content of each webpage, including the layout of forms, leave request details, and dashboards.

* **CSS:**

CSS controls the visual presentation of the application. It determins the visual design, layout, fonts, and colors for a user-friendly interface.

* **JavaScript:**

In this project, JavaScript is primarily used to implement auto-scrolling functionalities on certain webpages, making navigation smoother.

**Server-Side Functionality:**

* **PHP:**

This acts as the engine behind the scenes. It processes user requests, interacts with the MySQL database, and generates dynamic content. PHP handles tasks like:

* + Verifying user login credentials with hashed passwords for enhanced security.
  + Processing leave request submissions and storing data securely in the database.
  + Retrieving leave request information for employees and managers to view.
* **MySQL:**

This powerful database stores all the crucial information for the application. It manages data like employee details, leave request details, leave balances, and approval history.

**Version Control:**

* **Git with Sourcetree GUI:**

This version control system is essential for managing the codebase. It tracks changes made to the HTML, CSS, JavaScript, and PHP code throughout development. This allows you to:

* + Revert to previous versions if necessary (in case of bugs)
  + Maintain a clear history of code changes
  + Potentially collaborate with others on the project (future enhancement)

**Background and Review of Related Work:**

**Manual Leave Management Challenges:**

Traditionally, leave applications are managed through paper forms and manual processes. This approach suffers from several drawbacks:

* **Inefficiency:**

Manual processing is time-consuming for both employees submitting requests and managers reviewing them.

* **Prone to Errors:**

Data entry mistakes can occur during manual processing, leading to inconsistencies and delays.

* **Difficulties in Tracking:**

Monitoring status of a leave application, leave balances and trends becomes cumbersome with manual methods.

**Existing Leave Management Systems:**

Several commercial and open-source leave management systems address these challenges.

Some popular options include:

* **Zoho People** (<https://www.zoho.com/people/help/adminguide/leave-intro.html>)
* **BambooHR** (<https://www.bamboohr.com/>)
* **LeaveBoard** (<https://leaveboard.com/>)
* **OpenHRMS** (<https://www.openhrms.com/>)

Here's a brief review of them:

* **Commercial Systems:**

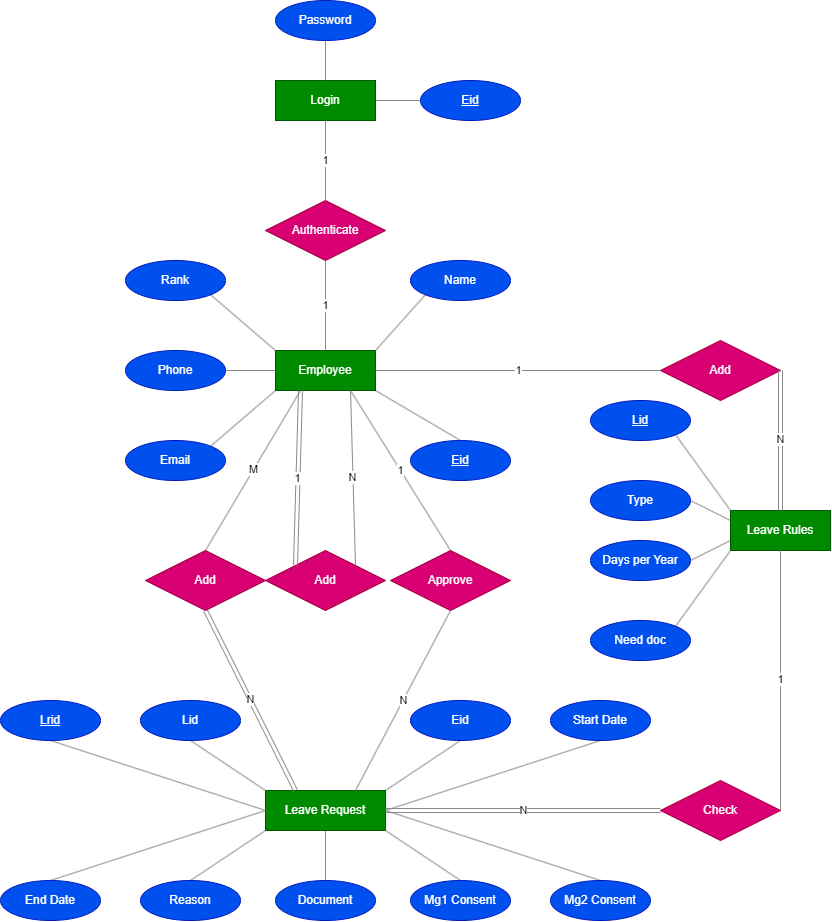
Paid leave management software offers comprehensive features like leave tracking, reporting, and integrations with payroll systems. However, these can be expensive for smaller organizations. (Examples: BambooHR, Zoho People)

* **Open-Source Systems:**

Free and open-source alternatives provide core functionalities like leave requests and approvals. However, customization options and user support may be limited. (Examples: OrangeHRM, OpenHRMS)

**Methodology:**

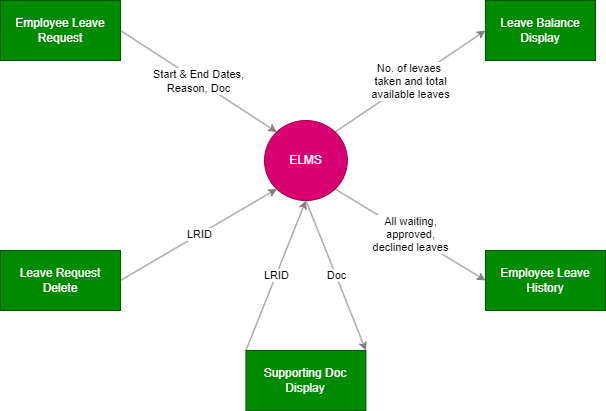
ER diagram of the Database used for this Employee Leave Management System.



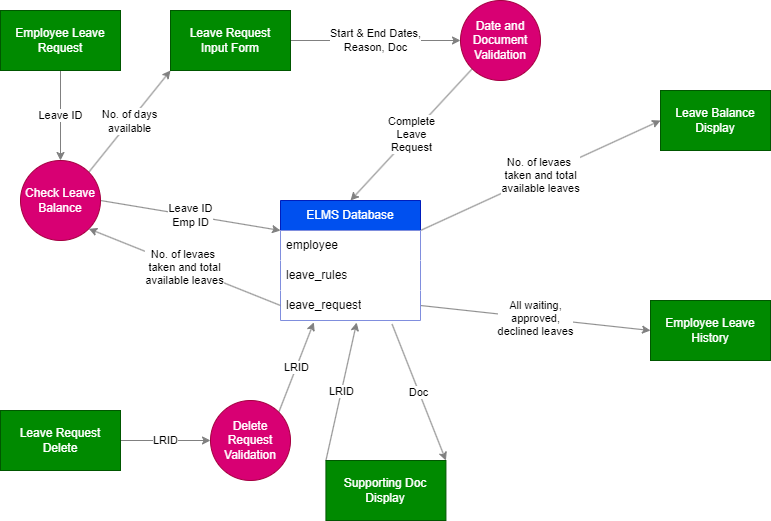
Here I present Data Flow Diagrams of the three sections in this project.

**Employee Section:**

**Level 0 DFD:**

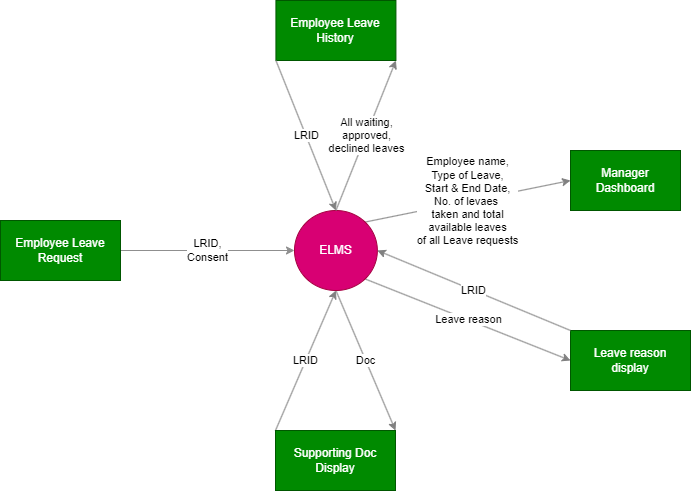


**Level 1 DFD:**

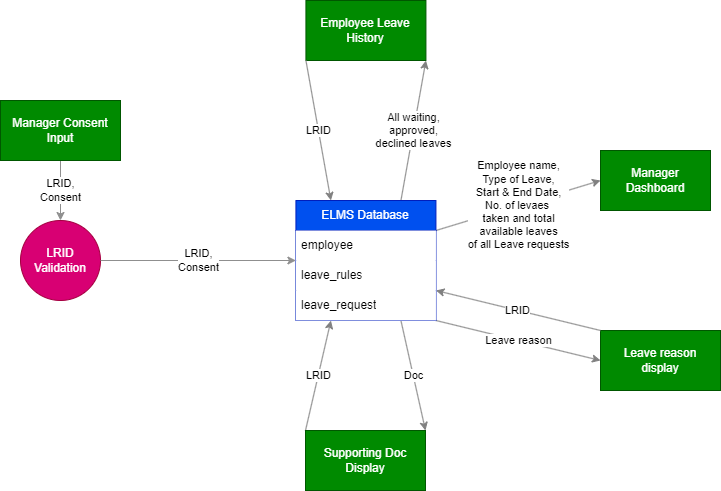


**Manager section:**

**Level 0 DFD:**

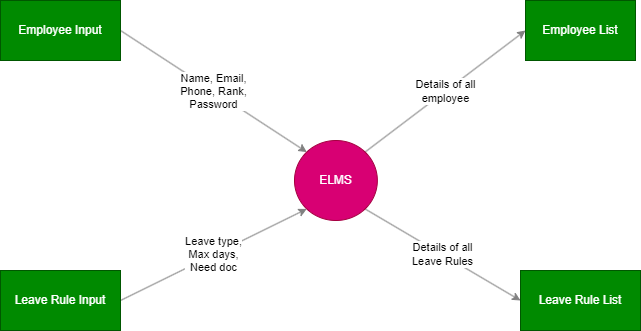


**Level 1 DFD:**

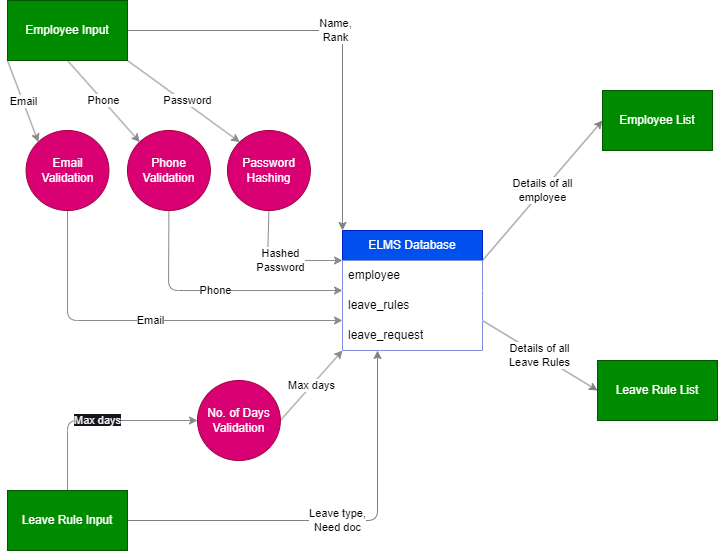


**HR section:**

**Level 0 DFD:**



**Level 1 DFD:**



**Implementation:**

**MySQL commands used to design the Database**

CREATE DATABASE ELMS;

# the login table

# employee(eid, name, email, phone, rank(em(0), m1(1), m2(2), hr(3)))

CREATE TABLE employee(

eid BIGINT AUTO\_INCREMENT,

name VARCHAR(30) NOT NULL,

email VARCHAR(50) NOT NULL,

phone BIGINT(10) NOT NULL,

ranks TINYINT NOT NULL,

# constraints

CONSTRAINT pk PRIMARY KEY(eid),

CONSTRAINT ec UNIQUE(email),

CONSTRAINT rk CHECK(ranks >= 0 AND ranks <= 3)

);

=====================================================================

=====================================================================

# the leave rules table

# leave\_rule(lid, name, days, need\_doc)

CREATE TABLE leave\_rule(

lid BIGINT AUTO\_INCREMENT,

type VARCHAR(30) NOT NULL,

days SMALLINT NOT NULL,

need\_doc BOOLEAN NOT NULL,

# constraints

CONSTRAINT pk PRIMARY KEY(lid)

);

=====================================================================

=====================================================================

# the leave request table

# leave\_request(lrid, eid, lid, start\_date, end\_date, support\_doc, ftype, mg1\_consent, mg2\_consent)

CREATE TABLE leave\_request(

lrid BIGINT AUTO\_INCREMENT,

eid BIGINT,

lid BIGINT,

start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

reason VARCHAR(500) NOT NULL,

support\_doc MEDIUMBLOB,

ftype VARCHAR(5), # file extension

mg1\_consent VARCHAR(500),

mg2\_consent VARCHAR(500),

# constraints

CONSTRAINT pk

PRIMARY KEY(lrid),

CONSTRAINT fk1

FOREIGN KEY(eid) REFERENCES employee(eid)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT fk2

FOREIGN KEY(lid) REFERENCES leave\_rule(lid)

ON DELETE CASCADE

ON UPDATE CASCADE

);

=====================================================================

=====================================================================

# the login table

# login(eid, password)

CREATE TABLE login(

eid BIGINT,

password VARCHAR(255) NOT NULL,

# constraints

CONSTRAINT pk PRIMARY KEY(eid),

CONSTRAINT fk

FOREIGN KEY(eid) REFERENCES employee(eid)

ON DELETE CASCADE

ON UPDATE CASCADE

);

**The CSS scripts used here**

**1. >>>> CSS\general styles.css**

@font-face {

    font-family: Caveat;

    src: url("../Caveat/Caveat-VariableFont\_wght.ttf");

}

@font-face {

    font-family: Segoe\_ui;

    src: url("../segoe\_ui/Segoe UI.ttf");

}

html

{

    box-sizing: *border-box*;

}

\*, \*::after, \*::before

{

    font-family: Segoe\_ui, sans-serif;

    box-sizing: *inherit*;

*/\* To prevent font resizing in android \*/*

    -webkit-text-size-adjust: *none*;

    -moz-text-size-adjust: *none*;

    -ms-text-size-adjust: *none*;

}

ul

{

    margin: 0;

    padding: 0;

    list-style-type: *none*;

}

body

{

*/\**

*Here, we consider 1vmax : 1vmin = 16 : 9, hence, we calculate (10/16 vmax + 10/9 vmin) \* 1.05*

*\*/*

    font-size: calc((.625vmax + 1.111vmin) \* 1.07);

    padding: 0;

}

main

{

    min-height: 100vh;

}

a

{

    text-decoration: *none*;

}

*/\**

*provids the nice shadow around any element, any element can use*

*it to get the nice shadow*

*\*/*

.nice\_shadow

{

    box-shadow: 0px 0px 128px rgba(0, 0, 0, 0.2), 0px 32px 100px -48px rgba(0, 0, 0, 0.8) !important;

*/\**

*here we use two overlapping shadows*

*First one has a blur radius of 128px, that means,*

*128px: This sets the blur radius of the shadow, which controls how soft*

*or hard the edges of the shadow appear. A higher value creates a softer,*

*more diffused shadow.*

*This creats an ambiant shadow around the image*

*For the second image we use spread as -48px*

*-48px: This introduces a new property called "spread". A positive spread*

*value expands the shadow beyond the blur radius, while a negative value*

*shrinks it inwards. Here, -48px shrinks the shadow inwards, creating a*

*darker inner area.*

*We also place this shadow slightly lower to create some round shadow below*

*the image.*

*\*/*

}

.no\_shadow

{

    box-shadow: *none* !important;

}

**2. >>>> CSS\form styles.css**

*/\* disable the zooming of the viewport with touch input \*/*

:root

{

    touch-action: *pan-x* *pan-y*;

    height: 100%;

}

*/\**

*body gets the margin to prevent the form from collasping into the border*

*\*/*

body

{

    line-height: 2em;

    padding: 1px;

}

*/\**

*main element contains the form, so making it flex box to keep the*

*form in the middle*

*\*/*

main

{

    margin: 2em 10%;

    display: *flex*;

    flex-direction: *column*;

*/\* keeping the things inside in center \*/*

    align-items: *center*;

    justify-content: *center*;

    min-height: 90dvh;

}

*/\**

*actual form structure,*

*Here we start to design the actual form using a flex box*

*here we use max width to let it srink in smaller screen*

*\*/*

*/\**

*An <ul> is given id = main\_box. It contains the flex box, where the*

*elements (<li>) are arranged from top to bottom*

*\*/*

.main\_box

{

    width: 27em;

    margin: .5em;

    padding: 2em;

    border-radius: .6em;

    display: *flex*;

    flex-direction: *column*;

    background-color: white;

}

*/\**

*input elements*

*\*/*

*/\**

*how input boxes and buttons should look like*

*\*/*

.main\_box > \* .heading, .main\_box > \* input, .main\_box > \* select, .main\_box > \* textarea, .main\_box > \* .button, .main\_box > \* .message, .main\_box > \* .file

{

*/\* fill entire space of parent element ".main\_box > \*" \*/*

    min-width: 100%;

    line-height: 2em;

    min-height: 2em;

    padding: 0 .5em;

    border-radius: .3em;

    border-width: 1px;

    font-size: *inherit*;

    border-color: rgb(71, 28, 114);

}

*/\**

*some extra property for form heading*

*\*/*

.main\_box > \* .heading

{

    padding: 0;

    color: orange;

}

*/\**

*some extra property for text area*

*\*/*

.main\_box > \* textarea

{

    font: *inherit*;

    resize: *vertical*;

    min-height: 6em;

    line-height: 1.2em;

    padding: .5em;

    margin-bottom: -.5em;

}

*/\**

*spacing the list items a bit*

*\*/*

.main\_box > \*

{

    margin: .5em;

    flex: 1 1 *auto*;

}

*/\**

*some extra property for messages*

*there after any element with button class will look like this*

*\*/*

.main\_box > \* .message

{

    border-style: *dashed*;

    height: *fit-content*;

}

*/\* for error message \*/*

.main\_box > \* .error

{

    background-color: rgba(255, 0, 0, 0.1);

}

*/\* for info \*/*

.main\_box > \* .info

{

    background-color: rgba(0, 255, 0, 0.1);

}

*/\**

*some extra property for buttons*

*there after any element with button class will look like this*

*\*/*

.main\_box > \* .button

{

    background-color: rgb(173, 200, 230);

}

*/\* increasing space above form input buttons only for asthetics \*/*

.main\_box > \* input.button

{

    margin-top: 1.25em;

}

.main\_box > \* .file

{

    box-shadow: 0px 0px 5px -1px rgba(0, 0, 255, 0.6);

}

**3. >>>> CSS\list styles.css**

*/\* enabling touch action for index pages \*/*

:root

{

    touch-action: *manipulation*;

}

*/\* changing some attributes of main\_box \*/*

.main\_box

{

    width: 95%;

    padding: 1em;

    box-shadow: 0px 0px 5px -1px rgba(0, 0, 0, 0.3);

    flex-direction: *row*;

    flex-wrap: *wrap*;

    justify-content: *space-evenly*;

}

*/\* for list we want to have buttons and messages of perticular size \*/*

.main\_box > \* .button, .main\_box > \* .message

{

*/\* base size of this thing (size can be extended from here but can't compressed) \*/*

    width: 15em;

}

*/\* making message look softer with a nice shadow \*/*

.main\_box > \* .message

{

    box-shadow: 0px 0px 5px -1px rgba(0, 0, 0, 0.1);

    border: *none*;

}

*/\* used to display the answer \*/*

.main\_box > \* .error

{

    text-align: *center*;

    box-shadow: *none*;

}

.main\_box > \* .button

{

    border: *none*;

    box-shadow: 0px 0px 5px -1px rgba(0, 0, 0, 0.1);

}

.main\_box > \* .greenbutton, .main\_box > \* .redbutton, .main\_box > \* .bluebutton, .main\_box > \* .whitebutton

{

    background-color: white;

}

.red\_box, .main\_box > \* .redbutton

{

    box-shadow: 0px 0px 5px -1px rgba(255, 0, 0, 0.6);

}

.green\_box, .main\_box > \* .greenbutton

{

    box-shadow: 0px 0px 5px -1px rgba(0, 140, 0, 0.8);

}

.blue\_box, .main\_box > \* .bluebutton

{

    box-shadow: 0px 0px 5px -1px rgba(0, 0, 204, 0.8);

}

.red\_text

{

    color: rgba(255, 0, 0, 0.6);

}

.green\_text

{

    color: rgba(0, 140, 0, 0.8);

}

.blue\_text

{

    color: rgba(0, 0, 204, 0.8);

}

**4. >>>> CSS\dashboard styles.css**

*/\**

*"dashboard\_menu" is an extension of message*

*"d\_box" is an extension of button*

*\*/*

.main\_box > \* .d\_box, .main\_box > \* .dashboard\_menu

{

*/\* base size of this thing (size can be extended from here but can't compressed) \*/*

    width: 20em;

*/\* these element will have slightly bigger size than message or button \*/*

}

.main\_box > \* .dashboard\_menu

{

    padding: .5em;

    display: *flex*;  */\* to properly align the progress bar \*/*

    white-space: *nowrap*;

    flex-direction: *column*;

}

.dashboard\_menu > span

{

    text-align: *center*;

}

**5. >>>> CSS\header styles.css**

*/\* header styles \*/*

*/\**

*we set height for header and main section in a way to prevent*

*the page from scrolling*

*\*/*

ul.header\_list

{

    min-height: 3.5em;

    width: 100%;

    display: *flex*;

    align-items: *center*;

    background: rgb(230, 203, 173);

    border-radius: .3em;

    box-shadow: 0px 0px 5px -1px rgba(0, 0, 0, 0.7);

    white-space: *nowrap*;    */\* prevent header text from wrapping \*/*

}

main

{

    min-height: 0dvh;   */\* only to prevent the header bar of mobile browsers from vanishing \*/*

    min-height: calc(90dvh - 3.5em);

}

*/\**

*each element inside flex box*

*\*/*

ul.header\_list li

{

    padding: 0 .6em;    */\* 1em padding on both side \*/*

}

*/\**

*this also a flex box element*

*\*/*

ul.header\_list img

{

    width: 1.5em;

    margin-left: 1em;

    padding: 0.09em;

    border-radius: 5px;

}

ul.header\_list img:hover

{

    box-shadow: 0px 0px 5px -1px rgba(0, 0, 204, 0.8);

}

*/\* the hypertexts in header tab looks like buttons \*/*

.header\_button

{

    width: *max-content*;

    margin: .5em;

}

*/\* right most button, thrown to extreme right of the header bar \*/*

.left\_most

{

    margin-left: *auto*;

}

**6. >>>> CSS\index styles.css**

*/\* this styles for the index page that contains some blue buttons only \*/*

*/\* enabling touch action for index pages \*/*

:root

{

    touch-action: *manipulation*;

}

*/\**

*removing the shadow around the header as*

*we are using the buttons with purple border only*

*\*/*

ul.header\_list

{

    box-shadow: *none*;

}

**>>>> The PHP and HTML scripts used here**

**>>>> PHP\check\_destroy\_secure\_session.php (to create check and destroy secure Session)**

<?php

*/\**

*perfect function to destroy a session and all its data*

*including the cookie generated for it*

*\*/*

**function** destroy\_session\_and\_data()

    {

*$\_SESSION* = [];

        setcookie(session\_name(), "", time() - 3000000, '/');

        @session\_destroy(); *// disable warning for nonexistence session*

    }

*/\**

*returns false if the session is not secure*

*\*/*

**function** check\_secure\_session()

    {

        session\_start();

*/\**

*prevent session fixation by changing session id each time a*

*session starts to make sure that attacker cannot use previously*

*used session id to enter an unclosed session*

*\*/*

        session\_regenerate\_id();

*/\**

*preventing session hijacking by hashing the combination of client*

*IP address and client's browser user agent string and storing it*

*into the session (here we check for it), so that another person*

*from another device cannot enter the session*

*\*/*

        if(isset(*$\_SESSION*['validate']) == false || *$\_SESSION*['validate'] != hash("ripemd128", *$\_SERVER*["REMOTE\_ADDR"] . *$\_SERVER*['HTTP\_USER\_AGENT']))

        {

            destroy\_session\_and\_data();

            return false;

        }

        return true;

    }

?>

**>>>> PHP\check\_emp\_session.php (similarly “HR\_NAME” and “MANAGER\_NAME” session variables are used to check hr and manager session)**

<?php

*// check if current session is a valid admin session else open index.php*

    include "check\_destroy\_secure\_session.php";

    if(!(check\_secure\_session() && isset(*$\_SESSION*['EMPLOYEE\_NAME'])))

    {

*// invalid emp session*

        header("location: ../index.php");

        exit();

    }

?>

**>>>> PHP\config.php (to established link with the database)**

<?php

*// disable the runtime exceptions due to failed queries*

    mysqli\_report(MYSQLI\_REPORT\_OFF);

*// establishing link to the server*

*$link* = @new mysqli("localhost", "root", "", "ELMS");

    if(*$link* -> *connect\_error*)

    {

        die("Failure (*$link* -> *connect\_error*)");

    }

?>

**>>>> PHP\doc\_display.php (function used to display an image or pdf)**

<?php

**function** doc\_display(*$doc*, *$ftype*)

    {

        if(*$ftype* == 'pdf')

        {

*// state that following data will be pdf document*

            header("Content-Type: application/pdf");

*/\**

*"Content-Disposition: inline" means that the pdf needed to be displayed in the browser not downloaded*

*we also give the pdf generated a new name, which will be used as the name of the pdf if it's downloaded*

*\*/*

            header("Content-Disposition: inline; filename=support\_doc.pdf");

*// if your browser doesn't display pdf, directly downloads it, you need to turn on display pdf feature*

        }

        else

        {

*// state that following data will be image*

            header("Content-type: image/*$ftype*");

        }

*/\**

*return binary data (file content extracted from img table stored in $row['image']*

*as string) of the content (image/pdf)*

*\*/*

        echo *$doc*;

    }

?>

**>>>> PHP\emp\_ranking\_system.php**

<?php

**function** get\_rank(*$ranks*)

    {

        switch(*$ranks*)

        {

            case 0: return "Employee"; break;

            case 1: return "Manager1"; break;

            case 2: return "Manager2"; break;

            case 3: return "HR"; break;

        }

    }

*// accepts integer rank and returns name of the folder containing the section*

**function** get\_rank\_section(*$ranks*)

    {

        switch(*$ranks*)

        {

            case 0: return "employee"; break;

            case 1:

            case 2: return "manager"; break;

            case 3: return "hr"; break;

        }

    }

**function** is\_manager(*$ranks*)

    {

        switch(*$ranks*)

        {

            case 0: return false; break;

            case 1:

            case 2: return true; break;

            case 3: return false; break;

        }

    }

?>

**>>>> PHP\form\_input\_check.php (counting no. of leave days available to certain employee)**

<?php

*/\**

*this function will check if given elements exists in get or post array*

*returns true if all of them exists else return false*

*\*/*

**function** elements\_exist(*$arr*, *$element\_list*)

    {

        for(*$i* = 0; *$i* < count(*$element\_list*); *$i*++)

        {

*$str* = *$arr*[*$element\_list*[*$i*]];

            if(!isset(*$str*) || strlen(*$str*) == 0)

            {

                return false;

            }

        }

        return true;

    }

**function** length\_check(*$str*, *$length*)

    {

        return strlen(*$str*) == *$length*;

    }

**function** max\_length\_check(*$str*, *$length*)

    {

        return strlen(*$str*) <= *$length*;

    }

*/\**

*concate multiple url parameters back to back each other*

*\*/*

**function** url\_parameters(*$get*, *$msg*)

    {

        if(isset(*$get*))

        {

*$get* = "*$get*&*$msg*";

        }

        else

        {

*$get* = "*$msg*";

        }

        return *$get*;

    }

?>

**>>>> PHP\leave\_days.php**

<?php

**function** count\_leave\_days(*$start\_date*, *$end\_date*)

    {

*$start\_date* = strtotime(*$start\_date*);

*$end\_date* = strtotime(*$end\_date*);

        return round((*$end\_date* - *$start\_date*) / (60 \* 60 \* 24)) + 1;

    }

**function** total\_leave\_days(*$link*, *$lid*)

    {

*// total no. of days for this leave*

*$result* = *$link* -> query("SELECT days FROM leave\_rule WHERE lid = *$lid*");

        if(*$result* -> *num\_rows* == 0)

        {

*// leave not exist*

            return 0;

        }

        return *$result* -> fetch\_assoc()['days'];

    }

**function** used\_leave\_days(*$link*, *$eid*, *$lid*)

    {

*// total no. of days of this leave are used by this employee (approved or requested leaves)*

*$query* = "SELECT SUM(days) AS total\_days FROM (SELECT DATEDIFF(end\_date, start\_date) + 1 AS days FROM leave\_request WHERE eid = *$eid* AND lid = *$lid* AND (mg2\_consent = 'A' OR mg2\_consent is NULL)) AS days\_table";

*$result* = *$link* -> query(*$query*);

*$used\_days* = *$result* -> fetch\_assoc()['total\_days'];

        if(*$result* -> *num\_rows* == 0 || *$used\_days* == false)

        {

*// leave not taken*

            return 0;

        }

        return *$used\_days*;

    }

**function** approved\_leave\_days(*$link*, *$eid*, *$lid*)

    {

*// total no. of days of this leave are used by this employee (approved or requested leaves)*

*$query* = "SELECT SUM(days) AS total\_days FROM (SELECT DATEDIFF(end\_date, start\_date) + 1 AS days FROM leave\_request WHERE eid = *$eid* AND lid = *$lid* AND (mg2\_consent = 'A')) AS days\_table";

*$result* = *$link* -> query(*$query*);

*$used\_days* = *$result* -> fetch\_assoc()['total\_days'];

        if(*$result* -> *num\_rows* == 0 || *$used\_days* == false)

        {

*// leave not taken*

            return 0;

        }

        return *$used\_days*;

    }

**function** leave\_days\_remaining(*$link*, *$eid*, *$lid*)

    {

        return total\_leave\_days(*$link*, *$lid*) - used\_leave\_days(*$link*, *$eid*, *$lid*);

    }

?>

**>>>> PHP\leave\_request\_status.php**

<?php

**function** get\_status\_emoji(*$mg2consent*)

    {

        switch(*$mg2consent*)

        {

            case'A': return "<b><span class = 'green\_text'>/ Approved</span></b>";  *//"Approved";*

            case'': return "<b><span class = 'blue\_text'>/ Waiting ...</span></b>"; *//"Waiting";*

            default: return "<b><span class = 'red\_text'>/ Declined</span></b>";    *//"Declined";*

        }

    }

**function** get\_status\_color(*$mg2consent*)

    {

        switch(*$mg2consent*)

        {

            case'A': return "green";    *//"Approved"*

            case'': return "blue";  *//"Waiting"*

            default: return "red";  *//"Declined"*

        }

    }

?>

**>>>> PHP\message\_box.php (uses message.php to display proper error message)**

<?php

*// uses message.php to display an error message along with a OK button (attached to a url)*

**function** message\_box(*$msg*, *$url*, *$error*)

    {

*$\_SESSION*['msg'] = *$msg*;

*$\_SESSION*['url'] = *$url*;

*$\_SESSION*['error'] = *$error*;

        header("location: ../message.php");

        exit(0);

    }

?>

**>>>> PHP\mysql\_sanitize\_input.php**

<?php

**function** mysql\_sanitize\_input(*$link*, *$string*)

    {

        return trim(htmlentities(*$link* -> real\_escape\_string(*$string*)));

    }

?>

**>>>> PHP\self\_redirect\_once.php (This provides a complex feature. It can be used to make a HTML form clean its inputs automatically when it’s reloaded or visited back after submission, so that same data can’t be input more than once. It also prevents form resubmission error or warning.)**

<?php

*/\**

*this function creates a new session variable, redirects to*

*the same file once, and deletes that variable.*

*this is used to clear previous inputs to the form while going*

*back to an already submitted copy of that form, thus preventing*

*possibility of resubmission of same data.*

*\*/*

**function** self\_redirect\_once(*$get\_variables* = '')

    {

        if(!isset(*$\_SESSION*['redirect\_once']))

        {

*$\_SESSION*['redirect\_once'] = true;

            header("location: " . *$\_SERVER*['SCRIPT\_NAME'] . *$get\_variables*);

            exit();

        }

        else

        {

            unset(*$\_SESSION*['redirect\_once']);

        }

    }

?>

**>>>> PHP\start\_secure\_session.php**

<?php

**function** start\_secure\_session()

    {

        session\_start();

*/\**

*prevent session fixation by changing session id each time a*

*session starts to make sure that attacker cannot use previously*

*used session id to enter an unclosed session*

*\*/*

        session\_regenerate\_id();

*/\**

*preventing session hijacking by hashing the combination of client*

*IP address and client's browser user agent string and storing it*

*into the session, so that another person from another device cannot*

*enter the session*

*\*/*

*$\_SESSION*['validate'] = hash("ripemd128", *$\_SERVER*["REMOTE\_ADDR"] . *$\_SERVER*['HTTP\_USER\_AGENT']);

    }

?>

**>>>> PHP\std\_date\_format.php**

<?php

**function** std\_date\_format(*$str\_date*)

    {

        return date("d/m/Y", strtotime(*$str\_date*));

    }

?>

**>>>> PHP\validate\_email.php**

<?php

**function** validate\_email(*$email*)

    {

*// testing valid email syntax*

        if(!filter\_var(*$email*, FILTER\_VALIDATE\_EMAIL))

        {

            return false;

        }

*// extracting domain name*

*$domain* = explode('@', *$email*)[1];

*/\**

*check if the domain has any dns record, if it has then it's*

*more probable that the email is valid*

*\*/*

        return checkdnsrr(*$domain*);

*/\**

*these checks are not enough, to make it water tight we need*

*to send a varification email*

*\*/*

    }

?>

**>>>> login.php**

<?php

*// check if any session exists*

    session\_start();

    if(!empty(*$\_SESSION*))

    {

*// if non empty session found head back to home or index*

        header("location: index.php");

    }

*// destroy any session that might get created here*

    session\_destroy();

    if(isset(*$\_POST*['submit']))

    {

*// form has been submitted*

*// linking with database*

        include "PHP/config.php";

        include "PHP/mysql\_sanitize\_input.php";

*$email* = strtolower(mysql\_sanitize\_input(*$link*, *$\_POST*['email']));

*$pass* = *$\_POST*['password'];

*$rank* = mysql\_sanitize\_input(*$link*, *$\_POST*['ranks']);

        if(*$rank* == 0)

        {

*// wants to login as an employee (every employee is allowed)*

*$query* = "select eid, name, ranks, password from employee natural join login WHERE email = '*$email*'";

        }

        else

        {

*// wants to login with his own rank (a manager can't enter as HR and vice versa)*

*$query* = "select eid, name, ranks, password from employee natural join login WHERE email = '*$email*' AND ranks = *$rank*";

        }

*$result* = *$link* -> query(*$query*);

        if(*$result* === false)

        {

            die("Form submission failure, head back to <a href = 'index.php'> Home </a>");

        }

*// getting first record*

*$arr* = *$result* -> fetch\_assoc();

        if(*$result* -> *num\_rows* === 1 && password\_verify(*$pass*, *$arr*['password']))

        {

*/\**

*an entry is found in employee and login tabel so we start a session and store*

*user name, user id*

*\*/*

            include "PHP/start\_secure\_session.php";

            include "PHP/emp\_ranking\_system.php";

            start\_secure\_session();

*// getting name of the directory where the section for this user is stored*

*$user\_section* = get\_rank\_section(*$rank*);

*$user* = strtoupper(*$user\_section*);

*$\_SESSION*[*$user* . "\_RANK"] = *$arr*['ranks'];

*$\_SESSION*[*$user* . "\_NAME"] = *$arr*['name'];

*$\_SESSION*[*$user* . "\_ID"] = *$arr*['eid'];

*$link* -> close();

            header("location: " . *$user\_section* . "/index.php");    *// entering user section*

        }

        else

        {

*/\**

*no entry is found hence with proper password, this login process fails, to indicate this*

*we redirect to this page and set the get variable fail*

*which is checked to indicate failure*

*\*/*

*$link* -> close();

            header("location: login.php?fail=true");

        }

    }

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>login</title>

        <style>

            @import url("CSS/general styles.css");

            @import url("CSS/form styles.css");

            label

            {

                margin-top: -0.5em;

            }

        </style>

    </head>

    <body>

*<!--*

*We don't keep any return to home button here to discourage*

*user from accidentally return to home page, I want login*

*to his profile and play*

*-->*

        <header></header>

        <main>

        <form method = "post" action = "login.php">

        <ul class = "main\_box nice\_shadow">

*<!-- indicates failed to login -->*

            <?php if(isset(*$\_GET*['fail'])) :?>

                <li>

                    <div class = "error message">

                        Invalid credentials. Try again.

                    </div>

                </li>

            <?php endif; ?>

*<!-- Maxlength is set according to size of uname field in login table -->*

            <li>

                <label> Email <input type = "email" name = "email" maxlength = 50 required> </label>

            </li>

            <li>

                <label> Password (For Profile) <input type = "password" name = "password" maxlength = 10 required> </label>

            </li>

            <li>

                <label>

                    Rank

                    <select name = "ranks">

                        <option value = 0> Employee </option>

                        <option value = 1> Manager1 </option>

                        <option value = 2> Manager2 </option>

                        <option value = 3> HR </option>

                    </select>

                </label>

            </li>

            <li>

                <input class = "button" type = "submit" name = "submit" value = "Login">

            </li>

        </ul>

        </form>

        </main>

        <footer></footer>

    </body>

</html>

**>>>> employee\index.php (employee dash board)**

<?php

*/\**

*check if it's valid emp session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_emp\_session.php";

    include "../PHP/config.php";

    include "../PHP/leave\_days.php";

    include "../PHP/std\_date\_format.php";

    include "../PHP/leave\_request\_status.php";

*$eid* = *$\_SESSION*['EMPLOYEE\_ID'];

*$result* = *$link* -> query("SELECT lid, type, days FROM leave\_rule");

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>Employee Index</title>

        <style>

            @import url("../CSS/general styles.css");

            @import url("../CSS/form styles.css");

            @import url("../CSS/header styles.css");

            @import url("../CSS/list styles.css");

            @import url("../CSS/dashboard styles.css");

        </style>

    </head>

    <body>

        <header>

            <?php include "header.php";?>

        </header>

        <main>

        <div class = "main\_box nice\_shadow">

*<!--*

*>>>> here we present a dashboard to display how many leave days has*

*been used by the employee, wrt total no. of days available for each*

*leave in leave rules table, we use progress bar for it*

*-->*

        <ul class = "main\_box blue\_box">

*<!-- one iteration of the loop creates the entry for one leave in the dashboard -->*

            <?php for(;*$row* = *$result* -> fetch\_assoc();):?>

            <li>

*<!-- from eid and lid (of a leave) we calculate no. of leave days used by the employee -->*

                <?php *$used\_days* = used\_leave\_days(*$link*, *$eid*, *$row*['lid'])?>

*<!--*

*little bit of color coding used here:*

*red shadow -> all leave days are spent*

*green shadow -> not all leave days are spent*

*-->*

                <div class = "message dashboard\_menu <?php echo (*$used\_days* >= *$row*['days']) ? "redbutton" : "greenbutton"?>">

                    <span><?php echo *$row*['type']?></span>

                    <span><progress max = '<?php echo *$row*['days']?>' value = '<?php echo *$used\_days*?>'></progress></span>

                    <span><?php echo *$used\_days* . "/" . *$row*['days']?></span>

                </div>

            </li>

            <?php endfor?>

        </ul>

*<!-- >>>> two buttons only -->*

        <ul class = "main\_box">

*<!-- view history button -->*

            <li>

                <a href = "view.php?type=0"><button class = 'button d\_box bluebutton'> Leave History </button></a>

            </li>

*<!-- request for leave buttom -->*

            <li>

                <a href = "choose\_leave.php"><button class = 'button d\_box bluebutton'> Leave Request </button></a>

            </li>

        </ul>

*<!--*

*>>>> Now we display all the \*\* waiting leave requests \*\* and also*

*add options to delete them and view supporting doc*

*we use blue shadow for waiting list as per our color convension*

*-->*

        <?php

*// getting waiting ones out of database*

*$query* = "SELECT lid, lrid, type, start\_date, end\_date, reason, need\_doc FROM leave\_request NATURAL JOIN leave\_rule WHERE eid = *$eid* AND mg2\_consent is NULL order by lrid desc";

*$result* = *$link* -> query(*$query*);

*$nav\_index* = 0;

        ?>

        <?php for(;*$row* = *$result* -> fetch\_assoc(); *$nav\_index*++): */\* index of the record read \*/*?>

        <?php

*$lid* = *$row*['lid'];

*$lrid* = *$row*['lrid'];

        ?>

*<!-- navid is used to fecilitate proper return after deletion -->*

        <ul id = "<?php echo "navid*$nav\_index*"?>" class = "main\_box blue\_box">

*<!-- leave type with delete and view doc button -->*

            <li>

*<!-- mark latest leave request with red -->*

                <div class = "<?php echo ((*$nav\_index* === 0) ? "message redbutton" : "message")?> <?php echo get\_status\_color('') . "\_text"?>">

*<!-- delete button -->*

                    <a href = "<?php echo "delete.php?navid=*$nav\_index*&lid=*$lid*&lrid=*$lrid*"?>">&#10006;</a>

*<!-- leave type -->*

                    <b><?php echo *$row*['type']?></b>

*<!-- create view doc button only if this leave needs doc -->*

                    <?php if(*$row*['need\_doc'] == true):?>

                        <a href = "<?php echo "support\_doc.php?lid=*$lid*&lrid=*$lrid*"?>">&#128209;</a>

                    <?php endif?>

                </div>

            </li>

*<!-- start and end dates of the leave -->*

            <li>

                <div class = "message">

                    <?php

                        if(*$row*['start\_date'] == *$row*['end\_date'])

                        {

                            echo std\_date\_format(*$row*['start\_date']);

                        }

                        else

                        {

                            echo std\_date\_format(*$row*['start\_date']) . " &#8594; " . std\_date\_format(*$row*['end\_date']);

                        }

                    ?>

                </div>

            </li>

*<!-- no. of days and emoji to indicate status -->*

            <li>

                <div class = "message">

                    <?php echo count\_leave\_days(*$row*['start\_date'], *$row*['end\_date']) . ((*$row*['start\_date'] == *$row*['end\_date']) ? " Day" : " Days")?> <?php echo get\_status\_emoji('')?>

                </div>

            </li>

*<!-- reason -->*

            <li>

                <div class = "message">

                    <?php echo *$row*['reason']?>

                </div>

            </li>

        </ul>

        <?php endfor?>

        <?php if(*$result* -> *num\_rows*):?>

*<!-- A link to the list of all leaves, don't display it no waiting leaves exists -->*

            <ul class = "main\_box">

                <li>

                    <a href = "view.php?type=0"><button class = "button bluebutton"><b>All Leaves</b></button></a>

                </li>

            </ul>

        <?php endif?>

*<!--*

*>>>> Now we display last 3 \*\* approved leave requests \*\**

*we use green shadow for approved list as per our color convension*

*-->*

        <?php

*// getting last 3 approved ones out of database*

*$query* = "SELECT type, start\_date, end\_date, need\_doc FROM leave\_request NATURAL JOIN leave\_rule WHERE eid = *$eid* AND mg2\_consent = 'A' order by lrid desc limit 3";

*$result* = *$link* -> query(*$query*);

        ?>

        <?php for(;*$row* = *$result* -> fetch\_assoc();):?>

        <ul class = "main\_box green\_box">

*<!-- leave type -->*

            <li>

                <div class = "message <?php echo get\_status\_color('A') . "\_text"?>">

                    <b><?php echo *$row*['type']?></b>

                </div>

            </li>

*<!-- dates -->*

            <li>

                <div class = "message">

                    <?php

                        if(*$row*['start\_date'] == *$row*['end\_date'])

                        {

                            echo std\_date\_format(*$row*['start\_date']);

                        }

                        else

                        {

                            echo std\_date\_format(*$row*['start\_date']) . " &#8594; " . std\_date\_format(*$row*['end\_date']);

                        }

                    ?>

                </div>

            </li>

*<!-- days and status emoji -->*

            <li>

                <div class = "message">

                    <?php echo count\_leave\_days(*$row*['start\_date'], *$row*['end\_date']) . ((*$row*['start\_date'] == *$row*['end\_date']) ? " Day" : " Days")?> <?php echo get\_status\_emoji('A')?>

                </div>

            </li>

        </ul>

        <?php endfor?>

        <?php if(*$result* -> *num\_rows*):?>

*<!-- A link to the list of all approved leaves, don't display it no approved leaves exists -->*

            <ul class = "main\_box">

                <li>

                    <a href = "view.php?type=1"><button class = "button greenbutton"><b>All Approved</b></button></a>

                </li>

            </ul>

        <?php endif?>

*<!--*

*>>>> Now we display last 3 \*\* declined leave requests \*\**

*we use green shadow for approved list as per our color convension*

*-->*

        <?php

*// getting last 3 declined ones out of database*

*$query* = "SELECT type, start\_date, end\_date, need\_doc FROM leave\_request NATURAL JOIN leave\_rule WHERE eid = *$eid* AND mg2\_consent <> 'A' AND mg2\_consent is NOT NULL order by lrid desc limit 3";

*$result* = *$link* -> query(*$query*);

*$link* -> close();

        ?>

        <?php for(;*$row* = *$result* -> fetch\_assoc();):?>

        <ul class = "main\_box red\_box">

*<!-- leave type -->*

            <li>

                <div class = "message <?php echo get\_status\_color('abc') . "\_text"?>">

                    <b><?php echo *$row*['type']?></b>

                </div>

            </li>

*<!-- dates -->*

            <li>

                <div class = "message">

                    <?php

                        if(*$row*['start\_date'] == *$row*['end\_date'])

                        {

                            echo std\_date\_format(*$row*['start\_date']);

                        }

                        else

                        {

                            echo std\_date\_format(*$row*['start\_date']) . " &#8594; " . std\_date\_format(*$row*['end\_date']);

                        }

                    ?>

                </div>

            </li>

*<!-- days and status emoji -->*

            <li>

                <div class = "message">

                    <?php echo count\_leave\_days(*$row*['start\_date'], *$row*['end\_date']) . ((*$row*['start\_date'] == *$row*['end\_date']) ? " Day" : " Days")?> <?php echo get\_status\_emoji('abc')?>

                </div>

            </li>

        </ul>

        <?php endfor?>

        <?php if(*$result* -> *num\_rows*):?>

*<!-- A link to the list of all declined leaves, don't display it no declined leaves exists -->*

            <ul class = "main\_box">

                <li>

                    <a href = "view.php?type=2"><button class = "button redbutton"><b>All Declined</b></button></a>

                </li>

            </ul>

        <?php endif?>

        </div>

        </main>

        <footer></footer>

        <script src="../JS/scroll\_back.js"></script>

    </body>

</html>

**>>>> employee\delete.php (deletes a waiting leave request)**

<?php

*/\**

*check if it's valid emp session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_emp\_session.php";

    if(isset(*$\_GET*["lrid"]) && isset(*$\_GET*["lid"]) && isset(*$\_GET*["navid"]))

    {

        include "../PHP/config.php";    *// connect to database*

        include "../PHP/mysql\_sanitize\_input.php";

*// DELETE request from emp index*

*$eid* = *$\_SESSION*['EMPLOYEE\_ID'];

*$lrid* = mysql\_sanitize\_input(*$link*, *$\_GET*["lrid"]);

*$lid* = mysql\_sanitize\_input(*$link*, *$\_GET*["lid"]);

*$nav* = *$\_GET*["navid"];

*// deleting leave requests, only ones not seen by top manager*

*$query* = "delete from leave\_request where eid = *$eid* and lid = *$lid* and lrid = *$lrid* and mg2\_consent is null";

*/\**

*deletion failure will cause 404 error*

*(this is very unlikely to happen)*

*\*/*

        if(*$link* -> query(*$query*) == false)

        {

*// Deletion failure*

            header("HTTP/1.0 404 Not Found", true, 404);

            exit();

        }

*$link* -> close();

*// redirect to the leave request right before the deleted quiz in index page*

        header("location: index.php#navid" . (*$nav* - 1));

    }

    else

    {

*// no URL parameters provided*

        header("location: index.php");

    }

?>

**>>>> employee\leave\_request.php**

<?php

*/\**

*check if it's valid emp session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_emp\_session.php";

    include "../PHP/config.php";

    include "../PHP/mysql\_sanitize\_input.php";

    include "../PHP/leave\_days.php";

    include "../PHP/message\_box.php";

*$max\_size* = 1;  *// 1 MB*

*$files\_accepted* = ".jpg, .jpeg, .png, .pdf";

    if(isset(*$\_GET*['lid']))

    {

*/\**

*this file receives 'lid' of a leave via get method*

*and prepares a form accordingly to accept a leave*

*request*

*\*/*

*$lid* = mysql\_sanitize\_input(*$link*, *$\_GET*['lid']);

*$eid* = *$\_SESSION*['EMPLOYEE\_ID'];

*// >>>> check if this leave exists and if we need supporting document for this leave*

*$result* = *$link* -> query("SELECT type, need\_doc FROM leave\_rule WHERE lid = *$lid*");

        if(*$result* -> *num\_rows* == 0)

        {

*// leave not exist hence, page not found*

            header("HTTP/1.0 404 Not Found", true, 404);

            exit();

        }

*$tuple* = *$result* -> fetch\_assoc();

*// supporting doc and type of leave*

*$need\_doc* = *$tuple*['need\_doc'];

*$leave\_name* = *$tuple*['type'];

*// >>>> check if the employee can take this leave or not*

*$days\_avail* = leave\_days\_remaining(*$link*, *$eid*, *$lid*);

        if(*$days\_avail* <= 0)

        {

*// cannot take this leave*

            message\_box("You can't apply for this leave", "employee/choose\_leave.php", true);

        }

*$link* -> close();

*// now  we prepare the form*

    }

    else if(isset(*$\_POST*['submit']))

    {

*// backup input (it will be used to retain the inputs in case of a failure)*

*$\_SESSION*['inputs'] = *$\_POST*;

*/\**

*After each successful submission we redirect to the same file to display*

*the error message.*

*GET variables are used to convey the error messages.*

*Frontend (form) gets modified accordingly to those GET variables to display*

*the errors in input data*

*\*/*

*$lid* = mysql\_sanitize\_input(*$link*, *$\_POST*['lid']);

*$eid* = *$\_SESSION*['EMPLOYEE\_ID'];

*// >>>> checking the input dates*

*$start\_date* = mysql\_sanitize\_input(*$link*, *$\_POST*['start\_date']);

*$end\_date* = mysql\_sanitize\_input(*$link*, *$\_POST*['end\_date']);

*// check if the dates are already booked or not (i.e., requested or approved)*

*$query* = "SELECT lrid FROM leave\_request WHERE eid = *$eid* AND (mg2\_consent is NULL OR mg2\_consent = 'A') AND end\_date >= '*$start\_date*' AND start\_date <= '*$end\_date*'";

*$result* = *$link* -> query(*$query*);

        if(*$result* === false)

        {

            message\_box("Failed to add new leave request (check the dates)", "employee/leave\_request.php?lid=*$lid*", true);

        }

        if(*$result* -> *num\_rows* > 0)

        {

*// the dates are already booked*

*$get* = "days\_booked=false";

        }

        else

        {

*$requested\_days* = count\_leave\_days(*$start\_date*, *$end\_date*);

*// check if start > end date or requested\_days are more than remaining days*

            if(*$requested\_days* <= 0 || *$requested\_days* > leave\_days\_remaining(*$link*, *$eid*, *$lid*))

            {

*// dates are not valid*

*$get* = "days\_valid=false";

            }

        }

*// >>>> checking the doc*

        if(isset(*$\_FILES*['document']))

        {

*// getting file data*

*$fname* = *$\_FILES*['document']['name'];

*$fsize* = *$\_FILES*['document']['size'];

*$error* = *$\_FILES*['document']['error'];

*$tname* = *$\_FILES*['document']['tmp\_name'];   *// imp*

*// getting file type (extension)*

*$ftype* = explode('.', *$fname*);

*$ftype* = strtolower(end(*$ftype*));   *// imp*

*// checks*

            if(*$error* !== 0)

            {

*$err* = "file\_transmission\_valid=false";

            }

            else if(!in\_array(".*$ftype*", explode(', ', *$files\_accepted*)))

            {

*$err* = "file\_type\_valid=false";

            }

            else if(*$fsize* > (*$max\_size* \* 1024 \* 1024))

            {

*$err* = "file\_size\_valid=false";

            }

            if(isset(*$err*))

            {

*// if $err exists (error detected) we add it to $get*

                if(isset(*$get*))

                {

*$get* = "*$get*&*$err*";

                }

                else

                {

*$get* = *$err*;

                }

            }

        }

*/\**

*>>>> if $get is created there is a problem and we reload the file by*

*rediecting to itself, none of the input data are stored*

*\*/*

        if(isset(*$get*))

        {

*$link* -> close();

            header("location: leave\_request.php?lid=*$lid*&*$get*");

            exit();

        }

*/\**

*>>>> no problem detected, hence we receive rest of the inputs*

*and load them into leave request table*

*\*/*

*// >>>> reason for leave*

*$reason* = mysql\_sanitize\_input(*$link*, *$\_POST*['reason']);

*// manager 1 doesn't need mg1\_consent and manager 2 doesn't need mg1\_consent or mg2\_consent*

*$mng\_attrib* = '';

*$mng\_value* = '';

        if(*$\_SESSION*['EMPLOYEE\_RANK'] == 1) *// manager1*

        {

*$mng\_attrib* = ", mg1\_consent";

*$mng\_value* = ", 'NA'";

        }

        elseif(*$\_SESSION*['EMPLOYEE\_RANK'] == 2) *// manager2*

        {

*$mng\_attrib* = ", mg1\_consent, mg2\_consent";

*$mng\_value* = ", 'NA', 'A'";

        }

*// checking if we need to store the document or not*

        if(isset(*$tname*))   *// $tname -> temporary location of the doc*

        {

*// preparing the document for storage*

*$fcontent* = *$link* -> real\_escape\_string(file\_get\_contents(*$tname*));

*$query* = "INSERT INTO leave\_request(eid, lid, start\_date, end\_date, reason, support\_doc, ftype*$mng\_attrib*) VALUES(*$eid*, *$lid*, '*$start\_date*', '*$end\_date*', '*$reason*', '*$fcontent*', '*$ftype*'*$mng\_value*)";

        }

        else

        {

*$query* = "INSERT INTO leave\_request(eid, lid, start\_date, end\_date, reason*$mng\_attrib*) VALUES(*$eid*, *$lid*, '*$start\_date*', '*$end\_date*', '*$reason*'*$mng\_value*)";

        }

*// fail check*

        if(*$link* -> query(*$query*) === false)

        {

            message\_box("Failed to add new leave request (check lenghts of the inputs)", "employee/leave\_request.php?lid=*$lid*", true);

        }

*$link* -> close();

*// success hence destroy input backup and redirect to employee index*

        unset(*$\_SESSION*['inputs']);

*// new waiting is kept on top, hence, "#navid0"*

        header("location: index.php#navid0");

    }

    else

    {

        header("location: index.php");

    }

*// checking any backed up inputs can be found or not*

    if(isset(*$\_SESSION*['inputs']))

    {

*$start\_date* = *$\_SESSION*['inputs']['start\_date'];

*$end\_date* = *$\_SESSION*['inputs']['end\_date'];

*$reason* = *$\_SESSION*['inputs']['reason'];

        unset(*$\_SESSION*['inputs']);

    }

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>Leave Request</title>

        <style>

            @import url("../CSS/general styles.css");

            @import url("../CSS/form styles.css");

            @import url("../CSS/header styles.css");

        </style>

    </head>

    <body>

        <header>

            <?php include "header.php";?>

        </header>

        <main>

*<!-- for forms sending images or documents we need, 'enctype = "multipart/form-data"' -->*

        <form method = "post" action = "leave\_request.php" enctype = "multipart/form-data">

*<!-- this is the only way to let backend know the 'lid' -->*

        <input name = "lid" type = hidden value = <?php echo *$lid*?> required>

        <ul class = "main\_box nice\_shadow">

*<!-- Displaying the leave type and no. of days available (user friendly) -->*

            <li>

                <div class = "heading">

                    <?php echo "<h1>*$leave\_name*</h1><em>\* Only <b>*$days\_avail* days</b> available</em>"?>

                </div>

            </li>

*<!-- date inputs -->*

            <li>

                <label> Start Date <input name = "start\_date" type = date required <?php if(isset(*$start\_date*)) echo "value = '*$start\_date*'";?>> </label>

            </li>

            <li>

                <label> End Date <input name = "end\_date" type = date required <?php if(isset(*$end\_date*)) echo "value = '*$end\_date*'";?>> </label>

            </li>

*<!-- date input errors -->*

            <?php if(isset(*$\_GET*['days\_booked'])) :?>

*<!-- the dates entered are booked -->*

                <li>

                    <div class = "error message">

                        These days are already used

                    </div>

                </li>

            <?php endif; ?>

            <?php if(isset(*$\_GET*['days\_valid'])) :?>

*<!-- the dates entered are invalid -->*

                <li>

                    <div class = "error message">

                        Invalid dates

                    </div>

                </li>

            <?php endif; ?>

*<!-- reason -->*

            <li>

                <label> Reason <textarea name = "reason" maxlength = 500 required><?php if(isset(*$reason*)) echo "*$reason*";?></textarea></label>

            </li>

*<!-- support doc input (only if required by the leave rule) -->*

            <?php if(*$need\_doc*):?>

*<!-- leave rule need support doc -->*

                <li>

                    <label> Supporting Doc

                        <div class = "file">

                            <input type = "hidden" name = "MAX\_FILE\_SIZE" value = <?php echo (*$max\_size* \* 1024 \* 1024)?>>

                            <input type = "file" name = "document" accept = "<?php echo *$files\_accepted*?>" required>

                        </div>

                    </label>

                </li>

*<!-- max size message -->*

                <li>

                    <div class = "error message">

                        <em><?php echo *$files\_accepted*?></em> accepted, <?php echo *$max\_size*?> MB or less

                    </div>

                </li>

*<!-- doc input errors -->*

*<!-- transmission error (i.e., file not sent properly) -->*

                <?php if(isset(*$\_GET*['file\_transmission\_valid'])) :?>

                    <li>

                        <div class = "error message">

                            Can't load the file, it may be too large

                        </div>

                    </li>

                <?php endif; ?>

*<!-- file type error -->*

                <?php if(isset(*$\_GET*['file\_type\_valid'])) :?>

                    <li>

                        <div class = "error message">

                            Invalid file type

                        </div>

                    </li>

                <?php endif; ?>

*<!-- file size error -->*

                <?php if(isset(*$\_GET*['file\_size\_valid'])) :?>

                    <li>

                        <div class = "error message">

                            File too large

                        </div>

                    </li>

                <?php endif; ?>

            <?php else:?>

*<!-- leave rule doesn't need support doc -->*

                <li>

                    <div class = "info message">

                        No supporting document needed

                    </div>

                </li>

            <?php endif?>

            <li>

                <input class = "button" type = "submit" name = "submit" value = "Submit Application">

            </li>

        </ul>

        </form>

        </main>

        <footer></footer>

    </body>

</html>

**>>>> employee\support\_doc.php**

<?php

    include\_once "../PHP/check\_emp\_session.php";

*// display support document for certain leave request*

    if(isset(*$\_GET*["lrid"]) && isset(*$\_GET*["lid"]))

    {

        include\_once "../php/config.php";

        include\_once "../PHP/mysql\_sanitize\_input.php";

        include\_once "../PHP/doc\_display.php";

*$eid* = *$\_SESSION*['EMPLOYEE\_ID'];

*$lrid* = mysql\_sanitize\_input(*$link*, *$\_GET*["lrid"]);

*$lid* = mysql\_sanitize\_input(*$link*, *$\_GET*["lid"]);

*$query* = "select ftype, support\_doc from leave\_request where eid = *$eid* and lid = *$lid* and lrid = *$lrid* and support\_doc is not null and ftype is not null";

*// read the file contents corresponding to the id from database*

*$result* = *$link* -> query(*$query*);

        if(*$result* === false || *$result* -> *num\_rows* == 0)

        {

*// Document not found hence page not found error*

            header("HTTP/1.0 404 Not Found", true, 404);

            exit();

        }

*$link* -> close();

*$row* = *$result* -> fetch\_assoc();

*// sending the file content and type to a function to print or display it*

        doc\_display(*$row*['support\_doc'], *$row*['ftype']);

    }

    else

    {

        header("location: index.php");

    }

?>

**>>>> employee\view.php**

Very similar to **>>>> manager\view.php** only difference is that here the employee details are not shown

**>>>> manager\index.php (dashboard for both manager 1 and 2)**

<?php

*/\**

*check if it's valid mg session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_mng\_session.php";

    include "../PHP/config.php";

    include "../PHP/leave\_days.php";

    include "../PHP/std\_date\_format.php";

    include "../PHP/emp\_ranking\_system.php";

*$mrank* = *$\_SESSION*["MANAGER\_RANK"];

*/\**

*manager1 can see the tuples where mg1\_consent and mg2\_consent both are null*

*manager2 can see the tuples where mg1\_consent is not null but mg2\_consent is null*

*\*/*

*$mg1\_consent* = (*$mrank* == 1) ? "IS NULL" : "IS NOT NULL";

*$query* = "SELECT eid, lid, lrid, name, type, start\_date, end\_date, days FROM leave\_request NATURAL JOIN leave\_rule NATURAL JOIN employee WHERE mg1\_consent *$mg1\_consent* AND mg2\_consent IS NULL";

*$result* = *$link* -> query(*$query*);

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>Manager Index</title>

        <style>

            @import url("../CSS/general styles.css");

            @import url("../CSS/form styles.css");

            @import url("../CSS/header styles.css");

            @import url("../CSS/list styles.css");

            @import url("../CSS/dashboard styles.css");

*/\**

*here span will contain important text information hence*

*getting nice border around it*

*\*/*

            .dashboard\_menu > span

            {

                border-radius: *inherit*;

                margin: .2em;

*/\* box-shadow: 1px 1px 5px -1px rgba(0, 0, 0, 0.1); \*/*

            }

        </style>

    </head>

    <body>

        <header>

            <?php include "header.php";?>

        </header>

        <main>

*<!--*

*>>>> here we present a dashboard to display how many leave days has*

*been used by the employee, wrt total no. of days available for each*

*leave in leave rules table, we use progress bar for it*

*-->*

        <ul class = "main\_box nice\_shadow">

*<!-- checking how many leave requests we have -->*

            <?php if(*$result* -> *num\_rows* == 0):?>

                <li>

                    <div class = "message redbutton"><center>Empty List &#128526;</center></div>

                </li>

            <?php endif?>

*<!-- one iteration of the loop creates the entry for one leave in the dashboard -->*

            <?php for(;*$row* = *$result* -> fetch\_assoc();):?>

*<!-- from eid and lid (of a leave) we calculate no. of leave days (approved) already used by the employee -->*

            <?php

*$eid* = *$row*['eid'];

*$lid* = *$row*['lid'];

*$lrid* = *$row*['lrid'];

*$approved\_days* = approved\_leave\_days(*$link*, *$eid*, *$lid*);

            ?>

            <li id = "<?php echo "lrid*$lrid*"?>">

*<!--*

*here we show emp details and leave stats*

*-->*

                <div class = "message dashboard\_menu <?php echo (*$approved\_days* == *$row*['days']) ? "redbutton" : "bluebutton"?>">

*<!-- emp name -->*

                    <span>

*<!-- button to display leave history of the employee -->*

                        <a href = "<?php echo "view.php?eid=*$eid*"?>"><?php echo *$row*['name']?></a>

*<!--<?php* echo *$row[*'name'*]?>-->*

                    </span>

*<!--*

*leave stats as a progress bar and a hidden button to display complete*

*leave stats of this emp in brief with progress bars*

*-->*

                    <span><a href = "<?php echo "stats.php?eid=*$eid*"?>"><progress max = '<?php echo *$row*['days']?>' value = '<?php echo *$approved\_days*?>'></progress></a></span>

*<!-- no. of leaves already approved to this emp -->*

                    <span>

                        <?php echo *$approved\_days* . "/" . *$row*['days']?> Taken

                    </span>

                </div>

            </li>

*<!--*

*leave type, reason and consent or approval*

*-->*

            <li>

                <div class = "message dashboard\_menu <?php echo (*$approved\_days* == *$row*['days']) ? "redbutton" : "bluebutton"?>">

*<!-- leave type and no. of days -->*

                    <span>

                        <?php echo *$row*['type']?>

                        <?php echo count\_leave\_days(*$row*['start\_date'], *$row*['end\_date']) . ((*$row*['start\_date'] == *$row*['end\_date']) ? " Day" : " Days")?>

*<!-- button to display leave reason and mg1 consent -->*

                        <a href = "<?php echo "consent.php?lrid=*$lrid*"?>">&#128209;</a>

                    </span>

*<!-- start and end date -->*

                    <span>

                        <?php

                            if(*$row*['start\_date'] == *$row*['end\_date'])

                            {

                                echo std\_date\_format(*$row*['start\_date']);

                            }

                            else

                            {

                                echo std\_date\_format(*$row*['start\_date']) . " &#8594; " . std\_date\_format(*$row*['end\_date']);

                            }

                        ?>

                    </span>

*<!-- approval and consent buttons -->*

                    <?php if(*$mrank* == 1):?>

*<!-- for mg1 -->*

                        <span>Comment <a href = "<?php echo "comment.php?lrid=*$lrid*"?>">&#128221;</a> Please</span>

                    <?php else:?>

*<!-- for mg2 -->*

                        <span>

                            Approve <a href = "<?php echo "approve.php?lrid=*$lrid*"?>">&#10004;</a>

                            Decline <a href = "<?php echo "comment.php?lrid=*$lrid*"?>">&#128221;</a>

                        </span>

                    <?php endif?>

                </div>

            </li>

            <?php endfor?>

        </ul>

        <?php *$link* -> close();?>

        </main>

        <footer></footer>

        <script src="../JS/scroll\_back.js"></script>

    </body>

</html>

**>>>> manager\view.php**

<?php

*/\**

*check if it's valid mg session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_mng\_session.php";

    if(!isset(*$\_GET*["eid"]))

    {

*// if eid URL parameter is not set go to index*

        header("location: index.php");

    }

    include "../PHP/config.php";    *// connect to database*

    include "../PHP/mysql\_sanitize\_input.php";

    include "../PHP/leave\_days.php";

    include "../PHP/std\_date\_format.php";

    include "../PHP/leave\_request\_status.php";

*$eid* = mysql\_sanitize\_input(*$link*, *$\_GET*["eid"]);

*// all leave history desc order*

*$query* = "SELECT lrid, type, start\_date, end\_date, reason, mg2\_consent FROM leave\_request NATURAL JOIN leave\_rule WHERE eid = *$eid* order by lrid desc";

*$result* = *$link* -> query(*$query*);

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>View</title>

        <style>

            @import url("../CSS/general styles.css");

            @import url("../CSS/form styles.css");

            @import url("../CSS/header styles.css");

            @import url("../CSS/list styles.css");

        </style>

    </head>

    <body>

        <header id = 'first'>

            <?php include "header.php";?>

        </header>

        <main>

*<!-- Displaying employee details -->*

        <?php if(*$result* -> *num\_rows* != 0):?>

            <?php

*// extracting employee data*

*$query* = "SELECT \* FROM employee WHERE eid = *$eid*";

*$details* = *$link* -> query(*$query*);

                if(*$details* -> *num\_rows* != 1)

                {

*// If no. of records is not 1 something is hrribly wrong, hence page not found error*

                    header("HTTP/1.0 404 Not Found", true, 404);

                    exit();

                }

*$row* = *$details* -> fetch\_assoc();

            ?>

*<!-- displaying emp data -->*

            <ul class = "main\_box blue\_box">

*<!-- the emp attributes -->*

                <li>

                    <div class = "message">

                        <?php echo *$row*["name"]?>

                    </div>

                </li>

                <li>

                    <div class = "message">

                        <a href = "<?php echo "mailto:" . *$row*["email"]?>"><?php echo *$row*["email"]?></a>

                    </div>

                </li>

                <li>

                    <div class = "message">

                        <?php echo *$row*["phone"]?>

                    </div>

                </li>

                <li>

                    <div class = "message">

                        <?php

                            include\_once "../PHP/emp\_ranking\_system.php";

                            echo get\_rank(*$row*['ranks']);

                        ?>

                    </div>

                </li>

            </ul>

        <?php endif?>

*<!-- Displaying the Leave history -->*

*<!-- 10 leaves between 2 navigation boxes -->*

        <?php *$nav\_interval* = 10; *$nav\_index* = 0;?>

*<!-- getting the tuples in emp rule table -->*

        <?php for(;*$row* = *$result* -> fetch\_assoc(); *$nav\_index*++): */\* index of the record read \*/*?>

*<!-- navigation box to reduce time to nevigate entire list -->*

        <?php if((*$nav\_index* % *$nav\_interval*) === 0):?>

            <ul id = <?php echo "menu*$nav\_index*";?> class = "main\_box">

*<!-- for first menu bar these two not necessary -->*

                <?php if(*$nav\_index* != 0):?>

*<!-- go to previous navigation box -->*

                    <li>

                        <a href = "<?php echo "#menu" . (*$nav\_index* - *$nav\_interval*)?>"><button class = 'button'> Previous </button></a>

                    </li>

*<!-- go to top of the page -->*

                    <li>

                        <a href = "#first"><button class = 'button'> Top </button></a>

                    </li>

                <?php endif; ?>

*<!-- go to bottom of the page -->*

                <li>

                    <a href = "#last"><button class = 'button'> Bottom </button></a>

                </li>

*<!-- go to next navigation box -->*

                <?php if(*$nav\_index* + *$nav\_interval* < *$result* -> *num\_rows*):?>

                    <li>

                        <a href = "<?php echo "#menu" . (*$nav\_index* + *$nav\_interval*)?>"><button class = 'button'> Next </button></a>

                    </li>

                <?php else:?>

*<!--*

*for last navigation box next button is same as Bottom button*

*as we keep a last navigation box at the bottom of the page*

*-->*

                    <li>

                        <a href = "<?php echo "#last"?>"><button class = 'button'> Next </button></a>

                    </li>

                <?php endif; ?>

            </ul>

        <?php endif; ?>

*<!-- the record box -->*

        <?php

*$id* = *$row*['lrid'];

        ?>

*<!--*

*We use both types of ids, navid for sequential access, id for direct access of these leaves*

*We use color code to differentiate between different types of leaves (waiting, approved or*

*declined)*

*-->*

        <div id = <?php echo "navid" . *$nav\_index*;?>></div>

        <ul id = <?php echo "id" . *$id*;?> class = "<?php echo "main\_box " . get\_status\_color(*$row*['mg2\_consent']) . "\_box"?>">

*<!-- type of leave requested -->*

            <li>

                <div class = "<?php echo "message " . get\_status\_color(*$row*['mg2\_consent']) . "\_text"?>">

                    <b><?php echo *$row*['type']?></b>

                </div>

            </li>

*<!-- starting and ending days of leave request -->*

            <li>

                <div class = "message">

                    <?php

                        if(*$row*['start\_date'] == *$row*['end\_date'])

                        {

                            echo std\_date\_format(*$row*['start\_date']);

                        }

                        else

                        {

                            echo std\_date\_format(*$row*['start\_date']) . " &#8594; " . std\_date\_format(*$row*['end\_date']);

                        }

                    ?>

                </div>

            </li>

*<!-- No. of leave days and status of leave request -->*

            <li>

                <div class = "message">

                    <?php echo count\_leave\_days(*$row*['start\_date'], *$row*['end\_date']) . ((*$row*['start\_date'] == *$row*['end\_date']) ? " Day" : " Days")?> <?php echo get\_status\_emoji(*$row*['mg2\_consent'])?>

                </div>

            </li>

*<!-- consent of manager 2 and reason -->*

            <?php if(get\_status\_color(*$row*['mg2\_consent']) == 'red'):?>

                <li>

                    <div class = "message">

                        <?php echo *$row*['reason']?>

                        <br>

                        <b><?php echo *$row*['mg2\_consent']?></b>

                    </div>

                </li>

            <?php else:?>

                <li>

                    <div class = "message">

                        <?php echo *$row*['reason']?>

                    </div>

                </li>

            <?php endif?>

        </ul>

        <?php endfor; ?>

*<!-- top and previous button at the end of the page -->*

        <ul class = 'main\_box'>

            <li>

                <a href = "<?php echo "#menu" . (*$nav\_index* - *$nav\_index* % *$nav\_interval*)?>"><button class = 'button'> Previous </button></a>

            </li>

            <li>

                <a href = "#first"><button class = 'button'> Top </button></a>

            </li>

        </ul>

        </main>

        <?php *$link* -> close()?>

        <footer id = "last"></footer>

    </body>

</html>

**>>>> manager\approve.php**

Similar to **>>>> manager\comment.php** only different is that instead of a message a letter ‘A’ is stored as mg2\_consent

**>>>> manager\comment.php**

<?php

*/\**

*check if it's valid mg session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_mng\_session.php";

    include "../PHP/config.php";

    include "../PHP/mysql\_sanitize\_input.php";

*$mrank* = *$\_SESSION*["MANAGER\_RANK"];

    if(isset(*$\_GET*["lrid"]))

    {

*$lrid* = mysql\_sanitize\_input(*$link*, *$\_GET*['lrid']);

*$link* -> close();

    }

    else if(isset(*$\_POST*['submit']))

    {

*// now we store manager consent*

*$lrid* = mysql\_sanitize\_input(*$link*, *$\_POST*['lrid']);

*$comment* = mysql\_sanitize\_input(*$link*, *$\_POST*['comment']);

*$consent* = (*$mrank* == 1) ? "mg1\_consent" : "mg2\_consent";

*/\**

*manager1 can enter his consent if mg1\_consent and mg2\_consent both are null*

*manager2 can enter his consent if mg1\_consent is not null but mg2\_consent is null*

*\*/*

*$mg1\_consent* = (*$mrank* == 1) ? "IS NULL" : "IS NOT NULL";

*$query* = "UPDATE leave\_request SET *$consent* = '*$comment*' WHERE lrid = *$lrid*  AND mg1\_consent *$mg1\_consent* AND mg2\_consent IS NULL";

*// fail check*

        if(*$link* -> query(*$query*) == false)

        {

            die("Form submission failure, head back to <a href = 'index.php'> Home </a>");

        }

*$link* -> close();

        header("location: index.php");

    }

    else

    {

        header("location: index.php");

    }

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>Comments</title>

        <style>

            @import url("../CSS/general styles.css");

            @import url("../CSS/form styles.css");

            @import url("../CSS/header styles.css");

        </style>

    </head>

    <body>

        <header>

            <?php include "header.php";?>

        </header>

        <main>

        <form method = "post" action = "comment.php">

        <input name = "lrid" type = hidden value = <?php echo *$lrid*?> required>

        <ul class = "main\_box nice\_shadow">

*<!-- input the comment or consent -->*

            <li>

                <textarea name = "comment" maxlength = 500 required></textarea>

            </li>

            <li>

                <div class = "error message">

                    Enter your comment carefully, it can't be modified in future.

                </div>

            </li>

            <li>

                <input class = "button" type = "submit" name = "submit" value = "Comment">

            </li>

        </ul>

        </form>

        </main>

        <footer></footer>

    </body>

</html>

**>>>> manager\consent.php**

<?php

*/\**

*check if it's valid mg session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_mng\_session.php";

    include "../PHP/config.php";

    include "../PHP/mysql\_sanitize\_input.php";

    include "../PHP/emp\_ranking\_system.php";

*$mrank* = *$\_SESSION*["MANAGER\_RANK"];

    if(isset(*$\_GET*["lrid"]))

    {

*$lrid* = mysql\_sanitize\_input(*$link*, *$\_GET*['lrid']);

*// collecting consent of junior manager and reason for leave*

*$data* = (*$mrank* == 1) ? "reason" : "reason, mg1\_consent";

*/\**

*manager1 can see only if mg1\_consent and mg2\_consent both are null*

*manager2 can see only if mg1\_consent is not null but mg2\_consent is null*

*\*/*

*$mg1\_consent* = (*$mrank* == 1) ? "IS NULL" : "IS NOT NULL";

*$query* = "SELECT *$data*, need\_doc FROM leave\_request NATURAL JOIN leave\_rule WHERE lrid = *$lrid* AND mg1\_consent *$mg1\_consent* AND mg2\_consent IS NULL";

*$result* = *$link* -> query(*$query*);

*$link* -> close();

*// fail check*

        if(*$result* === false)

        {

            die("Form submission failure, head back to <a href = 'index.php'> Home </a>");

        }

*// Prepare Proper consent string to display*

        if(*$result* -> *num\_rows* != 0)

        {

*$row* = *$result* -> fetch\_assoc();

*$reason* = *$row*["reason"];

            if(*$mrank* == 2)

            {

*// only mg2 sees mg1 consent*

*$consent* = *$row*["mg1\_consent"];

            }

        }

        else

        {

*// nothing to display hence, page not found*

            header("HTTP/1.0 404 Not Found", true, 404);

            exit();

        }

    }

    else

    {

        header("location: index.php");

    }

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>Consent</title>

        <style>

            @import url("../CSS/general styles.css");

            @import url("../CSS/form styles.css");

            @import url("../CSS/header styles.css");

        </style>

    </head>

    <body>

        <header>

            <?php include "header.php";?>

        </header>

        <main>

        <ul class = "main\_box nice\_shadow">

            <li>

                Reason

                <div class = "message">

                    <?php echo *$reason*?>

                    <?php if(*$row*['need\_doc'] == true):?>

                            <a href = "<?php echo "support\_doc.php?lrid=*$lrid*"?>">&#128209;</a>

                    <?php endif?>

                </div>

            </li>

            <?php if(isset(*$consent*)):?>

                <li>

                    <?php echo get\_rank(*$mrank* - 1)?><div class = "message error"><?php echo *$consent*?></div>

                </li>

            <?php endif?>

        </ul>

        </main>

        <footer></footer>

    </body>

</html>

**>>>> hr\emp\_input.php**

<?php

*/\**

*check if it's valid hr session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_hr\_session.php";

    include "../PHP/message\_box.php";

*// Empdetails input*

    if(isset(*$\_POST*['submit']))

    {

        include "../PHP/config.php";

        include "../PHP/mysql\_sanitize\_input.php";

        include "../PHP/validate\_email.php";

        include "../PHP/form\_input\_check.php";

*// backup input (it will be used to retain the inputs in case of a failure)*

*$\_SESSION*['inputs'] = *$\_POST*;

*// here we use get variables to send error messages while redirecting to itself*

*/\**

*After each successful submission we redirect to the same form to display*

*the error or success message. As the inputs submitted here will be stored*

*into the database, redirection is used to ensure that no resubmission*

*error will be generated upon reload or back. (More about this error in the*

*documentation)*

*While redirecting we use get method to send the error or success message.*

*\*/*

*// ------------- Error Detection -------------*

*// >>>> missing input parameter check*

        if(elements\_exist(*$\_POST*, ['name', 'email', 'phone', 'ranks', 'password', 'password\_reenter']))

        {

*// >>>> name check*

*/\* nothimg to check yet \*/*

*// >>>> password check*

*// password length check*

            if(max\_length\_check(*$\_POST*['password'], 10) == false)

            {

*$get* = url\_parameters(*$get*, "pass\_valid=0");

            }

            elseif(*$\_POST*['password'] != *$\_POST*['password\_reenter'])

            {

*// two password are different*

*$get* = url\_parameters(*$get*, "pass\_match=0");

            }

*// >>>> email check*

*// check if the email is new and valid or not*

*$email* = strtolower(mysql\_sanitize\_input(*$link*, *$\_POST*['email']));

*$result* = *$link* -> query("SELECT \* FROM employee WHERE email = '*$email*'");

            if(*$result* -> *num\_rows* > 0)

            {

*// the email is already used*

*$get* = url\_parameters(*$get*, "email\_used=1");

            }

            elseif(!validate\_email(*$email*))

            {

*// the email is not used but not valid*

*$get* = url\_parameters(*$get*, "email\_valid=0");

            }

*// >>>> phone check*

*// check if the phone no. is valid or not*

            if(length\_check(*$\_POST*['phone'], 10) == false)

            {

*// the phone no. is not valid*

*$get* = url\_parameters(*$get*, "phone\_valid=0");

            }

        }

        else

        {

*$get* = "input\_missing=1";

        }

*// if get is created there is a problem and we reload the file*

        if(!isset(*$get*))

        {

*/\**

*no problem detected, hence we receive rest of the inputs*

*and load them into employee and login table*

*\*/*

*$name* = mysql\_sanitize\_input(*$link*, *$\_POST*['name']);

*$phone* = mysql\_sanitize\_input(*$link*, *$\_POST*['phone']);

*$ranks* = mysql\_sanitize\_input(*$link*, *$\_POST*['ranks']);

*$pass* = *$\_POST*['password'];

*$password* = password\_hash(*$pass*, PASSWORD\_DEFAULT);

*$query* = "INSERT INTO employee(name, email, phone, ranks) VALUES('*$name*', '*$email*', *$phone*, *$ranks*);";

*// fail check*

            if(*$link* -> query(*$query*) === false)

            {

                message\_box("Failed to add new employee (check lenghts of the inputs or use different password)", "hr/emp\_input.php", true);

            }

*// collecting his eid*

*$eid* = *$link* -> query("SELECT eid FROM employee WHERE email = '*$email*'") -> fetch\_assoc()['eid'];

*// storing password in login table*

*$query* = "INSERT INTO login(eid, password) VALUES(*$eid*, '*$password*');";

*// fail check*

            if(*$link* -> query(*$query*) === false)

            {

                message\_box("Failed to add new employee (check lenghts of the inputs or use different password)", "hr/emp\_input.php", true);

            }

*$get* = "success=1";

*// inputs entered successfully hence destroy backup*

            unset(*$\_SESSION*['inputs']);

        }

*$link* -> close();

*$get* = "?*$get*";

    }

    else

    {

*$get* = '';

    }

*// add self redirect once*

*/\**

*redirecting to itself once to:*

*clear the past inputs to avoid resubmission of same data when*

*user returns back to previous input form*

*and to avoid form resubmission error on reload (more about this*

*in the documentation)*

*\*/*

    include "../PHP/self\_redirect\_once.php";

    self\_redirect\_once(*$get*);

*// checking any backed up inputs can be found or not*

    if(isset(*$\_SESSION*['inputs']))

    {

*$name* = *$\_SESSION*['inputs']['name'];

*$email* = *$\_SESSION*['inputs']['email'];

*$phone* = *$\_SESSION*['inputs']['phone'];

*$ranks* = *$\_SESSION*['inputs']['ranks'];

*$password* = *$\_SESSION*['inputs']['password'];

*$password\_reenter* = *$\_SESSION*['inputs']['password\_reenter'];

        unset(*$\_SESSION*['inputs']);

    }

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>emp\_input</title>

        <style>

            @import url("../CSS/general styles.css");

            @import url("../CSS/form styles.css");

            @import url("../CSS/header styles.css");

        </style>

    </head>

    <body>

        <header>

            <?php include "header.php";?>

        </header>

        <main>

        <form method = "post" action = "emp\_input.php">

        <ul class = "main\_box nice\_shadow">

*<!-- Indicate successfull Registration creation -->*

            <?php if(isset(*$\_GET*['success'])) :?>

                <li>

                    <div class = "info message">

                        Profile created, Successfully

                    </div>

                </li>

            <?php endif; ?>

            <?php if(isset(*$\_GET*['input\_missing'])) :?>

                <li>

                    <div class = "error message">

                        Missing inputs ...

                    </div>

                </li>

            <?php endif; ?>

*<!-- Maxlength is set according to size of uname field in login table -->*

            <li>

                <label> Emp Name <input name = "name" maxlength = 30 placeholder = "30 letters max" required <?php if(isset(*$name*)) echo "value = '*$name*'";?>> </label>

            </li>

            <li>

                <label> Email <input type = "email" name = "email" maxlength = 50 placeholder = "50 characters max" required <?php if(isset(*$email*)) echo "value = '*$email*'";?>> </label>

            </li>

            <?php if(isset(*$\_GET*['email\_used']) || isset(*$\_GET*['email\_valid'])) :?>

*<!-- the email entered is either invalid or already used -->*

                <li>

                    <div class = "error message">

                        Please use another Email.

                    </div>

                </li>

            <?php endif; ?>

            <li>

                <label> Phone <input type = "tel" name = "phone" maxlength = 10 placeholder = "10 characters max" required <?php if(isset(*$phone*)) echo "value = '*$phone*'";?>> </label>

            </li>

            <?php if(isset(*$\_GET*['phone\_valid'])) :?>

*<!-- the phone no. entered is invalid -->*

                <li>

                    <div class = "error message">

                        Invalid Phone No.

                    </div>

                </li>

            <?php endif; ?>

            <li>

                <label>

                    Rank

                    <select name = "ranks">

                        <option value = 0 <?php if(isset(*$ranks*) && *$ranks* == 0) echo "selected";?>> Employee </option>

                        <option value = 1 <?php if(isset(*$ranks*) && *$ranks* == 1) echo "selected";?>> Manager1 </option>

                        <option value = 2 <?php if(isset(*$ranks*) && *$ranks* == 2) echo "selected";?>> Manager2 </option>

                        <option value = 3 <?php if(isset(*$ranks*) && *$ranks* == 3) echo "selected";?>> HR </option>

                    </select>

                </label>

            </li>

            <li>

                <label> Password (For Profile) <input type = "password" name = "password" maxlength = 10 placeholder = "10 characters max" required <?php if(isset(*$password*)) echo "value = '*$password*'";?>> </label>

            </li>

            <?php if(isset(*$\_GET*['pass\_valid'])) :?>

*<!-- the password entered is invalid -->*

                <li>

                    <div class = "error message">

                        Invalid password.

                    </div>

                </li>

            <?php endif; ?>

            <li>

                <label> Re-enter Password <input type = "password" name = "password\_reenter" maxlength = 10 placeholder = "10 characters max" required <?php if(isset(*$password\_reenter*)) echo "value = '*$password\_reenter*'";?>> </label>

            </li>

*<!-- Original and reentered passwords must match -->*

            <?php if(isset(*$\_GET*['pass\_match'])) :?>

                <li>

                    <div class = "error message">

                        Reentered Password doesn't match

                    </div>

                </li>

            <?php endif; ?>

            <li>

                <div class = "error message">

                    Enter Password carefully, It can't be modified in future.

                </div>

            </li>

            <li>

                <input class = "button" type = "submit" name = "submit" value = "Create Profile">

            </li>

        </ul>

        </form>

        </main>

        <footer></footer>

    </body>

</html>

**>>>> hr\leave\_rules\_input.php**

Similar to **>>>> hr\emp\_input.php**

**>>>> hr\view.php (displays list of both the employee and leave rules)**

<?php

*/\**

*check if it's valid hr session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_hr\_session.php";

    include "../PHP/config.php";    *// connect to database*

*// all entries in employee or leave\_rule table*

*$type* = *$\_GET*["type"];

*// deleting emp or leave*

*$query* = (*$type* == true) ? "SELECT \* FROM employee" : "SELECT \* FROM leave\_rule";

*$result* = *$link* -> query(*$query*);

*$link* -> close();

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>view</title>

        <style>

            @import url("../CSS/general styles.css");

            @import url("../CSS/form styles.css");

            @import url("../CSS/header styles.css");

            @import url("../CSS/list styles.css");

        </style>

    </head>

    <body>

        <header id = 'first'>

            <?php include "header.php";?>

        </header>

        <main>

*<!-- Displaying the entries -->*

*<!-- 10 entries between 2 navigation boxes -->*

        <?php *$nav\_interval* = 10; *$nav\_index* = 0;?>

*<!-- getting the tuples in emp or leave\_rule table -->*

        <?php for(;*$row* = *$result* -> fetch\_assoc(); *$nav\_index*++): */\* index of the record read \*/*?>

*<!-- navigation box to reduce time to nevigate entire list -->*

        <?php if((*$nav\_index* % *$nav\_interval*) === 0):?>

            <ul id = <?php echo "menu*$nav\_index*";?> class = "main\_box green\_box">

*<!-- for first menu bar these two not necessary -->*

                <?php if(*$nav\_index* != 0):?>

*<!-- go to previous navigation box -->*

                    <li>

                        <a href = "<?php echo "#menu" . (*$nav\_index* - *$nav\_interval*)?>"><button class = 'button'> Previous </button></a>

                    </li>

*<!-- go to top of the page -->*

                    <li>

                        <a href = "#first"><button class = 'button'> Top </button></a>

                    </li>

                <?php endif; ?>

*<!-- go to bottom of the page -->*

                <li>

                    <a href = "#last"><button class = 'button'> Bottom </button></a>

                </li>

*<!-- go to next navigation box -->*

                <?php if(*$nav\_index* + *$nav\_interval* < *$result* -> *num\_rows*):?>

                    <li>

                        <a href = "<?php echo "#menu" . (*$nav\_index* + *$nav\_interval*)?>"><button class = 'button'> Next </button></a>

                    </li>

                <?php else:?>

*<!--*

*for last navigation box next button is same as Bottom button*

*as we keep a last navigation box at the bottom of the page*

*-->*

                    <li>

                        <a href = "<?php echo "#last"?>"><button class = 'button'> Next </button></a>

                    </li>

                <?php endif; ?>

            </ul>

        <?php endif; ?>

*<!-- the record box -->*

        <?php

*$id* = (*$type* == true) ? *$row*['eid'] : *$row*['lid'];

        ?>

        <div id = <?php echo "navid" . *$nav\_index*;?>></div>

        <ul id = <?php echo "id" . *$id*;?> class = "main\_box blue\_box">

            <?php if(*$type* == true):?>

*<!-- the emp attributes -->*

                <li>

                    <div class = "message">

                        <?php echo *$row*["name"]?>

                    </div>

                </li>

                <li>

                    <div class = "message">

                        <a href = "<?php echo "mailto:" . *$row*["email"]?>"><?php echo *$row*["email"]?></a>

                    </div>

                </li>

                <li>

                    <div class = "message">

                        <?php echo *$row*["phone"]?>

                    </div>

                </li>

                <li>

                    <div class = "message">

                        <?php

                            include\_once "../PHP/emp\_ranking\_system.php";

                            echo get\_rank(*$row*['ranks']);

                        ?>

                    </div>

                </li>

            <?php else:?>

*<!-- the leave attributes -->*

                <li>

                    <div class = "message">

                        <?php echo *$row*["type"]?>

                    </div>

                </li>

                <li>

                    <div class = "message">

                        <?php echo "Maximum ". *$row*["days"] . " Days"?>

                    </div>

                </li>

                <li>

                    <div class = "message">

                        <?php echo ((*$row*["need\_doc"]) ? "Need" : "Need no ") . " Supporting Document"?>

                    </div>

                </li>

            <?php endif?>

*<!-- update buttom -->*

            <li>

                <a href = "<?php echo "update.php?id=*$id*&type=*$type*"?>"><button class = 'button greenbutton'> Update </button></a>

            </li>

*<!-- delete buttom -->*

            <li>

                <a href = "<?php echo "delete.php?id=*$id*&navid=*$nav\_index*&type=*$type*"?>"><button class = 'button redbutton'> Delete </button></a>

            </li>

        </ul>

        <?php endfor; ?>

*<!-- top and previous button at the end of the page -->*

        <ul class = 'main\_box red\_box'>

            <li>

                <a href = "<?php echo "#menu" . (*$nav\_index* - *$nav\_index* % *$nav\_interval*)?>"><button class = 'button'> Previous </button></a>

            </li>

            <li>

                <a href = "#first"><button class = 'button'> Top </button></a>

            </li>

        </ul>

        </main>

        <footer id = "last"></footer>

        <script src="../JS/scroll\_back.js"></script>

    </body>

</html>

**>>>> hr\delete.php**

<?php

*/\**

*check if it's valid hr session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_hr\_session.php";

    if(isset(*$\_GET*["id"]) && isset(*$\_GET*["navid"]) && isset(*$\_GET*["type"]))

    {

        include "../PHP/config.php";    *// connect to database*

        include "../PHP/mysql\_sanitize\_input.php";

*// DELETE request from emp or leave display*

*$id* = mysql\_sanitize\_input(*$link*, *$\_GET*["id"]);

*$nav* = *$\_GET*["navid"];

*$type* = *$\_GET*["type"];

*// deleting emp or leave*

*$query* = (*$type* == true) ? "delete from employee where eid = *$id*" : "delete from leave\_rule where lid = *$id*";

*/\**

*emp or leave deletion failure will lead back to emp or leave Display*

*page (this is very unlikely to happen)*

*\*/*

        if(*$link* -> query(*$query*) === false)

        {

            die("Deletion failure, head back to <a href = 'view.php?type=*$type*#navid*$nav*'> Display </a>");

        }

*$link* -> close();

*// redirect to the entry right before the deleted entry*

        header("location: view.php?type=*$type*#navid" . (*$nav* - 1));

    }

    else

    {

        header("location: index.php");

    }

?>

**>>>> hr\update.php**

<?php

*/\**

*check if it's valid hr session or not if not redirect*

*to index or home page*

*\*/*

    include "../PHP/check\_hr\_session.php";

    if(isset(*$\_GET*["id"]) && isset(*$\_GET*["type"]))

    {

        include "../PHP/config.php";    *// connect to database*

        include "../PHP/mysql\_sanitize\_input.php";

*// update request from view.php*

*$id* = mysql\_sanitize\_input(*$link*, *$\_GET*["id"]);

*$type* = *$\_GET*["type"];  *// type = true for employee, type = false for leave*

*// extracting the tuple*

*$query* = (*$type* == true) ? "SELECT \* FROM employee WHERE eid = *$id*" : "SELECT \* FROM leave\_rule WHERE lid = *$id*";

*$result* = *$link* -> query(*$query*);

        if(*$result* === false || *$result* -> *num\_rows* == 0)

        {

*/\**

*Query execution failure or no tuples found, this leads back to emp Display*

*page (this is very unlikely to happen)*

*\*/*

            die("Failed to load the records, head back to <a href = 'view.php?type=*$type*#navid*$nav*'> Display </a>");

        }

*/\**

*loading a tuple as an associative array into session to*

*be accessed from emp or leave update form, to display current values*

*of the emp or leave*

*\*/*

*$\_SESSION*["tuple"] = *$result* -> fetch\_assoc();

*$link* -> close();

*/\**

*redirect to the form to accept inputs, also send the eid or lid of*

*the entry to be updated to that form as get value, so that if*

*anything goes wrong there we can head directly back to the*

*emp or leave details in emp or leave display*

*\*/*

        if(*$type* == true)

        {

            header("location: emp\_update.php?id=*$id*");

        }

        else

        {

            header("location: leave\_update.php?id=*$id*");

        }

    }

    else

    {

        header("location: index.php");

    }

?>

**>>>> hr\emp\_update.php**

<?php

*/\**

*check if it's valid hr session or not if not redirect*

*to index or home page (with updated error management system)*

*\*/*

    include "../PHP/check\_hr\_session.php";

    include "../PHP/message\_box.php";

*// Empdetails input*

    if(isset(*$\_POST*['submit']))

    {

        include "../PHP/config.php";

        include "../PHP/mysql\_sanitize\_input.php";

        include "../PHP/validate\_email.php";

*$eid* = mysql\_sanitize\_input(*$link*, *$\_POST*["eid"]);

*// check if the email is new and valid or not*

*$email* = strtolower(mysql\_sanitize\_input(*$link*, *$\_POST*['email']));

*$result* = *$link* -> query("SELECT \* FROM employee WHERE email = '*$email*' AND eid <> *$eid*");

        if(*$result* -> *num\_rows* > 0 || !validate\_email(*$email*))

        {

*// the email is already used*

*// here an error message can be displayed along with a OK button (attached to a url)*

            message\_box("Update failure (use different email)", "hr/update.php?id=*$eid*&type=1", true);

        }

*// check if the phone no. is valid or not*

        if(strlen(*$\_POST*['phone']) != 10)

        {

*// the phone no. is not valid*

            message\_box("Update failure (invalid phone number)", "hr/update.php?id=*$eid*&type=1", true);

        }

*$name* = mysql\_sanitize\_input(*$link*, *$\_POST*['name']);

*$phone* = mysql\_sanitize\_input(*$link*, *$\_POST*['phone']);

*$ranks* = mysql\_sanitize\_input(*$link*, *$\_POST*['ranks']);

*$query* = "UPDATE employee SET name = '*$name*', email = '*$email*', phone = *$phone*, ranks = *$ranks* WHERE eid = *$eid*;";

        if(*$link* -> query(*$query*) === false)

        {

*/\**

*emp update failure will lead back to emp Display page (this*

*is very unlikely to happen).*

*\*/*

            message\_box("Update failure (unknown reason), click <b>OK</b> to head back to <b>Display all Employees</b>", "hr/view.php?type=1#id*$eid*", true);

        }

*$link* -> close();

        header("location: view.php?type=1#id*$eid*");

    }

    else

    {

*/\**

*here we check for the session variable we created in "*

*update.php"*

*If that thing doesn't exist we return to emp display page*

*at certain emp detail*

*If that thing exists we store its data into local variables*

*and delete the session variable so that it can't be used*

*again to reload this form*

*\*/*

        if(!isset(*$\_SESSION*['tuple']))

        {

*/\**

*this file receives eid of the employee to be updated from*

*"update php" so that we can return to proper*

*emp in emp display page*

*if no such get value is received we simply go back to the*

*top of emp display page*

*\*/*

*// invalid access*

*$id* = (isset(*$\_GET*["id"])) ? "#id" . *$\_GET*["id"] : "";

            message\_box("Form initializing failure, click <b>OK</b> to head back to <b>Display all Employees</b>", "hr/view.php?type=1#id*$eid*", true);

        }

        else

        {

*$eid* = *$\_SESSION*["tuple"]['eid'];

*$name* = *$\_SESSION*["tuple"]['name'];

*$email* = *$\_SESSION*["tuple"]['email'];

*$phone* = *$\_SESSION*["tuple"]['phone'];

*$ranks* = *$\_SESSION*["tuple"]['ranks'];

            unset(*$\_SESSION*["tuple"]);

        }

    }

?>

<!DOCTYPE html>

<html lang = "en">

    <head>

        <meta charset = "UTF-8">

        <title>emp\_update</title>

        <style>

            @import url("../CSS/general styles.css");

            @import url("../CSS/form styles.css");

            @import url("../CSS/header styles.css");

        </style>

    </head>

    <body>

        <header>

            <?php include "header.php";?>

        </header>

        <main>

        <form method = "post" action = "emp\_update.php">

        <input type = 'hidden' name = "eid" value = <?php echo *$eid*;?>>

        <ul class = "main\_box nice\_shadow">

*<!-- Maxlength is set according to size of uname field in login table -->*

            <li>

                <label> Emp Name <input name = "name" maxlength = 30 value = "<?php echo *$name*;?>" required> </label>

            </li>

            <li>

                <label> Email <input type = "email" name = "email" maxlength = 50 value = "<?php echo *$email*;?>" required> </label>

            </li>

            <li>

                <label> Phone <input type = "tel" name = "phone" maxlength = 10 value = "<?php echo *$phone*;?>" required> </label>

            </li>

            <li>

                <label>

                    Rank

                    <select name = "ranks">

                        <option value = 0 <?php if(*$ranks* == 0) echo 'selected';?>> Employee </option>

                        <option value = 1 <?php if(*$ranks* == 1) echo 'selected';?>> Manager1 </option>

                        <option value = 2 <?php if(*$ranks* == 2) echo 'selected';?>> Manager2 </option>

                        <option value = 3 <?php if(*$ranks* == 3) echo 'selected';?>> HR </option>

                    </select>

                </label>

            </li>

            <li>

                <div class = "error message">

                    Invalid Inputs will Fail Update Process

                </div>

            </li>

            <li>

                <input class = "button" type = "submit" name = "submit" value = "Update">

            </li>

        </ul>

        </form>

        </main>

        <footer></footer>

    </body>

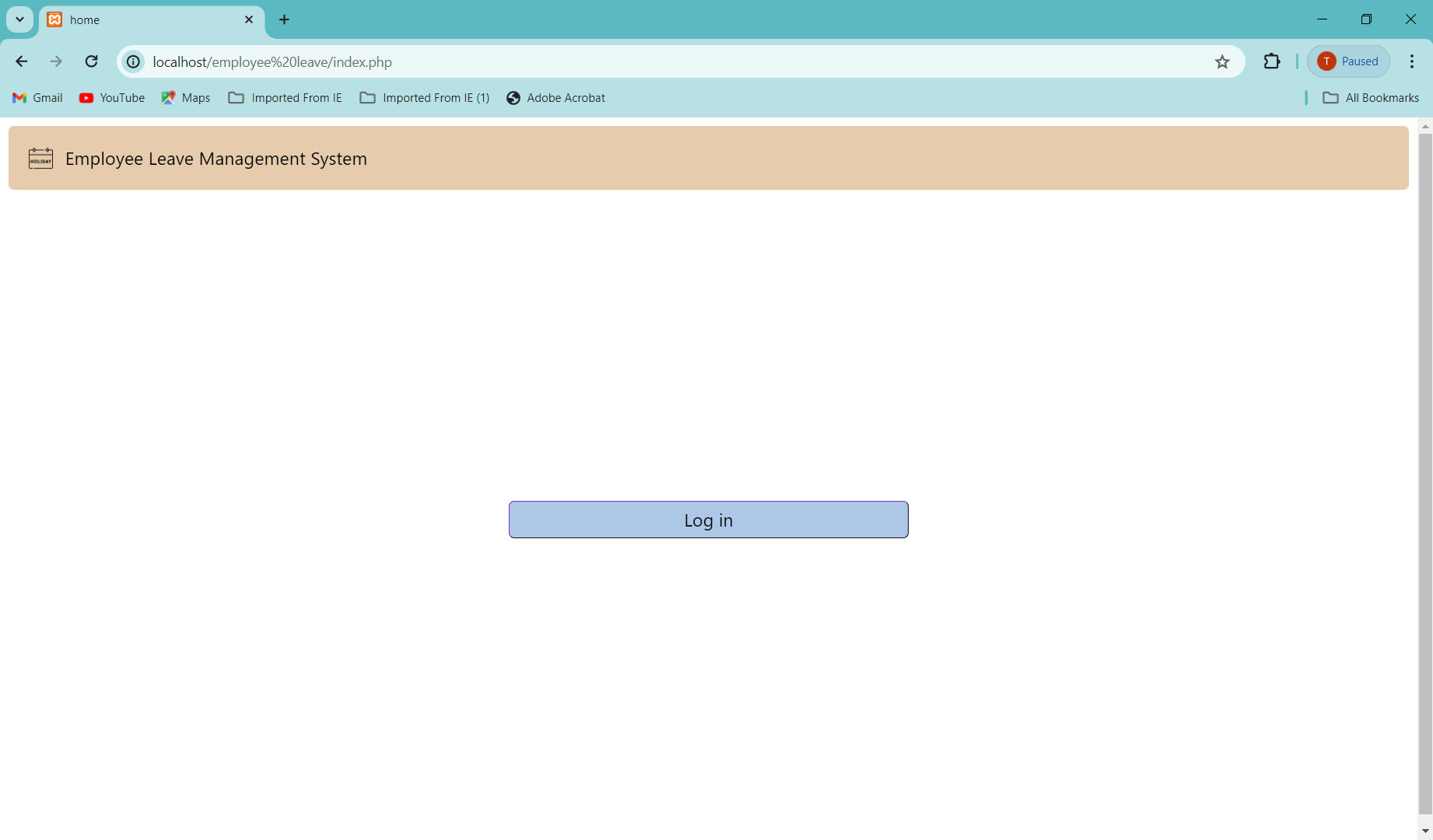
</html>

**>>>> hr\leave\_update.php**

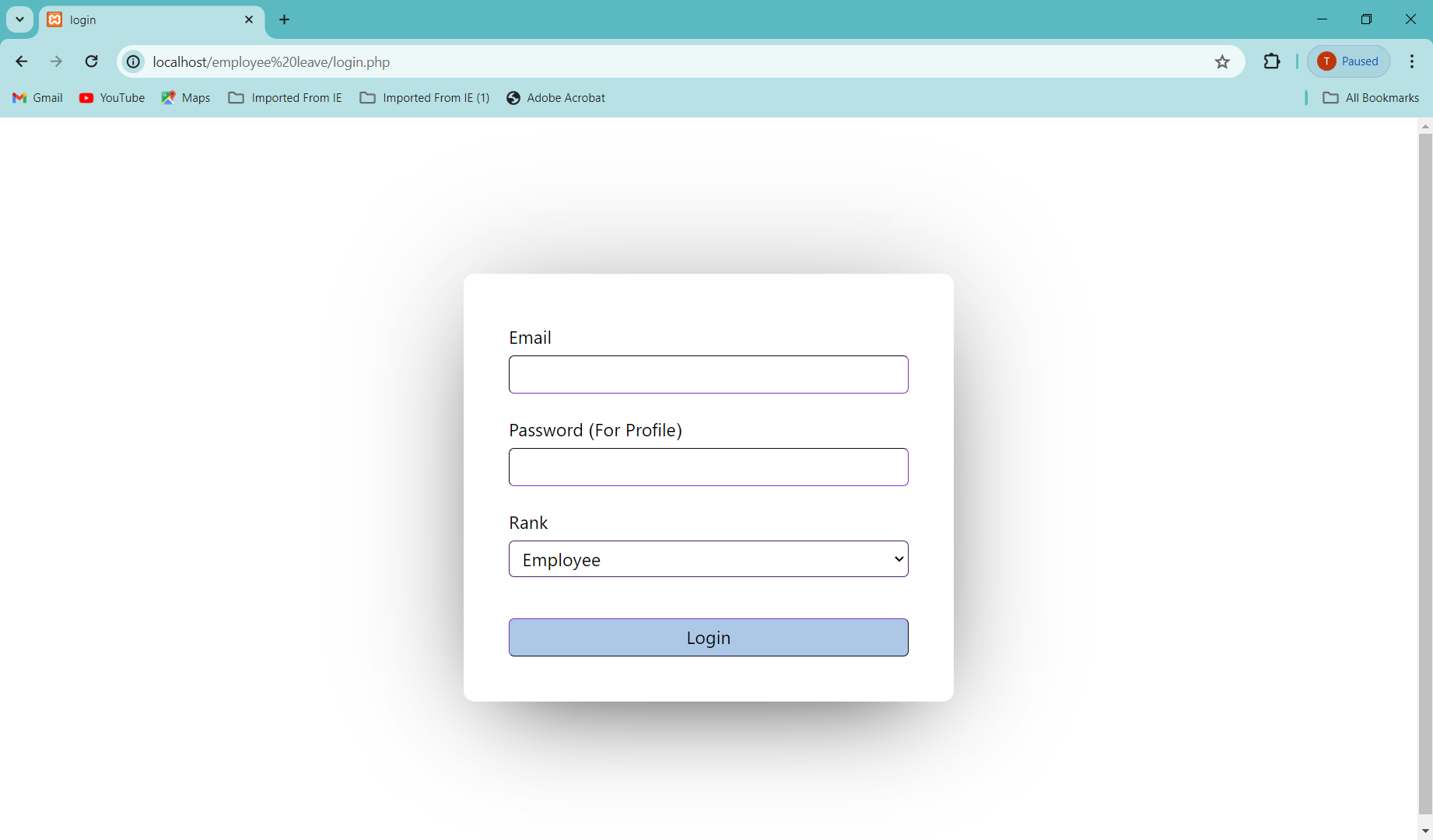
Similar to **>>>> hr\emp\_update.php**

**Results and Discussion:**

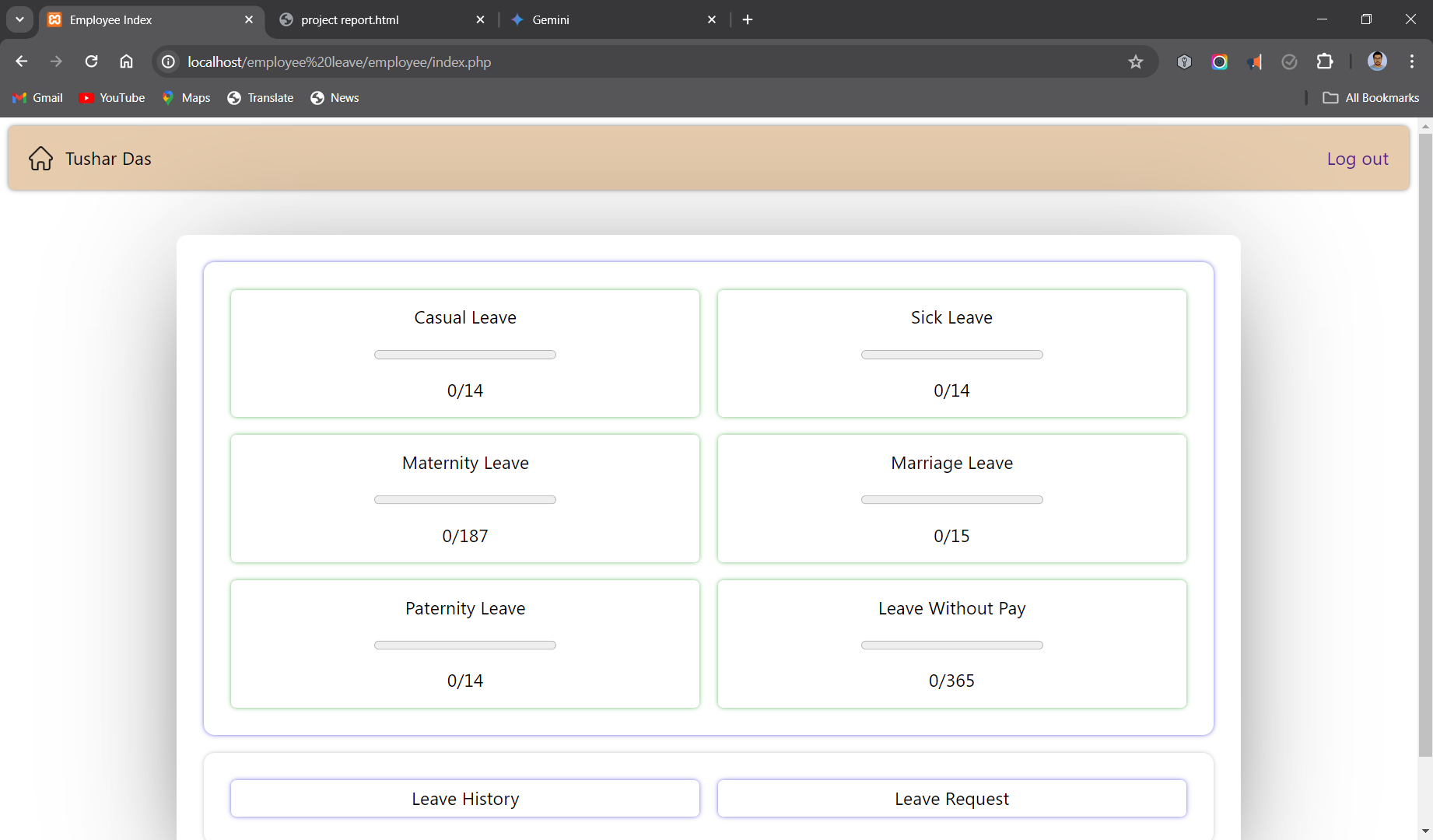
**Home page:**



**Login page:**

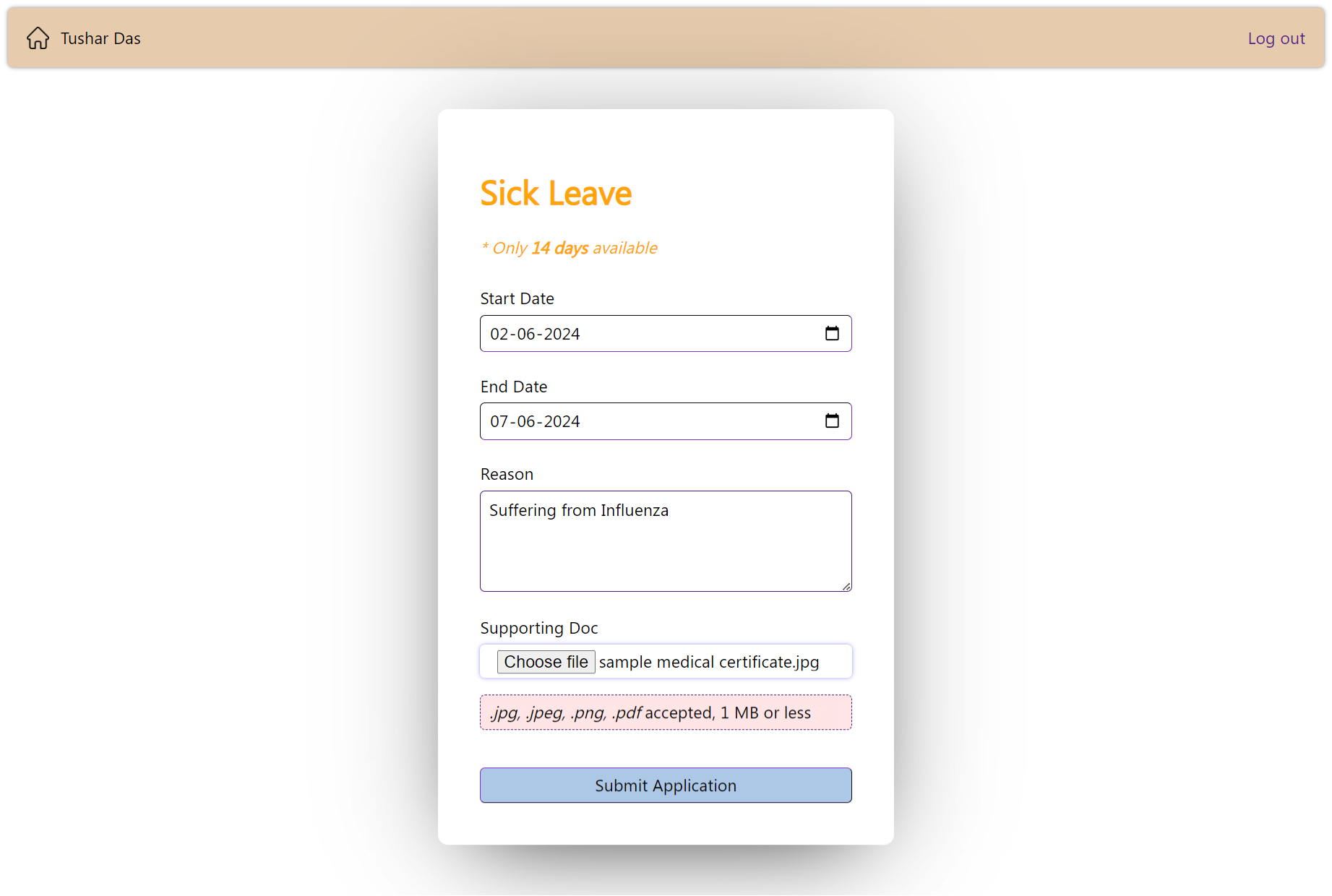


**Leave Balance of an employee:**

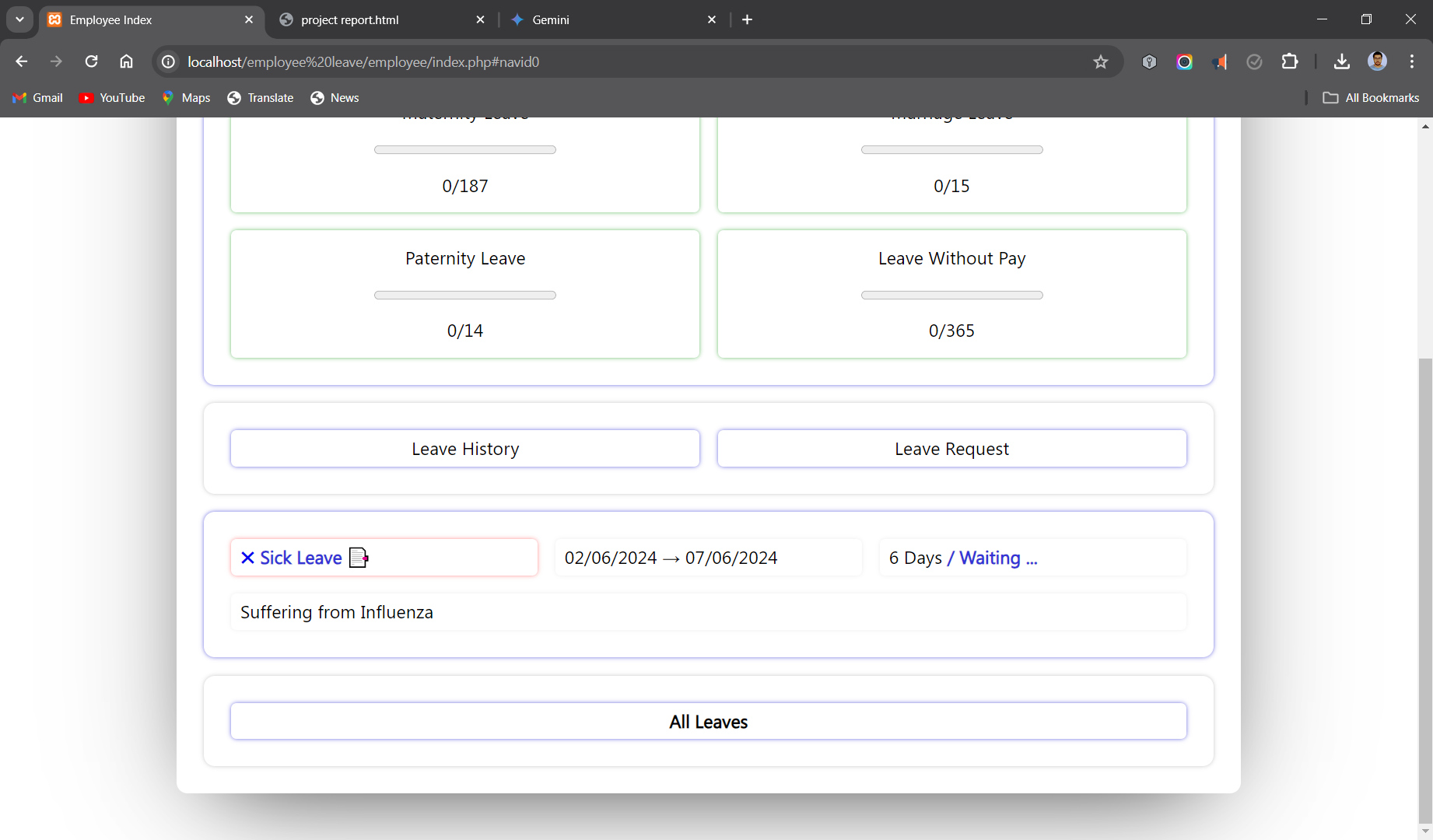


**Life cycle of a Leave Request:**

**Step 1:** Employee Submits a Leave Application:

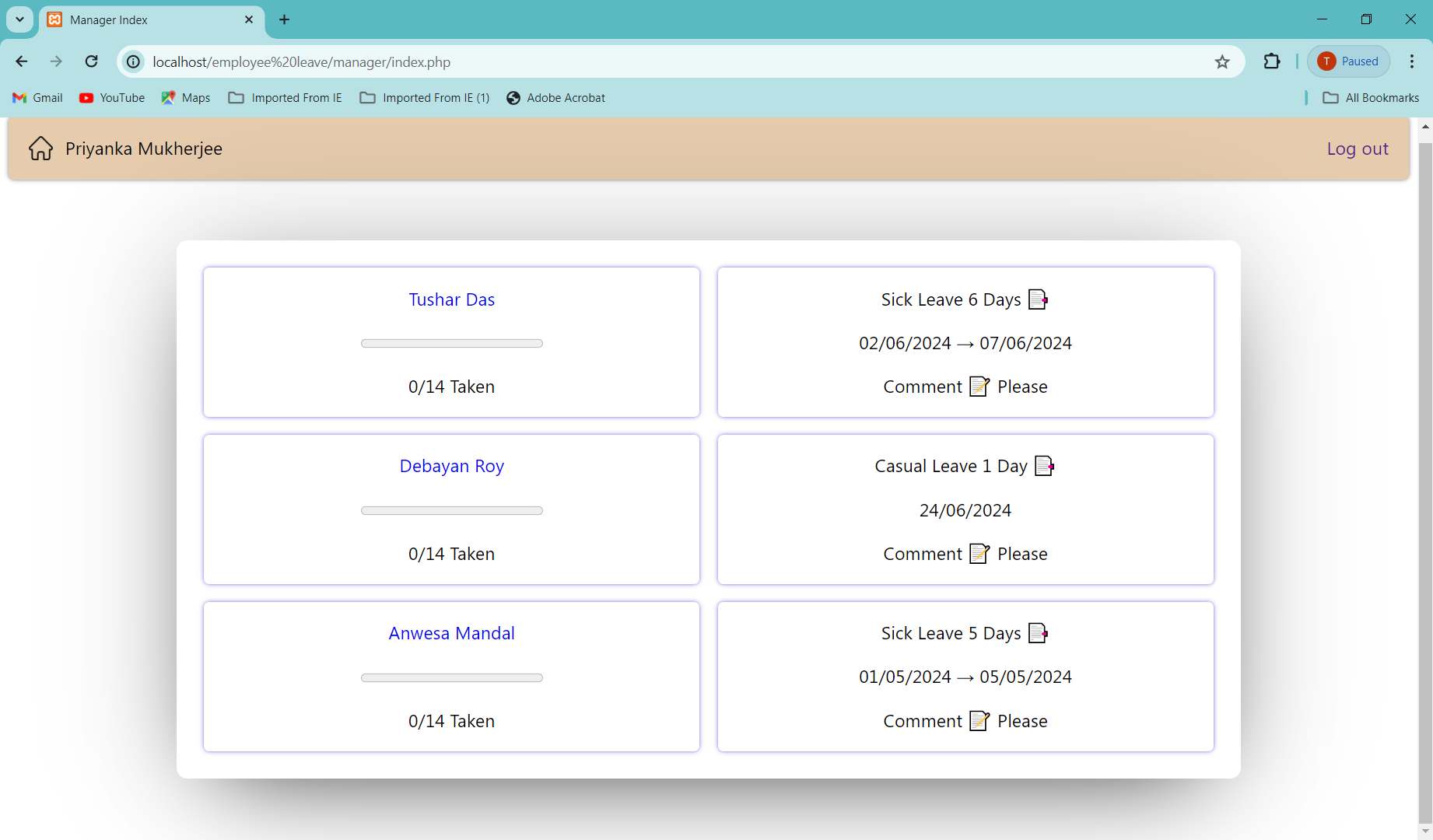


**Waiting Leave Application** is Displayed in Employee Dashboard



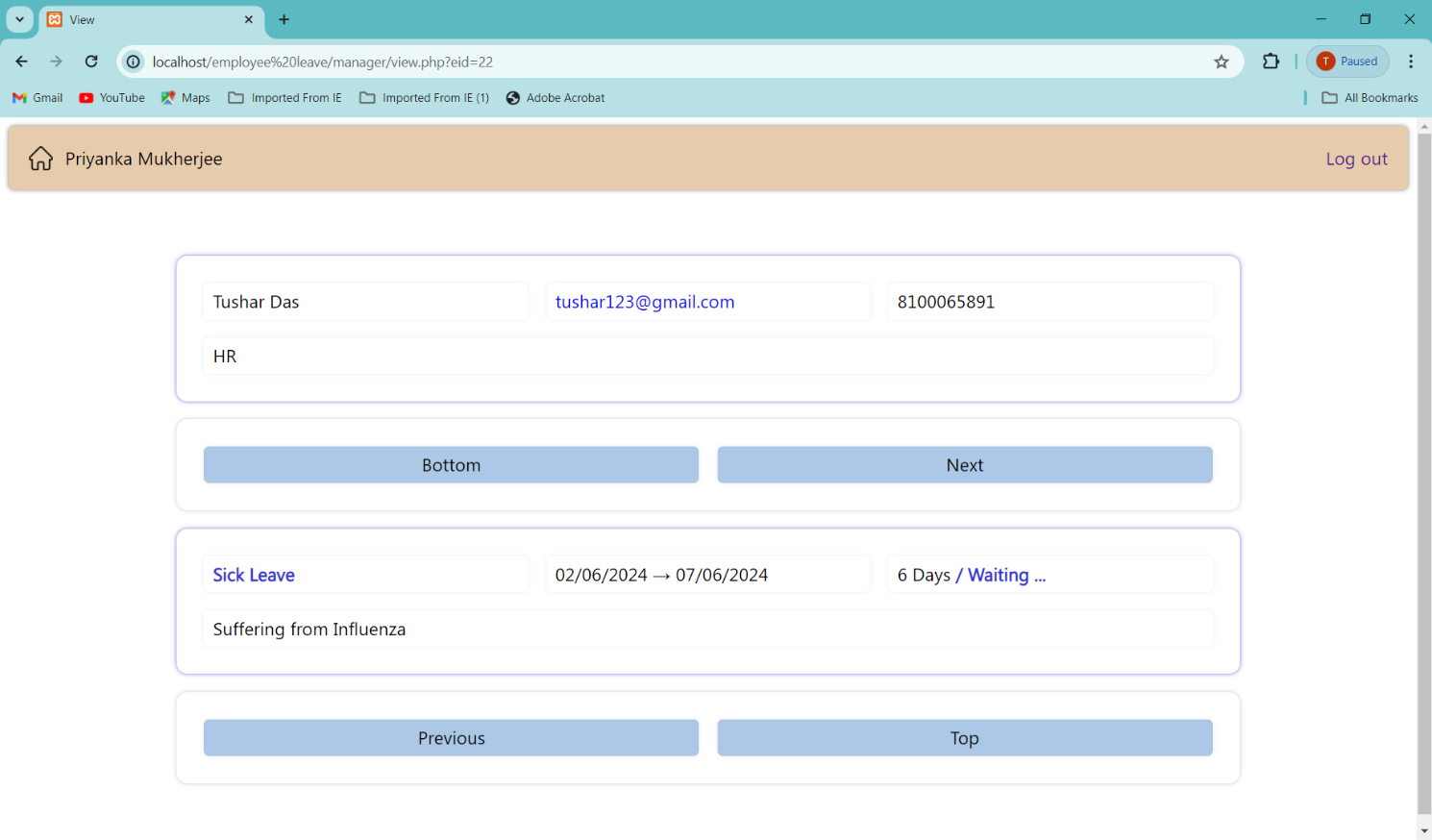
Employee can delete Leave Request by clicking on the cross button or view the supporting document by clicking on the attachment emoji.

**Step2:** Leave Application Appears At Manager 1 Dashboard and she enters her consent:

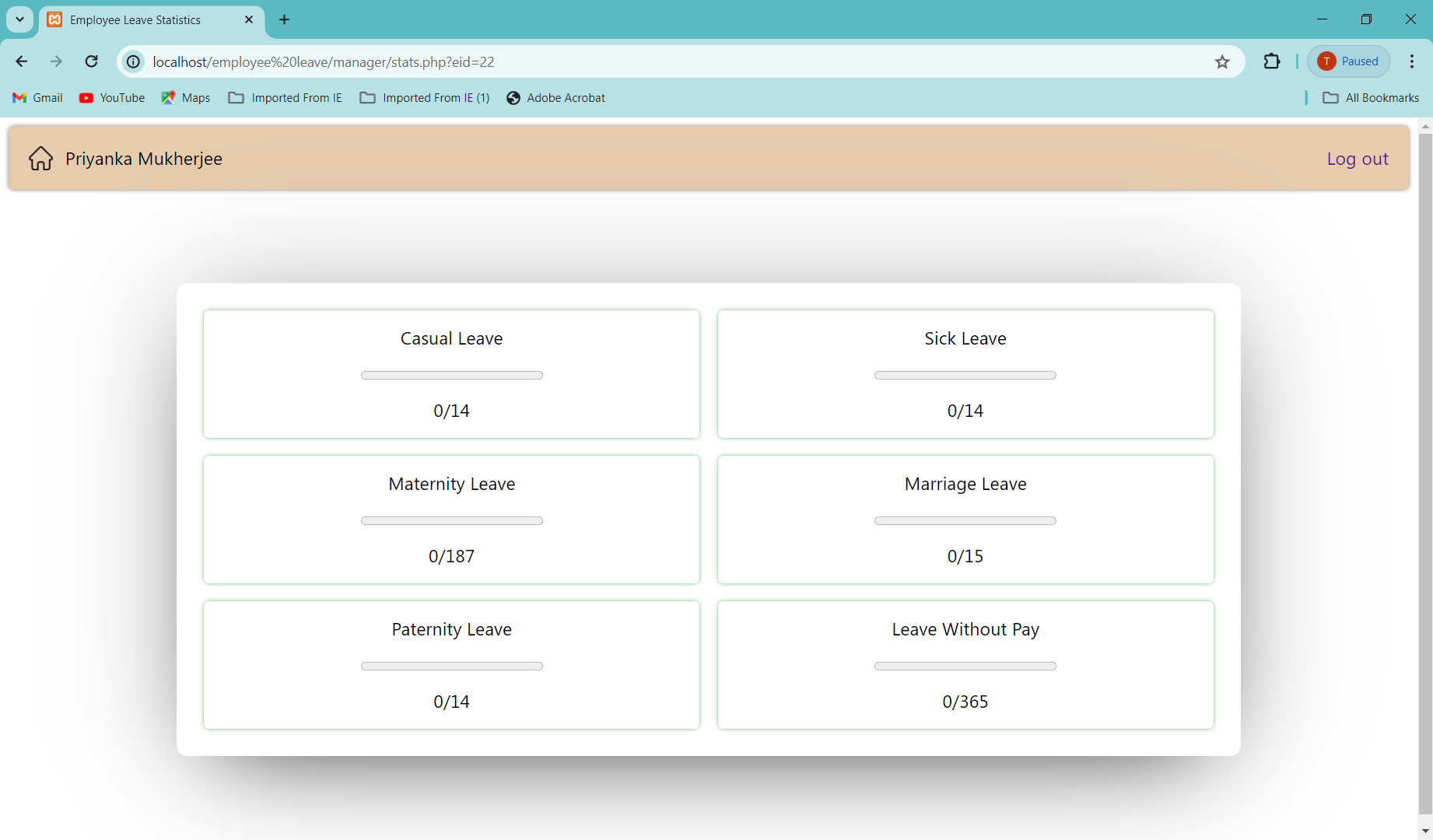


Manager 1 Dashboard controls:

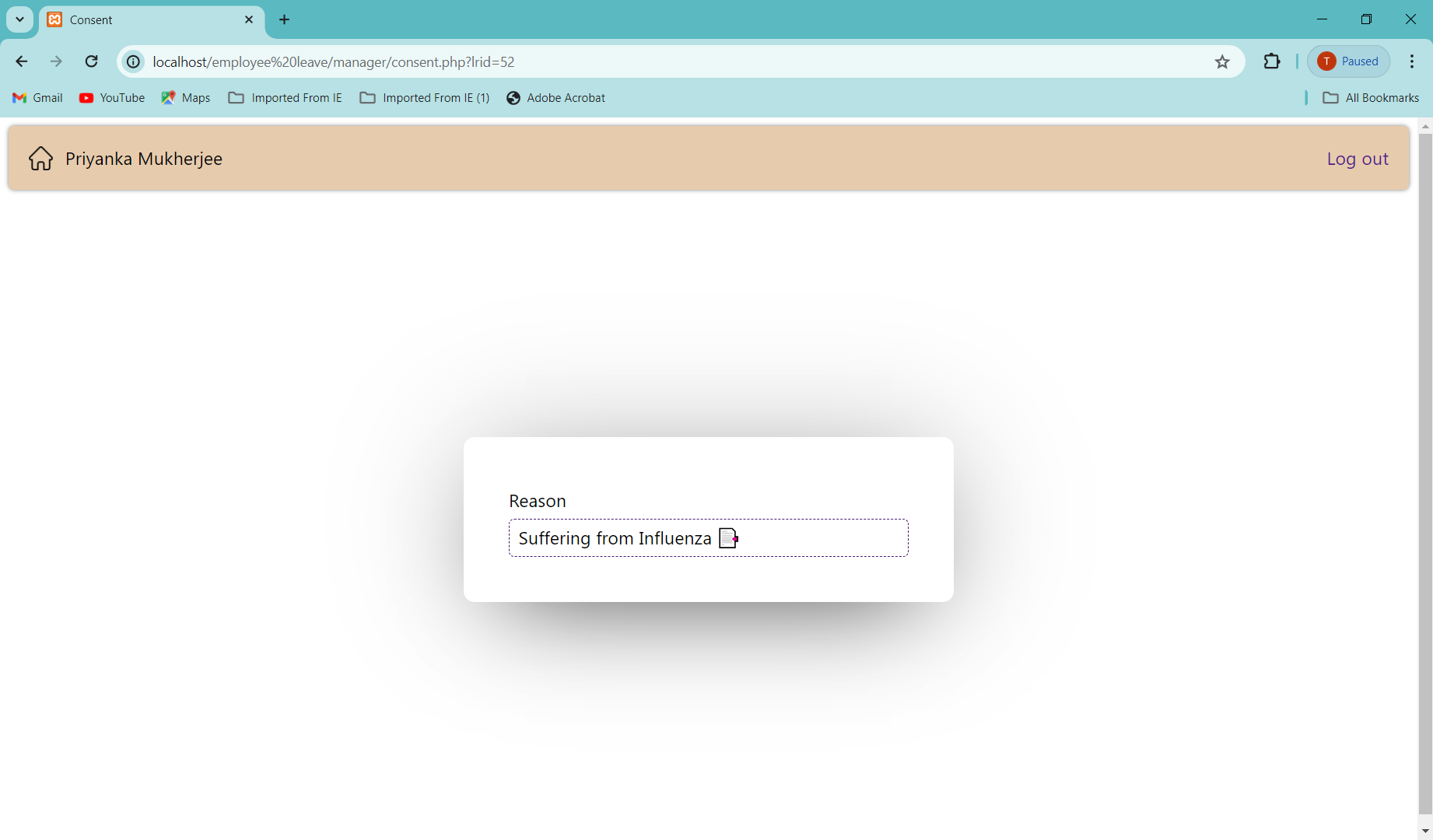
Click on Employee Name to view **Employee Details and leave history**



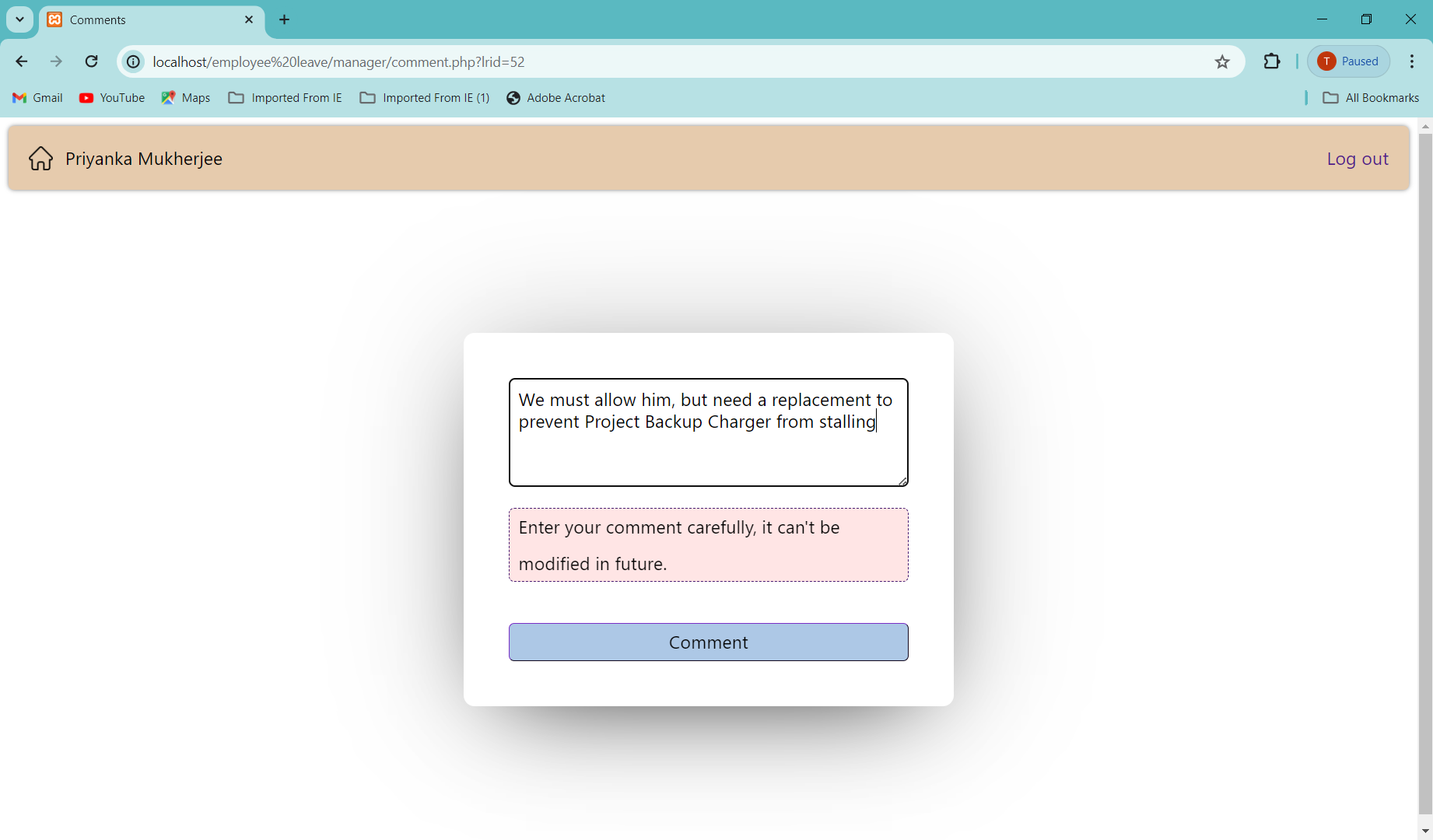
Click on Progress Bar to view **Leave Balance** of the Employee



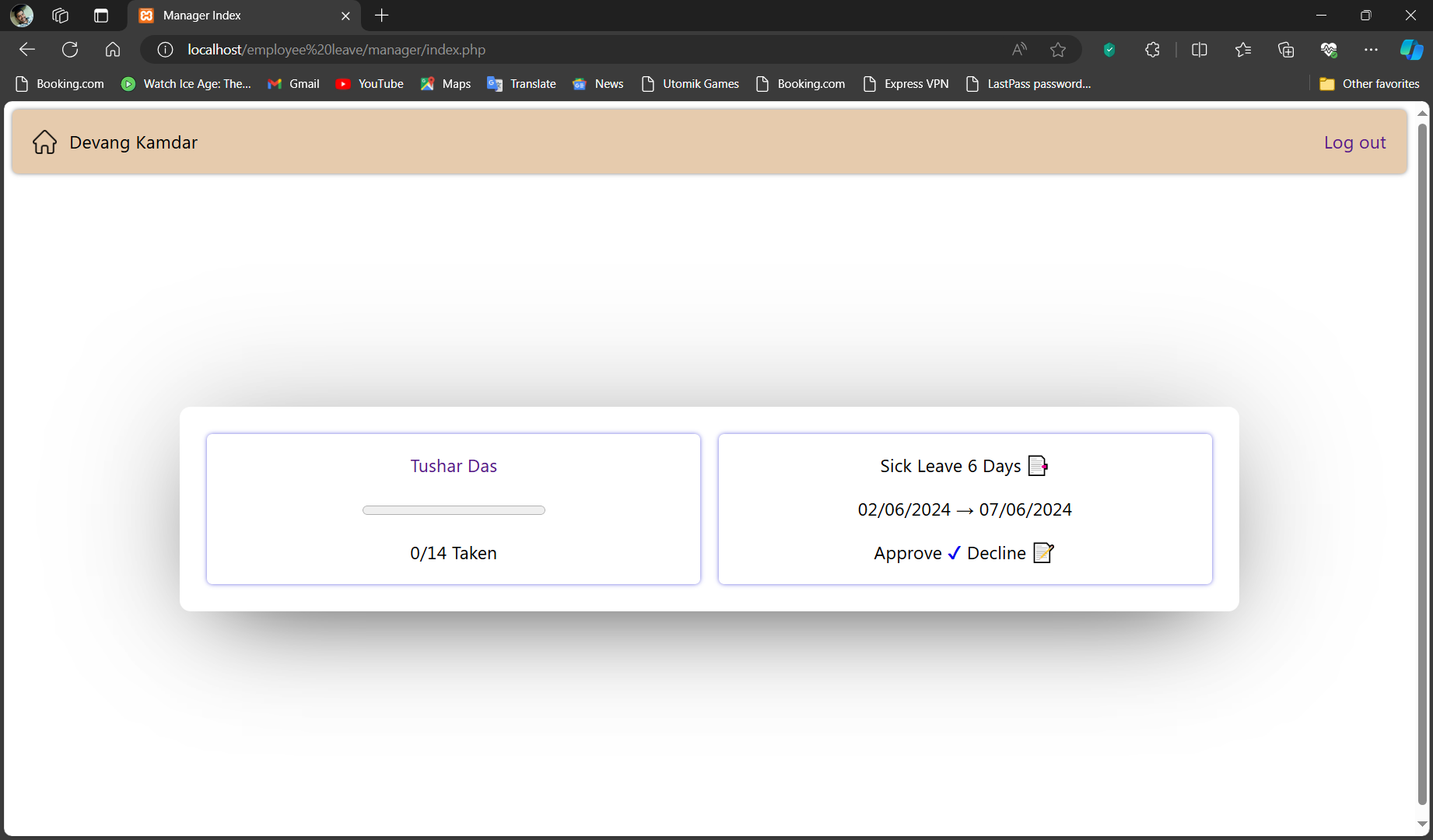
Click on Attachment emoji to view **Leave Reason**



Click on Comment emoji to enter **manager 1 consent**

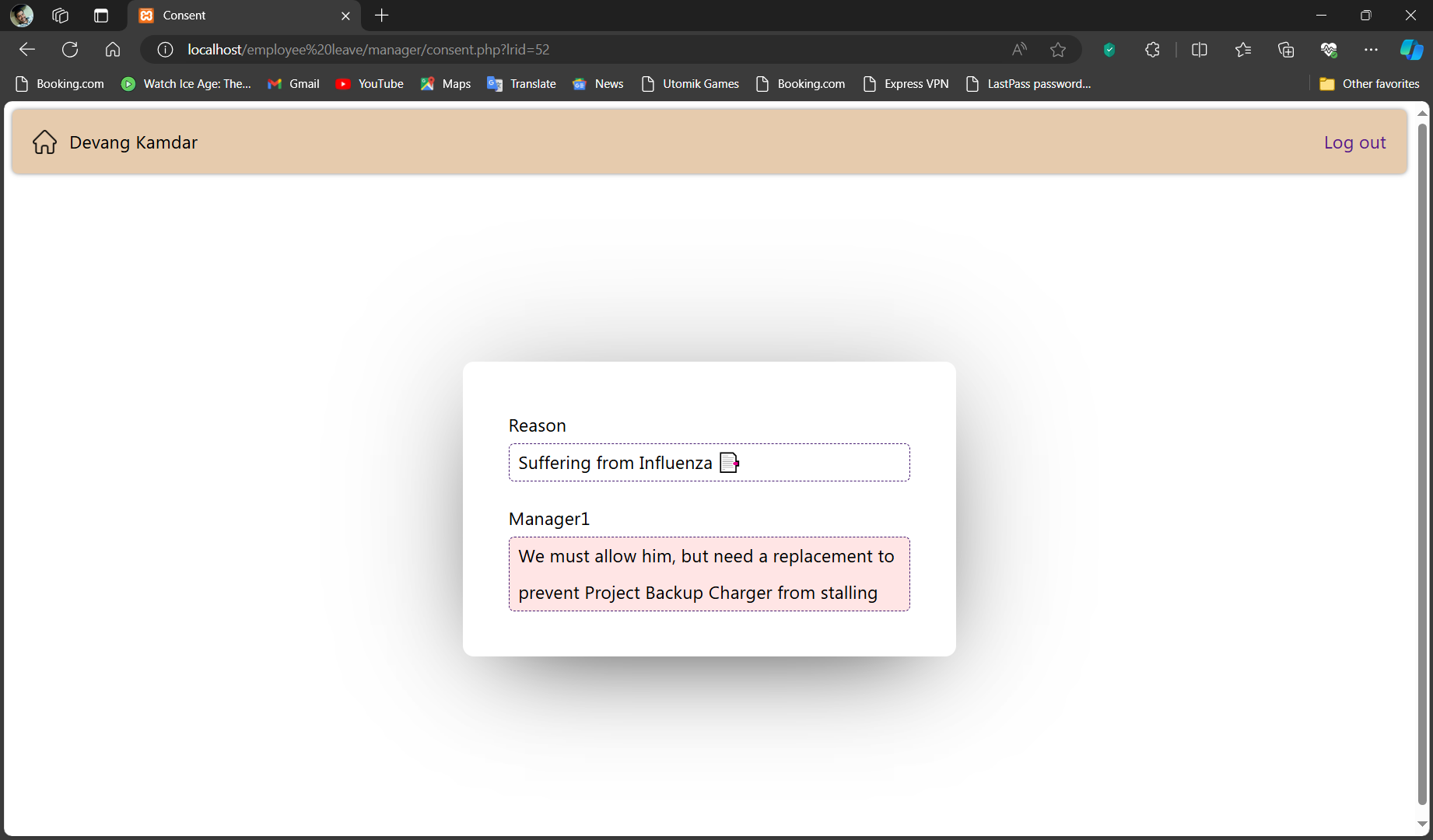


**Step 3:** Leave Application Appears At Manager 2 Dashboard After It’s Reviewed By Manager 1, he either approves it or declines it:

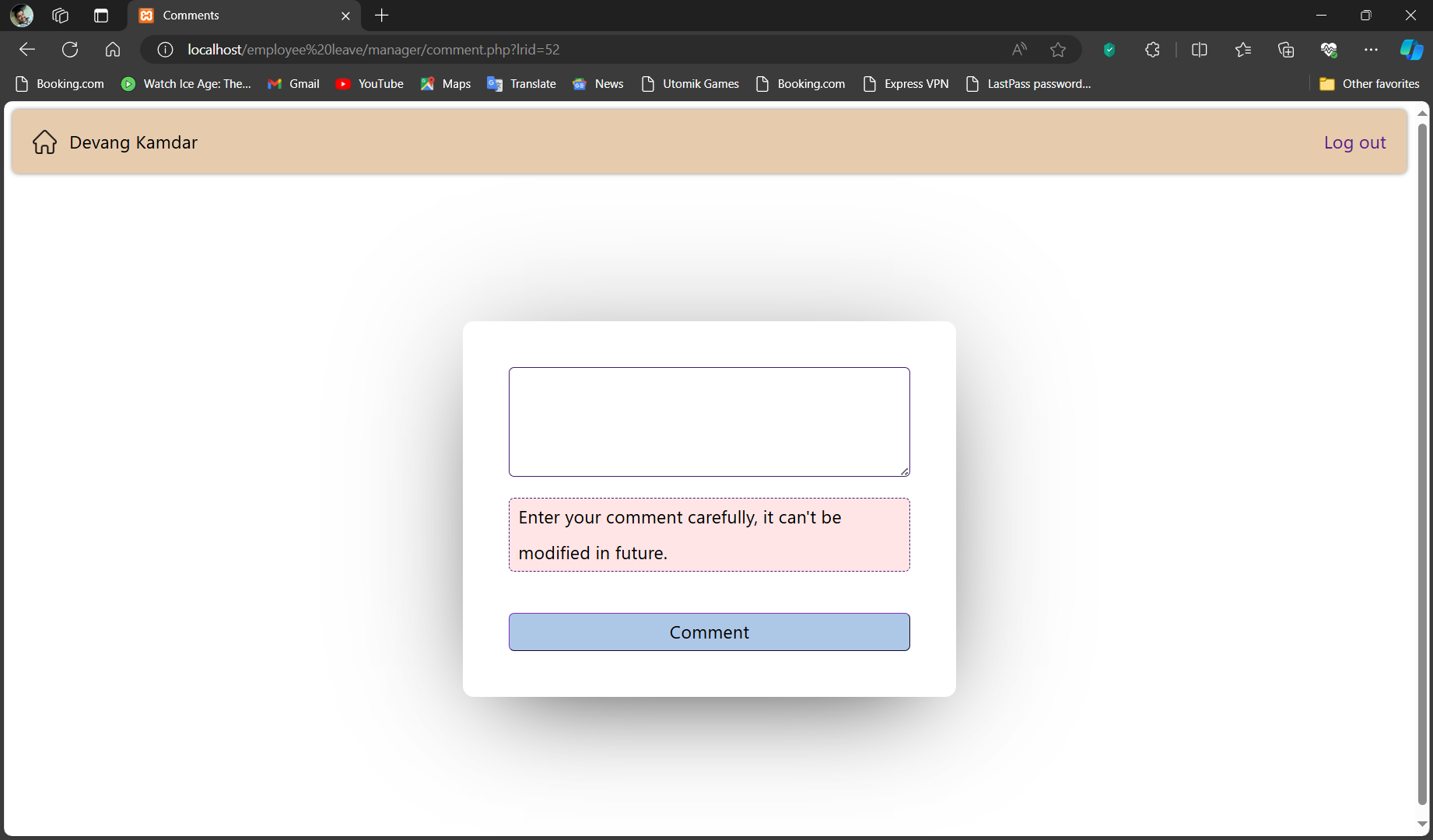


Almost every control is same as manager 1 dashboard. **Consent** option is replaced with **approval** and **decline** options respectively to either **approve** and **decline** the leave request with reason.

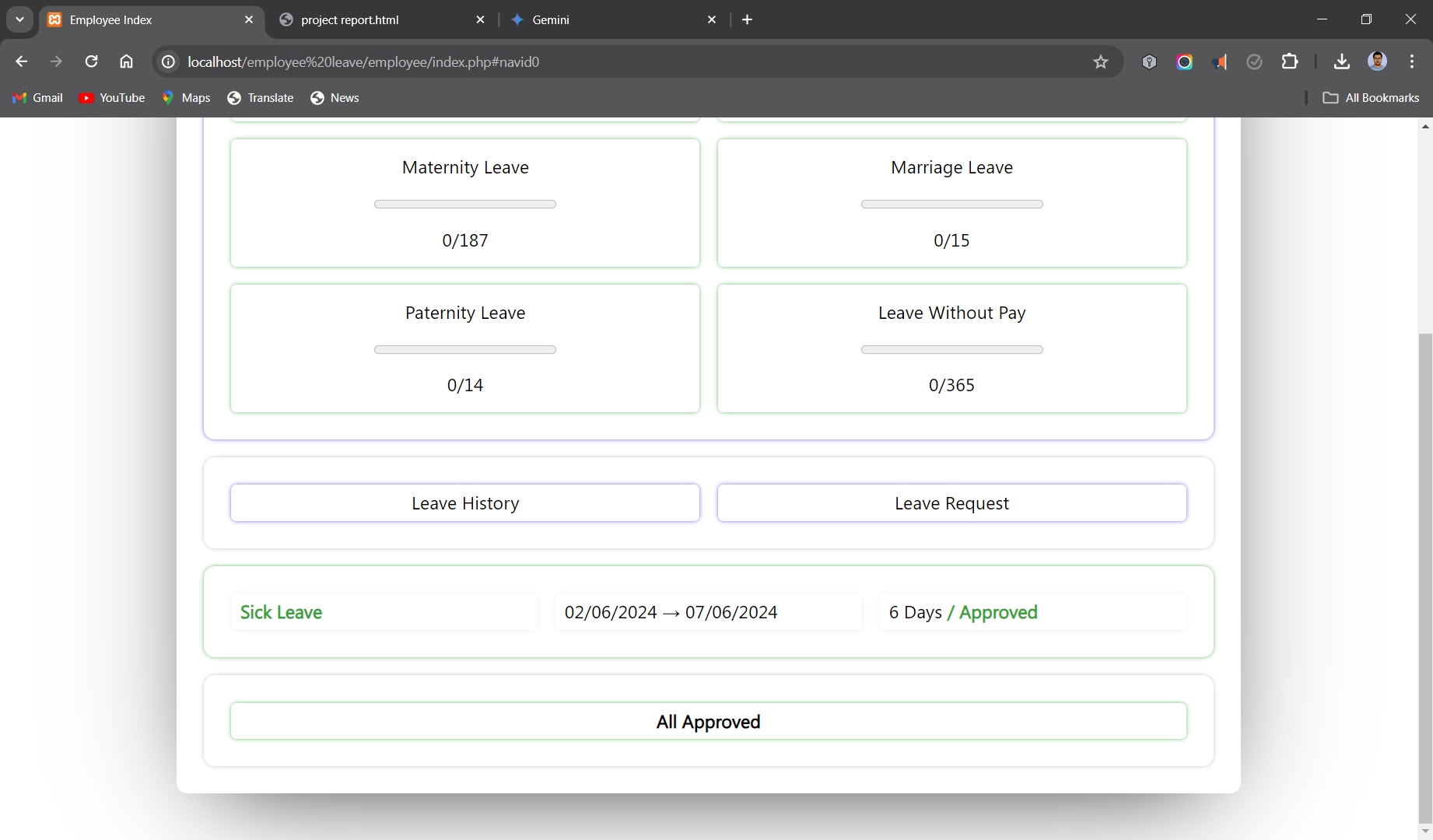
Attachment button here displays **manager 1 consent** along with **leave reason**.



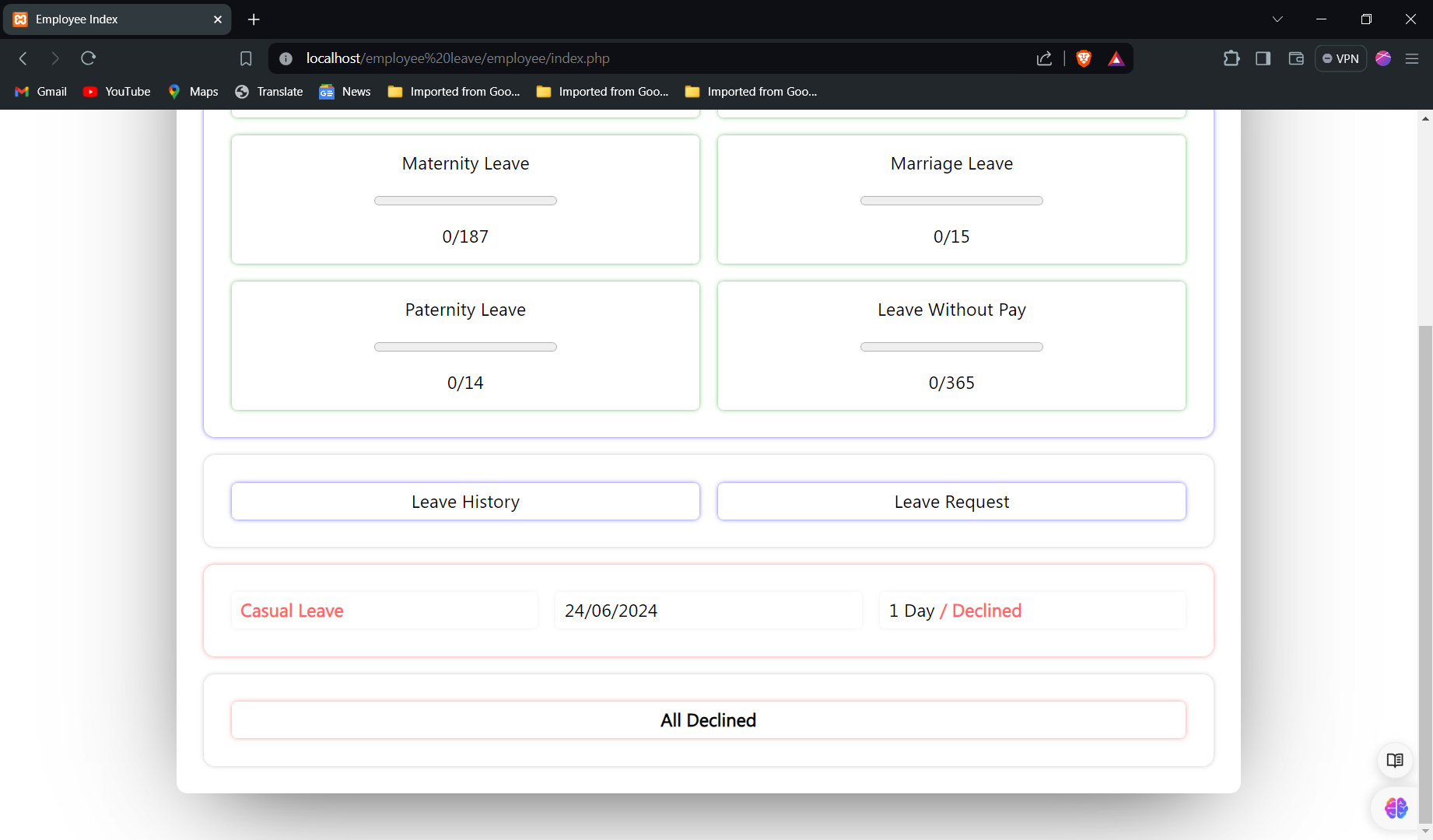
Decline button leads to this



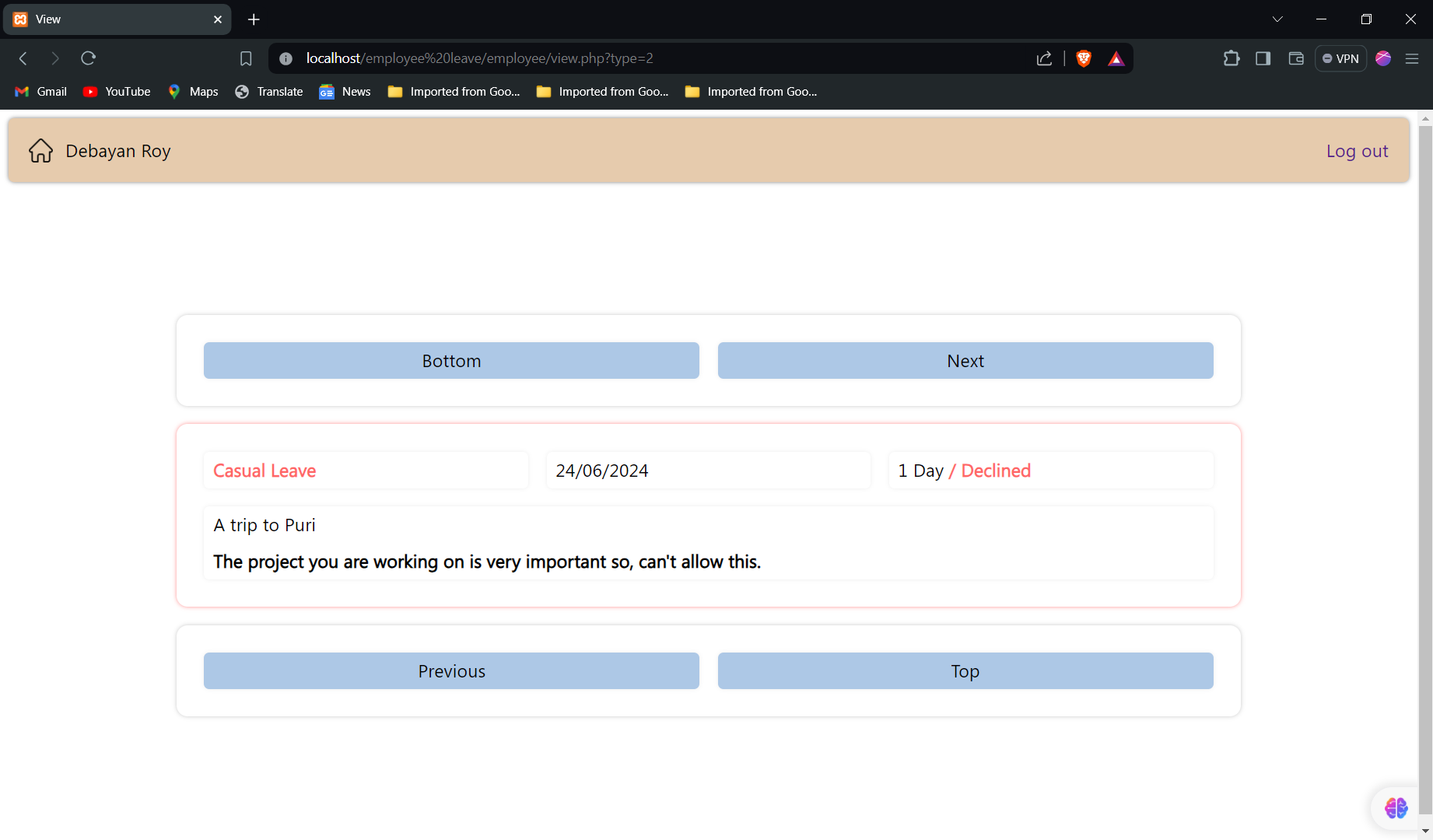
**Step 4a**: After a Leave Request Gets Approved It Appears In The Approval List In Employe Dashboard:



**Step 4b:** After a Leave Request Gets Declined It Appears In The Declined List In Employe Dashboard:

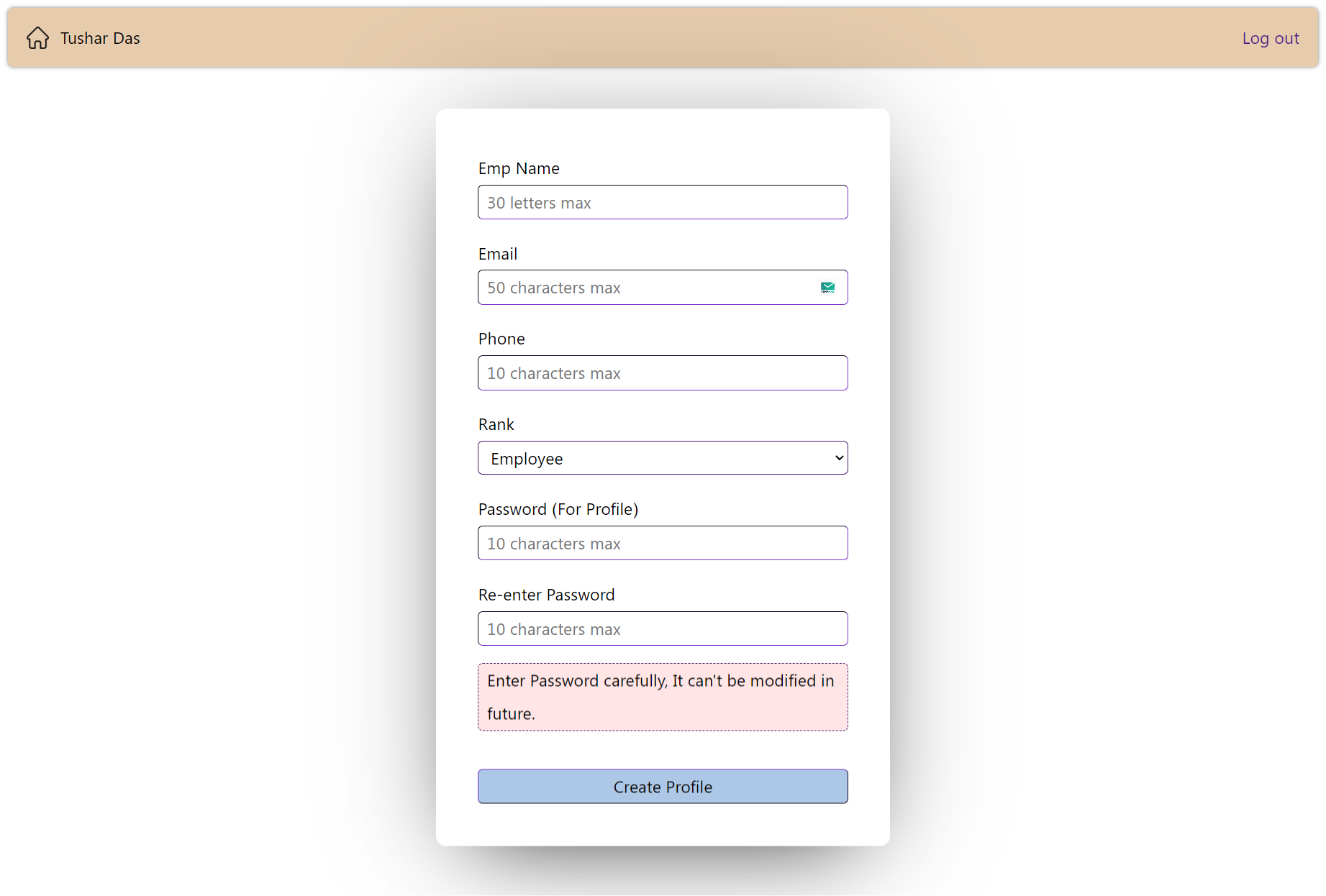


To know why leave request is declined employee need to open all declined list.

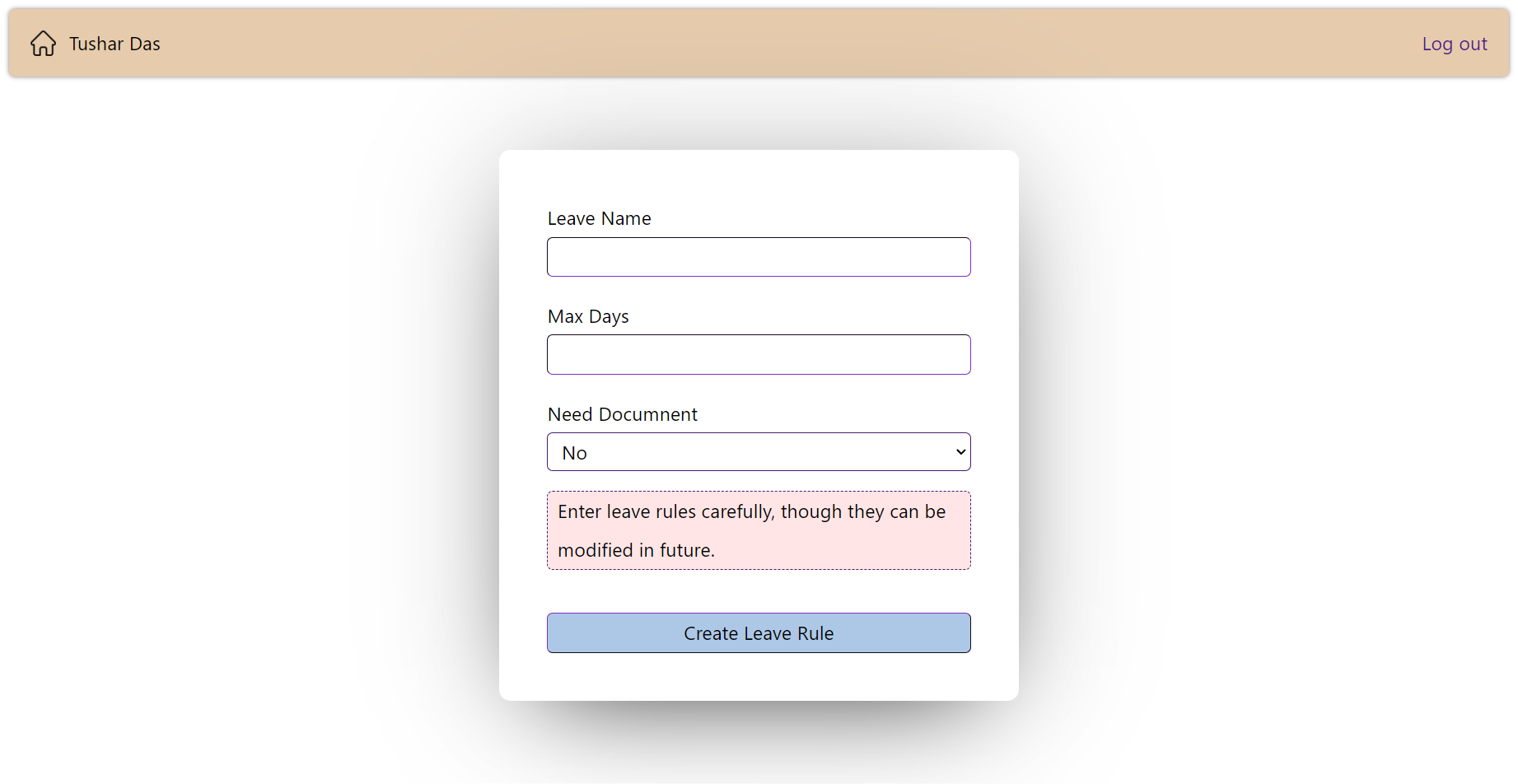


**HR section:**

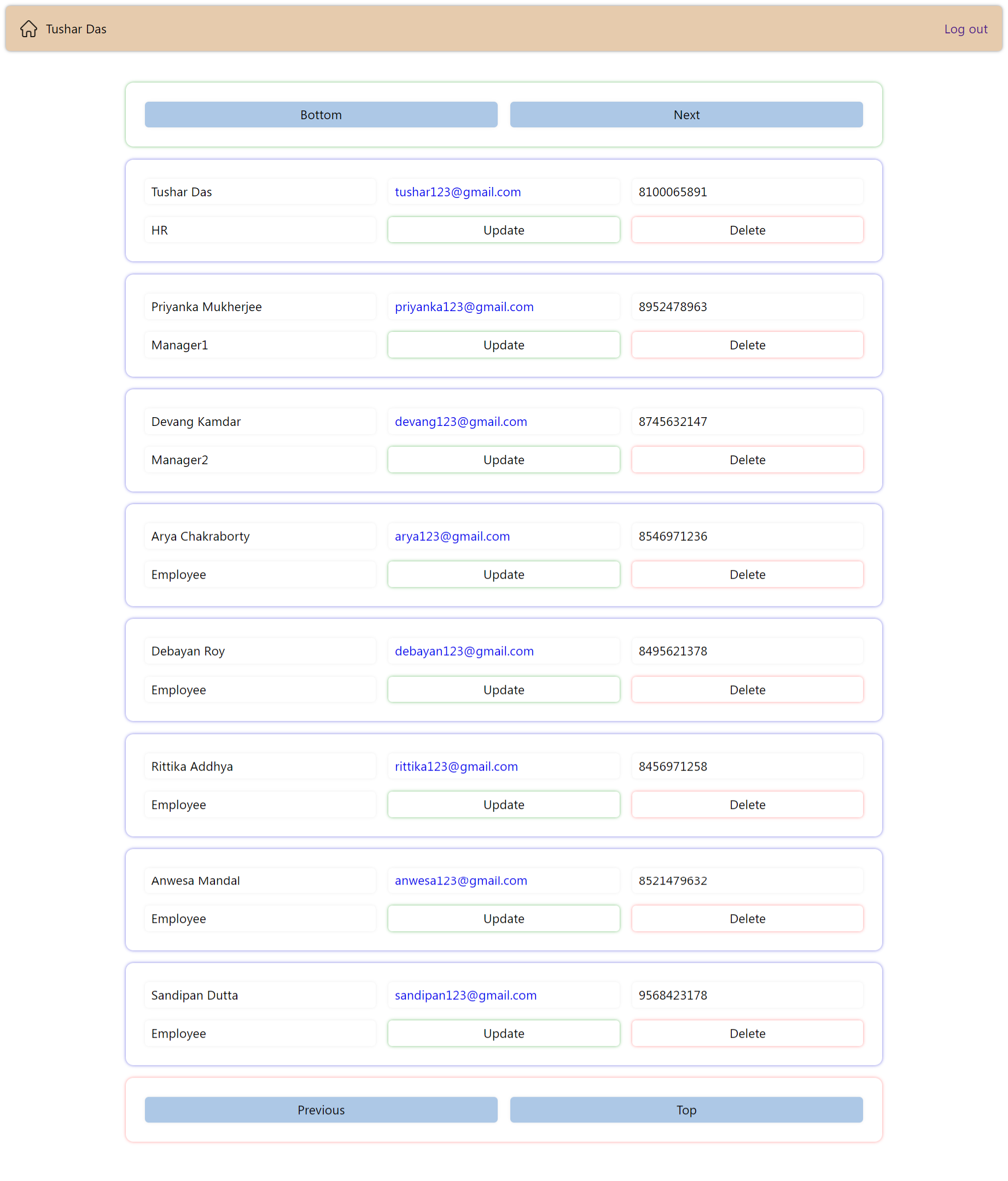
Employee Input

****

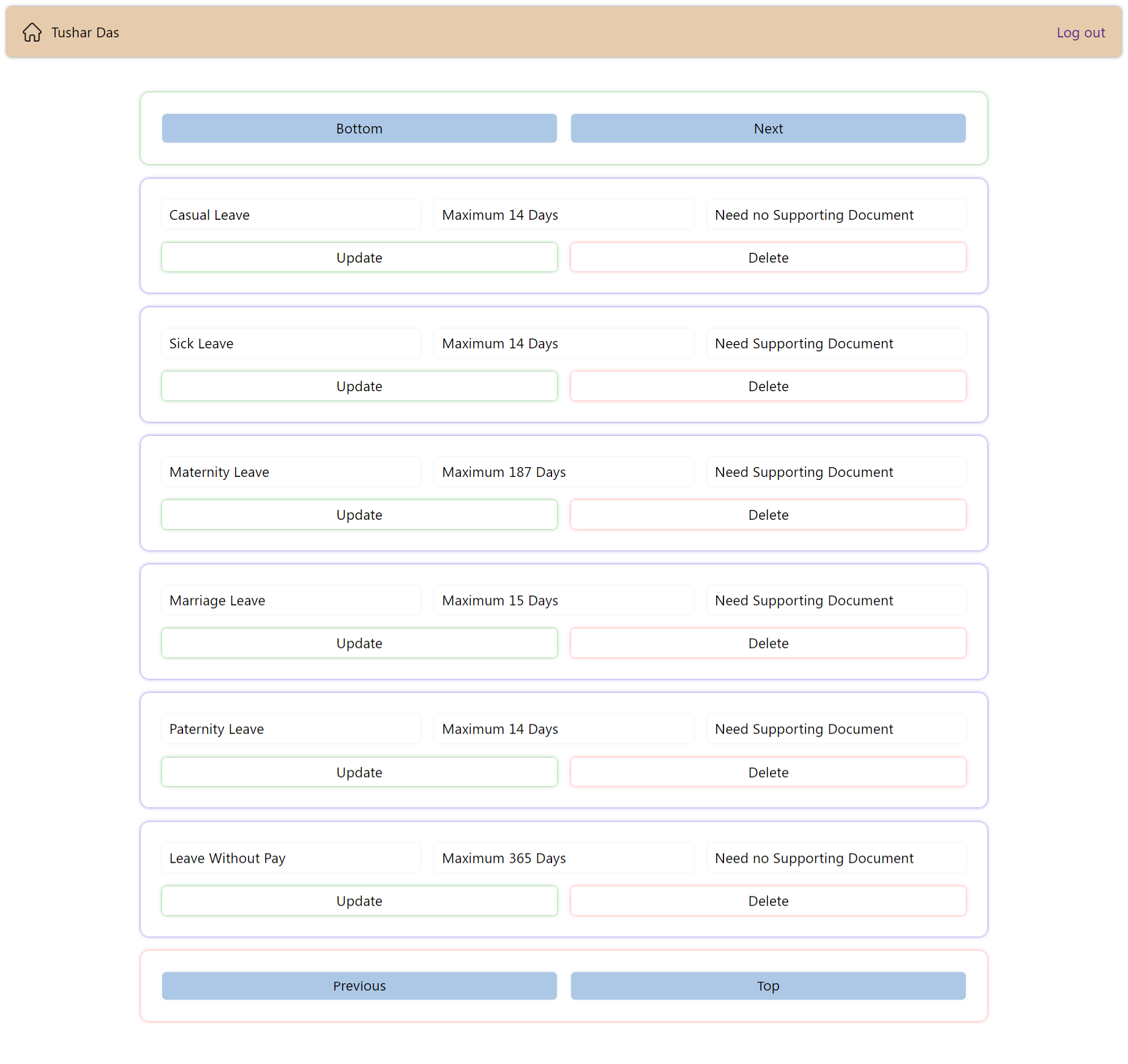
Leave Rules Input

****

List of employee:

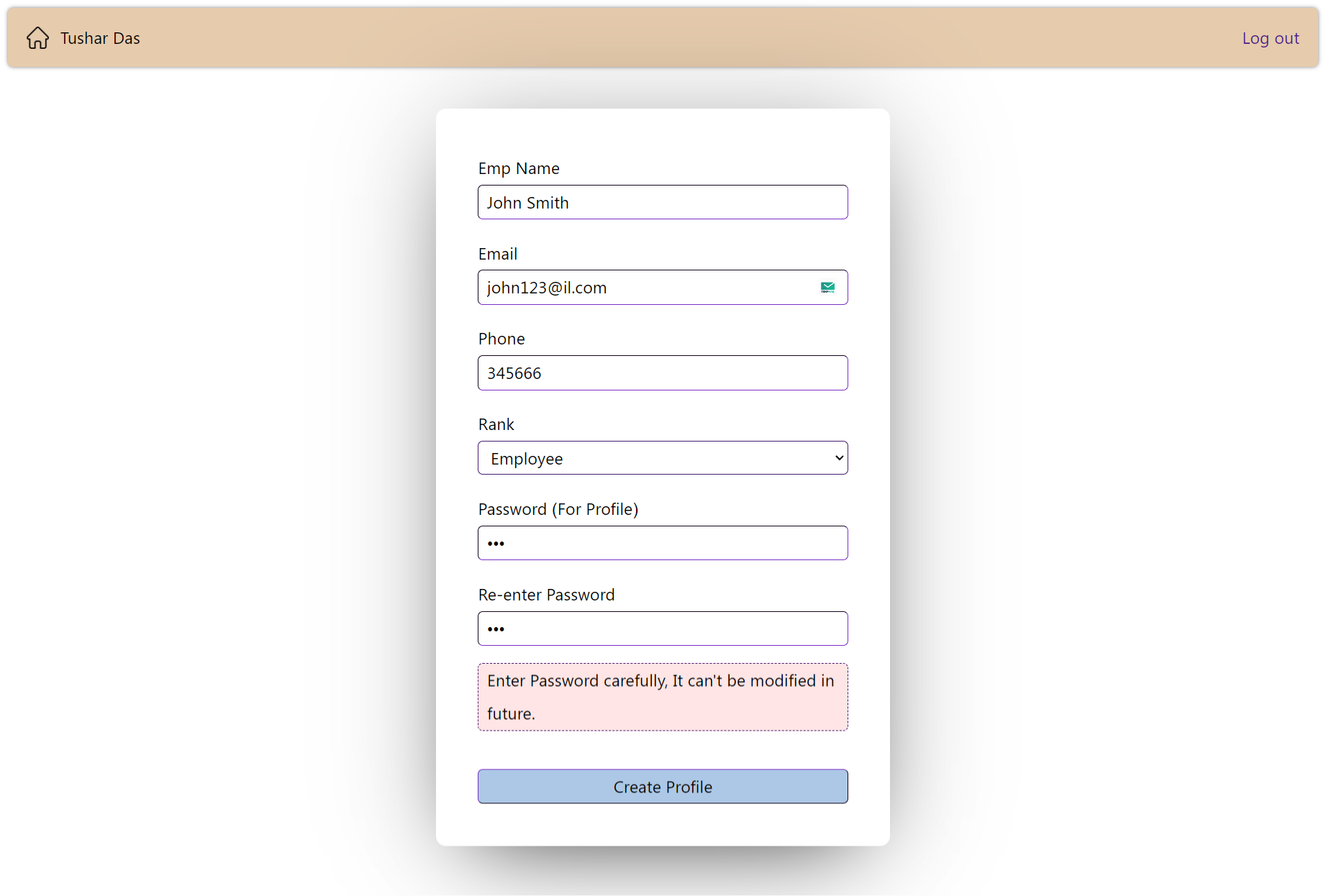
****

List of Leave Rules:

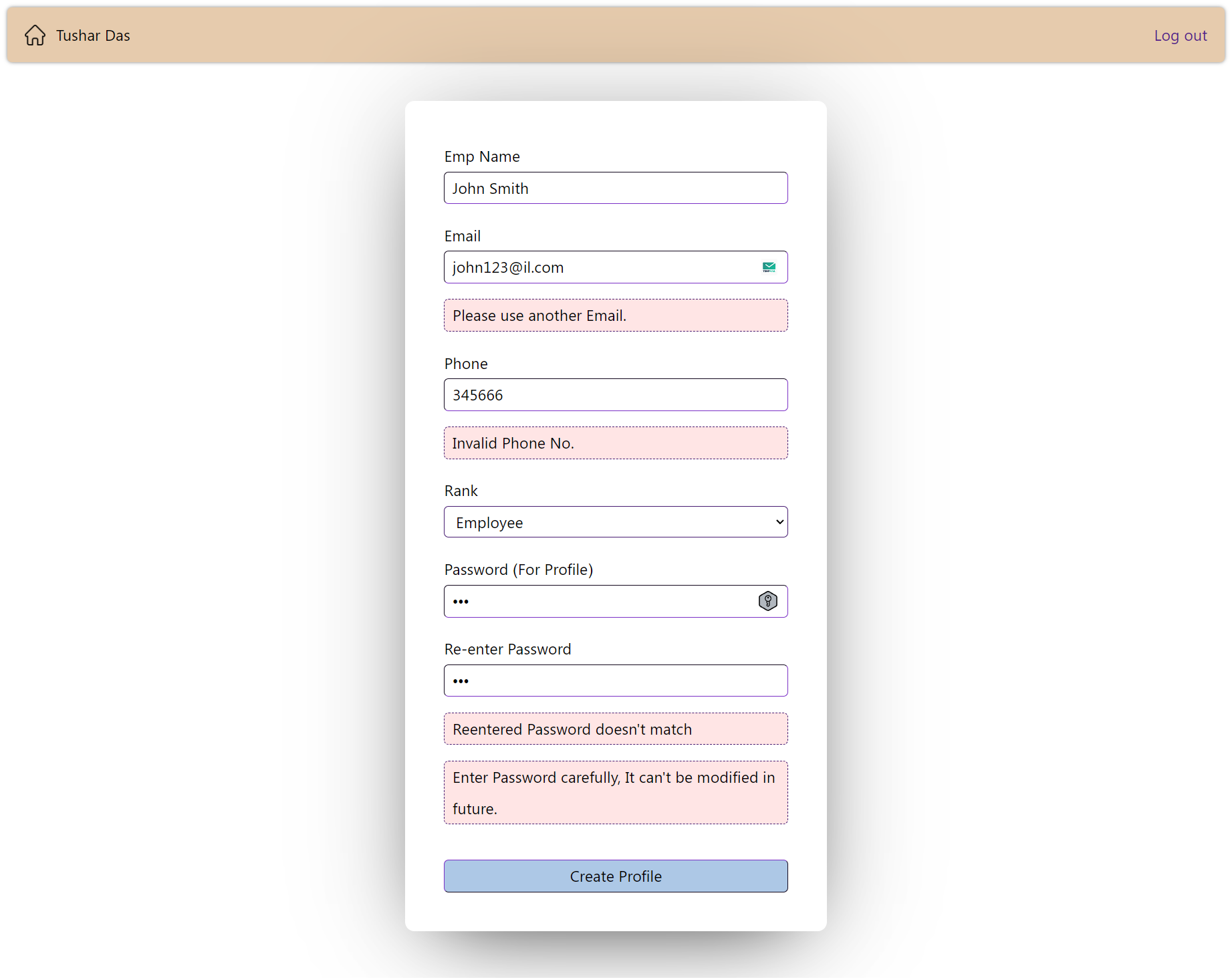
****

**Form input error handling:**

A form with invalid inputs:



After submitting it the error messages are shown along with the previous inputs (after editing the input values one can resubmit them):



**Conclusion:**

**Conclusion:**

This project successfully addressed the inefficiencies of manual leave application processes by developing a completely responsive and mobile-friendly web application. Built from scratch using HTML, CSS, JavaScript, PHP, and MySQL, the application offers a comprehensive suite of functionalities:

* Employees can electronically submit leave requests with supporting documents, track their leave balance, and access the application from any mobile device.
* Managers have a streamlined review process with access to additional employee information for informed decisions.
* A secure, role-based access control system ensures data safety and appropriate user permissions.
* An intuitive UI and robust error handling enhance user experience (though further implementation is planned for all forms).

This application offers significant benefits for both employees and managers, saving time, improving transparency, and streamlining leave management.

**Future Possibilities:**

While this project delivers a robust solution, there's room for further development:

* **Automated notifications:** It can keep employees and managers informed about application status.
* **List of Leaves:** It will allows all users to check out a list of available leaves and leave rules without logging in.
* **Leave trend reporting:** This can provide valuable insights for HR and management.
* **Advanced working day counting:** This feature can automatically consider national holidays and weekly days off when calculating leave entitlement, further enhancing accuracy.
* **Departmental Integration:** Implementing department concepts will allow for a more granular approval hierarchy based on departmental structures.
* **Continuous Optimization:** The implemented systems can be further optimized for improved performance and user experience.

By incorporating these enhancements, the application can become an even more powerful tool for improving leave management within your organization.

**References:**

**Web Development Technologies:**

* **HTML** (<https://developer.mozilla.org/en-US/docs/Learn/HTML>)
* **Cascading Style Sheets (CSS)** (<https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps>)
* **JavaScript** (<https://developer.mozilla.org/en-US/docs/Web/JavaScript>)
* **PHP Manual** (<https://www.php.net/manual/en/index.php>)
* **MySQL Documentation** (<https://dev.mysql.com/doc/>)

**Security:**

* Recommendation: Include a reference on password hashing in PHP. Here's an example:
  + OWASP - Password Storage Cheat Sheet (<https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html>)