

Online Employee Leave Management System

Diploma in Computer Systems Design 20.2F

Web Application Development
Coursework 2

Supervisor - Mr. Sandaruwan Herath

National Institute of Business Management-Kandy

No.02, Asgiriya Road, Kandy

Group Members:

| Name | Index Number |
|-----------------------|---------------------|
| Navin Chandrasiri | KADCSD202F-006 |
| Vileka Karunarithna | KADCSD202F-002 |
| Jalani Jayathilake | KADCSD202F-005 |
| R. Chamodi Thennakoon | KADCSD202F-004 |
| Supun Wijewardana | KADCSD202F-009 |

Contents

| | |
|--|----|
| Acknowledgement | 04 |
| Introduction..... | 05 |
| Description about the Web Application (JustLeave)..... | 06 |
| How to Run | |
| MySQL Database..... | 11 |
| Laravel Back-end..... | 11 |
| React Front-End..... | 12 |
| ScreenShots of the UI..... | 13 |
| Software/Hardware Used | |
| Macro Look on Architecture..... | 15 |
| React-Redux (Front-End)..... | 16 |
| Tools Used..... | 17 |
| Tools Used (Laravel (Back-end))..... | 19 |
| Conclusion..... | 23 |
| References..... | 24 |

Acknowledgement

We would like to express our gratitude towards Mr. Sandaruwan Herath for their support in the accomplishment of our Web Application Development coursework 2.

Finally, we would like to appreciate all our own members for their support and coordination. We hope that our team members will achieve more in their future endeavours.

Introduction

An employee leave management system is a platform that enables staff of an organisation to easily apply, correctly allocate and grant leave.

In many organisations, staff are entitled to different kinds of leave; sick leave, normal leave, short leave. These leaves are taken and recorded according to the organisation policy. The Human Resource (HR) Department is mostly considered the most important department in every organisation. It is a part of the HR Department to keep all the records of employees.

Most organisations use conventional methods for requesting and managing leave , where the academic staff is required to manually write and submit leave applications to the HR Department. This might take minutes to forward the request to the relevant staff in charge and the approval.

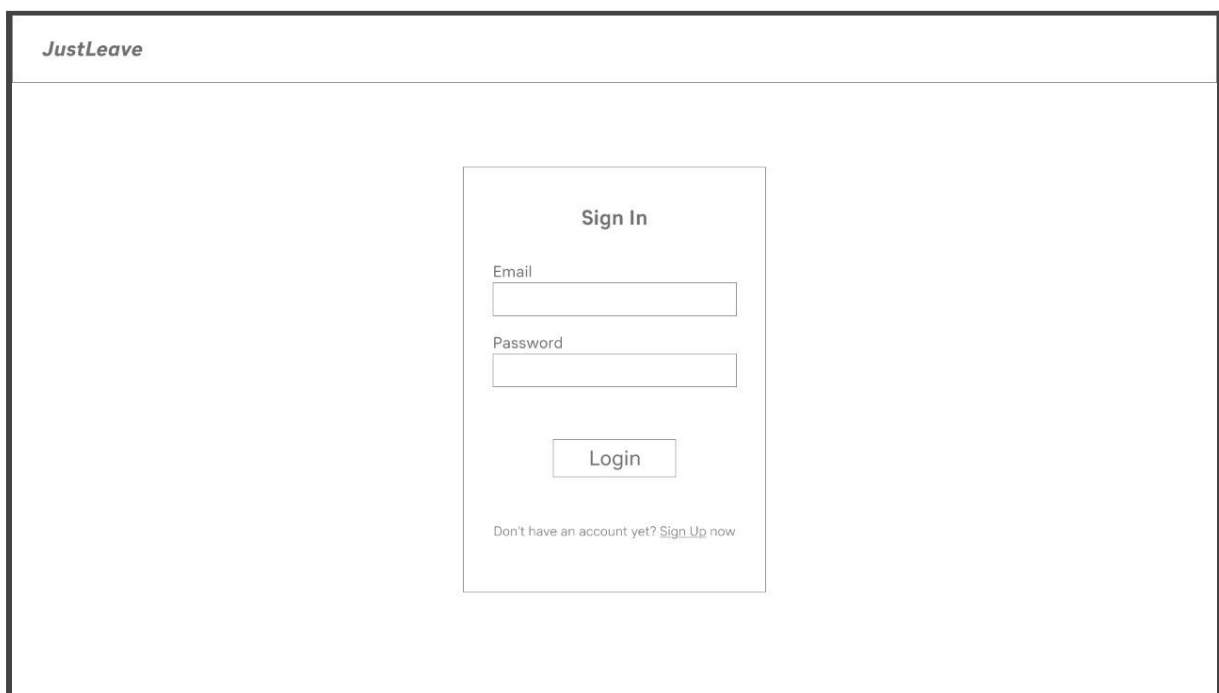
This method is time consuming, requires more paperwork and tedious to maintain. Hence the need for an online website that is faster, error free, with no paperwork and easy to manage.

Description about the web Application (JustLeave) (V.1.0 UI WireFrames included)

“JustLeave” Employee leave Management System is a web based application that can be easily accessed by staff and the HR management of an institution.

A sign-in and sign-up option is available for new users and existing users to login to the system.

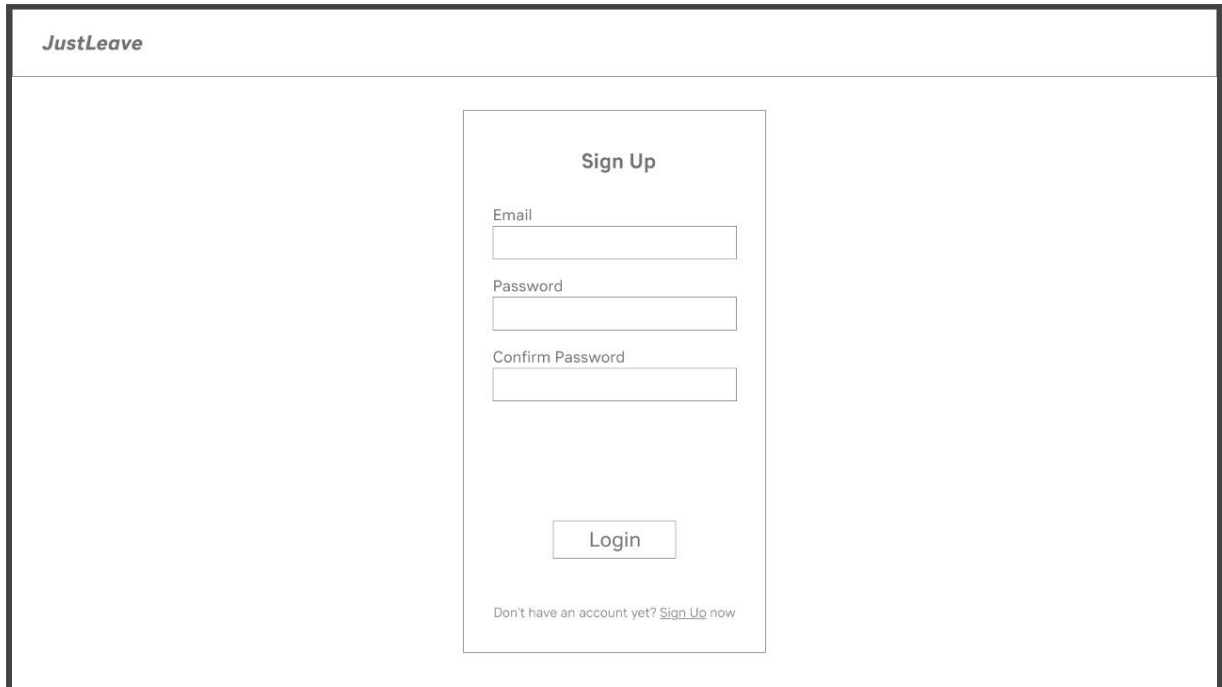
Existing users must enter their email and password to sign-in to the system.



The wireframe shows a web application interface for 'JustLeave'. At the top left, the text 'JustLeave' is displayed. The main content area is a large white rectangle. Centered within this area is a smaller white box with a thin border. Inside this box, the text 'Sign In' is centered at the top. Below it, the label 'Email' is positioned to the left of a rectangular input field. Below the 'Email' field, the label 'Password' is positioned to the left of another rectangular input field. Below the 'Password' field, a rectangular button labeled 'Login' is centered. At the bottom of the box, the text 'Don't have an account yet? [Sign Up](#) now' is displayed, with 'Sign Up' being a clickable link.

New users should create an account in the Employee Management System by entering their email and creating a new password.

They should follow the verification process by re-entering the password for confirmation.



The image shows a web form titled "JustLeave" in the top left corner. The main heading of the form is "Sign Up". Below this heading are three input fields: "Email", "Password", and "Confirm Password". Each field is represented by a text label followed by a rectangular input box. Below these fields is a "Login" button. At the bottom of the form, there is a link that says "Don't have an account yet? [Sign Up](#) now".

JustLeave

Sign Up

Email

Password

Confirm Password

Login

Don't have an account yet? [Sign Up](#) now

It makes it easy for an employee to request and track their own leave.

For every leave requested by an employee, the system will automatically deduct the applicant leave from total leave thereby allowing the employee to view his/her remaining leaves, taken leaves, sick days taken, and short leaves along with dates.

The employee can request a leave by selecting the leave type and the date.

JustLeaveHomeHelpSign Out

Javashan Pythonsinghe
Senior Kottu Inspector

26
Remaining Leaves

4
Taken Leaves

5
Sick Days Taken

5
Short Leaves

Leave Type

Date

Please note that there's a high chance of requests getting declined if they were not requested before 3 days

Request Leave

Requested Leaves

| | | | |
|---------|-------------|------------|----------|
| SH05367 | Short Leave | 12/03/2022 | Pending |
| SH05367 | Short Leave | 12/03/2022 | Declined |
| SH05367 | Sick Leave | 12/03/2022 | Declined |
| SH05367 | Leave | 12/03/2022 | Pending |
| SH05367 | Leave | 12/03/2022 | Approved |
| SH05367 | Short Leave | 12/03/2022 | Approved |
| SH05367 | Short Leave | 12/03/2022 | Approved |
| SH05367 | Short Leave | 12/03/2022 | Approved |

The HR Department on the other hand can easily allocate , grant and manage all leave requests .

The Employee Management system automatically reflects the request to relevant HR for approval or rejection .

JustLeave

Pending LeavesApproved LeavesRejected LeavesHelpSign Out

Human Resource Dashboard

Pending Leaves

Pending Leaves

| LeaveID | Name | Leave Type | Leave Date | Leaves Taken | | |
|---------|-------------------|-------------|------------|--------------|---------|---------|
| SH05367 | David Subassignhe | Short Leave | 12/03/2022 | 25 / 30 | Approve | Decline |
| SH05367 | David Subassignhe | Short Leave | 12/03/2022 | 25 / 30 | Approve | Decline |
| SH05367 | David Subassignhe | Short Leave | 12/03/2022 | 25 / 30 | Approve | Decline |
| SH05367 | David Subassignhe | Short Leave | 12/03/2022 | 25 / 30 | Approve | Decline |
| SH05367 | David Subassignhe | Short Leave | 12/03/2022 | 25 / 30 | Approve | Decline |
| SH05367 | David Subassignhe | Short Leave | 12/03/2022 | 25 / 30 | Approve | Decline |
| SH05367 | David Subassignhe | Short Leave | 12/03/2022 | 25 / 30 | Approve | Decline |

The HR can view the approved and rejected leaves along with the employee name, leave ID, leave type and leave date.

JustLeave

Pending LeavesApproved LeavesRejected LeavesHelpSign Out

Human Resource Dashboard

Approved Leaves

Approved Leaves

| LeaveID | Name | Leave Type | Leave Date | Leaves Taken |
|---------|------------------|-------------|------------|--------------|
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |

JustLeave

Pending LeavesApproved LeavesRejected LeavesHelpSign Out

Human Resource Dashboard

Rejected Leaves

Rejected Leaves

| LeaveID | Name | Leave Type | Leave Date | Leaves Taken |
|---------|------------------|-------------|------------|--------------|
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |
| SH05367 | David Subasignhe | Short Leave | 12/03/2022 | 25 / 30 |

How to run

MySQL Database

As this application uses MySQL, we must create a database in the localhost in our machines. To do that,

first download and install XAMPP from <https://www.apachefriends.org/download.html>. Once installed, open XAMPP and start the Apache and MySQL modules.

Now, we have to set up a database to save the tables of the system. In order to do that, go to <http://localhost/phpmyadmin/> and click on “New” button in the left side pane. Then give the database name as “justleave” click on create. (The tables will be created and dummy data is inserted when setting the backend)

Laravel Backend

The simplest way to setup the Laravel Backend is to set it up with Composer, the PHP package manager. Download the composer windows installer and install it via <https://getcomposer.org/download/>

Then open the backend with Command Line (CMD or powershell or git bash) and run,

```
>> php composer.phar install
```

This command will download and install the dependencies of the project.

Now we have to run the migration file of Laravel. This will ensure creating the required databases of the system automatically. It is essential to keep the apache/mysql localhost active while running the migrations. To run the migrations enter the following command in the Command line

```
>> php artisan migrate
```

Finally, we have to run a seeder to input the dummy values into the database. To run the seeder of this project, execute the following command.

```
>> php artisan db:seed
```

Laravel backend is now fully built.

To run the backend in the localhost, Run the following command:

```
>> php artisan serve
```

React Frontend

To set up React, download and install NodeJs from <https://nodejs.org/en/download/>. Then navigate to the frontend directory via CommandLine and run the following command to install the dependencies.

```
>> npm i
```

Now, the react project is built.

To run the front end in the localhost run the following command:

```
>> npm start
```

To login as the HR manager-

Username: bileka@bileka.com

Password: password

To login as an employee-

Username: ruby@bileka.com

Password: password

Username: talial@bileka.com

Password: password

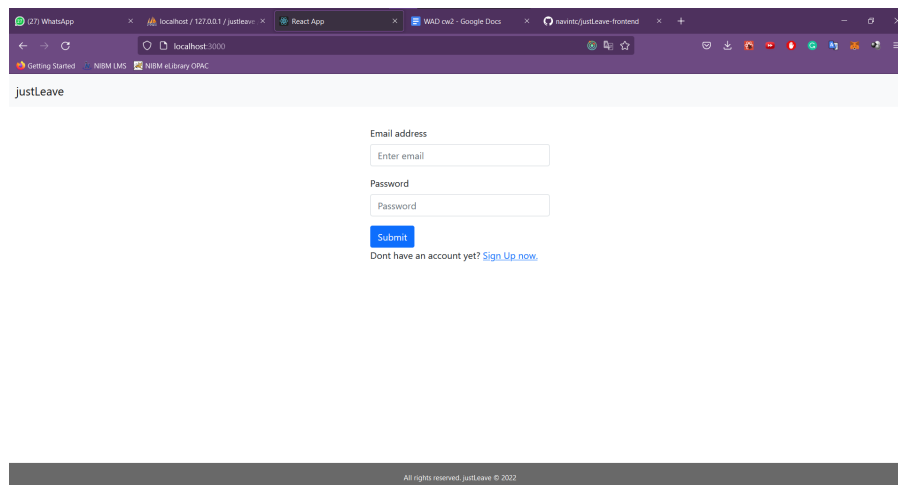
Username: jenkins@bileka.com

Password: password

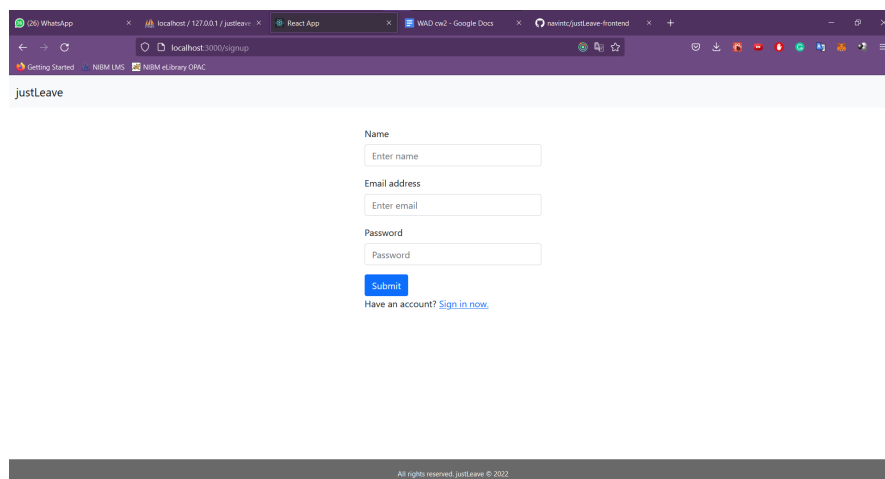
Username: applein@bileka.com

Password: password

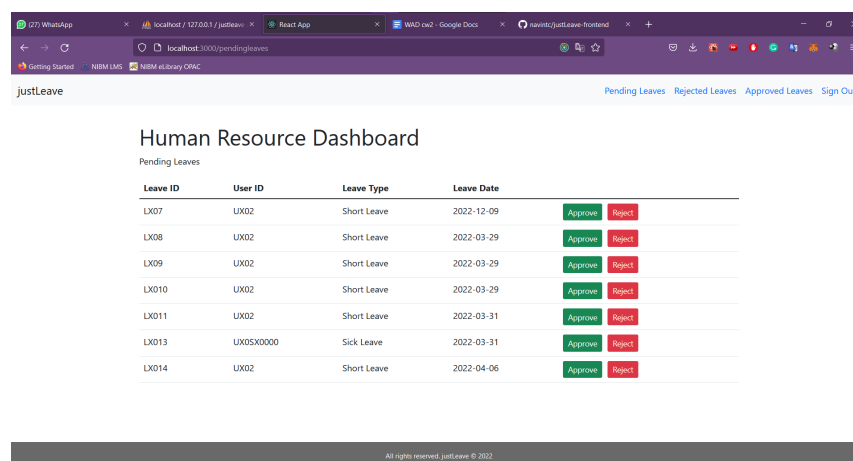
Screenshots:



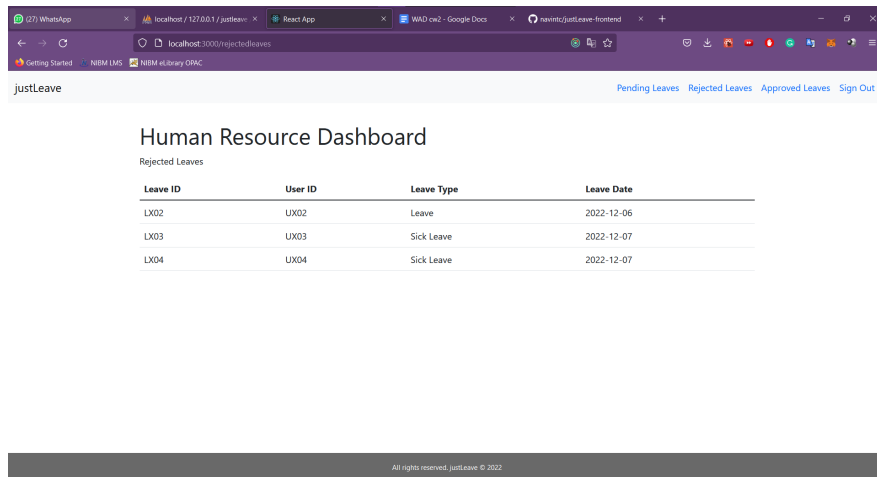
Login



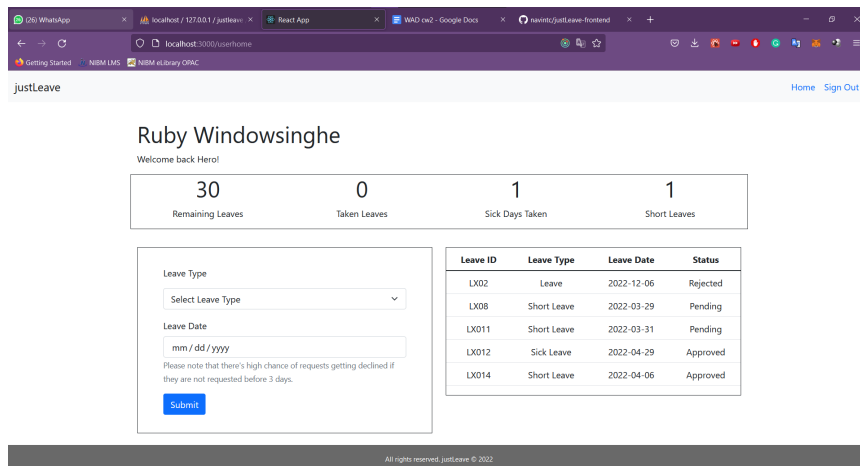
Signup



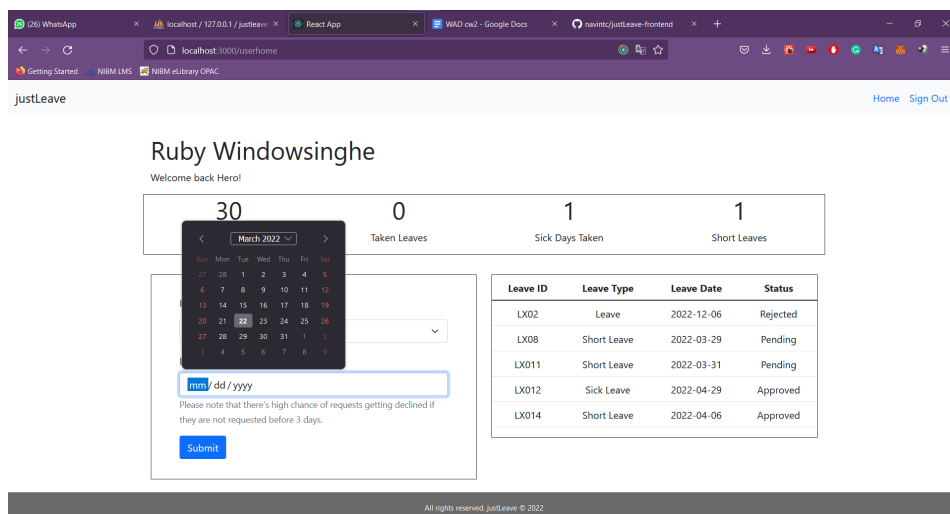
Pending Leaves



Rejected Leaves



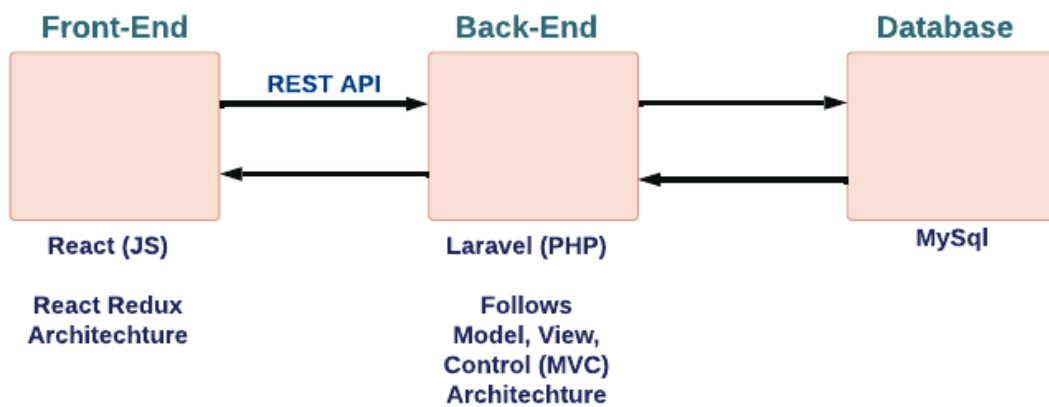
Employee Dashboard



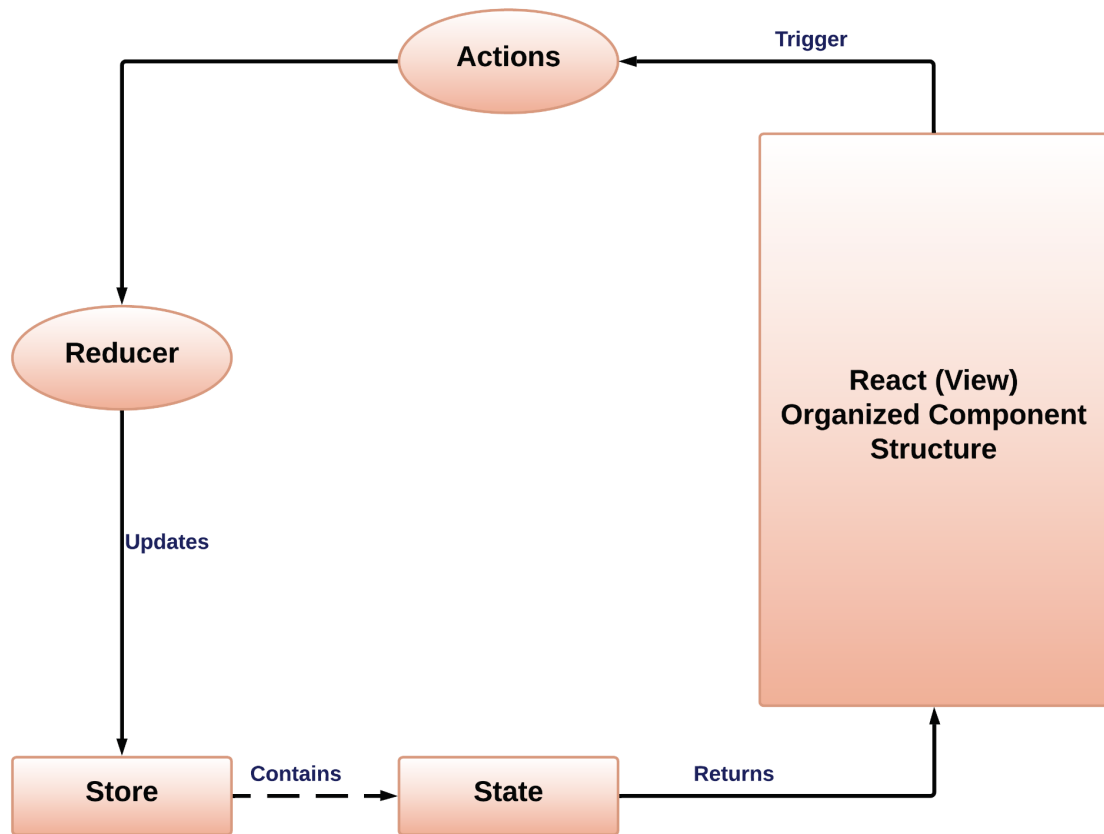
Employee Dashboard - Date Pick

Software/Hardware Used

Macro Look on Architecture



React-Redux (Front-End)



Tools Used ::

1. Use of configs

- A separate document (config.js) has been created to assign variables.
- These variables have a unique name and can be customised by the developer to perform various tasks.
- These variables can be called anywhere in the program to perform the user defined task, which helps to avoid repetition in the entire code.

```
1 //leave statuses
2 export const LEAVE_PENDING = 0;
3 export const LEAVE_APPROVED = 1;
4 export const LEAVE_REJECTED = 2;
5
6 //leaveTypes
7 export const LEAVE_SHORT = 0;
8 export const LEAVE_SICK = 1;
9 export const LEAVE_FULLDAY = 3;
10
11 //userTypes
12 export const HR_MANAGER = 0;
13 export const EMPLOYEE = 1;
```

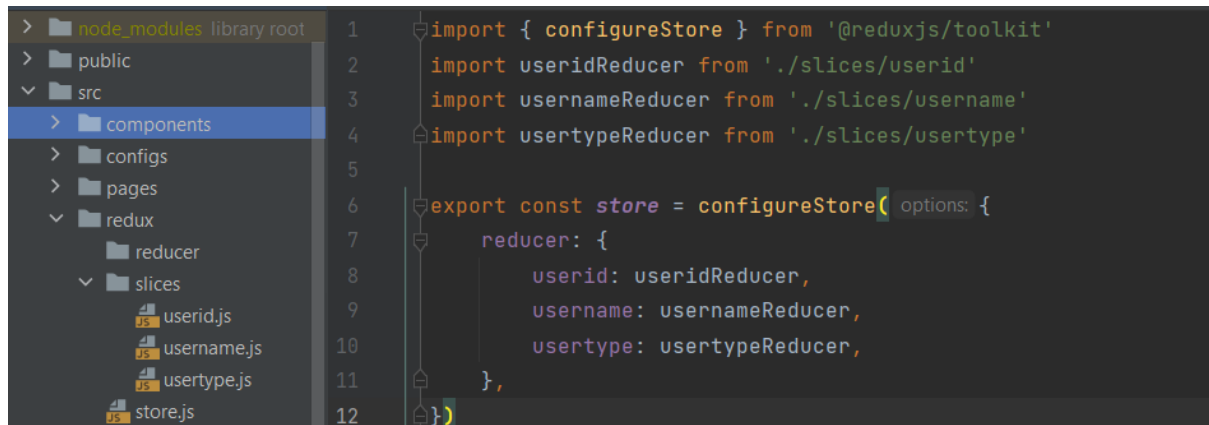
2. Using the npm package React-Router-Dom (<https://reactrouter.com/>)

- React-Router Dom is used to build single page applications.
- It allows you to display pages and allow users to navigate them.
- By applying React-Routers , the page does not have to be refreshed when a link to another page is clicked, instead the content is dynamically fetched based on the URL.

```
16
17
18 function App() {
19   return (
20     <div className="">
21       <BrowserRouter>
22         <TopNavbar/>
23         <Routes>
24           <Route path="/" element={<SignIn />} />
25           <Route path="/userhome" element={<UserHome />} />
26           <Route path="/pendingleaves" element={<PendingLeaves />} />
27           <Route path="/approvedleaves" element={<ApprovedLeaves />} />
28           <Route path="/rejectedleaves" element={<RejectedLeaves />} />
29         </Routes>
30         <Footer/>
31       </BrowserRouter>
32     </div>
33   );
34 }
```

3. Use of Slices in React-Redux (<https://react-redux.js.org>)

- A Redux Slice is a collection of reducer logic and actions for a single feature of our app.
- The name “slice” comes from the idea that we’re splitting up the root Redux state object into multiple “slices” of state.



```
1 import { configureStore } from '@reduxjs/toolkit'
2 import useridReducer from './slices/userid'
3 import usernameReducer from './slices/username'
4 import usertypeReducer from './slices/usertype'
5
6 export const store = configureStore({ options: {
7   reducer: {
8     userid: useridReducer,
9     username: usernameReducer,
10    usertype: usertypeReducer,
11  },
12 })
```

4. Used React-Bootstrap (<https://react-bootstrap.github.io/>)

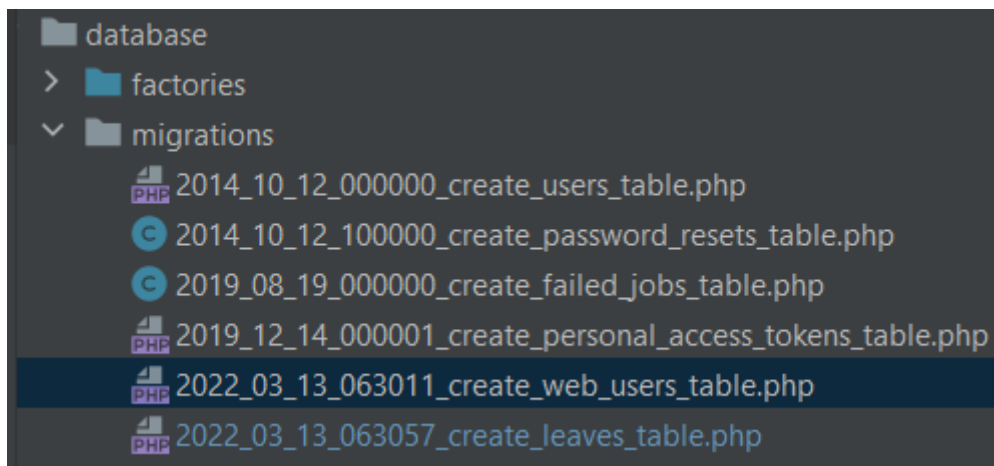
- React Bootstrap package is used to import standard components such as navbars, tables, inputs and date pickers.

Laravel (Back-End)

Tools Used::

1. Migrations

- A migration file is being created and we have defined the structure of the tables we require to create.
- It allows you to create tables in the database, without actually going to the database manager just by running the Migration File.
- If the Database gets Deleted or is no more , we can create the database again by running the Migration File.



```
public function up()
{
    Schema::create( table: 'web_users', function (Blueprint $table) {
        $table->id();
        $table->string( column: 'userID');
        $table->string( column: 'name');
        $table->string( column: 'email');
        $table->timestamp( column: 'email_verified_at')->nullable();
        $table->string( column: 'password');
        $table->integer( column: 'userType');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

2. Seeders

- Despite the fact that we have already created tables in the database using migrations, we do not have any data stored in the tables.
- Thereby Seeders allows us to add data into the tables in the database.

```
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Facades\DB;
8 class LeavesSeeder extends Seeder
9 {
10     /**
11      * Run the database seeds.
12      *
13      * @return void
14      */
15     public function run()
16     {
17         DB::table('leaves')->insert([
18             'leaveID' => "GX0001",
19             'userID' => "1",
20             'leaveType' => 1,
21             'leaveDate' => "2022-12-05",
22             'status' => 0,
23         ]);
24
25         DB::table('leaves')->insert([
26             'leaveID' => "GX0002",
27             'userID' => "2",
28             'leaveType' => 2,
29             'leaveDate' => "2022-12-06",
30             'status' => 0,
```

3. Password Hashing

- It is the process of transforming the string of characters (password) into incomprehensible digits.
- It provides a secure way for storing passwords.
- The process includes verifying the password in the database against the password provided by the user.

```

public function run()
{
    DB::table('web_users')->insert([
        'userID' => "SX0001",
        'name' => "Bileka Karunarathne",
        'email' => "bileka@bileka.com",
        'password' => Hash::make('password'),
        'userType' => 0,
    ]);

    DB::table('web_users')->insert([
        'userID' => "SX0002",
        'name' => "Ruby Windowsinghe",
        'email' => "ruby@bileka.com",
        'password' => Hash::make('password'),
        'userType' => 1,
    ]);
}

```

4. Used Get, Post, and Patch methods

- **Patch** - patch method is used to update entries in a table such as updating a certain column or a row.
- **Get**- It is used to view something.
- **Post**- It is used to change something.

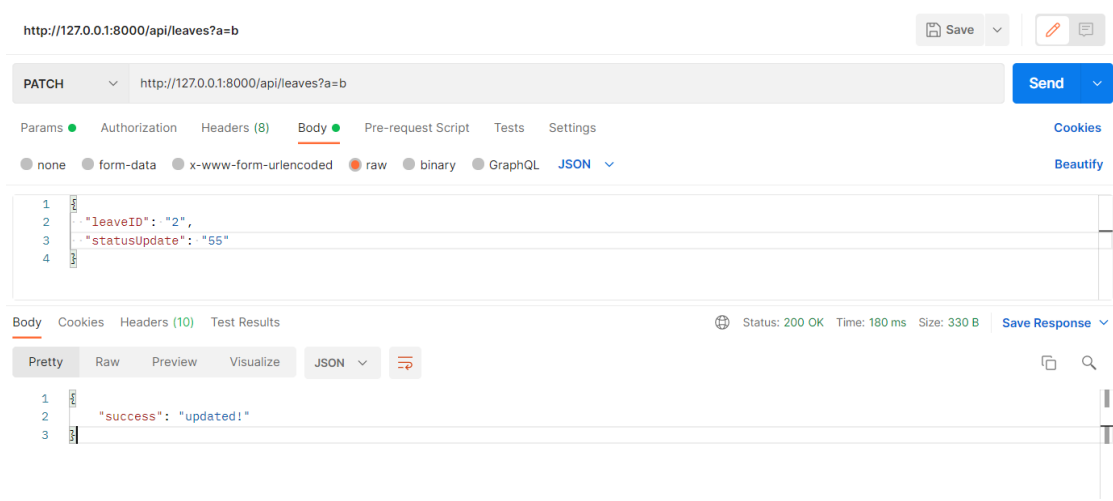
```

28 Route::apiResource( name: '/leaves', controller: '\App\Http\Controllers\LeavesController');
29
30 Route::apiResource( name: '/webuser', controller: '\App\Http\Controllers\WebUserController');
31
32 Route::post( uri: '/login', [WebUserController::class,'login']);
33
34 Route::patch( uri: '/leaves', [LeavesController::class,'updateLeaveStatus']);
35

```

5. Used Postman for testing

- It is used for testing end-points.



Conclusion

In this section, we summarise the things gained by the project team during the development of "JustLeave".

This project gave us a new experience and we have used the knowledge that we gained during our studies to mix up with additional knowledge to create something novel.

As for our achievements and skills that we gained from this project includes --improving communication skills, having more creative insights and learning good coding practices.

References

<https://reactjs.org/docs/getting-started.html>

<https://react-bootstrap.github.io/>

<https://reactrouter.com/docs/en/v6>

<https://react-redux.js.org/introduction/getting-started>

<https://laravel.com/docs/9.x>

https://www.youtube.com/watch?v=j942wKiXFu8&ab_channel=TheNetNinja

<https://www.youtube.com/watch?v=WfpmvgVZD1A&list=PL4cUxeGkcC9h7F1LWaQ7MAI8ptg5VjvxJ>

https://youtu.be/Ty_dCoAnGr0