# SMART LEAVE MANAGEMENT SYSTEM

Design and development of an web portal for managing employees leave online

Protik Chanda,
Roll: 17BT8029,
Department Of Biotechnology,
National Institute Of Technology Durgapur, Durgapur ,West Bengal.

# Project Name: Smart Leave Management System

Intern: Protik Chanda,
Position: Student Intern,
Duration : 24th May - 24th June, 2019,
Place: Bangladesh Computer Council, Agargaon, Dhaka

**Project Coordinator:** Md. Hossain Bin Amin,
Analyst, Information Security,
Bangladesh Computer Council

# Index

## 1.Introduction

This project was aimed to develop such a smart and web based/online system which will provide the facility to the employees of Bangladesh Computer Council or any other organization to apply for casual leaves rapidly and without providing any paperwork to the responsible authority. During this internship I along with my fellow intern developed a web portal under the guidance of Md.Hossain Bin Amin, an Analyst of BCC as a summer student intern developer. It was a web application which we developed with an aim to design and develop such a system which will be very much useful for any type of organization, where employees will be able to apply for leave online from their PC or mobile to their higher authority and in the same way the authorities or admins will be able to manage the leaves of their employees in a efficient way. It was a great learning experience for us to think and develop such a system from ground up to a working prototype. We had to pick the technologies carefully from thousands of available ones, ranging from the front end to the backend and databases. It was a little hurdle for a new learner like me. To keep up with the modern technologies and architecture we rarely sacrificed and tried to use all the modern technologies that are in use in the industries and in the production environment of today’s technological world.

## 2.Goals

For any project to be completed successfully we must have some goals set beforehand which are realistic and contributive in a way to the field of work. So as, while doing the project we had set certain goals to achieve once we finish the task

1. The main goal of the project was to develop a web app/ portal where verified employees of Bangladesh Computer Council can apply for leave smartly and rapidly .
2. Save time and simplify overwhelming manual procedures in office management .
3. Implementing hierarchy based authentication and privileges so that The Director will get applications once they are verified by the departmental heads
4. Design and develop an intuitive UI and UX for the ease of use.
5. Advanced functionalities and privileges for the admins.
6. A secure and encrypted authentication system.
7. Functionalities : Login, Sign Up, Verify, Reset Password, Apply For Leave, Change Department and Designation, Email Verification, Grant or Cancel Application, Admin

## 3.Core Architecture

Before doing any development project it's essential to understand the different software related architecture models available and their differences so that we may find the best suited technologies to build our project on. As a matter of fact the goals of our project easily aligned with most of the modern web and cloud technologies available in the trending industry now and then. So we chose to work with something popular and beginner friendly as well. So we chose the most popular web framework or runtime to be called correctly, Nodejs. We build our app based on the Nodejs (including it's different modules) and MongoDB.

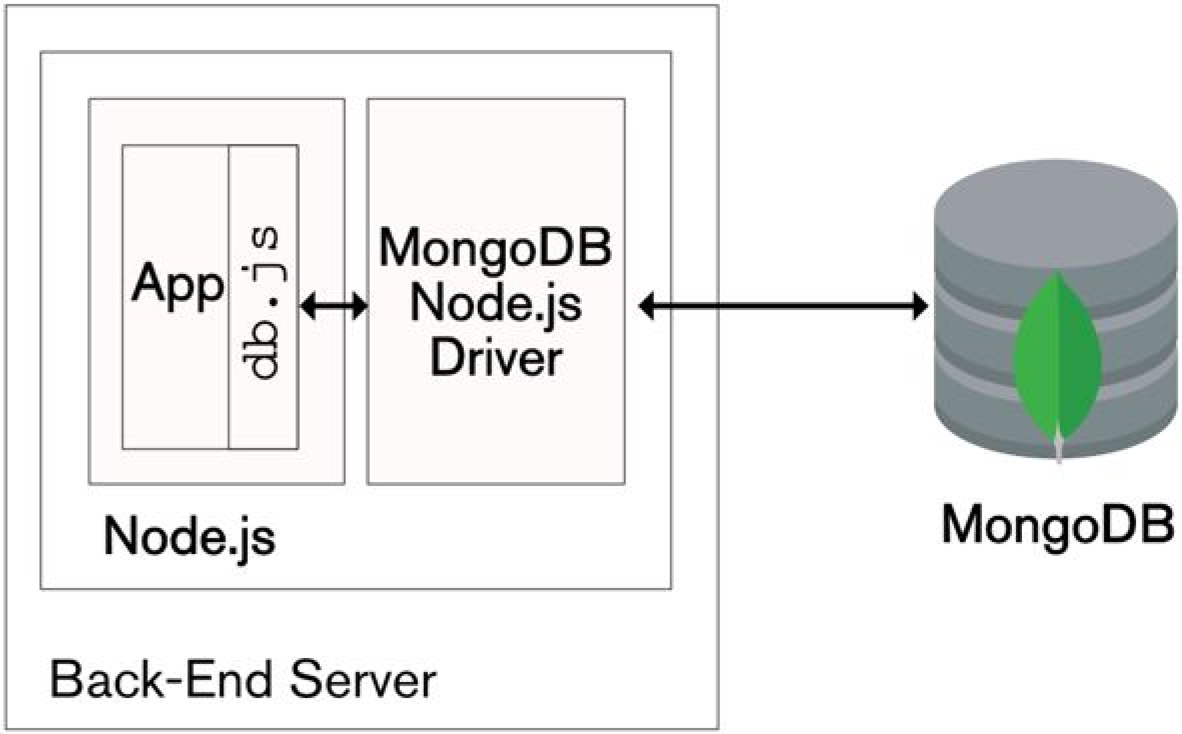


*Image source : MongoDB Blog*

Nodejs is a V8 based JavaScript Runtime Virtual Machine written in C++ and MongoDB is an open source document database. We chose them because they are modern, secure, easily scalable and seamless to work with. Nodejs is very useful in MVC architecture implementation whereas MongoDB makes it easier being a non-sql type database from the hurdles of managing complex RDB.

## 4.Technologies Used

1. Nodejs: Runtime for JS
2. Expressjs: Nodejs module for server
3. Bootstrap: For frontend UI
4. Jquery Date Picker: For Calender
5. Express-session: Session management module
6. EJS Templates: JavaScript based templating system
7. Passportjs: Nodejs authentication module
8. Bcrypt: Tool for encrypting user password
9. Crypto: Module for time sensitive password link generation
10. Nodemailer: Module for email transportation
11. Random string generator for signup verification.
12. MongoDB: As database of choice
13. Mongoose: MongoDB tools
14. MongoDB Compass: Frontend for MongoDB.

### ***NodeJS:***

Nodejs is a cross-platform JavaScript run-time environment that can execute JavaScript codes outside of the traditional web browser into the OS/System itself. It let’s developers use JavaScript to write command line tools and for server-side scripting that’s running server side programs can be achieved by using Nodejs using just the JavaScript language to produce dynamic web pages and apps. Consequently nodejs represents a “JavaScript everywhere” paradigm, unifying full-stack web application development around a single language, rather than old method of using different language in server side and client side scripts and programs.

The special characteristic of nodejs is it has event-driven architecture which is capable of asynchronous I/O

### ***EJS:***

Embedded Javascript Template refers to the client side data binding method implemented with the JavaScript language. This approach became popular thanks to JavaScript's increased use, its increase in client processing capabilities, and the trend to outsource computations to the client's web browser. A frequent practice is to use double curly brackets(i.e. {{key}}) to call values of the given key from data files, often JSON objects.

### ***Bootstrap:***

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

### ***Bcrypt:***

bcrypt is a password hashing function designed by Niels Provos and David Mazières, based on the Blowfish cipher, and presented at USENIX in 1999. Besides incorporating a salt to protect against rainbow table attacks, bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power.

### ***Passport:***

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

### ***Nodemailer:***

Nodemailer is a module for Node.js applications to allow easy as cake email sending. The project got started back in 2010 when there was no sane option to send email messages, today it is the solution most Node.js users turn to by default.

### ***Express-session:***

HTTP is stateless; in order to associate a request to any other request, you need a way to store user data between HTTP requests. Cookies and URL parameters are both suitable ways to transport data between the client and the server. But they are both readable and on the client side. Sessions solve exactly this problem. You assign the client an ID and it makes all further requests using that ID. Information associated with the client is stored on the server linked to this ID.

### *Mongodb:*

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL).

### *Mongoose:*

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

### *Gmail SMTP:*

The Simple Mail Transfer Protocol (SMTP) is a communication protocol for electronic mail transmission. As an Internet standard, SMTP was first defined in 1982 by RFC 821, and updated in 2008 by RFC 5321 to Extended SMTP additions, which is the protocol variety in widespread use today. Mail servers and other message transfer agents use SMTP to send and receive mail messages. Proprietary systems such as Gmail may use non-standard protocols internally, but all use SMTP when sending to or receiving email from outside their own systems. SMTP servers commonly use the Transmission Control Protocol on port number 25.

## 5. Workflow

1. Created a simple registration form and submitted to mongodb.
2. Published data stored in mongodb into a page.
3. Created schema with unique id hook as email for passport.
4. Encrypted password value with bcrypt.
5. Created passport authentication with email as hook.
6. Created user specific session
7. Created user specific profile
8. Created user specific application form
9. Stored form data in another collection
10. Created admin view with access to new collection
11. Integrated CRUD for editing application
12. Created status field for updating the user on the status
13. Created department specific admin
14. Created status page for employee to check in on the status of his/her application
15. Created view for Executive Director
16. Created new status so that only forwarded application can be seen by the Executive-Director
17. Integrated nodemailer and secret token for account activation
18. Created password reset view for each level of user
19. Integrated Bootstrap with each level of user for a seamless experience
21. Created a calendar with javascript validation to only count working days leave
22. Created user profile edit functionality

## 6. Backend Design And Architecture

The design and structure of the entire web Stack:

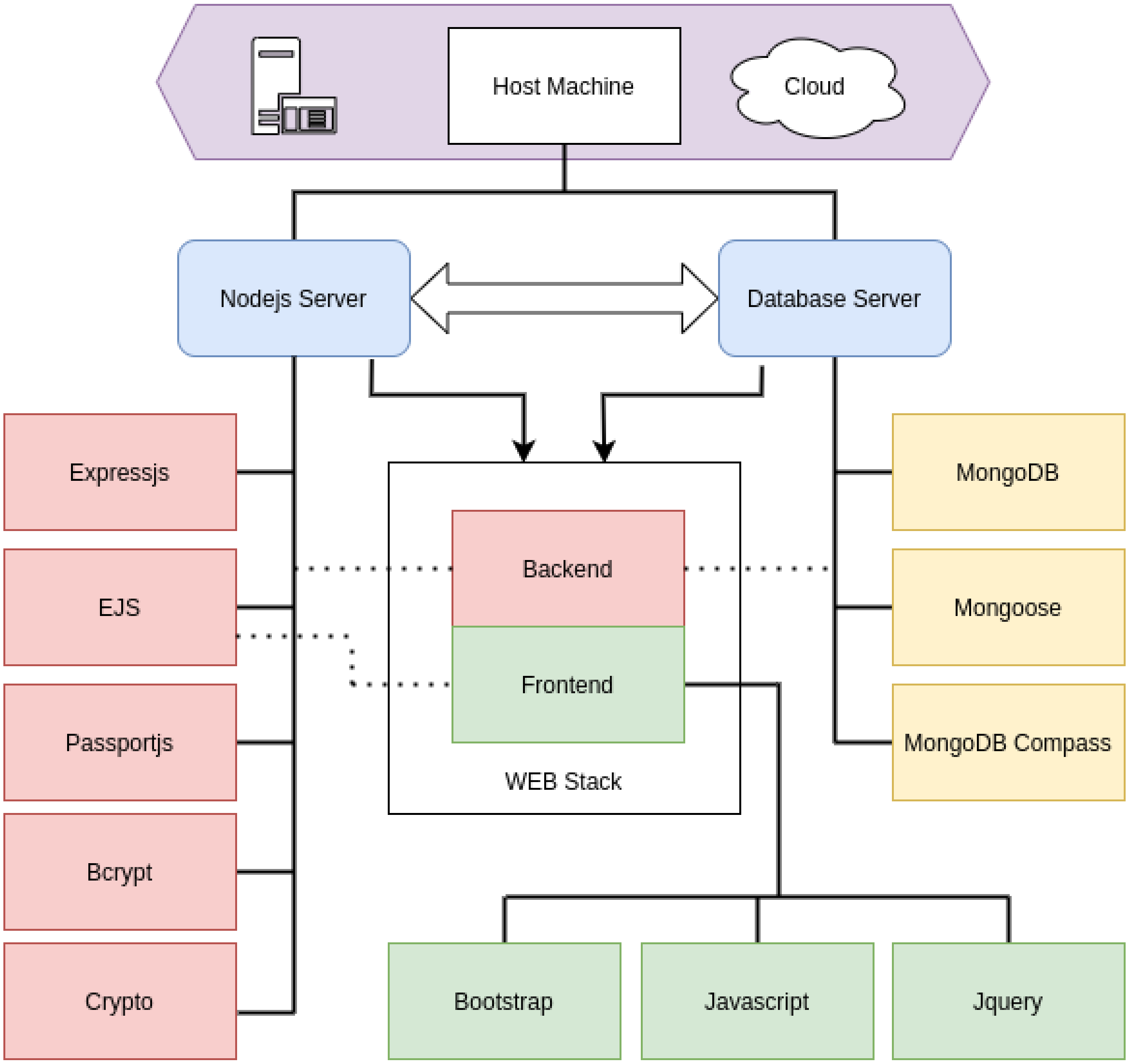
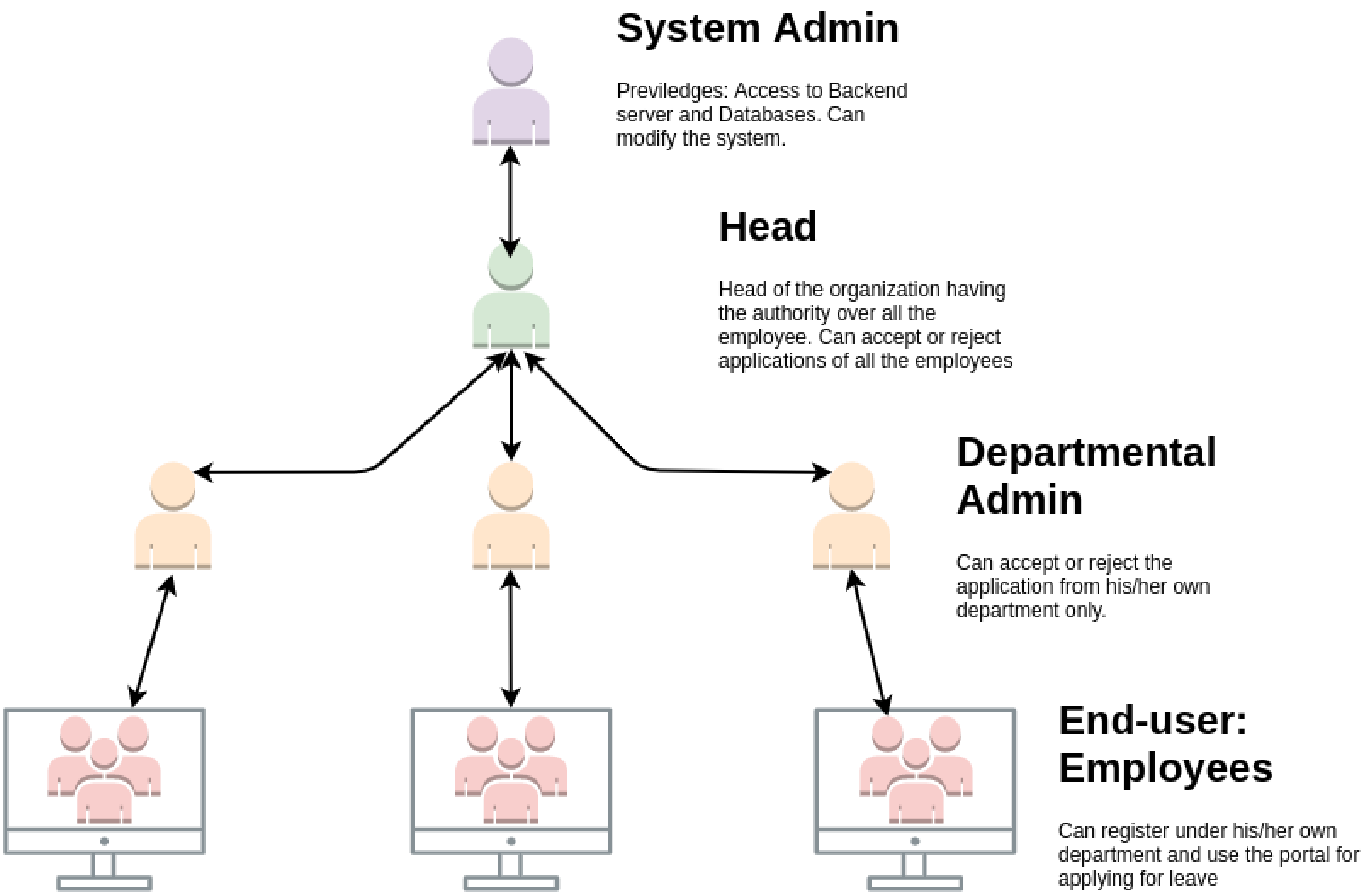


*Fig: System Architecture Diagram*

## 7. User Roles



**System Admin**

Previledges: Access to Backend server and Databases. Can modify the system.

**Head**

Head of the organization having the authority over all the employee. Can accept or reject applications of all the employees

**Departmental Admin**

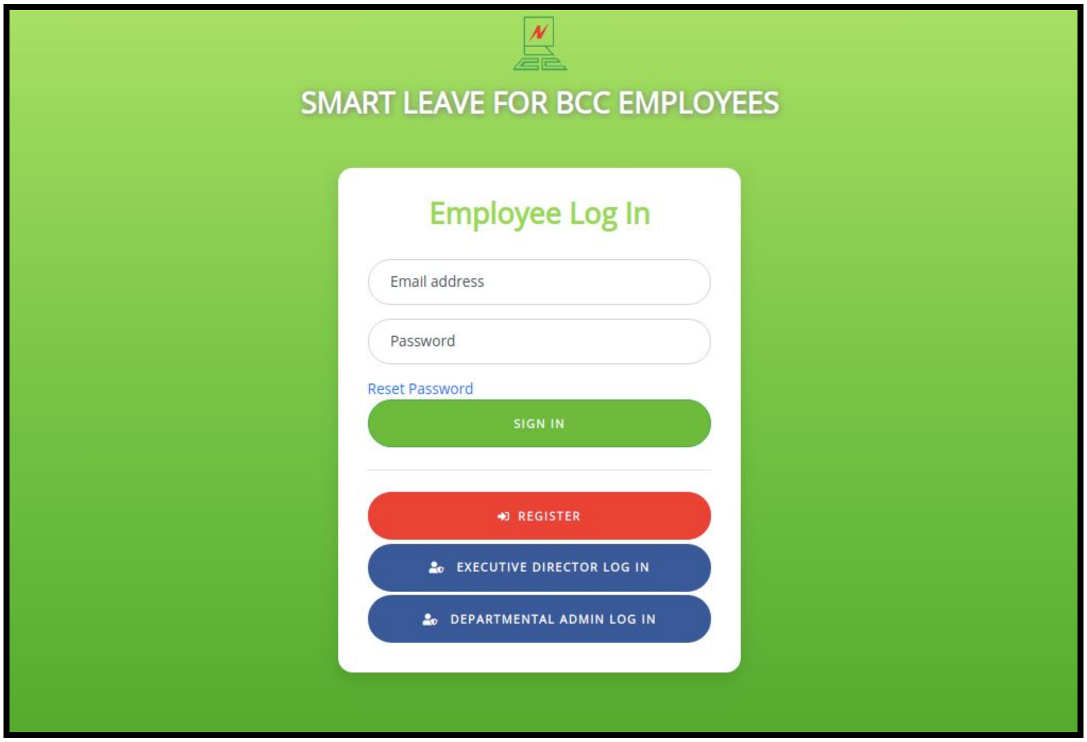
Can accept or reject the application from his/her own department only.

**End-user: Employees**

Can register under his/her own department and use the portal for applying for leave

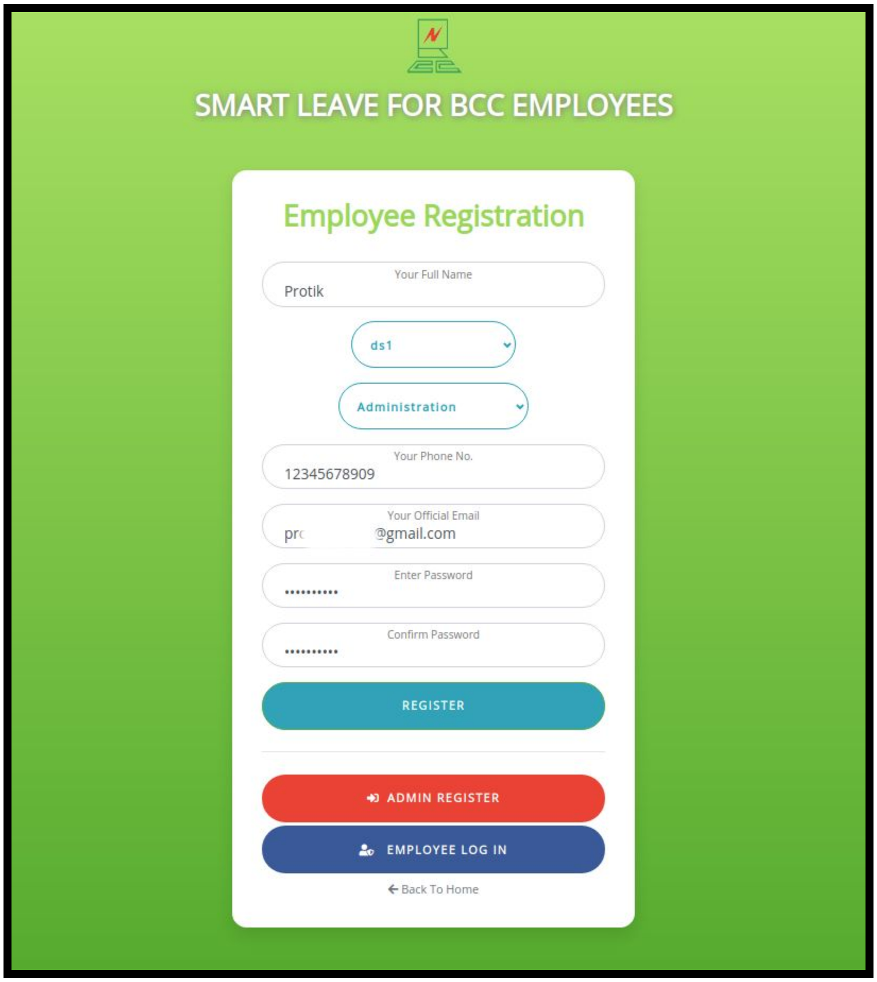
*Fig: User roles chart*

## 8. Functionalities & User Experience(UX)


SMART LEAVE FOR BCC EMPLOYEES
Employee Log In
Email address
Password
Reset Password
SIGN IN
REGISTER
EXECUTIVE DIRECTOR LOG IN
DEPARTMENTAL ADMIN LOG IN

*Home/Main Page*

This is the home page of the application where there’s a form for the registration of the employees with their official credentials. In this form they must use the official email of their organization as predefined. Also this form contains links and buttons to other forms and pages. One thing to be noted that the UI is designed based on Bootstrap, JS , Jquery etc.



SMART LEAVE FOR BCC EMPLOYEES

Employee Registration

Your Full Name: Protik

ds1

Administration

Your Phone No.: 12345678909

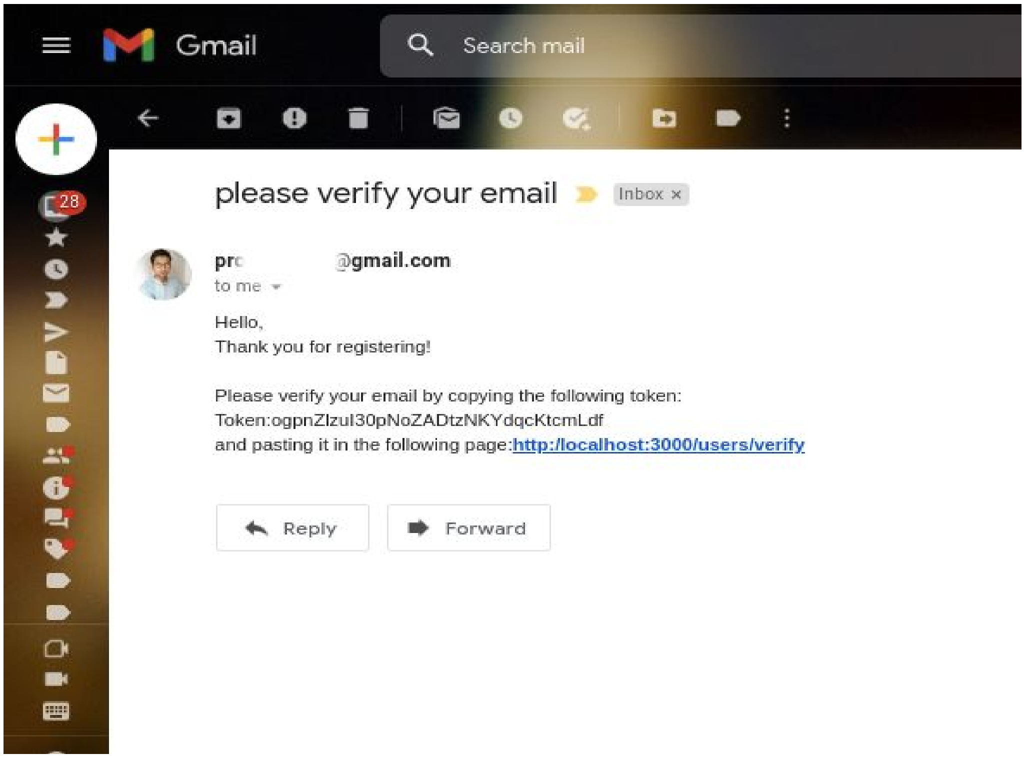
Your Official Email: pr @gmail.com

Enter Password: ••••••••••

Confirm Password: ••••••••••

*Employee Registration Page*

In this page there is a registration form for the employees where they need to fill in some information like Name, Email, Designation, Department etc. Once they fill and submit the form in the application they will be taken to a verification page. For the verification process we have implemented by using nodemailer module and google's smtp protocols.



please verify your email

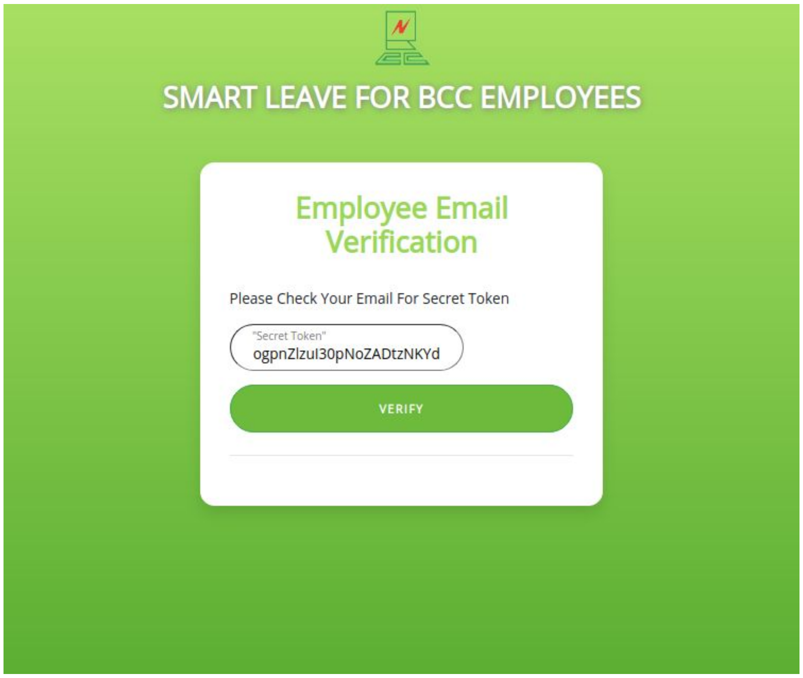
prc @gmail.com
to me

Hello,
Thank you for registering!

Please verify your email by copying the following token:
Token:ogpnZlzul30pNoZADtzNKYdqcKtcmLdf
and pasting it in the following page:http:/localhost:3000/users/verify

*Verification Mail*

An email containing a secret token will be sent upon registration to the registered user for verifying their email and account

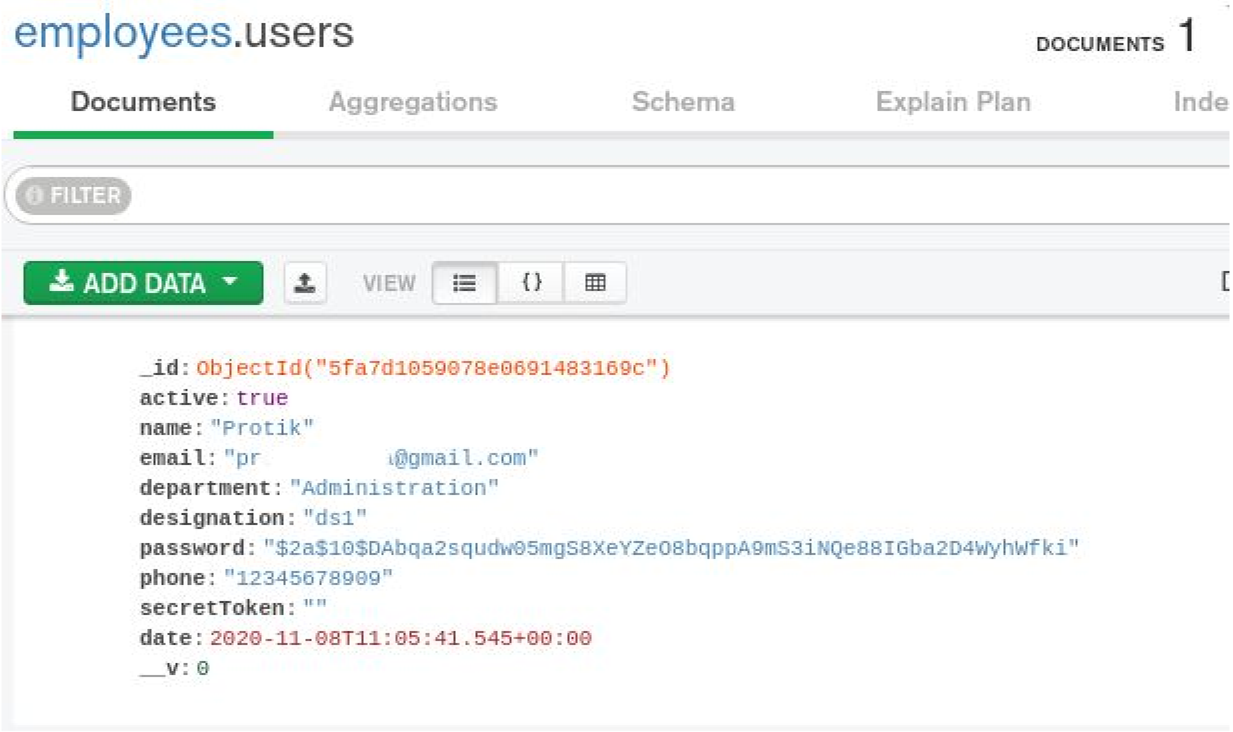


Employee Email Verification

Please Check Your Email For Secret Token

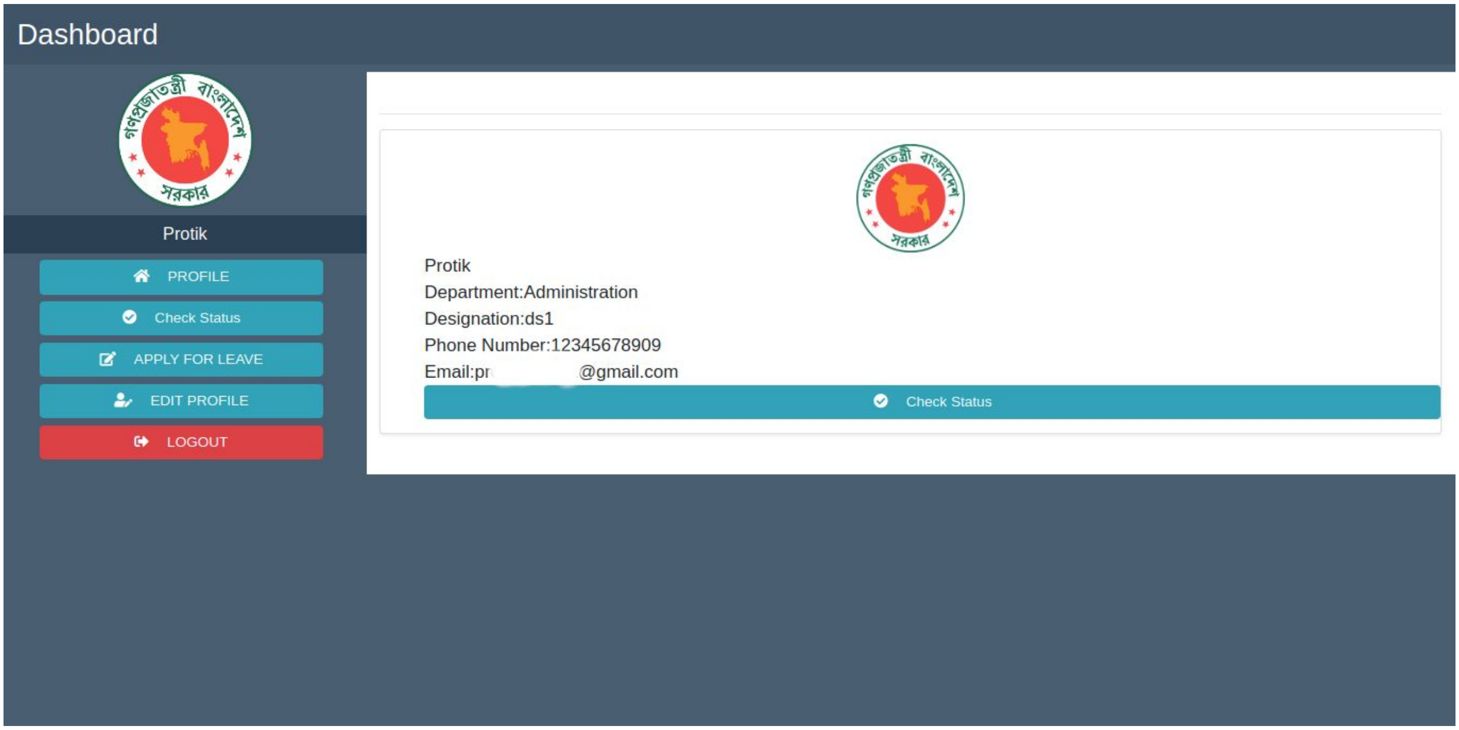
"Secret Token": ogpnZlzul30pNoZADtzNKYd

*Token Verification Page*



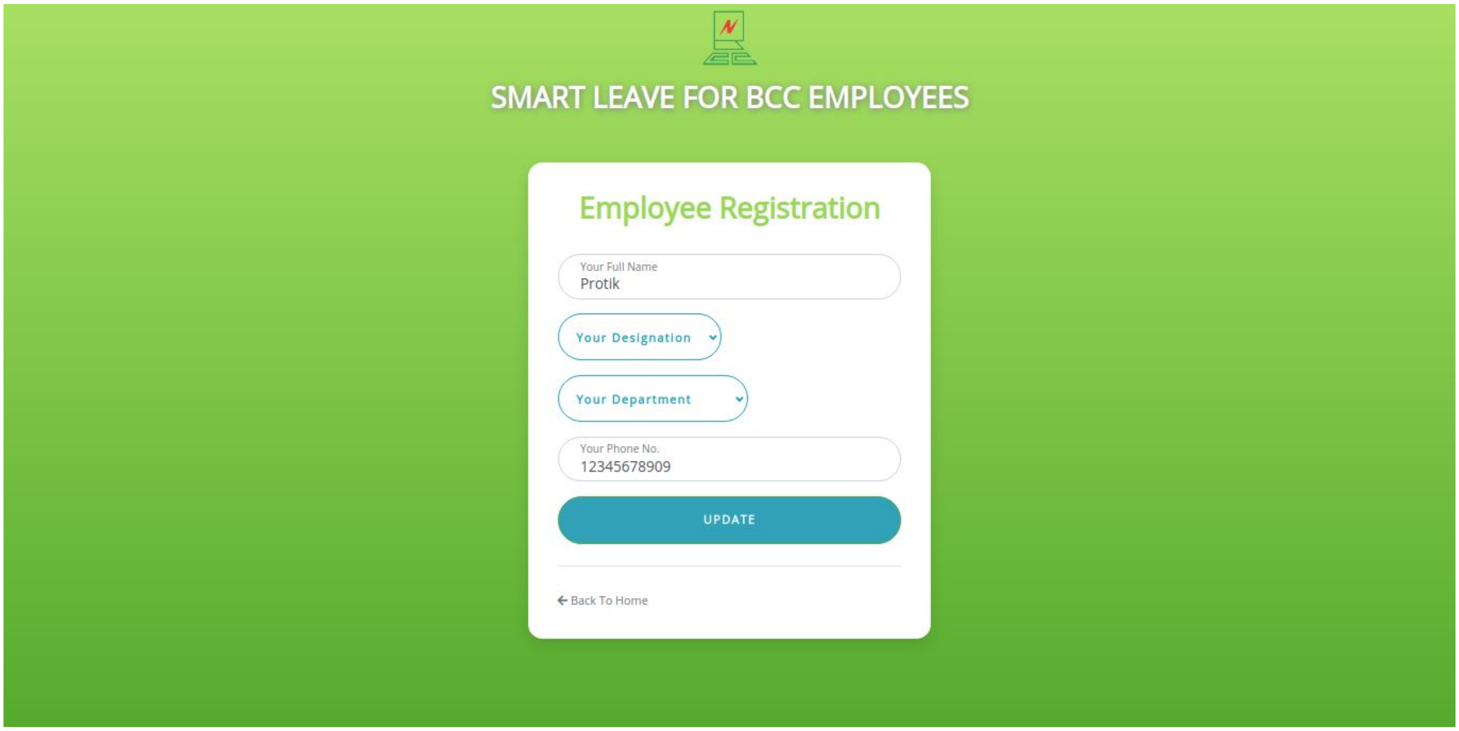
```
_id: ObjectId("5fa7d1059078e0691483169c")
active: true
name: "Protik"
email: "pr          @gmail.com"
department: "Administration"
designation: "ds1"
password: "$2a$10$DAbqa2squdw05mgS8XeYZeO8bqppA9mS3iNQe88IGba2D4WyhWfki"
phone: "12345678909"
secretToken: ""
date: 2020-11-08T11:05:41.545+00:00
__v: 0
```

Upon registration a data document is created in the MongoDB database. The schema is same as shown



Protik
Department:Administration
Designation:ds1
Phone Number:12345678909
Email:pr          @gmail.com

**User Dashboard :** It is the main navigation point and landing page for any logged in user. From here they can access their profile, edit profile, apply and review their leave application and ultimately log out of the app.



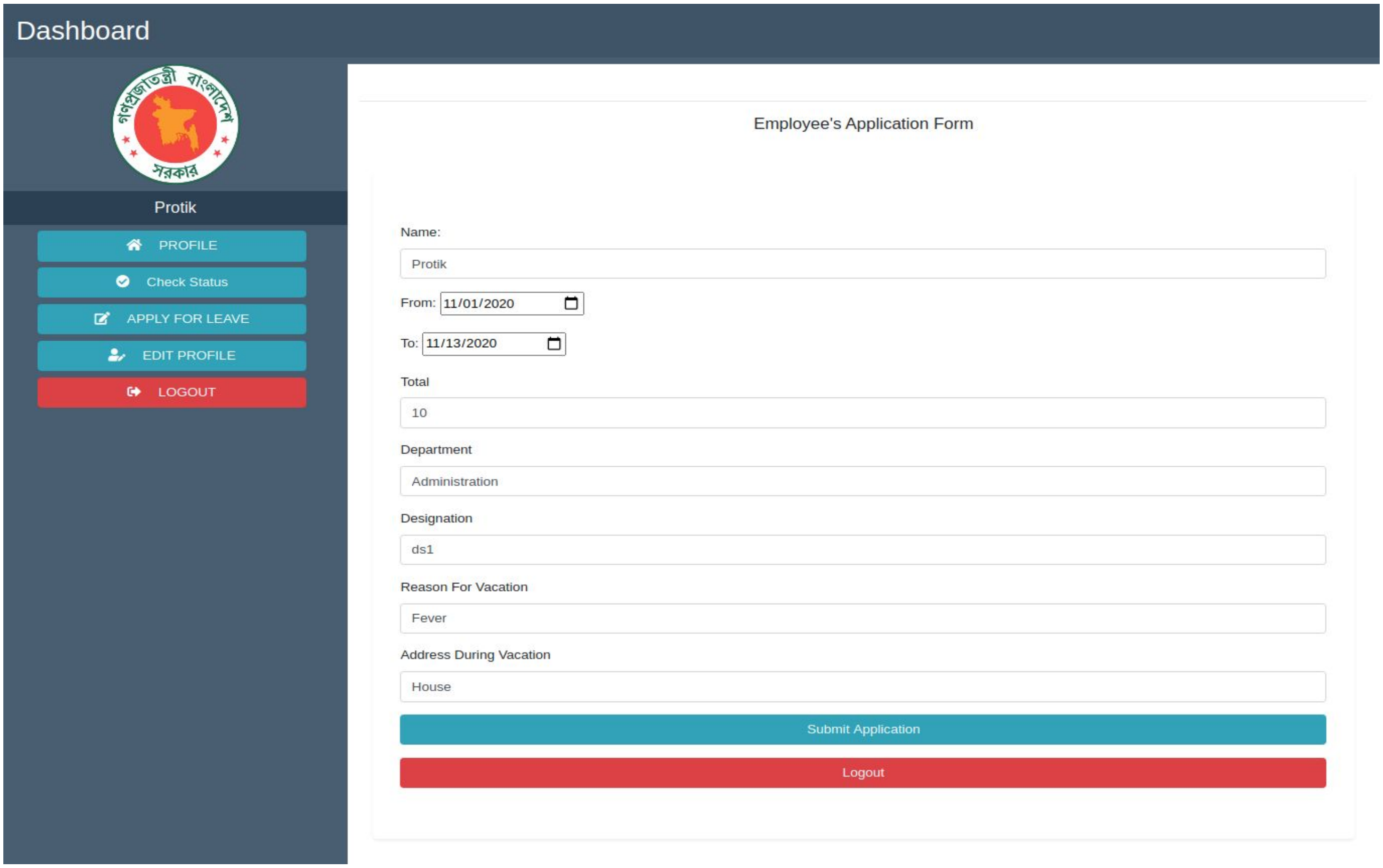
SMART LEAVE FOR BCC EMPLOYEES

Employee Registration

Your Full Name
Protik

Your Phone No.
12345678909

*Employee Profile Update Page*



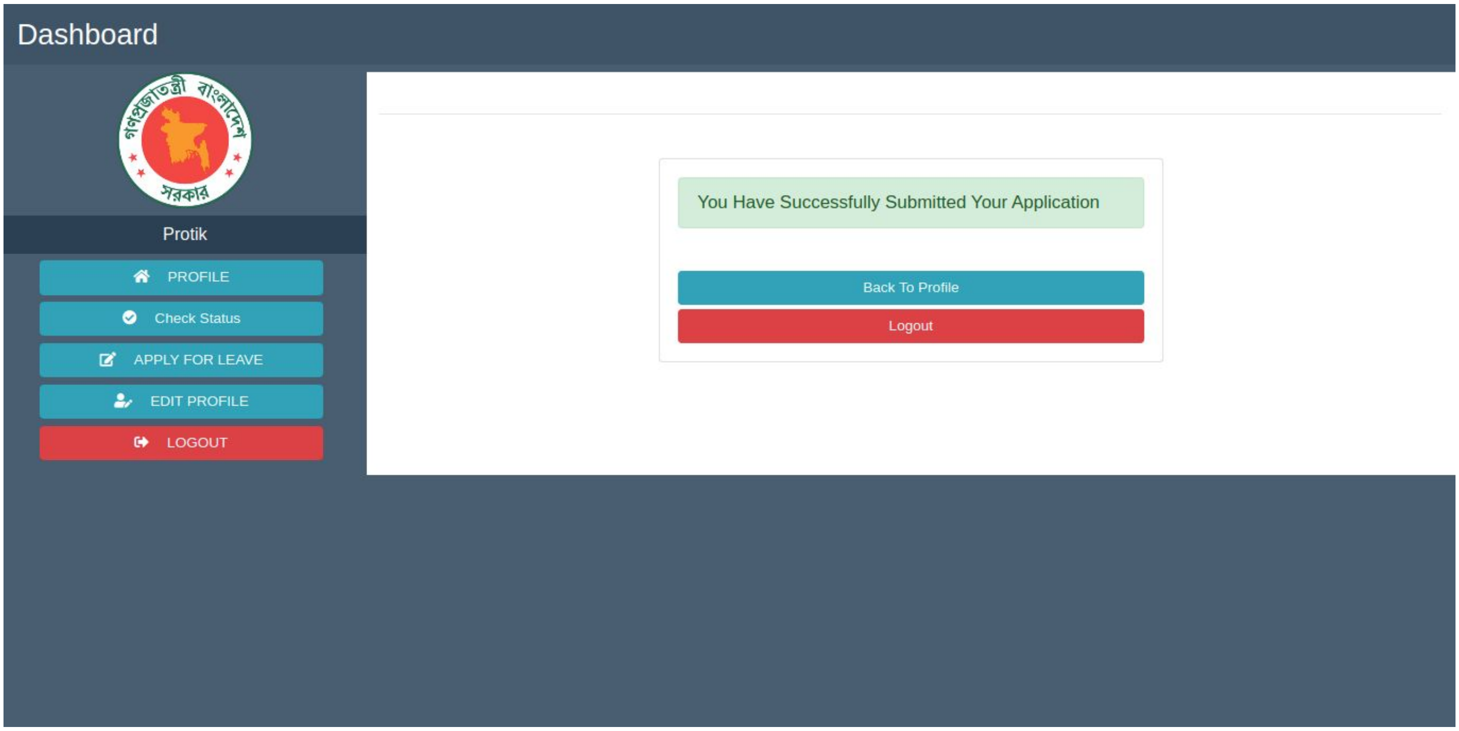
Employee's Application Form

| | |
|---|---|
| Name: | Protik |
| From: | 11/01/2020 |
| To: | 11/13/2020 |
| Total | 10 |
| Department | Administration |
| Designation | ds1 |
| Reason For Vacation | Fever |
| Address During Vacation | House |

*Leave Application Form*

From the dashboard there is a link “Apply For Leave” by using which any employee can apply for a leave for their absence to their higher authority without doing any paperwork. On that page there is a form containing the important information for their application.

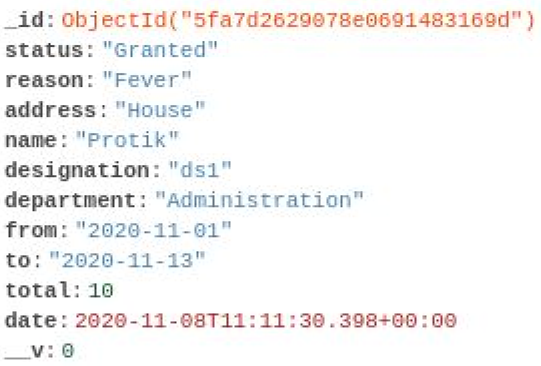
In the application form we have implemented jquery date picker



You Have Successfully Submitted Your Application

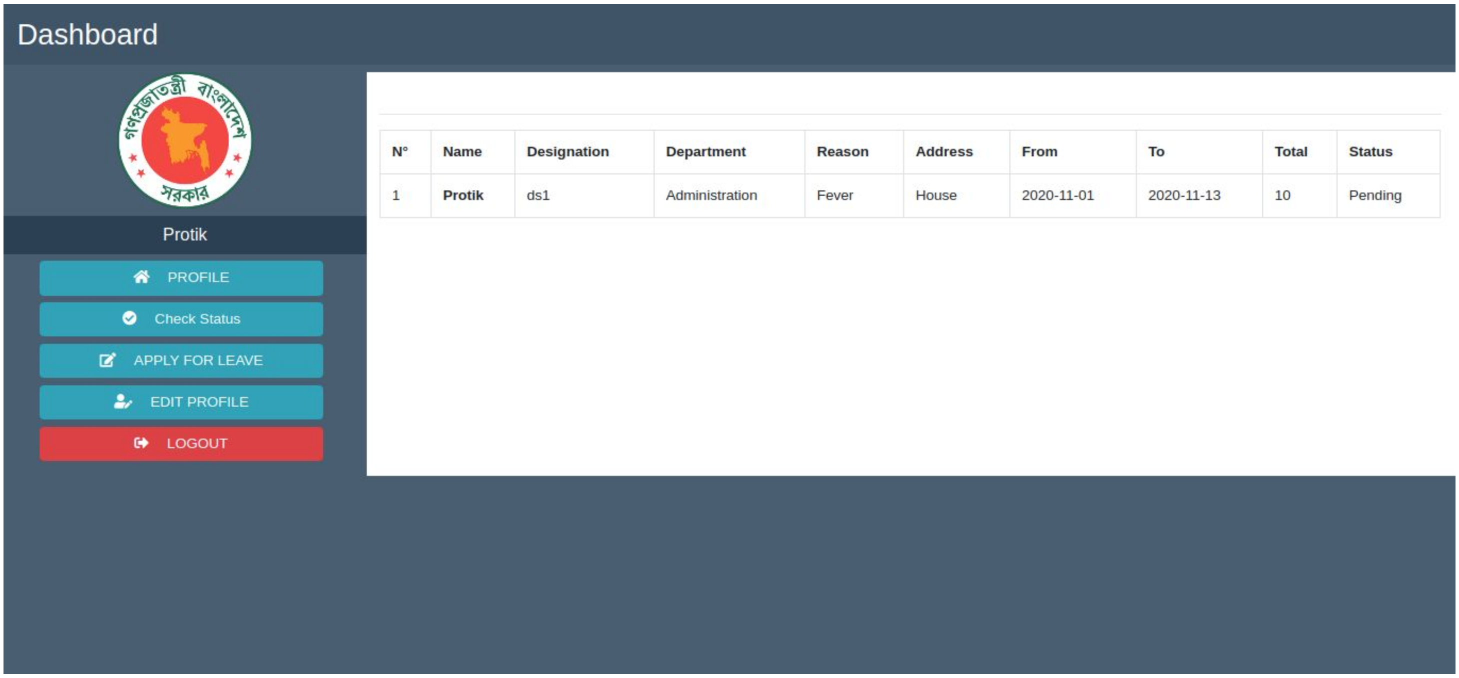
*Confirmation Page*

Once applied for a leave a document is created in MongoDB collection like below



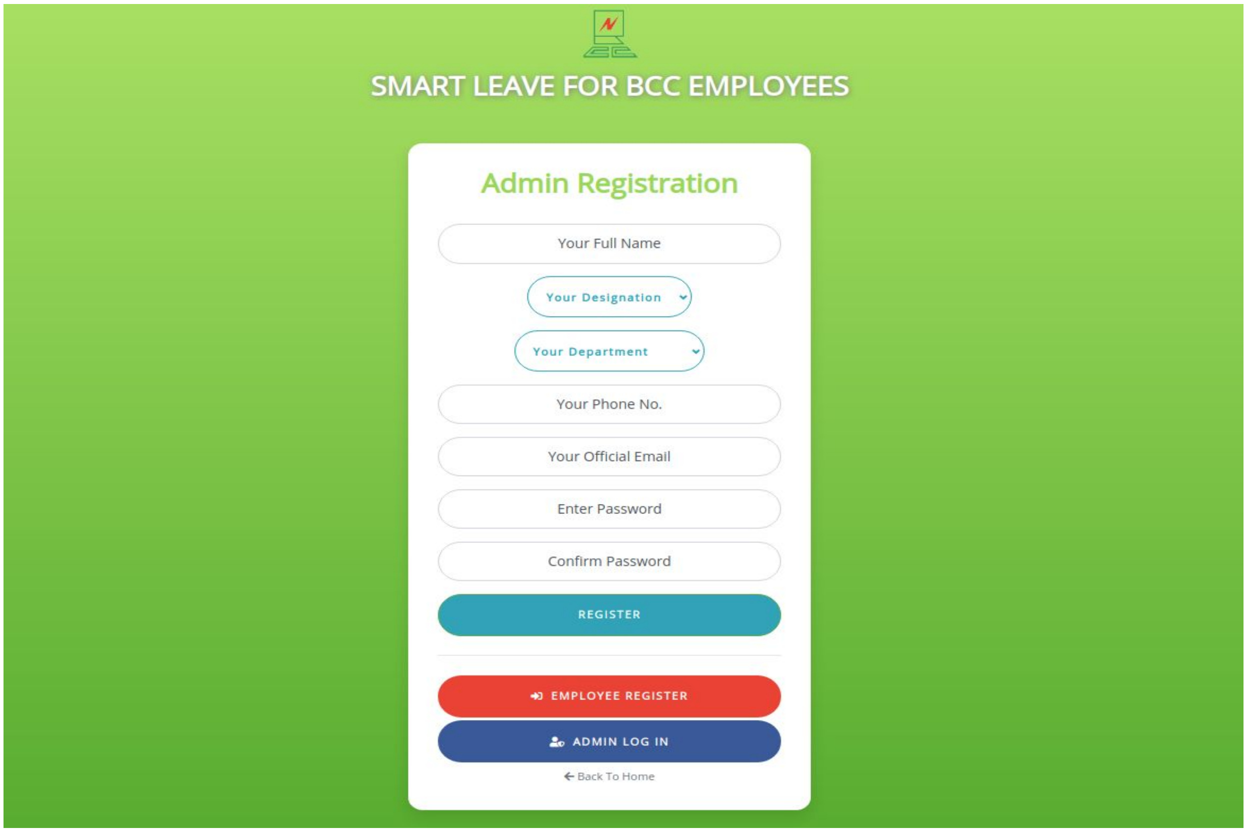
```
_id: ObjectId("5fa7d2629078e0691483169d")
status: "Granted"
reason: "Fever"
address: "House"
name: "Protik"
designation: "ds1"
department: "Administration"
from: "2020-11-01"
to: "2020-11-13"
total: 10
date: 2020-11-08T11:11:30.398+00:00
__v: 0
```

*Application Schema and Data in MongoDB*

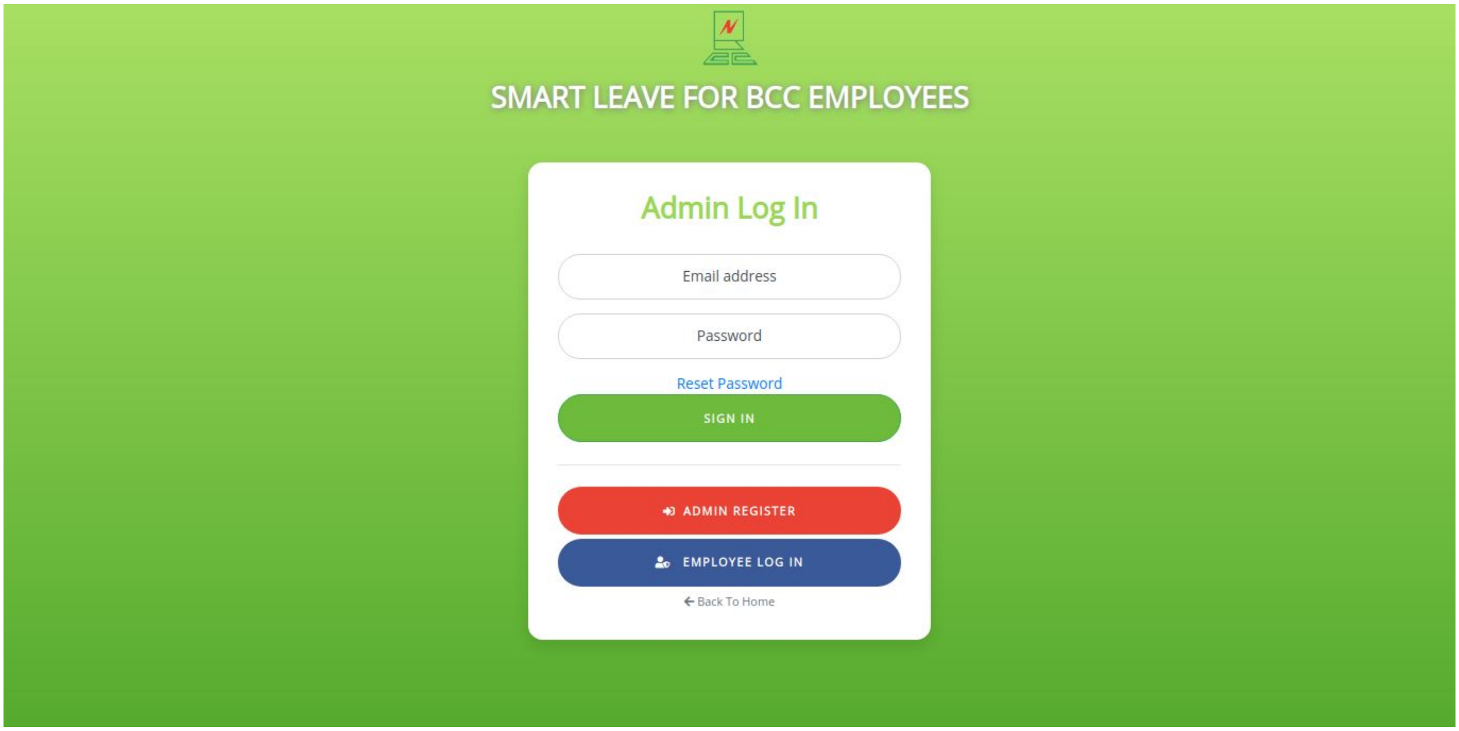


| N° | Name | Designation | Department | Reason | Address | From | To | Total | Status |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **Protik** | ds1 | Administration | Fever | House | 2020-11-01 | 2020-11-13 | 10 | Pending |

*Lists of Applied Leaves*

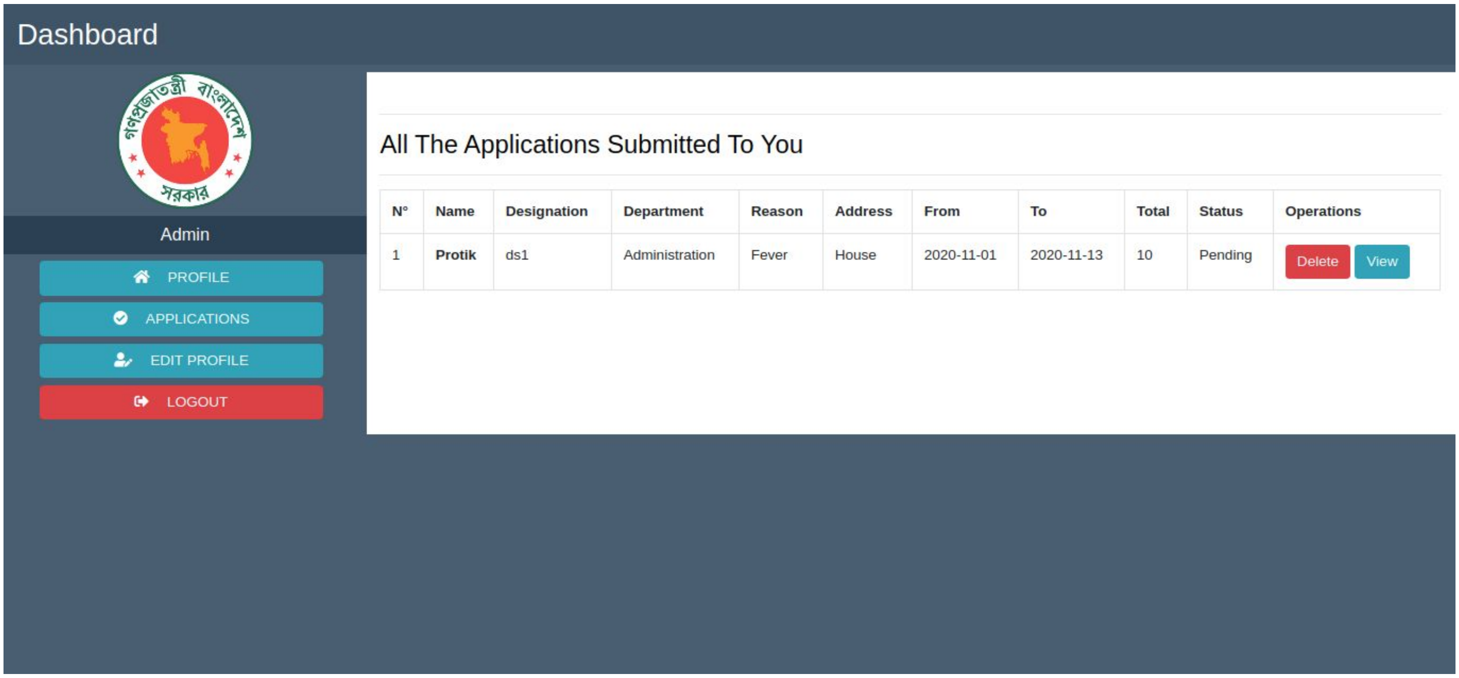


*Admin Registration Form*



*Admin Login*

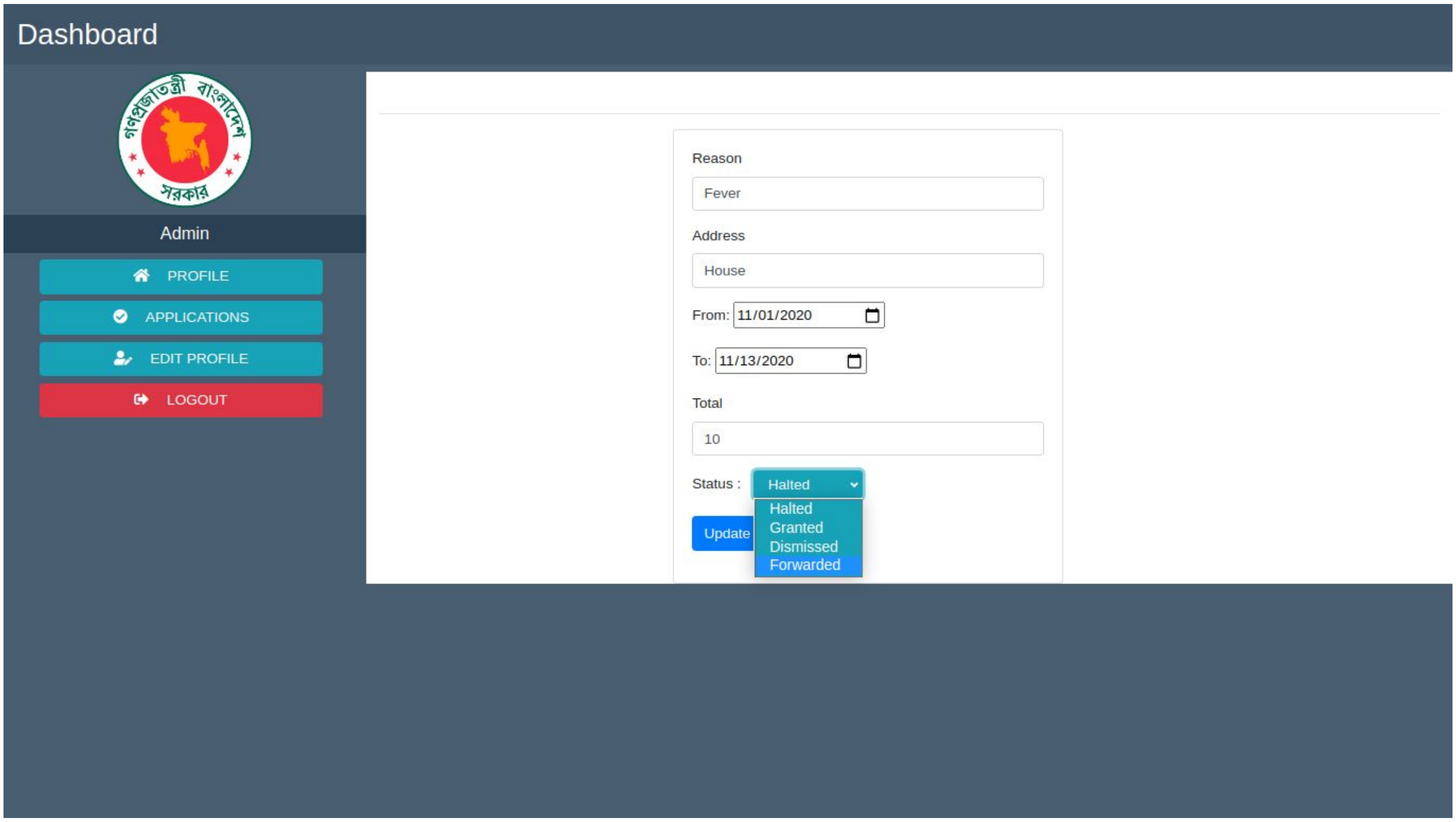
Once an Admin/Head of The Department is registered and logged in in the same way as employees he/she can see all the leave applications directed towards them from their respective departmental employees, asking for leaves



All The Applications Submitted To You

| N° | Name | Designation | Department | Reason | Address | From | To | Total | Status | Operations |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Protik** | ds1 | Administration | Fever | House | 2020-11-01 | 2020-11-13 | 10 | Pending | Delete View |

*Admin's List of Applications*



Reason: Fever

Address: House

From: 11/01/2020

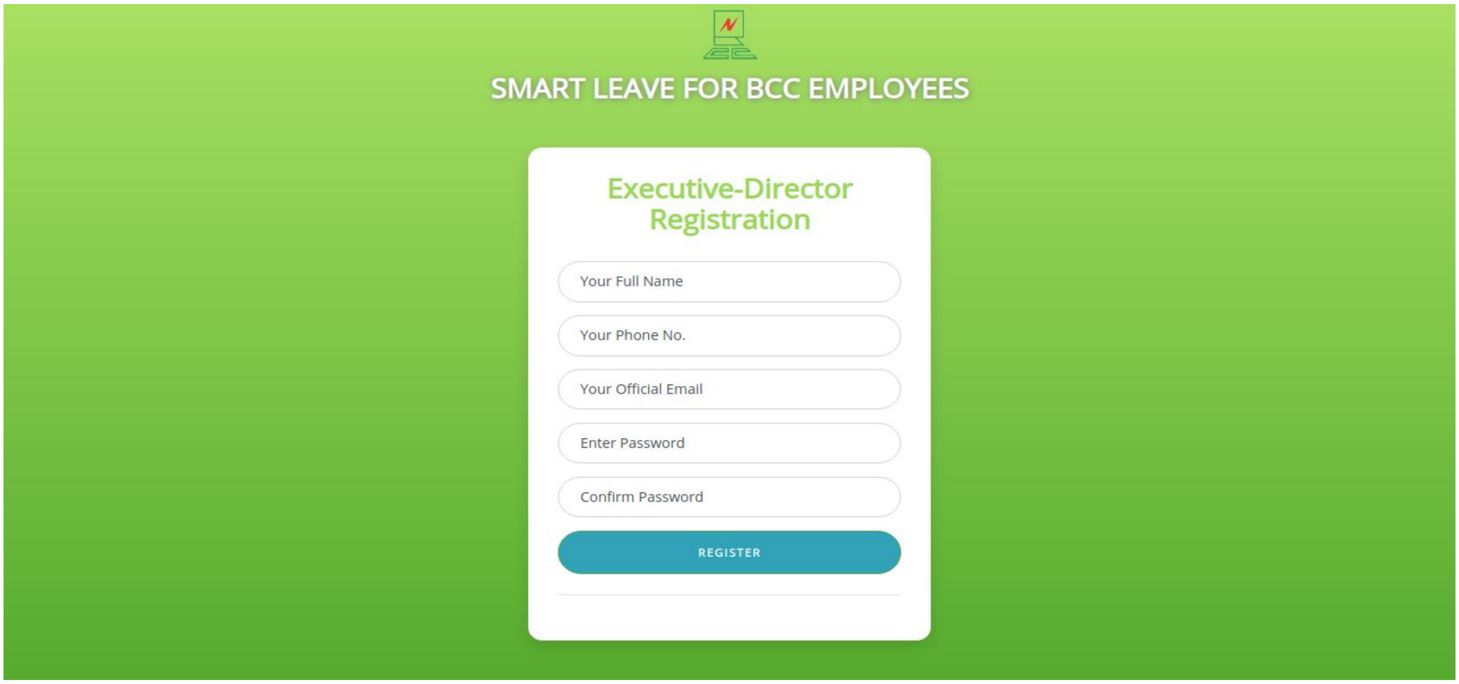
To: 11/13/2020

Total: 10

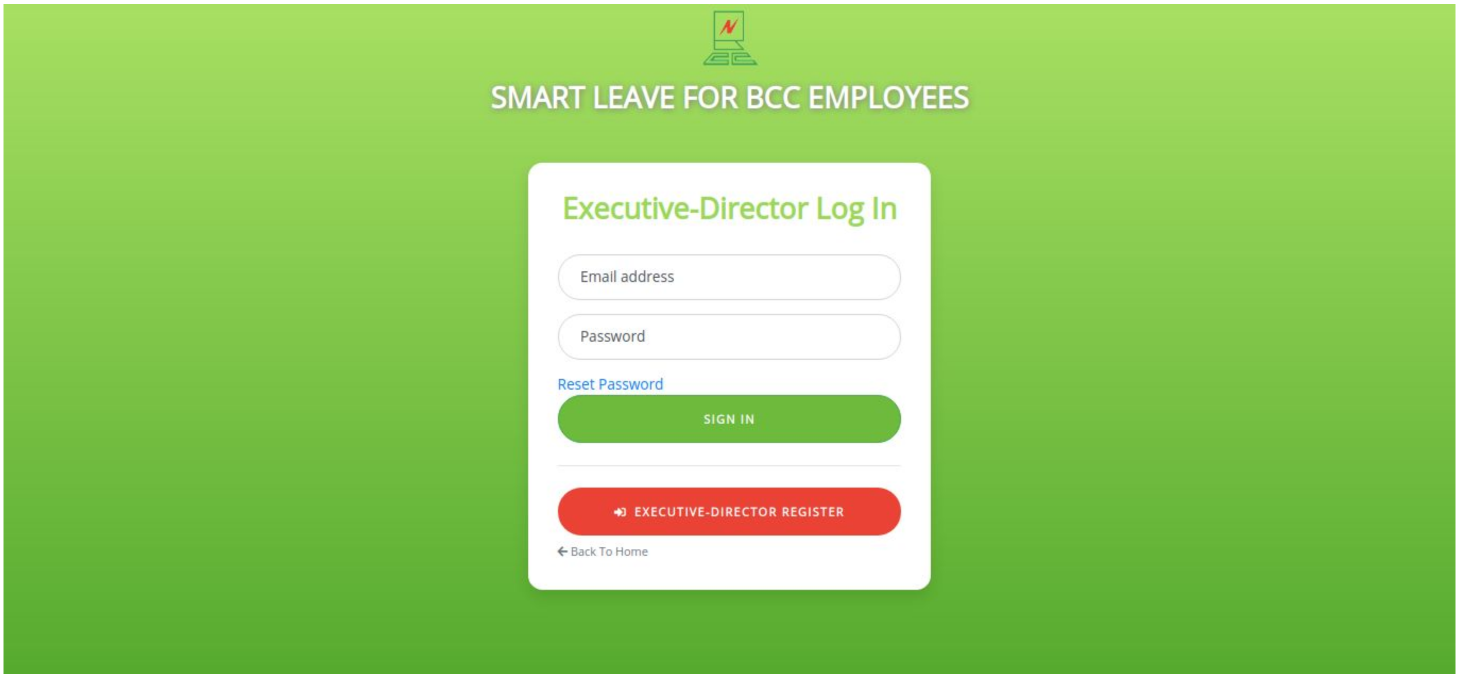
Status : Halted

*Inside An Application*

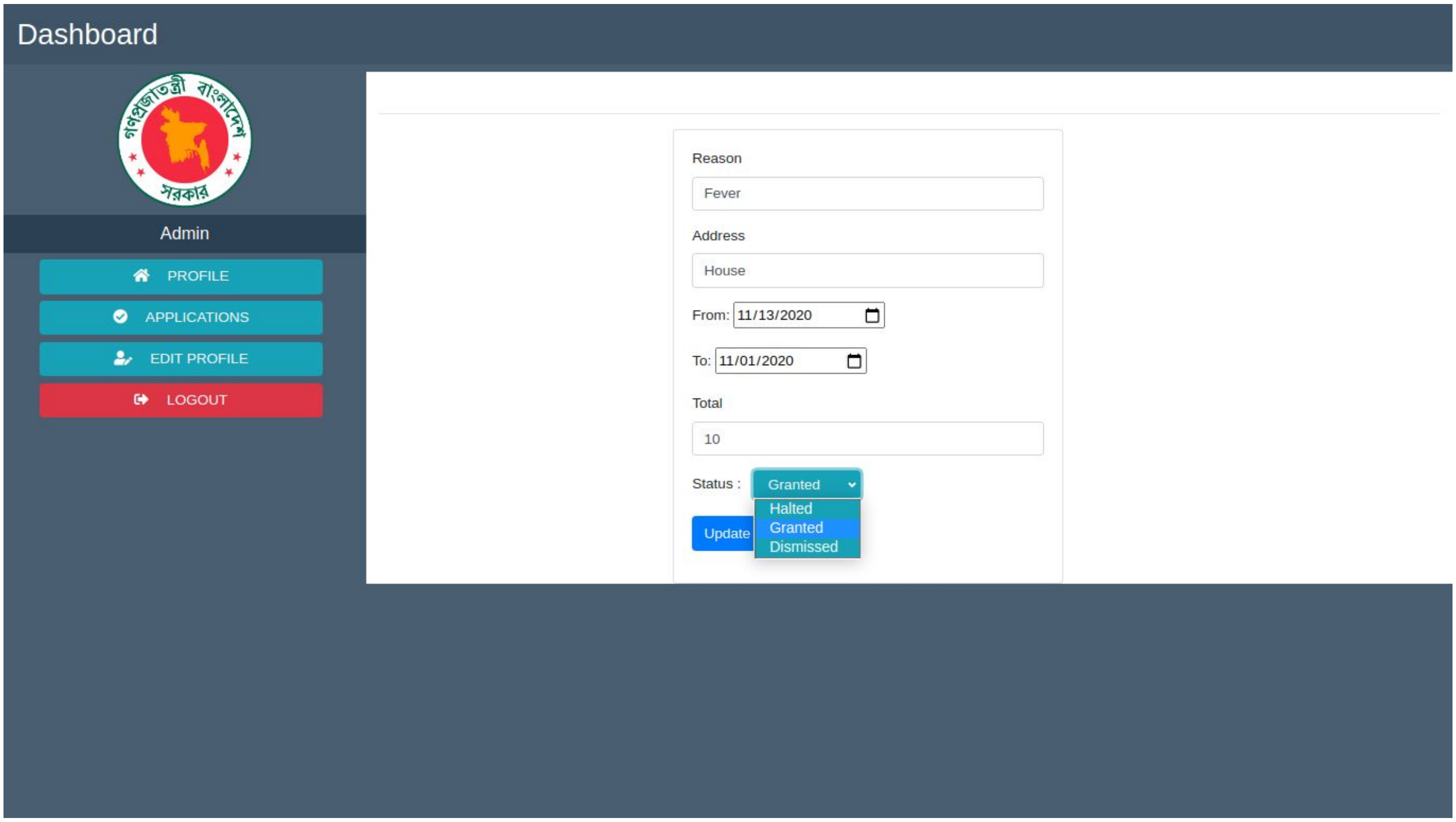
An Admin can open or delete an application from the list page. Once opened he/she can either Halt/Grant/Dismiss or Forward the application to the Director of the organization.



*Executive Director Registration*


SMART LEAVE FOR BCC EMPLOYEES
Executive-Director Log In
Email address
Password
Reset Password
SIGN IN
EXECUTIVE-DIRECTOR REGISTER
Back To Home

*Director Login Form*



Reason: Fever

Address: House

From: 11/13/2020

To: 11/01/2020

Total: 10

Status : Granted

Director can only see the applications that are forwarded to them by the Admins only. He/She then can Grant/Halt or Dismiss an Application from the status menu

## 9. Update Pipeline

In the further revision and update we can implement more functionalities like

1. Comprehensive profile
2. Adding file/supporting documents with application
3. Improving site security

## 10. Conclusion

This project was the very first of such by us. Despite being very new learners we tried to give the best by learning and implementing what we can. As a matter of fact here and there, there might be some mistakes and bugs. But overall the application works fine. We hope to improve it and make a production ready version so it can be used in a real life office management system which can save time and papers.