**Abstract**

Human Resource and Performance management website is the ultimate place for all your project needs. Registering as a client allows you to submit your project which will be developed by a host of experienced employees. The website caters to 4 types of users namely admin, manager, client and employee. As the admin accepts the project, he assigns a team of employees led by a manager to work on the project.

As the project progress, the manager updates the status of the project which is also updated in the status page of client and employee. Due to this the client is up to date with all the workings of the project .Once the project is finished, the manager can de allocate the team and deliver the project. During the construction of the project, employees and manager can rate each other depending on the performance. The average rating helps the admin choose a better team each time a project is completed.

Using this portal, the whole process is streamlined and everyone is updated on the latest proceedings.

**Table of Contents**

1. INTRODUCTION
2. LITERATURE SURVEY
3. SYSTEM REQUIREMENTS
4. SYSTEM DESIGN
5. IMPLEMENTATION
6. DATABASE IMPLEMENTATION
7. TESTING AND RESULT
8. CONCLUSION
9. BIBLIOGRAPHY

**Introduction**

Human Resource and Performance management allows the people who lack the resources to bring their project into reality by providing the necessary resources and manpower. This website which is built using HTML, CSS and PHP allows the client, admin, manager, employee all to work on the project together and keep track of it. The simplistic UI of the website helps in navigation even for the most out of touch person. At the backend It is supported by the access of database using mysql thus providing the information and details in a.

There are mainly 4 types of users this website adheres to:

Client:

This is the user who approaches the website with a project and its requirements. He/she can register if new or log in if they are a returning user. Once on the website they can specify their requirement after which they will have to wait for the project to be accepted. If accepted, the team will immediately begin working on the project, each project has 7 seven stages which are updated to the client momentarily. Once done, the project is delivered and the client can request a new project.

Admin:

He is the head of the website; He holds the responsibility for accepting or rejecting the projects. He can also add new employees and managers. Once the project has been accepted it is up to the admin to choose the leading manager and team of employers who support and work on the project. He can rate the employers and managers throughout the course of the project based on their performance. He is also updated the stage of which the project is being worked on.

Manager:

Once assigned, he gets to work on the project. It's the main duty of the manager to spear head the team to success. It's the responsibility of the manager to update the stages of the project .Once he is satisfied with the project he can deliver and de-allocate the team.

Employee- When assigned he works on the project under the watchful eyes of the manager; the status of the project can be seen by logging into the website. During the project, he can rate other colleagues and manager.

7 Stages of project:

**Initial stage:**

The Project has been accepted and the team has been assigned.

**Planning**:

Using the requirements given by the client, the project is planned out and the milestones are set.

**Designing:**

The basic Design of the project with surface level details is created.

**Developing:**

Actual development of project is done in this stage

**Testing:**

The project is extensively tested for bugs.

**Completed:**

At this stage the project is completed and delivered.

This website allows for easy development of project with proper coordination of the whole team. The client is updated at every stage of the project.

Literature Survey

**Existing System:**

Here the existing system is nothing but a manual system in which the employees has to fill their rating details in an excel sheet and send it to their supervisor then the supervisor has to merge all the employee rating details and arrange them in to a single sheet. Calculating the average rating of the employees based on their performance is a tedious process in this system.

**Drawbacks:**

* Difficulty in maintaining employee individual rating information in an excel sheet.
* More manual hours need to calculate average based on different constraints and generating
* Doesn’t provide effective communication between different levels of employees.
* Doesn’t provide security.
* Difficulty in generating the reports
* Client is not up to date with the status of the project

One way to overcome all these difficulties is so store all the information in database.  The computerization helps mitigate a lot of drawback and streamlines the process.

**Proposed System:**

Performance management system is to replace the existing manual system with a software solution. It allows all the employees in different sections of the company to rate their colleagues and managers. Depending on the average rating which is automatically calculated the admin can choose the best team for the job. The client is now up to date on the status of the project.

* Faster processing when compared to existing one
* Centralized database helps in avoiding conflicts
* Easy to use GUI that does not requires specific training
* Provides facility for the employees and managers to rate their fellow peers.
* Auto average calculation helps the admin asses the performance.
* The client is part of the project in every step and the project status is updated momentarily.

**System Requirements**

**Hardware Requirements:**

* Processor – Pentium IV or above
* Ram – 2 GB or more

**Software Requirements:**

* **Server Side:**

1. XAMPP
2. MySql
3. FTP accessibility to upload the content to the server or modify them.
4. Server running on windows.

* **Client Side:**

1. Browser supporting HTML 5 and CSS 3
2. Machine running windows.
3. Internet connection.

**SYSTEM DESIGN**

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

**UML DIAGRAMS**

The Unified modeling language (UML) is a general purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

**USE CASE DIAGRAM**

**Use case for admin**



Admin is the most important person in this system. Admin is a super user of this performance management System. Admin must have a login and complete authentication process.

Now, the admin has complete access into the system. He can view the all Manager, Employee, Projects and clients.

Admin can view client requirement and decide whether to accept/reject based on requirement and availability resources.

**Use case for Manager**



 Manager is second most member of this system. He must do log in with the authentication process. He can view the rating of employee. He can rate performance of the employee. Manager can also update the projects status.

**Use case for Employee**



The employee is working member of this system. He must do log in with the authentication process. He can view rating of co-worker. He has the ability to rate his manager and co-workers.

**Use case for Client**



**ER-DIAGRAM:**

The below figure shows **entity-relationship model** describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types and specifies relation that can exist between instances of those entity types.

**SCHEMA DIAGRAM**

The **database schema** of a [database system](https://en.wikipedia.org/wiki/Database_system) is its structure described in a [formal language](https://en.wikipedia.org/wiki/Formal_language) supported by the [database management system](https://en.wikipedia.org/wiki/Database_management_system) (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of [relational databases](https://en.wikipedia.org/wiki/Relational_databases)). The formal definition of a [database](https://en.wikipedia.org/wiki/Database) schema is a set of formulas (sentences) called [integrity constraints](https://en.wikipedia.org/wiki/Integrity_constraints) imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the [database language](https://en.wikipedia.org/wiki/Database_language). The states of a created [conceptual schema](https://en.wikipedia.org/wiki/Conceptual_schema) are transformed into an [explicit mapping](https://en.wikipedia.org/wiki/Explicit_and_implicit_methods), the database schema. This describes how real-world entities are [modeled](https://en.wikipedia.org/wiki/Data_modeling) in the database.

**Implementation**

Requirement analysis for web applications encompasses three major tasks: formulation, requirements gathering and analysis modeling. During formulation, the basic motivation and goals for the web application are identified, and the categories of users are defined. In the requirements gathering phase, the content and functional requirements are listed and interaction scenarios written from end-user’s point-of-view are developed. This intent is to establish a basic understanding of why the web application is built, who will use it, and what problems it will solve for its users.

**TOOLS and Software used**

**XAMP Application**

XAMMP server is a utility designed to allow creating web application and managing server and databases. XAMMP server is a windows web development environment. It allows creating web applications with Apache2, PHP and a MYSQL database. It also comes with PHP MYADMIN and SQLITE Manager to easily manage your database XAMMP server installs automatically (installer), and its usage is very easy. We can tune server without even touching the setting files. XAMMP server is the only packaged solution that will allow reproducing in production server. Once XAMMP server is installed, have the possibility to add as many Apache, MYSQL, and PHP releases as want. XAMMP server also has a tray icon to manage your server and its settings.

**HTML**

**HTML** or **Hyper Text Markup Language** is the standard markup language used to create web pages. HTML was created in 1991 by Tim Berners-Lee at CERN in Switzerland. It was designed to allow scientists to display and share their research.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <imp>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured

Documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as Java Script which affect the behavior of HTML web pages.

HTML is descriptive markup language. Library of various markup languages is defined in various browsers.

**CSS**

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

With plain HTML you define the colors and sizes of text and tables throughout your pages. If

You want to change a certain element you will therefore have to work your way through the

Document and change it. With CSS you define the colors and sizes in "styles". Then as you

Write your documents you refer to the styles. Therefore: if you change a certain style it will

Change the look of your entire site. Another big advantage is that CSS offers much more detailed attributes than plain HTML for defining the look and feel of your site.

**PHP**

PHP is now officially known as “**PHP: Hypertext Preprocessor**”. It is a server-side scripting language usually written in an HTML context. Unlike an ordinary HTML page, a PHP script is not sent directly to a client by the server; instead, it is parsed by the PHP binary or module, which is server-side installed. HTML elements in the script are left alone, but PHP code is interpreted and executed. PHP code in a script can query databases, create images, read and write files, talk to remote servers – the possibilities is endless. The output from PHP code is combined with the HTML in the script and the result sent to the user’s web-browser, therefore it can never tell the user whether the web-server uses PHP or not, because the entire browser sees is HTML.

PHP’s support for Apache and MySQL further increases its popularity. Apache is now the most-used web-server in the world, and PHP can be compiled as an Apache module. MySQL is a powerful free SQL database, and PHP provides a comprehensive set of functions for working with it. The combination of Apache, MySQL and PHP is all but unbeatable.

That doesn’t mean that PHP cannot work in other environments or with other tools. In fact, PHP supports an extensive list of databases and web-servers. While in the mid-1990s it was ok to build sites, even relatively large sites, with hundreds of individual hard-coded HTML pages, today’s webmasters are making the most of the power of databases to manage their content more effectively and to personalize their sites according to individual user preferences.

**Database Implementation**

**Table structure for table `admin`**

CREATE TABLE `admin` (

`username` varchar(10) DEFAULT NULL,

`Password` varchar(10) DEFAULT NULL

);

**Table structure for table `client`**

CREATE TABLE `client` (

`c\_name` varchar(20) DEFAULT NULL,

`username` varchar(40) DEFAULT NULL,

`password` varchar(20) DEFAULT NULL,

`phone` decimal(10,0) DEFAULT NULL,

`address` varchar(35) DEFAULT NULL,

`requirement` text,

`client\_id` int (11) NOT NULL

);

**Table structure for table `employee`**

CREATE TABLE `employee` (

`Emp\_id` int (11) NOT NULL,

`name` varchar(25) DEFAULT NULL,

`address` varchar(50) DEFAULT NULL,

`phone` decimal(10,0) DEFAULT NULL,

`gender` varchar(6) DEFAULT NULL,

`role` varchar(20) DEFAULT NULL,

`Skill` varchar(20) DEFAULT NULL,

`salary` int(11) DEFAULT NULL,

`username` varchar(35) DEFAULT NULL,

`password` varchar(15) DEFAULT NULL,

`rating` float DEFAULT NULL

)

**Table structure for table `emp\_state`**

CREATE TABLE `emp\_state` (

`Emp\_id` int(11) DEFAULT NULL,

`state` int(11) DEFAULT NULL

);

**Table structure for table `pre\_project`**

CREATE TABLE `pre\_project` (

`c\_id` int(11) DEFAULT NULL,

`p\_id` int(11) DEFAULT NULL,

`pre\_req` text,

`p\_state` int(11) DEFAULT NULL

);

**Table structure for table `project`**

CREATE TABLE `project` (

`p\_no` int(11) DEFAULT NULL,

`p\_name` varchar(30) DEFAULT NULL,

`mgr\_id` int(11) DEFAULT NULL,

`start\_date` date DEFAULT NULL,

`fianl\_date` date DEFAULT NULL,

`client\_id` int(11) DEFAULT NULL

);

**Table structure for table `project\_state`**

CREATE TABLE `project\_state` (

`p\_status` int(11) DEFAULT NULL,

`client\_id` int(11) DEFAULT NULL,

`pno` int(11) NOT NULL

);

**Table structure for table `works\_on`**

CREATE TABLE `works\_on` (

`Empid` int(11) DEFAULT NULL,

`pno` int(11) DEFAULT NULL

);

**Insertion queries**

**inserting data for table `admin`**

INSERT INTO `admin` (`username`, `password`) VALUES

('admin', 'admin');

**inserting data for table `client`**

INSERT INTO `client` (`c\_name`, `username`, `password`, `phone`, `address`, `requirement`, `client\_id`) VALUES

('shivakumar', 'shivakumarnarayana00@gmail.com', 'aa', '8548987355', 'bangalore', 'library', 100);

**inserting data for table `employee`**

INSERT INTO `employee` (`Emp\_id`, `name`, `address`, `phone`, `gender`, `role`, `Skill`, `salary`, `username`, `password`, `rateing`) VALUES

(101, 'shivakumar', 'banglore', '8548987355', 'male', 'Employee', 'web developer', 20000, 'shivakumarnarayana00@gmail.com', 'Nanditha', 4.96875);

**Dumping data for table `emp\_state`**

INSERT INTO `emp\_state` (`Emp\_id`, `state`) VALUES

(101, 0);

**inserting data for table `pre\_project`**

INSERT INTO `pre\_project` (`c\_id`, `p\_id`, `pre\_req`, `p\_state`) VALUES

(100, 20, 'hrmanagement', 7);

**Inserting data for table `project`**

INSERT INTO `project` (`p\_no`, `p\_name`, `mgr\_id`, `start\_date`, `final\_date`, `client\_id`) VALUES(20, 'hello', 105, '2018-11-01', '2018-11-28', 100);

**Inserting data for table `project\_state`**

INSERT INTO `project\_state` (`p\_status`, `client\_id`, `pno`) VALUES

(8, 100, 20);

**Trigger**

CREATE TRIGGER `pstate` AFTER INSERT ON `client` FOR EACH ROW INSERT into project\_state values(0,new.client\_id,null);

**Stored procedure**

CREATE DEFINER=`root`@`localhost` PROCEDURE `Getall` () BEGIN

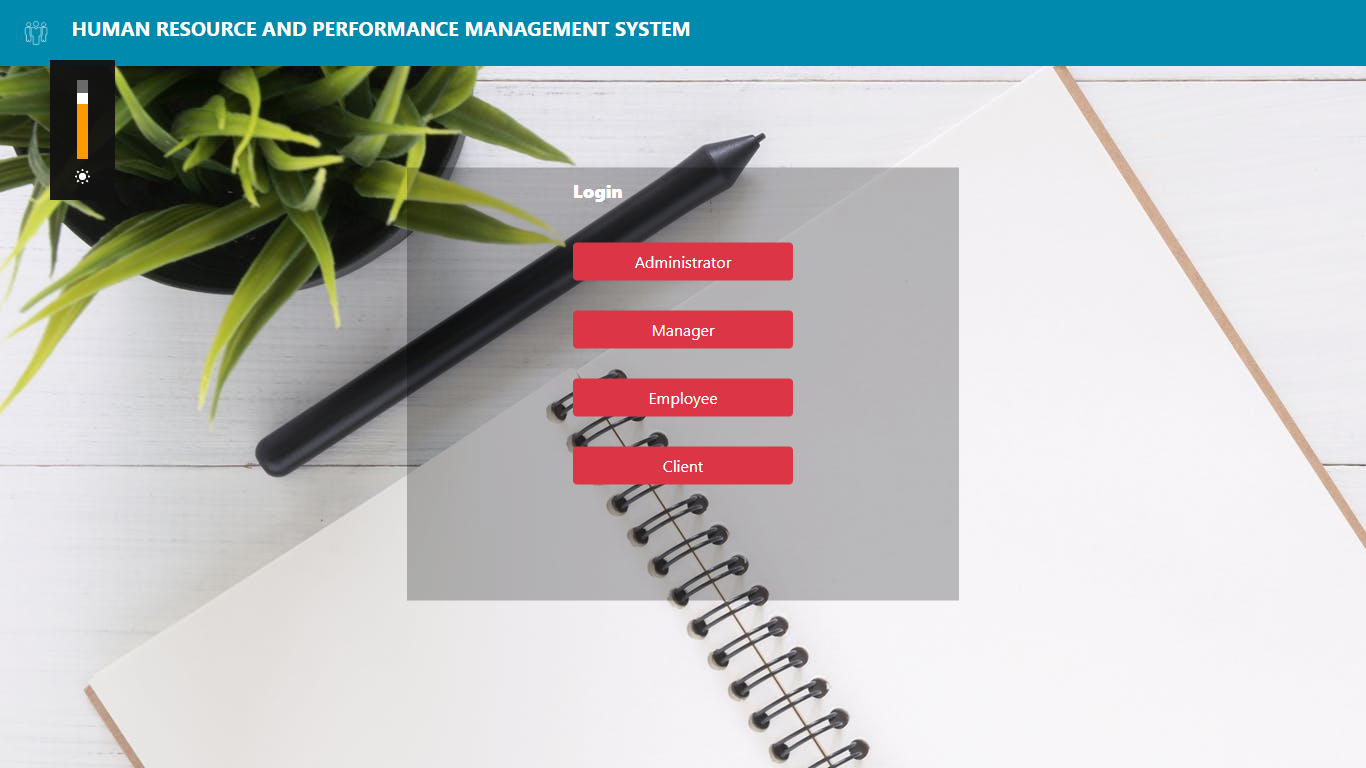
SELECT \*

FROM employee;

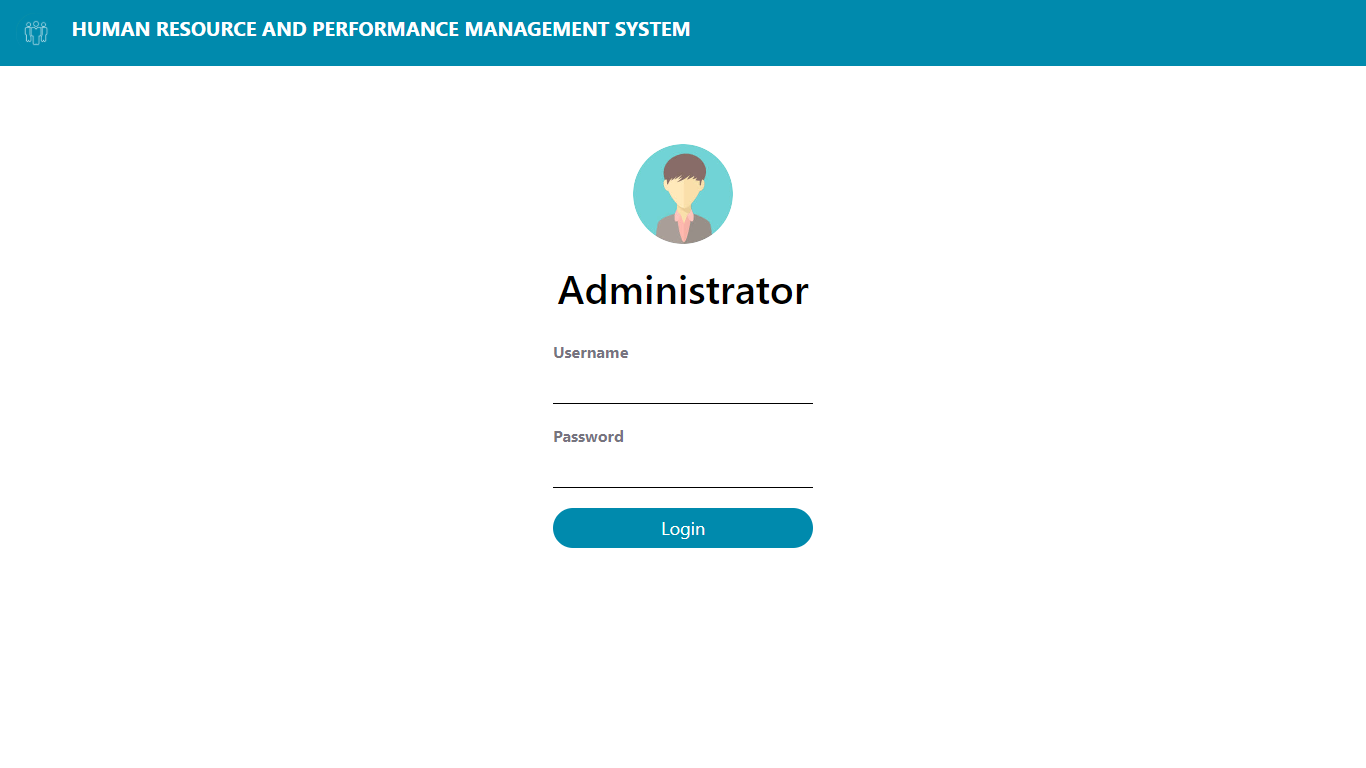
END

**Testing and Result**

Connecting to the website, we are first greeted with the User page. Here we can choose which user you want to login as.



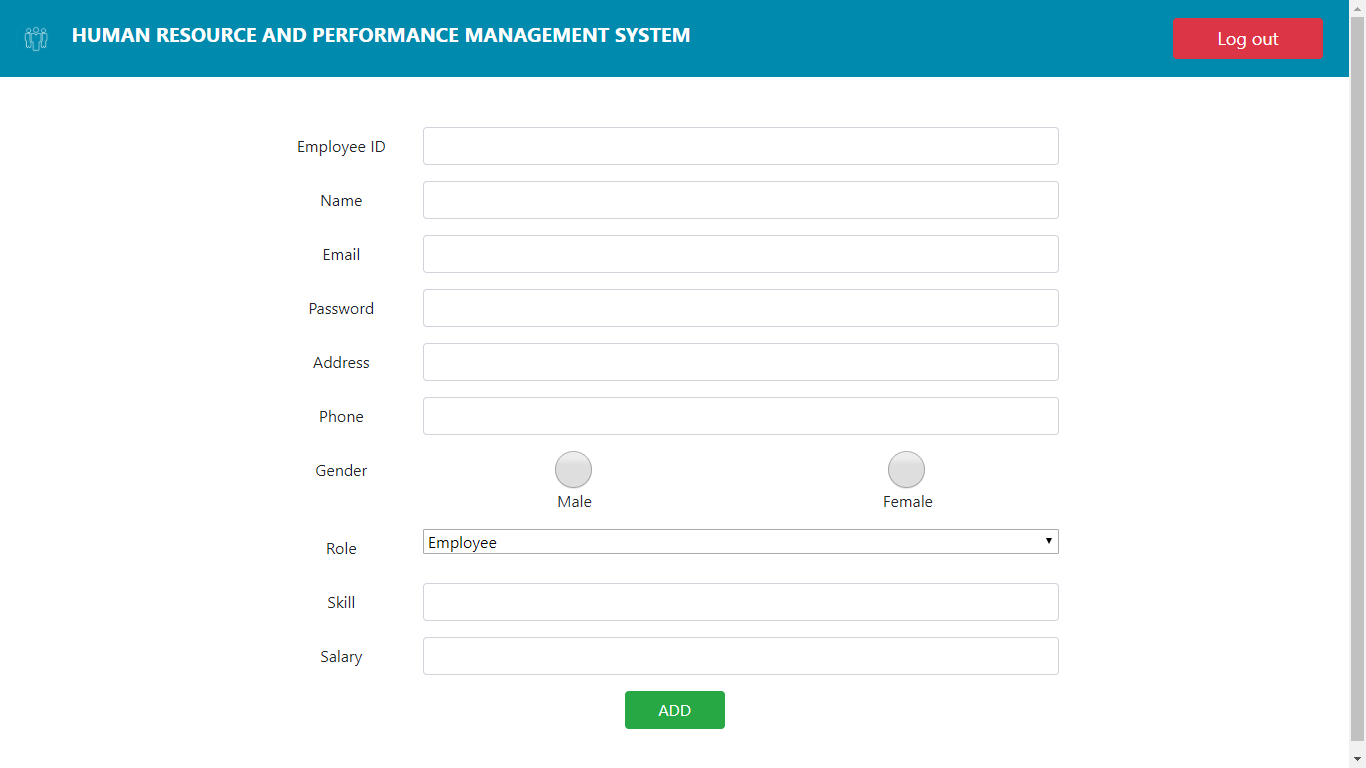
Selecting Administrator, we are taken to the login page for the admin

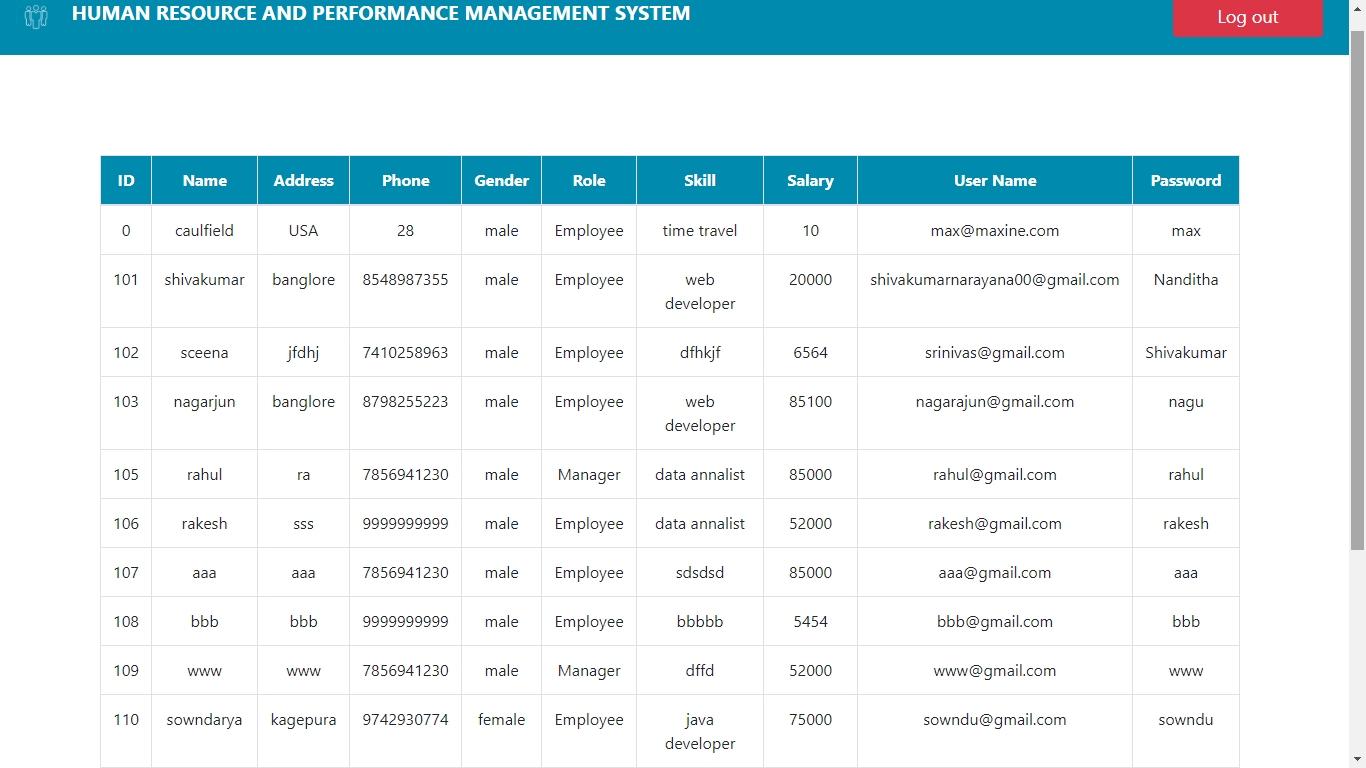


On successfully logging in, we get plethora of information and options to choose from.

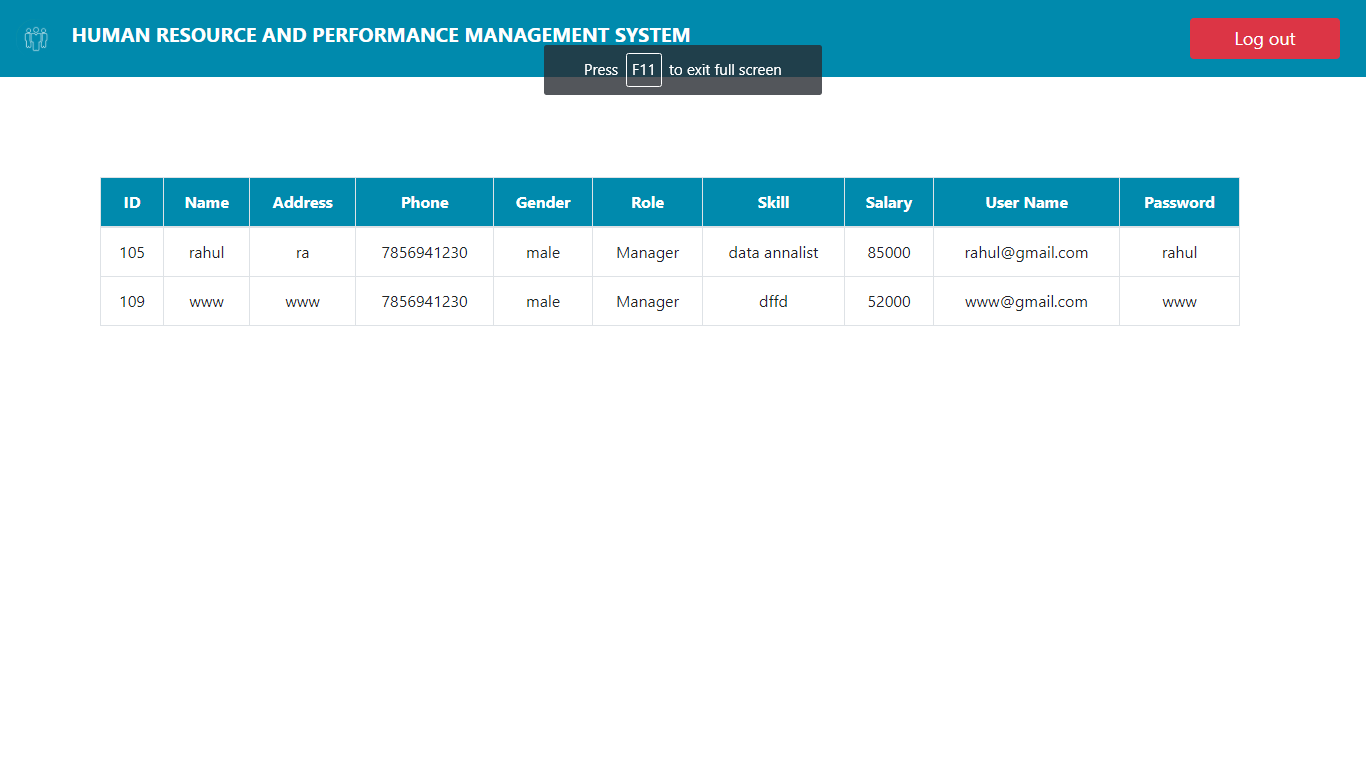


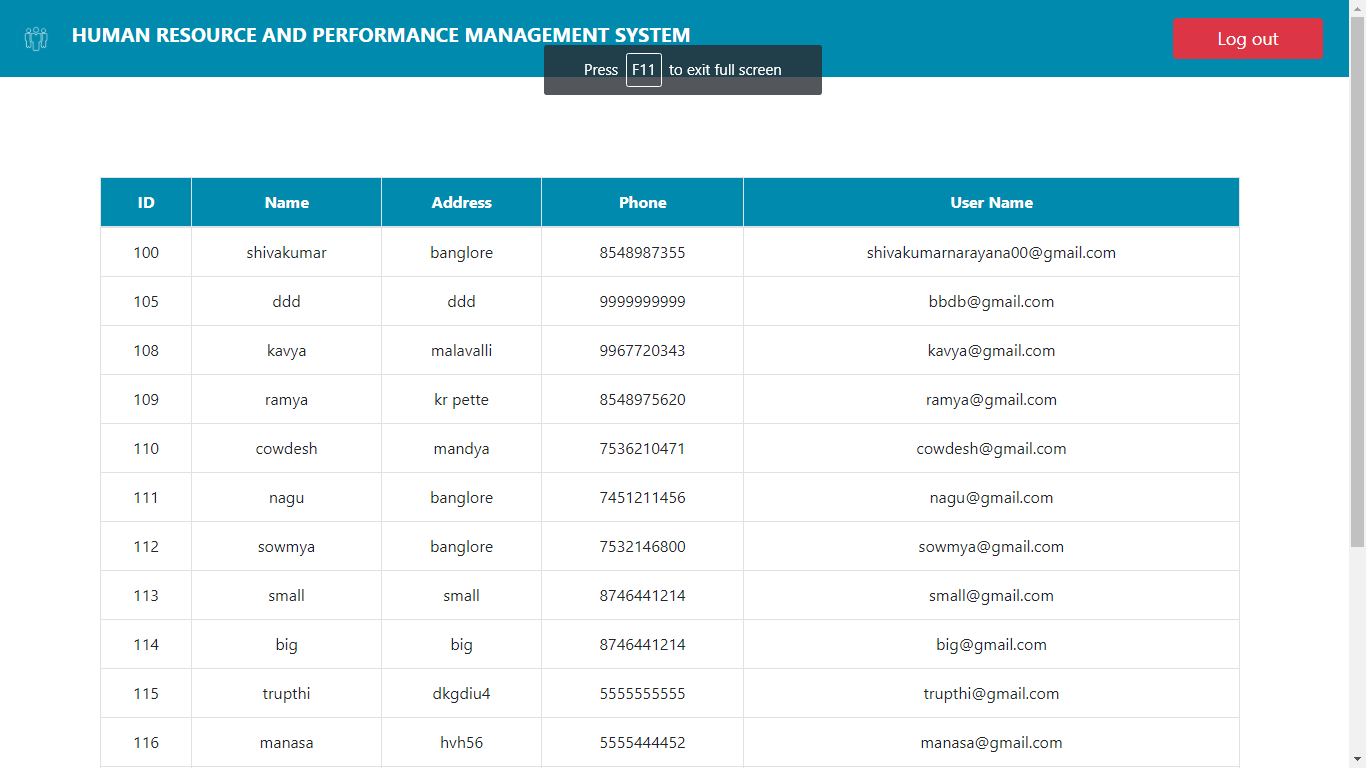
To add a new Employee we click on the button + Add Employee, on clicking we are taken to a page where we can fill out the details for the new employee

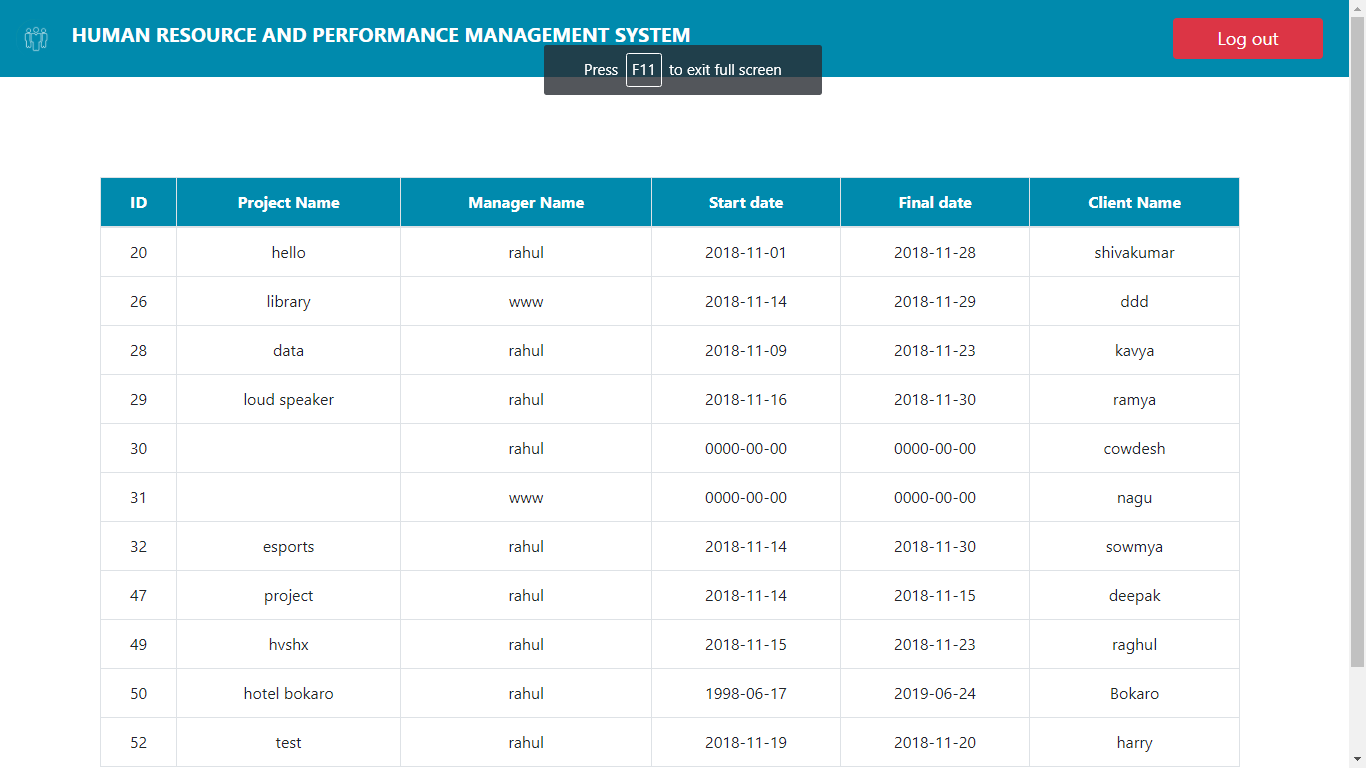


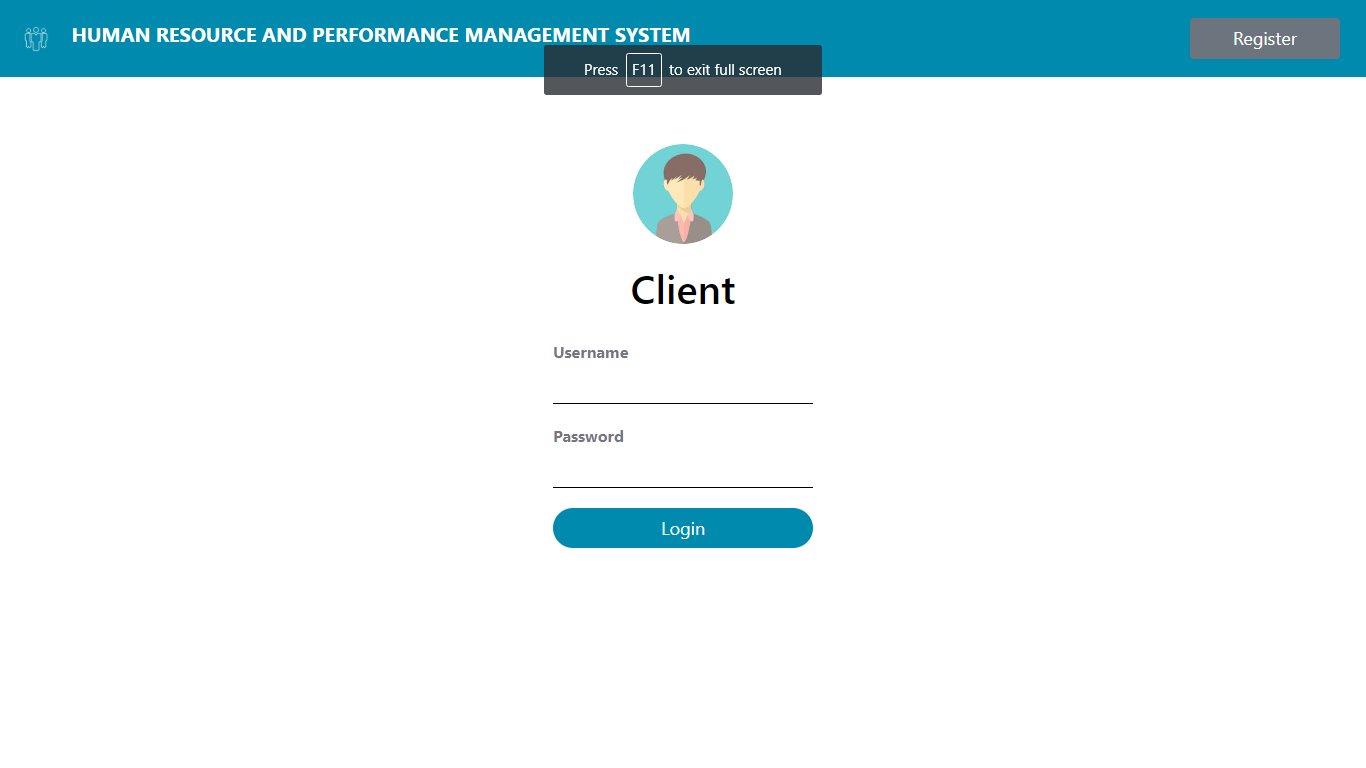
If we want to view all the present Employees, we choose “Employees” Button.

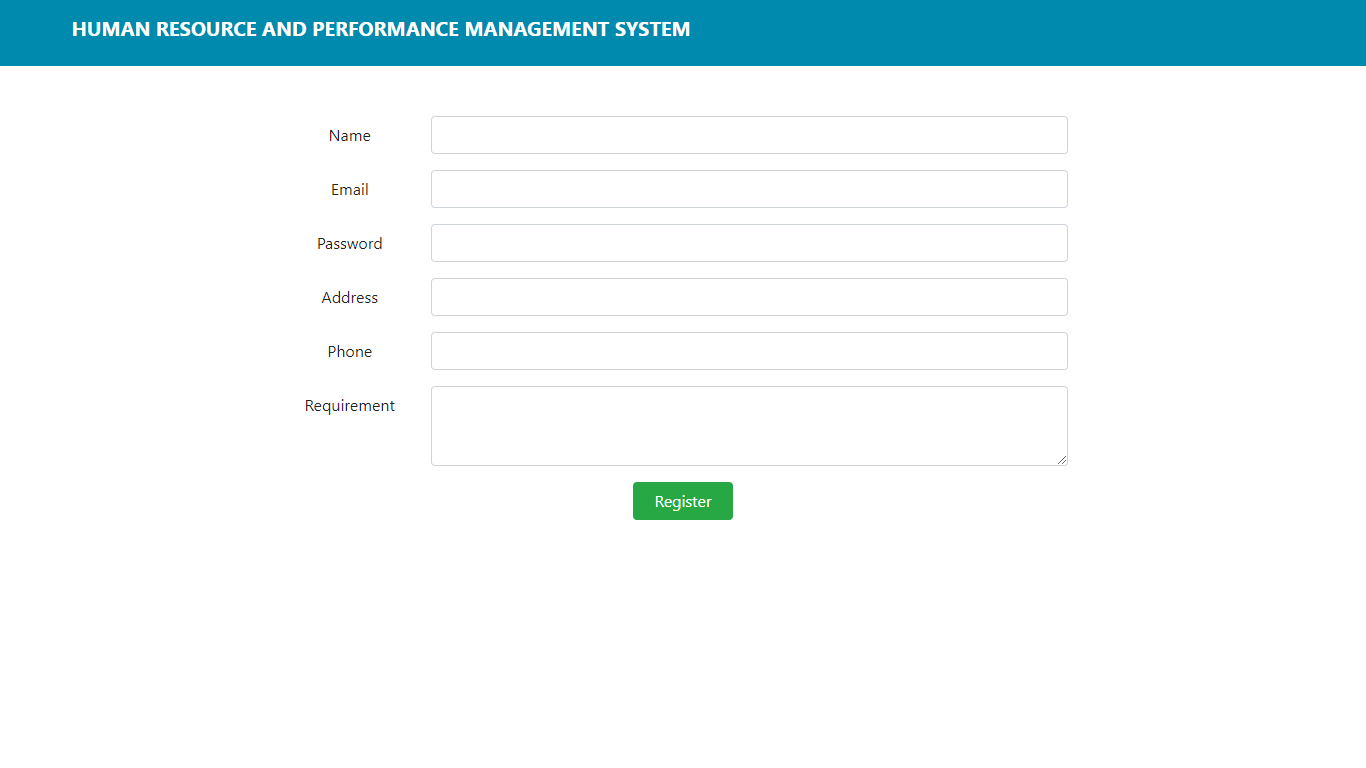
In order to view the managers we choose the option “Regional Managers”



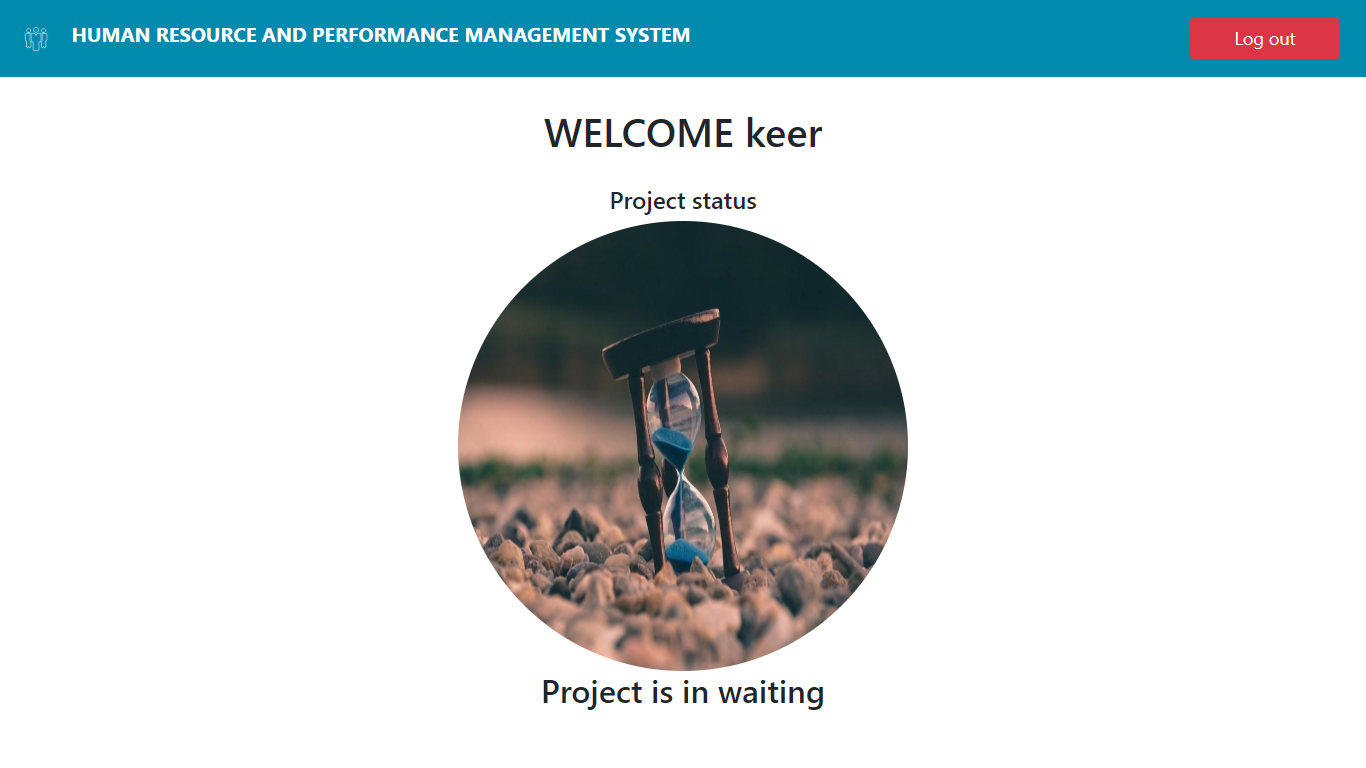
To see all current and previous clients we choose “Clients” Button

To view the projects we choose the button “Total Projects”

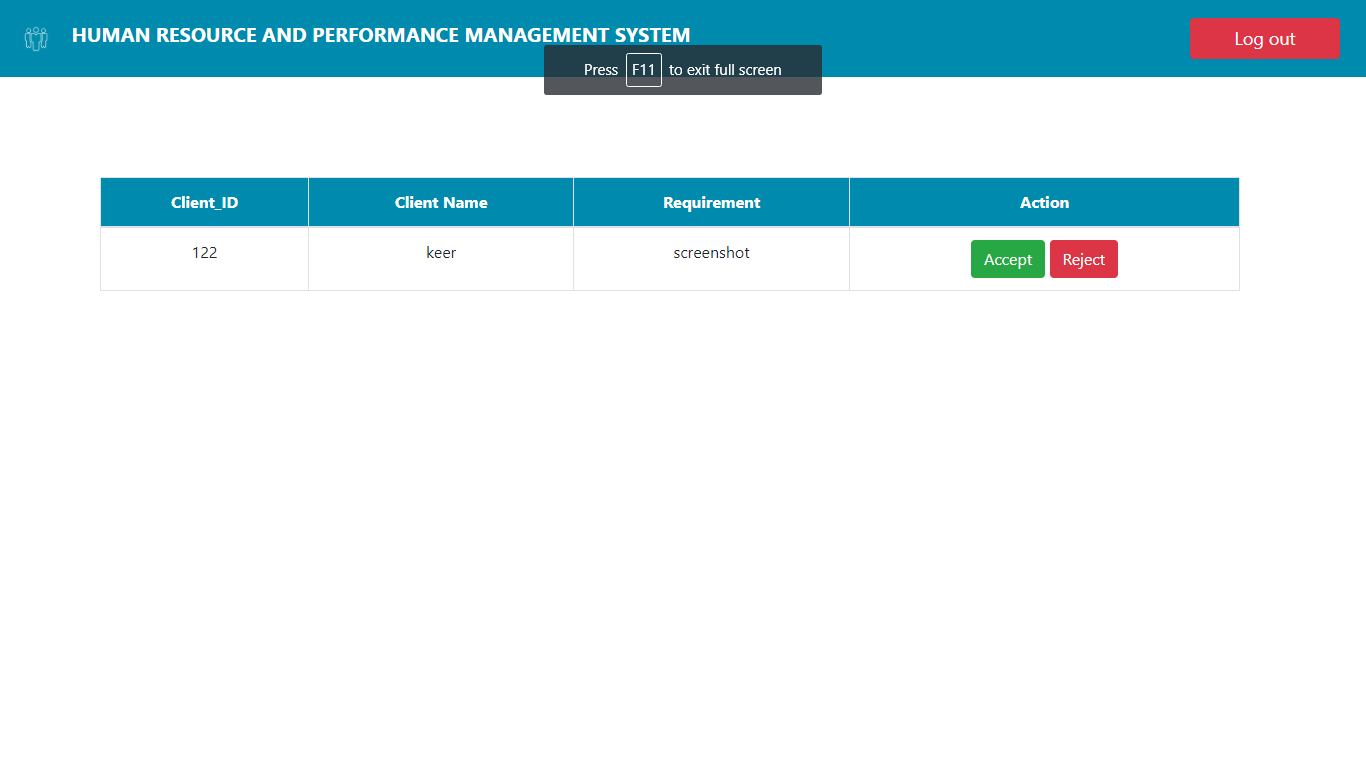
If we choose Client in the user page we are greeted with client login page,

If it’s a new client, he/she can register using the register button

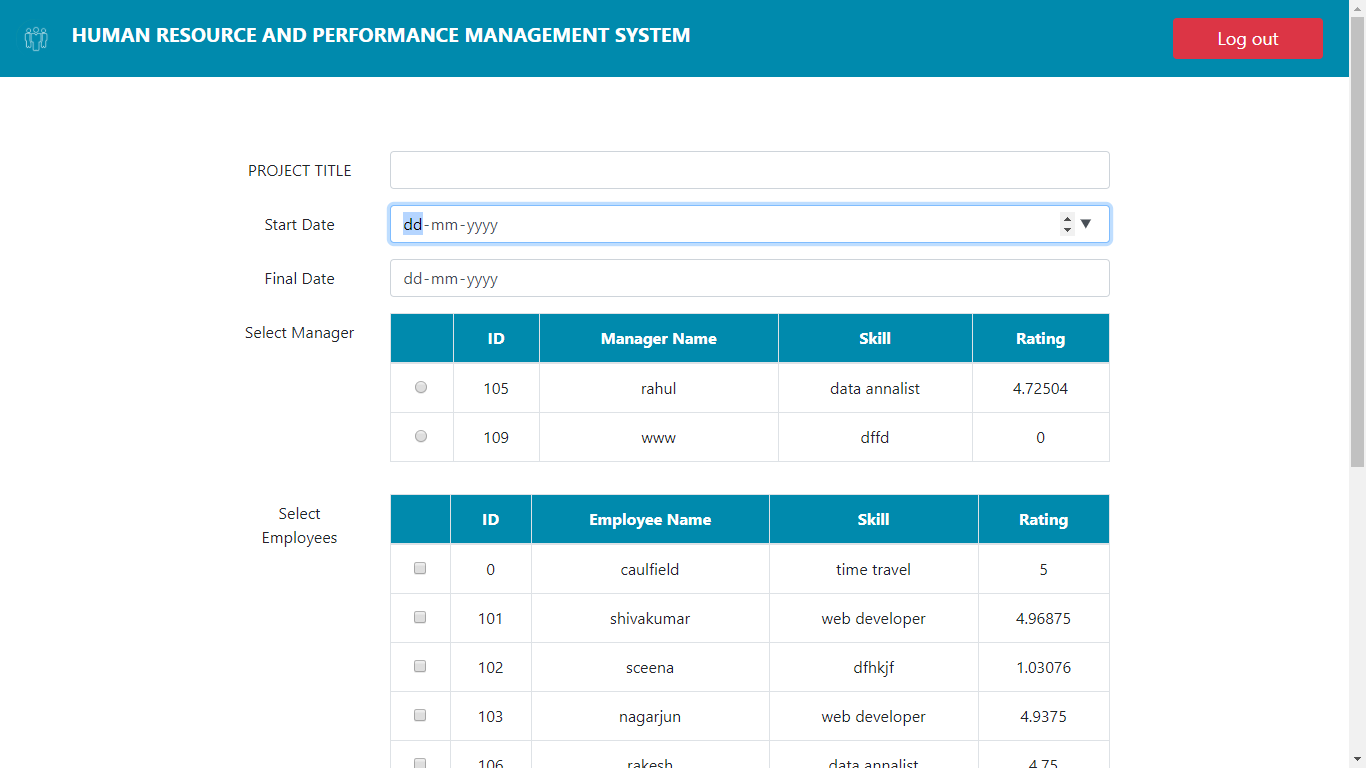
Once successfully logged in or registered, the client is taken to a page where he can see his status of his project



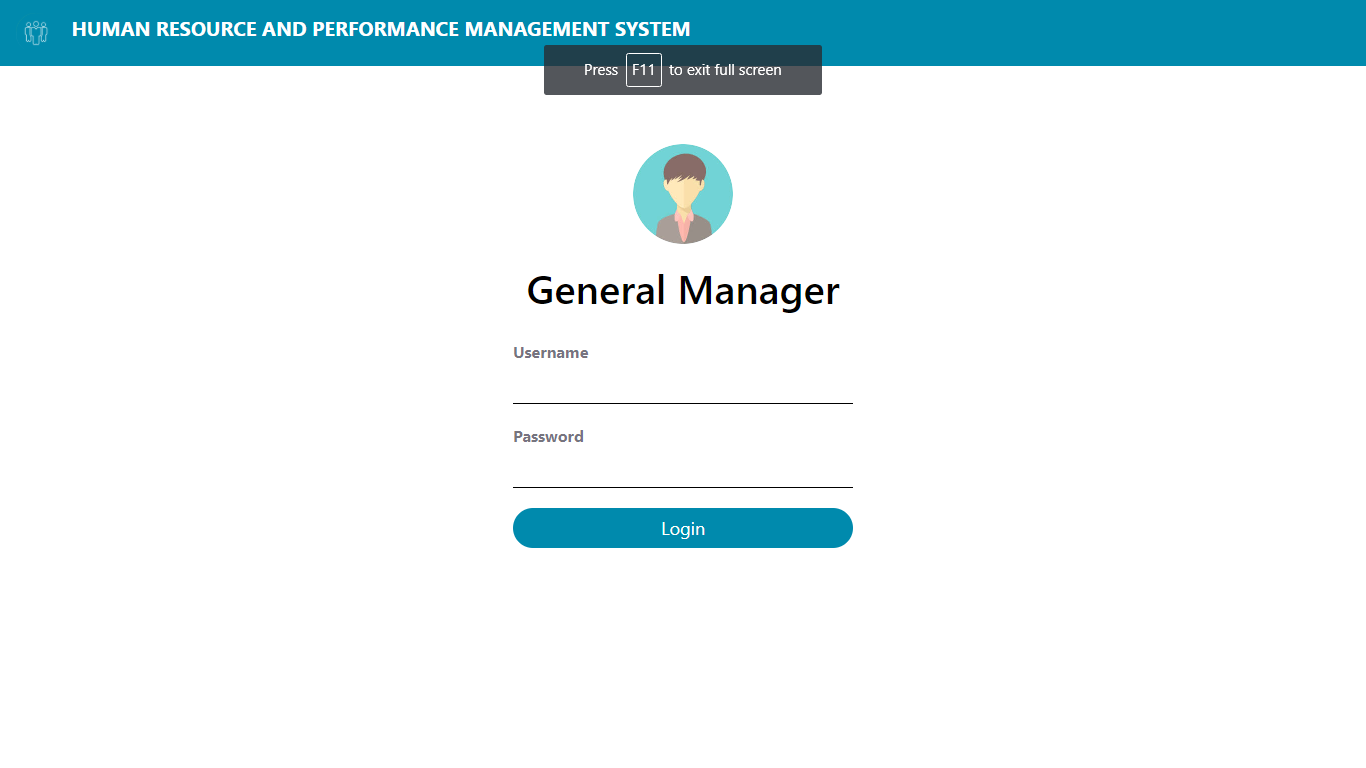
Before the project gets started it must first be approved by the admin, to accept choose “Accept”, to reject choose “Reject”

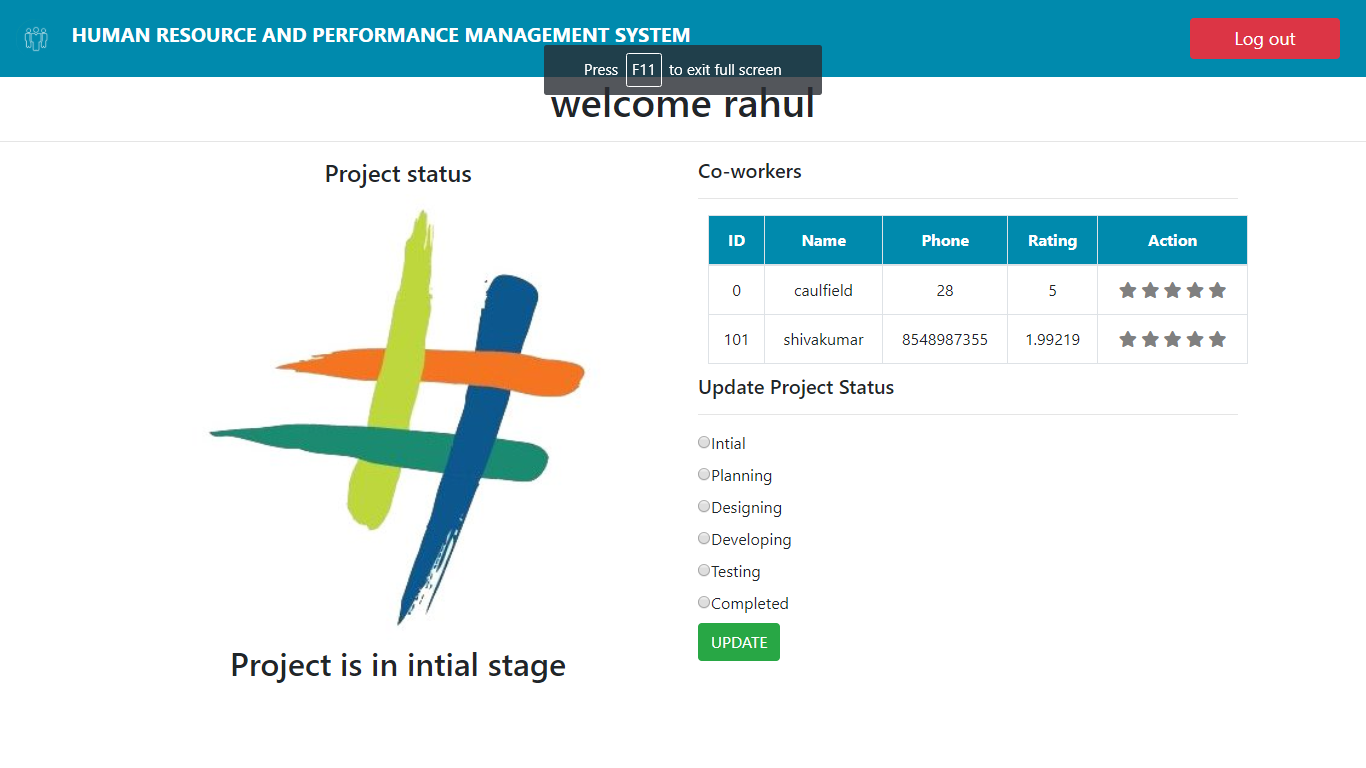


If rejected then the client can add a new requirement, but once accepted the admin can now assign the start and end date, employees and, managers.

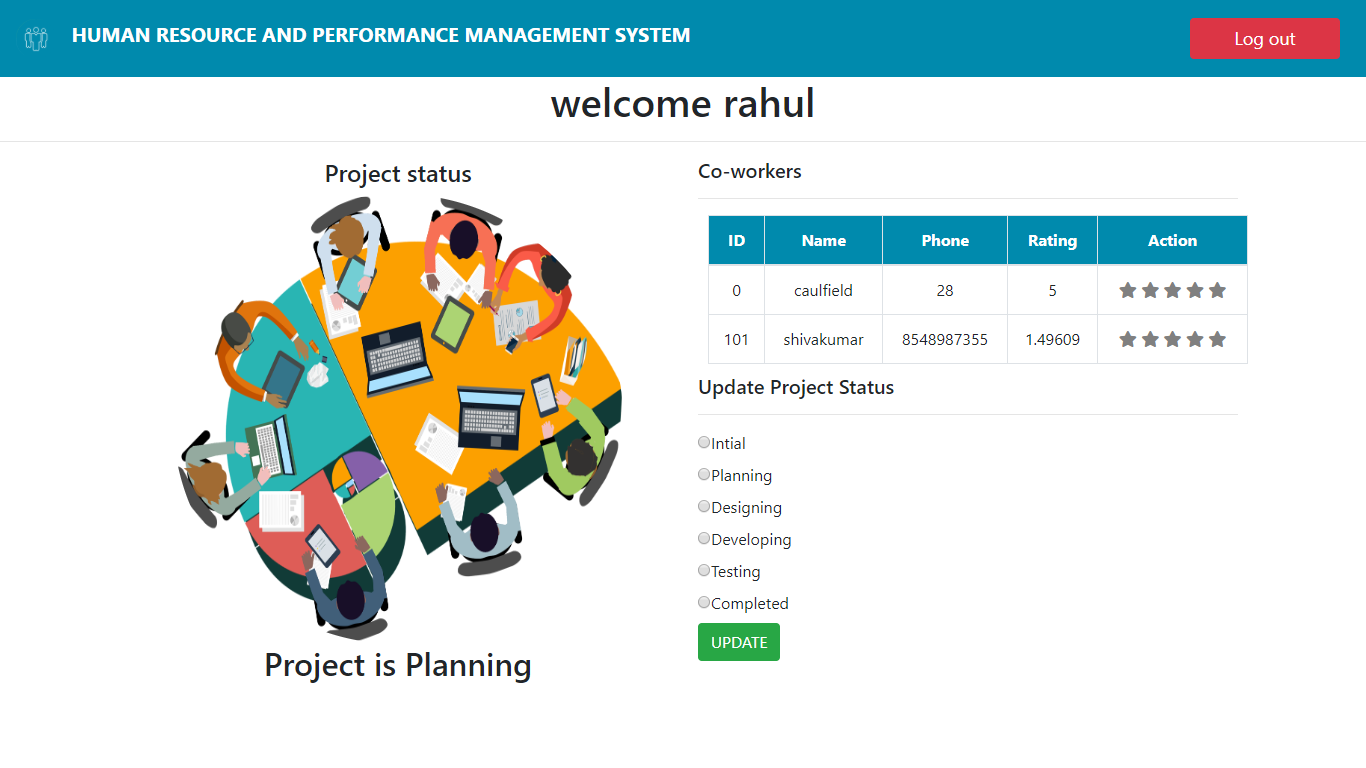


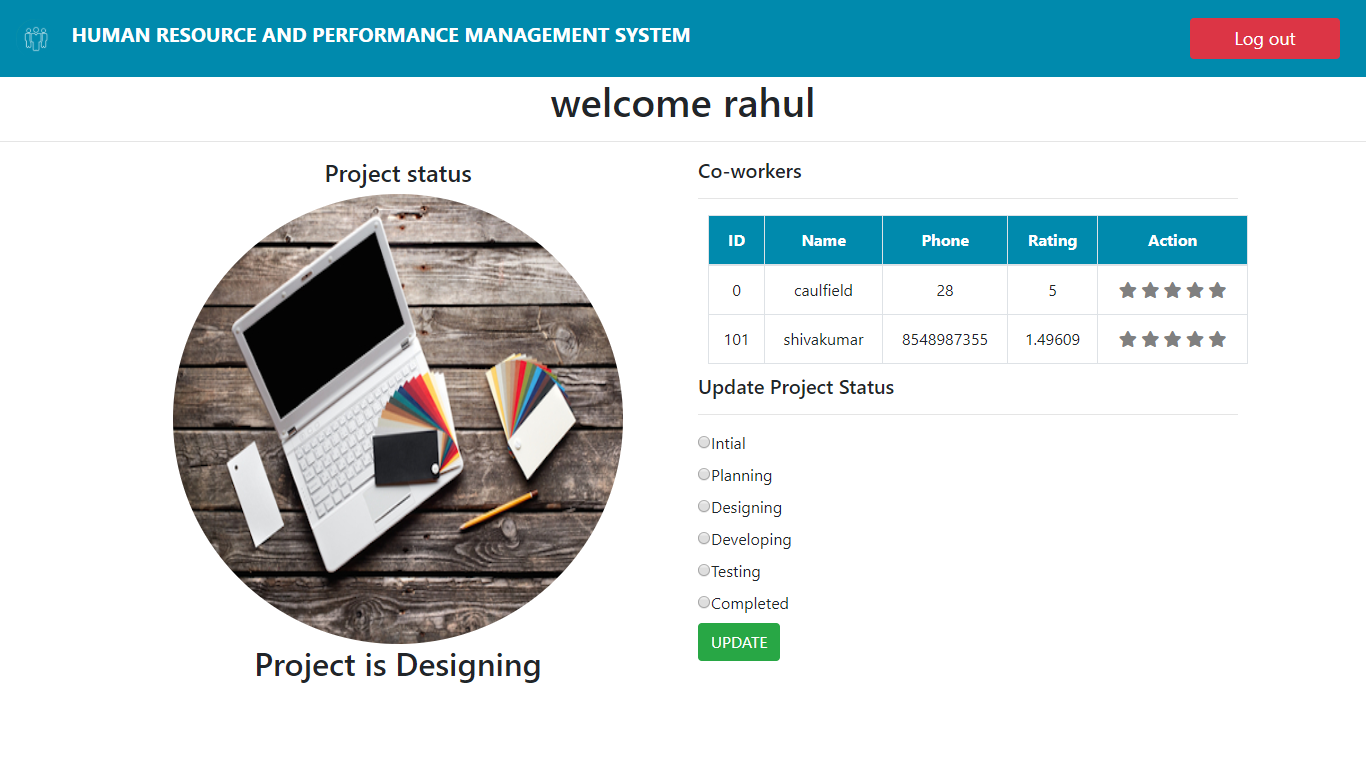
In the user page, if we choose manager we are greeted with a manager login page

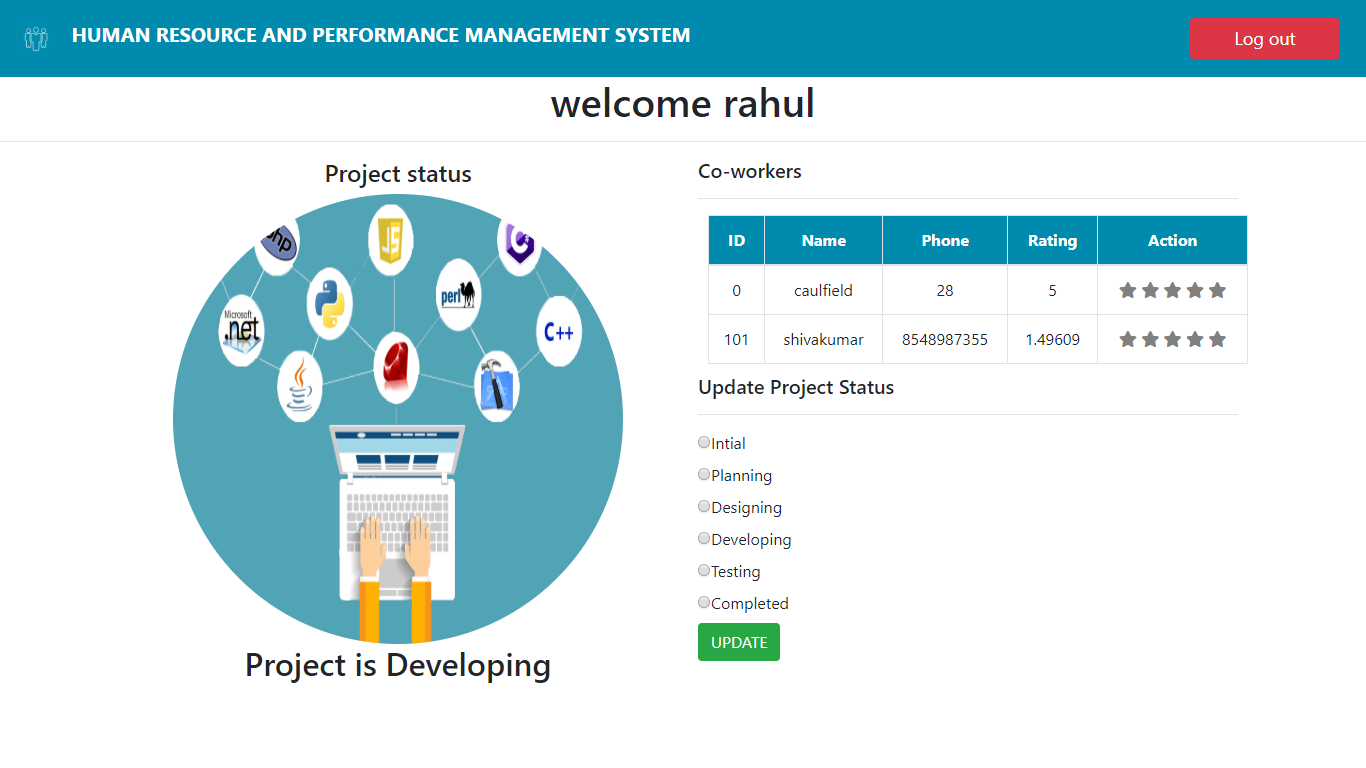


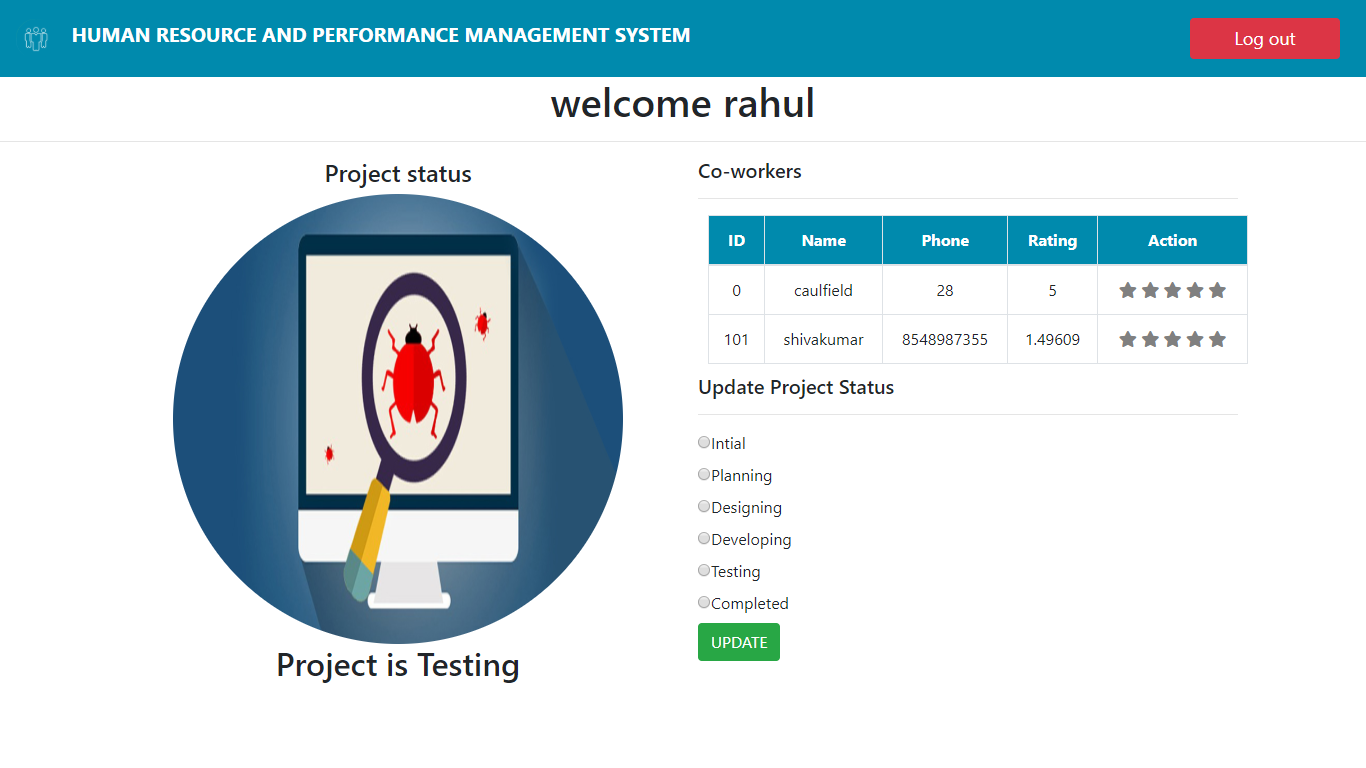
On successful login, the assigned manager can rate his fellow employees and also update the status of the project. There are mainly 6 stages that the manager can choose based on the progress of the project.

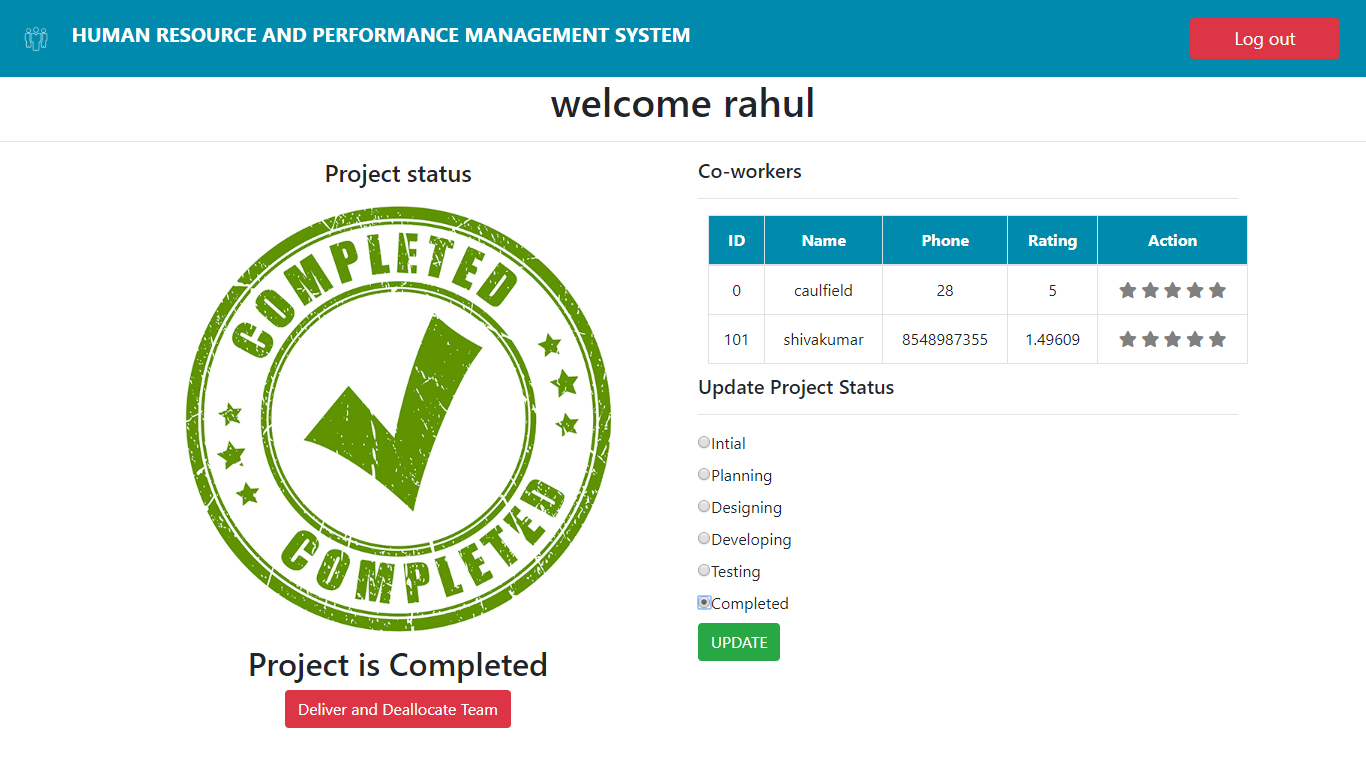
Choosing each stage and updating











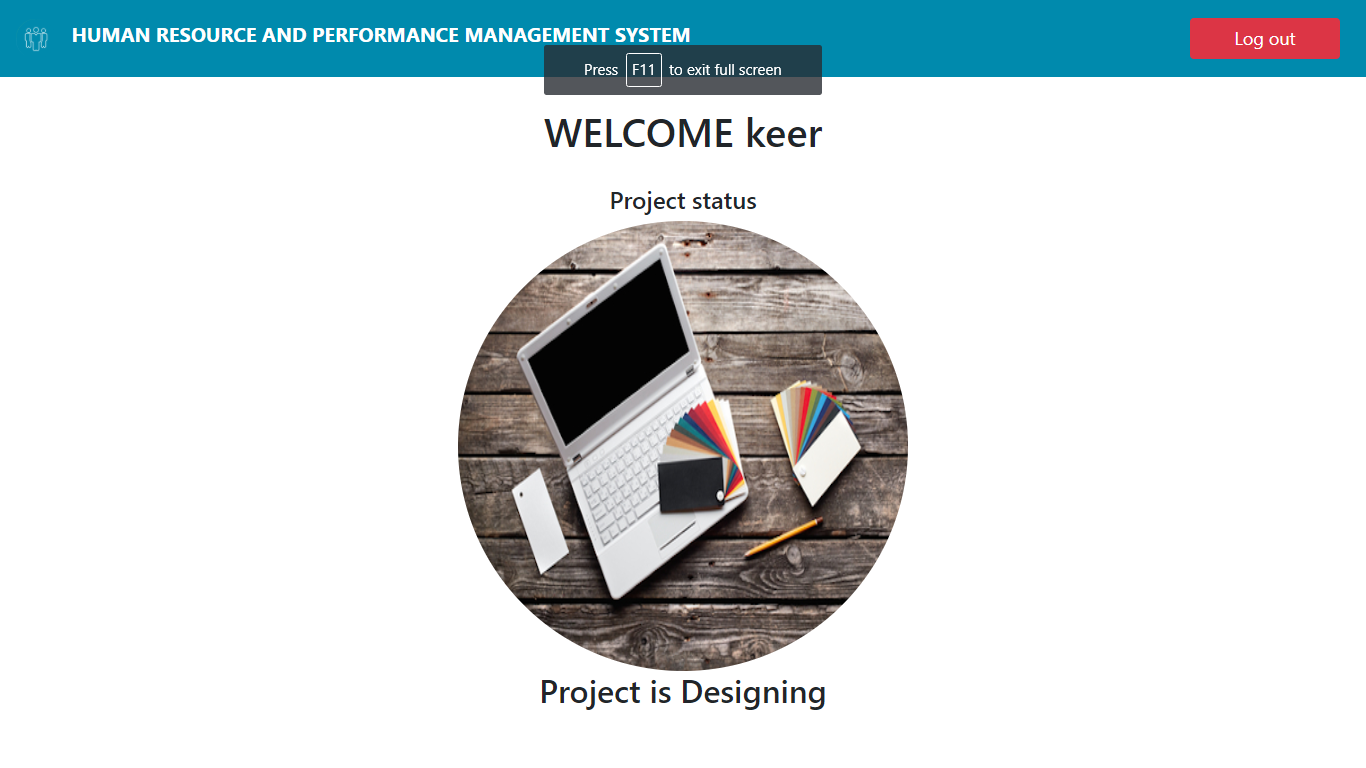
Once completed, the manager can deliver the project and deallocate the team.

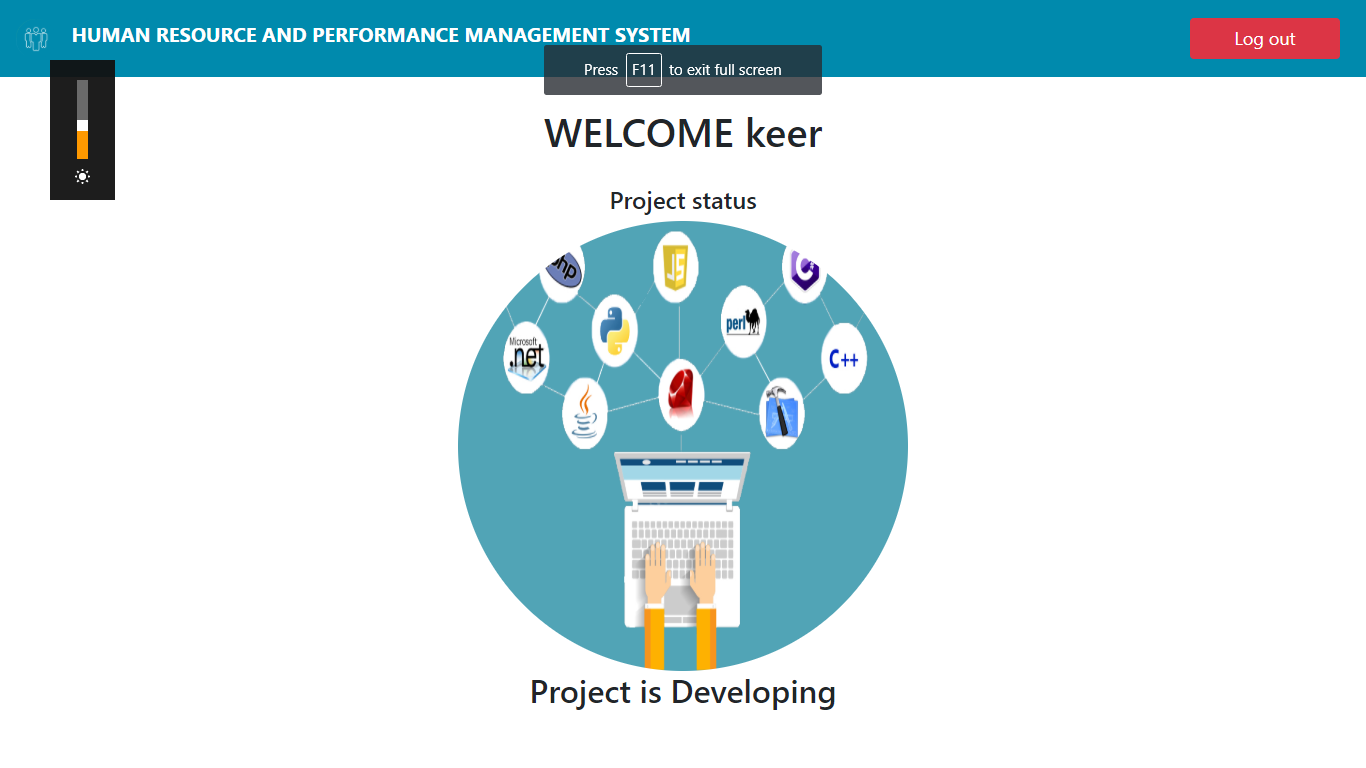
In the client side, if the project is updated then the status page displays:

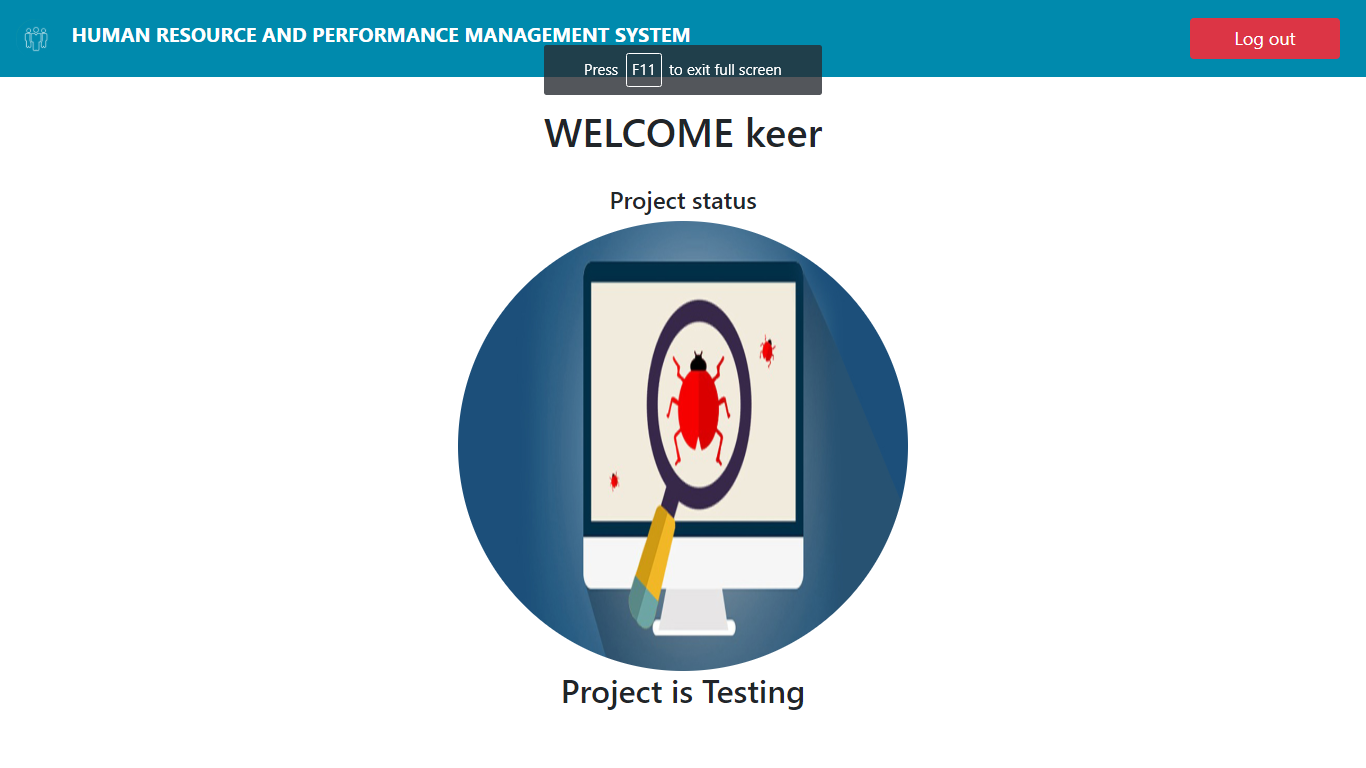


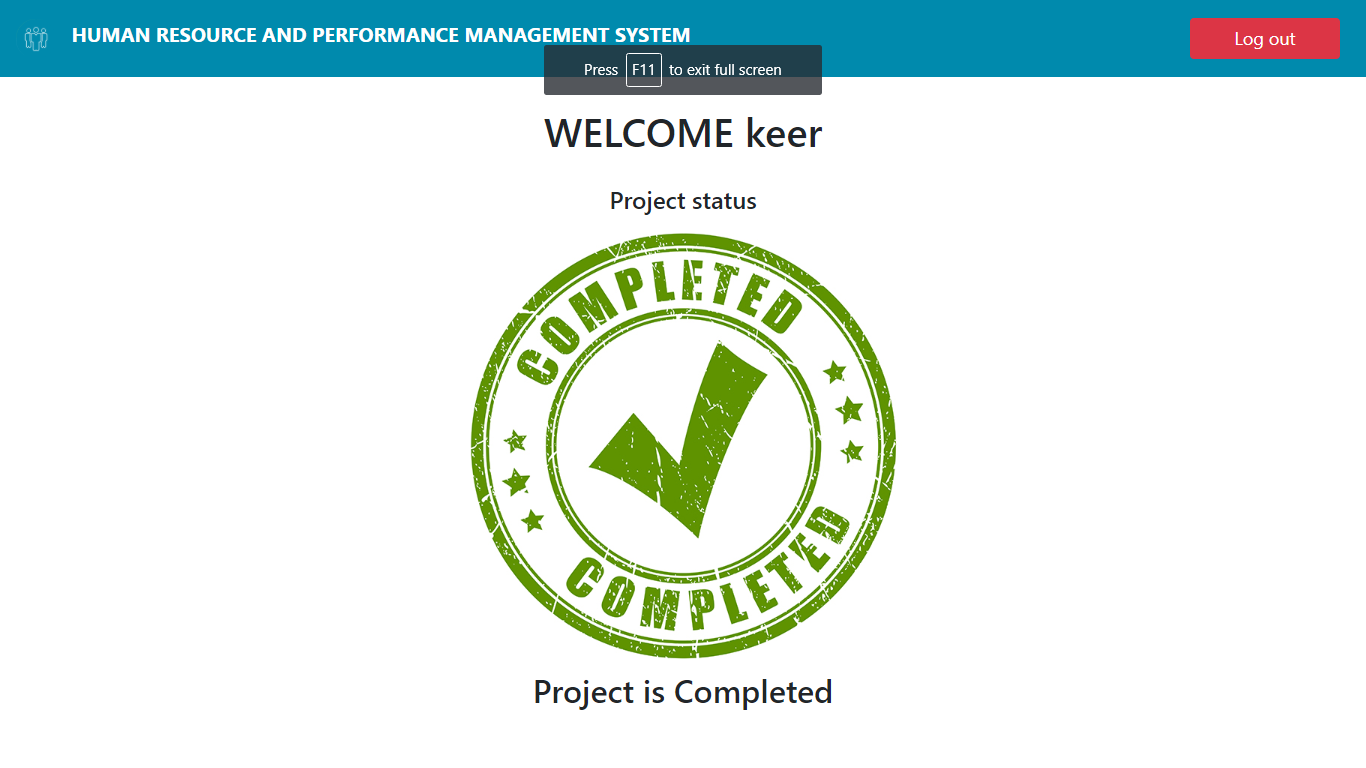
As the manager updates the progress of the project, this page shows the status to the client

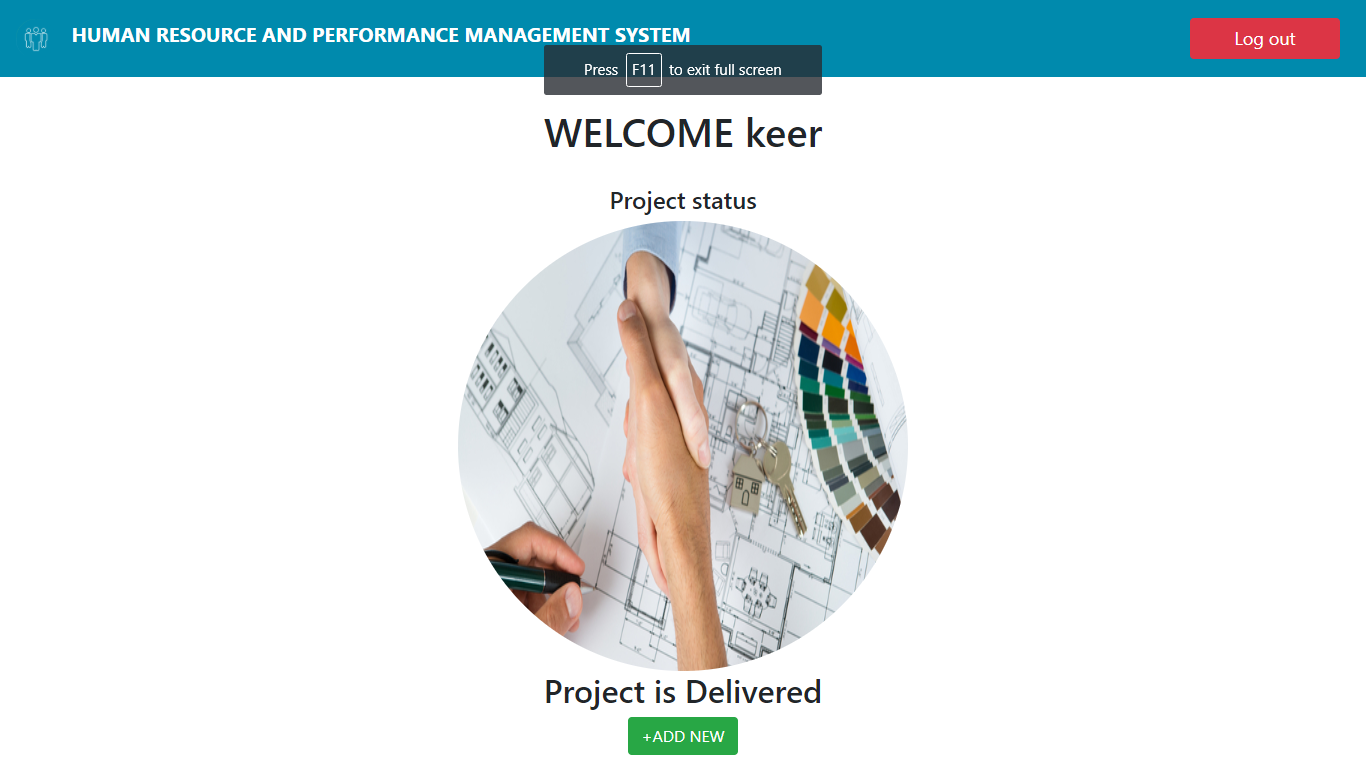




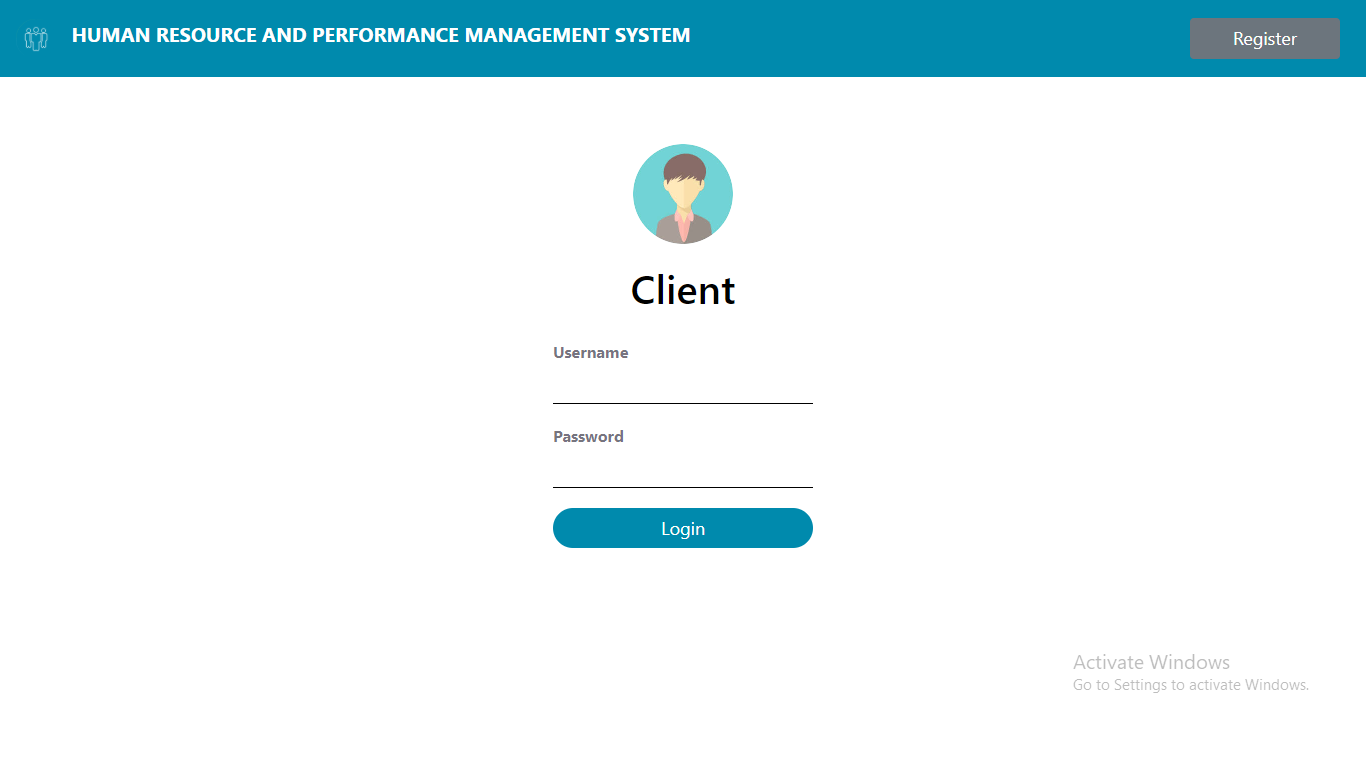


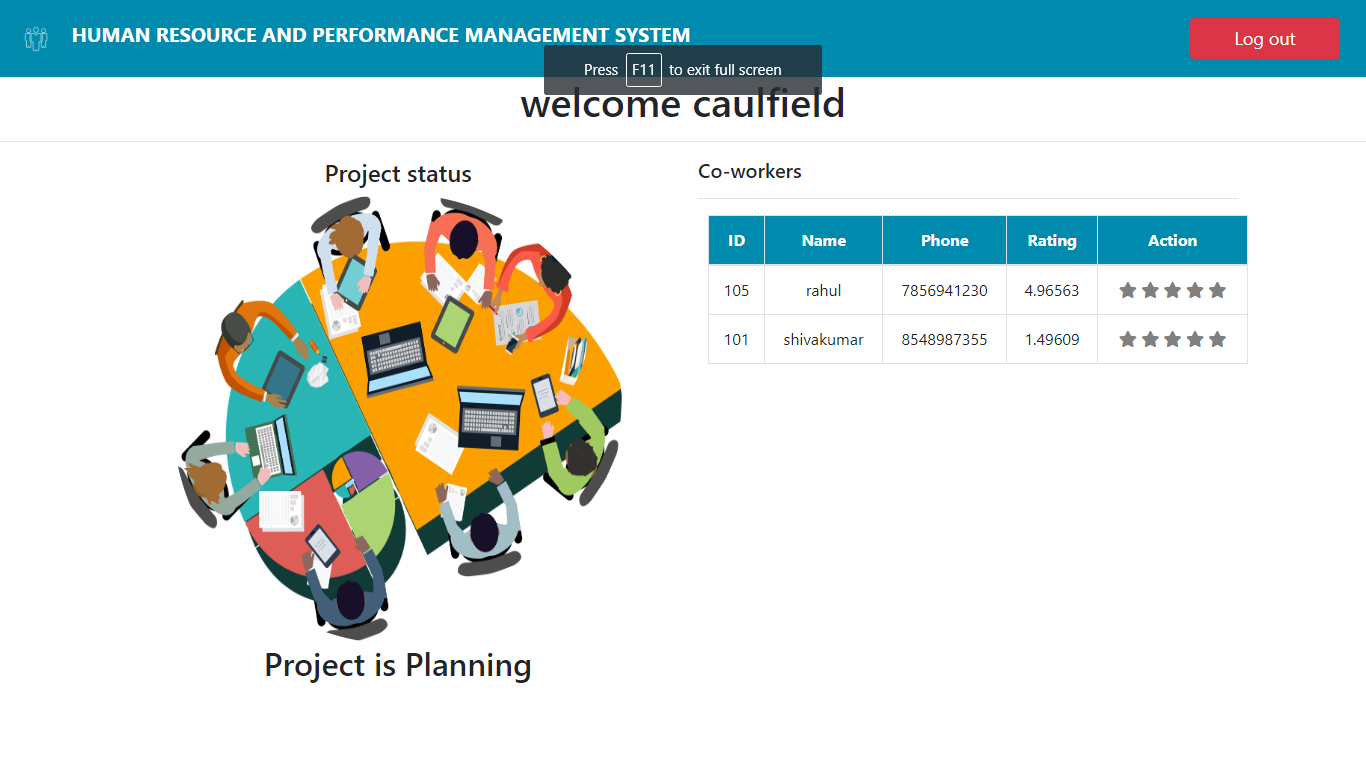






The client after delivery has the option to add another project using the “+ADD NEW”

If the employee is chosen at the user page we are greeted with employee login page

On successful login, the employee gets the status of the project and can rate his colleagues and peers.

This page also updates the status as the manager updates.

All the users can logout anytime using the “Logout” button on the top right side of the screen.

**Conclusion**

What we have here is easy to use, highly beneficial website. The design of the website is done by taking in to the consideration of all the user’s needs, boxing them and presenting it in the best way possible. Using the website saves the much needed time and helps streamline the entire process.

The website has been completed successfully and tested with suitable test cases .It is user friendly and contains suitable options for all users. This project is part of the spearhead the pierces the veil of redundancy and creates a future where everything is on the web and easily accessible.

**BIBLIOGRAPHY**

* The Bootstrap website
* PHP official Documentation
* Mysql Documentation
* Stack Overflow
* Fundamentals of Database Systems by Ramez Elmasri