**Chapter 1**

**Introduction**

**1.1 Human resource and performance management system**

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

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.

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.

**Chapter 2**

**LITERATURE SURVEY**

**2.1 Traditional File System**

In the early days of computing, data management and storage were a very new concept for organizations. The traditional approach to data handling offered a lot of convenience of the manual approach to business processes (e.g. handwritten invoices & account statements, etc.) as well as the benefits of storing data electronically.

The traditional approach usually consisted of custom-built data processes and computer information systems tailored for a specific business function. An accounting department would have their own information system tailored to their needs, where the sales department would have an entirely separate system for their needs.

Initially, these separate systems were very simple to set up as they mostly mirrored the business process that departments had been doing for years but allowed them to do things faster with less work. However, once the systems were in use for so long, they became very difficult for individual departments to manage and rely on their data because there was no reliable system in place to enforce data standards or management.

Separate information systems for each business function also led to conflict of interest within the company. Department felt a great deal of ownership for the data that they collected, processed, and managed which caused many issues among company-wide collaboration and a high rate of unreliable and inconsistent data.

**2.2 Procs and Cons of the Traditional Approach**

**Proc**

* Simple
* Matched existing business processes and functions
* Companies were not as interested in funding complicated system
* Initially low-cost
* Early computing was not viewed as beneficial for large funding
* Systems were designed to be cheap in order to save on cost

**Cons**

* Unmanaged redundancy
* Multiple instance of the same data appeared throughout various files, systems, and databases
* Information updated in one place was not replicated to the other locations.
* Disk space was very expensive, and redundancy had a big impact on storage.
* Data inconsistency
* Redundant data stored in various locations was usually never stored the same way
* Formatting was not centrally managed
* Lack of data sharing
* Same data stored in multiple locations
* Caused unnecessary doubling of efforts for processing and managing data.

**2.3 Downfall of Traditional Management System**

Conceived in a relatively centralized era when software was deployed in static environment, legacy database architecture fails to support an increasingly mobile world where applications are accessed anytime, anytime, anywhere. Today software users want consistent improvements in usability and expert vendors to deliver new features and functionality needed to achieve their business objectives.

However, legacy database technologies fall short in serving the needs of today’s distributed and cloud environments for the following reasons:

* Inadequate failover capabilities
* Latency issues.
* Increasing operational cost.
* Inability to meet the demands of global markets

For all of these reasons, traditional databases are unable to deliver results in a rapidly growing environment where the workload is geographically distributed across datacenters.

Updating to a move distributed data model is costly and complicated and your DBA’s can’t just sit back and give up on this situation. Hence due to these various reasons, the downfall of the traditional system was inevitable.

**2.4 Introduction to the Database Management System**

A database management system (DBMS) refers to the technology for creating and managing databases. Basically, a DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

The main aim of a DBMS is to supply a way to store and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft Access, or EXCEL to store data in the form of database. A datum is a unit of data.

**2.5 Advantages of a DBMS**

A Database Management System has many advantages over the traditional file system used in the earlier days.

* Data independence: Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
* Efficient data access: DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.
* Data integrity and security: If data is accessed through the DBMS, the DBMS can enforce integrity constraints on the data.
* Data administration: when several users shares the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.

**Chapter 3**

**SYSTEM REQUIREMENTS**

**3.1 Hardware Requirements**

* Processor: Pentium IV or above
* RAM: 2 GB or above
* Hard Disk: 2 GB or above

**3.2 Software Requirements**

Technologies used:

* Front End: HTML, CSS, PHP
* Connection: PHP
* Back-End/Database: MySQL

Software:

* Text Editor: Notepad ++
* Server: Apache (on XAMPP)
* Operating System: Windows 10
* Database Support: MySQL 3.2
* Back-End: PHP 7

**Chapter 4**

**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.

**4.1 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.

**4.1.1USE CASE DIAGRAM**

* **Use case for admin**



Fig.4.1 Use case of 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**



Fig.4.2 Use case of 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**



Fig.4.3 Use case of 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**



Fig.4.4 Use case of client

**4.2 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.



Fig.4.5 ER diagram

**4.3 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.

Employee

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Emp\_id | Name | Address | Phone | Gender | Role | Skill | Salary | Username | Password | rating |

Emp\_state

|  |  |
| --- | --- |
| Emp\_id | state |

Client

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ClientID | C\_name | Username | Password | Phone | Address | Requirement |

Project

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P\_no | P\_name | Mgr\_id | Start\_date | Final\_date | Client\_id |

Project\_state

|  |  |  |
| --- | --- | --- |
| P\_status | Client\_id | P\_no |

Works on

|  |  |
| --- | --- |
| Emp\_id | P\_no |

Pre\_project

|  |  |  |  |
| --- | --- | --- | --- |
| C\_id | P\_id | Pre\_req | P\_state |

Fig.4.6 Schema diagram

**Chapter 5**

**IMPLEMENTATION**

**5.1 HTML**

First developed Tim Berners-Lee in 1990, **HTML** is short for **Hypertext Markup Language**. HTML is used to create electronic documents (called pages) that are displayed on the world Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively **easy to learn**, with the basics being accessible to most people in one sitting; and quite **powerful** in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the organisation charged with designing and maintaining the language.

The definition of HTML is **Hyper Text Markup Language**.

* Hypertext is the method by which you move around on the web — by clicking on special text called **hyperlinks** which bring you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.
* Mark up is what **HTML tags** do to the text inside them. They mark it as a certain type of text (*italicized* text, for example).
* HTML is a Language, as it has code-words and syntax like any other language.

**5.2 CSS**

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the mark up languages HTML or XHTML.

**Advantages of CSS**

* **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

**5.3 JavaScript**

Java script is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **Live Script,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **Live Script**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

**Client-Side Java Script**

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

**Advantages of Java Script**

* **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.
* **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

**5.4 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.

**Reasons for using PHP**

There are some indisputable great reasons to work with PHP. As an open source product, PHP is well supported by a talented production team and a committed user community. Furthermore, PHP can be run on all the major operating systems with most servers.

* **Learning PHP is easy** Basic is easy any interpreted language should be easy to learn. Since you are isolated from the system (no pointers to use, no memory to allocate). The other advantage that all modern interpreted languages share is good associative array constructs.
* **Its Performance**

While we can build an application that serves millions of pages a day on a server, when we really look at the performance of the language it sucks. We are still orders of magnitude from real performance. Not only that, but since PHP is designed around a single process model our ability to share data structures or connection pool resources is left to native code libraries.

* **The low cost**

There are many languages which are available at very less cost. There are some languages which are available at very less cost .

* **It’s Open Source, we can modify it:**

We can modify it if you need a hole in your head! Technically the point is that it’s an open source project and they release patches often. You’re point is that the community is actively working out the bugs. So, what any active language is doing this...

Unfortunately, C, C++ and Perl have all “died” at this point and will pretty much remain static at their current functionality.

* **PHP Syntax**

You cannot view the PHP source code by selecting “View source” in the browser you will only see the output from the PHP file, which is plain HTML. This is because the scripts are executed on the server before the result is sent back to the browser

**5.5 SQL**

Structured Query Language (SQL) is a standard computer language for relational database management and data manipulation. SQL is used to query, insert, update and modify data. Most relational databases support SQL, which is an added benefit for database administrators (DBAs), as they are often required to support databases across several different platforms.

First developed in the early 1970s at IBM by Raymond Boyce and Donald Chamberlin, SQL was commercially released by Relational Software Inc. (now known as Oracle Corporation) in 1979. The current standard SQL version is voluntary, vendor-compliant and monitored by the American National Standards Institute (ANSI). Most major vendors also have proprietary versions that are incorporated and built on ANSI SQL, e.g., SQL\*Plus (Oracle), and Transact-SQL (T-SQL) (Microsoft).

**5.6 Creation of table**

* **Admin:**

CREATE TABLE `admin` (`username` varchar(10) DEFAULT NULL, `Password` varchar(10) DEFAULT NULL);

* **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);

* **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);

* **Emp\_state:**

CREATE TABLE `Emp\_state`(`Emp\_id` int(11) DEFAULT NULL, `state` int(11) DEFAULT NULL);

* **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);

* **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);

* **Project\_state:**

CREATE TABLE `project\_state` (`p\_status` int(11) DEFAULT NULL,`client\_id` int(11) DEFAULT NULL,`pno` int(11) NOT NULL);

* **Works\_on:**

CREATE TABLE `works\_on` (`Empid` int(11) DEFAULT NULL,`pno` int(11) DEFAULT NULL);

* **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

**Chapter 6**

**TESTING**

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**6.1 TESTING PROCESS**

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

**6.2 TESTING OBJECTIVES**

The main objectives of testing process are as follows.

1. Testing is a process of executing a program with the intent of finding an error.

2. A good test case is one that has high probability of finding undiscovered error.

3. A successful test is one that uncovers the undiscovered error.

**6.3 Test cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Test results** | **Expected Results** | **Observed Results** | **Remarks** |
| 1 | Insert a subject | New tuple should be inserted | Query ok 1 row inserted into database | PASS |
| 2 | Insert a test | New tuple should be inserted | Query ok 1 row inserted into database | PASS |
| 3 | Insert a question | New tuple should be inserted | Query ok 1 row inserted into database | PASS |
| 4 | Delete a subject | Delete a subject | Query ok 1 row deleted from database | PASS |
| 5 | Create trigger | Trigger created | Query ok trigger created | PASS |
| 6 | Create stored procedure | Stored procedure created | Query ok trigger created | PASS |

**Chapter 7**

**RESULTS**

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

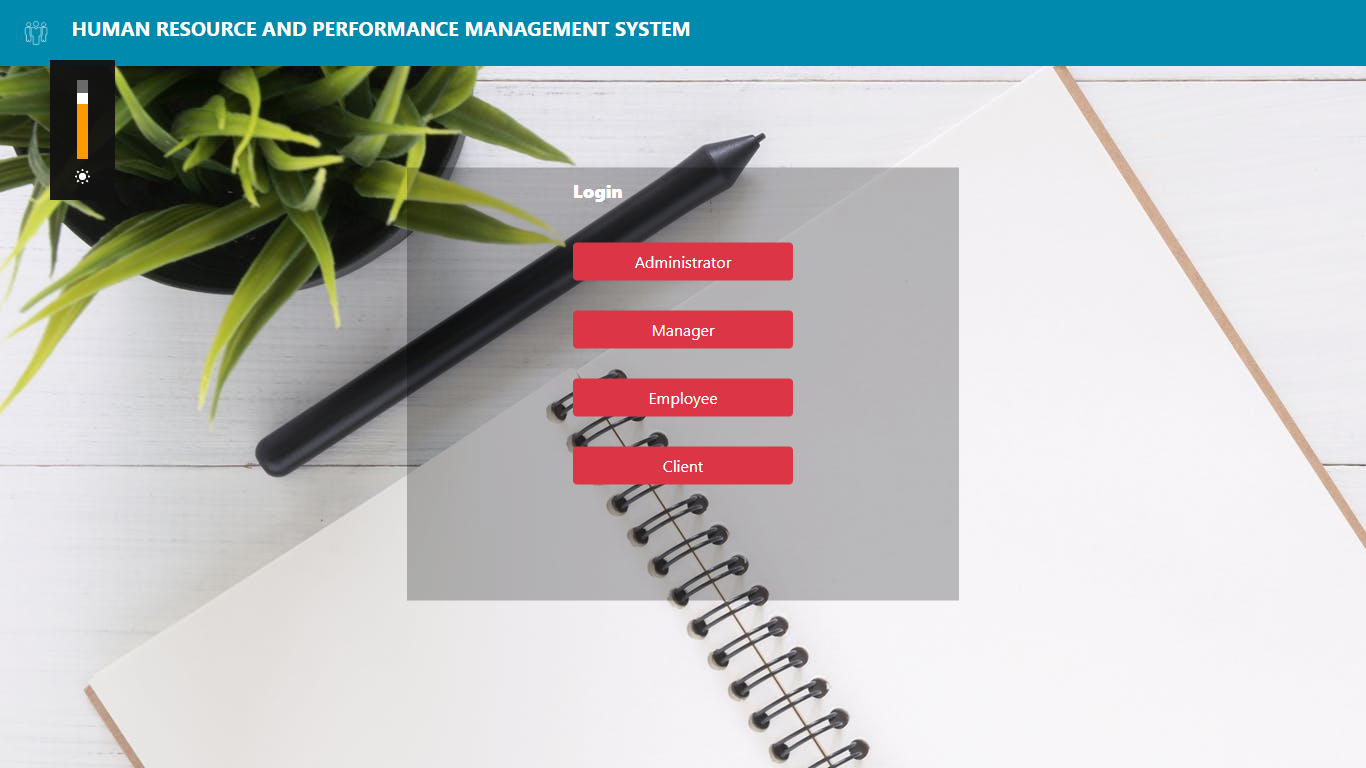


Fig.7.1 Login-direct page

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

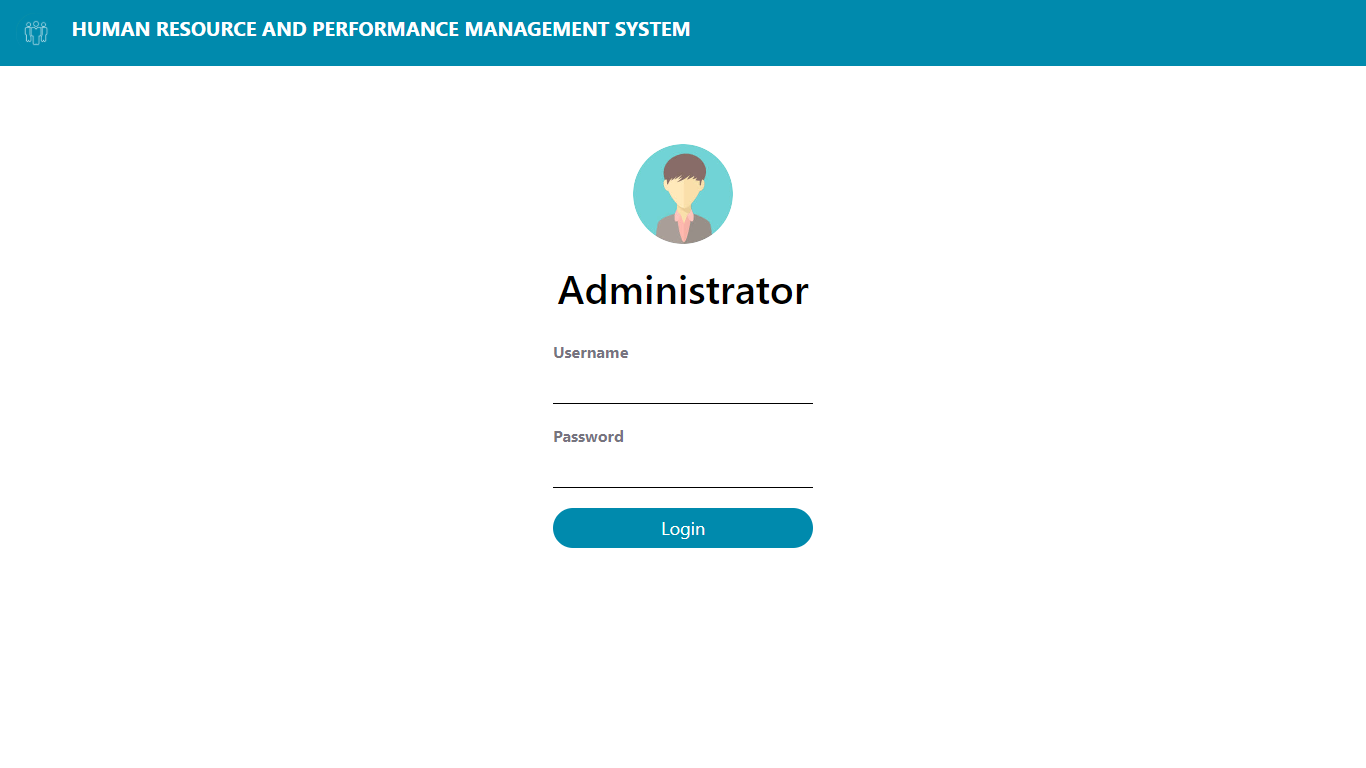


Fig.7.2 Admin login

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



Fig.7.3 Admin dashboard

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.

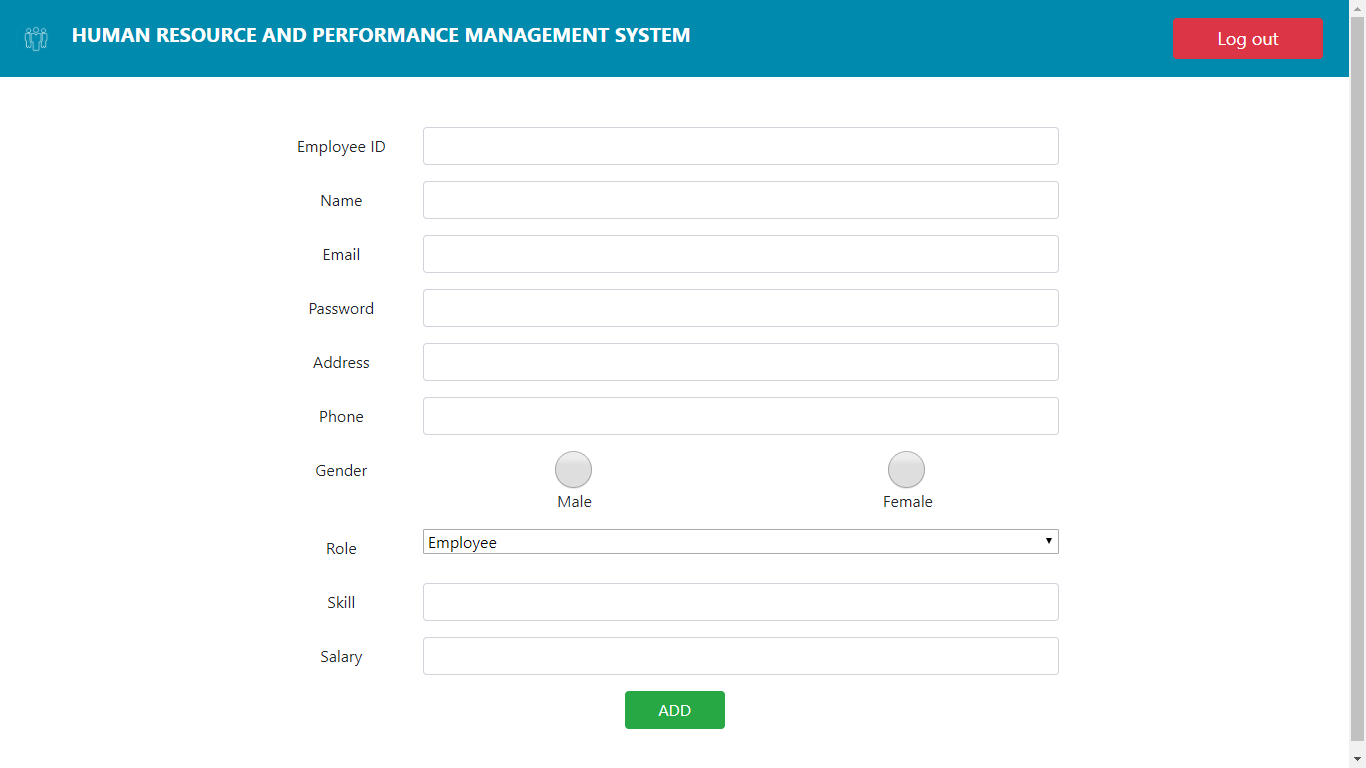


Fig.7.4 Employee insertion

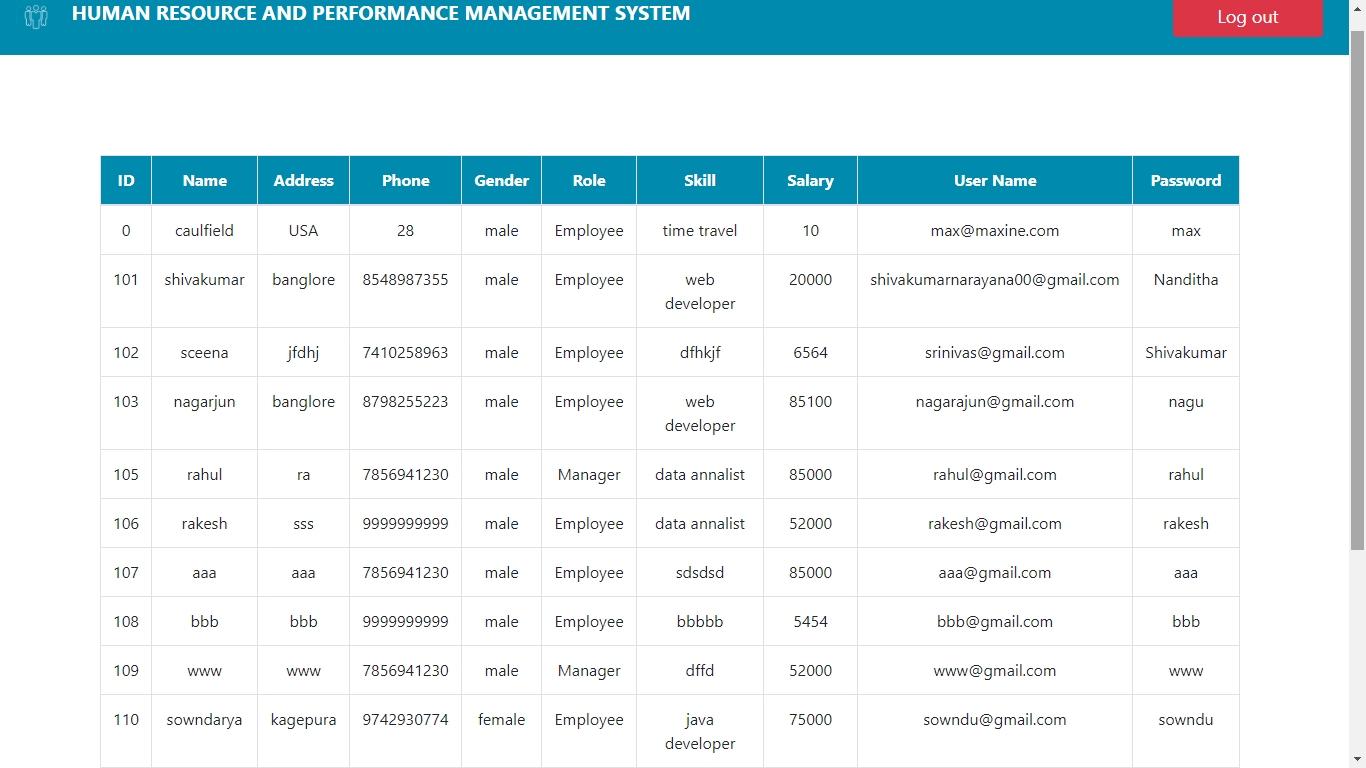
If we want to view all the present Employees, we choose “Employees” Button.

Fig.7.5 Display all Employee

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

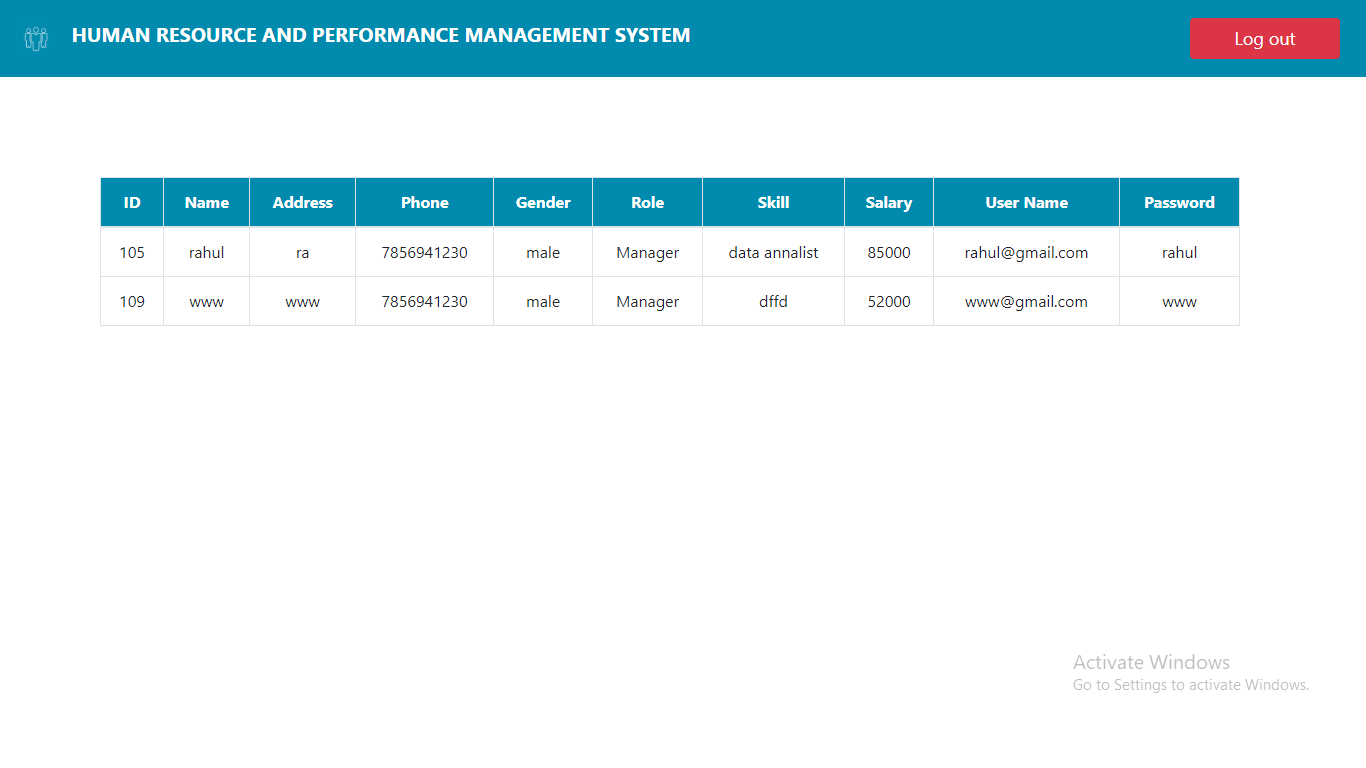


Fig.7.6 Display regional manager

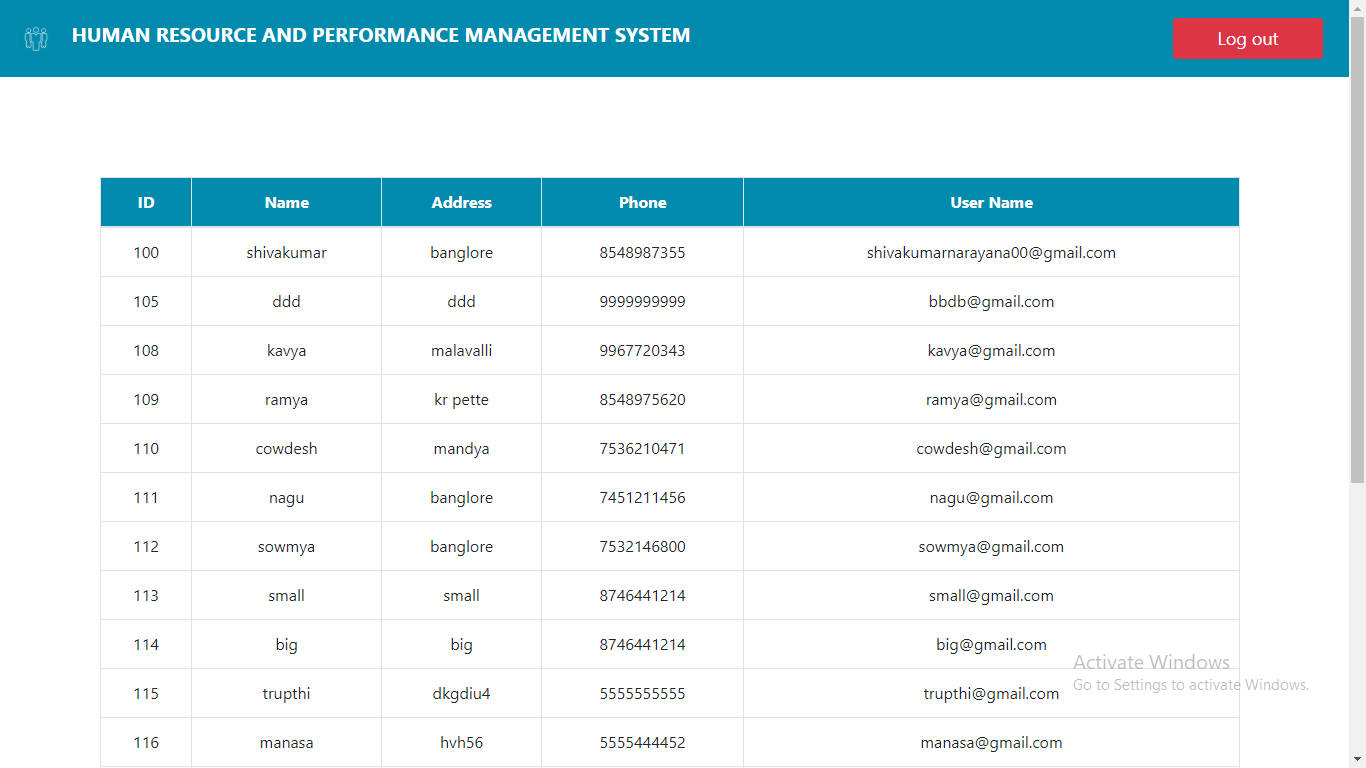
To see all current and previous clients we choose “Clients” Button

Fig.7.7 Display clients

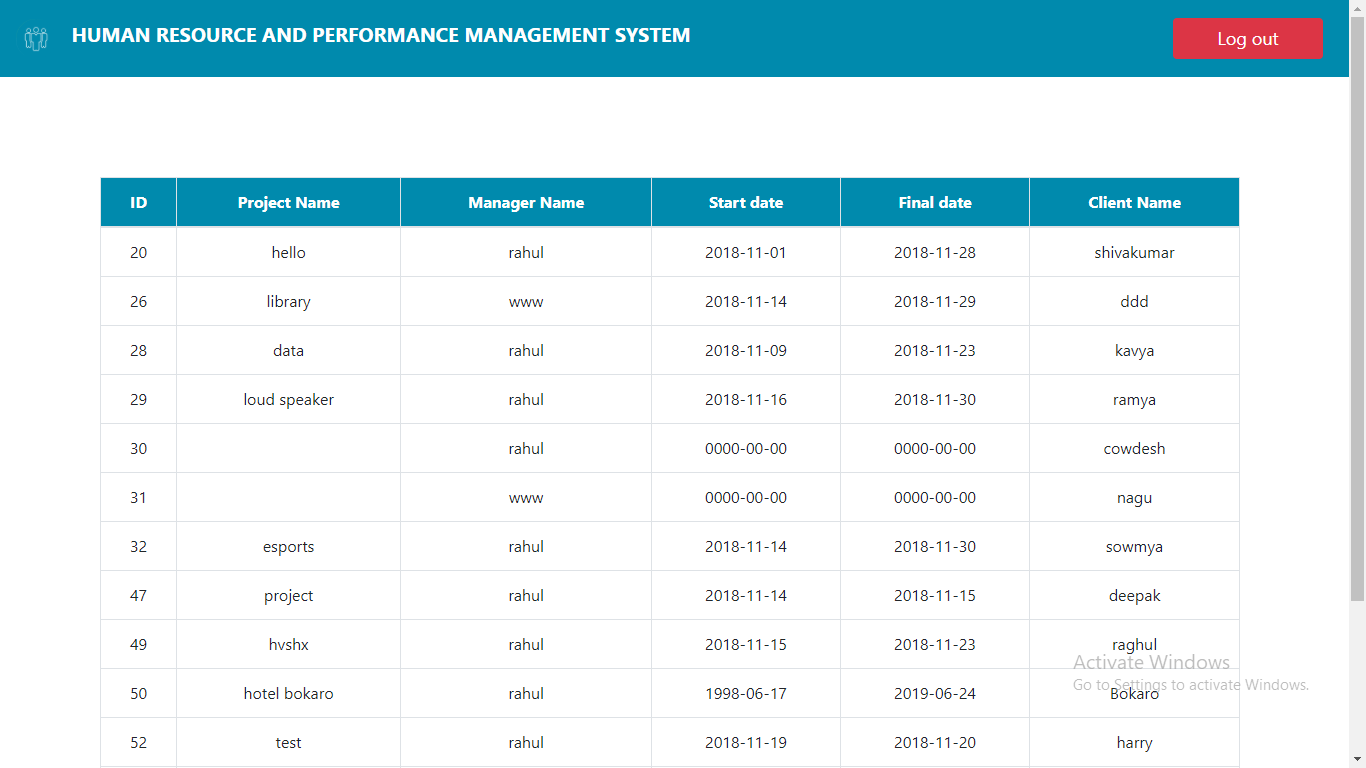
To view the projects we choose the button “Total Projects”

Fig.7.8 Display projects

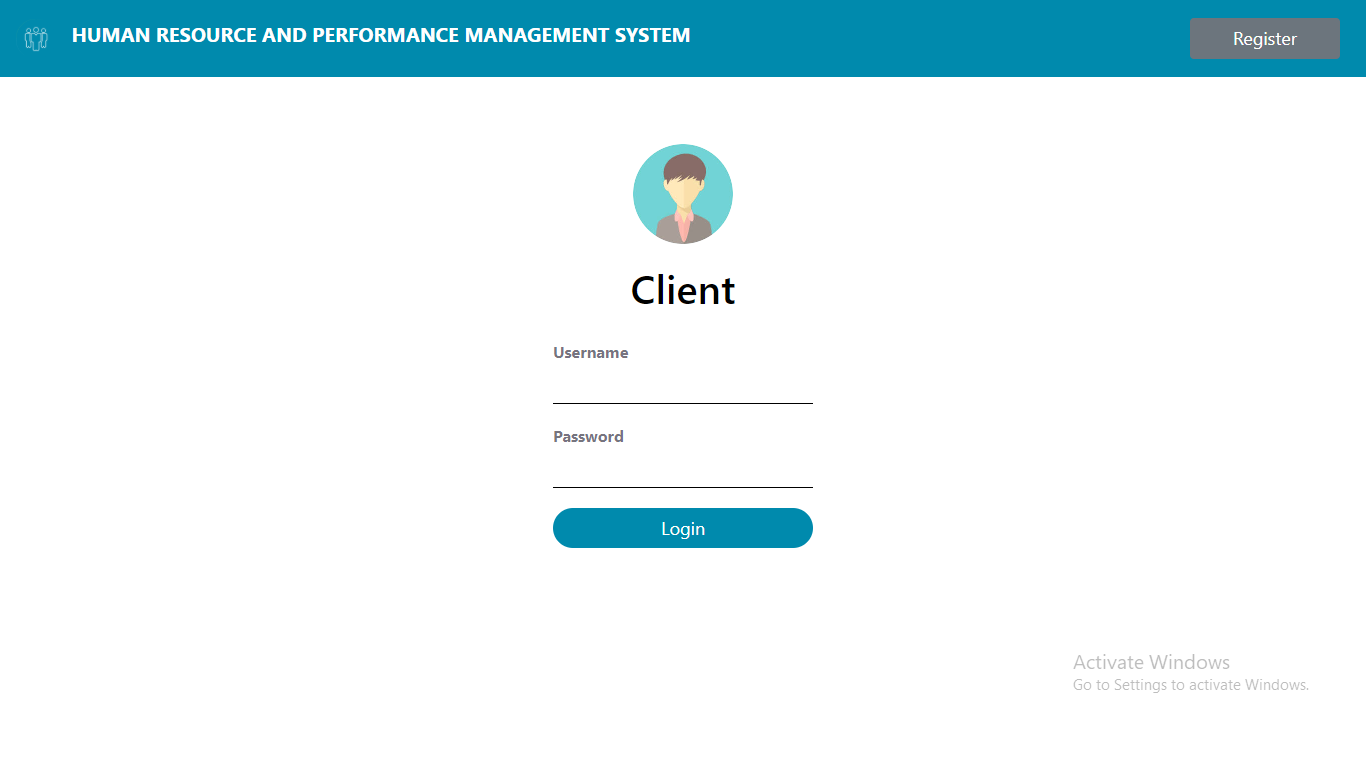
If we choose Client in the user page we are greeted with client login page,

Fig.7.9 Client login page

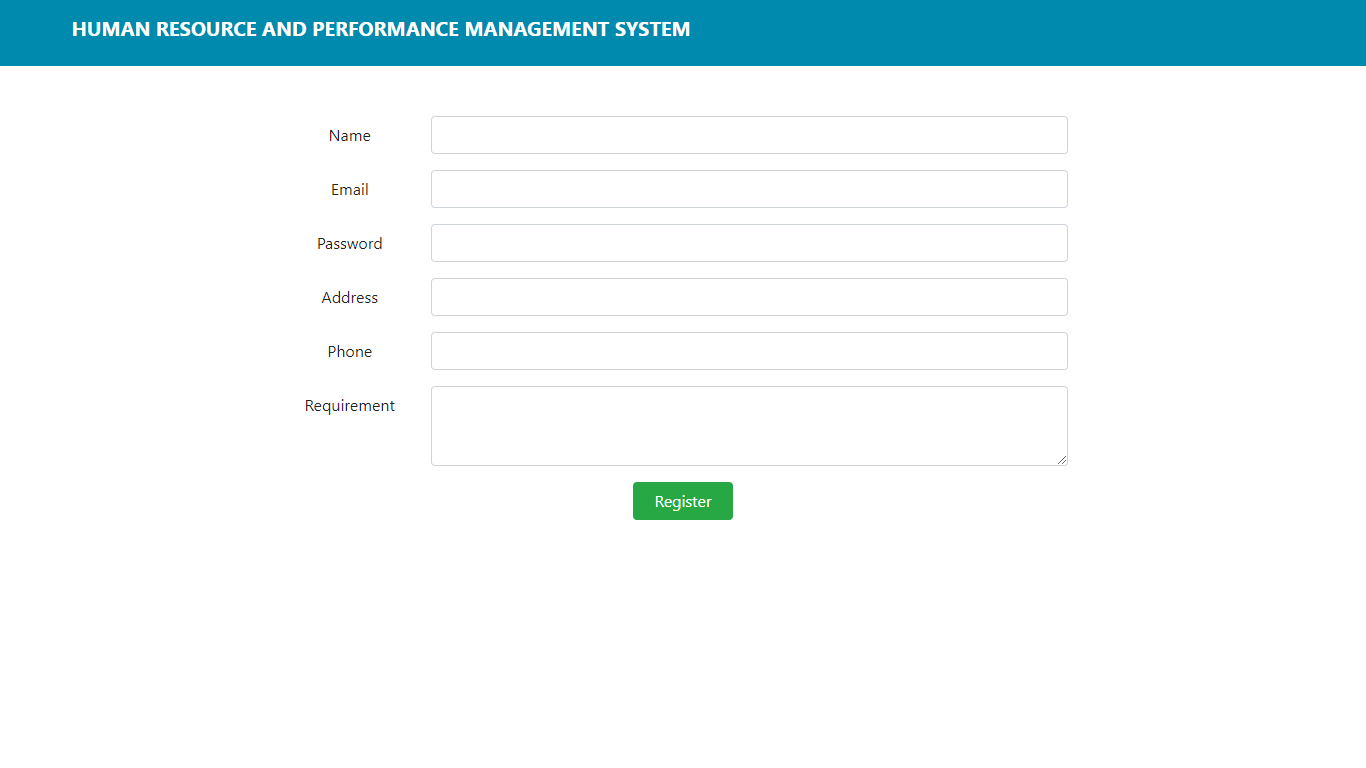
If it’s a new client, he/she can register using the register button

Fig.7.10 Client registration page

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

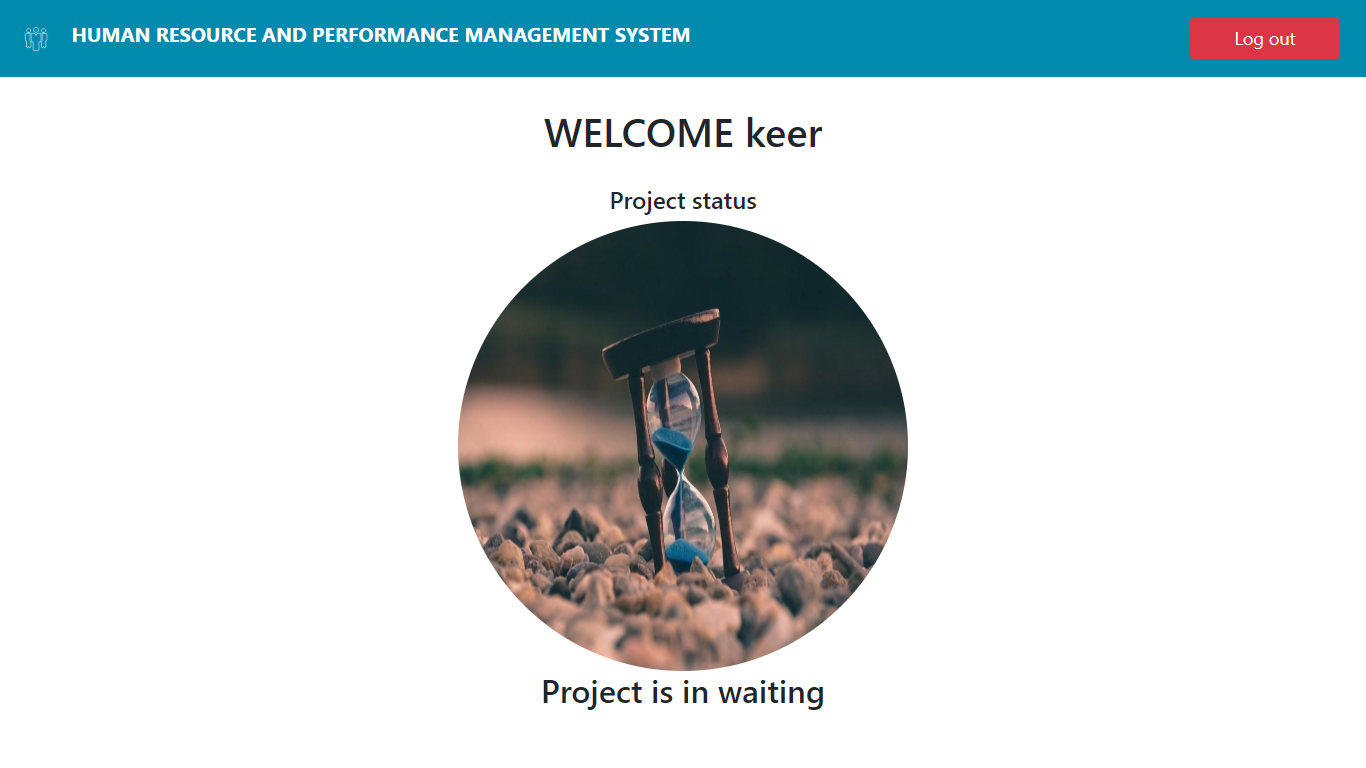


Fig.7.11 Client dashboard

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

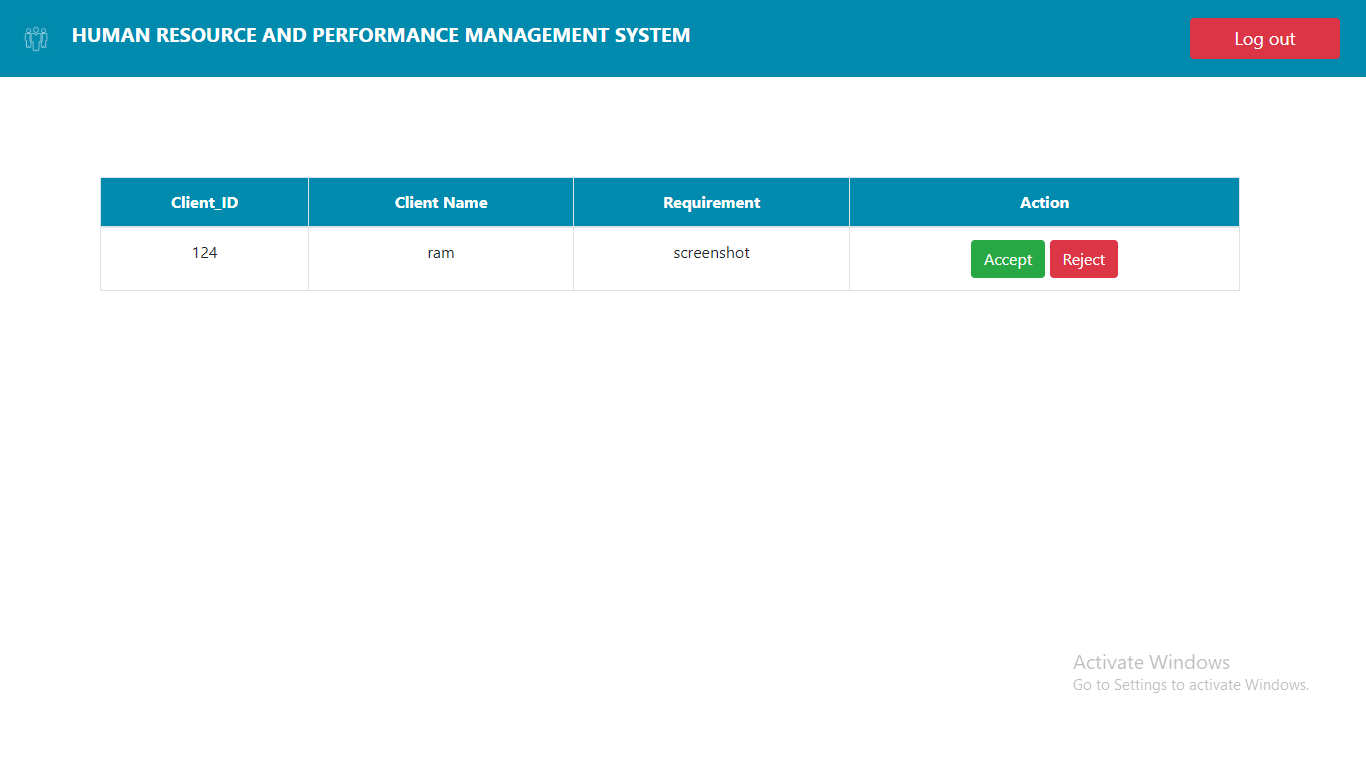


Fig.7.12 Accept/Reject project

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.

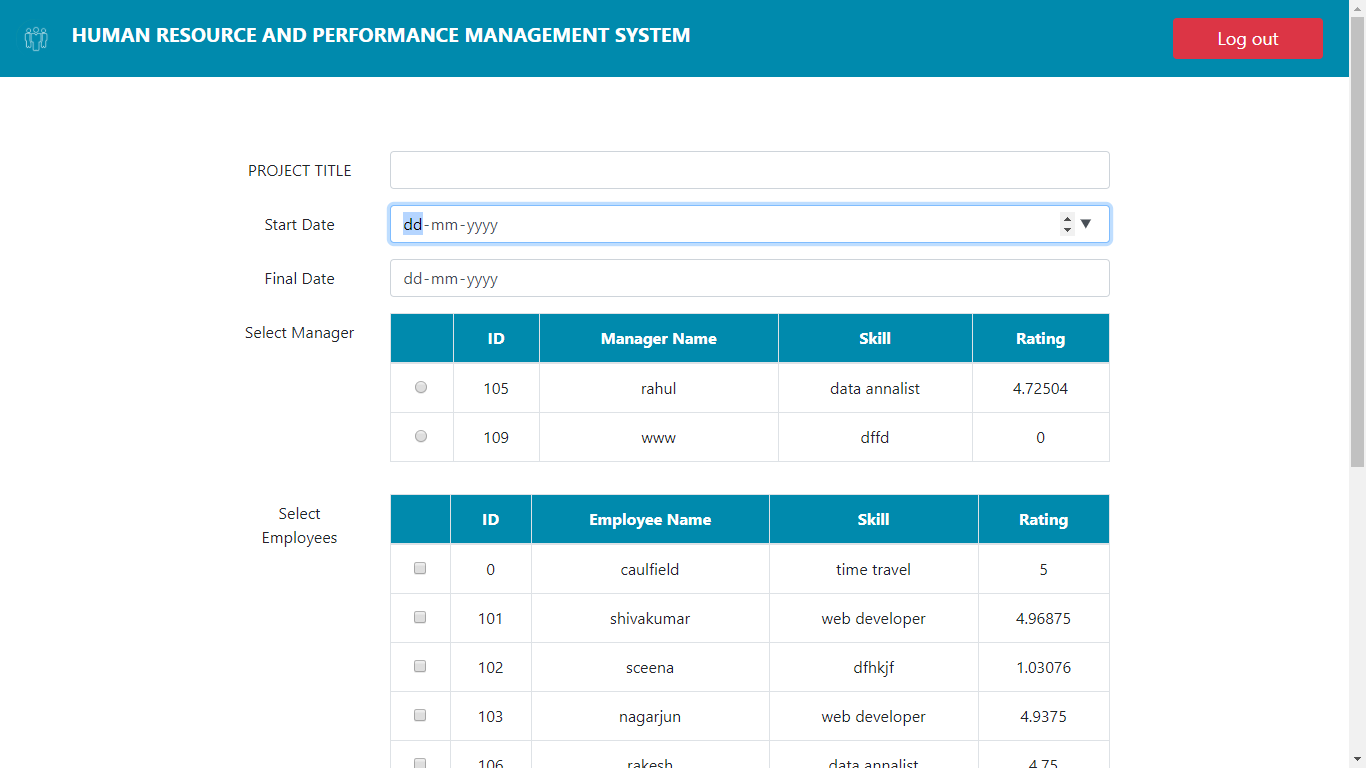


Fig.7.13 Initializing Employees to project

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

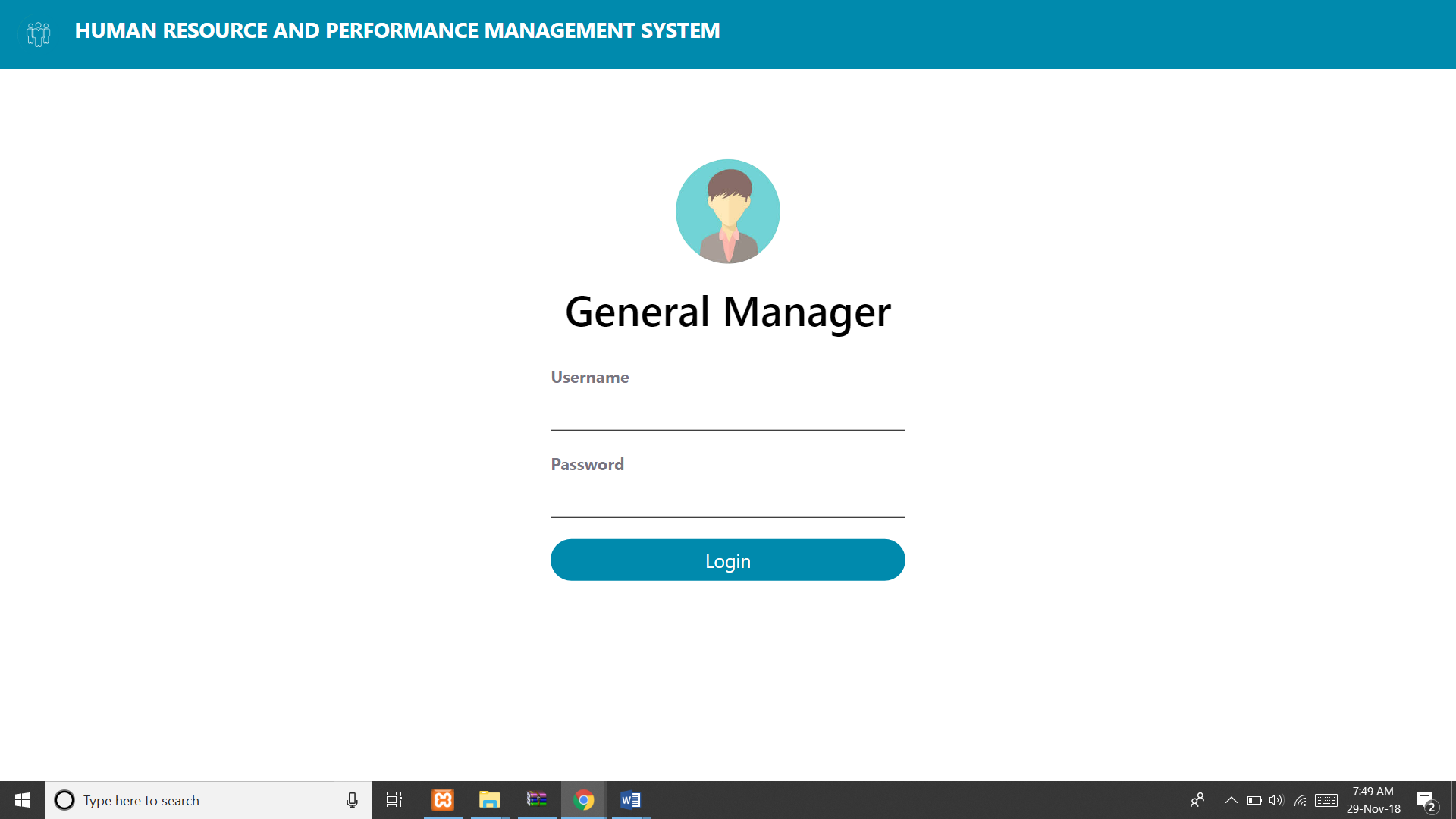


Fig.7.14 Manager Login

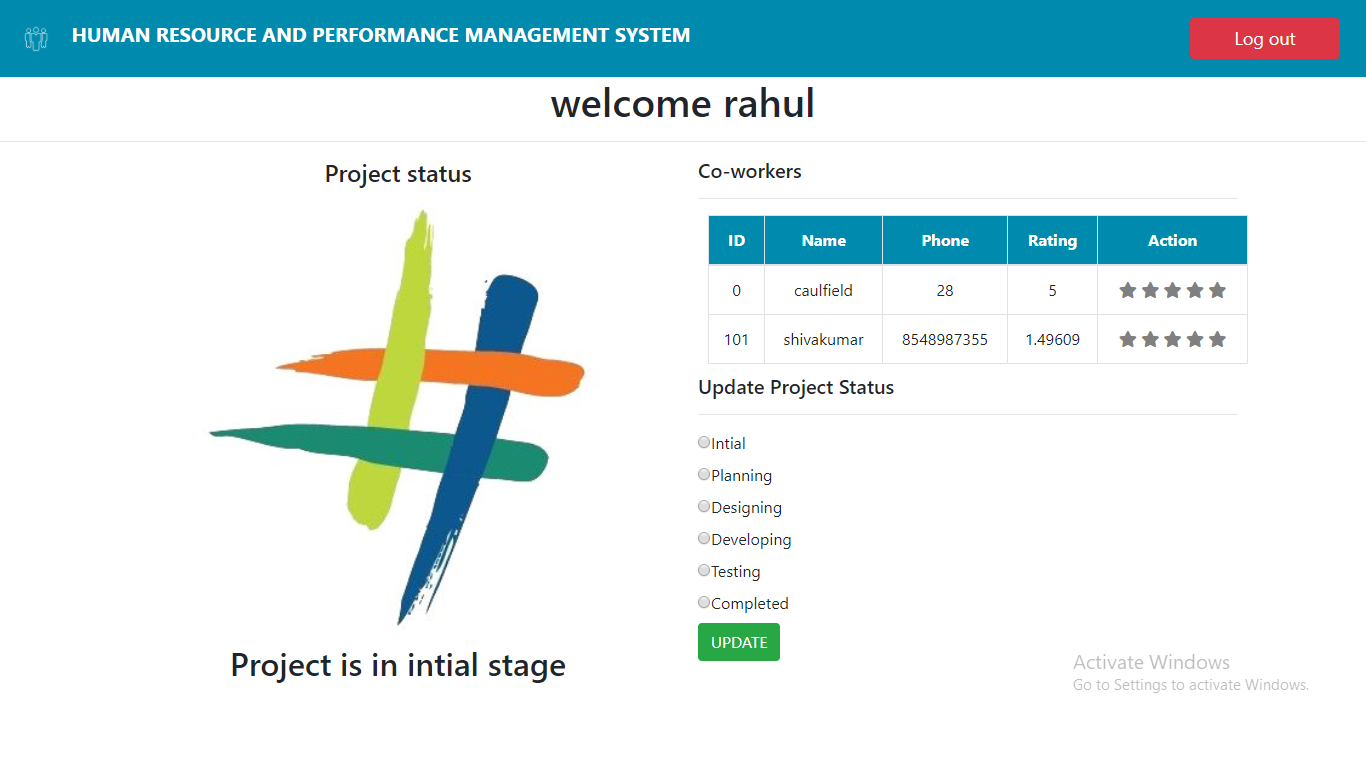
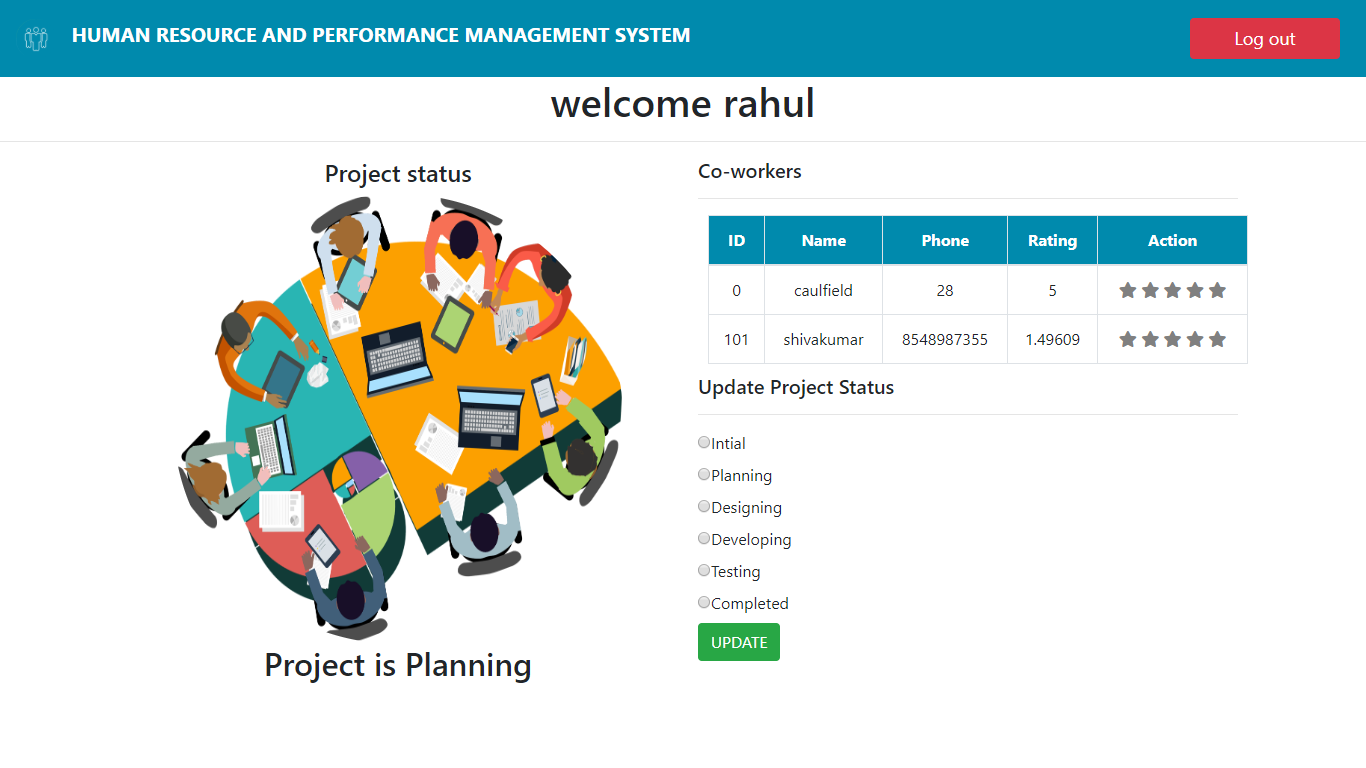
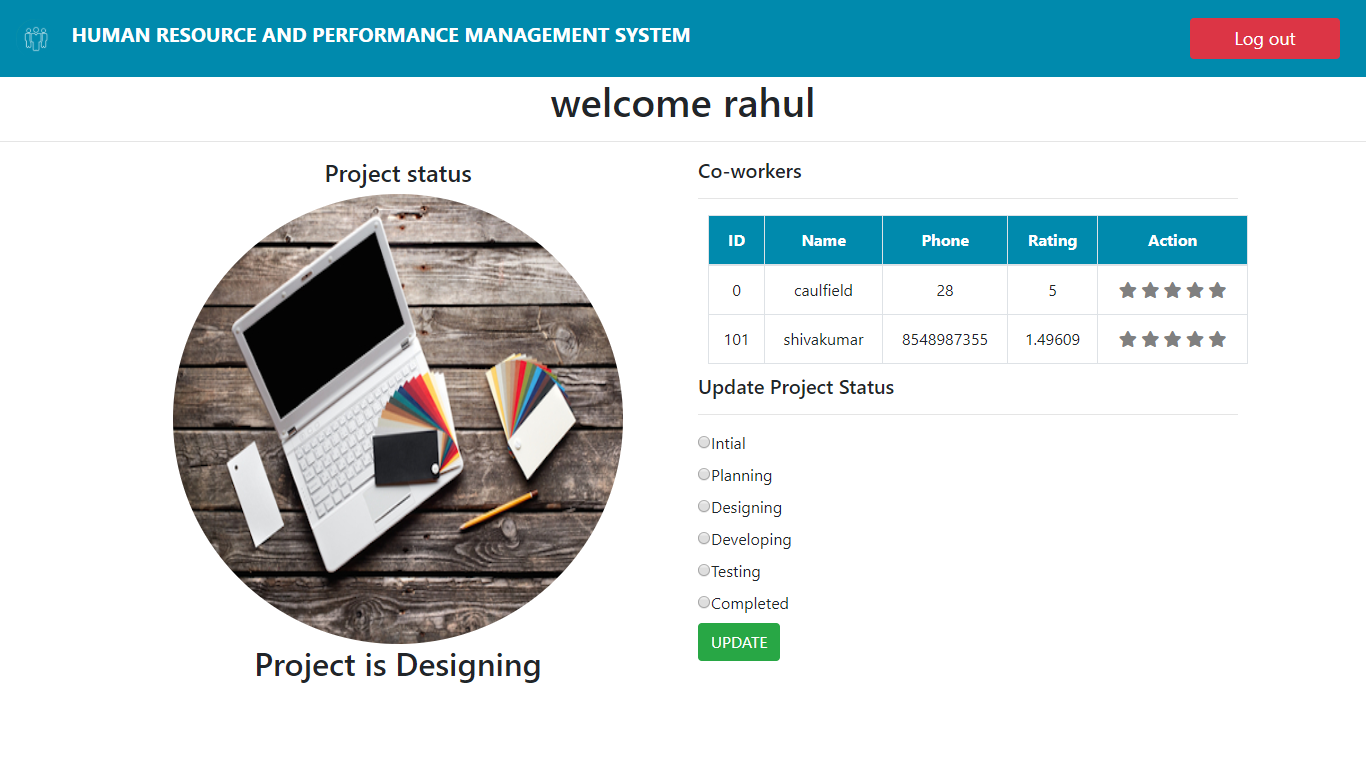
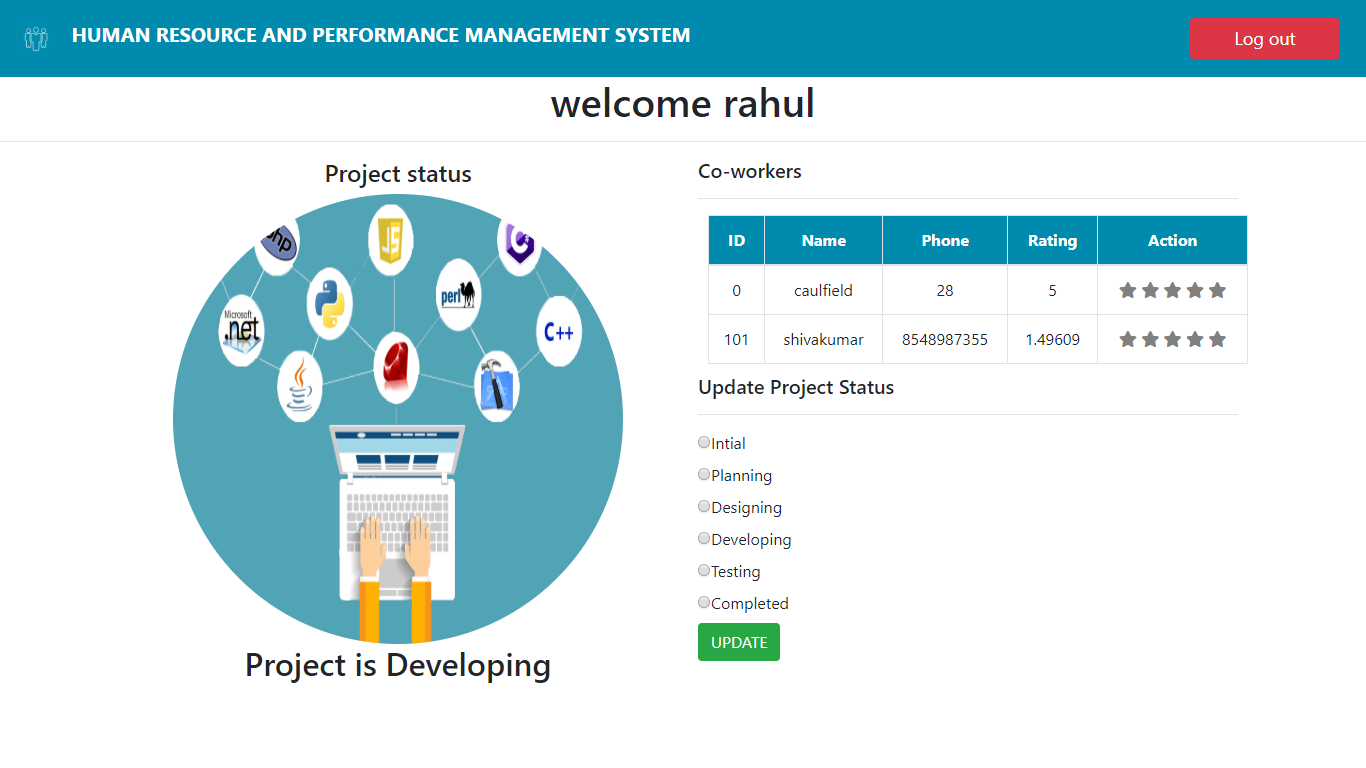
On successful login, the assigned manager can rate his fellow employees and also update the status of the project. There are mainly 6 stages that the manager can choose based on the progress of the project.

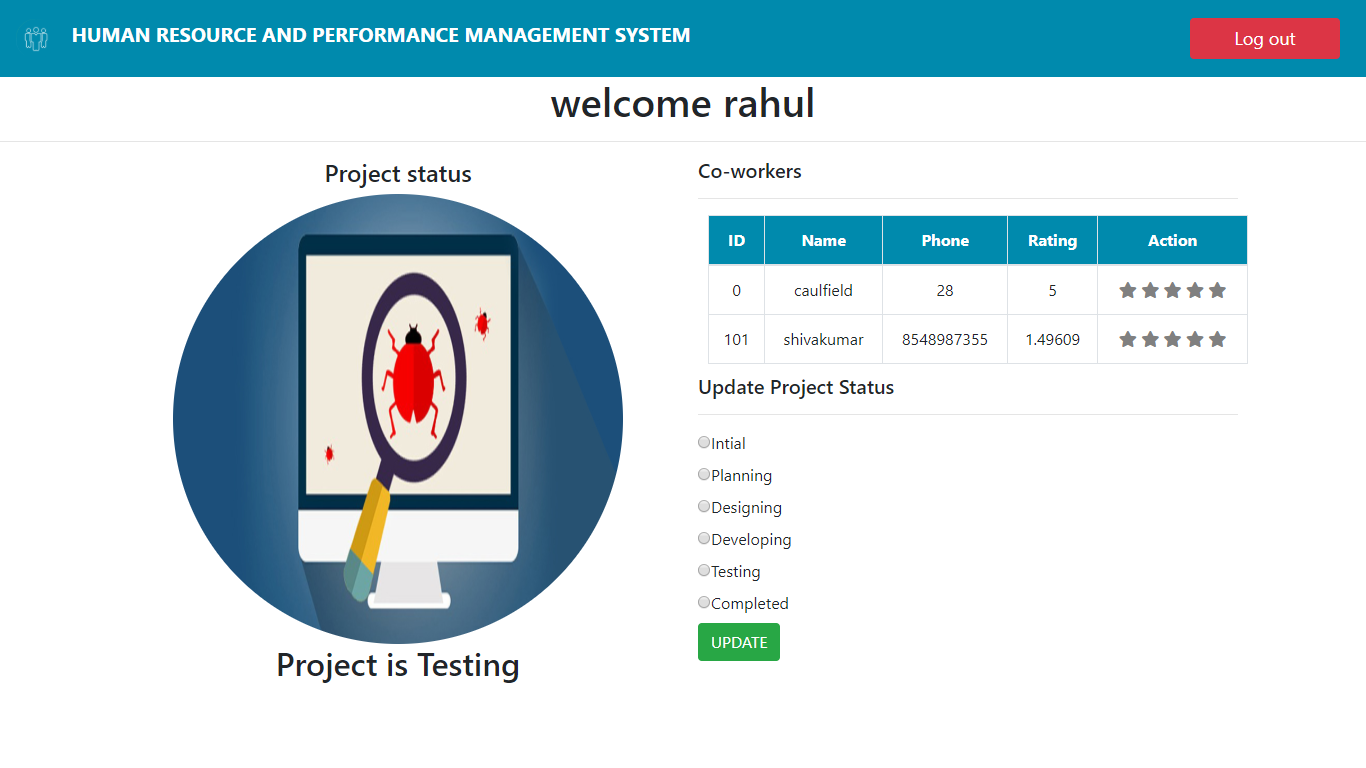
Fig.7.15 Manager Dashboard

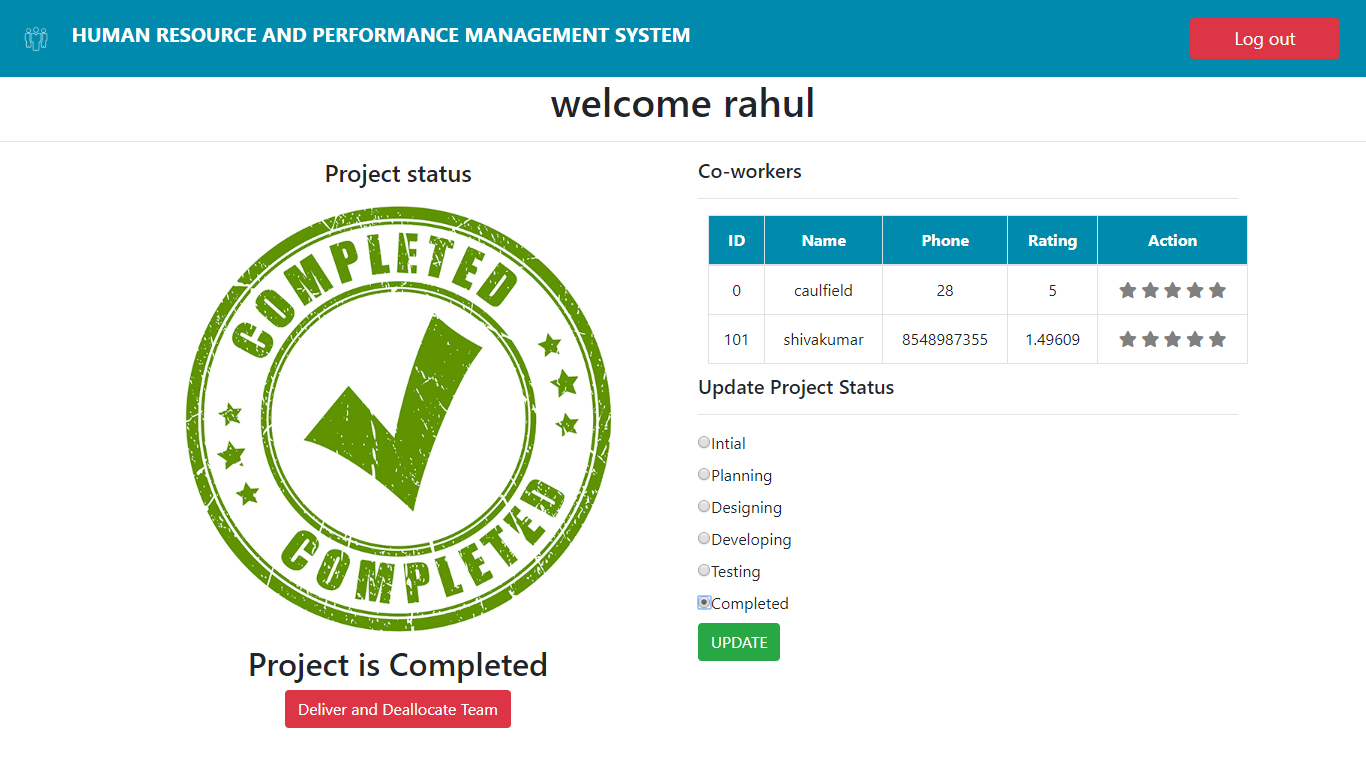
Choosing each stage and updating





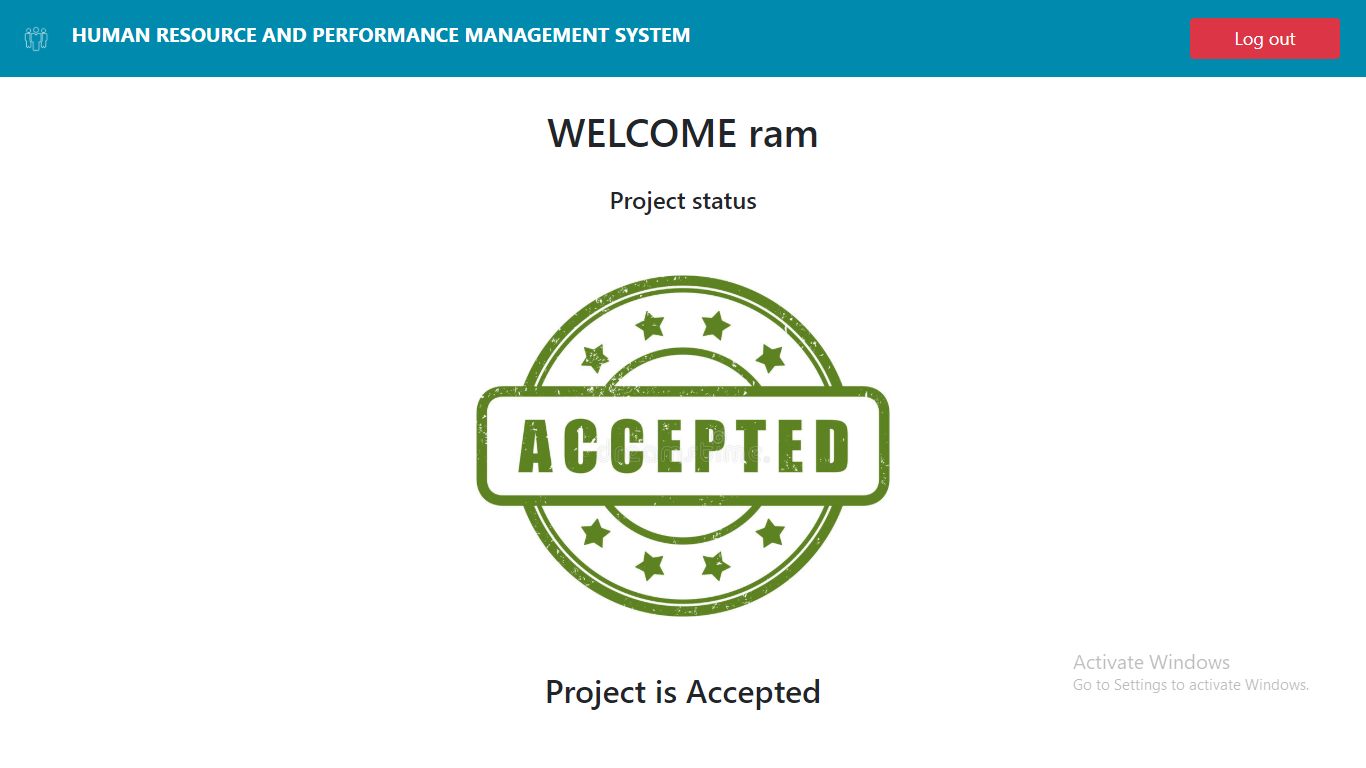




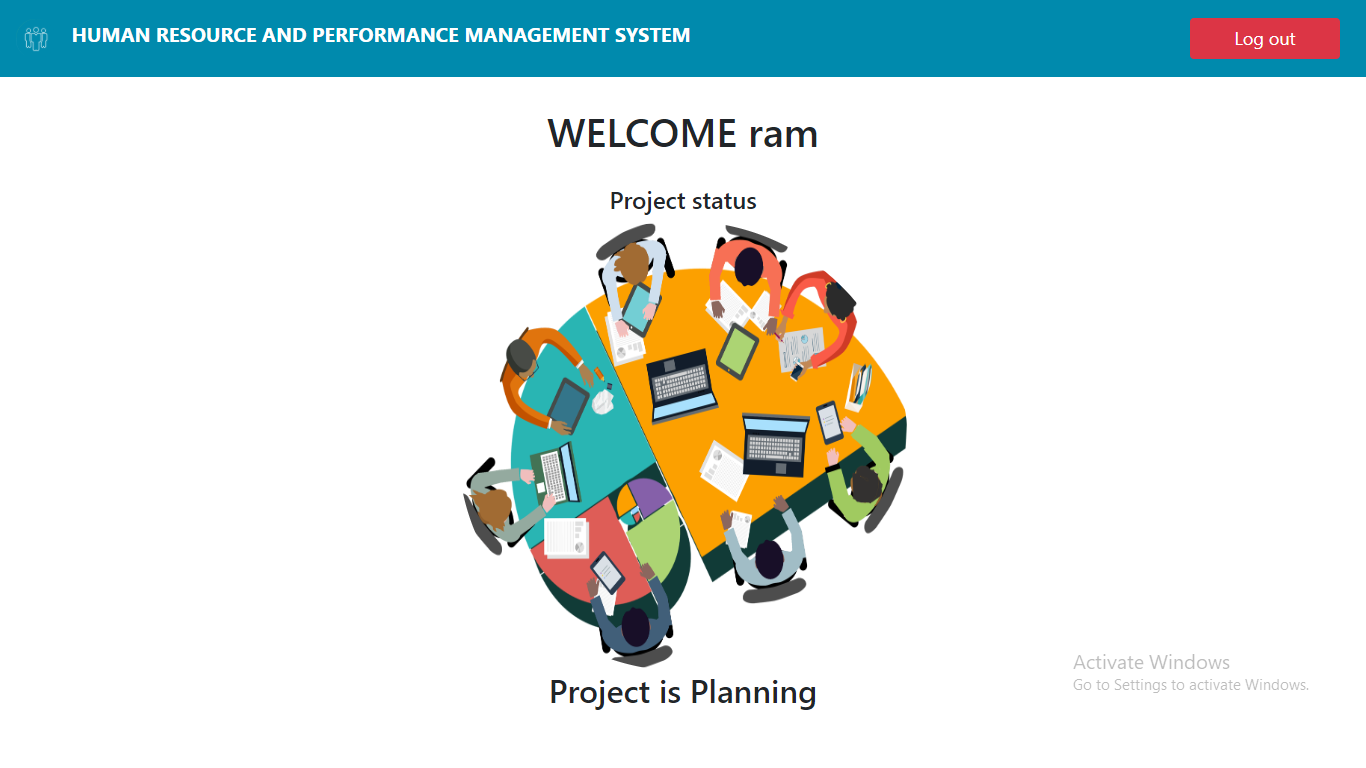


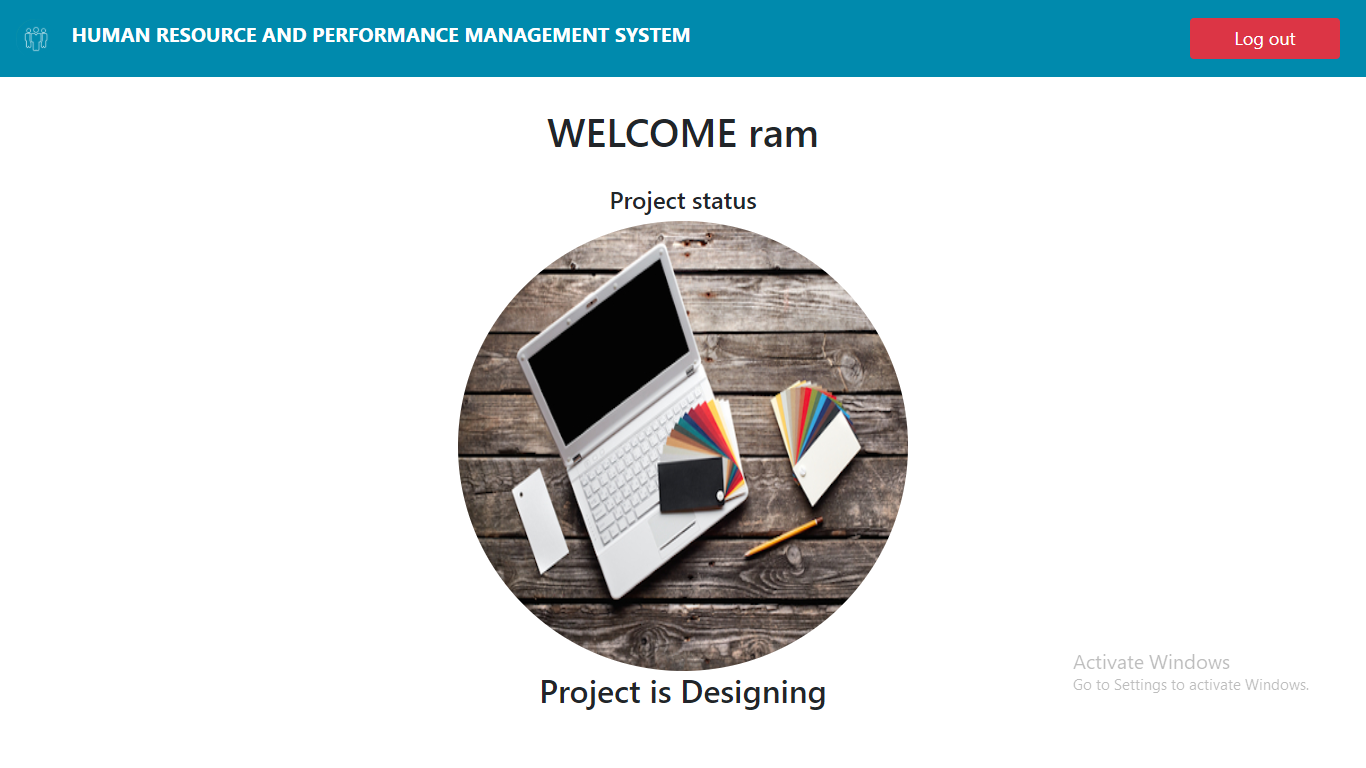
Once completed, the manager can deliver the project and deallocate the team.

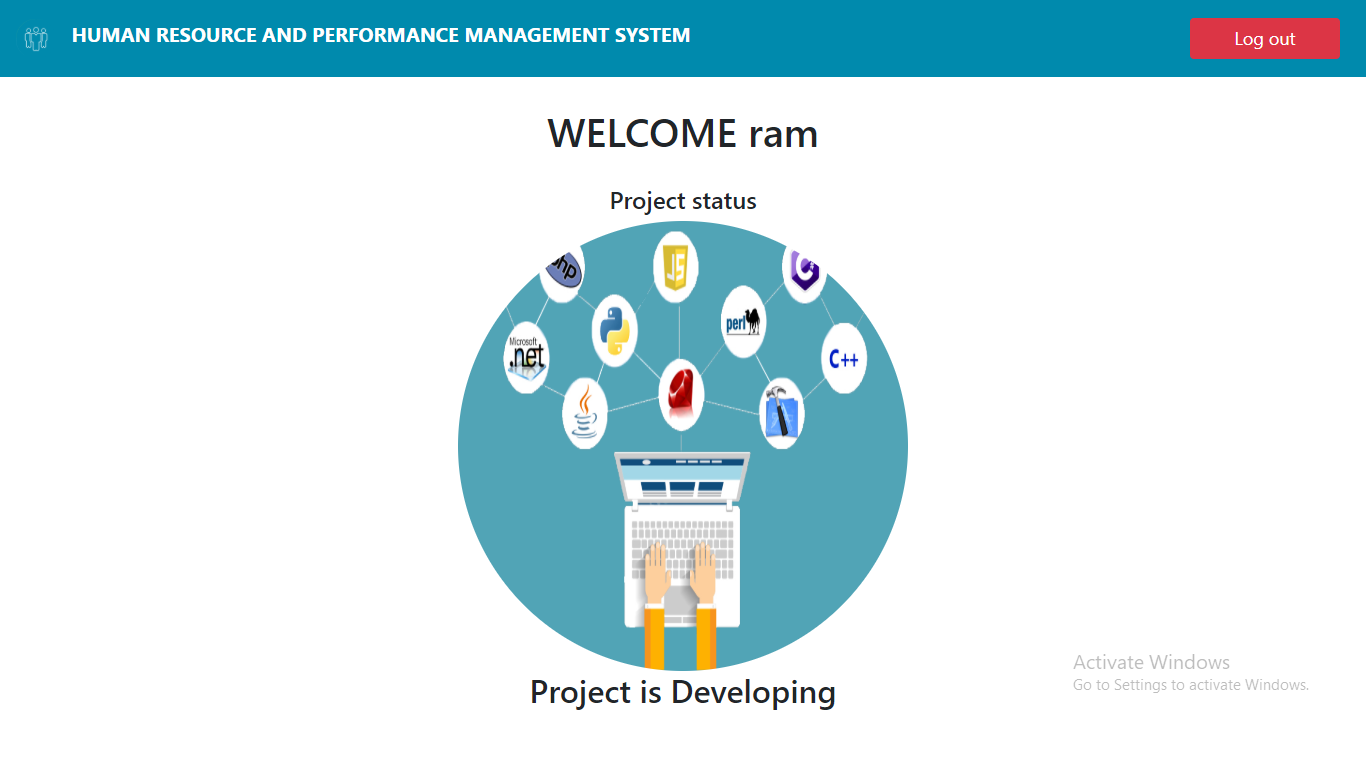
In the client side, if the project is updated then the status page displays:

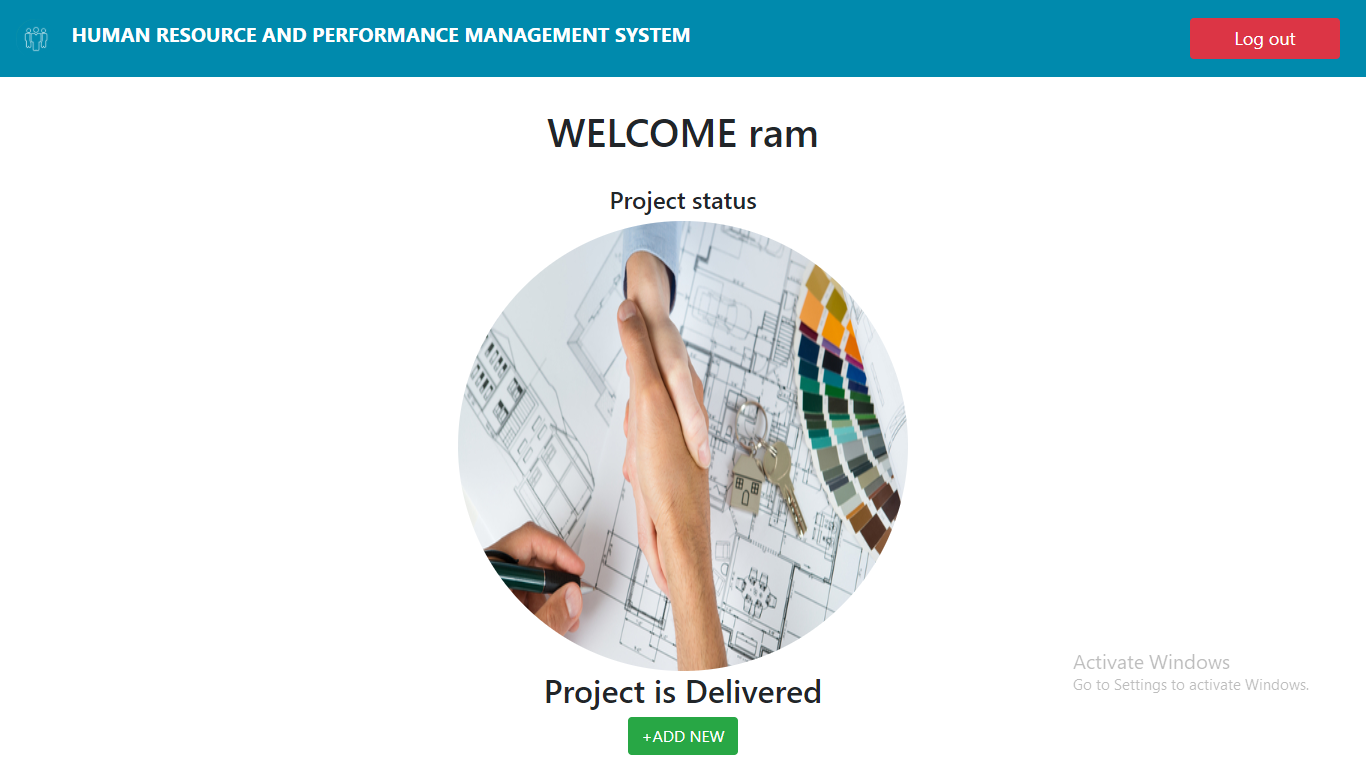


As the manager updates the progress of the project, this page shows the status to the client









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

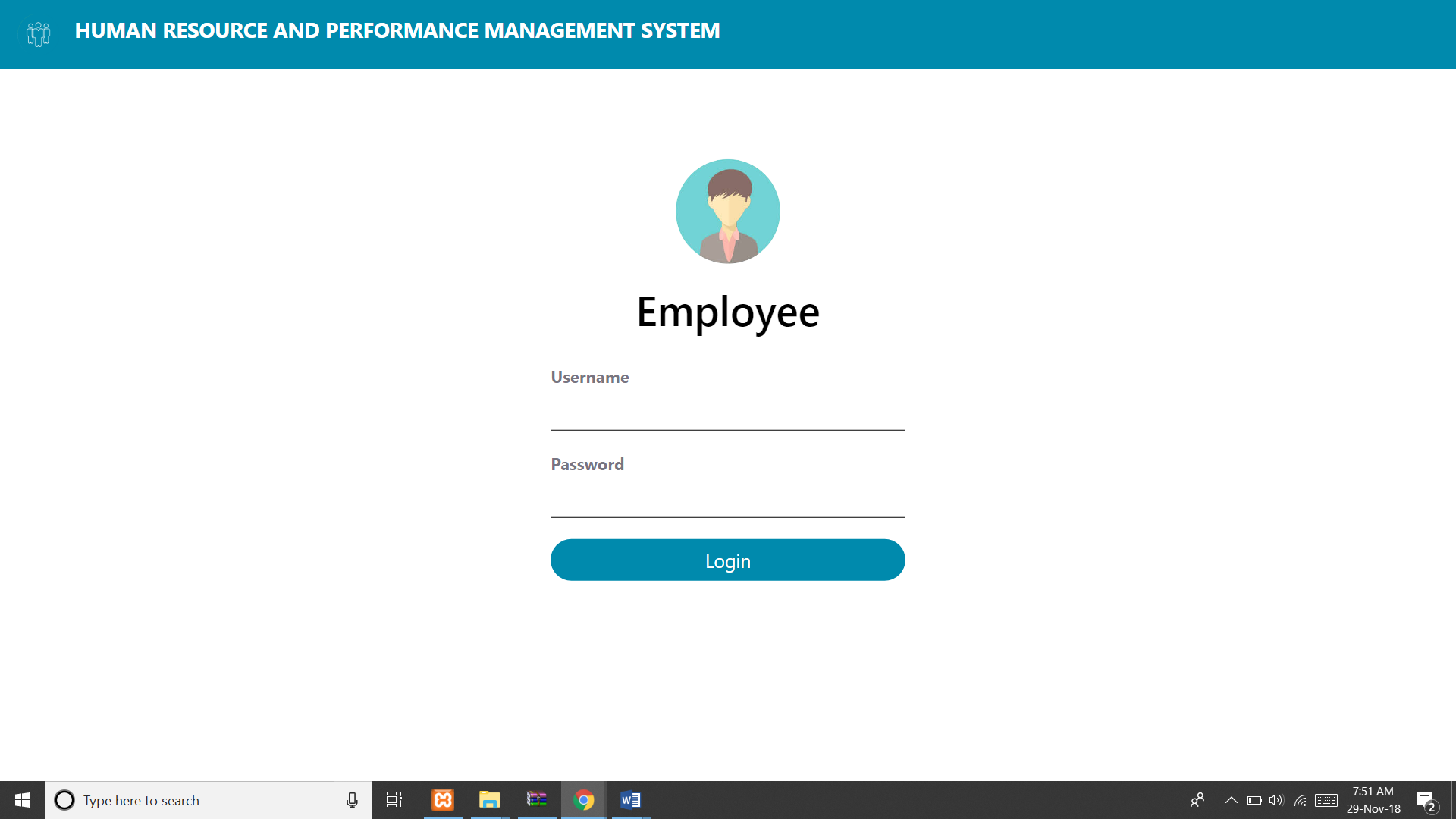
If the employee is chosen at the user page, we are greeted with employee login page On successful login, the employee gets the status of the project and can rate his colleagues and peers. 

Fig.7.16 Employee login page

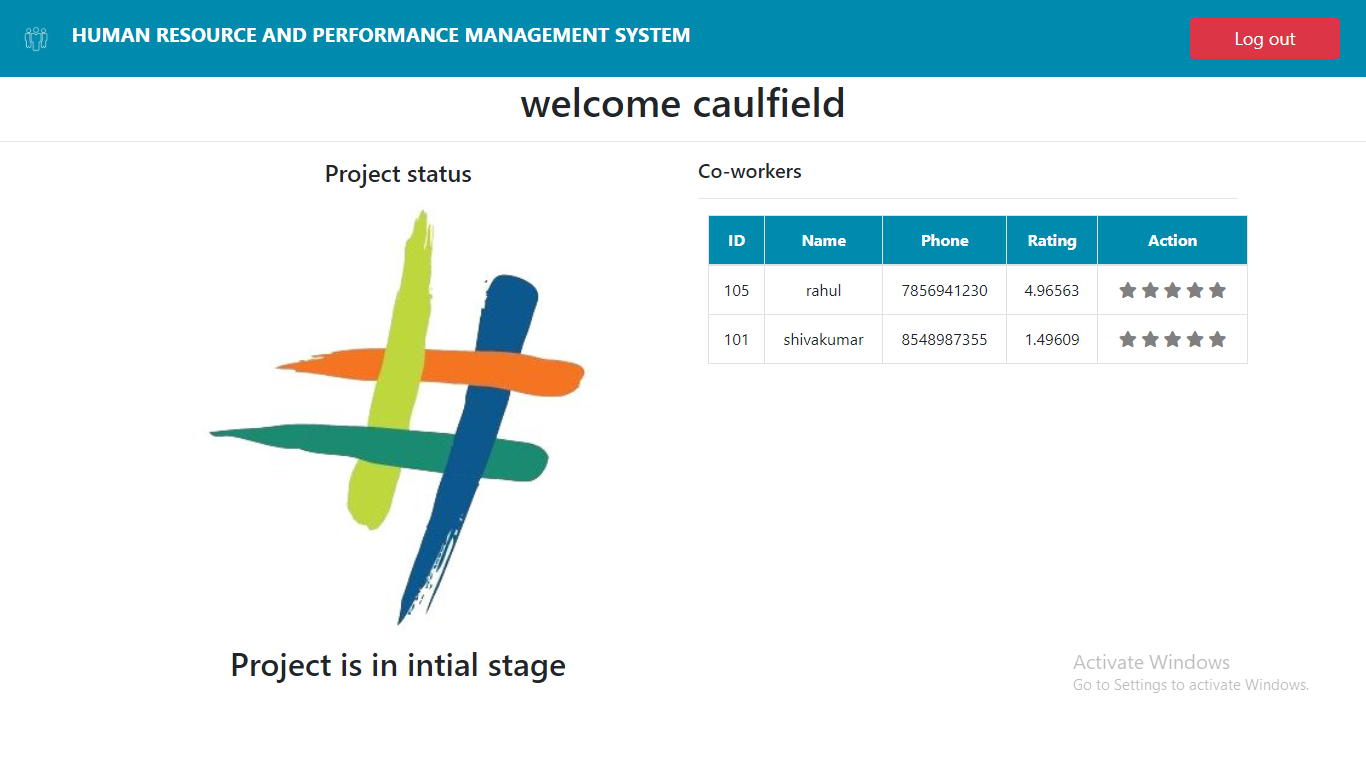


Fig.7.17 Employee Dashboard

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 .