**Chapter 1**

**Introduction**

HUMAN PAYROLL SYSTEM is the web technology project with backend in MySQL and front end code written using PHP and JavaScript. Through this project, we intend to bridge the gap between the lack of communication between employees in the company and their managers or between the employees themselves. Tracking the status of the projects, payrolls, etc. by multiple people can cause an ease in communication and the process of production can go on smoothly. Managers can track their employee’s progress and employees themselves can log their progress in so that it can be visible to the concerned people.

In the present scenario of industry where time is of the essence, people can not only rely on non-redundant techniques of progress tracking. A job that is unnecessarily repeated is a waste of time and man hours. In order to reduce this wastage, the application comes into picture. In this application, the project is added by the manager and then under that various tasks are created which are well defined. These well-defined tasks are then allocated to various employees. The employees know exactly what their role is in the production, and the manager can also track their progress and change the status of the tasks and the project as the production advances.

**1.1 Introduction to Web Technology**

Web technologies is a general term referring to the many languages and multimedia packages that are used in conjunction with one another, to produce dynamic web sites such as this one. Each separate technology is fairly limited on its own, and tends to require the dual use of at least one other such technology. Therefore, we can conclude that all of the components that make up a site are interdependent on one another [1].

HTML (Hyper Text Mark-up Language) is the glue that holds together every web site. Like building a house, you always build a strong foundation first. For any site, HTML is that foundation. HTML is an open source language (i.e. not owned by anyone), which is easy to learn, and requires no fancy (or expensive) packages to start using it. All you need is something to type with, such as Windows Notepad, and a lot of time and patience.

Although relatively limited by itself, it is the flexibility of HTML that allows web sites to grow in complexity. Like the foundation of your house, HTML is robust enough to support many kinds of languages integrated within your HTML pages.

DHTML (Dynamic HTML) is just as the name suggests, it adds dynamic, moving or changing content to your plain old HTML pages. Think of it as a more advanced version of HTML, although DTHML is in fact not a programming language in itself. DHTML is a broad term used to describe a group of applications, the main ones are described below:

**JavaScript:** JavaScript is a 'scripting' language. A bit like a script in a feature film, it is used to decide 'what happens next'. This may be a sequence of screen events, where one event is initiated by the end of another, or it could be a programmed response to a user interacting with the page in some way, e.g. moving their mouse over a link. JavaScript is a complex and powerful language, and may be placed directly inside a HTML page, or in a separate JavaScript file.

**CSS and CSS-P:** CSS (Cascading Style Sheets) is a relatively new language, designed to expand upon the limited style properties of HTML. Easy to learn and implement, CSS is an excellent way to control the style of your site, such as text styles like size, color and font.

CSS may also be placed inside the HTML page or in separate files. The real advantage of having all of the style properties for your entire site in one single CSS file is that you may edit that single file to effect changes on the whole site, rather than having to go through each HTML file one at a time. For this reason, it is perhaps the most useful web technology and certainly one of my favourites.CSS-P (CSS-Positioning) is a sub-set of CSSs, and is concerned mainly with the layout of your HTML pages. It allows the web designer to place any element (text, graphic etc.) exactly on the screen where they want it, to the pixel.

**DOM:** DOM (Document Object Model) allows the designer to access any element in a HTML page, such as an image, layer or table. Each element may be assigned a unique 'id' name to identify it. When combined with CSS and JavaScript, the DOM may be used to make changes to only "My Tag" and no other element, such as increasing text size or the position of "My Tag" on the screen. JavaScript can also be used to animated such changes to any identified element, such as gradually increasing the size of the text on screen. Each different browser has its own DOM, and this is often used to determine which browser the visitor is using. A particular action may be carried out if the person is using NS6, for example, and ignored if the person is using IE5.

**1.2 PHP**

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Redorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for *Personal Home Page*, but it now stands for the recursive acronym *PHP: Hypertext Pre-processor*. PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks.

PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command line interface (CLI) and can be used to implement standalone graphical applications. The standard PHP interpreter, powered by the Zend engine, is a free software released under the PHP License.

PHP has been widely ported and can be deployed on most web servers on almost every operating systems. The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

**1.3 JavaScript**

JavaScript is a programming language that started off simply as a mechanism to add logic and interactivity to an otherwise static Netscape browser. In the years since its introduction, it has not only supplanted a variety of other competing languages and technologies to become the standard for browser-based programming, but it has also expanded beyond the client space to become a dominant language on the server side, as well.

The Hyper Text Transport Protocol, or HTTP, was designed to do exactly what the name implies, which is to transport HTML text across a network for viewing. But the problem with HTML is that it is completely static, providing no capabilities to implement logic. Basic features we currently take for granted, such as input validation, autocorrect, progress bars or causing a graphic to change during a mouse rollover, would all be impossible for a browser to perform without some type of scripting capability. As a result, when the web first rose to prominence, every vendor in that space provided some mechanism to make HTML web pages more interactive.

Microsoft promoted a competitor to JavaScript named VBScript, along with plug-in technologies, such as ActiveX Data Object (ADO). Sun Microsystems promoted Java applets that were cross-platform and could be used in any browser.Many other technologies came and went as the internet matured. In the end, JavaScript won the battle for popularity, and it is the only client-side scripting technology supported by the Chrome, Firefox, Safari and Internet

JavaScript is an interpreted language, as opposed to compiled languages, such as C++ and Java. This means that the code written in JavaScript does not go through an intermediary compilation stage in which the source code is transformed into machine language that is easy for a CPU to process. Instead, JavaScript is interpreted on the fly by the computer processing it. Because JavaScript code is not packaged in a CPU-friendly form, such as Java bytecode or a binary executable, program execution can be slower than a comparable program written in Scala or F#. However, modern JavaScript engines are highly optimized, and inefficiencies are rarely noticeable when using modern hardware.

**1.4 XAMPP**

XAMPP is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server) [solution stack](https://en.wikipedia.org/wiki/Solution_stack) package developed by Apache Friends, consisting mainly of the [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server), [MariaDB](https://en.wikipedia.org/wiki/MariaDB" \o "MariaDB) [database](https://en.wikipedia.org/wiki/Database), and [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) for scripts written in the [PHP](https://en.wikipedia.org/wiki/PHP) and [Perl](https://en.wikipedia.org/wiki/Perl) [programming languages](https://en.wikipedia.org/wiki/Programming_language). Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a [WAMP](https://en.wikipedia.org/wiki/WAMP) or [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) stack can be installed quickly and simply on an operating system by a developer. WAMP is sometimes used as an abbreviated name for the software stack Windows, Apache, MySQL, and PHP. It is derived from LAMP which stands for Linux, Apache, MySQL, and PHP. As the name implies, while LAMP is used on Linux servers, WAMP is used on Windows servers. Because WordPress isn’t usually installed on Windows Servers, WAMP has become popular among developers as a method of installing WordPress on their personal computer.

The “A” in WAMP stands for Apache. Apache is server software that is used to serve web pages. Whenever someone types in your WordPress website’s URL, Apache is the software that “serves” your WordPress site. The “M” in WAMP stands for MySQL. MySQL is a database management system. Its job in the software stack is to store all of your website’s content, user profiles, comments, etc. The “P” in WAMP stands for PHP. PHP is the programming language that WordPress is written in. It is also the piece that holds the entire software stack together. It runs as a process in Apache and communicates with the MySQL database to dynamically build your webpages. WAMP software stack can be downloaded from wampserver project’s download. For Microsoft windows users, it comes in an easy installation package with a control panel.

**1.5 Web Application Pipeline Architecture**

The figure shown gives an abstract, high-level block diagram of how a web application processes data. In the diagram, commands enter from the left and proceed through what can be thought of as a processing pipeline. Some commands specify geometric objects to be drawn, and others control how the objects are handled during the various processing stages. Let’s now take a look at the separate stages

## 

**Figure 1.1:** Pipeline Architecture of Web Applications

The concept of the web application pipeline is what really sets it apart from general CPUs, although the idea of a pipeline is the same as that used by general purpose CPUs.  
The graphics pipeline is built in stages. Every stage is specialized in precisely one element of the rendering process. The main components of pipeline architecture include web browser, CDN, cloud storage and web app servers.

As shown in the Figure 1, rather than having all commands proceeds immediately through the pipeline, you can choose to accumulate some of them in a display list for processing at a later time. In a single project scenario, separation of concerns is achieved through the use of folders. The default template includes separate folders for MVC pattern responsibilities of Models, Views, and Controllers, as well as additional folders for Data and Services. In this arrangement, presentation details should be limited as much as possible to the Views folder, and data access implementation details should be limited to classes kept in the Data folder.

**1.6 Objectives of Project tracking**

The objectives of the application are as follows:

* Be systematically developed in a coordinated manner to facilitate interconnectivity.
* Recognize the needs of better employee & payroll management and distribution system.
* Improve effectiveness and efficiency of task resource utilization.
* Provide a rich user-friendly interface, for effective storage and retrieval of data.

**Chapter 2**

**Requirements Specification**

A computerized way of handling information about employee and project details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section. The specific requirements of the web application are stated as follows:

**2.1 Hardware Requirement**

The section of hardware configuration is an important task related to the software development insufficient random access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application

Processor : Intel PentiumT4200/ Intel Core Duo 2.0 GHz / more

RAM : Minimum RAM capacity

Hard disk : 80GB or more

Peripherals : Keyboard, Compatible mouse

Cache Memory : L2-1 MB

GPU : Intel HD Graphics

Monitor Resolution : 1024\*768 or 1336\*768 or 1280\*1024

|  |  |  |
| --- | --- | --- |
|  |  |  |

**2.2 Software Requirement**

A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system.

This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system. It allows the developer or analyst to understand the system, function to be carried out the performance level to be obtained and corresponding interfaces to be established.

Operating system : WINDOWS 10

Editor : Sublime text/Notepad ++

Programming language : PHP, JavaScript

Web browser : Internet Explorer or any web browser

Server : Apache on XAMPP

**Chapter 3**

**Function Descriptions**

PHP MySQL functions gives to access the MySQL database servers. PHP works with MySQL version 4.1.13 or newer.

The description of all the functions used in the program is given below:

Table 3.1: List of Functions

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Function & Description** | **PHP** |
| 1 | [**mysqli affected rows**](https://www.tutorialspoint.com/php/php_mysqli_affected_rows.htm)  It returns the number of affected rows in the previous SELECT, INSERT, UPDATE, REPLACE, or DELETE query. | 4 |
| 2 | [**mysqli client encoding**](https://www.tutorialspoint.com/php/php_mysqli_client_encoding.htm)  It is used to turn off or turn of auto-committing database modifications. | 4.3.0 |
| 3 | [**mysqli close**](https://www.tutorialspoint.com/php/php_function_mysqli_close.htm)  It is used to close MySQLi connection | 4 |
| 4 | [**mysqli connect**](https://www.tutorialspoint.com/php/php_function_mysqli_connect.htm)  It opens a connection to a MySQLi Server | 4 |
| 5 | [**mysqli create db**](https://www.tutorialspoint.com/php/php_mysqli_create_db.htm)  It is used to create MySQLi connection | 5 |
| 6 | [**mysqli change user**](https://www.tutorialspoint.com/php/php_mysqli_change_user.htm)  It is used to set the current database connection insted of specified data base connection. | 5 |
| 7 | [**mysqli character set name**](https://www.tutorialspoint.com/php/mysqli_character_set_name.htm)  It is returns default character set for the database connection | 5 |
| 8 | [**connect errno**](https://www.tutorialspoint.com/php/php_mysqli_connect_errno.htm)  It returns the error code from the last connection | 5 |
| 9 | [**data seek**](https://www.tutorialspoint.com/php/php_mysqli_data_seek.htm)  It is used to move internal result pointer. | 4 |
| 10 | [**mysqli debug**](https://www.tutorialspoint.com/php/php_function_mysqli_debug.htm)  It is used to performs debugging operations | 5 |
| 11 | [**dump debug info**](https://www.tutorialspoint.com/php/php_dump_debug_info.htm)  It is used dumps debugging info into the log | 5 |
| 12 | [**mysqli error list**](https://www.tutorialspoint.com/php/php_mysqli_error_list.htm)  It returns a list of errors from the last connection | 4 |
| 13 | [**mysqli\_error**](https://www.tutorialspoint.com/php/php_function_mysqli_error.htm)  It returns the last error description for the most recent function | 4 |
| 14 | [**mysqli fetch all**](https://www.tutorialspoint.com/php/php_mysqli_fetch_all.htm)  It is used to fetchs all result rows and returns the result set as an associative array | 4 |
| 15 | [**mysqli fetch array**](https://www.tutorialspoint.com/php/php_mysqli_fetch_array.htm)  It is used to fetchs a result row as an associative array | 5 |
| 16 | [**mysqli fetch assoc**](https://www.tutorialspoint.com/php/php_mysqli_fetch_assoc.htm)  It is used to fetches a result row as an associative array. | 5.3 |
| 17 | [**mysqli fetch field direct**](https://www.tutorialspoint.com/php/mysqli_fetch_field_direct.htm)  It is used to returns the next column in the result set as an object. | 4 |
| 18 | [**mysqli fetch fields**](https://www.tutorialspoint.com/php/php_mysqli_fetch_fields.htm)  It is used to returns an array of objects | 4 |
| 19 | [**mysqli fetch lengths**](https://www.tutorialspoint.com/php/php_mysqli_fetch_lengths.htm)  It is used to returns the length of the fields in the result | 4 |
| 20 | [**mysqli fetch object**](https://www.tutorialspoint.com/php/php_mysqli_fetch_object.htm)  It returns an object | 5 |
| 21 | [**mysqli field count**](https://www.tutorialspoint.com/php/php_mysqli_field_count.htm)  It returns the number of columns for the most recent query. | 5 |
| 22 | [**mysqli field seek**](https://www.tutorialspoint.com/php/php_mysqli_field_seek.htm)  This function sets the column cursor to the given column offset. | 5 |
| 23 | [**mysqli field tell**](https://www.tutorialspoint.com/php/php_mysqli_field_tell.htm)  It returns returns the position of the field cursor. | 5 |
| 24 | [**mysqli\_free\_result**](https://www.tutorialspoint.com/php/php_mysqli_free_result.htm)  It frees the memory associated with the result. | 5 |
| 25 | [**mysqli get charset**](https://www.tutorialspoint.com/php/php_mysqli_get_charset.htm)  It returns a character set object. | 4 |
| 26 | [**mysqli get client info**](https://www.tutorialspoint.com/php/mysqli_get_client_info.htm)  It returns the MySQL client library version. | 5 |
| 27 | [**mysqli get client stats**](https://www.tutorialspoint.com/php/mysqli_get_client_stats.htm)  It returns statistics about client per-process. | 5 |
| 28 | [**mysqli get client version**](https://www.tutorialspoint.com/php/mysqli_get_client_version.htm)  It returns the MySQLi client library version. | 5.1.0 |
| 29 | [**mysqli get connection stats**](https://www.tutorialspoint.com/php/mysqli_get_connection_stats.htm)  It returns statistics about the client connection. | 5 |
| 30 | [**mysqli get host info**](https://www.tutorialspoint.com/php/mysqli_get_host_info.htm)  It returns the MySQLi server hostname and the connection type. | 5 |
| 31 | [**mysqli get proto info**](https://www.tutorialspoint.com/php/mysqli_get_proto_info.htm)  It returns the MySQLi protocol version information | 5 |
| 32 | [**mysqli get server info**](https://www.tutorialspoint.com/php/mysqli_get_server_info.htm)  It returns the MySQLi server information. | 5 |
| 33 | [**mysqli info**](https://www.tutorialspoint.com/php/php_function_mysqli_info.htm)  It returns information about the most recently executed query. | 5 |
| 34 | [**mysqli init**](https://www.tutorialspoint.com/php/php_function_mysqli_init.htm)  It returns an object to use with the mysqli\_real\_connect() function. | 5 |
| 35 | [**mysqli insert id**](https://www.tutorialspoint.com/php/php_mysqli_insert_id.htm)  It returns an id of last query. | 5 |
| 36 | [**mysqli kill**](https://www.tutorialspoint.com/php/php_function_mysqli_kill.htm)  This function asks to the server to kill MySQLi thread specified by the processid parameter. | 5 |
| 37 | [**mysqli more results**](https://www.tutorialspoint.com/php/php_mysqli_more_results.htm)  This function checks if there are more results from a multi query. | 5 |
| 38 | [**mysqli multi query**](https://www.tutorialspoint.com/php/php_mysqli_multi_query.htm)  It used to separate the queries with a semicolon against the database. | 5 |
| 39 | [**mysqli next result**](https://www.tutorialspoint.com/php/php_mysqli_next_result.htm)  It prepares the next result set from mysqli\_multi\_query(). | 5 |
| 40 | [**mysqli num fields**](https://www.tutorialspoint.com/php/php_mysqli_num_fields.htm)  It returns the number of fields in a result set | 5 |
| 41 | [**mysqli num rows**](https://www.tutorialspoint.com/php/php_mysqli_num_rows.htm)  It returns the number of rows in a result set | 5 |
| 42 | [**mysqli options**](https://www.tutorialspoint.com/php/php_function_mysqli_options.htm)  It is used to sets connect options and change connection settings. | 5 |
| 43 | [**mysqli ping**](https://www.tutorialspoint.com/php/php_function_mysqli_ping.htm)  It is used to pings a server connection and reconnect to server if connection is lost. | 5 |
| 44 | [**mysqli query**](https://www.tutorialspoint.com/php/php_function_mysqli_query.htm)  It performs a query against the database. | 5 |
| 45 | [**mysqli query**](https://www.tutorialspoint.com/php/php_function_mysqli_query.htm)  It performs a query against the database. | 5 |
| 46 | [**mysqli real connect**](https://www.tutorialspoint.com/php/php_mysqli_real_connect.htm)  This function opens a new connection to the MySQLi | 5 |
| 47 | [**mysqli real escape string**](https://www.tutorialspoint.com/php/mysqli_real_escape_string.htm)  This function escapes special characters in a string for an SQL statement. | 5 |
| 48 | [**mysqli refresh**](https://www.tutorialspoint.com/php/php_function_mysqli_refresh.htm)  This function refreshes tables or caches, or resets the replication server information. | 5 |
| 49 | [**mysqli rollback**](https://www.tutorialspoint.com/php/php_function_mysqli_rollback.htm)  This function rolls back the current transaction for the specified database connection. | 5 |
| 50 | [**mysqli select db**](https://www.tutorialspoint.com/php/php_mysqli_select_db.htm)  This function changes the default database. | 5 |

**Chapter 4**

**System Design and Implementation**

**4.1 Introduction**

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.

This project is developed using PHP as a host language (the front-end) and with MYSQL as the database (back-end), under Windows 7/8/10 platform. We can use a wide range of functions (as seen in the previous chapter) to establish a connection with the database from the host language and also to execute the required queries. The interface to the program is provided by the input devices mouse and keyboard.

The program also uses the concept of embedded SQL (SQL statements present in the program itself, rather than executing them from the back end separately). This optimizes the executing time and hence the no. of codes are also reduced.

**4.2 Overall Design Process**

There are only a handful functions from Chapter 2 used for this project. Using the mouse and keyboard, we can select, deselect and enter data as per our convenience.

**4.3 Attributes of Entities/Tables**

A table is a collection of related data held in a structured format within a database. It consists of columns, and rows. In relational databases, and flat file databases, a table is a set of data elements (values) using a model of vertical columns (identifiable by name) and horizontal rows, the cell being the unit where a row and column intersect. A table has a specified number of columns, but can have any number of rows. Each row is identified by one or more values appearing in a particular column subset. The columns subset which uniquely identifies a row is called the primary key. "Table" is another term for "relation"; although there is the difference in that a table is usually a multi set (bag) of rows where a relation is a set and does not allow duplicates.

The below diagram represents the tables and their receptive attributes present in the application:

|  |  |
| --- | --- |
| **Table** | **Attributes** |
| admin | id, username, password, code |
| registration | Fname, Lname, DOB, Gender, Address, Phno, email,Pin, Uname, Password |
| salary | Uname, Designation, Salary, file |
| project | Uname, PrTitle, PrDetails |
| contact | Name, email, message |

**Table 4.1**: Attribute Table

**4.4 Pseudo codes**

**4.4.1 Algorithm for Employee Registration**

1. **BEGIN**
2. Enter the necessary details such as first name, last name, address, phone number, email-id, username and password
3. Verify if all the mandatory fields are entered.
4. If all the fields are entered, then store the entered data in the EMPLOYEE table proceed to the login page.

Else display the message “Enter all the mandatory fields”

1. **END**

**4.4.2 Algorithm for Viewing Visitor Enquiries**

1. **BEGIN**
2. All the queries and the regarding visitor’s details will be displayed on clicking this tab.
3. **END**

**4.4.3 Algorithm for Employee Login**

1. **BEGIN**
2. Enter the username and password
3. Verify the credentials of the entered data with that of the login table
4. If all the fields are entered, then proceed to the employee login page.
5. Else display the message “Incorrect username or password”.
6. **END**

**4.4.4 Algorithm for Administrator Login**

1. **BEGIN**
2. Enter the username and password
3. Verify the credentials of the entered data with that of the login table
4. If all the fields are entered, then proceed to the login page.
5. Else display the message “Incorrect username or password”
6. **END**

**4.4.5 Algorithm for Adding Salary Details**

1. **BEGIN**
2. Add the paycheck details and upload the paycheck softcopy for the employee to view.
3. **END**

**4.4.6 Algorithm for Add New Projects**

1. **BEGIN**
2. Enter the necessary details for adding a new project such as project id, name etc.
3. Store the entered data in the **PROJECT** table.
4. **END**

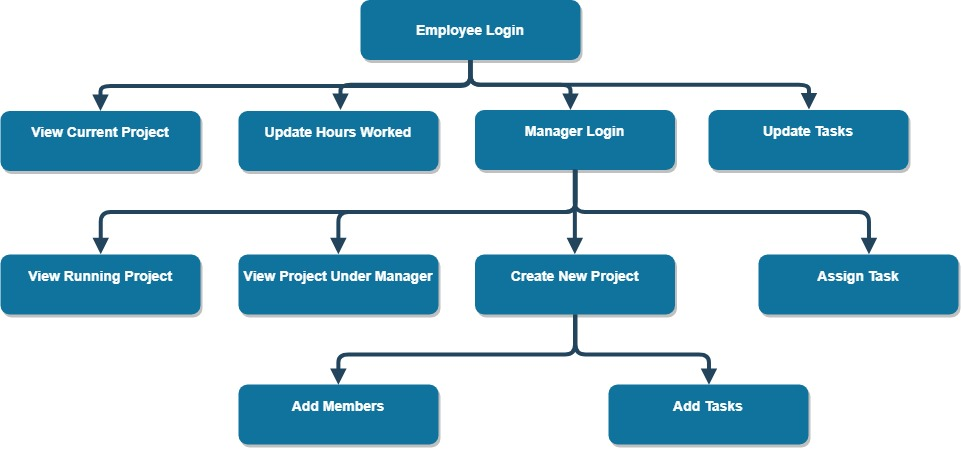
**4.4.7 Algorithm for Viewing the List of Employees**

1. **BEGIN**
2. Once a new employee is registered, display the list of registered employees in a table format.
3. **END**

**4.4.8 Algorithm for Viewing All Running Projects**

1. **BEGIN**
2. On clicking on the option “view all projects”, all the projects the company is working on will be displayed
3. By entering the specific project id, project progress can also be viewed.
4. **END**

**4.5 Flowchart**

The below figure represents the schema of all relations of the Project Tracking and Management Web Application System.

View Visitor Enquiries

Adding Details

View Employee Details

View Salary Details

Admin Login

**Figure 4.1**: Flowchart of the system.

Upload Softcopy of Pay check

Add the Salary Details

Raise Queries

Add Salary Details

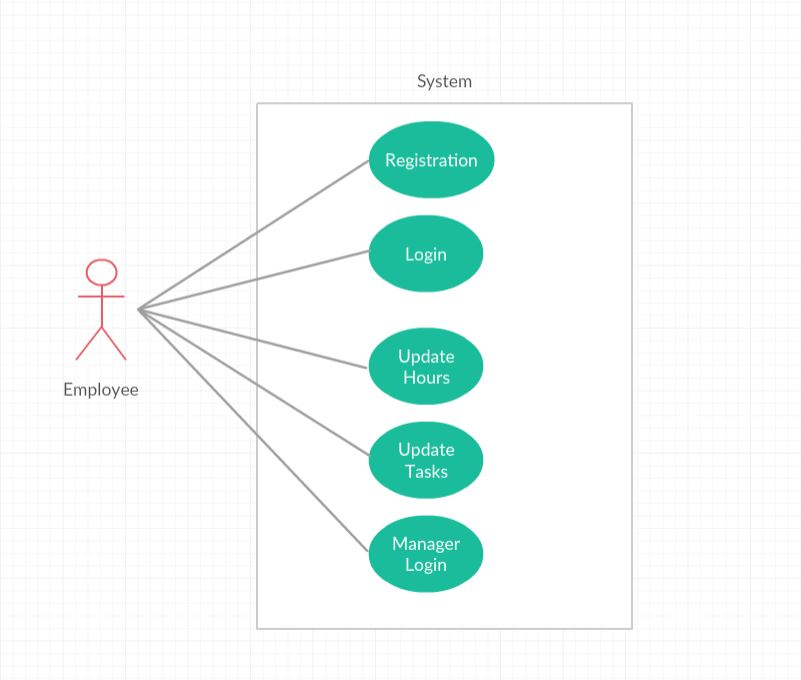
Add Project Details

Add Employee Personal Details

**4.6 Design**

**4.6.1 Use case of Employee**

The below figure depicts the modules for the actor ‘Employee’ using use case diagram.



Login

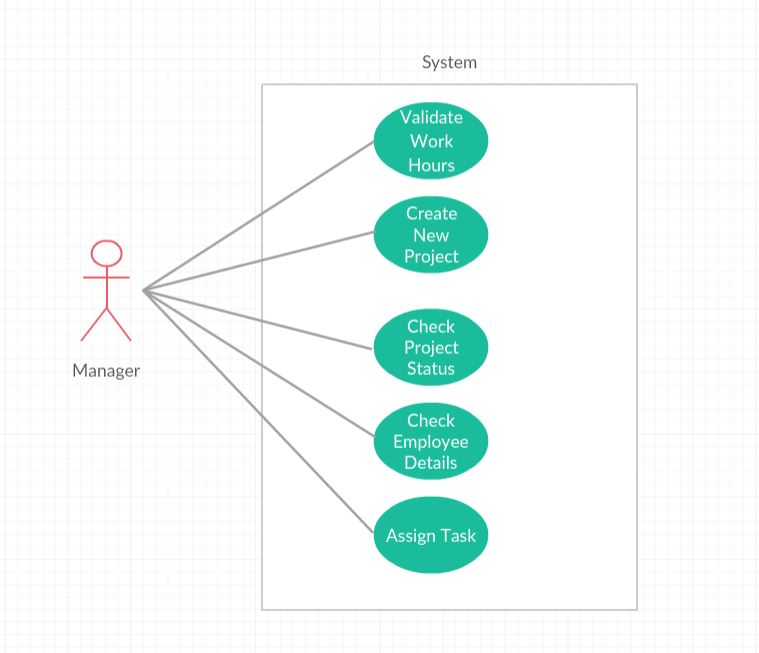
View All Details

Update Personal Information

**Figure 4.2**: Use case of the employee.

**4.6.2 Use case of Managers**

The below figure depicts the modules for the actor ‘Managers’ using use case diagram.



Login

Add Employee

View All Details

Add Project

Add Salary

**Figure 4.3**: Use case of the Manager

**4.7 Test Cases**

**4.7.1 Test case for Employee Login**

Table 4.2: UTC1 for employee Login

|  |  |  |
| --- | --- | --- |
| **Inputs** | **Testcase** | **Possible Outputs** |
| Username | Valid username | Log-in successful |
|  | Invalid username | Log-in fail, error message to prompt |
| Password | Valid password | Log-in successful |
|  | Invalid password | Log-in fail, error message to prompt |

**4.7.2 Test case for Administrator Login**

Table 4.3: UTC2 for Manager Login

|  |  |  |
| --- | --- | --- |
| **Inputs** | **Testcase** | **Possible Outputs** |
| Username | Valid username | Log-in successful |
|  | Invalid username | Log-in fail, error message to prompt |
| Password | Valid password | Log-in successful |
|  | Invalid password | Log-in fail, error message to prompt |

**4.7.3 Test case for Employee registration**

Table 4.4: UTC3 for Employee registration

|  |  |  |
| --- | --- | --- |
| **Inputs** | **Testcase** | **Possible Outputs** |
| Name | Valid name | Accept |
|  | Invalid name | Error message to prompt |
| Emp. ID | Valid emp-id | Accept |
|  | Invalid emp-id | Error message to prompt |
| Password | Valid password | Accept |
|  | Invalid password | Error message to prompt |

**4.7.4 Test case for Salary Details**

Table 4.5: UTC4 for hours worked update

|  |  |  |
| --- | --- | --- |
| **Inputs** | **Testcase** | **Possible Outputs** |
| Username | Valid Username | Accept |
|  | Invalid Username | Error message to prompt |
| Designation | Valid Designation | Accept |
|  | Invalid Designation | Error message to prompt |
| Salary | Valid Salary | Accept |
|  | Invalid Salary | Error message to prompt |

**4.7.5 Test case for creating new Projects**

Table 4.6: UTC5 for new project creation

|  |  |  |
| --- | --- | --- |
| **Inputs** | **Testcase** | **Possible Outputs** |
| Username | Valid Username | Accept |
|  | Invalid Username | Error message to prompt |
| Project Title | Valid Text | Accept |
|  | Invalid Text | Error message to prompt |
| Project Description | Valid Text | Accept |
|  | Invalid Text | Error message to prompt |

**Chapter 5**

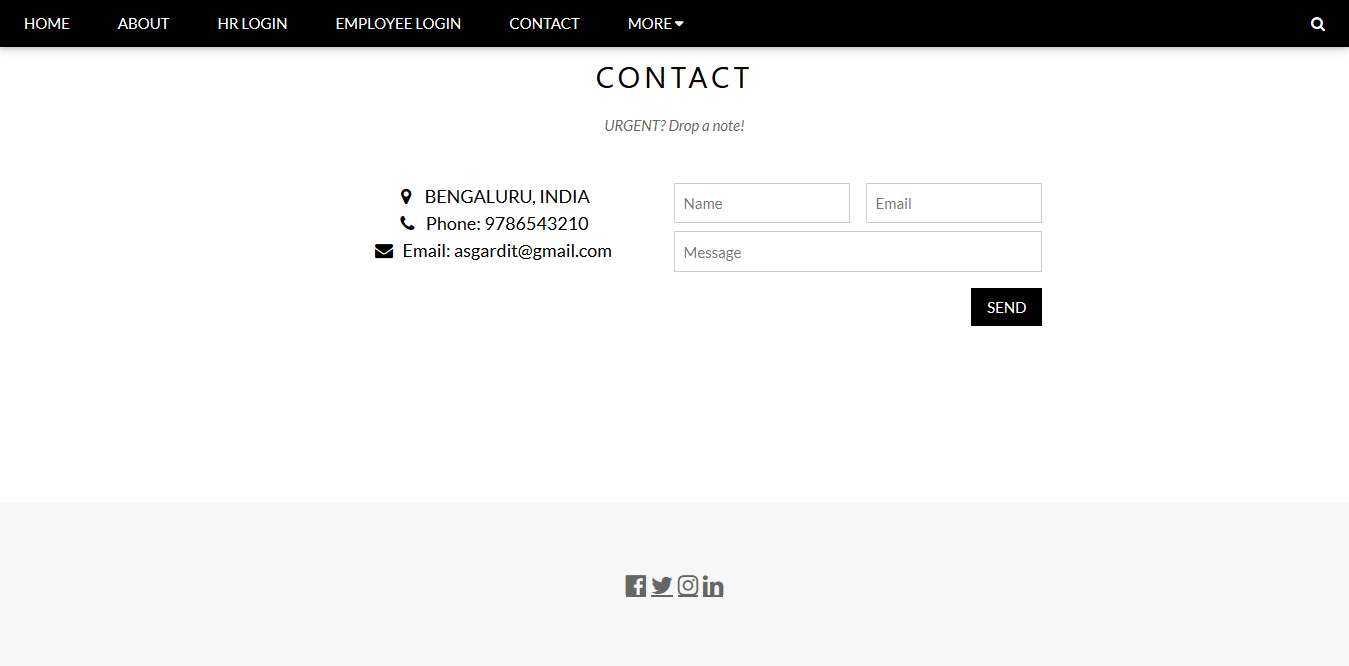
**Results and Discussions**

The following screenshot displays the initial window of web application.



**Figure 5.1**: Initial Window

The following screenshot displays the Contact Section for the Visitors.

.

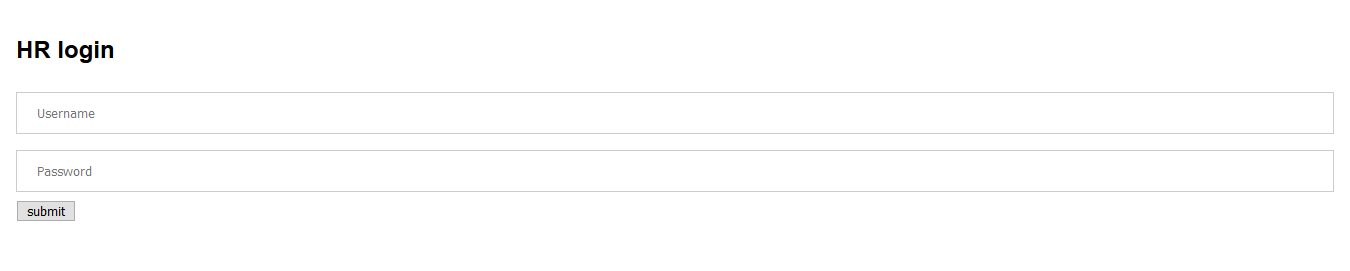
**Figure 5.2**: Contact

The following screenshot displays the Information about the Organization.



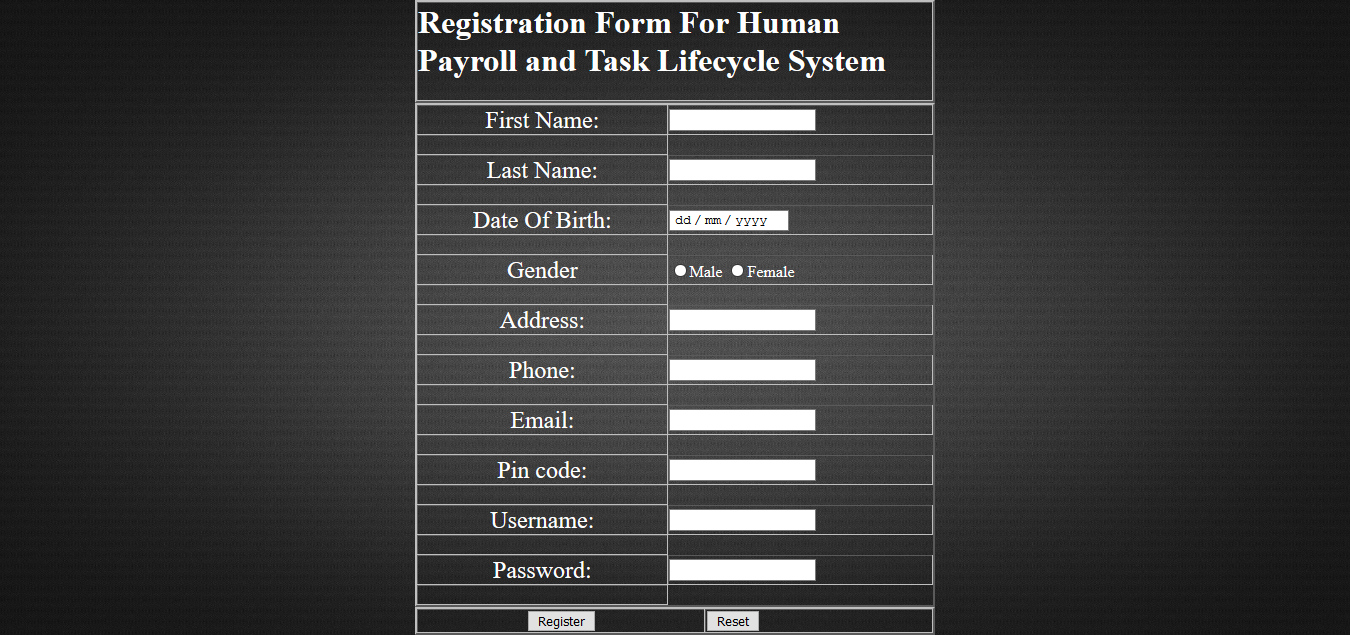
**Figure 5.3**: About

The following screenshot displays the HR Login Page



**Figure 5.4**: HR Login

The following screenshot displays the employee registration page



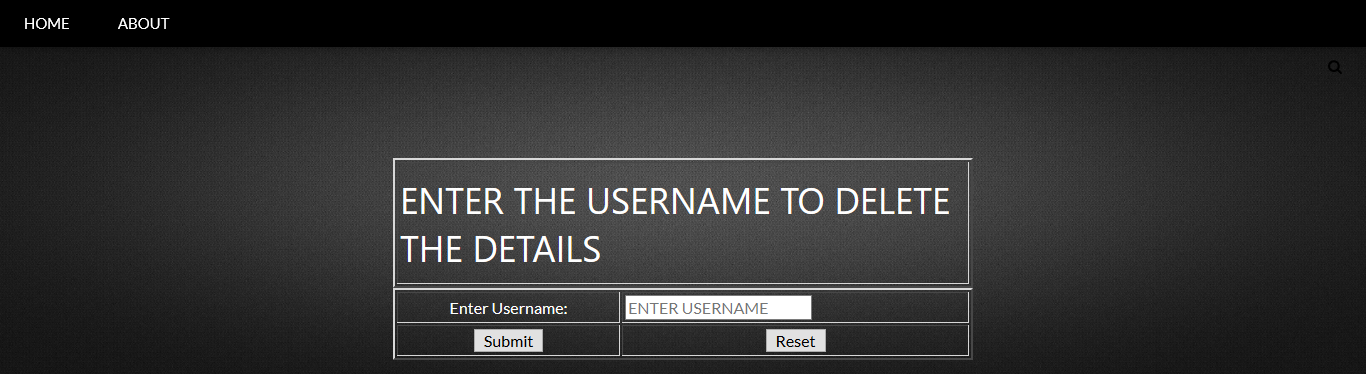
**Figure 5.5**: Employee Registration

The following screenshot displays the Enquiries posted by visitors page

****

**Figure 5.6**: Visitor Enquiries

The following screenshot displays the Employee deletion page



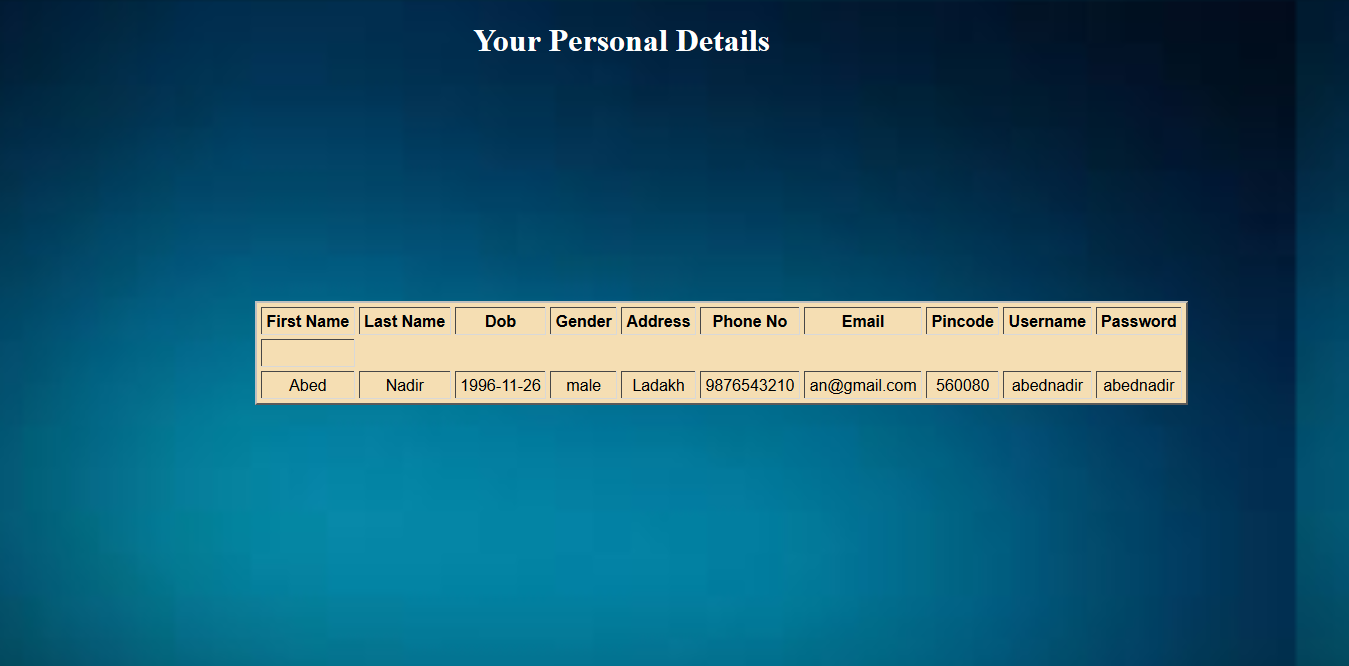
**Figure 5.7**: Employee Deletion Page

The following screenshot displays the Employee details page



**Figure 5.8**: Employee Details

The following screenshot displays the Employee Personal Information page.



**Figure 5.9**: Personal Information

The following screenshot displays the employee salary page along with the pay check softcopy



**Figure 5.10**: Employee Salary

**Chapter 6**

**Conclusion and Future Scope**

**6.1 Conclusion**

The Human Payroll system is a useful tool for businesses across the world. This project provides a solution to the various management problems a business would face when working on projects. By creating a common interface which can be accessed by both the employees and managers, the management of various projects is made easier. As the front-end is simple and easy to use, the project can help all types of businesses. However, the future improvement is necessary and continuous modification is needed to approach the final success:

* 1. The stronger authentication is required.
  2. A more user-friendly interface should be provided
  3. The system should be web-based.

**6.2 Future Enhancement**

This project has tremendous potential to gain wide acceptance among businesses. Poor management of projects and tasks can lead to various problems among the employees and can lead to the company’s downfall.

Through this project we intend to help businesses have a user-friendly tool to manage and delegate tasks towards the completion of their projects. This will contribute to the economy, thereby providing more jobs and improving the standard of living.

The future scope of our project is vast and can be used in extensive ways:

* The project can be enhanced to include notification pushing for employees.
* Functionality can be added to categorize the projects given in the details.
* It can be deployed in various cities and towns.
* An additional functionality that allows the clients to update their requirements.

**References**

1. https://en.wikibooks.org/wiki/Introduction\_to\_Information\_Technology/Web\_Technologies
2. http://stackoverflow.com/web-application-DevOp.
3. <http://github.com/mysql>
4. <https://en.wikipedia.org/wiki/HTML>
5. <https://www.w3schools.com>
6. https://www.youtube.com