Software Requirements Specification

For

Employee Management System

Version 1.0

Prepared by PseudoMavericks

03/23/2022

Table of Contents

1. Functional Requirements for Phase-1 4

1.1 Login 4

1.2 Registration Page 4

1.3 Dashboard/ Home Page 4

1.4 Database 4

2. UML Designn for Phase-1 5

2.1 Class Diagram 6

2.2 UseCase Diagram(Normal Case) 6

2.3 UseCase Diagram(Exception Case) 7

2.4 Sequence Diagram 7

3. Test Cases for Phase-1 8

4. User Manual 9

4.1 Login Page 9

4.2 Dashboard/ Home Page 9

4.3 Registration Page 9

5. How to Compile/Run the Application 9

5.1 List of instructions how to run and minimum requirements 9

5.2 Clear instructions on how to compile/run the program and testcases 9

6. The Peer review session………………………………………………………………….……………………………………9

7. A brief reflection on what has been accomplished, what went well and could be improved………………………………………………………………………….……………9

8. Member Contribution Table.………………………………………………………………10

1. **Functional Requirements for phase 1**
   1. **Log in:**

To begin with employees should be able to login to the application with their username and password. Login is a simple user-friendly web page designed using PHP, HTML and CSS which has two fields: username and password. Employees should enter their credentials in the respective fields. Then they are validated, if they are accurate then the user is redirected to dashboard else an error message pops up.

* 1. **Registration Page:**

Registration page is used to register the employees into the database. HR and team will be having these privileges to register the employees into the employee management system portal. They must fill up the registration form with employees’ details such as first-name, last-name, email address etc. Once the data is stored in database then employees will be able to login into the application with their credentials.

* 1. **Dashboard/ Home Page:**

Once the user is logged in, the user is redirected to the dashboard/ home page. This consists of a menu with links that redirects to other pages in the system such as payroll, billing hours etc. A logout button is given on top left which enables the user to logout from the system.

* 1. **Databases:**

In the backend, there are 6 tables created one for employees, for HR, billing hours, payroll details, projects, employee project details.

1. **Test Cases For Phase-1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | **Test Scenario Description** | **Test Case ID** | **Test Case Description** | **Test Steps** | **Preconditions** | **Post Conditions** | **Expected Result** | **Comments(if any)** |
| TS\_EMS  \_001 | Verify the Registration functionality of Employee page | TC\_EMS  \_Reg\_  001 | Enter new employee first name,Last name,email and password | Enter new Employees First Name,Last Name,Email and password  2.Click on register button | 1.Valid URL  2.Test Data  3.User must not have registered earlier | User must be able to register successfully | Successful registration |  |
| TS\_EMS  \_001 | Verify the Registration functionality of Employee page | TC\_EMS  \_Reg\_  002 | Enter already existing employees first name,last name,email id and password | 1.Enter existing Employees First Name,Last Name,Email and password  2.Click on register button | 1.Valid URL  2.Test Data  3.User must have registered earlier | User must see a message “User already exists” | A popup message box to show an error “User already exists” |  |
| TS\_EMS  \_002 | Verify the login functionality of Employee page | TC\_EMS  \_login\_  001 | Enter a valid Email  & password | 1.Enter valid Email  2.Enter valid password  3.Click on login button | 1.Valid URL  2.Test Data | User should able to see the homepage | Successful login |  |
| TS\_EMS  \_002 | Verify the login functionality of Employee page | TC\_EMS  \_login\_  002 | Enter a valid Email  & invalid password | 1.Enter valid Email  2.Enter invalid password  3.Click on login button | 1.Valid URL  2.Test Data | Error message  ”Invalid Email  or password” | A popup message box to show an error “Invalid Email  or password” |  |
| TS\_EMS  \_002 | Verify the login functionality of Employee page | TC\_EMS  \_login\_  003 | Enter an invalid Email  & valid password | 1.Enter invalid Email  2.Enter valid password  3.Click on login button | 1.Valid URL  2.Test Data | Error message  ”Invalid Email  or password” | A popup message box to show an error “Invalid Email  or password” |  |
| TS\_EMS  \_002 | Verify the login functionality of Employee page | TC\_EMS  \_login\_  004 | Enter an invalid Email  & invalid password | 1.Enter invalid Email  2.Enter invalid password  3.Click on login button | 1.Valid URL  2.Test Data | Error message  ”Invalid Email  or password” | A popup message box to show an error “Invalid Email  or password” |  |
| TS\_EMS  \_003 |  |  |  |  |  |  |  |  |

**4. User Manual:**

* 1. **Login Page:**

When user visits the application, it lands on the login page. The user will enter their credentials to login into the application in two fields: username and password. After entering their credentials user will be able to login into the application if the credentials entered are valid or else an error message pops up. It redirects to dashboard page once logged in.

Graphical user interface, application, Teams

Description automatically generated

Fig 1. Login Page snap of live website

* 1. **Dashboard/ Home Page:**

Once logged into the application the user will be redirected to dashboard page. It is the home page of the application with project title “Employee Management System” on top right. Here the user can find a menu on the right end of the page which contains different menu options like payroll, leave requests and billing hours. When clicked on any one of the links it redirects to their respective page. It also has a logout button on top left which enables the user to logout of the application.

Graphical user interface, text, application, Teams

Description automatically generated

Fig 2. Dashboard of employee after login

* 1. **Registration Page:**

The registration page is only accessible to the HR and team of the company. They have the privilege to register the employees into the Employee Management System portal. After registering the employees, they will be able to login into the system with their credentials. In registration page there are a couple of fields where the details of the employee should be entered such as first name, last name, email Id and password.

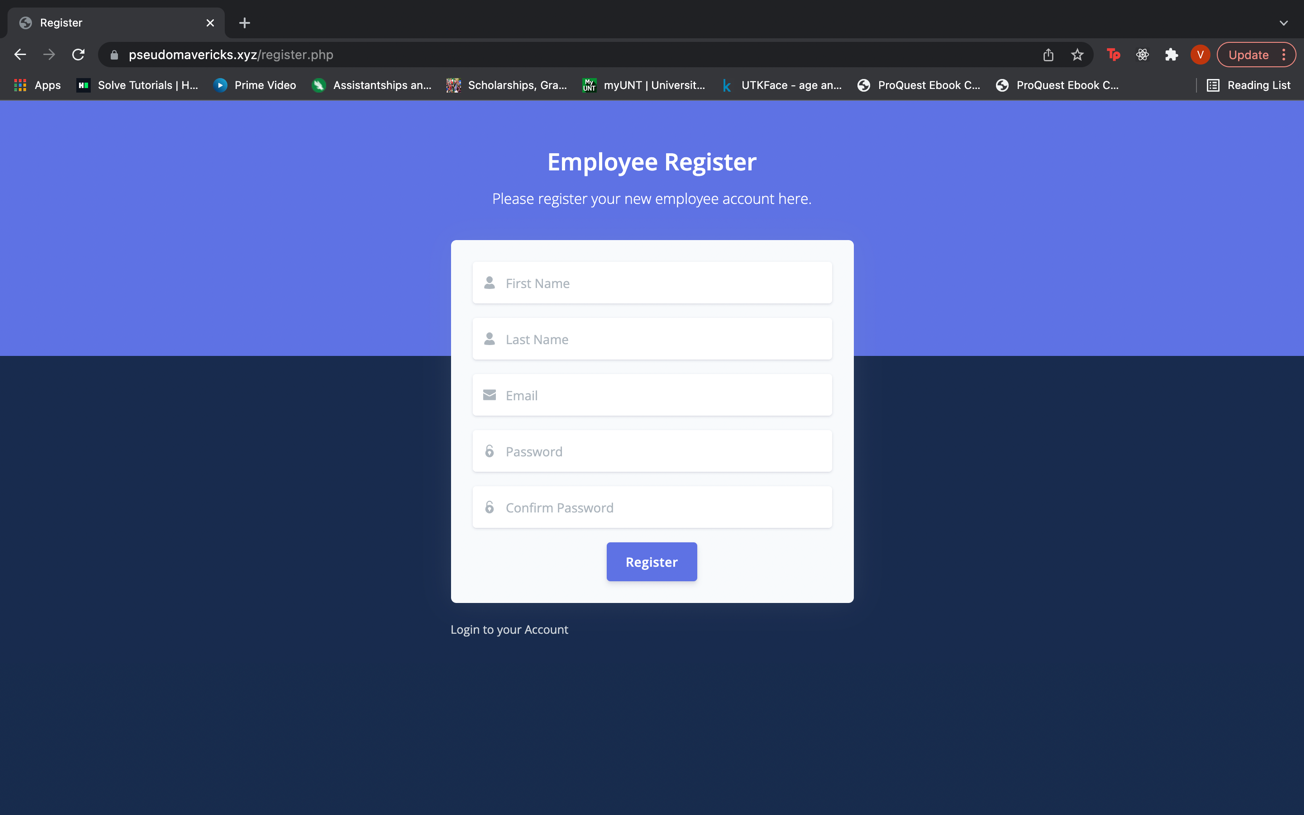


Fig 3. Employee registration page

**5. How to Compile/Run the Application**

**5.1 The list of instructions to run the project in their local/computer and the minimum requirements**

1. System specifications

* Minimum of intel 5th generation or higher.
* RAM: minimum of 4 GB.
* Around 5GB of hard disk storage.

1. To construct the application, the user must first install an IDE (Integrated Development Environment). We prefer Eclipse or NetBeans.
2. Import the project repository into the IDE and configure the PHP connection.
3. On the local PC, install the PHP libraries.
4. Additionally, they must set up a XAMPP server on their PC.
5. After installation, they need to start the Apache server, MySQL database admin control panels using XAMPP.
6. After launching, it will navigate the phpMyAdmin console page, where we need to create a database according to the connection file in the project.
7. After the database has been created, we need to import the SQL file to the database from the appropriate folder location.
8. Then the user can access the application using their localhost URL.

**5.2 Clear instructions on how to compile/run both application and test classes.**

1. Download the project zip file.

Can use GitHub to clone the repository using the below URL.

<https://github.com/amarendhar-reddy/project-pseudo-mavericks.git>

1. Extract the zip file.
2. Place the extracted files in XAMPP's htdocs folder.
3. In the IDE, create a workspace and import the project folder into it.
4. Configure the PHP to the workspace using the instructions below.
   1. On the toolbar, go to the preferences area beneath the window section.
   2. Select the Installed PHPs under the PHP section.
   3. Add the name and PHP executable file.

Graphical user interface, table

Description automatically generated

Fig 5. Configuring for php to run

1. Create a database in phpMyAdmin with the name “employee\_ms”.
2. In the database, import the "employee\_ms.sql" file.
3. To start as admin, set up the start page in the IDE's run configurations.
   1. /employee\_ms/login.php

Graphical user interface, application

Description automatically generated

Fig 6. How to run the application in eclipse IDE

1. Run the application and access the application using the localhost URL.
   1. Ex: [http://localhost/employee\_ms/login.php#](http://localhost/employee_ms/login.php)!
2. To run the test cases, we need to configure the PHPUnit in the IDE.
   1. Navigate to the run configurations, Add the PHPUnit properties.
   2. After adding it, run the application as a PHP unit test.

Graphical user interface, text, application, email

Description automatically generated

Fig 7. To run the test suite in the project

**6. The Peer review session**

1. During the peer review session, our team went through the features and code base and suggested on adding a new feature where we will save the customer login status in the browser session. With the login status being stored in browser session we can enhance security to the user information by redirecting to login page if someone alters the login status or Man-in-the-Browser attack. We have discussed and decided to include this feature in our project.
2. The team also suggested on adding an extra feature where we can send out a OTP to mobile or email when customers are logging in as a extra security measurement. Since our main objective is to create a basic EMS, we decided on not including the feature for now.

**7. A brief reflection on what has been accomplished, what went well and could be improved**

So far, we have created a working login screen, Employee registration screen and a functional logout feature and base skeleton of the project and planning to add new features in near future. We are planning to enhance the look and feel of the UI screens and also adding an additional security feature.

**8. Member Contribution Table**

|  |  |  |
| --- | --- | --- |
| **Member Name** | **Contribution description** | **Overall contribution** |
| Amarendhar Reddy Chevula | Functional Requirements for Phase-1, updating meeting miniutes | 12.5% |
| Subbarao Sanka | How to Compile/Run the Application | 12.5% |
| Vaishnavi Choppalli | User Manual | 12.5% |
| Kalyani Virivinti | Use Case Diagrams and Database design | 12.5% |
| Narasimharao Gurram | Functional Requirements for Phase-1 | 12.5% |
| Aparna Mirapakayala | Class Diagram and Sequence Diagram | 12.5% |
| Kavya Keerthi Tumarada | A brief reflection on what has been accomplished, what went well and could be improved | 12.5% |
| Neha Chowdary Puvvada | Test Cases for Phase-1 | 12.5% |