

School: Computer Science
Institution: University of Windsor
Term: Fall 2024

Course: Comp 4150: Advanced and Practical Database Systems

Instructor: Dr. Ezeife, C. I.

Project Phase: 3: Total: **100 marks**

Due date: Project Phase 3 (Given: Oct. 7, 2024 Due: Nov. 25, 2024)

Objective of Phase 3: Putting it all together; which involves implementing your database application using SQL, and a front-end database application tool (preferably web-based) (e.g. PHP) exploiting advanced features. You have so far developed many parts of this application in project phases 1 and 2. You are now required to put things together through a nice Web based user interface application such as your UWinsitestudent or the interface at www.wevax.ca. A sample such application for the company database system developed by one of my previous students, Sergio Oliviero can be accessed through the link: <https://oliveirs.myweb.cs.uwindsor.ca/complex-company/>

You can recreate a similar application as much as possible to include the features requested by this project phase. Keep learning as these are all full programming languages you can continue to explore. We have tried to introduce you to much of it.

Electronic Assignment Submission: Done through <http://brightspace.uwindsor.ca>

Marking Scheme: The mark for each of the questions is indicated beside each question.

Academic Integrity Statement: Submit only work that is yours and include the following confidentiality agreement and statement at the beginning of your assignment.

CONFIDENTIALITY AGREEMENT & STATEMENT OF HONESTY

I confirm that I will keep the content of this assignment/examination confidential.

I confirm that I have not received any unauthorized assistance in preparing for or doing this assignment/examination. I confirm knowing that a mark of 0 may be assigned for copied work.


Student Signature
110109609

Student I.D. Number

Darshan Sudhirkumar Bodara

Student Name (please print)
November 25, 2024

Date

For office Use only

Question	Mark
Part A	
Project report	/25
Project presentation	/25
Part B	/50
Total	/100

A: Project Report and Presentation Part require (worth 25%):

(1). Handing in an electronic copy of the following worth:

- i. An about 10 to 20 page Application system documentation report including:

 - (a) purpose of application,
 - (b) system designers and their contributions, (a and b .. 2.5 marks)
 - (c) database schema including constraints, business rules, (... 2.5 marks)
 - (d) summary of interface, (... 2.5 marks)
 - (e) features of the system, (...10 marks)
 - (f) installation guide, (...2.5 marks)
 - (g) listing of all files involved in the application and their functions, and relationships to each other. (...2.5 marks)
 - (h) user manual and any other pertinent information. (.. 2.5 marks)

(2). Making an In-class Project presentation ... (worth 25 %)

Note: Project presentation mark is split as 30% for own presentation and 20% for attendance and contribution to others' presentations.

B. Project Phase 3 (Database Application to be delivered) ... (worth 50%)

Hand in:

- i. a zipped file containing the application and all files needed to create the database and run it.

.... Main Components Graded:

 - a. Implementation of Minimal Set of 10 Functions for each person (but 20 functions for a team) for the application (15 %)
 - b. Correctness of Functionalities, constraints and business rules (15%)
 - c. Using many features of SQL querying and PHP(10 %)
 - d. Ease of Use of the application (5%)
 - e. Application interface design look ... (5%)

Guide on what constitutes a minimal set of functionalities for each individual is given in lab 3, part B as shown below [For reminder and information only]:

Part B: Design the Company Information System or other Database
Application based on the System Interface, database schema, constraints and query design, sample PL/SQL codes already designed in phases 1 and 2 of this project.

1. Design a web form based application with appropriate triggers to implement the main menu interface of your company information system application. The minimal functions provided should be similar to the four buttons for Student, Instructor/Faculty, Administration, Exit. Each of the buttons should take the user to another form or interface for an action like log in. Proper and complete navigation

from this main form with appropriate alerts and messages to the next interface completes this question.

2. Design the other component forms/interfaces similar to going from the Student part to implement a set of minimal functions (like add/dropping a course, viewing grade report, checking registration detail, viewing instructor information) for the company system you are building.
3. Design the other component forms/interfaces similar to going from Instructor/Faculty button to implement a set of minimal functions (like view course registration details, download class list, etc.) but for the company information system.
4. Design the other component forms/interfaces similar to going from the Administration button to implement a set of minimal functions (like create a course, create a student user account, create an instructor account, etc.). Note that the Admin function may be integrated with the instructor/faculty function and a select number of instructors can be given the admin power by your system but for your own company information system.
5. Design the appropriate interfaces to exit the forms main menu with appropriate messages if needed.

Answer

A) Purpose of the Application

1. Login System
 - a. Authenticates users with secure credentials.
 - b. Manages user sessions and prevents unauthorized access.
 - c. Acts as the entry point to the system.
2. Admin Portal
 - a. Allows administrators to manage system data (CRUD operations).
 - b. Monitors user activities and system performance.
 - c. Provides user account management and role-based access control.
 - d. Generates reports and tracks key metrics for better decision-making.
3. Manager Portal
 - a. Provides a comprehensive dashboard for tracking organizational metrics, including employee count, active employees, projects, and tasks.
 - b. Enables managers to view, add, update, and deactivate employee records efficiently.
 - c. Supports project management by assigning tasks, monitoring progress, and ensuring timely completion.
 - d. Tracks system activities through detailed audit logs, ensuring transparency and accountability.
 - e. Offers insights through visual data representations, such as charts and graphs, to aid in strategic decision-making.

4. User Portal
 - a. Offers a personalized dashboard displaying task status, performance scores, and recent activities for better task management.
 - b. Allows users to view and update assigned tasks with detailed descriptions and deadlines, ensuring they stay on track.
 - c. Provides a profile management feature for users to keep their personal and professional information updated.
 - d. Enhances user productivity through an intuitive interface with seamless navigation and responsive design.
 - e. Encourages task completion and progress tracking through performance analytics and visual task breakdowns.
 - f.

This application bridges the gap between management and task execution, ensuring secure, efficient, and user-friendly operations for both managers, users and admin.

B) System Designer and their Contribution

1. Smit Thesiya, Darshan Bodara
 - a. Login System Design
 - Designed and implemented the secure user authentication system.
 - Developed the login page (login.html) and session management mechanisms.
 - Ensured encryption of sensitive user data (e.g., passwords) using hashing techniques.
 - b. Admin Dashboard
 - Designed and implemented the features of the admin dashboard (admin_dashboard.php).
 - Managed system data CRUD operations (Create, Read, Update, Delete) and reporting functionalities.
 - Integrated performance monitoring tools and user activity logs.
 - c. Manager Portal
 - Designed and implemented feature of Manager Portal.
 - Having ability to see total employee, total task
 - d. User Portal
 - Designed and implemented feature of User Portal
 - Having ability to update profile, project participation
2. Nisarg Patel
 - a. Database Design and Integration
 - Designed and optimized the database schema to ensure seamless integration with the application.
 - Wrote and optimized SQL queries for both the login system and admin dashboard.
 - Implemented role-based access control to segregate user and admin privileges.

5. Database Schema and Business Rule

1. Database Schema

o Tables

- **audit_log**

I. Purpose

- ◆ Tracks changes and actions performed by users.

II. Columns

- i. log_id (PK): Unique log entry identifier.
- ii. action: Description of the action performed.
- iii. employee_id (FK): The employee affected.
- iv. performed_by (FK): The employee who performed the action.
- v. ip_address: IP address from which the action was performed.
- vi. timestamp: Date and time of the action.

- **Department**

I. Purpose

- i. Stores details about departments.

II. Columns

- i. department_id (PK): Unique department identifier.
- ii. department_name: Name of the department.
- iii. manager_id (FK): The ID of the department manager.
- iv. budget: Financial allocation for the department.

- **department_location**

I. Purpose

- i. Stores department locations.

II. Columns

- i. department_id (PK, FK): Refers to department.department_id.
- ii. city: The city where the department is located.

- **Employee**

I. Purpose

- i. Stores employee details

II. Columns

- i. employee_id (PK): Unique employee identifier.
- ii. SSN: Social Security Number.
- iii. username, password: Login credentials.

- iv. `first_name, last_name`: Employee name details.
 - v. `add`: Address of the employee.
 - vi. `DOB`: Date of birth.
 - vii. `role`: Role of the employee (Admin, Manager, User).
 - viii. `Status`: Current status (Active, Inactive).
 - ix. `performance_score`: Numeric score indicating performance.
- **`employee_absences`**
 - I. Purpose
 - i. Tracks employee absences.
 - II. Columns
 - i. `absence_id` (PK): Unique identifier for absence entries.
 - ii. `employee_id` (FK): Refers to `employee.employee_id`.
 - iii. `absence_date`: Date of the absence.
 - iv. `reason`: Reason for the absence.
- **`employee_contact`**
 - I. Purpose
 - i. Stores employee contact details.
 - II. Columns
 - i. `contact_id` (PK): Unique identifier for contact entries.
 - ii. `employee_id` (FK): Refers to `employee.employee_id`.
 - iii. `contact_type`: Type of contact (e.g., Email, Phone).
 - iv. `contact_value`: Value of the contact detail.
- **`employee_role`**
 - I. Purpose
 - i. Tracks roles assigned to employees within departments or projects.
 - II. Columns
 - i. `role_id` (PK): Unique identifier for the role.
 - ii. `employee_id` (FK): Refers to `employee.employee_id`.
 - iii. `role_name`: Name of the role (e.g., Manager, Developer).
 - iv. `dept_id` (FK): Refers to `department.department_id`.
 - v. `project_id` (FK): Refers to `project.project_id`.

- **employee_supervisor**
 - I. Purpose
 - i. Stores employee-supervisor relationships.
 - II. Columns
 - i. employee_id (PK, FK): Refers to employee.employee_id.
 - ii. supervisor_id (FK): Refers to the supervisor's employee_id.
- **Project**
 - I. Purpose
 - i. Stores project details.
 - II. Columns
 - i. project_id (PK): Unique project identifier.
 - ii. project_name: Name of the project.
 - iii. start_date, end_date: Project timeline.
 - iv. dept_id (FK): Refers to department.department_id.
- **project_assignment**
 - I. Purpose
 - i. Tracks assignments of employees to projects.
 - II. Columns
 - i. assignment_id (PK): Unique assignment identifier.
 - ii. employee_id (FK): Refers to employee.employee_id.
 - iii. assigned_by (FK): Refers to the employee who made the assignment.
 - iv. project_id (FK): Refers to project.project_id.
- **Task**
 - I. Purpose
 - i. Tracks tasks within projects.
 - II. Columns
 - i. task_id (PK): Unique task identifier.
 - ii. task_name: Name of the task.
 - iii. project_id (FK): Refers to project.project_id.
 - iv. due_date: Deadline for task completion.
 - v. status: Current status of the task (e.g., Pending, In Progress).
 - vi. progress: Completion percentage (0-100).
- **Project_department**
 - I. Purpose

- i. Establishes a relationship between projects and departments, allowing tracking of which departments are involved in specific projects.
- II. Columns
 - i. **project_department_id (PK)**: Unique identifier for each record in the table.
 - ii. **project_id (FK)**: Refers to project.project_id to associate a project with a department.
 - iii. **department_id (FK)**: Refers to department.department_id to specify the department involved.

- o Views

- **employeecertifications**: Combines employee details with their certifications and achieved dates.
- **view_employee_full_names**: Concatenates employee first and last names for reporting
- **vw_employee_certifications**: Displays employee certification details, including dates.
- **vw_employee_details**: Merges employee, department, and location details.
- **vw_employee_time_tracking**: Tracks employee task time logs within projects.
- **vw_project_status**: Summarizes project status, including completed and total tasks.

2. Business Rule

- o **User Authentication and Access Control**: Only users with valid credentials can log in. Role-based access control determines the actions users can perform (e.g., Admins can manage users and departments; regular users have limited access).
- o **User Status**: Only users with an "active" status can access the system. Deactivated users are denied access, but their data is retained in the system for audit purposes.
- o **Data Integrity**: All user and department data must be unique (e.g., unique usernames, department names). Relationships between tables must follow referential integrity (e.g., a user cannot belong to a non-existent department).
- o **Audit Logs**: Every system action (e.g., login, data modification) must be recorded in the table for accountability and traceability.
- o **CRUD Operations**: Administrators can perform Create, Read, Update, and Delete operations on users and departments. Deleting a

- department or user should not affect related data but may require reassignment of linked records.
- **Security:** Passwords must be securely hashed before being stored. Sensitive operations, such as user management, require administrator permissions.
- **Reporting and Analytics:** Active and Inactive user counts, user role distribution, and departments statistics are dynamically updated for reporting.

6. Summary of the Interface

- a. The application interface is designed to be intuitive and functional, catering to both regular users and administrators. The **Login Interface** provides a secure and user-friendly platform for users to enter their credentials, such as a username and password. It ensures data accuracy through validation and provides immediate feedback, such as success messages for valid credentials or error alerts for invalid ones. This interface serves as the gateway to the application's functionalities, guiding users seamlessly to their respective areas.
- b. The **Admin Portal**, tailored specifically for administrators, provides a comprehensive control panel with a clean, intuitive layout. It includes navigation menus that allow administrators to easily access and manage various system functionalities, such as viewing, adding, updating, and deleting records (CRUD operations). The dashboard supports user account management with role-based access control, enabling administrators to assign roles, reset passwords, and deactivate accounts as needed. Additionally, it provides real-time insights through system metrics, activity logs, and performance summaries, ensuring effective monitoring and decision-making. Advanced features, such as report generation and resource management, enhance the administrative experience, making it efficient and scalable. Together, these interfaces deliver a seamless, secure, and productive experience for both end-users and administrators, supporting the application's overall purpose.
- c. The application interface is designed to cater effectively to both managerial and user roles, providing a seamless and intuitive experience. The **Manager Portal** offers a comprehensive dashboard that displays key metrics such as Total Employees, Active Employees, Total Projects, and Total Tasks. It also includes a visual representation of Employee Status through a chart for quick insights. The sidebar navigation menu allows managers to easily access functionalities such as viewing employee details, managing projects, and monitoring audit logs for accountability. The interface is clean and minimalistic, ensuring that real-time data is presented in an accessible and actionable manner.
- d. On the other hand, the **User Portal** is tailored to enhance user productivity and engagement. It greets users with a personalized

welcome message and displays critical performance metrics, including the Performance Score and a breakdown of Completed, In-Progress, and Pending Tasks. A visually appealing chart further categorizes task statuses for clarity. The portal also provides users with functionalities to view their assigned tasks and update their profiles, ensuring a smooth workflow. Additionally, a recent activity log keeps track of user actions, promoting transparency and accountability. The design emphasizes usability, with clear navigation and role-specific features such as displaying the user's role, department, and status.

7. Feature Of the system

1. Login

- The login feature allows users to securely access the system by entering their credentials (username and password). It includes:
 - I. Secure authentication with hashed passwords.
 - II. Validation to prevent unauthorized access.
 - III. Role-based redirection (user vs. admin).
 - IV. Session Handing for active login sessions.

The image shows a mobile-style login screen. At the top center, it says "Login". Below that is a "Username" field containing "darshan.bodara". Below that is a "Password" field with the placeholder "Enter your password". Below that is a "Role" field with a dropdown menu showing "Admin". At the bottom center is a purple "Login" button. Below the button is a link "Don't have an account? [Sign up](#)".

2. Sign Up

- User input validation
- Automatic password hashing

Signup

SSN:

Username:

Password:

First Name:

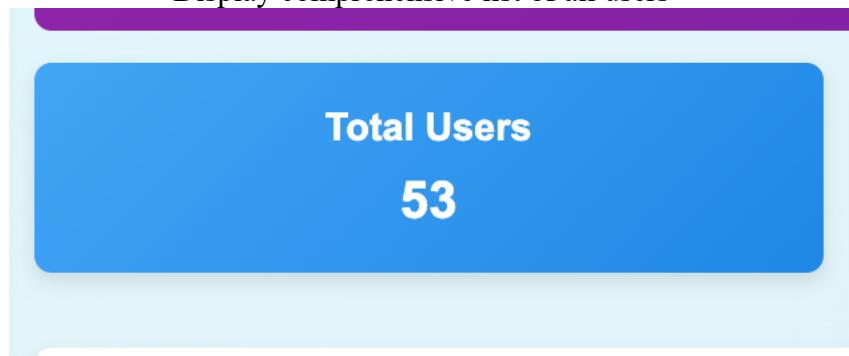
Last Name:

Address:

Signup

3. List of all the user

- Display comprehensive list of all users



ID	Username	First Name	Last Name	Role	Status	Actions
1	darshan.bodara	Darshan	Bodara	Admin	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
2	smit.thesiya	Smit	Thesiya	Manager	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
3	nisarg.patel	Nisarg	Patel	User	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
4	alice.johnson	Alice	Johnson	User	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
5	bob.smith	Bob	Smith	User	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
6	charlie.brown	Charlie	Brown	User	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
7	diana.prince	Diana	Prince	User	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
8	evan.wallace	Evan	Wallace	User	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>

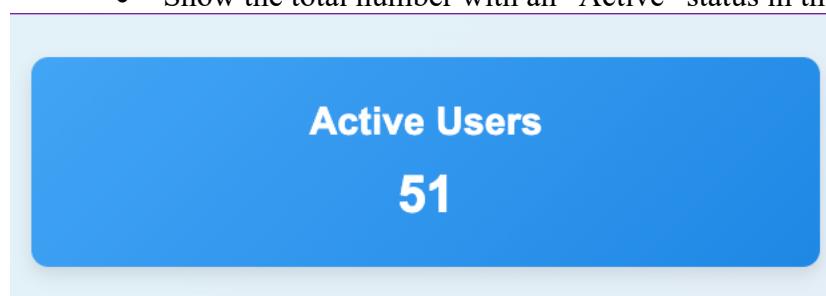
4. List of all the departments

- Provide categorized list of the departments in the organization including:
 - I. Department Id, Name, Manager, Budget

ID	Name	Manager	Budget	Actions
1	Software Engineering	Darshan Bodara	1,000.00	<button>Edit</button>
2	Product Management	Smit Thesiya	0.00	<button>Edit</button>
3	Human Resources	Nisarg Patel	0.00	<button>Edit</button>
4	Quality Assurance	Alice Johnson	0.00	<button>Edit</button>
5	Sales	Bob Smith	0.00	<button>Edit</button>
6	Marketing	Charlie Brown	0.00	<button>Edit</button>
7	Customer Service	Diana Prince	0.00	<button>Edit</button>
8	Research and Development	Evan Wallace	0.00	<button>Edit</button>
9	Finance	Fiona Gallagher	0.00	<button>Edit</button>
10	Legal	George Lucas	0.00	<button>Edit</button>

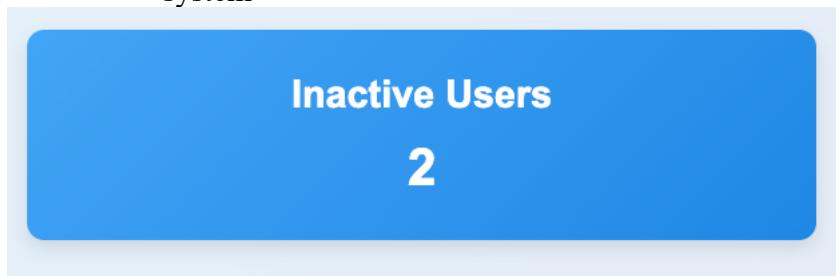
5. Count of the active users

- Show the total number with an “Active” status in the system



6. Count of the inactive users

- Show all the total numbers with an “Inactive” status in the system



7. Adding User

- Allows administrators to add new users by providing:
 - I. Fields for entering user details (SSN, First Name, DOB etc.)

Add User

SSN:

Username:

Password:

First Name:

Last Name:

Please fill out this field.

Address:

Date of Birth:

Role:

Status:

8. Editing User

- Enables administrators to edit existing user details, such as
 - I. Enables administrators to edit existing user details, such as name, role, etc.

Edit User

Username:

First Name:

Last Name:

Address:

Role:

Status:

9. Activating / Deactivating Users

- Provides a toggle for changing user status:
 - I. Activation: Grants system access.

- II. Deactivation: Revokes access without deleting data.
- III. Includes confirmation prompts to prevent accidental actions.

The screenshot shows a modal dialog box with a black background and white text. It says "bodara2.myweb.cs.uwindsor.ca says" at the top, followed by "Are you sure you want to Activate this user?". There are two buttons at the bottom: "Cancel" on the left and "OK" on the right. A red arrow points from the text "Includes confirmation prompts to prevent accidental actions." to the "OK" button. Below the dialog is a table titled "Manage Users". The table has columns: First Name, Last Name, Role, Status, and Action. Under the "Action" column, there are three buttons for each row: "Edit", "Activate" (highlighted in green), and "Deactivate". The table contains four rows of data.

First Name	Last Name	Role	Status	Action
Darshan	Bodara	Admin	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
Smit	Thesiya	Manager	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
Bob	Smith	User	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>

10. Searching User

- Facilitates quick lookup of users by:
 - I. First Name, Last Name
 - II. Supports partial matching
 - III. Integrate with filtering options for advanced search.

The screenshot shows a user management interface titled "Manage Users". At the top, there is a search bar with the placeholder "Smit" and a "Search" button. Below the search bar is a blue "Add User" button. The main area is a table with columns: ID, Username, First Name, Last Name, Role, Status, and Actions. The table contains four rows of user data. Each row has an "Actions" column with three buttons: "Edit", "Activate" (highlighted in green), and "Deactivate".

ID	Username	First Name	Last Name	Role	Status	Actions
2	smit.thesiya	Smit	Thesiya	Manager	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
5	bob.smith	Bob	Smith	User	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
18	Jane.Smith	Jane	Smith	Admin	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>
28	Lucas.Smith	Lucas	Smith	Manager	Active	<button>Edit</button> <button>Activate</button> <button>Deactivate</button>

11. Adding Departments

- Allows administrators to add new departments with:
 - I. Fields for department name, Manager, Budget
 - II. Validation to avoid duplicate entries.

Add Department

Department Name:

Manager:

Budget:

CancelSave

12. Editing Departments

- Enables modification of department details, including:
 - I. Name, Manager ID, Budget

Edit Department

Department Name:

Manager ID:

Budget:

Save ChangesCancel

13. Searching Departments

- Provides a search feature to find departments by:
 - I. Name or Keywords
 - II. Supports partial matching.

Manage Departments

Soft	Search			
Add Department				
ID	Name	Manager	Budget	Actions
1	Software Engineering	Darshan Bodara	1,000.00	<button>Edit</button>

14. View Audit Logs

- Displays system audit logs, tracking activities such as:
 - Activity such as Updating, Adding
 - Listing ID, Action, Employee, Time Stamp

Audit Logs			
ID	Action	Employee	Timestamp
14	Deactivated user ID: 15	John Doe	2024-11-22 16:28:42
13	Activated user ID: 2	Smit Thesiya	2024-11-22 16:28:26
12	Updated user details: smit.thesiya	Smit Thesiya	2024-11-22 16:24:57
7	Updated department details: Software Engineering	System	2024-11-22 09:16:41
6	Updated department details:	System	2024-11-22 09:13:07

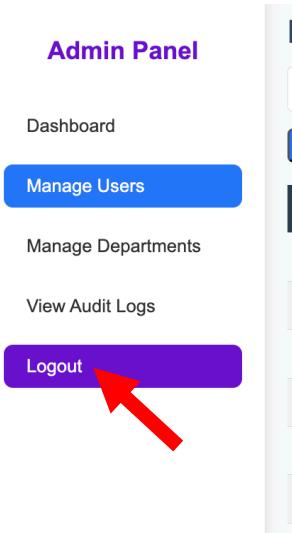
15. Search Audit Logs

- Offers search capabilities within the audit logs for:
 - Specific user activities.
 - Specific Employee
 - Filtering by type of action.

Audit Logs			
ID	Action	Employee	Timestamp
13	Activated user ID: 2	Smit Thesiya	2024-11-22 16:28:26
12	Updated user details: smit.thesiya	Smit Thesiya	2024-11-22 16:24:57

16. Log Out Feature

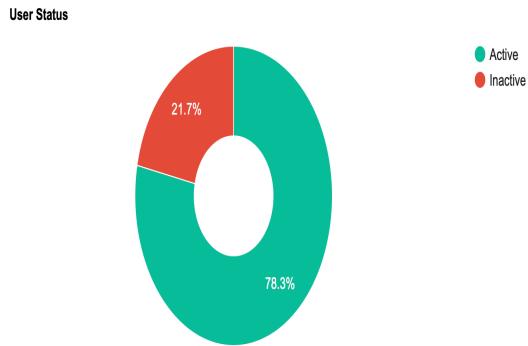
- Ensures a secure logout process by:
 - Redirecting users to the login page.
 - Logging the action in the audit logs for tracking.



17. Pie Chart for Active / Inactive Users

- Visualizes the proportion of active vs. inactive users using:
 - Dynamic data from the database.

- II. Hover functionality to display exact counts and percentages.
- III. Real-time updates reflecting user status changes.



18. Bar Graph for User Role Distributions

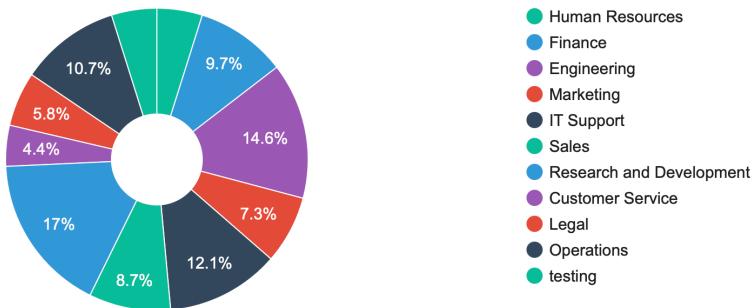
- Represents the distribution of users across roles (e.g., admin, user, manager):
 - I. Segments by role with count labels
 - II. Useful for analyzing system usage by role type.



19. Pie Chart for Budget

- Provides a visual representation of budget allocations across departments:
 - I. Percentage breakdown by department.
 - II. Dynamic update reflecting real-time data changes.

Department Budgets



20. Count Of Completed, In Progress, Pending Task

- Displays task status metrics for effective tracking
 - I. Shows the count of tasks categorized as "Completed."
 - II. Highlights tasks "In Progress."
 - III. Tracks the number of "Pending" tasks

Welcome, Smit!

Role: User | Department: N/A | Status: Active

Performance Score

72

Completed Tasks

2

In Progress Tasks

1

Pending Tasks

0

21. Update Profile

- Enables users to manage their personal information.
 - I. Allows users to update their contact details, password, and other profile information.
 - II. Ensures secure handling of sensitive data.

Update Profile

Username:

First Name:

Last Name:

Address:

Email:

Phone:

Save **Cancel**

22. View Task

- Allows users to view assigned tasks with relevant details
 - I. Displays task name, due date, and progress.
 - II. Enables filtering tasks based on status or deadlines.

User Panel

- Dashboard
- View Tasks**
- Update Profile
- Logout

Your Tasks

Task ID	Task Name	Status	Due Date	Project	Actions
1	Design Database Schema	Completed	2024-12-15	Website Redesign	<button>Edit</button>
2	Develop Backend API	In Progress	2024-12-20	Website Redesign	<button>Edit</button>
3	Write Unit Tests	Completed	2024-12-25	Website Redesign	<button>Edit</button>

23. Manager Portal: Total Employee

- Displays total employee count under a manager's supervision
 - I. Provides an overview of team size.
 - II. Supports detailed breakdown of team members when clicked.

Welcome, Manager manager2!

Total Employees

5

Active Employees

5

24. Count of Total Task, Project, Total Task

- Shows aggregated metrics for project and tasks
 - I. Displays total count of tasks in the system
 - II. Highlights the total number of projects.
 - III. Provides easy navigation to detailed task and project views.

25. Manage Project

- Offers functionality to add, update, and delete projects
 - I. Allows administrators or managers to create new projects.
 - II. Updates project details such as deadlines and assigned teams.
 - III. Removes obsolete or completed projects from the system.

Edit Project

Project Name:

Website Redesign

Start Date:

2024-01-15

End Date:

2024-03-30

Department:

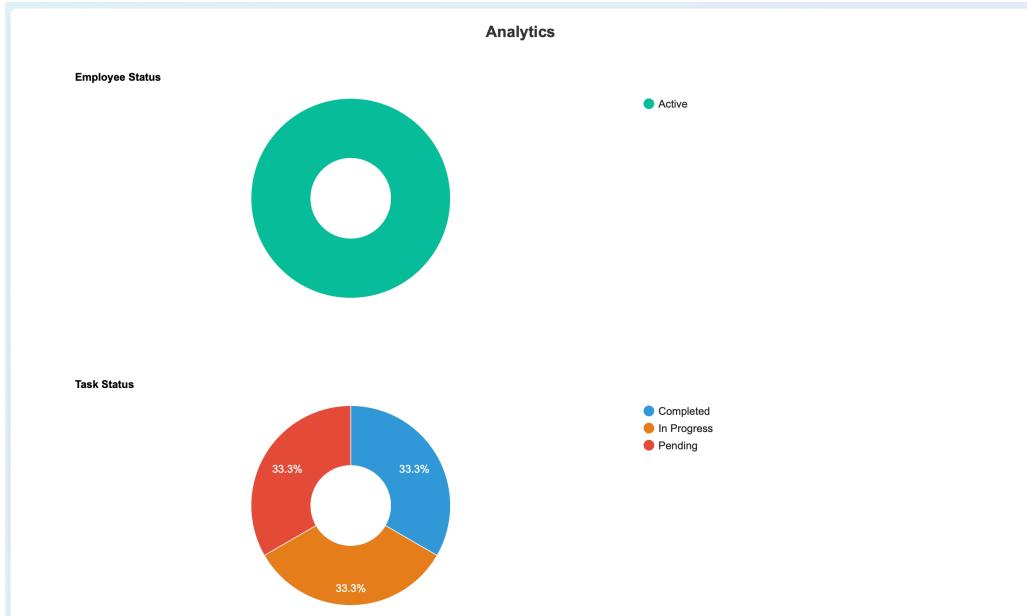
IT Support

Save

Cancel

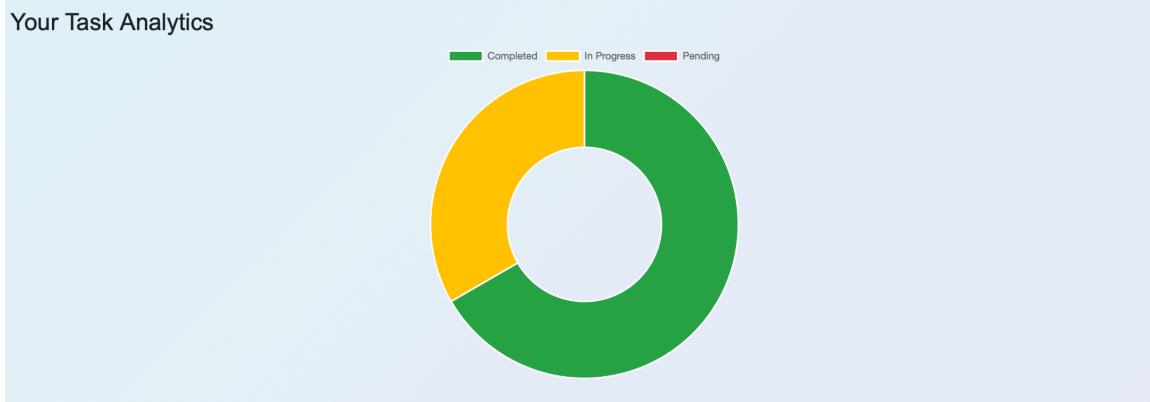
26. Chart For Manager

- I. Employee Status
- II. Task Status



27. Chart for User

I. Task Status



28. Recent Activity for User:

I. Display recent login and logout activity of the user

Recent Activities
Logged In - Nov 25, 2024, 05:26 PM
Logged Out - Nov 25, 2024, 04:04 PM
Logged In - Nov 25, 2024, 03:59 PM
Logged Out - Nov 25, 2024, 02:56 PM
Logged In - Nov 25, 2024, 02:54 PM

29. List of the Project

I. List of the project for the user

Your Projects

Project Name	Department	Start Date	End Date
Website Redesign	IT Support	2024-01-15	2024-03-30

8. Installation Guide

1. Prerequisites

- Web Server: Apache (use tools like XAMPP, WAMP, or an institutional server like CS PHP Direct Admin).
- PHP Admin: Version 7.4 or higher.
- Database Server: MySQL
- SFTP Client: FileZilla or Bitvise for file transfers.
- Browser: Google Chrome, Firefox, or Edge for testing.
- Direct Admin Access: Credentials to log in to UWindsor server

2. Configure the database

- Database Connection Detail

```
❖ DB_HOST=localhost
❖ DB_NAME=COMP_4150_project_trial
❖ DB_USERNAME=root
❖ DB_PASSWORD=
```



3. Access the application

- Server / Link : [Employee Management System](https://bodara2.myweb.cs.uwindsor.ca/Phase3/login.html)
[\(https://bodara2.myweb.cs.uwindsor.ca/Phase3/login.html\)](https://bodara2.myweb.cs.uwindsor.ca/Phase3/login.html)

4. Test the application

- Use the following credentials
- The system will determine your role based on your username and password so please test according to it.

I. Admin

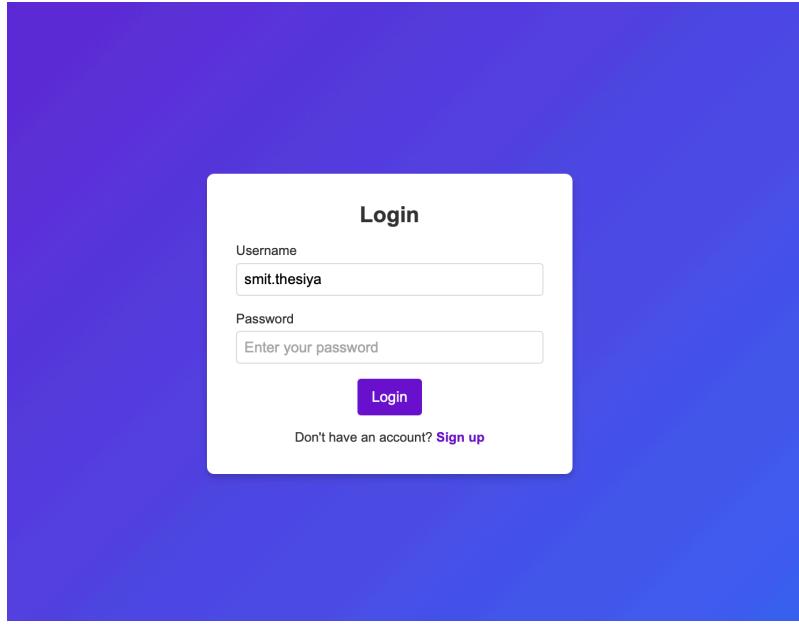
Username: darshan.bodara
Password: password123

II. Manager

Username: manager1
Password: password123

III. User

Username: user1
Password: password123



9. Listing all the file involved in the application and their functions, and relationships to each other.

- **activate_user.php**

- ❖ **Purpose:** Handles the activation of inactive user accounts in the system.
- ❖ **Functionality:**
 - Retrieves user information from the database.
 - Updates the Status field for a user to "Active."
- ❖ **Relationships:**
 - Connected to db_connection.php for database operations.
 - Linked to admin_dashboard.php for user management functionalities.
 - - **add_department_handler.php**
- ❖ **Purpose:** Handles the creation of new departments.
- ❖ **Functionality:**
 - Processes form data for adding departments.
 - Inserts department data (e.g., name, manager, budget) into the department table.
- ❖ **Relationships:**
 - Depends on db_connection.php for database transactions.
 - Interfaces with forms in admin_dashboard.php.
 - - **add_project.php**
- ❖ **Purpose:** Provides a user interface to add new projects.
- ❖ **Functionality:**
 - Displays input fields for project details (e.g., name, start date, department).
 - Sends data to add_projects_action.php for processing.
- ❖ **Relationships:**

- Connected to add_projects_action.php for back-end handling.
- Utilizes db_connection.php for data fetching or validation.
- - **add_projects_action.php**
- ❖ **Purpose:** Processes data from add_project.php and adds projects to the database.
- ❖ **Functionality:**
 - Validates project data.
 - Inserts records into the project table.
- ❖ **Relationships:**
 - Relies on db_connection.php for database operations.
 - Can trigger updates in related files, like admin_dashboard.php, for project listing.
 - - **add_user_handler.php**
- ❖ **Purpose:** Handles the creation of new user accounts.
- ❖ **Functionality:**
 - Processes user details from the front end.
 - Inserts user data (e.g., username, role, department) into the employee table.
- ❖ **Relationships:**
 - Works with db_connection.php for database transactions.
 - Linked to user management forms in admin_dashboard.php.
 - - **admin_dashboard.php**
- ❖ **Purpose:** Core dashboard for administrators.
- ❖ **Functionality:**
 - Displays user, project, and department statistics.
 - Provides navigation to other modules (e.g., user management, department handling).
- ❖ **Relationships:**
 - Central point connecting multiple modules, including:
 - activate_user.php
 - add_department_handler.php
 - add_projects_action.php
 - audit_logs.php
 - - **audit_logs.php**
- ❖ **Purpose:** Displays a log of system actions for auditing purposes.
- ❖ **Functionality:**
 - Fetches and displays records from the audit_log table.
 - Provides search and filter options for logs.
- ❖ **Relationships:**
 - Utilizes db_connection.php for retrieving logs.
 - Connected to actions across the system for logging purposes.
 - - **composer.json**
- ❖ **Purpose:** Defines PHP dependencies for the application.
- ❖ **Functionality:**

- Specifies packages to install (e.g., vlucas/phpdotenv for environment variables).
- ❖ **Relationships:**
 - Works with composer.lock for dependency management.
 - Ensures all dependencies, like db_connection.php, are loaded correctly.

 - **composer.lock**
- ❖ **Purpose:** Locks the specific versions of installed dependencies.
- ❖ **Functionality:**
 - Ensures consistent dependency installation across environments.
- ❖ **Relationships:**
 - Related to composer.json for dependency resolution.
 -

 - ◆ **db_connection.php**
- ❖ **Purpose:** Manages database connections.
- ❖ **Functionality:**
 - Contains methods to connect to the database using credentials.
 - Provides a reusable connection object for other files.
- ❖ **Relationships:**
 - Essential for all files interacting with the database (e.g., activate_user.php, add_department_handler.php).

- ❖
 - **deactivate_user.php**
- ❖ **Purpose:** Handles deactivation of user accounts.
- ❖ **Functionality:**
 - Updates the Status field for a user to "Inactive."
 - Provides error handling for invalid requests.
- ❖ **Relationships:**
 - Depends on db_connection.php for database operations.
 - Links with admin_dashboard.php for user management.

- ❖
 - **delete_department.php**
- ❖ **Purpose:** Manages the deletion of departments from the system.
- ❖ **Functionality:**
 - Validates the department ID before deletion.
 - Deletes department records from the department table.
- ❖ **Relationships:**
 - Utilizes db_connection.php for database interactions.
 - Connected to the department management forms in admin_dashboard.php.

- ❖
 - **edit_department.php**
- ❖ **Purpose:** Facilitates the editing of department details.
- ❖ **Functionality:**
 - Fetches existing department details for editing.
 - Updates department attributes in the database.
- ❖ **Relationships:**
 - Relies on db_connection.php for data retrieval and updates.

- Works with the department interface in `admin_dashboard.php`.
- ❖
 - **`edit_user.php`**
- ❖ **Purpose:** Enables editing of user profiles.
- ❖ **Functionality:**
 - Fetches user details from the employee table.
 - Updates user attributes such as role or status.
- ❖ **Relationships:**
 - Dependent on `db_connection.php` for database access.
 - Integrated with user management features in `admin_dashboard.php`.
- ❖
 - **`fetch_audit_logs.php`**
- ❖ **Purpose:** Retrieves audit logs for display.
- ❖ **Functionality:**
 - Queries the `audit_log` table to fetch system actions.
 - Outputs logs for viewing or exporting.
- ❖ **Relationships:**
 - Works with `audit_logs.php` to present log data.
 - Requires `db_connection.php` for database queries.
- ❖
 - **`fetch_employees.php`**
- ❖ **Purpose:** Fetches a list of employees for the admin dashboard.
- ❖ **Functionality:**
 - Retrieves employee records from the `employee` table.
 - Supports filtering (e.g., active vs. inactive users).
- ❖ **Relationships:**
 - Relies on `db_connection.php` for database access.
 - Interacts with `admin_dashboard.php` for real-time updates.
- ❖
 - **`index.php`**
- ❖ **Purpose:** Entry point of the application.
- ❖ **Functionality:**
 - Directs users to the login page if not authenticated.
 - Redirects authenticated users based on roles.
- ❖ **Relationships:**
 - Utilizes `db_connection.php` for session validation.
 - Links to `login.html` and `admin_dashboard.php`.
- ❖
 - **`log_function.php`**
- ❖ **Purpose:** Provides logging capabilities for system events.
- ❖ **Functionality:**
 - Inserts log entries into the `audit_log` table.
 - Tracks user actions (e.g., adding, editing, deleting).
- ❖ **Relationships:**
 - Called by various files (`add_user_handler.php`, `delete_department.php`) to log actions.

- Connected to fetch_audit_logs.php for retrieving logs.
- ❖
 - **login.html**
- ❖ **Purpose:** Front-end interface for user authentication.
- ❖ **Functionality:**
 - Collects user credentials and role.
 - Submits data to login.php for validation.
- ❖ **Relationships:**
 - Linked to login.php for back-end processing.
 - References styles/login.css for design consistency [\[101+source\]](#).
- ❖
 - **if (!isset(\$_SESSION['logged_in']) || ...)**
- ❖ **Purpose:** Session management and user access control.
- ❖ **Functionality:**
 - Verifies if a user is logged in before accessing restricted pages.
 - Redirects unauthorized users to login.html.
- ❖ **Relationships:**
 - Integrated into secure pages like admin_dashboard.php.
 - Works with db_connection.php for session validation.
- ❖
 - **login.php**
- ❖ **Purpose:** Backend script for user authentication.
- ❖ **Functionality:**
 - Validates user credentials against the database.
 - Creates user sessions for authenticated users.
- ❖ **Relationships:**
 - Works with db_connection.php for database validation.
 - Redirects authenticated users to the appropriate dashboard (admin_dashboard.php, manager_dashboard.php).
- ❖
 - **logout.php**
- ❖ **Purpose:** Handles user session termination.
- ❖ **Functionality:**
 - Destroys the active session and redirects to login.html.
- ❖ **Relationships:**
 - Works with all files that use sessions to manage access control.
- ❖
 - **manage_departments.php**
- ❖ **Purpose:** Provides an interface for managing departments.
- ❖ **Functionality:**
 - Displays department data.
 - Links to functionalities like adding, editing, or deleting departments.
- ❖ **Relationships:**
 - Connected to edit_department.php and delete_department.php for CRUD operations.
 - Fetches data using db_connection.php.

- ❖
 - **manage_projects.php**
- ❖ **Purpose:** Displays and manages project details.
- ❖ **Functionality:**
 - Lists projects and their associated departments.
 - Offers options to add, edit, or delete projects.
- ❖ **Relationships:**
 - Connected to manage_projects_action.php for back-end operations.
 - Uses db_connection.php for database interactions.
- ❖
 - **manage_projects_action.php**
- ❖ **Purpose:** Handles actions like adding or updating project data.
- ❖ **Functionality:**
 - Validates input data for projects.
 - Updates the project table in the database.
- ❖ **Relationships:**
 - Called by manage_projects.php.
 - Relies on db_connection.php for database transactions.
- ❖
 - **manage_users.php**
- ❖ **Purpose:** Central interface for user management.
- ❖ **Functionality:**
 - Lists all users, including active and inactive accounts.
 - Provides options to activate, deactivate, or edit user profiles.
- ❖ **Relationships:**
 - Interacts with edit_user.php and activate_user.php.
 - Fetches user data using fetch_employees.php.
- ❖
 - **manager_dashboard.php**
- ❖ **Purpose:** Dashboard for managers.
- ❖ **Functionality:**
 - Displays project and department statistics relevant to managers.
 - Provides navigation to specific management functionalities.
- ❖ **Relationships:**
 - Works with manager_sidebar.php for UI rendering.
 - Accesses shared modules like manage_projects.php and fetch_employees.php.
- ❖
 - **manager_sidebar.php**
- ❖ **Purpose:** Sidebar navigation for the manager dashboard.
- ❖ **Functionality:**
 - Provides quick links to manager-specific tools (e.g., project management, team overview).
- ❖ **Relationships:**
 - Included in manager_dashboard.php.
 - Links to files like manage_projects.php and manage_departments.php.

- **signup.html**
- ❖ **Purpose:** User registration interface.
- ❖ **Functionality:**
 - Collects user data (e.g., SSN, username, password, address).
 - Submits data to signup.php for processing.
- ❖ **Relationships:**
 - Styled using styles/signup.css.
 - Sends data to signup.php for back-end handling [\[116↑source\]](#).
- ❖

 - **signup.php**
- ❖ **Purpose:** Processes user registration.
- ❖ **Functionality:**
 - Validates and inserts user data into the employee table.
 - Ensures unique usernames and valid input data.
- ❖ **Relationships:**
 - Works with db_connection.php for database operations.
 - Receives data from signup.html.
- ❖

 - **update_performance.php**
- ❖ **Purpose:** Updates employee performance scores.
- ❖ **Functionality:**
 - Accepts employee ID and new performance score as inputs.
 - Updates the performance_score field in the employee table.
 - Includes input validation to ensure valid performance scores.
- ❖ **Relationships:**
 - Depends on db_connection.php for database operations.
 - Can be triggered from user or manager dashboards.
- ❖

 - **user_dashboard.php**
- ❖ **Purpose:** Dashboard for regular users.
- ❖ **Functionality:**
 - Displays tasks, projects, or certifications assigned to the logged-in user.
 - Provides quick access to personal information and updates.
- ❖ **Relationships:**
 - Connected to utils.php for common functionality.
 - Fetches data from tables like employee, project, and task via db_connection.php.
- ❖

 - **utils.php**
- ❖ **Purpose:** Contains utility functions used across the application.
- ❖ **Functionality:**
 - Includes helper functions for formatting, input validation, and session handling.
 - Provides reusable methods for common tasks.
- ❖ **Relationships:**
 - Used by multiple scripts, including user_dashboard.php and view_employees.php.
 - Enhances maintainability by centralizing shared logic.
- ❖

- **view_employees.php**
- ❖ **Purpose:** Displays a detailed list of employees.
- ❖ **Functionality:**
 - Fetches employee details from the database.
 - Supports filtering, sorting, and pagination for large datasets.
- ❖ **Relationships:**
 - Uses db_connection.php for database queries.
 - Often accessed from dashboards like admin_dashboard.php.
- ❖
- **view_audit_logs.php**
- ❖ **Purpose:** Displays system audit logs for administrative review.
- ❖ **Functionality:**
 - Queries and presents data from the audit_log table.
 - Supports search and filter functionality to find specific log entries.
- ❖ **Relationships:**
 - Utilizes fetch_audit_logs.php for retrieving logs.
 - Works with admin_dashboard.php for navigation and integration.
- ❖

10. User manual and any other pertinent information

1. User Manual:

- This system is designed to provide secure and efficient management for users, departments, and system activity logs. Users can log in via the [Login Page](#) using their credentials. The login feature ensures data security through validation, session management, and password hashing. New users can register using the optional sign-up feature, subject to admin approval. Administrators can access advanced features through admin dashboard where they can manage users, departments, and system activities.
- The dashboard includes functionalities like listing, adding, editing, activating/deactivating users, and searching for specific users or departments. Administrators can also view and search audit logs to track system activities. Visual data representations, such as pie charts for active/inactive users and budget distributions, and bar graphs for user role distribution, provide insights into system usage.
- The **Manager Portal** is designed to provide managers with a streamlined and efficient interface for overseeing organizational operations and employee activities. Managers can log in securely through the Login Page using their credentials, ensuring data security with features like validation and session management. Upon accessing the dashboard, managers are presented with key

metrics, such as the total number of employees, active employees, projects, and tasks. The portal enables managers to view detailed employee records, manage projects by assigning tasks, and monitor progress effectively. Additionally, the audit logs feature allows managers to track system activities, ensuring accountability and transparency. The system is built with responsiveness and scalability in mind, making it accessible across various devices while supporting future enhancements to meet organizational needs

- The **User Portal** is tailored to provide users with a personalized and intuitive interface to manage their tasks and track their performance. After securely logging in through the Login Page, users are welcomed with a dashboard displaying essential metrics such as their performance score and a breakdown of tasks, including completed, in-progress, and pending ones. Users can view assigned tasks with detailed descriptions and deadlines, enabling them to update their progress and stay organized. The portal also allows users to update their profile information, ensuring accurate records and effective communication within the system. Designed for ease of use, the User Portal ensures productivity by offering responsive design and seamless navigation, making it an ideal tool for managing individual responsibilities efficiently.
- Security features include session management, role-based access control. The system is responsive, working seamlessly across devices, and scalable for future enhancements. For troubleshooting, contact the administrator or refer to this manual for guidance.