

**A**  
**Mini Project Report**  
**on**  
**PerformAce - Employee Performance Management**  
**Second Year Engineering–Computer Science Engineering (Data**  
**Science)**

**Submitted By**

**Tushar Zaware                      23107089**

**Aditya Sharma                      23107111**

**Dhanshri Shinde                      23107107**

**Konika Yadav                      22107006**

**Under The Guidance Of**

**Ms. Aavani N**



---

**DEPARTMENT OF CSE (DATA SCIENCE)**

**A.P.SHAH INSTITUTE OF TECHNOLOGY**

**G.B. Road, Kasarvadavali, Thane (W), Mumbai-400615**

**UNIVERSITY OF MUMBAI**

**Academic year : 2024-25**

## **CERTIFICATE**

This to certify that the Mini Project report on PerformAce - Employee Performance Management been submitted by Tushar Zaware (23107089), Aditya Sharma (23107111), Dhanshri Shinde (23107107) and Konika Yadav(22107006) who are Bonafede students of A. P. Shah Institute of Technology, Thane, Mumbai, as a partial fulfilment of the requirement for the degree in **CSE(DATA SCIENCE)**, during the academic year **2024-2025** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

**Ms. Aavani N**  
**Guide**

**Ms. Anagha Aher**  
**HOD, CSE (Data Science)**

**Dr. Uttam D.Kolekar**  
**Principal**

**External Examiner(s)**  
1.

**Internal Examiner(s)**  
1.

**Place:** A. P. Shah Institute of Technology, Thane

**Date:**

## **ACKNOWLEDGEMENT**

This project would not have come to fruition without the invaluable help of our guide **Ms. Aavani N.** Expressing gratitude towards our HoD, **Ms. Anagha Aher**, and the Department of CSE(Data Science) for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our teachers **Ms. Richa Singh** who gave us her valuable suggestions and ideas when we were in need of them. We would also like to thank our peers for their helpful suggestions.

## TABLE OF CONTENTS

1. Introduction.....	1
1.1.Purpose.....	1
1.2.Problem Statement.....	2
1.3.Objectives.....	3
1.4.Scope.....	4
2. Proposed System.....	5
2.1.Features and Functionality.....	5
3. Project Outcomes.....	8
4. Software Requirements.....	10
5. Project Design.....	11
6. Project Scheduling.....	14
7. Results.....	16
8. Conclusion.....	25
References	

# **Chapter 1**

## **Introduction**

The project “PerformAce - Employee Performance Management” discussed in this report focuses on the development of a Python-based application designed to assist organizations with managing employee performance, task assignments, evaluations, and notifications of upcoming responsibilities. The primary goal of this project is to create a tool that streamlines performance management, making it easier for employees, managers, and administrators to track and evaluate work efficiency in an organized and structured manner.

The problem of fragmented task management, where employees often have to refer to different sources for various assignments and performance evaluations, is tackled in this system by integrating all necessary functions into one comprehensive platform.

This Python-based application will allow users to choose, manage, and track their tasks, update progress, and receive timely notifications. It will also enable managers to evaluate employee performance based on predefined assessment criteria. The development of this application reflects the growing demand for efficient employee performance tracking tools that cater to organizational needs. By consolidating several performance management tasks into one system, the application provides a seamless experience for users, ensuring better organization and productivity.

### **1.1. Purpose**

The purpose of this document is to present a comprehensive report on the development and design of PerformAce – Employee Performance Management, a Python-based application aimed at improving employee productivity and task management in an organization. The application is designed to assist employees in managing their tasks, setting goals, tracking deadlines, and evaluating their performance efficiently.

In today’s fast-paced corporate environment, effective task management and performance evaluation are crucial for both employees and employers. Many organizations face

challenges related to inefficient tracking of employee tasks, missed deadlines, and a lack of real-time performance insights. This application addresses these issues by providing a centralized platform where employees can:

- Select tasks based on priority and deadlines.
- Track their progress and manage workloads efficiently.
- Receive timely notifications for upcoming deadlines and pending tasks.
- Generate performance reports based on predefined evaluation criteria.

This report serves as a guide for developers, business managers, HR professionals, and IT teams interested in understanding the architecture, functionalities, and benefits of this application. It also provides insights into how organizations can streamline employee productivity and enhance performance assessment mechanisms using digital solutions.

## 1.2 Problem Statement

In modern workplaces, employees and managers face several challenges when it comes to managing tasks and tracking performance. The process of assigning work, updating progress, monitoring efficiency, and staying on top of deadlines often involves using multiple tools or platforms, which can be confusing and time-consuming.

- **Fragmented Task Management:** Employees often have to use different tools or platforms to manage tasks, track progress, and update status. This fragmentation makes it difficult to stay organized and efficient.
- **Lack of Timely Updates:** Without a reliable notification system, employees may forget about deadlines, leading to delayed projects or mismanagement of priorities.
- **Difficulty in Centralizing Performance Data:** With employee performance data scattered across different reports, managers struggle to evaluate efficiency and provide timely feedback.

- The proposed application seeks to address these problems by providing an integrated solution where users can manage all aspects of their tasks and performance evaluations within a single system, improving productivity and time management.

### 1.3. Objectives

The objectives of this project are to create a user-friendly standalone Python application that addresses the performance management needs of employees and managers. The application aims to facilitate efficient task tracking, performance evaluations, and deadline management, ultimately enhancing productivity and organizational efficiency. Additionally, it seeks to provide customizable features that cater to the diverse requirements of users, ensuring broad applicability across various business settings.

- **To integrate task management, employee performance tracking, and reports through tables from a database using MySQL:** The aim is to centralize work-related data, such as assigned tasks, performance evaluations, and work status, into a structured system.
- **To create employee task records and performance reports, making them easier to analyze using data normalization:** This objective focuses on organizing and streamlining work-related data to ensure efficient storage, retrieval, and reporting.
- **To generate employee performance reports via a predefined evaluator in the Evaluation table:** This involves automating the generation of performance reports by leveraging assessment metrics stored in a dedicated MySQL evaluation table.
- **To provide updates about upcoming tasks and deadlines by implementing a calendar system in MySQL:** The goal is to notify employees of assigned tasks and deadlines, ensuring they remain updated on their responsibilities.

## 1.4 Scope

The scope of this project includes the design and development of a standalone Python application that can be used by individuals, teams, or entire organizations. The application is intended to be deployed on personal computers, laptops, or office servers and serves a wide audience, from employees managing their daily tasks to managers overseeing team performance. The application will cover the following functionalities:

- **Task Management:** A feature where users can choose, update, and track their assigned tasks with completion status and deadlines.
- **Performance Evaluation:** A dedicated section where managers can assess employee efficiency based on predefined criteria and generate reports.
- **Deadline Management:** A feature that allows employees to set task reminders and receive notifications about due dates.
- **Task Notifications:** The application will include a notification system to alert users of important deadlines and assigned responsibilities.

While the initial scope focuses on providing these core features, the application can be extended to include additional functionalities such as integration with HR management tools, cloud storage solutions, and automated report generation based on AI-driven analytics, depending on the users' evolving needs.

By consolidating employee task management and performance evaluation into a single platform, the application aims to enhance organizational productivity and ensure better workflow management.



# Chapter 2

## Proposed System

The proposed system is a Python-based application designed to consolidate employee task management and performance evaluation into a single platform. It eliminates inefficiencies related to fragmented tools by offering a comprehensive, intuitive, and user-friendly solution for employees and managers. This system helps employees track tasks, manage deadlines, and monitor their performance while enabling managers to evaluate employee contributions effectively.

### 2.1. Features and Functionality

The proposed system is designed to streamline task management and enhance workplace productivity through a comprehensive suite of features tailored for both employees and managers. At its core, the platform offers secure user authentication, ensuring that only authorized personnel can access and interact with the system. By integrating essential tools such as task creation, progress tracking, automated performance evaluations, and timely reminders, the application supports efficient task handling and transparent performance monitoring. Its intuitive and user-friendly interface further simplifies navigation, enabling users to effectively manage responsibilities, stay organized, and make informed decisions based on real-time data insights.

- **User Authentication:** The system includes a secure login mechanism that ensures only authorized employees and managers can access the platform. This protects sensitive data and ensures role-based access, allowing employees to manage tasks while managers can oversee performance evaluations.
- **Task Management:** Employees can select tasks, input details such as deadlines, descriptions, and priority levels. They can update, edit, or delete tasks as needed. Completed tasks can be marked as "finished," providing clear progress tracking.

- **Performance Evaluation:** The system will include an automated performance assessment module that evaluates employees based on predefined criteria such as task completion rate, efficiency, and deadlines met.
- **Task Scheduling and Notifications:** Employees can schedule their tasks and receive automated reminders for upcoming deadlines, ensuring they stay on track and complete work efficiently.
- **Task Progress Insights:** The application will generate real-time task reports, offering employees and managers insights into completed, pending, and overdue tasks. This feature helps in better decision-making and workload distribution.
- **User-Friendly Interface:** The platform will feature a modern, easy-to-navigate interface with well-organized sections for task selection, progress tracking, and performance reports.

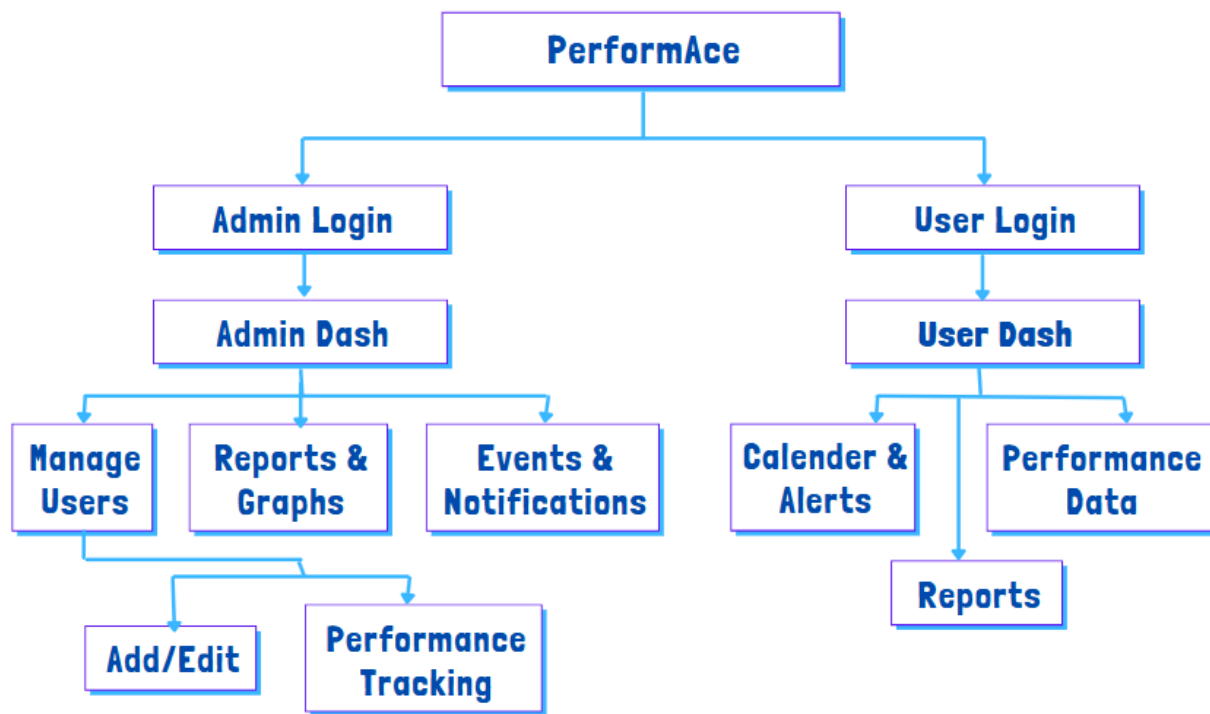


Figure 2.1: Block Diagram

The Figure 2.1 illustrates the structural block diagram of the PerformAce system, which is designed to streamline and manage employee performance through two primary user pathways: Admin and User

### **Admin Path:**

- **Admin Login:** Entry point for the administrator.
- **Admin Dash:** Central hub for administrative controls.
- **Manage Users:** Allows the admin to add or edit user details.
- **Reports & Graphs:** Displays analytics and performance metrics using visual tools.
- **Events & Notifications:** Enables scheduling via calendar integration and sets system alerts.
- **Performance Tracking:** Monitors progress and contributions of employees.

### **User Path:**

- **User Login:** Entry point for general users or employees.
- **User Dash:** Main interface for users after logging in.
- **Calendar & Alerts:** Displays upcoming tasks, events, and notifications.
- **Performance Data:** Shows user-specific performance insights.
- **Reports:** Allows users to view their own progress reports.

The block diagram presents a clear division of functionalities between admin and user roles, emphasizing real-time performance monitoring, data visualization, and efficient communication within an organization.

# Chapter 3

## Project Outcomes

The outcome of the project is the development of a fully functional Python-based Employee Performance Management application that successfully addresses challenges faced by employees and managers in managing tasks and evaluating performance. The system will enable users to efficiently handle task assignments, track progress, and receive timely notifications, thereby improving organizational efficiency. Key outcomes are as follows –

- **Enhanced user experience through a user-friendly Tkinter graphical interface** – The application boasts an intuitive layout and streamlined design, enabling effortless navigation for users across all skill levels. Whether you're a beginner or an advanced user, the interface facilitates smooth interaction and usability.
- **Efficient and real-time data visualization powered by Matplotlib** – Data is presented through dynamic charts and graphs, allowing users to instantly interpret trends, patterns, and anomalies. This visual feedback supports quicker decision-making and deeper insight into user activity.
- **Structured data collection and input tracking via Tkinter forms** – Organized and clearly labeled input fields minimize user error and ensure consistent data entry. This structure promotes reliability and improves the accuracy of collected information.
- **Automated event notifications and smart alerts** – The system proactively sends out reminders and alerts to keep users informed about upcoming tasks, deadlines,

and scheduled events. This automation helps prevent missed responsibilities and keeps users engaged.

- **Interactive scheduling and calendar-based event management with TkCalendar integration** – The application includes a fully interactive calendar feature that allows users to seamlessly view, add, and manage events. The integration simplifies planning and enhances visibility into future commitments.
- **Collaborative organization management with team creation and joining capabilities** – Users can either create or join teams within the platform, fostering a collaborative environment for shared goals, collective progress tracking, and group task delegation.
- **Secure user authentication system with robust login and signup functionality** – User data is protected through a secure authentication process, ensuring that only authorized individuals can access sensitive or personal information.
- **Increased efficiency through intelligent automation** – Routine and repetitive tasks are handled automatically by the system, significantly reducing manual effort and freeing up user time to focus on more strategic or creative activities.
- **Smarter planning with calendar tools and event reminders** – Built-in scheduling aids and timely notifications help users manage their time more effectively and ensure that no important tasks or events are overlooked.
- **Insightful analytics for performance monitoring and growth tracking** – The application provides actionable, data-driven feedback through built-in analytics. These tools support benchmarking, trend analysis, and the identification of opportunities for personal or team development.

# Chapter 4

## Software Requirements

PerformAce – The Employee Performance Management System is developed using a set of essential tools and technologies to ensure smooth performance, user-friendly interactions, and secure data handling. Below are the detailed software requirements:

- **Programming Language:** Python 3.13.3 – Core development language used for application logic, task management functions, and integration with the backend.
- **Framework & Libraries:**
  - Tkinter – Used for creating a user-friendly graphical user interface (GUI) that facilitates easy interaction for employees and managers.
  - MySQL Connector (mysql-connector-python) – Ensures secure and efficient interaction between the Python application and the MySQL database.
- **Database:** MySQL – A reliable relational database used to store employee records, performance data, and evaluation results. Supports real-time updates and data consistency for performance tracking.
- **Design Tools:** Figma (optional for UI prototyping) – Can be used for pre-development mockups and interface planning, ensuring intuitive and user-centered design.
- **Development Environment:**
  - Visual Studio Code (VS Code) – Main IDE used for writing, debugging, and testing the application. Integrated extensions for Python and MySQL improve development efficiency.
  - MySQL Workbench – A graphical tool for visualizing database schemas, executing queries, and managing relational structures.

# Chapter 5

## Project Design

This chapter focuses on the architectural and design decisions that shape the system's development, covering both the system architecture and its key components. It outlines how the design ensures scalability, usability, and efficiency in handling tasks and data.

### 5.1. User Interface Design

The user interface of the system has been carefully crafted to prioritize usability, clarity, and functional accessibility. It provides a seamless experience for both employees and managers, enabling them to interact with the system efficiently. The interface is structured into distinct sections, each tailored to a specific function—ranging from task management to performance evaluation. With an intuitive dashboard as the central hub, users can quickly view task statuses, upcoming deadlines, and performance summaries, ensuring a streamlined workflow and enhanced user experience. The user interface is designed with simplicity, efficiency, and ease of navigation in mind. Key sections include:

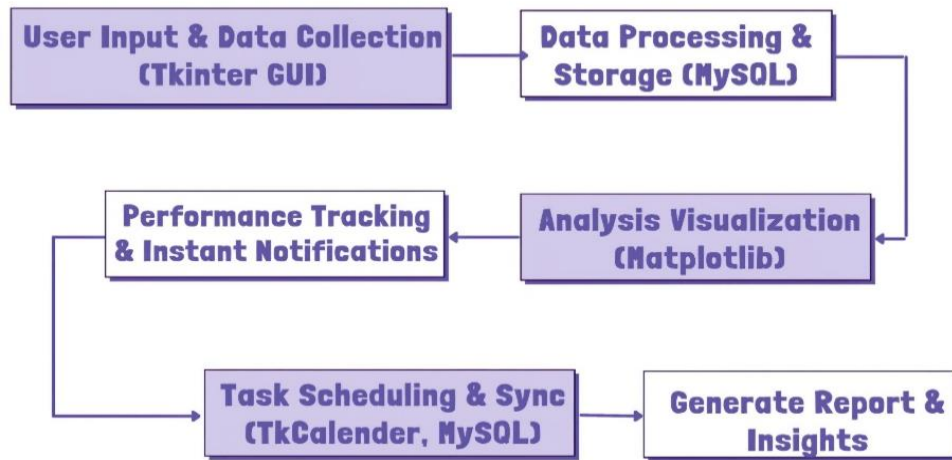
- **Dashboard:** The home page where users (employees and managers) can see an overview of assigned tasks, pending reviews, and performance metrics.
- **Task Management Page:** A structured table format displaying all assigned tasks with options to add, update progress, and mark tasks as completed.
- **Performance Review Page:** A dedicated section where managers can assess employees' performance based on predefined evaluation criteria.
- **Report Generation:** A module where employees and managers can generate and analyze performance reports over specific periods.

## 5.2. Database Design

The database architecture forms the backbone of the system, enabling reliable storage, retrieval, and management of all essential data. It is designed to ensure data integrity, scalability, and ease of access for various system operations. The database includes well-structured tables to hold user profiles, task-related information, performance evaluations, and historical reports. This organization supports smooth data flow across modules, allowing both employees and managers to perform operations such as logging in, tracking tasks, generating reports, and reviewing performance with accuracy and efficiency. The database is structured to store user information, task assignments, employee performance records, and reports. Key Tables in the Database:

- **User Table:** Stores user credentials, role (employee/manager), and contact details.
- **Task Table:** Holds details of assigned tasks, deadlines, status (pending/completed), and priority levels.
- **Performance Table:** Contains performance evaluations and feedback based on task completion and assessment criteria.
- **Reports Table:** Stores performance reports, historical data, and trends for analysis.





**Figure 5.1: Workflow**

The above figure 5.1 can be explained as following :

**User Input & Data Collection (Tkinter GUI):**

- Entry point for users to input data into the system.
- Ensures structured and intuitive data entry for performance and scheduling-related inputs.

**Data Processing & Storage (MySQL):**

- Collected data is processed and stored securely in the MySQL database.
- Enables efficient retrieval and management of user and performance data.

**Analysis Visualization (Matplotlib):**

- Transforms processed data into charts and graphs for easier understanding.
- Helps in identifying patterns, trends, and performance metrics visually.

**Performance Tracking & Instant Notifications:**

- Monitors real-time performance progress based on user activities.
- Sends instant notifications for events, deadlines, or performance alerts.

**Task Scheduling & Sync (TkCalendar, MySQL):**

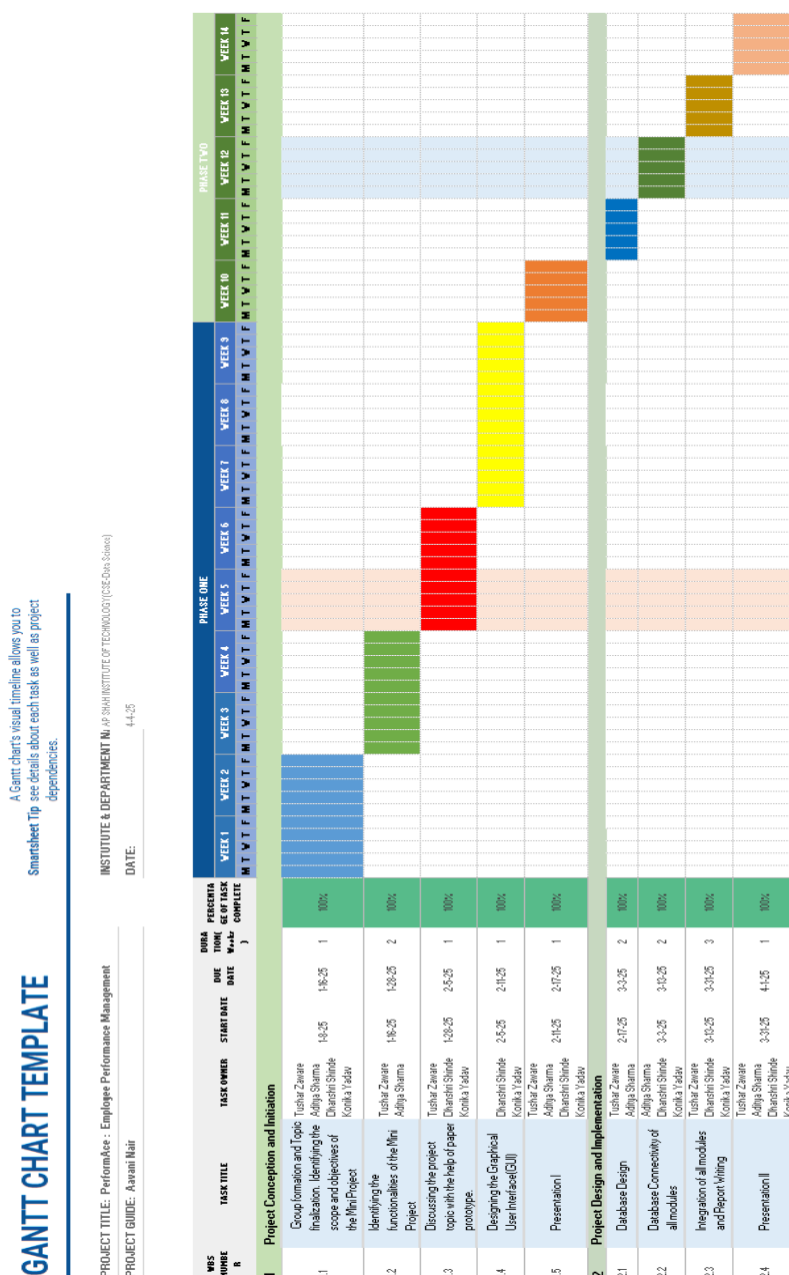
- Allows users to schedule tasks, set deadlines, and synchronize with stored data.
- Integrates with the calendar to manage and visualize event timelines.

**Generate Report & Insights:**

- Compiles collected and analyzed data into detailed reports.
- Provides insights for self-improvement, decision-making, and benchmarking.

## Project Scheduling

The project scheduling phase outlines the timeline for developing the Python application, detailing key milestones and deliverables. A Gantt chart has been employed to visualize the project timeline, providing a clear overview of tasks, their duration, and dependencies. This tool facilitates effective tracking of progress, ensuring that deadlines are met and resources are allocated efficiently.



### Figure 6.1: Gantt Chart

**Following is the detail of the Figure 6.1 (Gantt chart) –**

In the first and second weeks of January, Tushar Zaware, Aditya Sharma, Dhanshri Shinde, and Konika Yadav formed a group for the mini project. During this time, the team identified the project's topic, scope, and objectives. By the end of January, they had finalized the topic using a paper prototype and discussed the proposed ideas thoroughly.

From the first to the third week of February, Dhanshri Shinde and Konika Yadav, with support from the rest of the team, designed and developed the Graphical User Interface (GUI). The first project review was conducted on 11th February, where the team presented their progress.

Following the review, from mid-February through the end of March, the team moved into the development and implementation phase. Tushar Zaware and Aditya Sharma led the Database Design, while the database connectivity was managed collaboratively by the full team. Once the database was integrated successfully, the team focused on merging all project modules and completed the report writing.

The final presentation took place on 2nd April, marking the successful conclusion of the project, with faculty approval.

# Chapter 7

## Results

The anticipated results of this project include the successful development and deployment of a fully functional standalone Python application that effectively meets the performance management needs of employees and organizations.

The application does the following :

- Provide an intuitive and engaging user interface, enabling employees to efficiently track their tasks, deadlines, and performance metrics.
- Enhance task management, allowing employees to set, update, and mark tasks as completed, ensuring better productivity.
- Facilitate real-time performance tracking, helping employees and managers monitor progress through predefined evaluation criteria.
- Ensure reliable data persistence, leveraging MySQL to store employee performance data, task progress, and reports securely.
- Offer seamless access and updates, allowing users to view insights, reports, and performance trends through an interactive dashboard.

The implementation of this system will streamline performance tracking and reporting, ultimately enhancing organizational efficiency. The following pages have the rough explanation along with the results.

The screenshot shows a web browser window titled "Login Page". The page has a dark blue background. At the top center, the text ">> LOGIN <<" is displayed in yellow. Below this, there are two input fields: "Username" with the text "tushar" and "Password" with masked characters "\*\*\*\*\*". To the right of the password field, there are three yellow buttons: "SIGN UP", "Login", and "Close".

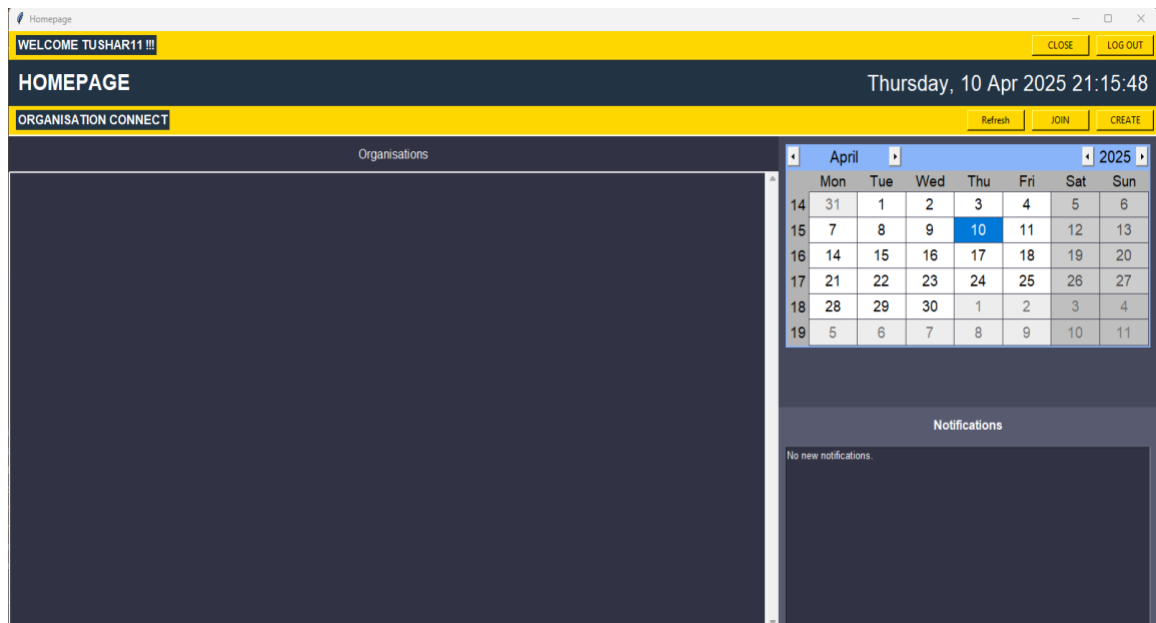
**Figure. 7.1: Login Page**

This Figure.7.1. describes the Login page which has connectivity to the credentials table in database and linked to the password reset, signup and Home page. Fetches the user details for redirecting to the homepage.

The screenshot shows a web browser window titled "Signup Page". The page has a dark blue background. It contains several input fields: "Full Name:", "Date of Birth:" (with a date picker showing "10/04/2025"), "Contact No:", "Email:", "Username:", and "Password:". At the bottom, there are two yellow buttons: "Back" and "Signup".

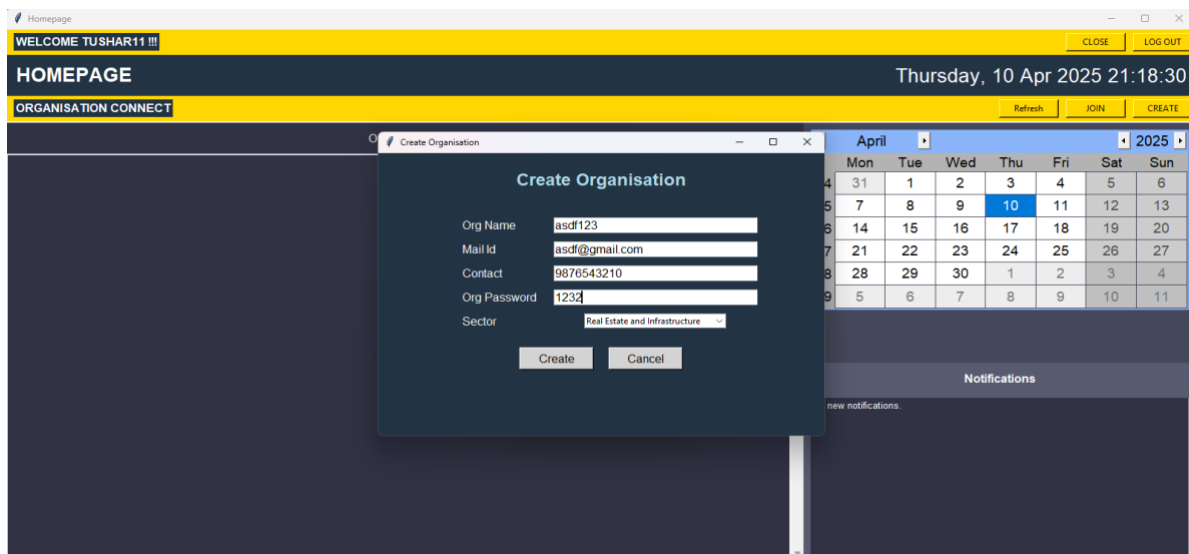
**Figure. 7.2: Signup Page**

This Figure.7.2. describes the signup page which has connectivity to the credentials table in database and linked to the password reset, signup and login page. Fetches the user details for redirecting to the loginpage.



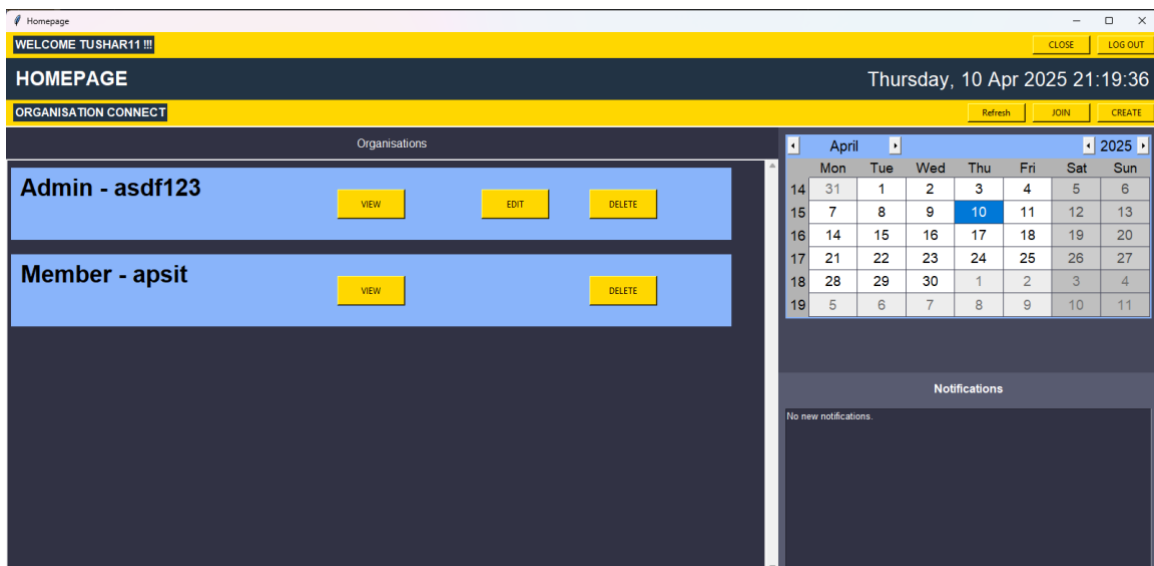
**Figure. 7.3: Home Page (before creating/joining any organization)**

This Figure.7.3. describes the home page which has connectivity to the multiple tables in database and linked to each other. Fetches the org details for redirecting to the admin/user dashboard.



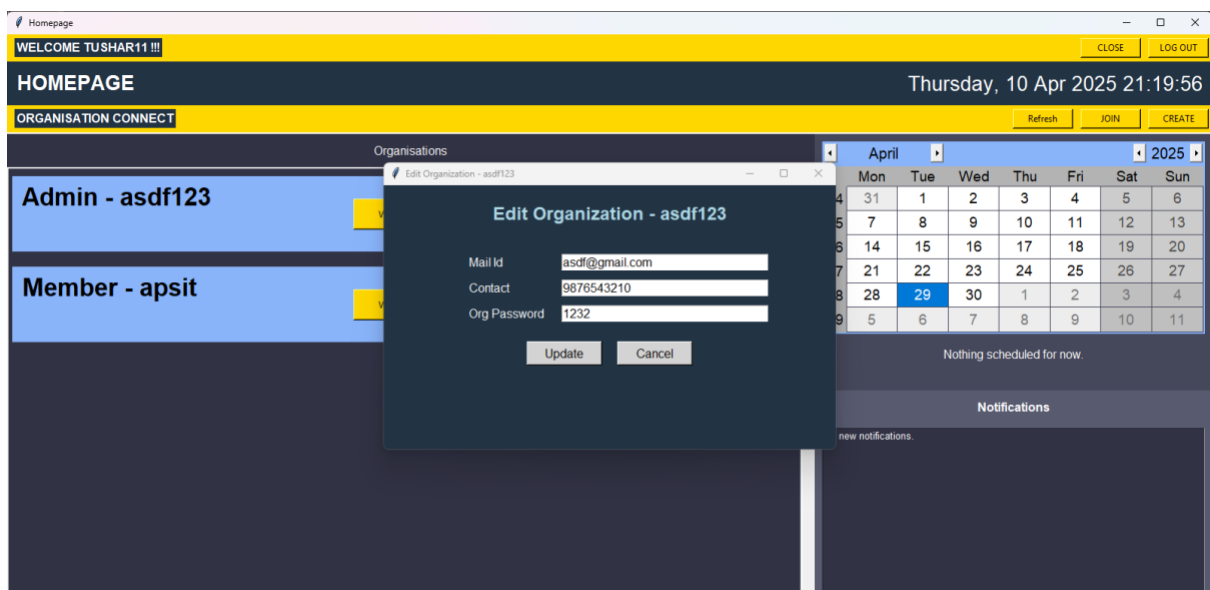
**Figure. 7.4: Create Organization Page**

This Figure.7.4. describes the create org page which has connectivity to the multiple table in database and linked to the org details and organization users.



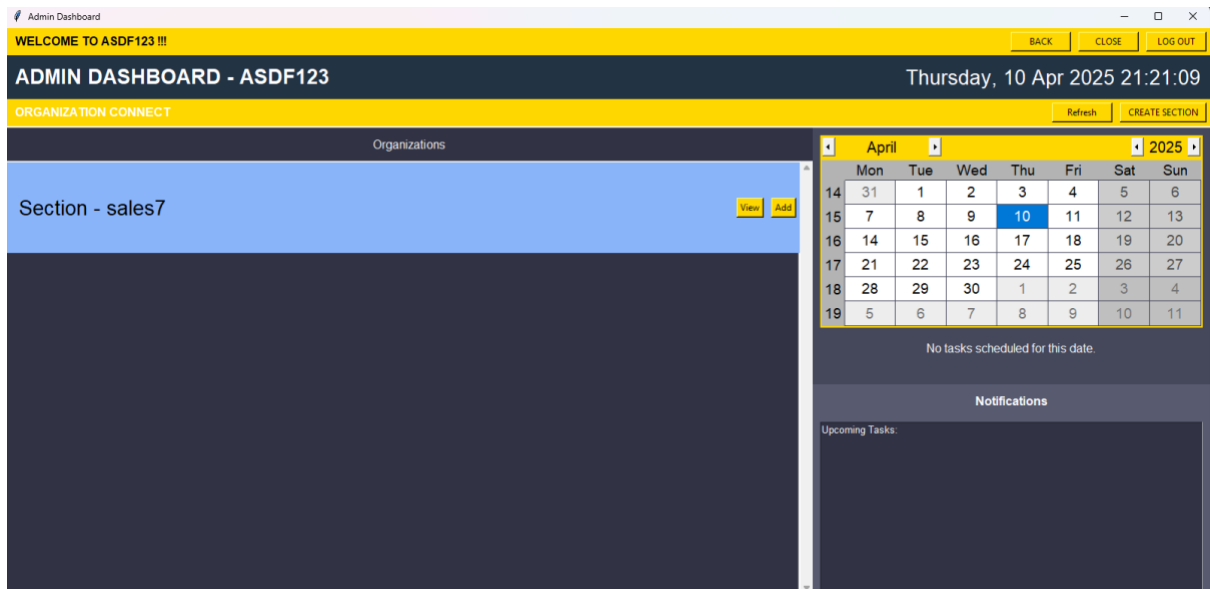
**Figure. 7.5: Home Page (after creating/joining any organization)**

This Figure.7.5. describes the home page which has connectivity to the multiple tables in database and linked to each other. Fetches the org details for redirecting to the admin/user dashboard.



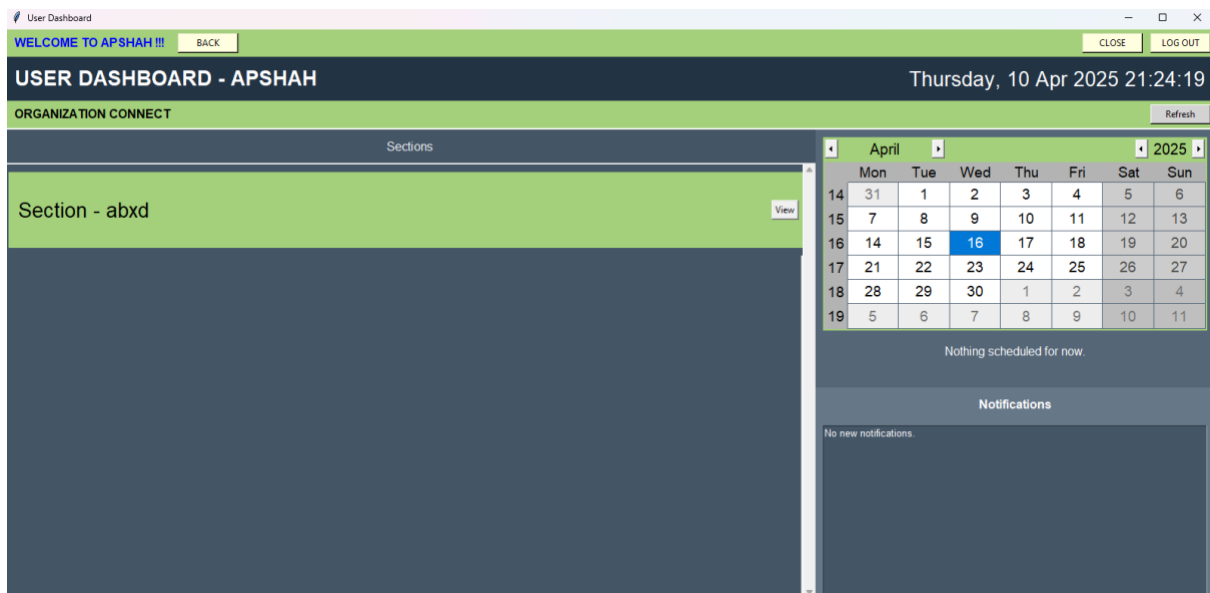
**Figure. 7.6: Edit org page (on homepage - after creating any organization)**

This Figure.7.6. describes the edit org page which has connectivity to the org and users tables in database and linked to each other. Fetches the org details for redirecting to the admin dashboard and make simultaneous changes in database.



**Figure. 7.7: admin dash Page (after creating any organization & viewing it)**

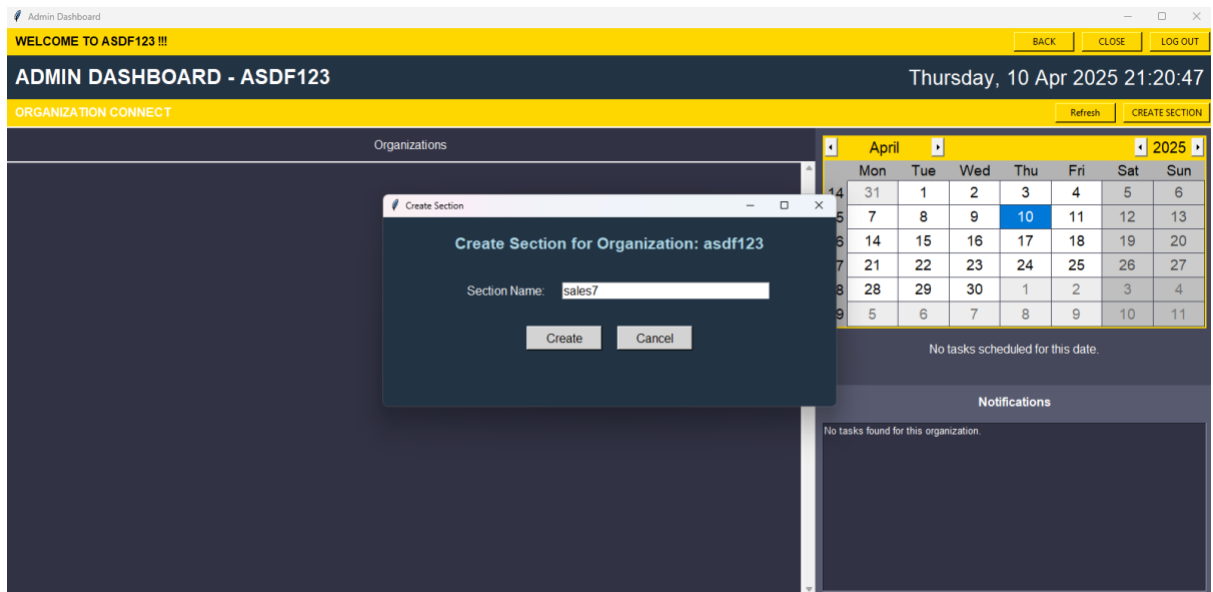
This Figure.7.7. describes the admin dash page which has connectivity to the multiple tables in database and linked to each other.



**Figure. 7.8: User Dash (After Joining Organization)**

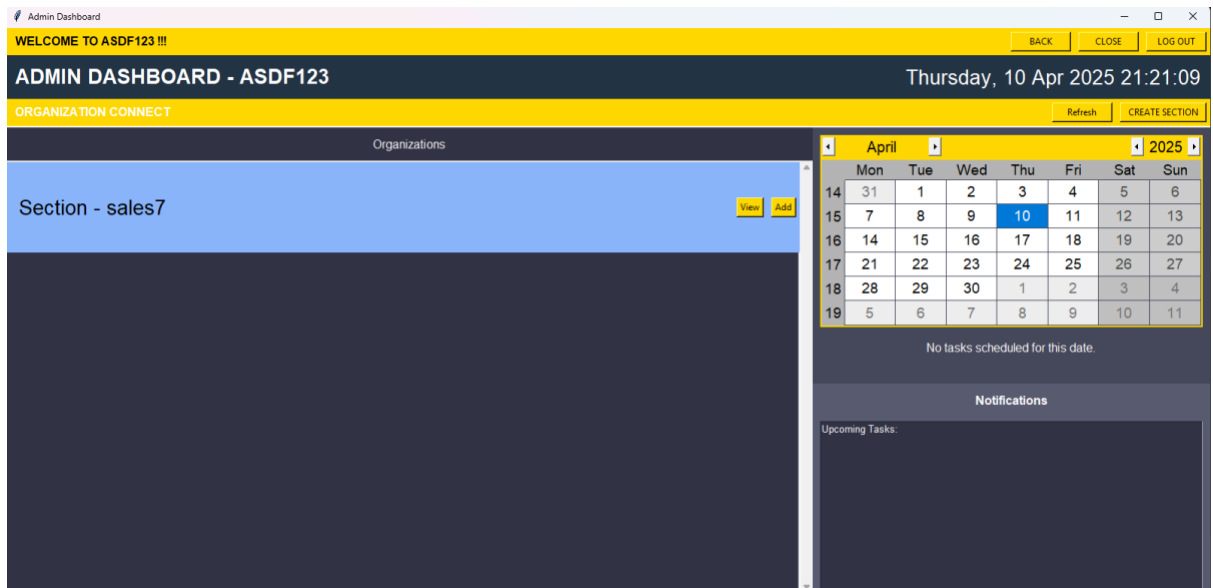
This Figure.7.8. describes the user dash page which has connectivity to the multiple tables in database and linked to each other. Fetches the org section details for redirecting to the user dashboard.





**Figure. 7.9: Create Section Page (After Creating Any Organization)**

This Figure.7.9. describes the create section page which has connectivity to the multiple tables in database and linked to each other. Fetches the org details for redirecting to the admin dashboard.



**Figure. 7.10: Admin Dash Page (After Creating Any Section & Refreshing It)**

This Figure.7.10: describes the admin dash page which has connectivity to the multiple tables in database and linked to each other.

Task Manager - asdf123 - sales7

**asdf123 - sales7**

Task Name:

Task Type:

Submission Date: 04/25/2025

Figure. 7.11: Task Page

This Figure.7.11. describes the task page which has connectivity to the multiple tables in database and linked to each other. Here tasks are created for the organizations/sections.

No.	Task Name	Task Type	File Name	Submission Date	Download All	View/Attempt
1	new1	Notes	combined_system_performance_updated.png performance_comparison_graph_updated.png system_efficiency_comparison_updated.png	2025-05-30	Download	View/Attempt
4	rdbms	Assignments	Report_Updated[1].docx combined_system_performance_updated.png	2025-05-16	Download	View/Attempt

Figure. 7.12: View/Attempt/Submission Page

This Figure.7.12. describes the view-attempt-submission page which has connectivity to the multiple tables (especially tasks) in database and linked to each other. Fetches the performance, date-time and overall details for redirecting to the previous page.

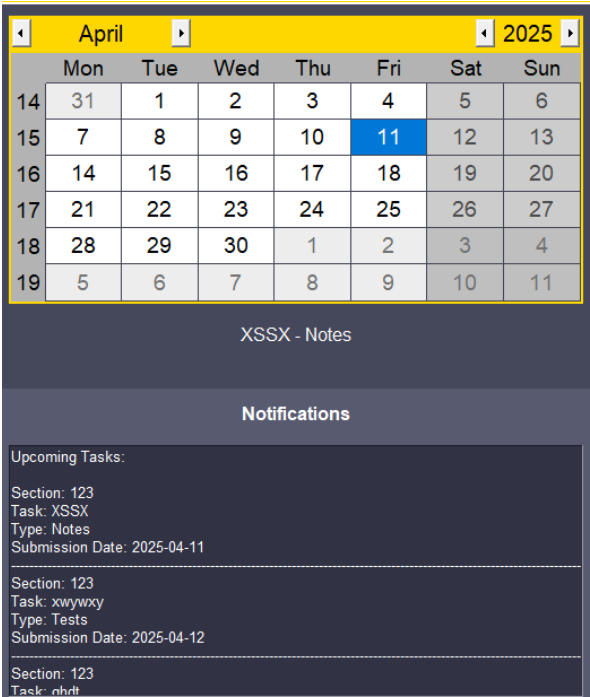


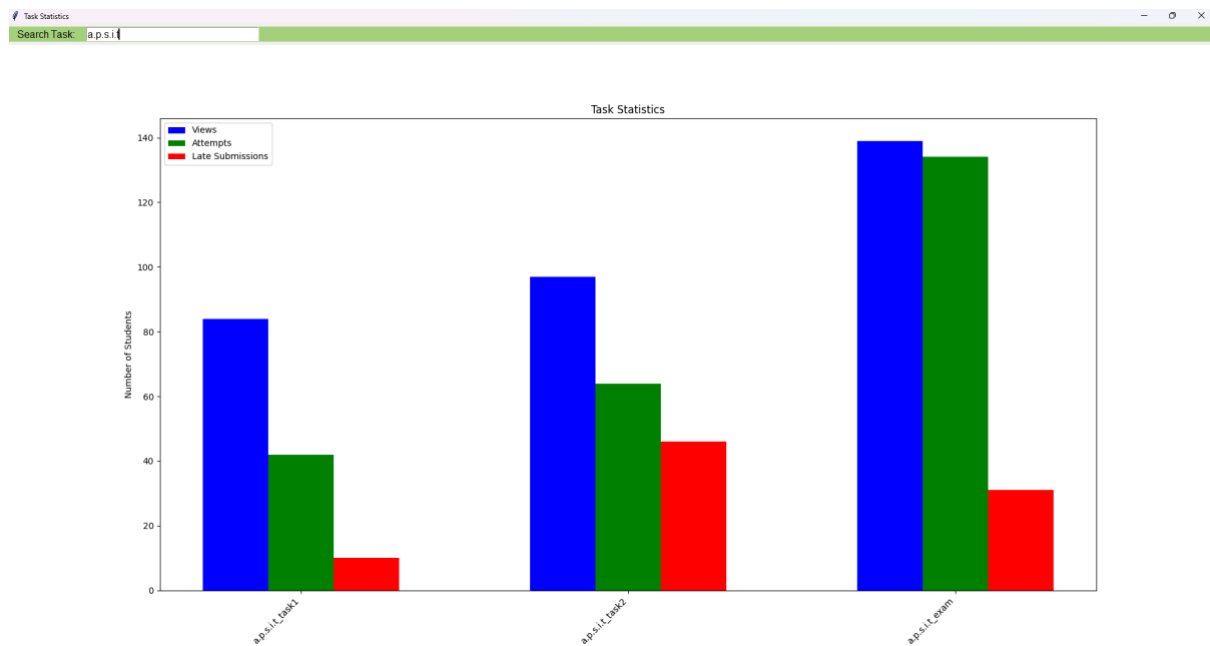
Figure. 7.13: Calendar & Notification Section (Right Panel On All Pages)

This Figure.7.13. describes the alerts section which has connectivity to the multiple tables in database and linked to each other. Fetches the user & related alerts details.

Name	Section	Task	Task Type	Start Date	End Date	Completion (%)	Rating	Remarks		
Tushar Zaveri	CSE-DS B	Group for	Planning	01-08-25	01-16-25	100	8.7	Completed on time		
Aditya Sharma	CSE-DS B	Identifyin	Analysis	01-16-25	01-28-25	100	8.5	Completed with good clarity		
Dhanshri	CSE-DS B	Designing	Design	02-05-25	02-11-25	100	8.5	GUI was appreciated		
Konika Yashwantrao	CSE-DS B	Designing	Design	02-05-25	02-11-25	100	8.5	GUI was appreciated		

Figure. 7.14: Report Page

This Figure.7.14. describes the report page which has connectivity to the multiple tables in database and linked to each other. Fetches the performance and activity details for redirecting to the graph page.



**Figure. 7.15: Report Graph Page**

This Figure.7.15. describes the graph page which has connectivity to the multiple tables in database and linked to each other. Fetches the org user activity details for redirecting to the admin dashboard.

## **Chapter 8**

### **Conclusion**

The project successfully achieved its primary goal of developing a Python-based application for employee performance management. This system integrates various performance evaluation tools, such as task tracking, performance assessments, and reporting, into a single, user-friendly platform. By eliminating manual tracking inefficiencies and offering real-time insights, the application helps organizations streamline employee performance evaluation and productivity monitoring.

The final product addresses challenges such as poor task organization, lack of timely performance feedback, and inefficient evaluation processes. With its intuitive interface and scalable architecture, the system offers room for future expansion, including cloud integration, automation of performance analytics, and AI-driven insights for better decision-making.

In conclusion, this project demonstrates the potential of Python applications in workforce management, providing an efficient and structured solution to common challenges in employee performance assessment. Future iterations could focus on mobile compatibility, real-time collaboration tools, and AI-powered recommendations for performance improvement.

## References

- [1] J. Zukowski, *Beginning Python Programming: The Object-Oriented Approach*. Apress, 2021.
- [2] L. Ramalho, *Fluent Python: Clear, Concise, and Effective Programming*. O'Reilly Media, 2022.
- [3] B. Forta, *MySQL Crash Course*. Sams Publishing, 2020.
- [4] A. Biffi, *Designing and Developing Performance Evaluation Systems: Technologies and Methodologies*. IGI Global, 2021.
- [5] S. Akhtar and M. Akhtaruzzaman, "Design and Development of an Employee Assessment System Using Python and MySQL," *Int. J. Comput. Appl. (IJCA)*, vol. 172, no. 3, 2022.
- [6] L. Bashir and M. S., "A Performance Assessment Tool Using Python and SQL: Development and Evaluation," *Int. J. Comput. Sci. Netw. (IJCSN)*, vol. 10, no. 5, 2022.
- [7] M. S. Lopes, "Developing Performance Evaluation Systems Using Relational Databases," *J. Syst. Softw.*, 2021.
- [8] S. Palaniappan and D. Rafique, "Development of a Knowledge-Based System for Employee Performance Appraisal Using Python Technologies," *Int. J. Comput. Appl. (IJCA)*, 2021.