

Mangalore University



Project Report On

“TASK MANAGEMENT SYSTEM”

Carried out and Submitted

To

**DEPARTMENT OF POST-GRADUATE STUDIES AND
RESEARCH IN COMPUTER SCIENCE**

**In Partial Fulfilment for the Award of Degree in Master of
Computer Application During the Academic Year**

2023-2024

Submitted By:

MANVITHA

P05AZ21S0052

II MCA Computer Science

Under the Guidance of:

Ass.Prof.Prakash

Professor

Department of Computer Science

Mangalore University

Department of Post-Graduate Studies and Research in Computer Science

Mangalore University

Mangalagangothri, Mangaluru – 574 199

SEPTEMBER– 2023

Mangalore University



DEPARTMENT OF POST GRADUATE STUDIES AND RESEARCH IN COMPUTER SCIENCE

Mangalore University
Mangalagangothri, Mangaluru – 574 199

CERTIFICATE

This is to certify that the project work entitled “**TASK MANAGEMENT SYSTEM**” has been successfully carried out in the Department of Post-Graduate Studies and Research in Computer Science by **Miss. MANVITHA(Reg. No: P05AZ21S0052)**, student of Fourth semester MCA, under the supervision and guidance of **Ass.Prof Mr.PRAKASH.M**, Professor, Department of Post-Graduate Studies and Research in Computer Science, Mangalore University. The project is partial fulfilment of the requirements for the award of **Master of Computer Application** by Mangalore University during the academic year **2023-2024**.

Internal Guide

Chairperson

Internal Examiner

External Examiner

Submitted for the viva-voce examination held on _____

DECLARATION

This Project work entitled “**TASK MANGEMENT SYSTEM**” has been successfully carried out by me under the supervision and guidance of **Asst.Prof Prakash.M**, Professor, Department of Post-Graduate Studies and Research in Computer Science, Mangalore University, Mangalagangotri. This project is submitted in partial fulfilment for the award in **Master of Computer Application** degree by **Mangalore University** during the academic year 2023-24. This work or any part of this work has not been submitted to any other University or Institute/School for the award of any other Degree or Diploma.

Date:

Name: MANVITHA

Place: Mangalagangotri

Reg.No: P05AZ21S0052

ACKNOWLEDGEMENT

I take this opportunity to express my sincere thanks to my supervisor **Assistant Professor Prakash M.**, Department of Post-Graduate Studies and Research in Computer Science, Mangalore University, for providing me this project and other relevant infrastructures work in the Department, for his all-round guidance, timely help at every stage of this project and for the valuable suggestions and unlimited support.

I would like to express of gratitude to **Dr. H. L. Shashirekha, Chairperson and Professor**, Department of Post-Graduate Studies and Research in Computer Science, Mangalore University, for according persimmon and for providing all kinds of infrastructure facilities in the department to carry out this project successfully.

I express my sincere thanks to **Dr. Manjaiah D. H., Professor** and **Dr.B.H.Shekar, Professor**, Department of Post-Graduate Studies and Research in Computer Science.

Last but not the least; it gives me immense pleasure and a great sense of fulfilment to thank all those who have directly and indirectly helped me in the completion of my project work.

MANVITHA

TABLE OF CONTENTS

<u>CONTENTS</u>	PAGE NO.
<u>CHAPTER 1. INTRODUCTION</u>	ERROR! BOOKMARK NOT DEFINED.-12
1.1 OVERVIEW	1
1.2 STATEMENT OF THE PROBLEM	1
1.3 MOTIVATION	ERROR! BOOKMARK NOT DEFINED.
1.4 PURPOSE	ERROR! BOOKMARK NOT DEFINED.
1.5 ADVANTAGES	4
1.6 CHALLENGES	6
1.7 OBJECTIVES	7
1.8 APPLICATIONS	7
1.9 ORGANIZATION OF REPORT	11
CHAPTER 2. ITERATURE SURVEY	13-15
2.1 LITERATURE REVIEW	ERROR! BOOKMARK NOT DEFINED.
CHAPTER 3. METHODOLOGY	16-33
3.1 DATA COLLECTION	16
3.2 _ARCHITECTURE	20
3.3 TECHNOLOGY STACK	26
3.3.1 HARDWARE_ REQUIREMENT	32
CHAPTER 4. SYSTEM DESIGN	34-39
4.1 REQUIREMENT ANALYSIS	34
4.2 DATABASE SCHEMA	36
4.3 CLASS DIAGRAM	36
4.4 ER-DIAGRAM	37
4.5 ACTIVITY DIAGRAM	39

CHAPTER 5. SYSTEM TESTING AND VALIDATION	40-41
5.1 TEST CASES	40
CHAPTER 6. EXPERIMENT AND RESULTS	42-51
CHAPTER 7.CONCLUSION AND FUTUREWORK	52

LIST OF FIGURES

Figure No.	Caption	Page No.
1.	Advantages of Task Management System	4
2.	Applications of Task Management System	10
3.	Xamp Control Panel	17
4.	Employee Table	18
5.	Task Table	18
6.	Subtask Table	19
7.	Taskhandle Table	19
8.	Employee Notification Table	20
9.	Architecture of Task Management System	20
10.	Class Diagram	36
11.	ER- Diagram	38
12.	Activity Diagram	39
13.	. Login Page	42
14.	subtask Assigned to Employee	42
15.	subtask is completed	43
16.	Employee personal Data	43

17.	send Message Request	44
18.	Admin Dashboard	44
19.	Number of subtask under each task	45
20.	All Added Task	45
21.	Add Task	46
22.	All Employee Details	46
23.	Add Employee	47
24.	Task Assigned Employees Details	47
25.	Assign Employee To Task	48
26.	Subtask Details	48
27.	Add Subtask	49
28.	Assign Subtask	49
29.	Assign Subtask to employee	50
30.	Task Report	50
31.	Remaining Employee	51

ABSTRACT

Effective task management is the key to organizational success in today's dynamic and cutthroat corporate environment. A strong Task Management System (TMS) is essential due to the ever-increasing complexity of projects and the requirement for smooth team cooperation. There are always a ton of tasks, projects, and deadlines at work today. By offering structure, clarity, and speed, the Task Management System acts as a digital compass, guiding businesses through the maze of assignments. It promotes effective teamwork and communication even in circumstances where individuals are spread out geographically.

Task Management System reduces bottlenecks by establishing a transparent environment where task owners are accountable for the tasks they perform. A task management system that is effective and efficient runs per the core ideas of "Getting Things Done." This includes a thorough approach to job lifecycle management that includes processes for planning, testing, tracking, and reporting. The essence of a task management system resides in its capacity to support individual users in reaching their goals and promote group collaboration, thus enabling the achievement of common objectives. This system serves as the foundation for efficient workflows within an organization and is essential to project management and process advancement. The project schedule offers dynamic and up-to-date information for project managers who use task-oriented methods of management, giving them the ability to effectively lead their teams and advance projects.

CHAPTER-1

INTRODUCTION

1.INTRODUCTION

1.1 Overview

Task Management Systems (TMS) are essential tools in today's fast-paced and increasingly digital world. These systems are designed to help individuals and teams efficiently organize, track, and complete tasks and projects. Here's an overview of what a Task Management System is and why it's vital in various aspects of our personal and professional lives.

A Task Management System is a software application or platform that allows users to create, organize, assign, prioritize, and track tasks and activities. It provides a structured approach to managing work, ensuring that tasks are completed on time, and nothing falls through the cracks. In today's fast-paced and highly competitive business environment, effective task management is the cornerstone of organizational success. A Task Management System (TMS) is a digital solution that has emerged as an indispensable tool for individuals, teams, and entire enterprises to efficiently plan, execute, and monitor tasks and projects. The Task Management System project seeks to solve these critical problems by developing a robust and intuitive platform. By doing so, it aims to enhance productivity, reduce stress, improve collaboration, and empower individuals and organizations to achieve their goals effectively and efficiently in an increasingly complex work environment.

1.2 Statement of the Problem

In today's rapidly evolving world of work, effective task management is essential for individuals, teams, and organizations. The ability to efficiently plan, execute, and monitor tasks and projects is fundamental to productivity and goal attainment. However, despite the advances in technology, many individuals and organizations continue to face significant challenges and pain points in the realm of task management.

- **Disjointed Task Tracking:** A prevalent problem is the use of disconnected tools and systems for task tracking. Some individuals rely on physical to-do lists, others on digital calendars, and many use a combination of email, spreadsheets, and sticky notes. This fragmented approach often results in tasks slipping through the cracks, duplication of efforts, and confusion regarding task status.
- **Deadline Management:** Meeting deadlines is a critical aspect of task management. However, without automated deadline notifications and a centralized system for tracking due dates, individuals and teams frequently struggle to prioritize tasks effectively, leading to missed deadlines and project delays.
- **Task Assignment and Ownership:** Assigning tasks to the right individuals with clear ownership can be challenging. Without a structured system for task assignment, there is often ambiguity regarding who is responsible for a particular task, which can result in delays and accountability issues.
- **Data-Driven Decision-Making:** In the absence of a data-driven approach to task management, organizations miss out on valuable insights. They can't to analyse task completion rates, identify bottlenecks, and make informed decisions to optimize workflows and resource allocation.
- **Stress and Productivity Loss:** Inefficient task management processes often lead to high levels of stress among employees. The constant struggle to keep track of tasks and meet deadlines can have detrimental effects on mental well-being and overall productivity.

1.3 Motivation

In a world driven by rapid technological advancements and increasing demands, the need for effective task management has never been more critical. The motivation to develop a comprehensive Task Management System stems from a

deep understanding of the challenges individuals and organizations face in managing their tasks and projects. The primary motivation is to boost productivity. We recognize that inefficient task management leads to wasted time and effort. By streamlining the task management process, we aim to help individuals and teams accomplish more in less time. Stress and burnout are prevalent in today's fast-paced work environment. Many individuals struggle with overwhelming to-do lists and competing priorities. A well-designed Task Management System can alleviate this stress by providing clarity, structure, and support in managing tasks.

Meeting deadlines is crucial for personal and professional success. Our motivation is to create a system that empowers users to plan their tasks effectively, prioritize deadlines, and receive Date reminders, reducing the incidence of missed deadlines. the motivation behind developing a Task Management System is rooted in the belief that effective task management is a cornerstone of success and well-being. By addressing the challenges faced by individuals and organizations, we aim to contribute to a more productive, less stressful, and highly efficient work environment.

1.4 Purpose

The purpose of a Task Management System (TMS) is to provide a structured and efficient framework for planning, organizing, executing, and monitoring tasks and projects. TMS serves as a digital assistant, helping individuals, teams, and organizations streamline their work processes and achieve a wide range of objectives. empower users and organizations with the tools and capabilities needed to manage tasks efficiently, collaborate effectively, meet deadlines consistently, and ultimately achieve their goals in an organized and stress-free

manner. It serves as a digital ally in the fast-paced and demanding landscape of modern work.

1.5 Advantages

Task Management Systems offer a wide array of benefits for individuals, teams, and organizations. These advantages streamline workflow, boost productivity, and contribute to overall efficiency.

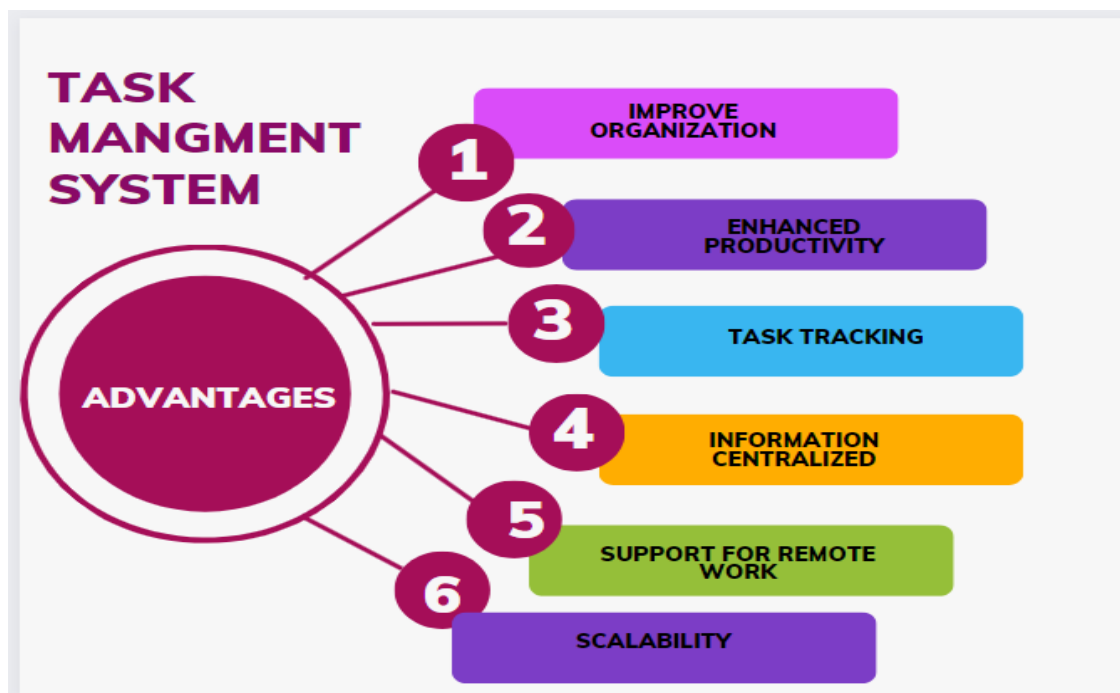


Figure 1. Advantage of Task Management System

- **Improved Organization:** Task Management Systems provide a structured framework for organizing tasks. Users can categorize tasks, set priorities, and allocate resources efficiently. This organization ensures that nothing falls through the cracks, leading to higher productivity.
- **Enhanced Productivity:** Task Management Systems enable individuals and teams to focus on essential tasks. They can easily identify high-priority items and allocate time and resources accordingly. This increased focus on critical work results in enhanced productivity.

- **Effective time management:** Scheduling and time tracking functions are frequently found in task management systems. Effective time management is encouraged by the ability of users to set deadlines, issue reminders, and allocate particular time slots for tasks.
- **Task tracking:** These programs give users access to up-to-the-minute task information. Team members may monitor task progress, spot bottlenecks, and guarantee that projects stay on schedule
- **Task Delegation:** Delegating tasks is made easier with the use of task management systems. Team members can be given tasks by managers, who can also monitor performance and set clear expectations. This delegation makes sure that work is carried out effectively and allocated equally.
- **Information centralized:** All task-related data is kept in a single location. It is now possible to view and change task details without using haphazard notes, emails, or spreadsheets.
- **Flexibility:** These solutions are adaptable to a range of businesses and work processes. They can be tailored to meet particular demands whether they are used for project management, customer support, or personal task management.
- **Support for Remote Work:** Task Management Systems are helpful for remote work. Team members are crucial to distributed teams because they can access work and collaborate from any location with an internet connection.
- **Scalability:** Task Management Systems may grow alongside enterprises as they do. They maintain performance while supporting an expanding range of projects, users, and tasks.

- **Data Insights:** Task Management Systems often offer analytics and reporting features. Users can gain insights into task completion rates, identify areas for improvement, and make data-driven decisions to optimize processes.

1.6 Challenges

Task Management Systems have revolutionized the way we handle work, but they are not without their challenges.

- **User Adoption:** Getting everyone in a team or organization to consistently use the system can be challenging. Resistance to change or a preference for traditional methods can hinder user adoption.
- **Data Security:** Task Management Systems store sensitive data, which makes them attractive targets for cyberattacks. Ensuring robust security measures is crucial to protect this information.
- **Information Overload:** An excessive number of tasks, subtasks, and details can lead to information overload. Users may struggle to filter and focus on essential tasks.
- **Task Duplication:** Users may inadvertently create duplicate tasks, leading to confusion and inefficiency. Duplicate tasks can clutter the system and make tracking challenging.
- **Data Backup and Recovery:** Ensuring data backup and recovery mechanisms are in place is crucial. Data loss can have severe consequences if not adequately addressed.
- **Task Prioritization:** Users may struggle with task prioritization. Failing to prioritize effectively can lead to missing deadlines or neglecting critical tasks.

- **Cost Management:** The cost of implementing and maintaining Task Management Systems, especially for larger organizations, can be a concern.

1.7 Objectives

Task Management Systems are designed to streamline work processes, improve efficiency, and enhance collaboration. The objectives of implementing a Task Management System within an organization or for individual use are multifaceted:

- **Efficient Task Handling:** The primary objective is to efficiently manage and handle tasks from creation to completion. This includes task assignment, tracking, and monitoring progress.
- **Helping users prioritize tasks based on urgency and relevance** is a crucial goal, ensuring that crucial jobs get the attention they require.
- **Meeting deadlines** is made easier by task management systems' reminders, due dates, and notifications of upcoming tasks.
- **Workflow streamlining:** These systems' automation capabilities attempt to make workflows more efficient while eliminating errors and manual labor.
- **Reporting and analysis:** Reporting and data analysis are important for assessing performance, finding bottlenecks, and making well-informed decisions.

1.8 Applications

Task Management Systems find application in various sectors and industries due to their versatility and effectiveness in organizing and streamlining work processes. Below are some key applications of these systems:

- **Project management:** To plan, track, and finish activities within a project, task management systems are frequently employed. They support project managers in setting priorities, allocating resources, and ensuring that deadlines are met.
- **Time Management:** In today's fast-paced environment, effective time management is essential. Task management tools let users allocate time to certain projects, prevent overcommitting, and maintain a positive work-life balance.
- **Customer Relationship Management (CRM) systems:** CRM systems frequently have task management tools to aid sales and support staff in effectively managing customer contacts, follow-ups, and appointments.
- **Content Creation:** The management of content production projects, editorial calendars, and publication schedules is handled by content creators using these platforms, including writers, designers, and marketers.
- **Team Collaboration:** Teams from many departments and locations work together on tasks and projects using these systems. They offer a centralized system for assigning tasks, sharing files, and team communication.
- **Personal Productivity:** To plan their days, set objectives, and prioritize their work, people use task management systems. Users who use these devices report increased concentration and productivity.
- **Team Collaboration:** Teams from many departments and locations work together on tasks and projects using these systems. They offer a centralized system for assigning tasks, sharing files, and team communication.

- **Personal Productivity:** To plan their days, set objectives, and prioritize their work, people use task management systems. Users who use these devices report increased concentration and productivity.
- **Manufacturing and Production:** Task Management Systems support the scheduling of production tasks, the oversight of equipment maintenance, and the achievement of production goals in the manufacturing sector.
- **Healthcare:** These systems are used by healthcare practitioners to organize patient appointments, treatment schedules, and medical records. They contribute to ensuring the effectiveness of healthcare services.
- **Education:** Educational institutions utilize task management to schedule classes, assignments, and exams. These systems aid teachers in managing course materials and student assessments.
- **Event Planning:** Task Management Systems are used by event planners to manage guest lists, handle vendor contracts, and organize event logistics.
- **Sales and marketing teams** utilize these tools to handle leads, organize campaigns, and keep tabs on sales activities. They make sure that marketing and sales objectives are achieved.
- **Financial Management:** To handle budgeting, financial reporting, and audits, finance departments use task management. These programs support compliance and financial planning.
- **Legal Services:** To keep track of case deadlines, court appearances, and document files, law firms and legal practitioners employ task management. It aids in keeping precise legal records.
- **Government and Nonprofit:** Task Management Systems are used in government and nonprofit organizations to manage projects, keep track of grants, and provide public services.
- **Research and development:** These systems are used by research teams to coordinate experiments, data gathering, and project milestones.

- **Customer Support:** To manage customer requests, grievances, and support ticket assignments, customer support teams rely on task management. It guarantees prompt problem resolution.
- **Field Service Management:** Field service businesses utilize these systems to assign jobs to field representatives, track job progress, and boost the effectiveness of their services.
- **Supply Chain and Logistics:** Task management helps with inventory control, order fulfillment, and supply chain coordination, ensuring that goods reach customers on schedule.

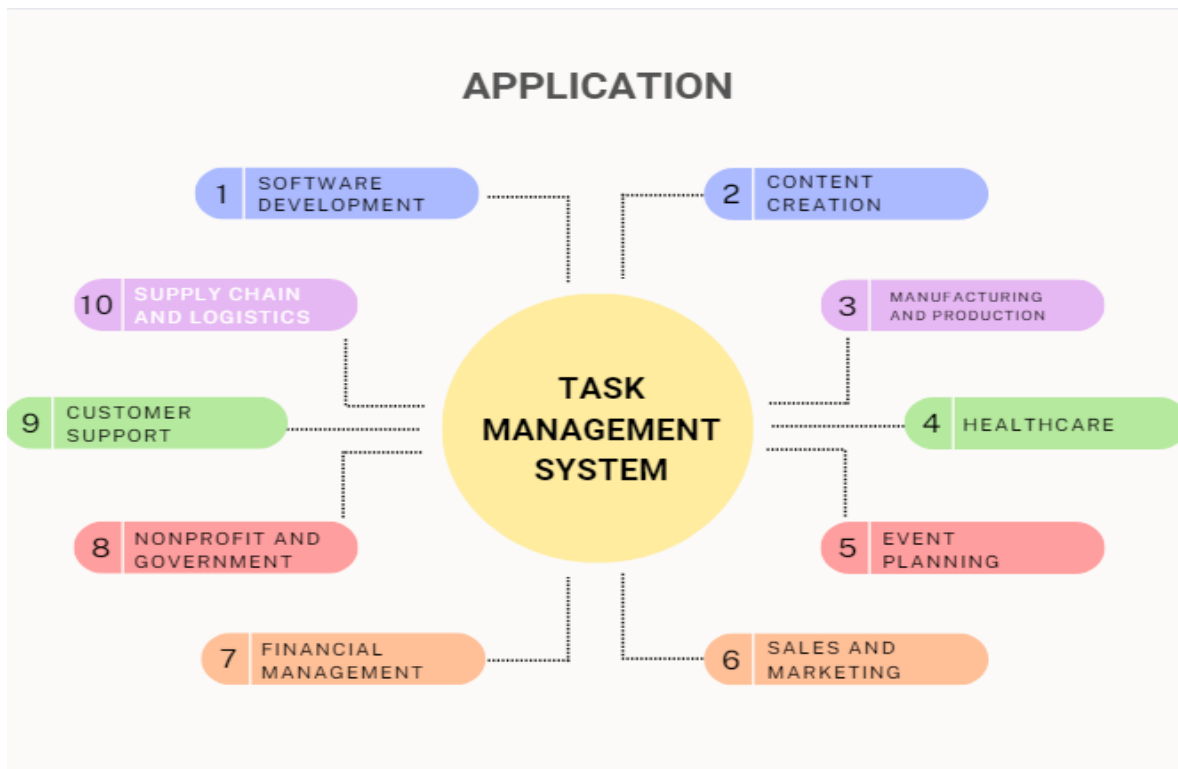


Figure 2. Applications of Task Management System

1.9 Organization of Report

Abstract:

A brief summary of the entire report, including the problem statement, methodology, key findings, and recommendations.

Table of Contents:

List of sections, subsections, and their respective page numbers for easy navigation.

List of Figures and Tables:

If applicable, list all figures and tables used in the report with their corresponding page numbers.

Chapter-1. Introduction:

Background and context of the Task Management System project.

Statement of the problem or need for the system.

Objectives and scope of the report.

Overview of the report structure.

Chapter-2. Literature Review:

Review of relevant literature and studies related to task management systems. Explanation of existing systems and their advantages and limitations. Identification of best practices in task management.

Chapter-3. Methodology and Requirements:

Description of the research methods and tools used to analyze and implement the Task Management System.

Details of data collection, surveys any experiments conducted.

Information on the software or technology stack used for system development.

Chapter- 4. System Design:

User interface design and wireframes. Database schema and system flowcharts. Integration with existing systems

Chapter-5. System Testing and Validation:

Details of how the system was tested for functionality, performance, and security.

Test cases and results.

Validation against user requirements and expectations.

Chapter-6.Experiments and Results:

Presentation of the results obtained from the Task Management System.

Data analysis, graphs, and visual representations.

Discussion of key findings and their implications.

Chapter-7. Conclusion and Future work:

Summarization of the report's main points.

Restate the significance of the Task Management System.

Concluding remarks on the project's success and impact.

Chapter-8. References:

List of all sources cited in the report.

CHAPTER-2

LITRATURE REVIEW

2.Litrature Survey

A "Literature Survey" or "Literature Review" is a critical examination and evaluation of existing literature, research, and scholarly works relevant to a particular topic, research question, or field of study. It is a systematic and comprehensive analysis of the knowledge and studies that have been previously published or conducted in a specific area.

Online Task Management System (OTMS), GRISHMA HEDAOO¹, PRIYANKA THOKE², RAKSHA TABHANE³, SHUBHAM MESHARAM⁴, SWAPNIL KUMBHALKAR⁵, PROF. MUKESH BARAPATRE⁶ Department of Information Technology and Engineering, S. B. Jain Institute of Technology, Management and Research, Nagpur, India.[1]:

The system easily assigns tasks so as to avoid all the time-consuming and unnecessary meetings. Teacher can periodically share all the details regarding the assignment with the Student. The management of assignment or task is easy from both end. The teacher is able to assign task to the student. The task will observe and controlled by the Head of Department. This software provides facilities to assign task, send message, send and view notification to the users of this software. HOD maintains a database system and is responsible for the performance of the system. HOD have authority to control the entire system. IT Infrastructure is privileged user who overlook over Task module. The purpose for developing this project is to generate a platform where Head of Department can do operations like assign, add users and view operations. HOD, IT Infrastructure and Teachers are privileged users who can add/modify a task. Online task management software as it enables users to organize and prioritize projects efficiently and with flexibility. The Online task manager Software starts with the task and end up with checkmarks, the deadline and track the works will be notified.

A Framework for Web Based Student Record Management System using PHP Er. Saurabh Walia¹, Er. Satinderjit Kaur Gill[2]:

the framework is created by utilizing Xampp Server interfacing with database that is using 'PHP' language as the dialect or guideline of the framework. The

proposed framework is a standalone framework. This framework centered on recording and updating the information. It is additionally given report on the other hand printed record to the client in the framework which will make the status of the student simpler to be checked. This new framework utilized database idea to store all the data which related with area application forms. This framework focused around database idea which is more solid. All learner information will be kept in a devoted database.

Web based chat Application using React Ajay Sahu, Vishesh Vishwakarma, Shubham Yadav, Shubham Yadav, Greater Noida Institute of Technology (GNIT) Date Written: February 22, 2023,[3]

In this research paper, we have mentioned a react-based web chat application and also we have compared react based chat application with another technology " PHP "that will help us to understand which technology is suitable for developing chat applications. There's no need for third-party client software with this web-based real-time chatting tool. The app is built with React. js, and Node.js with the Express package, as well as a Mongo DB database. Text messages are sent to and received from the server, and via a point-to-point connection between the servers. Transfer of data from is enhanced. The concept of virtual environments has been adopted due to the use of the React framework, which increases performance by a factor of about 6 over traditional PHP-based apps

Evaluation System for Software Testing Tools in Complex Data Environment[4]

popularization of the Internet in recent years, the software industry is booming. In order to develop software products to meet the needs of users, software testing must be carried out. However, due to the complexity of the test environment, software testing faces great challenges. Therefore, an efficient system of test and evaluation is necessary. Unfortunately, there is no single tool that can fully test complex data at present. It is of great importance to select test tools for the reason that different test tools have different effects on the efficiency of integration

testing. Therefore, this paper constructs a test and evaluation system based on different data environments. It helps testers to choose suitable test tools which include three commonly test software, Postman, JMeter, and Robot Framework. The performance of those test tools through the evaluation methods of standard deviation and incidence matrix. The experimental results show that the comprehensive performance of Postman is the better than JMeter and Robot Framework.

Modern Web-Development using ReactJS[5]

ReactJS is JavaScript library which is deployed to develop reusable user interface (UI) components. According to React official documentation, following is the definition React is a library for building modular user interfaces[9] . React basically enables development of large and complex web based applications which can change its data without subsequent page refreshes. It is used as the View (V) in the Model-View-Controller(MVC). React abstracts the Document Object Model (DOM), thus offering a simple, performing and robust application development experience. React mostly renders on server side using NodeJS, and support for native mobile apps is offered using React Native. React implements unidirectional data flow thus simplifying the boilerplate and hence proves to be much easier than traditional data binding.

CHAPTER-3

METHODOLOGY

3.Methodology

3.1 Data Collection

XAMPP (which stands for Cross-Platform, Apache, MySQL, PHP, and Perl) is a free and open-source web server solution stack. It's primarily used for local development and testing of web applications. XAMPP itself doesn't handle data collection or storage directly; instead, it provides the necessary environment to run web applications that can handle data collection and storage.

XAMPP components play a role in data collection and storage:

- **Apache (Web Server):** Apache is responsible for serving web pages and handling HTTP requests. When a user interacts with a web application, Apache receives the request and passes it to the appropriate script or application that collects or processes data. This data can be collected from HTML forms, APIs, or other sources.
- **MySQL (Database Management System):** MySQL is a relational database management system (RDBMS) that can be used to store and manage structured data. Most web applications rely on databases to store user data, content, and application information. You can use MySQL, included in XAMPP, to create and manage databases for your web applications.
- **PHP and Perl (Server-Side Scripting Languages):** PHP and Perl are scripting languages used for server-side development. They are often used to create web applications that interact with databases (like MySQL) to collect, process, and store data. These languages are essential for building the logic that handles data on the server-side.
- **Data Will Be Store in Tables inside Database.**
 1. **Database :**A database is a structured collection of data. It is designed to efficiently manage, store, retrieve, and update information.
 2. **Table:** A table is a fundamental structure in a relational database. It is also known as a relation. Tables are used to organize data into rows and columns. Each row represents a single record, while each column represents a data field.

3. **Primary Key (PK):** A primary key is a field or a set of fields in a table that uniquely identifies each record (row) in the table. It enforces the entity integrity constraint, ensuring that there are no duplicate rows in the table. Typically, the primary key is used as a reference point for creating relationships between tables. An example of a primary key is an "ID" field, which contains a unique identifier for each record.

4. **Foreign Key (FK):** A foreign key is a field in a table that is used to establish a link between two tables. It creates a referential integrity constraint, ensuring that the values in the foreign key column match the values in the primary key of another table. Foreign keys are used to create relationships between tables, defining how data in one table relates to data in another.

5. **Relational Database:** A relational database is a type of database that uses a tabular structure to organize data. It stores data in tables with rows and columns. Relationships between tables are established using primary and foreign keys. The relational database model is based on mathematical set theory and provides a way to efficiently manage and query structured data.



Figure 3. Xamp Control Panel

Employee Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(50)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 taskId	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/>	3 employeeName	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 employeeEmail	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 employeeRole	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	6 employeePassword	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	7 employeeContact	bigint(30)			No	None			Change Drop More
<input type="checkbox"/>	8 empUid	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

☐ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Move columns Normalize

Figure 4. Employee Table

Task Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 taskName	varchar(80)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 taskDescription	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 taskAssignDate	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 taskEndDate	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	6 taskPriority	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	7 finalTaskComplete	varchar(30)	utf8mb4_general_ci		No	Not Completed			Change Drop More
<input type="checkbox"/>	8 taskCompletedPercentage	int(11)			No	None			Change Drop More

☐ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after taskCompletedPercentage Go

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	2	A	No	

Create an index on 1 columns Go

Figure 5. task Table

Subtask Table:

Server: 127.0.0.1 » Database: taskmanagementsystem » Table: subtask

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	taskId	int(11)			No	None			Change Drop More
3	employeeid	int(11)			Yes	NULL			Change Drop More
4	Name	varchar(80)	utf8mb4_general_ci		No	None			Change Drop More
5	Description	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
6	startButton	varchar(10)	utf8mb4_general_ci		No	false			Change Drop More
7	completeButton	varchar(10)	utf8mb4_general_ci		No	true			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after completeButton Go

Indexes

Figure 6. Subtask Table

Taskhandle Table:

Server: 127.0.0.1 » Database: taskmanagementsystem » Table: taskhandle

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	taskId	int(11)			Yes	NULL			Change Drop More
3	subtaskId	int(11)			Yes	NULL			Change Drop More
4	employeeid	int(11)			Yes	NULL			Change Drop More
5	empName	varchar(80)	utf8mb4_general_ci		No	None			Change Drop More
6	taskAssignDate	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
7	assignedTaskEndDate	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
8	taskStartedDate	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
9	taskCompletedDate	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
10	status	varchar(80)	utf8mb4_general_ci		No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Figure 7. Taskhandle Table

Employee Notification Table:

Table structure for **empnotification**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	emplid	int(11)			No	None			Change Drop More
3	Date	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
4	sentMessage	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	9	A	No	
Edit Rename Drop	emplid	BTREE	No	No	emplid	4	A	No	

Figure 8. Employee Notification Table

3.2 Architecture

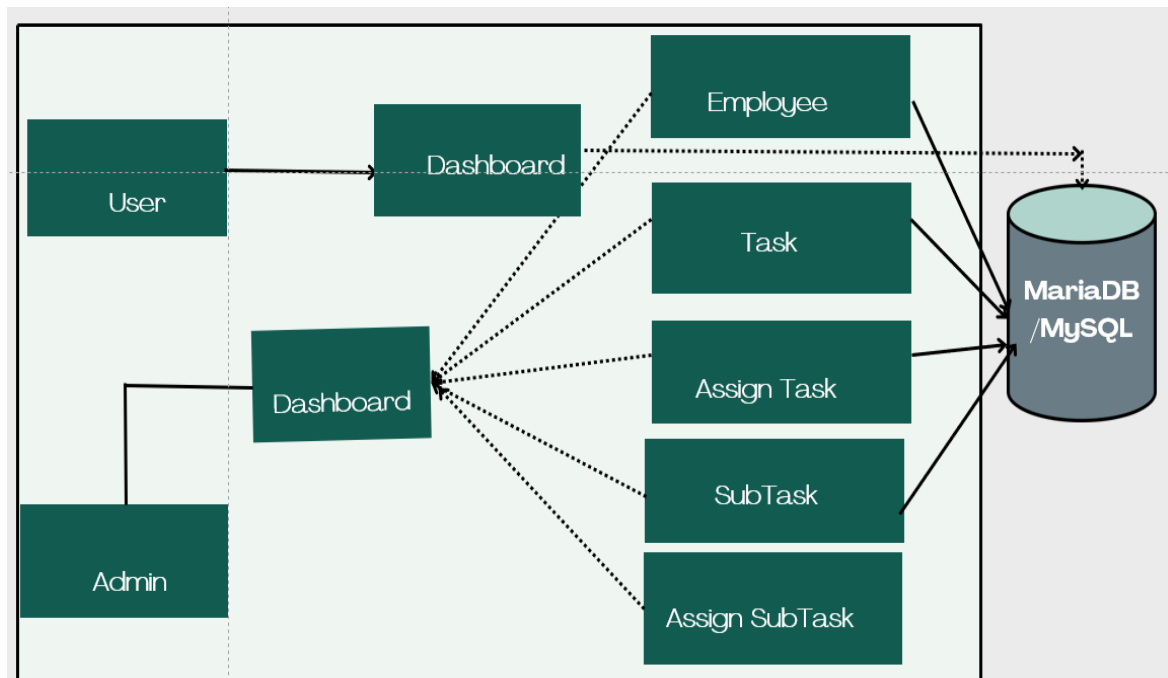


Figure 9. Architecture of Task Management System

User:

Employee one who is part of the organization is added by the admin and Email and password is given by the admin using that employee can login.

The Employee Side Dashboard is a secure web-based platform that provides access to specific features and information tailored for employees within an organization. It operates on a role-based access control system, where the user's role is determined during the login process. Here's a detailed explanation of how it works:

1. User Authentication:

When an employee attempts to log in, the system verifies their identity using a combination of their email and password. These login credentials are unique to each user and serve as a means of authentication.

2. Role-Based Access Control (RBAC):

RBAC is a system that grants or restricts access to specific features and data based on a user's role within the organization.

In this case, there are different roles, and one of them is the "Employee" role.

3. Role Matching:

After successful login, the system checks the user's role. If the user's role is identified as "Employee," they are granted access to the Employee Side Dashboard.

Employee Side Dashboard:

The Employee Side Dashboard is a user interface designed specifically for employees. It provides a user-friendly and organized view of relevant information and tasks.

1. View Task Assignment :

Within the Employee Side Dashboard, employees can view tasks assigned to them. These tasks may include various responsibilities or assignments they need to complete.

2. Subtask Details:

For each task, employees can access additional details, including subtasks related to that task. Subtasks are smaller components or steps that contribute to the completion of the main task.

The Employee Side Dashboard serves multiple purposes:

It centralizes task management for employees, making it easier to track and complete assignments.

It provides transparency by allowing employees to see what tasks they need to work on and their progress.

It improves efficiency by breaking down larger tasks into manageable subtasks.

It helps employees prioritize their work and meet deadlines.

It can include features such as task deadlines, progress tracking, and communication tools to facilitate collaboration.

Data Security:

The system ensures data security by requiring authentication before granting access to the Employee Side Dashboard.

Access is restricted to authorized users with the "Employee" role, preventing unauthorized access to sensitive information.

In summary, the Employee Side Dashboard is a role-based portal that empowers employees to efficiently manage their tasks, view assigned tasks, and access details about subtasks. It enhances organization, transparency, and collaboration within the workplace while maintaining data security.

Admin:

The Admin plays a crucial role in the Task Management System and has extensive authority and responsibilities. Here's a detailed explanation of the Admin's role within the system:

1.Admin Authentication:

Like employees, the Admin uses a unique combination of email and password for authentication during login.

2. Role-Based Access Control (RBAC):

The system employs RBAC to determine the user's role and subsequently grant access to specific features and data.

In this case, the Admin has a distinct "Admin" role that provides elevated privileges.

Admin Dashboard:

Upon successful login, the Admin gains access to the Admin Dashboard, a specialized interface designed for administrators.

The Admin Dashboard is a powerful tool that allows the Admin to manage various aspects of the Task Management System.

1. Task Assignment and Management:

The Admin has the authority to create, assign, and manage tasks within the organization.

This includes defining task details such as names, descriptions, deadlines, priorities, and associated employees.

2. Subtask Assignment:

The Admin can assign subtasks to employees who are already assigned to specific tasks.

Subtasks are allocated to employees as part of the overall task management process, and the Admin ensures that these assignments are appropriate.

3. Employee Management:

The Admin has the capability to add, remove, or modify employee data, taskdata, subtaskdata. This includes setting employee roles, updating contact information, and managing access permissions.

4. Task Monitoring and Reporting:

Monitoring task progress is a significant responsibility of the Admin.

They can view the status of tasks and subtasks, track deadlines, and generate reports to assess productivity and efficiency.

5. Communication and Collaboration:

Admins often serve as a bridge for communication between employees and Task. They can use the system to send notifications or updates to employees and teams about the task.

6. Data Security and Backup:

Ensuring data security and integrity is a priority for the Admin. They may implement security measures, conduct regular backups, and maintain the overall reliability of the system.

Task

1. Task Creation by Admin:

The Admin, who has special privileges, creates tasks within the system.

Each task is defined by a name, description, and other details, such as its priority, assigned group of employees, start date, and completion deadline (end-date).

2. Group Assignment:

Admins can assign tasks to specific groups of employees within the organization.

This feature allows for efficient distribution of workload among teams or departments.

Subtask:

1. Subtask Creation:

Within each task, multiple subtasks can be created. Subtasks break down the main task into smaller, manageable parts.

Subtasks have their own attributes, including names, descriptions, start dates, and end dates.

2. Subtask Assignment:

Admins or task managers assign subtasks to individual employees or team members.

The assignment includes specifying who is responsible for each subtask.

3. Subtask Completion:

As employees work on their assigned subtasks, they update their progress within the system. If a subtask is completed before its end date, it is marked as "Completed."

4. Subtask Delay:

If a subtask is not completed by its end date, it is considered "Delayed."

Delayed subtasks are flagged for further attention and management.

5. Task Completion:

A task is considered completed when all of its subtasks are marked as "Completed."

If any subtask remains "Delayed," the task remains incomplete until all subtasks are resolved.

6. Task Status Tracking:

The system provides real-time status tracking for both tasks and subtasks.

This tracking includes information on task/subtask completion percentages, deadlines, and any delays.

7. Reporting and Analysis:

Admins can generate reports and perform analysis to evaluate the progress of tasks and subtasks. Reports may include completion rates, task durations, and other performance metrics.

8. Task Modification:

Admins have the flexibility to modify tasks and subtasks as needed. This includes adjusting deadlines, reassigning subtasks, or updating task details.

9. User Roles:

User roles, such as Admin and Employee, ensure that individuals have the appropriate permissions and access levels within the system.

3.3 Technology Stack

A software development tool is a program used to create, maintain, test, debug, build and support other applications and software. There are various types of tools like languages, monitoring platforms, databases, frameworks, etc. that programmers use to create software. Each team defines specific factors for the specific project for the decision of the right software development toolset.

software technology is a general term covering the development methods, programming languages, and tools to support them that may be used in the development of software.

VISUAL STUDIO CODE EDITOR:

A standalone source code editor that runs on Windows, macOS, and Linux. The top pick for JavaScript and web developers, with tons of extensions to support just about any programming language. Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

XAMPP:

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP

Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP has the ability to serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package. It is free and easy to use and easily available for Windows, Linux, and Mac OS. It is a beginner's friendly solution package for full-stack web development. It is an open-source software package that gives an easy installation experience.

XAMPP has Advantages like being free and easy to use and easily available for Windows, Linux, and Mac OS. It is a beginner's friendly solution package for full-stack web development. An open-source software package that gives an easy installation experience. Very simple and lightweight to create a setup for development, testing, and deployment. Time-saver and provides several ways for managing configuration changes. It handles many administrative tasks like checking the status and security.

MySQL:

MySQL is the most popular Open-Source Relational SQL Database Management System. MySQL is one of the best RDBMS being used for developing various web-based software applications. MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL. It follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc. MySQL is considered one of the very fast database languages, backed by many benchmark tests, and supports many embedded applications, which makes MySQL very flexible. MySQL is faster, more reliable, and cheaper because of its unique storage engine architecture.

It provides very high-performance results in comparison to other databases without losing an essential functionality of the software. It has fast loading utilities because of the different cache memory. Also uses Triggers, Stored procedures, and views that allow the developer to give higher productivity.

POSTMAN:

The Postman VS code extension is in beta. The Postman VS Code extension is available on Free, Basic, and Professional plans. The Postman VS Code extension enables you to develop and test your APIs in Postman directly from Visual Studio Code. The VS Code extension enables you to:

- Create and send HTTP, raw WebSocket, and RPC requests.
- Import data with URL commands.
- Edit and send requests from a workspace's request history.
- Create cookies and send them with requests.
- Create and manage collections and environments.

Languages:

1. REACT JS:

ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based frontend library responsible only for the view layer of the application. It was created by Jordan Walke, who was a software engineer at Facebook. Facebook developed ReactJS in 2011 in its newsfeed section, but it was released to the public in the month of May 2013.

A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code. The components are the heart of all React applications.

A Component is considered as the core building blocks of a React application. It makes the task of building UIs much easier. Each component exists in the same space, but they work independently from one another and merge all in a parent component, which will be the final UI of your application.

Currently, ReactJS gaining quick popularity as the best JavaScript framework among web developers. It is playing an essential role in the front-end ecosystem. The important features of ReactJS are as following.

- JSX
- Components
- One-way Data Binding
- Virtual DOM
- Simplicity
- Performance

NODE JS:

Node.js is a cross-platform environment and library for running JavaScript applications which is used to create networking and server-side applications. Node.js is a cross-platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications. It is open source and free to use. Many of the basic modules of Node.js are written in JavaScript. Node.js is mostly used to run real-time server applications.

The definition given by its official documentation is as follows:

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. Node.js also provides a rich

library of various JavaScript modules to simplify the development of web applications.

HTML:

The Hyper Text Mark-up Language or HTML is the standard mark-up language for documents designed to be displayed in a web browser. As it is a mark-up language, it consists of many tags. There are tags to display text, tables, ordered lists and unordered lists, etc. There are various HTML versions. The newest version is HTML 5. it's more advanced features like Geo- location, native audio, and video support, Canvas, web socket, etc. Usually, HTML is a simple language to find out and use. A programmer can create an HTML file employing a simple text editor and execute it employing a browser.

HTML has advantages as follows: HTML helps to build the structure of a website and is a widely used Mark-up language. Every browser supports HTML Language, and it is light and weighted, fast to load. Other languages such as JavaScript, CSS, etc. are easily integrated. HTML also allows the utilization of templates, which makes designing a webpage easy.

CSS:

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. In other words, CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.

CSS handles the look and feel part of a web page. Using CSS, one can control the color of the text, the style of fonts, the spacing between paragraphs,

how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is used to style and layout web pages - for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path toward CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

JAVASCRIPT

JavaScript is an object-based scripting language which is lightweight and cross-platform. JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

Features of JavaScript

1. All popular web browsers support JavaScript as they provide built-in execution environments.

2. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
3. It is a light-weighted and interpreted language.
4. It is a case-sensitive language.
5. It provides good control to the users over the web browsers.

3.3.1 Hardware Requirement

The hardware requirements for a React and Node.js project are relatively modest, as these technologies are known for their efficiency. However, the exact hardware specifications can vary depending on the complexity and scale of your project. Here are the general hardware requirements:

1. Computer:

A modern computer with a multi-core processor (e.g., Intel i5 or AMD Ryzen 5) is recommended.

For development purposes, a laptop or desktop with at least 8 GB of RAM is ideal, but 16 GB or more is better for larger projects.

SSD (Solid State Drive) for faster read/write speeds is highly recommended, especially for larger projects.

2. Operating System:

React and Node.js are cross-platform, so you can develop on Windows, macOS, or Linux.

Choose the operating system you're most comfortable with.

3. Internet Connection:

A stable and reasonably fast internet connection is essential for downloading packages, libraries, and updates during development.

4. Text Editor or Integrated Development Environment (IDE):

You can use text editors like Visual Studio Code, Sublime Text, or IDEs like WebStorm for coding.

These tools should run smoothly on most modern computers.

5. Browser:

For testing and debugging React applications, having the latest versions of popular browsers like Chrome, Firefox, and Edge is essential.

6. Version Control:

Consider using version control systems like Git for code management. Git doesn't have high hardware requirements.

7. Memory and CPU:

The more RAM (memory) you have, the better, especially for running development tools, browsers, and your application simultaneously.

While React and Node.js aren't particularly CPU-intensive, a modern CPU will improve the performance of your development environment.

CHAPTER-4

SYSTEM DESIGN

4.SYSTEM DESIGN

System design in web development is the process of creating a detailed specification of a software system's architecture, components, modules, interfaces, and data for a web application. It's a crucial phase that comes after requirements gathering and analysis. Here's a high-level overview of the steps involved in system design for web development.

4.1 Requirement Analysis:

4.1.1 Functional Requirement

User Registration and Authentication:

Users should be able to use created accounts with unique usernames and passwords.

Users should log in securely using their credentials.

Task Creation and Assignment:

Users (typically Admins) should be able to create tasks with the following details:

Task name and description, Due date, Assigned employees or teams.

Admins should assign tasks to individual employees or groups.

Admins should set task priorities (e.g., high, low).

Subtask Management:

Each task can have multiple subtasks.

Subtasks should have their own due dates and descriptions.

Users should mark subtasks as completed or delayed.

Task Tracking:

Users should be able to view the status of tasks and subtasks.

Task status should be updated automatically based on subtask statuses:

If all subtasks are completed, the task is considered completed. If any subtask is delayed, the task is considered delayed. Users should receive notifications when tasks are assigned, completed, or delayed.

Reporting and Analytics:

Users should generate reports on task and employee performance.

Reports should include metrics like completion rates, task aging, and employee productivity.

4.1.2 Non-Functional Requirement:

Performance:

The system should respond quickly to user interactions.

It should support a specified number of concurrent users without significant performance degradation.

Security:

User data should be stored securely and protected from unauthorized access.

Communication between clients and the server should be encrypted (e.g., HTTPS).

User authentication should be robust, with password hashing and salting.

Usability:

The user interface should be intuitive and easy to navigate.

It should follow accessibility standards to accommodate users with disabilities.

Scalability:

The system should be able to handle an increase in users and tasks over time.

Scalability should include the database, server infrastructure, and application components.

4.2 Database Schema

A database schema is a visual representation of the structure of a database, including tables, columns, data types, relationships, and constraints. In the context of a project document, explaining the database schema is crucial for developers, stakeholders, and anyone involved in the project to understand how data is organized and stored in the database.

4.2.1 Class Diagram

A class diagram is a type of diagram in the Unified Modeling Language (UML) that represents the structure and relationships of classes within a system or application. It is a fundamental tool in object-oriented modeling and design, providing a visual representation of the system's classes, attributes, methods, and

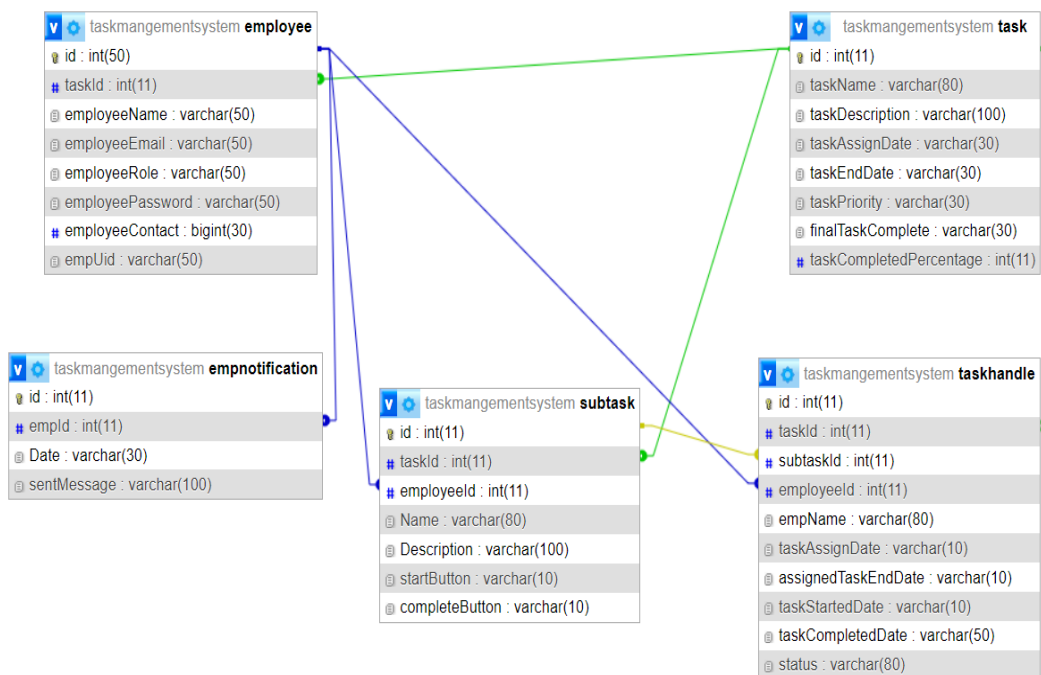


Figure 10. Class Diagram

4.2.2 ER-Diagram

An ER Diagram, or Entity-Relationship Diagram, is a visual representation of the data model that describes how entities (or objects), attributes (or properties), and relationships are connected within a system or a database. ER diagrams are widely used in database design and serve as a powerful tool for modeling the structure and organization of data.

Entity: An entity is a real-world object or concept that has data attributes to describe it. In a database context, entities are often used to represent tables. For example, in a library database, "Book," "Author," and "Library Member" could be entities.

Attribute: An attribute represents a property or characteristic of an entity. Attributes describe the data associated with an entity. For instance, a "Book" entity might have attributes like "Title," "Author," and "Publication Date."

Relationship: A relationship shows how two or more entities are related or connected. Relationships can be one-to-one, one-to-many, or many-to-many, and they define how data flows between entities. For instance, a "Library Member" entity may have a relationship with a "Book" entity to indicate that a member can borrow many books.

Cardinality: Cardinality defines the number of instances of one entity that can be related to another entity through a relationship. Common cardinalities include "1" (one), "0..1" (zero or one), "0.." (zero or more), and "1.." (one or more). For example, a "Student" entity might have a one-to-many relationship with a "Course" entity, indicating that one student can enroll in many courses.

Primary Key: A primary key is an attribute or a set of attributes that uniquely identify each instance (or row) of an entity in a database table. It ensures that there are no duplicate records. In an ER diagram, a primary key is typically underlined.

Foreign Key: A foreign key is an attribute in one entity that refers to the primary key of another entity. It establishes a link between entities and enforces referential integrity. In an ER diagram, a foreign key is usually marked with an "FK."

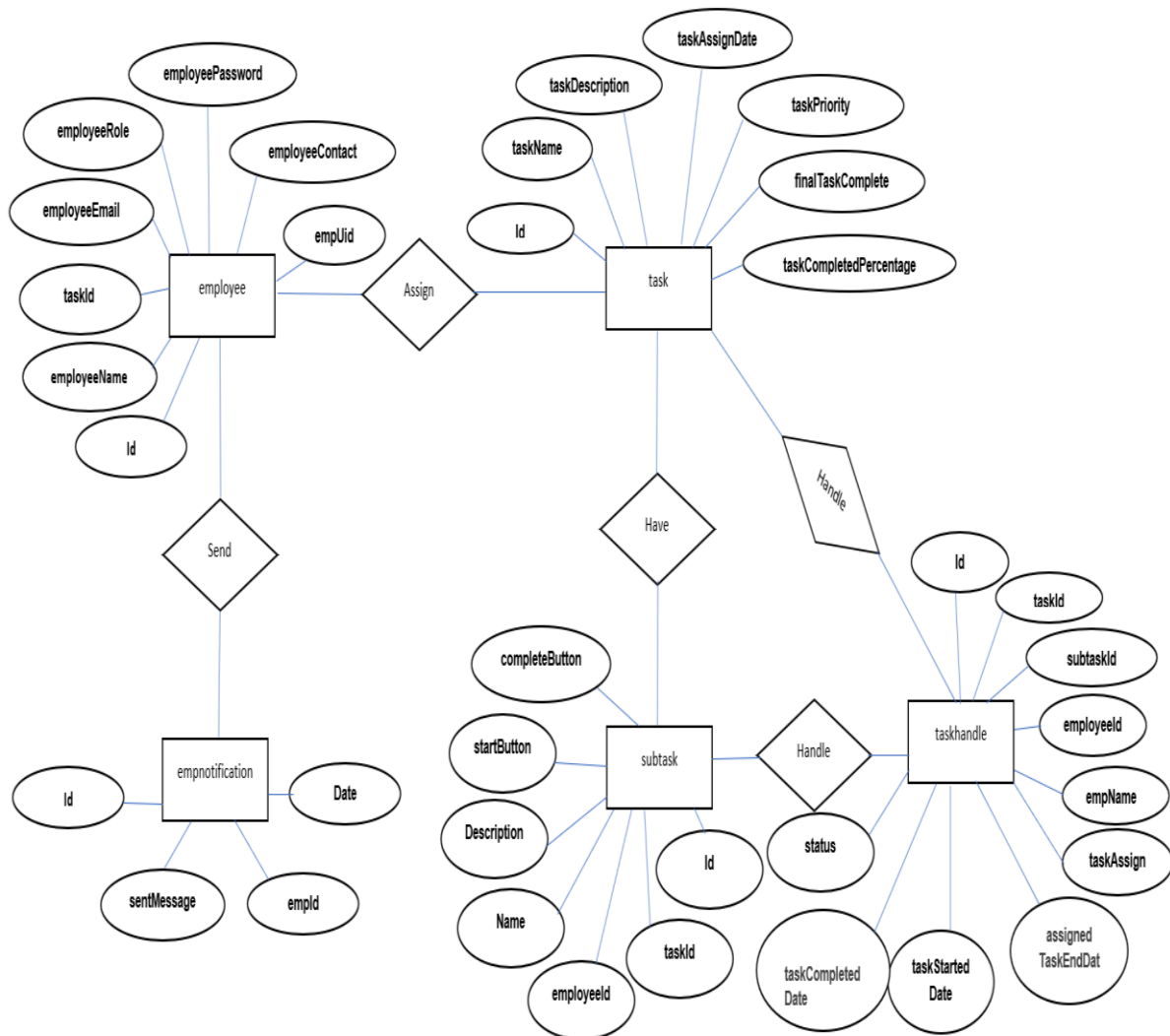
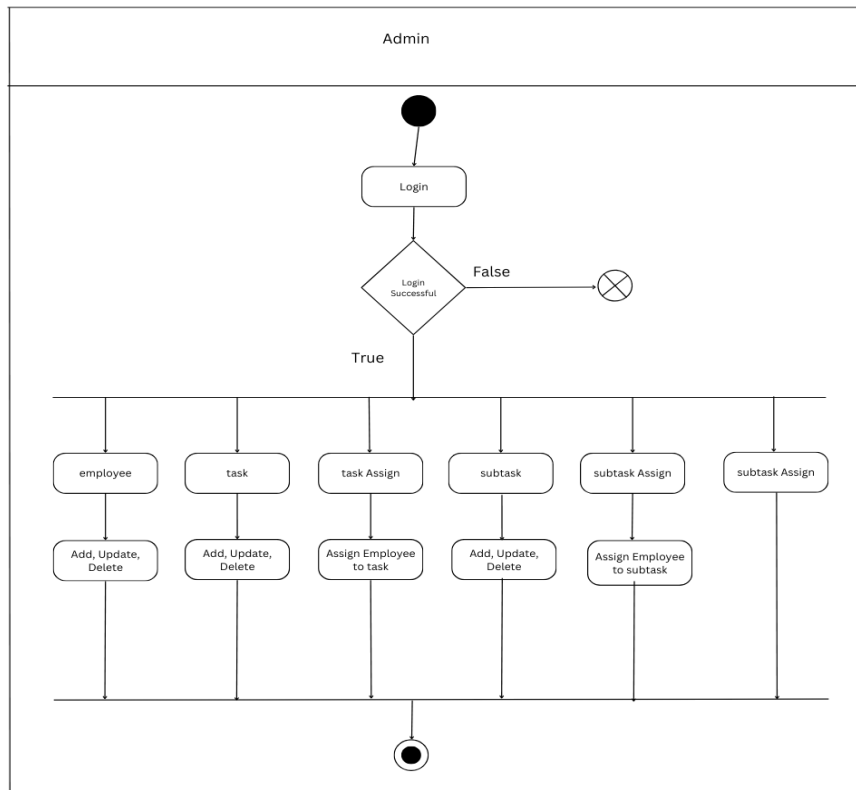


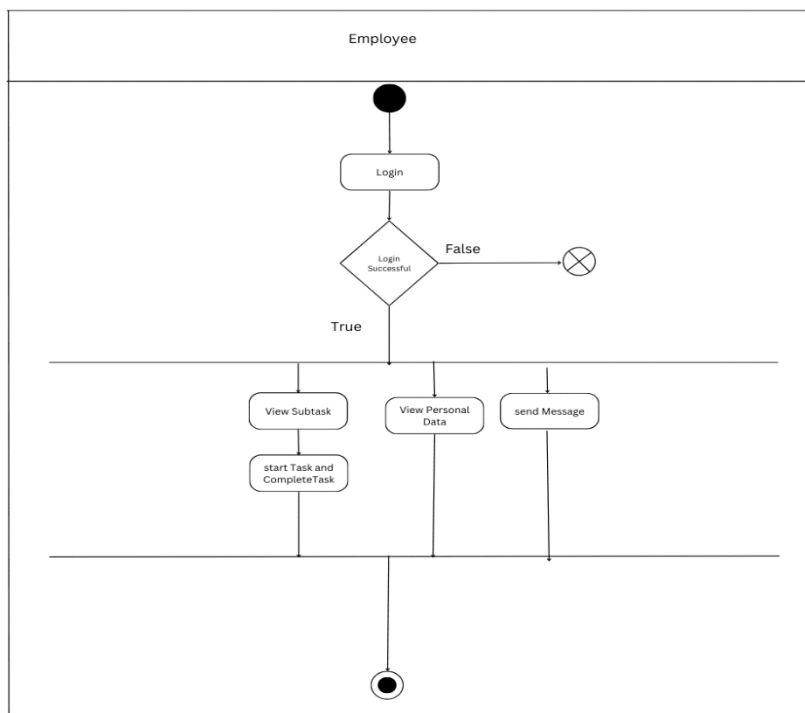
Figure 11. ER- Diagram

4.2.3 Activity- Diagram

Admin:



Customer:



CHAPTER-5

SYSTEM TESTING

AND VALIDATION

5 SYSTEM TESTING AND VALIDATION

5.1 Test Cases

Creating test cases for a task management system involves detailing scenarios and conditions to verify that the system functions correctly.

Test Case ID	Test Case Description	Test Steps	Expected Results	Pass/Fail
TC001	Verify user can log in as an employee	1. Navigate to the login page 2. Enter valid employee credentials 3. Click the login button	Employee is logged into their dashboard	Pass
TC002	Verify user cannot log in with incorrect credentials	1. Navigate to the login page 2. Enter incorrect credentials 3. Click the login button	Error message is displayed, login fails	Pass
TC003	Verify admin can log in with correct credentials	1. Navigate to the login page 2. Enter valid admin credentials 3. Click the login button	Admin is logged into the admin dashboard	Pass
TC004	Verify admin can add a new task	1. Log in as an admin 2. Navigate to the "Create Task" page 3. Fill in task details 4. Click the "Create" button	Task is created and visible in the task list	Pass
TC005	Verify admin can assign a task to an employee	1. Log in as an admin 2. Navigate to the task details page 3. Assign the task to an employee 4. Save changes	Task is assigned to the employee	Pass

Test Case ID	Test Case Description	Test Steps	Expected Results	Pass/Fail
TC006	Verify admin can create a subtask for a task	1. Log in as an admin 2. Navigate to a task details page 3. Click "Add Subtask" 4. Fill in subtask details 5. Click "Save"	Subtask is created and associated with the task	Pass
TC007	Verify user can mark a subtask as completed	1. Log in as an employee 2. Navigate to a task with subtasks 3. Mark a subtask as completed	Subtask status is changed to "Completed"	Pass
TC008	Verify system calculates task completion percentage	1. Log in as an employee 2. Navigate to a task with subtasks 3. Complete all subtasks	Task status is changed to "Completed (100%)"	Pass
TC009	Verify user can mark a subtask as completed	1. Log in as an employee 2. Navigate to a task with subtasks 3. Mark a subtask as completed	Subtask status is changed to "Delay", If subtask completed date exceeded	Pass
TC010	Verify admin can edit task details	1. Log in as an admin 2. Navigate to a task details page 3. Edit task details and save changes	Task details are updated successfully	Pass

Test Case ID: A unique identifier for each test case.

Test Case Description: A brief description of what the test is checking.

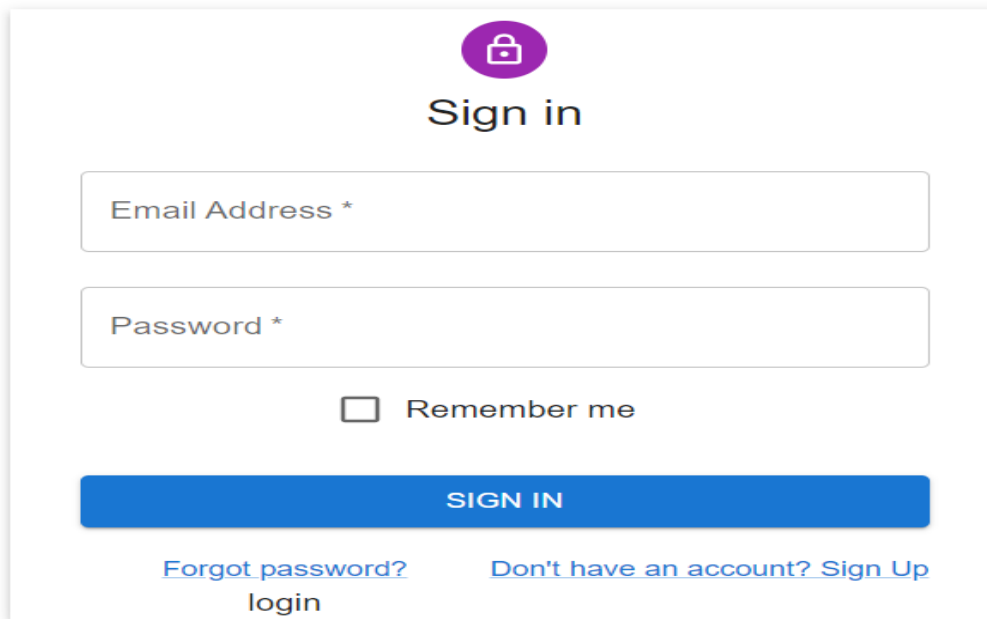
Test Steps: A step-by-step guide for executing the test.

Expected Results: What should happen when the test is executed.

Pass/Fail: Whether the test passed or failed during execution.

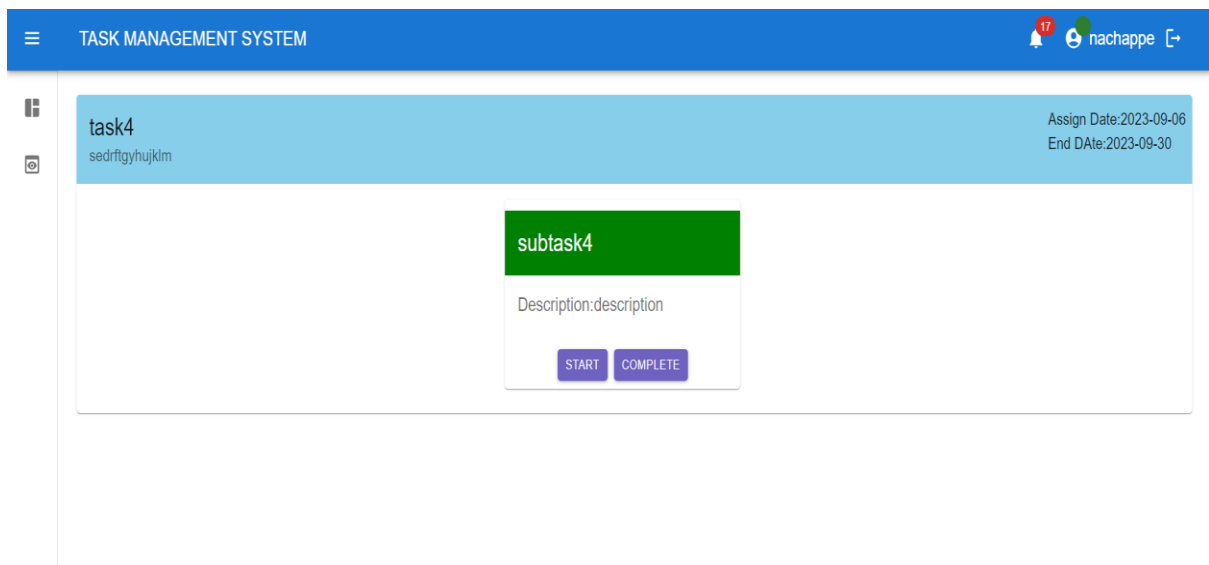
CHAPTER-6
EXPERIMENTS
AND
RESULTS

6. Experiments and Results



The login page features a purple lock icon at the top center. Below it, the text "Sign in" is displayed. There are two input fields: "Email Address *" and "Password *". A checkbox labeled "Remember me" is positioned below the password field. A prominent blue button labeled "SIGN IN" is centered below the inputs. At the bottom, there are two links: "Forgot password? login" and "Don't have an account? Sign Up".

Figure 12. Login Page



The screenshot shows a "TASK MANAGEMENT SYSTEM" interface. The top blue header contains a menu icon, the system name, and a user profile for "nachappe" with a notification bell showing 17 alerts. The main content area displays a task card for "task4" with a description "sedrtgyhujklm". The task is assigned to an employee, indicated by a small icon. The task's status is "subtask4", shown in a green box. Below this, the description "Description:description" is visible, followed by "START" and "COMPLETE" buttons. The task's timeline is shown as "Assign Date:2023-09-06" and "End DAte:2023-09-30".

Figure 13. subtask Assigned to Employee

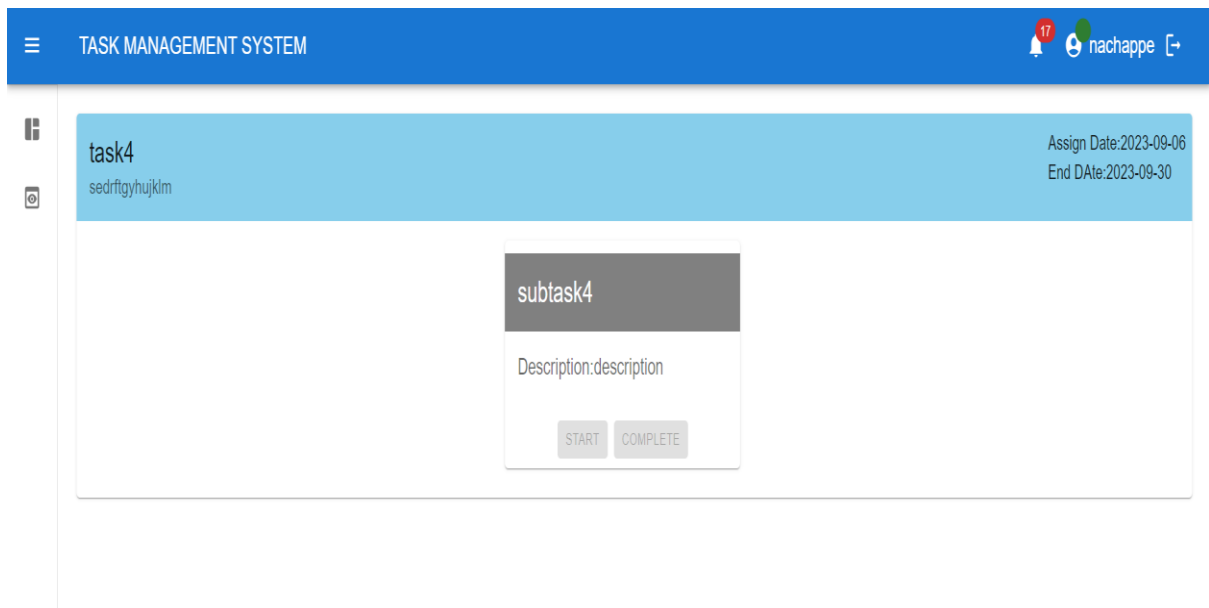


Figure 14. subtask is completed

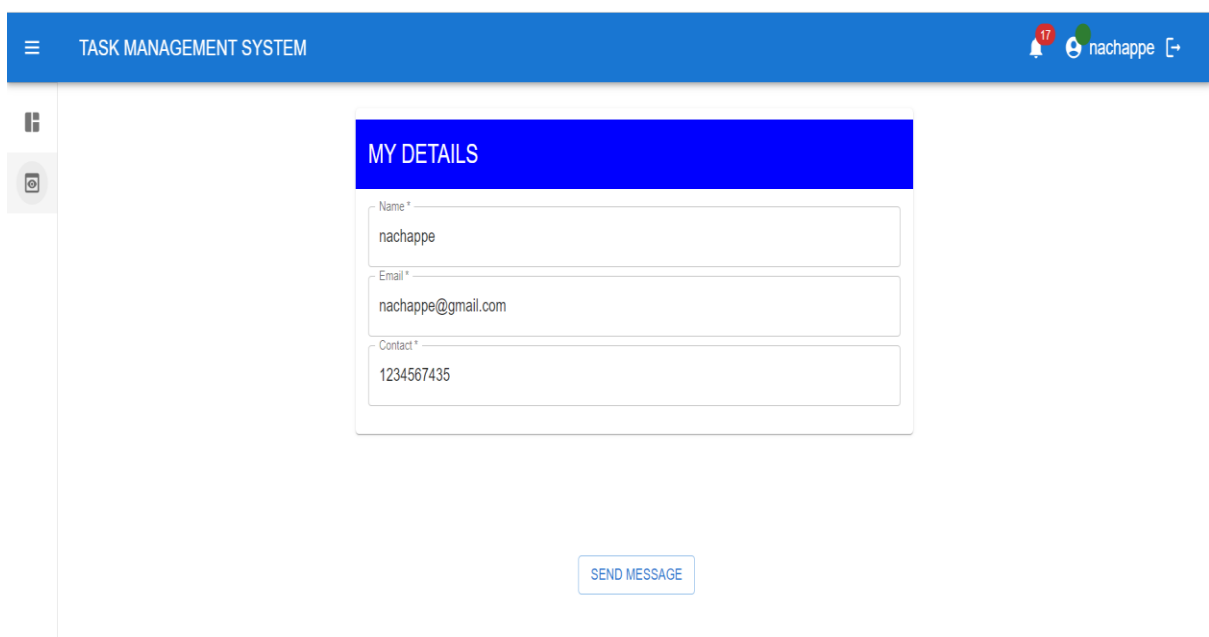


Figure 15. Employee personal Data

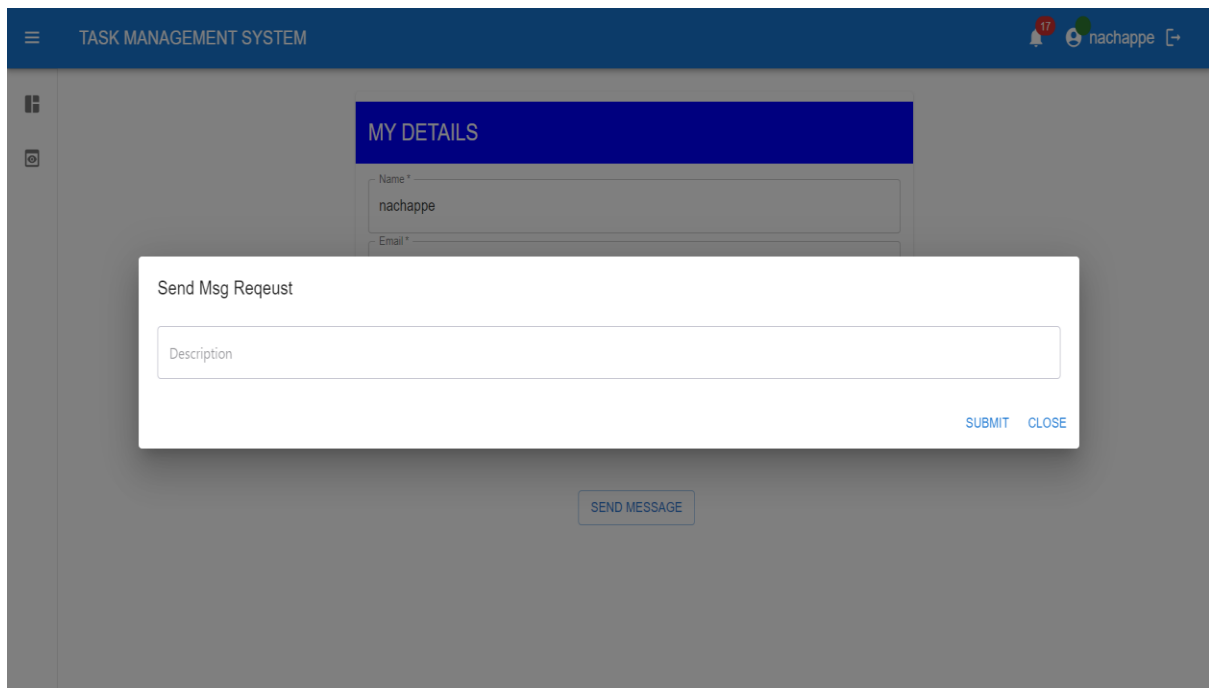


Figure 16. send Message Request

TASK MANAGEMENT SYSTEM																																																											
<div> <div>COLUMNS</div> <div>FILTERS</div> <div>DENSITY</div> <div>EXPORT</div> </div> <table> <tr> <th><input type="checkbox"/></th> <th>Title</th> <th>Description</th> <th>Assigned Date</th> <th>End Date</th> <th>Priority</th> <th>Percentage</th> <th>Status</th> <th>Assigned Employee</th> <th>View SubTasks</th> </tr> <tr> <td><input type="checkbox"/></td> <td>Task1</td> <td>task1</td> <td>2023-08-03</td> <td>2023-08-16</td> <td>High</td> <td>100%</td> <td>Completed</td> <td>spooki</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td>Task2</td> <td>task2</td> <td>2023-08-04</td> <td>2023-08-25</td> <td>Low</td> <td>0%</td> <td>Not Completed</td> <td>indu,indu,manvitha</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td>Task3</td> <td>task3</td> <td>2023-08-10</td> <td>2023-08-31</td> <td>Low</td> <td>0%</td> <td>Not Completed</td> <td>indu3</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td>task4</td> <td>sedrftgyhujkim</td> <td>2023-09-06</td> <td>2023-09-30</td> <td>Low</td> <td>100%</td> <td>Completed</td> <td>nachappe,manvl...</td> <td></td> </tr> </table> <div> <div>Rows per page: 100</div> <div>1-4 of 4</div> <div>< ></div> </div>										<input type="checkbox"/>	Title	Description	Assigned Date	End Date	Priority	Percentage	Status	Assigned Employee	View SubTasks	<input type="checkbox"/>	Task1	task1	2023-08-03	2023-08-16	High	100%	Completed	spooki		<input type="checkbox"/>	Task2	task2	2023-08-04	2023-08-25	Low	0%	Not Completed	indu,indu,manvitha		<input type="checkbox"/>	Task3	task3	2023-08-10	2023-08-31	Low	0%	Not Completed	indu3		<input type="checkbox"/>	task4	sedrftgyhujkim	2023-09-06	2023-09-30	Low	100%	Completed	nachappe,manvl...	
<input type="checkbox"/>	Title	Description	Assigned Date	End Date	Priority	Percentage	Status	Assigned Employee	View SubTasks																																																		
<input type="checkbox"/>	Task1	task1	2023-08-03	2023-08-16	High	100%	Completed	spooki																																																			
<input type="checkbox"/>	Task2	task2	2023-08-04	2023-08-25	Low	0%	Not Completed	indu,indu,manvitha																																																			
<input type="checkbox"/>	Task3	task3	2023-08-10	2023-08-31	Low	0%	Not Completed	indu3																																																			
<input type="checkbox"/>	task4	sedrftgyhujkim	2023-09-06	2023-09-30	Low	100%	Completed	nachappe,manvl...																																																			

Figure 17. Admin Dashboard

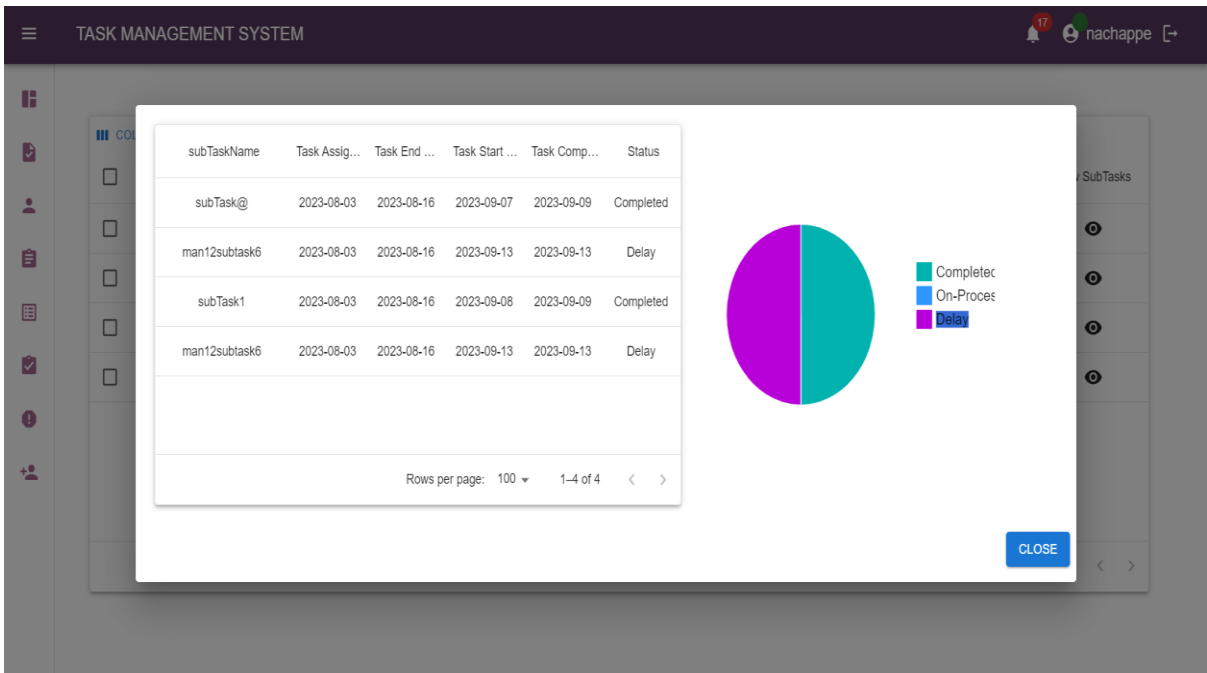


Figure 18. number of subtask under each task

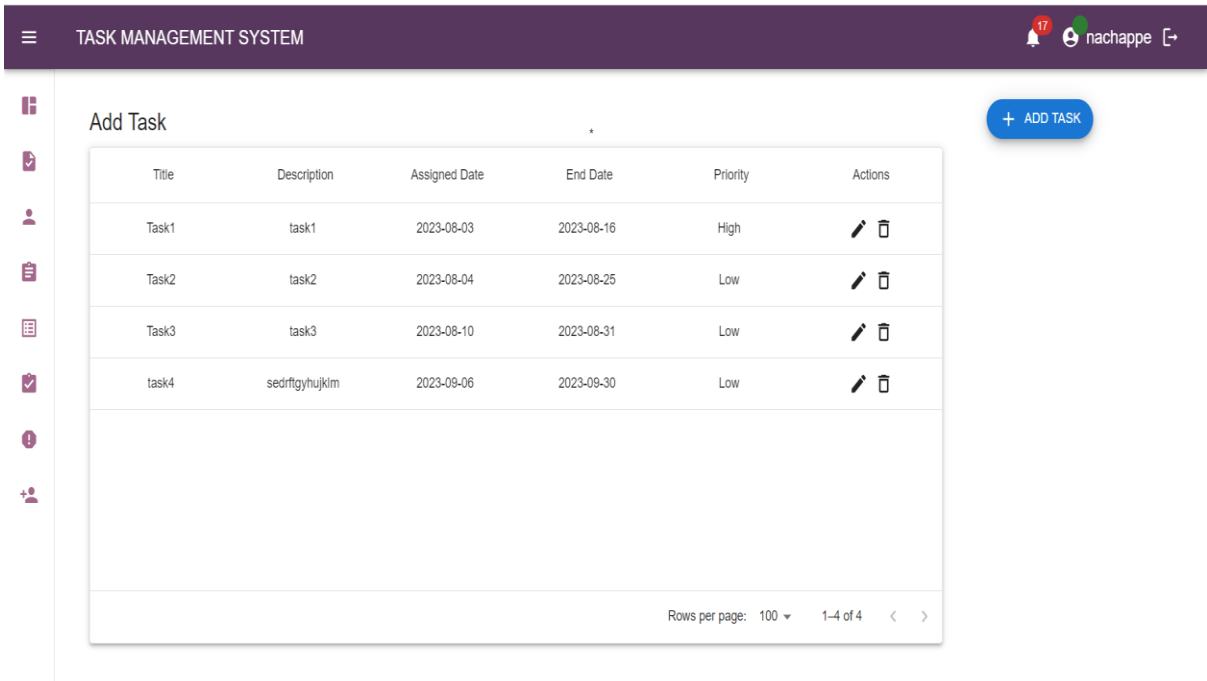


Figure 19. all Added task

ADD Task

Title *

Assign Date * dd-mm-yyyy

End Date * dd-mm-yyyy

TaskType

Description

Priority *

ADD CANCEL

Figure 20. Add Task

Add Employee

UID	Name	Role	Email	Contact No	Actions
rdl13	indu	Employee	indu@gmail.com	1234567877	
rdl123	spooki	Employee	spoo@gmail.com	4676535367	
rdl14	nachappe	Employee	nachappe@gmail.com	1234567435	
rdl13	indu1	Employee	indu1@gmail.com	1234567877	
rdl13	indu	Admin	admin@gmail.com	1234567877	
111	manviltha	Employee	ma@gmail.com	4545655768	
rgg	ggdf	Admin	manviltha@gmail.com	4545655768	
1	R15	Employee	R15@gmail.com	4545655768	

Rows per page: 100 1-8 of 8

Figure 21. All Employee Details

ADD Employee

Employee ID *

Email *

Name *

Password *

Role *

Contact(10 digits only) *

ADD CANCEL

UID	Name	Role	Email	Contact No	Actions
rdl13					
rdl123					
rdl14					
rdl13					
rdl13					
111					
rgg					

Rows per page: 100 1-8 of 8

Figure 22. Add Employee

Assign Employee

+ ASSIGN EMPLOYEE

Task	Start Date	End Date	Employee UID	EmployeeName
Task1	2023-08-03	2023-08-16	rdl123	spooki
Task2	2023-08-04	2023-08-25	rdl13	indu
Task2	2023-08-04	2023-08-25	rdl13	indu
Task2	2023-08-04	2023-08-25	111	manvitha
task4	2023-09-06	2023-09-30	rdl14	nachappe
task4	2023-09-06	2023-09-30	1	R15

Rows per page: 100 1-6 of 6

Figure 23.Task Assigned Employees Details

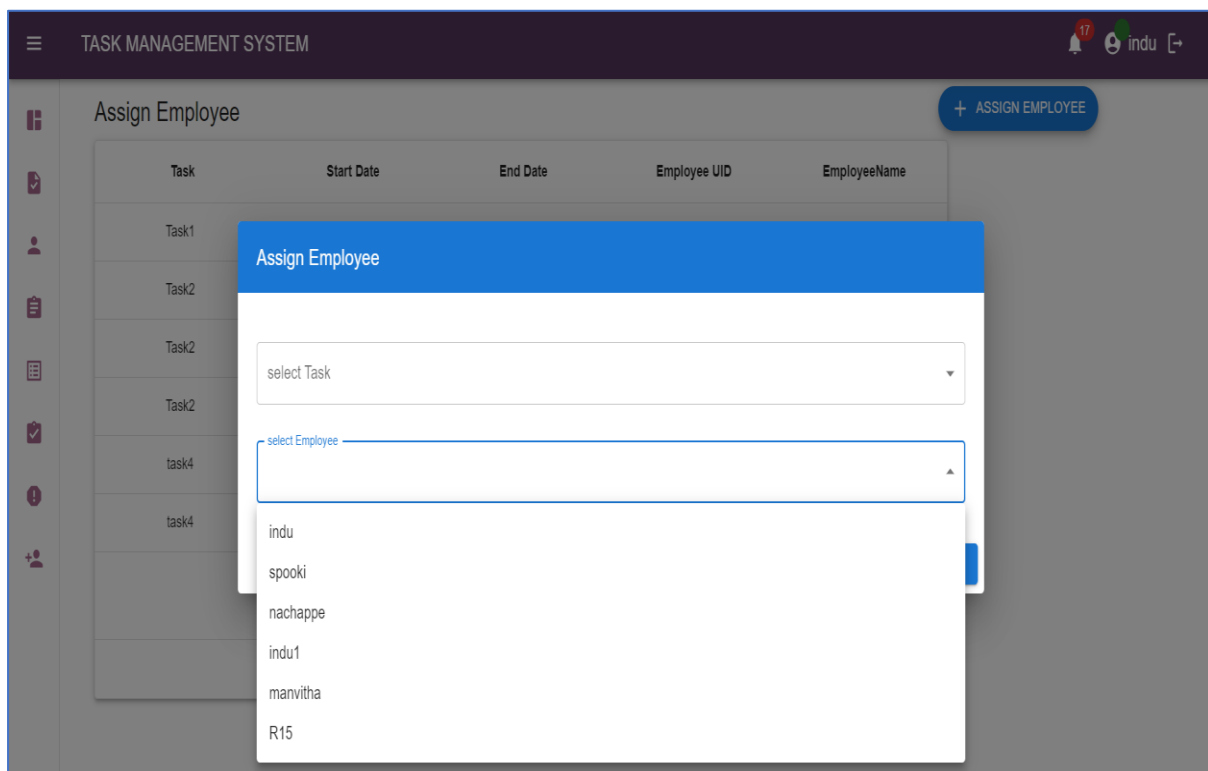


Figure 24. Assign Employee

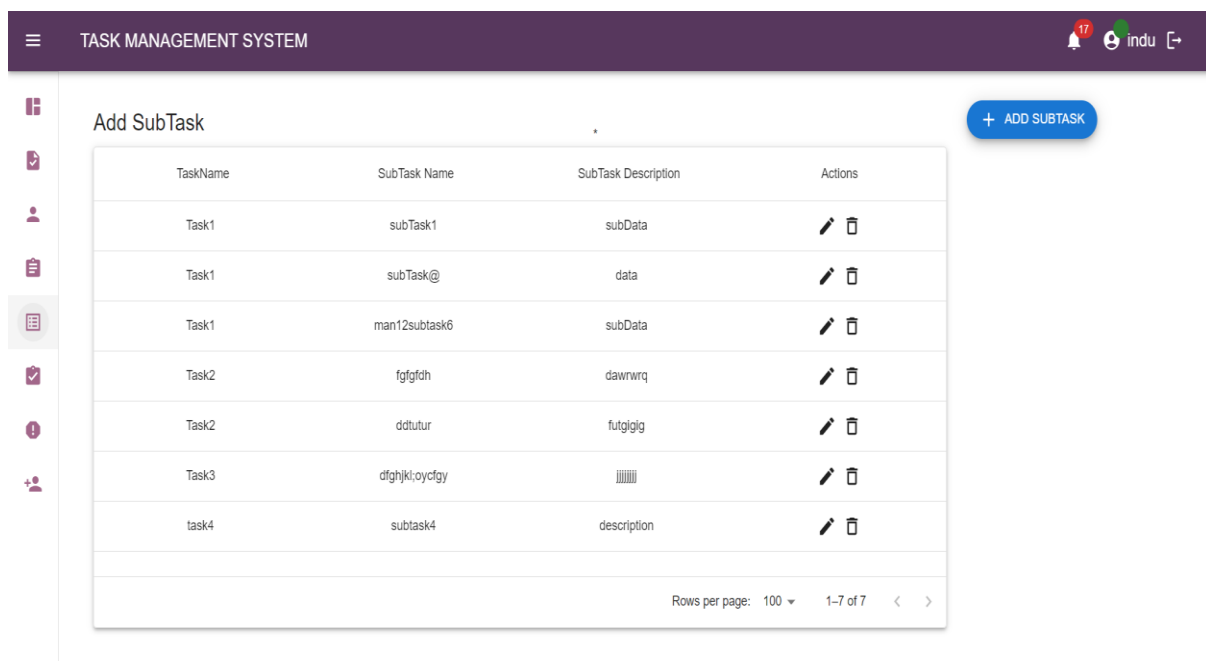


Figure 25. Subtask Details

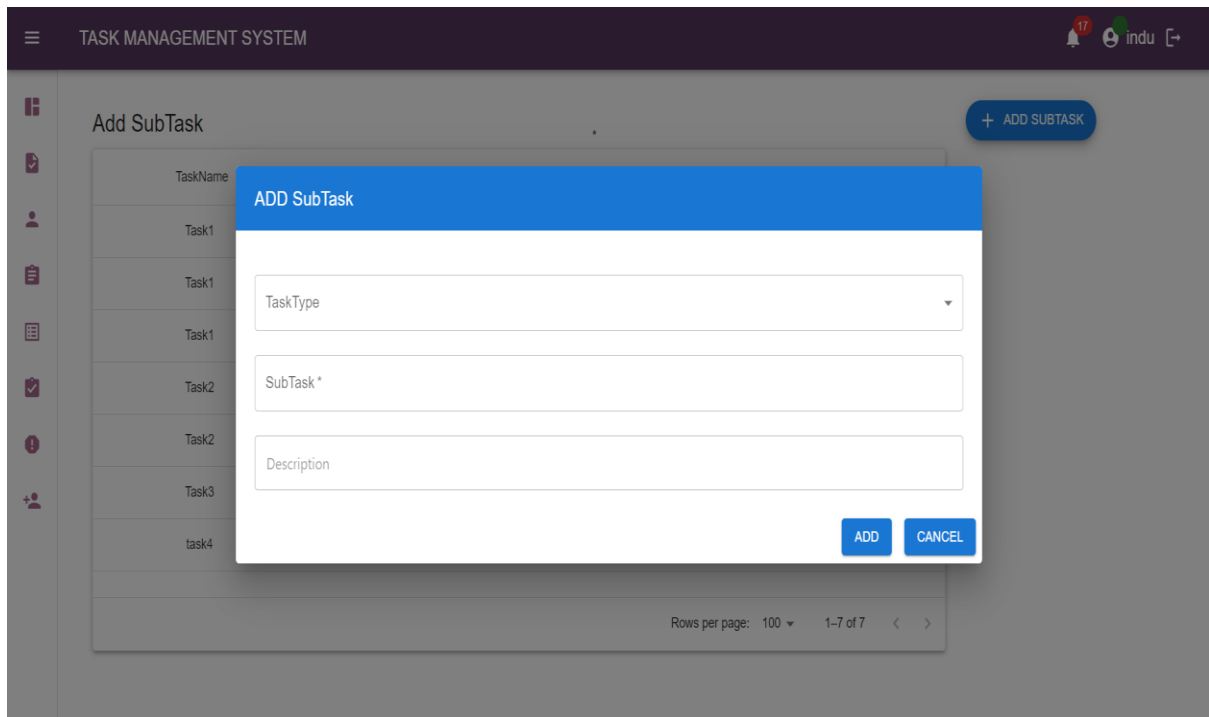


Figure 26 .Add Subtask

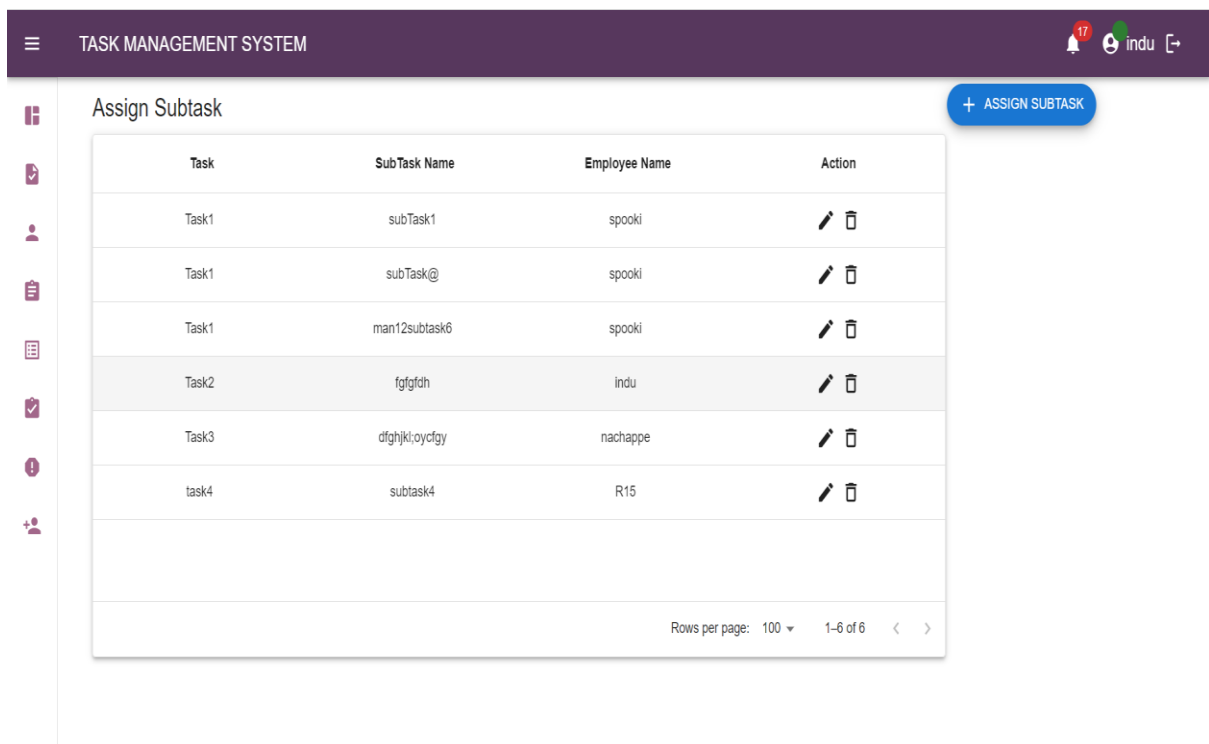


Figure 27.Assign Subtask

The screenshot shows the 'Assign Subtask' modal in the Task Management System. The modal has a blue header with the text 'ADD Employee'. Below the header, there are three dropdown menus labeled 'Task *', 'SubTask *', and 'Employee *'. At the bottom right of the modal, there are two buttons: 'ADD' and 'CANCEL'. In the background, a table is visible with columns: Task, SubTask Name, Employee Name, and Action. The table has six rows with data: Task1, Task1, Task1, Task2, Task3, and task4. The table is part of a larger interface with a sidebar on the left and a top navigation bar.

Figure 28.Assign Subtask to employee

The screenshot shows the 'Task Report' page in the Task Management System. At the top, there is a 'Task Name' dropdown menu set to 'Task1'. Below this, there is a table titled 'Task Report' with the following columns: TaskName, SubTask Name, Employee Id, Employee Name, TaskAssignDate, TaskEndDate, TaskStartedDate, TaskCompleted..., and Status. The table contains four rows of data:

TaskName	SubTask Name	Employee Id	Employee Name	TaskAssignDate	TaskEndDate	TaskStartedDate	TaskCompleted...	Status
Task1	subTask@	72	spooki	2023-08-03	2023-08-16	2023-09-07	2023-09-09	Completed
Task1	man12subtask6	72	spooki	2023-08-03	2023-08-16	2023-09-08	2023-09-13	Delay
Task1	subTask1	72	spooki	2023-08-03	2023-08-16	2023-09-08	2023-09-09	Completed
Task1	man12subtask6	72	spooki	2023-08-03	2023-08-16	2023-09-13	2023-09-13	Delay

Figure 29 .Task Report

CHAPTER-7

Conclusion And Future Work

7.1 Conclusion

development and implementation of the Task Management System represent a significant milestone in streamlining task-related workflows and enhancing productivity within the organization. This project aimed to automate and simplify the process of task assignment, monitoring, and completion, benefiting both individual employees and the administrative team.

7.2 Future Scope

1. Personalized Productivity Insights: TMS will provide personalized productivity recommendations based on an individual's work habits and goals, helping users work more efficiently and effectively.

2.Artificial Intelligence and Automation: TMS will increasingly leverage artificial intelligence (AI) and machine learning (ML) to automate repetitive tasks, predict task completion times, suggest task prioritization, and even perform data analysis to identify trends and insights within task data.

3.Integration with Other Tools: TMS will integrate seamlessly with a wide range of software tools such as calendar apps, email clients, project management software, and communication platforms. This will ensure a unified and efficient workflow by eliminating the need for manual data entry.

4.Mobile and Wearable Devices: With the growing use of mobile devices and wearables, TMS will be designed to work flawlessly on these platforms, enabling users to manage tasks and receive updates on the go. Voice-activated task management via smart speakers and wearables will also become more prevalent.

5.Collaboration and Teamwork: Collaboration features will be further enhanced, allowing for real-time collaboration on tasks, shared task lists, and integrated communication channels. Enhanced virtual reality and augmented reality features may also play a role in remote collaboration within TMS.

