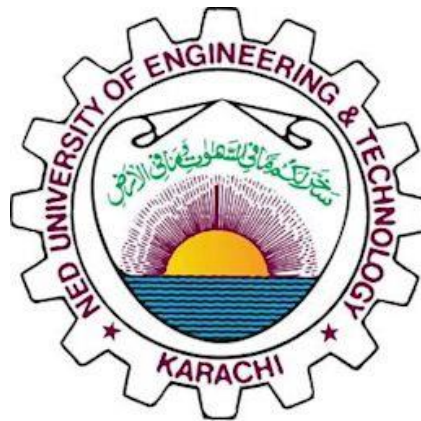


**SOFTWARE ENGINEERING (CS-326)**  
**PROJECT: WORKSPY PROJECT**  
**REPORT**



GROUP: G3  
MAHNOOR ZIA (CS22101)  
ANOOSHA KHALID (CS22104)  
LAIBA IQRAR (CS22112)

SUBMITTED TO: SIR ALI AKHTAR

NOVEMBER 29, 2024

## Table of Contents

<b>1- Abstract.....</b>	<b>3</b>
<b>2- Introduction .....</b>	<b>3</b>
2.1 Background .....	3
2.2 Objectives .....	3
2.3 Scope.....	3
<b>3- Requirement Analysis.....</b>	<b>3</b>
3.1 Functional Requirements.....	3
3.2 Non-Functional Requirements .....	4
3.3 Use Case.....	4
<b>4- System Design .....</b>	<b>5</b>
4.1 Architecture Overview .....	5
4.2 UML Diagrams .....	5
4.3 Database Design.....	7
4.4 Technology Stack.....	7
<b>5- Implementation .....</b>	<b>7</b>
5.1 Development Environment .....	7
5.2 Code Structure .....	7
5.3 Key feature .....	8
<b>6- Project Plan.....</b>	<b>8</b>
6.1 Project Timeline and Resource Allocation.....	8
6.2 Risk Management .....	11

<b>7- Design Document .....</b>	<b>12</b>
7.1 System Architecture.....	12
7.2 Detailed Design .....	13
7.3 User Interface Design .....	14
<b>8- Result and Analysis .....</b>	<b>17</b>
9.1 Performance Metrics .....	17
<b>9- Challenges and Solutions.....</b>	<b>17</b>
<b>10- Future Work.....</b>	<b>18</b>
<b>11- Conclusion.....</b>	<b>18</b>
<b>12- References .....</b>	<b>18</b>
<b>13- Appendices .....</b>	<b>18</b>
<b>14- Testing Document.....</b>	<b>20</b>

## 1. ABSTRACT

WorkSpy is a streamlined task and project management system designed for enterprises. It empowers teams with intuitive dashboards for managers and employees, allowing for task creation, project tracking, and deadline management. Employees can submit leave applications, while managers can provide feedback, approve requests, and oversee performance through timesheets and analytics.

The system uses the MVC design pattern, ensuring clear separation of logic for better scalability and maintenance. Built with Node.js, Express, and MySQL, WorkSpy combines a robust back end with a responsive front-end to deliver a user-friendly experience.

## 2. INTRODUCTION

### 2.1 Background

Managing tasks efficiently is essential for personal and professional productivity. WorkSpy was developed to provide users with a robust platform to track, organize, and complete their projects and tasks with ease.

### 2.2 Objectives

- To create a user-friendly task management system.
- To enable efficient tracking of daily progress.
- To provide features like task categorization, prioritization, and collaboration

### 2.3 Scope

WorkSpy focuses on efficient workplace management by providing role-based authentication, dashboards for managers and employees, task and project management, timesheets, leave applications, feedback mechanisms, and analytics. It ensures seamless communication and productivity tracking in professional settings.

## 3. REQUIREMENT ANALYSIS

### 3.1 Functional Requirements

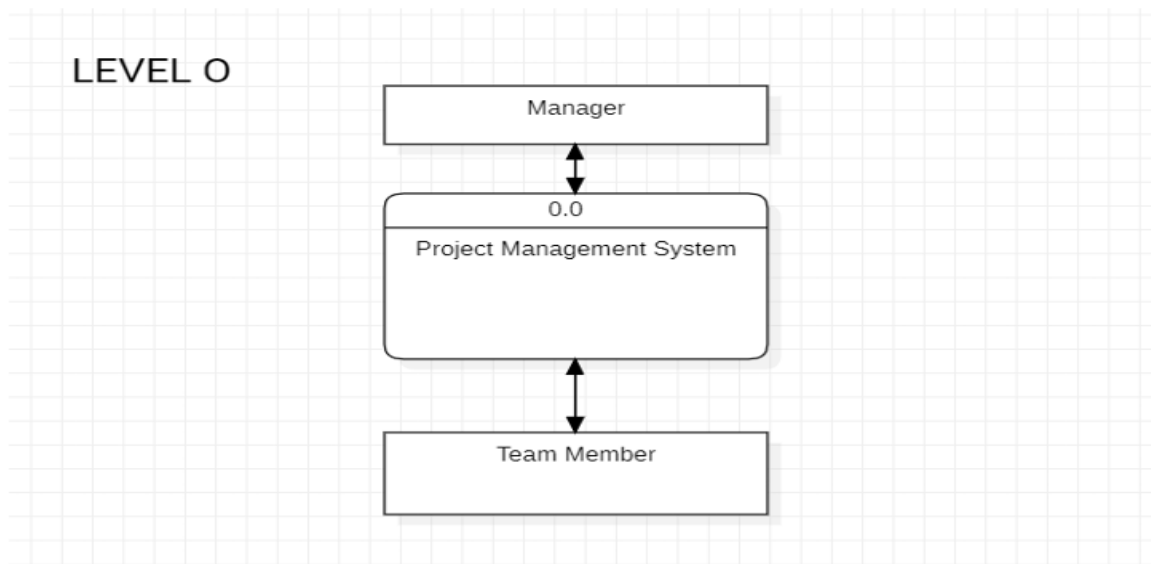
- User authentication and authorization.
- Manager and employee dashboards.
- Add users (Co-Manager or Employees') by the super admin only which is also a manager.

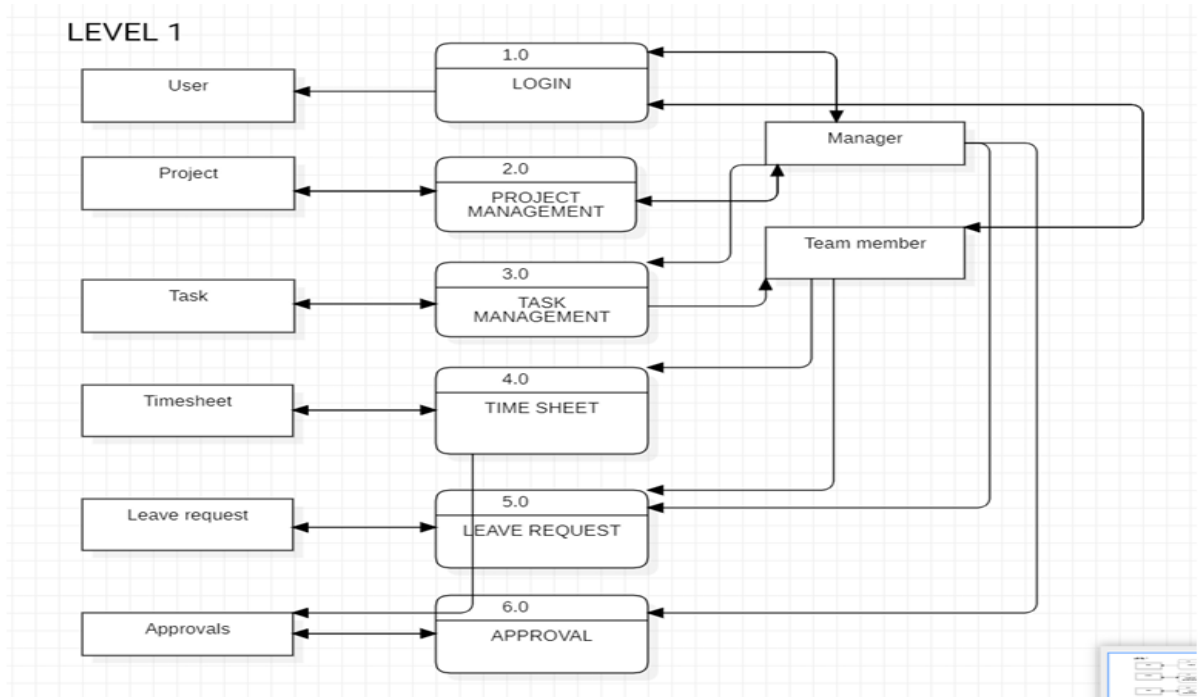
- Task and Project Creation.
- Deadlines for Project & tasks.
- Maintaining Timesheets to keep track of employees' work
- Submission of leave Applications from the employee side
- Approval or rejection of Applications from the manager's side
- Managers comments (feedback) on timesheets to give a better idea about the employee's progress.
- Report Analytics for employee and manager dashboards.

### 3.2 Non-Functional Requirements

- High performance and reliability.
- User-friendly interface
- Data security and encryption.

### 3.3 Use Case Diagram





## 4. SYSTEM DESIGN

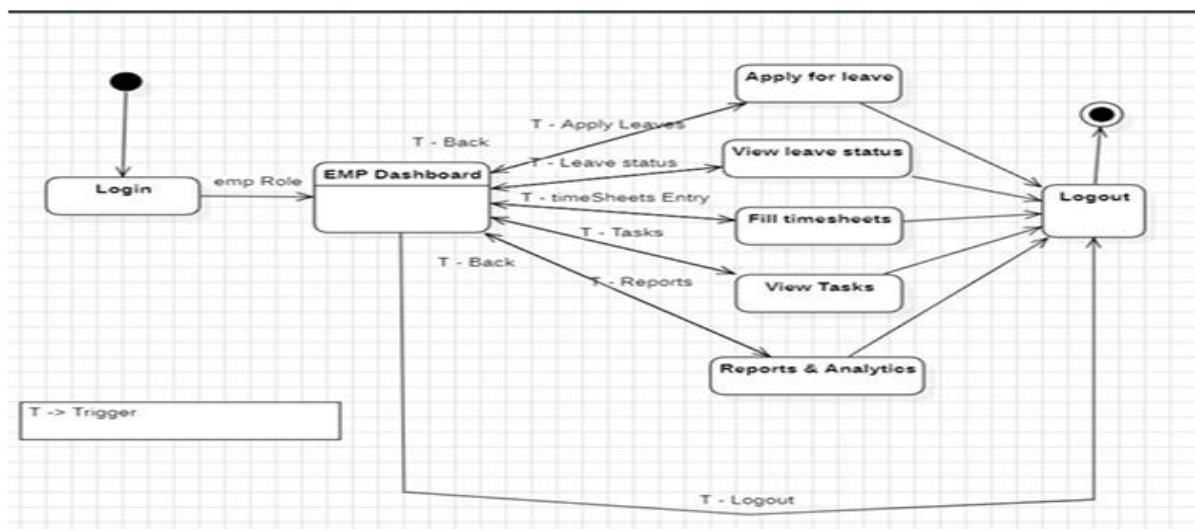
### 4.1 Architecture Overview

WorkSpy follows a client-server architecture with a responsive front-end and a robust back end for data processing.

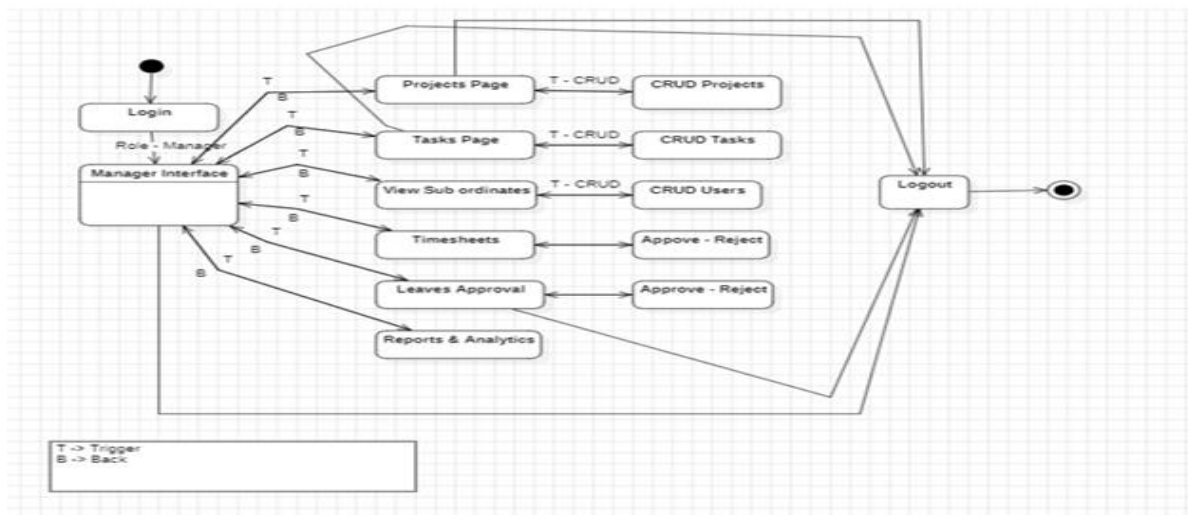
### 4.2 UML Diagrams

#### - State Transition Diagram

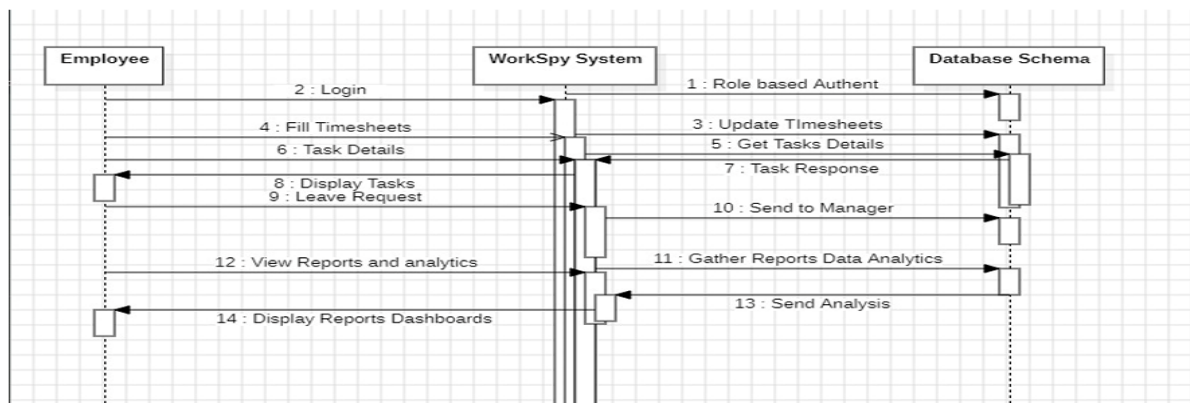
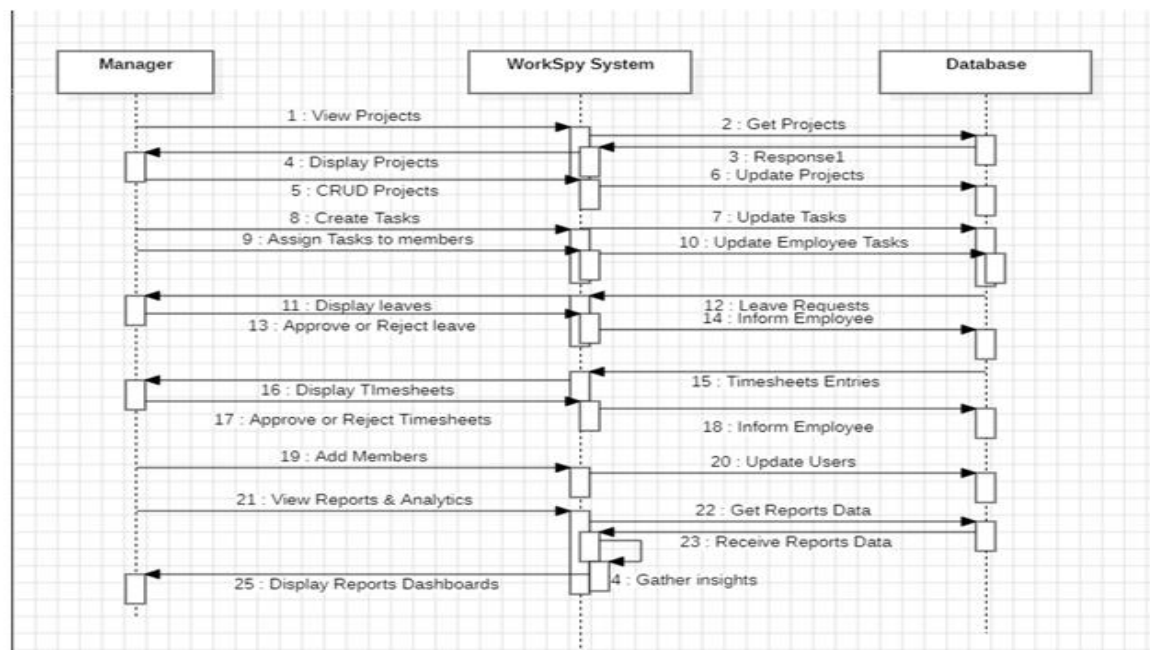
For employee:



For Manager:

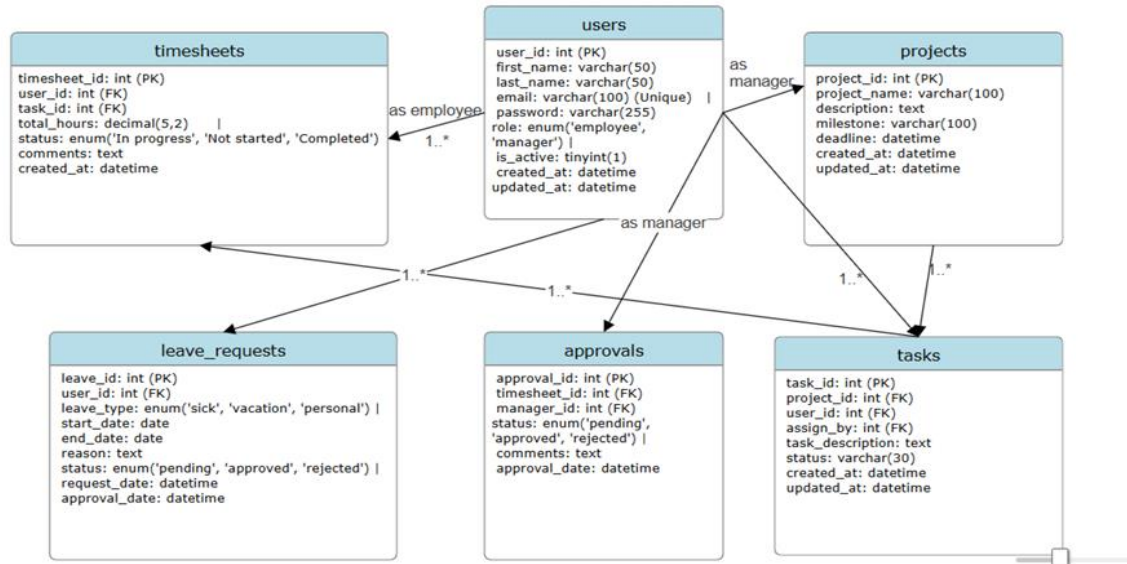


- Sequence Diagram



### 4.3 Database Design

We used MySQL (Xampp), a relational database is used to store user data, task details, and system logs efficiently.



### 4.4 Technology Stack

- **Front-End:** HTML, CSS, JavaScript.
- **Back-End:** Node.js, Express.
- **Database:** MySQL.

## 5. IMPLEMENTATION

### 5.1 Development Environment

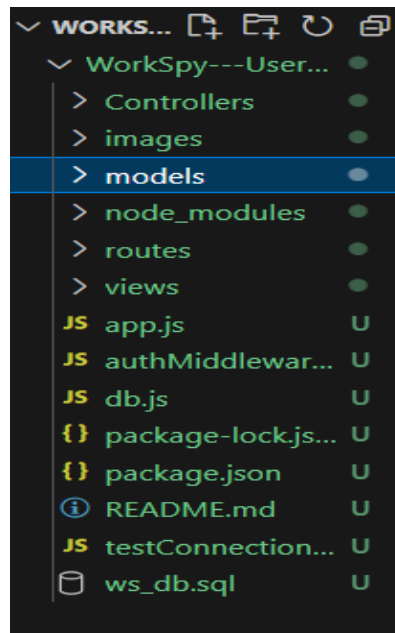
WorkSpy was developed using Visual Studio Code on Windows.

### 5.2 Code Structure

#### - Design Pattern:

WorkSpy follows the Model-View-Controller (MVC) design pattern. This ensures a clear separation of concerns, making the system more maintainable and scalable.





### 5.3 Key Features

- Dashboards for User Interfaces.
- Data visualization for productivity analysis (both Manager, Employee).
- Keep track of employee's daily progress by reviewing timesheets
- Achieving milestones to obtain a better idea about the progress of a project.

## 6. PROJECT PLAN

### 6.1 Project Timeline and Resource Allocation

Task 1: Research on how Project Management Apps look like and work

- Start Date: 2024-10-15
- Duration: 2 days
- Resource: Laiba

Task Continuation:

- Start Date: 2024-10-17
- Duration: 2 days
- Resource: Mahnoor

Laiba begins this task on 2024-10-15 and works on it for 2 days, finishing on 2024-10-16. Then, Mahnoor takes over and continues the task for another 2 days, ending on 2024-10-18.

---

## 2. Task 2: Setting up database

- Start Date: 2024-10-16
- Duration: 8 days
- Resource: Mahnoor

Mahnoor works on this task starting from 2024-10-16 and completes it by 2024-10-23.

---

## 3. Task 3: Design Document & SRS

- Start Date: 2024-10-18
- Duration: 5 days
- Resource: Anoosha

Anoosha works on creating the Design Document, Testing Document format & SRS from 2024-10-18 to 2024-10-23.

---

## 4. Task 4: Backend of Management Interface

- Start Date: 2024-10-23
- Duration: 15 days
- Resource: Laiba

Laiba begins this task on 2024-10-23 and works on it until 2024-11-06.

---

## 5. Task 5: Backend of User Interface

- Start Date: 2024-11-07
- Duration: 7 days
- Resource: Laiba

Laiba works on the Backend of the User Interface from 2024-11-07 to 2024-11-13.

---

## 6. Task 6: UML Diagrams

- Start Date: 2024-11-18
- Duration: 3 days
- Resource: Laiba

Task Continuation:

- Start Date: 2024-11-15
- Duration: 3 days
- Resource: Mahnoor

Laiba works on UML Diagrams from 2024-11-18 to 2024-11-20. Meanwhile, Mahnoor starts the task earlier on 2024-11-15 and completes it by 2024-11-17.

---

#### 7. Task 7: Management Interface UI

- Start Date: 2024-11-07
- Duration: 10 days
- Resource: Anoosha

Anoosha works on the Management Interface UI from 2024-11-07 to 2024-11-17.

---

#### 8. Task 8: User Interface UI

- Start Date: 2024-11-15
- Duration: 5 days
- Resource: Anoosha

Anoosha works on the User Interface UI from 2024-11-15 to 2024-11-19.

---

#### 9. Task 9: User Dashboard

- Start Date: 2024-11-18
- Duration: 5 days
- Resource: Mahnoor

Mahnoor works on the User Dashboard from 2024-11-18 to 2024-11-22.

---

#### 10. Task 10: Management Dashboard

- Start Date: 2024-11-21
- Duration: 4 days
- Resource: Anoosha

Anoosha works on the Management Dashboard from 2024-11-21 to 2024-11-24.

---

## 11. Task 11: Project Report

- Start Date: 2024-11-23
- Duration: 2 days
- Resource: Mahnoor

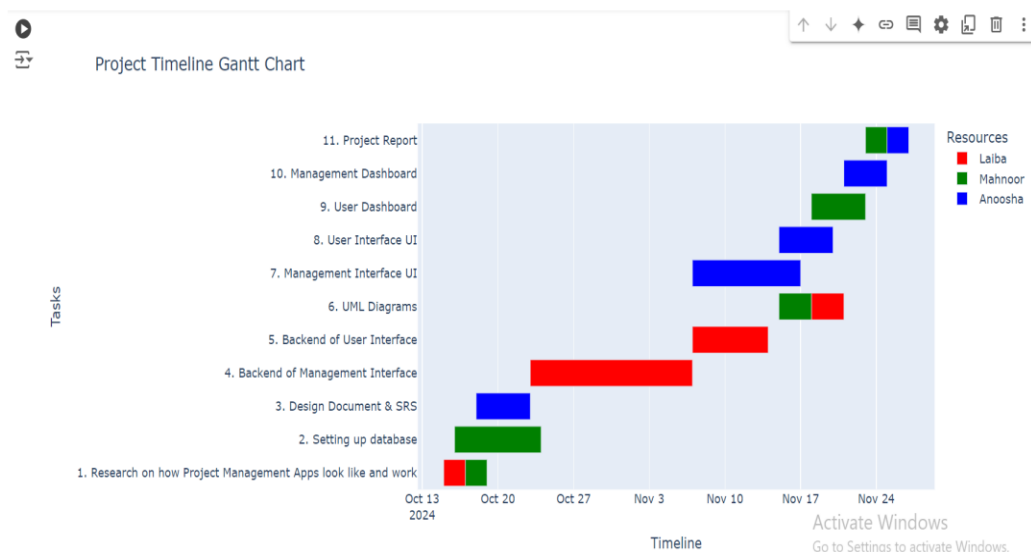
### Task Continuation:

- Start Date: 2024-11-25
- Duration: 2 days
- Resource: Anoosha

Mahnoor starts working on the Project Report from 2024-11-23 to 2024-11-24, and Anoosha continues the task from 2024-11-25 to 2024-11-26.

### Project Start and End Date:

- Project Start Date: 2024-10-15
- Project End Date: 2024-11-26



## 6.2 Risk Management

- **Risk Severity:** We classify risks as high, medium, or low severity, focusing more effort on those with the highest potential impact.
- **Preventive Measures:** For high-likelihood risks, we take proactive steps to prevent issues. For instance, we conduct regular code reviews and ensure clear communication among team members to avoid delays.

- **Contingency Plans:** For risks that have high impact but lower probability, we prepare contingency plans. For example, setting aside extra time in the schedule for unforeseen challenges.
- **Monitoring and Reporting:** Throughout the project, we regularly assess our progress and risks, adjusting mitigation strategies as needed to stay on track.
- **Status Updates:** We maintain a project status document, which helps in tracking task completion, resource allocation, and risk mitigation efforts.
- **Git for Version Control:** We use Git to ensure proper version control and collaboration. It allows us to track changes, resolve conflicts, and collaborate on code without disrupting each other's work.

## 7. Design Document

### 7.1 System Architecture

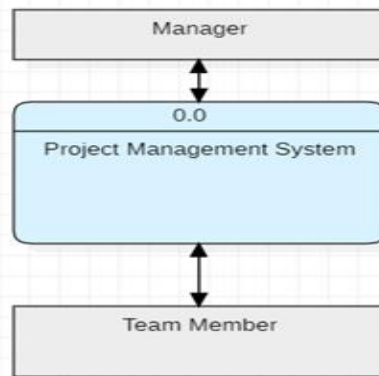
```

workspy/
├── controllers/
│   ├── userController.js      # User-related logic
│   ├── projectController.js   # Project-related logic
│   └── reportController.js    # Reports and charts
├── models/
│   ├── userModel.js          # User database queries
│   └── projectModel.js        # Project database queries
├── routes/
│   ├── userRoutes.js         # User API endpoints
│   └── projectRoutes.js       # Project API endpoints
├── views/
│   ├── dashboard.ejs         # Dashboard page
│   └── login.ejs              # Login page
├── public/
│   ├── styles.css            # Frontend styles
│   └── scripts/
│       └── charts.js          # Chart.js scripts
├── app.js                    # Entry point
├── db.js                     # MySQL database connection
├── package.json              # Dependencies and scripts
├── schema.sql                # Database schema
└── README.md                 # Documentation

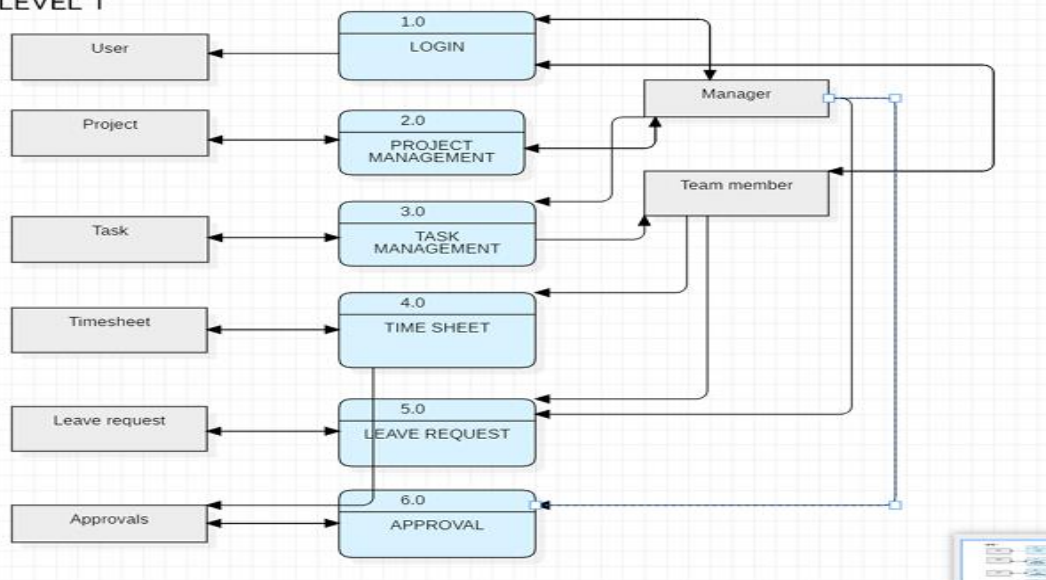
```

## 7.2 Data flow Design

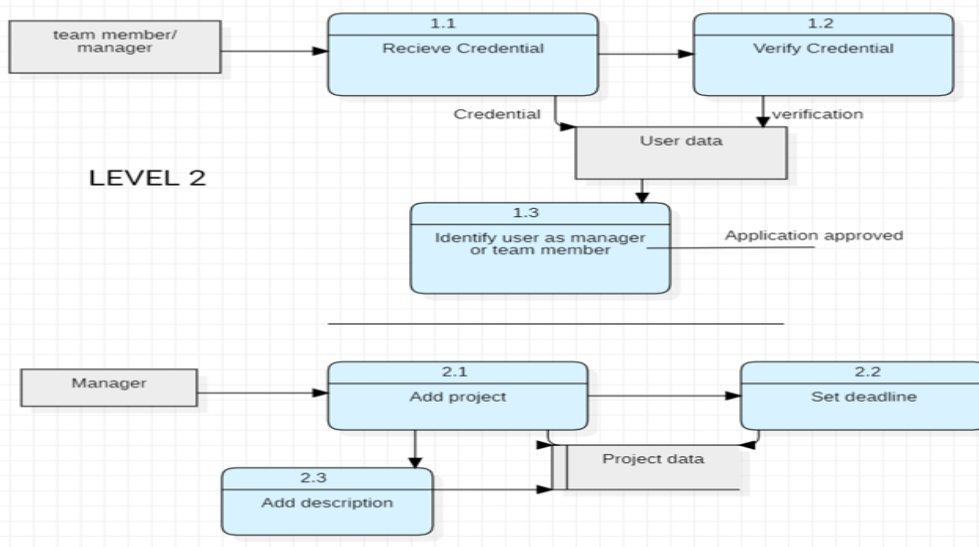
### LEVEL 0

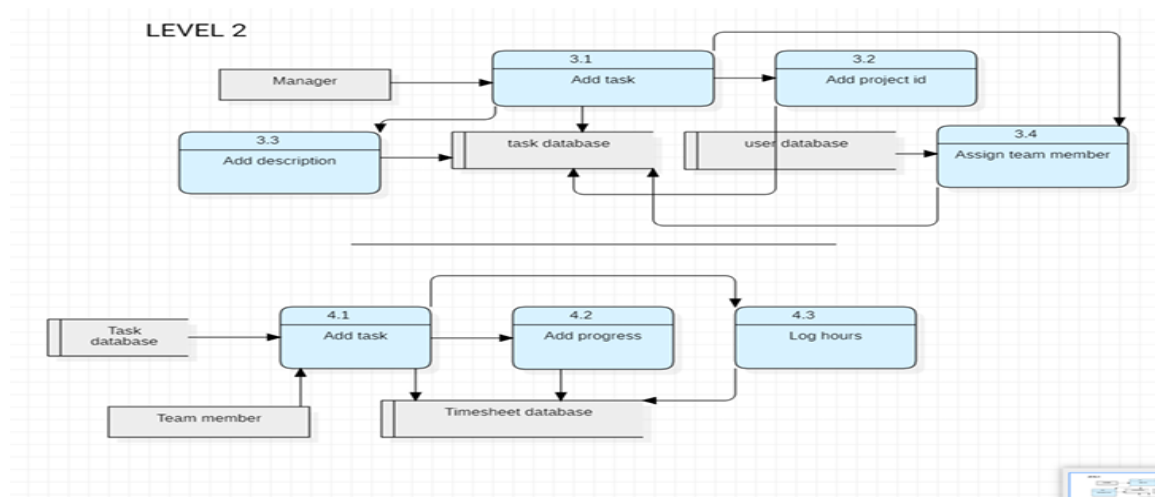


### LEVEL 1

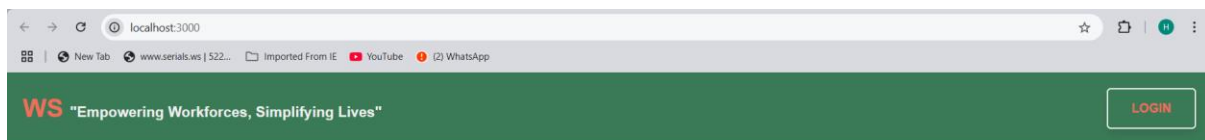


### LEVEL 2





## 7.3 User Interface Design

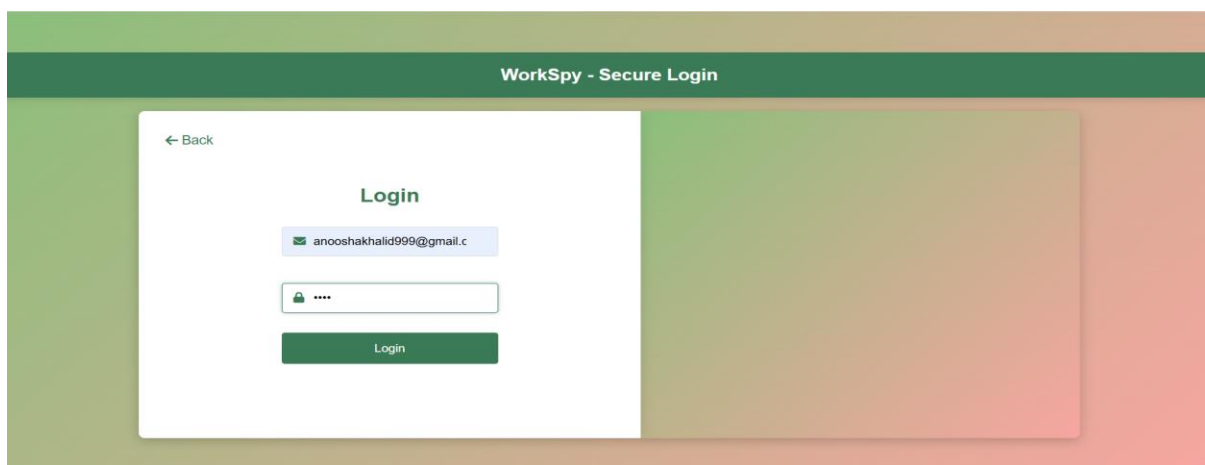


# WORKSPY

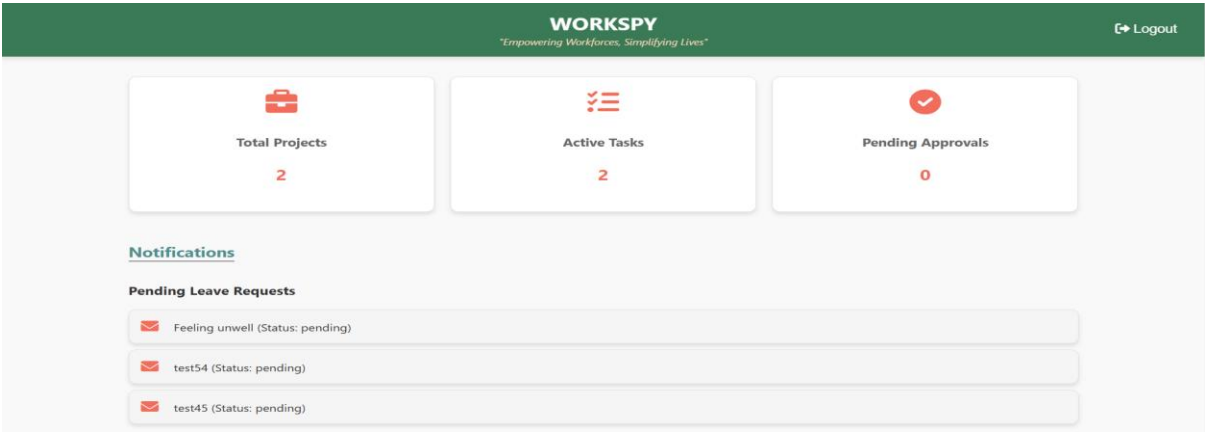
**Collaborate with team, & achieve best work**

-Team tasks at an organization can be a challenge, but clear goals, open communication, and expectations can lead to success and a tighter team-

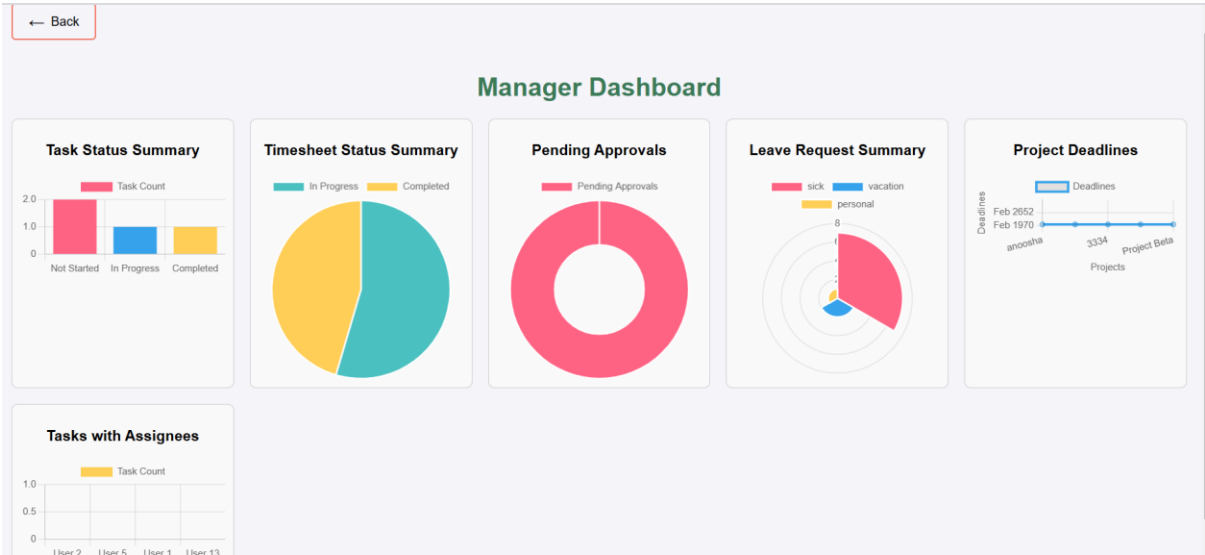
© 2024 WorkSpy. All rights reserved to 'M-A-L'.



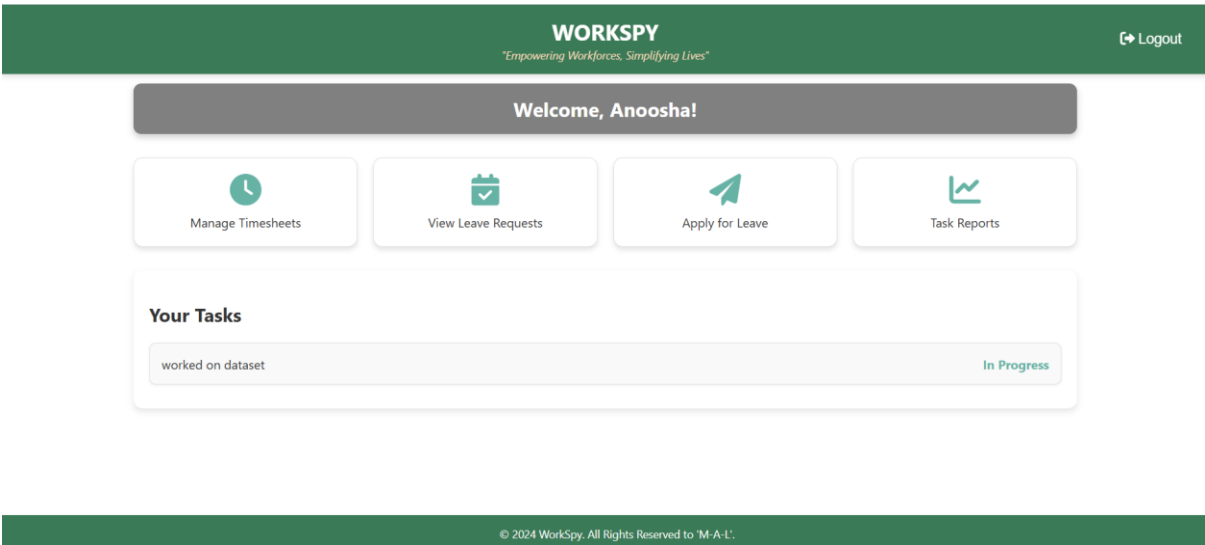
## Manager Dashboard



## Data Analytics- Report



## Employee Dashboard





## Timesheets

WORKSPY  
"Empowering Workforces, Simplifying Lives"

TimesheetsApply for LeaveLeave ListView Task Reports

Task ID:

Total Hours:

Status:Not Started

Comments:  
Add any additional details here...

SubmitBack

Submitted Timesheets

worked on dataset

12.00 hrs

In Progress

## Leave Applications form

Leave

Submit your leave application with complete details for prompt approval.

Leave Type

Select leave type

Start Date

dd/mm/yyyy

End Date

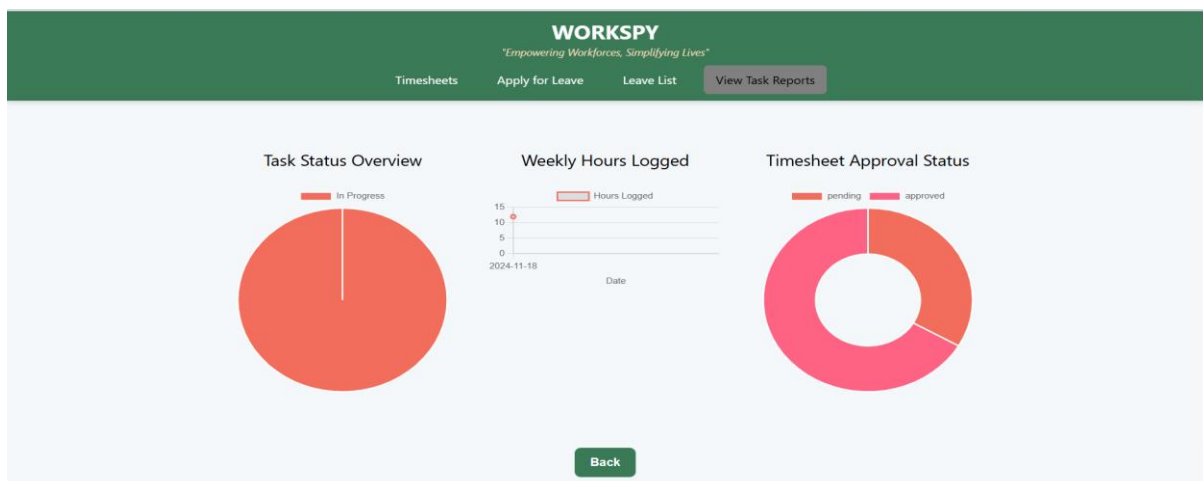
dd/mm/yyyy

Reason

Explain your reason here...

Submit Leave RequestBack

## Data Analytics- Report



## 9. RESULT AND ANALYSIS

### 8.1 Performance Metrics

-Authentication and Login Time:

- **Metric:** Authentication and login processes must complete within 2 seconds.

-Concurrent User Support:

- **Metric:** System must support at least 100 concurrent users without performance degradation.

-Data Retrieval Time:

- **Metric:** Data retrieval operations should complete within 2 seconds under normal load.

System Availability:

- **Metric:** System uptime must be 99.9% 24/7 hours.

Interaction Response Time:

- **Metric:** Response time for interactions (button clicks, form submissions) must be under 500 milliseconds.

## 10. CHALLENGES AND SOLUTIONS

### 1. Learning Node.js and Implementing the MVC Design Pattern

- **Challenge:** We worked with Node.js for the first time, which was a learning curve for the team. Specifically, we had to understand how to implement the Model-View-Controller (MVC) design pattern, which was crucial for maintaining clean, modular code.
- **Solution:** To overcome this challenge, we referred to multiple online resources, tutorials, and documentation. We also reviewed some open-source projects built with Node.js to understand how MVC was applied in real-world applications.

### 2. Struggling to Make the UI User-Friendly

- **Challenge:** Designing a user interface that was intuitive and user-friendly was a significant challenge. We had to ensure that the UI was easy to navigate and suitable for both the managers and users of the project management app.
- **Solution:** We conducted user research, studied UI design principles.

## 11. FUTURE WORK

- We can add data analytics on a large scale and make it a spotlight for our software.

- We can further add a proper pop-up notification functionality.
- Further we can add the functionality of invoices.
- Add features such as shared task boards, real-time file sharing, and in-app chat or video conferencing to improve team collaboration.

## 12. CONCLUSION

Our software project management application will provide a powerful and intuitive platform for both managers and team members, enhancing productivity and project management capabilities. By integrating advanced features and leveraging modern technology, we tried to deliver a solution that meets the needs of contemporary businesses and drives operational excellence.

## 13. REFERENCES

- Sommerville, I., Software Engineering.
- Pressman, R. S., Software Engineering: A Practitioner's Approach.

## 14. APPENDICES

### Appendix A: Glossary

- Node.js: A JavaScript runtime for building scalable applications outside the browser.
- MVC (Model-View-Controller): A design pattern separating data, logic, and user interface.

### Appendix B: Code Review Process

-Checklist:

- Node.js best practices.
- MVC implementation.
- Code readability.
- Error handling and performance checks.

### Appendix C: Tools Used

- Editor: Visual Studio Code.
- Version Control: Git and GitHub.

### Appendix D: Project Timeline

- Development: 1 month.
- Testing: 2 Weeks.

### Appendix E: Documentation Timeline

- Proposal: 1 week
- SRS: 1 week
- Design Document: 3 days

- Testing Document: 5 days
- Project Report: 5 days

#### Appendix F: Acknowledgements

- Thanks to my team Laiba and Mahnoor.
- Huge Appreciation for Sir Ali Akhtar.