

Weekly Time Tracker Report

Student Name: Sharmila Kumari Sharma

University of Arkansas Little Rock

IFSC 4398: Capstone Project II

Instructor: Bruce Bauer, Dr. Elizabeth Pierce

Date of Submission: April 29, 2024

Table of Contents

Introduction	3
Client Introduction / Organization Background.....	3
Organization Type:.....	3
Mission:	3
Customers/Beneficiaries:	3
History and Achievements:.....	3
Organization Size and Structure:	4
Challenges:	4
Additional Information:	4
Analysis Work Performed.....	4
Problem Statement.....	4
Current State Analysis	4
Desired Future State	5
Options for Achieving Future State	5
Project Plan	5
Design & Development Work Performed	6
IT Infrastructure	6
Data Architecture	6
Application Interface/Interactions	7
Code/Application Development	11
Professional Responsibilities	12
Ethical Responsibilities	12
Legal Responsibilities.....	13
Societal Responsibilities.....	13
Continuous Improvement	13
Accountability.....	13
Individual Work	13
Roles and Responsibilities	13
Challenges and Learning Experiences.....	14
Support and Integration.....	14
Integration with Existing Systems	14
User Support	15
Summary / Conclusions / Lessons Learned	15
Summary.....	15
Conclusions.....	15
Lessons Learned	16
References.....	16

Introduction

The need for an integrated time tracking solution is driven by the current digital transformation in the workplace. The Weekly Time Tracker project aims to fill the gap in the existing manual processes by introducing a digital solution that simplifies and automates time and attendance management, thereby improving productivity and reducing potential disputes over timekeeping. This project is a response to the increasing demand for digital solutions in the workplace and the need for more efficient and accurate time tracking systems.

Client Introduction / Organization Background

The weekly time tracker system is a project of the University of Arkansas at Little Rock, a dynamic organization dedicated to improving worker engagement and operational effectiveness through modern technology. The organization's main goals are to improve productivity by improving the time management process and offering accurate, transparent time tracking to all relevant stakeholders within the organization.

Organization Type:

UA Little Rock works as a non-profit organization. Our solutions serve companies of all sizes by providing them with an opportunity to more effectively handle staff schedules, track time, and process payroll.

Mission:

Our mission is to provide businesses with accurate, reliable, and user-friendly time-tracking software so they can efficiently manage their workforces. we aim to reduce the administrative load and create a trustworthy workplace so that staff members are able to focus on their main responsibilities.

Customers/Beneficiaries:

The system is designed to serve a wide range of users in the organization, including administrators who must integrate time-tracking data with other business systems, supervisors who need strong reporting and oversight capabilities, and part-time employees who want a simple way to log their hours.

History and Achievements:

UA Little Rock has been leading the way in delivering modern time management solutions since it was established. The organization has been recognized for its user-friendly software, which has significantly increased user happiness and time-tracking accuracy.

Organization Size and Structure:

The organization has established itself to respond rapidly to the changing demands of modern work environments. It employs a lot of people and has a specialized IT development team. The members of the leadership team have extensive backgrounds in business process optimization, software development, and human resources.

Challenges:

Adapting to the various and changing needs of different industries that need time-tracking systems is the organization's main difficulty. Another important area of focus is making sure the software remains user-friendly while being adaptable enough to meet different user needs.

Additional Information:

You can get more thorough details about UALR, its services, and customer reviews by visiting the official website or contacting the customer care department directly.

Analysis Work Performed

Problem Statement

In the contemporary work environment, efficient time tracking is crucial for both operational management and employee productivity. The current issue within the organization is the lack of an integrated system for tracking employee work hours, attendance, and managing leave requests. This leads to inaccuracies in payroll, inefficiencies in leave management, and a deficit in transparent employee performance data for supervisors. Additionally, the absence of a centralized platform results in the fragmented storage of employee data, making it challenging for administrators to oversee and streamline operations effectively.

Current State Analysis

Current processes are disjointed, relying heavily on manual input, leading to inefficiencies and inaccuracies. Data is inconsistently formatted and stored, making it difficult for administrative analysis and reporting. This lack of standardization and automation leads to delays in processing time data, inaccuracies in payroll and benefits calculations, and potential disputes over timekeeping.

- **Data:** Inconsistently formatted time logs, scattered across multiple mediums.
- **Process:** Manually intensive with no uniform procedure for clocking in/out or leave requests.
- **People:** Employees and supervisors and admin are engaged in labor-intensive tracking and approval processes.

Desired Future State

A unified system where:

- Data is centrally stored and structured, allowing for easy retrieval and analysis.
- The process for logging time and managing leave is standardized and automated, reducing the potential for errors and disputes.
- All roles have streamlined tasks appropriate to their responsibilities, improving efficiency and productivity.
- The technology enables real-time processing and accessibility, allowing for timely updates and reports.

Options for Achieving Future State

To reach the desired future state, the following options were considered:

- 1. Custom Web Application Development:** Tailoring a solution to meet the exact needs of our organization, with scalability and customizability.
- 2. Off-the-Shelf Software:** Implementing a commercial time-tracking solution, potentially requiring less initial development but possibly incurring ongoing licensing fees.
- 3. Open-Source Software:** Utilizing a free, open-source time tracking tool that can be customized, albeit with a potential need for in-house or contracted technical support.

After careful consideration, the decision was to pursue the first option: developing a custom web application. This approach aligns best with the organization's requirement for flexibility, data privacy, and the integration of specific features tailored to our operational workflow.

Project Plan

The development of the Weekly Time Tracker System was planned as follows:

Phase 1: Requirements Gathering and System Design (2 weeks)

- Draft initial system requirements.
- Create a system design and architecture plan.

Phase 2: Development and Implementation (8 weeks)

- Set up development environment and version control system.

- Implement the backend infrastructure and database.
- Develop the frontend user interface for employees, supervisors, and administrators.
- Integrate features such as clock-in/out, leave management, and reporting dashboards.

Phase 3: Testing and Deployment (4 weeks)

- Perform unit testing, system testing, and user acceptance testing.
- Conduct a beta release and gather feedback.
- Roll out the system organization-wide.

Phase 4: Evaluation and Maintenance (Ongoing)

- Monitor system performance and gather user feedback.
- Perform regular maintenance and updates.
- Evaluate the system's impact on productivity and efficiency.

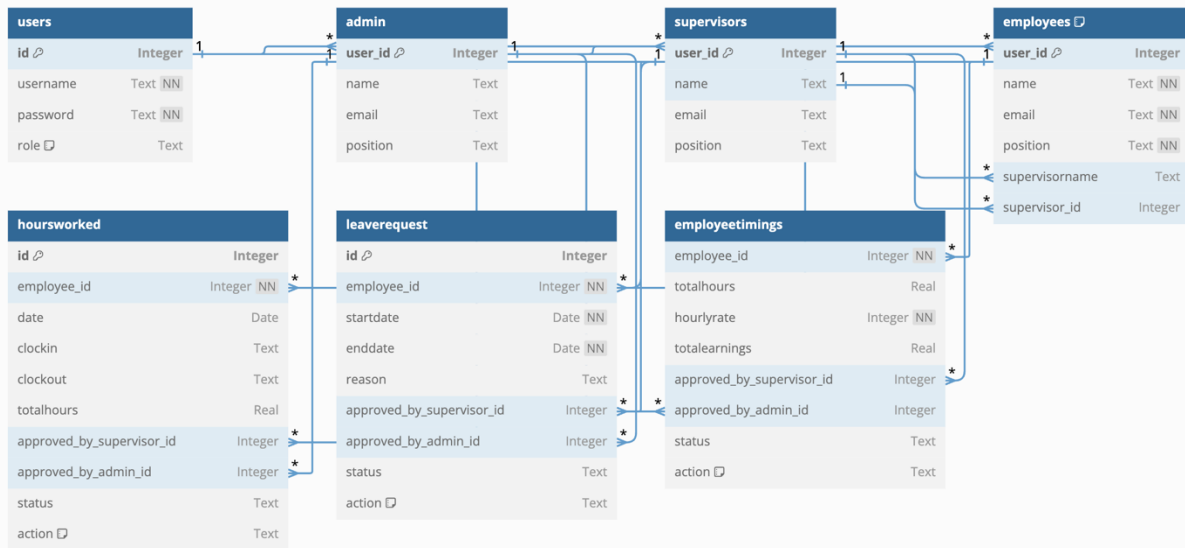
Design & Development Work Performed

IT Infrastructure

The weekly time tracker system is designed to be a lightweight, web-based application that requires minimal IT infrastructure, ensuring ease of access and maintainability. The front-end interface is hosted via the local host of the school's PC, allowing for scalability and reliability. The back end, comprising the application logic and data handling using Python Flask, is implemented on a serverless architecture to manage compute resources efficiently and handle variable workloads. Authentication services are used to ensure secure access to the system, and user roles are defined to control the level of access among employees, supervisors, and administrators.

Data Architecture with Entity-Relationship Diagram:

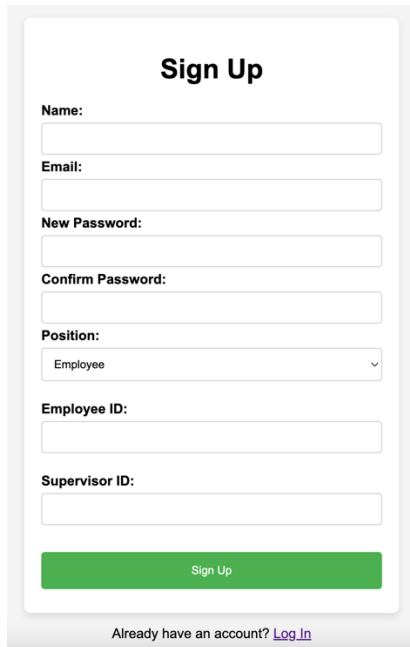
The application's data architecture is constructed to ensure data integrity and facilitate quick retrieval of time-tracking information. It employs SQLite3 to maintain structured data, including user profiles, time entries, and leave requests. The database schemas are designed to capture essential details such as employee ID, clock-in and clock-out timestamps, total hours worked, leave start and end dates, and leave reasons. The relational nature of the database also allows for generating reports and analytics, such as total hours worked per week per employee, which are crucial for payroll processing.



Application Interface/Interactions

The user interface, developed using HTML, CSS, and JavaScript, is tailored to the needs of three distinct types of users:

For the signup interface, new employees are required to enter their full name, email and password. There is also a field to confirm password, ensuring accuracy in password entry. They can select their position as employees in the dropdown menu. Additionally, they can enter their employee ID and supervisor ID, which are crucial for roles and hierarchy management within the application. Upon completing the form, they can submit it by clicking the 'Sign Up' button, which redirects them to the login page and saves their information to the employees table in the database to add them into the system. Additionally, at the bottom of the page, there's a footer with a link to the login page for users who have already setup an account and are registered into the system.



The Sign Up form is a vertical stack of input fields within a white card. It starts with a title 'Sign Up' in bold. Below it are fields for Name, Email, New Password, and Confirm Password, each with a label and a text input. The Position field is a dropdown menu with 'Employee' selected. Below that are fields for Employee ID and Supervisor ID. A green 'Sign Up' button is at the bottom of the form. At the very bottom of the card, there is a link 'Already have an account? Log In'.

Sign Up

Name:

Email:

New Password:

Confirm Password:

Position:

Employee ID:

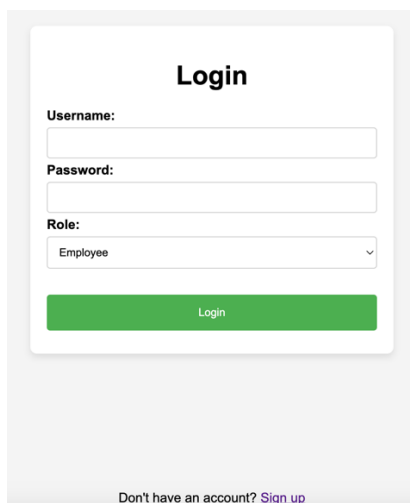
Supervisor ID:

[Sign Up](#)

Already have an account? [Log In](#)

When a user visits the login interface, they'll see a form within a styled card on their screen, centered both horizontally and vertically. The page itself has a light gray background, while the login form is on a white background with rounded corners and a subtle shadow for a modern, clean look.

To log in, a user needs to enter their Username, Password, and select their role from the dropdown menu. They'll find a text input field labeled "Username" where they should type their username. Below the username field, there's a password input field labeled "Password" for entering their secure password. There's a dropdown menu labeled "Role" with options for "Employee", "Supervisor", and "Administrator". The user must select the role that applies to them. After filling out these fields, the user can submit the form by clicking the green "Login" button. The form data is sent to the server which redirects them to their respective dashboard after fetching their data from the database. Additionally, at the bottom of the page, there's a footer with a link to a signup page for users who aren't yet registered into the system.



The Login form is a vertical stack of input fields within a white card. It starts with a title 'Login' in bold. Below it are fields for Username, Password, and Role, each with a label and a text input. The Role field is a dropdown menu with 'Employee' selected. A green 'Login' button is at the bottom of the form. At the very bottom of the card, there is a link 'Don't have an account? Sign up'.

Login

Username:

Password:

Role:

[Login](#)

Don't have an account? [Sign up](#)

An employee is redirected to their respective dashboard after entering their credentials in the login page.

Located on the left side of the screen, this fixed panel offers links to different sections like "Employee Dashboard", and "Leave Requests". There is also a "Logout" link for users to exit their session which redirects them to the login page. At the top of the main content area, the dashboard greets the user by name and displays the name of their supervisor. This is dynamically filled in with template tags indicating server-side rendering. The worked hours table displays the record of the employee's work hours with details such as date, clock-in and clock-out times, total hours worked, and status (pending, approved, etc.). Two buttons allow the employee to clock in or clock out. Clicking these buttons triggers AJAX requests to server-side endpoints, which allows the employee to clock in and clock out. Success or failure messages are displayed, and the table updates accordingly without needing to reload the page. Clock-in and clock-out processes are handled asynchronously, updating the user interface in real-time based on the responses from the server. Employees can submit leave requests through the leave request form that collects the start date, end date, and reason for the leave. Upon submission, this information is sent to the server for processing. After the server processing, the server sends all the submitted information by the employees to the supervisor's dashboard for approval.

Employee Dashboard

Welcome, John Smith

Supervisor: James Anderson

Worked Hours

Date	Clock In	Clock Out	Total Hours	Status
4/23/2024	1:35:48 PM	1:35:51 PM	0.00	Pending

Clock InClock Out

Leave Request

Start Date:

End Date:

Reason:

Family Vacation.

Submit

The supervisor dashboard is designed to help supervisors monitor and manage employee activities such as clock-in/out times and leave requests. Here's a general overview of the main features and interactions provided in the dashboard:

A fixed sidebar on the left allows supervisors to navigate through different sections like "Supervisor Dashboard", "Employee Clock-In/Out Times," and "Leave Requests". It also includes a logout link for session termination.

To the right of the navigation sidebar, this area displays various sections of content including tables for employee clock-in/out times and leave requests. Employee Clock-In/Out time is a table with details such as Employee ID, Date, Clock In, Clock Out, Total Hours, Status of the entry, and Actions (Approve or Deny). Supervisors can approve or deny the clock-in/out entry via buttons in the Actions column. Another table shows leave requests detailing Employee ID, Start and End Date, Reason for leave, Status, and Action buttons to approve or deny the request.

Each entry in the clock-in/out and leave request tables includes "Approve" and "Deny" buttons. Supervisors can click these to send a status update request to the server for the corresponding employee entry. Using AJAX calls, the dashboard fetches the latest data for clock-in/out times and leave requests from the server when the page loads. It also allows real-time updating of this data without needing to reload the page. After actions (approve or deny), alerts provide feedback to the employees, and the tables are updated to reflect the new status of entries.

Supervisor Dashboard						
Employee Clock-In/Out Times						
Employee ID	Date	Clock In	Clock Out	Total Hours	Status	Action
103	2024-04-21	17:02	17:06	0.066666666666667	None	<button>Approve</button> <button>Deny</button>
103	2024-04-21	17:14	17:14	0.0	None	<button>Approve</button> <button>Deny</button>
101	2024-04-21	18:24	18:24	0.0	pending	<button>Approve</button> <button>Deny</button>
102	2024-04-21	19:37	19:37	0.0	None	<button>Approve</button> <button>Deny</button>
102	2024-04-21	20:14	20:14	0.0	None	<button>Approve</button> <button>Deny</button>
102	2024-04-21	21:04	21:04	0.0	None	<button>Approve</button> <button>Deny</button>
102	2024-04-21	21:04	21:05	0.02	None	<button>Approve</button> <button>Deny</button>
102	2024-04-21	21:05	21:05	0.02	None	<button>Approve</button> <button>Deny</button>
102	2024-04-21	21:05	None	None	None	<button>Approve</button> <button>Deny</button>
101	2024-04-23	13:35	13:35	0.0	None	<button>Approve</button> <button>Deny</button>
Leave Requests						
Employee ID	Start Date	End Date	Status	Action		
	2024-04-18	2024-04-19	dffg			
	2024-04-18	2024-04-19	jjjh			

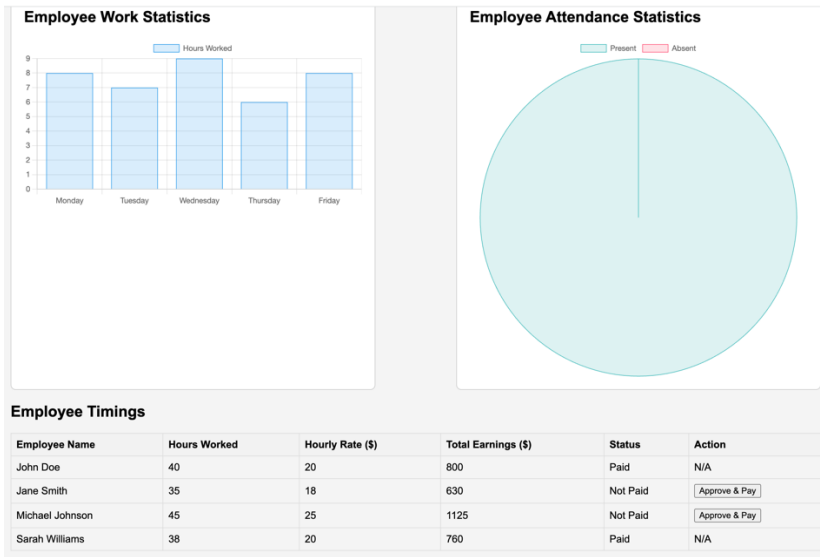
Administrators have access to a broader dashboard that aggregates data across the organization for oversight and payroll purposes. The interface design uses CSS styles to make the layout responsive and visually appealing. There's a container for central content with a white background, rounded corners, and a drop shadow for a modern look. The navigation sidebar and charts have specific styling to ensure the dashboard is user-friendly and functional.

On the left side, there's a fixed navigation sidebar with links to different sections of the admin dashboard such as Admin Dashboard, Statistics, Employee Timings, and a logout option.

Employee Work Statistics displays a bar chart that shows work-related statistics such as hours worked. This is visualized using Chart.js, a popular JavaScript library for rendering charts.

Employee Attendance Statistics shows a pie chart for attendance details like presence and absences.

Employee Timings table displays details about each employee including their ID, name, total hours worked, hourly rate, total earnings, status, and an action button to approve and pay. The data for this table is fetched dynamically using JavaScript and populated into the table body. Each row in the employee timings table includes a button that allows you to approve and process payroll for that employee. The dashboard fetches statistical data from two API endpoints for work and attendance stats. It then renders this data into charts using Chart.js, configured for bar and pie charts respectively.



Code/Application Development

The codebase is developed using a combination of HTML, CSS, and JavaScript, with the inclusion of SQLite3 Database. The application uses AJAX for asynchronous server communication and emphasizes code modularity for maintainability.

The Python Flask application serves as a backend for managing the weekly time tracker system, facilitating various functionalities centered around user authentication, time tracking, and leave management. It uses SQLite as its database to store data about employees, their work hours, and leave requests.

The Flask app configuration is set up with a secret key for session security, session type, and database location. It then initializes session handling, CORS for handling requests from different origins, and Bcrypt for password security.

The core functionality of this server includes several web routes that handle different aspects of employee management:

- 1. Database Initialization and Connection:** A function to create database tables from a script and another to establish database connections are implemented, ensuring that the app interacts with the database effectively.
- 2. User Authentication and Management:** The code supports user sign-up and login functionalities. During signup, user details are collected and stored in the database, including hashing passwords for security. The login functionality checks user credentials and manages user sessions, redirecting users to appropriate dashboards based on their roles (admin, supervisor, or employee).
- 3. Dashboard Management:** Depending on the user's role, different dashboards are served. Each role has access to specific data: employees can view their work hours, supervisors can manage

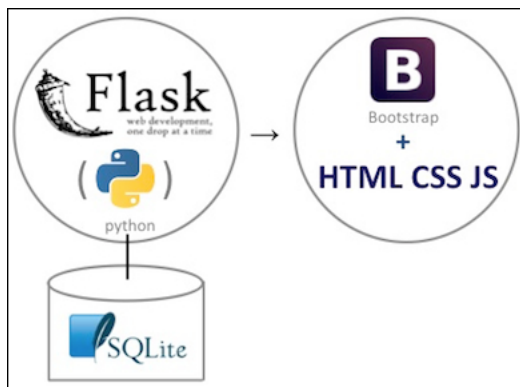
their team's work hours and leave requests, and admins can approve or deny work hours and leave requests across all users.

4. Time Tracking: Employees can clock in and clock out, with the server recording these times. It calculates total work hours based on these timestamps.

5. Leave Management: Employees can submit leave requests, which can then be approved or denied by supervisors or admins.

6. Reporting and Statistics: Additional routes provide summaries of work hours and leave statuses, which could be used for generating reports or analytics.

Throughout, the server handles exceptions and errors, ensuring that any database operations that fail do not crash the application but instead provide useful error messages. This makes the system robust and user-friendly, offering detailed feedback for troubleshooting.



Professional Responsibilities

In the context of the Weekly Time Tracker project, professional responsibilities include social, legal, and ethical duties to ensure that the system is developed and implemented with integrity, transparency, and respect for all those involved. These responsibilities are integral to fostering a culture of accountability and efficiency within the organization.

Ethical Responsibilities

- **User Privacy:** Safeguarding user privacy by ensuring that personal and sensitive data are protected. Implementing robust security protocols to prevent unauthorized access or breaches.
- **Transparency:** Providing clear communication about how employee data is collected, stored, and used. Ensuring that users have access to their own data.
- **Integrity:** Upholding high standards of honesty in reporting work hours. The system should promote accuracy and fairness in time tracking.

Legal Responsibilities

- **Compliance:** Adhering to legal standards related to labor laws, privacy regulations such as General Data Protection Regulation (GDPR), and other relevant legal requirements.
- **Documentation:** Maintaining comprehensive records to support audits or inspections. Ensuring that time tracking records are accessible and transparent.

Societal Responsibilities

- **Inclusivity:** Designing a user interface that is intuitive and accessible for a diverse workforce with varying levels of technical proficiency.
- **Well-being:** Considering the well-being of employees by avoiding features that could lead to overwork or stress. Promoting a balanced work-life environment.
- **Environmental Impact:** Minimizing the environmental footprint of the digital solution by optimizing energy efficiency in servers and other infrastructure components.

Continuous Improvement

- **Feedback Mechanism:** Implementing mechanisms for collecting feedback from users (employees, supervisors, administrators) for continuous improvement.
- **Training & Support:** Providing adequate training resources for users and offering support channels for addressing issues or concerns promptly.

Accountability

- **Monitoring & Reporting:** Regularly monitoring system performance; ensuring accurate reporting tools are available for auditing employee work hours effectively.

Individual Work

As this was an individual project, the responsibility for all aspects of the project fell on me. This included project management, requirements gathering, system design, development, testing, and deployment.

Roles and Responsibilities

As the sole member of the project, I was responsible for overseeing the project's progress, ensuring that it stayed on schedule and within budget. I facilitated communication with stakeholders, managed risks, and resolved any conflicts that arose

during the project. I was responsible for the technical implementation of the Weekly Time Tracker. I worked on both the front-end and back-end development, ensuring that the application was functional, efficient, and user-friendly. I focused on the user interface and overall user experience of the application. I designed the layout of the dashboards and ensured that the application was intuitive and easy to use. I was responsible for testing the application to ensure it met all requirements and functioned as expected. I conducted various types of testing, including unit testing, integration testing, and user acceptance testing. I was responsible for creating and maintaining all project documentation, including the project plan, requirements specifications, and user manuals.

Challenges and Learning Experiences

Working on this project individually presented unique challenges and learning opportunities. Without a team to rely on, I had to quickly adapt and learn new skills to complete the project successfully. This included learning new programming languages, understanding user experience design principles, and managing project timelines and resources effectively.

Despite the challenges, this project provided a valuable learning experience. It allowed me to gain a comprehensive understanding of all aspects of a software development project, from initial planning to final deployment. It also reinforced the importance of time management, problem-solving, and adaptability in successfully completing a project.

Overall, the experience of working on this project individually has been rewarding and educational, providing me with a solid foundation for future projects. It has also underscored the importance of continuous learning and adaptability in the field of software development.

Support and Integration

The Weekly Time Tracker system is designed to integrate seamlessly with the organization's existing IT infrastructure. It is a web-based application, which means it can be accessed from any device with an internet connection, without the need for additional hardware or software installations. This ensures that the system is easily accessible to all employees, regardless of their location.

Integration with Existing Systems

The system is designed to integrate with the organization's existing HR and payroll systems. This allows for the automatic transfer of time-tracking data to these systems,

eliminating the need for manual data entry and reducing the risk of errors. The system also supports integration with the organization's single sign-on (SSO) system, providing a seamless login experience for users and enhancing security. Whenever new employees are hired, they have to sign up then only they can log in.

User Support

To ensure that all users can effectively use the system, a comprehensive user manual is provided, detailing all features and functionalities of the system. This includes step-by-step instructions for common tasks such as clocking in and out, submitting leave requests, and reviewing timesheets.

In addition to the user manual, online support is available to assist users with any issues they may encounter while using the system. This includes a dedicated support email and a live chat feature on the system's website. The support team is trained to resolve issues quickly and efficiently, ensuring minimal disruption to the user's workflow.

In conclusion, the support and integration plan for the Weekly Time Tracker system is designed to ensure a smooth implementation and continued operation of the system, providing users with the tools they need to effectively manage their time and leave requests. This plan underscores the organization's commitment to leveraging technology to improve efficiency and accountability in the workplace.

Summary / Conclusions / Lessons Learned

Summary

The project's completion has resulted in a marked improvement in the management of employee time records, an increase in productivity through focused and responsible time allocation, and a reduction in time-related disputes due to the system's transparency and accuracy. The implementation of the Weekly Time Tracker underscores the organization's commitment to operational excellence and the leveraging of technology to foster a culture of integrity and professionalism in timekeeping practices.

Conclusions

The Weekly Time Tracker system has proven to be an essential tool for both internal and remote teams in the increasingly digital workplace. The system's successful implementation has demonstrated the value of investing in technology to improve workforce management processes. The organization has seen tangible benefits in terms

of improved productivity, reduced administrative load, and increased transparency in timekeeping practices.

Lessons Learned

The development of the Weekly Time Tracker system provided several valuable lessons:

1. **User-Centric Design:** The importance of designing the system with the end-user in mind was a key lesson. By focusing on the needs of the users (employees, supervisors, and administrators), the system was able to provide a user-friendly interface and features that met their specific needs.
2. **Iterative Development:** The project reinforced the value of an iterative development approach. Regular feedback loops with stakeholders allowed for continuous improvement and ensured that the final product met the organization's needs.
3. **Testing and Quality Assurance:** The project highlighted the importance of robust testing methods in ensuring a quality product. The use of unit tests, integration tests, and user acceptance testing (UAT) helped to identify and resolve issues early in the development process.
4. **Change Management:** Implementing a new system requires careful change management. Training sessions and support resources were crucial in helping users adapt to the new system.

In conclusion, the Weekly Time Tracker project has been a success, delivering a system that meets the organization's needs and contributes to improved workforce management. The lessons learned from this project will be valuable in guiding future technology initiatives within the organization.

Hours worked: 300

References

1. "Workforce Management." Wikipedia, Wikimedia Foundation, www.wikipedia.org/wiki/Workforce_management.
2. "Relational Database Design." W3Schools Online Web Tutorials, www.w3schools.com/sql/sql_intro.asp.

3. “Software Testing Fundamentals.” Tutorialspoint,
www.tutorialspoint.com/software_testing_dictionary/index.htm.
4. “Project Management.” Project Management Institute,
www.pmi.org/learning/library.