A PROJECT REPORT

**ON**

**“TIMESHEET ”**

**Team Name : Tech Army**

**Submitted to: Submitted by:**

**Archana K**

**Monisha A**

**Pavithra A**

**Saranya M**

**Abstract**

The Timesheet Project is a comprehensive application designed to streamline the process

of tracking and managing employee work hours. The Timesheet Project is designed to

streamline the process of logging, submitting, and approving employee working hours.

Built using Django for the backend and React.js for the frontend, the project is segmented into

Four core modules: Employee, Lead, and Manager and Human Resource(HR) . Each module

caters to the unique needs of its users, ensuring a seamless workflow for logging, submitting, and

approving employee working hours.The Employee Module allows users to log in securely,

submit their weekly work hours, and view their submitted timesheets in a clear, tabular format.

Employees can also log out safely, ensuring that their session remains secure. This module focuses

on ease of use for employees to manage their work hour submissions and track their records

efficiently. Leads can log in to their accounts, view comprehensive details about all employees,

and make decisions on timesheet approvals. They have the authority to approve or reject

submitted timesheets, with all actions being accurately recorded in the database to maintain an

organized record of approvals and rejections.Managers have similar capabilities to leads, providing

them with the ability to log in, access detailed employee information, and handle timesheet

approvals or rejections. Managers can perform all tasks available to leads, with their decisions

also being recorded in the timesheet database to ensure transparency and accountability.

HR has comprehensive access to view all employee details, timesheets, and a salary calculation

page to determine pay based on working hours and leave days. All modules ensure accurate data

recording and processing in the timesheet database.

**TABLE CONTENT**

|  |  |  |
| --- | --- | --- |
| s.no | Content | Page no |
| 1. | Abstract | 2 |
| 2. | Introduction   * + - Project overview     - Key objective     - Use Case Diagram | 4 |
| 3. | Module   * + Employee Module   + Lead Module   + Manager Module   + HR Module | 10 |
| 4. | System Architecture | 14 |
| 5. | Software Requirements | 16 |
| 6. | User Interface Requirements | 21 |
| 7. | Implementation Plan | 33 |
| 8. | Data Requirements | 36 |
| 9. | Workflow | 37 |
| 10. | Future Enhancement | 55 |
| 11. | Conclusion | 59 |

**Introduction**

The Timesheet Project is an advanced application designed to facilitate the management and

tracking of employee work hours within an organization. This comprehensive system is

built using Django for the backend and React.js for the frontend, ensuring a robust, secure,

and user-friendly experience. The project is divided into four main modules—Employee,

Lead, and Manager,HR —each tailored to meet the specific needs of its users.The primary

objectives of the Timesheet Project include secure authentication and authorization,

accurate time tracking and submission, efficient timesheet management, and a streamlined

timesheet approval workflow. By addressing these objectives, the project

aims to enhance the efficiency and accuracy of timesheet management processes,

ultimately contributing to better operational oversight and productivity.

**Project Overview**

This project is designed to manage and track employee timesheets. It has four

main modules: Employee, Lead, and Manager and Human Resources(HR).

Each module includes specific functionalities for managing timesheets, employee details, and approvals and payroll calculation . The project uses Django for the backend and React.js

for the frontend**.**

**Key Objectives**

**Secure Authentication and Authorization:**

* Implement robust login mechanisms for employees, leads, and managers to ensure

secure access to the system.

* Ensure that users can log out safely, maintaining the security of their sessions.
* A primary objective of the Timesheet Project is to ensure secure authentication

and authorization across all user roles.

This involves implementing robust login mechanisms for employees, leads, and

managers to protect sensitive data and maintain the integrity of the system.

* Each user is required to authenticate their identity using secure credentials, such as

email and password, which helps prevent unauthorized access.

* Additionally, the system includes a secure logout feature that ensures users can

safely terminate their sessions, preventing unauthorized access to their accounts and

protecting their personal and work-related data.

* By prioritizing security in authentication and session management, the project

aims to provide a safe and reliable environment for managing timesheet information.

**Accurate Time Tracking and Submission:**

* Develop an intuitive Update Timesheet page for employees to accurately submit their

working hours.

* Store all submitted timesheet data reliably in the timesheet database to maintain

accurate records.

* + - * Another critical objective is to enable accurate time tracking and submission.
* The Update Timesheet page is designed to be intuitive and user-friendly,

allowing employees to easily input their working hours for each day of the week.

* This page includes well-organized input fields and a submission button,

Ensuring that employees can record their hours accurately without confusion.

* Once submitted, all timesheet data is reliably stored in the timesheet database.
* The database is engineered to handle and preserve the integrity of the data,

ensuring that records of working hours are accurate and up-to-date.

* This objective is crucial for maintaining an accurate history of employee work

hours and supporting effective timesheet management.

**Efficient Timesheet Management:**

* Create a View Timesheet page for employees to easily access and review their

submitted timesheet data in a table format.

* Ensure that the data fetched from the timesheet database is displayed

accurately and comprehensively.

* Efficient timesheet management is achieved through the creation of dedicated

pages for viewing and managing timesheet data.

* The View Timesheet page allows employees to access and review their submitted

timesheets in a clear and organized table format.

* This page retrieves data from the timesheet database and displays it comprehensively,

making it easy for employees to monitor their recorded hours and ensure accuracy.

* The system is designed to present this data without errors, providing employees with a

reliable overview of their time records.

* Efficient management is further supported by the accurate and comprehensive

display of data, which facilitates better tracking and oversight of work hours.

* This objective ensures that both employees and managers can manage timesheet

information effectively, contributing to overall operational efficiency.

**Streamlined Timesheet Approval Workflow**:

* Develop a Timesheet Approval page for both leads and managers where they

can review employee-submitted timesheets.

* Implement functionality to approve or reject timesheets with a single click,

and store the decision in the timesheet database.

ensuring that leads and managers can efficiently review and manage submitted

timesheets.

* To achieve this, a dedicated Timesheet Approval page is developed for both

leads and managers. This page is designed to provide a comprehensive view of all

submitted timesheets, allowing supervisors to quickly and accurately assess the

work hours recorded by their team members.

* The Timesheet Approval page is equipped with functionality that enables leads and

managers to approve or reject timesheets with a single click.

* Each submitted timesheet is displayed with relevant details, such as the employee's

name, the dates covered, and the hours worked each day.

* Leads and managers can review this information and make informed decisions

about the accuracy and validity of the recorded hours.

* The interface includes intuitive buttons for approval and rejection,

simplifying the decision-making process.

* When a timesheet is approved or rejected, the system records the decision in the

timesheet database.

* This action ensures that there is a clear and auditable trail of all timesheet

approvals and rejections, maintaining the integrity of the records.

**Project Structure**

│Tech\_Army

├── backend/

│ ├── manage.py

│ ├── tech\_army\_app/

│ │ ├── \_\_init\_\_.py

│ │ ├── admin.py

│ │ ├── apps.py

│ │ ├── models.py

│ │ ├── serializers.py

│ │ ├── urls.py

│ │ └── views.py

│ └── Tech\_Army/

│ ├── \_\_init\_\_.py

│ ├── settings.py

│ ├── urls.py

│ └── wsgi.py

│

├── frontend/

│ ├── public/

│ ├── src/

│ │ ├── components/

│ │ │ ├── Employee/

│ │ │ ├── Lead/

│ │ │ ├── Manager/

|\_\_\_\_HR/

│ │ ├── App.js

│ │ ├── index.js

│ │ ├── services/

│ │

│ └── package.json

│

├── requirements.txt

└── README.md

**Use Case Diagram**

**Login**

**Update timesheet**

**Add employee**

**Employee**

**Number of Leave days**

**Edit**

**View Timesheet**

**Delete**

**Update details**

**Admin**

**Lead**

**Approve or reject timesheet**

**View Employee**

**,lead ,manager ,HR**

**timesheet**

**View Employee Details**

**Manager**

**Project assignment**

**Payroll calculation**

**HR**

**logout**

**Modules**

**Employee Module**

**Overview:**

The Employee Module is to provide employees with the tools they need to manage

their own timesheets. Employees can securely log in to access their personal timesheet data.

Once authenticated, they can enter their weekly working hours via the Update Timesheet Page,

which simplifies the process of recording and updating their hours. The module also includes

a View Timesheet Page, where employees can review their submitted timesheets in a clear,

tabular format. This allows them to monitor their recorded hours and make any necessary

adjustments. To ensure data security, the module features a Logout function that allows

employees to safely end their session.

The Employee Module is designed to facilitate employees' interaction with their timesheets.

* **Login Page**: Employees start by logging into the system using secure credentials (email and password). This authentication process ensures that only authorized users can access their timesheet data.
* **Update Timesheet Page**: After logging in, employees can navigate to the Update Timesheet Page. This feature allows them to submit their weekly working hours by filling out input fields for each day. A submission button records these hours into the timesheet database, making it easy for employees to update their time records accurately.
* **View Timesheet Page**: Employees can also view their submitted timesheets on the View Timesheet Page. This page displays the recorded hours in a tabular format, providing a clear overview of their work hours. The data is retrieved from the database, ensuring that employees see the most current and accurate information.

#### **Number of Leave Days**

**Functionality**: Tracks leave days based on incomplete timesheet updates.

**Features**:

* + Automatically calculates leave days if an employee fails to update their timesheet for any weekday within a week.
  + Updates the leave records in the Employee Detail database.
* **Logout Feature**: To maintain security, the module includes a Logout feature that allows employees to securely end their session. This feature protects user data by ensuring that sessions are properly terminated.

**Lead Module**

**Overview:**

The Lead Module is designed to facilitate team leads in managing and overseeing

their team's timesheets. After logging in, leads can access the Employee Detail Page,

which provides comprehensive information about all employees under their supervision.

This functionality enables leads to monitor their team's performance and track work hours

effectively. The Timesheet Approval Page is a crucial feature of this module, allowing leads to

review submitted timesheets and make decisions on approval or rejection. All actions are

recorded in the timesheet database, ensuring an accurate and auditable record of

timesheet management**.**

T**his module manages operations for leads.**

* **Login Page:** Authenticates the lead. Leads log into the system using

their credentials(userid,password) gaining access to features specific to their role.

* **Employee Detail Page:** Displays details of all employees. This page provides

leads with detailed information about all employees under their supervision.

It includes comprehensive profiles and timesheet details for effective

monitoring and management of their team.

* **Timesheet Approval Page:** The Timesheet Approval Page is a key component of this

module. Leads can review submitted timesheets, and decide to approve or reject them.

Each decision is recorded in the timesheet database, ensuring an accurate record

of all timesheet actions. Approved timesheets are marked accordingly, while rejected

ones are also noted with a rejection status.

* **Logout Feature:** To maintain security, the module includes a Logout feature that

allows employees to securely end their session. This feature protects user data

by ensuring that sessions are properly terminated.

**Manager Module**

**Overview:**

The Manager Module provides a higher level of oversight, offering functionalities similar

to those available to leads but with a broader scope. Managers log in to view detailed

information about all employees across the organization through the Employee Detail Page.

This comprehensive view supports effective management and decision-making. The

Timesheet Approval Page in this module allows managers to handle the approval or

rejection of timesheets at an organizational level. Like the lead module, all decisions are

recorded in the timesheet database, maintaining a transparent and accurate record of

timesheet approvals and rejections**.**

**This module manages operations for managers.**

* **Login Page:** Managers access the system through a secure login, which grants them

access to advanced management features.

* **Employee Detail Page:** Displays details of all employees. This page provides managers

with a comprehensive view of all employees within the organization, including

detailed timesheet information. It allows managers to monitor and evaluate the

performance and work hours of all employees at a higher level.

* **Timesheet Approval Page:** Shows submitted timesheets with options to approve

or reject. Approved or rejected timesheets are stored in the timesheet database.

Managers can handle timesheet approvals and rejections similar to leads.

They can review all submitted timesheets, make decisions on their approval or

rejection, and record these decisions in the timesheet database.

### **Project Assigning Page:**The Project Assigning Page allows managers assign projects to employees and set deadlines. This functionality ensures that employees are properly allocated to various projects with clear deadlines, which facilitates effective project management and tracking.

* **Logout Feature:** Logs out the manager from the system. To maintain security,

the module includes a Logout feature that allows employees to securely end their

session. This feature protects user data by ensuring that sessions are properly

terminated.

**HR(Human Resource) Module**

**Overview**

The HR module in the timesheet management system is designed to provide

human resources personnel with comprehensive tools to manage employee data, review

timesheets, and calculate salaries efficiently. This module serves as a centralized hub for

HR-related activities, ensuring that all employee records and timesheet data are accurate

and up-to-date.

**This module manages operations for managers.**

* **Login Page:** HR access the system through a secure login, which grants them

access to advanced management features.

* **Employee Detail Page:** Displays details of all employees. This page provides HR

with a comprehensive view of all employees within the organization, including

detailed timesheet information. It allows managers to monitor and evaluate the

performance and work hours of all employees at a higher level.

* **Employee Timesheet Page:** Displays details of all employees timesheet.

This page provides HR with a comprehensive view of all employees within the organization, including detailed timesheet information. It allows HR to monitor and evaluate the

performance and work hours of all employees at a higher level.

* **Salary Calculation Page:** HR can also use a salary calculation page, featuring a

dropdown menu for employee IDs, to compute salaries based on total working hours

and leave days. The results, including employee salary, total working hours, and

leave days, are displayed after calculations.

**System Architecture**

### **Overview**

The Timesheet Project is structured into four main modules: Employee, Lead, and Manager,HR. Each module has its distinct functionalities, and they all interact with the backend services to store and retrieve data from the database.

The architecture includes both the frontend and backend components.

**Architecture:**

**Frontend (React.js)**

(React.js)

**Employee Module**

* Login Page
* Update Timesheet
* View Timesheet
* Logout Feature

**Lead Module**

* Login Page
* Employee Detail
* Timesheet Approval

**HR Module**

* Login Page
* Employee Detail
* Timesheet Detail
* Salary Calculation

**Manager Module**

* Login Page Header
* Employee Detail
* Timesheet Approval
* Project Assigning

**Backend (Djangorestframework)**

T**imesheet Service**

* /api/timesheet/ update
* /api/timesheet/view details/
* /api/timesheet/approval
* /api/timesheet/calculate-salary/

**Auth Service**

* /api/login/
* /api/timesheet/
* /api/logout/

**Employee Detail Service**

* api/employee detail

**Database (db.sqlite3)**

**Timesheet Details**

* ID
* Employee id
* Date
* Week(Mon-Sat)
* Total hours Worked
* Status
* No.of leave days
* salary

**User table**

* ID
* Email
* Password
* Phone number
* Role
* role

**Employee details**

* ID
* Name
* Position
* Comments

**Software Requirements**

The Timesheet Project is built using a combination of server-side and client-side technologies,

ensuring a robust and user-friendly application. Below are the detailed software requirements

for the server, client, development environment, dependencies, styling, and database.

**Python**

* **Version**: 3.6
* **Purpose**: Version 3.6 or later. Python is required as the primary language for

backend development, providing support for Django and various libraries used

in the project.Core programming language used for backend development.

Python is known for its simplicity and readability, making it ideal for

developing and maintaining complex applications.

**Django**

* **Version**: 3.0
* **Purpose**: High-level Python web framework used for building the backend of the

application. Django serves as the framework for backend development,

handling API endpoints, database management, and authentication.

providing built-in features like ORM, and an admin interface.

**Node.js**

* **Version**: 12
* **Purpose**: Node.js is necessary for running the JavaScript runtime environment,

which is essential for building and running the React.js frontend. Node.js enables

server-side scripting, allowing the creation of dynamic web page content before the

page is sent to the user's web browser.

**npm (Node Package Manager)**

* **Version**: 6
* **Purpose**: Package manager for JavaScript, used to install, share, and distribute code

modules for the frontend. npm handles the management of project dependencies,

ensuring that the correct versions of libraries and tools are used.

**Client Requirements:**

For optimal user experience and development, the client environment should include:

**Operating System:** Windows 10.

This OS is recommended for its compatibility with development tools and its ability to

run the necessary software effectively.

**Web Browser:**Latest version of Google Chrome.

Chrome is chosen for its robust support of modern web technologies and its performance

in handling complex web applications.

**Development Environment**

**Code Editor**

* **Visual Studio Code**
* **Purpose**: An open-source, cross-platform code editor that provides robust support

for various programming languages and frameworks. It offers features

like syntax highlighting, code completion, debugging, and integrated Git support.

**Version Control System**

* **Git**
* **Purpose**: A distributed version control system used to track changes in the source

code during development. Git facilitates collaboration among developers, allowing

them to work on different parts of the project simultaneously and merge their

changes seamlessly.

**Dependencies:**

**Backend Dependencies:**

**Django REST Framework**

* **Purpose**: A powerful and flexible toolkit for building Web APIs in Django.

It simplifies the creation of RESTful APIs, making it easier to handle HTTP

requests and responses in a standardized way.

Features:\* Provides tools for creating and managing RESTful APIs, including

serialization, authentication, and view sets.

**Django's Built-in Authentication System**

* **Purpose**: Provides user authentication, including login, logout, password

management, and user session handling. This system ensures secure access to the

application and protects user data.

Features:Securely manages user sessions, password hashing, and user role

assignments.

**Frontend Dependencies**

* **React.js**
  + **Purpose**: A JavaScript library for building user interfaces. React allows

developers to create reusable UI components, manage application state, and

efficiently update and render components in response to data changes.

* **Axios**
  + **Purpose**: A promise-based HTTP client for making API requests. Axios

simplifies the process of sending asynchronous HTTP requests to the

backend and handling responses, allowing for easy integration with RESTful

APIs.

* **React Router**
  + **Purpose**: A standard library for routing in React applications.

It enables navigation between different views or components, allowing the

application to handle multiple routes and dynamically change the content

displayed to the user.

**Styling**

**CSS**

**Purpose**: Cascading Style Sheets (CSS) are used to style the application and create

a visually appealing user interface. The primary, secondary, tertiary, and additional

colors are defined to maintain a consistent color scheme throughout the application.

The application uses CSS for styling, with the following color scheme:

Primary Color:#0F1035

– A deep, dark blue used for major components and branding.

Secondary Color:#365486

– A slightly lighter blue used for accents and secondary elements.

Tertiary Color: #7FC7D9

– A teal blue used for highlights and additional UI elements.

Additional Color:#DCF2F1

– A light gray used for backgrounds and subtle UI elements.

These colors are chosen to create a cohesive and visually appealing design, ensuring that

the user interface is both functional and aesthetically pleasing.

**Database**

* **Relational Database**
  + **Purpose**: A relational database is used to store user information,

timesheets, and approval status. Relational databases are structured and use

tables to store data in rows and columns, enabling complex queries and

ensuring data integrity.

* **SQLite**
  + **File**: db.sqlite3
  + **Purpose**: SQLite is a lightweight, self-contained relational database that is

easy to set up and use. It is used during development for simplicity and

portability, storing all data in a single file (db.sqlite3). This allows

for quick development iterations and testing without the need for a separate

database server.

Features:Lightweight, file-based database that requires minimal setup,

making it suitable for development and testing.

**Database Schema:**

* **Users Table:**Stores user credentials and roles (Employee, Lead, Manager).
* **Employee Database**
* **Fields**:
  + id: Unique identifier for each employee.
  + name: Employee's name.
  + empid: Employee's ID.
  + email: Employee's email address.
  + username: Username for login.
  + password: Password for login.
  + personstatus: Role of the employee (employee, lead, manager).

### **Timesheet Database**

**Fields**:

* + id: Unique identifier for each timesheet entry.
  + employee\_id: Identifier for the employee.
  + date: Date of the timesheet entry.
  + hours\_worked: Number of hours worked.
  + status: Status of the timesheet (Pending, Approved, Rejected).
  + Salary:employee salary
  + lead\_approval: Lead's approval status.
  + manager\_approval: Manager's approval status.
* **Approvals Table:**

Tracks approval or rejection status for submitted timesheets, including

lead or manager ID, and decision timestamps.

**User interface Requirements**

**General Requirements**

1. **Responsiveness:**

Objective:Ensure that the Timesheet Project’s user interface (UI) is adaptable and

functional across a range of devices, including desktops, tablets, and smartphones.

**Flexible Layouts**

Objective: Design the application layout to adjust smoothly to different screen sizes,

including desktops, tablets, and smartphones.

Implementation: Use CSS Flexbox or Grid layouts to create flexible and

adaptive page structures. Ensure that elements rearrange and resize gracefully as the

viewport dimensions change

**Media Queries**

* **Objective**: Apply media queries in the CSS to handle different device

resolutions and orientations.

* **Implementation**: Define breakpoints for common device widths (e.g., 320px, 480px,

768px, 1024px, and 1200px). Adjust the layout, font sizes, and other UI elements

at each breakpoint to maintain usability and aesthetics.

Example:

css

@media (max-width: 768px) {

.container {

flex-direction: column;

}

.header {

font-size: 1.5rem;

}

}

**Touch-Friendly Elements:**

Button Sizes: Ensure buttons and interactive elements are large enough for easy

tapping on touch screens. Recommended minimum size is 44x44 pixels.

Spacing:Provide adequate spacing around clickable elements to prevent accidental selections.

**2.User Experience (UX):**

* + Design an intuitive and user-friendly interface with clear navigation

and consistent design elements.

**Employee Module**

**1.Login Page**:

**Objective:**Provide a secure and user-friendly login interface for employees to

access the Timesheet Project application.

**Fields:**

- Username:

- Type: Text

- Placeholder: "Enter your username"

- Validation: Required field, must be a valid username registered in the system.

- Password:

- Type: Password

- Placeholder: "Enter your password"

- Validation: Required field, must be at least 8 characters long.

**Features**: Error messages for invalid login attempts, password visibility toggle

**Buttons:**

- Login:

- Action: Authenticates the user based on the provided username and password.

- Functionality: Submits the login form for verification against stored

credentials.

- Design: Prominently displayed with a contrasting color to ensure visibility.

**Features:**

- Error Messages:

- Displayed for invalid login attempts (e.g., incorrect username/password).

- Messages should be clear and guide users on what went wrong

(e.g., "Incorrect username or password").

- Password Visibility Toggle:

- Allows users to show or hide their password as they type.

- Toggle icon (eye symbol) positioned within the password field.

**2.Update Timesheet Page**:

**Objective**:Enable employees to record their weekly work hours accurately.

**Fields:**

- Employee ID:

- Type: Text (Auto-filled from the user’s profile)

- Placeholder: "Employee ID"

- Validation: Automatically populated, should match the logged-in user’s ID.

- Week:

- Type: Date/Week Picker

- Placeholder: "Select the week"

- Validation: Required field, must be a valid week within the current year.

- Monday to Saturday:

- Type: Number

- Placeholder: "Enter hours worked"

- Validation: Must be a non-negative number. Total hours should be accurate.

-Total Hours:

- Type: Calculated field

- Placeholder: "Total hours worked"

- Validation: Calculated automatically from the sum of hours entered for each day.

**Buttons:**

- Submit:

- Action: Submits the timesheet data to the backend.

- Functionality: Performs final validation and sends the data for processing.

- Design: Prominent and distinguished button color.

- Cancel:

- Action: Discards changes and redirects to the previous page or dashboard.

- Functionality: Resets the form or redirects the user without saving data.

- Design: Styled to be less prominent than the submit button.

**Features:**

- Validation:

- Enforces valid data entry for hours and dates (e.g., no negative numbers,

correct format).

**- Confirmation Popup:**

- Displays a confirmation popup upon successful submission to inform the user

that their timesheet has been submitted.

- Contains a "Back to dashboard" button to dismiss the popup.

3.**View Timesheet Page**:

**Objective:**Allow employees to view and review their submitted timesheets.

**Table Columns:**

- Employee ID:

- Display: Shows the unique identifier for the employee.

- Data: Populated from the employee's profile.

- Hours Worked:

- Display: Total hours worked per week.

- Data: Retrieved from the timesheet database.

- Status:

- Display: Indicates the approval status of the submitted timesheet

(e.g., "Approved," "Pending," "Rejected").

- Data: Fetched from the database based on the latest approval status.

**Buttons:**

- Back to Dashboard:

- Action: Redirects the user back to the main dashboard.

- Functionality: Allows the user to navigate away from the current page.

- Design: Clearly labeled for easy access.

#### **4. Number of Leave Days**

* **Functionality**: Tracks leave days based on incomplete timesheet updates.
* **Features**:
  + Automatically calculates leave days if an employee fails to update their timesheet for any weekday within a week.
  + Updates the leave records in the Employee Detail database.

**5.Logout:**

**Objective:**Provide a secure way for employees to end their session.

**Button:**

-Logout

- Action: Ends the user’s session and redirects them to the login page or a public area.

- Functionality: Logs out the user securely, clearing session data.

- Design: Clearly visible and easily accessible.

**Lead Module**

1. **Login Page**:

**Objective:**To provide a secure and familiar login interface for leads, ensuring they

can access their accounts seamlessly.

**Fields:**

- UserID:

- Type:Text

- Placeholder:"Enter your ID"

- Validation: Required field. Must be a valid userid registered in the system.

- Password:

- Type:Password

- Placeholder:"Enter your password"

- Validation:Required field. Must adhere to security requirements, such as a minimum

length of 8 characters.

**Buttons:**

- Login:

- Action:Authenticates the user based on the entered credentials.

- Functionality:Submits the login form and verifies user credentials against the

database.

- Design:Clearly distinguished with a prominent color to ensure high visibility.

**Features:**

- Error Messages:

- Description:Provides feedback for incorrect login attempts, such as invalid

username or password.

- Example Message:”The username or password you entered is incorrect."

- Password Visibility Toggle:

- Description: Allows users to toggle password visibility for ease of entry.

- Design:Icon (usually an eye symbol) next to the password field.

1. **Employee Detail Page**:

**Objective:** To provide leads with comprehensive information about all employees to

facilitate efficient management and approval processes.

**Table Columns:**

- Employee ID:

- Description:Displays the unique identifier for each employee.

- Data Source:Pulled from the employee's profile information in the database.

- Position:

- Description:Indicates the job title or role of the employee.

- Data Source:Sourced from employee records.

- Email:

- Description:Displays the employee's email address.

- Data Source:Extracted from the employee’s profile information.

- Lead Approval:

- Description:Shows the approval status of the employee’s timesheets or specific

actions.

- Data Source:Fetched from the timesheet records or approval logs.

**Buttons:**

- Back to Dashboard:

- Action:Redirects the user back to the main dashboard or previous page.

- Functionality:Provides an easy way to navigate away from the current page.

- Design:Clearly labeled for intuitive navigation.

* + **Table Columns**: Employee ID, Name, Position, Email, lead approval
  + **Buttons**: Back to dashboard

1. **Timesheet Approval Page**:

**Objective:**The Timesheet Approval Page is designed to enable leads to review and

manage timesheet submissions made by employees. This page allows leads to

approve or reject timesheets, ensuring accurate tracking and compliance with

company policies.

**Table Columns**

1. Employee ID

- Description:Displays the unique identifier of the employee who submitted the

timesheet.

- Data Source:Pulled from the employee's profile or the timesheet database.

- Display:Textual representation in a tabular format.

2.Week

- Description:Shows the week for which the timesheet is being reviewed.

- Data Source:Sourced from the timesheet data.

- Display:Week number or date range (e.g., "Week of August 1 - August 7").

Monday to Saturday

- Description:Displays the hours worked each day from Monday to Saturday.

- Data Source:Detailed in the timesheet entry.

- Display:Numeric values representing hours worked each day.

Hours Worked

- Description:Total number of hours worked during the week.

- Data Source:Calculated from the daily hours entries.

- Display:Numeric total.

Status (Approve/Reject)

- Description:Current status of the timesheet, indicating whether it has been approved

or rejected.

- Data Source:Updated based on lead’s actions.

- Display:Status indicator with options to change the status.

- Visual Feedback:The status column updates to show "Approved" or "Rejected" based

on the action taken.

**Features**: Confirmation dialogs for approval/rejection, status update of

timesheets

**Manager Module**

**Objective:**To provide a secure and familiar login interface for leads, ensuring they

can access their accounts seamlessly.

**Fields:**

- UserID:

- Type:Text

- Placeholder:"Enter your ID"

- Validation: Required field. Must be a valid userid registered in the system.

- Password:

- Type:Password

- Placeholder:"Enter your password"

- Validation:Required field. Must adhere to security requirements, such as a minimum

length of 8 characters.

**Buttons:**

- Login:

- Action:Authenticates the user based on the entered credentials.

- Functionality:Submits the login form and verifies user credentials against the

database.

- Design:Clearly distinguished with a prominent color to ensure high visibility.

**Features:**

- Error Messages:

- Description:Provides feedback for incorrect login attempts, such as invalid

username or password.

- Example Message:”The username or password you entered is incorrect."

- Password Visibility Toggle:

- Description: Allows users to toggle password visibility for ease of entry.

- Design:Icon (usually an eye symbol) next to the password field.

**1.Employee Detail Page**:

**Objective:** To provide leads with comprehensive information about all employees to

facilitate efficient management and approval processes.

**Table Columns:**

- Employee ID:

- Description:Displays the unique identifier for each employee.

- Data Source:Pulled from the employee's profile information in the database.

- Position:

- Description:Indicates the job title or role of the employee.

- Data Source:Sourced from employee records.

- Email:

- Description:Displays the employee's email address.

- Data Source:Extracted from the employee’s profile information.

- Lead Approval:

- Description:Shows the approval status of the employee’s timesheets or specific

actions.

- Data Source:Fetched from the timesheet records or approval logs.

**Buttons:**

- Back to Dashboard:

- Action:Redirects the user back to the main dashboard or previous page.

- Functionality:Provides an easy way to navigate away from the current page.

- Design:Clearly labeled for intuitive navigation.

* + **Table Columns**: Employee ID, Name, Position, Email, lead approval
  + **Buttons**: Back to dashboard

**2.Timesheet Approval Page**:

**Objective:**The Timesheet Approval Page is designed to enable leads to review and

manage timesheet submissions made by employees. This page allows leads to

approve or reject timesheets, ensuring accurate tracking and compliance with

company policies.

**Table Columns**

1. Employee ID

- Description:Displays the unique identifier of the employee who submitted the

timesheet.

- Data Source:Pulled from the employee's profile or the timesheet database.

- Display:Textual representation in a tabular format.

2.Week

- Description:Shows the week for which the timesheet is being reviewed.

- Data Source:Sourced from the timesheet data.

- Display:Week number or date range (e.g., "Week of August 1 - August 7").

Monday to Saturday

- Description:Displays the hours worked each day from Monday to Saturday.

- Data Source:Detailed in the timesheet entry.

- Display:Numeric values representing hours worked each day.

Hours Worked

- Description:Total number of hours worked during the week.

- Data Source:Calculated from the daily hours entries.

- Display:Numeric total.

Status (Approve/Reject)

- Description:Current status of the timesheet, indicating whether it has been approved

or rejected.

- Data Source:Updated based on lead’s actions.

- Display:Status indicator with options to change the status.

- Visual Feedback:The status column updates to show "Approved" or "Rejected"

based on the action taken.

**Features**: Confirmation dialogs for approval/rejection, status update of

Timesheets

### **3.Project Assigning Page**

#### 

#### **Functionality**

The Project Assigning Page allows managers to assign projects to employees and set deadlines. This functionality ensures that employees are properly allocated to various projects with clear deadlines, which facilitates effective project management and tracking.

#### **Features**

1. **Input Fields for Project Information**
   * **Project Name**
     + **Type**: Text field
     + **Description**: A field where managers can enter the name of the project.
   * **Project Deadline**
     + **Type**: Date picker
     + **Description**: A field where managers can select the deadline date for the project.

**4.Logout**:

**Objective:**Provide a secure way for employees to end their session.

Button:

-Logout

- Action: Ends the user’s session and redirects them to the login page or a public area.

- Functionality: Logs out the user securely, clearing session data.

- Design: Clearly visible and easily accessible.

**HR Module**

**Objective:** The HR module in the timesheet management system is designed to provide

human resources personnel with comprehensive tools to manage employee data, review

timesheets, and calculate salaries efficiently**.**

The HR module begins with a secure login page, ensuring that only authorized HR

personnel can access sensitive employee information. Once logged in, HR users have access to

several key features:

* **Employee Detail View**: This feature allows HR to view detailed information about

every employee in the organization, including their department, role, and timesheet

status. It provides a quick and easy way for HR to access and manage employee

records.

* **Employee Timesheet View**: HR can also review timesheet entries submitted by all

employees. This view provides a detailed breakdown of each employee's working hours,

tasks, and projects, enabling HR to monitor work patterns and ensure

compliance with company policies.

* **Salary Calculation**: One of the most critical features of the HR module is the

salary calculation tool. By selecting an employee from a dropdown menu, HR can

calculate their salary based on the total working hours and leave days recorded in the

timesheet database. This tool ensures that salaries are calculated accurately and

efficiently, reflecting each employee's actual work and leave records.

**Implementation Plan**

1. **Setup and Initialization**
   * Set up the project structure with Django for the backend and React.js

for the frontend.

* + Configure the database settings in Django.
  + Initialize a new React project.

1. **Backend Implementation**
   * **Create Models**:
     + Define models for Employee, Timesheet, TimesheetEntry

and User (for authentication).

* + **Create Serializers**:
    - Create serializers for the models to handle data conversion.
  + **Create Views and Endpoints**:
    - Define views and endpoints for each module's functionality

(login, update timesheet, view timesheet, employee details,

timesheet approval/rejection).

* + **Authentication**:
    - Implement user authentication using Django's built-in authentication

system.

**Initialize Project**:

Start a new Django project using django-admin startproject tech\_army.

- Create App:Within the project, create a new Django app for the timesheet

functionality using python manage.py startapp tech\_army\_app.

**- Project Structure:**

Tech\_Army/

├── backend/

│ ├── manage.py

│ ├── tech\_army\_app/

│ │ ├── \_\_init\_\_.py

│ │ ├── admin.py

│ │ ├── apps.py

│ │ ├── models.py

│ │ ├── serializers.py

│ │ ├── urls.py

│ │ └── views.py

│ └── Tech\_Army/

│ ├── \_\_init\_\_.py

│ ├── settings.py

│ ├── urls.py

│ └── wsgi.py

1. **Frontend Implementation**

Initialize Project:Use npx create-react-app frontend to create a new React project.

**- Project Structure:**

frontend/

├── public/

├── src/

│ ├── components/

│ │ ├── Employee/

│ │ ├── Lead/

│ │ ├── Manager/

|------HR

│ ├── App.js

│ ├── index.js

│ └── services/

└── package.json

* + **Create Components**:
    - Employee Module: Login, UpdateTimesheet, ViewTimesheet, Logout
    - Lead Module: Login, EmployeeDetail, TimesheetApproval
    - Manager Module: Login, EmployeeDetail, TimesheetApproval,

Project assigning

* HR Module: Login, EmployeeDetail,Employee Timesheet,Salary

Calculation

* + **Implement Routing**:
    - Set up routing to navigate between different pages (login, update

timesheet,view timesheet, employee detail, timesheet approval).

**Setup React Router:**

- Define Routes:Configure routing in App.js to handle navigation

between components.

* + **Handle State Management**:
    - Use state management to manage application state

and handle data fetching and submission.

Use React’s Context API for managing application state and

handling data fetching and submission.

**4.Database Configuration:**

- Database Settings:Configure SQLite (default in Django) in settings.py.

Update DATABASES settings as needed:

python

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

**5.Styling**

* + Define global styles and component-specific styles.
  + Ensure consistent and user-friendly UI across all pages.

**Data Requirements**

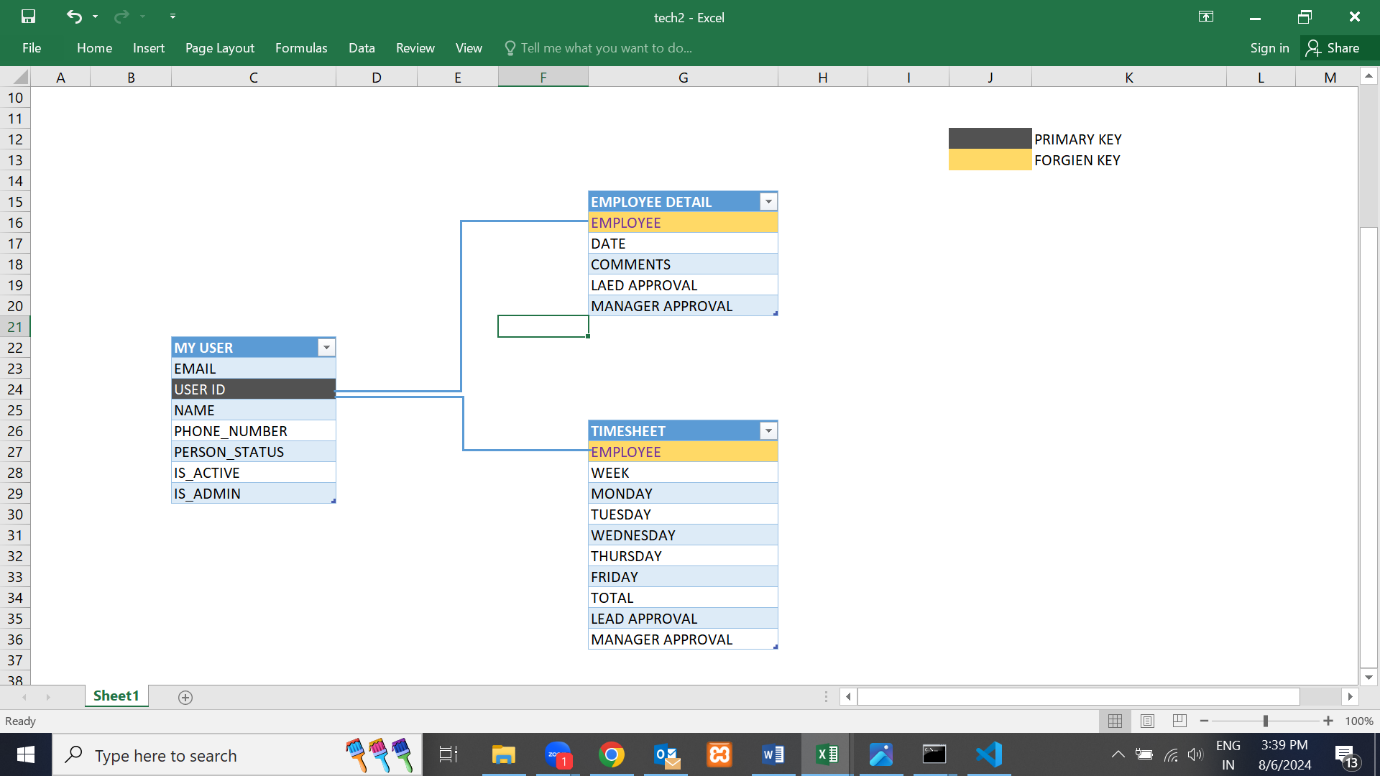
**Overview:**

To implement the Timesheet Project, you need to define the data requirements for

each module, including the necessary fields and data types for the database tables, as well as

the API endpoints for interaction between the frontend and backend. Below is a detailed

breakdown of the data requirements for the project:

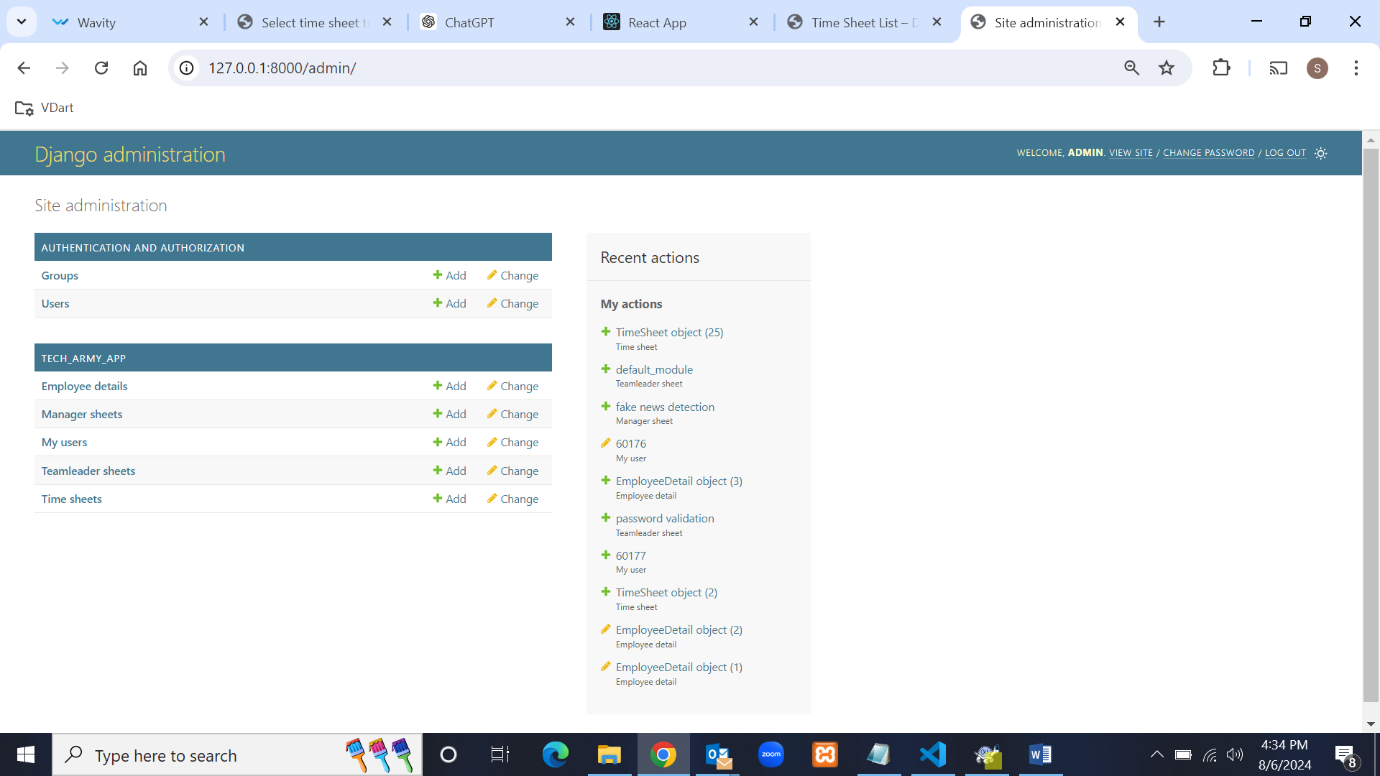
****

**Workflow Process**

**Backend Implementation**:

**1.**The admin panel has four databases: Employee Details, Manager sheets,

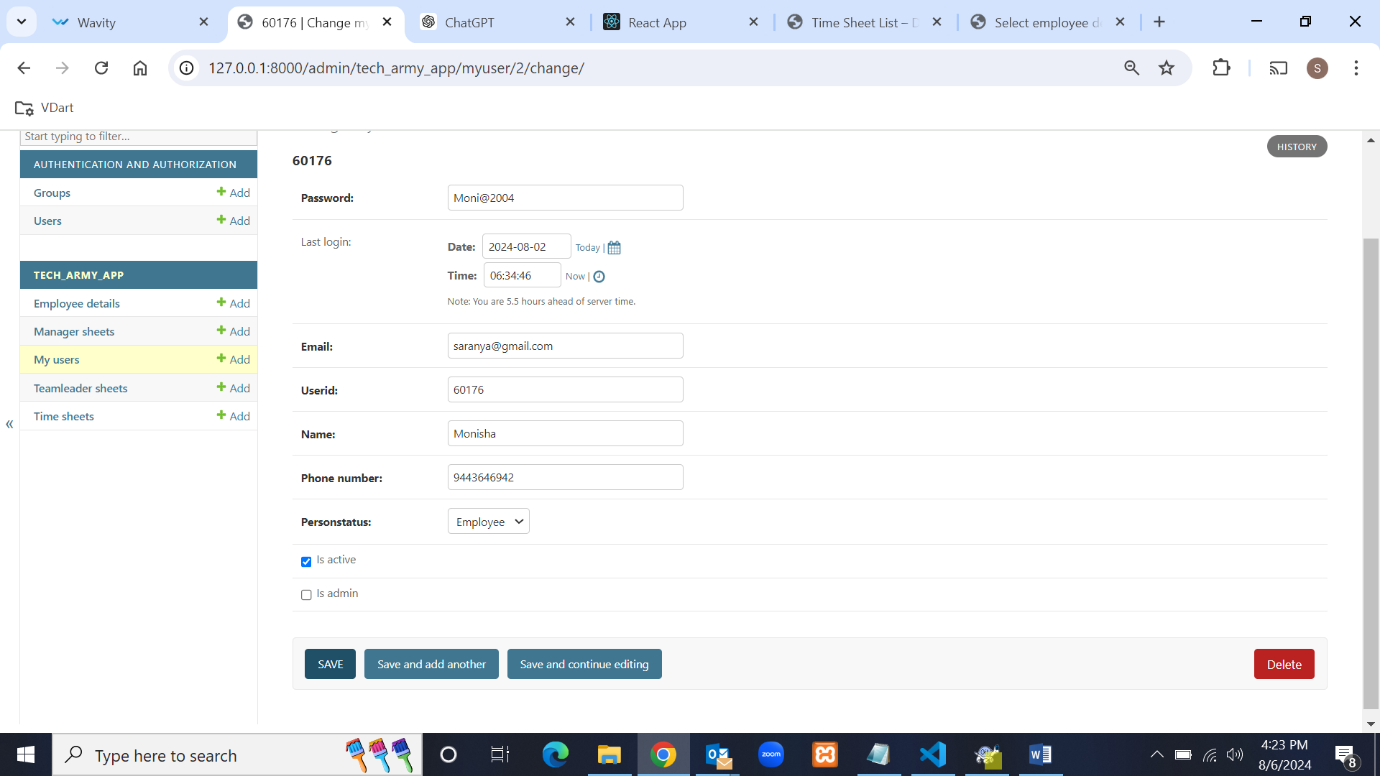
My users,Teamleader sheets, and Timesheets.

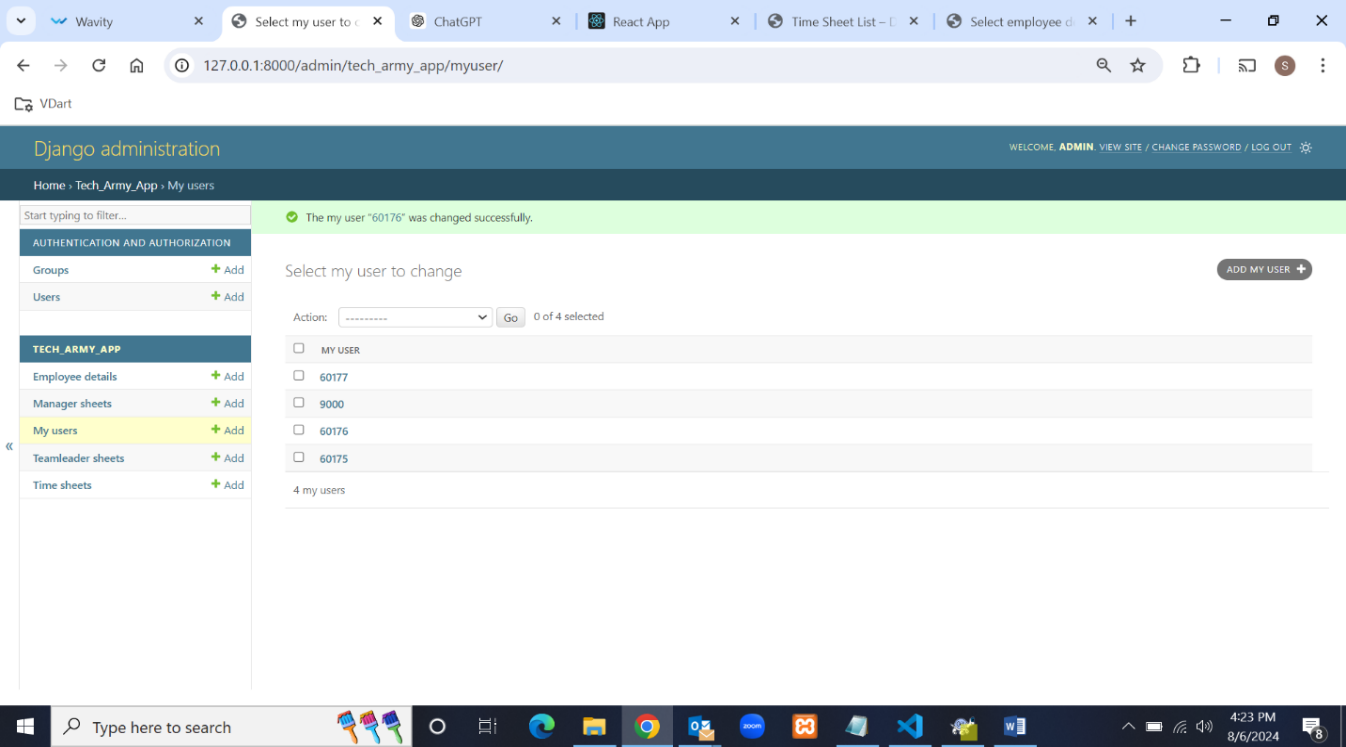


2.The Users database is used for registration and stores basic details such as user ID,

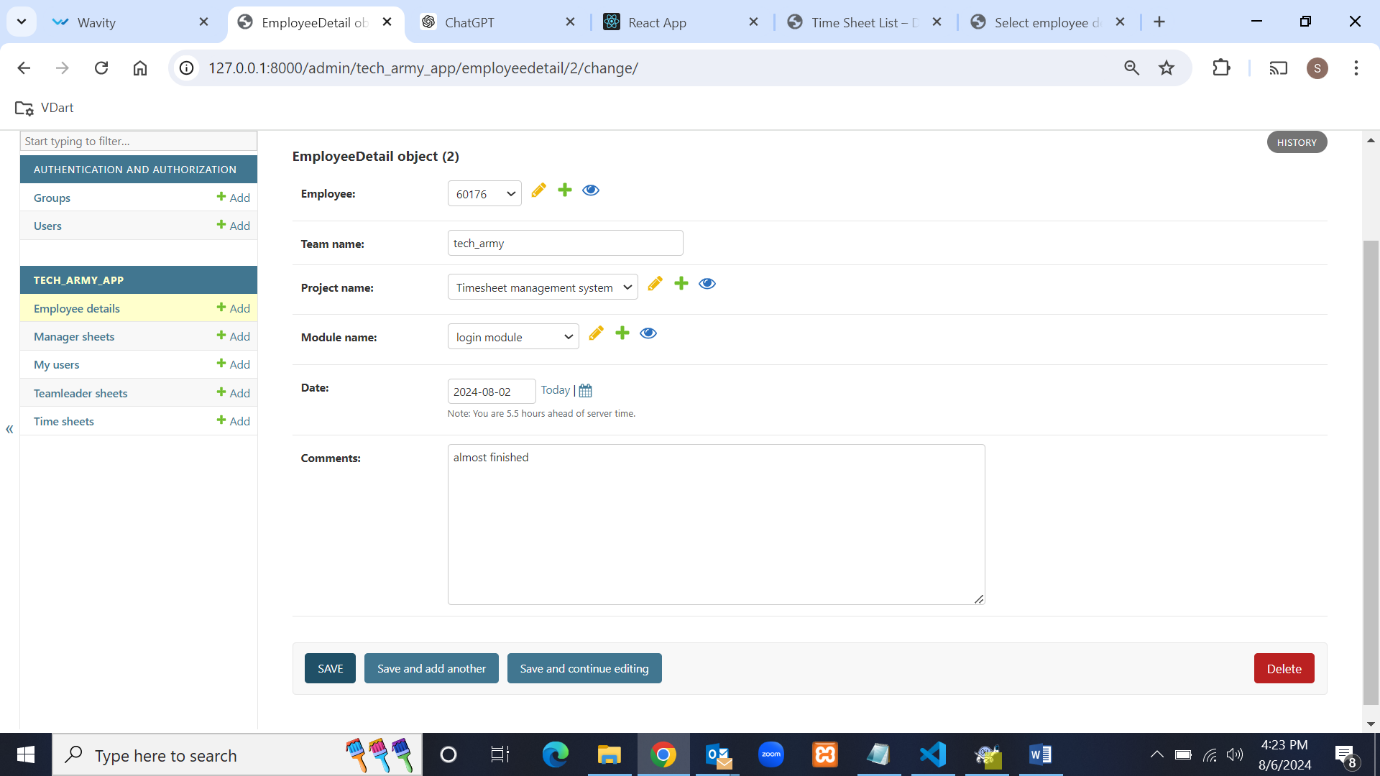
name, password, phone number, email, and the person status (lead, manager, or employee)

of the employee.



3.The Employee Details database is used to store employee information, including employee ID

(which is a foreign key), team name, project name, date, and comments.

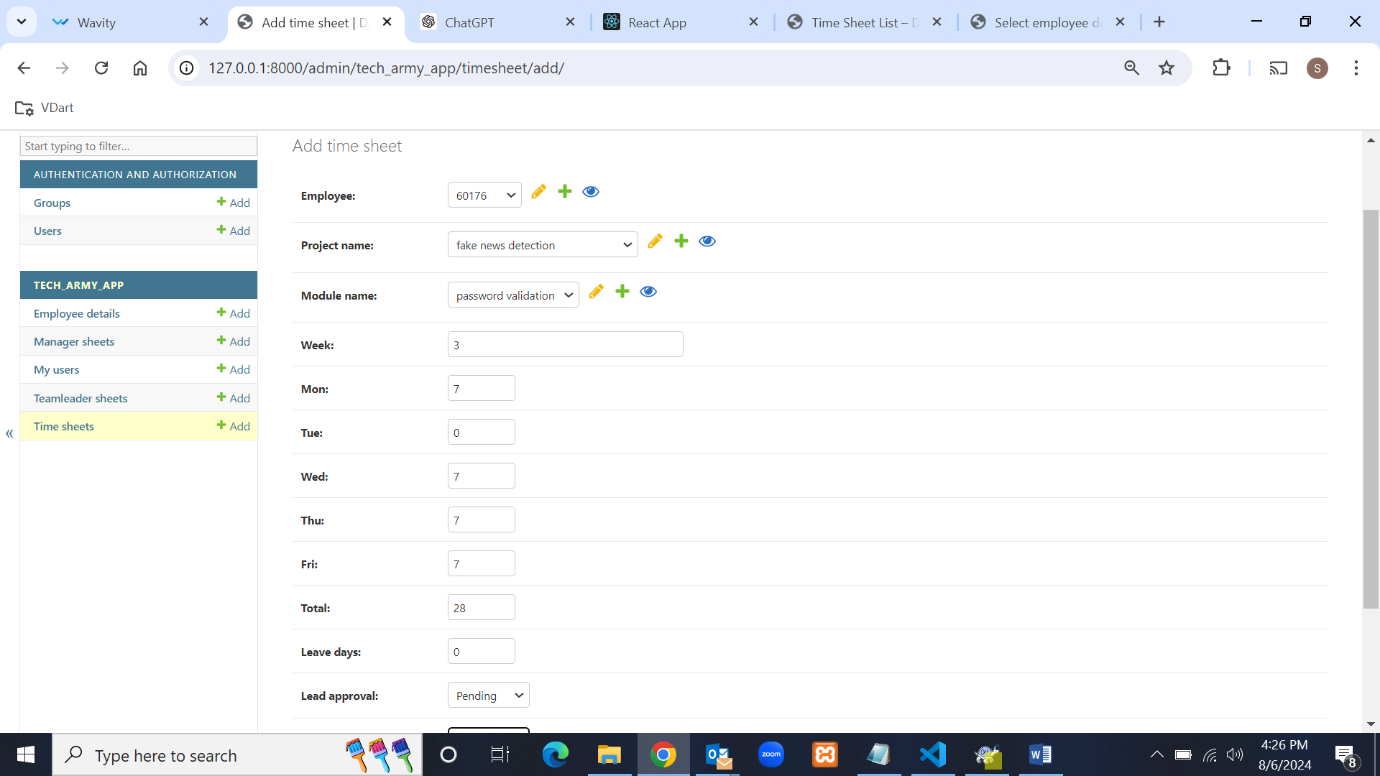


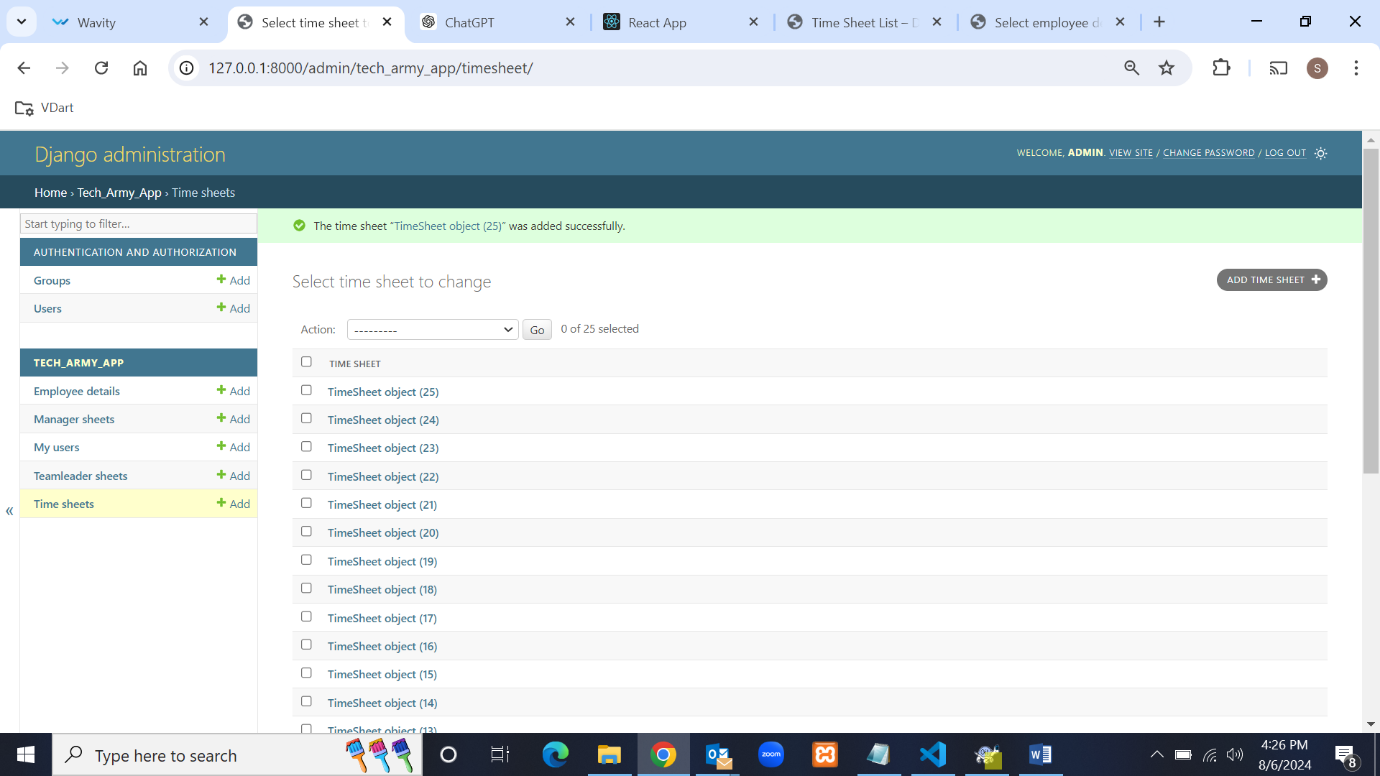
4.The Timesheet database includes the following fields: employee ID (foreign key), project name

foreign key), module name (foreign key), week, Monday, Tuesday, Wednesday, Thursday, Friday

, total hours, leave days (calculated automatically if any weekday is not filled in, it is considered

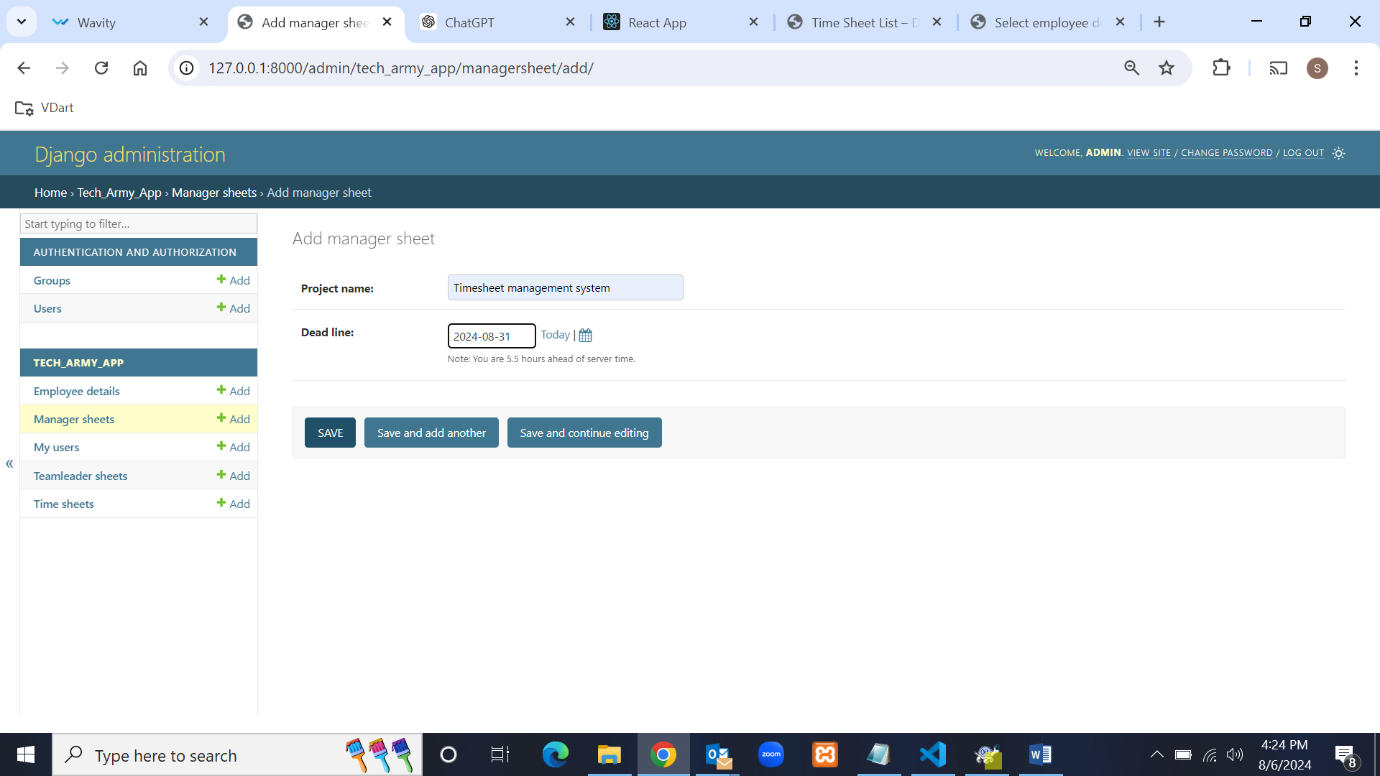
as leave),and lead approval, and manager approval.

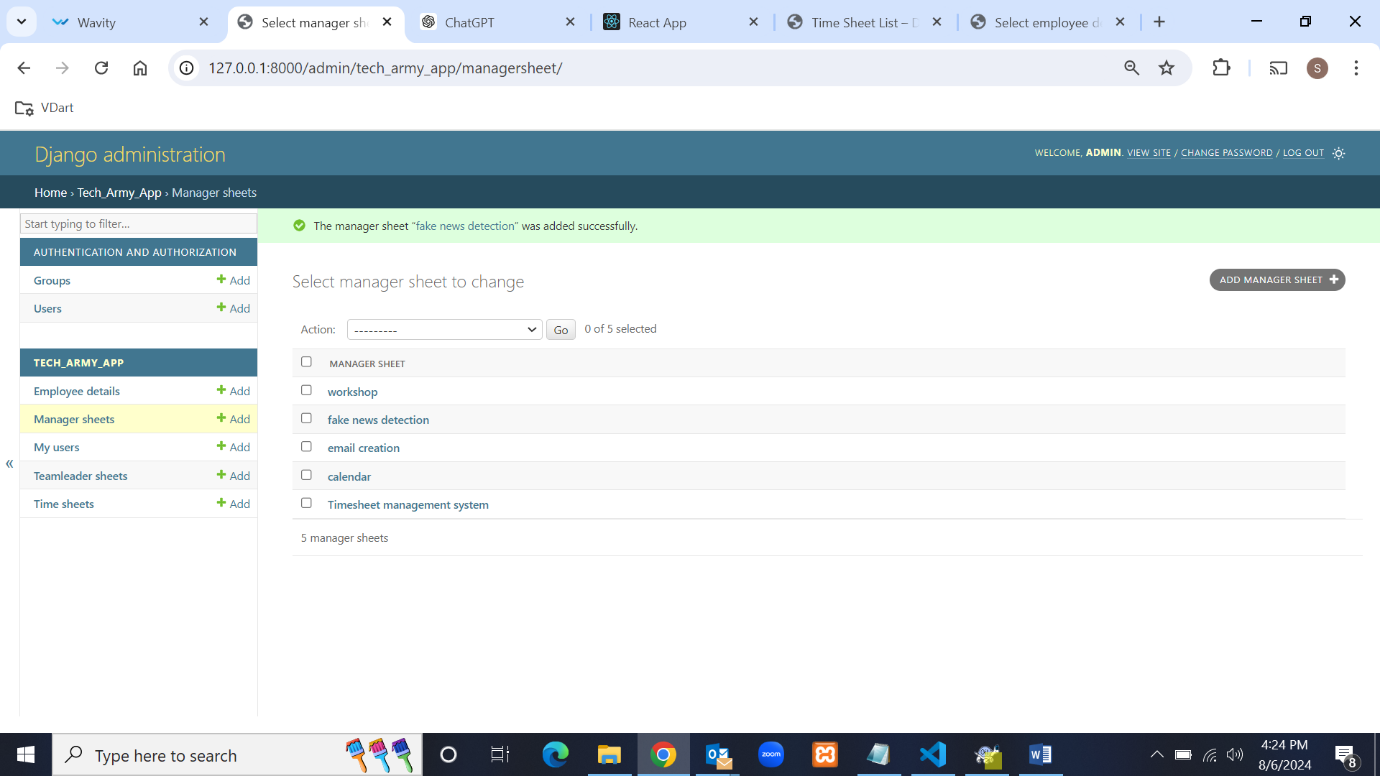




5.The ManagerSheet database has fields for project name and deadline to assign projects to

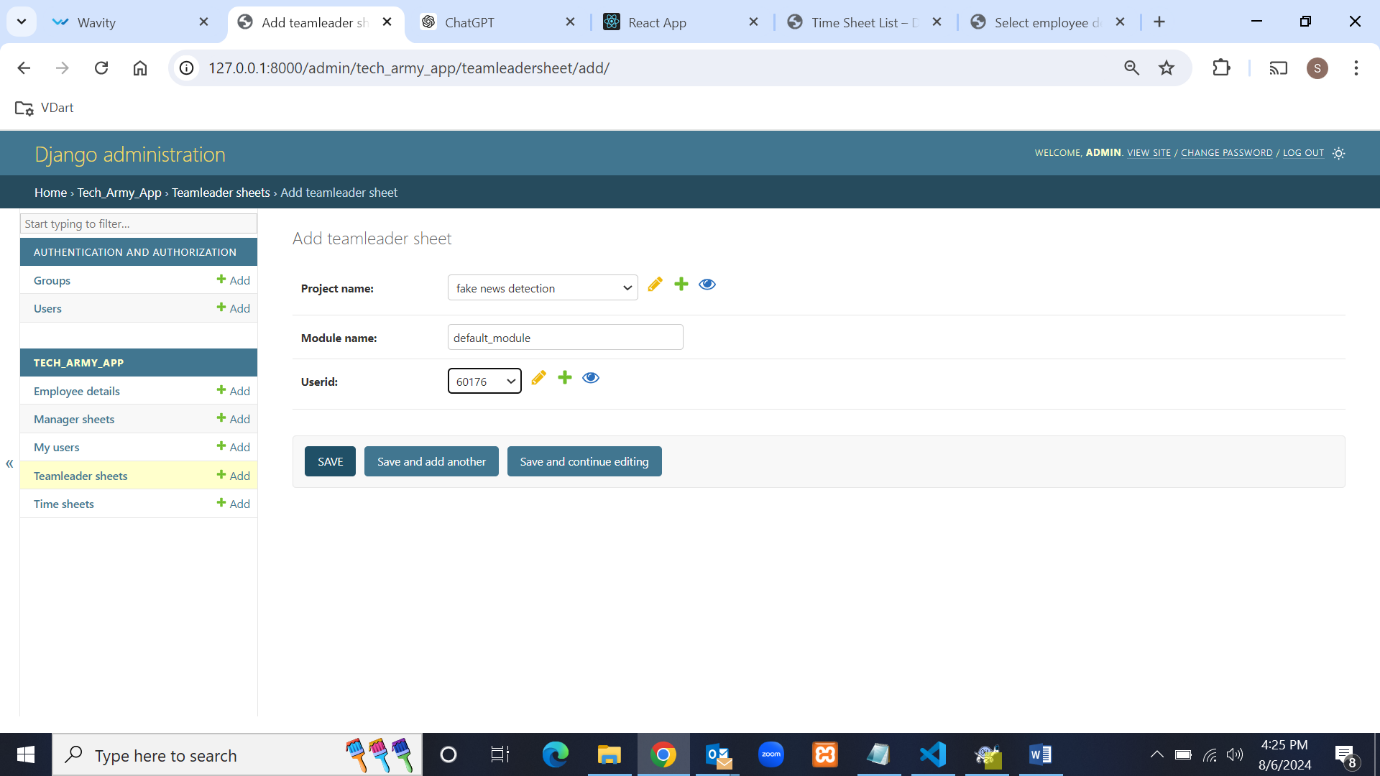
employees.

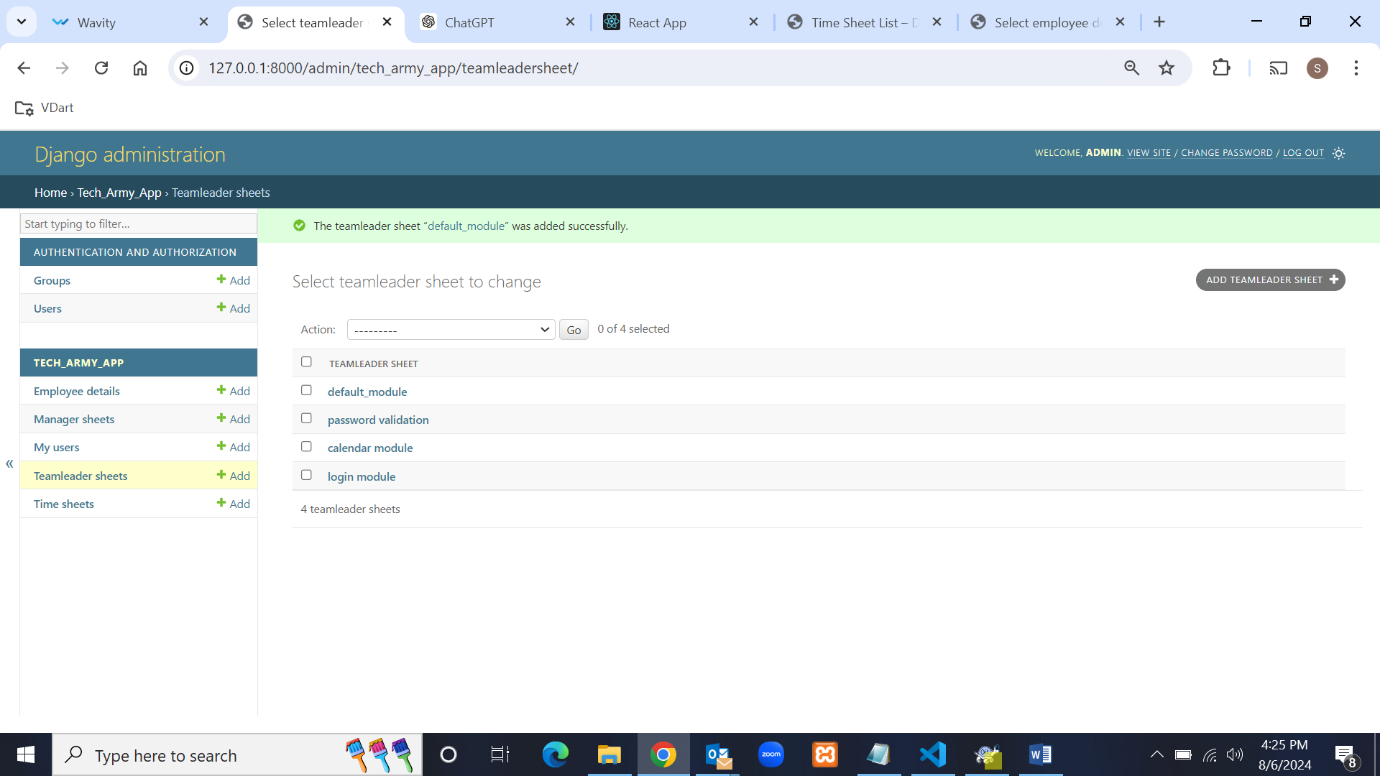




6.The TeamLeader Sheets database has the following fields: project name (foreign key),

module name, and user ID (foreign key) to assign modules to employees.



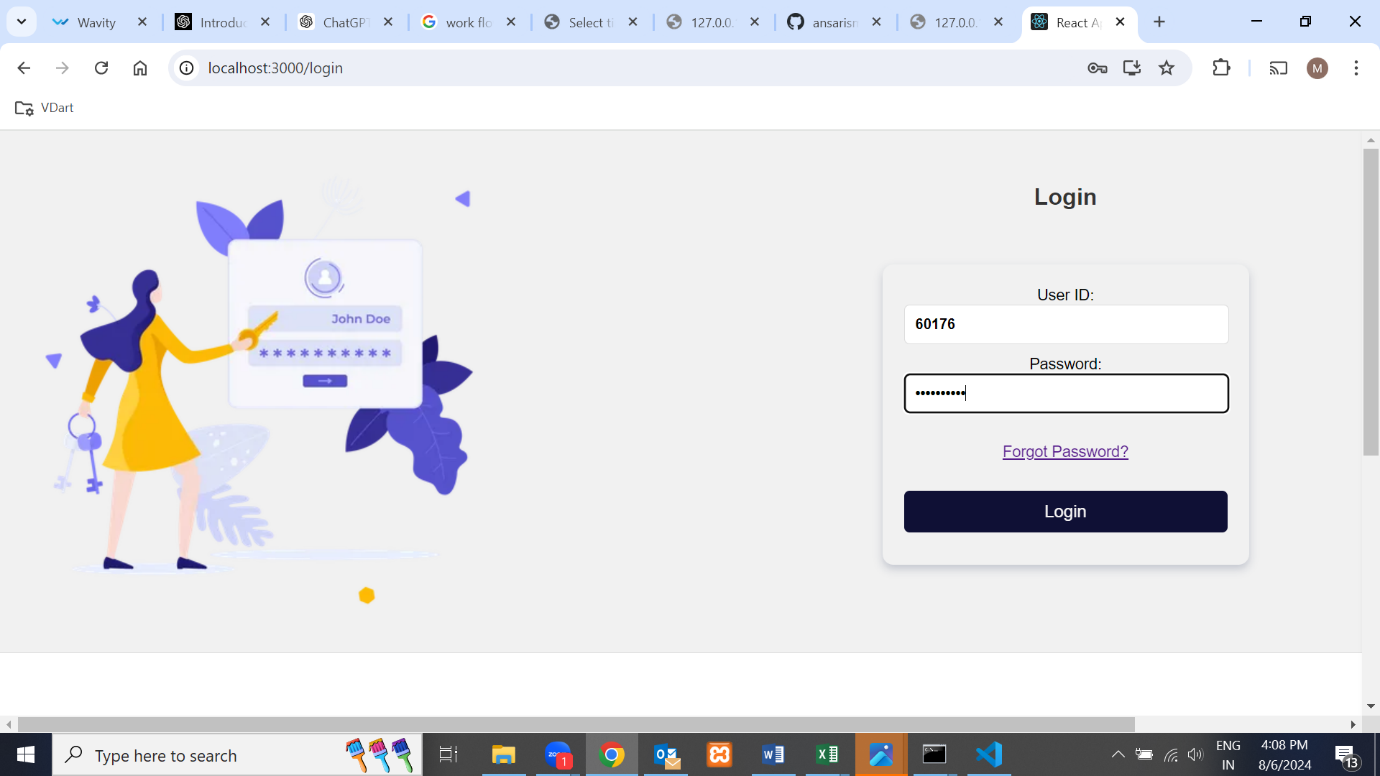


**Frontend Implementation:**

**1.Employee module:**

Log in using the employee ID and password. During login, the system checks the person's status.

If the status is 'employee', it redirects to the employee dashboard.

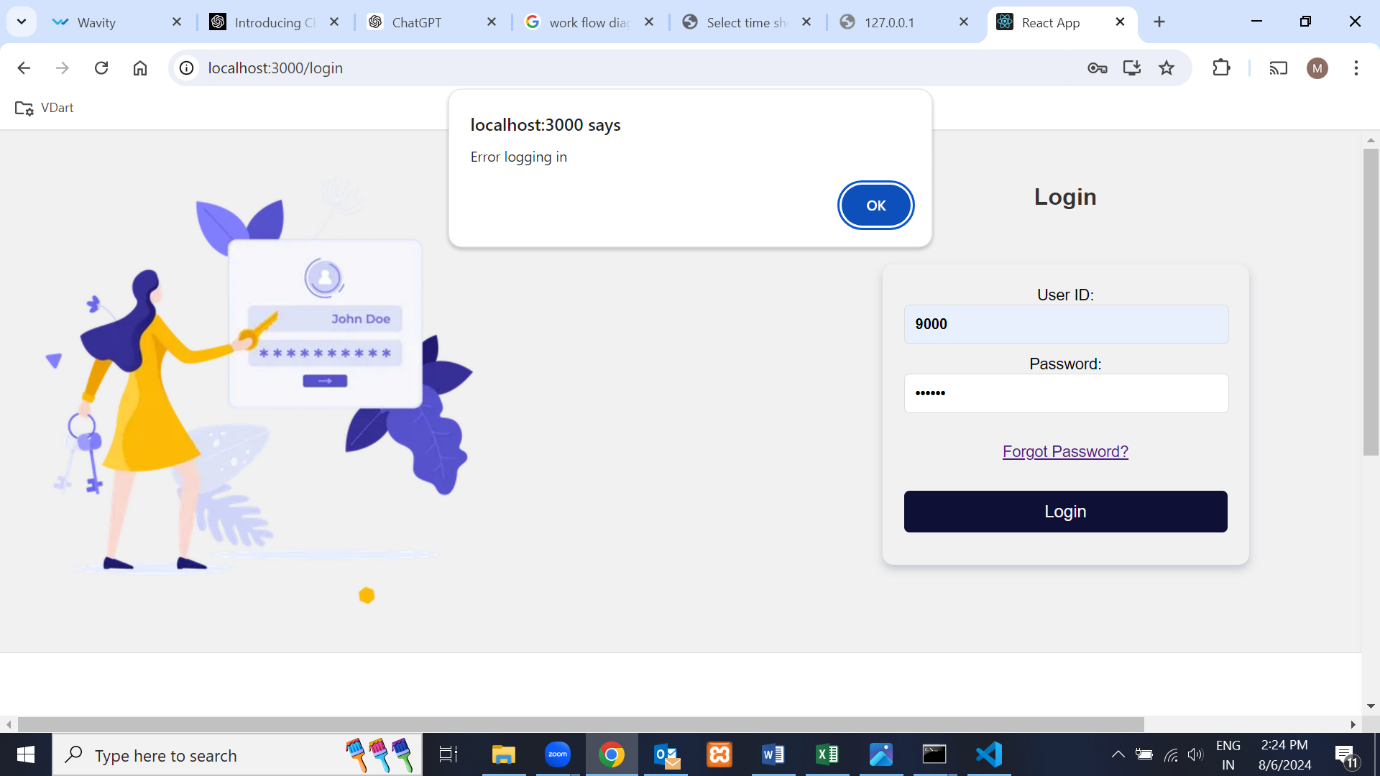
****

If an incorrect employee ID or password is provided during login, an error message will be

displayed.

**Top of Form**

**Bottom of Form**

****

In the employee dashboard, there are four functionalities. The first one is updating the timesheet,

where employees can update their timesheets. After updating, the timesheet information is

stored in the Timesheets database.

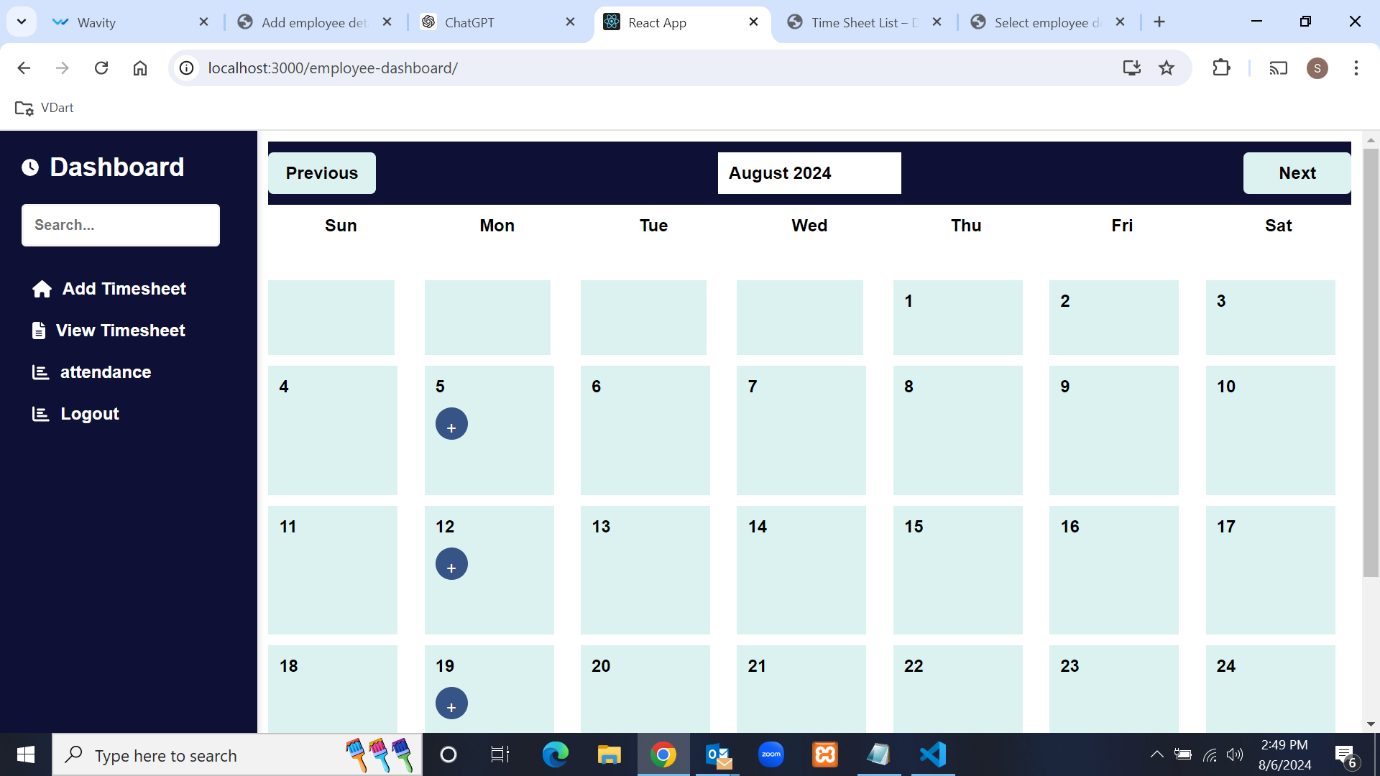
In the update timesheet section, it looks like a calendar, but only the Monday column has a "+"

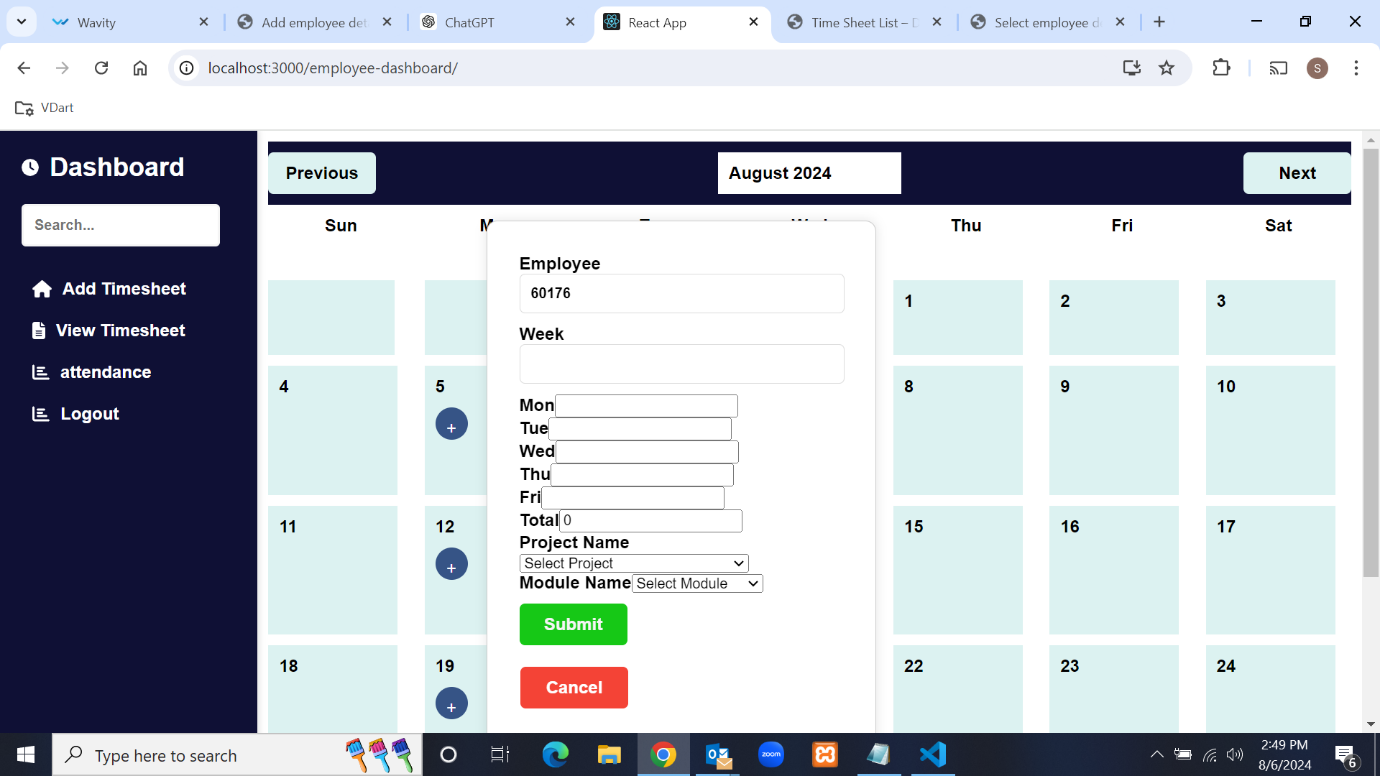
button. If we click the "+" button, a form pops up with fields for employee ID, project name,

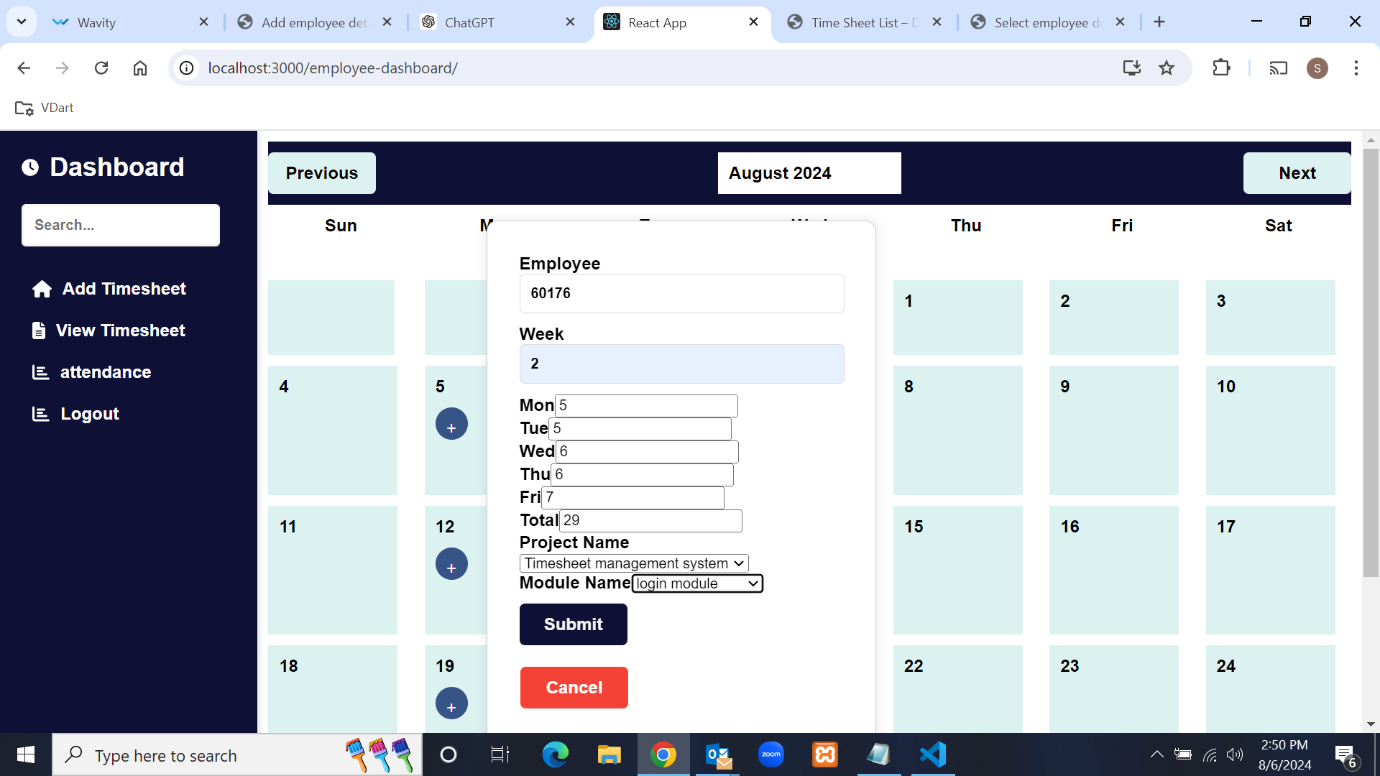
module name, week, Monday, Tuesday, Wednesday, Thursday, Friday, and total hours, along with

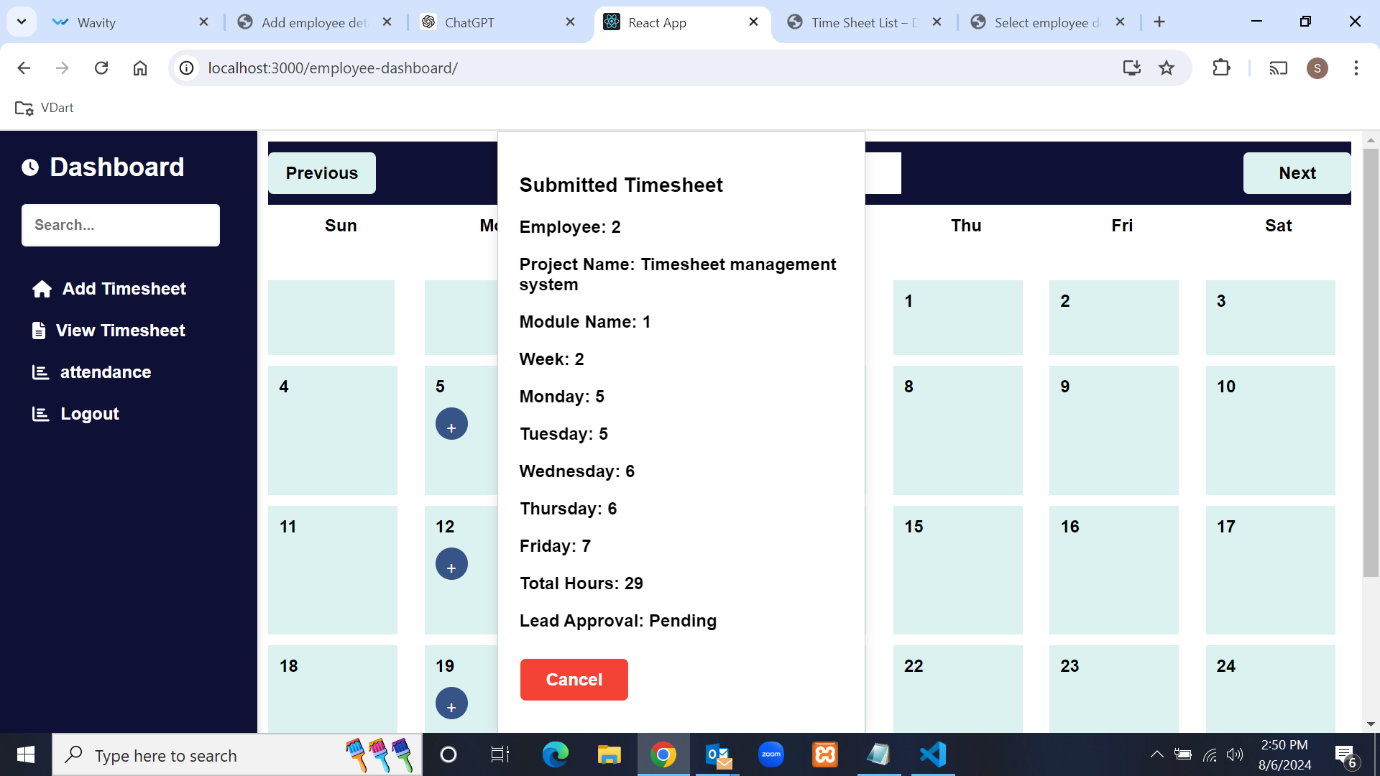
submit and cancel buttons where the employee can submit the data. Other days do not have a "+"

button. After successfully submitting the data, a confirmation message is shown.









Another functionality in the employee dashboard is viewing timesheets. In this section, the

submitted timesheets of the employee are shown in a table format with columns for lead

approval and manager approval.

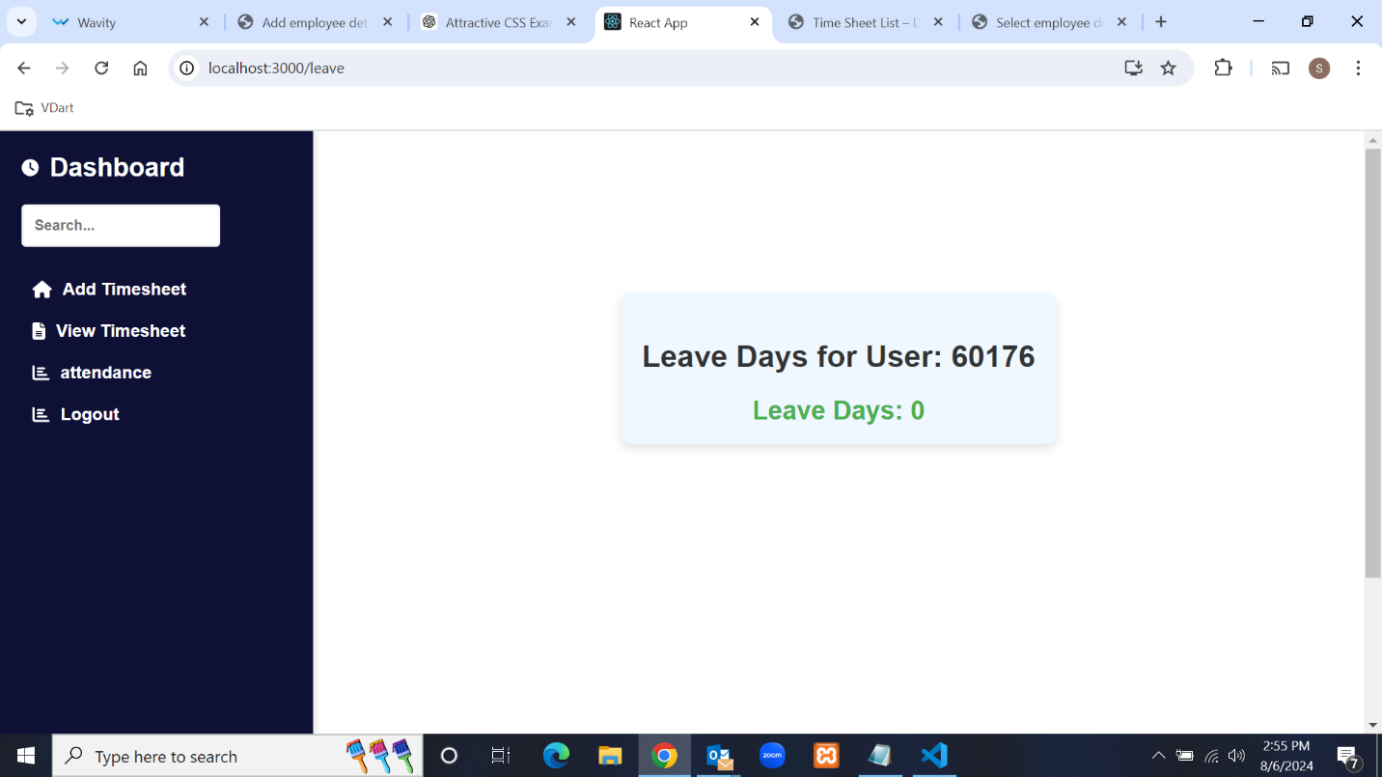


Another functionality of the employee dashboard is calculating employee leave days. If any

weekday in a week is not filled in, it is considered as leave.

**Top of Form**

**Bottom of Form**

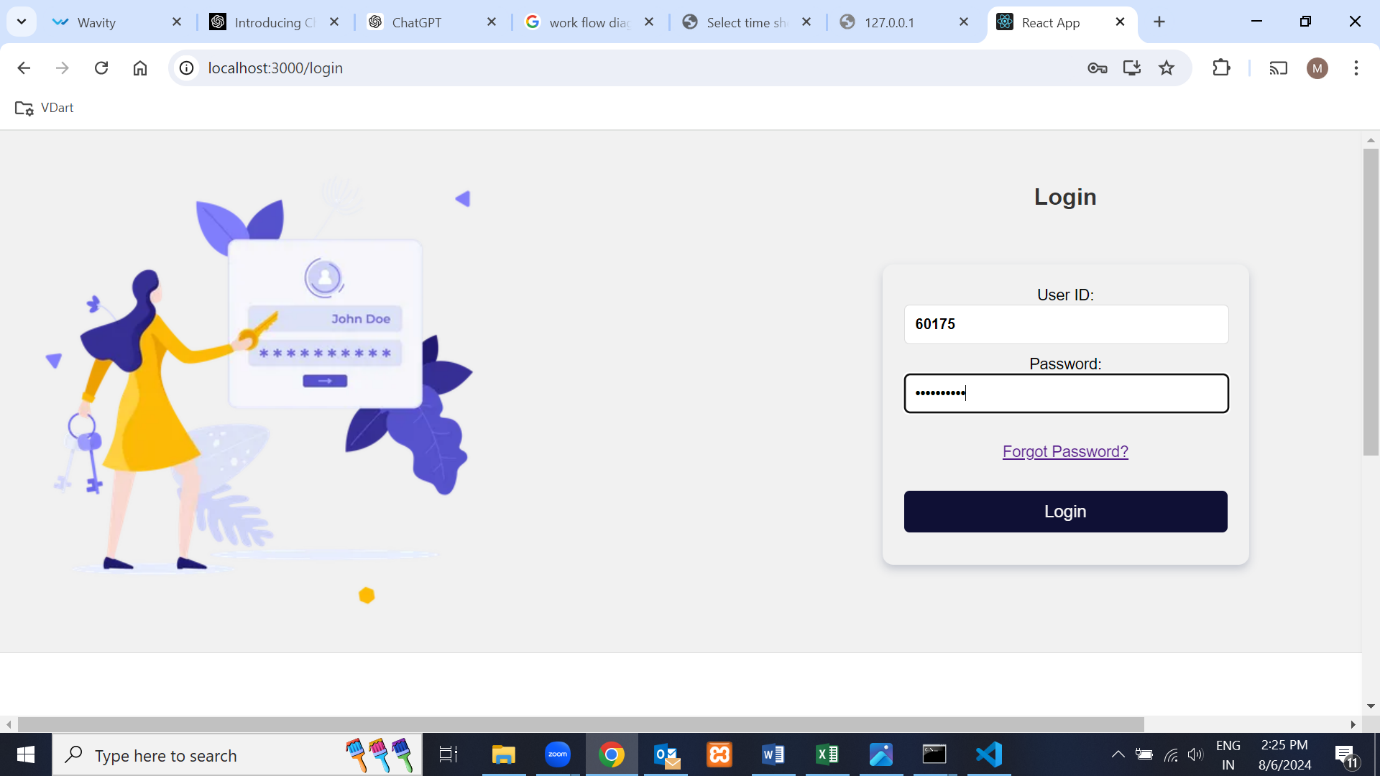
Another functionality in the employee dashboard is logout, which removes the current user ID and

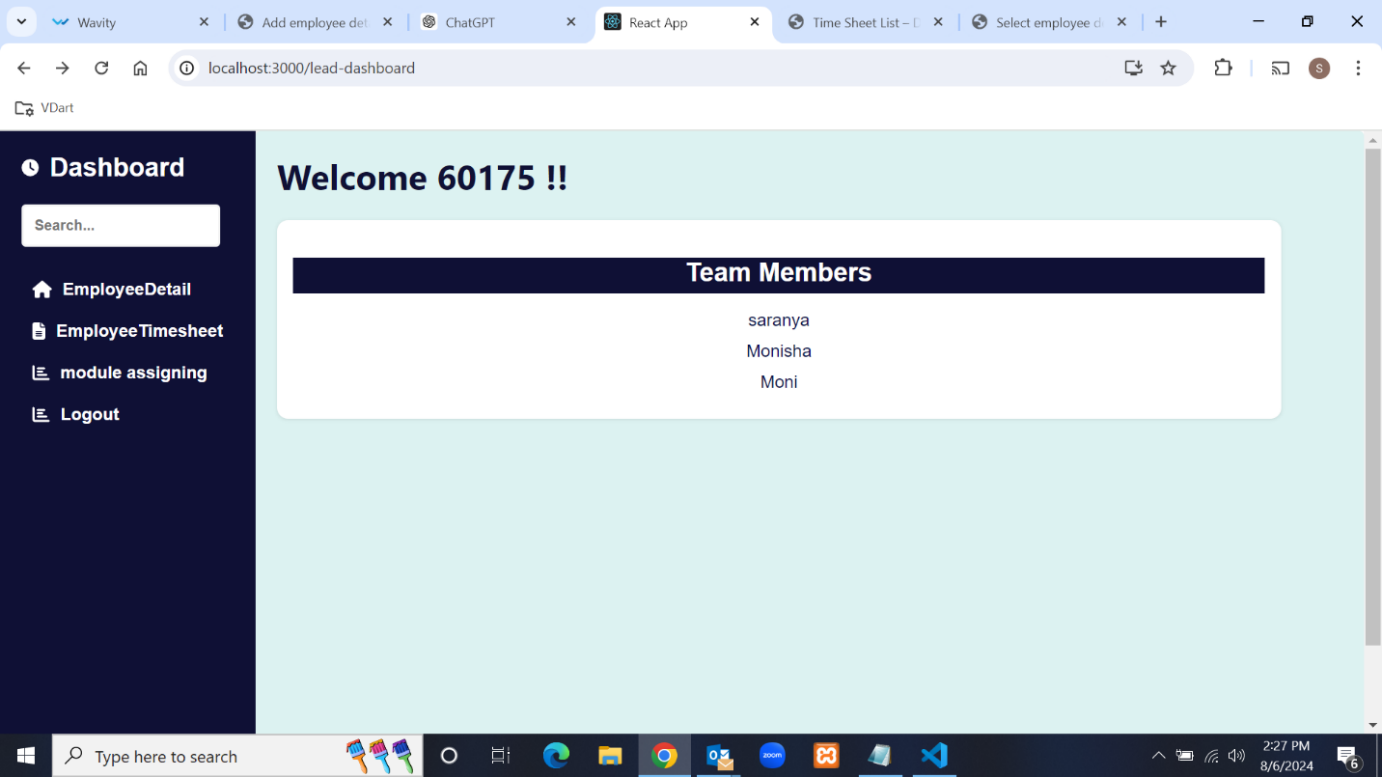
password and redirects to the login page.

**2.Lead Module**

Log in using the employee ID and password. During login, the system checks the person's status.

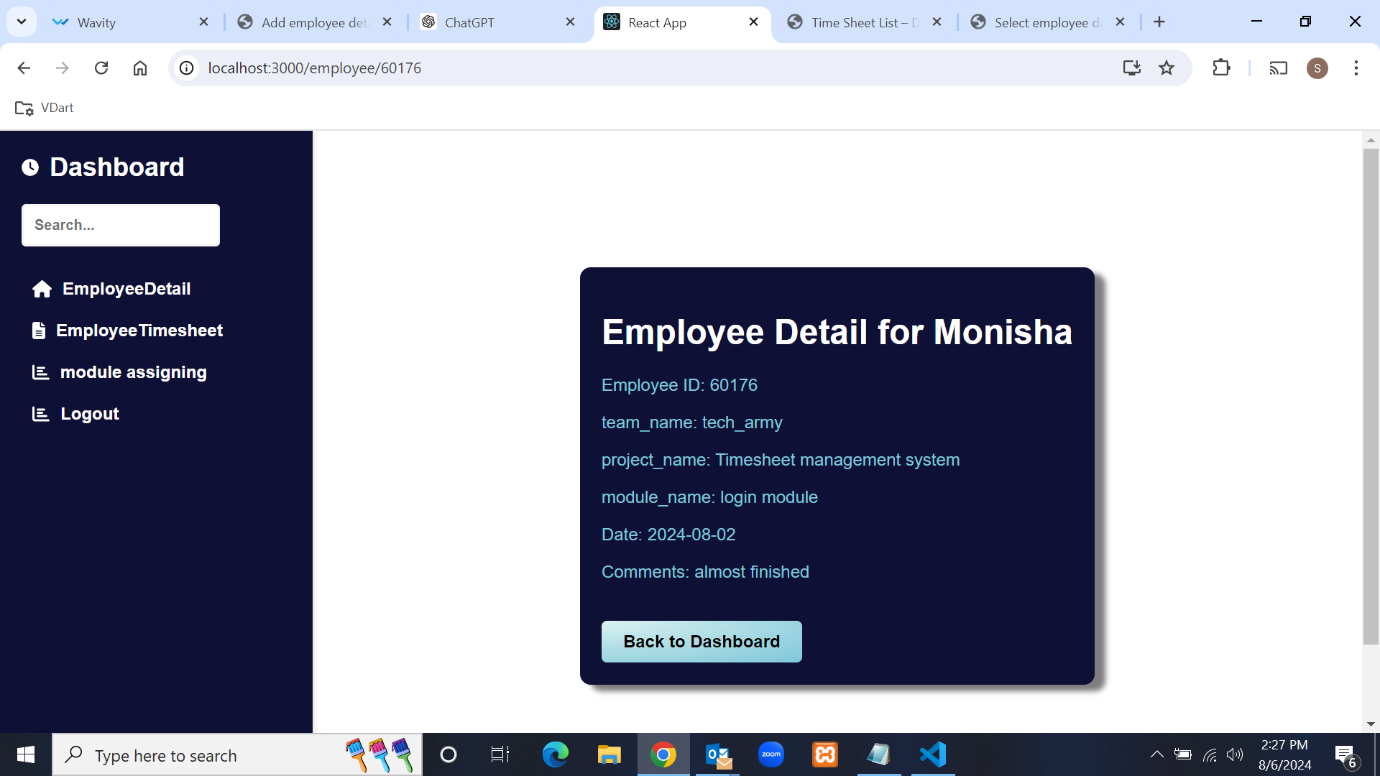
If the status is 'Lead', it redirects to the Lead dashboard.

****



The lead dashboard has three functionalities. The first is that the Lead can view the details

of the employees.

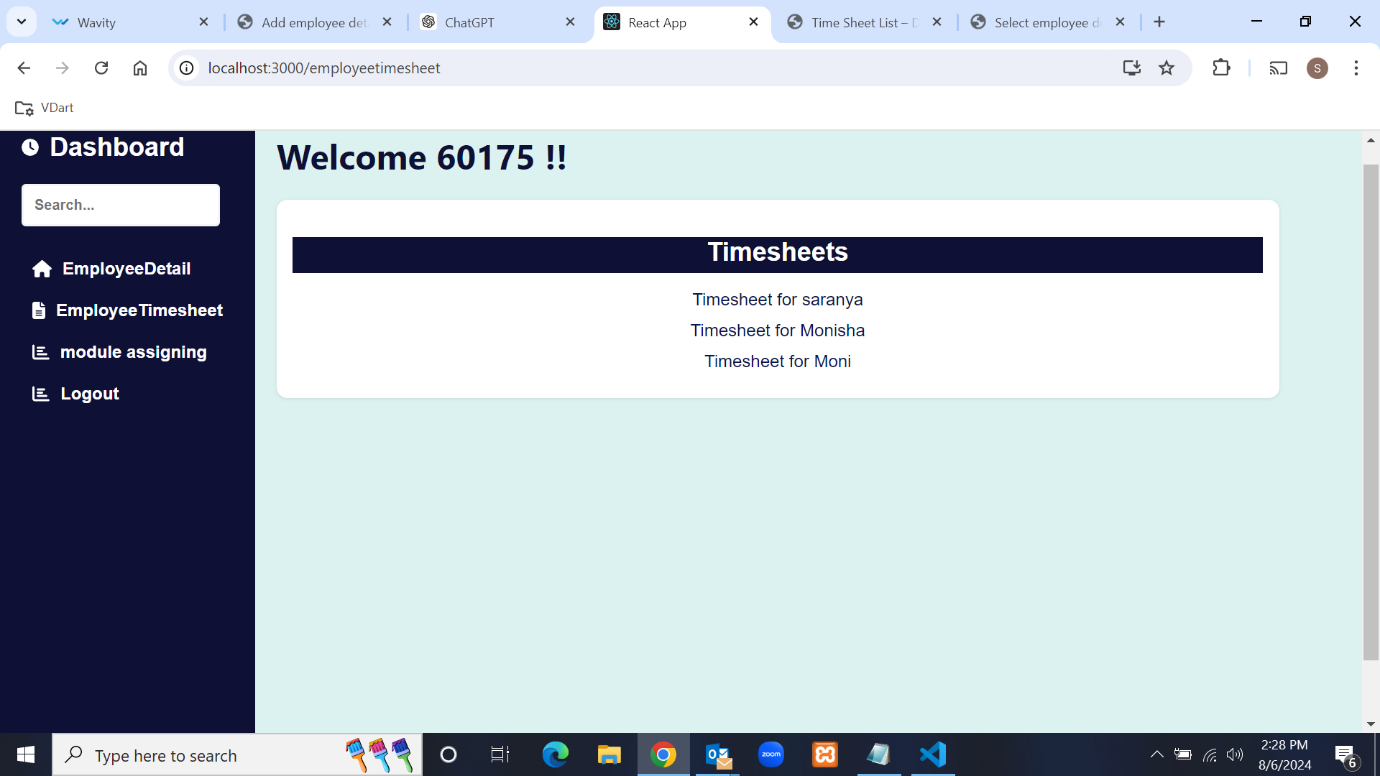


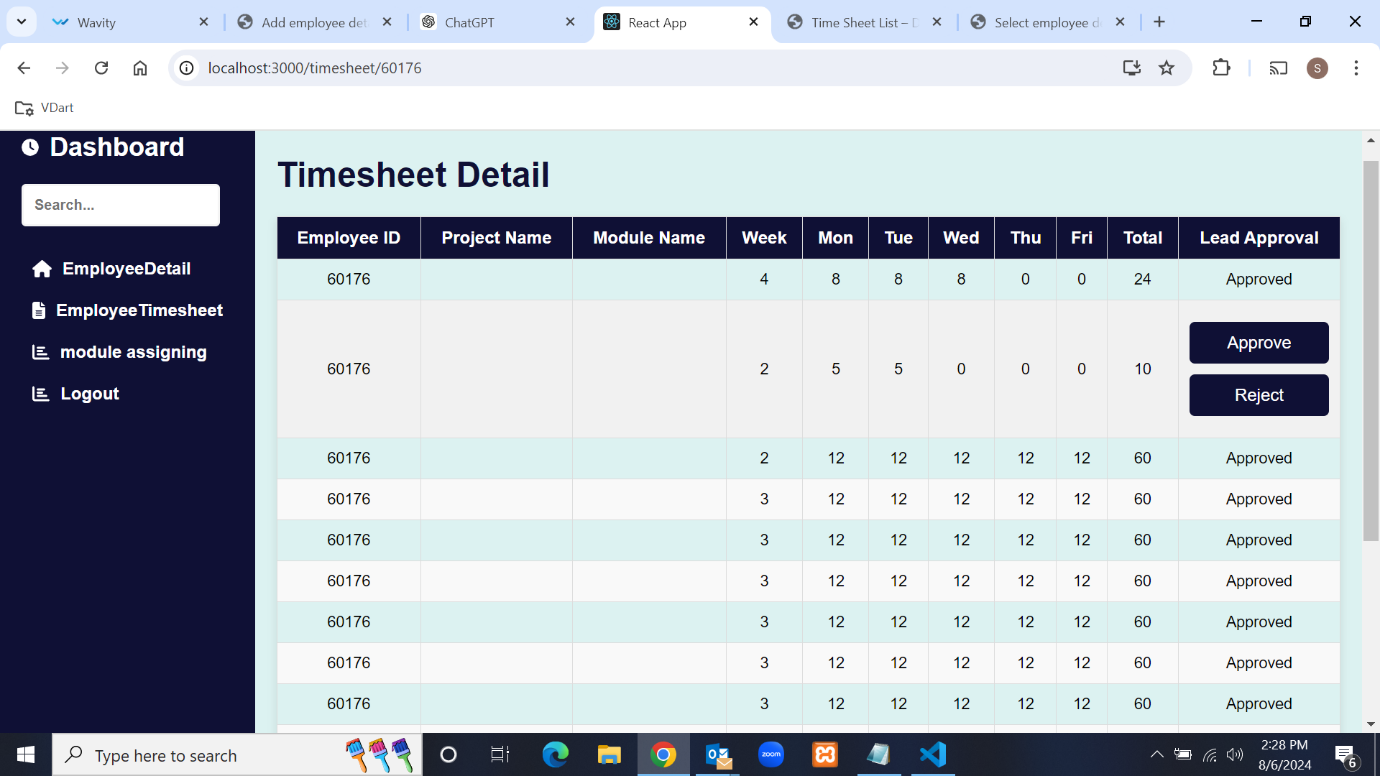
Another functionality of the lead dashboard is the employee timesheet section, where the lead

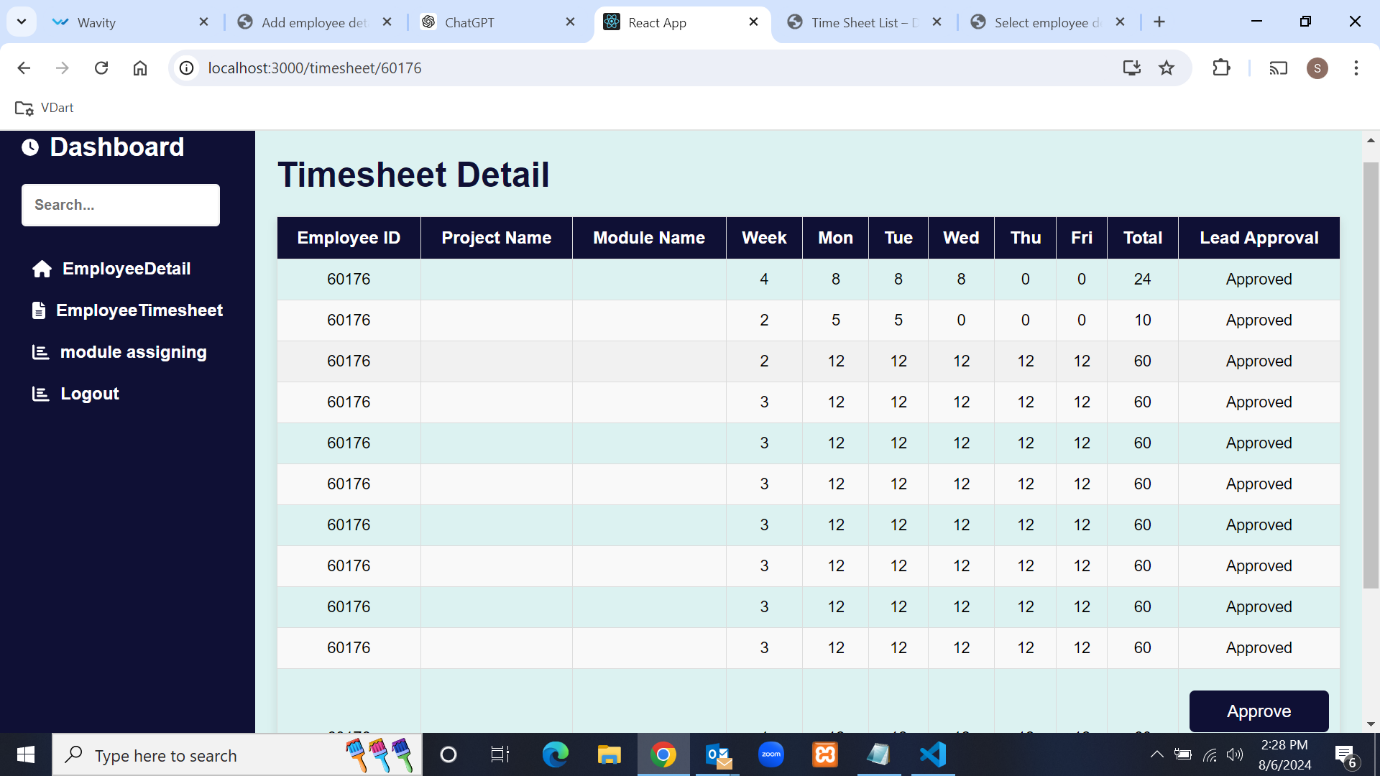
can view employee timesheets in a table format. This section includes a column for lead

approval with two buttons: "Approve" and "Reject." The lead can use these buttons to

approve or reject the employee's timesheet.







Another functionality in the lead dashboard is logout, which removes the current user ID and

password and redirects to the login page.

**3.Manager module:**

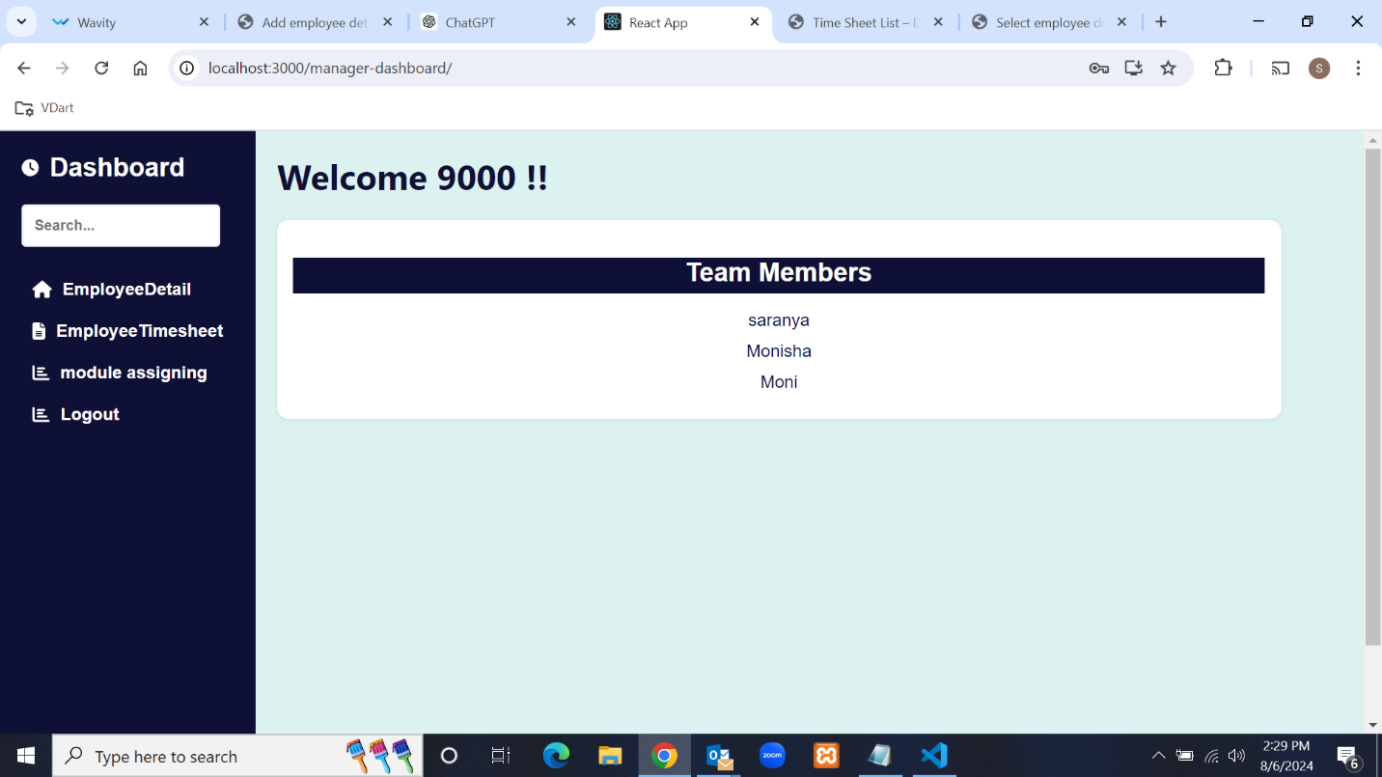
Log in using the employee ID and password. During login, the system checks the person's status.

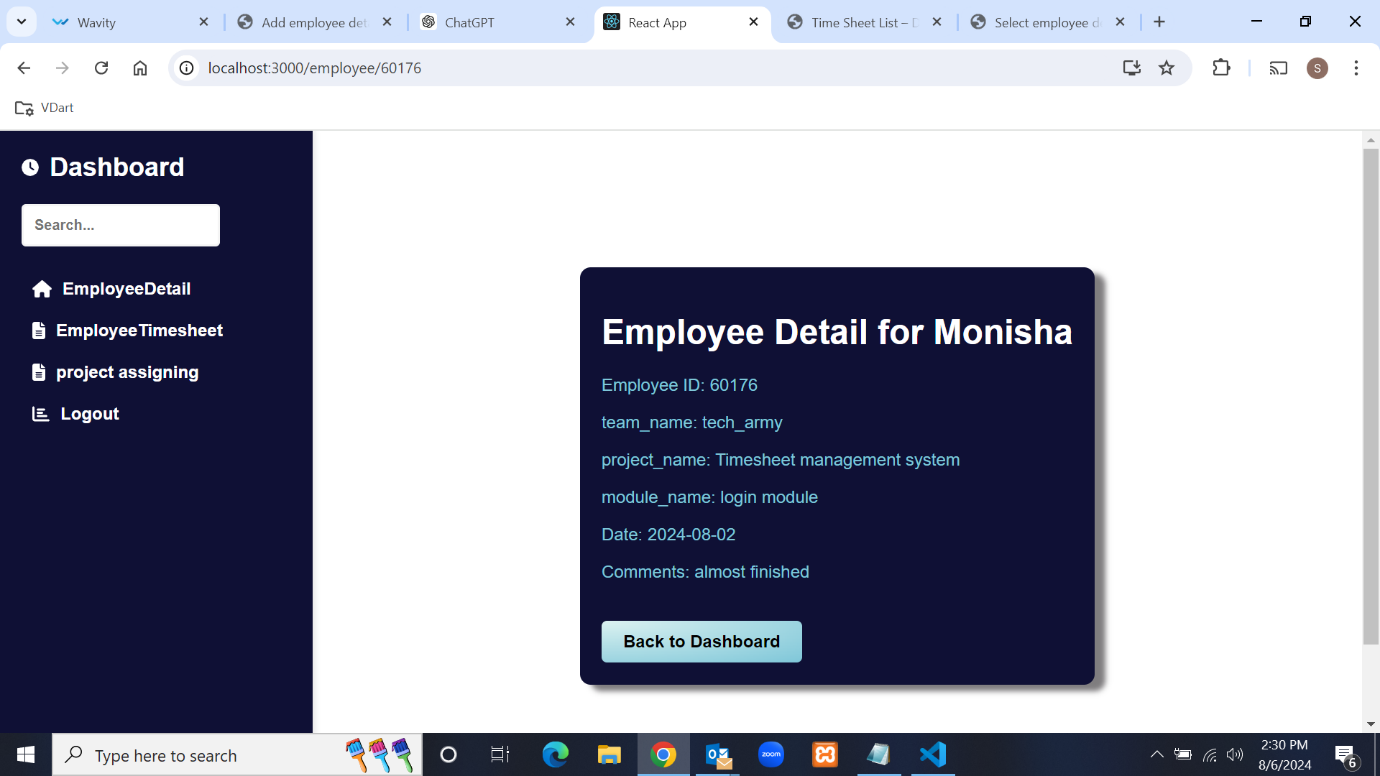
If the status is 'Manager', it redirects to the Manager dashboard.

****

The Manager dashboard has four functionalities. The first is that the Manager can view the details

of the employees.



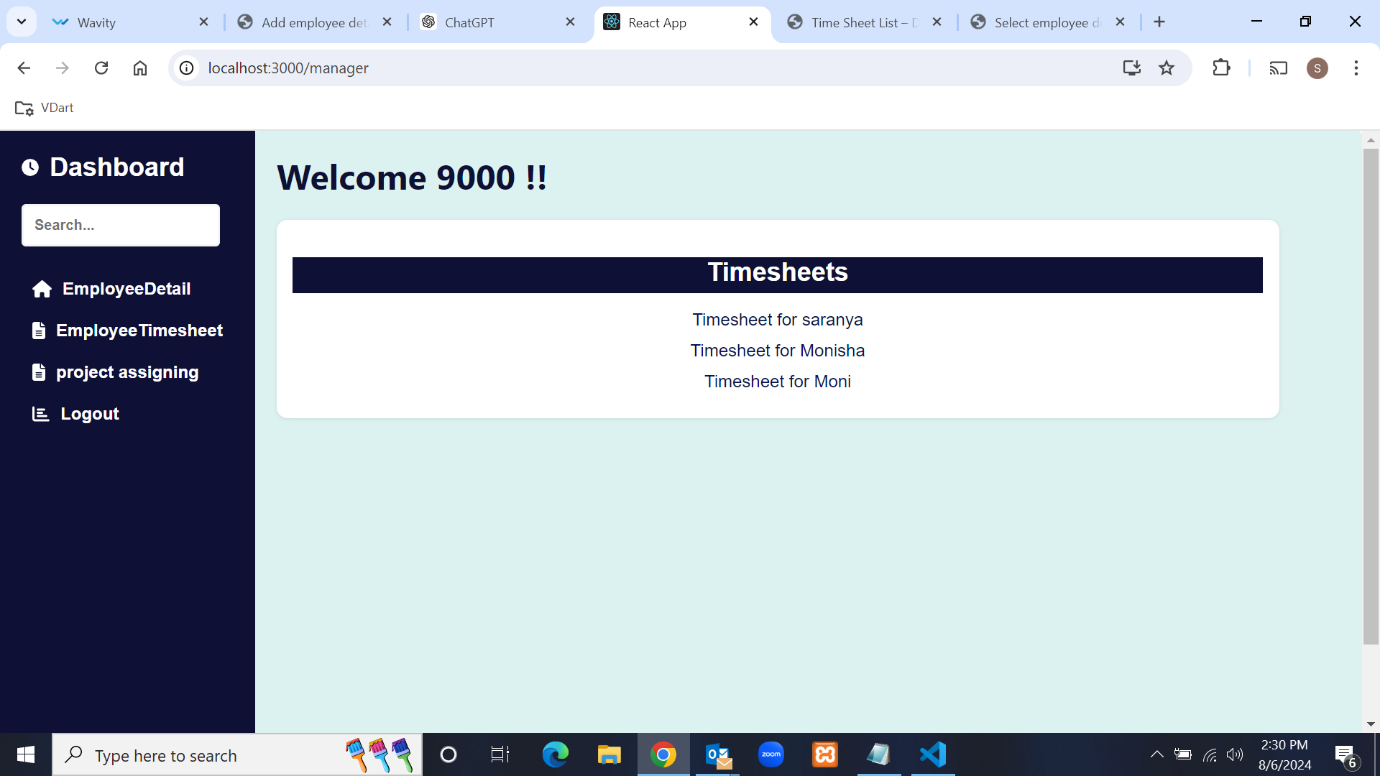


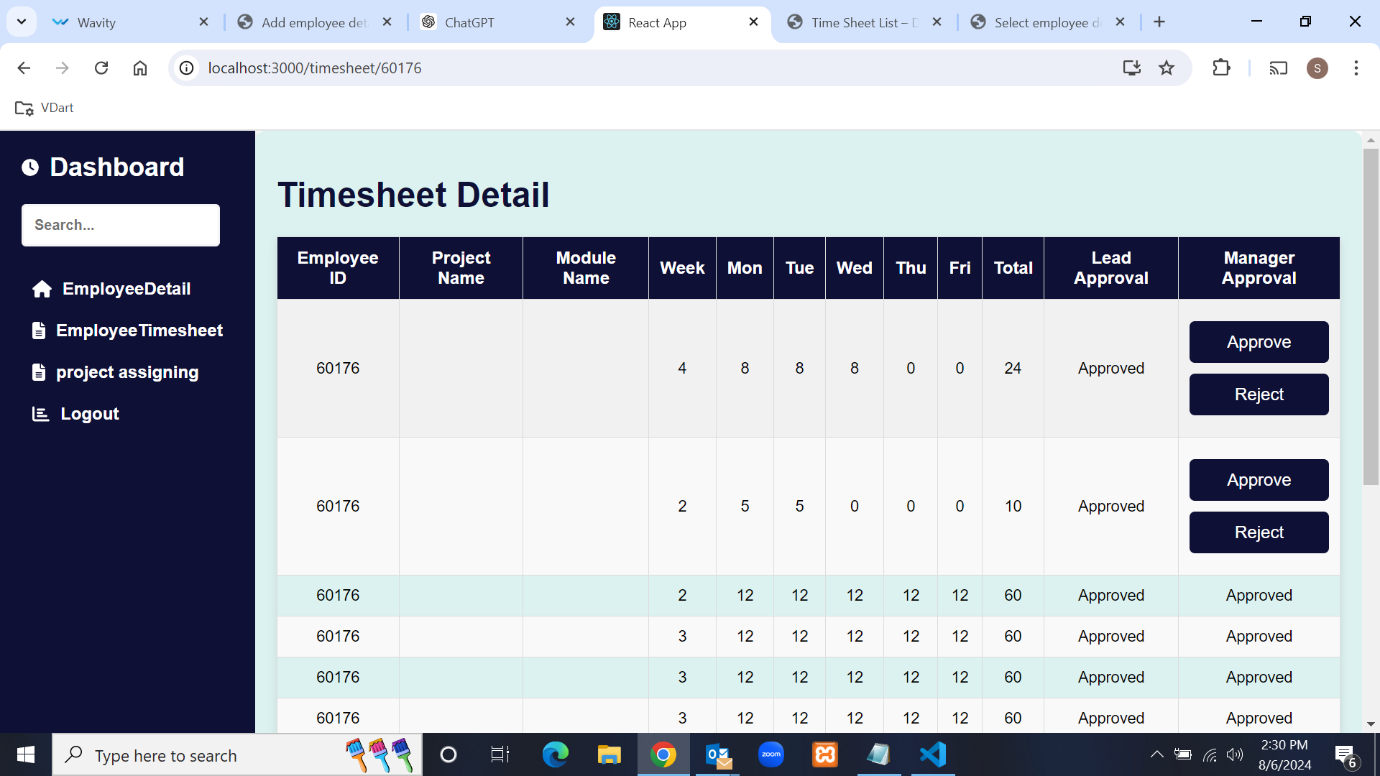
Another functionality of the lead dashboard is the employee timesheet section, where the lead

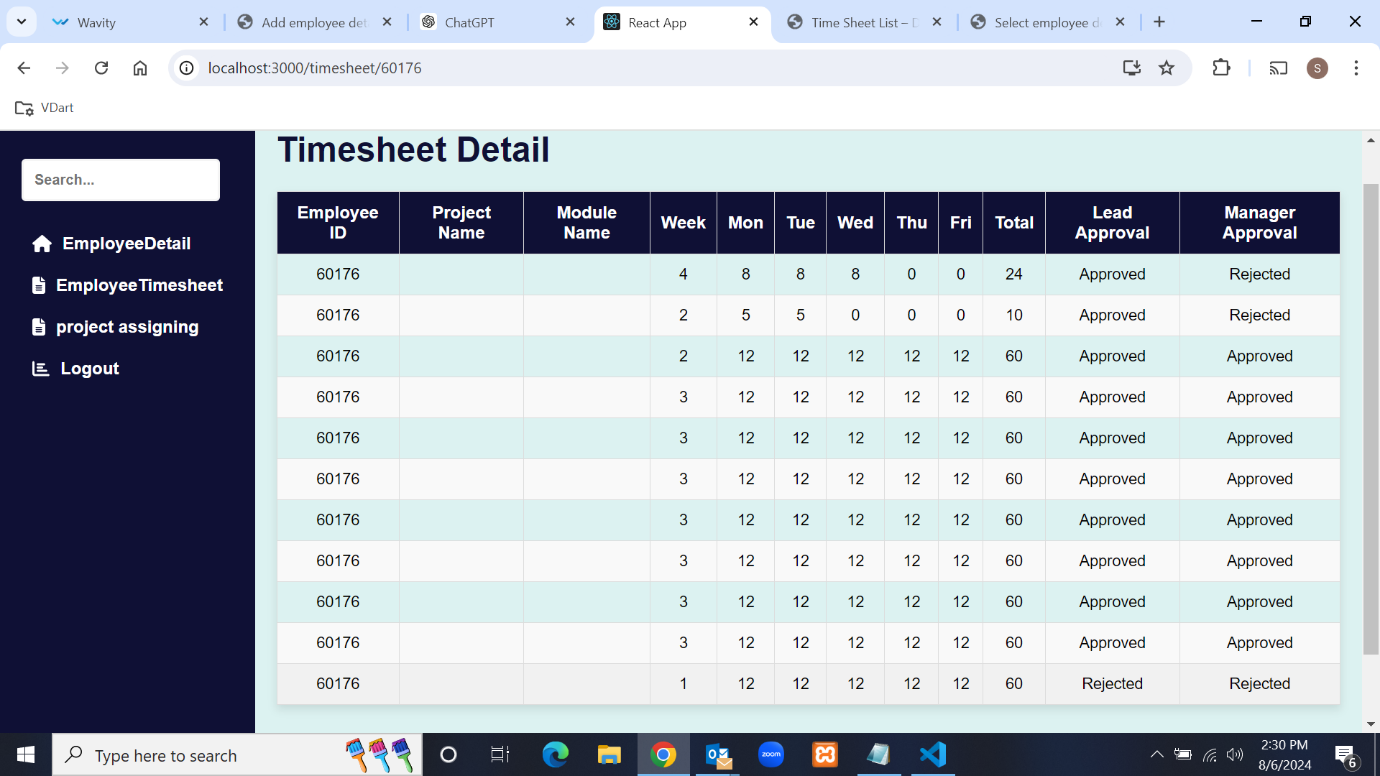
can view employee timesheets in a table format. This section includes a column for

approval with two buttons: "Approve" and "Reject." The lead can use these buttons to

approve or reject the employee's timesheet.



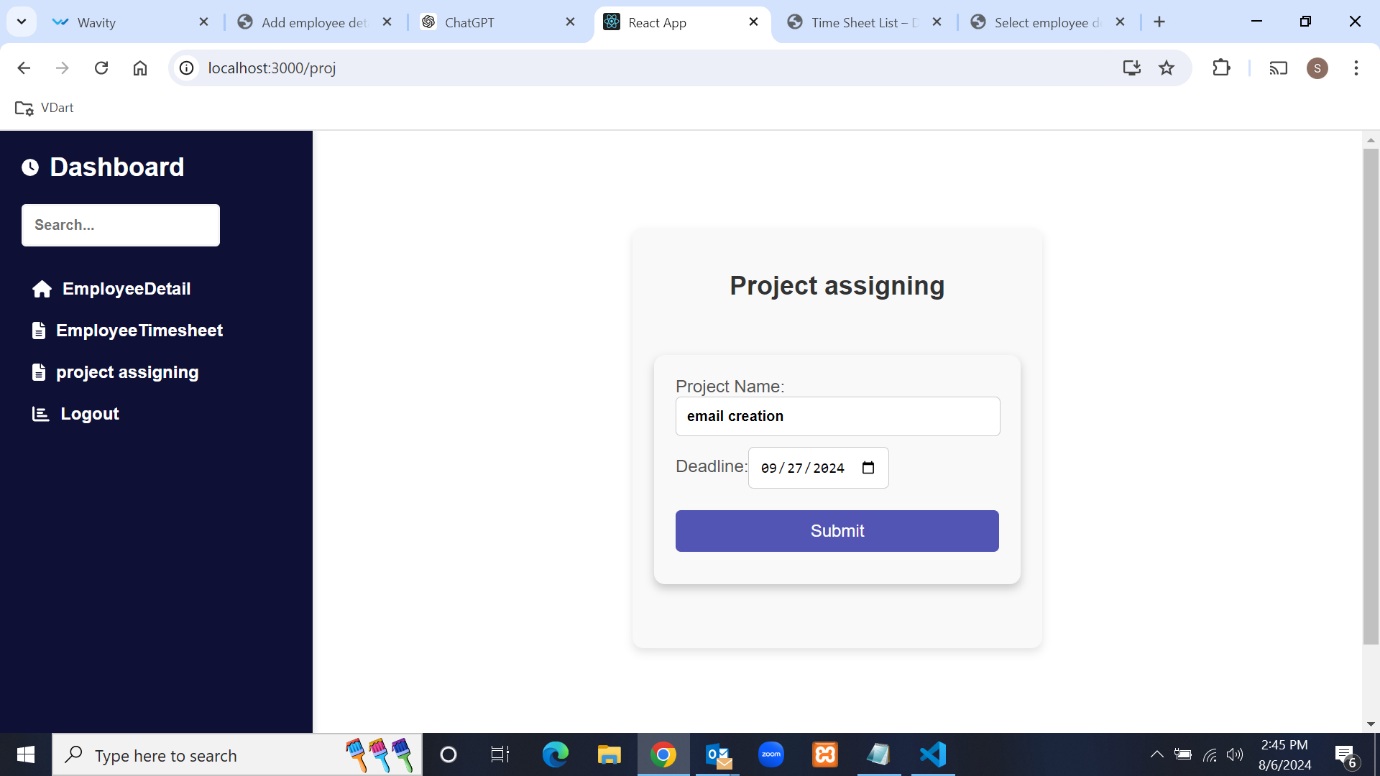


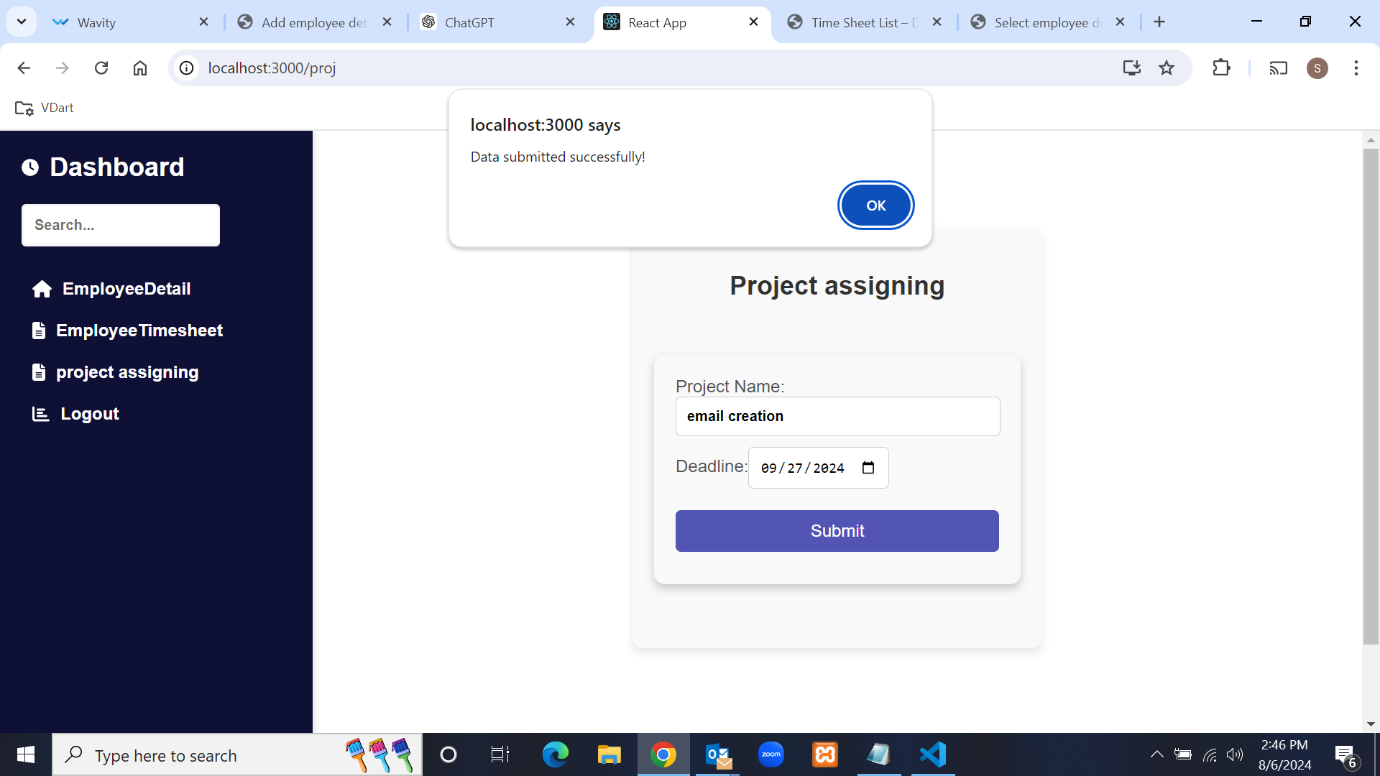


Another functionality of the manager dashboard is project assignment, where the manager

can assign projects and deadlines to employees.



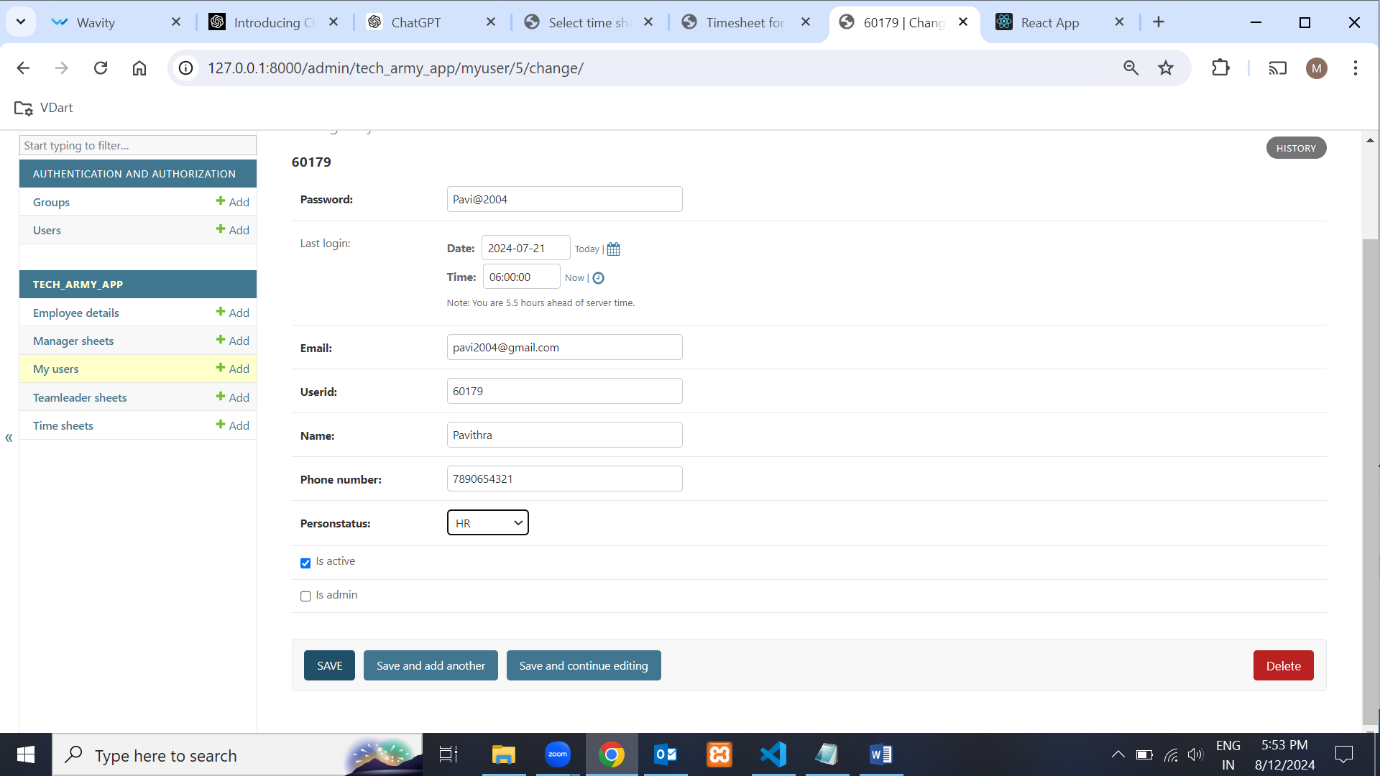


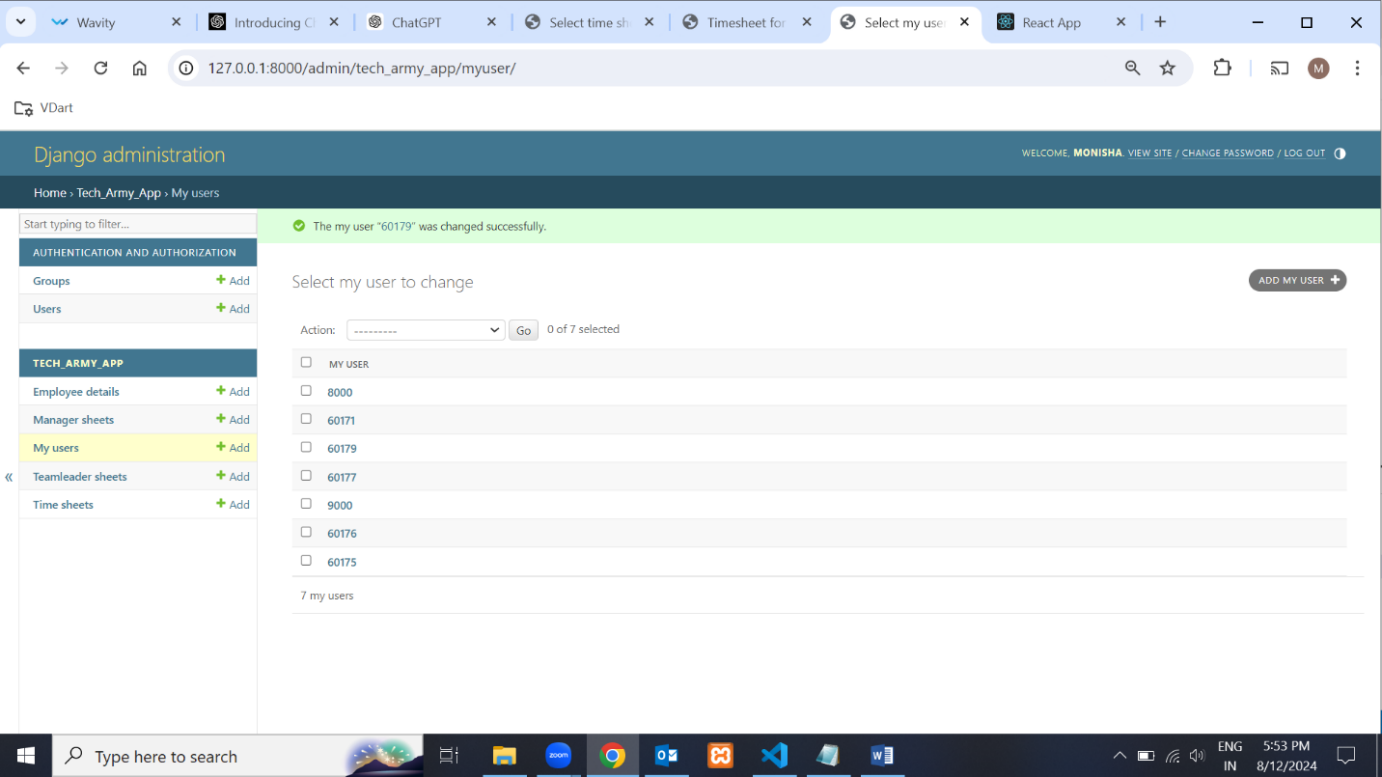


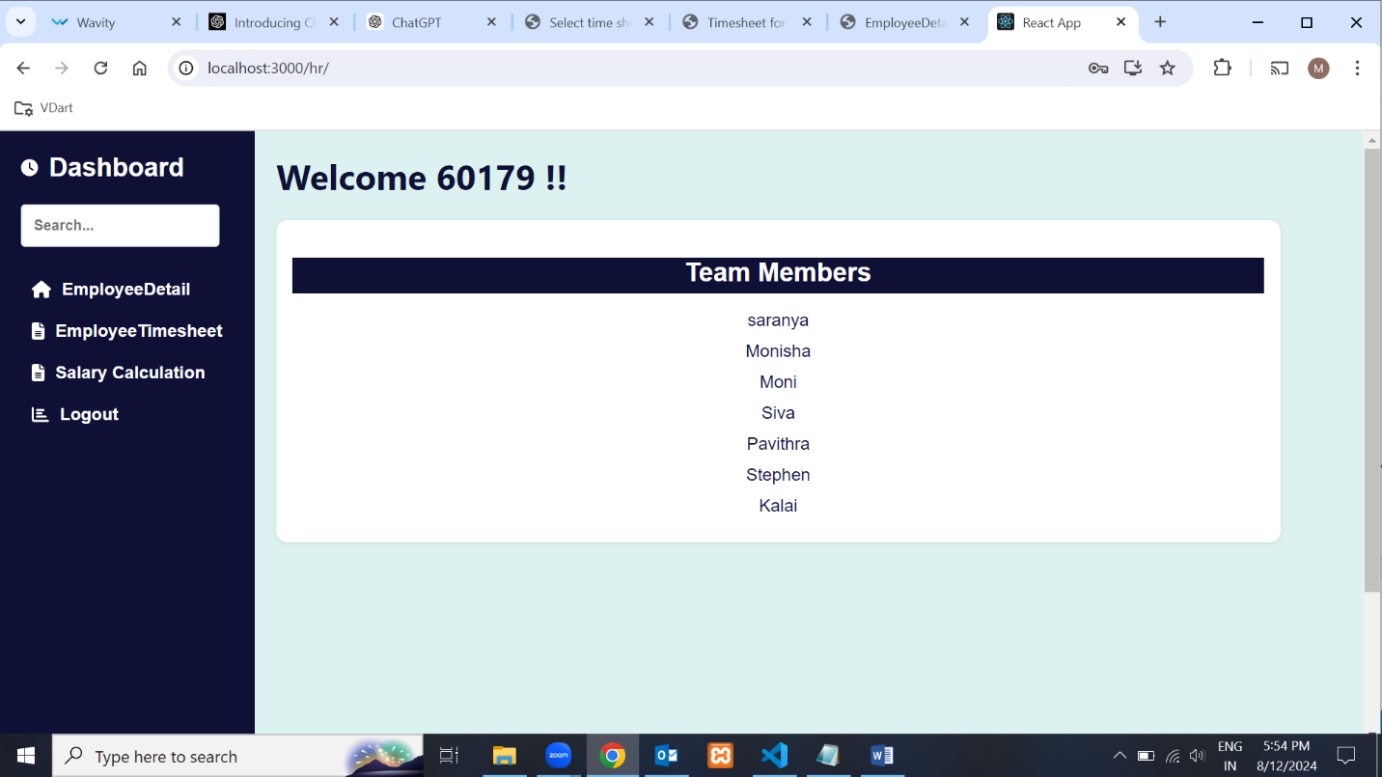
Another functionality in the lead dashboard is logout, which removes the current user ID and

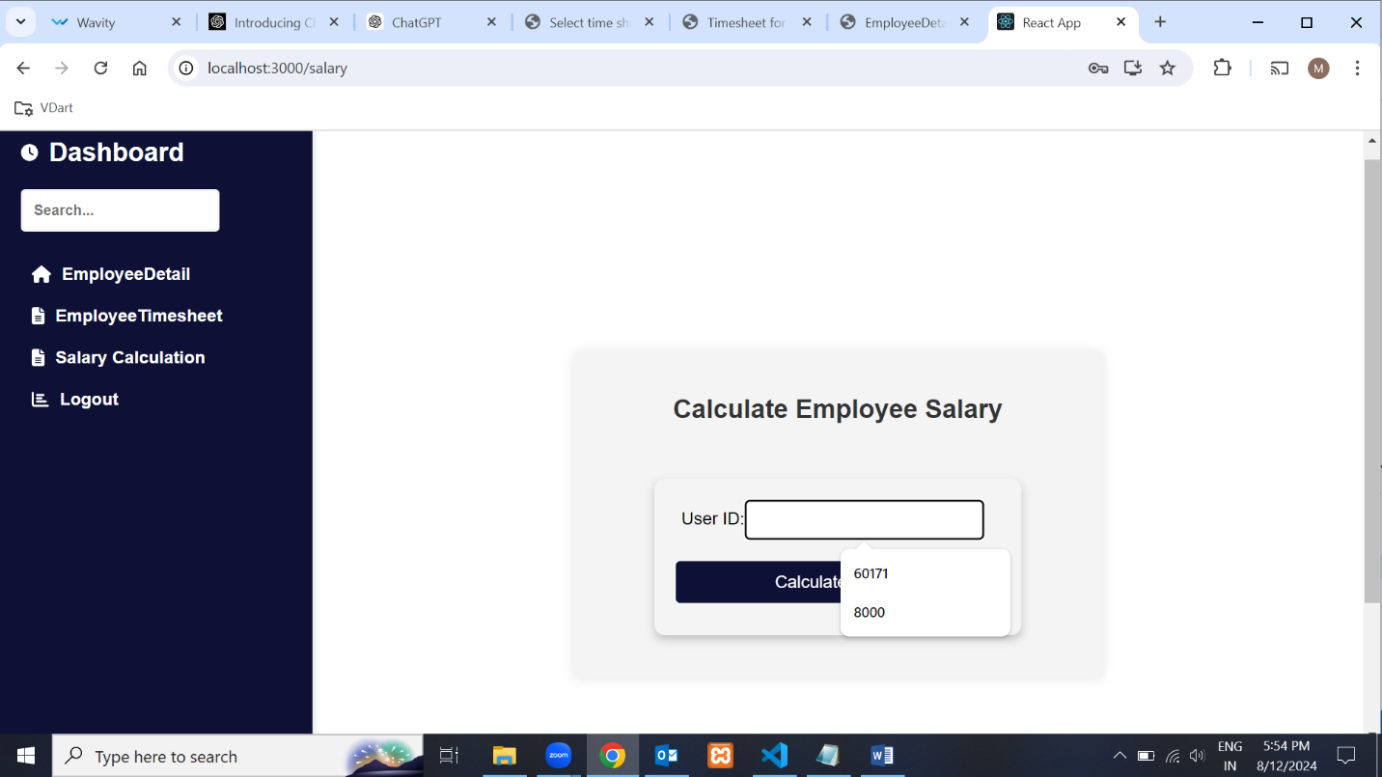
password and redirects to the login page.

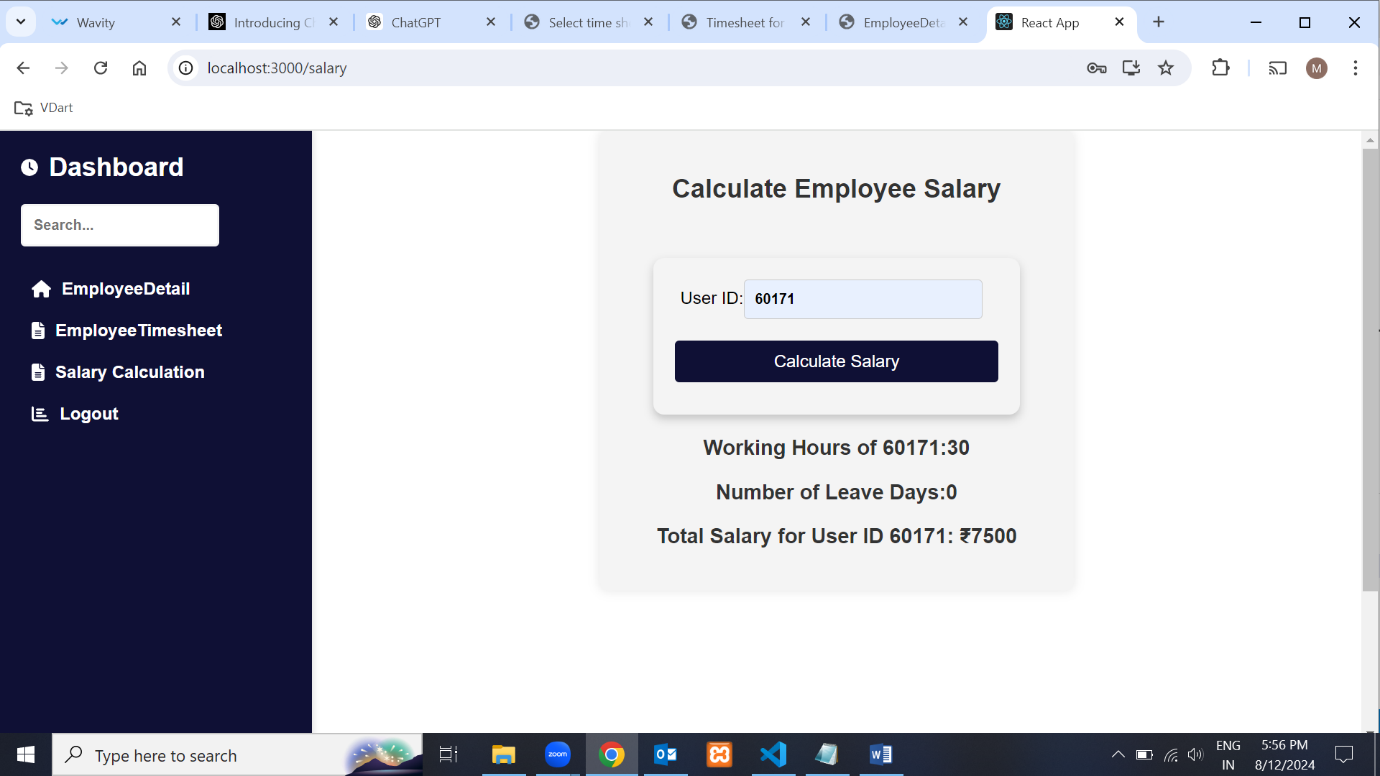
4.HR Module:











**Future Enhancement**

The Timesheet Project is poised for several significant enhancements that will further

refine its functionality and streamline the management of employee work hours. These

improvements include detailed adjustments to submission workflows, new

features for both employees and managers, and the introduction of an HR module

for salary calculations.

Below is a comprehensive outline of the proposed future enhancements:

**Enhanced Submission Workflow**

**Team Leader Specific Submissions**

Currently, timesheets are reviewed by all team leaders, which can lead to inefficiencies. The

enhanced workflow will ensure that an employee's timesheet is only submitted to their assigned

team leader.

**Example Workflow:**

1. Timesheet Submission:Employee A submits their weekly timesheet.

2. Assignment Identification:The system identifies that Employee A is assigned to Team

Leader B.

3. Submission:Employee A’s timesheet is directed solely to Team Leader B.

4. Notification:Team Leader B receives a notification alerting them to review the timesheet.

5. Approval/ Rejection:Team Leader B reviews and approves or rejects the timesheet. The

status is updated accordingly.

This focused approach will optimize the review process, reduce administrative burden, and

ensure that timesheet evaluations are handled by the appropriate authority.

**Improved Timesheet Entry and Calculation**

**Detailed Daily Entries**

Employees will be required to input their working hours with greater specificity:

1. Daily Hours Entry:Employees will specify start and end times for each workday.

2. Date and Day Information:Each entry will include the date and day of the week.

3. Automatic Calculation:The system will automatically calculate the total number of hours

worked each day based on the input start and end times.

4. Weekly Aggregation:Calculated total hours for each day will be stored in a weekly

database, compiling the weekly timesheet.

This enhancement will ensure accuracy in timesheet entries and automate the calculation

process, reducing errors and manual calculations.

**Leave Day Calculation**

To account for non-submission of daily timesheets:

1. Exclusions:Sundays and common public holidays (e.g., Pongal, Diwali) will be

excluded from this leave day calculation.

This feature will facilitate accurate leave tracking and ensure compliance with holiday policies.

**Update Timesheet Feature for Leads and Managers**

An 'Update Timesheet' feature will be added for Leads and Managers, similar to that

available to Employees:

1. **Modification Capabilities**:Leads and Managers will have the ability to update

timesheets, ensuring that corrections can be made post-submission.

1. **Approval and Rejection**: Updated timesheets will require review and

re-approval if modifications are made.

This feature will enhance flexibility and accuracy in timesheet management for higher-level

users.

**Break and Submission Notifications**

1.**Continuous Work Notifications**:Employees who work continuously for more than 4

hours will receive a notification reminding them to take a break. This feature aims to

promote employee well-being and adherence to work-rest balance guidelines.

2.**Timesheet Submission Reminders**:Employees will receive a notification within

5 minutes of logging in, reminding them to submit their timesheet if it has not been

done already.

These notifications will help maintain compliance and support employee health.

**HR Module for Salary Calculation**

An HR module will be introduced to calculate employee salaries based on recorded working

hours:

1. **Salary Calculation**:The module will compute salaries considering the total hours worked,

leave days, and any other applicable deductions or bonuses.

1. **Integration**:The HR module will integrate with the timesheet database to fetch accurate

work-hour data for salary computations.

This addition will automate and streamline salary processing, reducing manual effort and

ensuring accuracy.

**Manager and Lead Module Enhancements**

**Manager Module**

1.**Project Management**:The Manager module will include features to manage project

names, deadlines, and assignment of team leaders based on employee status and ID.

**Lead Module**

1**.Project and Module Assignment**:The Lead module will incorporate the project name

assigned by the manager and allocate the module name to employees based on their person

status and ID.

These enhancements will improve project tracking and ensure that team assignments

and responsibilities are well-defined and managed effectively.

**Conclusion**

The timesheet project is designed to streamline the management of employee working

hours and facilitate approval processes across different roles. By providing distinct

modules for employees, leads, and managers, the project ensures that each user type

has access to the tools and functionalities required for their specific responsibilities.

The integration of these modules into a cohesive system ensures that timesheet

management is efficient, transparent, and aligned with organizational requirements.

By handling authentication, data submission, and approval processes effectively,

the project enhances productivity and accountability within the organization.