**Timesheet and Feedback System**

Major Project

Contents

[1 Project Overview 2](#_Toc163808305)

[1.1.1 Brief introduction of the project. 2](#_Toc163808306)

[1.1.2 Purpose of the project. 2](#_Toc163808307)

[1.1.3 Goals of the project: 2](#_Toc163808308)

[2 Project Scope 3](#_Toc163808309)

[2.1.1 Web app support 3](#_Toc163808310)

[2.1.2 Key features and functionalities of the web app. 3](#_Toc163808311)

[3 Architecture and Technology Stack 3](#_Toc163808312)

[3.1.1 Overall architecture of the mobile app. 3](#_Toc163808313)

[3.1.2 Architecture 4](#_Toc163808314)

[3.1.3 Technology stack (e.g., programming languages, frameworks, libraries). 4](#_Toc163808316)

[3.1.4 Rationale behind the chosen technology stack. 5](#_Toc163808317)

[4 Web App Components 7](#_Toc163808318)

[4.1.1 Main components of the mobile app. 7](#_Toc163808319)

[4.1.2 Purpose of each component. 7](#_Toc163808320)

[5 User Interface Design 8](#_Toc163808321)

[5.1.1 User interface (UI) 8](#_Toc163808322)

[6 Work with Data 12](#_Toc163808323)

[6.1.1 Approach for the Data Engineering. 12](#_Toc163808324)

[6.1.2 Approach for the Machine Learning. 12](#_Toc163808325)

[7 Project Resources 14](#_Toc163808326)

[7.1.1 Roles and responsibilities of the project team members. 14](#_Toc163808327)

[8 Risks and Mitigation Strategies 14](#_Toc163808328)

[8.1.1 Potential risks and challenges associated with the project. 14](#_Toc163808329)

# Project Overview

### Brief introduction of the project.

Timesheet and Feedback System is a web-based application designed to streamline the process of tracking employees' daily work activities and collecting feedback on projects they are involved in. This system is intended for use by employees working on various projects within an organization. It provides a centralized platform where employees can log their daily tasks, hours worked, and submit feedback for the projects they are assigned to.

### Purpose of the project.

The purpose of the Employee Timesheet and Feedback System project is to provide an efficient and user-friendly solution for managing and tracking the daily work activities of employees working on various projects within an organization. The project serves several key purposes:

1. **Timesheet Management**:
   * Enable employees to log their daily tasks, hours worked, and descriptions of work completed.
   * Facilitate accurate tracking of project progress and employee productivity.
2. **Project Feedback Collection**:
   * Allow employees to provide feedback on the projects they are involved in, including suggestions, concerns, and overall impressions.
   * Provide a platform for employees to share their insights and opinions, contributing to continuous improvement efforts.
3. **Transparency and Accountability**:
   * Promote transparency by providing visibility into employees' daily work activities and project contributions.
   * Enhance accountability by ensuring that employees accurately report their work hours and progress.
4. **Communication and Collaboration**:
   * Facilitate communication between employees, managers, and project teams through feedback submission and review.
   * Foster collaboration by providing a centralized platform for sharing project-related information and updates.
5. **Efficiency and Productivity**:
   * Streamline timesheet management and feedback collection processes, reducing administrative overhead and improving efficiency.
   * Empower employees to focus on their core tasks and responsibilities by automating routine tasks related to time tracking and feedback submission.

### Goals of the project:

* Develop an Employee Timesheet and Feedback System.
* Ensure the platform is intuitive and easy to navigate for both administrators and employees.
* Define the scope of work for the Development Phase based on the Requirements Analysis Phase.
* Deliver a Minimum Viable Product (MVP) by the end of the development phase, aligned with the scoped requirements.
* Design the core application tier, including:
  + Front End tier for the end-user (employee), accessible via a web application.
  + Back End API provided by JMAN.
* Conduct QA testing post-development to ensure quality and functionality.
* Submit the application for user acceptance testing (UAT).
* Address any testing failures by fixing defects and rectifying functional gaps at no additional cost during UAT.

# Project Scope

### Web app support

|  |  |
| --- | --- |
| Platforms | Version |
| Google Chrome | 97 and above |
| Microsoft Edge | 97 and above |

### Key features and functionalities of the web app.

**User Registration and Authentication:**

User-friendly registration process for admin to create employee accounts.

Secure authentication mechanisms, including username/password.

**Personalized Admin Dashboard**:

Customized dashboard for admins to create users, add projects and allocate resources.

**Personalized Employee Dashboard**:

Customized dashboard for each employee, display the timesheet to enter their work , after submitting the timesheet, feedback will be opened.

**Responsive Design:**

Responsive user interface for optimal viewing and usability across different screen sizes.

# Architecture and Technology Stack

### Overall architecture of the web application.

Build a timesheet and feedback system offering secure login, timesheet submission, feedback provision, email notifications, and workload management. Establish a robust data platform for reporting and analytics, enhancing efficiency and employee engagement.

**Front End Application:**

* User Registration and Authentication: This module facilitates user registration and authentication processes, offering a secure login page for employees. It includes mechanisms for username/password authentication.
* Timesheet Creation: Allows employees to create, update, and submit timesheets for their daily work activities. Provides an intuitive interface for entering time worked, tasks completed, and project details.
* Feedback Submission: Enables employees to provide feedback on projects they are working on. Includes options for rating satisfaction, providing comments, and suggesting improvements.
* Capacity Management: Helps managers and administrators track employee workload and project capacity, ensuring efficient resource allocation and workload balancing.

**Back End Services:**

* Authentication Service: Manages user authentication and authorization, verifying user credentials and enforcing access controls.
* Timesheet Management Service: Handles the creation, updating, and submission of timesheets, including validation and approval workflows.
* Feedback Management Service: Manages feedback submissions from employees, including creation, updating, and notification workflows.
* Email Notification Service: Sends automated email notifications to employees for reminders, deadlines, and feedback requests.
* Capacity Tracking Service: Tracks employee workload and project capacity, providing insights for resource planning and allocation.

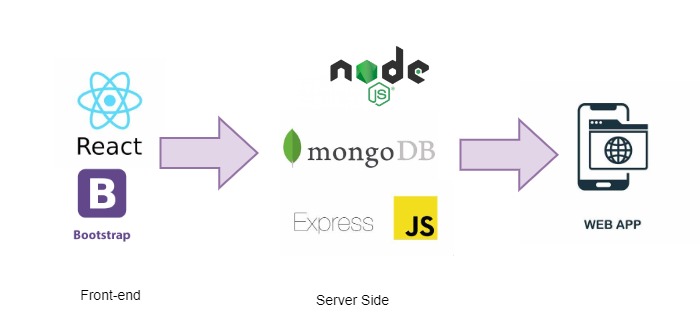
**Data Engineering Pipeline:**

* Data Extraction: Connects to various data sources to extract relevant data related to timesheet entries, user profiles, historical feedback submissions, etc.
* Data Cleaning and Transformation: Cleans and preprocesses the extracted data to ensure data quality and consistency, transforming it into a suitable format for analysis.
* Reporting Tables: Creates reporting tables and views to support operational reporting and analytics, providing insights into timesheet submissions, feedback trends, etc.

**Machine Learning Pipeline:**

* Feature Engineering: Extracts relevant features from the data, including employee attributes, timesheet characteristics, historical feedback, etc., to feed into the model.
* Model Training: Trains machine learning models using data to predict who need to have attendance regularisation in future based on the timesheet submissions and feedback submission patterns.

### Architecture



### Technology stack (e.g., programming languages, frameworks, libraries).

**Technology Stack for the Web Application Platform:**

The technology stack for the Mobile Application Platform for employee learning platform will include the following components:

**React JS**:

React JS is an open-source JavaScript library for building user interfaces. It is widely used for creating interactive and dynamic web applications with a focus on component-based architecture.

**Bootstrap:**

Bootstrap is a popular front-end framework that provides developers with a collection of responsive and customizable UI components and utilities. Inspired by Twitter's design language, Bootstrap enables the creation of sleek and mobile-friendly web interfaces with ease.

**Axios:**

Axios is a JavaScript library for making HTTP requests in both the browser and Node.js environments. With a simple and intuitive API, Axios simplifies the process of sending and receiving data between the client and server. It supports asynchronous operations and works seamlessly with JSON data.

**Node.js:**

Node.js is a versatile and powerful platform for building modern web applications, APIs, and micro-services, offering high performance, scalability, and a vibrant ecosystem of tools and libraries. Its combination of JavaScript on both the client and server-side streamlines development and enables full-stack JavaScript development.

**MongoDB:**

MongoDB is a versatile and scalable database solution that is well-suited for a wide range of use cases, including web applications, mobile apps, content management systems, and real-time analytics. Its flexible data model, scalability, and rich feature set make it a popular choice for modern application development.

### Rationale behind the chosen technology stack.

In developing the Web Application Platform for Timesheet and Feedback System, we have chosen React JS as the primary technology stack. The rationale behind this selection is based on the following considerations:

**Cross-Platform Compatibility:**

While React.js itself is primarily focused on building web applications, the React ecosystem provides various tools, frameworks, and libraries that enable developers to achieve cross-platform compatibility for their applications, extending the reach and usability of React applications across different platforms and devices.

**Time and Cost Efficiency:**

By utilizing React JS, we can optimize development resources and streamline the development process. This efficiency enables faster time-to-market, ensuring the application reaches the organisation sooner.

**Large Developer Community and Ecosystem:**

React JS boasts a vast and active developer community. This thriving community provides extensive support, resources, and libraries that accelerate development and problem-solving. The availability of a wide range of open-source libraries and components allows us to leverage existing solutions and integrate additional functionalities efficiently.

**Code Reusability:**

React JS component-based architecture promotes code reusability. By developing reusable components, we can efficiently manage the application's UI elements and logic across different screens. This approach not only simplifies development but also facilitates maintenance and future updates, as changes made to shared components propagate throughout the application.

In developing the Data Engineering Pipeline for Timesheet and Feedback System, we have chosen Snowflake and Data Build Tool (dbt) as the primary technology stack. The rationale behind this selection is based on the following considerations:

**Scalability:**

Snowflake is a cloud-based data warehouse that offers elastic scalability. It can handle large volumes of data and concurrent queries, making it suitable for growing businesses and handling fluctuating workloads without performance degradation.

**Security and Compliance:**

Snowflake provides robust security features, including role-based access control (RBAC), encryption at rest and in transit, and compliance certifications such as SOC 2 and HIPAA. This ensures data security and compliance with regulatory requirements.

**dbt for Data Transformation:**

dbt is a powerful data transformation tool that allows data engineers and analysts to build, test, and deploy data transformation pipelines using SQL. It promotes a modular and version-controlled approach to data transformation, making it easy to manage and collaborate on data workflows.

**Community Support:**

Both Snowflake and dbt have active and supportive communities of users and contributors. This provides access to resources, best practices, and community-developed packages and plugins that can enhance the functionality and usability of the platforms.

In developing the Machine Learning Model for Timesheet and Feedback System, we have chosen python as the primary technology stack. The rationale behind this selection is based on the following considerations:

**Rich Ecosystem of Libraries:**

Python boasts a vast and robust ecosystem of libraries and frameworks specifically tailored for machine learning and data science, such as TensorFlow, PyTorch, scikit-learn, pandas, NumPy, and matplotlib. These libraries provide powerful tools for data manipulation, modelling, visualization, and evaluation, allowing developers to efficiently implement and experiment with ML algorithms.

Based on these considerations, we have chosen the above technology stack as our primary technology stack.

# Web App Components

### Main components of the web app.

* Admin Dashboard
* AdminLogin
* Create User
* Add Project
* Allocate Resources
* EmployeeDashboard
* EmployeeLogin
* Timesheet
* Feedback History
* Project History

### Purpose of each component.

#### AdminDashboard

The Admin Dashboard provides navigation links or buttons for administrators to access the following pages: Create User ,Add projects ,Allocate Resources .These links/buttons enable admins to easily navigate between different sections of the system.

#### AdminLogin

An admin login typically refers to a secure entry point or interface within a system, website, or application that grants administrative privileges and access to certain features or data that regular users / employees do not have permission to access.

#### Create User

The Create User component is utilized to add a new user to a system, typically requiring details such as first name, last name, email, password, and role. This function enables the creation of user accounts, allowing administrators or users with appropriate privileges to add individuals to the system with specified attributes.

#### Add Project

The Add Project component serves the purpose of facilitating the addition of projects to a system or application. It typically requires details such as project name, project type, start date, and end date. This component enables users to input and submit project-related information, allowing for the creation and management of projects within the system.Top of Form

#### Allocate Resources The Allocate Resources component is designed for allocating users to specific projects within a system or application. It typically involves listing out project names, selecting a project, and specifying the users to be allocated, along with their corresponding start and end dates. This component facilitates the management of project resources by allowing administrators or authorized users to assign individuals to projects for defined time periods.

#### Employee Dashboard The Employee Dashboard offers quick access to key functionalities: Timesheet for managing work hours, Feedback History for past input, Project History for project details, and Log Out for secure session termination.

#### Timesheet

The Timesheet feature offers fields for Project Type, Project Name, Task, and daily work hours input, with automatic total hours calculation. It supports adding and deleting rows for flexibility and enables submission upon completion of entries.

#### Feedback History

The Feedback History feature provides users with a comprehensive overview of their past feedback contributions within the system. It displays stored feedback specific to the user, offering insights into their interactions, suggestions, and comments across various projects and activities.

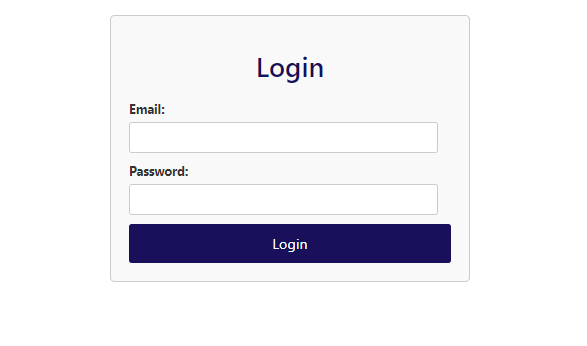
#### Project History

The Project History feature provides users with a detailed summary of their past project engagements within the system. It displays information such as project names, types, durations, and roles, offering insights into their involvement and contributions over time.

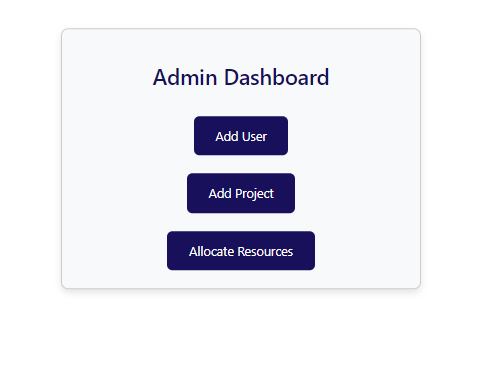
# User Interface Design

### User interface (UI)

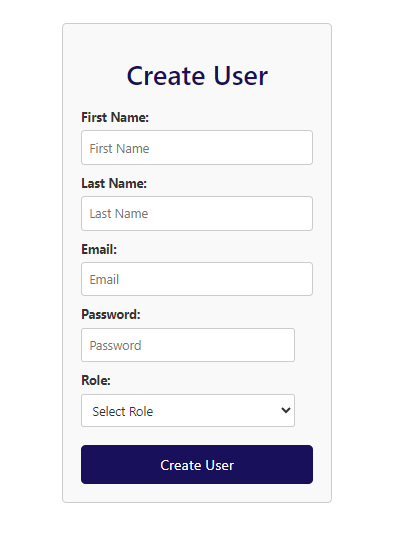
**Login**



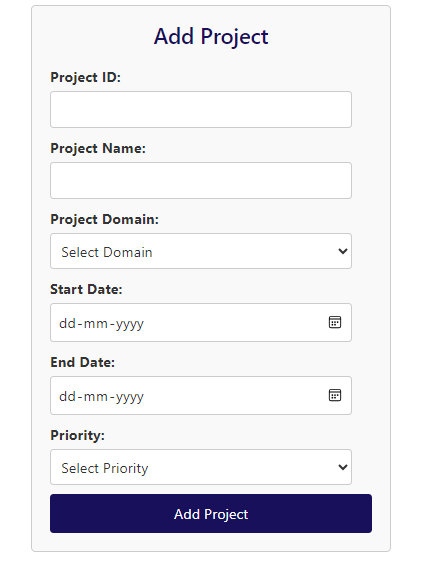
**Admin Dashboard**



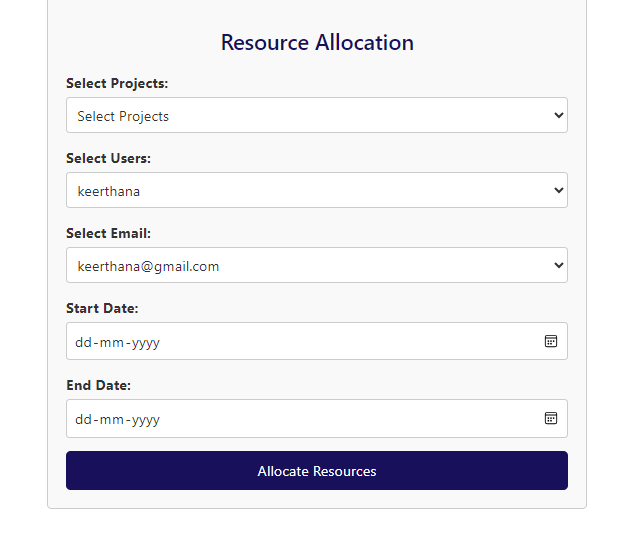
**User Creation**



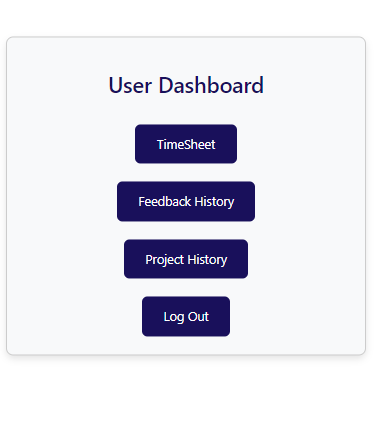
**Add Projects**



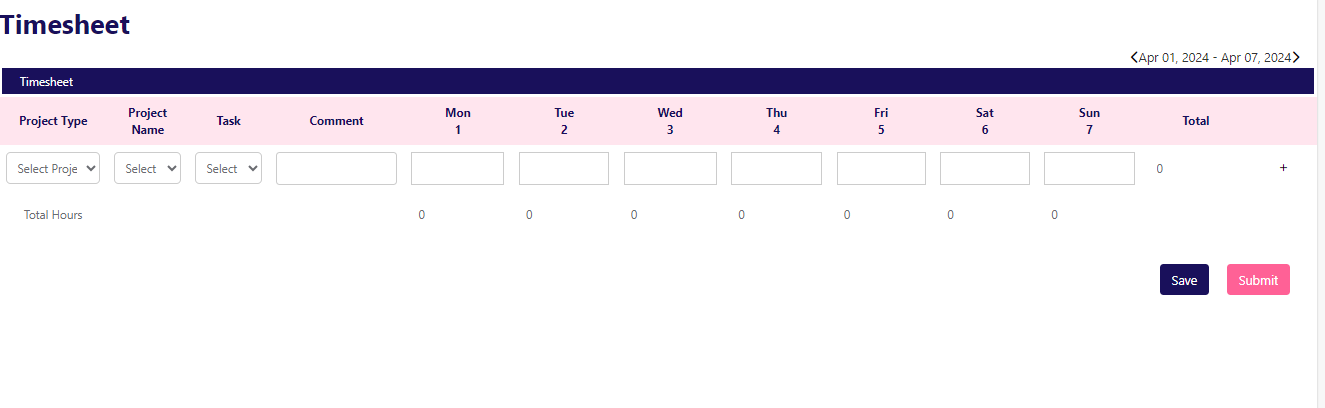
**Allocate Resources**

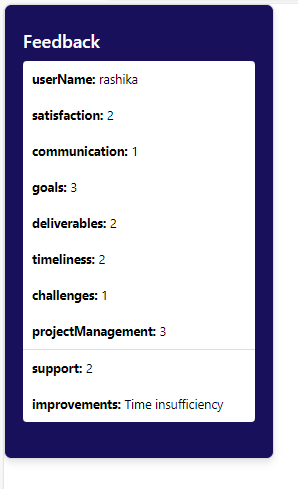


**Employee Dashboard**

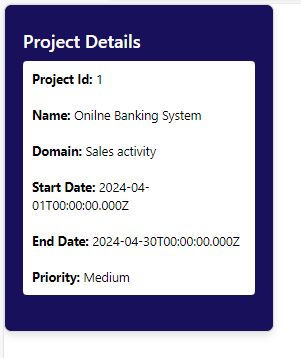


**Timesheet**

**Feedback History**



**Project History**



# Work with Data

Testing and quality assurance are essential processes in software development that aim to ensure the reliability, functionality and ultimately delivering a stable and dependable software product to the end users.

### Approach for the Data Engineering.

**Generating Synthetic Timesheet and Feedback Data**

This initial phase involves generating synthetic data for timesheets and feedback systems using Python scripts and tools like Mockaroo. The created dataset encompasses various time entries, employee details, project information, and feedback records.

**Uploading Data to Snowflake**

Next, the generated timesheet and feedback data are uploaded into Snowflake, a cloud-based data warehousing platform known for its scalability and security features. Snowflake serves as a centralized repository for efficiently storing and managing large volumes of data.

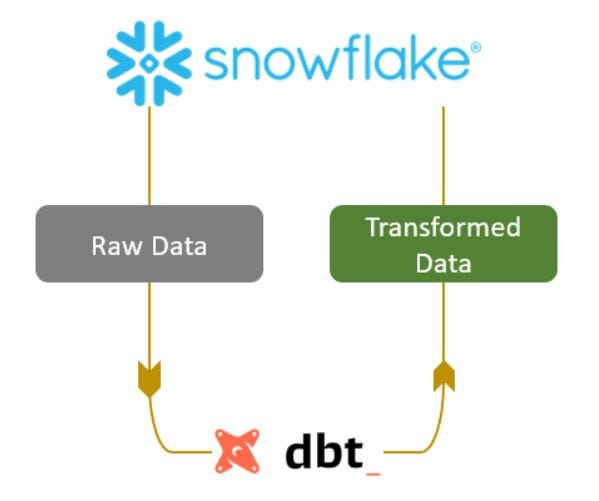
**Data Transformation using dbt**

The uploaded data undergoes transformation processes within dbt (data build tool), an open-source tool for data transformation and analytics engineering. Dbt enables the definition and execution of data pipelines, applying SQL-based transformations to prepare the data for analysis and generate actionable insights.

**Implementing Transformations**

Various transformations are applied to the timesheet and feedback data stored in Snowflake using dbt. These transformations include data cleaning, aggregation, filtering, and other operations aimed at enhancing the quality of the data and extracting meaningful insights.

**Deriving Analytical Insights**

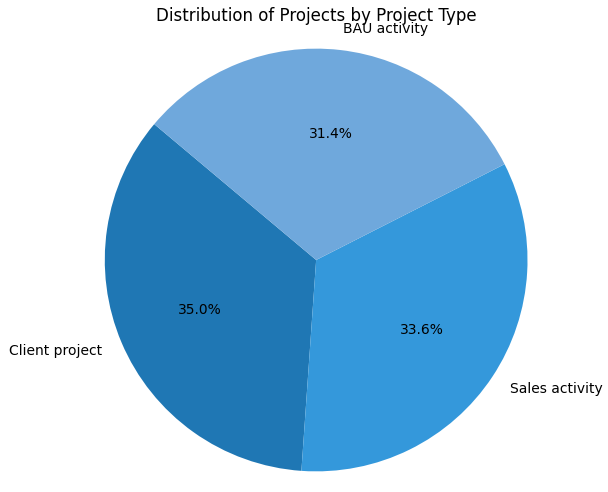
Dbt outputs insights derived from the transformed timesheet and feedback data, providing valuable insights into various aspects of employee performance,project efficiency.These insights aid in decision making processes and drive continuous improvement within the organization's operations.

### Approach for the Machine Learning.

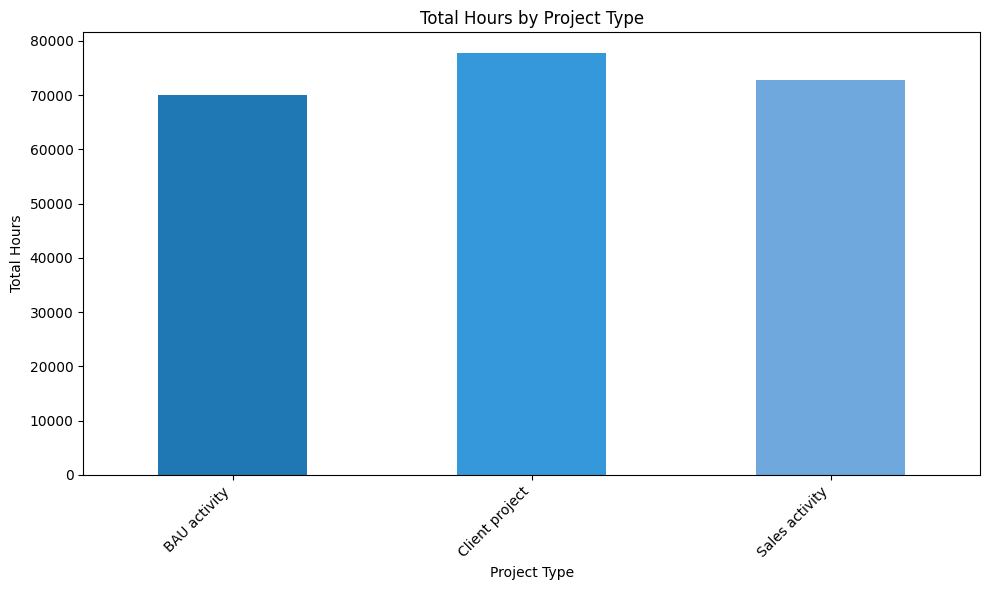
**Exploratory Data Analysis (EDA):**

Exploratory Data Analysis (EDA) is a crucial initial step in any data analysis or machine learning project. It involves exploring and summarizing the main characteristics of a dataset to understand its underlying structure, patterns, and relationships.

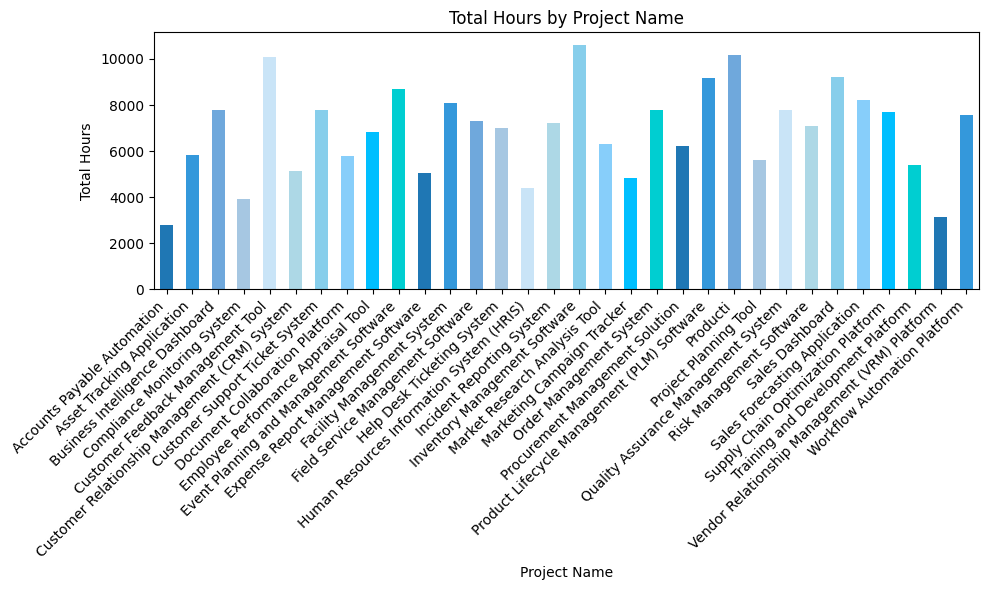
**Distribution of Projects by Project Type**



**Distribution of Total Hours by Project Type**

****

**Distribution of Total Hours by Project Name**

****

**Feature Extraction:**

In the feature engineering phase of the timesheet project, I began by employing label encoding to transform categorical variables into numerical representations. This step is crucial for machine learning algorithms, as they typically operate on numerical data. By encoding categorical variables, I enable the algorithms to process and derive meaningful insights from these features.

Following label encoding, I constructed a correlation matrix. This matrix quantifies the relationship between each pair of variables in the dataset, offering valuable insights into their interdependencies. Understanding correlations helps in identifying redundant or highly correlated features, which can be omitted to reduce model complexity and multicollinearity issues.

Lastly, after analyzing the correlation matrix, I selected the required columns for further analysis or model development. This step involves strategic decision-making based on the project's objectives, domain knowledge, and the insights gained from previous feature engineering steps.

By employing label encoding, correlation analysis, and column selection,I have effectively prepared the dataset for subsequent stages of the project, such as model training and evaluation.

**Model Development:**

Model development involves analysing the preprocessed data from exploratory data analysis and feature extraction to forecast employees likely to have less than 35 hours of attendance in the future. Utilizing machine learning techniques, the goal is to develop accurate predictive models capable of identifying at-risk employees. Employing methods like SVM or GradientBoostingClassifier, historical attendance data and employee attributes are leveraged to generate actionable insights, aiding in resource planning and performance management..

|  |  |
| --- | --- |
| ALGORITHM | ACCURACY |
| SVM | 0.89 |
| MLP | 0.94 |
| Gradient Boosting Classifier | 0.94 |

# Project Resources

### Roles and responsibilities of the project team members.

Keerthana - Involved in developing the full stack application, data engineering and data science

# Risks and Mitigation Strategies

### Potential risks and challenges associated with the project.

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO | Risk / Challenges | Impact | Mitigation Plan |
| 1 | Timesheet Creation | The entire logic will be collapsed if it fails | Need to have the proper logic of the timesheet and its controllers |
| 2 | Database schema creation | Correct schema is required to work with data | Design the schema ahead of building the application |

**Appendix Title**

Document Title