**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. 3](#_Toc163808307)

[1.1.3 Goals of the project: 3](#_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 4](#_Toc163808312)

[3.1.1 Overall architecture of the mobile app. 4](#_Toc163808313)

[3.1.2 Architecture 5](#_Toc163808314)

[3.1.3 Technology stack (e.g., programming languages, frameworks, libraries). 5](#_Toc163808316)

[3.1.4 Rationale behind the chosen technology stack. 6](#_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 13](#_Toc163808323)

[6.1.1 Approach for the Data Engineering. 13](#_Toc163808324)

[6.1.2 Approach for the Machine Learning. 14](#_Toc163808325)

[7 Project Resources 16](#_Toc163808326)

[7.1.1 Roles and responsibilities of the project team members. 16](#_Toc163808327)

[8 Risks and Mitigation Strategies 16](#_Toc163808328)

[8.1.1 Potential risks and challenges associated with the project. 16](#_Toc163808329)

# Project Overview

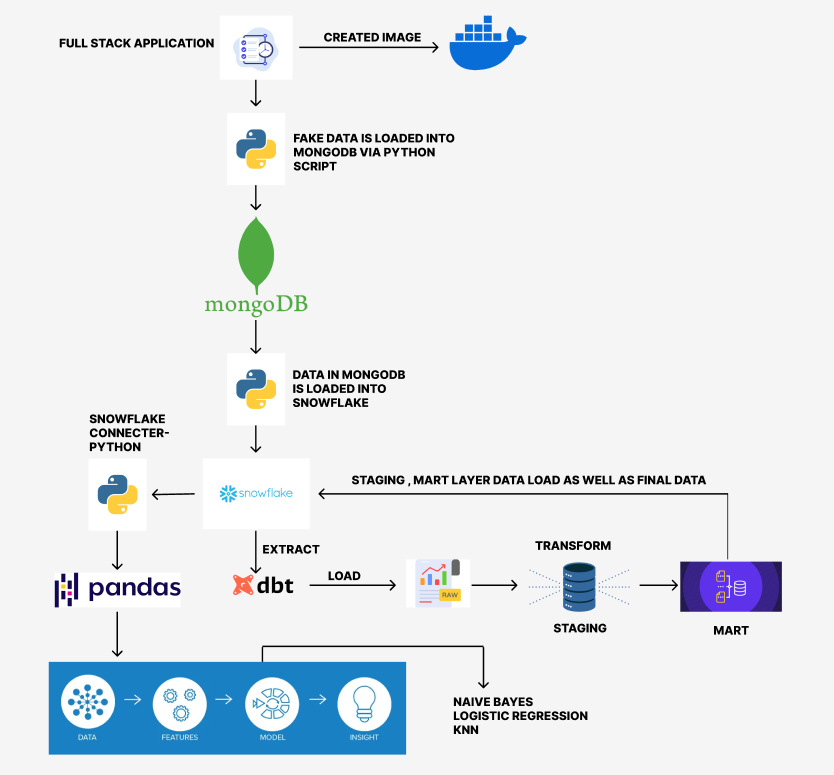
### Brief introduction of the project.

Timesheet and Feedback System is a web-based application ,designed to streamline two crucial processes for employees within an organization: timesheet creation and feedback submission. Both tasks will be mandatory, ensuring that employees fulfill these responsibilities consistently.

The timesheet creation component aims to simplify the process of logging hours worked by employees. This includes capturing details such as the days worked, total hours worked each day, and any additional project-related notes or tasks accomplished during that time. By making timesheet creation mandatory, the project ensures accurate tracking of employee work hours, which is essential for payroll processing, project management, and compliance purposes.

On the other hand, the feedback submission component focuses on gathering feedback from employees regarding their work experiences, challenges faced, and suggestions for improvement. The feedback questions will be tailored based on each employee's role within the organization and the specific projects they are currently working on. This personalized approach ensures that the feedback collected is relevant, actionable, and valuable for both employees and the organization as a whole.

**Overall Workflow**



### Purpose of the project.

The purpose of this project is to Simplifying Employee Work Processes within an organization: timesheet creation and feedback submission. The project has several key objectives:

**Project Management Integration:**

Integrating timesheet data with project management tools for better resource allocation and project tracking.

Enabling managers to monitor project progress based on actual work hours logged by employees.

**Insightful Feedback Collection:**

Tailoring feedback questions based on employees' roles and projects enables the organization to gather meaningful insights into employee experiences, challenges, and suggestions.

This feedback is valuable for identifying areas of improvement, enhancing employee engagement, and fostering a positive work culture.

### Goals of the project:

Design of the UI/UX components for Timesheet and Feedback with a responsive manner as per specification developed during requirement analysis phase.

Ensure the platform is intuitive and easy to navigate for both administrators and employees.

· Design of the Core application tier including:

o Front End tier for the end-user, accessed through web.

o Back End API for creating timesheet and feedbacks and to allocate projects.

# 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 Creation and Authentication:**

Admin is allowed to create employees.

After the employee creation default password is sent to the employees.

They can change the password using the link provided in the email.

**Personalized Admin Dashboard**:

Admin Specific dashboard for Employee creation add the new projects, allocate the projects to the

employees

Power BI integrated dashboard for visualization.

**Personalized Employee Dashboard**:

Employee Specific Dashboard for timesheet filling and also for feedback.

**Responsive Design:**

The Responsive UI is done with Prime React and Some features with custom css

# Architecture and Technology Stack

### Overall architecture of the web application.

Develop a timesheet and feedback system providing secure authentication, timesheet submission, feedback provision, email alerts, and workload oversight. Establish a resilient data platform for reporting and analysis, boosting productivity and employee involvement.

**Front end Application:**

**User Authentication** :

The User Authentication offers a secure login page for the employees which has email and password authentication

**Timesheet Module** :

After the employee login , the employee is displayed with a dashboard with has the timesheet button ,by clicking the button the employee is redirected to the timesheet Page with has date based timesheets

**Feedback Module** :

If the employee filles the timesheet for the specified week he/she will be redirected to the feedback page for that particular week and the feedback questions are displayed based on their role like software engineer, consultant etc ..

**Back end Services:**

**Authentication and Authorization** :

The Login Api is used for authentication and is done through JWT in which the access token is send to

the frontend and authorization is done , by showing only the resources that they can access.

**Timesheet API / Route** :

It allowes the user for creating , updating timesheets based on the projects that the employee is

assigned.

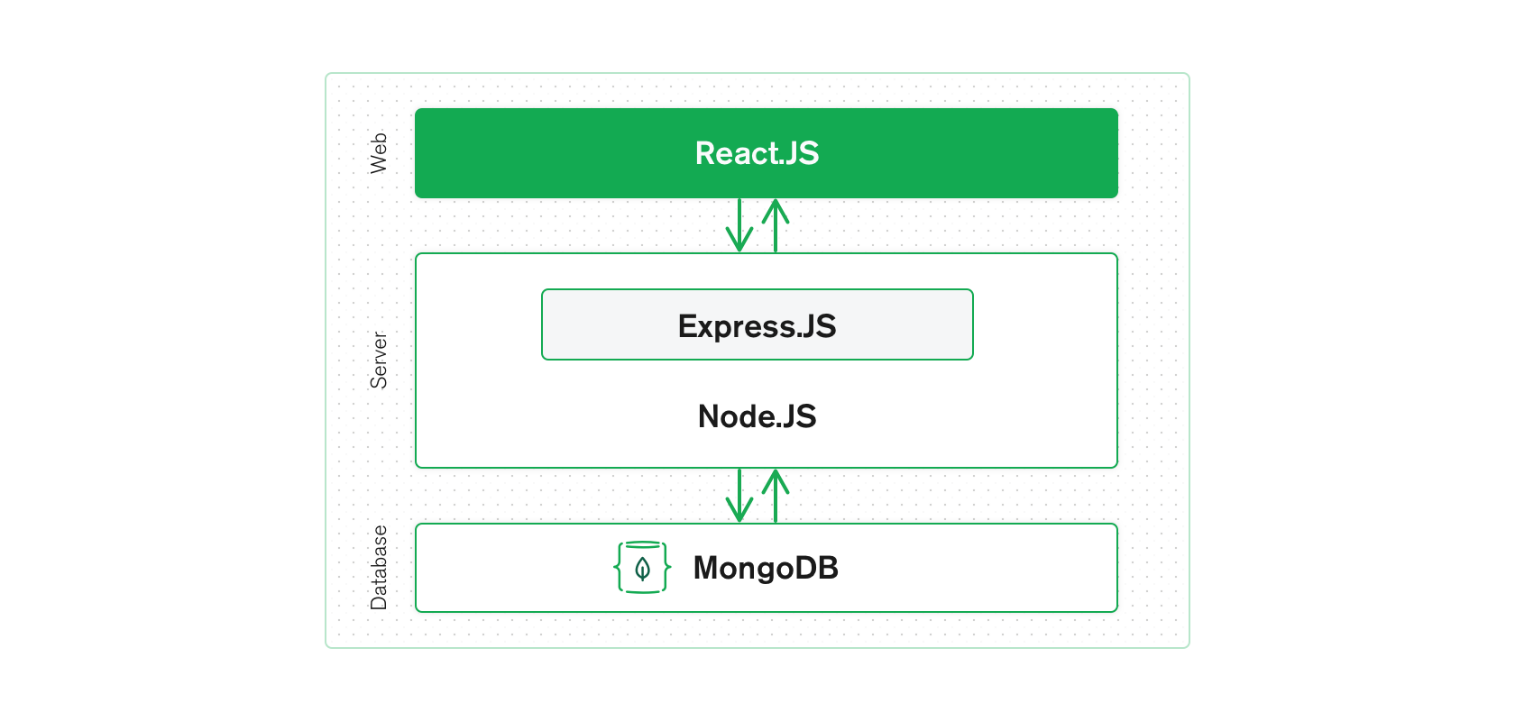
**Feedback API / Route** :

Manages feedback submissions from employees after filling the timesheet for the particular week.

**Email Notification** :

* If the employee is created by the admin , the email is sent to the employee with the default password and is triggered to change the password.
* If the employee is not filled the feedback for a week , he/she will be notified that your feedback for the specific timesheet is not filled.

### Architecture



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

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

The technology stack for the web 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.

**Prime React:**

PrimeReact is a rich set of UI components for React applications. It offers a wide range of ready-to-use components like buttons, tables, forms, and charts. PrimeReact is built on top of the popular PrimeFaces library, providing responsive and customizable components. It simplifies front-end development by offering feature-rich UI components out of the box.

**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.

**Docker**:

Docker simplifies application deployment by packaging code and dependencies into containers, ensuring consistency and portability across different environments. It provides tools for building, managing, and running containers efficiently, enhancing development workflow and resource utilization.

### 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 / User Login
* Forgot Password
* Reset / change Password
* User Creation
* Create Project
* Project Allocation
* Dashboard
* Timesheet
* Assigned Projects
* Feedback
* Logout

### Purpose of each component.

#### Login

The form has the fields of email and password , it configures the email to segregate the email is employee or admin and redirect according to the user type.

#### Forgot Password

If the employee or admin forgots the password , he/she will be redirected to the change password page via the email is sent to them to change the password

#### Reset Password

The employee is created by the admin , the email is sent to the employee to reset the password

#### Add Project

The Add Project module facilitates the inclusion of new projects into a system or application. It typically collects essential project details like project title, type, start and end dates, empowering users to input and submit project-specific information, thereby supporting project creation and management within the system.

#### Allocate Projects The Project Allocation feature is intended for assigning users to particular projects within a system or application. It typically entails displaying project names, choosing a project, and defining the users to be allocated, alongside their respective start and end dates.

#### Timesheet

#### The Timesheet module provides input fields for Project Category, Project Title, Task Description, and daily hours worked, automatically calculating total hours. It allows for adding and removing rows as needed, offering flexibility, and enables submission after completing entries.

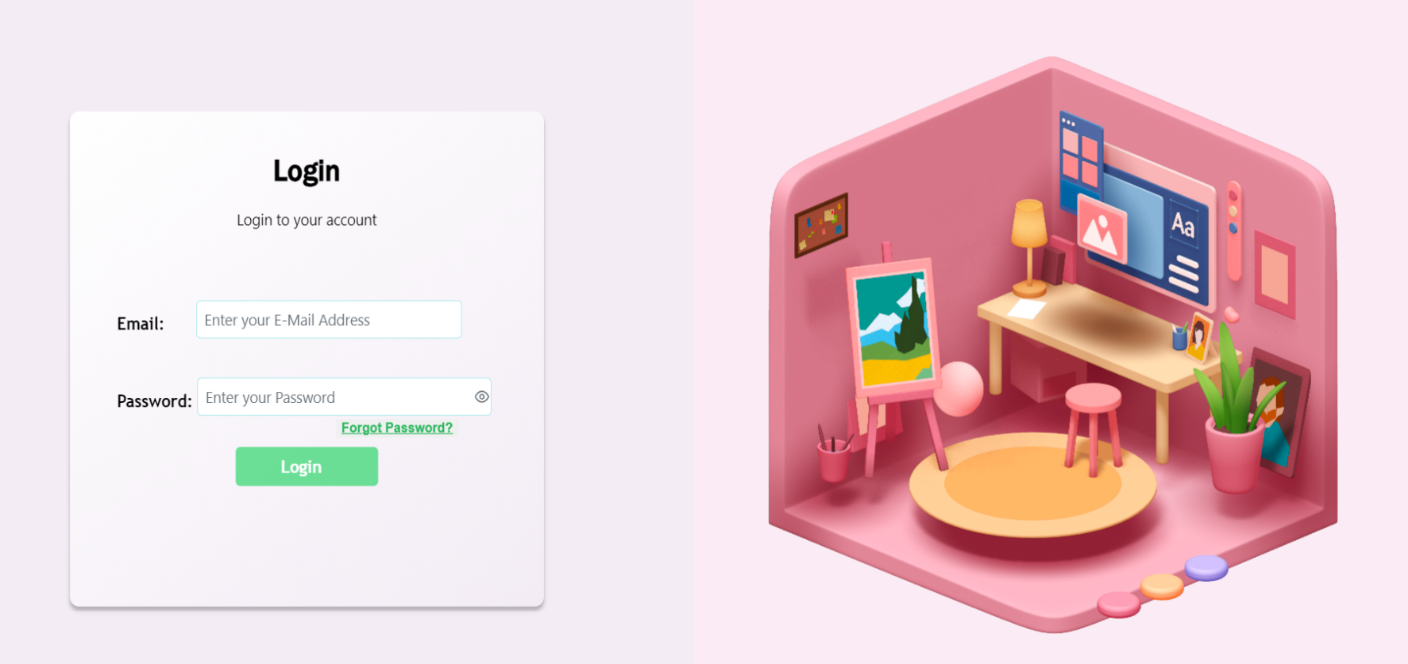
#### Feedback

Feedback is a valuable mechanism for gathering insights and opinions from users, helping to identify areas of improvement and enhance user experience. It provides a platform for users to share their thoughts, suggestions, and feedback, fostering continuous improvement and collaboration within the system or application.

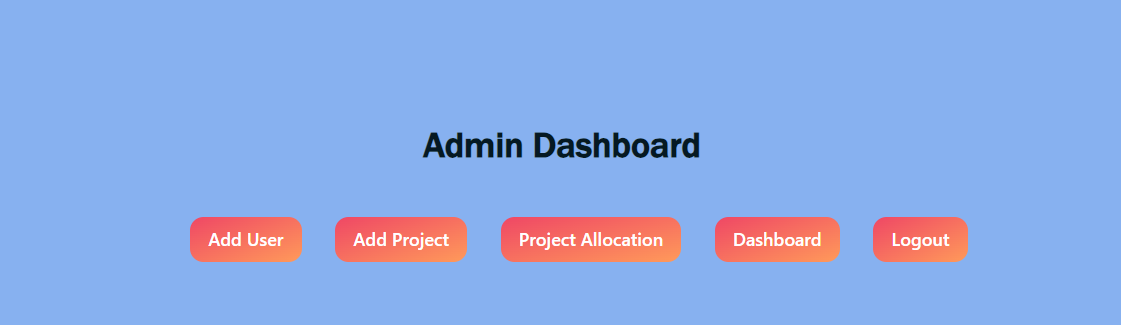
# User Interface Design

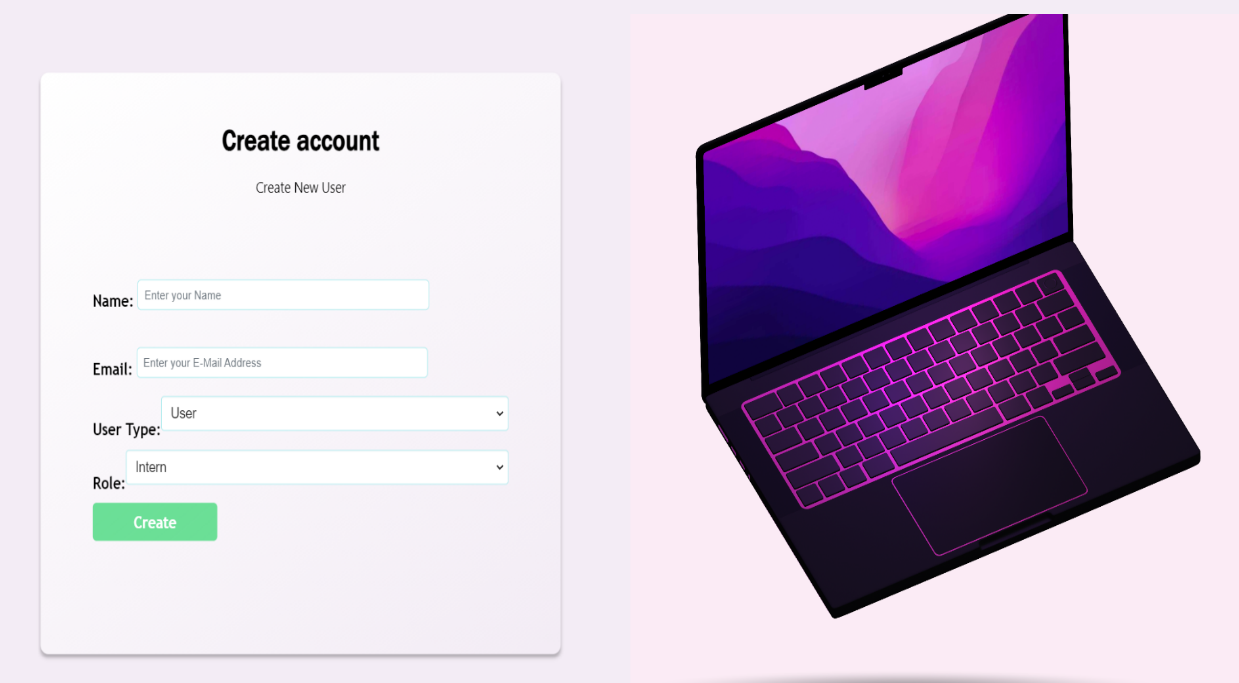
### User interface (UI)

**Login**

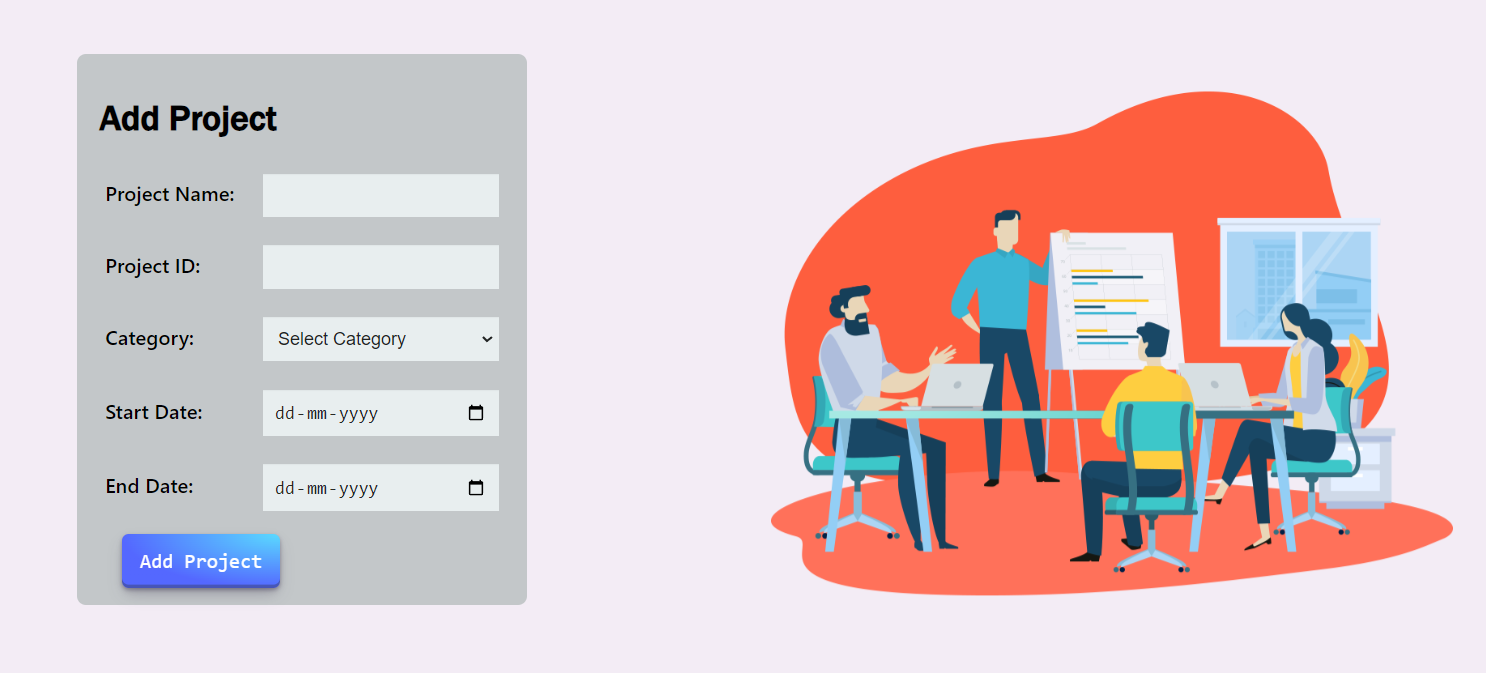


**Admin Dashboard**

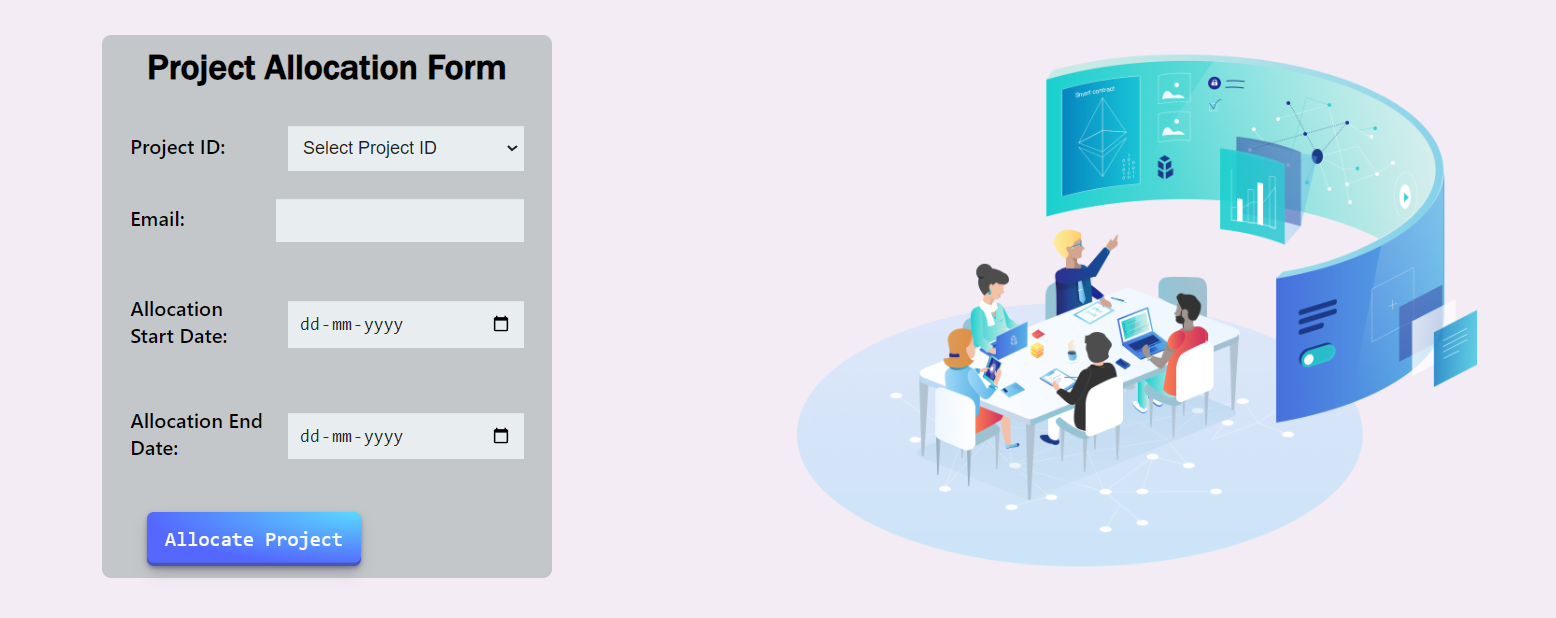


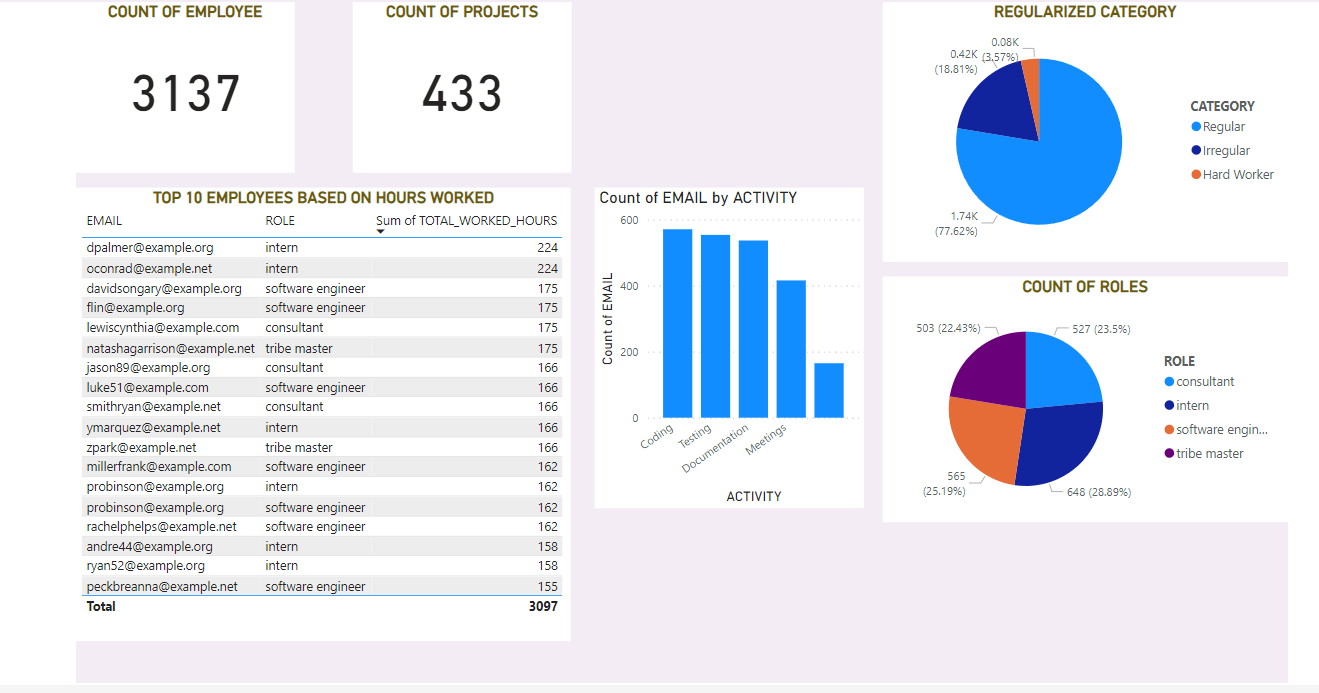
**User Creation**

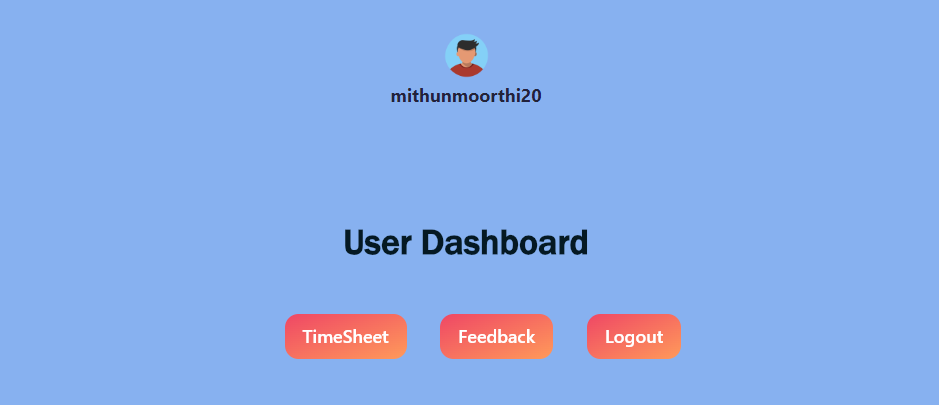
**Add Projects**



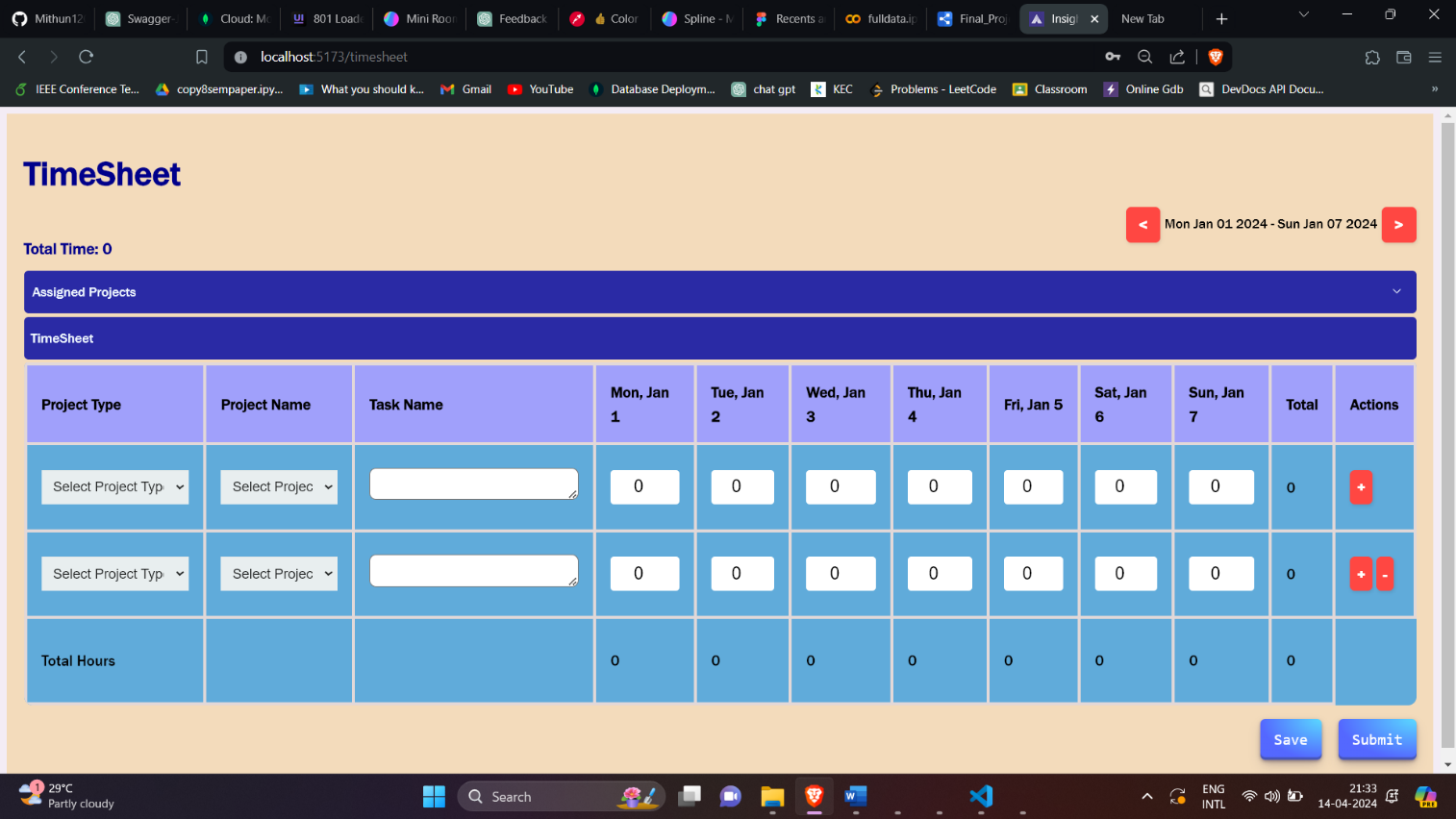
**Allocate Projects**

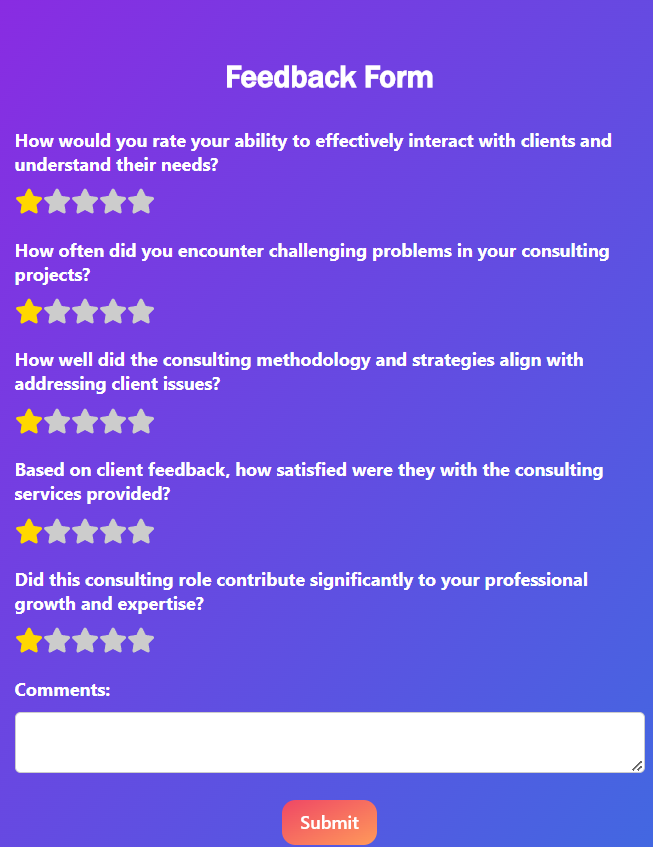


**PowerBI Dashboard**

**User DashBoard**

**Timesheet**

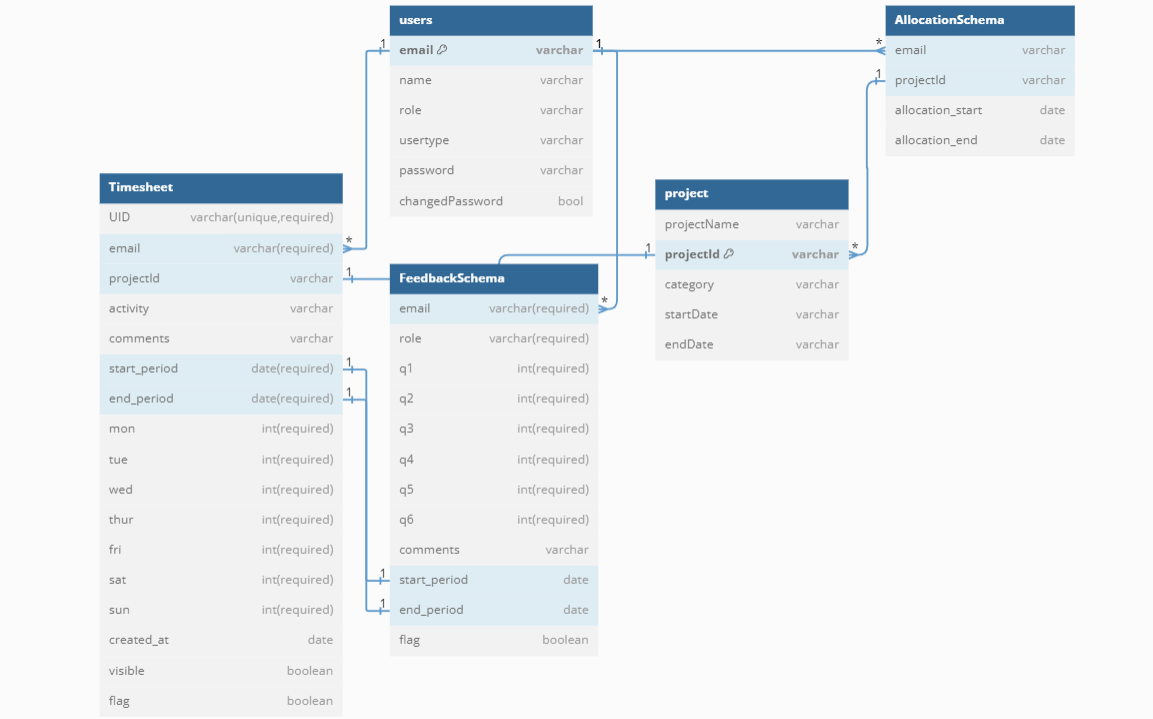


**Feedback**

# 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.

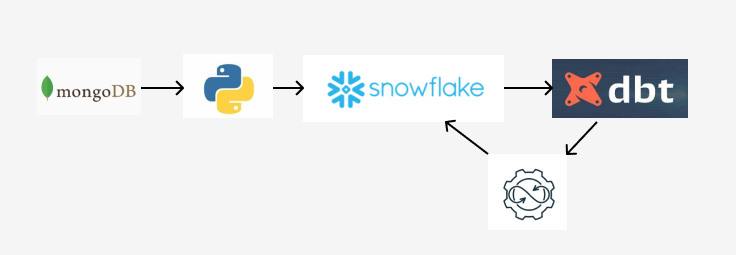
**Data Modeling**



### Approach for the Data Engineering.

I have implemented a Python script to facilitate data migration from MongoDB to Snowflake. Utilizing PyMongo, we established a connection to our MongoDB database, fetched the relevant data, and performed necessary transformations to align with the Snowflake schema. Leveraging the Snowflake Connector for Python, we established a connection to our Snowflake database.

For Snowflake integration, create a stage pointing to the location where MongoDB Atlas Data Lake exports data. Configure an external table in Snowflake referencing the files exported by MongoDB Atlas Data Lake. Utilize Snowflake's COPY INTO command to load data from the external table into Snowflake tables for further processing.

DBT integration streamlines and standardizes data mart transformation and preparation in Snowflake, automating the process and ensuring reliable analytics.

### Approach for the Machine Learning.

**Data Exploration:**

Data exploration is an essential initial phase in any data analysis or machine learning project. It encompasses examining and summarizing the primary characteristics of a dataset to comprehend its underlying structure, trends, and associations.

**Feature Engineering:**

Feature engineering refers to the process of converting raw data into a set of relevant features that can serve as input for machine learning algorithms.

**Model Building:**

Model building involves analyzing preprocessed data from data exploration and feature engineering to predict employees likely to have attendance less than 35 hours in the future. Using machine learning techniques, the objective is to create accurate predictive models capable of identifying employees at risk. Techniques such as logistic regression or decision trees leverage historical attendance data and employee attributes to provide actionable insights, assisting in resource allocation and performance monitoring.

**NAIVE BAYES** :

Naive Bayes is a probabilistic classifier based on Bayes' theorem and assumes independence between features.It calculates the probability of each class given the input features and selects the class with the highest probability as the prediction.It's commonly used for text classification tasks like spam detection and sentiment analysis due to its simplicity and efficiency.

**KNN :**

KNN is a non-parametric and lazy learning algorithm used for classification and regression tasks.It classifies a data point based on the majority class among its k-nearest neighbors in the feature space.It's effective for nonlinear data patterns and does not make strong assumptions about the underlying data distribution.

**LOGISTIC REGRESSION :**

Logistic Regression is a binary classification algorithm that estimates the probability of a binary outcome based on input features.It models the relationship between the dependent variable and one or more independent variables by fitting a logistic curve.It's suitable for problems where the dependent variable is binary, such as predicting whether an email is spam or not.

**RANDOM FOREST :**

A Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the mode of the classes (classification) or the average prediction (regression) of the individual trees. It combines the concepts of bagging and feature randomness to create a diverse set of trees that collectively provide more accurate and stable predictions compared to a single decision tree.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MODELS | ACCURACY | F1 – SCORE | RECALL | PRECISION |
| **NAIVE BAYES** | **0.86** | **0.89** | **0.99** | **0.86** |
| **KNN** | **0.87** | **0.93** | **0.98** | **0.88** |
| **LOGISTIC REGRESSION** | **0.90** | **0.69** | **0.52** | **0.91** |
| **RANDOM FOREST** | **0.87** | **0.93** | **0.95** | **0.87** |

# Project Resources

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

Mithun M - 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 | If there is a delay in API | Project deliverables will be delayed | Need to have all the API ahead of development team |
| 2 | Database schema creation | Correct schema is required to work with data | Design the schema ahead of building the application |

**Appendix Title**

Document Title