**TIMESHEET AND FEEBACK SYSTEM**

Final Project – Ashmitha Laxmi

Contents

[1 Project Overview 2](#_Toc137748514)

[1.1.1 Brief introduction of the project. 2](#_Toc137748515)

[1.1.2 Purpose of the project. 2](#_Toc137748516)

[1.1.3 Goals of the project: 2](#_Toc137748517)

[2 Project Scope 3](#_Toc137748518)

[2.1.1 Key features and functionalities of the website. 3](#_Toc137748520)

[3 Architecture and Technology Stack 3](#_Toc137748521)

[3.1.1 Overall architecture of the website. 3](#_Toc137748522)

[3.1.2 Architecture 4](#_Toc137748523)

[3.1.3 Technology stack (e.g., programming languages, frameworks, libraries). 4](#_Toc137748524)

[3.1.4 Rationale behind the chosen technology stack. 5](#_Toc137748525)

[4 Web App Components 6](#_Toc137748526)

[4.1.1 Main components of the website. 6](#_Toc137748527)

[4.1.2 Purpose of each component. 6](#_Toc137748528)

[5 User Interface Design 8](#_Toc137748529)

[5.1.1 User interface (UI) design approach. 8](#_Toc137748530)

[6 Testing and Quality Assurance 12](#_Toc137748531)

[6.1.1 Testing approach for the website 12](#_Toc137748532)

[6.1.2 Types of testing to be performed. 13](#_Toc137748533)

[6.1.3 Quality assurance processes and tools to ensure app reliability. 14](#_Toc137748534)

[7 Project Timeline and Resources 15](#_Toc137748535)

[7.1.1 Estimated project timeline, including major milestones. 15](#_Toc137748536)

# Project Overview

### Brief introduction of the project.

### The Timesheet and Feedback System streamlines timesheet creation and feedback submission for employees. With user-friendly interfaces, it simplifies work hour reporting and task tracking. Customized feedback questions based on roles and projects enhances engagement. Automated reminders ensure timely submissions. Role-based access controls and predictive analytics for attendance regularization further optimize organizational efficiency.

### Purpose of the project.

The purpose of the Timesheet and Feedback System is to improve efficiency and accountability within the organization by automating the process of timesheet creation and feedback submission. The key goals of the project include:

* + - * Accountability Improvement: By making timesheet and feedback submissions mandatory, the project promotes accountability among employees regarding their work hours and project contributions.
      * Feedback Facilitation: Providing a structured platform for feedback enables employees to share insights and suggestions, fostering a culture of continuous improvement and innovation.
      * Role-Based Access Control: Customizing access to features based on employee roles ensures that each user has access to relevant functionalities, enhancing security and data integrity.
      * Project-Specific Customization: Tailoring feedback questions based on project involvement helps gather relevant insights and improve project outcomes by addressing specific challenges and requirements.

### Goals of the project:

* Efficiency: Streamline administrative tasks related to timesheet creation and feedback submission, reducing time and effort for employees and administrators alike.
* Accountability: Ensure that all employees are accountable for accurately reporting their work hours and providing constructive feedback, promoting transparency and responsibility.
* Productivity: Improve overall productivity by automating repetitive tasks, allowing employees to focus more on their core duties and responsibilities.
* Quality Feedback: Establish a structured framework for collecting feedback, enabling employees to offer valuable insights and suggestions for organizational improvement.
* Customization and Personalization: Tailor the system to meet the specific needs of different employee roles and project requirements, enhancing relevance and effectiveness.
* Timeliness: Implement automated reminders to prompt employees to complete timesheets and feedback submissions on time, ensuring timely data collection and processing.

These goals collectively aim to enhance organizational efficiency, foster a culture of accountability and continuous improvement, and ultimately contribute to the overall success of the organization.

# Project Scope

### Key features and functionalities of the website.

**User Registration and Authentication:**

User-friendly registration process for patients to create their accounts. Secure authentication mechanisms, including username/password.

**Timesheet Creation:**

Employees can easily create and submit timesheets, detailing their work hours and tasks completed.

**Feedback Question Creation:**

Administrators can create feedback questions tailored to employees' roles and project involvement.

**Mandatory Submissions:**

Both timesheet and feedback submissions are mandatory, ensuring accountability and completeness.

**Role-Based Access Control:**

Access to features is controlled based on employees' roles, ensuring appropriate permissions and security.

**Project-Specific Feedback:**

Feedback questions are customized based on the project an employee is working on, enhancing relevance.

**Seamless Integration:**

Integration with external APIs for accessing information about user, project timesheets and related resources.

# Architecture and Technology Stack

### Overall architecture of the web app.

The Timesheet and Feedback System utilizes a comprehensive architecture to ensure efficiency, security, and usability. The architecture comprises the following components:

1. Client-Side Interface:

The user-facing interface enables employees to interact with timesheet creation, feedback submission, and other features seamlessly. It offers a responsive and intuitive design, supporting various devices and screen sizes for a consistent user experience.

2. Backend Infrastructure:

The backend infrastructure handles server-side logic, database management, and authentication processes. It manages data storage, user authentication, and business logic, ensuring secure and efficient operations.

3. Database Management System:

Stores essential data such as user information, timesheets, feedback questions, and project details. Ensures data integrity, accessibility, and efficient retrieval for seamless functionality.

4. Authentication and Authorization System:

Manages user authentication and authorization processes, ensuring secure access to features based on roles and permissions. Enhances security and controls access to sensitive information within the system.

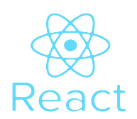
5. Notification Mechanism:

Sends automated reminders to employees for pending timesheets and feedback submissions. Improves user engagement and compliance by prompting timely actions.

6. Integration Layer:

Facilitates seamless communication between frontend, backend, and external systems. Ensures smooth data flow and interoperability, enhancing overall system efficiency and functionality.

### Architecture



**Backend**

**Frontend**



**Database**

API interceptor – Axios

Nodemailer – Mail management



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

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

The technology stack for the Web Application Platform for Timesheet and Feedback system will include the following components:

**Prime React**:

PrimeReact is a comprehensive UI component library for React applications, offering a wide range of customizable components and themes. It provides a rich set of features out of the box, such as data visualization, form components, and layout utilities, enabling developers to build modern and responsive user interfaces efficiently.

**Express:**

Express.js is known for its minimalist and flexible design, making it an ideal choice for building APIs and web servers. It provides a wide range of features and middleware to handle various tasks efficiently, allowing developers to create robust and scalable applications.

**Mongo DB:**

MongoDB is a popular NoSQL database known for its flexibility and scalability. It stores data in a JSON-like format, making it suitable for handling unstructured or semi-structured data. MongoDB's distributed architecture enables horizontal scaling, making it well-suited for applications with large datasets and high traffic.

**Axios:**

Axios is a widely used JavaScript library for making HTTP requests from the application to the server. It simplifies the process of sending and receiving data, handling API calls, and managing network requests in the web app.

**JSON WebTokens:**

JSON Web Token (jsonwebtoken) is a widely used standard for securely transmitting information between parties as a JSON object. It is commonly used for authentication and authorization in web applications, providing a stateless, secure, and efficient mechanism for user authentication.

### Rationale behind the chosen technology stack.

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

**Cross-Platform Compatibility:**

React Prime allows us to build a single codebase that can run on both platforms. This cross-platform compatibility significantly reduces development time and effort, as we can leverage a shared codebase to target multiple devices and operating systems. It ensures consistent user experience and functionality across different platforms, minimizing the need for platform-specific development.

**Time and Cost Efficiency:**

By utilizing React Prime, we can optimize development resources and streamline the development process. The ability to write code once and deploy it across multiple platforms results in time and cost savings. This efficiency enables faster time-to-market, ensuring the application reaches patients sooner.

**Large Developer Community and Ecosystem:**

React 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:**

It’s 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.

**Flexibility and Customizability:**

It offers flexibility in terms of customization and integration with native modules. It allows us to leverage platform-specific functionalities by incorporating native modules when necessary. This flexibility ensures that we can meet specific requirements, integrate with existing systems, and provide a tailored instrument tracking solution that aligns with your unique needs.

Based on these considerations, React Prime emerges as the optimal technology stack for developing the Web Application Platform. It enables us to deliver a high-quality, cross-platform application performance, code reusability, and a vibrant developer community, while ensuring time and cost efficiency in the development process.

# Web App Components

### Main components of the application.

* body-parser
* cors
* express
* jsonwebtoken
* mongoose
* nodemailer
* nodemon
* axios
* primereact
* react-datepicker
* react-router-dom

### Purpose of each component.

**Body-parser:**

Parses incoming request bodies in a middleware before handlers, simplifying data extraction.

**cors:**

Enables cross-origin resource sharing, allowing restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served.

**express:**

Web application framework for Node.js, providing robust features for building web applications and APIs.

**jsonwebtoken:**

Used for creating JSON Web Tokens (JWTs), facilitating secure authentication and authorization of users in web applications.

**mongoose:**

MongoDB object modeling tool designed to work in an asynchronous environment, providing a straightforward schema-based solution for modeling application data.

**nodemailer:**

Library for sending emails from Node.js applications, supporting HTML content, attachments, and more.

**nodemon:**

Utility that automatically restarts the Node.js application when file changes in the directory are detected, facilitating development by eliminating the need to manually restart the server.

**axios:**

Promise-based HTTP client for the browser and Node.js, providing an easy-to-use interface for making HTTP requests.

**Primereact**:

UI component library for React applications, offering a rich set of components for building modern and responsive user interfaces.

**react-datepicker:**

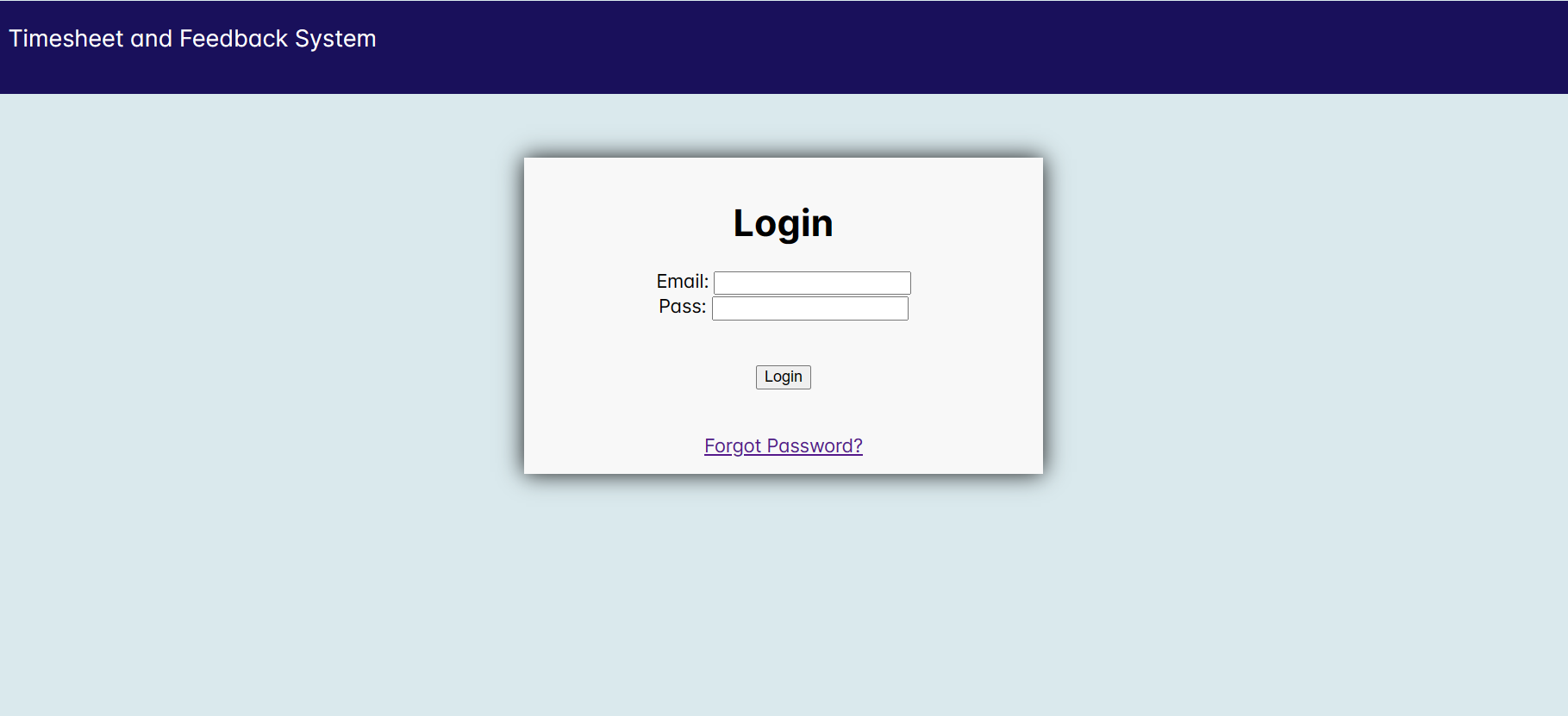
Flexible and customizable datepicker component for React applications, allowing users to select dates and times easily.

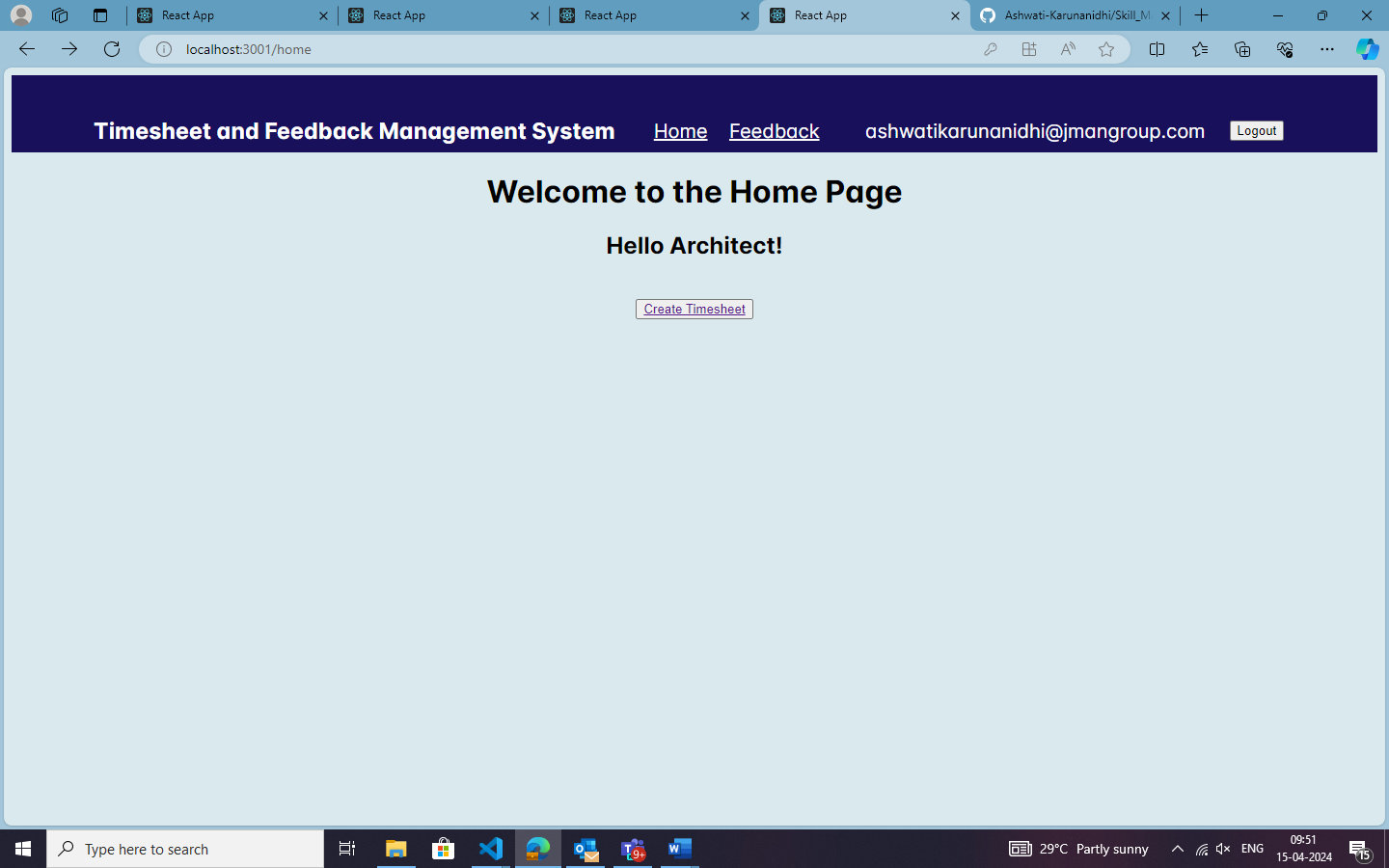
**react-router-dom:**

Routing library for React applications, enabling navigation between different components in response to browser URL changes.

# User Interface Design

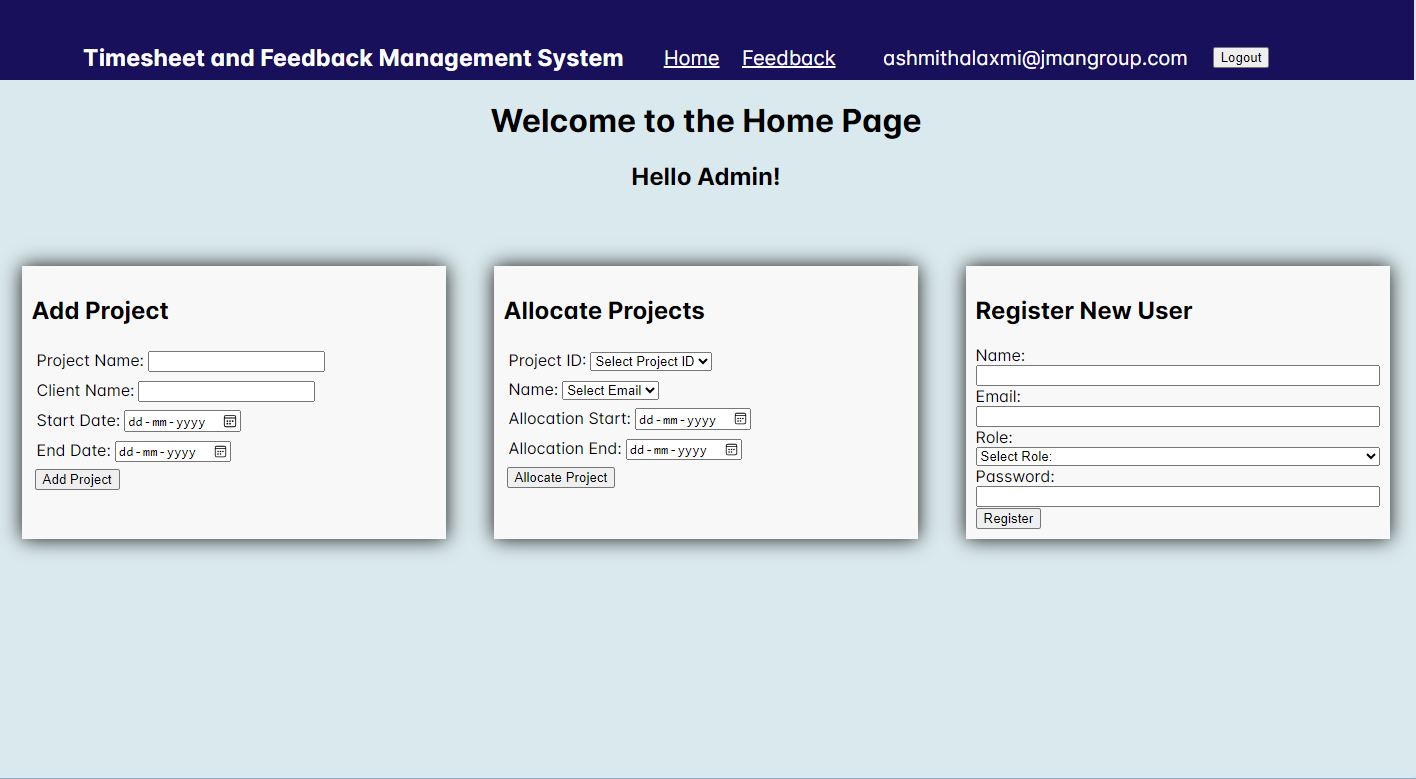
### User interface (UI) design approach.

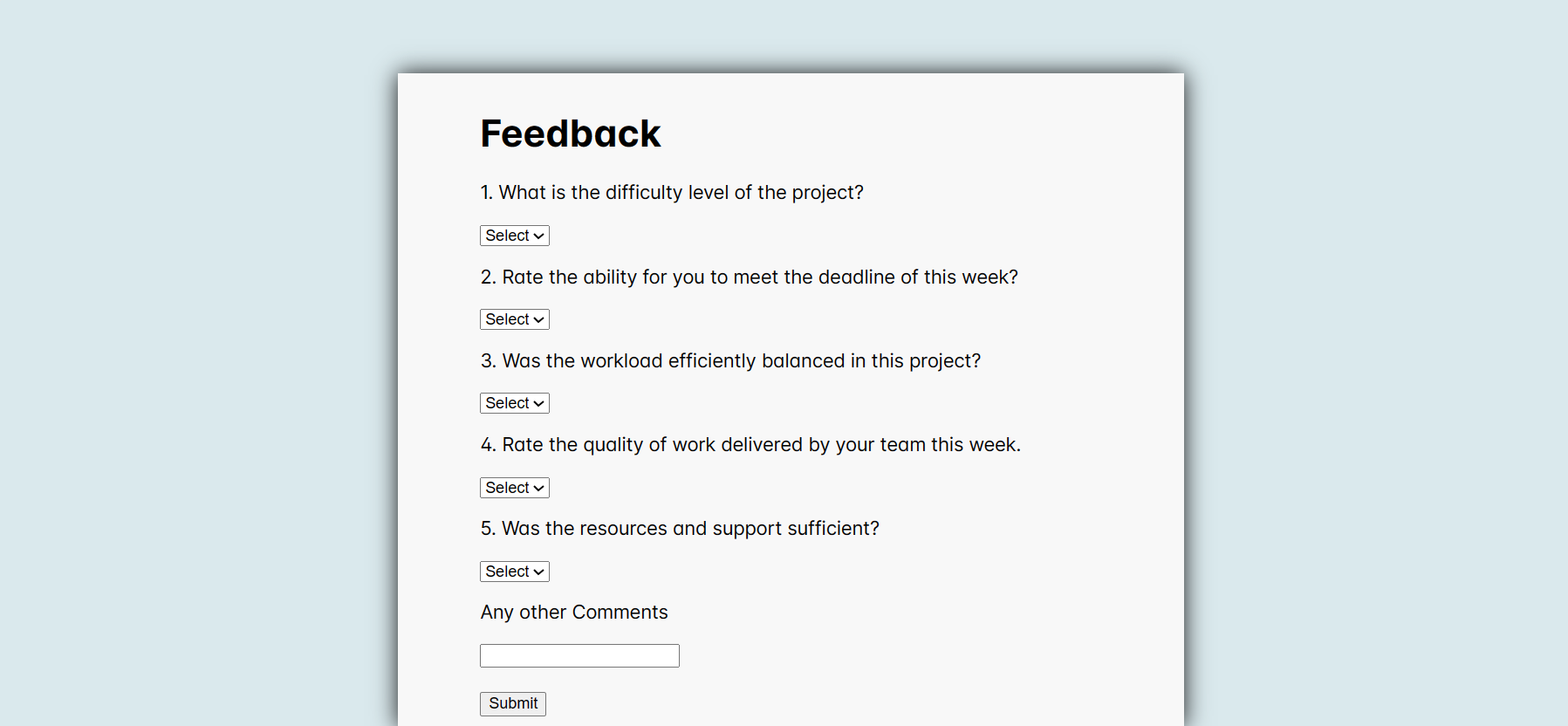




A screenshot of a computer

Description automatically generated





# Testing and Quality Assurance

Testing and quality assurance are essential processes in software development that aim to ensure the reliability, functionality, and overall quality of a software product. These processes involve systematically examining and validating the software at various stages to identify defects, bugs, and any deviations from the expected behaviour. By conducting thorough testing and quality assurance, developers can uncover and rectify issues early in the development cycle, enhancing user satisfaction, minimizing risks, and ultimately delivering a stable and dependable software product to the end users.

### Testing approach for the app.

1. Requirement Analysis:

Understand the requirements of the application thoroughly, including its intended functionality, target audience, supported platforms and any specific device requirements.

1. Test Planning:

Develop a comprehensive test plan that outlines the testing objectives, scope, test environments, test cases, and testing techniques to be used. Determine the types of testing to be performed.

1. Test Environment Setup:

Prepare the necessary test environments, including physical devices, emulators, simulators, or cloud-based testing platforms. Ensure that the test environments closely match the actual user environment.

1. Functional Testing:

Verify that the web application meets the specified functional requirements. Test various features and functionalities of the application, such as user interface, navigation, user input validation, data processing, and integration with backend services.

1. User Interface Testing:

Test the application's user interface (UI) to ensure consistency, responsiveness, and adherence to platform-specific design guidelines. This involves checking for proper alignment of elements, accurate rendering of fonts and images, correct color schemes, and intuitive navigation across different devices and screen sizes.

### Types of testing to be performed.

Functional Testing:

* Verify that each function in the application behaves as expected.
* Ensure that all requirements are met and that all functionalities work properly.

Usability Testing:

* Assess user satisfaction, navigation, intuitiveness, and learnability of the application.
* Identify areas where users may face difficulties or frustrations.
* Evaluate error frequency and severity to enhance user experience.

System Testing:

* Evaluate the entire system's compliance against specified requirements.
* Perform end-to-end testing to ensure proper functionality across the application.

Retesting:

* Re-execute previously failed tests against new software to verify if issues are resolved.
* Verify that problems identified in earlier testing phases have been adequately addressed.

Regression Testing:

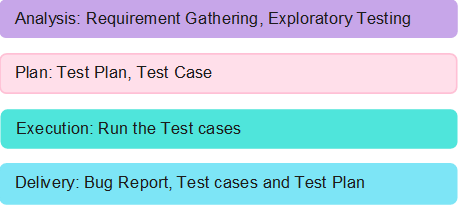
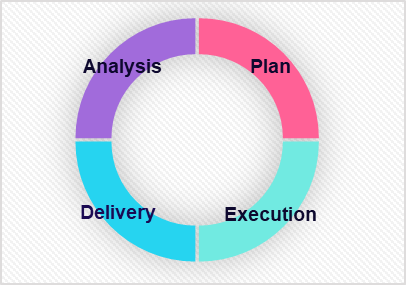
* Verify that existing features continue to function correctly after changes or additions to the application.
* Ensure that fixes do not adversely impact the existing functionality.

Compatibility Testing:

* Check if the application functions correctly on different devices, operating systems, and screen sizes.
* Identify and resolve compatibility issues early to minimize user complaints and negative reviews.
* Ensure a seamless user experience across various platforms to enhance customer satisfaction and market acceptance.

These types of testing ensure that the Timesheet and Feedback system application meets quality standards, functions properly across various scenarios, and provides an optimal user experience for its intended audience.

### Quality assurance processes and tools to ensure app reliability.

The objective of the test is to define the goals and purpose of the testing effort. It aims to provide a comprehensive and focused statement of what is to be accomplished through testing. The test objectives serve as a guiding principle for the testing activities and ensure that they are aligned with the overall objectives of the project.  
  


# Project Timeline and Resources

### Estimated project timeline, including major milestones.

|  |  |  |  |
| --- | --- | --- | --- |
| TASKS | W1 | W2 | W3 |
| Wireframe and requirement gathering |  |  |  |
| Login page creation |  |  |  |
| User and Admin Dashboard creation |  |  |  |
| Create Timesheet and Feedback Feature |  |  |  |
| Connect to Database and Warehouse |  |  |  |
| Data Engineering for the project |  |  |  |
| Model development for the problem |  |  |  |
| UI/UX design of the website |  |  |  |

**Appendix Title**

Document Title