**High-Level Design Document**

Final Project

Contents

[1 Project Overview 1](#_Toc163934858)

[1.1.1 Brief introduction of the project. 1](#_Toc163934859)

[1.1.2 Purpose of the project. 1](#_Toc163934860)

[1.1.3 Goals of the project: 2](#_Toc163934861)

[2 Project Scope 2](#_Toc163934862)

[2.1.1 Platforms and devices the mobile app will support (e.g., iOS, Android). 2](#_Toc163934863)

[2.1.2 Key features and functionalities of the application. 2](#_Toc163934864)

[3 Architecture and Technology Stack 3](#_Toc163934865)

[3.1.1 Overall architecture of the application. 3](#_Toc163934866)

[3.1.2 Architecture 3](#_Toc163934867)

[3](#_Toc163934868)

[3.1.3 Technology stack (e.g., programming languages, frameworks, libraries). 3](#_Toc163934869)

[3.1.4 Rationale behind the chosen technology stack. 5](#_Toc163934870)

[4 User Interface Design 6](#_Toc163934871)

[4.1.1 User interface (UI) design approach. 6](#_Toc163934872)

[5 Full Stack Application 7](#_Toc163934873)

[5.1.1 Frontend Application Approach. 7](#_Toc163934874)

[5.1.2 Backend Development Approach. 8](#_Toc163934875)

[6 Data Engineering. 9](#_Toc163934876)

[6.1.1 Data Generation. 9](#_Toc163934877)

6.1.2 Data Warehouse ………………………………………………………………………………………………......10

**7 Data Science…………………………………………………………………………………………………….…11**

7.1.1 Data Science Approach ………………………………………………………………………………………….12

7.1.2 Model Development ………………………………………………………………………………………………12

# Project Overview

### Brief introduction of the project.

Our project, the "Timesheet and Feedback System," is designed to streamline essential administrative tasks for employees. By simplifying timesheet creation and feedback submission while implementing role-based access and project-specific customization, we aim to enhance efficiency and accountability within the organization. With automated reminders ensuring timely completion, our platform facilitates seamless communication and fosters a culture of continuous improvement.

### Purpose of the project.

To simplify timesheet creation and feedback submission processes, enhancing efficiency and accountability for employees.

The key goals of the project include:

* Efficiency enhancement: By streamlining timesheet creation and feedback submission processes, the project aims to reduce the time and effort required for administrative tasks, thus increasing overall organizational efficiency and productivity.
* Communication optimization: Through the implementation of role-based access controls and project-specific feedback customization, the project seeks to facilitate more targeted and meaningful communication between employees and administrators. This not only enhances accountability but also promotes a culture of transparency and collaboration within the organization.

### Goals of the project:

* Streamline Administrative Processes: Simplify timesheet creation and feedback submission procedures to minimize manual effort and increase productivity.
* Ensure Compliance and Accountability: Implement mandatory submission requirements for timesheets and feedback to ensure adherence to organizational policies and foster accountability among employees.
* Development of Core Web Application Components:
  + Front End Interface: Design a user-friendly web interface for timesheet creation and feedback submission, optimized for accessibility and ease of use across different browsers.
  + Backend Implementation: Develop robust backend systems to handle data processing, storage, and user authentication for seamless functionality of the web application.
* Browser Compatibility:
  + Ensure compatibility of the web application with popular web browsers such as Chrome, Firefox, and Safari, to provide a consistent experience for all users.

By simplifying administrative tasks, promoting accountability, fostering collaboration, optimizing resource allocation, and enhancing employee satisfaction, the project aims to drive efficiency and productivity while nurturing a culture of continuous improvement within the organization.

# Project Scope

### Platforms and devices the mobile app will support (e.g., iOS, Android).

|  |  |
| --- | --- |
| Platforms | Version |
| Windows, macOS, Linux | Latest Version |
| Android, IOS | 10 and above, 11 and above |

### Key features and functionalities of the application.

**User Registration and Authentication:**

User-friendly registration process for patients to create their accounts.

Secure authentication mechanisms, including username/password

**Timesheet Creation:**

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

Real-time updates on instrument availability and stock levels.

**Feedback Submission:**

Users can provide feedback on various aspects such as projects, tasks, and team collaboration, with questions tailored based on their roles and project involvement.

**Role-Based Access Control:**

Access to timesheets and feedback features is controlled based on users' roles within the organization, ensuring data security and privacy.

**Project-Specific Feedback:**

Feedback questions are customized based on the specific project an employee is working on, allowing for targeted and relevant feedback.

**Automated Reminders:**

Automated notifications remind users of pending timesheets and feedback submissions, reducing the likelihood of missed deadlines.

**User-Friendly Interface:**

The application features an intuitive and responsive web interface, optimized for ease of use across various devices and screen sizes.

# Architecture and Technology Stack

### Overall architecture of the application.

The architecture and technology stack for the web application will be designed to ensure scalability, reliability, and performance. Here's an outline of the proposed architecture and technology stack:

**Client-Side Interface: (JMAN)**

The client-side interface forms the user-facing component of the mobile application. It provides an intuitive and responsive user interface, allowing patients to access their personalized dashboards, track orders and receive message. The interface is designed to support multiple devices, including iPhone, iPads and Android devices, ensuring a seamless user experience across different screen sizes and resolutions.

**Application Logic Layer: (180-Medical)**

The application logic layer serves as the backbone of the mobile application, encompassing the core business logic and functionality. It handles the processing of user requests, orchestrates data retrieval and storage, and performs necessary calculations and validations.

### Architecture

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

**Technology Stack for the Application Platform:**

The architecture and technology stack for the web application will be designed to ensure scalability, reliability, and performance. Here's an outline of the proposed architecture and technology stack:

**Frontend Framework:**

**React.js**: A popular JavaScript library for building user interfaces. It offers a component-based architecture, making it easy to develop interactive and responsive frontend applications.

**Backend Framework:**

Node.js: A runtime environment for executing JavaScript code on the server-side. Node.js provides non-blocking, event-driven architecture, ideal for building scalable and real-time web applications.

**Database:**

MongoDB: A NoSQL database that offers flexibility and scalability for storing and managing semi-structured data. MongoDB's document-oriented model allows for easy integration with Node.js applications.

For the data engineering aspect of the project, Snowflake will serve as the cloud-based data warehouse, providing scalability and performance for handling large volumes of data. dbt (data build tool) will be utilized for data transformation and modelling, allowing for the creation of data pipelines and the orchestration of data workflows.

**Snowflake**:

**Data Storage**: Snowflake will store all relevant data, including timesheet records, feedback submissions, user information, and project details, in a centralized and scalable manner.

**Data Processing:** Snowflake's data processing capabilities will be leveraged for performing complex queries, aggregations, and analytics on the stored data, enabling efficient data retrieval and analysis.

**Data Integration:** Snowflake supports seamless integration with various data sources and tools, facilitating data ingestion from multiple sources into the data warehouse.

**dbt (Data Build Tool):**

**Data Transformation:** dbt will be used for data transformation tasks, including cleaning, structuring, and modelling the raw data stored in Snowflake. It allows for the creation of SQL-based transformation pipelines, enabling the development of curated datasets and analytical models.

**Data Orchestration:** dbt's workflow management capabilities will orchestrate the execution of data transformation processes, ensuring the consistency and reliability of the data pipelines.

**Data Documentation:** dbt provides features for documenting the data transformation logic and business rules, enabling better understanding and collaboration among data engineers and analysts.

For the data science aspect of the project, machine learning models will be developed to analyse feedback data and derive insights for improving employee performance and project outcomes.

**Data Science (Machine Learning):**

Model Development: Machine learning algorithms will be employed to analyse feedback data and identify patterns or trends related to employee performance, project effectiveness, and overall organizational dynamics.

**Feature Engineering:** Data scientists will perform feature engineering to extract relevant features from the feedback data, such as sentiment analysis, topic modelling, or user behaviour metrics, to train predictive models.

**Model Training and Evaluation:** Supervised or unsupervised learning techniques will be applied to train machine learning models on historical feedback data. The models will be evaluated using appropriate performance metrics to assess their effectiveness and generalization ability.

**Model Deployment:** Deployed models will provide actionable insights and recommendations to stakeholders, enabling data-driven decision-making for improving organizational processes and performance.

### Rationale behind the chosen technology stack.

In our endeavour to create an efficient Timesheet and Feedback System, React.js stands as our technology of choice. With its dynamic user interface capabilities and streamlined development workflows, React.js ensures a seamless user experience and enhanced organizational productivity. The rationale behind this selection is based on the following considerations:

**Scalability**: The selected technologies, such as Node.js for backend and Snowflake for data warehouse, are inherently scalable, allowing the web application to handle growing data volumes and user loads efficiently.

**Performance**: With technologies like React.js for frontend and MongoDB for database, the application can deliver fast response times and smooth user experiences, enhancing overall performance.

**Flexibility**: The architecture supports flexibility in development, allowing for easy integration of new features and adaptation to changing business requirements.

**Cross-platform Compatibility**: React.js for frontend development and Node.js for backend allow for building applications that are compatible with multiple platforms, ensuring wider reach and accessibility.

**Cost Efficiency**: Utilizing open-source technologies like React.js, Node.js, and MongoDB can result in cost savings by reducing licensing fees and development time.

**Community Support**: The selected technologies have large and active developer communities, providing extensive support, resources, and libraries for development and problem-solving.

**Ease of Development**: The component-based architecture of React.js and the event-driven architecture of Node.js simplify development efforts, enabling faster time-to-market for the web application.

**Reliability**: Snowflake's cloud-based data warehouse ensures data reliability and availability, minimizing the risk of data loss or downtime.

**Data Processing and Analysis**: The combination of Snowflake for data storage and dbt for data transformation enables efficient data processing and analysis, empowering organizations to derive valuable insights from their data.

**Futureproofing**: By choosing modern and widely adopted technologies, the web application is well-positioned for future scalability, updates, and enhancements, ensuring its longevity and relevance in the rapidly evolving technological landscape.

Overall, the chosen architecture and technology stack offer a robust foundation for building a scalable, reliable, and high-performance web application that meets the organization's needs and objectives effectively.

# User Interface Design

### User interface (UI) design approach.

For our project, the user interface (UI) design approach will prioritize simplicity, intuitiveness, and efficiency. We aim to create a clean and user-friendly interface that enhances the user experience and facilitates seamless interaction with the timesheet and feedback system. Key aspects of our UI design approach include:

**User-Centric Design:** We will adopt a user-centric approach, focusing on understanding the needs, preferences, and behaviours of our target users. By empathizing with users and incorporating their feedback throughout the design process, we ensure that the UI meets their expectations and supports their workflow.

**Minimalist Design:** Our UI design will follow a minimalist approach, emphasizing clarity, simplicity, and ease of use. We will strive to eliminate unnecessary clutter and distractions, presenting only the essential elements and information relevant to the task at hand.

**Intuitive Navigation:** Navigation within the application will be intuitive and straightforward, allowing users to easily access different features and functionalities. Clear navigation pathways, intuitive menus, and logical information architecture will guide users seamlessly through the application.

**Consistent Design Language:** We will maintain consistency in design elements, such as colors, typography, icons, and layout, across the entire application. Consistency not only enhances visual appeal but also fosters familiarity and usability, enabling users to navigate the application with confidence.

**Responsive Design:** Our UI design will be responsive and adaptable to different screen sizes and devices, including desktops, tablets, and smartphones. Responsive design ensures a consistent and optimal user experience across various devices, enhancing accessibility and usability.

**Feedback and Error Handling:** The UI will provide clear feedback to users, confirming actions, acknowledging inputs, and providing informative error messages when necessary. Effective feedback mechanisms enhance user confidence and help prevent errors and misunderstandings.

**Accessibility:** We will prioritize accessibility in our UI design, ensuring that all users, including those with disabilities, can access and interact with the application effectively. This includes considerations such as colour contrast, keyboard navigation, screen reader compatibility, and other accessibility best practices.

By incorporating these principles into our UI design approach, we aim to create a user interface that not only meets the functional requirements of the timesheet and feedback system but also delivers an engaging, intuitive, and enjoyable user experience.

# Full Stack Application

### Frontend Application Approach.

Our frontend application approach prioritizes a seamless user experience and efficient development process. By leveraging React.js and modern frontend technologies, we aim to build a responsive, intuitive, and performant web application for our timesheet and feedback system.

**Technology Stack Selection:**

**React.js:** As the primary frontend framework, React.js offers a component-based architecture, making it easier to build reusable UI components and manage state efficiently.

**React Router:** For client-side routing, React Router enables navigation between different views within the application.

**Responsive Design:**

The frontend application will be designed with responsiveness in mind, ensuring that it functions well across various devices and screen sizes, including desktops, tablets, and smartphones.

CSS frameworks like Bootstrap or Material-UI may be utilized to streamline the implementation of responsive layouts and UI components.

**Component-Based Architecture:**

The UI will be built using a modular and component-based architecture, with each component responsible for a specific piece of functionality or user interface element.

Components will be reusable, enabling efficient development and maintenance of the application.

**User Interface Design:**

The user interface will follow a clean and intuitive design approach, focusing on simplicity, clarity, and ease of use.

Consistent design patterns, typography, colour schemes, and visual elements will be employed throughout the application to ensure a cohesive user experience.

Interactive elements, such as buttons, forms, and menus, will be designed to provide clear feedback and guide users through the application's functionalities.

**State Management:**

Redux or the Context API will be utilized for managing the application's state, including user authentication, timesheet data, feedback submissions, and UI state.

State will be normalized and stored in a single immutable state tree, facilitating predictable data flow and easier debugging.

**API Integration:**

The frontend application will communicate with the backend server through RESTful APIs or GraphQL queries for fetching and updating data.

Axios or the native Fetch API may be used for making asynchronous HTTP requests to the backend server.

**Authentication and Authorization:**

User authentication and authorization will be implemented to ensure secure access to the application's features and data.

**Performance Optimization:**

Performance optimization techniques, such as code splitting, lazy loading, and caching, will be employed to improve the application's loading speed and runtime performance.

Bundle size will be minimized through tree shaking and code splitting, optimizing the application for faster load times.

By following this frontend application approach, we aim to deliver a robust, user-friendly, and performant web application that meets the requirements of our timesheet and feedback system while providing an exceptional user experience.

### Backend Development Approach.

Our backend approach focuses on building a robust, scalable, and secure system to handle the business logic, data management, and API interactions for our timesheet and feedback application.

**Technology Stack Selection:**

**Node.js**: As the foundation of our backend, Node.js provides a non-blocking, event-driven architecture that is well-suited for building scalable and real-time applications.

**Express.js:** A minimalist web framework for Node.js, Express.js simplifies the development of RESTful APIs and middleware components.

**MongoDB:** A NoSQL database, MongoDB offers flexibility and scalability for storing and managing semi-structured data, such as timesheet records and user feedback.

**Mongoose:** A MongoDB object modelling library for Node.js, Mongoose simplifies interactions with the database through schema validation, query building, and data manipulation.

**RESTful API Design:**

Our backend will expose RESTful APIs to handle CRUD operations for timesheets, feedback submissions, user management, and authentication.

API endpoints will follow best practices for resource naming, HTTP methods, status codes, and request/response formats to ensure consistency and ease of consumption.

**Authentication and Authorization:**

User authentication will be implemented using session-based authentication to securely manage user sessions and access control.

Middleware functions will be used to authenticate and authorize incoming requests, ensuring that only authorized users can access protected resources.

**Data Modelling and Persistence:**

Data models will be defined using Mongoose schemas, representing entities such as users, timesheets, feedback submissions, and projects.

MongoDB will serve as the primary data store, providing fast and scalable storage for our application's data.

**Validation and Error Handling**

Input validation and error handling will be implemented at various layers of the application to ensure data integrity and prevent security vulnerabilities.

Middleware functions and schema validation mechanisms provided by Mongoose will be utilized to validate incoming requests and enforce data constraints.

**Scalability and Performance Optimization:**

Our backend architecture will be designed with scalability in mind, allowing horizontal scaling through load balancing and clustering.

Performance optimization techniques such as caching, query optimization, and asynchronous processing will be employed to improve response times and throughput.

By following this backend approach, we aim to develop a reliable, efficient, and scalable backend system that forms the foundation of our timesheet and feedback application, ensuring seamless functionality and optimal performance for users.

# Data Engineering.

### Data Generation.

In our Data Engineering endeavours, we prioritize the meticulous generation of datasets that serve as the backbone of our systems. Through a blend of real-world data extraction from websites and synthetic data creation via platforms like Mockaroo, we ensure the richness and diversity of our datasets, enabling robust testing and analysis.

**Comprehensive Data Strategy:** Our approach to data generation is all-encompassing, combining the authenticity of real-world data with the flexibility of synthetic data. This dual strategy allows us to cover a broad spectrum of scenarios, from typical use cases to rare edge cases, providing a holistic view of data interactions within our system.

**Realism and Relevance:** We strive for realism in our datasets, ensuring that they closely mirror the complexities of real-world scenarios. By incorporating actual data sourced from websites and augmenting it with synthetic data, we create datasets that accurately reflect the intricacies of our domain, enhancing the effectiveness of our testing and analysis efforts.

**Diverse Representation:** Our datasets boast a diverse representation of data variations and edge cases, capturing the full spectrum of possible scenarios. This diversity enables thorough testing and analysis, allowing us to identify and address potential issues before they impact the system's performance or user experience.

**Integration of Real Data Sources**: Real data sourced from websites provides valuable insights into real-world user behaviours and interactions. By integrating this data with synthetic datasets, we enrich our datasets with real-world context, enhancing their relevance and applicability to our system's functionality and user requirements.

In conclusion, our data generation approach combines the best of both worlds – the authenticity of real-world data and the flexibility of synthetic data. By creating rich, diverse, and reliable datasets, we empower our systems with the insights needed to make informed decisions, drive innovation, and deliver exceptional user experiences.

### Data Warehouse.

In our data engineering infrastructure, we leverage Snowflake as our cloud-based data warehouse solution and dbt (data build tool) for efficient data transformation. This combination enables us to build a robust and scalable data ecosystem that supports our analytics, reporting, and decision-making processes.

**Snowflake Data Warehouse:**

Snowflake serves as the cornerstone of our data infrastructure, providing a centralized and scalable platform for storing, processing, and analysing large volumes of data.

Its cloud-native architecture offers elasticity, allowing us to scale compute and storage resources dynamically to meet changing demands.

Snowflake's unique multi-cluster architecture ensures high availability and performance, enabling us to execute complex queries and analytics tasks efficiently.

**dbt for Data Transformation:**

dbt complements Snowflake by offering a powerful set of data transformation capabilities, allowing us to build and maintain data pipelines with ease.

With dbt, we can transform raw data into curated datasets and analytical models, leveraging SQL-based transformations and version-controlled workflows.

Its modular and extensible nature enables collaboration among data engineers and analysts, fostering a culture of data-driven decision-making and continuous improvement.

**Integration and Orchestration:**

We integrate Snowflake with dbt to orchestrate end-to-end data transformation workflows seamlessly.

Snowflake's native support for dbt enables direct integration, simplifying the deployment and execution of dbt models within our data warehouse environment.

This integration streamlines the data transformation process, from raw data ingestion to final analysis, ensuring data consistency and reliability throughout.

**Scalability and Performance:**

The combination of Snowflake and dbt offers unparalleled scalability and performance for our data engineering tasks.

Snowflake's scalable compute and storage resources, combined with dbt's optimized data transformation processes, enable us to handle large and complex datasets efficiently.

This scalability ensures that our data infrastructure can grow seamlessly to accommodate evolving business needs and increasing data volumes.

**Data Governance and Documentation:**

Snowflake and dbt provide robust data governance and documentation features, enabling us to maintain data lineage, lineage, and documentation.

With Snowflake's metadata management capabilities and dbt's built-in documentation tools, we can track data lineage, document transformations, and ensure compliance with regulatory requirements.

By leveraging Snowflake as our data warehouse solution and dbt for data transformation, we have built a modern and efficient data infrastructure that powers our analytics and decision-making processes. This combination offers scalability, performance, and flexibility, enabling us to derive actionable insights and drive business growth effectively.

A screenshot of a computer

Description automatically generated

# Data Science.

### Data Science Approach.

Incorporating data science into our project enhances decision-making, improves efficiency, and drives innovation. Here's our approach to integrating data science methodologies and techniques:

**Problem Definition and Goal Setting:**

We begin by clearly defining the problem statement and setting specific goals that align with our project objectives.

Understanding the business context and user requirements is crucial for identifying relevant data science tasks and outcomes.

**Data Collection and Exploration:**

We gather relevant data sources, including timesheet records, feedback submissions, and project details, ensuring data completeness and quality.

Exploratory data analysis (EDA) techniques are employed to gain insights into the data, identify patterns, and uncover potential correlations.

**Feature Engineering and Selection:**

We engineer meaningful features from the raw data to capture relevant information and enhance model performance.

Feature selection techniques are applied to identify the most predictive features while reducing dimensionality and improving model interpretability.

**Model Development and Evaluation:**

We develop machine learning models tailored to address specific use cases, such as predicting employee performance or project outcomes based on feedback data.

Models are trained using appropriate algorithms, validated using cross-validation techniques, and evaluated based on relevant performance metrics.

**Iterative Refinement and Optimization:**

We iteratively refine and optimize our models based on feedback and performance evaluation results.

Techniques such as hyperparameter tuning, ensemble methods, and model interpretation are employed to enhance model accuracy and generalization ability.

**Deployment and Integration:**

Once models have been validated and optimized, they are deployed into our production environment, integrated with our existing systems, and made accessible for end-users.

Model performance is continuously monitored, and updates are rolled out as needed to ensure ongoing relevance and effectiveness.

**User Feedback and Iteration:**

We solicit feedback from end-users and stakeholders to assess model utility, usability, and impact on decision-making processes.

Iterative improvements are made based on user feedback, allowing us to continuously enhance model performance and relevance over time.

**Ethical Considerations and Transparency:**

We adhere to ethical principles and guidelines throughout the data science process, ensuring fairness, transparency, and accountability in model development and deployment.

Bias mitigation techniques and fairness metrics are employed to mitigate potential biases and ensure equitable outcomes for all stakeholders.

By following this data science approach, we aim to leverage data-driven insights to enhance employee performance, project outcomes, and organizational processes effectively. Through rigorous problem definition, data exploration, model development, and iteration, we strive to deliver impactful solutions that drive innovation and business success.

A graph of blue bars

Description automatically generated

### Model Development.

Model development is a critical phase in our project, where we harness machine learning algorithms to derive insights from feedback data and improve employee performance and project outcomes. Here's our approach to model development:

**Problem Formulation and Dataset Preparation:**

We start by formulating the problem statement and defining the target variables based on project objectives.

Relevant datasets, including feedback submissions, employee profiles, and project details, are collected, cleaned, and prepared for analysis.

**Exploratory Data Analysis (EDA):**

Exploratory data analysis techniques are applied to gain a deeper understanding of the data, identify patterns, and uncover potential relationships between variables.

Visualization tools and statistical methods help us explore data distributions, correlations, and outliers, guiding feature selection and engineering efforts.

**Feature Engineering and Selection:**

We engineer meaningful features from the raw data, leveraging domain knowledge and EDA insights to capture relevant information.

Feature selection techniques, such as correlation analysis and feature importance scores, are employed to identify the most predictive features for model training.

**Model Selection and Training:**

Based on the problem formulation and dataset characteristics, appropriate machine learning algorithms are selected for model training.

Models such as regression, classification, or clustering algorithms are trained using labeled data, optimizing hyperparameters to maximize performance.

**Model Evaluation and Validation:**

Trained models are evaluated using suitable performance metrics, such as accuracy, precision, recall, or F1-score, depending on the nature of the problem.

Validation techniques such as cross-validation or holdout validation are employed to assess model generalization and robustness.

**Hyperparameter Tuning and Optimization:**

Hyperparameters of the selected models are tuned using techniques like grid search or random search to optimize model performance.

Ensemble methods or model stacking may be utilized to combine multiple models for improved accuracy and stability.

**Interpretability and Explainability:**

We prioritize model interpretability and explainability to enhance trust and understanding among stakeholders.

Techniques such as feature importance analysis, SHAP values, or partial dependence plots are employed to interpret model predictions and insights.

**Model Deployment and Monitoring:**

Deployed models are integrated into our production environment, where they generate predictions or recommendations based on real-time data inputs.

Model performance is continuously monitored, and updates are rolled out as needed to maintain accuracy and relevance over time.

Through systematic model development processes, we aim to leverage machine learning techniques effectively to extract actionable insights from feedback data, driving continuous improvement in employee performance and project outcomes. By ensuring model robustness, interpretability, and scalability, we empower stakeholders to make informed decisions and achieve organizational objectives effectively.

A screenshot of a computer screen

Description automatically generated