Logging Framework & Documentation

Introduction

This documentation provides guidelines on configuring and effectively utilizing the logging framework implemented in the project. The logging framework is designed to enhance debugging, monitoring, and troubleshooting by providing multi-level log messages.

Implementation & Mechanism

1) Back-End (Express.js in TypeScript):
   Framework: log4js
   Description: log4js is a powerful logging framework for Node.js applications, written in TypeScript. It allows developers to log messages at different levels and provides flexibility in configuring appenders.

2) Front-End (React-Native):
   Framework: Built-in Console Logging
   Description: In React-Native, the built-in console logging mechanism is utilized. This straightforward approach allows logging messages to the console for debugging and monitoring purposes.
3) Database Logging (MySQL):
   Framework: MySQL's Built-in Logging Features
   Description: MySQL provides built-in logging features that can be leveraged for storing logs directly in the database. This ensures persistent storage of log data.

Logging Usage & Agenda

- Follow documentation on how to configure and use the logging framework effectively. It includes guidelines on interpreting log levels. The logging system is extensively used in the project to capture levels of detail and urgency.
- By incorporating these logging mechanisms, we ensure the project uses effective debugging, monitoring, and troubleshooting capabilities at both the back end and front end, with persistent storage in the database. The use of different log levels enhances the granularity of log messages, aiding developers in identifying and addressing issues.

Logging Levels

Debug	Use for detailed debugging information. Messages at this level provide in-depth insights into the application's internal processes.
Info	Use for general information about the application's execution. INFO messages highlight key events and milestones.
Warn	Use to indicate potential issues or situations that require attention. WARN messages signal cautionary information.
Error	Use to report errors or critical issues that require immediate attention. ERROR messages indicate a malfunction or unexpected behavior.