**Project Summary: Inventory Management System**

**Introduction**

The Inventory Management System (IMS) is conceived as a comprehensive solution designed to streamline the process of managing and tracking inventory within ROC-Nijmegen. This offline, locally-run system aims to automate the lending and tracking of inventory items, enhancing the efficiency of resource utilization while ensuring easy access and management.

**Project Objectives**

* To provide an accurate, real-time overview of the inventory status.
* To simplify the process of lending out and returning inventory items for both students and staff.
* To implement secure and efficient user identification through RFID technology.
* To ensure the system is operable offline and locally on a single computer, catering to the institution's infrastructure limitations.

**System Overview**

The IMS is structured to accommodate various user roles, including students, teachers, and administrators, each with customized access levels to suit their needs. The system utilizes QR codes for item identification and RFID technology for user authentication, facilitating a seamless and automated transaction process.

**Features and Functionalities**

* **Inventory Tracking**: Utilize QR codes for easy identification and tracking of inventory items, with a manual selection option for items not compatible with QR codes.
* **User Authentication**: Employ RFID technology for a secure and efficient user identification process, distinguishing between students, teachers, and administrators.
* **Database Management**: Maintain a centralized database to store comprehensive records of users, inventory items, transactions, and item availability.
* **User Interface**: Develop intuitive user interfaces tailored to different user roles, supporting functionalities such as item lending/return, user and item management, and access to lending history and statistics.
* **Access Control**: Implement role-based access control (RBAC) to ensure sensitive or valuable items are only accessible by authorized personnel.
* **Notifications and Alerts**: Generate automated alerts for overdue items, inventory restocks, and maintenance needs to ensure the timely return of items and upkeep of the inventory.

**Technology Stack**

* **Frontend/Backend**: Next.js for building the user interface and handling server-side functionalities.
* **Database**: SQLite for a lightweight, local database solution that supports the system's offline functionality.

**Operational Environment**

The IMS is designed to operate in an offline environment, running locally on hardware meeting minimal performance specifications. This design consideration ensures that the system remains functional without reliance on internet connectivity, addressing the institution's infrastructure constraints.