PROJECT INITIATION DOCUMENTATION

Problem Statement

The NGCDF Board Organization currently lacks an efficient system for managing and monitoring its inventory for ICT gadgets. Its current manual tracking system is inefficient in that it can lead to challenges such as over-ordering, inefficient tracking, and errors in issuing or maintenance. Such challenges have the capability to compromise decision making, increase risks and hinder strategic resource management.

Proposed Solution

A proposed solution was reached, which sought the need for an automated inventory management system that is mainly web based, to be able to address such challenges and enhance organization efficiency.
The system was preferred to be web based because of various reasons such as easy accessibility, lower deployment cost, a unified database and a simplified maintenance

For this ICT Gadget Inventory System, the following technologies and languages can be considered:

1. Backend Development:

   a) Node.js with Express: For building the server-side logic and APIs.
   b) JSON Web Tokens (JWT): For implementing user authentication and authorization.

2. Frontend Development:
   a) React.js or Angular: For building the user interface and frontend components.
   b) HTML/CSS/JavaScript: Standard tech for structuring web pages and adding interactivity.
   c) Redux or NgRx: State management libraries to manage application state in React or Angular applications.
   d) Figma or Vue UI: For designing the User Interface

3. Database Development:
   a) MySQL
   b) MongoDB or PostgreSQL: As the database management system for storing inventory data. MongoDB can be suitable for handling unstructured data, while PostgreSQL offers robust relational database capabilities.
   c) Mongoose (for MongoDB) or Sequelize (for PostgreSQL): As object modeling libraries to interact with the database.

Ultimately, I'd go for MySQL since it provides a reliable and scalable database solution that can meet the requirements system.

4. Development Tools and Utilities:

Visual Studio Code: A popular code editor for writing and debugging code.
Git: Version control system for collaborative development.
Webpack or Parcel: Module bundlers for optimizing and bundling frontend assets.
Postman: For testing APIs and endpoints during development.

5. Deployment and Hosting:

   a) Heroku, AWS, Google cloud or Azure: Cloud platforms for deploying and hosting the application.

6. Additional Libraries and Frameworks:

   a) Socket.IO: For real-time communication and updates.
   b) Passport.js: For authentication strategies in Node.js applications.

Expected Benefits

1. Improved Inventory Management which will allow staff members to to track, monitor, and manage inventory much more effectively.

2. Optimized resource allocation in that decision-makers in the organization will be able to make informed decisions regarding resource allocation, procurement, and maintenance of gadgets and their consumables.

3. Enhanced accountability whereby accountability of gadgets among staff members and various offices will be enhanced.

4. Improved quality of User experience which will lead to easy adaptation and satisfaction with the system and eventually boosting staff productivity and morale

Project schedule

1. Project Initiation and Planning Phase:

   ✓ Define project scope, objectives, and stakeholders and gathering requirements
   and prioritize features }
   ✓ Design database schema and system architecture } - 1 week
   ✓ Create wireframes and UI design }

2. Development Phase:

- ✓ Set up development environment and version control and Backend development (APIs, database integration) - 2 weeks
- ✓ Frontend development (UI components, user interface) - 1 week
- ✓ Integration testing and bug fixing - 1 weeks

3. Testing and Quality Assurance Phase:

- ✓ Conduct unit testing for backend and frontend components, Perform integration testing and system testing and addressing bugs and issues identified during testing - 1 week

4. Deployment and Implementation Phase:

- ✓ Preparing deployment environment (cloud platform, servers) and deploying system staging environment for final testing - 1 week
- ✓ Conduct user acceptance testing (UAT) and gather feedback, addressing feedback and make final adjustments }
- ✓ Deploy system to production environment } - 1 week

5. Project Closeout Phase:

- ✓ Finalizing project documentation and deliverables and handing over system to operations and maintenance team - 1 week