

Read Me file for Mobile app

## Overview

The Mobile Inventory Management App is designed to simplify inventory tracking and management for businesses. It allows users to add, update, and delete items, view stock levels, and generate reports through an intuitive mobile interface. Integrated with a web-based dashboard, the app ensures seamless data synchronization and accessibility for users on the go. This project was developed as part of a Work Integrated Learning (WIL) experience, with Liteboho Solutions as the client.

## Features

- **User Authentication:** Secure login and registration functionality.
- **CRUD Operations:** Users can create, read, update, and delete inventory items.
- **Search Functionality:** Easy search feature to quickly locate items in inventory.
- **Real-Time Data Syncing:** Sync data with a web-based dashboard for comprehensive inventory management.
- **Help Feature:** In-app help buttons to assist users with navigating the app.
- **Reporting:** View inventory levels and generate reports on stock status.

## Technologies Used

- **Frontend:** Java (Android Studio)
- **Backend:** PHP, MySQL
- **Database:** MySQL for data storage and management
- **Server:** WAMP (Windows, Apache, MySQL, PHP) server for local development
- **API:** RESTful APIs for backend communication
- **Documentation:** Google and YouTube tutorials for research and implementation guidance

## Installation

### Prerequisites

- **Android Studio** for running and building the mobile application.
- **WAMP Server** (or any LAMP server equivalent) to host the backend and database.

### Steps

#### 1. Clone the Repository:

```
git clone [https://github.com/BlackSIYA11/Mobile-App-Inventory-System-/tree/main]
```

#### 2. Set Up Backend and Database:

- Install WAMP server.
- Set up the MySQL database in phpMyAdmin with the necessary tables:

- **Users:** For authentication and user management.
  - **Items:** For storing inventory data.
  - Import the provided SQL script (if any) to create the required tables and fields.
3. **Configure API Endpoints:**
- Place backend PHP files in the WAMP server's www directory.
  - Update API endpoint URLs in the Android app code if necessary to match your server configuration.
4. **Run the App:**
- Open the project in Android Studio.
  - Configure the server IP address in the app code for backend communication.
  - Run the app on an Android emulator or physical device.

## Usage

- **Login/Registration:** Register a new account or log in to access the inventory system.
- **Manage Inventory:** Add, edit, delete, and search for items in your inventory.
- **View Reports:** Access inventory reports and stock levels on the dashboard.
- **Help:** Tap the help button for guidance on navigating the app.

## Troubleshooting

- **Database Connection Errors:** Ensure the WAMP server is running, and that the database configuration in the app matches your server settings.
- **API Issues:** Verify API endpoints and server IP addresses in the code.
- **UI or Data Sync Delays:** Restart the app or WAMP server if data is not syncing.

## Acknowledgments

This app was developed as part of a WIL project at Liteboho Solutions, with team members contributing to different aspects including mobile app development, backend integration, and project management.