# Inventory Management System (IMS)

Project Report

By:

**Onkar Uday Deokate**

**219303061**

Contact Information:

[officialonkardeokate@gmail.com](mailto:officialonkardeokate@gmail.com)

[**https://www.linkedin.com/in/onkardeokate/**](https://www.linkedin.com/in/onkardeokate/)

**9420920656**

A close up of a logo

Description automatically generated

2021-2025

Department of Computer and Communication Engineering

School of Computing and Intelligent Systems

Manipal University Jaipur

VPO. Dehmi Kalan, Jaipur, Rajasthan, India – 303007

## Introduction

The Inventory Management System (IMS) is a user-friendly application designed for efficient inventory management. It provides features for user authentication, item management, and report generation, all through a modern and intuitive interface built with Python's Tkinter library and the ThemedTk theme.

## Features

* User Authentication: Secure login for users with role-based access.
* Inventory Management:
* Add new items.
* Edit existing items.
* Delete items.
* View and search inventory.
* Reporting:
* Generate reports on low-stock items.
* User Interface: Modern design using Tkinter with ThemedTk for enhanced aesthetics.

## Prerequisites

* Python 3.x
* Tkinter library
* `ttkthemes` library
* `json` and `os` for data handling and file operations

## Configuration

The IMS requires two JSON files for storing user and inventory data:

**`users.json`**

Stores user credentials.

A screenshot of a computer program

Description automatically generated

`**inventory.json`**

Stores inventory item details.

A screenshot of a computer

Description automatically generated

## Usage

1. Start the Application:

Run the application using Python:

python main.py

2. Login:

- Enter your username and password on the login screen.

- Click "Login" to access the main menu.

3. Main Menu:

- Add Product: Input product details and add a new product to the inventory.

- Edit Product: Modify details of an existing product.

- Delete Product: Remove a product from the inventory.

- View Inventory: Display the current inventory.

- Generate Reports: View a report on low-stock items.

- Logout: Exit the application and return to the login screen.

## Code Structure

`**main.py`**

Contains the main application code, organized into several classes:

- `User`: Represents a user with credentials.

- `InventoryItem`: Represents an inventory item with attributes.

- `InventoryManagementSystem`: Handles user authentication and inventory operations.

- `IMSInterface`: Manages the user interface and user interactions.

Key Methods

- `load\_users()`: Loads user data from `users.json`.

- `load\_inventory()`: Loads inventory data from `inventory.json`.

- `save\_inventory()`: Saves the current inventory to `inventory.json`.

- `authenticate(username, password)`: Authenticates a user.

- `add\_item(product\_id, name, quantity, price)`: Adds a new item to inventory.

- `edit\_item(product\_id, name, quantity, price)`: Edits an existing inventory item.

- `delete\_item(product\_id)`: Deletes an inventory item.

- `create\_login\_screen()`: Generates the login interface.

- `create\_main\_menu()`: Generates the main menu interface.

- `create\_add\_product\_screen()`: Generates the add product interface.

- `create\_edit\_product\_screen()`: Generates the edit product interface.

- `create\_delete\_product\_screen()`: Generates the delete product interface.

- `create\_view\_inventory\_screen()`: Generates the view inventory interface.

- `create\_reports\_screen()`: Generates the reports interface.

## Security Considerations

- Password Storage: Currently, passwords are stored in plain text. For improved security, implement password hashing.

- Error Handling: Ensure robust error handling for file operations and input validation.

## Future Enhancements

- Implement password hashing for secure storage.

- Add input validation for user entries.

- Introduce logging for tracking and auditing.

- Enhance the user interface with search, filter, and sort functionalities.