Inventory Management System

Developer Guidelines

EEC 521/CIS 534 – Software Engineering Project

Submitted By

Jaydeep Ravaliya (2873999)

Kevin Panchal (2871507)

Jaykumar Mistry (2875386)

1.0 Introduction

This guide provides an overview of the Inventory Management System (IMS), a Django-based web application designed for efficient inventory tracking. IMS allows businesses to manage products, stock levels, suppliers, and reports. This guide outlines the structure of the codebase, tools used, and setup instructions.

Functionality

IMS streamlines:

- Product management (add/edit/delete).
- Stock level tracking and real-time updates.
- Generating reports for inventory analysis.
- Sending low-stock alerts via email.

Technologies Used

- **Framework:** Django
- **Frontend:** HTML, CSS, Bootstrap
- **Database:** SQLite3
- **Libraries:** Django ORM, email libraries

2.0 Getting Started

2.0 Getting Started Welcome aboard! Here's how to dive into Inventory Management System development:

1. Clone the Repository: Begin by obtaining the latest project code. Navigate to your terminal and use Git to clone the Inventory Management System repository from GitHub. You'll find the specific clone URL on the project's GitHub page.
   Github link: https://github.com/jaydeepravaliya/inventory_management_sys.git

2. Python Version: Python 3.9.6 or higher is required for the development.

3. Install Dependencies: IMS relies on several external libraries to function correctly. To install these dependencies, open your terminal and navigate to the project directory. Then, execute the following command: pip install -r requirements.txt This command instructs the pip package manager to install all the libraries listed in the requirements.txt file.

4. Set Up Your Development Environment: Having a comfortable and efficient development environment is crucial. Choose your preferred Integrated Development Environment (IDE) and configure it according to your preferences. Ensure you have a compatible Python interpreter installed as well. Most IDEs allow you to manage Python interpreters within their settings. Visual Studio is a great IDE and recommended for the development of this application.

5. Explore the Codebase: Now that you have the development environment set up, it's time to delve into the project's code. Take some time to explore the codebase and familiarize yourself with its structure.

3.0 Understanding the Codebase

Main Modules

- **Product Management Module:** Handles CRUD operations for products.
- **Stock Management Module:** Tracks stock levels and updates.
- **Reporting Module:** Generates detailed inventory reports.
- **Notification Module:** Sends email alerts for low stock.

Code Structure

- **models.py:** Defines database models for products, categories, and suppliers.
- **views.py:** Contains business logic for handling user requests.
- **templates/**: HTML templates for rendering the user interface.
- **static/**: Static files such as CSS, JavaScript, and images.

Key Files

- **`settings.py`**: Configuration settings for the Django project.
- **`urls.py`**: URL routing for the application.
- **`forms.py`**: Handles form validation for user input.
- **`admin.py`**: Admin panel configuration for managing models

Running the application

- Ensure you have Python, and the required libraries installed. Refer to their respective official documentation for installation instructions.
- Navigate to the project directory containing the main file.
- Run the application directly from the VS Code or using terminal: python manage.py run server

4.0 Conclusion

This guide serves as a starting point for developing and maintaining the IMS. For further assistance, refer to the official Django documentation or contact the project maintainers. Contributions are welcome to improve and expand the system.

For any questions or support:

Kevin Panchal- k.panchal60@vikes.csuohio.edu

Jaydeep Ravaliya- j.ravaliya@vikes.csuohio.edu

Jaykumar Mistry- j.mistry@vikes.csuohio.edu

Happy coding! 🎉