

## Smart Stock Inventory For Retail Stores

### Day1:

#### Overview of the project

Received an introduction to the internship structure, rules, and expectations. Understood the overall flow of tasks and learning outcomes.

We discussed the main problems in retail inventory management such as overstock and out-of-stock situations.

### DAY2:

We discussed and finalized the technical components required for building the Smart Stock Inventory Optimization System.

Frontend: HTML, CSS, JavaScript for creating the user interface and dashboard.

Backend: Python for implementing the core logic, stock calculations, alerts, and prediction models.

Database: MySQL / SQL / MongoDB for storing product details, sales records, supplier info, and stock levels.

NumPy: A Python library used in the backend to perform numerical calculations, mathematical operations, and forecasting formulas efficiently.

Pandas: A data handling library used for loading, cleaning, analyzing, and processing datasets such as product data and sales records.

Python Format Strings: Formatting feature in Python used to print structured, clean, and readable outputs in the backend system.

API Basics: Understanding how different systems communicate APIs such as FastAPI and Flask APIs, used if the project needs frontend-backend data exchange.

### DAY3:

Flask API: Flask is a lightweight Python web framework used to create simple APIs.

FastAPI: A modern and very fast Python framework supporting high performance, automatic documentation, and faster request handling.

#### Difference Between Flask and FastAPI:

- Flask: Simple, lightweight, beginner-friendly.
- FastAPI: Faster, modern, supports automatic documentation.

Postman: A tool used to test APIs by sending requests and checking responses.

DAY4:

GitHub Overview: GitHub is an online platform that stores Git repositories.

Repository (Repo): A folder where your project files and their history are stored.

Local Repository: A Git repository stored on your computer.

Cloning: Copying a GitHub repository to your local system.

Committing: Saving code changes with a message.

Pushing: Sending local commits to the GitHub remote repository.

Pulling: Getting the latest changes from GitHub.

Merge: Combining changes from one branch into another.

Fork: Creating your own copy of another repository.

Pull Request (PR): Request to merge changes into the main project.

Git Commands:

- git init
- git clone
- git status
- git add .
- git commit -m "message"
- git push origin main
- git pull
- git branch

Day5:

Introduction to SQL

SQL (Structured Query Language) is used to interact with databases (CRUD operations).

Create, Read, Update, Delete.

Database: Collection of tables.

Table: Rows and columns.

Primary Key: Unique identifier.

Foreign Key: Links tables.

Artificial Intelligence and Machine Learning basics were also covered.

Day6:

Normalization: Scaling or adjusting data to fit within a standard format.

In databases, normalization reduces redundancy and improves data integrity.

Explained 1NF, 2NF, 3NF, and BCNF.

Day7:

GitHub repository setup and project review conducted.

Day8:

Milestone 1 conducted – Initial module verification and basic functionality testing completed.

Day9:

Milestone 1 continuation – Validation of workflow, correction of minor issues, and preparation for next milestone.