K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

A Project Report

on

INVENTORY MANAGEMENT SYSTEM FOR CAR MANUFACTURES

Submitted in partial fulfillment of requirements for the award of the course

of

MGB1201 – PYTHON PROGRAMMING

Under the guidance of

Mrs.S.RAJESWARI M.E.

Assistant Professor / CSE

Submitted By

SANTHOSH M (8115U23ME039)

DEPARTMENT OF MECHANICAL ENGINEERING

K.RAMAKRISHNAN COLLEGE OF ENGINEERING
(Autonomous)

TRICHY-621 112

DECEMBER 2024

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to Anna University, Chennai)

TRICHY-621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"INVENTORY MANAGEMENT SYSTEM FOR CAR MANUFACTURES"** is the bonafide work of **SANTHOSH M (8115U23ME039)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE
Dr. T. M. NITHYA, M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
ASSOCIATE PROFESSOR
Department of CSE
K.Ramakrishnan College of Engineering (Autonomous)
Samayapuram–621112.

SIGNATURE
Mrs.S.RAJESWARI M.E.
SUPERVISOR
ASSISTANT PROFESSOR
Department of CSE
K.Ramakrishnan College of Engineering (Autonomous)
Samayapuram–621112.

Submitted for the End Semester Examination held on…….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on **"INVENTORY MANAGEMENT SYSTEM FOR CAR MANUFACTURES"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **MGB1201 – PYTHON PROGRAMMING**

.

Signature

_____

SANTHOSH M

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Engineering (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. M. NITHYA, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mrs.S.RAJESWARI M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE INSTITUTION

To achieve a prominent position among the top technical institutions

MISSION OF THE INSTITUTION

M1: To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.

M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.

M3: To provide education for developing high-quality professionals to transform the society.

VISION OF THE DEPARTMENT

To create eminent professionals of Computer Science and Engineering by imparting quality education.

MISSION OF THE DEPARTMENT

M1: To provide technical exposure in the field of Computer Science and Engineering through state of the art infrastructure and ethical standards.

M2: To engage the students in research and development activities in the field of Computer Science and Engineering.

M3: To empower the learners to involve in industrial and multi-disciplinary projects for addressing the societal needs.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

Our graduates shall

PEO1: Analyse, design and create innovative products for addressing social needs.

PEO2: Equip themselves for employability, higher studies and research.

PEO3: Nurture the leadership qualities and entrepreneurial skills for their successful career.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write

11. effective reports and design documentation, make effective presentations, and give and receive clear instructions.
12. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
13. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Apply the basic and advanced knowledge in developing software, hardware and firmware solutions addressing real life problems.

- **PSO2:** Design, develop, test and implement product-based solutions for their career enhancement.

ABSTRACT

The Inventory Management System for Car Manufacturers is a robust Python-based solution designed to streamline inventory tracking, enhance data accuracy, and reduce operational costs.By implementing this system, car manufacturers can achieve greater transparency in their inventory processes, minimize the risk of stockouts and overstocking, and ultimately improve their overall production efficiency. The project underscores the potential of Python in developing comprehensive, scalable solutions for complex industrial challenges. In the fast-paced automotive industry, managing inventory efficiently is crucial for manufacturers to meet production deadlines and optimize supply chain operations.

ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
The Inventory Management System for Car Manufacturers is a robust Python-based solution designed to streamline inventory tracking, enhance data accuracy, and reduce operational costs.By implementing this system, car manufacturers can achieve greater transparency in their inventory processes, minimize the risk of stockouts and overstocking, and ultimately improve their overall production efficiency. The project underscores the potential of Python in developing comprehensive, scalable solutions for complex industrial challenges. In the fast-paced automotive industry, managing inventory efficiently is crucial for manufacturers to meet production deadlines and optimize supply chain operations.	**PO1,PO2, PO3,PO12**	**PSO1**

Note: 1- Low, 2-Medium, 3- High

SUPERVISORHEAD OF THE DEPARTMENT

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	vi
1	**INTRODUCTION**	1
	1.1 Objective	1
	1.2 Overview	1
	1.3 Python Programming Concepts	2
2	**PROJECT METHODOLOGY**	3
	2.1 Proposed Work	3
	2.2 Block Diagram	4
3	**MODULE DESCRIPTION**	5
	3.1 Module 1 ADD INVENTORY ITEM	5
	3.2 Module 2 UPDATE INVENTORY ITEM	6
	3.3 **Module 3 DISPLAY INVENTORY ITEM**	7
4	**RESULTS AND DISCUSSION**	8
5	**CONCLUSION**	11
	REFERENCES	12
	APPENDIX	13

CHAPTER 1

INTRODUCTION

1.1 Objective

The Inventory Management System for Car Manufacturers is designed to optimize and streamline the inventory management processes within car manufacturing operations.Automate inventory tracking to minimize manual errors and save time, thereby improving overall operational efficiency.Provide real-time updates and ensure accurate inventory data to support informed decision-making.Maintain optimal stock levels to prevent both overstocking and stockouts, reducing carrying costs and avoiding production delays.Utilize historical data and predictive analytics to accurately forecast future inventory needs and plan accordingly

1.2 Overview

The Inventory Management System for Car Manufacturers is a comprehensive Python-based application designed to streamline and optimize the inventory management process within the automotive manufacturing industry. This system is intended to address the unique challenges faced by car manufacturers in managing their inventory effectively.Continuously monitors inventory levels, providing up-to-date information on stock availability.Automatically tracks stock levels, sending alerts for low inventory and reorder points to prevent stockouts and overstocking.

1.3 Python Programming Concepts

1. **Data Structures**: Using lists, dictionaries, and tuples to store and manage inventory data.
2. **File Handling**: Reading and writing data to files to save and load inventory information. This ensures that inventory data persists between program runs
3. **Functions**: Creating reusable functions to perform specific tasks, such as adding new inventory items, updating stock levels, and generating reports
4. **Classes and Objects**: Implementing object-oriented programming (OOP) principles to model inventory items and their attributes. This helps in organizing code and making it more modular and maintainable
5. **Database Integration**: Connecting to a database (e.g., SQLite, MySQL) to store and retrieve inventory data efficiently
6. **Object-Oriented Programming (OOP):** Encapsulating data and functionality in classes, such as User, Expense, and BudgetTracker, for a structured approach.
7. **Error Handling**: Implementing try-except blocks to manage potential errors and exceptions gracefully
8. **GUI Development**: Optionally, creating a graphical user interface (GUI) using libraries like Tkinter or PyQt to make the system more user-friendly.

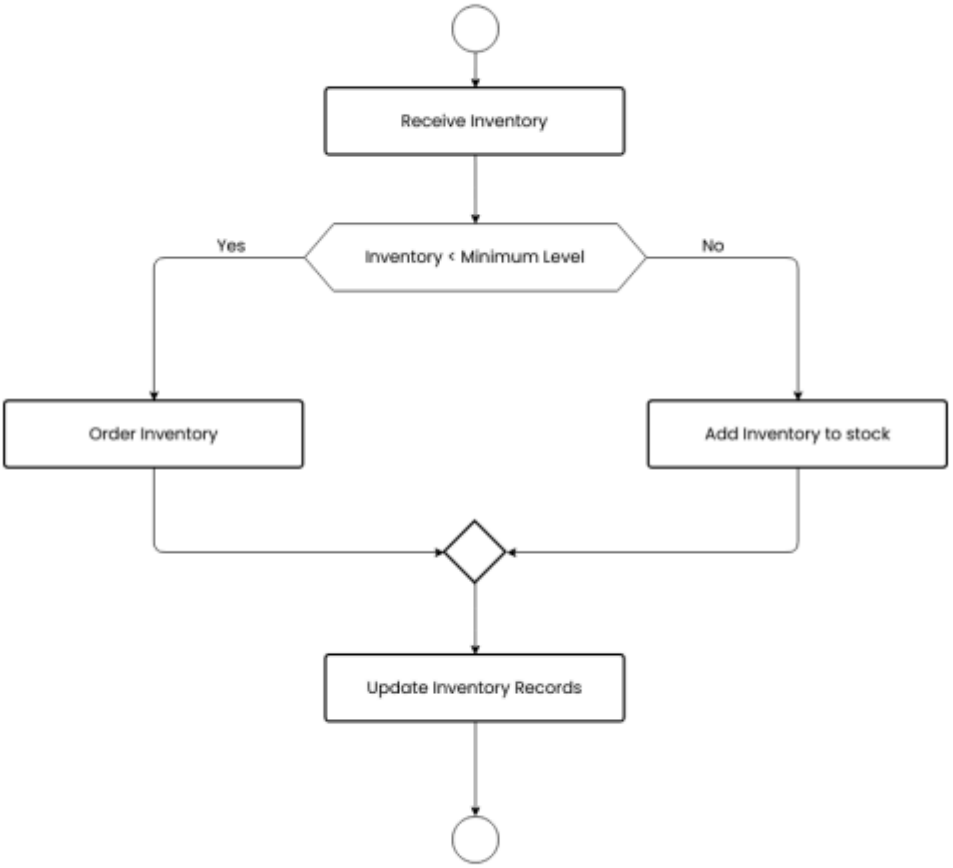
CHAPTER 2

PROJECT METHODOLOGY

2.1Proposed Work

1. **Data Modeling**: Design the database schema to support efficient data storage and retrieval. Define tables, relationships, and constraints.
2. **User Interface Design**: Develop UI/UX mockups and wireframes to ensure a user-friendly and intuitive interface. Focus on ease of navigation and usability.
3. **Development Environment Setup**: Set up the development environment, including version control (Git), necessary libraries, and frameworks.
4. **Core Functionality Implementation**: Develop the core functionality, such as adding, updating, and deleting inventory items, tracking stock levels, and generating reports.
5. **Database Integration**: Implement database connections and ensure efficient data handling. Use ORM (Object-Relational Mapping) for seamless interaction with the database.
6. **User Interface Development**: Build the graphical user interface using libraries like Tkinter or PyQt for a desktop application, or HTML/CSS/JavaScript for a web application.
7. **Simple Analytics:** Provide basic insights, such as total expenses per category, using dictionary operations or comprehensions.**System Architecture**: Define the overall system architecture, choosing suitable technologies and frameworks (e.g., Python, Flask/Django for web applications, SQLite/MySQL/PostgreSQL for databases).
8. **Documentation:** Provide user and developer documentation for system usage and future enhancement.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1 Module 1 : ADD INVENTORY ITEM

Function Name: `def add_car()`

Description: The add_car() function is likely used to add cars in inventory management system. Its purpose is to add a new cars record to the system. This function would typically require some input parameters, such as the id of an cars, After receiving this information, the function would process it and store the new add data in a suitable format within the program.

Steps:

- Add: The adding cars is used to add a new type of cars , expected to be in string.
- Category: The category of the adding cars, represented by a string (e.g., "brand", "popularity").

3.2 Module 2 : UPDATE INVENTORY ITEMS

Function Name : 'def update_car()'

Description:-Updates the cars from the input structure updates new records or collections and other specification.

Steps:

- Method Definition: Define the method update_car() within the new cars.
- Updating method : The updation method is the it can be mostly performed in the database like sql,or even in the code also.
- update items: If we add so many items in the programs it is much more uncomfortable and more difficult so it recommended in to update in the database.

3.3 Module 3 : DISPLAY INVENTORY ITEMS

Function Name : '**def display_car()**'

Description:- delete_car() is used to delete the inventory items if it is removed or can be deleted from the database.

Steps:

- Function Definition:Define the function delete_cars that deletes the inventory items in this code module.
- Check for Empty List: Check if the list is empty and print a message if there are no expenses.
- Print Details: It results it is found,when it is available in it.

CHAPTER 4

RESULTS AND DISCUSSION

PROGRAM

```
class Car:
    def __init__(self, car_id, make, model, year, quantity):
        self.car_id = car_id
        self.make = make
        self.model = model
        self.year = year
        self.quantity = quantity

    def __str__(self):
        return f"ID: {self.car_id}, Make: {self.make}, Model: {self.model}, Year:
            {self.year}, Quantity: {self.quantity}"

class Inventory:
    def __init__(self):
        self.cars = {}

    def add_car(self, car):
        if car.car_id in self.cars:
            print("Car ID already exists. Use update_car to update existing car.")
        else:
            self.cars[car.car_id] = car
            print("Car added to inventory.")

    def update_car(self, car_id, quantity):
        if car_id in self.cars:
            self.cars[car_id].quantity = quantity
            print("Car quantity updated.")
```

```
        else:
            print("Car ID not found in inventory.")

    def display_inventory(self):
        if not self.cars:
            print("Inventory is empty.")
        else:
            for car in self.cars.values():
                print(car)


inventory = Inventory()


car1 = Car(car_id=1, make="Toyota", model="Corolla", year=2020, quantity=10)
car2 = Car(car_id=2, make="Honda", model="Civic", year=2021, quantity=5)

inventory.add_car(car1)
inventory.add_car(car2)


inventory.update_car(car_id=1, quantity=15)


inventory.display_inventory()

```

OUTPUT

Output

```
Car added to inventory.
Car added to inventory.
Car quantity updated.
ID: 1, Make: Toyota, Model: Corolla, Year: 2020, Quantity: 15
ID: 2, Make: Honda, Model: Civic, Year: 2021, Quantity: 5

=== Code Execution Successful ===
```

CHAPTER 5

CONCLUSION

The Inventory Management System for car manufacturers developed using Python successfully automates the tracking and management of inventory. The system enhances efficiency by reducing manual errors, improving data accuracy, and facilitating real-time monitoring of stock levels. It integrates seamlessly with existing databases and provides a user-friendly interface for manufacturers. The project demonstrates the power of Python in creating robust and scalable solutions tailored to the specific needs of the automotive industry. Future enhancements may include incorporating machine learning algorithms for predictive inventory management and expanding the system to support multi-location operations.

REFERENCES:

1.**"Achieving Effective Inventory Management"** by John Schreibfeder - This book helps manufacturers and suppliers optimize their inventory systems to maximize profits

2.**"Inventory Management Explained" by David J. Piasecki** - This book provides a solid understanding of key planning aspects of inventory management, including stock sizing and ordering systems

3.**Inventory Management System on GitHub** - A collection of various inventory management system projects on GitHub, showcasing different approaches and implementations in Python.

APPENDIX

(Coding)

```
class Car:
def _init_(self, car_id, make, model, year, quantity):
self.car id = car id
self.make = make
self .model = model
self.year =  year
self. quantity = quantity
def _str_(self):
return f"ID: {self .car_id}, Make: {self. make},
{self-year}, Quantity: {self-quantity}"
class Inventory:
def _init_(self):
self.cars = {}
def add_car(self, car):
if car.car_id in self.cars:
print("Car ID already exists. Use update_car to update existing car. "
else:
self.cars[car.car_id] = car
print("Car added to inventory. ")
def update_car(self, car_id, quantity):
if car id in self.cars:
self.cars[car_id].quantity = quantity
print("Car quantity updated. ")
else:
print(“Car ID not found in inventory. ”)
```

```
def display_inventory(self):
if not self.cars:
print("inventory is empty. ")
else:
for car in self.cars.values():
print(car)
inventory = Inventory()
car1 = Car(car_id=1, make=”Toyota”, model=”Corolla”, year=2020, quantity=10)
car1 = Car(car_id=2, make=”Honda”, model=”Civic”, year=2021, quantity=5)
inventory.add_car(car1)
inventory.add_car(car2)
inventory. update_car (car_id=1, quantity=15)
inventory.display_inventory()
```

Car added to inventory.

Car added to inventory.

Car quantity updated.

ID: 1, Make: Toyota, Modal: Corolla, Year: 2020, Quantity: 15

ID: 2, Make: Honda, Modal: Civic, Year: 2021, Quantity: 5

=== YOUR PROGRAM HAS ENDED ===