

**A
DBMS Project REPORT
ON
"Inventory Management System"
OF
T.E.(AI & DS)
(Academic Year: 2023-2024)**

**SUBMITTED BY
Shivam Awasare
Roll No :- T1411004**

**GUIDED BY
Smruti Vyavahare**



**Department of AI & DS
Zeal Education Society's
Zeal College of Engineering & Research
Narhe, Pune-411041**

A Project REPORT
ON
”Inventory Management System”

Submitted By

Shivam Awasare
*in partial fulfilment for the award of the degree
of*

Bachelor of Engineering
of
Savitribai Phule Pune University

IN
AI & DS



Zeal College Of Engineering and Research, Narhe, Pune
2022 - 2023

Zeal Education Society's
Zeal College of Engineering & Research
Department of AI & DS



CERTIFICATE

This is to certify that DBMS Project entitled

"Inventory Management System"

have successfully completed by **"Shivam Awasare"** of TE (AI & DS) in the academic year 2023-2024 in partial fulfillment of the third Year of Bachelor degree in "AI & DS Engineering" as prescribed by the Savitribai Phule Pune University.

Smruti Vyavahare
Project Guide

Prof. D. D. Sarpate
HOD

Dr.A. M. Kate
Principal

Place: ZCOER, Pune.

Date: 31/10/2023

ACKNOWLEDGEMENT

We take this opportunity to thank our guide **Prof. Smruti Vyavahare** and Head of the Department **Prof. D. D. SARPATE** for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of AI & DS the Department for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

Shivam Awasare

Roll No. T1411004

TE

ABSTRACT

INVENTORY MANAGEMENT SYSTEM This Inventory Management DBMS mini-project aims to create a system for efficiently overseeing inventory within an organization. It offers a user-friendly interface for tracking stock levels, managing suppliers, generating reports, and automating alerts for reordering. The project focuses on database structuring, user interface design, inventory tracking, automated alerts, reporting, and security measures. By implementing relational databases, front-end frameworks, and backend programming, this mini-project demonstrates practical application of database management and software engineering principles in a real-world scenario. The system serves as a valuable tool for businesses to enhance inventory control, operational efficiency, and decision-making processes based on accurate, real-time inventory data.

Keyword - Inventory Management, DBMS, inventory tracking

Contents

1	Introduction	7
2	Features	8
3	Dependencies	9
4	Installation	10
4.1	Prerequisites	10
4.2	Installation Steps	10
5	Uses	11
6	Project Structure	12
7	Code	13
7.1	static	13
7.1.1	about_us.css	13
7.1.2	add_edit_product.css	14
7.2	description.css	15
7.2.1	favicon.ico	16
7.2.2	style.css	16
7.3	templates	18
7.3.1	about_us.html	18
7.3.2	add_product.html	19
7.3.3	base.html	19
7.3.4	description.html	20
7.3.5	edit_product.html	22
7.3.6	index.html	23
7.4	app.py	23
7.5	config.py	24
7.6	models.py	24
8	Output	26

1 Introduction

Inventory management is a critical aspect of any organization involved in buying, storing, and selling products. Efficiently handling inventory is essential for ensuring that the right products are available in the right quantities at the right time. In this context, the Inventory Management Database Management System (DBMS) mini-project aims to address the complexities and challenges inherent in managing inventory.

The primary goal of this project is to develop a comprehensive and user-friendly system that centralizes and automates inventory-related tasks. By leveraging database management principles, the project seeks to streamline inventory operations, including tracking stock levels, managing supplier information, processing purchase orders, and generating insightful reports for informed decision-making.

The system's functionalities include real-time tracking of inventory movement, automated alerts for low stock levels, and comprehensive reporting capabilities to provide insights into inventory turnover and trends.

Throughout the development of this mini-project, it will incorporate various technologies, such as relational databases for data management, front-end frameworks for a user-friendly interface, and backend programming languages for implementing system functionalities.

Ultimately, this Inventory Management DBMS mini-project serves as a practical demonstration of applying database management concepts and software engineering principles to address real-world inventory management challenges. By enhancing inventory control processes, improving operational efficiency, and enabling data-driven decision-making, the system aims to be a valuable tool for businesses striving to optimize their inventory management processes.

2 Features

These features collectively aim to provide a comprehensive solution for efficient inventory management, enabling businesses to optimize their inventory control processes, improve operational efficiency, and make informed decisions based on accurate and up-to-date inventory data.

1. **User-Friendly Interface:** Develop an intuitive user interface allowing easy access to inventory data, facilitating tasks such as adding new items, updating stock levels, and generating purchase orders.
 2. **Real-Time Inventory Tracking:** Implement a system to monitor and track inventory movements in real-time, providing accurate and updated stock levels and minimizing discrepancies.
 3. **Supplier Management:** Incorporate a feature to manage supplier information, including contact details, pricing, and delivery schedules to streamline communication and collaboration.
 4. **Comprehensive Reporting:** Develop reporting functionalities that provide insights into inventory turnover, trends, cost analysis.
 5. **Multi-location Inventory Management:** Facilitate managing inventory across multiple locations or warehouses, allowing for efficient transfer of stock between sites and maintaining accurate records.
-

3 Dependencies

requirements.txt

Python version

python \geq 3.9

Flask web framework

Flask==2.0.2

Flask-SQLAlchemy extension

Flask-SQLAlchemy==3.0

MySQL database connector

mysqlclient==2.0.3

It includes Flask for web development, Flask-SQLAlchemy for integration with a SQLAlchemy ORM within Flask, and the MySQL database connector (mysqlclient) to facilitate communication with a MySQL database. This is what we were dependent on to complete this mini project.

4 Installation

4.1 Prerequisites

- Ensure Python 3.9 or higher is installed on your system.
- Set up a MySQL database or have access to an existing one.

4.2 Installation Steps

1. Clone the project repository:

GitHub clone :- `https : //github.com/zargon01/Inventorymanagement`
Navigate to the project directory :- `cd inventory_management`

2. Install the project dependencies:

Navigate to the project root directory where the requirements.txt file is located
`pip install -r requirements.txt`

Requirements:- Python 3.9+ , Flask:Web framework for Python , Flask- SQLAlchemy: SQLAlchemy extension for Flask , MySQL database (you can replace this with your preferred database), flask==2.0.2 , Flask-SQLAlchemy==3.0 , mysqlclient==2.0.3

3. Set up Configuration:

Update the 'config.py' file with your MySQL database connection details:
`SQLALCHEMY_DATABASE_URI = 'mysql : //username : password@host/database_name'`

4. Run the Flask Application:

Set the Flask application entry point (app.py)
`export FLASK_APP=app.py` # For Unix/Linux#
`set FLASK_APP=app.py` # For Windows#

Run the Flask application
`flask run`

5. Access the Application:

Once the Flask application is running, open a web browser and navigate to '`http : //localhost : 5000`' (or the specified address) to access the Inventory Management system.

Following these steps will help set up the Inventory Management system with the specified dependencies, allowing you to run the application and manage your inventory through the provided Flask-based interface.

5 Uses

The Inventory Management system created with Python, Flask, Flask-SQLAlchemy, and MySQL database can be used for:

1. Business Inventory Control
2. Warehouse Management
3. Retail and E-commerce
4. Supply Chain Management
5. Manufacturing and Production
6. Data-Driven Decision Making
7. Automating Inventory Processes

It serves as a versatile tool for managing inventory, generating reports, predicting demand, and streamlining operations, benefiting various industries and businesses.

6 Project Structure

```
inventory_management_project/
├── app.py                    # The main Flask application script.
├── config.py                # Configuration settings for your Flask app.
├── models.py                # Defines the database models
├── static/                  # Static files like CSS, and images.
│   ├── style.css            # Global CSS styles.
│   ├── add_edit_product.css # CSS styles for add and edit product pages.
│   ├── description.css      # CSS styles for the "Description" page.
│   └── favicon.ico          # Favicon icon.
├── templates/               # HTML templates for your application.
│   ├── base.html            # Base template used for all pages.
│   ├── index.html           # Template for displaying the list of products.
│   ├── add_product.html     # Template for adding a product.
│   ├── edit_product.html    # Template for editing a product.
│   ├── description.html     # Template for the "Description" page.
│   └── about_us.html        # Template for the "About Us" page.
```

This structure organizes the Inventory Management project into various directories and files, adhering to the Flask application convention for templates, static files, and other essential components. This setup allows for a clear separation of concerns and ease of maintenance, with the Flask application (`app.py`), HTML templates, static files, database models (`models.py` using SQLAlchemy), and configuration settings (`config.py`) neatly organized within the project directory.

7 Code

7.1 static

7.1.1 about_us.css

```
1  /* Styling for the about us page */
2  .team-container {
3      display: flex;
4      flex-wrap: wrap;
5      justify-content: space-evenly;
6  }
7
8  .member {
9      flex: 0 1 calc(25% - 20px);
10     margin: 10px;
11     text-align: center;
12     background-color: #f7f7f7;
13     border-radius: 5px;
14     padding: 10px;
15     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
16 }
17
18 .member img {
19     max-width: 60%;
20     border-radius: 100%;
21 }
22
23 .member-description {
24     margin-top: 10px;
25     font-size: 14px;
26     color: #333;
27 }
28
29 /* Center the project guide's photo */
30 .project-guide img {
31     display: block;
32     margin: 0 auto;
33     max-width: 70%;
34     border-radius: 100%;
35 }
36 }
37
38 /* Style for project guide's description */
39 .project-guide-description {
40     margin-top: 10px;
41     font-size: 16px;
42     font-weight: bold;
43     color: #f394ba;
44     text-align: center;
45 }
```

7.1.2 add_edit_product.css

```
1  /* Add and edit product styles */
2
3  body {
4      font-family: 'Roboto', sans-serif;
5      background-color: #F7F7F7;
6      margin: 0;
7      padding: 0;
8  }
9
10 h2{
11     color: #333333;
12 }
13
14 header {
15     background-color: #FFDFDF;
16     color: white;
17     text-align: center;
18     padding: 10px;
19 }
20
21
22 .content {
23     width: 80%;
24     margin: 20px auto;
25     background-color: #FFF6F6;
26     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
27     padding: 20px;
28     border-radius: 10px;
29     text-align: center;
30 }
31
32 /* Table styles */
33 table {
34     width: 100%;
35     border-collapse: collapse;
36 }
37
38 table th, table td {
39     padding: 10px;
40     border: 1px solid #F7F7F7;
41     text-align: left;
42 }
43
44 /* Button styles */
45 .button {
46     background-color: #f394ba;
47     color: white;
48     border: none;
49     padding: 10px 20px;
50     text-align: center;
51     display: inline-block;
52     font-size: 16px;
```

```
53     margin: 10px;
54     cursor: pointer;
55     border-radius: 5px;
56 }
57
58 .button:hover {
59     background-color: #f288b2;
60 }
61
62 #inputbox{
63     border-radius: 50px;
64     background-color: white;
65     border-color: white;
66     text-align: center;
67 }
```

7.2 description.css

```
1
2
3 /* Add your custom styles here */
4 .heading {
5     text-align: center;
6     color: white;
7     padding: 10px;
8 }
9
10 .page-content {
11     background-color: #FFF6F6;
12     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
13     padding: 20px;
14     border-radius: 10px;
15     text-align: left;
16     margin-left: 50px;
17 }
18
19 .table-of-contents {
20     background-color: #FFDFDF;
21     color: rgb(0, 0, 0);
22     padding: 10px;
23     border-radius: 5px;
24 }
25
26 .table-of-contents ol {
27     list-style-type: decimal;
28     padding-left: 20px;
29 }
30
31 .project-structure pre {
32     white-space: pre-wrap;
33     background-color: #f7f7f7;
34     padding: 15px;
35     border-radius: 5px;
36 }
37
```

```
38 .project-structure pre code {
39     font-family: 'Courier New', Courier, monospace;
40     font-size: 14px;
41 }
42
43 .how-to-use, .features, .technologies, .dependencies, .use-cases, .project-
    structure {
44     margin-top: 20px;
45     padding: 15px;
46     background-color: #f7f7f7;
47     border-radius: 5px;
48 }
49
50 .how-to-use h3, .features h3, .technologies h3, .dependencies h3, .use-cases h3, .
    project-structure h3 {
51     background-color: #FFDFDF;
52     color: black;
53     padding: 10px;
54     border-radius: 5px 5px 0 0;
55 }
```

7.2.1 favicon.ico

7.2.2 style.css

```
1  /* Global styles */
2  body {
3      font-family: 'Roboto', sans-serif;
4      background-color: #F7F7F7;
5      margin: 0;
6      padding: 0;
7  }
8
9  h1{
10     color: #333333;
11 }
12
13 header {
14     background-color: #FFDFDF;
15     color: white;
16     text-align: center;
17     padding: 10px;
18 }
19
20 nav {
21     justify-self: center;
22     background-color: #d6acac;
23     margin-top: 10px;
24     padding: 8px;
25     border-radius: 50px;
26 }
27
28 nav a {
29     color: rgb(19, 17, 17);
30     text-decoration: none;
```



```
31     padding: 10px;
32     align-items: center;
33 }
34
35 nav a:hover {
36     text-decoration: underline;
37
38 }
39
40
41 /* Center the navigation bar elements */
42 .center-nav {
43     text-align: center;
44     padding: 8px 0; /* Adjust the padding as needed */
45 }
46
47 .center-nav a {
48     display: inline-block;
49     margin: 0 10px; /* Adjust the margin to control spacing between navigation
50 links */
51 }
52
53
54 .content {
55     width: 80%;
56     margin: 20px auto;
57     background-color: #FFF6F6;
58     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
59     padding: 20px;
60     border-radius: 10px;
61 }
62
63 /* Table styles */
64 table {
65     width: 100%;
66     border-collapse: collapse;
67 }
68
69 table th, table td {
70     padding: 10px;
71     border: 1px solid #e56666;
72     text-align: left;
73 }
74
75 /* Button styles */
76 .button {
77     background-color: #f394ba;
78     color: white;
79     /*border: none;*/
80     padding: 10px 20px;
81     text-align: center;
82     display: inline-block;
83     font-size: 16px;
```

```
84     margin: 10px;
85     /*cursor: pointer;*/
86     border-radius: 5px;
87 }
88
89 .button:hover {
90     background-color: #f288b2;
91 }
```

7.3 templates

7.3.1 about_us.html

```
1 {% extends "base.html" %}
2
3 {% block content %}
4     <div class="heading">
5         <h2>About Us</h2>
6     </div>
7
8     <div class="team-container">
9         <div class="member project-guide">
10             
11             <div class="project-guide-description">
12                 <p>Prof. Smruti Vyavahare</p>
13                 <p>Project Guide</p>
14             </div>
15         </div>
16     </div>
17     <div class="team-container">
18         <div class="member">
19             
20             <div class="member-description">
21                 <p>Shivam Awasare</p>
22                 <p>T1411004</p>
23             </div>
24         </div>
25
26         <div class="member">
27             
28             <div class="member-description">
29                 <p>Lavanya Moolya</p>
30                 <p>T1411050</p>
31             </div>
32         </div>
33     </div>
34     <div class="team-container">
35         <div class="member">
36             
37             <div class="member-description">
38                 <p>Avishkar Karche</p>
39                 <p>T1411036</p>
40             </div>
41         </div>
42     </div>
```



```
17 </header>
18 <nav class= "center-nav">
19 <a href="/">Home</a>
20 <a href="/description">Description</a>
21 <a href="/about_us">About Us</a>
22
23 </nav>
24
25 <div class="content">
26 {% block content %}{% endblock %}
27 </div>
28 </body>
29 </html>
```

7.3.4 description.html

```
1 {% extends "base.html" %}
2
3 {% block content %}
4 <div class="heading">
5 <h2>Project Description</h2>
6 </div>
7
8
9 <div class="page-content">
10 <div class="table-of-contents">
11 <h3>Table of Contents</h3>
12 <ol>
13 <li><a href="#how-to-use">How to Use Application</a></li>
14 <li><a href="#features">Features</a></li>
15 <li><a href="#project-structure">Project Structure</a></li>
16 <li><a href="#technologies">Concepts and Technologies Used</a></li>
17 >
18 <li><a href="#dependencies">Dependencies</a></li>
19 <li><a href="#use-cases">Use Cases</a></li>
20 </ol>
21 </div>
22 <div id="how-to-use" class="how-to-use">
23 <h3>How to Use Application</h3>
24 <p>
25 To use the Inventory Management application, follow these steps:
26 </p>
27 <ol>
28 <li>Access the application by visiting the home page.</li>
29 <li>View the list of products, including their names, descriptions
30 , and quantities.</li>
31 <li>To add a new product, click the "Add Product" link and provide
32 the necessary details.</li>
33 <li>To edit an existing product, click the "Edit" link next to the
34 product, and update the information.</li>
35 <li>To delete a product, click the "Delete" link next to the
36 product.</li>
37 </ol>
38 </div>
39 <div id="features" class="features">
```

```
35     <h3>Features</h3>
36     <p>
37         The Inventory Management application offers the following features
38     :
39     </p>
40     <ul>
41         <li>View a list of products with their names, descriptions, and
quantities.</li>
42         <li>Add new products with details.</li>
43         <li>Edit existing product information.</li>
44         <li>Delete products when needed.</li>
45     </ul>
46 </div>
47 <div id="project-structure" class="project-structure">
48     <h3>Project Structure</h3>
49     <br><p>The project structure of Inventory Management is as follows: </
p><br>
50     <pre>
51         inventory_management_project/
52
53             app.py                                # The main
Flask application script.
54             config.py                             #
Configuration settings for your Flask app.
55             models.py                             # Defines
the database models
56
57             static/                               # Static
files like CSS, and images.
58             style.css                             # Global
CSS styles.
59             add_edit_product.css                  # CSS
styles for add and edit product pages.
60             description.css                       # CSS
styles for the "Description" page.
61             favicon.ico                           # Favicon
icon.
62
63             templates/                            # HTML
templates for your application.
64             base.html                             # Base
template used for all pages.
65             index.html                            # Template
for displaying the list of products.
66             add_product.html                     # Template
for adding a product.
67             edit_product.html                    # Template
for editing a product.
68             description.html                     # Template
for the "Description" page.
69             about_us.html                        # Template
for the "About Us" page.
70     </pre>
```

```

    </div>

    <div id="technologies" class="technologies">
        <h3>Concepts and Technologies Used</h3>
        <p>
            The Inventory Management application uses the following
            technologies and concepts:
        </p>
        <ul>
            <li>Flask: A Python web framework for building web applications.</li>
            <li>SQLAlchemy: An Object-Relational Mapping (ORM) library for
            database interaction.</li>
            <li>MySQL: The database used for data storage.</li>
            <li>HTML and CSS: For creating the user interface and styling the
            application.</li>
        </ul>
    </div>
    <div id="dependencies" class="dependencies">
        <h3>Dependencies</h3>
        <p>
            The application relies on the following dependencies:
        </p>
        <ul>
            <li>Flask: Web framework for Python.</li>
            <li>Flask-SQLAlchemy: SQLAlchemy integration for Flask.</li>
            <li>MySQL: The database management system.</li>
        </ul>
    </div>
    <div id="use-cases" class="use-cases">
        <h3>Use Cases</h3>
        <p>
            The Inventory Management application can be useful in various
            scenarios, including:
        </p>
        <ul>
            <li>Small businesses managing their product inventory.</li>
            <li>Retail stores keeping track of available products.</li>
            <li>Warehouses monitoring stock levels.</li>
        </ul>
    </div>
</div>
{% endblock %}

```

[illegible]

7.3.6 index.html

7.4 app.py

ZCOER, Dept of AI & DS

```
5 app = Flask(__name__, static_url_path='/static', static_folder='static')
6 app.config.from_object(Config)
7 db.init_app(app)
8 app.secret_key = 'supersecretkey'
9
10 @app.route('/')
11 def index():
12     products = Product.query.all()
13     return render_template('index.html', products=products)
14
15 @app.route('/add_product', methods=['GET', 'POST'])
16 def add_product():
17     if request.method == 'POST':
18         name = request.form['name']
19         description = request.form['description']
20         quantity = request.form['quantity']
21
22         product = Product(name, description, quantity)
23         db.session.add(product)
24         db.session.commit()
25         flash('Product added successfully', 'success')
26         return redirect(url_for('index'))
27
28     return render_template('add_product.html')
29
30 @app.route('/edit_product/<int:id>', methods=['GET', 'POST'])
31 def edit_product(id):
32     product = Product.query.get(id)
33
34     if request.method == 'POST':
35         product.name = request.form['name']
36         product.description = request.form['description']
37         product.quantity = request.form['quantity']
38
39         db.session.commit()
40         flash('Product updated successfully', 'success')
41         return redirect(url_for('index'))
42
43     return render_template('edit_product.html', product=product)
44
45 @app.route('/delete_product/<int:id>')
46 def delete_product(id):
47     product = Product.query.get(id)
48     db.session.delete(product)
49     db.session.commit()
50     flash('Product deleted successfully', 'success')
51     return redirect(url_for('index'))
52
53 if __name__ == '__main__':
54     with app.app_context():
55         db.create_all()
56     app.run(debug=True)
```

7.5 config.py

```
1 class Config:
2     SQLALCHEMY_DATABASE_URI = 'mysql://root:root@localhost/Inventory_Management'
3     SQLALCHEMY_TRACK_MODIFICATIONS = False
```

7.6 models.py

```
1 from flask_sqlalchemy import SQLAlchemy
```



```
2
3 db = SQLAlchemy()
4
5 class Product(db.Model):
6     id = db.Column(db.Integer, primary_key=True)
7     name = db.Column(db.String(100), nullable=False)
8     description = db.Column(db.String(200))
9     quantity = db.Column(db.Integer, default=0)
10
11     def __init__(self, name, description, quantity):
12         self.name = name
13         self.description = description
14         self.quantity = quantity
```

8 Output

Inventory Management

[Home](#) [Description](#) [About Us](#)

Product List

Add Product

Name	Description	Quantity	Action
Lays	Tomato	354	<div>Edit Delete</div>
Parle-G	Biscuit	241	<div>Edit Delete</div>
Cold Drink	CocaCola	453	<div>Edit Delete</div>

Inventory Management

[Home](#) [Description](#) [About Us](#)

Add Product

Name:

Description:

Quantity:

Add Product