KISHKINDA UNIVERSIT

BELLARI

CONCESSION SALES TRACKER POC

STUDENT NAME :- V SHASHIDHARA

REG NUMBER :- TEMPBTECH-EEE120

DEPARTMENT NAME :- ELECTRICAL &ELCTRONICS ENGG

YEAR OF STUDY :- 2024-2025

DATA OF SUB :-

TABLE OF CONTENTS:

1. INTRODUCTION:

The Concession Sales Tracker is designed to streamline and enhance the management of concession sales across various events. This Proof of Concept (PoC) aims to demonstrate the feasibility and effectiveness of the proposed solution, showcasing its capabilities in tracking sales, inventory, and customer engagement in real-time.

T0 efficiently monitor and analyze concession sales data, enabling better inventory management, sales forecasting, and improved customer service.

This outline serves as a starting point for developing a more detailed poc. Would you like more information on any specified part.

- Time of sale

- Salesperson name

- Payment method (cash, credit, etc.)

- Inventory remaining

The Proof of Concept (POC) for the Sales Tracker aims to demonstrate a scalable and efficient solution for monitoring

and analyzing sales data in real-time. This POC outlines the core functionalities designed to streamline sales processes across various events and locations.

Key components include intuitive dashboards for visualizing sales trends, category breakdowns, and payment methods, all of which allow users to derive actionable insights. Integration capabilities with existing point-of-sale systems ensure seamless data flow and accessibility, while user access controls maintain security and data integrity.

Key features include customizable data fields, integration capabilities with existing POS systems, and user-friendly access controls. Additionally, the Sales Tracker supports trend analysis over time, allowing for performance evaluations and identification of top-selling items. This tool not only enhances operational efficiency but also fosters a data-driven culture, ultimately driving revenue growth and improving customer satisfaction.

Provide comprehensive reporting features that allow users to analyze sales metrics, identify trends, and generate insights for strategic planning.

The Sales Tracker is a robust tool designed to streamline the monitoring and analysis of sales data across various channels. This system facilitates real-time data entry, enabling users to capture vital sales metrics, including date, product categories, quantities sold, and total revenue. By employing intuitive dashboards and visual reporting tools, the Sales Tracker transforms complex.

2. OBJECTIVE OF PROJECT:

To develop a system that tracks concession sales in real-time, providing insights into inventory, sales trends, and customer preferences.

Components of the POC

User Roles: Identify users (e.g., managers, sales staff) and their access levels.

Key Features:

- Sales tracking (real-time)
- Inventory management
- Reporting and analytics
- User-friendly interface
- Choose Technology Stack

- Frontend: Consider using frameworks like React or Angular for a responsive interface.
- Backend: Node.js or Python with Flask/Django for handling requests.
- Database: SQLite or PostgreSQL for data storage.
- Deployment: Heroku, AWS, or a similar service for hosting.
- Design the User Interface
- Dashboard: A visual summary of sales, inventory levels, and trends.

- Sales Entry Form: Simple form to input sales data.
- Reports Section: Area to generate and view reports.
- Database Schema
- Tables:
- Products: ID, name, category, price, stock level.
- Sales: ID, product ID, quantity sold, timestamp.
- Users: ID, name, role, login credentials.
- Implement Core Functionality
- Sales Recording: Allow users to log sales through the interface.
- Inventory Updates: Automatically adjust inventory levels based on sales.
- Data Retrieval: Create endpoints for fetching sales and inventory data.
- Testing
- Conduct unit tests for individual components.
- Perform integration testing to ensure the system works as a whole.
- Gather feedback from potential users to identify any issues or additional needs.

DemonstrationPrepare a short presentation showcasing the key features.Include a live demo of the application, highlighting how to enter sales and view reports.

3. METHODOLOGY:

Research and Requirements Gathering Stakeholder Interviews: Engage with potential users (e.g., concession staff, managers) to understand their needs and pain points.

1.Competitor Analysis: Review existing concession tracking solutions to identify strengths and weaknesses.

2. Define Scope: List core features (e.g., sales entry, inventory management) and prioritize them based on user needs.

Limitations: Identify any constraints (e.g., budget, timeline) to manage expectations.

3. Design Phase: Create low-fidelity wireframes to outline the user interface and flow.

4. Development Phase: Choose appropriate technologies (e.g., React for frontend, Node.js for backend, MongoDB for database).

5. Testing Phase: Test individual components to ensure they work as intended. Ensure that all parts of the system work together seamlessly.

User Acceptance Testing (UAT): Allow real users to test the application and provide feedback on functionality and usability.

6. Deployment: Choose a cloud service (e.g., AWS, Heroku) for hosting the application.Deploy the application and ensure

that all services are running smoothly.Set up monitoring tools to track performance and user engagement

4. RESULT/FINDING :

- INPUT:

```
class SalesTracker:
def _init_(self, filename='sales_data.json'):
        self.filename = filename
        self.load_sales()
def load_sales(self):
        try:
       with open(self.filename, 'r') as file:
```

```
            self.sales_data = json.load(file)
    except FileNotFoundError:
            self.sales_data = []
def save_sales(self):
        with open(self.filename, 'w') as file:
            json.dump(self.sales_data, file, indent=4)
def add_sale(self, date, event, location, product, quantity, unit_price, payment_method):
        total_price = quantity * unit_price


sale_record = {
            'date': date,
            'event': event,
            'location': location,
            'product': product,
            'quantity': quantity,
            'unit_price': unit_price,
            'total_price': total_price,
            'payment_method': payment_method
        }
```

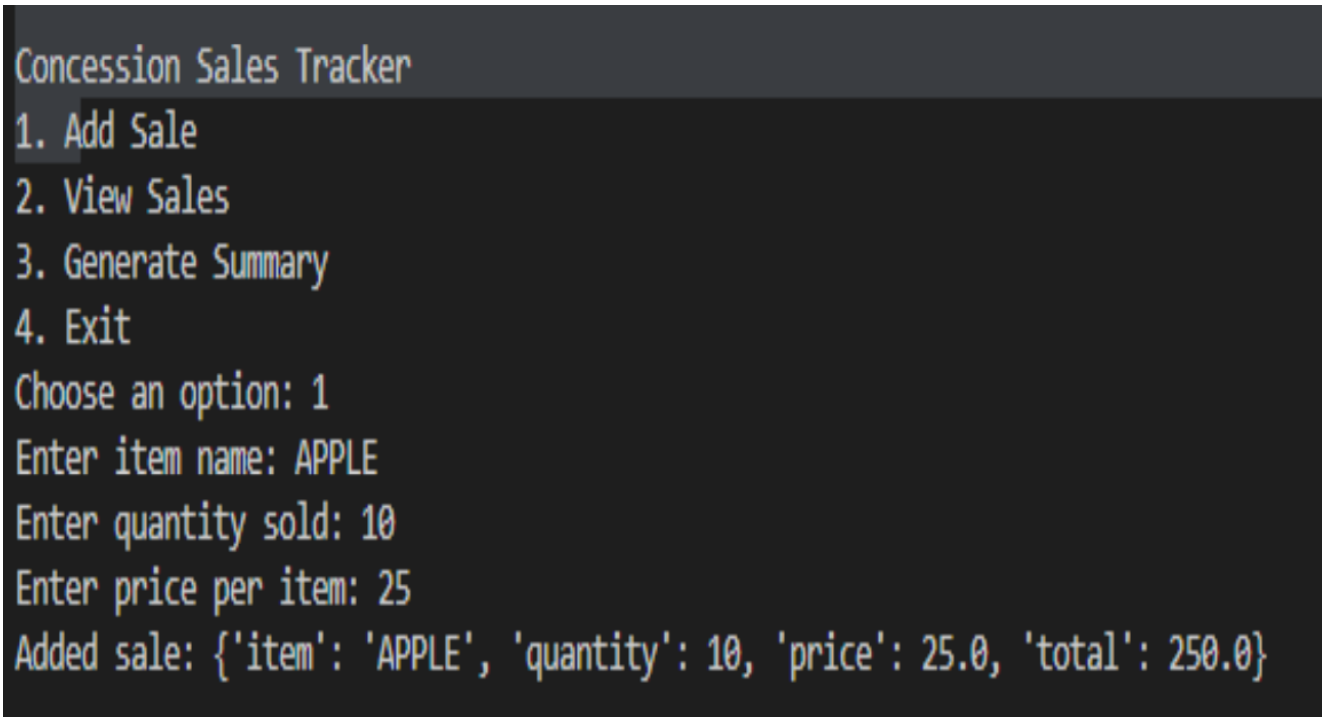
```
self.sales_data.append(sale_record)
 self.save_sales()
 print("Sale added successfully.")
def view_sales(self):
        if not self.sales_data:
            print("No sales data found.")
            return
        print("\nSales Records:")
for sale in self.sales_data:
            print(sale)
 def generate_report(self):
        total_sales = sum(sale['total_price'] for sale in self.sales_data)
        product_sales = {}
 for sale in self.sales_data:
            product = sale['product']
            product_sales[product] = product_sales.get(product, 0) + sale['quantity']
        print(f"\nTotal Sales Amount: ${total_sales:.2f}")
        print("Sales by Product:")
def main():
```

```
    tracker = SalesTracker()
while True:
        print("\nConcession Sales Tracker")
        print("1. Add Sale")
        print("2. View Sales")
        print("3. Generate Sales Report")
        print("4. Exit")


choice = input("Select an option: ")
if choice == '1':
            date = input("Enter date (YYYY-MM-DD): ")
            event = input("Enter event name: ")
            location = input("Enter location: ")
            product = input("Enter product name: ")
            quantity = int(input("Enter quantity sold: "))
            unit_price = float(input("Enter unit price: "))
            payment_method = input("Enter payment method: ")
            tracker.add_sale(date, event, location, product, quantity, unit_price, payment_method)
        elif choice == '2':
```

```
        tracker.view_sales()
    elif choice == '3':
        tracker.generate_report()
    elif choice == '4':
        print("Exiting the tracker.")
    else:
      if _name_ == "_main_":
                main()
```

- OUTPUT:



5. CONCLUSION :

The Concession Sales Tracking project will provide a comprehensive solution for managing concession sales, improving operational efficiency and enabling data-driven decision-making. By focusing on essential features and user-friendly design, this project will enhance the overall experience of tracking sales at events.

FUTURE WORK :

1. Data Persistence

2. User Interface Enhancements

3. Advanced Reporting

4. Data Visualization

5. Inventory Management

6. User Authentication

7. Mobile Compatibility

8. Sales Forecasting

6. REFERNCES :

Here are some references and resources that can help you further develop your concession sales tracker POC:

Books

1. "Python Crash Course" by Eric Matthes
2. "Fluent Python" by Luciano Ramalho

Online Tutorials

1. Real Python(Real Python Tutorials)
2. W3Schools Python Tutorial

Python Libraries

1. Pandas

2. Matplotlib
3. Tkinter(Tkinter Documentation)
4. Flask

Video Courses

1. Coursera:
2. Udemy:

Forums and Communities

1. Stack Overflow
2. Reddit

Sample Projects

1. GitHub Repositories
    - Search for similar projects on GitHub to learn from existing code and best practices.
2. Kaggle Datasets
    - Explore datasets that could be relevant for analysis and practice: Kaggle Datasets