

SmartStock Inventory Optimization for Retail Stores

Internship Platform: Infosys Springboard

Student Name: Kaparapu Veera Venkata Mahendra

Milestone: 4 – Final Dashboard

1. Introduction

This milestone represents the final stage of the SmartStock project, building directly upon the functional inventory optimization logic and dashboard created in Milestone 3. The primary focus of Milestone 4 was to completely redesign the User Interface (UI) and enhance the User Experience (UX).

The objective was to transform the functional prototype into a polished, professional, and intuitive application. This involved implementing a custom 'modern & airy' color palette (off-white, teals, and blues), adding advanced CSS for styling, and reorganizing the entire application into a multi-tab interface to improve navigation and clarity for the end-user.

2. Objectives

- UI Redesign: Implement a custom, professional color palette and apply it to all visual components using Streamlit's CSS injection.
- Tabbed Navigation: Re-organize the dashboard from a single-page application into a multi-tab interface for Forecast Visualization, Inventory Planning, Stock Alerts, and Reports.
- New Feature (Forecast Visualization): Create a new tab to visualize historical Actual Sales vs Forecasted Sales.
- New Feature (Stock Alerts): Add a Stock Alerts tab that simulates inventory levels and flags Reorder or OK statuses.
- New Feature (Data Refresh): Implement a file uploader to allow users to upload new sales data.
- Advanced Styling: Apply custom styling to all Streamlit elements for a cohesive and modern look.

3. Implementation & Code

```
# dashboard.py - Milestone 4: Final Streamlit Dashboard
```

```
import pandas as pd
import numpy as np
import streamlit as st
import matplotlib.pyplot as plt
import os

# --- UPDATED: Palette with new sidebar and button colors ---
PALETTE = {
```

```
"background": "#F6F6F2",      # Off-white background
"text": "#31333F",          # Standard dark text
"heading_text": "#388087",    # Darkest Teal for headings
"primary_accent": "#6FB3B8", # Main Teal for charts/highlights
"secondary_accent": "#BADFE7", # Light Blue for contrast
"sidebar_bg": "#E9EEF2",     # NEW: Light Gray for Sidebar
"button_blue": "#4C8BF5",    # NEW: Modern Soft Blue for buttons
"white": "#FFFFFF",
"light_gray": "#E0E0E0",     # For subtle borders
"table_header": "#F0F2F6",    # Background for table headers
"table_row_alt": "rgba(233, 238, 242, 0.5)" # Semi-transparent alt row color
}
```

--- UPDATED: CSS Styling with all requested changes ---

```
st.markdown(f"""
<style>
/* Main app background */
.stApp {{
    background-color: {PALETTE['background']};
}}
```



```
/* General text and heading colors */
.stApp, p {{ color: {PALETTE['text']}; }}
h1, h2, h3 {{ color: {PALETTE['heading_text']}; }}
```

```
/* --- NEW: Sidebar color --- */
.st-emotion-cache-16txtl3 {{
    background-color: {PALETTE['sidebar_bg']};
}}
```

```
.st-emotion-cache-16txtl3 h2, .st-emotion-cache-16txtl3 p, .st-emotion-cache-16txtl3 small {{  
  color: {PALETTE['heading_text']};  
}  
  
/* --- NEW: Dropdown list styling --- */  
  
[data-baseweb="select"] ul {{  
  background-color: {PALETTE['white']};  
}  
  
[data-baseweb="select"] ul li:hover {{  
  background-color: {PALETTE['secondary_accent']};  
}  
  
[data-baseweb="select"] ul li span {{  
  color: {PALETTE['primary_accent']};  
  font-weight: 500;  
}  
  
/* --- NEW: Download button color --- */  
  
.stDownloadButton>button {{  
  color: {PALETTE['white']};  
  background-color: {PALETTE['button_blue']};  
  border: 1px solid {PALETTE['button_blue']};  
}  
  
.stDownloadButton>button:hover {{  
  background-color: #3a7bf0;  
  border: 1px solid #3a7bf0;  
}  
  
/* --- NEW: Transparent text box for st.info --- */  
  
[data-testid="stAlert"] {{
```

```
background-color: rgba(233, 238, 242, 0.7) !important; /* Semi-transparent background */
border: 1px solid {PALETTE['secondary_accent']} !important;
border-radius: 0.5rem !important;
padding: 1rem !important;
}

[data-testid="stAlert"] .st-emotion-cache-1wivap2 {{
  color: {PALETTE['heading_text']} !important; /* Text color inside the box */
}

[data-testid="stAlert"] svg {{
  display: none !important; /* Hides the default icon */
}

/* --- NEW: Table/DataFrame Styling --- */

[data-testid="stDataFrame"] {{
  border: none;
  box-shadow: 0 2px 4px rgba(0,0,0,0.05);
  border-radius: 0.5rem;
  overflow: hidden; /* Ensures child elements conform to border radius */
}

[data-testid="stDataFrame"] .col_heading {{ /* Table header */
  background-color: {PALETTE['table_header']};
  color: {PALETTE['heading_text']};
  font-weight: bold;
  text-align: left !important;
}

[data-testid="stDataFrame"] tbody tr {{ /* Table rows */
  background-color: {PALETTE['white']};
}

[data-testid="stDataFrame"] tbody tr:nth-child(even) {{ /* Alternating row color */
```

```

background-color: {PALETTE['table_row_alt']};

}

[data-testid="stDataFrame"] tbody tr td {{ /* Table cells */ 

color: {PALETTE['text']}; 

border-bottom: 1px solid {PALETTE['light_gray']}; 

}

</style>

"""", unsafe_allow_html=True)

st.set_page_config(page_title="Smart Inventory Dashboard", layout="wide")

st.title("📊 Smart Inventory Management Dashboard")

try:

df = pd.read_csv("data/forecast_results.csv")

df['Date'] = pd.to_datetime(df['Date'])

except FileNotFoundError:

    st.error("⚠️ 'forecast_results.csv' not found! Please make sure the data file is in the 'data' sub-directory.")

    st.stop()

# Sidebar for User Inputs

st.sidebar.header("Inventory Parameters")

lead_time = st.sidebar.slider("Lead Time (Days)", 1, 30, 7)

ordering_cost = st.sidebar.slider("Ordering Cost ($)", 10, 200, 50)

holding_cost = st.sidebar.slider("Holding Cost ($ per unit per month)", 1.0, 20.0, 2.0, 0.5)

service_level_option = st.sidebar.selectbox("Service Level", ["90%", "95%", "99%"], index=1)

service_level_z = {"90%": 1.28, "95%": 1.65, "99%": 2.33}[service_level_option]

#Main Page Tabs

```

```
tab1, tab2, tab3, tab4 = st.tabs(["📈 Forecast Visualization", "📋 Inventory Planning", "🔔 Stock Alerts", "📝 Reports"])
```

```
#Tab 1: Forecast Visualization
```

```
with tab1:
```

```
    st.header("Sales Forecast Visualization")
```

```
    selected_product = st.selectbox("Select Product to View Forecast", df["Product_ID"].unique())
```

```
if selected_product:
```

```
    product_data = df[df["Product_ID"] == selected_product].sort_values('Date')
```

```
    fig, ax = plt.subplots(figsize=(10, 5))
```

```
        ax.plot(product_data["Date"], product_data["Forecasted_Sales"], label="Forecasted Sales",  
color=PALETTE['primary_accent'], marker='o', markersize=4, linestyle='--')
```

```
        ax.plot(product_data["Date"], product_data["Actual_Sales"], label="Actual Sales",  
color=PALETTE['heading_text'], marker='x', markersize=4, linestyle=':')
```

```
    ax.set_title(f"Sales Forecast for Product: {selected_product}", fontsize=16, color=PALETTE['heading_text'])
```

```
    ax.set_xlabel("Date", fontsize=12, color=PALETTE['text'])
```

```
    ax.set_ylabel("Sales Quantity", fontsize=12, color=PALETTE['text'])
```

```
    ax.legend(labelcolor=PALETTE['text'])
```

```
    ax.grid(True, which='both', linestyle='--', linewidth=0.5, color=PALETTE['light_gray'])
```

```
    ax.tick_params(axis='x', colors=PALETTE['text'])
```

```
    ax.tick_params(axis='y', colors=PALETTE['text'])
```

```
    fig.patch.set_facecolor(PALETTE['white'])
```

```
    ax.set_facecolor(PALETTE['white'])
```

```
    plt.xticks(rotation=45)
```

```

plt.tight_layout()
st.pyplot(fig)

#Tab 2: Inventory Planning

with tab2:
    st.header("Inventory Optimization Plan")
    inventory_plan = []
    for p_id in df["Product_ID"].unique():
        product_df = df[df["Product_ID"] == p_id]

        total_demand = product_df["Forecasted_Sales"].sum()
        avg_daily_demand = product_df["Forecasted_Sales"].mean()
        std_dev_demand = product_df["Forecasted_Sales"].std()

        eoq = np.sqrt((2 * total_demand * ordering_cost) / (holding_cost / 30)) if holding_cost > 0 else 0
        safety_stock = service_level_z * std_dev_demand * np.sqrt(lead_time) if not pd.isna(std_dev_demand) else 0
        reorder_point = (avg_daily_demand * lead_time) + safety_stock

        inventory_plan.append({
            "Product ID": p_id,
            "Avg Daily Demand": f"{avg_daily_demand:.2f}",
            "Std Dev of Demand": f"{std_dev_demand:.2f}" if not pd.isna(std_dev_demand) else "0.00",
            "Economic Order Quantity (EOQ)": f"{eoq:.0f}",
            "Safety Stock": f"{safety_stock:.0f}",
            "Reorder Point": f"{reorder_point:.0f}"
        })

    inventory_df = pd.DataFrame(inventory_plan)
    st.dataframe(inventory_df, use_container_width=True)

```

```
#Tab 3: Stock Status & Alerts
```

```
with tab3:
```

```
    st.header("Current Stock Status and Reorder Alerts")
```

```
    alert_df = pd.DataFrame(inventory_plan).rename(columns={"Product ID": "Product"})
```

```
    alert_df["ReorderPoint"] = pd.to_numeric(alert_df["Reorder Point"])
```

```
    if not alert_df.empty:
```

```
        alert_df["CurrentStock"] = np.random.randint(50, 400, size=len(alert_df))
```

```
        alert_df["Action"] = np.where(alert_df["CurrentStock"] < alert_df["ReorderPoint"], "Reorder ⚠️", "OK ✅")
```

```
    st.dataframe(alert_df[["Product", "CurrentStock", "ReorderPoint", "Action"]], use_container_width=True)
```

```
    st.subheader("Stock Level vs. Reorder Point")
```

```
    chart_df = alert_df.set_index("Product")[["CurrentStock", "ReorderPoint"]].sort_values(by="CurrentStock")
```

```
    st.bar_chart(chart_df, color=[PALETTE['primary_accent'], PALETTE['heading_text']])
```

```
else:
```

```
    st.warning("No inventory data to display.")
```

```
#Tab 4: Download Reports
```

```
with tab4:
```

```
    st.header("Download Inventory Report")
```

```
    report_csv = pd.DataFrame(inventory_plan).to_csv(index=False).encode('utf-8')
```

```
    st.download_button(
```

```
        label="⬇️ Download Daily Reorder Report",
```

```
        data=report_csv,
```

```
        file_name="daily_reorder_report.csv",
```

```

        mime="text/csv",
    )

st.info("This report contains the calculated EOQ, safety stock, and reorder points for all products.")

#Sidebar: Upload New Data
st.sidebar.header("Refresh Data")
uploaded_file = st.sidebar.file_uploader("Upload New Sales Data (CSV)", type="csv")
if uploaded_file:
    os.makedirs("data", exist_ok=True)
    upload_path = os.path.join("data", "new_sales_data.csv")
    with open(upload_path, "wb") as f:
        f.write(uploaded_file.getbuffer())
    st.sidebar.success("File uploaded successfully! ✅")
    st.sidebar.info("To see updated forecasts, please re-run the external forecasting script and then refresh this dashboard.")

```

4. Dashboard & Results

- Forecast Visualization: Allows users to select products and compare Actual vs Forecasted Sales using line charts.
- Inventory Planning: Displays EOQ, Safety Stock, and Reorder Points in a styled table for all products.
- Stock Alerts: Shows current stock vs reorder points and visualizes them in a bar chart.
- Reports & Data Upload: Allows downloading daily reorder reports and uploading new sales data for refresh.

5. Observations

- Tabbed structure improves user navigation and clarity.
- Custom color palette improves professional aesthetics.
- Stock Alerts tab simplifies decision-making for inventory.
- Forecast Visualization enhances trust in the forecasting model.
- Data uploader enables real-world adaptability of the dashboard.

6. Outputs

Smart Inventory Dashboard + http://localhost:8501 Assistant Deploy

Inventory Parameters
7
50
2.00
95% Refresh Data

Drag and drop file here Limit 200MB per file + CSV Browse files

28°C Mostly cloudy

Smart Inventory Management Dashboard

Forecast Visualization Inventory Planning Stock Alerts Reports

Sales Forecast Visualization

Select Product to View Forecast P0001

Sales Forecast for Product: P0001

Sales Quantity

Forecasted Sales
Actual Sales

07:02 PM 26-10-2025

Smart Inventory Dashboard + http://localhost:8501 Assistant Deploy

Inventory Parameters
7
50
2.00
95% Refresh Data

Drag and drop file here Limit 200MB per file + CSV Browse files

28°C Mostly cloudy

Smart Inventory Management Dashboard

Forecast Visualization Inventory Planning Stock Alerts Reports

Inventory Optimization Plan

	Product ID	Avg Daily Demand	Std Dev of Demand	Economic Order Quantity (EOQ)	Safety Stock	Reorder Point
0	P0001	675.70	4.16	12206	18	4748
1	P0002	645.53	1.23	11931	5	4524
2	P0003	663.12	0.91	12092	4	4646
3	P0004	696.04	46.37	12389	202	5075
4	P0005	690.82	1.39	12342	6	4842
5	P0006	676.49	58.08	12213	254	4989
6	P0007	683.49	1.77	12276	8	4792
7	P0008	707.58	5.62	12491	25	4978
8	P0009	668.98	0.62	12145	3	4686
9	P0010	672.86	3.17	12181	14	4724

07:02 PM 26-10-2025

Smart Inventory Dashboard

http://localhost:8501

Inventory Parameters

- 7
- 50
- 2.00

95%

Refresh Data

Drag and drop file here
Limit 200MB per file - CSV

Browse files

Smart Inventory Management Dashboard

Forecast Visualization | Inventory Planning | Stock Alerts | Reports

Current Stock Status and Reorder Alerts

Product	CurrentStock	ReorderPoint	Action
P0001	227	4748	Reorder ⚠
P0002	238	4524	Reorder ⚠
P0003	166	4646	Reorder ⚠
P0004	87	5075	Reorder ⚠
P0005	328	4842	Reorder ⚠
P0006	193	4989	Reorder ⚠
P0007	203	4792	Reorder ⚠
P0008	334	4978	Reorder ⚠
P0009	290	4686	Reorder ⚠
P0010	318	4724	Reorder ⚠

Stock Level vs. Reorder Point

28°C Mostly cloudy

07:03 PM 26-10-2025

Smart Inventory Dashboard

http://localhost:8501

Inventory Parameters

- 7
- 50
- 2.00

95%

Refresh Data

Drag and drop file here
Limit 200MB per file - CSV

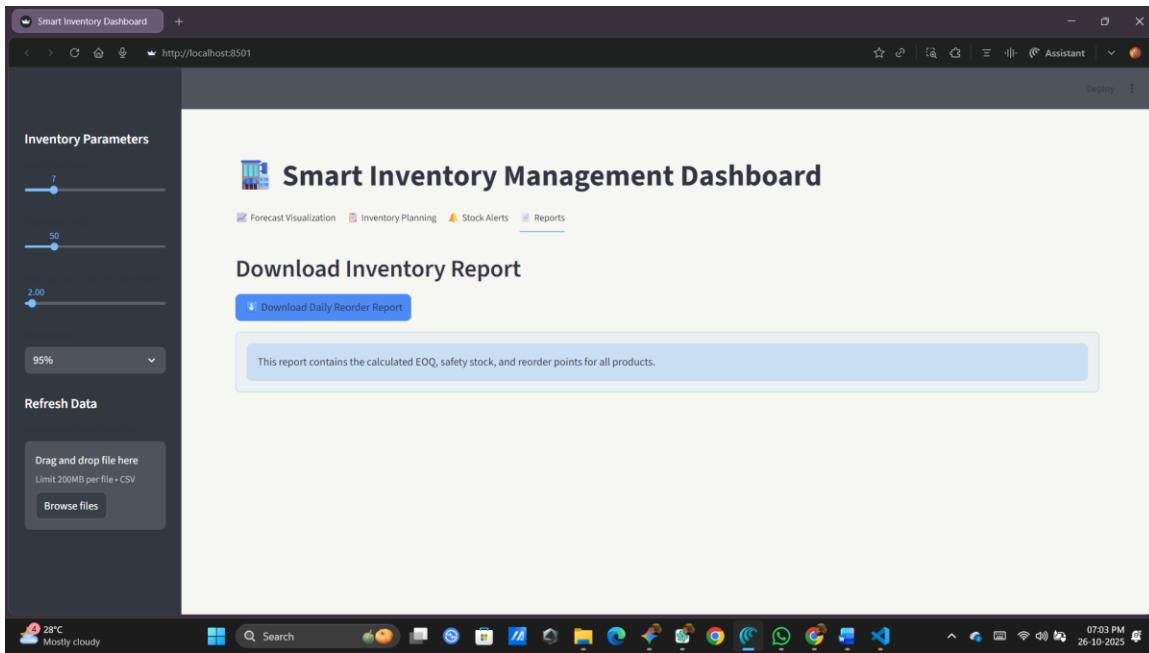
Browse files

Stock Level vs. Reorder Point

P0007	203	4792	Reorder ⚠
P0008	334	4978	Reorder ⚠
P0009	290	4686	Reorder ⚠
P0010	318	4724	Reorder ⚠

CurrentStock ReorderPoint

07:03 PM 26-10-2025



7. Conclusion & Next Steps

Milestone 4 successfully transformed the SmartStock project into a polished, user-friendly application. The redesigned interface, organized features, and new functionalities enhance usability and provide complete visibility into inventory insights. Future steps include deploying the solution on the cloud and integrating real-time database connectivity for enterprise use.