**Capstone project 3.**

**Azim Aydogdyyev**

**1. Dashboard Plan (Step 1)**

The dashboard is designed to provide key insights into the online retail sales data for different user groups within the organization. It is structured into two main sections: an Analytical Dashboard and an Operational Dashboard, each tailored to the specific needs of its intended audience.**Analytical Dashboard:**

* **Target Audience:** Business Analysts and Data Scientists.
* **Purpose:** To provide a comprehensive view of historical sales trends, product performance, customer behavior, and geographical sales distribution over customizable time periods. This helps in identifying long-term trends, making strategic decisions, and performing in-depth data analysis.
* **Key Sections/Metrics:**
* **Sales Trend:** Visualizes total sales over the selected date range to show performance fluctuations and growth patterns.
* **Top Selling Products:** Lists the products generating the highest revenue and order counts, helping to identify best-sellers.
* **Sales by Category:** Breaks down sales and order counts by product category to understand category performance.
* **Customer Segments:** Shows customer counts and total revenue grouped by country, providing insights into geographical market performance.
* **Revenue by Country:** Detailed view of revenue, order counts, and customer counts per country, allowing for country-specific analysis.
* **Features:** Supports filtering data by date range, product category, and search terms for products/stock codes.

**Operational Dashboard:**

* **Target Audience:** Operations Team and Management.
* **Purpose:** To monitor daily operational performance and track recent activity. This helps in overseeing day-to-day sales, identifying issues, and managing immediate operational needs.
* **Key Sections/Metrics:**
* **Daily Metrics:** Displays key performance indicators for a selected day, including total order count, total sales, and average order value.
* **Invoice Distribution by Country:** Shows the breakdown of invoices and total sales for the day by country.
* **Recent Invoices:** Lists recent transactions with details like invoice number, date, country, and total amount.
* **Features:** Allows selection of a specific date (defaults to the latest data available or today) and filtering by country.

**Additional Features:**The application also includes functionalities to retrieve stock alerts for low-stock products and identify repeat customers, which can support both analytical and operational decision-making.**Design Considerations:**The dashboard is designed to be responsive and user-friendly, providing clear visualizations and easy-to-use filtering options to allow users to quickly access the information they need. It leverages the database connection and efficient SQL queries (implemented via SQLAlchemy) to fetch and process data dynamically based on user selections.

**2. Database Connection & SQL Queries (Step 2)**

**Database Connection:**The application connects to the database using the following connection string, defined in config.py:

text

Apply to config.py

SQLALCHEMY\_DATABASE\_URI = 'mysql+mysqlconnector://root:5h5hjfkC@localhost:3306/online\_retail\_db'

**SQL Queries:**Here are some of the key SQL queries used in the application (found primarily in app/routes.py and app/import\_data.py), implemented using SQLAlchemy's ORM and Core features:

python

Apply to config.py

*# Example Query 1: Sales Trend (from app/routes.py)*

sales\_trend = (

    db.session.query(

        func.date(Invoice.invoice\_date).label('date'),

        func.sum(InvoiceLine.quantity \* InvoiceLine.unit\_price).label('total\_sales')

    )

    .select\_from(Invoice)

    .join(InvoiceLine, Invoice.invoice\_no == InvoiceLine.invoice\_no)

    .filter(Invoice.invoice\_date.between(start\_date\_obj, end\_date\_obj))

    .group\_by(func.date(Invoice.invoice\_date))

    .order\_by('date')

    .all()

)

python

Apply to config.py

*# Example Query 2: Top Selling Products (from app/routes.py)*

top\_products = (

    db.session.query(

        Product.description.label('product\_name'),

        func.sum(InvoiceLine.quantity \* InvoiceLine.unit\_price).label('revenue'),

        func.count(Invoice.invoice\_no.distinct()).label('order\_count')

    )

    .select\_from(Product)

    .join(InvoiceLine, Product.product\_id == InvoiceLine.product\_id)

    .join(Invoice, InvoiceLine.invoice\_no == Invoice.invoice\_no)

    .filter(Invoice.invoice\_date.between(start\_date\_obj, end\_date\_obj))

    .group\_by(Product.product\_id, Product.description)

    .order\_by(desc('revenue'))

    .limit(10)

    .all()

)

python

Apply to config.py

*# Example Query 3: Data Import Query (from app/import\_data.py)*

query = """

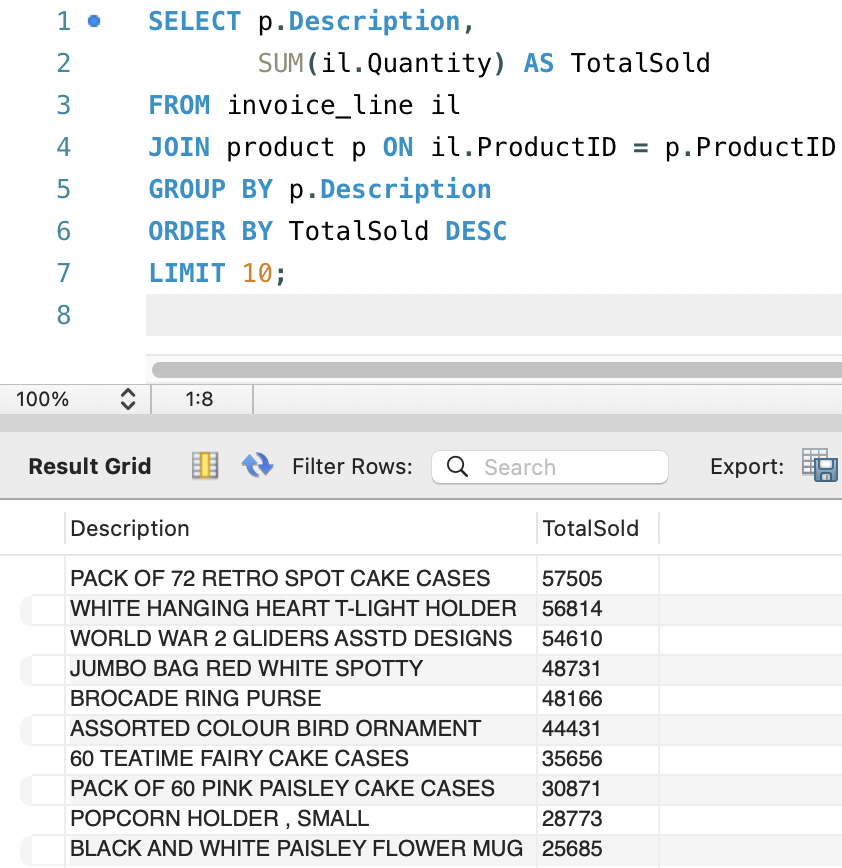
    SELECT \*

    FROM online\_retail\_source

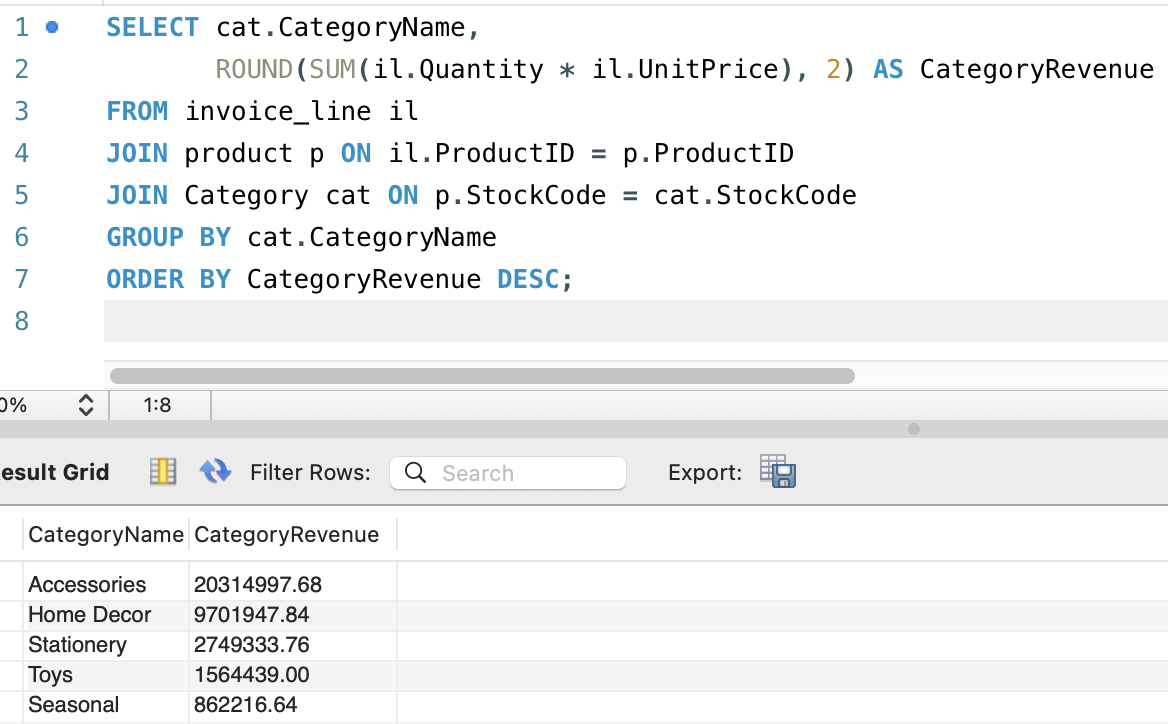
"""

source\_data\_df = pd.read\_sql(query, conn)

**Top 10 Best-Selling Products**



**Product Categories with Total Revenue**



**3. N/A**

**4. Dashboard 1 Code + Screenshots**

{% extends "base.html" %}

{% block title %}Analytical Dashboard - Online Retail{% endblock %}

{% block content %}

<div class="row mb-4">

<div class="col-12">

<h2>Analytical Dashboard</h2>

<p class="text-muted">Sales Analytics & Insights</p>

</div>

</div>

*<!-- Filters -->*

<div class="row mb-4">

<div class="col-md-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Date Range</h5>

<div class="input-group">

<input type="date" class="form-control" id="startDate" value="{{ start\_date }}" min="{{ sales\_dates|first }}" max="{{ sales\_dates|last }}">

<span class="input-group-text">to</span>

<input type="date" class="form-control" id="endDate" value="{{ end\_date }}" min="{{ sales\_dates|first }}" max="{{ sales\_dates|last }}">

</div>

</div>

</div>

</div>

<div class="col-md-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Category Filter</h5>

<select class="form-select" id="categoryFilter">

<option value="">All Categories</option>

{% for category in categories %}

<option value="{{ category }}" {% if category == selected\_category %}selected{% endif %}>{{ category }}</option>

{% endfor %}

</select>

</div>

</div>

</div>

<div class="col-md-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Country Filter</h5>

<select class="form-select" id="countryFilter">

<option value="">All Countries</option>

{% for country in countries %}

<option value="{{ country }}" {% if country == selected\_country %}selected{% endif %}>{{ country }}</option>

{% endfor %}

</select>

</div>

</div>

</div>

</div>

<div class="row mb-4">

<div class="col-md-12">

<div class="card">

<div class="card-body">

<h5 class="card-title">Search</h5>

<div class="input-group">

<input type="text" class="form-control" id="searchInput" placeholder="Search by product description or stock code..." value="{{ search }}">

<button class="btn btn-primary" type="button" id="searchButton">Search</button>

</div>

</div>

</div>

</div>

</div>

*<!-- Sales Trend -->*

<div class="row mb-4">

<div class="col-md-12">

<div class="card">

<div class="card-body">

<h5 class="card-title">Sales Trend</h5>

<canvas id="salesTrendChart"></canvas>

</div>

</div>

</div>

</div>

*<!-- Top Products and Category Distribution -->*

<div class="row mb-4">

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Top Products</h5>

<canvas id="topProductsChart"></canvas>

</div>

</div>

</div>

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Category Distribution</h5>

<canvas id="categoryDistributionChart"></canvas>

</div>

</div>

</div>

</div>

*<!-- Revenue by Country and Customer Segments -->*

<div class="row mb-4">

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Customer Distribution by Country</h5>

<canvas id="customerSegmentsChart"></canvas>

</div>

</div>

</div>

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Revenue by Country</h5>

<canvas id="revenueByCountryChart"></canvas>

</div>

</div>

</div>

</div>

*<!-- Stock Alerts and Repeat Customers -->*

<div class="row mb-4">

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Stock Alerts</h5>

<div class="table-responsive">

<table class="table table-hover" id="stockAlertsTable">

<thead>

<tr>

<th>Product</th>

<th>Stock Level</th>

<th>Status</th>

</tr>

</thead>

<tbody>

*<!-- Will be populated by JavaScript -->*

</tbody>

</table>

</div>

</div>

</div>

</div>

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Repeat Customers</h5>

<div class="table-responsive">

<table class="table table-hover" id="repeatCustomersTable">

<thead>

<tr>

<th>Customer</th>

<th>Orders</th>

<th>Total Spent</th>

</tr>

</thead>

<tbody>

*<!-- Will be populated by JavaScript -->*

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

{% endblock %}

{% block scripts %}

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script src="{{ url\_for('static', filename='js/dashboard1.js') }}"></script>

{% autoescape false %}

<script type="text/javascript">

window.dashboard1Initial = {

sales\_trend: {{ sales\_trend|tojson }},

top\_products: {{ top\_products|tojson }},

category\_distribution: {{ category\_distribution|tojson }},

customer\_segments: {{ customer\_segments|tojson }},

revenue\_by\_country: {{ revenue\_by\_country|tojson }},

start\_date: {{ start\_date|tojson }},

end\_date: {{ end\_date|tojson }},

categories: {{ categories|tojson }},

countries: {{ countries|tojson }},

selected\_category: {{ selected\_category|tojson }},

selected\_country: {{ selected\_country|tojson }},

search: {{ search|tojson }},

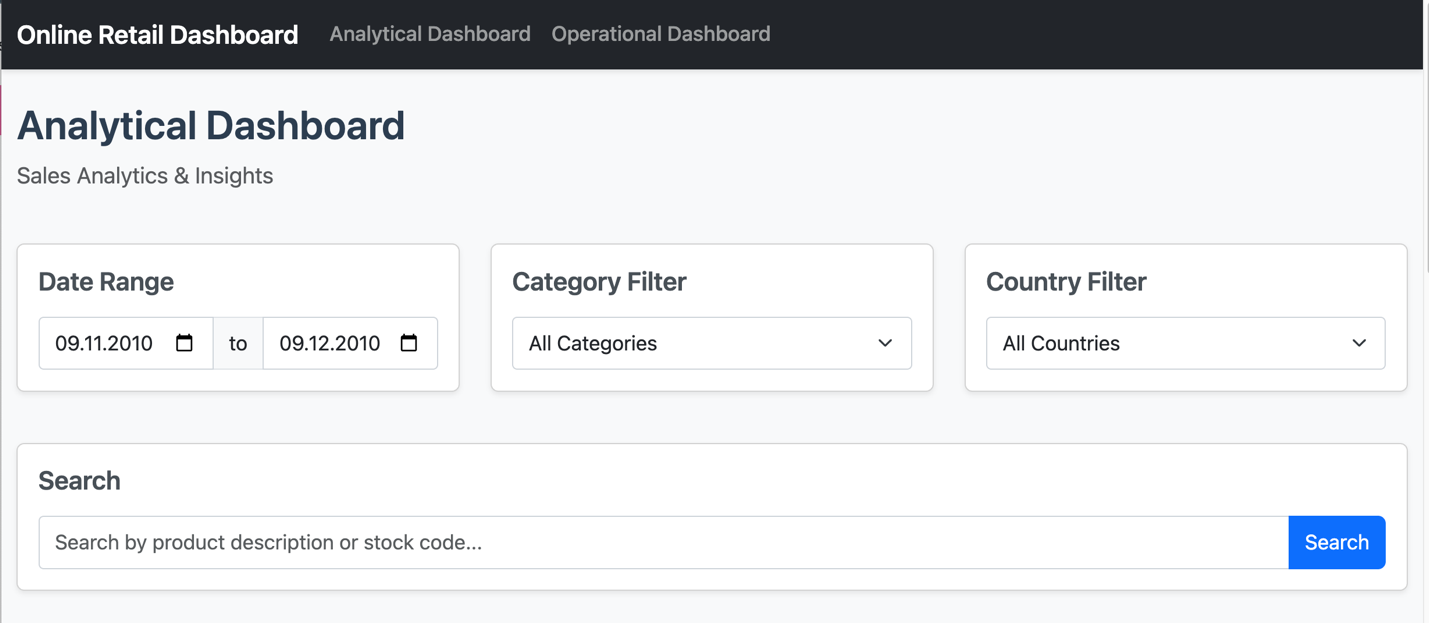
sales\_dates: {{ sales\_dates|tojson }}

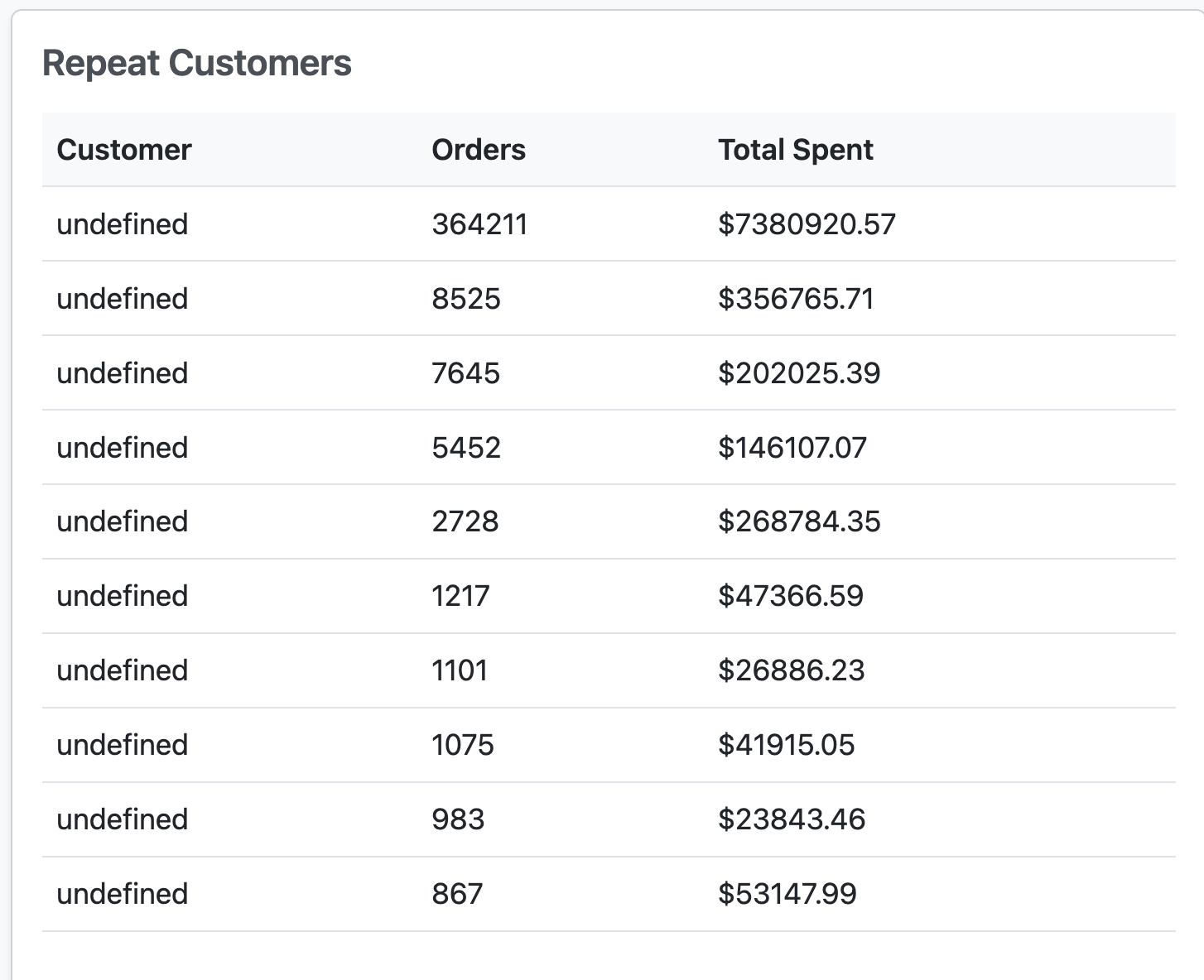
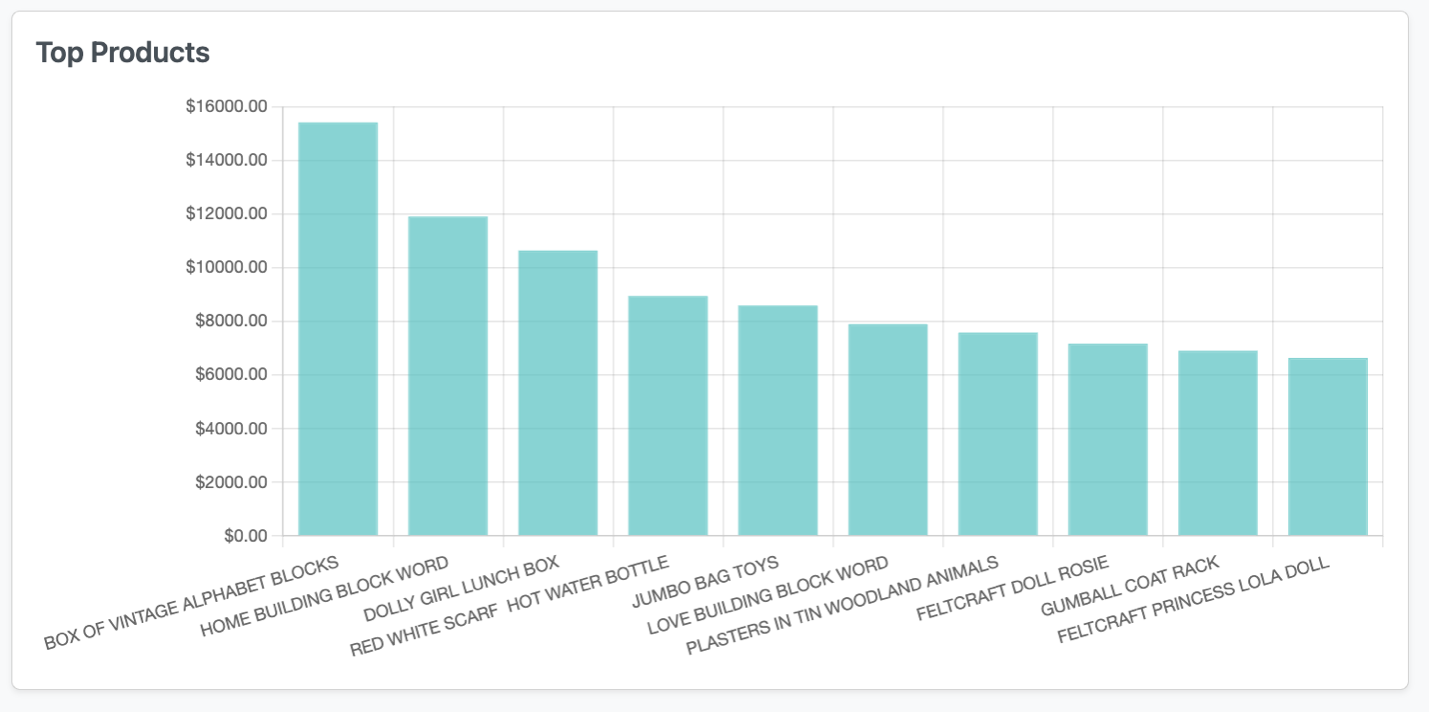
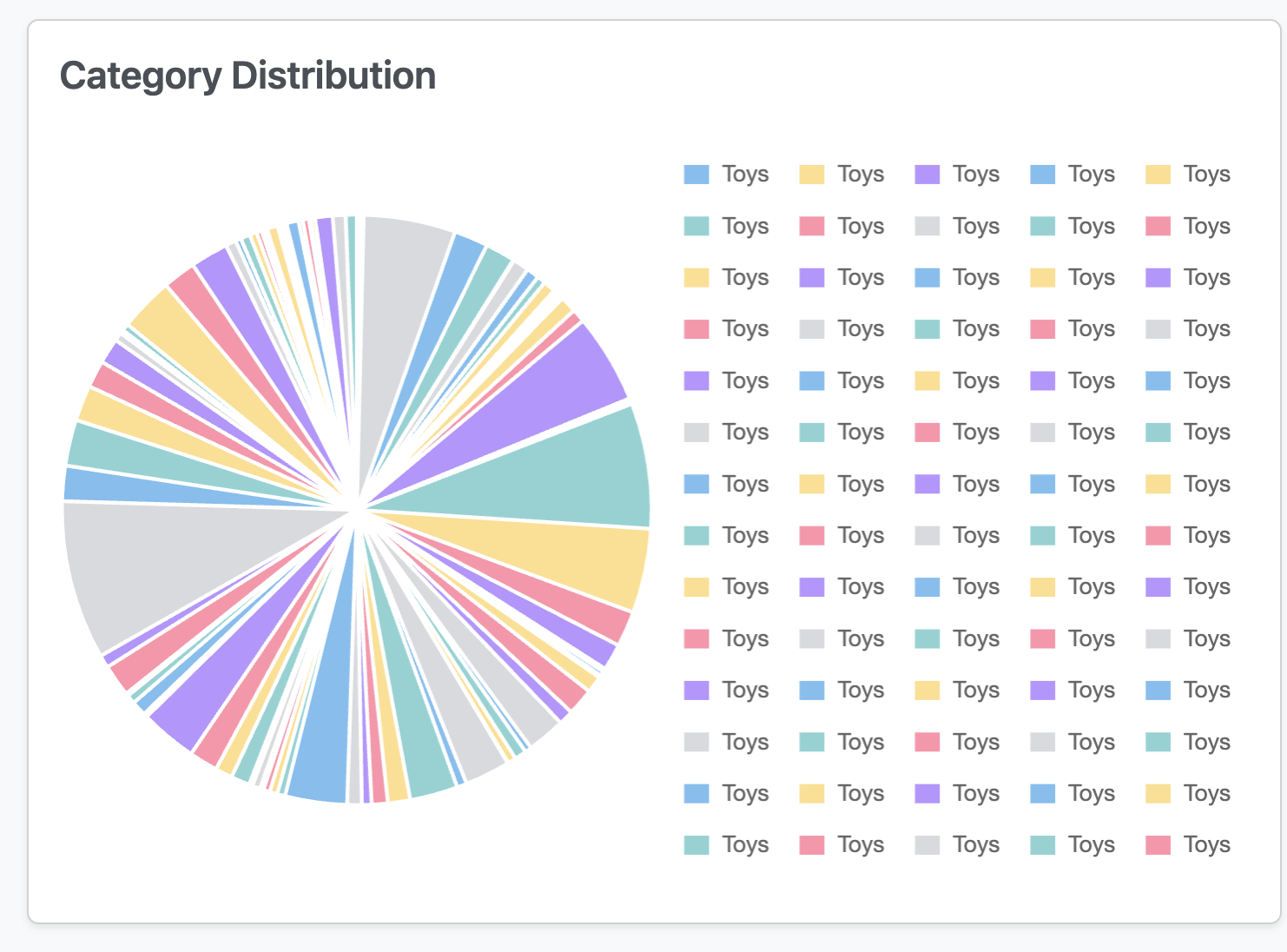
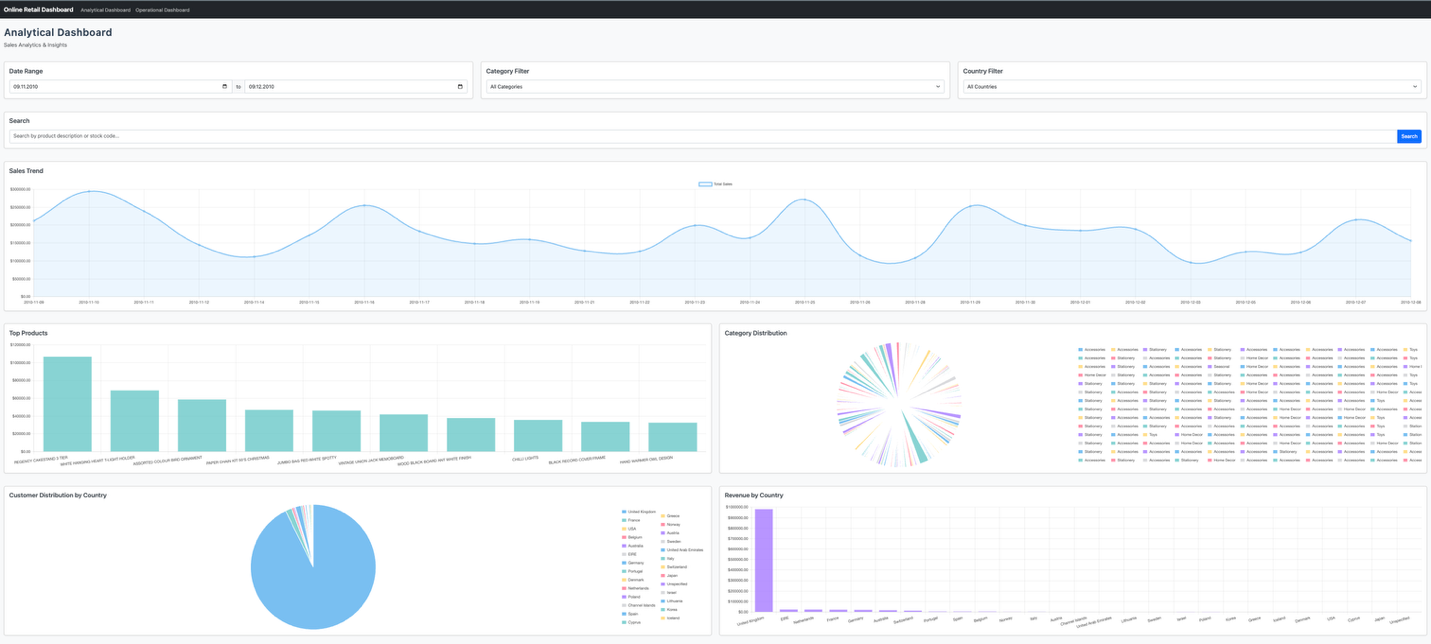
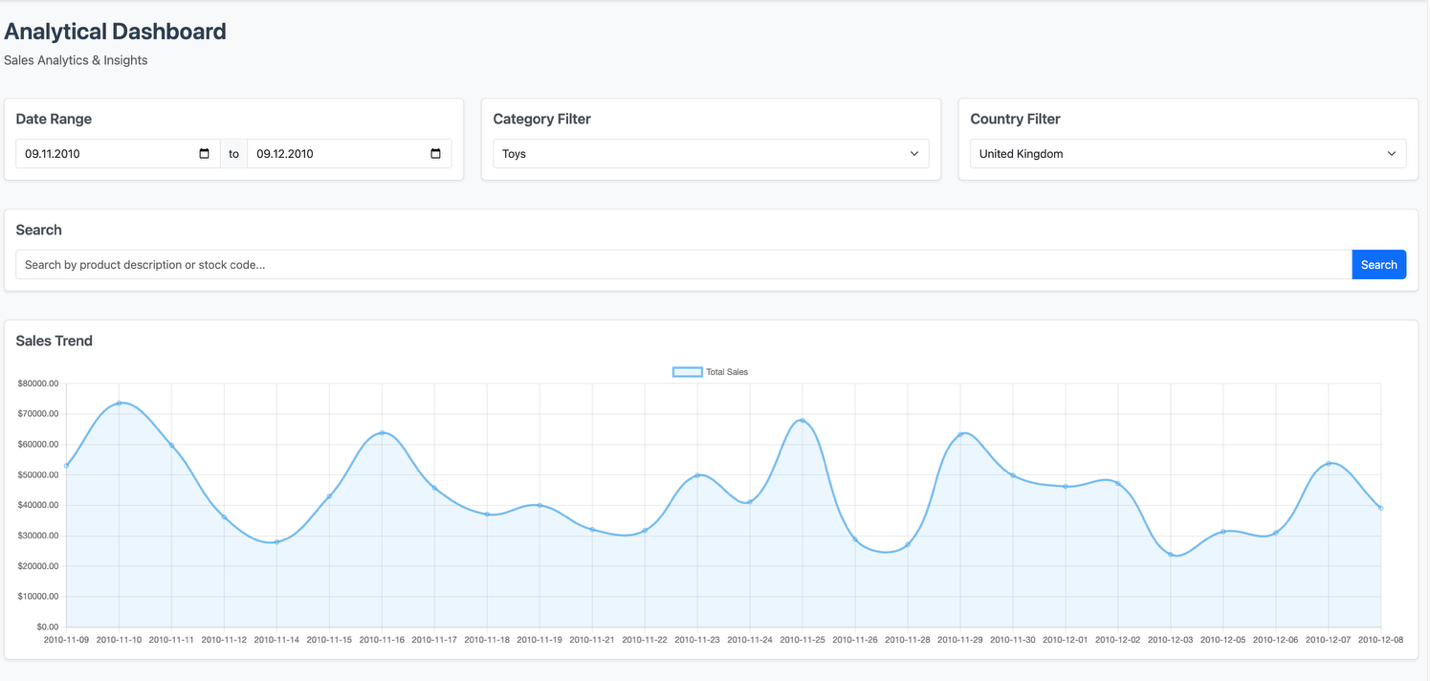
};

</script>

{% endautoescape %}

{% endblock %}

****

****

**5. Dashboard 2 Code + Screenshots**

{% extends "base.html" %}

{% block title %}Operational Dashboard - Online Retail{% endblock %}

{% block content %}

<div class="row mb-4">

<div class="col-12">

<h2>Operational Dashboard</h2>

<p class="text-muted">Daily Operations Overview</p>

</div>

</div>

*<!-- Filters -->*

<div class="row mb-4">

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Date Filter</h5>

<div class="form-check form-check-inline">

<input class="form-check-input" type="radio" name="dateMode" id="todayRadio" value="today" {% if date\_mode == 'today' %}checked{% endif %} {% if not today\_is\_available %}disabled{% endif %}>

<label class="form-check-label" for="todayRadio">Latest Available Date</label>

</div>

<div class="form-check form-check-inline">

<input class="form-check-input" type="radio" name="dateMode" id="customRadio" value="custom" {% if date\_mode == 'custom' %}checked{% endif %}>

<label class="form-check-label" for="customRadio">Custom Date</label>

</div>

<input type="date" class="form-control mt-2" id="customDate"

min="{{ available\_dates|first }}"

max="{{ available\_dates|last }}"

value="{{ current\_date }}"

{% if date\_mode == 'today' %}disabled{% endif %}>

<small class="form-text text-muted">Available dates: {{ available\_dates|first }} to {{ available\_dates|last }}</small>

</div>

</div>

</div>

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Country Filter</h5>

<select class="form-select" id="countryFilter">

<option value="">All Countries</option>

{% for country in countries %}

<option value="{{ country }}" {% if country == selected\_country %}selected{% endif %}>{{ country }}</option>

{% endfor %}

</select>

</div>

</div>

</div>

</div>

{% if no\_data %}

<div class="alert alert-info" role="alert">

<h4 class="alert-heading">No Data Available</h4>

<p>{{ no\_data\_message }}</p>

</div>

{% else %}

<div class="row mb-4">

<div class="col-md-4">

<div class="card bg-primary text-white">

<div class="card-body">

<h5 class="card-title">Invoices for {{ current\_date }}</h5>

<h2 class="display-4" id="orderCount">{{ daily\_metrics.order\_count }}</h2>

<p class="card-text text-white-50">Total invoices for selected date</p>

</div>

</div>

</div>

<div class="col-md-4">

<div class="card bg-success text-white">

<div class="card-body">

<h5 class="card-title">Sales for {{ current\_date }}</h5>

<h2 class="display-4" id="totalSales">${{ '%.2f'|format(daily\_metrics.total\_sales) }}</h2>

<p class="card-text text-white-50">Total revenue for selected date</p>

</div>

</div>

</div>

<div class="col-md-4">

<div class="card bg-info text-white">

<div class="card-body">

<h5 class="card-title">Average Order Value</h5>

<h2 class="display-4" id="averageOrderValue">${{ '%.2f'|format(daily\_metrics.average\_order\_value) }}</h2>

<p class="card-text text-white-50">Average revenue per invoice</p>

</div>

</div>

</div>

</div>

<div class="row mb-4">

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Sales by Country</h5>

<canvas id="statusDistributionChart"></canvas>

</div>

</div>

</div>

<div class="col-md-6">

<div class="card">

<div class="card-body">

<h5 class="card-title">Recent Invoices</h5>

<div class="table-responsive">

<table class="table table-hover">

<thead>

<tr>

<th>Invoice No</th>

<th>Date</th>

<th>Country</th>

<th>Amount</th>

</tr>

</thead>

<tbody id="recentInvoicesTable">

{% for invoice in recent\_invoices %}

<tr>

<td>{{ invoice.invoice\_no }}</td>

<td>{{ invoice.invoice\_date }}</td>

<td>{{ invoice.country }}</td>

<td>${{ '%.2f'|format(invoice.total\_amount) }}</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

<div class="row mb-4">

<div class="col-md-12">

<div class="card">

<div class="card-body">

<h5 class="card-title">Stock Alerts</h5>

<div class="table-responsive">

<table class="table table-hover">

<thead>

<tr>

<th>Product</th>

<th>Current Stock</th>

<th>Status</th>

</tr>

</thead>

<tbody id="stockAlertsTable">

{% for alert in stock\_alerts %}

<tr>

<td>{{ alert.product\_name }}</td>

<td>{{ alert.quantity\_in\_stock }}</td>

<td>

{% if alert.quantity\_in\_stock <= 10 %}

<span class="badge bg-danger">Low Stock</span>

{% elif alert.quantity\_in\_stock <= 20 %}

<span class="badge bg-warning">Medium Stock</span>

{% else %}

<span class="badge bg-success">In Stock</span>

{% endif %}

</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

{% endif %}

{% endblock %}

{% block scripts %}

<script src="https://cdn.jsdelivr.net/npm/flatpickr"></script>

<script src="{{ url\_for('static', filename='js/dashboard2.js') }}"></script>

<script type="text/javascript">

*// Initialize dashboard data*

window.dashboard2Initial = {

daily\_metrics: {{ daily\_metrics|tojson }},

status\_distribution: {{ status\_distribution|tojson }},

recent\_invoices: {{ recent\_invoices|tojson }},

stock\_alerts: {{ stock\_alerts|tojson }},

current\_date: {{ current\_date|tojson }},

countries: {{ countries|tojson }},

selected\_country: {{ selected\_country|tojson }},

date\_mode: {{ date\_mode|tojson }},

available\_dates: {{ available\_dates|tojson }}

};

*// Initialize date picker when DOM is loaded*

document.addEventListener('DOMContentLoaded', function() {

flatpickr("#customDate", {

dateFormat: "Y-m-d",

minDate: window.dashboard2Initial.available\_dates[0],

maxDate: window.dashboard2Initial.available\_dates[window.dashboard2Initial.available\_dates.length - 1],

defaultDate: window.dashboard2Initial.current\_date,

**onChange**: function(*selectedDates*, *dateStr*) {

document.getElementById('customRadio').checked = true;

const filters = getFilters();

fetchDashboardData(filters).then(updateDashboard);

}

});

});

</script>

{% endblock %}

