**1.Source code**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" />

<title>Retail Customer Purchasing Behavior Dashboard</title>

<style>

/\* Reset and base \*/

\* {

box-sizing: border-box;

}

body {

margin: 0;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background: linear-gradient(135deg, #667eea, #764ba2);

color: #fff;

min-height: 100vh;

display: flex;

justify-content: center;

align-items: flex-start;

padding: 1rem;

}

#app {

background: #1e1e2f;

border-radius: 12px;

width: 100%;

max-width: 350px;

max-height: 600px;

overflow: hidden;

display: flex;

flex-direction: column;

box-shadow: 0 0 20px rgba(0,0,0,0.6);

}

header {

padding: 1rem 1.5rem;

background: #3a3a55;

text-align: center;

font-weight: 700;

font-size: 1.2rem;

letter-spacing: 1px;

border-top-left-radius: 12px;

border-top-right-radius: 12px;

}

main {

flex: 1;

overflow-y: auto;

padding: 1rem 1.5rem 1.5rem;

}

.summary-cards {

display: flex;

justify-content: space-between;

gap: 10px;

margin-bottom: 1rem;

}

.card {

background: #2a2a45;

flex: 1;

border-radius: 8px;

padding: 1rem;

text-align: center;

box-shadow: inset 0 0 10px #0008;

}

.card .number {

font-size: 1.8rem;

font-weight: 700;

margin-bottom: 0.3rem;

color: #90cdf4;

}

.card .label {

font-size: 0.85rem;

color: #aaa;

letter-spacing: 0.5px;

}

.filters {

margin-bottom: 1rem;

}

label {

font-size: 0.85rem;

color: #ccc;

display: block;

margin-bottom: 0.25rem;

}

select, input[type="date"] {

width: 100%;

padding: 6px 10px;

border-radius: 6px;

border: none;

background: #2a2a45;

color: #eee;

font-size: 0.9rem;

appearance: none;

outline-offset: 2px;

outline-color: #90cdf4;

margin-bottom: 0.75rem;

}

canvas {

background: #2a2a45;

border-radius: 10px;

padding: 10px;

}

footer {

font-size: 0.7rem;

color: #666;

text-align: center;

padding: 0.5rem 0;

user-select: none;

}

/\* Scrollbar styling \*/

main::-webkit-scrollbar {

width: 6px;

}

main::-webkit-scrollbar-track {

background: #1e1e2f;

}

main::-webkit-scrollbar-thumb {

background: #764ba2;

border-radius: 3px;

}

/\* Responsive font scaling \*/

@media (max-width: 400px) {

body {

padding: 0.5rem;

}

header {

font-size: 1.1rem;

padding: 0.75rem 1rem;

}

.card .number {

font-size: 1.5rem;

}

}

</style>

</head>

<body>

<div id="app" role="main" aria-label="Retail Customer Purchasing Behavior Dashboard">

<header>

Retail Customer Purchasing Behavior

</header>

<main>

<section class="summary-cards" aria-label="Summary statistics">

<div class="card" tabindex="0" aria-live="polite" aria-atomic="true">

<div class="number" id="totalSales">$0</div>

<div class="label">Total Sales</div>

</div>

<div class="card" tabindex="0" aria-live="polite" aria-atomic="true">

<div class="number" id="totalOrders">0</div>

<div class="label">Total Orders</div>

</div>

<div class="card" tabindex="0" aria-live="polite" aria-atomic="true">

<div class="number" id="uniqueCustomers">0</div>

<div class="label">Unique Customers</div>

</div>

</section>

<section class="filters" aria-label="Filter customer purchase data">

<label for="filterCategory">Product Category</label>

<select id="filterCategory" aria-controls="chartCategory" title="Filter by product category">

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

</select>

<label for="filterStartDate">Start Date</label>

<input type="date" id="filterStartDate" aria-controls="chartCategory" />

<label for="filterEndDate">End Date</label>

<input type="date" id="filterEndDate" aria-controls="chartCategory" />

</section>

<section aria-label="Purchases by category chart" style="margin-bottom:20px;">

<canvas id="chartCategory" width="320" height="220" role="img" aria-label="Bar chart showing purchases by category"></canvas>

</section>

<section aria-label="Top products chart">

<canvas id="chartTopProducts" width="320" height="220" role="img" aria-label="Pie chart showing top products purchased"></canvas>

</section>

</main>

<footer>

&copy; 2024 Retail Insights Dashboard

</footer>

</div>

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

<script>

(() => {

// Sample purchase data of customers (customerId, date, product, category, amount)

const purchaseData = [

{id: 1, customerId: 101, date: '2024-05-25', product: 'T-Shirt', category: 'Apparel', amount: 19.99},

{id: 2, customerId: 102, date: '2024-05-25', product: 'Jeans', category: 'Apparel', amount: 49.99},

{id: 3, customerId: 103, date: '2024-05-24', product: 'Sneakers', category: 'Footwear', amount: 74.99},

{id: 4, customerId: 104, date: '2024-05-23', product: 'Coffee Maker', category: 'Electronics', amount: 89.99},

{id: 5, customerId: 101, date: '2024-05-22', product: 'T-Shirt', category: 'Apparel', amount: 19.99},

{id: 6, customerId: 105, date: '2024-05-21', product: 'Smartphone', category: 'Electronics', amount: 299.99},

{id: 7, customerId: 106, date: '2024-05-21', product: 'Headphones', category: 'Electronics', amount: 59.99},

{id: 8, customerId: 107, date: '2024-05-21', product: 'Sneakers', category: 'Footwear', amount: 79.99},

{id: 9, customerId: 103, date: '2024-05-20', product: 'Jeans', category: 'Apparel', amount: 49.99},

{id: 10, customerId: 108, date: '2024-05-19', product: 'Backpack', category: 'Accessories', amount: 39.99},

{id: 11, customerId: 109, date: '2024-05-18', product: 'Sunglasses', category: 'Accessories', amount: 24.99},

{id: 12, customerId: 110, date: '2024-05-18', product: 'T-Shirt', category: 'Apparel', amount: 19.99},

{id: 13, customerId: 101, date: '2024-05-17', product: 'Sneakers', category: 'Footwear', amount: 79.99},

{id: 14, customerId: 111, date: '2024-05-16', product: 'Smartwatch', category: 'Electronics', amount: 199.99},

{id: 15, customerId: 112, date: '2024-05-15', product: 'Backpack', category: 'Accessories', amount: 39.99},

{id: 16, customerId: 113, date: '2024-05-14', product: 'Hat', category: 'Accessories', amount: 15.99},

{id: 17, customerId: 114, date: '2024-05-13', product: 'T-Shirt', category: 'Apparel', amount: 19.99},

{id: 18, customerId: 115, date: '2024-05-12', product: 'Jeans', category: 'Apparel', amount: 49.99},

{id: 19, customerId: 116, date: '2024-05-11', product: 'Coffee Maker', category: 'Electronics', amount: 89.99},

{id: 20, customerId: 117, date: '2024-05-10', product: 'Sneakers', category: 'Footwear', amount: 74.99},

];

const totalSalesEl = document.getElementById('totalSales');

const totalOrdersEl = document.getElementById('totalOrders');

const uniqueCustomersEl = document.getElementById('uniqueCustomers');

const filterCategoryEl = document.getElementById('filterCategory');

const filterStartDateEl = document.getElementById('filterStartDate');

const filterEndDateEl = document.getElementById('filterEndDate');

// Populate category filter dropdown

function populateCategoryFilter() {

let categories = [...new Set(purchaseData.map(p => p.category))].sort();

categories.forEach(cat => {

let option = document.createElement('option');

option.value = cat;

option.textContent = cat;

filterCategoryEl.appendChild(option);

});

}

// Helpers for date comparison

function parseDate(d) {

return new Date(d + 'T00:00:00');

}

// Filter data by user selections

function filterData() {

let selectedCategory = filterCategoryEl.value;

let startDate = filterStartDateEl.value ? parseDate(filterStartDateEl.value) : null;

let endDate = filterEndDateEl.value ? parseDate(filterEndDateEl.value) : null;

return purchaseData.filter(p => {

let purchaseDate = parseDate(p.date);

let categoryMatch = (selectedCategory === 'all' || p.category === selectedCategory);

let startDateMatch = startDate ? (purchaseDate >= startDate) : true;

let endDateMatch = endDate ? (purchaseDate <= endDate) : true;

return categoryMatch && startDateMatch && endDateMatch;

});

}

// Summary Stats Calculation

function updateSummary(filteredData) {

const totalSales = filteredData.reduce((acc, cur) => acc + cur.amount, 0);

const totalOrders = filteredData.length;

const uniqueCustomers = new Set(filteredData.map(p => p.customerId)).size;

totalSalesEl.textContent = `$${totalSales.toFixed(2)}`;

totalOrdersEl.textContent = totalOrders.toString();

uniqueCustomersEl.textContent = uniqueCustomers.toString();

}

// Chart Instances

let chartCategory = null;

let chartTopProducts = null;

// Build data for charts

function buildCharts(filteredData) {

// Aggregate purchase amount by category

let categoryMap = {};

filteredData.forEach(p => {

categoryMap[p.category] = (categoryMap[p.category] || 0) + p.amount;

});

let categories = Object.keys(categoryMap);

let categoryValues = categories.map(c => categoryMap[c]);

// Aggregate purchase amount by product (top 5)

let productMap = {};

filteredData.forEach(p => {

productMap[p.product] = (productMap[p.product] || 0) + p.amount;

});

let topProductsArr = Object.entries(productMap)

.sort((a, b) => b[1] - a[1])

.slice(0, 5);

let topProductLabels = topProductsArr.map(p => p[0]);

let topProductValues = topProductsArr.map(p => p[1]);

// Destroy old charts if any

if(chartCategory) chartCategory.destroy();

if(chartTopProducts) chartTopProducts.destroy();

// Category Bar Chart

const ctxCategory = document.getElementById('chartCategory').getContext('2d');

chartCategory = new Chart(ctxCategory, {

type: 'bar',

data: {

labels: categories,

datasets: [{

label: 'Sales Amount ($)',

data: categoryValues,

backgroundColor: categories.map(() => 'rgba(144, 205, 244, 0.8)'),

borderColor: categories.map(() => 'rgba(64, 117, 199, 1)'),

borderWidth: 1,

borderRadius: 6,

maxBarThickness: 50,

hoverBackgroundColor: 'rgba(255,255,255,0.8)'

}]

},

options: {

responsive: false,

maintainAspectRatio: false,

scales: {

y: {

beginAtZero: true,

ticks: { color: '#ccc' },

grid: { color: '#333' }

},

x: {

ticks: { color: '#ccc' },

grid: { color: '#333' }

}

},

plugins: {

legend: {

labels: { color: '#ccc' }

},

tooltip: {

backgroundColor: '#444',

titleColor: '#ddd',

bodyColor: '#eee',

cornerRadius: 6

}

}

}

});

// Top Products Pie Chart

const ctxTopProducts = document.getElementById('chartTopProducts').getContext('2d');

const colors = [

'rgba(153, 102, 255, 0.8)',

'rgba(255, 159, 64, 0.8)',

'rgba(255, 99, 132, 0.8)',

'rgba(54, 162, 235, 0.8)',

'rgba(75, 192, 192, 0.8)'

];

chartTopProducts = new Chart(ctxTopProducts, {

type: 'pie',

data: {

labels: topProductLabels,

datasets: [{

data: topProductValues,

backgroundColor: colors,

borderColor: '#1e1e2f',

borderWidth: 2

}]

},

options: {

responsive: false,

maintainAspectRatio: false,

plugins: {

legend: {

position: 'bottom',

labels: { color: '#ccc', font: { size: 13 } }

},

tooltip: {

backgroundColor: '#444',

titleColor: '#ddd',

bodyColor: '#eee',

cornerRadius: 6,

callbacks: {

label: function(ctx) {

let label = ctx.label || '';

let val = ctx.parsed || 0;

return `${label}: $${val.toFixed(2)}`;

}

}

}

}

}

});

}

// Update dashboard based on filters

function updateDashboard() {

const filteredData = filterData();

updateSummary(filteredData);

buildCharts(filteredData);

}

// Initialize UI

function init() {

populateCategoryFilter();

// Set default date filter range: last 15 days

let today = new Date();

let priorDate = new Date();

priorDate.setDate(today.getDate() - 15);

filterStartDateEl.value = priorDate.toISOString().substring(0, 10);

filterEndDateEl.value = today.toISOString().substring(0, 10);

// Add filter event listeners

filterCategoryEl.addEventListener('change', updateDashboard);

filterStartDateEl.addEventListener('change', updateDashboard);

filterEndDateEl.addEventListener('change', updateDashboard);

updateDashboard();

}

window.addEventListener('load', init);

})();

</script>

</body>

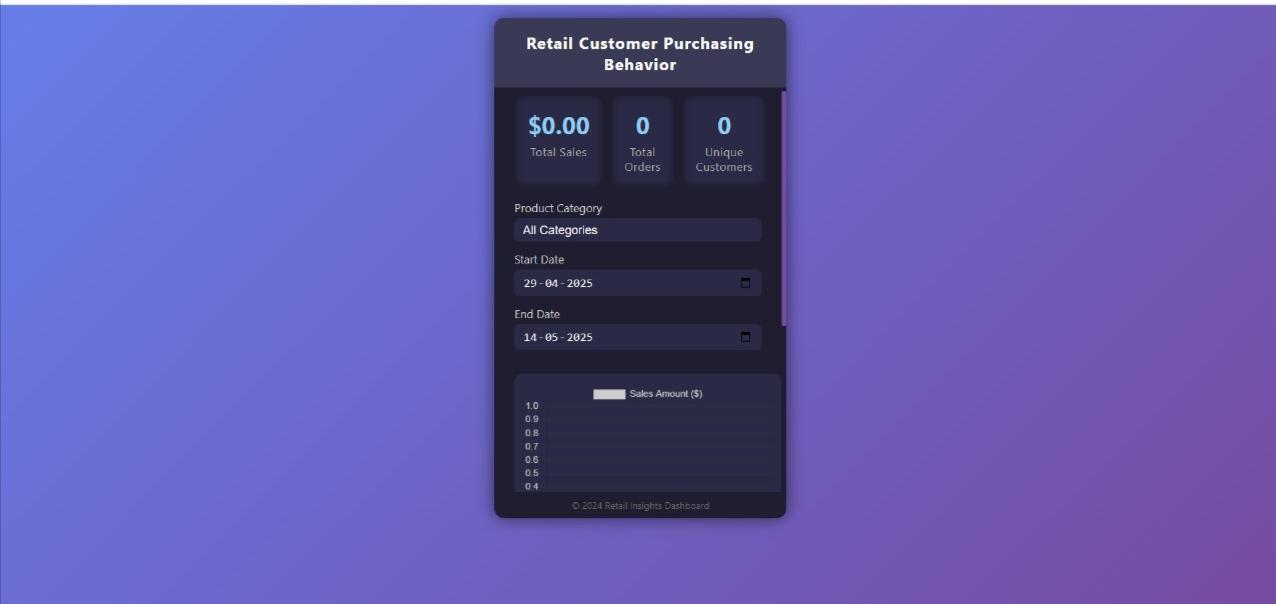
</html>

</content>

</create\_file>

**2.Visualizations/Dashboard**

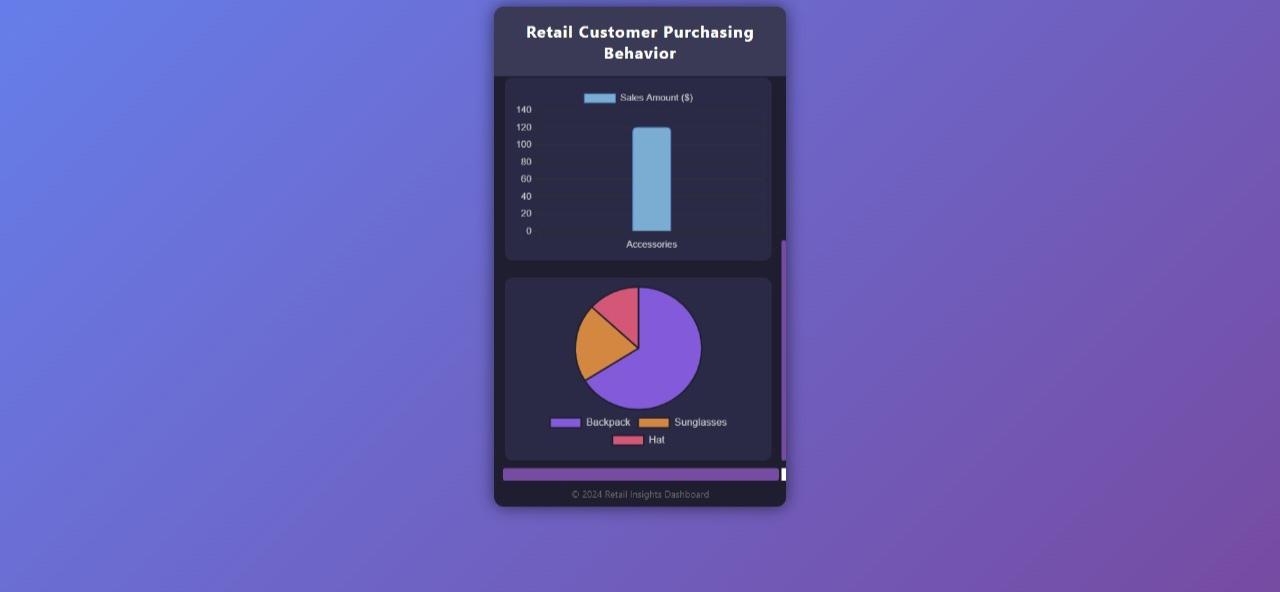
1.This section of the dashboard serves as a summary and filtering interface for analyzing retail purchasing data across a specific time period and product categories.



Allowing dynamic querying of the dataset

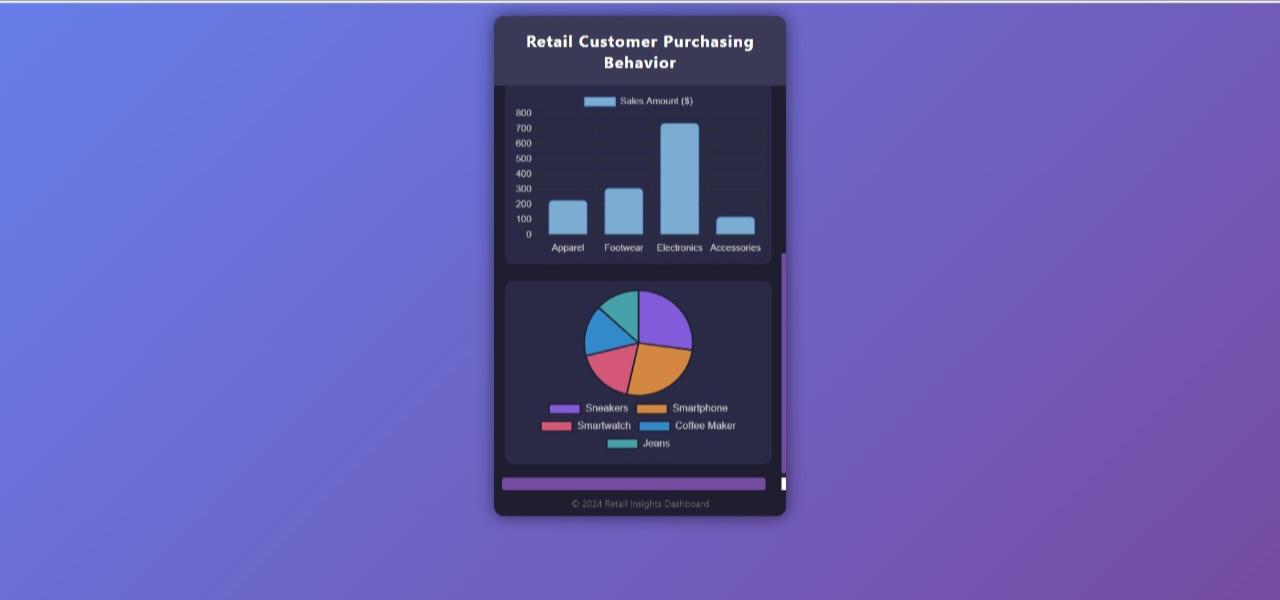
* Providing an at-a-glance summary of retail performance
* Helping identify data gaps or inactivity, as seen in this example with zero sales

2.This dashboard zooms into the Accessories category within the retail purchasing behavior analysis. It helps stakeholders understand the composition of sales within a single product category.



* Inventory planning within a category
* Product-specific promotions (e.g., boosting Hat sales)
* Identifying customer preferences at a micro level

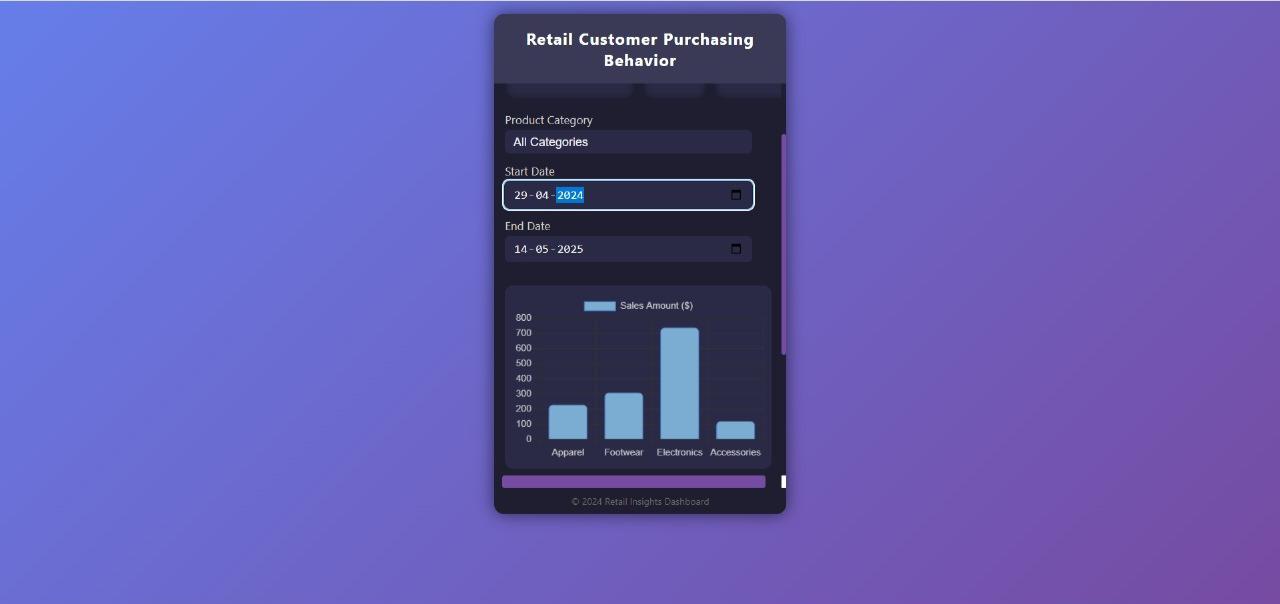
3. This extended view of the Retail Insights Dashboard provides a more granular analysis of product-level performance, building on category-level insights.



Compare broad category performance (bar chart) with individual product contributions (pie chart)

* Identify best-selling items to optimize stock levels
* Make data-backed marketing decisions by focusing on popular products

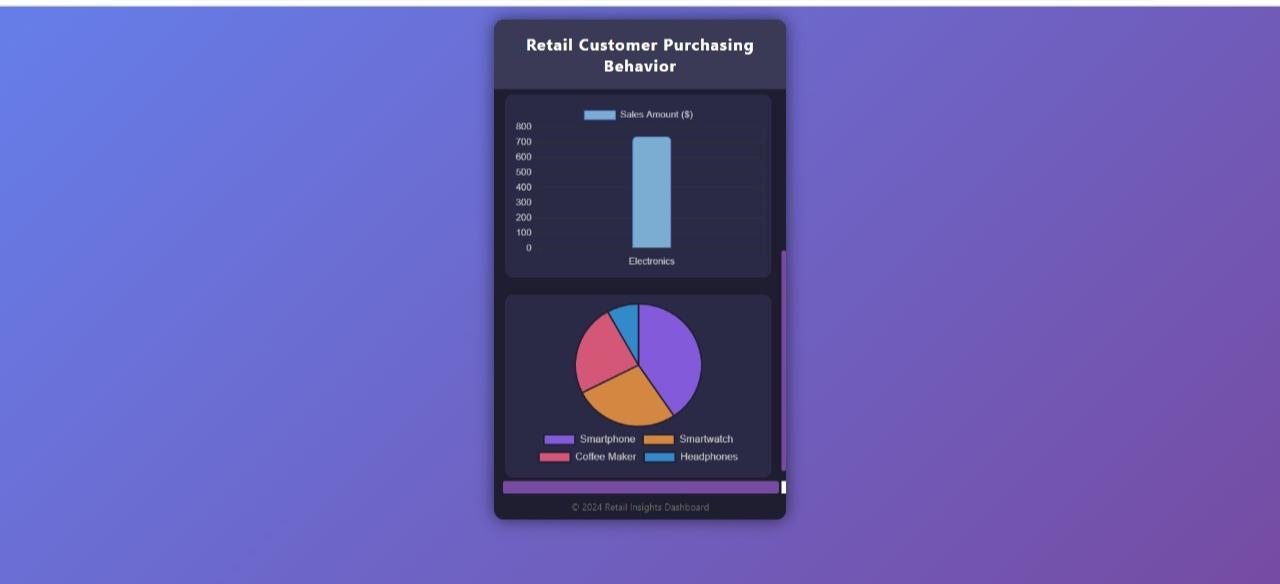
4. This dashboard visualizes customer purchasing trends across various product categories over a selected date range. It helps stakeholders understand which product categories are performing best in terms of sales.



This type of dashboard is valuable for:

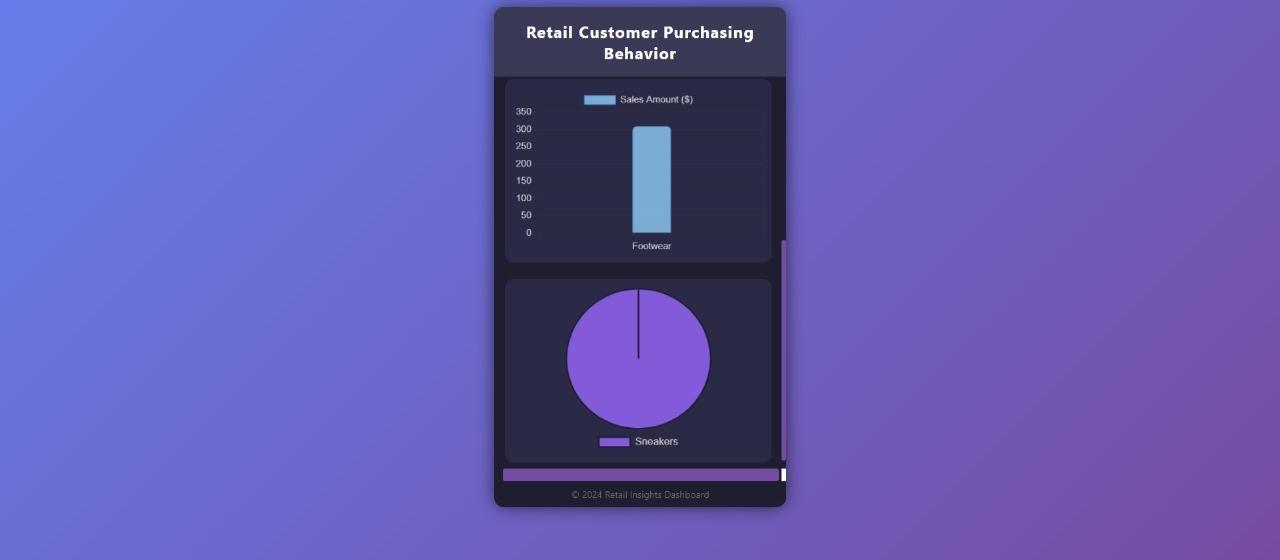
* Business analysts to track performance
* Retail managers to make inventory decisions
* Marketing teams to identify high-performing product categories

5. This dashboard panel presents a detailed view of customer purchasing behavior specific to the Electronics category, combining both sales performance and product distribution insights.



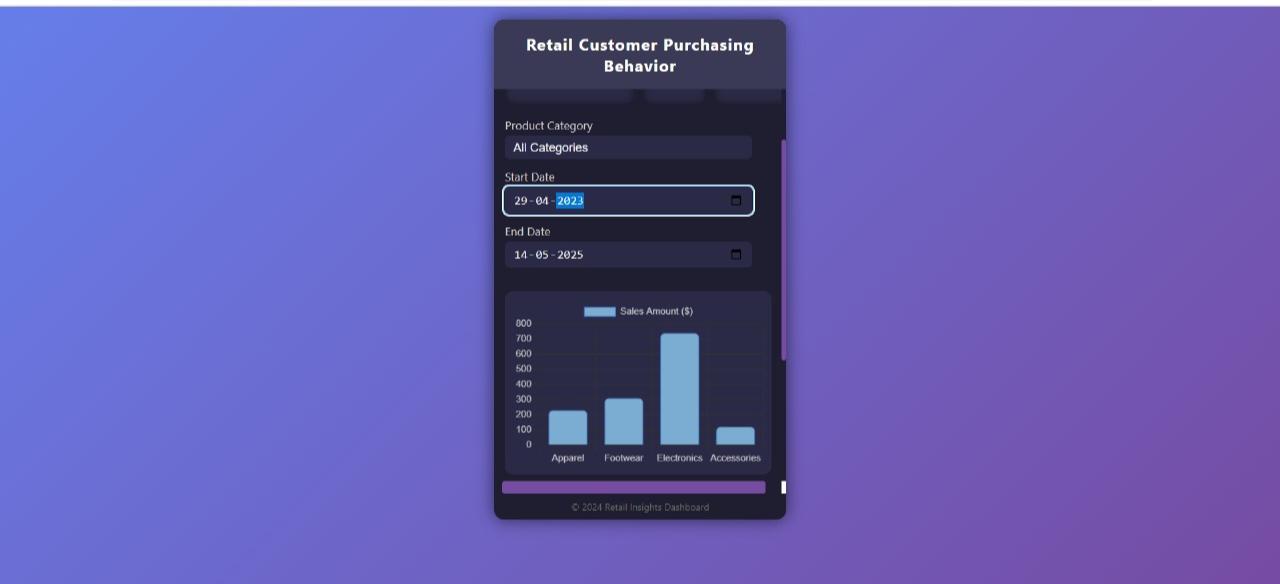
* Understand which electronics products drive revenue
* Allocate marketing efforts more effectively
* Make data-backed inventory decisions

6. This dashboard segment provides a focused view on sales and customer preferences within the Footwear category, offering insights into product-specific performance.



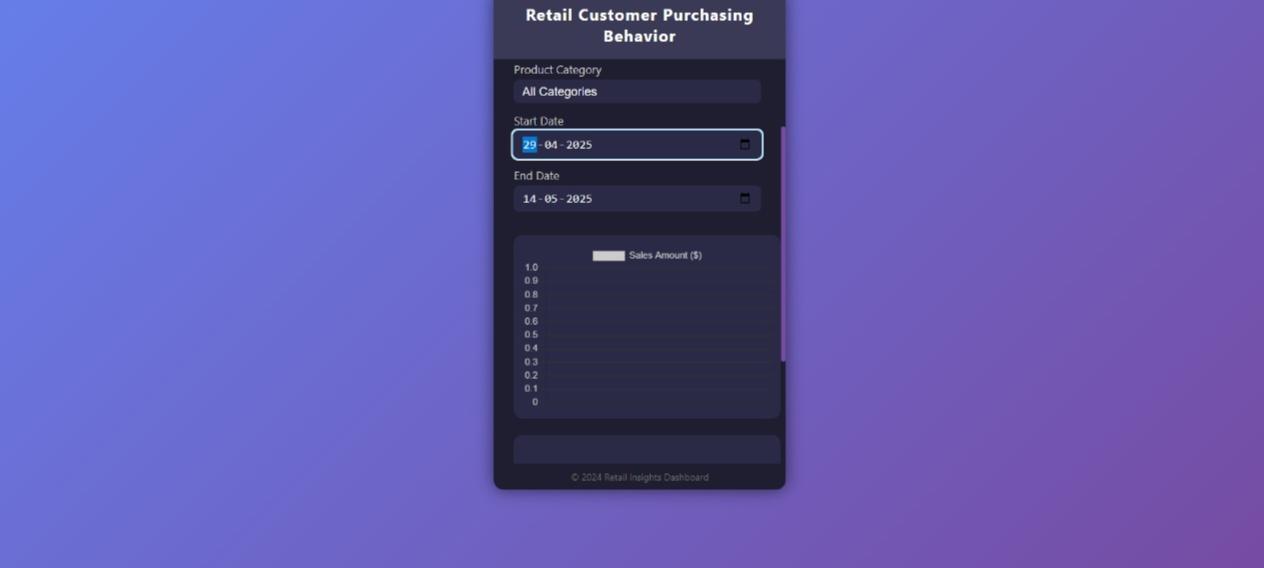
* Recognize the monopoly of sneakers in current footwear sales.
* Identify gaps in product offerings within the category.
* Plan for diversified inventory or promotional campaigns to boost revenue.

7.This dashboard provides a comprehensive overview of customer purchasing patterns across multiple product categories within a selected date range.



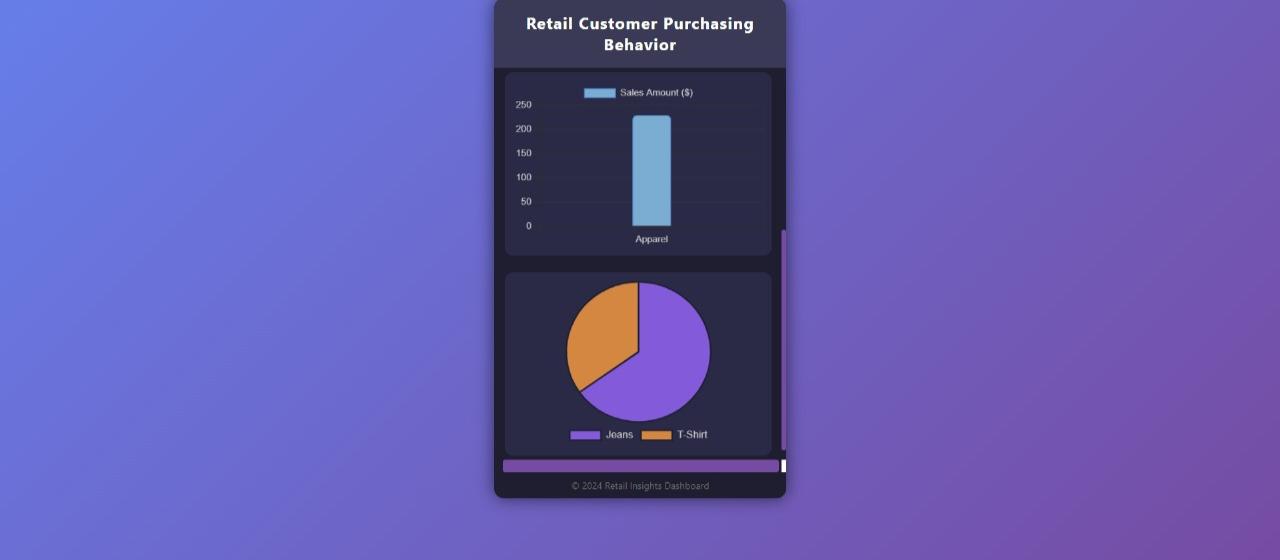
* Monitoring sales trends across multiple categories.
* Identifying high- and low-performing segments.
* Making data-driven decisions about inventory, marketing, and promotions.

8.This screen represents a scenario where no customer purchase data is available for the selected time frame.



* Demonstrates how the dashboard gracefully handles null or zero-value data states.
* Validates the user interface's capability to remain functional and informative, even when no sales activity is present.
* Reinforces the importance of date selection in data analysis workflows.

9. This screen provides a focused analysis of a specific product category, offering both a bar chart and a pie chart to break down performance.



* Highlights how the dashboard can drill down into specific categories for deeper insights.
* Useful for category managers and marketing teams aiming to optimize product strategies.
* Demonstrates the dashboard’s ability to present multi-dimensional insights effectively.