

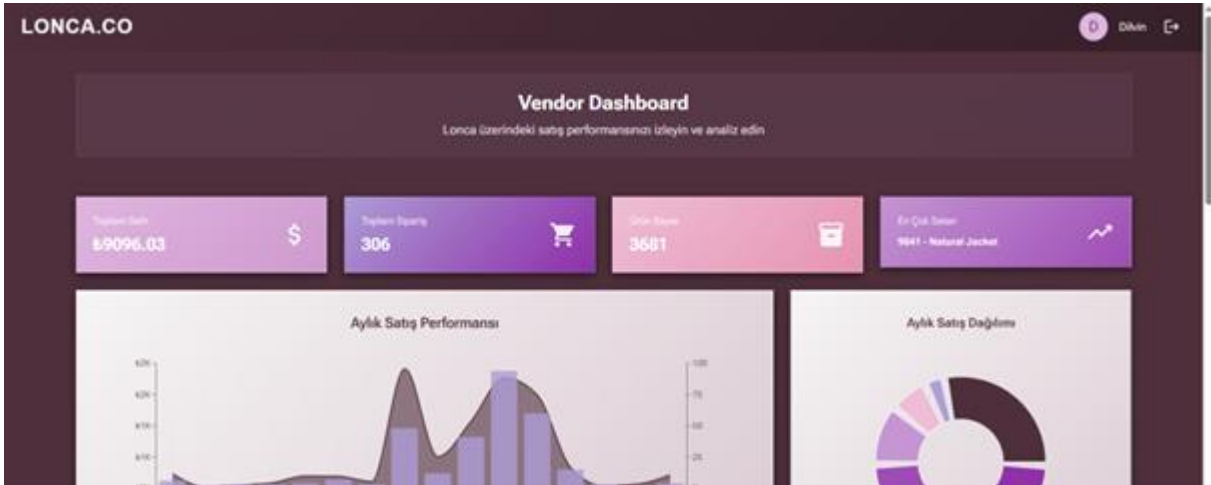
1. Project Overview

Lonca Vendor Dashboard is a comprehensive analytics dashboard application developed to monitor and analyze supplier performance on e-commerce platforms. The project offers the ability to visualize supplier sales data, track product performance, and view revenue analysis in real time.



Şekil 1 Login Page

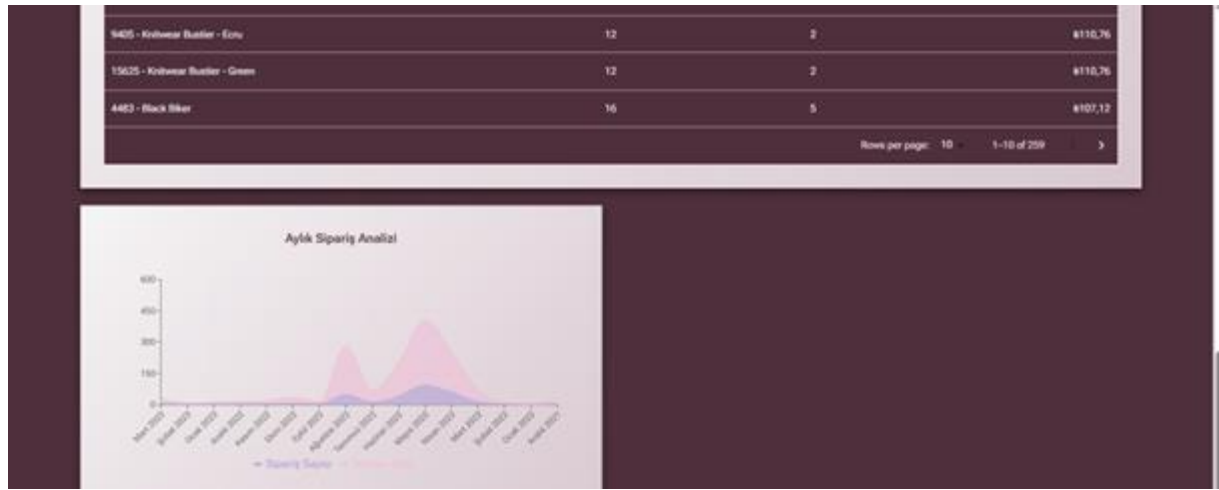
The system features a JWT-based secure authentication mechanism. Suppliers can only view their own data by logging into their own accounts. The dashboard provides monthly sales charts, product-based performance metrics, total revenue calculations, and detailed statistics.



Şekil 2 Dashboard

Ürün Satış Detayları			
Ürün Adı	Toplam Adet	Sipariş Sayısı	Toplam Satış (₺)
9641 - Natural Jacket	12	3	\$169,80
9278 - Crop Shirt - Light Green	8	1	\$122,96
12290 - Pants - Gray	12	1	\$119,28
16539 - Dress - Mustard	4	1	\$116,60
15630 - Bustier - Light Green	16	2	\$114,72
9369 - Jumpsuit - Black	9	1	\$114,39
16561 - Shorts Skirt - Pistachio Green	9	3	\$111,20
9405 - Knitwear Bustier - Ecu	12	2	\$110,76
15625 - Knitwear Bustier - Green	12	2	\$110,76
4483 - Black Bikini	16	5	\$107,12

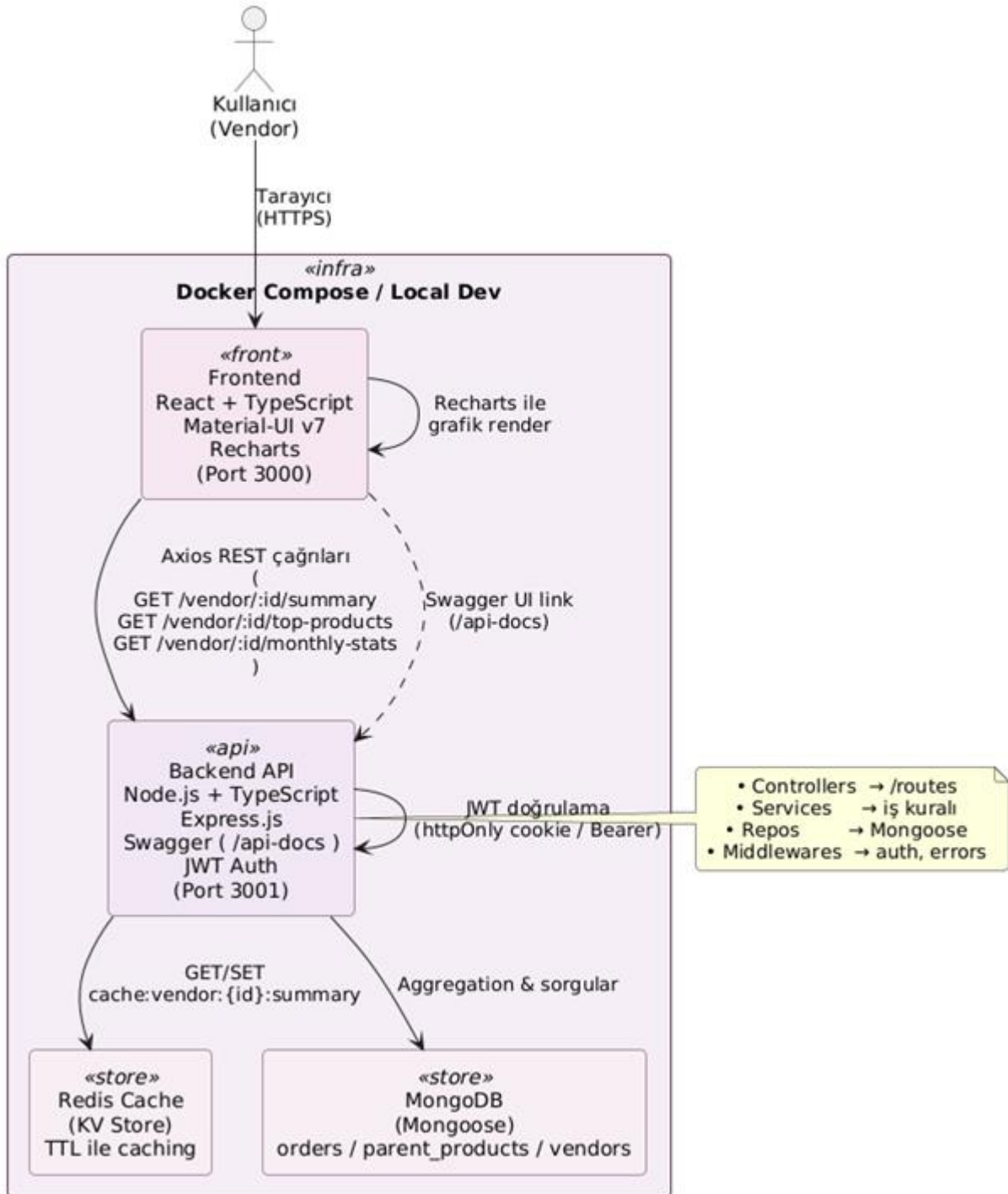
Şekil 3 Dashboard



Şekil 4 Dashboard

2. System Architecture

Backend services were developed using TypeScript on Node.js v18+. A RESTful API architecture was established using the Express.js framework. The async/await pattern was implemented for asynchronous operations, and a centralized middleware structure was created for error handling.



Şekil 5 System

A JWT (JSON Web Token)-based stateless authentication system has been implemented. Tokens are created with a 7-day expiration time. Password hashing is performed using the

Bcrypt algorithm. HTTP header security is ensured with Helmet.js, and CORS policies are configured in detail. Express-rate-limit protects against DDoS attacks by applying rate limiting to API endpoints.

Type-safe schema models were created with Mongoose ODM. Reference relationships were established between the Order, Product, and Vendor collections. Accuracy in financial calculations was preserved with the Decimal.js library. Query performance was optimized with Compound indexes.

Performance was improved by integrating a Redis 7 cache server. A cache invalidation strategy was implemented using TTL (Time To Live) values. Isolation was achieved with a vendor-based cache key structure. A 5-minute cache period was set for frequently used analytical data.

Interactive charts were developed using the Recharts library. Responsive design ensures mobile compatibility. Line chart, bar chart, and pie chart components were customized.