

CASE STUDY: ZENFULFILLMENT

Connecting online stores to warehouse logistics through one platform

Partner information:

Partner: Zenfulfillment

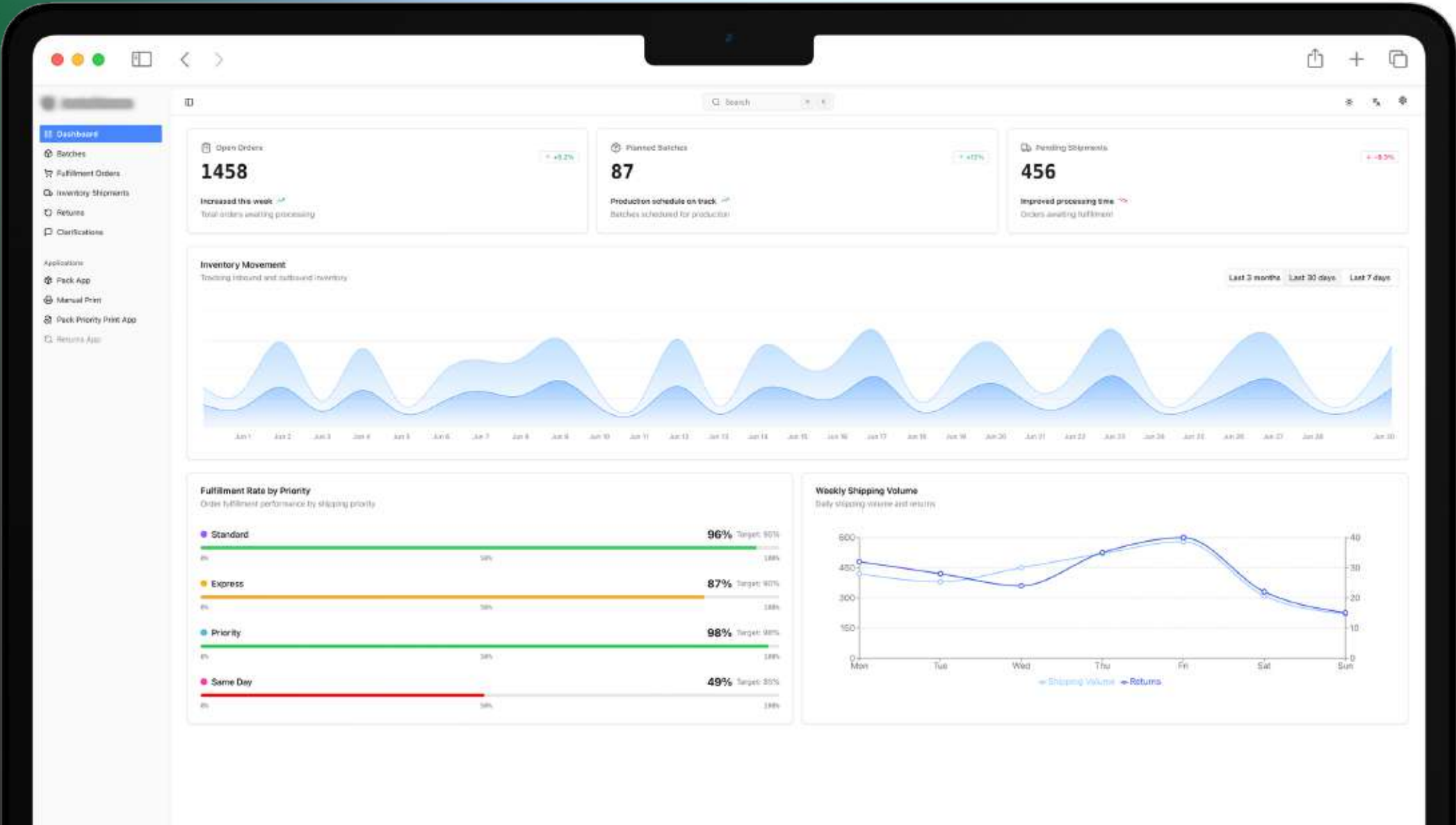
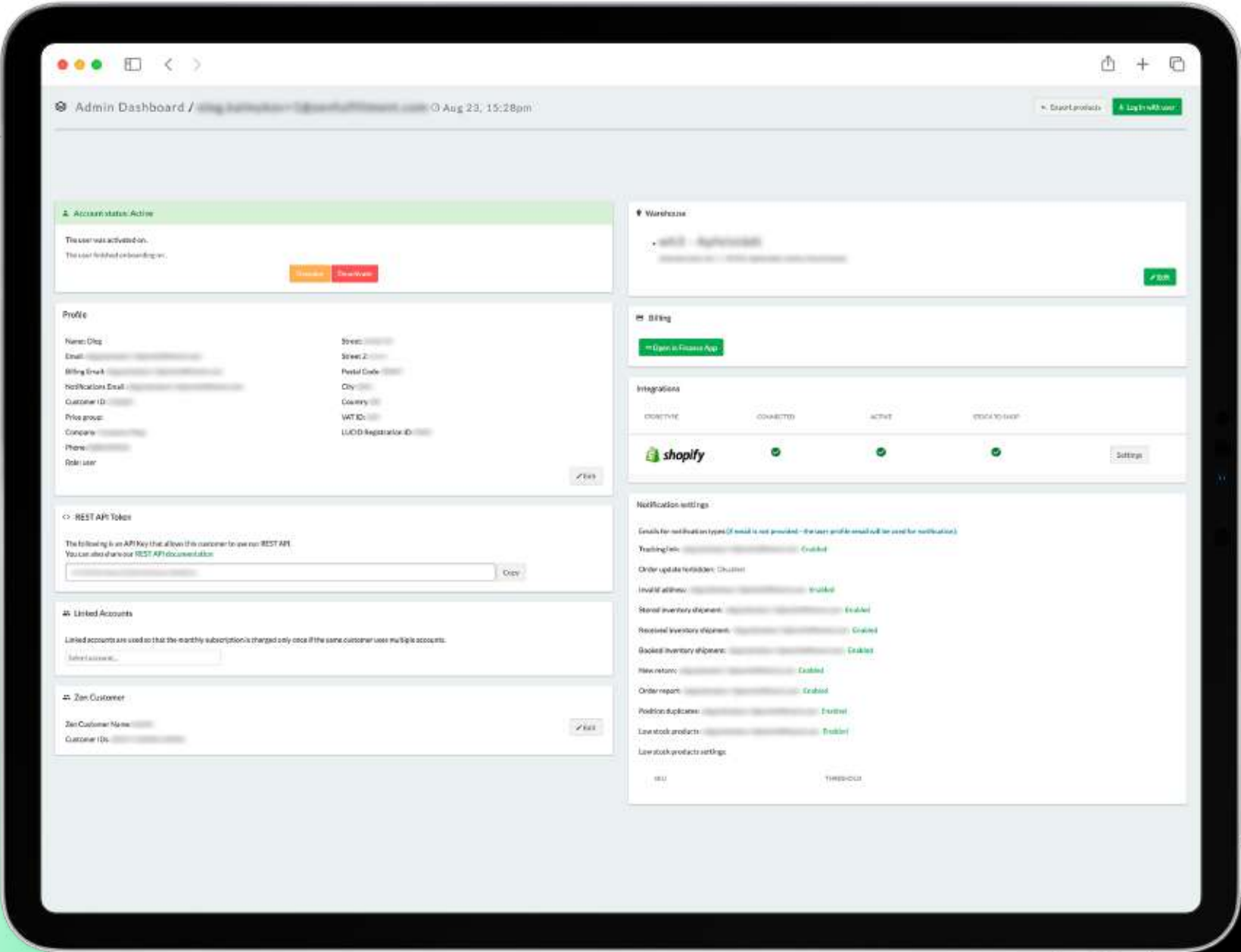
Company size: 50-201

Headquarters: Germany

Team size:

Full-stack developers: 2

DevOps engineer: 1

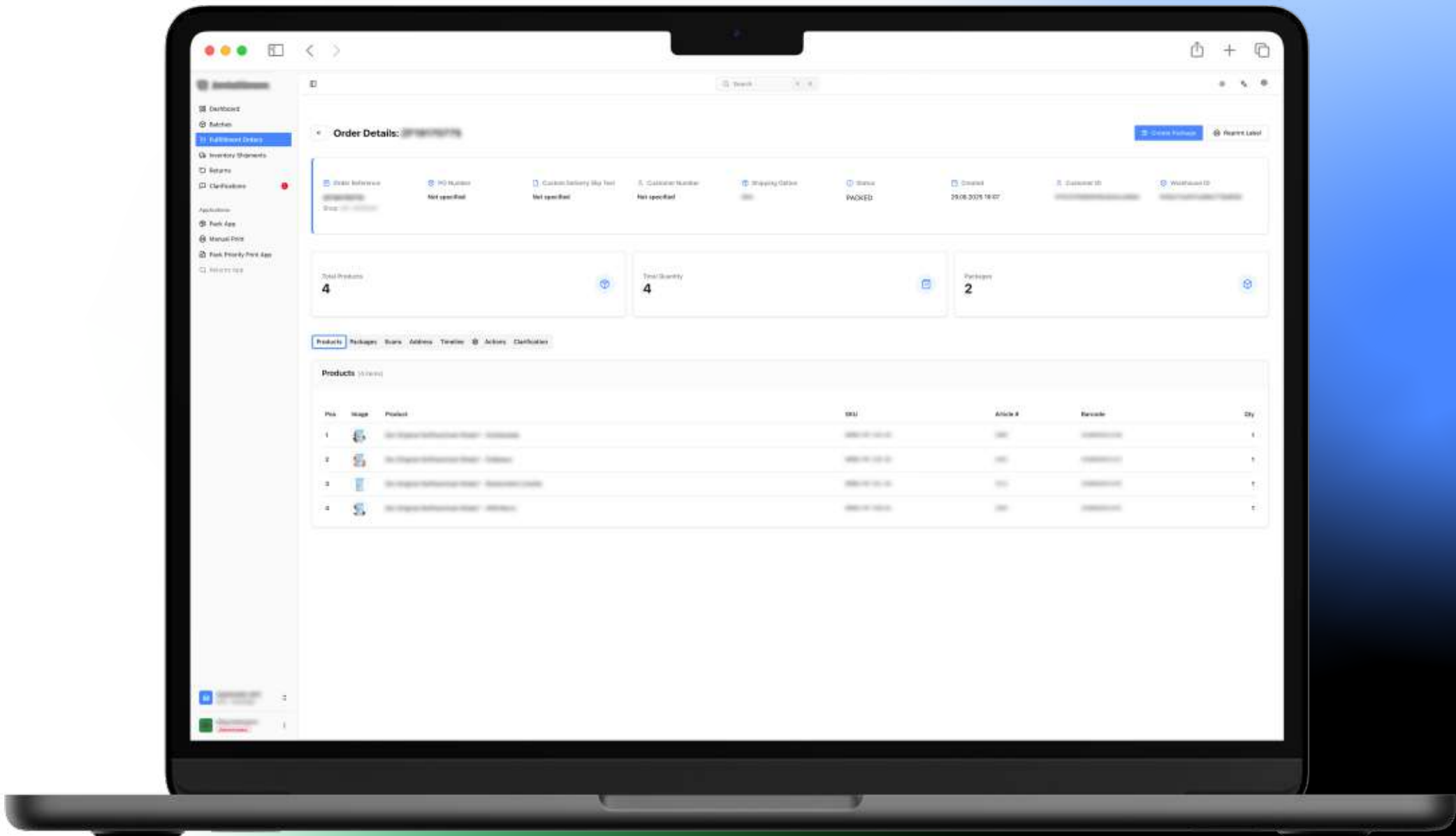


PROJECT OVERVIEW

Zenfulfillment is a logistics company providing warehousing and fulfillment services for ecommerce businesses. Their platform integrates with popular online store systems like Shopify and WooCommerce to automate order processing and logistics workflows.

But initially, it faced major maintenance challenges. Limited monitoring capabilities, performance bottlenecks, and excessive reliance on third-party services for key warehouse processes were major blockers.

To address these challenges and lay the groundwork for future growth, Zenfulfillment partnered with our team. We became an integral part of their development efforts, contributing to backend and frontend engineering, DevOps and infrastructure improvements.



Key parts of our collaboration involved ongoing warehouse management system (WMS) development, streamlining packaging and order flows, developing customer-specific features and supporting critical integrations.

And we continue working together today, constantly improving their platform as Zenfulfillment grows.

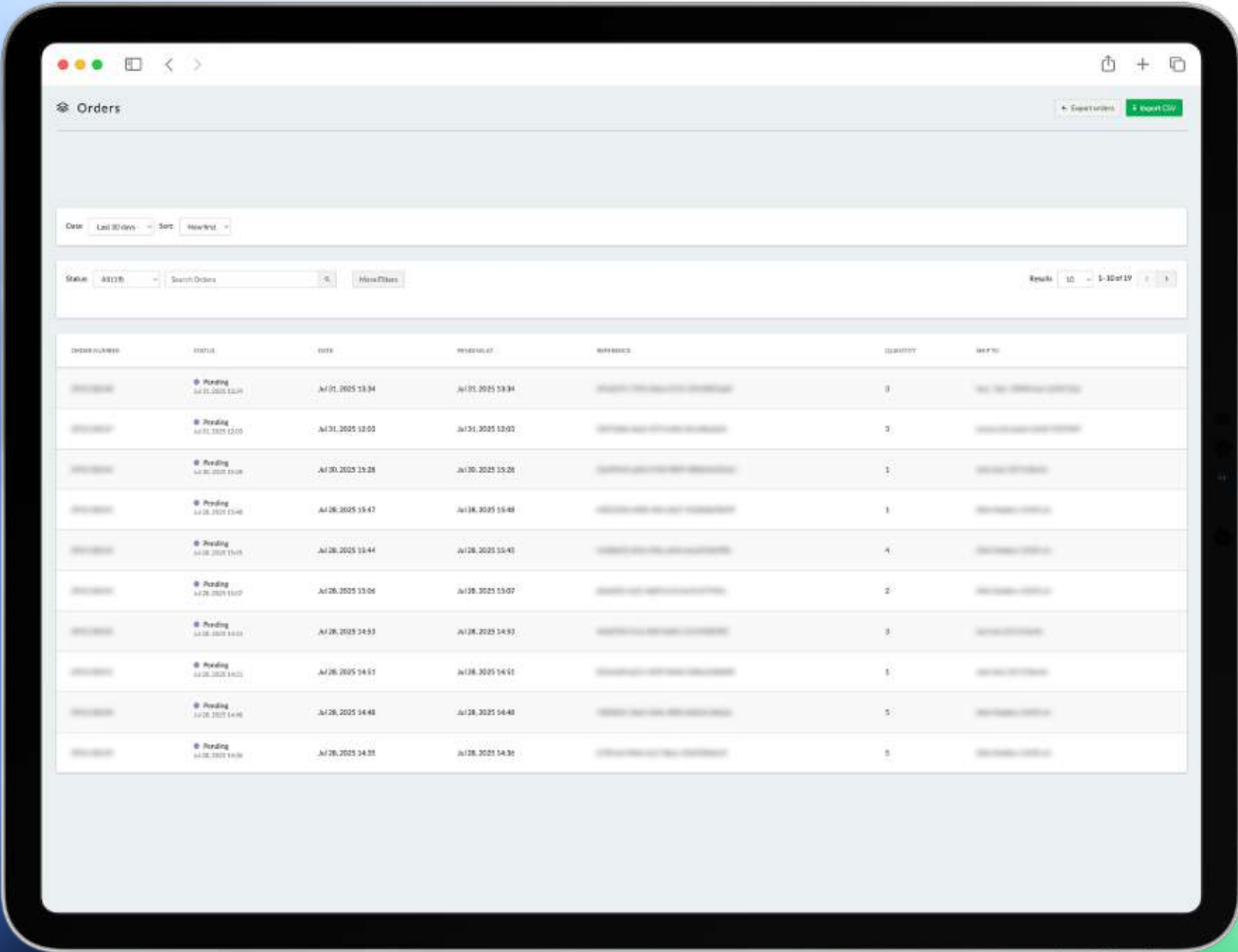
SERVICES & TECHNOLOGY STACK

Provided services

- Backend development
- Frontend development
- DevOps & infrastructure support
- Maintenance

Technologies & tools

- **Frontend:** Angular JS, Next.js, React-router, React
- **Backend:** Node.js, TypeScript (TS), Hapi, Lodash, Mongoose, GraphQL, RabbitMQ, Redis, NestJS, Nx
- **DevOps:** Terraform, GKE, HTTPS Load Balancer, CircleCI, OpenTelemetry, Prometheus, Grafana
- **Third-party tools:** MailGun, Shopify, WooCommerce, ReyBex, BillBee, Docker, Google Cloud Auth, SendIt



REQUIREMENTS & CHALLENGES

When we started working with Zenfulfillment, their focused team of just three engineers was managing nearly 40 repositories—handling everything from customer support to infrastructure. As a growing company, they needed supplemental DevOps expertise to support their expansion.

Our DevOps specialist joined their team to establish monitoring, automation and GCP deployments. As our partnership developed, two of our full-stack developers integrated into their daily operations, supporting ongoing development work.

Over time, we took on more responsibility—from backend and frontend tasks to building features tied directly into warehouse operations. This independence gave our partner’s engineers the space to focus on strategic planning while we handled daily development work.

Among these tasks were:

Reworking infrastructure and deployment processes

Replace manual deployments with automated DevOps infrastructure using Infrastructure-as-Code (IaC), comprehensive monitoring, logging and autoscaling to improve system reliability and reduce operational overhead.

Expanding the custom warehouse system

Extend Zenfulfillment's internal warehouse software to reduce reliance on outside tools, including barcode scanning workflows, batch handling and label printing.

Maintaining and upgrading ecommerce platform integrations

Support and improve integrations with Shopify, WooCommerce and XML feeds, including API updates, error handling and data syncing.

Implementing customer-specific customizations at scale

Build flexible service logic to handle customer-specific requirements—such as packaging methods, label formats, pricing rules and dynamic business workflows—while maintaining system scalability.

Refactoring and maintaining legacy codebases

Improve existing code quality, reduce technical debt, and improve maintainability to support rapid feature development and enterprise client requirements.

Migrating API integrations

Refactor their Shopify API integration by switching from REST to GraphQL to improve performance and data handling efficiency.

HOW WE WORK



Collaboration approach & process

Our team initially partnered with Zenfulfillment to address urgent technical needs. As we **earned their trust through close collaboration and reliable results**, our partnership deepened: two more engineers came on board, and our involvement expanded across core parts of their platform.

Instead of a layered setup, our developers **worked directly with stakeholders**, speeding up feedback and reducing delays. Daily standups kept the team aligned, while a flexible sprint process made it easier to **respond to urgent customer needs**.

Our team gained direct access to customer accounts for efficient troubleshooting, **supporting our partner’s customer-centric development philosophy**.

STEP-BY-STEP PROJECT FLOW

Stage 1: DevOps onboarding and infrastructure overhaul

Our collaboration began with a request to fix and automate Zenfulfillment's deployment process. Our team started by replacing their manual setup with automated infrastructure using Terraform and built monitoring dashboards with Grafana. We also improved network performance and merged five AWS accounts into one production system.

Stage 2: Reviving and extending their internal WMS

Zenfulfillment had previously initiated development of custom warehouse software but paused the project due to resource constraints. Our engineers helped their team resume active development, building barcode scanning for packing stations, batch handling and label printing.

We rolled out the new system across multiple warehouse stations, supporting configurable workflows that adapt to each station's layout, tools, and procedures. This also marked the start of reducing reliance on external services. Simultaneously, we began developing internal tools like color-coded batch tags to help staff prioritize tasks.

STEP-BY-STEP PROJECT FLOW

Stage 3: Maintaining and upgrading ecommerce platform integrations

Our team supported their Shopify, WooCommerce and XML-based connections, adapting to API changes and improving error handling. In some cases, we reworked key flows, for example, rewriting the Shopify integration using GraphQL, for better performance. We tailored each integration to client-specific logic, including shipping rules, weight-based pricing and preferred carrier selection.

Stage 4: Delivering client-specific functionality across the platform

With a client base that includes large ecommerce brands and complex fulfillment setups, Zenfulfillment prioritized advanced features to support their most demanding use cases.

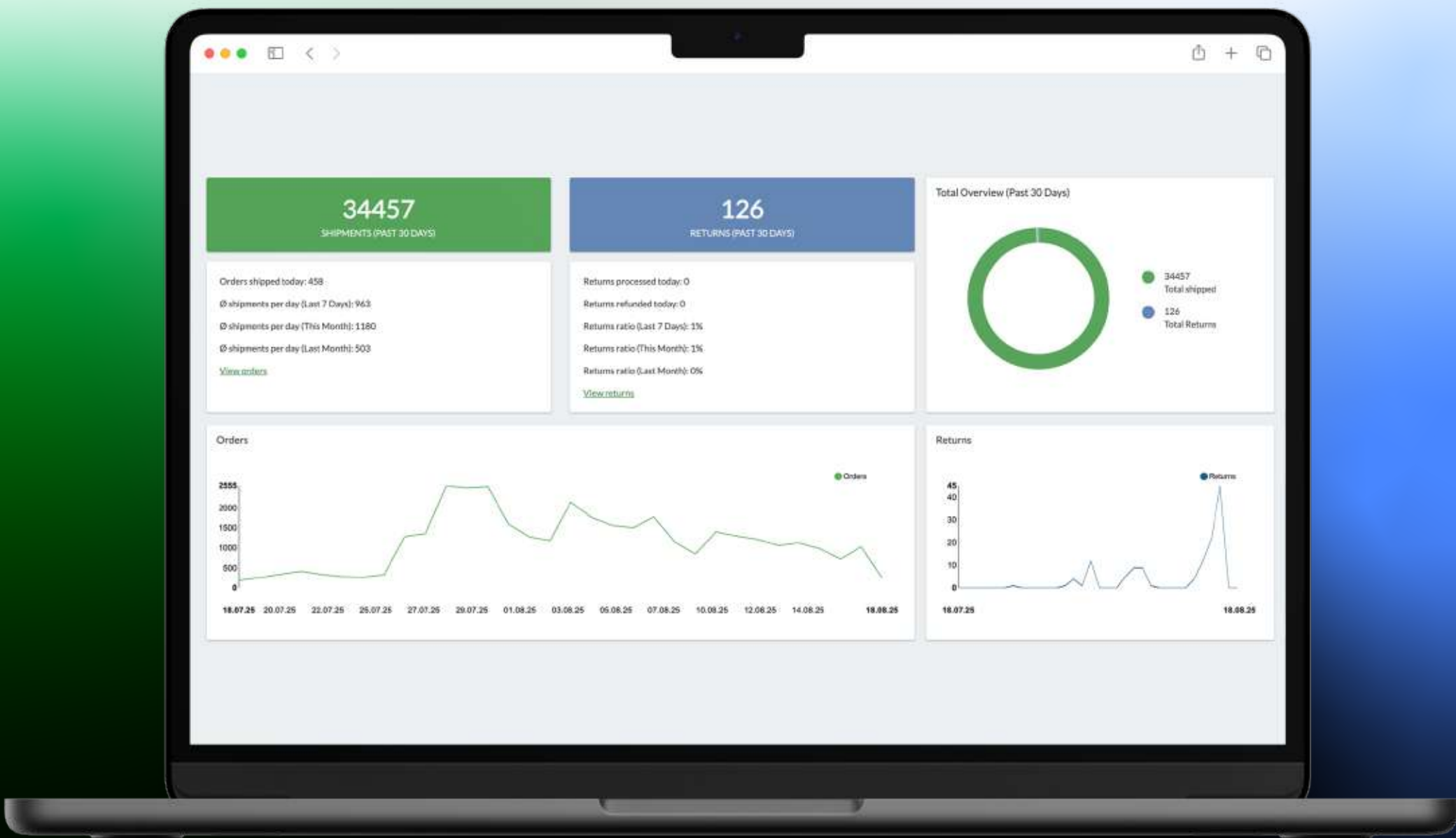
To meet their needs, we developed dynamic pricing updates, better error messages (e.g., for address validation), informational banners for the customer portal and packaging logic adapted to individual client needs. These improvements reduced support requests and gave non-technical users the tools to resolve issues independently.

Stage 5: Refactoring and maintaining legacy codebases

As business grew, parts of their platform required updates to handle increased volume and complexity. We refactored legacy components built with AngularJS, Hapi.js and early Next.js to eliminate bottlenecks.

This included fixing major UI freezing issues by adding virtualization to the admin panel, correcting pricing logic bugs, and improving slow database queries that were affecting core workflows.

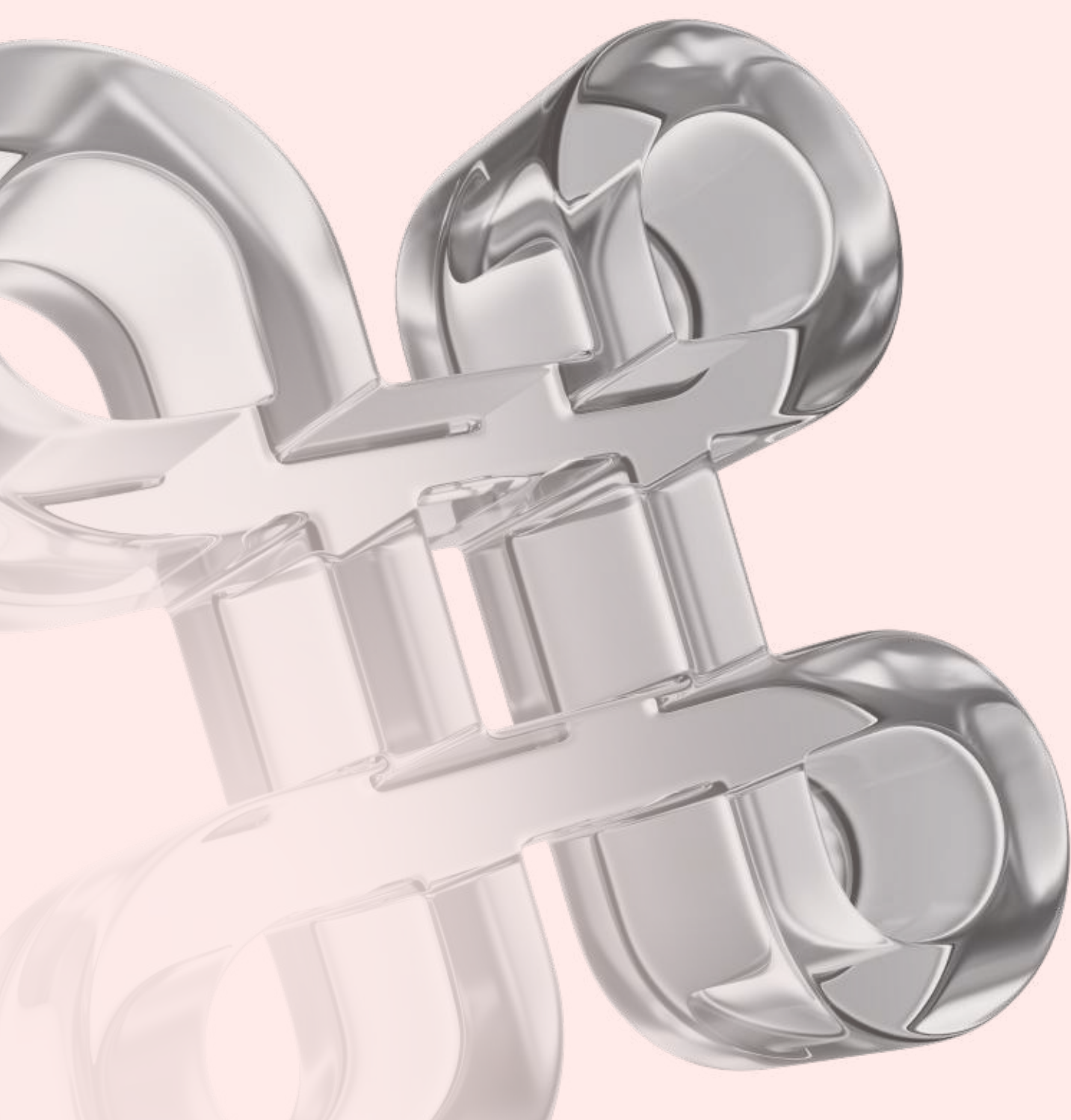
DELIVERED PRODUCTS & FUNCTIONALITY



- **Customer dashboard upgrades:** Built an advanced analytics dashboard that provides real-time tracking of 30-day shipments, returns and orders with clear filtering options across all pages. Customers can now track fulfillment performance and identify issues before they impact their operations.
- **Admin panel improvements:** Added customer impersonation, built in support ticketing and simplified user management in the admin panel. Now, support teams can securely view a customer's dashboard, troubleshoot more effectively and resolve issues faster.
- **Warehouse Management System (WMS):** Replaced Google Scripts and spreadsheets with a unified app that automates barcode scanning, batch processing and label printing. Staff can quickly resolve discrepancies through the clarification feature, process multiple orders simultaneously and make corrections without switching between systems.
- **Custom integrations:** Built enterprise-grade features like automated pickup coordination and tailored workflows for large clients. By improving data exchange between Zenfulfillment and platforms like Shopify, WooCommerce, ReyBex and BillBee—and integrating data from existing warehouse scanners—we reduced manual input and enabled faster, more reliable order processing.
- **Automated infrastructure:** Stabilized the platform with a Google Kubernetes environment with automatic deployments and monitoring. Engineering teams can manage infrastructure changes via code, respond to incidents faster, and release updates more frequently without downtime or manual server work.
- **Improved UI responsiveness:** Delivered faster, more responsive interfaces for warehouse and admin teams. Large datasets no longer freeze the UI thanks to virtualization improvements, and better address validation and pricing logic makes international orders easier to process.

PROJECT IN NUMBERS

Our team continues to strengthen their platform. Better infrastructure provides smooth, automated workflows and faster onboarding for complex clients—freeing Zenfulfillment to invest in expansion rather than maintenance.



50%

lower infrastructure costs through Google Cloud optimization and auto-scaling



14

warehouse packing stations now running the new WMS software

<1 HOUR

average incident response time (down from days of undetected downtime)

60-70%

reduction in daily support tickets through better error messages and UI improvements

PROJECT OUTCOMES

Scalable ops and growth readiness Zenfulfillment can now handle significantly larger client volumes without proportional increases in operational overhead. The platform supports over 20,000 monthly shipments across multiple regions and smooth onboarding of enterprise clients, regardless of complexity. Competitive advantage in enterprise markets Advanced customization capabilities help Zenfulfillment win and retain large ecommerce brands that require specialized fulfillment workflows. Now the platform adapts to the complex requirements unique to each client without custom development. Better client satisfaction, retention and positioning Faster issue resolution, plus improved system reliability and user experience, have strengthened client relationships. The platform now provides enterprise-grade service that supports Zenfulfillment's premium positioning.	Lowered costs with smarter resource allocation Infrastructure improvements reduced operational costs by 50% while eliminating the need for manual DevOps work. The streamlined support process cut overall daily ticket amounts by 60-70%, freeing technical teams to spend time on strategic development rather than putting out fires. Accelerated innovation and time-to-market Development velocity increased significantly with stable infrastructure, automated deployments and reduced technical debt. New features can be rolled out faster to meet changing client needs and market demands. Future-proof architecture Zenfulfillment's new modular system makes it easier to add new features, connect with other tools and adjust to changing requirements—all without rebuilding the platform.
--	--



aimprosoft

contact@aimprosoft.com