

Medusa JS Roadmap (6 Weeks)

Week 1: Getting Started with Medusa Core

Duration: ~7 days

Topics Covered:

- What is Medusa, & architecture overview (modular commerce, server components) [npm](#) +15 [Reddit](#) +2
- Installing & configuring the Medusa backend
- Overview of **Cart**, **Product**, **Customer**, **Order**, and **Inventory** modules [docs.medusajs.com](#) +2 [Medusa](#) +15
- Running Medusa server and connecting to admin UI

Goal: Get a Medusa server running with Admin dashboard.

Week 2: Commerce Modules & Inventory Management

Duration: ~7 days

Topics:

- Deep dive into Product, Pricing, Payment, and Promotion modules [Medusa](#) [Reddit](#)
- **Multi-warehouse inventory**, **Nested categories** for product hierarchy [Medium](#) +8
- Managing stock, categories, channels, and price lists in Admin

Goal: Real-world inventory control and commerce feature configuration.

Week 3: Medusa Framework & Customization

Duration: ~7 days

Topics:

- Extending Medusa: Admin Extensions (widgets, UI routes) and custom modules [Medusa](#) +15
- Using **Subscribers** to listen to events (eg: order.placed) and **Scheduled Jobs** (cron automation) [Reddit](#)
- Creating custom **API Routes** within Medusa server for custom logic [Reddit](#) [Medusa](#)

Goal: Learn how to customize backend behavior and UI.

Week 4: Workflows, EventBus & Composable Features

Duration: ~7 days

Topics:

- Understanding the **Workflows SDK** to orchestrate complex processes with retries/rollbacks [Reddit](#) +6
- Configuring **EventBus Modules** (Redis, RabbitMQ, Kafka, etc.) [Medium](#) +1
- Using **Link Modules** and **Index Engine** for cross-module querying [Medusa](#) +1

Goal: Build advanced reliable workflows and modular integrations.

Week 5: Building a Storefront with Medusa

Duration: ~7 days

Topics:

- Setting up a frontend: Using **Next.js** storefront starter or custom build [Reddit +1](#)
- Configure **store CORS**, publishable API keys, sales channels [docs.medusajs.com +2](#)
- Implement product listing, cart, checkout, customer profiles, and orders in storefront [docs.medusajs.com](#) [docs.medusajs.com](#)

Goal: Connect backend to frontend storefront and complete purchasing flows.

Week 6: Performance & Ecosystem Optimization

Duration: ~7 days

Topics:

- Use **Product Module** for serverless performance and Next.js integration [Reddit +4](#)
- Optimize performance: Medusa 250× speed improvements in recent releases, v2 modular core [Reddit +2](#)
- Final end-to-end project: deploying backend (self-host or cloud), deploying Next.js storefront (Vercel), and connecting modules

Goal: Launch a full-stack Medusa store optimized for scale and speed.

Summary Table

Week	Focus	Key Topics
1	Medusa fundamentals	Core modules, Backend + Admin
2	Commerce & Inventory	Products, Pricing, Promotions, Multi-warehouse
3	Customization	Admin extensions, events, API Routes
4	Engineering advanced flows	Workflows, EventBus, Index Engine
5	Build storefront UX	API keys, storefront functions, UI flow
6	Production-ready	Product module, performance, deploy

Community Insights & Future Roadmap

- Medusa 2.0 introduced **17 independent commerce modules**, a revamped modular architecture, redesigned Admin with Vite + React and built-in customization workflows
[Reddit](#) +8 [Medusa](#) +6 [docs.medusajs.com](#) +2
 - Previous and upcoming roadmap items include **Multi-warehouse API**, **Nested categories**, **Payment Processor API**, **Admin extensibility**, **Workflow orchestration**, **EventBus integration**, and automated API specs/typing tools [Medium](#) +1
-

✅ Practice Project Ideas:

- Multi-channel Medusa store (web + mobile) with purchase, customer profiles
- Personalized storefront using Product Module and Next.js serverless
- Custom admin widget / extension (e.g. bulk editor, specialized dashboard)
- Workflow implementation: automate fulfillment or order sync to ERP