

## Projects Description

Detailed description of every project during the entire roadmap.

# BLOCK 1 — JOB-FOCUSED FULL STACK ENGINEER (17 Weeks)

Master JavaScript, TypeScript, React, backend APIs, AI integrations, testing, and scalable deployment — everything needed to become job-ready for top product companies.

## THASE 0 — SURVIVAL SETUP (WEEK 0)

**Goal:** Set up your engineering environment like a real professional.

#### Week 0:

- Git, GitHub, Markdown, Conventional Commits
- Dev Diary Setup (Notion or Markdown)
- VSCode Extensions: Prettier, ESLint, GitLens, Tailwind

- Chrome DevTools Mastery
- Git Branching Strategy + PR Simulation
- README-first workflow initiation
- Final Setup Review + Dummy PR

Project: Developer Portfolio Bootstrap

**Deadline:** 1 week

**Assigned by:** CTO onboarding checklist for new team members

**Description:** Simulate setting up your personal dev environment like an internal company repo. Mimic a real onboarding experience where you're expected to prepare your development environment, documentation habits, and workflows before touching codebases.

**Outcome:** GitHub portfolio base with structured commit history and project diary, local + cloud dev setup.

Stack: Git, Markdown, GitHub, VSCode

**Concepts:** Git branching, PR simulation, setup automation, markdown docs

## PHASE 1 — MODERN JAVASCRIPT SYSTEMS (WEEK 1–4)

#### Week 1:

Project: InsightBoard — Async Standup Dashboard

**Deadline:** 1 week

Assigned by: Internal DevOps Lead

**Description:** Build a dashboard where team members asynchronously post their daily updates. Simulates a tool used in remote-first SCRUM teams. Includes JS logic for dynamic updates, validations, and persistent storage.

Outcome: Daily standup dashboard with add/view logs, localStorage, basic filters

Stack: HTML, Vanilla JS, CSS

Concepts: Arrays, DOM manipulation, object handling, JS loops, event handling

#### Week 2:

Project: InvoiceGen — Invoice Generator Tool

**Deadline:** 1 week

**Assigned by:** SME product manager simulation

**Description:** An SME wants an internal invoice generation tool that calculates tax, formats line items, and allows saving/printing. Think like a solo founder building a mini financial tool.

mini financial tool.

Outcome: Web app that generates PDF-ready invoices

**Stack:** HTML, JS, Bootstrap

Concepts: Form validation, event handling, dynamic UI creation, fetch API

#### Week 3:

Project: MediScan — Hospital Shift Tracker

**Deadline:** 1 week

Assigned by: Simulated hospital IT team

**Description:** Manage shift schedules for nurses and doctors. Allows calendar view, editable shifts, and export to PDF for record keeping. Real-world utility in hospital scheduling.

**Outcome:** Shift management app with export view

Stack: JS, HTML, DOM API

**Concepts:** Date objects, array filtering, print-ready styling

#### Week 4:

Project: CertiTrack — HR Certification Tracker

Deadline: 1 week

**Assigned by:** HR operations manager

**Description:** Track team certifications with expiry dates and reminders. Realistic internal tool request. Focus on building reusable components and notification logic.

Outcome: Certificate dashboard with add/edit/delete and expiry alerts

Stack: HTML, JS, Tailwind CSS

Concepts: CRUD logic, date comparison, localStorage

# **V** PHASE 2 — TYPESCRIPT + TESTING + GIT MASTERY (WEEK 5–8)

#### Week 5:

Project: NotesPro — Typed Notes App

**Deadline:** 1 week

**Assigned by:** Engineering productivity team

**Description:** Build a secure, type-safe note-taking app to demonstrate mastery in

TypeScript types and Zod validation.

Outcome: Typed note-taking tool with validation and state handling

Stack: TypeScript, Zod, Tailwind CSS

Concepts: TS types, interfaces, union types, Zod schemas

#### Week 6:

Project: API Buddy — Schema Validated Fetch Tool

**Deadline:** 1 week

**Assigned by:** Internal tooling team

**Description:** Build a small tool to fetch, validate and show third-party API

responses using Zod schemas. Test-driven development required.

**Outcome:** Schema-validated API integration UI + tests

Stack: TS, Zod, Vitest, Fetch API

**Concepts:** API schemas, testing strategy, CI basics

#### Week 7:

Project: ContentSim — Markdown + PR Workflow Tool

**Deadline:** 1 week

Assigned by: Dev content team

**Description:** Create a markdown editor with preview + PR simulation to mimic

content creation in large codebases.

Outcome: Editor UI with PR-style diff preview and export

Stack: TypeScript, Markdown, Zustand

**Concepts:** State management, input validation, markdown parsing

#### Week 8:

Project: TestCalcLib — Fully Tested Math Library

**Deadline:** 1 week

Assigned by: QA Team Lead

**Description:** Create a test-first utility lib with full coverage, including CI feedback.

Build TDD discipline.

Outcome: 100% test coverage with CI setup

Stack: TypeScript, Vitest, GitHub Actions

Concepts: TDD, CI pipelines, edge case testing

## 

P Each project builds on frontend architecture using modern stacks and design systems like ShadCN, Zustand, Tailwind, Clerk.

#### Week 9: React + Zustand Foundations

Project: TweetView — Twitter-style Feed

Deadline: 1 week

Assigned by: Internal Product Prototype for Social UI Team

**Description:** 

You are tasked with building a simulated social media feed interface — a clone of Twitter's feed page. The component must support infinite scrolling via pagination logic. Users can like tweets, simulate comments (UI only), and scroll through a growing list of posts fetched from a mocked API or JSON file. Zustand will be used to globally manage feed state and interactions, such as likes and new tweet additions.

You will design the full UI using Tailwind and ShadCN components with full responsiveness and loading skeletons. This project simulates the frontend of a startup building a minimum viable social platform.

#### Outcome:

A fully interactive Twitter-style feed UI featuring paginated loading, persistent global state for likes/interactions, and scroll-aware rendering.

#### Stack:

React, Zustand, Tailwind CSS, ShadCN UI

#### **Concepts Covered:**

- Zustand global store architecture
- Infinite scroll logic using intersection observers
- Skeleton UI components for loading state
- Component structure and abstraction
- Responsive layout with Tailwind
- State management for like and interaction behavior

### Week 10: Component Systems

Project: UlForge — Design System Kit

**Deadline:** 1 week

Assigned by: Design systems team lead

**Description:** Create a sharable, reusable UI kit using ShadCN components and Tailwind. Include buttons, forms, modals, dropdowns, tooltips, etc. Structure the repo for scalable design system reuse.

Projects Description 6

Outcome: Production-grade component library integrated with Storybook

Stack: React, TypeScript, ShadCN, Tailwind, Storybook

Concepts: component theming, accessibility, atomic design principles

Week 11: Forms + UX

Project: Onboardly — Signup + Settings Flow

**Deadline:** 1 week

**Assigned by:** Product UX team

**Description:** Build a multi-step signup form with animations, real-time validation, error UX, and state persistence across steps. Integrate theme switching and accessibility handling.

Outcome: Interactive auth/onboarding UI with persistent form state

Stack: React, Clerk, Tailwind, Framer Motion

Concepts: step-based forms, UX flow, accessibility, dark mode

Week 12: Internal Dashboard

Project: OpsPanel — Internal Admin Dashboard

**Deadline:** 1 week

**Assigned by:** Internal ops team

**Description:** Build a simulated dashboard with cards, KPIs, chart widgets

(Recharts), and role-based access. Fake API integration and mock user switching.

Outcome: Role-gated admin interface with charts, tables, filters

Stack: React, Recharts, Zustand, Tailwind

Concepts: dashboard layout, conditional rendering, global state

Week 13: Final Polish

Project: PixelFix — UI Bug Tracker & Polish Tool

**Deadline:** 1 week

Assigned by: UI QA Lead

Projects Description 7

**Description:** Simulate a UI QA bug sheet and refactor + polish all existing UI, animations, and responsiveness. Create reusable layout helpers and page templates.

**Outcome:** Final version of frontend polished to ship-ready quality

Stack: All frontend stack to date

Concepts: refactoring, responsive tweaks, final UX tuning

## **№** PHASE 4 — BACKEND + AUTH + DB FOUNDATIONS (WEEK 14–17)

This phase introduces the foundations of backend APIs, database interaction with Prisma, authentication with Clerk, and clean architecture — simulating real-world backend workflows in modern full-stack apps.

## Week 14:

Project: Taskify API — REST Backend for Task Manager

**Deadline:** 1 week

Assigned by: Backend Engineering Manager for internal API service

#### **Description:**

You're building a standalone RESTful API to support a simple task management frontend. The API must support routes for tasks CRUD (Create, Read, Update, Delete), user-scoped access, and persistent storage using SQLite via Prisma. Use Express.js and apply route grouping, schema validation with Zod, and basic error handling.

#### Outcome:

Fully working Express REST API with user-based task management and clear route contracts.

#### Stack:

Node.js, Express, TypeScript, Prisma, Zod, SQLite

#### **Concepts Covered:**

- REST routes and route grouping
- Prisma models and migrations
- Type-safe controllers
- Error handling with try/catch
- Express middleware architecture

### Week 15:

Project: AuthVault — Full Auth System with Clerk

Deadline: 1 week

**Assigned by: Product Security Architect** 

#### **Description:**

Implement a complete auth flow in a Next.js app using Clerk. Include signup/login, email verification, protected routes, user session checks, and role-based UI logic. Add route guards and simulate behavior for both authenticated and unauthenticated users. This mirrors a real-world auth integration flow you'd perform for a SaaS app.

#### **Outcome:**

End-to-end authentication system with role-based layout control and secure routing.

#### Stack:

Next.js 14 (App Router), TypeScript, Clerk, Tailwind

#### **Concepts Covered:**

- Clerk Provider and user context
- Server-side and client-side protection
- Role-based access logic
- Layout guards and protected pages
- Sign-in/up with redirect control

## Week 16:

Project: DataDeck — Dashboard + Database Integration

Deadline: 1 week

**Assigned by:** Internal Tools Team for Ops Dashboard

#### **Description:**

Build a simple internal dashboard that connects to a Prisma database and visualizes metrics via charts and tables. Include features like data filtering, user-specific data views, and pagination. Recharts should be used for KPI charts. This simulates a tool often used by internal ops teams.

#### **Outcome:**

Interactive dashboard with persistent backend data and real-time views.

#### Stack:

Next.js, Prisma, Tailwind, Recharts, Zustand

#### **Concepts Covered:**

- SSR with Prisma
- · Zustand-based state logic
- Recharts setup
- Filterable + paginated tables
- Backend-to-UI data flow

### Week 17:

Project: FullStackFusion — End-to-End Integration Sprint

Deadline: 1 week

**Assigned by:** Technical Director of Fullstack Enablement Team

#### **Description:**

This capstone project combines everything from Phase 4. You'll build a full-stack task manager app with:

Next.js frontend

- Clerk auth
- Prisma backend
- Zustand for state
- Authenticated dashboard views
- Editable tasks
- Status filters
- Deployed backend on Vercel/Postgres

It simulates what a real-world dev is expected to build in a startup or product company setting.

#### **Outcome:**

Full-stack CRUD SaaS MVP with modern architecture, auth, and DB integration.

#### Stack:

Next.js 14 (App Router), Clerk, Prisma, Zustand, Tailwind, PostgreSQL (or SQLite), Vercel

#### **Concepts Covered:**

- End-to-end auth + data protection
- Database design and modeling
- Full CRUD integration
- API route organization
- · Role-based layout rendering
- Deployment and ENV management

# BLOCK 2 — FREELANCE DEVELOPER SIMULATION (12 Weeks)

Simulate a full-stack freelance career: build client projects, create a growth engine, and simulate working with real-world stakeholders.

## PHASE 1 — Freelance Foundations (Week 1–2)

Project: FreelanceOS — Freelance Business Toolkit

Deadline: End of Week 2

**Assigned by:** Yourself (building your own business system)

**Description:** Build a Notion-based "Freelance Operating System" that includes your client onboarding checklist, pricing tiers, portfolio links, gig tracker, contract templates, and outreach playbooks. This tool simulates what real freelancers use to manage their operations efficiently.

Outcome: A personal freelance operating system (Notion dashboard) + hosted

portfolio

Stack: Notion, GitHub, Linktree, Tailwind, Next.js

#### **Concepts Covered:**

- Branding and value proposition
- Service packaging and offer clarity
- Personal portfolio structure
- Pricing experiments + link hub

## PHASE 2 — Client Simulation Builds (Week 3–6)

Each project is a real client scenario you may face on Upwork, Toptal, or Fiverr. You act as a dev agency.

## Week 3

Project: LocalBiz Booster — SEO Website for Local Business

**Deadline:** 5 Days

**Assigned by:** Simulated client — a bakery owner seeking visibility

**Description:** Build an SEO-first, blazing fast landing site for a bakery business. Simulate a small business with no prior tech knowledge and deliver a site that loads fast, indexes well, and follows on-page SEO best practices.

**Outcome:** Deployed responsive business website optimized for mobile and search

Stack: Next.js, Tailwind CSS, React Helmet

#### **Concepts Covered:**

- Google indexing basics
- Mobile-first design
- Metadata, schema, and performance optimizations
- Lighthouse audit

## Week 4

Project: AppPromo Landing — Animated Pre-Launch Page

**Deadline:** 1 Week

Assigned by: Startup founder launching an Al app

**Description:** Design a modern, scroll-animated landing page with a CTA that captures email leads. You'll use motion libraries to simulate a beautiful experience similar to YC startup pages.

**Outcome:** Animated landing page with form submission

Stack: Next.js, Tailwind CSS, Framer Motion, EmailJS

#### **Concepts Covered:**

- Scroll-triggered animations
- CTA placement and lead capture
- Form submission logic
- UI/UX polish

## Week 5

Project: CRM Lite — Lead Tracker Dashboard

**Deadline:** 1 Week

**Assigned by:** Solo business consultant managing client leads

**Description:** Build a local-only CRM where leads can be added, updated, and filtered. Simulate working with a non-technical client who wants the app to "just

work" from a browser.

Outcome: A full CRUD system with contact tagging, notes, and filters

Stack: React, Zustand, Tailwind, LocalStorage

#### **Concepts Covered:**

Zustand for local state

- Filter and sort logic
- Form validation and modals
- Persistent storage in browser

## Week 6

Project: StripeFlow — Custom Checkout for SaaS

**Deadline:** 5 Days

Assigned by: SaaS startup on a budget needing billing integration

**Description:** Implement a Stripe-powered checkout flow with pricing tiers, coupon logic, and webhook mocks. You'll simulate real billing workflows that power actual products.

Outcome: Functional checkout UI + webhook simulation + thank-you page

Stack: Next.js, Stripe API, Tailwind, Supabase (mock DB), Webhooks

#### **Concepts Covered:**

- Stripe products + pricing
- Dynamic checkout session generation
- Webhook simulation using serverless API route
- Coupon + discount handling

## PHASE 3 — Freelance Growth Systems (Week 7–12)

In this phase, you simulate being your own growth engine. You'll launch campaigns, redesign your portfolio, systematize outreach, and build assets that attract clients 24/7.

## Week 7-8

Project: Client Magnet Kit — Cold Outreach Engine

Deadline: 2 weeks

Assigned by: Your internal "Brand Strategist" persona

**Description:** Build and test cold outreach systems to generate freelance leads. Design and A/B test 3 variations of a cold email sequence. Create a lead sheet with ideal clients from LinkedIn, Clutch, and X. Automate your process using tools like Hunter.io and GMass.

Outcome: Documented outreach funnel + client tracker + 10 responses target

Stack: Notion, Gmail, Hunter.io, Snov.io, Canva

#### **Concepts Covered:**

- Cold outreach copywriting
- CRM for tracking conversations
- Email deliverability and follow-ups
- A/B testing subject lines + CTA conversion

## Week 9

Project: Portfolio V2 — Visual Redesign + Niching

**Deadline:** 1 week

**Assigned by:** Growth consultant helping you niche down

**Description:** Redesign your freelance portfolio to highlight a specific niche (e.g., local businesses, SaaS landing pages, or Stripe integrations). Add case study pages, testimonials, and before-after comparisons. Focus on performance, SEO, and storytelling.

**Outcome:** Niche-focused portfolio optimized for conversion

Stack: Next.js, Tailwind CSS, Dev.to/Medium (for case study posts), Canva

#### **Concepts Covered:**

Case study writing and storytelling

Projects Description 15

- SEO optimization
- Visual identity and trust-building elements
- Niche targeting

## **Week 10**

Project: Gig Optimizer — Upwork + Fiverr Profile Revamp

Deadline: 1 week

Assigned by: Freelance marketplace strategist

**Description:** Research top freelancers in your niche and reverse-engineer their Upwork, Fiverr, and Contra profiles. Optimize your gig descriptions, cover letters, pricing packages, and create visual thumbnails for your gigs.

**Outcome:** High-converting freelance marketplace profiles

Stack: Upwork, Fiverr, Canva, Notion

#### **Concepts Covered:**

- Platform-specific copywriting
- Market positioning
- Visual differentiation
- · Gig funnel testing

## **Week 11**

Project: DevInfluence — X & Dev.to Authority Engine

**Deadline:** 1 week

**Assigned by:** Simulated mentor from dev influencer community

**Description:** Write and publish 3 Dev.to blogs based on your freelance work and post daily short-form content on X (formerly Twitter) to grow a personal brand. Experiment with threads, carousels, and thought leadership pieces.

**Outcome:** 3 blogs + 10 content posts with engagement tracking

Stack: Dev.to, X, Canva, Notion

#### **Concepts Covered:**

- Content repurposing
- Audience targeting and engagement
- Thought leadership strategy
- Analytics-based growth loops

## **Week 12**

Project: Systems Hub — Freelance Process Automation

Deadline: 1 week

**Assigned by:** Your internal COO

**Description:** Systematize your freelance work with automation — create email templates, project delivery pipelines, proposal templates, and time-blocking systems. You'll simulate running a 1-person dev agency.

Outcome: Streamlined Notion-based process automation toolkit

Stack: Notion, Calendly, EmailJS, Zapier

#### **Concepts Covered:**

- Freelance operations
- Automation with Zapier / scheduling
- Proposal + invoice templates
- Repeatable process documentation

# BLOCK 3 — AI SAAS PRODUCT BUILDER (16 Weeks)

Go from AI startup idea to working product with payment, auth, data pipelines, prompt workflows, and full launch systems. Simulates solo founder journey building AI-powered SaaS tools.

## PHASE 1 — Al Knowledge + Ideation (Week 1–2)

Project: PromptLab — Al Prompt Playground

Deadline: End of Week 2

Assigned by: Internal Developer Tools Team Lead

#### **Description:**

You're building an internal tool for engineers experimenting with AI prompts and use cases. This playground must allow saving, testing, versioning, and comparing prompt outputs using OpenAI's API. It's also used by prompt engineers in the company to iterate on customer-facing tools.

#### **Outcome:**

Prompt experimentation dashboard with prompt history, version control, and export capabilities.

#### Stack:

Next.js, Tailwind, Supabase, OpenAl API

#### Concepts:

Prompt engineering, completion tuning, API rate handling, UX for iteration, Supabase auth/storage

## THASE 2 — Al Product Build Sprints (Week 3–10)

Each 2-week sprint results in a production-ready Al SaaS MVP.



Project: LegalDoc AI — Contract Assistant

**Deadline:** 2 weeks

**Assigned by:** Legaltech Startup Founder

#### **Description:**

Build an Al-powered document assistant where users can upload legal contracts (PDFs) and ask questions about them. You'll use embeddings to create vector

representations and implement retrieval-augmented generation (RAG) for semantic search.

#### **Outcome:**

Upload → Process → Ask questions → Get answers flow with contextual accuracy

#### Stack:

Next.js, UploadThing, OpenAl, Pinecone, Supabase, Clerk

#### Concepts:

PDF parsing, RAG pipeline, vector search, prompt chaining, auth-protected uploads



Project: ResumeRanker — Resume Scoring Engine

**Deadline:** 2 weeks

Assigned by: HR SaaS Founder Persona

#### **Description:**

Develop an AI resume evaluator that takes a resume and job description and returns a score with suggestions. Use OpenAI for matching logic, and add exportable reports. Integrate input validation and prompt refinement.

#### Outcome:

Upload Resume + JD → Get ranked → View suggestions → Download report

#### Stack:

Next.js, PDF-lib, OpenAl, Tailwind, Prisma, Clerk

#### Concepts:

LLM feedback generation, PDF parsing, schema validation, prompt structuring, exports

₩eek 7-8:

Project: GhostWriterX — Al Ad Copywriter

Deadline: 2 weeks

Assigned by: Marketing Tech Lead

#### **Description:**

An Al-powered copywriting tool for generating Facebook and Google Ads using AIDA and PAS frameworks. You'll offer multiple tones, styles, and campaign hooks. Include live preview and export options.

#### **Outcome:**

Input offer → Select style → Get multiple ad variants → Save/export

#### Stack:

Next.js, OpenAI, Stripe, Markdown

#### **Concepts:**

Framework-based prompting, multi-response handling, Stripe plan gating, markdown editor



Project: SaaSKit — Al SaaS Boilerplate

**Deadline:** 2 weeks

**Assigned by:** Al Startup Accelerator

#### **Description:**

You are building the foundational SaaS boilerplate to rapidly launch future Al apps. It must include auth, billing, feature usage tracking, webhook handling, and a clean dashboard layout.

#### **Outcome:**

Plug-and-play SaaS starter with built-in auth, plans, usage limits, and templates

#### Stack:

Next.js, Clerk, Supabase, Stripe, Resend (email), tRPC

#### Concepts:

Authentication, usage limits, Stripe integration, reusable SaaS layout, API routes



Projects Description 20

Project: GoPublic Pipeline — Al App Launch System

Deadline: 6 Weeks

Assigned by: CEO and Growth Strategist Persona

**Description:** 

You will now take each AI app and build its product hunt launch page, email sequence, and public content funnel. Simulate startup growth from MVP  $\rightarrow$  market feedback  $\rightarrow$  v2 roadmap. This phase simulates real product market fit validation.

#### **Outcome:**

Launch-ready microsites, PH launches, email campaigns, post-launch feedback

#### Stack:

Notion, ProductHunt, X (Twitter), Revue, PostHog, EmailJS

#### **Concepts:**

Launch strategies, growth funnels, tracking, user onboarding, testimonials, analytics loop

# BLOCK 4 — DEVOPS & SYSTEMS THINKING (12 Weeks)

Master scalable deployments, performance monitoring, automation, and system design principles — simulating work as a senior engineer managing production-grade software systems.



Week 1:

Project: Cl Wizard — GitHub Actions Pipeline Setup

Deadline: 1 week

**Assigned by:** Internal Tooling Team

**Description:** 

Set up a full CI pipeline for any of your previous projects. Include linting, testing, type checks, and build steps. Must simulate a production CI config using GitHub Actions.

#### Outcome:

Complete CI pipeline that runs on every PR + branch merge

#### Stack:

GitHub Actions, Node.js, Vitest, ESLint, TypeScript

#### **Concepts:**

Pipeline setup, job/matrix configs, caching, fail feedback loops, test coverage reports



Week 2:

Project: DevDocker — Containerized Dev Environment

Deadline: 1 week

**Assigned by:** Infrastructure Architect

#### **Description:**

Dockerize a full-stack app (frontend + backend + DB) using multi-stage builds. Add .env support, volume mapping, and optimized dev workflow.

#### **Outcome:**

Run full app stack using a single docker-compose up

#### Stack:

Docker, Docker Compose, Node.js, PostgreSQL, Next.js

#### Concepts:

Container isolation, volume sharing, build optimization, network linking, dev images



Project: ImageVault — Custom Docker Registry + Cl Deploy

Deadline: 1 week

Assigned by: DevOps Lead

**Description:** 

Create a GitHub Action that builds and pushes your Docker image to GitHub Container Registry (GHCR) or DockerHub. Include proper tagging with version control and rollback capabilities.

**Outcome:** 

Docker images auto-built on PR merge with tags like v1.0.0, latest

Stack:

Docker, GHCR, GitHub Actions

**Concepts:** 

Image tagging, registry auth, versioning strategy, rollback workflow



Week 4:

Project: CloudForge — Fullstack Deploy + Domain Setup

**Deadline:** 1 week

Assigned by: DevOps Team

**Description:** 

Deploy a full-stack app (frontend + backend + DB) to production using either Render, Railway, or Fly.io. Add custom domain, HTTPS, and uptime monitoring.

**Outcome:** 

Fully deployed, live app with custom domain and HTTPS

Stack:

Render/Railway, Next.js, PostgreSQL, GitHub, UptimeRobot

**Concepts:** 

Domain linking, HTTPS setup, auto-deploy via GitHub, .env production handling

Week 5:

Project: PerfWatch — Performance Logger

**Deadline:** 1 week

Assigned by: Product QA Team

**Description:** 

Add performance metrics tracking to an app using tools like Vercel Analytics, PostHog, or custom logging. Track TTFB, load time, API latency, and frontend errors.

#### **Outcome:**

Analytics dashboard + alerting on app performance

#### Stack:

PostHog, Vercel Analytics, Sentry, Next.js

#### **Concepts:**

Web vitals, log aggregation, alert triggers, telemetry



Project: ErrorSleuth — Full Error Monitoring Setup

**Deadline:** 1 week

Assigned by: Site Reliability Engineer

#### **Description:**

Implement global error boundaries in your React apps. Setup error reporting using Sentry and a notification system via Slack or email.

#### **Outcome:**

Every crash/error is reported in real-time with logs and context

#### Stack:

Sentry, React Error Boundaries, Slack Webhooks, Resend

#### **Concepts:**

Error boundaries, try/catch, alert routing, developer observability

Week 7:

Project: DevOpsOS — Centralized Infra Dashboard

**Deadline:** 1 week

**Assigned by:** Head of Platform Engineering

**Description:** 

Build a dashboard that aggregates status from deployments, alerts, monitoring tools, and CI pipelines into one place.

#### **Outcome:**

Command center for your stack's health and status

#### Stack:

Next.js, Zustand, PostHog API, GitHub Actions API

#### Concepts:

Third-party API fetching, status polling, dashboard layout

## PHASE 3 — Systems Thinking + Optimization (Week 8–12)

₩eek 8-9:

Project: CacheRush — Optimized Blog API

Deadline: 2 weeks

**Assigned by:** Performance Optimization Squad

#### **Description:**

Refactor a blog API to include caching via Redis. Add stale-while-revalidate (SWR) logic and background regeneration on updates.

#### **Outcome:**

Faster API with caching and revalidation behavior

#### Stack:

Redis, Prisma, SWR, Next.js API Routes

#### **Concepts:**

#### Caching patterns, performance profiling, stale regeneration, SWR

Week 10:

Project: PlanOps — Scalable Feature Flag System

Deadline: 1 week

**Assigned by:** Feature Ops Team

**Description:** 

Implement a feature flag manager to control beta features in your product. Users in different plans or groups see different UI components.

#### **Outcome:**

Feature flag system integrated with role-based logic

#### Stack:

Next.js, Supabase, Clerk, Zustand

#### **Concepts:**

Plan detection, conditional rendering, role segmentation

₩ Week 11-12:

Project: SystemMastery — DevOps Knowledge Wiki

Deadline: 2 weeks

Assigned by: DevOps Education Squad

#### **Description:**

Create a documentation portal containing all your DevOps knowledge, terminal commands, CI/CD notes, and system diagrams. Simulate onboarding material for future devs.

#### Outcome:

Public or private DevOps wiki covering your full infrastructure knowledge

#### Stack:

Notion, Markdown, Mermaid.js, GitHub Pages

#### **Concepts:**

Knowledge sharing, internal wiki systems, diagramming, process documentation