Privacy-First Analytics Dashboard - Comprehensive Engineering Plan

1. Executive Summary

The Privacy-First Analytics Dashboard is a portfolio-grade full-stack application designed to demonstrate enterprise-level web analytics capabilities while maintaining strict GDPR compliance and privacy-by-design principles. The system tracks website usage through anonymized event logging, provides real-time visual dashboards with filtering capabilities, and integrates AI-powered insights via a RAG (Retrieval Augmented Generation) chatbot that queries historical analytics data using natural language. Built with Next.js 15, PostgreSQL with pgvector, and OpenAI integration, the platform targets EU compliance requirements (CSRD, GDPR) while showcasing modern web development practices including React Server Components, edge computing, comprehensive testing, and DevOps automation.

The project serves dual purposes: (1) demonstrating technical proficiency across full-stack development, AI integration, security, and compliance for recruitment in Nordic markets, and (2) providing a production-ready foundation for privacy-conscious analytics that can be extended to multi-tenant SaaS. Timeline estimate is 6-8 weeks for MVP with stretch goals extending to 12 weeks. The system prioritizes developer experience (one-click setup), operational excellence (observability, CI/CD), and measurable performance metrics (Core Web Vitals, accessibility scores) suitable for portfolio presentation.

2. Feature & Requirement Extraction

Functional Requirements

ID	Feature	Description	Source	Priority
FR-001	User Authentication	Support FIDO2 Passkeys and OAuth (Google/GitHub) via NextAuth for admin login	User Stories p.5, Security Model p.7	High
FR-002	Anonymous Event Tracking	Log pageviews, clicks, form submissions with anonymized visitor_hash, no PII storage	User Stories p.5, Data Model p.6	High
FR-003	GDPR Consent Management	Display cookie banner, only track after explicit opt-in consent	User Stories p.5, Privacy & GDPR p.7	High
FR-004	Analytics Dashboard	Display charts showing pageviews over time, top pages, user stats with date range filtering	User Stories p.5, System Architecture p.5	High

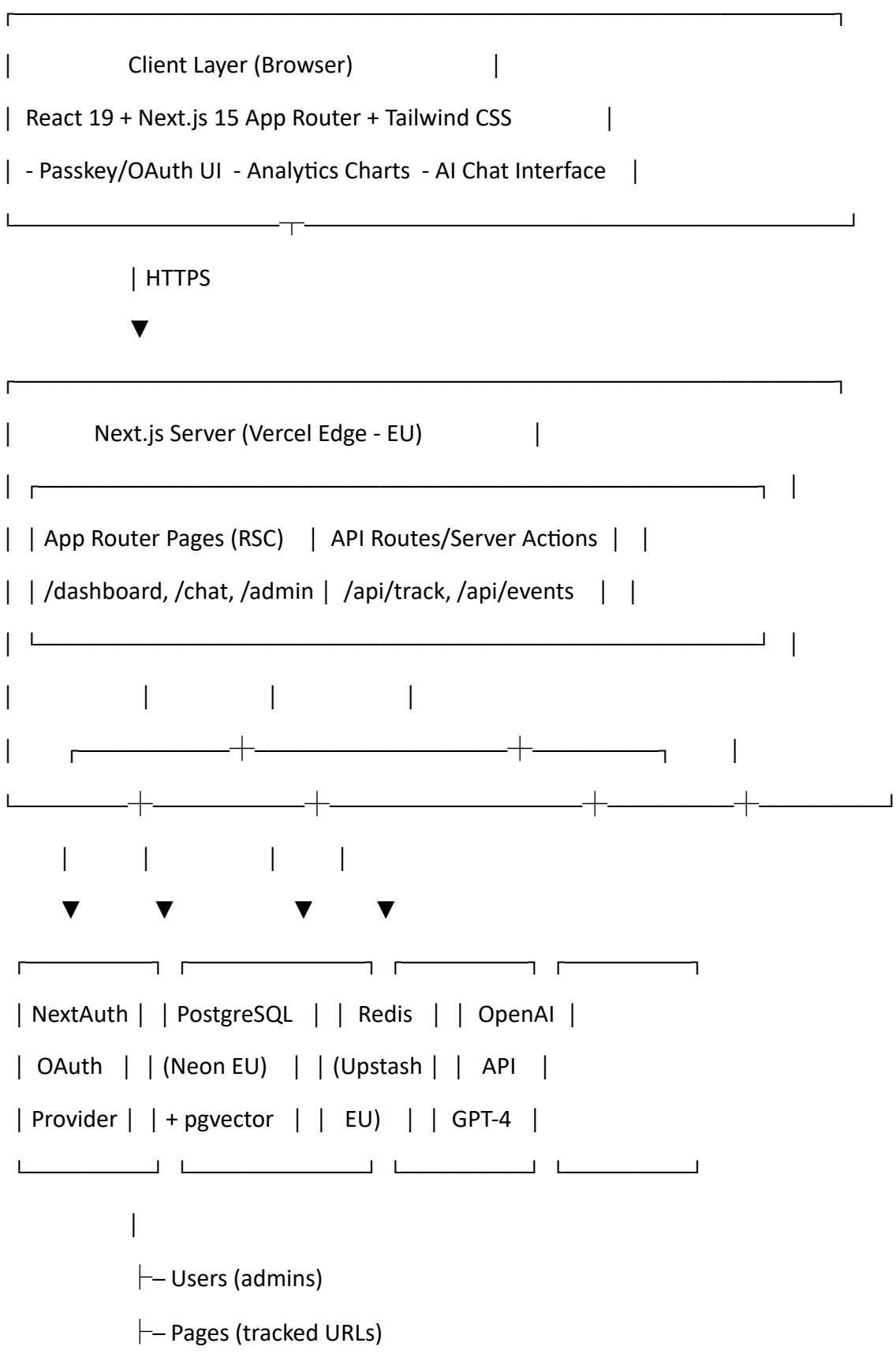
ID	Feature	Description	Source	Priority
FR-005	AI Insights Chatbot	Natural language query interface using RAG (pgvector + GPT-4) to answer analytics questions	User Stories p.5, AI System Design p.7	High
FR-006	Data Deletion UI	Admin interface to fulfill GDPR "right to erasure" requests, erasing all user-linked events	User Stories p.6, Privacy & GDPR p.7	High
FR-007	Role-Based Access Control	Admin (full access) vs Viewer (read-only) roles with permission matrix	Security Model p.7	Medium
FR-008	Multi-Language Support	English + Norwegian Bokmål with Next.js i18n routing (/nb/*)	Accessibility & I18n p.7	Medium
FR-009	API Documentation	Auto-generated OpenAPI/Swagger docs for all endpoints	API Design p.6, DevOps p.9	Medium
FR-010	Offline Dashboard Viewing	PWA with IndexedDB for viewing last-fetched data offline	Performance Targets p.8	Low
FR-011	Real-time Data Updates	Server-Sent Events or polling for live dashboard updates	System Architecture p.5	Low
FR-012	Multi-Tenancy (Stretch)	Support separate analytics projects for different customers	Stretch Goals p.9	Low

Technical Constraints

- **Platform**: Vercel Edge (EU region) or equivalent (Fly.io, Render) with EU hosting
- **Database**: Neon Serverless PostgreSQL with pgvector extension for vector embeddings
- **Libraries**: React 19, Next.js 15 App Router, Prisma, NextAuth, lucide-react, recharts, OpenAI SDK
- **No Browser Storage**: Cannot use localStorage/sessionStorage (Claude.ai limitation if demo'd there)
- **Node Version**: Node.js 18+ for Server Actions compatibility
- **AI Model**: OpenAI GPT-4 or local LLM with vector embedding support (1536 dimensions)

3. System Architecture

3.1 High-Level Block Diagram



├– Events (anonymized analytics)

└– EventEmbeddings (vectors)

Design Justification: Vercel Edge provides low-latency global CDN with EU region pinning for GDPR compliance. Next.js 15 Server Components minimize client bundle size while Server Actions eliminate API boilerplate. PostgreSQL with pgvector enables unified transactional + vector storage, avoiding separate vector DB costs. Redis caches aggregated stats to prevent DB overload on dashboard queries.

Alternative Design: Separate Node.js/Express backend + React SPA. Tradeoff: Better separation of concerns but loses Next.js SSR benefits, requires CORS configuration, and increases deployment complexity (2 services vs 1).

3.2 Data Model Summary

```
-- Users Table (Admins only)
CREATE TABLE users (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  email VARCHAR(255) UNIQUE NOT NULL,
  roles TEXT[] DEFAULT ARRAY['viewer'], -- ['admin', 'viewer']
  created_at TIMESTAMP DEFAULT NOW(),
  passkey_credential TEXT, -- WebAuthn credential ID
  oauth_provider VARCHAR(50),
  oauth_id VARCHAR(255)
);

-- Pages Table
CREATE TABLE pages (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  url TEXT UNIQUE NOT NULL,
  title TEXT,
  created_at TIMESTAMP DEFAULT NOW()
);
```

```
-- Events Table (Core analytics data)
CREATE TABLE events (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  page_id UUID REFERENCES pages(id) ON DELETE CASCADE,
  timestamp TIMESTAMP DEFAULT NOW(),
  event_type VARCHAR(50) NOT NULL, -- 'pageview', 'click', 'form_submit'
  visitor_hash CHAR(64) NOT NULL, -- SHA-256 hash of IP+UA (anonymized)
  user_agent TEXT,
  metadata JSONB, -- Extra event data
  consent_given BOOLEAN DEFAULT FALSE,
  INDEX idx_events_timestamp (timestamp DESC),
  INDEX idx_events_page (page_id),
  INDEX idx_events_visitor (visitor_hash)
);

-- EventEmbeddings Table (For AI RAG)
CREATE EXTENSION IF NOT EXISTS vector;
CREATE TABLE event_embeddings (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  event_id UUID REFERENCES events(id) ON DELETE CASCADE,
  embedding VECTOR(1536), -- OpenAI ada-002 embedding size
  summary_text TEXT, -- Human-readable summary of event
  created_at TIMESTAMP DEFAULT NOW()
);
CREATE INDEX ON event_embeddings USING ivfflat (embedding vector_cosine_ops);
```

Design Justification: UUID primary keys for distributed safety. visitor_hash provides session grouping without PII. JSONB metadata allows flexible event properties without schema migrations. pgvector IVFFlat index optimized for cosine similarity searches (AI queries).

3.3 API Surface

EndpointMethodAuthRequestResponsePurpose/api/auth/[...nextauth]*PublicNextAuth flowSession tokenOAuth/Passkey authentication/api/trackPOSTPublic{page, type, metadata}{success: true}Log analytics event (rate-limited)/api/eventsGETAdmin?startDate&endDate&metric{totalViews, pageStats[]}Fetch aggregated analytics/api/chatPOSTAdmin{question: string}{answer, contextUsed[]}AI insights query (RAG)/api/consentPOSTPublic{action: "grant"|"revoke"}{success: true}Record GDPR consent/api/users/[id]/deleteDELETEAdminUser ID param{deleted: true}GDPR data erasure/api/healthGETPublic-{status: "ok", db: true}Health check for monitoring

OpenAPI Specification: Auto-generated via next-swagger-doc or manual schema in /public/swagger.json. Served at /api-docs using swagger-ui-react.

3.4 Infrastructure Plan

ComponentTechnologySpecificationJustificationComputeVercel Edge FunctionsServerless, EU region (Amsterdam)Auto-scales, zero cold starts for Next.js, GDPR-compliantDatabaseNeon Serverless PostgreSQL1GB storage, 100ms latency, EU CentralServerless pricing, pgvector support, auto-suspendCacheUpstash Redis1GB, EU region, REST APIDurable Redis, serverless billing, low latencyAI ServiceOpenAI APIGPT-4 Turbo, ada-002 embeddingsIndustry-standard, reliable, EU data processing optionAuthNextAuth.jsSelf-hosted on VercelNo third-party lock-in, supports Passkeys + OAuthCDNVercel Edge NetworkGlobal with EU originAutomatic, bundled with VercelMonitoringOpenTelemetry + Grafana CloudFree tier (500GB traces/month)Standard observability stack, Prometheus metricsCI/CDGitHub Actions2000 free minutes/monthNative Git integration, matrix buildsSecretsVercel Environment VariablesEncrypted at restBuilt-in, no external vault needed for MVP

MCP/Server Provisioning: For local development, Docker Compose provides PostgreSQL + Redis containers. Production uses managed services (no custom servers). If custom MCP required, provision Ubuntu 22.04 VM with 2vCPU/4GB RAM, install Docker, deploy via docker-compose.prod.yml.

Alternative Infrastructure: AWS (RDS Aurora Serverless + ElastiCache + Lambda). Tradeoff: More control but higher operational overhead, complex VPC setup, and costlier for low traffic.

4. Implementation Roadmap

Phase 1: MVP - Core Analytics

Goal: Functional analytics tracking, basic dashboard, authentication

Milestone Criteria:

- User can sign in with OAuth
- Events logged to database with consent flow
- Dashboard displays pageview charts (daily/weekly)
- 80% test coverage on core flows
- Deployed to staging environment

Estimated Effort: 80 story points (160 person-hours)

Phase 2: AI Integration - Insights Chatbot

Goal: AI-powered query interface with RAG

Milestone Criteria:

- pgvector embeddings generated for events
- Chatbot UI functional with 5+ test queries
- AI responses cite specific events
- Prompt safety filter implemented
- 90% test coverage including AI mocks

Estimated Effort: 40 story points (80 person-hours)

Phase 3: Production Hardening (Weeks 7-8) - Compliance & Ops

Goal: GDPR compliance, accessibility, observability

Milestone Criteria:

- WCAG 2.2 AA audit passed (jest-axe + manual)
- Data deletion workflow tested
- Internationalization (EN + NB) complete
- OpenTelemetry metrics dashboard live
- Load testing confirms 1000 req/min capacity
- Core Web Vitals meet targets (LCP < 2.5s)

Estimated Effort: 40 story points (80 person-hours)

Sprint Breakdown (First 3 Sprints)

Sprint 1 : Foundation

- Setup Next.js 15 project with TypeScript, Tailwind, ESLint
- Configure Prisma with PostgreSQL schema
- Implement NextAuth with OAuth (Google/GitHub)
- Create basic layout and navigation
- Setup GitHub Actions CI (lint, type-check)
- Write project README and setup.sh script
- **Deliverables**: Running Next.js app with auth, DB migrations

Sprint 2 : Analytics Core

- Implement POST /api/track with rate limiting
- Create Events table and seed data
- Build dashboard page with Recharts (line/bar charts)
- Add date range filter UI
- Implement GDPR consent banner
- Write E2E tests for tracking flow (Playwright)
- Setup Vercel deployment pipeline
- **Deliverables**: Functional analytics tracking + dashboard

Sprint 3 (Week 5-6): AI Chatbot

- Integrate pgvector extension
- Write embedding generation script (cron or on-demand)
- Create chat UI component
- Implement POST /api/chat with RAG logic
- Add AI response streaming (optional)
- Mock OpenAI in tests (100% coverage)
- Add error boundaries and fallbacks
- **Deliverables**: Working AI insights chatbot

5 CI/CD Pipeline

```
# .github/workflows/ci.yml
name: CI/CD Pipeline
on: [push, pull_request]

jobs:
  lint-and-type-check:
    runs-on: ubuntu-latest
    steps:
      - Checkout code
      - Setup Node 18
      - Install dependencies (npm ci)
      - Run ESLint (npm run lint)
      - Run TypeScript check (npm run type-check)

  unit-tests:
    runs-on: ubuntu-latest
    steps:
      - Checkout code
      - Setup Node 18
      - Start Postgres/Redis (docker-compose)
      - Run Vitest (npm run test:unit)
      - Upload coverage to Codecov

  e2e-tests:
    runs-on: ubuntu-latest
    steps:
      - Checkout code
```

```
    - Setup Node 18
    - Start services (docker-compose)
    - Run DB migrations
    - Run Playwright (npm run test:e2e)
    - Upload test artifacts (screenshots/videos)

security-scan:
  runs-on: ubuntu-latest
  steps:
    - Checkout code
    - Run npm audit --production
    - Run Snyk test (if token available)

build:
  runs-on: ubuntu-latest
  needs: [lint-and-type-check, unit-tests, e2e-tests]
  steps:
    - Checkout code
    - Setup Node 18
    - Build Next.js (npm run build)
    - Upload build artifacts

deploy-staging:
  runs-on: ubuntu-latest
  needs: build
  if: github.ref == 'refs/heads/main'
  steps:
    - Deploy to Vercel (staging)
```

- Run smoke tests against staging
- Post Slack notification

deploy-production:
runs-on: ubuntu-latest
needs: deploy-staging
if: github.ref == 'refs/heads/main' && manual approval
steps:
- Deploy to Vercel (production)
- Run Lighthouse CI
- Post Slack notification

6. Security & Compliance

6.1 Authentication & Authorization

Authentication Flow:

1. User visits /login → Redirected to NextAuth
2. Option A (Passkey): WebAuthn ceremony → Credential verified → Session created
3. Option B (OAuth): Redirect to Google/GitHub → Callback with code → Exchange for token → Session created
4. Session stored in JWT (encrypted, httpOnly cookie)
5. Refresh token rotation on every request (NextAuth default)

Authorization (RBAC):

```
// Permission matrix
const permissions = {
  admin: ['read:events', 'write:events', 'delete:data', 'manage:users', 'use:ai'],
  viewer: ['read:events', 'use:ai']
};
```

```
// Middleware enforcement
export async function middleware(req: NextRequest) {
  const token = await getToken({ req });
  if (!token) return NextResponse.redirect('/login');

  const requiredPermission = routePermissions[req.nextUrl.pathname];
  const userPermissions = permissions[token.role];

  if (!userPermissions.includes(requiredPermission)) {
    return NextResponse.json({ error: 'Forbidden' }, { status: 403 });
  }

  return NextResponse.next();
}
```

6.3 GDPR Compliance Checklist

- **Lawful Basis**: Consent (Art. 6(1)(a)) - Cookie banner before tracking
- **Data Minimization**: Only track events needed, no names/emails in events
- **Purpose Limitation**: Data used only for analytics, stated in privacy policy
- **Storage Limitation**: 1-year retention with automated deletion
- **Right to Access**: Admin can export user's events as JSON
- **Right to Erasure**: Delete button triggers cascading event deletion
- **Data Portability**: Export events in CSV format (stretch goal)
- **Privacy by Design**: Anonymization by default, consent before tracking
- **Data Processing Agreement**: Use EU-hosted services (Neon EU, Vercel Amsterdam)
- **DPIA**: Documented in /docs/DPIA.md (low-risk due to no PII)
- **Breach Notification**: Alert mechanism for security incidents (72-hour window)

Privacy Policy Requirements: Must disclose: (1) What data collected (events, hashes), (2) Retention period (1 year), (3) Third parties (OpenAI for AI queries), (4) User rights (access, delete), (5) Cookie usage (analytics cookie after consent).

10. Acceptance Criteria & KPIs

Phase 1 (MVP) Acceptance Criteria

- **Admin can sign in with Google OAuth in < 5 seconds**
- **Cookie banner displays on first visit, tracks only after consent**
- **100 test events logged successfully without errors**
- **Dashboard renders 3 chart types (line, bar, pie) with correct data**
- **Date range filter updates charts within 2 seconds**
- **Mobile responsive (viewport 320px-1920px)**
- **Lighthouse score ≥ 85 (all categories)**
- **Zero WCAG 2.2 AA violations detected by jest-axe**
- **Test coverage ≥ 80% (unit + integration)**
- **Deployed to staging.domain.com with SSL**

Phase 2 (AI) Acceptance Criteria

- **AI chatbot responds to "What page had most views today?" in < 5 seconds**
- **AI response cites specific event IDs or dates**
- **5/5 test queries return relevant answers (manual evaluation)**
- **pgvector embeddings generated for 100+ events**
- **AI errors (OpenAI 500) handled gracefully with fallback message**
- **AI query logs stored for analytics (excluding sensitive content)**

Phase 3 (Production) Acceptance Criteria

- **Norwegian Bokmål translation covers 100% of UI strings**
- **Data deletion workflow erases all events within 30 seconds**
- **Rate limiting blocks requests after 100/min, returns 429**
- **OpenTelemetry metrics visible in Grafana dashboard**
- **Load test confirms 1000 req/min sustained for 10 minutes**
- **Core Web Vitals: LCP 1.8s, FID 50ms, CLS 0.05 (actual measurements)**

- **Privacy policy and terms of service published at /privacy, /terms**

11. Actionable Step-by-Step Process + To-Do List

Step-by-Step Implementation Process

1. **Environment Setup** : Clone template, install dependencies, configure pre-commit hooks
2. **Database Design** : Create Prisma schema, generate migrations, seed test data
3. **Authentication** : Integrate NextAuth, configure OAuth providers, test login flow
4. **API Foundation** : Build /api/track endpoint with validation, rate limiting, and tests
5. **Dashboard UI** : Create chart components, implement data fetching, add filtering
6. **GDPR Compliance** : Cookie banner, consent logic, data deletion endpoint
7. **AI Integration** : Setup pgvector, embedding generation, RAG chatbot
8. **Internationalization** : Add Norwegian translations, test locale routing
9. **CI/CD Pipeline** : Configure GitHub Actions, Vercel deployment, monitoring


```
[
 {
  "id": "task-001",
  "title": "Initialize Next.js 15 Project",
  "description": "Create Next.js 15 app with TypeScript, Tailwind CSS, and App Router",
  "owner": "full_stack_developer",
  "dependencies": [],
  "estimate_hours": 4,
  "priority": "high",
  "status": "todo",
  "acceptance_criteria": "Project runs on localhost:3000, ESLint configured, no build errors",
  "created_at": "2025-10-25",
  "due_date": "2025-10-26"
 },
```

```
  {
    "id": "task-002",
    "title": "Setup Prisma with PostgreSQL",
    "description": "Configure Prisma ORM, define schema for Users/Events/Pages tables, create initial migration",
    "owner": "full_stack_developer",
    "dependencies": ["task-001"],
    "estimate_hours": 6,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Prisma migrate dev succeeds, can query database from app",
    "created_at": "2025-10-25",
    "due_date": "2025-10-27"
  },
  {
    "id": "task-003",
    "title": "Configure Docker Compose for Local Development",
    "description": "Create docker-compose.yml with PostgreSQL 15 and Redis 7 services",
    "owner": "devops_engineer",
    "dependencies": [],
    "estimate_hours": 3,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "docker-compose up runs both services, app connects successfully",
    "created_at": "2025-10-25",
    "due_date": "2025-10-27"
  },
  {
```

```
    "id": "task-004",
    "title": "Integrate NextAuth with OAuth",
    "description": "Setup NextAuth.js with Google and GitHub OAuth providers, configure session strategy",
    "owner": "full_stack_developer",
    "dependencies": ["task-002"],
    "estimate_hours": 8,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "User can sign in with Google/GitHub, session persists across page refreshes",
    "created_at": "2025-10-25",
    "due_date": "2025-10-29"
  },
  {
    "id": "task-005",
    "title": "Implement Passkey Authentication",
    "description": "Add FIDO2 WebAuthn support using @simplewebauthn library",
    "owner": "full_stack_developer",
    "dependencies": ["task-004"],
    "estimate_hours": 10,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "User can register and login with passkey on Chrome/Safari",
    "created_at": "2025-10-25",
    "due_date": "2025-11-01"
  },
  {
    "id": "task-006",
```

```
    "title": "Create POST /api/track Endpoint",
    "description": "Build event tracking API with Zod validation, rate limiting, and anonymized visitor hash",
    "owner": "full_stack_developer",
    "dependencies": ["task-002"],
    "estimate_hours": 6,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Endpoint accepts events, stores in DB, rate limits at 100/min, returns 201",
    "created_at": "2025-10-25",
    "due_date": "2025-10-30"
  },
  {
    "id": "task-007",
    "title": "Implement Rate Limiting with Upstash",
    "description": "Integrate @upstash/ratelimit library for Redis-based rate limiting on /api/track",
    "owner": "full_stack_developer",
    "dependencies": ["task-006"],
    "estimate_hours": 4,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Requests exceed limit return 429, Redis stores sliding window counters",
    "created_at": "2025-10-25",
    "due_date": "2025-10-31"
  },
  {
```

```
    "id": "task-008",
    "title": "Build GDPR Cookie Consent Banner",
    "description": "Create React component with consent/reject buttons, store choice in cookie",
    "owner": "full_stack_developer",
    "dependencies": ["task-001"],
    "estimate_hours": 5,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Banner shows on first visit, hides after choice, blocks tracking until consent",
    "created_at": "2025-10-25",
    "due_date": "2025-11-02"
  },
  {
    "id": "task-009",
    "title": "Create Analytics Dashboard Page",
    "description": "Build /dashboard route with Recharts line/bar charts showing pageviews over time",
    "owner": "full_stack_developer",
    "dependencies": ["task-006"],
    "estimate_hours": 12,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Dashboard renders 3 charts with real data, responsive on mobile",
    "created_at": "2025-10-25",
    "due_date": "2025-11-05"
  },
  {
```

```
    "id": "task-010",
    "title": "Implement Date Range Filter",
    "description": "Add date picker component to dashboard, filter events by startDate/endDate",
    "owner": "full_stack_developer",
    "dependencies": ["task-009"],
    "estimate_hours": 6,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "Selecting date range updates charts within 2 seconds, URL params persist filter",
    "created_at": "2025-10-25",
    "due_date": "2025-11-06"
  },
  {
    "id": "task-011",
    "title": "Setup pgvector Extension",
    "description": "Enable pgvector in Neon, create EventEmbeddings table with vector column and index",
    "owner": "full_stack_developer",
    "dependencies": ["task-002"],
    "estimate_hours": 4,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "pgvector extension loaded, IVFFlat index created, can insert/query vectors",
    "created_at": "2025-10-25",
    "due_date": "2025-11-08"
  },
```

```
  {
    "id": "task-012",
    "title": "Generate Event Embeddings",
    "description": "Write script to summarize events and generate OpenAI ada-002 embeddings",
    "owner": "full_stack_developer",
    "dependencies": ["task-011"],
    "estimate_hours": 8,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Script processes 100 events, stores embeddings, completes in < 2 minutes",
    "created_at": "2025-10-25",
    "due_date": "2025-11-10"
  },
  {
    "id": "task-013",
    "title": "Build AI Chatbot UI",
    "description": "Create /chat page with input field, message history, streaming response support",
    "owner": "full_stack_developer",
    "dependencies": ["task-001"],
    "estimate_hours": 10,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "User can type question, see loading state, receive AI response in chat bubble",
    "created_at": "2025-10-25",
    "due_date": "2025-11-12"
```

```
  },
  {
    "id": "task-014",
    "title": "Implement RAG Logic in POST /api/chat",
    "description": "Build endpoint that queries pgvector, retrieves similar events, sends context to GPT-4",
    "owner": "full_stack_developer",
    "dependencies": ["task-012", "task-013"],
    "estimate_hours": 12,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Chatbot answers 5 test queries with relevant responses, cites event IDs",
    "created_at": "2025-10-25",
    "due_date": "2025-11-14"
  },
  {
    "id": "task-015",
    "title": "Create Data Deletion Endpoint",
    "description": "Build POST /api/users/[id]/delete with cascading event deletion and audit logging",
    "owner": "full_stack_developer",
    "dependencies": ["task-002"],
    "estimate_hours": 6,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Endpoint deletes all user events, logs action, returns 200 within 5 seconds",
    "created_at": "2025-10-25",
```

```
    "due_date": "2025-11-16"
  },
  {
    "id": "task-016",
    "title": "Write Unit Tests for API Routes",
    "description": "Use Vitest to test /api/track, /api/events, /api/chat with mocked dependencies",
    "owner": "qa_engineer",
    "dependencies": ["task-006", "task-014"],
    "estimate_hours": 16,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "90% coverage on API routes, all tests pass, OpenAI mocked",
    "created_at": "2025-10-25",
    "due_date": "2025-11-18"
  },
  {
    "id": "task-017",
    "title": "Write E2E Tests with Playwright",
    "description": "Create tests for login, tracking, dashboard, AI chat, data deletion flows",
    "owner": "qa_engineer",
    "dependencies": ["task-009", "task-013", "task-015"],
    "estimate_hours": 20,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "5 E2E tests cover critical paths, all pass in CI, screenshots saved on failure",
    "created_at": "2025-10-25",
    "due_date": "2025-11-20"
```

```
  },
  {
    "id": "task-018",
    "title": "Implement Internationalization (i18n)",
    "description": "Setup next-intl, create EN and NB locale files, configure routing",
    "owner": "full_stack_developer",
    "dependencies": ["task-009"],
    "estimate_hours": 10,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "User can switch language via /nb/dashboard, all UI strings translated",
    "created_at": "2025-10-25",
    "due_date": "2025-11-22"
  },
  {
    "id": "task-019",
    "title": "Run Accessibility Audit with jest-axe",
    "description": "Integrate jest-axe in tests, fix all WCAG 2.2 AA violations",
    "owner": "qa_engineer",
    "dependencies": ["task-009", "task-013"],
    "estimate_hours": 12,
    "priority": "high",
"status": "todo",
    "acceptance_criteria": "Zero axe violations detected, all interactive elements keyboard accessible",
    "created_at": "2025-10-25",
    "due_date": "2025-11-24"
  },
```

```
  {
    "id": "task-020",
    "title": "Configure GitHub Actions CI Pipeline",
    "description": "Create workflow with lint, type-check, unit tests, E2E tests, and security scan jobs",
    "owner": "devops_engineer",
    "dependencies": ["task-016", "task-017"],
    "estimate_hours": 8,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Pipeline runs on push/PR, all stages pass, coverage uploaded to Codecov",
    "created_at": "2025-10-25",
    "due_date": "2025-11-25"
  },
  {
    "id": "task-021",
    "title": "Setup Vercel Deployment",
    "description": "Configure Vercel project with EU region, environment variables, preview/production environments",
    "owner": "devops_engineer",
    "dependencies": ["task-020"],
    "estimate_hours": 4,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "App deploys to staging.domain.com, production requires manual approval",
    "created_at": "2025-10-25",
    "due_date": "2025-11-26"
```

```
  },
  {
    "id": "task-022",
    "title": "Integrate OpenTelemetry Monitoring",
    "description": "Setup OpenTelemetry SDK, configure metrics/traces export to Grafana Cloud",
    "owner": "devops_engineer",
    "dependencies": ["task-021"],
    "estimate_hours": 10,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "Metrics visible in Grafana dashboard, traces show API request flows",
    "created_at": "2025-10-25",
    "due_date": "2025-11-28"
  },
  {
    "id": "task-023",
    "title": "Create Structured Logging System",
    "description": "Implement Winston logger with JSON format, log levels, and privacy filters",
    "owner": "full_stack_developer",
    "dependencies": ["task-006"],
    "estimate_hours": 6,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "All API routes log structured JSON, no PII in logs, searchable in Grafana",
    "created_at": "2025-10-25",
```

```
    "due_date": "2025-11-29"
  },
  {
    "id": "task-024",
    "title": "Setup Alert Rules",
    "description": "Configure Grafana alerts for error rates, latency, database failures with Slack integration",
    "owner": "devops_engineer",
    "dependencies": ["task-022"],
    "estimate_hours": 6,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "4 alert rules active, test alert delivered to Slack within 2 minutes",
    "created_at": "2025-10-25",
    "due_date": "2025-11-30"
  },
  {
    "id": "task-025",
    "title": "Run Lighthouse CI Performance Tests",
    "description": "Integrate Lighthouse CI in GitHub Actions, set performance budget thresholds",
    "owner": "qa_engineer",
    "dependencies": ["task-020"],
    "estimate_hours": 4,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Lighthouse runs on every PR, fails if score < 85, results posted as comment",
    "created_at": "2025-10-25",
```

```
    "due_date": "2025-12-01"
  },
  {
    "id": "task-026",
    "title": "Conduct Load Testing with k6",
    "description": "Write k6 scripts for 1000 req/min load test on /api/track and /api/events",
    "owner": "qa_engineer",
    "dependencies": ["task-021"],
    "estimate_hours": 8,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "System sustains 1000 req/min for 10 minutes, error rate < 1%, p95 latency < 500ms",
    "created_at": "2025-10-25",
    "due_date": "2025-12-03"
  },
  {
    "id": "task-027",
    "title": "Write Comprehensive README.md",
    "description": "Create README with badges, demo link, architecture diagram, setup instructions, API docs",
    "owner": "full_stack_developer",
    "dependencies": ["task-021"],
    "estimate_hours": 6,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "README includes 10+ sections, setup.sh works on fresh clone, screenshots added",
    "created_at": "2025-10-25",
```

```
    "due_date": "2025-12-04"
  },
  {
    "id": "task-028",
    "title": "Generate OpenAPI Documentation",
    "description": "Create OpenAPI 3.0 spec for all API endpoints, integrate Swagger UI at /api-docs",
    "owner": "full_stack_developer",
    "dependencies": ["task-006", "task-014", "task-015"],
    "estimate_hours": 8,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "Swagger UI displays all endpoints with request/response schemas, try-it-out works",
    "created_at": "2025-10-25",
    "due_date": "2025-12-05"
  },
  {
    "id": "task-029",
    "title": "Create setup.sh and demo.sh Scripts",
    "description": "Write bash scripts for one-click local setup and demo data seeding",
    "owner": "devops_engineer",
    "dependencies": ["task-003"],
    "estimate_hours": 4,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "setup.sh installs deps, starts DB, runs migrations; demo.sh seeds 1000 test events",
    "created_at": "2025-10-25",
```

```
    "due_date": "2025-12-06"
  },
  {
    "id": "task-030",
    "title": "Write Privacy Policy and Terms of Service",
    "description": "Draft legal documents covering data collection, retention, user rights, cookie usage",
    "owner": "product_owner",
    "dependencies": [],
    "estimate_hours": 8,
    "priority": "high",
    "status": "blocked",
    "acceptance_criteria": "Approved by legal/compliance, published at /privacy and /terms routes",
    "created_at": "2025-10-25",
    "due_date": "2025-12-07"
  },
  {
    "id": "task-031",
    "title": "Conduct Security Audit",
    "description": "Run OWASP ZAP scan, npm audit, Snyk scan; fix high/critical vulnerabilities",
    "owner": "security_consultant",
    "dependencies": ["task-021"],
    "estimate_hours": 12,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "Zero high/critical vulnerabilities, audit report documented, fixes merged",
```

```
    "created_at": "2025-10-25",
    "due_date": "2025-12-08"
  },
  {
    "id": "task-032",
    "title": "Build AI Evaluation Harness",
    "description": "Create test dataset with 20 queries and expected answers, automated accuracy scoring",
    "owner": "full_stack_developer",
    "dependencies": ["task-014"],
    "estimate_hours": 10,
    "priority": "medium",
    "status": "todo",
    "acceptance_criteria": "Harness runs in CI, reports accuracy score, blocks merge if < 80% accuracy",
    "created_at": "2025-10-25",
    "due_date": "2025-12-09"
  },
  {
    "id": "task-033",
    "title": "Implement Redis Caching for Aggregated Stats",
    "description": "Cache GET /api/events results with 1-hour TTL, implement cache invalidation on new events",
    "owner": "full_stack_developer",
    "dependencies": ["task-009"],
    "estimate_hours": 6,
    "priority": "medium",
    "status": "todo",
```

"acceptance_criteria": "Cache hit rate > 80% for dashboard queries, response time reduced by 50%",

"created_at": "2025-10-25",

"due_date": "2025-12-10"

},

{

"id": "task-034",

"title": "Create Demo Video for Portfolio",

"description": "Record 3-minute walkthrough: login, tracking, dashboard, AI chat, data deletion",

"owner": "full_stack_developer",

"dependencies": ["task-029"],

"estimate_hours": 4,

"priority": "medium",

"status": "todo",

"acceptance_criteria": "Video uploaded to YouTube, embedded in README, shows all key features",

"created_at": "2025-10-25",

"due_date": "2025-12-11"

},

{

"id": "task-035",

"title": "Write Operational Runbook",

"description": "Document monitoring, alerting, backup/recovery, incident response procedures",

"owner": "devops_engineer",

"dependencies": ["task-022", "task-024"],

"estimate_hours": 6,

"priority": "medium",

```
    "status": "todo",
    "acceptance_criteria": "Runbook covers 5+ scenarios, stored in /docs/RUNBOOK.md, reviewed by team",
    "created_at": "2025-10-25",
    "due_date": "2025-12-12"
  },
  {
    "id": "task-036",
    "title": "Prepare Resume Bullet Points",
    "description": "Write STAR-format bullet points with metrics for resume and LinkedIn",
    "owner": "full_stack_developer",
    "dependencies": ["task-025", "task-026"],
    "estimate_hours": 3,
    "priority": "low",
    "status": "todo",
    "acceptance_criteria": "5 bullet points written, include metrics (test coverage, performance scores, etc.)",
    "created_at": "2025-10-25",
    "due_date": "2025-12-13"
  },
  {
    "id": "task-037",
    "title": "Final Production Deployment",
    "description": "Deploy to production environment, verify all features, announce launch",
    "owner": "devops_engineer",
    "dependencies": ["task-031", "task-025", "task-026"],
    "estimate_hours": 4,
    "priority": "high",
    "status": "todo",
```

```
    "acceptance_criteria": "App live at production URL, all health checks pass, monitoring active",
    "created_at": "2025-10-25",
    "due_date": "2025-12-14"
  },
  {
    "id": "task-038",
    "title": "Post-Launch Monitoring and Bug Fixes",
    "description": "Monitor production for 1 week, address any issues, optimize based on real usage",
    "owner": "full_stack_developer",
    "dependencies": ["task-037"],
    "estimate_hours": 16,
    "priority": "high",
    "status": "todo",
    "acceptance_criteria": "No critical bugs reported, uptime > 99%, performance targets met",
    "created_at": "2025-10-25",
    "due_date": "2025-12-21"
  }
]
```