KẾ HOẠCH TRIỂN KHAI DỰ ÁN

Nền tảng Blog & Video Cá nhân

Thế hệ Mới - 12 Tuần

Phiên bản:	1.0
Ngày tạo:	27 Tháng 11, 2025
Tác giả:	Claude & Development Team
Loại tài liệu:	Technical Planning Document

MỤC LỤC

1. TỔNG QUAN DỰ ÁN

1.1 Mô tả Dự án

Phát triển một nền tảng Blog & Video cá nhân thế hệ mới với các đặc điểm:

- **Giao diện:** Pastel theme - nhẹ nhàng, hiện đại, thân thiện
- **Nền tảng:** Web (Next.js) + Mobile (React Native/Expo)
- **Tính năng cốt lõi:**
  - Tạo và quản lý bài viết blog với Rich Text Editor
  - Upload và streaming video chất lượng cao
  - Tìm kiếm thông minh với AI (Semantic Search + RAG)
  - Hệ thống bình luận và tương tác
  - Analytics và thống kê chi tiết

1.2 Mục tiêu Kỹ thuật

Tiêu chí	Mục tiêu
Hiệu năng	API response < 500ms (p95)
Khả năng mở rộng	Hỗ trợ 1,000 DAU ban đầu
Code Quality	Test coverage ≥ 80%
Bảo mật	Zero critical vulnerabilities
Deployment	Docker-based, CI/CD tự động

2. PHÂN TÍCH NGHIỆP VỤ CHI TIẾT

2.1 Actors (Người dùng hệ thống)

Guest (Khách)

- ✅ Xem blog công khai
- ✅ Xem video công khai
- ✅ Tìm kiếm nội dung
- ❌ Không thể bình luận, like, bookmark
- ❌ Không thể tạo nội dung

Registered User (Người dùng đã đăng ký)

- ✅ Tất cả quyền của Guest
- ✅ Tạo, chỉnh sửa, xóa bài viết của mình
- ✅ Upload video
- ✅ Bình luận trên bài viết
- ✅ Like, bookmark bài viết
- ✅ Xem thống kê cá nhân

Author (Tác giả)

- ✅ Tất cả quyền của Registered User
- ✅ Quản lý bài viết của mình (Draft, Publish, Archive)
- ✅ Xem chi tiết analytics của bài viết
- ✅ Kiểm duyệt comment trên bài viết mình
- ✅ Xóa/ẩn comment spam

Admin (Quản trị viên)

- ✅ Toàn quyền với mọi nội dung
- ✅ Quản lý users (block, delete)
- ✅ Xóa nội dung vi phạm
- ✅ Xem dashboard analytics tổng thể
- ✅ Quản lý categories, tags
- ✅ Cấu hình hệ thống

2.2 Use Cases (Tính năng chính)

UC-01: Quản lý Tài khoản

- **UC-01.1:** Đăng ký tài khoản (Email + Password)
- **UC-01.2:** Đăng nhập
- **UC-01.3:** Quên mật khẩu / Đặt lại mật khẩu
- **UC-01.4:** Cập nhật profile (Avatar, Bio, Social Links)
- **UC-01.5:** Đổi mật khẩu
- **UC-01.6:** Xóa tài khoản (Soft delete)

UC-02: Quản lý Bài viết

- **UC-02.1:** Tạo bài viết mới
  - Input: Tiêu đề, Nội dung (Rich Text), Danh mục, Tags, Video
  - Business Rules: Tiêu đề 10-200 ký tự, Nội dung min 50 ký tự, Max 5 tags
- **UC-02.2:** Chỉnh sửa bài viết (chỉ tác giả)
- **UC-02.3:** Xóa bài viết (Soft delete, hard delete sau 30 ngày)
- **UC-02.4:** Xuất bản/Hủy xuất bản
- **UC-02.5:** Xem danh sách bài viết (Feed với filter & sort)

UC-03: Quản lý Video

- **UC-03.1:** Upload video (MP4, MOV, AVI - max 2GB)
  - Auto-transcode: 1080p, 720p, 480p
  - Status: Uploading → Processing → Ready → Failed
- **UC-03.2:** Xem video (Adaptive streaming HLS)
- **UC-03.3:** Xóa video

UC-04: Tương tác (Engagement)

- **UC-04.1:** Bình luận (max 500 ký tự, 1 level reply)
- **UC-04.2:** Thích bài viết (Like/Unlike toggle)
- **UC-04.3:** Lưu bài viết (Bookmark với folders)

UC-05: Tìm kiếm & Khám phá

- **UC-05.1:** Tìm kiếm từ khóa (Full-text search)
- **UC-05.2:** Tìm kiếm ngữ nghĩa (Vector embeddings)
- **UC-05.3:** RAG - Hỏi đáp thông minh với AI

UC-06: Thống kê & Analytics

- **UC-06.1:** Dashboard tác giả (stats, charts)
- **UC-06.2:** View tracking (unique per session per day)

2.3 Business Rules (Quy tắc nghiệp vụ)

BR-01: Xác thực & Phân quyền

- Chỉ user đã xác thực email mới được tạo bài viết
- User chỉ được sửa/xóa bài viết của chính mình
- Admin có toàn quyền với mọi nội dung

BR-02: Giới hạn Upload

- Mỗi user: Tối đa 100 bài viết
- Mỗi bài viết: Tối đa 1 video
- Video: Tối đa 30 phút, 2GB
- Ảnh thumbnail: Tối đa 5MB

BR-03: Nội dung

- Tiêu đề bài viết phải unique trong phạm vi user
- Không cho phép nội dung spam (kiểm tra duplicate)
- Comment có từ ngữ nhạy cảm sẽ được đánh dấu review

BR-04: Video Processing

- Video chỉ hiển thị khi trạng thái = 'ready'
- Nếu processing fail > 3 lần → status 'failed' vĩnh viễn
- Tự động xóa raw video sau encode thành công

BR-05: Search & AI

- Embedding auto-generate khi bài viết publish
- Re-generate embedding khi content thay đổi > 20%
- Cache kết quả search trong 5 phút

3. THIẾT KẾ DATABASE SCHEMA

3.1 Entity Relationship Diagram (ERD)

Hệ thống database được thiết kế với các quan hệ chính sau:

- USERS (1) ————< (N) POSTS
- POSTS (1) ————< (N) COMMENTS
- POSTS (1) ————< (N) LIKES
- POSTS (1) ————< (N) BOOKMARKS
- POSTS (1) ———— (0..1) VIDEOS
- POSTS (N) ———— (N) CATEGORIES (via post_categories)
- POSTS (N) ———— (N) TAGS (via post_tags)
- VIDEOS (1) ————< (N) VIDEO_VIEWS

3.2 Các Bảng Chính (Core Tables)

Bảng	Mục đích	Số trường
users	Lưu thông tin người dùng, authentication	15+ fields
posts	Bài viết chính với content, metadata, AI embeddings	20+ fields
videos	Quản lý video files, encoding status, analytics	25+ fields
comments	Bình luận trên bài viết, hỗ trợ 1-level reply	12+ fields
categories	Danh mục bài viết (Technology, Travel, Food...)	7 fields
tags	Tags cho bài viết (nhiều-nhiều)	5 fields
sessions	Quản lý phiên đăng nhập	9 fields
likes	Lượt thích bài viết	4 fields
bookmarks	Bài viết đã lưu với folders	6 fields
post_views	Analytics lượt xem bài viết	10 fields
video_views	Analytics lượt xem video	12 fields
activity_logs	Audit trail cho hành động quan trọng	8 fields

Bảng	Mục đích	Số trường
search_queries	Lưu queries để phân tích	6 fields

3.3 Data Dictionary - Bảng USERS

Field	Type	Constraints	Mô tả
id	UUID	PK	Unique identifier
email	VARCHAR(255)	UNIQUE, NOT NULL	Email đăng nhập
username	VARCHAR(50)	UNIQUE, NOT NULL	Username (a-z0-9_-)
password_hash	VARCHAR(255)	NOT NULL	Bcrypt hash
full_name	VARCHAR(100)	NULL	Tên đầy đủ
bio	TEXT	NULL	Tiểu sử ngắn
avatar_url	VARCHAR(500)	NULL	Link ảnh đại diện
email_verified	BOOLEAN	DEFAULT FALSE	Đã verify email?
is_active	BOOLEAN	DEFAULT TRUE	Account hoạt động?
is_admin	BOOLEAN	DEFAULT FALSE	Có phải admin?
created_at	TIMESTAMP	DEFAULT NOW()	Ngày tạo
updated_at	TIMESTAMP	DEFAULT NOW()	Ngày cập nhật
deleted_at	TIMESTAMP	NULL	Soft delete timestamp

3.4 Data Dictionary - Bảng POSTS

Field	Type	Constraints	Mô tả
id	UUID	PK	Post ID
author_id	UUID	FK → users	Tác giả
title	VARCHAR(200)	NOT NULL	Tiêu đề (10-200 chars)
slug	VARCHAR(250)	NOT NULL	URL slug
content	TEXT	NOT NULL	Nội dung HTML
excerpt	TEXT	NULL	Tóm tắt ngắn
featured_image_url	VARCHAR(500)	NULL	Ảnh đại diện
video_id	UUID	FK → videos	Video đính kèm
status	VARCHAR(20)	DEFAULT 'draft'	draft/published/archived
visibility	VARCHAR(20)	DEFAULT 'public'	public/private/unlisted
view_count	INTEGER	DEFAULT 0	Lượt xem
like_count	INTEGER	DEFAULT 0	Lượt thích
embedding	vector(1536)	NULL	Vector cho AI search
published_at	TIMESTAMP	NULL	Ngày xuất bản
created_at	TIMESTAMP	DEFAULT NOW()	Ngày tạo

3.5 Indexes Quan trọng

Posts Table:

- **idx_posts_author_slug** (UNIQUE) - Prevent duplicate slugs per author
- **idx_posts_embedding** (IVFFlat) - Vector similarity search
- **idx_posts_fulltext** (GIN) - Full-text search
- **idx_posts_published_at** (DESC) - Fast feed queries

Users Table:

- **idx_users_email** - Login queries

- **idx_users_username** - Profile lookups

Videos Table:

- **idx_videos_status** - Filter by encoding status
- **idx_videos_post_id** - Join with posts

> **Performance Notes:**
>
> - Partial indexes (WHERE deleted_at IS NULL) - Exclude soft-deleted records
> - Composite indexes for common query patterns
> - GIN indexes for JSONB columns (activity_logs.metadata)

3.6 Database Triggers

Trigger 1: Auto-update updated_at

Tự động cập nhật trường `updated_at` khi có UPDATE trên tables: users, posts, videos, comments

Trigger 2: Auto-increment counters

- Khi INSERT vào *likes* → Tăng *posts.like_count*
- Khi DELETE khỏi *likes* → Giảm *posts.like_count*
- Tương tự cho *comment_count*, *bookmark_count*
- Trigger cho *tags.usage_count* khi thêm/xóa post_tags

Trigger 3: Auto-generate slug

Tự động tạo URL slug từ title khi INSERT/UPDATE posts. Nếu slug trùng, append số (-1, -2, ...)

4. KIẾN TRÚC HỆ THỐNG

4.1 Clean Architecture (Backend)

Presentation Layer (Controllers)

- HTTP Request handling
- Input validation với Zod
- Response formatting
- Error handling middleware

Application Layer (Use Cases)

- Business logic workflows
- CreatePostUseCase, PublishVideoUseCase, etc.
- Định nghĩa Repository Interfaces (Ports)
- Không phụ thuộc vào infrastructure

Domain Layer (Entities)

- Core business entities (User, Post, Video)
- Business rules thuần túy
- Pure TypeScript, zero dependencies
- Dễ dàng unit test

Infrastructure Layer (Adapters)

- Repository implementations (PostgresPostRepository)
- External service integrations (MinioVideoService)
- Database connection, caching (Redis)
- Import thư viện bên thứ ba tại đây

4.2 Monorepo Structure (Nx Workspace)

apps/

- *web-client/* - Next.js 14 App Router
- *mobile-app/* - React Native + Expo
- *api-server/* - Node.js Backend (Express/Fastify)

libs/shared/

- *domain/* - Entities, Interfaces, DTOs (Zod schemas)
- *ui-kit/* - Shared React components (Button, Card, Input)
- *utils/* - Helper functions (date, string, validation)
- *data-access/* - API clients, TanStack Query hooks

libs/backend/

- *core/* - Use Cases (business logic)
- *infrastructure/* - Repository implementations

4.3 Video Processing Pipeline

Step 1: Client requests upload URL

- POST /api/videos/upload-url
- Validate file type, size
- Generate presigned MinIO URL (expire 1h)
- Return { uploadUrl, videoId }

Step 2: Direct upload to MinIO

- Client uploads directly (no server bandwidth)
- Chunked upload (resumable)
- Progress tracking

Step 3: Confirm & queue encoding

- POST /api/videos/{videoId}/confirm
- Update DB: status = 'processing'
- Queue job in BullMQ/Redis

Step 4: FFmpeg worker processes

- Download raw video from MinIO
- Extract metadata (duration, resolution, codec)
- Generate thumbnail (frame @ 2s)
- Encode to HLS: 1080p, 720p, 480p, 360p
- Upload encoded files to MinIO
- Update DB: status = 'ready'
- Delete raw file (save storage)

Step 5: Client polls status

- GET /api/videos/{videoId}/status
- When status='ready' → Show video player

4.4 AI Search & RAG Architecture

Part 1: Embedding Generation (Background Job)

1. Trigger: Post published or content updated significantly
2. Queue job: { postId, title, content }
3. Worker calls Ollama API: POST http://ollama:11434/api/embeddings
4. Receive: embedding [1536 dimensions]
5. Update DB: posts.embedding = $embedding

Part 2: Semantic Search

1. User query: 'beach vacation summer'
2. Generate query embedding (Ollama)
3. Vector similarity search (PostgreSQL pgvector)
4. Return results with similarity scores

Part 3: RAG (Retrieval Augmented Generation)

1. User question: "What did I write about my Dalat trip?"
2. Generate question embedding
3. Vector search in user's posts
4. Get top 5 relevant posts
5. Build context prompt
6. Call Ollama Llama 3 (streaming)
7. Stream response to client (SSE)

5. TECHNOLOGY STACK

Component	Technology	Version
Frontend Web	Next.js + TypeScript	14+
	Material UI (Pastel Theme)	v6
	TanStack Query	v5
	React Hook Form + Zod	Latest
Mobile	React Native + Expo	SDK 50+
	Expo Router	Latest
Backend	Node.js + TypeScript	20 LTS
	Express / Fastify	Latest
	PostgreSQL + pgvector	16
	Prisma (ORM)	Latest
Infrastructure	Docker + Docker Compose	Latest
	MinIO (S3-compatible)	Latest
	Redis (Cache + Queue)	7
	BullMQ (Job Queue)	Latest
	FFmpeg (Video Encoding)	Latest
AI/ML	Ollama (Local LLM)	Latest
	nomic-embed-text + Llama 3	Latest
Testing	Jest + Supertest	Latest
	Playwright (E2E)	Latest
DevOps	GitHub Actions (CI/CD)	-

6. LỘ TRÌNH 12 TUẦN

🎯 TUẦN 0: PHÂN TÍCH & THIẾT KẾ (Tiền kỳ)

Thời gian: 5-7 ngày (trước khi bắt đầu coding chính thức)

Ngày 1-3: Phân tích Nghiệp vụ

- ✓ Xác định Actors và Use Cases chi tiết
- ✓ Document Business Rules
- ✓ Vẽ User Flow Diagrams

Ngày 4-7: Thiết kế Database

- ✓ Vẽ ERD (Entity Relationship Diagram)
- ✓ Viết SQL schema đầy đủ (15+ tables)
- ✓ Định nghĩa Indexes, Triggers, Constraints
- ✓ Tạo Data Dictionary

📦 GIAI ĐOẠN 1: NỀN MÓNG (Tuần 1-3)

TUẦN 1: Infrastructure & Monorepo

- ✓ Docker Compose với 6+ services
- ✓ Nx Workspace setup hoàn chỉnh
- ✓ Domain entities với Zod schemas
- ✓ Repository interfaces
- ✓ Unit tests cho entities

TUẦN 2: Authentication & Database

- ✓ Supabase Auth integration
- ✓ PostgreSQL schema deployed
- ✓ Seed data loaded
- ✓ Repository implementations
- ✓ Integration tests

TUẦN 3: API Server Foundation

- ✓ 5-7 core Use Cases

- ✓ Dependency Injection setup
- ✓ RESTful API endpoints (Swagger)
- ✓ GitHub Actions CI/CD

🎨 GIAI ĐOẠN 2: TÍNH NĂNG CỐT LÕI (Tuần 4-6)

TUẦN 4: Web Frontend

- ✓ Next.js 14 setup
- ✓ MUI Pastel Theme
- ✓ 8-10 UI components với Storybook
- ✓ Feed page + Post creation form

TUẦN 5: Video Infrastructure

- ✓ MinIO + presigned URLs
- ✓ FFmpeg worker (4 qualities)
- ✓ BullMQ job queue
- ✓ Video Player component (HLS)

TUẦN 6: State Management

- ✓ TanStack Query (10+ hooks)
- ✓ Optimistic updates
- ✓ Form management
- ✓ Integration tests

📱 GIAI ĐOẠN 3: MOBILE (Tuần 7-9)

TUẦN 7: React Native Foundation

- ✓ Expo project + Expo Router
- ✓ 8-10 mobile components
- ✓ Authentication flow

TUẦN 8: Mobile Video Upload

- ✓ Video picker & recording
- ✓ expo-file-system upload

- ✓ Native video player

TUẦN 9: Advanced Features

- ✓ Image optimization
- ✓ Comments system
- ✓ User profile pages

🤖 GIAI ĐOẠN 4: AI & HOÀN THIỆN (Tuần 10-12)

TUẦN 10: AI Search

- ✓ Ollama setup
- ✓ Embedding generation
- ✓ Semantic search API
- ✓ RAG implementation

TUẦN 11: Testing & Optimization

- ✓ Query optimization
- ✓ Redis caching
- ✓ Load testing
- ✓ Security hardening

TUẦN 12: Deployment

- ✓ UI/UX polish
- ✓ Documentation
- ✓ Final testing
- ✓ Production setup

7. CHECKLIST THEO DÕI TIẾN ĐỘ

Giai đoạn	Tuần	Tasks	Status
Tiền kỳ	0	Phân tích nghiệp vụ + DB design	☐ To Do
Nền móng	1	Docker + Nx + Entities	☐ To Do
	2	Auth + Database	☐ To Do
	3	API Server + CI/CD	☐ To Do
Cốt lõi	4	Web Frontend	☐ To Do
	5	Video Infrastructure	☐ To Do
	6	State Management	☐ To Do
Mobile	7	React Native	☐ To Do
	8	Mobile Video	☐ To Do
	9	Advanced Features	☐ To Do
AI & Polish	10	AI Search & RAG	☐ To Do
	11	Testing & Optimization	☐ To Do
	12	Deployment	☐ To Do

Cách sử dụng Checklist:

- ☐ To Do - Chưa bắt đầu
- 🔄 In Progress - Đang thực hiện
- ✅ Done - Hoàn thành
- ⚠️ Blocked - Bị chặn
- 🔴 Delayed - Trễ deadline

7.1 Success Criteria

Metric	Target	Measurement
Test Coverage	≥ 80%	Jest coverage report

Metric	Target	Measurement
API Response Time	< 500ms (p95)	Load testing
Video Encoding Time	< 5 min for 10-min video	BullMQ metrics
Search Accuracy	> 70% similarity	Manual testing
Mobile App Size	< 50MB	Build output
Zero Critical Bugs	0 critical issues	Security scan
All Features Working	100% use cases	E2E tests
Documentation	100% coverage	All docs written

8. TÀI LIỆU THAM KHẢO

Technology Documentation:

- Next.js: https://nextjs.org/docs
- React Native + Expo: https://docs.expo.dev
- PostgreSQL + pgvector: https://github.com/pgvector/pgvector
- Nx Workspace: https://nx.dev/getting-started/intro
- Material UI: https://mui.com/material-ui/getting-started/
- TanStack Query: https://tanstack.com/query/latest

Architecture Patterns:

- Clean Architecture (Robert C. Martin)
- Domain-Driven Design (Eric Evans)
- Microservices Patterns (Chris Richardson)

AI/ML Resources:

- Ollama Documentation: https://github.com/ollama/ollama
- pgvector Guide: https://github.com/pgvector/pgvector
- RAG Implementation: https://www.pinecone.io/learn/retrieval-augmented-generation/

Best Practices:

- TypeScript Best Practices
- PostgreSQL Performance Tuning
- Docker Best Practices
- API Security Checklist (OWASP)

Tài liệu này được tạo tự động. Mọi thắc mắc xin liên hệ Development Team.