VelocityLLM
60-Day Development Roadmap

Production-Grade LLM Inference Engine

November 25, 2025

Contents

1 Project Overview

1.1 Project Goal

Build a production-grade, enterprise-ready LLM inference engine with comprehensive features including intelligent routing, real-time streaming, advanced caching, user management, monitoring, and scalability.

1.2 Technology Stack

Backend:

- Go (Golang) 1.21+
- PostgreSQL 15+
- Redis 7+
- WebSockets & Server-Sent Events
- Docker & Kubernetes

Frontend:

- React 18 + Next.js 14
- TailwindCSS
- shadcn/ui Components
- Zustand/Redux Toolkit
- React Query (TanStack Query)
- Recharts + Chart.js

1.3 Progress Overview

Category	Days	Percentage
Backend Development	30 days	50%
Frontend Development	20 days	33%
Full-Stack Integration	10 days	17%
Total	**60 days**	**100%**

Table 1: Project Time Distribution

2 Week 1: Backend Foundation (Days 1-5)

STATUS: COMPLETED

2.1 Day 1: Project Setup & Database Layer

- Project structure initialization
- PostgreSQL setup and connection
- GORM integration
- Basic data models (User, Request, Model)
- Database migrations

2.2 Day 2: Caching & Basic Routing

- Redis integration
- Cache layer implementation
- TTL management
- Basic routing logic
- Cache invalidation strategies

2.3 Day 3: Model Repository & API

- Model management system
- Request tracking
- RESTful API structure
- Error handling
- Response formatting

2.4 Day 4: Intelligent Router & Reliability

- Smart routing (5 strategies)
- Circuit breakers implementation
- Health monitoring system
- Failover handling
- Retry mechanisms with exponential backoff

2.5 Day 5: Concurrency, Protection & Optimization

- Worker pool (10 concurrent workers)
- Priority queue implementation
- Rate limiting (Token bucket algorithm)
- Backpressure handling
- Performance metrics (P50/P90/P95/P99)
- Connection pooling (DB, Redis, HTTP)
- Request batching (95% cost savings)

Week 1 Achievements:

- 52 API endpoints
- ∼10,000 lines of code
- 15x performance improvement
- 95% cost reduction
- Production-ready core engine

3 Week 2: Advanced Backend Features (Days 6-10)

TYPE: BACKEND

3.1 Day 6: Streaming Responses & WebSockets

- Server-Sent Events (SSE) implementation
- Token-by-token streaming
- WebSocket server setup
- Real-time connection management
- Stream cancellation support

3.2 Day 7: Advanced Caching Strategies

- Multi-level caching (L1: Memory, L2: Redis)
- Cache warming strategies
- Semantic caching
- Cache hit rate optimization
- Cache analytics dashboard

3.3 Day 8: Multi-Model Orchestration

- Model chaining
- Conditional routing
- Fallback chains enhancement
- Model composition
- Response aggregation

3.4 Day 9: Prompt Engineering Tools

- Prompt templates system
- Variable interpolation
- Prompt versioning
- A/B testing framework
- Prompt performance analytics

3.5 Day 10: Context & Token Management

- Context window optimization
- Token counting utilities
- Automatic truncation
- Context compression
- Token budget management

4 Week 3: Frontend Dashboard (Days 11-17)

TYPE: FRONTEND

4.1 Day 11: Frontend Setup & Architecture

- Next.js 14 project initialization
- TailwindCSS configuration
- Component library setup (shadcn/ui)
- Project structure organization
- Development environment

4.2 Day 12: Authentication UI

- Login page design
- Signup page design
- JWT integration
- Protected routes implementation
- Session management UI
- Password reset flow

4.3 Day 13: Main Dashboard

- Overview page layout
- Real-time metrics display
- Request statistics cards
- Cost analytics widgets
- Interactive charts (Recharts)
- Responsive design

4.4 Day 14: API Playground

- Interactive prompt interface
- Model selection dropdown
- Parameter controls (temperature, max tokens, top-p)
- Real-time response display
- Code examples (curl, Python, JavaScript, Go)
- Request/response history

4.5 Day 15: Monitoring Dashboard

- Worker pool visualization
- Queue status (real-time)
- Connection pool graphs
- Rate limiting status indicators
- System health dashboard
- Alert notifications

4.6 Day 16: Analytics Dashboard

- Latency charts (P50/P90/P95/P99)
- Throughput graphs
- Cost breakdown visualizations
- Model comparison charts
- Request history table with filtering
- Export functionality (CSV, PDF)

4.7 Day 17: Settings & Configuration

- API key management interface
- Rate limit configuration
- Model preferences UI
- Billing settings
- User profile management
- Notification preferences

5 Week 4: Enterprise Backend Features (Days 18-24)

TYPE: BACKEND

5.1 Day 18: Authentication & Authorization Backend

- JWT implementation (access + refresh tokens)
- OAuth2 integration (Google, GitHub)

- Password hashing (bcrypt)
- Token refresh mechanism
- Session management

5.2 Day 19: User Management System

- User CRUD operations
- Role management (Admin, Developer, Viewer)
- Permission system
- User groups/teams
- Activity logging

5.3 Day 20: API Key Management

- API key generation
- Key rotation mechanisms
- Usage tracking per key
- Key scopes and permissions
- Key revocation
- Rate limiting per key

5.4 Day 21: Billing & Usage Tracking

- Usage metering system
- Cost calculation engine
- Invoice generation
- Payment integration (Stripe)
- Subscription management
- Usage reports

5.5 Day 22: Quota Management

- Per-user quota system
- Rate limiting by user tier
- Usage alerts and notifications
- Overage handling
- Soft and hard limits
- Quota reset schedules

5.6 Day 23: Audit Logging

- Comprehensive request logging
- User activity tracking
- Log aggregation
- Search functionality
- Compliance features (GDPR, SOC2)
- Log retention policies

5.7 Day 24: RBAC (Role-Based Access Control)

- Role hierarchy (Admin, Developer, Viewer)
- Permission matrices
- Resource-level controls

- Dynamic permission checking
- Role assignment workflows

6 Week 5: Advanced UI Features (Days 25-31)

TYPE: FRONTEND

6.1 Day 25: Admin Dashboard

- User management interface
- System configuration panel
- Model management UI
- API key administration
- System health monitoring
- Activity logs viewer

6.2 Day 26: Billing Dashboard

- Usage overview charts
- Cost breakdown by model
- Invoice history table
- Payment methods management
- Subscription plan selection
- Usage forecasting

6.3 Day 27: Logs & Debugging UI

- Request logs viewer
- Error tracking interface
- Advanced search and filtering
- Log export functionality
- Real-time log streaming
- Debug tools

6.4 Day 28: Team Management UI

- Team creation interface
- Member invitation system
- Role assignment UI
- Team analytics dashboard
- Collaborative workspaces
- Permission management

6.5 Day 29: API Documentation Site

- Interactive docs (Swagger/OpenAPI)
- Code examples in multiple languages
- Tutorials and guides
- Best practices section
- Changelog
- Search functionality

6.6 Day 30: Mobile-Responsive Design

- Mobile optimization
- Progressive Web App (PWA) setup
- Touch interactions
- Responsive layouts for all pages
- Mobile navigation
- Performance optimization

6.7 Day 31: Dark Mode & Themes

- Dark/Light mode toggle
- Custom theme system
- User preference persistence
- Accessibility improvements (WCAG 2.1 AA)
- Color palette management
- High contrast mode

7 Week 6: Scalability & Distribution (Days 32-38)

TYPE: BACKEND

7.1 Day 32: Horizontal Scaling

- Stateless service design
- Load distribution strategies
- Session management (Redis)
- Distributed locks
- Cluster coordination

7.2 Day 33: Load Balancing

- Nginx/HAProxy configuration
- Health check endpoints
- Weighted routing
- Sticky sessions
- SSL termination
- Request routing algorithms

7.3 Day 34: Distributed Caching

- Redis Cluster setup
- Cache sharding strategies
- Consistent hashing
- Cache replication
- Failover mechanisms
- Cache synchronization

7.4 Day 35: Message Queues

- RabbitMQ or Kafka integration
- Async job processing
- Task queues

- Event-driven architecture
- Message persistence
- Dead letter queues

7.5 Day 36: Service Mesh

- Istio setup
- Service discovery
- Traffic management
- Observability
- Security policies
- Circuit breaking at mesh level

7.6 Day 37: Multi-Region Deployment

- Geographic distribution
- Data replication strategies
- Latency optimization
- Region failover
- DNS-based routing
- Global load balancing

7.7 Day 38: CDN Integration

- CloudFlare setup
- Static asset optimization
- Edge caching
- DDoS protection
- SSL/TLS management
- Cache purging strategies

8 Week 7: Advanced ML & AI Features (Days 39-45)

TYPE: FULL-STACK

8.1 Day 39: Streaming UI Components

- Real-time token streaming display
- Typewriter effects
- Stop generation button
- Progress indicators
- SSE client implementation
- Stream error handling

8.2 Day 40: Fine-Tuning System

- Dataset upload interface
- Training configuration UI
- Fine-tuning job management
- Progress monitoring
- Model testing interface
- Cost estimation

8.3 Day 41: Model Versioning

- Version control system
- Version comparison UI
- Rollback functionality
- A/B test configuration
- Performance metrics per version
- Automated testing

8.4 Day 42: RAG Implementation

- Document upload system
- Text chunking and embedding
- Vector search integration
- Knowledge base management UI
- Context injection
- Relevance scoring

8.5 Day 43: Vector Database Integration

- Pinecone/Weaviate/Qdrant setup
- Semantic search implementation
- Embedding generation
- Vector storage optimization
- Search UI interface
- Similarity visualization

8.6 Day 44: Prompt Library System

- Prompt templates database
- Version control for prompts
- Sharing functionality
- Community prompts marketplace
- Prompt testing interface
- Performance analytics

8.7 Day 45: Custom Model Hosting

- Model upload interface
- Configuration wizard
- Deployment automation
- Endpoint generation
- Testing tools
- Resource monitoring

9 Week 8: Real-Time & Collaboration (Days 46-52)

TYPE: FRONTEND

9.1 Day 46: WebSocket Dashboard

- Real-time metrics updates
- Live system status

- Instant notifications
- Active users display
- Connection status indicators
- Auto-reconnection logic

9.2 Day 47: Collaboration Features

- Shared workspaces
- Real-time editing (Operational Transform)
- Comments and annotations
- Version history
- Conflict resolution
- Presence indicators

9.3 Day 48: Notification System

- In-app notifications
- Email alert system
- Webhook notifications
- Notification preferences UI
- Push notifications
- Notification history

9.4 Day 49: Chat Interface

- ChatGPT-style UI
- Conversation history
- Message editing
- Conversation export
- Multi-turn conversations
- Context management

9.5 Day 50: Workflow Builder

- Visual workflow designer
- Drag-and-drop interface
- Conditional logic nodes
- Integration nodes
- Workflow templates
- Execution monitoring

9.6 Day 51: Testing & Playground Enhancements

- Batch testing UI
- Response comparison mode
- Performance benchmarks
- Result export (CSV, JSON)
- Test case management
- Automated testing

9.7 Day 52: Advanced Data Visualization

- Custom chart builder

- Dashboard customization
- Widget library
- Export to PDF/PNG
- Interactive data exploration
- Real-time chart updates

10 Week 9: Production & Launch (Days 53-60)

TYPE: FULL-STACK

10.1 Day 53: UI Performance Optimization

- Code splitting
- Lazy loading components
- Image optimization
- Bundle size reduction
- Caching strategies
- Performance profiling

10.2 Day 54: End-to-End Testing

- Cypress/Playwright setup
- User flow tests
- Visual regression testing
- Cross-browser testing
- API integration tests
- Load testing

10.3 Day 55: Kubernetes Deployment

- Kubernetes cluster setup
- Deployment configurations
- Service definitions
- Ingress configuration
- Auto-scaling policies
- Rolling updates

10.4 Day 56: CI/CD Pipelines

- GitHub Actions workflows
- Automated testing pipeline
- Build optimization
- Deployment automation
- Preview deployments
- Environment management

10.5 Day 57: Monitoring & Observability

- Prometheus setup
- Grafana dashboards
- Alert management
- Log aggregation (ELK/Loki)
- Distributed tracing (Jaeger)

- SLA monitoring

10.6 Day 58: Security Hardening

- HTTPS enforcement
- CSP headers configuration
- XSS prevention
- CSRF protection
- SQL injection prevention
- Security audit
- Penetration testing

10.7 Day 59: Documentation & Help Center

- User guides
- Video tutorials
- FAQ section
- Support chat integration
- Knowledge base
- API reference documentation

10.8 Day 60: Final Polish & Launch

- Bug fixes and refinements
- Performance tuning
- Marketing website
- Launch checklist completion
- Production deployment
- Post-launch monitoring

11 Project Deliverables

11.1 Backend Components

- RESTful API (60+ endpoints)
- WebSocket server
- Authentication system
- Intelligent routing engine
- Worker pool system
- Connection pooling
- Request batching
- Caching layer
- Monitoring system
- Admin tools

11.2 Frontend Components

- Dashboard (20+ pages)
- API playground
- Analytics interface
- Admin panel
- Documentation site
- Mobile-responsive design

- Dark mode support
- Real-time features

11.3 Infrastructure

- Docker containers
- Kubernetes manifests
- CI/CD pipelines
- Monitoring stack
- Load balancing
- CDN configuration
- Security setup

11.4 Documentation

- API documentation
- User guides
- Deployment guides
- Architecture documentation
- Code documentation
- Video tutorials

12 Success Metrics

12.1 Performance Targets

- P99 latency < 500ms
- Throughput > 1000 req/sec
- Cache hit rate $> 50\%$
- Uptime $> 99.9\%$
- Error rate $< 0.1\%$

12.2 Cost Optimization

- 95% cost reduction via batching
- 30% resource efficiency via pooling
- 50% cache savings

12.3 Code Quality

- Test coverage $> 80\%$
- Zero critical security vulnerabilities
- All endpoints documented
- Code review compliance

13 Technical Specifications

13.1 Backend Stack

13.2 Frontend Stack

13.3 Infrastructure

Component	Technology
Language	Go 1.21+
Database	PostgreSQL 15+
Cache	Redis 7+
ORM	GORM
HTTP Framework	net/http
WebSocket	gorilla/websocket
Testing	testify

Component	Technology
Framework	Next.js 14
UI Library	React 18
Styling	TailwindCSS
Components	shadcn/ui
State Management	Zustand
Data Fetching	React Query
Charts	Recharts, Chart.js
Forms	React Hook Form + Zod

14 Conclusion

This 60-day roadmap provides a comprehensive path to building a production-grade LLM inference engine. The project is structured to deliver:

- **Week 1-2:** Core backend with advanced features
- **Week 3-5:** Complete frontend with enterprise UI
- **Week 6-7:** Scalability and advanced AI features
- **Week 8-9:** Real-time capabilities and production launch

Expected Outcomes:

- Enterprise-ready LLM platform
- 15x performance improvement
- 95% cost optimization
- Full-stack modern architecture
- Production deployment

Project Start: Day 1
Current Progress: Day 5 (8.3% complete)
Estimated Completion: Day 60

Component	**Technology**
Containerization	Docker
Orchestration	Kubernetes
CI/CD	GitHub Actions
Monitoring	Prometheus + Grafana
Logging	ELK Stack / Loki
Load Balancer	Nginx / HAProxy
CDN	CloudFlare
Cloud Provider	AWS / GCP / Azure