YTEmpire Week 0 Execution Plan

Leadership Team

Role: CEO/Founder

Task 1: Team Kickoff and Vision Alignment

Description: Conduct all-hands meeting to align entire team on YTEmpire's vision and MVP goals. **Steps**:

1. Prepare presentation covering business model, target metrics ($10K/month per user), and 90-day timeline
2. Host 2-hour kickoff meeting with all 17 team members
3. Document Q&A responses and concerns raised
4. Create shared vision document in Confluence **Duration**: 4 hours **Dependencies**: None **Deliverable**: Vision document and recorded kickoff meeting **Priority**: P0

Task 2: Equity and Compensation Finalization

Description: Complete all employment agreements and equity grants for team members. **Steps**:

1. Review equity pool allocation (ensuring sufficient runway)
2. Execute employment agreements with all 17 team members
3. Set up payroll and benefits systems
4. Document equity vesting schedules **Duration**: 8 hours **Dependencies**: Legal counsel availability **Deliverable**: Executed agreements and HRIS setup **Priority**: P0

Task 3: Investor Communication Setup

Description: Establish regular investor update cadence and initial communication. **Steps**:

1. Create investor update template
2. Send Week 0 kickoff announcement
3. Schedule bi-weekly update calls
4. Set up investor Slack channel for async updates **Duration**: 3 hours **Dependencies**: None **Deliverable**: First investor update sent **Priority**: P2

Role: CTO/Technical Director

Task 1: Technical Architecture Documentation

Description: Create comprehensive technical architecture document for all teams to reference. **Steps**:

1. Document service boundaries and data flow diagrams

2. Define API contract standards and naming conventions
3. Establish technology choices and justifications
4. Create architecture decision records (ADR) template **Duration**: 8 hours **Dependencies**: Initial team input **Deliverable**: Architecture documentation in wiki **Priority**: P0

Task 2: Development Environment Standardization

Description: Define and document standard development environment for all engineers. **Steps**:

1. Create Docker-based development environment specification
2. Document IDE configurations and required plugins
3. Set up shared development seeds and test data
4. Create environment setup scripts **Duration**: 6 hours **Dependencies**: Platform Ops hardware setup **Deliverable**: Dev environment setup guide and scripts **Priority**: P0

Task 3: Cross-Team Communication Protocol

Description: Establish communication channels and meeting cadences for all teams. **Steps**:

1. Create Slack workspace with appropriate channels (#backend, #frontend, #ai, #platform-ops)
2. Schedule recurring cross-team sync meetings
3. Set up GitHub organization and team access
4. Document escalation procedures **Duration**: 4 hours **Dependencies**: Team member onboarding **Deliverable**: Communication matrix and meeting calendar **Priority**: P1

Task 4: Technical Risk Assessment

Description: Identify and document top technical risks with mitigation strategies. **Steps**:

1. Review YouTube API quotas and rate limits
2. Assess GPU processing bottlenecks
3. Document cost-per-video risk factors
4. Create risk register with mitigation plans **Duration**: 4 hours **Dependencies**: Architecture documentation **Deliverable**: Risk assessment document **Priority**: P1

Role: VP of AI

Task 1: AI Infrastructure Requirements Documentation

Description: Define GPU, model serving, and compute requirements for AI pipeline. **Steps**:

1. Document GPU memory requirements for each model type
2. Specify CUDA version and driver requirements

3. Define model serving architecture (Triton/TorchServe)
4. Calculate throughput requirements for 50 videos/day **Duration**: 6 hours **Dependencies**: None **Deliverable**: AI infrastructure requirements doc **Priority**: P0

Task 2: External API Account Setup

Description: Establish and configure all AI service provider accounts. **Steps**:

1. Set up OpenAI API account with GPT-4 access ($500 initial credit)
2. Configure ElevenLabs account for voice synthesis
3. Create Google Cloud TTS backup account
4. Document API keys in secure vault **Duration**: 4 hours **Dependencies**: Budget approval **Deliverable**: Configured API accounts with keys **Priority**: P0

Task 3: Cost Model Development

Description: Create detailed cost model for AI pipeline to ensure <$3/video target. **Steps**:

1. Calculate token costs for GPT-4 script generation
2. Estimate voice synthesis costs per minute
3. Model GPU compute costs for video processing
4. Create cost tracking spreadsheet with alerts **Duration**: 4 hours **Dependencies**: API pricing documentation **Deliverable**: Cost model with per-component breakdown **Priority**: P1

Task 4: AI Team Onboarding

Description: Onboard AI team members and establish working protocols. **Steps**:

1. Conduct AI team kickoff meeting
2. Assign initial research topics to each member
3. Set up ML experiment tracking (MLflow)
4. Create AI development guidelines **Duration**: 4 hours **Dependencies**: Team member availability **Deliverable**: AI team charter and guidelines **Priority**: P1

Role: Product Owner

Task 1: User Journey Documentation

Description: Create detailed user journey maps for beta users. **Steps**:

1. Map onboarding flow from signup to first video
2. Document channel setup wizard requirements
3. Define dashboard information architecture

4. Create wireframes for critical screens **Duration**: 8 hours **Dependencies**: None **Deliverable**: User journey documentation and wireframes **Priority**: P0

Task 2: Success Metrics Definition

Description: Define and document all MVP success metrics and KPIs. **Steps**:

1. Define user success metrics (5 channels, $10K/month)
2. Establish technical KPIs (uptime, processing time)
3. Create cost tracking metrics (<$3/video)
4. Set up measurement framework **Duration**: 4 hours **Dependencies**: CEO vision alignment **Deliverable**: KPI dashboard specification **Priority**: P1

Task 3: Beta User Recruitment Plan

Description: Develop strategy for recruiting and onboarding 10 beta users. **Steps**:

1. Define ideal beta user profile
2. Create recruitment channels list
3. Draft beta user agreement
4. Design feedback collection process **Duration**: 4 hours **Dependencies**: Legal review **Deliverable**: Beta user recruitment plan **Priority**: P2

Technical Team (Under CTO)

Role: Backend Team Lead

Task 1: API Architecture Design

Description: Design RESTful API architecture and establish standards. **Steps**:

1. Define API versioning strategy
2. Create endpoint naming conventions
3. Design authentication/authorization flow
4. Document error response formats **Duration**: 6 hours **Dependencies**: CTO architecture documentation **Deliverable**: API design document with OpenAPI spec template **Priority**: P0

Task 2: Database Schema Design

Description: Create initial database schema for MVP features. **Steps**:

1. Design user and authentication tables
2. Create channel management schema
3. Define video queue and processing tables

4. Set up migration framework (Alembic) **Duration**: 6 hours **Dependencies**: Product requirements **Deliverable**: Database ERD and migration scripts **Priority**: P0

Task 3: Development Environment Setup

Description: Set up local development environment for backend team. **Steps**:

1. Create Docker Compose configuration for PostgreSQL/Redis
2. Set up FastAPI project structure
3. Configure pytest and testing framework
4. Create seed data scripts **Duration**: 4 hours **Dependencies**: Platform Ops Docker setup **Deliverable**: Backend development environment **Priority**: P1

Task 4: CI/CD Pipeline Foundation

Description: Establish basic CI/CD pipeline for backend services. **Steps**:

1. Set up GitHub Actions for backend repository
2. Configure automated testing on PR
3. Create Docker build pipeline
4. Set up code quality checks (pylint, black) **Duration**: 4 hours **Dependencies**: GitHub organization setup **Deliverable**: Working CI/CD pipeline **Priority**: P2

Role: API Developer Engineer

Task 1: FastAPI Project Scaffolding

Description: Create initial FastAPI project structure with best practices. **Steps**:

1. Initialize FastAPI project with proper folder structure
2. Set up Pydantic models for request/response validation
3. Create base API router configuration
4. Implement health check endpoint **Duration**: 4 hours **Dependencies**: Backend lead architecture design **Deliverable**: Base FastAPI application **Priority**: P1

Task 2: Authentication Module Setup

Description: Implement JWT-based authentication system foundation. **Steps**:

1. Create user registration endpoint scaffold
2. Implement JWT token generation logic
3. Set up password hashing with bcrypt

4. Create authentication middleware **Duration**: 6 hours **Dependencies**: Database schema design **Deliverable**: Authentication module code **Priority**: P1

Task 3: API Documentation Configuration

Description: Set up automatic API documentation generation. **Steps**:

1. Configure Swagger/OpenAPI documentation
2. Add example requests/responses
3. Set up ReDoc alternative documentation
4. Create API testing collection in Postman **Duration**: 3 hours **Dependencies**: API scaffolding complete **Deliverable**: Auto-generated API documentation **Priority**: P2

Role: Data Pipeline Engineer

Task 1: Queue System Architecture

Description: Design and implement message queue system for video processing. **Steps**:

1. Set up Celery with Redis as broker
2. Create task queue structure for video pipeline
3. Implement priority queue logic
4. Create dead letter queue for failed jobs **Duration**: 6 hours **Dependencies**: Redis setup **Deliverable**: Working queue system **Priority**: P0

Task 2: Data Flow Documentation

Description: Document complete data flow from request to video generation. **Steps**:

1. Create data flow diagrams for video pipeline
2. Document state transitions in processing
3. Define data retention policies
4. Map integration points with AI services **Duration**: 4 hours **Dependencies**: Architecture documentation **Deliverable**: Data flow documentation **Priority**: P1

Task 3: Monitoring Integration

Description: Set up basic monitoring for data pipeline. **Steps**:

1. Integrate Prometheus metrics for queue depth
2. Create pipeline health check endpoints
3. Set up basic alerting rules

4. Document monitoring procedures **Duration**: 4 hours **Dependencies**: Platform Ops monitoring setup **Deliverable**: Pipeline monitoring configuration **Priority**: P2

Role: Integration Specialist

Task 1: YouTube API Setup

Description: Configure YouTube Data API v3 access and test basic operations. **Steps**:

1. Create Google Cloud project and enable YouTube API
2. Generate OAuth 2.0 credentials for 15 accounts
3. Implement token refresh mechanism
4. Test upload and metadata update endpoints **Duration**: 6 hours **Dependencies**: VP AI approval for accounts **Deliverable**: Working YouTube API integration **Priority**: P0

Task 2: External API Integration Framework

Description: Create reusable framework for external API integrations. **Steps**:

1. Design retry logic with exponential backoff
2. Implement rate limiting mechanism
3. Create API response caching layer
4. Set up circuit breaker pattern **Duration**: 4 hours **Dependencies**: Backend architecture **Deliverable**: API integration framework **Priority**: P1

Task 3: Webhook Infrastructure

Description: Set up webhook receivers for Stripe and other services. **Steps**:

1. Create webhook endpoint structure
2. Implement signature verification
3. Set up event processing queue
4. Create webhook testing tools **Duration**: 4 hours **Dependencies**: API scaffolding **Deliverable**: Webhook receiving infrastructure **Priority**: P2

Role: Frontend Team Lead

Task 1: Frontend Architecture Design

Description: Design React application architecture and component hierarchy. **Steps**:

1. Create component tree diagram
2. Design state management structure with Zustand
3. Define routing strategy with React Router

4. Document code organization standards **Duration**: 6 hours **Dependencies**: Product wireframes **Deliverable**: Frontend architecture document **Priority**: P0

Task 2: Development Environment Setup

Description: Configure frontend development environment with Vite. **Steps**:

1. Initialize React 18 project with Vite
2. Configure TypeScript with strict mode
3. Set up ESLint and Prettier
4. Configure Material-UI theme **Duration**: 4 hours **Dependencies**: None **Deliverable**: Frontend development environment **Priority**: P0

Task 3: Component Library Foundation

Description: Create base component library structure. **Steps**:

1. Set up Storybook for component development
2. Create base layout components
3. Implement theme provider
4. Document component guidelines **Duration**: 4 hours **Dependencies**: Design system requirements **Deliverable**: Component library foundation **Priority**: P1

Role: React Engineer

Task 1: Authentication UI Components

Description: Build login and registration form components. **Steps**:

1. Create login form with Material-UI
2. Build registration form with validation
3. Implement password reset flow UI
4. Add form error handling **Duration**: 6 hours **Dependencies**: Frontend setup complete **Deliverable**: Authentication UI components **Priority**: P1

Task 2: API Client Setup

Description: Configure Axios client for backend communication. **Steps**:

1. Set up Axios with interceptors
2. Implement JWT token management
3. Create API service layer

4. Add request/response logging **Duration**: 4 hours **Dependencies**: Backend API specification **Deliverable**: API client configuration **Priority**: P1

Task 3: Zustand Store Configuration

Description: Set up state management stores. **Steps**:

1. Create authentication store
2. Set up user preferences store
3. Implement persist middleware
4. Add DevTools integration **Duration**: 3 hours **Dependencies**: State management design **Deliverable**: Configured Zustand stores **Priority**: P2

Role: Dashboard Specialist

Task 1: Dashboard Layout Design

Description: Create responsive dashboard layout structure. **Steps**:

1. Build sidebar navigation component
2. Create main content area with grid system
3. Implement responsive breakpoints
4. Add loading states **Duration**: 6 hours **Dependencies**: Frontend architecture **Deliverable**: Dashboard layout components **Priority**: P1

Task 2: Chart Component Research

Description: Evaluate and set up charting library. **Steps**:

1. Compare Recharts vs Chart.js for requirements
2. Create proof-of-concept charts
3. Implement chart wrapper components
4. Document chart usage patterns **Duration**: 4 hours **Dependencies**: Dashboard requirements **Deliverable**: Chart component examples **Priority**: P2

Task 3: Real-time Update Infrastructure

Description: Set up WebSocket connection for live updates. **Steps**:

1. Configure Socket.io client
2. Create connection management hooks
3. Implement reconnection logic

4. Add connection status indicator **Duration**: 4 hours **Dependencies**: Backend WebSocket support **Deliverable**: WebSocket client setup **Priority**: P2

Role: UI/UX Designer

Task 1: Design System Creation

Description: Establish comprehensive design system for YTEmpire. **Steps**:

1. Define color palette and typography scale
2. Create spacing and sizing tokens
3. Design icon set requirements
4. Document accessibility guidelines **Duration**: 8 hours **Dependencies**: Brand guidelines **Deliverable**: Design system documentation **Priority**: P0

Task 2: Critical Screen Mockups

Description: Design high-fidelity mockups for core screens. **Steps**:

1. Design dashboard overview screen
2. Create channel management interface
3. Design video queue visualization
4. Mock up settings pages **Duration**: 8 hours **Dependencies**: User journey documentation **Deliverable**: Figma mockups for 10 screens **Priority**: P1

Task 3: Component Library Specs

Description: Create detailed specifications for UI components. **Steps**:

1. Document button variations and states
2. Specify form field components
3. Design card and list components
4. Create loading and empty states **Duration**: 4 hours **Dependencies**: Design system creation **Deliverable**: Component specification document **Priority**: P2

Role: Platform Ops Lead

Task 1: Hardware Setup and Configuration

Description: Set up and configure the Ryzen 9 9950X3D server for development. **Steps**:

1. Install Ubuntu 22.04 LTS with optimized kernel
2. Configure NVIDIA drivers for RTX 5090
3. Set up RAID configuration for data redundancy

4. Configure network settings and firewall **Duration**: 8 hours **Dependencies**: Hardware delivery **Deliverable**: Operational server with remote access **Priority**: P0

Task 2: Docker Environment Setup

Description: Install and configure Docker ecosystem for all services. **Steps**:

1. Install Docker Engine and Docker Compose
2. Configure GPU support for Docker
3. Set up local Docker registry
4. Create base images for services **Duration**: 4 hours **Dependencies**: Server setup complete **Deliverable**: Working Docker environment **Priority**: P0

Task 3: Monitoring Stack Deployment

Description: Deploy Prometheus and Grafana for monitoring. **Steps**:

1. Deploy Prometheus with node exporter
2. Set up Grafana with initial dashboards
3. Configure GPU monitoring
4. Create alerting rules **Duration**: 4 hours **Dependencies**: Docker environment ready **Deliverable**: Operational monitoring stack **Priority**: P1

Task 4: Backup System Implementation

Description: Set up automated backup system for critical data. **Steps**:

1. Configure automated PostgreSQL backups
2. Set up file system snapshots
3. Implement backup to external drive
4. Create restoration procedures **Duration**: 4 hours **Dependencies**: Storage configuration **Deliverable**: Automated backup system **Priority**: P1

Role: DevOps Engineer

Task 1: CI/CD Pipeline Setup

Description: Create GitHub Actions workflows for all repositories. **Steps**:

1. Set up GitHub organization and repositories
2. Create build workflows for each service
3. Implement automated testing gates

4. Configure Docker image building **Duration**: 6 hours **Dependencies**: Repository structure defined **Deliverable**: Working CI/CD pipelines **Priority**: P1

Task 2: Environment Configuration Management

Description: Set up configuration management for all environments. **Steps**:

1. Create environment variable templates
2. Set up secrets management with git-crypt
3. Document configuration procedures
4. Create environment provisioning scripts **Duration**: 4 hours **Dependencies**: Service requirements documented **Deliverable**: Configuration management system **Priority**: P1

Task 3: Deployment Automation

Description: Create automated deployment scripts. **Steps**:

1. Write Docker Compose orchestration scripts
2. Implement blue-green deployment logic
3. Create rollback procedures
4. Document deployment process **Duration**: 4 hours **Dependencies**: CI/CD pipeline complete **Deliverable**: Deployment automation scripts **Priority**: P2

Role: Security Engineer

Task 1: Security Baseline Configuration

Description: Establish security baseline for all systems. **Steps**:

1. Configure UFW firewall rules
2. Set up Fail2ban for intrusion prevention
3. Implement SSH key-only access
4. Configure audit logging **Duration**: 6 hours **Dependencies**: Server setup complete **Deliverable**: Hardened server configuration **Priority**: P0

Task 2: Secrets Management Setup

Description: Implement secure secrets management system. **Steps**:

1. Set up environment variable encryption
2. Configure API key rotation procedures
3. Implement secret scanning in CI/CD

4. Document secrets handling policies **Duration**: 4 hours **Dependencies**: CI/CD pipeline exists **Deliverable**: Secrets management system **Priority**: P1

Task 3: SSL/TLS Configuration

Description: Set up HTTPS for all services. **Steps**:

1. Generate Let's Encrypt certificates
2. Configure Nginx with SSL
3. Set up automatic renewal
4. Test SSL configuration **Duration**: 3 hours **Dependencies**: Domain names configured **Deliverable**: Working HTTPS setup **Priority**: P2

Role: QA Engineer

Task 1: Test Framework Setup

Description: Establish testing frameworks for all components. **Steps**:

1. Set up Jest for React testing
2. Configure Pytest for backend
3. Install Selenium for E2E tests
4. Create test data generators **Duration**: 6 hours **Dependencies**: Development environments ready **Deliverable**: Testing frameworks configured **Priority**: P1

Task 2: Test Plan Documentation

Description: Create comprehensive test plan for MVP. **Steps**:

1. Define test coverage requirements (70% target)
2. Create test case templates
3. Document testing procedures
4. Set up bug tracking system **Duration**: 4 hours **Dependencies**: Product requirements **Deliverable**: MVP test plan document **Priority**: P1

Task 3: Performance Testing Setup

Description: Configure performance testing tools. **Steps**:

1. Install and configure k6 for load testing
2. Create baseline performance tests
3. Set up performance monitoring

4. Document performance targets **Duration**: 4 hours **Dependencies**: Services deployed **Deliverable**: Performance testing framework **Priority**: P2

AI Team (Under VP of AI)

Role: AI/ML Team Lead

Task 1: AI Pipeline Architecture Design

Description: Design end-to-end AI pipeline for content generation. **Steps**:

1. Document model serving architecture
2. Design inference optimization strategy
3. Create pipeline orchestration plan
4. Define model versioning approach **Duration**: 6 hours **Dependencies**: Infrastructure requirements **Deliverable**: AI pipeline architecture document **Priority**: P0

Task 2: Model Evaluation Framework

Description: Establish framework for evaluating model performance. **Steps**:

1. Define quality metrics for content
2. Create A/B testing framework design
3. Set up model performance tracking
4. Document evaluation procedures **Duration**: 4 hours **Dependencies**: None **Deliverable**: Model evaluation framework **Priority**: P1

Task 3: Team Research Assignments

Description: Assign initial research topics to team members. **Steps**:

1. Allocate trend prediction research
2. Assign prompt engineering tasks
3. Distribute voice synthesis evaluation
4. Schedule research review meetings **Duration**: 3 hours **Dependencies**: Team onboarding complete **Deliverable**: Research assignment matrix **Priority**: P2

Role: ML Engineer

Task 1: GPU Environment Setup

Description: Configure CUDA and deep learning frameworks. **Steps**:

1. Install CUDA 12.x toolkit
2. Set up PyTorch with GPU support

3. Configure TensorFlow GPU
4. Test GPU performance benchmarks **Duration**: 4 hours **Dependencies**: Platform Ops GPU setup **Deliverable**: Working GPU development environment **Priority**: P0

Task 2: Model Serving Infrastructure

Description: Set up initial model serving framework. **Steps**:

1. Install NVIDIA Triton Inference Server
2. Configure model repository structure
3. Create model loading procedures
4. Test inference endpoints **Duration**: 6 hours **Dependencies**: GPU environment ready **Deliverable**: Model serving infrastructure **Priority**: P1

Task 3: Training Pipeline Setup

Description: Create infrastructure for model training. **Steps**:

1. Set up MLflow for experiment tracking
2. Configure data versioning with DVC
3. Create training script templates
4. Set up TensorBoard monitoring **Duration**: 4 hours **Dependencies**: GPU environment ready **Deliverable**: Training pipeline infrastructure **Priority**: P2

Role: Data Engineer (AI Team)

Task 1: Data Lake Architecture

Description: Design data storage architecture for AI training data. **Steps**:

1. Design folder structure for training data
2. Set up data versioning system
3. Create data ingestion pipelines
4. Document data governance policies **Duration**: 6 hours **Dependencies**: Storage allocation **Deliverable**: Data lake architecture design **Priority**: P0

Task 2: Feature Store Foundation

Description: Set up basic feature store for ML features. **Steps**:

1. Design feature storage schema
2. Create feature extraction pipelines
3. Implement feature versioning

4. Document feature definitions **Duration**: 4 hours **Dependencies**: Database setup **Deliverable**: Feature store foundation **Priority**: P1

Task 3: YouTube Data Collection

Description: Set up YouTube trending data collection pipeline. **Steps**:

1. Create YouTube API data fetcher
2. Implement trending video analyzer
3. Set up scheduled data collection
4. Store data in structured format **Duration**: 4 hours **Dependencies**: YouTube API access **Deliverable**: Data collection pipeline **Priority**: P2

Role: Analytics Engineer

Task 1: Metrics Database Design

Description: Design database schema for analytics metrics. **Steps**:

1. Create video performance metrics schema
2. Design channel analytics tables
3. Set up cost tracking tables
4. Create aggregation procedures **Duration**: 4 hours **Dependencies**: Database access **Deliverable**: Analytics database schema **Priority**: P1

Task 2: Reporting Infrastructure

Description: Set up basic reporting and visualization tools. **Steps**:

1. Configure Apache Superset
2. Create initial dashboard templates
3. Set up automated report generation
4. Document metrics definitions **Duration**: 4 hours **Dependencies**: Database setup **Deliverable**: Reporting infrastructure **Priority**: P2

Task 3: Cost Tracking Implementation

Description: Implement detailed cost tracking for AI operations. **Steps**:

1. Create cost allocation model
2. Implement API usage tracking
3. Set up cost alerting thresholds

4. Create cost optimization recommendations **Duration**: 4 hours **Dependencies**: API accounts configured **Deliverable**: Cost tracking system **Priority**: P2

Week 0 Timeline Overview

Day 1 (Monday) - P0 Tasks

- **Morning (9 AM - 1 PM)**:
  - CEO: Team kickoff meeting
  - CTO: Begin architecture documentation
  - Platform Ops Lead: Start server setup
  - VP AI: Document infrastructure requirements
- **Afternoon (2 PM - 6 PM)**:
  - All teams: Complete environment setup
  - Security: Begin security baseline
  - Product Owner: Start user journey documentation

Day 2 (Tuesday) - P0 Completion

- **Morning**:
  - Complete all remaining P0 tasks
  - Backend: Database schema design
  - Frontend: Complete setup tasks
  - AI Team: GPU environment configuration
- **Afternoon**:
  - Cross-team sync meeting
  - Dependency resolution
  - P1 task kickoff

Day 3 (Wednesday) - P1 Tasks

- **Morning**:
  - Backend: API development begins
  - Frontend: Component development
  - AI Team: Model serving setup
- **Afternoon**:
  - Platform Ops: Monitoring deployment
  - Integration: External API setup
  - QA: Test framework configuration

Day 4 (Thursday) - P1 Completion

- **Morning**:
  - Complete P1 critical path items
  - Integration testing of basic setup
  - Documentation updates
- **Afternoon**:
  - Team demos of completed work
  - Dependency validation
  - P2 task planning

Day 5 (Friday) - P2 Tasks & Week Wrap-up

- **Morning**:
  - Complete P2 tasks
  - Final integration testing
  - Documentation finalization
- **Afternoon**:
  - Week 0 retrospective
  - Week 1 planning session
  - Celebration and team building

Success Criteria Checklist

Infrastructure Ready

- □ Server operational with GPU support
- □ Docker environment configured
- □ All development environments accessible
- □ CI/CD pipelines functional
- □ Monitoring dashboards live

Team Alignment

- □ All 17 team members onboarded
- □ Communication channels established
- □ Documentation wikis created
- □ Meeting cadences set
- □ Role responsibilities clear

Technical Foundation

- □ API architecture documented
- □ Database schemas designed
- □ Frontend framework configured
- □ AI pipeline architecture defined
- □ Security baseline implemented

External Integrations

- □ YouTube API access verified
- □ OpenAI account configured
- □ Voice synthesis APIs ready
- □ Payment processing setup initiated
- □ Monitoring tools integrated

Process & Quality

- □ Test frameworks installed
- □ Code review process defined
- □ Deployment procedures documented
- □ Backup systems operational
- □ Cost tracking implemented

Risk Mitigation Completed

Technical Risks Addressed

- □ YouTube API quota management plan
- □ GPU resource allocation strategy
- □ Cost optimization framework
- □ Scaling architecture documented
- □ Disaster recovery procedures

Team Risks Addressed

- □ Knowledge transfer protocols
- □ Documentation standards set
- □ Escalation procedures defined
- □ On-call rotation planned
- □ Cross-training initiated

Handoff Points for Week 1

Backend → Frontend

- API endpoint specifications

- Authentication flow documentation
- WebSocket event definitions

Platform Ops → All Teams

- Development environment access
- CI/CD pipeline usage guides
- Monitoring dashboard links

AI Team → Backend

- Model serving endpoints
- Cost per operation metrics
- Processing time estimates

Product → All Teams

- Prioritized feature list
- Success metrics definition
- User acceptance criteria

Week 0 Deliverables Summary

Documentation Produced

- Technical architecture document
- API specifications
- Database schemas
- User journey maps
- Security policies
- Test plans

Infrastructure Deployed

- Development server operational
- Docker environments configured
- CI/CD pipelines active
- Monitoring stack deployed
- Backup systems running

Team Readiness

- All members onboarded

- Tools and access configured
- Communication established
- Roles and responsibilities clear
- Week 1 plan approved

Document Status: COMPLETE **Total Tasks**: 68 (17 roles × 4 average tasks) **P0 Tasks**: 22 (must complete by Day 2) **P1 Tasks**: 28 (must complete by Day 4) **P2 Tasks**: 18 (complete by Day 5) **Estimated Team Utilization**: 85% capacity **Risk Buffer**: 15% time reserved for unknowns