YTEMpire Week 0 Execution Plan

Executive Leadership

Role: CEO/Founder

Task 1: Strategic Vision Alignment Session

Description: Conduct comprehensive strategy workshop to align all department heads on MVP goals and success metrics. **Steps**:

1. Prepare vision deck with 90-day milestones and $10K/month revenue target
2. Schedule 4-hour workshop with CTO, VP AI, and Product Owner
3. Define clear success metrics: 5 channels/user, <$3/video cost, 95% automation
4. Document decisions and distribute to all team leads **Duration**: 4 hours **Dependencies**: None **Deliverable**: Strategic alignment document with signed commitments **Priority**: P0

Task 2: Budget Allocation and Resource Planning

Description: Finalize budget distribution across teams and approve critical purchases. **Steps**:

1. Review $200K budget allocation proposals from each team lead
2. Approve hardware purchases (Ryzen 9 9950X3D system, RTX 5090)
3. Allocate API credits budget ($10K for Month 1)
4. Set up corporate credit cards and spending limits **Duration**: 3 hours **Dependencies**: Team lead budget requests **Deliverable**: Approved budget spreadsheet with spending authority matrix **Priority**: P0

Task 3: Legal and Compliance Framework

Description: Establish legal structure for YouTube automation and content generation. **Steps**:

1. Review YouTube ToS with legal counsel regarding automation
2. Set up business entity structure for channel ownership
3. Create content copyright and fair use guidelines
4. Document compliance requirements for AI-generated content **Duration**: 4 hours **Dependencies**: Legal counsel availability **Deliverable**: Compliance guidelines document and legal entity registration **Priority**: P1

Role: Product Owner

Task 1: User Journey Mapping

Description: Create detailed user journey maps for the MVP focusing on 5-channel management workflow. **Steps**:

1. Map onboarding flow from signup to first video generation
2. Define channel setup wizard requirements (10 questions max)
3. Document daily user interaction points (target: 15 minutes/day)
4. Create workflow for channel performance monitoring **Duration**: 4 hours **Dependencies**: CEO vision alignment **Deliverable**: User journey diagrams in Figma with annotations **Priority**: P0

Task 2: MVP Feature Prioritization

Description: Define and prioritize core features for 12-week MVP delivery. **Steps**:

1. Create feature matrix with must-have vs nice-to-have classification
2. Map features to user value and technical complexity
3. Define acceptance criteria for each core feature
4. Get sign-off from CTO and VP AI on feasibility **Duration**: 3 hours **Dependencies**: Strategic alignment session **Deliverable**: Prioritized product backlog in JIRA with user stories **Priority**: P0

Task 3: Beta User Recruitment Planning

Description: Develop strategy to recruit and onboard 10 beta users by Week 12. **Steps**:

1. Define ideal beta user profile (digital entrepreneurs, $2-5K budget)
2. Create recruitment channels list (Reddit, Facebook groups, Twitter)
3. Draft beta user agreement and NDA templates
4. Design feedback collection framework **Duration**: 3 hours **Dependencies**: Legal framework completion **Deliverable**: Beta recruitment strategy document with templates **Priority**: P2

Role: CTO/Technical Director

Task 1: Technical Architecture Documentation

Description: Create comprehensive technical architecture blueprint for all teams. **Steps**:

1. Design system architecture diagram (microservices vs monolith decision)
2. Define API contract standards and versioning strategy
3. Document data flow from user input to YouTube upload
4. Specify technology stack choices with justifications **Duration**: 4 hours **Dependencies**: CEO strategic alignment **Deliverable**: Technical architecture document v1.0 in Confluence **Priority**: P0

Task 2: Development Environment Standardization

Description: Set up standardized development environment for all engineers. **Steps**:

1. Create Docker compose files for local development
2. Set up GitHub organization and repository structure
3. Configure CI/CD pipeline templates in GitHub Actions
4. Document setup instructions and troubleshooting guide **Duration**: 4 hours **Dependencies**: None **Deliverable**: Development environment setup guide and Docker configurations **Priority**: P0

Task 3: Security and Infrastructure Planning

Description: Establish security baseline and infrastructure requirements. **Steps**:

1. Define security requirements (OAuth2, JWT, encryption standards)
2. Plan hybrid cloud/local infrastructure allocation
3. Set up VPN and secure access protocols
4. Create disaster recovery initial plan **Duration**: 3 hours **Dependencies**: Budget approval **Deliverable**: Security standards document and infrastructure diagram **Priority**: P1

Task 4: Cross-Team Integration Points

Description: Define integration contracts between all technical teams. **Steps**:

1. Map dependencies between Backend, Frontend, and Platform Ops
2. Create API documentation templates
3. Define code review and merge request processes
4. Set up shared Slack channels and communication protocols **Duration**: 2 hours **Dependencies**: Team leads availability **Deliverable**: Integration matrix and communication protocol document **Priority**: P1

Role: VP of AI

Task 1: AI Model Selection and Cost Analysis

Description: Evaluate and select AI models optimizing for quality and <$3/video cost target. **Steps**:

1. Benchmark GPT-3.5 vs GPT-4 for script generation quality/cost
2. Compare ElevenLabs vs Google TTS for voice synthesis
3. Test Stable Diffusion vs DALL-E for thumbnail generation
4. Create cost projection model for 50 videos/day **Duration**: 4 hours **Dependencies**: API access credentials **Deliverable**: AI model comparison matrix with cost projections **Priority**: P0

Task 2: Content Generation Pipeline Architecture

Description: Design end-to-end pipeline from trend detection to video upload. **Steps**:

1. Map data sources for trend analysis (YouTube API, Google Trends, Reddit)
2. Define pipeline stages: trend → script → voice → video → thumbnail
3. Specify quality gates and approval thresholds
4. Design fallback mechanisms for API failures **Duration**: 4 hours **Dependencies**: CTO architecture approval **Deliverable**: Pipeline architecture diagram with stage specifications **Priority**: P0

Task 3: Prompt Engineering Framework

Description: Develop initial prompt templates for consistent content generation. **Steps**:

1. Create base prompts for 5 content styles (educational, entertainment, news)
2. Design prompt variables for channel personalization
3. Test prompts for consistency and quality
4. Document prompt versioning and A/B testing strategy **Duration**: 3 hours **Dependencies**: AI model selection **Deliverable**: Prompt template library with testing results **Priority**: P1

Technical Teams (Under CTO)

Role: Backend Team Lead

Task 1: API Framework Setup

Description: Initialize FastAPI project structure with best practices. **Steps**:

1. Create FastAPI project with modular structure
2. Set up SQLAlchemy ORM with PostgreSQL connection
3. Implement basic JWT authentication scaffold
4. Configure pytest for unit testing **Duration**: 4 hours **Dependencies**: Development environment setup **Deliverable**: Base API project with authentication in GitHub **Priority**: P0

Task 2: Database Schema Design

Description: Design and implement initial database schema for MVP. **Steps**:

1. Create ERD for users, channels, videos, and costs entities
2. Write Alembic migrations for initial schema
3. Set up Redis for caching and session management
4. Document indexing strategy for performance **Duration**: 3 hours **Dependencies**: Architecture documentation **Deliverable**: Database schema with migrations and documentation **Priority**: P1

Task 3: External API Integration Planning

Description: Map all external API requirements and rate limits. **Steps**:

1. Document YouTube API quotas (10,000 units/day per key)
2. Plan 15-account rotation system for quota management
3. Create API client abstraction layer design
4. Define retry and circuit breaker patterns **Duration**: 2 hours **Dependencies**: VP AI model selection **Deliverable**: External API integration specification document **Priority**: P1

Role: API Developer Engineer

Task 1: User Management Endpoints

Description: Implement core user registration and authentication endpoints. **Steps**:

1. Create /auth/register endpoint with email validation
2. Implement /auth/login with JWT token generation
3. Add /auth/refresh for token refresh flow
4. Create /users/profile CRUD endpoints **Duration**: 4 hours **Dependencies**: Database schema, JWT setup **Deliverable**: Working authentication endpoints with tests **Priority**: P1

Task 2: Channel Management Scaffold

Description: Create basic channel CRUD operations structure. **Steps**:

1. Design channel data model with 5-channel limit validation
2. Implement POST /channels endpoint
3. Create GET /channels with user filtering
4. Add channel status management (active/paused) **Duration**: 3 hours **Dependencies**: User management completion **Deliverable**: Channel management endpoints with validation **Priority**: P2

Role: Data Pipeline Engineer

Task 1: Queue System Setup

Description: Configure Celery with Redis for async task processing. **Steps**:

1. Install and configure Celery with Redis broker
2. Create base task classes for video generation
3. Implement task monitoring and status tracking
4. Set up Flower for queue monitoring **Duration**: 4 hours **Dependencies**: Redis setup **Deliverable**: Working Celery configuration with example tasks **Priority**: P1

Task 2: Cost Tracking Framework

Description: Build foundation for tracking per-video costs. **Steps**:

1. Design cost tracking data model
2. Create cost calculation utilities for each API call
3. Implement cost aggregation functions
4. Add cost alerting thresholds ($3/video limit) **Duration**: 3 hours **Dependencies**: Database schema **Deliverable**: Cost tracking module with unit tests **Priority**: P2

Role: Integration Specialist

Task 1: n8n Workflow Engine Setup

Description: Deploy and configure n8n for workflow automation. **Steps**:

1. Deploy n8n using Docker compose
2. Configure webhook endpoints for API integration
3. Create sample workflow for video generation trigger
4. Set up n8n credentials for external services **Duration**: 3 hours **Dependencies**: Docker environment **Deliverable**: Running n8n instance with sample workflow **Priority**: P1

Task 2: YouTube API Client Implementation

Description: Create robust YouTube API client with quota management. **Steps**:

1. Implement OAuth2 flow for YouTube authentication
2. Create upload client with resumable uploads
3. Add quota tracking and account rotation logic
4. Implement error handling and retry mechanisms **Duration**: 4 hours **Dependencies**: API framework setup **Deliverable**: YouTube API client library with tests **Priority**: P2

Role: Frontend Team Lead

Task 1: React Project Initialization

Description: Set up React 18 project with TypeScript and modern tooling. **Steps**:

1. Initialize Vite project with React 18 and TypeScript
2. Configure ESLint, Prettier, and Husky for code quality
3. Set up Zustand for state management
4. Implement React Router for navigation **Duration**: 3 hours **Dependencies**: Development environment **Deliverable**: Base React project with tooling configured **Priority**: P0

Task 2: Design System Foundation

Description: Establish Material-UI theme and component library structure. **Steps**:

1. Configure Material-UI with custom theme
2. Create base component structure (Button, Input, Card)
3. Set up Storybook for component documentation
4. Define responsive breakpoints (desktop-first) **Duration**: 3 hours **Dependencies**: React project setup **Deliverable**: Design system with 5 base components in Storybook **Priority**: P1

Task 3: API Client Architecture

Description: Build API communication layer with error handling. **Steps**:

1. Create Axios instance with interceptors
2. Implement authentication token management
3. Build API hooks using React Query
4. Add global error handling and toast notifications **Duration**: 2 hours **Dependencies**: Backend API documentation **Deliverable**: API client utilities with authentication flow **Priority**: P1

Role: React Engineer

Task 1: Authentication UI Components

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

1. Create Login form with validation
2. Build Registration component with password requirements
3. Implement protected route wrapper
4. Add loading states and error handling **Duration**: 4 hours **Dependencies**: Design system, API client **Deliverable**: Working auth flow with API integration **Priority**: P1

Task 2: Dashboard Layout Structure

Description: Create main application layout with navigation. **Steps**:

1. Build responsive sidebar navigation component
2. Create header with user menu
3. Implement main content area with routing
4. Add breadcrumb navigation **Duration**: 3 hours **Dependencies**: React Router setup **Deliverable**: Application shell with navigation **Priority**: P2

Role: Dashboard Specialist

Task 1: Recharts Setup and Configuration

Description: Initialize data visualization library for metrics display. **Steps**:

1. Install and configure Recharts
2. Create reusable chart components (Line, Bar, Pie)
3. Implement responsive chart containers
4. Add chart theming to match design system **Duration**: 3 hours **Dependencies**: React project setup **Deliverable**: Chart component library with examples **Priority**: P2

Task 2: Real-time Update Architecture

Description: Design polling and WebSocket strategy for live data. **Steps**:

1. Implement polling hook with configurable intervals
2. Create WebSocket connection manager
3. Build store update mechanisms for real-time data
4. Add connection status indicators **Duration**: 2 hours **Dependencies**: Zustand setup **Deliverable**: Real-time data update utilities **Priority**: P2

Role: UI/UX Designer

Task 1: Design System Documentation

Description: Create comprehensive design system in Figma. **Steps**:

1. Define color palette, typography, and spacing scales
2. Design component library (buttons, inputs, cards)
3. Create desktop-first responsive grid system
4. Document component states and interactions **Duration**: 4 hours **Dependencies**: Brand guidelines **Deliverable**: Figma design system with 15 components **Priority**: P0

Task 2: MVP Screen Wireframes

Description: Design low-fidelity wireframes for core user flows. **Steps**:

1. Create dashboard overview wireframe
2. Design channel management interface
3. Sketch video queue and generation flow
4. Layout settings and configuration screens **Duration**: 4 hours **Dependencies**: User journey maps **Deliverable**: Complete wireframe set for 20 screens **Priority**: P1

Task 3: High-Fidelity Dashboard Mockup

Description: Create detailed dashboard design for development reference. **Steps**:

1. Design metrics cards and KPI displays

2. Create chart layouts and data visualizations
3. Add interactive elements and hover states
4. Export assets and specifications for developers **Duration**: 3 hours **Dependencies**: Wireframe approval **Deliverable**: Pixel-perfect dashboard mockup with specs **Priority**: P2

Role: Platform Ops Lead

Task 1: Local Server Infrastructure Setup

Description: Configure Ryzen 9 9950X3D server for development and testing. **Steps**:

1. Install Ubuntu 22.04 LTS with optimizations
2. Configure NVIDIA drivers for RTX 5090
3. Set up Docker and Docker Compose
4. Implement basic security hardening (firewall, SSH) **Duration**: 4 hours **Dependencies**: Hardware delivery **Deliverable**: Operational server with remote access **Priority**: P0

Task 2: CI/CD Pipeline Foundation

Description: Create GitHub Actions workflow for automated testing and deployment. **Steps**:

1. Set up GitHub Actions runners configuration
2. Create workflow for automated testing on PR
3. Implement Docker image building pipeline
4. Configure deployment scripts for staging **Duration**: 3 hours **Dependencies**: GitHub repository setup **Deliverable**: Working CI/CD pipeline with test workflow **Priority**: P1

Task 3: Monitoring Stack Deployment

Description: Set up Prometheus and Grafana for system monitoring. **Steps**:

1. Deploy Prometheus with Docker
2. Configure Grafana with basic dashboards
3. Set up node exporters for system metrics
4. Create alerting rules for critical thresholds **Duration**: 3 hours **Dependencies**: Docker environment **Deliverable**: Monitoring dashboard with system metrics **Priority**: P2

Role: DevOps Engineer

Task 1: Docker Environment Configuration

Description: Create comprehensive Docker compose setup for all services. **Steps**:

1. Write Docker compose for PostgreSQL, Redis, and API

2. Configure volume mounts for persistence
3. Set up networking between containers
4. Create environment variable management **Duration**: 4 hours **Dependencies**: Service specifications **Deliverable**: docker-compose.yml with all services **Priority**: P0

Task 2: Backup and Recovery Setup

Description: Implement automated backup system for critical data. **Steps**:

1. Configure PostgreSQL automated backups
2. Set up backup rotation (daily/weekly/monthly)
3. Create restoration scripts and test procedures
4. Document recovery time objectives **Duration**: 3 hours **Dependencies**: Database setup **Deliverable**: Backup scripts with restoration procedures **Priority**: P1

Role: Security Engineer

Task 1: Security Baseline Implementation

Description: Establish fundamental security measures for MVP. **Steps**:

1. Configure UFW firewall rules
2. Set up Fail2ban for brute force protection
3. Implement SSL/TLS with Let's Encrypt
4. Create security scanning automation **Duration**: 4 hours **Dependencies**: Server setup **Deliverable**: Hardened server with security documentation **Priority**: P1

Task 2: Secrets Management System

Description: Implement secure handling of API keys and credentials. **Steps**:

1. Set up environment variable encryption
2. Create secrets rotation procedures
3. Implement vault for sensitive data
4. Document access control policies **Duration**: 3 hours **Dependencies**: Infrastructure setup **Deliverable**: Secrets management with documentation **Priority**: P1

Role: QA Engineer

Task 1: Test Framework Setup

Description: Establish testing infrastructure for automated quality assurance. **Steps**:

1. Configure Jest for backend unit testing

2. Set up Cypress for frontend E2E tests
3. Create test data factories
4. Implement test coverage reporting **Duration**: 4 hours **Dependencies**: Development environment **Deliverable**: Testing framework with example tests **Priority**: P2

Task 2: Test Planning Documentation

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

1. Define test scenarios for critical user paths
2. Create test case templates
3. Establish bug severity classifications
4. Document testing timeline and milestones **Duration**: 3 hours **Dependencies**: Feature specifications **Deliverable**: Test plan document with 20 test cases **Priority**: P2

AI Teams (Under VP of AI)

Role: AI/ML Team Lead

Task 1: ML Infrastructure Planning

Description: Design ML model serving architecture for production. **Steps**:

1. Evaluate model serving options (Triton, TorchServe, FastAPI)
2. Design GPU scheduling strategy for inference
3. Plan model versioning and rollback procedures
4. Create performance benchmarking framework **Duration**: 4 hours **Dependencies**: Infrastructure availability **Deliverable**: ML infrastructure design document **Priority**: P0

Task 2: Model Evaluation Framework

Description: Establish metrics and testing procedures for AI models. **Steps**:

1. Define quality metrics for generated content
2. Create A/B testing framework for model comparison
3. Implement automated testing pipeline
4. Set up model performance monitoring **Duration**: 3 hours **Dependencies**: None **Deliverable**: Model evaluation framework with metrics **Priority**: P1

Task 3: Team Coordination Setup

Description: Establish AI team workflows and collaboration tools. **Steps**:

1. Set up MLflow for experiment tracking

2. Configure shared Jupyter environment
3. Create model documentation templates
4. Establish code review process for ML code **Duration**: 2 hours **Dependencies**: Development environment **Deliverable**: AI team workspace with collaboration tools **Priority**: P2

Role: ML Engineer

Task 1: Trend Prediction Model Prototype

Description: Build initial trend detection system using YouTube data. **Steps**:

1. Collect sample YouTube trending data
2. Implement basic time series analysis
3. Create feature extraction pipeline
4. Train initial Prophet model for forecasting **Duration**: 4 hours **Dependencies**: YouTube API access **Deliverable**: Working trend prediction prototype **Priority**: P1

Task 2: Model Serving API Setup

Description: Create FastAPI endpoints for model inference. **Steps**:

1. Design inference API contract
2. Implement model loading and caching
3. Create batch prediction endpoints
4. Add performance monitoring **Duration**: 3 hours **Dependencies**: API framework **Deliverable**: Model serving API with documentation **Priority**: P2

Role: Data Team Lead

Task 1: Data Pipeline Architecture

Description: Design data collection and processing infrastructure. **Steps**:

1. Map data sources and ingestion requirements
2. Design ETL pipeline architecture
3. Plan data warehouse schema
4. Create data quality monitoring strategy **Duration**: 4 hours **Dependencies**: Technical architecture **Deliverable**: Data pipeline architecture document **Priority**: P0

Task 2: Analytics Database Setup

Description: Initialize analytics data storage and access patterns. **Steps**:

1. Set up PostgreSQL analytics schema

2. Configure read replicas for reporting
3. Create data access layer
4. Implement caching strategy **Duration**: 3 hours **Dependencies**: Database setup **Deliverable**: Analytics database with access patterns **Priority**: P1

Role: Data Engineer

Task 1: YouTube Data Collector

Description: Build automated YouTube metrics collection system. **Steps**:

1. Implement YouTube Analytics API client
2. Create scheduled data collection jobs
3. Design data normalization procedures
4. Set up data validation checks **Duration**: 4 hours **Dependencies**: YouTube API access **Deliverable**: Automated data collection pipeline **Priority**: P1

Task 2: Cost Analytics Pipeline

Description: Create system for tracking and analyzing per-video costs. **Steps**:

1. Design cost data model
2. Implement API usage trackers
3. Create cost aggregation jobs
4. Build cost reporting endpoints **Duration**: 3 hours **Dependencies**: Database schema **Deliverable**: Cost tracking pipeline with reports **Priority**: P2

Role: Analytics Engineer

Task 1: Metrics Definition and Calculation

Description: Define KPIs and implement calculation logic. **Steps**:

1. Define core metrics (CAC, LTV, engagement)
2. Create SQL queries for metric calculation
3. Implement metric caching layer
4. Document metric definitions **Duration**: 3 hours **Dependencies**: Data schema **Deliverable**: Metrics calculation library with documentation **Priority**: P2

Task 2: Reporting API Development

Description: Build API endpoints for analytics dashboards. **Steps**:

1. Design reporting API structure

2. Implement aggregation endpoints
3. Add time-series data endpoints
4. Create export functionality **Duration**: 3 hours **Dependencies**: Analytics database **Deliverable**: Reporting API with example queries **Priority**: P2

Week 0 Timeline Overview

Day 1-2 (P0 Tasks)

- **All Teams**: Environment setup, tool installation, access configuration
- **Leadership**: Strategic alignment, budget approval, legal framework
- **Technical Leads**: Architecture documentation, infrastructure setup

Day 3-4 (P1 Tasks)

- **Backend**: API scaffolding, database design, authentication
- **Frontend**: React setup, design system, component library
- **AI Team**: Model selection, pipeline architecture, cost analysis
- **Platform Ops**: CI/CD setup, monitoring deployment

Day 5 (P2 Tasks)

- **All Teams**: Integration testing, documentation review, handoff preparation
- **QA**: Test framework completion, test planning
- **Product**: Beta user recruitment planning

Success Criteria for Week 0

1. **Development Environment**: All 17 team members have working local environment
2. **Architecture**: Complete technical architecture documented and approved
3. **Infrastructure**: Ryzen server operational with Docker stack running
4. **API Foundation**: Basic authentication and user management working
5. **Frontend Shell**: React application with routing and design system
6. **AI Pipeline**: Cost projections validated, model selection complete
7. **CI/CD**: Automated testing running on every commit
8. **Documentation**: All architectural decisions documented in Confluence
9. **Team Alignment**: Every team member clear on Week 1 deliverables
10. **Cost Tracking**: Mechanisms in place to monitor <$3/video target

Risk Mitigation for Week 0

1. **Hardware Delays**: Have cloud backup plan ready (AWS/GCP)
2. **API Access Issues**: Pre-register all API accounts in advance
3. **Team Availability**: Identify backup resources for critical roles
4. **Technical Blockers**: Daily standup at 9 AM to surface issues early
5. **Integration Failures**: End-of-day integration testing sessions

Handoff Points for Week 1

1. **Backend** → **Frontend**: API documentation and authentication flow
2. **AI** → **Backend**: Model serving endpoints and cost estimates
3. **Platform Ops** → **All**: CI/CD pipelines and monitoring dashboards
4. **Product** → **All**: Finalized feature specifications and priorities
5. **Frontend** → **Product**: UI mockups for user feedback