YTEmpire Week 0 Execution Plan

Leadership Team

Role: CEO/Founder

Task 1: Strategic Vision Alignment Session (P0)

Description: Conduct all-hands meeting to align entire team on MVP vision, success metrics, and 3-month roadmap. **Steps**:

1. Prepare comprehensive deck covering product vision, market opportunity, and success criteria
2. Schedule 2-hour all-hands meeting for Day 1 morning (9-11 AM)
3. Present vision emphasizing 95% automation goal and $10K/month revenue target
4. Document Q&A and concerns raised for follow-up **Duration**: 4 hours prep + 2 hours meeting **Dependencies**: None **Deliverable**: Vision deck, meeting recording, alignment document

Task 2: Beta User Pipeline Setup (P1)

Description: Establish early access program and identify first 10 beta users from existing network. **Steps**:

1. Create beta user criteria document (ideal customer profile)
2. Reach out to 30 potential beta users via personal network
3. Schedule initial interest calls with respondents
4. Create beta user onboarding checklist **Duration**: 8 hours **Dependencies**: Vision alignment complete **Deliverable**: Beta user pipeline with 30+ qualified leads

Task 3: Investor Update Framework (P2)

Description: Establish weekly investor update cadence and metrics dashboard. **Steps**:

1. Create investor update template with key metrics sections
2. Set up automated data collection points with CTO
3. Draft Week 0 update highlighting team assembly and kickoff
4. Schedule recurring Friday update sends **Duration**: 4 hours **Dependencies**: Metrics definition with Product Owner **Deliverable**: Update template and first investor email

Role: CTO/Technical Director

Task 1: Technical Architecture Documentation (P0)

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

1. Document microservices architecture with service boundaries

2. Define API contract standards and versioning strategy
3. Create data flow diagrams for video processing pipeline
4. Establish technology stack decisions and justifications **Duration**: 8 hours **Dependencies**: None **Deliverable**: Architecture documentation in Confluence

Task 2: Development Environment Standardization (P0)

Description: Set up standardized development environment for all technical team members. **Steps**:

1. Create Docker Compose configuration for local development
2. Write environment setup script (install.sh) for one-command setup
3. Configure VS Code workspace with extensions and settings
4. Test setup process on clean Ubuntu 22.04 installation **Duration**: 6 hours **Dependencies**: None **Deliverable**: Docker configs, setup scripts, documentation

Task 3: GitHub Repository Structure (P0)

Description: Initialize monorepo structure with proper CI/CD foundation. **Steps**:

1. Create GitHub organization and repository structure
2. Set up branch protection rules and PR templates
3. Configure GitHub Actions for basic CI pipeline
4. Create CODEOWNERS file for automatic review assignment **Duration**: 4 hours **Dependencies**: None **Deliverable**: Configured GitHub repository with CI/CD

Task 4: Resource Allocation Planning (P1)

Description: Define compute resource allocation across local server and cloud services. **Steps**:

1. Document GPU allocation strategy (3 concurrent video renders max)
2. Define memory allocation per service (128GB total distribution)
3. Create cost projection model for hybrid infrastructure
4. Set up resource monitoring baseline **Duration**: 4 hours **Dependencies**: Architecture documentation complete **Deliverable**: Resource allocation spreadsheet and monitoring dashboard

Role: VP of AI

Task 1: AI Service Access Setup (P0)

Description: Establish API access and rate limits for all AI services (OpenAI, ElevenLabs, etc.). **Steps**:

1. Create OpenAI organization account with $5,000 initial credit
2. Set up ElevenLabs API with starter plan

3. Configure Google Cloud TTS as fallback option
4. Document API keys in secure vault with access controls **Duration**: 4 hours **Dependencies**: Budget approval from CEO **Deliverable**: Active API accounts with documented rate limits

Task 2: Cost Optimization Strategy (P0)

Description: Design cost control mechanisms to achieve <$3 per video target. **Steps**:

1. Create cost breakdown model per video component
2. Define GPT-3.5 vs GPT-4 usage rules
3. Establish caching strategy for common responses
4. Set up real-time cost tracking webhook **Duration**: 6 hours **Dependencies**: API access setup complete **Deliverable**: Cost optimization playbook and tracking system

Task 3: ML Pipeline Architecture (P1)

Description: Design end-to-end ML pipeline for content generation and quality scoring. **Steps**:

1. Define pipeline stages (trend → script → audio → video → quality)
2. Specify model serving infrastructure requirements
3. Create pipeline orchestration flow diagram
4. Establish SLA targets for each stage **Duration**: 8 hours **Dependencies**: Technical architecture from CTO **Deliverable**: ML pipeline design document

Role: Product Owner

Task 1: MVP Feature Prioritization (P0)

Description: Define and prioritize core features for 3-month MVP delivery. **Steps**:

1. Create feature backlog with user stories
2. Run MoSCoW prioritization exercise
3. Map features to 2-week sprint cycles
4. Create MVP acceptance criteria document **Duration**: 6 hours **Dependencies**: Vision alignment from CEO **Deliverable**: Prioritized product backlog and sprint plan

Task 2: Success Metrics Definition (P0)

Description: Establish quantifiable success metrics for MVP launch. **Steps**:

1. Define primary KPIs (channels managed, automation %, revenue)
2. Create secondary metrics (video quality, user time saved)
3. Set up measurement framework with Data Engineer

4. Create metrics dashboard mockup **Duration**: 4 hours **Dependencies**: Strategic vision session complete **Deliverable**: KPI framework and measurement plan

Task 3: User Journey Mapping (P1)

Description: Document end-to-end user journey from signup to first automated video. **Steps**:

1. Create detailed user flow diagram
2. Identify friction points and automation opportunities
3. Define success checkpoints in journey
4. Map journey stages to development priorities **Duration**: 6 hours **Dependencies**: Feature prioritization complete **Deliverable**: User journey map and experience requirements

Backend Team (Under CTO)

Role: Backend Team Lead

Task 1: API Gateway Setup (P0)

Description: Initialize FastAPI application structure with modular architecture. **Steps**:

1. Create FastAPI project with proper folder structure
2. Implement basic health check and metrics endpoints
3. Set up Swagger/OpenAPI documentation auto-generation
4. Configure CORS and security middleware **Duration**: 4 hours **Dependencies**: GitHub repository ready **Deliverable**: Running API gateway with documentation

Task 2: Database Schema Design (P0)

Description: Design and implement initial database schema for users, channels, and videos. **Steps**:

1. Create ERD for core entities and relationships
2. Write SQL migration scripts using Alembic
3. Set up PostgreSQL with proper indexes
4. Configure Redis for caching layer **Duration**: 6 hours **Dependencies**: Architecture documentation from CTO **Deliverable**: Database schema with migration scripts

Task 3: Team Onboarding Plan (P1)

Description: Create onboarding materials and assign initial tasks to backend team members. **Steps**:

1. Document coding standards and review process
2. Create Jira board with Week 1 sprint tasks
3. Assign domain ownership to team members

4. Schedule daily standup routine **Duration**: 4 hours **Dependencies**: Team members available **Deliverable**: Onboarding docs and sprint board

Role: API Developer Engineer (x2)

Task 1: Authentication Service Setup (P0)

Description: Implement JWT-based authentication with refresh token support. **Steps**:

1. Create user registration endpoint with email verification
2. Implement login endpoint with JWT generation
3. Add refresh token mechanism with proper expiry
4. Create middleware for route protection **Duration**: 8 hours **Dependencies**: API gateway setup complete **Deliverable**: Working authentication system with tests

Task 2: Channel Management CRUD (P1)

Description: Build RESTful endpoints for YouTube channel management. **Steps**:

1. Create channel model with YouTube integration fields
2. Implement CRUD endpoints (GET, POST, PUT, DELETE)
3. Add validation for 5-channel limit per user
4. Write unit tests for all endpoints **Duration**: 6 hours **Dependencies**: Database schema ready **Deliverable**: Channel management API with documentation

Task 3: WebSocket Foundation (P2)

Description: Set up WebSocket support for real-time video processing updates. **Steps**:

1. Configure WebSocket endpoint in FastAPI
2. Implement connection manager for client tracking
3. Create event emission system for progress updates
4. Test with simple echo functionality **Duration**: 4 hours **Dependencies**: API gateway running **Deliverable**: WebSocket infrastructure ready for use

Role: Data Pipeline Engineer (x2)

Task 1: Message Queue Setup (P0)

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

1. Install and configure Celery with Redis broker
2. Create task worker structure for video processing
3. Implement task result backend for status tracking

4. Set up Flower for task monitoring **Duration**: 6 hours **Dependencies**: Redis installation complete **Deliverable**: Working task queue with monitoring

Task 2: Video Processing Pipeline Scaffold (P1)

Description: Create basic pipeline structure for video generation workflow. **Steps**:

1. Define pipeline stages as Celery tasks
2. Implement task chaining for sequential processing
3. Add error handling and retry logic
4. Create pipeline status tracking system **Duration**: 8 hours **Dependencies**: Message queue setup complete **Deliverable**: Video pipeline framework ready for integration

Task 3: Cost Tracking System (P1)

Description: Build real-time cost tracking for API usage and resource consumption. **Steps**:

1. Create cost tracking database table
2. Implement cost calculation functions per service
3. Add cost aggregation endpoints
4. Set up cost threshold alerts **Duration**: 4 hours **Dependencies**: Database schema ready **Deliverable**: Cost tracking system with API endpoints

Role: Integration Specialist

Task 1: YouTube API Integration Setup (P0)

Description: Establish YouTube Data API v3 integration with OAuth 2.0 flow. **Steps**:

1. Create Google Cloud project and enable YouTube API
2. Implement OAuth 2.0 flow for channel authorization
3. Create YouTube service wrapper class
4. Test with basic channel information retrieval **Duration**: 6 hours **Dependencies**: API gateway ready **Deliverable**: Working YouTube OAuth integration

Task 2: N8N Workflow Engine Setup (P1)

Description: Deploy and configure N8N for workflow automation. **Steps**:

1. Deploy N8N using Docker Compose
2. Configure webhook endpoints for API communication
3. Create first test workflow for video processing

4. Document webhook authentication mechanism **Duration**: 6 hours **Dependencies**: Docker environment ready **Deliverable**: N8N instance with test workflow

Task 3: Payment Gateway Initial Setup (P2)

Description: Configure Stripe test environment for subscription management. **Steps**:

1. Create Stripe test account and obtain API keys
2. Install Stripe SDK and configure webhook endpoint
3. Create basic subscription product tiers
4. Implement webhook signature verification **Duration**: 4 hours **Dependencies**: API gateway running **Deliverable**: Stripe test integration ready

Frontend Team (Under CTO)

Role: Frontend Team Lead

Task 1: React Project Initialization (P0)

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

1. Initialize Vite project with React TypeScript template
2. Configure ESLint, Prettier, and Husky pre-commit hooks
3. Set up Material-UI theme and component library
4. Create folder structure for features and components **Duration**: 4 hours **Dependencies**: GitHub repository ready **Deliverable**: Configured React project with tooling

Task 2: State Management Architecture (P1)

Description: Implement Zustand store structure for application state. **Steps**:

1. Install and configure Zustand with TypeScript
2. Create store slices for auth, channels, and videos
3. Implement persistence middleware for local storage
4. Add DevTools integration for debugging **Duration**: 4 hours **Dependencies**: React project initialized **Deliverable**: State management system ready

Task 3: Component Library Foundation (P1)

Description: Create base component library with design system. **Steps**:

1. Define design tokens (colors, spacing, typography)
2. Create base components (Button, Input, Card)
3. Set up Storybook for component documentation

4. Implement dark mode support **Duration**: 6 hours **Dependencies**: Material-UI configured **Deliverable**: Component library with Storybook

Role: React Engineer

Task 1: Authentication UI Flow (P1)

Description: Build login, registration, and password reset interfaces. **Steps**:

1. Create login page with form validation
2. Build registration flow with email verification
3. Implement password reset request interface
4. Add loading states and error handling **Duration**: 8 hours **Dependencies**: Component library ready **Deliverable**: Complete authentication UI flow

Task 2: Dashboard Layout Structure (P1)

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

1. Build responsive sidebar navigation component
2. Create header with user menu and notifications
3. Implement main content area with routing
4. Add breadcrumb navigation system **Duration**: 6 hours **Dependencies**: React Router configured **Deliverable**: Dashboard layout framework

Role: Dashboard Specialist

Task 1: Chart Library Integration (P2)

Description: Set up Recharts for data visualization capabilities. **Steps**:

1. Install and configure Recharts library
2. Create wrapper components for common chart types
3. Implement responsive chart containers
4. Add chart theming to match design system **Duration**: 4 hours **Dependencies**: React project ready **Deliverable**: Chart components ready for use

Task 2: Real-time Data Architecture (P2)

Description: Implement WebSocket client for live updates. **Steps**:

1. Create WebSocket service class with reconnection
2. Integrate with Zustand for state updates
3. Add connection status indicator component

4. Implement event handler system **Duration**: 6 hours **Dependencies**: State management ready **Deliverable**: Real-time update system

Role: UI/UX Designer

Task 1: Design System Documentation (P0)

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

1. Define color palette and typography scale
2. Create component specifications with states
3. Document spacing and layout grid system
4. Export design tokens for development **Duration**: 8 hours **Dependencies**: Brand guidelines from CEO **Deliverable**: Figma design system file

Task 2: MVP Screen Designs (P1)

Description: Design key screens for MVP functionality. **Steps**:

1. Create dashboard overview mockup
2. Design channel management interface
3. Build video queue visualization
4. Create onboarding flow screens **Duration**: 8 hours **Dependencies**: User journey map from Product Owner **Deliverable**: Figma mockups for 10 key screens

Platform Operations Team (Under CTO)

Role: Platform Ops Lead

Task 1: Local Server Setup (P0)

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

1. Install Ubuntu 22.04 LTS with optimized kernel
2. Configure NVIDIA drivers and CUDA 12.x
3. Set up RAID configuration for data redundancy
4. Implement basic firewall and SSH hardening **Duration**: 8 hours **Dependencies**: Hardware delivered **Deliverable**: Operational local server

Task 2: Monitoring Stack Deployment (P1)

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

1. Deploy Prometheus with node and container exporters
2. Configure Grafana with initial dashboards

3. Set up basic alerting rules
4. Create documentation for adding new metrics **Duration**: 6 hours **Dependencies**: Docker environment ready **Deliverable**: Monitoring system with dashboards

Role: DevOps Engineer (x2)

Task 1: Docker Infrastructure Setup (P0)

Description: Create Docker and Docker Compose environment for all services. **Steps**:

1. Install Docker Engine and Docker Compose
2. Configure Docker networks for service isolation
3. Set up local Docker registry for custom images
4. Create base Dockerfiles for each service type **Duration**: 6 hours **Dependencies**: Server setup complete **Deliverable**: Docker infrastructure ready

Task 2: CI/CD Pipeline Foundation (P1)

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

1. Create workflow for automated testing on PR
2. Set up build pipeline for Docker images
3. Implement staging deployment workflow
4. Add security scanning with Snyk **Duration**: 8 hours **Dependencies**: GitHub repository configured **Deliverable**: Working CI/CD pipeline

Task 3: Backup Strategy Implementation (P2)

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

1. Configure automated PostgreSQL backups
2. Set up file system snapshots for media
3. Implement backup rotation policy
4. Test restore procedure **Duration**: 4 hours **Dependencies**: Database deployed **Deliverable**: Backup system with tested restore

Role: Security Engineer (x2)

Task 1: Security Baseline Configuration (P0)

Description: Implement essential security measures for MVP. **Steps**:

1. Configure UFW firewall with strict rules
2. Set up Fail2ban for brute force protection

3. Implement SSH key-only authentication
4. Create security scanning automation **Duration**: 6 hours **Dependencies**: Server access available **Deliverable**: Hardened server environment

Task 2: Secrets Management Setup (P1)

Description: Establish secure storage for API keys and credentials. **Steps**:

1. Deploy HashiCorp Vault or use environment variables
2. Create secret rotation procedures
3. Document access control policies
4. Implement audit logging for secret access **Duration**: 4 hours **Dependencies**: Docker environment ready **Deliverable**: Secrets management system

Task 3: SSL/TLS Configuration (P2)

Description: Set up HTTPS with Let's Encrypt certificates. **Steps**:

1. Configure Nginx as reverse proxy
2. Obtain Let's Encrypt certificates
3. Set up auto-renewal with Certbot
4. Implement HSTS and security headers **Duration**: 4 hours **Dependencies**: Domain names configured **Deliverable**: HTTPS-enabled endpoints

Role: QA Engineer (x2)

Task 1: Test Framework Setup (P1)

Description: Establish automated testing infrastructure. **Steps**:

1. Set up Jest for JavaScript unit testing
2. Configure Pytest for Python backend tests
3. Install Selenium for E2E testing
4. Create test data generation utilities **Duration**: 6 hours **Dependencies**: Development environment ready **Deliverable**: Test frameworks configured

Task 2: Test Environment Creation (P2)

Description: Set up isolated testing environment. **Steps**:

1. Create Docker Compose for test environment
2. Set up test database with seed data
3. Configure test API keys and limits

4. Document test environment usage **Duration**: 4 hours **Dependencies**: Docker infrastructure ready **Deliverable**: Isolated test environment

Task 3: Performance Testing Setup (P2)

Description: Prepare load testing infrastructure. **Steps**:

1. Install and configure k6 for load testing
2. Create basic performance test scripts
3. Set up performance baseline metrics
4. Document performance testing procedures **Duration**: 4 hours **Dependencies**: API endpoints available **Deliverable**: Performance testing capability

AI Team (Under VP of AI)

Role: AI/ML Team Lead

Task 1: Model Serving Architecture (P0)

Description: Design model serving infrastructure for inference at scale. **Steps**:

1. Evaluate serving options (TorchServe, Triton, custom)
2. Create model versioning strategy
3. Design request routing and load balancing
4. Document deployment procedures **Duration**: 6 hours **Dependencies**: ML pipeline architecture from VP **Deliverable**: Model serving design document

Task 2: Team Task Allocation (P1)

Description: Assign specific model development tasks to team members. **Steps**:

1. Break down ML pipeline into components
2. Assign ownership based on expertise
3. Create development timeline
4. Set up progress tracking system **Duration**: 4 hours **Dependencies**: Team members available **Deliverable**: Task assignment matrix

Role: ML Engineer

Task 1: Trend Prediction Prototype (P1)

Description: Build initial trend detection model using Prophet. **Steps**:

1. Set up Prophet and required dependencies
2. Create data ingestion pipeline from YouTube API

3. Train baseline model on historical data
4. Implement basic prediction endpoint **Duration**: 8 hours **Dependencies**: YouTube API access ready **Deliverable**: Working trend prediction prototype

Task 2: Model Evaluation Framework (P2)

Description: Create system for tracking model performance. **Steps**:

1. Set up MLflow for experiment tracking
2. Define evaluation metrics (accuracy, latency)
3. Create automated evaluation pipeline
4. Build performance dashboard **Duration**: 6 hours **Dependencies**: Model serving infrastructure planned **Deliverable**: Model evaluation system

Role: Data Engineer

Task 1: Data Pipeline Architecture (P0)

Description: Design data collection and processing pipeline for training data. **Steps**:

1. Create data schema for video metadata
2. Set up Apache Airflow for pipeline orchestration
3. Implement YouTube Analytics data collector
4. Design feature store structure **Duration**: 8 hours **Dependencies**: Database schema from Backend team **Deliverable**: Data pipeline design and initial implementation

Task 2: Training Data Collection (P1)

Description: Begin collecting initial training dataset. **Steps**:

1. Write scrapers for trending video data
2. Implement data validation and cleaning
3. Set up data versioning with DVC
4. Create data quality monitoring **Duration**: 6 hours **Dependencies**: Data pipeline architecture complete **Deliverable**: Initial training dataset (1000+ videos)

Role: Data Scientist/Analyst

Task 1: Metric Definition and Tracking (P1)

Description: Define success metrics and create tracking system. **Steps**:

1. Define video performance metrics
2. Create quality scoring rubric

3. Set up A/B testing framework
4. Build analytics dashboard mockup **Duration**: 6 hours **Dependencies**: KPI framework from Product Owner **Deliverable**: Metrics tracking system design

Task 2: Competitive Analysis (P2)

Description: Analyze competitor platforms and identify opportunities. **Steps**:

1. Research 5 competitor platforms
2. Document feature comparison matrix
3. Identify unique value propositions
4. Create recommendations report **Duration**: 8 hours **Dependencies**: Product vision from CEO **Deliverable**: Competitive analysis report

Risk Mitigation Checklist

Day 1 Critical Path (P0 Tasks)

- [ ] CEO: Vision alignment meeting completed
- [ ] CTO: Development environment ready
- [ ] VP of AI: AI service accounts created
- [ ] Backend Lead: API gateway running
- [ ] Platform Ops: Server operational
- [ ] Frontend Lead: React project initialized

Day 2-3 Checkpoints (P1 Tasks)

- [ ] Database schema implemented
- [ ] Authentication system functional
- [ ] YouTube API integration working
- [ ] Docker infrastructure complete
- [ ] ML pipeline architecture defined

Day 4-5 Goals (P2 Tasks)

- [ ] CI/CD pipeline operational
- [ ] Test frameworks ready
- [ ] Monitoring dashboards live
- [ ] Initial UI mockups complete
- [ ] Cost tracking active

Success Criteria for Week 0

Technical Milestones

- ✅ Development environment accessible to all team members
- ✅ Core services scaffolded and running locally
- ✅ API documentation available
- ✅ Database schema migrated
- ✅ CI/CD pipeline executing on commits

Team Alignment

- ✅ All team members onboarded and productive
- ✅ Daily standup routine established
- ✅ Communication channels active
- ✅ Task tracking system populated
- ✅ Week 1 sprint planned

Foundation Readiness

- ✅ Architecture decisions documented
- ✅ Security baseline implemented
- ✅ Cost tracking mechanisms in place
- ✅ AI service integrations tested
- ✅ First test video successfully generated

Week 1 Handoff Points

Backend → Frontend

- API endpoints documented and accessible
- Authentication flow ready for integration
- WebSocket events defined

Platform Ops → All Teams

- Docker development environment stable
- CI/CD pipeline ready for use
- Monitoring dashboards accessible

AI Team → Backend

- Model serving endpoints defined
- Data schema requirements communicated
- Cost optimization strategies documented

Product Owner → All Teams

- Sprint 1 backlog prioritized
- Success metrics clearly defined
- User stories ready for implementation