

# Complete AI Development Prompts - Malawi Online Secondary School

## Phase 1: Backend Foundation & Setup

### 1.1 Laravel Project Setup

Create a fresh Laravel 10 project for Malawi Online Secondary School with the following requirements:

- Configure database connection for MySQL
- Set up environment variables for development and production
- Install and configure Laravel Sanctum for API authentication
- Set up CORS middleware for Vue.js frontend
- Configure file storage for video uploads with local and cloud options
- Set up queue workers for background jobs
- Install Laravel Telescope for debugging
- Configure logging with daily rotation
- Set up rate limiting for API endpoints
- Create basic folder structure following Laravel best practices

### 1.2 Database Migrations

Create Laravel migrations for the Malawi Online Secondary School database schema with the following tables:

[Include the entire database schema from the previous artifact here - all tables with proper foreign keys, indexes, and constraints]

Make sure to include:

- Proper foreign key relationships with cascading
- Indexes for performance optimization
- Default values and constraints
- Enum values for status fields
- JSON columns where specified
- Proper timestamp handling

### 1.3 Eloquent Models with Relationships

Create comprehensive Laravel Eloquent models for the Malawi Online Secondary School with the following requirements:

1. User model with:

- Roles and permissions using Spatie Laravel Permission
- Relationships to all related models
- Attribute casting for JSON fields
- Accessors and mutators for computed fields
- Scopes for different user types (students, teachers, admins)

2. Subject model with:

- Form relationships
- Topic relationships
- Teacher assignments
- Pricing relationships
- Preview video handling

3. VideoLesson model with:

- File storage handling
- Progress tracking
- View counting
- Thumbnail generation
- Duration calculation

4. All other models from the database schema with proper relationships, casting, and business logic

Include model factories and seeders for testing data.

## 1.4 Database Seeders

Create comprehensive Laravel database seeders for the Malawi Online Secondary School that populate:

1. Default admin user with proper permissions
2. Sample teachers and students
3. Academic years and terms
4. All 14 Malawian secondary school subjects
5. Forms 1-4 with subject associations
6. Sample topics and video lessons
7. Payment methods (Airtel Money, TNM Mpamba, banks)
8. Subscription plans with Malawian pricing
9. Website settings and content
10. Navigation menus and items
11. Sample testimonials and FAQs
12. Forum categories
13. System settings
14. Sample quizzes and assignments

Use realistic Malawian data including districts, regions, phone numbers, and cultural context.

## **Phase 2: Core Backend APIs**

### **2.1 Authentication & User Management**

Create comprehensive authentication system for Malawi Online Secondary School using Laravel Sanctum with:

1. Registration API with:

- Student registration with form selection
- Teacher application with qualification verification
- Email verification with Malawian context
- Phone number verification using SMS
- Profile completion wizard

2. Login API with:

- Multi-role authentication (admin/teacher/student)
- Remember me functionality
- Rate limiting and security measures
- Activity logging

3. User Profile APIs:

- Profile viewing and updating
- Avatar upload with image processing
- Password change with validation
- Account deactivation
- Parent/guardian contact management

4. Admin user management:

- User listing with filters and search
- Role assignment and permissions
- Account approval/rejection
- Bulk operations

Include proper validation, error handling, and response formatting.

## 2.2 Content Management APIs

Create comprehensive content management APIs for the school platform with:

1. Subject Management:

- CRUD operations for subjects
- Form association management
- Preview video handling
- Subject statistics and analytics

2. Topic Management:

- CRUD operations with proper ordering
- Prerequisite management
- Progress tracking integration

3. Video Lesson APIs:

- Video upload with chunked upload support
- Thumbnail generation and processing
- Video compression and optimization
- CDN integration for faster loading
- Progress tracking and resume functionality
- View analytics and engagement metrics

4. Assignment & Quiz APIs:

- CRUD operations with rich content support
- File attachment handling
- Automated grading where possible
- Deadline management and notifications
- Analytics and performance tracking

5. Live Session Management:

- Session scheduling and management
- Participant registration and tracking
- Recording management
- Integration preparation for video conferencing

Include proper authorization, validation, and file handling.

## 2.3 Payment System APIs

Create manual payment processing APIs for Malawian payment methods:

1. Payment Method Management:

- CRUD for payment methods (Airtel Money, TNM Mpamba, banks)
- Account details management
- Method availability and ordering

2. Subscription Management:

- Package selection and pricing calculation
- Subscription creation with pending status
- Payment submission with file upload for proof
- Reference number and sender details collection

3. Payment Verification System:

- Admin dashboard for payment verification
- Bulk payment processing
- Payment status updates with notifications
- Subscription activation upon verification
- Payment history and reporting

4. Subscription APIs:

- Active subscription checking
- Content access validation
- Renewal reminders and management
- Usage analytics and reporting

Include comprehensive logging, notification system, and audit trails.

## 2.4 Assessment & Progress APIs

Create comprehensive assessment and progress tracking APIs:

1. Quiz Management:

- Quiz creation with multiple question types
- Question bank management
- Randomization and time limits
- Auto-grading for objective questions

2. Quiz Taking APIs:

- Quiz session management
- Answer submission and validation
- Time tracking and auto-submission
- Progress saving and resume functionality

3. Assignment APIs:

- Assignment submission with file uploads
- Plagiarism checking preparation
- Teacher grading interface
- Feedback and revision management

4. Progress Tracking:

- Individual student progress calculation
- Topic completion tracking
- Video watch time and completion
- Overall performance analytics
- Parent progress reports

5. Grading System:

- Gradebook management
- Grade calculation and weighting
- Report card generation
- Performance analytics and insights

Include proper validation, security measures, and performance optimization.

## Phase 3: Advanced Backend Features

### 3.1 Communication & Community APIs

Create comprehensive communication system APIs:

1. Forum System:

- Category and topic management
- Post creation and threading
- Like and reaction system
- Moderation tools and reporting
- Search and filtering

2. Private Messaging:

- Teacher-student communication
- Message threading and attachments
- Read status and notifications
- Bulk messaging for announcements

3. Live Session Communication:

- Session chat and Q&A
- Participant management
- Recording and playback features
- Integration with video platforms

4. Notification System:

- Real-time notifications using WebSockets
- Email notification templates
- SMS integration for critical updates
- Parent notification system
- Notification preferences management

Include proper moderation tools, spam prevention, and communication analytics.

## 3.2 Admin Dashboard APIs



Create comprehensive admin dashboard APIs with:

1. Analytics and Reporting:

- Student enrollment and engagement metrics
- Financial reporting and payment tracking
- Content performance and usage analytics
- Teacher performance metrics
- System health and usage statistics

2. Content Management:

- Website settings and customization APIs
- Landing page content management
- Navigation menu management
- Testimonial and FAQ management
- Media library with search and filtering

3. System Management:

- User management with bulk operations
- Payment verification and processing
- System settings and configuration
- Email template management
- Backup and maintenance tools

4. Reporting and Export:

- Student progress reports
- Financial reports and exports
- Usage analytics and insights
- Custom report generation
- Data export in multiple formats

Include proper caching, performance optimization, and security measures.

### 3.3 Video Optimization & CDN APIs

Create video optimization and delivery system:

1. Video Processing Pipeline:

- Chunked video upload with progress tracking
- Automatic video compression and encoding
- Multiple quality generation (480p, 720p, 1080p)
- Thumbnail extraction and generation
- Video metadata extraction and storage

2. CDN Integration:

- Cloud storage integration (AWS S3, Google Cloud)
- CDN configuration for global delivery
- Adaptive bitrate streaming setup
- Video caching and optimization
- Bandwidth usage tracking and optimization

3. Progressive Loading System:

- Video preloading and buffering optimization
- Adaptive quality based on connection speed
- Resume playback from last position
- Offline video download for premium users
- Video analytics and engagement tracking

4. Performance Optimization:

- Lazy loading implementation
- Video compression algorithms
- Bandwidth-aware streaming
- Cache management and optimization
- Mobile-optimized video delivery

Focus on Malawi's internet infrastructure and mobile-first approach.

## Phase 4: Frontend Foundation

### 4.1 Vue.js Project Setup

Set up a Vue.js 3 project for the Malawi Online Secondary School frontend with:

1. Project Structure:

- Vue 3 with Composition API
- TypeScript configuration
- Tailwind CSS for styling
- Vue Router for navigation
- Pinia for state management
- Axios for API communication

2. Development Environment:

- Vite for fast development builds
- ESLint and Prettier for code quality
- Development proxy for Laravel API
- Hot reload configuration
- Environment variable setup

3. Authentication Setup:

- JWT token management
- Route guards and middleware
- User role-based access control
- Automatic token refresh
- Logout and session management

4. UI Framework Integration:

- Tailwind CSS configuration with custom colors
- Component library setup (Headless UI)
- Icon system (Heroicons or Lucide)
- Responsive design system
- Dark mode support preparation

Include proper folder structure and coding standards.

## 4.2 Core Frontend Components

Create core Vue.js components for the application:

1. Layout Components:

- AppLayout with responsive sidebar/header
- GuestLayout for authentication pages
- AdminLayout with enhanced navigation
- MobileLayout for mobile optimization
- Loading states and error boundaries

2. Authentication Components:

- LoginForm with role selection
- RegisterForm with multi-step wizard
- PasswordReset with email verification
- EmailVerification component
- ProfileUpdate with image upload

3. Navigation Components:

- MainNavigation with role-based menus
- Breadcrumb with dynamic generation
- TabNavigation for content sections
- MobileNavigation with drawer
- UserMenu with profile options

4. UI Components:

- Modal with various sizes and types
- DataTable with sorting and filtering
- FileUpload with drag-drop and progress
- FormInput with validation states
- Button with loading states and variants
- Alert/Toast notification system

Focus on reusability, accessibility, and mobile-first design.

## 4.3 State Management & API Integration

Implement comprehensive state management and API integration:

1. Pinia Store Setup:

- Auth store with user management
- Content store for subjects and lessons
- UI store for global UI state
- Payment store for subscription management
- Progress store for learning analytics

2. API Service Layer:

- Axios configuration with interceptors
- Request/response transformation
- Error handling and retry logic
- Loading state management
- Cache management for static content

3. Authentication Flow:

- Login/logout with token management
- Role-based route protection
- Automatic token refresh
- Session persistence
- Multi-tab synchronization

4. Real-time Updates:

- WebSocket integration for notifications
- Live progress updates
- Chat and messaging real-time features
- Payment status updates
- System announcements

Include proper error handling, loading states, and offline support.

## Phase 5: Student Portal Development

### 5.1 Student Dashboard

Create comprehensive student dashboard with:

1. Dashboard Overview:

- Learning progress visualization
- Upcoming assignments and deadlines
- Recent activities and achievements
- Quick access to favorite subjects
- Performance analytics and insights

2. Subject Browser:

- Subject grid with preview videos
- Form-based filtering and search
- Pricing display and comparison
- Free preview access
- Subscription status indication

3. Learning Progress:

- Topic completion tracking
- Video watch time and progress
- Quiz and assignment scores
- Study streak and achievements
- Personalized recommendations

4. Calendar Integration:

- Assignment due dates
- Live session schedules
- Exam timetables
- Personal study planning
- Reminder system

Include interactive charts, progress bars, and achievement badges.

## 5.2 Video Player & Learning Interface

Create advanced video learning interface with:

1. Custom Video Player:

- HTML5 video with custom controls
- Playback speed control (0.5x to 2x)
- Quality selection based on connection
- Fullscreen and theater modes
- Keyboard shortcuts and accessibility

2. Progressive Loading & Optimization:

- Adaptive bitrate streaming
- Preloading next videos
- Bandwidth detection and optimization
- Resume from last position
- Offline download for premium users

3. Interactive Learning Features:

- Video bookmarks and notes
- Discussion threads at specific timestamps
- Quiz integration within videos
- Interactive transcripts and captions
- Related content suggestions

4. Progress Tracking:

- Watch time tracking
- Completion percentage
- Video engagement analytics
- Learning pace analysis
- Automatic progress saving

5. Mobile Optimization:

- Touch-friendly controls
- Gesture support (swipe, pinch)
- Picture-in-picture mode
- Mobile data usage optimization
- Offline viewing capabilities

Focus on fast loading, smooth playback, and mobile experience for Malawian users.

## 5.3 Assessment & Quiz Interface

Create comprehensive quiz and assessment interface:

1. Quiz Taking Interface:

- Clean, distraction-free design
- Question navigation and progress
- Timer with visual countdown
- Auto-save and resume functionality
- Multiple question type support

2. Question Types:

- Multiple choice with single/multiple answers
- True/false with explanations
- Short answer with character limits
- Essay questions with rich text editor
- Image-based questions and answers

3. Assessment Features:

- Randomized question order
- Time limits with warnings
- Progress saving every 30 seconds
- Review mode before submission
- Detailed results and feedback

4. Assignment Submission:

- File upload with drag-drop
- Multiple file format support
- Plagiarism detection warnings
- Draft saving and versioning
- Submission confirmation and receipt

Include proper validation, security measures, and accessibility features.

## 5.4 Community & Communication Features



Create student community and communication features:

1. Discussion Forums:

- Category-based organization
- Topic creation and threading
- Rich text editor with formatting
- Image and file attachment support
- Like, reply, and reaction system

2. Study Groups:

- Group creation and management
- Private group discussions
- File sharing within groups
- Study session scheduling
- Progress sharing and comparison

3. Peer Communication:

- Private messaging system
- Teacher-student communication
- Group messaging for projects
- Video call integration preparation
- Notification management

4. Social Features:

- Student profiles with achievements
- Leaderboards and competitions
- Study streak tracking
- Achievement badges and rewards
- Success story sharing

Focus on creating a supportive learning community with proper moderation tools.

## **Phase 6: Teacher Portal Development**

### **6.1 Teacher Dashboard & Content Management**

Create comprehensive teacher dashboard with:

1. Teacher Dashboard:

- Class overview and student progress
- Upcoming sessions and deadlines
- Recent student activities
- Teaching analytics and insights
- Quick access to content creation tools

2. Content Management:

- Video upload with chunked processing
- Lesson planning and organization
- Topic creation and sequencing
- Resource attachment and management
- Content preview and testing

3. Video Upload & Processing:

- Drag-drop video upload interface
- Upload progress with pause/resume
- Automatic thumbnail generation
- Video quality preview
- Bulk upload capabilities

4. Lesson Planning:

- Curriculum alignment tools
- Learning objective setting
- Prerequisites and sequencing
- Resource linking and organization
- Lesson notes and supplementary materials

Include proper file handling, progress tracking, and content organization tools.

## 6.2 Assessment Creation & Grading

Create comprehensive assessment and grading tools:

1. Quiz Builder:

- Drag-drop question builder
- Question bank and reuse system
- Multiple question type support
- Randomization and configuration
- Preview and testing tools

2. Assignment Creation:

- Rich text assignment descriptions
- File submission requirements
- Grading rubrics and criteria
- Due date and extension management
- Plagiarism detection settings

3. Grading Interface:

- Efficient grading workflow
- Side-by-side comparison tools
- Audio/video feedback recording
- Batch grading capabilities
- Grade distribution analytics

4. Analytics & Reporting:

- Student performance tracking
- Class progress visualization
- Individual student insights
- Parent report generation
- Performance trend analysis

Focus on teacher efficiency and comprehensive student assessment tools.

## 6.3 Live Session Management

Create live session management system:

1. Session Planning:

- Session scheduling with calendar integration
- Participant invitation and management
- Session agenda and material preparation
- Recording settings and permissions
- Follow-up task assignment

2. Session Delivery:

- Integration with video conferencing platforms
- Screen sharing and presentation tools
- Interactive whiteboard features
- Participant management and controls
- Real-time chat and Q&A

3. Session Management:

- Attendance tracking and reporting
- Recording processing and storage
- Session analytics and engagement metrics
- Follow-up communication tools
- Session feedback collection

4. Student Interaction:

- Breakout room management
- Poll and quiz integration
- Hand-raising and participation tracking
- Collaborative activities
- Session recording access

Prepare for integration with popular video conferencing solutions.

## **Phase 7: Admin Panel Development**

### **7.1 Admin Dashboard & Analytics**

Create comprehensive admin dashboard with:

1. Overview Dashboard:

- Key performance indicators (KPIs)
- Real-time user activity monitoring
- Revenue and subscription metrics
- Content engagement analytics
- System health monitoring

2. User Management:

- User listing with advanced filtering
- Role assignment and permissions
- Account approval and verification
- Bulk operations and exports
- User activity and audit logs

3. Analytics & Reporting:

- Interactive charts and visualizations
- Custom date range filtering
- Export capabilities (PDF, Excel, CSV)
- Automated report generation
- Performance benchmarking

4. System Monitoring:

- Server performance metrics
- Database query optimization
- Error logging and alerting
- Security incident tracking
- Backup and maintenance scheduling

Include real-time updates, interactive charts, and comprehensive filtering options.

## 7.2 Content Management System

Create comprehensive CMS for website content:

1. Website Content Management:

- Dynamic homepage content editing
- Landing page section builder
- Navigation menu management
- Testimonial and FAQ management
- Media library with organization

2. Visual Content Editor:

- WYSIWYG editor for content sections
- Drag-drop section reordering
- Image and video embedding
- Custom CSS injection
- Preview and publish workflow

3. Theme Customization:

- Color scheme customization
- Font and typography settings
- Layout and spacing controls
- Custom CSS editor
- Mobile responsiveness preview

4. SEO & Marketing Tools:

- Meta title and description editing
- Social media integration
- Analytics integration
- A/B testing capabilities
- Marketing campaign management

Focus on user-friendly content editing with live preview capabilities.

## 7.3 Payment & Subscription Management

Create comprehensive payment management system:

1. Payment Verification Dashboard:

- Pending payment queue with filters
- Payment details verification interface
- Bulk approval/rejection tools
- Payment history and audit logs
- Automated verification rules

2. Subscription Management:

- Active subscription monitoring
- Renewal reminder system
- Subscription modification tools
- Refund and cancellation processing
- Revenue analytics and forecasting

3. Financial Reporting:

- Revenue tracking and analysis
- Payment method performance
- Subscription conversion metrics
- Churn analysis and prevention
- Financial forecasting tools

4. Integration Management:

- Payment method configuration
- Bank account management
- Transaction reconciliation
- Automated notification settings
- Fraud detection and prevention

Include comprehensive audit trails and security measures for financial data.

## Phase 8: Mobile Optimization & PWA

### 8.1 Mobile-First Optimization

Optimize the entire application for mobile devices:

1. Responsive Design Enhancement:

- Mobile-first CSS architecture
- Touch-friendly interface elements
- Gesture support implementation
- Screen size optimization for all components
- Mobile navigation patterns

2. Performance Optimization:

- Lazy loading implementation
- Image optimization and compression
- Bundle splitting for faster loads
- Service worker implementation
- Offline functionality

3. Mobile-Specific Features:

- Touch gestures for video controls
- Swipe navigation between lessons
- Pull-to-refresh functionality
- Mobile-optimized forms
- Device camera integration for file uploads

4. Data Usage Optimization:

- Compressed video streaming
- Progressive image loading
- Offline content caching
- Data usage tracking and warnings
- Low-bandwidth mode

Focus on the mobile experience considering Malawi's mobile-first internet usage.

## 8.2 Progressive Web App Implementation



Convert the application to a Progressive Web App:

1. PWA Core Features:

- Service worker for offline functionality
- Web app manifest configuration
- Install prompt and app-like experience
- Push notifications support
- Background sync capabilities

2. Offline Functionality:

- Offline content caching strategy
- Local database synchronization
- Offline form submission queuing
- Cached video playback
- Offline progress tracking

3. Native App Features:

- Home screen installation
- Splash screen customization
- Status bar styling
- Native app navigation feel
- Hardware button integration

4. Performance Optimization:

- App shell architecture
- Critical resource preloading
- Runtime caching strategies
- Background updates
- Performance monitoring

Ensure the PWA works reliably on low-end Android devices common in Malawi.

## Phase 9: Integration & Advanced Features

### 9.1 Video Streaming Optimization

Implement advanced video streaming and optimization:

1. Adaptive Streaming:

- HLS (HTTP Live Streaming) implementation
- DASH (Dynamic Adaptive Streaming) support
- Automatic quality adjustment based on bandwidth
- Multiple resolution encoding pipeline
- CDN integration for global delivery

2. Video Processing Pipeline:

- Automated video encoding and compression
- Thumbnail and preview generation
- Video analytics and engagement tracking
- Subtitle and caption support
- Video security and DRM preparation

3. Optimized Video Player:

- Custom video player with advanced controls
- Buffering optimization for slow connections
- Resume playback across devices
- Keyboard shortcuts and accessibility
- Analytics integration for viewing behavior

4. Mobile Video Optimization:

- Mobile-specific video encoding
- Reduced data usage modes
- Background downloading for premium users
- Picture-in-picture support
- Gesture-based controls

Focus on smooth playback even on slower Malawian internet connections.

## 9.2 Real-time Features & WebSocket Integration

Implement real-time features using WebSockets:

1. Real-time Notifications:

- Instant notification delivery
- Browser push notifications
- In-app notification center
- Notification preferences and management
- Multi-device synchronization

2. Live Features:

- Real-time progress updates
- Live chat during sessions
- Collaborative features for group work
- Real-time quiz participation
- Live polling and Q&A

3. WebSocket Implementation:

- Laravel WebSocket server setup
- Vue.js WebSocket client integration
- Connection management and reconnection
- Message queuing and delivery confirmation
- Scalable WebSocket architecture

4. Performance Optimization:

- Connection pooling and management
- Message compression and batching
- Fallback to HTTP polling
- Bandwidth usage optimization
- Battery usage consideration for mobile

Ensure reliable real-time communication even with intermittent connectivity.

## 9.3 Search & AI Features

Implement advanced search and AI-powered features:

1. Advanced Search System:

- Full-text search across all content
- Faceted search with filters
- Search suggestions and autocomplete
- Search analytics and optimization
- Elasticsearch integration preparation

2. AI-Powered Recommendations:

- Personalized content recommendations
- Learning path suggestions
- Difficulty adjustment based on performance
- Similar student comparison
- Predictive analytics for success

3. Smart Features:

- Automated content tagging
- Intelligent progress predictions
- Adaptive assessment difficulty
- Smart notification timing
- Content quality analysis

4. Analytics & Insights:

- Learning behavior analysis
- Performance prediction models
- Content effectiveness measurement
- Student engagement patterns
- Personalized study recommendations

Focus on features that improve learning outcomes and student engagement.

## Phase 10: Testing & Quality Assurance

### 10.1 Comprehensive Testing Strategy

Implement comprehensive testing for the entire application:

1. Backend Testing:

- Unit tests for all models and services
- Feature tests for API endpoints
- Integration tests for third-party services
- Performance testing for video handling
- Security testing for authentication and payments

2. Frontend Testing:

- Component unit tests with Vue Test Utils
- Integration tests for user workflows
- End-to-end tests with Cypress
- Visual regression testing
- Accessibility testing compliance

3. Mobile Testing:

- Responsive design testing across devices
- Touch interaction testing
- Performance testing on low-end devices
- Network condition simulation
- PWA functionality testing

4. Load Testing:

- Concurrent user simulation
- Video streaming load testing
- Database performance under load
- CDN performance testing
- Payment system stress testing

Include automated testing pipelines and continuous integration setup.

## 10.2 Performance Optimization & Monitoring

Implement performance optimization and monitoring:

1. Backend Performance:

- Database query optimization
- Caching strategy implementation
- API response time optimization
- Video processing optimization
- Background job queue management

2. Frontend Performance:

- Bundle size optimization and code splitting
- Image and video optimization
- Lazy loading implementation
- Critical CSS extraction
- Performance budget monitoring

3. Monitoring Setup:

- Application performance monitoring (APM)
- Error tracking and alerting
- User experience monitoring
- Performance metrics dashboards
- Automated performance alerts

4. Malawi-Specific Optimization:

- Slow connection optimization
- Mobile data usage minimization
- Offline functionality enhancement
- Local CDN setup consideration
- Network resilience improvements

Focus on creating smooth user experiences even with limited internet connectivity.

## Phase 11: Deployment & DevOps

### 11.1 Production Deployment Setup

Set up production-ready deployment infrastructure:

1. Server Configuration:

- Laravel application server setup (Apache/Nginx)
- MySQL database optimization for production
- Redis cache and session storage
- Queue worker setup and monitoring
- SSL certificate installation and HTTPS enforcement

2. Cloud Infrastructure:

- CDN setup for static assets and videos
- Cloud storage for media files
- Database backup and replication
- Load balancer configuration
- Auto-scaling preparation

3. Security Implementation:

- Firewall and security group configuration
- Rate limiting and DDoS protection
- SQL injection and XSS prevention
- CSRF protection and security headers
- Regular security updates and patches

4. Monitoring and Logging:

- Application log monitoring
- Error tracking and alerting
- Performance monitoring dashboards
- Uptime monitoring and alerts
- Security incident detection

Include backup strategies and disaster recovery plans.

## 11.2 CI/CD Pipeline Setup

Implement continuous integration and deployment:

1. Version Control & Branching:

- Git workflow with feature branches
- Code review process setup
- Automated testing on pull requests
- Branch protection rules
- Release management process

2. Build Pipeline:

- Automated testing execution
- Code quality checks and linting
- Security vulnerability scanning
- Performance testing integration
- Documentation generation

3. Deployment Automation:

- Staging environment deployment
- Production deployment with rollback capability
- Database migration automation
- Asset compilation and optimization
- Zero-downtime deployment strategy

4. Monitoring and Alerts:

- Deployment success/failure notifications
- Performance regression detection
- Error rate monitoring post-deployment
- User feedback collection
- Rollback trigger automation

Ensure reliable and safe deployment processes with minimal downtime.

## Phase 12: Launch & Maintenance

### 12.1 Pre-Launch Testing & Optimization



Conduct comprehensive pre-launch testing:

1. User Acceptance Testing:

- Beta testing with real Malawian students
- Teacher training and feedback collection
- Admin workflow validation
- Payment process testing with actual methods
- Mobile device testing across different models

2. Performance Validation:

- Load testing with expected user volumes
- Video streaming testing under various network conditions
- Payment processing validation
- Mobile app installation and usage testing
- Offline functionality verification

3. Content Preparation:

- Educational content review and approval
- Video quality and compression optimization
- Assessment content validation
- Translation accuracy verification
- Cultural appropriateness review

4. Final Optimizations:

- SEO optimization and meta tag completion
- Analytics integration and tracking setup
- Social media integration testing
- Email template testing and delivery
- Documentation completion and user guides

Ensure everything works smoothly for the target Malawian audience.

## 12.2 Launch Strategy & Post-Launch Monitoring

Plan and execute the application launch:

1. Soft Launch Preparation:

- Limited user rollout strategy
- Marketing material preparation
- Support documentation and FAQs
- Customer support team training
- Payment reconciliation process setup

2. Marketing Integration:

- Social media campaign setup
- WhatsApp Business integration
- Radio and local media preparation
- Referral program implementation
- Community outreach strategy

3. Post-Launch Monitoring:

- Real-time user behavior monitoring
- Performance metrics tracking
- Error rate and issue monitoring
- Payment processing monitoring
- User feedback collection and analysis

4. Ongoing Maintenance:

- Regular security updates
- Content updates and new course additions
- Feature enhancements based on user feedback
- Performance optimization continuation
- Scaling preparation as user base grows

Include comprehensive support documentation and training materials.

## Critical Implementation Notes

### AJAX Implementation Focus:

- Use Axios for all API communication with proper interceptors
- Implement loading states and error handling for all AJAX requests
- Use progressive enhancement for better user experience
- Implement proper caching for frequently accessed data
- Handle network failures gracefully with retry mechanisms

### Video Optimization Priority:

- Implement adaptive bitrate streaming for all video content

- Use video compression optimized for mobile devices
- Implement preloading strategies for smoother playback
- Create multiple quality options based on connection speed
- Use CDN for video delivery to reduce loading times

### **Mobile-First Development:**

- Design all components mobile-first then scale up
- Optimize for touch interactions and gestures
- Minimize data usage with smart caching strategies
- Implement offline functionality where possible
- Test extensively on actual devices used in Malawi

### **Malawian Context Considerations:**

- Use local payment methods and currencies
- Implement Chichewa language support
- Consider local internet infrastructure limitations
- Use culturally appropriate design and content
- Optimize for popular mobile devices in the region

This comprehensive prompt list will guide you through building a complete, production-ready online school platform optimized for Malawian students with fast video loading, efficient AJAX communication, and mobile-first design principles.