Subject Line Pro Phase 3: Admin Dashboard Implementation Roadmap

1. Authentication System

Step 1.1: Login Page Design and Implementation

Create responsive login form with username/password fields

Implement form validation and error handling

Add "Remember Me" functionality using localStorage

Design visual feedback for login states (loading, error, success)

Step 1.2: JWT Authentication Setup

Implement JWT generation on backend login endpoint

Create token storage and retrieval methods in frontend

Set up automatic token refresh mechanism

Implement secure token storage using HttpOnly cookies

Step 1.3: Protected Routes Configuration

Create higher-order component for route protection

Implement redirect logic for unauthenticated users

Add route configuration for admin-only pages

Set up authentication state management with context API

Step 1.4: Session Management

Implement token expiration handling

Create session timeout and automatic logout functionality

Add "Stay Logged In" vs. strict session timeout options

Implement session activity monitoring

2. Dashboard Layout

Step 2.1: Admin Layout Design

Create consistent layout with sidebar navigation

Implement responsive design patterns for desktop/tablet/mobile

Design header with user info and quick actions

Add breadcrumb navigation for improved usability

Step 2.2: Dashboard Homepage

Design overview cards with key metrics

Implement quick navigation to frequent tasks

Add recent activity feed showing latest leads and analyses

Create status indicators for system health

Step 2.3: Responsive Implementation

Apply mobile-first approach to dashboard components

Implement collapsible sidebar for smaller screens

Create adaptive data visualizations for various screen sizes

Test and optimize touch interactions for tablet users

3. Analytics Implementation

Step 3.1: Data Aggregation Service

Create backend service for metrics calculation

Implement caching strategy for performance optimization

Design data transformation methods for frontend visualization

Add date range filtering capabilities

Step 3.2: Core Metrics Dashboard

Display total subject lines analyzed

Show lead conversion rates with trend indicators

Visualize average scores (effectiveness, spam risk)

Create engagement metrics dashboard

Step 3.3: Chart Implementation

Implement time-series charts for daily/weekly/monthly analysis

Create distribution charts for score ranges

Add comparison charts for subject line performance

Implement drill-down capabilities for detailed analysis

Step 3.4: Custom Reporting

Create date range selector component

Implement metric selection interface

Add export functionality for reports (PDF, CSV)

Create scheduled report generation capability

4. Lead Management

Step 4.1: Lead Listing Interface

Design tabular view with sortable columns

Implement pagination with configurable page size

Create quick filter options (date, score range, status)

Add bulk action capabilities (export, delete, tag)

Step 4.2: Search and Advanced Filtering

Implement full-text search across lead data

Create advanced filter builder interface

Add saved filter functionality

Implement filter history for quick access

Step 4.3: Lead Detail View

Design comprehensive lead profile view

Show subject line history with scores

Create communication history section

Implement lead status management

Step 4.4: Export Functionality

Create CSV and Excel export services

Implement customizable export fields

Add scheduled/automated exports option

Create export history tracking

5. Content Management

Step 5.1: Spam Trigger Management

Design interface for viewing all spam triggers

Implement add/edit/delete functionality

Create bulk import/export capabilities

Add impact score adjustment features

Step 5.2: Power Word Management

Create power word category organization

Implement CRUD operations for power words

Add effectiveness rating adjustment

Implement search and filter functionality

Step 5.3: Suggestion Template System

Design template editor for customizable suggestions

Create variable insertion system for dynamic content

Implement template testing functionality

Add version history and rollback capabilities

Step 5.4: A/B Testing Framework

Design test creation interface for different analysis rules

Implement results comparison visualization

Add statistical significance indicators

Create test scheduling and automation

6. Settings & Configuration

Step 6.1: Application Settings

Implement general settings management interface

Create email notification configuration

Add API rate limit configuration

Implement frontend customization options

Step 6.2: User Profile Management

Create admin profile edit functionality

Implement password change feature with validation

Add two-factor authentication option

Create login history and device management

Step 6.3: System Monitoring

Implement system health dashboard

Add storage usage visualization

Create performance metrics tracking

Implement automated alert configuration

Step 6.4: Backup & Restore

Create manual and scheduled backup capabilities

Implement restore functionality with validation

Add backup storage location configuration

Create backup retention policy management

7. Security Hardening

Step 7.1: Role-Based Access Control

Design role and permission model (for future expansion)

Implement permission checking middleware

Create frontend permission directive system

Add role-based UI adaptation

Step 7.2: Input Validation Enhancement

Implement comprehensive server-side validation

Add advanced sanitization for all admin inputs

Create validation rule management system

Implement content security policy configuration

Step 7.3: Audit Logging

Design comprehensive audit logging system

Implement log viewing interface with search

Create log retention policy

Add automated suspicious activity detection

Step 7.4: Security Configuration

Implement CSRF protection mechanisms

Add IP restriction capabilities for admin access

Create secure header configuration

Implement security policy documentation

8. Testing & Deployment

Step 8.1: Test Plan Development

Create comprehensive test cases for all admin features

Design integration test suite for admin-frontend-backend flow

Implement automated test scripts

Create user acceptance testing plan

Step 8.2: Testing Execution

Perform unit testing of all components

Execute integration tests for end-to-end flows

Conduct security-focused penetration testing

Perform cross-browser and device compatibility testing

Step 8.3: Deployment Preparation

Create deployment checklist

Implement database backup before deployment

Design rollback strategy

Create deployment documentation

Step 8.4: Production Deployment

Execute staged deployment process

Perform smoke testing after deployment

Monitor system during initial period

Document deployment results and lessons learned

Dependencies & Integration

Frontend Dependencies

JWT Decode: For token handling

React Table: For data grid functionality

Recharts: For analytics visualization

React Hook Form: For advanced form handling

React Query: For efficient data fetching and caching

Backend Dependencies

jsonwebtoken: For JWT generation and validation

express-validator: For enhanced input validation

winston: For advanced logging

node-cron: For scheduled tasks

multer: For file upload handling

Integration Points

Authentication service connects to existing Express routes

Analytics uses existing data storage with new aggregation methods

Lead management extends current lead storage system

Settings configuration stored in new JSON configuration files

This roadmap provides a structured approach to implementing the admin dashboard while building on your existing frontend and backend architecture. Each section is designed to be implemented sequentially, with clear dependencies and integration points.