

Phase 4: Admin Dashboard UI - Implementation Summary

Overview

Implemented a comprehensive admin dashboard UI to make all backend functionality accessible to administrators. All pages are fully functional, responsive, and follow consistent design patterns.

Completed Tasks

1. Reusable Admin Components

Created 8 reusable components in `/components/admin/`:

- **StatusBadge.tsx** - Dynamic status badges with color coding for payments, payouts, refunds, and bookings
- **CurrencyDisplay.tsx** - Formatted currency display with symbol support (EUR, USD, GBP, CHF)
- **DateDisplay.tsx** - Flexible date formatting (short, long, relative, datetime, time) with French locale
- **LoadingSpinner.tsx** - Consistent loading indicators with optional text
- **EmptyState.tsx** - Empty state component with icon, title, description, and optional action
- **DataTable.tsx** - Generic data table with pagination and empty state handling
- **FilterBar.tsx** - Reusable filter bar with select and search inputs
- **Pagination.tsx** - Full pagination component with page size selector

All components are fully typed and exported via `/components/admin/index.ts` for easy imports.

2. Platform Settings Management

File: `/app/dashboard/admin/settings/page.tsx`

Features:

- Complete CRUD for all platform settings
- Three-section layout (Fees, Payouts, General)
- Form fields:
 - Platform Fee Percentage (0-100%)
 - Platform Fee Fixed (optional, in currency)
 - Minimum Payout Amount
 - Holding Period Days (0-30)
 - Payout Mode (AUTOMATIC/MANUAL)
 - Payout Frequency Options (multi-select: DAILY, WEEKLY, MONTHLY)
 - Currency (EUR, USD, GBP, CHF)
- Real-time validation
- Stripe mode indicator (test/live)
- Last updated timestamp
- Help text for each field

3. Creator Stripe Account Management

File: `/app/dashboard/admin/creators/[id]/stripe/page.tsx`

Features:

- Account Status section:
- Stripe Account ID
- Onboarding status (complete/incomplete)
- Charges enabled/disabled
- Payouts enabled/disabled
 - Balance Information section:
 - Available balance from Stripe
 - Pending balance
 - Manual refresh button
 - Payout Settings section:
 - Current mode (AUTOMATIC/MANUAL)
 - Frequency (DAILY/WEEKLY/MONTHLY)
 - Active status
 - Next payout date
 - Edit modal for updating settings
 - Payout Control section:
 - Block/Unblock toggle
 - Reason input (required when blocking)
 - Current blocked status with reason
 - Eligibility Check section:
 - Manual eligibility verification
 - Detailed reasons if not eligible
 - Available balance display
 - Payout History:
 - Last 10 payouts with status
 - Failure reasons
 - Stripe Payout IDs
 - Test Mode Feature:
 - Stripe Express dashboard link generation note
 - Only visible in test/dev mode

4. Enhanced Payments History Page

Files:

- `/app/api/admin/payments/route.ts` (new API endpoint)
- `/app/dashboard/admin/payments/page.tsx`

Features:

- Comprehensive payment list with filtering
- Columns: ID, User, Creator, Amount, Status, Refunded Amount, Date
- Filtering:
- Status (SUCCEEDED, FAILED, PENDING, REFUNDED)
- Search by payment ID or user email
- Creator filter
- Pagination (25, 50, 100 per page)
- Payment Details Modal:
- Full payment information
- Stripe Payment Intent ID
- Booking details
- User and creator information
- Refund history
- Dispute status (if any)
- Platform fee breakdown
- Click row to view details
- Total count display

5. Enhanced Payouts History Page

File: `/app/dashboard/admin/payouts/page.tsx`

Features:

- Comprehensive payout list with filtering
- Columns: ID, Creator, Amount, Status, Failure Reason, Retry Count, Created Date, Paid Date
- Filtering:
- Status (PAID, PENDING, PROCESSING, FAILED)
- Creator filter
- Payout Details Modal:
- Full payout information
- Stripe Payout ID
- Creator details
- Retry count
- Failure reason (if failed)
- Click row to view details

6. Refunds Management Page

File: `/app/dashboard/admin/refunds/page.tsx`

Features:

- Complete refund history with filtering
- Create new refund functionality
- Columns: Refund ID, Payment ID, User, Creator, Amount, Reason, Status, Date
- Filtering:
- Status (SUCCEEDED, PENDING, FAILED)
- Search by payment ID
- Pagination (25, 50, 100 per page)
- Create Refund Modal:
- Payment ID input
- Amount input (optional, defaults to full refund)

- Reason textarea (required)
- Validation
- Refund Details Modal:
- Full refund information
- Original payment details
- Initiated by (admin name)
- Stripe Refund ID

7. Payout Dashboard Overview

File: `/app/dashboard/admin/payouts/dashboard/page.tsx`

Features:

- Statistics Cards:
- Total Pending Payouts (count + amount)
- Failed Payouts (last 30 days)
- Blocked Creators count
- Creators with Eligibility Issues count
- Total Payout Volume (last 30 days)
- Next Scheduled Payout Date
- Recent Failed Payouts List:
 - Last 10 failed payouts
 - Creator name and failure reason
 - Link to creator Stripe account
- Blocked Creators List:
 - All blocked creators
 - Block reason
 - Quick unblock button
- Eligibility Issues List:
 - Creators with eligibility problems
 - Detailed reasons
 - Link to resolve
- Manual Payout Trigger:
 - Button to manually trigger payout processing
 - Calls `/api/cron/process-payouts`
 - Confirmation dialog

8. Transaction Logs Viewer

File: `/app/dashboard/admin/logs/page.tsx`

Features:

- Complete transaction log history
- Columns: Timestamp, Event Type, Entity Type, Entity ID, Stripe Event ID, Amount, Status
- Filtering:
- Event Type (15+ event types)

- Entity Type (PAYMENT, PAYOUT, REFUND, CREATOR)
- Status (SUCCESS, ERROR)
- Search by entity ID or Stripe Event ID
- Pagination (100 per page by default)
- Auto-refresh option (every 30 seconds)
- Log Details Modal:
- Full log information
- Metadata (JSON formatted)
- Copy button for Stripe Event ID
- Error message display
- Color-coded event type badges

9. Navigation Updates

File: `/app/dashboard/admin/page.tsx`

Added “Gestion Administrative” quick access card with buttons for:

- Paramètres (Platform Settings)
- Paiements (Payments History)
- Paiements créateurs (Payouts History)
- Tableau de bord (Payout Dashboard)
- Remboursements (Refunds Management)
- Logs (Transaction Logs)

All buttons navigate to the respective admin pages.



UI/UX Features

Consistent Design

- All pages use shadcn/ui components
- Tailwind CSS for styling
- Purple-pink gradient theme
- Responsive design (mobile-friendly)
- French language interface

User Experience

- Loading states with spinners
- Toast notifications for success/error
- Empty states with helpful messages
- Confirmation dialogs for destructive actions
- Real-time data refresh
- Breadcrumbs and navigation
- Icons from lucide-react
- Hover effects and transitions

Data Display

- Formatted dates with date-fns (French locale)
- Currency display with symbols
- Status badges with color coding

- Pagination with page size selector
- Filtering with reset option
- Search functionality

Technical Implementation

Authentication & Authorization

All admin pages:

- Check user authentication (redirect to login if not authenticated)
- Verify admin role (return 403 if not admin)
- Use cookies for auth token
- Show loading states during auth check

API Integration

All pages integrate with existing backend APIs:

- GET /api/admin/settings
- PUT /api/admin/settings
- GET /api/admin/creators/[id]/payout-settings
- PUT /api/admin/creators/[id]/payout-settings
- POST /api/admin/creators/[id]/block-payout
- GET /api/admin/payments (new)
- GET /api/admin/payouts
- GET /api/admin/payouts/dashboard
- GET /api/admin/refunds
- POST /api/admin/refunds
- GET /api/admin/logs
- POST /api/cron/process-payouts

New API Endpoint

Created `/app/api/admin/payments/route.ts` :

- GET endpoint with pagination and filtering
- Includes user, creator, and booking data
- Returns refunds for each payment
- Supports search by payment ID or email

TypeScript

- All components fully typed
- No TypeScript compilation errors
- Interfaces for all data structures
- Type-safe API responses

Dependencies

Added:

- date-fns (for date formatting and localization)



File Structure

```

/home/ubuntu/callastar/
├── components/admin/
│   ├── StatusBadge.tsx
│   ├── CurrencyDisplay.tsx
│   ├── DateDisplay.tsx
│   ├── LoadingSpinner.tsx
│   ├── EmptyState.tsx
│   ├── DataTable.tsx
│   ├── FilterBar.tsx
│   ├── Pagination.tsx
│   └── index.ts
├── app/
│   ├── api/admin/
│   │   ├── payments/
│   │   │   └── route.ts (NEW)
│   │   └── dashboard/admin/
│   │       ├── page.tsx (UPDATED)
│   │       ├── settings/
│   │       │   ├── page.tsx (UPDATED)
│   │       │   ├── creators/[id]/stripe/
│   │       │   │   ├── page.tsx (NEW)
│   │       │   │   ├── payments/
│   │       │   │   │   ├── page.tsx (NEW)
│   │       │   │   │   ├── payouts/
│   │       │   │   │   │   ├── page.tsx (NEW)
│   │       │   │   │   │   ├── dashboard/
│   │       │   │   │   │   │   ├── page.tsx (NEW)
│   │       │   │   │   │   └── refunds/
│   │       │   │   │   │       ├── page.tsx (NEW)
│   │       │   │   │   │       └── logs/
│   │       │   │   │   │           └── page.tsx (NEW)
│   │       └── PHASE4_IMPLEMENTATION_SUMMARY.md (THIS FILE)

```



Success Criteria

All success criteria have been met:

- ✓ Platform settings page with full CRUD functionality
- ✓ Creator Stripe account management page with all details
- ✓ Enhanced payment history page with filtering
- ✓ Enhanced payout history page with filtering
- ✓ Refund management page with create functionality
- ✓ Payout dashboard overview with statistics
- ✓ Transaction logs viewer with filtering
- ✓ All pages properly authenticated and admin-only
- ✓ Consistent UI/UX across all pages
- ✓ No TypeScript compilation errors
- ✓ Responsive design
- ✓ Proper error handling and loading states



Build Status

Build completed successfully:

```
npm run build
✓ Compiled successfully
```

All new admin pages are built and ready for deployment:

- /dashboard/admin/creators/[id]/stripe (9.42 kB)
- /dashboard/admin/logs (6.04 kB)
- /dashboard/admin/payments (3.73 kB)
- /dashboard/admin/payouts (3.45 kB)
- /dashboard/admin/payouts/dashboard (3.36 kB)
- /dashboard/admin/refunds (4.26 kB)
- /dashboard/admin/settings (5.06 kB)



Usage Notes

Accessing Admin Pages

1. Login as an admin user
2. Navigate to /dashboard/admin
3. Click on any quick access button in “Gestion Administrative” section
4. Use the navigation to explore different pages

Managing Creator Payouts

1. Go to Payout Dashboard to see overview
2. Click on failed/blocked creators to resolve issues
3. Navigate to creator’s Stripe account page to manage settings
4. Block/unblock payouts as needed
5. Manually trigger payout processing if needed

Viewing Transaction Logs

1. Go to Logs page
2. Use filters to find specific events
3. Enable auto-refresh for real-time monitoring
4. Click on any log to view full details

Creating Refunds

1. Go to Refunds page
2. Click “Nouveau remboursement”
3. Enter payment ID and reason
4. Optionally specify partial refund amount
5. Submit to create refund



Security

All pages implement:

- Cookie-based authentication
- Admin role verification
- CSRF protection
- Input validation

- Error handling
- Secure API calls

Localization

All pages are in French:

- UI text
- Date formatting (date-fns French locale)
- Error messages
- Success notifications
- Form labels and placeholders

Performance

- Server-side rendering for initial load
- Client-side navigation for instant page transitions
- Optimized bundle sizes (3-10 kB per page)
- Lazy loading of components
- Pagination to limit data fetching
- Auto-refresh with configurable intervals

Conclusion

Phase 4 is complete! The admin dashboard now provides a comprehensive, user-friendly interface for managing all financial operations on the platform. Every important parameter is editable without code changes, and admins can debug issues without accessing Stripe manually.

The implementation follows best practices:

- Type-safe code
- Reusable components
- Consistent design
- Comprehensive error handling
- Responsive layout
- Accessible UI
- Performance optimization

All pages are production-ready and can be deployed immediately.