

Branch Fix Summary - Payment System Migration

Date: December 25, 2025

Repository: <https://github.com/StreallyX/callastar>

Objective: Move payment system implementation from `main` to `feature/stripe-payout-automation`

Operation Completed Successfully

All payment system commits have been successfully moved from `main` to `feature/stripe-payout-automation` branch, preserving the user's existing work.

What the User Had on `feature/stripe-payout-automation`

The user had already implemented a comprehensive Stripe-controlled payout system with 3 commits:

Commit 1: Backend Infrastructure (9ce8762)

“feat: Implement backend infrastructure for Stripe-controlled payouts”

Created backend APIs and database schema:

- **Admin Payout APIs:**
 - Block/unblock creator payouts
 - View pending payouts
 - Manually trigger payouts
- **Creator APIs:**
 - Payout settings management
 - Request manual payouts
 - View payout history
- **Automation:**
 - Cron job for automatic payout processing
 - Stripe balance checking
- **Database Schema:**
 - Added payout-related enums and models
 - Migration file for database changes
- **Utilities:**
 - `lib/payout-validation.ts` - Comprehensive validation logic

Files Added: 12 files, 2,180+ insertions

Commit 2: Creator Dashboard UI (d79f32c)

“feat: Build creator dashboard features for Stripe-controlled payouts”

Built comprehensive creator-facing UI:

- **Dashboard Pages:**

- Payment setup wizard
- Payout overview and history
- Payout request interface
- Payout settings configuration

- **Navigation:**

- Updated navbar with payout links
- Enhanced creator dashboard navigation

Files Added: 7 files, 1,683+ insertions

Commit 3: Migrations (f1c7cb3)

“migrations”

Fixed migration file structure and TypeScript build configuration.

Files Modified: 3 files



Payment System Commits Applied (4 Commits)

Successfully cherry-picked and applied the following commits from `main`:

1. Phase 1: Foundation (20be8c6 → 37f9de4)

- Schema enhancements (enums, models)
- Settings management system
- Logging infrastructure
- Database seed utilities

2. Phase 2: Refunds & Webhooks (b080fcc → ccf465f)

- Comprehensive refund system
- Advanced webhook handlers
- Webhook testing utilities

3. Phase 3: Automatic Payouts (8b9758d → 1b7e943)

- Complete automatic payout system
- Payout eligibility checking
- Admin payout controls
- Cron job for automated processing

4. Phase 4: Admin Dashboard (d7e10c8 → 767aae4)

- Comprehensive admin UI
 - Payment management
 - Payout dashboard
 - Refund management
 - Transaction logs viewer
-

Conflicts Encountered and Resolutions

Conflict 1: `prisma/schema.prisma`

Issue: Both branches added different enums and modified the `Creator` model.

Resolution:

- **Merged all enums** from both implementations:
- User's enums: `PayoutSchedule`, `PayoutAction`
- Payment system enums: `PayoutMode`, `PayoutFrequency`, `TransactionEventType`, `EntityType`, `RefundStatus`, `DisputeStatus`
- **Merged Creator model fields:**
 - Kept user's payout settings: `payoutSchedule`, `payoutMinimum`, `isPayoutBlocked`, `payoutBlockedReason`
 - Added payment system fields: `payoutBlocked`, `payoutBlockedReason`
 - Merged all relations: `payoutAuditLogs`, `payoutScheduleNew`, `refundsInitiated`
- **Model naming conflict:**
 - Renamed payment system's `PayoutSchedule` model to `PayoutScheduleNew` to avoid conflict with user's `PayoutSchedule` enum
 - Updated all references in codebase accordingly

Conflict 2: TypeScript Compilation Issues

Issues Found:

1. References to `prisma.payoutSchedule` needed to be updated to `prisma.payoutScheduleNew`
2. Variable naming inconsistency: `db` vs `prisma` in various files
3. Missing import statements

Fixed Files:

- `app/api/admin/creators/[id]/payout-settings/route.ts`
- `app/api/admin/payouts/dashboard/route.ts`
- `app/api/admin/payouts/test-eligibility/route.ts`
- `app/api/cron/process-payouts/route.ts`
- `app/api/payments/webhook/route.ts`
- `app/api/stripe/connect-onboard/route.ts`

Final Verification:  TypeScript compilation successful with no errors

Final Branch State

Main Branch (origin/main)

Current HEAD: `c76f1cf` - "try fix"

Commits:

```
c76f1cf try fix
278bfb2 Starting Code
f03a2f5 Initial commit
```

✓ **Cleaned** - All payment system commits removed

Feature Branch (origin/feature/stripe-payout-automation)

Current HEAD: b515d9e - "fix: Resolve schema conflicts..."

Commits (in order):

```
b515d9e fix: Resolve schema conflicts and update references to PayoutScheduleNew
767aae4 feat: Phase 4 - Implement comprehensive admin dashboard UI
1b7e943 Phase 3: Implement complete automatic payout system
ccf465f feat: Phase 2 - Implement comprehensive refund system and webhook handlers
37f9de4 feat(phase1): Implement foundation - schema, settings & logging
f1c7cb3 migrations
d79f32c feat: Build creator dashboard features for Stripe-controlled payouts
9ce8762 feat: Implement backend infrastructure for Stripe-controlled payouts
c76f1cf try fix
278bf62 Starting Code
```

✓ **Complete** - All features integrated



Backup Branches Created

Safety backups were created before any operations:

- backup-main-20251225-182850 - Main branch before reset
- backup-feature-stripe-payout-automation - Feature branch before cherry-picking

These can be deleted after verification, or kept for reference.



Verification Checklist

- ✓ All 4 payment system commits successfully cherry-picked
- ✓ User's original work preserved on feature branch
- ✓ Schema conflicts resolved with all features merged
- ✓ TypeScript compilation passes without errors
- ✓ Main branch reset to clean state (c76f1cf)
- ✓ Feature branch pushed to GitHub
- ✓ Main branch pushed to GitHub
- ✓ Backup branches created for safety



What's Next?

1. Test the feature branch:

```
bash
git checkout feature/stripe-payout-automation
npm install
npx prisma migrate dev
npm run dev
```

2. Review the integrated features:

- Test creator dashboard payout features
- Test admin dashboard payment management
- Verify webhook handling
- Test automatic payout processing

3. When ready to merge:




```
bash
# After thorough testing and review
git checkout main
git merge feature/stripe-payout-automation
git push origin main
```

4. Cleanup (optional):







```
bash
# Delete backup branches if no longer needed
git branch -D backup-main-20251225-182850
git branch -D backup-feature-stripe-payout-automation
```

Key Files Modified/Created




Schema & Database

-  `prisma/schema.prisma` - Merged models and enums
-  `prisma/migrations/` - All migration files
-  `prisma/seed.ts` - Seed utilities





Backend APIs

-  `app/api/admin/payouts/*` - Admin payout management
-  `app/api/admin/refunds/*` - Refund management
-  `app/api/admin/payments/*` - Payment management
-  `app/api/creators/payouts/*` - Creator payout features
-  `app/api/cron/*` - Automated processing jobs
-  `app/api/payments/webhook/route.ts` - Enhanced webhooks

Frontend

-  `app/dashboard/admin/*` - Admin dashboard pages
-  `app/dashboard/creator/payouts/*` - Creator payout pages
-  `components/admin/*` - Reusable admin components

Libraries & Utilities

-  `lib/logger.ts` - Logging system
-  `lib/settings.ts` - Settings management
-  `lib/payout-validation.ts` - Validation logic
-  `lib/payout-eligibility.ts` - Eligibility checking

Important Notes

1. **Database Migration Required:** The schema changes will need to be applied:

```
bash
```

```
npx prisma migrate dev
```

2. **Environment Variables:** Ensure all Stripe-related environment variables are configured
 3. **Testing Recommended:** Thoroughly test all payout workflows before production deployment
 4. **No Data Loss:** All work from both the user and the payment system implementation has been preserved
-

Status:  **COMPLETE - All objectives achieved successfully!**