

Shellff Platform - Post-MVP Features




Overview

This document contains features that are planned for implementation after the MVP (Minimum Viable Product) release. These features will enhance the platform but are not critical for the initial launch.

Code Slice 9 - Part 3: Geo-Restriction System

Status: Post-MVP

Features:

-  **Location-Based Code Redemption**
 - Country/region-specific code locking
 - Geographic validation during redemption
 - Admin-configurable country restrictions
-  **VPN Detection and Handling**
 - Advanced VPN detection algorithms
 - Policy enforcement for VPN usage
 - Bypass detection and blocking
-  **IP Geolocation Services**
 - Real-time location detection
 - Accurate country/region mapping
 - Fallback location services

Database Schema:

```
-- Geo-restriction tables
CREATE TABLE geo_restrictions (
  id SERIAL PRIMARY KEY,
  release_id INTEGER REFERENCES releases(id),
  allowed_countries TEXT[], -- Array of ISO country codes
  blocked_countries TEXT[], -- Array of ISO country codes
  restriction_type VARCHAR(20) DEFAULT 'whitelist', -- 'whitelist', 'blacklist'
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE code_redemption_locations (
  id SERIAL PRIMARY KEY,
  code_id INTEGER REFERENCES unlock_codes(id),
  ip_address INET,
  country_code VARCHAR(2),
  region VARCHAR(100),
  city VARCHAR(100),
  is_vpn_detected BOOLEAN DEFAULT false,
  location_provider VARCHAR(50),
  redeemed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Files to Create:

- components/creator/GeoRestrictionSettings.tsx
- components/admin/GeoRestrictionManager.tsx
- api/geo/validate-location/route.ts
- api/admin/geo-restrictions/route.ts
- lib/utils/geoLocation.ts
- lib/utils/vpnDetection.ts

Code Slice 9 - Part 4: USB Integration & Advanced Admin

Status: Post-MVP

Features:

- 🛡️ **Preloaded USB Integration**
 - API endpoints for USB manufacturers to integrate with Shellff
 - Embedded code validation system for USB devices
 - Offline-to-online synchronization capabilities
 - USB device authentication and security protocols
- 🏢 **USB Partner Management**
 - Partner registration and API key management
 - USB manufacturer onboarding workflow
 - Integration status monitoring and diagnostics

- 📊 **USB Analytics & Monitoring**

- Track USB code usage patterns
- Monitor offline-to-online sync success rates
- USB partner performance metrics

- 🛡️ **Security & Compliance**

- USB device authentication protocols
- Anti-tampering measures for USB codes
- Audit trails for USB-based redemptions

Database Schema:

```
-- USB Integration tables
CREATE TABLE usb_partners (
  id SERIAL PRIMARY KEY,
  partner_name VARCHAR(255) NOT NULL,
  company_name VARCHAR(255) NOT NULL,
  api_key VARCHAR(255) UNIQUE NOT NULL,
  contact_email VARCHAR(255) NOT NULL,
  status VARCHAR(20) DEFAULT 'pending', -- 'pending', 'active', 'suspended'
  integration_type VARCHAR(50) DEFAULT 'standard',
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE usb_devices (
  id SERIAL PRIMARY KEY,
  partner_id INTEGER REFERENCES usb_partners(id),
  device_id VARCHAR(255) UNIQUE NOT NULL,
  device_model VARCHAR(255),
  firmware_version VARCHAR(50),
  embedded_codes TEXT[], -- Array of embedded unlock codes
  sync_status VARCHAR(20) DEFAULT 'offline', -- 'offline', 'syncing', 'synced'
  last_sync_at TIMESTAMP,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE usb_code_redemptions (
  id SERIAL PRIMARY KEY,
  device_id INTEGER REFERENCES usb_devices(id),
  code_id INTEGER REFERENCES unlock_codes(id),
  redemption_method VARCHAR(20) DEFAULT 'usb', -- 'usb', 'online'
  is_offline_redemption BOOLEAN DEFAULT false,
  sync_batch_id VARCHAR(255),
  redeemed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  synced_at TIMESTAMP
);
```

Files to Create:

- `api/partners/usb-integration/route.ts` - Main USB integration API
- `api/partners/usb-integration/validate/route.ts` - USB code validation
- `api/partners/usb-integration/sync/route.ts` - Offline-to-online sync
- `components/admin/USBPartnerManager.tsx` - Admin interface for managing USB partners
- `components/admin/USBIntegrationDashboard.tsx` - Dashboard for USB integration analytics

- `lib/Utils/usbValidation.ts` - USB-specific validation utilities
 - `lib/Utils/offlineSync.ts` - Offline sync management
-

Implementation Timeline

These features will be prioritized and implemented after the MVP launch based on:

1. User feedback and demand
2. Market requirements
3. Technical capacity
4. Strategic priorities

Technical Dependencies

- **Geo Services:** IP geolocation APIs, VPN detection services
 - **USB Integration:** Hardware partner SDKs, device communication protocols
 - **Security:** Enhanced authentication systems, tamper detection
-

This document will be updated as new post-MVP features are identified and prioritized.