

Work Order 13 - Membership Tiers & Lesson Limits

Implementation Complete

Summary

Implemented a clean membership/usage system for CatchBarrels tied to Whop memberships with:

- **4 Subscription Tiers:** Free, Athlete (\$49/mo), Pro (\$99/mo), Elite (\$199/mo)
 - **Assessment Products:** 30-day access with limited sessions
 - **Weekly Session Limits:** 0, 1, 2, or unlimited sessions per week
 - **Dashboard UI:** Shows plan, usage, and upgrade CTAs
 - **Upload Enforcement:** Blocks uploads when weekly cap is reached
-

What Was Built

1. Centralized Configuration (`lib/membership-tiers.ts`)

New File: Single Source of Truth

- **MEMBERSHIP_TIERS** : Complete config for all 4 tiers
- Tier metadata (name, price, icon, color)
- Session limits (per week & monthly estimate)
- Whop product ID mapping
- Feature lists
- Upgrade messages
- **ASSESSMENT_PRODUCTS** : Assessment product configs
- 30-day access windows
- Session allowances
- Pricing details
- **Helper Functions:**
 - `getTierConfig(tier)` : Get full tier details
 - `getTierFromProductId(productId)` : Map Whop product → tier
 - `canStartNewSession(tier, sessionsThisWeek)` : Check if upload allowed
 - `hasTierAccess(userTier, requiredTier)` : Tier hierarchy checks

Membership Tiers Definition:

Tier 1 - Free:

- Price: \$0
- Sessions: 0 per week
- Features: View dashboard, browse drills
- Whop Product IDs: []

Tier 2 - BARRELS Athlete:

- Price: \$49/month
- Sessions: 1 per week (~4 per billing month)
- Features: 1 analyzed upload/week, BARREL Score, basic Coach Rick
- Whop Product IDs: ['prod_kNyobCww4tc2p']

Tier 3 - BARRELS Pro:

- Price: \$99/month
- Sessions: 2 per week (~8 per billing month)
- Features: 2 uploads/week, advanced biomechanics, kinematic sequence, full Coach Rick
- Whop Product IDs: ['prod_04CB6y0IzNJLe']

Tier 4 - BARRELS Elite:

- Price: \$199/month
- Sessions: Unlimited
- Features: Unlimited uploads, assessments, priority support, custom plans
- Whop Product IDs: ['prod_vCV6UQH3K18QZ', 'prod_zH1wnZs0JJKfd']

Assessment Products:

Standard Assessment:

- Product ID: 'prod_assessment_standard_299' (PLACEHOLDER)
- Price: \$299 one-time
- Access: 30 days
- Sessions: 2 during 30-day window

Pro Assessment + S2 Cognition:

- Product ID: 'prod_assessment_pro_399' (PLACEHOLDER)
- Price: \$399 one-time
- Access: 30 days
- Sessions: 2 during 30-day window

2. Updated Existing Files

lib/whop-client.ts

- ✓ Updated `getWhopProductTier()` to use centralized `getTierFromProductId()`
- ✓ Updated `isAssessmentProduct()` to use centralized `isAssessmentProductId()`
- ✓ Removed hardcoded product mappings (now in `membership-tiers.ts`)

app/api/videos/upload/route.ts

- ✓ Imports `canStartNewSession` from `membership-tiers.ts`
- ✓ Already enforces weekly session caps (no changes needed)
- ✓ Returns clear error messages when limit reached

app/dashboard/page.tsx

- ✓ Calculates `sessionsThisWeek` (Monday-Sunday)

- Passes session count to `DashboardClient`
- Includes membership info in props

`app/dashboard/dashboard-client.tsx`

- Imports `MembershipUsageCard` component
 - Renders usage card below “Start New Session” CTA
 - Passes tier, sessions used, status, and expiry date
-

3. New Component: `MembershipUsageCard`

Location: `components/membership-usage-card.tsx`

Features:

- **Plan Display:** Shows tier icon, name, and price
- **Status Badge:** Active/Inactive indicator
- **Weekly Usage:** “X / Y sessions” with progress bar
- **Monthly Estimate:** “~4 per month” for context
- **Color-Coded Progress:**
 - Green: 0-74% used
 - Yellow: 75-99% used
 - Red: 100% used (capped)
- **Warning Messages:** When user hits weekly cap
- **Upgrade CTA:** For free users
- **Assessment Mode:** Special display for 30-day assessment users
 - Shows days remaining
 - Shows sessions used/allowed
 - Warning when <7 days left

UI States:

1. **Free Tier:**
 - Shows “0 sessions” with gray styling
 - Prominent “Upgrade to Start Training” button
2. **Limited Tier (Athlete/Pro):**
 - Shows “X / Y sessions” with progress bar
 - Green when below cap, yellow near cap, red at cap
 - Warning message when capped: “You’ve used all X sessions... Next session unlocks Monday”
 - Upgrade link to next tier
3. **Unlimited Tier (Elite):**
 - Shows “X sessions • Unlimited” in gold
 - No progress bar
 - No cap warnings
4. **Assessment Mode:**
 - Purple theme instead of tier color
 - “30 days left” countdown
 - “Sessions used: X / Y”
 - Warning when <7 days remaining

UI/UX Design

Visual Hierarchy

1. Tier name + icon + price (top)
2. Active/Inactive status badge (top-right)
3. Weekly usage line (“This week: X / Y”)
4. Progress bar (color-coded)
5. Monthly estimate (“~4 per month”)
6. Warning/upgrade messages (when applicable)

Color Coding

- **Free:** Gray (`text-gray-400`)
- **Athlete:** Blue (`text-blue-400`)
- **Pro:** Gold (`text-barrels-gold`)
- **Elite:** Purple (`text-purple-400`)
- **Assessment:** Purple (`text-purple-400`)

Animations

- Card fades in with slight upward motion
- Progress bar animates from 0% to usage % on load
- Buttons have hover/tap scale effects

How It Works

Session Counting Logic

```
// Get start of current week (Monday)
const now = new Date();
const dayOfWeek = now.getDay();
const diff = (dayOfWeek === 0 ? -6 : 1) - dayOfWeek; // Adjust to Monday
const startOfWeek = new Date(now);
startOfWeek.setDate(now.getDate() + diff);
startOfWeek.setHours(0, 0, 0, 0);

// Count sessions uploaded this week
const sessionsThisWeek = await prisma.video.count({
  where: {
    userId,
    uploadDate: { gte: startOfWeek },
  },
});
```

Upload Blocking

```
// In app/api/videos/upload/route.ts
const canStart = canStartNewSession(user.membershipTier, sessionsThisWeek);

if (!canStart.allowed) {
  return NextResponse.json(
    {
      error: 'Session limit reached',
      message: canStart.reason, // Tier-specific message
      sessionsThisWeek,
      membershipTier: user.membershipTier,
    },
    { status: 403 }
  );
}
```

Whop Product Mapping

```
// When syncing from Whop:
for (const membership of memberships) {
  const tier = getTierFromProductId(membership.productId);
  // Update user.membershipTier in database
}
```

Mobile-First Design

- Card is responsive (max-width on desktop)
- Touch-friendly buttons (min-height: 44px)
- Clear typography hierarchy
- Progress bars visible at all screen sizes
- Inline upgrade links vs. full-width buttons on mobile

Testing Checklist

Manual Testing Plan

Test User Setup

Create 5 test users with different tiers:

1. **Free User** (test-free@barrels.com)
 - membershipTier : 'free'
 - membershipStatus : 'inactive'
 - Expected: 0 sessions allowed, upgrade CTA shown
2. **Athlete User** (test-athlete@barrels.com)
 - membershipTier : 'athlete'
 - membershipStatus : 'active'
 - Expected: 1 session/week, cap enforced, upgrade link to Pro

3. **Pro User** (test-pro@barrels.com)
 - membershipTier : 'pro'
 - membershipStatus : 'active'
 - Expected: 2 sessions/week, cap enforced, upgrade link to Elite

4. **Elite User** (test-elite@barrels.com)
 - membershipTier : 'elite'
 - membershipStatus : 'active'
 - Expected: Unlimited sessions, no cap, gold styling

5. **Assessment User** (test-assessment@barrels.com)
 - membershipTier : 'free'
 - assessmentCompletedAt : <30 days ago>
 - assessmentVipExpiresAt :
 - Expected: 30-day window displayed, 2 sessions allowed

Test Scenarios

1. Dashboard Display

- [] Free user sees “Upgrade to Start Training” button
- [] Athlete user sees “0 / 1 sessions” on first login
- [] Pro user sees “0 / 2 sessions” on first login
- [] Elite user sees “0 sessions • Unlimited”
- [] Assessment user sees “30 days left” and “0 / 2 sessions”

2. Session Upload (Athlete, 1/week)

- [] Upload 1 video → Success
- [] Dashboard shows “1 / 1 sessions” with red progress bar
- [] Warning message: “You’ve used all 1 session...”
- [] Attempt 2nd upload → Blocked with error message
- [] Error message suggests upgrading to Pro (\$99/month)

3. Session Upload (Pro, 2/week)

- [] Upload 1 video → Success, “1 / 2 sessions” (green)
- [] Upload 2nd video → Success, “2 / 2 sessions” (red)
- [] Warning message shown
- [] Attempt 3rd upload → Blocked
- [] Error message suggests upgrading to Elite (\$199/month)

4. Session Upload (Elite, unlimited)

- [] Upload 1 video → Success
- [] Upload 2 videos → Success
- [] Upload 5 videos → Success
- [] Dashboard shows “5 sessions • Unlimited” in gold
- [] No warnings or caps

5. Session Upload (Free)

- [] Attempt upload → Blocked immediately
- [] Error message: “You need an active BARRELS membership...”
- [] Suggests upgrading to Athlete (\$49/month)

6. Assessment User (30-day window)

- [] Day 1: Upload 1 video → Success
- [] Dashboard shows “30 days left” and “1 / 2 sessions”
- [] Day 15: Upload 2nd video → Success
- [] Dashboard shows “15 days left” and “2 / 2 sessions”
- [] Day 15: Attempt 3rd upload → Blocked
- [] Day 25: Warning shown (“<7 days left”)
- [] Day 31: Access expired, must subscribe

7. Weekly Reset

- [] Upload sessions on Friday (use up cap)
- [] Wait until next Monday
- [] Dashboard shows “0 / X sessions” (reset)
- [] Upload allowed again

8. Admin/Coach Bypass

- [] Admin user can upload regardless of tier
- [] Coach user can upload regardless of tier
- [] No caps enforced for admin/coach roles



Configuration Guide

How to Update Pricing

File: lib/membership-tiers.ts

```
// Change monthly price:
athlete: {
  price: 59, // Was $49, now $59
  // ...
}
```

How to Change Session Limits

File: lib/membership-tiers.ts

```
// Change weekly sessions:
athlete: {
  sessionsPerWeek: 2, // Was 1, now 2
  sessionsPerMonth: '8 per month', // Update display string
  // ...
}
```

How to Add New Whop Products

File: lib/membership-tiers.ts

```
// Add new product ID to existing tier:
athlete: {
  whopProductIds: [
    'prod_kNyobCww4tc2p',
    'prod_NEW_ATHLETE_ANNUAL', // NEW
  ],
  // ...
}
```

How to Update Assessment Products

File: lib/membership-tiers.ts

Replace placeholder IDs with real Whop product IDs:

```
export const ASSESSMENT_PRODUCTS: AssessmentConfig[] = [
  {
    productId: 'prod_REAL_WHOP_ID_299', // ✓ Replace placeholder
    displayName: 'Standard Assessment',
    price: 299,
    accessDays: 30,
    sessionsAllowed: 2,
    // ...
  },
];
```



Whop Product Mapping Summary

Current Whop Products (from existing codebase):

Whop Product ID	Product Name	CatchBarrels Tier	Sessions/Week
prod_kNyobCww4tc2p	BARRELS Athlete	athlete	1
prod_04CB6y0IzNJLe	BARRELS Pro	pro	2
prod_vCV6UQH3K18QZ	BARRELS Elite (Inner Circle)	elite	Unlimited
prod_zH1wnZs0JKKfd	The 90-Day Transformation	elite	Unlimited

Assessment Products (placeholders):

Whop Product ID	Product Name	Access Window	Sessions
prod_assessment_standard_299	Standard Assessment	30 days	2
prod_assessment_pro_399	Pro Assessment + S2	30 days	2

 **Action Required:** Replace placeholder IDs with real Whop product IDs when available.

Files Changed/Created

New Files:

- ✓ lib/membership-tiers.ts - Centralized tier configuration
- ✓ components/membership-usage-card.tsx - Dashboard usage widget
- ✓ docs/W013_INSPECTION_REPORT.md - Initial inspection findings
- ✓ docs/W013_MEMBERSHIP_TIERS_COMPLETE.md - This document

Modified Files:

- ✓ lib/whop-client.ts - Uses centralized config
- ✓ app/api/videos/upload/route.ts - Imports from membership-tiers.ts
- ✓ app/dashboard/page.tsx - Calculates sessions this week
- ✓ app/dashboard/dashboard-client.tsx - Renders usage card

No Changes Needed:

- ✓ prisma/schema.prisma - Already has all necessary fields
 - ✓ lib/tier-access.ts - Still used for feature gating (complementary)
 - ✓ lib/assessment-vip-config.ts - VIP logic separate from session limits
-

Deployment Status

Build Verification

- ✓ TypeScript compilation: **PASSED**
- ⌚ Next.js build: **PENDING**
- ⌚ Checkpoint saved: **PENDING**

Pre-Deployment Checklist

- ✓ All TypeScript types defined
- ✓ Component imports correct
- ✓ No runtime errors in local dev
- ⌚ Manual testing completed
- ⌚ Screenshot verification

Success Metrics

User Experience

- Users can see their plan and usage at a glance
- Clear messaging when session cap is reached
- Upgrade CTAs are contextual and non-intrusive
- Color-coded progress provides instant feedback

Business Logic

- Session limits enforced accurately
- Weekly reset logic works (Monday-Sunday)
- Admins/coaches bypass all limits
- Assessment 30-day windows tracked

Code Quality

- Single source of truth for tier config
- Easy to update pricing/limits
- Type-safe throughout
- Reusable components

Future Enhancements

Phase 2 (Optional):

1. **Usage History Chart:** Show weekly usage over time
2. **Next Reset Countdown:** “Next session in 3 days”
3. **Rollover Sessions:** Unused sessions carry to next week (configurable)
4. **Session Packs:** One-time purchase of 5 sessions
5. **Family Plans:** Share sessions across multiple users

Admin Features:

1. **Override Limits:** Admin can manually grant extra sessions
2. **Usage Analytics:** Dashboard showing tier distribution and usage rates
3. **A/B Test Pricing:** Test different prices per tier

Support & Maintenance

Common Issues

Q: User reports session cap not resetting on Monday

A: Check server timezone. Weekly reset uses server's Monday 00:00. Verify with:

```
date
timedatectl
```

Q: New Whop product not mapping to tier

A: Add product ID to `lib/membership-tiers.ts` in the appropriate tier's `whopProductIds` array.

Q: Assessment users seeing regular membership UI

A: Ensure `isAssessment` prop is true and assessment dates are set in database.

Debugging Commands

```
-- Check user's tier and session count
SELECT id, email, membershipTier, membershipStatus,
       (SELECT COUNT(*) FROM "Video" WHERE "userId" = u.id
        AND "uploadDate" >= date_trunc('week', NOW())) as sessionsThisWeek
FROM "User" u
WHERE email = 'test@example.com';

-- Reset user's session count (for testing)
DELETE FROM "Video"
WHERE "userId" = 'USER_ID'
AND "uploadDate" >= date_trunc('week', NOW());
```



Final Checklist

- [x] Centralized tier configuration created
- [x] Whop product mapping updated
- [x] Upload API uses new config
- [x] Dashboard displays usage card
- [x] TypeScript compilation passes
- [] Manual testing with 5 test users
- [] Next.js build succeeds
- [] Checkpoint saved
- [] Screenshot documentation



Next Steps

1. **Run Next.js Build:** `yarn build`
2. **Create Test Users:** Use Prisma Studio or admin panel
3. **Manual Testing:** Verify each tier's behavior
4. **Take Screenshots:** Dashboard for each tier
5. **Save Checkpoint:** `build_and_save_nextjs_project_checkpoint`
6. **Deploy:** Test in production

Status: **Implementation Complete**

Pending: Manual testing and deployment