

PRD: Kuji App for CARIS (Carol × Iris)

1. Overview

The Kuji App enables Carol and Iris to host digital lucky draw (“Kuji”) sessions for their fans. Admins (Carol/Iris) can upload prize data, configure draw prices, and track prize stock. Fans can purchase draws (offline payment confirmed by admins) and perform randomized draws with animations. Data persists in local storage (localStorage or SQLite) for small-scale use.

2. Objectives

- Admins upload/manage prize tiers and configure draw pricing.
 - Fans perform draws; prizes are decremented from stock automatically.
 - Data stored locally for offline, small-scale use.
 - Simple, engaging draw experience with animation.
-

3. User Roles

Admin (Carol/Iris)

- Upload/parse CSV prize pool and pricing data.
- Configure pricing presets and bonus draws.
- Start/end sessions, track stock, and export history.

Fan (End User)

- Confirm payment offline.
 - Select preset or custom number of draws.
 - Perform draw(s) with animation and see results.
-

4. Core Features

4.1 Prize Pool Management

- CSV import/export.
- Manual add/edit rows.

- Stock decrements automatically after draws.
- Reporting of exhausted prizes.

4.2 Pricing & Bonus

- Multiple presets with optional bonus draws.
- Example: 10 draws (\$15) → +1 bonus draw.

4.3 Draw System

- Weighted random draw, excluding exhausted prizes.
- Decrement quantities atomically.
- Support bonus draws.

4.4 Storage

- Browser localStorage (default).
- Optional SQLite (mobile/desktop).
- Persist prizes, pricing, settings, and draw history.

5. CSV Templates

Prize Pool (prizes.csv)

tier,prize_name,quantity,weight,sku,notes

A,Scale Figurine - "Starlight",3,1,A-001,Top prize

B,Acrylic Standee - Carol,10,2,B-010,—

B,Acrylic Standee - Iris,10,2,B-011,—

C,Keychain - CARIS Logo,25,3,C-101,—

C,Sticker Pack (5 pcs),30,3,C-201,—

D,Thank You Card (signed),100,5,D-301,Consolation

Pricing (pricing.csv)

preset_id,label,draw_count,price_minor,bonus_draws,active

p1,1 Draw,1,200,0,true

p5,5 Draws,5,800,0,true

p10,10 Draws (Buy 10 Get 1),10,1500,1,true

6. Wireframes

Admin Setup Screen

```
+-----+
| [CARIS Kuji Admin]           Session: ACTIVE ● [Logout] |
+-----+
| [Tabs] Prize Pool | Pricing & Bonus | Settings | Reports      |
+-----+
| Import Prizes CSV | Add Row | Export CSV                      |
| Tier | Prize Name | Qty Rem | Weight | SKU | Notes | Actions (Edit/Delete) |
|-----|
| A   | Scale Figurine | 3 | 1 | A-001 | Top prize | [Edit] [X]          |
...
[Save] [Reset Quantities from CSV]
```

Draw Screen

```
+-----+
| [CARIS Kuji Draws]           Session: ACTIVE ● |
+-----+
| Presets: [1 Draw – $2.00] [5 Draws – $8.00] [10 Draws – $15.00] |
| Or Custom: Draw Count [ __ ] |
+-----+
| Stock Snapshot: A:3 B:20 C:55 D:100 |
+-----+
| [ Start Draw ] [ Clear ] [ History ] |
```

+-----+	
[Animation Area - Kuji Drum/Wheel]	
Results:	
#1 → B Acrylic Standee - Iris	
#2 → C Sticker Pack	
+-----+	

7. React + Tailwind Starter (MVP Design)

Tech Stack

- **React 18+**
- **Tailwind CSS** for UI
- **PapaParse** for CSV import/export
- **localForage** (wrapper for localStorage/IndexedDB)
- **Framer Motion** for draw animations

File Structure

src/

components/

Admin/

PrizePoolManager.jsx

PricingManager.jsx

Settings.jsx

Draw/

DrawScreen.jsx

ResultCard.jsx

hooks/

useLocalStorageDAO.js

utils/

csvUtils.js

randomDraw.js

pages/

Admin.jsx

Draw.jsx

Home.jsx

App.jsx

index.css

Key Components

useLocalStorageDAO.js

- Wrapper around localStorage/localForage.
- Methods: getPrizes(), setPrizes(), getPricing(), setPricing(), saveHistory(), getHistory().

PrizePoolManager.jsx

- Import prizes CSV with PapaParse.
- Table editor for tiers/quantities.
- Export CSV.

PricingManager.jsx

- Manage presets (add/edit/remove).
- Import/export CSV.

DrawScreen.jsx

- Show presets and stock snapshot.
- Animate draw with Framer Motion.
- Deduct prize quantities, save history.

randomDraw.js

- Weighted random algorithm:

- export function drawPrize(prizes) {
 - const weightedPool = prizes.flatMap(p =>
 - Array(p.weight).fill(p)
 -).filter(p => p.quantity > 0);
 -
 - if (!weightedPool.length) return null;
 - const result = weightedPool[Math.floor(Math.random() * weightedPool.length)];
 - result.quantity -= 1;
 - return result;
 - }
-

8. Non-Functional Requirements

- Lightweight, offline-first.
 - Simple UI with responsive Tailwind layout.
 - Draw latency < 1s.
 - Data survives browser reloads.
-

9. Success Metrics

- Admin can upload CSV and configure pricing successfully.
 - Fan draws decrement prize stock accurately.
 - Data persists across reloads.
 - App runs offline with no external dependencies.
-

10. Future Enhancements

- Multi-admin accounts with auth.
- Online payments (Stripe/PayPal).

- Cloud DB sync.
- Reports export to PDF/Excel.
- Leaderboard or fan profiles.