Loyalty Program Outline:
Capturing 2025 Consumer Priorities & Gamification

The revised outline integrates core 2025 consumer priorities: **convenience**, **personalized value**, and **simplicity**, now enhanced with immediate, gamified, and personalized push notifications to ensure the program feels like a seamless part of a member's daily routine (pp. 1-3).

1. Program Architecture Overview

A unified loyalty platform blending local commerce rewards with prediction market engagement, driven by AI analytics to provide sponsors with superior decision-making intelligence while maintaining member privacy (p. 1). The architecture focuses on **ease of use** and a **seamless mobile experience** (p. 1).

- **System Components (Python-based application framing):**
    - **Member Mobile App:** User interface for shopping, betting (points only), recommendations, and **effortless reward tracking**. Now includes immediate push notification functionality for point-earning events and relevant prediction market suggestions (pp. 1-2).
    - **Integration Points:** APIs for POS/QR code systems (transaction capture for **instant gratification**), Kalshi (market odds, order execution), and an ML Engine (predictions) which now triggers the personalized push notifications (pp. 1-2).
    - **Data Processing:** Utilizes a real-time stream (e.g., Apache Kafka or Event Hub) capable of handling 10,000+ events/sec, allowing for **real-time point tracking** and immediate feedback (p. 1).
    - **Backend Systems:**
        - Data Warehouse (PostgreSQL/Snowflake): Used for analytics with fact and dimension tables (p. 1).
        - ML Models & Inference: Runs predictive models for churn, LTV, and **next-best-action (hyper-personalization)**, specifically for in-the-moment recommendations (p. 1).

- Compliance System: Manages privacy controls and responsible gaming safeguards (p. 1).

2. Member Experience & Program Mechanics

The program design emphasizes **engaging members** through a unified currency and a clear, **simple value proposition** that feels convenient and flexible, integrating point earning with immediate gamified opportunities (p. 2).

2.1. Point System (Unified Currency, Points Only)

Points are the sole currency for all activities (no cash wagering), simplifying the user experience and offering "points as currency" **flexibility** (p. 2).

- **Earning & Immediate Gamification:** Members earn points effortlessly from local shopping, daily logins, referrals, and social shares (p. 2). Upon a transaction (point earned), the user immediately receives a personalized push notification (e.g., "You earned 50 points at (Merchant Name)! 🏈 Bet them now on tonight's game.") suggesting relevant, same-day "Predictions you might win" based on AI insights.
- **Redemption Value & Flexibility:** The program structure encourages local spending while providing flexible options, allowing members to spend small point balances (p. 2).
- **Point Expiration:** Points never expire, allowing unlimited holding (p. 2).

2.2. Prediction Market Betting & Gamification

Members can stake points on various outcomes within the app, leveraging **gamification** and **mobile-first interactions** tied directly to real-time events (p. 2).

- **Categories:** Sports, Events, Weather, and Local events via Kalshi API integration, focusing on same-day events to facilitate immediate action from push notifications (p. 2).
- **Mechanism:** Uses the Kalshi API for real-time odds sync, order execution, and automatic settlement (within 60 seconds), providing **instant gratification** for bets placed from the new feature (p. 2).
- **Limits & Compliance:** Enforces age (18+), geolocation, daily/weekly betting limits, and responsible gaming features (p. 2).

2.3. AI-Powered Personalization & Engagement

An AI engine analyzes member behavior to provide targeted recommendations and insights, meeting the demand for **hyper-personalization** in real time (p. 2).

- **Recommendations:** "Predictions you might win," "**Local offers nearby**," and "Friends' insights" tailored to the user's specific shopping behaviors and lifestyles (p. 2). This now includes the immediate, post-purchase/action push notifications.
- **Social Features & Community:** Friend leagues and weekly challenges drive **community engagement** and offer non-transactional rewards (p. 2).

3. Sponsor Benefits & Feedback Loop Narrative

The program aims to drive sponsor revenue gain and improve efficiency through intelligent data use, while ensuring member data privacy (p. 3).

- **Responsible Data Collection and Usage:** Data is processed through a tokenization layer that immediately removes PII, replacing it with an anonymous `member_token` (p. 3).
- **Driving Daily Member Engagement (AI Features):**
    - **Next-Best-Action Engine:** Scores potential actions to determine the single best, **personalized offer** to show a member at any given time, including the new immediate push notifications (p. 3).
    - **Demand Forecasting:** Predicts daily/hourly redemption volume, helping sponsors optimize staffing and inventory (p. 3).
    - **Churn Prediction:** Identifies at-risk members and triggers automated re-engagement offers (p. 3).

4. Technical Implementation Outline (Python Focus)

This section frames key components for a development team, with updates for the new immediate engagement features (p. 4).

4.1. Data Infrastructure & Schemas

The application uses Python libraries (e.g., Pandas, Scikit-learn for ML; Kafka-Python for streaming; SQLAlchemy for DB interaction) (p. 4).

- **Fact Tables:** Store granular events (p. 4).

    - `fct_transactions`: `transaction_id`, `member_token_id`, `merchant_id`, `amount_cents` (p. 4).
    - `fct_predictions`: `prediction_id`, `member_token_id`, `market_id`, `stake_cents`, `outcome_result` (p. 4).
- **Dimension Tables:** Store attributes (p. 4).
    - `dim_members`: `member_token_id`, `age_bracket`, `location_zip`, `data_sharing_status` (p. 4).
- **Stream Processing:** Python scripts using Spark Streaming or Flink for real-time feature engineering and anomaly detection, now also used to identify **immediate push notification triggers** post-transaction (p. 4).

4.2. Kalshi API Integration Framework

The Kalshi platform is a core component for the prediction market mechanics, functioning as the staking, executor, anonymized accounts administrator (for bets), payor, and manager of the predictions component (p. 4).

- **Core Endpoints & Functionality (Python API Client):**
    - `GET /markets`: Browse available contracts by category (`sports`, `events`, etc.) and status (`open`) (p. 4).
    - `POST /orders`: Place a bet (order execution). The platform converts points to dollars internally before sending the amount to Kalshi (p. 5).
    - `GET /positions`: Check a member's current open holdings and unrealized P&L (p. 5).
    - `POST /webhooks/settlement`: Kalshi sends a webhook to the application when a market closes, detailing the winning outcome and payouts (p. 5).
- **Order Flow & Execution (Platform's Role):**

1. **Validate:** The application first checks member age (18+), balance, jurisdiction, and responsible gaming limits locally (p. 5).
2. **Reserve Points:** Points are locked in escrow within the platform's database (p. 5).

3. **Submit to Kalshi:** A Python function converts the point stake into a dollar amount ($) and calls the Kalshi `POST /orders` endpoint (p. 5).

4. **Trigger Notification (New):** Concurrently, the Next-Best-Action Engine is alerted to the completed transaction event via the data stream and triggers an immediate, personalized push notification with a relevant, open prediction market suggestion.

5. **Settlement (Payor/Executor Role):** Upon receiving the settlement webhook from Kalshi, the application credits the member's point balance (payout) and updates their historical record (p. 5).

4.3. Kalshi Webhook Data Schemas (JSON/Python Dictionary)

The following schemas represent the expected data structures for seamless integration, consistent with the Kalshi API documentation structure (p. 5).

- **A. Incoming Settlement Webhook (from Kalshi to Platform):** Kalshi sends a POST request with details like `market_id`, `winning_outcome`, `payout`, etc., using the platform's internal `member_id` token (pp. 5-6).

- **B. Outgoing Acknowledgment Response (from Platform back to Kalshi):** The platform responds with a simple JSON object confirming receipt and the amount to credit (p. 6).

4.4. AI Insights Translation to Sponsor Portal Features

The AI Analytics Engine models are consumed by the Sponsor Intelligence Portal (Dashboard) and translated into actionable features that drive business outcomes (p. 6).

- **Member LTV Prediction Model:** Portal Feature: "**High-Value Members**" Dashboard Segment (p. 6).

- **Churn Prediction Model:** Portal Feature: "**At-Risk Members**" Alerts and Intervention Plans (p. 7).

- **Next-Best-Action Engine:** Portal Feature: "**AI Recommendations**" Card on Home Screen, now also driving the real-time push notifications (p. 7).

- **Demand Forecasting Model:** Portal Feature: "**Forecasted Demand**" in Metrics and Staffing Suggestions (p. 7).