LoyalNest Testing Strategy Document

1. Overview

This document defines the testing strategy for the LoyalNest Shopify app, ensuring robust functionality, performance, accessibility (WCAG 2.1 AA, Lighthouse score 90+), and security for the Merchant Dashboard, Customer Widget, Admin Module, and backend integrations. The app supports 5,000+ merchants and 50,000+ customers (Shopify Plus), with GDPR/CCPA compliance and multilingual support (`en` , `es` , `fr` , `ar` via i18next). The testing strategy covers Phases 1–3, aligning with `user_stories.md` , `wireframes.txt` , and `ui_ux_requirements.md` .

2. Testing Objectives

- **Functionality**: Validate all user stories (US-CW1–CW15, US-MD1–MD18, US-AM1–AM13, US-BI1–BI5) against acceptance criteria.
- **Performance**: Ensure <2.5s LCP, <100ms FID, <0.1 CLS, and handle 1,000 orders/hour (Plus-scale: 50,000+ customers).
- **Accessibility**: Achieve WCAG 2.1 AA compliance, with ARIA labels and 90+ Lighthouse accessibility score.
- **Security**: Validate Shopify OAuth, HMAC, AES-256 encryption, and GDPR/CCPA compliance (90-day data retention).
- **Scalability**: Support 5,000+ merchants with Redis caching and PostgreSQL partitioning (`merchant_id`).
- **Localization**: Verify UI and notifications in `en` , `es` , `fr` , `ar` with fallback to `en` for unsupported locales.

3. Testing Types and Tools

3.1 Unit Testing

- **Purpose**: Validate individual React components, gRPC services, and backend logic.
- **Tools**: Jest (React), Jest-Node (Node.js), ts-jest (TypeScript).
- **Scope**:
  - **Frontend**: Test components (e.g., `PointsBalance` , `ReferralPopup` , `RFMWizard` , `BadgesSection`) for rendering, state changes, and event handlers (e.g., badge animations, US-CW11).
  - **Backend**: Test gRPC services (`/points.v1/*` , `/referrals.v1/*` , `/admin.v1/*`) for request/response validation and error handling (400, 429, 500).

- **Examples**:
  - US-CW1: Jest test for `PointsBalance` rendering "500 Stars" with WebSocket updates (Polaris `Badge`).
  - US-MD11: Jest test for `RFMWizard` validation (recency <= 360 days).
  - US-CW11: Jest test for badge animations in `BadgesSection` (300ms scale effect, aria-label: "View badge details").
- **Coverage**: 90%+ for critical paths (points, referrals, RFM, gamification).
- **Execution**: Run on every PR via GitHub Actions (`jest --coverage`).

3.2 Integration Testing

- **Purpose**: Validate interactions between frontend (React), backend (gRPC), and external APIs (Shopify, Klaviyo, Twilio, Square).
- **Tools**: Jest, Supertest (REST/gRPC), Mock Service Worker (MSW) for API mocks, `ws` for WebSocket mocks.
- **Scope**:
  - Test API flows (e.g., `POST /v1/api/points/earn`, `/points.v1/EarnPoints` with Shopify OAuth).
  - Validate Redis caching (`points:{customer_id}`, `rfm:preview:{merchant_id}`).
  - Test WebSocket streams (`/points.v1/PointsStream`, `/admin/v1/imports/stream`) for points updates (US-CW1) and import progress (US-MD4).
  - Test webhook handling (e.g., Shopify `orders/create`, HMAC validation).
  - Test edge cases: Klaviyo/Postscript timeouts (5s retry, US-CW4, PostHog: `referral_failed`), invalid referral codes (400, "Invalid code", US-CW4, PostHog: `referral_error`), GDPR webhook retries (`/v1/customers/redact`, 3 retries, US-AM6, PostHog: `gdpr_retry_failed`).
  - Test error handling for 400 (invalid input), 429 (rate limit, retry after 1s), and 500 (server error, exponential backoff: 100ms–1s).
  - **Examples**:
    - US-BI1: Supertest for `POST /v1/api/orders/points` inserting into `points_transactions`.
    - US-CW4: MSW mock for Twilio/Klaviyo in referral sharing, test 5s timeout retry.
    - US-MD4: Supertest for customer imports (`/admin/customers/import`, PostHog: `import_failed`).
    - US-AM12: Supertest for RFM exports (`/admin/v1/rfm/export`, PostHog: `rfm_export_failed`).
  - **Coverage**: 85%+ for API endpoints, database interactions, and WebSocket streams.
- **Execution**: Run nightly via GitHub Actions (`jest --integration`).

3.3 End-to-End (E2E) Testing

- **Purpose**: Simulate user flows across Merchant Dashboard, Customer Widget, and Admin Module.
- **Tools**: Cypress, Playwright (backup for cross-browser).
- **Scope**:
  - Test critical flows: onboarding (US-MD1), points earning/redemption (US-CW2, US-CW3), referral sharing (US-CW4), GDPR requests (US-CW8, US-AM6).
  - Validate UI interactions (Polaris `Banner`, `Modal`, `DataTable`) and PostHog events (e.g., `points_earned`, `referral_status_viewed`).
  - Test mobile layouts (Tailwind `sm: 320px, md: 768px`) with swipe gestures for referrals (US-CW7) and leaderboards (US-CW12).
  - Test real-time updates: points balance (US-CW1, WebSocket `/points.v1/PointsStream`), import progress (US-MD4, WebSocket `/admin/v1/imports/stream`), onboarding checklist (US-MD1, WebSocket `/admin/v1/setup/stream`).
  - Test error handling: 400 (invalid input), 429 (rate limit, retry after 1s), 500 (server error, exponential backoff: 100ms–1s).
  - **Examples**:
    - US-CW1: Cypress test for points balance display with real-time updates (Polaris `Badge`, aria-live: "Points balance updated", PostHog: `points_stream_updated`).
    - US-MD4: Cypress test for customer search with pagination (50 rows) and import errors (400: "Missing email column", PostHog: `import_failed`).
    - US-MD1: Cypress test for onboarding checklist with real-time progress (WebSocket `/admin/v1/setup/stream`, aria-label: "View checklist progress", PostHog: `setup_progress_viewed`) and skipped task `Banner` (aria-label: "Resolve skipped task", PostHog: `skipped_task_viewed`).
    - US-CW7: Cypress test for referral status with swipe gestures (aria-label: "Swipe to view referrals", PostHog: `referral_status_viewed`).
    - US-CW12: Cypress test for paginated leaderboard (50 ranks/page, aria-label: "View leaderboard page", PostHog: `leaderboard_page_viewed`).
    - US-MD12: Cypress test for RFM heatmaps and line charts (aria-label: "View RFM heatmap", PostHog: `rfm_heatmap_viewed`).
    - US-CW4: Cypress test for Klaviyo/Postscript timeout handling (5s retry, Polaris `Banner`, PostHog: `referral_failed`).
    - US-AM6: Cypress test for GDPR webhook retries (3 retries, PostHog: `gdpr_retry_failed`).
  - **Coverage**: 100% for critical user flows (Phase 1: US-CW1–CW8, US-MD1–MD8).
- **Execution**: Run on PR merges and daily (`cypress run --browser chrome`).

3.4 Performance Testing

- **Purpose**: Ensure scalability and responsiveness under load.
- **Tools**: k6, Lighthouse CI.
- **Scope**:
  - **Load Testing**: Simulate 1,000 orders/hour (US-BI1), 50,000+ customer records (US-MD4, US-BI3), and 100 concurrent referral shares (US-CW4).
  - **Frontend Performance**: Validate LCP <2.5s, FID <100ms, CLS <0.1 using Lighthouse CI.
  - **Backend Performance**: Test gRPC endpoints (`/points.v1/*` , `/analytics.v1/*`) for <1s response time with Redis and PostgreSQL partitioning.
  - **Examples**:
    - US-BI1: k6 test for 1,000 orders/hour with `points_transactions` partitioning.
    - US-CW14: Lighthouse CI for sticky bar (Tailwind `sm: hidden, md: block`).
    - US-AM12: k6 test for RFM exports with 50,000+ customers (US-AM12).
  - **Metrics**: 95%+ requests <1s, error rate <0.1%.
- **Execution**: Weekly k6 runs, Lighthouse CI on PRs (`lighthouse-ci --score 90`).

3.5 Accessibility Testing

- **Purpose**: Ensure WCAG 2.1 AA compliance and screen reader support.
- **Tools**: Lighthouse CI, axe-core, VoiceOver (macOS), NVDA (Windows).
- **Scope**:
  - Test ARIA labels (e.g., `aria-label="Redeem points"` in US-CW3, `aria-label="Search customers"` in US-MD4).
  - Validate keyboard navigation (Polaris `Modal` , `DataTable`) and swipe gestures (US-CW7, US-CW12).
  - Ensure 4.5:1 contrast ratio and `aria-live="polite"` for dynamic updates (e.g., US-CW1 points balance, US-MD12 RFM charts).
  - **Examples**:
    - US-CW8: axe-core test for GDPR form accessibility in Polaris `Modal` .
    - US-MD5: Lighthouse CI for RFM heatmap and line charts (Chart.js, `aria-live="Chart data available"`).
    - US-CW12: VoiceOver/NVDA test for leaderboard pagination (aria-label: "View leaderboard page").
  - **Metrics**: Lighthouse accessibility score 90+, zero critical axe violations.

- **Execution**: Run axe-core with Jest, Lighthouse CI on PRs (`lighthouse-ci --accessibility 90`).

3.6 Security Testing

- **Purpose**: Validate data encryption, authentication, and GDPR/CCPA compliance.
- **Tools**: OWASP ZAP, Snyk, custom scripts for GDPR checks.
- **Scope**:
  - Test Shopify OAuth, HMAC validation (US-BI1), and AES-256 encryption (`customers.email` , `reward_redemptions.discount_code`).
  - Validate RBAC (`merchants.staff_roles` , `admin_users.metadata`) for restricted endpoints (US-MD1, US-AM4).
  - Ensure 90-day data retention for GDPR requests (US-CW8, US-AM6) with webhook retries (3 retries).
  - **Examples**:
    - US-AM6: OWASP ZAP test for GDPR webhook (`/v1/customers/redact` , 3 retries, PostHog: `gdpr_retry_failed`).
    - US-MD4: Snyk test for SQL injection in customer search (`pgcrypto` decryption).
  - **Metrics**: Zero high-severity vulnerabilities, 100% GDPR compliance.
- **Execution**: OWASP ZAP weekly, Snyk on PRs (`snyk test`).

3.7 Localization Testing

- **Purpose**: Verify UI and notifications in `en` , `es` , `fr` , `ar` .
- **Tools**: Cypress, i18next-parser.
- **Scope**:
  - Test UI rendering with i18next for `en` , `es` , `fr` , `ar` (Phase 3, US-CW13). Validate fallback to English for unsupported locales or malformed JSONB (`email_templates.body->>'en'` , `program_settings.settings->>'en'`).
  - Validate notification templates (`email_templates.body` , JSONB, `CHECK ?| ARRAY['en', 'es', 'fr', 'ar']`) for referrals (US-CW4, US-MD8).
  - **Examples**:
    - US-CW13: Cypress test for language switch in `LanguageSelector` with fallback to `en` (PostHog: `language_fallback_triggered`).
    - US-MD8: i18next-parser validation for notification template translations (PostHog: `template_fallback_triggered`).
- **Execution**: Run on PRs (`cypress run --spec localization`).

4. Testing Scope by Phase

Phase 1: Core Features

- **User Stories**: US-CW1–CW8, US-MD1–MD8, US-AM1–AM6, US-BI1–BI2.
- **Focus**: Points earning/redemption, referral sharing, onboarding, basic analytics, GDPR requests.
- **Tests**:
  - Unit: Jest for `PointsBalance`, `ReferralPopup`, `WelcomePage` (90% coverage).
  - Integration: Supertest for `/points.v1/EarnPoints`, `/referrals.v1/CreateReferral`, WebSocket `/points.v1/PointsStream`.
  - E2E: Cypress for points display (US-CW1, WebSocket `/points.v1/PointsStream`), onboarding flow with real-time progress (US-MD1, WebSocket `/admin/v1/setup/stream`), GDPR form (US-CW8).
  - Performance: k6 for 1,000 orders/hour (US-BI1), Lighthouse CI for LCP <2.5s.
  - Accessibility: axe-core for ARIA labels, Lighthouse CI for Polaris components.
  - Security: OWASP ZAP for Shopify OAuth and HMAC.

Phase 2: Enhanced Features

- **User Stories**: US-CW9–CW10, US-CW14–CW15, US-MD9–MD13, US-AM7–AM11, US-BI3.
- **Focus**: VIP tiers, RFM nudges, sticky bar, post-purchase widget, checkout extensions.
- **Tests**:
  - Unit: Jest for `VIPTier`, `NudgeBanner`, `StickyBar`, `PostPurchaseWidget`.
  - Integration: Supertest for `/analytics.v1/GetNudges`, `/frontend.v1/GetWidgetConfig`, Klaviyo/Postscript timeouts (5s retry).
  - E2E: Cypress for VIP tier display (US-CW9), RFM config (US-MD11), sticky bar click (US-CW14), import errors (US-MD4, PostHog: `import_failed`).
  - Performance: k6 for 100 concurrent shares (US-CW4), Lighthouse CI for sticky bar (CLS <0.1).
  - Accessibility: axe-core for `RangeSlider` in RFM wizard (US-MD11).
  - Security: Snyk for Klaviyo/Postscript integrations (US-CW4).

Phase 3: Advanced Features

- **User Stories**: US-CW11–CW13, US-MD14–MD18, US-AM12–AM13, US-BI4–BI5.

- **Focus**: Gamification, bonus campaigns, RFM exports, advanced integrations.
- **Tests**:
  - Unit: Jest for `BadgesSection` (300ms scale animation), `CampaignManagement`, `RFMSegmentExport`.
  - Integration: Supertest for `/analytics.v1/AwardBadge`, `/points.v1/CreateCampaignService` (Shopify Discounts API), `/admin/v1/integrations/square`, `/admin/v1/settings` (multi-currency).
  - E2E: Cypress for badge earning (US-CW11, 300ms animation), campaign creation (US-MD14, Shopify Discounts API), RFM export (US-AM12, PostHog: `rfm_export_failed`), multi-currency toggle (US-MD6), Square integration (US-AM13).
  - Performance: k6 for 50,000+ customer exports (US-AM12), Lighthouse CI for leaderboard (US-CW12).
  - Accessibility: axe-core for Chart.js in leaderboard (US-CW12) and RFM heatmaps (US-MD12).
  - Security: OWASP ZAP for Square integration (US-AM13).

5. Test Environment

- **Staging**: Dockerized setup (PostgreSQL, Redis, gRPC services) on AWS ECS, mimicking production.
- **Mock Data**: 5,000 merchants, 50,000 customers (Plus-scale), generated via Faker.js.
- **API Mocks**: MSW for Shopify, Klaviyo, Twilio, Square APIs.
- **Browsers**: Chrome, Firefox, Safari (Playwright for cross-browser E2E).
- **Devices**: Desktop (1024px), mobile (320px, Tailwind `sm` breakpoint).

6. Test Execution Plan

- **PR Testing**: Run Jest, axe-core, Lighthouse CI, and select Cypress tests (`cypress run --spec critical`).
- **Nightly Builds**: Full integration and E2E suites (`jest --integration`, `cypress run`).
- **Weekly Performance**: k6 for load testing, OWASP ZAP for security scans.
- **Pre-Release**: Full regression suite (Cypress, k6, Lighthouse CI) before deployment.
- **Monitoring**: PostHog for UI events (e.g., `sticky_bar_clicked`, `rfm_exported`, `points_stream_updated`), Sentry for runtime errors.

7. Success Metrics

- **Functional**: 100% critical user story coverage (Phase 1), 85%+ for Phases 2–3.
- **Performance**: 95%+ requests <1s, LCP <2.5s, FID <100ms, CLS <0.1.
- **Accessibility**: Lighthouse score 90+, zero critical axe violations.
- **Security**: Zero high-severity vulnerabilities, 100% GDPR/CCPA compliance.
- **Defect Rate**: <0.5% critical bugs in production (tracked via Sentry).

8. Risks and Mitigation

- **Risk**: Shopify API rate limits (50 points/s, 100 points/s Plus).
  - **Mitigation**: Implement exponential backoff (3 retries, 500ms base delay), test with k6 for 429 errors.
- **Risk**: Localization errors in `es`, `fr`, `ar`.
  - **Mitigation**: Use i18next-parser, test with Cypress for all locales, validate fallback to `en`.
- **Risk**: Scalability for 50,000+ customers.
  - **Mitigation**: Partition `points_transactions`, `referrals` by `merchant_id`, cache in Redis, test with k6.
- **Risk**: Accessibility violations.
  - **Mitigation**: Run axe-core and Lighthouse CI on PRs, manual VoiceOver/NVDA testing.
- **Risk**: Integration failures (Klaviyo/Postscript, Square).
  - **Mitigation**: Test timeouts (5s retry) and API health with Supertest, OWASP ZAP.

9. Roles and Responsibilities

- **Developers**: Write unit and integration tests (Jest, Supertest).
- **QA Engineers**: Author E2E tests (Cypress), perform manual accessibility testing.
- **DevOps**: Configure GitHub Actions, k6, and OWASP ZAP pipelines.
- **Product Owner**: Validate acceptance criteria against test results.

10. Deliverables

- Test suites in `/tests` (Jest, Cypress, k6 scripts).
- Test reports in GitHub Actions artifacts (`coverage.html` , `lighthouse-report.json`).
- PostHog dashboards for UI event tracking (e.g., `sticky_bar_clicked` , `rfm_exported` , `points_stream_updated`).
- Sentry integration for production error monitoring.