ChatGPT

Continental Smart Venue & Stadium Management Checklist

Modern sports venues are evolving into **smart stadiums** by integrating advanced technologies to offer personalized fan experiences [1] . A robust management platform must handle comprehensive event booking and ticketing (with digital tickets and online registration [2]), as well as CRM and promotions. For example, promo and discount codes are known to drive early registrations and boost sales [3] . The checklist below outlines key modules, extended features, recommended tech stack, and phased implementation steps for a continental-scale venue management system.

Core Modules and Features

<details><summary>Booking & Event Management</summary>

- [ ] Define and manage events (title, description, schedule, venue, capacity).
- [ ] Provide public event listings with search/filter (by date, category, location).
- [ ] Online registration and ticket purchase workflow; deliver digital tickets/receipts [2] .
- [ ] Support seat selection and holds (auto-expiring reservations) for reserved seating.
- [ ] Manage event lifecycle (draft → publish → archive) and send automated reminders/invitations [4] .

</details>

<details><summary>Venue Management</summary>

- [ ] Maintain venue records (name, address, capacity, facilities) and sub-venues/sections.

• [ ] Handle floor plans and seating charts: design layouts, allocate seats or standing zones [5] .
• [ ] Track venue availability and maintenance schedules.
• [ ] Define venue rules (age restrictions, accessibility, pricing tiers).
• [ ] Attach multimedia (maps, photos) and documents (venue rental contracts) to venue profiles.

</details>

<details><summary>Ticketing (General, Reserved, Season & Group)</summary>

• [ ] Issue general admission and reserved seats tickets; generate unique QR/barcodes for each ticket.
• [ ] Flexible ticket options: single-entry, multi-use (season passes), group packages [6] .
• [ ] Enable multi-visit or unlimited season passes, and bundle tickets into group discounts [6] .
• [ ] Dynamic pricing (e.g. peak/off-peak rates, early-bird discounts) to maximize revenue [7] .
• [ ] Support mobile wallets (Apple/Google Wallet) and send tickets via email/SMS.

</details>

<details><summary>User Roles & Permissions</summary>

• [ ] Define user roles (Admin, Venue Manager, Promoter/Organizer, Agent/Reseller, Staff, Public User, Security, etc.).
• [ ] Implement role-based access control (RBAC) so each role sees only their scope of data and actions [8] .
• [ ] Admin UI for managing users and assigning roles/permissions.
• [ ] Apply the principle of least privilege (roles get minimum permissions needed) [8] .
• [ ] Audit logs for privileged actions (user creation, permission changes).

</details>

<details><summary>Payment Integrations</summary>

• [ ] Integrate African mobile money (EcoCash, OneMoney) via Paynow or similar gateway [9] .
• [ ] Integrate M-Pesa (Kenya/Uganda) – M-Pesa integration is essential in East Africa [10] .
• [ ] Support card and wallet payments via Paystack, Flutterwave, Stripe (USD) and local switches.
• [ ] Multi-currency support (accept and settle in ZWL, USD, ZAR, etc. with FX conversion).
• [ ] Handle payment flows: invoicing, receipts, refunds, and optional offline (cash/cheque) orders.
• [ ] Ensure PCI compliance, SSL/TLS, and secure tokenization of payment data.

</details>

<details><summary>Digital Event / Streaming (Pay-per-View)</summary>

• [ ] Support live or on-demand streaming for virtual events.
• [ ] Generate expiring secure stream URLs for paid attendees.
• [ ] Implement DRM (e.g. Widevine, PlayReady, FairPlay) and watermarking to protect content [11] .
• [ ] Track viewer logins and duration for analytics.
• [ ] Integrate with CDNs or streaming platforms (YouTube Live, Vimeo OTT, etc.).

</details>

<details><summary>QR & Biometric Check-In</summary>

- [ ] Provide mobile/web check-in apps (with camera) and hardware scanners for QR/barcode validation.
- [ ] Mark tickets as used upon scan and log entry times.
- [ ] (Advanced) Integrate biometric verification (fingerprint/facial) for VIP or staff access.
- [ ] View check-in dashboard with real-time totals and gate-by-gate reports.
- [ ] Support group check-in (auto-increment multi-use ticket counts) [12] .

</details>

<details><summary>Promotions & Discounts</summary>

- [ ] Create customizable promotion codes and discount campaigns [3] .
- [ ] Flexible discount types: percentage off, flat rate, or bundled deals.
- [ ] Targeted use: apply codes to specific events, groups, or all events.
- [ ] Schedule promotions (early-bird, limited-time offers) and cap usage or validity dates.
- [ ] Display promo banners/links and track code redemptions (for ROI analysis).

</details>

<details><summary>Staff & Resource Scheduling</summary>

- [ ] Assign staff/volunteers to events and roles (ushering, ticketing, security, etc.) [13] .
- [ ] Calendar and shift scheduling with drag-and-drop interface.
- [ ] Track employee availability, time-off requests, and time-clock (for payroll) [13] .
- [ ] Set automated schedule reminders and conflict alerts [14] .
- [ ] Mobile access: staff can view schedules, accept/decline shifts, and complete task checklists.

</details>

<details><summary>Contracts & E-Signatures</summary>

- [ ] Store and manage contract templates (venue rental, artist, sponsorship) within the system.
- [ ] Integrate e-sign providers (DocuSign, Adobe Sign, Sertifi, etc.) to send and sign documents electronically.
- [ ] Automate combined contract & payment requests (one step) [15] .
- [ ] Store signed PDFs and track signature status with reminders for pending agreements.
- [ ] Maintain an audit trail for contracts (who signed and when).

</details>

<details><summary>CRM-lite (Contacts & Promoters)</summary>

- [ ] Maintain a contact database of attendees, promoters, partners, and leads.
- [ ] Track interactions and notes on each contact/promotion deal.

• [ ] Allow segmenting (VIPs, first-time, high-spenders) for targeted outreach.
• [ ] Built-in email/SMS newsletter to contact lists (integrated with ticket data) [16] .
• [ ] Affiliate/promoter portal for referral links and commission tracking.

</details>

<details><summary>Analytics & Reporting Dashboards</summary>

• [ ] Real-time dashboards for sales, revenue, and attendance by event/venue.
• [ ] Charts for ticket sales over time, capacity utilization, and event popularity trends.
• [ ] Exportable reports (CSV/PDF) of transactions, customer lists, and promo performance.
• [ ] Attendance heatmaps or flow analyses for crowd management (if IoT enabled).
• [ ] Integrate BI tools or use chart libraries (Recharts, Chart.js) for visualization.

</details>

<details><summary>Notifications (Email, SMS, WhatsApp)</summary>

• [ ] Email notifications for booking confirmations, receipts, and reminders.
• [ ] SMS gateway (Twilio, Africa's Mobiz, etc.) for urgent alerts and two-factor OTPs.
• [ ] WhatsApp API notifications (via Twilio or similar) for booking updates and marketing [17] .
• [ ] Customizable templates with personalization (name, event details).
• [ ] Manage opt-in preferences and unsubscribe links.

</details>

<details><summary>POS / Kiosk / Agent Portal</summary>

• [ ] On-site POS solutions: cashier desks, self-service kiosks, and mobile POS devices [18] .
• [ ] Offline-capable: continue ticket sales and payment processing even without internet [18] .
• [ ] Agent/reseller portal: login for authorized sellers or affiliates, with commission tracking.
• [ ] Group check-in and booking modifications directly via POS terminal [19] .
• [ ] Custom user roles and permissions for POS staff (cashier, manager, etc.) [19] .

</details>

<details><summary>PWA Public Booking App</summary>

• [ ] Responsive Progressive Web App for customer booking (React PWA or similar).
• [ ] Offline support via service worker: cache event data and basic UI for low-connectivity areas [20] .
• [ ] Fast loading and installable (home screen) for repeat users.
• [ ] Push notifications for upcoming events or promotions.
• [ ] Option for a companion React Native mobile app if deep device integration needed.

</details>

Extended Capabilities

<details><summary>Multi-currency & FX Support</summary>

• [ ] Display and accept payments in multiple currencies (ZWL, USD, EUR, etc.).
• [ ] Real-time FX conversion (at checkout/showing local-currency pricing).
• [ ] Reconcile multi-currency transactions: settle and report in base currency.
• [ ] Regulatory compliance: VAT/GST handling per region.

</details>

<details><summary>Sales Channel Tagging</summary>

• [ ] Tag each sale by source/channel (website, app, reseller, physical POS, campaign).
• [ ] Filter sales data by channel for reporting (to analyze ROI of marketing channels).
• [ ] Support affiliate coupon codes or links for channel tracking.

</details>

<details><summary>Streaming DRM</summary>

• [ ] Apply Digital Rights Management (DRM) encryption to all livestreams [11] .
• [ ] Use multi-DRM to cover all devices (Widevine, PlayReady, FairPlay).
• [ ] Watermark video streams with user identifiers to deter piracy.
• [ ] Ensure the streaming platform/player checks licenses before playback.

</details>

<details><summary>Offline Access (Rural Staff/Agents)</summary>

• [ ] Enable offline data entry and sync: allow staff to take bookings or check-in attendees without network.
• [ ] Progressive Web App functionality: cache critical pages for offline browsing [20] .
• [ ] Example: EventSprout's app can sell and scan tickets completely offline [21] .
• [ ] USSD or SMS booking options for areas without smartphones (optional integration).
• [ ] Local kiosk software that syncs when reconnected.

</details>

Technical Stack Recommendations

<details><summary>PostgreSQL Database</summary>

• [ ] Use PostgreSQL for relational data (events, tickets, users, transactions).
• [ ] Design a normalized schema: e.g. tables for Users, Venues, Events, Tickets, Sales, etc.
• [ ] Index key fields (event dates, email) for performance.
• [ ] Use JSONB fields for flexible configs (e.g. event metadata, audit logs).

   • [ ] Consider PostGIS for spatial queries (seat maps, geolocation).

</details>

<details><summary>Node.js / Express Backend</summary>

   • [ ] Build RESTful (or GraphQL) APIs using Node.js with Express or a framework (NestJS, Fastify).
   • [ ] Implement authentication (JWT/OAuth2) and middleware for RBAC [8] .
   • [ ] Structure by module: separate controllers/services for Events, Tickets, Payments, etc.
   • [ ] Integrate third-party SDKs (Twilio, Paystack, DocuSign, etc.).
   • [ ] Write unit and integration tests (Jest, Mocha) for all endpoints.

</details>

<details><summary>React Admin UI</summary>

   • [ ] Use React.js (with an admin template or library) for the admin dashboard.
   • [ ] Components for CRUD operations on events, venues, users, and promotions.
   • [ ] Use Material-UI, TailwindCSS, or similar for a modern interface.
   • [ ] Role-aware UI: hide features not permitted to the logged-in user.
   • [ ] Charts and tables (Recharts, Chart.js) for analytics widgets.

</details>

<details><summary>PWA Booking Portal (React)</summary>

   • [ ] Build the public-facing portal as a React PWA.
   • [ ] Service workers to cache assets and enable offline browsing [20] .
   • [ ] Mobile-responsive design for bookings and ticket management.
   • [ ] If demand grows, consider a React Native app for native performance.
   • [ ] Integrate push notifications and deep linking for repeat engagement.

</details>

Implementation Phases

<details><summary>Phase 1: Database Schema & Seed Data</summary>

   • [ ] Design and create PostgreSQL schema (tables for users, events, tickets, venues, sales, etc.).
   • [ ] Develop seed scripts with sample venues, events, and user accounts for testing.
   • [ ] Set up database migrations (using Sequelize, Knex, or TypeORM).
   • [ ] Document schema (ER diagrams, field descriptions).

</details>

<details><summary>Phase 2: API Development (By Module)</summary>

   • [ ] Implement core APIs: Events, Tickets, Venue, User management, Reports.

- [ ] Develop Booking/Event logic (reservation holds, availability checks).
- [ ] Build Ticketing API: issue tickets, validate/check-in endpoints.
- [ ] Integrate Payment APIs (EcoCash/OneMoney via Paynow, Mpesa, Stripe/Paystack).
- [ ] Enforce RBAC in all endpoints (check user role before actions).
- [ ] Write automated tests for each module.

</details>

<details><summary>Phase 3: Admin UI Setup</summary>

- [ ] Scaffold React Admin interface; implement login/auth flow.
- [ ] Build pages for managing events, venues, users, tickets, promotions.
- [ ] Create dashboards: sales summary, upcoming events, and pending tasks.
- [ ] Form flows for creating/editing modules (venue floorplans, schedules).
- [ ] Connect to backend APIs and handle validation/errors.

</details>

<details><summary>Phase 4: Public PWA Build</summary>

- [ ] Develop user-facing React PWA for browsing events and booking tickets.
- [ ] Implement offline caching (service worker) for event listings and basic assets [20].
- [ ] Build checkout flow: seat selection, payment, ticket delivery (email/SMS).
- [ ] Responsive/mobile-first design; test on various devices.
- [ ] (Optional) Develop React Native mobile app with the same APIs.

</details>

<details><summary>Phase 5: Integrations & Testing</summary>

- [ ] Integrate external services: payment gateways (test mode), SMS/WhatsApp (Twilio) [17], email (SMTP).
- [ ] Implement e-signature flows for contracts (e.g. DocuSign/Adobe Sign).
- [ ] Add streaming/video delivery for digital events and enforce DRM [11].
- [ ] Perform security review: SSL, auth, input validation, rate limiting.
- [ ] Conduct performance/load testing on APIs and streaming.
- [ ] QA all features: booking, check-in, offline mode (simulate no Wi-Fi) [21].

</details>

<details><summary>Phase 6: Deployment & Documentation</summary>

- [ ] Containerize services (Docker) and configure cloud hosting (AWS, DigitalOcean, or local).
- [ ] Set up CI/CD pipelines (Github Actions/Gitlab CI) for auto-build and deployment.
- [ ] Write API documentation (Swagger/OpenAPI) and user/admin manuals.
- [ ] Provide training for staff and run a pilot event.
- [ ] Establish monitoring (logging, metrics) and backup procedures.

</details>

Sources: Best practices and features are compiled from industry resources on event/venue management software [2] [3] [17] [21] . Each checklist item reflects capabilities seen in leading ticketing and stadium platforms.

---

[1] Revolutionizing the Sports and Venues Experience: A Smart Stadium Layout | Extreme Networks

https://www.extremenetworks.com/resources/infographic/revolutionizing-the-stadium-experience-a-smart-stadium-layout

[2] [4] [5] Event Management Software Features Checklist

https://theonetechnologies.com/blog/post/event-management-software-features-checklist

[3] WordPress Discount and Promotion Codes | Event Espresso

https://eventespresso.com/addons/wordpress-discount-promotion-codes?lang=en-US

[6] [12] [16] Season & Event Passes - TicketSignup

https://info.ticketsignup.io/use-cases/season-event-passes/

[7] Freshtix - Online Ticket Sales, Box Office and Ticketing Software

https://www.freshtix.com/blog/maximizing-revenue-with-dynamic-ticket-pricing-strategies

[8] Role-Based Access Control

https://auth0.com/docs/manage-users/access-control/rbac

[9] Initiate a mobile money transaction · Paynow Developer Hub

https://developers.paynow.co.zw/docs/initiate_mobile_transaction.html

[10] Integrating M-Pesa STK Push with Node.js: A Comprehensive Guide | by Faith Kariuki | Medium

https://medium.com/@techtinkerbell/integrating-m-pesa-stk-push-with-node-js-a-comprehensive-guide-d75323c679fd

[11] How to Protect Video Content with Digital Rights Management (DRM)

https://www.brightcove.com/tech-talk/dealing-drm-understanding-drm-and-how-produce-protected-content/

[13] [14] Event Employee Scheduling System | Releventful

https://www.releventful.com/features/event-employee-scheduling-system

[15] e-Signature Payment Software for Entertainment and Restaurant Venues | Sertifi

https://corp.sertifi.com/solutions/entertainment-venues/

[17] Twilio/SMS/WhatsApp Integration with FluentBooking - FluentBooking

https://fluentbooking.com/docs/twilio-integration-with-fluentbooking/

[18] [19] Onsite sales - Prioticket

https://www.prioticket.com/onsite-sales

[20] Offline and background operation - Progressive web apps | MDN

https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Guides/Offline_and_background_operation

[21] How to Sell or Scan Tickets Anywhere Even Without Internet Connection - EventSprout

https://www.eventsprout.com/blog/how-to-sell-or-scan-tickets-anywhere-even-without-internet-connection/