What the parent must see/do •

Two payment choices at checkout:

1. Card (instant confirmation)

2. Tax-Free Childcare (booking held as Pending Payment until funds arrive) • If TFC is chosen, show a clear instruction panel with: • Provider’s TFC payee details (as you want them displayed) • A unique payment reference for the parent to use in their HMRC payment • A deadline/hold period (e.g., “Please pay within 5 days to keep your place”) • A Copy Reference button • After placing the order with TFC, the parent sees a Pending Payment screen and gets an email with the same instructions and deadline. • The booking is reserved but not confirmed until payment is received. If no payment by the deadline → auto-cancel and free the space. What the admin must have • A Bookings/Payments view showing a TFC “Pending” queue with: Child, activity/date(s), amount due, payment reference, due date/status. • Easy actions per booking: Mark as Paid, Part-paid, Cancel Unpaid, Convert to Credit (if needed). Bulk “Mark as Paid” for multiple matches. • Reminders: automatic email to parents (e.g., 48 hours before due). • Auto-cancel at the deadline with an email telling the parent it was released. How the money handling should work • Card payments: instant confirmation (as you already do). • TFC payments: treated like a bank transfer — you confirm manually when the money lands, using the parent’s reference to match it. • If the parent cancels a TFC booking: • Before 24 hours of the first session: credit or refund as per your policy (note: TFC funds typically must return as credit, not cash). • During a course: issue pro-rata credit for unused sessions (minus your admin fee if applicable). • Credits must be visible to parents and selectable at future checkout. Notifications & comms • On TFC selection: send instructions email with reference + deadline. • Reminder email before the deadline. • Confirmation email when admin marks as paid. • Cancellation email if unpaid at deadline. Settings you’ll need (per provider/venue) • Toggle: Enable Tax-Free Childcare • Hold period (number of days bookings are kept pending) • Instruction text (what parents see when choosing TFC) • Optional: default to credit for any refunds on TFC payments.

Safeguards & clarity • Clearly label bookings as Pending Payment until confirmed. • Don’t let pending bookings appear on registers or capacity as confirmed. • Always show the parent the financial impact of changes before they confirm (e.g., “No extra cost” or “+£5 late pick-up”). Nice-to-have (future) • Bank feed/Open Banking to auto-match incoming payments by reference and auto-confirm them.  
  
1) The rules (system logic) • Parent cancels ≥24 hours before first session starts • Outcome: Cash refund to original payment method, minus £2 admin fee. • Applies to: card payments. • For TFC/vouchers: issue account credit (no cash), minus £2. • Parent cancels after the course has started (any time <24h or mid-block) • Outcome: Pro-rata credit for all unused future sessions, minus £2 once per cancellation action. • No cash refunds (unless provider manually approves an exception). • Provider cancels a session or the whole activity • Outcome: Full refund or full credit (parent chooses), no admin fee. • No-show (parent did not cancel; session has passed) • Outcome: No refund/credit. • Extras (early drop/late pick-up) • If attached to a refundable session, treat the extra the same way as the session. • If a session is transferred, move extras with it when possible; otherwise pro-rata. 2) How the system decides (automation flow) • On “Cancel/Change”: • Calculate: time to session start, how many sessions remain, payment method (card vs TFC/voucher), extras attached. • Show parent a pre-decision summary before they confirm: • “Refund to card: £X (-£2 fee)” or • “Credit to your BookOn wallet: £Y (-£2 fee)” • For provider-cancelled: “Full £X back (no fee).” • If parent confirms: • Card: auto-create refund for the calculated amount; mark booking updated. • TFC/Voucher: auto-create credit in wallet; mark booking updated. • Decrement capacity / free places; update registers. 3) Pro-rata calculator (what needs to be built) • Inputs: total paid, list of booked sessions with dates, which have not occurred, value per session, extras per session. • Output: • Refundable cash (only if ≥24h before first session and card) • Credit amount (unused future sessions + refundable extras) • Apply one £2 fee per cancellation action, except provider-cancelled = no fee. • Round sensibly (to nearest penny), show breakdown to parent and in admin logs. 4) Wallet credit behaviour • Credits show as available balance on parent account (per provider). • At checkout, let parents toggle “Use £X credit”; auto-apply up to total. • Expiry: 12 months (configurable) with an expiry reminder 30 days before.  
  
5) TFC & voucher specifics • Never send cash back for TFC/voucher payments. • Always convert eligible amounts to credit (minus £2 where applicable). • If a booking was mixed (part card, part credit), refund the card part first, then return remainder as credit. 6) Admin controls (for exceptions) • Cancellation Requests queue with each auto-decision and amount shown. • Actions: Accept as shown, Offer credit instead, Decline with reason, Override amount. • Ability to waive the £2 fee on a case-by-case basis. • Full audit trail: who triggered, timestamps, amounts, method (cash/credit), reason. 7) Messaging & receipts (automation) • Parent sees before confirming: “You’ll receive £X back to card (-£2 fee)” or “£Y credit (-£2 fee).” • After completion: • Email + in-app notification with breakdown (sessions cancelled, fee, net refund/credit, credit balance). • For provider-cancelled: state “No admin fee applied.” 8) Edge cases to handle • Inside 24h of first session and parent asks to cancel entire block: offer pro-rata credit only. • Past sessions are never credited/refunded. • Transfers: • If moving to a more expensive session, take balance payment. • If cheaper, issue credit for the difference (no fee for transfers). • Chargebacks: lock refund actions while dispute is open. • Multiple children/activities: apply fee once per cancellation action per activity, not per child per session. 9) Reporting the business needs • Daily report: refunds issued (cash), credits issued, fees collected, provider-cancelled totals. • Liability report: unredeemed credit balance by provider. 10) Acceptance criteria (what you should test) • Cancelling ≥24h before first session (card): refund −£2 happens automatically; confirmation sent. • Cancelling mid-course: credit −£2 equals sum of future sessions; usable at next checkout. • Provider cancel: full refund/credit with no fee. • TFC bookings: no cash refunds; credit issued correctly. • Wallet credit applies correctly across future purchases and reduces amount charged.  
  
  
We need to make this the best booking tool Sakhawat. I have people waiting to use this, will give you work for many years to come.  
  
  
  
  
  
more requirements:  
  
[2:24 PM, 9/9/2025] Sakhawat Ali: Admin Dashboard -

1) Page purpose

A single “control center” that answers, at a glance:

• What’s running today?

• How much money moved?

• Do I have unread alerts?

• What do I need to do next?

2) Layout & components

Header bar (sticky)

• Title: “Dashboard”

• Notification bell: badge with unread count (e.g., 3). Click → opens Notifications panel.

• Profile menu: account, settings, log out.

Top row – Snapshot cards (4)

Each card is clickable and routes to the relevant area.

• Activities Running Today

• Metric: number of sessions + total attendees today (e.g., “3 activities · 60 children”).

• Parents Registered

• Metric: total active parent accounts (filter by term if available).

• Payments Collected Today

• Metric: £ total settled today (from payment gateway).

• Refunds / Credits Issued

• Metric: £ refunds + £ credits (today). Show deltas vs. yesterday/week.

Visuals: BookOn colors (navy/teal/white), large numeral, small label, subtle icon. Cards have hover state and aria-label.

Middle left – Upcoming Activities (preview list)

• Show the next 3–5 sessions (soonest first).

• Each row shows:

• Activity name (bold)

• Time (e.g., “3:30–4:30 PM”) and Venue (muted)

• Attendees = colored badge:

• Green Transactions.

Bottom – Notifications (feed)

• Ordered by recency. Each item has type, time, and quick action when relevant:

• New booking (open booking)

• Cancellation (open booking)

• Waitlist space created (open activity → Waitlist)

• Refund executed (open transaction)

• “Mark all read” and “View all notifications”.

[2:24 PM, 9/9/2025] Sakhawat Ali: 3) Interactions & behavior

• Date range control (optional, top right of snapshots): Today / This week / This month. Snapshots and finance card re-query on change.

• Bell badge updates in real time (websocket or polling).

• Snapshot cards & list rows have hover and focus states; entire card/row is a clickable target.

4) Responsive rules

• Desktop (≥1024px): two-column grid beneath snapshots (Upcoming Activities left, Finance right). Notifications full width under both.

• Tablet (768–1023px): snapshots in 2×2 grid; content stacks: Activities → Finance → Notifications.

• Mobile (<768px): everything stacks; snapshots become horizontal card scroller; bell stays in header.

5) Data requirements (examples)

• /api/dashboard/snapshot?range=today|week|month

• { activities\_running, attendees\_today, parents\_registered, payments\_total, refunds\_total, credits\_total }

• /api/activities/upcoming?limit=5

• [ { id, name, start\_time, end\_time, venue\_name, capacity, booked, waitlist\_count } ]

• /api/finance/summary?range=today|week|month

• { income, refunds, credits, timeseries: [{date,value}] }

• /api/notifications?limit=10&unread=true|false

• [ { id, type, title, created\_at, action\_url, read } ]

• /api/notifications/mark-read (POST)

6) States & empty handling

• No upcoming activities: show empty state with CTA “Add activity”.

• No payments today: “£0 collected today” with link to Transactions.

• No notifications: “You’re all caught up 🎉”.

• Errors: inline toast + retry for each card.

7) Accessibility

• Bell and cards require descriptive aria-labels (e.g., “You have 3 unread notifications”).

• Keyboard focus order: header → snapshots (left→right) → activities list → finance → notifications.

• Hit targets ≥44×44px on touch devices.

• Color is not the only signal: include text (e.g., “Full”) alongside red capacity badge.

8) Visual tokens (BookOn)

• Colors: Navy (#0F2230 approx), Teal (#2C8F7A approx), Success/Green, Warning/Amber, Danger/Red, Neutral greys.

• Badges: Capacity badge (solid pill with white text), Waitlist pill (red outline or solid).

• Typography: Activity name 16–18px bold; metadata 12–14px muted.

9) Acceptance criteria

• Page loads in <1s after auth (data may stream in).

• Switching range updates snapshots and finance within 500ms.

• Upcoming list is correctly sorted by start time and shows colored capacity + waitlist when applicable.

• Bell badge mirrors unread count; clicking opens panel with actionable items.

• Fully responsive; Lighthouse a11y score ≥95.

3 Templates Grid/List

Each Template Card shows:

• Name (e.g., “Year 1–2 Football”)

• Meta: Type (After-School), Years (Y1–Y2)

• Default Price (e.g., £6 per session) and Default Capacity (e.g., 20)

• Optional: small image/icon

• Actions (buttons on card):

• Create Course (primary)

• Edit Template

• Archive (overflow menu)

• Badges (if applicable): Requires Photo Consent, Medical Reminder, Popular (optional)

3.4 Empty States

• No templates yet → “You don’t have any templates.”

Buttons: + New Template and + Create New Course

⸻

4) Mobile UI (Layouts & Components)

4.1 Header

• Title: Activities

• Top-right + menu:

• Create New Course

• New Template

4.2 Filters (collapsible)

• Search

• Type

• Year/Key Stage

• Status

4.3 Template Cards (stacked)

• Name, Type, Years, Default Price/Capacity

• Buttons: Create Course (primary), Edit (secondary overflow)

5) Flows

5.1 Create Course from Template

1. Click Create Course on a template

2. Side panel (desktop) / bottom sheet (mobile) opens:

• Pre-filled: Name, Type, Years, default price & capacity

• Venue Selection (one or many)

• Date Range (start/end; recurrence weekday)

• Time

• Per-venue overrides: price, capacity, time

• Options based on Type:

• Holiday Club: Extras (Early Drop-off / Late Pick-up)

• After-School: fixed sessions (parents can’t edit dates)

• Preview Sessions (auto-generate; allow removing bank/half-term days)

• Publish (creates Course record(s) and session registers)

3. Success toast + deep link to the new Course in Courses/Registers tab

5.2 Create New Course (manual; not from template)

1. Click + Create New Course

2. Form collects: Name, Type, Years, Venue(s), Date Range, Time, Price/Capacity (per venue), flags (photo consent/medical)

3. Optional toggle: Save as new Template

4. Preview Sessions → Publish

5. If “Save as new Template” checked, create Template + Course; otherwise Course only

5.3 Edit Template

• Editable fields: Name, Type, Years, Description/What to bring, Default Price/Capacity, Flags, Tags, Image

• Important: Template edits affect future courses only (banner: “Changes don’t modify existing courses.”)

5.4 Archive Template

• Archive = hidden from default views; cannot create new courses from it

• Unarchive available via Status filter

6) Visual & Interaction Details

• BookOn palette:

• Navy (#0F2230 approx) for headings/navigation

• Teal (#2C8F7A approx) for primary actions (e.g., Create Course)

• Greys for cards/backgrounds; red/amber/green badges as needed

• Buttons: pill or rounded-rect; large tap targets

• Cards: 16px padding, clear hierarchy (Name > Meta > Price/Capacity > Actions)

• Keyboard & screen reader support; 44×44px minimum hit areas

7) Data Model (high level)

id

name

type // after\_school | breakfast | holiday | other

years // "Y1–Y2"

description

what\_to\_bring

default\_price // numeric

default\_capacity // integer

flags { photo\_consent\_required: bool, medical\_reminder: bool }

tags[] // e.g., ["Football","KS1"]

image\_url

status // active | archived

created\_at, updated\_at

Course (created from template or manual)

id

template\_id (nullable)

venue\_id

name

type

years

price // per venue override

capacity // per venue override

start\_date, end\_date

weekday, time

extras { early\_dropoff: bool, late\_pickup: bool } // holiday only

status // draft | published | archived

created\_at, updated\_at

Session

id

course\_id

date

start\_time, end\_time

status // scheduled | cancelled | completed

8) API Contracts (examples)

• GET /api/templates?search=&type=&years=&status= → list of ActivityTemplate

• POST /api/templates → create template

• PATCH /api/templates/:id → edit template (future courses only)

• POST /api/courses → create course (from template\_id or manual)

• POST /api/courses/:id/sessions/preview → returns generated session dates

• POST /api/courses/:id/publish → publish and create sessions

⸻

9) Sorting, Filtering & Permissions

• Default sort: Most recently edited template first

• Filters persist per user (local storage)

• Permissions:

• Admin: full CRUD on templates and courses

• Coordinator: create courses from templates; view/edit within venue scope

• Coach: view courses; no template editing

⸻

10) Edge Cases & States

• Template without price/capacity → show “Set defaults” warning on card

• Attempt to create course with missing fields → inline validation

• Multi-venue: create one Course per venue (each with its own sessions/register)

• Preview sessions must remove bank holidays (UK calendars) and allow manual untick

11) Acceptance Criteria

• Templates grid loads < 1s (cached; paginated if needed)

• Create Course from a template succeeds with per-venue overrides

• Manual Create New Course flow can optionally Save as Template

• Editing a template does not alter existing courses (banner shown)

• Accessibility: tab order logical; aria-labels on actions; color not sole indicator

Venue and payment route

BookOn — Venues & Payout Routing (Spec)

📌 Developer Requirements – Venue Bank Account + Franchise Percentage Cut

1. Venue → Business Account Relationship

• Each Venue must be linked to a Business Account.

• A Business Account represents the Stripe Connect account (Express/Standard) for payouts.

• When creating/editing a Venue in the admin dashboard:

• Dropdown to select Business Account (list of all onboarded accounts).

• Show status of the linked Stripe account (e.g., “Onboarded”, “Action Required”).

⸻

2. Franchise Percentage Cut (at source)

• Implement franchise fee deduction at payment source using Stripe Connect Destination Charges.

• For every booking/payment:

• Funds should flow directly to the venue’s connected account.

• BookOn takes its franchise fee % or fixed fee via application\_fee\_amount.

• Admin Settings → Franchise Fee Configuration:

• Fee type: % or fixed (£).

• Fee value.

• VAT mode: inclusive / exclusive.

⸻

3. Overrides

• Business Account (global default): set franchise fee rules at account level.

• Venue (override option):

• Checkbox: “Inherit Business Account fee” (default ON).

• If OFF → specify custom % or fixed fee for that Venue.

⸻

4. Refunds & Credits

• On refund, the franchise fee should reverse proportionally (default Stripe behaviour).

• If BookOn policy charges an admin fee (e.g., £2):

• This must be added as a separate line item in pricing logic.

• Refunds then exclude the admin fee automatically.

⸻

5. UI Requirements

Admin Dashboard → Venue Setup

• Fields:

• Venue Name

• Linked Business Account (dropdown)

• Override franchise cut? (toggle)

• Franchise Fee Type & Value (if override is ON)

Finance Reporting

• Transaction breakdown should show:

• Gross amount

• BookOn franchise fee

• Stripe fee

• Net to venue (with linked Business Account name)

⸻

6. Developer Implementation Notes

• Use Stripe Connect – Destination Charges with application\_fee\_amount.

• Store:

• business\_account\_id

• franchise\_cut\_type (percent or fixed)

• franchise\_cut\_value (e.g., 20 → 20%)

• Refunds: ensure refund\_application\_fee=true.

• Support multiple venues linked to the same or different business accounts.

⸻

✅ Result:

• Venues can route funds directly to their chosen bank account (via Stripe Connect).

• BookOn automatically deducts its % franchise cut at the source.

• Flexible: global franchise % at business account level, with per-venue overrides.

• Clean reporting for providers + franchise owners.

Communications Tab

⚠️ Areas to Improve

1. Template Management

• Currently, it just lists a few. Admins will want:

• Ability to create new templates (not just fixed 3).

• Rich text editor with branding (logo, colors).

• Dynamic placeholders (e.g., {ParentName}, {ChildName}, {ActivityName}, {Venue}).

• Toggle on/off per template.

• Preview before saving.

2. Broadcast Messages

• Right now they look like flat notifications.

• Add:

• Audience segmentation (all parents, specific venue, specific activity group, specific year group).

• Channel choice (Email, SMS, In-app notification).

• Scheduling (send now vs schedule).

• Delivery stats (sent, opened, clicked).

3. Notifications vs Email

• “Communications” should ideally cover all channels. Consider tabs for:

• Emails

• SMS (if enabled via Twilio or similar)

• Push / In-app notifications

4. Logs & Compliance

• Show history of automated sends (e.g., “Booking confirmation sent to Jane Smith at 14:32”).

• GDPR / opt-in toggle → parents must be able to unsubscribe from marketing but still get essential booking emails.

5. Integration Hooks

• Option to connect Mailchimp, SendGrid, or HubSpot if providers want advanced marketing.

⸻

BookOn — Communications Tab (v1.0)

Goal

Give providers one place to configure automated emails, send broadcasts, and track message delivery/engagement with clear, mobile-first UI.

Success criteria

• Providers can create/edit templates, send broadcasts, and see real-time delivery, open, click, bounce, unsubscribe for every message.

• Works great on desktop and mobile.

• No “Untracked” rows—every message shows a status (even if “Pending events…”).

⸻

Information architecture (3 sub-tabs)

1. Automated Emails — manage templates & triggers

2. Broadcasts — one-off/send-later messages to targeted audiences

3. Email Log — searchable message ledger with delivery/engagement

Sub-tab 2 — Broadcasts

What shows (desktop)

• Header: Broadcasts  [Send Broadcast]

• Filters: Date range | Audience (All/By Venue/By Activity/By Year Group) | Channel (Email/SMS/In-app)

• Table columns: Title, Channel(s), Audience, Sent, Delivered, Open %, Click %, Date, Actions (View report)

Send Broadcast (flow)

• Step 1 — Compose: Title, Subject, Body (rich text), Channel(s)

• Step 2 — Audience: All parents / by Venue(s) / Activity(ies) / Year group(s); show current reach count

• Step 3 — Schedule: Send now / pick date & time

• Step 4 — Review & Send: summary + warning about unsubscribes for marketing

• After send: land on Broadcast report with KPIs (Sent, Delivered, Opened, Clicked, Bounced, Unsubscribed) and top links clicked.

⸻

Sub-tab 3 — Email Log (NEW)

Desktop table

• Header: Email Log

• Filters (sticky): Search | Date range | Activity | Venue | Message type | Status

• Broadcast summary bar (if filtered by a broadcast or message type):

“Booking Confirmations — 95 Sent • 90 Delivered • 75 Opened (83%) • 20 Clicked (22%)”

• Columns:

1. Date/Time

2. Recipient (Parent Name • email)

3. Message Type

4. Status (badges above) — show most advanced status; hover reveals timeline (e.g., Delivered 10:12 → Opened 10:18 → Clicked 10:21)

5. Actions: View Order, View Message (side panel), Resend (if failed)

Mobile layout

• Each email = card with:

• Top: Parent Name • email

• Second line: Message Type • Date/Time

• Status badges row

• Action buttons: View Order | View Message | Resend

• Filter button opens sheet with same filters as desktop.

Message preview (side panel/drawer)

• Shows the rendered email (with tokens resolved for that recipient).

• Right column: Event timeline (Sent/Delivered/Open/Click/Bounce) with timestamps & metadata.

⸻

Technical requirements

ESP & tracking

• Use a transactional ESP (SendGrid, Postmark, Mailgun, or SES + SNS).

• Enable open & click tracking (pixel + tracked links).

• Ingest events via webhooks and map to internal statuses.

Event model (suggested tables)

• emails

• id, broadcast\_id (nullable), template\_id (nullable), parent\_id, order\_id (nullable),

• to\_email, to\_name, message\_type, subject, channel='email',

• provider\_message\_id, sent\_at, last\_status, last\_status\_at

• email\_events

• id, email\_id, event\_type (delivered|opened|clicked|bounced|unsubscribed),

• provider\_event\_id, meta (JSON for bounce reason, link URL), occurred\_at

• broadcasts

• id, title, subject, body\_html, audience\_query (JSON), channels, scheduled\_for, created\_by

• email\_templates

• id, name, trigger, subject\_tmpl, body\_html\_tmpl, active, brand\_overrides (JSON)

Webhook handling

• Verify signature (HMAC) from ESP.

• Idempotently upsert events (provider\_event\_id).

• Derive latest status per email from most advanced event.

Performance & UX

• Paginate logs (50–100 rows/page), infinite scroll on mobile.

• Async event updates (poll or websocket), so statuses update without reload.

Compliance

• Classify messages as Transactional vs Marketing.

• Enforce unsubscribe only on Marketing; always send Transactional.

• Include provider-specific reply-to and footer info.

• Store consent & unsubscribe events per parent.

Accessibility

• Badge colors meet contrast (WCAG AA).

• Icons + text labels (don’t rely on color only).

• Keyboard focus states on desktop.

⸻

Copy & micro-interactions

• Empty Email Log: “No messages yet. They’ll appear here as you send confirmations, reminders, and broadcasts.”

• Bounce tooltip: “Mailbox full / Address does not exist / Blocked by recipient.”

• Resend confirm: “Resend this message to {email}? This will re-issue the latest version.”

• Broadcast report badge: “Open Rate 62% • Click Rate 21%”.

Finance tab

Updated Design Brief – Finance Tab (Admin Dashboard)

Objective

Build a Finance Dashboard within the admin panel that gives providers a single source of truth for all money movement: transactions, discounts, credits, refunds, and reports.

⸻

Structure – 5 Sub-Tabs

1. Transactions

2. Discounts

3. Credits

4. Refunds

5. Reports

⸻

1. Transactions Tab

• Purpose: Track all payments.

• Data Fields:

• Parent/Customer Name

• Activity booked

• Amount paid (£)

• Payment method (Card, Tax-Free Childcare, Credit)

• Status (Paid, Pending, Failed)

• Date/time

• Actions: Export, filter by activity/venue/date.

⸻

2. Discounts Tab

• Purpose: Manage promotional tools.

• Features:

• List of active/expired discounts (name, type, % or £, expiry date, usage).

• Revenue impact tracking.

• Create Discount form with conditions (min spend, venue, sibling).

3. Credits Tab

• Purpose: Handle credit balances as refund alternatives.

• Features:

• Parent name + current balance (£).

• Issued credits (manual or automated via policy).

• Redeemed credits (activity + date).

• Expiry (if set).

• Action: “Issue Credit” button.

• Parent-side integration: “Use Credit” toggle at checkout.

⸻

4. Refunds Tab ✅ (NEW)

• Purpose: Give transparency + automation control.

• Data Fields:

• Parent/Customer Name

• Activity/Booking reference

• Amount refunded (£)

• Refund method (Original payment / Credit)

• Status (Processing, Refunded, Failed)

• Date/time

• Features:

• Automated refunds logged with policy notes (e.g., “Refund issued – £2 admin fee deducted”).

• Manual override option (admin can trigger refund).

• Exportable for finance reconciliation.

⸻

5. Reports Tab

• Purpose: Show top-level insights.

• Metrics:

• Gross vs Net Revenue

• Discounts applied

• Credits issued/redeemed

• Refund totals (number + value)

• Weekly/monthly/yearly trends.

• Visuals: Bar charts, line graphs, pie chart for breakdown.

• Export Options: CSV/PDF.

Create activity

Here’s a clear expert-style brief you can give your developer for the “Add Activity” feature in BookOn:

⸻

🎯 Goal

Allow admins (schools, clubs, providers) to easily create new activities (after-school clubs, holiday camps, etc.) with minimal friction while ensuring flexibility for different formats (single-day, recurring weekly, holiday camps with early/late options).

⸻

🖥️ Form Fields & Logic

1. Basic Details

• Activity Name → Free text (e.g., Year 1 & 2 Football Club).

• Activity Type (dropdown) → Options: Afterschool, Breakfast, Holiday Club, Other.

• Venue (dropdown) → Preloaded from Venues tab.

• Description (optional) → Free text, supports formatting for parents.

⸻

2. Scheduling

• Dates (toggle) →

• Start Date → calendar picker.

• End Date → calendar picker.

• Auto-generate weekly sessions.

• Admin can untick specific dates (e.g., half-term, holidays).

• Times → Start & End time (HH:MM).

• Special Logic:

• Holiday Club only → Extra options:

• Early drop-off (toggle + price).

• Late pick-up (toggle + price).

⸻

3. Capacity & Pricing

• Capacity → Max number of children per activity.

• Price →

• Flat fee OR per-session fee (pro-rata auto-calculated).

• Ability to mark Free (if £0).

• Discounts → Apply existing promo codes (from Finance > Discounts).

⸻

4. Registers & Tracking

• Each session auto-creates a register entry tied to the activity.

• Admin can view/edit registers later.

this is my clients requirements i copied and paste here please make pdf for that