

TRAVELPASS

Technical Proposal & Architecture Blueprint

Prepared for **Kirsty Norton**

Prepared by **Jagat Jaishankar**

Project timeline – **90 days**

(70 days of building + 20 days of testing and launch support)

Proposal cost of **\$120 USD**

Created on **February 18, 2026**

A curated directory of vetted safe spaces and adventure experiences for LGBTQIA+ travellers, solo female adventurers, and the skate/surf community.

TABLE OF CONTENTS

- SECTION 01 – [Executive Summary](#)
 - SECTION 02 – [Final Vision & Product Description](#)
 - SECTION 03 – [User Roles & Flows](#)
 - SECTION 04 – [Full Product Pages Breakdown](#)
 - SECTION 05 – [Tech Stack](#)
 - SECTION 06 – [System Architecture](#)
 - SECTION 07 – [Database Structure](#)
 - SECTION 08 – [Security & Hosting Plan](#)
 - SECTION 09 – [Running Costs](#)
 - SECTION 10 – [Development Cost Breakdown](#)
 - SECTION 11 – [90-Day Build Plan](#)
 - SECTION 12 – [Equity Hybrid Option](#)
-

EXECUTIVE SUMMARY

TravelPass is a curated directory of vetted safe spaces, adventure experiences, and community driven business... made for travellers who value authenticity, inclusivity and also independence. The platform focuses on three core communities:

- LGBTQIA+travellers seeking affirming spaces
 - solo female adventures who prioritise safety and connection
 - skate / surf / adventure community looking authentic local experiences
-

HOW IT WORKS

Travellers discover hand picked partner LGBTQIA+ friendly cafes, solo-female-run events, surf schools with inclusive vibes, adventure tour operators who value safety connection.

They purchase a time-based pass (7 days, 14 days, 1 month, 3 months or pay-per-day pass) and unlock exclusive discounts at every partner venue. Payment is instant via Stripe. They receive the QR code via email, show it at the venue, and access their perks.

Partners aren't just listings... they are value aligned businesses who share the platform's mission. They write blog content (solo travel tips, LGBTQIA+ safety guidelines, surf spot recommendations etc.) and actively engage with the community.

Travellers have the ability to refer other travels and earn a commission on a successful pass purchase as well (affiliates). Each partner is personally vet by the platform owner to ensure they align with the platform's values... safety, inclusivity, authenticity and adventure.

WHAT THIS PROPOSAL COVERS

This document is the complete technical blueprint for building TravelPass in 90 days. It includes full system architecture, database design, user flows, security plan, cost breakdown, and a milestone-by-milestone build plan. By the end, this proposal should give a clear idea on what exactly gets built, how it gets built and also what it costs to launch and run.

FINAL VISION & PRODUCT DESCRIPTION

This isn't a generic discount platform... it's a venue driven ecosystem connecting conscious travellers with vetted local businesses. Every partner is hand-selected because they align with the mission.

This is about trust. Travellers don't just want discounts... they want to know a cafe won't misgender them, a guesthouse that is genuinely safe for solo women, a surf school that respects personal values. Every business is vetted personally and that curation is the product.

THE CORE PRODUCT

The TravelPass does four things:

- Curates trusted directory of value-aligned partners
- Sells time-based discount passes to travellers
- Verifies passes at partner venues
- Empowers partners to build community through content (blogs and articles)

The admin approves posts before they go live. Travellers earn affiliate commissions by referring other travellers to the travel pass. This creates a self-sustaining community-powered ecosystem.

REVENUE MODEL

The platform generates revenue from pass sales only. every time a traveller buys a pass 100% of the payments goes through Stripe account (minus the platform fee of 2.9% + 30c processing fee).

No partner fees. No subscription costs. Just transaction revenue.

Projected Revenue:

- 100 passes/month × \$30 avg = \$3,000/month
 - 500 passes/month × \$30 avg = \$15,000/month
-

WHO THIS SERVES

- LGBTQIA+ travellers (finding safe affirming spaces by pre-vetting businesses)
 - solo female travellers (safety is non-negotiable)
 - skate / surf / adventure community (local spots that don't miss culture)
-

THE END VISION

Within 90 days, TravelPass launches with 20-30 hand picked partners across key traveler hubs. The directory is live. Passes are selling. Partners are writing blog content. The community is growing.

This isn't just a discount pass... it's a movement. A trusted ecosystem for travellers who've been under served by mainstream platforms.

This proposal outlines how to build it.

USER ROLES & FLOWS

This platform (TravelPass) serves three types of users:

- admin (platform owner)
 - partners (vetted businesses)
 - travellers (the pass buyers)
-

01 – ADMIN

(access to full platform)

Core Responsibilities:

- personally vet and approve every partner application
- review and publish partner blog posts
- monitor pass sales and revenue
- monitor / manage affiliate payouts
- resend lost passes to travelers
- update platform settings and pricing

Vetting Partners Flow:

- partner applies (either through public form or from a personal invite)
- receive email notification
- review their application
 - do they align with platform values?
 - are they LGBTQIA+-friendly, solo-female-safe, or community-focused?
 - do they have a track record of inclusivity?
- approve → partner goes live immediately
- reject → send feedback explaining why (e.g. "not values-aligned")

Reviewing Blog Content Flow:

- partner submits blog post
- receive notification
- review content
 - is it helpful to the community?
 - does it align with platform tone (authentic, inclusive, adventure-focused)?
 - any safety concerns or misinformation?
- approve → post goes live
- request edits → send feedback to partner

Monitoring Sales & Community Growth:

- view dashboard: pass sales, revenue, active partners
- see which partners are driving the most engagement
- track affiliate referrals

Key Actions:

- approve/reject partner applications
 - approve/reject blog posts
 - resend lost passes
 - manage pricing
 - process affiliate payouts
-

O2 – PARTNER

(access to manage own listing, write blogs, scan QR codes)

Core Responsibilities:

- keep listing updated
- write blog content that attracts travelers
- scan traveller QR codes to verify passes

Onboarding Flow:

- receives invite from admin or applies via public form
- fills out application:
 - business name, category (yoga/wellness, cafe, guesthouse etc.)
 - description (highlighting inclusivity, safety, community focus)
 - photos (up to 5)
 - discount / perks offered to pass holders
 - Google Maps link, address, open hours
 - contact info (email + phone)
 - key question? "how does your business create safe/inclusive spaces for LGBTQIA+ travelers, solo female travelers or the adventure community?"
- submits application
- admin reviews and approves
- partner receives login credentials
- listing goes live

Day-to-Day Flow:

1. logs into dashboard
2. writes blog post (e.g., "Why Gaia Soul is a safe space for queer yogis")
3. submits for your review
4. approved → post goes live on their listing page
5. scans traveler QR codes when they visit

Key Actions:

- edit listing details
 - write blog posts
 - scan QR codes
 - share affiliate link
 - track earnings
-

03 – TRAVELLER

(access to browse directory, read blogs, buy passes - no account needed)

Discovering the Platform Flow:

1. comes across TravelPass (google search / instagram / referrals etc.)
2. lands on homepage
3. sees mission statement and values upfront
4. browses partners by category or community tag

Buying a Pass Flow:

1. clicks "buy pass"
2. selects duration (7/14/30/90 days or pay-per-day)
3. enters name and email
4. redirects to Stripe checkout
5. completes payment
6. receives confirmation page with QR code
7. email arrives with same QR code and pass details
8. saves QR code (screenshot or email)

Using the Pass Flow:

1. visits a partner venue (e.g., Gaia Soul)
2. shows QR code on phone
3. partner scans it
4. system shows valid/expired
5. partner applies discount

Engaging with Content Flow:

1. reads partner blog posts
2. discovers new partners through blog content
3. shares posts with friends in their community

Key Actions:

- browse directory
 - read blog posts
 - buy pass
 - show QR code at venues
 - refer other travellers via affiliate link
 - track referral earnings (if enrolled)
-

HOW THE ROLES INTERACT

Admin ↔ Partner:

The admin personally vets every partner, the partners know they're part of a curated community. They write content and the admin approves.

Admin ↔ Traveler:

Travelers trust the directory because admin curated it. The admin handles lost passes. Admin becomes the guardian of the community's values.

Partner ↔ Traveler:

Partners provide safe spaces. Travelers support businesses that align with their values. Partners write content that helps travelers navigate the local culture.

FULL PRODUCT PAGES BREAKDOWN

01 – ADMIN DASHBOARD (HOME)

URL: [/admin]

Purpose: Platform overview and quick actions

Key Elements:

- total passes sold (today, this week, this month)
- revenue earned (today, this week, this month)
- active partner count
- pending partner applications badge
- pending blog posts badge
- recent pass sales list (traveler name, email, pass type, date)
- traveler affiliate activity summary (total affiliates, pending payouts)
- quick action buttons
("view partners", "view blog posts", "view sales", "view affiliates", "settings")

Primary Actions:

- view pending partner applications
 - view pending blog posts
 - monitor sales in real-time
 - navigate to key sections
-

02 – PARTNER MANAGEMENT PAGE

URL: [/admin/partners]

Purpose: Review, approve and manage partner venues

Key Elements:

- tabs (pending, approved, rejected)
- each partner card shows:
 - business name, category
 - photos preview (up to 5 photos)
 - discount offer / perks offered
 - inclusivity statement (how they serve the community)
 - application date
 - approve and reject buttons (pending)
 - edit and delete buttons (approved)

Primary Actions:

- review partner applications
 - approve or reject partners (with feedback)
 - edit or remove existing partners
 - view partner listing preview
-

03 – BLOG MANAGEMENT PAGE

URL: [/admin/blog-posts]

Purpose: Review and publish partner blog posts

Key Elements:

- tabs (pending, published, rejected)
- each blog post card shows:
 - post title
 - author (partner name)
 - featured image preview
 - excerpt (first 100 characters)
 - submission date
 - approve/reject/request edits buttons (when pending)
 - edit/delete buttons (when published)

Primary Actions:

- review blog post content
 - approve or reject posts
 - request edits (send feedback to partner)
 - delete published posts
-

04 – PASS SALES PAGE

URL: [/admin/sales]

Purpose: View all pass purchases and resend lost passes

Key Elements:

- searchable table:
 - traveler name, email
 - pass type, purchase date, expiry date
 - status (active/expired)
 - price paid
 - referred by (if purchased via affiliate link)
- search by name or email
- filter (active, expired, all)
- resend pass button for each entry
- export sales data (CSV)

Primary Actions:

- search for travelers
 - resend QR codes
 - view pass details
 - export sales data
-

O5 – TRAVELLER AFFILIATE DASHBOARD

URL: [/admin/affiliates]

Purpose: track traveller affiliates referrals and manage payouts

Key Elements:

- list of all travelers enrolled in affiliate program
- each row shows:
 - traveler name, email
 - affiliate link
 - referrals made (total count)
 - successful pass purchases from referrals
 - earnings owed
 - stripe Connect status (connected/not connected)
 - payout status (pending/paid)
- badge labelled "commission paid" per affiliate
- badge labelled "commission is pending" per affiliate
- total affiliate payouts pending

Primary Actions:

- view affiliate activity
 - mark payouts as paid
 - export payout report
 - view individual affiliate's referral history
-

06 – SETTINGS PAGE

URL: [/admin/settings]

Purpose: Manage platform settings and prices

Key Elements:

- pass pricing configuration:
 - fixed tier prices (7/14/30/90 days)
 - pay-per-day rate (\$5/day)
- traveler affiliate commission rate (% per successful referral)
- platform branding (logo, banner, colors, tagline)
- Stripe account status
- email notification settings

Primary Actions:

- update pass pricing
 - configure affiliate commission rate
 - edit platform branding
 - manage Stripe account
-

07 – PARTNER DASHBOARD

URL: [/partner/dashboard]

Purpose: View listing, blog posts, and access tools

Key Elements:

- listing preview (as it appears on directory)
- QR scanner button (large, prominent)
- blog posts list (draft/pending/published)
- total scans count (lifetime)
- recent scans list (traveler name, date, pass type)
- edit listing button

Primary Actions:

- scan QR codes
 - write new blog post
 - edit listing
 - view scan history
-

08 – EDIT LISTING PAGE

URL: [/partner/edit-listing]

Purpose: Update business details and discount

Key Elements:

- all registration fields (editable):
 - business name, category, description
 - photos (upload up to 5)
 - discount offer + perks
 - address, Google Maps link
 - opening hours
 - contact info (email, phone, website)
 - inclusivity statement
- save changes button (triggers re-approval if any changes are made)

Primary Actions:

- edit business details
 - update photos
 - change discount offer
 - update inclusivity statement
 - submit for re-approval
-

09 – BLOG POST EDITOR

URL: [/partner/write-blog]

Purpose: Write and submit blog posts

Key Elements:

- post title input
- rich text editor (formatting, headings, lists, links, images)
- featured image upload
- category selector
(local guides, LGBTQIA+ resources, solo female safety, surf/skate culture, etc.)
- SEO meta description (optional)
- save as draft button
- submit for review button

Primary Actions:

- write blog post
 - upload featured image
 - save draft
 - submit for admin review
-

10 – QR SCANNER PAGE

URL: [/partner/scanner]

Purpose: Verify traveller passes in real-time

Key Elements:

- camera viewfinder (activates device camera)
- scan result display:
 - **VALID:** green check, traveler name, pass expiry, discount to apply
 - **INVALID:** red cross, "pass expired" or "invalid code"
- scan history log below scanner (name + type of pass + time scanned)

Primary Actions:

- scan QR code
 - view scan result
 - return to dashboard
-

11 – TRAVELLER AFFILIATE PAGES

URL: [/affiliates/dashboard]

Purpose: track referrals and earnings (for travellers enrolled in affiliate program)

Key Elements:

- affiliate referral link (copy button)
- referral instructions ("share this link with friends and earn 10% when they buy")
- referrals summary:
 - total clicks on link
 - total pass purchases from referrals
 - total earnings (pending + paid)
- recent referrals list:
 - referral date
 - pass type purchased
 - earnings from referral
 - payout status (pending/paid)
- Stripe Connect status:
 - if not connected: "Connect Stripe account" button
 - if connected: "manage Stripe account" link
- payout history (date, amount, status)

Primary Actions:

- copy affiliate link
 - share link (social media, WhatsApp, email)
 - connect Stripe account for payouts
 - view referral activity
 - track earnings
-

12 – TRAVEL AFFILIATE ONBOARDING PAGE

URL: [/affiliate/signup]

Purpose: Enroll in traveller affiliate program

Key Elements:

- affiliate program explainer "earn 10% on every pass your friends buy"
- signup form:
 - name
 - email
 - password (create account)
 - agree to affiliate terms
- button labelled "create account"
- After signup... redirected to affiliate dashboard with prompt to connect Stripe

Primary Actions:

- create affiliate account
 - connect Stripe for payouts
 - get affiliate link
-

13 – DIRECTORY HOME PAGE

URL: [/]

Purpose: Browse partners and buy passes

Key Elements:

- hero section
- community tags (filter buttons)
(LGBTQIA+-friendly, solo-female-safe, skate/surf/adventure etc.)
- partner venue grid/list with filters
(yoga/wellness, cafe, guesthouse, surf school, tour operator, all)
(search by name or location)
- each partner card shows:
(business name, category badge, primary photo, tags, star rating etc.)
- footer (FAQ, terms, privacy, become a partner, become an affiliate)

Primary Actions:

- browse partners by category or community tag
 - search partners
 - click to view partner details
 - click "Buy Pass"
 - click "Become an Affiliate"
-

14 – PARTNER DETAIL PAGE

URL: [/partner/[id]]

Purpose: View full partner information and blog posts

Key Elements:

- business name
- community tags (LGBTQIA+, solo-female-safe etc.)
- photo gallery (up to 5 images as carousel)
- description
- inclusivity statement (highlighted)
- star rating + review count (Google Maps)
- discount offer (highlighted and bold)
- address + embedded Google Maps
- opening hours
- contact info (email, phone, website optional)
- partner blog posts (if any)
- button labelled "buy pass"

Primary Actions:

- view full partner details
 - see location on map
 - read partner blog posts
 - click "buy pass"
-

15 – BLOG POST PAGE

URL: [/blog/[post-id]]

Purpose: Read full partner blog post

Key Elements:

- post title
- author (partner name + link to their listing)
- publication date
- community tags (LGBTQIA+, solo female, surf/skate, etc.)
- featured image
- full blog content (rich text with images)
- related partners (other businesses in same category or community tag)
- button labelled "buy pass"(sticky)
- social share buttons

Primary Actions:

- read blog content
 - click author name to view their listing
 - click related partners
 - click "buy pass"
-

16 – CHECKOUT PAGE

URL: [/checkout]

Purpose: Purchase the pass

Key Elements:

- pass type selector:
 - fixed tiers: 7 days (\$29), 14 days (\$49), 1 month (\$79), 3 months (\$199)
 - pay-per-day: slider/input to select days price updates (\$5/day)
- name input
- email input
- affiliate tracking (if arrived via affiliate link "referred by [affiliate name]")
- total price display
- button labelled "complete purchase" (redirects to Stripe)

Primary Actions:

- select pass duration
 - enter name and email
 - complete Stripe checkout
-

17 – CONFIRMATION PAGE

URL: [/confirmation/[pass-id]]

Purpose: Display purchased pass and QR code

Key Elements:

- success message: "your pass is ready!"
- QR code (large, scannable)
- pass details:
 - traveler name
 - pass type (e.g., "14 Day Pass")
 - purchase date and expiry date
- prompt: "screenshot this or check your email"
- secondary CTA: "want to earn commissions? become an affiliate"
- button labelled "return to directory"

Primary Actions:

- screenshot confirmation page
 - click "become an affiliate"
 - return to directory home
-

18 – AFFILIATE SIGNUP PAGE

URL: [/affiliate/join]

Purpose: Enroll travellers in affiliate program

Key Elements:

- affiliate program explainer
- how it works:
 1. create free account
 2. get unique referral link
 3. share link with friends
 4. earn 10% on every pass they purchase
- signup form (name, email, password)
- "join now" button
- login link (for existing affiliates)

Primary Actions:

- create affiliate account
 - login (if already enrolled)
-

TECH STACK

Every technology choice in this stack is justified by three criterias... speed of develop, long-term maintainability, and cost efficiency at scale. This is a modern JavaScript full-stack with managed services to minimise operational overhead.

FRONT END

NEXT.JS (APP ROUTER)

Why:

- single codebase for frontend and backend
 - app Router provides (file-based routing, server components, and built-in data fetching)
 - server-Side Rendering (SSR) for fast loads and good SEO
 - API routes handle all backend logic without needing Express
 - Vercel deployment is native and effortless (one-click deploy)
-

TAILWIND CSS

Why:

- quick UI development
 - consistent design system
 - responsive by default
 - tiny production bundle
 - easy to customise
-

DAISYUI

Why:

- built on TailwindCSS
 - theme system (easily customise colors, typography, and spacing)
 - lightweight
 - fast development (buttons, cards, modals, forms, tables already styled)
 - fully customisable (can override any component style)
-

BACKEND & DATABASE

MONGODB ATLAS

Why:

- document-based storage fits perfectly with this data model
- managed service (automatic backups, scaling, encryption at rest)
- free tier for MVP (512MB storage, shared cluster, perfect for initial launch)
- scales seamlessly (upgrading to dedicated clusters when traffic grows)
- mongoose integration

MONGOOSE

Why:

- schema validation (define data structures and enforce them)
 - relationship modeling (easy to reference related documents)
 - middleware hooks (run logic before/after database operations)
 - query builder (cleaner, more readable queries than raw MongoDB syntax)
-

AUTHENTICATION

NEXTAUTH.JS (V5)

Why:

- built for Next.js
- magic link authentication (passwordless login via email)
- role-based access control (easy to assign roles... admin, partner, affiliates)
- session management (database sessions out of the box)
- adapters for MongoDB

Authentication methods:

- admin: magic link
 - partners: magic link
 - traveler affiliates: username + password (account required for tracking)
 - regular travelers: no authentication (guest checkout)
-

PAYMENTS

STRIPE CHECKOUT

Why:

- hosted payment page (Stripe handles the card form)
- supports one-time payments (perfect for pass purchases)
- guest checkout (no Stripe account required for travelers)
- automatic email receipts (Stripe sends confirmation emails upon payment)
- webhooks (notifies your server when payment succeeds/fails)

STRIPE CONNECT

Why:

- required for traveler affiliate payouts
- platform creates connected accounts for each affiliate
- automated payouts (affiliates receive commissions directly to their bank)

Payment flow:

- pass purchases → go to admin's Stripe account (minus Stripe fees)
 - affiliate commissions → paid out via Stripe Connect to affiliate's bank account
-

EMAIL

RESEND

Why:

- simple API (send emails with 3 lines of code)
- react email templates (write emails in JSX, not HTML strings)
- free tier (3,000 emails/month which is plenty for MVP)
- built-in analytics (see open rates, click rates, bounces)

Emails sent:

- pass purchase confirmation (traveler)
 - partner application notification (admin)
 - partner approval/rejection (partner)
 - blog post approval/rejection (partner)
 - pass resend (traveler)
 - affiliate signup confirmation (traveler affiliate)
 - affiliate payout notification (traveler affiliate)
-

HOSTING & DEPLOYMENT

VERCEL

Why:

- native Next.js support (built by the same team that makes Next.js)
 - zero-config deployment
 - automatic HTTPS (SSL certificates generated and renewed automatically)
 - edge caching (faster load times globally)
 - serverless functions (API routes scale automatically, only pay for usage)
 - free tier (unlimited personal projects, perfect for MVP)
 - environment variables (secure storage for API keys and secrets)
-

TECH STACK SUMMARY

Layer	Technology
Frontend	Next.js 16 (App Router)
Styling	Tailwind CSS + DaisyUI
Database	MongoDB Atlas + Mongoose
Auth	NextAuth.js
Payments	Stripe Checkout + Stripe Connect
Email	Resend
Hosting	Vercel
File Storage	Vercel Blob
Blog Editor	Lexical

SYSTEM ARCHITECTURE

TravelPass is a full-stack application built on NextJS. everything lives in one codebase (frontend, backend, API routes and database operations)

The Four Layers:

- frontend (NextJS + React + TailwindCSS + DaisyUI)
 - API routes (NextJS handles all backend logic)
 - Database (MongoDB Atlas + Mongoose)
 - external services (Stripe Checkout + Connect, Resend, Auth, Vercel Blob)
-

HOW DATA FLOWS

Traveler Buys a Pass:

- traveler visits directory → selects pass → clicks checkout
- if arrived via affiliate link, affiliate tracking cookie is set
- redirected to Stripe's hosted payment page
- stripe processes payment → sends webhook to backend
- backend:
 - creates pass record in MongoDB
 - generates QR code
 - if purchase came from affiliate link, creates commission record
 - sends confirmation email to traveler (with QR code)
 - sends notification to admin
 - if affiliate referral, sends notification to affiliate
- traveler redirected to confirmation page showing QR code

Partner Applies to Directory:

- partner fills out registration form (including inclusivity statement) → submits
- backend uploads photos to Vercel Blob, creates partner record (status: pending)
- admin gets email notification
- admin reviews application in dashboard
- admin approves → backend updates status to "approved"
- sends login email to partner
- partner clicks magic link → NextAuth creates session → partner logs in

Partner Writes Blog Post:

- partner writes blog post in rich text editor → submits for review
- backend creates blog post record (status: pending)
- admin gets email notification
- admin reviews post in dashboard
- admin approves → post status changes to "published"
(goes live on partner's listing page and blog archive)

Traveler Enrolls in Affiliate Program:

- traveler clicks "become an affiliate"
- fills out signup form (name, email, password)
- backend creates user account with role: "affiliate"
- generates unique affiliate referral link
- sends welcome email with link
- traveler logs in, sees affiliate dashboard
- traveler clicks "Connect Stripe Account"
- redirected to Stripe Connect onboarding
- completes bank details on Stripe's domain
- stripe returns connected account ID to platform
- platform saves account ID in affiliate record
- affiliate can now receive payouts

Traveler Shares Affiliate Link:

- affiliate copies link from dashboard
- shares link on social media, WhatsApp, blog, etc.
- friend clicks link → lands on TravelPass homepage with affiliate tracking
- friend buys pass → backend creates commission record
- affiliate sees earnings update in dashboard
- admin marks payout as paid → Stripe Connect transfers to affiliate's bank

Venue Scans QR Code:

- partner opens scanner on dashboard → scans traveler's QR code
 - backend looks up pass in MongoDB, checks expiry date
 - returns VALID or INVALID
 - scanner displays result + logs scan event
 - scan record includes partner ID and timestamp
-

AUTHENTICATION

NextAuth handles two types of login:

Magic Link (Admin + Partners + Traveller Affiliates):

- user enters email → NextAuth generates token
- resend sends magic link email
- user clicks link → NextAuth verifies token, creates session
- user redirected to dashboard based on role (admin/partner)

Sessions stored in MongoDB with role-based access control on every protected route.

SECURITY

- API routes protected with session checks and role-based access (admin/partner/affiliate)
 - sensitive keys stored as Vercel environment variables
 - HTTPS enforced everywhere
 - magic link tokens expire after 24 hours
 - Stripe Connect signatures verified on all webhooks
 - file uploads sanitised and size-limited
-

SCALING

The architecture scales horizontally:

- Vercel serverless functions scale automatically
- MongoDB Atlas upgrades to dedicated cluster in one click
- Stripe Connect handles unlimited affiliates
- Vercel Blob CDN serves images globally

At 1,000+ passes/month, only two changes needed:

1. Upgrade MongoDB to dedicated cluster (~\$25-50/month)
2. Upgrade Vercel to Pro plan (\$20/month)

No code changes required.

DATABASE STRUCTURE

The database is built on MongoDB Atlas with Mongoose for schema validation. Data is organised into collections, each representing an entity in the platform. Relationships between collections are handled through reference IDs.

COLLECTION OVERVIEW

- users (admin, partners, and traveler affiliates)
 - partners (partner venue listings)
 - passes (traveler pass purchases)
 - scans (QR code scan events)
 - blog_posts (partner blog content)
 - affiliate_referrals (traveler affiliate tracking)
-

01 – COLLECTION USERS

Stores authentication data for admin, partners and traveller affiliates.

Fields:

- _id (unique user ID)
- email (user email, unique, indexed)
- password (hashed password, only for affiliates)
- role ("admin" | "partner" | "affiliate")
- createdAt (account creation timestamp)
- updatedAt (last update timestamp)

Relationships:

- role = "admin" → user is platform owner
- role = "partner" → user has one record in partners collection
- role = "affiliate" → user has affiliate tracking data in affiliate_referrals collection

Indexed Fields:

- email (for fast login lookups)
 - role (for role-based queries)
-

02 – COLLECTION PARTNERS

Stores partner venue listings.

Fields:

- _id (unique partner ID)
- userId (reference to users collection)
- businessName (venue name)
- category ("yoga/wellness" | "cafe" | "guesthouse" | "surf_school" | "tour_operator" | "other")
- description (text description)
- inclusivityStatement (how they serve LGBTQIA+, solo female, or adventure communities)
- photos (array of image URLs, max 5)
- discount (discount or perk text)
- address (street address)
- googleMapsLink (Google Maps URL)
- openingHours (text description)
- contactEmail (contact email)
- contactPhone (contact phone)
- website (website URL, optional)
- rating (Google Maps star rating, 0-5, manually entered)
- reviewCount (Google Maps review count, manually entered)
- communityTags (array of tags: "lgbtqia_friendly", "solo_female_safe", "skate_surf_adventure", "eco_conscious", "wellness")
- status ("pending" | "approved" | "rejected")
- rejectionReason (text reason if rejected)
- createdAt (application timestamp)
- approvedAt (approval timestamp if approved)
- updatedAt (last update timestamp)

Relationships:

- one partner has many blog posts
- one partner has many scans

Indexed Fields:

- status (filter pending/approved partners)
 - userId (user to partner mapping)
 - communityTags (filter by community)
-

03 – COLLECTION PASSES

Fields:

- `_id` (unique pass ID)
- `travelerName` (traveler's name)
- `travelerEmail` (traveler's email, indexed)
- `passType` ("7_days" | "14_days" | "1_month" | "3_months" | "pay_per_day")
- `daysPurchased` (number of days, only for pay-per-day)
- `purchaseDate` (purchase timestamp)
- `expiryDate` (pass expiry timestamp, indexed)
- `qrCode` (unique QR code string, indexed)
- `qrCodeImageUrl` (QR code image URL)
- `priceInCents` (total price paid in cents)
- `stripeCheckoutSessionId` (Stripe session ID)
- `affiliateReferrerId` (user ID of affiliate who referred this purchase, optional, indexed)
- `status` ("active" | "expired")
- `createdAt` (pass creation timestamp)

Relationships:

- one pass has many scans
- one pass may belong to one affiliate (if referred)

Indexed Fields:

- `qrCode` (QR verification lookups)
 - `travelerEmail` (resend pass searches)
 - `expiryDate` (filter active/expired passes)
 - `affiliateReferrerId` (track affiliate referrals)
-

04 – COLLECTION SCANS

Stores QR code scan events.

Fields:

- _id (unique scan ID)
- passId (reference to passes collection, indexed)
- partnerId (reference to partners collection, indexed)
- scanTimestamp (when scan occurred)
- scanResult ("valid" | "invalid")
- invalidReason ("expired" | "not_found", if invalid)

Relationships:

- one scan belongs to one pass
- one scan belongs to one partner

Indexed Fields:

- passId (pass scan history)
 - partnerId (partner scan history)
-

05 – COLLECTION BLOG POSTS

Stores partner blog content.

Fields:

- `_id` (unique post ID)
- `partnerId` (reference to partners collection, indexed)
- `title` (post title)
- `slug` (URL-friendly slug, unique, indexed)
- `content` (rich text content, HTML)
- `featuredImage` (image URL)
- `category` ("travel_tips" | "lgbtqia_resources" | "solo_female_safety" | "surf_skate_culture" | "local_guides" | "other")
- `communityTags` (array matching partner tags: "lgbtqia_friendly", "solo_female_safe", etc.)
- `metaDescription` (SEO description, optional)
- `status` ("draft" | "pending" | "published" | "rejected")
- `rejectionReason` (admin feedback if rejected)
- `publishedAt` (publish timestamp if published)
- `createdAt` (post creation timestamp)
- `updatedAt` (last update timestamp)

Relationships:

- one blog post belongs to one partner

Indexed Fields:

- `partnerId` (fetch all posts by a partner)
 - `status` (filter pending/published posts)
 - `slug` (URL lookups)
 - `communityTags` (filter by community)
-

06 – COLLECTION AFFILIATE REFERRALS

Stores traveler affiliate tracking and earning.

Fields:

- `_id` (unique affiliate record ID)
- `userId` (reference to users collection with role = "affiliate", indexed)
- `affiliateLink` (unique referral link/code, unique, indexed)
- `stripeConnectId` (Stripe connected account ID for payouts, indexed)
- `stripeConnectStatus` ("pending" | "connected" | "disconnected")
- `totalReferrals` (count of passes purchased via this affiliate)
- `totalEarnings` (total commission earned in cents)
- `pendingPayouts` (amount owed but not yet paid, in cents)
- `paidPayouts` (amount already paid out, in cents)
- `createdAt` (affiliate signup timestamp)
- `updatedAt` (last update timestamp)

Relationships:

- one affiliate has many referred passes (via `passes.affiliateReferrerId`)

Indexed Fields:

- `userId` (user to affiliate mapping)
 - `affiliateLink` (track referral clicks and purchases)
 - `stripeConnectId` (webhook processing for payouts)
-

07 – COLLECTION AFFILIATE PAYOUTS

Stores individual affiliate payout transactions.

Fields:

- `_id` (unique payout ID)
- `affiliateId` (reference to `affiliate_referrals` collection, indexed)
- `amountInCents` (payout amount in cents)
- `status` ("pending" | "paid" | "failed")
- `stripeTransferId` (Stripe transfer ID if paid via Stripe Connect)
- `paidAt` (timestamp if paid)
- `createdAt` (payout creation timestamp)

Relationships:

- one payout belongs to one affiliate

Indexed Fields:

- `affiliateId` (fetch all payouts for an affiliate)
 - `status` (filter pending/paid payouts)
-

SECURITY AND HOSTING PLAN

Security and hosting are built into the architecture from day one. This section covers how the platform is deployed, how sensitive data is protected, and how the system stays secure at scale.

HOSTING (VERCEL)

- GitHub repository connected to Vercel
 - every push to main branch triggers automatic deployment
 - preview deployments created for every pull request
 - rollback to any previous deployment in one click
 - Custom domain configuration
 - primary domain: travelpass.com
 - SSL certificates auto-generated and renewed by Vercel
 - HTTPS enforced on all requests
 - Environment variables
 - all secrets stored in Vercel project settings
 - separate variables for production and preview environments
 - never committed to Git, never exposed to frontend
 - Serverless function configuration
 - API routes deployed as serverless functions
 - auto-scaling based on traffic
 - cold start optimisation with edge middleware
-

SECURITY MEASURES

API Route Protection

Every protected API route checks authentication and authorisation

Role-Based Access Control:

- admins can only access `/api/admin/*` routes and their own data
- partners can only access `/api/partner/*` routes and their own listing
- travellers can only access `/api/traveller/*` routes and their own affiliation
- all data queries filtered by user ID to prevent data leaks

Stripe Webhook Security:

Stripe webhooks verify signatures to prevent spoofing

Magic Link Token Security:

NextAuth magic links include multiple security layers:

- tokens are cryptographically signed (HMAC-SHA256)
- tokens expire after 24 hours
- tokens are single-use only (deleted after verification)

Database Security:

- encryption at rest (AES-256)
- encryption in transit (TLS 1.2+)
- IP whitelist (only Vercel IPs can connect)
- database user authentication (username + password)

File Upload Security:

- max file size: 5MB per upload
- allowed file types: `image/png`, `image/jpeg`, `image/webp` only
- files scanned before storage
- CDN URLs are public but unpredictable (long random strings)

Session Management:

- session duration: 30 days
- session stored in MongoDB
- session token rotated on every request
- automatic logout on inactivity (30 days)

Data Isolation:

- every database query scoped to the requesting user ID
 - partners cannot see other partners' data
 - travellers cannot see other travellers' data
 - admin has read-only access (cannot modify partner and travellers data)
-

BACKUP & DISASTER RECOVERY

Vercel Deployment Rollback:

- every deployment saved permanently
- instant rollback to any previous deployment
- zero downtime rollback (traffic switches immediately)

Data Export:

- admin can export all platform data as JSON
 - regular exports scheduled weekly (automated via cron)
 - exports stored in encrypted cloud backup
-

MONITORING & ALERTS

What Gets Monitored:

- API error rates (500 errors, failed requests)
- stripe webhook failures
- database connection failures
- email delivery failures

Alert Channels:

- email notifications to admin
- vercel dashboard alerts
- stripe dashboard alerts

Logging:

- all API requests logged with timestamps
 - error stack traces captured
 - webhook events logged for debugging
 - logs retained for 30 days
-

RUNNING COSTS

This section breaks down the monthly operational costs for running the platform. The infrastructure is designed to start free and scale costs proportionally with growth.

Only paying more when the platform is making more.

- At MVP launch with 0-50 passes, the platform costs less than \$5/month to run.
- At 100+ passes operational costs are around \$60-90/month (~2% of revenue).

All services offer free tiers that cover initial launch, with clear upgrade triggers based on usage thresholds.

COST BREAKDOWN BY SERVICE

Service	Free Tier	At Scale (500+ passes/month)
Vercel	Unlimited	\$20/month (Pro plan)
MongoDB Atlas	512MB shared cluster	\$25-50/month (dedicated)
Vercel Blob	1GB storage	\$5-10/month
Resend	3,000 emails/month	\$10/month (10k emails)
Stripe Checkout	Pay-per-transaction (2.9% + 30¢)	Pay-per-transaction (2.9% + 30¢)
Stripe Connect	Included with Stripe	Included with Stripe
Domain	\$1.50/month (\$18/year)	~\$1.50/month
Total	~\$2/month	\$60-90/month

COST AT DIFFERENT STAGES

Stage 1: Launch (0-50 passes/month)

- monthly cost: ~\$2-5
- monthly revenue (50 passes × \$30 avg): \$1,500

Stage 2: Growth (50-200 passes/month)

- monthly cost: ~\$12-15
- monthly revenue (125 passes × \$30 avg): \$3,750
- affiliate payouts (10% of \$3,750): \$375
- net profit: ~\$3,360

Stage 3: Scale (500+ passes/month)

- monthly cost: ~\$60-90
 - monthly revenue (500 passes × \$30 avg): \$15,000
 - affiliate payouts (10% of \$15,000): \$1,500
 - net profit: ~\$13,410
-

WHEN TO UPGRADE

- Vercel Pro (\$20/month): when bandwidth hits 80GB+/month
 - MongoDB Dedicated (\$25-50/month): when database hits 400MB+
 - Resend Paid (\$10/month): when email volume hits 2,500+/month consistently
-

STRIPE FEES

Stripe charges 2.9% + 30¢ per transaction.

Stripe Connect fees for affiliate payouts:

- No additional platform fee beyond the standard 2.9% + 30¢ on transaction
 - Stripe Connect transfers to affiliates are free (included in standard pricing)
-

DEVELOPMENT COST BREAKDOWN

This section covers the one-time cost to build the platform.

Solo Build: One Developer (ME)

Price breakdown estimate is based on:

- Scope: single curated directory for Sri Lanka
 - Features:
 - partner listings
 - pass sales
 - QR verification
 - partner blogs
 - traveler affiliate program
 - admin dashboard
 - Timeline:
 - 70 days active build
 - 20 days testing/launch support
-

PHASE 01 – FOUNDATION

(WEEKS 1 TO 2) – \$700

- project setup (Next.js, MongoDB, Vercel, Tailwind, DaisyUI)
- authentication system (NextAuth magic links, role-based access)
- database setup (all collections, schemas, indexes)
- admin dashboard shell (layout, navigation, protected routes)

PHASE 02 – CORE DIRECTORY

(WEEKS 3 TO 4) – \$700

- public directory homepage (partner grid, filters, search)
- partner detail pages (photos, description, map, perks)
- partner onboarding flow (application form, photo uploads)
- admin partner management (approve, reject, edit, delete)
- email notifications (Resend integration)

PHASE 03 – PASS SYSTEM

(WEEKS 5 TO 6) – \$750

- stripe Checkout integration (pass purchasing)
- QR code generation and email delivery
- confirmation page
- partner QR scanner (camera integration + verification)
- admin pass sales dashboard (search, resend, export)
- pass expiry logic

PHASE 04 – CONTENT AND AFFILIATES

(WEEKS 7 TO 8) – \$750

- blog post editor (rich text, image upload)
- blog moderation flow (submit, review, approve, reject)
- blog pages (listing, individual post, categories)
- affiliate signup and dashboard
- Stripe Connect for affiliate payouts
- affiliate tracking (referral links, commission calculation)
- admin affiliate management

PHASE 05 – POLISH & LAUNCH

(WEEKS 9 TO 10) – \$600

- mobile responsiveness across all pages
- performance optimisation (Lighthouse 90+)
- SEO setup (meta tags, sitemap, structured data)
- final testing across all user flows
- domain connection and deployment
- handover documentation + loom walkthrough

Total: ~~\$3,500~~ \$2,000

Ongoing Maintenance (optional) \$200/month for bug fixes, minor updates, and technical support after launch.

ALTERNATIVE: EQUITY HYBRID \$1,500 upfront (across the same milestones) + 15% equity in TravelPass. Open to discussion.

Payment Structure:

- 30% upfront (\$600)... before work begins
- 40% at phase 2 completion (\$800)... when features are built
- 30% at launch (\$600)... when platform goes live

Equity Option: If you'd prefer a hybrid structure, I'm open to discussing:

- reduced cash payment (\$1,500)
- small equity stake (20%)
- this aligns long-term incentives and reduces upfront cost

What is Included:

- full platform build (all features listed in this proposal)
- 30 days post-launch bug fixes and support
- source code ownership (you own everything)
- deployment and hosting setup
- basic admin documentation

What is NOT included:

- content creation (partner recruitment, copywriting, photography)
 - ongoing maintenance beyond 30 days (available separately at \$200/month)
 - legal docs (terms, privacy policy... you'll need a lawyer)
 - marketing, SEO, or Google Ads management
-

90 DAY BUILD PLAN

The build is split into 5 phases over 10 weeks...

(70 days active development + 20 days testing and launch support)

01 – FOUNDATION (WEEKS 1-2)

Timeline: Days 1-14

Goals:

- project infrastructure set up
- authentication working
- database operational
- admin dashboard shell live

Tasks:

- set up Next.js project with App Router
- configure Tailwind CSS + DaisyUI
- connect MongoDB Atlas cluster
- design and implement database schema... all collections
(users, partners, passes, scans, blog_posts, affiliate_referrals, affiliate_payouts)
- create Mongoose models with validation
- set up NextAuth with magic link authentication
- implement role-based access control (admin, partner, affiliate)
- build admin dashboard shell (layout, navigation, protected routes)
- configure Vercel deployment pipeline
- set up environment variables

Deliverables:

- empty Next.js project deployed to Vercel
 - database live with all collections created
 - admin can log in via magic link
 - protected routes working
-

02 – CORE DIRECTORY (WEEKS 3-4)

Timeline: Days 15-28

Goals:

- public directory functional
- partner onboarding complete
- admin partner management operational

Tasks:

- build public directory homepage:
 - partner grid/list layout
 - community tag filters (LGBTQIA+, solo female, surf/skate etc.)
 - category filters and search bar
 - responsive design
- build partner detail pages:
 - photo gallery (up to 5 images)
 - business description and inclusivity statement
 - Google Maps embed
 - discount/perks display
 - opening hours and contact info
- build partner onboarding flow:
 - public application form
 - photo upload to Vercel Blob
 - inclusivity statement field
- build admin partner management:
 - view pending/approved/rejected partners
 - approve or reject applications with feedback
 - edit or delete existing partners
- integrate Resend for email notifications:
 - partner application received (to admin)
 - partner approved/rejected (to partner)

Deliverables:

- public directory live with browsable partners
 - partners can apply via form
 - admin can approve/reject partners
 - email notifications working
-

03 – PASS SYSTEM (WEEKS 5-6)

Timeline: Days 29-42

Goals:

- travelers can buy passes
- QR codes generated and delivered
- partners can scan and verify passes
- admin can manage pass sales

Tasks:

- integrate Stripe Checkout:
 - build checkout page (pass type selector, name/email inputs)
 - implement Stripe session creation
 - handle webhook for successful payments
- build pass creation logic:
 - generate unique QR code
 - calculate expiry date based on pass type
 - store pass in database
 - upload QR code image to Vercel Blob
- build confirmation page:
 - display QR code and pass details
 - prompt to screenshot or check email
- email pass to traveler (QR code embedded)
- build partner QR scanner:
 - camera integration (browser native API)
 - scan QR code and send to verification API
 - display VALID or INVALID result
 - log scan events in database
- build admin pass sales dashboard:
 - searchable table (traveler name, email, pass type, date)
 - filter by active/expired
 - resend pass button
 - export to CSV
- implement pass expiry logic (automated via cron job)

Deliverables:

- QR codes generated and emailed
 - partners can scan QR codes
 - admin can view and manage all pass sales
-

04 – CONTENT AND AFFILIATES (WEEKS 7-8)

Timeline: Days 43-56

Goals:

- partners can write and submit blog posts
- admin can review and publish blog content
- travelers can enroll as affiliates
- affiliate tracking and payouts functional

Tasks:

- build blog post editor for partners:
 - lexical rich text editor integration
 - featured image upload
 - category and community tag selectors
 - save as draft and submit for review
- build admin blog moderation flow:
 - view pending/published/rejected posts
 - approve, reject, or request edits
 - email notifications to partners
- build public blog pages:
 - individual blog post pages (SEO optimised)
 - blog archive page with filters
 - related partners section
- build traveler affiliate signup:
 - public signup form (email + password)
 - account creation with role: "affiliate"
 - generate unique referral link
 - welcome email with instructions
- build affiliate dashboard:
 - display referral link with copy button
 - show referral stats (clicks, conversions, earnings)
 - Stripe Connect "Connect Account" button
 - payout history
- integrate Stripe Connect:
 - create connected accounts for affiliates
 - implement onboarding flow
 - store connected account ID in database

- implement affiliate tracking:
 - track referral link clicks (cookies/URL params)
 - attribute pass purchases to affiliates
 - calculate 10% commission on successful referrals
 - store commission records in database
- build admin affiliate management:
 - view all affiliates and their earnings
 - mark payouts as paid
 - export payout reports

Deliverables:

- partners can write blog posts
- blog posts published on public pages
- travelers can enroll as affiliates
- affiliates can track referrals and earnings
- Stripe Connect payouts functional

05 – POLISH & LAUNCH (WEEKS 9-10)

Timeline: Days 57-70

Goals:

- platform fully tested and optimised
- SEO configured
- deployed to production
- launched live

Tasks:

- mobile responsiveness audit:
 - test all pages on mobile devices
 - fix layout issues
 - optimise touch targets and forms
- performance optimisation:
 - image compression and lazy loading
 - code splitting
 - database query optimisation
 - lighthouse score 90+ target

- SEO setup:
 - meta tags for all public pages
 - open Graph tags for social sharing
 - generate sitemap.xml
 - structured data for blog posts and partners
 - Robots.txt configuration
- final testing:
 - admin flows (partner approval, blog moderation, affiliate payouts)
 - partner flows (listing management, blog writing, QR scanning)
 - affiliate flows (signup, tracking, Stripe Connect)
 - traveler flows (browsing, pass purchase, QR usage)
 - edge cases and error handling
- security audit:
 - verify API route protection
 - test input validation
 - check Stripe webhook signatures
 - review password hashing
- domain connection:
 - configure custom domain (e.g., travelpass.com)
 - set up wildcard DNS if needed
 - SSL certificate verification
- deploy to production:
 - push to main branch
 - verify environment variables
 - run smoke tests on live site
- create admin documentation:
 - partner approval process
 - blog moderation guidelines
 - affiliate payout workflow
 - platform settings management
- record Loom walkthrough video

Deliverables:

- platform live and accessible at custom domain
 - all features tested and working
 - SEO configured
 - admin documentation provided
 - platform ready for partner onboarding
-

WEEKS 11-13 (DAYS 71-90): BUFFER & POST-LAUNCH SUPPORT

06 – BUFFER & POST LAUNCH SUPPORT (WEEKS 11-13)

Timeline: Days 71-90

Goals:

- monitor platform stability
- fix any post-launch bugs
- support initial partner and affiliate onboarding
- optimise based on real-world usage

Activities:

- monitor error logs and performance metrics
- fix bugs as they're discovered
- support your first partner approvals and blog post reviews
- support first affiliate payouts
- answer questions and provide technical guidance
- optimise database queries if performance issues arise
- adjust UI/UX based on user feedback

This buffer period ensures:

- any unexpected delays in earlier phases don't impact launch
- platform is stable before going fully live
- you have support during critical early days
- we can iterate based on real user behavior

MILESTONE-BASED PAYMENT SCHEDULE

Payment 1 (30% = \$600):

Due: before work begins (Day 0)

Unlocks: phase 1 & 2 (Foundation + Core Directory)

Payment 2 (40% = \$800):

Due: upon completion of Phase 3 & 4 (Pass System + Content & Affiliates)

Milestone: all core features built and functional in staging environment

Payment 3 (30% = \$600):

Due: upon launch (Phase 5 complete)

Milestone: platform live on production domain and admin documentation delivered

WHAT HAPPENS IF DEVELOPMENT FINISHES EARLY?

If all phases complete ahead of schedule, the extra time goes toward:

- additional polish and refinement
- more thorough testing
- early launch support
- feature enhancements you request

The 90-day timeline is a maximum and not a minimum. The goal is to launch a stable, high-quality platform... whether that takes 70 days or 90.

EQUITY HYBRID OPTION

This is an alternative to the standard \$2,000 fixed fee. It reduces upfront cost while aligning long-term incentives.

Structure:

\$1,500 upfront payment (across the same milestone schedule)
20% equity stake in the TravelPass brand/business

Payment Schedule (Equity Option):

- Payment 1 (30% = \$450): before work begins
- Payment 2 (40% = \$600): upon completion of phase 3 & 4
- Payment 3 (30% = \$450): upon launch

What Equity Means:

- 20% ownership of the TravelPass business
- Revenue share: 20% of net profits after operating costs
- Decision-making: Kirsty retains full operational control
- Equity is tied to the TravelPass brand, not the codebase

What This Requires:

- a simple equity agreement signed by both parties
- clear definition of profit calculation
- agreed review period (e.g., annual review of terms)

Either option (fixed fee or equity hybrid) includes the same full scope, timeline, and deliverables outlined in this proposal.
