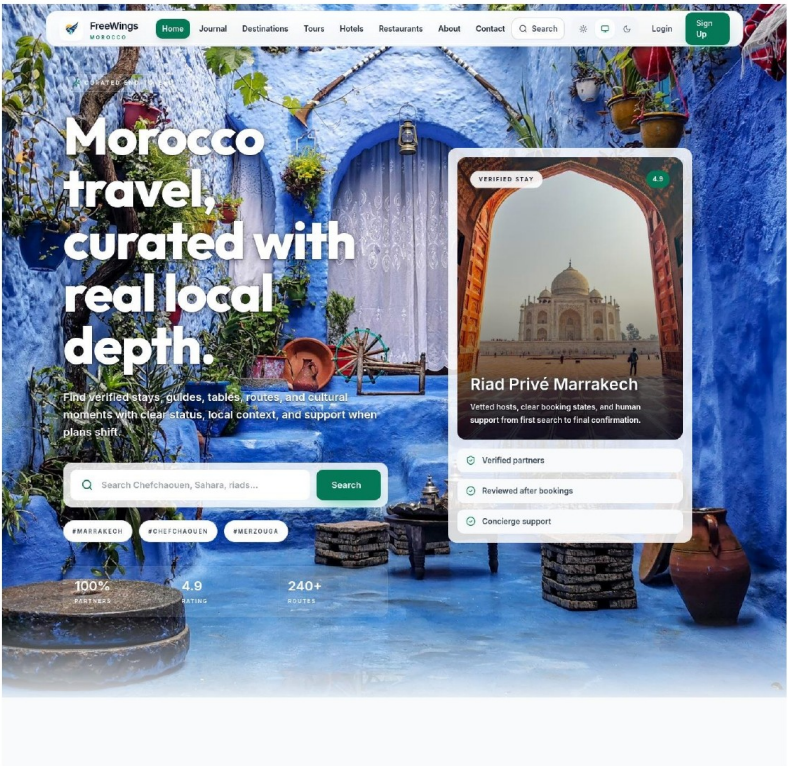


# Engineering a unified Moroccan tourism platform from discovery to reservation and payment

A decoupled Laravel and React product connecting tourism discovery, recommendations, bookings and CMI payment workflows.

| STATUS                                                    | TEAM               | DEMO                               | SOURCE                 |
|-----------------------------------------------------------|--------------------|------------------------------------|------------------------|
| Two-person academic product<br>• Functional demonstration | Two-person project | Functional demo deployment planned | Private project source |



Primary visual evidence

ANASS BENAMEUR · FULL-STACK PRODUCT ENGINEER

# The product, the problem and the proof

FreeWings is a two-person final-cycle product exploring a unified Moroccan tourism experience. It brings destinations, hotels, tours, restaurants, activities and editorial content into one discovery journey, then connects search and recommendation to a polymorphic booking flow and a real integration with CMI — Morocco's interbank card payment gateway — implemented with HMAC-SHA512 signing, idempotent webhook handling and transactional reconciliation, running in certification/preproduction mode.

|                                            |                                     |                                           |                                                 |
|--------------------------------------------|-------------------------------------|-------------------------------------------|-------------------------------------------------|
| <b>2 people</b><br>DOCUMENTED PROJECT TEAM | <b>94 pages</b><br>TECHNICAL REPORT | <b>6 domains</b><br>TOURISM CONTENT TYPES | <b>CMI</b><br>CERTIFICATION-MODE<br>INTEGRATION |
|--------------------------------------------|-------------------------------------|-------------------------------------------|-------------------------------------------------|

## Context

Travellers researching Morocco bounce between a hotel site, a tour operator, a restaurant listing and a blog, then often book nowhere in particular because nothing connects the pieces into one journey. FreeWings, built for a licence-level final-cycle project at SUPMTI with my partner Oumaima Bakkali, tests whether a single product — for travellers, tourism professionals and administrators — can hold that whole journey without collapsing the real differences between what a hotel, a restaurant and a multi-day tour actually need to sell.

## My role and ownership

- Shared product and technical design
- Laravel REST API and domain modelling
- React/TypeScript product interface
- Search, recommendation and booking workflows
- Payment, security and observability design

## Personal contribution

Shared architecture and implementation across API, SPA, booking, payment, security and observability.

## Why the existing workflow needed a system

A hotel, a restaurant and a guided tour are not the same kind of thing to sell — different pricing logic, different availability rules, different cancellation terms — but a traveller wants to compare and book all three without noticing the difference underneath. On top of that, a payment gateway callback can arrive twice, arrive late, or arrive with a tampered amount, and a booking system that isn't built for that will either double-charge someone or lose track of whether they paid at all.

### Before and after

#### BEFORE

- Fragmented tourism discovery across unrelated services
- No shared search and recommendation experience
- Different reservation flows for each service type
- Payment callbacks vulnerable to duplicate processing without idempotency
- Limited cross-layer visibility into failures

#### AFTER

- One catalogue spanning destinations, hotels, tours, restaurants and activities
- Unified search and recommendation experience
- Polymorphic booking lifecycle shared across service types
- HMAC-SHA512 validation, amount checks and idempotent reconciliation design
- Request IDs, HTTP telemetry, Web Vitals and client error reporting

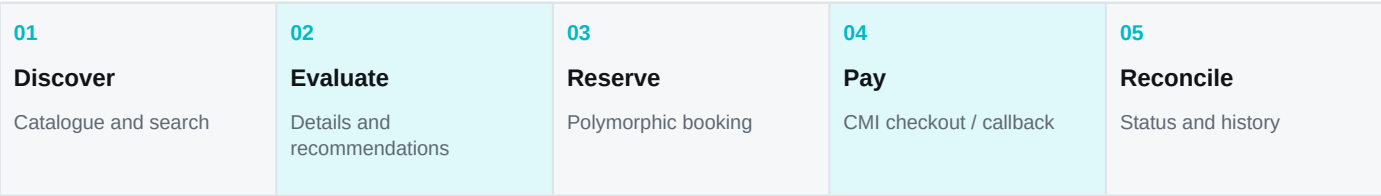
### Meaningful result

One coherent product experience across discovery, reservation and payment states.

# How the system works

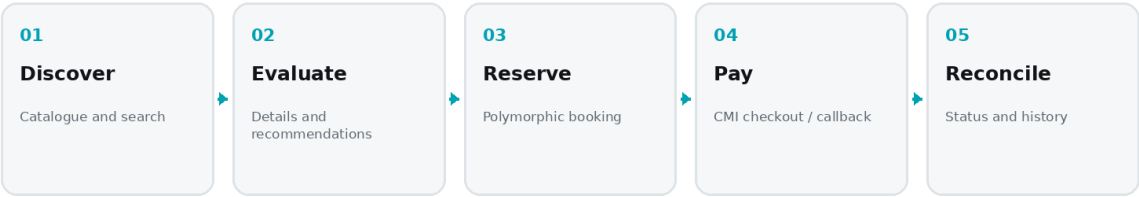
Two independent applications talk over REST. Laravel handles routing, FormRequest validation, Policy-based authorization, services and API resources; React handles the interface with TanStack Query for server state and Zod for typed validation. A booking is polymorphic — one bookings table serves hotels, tours and restaurants alike via bookable\_type/bookable\_id — so adding a fourth service type doesn't mean building a fourth booking system from scratch.

## Operational workflow



### SYSTEM ARCHITECTURE

## FreeWings — Intelligent Tourism Platform



Evidence-led architecture diagram · generated from the verified project workflow

Payment runs in three phases: the backend signs a payload with HMAC-SHA512 and hands the traveller off to CMI's hosted checkout, so card data never touches FreeWings' own servers; CMI posts a webhook back, verified against the signature, checked for amount match, and processed through a firstOrCreate idempotency guard so a duplicate callback is silently absorbed instead of double-processed; and the traveller's return page polls booking status every five seconds until the reconciliation lands. A parallel observability layer — request IDs, HTTP telemetry, Web Vitals, client-side error reporting via sendBeacon — exists so a failure anywhere in that chain is diagnosable, not a mystery.

# Architecture and decisions

Two independent applications communicate through REST. Laravel organizes routes, middleware, FormRequests, controllers, policies, services, models and API resources. React uses TypeScript, TanStack Query, Zod and feature-oriented components. The recommendation score is computed directly in SQL — rating times 20, plus review count times 4, plus active hotels, restaurants, tours and activities weighted separately — ordered with orderByRaw so ranking stays fast without pulling every destination into PHP to sort it.

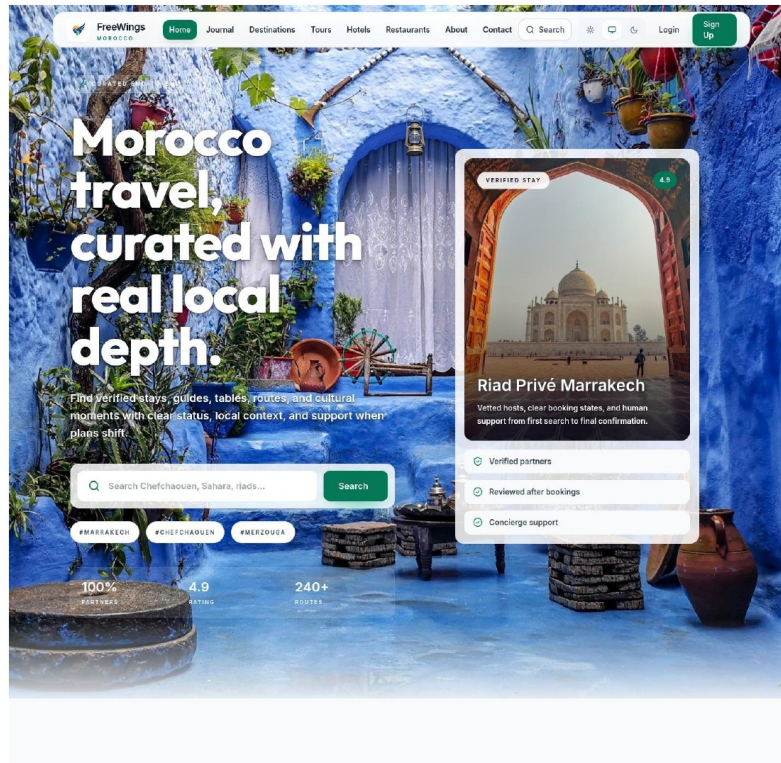
## Critical engineering decisions

|                                                                                                                                                                                                         |                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DECISION 01</b><br><b>Decoupled API and SPA</b><br>The split supports a richer customer experience and reusable contracts, while requiring deliberate authentication and deployment coordination.    | <b>DECISION 02</b><br><b>Polymorphic booking model</b><br>Hotels, tours and activities share reservation lifecycle logic without forcing every service into an identical data model.       |
| <b>DECISION 03</b><br><b>Idempotent payment reconciliation</b><br>CMI callbacks may repeat, so signature checks, amount validation and controlled reconciliation protect reservation and payment state. | <b>DECISION 04</b><br><b>Client and server instrumentation</b><br>Request IDs, HTTP telemetry, Web Vitals and sendBeacon error reporting connect user experience with backend diagnostics. |

## Technology stack

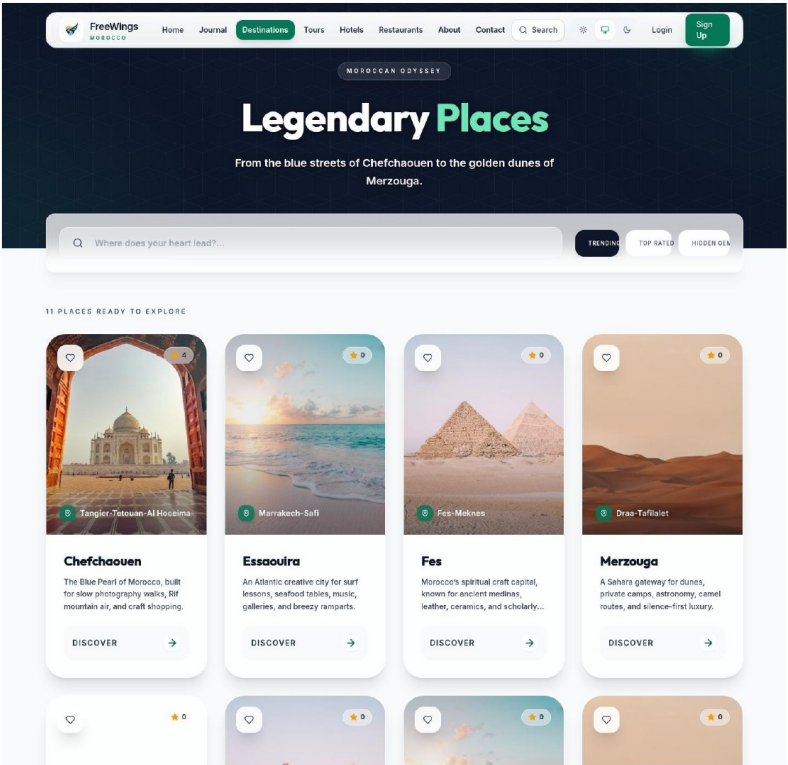
|            |         |          |            |         |               |
|------------|---------|----------|------------|---------|---------------|
| Laravel 10 | PHP 8.2 | React 18 | TypeScript | Sanctum | CMI 3D Secure |
|------------|---------|----------|------------|---------|---------------|

## Evidence 1 of 4



Homepage combining discovery, positioning and recommendations.

# Evidence 2 of 4



Destination catalogue designed for comparison and exploration.



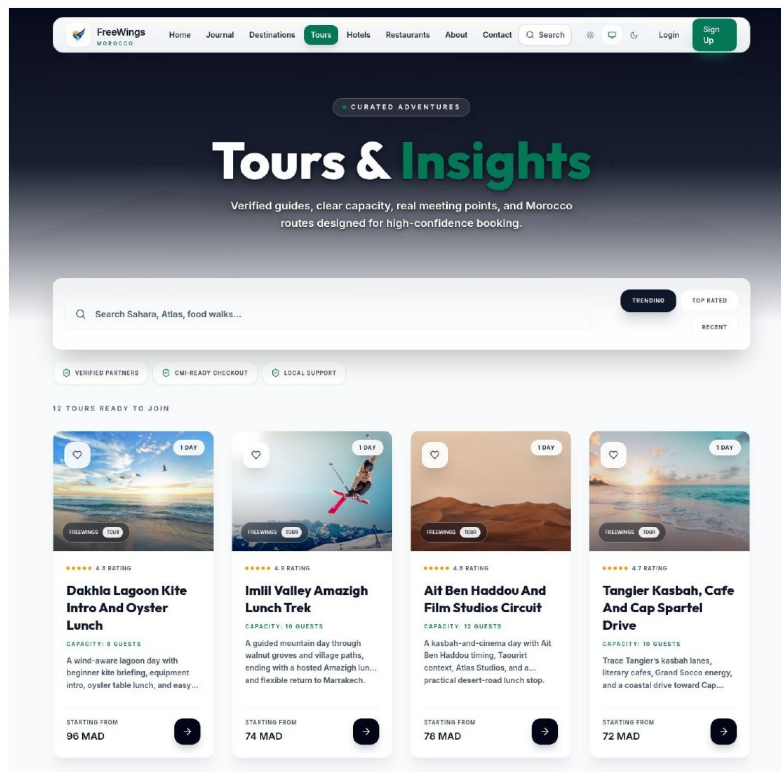
## Evidence 3 of 4



Destination detail connecting context, content and conversion actions.



## Evidence 4 of 4



Tour catalogue presenting bookable experiences.

# Value, proof and honest limitations

## Operational impact

FreeWings demonstrates the harder thing: a coherent multi-entity marketplace experience, not a single CRUD module wearing a nice UI. The supplied 94-page report and interface captures show a complete discovery journey, detailed service pages, authentication, unified search, both payment-return states, and editorial content — the full shape of a real product, not a slice of one.

## Verified evidence

- 94-page final-cycle report with architecture and UML
- Real UI captures for catalogues, detail pages, search and authentication
- Payment-return screens for pending and failed states
- Documented CMI signature, idempotency and reconciliation design
- Health, request telemetry and Web Vitals architecture

## Current limitations

- Developed by two contributors and never presented as a solo build.
- CMI was implemented in certification/preproduction mode, not proven live commerce.
- Bearer-token storage in localStorage should be hardened because it increases XSS exposure.
- No independent production load test or security audit is claimed.

## Current status

| STATUS                                                    | TEAM               | DEMO                                  | SOURCE                 |
|-----------------------------------------------------------|--------------------|---------------------------------------|------------------------|
| Two-person academic product<br>· Functional demonstration | Two-person project | Functional demo deployment<br>planned | Private project source |

### WHY THIS PROJECT MATTERS

## One coherent product experience across discovery, reservation and payment states.

This case study is designed to be inspectable: strong claims are connected to screenshots, source evidence or explicitly stated limitations.