

Yassir LMD Ordering Website

— Project Handover

Documentation

Table of Contents

- Yassir LMD Ordering Website — Project Handover Documentation
 - Table of Contents
 - Introduction
 - Live URLs
 - Business Overview
 - Purpose & Strategic Value
 - Target Segments
 - Key KPIs
 - Technical Architecture
 - Tech Stack
 - Architecture Diagram
 - Core Features & User Flows
 - Authentication
 - Restaurant Discovery
 - Menu & Cart Management
 - Order Processing
 - Address Management
 - Project Structure
 - Key Components
 - API Integrations
 - Security Implementations
 - Monitoring & Analytics
 - Setup & Environment Management
 - Local Setup
 - Environments
 - Secrets/Keys Required
 - Testing & Quality Assurance
 - Deployment Process
 - Workflow
 - Environments
 - Known Limitations & Future Improvements

- [Troubleshooting](#)
 - [Contribution Guidelines](#)
-

Introduction

The **Yassir LMD Ordering Website** is a multilingual, SEO-optimized food delivery web application built with **Next.js**. It allows users to discover nearby restaurants, browse menus, place and track orders, and manage personal delivery details. It is part of Yassir's broader Last Mile Delivery (LMD) product offering and complements the existing mobile experience.

Live URLs

- **Production:** <https://delivery.yassir.io/> (<https://delivery.yassir.io/>)
- **QA:** <https://qa-delivery.yassir.io/> (<https://qa-delivery.yassir.io/>)
- **Staging:** <https://stg-delivery.yassir.io/> (<https://stg-delivery.yassir.io/>)

Business Overview

Purpose & Strategic Value

This platform is central to Yassir's omnichannel ordering strategy, enabling:

- Increased reach by targeting desktop and web-first users
- Improved accessibility for users without smartphones
- Enhanced order management and discoverability via SEO

Target Segments

- Urban residents familiar with food delivery services
- Users preferring desktop/laptop ordering
- New customers exploring Yassir through search

Key KPIs

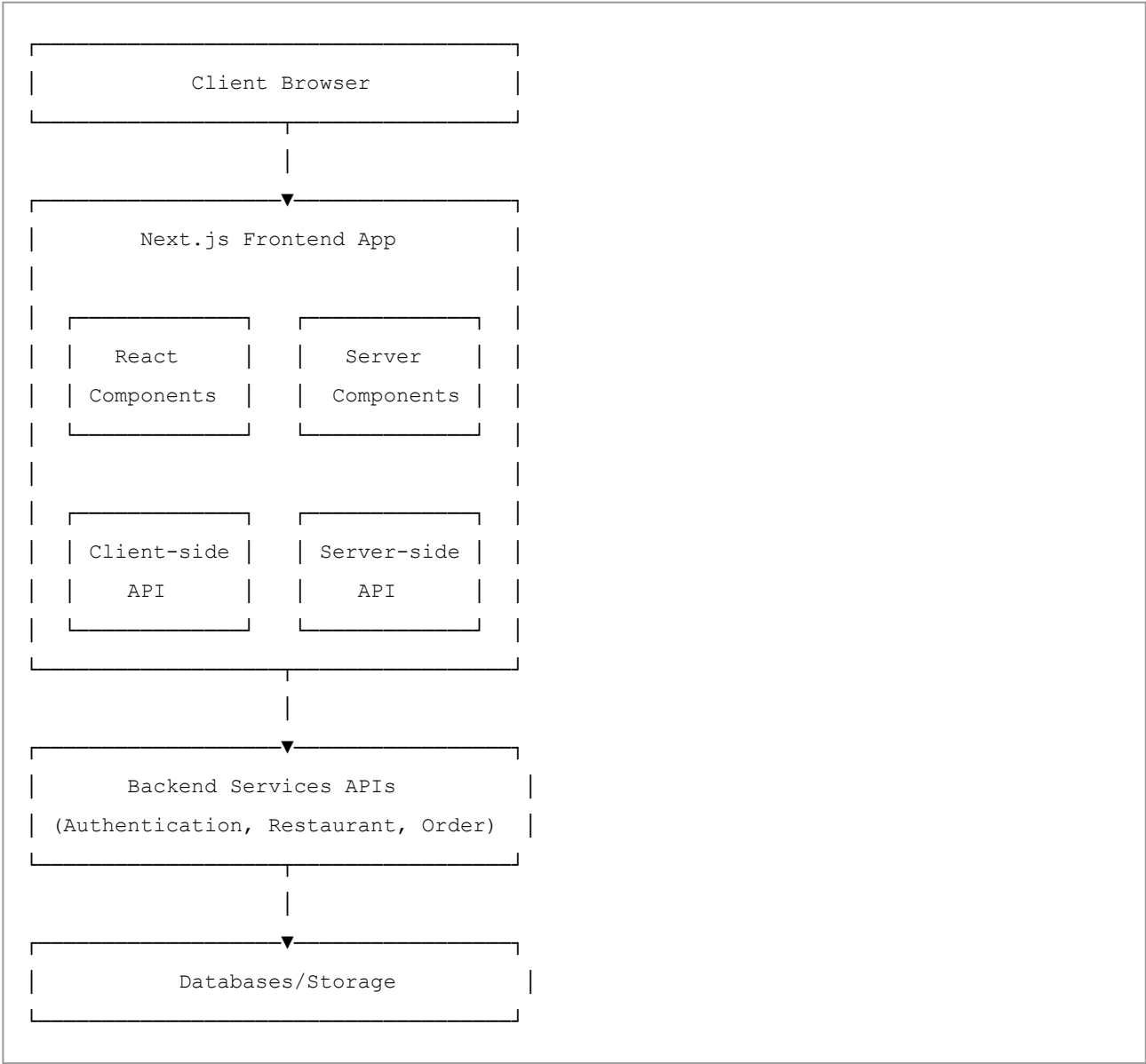
- Conversion rate
- Session duration & bounce rate
- Customer retention
- Average order value
- Order fulfillment time

Technical Architecture

Tech Stack

Area	Tech/Tools
Frontend Framework	Next.js (App Router)
Styling	Tailwind CSS
Component Library	Mantine, Custom Components
Design System	SEFAR
Forms & Validation	React Hook Form + Zod
Internationalization	next-intl
State Management	React Context API + Custom Hooks
Error Tracking	Sentry
User Analytics	Firebase, PostHog
Feature Flags	LaunchDarkly
Testing	Jest (Unit), Playwright (E2E)

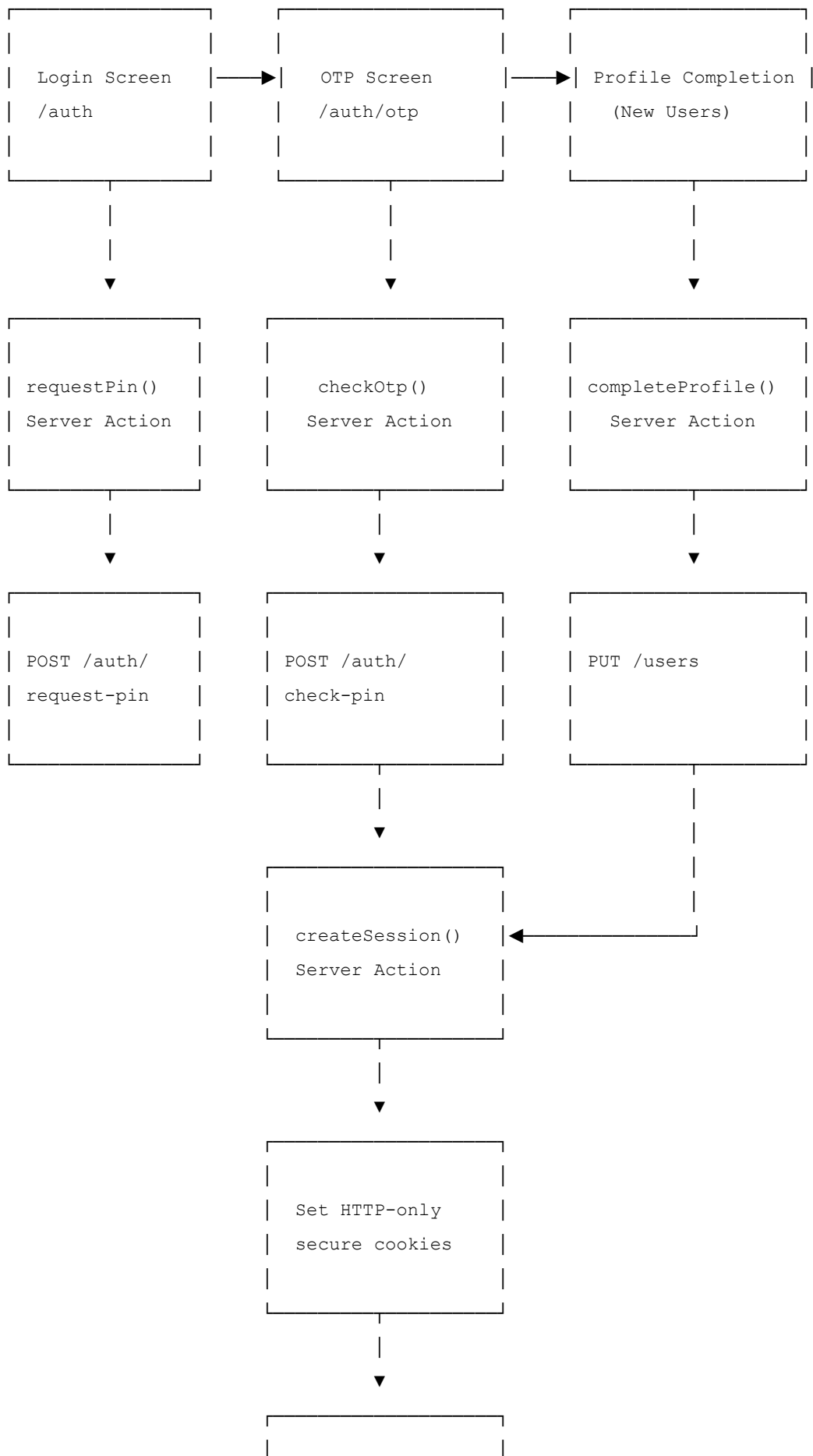
Architecture Diagram



Core Features & User Flows

Authentication

- OTP-based login via phone number
- Profile and session management



Redirect to
authenticated area

Restaurant Discovery

- Geolocation and manual location input
- Filters by cuisine, delivery time, rating
- Promotion banners and featured listings

Menu & Cart Management

- Dynamic menu browsing by categories
- Item-level customization and modifiers
- Cart summary and item edit flow

Order Processing

- Checkout with saved or new address
- Payment selection (COD currently)
- Order confirmation and backend sync

Address Management

- Address CRUD
- Google Maps and Places API integrations

Project Structure

```

src
├── __mocks__
├── app
│   ├── (sitemap)                // Sitemap routes
│   │   └── sitemap_index.xml
│   ├── [locale]
│   │   ├── [...not_found]        // 404 page
│   │   ├── auth
│   │   │   ├── otp
│   │   │   └── profile
│   │   ├── checkout
│   │   ├── mobile-app-redirect
│   │   ├── profile
│   │   │   └── addresses
│   │   └── restaurants
│   │       ├── (restaurants)
│   │       └── [restaurantId]
│   │           └── [restaurantName]
│   ├── actions                  // Server Actions
│   └── api                      // API routes
│       └── address-autocomplete
├── components                   // Reusable components
│   ├── AddressSearch
│   ├── AddressesList
│   ├── AllRestaurants
│   ├── AsyncWithLDAPProvider
│   ├── AuthCard
│   ├── BackButton
│   ├── Button
│   ├── Carousel
│   ├── Cart
│   ├── Checkout
│   ├── Common
│   ├── CreateAddress
│   ├── Dropdown
│   ├── DropdownButton
│   ├── FoodCategoriesCarousel
│   ├── FoodItemCard
│   ├── Footer
│   ├── Header
│   ├── Icon
│   ├── ImageLoader
│   └── LanguageSelect

```

- | |─ Link
- | |─ ListingCarousal
- | |─ LoadingComponents
- | |─ LocationSelect
- | |─ MapView
- | |─ MenuSearchFallback
- | |─ MobileBackdrop
- | |─ Modal
- | |─ NotFoundRestaurants
- | |─ OtpForm
- | |─ PageLoadLogger
- | |─ ProfileCard
- | |─ PromotionBanner
- | |─ RestaurantCard
- | |─ RestaurantDetailsHeader
- | |─ RestaurantFoodSection
- | |─ RestaurantListingFilters
- | |─ RestaurantMenuHeader
- | |─ RestaurantOffers
- | |─ RestaurantsSection
- | |─ ScrollBehaviorLogger
- | |─ SearchBar
- | |─ Shared
- | |─ SocialButton
- | |─ StoreDownloadButton
- | |─ Toast
- | |─ Typography
- |─ constants
- |─ context
- |─ hooks
- |─ lib
 - | |─ api
 - | |─ events
 - | |─ firebase
 - | |─ ldServer
 - | |─ mongodb
 - | |─ sitemap
- |─ styles
 - | |─ fonts
- |─ tests
- |─ utils

Key Components

Shared/Utility Components:

- Modal, Drawer, Toast, Button, Typography

Feature-Specific Components:

- FoodItemCard – menu detail modals
- CartCheckout, RestaurantCard, AddressSearch

Tracking & Logging Components:

- PageLoadLogger, ScrollBehaviorLogger

API Integrations

Module	Endpoint	Purpose
Auth	/auth/request-pin, /auth/check-pin	User login via OTP
User	/users	Profile fetch & update
Restaurants	/restaurant/list, /get/one-restaurant	Discovery & detail pages
Menu	/menu, /get/products	Category & item data
Orders	/orders/cart/*, /orders/checkout	Cart, checkout, order submission
Locations	Google Maps APIs	Geolocation and reverse geocoding

Security Implementations

- Enforced HTTPS and HSTS headers
- Content Security Policy
- HTTP-only cookies for sessions
- Input and schema validation (Zod)
- Rate-limiting via backend

Monitoring & Analytics

Tool	Use Case
Sentry	Error tracking and performance
Firebase	User behavior and funnel tracking
PostHog	Session replays and heatmaps
LaunchDarkly	A/B testing and feature toggles

Setup & Environment Management

Local Setup

```
git clone <repo-url>
yarn install
cp .env.local.example .env.local
yarn dev
```

Environments

- `.env.local` – Development
- `.env.qa` – QA environment
- `.env.staging` – Preprod testing
- `.env.prod` – Production deployment

Secrets/Keys Required

- Google Maps key
- MongoDB URI
- Firebase credentials
- LaunchDarkly client/server IDs
- Sentry DSN
- JWT/Session secret

Testing & Quality Assurance

Type	Tools	Notes
Unit	Jest	Run: <code>yarn test</code>
E2E	Playwright	Setup via <code>tests/</code> folder
Linting	ESLint + Prettier	Automated with pre-commit hooks

Deployment Process

Workflow

1. Push to feature branch
2. Open PR against `staging`, `preprod`, or `main`
3. GitHub Actions triggers build
4. Docker image built and pushed
5. Ship repo triggers Kubernetes deployment

Environments

- QA → `preprod` branch
- Staging → `staging` branch
- Production → `main` branch

Known Limitations & Future Improvements

- Order tracking UI not yet implemented
- Improved geolocation accuracy needed
- Offline support via Service Worker (planned)
- LCP optimization for mobile devices
- International payment gateway integration

Troubleshooting

Problem	Solution
API not responding	Check environment variables and network
Build fails	Validate TypeScript and Zod schemas
Missing translations	Check locale files under <code>public/locales/</code>
LaunchDarkly not toggling	Check environment ID and SDK key configuration

Contribution Guidelines

- Feature branches from `staging`
- PRs must include changelog summary
- Include relevant unit or E2E tests
- Tag appropriate reviewers based on ownership
- Update related documentation in `/docs`

For further assistance, please reach out to the **Yassir LMD Core XP team**.