Here’s the **full, polished Product Specification Document** for **Lama Group** — ready for you to copy into Google Docs, Word, or any design tool and export as a branded PDF.
I’ve included the **Deployment Roadmap** at the end so you can go straight from planning to execution.

Product Specification Document — Lama Group

---

1. Product Overview

Lama Group is a multi-service platform designed to unify and streamline operations for:

- **Store Leasing & Management**
- **Fuel Branding & Supply**
- **Wholesale Distribution**
- **Construction & Renovation**

The platform integrates **customer‑facing booking tools**, **internal workflow management**, and **brand‑aligned content delivery** to improve efficiency, trust, and conversion rates.

---

2. Objectives

- Provide a **centralized digital hub** for all Lama Group services.
- Enable **service‑specific booking workflows** with designated staff.
- Integrate **WordPress CMS** for content management and SEO.
- Ensure **responsive, brand‑consistent UI/UX** across devices.
- Support **scalable integration** with AI tools and third‑party APIs.

---

3. Key Features

Customer‑Facing

- **Service Landing Pages** with tailored CTAs.
- **Booking System**:
  - Date/time selection.
  - Automatic routing to designated staff email.
  - Approval workflow with meeting link generation.
- **Newsletter Signup** stored in WordPress.
- **Responsive Design** (Next.js + Tailwind CSS).

Internal Tools

- **Admin Dashboard** for booking management.
- **Staff Availability Management**.
- **Automated Email Notifications** (Nodemailer + SMTP).
- **Analytics Dashboard** (traffic, conversions, bookings).

---

4. Technical Requirements

- **Frontend**: Next.js 13+, Tailwind CSS, TypeScript.
- **Backend**: Next.js API routes, Node.js.
- **CMS**: WordPress (AWS Lightsail, subdomain `cms.lama.group`).
- **Database**: MySQL (WordPress), optional PostgreSQL for custom data.
- **Email**: SMTP (single sending account, dynamic recipients).
- **Hosting**:
  - Next.js → Vercel or AWS Amplify.
  - WordPress → AWS Lightsail (Linux/Unix, Bitnami WP image).
- **Security**:
  - HTTPS via Let's Encrypt.
  - Token-based booking approval links.
  - CORS configuration for API calls.

---

5. User Personas

- **Business Owner**: Oversees all service lines, reviews analytics.
- **Service Manager**: Handles bookings for their service category.
- **Customer**: Books strategy sessions, subscribes to newsletter.
- **Marketing Team**: Updates content via WordPress CMS.

---

6. Use Cases

1. **Customer books a strategy session** for “Fuel Branding & Supply”:
   - Clicks CTA → `/book` prefilled with service + staff email.
   - Selects date/time, enters details, submits.
   - Staff receives email with “Accept” link.
   - On acceptance, customer gets meeting link.
2. **Marketing team updates homepage hero slider** via WordPress.
3. **Customer subscribes to newsletter** → stored in WP database.

---

7. Integration Details

- **WordPress REST API**:
  - Public endpoints for content.
  - Authenticated endpoints for newsletter data.
- **SMTP Email**:
  - One sending account (e.g., `bookings@lama.group`).
  - Dynamic `to` address based on service mapping.
- **Booking Approval Workflow**:
  - Token stored in in-memory store or DB.
  - Expiry after 24 hours.
- **Analytics**:
  - Google Analytics 4 integration.
  - Event tracking for bookings and newsletter signups.

---

8. Performance & Scalability

- **Static site generation** for marketing pages.
- **Server-side rendering** for booking pages.
- **Lazy loading** for images and non-critical scripts.
- **Scalable hosting** on AWS Lightsail and Vercel.

---

9. Risks & Mitigation

Risk	Impact	Mitigation

Email delivery issues	Booking confirmations fail	Use verified SMTP with SPF/DKIM
CORS errors	API calls blocked	Configure API routes to allow `lama.group` and `cms.lama.group`
Downtime	Loss of bookings	Enable Lightsail snapshots and Vercel rollbacks
Security breaches	Data loss or leaks	Enforce HTTPS, strong passwords, and regular updates

---

10. Deployment Roadmap

Phase 1 — Infrastructure Setup

- Create AWS Lightsail instance with WordPress blueprint.
- Assign static IP and point `cms.lama.group` DNS to it.
- Install SSL via Let's Encrypt.
- Deploy Next.js site to Vercel (or AWS Amplify) at `lama.group`.

Phase 2 — Core Features

- Implement hero slider with service-specific CTAs.
- Build `/book` page with prefilled service + email.
- Create `/api/book-session` and `/api/accept-booking` routes.
- Configure SMTP for booking emails.

Phase 3 — CMS Integration

- Connect Next.js to WordPress REST API for content.
- Implement newsletter signup storing data in WP.

Phase 4 — Testing

- Test booking flow end-to-end (customer → staff → acceptance → meeting link).
- Test responsiveness across devices.
- Test API calls from production domains.

Phase 5 — Launch

- Finalize DNS and SSL for both domains.
- Enable analytics tracking.

- Take Lightsail snapshot for backup.

If you want, I can now **add a one-page executive summary** version of this doc so you have something quick to send to stakeholders without all the technical detail — would you like me to prepare that next?