

MMOptiBuilds

Master Project Handbook

Systems / Studio / Shared Operations

Authoritative product, design, engineering, launch, and AI-orchestration plan
Version 1.0 • 11 August 2026

Two kinds of technical work. One standard of care.

How to use this handbook

A new explicit owner instruction has highest authority. Then use the decision register, source of truth, task brief, and only then the historical interview. Legal and tax sections are implementation briefs, not professional advice.

Contents

Open in Google Docs or Word and update the table of contents.

Executive summary

MMOptiBuilds will be a founder-led technology brand with two divisions on one permanent domain:

- **MMOptiBuilds Systems** sources and supplies custom PCs, workstations, business systems, and exact-spec enterprise hardware.
- **MMOptiBuilds Studio** creates conversion-focused websites and bespoke premium digital experiences.

The root page is a clear brand gateway, not an obscure splash screen. Systems and Studio share infrastructure but not a visible design system. Systems should feel engineered, precise, dark, and technical. Studio should feel editorial, expressive, human, and art-directed.

Business model

Systems holds no stock. The customer submits requirements, MMOptiBuilds confirms distributor availability and terms, and then issues a tailored quote. The founder may assemble and test PCs. Enterprise customers must know the hardware they need; enterprise fulfillment is sourcing and delivery only.

Studio accepts both focused lower-budget projects and premium bespoke work. Every project retains the same baseline: accessibility, mobile-first usability, clear conversion, technical SEO, security, and maintainability. Clients own their domains and hosting. MMOptiBuilds may deploy once using temporary collaborator access, then removes access and provides a handoff. A 30-day bug-fix warranty excludes new features and content changes.

Product model

The public experience contains unique service pages and conversion flows for each search intent. All inquiries feed one owner-only dashboard with separate Systems and Studio workspaces, a shared inbox, quotes, contacts, analytics, and settings.

Zero-spend launch architecture

- Next.js 16 App Router, TypeScript, React 19, Tailwind CSS 4
- Cloudflare Workers and OpenNext for commercial production
- Supabase Postgres, Auth, and Row Level Security
- Cloudflare Turnstile, DNS, TLS, Web Analytics, and Email Routing
- GitHub private repository and Actions within free quotas
- Resend free tier only if visitor acknowledgements or owner authentication emails are needed in production

Vercel Hobby is not selected for the commercial production site because its documented terms restrict the plan to non-commercial personal use.

Release strategy

Launch a complete, narrow foundation first: root gateway, Systems overview, Studio overview, focused quote forms, About, Contact, one honest case study, and the owner dashboard. Add rich 3D, advanced motion, builders, and deeper service pages only after the core journey is fast, accessible, truthful, and measurable.

Principal risks

- Publishing unverified claims, policies, warranties, or tax language
- Building too much motion before proving mobile performance
- Using Vercel Hobby commercially
- Treating free Supabase email as production SMTP
- Letting agents edit the same files or install overlapping visual libraries
- Launching without a verified privacy notice, terms, quote rules, warranty rules, and tax review

The implementation plan addresses each risk with explicit gates.

Source of truth

Identity

- Brand: mmoptibuilds, lowercase and one word in running text.
- Public name may be used; title: Founder & Technical Director.
- Primary launch market: India, with Bengaluru as the strongest local context.
- Permanent domain: mmoptibuilds.com; no additional domains are planned.
- Brand posture: founder-visible for trust, organization-led for scale.

Site structure

```
mmoptibuilds.com/
├── systems/
├── studio/
├── about/
├── contact/
└── admin/   private, noindex
```

The root introduces the parent brand and immediately explains both divisions. Systems and Studio use distinct design systems, language, navigation emphasis, content models, motion signatures, and SEO strategies. Shared code is limited to invisible foundations such as forms, validation, data access, accessibility utilities, analytics, and testing.

Systems

Offer gaming PCs, creator and professional workstations, business systems, servers, storage, enterprise SAS/HDD/SSD, network switches, SFP modules, racks, and exact-spec hardware. Do not offer PC optimization as a standalone service.

No inventory is held. Products are sourced from Indian distributors after a request. Public prices are prohibited because availability, warranty, tax, logistics, and distributor cost must be confirmed. MMOptiBuilds may source and deliver, or source, assemble, test, and deliver a complete PC. Enterprise work is sourcing and delivery only and begins with an exact specification or part number.

Studio

Serve small businesses, startups, and organizations replacing outdated websites. Focused projects prioritize clarity and conversion. Premium projects add bespoke strategy, art direction, motion, custom assets, and deeper storytelling. Both modes remain mobile-first, accessible, SEO-ready, secure, and maintainable.

Clients own hosting and domain accounts. MMOptiBuilds uses collaborator access, performs a one-time deployment, removes unnecessary access, and hands over documentation. Static delivery is the default. Add a CMS only for a real editorial need. The working warranty is 30 days for reproducible implementation defects; content changes and new functionality are new scope.

Operations

One owner-only dashboard contains Overview, Systems, Studio, Shared Inbox, Quotes, Contacts, Analytics, and Settings. Visitors submit as guests. Launch does not require email verification or an account. Turnstile, server validation, honeypots, throttling, deduplication, and manual review reduce abuse without adding conversion friction.

Trust rules

- Never invent testimonials, reviews, clients, awards, certifications, stock, distributor relationships, savings, delivery times, or performance figures.
- Label concepts, experiments, and generated media honestly.
- Publish a case study only when ownership, scope, and claims are documented.
- Do not expose personal phone numbers, private email, family history, finances, credentials, supplier rates, or infrastructure details.

Current owned-work decision

Do not export old hosted sites into this codebase. Build MMOptiBuilds from scratch. Coldharbour may become an accurately labelled independent-project case study. Studio Effects and the Glass Dreamscape builder may inform interaction and structure, but their source is not a dependency. Other sites remain references, experiments, or separate products unless later verified.

Decision register

ID	Decision	Status	Rationale
D-001	Use `mmoptibuilds.com` permanently	Locked	One brand, no future domain purchases
D-002	Use `/systems`, `/studio`, and `/admin` paths	Locked	Seamless parent brand with distinct sections
D-003	Use one repository and one Next.js application initially	Locked baseline	Lowest operational burden; preserves route-level separation
D-004	Keep two visible design systems	Locked	Different audiences and buying contexts
D-005	Share backend, forms infrastructure, analytics, and owner dashboard	Locked	Centralized operations without visual sameness
D-006	Do not show public prices	Locked	Quotes depend on requirements and current sourcing
D-007	Hold no hardware inventory	Locked	Source from Indian distributors after inquiry
D-008	Founder assembles custom PCs when assembly is purchased	Locked	Current delivery model
D-009	Enterprise is exact-spec sourcing and delivery only	Locked	Avoids unsupported consulting and deployment promises
D-010	Studio supports focused and premium projects	Locked	Accept broad budgets while protecting baseline quality
D-011	Client owns hosting and domain	Locked	Reduces long-term operational and legal exposure
D-012	One-time deployment and documented handoff	Locked	Clear responsibility boundary
D-013	Thirty-day implementation bug warranty	Working policy	Must be reviewed in final contract language
D-014	No visitor accounts at launch	Locked baseline	Better conversion and less auth/email complexity
D-015	Owner-only Supabase authentication	Locked baseline	Dashboard requires controlled access
D-016	Cloudflare Workers/OpenNext for production	Locked baseline	Commercially usable zero-spend path
D-017	Vercel only for eligible previews or paid future use	Locked baseline	Hobby plan is non-commercial
D-018	Supabase for Postgres/Auth/RLS	Locked baseline	Shared data and practical free start
D-019	Cloudflare Turnstile and Web Analytics	Locked baseline	Free abuse control and privacy-conscious analytics
D-020	Resend only when arbitrary transactional email is required	Conditional	Supabase built-in mail is demonstration-only
D-021	Build from scratch; do not export old sites	Locked	Export effort is not justified
D-022	No fake testimonials or synthetic proof	Locked	Trust and legal safety
D-023	AI-generated launch visuals may be used if labelled internally and replaceable	Locked	Enables progress without false claims
D-024	Add advanced motion/3D only after performance gates	Locked	Premium quality depends on restraint
D-025	Production release requires	Locked	Prevents accidental

Open decisions that must not block foundation work

- Final public legal name and business entity details
- GST registration and invoice requirements
- Approved privacy, terms, cancellation, refund, procurement, and warranty wording
- Exact owner email destination and support aliases
- Confirmed supplier, logistics, and manufacturer-warranty language
- Which real projects have permission and evidence for public case studies
- Whether Resend will be configured at launch

Project brief

Objective

Create a premium, conversion-led commercial website that makes MMOptiBuilds credible in two categories without making it feel like two unrelated companies. The experience should be strong enough for design-award consideration, but awards are an ambition rather than a usability exception.

Audiences

Systems

- Buyers who need a gaming PC or workstation but want expert specification help
- Creators and professionals with workload requirements
- Small and medium businesses sourcing systems
- Procurement teams that already know enterprise part numbers and quantities

Studio

- Small businesses needing a credible first website
- Startups preparing to launch, sell, hire, or raise
- Established businesses replacing an outdated website
- Premium clients seeking distinctive art direction and interaction

Primary outcomes

1. A qualified visitor understands the relevant offer in under ten seconds.
2. The visitor can choose the right inquiry path without decoding jargon.
3. The owner receives a structured brief that is useful for research and quoting.
4. The site establishes trust without fabricated proof.
5. The owner can review, classify, quote, and follow up from one private dashboard.

Success measures

- Qualified inquiry completion rate by route and division
- Form start-to-completion rate and field-level abandonment
- Organic impressions and qualified clicks by service intent
- Response time from new inquiry to first manual reply
- Accepted quote rate, tracked manually at first
- Core Web Vitals at the 75th percentile: LCP $\leq$ 2.5 s, INP $\leq$ 200 ms, CLS $\leq$ 0.1
- Zero critical accessibility or security findings at launch

Out of scope for launch

- Public ecommerce and checkout
- Live stock or public component pricing
- Automated hardware compatibility guarantees
- Enterprise installation, rack work, network deployment, or managed infrastructure
- Client portals, visitor accounts, subscriptions, or managed hosting
- A generic content-management system for every page
- Multiple production repositories or microfrontends
- Heavy 3D on every page

Requirements

Functional

- Root gateway with clear Systems and Studio paths
- Independent section navigation and route metadata
- Systems guided build inquiry and exact-spec enterprise RFQ
- Studio discovery brief with project mode, goals, audience, assets, timing, and constraints
- Shared contact form
- Server-side validation, Turnstile validation, throttling, honeypot, and normalized storage
- Submission confirmation screen with reference ID
- Owner email notification to a verified mailbox
- Owner login and dashboard with filters, statuses, notes, quote records, follow-up dates, and CSV export
- Sitemap, robots, canonical URLs, Open Graph cards, and JSON-LD where factually supported
- Consent records and privacy-notice version stored with every inquiry
- Error, empty, offline, retry, and success states

Non-functional

- Mobile-first from 320 px upward
- Keyboard-complete navigation and forms
- WCAG 2.2 AA target
- Reduced-motion and data-saving fallbacks
- Semantic HTML before ARIA
- Core Web Vitals budgets enforced in QA
- No client-exposed service keys or dashboard data
- RLS on all exposed Supabase tables
- Idempotent form submission and audit history
- Dependable route refresh and deep linking
- No console errors, broken links, missing assets, or mixed-content requests

Content

- Clear statement that systems are sourced to requirements and subject to availability
- Clear enterprise sourcing-only boundary
- Clear Studio client-ownership and handoff model
- Honest case-study labels
- No public pricing or guaranteed outcomes
- Legal copy uses plain language and is professionally reviewed before commercial launch

Quality gates

Each route requires copy review, responsive review, keyboard review, reduced-motion review, metadata review, analytics verification, and visual-regression evidence. A task is not done because it renders once.

Information architecture

Launch routes

Route	Purpose	Primary CTA
`/`	Parent proposition and division choice	Explore Systems / Explore Studio
`/systems`	Systems overview and buyer routing	Describe your requirements
`/systems/gaming-pcs`	Gaming build intent	Request a gaming build
`/systems/workstations`	Creator/pro workload intent	Describe your workload
`/systems/enterprise-hardware`	Exact-spec procurement	Request a sourcing quote
`/studio`	Studio overview and work modes	Start a website brief
`/studio/business-websites`	First/serious business website	Discuss your website
`/studio/startup-websites`	Startup launch credibility	Start a launch brief
`/studio/website-redesign`	Outdated-site replacement	Request a site review
`/studio/work`	Verified portfolio and concepts	View a case study
`/studio/work/coldharbour`	Honest independent-project case study	Discuss a similar project
`/about`	Organization, approach, and founder	Contact MMOptiBuilds
`/about/story`	Deeper professional story	Start a conversation
`/contact`	Choose an inquiry type	Send inquiry
`/privacy`	Data use and rights	Contact privacy owner
`/terms`	Website and inquiry terms	Contact MMOptiBuilds
`/admin`	Owner-only operations	Sign in

Later routes

Add `/systems/servers`, `/systems/storage`, and `/systems/networking` only after each has authoritative copy and distinct search value. Add `/systems/builder` only as a guided brief, never as a static-price compatibility promise.

Navigation

The global mark returns to `/`. On root, both divisions have equal prominence. Inside a division, the local navigation leads; a discreet switcher exposes the sibling division. Contact remains globally available. Admin is never in public navigation.

URL rules

- Lowercase, descriptive, hyphenated slugs
- One canonical URL per intent
- No date or technology name in evergreen service URLs
- No indexable filter or form-step URLs
- Redirect retired routes permanently and update internal links

User journeys

Custom PC or workstation

6. Arrive on a use-case page.
7. Understand the requirement-led, no-stock model.
8. Choose workload, existing preferences, private budget range, location, assembly need, and timing.
9. Submit after server-side Turnstile validation.
10. Receive an on-screen reference ID and response expectation without a guaranteed deadline.
11. Owner researches availability and prepares a tailored quote.

Enterprise sourcing

12. Arrive through a part-number or enterprise-intent query.
13. See the sourcing-and-delivery boundary immediately.
14. Provide manufacturer, exact part/SKU, quantity, condition, location, required date, warranty, tax/PO needs, and documents.
15. Owner validates the specification and suppliers manually before quoting.

Focused website

16. Arrive on business website or redesign page.
17. See the practical outcome: credibility, clarity, mobile usability, and conversion.
18. Explain organization, audience, primary action, required pages, current assets, domain/hosting state, and timing.
19. Owner reviews scope and replies manually with questions or a quote.

Premium website

20. Enter through Studio positioning, work, or referral.
21. Explore art direction and case-study reasoning without blocked navigation or forced motion.
22. Select bespoke mode and describe ambitions, differentiation, content, assets, and approval process.
23. Owner qualifies fit before proposal and discovery.

Owner operations

24. Receive a verified owner notification.
25. Open the authenticated dashboard.
26. Review full submission, consent version, source route, spam signals, and history.
27. Set status, priority, notes, quote amount privately, and next follow-up date.
28. Reply manually from email; record the action.
29. Close as won, lost, spam, duplicate, or archived.

Failure journeys

Every form preserves non-sensitive input after a recoverable error, identifies the field that needs attention, supports retry, and never creates duplicate inquiries from repeated taps.

Systems design system

Character

Engineered, calm, exact, and physically grounded. The experience should suggest instrumentation and industrial materials without turning into a gaming cliché.

Visual grammar

- Near-black graphite foundation with warm metal, restrained electric cyan, and precise signal green
- Condensed display typography for technical headings; neutral sans for reading; tabular numerals for specifications
- Grids, datum lines, calibrated labels, exploded details, restrained optical glass, and meaningful status signals
- Photography and renders focus on components, assembly, thermals, material texture, and finished systems
- Avoid cyberpunk purple, noisy HUD overlays, fake benchmark dashboards, and green-on-black hacker motifs

Layout

- Dense information is chunked into clear cards and comparison rows
- Critical trust and sourcing boundaries appear near conversion points
- Mobile layouts use single-column progression, short labels, and large tap targets
- Specifications use native tables only when comparisons are genuinely tabular

Motion

- Motion explains physical relationships, state changes, or process
- Short mechanical easing for controls; slow parallax only in non-essential art layers
- Optional WebGL hero is lazy-loaded after meaningful content and replaced with a static image for reduced motion, weak GPUs, or data saving
- No ball pit; no cursor-dependent interaction; no scroll hijacking

Voice

Direct and factual: “Built around your workload,” “Sourced after confirmation,” and “Exact-spec enterprise procurement.” Avoid “ultimate,” “unbeatable,” and unverified superlatives.

Studio design system

Character

Editorial, inventive, assured, and human. It should look art-directed rather than assembled from a fashionable agency template.

Visual grammar

- Warm paper or soft mineral background, ink typography, one saturated seasonal accent, and occasional monochrome sections
- Expressive display face paired with a highly readable text sans; generous optical sizing and deliberate line breaks
- Asymmetric editorial composition, project crops, margin notes, typographic rhythm, and controlled negative space
- Motion, texture, and depth support each story rather than becoming a permanent effect layer
- Avoid generic purple gradients, endless glass cards, generic marquee walls, and fake award laurel graphics

Layout

- Copy and proof lead before spectacle
- Focused-project pages are simpler and more direct than premium case studies
- Work pages explain the problem, constraints, decisions, implementation, and honest result
- Mobile composition is designed independently, not merely stacked from desktop

Motion

- Page transitions preserve orientation and never delay navigation
- Reveals follow reading order; hover is additive and never required
- Reduced-motion mode preserves composition and meaning without large transforms
- Video and WebGL require posters, captions where relevant, lazy loading, and a still-image fallback

Voice

Specific and conversational: “A clearer website for the next stage of your business.” Avoid agency fog such as “we craft transformative digital ecosystems.”

Content and copy framework

Parent homepage

Working headline: Two kinds of technical work. One standard of care.

Working introduction: MMOptiBuilds sources systems around real workloads and builds websites around real business goals. Choose the path that matches what you need.

Systems card: “Custom PCs, workstations, and exact-spec enterprise hardware—researched and quoted after your requirements are clear.”

Studio card: “Conversion-focused websites for businesses, startups, and teams ready to replace an outdated presence.”

Systems messages

- Built around the workload, not a shelf bundle.
- Availability, warranty, and final price are confirmed before the quote.
- Enterprise sourcing starts with an exact part number or specification.
- Assembly and testing are available for complete PCs; enterprise supply is sourcing and delivery only.

Studio messages

- Clear enough to convert. Distinctive enough to remember.
- Focused sites keep scope lean, not quality low.
- Premium work adds bespoke strategy, art direction, motion, and original assets.
- Your domain and hosting stay in your control.

Proof hierarchy

30. Verified work and explainable decisions
31. Real process artifacts and technical evidence
32. Founder expertise described without inflated claims
33. Clearly labelled concepts or experiments
34. No placeholder testimonials presented publicly

Copy rules

- Lead with the customer situation, not the company biography.
- Use concrete nouns and verbs.
- State boundaries near the CTA.
- Do not guarantee rankings, conversion, availability, price, shipping date, warranty coverage, or savings.
- Replace every unresolved fact with a review marker in the private content system; never ship bracketed placeholders.

SEO strategy

Principle

Separate SEO by search intent, not by visual identity alone. Every indexable page must answer a distinct query with original, useful content and a conversion path. Do not generate thin city pages, duplicate service pages, or mass AI copy.

Intent map

Cluster	Example intent	Target route
Custom gaming PC	custom gaming PC builder/supplier in Bengaluru or India	<code>`/systems/gaming-pcs`</code>
Workstation	workstation for editing, 3D, AI, CAD, or engineering	<code>`/systems/workstations`</code>
Enterprise sourcing	source server/storage/network part by SKU	<code>`/systems/enterprise-hardware`</code>
Business website	website design for a small business	<code>`/studio/business-websites`</code>
Startup website	launch or investor-ready startup site	<code>`/studio/startup-websites`</code>
Redesign	replace an outdated business website	<code>`/studio/website-redesign`</code>
Brand	MMOptiBuilds identity and services	<code>`/`, `/about`</code>

On-page requirements

- Unique title, description, H1, canonical, Open Graph data, and useful body copy
- Descriptive internal links and breadcrumbs where hierarchy helps
- Accurate Organization, WebSite, Service, BreadcrumbList, and Article/CreativeWork JSON-LD only where facts support them
- Indexable text must not depend on canvas, animation, or client-only rendering
- Image dimensions, descriptive filenames, useful alt text, and responsive formats
- One meaningful page per intent; consolidate cannibalizing pages

Technical controls

- Generate sitemap from public routes; exclude `/admin`, form steps, previews, and internal filters
- `noindex` admin, staging, search results, and unapproved case studies
- Server-render primary metadata and content
- Monitor Search Console coverage, queries, CWV, canonical selection, and structured-data errors
- Use permanent redirects for retired URLs and preserve backlinks during future changes

Editorial roadmap

Publish only articles grounded in actual expertise: workload-to-hardware explanations, procurement checklists, website redesign diagnostics, accessibility decisions, and honest project breakdowns. Every article must have an accountable author, review date, sources where needed, and a relevant next step.

Technical architecture

Baseline

Use one private repository and one Next.js 16 App Router application. Route groups isolate public experiences without imposing a shared visual shell:

```
app/
├── (gateway)/
├── (systems)/systems/
├── (studio)/studio/
├── (legal)/
├── admin/
└── api/
```

components/shared contains only behavior and accessibility primitives. components/systems and components/studio own visible patterns. Each division has independent design tokens, fonts, art direction, motion configuration, and content schemas.

Runtime

- Cloudflare Workers through @opennextjs/cloudflare
- Node.js compatibility enabled only where required and supported
- Static generation for stable marketing pages
- Dynamic route handlers for inquiries, dashboard, and signed operations
- Server Components by default; Client Components only for interaction
- Supabase Postgres and owner Auth
- Cloudflare Turnstile verified server-side before accepting public submissions
- Cloudflare Web Analytics for baseline traffic and CWV signals

Environments

Environment	Purpose	Data rule
Local	Development and automated tests	Local Supabase or isolated test project
Preview	Review branch output	Synthetic data only
Production	`mmoptibuilds.com`	Real inquiries and owner access

Never share production secrets or personal inquiry data with AI tools, preview deployments, logs, analytics events, or screenshots.

Request flow

35. Browser submits normalized form data and Turnstile token.
36. Route handler checks content type, origin strategy, body size, Turnstile, honeypot, rate limit, and Zod schema.
37. A database function inserts the inquiry and immutable consent/audit facts.
38. A reference ID is returned; duplicate retries are idempotent.
39. A non-blocking owner notification is sent to a verified destination.
40. Dashboard queries remain behind authenticated RLS policies.

Why not a monorepo or microfrontends now

The visible design systems do not require deployment fragmentation. One application makes local development, forms, auth, SEO, analytics, testing, and deployment easier. Split only after measured organizational or runtime pressure, not aesthetic preference.

Hosting constraint

Do not deploy the commercial production site to Vercel Hobby. Cloudflare's official OpenNext adapter supports the App Router, route handlers, RSC, SSG, SSR, ISR, Server Actions, streaming, and middleware; avoid unsupported Node.js middleware assumptions. Use Vercel only for eligible previews or a paid plan.

Data model

Core tables

Table	Purpose	Sensitive fields
`contacts`	Normalized person or organization	email, phone if volunteered
`inquiries`	Shared envelope and workflow	message, source, status, assignment
`systems_briefs`	PC/workstation requirements	budget range, workload, location
`enterprise_rfqs`	Exact-spec procurement	part numbers, quantities, documents
`studio_briefs`	Website discovery	business goals, budget band, assets
`quotes`	Private proposal metadata	amount, tax, validity, status
`notes`	Owner-only internal notes	free text
`activities`	Append-only audit trail	actor, action, before/after summary
`consents`	Notice version and consent facts	timestamp, policy version, purpose
`attachments`	Metadata for approved files	object key, type, size, scan status

Status models

inquiries.status: new, reviewing, needs_information, qualified, quoted, won, lost, spam, archived.

quotes.status: draft, sent, accepted, declined, expired, withdrawn.

Security rules

- Enable RLS on every exposed table before inserting production data.
- Public users receive no direct table privileges. Public forms call a narrow server route or security-definer function with strict validation.
- Authenticated users still need owner-role policies; authentication alone is not authorization.
- Secret/service-role keys remain backend-only and are never prefixed NEXT_PUBLIC_.
- Keep public form payloads minimal and delete data according to an approved retention schedule.
- Attachments are private, allowlisted, size-limited, renamed, malware-scanned before use, and served through short-lived signed URLs.

Indexes

Index created_at, status, division, contact_id, follow_up_at, lowercased email, and exact enterprise part number where useful. Add indexes from measured queries, not guesswork.

Migrations

Every schema change is a versioned SQL migration reviewed in pull request. CI creates an isolated database or applies migrations to a disposable environment. Production changes are backed up and deployed before compatible application code when required.

Forms and conversion

Form families

Systems build brief

Collect use case, key applications/games, performance priorities, existing preferred parts, private budget range, assembly choice, delivery location, target timing, and free-form constraints. Do not imply instant compatibility or availability.

Enterprise RFQ

Collect organization, manufacturer, model, part/SKU, quantity, acceptable condition, technical requirements, delivery postcode, required date, warranty, tax invoice/PO needs, and optional procurement document.

Studio brief

Collect organization, project type, current URL, audience, primary conversion, required pages/capabilities, assets already available, domain/hosting state, budget band, timeline, decision makers, and a free-form description.

UX rules

- Begin with a small choice, then progressively disclose relevant fields.
- Show a persistent plain-language summary and allow edits before submission.
- Use real labels, descriptions, examples, and inline errors; never placeholder-only labels.
- Budget is private and expressed as a range or “not decided”; public prices remain absent.
- Never require account creation or email verification at launch.
- Preserve non-sensitive input after recoverable failures.
- Support keyboard, autofill, mobile input modes, and 44×44 px minimum targets.

Abuse controls

Validate Turnstile on the server, enforce body and attachment limits, use a honeypot and minimum-completion-time signal, normalize email, throttle by several signals, and record a privacy-safe abuse fingerprint. Do not rely on client validation, CAPTCHA alone, or IP address alone.

Conversion measurement

Track route view, meaningful CTA, form start, step completion, validation error category, successful submission, and reference screen. Never send names, email addresses, phone numbers, messages, part numbers, or budget details to analytics.

Response expectation

Use honest wording such as “We’ll review your requirements and reply by email.” Publish a timeframe only after the owner can reliably meet it.

Security, privacy, and legal baseline

This document is an implementation and review checklist, not legal advice. An Indian lawyer and chartered accountant should approve commercial policies, GST treatment, invoice language, warranties, cancellations, and data practices before launch.

Security controls

- Owner-only authentication with strong password, MFA when available, secure recovery, and short idle timeout
- Server-side authorization on every dashboard read and mutation
- RLS on all exposed Supabase tables
- Turnstile, validation, rate limits, idempotency, and request-size limits on public forms
- CSRF-resistant patterns for authenticated mutations, strict origin checks where applicable, and SameSite cookies
- CSP, HSTS after HTTPS validation, frame protection, MIME sniffing protection, and restrictive permissions policy
- Secrets only in managed environment storage; rotate any exposed credential immediately
- Dependency lockfile, automated audit, minimal permissions, and release-time security review
- Logs redact personal data and secrets; error messages reveal no internals
- Private attachments with allowlisted types, content inspection, random names, and signed access

Privacy

Collect only data needed to respond, source, quote, or deliver. State the purpose at collection, identify the business contact, record the notice version, provide correction/deletion/contact routes, define retention, and restrict access. Do not train models on inquiry data or send it to agents without explicit anonymization and a lawful reason.

India's Digital Personal Data Protection framework and its 2025 Rules require professional assessment of notices, consent, rights, security safeguards, breach handling, and retention for the actual business setup.

Commercial boundaries

- Quotes are subject to availability, supplier confirmation, tax, logistics, and written acceptance.
- Manufacturer warranties and MMOptiBuilds workmanship obligations must be distinguished.
- Enterprise supply excludes installation, architecture, compatibility certification, managed infrastructure, and on-site work unless separately agreed.
- Studio clients own domains, hosting, accounts, and production content. One-time deployment access is removed at handoff.
- The 30-day bug warranty covers reproducible deviations from approved scope; it excludes third-party outages, client changes, new features, content updates, abuse, and credentials.
- Never host or manage a client site after handoff unless a separate written service is introduced later.

Launch blockers

- Legal identity and registered address decision
- GST applicability and invoice workflow confirmed with a CA
- Approved Privacy Notice, Website Terms, Quote Terms, Hardware Supply Terms, Studio Agreement, Warranty Policy, and Cancellation/Refund language
- Support email routing tested
- Supplier and logistics claims verified
- Case-study permissions and factual proof recorded

Performance, accessibility, and motion

Budgets

Measure	Launch target
LCP	≤ 2.5 s at p75
INP	≤ 200 ms at p75
CLS	≤ 0.1 at p75
Initial route JS	Prefer ≤ 170 KB compressed; justify exceptions
Hero media	Prefer ≤ 500 KB mobile and ≤ 1 MB desktop
Fonts	Two families maximum per division; subset and self-host legally

Accessibility target

WCAG 2.2 AA. Use semantic landmarks and headings, visible focus, logical order, accessible names, contrast, error summaries, status announcements, captions/transcripts, and a complete keyboard path. Test at 200% zoom, narrow viewports, high contrast, reduced motion, and coarse pointer.

Motion policy

- Start from static, complete content.
- Use CSS for simple states, Motion for component/layout transitions, GSAP only for orchestrated editorial sequences, and WebGL only for a unique scene that survives a performance review.
- Honor `prefers-reduced-motion`; also reduce autoplay video and parallax for data-saving or constrained devices.
- Never trap scrolling, delay navigation, move focus unexpectedly, or make hover the only discovery method.
- Pause offscreen animation and clean up observers, timelines, canvases, and event listeners.

Asset loading

Responsive AVIF/WebP, correct intrinsic dimensions, lazy-loaded below the fold, priority only for the true LCP asset, and posters for video. Load 3D and shader libraries only on routes that use them, behind dynamic imports and a static fallback.

Verification

Use Lighthouse for regressions, Playwright for route and keyboard journeys, axe for automated accessibility signals, browser performance traces for heavy pages, and field CWV from Cloudflare Web Analytics. Automated checks supplement manual assistive-technology review.

Asset strategy

Launch approach

Temporary AI-generated visuals are allowed when original assets do not exist. They are never proof of a real product, client, team, award, facility, or result. Keep an internal manifest so every temporary item can be replaced without redesign.

Manifest fields

id, route, slot, filename, purpose, source, license, generated, truth_label, alt_text, dimensions, responsive_variants, compression, replacement_owner, replacement_deadline, approval_status.

Systems assets

Prefer real macro component photography, finished-build photography, assembly details, cable and thermal details, packaging, and diagrams. Generated hero objects should be abstract or clearly illustrative, not fake inventory.

Studio assets

Use real project captures, process artifacts, wireframes, motion studies, and labelled concepts. Coldharbour may be shown only with an accurate ownership and project-status description. Do not reuse fictional identities, company logos, testimonials, or metrics from experiments.

File structure

```
public/media/  
├── gateway/  
├── systems/  
├── studio/  
├── work/  
└── shared/
```

Source files stay outside the web bundle. Exported files use descriptive lowercase names, dimensions, and a stable crop plan. Decorative images use empty alt text; informative images use concise contextual alt text.

Component and dependency allowlist

Core launch dependencies

Next.js, React, TypeScript, Tailwind CSS, Supabase SSR/client, Zod, React Hook Form, resolver adapters, Motion, Lucide, clsx, tailwind-merge, class-variance-authority, Radix primitives as needed, and server-only.

Use GSAP only for a reviewed sequence that Motion cannot express cleanly. Add Lenis only if native scrolling plus CSS cannot meet the intended feel; it must preserve keyboard, anchors, history, and reduced motion.

Approved visual lab—not installed by default

Library	Possible use	Gate
<code>`three`, `@react-three/fiber`, `@react-three/drei`, `maath`</code>	One Systems hero or product scene	Static fallback and GPU/performance proof
<code>`postprocessing`, `@react-three/postprocessing`</code>	Carefully scoped scene effect	No readability or battery regression
<code>`@paper-design/shaders-react`</code>	Studio art layer	License and reduced-motion fallback
<code>`@tsparticles/react`, engine, slim</code>	Sparse contextual particles	Prove value over CSS/canvas
<code>`@splinetool/runtime`, React Spline</code>	Authored scene	Export weight and tracking review
<code>`maplibre-gl`</code>	Real service-area or logistics map	A map must answer a user question
<code>`@tabler/icons-react`</code>	Missing specialist icons	Do not ship two complete icon sets casually

Referenced patterns requiring adaptation

Map arcs, 3D card, lamp, aurora, card stack, grid distortion, dither, paper shaders, sparkles, cover, stateful button, spotlight/canvas reveal, encrypted text, expandable cards, MacBook scroll, fluid glass, glass surface, notch, resizable navbar, Spline scene, parallax feature, immersive gallery, sticky reveal, tracing beam, and label/input references are inspiration—not an instruction to install or combine all of them.

Excluded

- Ball-pit animation
- Dependency bundles installed “just in case”
- Unmaintained packages without a containment plan
- Components that duplicate an existing primitive
- Visual effects that require pointer hover or block reading

Every addition needs a named route, purpose, bundle impact, license check, fallback, and removal owner.

Agent orchestration

Context contract

Every agent receives only the current decision register, relevant source-of-truth section, task brief, owned files, acceptance criteria, and required output. The million-character transcript is an archive, not working memory.

Roles

Codex

Own architecture, repository inspection, planning, integration, data/security implementation, verification, and final decisions. Use Luna/Terra models for substantial coding and difficult synthesis when available; use the default model when routing is uncertain.

fcc-claude

Own focused implementation, refactoring, debugging, test repair, and code review inside an approved task. Use Haiku with low context for small mechanical work, Sonnet with moderate context for normal implementation, and Opus rarely for architecture deadlocks, difficult security analysis, or final critical review.

Agy / Antigravity / Gemini

Own visual exploration, moodboards, image/video generation, layout alternatives, creative critique, and multimodal inspection. Use a fast Gemini model for iterations and a stronger model for final art direction or complex multimodal reasoning.

OpenCode and other tools

Use as isolated reviewers, test runners, or focused implementation assistants after confirming installed capabilities. They never become a second simultaneous owner of the same files.

Terminal layout

Keep Codex, fcc-claude, Agy/Gemini, and OpenCode in separate tabs. One coordinator assigns file ownership. Before editing, each agent reads the latest checkpoint. After editing, it writes a handoff with files, dependencies, commands, tests, risks, and next action.

Safety

- No agent receives credentials or personal inquiry data.
- No simultaneous edits to the same files.
- Pin dependencies and preserve lockfiles.
- Agents may install a task-relevant package only after checking existing capabilities and recording the reason.
- A green build is not permission to deploy production.
- Run security, accessibility, performance, and content-truth reviews before release.

Phase sequence

MAP → DECIDE → DESIGN → STRUCTURE → IMPLEMENT → REVIEW → TEST → REFINE → HAND OFF

Each phase produces a named, versioned artifact. Context is refreshed from those artifacts, not from stale chat memory.

Model routing and context budgets

Model names and limits change. Before a long run, inspect the current CLI help/account status and substitute the closest available model. Never hard-code an unverified context limit.

Task	Preferred tier	Context budget	Escalate when
Rename, format, extract, small copy edit	Haiku/Flash/small Codex	Low: task + exact files	Requirements are ambiguous
Isolated component or test	Haiku or fast Gemini	Low–medium	State or architecture crosses boundaries
Normal feature implementation	Sonnet or Luna/Terra	Medium	Security, data migration, or repeated failures appear
UI system or art direction	Strong Gemini + Sonnet/Luna review	Medium	Multimodal consistency remains unresolved
Architecture/data model	Luna/Terra or Sonnet	Medium–high, curated	Tradeoffs remain high impact
Security review	Strong Codex/Sonnet; Opus rarely	High, code-focused	Finding spans several trust boundaries
Hard debugging	Luna/Terra/Sonnet	Start narrow, expand by evidence	Two disciplined attempts fail
Image/video generation	Agy/Gemini media model	Prompt + references only	Brand consistency requires final review
Final integration review	Strongest available, Opus rarely	High but curated	Launch risk is material

Memory rule

“Low,” “medium,” and “high” describe task scope, not maximum token consumption. Low receives one brief and a few files. Medium receives relevant architecture, decisions, and a bounded directory. High receives a curated cross-cut, not the raw interview. Summarize completed work into the shared artifacts after every phase.

Escalation protocol

- 41. State the unresolved question and evidence.
- 42. Reduce the problem to a reproducible case.
- 43. Ask a stronger model for a decision or diagnosis, not a full rewrite.
- 44. Have another model review high-risk output.
- 45. Verify with tests and source evidence.

Use expensive models for uncertainty and consequence, not for volume.

Master execution prompt

Use this as the coordinator prompt after the repository is created. It is deliberately a control brief, not permission for one giant unreviewed generation.

You are the implementation coordinator for MMOptiBuilds. Build from the repository and the supplied project pack. Current owner messages override every document; 02-DECISION-REGISTER.md overrides 01-SOURCE-OF-TRUTH.md; the old transcript is historical evidence only.

MMOptiBuilds is one parent brand on mmoptibuilds.com with /systems, /studio, and private /admin sections. Systems and Studio must have independent visible design systems and SEO strategies while sharing secure operational infrastructure. Systems is a no-stock, quote-first hardware sourcing business. Studio sells focused conversion sites and premium bespoke sites, with client-owned hosting and one-time handoff. Never invent prices, stock, testimonials, clients, certifications, results, availability, deadlines, warranties, or legal conclusions.

Use Codex, fcc-claude, Agy/Antigravity/Gemini, OpenCode, installed skills, plugins, and MCP capabilities intelligently. Keep each tool in a separate terminal tab and assign one owner to each file. Before using a model, choose the least expensive capable tier and give it only task-sized context:

- Small mechanical work: Haiku/Flash/small model, low context.
- Normal implementation and debugging: Sonnet or Luna/Terra, medium context.
- Visual exploration and image/video: Gemini/Agy.
- Architecture, security, hard debugging, and final review: strongest available model; Opus only rarely.

Do not assume exact model names or limits. Inspect the installed CLI and use the closest current option. Do not pass secrets, customer data, or the raw transcript to agents.

Follow MAP → DECIDE → DESIGN → STRUCTURE → IMPLEMENT → REVIEW → TEST → REFINE → HAND OFF. For each task:

46. Read the relevant pack documents and inspect the code.
47. State the task, owned files, acceptance criteria, and exclusions.
48. Add no dependency until existing capabilities are checked; pin and record approved additions.
49. Implement the smallest complete vertical slice.
50. Test behavior, responsive layout, keyboard access, reduced motion, metadata, performance, security, and failure states.
51. Review all public claims for truth.
52. Write a handoff with changed files, commands, evidence, risks, and next action.

Use Cloudflare Workers/OpenNext for production, Supabase with RLS for data and owner auth, Turnstile for public forms, and Cloudflare Web Analytics. Do not use Vercel Hobby for commercial production. Do not create visitor accounts at launch. Do not deploy production until the release checklist passes and the owner-controlled gate is satisfied.

Start with the detailed implementation plan. Work one phase at a time. Never trade conversion, accessibility, mobile usability, security, or truthful content for visual spectacle.

Implementation roadmap

Phase 0 — Business and truth gate

Confirm the legal identity, GST position, invoice method, warranty boundary, supplier wording, delivery boundaries, support email, and case-study permissions. Create an approved fact sheet. Development may begin with private markers, but unresolved legal or factual copy cannot ship.

Phase 1 — Foundation

Create the private repository, dependency baseline, lint/format/test scripts, environment templates, Cloudflare/OpenNext config, local Supabase config, route groups, CI, and security policy. Establish Systems and Studio token boundaries before building components.

Exit: clean install, typecheck, lint, unit tests, production build, Cloudflare preview build, and secret scan pass.

Phase 2 — Content structure

Implement semantic route shells, metadata factory, sitemap/robots, navigation, footer, legal routes, and structured content schemas. Add root, Systems, Studio, About, Contact, and one case-study skeleton using approved copy.

Exit: every launch route works by direct URL, refresh, keyboard, and mobile viewport; no effect libraries are required.

Phase 3 — Conversion vertical slice

Build Contact first from browser through Turnstile, validation, database insertion, owner notification, reference screen, and dashboard visibility. Then reuse the proven invisible foundation for Systems, Enterprise, and Studio briefs.

Exit: success, validation, spam, duplicate, server failure, notification failure, and retry paths are tested.

Phase 4 — Owner dashboard

Add owner auth, protected layout, inbox, division filters, status changes, notes, quote metadata, follow-up dates, audit history, and CSV export. Keep it practical; do not build a customer CRM platform.

Exit: unauthorized reads and writes fail; owner workflows pass end to end.

Phase 5 — Visual systems

Implement Systems and Studio typography, color, spacing, media, layout, component, and motion rules independently. Add responsive art direction and generated temporary assets with a manifest.

Exit: visual review confirms distinction and parent-brand coherence; WCAG checks pass.

Phase 6 — Premium interaction

Add only effects with a named purpose. Profile each route before and after. Dynamic-import route-specific libraries, provide static fallbacks, and remove effects that exceed budgets or blur the offer.

Exit: CWV budgets, reduced-motion experience, touch behavior, and weak-device fallback pass.

Phase 7 — SEO and proof

Finalize intent-specific content, metadata, JSON-LD, internal linking, imagery, case-study evidence, Search Console, and analytics events. Do not publish thin later routes merely to increase page count.

Exit: crawl, canonical, indexability, schema, and content-truth reviews pass.

Phase 8 — Security, legal, and release

Run dependency, secret, authorization, form-abuse, header, RLS, privacy, content, and accessibility reviews. Obtain professional sign-off for applicable legal/tax documents. Test backup/export and rollback.

Exit: launch checklist is signed, production variables are present, email routing works, and no critical finding remains.

Phase 9 — First 90 days

Review qualified inquiry quality weekly, Search Console monthly, CWV monthly, spam patterns weekly, dependencies monthly, and content accuracy quarterly. Add pages or premium interactions only from evidence.

QA and acceptance criteria

Global

- All launch routes return the intended status and work on direct refresh.
- There are no broken internal links, console errors, failed required requests, or missing assets.
- Public copy contains no unapproved claim, fake proof, price, stock statement, or placeholder.
- 320, 375, 768, 1024, 1440, and wide layouts are visually reviewed.
- Keyboard-only use, visible focus, skip navigation, 200% zoom, reduced motion, and touch are reviewed.
- Titles, descriptions, canonicals, Open Graph data, sitemap, robots, headings, and structured data are correct.

Forms

- Client and server schemas agree, but the server remains authoritative.
- Turnstile is validated server-side; fake, expired, missing, and replayed tokens fail safely.
- Repeated taps and retries do not create duplicates.
- Errors preserve safe input, focus the summary, identify fields, and support retry.
- Analytics contain no personal or sensitive values.
- Notification failure does not lose a stored inquiry.

Dashboard

- Anonymous and non-owner access fails for every read and mutation.
- RLS tests prove cross-role isolation.
- Status, notes, quotes, follow-ups, export, and audit history work.
- Admin routes are noindex and absent from public navigation and sitemap.

Performance

- Field target: LCP $\leq 2.5s$, INP $\leq 200ms$, CLS ≤ 0.1 at p75.
- No unnecessary global client component or global visual library.
- Images have dimensions and responsive sizing; video/3D has poster/fallback.
- Reduced-motion mode removes nonessential transforms and autoplay behavior.

Release evidence

Archive command output, test report, route screenshots, mobile screenshots, accessibility report, security findings, Lighthouse/trace summary, migration status, policy approvals, and rollback procedure with the release.

Deployment and handoff

Production

Deploy the Next.js application to Cloudflare Workers using OpenNext. Route `mmoptibuilds.com` through Cloudflare DNS and TLS. Keep preview data synthetic and separate from production Supabase.

Release sequence

53. Freeze content and migrations for the release candidate.
54. Run install from lockfile, typecheck, lint, tests, production build, OpenNext build, security checks, and browser suite.
55. Apply reviewed database migrations and verify RLS.
56. Deploy to a production-equivalent preview and run smoke tests.
57. Confirm environment variables, Turnstile hostnames, email destination, analytics, sitemap, robots, and canonical host.
58. Deploy production, run smoke tests, and monitor logs and inquiries.
59. Record release SHA, migration version, operator, evidence, and rollback point.

Rollback

Application rollback must not assume a destructive schema rollback. Prefer backward-compatible migrations, feature flags for risky behavior, and a previous known-good Worker deployment. Pause form acceptance with a clear message if data integrity is uncertain.

Studio client handoff model

- Client creates and owns domain, hosting, email, analytics, database, and third-party accounts.
- Client grants time-limited collaborator access; primary passwords are not shared.
- MMOptiBuilds records configuration, environment-variable names, repository ownership, deployment steps, backups, and recovery contacts.
- Remove access after acceptance and confirm removal in writing.
- The 30-day warranty starts at documented acceptance or launch, according to the approved agreement.
- Later changes are separately scoped.

No managed-hosting promise

Do not describe deployment help as permanent hosting management. The agreement must identify who monitors uptime, renews domains, pays vendors, handles abuse notices, and maintains content after handoff.

Operations and maintenance

Daily and weekly

- Review inquiries and spam; respond manually by email.
- Record status, notes, quote state, and next follow-up.
- Check production errors after a release and investigate repeated form failures.
- Verify supplier, availability, warranty, and delivery facts before every Systems quote.

Monthly

- Review dependency and security updates in a branch.
- Review Search Console, analytics, Core Web Vitals, broken links, 404s, and top inquiry routes.
- Export the inquiry database to encrypted owner-controlled storage if the free platform lacks suitable downloadable backups.
- Test owner recovery, email routing, Turnstile, and one complete form journey.
- Remove stale collaborator access and unused secrets.

Quarterly

- Review service copy, case-study truth, privacy notice, retention, warranty and terms versions.
- Delete data that has exceeded the approved retention period.
- Replace temporary visuals with real assets where possible.
- Review whether free-tier limits or pausing risk now justify paid infrastructure.

Incident priorities

60. Protect people and data.
61. Stop unsafe writes or public exposure.
62. Preserve evidence without logging more personal data.
63. Restore a known-good service.
64. Notify affected parties and authorities when legally required.
65. Document cause, impact, remediation, and preventive work.

Scaling triggers

Pay for infrastructure only when a measured limit affects reliability, commercial terms, recovery, email deliverability, support, or owner time. Likely first upgrades are transactional email, Supabase Pro for non-pausing/backups, and paid hosting if Cloudflare limits or requirements change.

Research report

Research date: 2026-08-11. Recheck commercial terms before launch because services change.

Key findings

- Vercel documents Hobby as non-commercial personal use. It is unsuitable for the commercial production site without an eligible paid plan.
- Cloudflare officially supports most Next.js App Router features through its OpenNext adapter on Workers. Node.js middleware remains a documented limitation; PPR and composable caching remain experimental in Next.js.
- Supabase free projects can pause after inactivity and do not include downloadable database backups. The platform's built-in auth email service is best-effort and limited to two emails per hour; production auth should use custom SMTP.
- Cloudflare Turnstile has a free plan suitable for small businesses and requires server-side token verification.
- Cloudflare Web Analytics is available on all plans and is positioned as privacy-conscious.
- Cloudflare Email Routing is free. On the free Workers plan, application emails can be sent free to verified destination addresses; arbitrary outbound recipients require Workers Paid. Resend currently offers 3,000 emails per month with a 100-per-day free limit and one domain.
- GitHub Free currently includes private repositories and 2,000 Actions minutes per month, with storage quotas.
- Google recommends people-first content, descriptive URL structures, mobile parity, and structured data that reflects visible content.
- Current Core Web Vitals "good" thresholds are LCP $\leq 2.5s$, INP $\leq 200ms$, and CLS ≤ 0.1 at the 75th percentile.
- WCAG 2.2 is the accessibility baseline.
- Awwwards scoring examples weight Design 40%, Usability 30%, Creativity 20%, and Content 10%; its developer criteria also emphasize SEO/semantics, animation, accessibility, performance, responsive design, and metadata.

Recommended reading

- Vercel Hobby: <https://vercel.com/docs/plans/hobby>
- Cloudflare Next.js: <https://developers.cloudflare.com/workers/framework-guides/web-apps/nextjs/>
- Cloudflare Workers pricing: <https://developers.cloudflare.com/workers/platform/pricing/>
- Cloudflare Turnstile: <https://developers.cloudflare.com/turnstile/>
- Cloudflare Web Analytics: <https://developers.cloudflare.com/web-analytics/>
- Cloudflare Email Service pricing: <https://developers.cloudflare.com/email-service/platform/pricing/>
- Supabase production checklist: <https://supabase.com/docs/guides/deployment/going-into-prod>
- Supabase SMTP: <https://supabase.com/docs/guides/auth/auth-smtp>
- Supabase API keys: <https://supabase.com/docs/guides/getting-started/api-keys>
- Resend pricing: <https://resend.com/pricing>
- GitHub Actions billing: <https://docs.github.com/billing/managing-billing-for-github-actions/about-billing-for-github-actions>
- Google SEO Starter Guide: <https://developers.google.com/search/docs/fundamentals/seo-starter-guide>
- Google helpful content: <https://developers.google.com/search/docs/fundamentals/creating-helpful-content>
- Google mobile-first indexing: <https://developers.google.com/search/docs/crawling-indexing/mobile/mobile-sites-mobile-first-indexing>
- Core Web Vitals: <https://web.dev/articles/vitals>
- WCAG 2.2: <https://www.w3.org/TR/WCAG22/>
- OWASP Cheat Sheet Series: <https://cheatsheetseries.owasp.org/>
- India DPDP Act: <https://www.meity.gov.in/static/uploads/2024/06/2bf1f0e9f04e6fb4f8fef35e82c42aa5.pdf>
- India DPDP Rules 2025: <https://www.meity.gov.in/documents/act-and-policies/digital-personal-data-protection-rules-2025-gDOxUjMtQWa>
- CBIC GST FAQ: <https://cbic-gst.gov.in/faq.html>

Interpretation

The best no-new-spend architecture is Cloudflare Workers/OpenNext plus Supabase, not Vercel Hobby. At launch, owner-only email notification can use Cloudflare's verified-destination path. Visitor acknowledgement can remain on-screen until a verified transactional email provider is configured. This reduces cost, abuse surface, and friction.

Owned site inventory summary

The current decision is to build MMOptiBuilds from scratch and not export any existing hosted site.

Useful references

- [coldharbour-mm.vercel.app](#): strongest independent project and potential honest case study.
- [studio-effects-1.vercel.app](#): strongest Studio motion and monochrome interaction reference.
- [optibuilds-glass-dreamscape.lovable.app/builder](#): useful structural reference for a future quote-first guided brief.
- [test8sm.netlify.app](#): strongest S.M. Industrial variant, but it is a separate business/project and needs factual verification.
- [orta-oslo.netlify.app](#): premium editorial reference only.

Keep separate or private

S.M. Industrial variants, Network Pro Systems, OmniGPU, CodeShift, trading experiments, Weekend OS, and unrelated product prototypes should not become routes on [mmoptibuilds.com](#) merely because they exist.

Unsafe proof

Do not publish fictional identities, companies, testimonials, awards, security claims, compliance claims, inventory claims, years in business, or performance metrics from experiments. The old Google-like MMOptiBuilds app is not the current brand direction. Incomplete or unpublished Lovable builds have no launch value.

Portfolio rule

Every case study requires an ownership label, project type, real scope, evidence, permission if another organization is involved, and honest outcome language. A concept must say “concept.” An independent product must not be called a client engagement.

Coverage and contradictions

Resolved

Earlier idea	Current authority
Broad technical consultancy	Two divisions: Systems and Studio
Shared visible design system	Distinct visible systems; shared invisible infrastructure
Public packages or prices	Requirements first; private tailored quote
Inventory-led PCs	No stock; source after inquiry
Enterprise consulting/installation	Exact-spec sourcing and delivery only
Managed client hosting	Client-owned accounts and one-time deployment
Giant all-in-one build prompt	Phased execution with checkpoints and handoffs
Export old projects	Build from scratch; old work is reference/case-study material
Vercel as free production default	Cloudflare Workers/OpenNext due commercial-plan terms
Visitor email verification	No visitor account or verification at zero-spend launch

High confidence

Brand, permanent domain, division structure, audiences, no public prices, no inventory, enterprise boundary, Studio handoff model, dashboard ownership, design separation, truth rules, and build-from-scratch decision.

Requires owner or professional evidence before launch

Legal entity, address, GST, invoices, final warranties, cancellation/refund terms, supplier representation, delivery reach, response times, email alias, case-study permissions, and any claim involving customers or measurable results.

Transcript handling

The attached June 2026 transcript contains useful preferences and referenced components, but also obsolete architectures, speculative details, and private information. Agents should not read it unless a current document lacks evidence. discard1.txt and discard2.txt are generic prompt examples, not facts.

Risk register

Risk	Likelihood	Impact	Control
Commercial use on an ineligible free plan	Medium	High	Cloudflare production; verify terms before release
Unverified legal/tax language	High	High	Lawyer/CA launch gate
Fake or ambiguous portfolio proof	Medium	High	Evidence file and content-truth review
Form spam or notification abuse	High	Medium	Turnstile, throttling, honeypot, verified destination
Public data exposure through Supabase	Medium	Critical	RLS, server-only secrets, authorization tests
Free Supabase pause or weak backup	Medium	High	Activity monitoring, encrypted exports, paid trigger
Heavy motion damages CWV	High	High	Static-first, budgets, dynamic imports, fallbacks
Agent context drift	High	High	Authority hierarchy and task-sized briefs
Concurrent agent file conflicts	Medium	Medium	One owner per file and checkpoints
Dependency sprawl	High	Medium	Core allowlist and per-dependency gate
Quote error from changing supply	High	High	Manual distributor confirmation and quote validity
Warranty confusion	Medium	High	Separate manufacturer, workmanship, and Studio terms
Owner workload becomes bottleneck	Medium	Medium	Structured forms, dashboard, templates, follow-ups
Inquiry data retained too long	Medium	High	Retention schedule and quarterly deletion
Client illegal use after handoff	Low-medium	High	Client ownership, contract, access removal, no management

Risk owners and review dates belong in the dashboard or project tracker once implementation begins.

Launch checklist

Business and content

- ☐ Legal identity, GST, invoice, warranty, delivery, and supplier facts approved
- ☐ Privacy, terms, hardware supply, Studio agreement, and warranty language reviewed
- ☐ Every testimonial, case study, image, metric, certification, and claim has evidence
- ☐ No public prices, fake stock, guaranteed results, or unresolved placeholders

Engineering

- ☐ Lockfile install, lint, typecheck, unit, integration, browser, and production builds pass
- ☐ Cloudflare/OpenNext preview and production bindings verified
- ☐ Supabase migrations applied; RLS and owner authorization tests pass
- ☐ Secrets present only in managed environments; old secrets rotated
- ☐ Turnstile, throttling, idempotency, and notifications tested
- ☐ Backup/export and rollback tested

Experience

- ☐ All routes, links, refreshes, 404s, forms, and error states verified
- ☐ Mobile, keyboard, zoom, contrast, focus, screen-reader smoke test, and reduced motion reviewed
- ☐ CWV and bundle budgets pass on representative devices
- ☐ Metadata, sitemap, robots, canonicals, structured data, and Search Console ready
- ☐ Analytics events contain no personal data

Operations

- ☐ Owner dashboard and recovery tested
- ☐ info@mmoptibuilds.com routing and reply workflow verified
- ☐ Inquiry response, quoting, follow-up, retention, and incident procedures rehearsed
- ☐ Production release SHA, operator, evidence, and rollback point recorded
- ☐ Post-launch checks scheduled for 1 hour, 24 hours, 7 days, and 30 days

Glossary

- **Focused project:** A lean Studio engagement with constrained scope and baseline quality intact.
- **Premium project:** A bespoke Studio engagement with deeper strategy, art direction, interaction, and asset work.
- **RFQ:** Request for quotation, used for exact-spec enterprise hardware.
- **RLS:** Row Level Security, database policies that restrict which rows a role can access.
- **OpenNext:** Adapter that packages Next.js for non-Vercel runtimes such as Cloudflare Workers.
- **Turnstile:** Cloudflare's bot challenge, validated by the application server.
- **Core Web Vitals:** Google's field measures for loading, interaction responsiveness, and layout stability.
- **Truth gate:** Review that verifies every public claim and proof asset.
- **Handoff:** Documented transfer of a Studio client's code, accounts, deployment, and responsibilities.
- **Invisible foundation:** Shared validation, data, accessibility, analytics, and testing code that does not force shared appearance.
- **Task-sized context:** Only the decisions, files, and evidence needed for one agent task.

Legal and tax review brief

Ask an India-qualified lawyer and chartered accountant to review:

- Business identity, place of supply, GST registration and invoicing
- Hardware quote acceptance, price validity, substitutions, payment, title, risk, shipping, inspection, returns, cancellation, and warranty allocation
- Studio scope, approvals, client content, IP ownership, third-party licenses, payment, cancellation, deployment, access removal, warranty, support, limitation of liability, indemnities, and prohibited use
- Privacy Notice, consent language, retention, processors, security, data-principal requests, grievance contact, and breach process under applicable DPDP requirements
- Website Terms and acceptable-use language
- Whether international inquiries, cross-border data, or interstate supply require additional terms

Provide the reviewer with the source of truth, data model, form fields, delivery model, and proposed policies. Do not ask them to approve generic templates without the real operating facts.

Policy drafting outline

This is a drafting structure, not publishable legal text.

Privacy Notice

Identity and contact; data collected; purpose and legal basis/consent; source; recipients/processors; storage and retention; security; rights and request process; grievance contact; children; cross-border processing if any; changes and version date.

Website Terms

Informational purpose; no offer until written quote; intellectual property; acceptable use; third-party links; availability; disclaimer; liability; governing law; contact; changes.

Hardware Supply Terms

Requirements and quote; availability and substitutions; taxes and logistics; payment and acceptance; cancellation; delivery and inspection; manufacturer warranty versus assembly workmanship; returns; excluded enterprise services; data on customer devices; liability; disputes.

Studio Agreement

Scope and deliverables; client responsibilities; content and approvals; milestones; fees and changes; domain/hosting ownership; collaborator access; third-party services; intellectual property and portfolio permission; accessibility/SEO boundaries; deployment and acceptance; 30-day bug warranty; support exclusions; termination; liability; prohibited use.

Owner runbook

New inquiry

Open the dashboard from a trusted device, check the spam signals, read the complete brief, verify consent version, classify the division and service, and set a next action. Reply from the business mailbox, then record the activity.

Systems quote

Confirm exact requirements, distributor availability, condition, warranty source, tax, delivery, assembly, testing, setup, validity, and exclusions. Separate each cost internally. Do not purchase until written acceptance and the agreed payment condition.

Studio quote

Confirm goals, audience, primary conversion, page scope, content/assets, integrations, accessibility needs, domain/hosting ownership, approvals, timeline, warranty, and exclusions. Use change control when new work appears.

Handoff

Transfer repository and accounts to the client's ownership, provide deployment and recovery notes, confirm acceptance, remove access, and record warranty dates.

Suspected incident

Stop unsafe behavior, preserve minimal evidence, rotate affected credentials, assess exposure, restore known-good service, seek legal/security advice where needed, notify as required, and document corrective action.

Inquiry email templates

Initial reply — Systems

Subject: We received your MMOptiBuilds Systems request — {{reference}}

Hello {{name}},

Thank you for sharing your requirements. I'll review the workload, sourcing options, availability, warranty, and delivery details before replying with questions or a tailored quote.

Reference: {{reference}}

No purchase is confirmed until you accept a written quote.

Regards,

Founder & Technical Director

MMOptiBuilds

Initial reply — Enterprise

Thank the contact, restate the part number and quantity, request any missing condition/warranty/delivery facts, and state that sourcing and delivery are the current scope. Do not promise availability or compatibility before verification.

Initial reply — Studio

Thank the contact, restate the business goal and selected project mode, identify missing copy/assets/domain/hosting information, and explain the next manual review step. Do not promise rankings, conversion, or a launch date before scope.

Quote follow-up

State the quote reference, validity date, what is included, what is excluded, dependencies, payment/acceptance step, and the exact question needed to proceed. Avoid manipulative urgency.

MMOptiBuilds implementation plan

Required sub-skill: execute this plan with `superpowers:executing-plans`; use test-driven development for each behavior change and verification-before-completion before every completion claim.

Goal: Build and launch the first commercial MMOptiBuilds website with distinct Systems and Studio experiences, secure inquiry ingestion, and an owner-only dashboard.

Architecture: One Next.js App Router application deployed to Cloudflare Workers through OpenNext. Supabase provides Postgres and owner authentication with RLS. Public forms use server routes, Zod, Turnstile, throttling, and idempotency. Route groups isolate visible design systems.

Stack: Next.js 16, React 19, TypeScript, Tailwind CSS 4, Supabase SSR/client, Zod, React Hook Form, Motion, Vitest, Playwright, axe, Wrangler, OpenNext.

Task 1: Bootstrap the repository

Files: package.json, lockfile, TypeScript, ESLint, PostCSS, OpenNext, Wrangler, Vitest, Playwright configs; .env.example; .gitignore; README.md.

66. Copy the verified starter manifests into a new private repository.
67. Run a clean lockfile install.
68. Add the minimum app/layout.tsx, app/page.tsx, and global stylesheet.
69. Write a smoke test that expects the root heading.
70. Run typecheck, lint, unit test, Next build, and OpenNext build.
71. Commit chore: establish verified application foundation.

Task 2: Establish route and theme boundaries

Files: app/(gateway), app/(systems), app/(studio), styles/tokens, division layouts, navigation tests.

72. Write route tests for /, /systems, /studio, /about, /contact, and /admin.
73. Create semantic layouts with independent CSS variable scopes and font contracts.
74. Implement the global wordmark and accessible division switcher.
75. Verify direct refresh, skip links, focus order, and mobile navigation.
76. Commit feat: establish independent division shells.

Task 3: Add metadata and content model

Files: metadata utilities, route content modules, sitemap, robots, JSON-LD helpers, tests.

77. Write tests for canonical host, unique titles, noindex admin, and sitemap exclusions.
78. Build typed page-content schemas and metadata factories.
79. Implement launch copy from the approved pack; reject unapproved placeholders in CI.
80. Add JSON-LD only for verified visible facts.
81. Commit feat: add truthful content and SEO foundation.

Task 4: Create Supabase schema and security tests

Files: supabase/migrations, supabase/tests, generated types, data-access layer.

82. Write failing RLS tests for anonymous, authenticated non-owner, and owner access.
83. Create contacts, inquiries, division briefs, quotes, notes, activities, and consents.
84. Enable RLS and implement narrow owner policies plus a controlled public insert function.
85. Add indexes and immutable activity/consent behavior.
86. Run migration reset and security tests.
87. Commit feat: create secured inquiry data model.

Task 5: Build the Contact vertical slice

Files: contact form, schema, route handler, Turnstile adapter, notification adapter, tests.

88. Write browser and integration tests for success, invalid data, missing/invalid Turnstile, duplicate retry, storage failure, and notification failure.
89. Implement accessible fields, summary, submission state, and reference screen.

90. Implement server validation, body limit, origin policy, honeypot, rate control, idempotency, database insert, and non-blocking owner notification.
91. Confirm analytics contains no payload data.
92. Run tests and manually verify with keyboard and mobile.
93. Commit feat: deliver secure contact inquiry slice.

Task 6: Add Systems and enterprise briefs

Files: Systems pages, form schemas, handlers, shared ingestion adapter, tests.

94. Write journey tests for gaming, workstation, and exact-spec enterprise flows.
95. Implement progressive fields and private budget ranges.
96. State no-stock, availability, quote, and enterprise service boundaries near submission.
97. Store normalized division-specific records under one inquiry envelope.
98. Commit feat: add requirement-led systems inquiries.

Task 7: Add Studio brief and portfolio evidence

Files: Studio pages, discovery form, Work index, Coldharbour case study, evidence metadata, tests.

99. Write tests for focused/premium branching, required asset questions, and honest labels.
100. Implement conversion-led service pages and discovery flow.
101. Add Coldharbour only with verified ownership/scope copy and no synthetic performance claim.
102. Commit feat: add studio discovery and honest work proof.

Task 8: Build owner authentication and dashboard

Files: auth middleware/proxy compatible with Cloudflare, admin pages, server actions/handlers, tests.

103. Write authorization tests for every read and mutation.
104. Implement owner login, recovery through configured production SMTP, and protected layout.
105. Add inbox, filters, detail, status, notes, quote metadata, follow-up, audit trail, and CSV export.
106. Confirm admin noindex and no personal data in logs.
107. Commit feat: deliver owner inquiry workspace.

Task 9: Implement visual systems

Files: division tokens, components, media manifest, route styles, visual tests.

108. Build static mobile-first compositions first.
109. Implement Systems technical material language and Studio editorial language independently.
110. Add approved generated placeholders with alt text and replacement metadata.
111. Run contrast, zoom, typography, and responsive review.
112. Commit feat: apply distinct systems and studio art direction.

Task 10: Add reviewed motion

Files: route-specific motion modules, reduced-motion utilities, performance tests.

113. Record baseline bundle and performance traces.
114. Add Motion transitions with reduced-motion alternatives.
115. Add GSAP or WebGL only for a named high-value sequence after a separate review.
116. Compare traces and remove anything outside budgets.
117. Commit feat: add purposeful accessible motion.

Task 11: Security, accessibility, SEO, and release review

Files: test reports, security findings, release checklist, handoff.

118. Run clean install, lint, typecheck, unit, integration, browser, axe, build, OpenNext preview, dependency audit, secret scan, and RLS tests.
119. Inspect every route at target viewports and reduced motion.
120. Validate sitemap, robots, canonicals, structured data, Search Console, Turnstile, notifications, backups, and rollback.

MMOPTIBUILDS / MASTER PROJECT HANDBOOK

121. Resolve all critical/high findings and document accepted residual risks.
122. Obtain legal/tax/content truth approvals.
123. Deploy through the owner-controlled release gate and run post-deploy smoke tests.
124. Commit release: launch mmoptibuilds foundation.