

VOLUME 1

FROM LOVABLE TO WORDPRESS

The AI Workflow

A practical path from a client requirement to a clear, conversion-ready website interface.

Yukesh Chaudhary

yukesh.com.np

REQUIREMENTS

LOVABLE UI

GITHUB

WORKING TITLE

A practical series built around one real workflow

From Lovable to WordPress is a working series title. It can be changed before publication without changing the method inside the books.

Volume 1: The AI Workflow

Turn a client conversation into a specification, plan the interface, guide Lovable with a strong prompt, and move the approved project through GitHub.

Volume 2: The WordPress Conversion

Inspect the exported code, map it into native WordPress theme files, and create templates, settings, menus, and editable content.

Volume 3: From Local Test to Delivery

Build content models, test locally, solve defects, configure search visibility and performance, migrate, and hand the website to the client.

The promise

Follow a modern AI-assisted workflow for turning a user interface into a custom WordPress website. The focus is disciplined delivery, not shortcuts.

Copyright notice. Copyright 2026 Yukesh Chaudhary. All rights reserved. Website: yukesh.com.np.

Product names belong to their respective owners. Screens and pricing may change. Always follow the current terms of every service you use.

PRACTITIONER NOTES

**Yukesh
Chaudhary**

About the author

Yukesh Chaudhary is a digital entrepreneur and the founder of Niguro. His work sits at the meeting point of websites, search, content, and practical business systems.

He developed this guide from a simple question: how can one person move from a client brief to a polished WordPress website without getting lost between design tools and code?

This is not a textbook about every WordPress feature. It is a field guide for the job in front of you. I show the decisions I make, the prompts I use, and the checks that stop a fast build from becoming a weak build.

Read more at yukesh.com.np.

AI can make the first draft faster. It cannot decide what the client truly needs, review every page, or accept responsibility for the final result.

How to use this volume

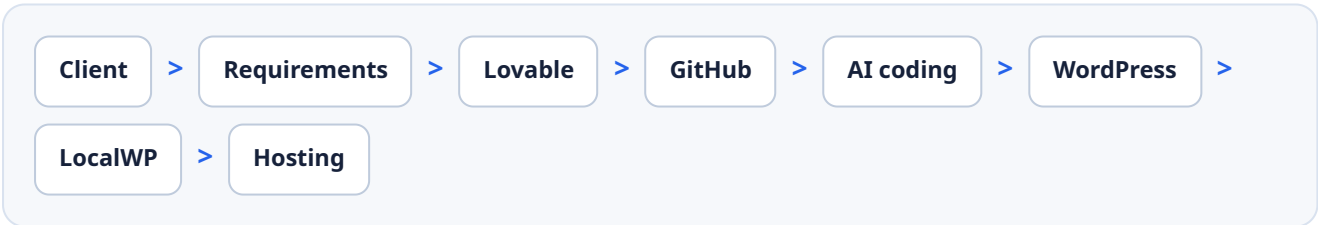
1. Choose a real or fictional business to follow alongside ABC Construction.
2. Complete each worksheet before moving to the next chapter.
3. Save every approved decision in one requirement document.
4. Do not export the interface until every page has been reviewed.

Contents

01	The Old Way vs the AI Way	5
02	What You Actually Need	8
03	Turning a Client Into a Specification	11
04	Designing Before Coding	16
05	Prompting Lovable Correctly	18
06	Lovable to GitHub	22

Running project: ABC Construction. Across all three volumes, we will build one fictional client website with Home, About, Services, Projects, Team, and Contact pages. Using one project makes each decision easier to see.

The complete journey

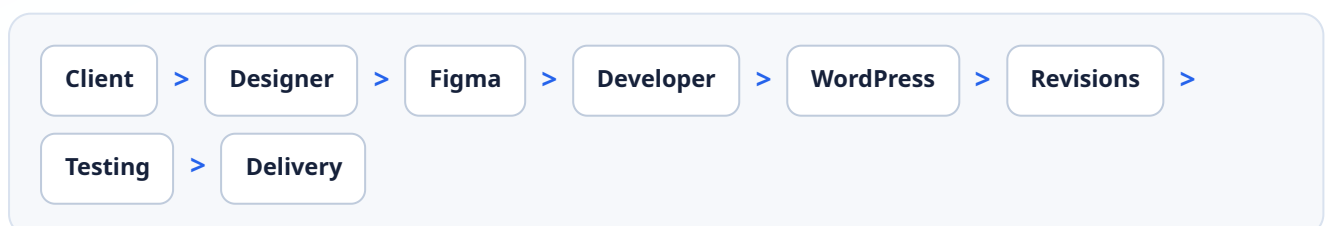


This volume covers the path from Client to GitHub. Volume 2 begins with the exported project.

01

The old process creates expensive handoffs

The traditional website process can produce excellent work. Its weakness for a smaller project is the number of separate handoffs between people, files, and decisions.



Every arrow is a translation. A brief becomes a design. A design becomes code. Code becomes a content system. Each translation can add delay or lose intent.

The AI way joins the stages without removing judgment

The new workflow does not remove design, development, or testing. It lets one capable builder move through those stages with AI support.



What becomes faster

- Early interface drafts
- Repeated layout work
- Mapping components into templates
- Generating a first code version
- Exploring possible fixes

What still needs you

- Understanding the client
- Choosing the right content model
- Rejecting unnecessary features
- Testing every important path
- Accepting responsibility for delivery

Speed is not proof. An interface that appears in minutes can still be inaccessible, confusing, hard-coded, insecure, or impossible for the client to update.

The new role of the builder

I act as translator, editor, system designer, and tester. AI produces drafts. I decide what the website means, how WordPress should control it, and when it is ready.

Small projects become more practical when rework falls

In a traditional project, each specialist must understand the brief and hand the result forward. That can be the right choice for a large or complex website. For a small or medium business site, the coordination cost may be larger than the actual build.

An AI-assisted workflow can reduce the time spent on blank-page design, repetitive code, and routine adjustments. This lets a solo builder or small team offer custom work where a fully staffed process may not fit the budget.

Cost pressure	Traditional risk	AI-assisted response
Discovery	Weak notes travel through several people.	One written specification guides every tool.
Design	Many revisions before the client sees a complete page.	Generate quickly, then review in focused passes.
Development	Repeated component and template setup.	Use AI for a first implementation, then inspect and test.
Change requests	Intent is lost between design and code.	Trace every change back to the approved requirement.

ABC Construction: the economic advantage

ABC needs six main pages and a Projects section. A small contractor does not need an enterprise design team, but it does need a professional result. The workflow lets us prepare a tailored interface quickly while keeping Projects editable in WordPress later.

The business model is not “cheap because AI did it.” The value is a clear process, faster iteration, fewer handoff mistakes, and a website the client can use.

02

A small stack works when every tool has one job

The best stack is not the one with the most tools. It is the one where each stage has a clear owner and a clear output.

Plan and design

Requirements document
Lovable

Move and build

GitHub
AI coding environment
DeepSeek

Run and deliver

LocalWP
WordPress
Migration, SEO, cache

Use tools as a chain, not as a collection

Tool	Its job in this method	What you must provide
Lovable	Generates and refines the visual interface.	A detailed brief, page structure, visual direction, and focused feedback.
GitHub	Stores the generated code and forms the bridge to local development.	A connected repository, sensible commits, and a clean approved version.
AI coding environment	Lets an AI model inspect and modify the project files.	The correct project folder, a precise task, constraints, and a test request.
DeepSeek	Acts as a coding model for theme conversion, debugging, and refactoring.	Enough context to understand the code and one bounded change at a time.

The model is not the workflow. A strong model with vague instructions can still damage the project. The repeatable value comes from prompt, context, change, test, and review.

Use free tiers responsibly

A free or free-tier coding environment is useful when it supports the project size and its terms allow your use. Do not build a business process around account tricks, artificial limits, or a promise of unlimited access. Plans and policies change.

Before you commit to a tool

- ☐ Confirm it can open or import your project.
- ☐ Confirm the model can read the relevant files.
- ☐ Confirm export and ownership terms.
- ☐ Confirm you can continue manually if the free plan changes.

Local work is separate from the live website

Tool	Purpose	When it enters
LocalWP	Runs a WordPress site on your own computer for development and testing.	After the exported interface is ready for conversion.
WordPress	Provides the content system, administration, templates, users, media, and extensions.	During conversion and throughout the life of the website.
All-in-One WP Migration	Packages the local site for transfer to production, subject to current limits and compatibility.	After local testing passes.
Rank Math	Helps manage titles, descriptions, sitemaps, index settings, and supported schema.	During pre-launch search setup.
LiteSpeed Cache	Provides caching and optimization features, especially on compatible LiteSpeed hosting.	On the production environment after compatibility checks.

Do not install tools by habit. A plugin is justified by a clear job. Hosting technology, server rules, theme code, and plugin features must work together.

Where the real cost begins

You may be able to build the development workflow for **approximately \$21.50 in tool expenditure, excluding production WordPress hosting and any optional paid services**. Tool pricing and free plans can change, so verify current terms before presenting a quote.

Local development can be free. A public website still needs suitable production hosting, a domain when the client does not already own one, and any paid service the project truly requires.

Ask for decisions, not adjectives

Business and audience

- What does the business sell?
- Who is the main customer?
- Which locations does it serve?
- Why should a visitor trust it?

Pages and navigation

- Which pages are required?
- What belongs in the main menu?
- What is the main action on each page?

Brand and interface

- Which colors and type styles are approved?
- What should the website feel like?
- Which existing assets can be used?

Content and administration

- Will the client publish blog posts?
- Which items repeat, such as services or projects?
- What must the client edit without code?

Behavior and search

- What should happen on mobile?
- Which forms, filters, or special functions are needed?
- Which search topics and locations matter?

Boundaries

- What is not included?
- Who provides text and images?
- Who approves the final page list?

Ask “what happens next?” When a visitor presses Request a Quote, does the form open, does a contact page load, or does a phone call begin? Every important button needs a defined result.

Separate facts, choices, and open questions

Client notes often mix confirmed information with ideas. If you treat every idea as approval, the project grows without control.

Status	Meaning	ABC Construction example
Confirmed	The client has approved it.	Six main pages. The primary call to action is Request a Quote.
Proposed	You recommend it, but approval is pending.	Add project filters by service type.
Open	A missing answer blocks design or development.	Who receives quote form submissions?
Excluded	It is outside the current scope.	No customer login or online payment.

Translate vague words into visible rules

“Modern”

This could mean almost anything.

Visible rule

Large project photography, strong headings, simple two-color palette, generous spacing, and no decorative animation.

“Easy to edit”

It does not say what needs editing.

Visible rule

The client can add services, projects, team members, phone details, and footer social links from WordPress.

Use one document as the source of truth

Copy these sections into your own document. Keep answers short, specific, and approved.

1. Project summary

2. Business and audience

3. Primary goal and calls to action

4. Required pages

5. Navigation

6. Brand colors and typography

7. Page sections

8. Forms and destinations

9. Blog and repeatable content

10. Admin editing needs

11. Responsive behavior

12. Search visibility needs

13. Special functionality

14. Exclusions and approval

Approval checkpoint. Ask the client to approve the page list, key functions, editable areas, and exclusions before interface generation begins.

The running example becomes buildable

Business	Residential and light commercial construction company serving one regional market.
Main audience	Homeowners, property managers, and small business owners comparing contractors.
Primary goal	Generate qualified quote requests. Secondary goal: build trust through completed projects.
Pages	Home, About, Services, Projects, Team, Contact.
Main navigation	Logo, six page links, and a highlighted Request a Quote button.
Brand	Deep navy, construction orange, warm white, charcoal text. Strong sans-serif headings and simple body text.
Forms	Contact and quote form with name, email, phone, project type, location, budget range, timeline, and message.
Repeatable content	Services, Projects, and Team members must be manageable as structured content.
Responsive behavior	Mobile menu, readable tap targets, stacked cards, optimized project galleries, and persistent access to the quote action.
Search needs	Clear page titles, service pages, project detail pages, local business information, and indexable content.
Excluded	No payments, customer portal, live chat, or job application system in the first release.

Each page gets one job and one next step

Page	Job	Main sections	Next step
Home	Build trust and route visitors.	Hero, services, project proof, process, testimonials, quote callout.	Request a quote
About	Explain experience and working standards.	Story, values, safety, approach, team preview.	Meet the team
Services	Help visitors match a need to an offer.	Service cards, process, service area, frequently asked questions.	Discuss a project
Projects	Provide proof through completed work.	Project grid, optional filters, project summaries.	Open a project
Team	Show the people behind the work.	Leadership, team cards, standards, contact callout.	Contact ABC
Contact	Capture useful enquiries.	Contact details, quote form, service area, response expectation.	Submit request

Scope test

If Lovable adds a shop, careers portal, booking calendar, or customer dashboard, remove it. It was not approved. A polished extra is still scope creep.

Plan the page as sections before styling it

Homepage

- Header
- Hero + quote action
- Trust indicators
- Service cards
- Featured projects
- How we work
- Testimonials
- Final quote callout
- Footer

About

- Page hero
- Company story
- Values
- Safety and standards
- Team preview
- Contact callout

Services

- Page hero
- Service grid
- Individual summaries
- Process
- Service area
- Questions
- Quote callout

Contact

- Page hero
- Contact details
- Quote form
- Service area
- Response note
- Alternative contact

Design from priority. The hero should say what ABC builds, who it serves, and what the visitor should do. Decoration comes after comprehension.

Use the same process for Projects and Team. Start with the page job, list the sections, then define the information each section needs.

Decide what changes before you decide how it looks

A static interface displays values written directly in the code. A WordPress-driven interface reads content or settings that an administrator can change.

Element	Static or driven?	Reason
Decorative line and card shape	Static	It is part of the design system, not business content.
Primary navigation	WordPress-driven	The client may change menu items and order.
Project cards	WordPress-driven	New projects will be added after launch.
Company phone and email	WordPress-driven global setting	The values appear in several places and may change.
Homepage section order	Decide by scope	Do not make every layout choice editable unless the client needs it.
Footer background color	Static theme style	It should stay consistent with the approved brand.

If the client reasonably needs to change it later, plan a WordPress control for it. If changing it would break the design system, keep it in the theme.

Mark up your sitemap

Add one label beside every repeated or changeable area: Page content, Menu, Global setting, Custom post type, Taxonomy, or Theme style. Volume 2 will turn those labels into WordPress architecture.

Give Lovable decisions it can see

- 1 Context.**
Name the business, audience, location, trust signals, and primary action.
- 2 Pages.**
List every page and the purpose of each one. Do not leave the tool to invent a sitemap.
- 3 Global elements.**
Define the header, navigation, call-to-action button, footer, and mobile menu.
- 4 Page sections.**
List sections in order. Name the content and action inside each section.
- 5 Visual rules.**
Describe colors, typography, spacing, image treatment, card style, and desired feeling in visible terms.
- 6 Behavior.**
Require responsive desktop, tablet, and mobile layouts. Define menus, forms, cards, and galleries.
- 7 Limits.**
State what must not be added or redesigned.

Prompt rule. What changes + where it changes + required behavior + what must remain unchanged.

Start with a complete, bounded brief

Create a responsive multi-page website interface for ABC Construction, a residential and light commercial construction company.

Audience:

Homeowners, property managers, and small business owners who need a reliable contractor.

Primary action:

Request a Quote. Make this action clear in the header and at natural decision points.

Brand:

Use deep navy, construction orange, warm white, and charcoal. Use strong sans-serif headings, highly readable body text, generous spacing, and real-project image placeholders. The tone should feel capable, direct, safe, and trustworthy.

Pages:

Home, About, Services, Projects, Team, Contact.

Header:

Logo on the left, primary navigation, Request a Quote button, and an accessible mobile menu.

Homepage sections:

Hero with clear service promise and quote action; trust indicators; service cards; featured projects; simple working process; testimonials; final quote callout; footer.

Inner pages:

Use the approved page purpose and section plan supplied below. Keep a consistent header, footer, typography, card system, button system, and spacing scale.

Responsive behavior:

Create deliberate desktop, tablet, and mobile layouts. Keep tap targets clear, stack cards cleanly, preserve readable line lengths, and keep the quote action easy to reach.

Constraints:

Do not create pages, features, dashboards, shops, booking systems, or animations that are not specified. Do not change the business name. Do not add fake awards, statistics, certifications, addresses, or testimonials.

Do not try to perfect everything in prompt one

The first generation proves the structure. Later prompts should change one category at a time so you can see whether the change helped.

Pass	Review	Example instruction
1. Structure	Pages, sections, order, and missing content.	Keep the visual style. Add the approved safety section to About.
2. Layout	Width, balance, hierarchy, and alignment.	Make the featured projects section a clear three-card grid on desktop.
3. Typography	Scale, line length, contrast, and consistency.	Reduce heading size on mobile and keep body lines comfortably readable.
4. Responsive	Tablet and phone behavior.	Stack service cards at small widths and keep buttons full width only on phones.
5. Navigation	Menu clarity and mobile interaction.	Preserve the header design. Make the mobile menu obvious and keyboard reachable.
6. Components	Buttons, forms, cards, galleries.	Use one primary button style across every page.
7. Spacing	Rhythm, crowding, and empty areas.	Use a consistent section spacing scale and remove accidental gaps.

Do not combine unrelated fixes. “Improve everything” makes it difficult to know what changed and easier for the tool to damage an approved part.

An attractive homepage does not prove the site is ready

Page review

- ☐ The page has one clear purpose.
- ☐ The heading explains the page.
- ☐ Every section earns its place.
- ☐ The next action is obvious.
- ☐ No fake claims or invented facts appear.
- ☐ Navigation and footer stay consistent.

Responsive review

- ☐ Menu works at small widths.
- ☐ Text remains readable.
- ☐ Buttons are easy to tap.
- ☐ Cards stack in a sensible order.
- ☐ Images do not crop important content.
- ☐ Forms fit without sideways scrolling.

Conversion review

- ☐ Repeatable content is identified.
- ☐ Editable global values are identified.
- ☐ No essential feature depends on a visual mock only.
- ☐ Form fields and outcomes are defined.

Approval review

- ☐ Page list matches the scope.
- ☐ Brand direction is approved.
- ☐ Desktop and mobile are reviewed.
- ☐ Open questions are recorded.
- ☐ The client approves export.

ABC final interface check

Open Home, About, Services, Projects, Team, and Contact one by one. Follow every navigation item and call to action. Confirm that Projects looks repeatable, not like three permanently hard-coded cards.

Move an approved version, not a moving target

- 1 Finish the interface review.**
Confirm every page, responsive behavior, key component, and scope boundary.
- 2 Connect the intended GitHub account and repository.**
Use the repository setting that matches the client's privacy needs. Confirm the project name before syncing.
- 3 Sync or export the project.**
Follow the current Lovable interface. Product screens change, so rely on the available export or GitHub connection controls rather than an old screenshot.
- 4 Open the repository.**
Confirm that source files, assets, configuration, and page or component code are present.
- 5 Download or clone the repository.**
Place it inside the project workspace. Keep the original export unchanged and use a working copy for conversion.
- 6 Record the handoff.**
Note the approved version, unresolved items, and the next task: inspect before modifying.

Never assume the sync succeeded. Open the repository and verify the expected files before leaving Lovable.

The first coding instruction is to inspect, not rewrite

Once the project is downloaded, the AI coding environment needs the correct folder and a safe first task.

First inspect this entire project.

Do not modify anything yet.

Identify:

- the framework and build system
- entry points and routing
- pages and reusable components
- assets and styling system
- forms and interactive behavior
- static content areas
- repeated data areas
- likely WordPress-driven areas

Then explain how you would convert this interface into a native WordPress theme while preserving the approved visual design. List open questions before proposing changes.

Why inspection matters. The coding model must understand what already exists before it can choose a safe conversion path. A rewrite request without inspection can discard working components or invent a new design.

ABC handoff note

The exported interface contains six pages. Services, Projects, and Team appear as repeated cards. The header menu, contact details, and footer links must become administrator-controlled. No payments, portal, booking, or chat features are in scope.

You are ready for conversion when the decisions are stable

Requirements

- ☐ Business and audience are clear.
- ☐ Pages and navigation are approved.
- ☐ Each page has one job.
- ☐ Forms and actions have outcomes.
- ☐ Editable areas are identified.
- ☐ Exclusions are written.

Interface

- ☐ Every page is reviewed.
- ☐ Mobile and tablet layouts are reviewed.
- ☐ Components use one visual system.
- ☐ No fake facts or claims remain.

Handoff

- ☐ The approved version is in GitHub.
- ☐ The repository contains the expected files.
- ☐ A working copy is downloaded.
- ☐ The original export is preserved.
- ☐ Open questions are recorded.
- ☐ The AI is told to inspect first.

ABC Construction result

We now have a clear requirement document, a six-page interface plan, a reviewed Lovable project, and a stable GitHub export ready for WordPress conversion.

The end of interface design is not the end of the project. It is the point where visual decisions become a maintainable content system.

CONTINUE THE METHOD

In Volume 2: turn the interface into a real WordPress theme

The next volume begins with the downloaded project and converts it into native WordPress architecture without losing the approved design.

Inspect the project

Understand framework, routes, components, assets, styles, and dynamic areas before changing code.

Map the theme

Create the correct theme files, template parts, asset handling, and WordPress setup.

Make it editable

Replace hard-coded menus, settings, repeated content, and client-managed values with suitable WordPress controls.

Build ABC for real

Turn Services, Projects, and Team into maintainable content while preserving the approved interface.

Carry forward: the requirement document, sitemap, WordPress-driven UI labels, approved export, repository, and handoff notes.