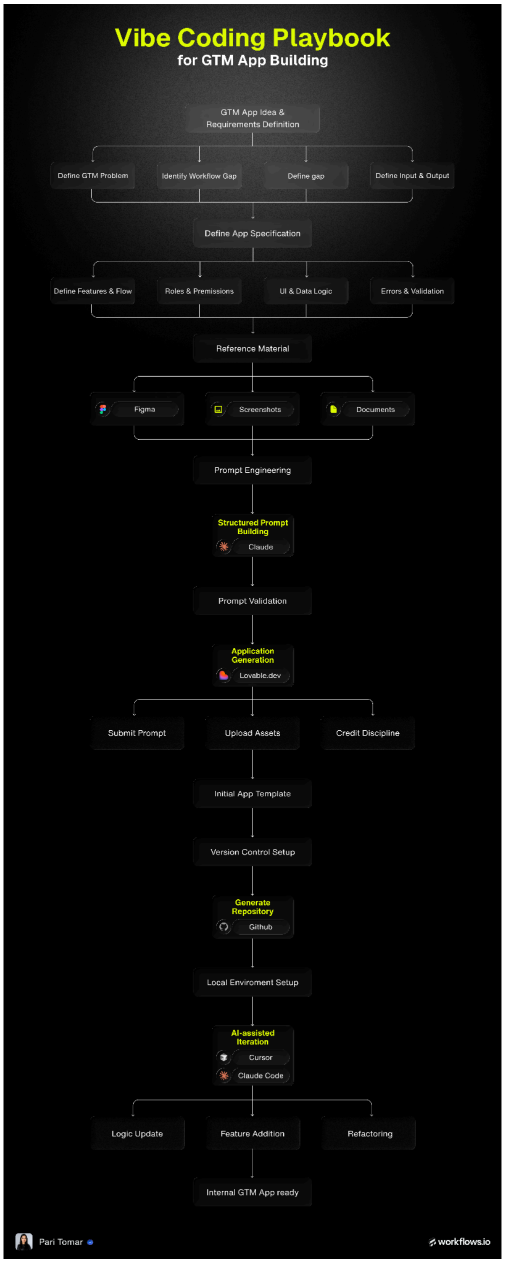
Vibe Coding Playbook for GTM App Building



Vibe Coding Playbook

for GTM App Building

Pari Tomar

workflows.io

How it works?

Step 1: Step 1: Define the GTM app problem

- Identify the GTM problem the app should solve
  Examples: routing, qualification, reporting, enrichment
- Identify the workflow gap in the current GTM process
- Define the data inputs and expected outputs of the app
- Clearly specify who will use the app and for what purpose

Step 2: Define the app specification

- Define all features and workflows
- Specify user roles and permissions
- Define UI behavior and data flow
- Define validations and edge cases
- Specify theme, layout preferences, and UX expectations

Step 3: Use reference material

Provide clear visual and functional context.

- Add Figma designs, if available
- Add screenshots of similar tools or workflows
- Add documents, notes, or written explanations

Step 4: Convert requirements into a structured prompt

Use Claude to prepare a detailed build prompt.

- Share all app requirements and reference material with Claude
- Ask Claude to generate one complete, structured prompt
- Make sure the prompt clearly defines:
  - Pages and components
  - Data models
  - Logic rules

    - Integrations
    - UI and layout instructions

Step 5: Validate the prompt

Review the prompt before submitting it.

- Confirm every feature is clearly described
- Check for missing flows or logic
- Ensure there are no instructions asking the tool to “decide” or “assume”

Step 6: Generate the MVP

Build the initial app using Lovable.dev.

- Paste the validated prompt into Lovable
- Upload all reference files and images
- Attach Figma designs, if available
- Use credits carefully:
    - Each request consumes credits
    - Avoid unnecessary iterations

Step 7: Set up the local environment

Prepare the project for iteration and maintenance.

- Connect the Lovable project to GitHub
- Generate a repository
- Copy the Git repository URL
- Clone the repository using Git
- Open the project folder locally in VS Code or Cursor

Step 8: Iterate using AI-assisted development

Use Cursor or Claude Code for changes and improvements.

- In Cursor, use the built-in chat to request changes
- In Claude Code, use the terminal inside VS Code and run `claude`
- Request updates such as:
  - Logic changes
  - Feature additions
  - UI improvements
  - Refactoring or cleanup

Step 9: Finalize the internal GTM app

Complete and validate the application.

- Test core workflows
- Confirm data accuracy and outputs
- Ensure the app is usable by the intended internal teams