「React is not defined」と言われて心が折れかけた

社長が、4 か月で社内ツール 11 個を作るまで

― プログラミング未経験の不動産会社経営者が、AI（Claude）と歩いた記録 ―

ViVi 不動産株式会社　代表取締役　矢郷 修治

富山県富山市／中古住宅・マンション売買仲介 ＋ リノベーション

はじめに：この文章を書く理由

富山市で中古住宅とリノベーションを扱う不動産会社、ViVi 不動産の矢郷と申します。エンジニアではありません。プログラミングを学んだこともありません。

その私が、いま社内で使っているツールは 11 個あります。決済案内書の自動作成、リノベ見積、資金計画シミュレーター、テレアポツール、買取再販の管理ボード、採用適性検査――全部、Claude（クロード）という生成 AI と会話しながら作りました。

先日、大阪の展示会で 45 分間このことを話す機会をいただきました。話し終わったあとに一番多かった質問が、これです。

> **「最初、どうやって始めたんですか？」**

正直に言います。**最初はまったく分かりませんでした。** 何を言われているのかも分からず、画面のどこを押せばいいのかも分からず、エラーが出るたびに固まっていました。

この文章は、その「まったく分からなかった頃」の記録です。うまくいった話だけを書いても役に立たないので、つまずいたところ、失敗したところ、いまだにできないことも、隠さずに全部書きます。

これから始めたい方が、「なんだ、この程度から始めていいのか」と思ってもらえたら本望です。

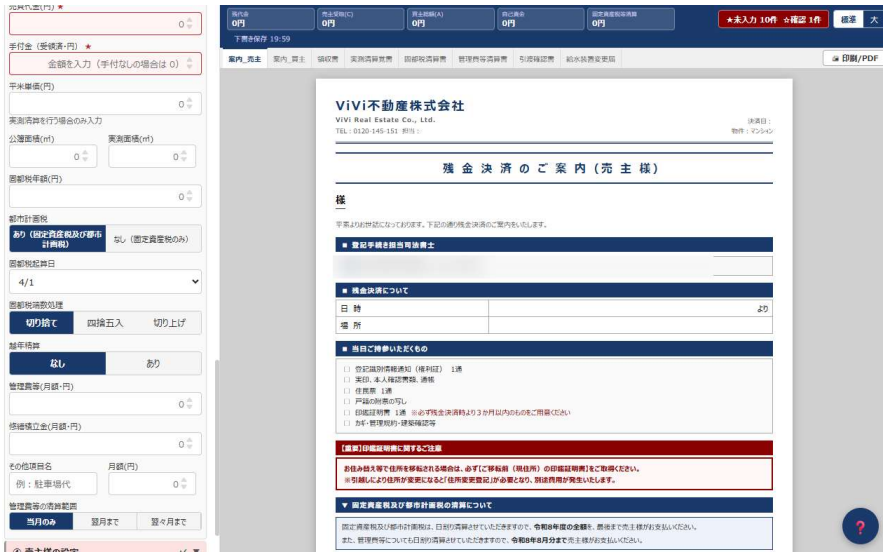
第 1 章　すべては「決済の清算書、毎回手作りなんだけど」から始まった

きっかけは業務の面倒くささ

不動産の決済（引き渡し）のとき、お客様にお渡しする清算書や案内書があります。固定資産税の日割り、管理費の精算、ローンの残債……毎回 Excel を開いて手で作っていました。担当者によって書式もバラバラ。　ミスも起きる。

「これ、自動でできないの？」

そう思って Claude に相談したのが 2026 年 4 月です。これが全部の始まりでした。



4 か月後。いま社内で使っている残金決済ツール（司法書士事務所名は伏せています）

ポイント：「AI で何かやってみよう」から始めていません。「この業務が面倒くさい」から始まっています。 これは今でも大事だと思っています。

最初の返答が、まったく理解できなかった

Claude は親切に（？）、こう答えてくれました。

> 1. この Artifacts のコードを Next.js / Vite プロジェクト として保存
> 2. GitHub にアップロード

> 3. vercel.com でそのリポジトリを連携するだけ

2026 年 4 月 7 日、実際の画面。「連携するだけ」と書かれています。

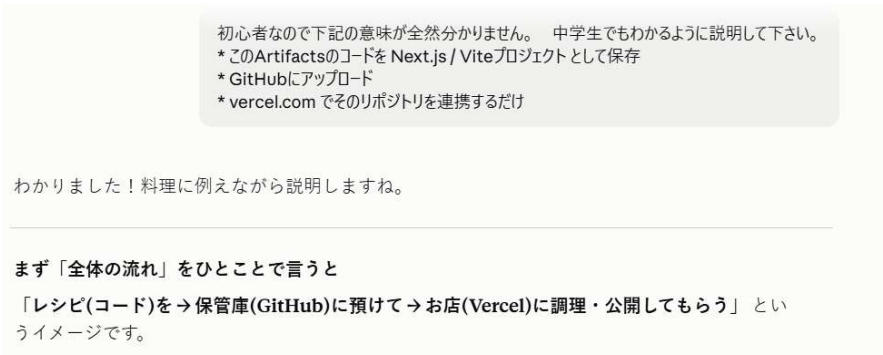
「だけ」って言われても、一つも分からない。

Artifacts？　Next.js？　Vite？　リポジトリ？　連携？

ここで多くの人が離脱すると思います。私も正直「あ、これは無理なやつだ」と思いました。

でも、ダメ元でこう返しました。

> **「初心者なので下記の意味が全然分かりません。中学生でもわかるように説明して下さい。」**

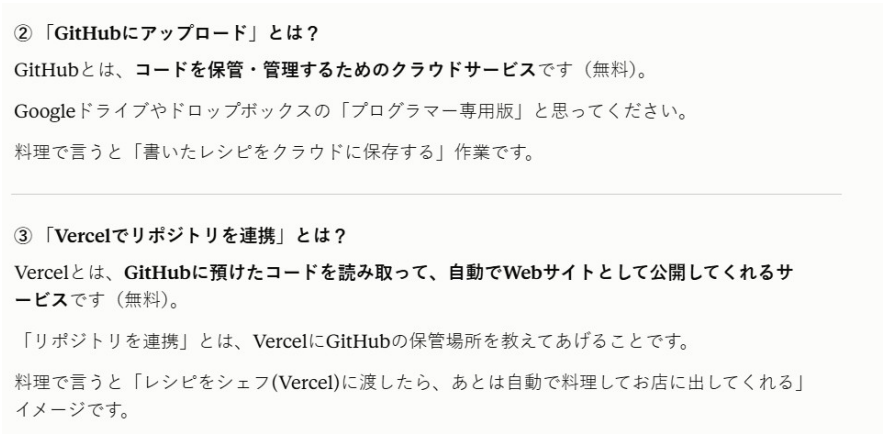


私の質問と、返ってきた答え。実際の画面です。

すると、こう返ってきました。

> 「レシピ（コード）を → 保管庫（GitHub）に預けて → お店（Vercel）に調理・公開してもらう」

> GitHub は、Google ドライブやドロップボックスの「プログラマー専用版」と思ってください。



続けて、*GitHub* と *Vercel* が何なのかも料理のたとえで説明されました。

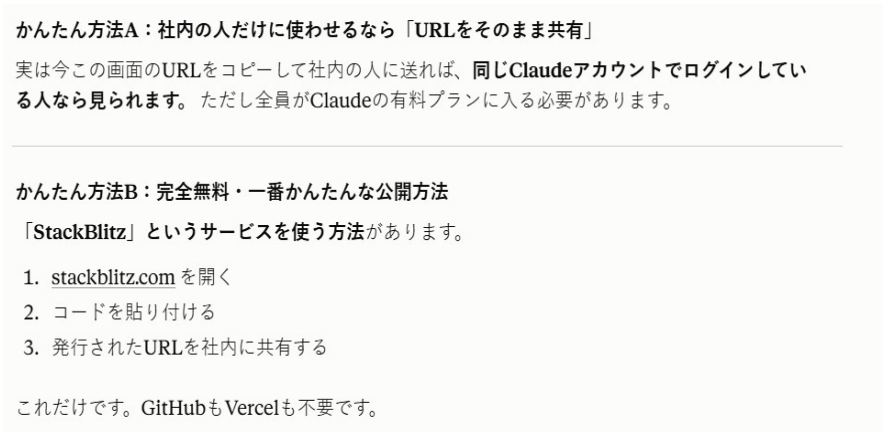
……分かった。急に分かった。

これが 1 つ目の学びです。分からないときは「分かりません」と言えばいい。 恥ずかしがる必要はまったくありません。相手は AI です。何度聞いても嫌な顔をしませんし、聞き方を変えて何度でも説明してくれます。

しかも、このとき Claude は「正直なところをお伝えすると、この 3 ステップは初心者の方には少しハードルが高いです」と言って、**もっと簡単な代替案を出してくれました。** 私の実力に合わせて道を変えてくれたわけです。

第 2 章　スクリーンショットを貼るまでの、地獄の数時間

最初に案内されたのは「StackBlitz（スタックブリッツ）」という、ブラウザ上でコードを動かせるサービスでした。GitHub より簡単、という理由です。



「*GitHub* も *Vercel* も不要です」と案内された場面。ここから数時間の格闘が始まりました。

ここからの私の質問を、そのまま並べます。**編集していません。これが本当の初心者です。**

- 「サインインしろって言われる」
- 「画面右上の『Share』ボタンが見つからない」
- 「これをどうやってストラックブリッツに張り付ければいいの？」（※スタックブリッツを打ち間違えている）
- 「『Start new project』→『HTML/JS/CSS』が見つからない」
- 「日本語表記の場合で教えて」
- 「新しいプロジェクト押したらこんな画面出ました。どれ選ぶの？」
- 「Error in /~/src/App.js (811:5) React is not defined って出ます」

この調子で延々とやっていました。1 時間経っても、画面には何も表示されていません。

転機：Claude の「白旗」とスクリーンショット

途中で Claude がこう言いました。

> 「申し訳ありません、StackBlitz の日本語版 UI を正確に把握していないため、誤った案内をしてしまう可能性があります。画面のスクリーンショットを撮ってここに貼り付けてください。今表示されている画面を見れば、正確にどのボタンを押せばいいか案内できます！」

半信半疑でスクショを貼りました。すると。

> 「画面が確認できました！　今開いているのは Angular プロジェクトなので、HTML ファイルを貼る場所が違います」

一発でした。それまでの 1 時間は何だったのか。

2 つ目の学び：詰まったら、言葉で説明せずにスクリーンショットを貼る。

これは今でも一番使っているテクニックです。「ボタンが見つからない」と文字で書くより、画面をそのまま見せたほうが 100 倍速い。Windows なら **Windows キー + Shift + S** で範囲を選ん

でコピー、そのままチャット欄に **Ctrl + V** で貼れます。

同じことがエラーメッセージにも言えます。「エラーが出ました」ではなく、**赤い文字を全部そのままコピーして貼り付ける。** 意味が分からなくても構いません。むしろ意味が分からないまま丸ごと貼るのが正解です。

> telapo/:582 Uncaught ReferenceError: buyBukkenScript is not defined at buildScript...

こういうのを、そのままドサッと貼る。これで直ります。

第 3 章　GitHub に引っ越して、やっと「型」ができた

なぜ GitHub なのか、そもそも聞いてみた

5 月に入って、StackBlitz では限界を感じ始めました。そこで改めてこう聞きました。

> 「GitHub はなぜ無料なの？　いろいろなツールをクロウド Ai で作って GitHub に入れて使ってるけどメリットデメリットそれぞれ教えて。」

（※「クロウド Ai」も打ち間違いです。それでも通じます）

返ってきた答えで納得したのは、**GitHub Pages という機能を使えば、HTML ファイルを 1 つ置くだけで無料で Web サイトとして公開できる**ということでした。社内の誰でも URL ひとつで使える。スマホからも開ける。インストール不要。

不動産会社の社内ツールとして、これ以上に都合のいい形はありませんでした。

私が最終的にたどり着いた「型」

いろいろ試した結果、いまはこの形に完全に固定しています。

項目	内容

ファイル形式	HTML ファイル 1 個だけ（分割しない）
置き場所	GitHub Pages（無料）
ファイル名	必ず index.html
外部サービス	原則なし（データ共有が必要なときだけ Google スプレッドシートの CSV）
更新方法	GitHub の Web 画面で鉛筆マーク → Ctrl+A → 貼り付け → コミット

この「型」が決まってから、スピードが一気に上がりました。

毎回違うやり方を試していた頃は、そのたびに調べ直しでした。型を決めてしまえば、Claude にも「いつものやり方で」と言えば通じます。実際いまは「これ GITHUB にあっぷするんだっけ？」くらいの雑な質問で話が通じます。

「HTML ファイル 1 個」にこだわる理由

普通のエンジニアなら、ファイルを機能ごとに分けます。そのほうが管理しやすいからです。

でも私は分けません。理由は単純で、**1 個なら「全部消して、全部貼り直す」で更新できるから**です。ファイルが 5 個あると、どれを直すのか、どこを直すのかを自分で判断しないといけない。素人にはそれが一番きつい。

「全消し・全貼り」なら、間違えようがありません。これは効率の話ではなく、**素人が事故を起こさないための設計**です。

第４章　正直に書く、つまずいたところ全部

「隠さなくていい」ということなので、恥ずかしいものも含めて全部書きます。**たぶんこの章が一番役に立ちます。**

つまずき①　画面に「404」と出て、真っ白になる

新しいツールを公開して、URL を開いたら真っ白な画面に「404」とだけ表示される。何度もやりました。

そもそも「404」とは何か。これはブラウザが出す番号で、「そのページは見つかりません」という意味です。ツールが壊れたわけでも、中身を書き間違えたわけでもありません。「言われた場所を探しに行ったけれど、そこには何も置いていなかった」というだけの状態です。

原因はほぼ毎回同じで、ファイル名が index.html になっていないか、公開の設定（Settings → Pages）をしていないかのどちらかでした。

公開の仕組みは、index.html という名前のファイルをそのページのトップとして表示する決まりになっています。別の名前を付けていると、URL の最後にファイル名を足さないと開けません。

> **対策：アップロードする前にファイル名を index.html に変える。例外なし。**

つまずき②　ファイル名が index.html.html になっていた

名前を変えるとき、うっかり「.html」を 2 回付けてしまい、ファイル名が index.html.html になっていたことがあります。当然、開けません。

見た目ではなかなか気づけないので、うまくいかないときはファイル名を最後の一文字まで確認するクセをつけました。

> **対策：うまくいかないときは、まずファイル名を疑う。**

つまずき③　画像を日本語ファイル名でアップして表示されない

リノベ診断ツールに間取り図を入れたとき、「マンション guest.png」のような日本語名のままアップしました。ツール側は英数字のファイル名を探しているので、当然どれも表示されません。

しかも「何も表示されない」だけでエラーも出ないので、原因にたどり着くまでに時間がかかりました。

> **対策：画像やファイルの名前は必ず半角英数。日本語・スペース・記号は入れない。**

つまずき④　ブラウザの更新（キャッシュ）で古い画面が出続ける

直したはずなのに変わらない。「Claude が直してないのでは」と疑ったことが何度もありますが、だいたいブラウザが古い画面を覚えているだけでした。

> **対策：Ctrl + Shift + R で強制再読み込み。それでもダメならシークレットウィンドウで開く。**

つまずき⑤　長すぎるコードを貼ると、AI の出力がおかしくなる

買取再販の管理ボードは機能が多く、HTML ファイルがかなり長くなりました。この全文をチャットに貼り付けると、Claude の出力が明らかにおかしくなるという現象が起きました（関係のない単語が混ざる、途中で崩れる）。

これは私の環境特有かもしれませんが、以来ルールを決めています。

> **対策：巨大なファイルは全文を貼らない。直したい部分だけを貼るか、「〇〇のボタンを押したときの計算がずれる」と言葉で説明する。それで十分伝わります。**

つまずき⑥　作ったけど使われなかったツール（失敗作）

会議の録音から議事録を自動生成するツールを作りました。仕組みとしては動きました。

でも**実務では使い物になりませんでした。** ブラウザの音声認識は、マイクとの距離、雑音、複数人が同時に喋る状況にとても弱い。不動産会社の会議室という現実の環境では、精度が足りなかったんです。

これは大阪の講演でも正直に話しました。**AI で作っても、うまくいかないものはうまくいきません。** 打率 10 割ではありません。

大事なのは、**ダメだと分かったら早めに諦めて、次に行くこと**だと思っています。1 個作るのに何日もかかるわけではないので、ダメでも損失は小さい。

つまずき⑦　こちらが口に出さなかったことは、AI には伝わらない

これは技術ではなく、頼み方の話です。そして、これが一番よく起きます。

以前、見積書に「コメント欄を付けてほしい」と頼んだことがあります。私の頭の中にあったのは、お客様にお渡しする書面に印刷される補足説明の欄でした。「この工事は別途お見積りになります」といったことを書き添えるための欄です。

ところが、出来上がってきたのは、社内の担当者だけが見る内部メモ欄でした。お客様には何も表示されません。

AI が間違えたわけではありません。**私が「誰が読むものなのか」を一度も言っていなかったから**です。「コメント欄」という言葉だけなら、お客様向けにも社内向けにも取れます。取り違えられて当然でした。

しばらく気づかないまま先に進めてしまい、後から作り直しになりました。

> **対策：頼むときに「これは誰が見るものか」を一言添える。それだけで、この種のズレはほとんどなくなります。**

第 5 章　これから始める人へ、私からの実践的なコツ

コツ①　「AI で何かする」ではなく「この業務が面倒くさい」から始める

私が作ったツールは全部、実際に社内で困っていたことが出発点です。

- 決済案内書を毎回手作りしていた → 決済案内ツール
- 再見積の変更点をお客様に説明しきれずトラブルになった → 見積比較ツール
- 資金計画で銀行ごとの商品が複雑すぎる → 資金計画シミュレーター
- 新人がテレアポで何を話せばいいか分からない → テレアポツール

「AI で何かできないかな」と考えている限り、たぶん何もできません。目の前の面倒くさいことを 1 個だけ選んでください。

コツ②　最初の 1 個は「自分だけが使うもの」にする

いきなり全社で使うものを作ろうとすると、失敗が怖くて手が止まります。まず自分だけが使う小さいものを作って、感覚をつかむのがいいと思います。

コツ③　分からないことは、分からないと言う

繰り返しになりますが、これが一番大事です。

- 「中学生でもわかるように説明して」
- 「その単語の意味が分かりません」
- 「どこを押せばいいのか、画面のどこか教えて」

これで機嫌を損ねることは絶対にありません。知ったかぶりをして進むほうが、はるかに遠回りになります。

コツ④　詰まったら、まずスクリーンショット

エラー、画面、設定、なんでもいいので貼る。言葉で説明しようとしない。

コツ⑤　一度に全部作ろうとしない

「あれもこれも」と最初に盛り込むと、どこが壊れているのか分からなくなります。

私のやり方は、まず一番シンプルな形で動かす → 実際に使ってみる → 気になったところを 1 個ずつ直す、です。実際、いま社内で使っているツールはどれも 20 回 30 回と修正を重ねています。最初のバージョンとは別物です。

コツ⑥　スタッフに使わせて、文句を言ってもらう

自分ひとりで完璧を目指すより、早めにスタッフに触らせるほうが圧倒的に良いものになります。

「ここが分かりにくい」「この欄が要らない」「この項目が足りない」——全部そのまま Claude に伝えれば直せます。現場の文句が、そのまま設計書になります。

コツ⑦　秘密情報はコードに書かない

無料プランで公開している場合、そのコードは世界中の誰でも見られます。お客様の個人情報、パスワード、API キーの類は絶対に書かないでください。

私のツールも簡易的なパスワードは付けていますが、あれは「社内の人が間違って開かないため」であって、本格的なセキュリティではありません。そこは割り切って、扱う情報の種類を選んでいます。

第 6 章　無料版と有料版のこと

講演のあとで一番よく聞かれるのが、この話です。正直なところを書いておきます。

無料版 ―「まず試してみる」には十分

ここまで書いてきたやり取りは、無料でも同じように始められます。まずはそこからで構いません。

ただし、一定量を使うと「しばらく待ってください」と止まります。ツール作りはコードのやり取りが長くなるので、この上限にすぐ当たります。作りかけのまま、その日は先に進めない、ということが起きます。

「面白い」までは行けますが、**業務で使い続けるには足りない**、というのが私の実感です。

有料版 ―「仕事で使う」ならこちら

使える量が大きく増えて、作りかけのまま止まらなくなります。ここが一番大きい違いです。

それから、**より性能の高いモデルを選べます。** 長い指示を出したときや、複雑なコードを渡したときの正確さが、はっきり変わります。ツールが大きくなってくると、この差が効いてきます。

ファイルの添付や、案件ごとに会話をまとめておく機能も使えます。結果として、気になったところをその日のうちに直しきれるので、翌日には現場で使えます。

私の失敗（笑ってください）

> **有料版を使うべきです。ちなみに私は、最初に間違えて年間 20 万円の契約をしてしまい、後に引けなくなりました（笑）**

冷静に考えれば、まず月払いで始めて、続きそうなら年払いに切り替えればよかった話です。

ただ、結果的には**引けなくなったおかげで毎日触ることになり**、4 か月で 11 個まで来ました。人にお勧めできる始め方ではありませんが、腹をくくるという意味では効きました。

料金と上限は、必ずご自分で確認してください

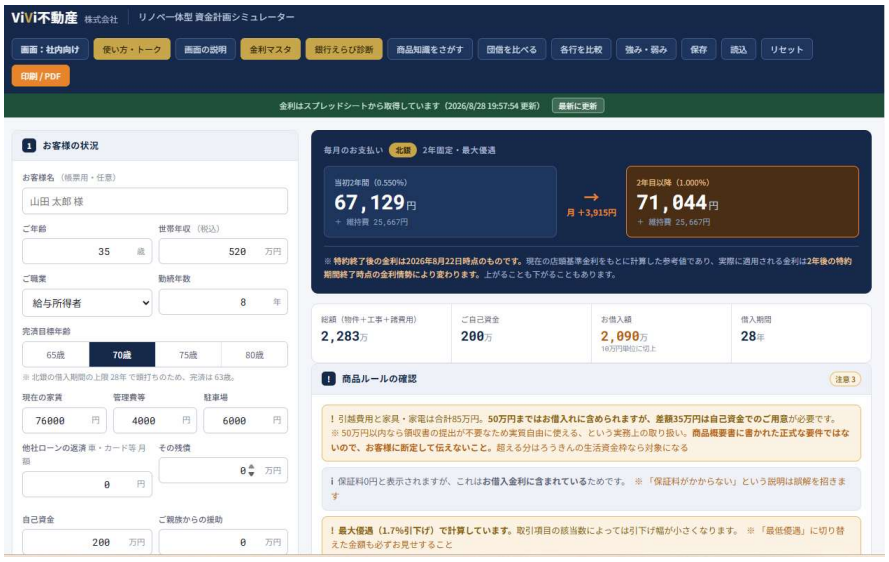
プラン名・料金・使える量の上限は改定されます。ここに具体的な金額を書いても、読まれる頃には変わっている可能性があります。始める前に、公式サイトで最新の内容をご確認ください。

第 7 章　4 か月やってみて、いま思っていること

できるようになったこと

社内で使うツールは、だいたい自分で作れるようになりました。数えたら 11 個あったので、社内用のポータルサイトも作って、そこから全部開けるようにしています。

社内ツールポータル。ここから全部のツールを開けます（単価とパスワードは伏せています）



資金計画シミュレーター。地元 3 行のローン条件を比較し、商談中にその場で試算できます。

外注していたら、1 個あたり数十万円〜、期間も数か月。それが実質ゼロ円と数日でできる。しかも、**「使ってみたらここが違った」を翌日に直せる。** この速さが一番の価値だと思っています。

いまだにできないこと

正直に書きます。

- コードは、いまでもほとんど読めません。書けません。
- なぜ動いているのか、原理は分かっていません。
- 大規模なシステム（本格的な顧客管理など）は作れません。既存のシステムを使っています。
- 複雑なデータ連携も苦手です。表計算ソフトを経由する程度が限界です。

つまり、**「エンジニアになった」わけではまったくありません。** できるようになったのは、「困っていることを AI に正確に説明して、出てきたものを現場で試して、直してもらう」というやり取りだけです。

でも、中小企業の社内ツールって、それで足りるんですよね。

一番変わったこと

技術より、こちらのほうが大きい変化かもしれません。

「これ不便だな」と思ったときに、諦めなくなりました。

以前は、社内の非効率に気づいても「まあシステム入れるほどじゃないし」で終わっていました。いまは「じゃあ今日作るか」になります。この差が、たぶん一番大きい。

おわりに：最初の一歩は、たぶん想像より低い

もう一度書きます。私の最初の質問は、これでした。

> 「初心者なので下記の意味が全然分かりません。中学生でもわかるように説明して下さい。」

そして途中の質問がこれです。

> 「サインインしろって言われる」

> 「Share ボタンが見つからない」

こんなレベルから始めて、4 か月で社内ツール 11 個です。特別な才能も、時間も、お金もかけていません。

もしこれを読んで「自分にもできるかも」と思われたら、今日、目の前の一番面倒くさい作業を 1 個思い浮かべてください。それをそのまま AI に話してみてください。専門用語も、正しい書き方も、要りません。

> **「〇〇の作業、毎回手でやってて面倒なんだけど、なんとかならない？」**

そこから全部始まります。

本記事は、*ViVi* 不動産株式会社 代表取締役・矢郷修治が自身の体験をもとに執筆したものです。特定のサービスを推奨するものではありません。また、記載の手順や画面表示は執筆時点のものであり、各サービスの仕様変更により変わる場合があります。

ViVi 不動産株式会社

富山県富山市／中古住宅・マンション売買仲介 ＋ リノベーション

※お問い合わせ先・URL は掲載媒体に合わせて追記してください