# OUR OBJECTIVE & THE COMPETITIVE LANDSCAPE

- At 38.2% CAGR, the AI-powered builder market is software development's fastest-growing segment, expanding from $4.9B (2024) to $24.8B (2029)
- **Lovable** - ***AI engineer*** *builds full-stack apps conversationally.*

**$200M ARR by November 2025**

**100,000 new products build per day**

**2.3M Active user**

**1,500 customers per day**

## Objective

- Develop a solution that **removes the friction between building (via templates or tutorials) and deploying** on Lovable.
- Accelerating activation, word-of-mouth, and monetization through new solution.

## Business Model

Freemium Subscription with Usage-Based Pricing.

Usage-based pricing for AI generation tasks (basic games cost ~$1 in credits, complex applications can cost $50+)

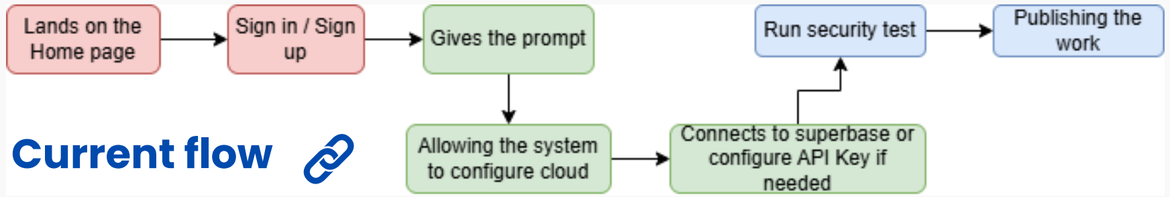
## Competitor Analysis

| Feature | Lovable | Replit | Bolt | V0 |
|---|---|---|---|---|
| Edit code directly | ✗ | ✓ | ✓ | ✓ |
| Ease integration of API key | ✗ | ✗ | ✗ | ✗ |
| AI testing agents | ✗ | ✓ | ✗ | ✗ |
| Autoscaling shown | ✗ | ✓ | ✗ | ✓ |
| One-click deploy | ✓ | ✓ | ✓ | ✗ |
| Planning mode | ✓ | ✓ | ✓ | ✓ |
| No code Friendly | ✓ | ✗ | ✓ | ⚠ |
| Full IDE | ✗ | ✓ | ✗ | ✗ |

## Key Actor

- **Primary Users -(Customer)**
- **AI Model Providers** - Open AI, Google, Groq, Cohere Command R+
- **Integration Partners** - Superbaase, Stripe, GitHub, Figma, Entri.
- **Framework Providers** - React and Vite

## Current flow



# VOICE FROM THE FIELD: SURVEY INSIGHT THAT MATTERS

## Hypothesis

- Users avoid deploying because they cannot **validate whether their projects are secure, tested, and ready for real users**.
- Users **abandon projects because backend setup** difficult and complex to do.
- Users **avoids developing backend logic** because they feel lovable is not capable of doing it .

## User Insight from primary data: Survey link

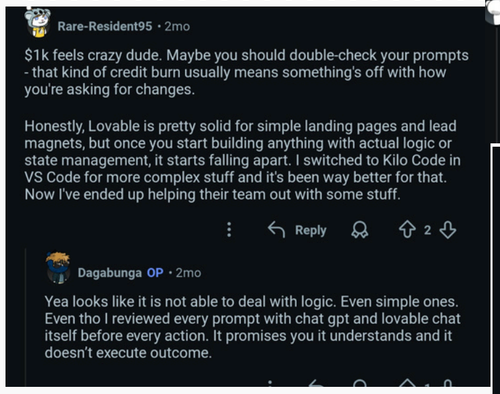
- Total responses collected through survey - 41
- 87.8% have used Lovable.ai.
- 40.5% of the user have built 2-3 project in last 3 months.

Successfully Deployed

Started but abandoned partway

Build but never Deployed

Still working on it


Rare-Resident95 • 2mo

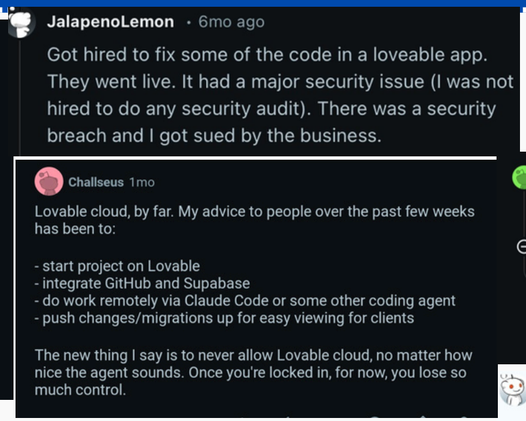
$1k feels crazy dude. Maybe you should double-check your prompts - that kind of credit burn usually means something's off with how you're asking for changes.

Honestly, Lovable is pretty solid for simple landing pages and lead magnets, but once you start building anything with actual logic or state management, it starts falling apart. I switched to Kilo Code in VS Code for more complex stuff and it's been way better for that. Now I've ended up helping their team out with some stuff.

Reply 2

Dagabunga OP • 2mo

Yea looks like it is not able to deal with logic. Even simple ones. Even tho I reviewed every prompt with chat gpt and lovable chat itself before every action. It promises you it understands and it doesn't execute outcome.


JalapenoLemon • 6mo ago

Got hired to fix some of the code in a loveable app. They went live. It had a major security issue (I was not hired to do any security audit). There was a security breach and I got sued by the business.

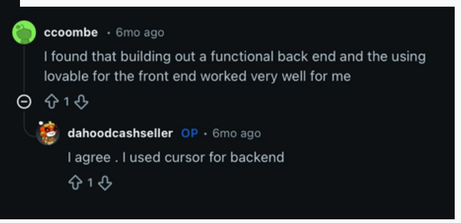
Challseus 1mo

Lovable cloud, by far. My advice to people over the past few weeks has been to:

- start project on Lovable
- integrate GitHub and Supabase
- do work remotely via Claude Code or some other coding agent
- push changes/migrations up for easy viewing for clients

The new thing I say is to never allow Lovable cloud, no matter how nice the agent sounds. Once you're locked in, for now, you lose so much control.

## Secondary Research


ccoombe • 6mo ago

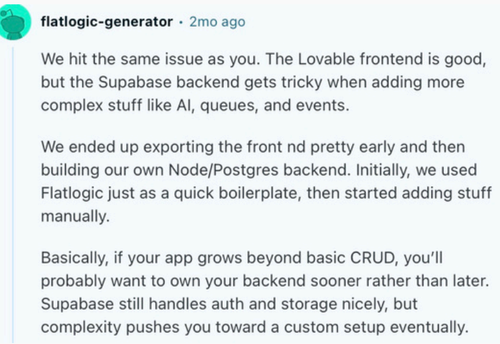
I found that building out a functional back end and the using lovable for the front end worked very well for me

1

dahoodcashseller OP • 6mo ago

I agree . I used cursor for backend

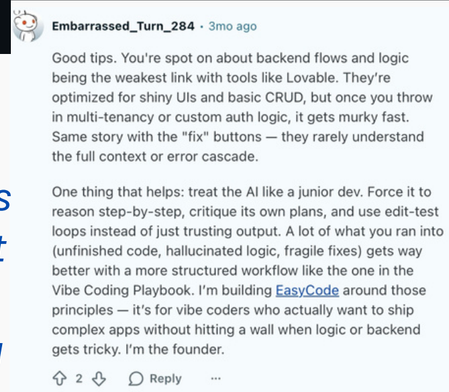
1


flatlogic-generator • 2mo ago

We hit the same issue as you. The Lovable frontend is good, but the Supabase backend gets tricky when adding more complex stuff like AI, queues, and events.

We ended up exporting the front nd pretty early and then building our own Node/Postgres backend. Initially, we used Flatlogic just as a quick boilerplate, then started adding stuff manually.

Basically, if your app grows beyond basic CRUD, you'll probably want to own your backend sooner rather than later. Supabase still handles auth and storage nicely, but complexity pushes you toward a custom setup eventually.


Embarrassed_Turn_284 • 3mo ago

Good tips. You're spot on about backend flows and logic being the weakest link with tools like Lovable. They're optimized for shiny UIs and basic CRUD, but once you throw in multi-tenancy or custom auth logic, it gets murky fast. Same story with the "fix" buttons — they rarely understand the full context or error cascade.

One thing that helps: treat the AI like a junior dev. Force it to reason step-by-step, critique its own plans, and use edit-test loops instead of just trusting output. A lot of what you ran into (unfinished code, hallucinated logic, fragile fixes) gets way better with a more structured workflow like the one in the Vibe Coding Playbook. I'm building EasyCode around those principles — it's for vibe coders who actually want to ship complex apps without hitting a wall when logic or backend gets tricky. I'm the founder.

2 Reply ...

*"Users want powerful backend, security, and logic clarity, but Lovable's backend tools and agent logic are not reliable or intuitive enough, pushing users to external platforms before deployment."*

**Insights of secondary research**

## Major findings:

- 78.6% uncertain if projects are production-ready enough.
- 38% anxious about testing adequacy and product quality.
- 48.6% of the user felt that superbase configuration is complex.
- Out of 35% still working user 38% stuck at testing whether the project is production ready.
- Among 27% of successfully deployed user 50% of user faced no issues while deploying.

# KNOW YOUR AUDIENCE: SEGMENTATION & PERSONA PROFILES

Aspiring builders with low to moderate technical experience (i.e students, entrepreneurs, and product managers) who are exploring AI-powered development tools to create real products but lack the expertise, clarity, or confidence needed to deploy them to production.

**Why this segment?**

- Large & Growing Market - Expanding no-code/low-code movement.
- Willing to explore new AI tools and provide feedback for product improvement.
- Marketing campaigns, portfolio projects, MVPs—continuous need for building and deploying.

- Backend logic becomes automatic & reliable.
- Deployment friction disappears (testing + readiness automated).
- Users trigger more backend, testing, and deployment AI tasks
- Consumption of high-credit project development increases.

**Lovable's usage-based revenue grow**

**Job to be done**

*"When I have an idea for a digital product, I want an easy way to build and deploy it end-to-end including the logic, backend, and integrations without technical expertise, so I can validate, showcase, or launch my business independently."*

**SREE; Age: 27 (M) ; Marketing executive**

*"I want to move fast and launch campaigns independently, not wait weeks for developers."*

**Needs:**

- Rather relaying on the developer need develop campaign page real quick.
- Need production checklist and quality testing.

**Pain points:**

- Cannot validate if production-ready.
- Quality anxiety - will it crash?
- Whether it withstand the user traffic.

**Vani ; Age: 25 (F) ; Business Owner**

*"I need the app to work smoothly because slow performance affects my business operations."*

**Needs:**

- To develop my MVP to create a inventory management.
- Need the solution to be optimized for reliable, high-performance operations.

**Pain points:**

- Third- party integration are complex and confusing.
- Need to click through the entire app repeatedly, missing hidden flows, edge cases, and regressions.

# THE PROBLEM CANVAS

## What is the true problem?

Users experience a two-stage failure pattern with Lovable: First, while the platform delivers excellent frontend development, it falls short on backend logic implementation like user-based permissions and calculation-driven features. Second, even when users complete their builds, 72% user not deploy due to database (superbase) configuration complexity (48.6%), production-readiness uncertainty (78.6%) and anxious about testing adequacy (38%).

## How do we know it's a real problem?

**User Insights :** Of 41 responses. it indicates user are actively building their ideas but only 30% of them deployed their work. Secondary review validates the short fall of backend -logic.

**Barrier analysis:** Users face key blockers in production-readiness uncertainty (78.6%), database (superbase) configuration complexity (48.6%) and even 31.7% of basic programmers struggle, showing the issue extends beyond coding skill.

## Why should we solve this problem NOW?

**Current user base is highly engaged but not converting:**

- 87.8% try Lovable, 29.7% build never deployed but only 27% of deploy
- 64% stuck in "frozen at 90%" state
- 86.1% remain on free tier

**Competitor pressure:**

- "Best path is to dev frontend on Lovable, then move to Cursor for backend" (Source: Reddit user Beginning-Willow-801)

**Positioning risk:**

- Great for prototyping, NOT production" (Source: Reddit r/lovable, recurring theme across multiple threads)

## What value will be generated by solving this problem?

**For the User**

- More users successfully ship live applications instead of getting stuck at 80–90%.
- Clear, guided deployment and backend experience that reduces confusion and fear.

**For Business**

- Higher Activation rate - Increasing deployment rate to 60-70% which influence the LTV of the user.
- If deployment success increases from 27% to even 60%, this could double the conversion rate to ~15%
- Revenue impact- With 1.40 million target users, each 1% increase in conversion = 14,000 additional paying customers × $20-100/month = $280K - $1.4M additional MRR ($3.36M–$16.8M ARR).

# THE FIX: OUR SOLUTION & WHAT WE'RE BUILDING FIRST

### Testing Agent

An autonomous AI agent that automatically creates an isolated sandbox environment, comprehensively tests the user's project with live navigation through the flow identifies bugs, and auto-fixes issues and provides the confidence score. Also, user will be able to invite others to test their solution usability before deploy the application to live. This agent cover scecurity and load testing too.

### Portfolio-First Onboarding Flow

New users begin their journey by creating a personalized portfolio. Upon landing on the homepage, they encounter a pre-populated "Build a Portfolio" prompt that initiates the onboarding flow by providing their profession after registaration. Once they enter the platform, a guided popup helps them build their portfolio using profession-specific templates and prompts, culminating in their first deployment.

### Logical Agent

Once system plans and develop the product as per the specification provided by the user. The user can selects the 'Logical Agent' and provide their logical specification. System interprets the context of the specification and moves to analyze the folder that are developed .Provides the best logical code, user will able to see the generated logical code with the code preview option. This agent can also act as the connectionhub if any integration needed, they will automatically get all the Open source API key that we are partner with rather moving to another system

**RICE FRAMEWORK** **RICE = (Reach x Impact x Confidence) / Efforts**

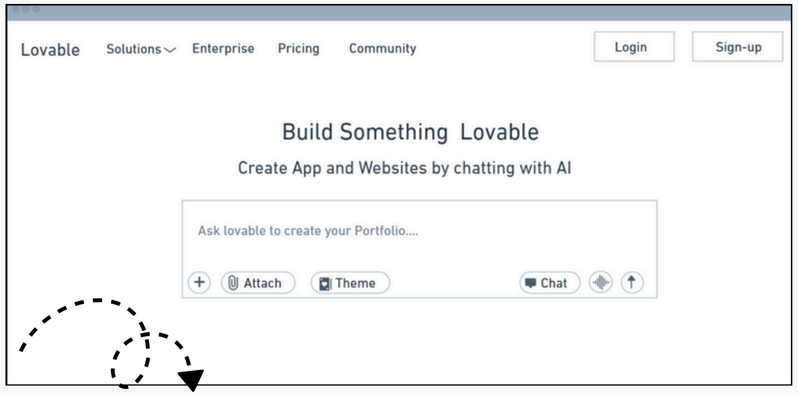
| Feature | Reach | Impact | Confidence | Effort | RICE Score | Priority |
|---|---|---|---|---|---|---|
| Logical Agent | 4 | 5 | 3 | 2 | 14 | P1 |
| Testing Agent | 5 | 5 | 4 | 3 | 17 | P0 |
| Portfolio-First Onboarding Flow | 5 | 4 | 4 | 4 | 17 | P0 |

### How does this solves the problem?

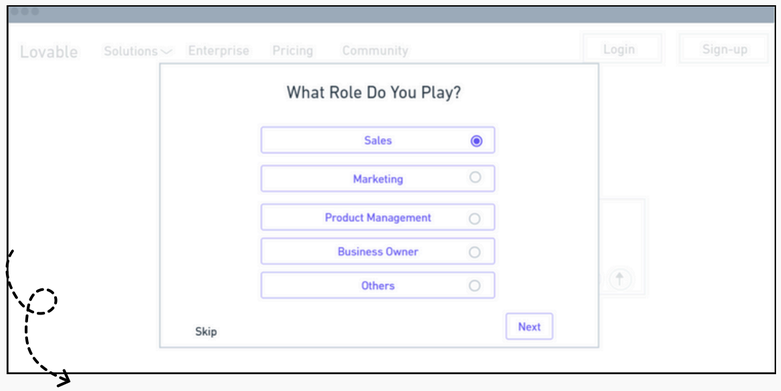
- Reduces production-readiness uncertainty (78.6%) through confidence scores and eliminates the 38% testing adequacy concern.
- Analyzes user specifications in context of existing code, then generates the missing logical components.
- Builds confidence through a working example before they tackle complex custom projects.

# Blueprint to Reality: Wireframe Walkthrough

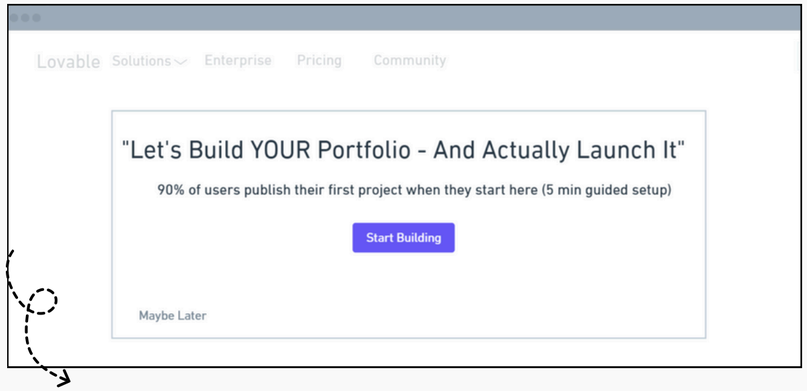
View Complete Prototype



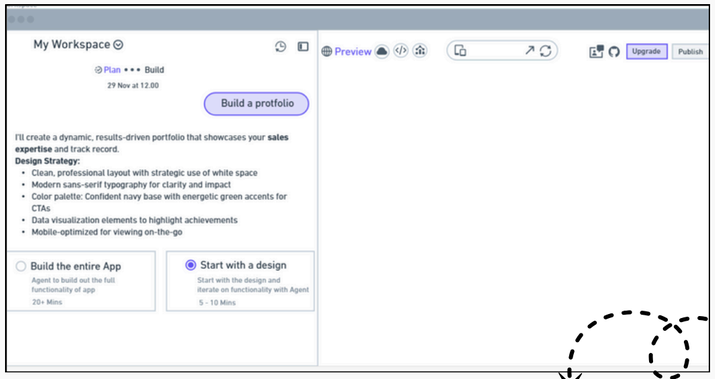

*"On landing at the Lovable website, users see the prompt 'Ask Lovable to build your portfolio.' Continuing the flow takes them to the sign-up page."*



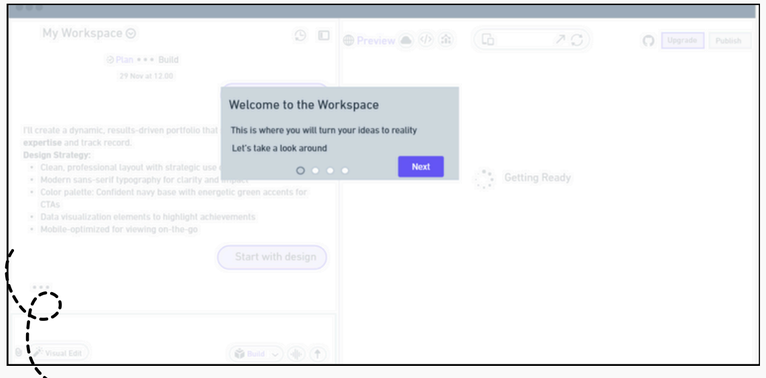

*"Post-registration, the user submits their professional role for lead validation."*



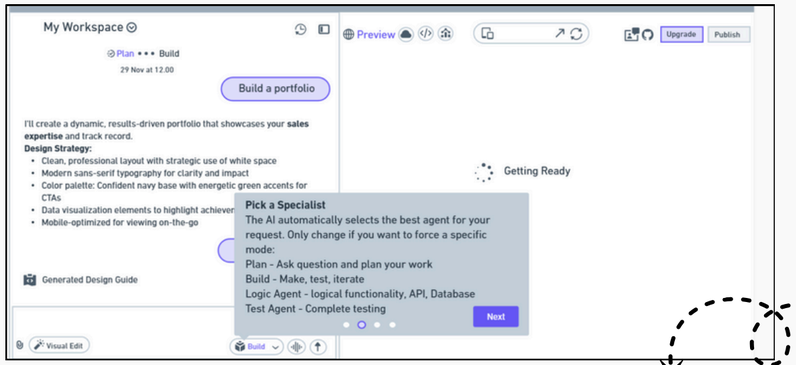

*"A portfolio-building pop-up appears for new users and those without live deployments, with a skip option. If user skips and after 15 days of inactivity, the pop-up resurfaces when they log in again."*



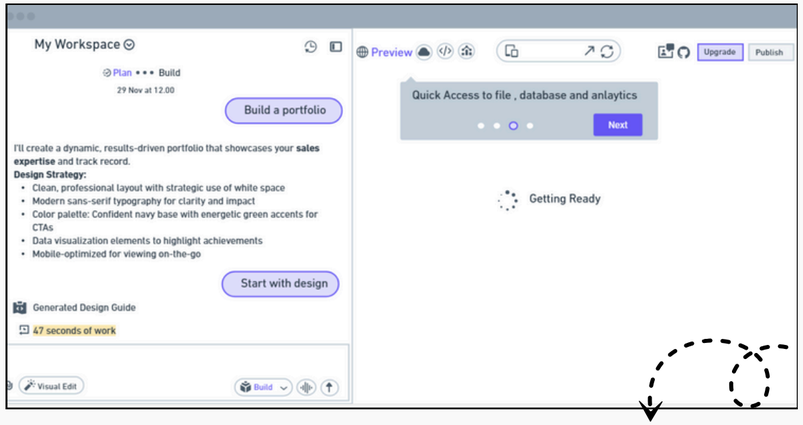

*"Each time the user prompts the chat, they can choose between full AI app generation or basic UI generation with user-driven enhancements. This ensures control and flexibility in shaping their ideas."*



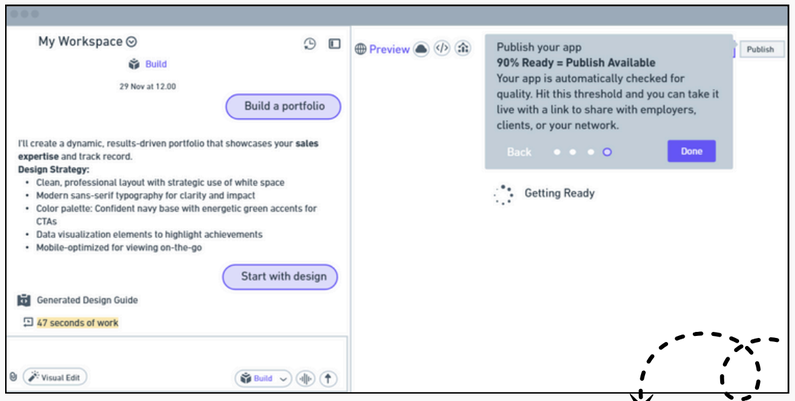

*"New and non-deploying users will automatically receive a guided tour, and a guide button will remain accessible to help them explore Lovable's features."*



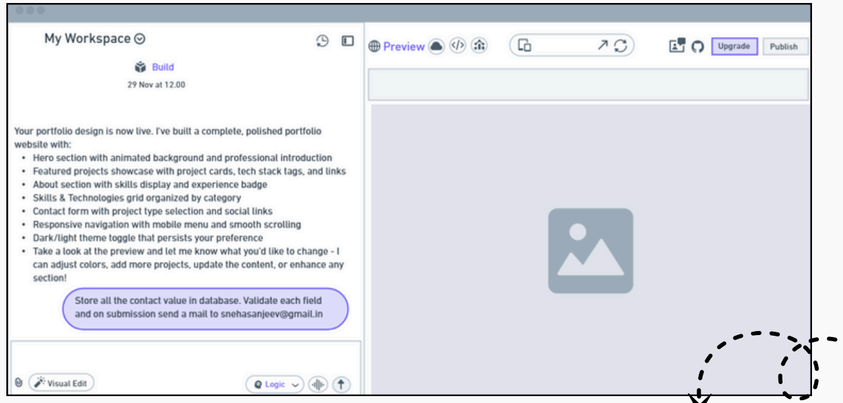

*"The toolkit features a planning agent, a high-level build agent for UI, logic, and unit tests, a logic-focused agent for complex tasks, and a test agent that performs live, security, and load testing to produce a confidence score."*



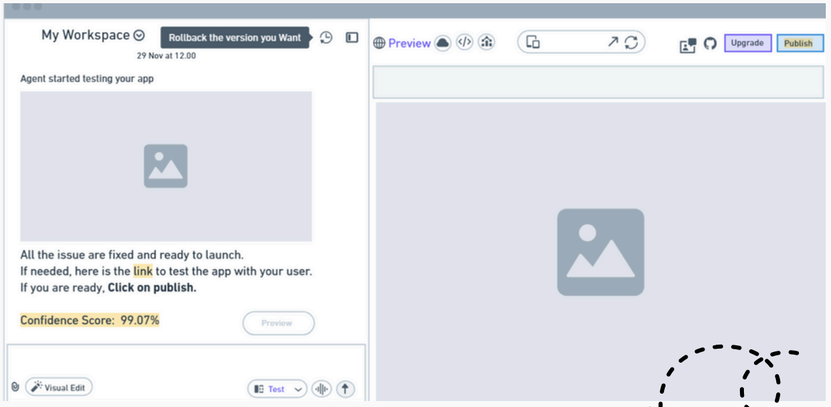

*"Displays ongoing tooltips that explain the file preview, security dashboard, and code files to guide the user."*



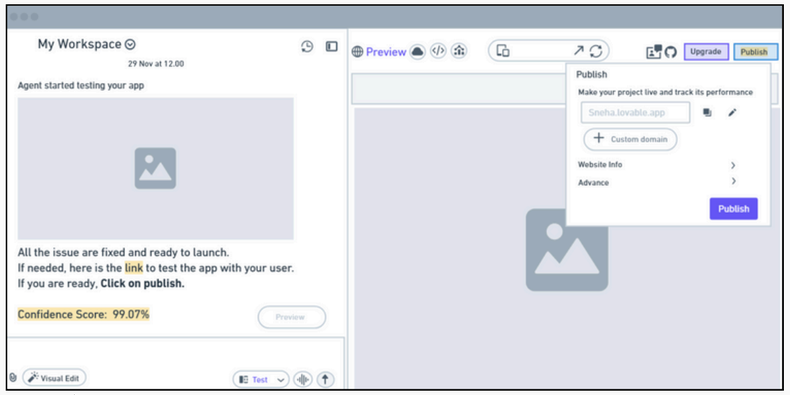

*"The tour concludes by highlighting the deployment flow—where the publish button unlocks only when the app reaches an 85% confidence score. User need to reinitiate testing and once the vulnerability fixed.They can proceed"*

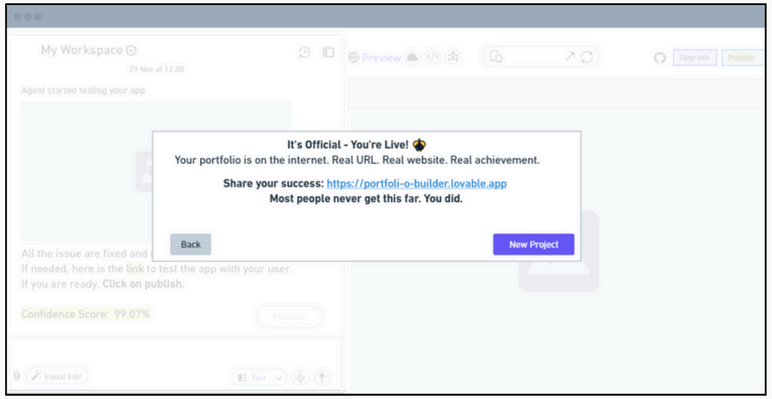



*"Users can enable the Logical Agent to add complex calculations or logic flows, and use its capabilities to integrate any open-source API without leaving Lovable."*

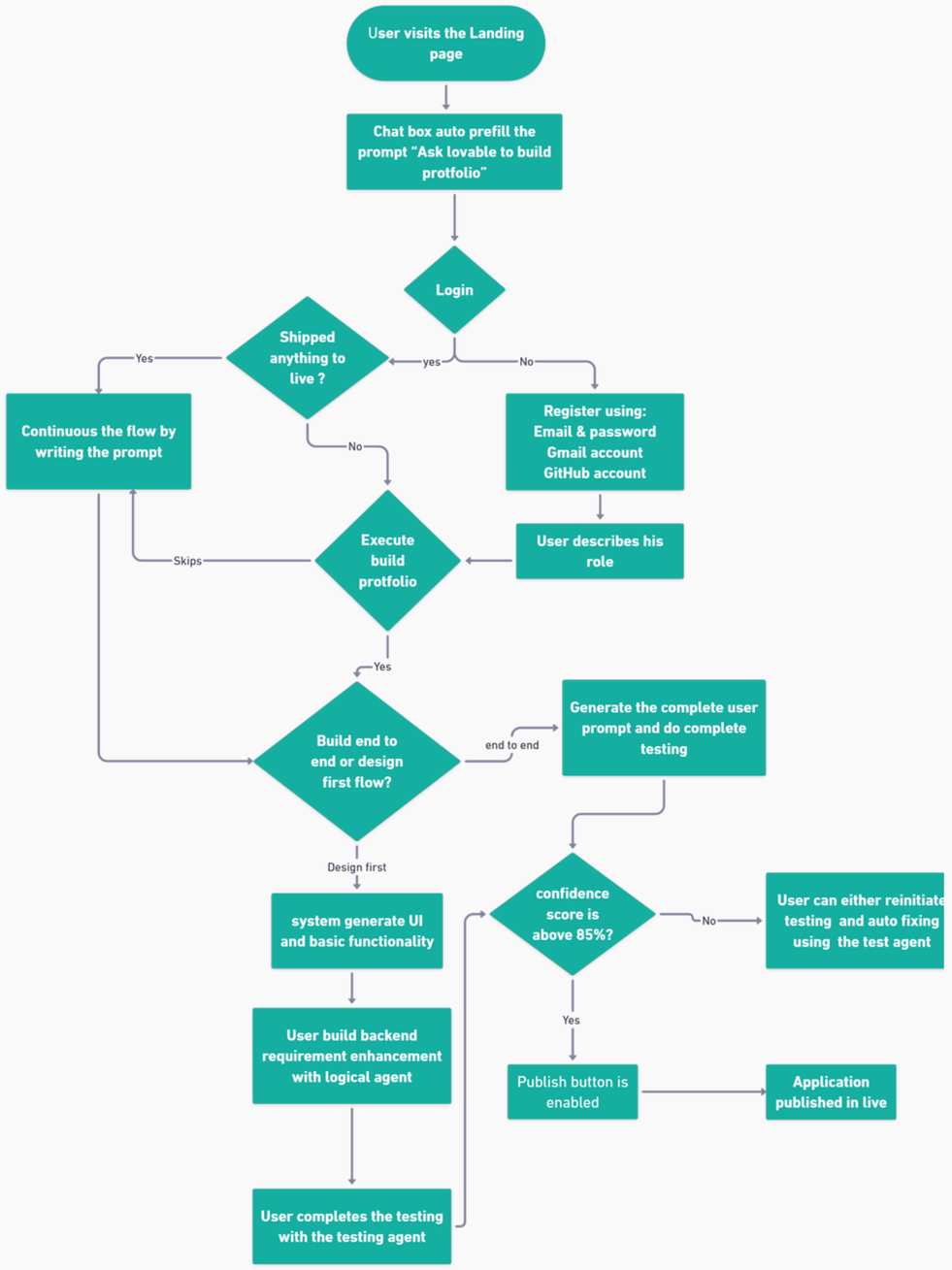



*"The Testing Agent performs live end-to-end testing, navigating the app to detect and automatically fix issues while running usability, security, and load tests.It then generates a detailed confidence report and provides a shareable staging link for external feedback before deployment."*




*"Once the Publish button is enabled, a tooltip informs the user that the app is ready to go live on a Lovable domain or a custom domain. When the user clicks Publish, the application is deployed to the live environment."*




**TO-BE USER FLOW DIAGRAM**

| Type of Metric | Metric |
|---|---|
| North Star Metrics | % of users who successfully deploy within 7 days of starting a project |
| Adoption Metric | Testing Agent Adoption = (Users who ran Testing Agent in last 7 days / Active users who created >= 1 project in last 7 days) × 100<br>Logical Agent Adoption = (Users who ran Logical Agentin last 7 days / Active users who created >= 1 project in last 7 days) × 100 |
| Acquisition Metric | New Signups = Count of new user registrations in the last 7 days |
| Activation Metric | Portfolio Completion Rate = (Users who deployed first portfolio / Users who clicked "Start Building Protfolio" button) × 100 |
| Conversion Metric | Portfolio CVR = (Users who completed their first portfolio within 7 days of finishing onboarding / Users who completed onboarding) × 100<br>Test to Deploy = (Users who deployed after utilizing testing agent / Users who used testing agent) × 100 |
| Engagement Metric | Logical Agent Engagement =(Users who used Logical Agent >= 1 time in last 7 days / WAU) × 100 |
| Outcome Metrics | Confidence Achievement Rate = (Users who reached >= 80% test pass rate in last 7 days /<br>Users who ran Testing Agent >= 3 times in last 7 days) × 100<br>Portfolio Sharing Rate = (Portfolios shared publicly / Total portfolios) × 100 |
| Business Impact Outcomes | Deploy Velocity = Total successful deployments in last 30 days / MAU |

## RISK AND MITIGATION

| Feature | Risk | Mitigation |
|---|---|---|
| Testing agent | **Slow testing (>2 minutes)** - Users abandon during wait, defeating activation purpose.<br>**False confidence** - AI gives 94% score but project crashes in production.<br>**Sandbox environment ≠ production** - Tests pass in sandbox but fail when deployed live. | **Cache common tests :** Store results for standard checks (security, speed) so repeat tests run instantly.<br>**Conservative scoring + disclaimer:** Penalize unknowns heavily, show "tested in sandbox only" warning, monitor post-deploy crashes.<br>**Mirror production exactly:** Use identical infrastructure, same database, same API endpoints as live deployment. |
| Protfolios first onboarding | **Generic AI portfolios** look alike - All portfolios look same, no differentiation. | **5+ themes + profession layouts:** Offer minimalist/bold/creative designs, tailor layout to job type (designer portfolio ≠ developer portfolio), easy color/section customization. |
| Logical agent | **AI generates insecure code** - vulnerable backend code, causing data breaches and legal liability. | **Security scanning + whitelist + manual review:** Auto-scan all generated code, only allow proven-safe patterns, require human review for auth/payments, penetration test before launch. |

**DETAIL SEQUENTIAL DIAGRAM ON THE SYSTEM ARCHITECTURE**

## LAUNCH STRATEGY

***Phase 1: Internal Testing (Weeks 1-2)***

***Focus Area:*** *Validate core functionality, fix critical bugs, and ensure AI accuracy before any user sees features.*

**Phase 2: Closed Beta (Weeks 3-6)**

**Focus Area:** Gather qualitative feedback from 50 diverse real users to identify usability issues and refine user experience.

**Phase 3: Freemium Pilot Rollout (Weeks 7-10)**

**Focus Area:** Test real-world adoption, retention impact, and measure quantitative metrics (deploy rate, conversion) with 20% of users.

**Phase 4: Data Review & Optimization (Weeks 11-12)**

**Focus Area:** Analyze pilot data, fix top friction points, calibrate AI accuracy, and refine experience before full-scale launch.

**Phase 5: Full Public Launch (Month 4 onward)**

**Focus Area:** Scale features to 100% of users globally, execute marketing campaign, and position as Lovable's core differentiator.

**Phase 6: Monetization Rollout (After 5+ Months)**

**Focus Area:** Convert proven free users to paid premium tier by offering advanced features after demonstrating clear value delivery.