

# ASSESSMENT BRIEF

## AI / Growth Engineer (Vibe Coder)

*Build a Working AI-Powered Marketing Tool From Scratch*

<b>Company</b>	upGrowth Digital LLP
<b>Role</b>	AI / Growth Engineer
<b>Assessment Type</b>	Technical build assessment
<b>Duration</b>	5 calendar days from date of issue
<b>Estimated Effort</b>	3-5 hours of focused work
<b>Compensation</b>	Unpaid (you own the output as a portfolio piece)
<b>Submission</b>	GitHub repo link + live deploy URL to <a href="mailto:amol@upgrowth.in">amol@upgrowth.in</a>

## Context

upGrowth is a growth marketing consultancy that builds AI-powered tools for internal use and client delivery. We ship fast. Our stack leans heavily on AI APIs, modern frontend frameworks, and rapid deployment platforms like Vercel and Railway.

The AI / Growth Engineer role sits at the intersection of marketing knowledge and technical execution. You won't get detailed PRDs or pixel-perfect Figma files. You'll get a problem statement and an afternoon. The expectation is that you can go from 'this would be useful' to 'here's the live URL' using AI-assisted development tools, pre-built components, and good judgment about what to build versus what to skip.

This assessment is designed to mirror that reality. We're giving you a problem. You decide the architecture, the UI, the AI prompt engineering, and the deployment approach. There's no single correct answer.

## The Assignment

### Build an AI-Powered Blog Content Repurposer.

The tool takes a blog post URL as input, extracts the content, and uses an AI model to generate repurposed content assets. Specifically, the tool should:

1. **Accept a blog URL** as input (paste or type)
2. **Extract the blog content** programmatically (scraping, API, or any method that works)

3. **Send the extracted content to an AI model** (Claude API, OpenAI, Gemini, or any LLM of your choice)
  4. **Generate and display** at least these four output types:
    - **3 LinkedIn post variations** (different angles: educational, controversial take, personal story hook)
    - **3 Twitter/X thread hooks** (first tweet of a thread, each with a different framing)
    - **1 optimized meta description** (under 160 characters, SEO-focused)
    - **1 YouTube video title + description** (if this blog were turned into a video)
1. **Present the outputs in a clean, usable UI** where someone can copy individual pieces
  2. **Deploy it live** on any free-tier platform (Vercel, Netlify, Railway, Render, anything)

That's the entire spec. Everything else is up to you.

## What Each Element Tests

### URL Input and Content Extraction

**This tests your problem-solving instinct.** Blog URLs come in a hundred different formats. WordPress, custom CMSes, single-page apps, server-rendered pages. A strong candidate picks an extraction approach that handles edge cases gracefully instead of breaking on the first non-standard page. You might use Cheerio, Puppeteer, a readability library, or an API. The approach matters less than whether it actually works on real-world URLs.

### AI Integration and Prompt Engineering

**This tests whether you can get useful output from an LLM.** Anyone can call an API. The difference shows up in prompt design. Do your LinkedIn posts sound like they were written by a human who actually uses LinkedIn, or do they sound like ChatGPT? Do your Twitter hooks have genuine tension and curiosity, or are they generic 'Here are 5 things about X' templates? The quality of AI output is a direct reflection of prompt engineering skill.

### Frontend and UX

**This tests your sense of what 'done' looks like for a user.** We're not evaluating design chops. We're checking whether you think about the person using the tool. Can they copy a single LinkedIn post without selecting text manually? Is there a loading state while the AI processes? Does it handle errors without crashing? These details reveal whether you ship products or just write code.

## Deployment

**This tests whether you can get something live.** A local demo is not a submission. The tool must be accessible via a public URL. We don't care which platform you use. We care that you can take code from your machine to a working deployment without getting stuck on environment variables, build configs, or CORS issues.

# Technical Requirements

These are hard requirements. Everything else is your call.

- **Frontend:** Any modern framework or plain HTML/JS. React, Next.js, Svelte, Vue, or even vanilla JavaScript with Tailwind. Your choice.
- **AI Model:** Any LLM API. Claude, GPT-4, Gemini, Llama via Groq, Mistral, or anything else. If you need a free API key for testing, use a provider with a free tier.
- **Content Extraction:** Must work on at least standard WordPress blog posts. Bonus if it handles other CMS platforms gracefully.
- **Deployment:** Live public URL on any platform with a free tier. No localhost submissions.
- **Code:** Public GitHub repo. We will read the code.

**You are explicitly encouraged to use AI coding assistants** (Cursor, GitHub Copilot, Claude, v0, Bolt, Lovable, or any tool you prefer). We don't care how you write the code. We care about what ships.

## Bonus Points (Not Required)

These are not required for a passing submission, but they signal a stronger candidate:

- Tone/audience selector (let the user pick B2B vs B2C, formal vs casual)
- Export functionality (copy all outputs as markdown, download as text file)
- History of previous generations (even just in localStorage)
- Rate limiting and error handling that a real user would encounter
- Responsive design that works on mobile
- Any other feature you think makes the tool genuinely more useful

Don't add bonus features if it means the core functionality suffers. A tool that does four things well beats one that does eight things poorly.

## Evaluation Criteria

CRITERIA	WEIGHT	WHAT WE LOOK FOR
Working Product (Does it work?)	30%	Can we paste a URL and get useful output? Does it handle errors? Is the deploy stable?

AI Output Quality	<b>25%</b>	Do the generated posts sound human? Is the prompt engineering thoughtful? Different variations actually different?
Code Quality and Architecture	<b>20%</b>	Clean structure, readable code, sensible file organization. AI-generated code is fine if you understand it.
UX and Polish	<b>15%</b>	Copy buttons, loading states, error messages, mobile responsiveness. Does it feel like a product?
Speed and Judgment	<b>10%</b>	Did you ship in 3-5 hours or spend 20 hours gold-plating? We value velocity.

## Submission Requirements

5. **GitHub Repository** (public) with a README that includes:
  - Setup instructions (how to run locally)
  - Tech stack choices and brief reasoning (2-3 sentences per choice)
  - What AI tools you used to build it (be honest, this is a positive)
  - What you would improve with another 2 hours
6. **Live Deploy URL** (must be accessible without authentication)
7. **Time Log** (rough breakdown: how long on extraction, AI integration, frontend, deployment)

Send everything to: [amol@upgrowth.in](mailto:amol@upgrowth.in)

Subject line: [Your Name] - AI Growth Engineer Assessment

## What We Don't Care About

- Whether you wrote every line by hand. Use Cursor, Copilot, Claude, v0, whatever. The output matters, not the input method.
- Pixel-perfect design. Clean and usable beats beautiful and broken.
- Test coverage. For a 3-5 hour build, manual testing is fine.
- Which AI model you chose. GPT-4, Claude, Gemini, open-source. Pick what you work fastest with.
- Whether you used a UI component library. Shadcn, MUI, Chakra, hand-rolled Tailwind. All fine.

## What Will Immediately Concern Us

- No live deployment. A GitHub repo with 'run npm start' and no public URL is incomplete.
- AI outputs that are clearly default ChatGPT style with no prompt engineering. If every LinkedIn post starts with 'In today's rapidly evolving landscape,' the prompts need work.
- A README that says 'built with React' and nothing else. We want to see how you think about technical decisions.
- The tool crashes on the first URL we try. Test it with at least 5 different blog URLs before submitting.
- Copy-pasted boilerplate that you clearly don't understand. We will ask you to walk through the code live.
- Over-engineering. If you built a microservices architecture with Docker Compose for a simple content tool, that's a signal you optimized for the wrong thing.

## What This Role Looks Like

If you join upGrowth, your first month looks something like this:

**Week 1-2:** Ship a production-ready internal tool (similar complexity to this assessment but with real business requirements). Get familiar with upGrowth's existing AI apps, the marketing domain, and how we work with clients.

**Week 3-4:** Own a client-facing tool end-to-end. This means talking to the growth team about what's needed, building it, deploying it, and iterating based on usage data. Not just the code, but the product decisions around it.

**Month 2 onwards:** You're building and maintaining multiple tools across SEO auditing, content optimization, AI-powered analysis, and growth automation. Some are internal. Some are client-facing. Some become lead-generation tools on the upGrowth website. You'll also help evaluate and integrate new AI capabilities as the landscape evolves.

The role is high-autonomy. You'll get problems, not specifications. You'll ship weekly, not quarterly. And you'll work alongside AI tools as a core part of your workflow, not as an afterthought.

## Test Your Tool With These URLs

Before submitting, verify your tool works with at least these blog posts:

- <https://upgrowth.in/questions-ask-performance-marketing-agency-before-hiring/>
- <https://upgrowth.in/essential-questions-startup-founders-should-ask-before-hiring-an-seo-agency/>
- <https://upgrowth.in/blogs/> (pick any 2-3 additional posts from this page)

If your tool only works on upGrowth URLs, that's fine for this assessment. If it also handles Medium, Substack, or generic WordPress blogs, that's a bonus.

## Reference Material

These give you context about what upGrowth builds:

- **AI-Powered Apps:** <https://upgrowth.in/ai-powered-apps/>
- **upGrowth Blog:** <https://upgrowth.in/blogs/>
- **Case Studies:** <https://upgrowth.in/case-study/>

## Timeline and Communication

You have **5 calendar days** from receiving this brief.

We chose 5 days instead of 7 because this should not take a full weekend. If you're spending more than 5 hours, you're likely over-engineering. The constraint is intentional. We want to see what you prioritize when time is limited.

If you need a day extra, email us before the deadline with a brief note on where you are. We'd rather see a working product one day late than a broken one on time.

## A Note on Compensation

This assessment is unpaid. Here's why, and what you get in return:

The output is yours. Unlike production code built for a specific business requirement, this tool is a standalone portfolio piece. You own it completely. Put it on your GitHub, add it to your portfolio, deploy it on your own domain. It demonstrates AI integration skills that are increasingly valuable in the market.

If you pass the assessment and join upGrowth, your first paid project will be similar in nature but with direct business value and a clear product specification.

## Questions?

Email [amol@upgrowth.in](mailto:amol@upgrowth.in) for scope or logistics clarifications.

We will not answer questions about which framework to use, which AI model is best, or how to structure the code. Those decisions are the test.

### **Amol Ghemud**

Chief Growth Officer, upGrowth Digital LLP  
[upgrowth.in](http://upgrowth.in)