

Ideas and questions from last night:

- Can train ML model to keep score based on appeasibility of resume
 - Resume AI screeners can grade on appeasibility as well as just key words before human grader sees
 - For this, would need to get data, can ask someone if anyone keeps track of data and in the long run can get professionals from whose data set I build it off of
- Should this be part of a Hackathon or a startup instead? Idea given by Adobe Express AI add-ons, are these sold to the company? Victor has a resume builder as a startup
 - Should my project be a startup instead? Can I submit in hackathon and still retain rights to my project?
- Can I add a suggestions feature as well? Not all resumes have same experiences and I can have algorithm that ranks importance of persons experiences, like suggesting them to put previous work experience over high school rewards.

Tasks Done Today:

- Quickly recapped ideas:
 - Stand-alone vs add-on/feature in a larger app, (Thinking which way to go)
 - User writes resume contents and uploads sample resume for a template from internet. My program makes a resume with their contents in that template style
 - Challenges: mapping for different templates, having to let edit before finalize pdf, be able to do for free
 - Probably need both front and back end
 - Need to be able to create an MVP for hackathon
- Free chatgpt suggestions:
 - Tesseract OCR (free, open-source)
 - Firebase/Supabase (free tiers for hosting + file storage)
 - pdf-lib or html2pdf (free JS libs for generating PDFs)
 - Vercel/Render for backend/frontend hosting (free tiers)
- Talked to and created looking for teammate post
 - Scrolled through help posts for ideas
- Chat-gpt wrapper vs NLP & AI resume builder
 - Settled plan: Rough work estimate (free + open-source):
 - Template parsing & placeholders: 15-25 hours
 - Content parsing & NLP: 10-20 hours
 - Content mapping & rendering: 15-25 hours
 - UI development: 10-15 hours
 - Testing & debugging: 10-20 hours
 - **Total: ~60-100 hours** depending on scope.

⚙️ Core MVP Tasks & Time Estimates

Task	Description	Est. Time
1. UI for Upload + Text Input	Simple frontend with file upload (PDF/image) and textarea for raw content	4–6 hrs
2. OCR + Layout Extraction	Use an OCR tool (e.g., Tesseract, pdfplumber) to extract layout and text	6–10 hrs
3. Parse & Represent Template Structure	Identify and extract section headers, font sizes, positions (basic layout logic)	6–8 hrs
4. Format Raw Text to Match Template	Prompt GPT to structure user text based on layout extracted	6–10 hrs
5. Resume Rendering	Convert structured content to HTML/CSS (template-like)	4–6 hrs
6. Export to PDF	Use something like <code>html2pdf</code> , <code>puppeteer</code> , or Kiro to export	3–5 hrs
7. Backend API (if needed)	Basic Flask/FastAPI backend to connect OCR, GPT, and export steps	6–10 hrs
8. Integration & Testing	Plug everything together, fix bugs, test edge cases	8–12 hrs
9. Polish & UX Improvements (optional)	Loading indicators, error handling, minor styling	4–6 hrs

✅ Step-by-step: Free or Not?

Step	Tool/Service	Free?	Notes
User uploads template	Kiro file upload	✅ Free	Basic file handling is free
OCR & layout extraction	Kiro's Vision API (or external OCR like Tesseract.js or Google Vision)	💎 <i>Maybe</i>	If you're using a custom Kiro block or integrating third-party OCR, costs depend on the service. Tesseract is free. Google Vision API has a generous free tier (1K units/month).
User inputs raw content	Kiro form block	✅ Free	No charge for form inputs
GPT formatting step	Kiro + OpenAI GPT (via OpenAI block)	❌ Not free (if GPT-4) / ✅ Low-cost (GPT-3.5)	GPT-3.5 is inexpensive, but GPT-4 usage costs apply. You'll need your OpenAI API key or pay via Kiro's billing.
Export to PDF/HTML	Kiro export block	✅ Free	Kiro can export HTML & PDF for free

Next goals: Determine costs, Partnerships, what technologies to use, general roadmaps, etc

Suggested Initial Tech Stack

Layer	Tech Stack	Cost
Frontend	Next.js + Tailwind CSS	Free (Vercel deploy optional)
Backend	Python + FastAPI	Free
OCR/Layout	<code>layoutparser</code> + <code>pytesseract</code>	Free, local Tesseract OCR
GPT	OpenAI API (GPT-4 or GPT-3.5)	Free trial or usage-based
Export	Kiro (optional)	Free plan available
Storage	Local or use temporary S3/Firestore	Free tiers exist

for MVP