



Coding Assignment

Senior Front-End Engineer

Thank you for your interest in Modelia!

This assignment is designed to evaluate your front-end engineering skills, code quality, and how you use modern tools (including AI) to develop, test, and collaborate.



SUBMISSION REQUIRED - READ FIRST



⚠️ CRITICAL: You **MUST** send ALL requested deliverables to frontend@modelia.ai to be considered for this position. Incomplete submissions will be automatically disqualified - no exceptions! ⚠️



Required Email Contents:



GitHub repo link



Links to at least 2 Pull Requests



Optional: Screen-recorded demo



All files mentioned in deliverables section

No email = No consideration. Make sure you hit SEND! 📧

Now go to the next page and read the assignment. Good luck!!

Goal

Build a small React web app that simulates a simplified version of an AI studio.

Requirements

1. Upload & Preview

- Upload a PNG/JPG ($\leq 10\text{MB}$). Show a preview.
- If the file is too large, downscale it client-side to $\leq 1920\text{px}$ before sending.

2. Prompt & Style

- Input field for a text prompt.
- Dropdown with at least 3 style options (e.g. Editorial, Streetwear, Vintage).
- Show a live summary (image + prompt + style).

3. Generate (Mock API)

- Button to “Generate” that calls a mocked API you implement:
 - POST body: { imageDataUrl, prompt, style }
 - Response: { id, imageUrl, prompt, style, createdAt } after $\sim 1\text{--}2\text{s}$
 - In 20% of cases, return a simulated error { message: "Model overloaded" }
- Show a loading spinner.
- On error: retry automatically with **exponential backoff** (max 3 attempts).
- Allow user to **Abort** an in-flight request.

4. History

- Save the last 5 generations in localStorage (id, imageUrl, prompt, style, createdAt).
- Clicking a history item restores it in the main preview.

5. Accessibility

- The app should be keyboard navigable with visible focus states and ARIA where needed.

Tech Stack

- **React + TypeScript** (Next.js or Vite).
- **TailwindCSS** for styling.
- TypeScript strict mode enabled.
- ESLint + Prettier.

Bonus (Optional, but highly valued)

- Unit tests (React Testing Library).
- End-to-end test (Playwright or Cypress).
- Git branching + Pull Requests with clear commit messages.
- Performance optimizations (memoization, code splitting, client image downscale).
- PWA basics (manifest + offline caching).
- Error boundaries and empty states.

Deliverables

- Public GitHub repo with:
 - Source code.
 - README.md with install/run/test instructions + design notes.
 - **AI_USAGE.md** → short notes on how you used AI (Copilot, ChatGPT, Cursor, etc.) to code, test, debug, or automate PRs.
- At least 2 Pull Requests with meaningful descriptions.
- Optional: small evidence/ folder (screenshots of AI usage, CI runs, etc.).

Timebox

The task should take around **6–8 hours**. Please keep scope reasonable; leaving clear TODOs is better than rushing.



Submission

1. Share your GitHub repo link.
2. Share links to at least 2 Pull Requests (merged or open).
3. (Optional) Provide a short screen-recorded demo.

Send to frontend@modelia.ai



Evaluation

We will score your submission across:

- Functional correctness
- Code structure and TypeScript strictness
- UI/UX quality and accessibility
- Testing depth
- Git workflow and documentation
- Clear evidence of AI usage

👉 That's it. Good luck!

And don't hesitate to showcase how you **use AI to accelerate your workflow!!**