

Full Stack Assignment – AI Coding Agent (Claude-style)

Objective

Build a Claude-like AI Coding Agent UI using the Gemini 2.5 API (free tier) with streaming response, that lets users chat with an agent and view generated code artifacts in a toggled sidebar view.

Deliverables

- A web app with a Claude-style layout:
 - Chat on the left
 - Dynamic right-hand sidebar that opens when code is generated
 - A button group to toggle between `Code` and `Preview (Sandbox)` views
 - A button labeled "View Generated Artifact" should appear once a code artifact is generated — this opens/closes the sidebar.
 - The code artifacts should (optional but bonus):
 - Be shown in a syntax-highlighted editor
 - Support minimal preview (render simple HTML/CSS/JS if applicable)
 - AI responses must:
 - Be streamed from Gemini 2.5 API (free tier)
 - Have short-term memory, using in-memory store (e.g. LangGraph memory or local memory saver)
-

Tech Stack

You are free to choose your stack, but we suggest:

- Frontend:
 - React
 - TailwindCSS (or any Tailwind-based UI library like ShadCN, Radix UI)
- Backend:
 - FastAPI (or any backend framework/language you're comfortable with)
- LLM API:
 - Gemini 2.5 (free tier)
- Optional Agent Orchestration:
 - LangGraph (for managing flows/memory)
- State Management:
 - Zustand / Redux / Context API
 - React Query

- AI Tools:
 - Cursor / Copilot / ChatGPT / Claude, feel free to use anything
-






Bonus

- Use an editor like `Monaco` for code preview
 - Use a sandbox like `e2b.dev` for runtime preview of the generated code
 - Implement file download for generated code
 - Allow resending a message or editing past prompts
 - Mobile responsiveness
-

Submission Guidelines

- Upload your code to a GitHub repo (make it public or share access)
 - Include a README with:
 - Setup instructions
 - How you approached the problem
 - Link to a deployed version (e.g. Vercel, Render, etc.)
-

Evaluation Criteria

-  Product Thinking (UX/UI polish and Claude-like flow)
-  Engineering Depth (API integration, streaming, component structure)
-  AI Integration Skill (prompting, memory handling)
-  Thoughtfulness in architecture
-  Code quality, modularity, and documentation