

Fable vs Opus: The 4 Test Prompts

The exact single prompts I gave Claude in the video. Run each one in Claude Code, switch the model between Fable 5 and Opus, and compare the builds yourself. Tip: send the whole prompt in one message — no follow-ups — so the head-to-head stays fair.

01

Test 1 — MemoFlow (Mobile Quiz App)

Watch what each model adds unprompted. Opus builds a solid app; Fable tends to go a few layers deeper — extra features and edge cases you never asked for.

Build MemoFlow, a mobile-first flashcard and quiz web app inspired by Quizlet and Anki. It should feel polished, playful, and premium, optimized for a 390px mobile viewport while staying fully responsive.

Core features: home screen of study decks, flashcard mode with animated card flipping, quiz mode with multiple-choice questions, a Create Study Set screen, bottom navigation, and local persistence via LocalStorage.

Flashcards: flip, navigate forward/back, shuffle, track progress, resume the previous session.

Quiz: randomized multiple-choice, progress indicator, instant feedback, final score, restart.

Prepopulate three sets: Basic Math, World Capitals, Human Biology. Users can create unlimited sets.

Design: premium iOS-inspired language with smooth animations and delightful interactions.

Tech: single HTML file, vanilla HTML/CSS/JS, Google Fonts allowed, no frameworks, runs on localhost.

02

Test 2 — MeetFlow (Calendly-Style Booking App)

This one exposes design taste. Compare the color choices — Fable avoids the generic “purple AI template” look that most models default to.

Build MeetFlow, a modern scheduling app inspired by Calendly and Cal.com.

Booking flow: select date, select time, enter name/email, confirmation screen.

Dashboard: view bookings, search, cancel, reschedule, manage availability. Generate realistic demo bookings.

Availability: working days, working hours, blocked dates, meeting duration. Persist with LocalStorage.

Design: premium SaaS interface with clean UX.

Tech: single HTML file, vanilla HTML/CSS/JS, no frameworks.

03

Test 3 — EchoWrite AI (Landing Page)

Look at how many sections each model includes and how the animations feel. Fable made editorial calls about what a high-converting page actually needs.

Build a premium landing page for EchoWrite AI, an AI voice dictation platform.

Sections: Navbar, Hero, Features, How It Works, Screenshots, Testimonials, Pricing, FAQ, CTA, Footer.

Pricing: Free and Pro tiers.

Design: inspired by Linear, Notion, Grammarly, and ElevenLabs.

Tech: single HTML file, vanilla HTML/CSS/JS, Google Fonts allowed.

04

Test 4 — MetroVerse (SimCity-Style Game)

The real stress test. Check whether the city feels alive — moving cars, traffic lights, citizens, day/night — versus static boxes with labels.

Build MetroVerse, a browser-based isometric city simulation game inspired by SimCity and Cities: Skylines.

World generation: roads, buildings, parks, trees, bridges, districts, landmarks.

Citizens: autonomous named citizens walking through the city.

Vehicles: autonomous traffic obeying traffic lights.

Simulation: day/night cycle, dynamic lighting, ambient animation — a living city.

HUD: population, vehicles, time, budget, happiness.

Controls: pause, play, speed, zoom, camera pan.

Tech: single HTML file, vanilla HTML/CSS/JS, no external libraries. Prioritize smooth performance and a believable simulation.

Ready to go deeper?

These prompts are just the start. In Vibe Coding Mastery you get the full system for building and shipping real apps with Claude Code — no dev team needed.

[Join the Vibe Coding Build →](https://builds.digicuratoragency.com/)

<https://builds.digicuratoragency.com/>