SOLUTION STREET STREE

(Creative & Motion Direction per Fylle Onboarding Chat Experience)

Visione Creativa

Creare un'esperienza visiva che **trasmetta la sensazione di un laboratorio Al vivo**, dove agenti collaborano per generare intelligenza collettiva. L'utente deve percepire:

"Sto parlando con un sistema intelligente, ma progettato con cura umana." L'esperienza non deve sembrare un form, né un'interfaccia marketing — deve sembrare una **collaborazione in tempo reale con il team Fylle**.

Elementi Chiave

Elemento	Direzione Creativa	Note
Mood generale	Tech minimal + organic intelligence	Toni puliti, neutri, luce diffusa, effetto "white grid lab".
Colore Primario	Verde Fylle #00F500	Glow morbido, mai saturazione piena. Usalo per accent, pulsazioni e CTA.
Colore secondario	Nero profondo #181A0F	Background principale o contrasto tipografico.
Accenti	Grigio neutro #5F6A77, bianco #F5F5F5	Per testi e sfondi neutri.
Font	Sora (titoli), Inter (testo)	Già coerenti con il brand site. Nessun nuovo font.
Background pattern	Griglia vettoriale sottile #F5F5F5 su bianco o su scuro trasparente.	Richiama il sistema "grid" del sito.
Iconografia	Flat line / vector Al	Stile "system graph" – linee di connessione tra agenti.
СТА	Glow morbido, effetto gradient verde/bianco.	Hover con shadow diffuso.
Micro-interazioni	Pulsazioni agent, linee animate, typing dots, crossfade di stato.	Nessun effetto "gaming". Movimento fluido e elegante.

Spacing	Aria, focus,	max-width 720px,
	modularità.	padding 2rem, spazio
		tra messaggi 1.25rem.

Motion & Composizione

1. Chat Flow Animation

- Ogni messaggio Al appare con un leggero slide-up + fade (0.5s).
- Delay tra messaggi: 600–800 ms (randomizzato per realismo).
- Typing indicator (3 dots) per 0.8–1.2 s prima di ogni messaggio Al.

2. Agent Activity Mock

- 3 card parallele (Perplexity, Gemini, CGS).
- Animazione: glow pulsante verde → attivo / inattivo.
- Linee curve (SVG path animate) tra le card per dare senso di connessione.
- Micro-transizioni di testo tipo "terminal simulation" (effetto digit 30– 40 ms per carattere).

3. Phase Transition

- Passaggio tra stati ("researching" → "quiz" → "executing" → "result")
 con fade-to-white + smooth scroll.
- Tempo massimo transizione: 0.8s.

4. Final Redirect

- Al completamento, "fade to black" + transizione soft del logo Fylle + redirect a URL configurato.
- Logo centrale con glow verde: "Redirecting you to your workspace..."

Nisual Composition Layout

Dark Mode by default

- Glow verde per interazione (focus, click, typing)
- Transizioni verticali → il flusso scorre sempre verso il basso
- Uso di blur (10–20 px) per effetti di profondità

Linguaggio visivo del testo

Elemento Stile linguistico Esempio	
------------------------------------	--

Al Message	Confidente + empatico	"Mapping your brand ecosystem"
Agent Mock	Tecnico-descrittivo	"Gemini: Synthesizing tone and structure."
CTA finale	Motivante + umano	"Let's see what your Al Team created."

Emozione finale da trasmettere

"Sto collaborando con un team di intelligenze che lavorano per me, con stile e precisione."

FRONT-END TECH BRIEF

(Implementation & Integration with Backend)



Frontend (Next.js 14 + React 18)

↓ REST calls

Onboarding API (FastAPI:8001)

CGS Core Backend (:8000)

Brevo Email Delivery + Redirect URL

X Stack Tecnico

Area	Tecnologia	Note
Framework	Next.js 14 (App Router)	Per SSR + SEO ready
Lang	TypeScript	Interfacce già definite (OnboardingSession, etc.)
Styling	Tailwind CSS + Framer Motion	Motion design coerente
State	Zustand o useReducer	Polling + session tracking
НТТР	fetch API + SWR	Gestione cache e polling semplice
Markdown Render	react-markdown + rehype-highlight	Per content preview finale
Env vars	.env.local → NEXT_PUBLIC_ONBO ARDING_API_URL, NEXT_PUBLIC_SUCCE SS_URL	

Mail	Gestita da backend via	Il frontend mostra
	Brevo	conferma e redirect

Integrazione API Backend

Endpoints usati

API	Metodo	Descrizione
/start	POST	Avvia onboarding (brand_name, website, goal, email)
/{session_id}	GET	Recupera stato sessione (poll ogni 3 s / 5 s)
/{session_id}/answers	POST	Invia risposte quiz
/{session_id}/execute	POST	Avvia generazione contenuto
/health	GET	Debug, health check

Gestione stati frontend

const asyncStates = ['researching', 'synthesizing', 'executing', 'delivering'];
const finalStates = ['awaiting_user', 'done', 'failed'];

- Polling dinamico (3 s async / 5 s executing)
- Stop polling quando finalStates.includes(state)

Session Management

- session id salvato in localStorage
- Se la sessione è attiva → ripristina stato chat (per reload/resume)
- useOnboarding() gestisce polling e parsing response JSON
- Error handling centralizzato (try/catch + toast error)

Chat Flow Engine Hook useChatFlow()

- Controlla:
 - Sequenza messaggi Al
 - Delay e typing indicator
 - Trigger automatico API (/start, /answers, /execute)
- Gestisce stati interni:

form \rightarrow research \rightarrow quiz \rightarrow generate \rightarrow result \rightarrow redirect

Hook useAgentMock(state)

- Attiva messaggi animati in base allo stato corrente (mock dinamico)
- Cleanup su unmount

Rendering dinamico dei messaggi

Tipo	Component	Azione
1.100		7 (210110

system	SystemMessage	Mostra status (es. "Analyzing")
ai	ChatMessage	Mostra messaggi con typing
user_input	ChatInput	Input dinamico (string, enum, boolean)
agent	AgentActivity	Mostra card agent con log simulati

Redirect finale

- Backend invia mail via Brevo → state: done
- Frontend mostra:
 - " Content sent to your inbox via Brevo."
- Dopo 4 s → redirect automatico a:

```
window.location.href = `$
{process.env.NEXT_PUBLIC_SUCCESS_URL}?session_id=$
{session.session_id}`;
```

Folder Structure

```
/src
 /components
  ChatContainer.tsx
  ChatMessage.tsx
  ChatInput.tsx
  AgentActivity.tsx
  ResultPreview.tsx
  MetricsCard.tsx
 /hooks
  useOnboarding.ts
  useChatFlow.ts
  useAgentMock.ts
 /utils
  agentMessages.ts
 /api
  onboarding.ts
```

Security & Performance

▼ CORS già configurato su backend per sviluppo

- In produzione → allow_origins solo su fylle.ai, app.fylle.ai
- 💾 Lazy load markdown + images
- A HTTPS enforced (Vercel default)

QA Checklist

Area	Test	Metodo
Polling	stop/start test	Mock API delay
Quiz Validation	enum/boolean errors	form test
Email Delivery	confirm Brevo state=delivered	manual test
Redirect	URL dynamic test	local env var
Responsiveness	mobile/desktop	Chrome dev tools
Accessibility	keyboard nav + contrast	Lighthouse

Deliverable Finali

- 1. Interfaccia conversazionale live (Next.js)
- 2. Animazioni Framer Motion integrate
- 3. Agent Mock visivo sincronizzato agli stati
- 4. Email confirmation (mock Brevo)
- 5. Redirect finale con session tracking
- 6. Struttura pulita per futura API streaming (SSE/WebSocket)