Product Requirements Document: Feno – The Infinite Canvas Career Operating System

1. Executive Summary

1.1. Strategic Product Vision

The contemporary employment landscape is characterized by fragmentation. Job seekers are forced to navigate a disjointed ecosystem of static document editors, isolated job boards, manual application forms, and disconnected personal branding tools. **Feno** aims to unify this fragmented workflow into a single, cohesive Career Operating System (CareerOS). The fundamental innovation of Feno lies in its departure from the traditional, linear "A4 document" metaphor for resume creation. Instead, Feno introduces an **Infinite Canvas** interface—a spatial, non-linear workspace powered by generative AI—that allows users to ideate, draft, visualize, and deploy their professional narratives across multiple mediums simultaneously.

The vision for Feno is to transition the user's mental model from "writing a resume" to "managing a career database." In this new paradigm, the user's professional history is treated as a structured dataset (Single Source of Truth), which the platform dynamically renders into various artifacts: a PDF resume optimized for Applicant Tracking Systems (ATS), a responsive personal portfolio website, or a structured data payload for automated job applications. Feno is not merely a builder; it is an autonomous agent that manages the labor-intensive logistics of the job hunt, allowing the candidate to focus on skill acquisition and interview preparation.

1.2. Value Proposition and Market Differentiation

The current market for career tools is saturated with "form-fillers" (e.g., Resume.io, Zety) and "trackers" (e.g., Teal, Huntr). Feno differentiates itself through three core pillars of innovation:

1. **Spatial Cognition over Linear Editing:** By utilizing a node-based interface (via React Flow), Feno aligns with the non-linear way creative and technical professionals think. Users can map skills to job descriptions visually, creating a tangible connection between their experience and market requirements.[1]
2. **One-Click Transformation:** The platform eliminates the "Cold Start Problem" of portfolio creation. By leveraging Large Language Models (LLMs) to transform structured resume JSON into semantic HTML, Feno creates high-quality personal websites instantly, deploying them to live URLs without a single line of code from the user.[1]
3. **Autonomous Execution:** Moving beyond passive "tracking," Feno's browser extension actively participates in the application process, auto-filling complex ATS forms (Workday, Lever, Greenhouse) using the user's verified data, significantly reducing the "time-to-apply" metric.[1]

1.3. Document Scope

This Product Requirements Document (PRD) provides a comprehensive technical and functional specification for Feno. It covers the end-to-end user journey, the architectural decisions required to integrate rich text editors within canvas libraries, the legal and technical strategies for automated applying, and the roadmap for monetization. This document is intended for stakeholders in Product Management, Engineering, and Design to ensure alignment on the complex interdependencies of the platform.

---

2. Market Analysis and Competitive Landscape

2.1. The State of CareerTech in 2025

The CareerTech sector has shifted from simple document formatting to AI-driven optimization. The ubiquity of ATS software means that 75% of resumes are never seen by

human eyes, necessitating a "machine-first" approach to resume writing.[4] Furthermore, the rise of the "creator economy" has pressured professionals to maintain personal websites, yet existing website builders (Wix, Squarespace) remain too generic and time-consuming for the average job seeker.

2.2. Detailed Competitor Analysis

Competitor	Core Offering	Pricing Model	Strengths	Feno's Strategic Advantage
Teal	Job Tracking & Keyword Analysis	Freemium ($29/mo)	Excellent job tracker and keyword scanner; strong LinkedIn integration.[5]	Teal is list-heavy and form-based. Feno's **Spatial UI** offers superior organization for complex career histories, and Feno *applies* for the user, whereas Teal only *tracks*.
Rezi	ATS Optimization	Freemium ($29/mo)	"Rezi Score" is the industry standard for ATS compliance; content-first approach.[4]	Rezi offers no portfolio capabilities. Feno matches the ATS scoring but adds the **Portfolio Builder**, providing two products for

				the price of one.
Kickresume	Visual Templates	Freemium ($19/mo)	High-quality design templates; basic website builder.[8]	Kickresume's website builder is template-bound. Feno's **Generative UI** allows users to prompt code changes ("Move this section here"), offering greater flexibility.
LazyApply	Automation Bot	One-time ($129+)	Mass application submission (spam-like approach).[3]	LazyApply is high-risk and low-quality ("spray and pray"). Feno focuses on **Tailored Automation**, creating a unique resume for each auto-application to ensure quality over quantity.[11]
Gamma	AI Presentations/ Sites	Freemium	incredible generative UI for decks and simple sites.[12]	Gamma is not resume-specific. Feno adapts Gamma's "text-to-site" fluidity specifically for

				career artifacts, integrating deep ATS logic Gamma lacks.

2.3. Target Audience Segmentation

- **The "Volume Applicant" (Junior/Mid-Level):** Facing a numbers game. Needs to send 50+ tailored applications a week. Value driver: **Speed & Automation**.
- **The "Visual Thinker" (Designers/Product Managers):** Feels constrained by A4. Wants to visualize their career trajectory. Value driver: **Spatial Interface & Portfolio**.
- **The "Optimizer" (Engineers/Data Scientists):** Obsessed with metrics. Wants to see their "ATS Score" and optimize keywords like SEO. Value driver: **Analytics & Data**.

---

3. Product Pillars and Strategic Objectives

3.1. Pillar I: The Infinite Workspace

The objective is to break the skeuomorphic bond to the physical sheet of paper during the *drafting* phase. While the final output must be a PDF, the creation process should be unconstrained.

- **Metric:** Increase average session duration by 40% compared to form-based editors (Benchmark: >12 mins).
- **Mechanism:** React Flow-based canvas allowing users to arrange resume sections, job descriptions, and AI notes spatially.[1]

3.2. Pillar II: Generative Synergy

The objective is to leverage AI not just for text generation, but for structural transformation.

- **Metric:** Achieve a 50% adoption rate of the Portfolio feature among Resume Builder users.
- **Mechanism:** An LLM pipeline that acts as a "Compiler," taking Resume JSON as input and outputting Portfolio React Code.[13]

3.3. Pillar III: Autonomous Agency

The objective is to shift the burden of logistics from human to machine.

- **Metric:** Users save an average of 5 hours per week on applications.
- **Mechanism:** A localized browser extension that bridges the gap between the Feno database and third-party job board DOMs.[14]

---

4. Detailed Functional Requirements

4.1. Module A: The Infinite Canvas Resume Builder

4.1.1. The Spatial Interface (React Flow)

The core interface is an infinite, pannable 2D plane constructed using **React Flow**. This library provides the necessary primitives (Nodes, Handles, Edges, Controls) to build a node-based editor.[15]

- **Canvas Behavior:**
  - **Infinite Panning:** Users can pan in any direction to create space for new ideas or "parking lots" for text snippets.
  - **Semantic Zoom:** Zooming out reveals high-level structure (sections); zooming in reveals edit controls (Tiptap toolbar).
  - **MiniMap:** A navigational aid in the bottom-right corner to orient the user within large

workspaces.[17]

- **Node Architecture:**
  - **The Document Node:** The central node representing the resume. Unlike standard flowcharts, this node is massive and contains a fully functional rich text editor.
  - **The Context Node:** Users can paste a Job Description (JD) into a secondary node. This node serves as the "Ground Truth" for AI analysis.
  - **The Insight Node:** Floating nodes generated by the system that contain AI suggestions (e.g., "Missing Skill: Kubernetes"). These nodes connect visually via **Edges** to the specific section of the Document Node they refer to.[1]

4.1.2. The Embedded Editor (Tiptap)

Inside the Document Node, Feno must implement a robust WYSIWYG editor. **Tiptap v3** (based on ProseMirror) is the mandated technology due to its headless nature and ability to handle complex schema validation.[1]

- **Text Editing Capabilities:**
  - Standard formatting (Bold, Italic, Lists, Headers).
  - **Slash Commands:** Typing / triggers a menu to insert tailored blocks (e.g., "Experience Block", "Skill Bar").
  - **Ghost Text:** AI-powered autocomplete (similar to GitHub Copilot) that suggests bullet points based on the user's role title.[19]
- **Drag & Drop Compatibility (Critical):**
  - *Requirement:* Selecting text inside the Tiptap editor must **not** trigger the drag movement of the React Flow node.
  - *Implementation:* The Tiptap container must be wrapped in a div with the class .nodrag (a utility class provided by React Flow) to stop event propagation to the canvas.[21]
  - *Node Movement:* A specific "Drag Handle" (grip strip) must be placed at the top of the Document Node to allow intentional repositioning.[23]

4.1.3. AI-Powered ATS Intelligence

- **Real-Time Scoring:** As the user types or edits, the system runs a background analysis comparing the Document Node content against the Context Node (Job Description).
- **Visual Feedback:**
  - **Edges as Data Connectors:** If the JD mentions "Python" and the Resume mentions

"Python," a green edge (connection line) temporarily draws between them to visualize the match.
    - **Gap Analysis:** If "Docker" is in the JD but missing from the Resume, a Red Warning Node appears, linked to the "Skills" section.[1]
- **One-Click Optimization:** The Warning Node contains a "Fix with AI" button. Clicking it triggers an LLM prompt to generate a justified bullet point incorporating the missing skill, which the user can accept or reject.

4.2. Module B: The Generative Portfolio Builder

4.2.1. Resume-to-Website Transformation

This feature allows users to "export" their resume as a website.

- **Data Pipeline:**
    1. **Extraction:** The system extracts the JSON content from the Tiptap/ProseMirror document.
    2. **Schema Mapping:** This JSON is mapped to a "Portfolio Schema" (JSON Resume standard or proprietary).[25]
    3. **Theme Selection:** The user selects a visual theme (e.g., "Minimalist," "Creative," "Corporate").
    4. **Generation:** An LLM (GPT-4o) generates the website copy and structure based on the resume data and selected theme. It expands brief resume bullets into engaging case study prose for the website.[1]

4.2.2. Chat-Based Customization

- **Natural Language Editing:** Users interact with the portfolio via a chat interface.
    - *User Prompt:* "Change the hero section to be dark mode and emphasize my work at Google."
    - *System Action:* The LLM interprets this, modifies the underlying JSON configuration of the portfolio, and triggers a re-render.[26]
- **Section-Specific Regeneration:** Users can click a "Regenerate" button on specific components (e.g., the "About Me" section) to get alternative copy or layout variations.

4.2.3. Deployment Infrastructure

- **One-Click Deploy:** Utilizing Vercel's Platform API, Feno creates a live deployment instantly.
- **Subdomain Management:** Every user gets a username.feno.site subdomain automatically.
- **Custom Domains:** Premium users can connect www.johndoe.com via CNAME configuration, managed through the Vercel API.[27]

4.3. Module C: The "Auto-Apply" System

4.3.1. Browser Extension Architecture

To avoid the legal and technical pitfalls of server-side scraping (IP bans, CAPTCHAs), the auto-apply functionality is architected as a **Chrome Extension**.

- **Mechanism:**
  1. **Detection:** The extension detects when the user is on a supported job board (LinkedIn, Indeed, Lever, Greenhouse).
  2. **Mapping:** It identifies input fields (First Name, Last Name, Resume Upload, "Why do you want this job?").
  3. **Injection:** It retrieves the user's data from the Feno local storage or API and injects it into the fields.[29]
  4. **Tailoring:** For open-ended questions ("Cover Letter"), the extension sends the Job Description from the current page to Feno's AI, generates a specific response, and fills the text area.

4.3.2. "Human-in-the-Loop" Safety

- **Review Mode:** The extension fills the data but *pauses* before hitting submit, allowing the user to verify the information. This "Assisted Apply" mode reduces the risk of hallucinated

data being sent to employers.[11]

4.4. Module D: Analytics & Insights

4.4.1. Resume Telemetry

- **Link Tracking:** When users share a link to their resume (instead of a PDF), Feno tracks unique views, time spent, and download events.
- **Heatmap Proxy:** By tracking scroll depth on the web-view of the resume, Feno provides a proxy for which sections recruiters are reading most.[1]

4.4.2. Portfolio Analytics

- **Implementation:** Integration of privacy-friendly, open-source analytics (e.g., **Umami** or **Plausible**) self-hosted by Feno. This avoids the need for cookie banners while providing essential data like page views and referrers.[31]
- **Visualization:** Data is visualized using **Recharts** or **Visx** within the Feno dashboard.[33]

---

5. Technical Architecture & Implementation Details

5.1. Frontend Architecture (React + Next.js)

The application will be built on **Next.js 14+** (App Router) to leverage React Server Components (RSC) for performance and SEO.

5.1.1. State Management: The Synchronization Challenge

The application has three distinct state layers that must stay synchronized:

1. **Canvas State (React Flow):** nodes, edges, viewport. Managed via **Zustand** (recommended by React Flow creators).[15]
2. **Editor State (Tiptap):** doc (ProseMirror JSON), selection. Managed internally by Tiptap but must sync to the global store for AI access.[34]
3. **Application State:** User auth, credits, active theme.

- **Solution:** A centralized Zustand store will act as the "Single Source of Truth." The Tiptap editor's onUpdate callback will debounce and persist changes to the Zustand store. This ensures that if the user switches to the Portfolio view, the data is immediately available without parsing the DOM.

5.1.2. Custom Node Implementation

The ResumeNode is a custom React Component registered with React Flow.

JavaScript

```
// Conceptual implementation of the Resume Node
import { Handle, Position } from '@xyflow/react';
import { EditorContent } from '@tiptap/react';

const ResumeNode = ({ data }) => {
  return (
    <div className="resume-node-container shadow-xl bg-white">
      {/* Drag Handle - ONLY this area triggers drag */}
      <div className="drag-handle h-6 bg-gray-200 cursor-grab" />

      {/* Editor Area - nodrag prevents canvas movement during text selection */}
      <div className="editor-wrapper nodrag cursor-text p-8">
        <EditorContent editor={data.editorInstance} />
      </div>

      {/* Dynamic Handles for AI connections */}
      <Handle type="target" position={Position.Left} id="a" />
```

```
    <Handle type="source" position={Position.Right} id="b" />
  </div>
 );
};
```

- **Key Technical Detail:** The .nodrag class is essential. Without it, selecting text in Tiptap triggers the React Flow pan gesture, creating a broken UX.[21]

5.2. Backend Architecture (Node.js + Python)

5.2.1. Hybrid Microservices

- **Core API (Node.js/Next.js):** Handles auth (Auth.js/Clerk), CRUD operations, and payment processing (Stripe).
- **AI Worker (Python/FastAPI):** Python is preferred for the AI/ML workload due to superior libraries for parsing and vector operations.
  - **Resume Parsing:** Using **PyResparser** or custom **Spacy** models to extract entities (Skills, Companies) from uploaded PDFs.[35]
  - **Scoring Logic:** Uses **scikit-learn** to calculate cosine similarity between the Resume Vector and JD Vector.[37]

5.2.2. Database Schema

- **Primary DB: PostgreSQL**.
  - Table resumes: Stores the ProseMirror JSON structure.
  - Table portfolios: Stores the site configuration JSON.
- **Vector DB: pgvector** (PostgreSQL extension) or **Pinecone**.
  - Stores embeddings of the user's resume sections. This allows the AI to perform RAG (Retrieval Augmented Generation). When the user asks "Write a cover letter," the system retrieves only the relevant experience blocks from the vector store instead of passing the whole resume token limit.[38]

5.3. The AI Engine: Structured Outputs

Generating reliable JSON from LLMs is notoriously difficult (hallucinations, syntax errors). Feno requires strict schema adherence for the Portfolio Builder to function.

- **Strategy:** Use **OpenAI's Structured Outputs** (response_format: { type: "json_object" }) combined with **Zod** schema validation on the client side.[26]
- **Schema Definition:**

```
TypeScript
// Zod Schema for Portfolio Generation
const PortfolioSchema = z.object({
  hero: z.object({
    headline: z.string(),
    subheadline: z.string(),
  }),
  projects: z.array(z.object({
    title: z.string(),
    description: z.string(),
    technologies: z.array(z.string())
  })),
  //...
});
```

  If the LLM output fails PortfolioSchema.parse(), the system automatically triggers a retry loop, feeding the validation error back to the LLM to self-correct.[40]

5.4. Portfolio Deployment (Vercel API)

- **Mechanism:** Feno acts as a Vercel Platform Partner.
- **Workflow:**
  1. User clicks "Publish."
  2. Feno backend calls POST https://api.vercel.com/v9/projects to create a project (if new).
  3. Feno calls POST https://api.vercel.com/v6/deployments passing the Portfolio JSON as an Environment Variable or a JSON file artifact.[41]
  4. The **Portfolio Renderer** (a generic Next.js app) is deployed. It reads the JSON config at runtime (SSR) to render the unique site. This ensures the site is SEO-friendly (server-rendered) rather than a client-side shell.[42]

---

6. Legal, Ethical, and Security Considerations

6.1. The Legality of Auto-Applying

Automated bots that scrape LinkedIn and Indeed face significant legal challenges. The case *hiQ Labs v. LinkedIn* established some protections for scraping public data, but platforms enforce strict Terms of Service (TOS) against automation.[43]

- **Risk:** Server-side bots (running on AWS/GCP) are easily detected and blocked via IP fingerprinting.
- **Feno's Mitigation:**
  - **Client-Side Execution:** The extension runs on the user's local machine. It shares the user's cookies, User-Agent, and fingerprint. This makes it indistinguishable from a human user pasting text quickly.[3]
  - **Rate Limiting:** The extension must enforce a "Human Speed" limit (e.g., max 1 application per 2 minutes) to prevent triggering account flags.
  - **Disclaimer:** Feno must explicitly state in its TOS that the user assumes responsibility for compliance with third-party platform rules.

6.2. Data Privacy (GDPR/CCPA)

Resumes contain highly sensitive PII (addresses, phone numbers).

- **Encryption:** All resume data in PostgreSQL must be encrypted at rest (AES-256).
- **AI Data Privacy:** Feno should use OpenAI's **Enterprise API** (or zero-retention policy) to ensure user data is not used to train OpenAI's public models. This is a critical selling point for enterprise or privacy-conscious users.[31]

---

7. UX/UI Design Paradigms

7.1. The "Dynamic Dock"

To maximize the canvas area, Feno minimizes UI clutter using a floating dock component, inspired by the macOS Dock and iOS Dynamic Island.[1]

- **Context-Awareness:**
  - *Default:* Shows[AI Chat][Preview].
  - *Selection Mode:* When a node is selected, the dock morphs to show [Edit][Connect].
  - *AI Mode:* When generating content, the dock expands to show a streaming text interface.
- **Interaction:** Built with **Framer Motion** for fluid, spring-physics-based animations that delight the user.[46]

7.2. Visualizing the "Invisible"

A key UX goal is to visualize ATS scoring logic.

- **The "String" Metaphor:** When a user clicks "Analyze," the system draws bezier curves (strings) from the Job Description keywords to the Resume keywords.
- **Broken Strings:** Missing keywords are visualized as "dangling" strings that snap to a "Warning Node," prompting the user to reconnect them by adding the skill. This gamifies the optimization process.[1]

---

8. Go-to-Market & Monetization Strategy

8.1. Pricing Tiers

Feno will adopt a Product-Led Growth (PLG) freemium model standard in the SaaS industry.

Tier	Price	Features	Rationale
Free	$0	1 Resume, Basic Templates, 5 AI Credits/day, Manual PDF Export.	Sufficient for a user to build one version and taste the "Canvas" experience. High conversion potential via watermarks or feature gating.
Pro	$19/mo	Unlimited Resumes, Unlimited AI, **Portfolio Website**, **Auto-Apply Extension**, Analytics.	Aligned with competitors (Teal/Rezi). The "Killer Feature" bundle (Resume + Site + Bot) justifies the price point.[6]
Lifetime	$149	One-time payment for Pro features.	Appeals to the "Volume Applicant" who expects a 3-6 month job search but dislikes subscriptions. Improves cash flow.[48]

8.2. Growth Loops

- **Viral Footer:** Free portfolios will have a "Built with Feno" badge in the footer. This acts as a high-visibility viral loop when users share their portfolios on LinkedIn.
- **Side-Project Marketing:** Release the "Resume Scorer" as a free, standalone tool (similar to how HubSpot offers free graders) to capture top-of-funnel traffic, then upsell the Builder.[49]

---

9. Implementation Roadmap

Phase 1: The Foundation (Months 1-3)

- **Objective:** Stable Resume Builder.
- **Tech:** React Flow + Tiptap integration.
- **Milestone:** Users can drag nodes, edit rich text without bugs, and export a clean PDF.

Phase 2: The Intelligence (Months 4-6)

- **Objective:** ATS Scoring & AI Chat.
- **Tech:** Python Parser, OpenAI API, Vector DB.
- **Milestone:** "Review Nodes" appear on the canvas. AI can rewrite text contextually.

Phase 3: The Expansion (Months 7-9)

- **Objective:** Portfolio & Auto-Apply.
- **Tech:** Vercel API, Chrome Extension.
- **Milestone:** Users can deploy user.feno.site and use the extension to fill a LinkedIn Easy Apply form.

Phase 4: The Ecosystem (Months 10+)

- **Objective:** Analytics & Community.
- **Tech:** Analytics ingestion, Template marketplace.
- **Milestone:** Dashboard showing "Who viewed my resume" and heatmaps.

---

10. Conclusion

Feno represents a necessary evolution in the career tools market. By treating the resume not as a static document but as a dynamic data node within an infinite workspace, Feno empowers users to manage the increasing complexity of the modern job search. The technical synergy of **React Flow's spatial UI**, **Tiptap's structured editing**, and **Generative AI's transformation capabilities** creates a defensible moat against generic resume builders. With a careful approach to legal compliance in automation and a robust, scalable architecture, Feno is positioned to become the standard operating system for career growth.

Detailed Feature & Technical Breakdown

1. The Resume Builder (Infinite Canvas)

1.1. Core UX: The React Flow Implementation

The choice of **React Flow** transforms the resume building experience from a linear form into a spatial playground. This aligns with the "infinite canvas" trend seen in tools like Miro and Figma, which allows for non-linear thinking—crucial for users trying to connect diverse experiences to job requirements.

- **Node Types Definition:**

Node Type	Description	Functionality
ResumeNode	The main document A4 page(s).	Hosts the **Tiptap** instance. Fixed width, dynamic height. nodrag class applied to editor area. Drag handle at top.

SuggestionNode	AI feedback bubbles.	Small, dismissible nodes. Connected via edges to specific anchor points in the text. Contains "Accept/Reject" buttons.
JobDescNode	Container for pasted Job Description.	Allows users to paste the JD they are targeting. AI uses this node as the "Ground Truth" for scoring the ResumeNode.
PromptNode	Sticky notes for user thoughts.	Simple text areas for users to leave notes for themselves (e.g., "Need to find specific sales numbers").

- **Edge Logic:**
  - Edges are not just visual; they represent data relationships. An edge from a JobDescNode keyword ("Python") to a ResumeNode skill ("Python") visually confirms a match.
  - **Dynamic Anchors:** We must calculate anchor handles dynamically based on the DOM position of specific text paragraphs inside the Tiptap editor. This is complex; as the user types, the text moves.
  - *Technical Strategy:* Use Tiptap decorations to mark text segments with unique IDs. Use ResizeObserver to update React Flow handle positions relative to these DOM elements.[50]

1.2. The Tiptap Editor (WYSIWYG)

Tiptap is chosen over Quill or Draft.js because of its headless nature and robust support for custom node views (React components inside the editor).

- **Custom Extensions:**
  - **Auto-Complete:** As the user types, ghost text appears (greyed out) suggesting the rest of the sentence based on the user's history and the job description. Tab to accept.
  - **Magic Blocks:** A / command menu (Notion-style) to insert complex layouts like "Skill

Bars," "Project Cards," or "Timeline" views.

- **ProseMirror to PDF:**
  - Exporting HTML to PDF while maintaining exact layout is difficult.
  - *Solution:* We will use a headless browser service (e.g., **Puppeteer** or **Playwright**) on the backend. The frontend sends the current JSON state -> Backend renders it to a print-optimized HTML page -> Puppeteer prints to PDF -> Returns Buffer to frontend.[1]

1.3. AI Integration (The "Brain")

- **ATS Scoring Algorithm:**
  - We will not rely solely on LLMs for scoring (which can be inconsistent). We will use a hybrid approach:
    1. **Keyword Matching (Deterministic):** Use a library like natural or compromise in Node.js to extract nouns/skills from the JD and check for existence in the Resume. This provides a "Hard Match" score.[24]
    2. **Semantic Analysis (Probabilistic):** Use OpenAI Embeddings (text-embedding-3-small) to calculate the cosine similarity between the Resume Summary and the Job Description. This captures "vibe" and soft skills matches.[53]
  - **Visualizing Score:** The Score is displayed on the Dock. Clicking it reveals the breakdown.

---

2. The Portfolio Builder

2.1. Resume-to-Website "Magic"

This feature addresses the "Cold Start Problem" of building a portfolio. Users hate writing content. Feno reuses the Resume content.

- **Data Mapping:**
  - *Resume* Experience -> *Portfolio* Timeline Component
  - *Resume* Projects -> *Portfolio* Project Gallery
  - *Resume* Summary -> *Portfolio* Hero Section Bio
  - *Resume* Contact -> *Portfolio* Footer & Contact Form

- **LLM Generation Pipeline:**
  1. **User Action:** Selects "Creative Theme."
  2. **System:** Sends Resume JSON + Theme ID to GPT-4o.
  3. **Prompt:** "Convert this resume data into a website structure using the 'Creative' design system. Return JSON fitting the schema: { hero: { title, subtitle }, projects: }."
  4. **Strict Mode:** Enforce the output schema using OpenAI Structured Outputs to guarantee the UI doesn't break.[26]

2.2. Dynamic Deployment (Vercel Integration)

- **The Challenge:** How to deploy thousands of user sites dynamically?
- **The Solution: Wildcard Subdomains**.
  - Feno app lives at app.feno.com.
  - User portfolios live at *.feno.site.
  - **Next.js Middleware:** Detects the subdomain (e.g., john.feno.site), fetches John's portfolio JSON from the database, and Server-Side Renders the generic Portfolio Page template using John's data.[55]
  - *Advantage:* No build time. Instant updates. The "deployment" is actually just a database update.

---

3. Analytics & Heatmaps

3.1. Resume Link Tracking

- **Mechanism:** When a user shares a resume link (e.g., feno.com/r/xyz), we serve an intermediate page that logs the visit, User-Agent (device/browser), and Referrer before redirecting or rendering the PDF viewer.
- **Data Point:** "Time spent viewing" can be approximated by sending "heartbeat" pings from the viewer to the server every 5 seconds.

3.2. Portfolio Heatmaps

- **Technical Difficulty:** True heatmaps (like Hotjar) require heavy script recording which slows down sites.
- **Feno Approach:** "Lightweight Interaction Tracking."
    - Instead of recording video, we track click coordinates (x,y) relative to the viewport width.
    - **Visualization:** On the Feno dashboard, we overlay these (x,y) points on an iframe of the user's portfolio, generating a canvas-based heatmap overlay. This is cheaper and privacy-friendlier than full session recording.[56]

---

4. Risks and Mitigation Strategies

Risk Area	Description	Mitigation Strategy
Platform Risk	LinkedIn/Indeed blocking the Auto-Apply extension.	Use **client-side execution** only. Randomize delays between actions to mimic human behavior. Do not scrape data aggressively; only interact with the DOM to fill fields.[44]
AI Quality	LLM generating fake skills or hallucinations in the resume.	**User-in-the-loop:** AI suggestions must be explicitly accepted. Never auto-replace text without user confirmation. Use low-temperature settings (0.2) for factual tasks.
Performance	Infinite Canvas becoming laggy with many nodes.	Use **virtualization** (only render nodes in the viewport). Debounce Tiptap editor updates to the React Flow state. Optimize re-renders using React.memo.[57]

Legal	Storing user data (GDPR/CCPA).	Implement "Data Purge" settings. Store resumes in specific regions if required. Anonymize analytics data.

---

5. Go-to-Market & Future Roadmap

5.1. Launch Strategy

- **Product Hunt Launch:** Focus on the "Visual" aspect. The Infinite Canvas is the hook.
- **Influencer Marketing:** Target "Tech Twitter" and LinkedIn influencers with the "One-Click Portfolio" feature, as it gives immediate visual gratification.

5.2. Roadmap

- **Q1: Core Resume Builder.** Canvas, Tiptap, PDF Export.
- **Q2: Portfolio & Deploy.** Resume-to-Web pipeline, Vercel integration.
- **Q3: Intelligence Layer.** Auto-Apply Extension, Analytics Dashboard.
- **Q4: Community.** "Remixable" Resume Templates (Figma Community style), public profiles.

---

6. Detailed User Flow Scenarios

6.1. Scenario A: The "Sprint" Application

1. **User** finds a job link on LinkedIn.
2. **User** pastes the link into Feno's "JobDescNode" on the canvas.

3. **System** scrapes the JD text.
4. **Feno AI** analyzes the Resume vs. JD.
5. **System** highlights 3 missing keywords in red on the canvas.
6. **User** clicks "Fix". AI rewrites the bullet points to include keywords.
7. **User** clicks "Export PDF" -> "Tailored_Resume_Google.pdf".
8. **User** opens the LinkedIn "Easy Apply" form.
9. **Feno Extension** detects the form, uploads the specific PDF, and fills the questions.
10. **User** hits Submit.

6.2. Scenario B: The "Portfolio Update"

1. **User** adds a new "Senior Developer" role to their Resume on the canvas.
2. **User** clicks "Update Portfolio".
3. **System** detects the new entry.
4. **AI** asks: "Do you want to feature this new role on your homepage?"
5. **User** says "Yes".
6. **System** regenerates the Portfolio JSON, adding a "Featured Role" badge to the timeline.
7. **System** pushes the update to john.feno.site instantly.

Works cited

1. Feno Landing ideas.pdf
2. Durable AI Website Builder: Features, Pricing, Plans and Set-up - Neo email, accessed on November 26, 2025, https://www.neo.space/blog/durable-ai-website-builder
3. LazyApply : Job Application Bot - Chrome Web Store, accessed on November 26, 2025, https://chromewebstore.google.com/detail/lazyapply-job-application/pgnfaifdbfoiehcndkoeemaifhhbgkmm
4. Kickresume Alternative: Rezi vs Kickresume (AI Resume Builder Comparison) - Rezi AI, accessed on November 26, 2025, https://www.rezi.ai/posts/kickresume-alternative
5. Free AI Resume Builder - Create ATS-Friendly Resumes - Teal, accessed on November 26, 2025, https://www.tealhq.com/tools/resume-builder
6. Is Kickresume a good alternative to Teal? Here's a quick comparison, accessed on November 26, 2025, https://www.kickresume.com/en/help-center/alternative-to-teal/
7. Free AI Resume Builder Trusted by +4 Million Job Seekers, accessed on November 26, 2025, https://www.rezi.ai/
8. Kickresume | Best Online Resume & Cover Letter Builder, accessed on November 26, 2025, https://www.kickresume.com/en/

9. Is Kickresume a good alternative to Rezi? Here's a quick comparison, accessed on November 26, 2025, https://www.kickresume.com/en/help-center/alternative-to-rezi/
10. Which auto job application tool do you recommend? - Reddit, accessed on November 26, 2025, https://www.reddit.com/r/jobs/comments/1b773h6/which_auto_job_application_tool_do_you_recommend/
11. The Pitfalls Of Using Auto-Submitter Bots For Job Applications - Forbes, accessed on November 26, 2025, https://www.forbes.com/councils/forbescoachescouncil/2025/04/14/the-pitfalls-of-using-auto-submitter-bots-for-job-applications/
12. Personal Portfolio Site Template - Gamma, accessed on November 26, 2025, https://gamma.app/templates/personal-portfolio-site-7AtX1g9ENCXmHALB4qhZwL
13. Generative UI with React Components - hashbrown.dev, accessed on November 26, 2025, https://hashbrown.dev/docs/react/concept/components
14. Simplify Copilot | Autofill Job Applications and Track Jobs, accessed on November 26, 2025, https://simplify.jobs/copilot
15. Custom Nodes - React Flow, accessed on November 26, 2025, https://reactflow.dev/learn/customization/custom-nodes
16. React Flow: Node-Based UIs in React, accessed on November 26, 2025, https://reactflow.dev/
17. Built-In Components - React Flow, accessed on November 26, 2025, https://reactflow.dev/learn/concepts/built-in-components
18. ueberdosis/tiptap: The headless rich text editor framework for web artisans. - GitHub, accessed on November 26, 2025, https://github.com/ueberdosis/tiptap
19. AI in Rich-Text Editor: Implementing Smart Features | by Michal Ganani Shuvi, accessed on November 26, 2025, https://blog.ai21.engineering/ai-in-rich-text-editor-implementing-smart-features-3bba98e46b2e
20. Semantic Search | Tiptap Editor Docs, accessed on November 26, 2025, https://tiptap.dev/docs/collaboration/documents/semantic-search
21. Drag Handle - React Flow, accessed on November 26, 2025, https://reactflow.dev/examples/nodes/drag-handle
22. Utility Classes - React Flow, accessed on November 26, 2025, https://reactflow.dev/learn/customization/utility-classes
23. Drag Handle React | Tiptap Editor Docs, accessed on November 26, 2025, https://tiptap.dev/docs/editor/extensions/functionality/drag-handle-react
24. Creating a Game-Changer in Job Search: An Open Source ATS Resume Matcher, accessed on November 26, 2025, https://dev.to/srbhr/creating-a-game-changer-in-job-search-an-open-source-ats-resume-matcher-31g9
25. Schema - JSON Resume, accessed on November 26, 2025, https://jsonresume.org/schema
26. Structured Outputs: Everything You Should Know - Humanloop, accessed on

November 26, 2025, https://humanloop.com/blog/structured-outputs
27. Hosting your API on Vercel | Knowledge Base, accessed on November 26, 2025, https://vercel.com/kb/guide/hosting-backend-apis
28. How to Dynamically Add Custom Domains in Vercel for Your SaaS Application - Medium, accessed on November 26, 2025, https://medium.com/@nurulislamrimon/how-to-dynamically-add-custom-domains-in-vercel-for-your-saas-application-9cdf16268187
29. Simplify Copilot - Autofill job applications, job tracker & AI resumes - Chrome Web Store, accessed on November 26, 2025, https://chromewebstore.google.com/detail/simplify-copilot-autofill/pbanhockgaggenencehbnadejlgchfc?hl=en
30. Job Application Automation: Is it Safe and Ethical?, accessed on November 26, 2025, https://blog.fastapply.co/job-application-automation-is-it-safe-and-ethical
31. plausible/analytics: Simple, open source, lightweight and privacy-friendly web analytics alternative to Google Analytics. - GitHub, accessed on November 26, 2025, https://github.com/plausible/analytics
32. Umami, accessed on November 26, 2025, https://umami.is/
33. 10 of the Best Web Analytics Tools for React Websites - ButterCMS, accessed on November 26, 2025, https://buttercms.com/blog/web-analytics-tools-react-websites/
34. React | Tiptap Editor Docs, accessed on November 26, 2025, https://tiptap.dev/docs/editor/getting-started/install/react
35. Top Free Resume Parser tools, APIs, and Open Source models - Eden AI, accessed on November 26, 2025, https://www.edenai.co/post/top-free-resume-parser-tools-apis-and-open-source-models
36. srbhr/Resume-Matcher: Improve your resumes with Resume Matcher. Get insights, keyword suggestions and tune your resumes to job descriptions. - GitHub, accessed on November 26, 2025, https://github.com/srbhr/Resume-Matcher
37. Using OpenAI and Python to Enhance Your Resume: A Step-by-Step Guide - Medium, accessed on November 26, 2025, https://medium.com/data-science/using-openai-and-python-to-enhance-your-resume-a-step-by-step-guide-e2c1a359e194
38. Open Source ATS Tool with Streamlit "Resume Matcher" powered by Vector Search, accessed on November 26, 2025, https://discuss.streamlit.io/t/open-source-ats-tool-with-streamlit-resume-matcher-powered-by-vector-search/47412
39. How to use structured outputs with Azure OpenAI in Microsoft Foundry Models, accessed on November 26, 2025, https://learn.microsoft.com/en-us/azure/ai-foundry/openai/how-to/structured-outputs?view=foundry-classic
40. Enhancing Reliability of LLM Outputs with Structured JSON and Pydantic Models - Medium, accessed on November 26, 2025, https://medium.com/@tajinderpalsingh61/enhancing-reliability-of-llm-outputs-wit

h-structured-json-and-pydantic-models-6b528dadede5
41. Using the REST API - Vercel API Docs, accessed on November 26, 2025, https://vercel.com/docs/rest-api
42. Programmatic SEO in Next.js 15 - Full Guide - YouTube, accessed on November 26, 2025, https://www.youtube.com/watch?v=290Ytj96vL4
43. Is Job Scraping Legal? Yes, when done in compliance. - Mantiks, accessed on November 26, 2025, https://en.blog.mantiks.io/is-job-scraping-legal/
44. Ask HN: LinkedIn sent me a cease and desist for my Chrome extension. Help?, accessed on November 26, 2025, https://news.ycombinator.com/item?id=34583932
45. macOS Dock - shadcn.io, accessed on November 26, 2025, https://www.shadcn.io/components/dock/mac-os-dock
46. Dock | React Components & Templates - Magic UI, accessed on November 26, 2025, https://magicui.design/docs/components/dock
47. AI Resume Builders vs. Professional Resume Writers: Cost Comparison - Career Pro Guider, accessed on November 26, 2025, https://careerproguider.com/blog/ai-vs-professional-resume-cost
48. Looking For The Best AI Resume Builder? Here Are 7 Options | Resumonk Articles, accessed on November 26, 2025, https://www.resumonk.com/articles/best-ai-resume-builder
49. How do you decide on pricing tiers for a SaaS? Free tier vs free trial? How do you split features? - Reddit, accessed on November 26, 2025, https://www.reddit.com/r/SaaS/comments/1mf0ty9/how_do_you_decide_on_pricing_tiers_for_a_saas/
50. React node views | Tiptap Editor Docs, accessed on November 26, 2025, https://tiptap.dev/docs/editor/extensions/custom-extensions/node-views/react
51. How to Add a Visual Indicator to a Node in Tiptap During Drag-and-Drop Operations?, accessed on November 26, 2025, https://stackoverflow.com/questions/78433241/how-to-add-a-visual-indicator-to-a-node-in-tiptap-during-drag-and-drop-operation
52. The Benefits of Semantic Search Over Keyword Matching in Resume Screening - Brainner, accessed on November 26, 2025, https://www.brainner.ai/blog/article/the-benefits-of-semantic-search-over-keyword-matching-in-resume-screening
53. Improved Candidate-Career Matching Using Comparative Semantic Resume Analysis, accessed on November 26, 2025, https://www.astesj.com/v09/i01/p03/
54. Introducing Structured Outputs in the API - OpenAI, accessed on November 26, 2025, https://openai.com/index/introducing-structured-outputs-in-the-api/
55. Subdomain-Based Routing in Next.js: A Complete Guide for Multi-Tenant Applications, accessed on November 26, 2025, https://medium.com/@sheharyarishfaq/subdomain-based-routing-in-next-js-a-complete-guide-for-multi-tenant-applications-1576244e799a
56. What is a Heatmap - Comprehensive Guide and Best Practices - VWO, accessed on November 26, 2025, https://vwo.com/website-heatmap/what-is-a-heatmap/
57. Performance - React Flow, accessed on November 26, 2025,

https://reactflow.dev/learn/advanced-use/performance