A Comprehensive Analysis of Open-Source Templates and Libraries for the "old.new" AI Generation Platform

Executive Summary: A Strategic Asset Integration Framework for "old.new"

This report presents a comprehensive analysis of open-source assets on GitHub, tailored for integration into the "old.new" AI tool for generating documents, diagrams, and infographics. The investigation identifies and evaluates top-tier repositories that are customizable, modern, and embeddable within a React/Node.js application architecture. The findings are organized into three core domains, each with a distinct strategic recommendation designed to maximize quality, scalability, and development efficiency.

The key findings and high-level recommendations are as follows:

- **Documents:** The analysis strongly recommends adopting the **JSON Resume** standard as the foundational technology for document generation. Its mature ecosystem, inherent Applicant Tracking System (ATS) friendliness, and clear architectural separation of content from presentation provide a robust and scalable framework for generating professional resumes and CVs.[1]
- **Diagrams:** A dual-strategy approach is proposed. For the initial product offering, leveraging **Mermaid.js** for its powerful text-to-diagram generation capabilities offers the most rapid path to market.[2] For future enhancements and user-centric features, integrating
  React Flow will enable fully interactive, user-editable diagrams, creating a significant value-add.[3]
- **Infographics:** A tiered strategy is advised to balance breadth of coverage with implementation complexity. The report recommends starting with **Chart.js** and its extensive plugin ecosystem for its versatility and ease of integration.[4] For highly polished, React-native components,
  Recharts is the preferred solution.[5] Finally,

D3.js should be reserved for creating bespoke, high-value visualizations that can serve as a unique brand differentiator.[6]

This report provides a detailed examination of these technologies, offering a curated catalog of assets and a strategic roadmap for their phased implementation within the "old.new" platform.

Part I: Document Generation Templates & Frameworks

This section focuses on identifying the most robust, customizable, and ATS-friendly solutions for generating professional documents, with a primary focus on resumes and curricula vitae. The analysis centers on a programmatic approach that aligns with the capabilities of an AI content generator.

1.1 The JSON Resume Ecosystem: A Foundation for Programmatic, ATS-Friendly Documents

The JSON Resume project is an open-source initiative to create a standardized, JSON-based schema for resumes.[1] Its core principle is the decoupling of resume

content from its *presentation*. A user's professional history, skills, and education are stored in a structured, machine-readable JSON file, which can then be rendered into a multitude of visual styles using themes. This paradigm is exceptionally well-suited for "old.new," as the AI's primary task becomes generating the structured JSON data, which can then be applied to any user-selected template without altering the underlying information.

The ecosystem is supported by a mature set of tools. The primary command-line interface, resume-cli, is used for validating schemas and rendering resumes with different themes.[7] A more powerful alternative,

HackMyResume, supports both the JSON Resume standard and its own "FRESH" format, offering advanced capabilities such as merging multiple resume files and analyzing content for keywords and employment gaps.[9] This tool could serve as a powerful backend engine for "old.new"'s generation and analysis features. The official

jsonresume.org monorepo, which hosts the project's homepage, registry, and a set of official

themes, signals a well-maintained and active core project.[1]

The maturity of this ecosystem is evident in the sheer volume of available themes, with community-maintained lists cataloging over 400 distinct options.[10] While this breadth indicates a vibrant and engaged community, it also presents a significant challenge. The quality, maintenance status, and design modernity of these themes vary dramatically. Many individual theme repositories have not been updated in years, and community-curated "best of" lists have emerged to help navigate the noise.[10] Therefore, the primary technical task for "old.new" is not simply finding themes, but implementing a robust system for vetting, curating, and quality-controlling a select subset of high-quality templates to offer its users. A direct, unfiltered integration of all available themes would result in a poor and inconsistent user experience. The platform's value will be derived from offering a pre-vetted gallery of modern, reliable, and ATS-friendly templates.

1.2 Analysis of Top-Tier JSON Resume Themes

A detailed review of the JSON Resume ecosystem reveals several standout themes that serve as excellent candidates for "old.new"'s curated template gallery. These themes are distinguished by their modern design, advanced features, and high degree of customizability.

The analysis begins with the officially supported themes listed in the main jsonresume.org repository, which represent a baseline of quality and compatibility. These include widely used themes like jsonresume-theme-flat [11],

jsonresume-theme-elegant [12],

jsonresume-theme-onepage [13], and the modern

jsonresume-theme-tailwind.[1]

Beyond the official list, several community themes demonstrate superior features and engineering:

- **jsonresume-theme-even**: This theme is a prime example of a modern, feature-rich template. Inspired by jsonresume-theme-flat, it adds crucial enhancements such as Markdown support for rich text formatting, light and dark modes, and—most importantly—the ability to customize colors directly within the resume.json file itself. This level of programmatic customization is highly desirable for an AI platform, allowing for dynamic styling based on user preferences or AI suggestions.[14]
- **jsonresume-theme-caffeine**: This repository is notable for its developer-friendly setup, featuring a modular file system and detailed instructions for customization. It uses SCSS

variables for easy modification of colors and layout, providing a high degree of control for developers looking to create a branded or unique version for the "old.new" platform.[15]

- **@bluesialia/jsonresume-theme-bluetime**: Representing the cutting edge of the ecosystem, this theme is built with TypeScript and designed as an ES module, making it ideal for seamless integration into modern React/Node applications. It distinguishes itself by embedding all CSS, meaning there are no external stylesheets to manage during rendering. Furthermore, its high test coverage indicates excellent code quality and reliability, reducing the risk of rendering bugs.[16]
- **jsonresume-theme-kendall**: A popular and well-documented theme that relies on the familiar Bootstrap and FontAwesome libraries. Its widespread use and conventional technology stack make it easy to understand, extend, and maintain.[17]

Despite the power of this ecosystem, its tooling layer presents integration challenges. Multiple sources report that the command-line tools can be brittle, and theme integration is not always seamless.[7] Specific GitHub issues reveal that

resume-cli can fail to resolve theme paths correctly, particularly when dealing with globally versus locally installed packages, often requiring developers to use verbose, relative paths as a workaround.[18] This indicates that a naive implementation that simply wraps the existing CLI tools would be fragile and prone to failure. To ensure a robust and reliable service, the "old.new" engineering team should develop a dedicated backend rendering service. This service would need to manage theme dependencies explicitly, for instance by installing them programmatically into a sandboxed environment for each render job, rather than relying on a potentially unstable

node_modules structure.

1.3 Catalog of Document & Resume Templates

The following table provides a curated, pre-vetted list of high-quality JSON Resume themes. This catalog serves as an actionable starting point for building the "old.new" template gallery, saving significant time in manual review and evaluation. The themes have been selected based on modern design principles, customizability, and features that align with the requirements of a professional, ATS-friendly document generation platform.

Table 1: Curated Catalog of High-Quality JSON Resume Themes

Theme	Repository	Community	Key	ATS-Friendl	Notes

Name	Link	Metrics (Stars/Forks)	Features	iness Score (1-5)	
jsonresume-theme-even	rbardini/jsonresume-theme-even	36 / 40 [14]	Dark Mode, Markdown, Customizable Colors, TypeScript, CSS Grid	5	A top-tier modern theme with excellent programmatic customization options.
@bluesialia/jsonresume-theme-bluetime	(https://github.com/BlueSialia/jsonresume-theme-bluetime)	(N/A)	TypeScript, ES Module, Responsive, Embedded CSS, High Test Coverage	5	Ideal for modern React/Node apps due to its self-contained, high-quality build.
jsonresume-theme-elegant	mudassir0909/jsonresume-theme-elegant	126 / 146 [12]	Responsive, Card Layout, LESS, Highly Customizable	4	A very popular and visually appealing theme with a card-based design.
jsonresume-theme-kendall	(https://github.com/LinuxBozo/jsonresume-theme-kendall)	66 / 87 [17]	Bootstrap, FontAwesome, Well-Documented, Responsive	5	A solid, popular choice based on familiar and reliable technologies.

jsonresume-theme-caffeine	kelyvin/jsonresume-theme-caffeine	(N/A)	SCSS, Modular, FontAwesome, Highly Customizable	4	Excellent developer experience and deep customization via SCSS variables.
jsonresume-theme-flat	erming/jsonresume-theme-flat	42 / 69 [11]	Minimalist, Clean, Official Theme	5	A classic, clean, and highly readable theme that is very ATS-friendly.
jsonresume-theme-onepage-plus	vkcelik/jsonresume-theme-onepage-plus	8 / 8 [13]	Compact, Print-Optimized, Official Theme	5	An updated version of a classic theme, designed for a clean, single-page layout.
jsonresume-theme-kards	(https://github.com/XuluWarrior/jsonresume-theme-kards)	10 / 8 [19]	Card-based, LESS, Customizable Backgrounds	3	Visually striking design, but may be less ATS-friendly due to its layout.
jsonresume-theme-orbit	(https://github.com/XuluWarrior/jsonresume-theme-orbit)	9 / 7 [20]	Sidebar Layout, 6 Color Schemes, LESS	4	Professional sidebar design with built-in color variants for easy

					customizati on.
jsonresume -theme-sta ckoverflow	anthonyjdel la/customiz ed-jsonresu me-theme- stackoverfl ow	3 / 2 [21]	2-Page PDF, Custom Sections (Speaking, Articles)	4	Based on the recognizabl e Stack Overflow developer story format.
jsonresume -theme-tail wind	jsonresume /jsonresum e.org	161 / 39 [1]	Tailwind CSS, Modern, Official Theme	5	Utilizes the popular Tailwind CSS framework for a modern, utility-first design.

Part II: Diagramming & Graph Visualization Libraries

This section evaluates libraries for generating diagrams, ranging from simple flowcharts to complex, interactive node-based graphs. The analysis prioritizes solutions that align with an AI-driven generation workflow and offer a path for future interactivity and customization.

2.1 Interactive Node-Based Diagrams with React Flow

React Flow, part of the **xyflow** project, stands out as the leading open-source React library for building interactive, node-based editors and diagrams.[3] Its architecture, where nodes and edges are treated as fully customizable React components, makes it a perfect fit for the "old.new" application. This gives developers complete control over the visual appearance and

behavior of every element in a generated diagram.

While the primary use case for "old.new" is AI-driven generation, React Flow's built-in interactivity features—such as zooming, panning, and element dragging—open a clear and powerful path for a future "version 2.0" feature where users can manually edit or refine the diagrams produced by the AI. The ecosystem is rich with examples and starter kits for various applications, including workflow automation builders, mind maps, and AI-powered GUIs, providing excellent, real-world implementation patterns.[3] Repositories like

Azim-Ahmed/Automation-workflow [22] and

xyflow/react-flow-example-apps [23] serve as valuable references.

For more advanced or specialized use cases, other libraries in this space are worth noting. **jsPlumb** is a powerful, framework-agnostic alternative for visual connectivity, indicated by its maturity and high community adoption.[24]

SuperViz offers a unique capability by providing SDKs to add real-time collaboration features, such as multiplayer cursors and contextual comments, on top of existing visualization tools, including React Flow.[25] This represents a potential future enhancement for collaborative work within "old.new".

2.2 Declarative & Text-to-Diagram Generation with Mermaid.js

Mermaid.js is a JavaScript library that generates a wide variety of complex diagrams—including flowcharts, sequence diagrams, Gantt charts, and class diagrams—from a simple, Markdown-like text syntax.[2] This declarative approach holds immense strategic value for "old.new." For an AI tool that generates output from user text, Mermaid.js is a perfect architectural match. The AI's task is simplified from generating complex vector graphics or a tree of React components to generating a simple, structured block of text. This dramatically reduces the complexity of the AI model and the entire generation pipeline, making it an ideal choice for a core feature in the initial product release.

The Mermaid ecosystem is robust, with the core mermaid library serving as the rendering engine [2] and the popular

mermaid-live-editor providing an interactive playground for development and user education.[28]

However, this approach involves a trade-off. Mermaid.js offers unparalleled ease of generation

but provides less granular control over specific styling and layout compared to component-based libraries like React Flow. While Mermaid supports theming and CSS overrides, its fundamental layout algorithms are less configurable than a library where every node's position and style can be controlled programmatically. This leads to a clear strategic path for "old.new": for its initial version, the simplicity and power of Mermaid's text-to-diagram paradigm is the optimal choice. For future versions that require highly bespoke, brand-specific, or interactive diagrams, a more complex integration with a library like React Flow would be necessary. The platform's architecture should be designed to accommodate multiple rendering engines, allowing the AI to generate a high-level diagram definition that can be compiled into either Mermaid syntax or a React Flow component structure as needed.

2.3 Specialized Graph Layouts: D3-DAG and Dagre

For advanced diagramming scenarios, d3-dag and dagre-d3 are not charting libraries themselves, but powerful, low-level layout *algorithms* designed specifically for Directed Acyclic Graphs (DAGs).[29] These libraries are essential when the default layouts of higher-level tools like Mermaid or React Flow are insufficient. For example, if "old.new" needs to visualize complex software dependency chains, genetic data, or intricate process flows, these libraries can produce mathematically optimal, non-overlapping layouts using established algorithms like the Sugiyama or Zherebko methods.[29]

The typical integration pattern involves using these libraries in a Node.js backend. The process is as follows: first, the graph structure (nodes and edges) is defined; second, this structure is passed to a library like d3-dag to calculate the precise coordinates for each node; finally, these coordinates are used to render the final output on the frontend using a library like React Flow or even plain SVG. For direct client-side rendering, react-d3-dag provides a convenient React wrapper that simplifies this process.[31]

2.4 Catalog of High-Impact Diagramming Repositories

The following table provides a strategic overview of the different classes of diagramming tools. It is designed to help the "old.new" team understand the trade-offs between different architectural approaches (text-based vs. component-based vs. algorithm-driven) and select the right tool for specific product features.

Table 2: High-Impact Diagramming & Graphing Repositories

Library/Repository	Category/Approach	Repository Link	Community Metrics (Stars/Forks)	React/Node Integration Notes	Primary "old.new" Use Case
xyflow/react-flow	Interactive/Node-based	xyflow/react-flow	21.2k / 1.4k	Native React components, highly extensible.	Future user-editable workflow builder, interactive mind maps.
mermaid-js/mermaid	Declarative/Text-to-Diagram	mermaid-js/mermaid	65.5k / 5.7k [2]	Renders SVG from text string; easily integrated in Node/React.	MVP static diagrams (flowcharts, sequence, Gantt) generated by AI.
erikbrinkman/d3-dag	Layout Engine	erikbrinkman/d3-dag	1.5k / 91 [29]	Node.js backend processing to calculate node coordinates.	Advanced layout for complex, non-overlapping directed graphs.
dagrejs/dagre-d3	Layout Engine & Renderer	dagrejs/dagre-d3	2.9k / 595 [30]	Renders D3 graphs from a calculated layout.	Backend or client-side layout for directed graphs.
jsplumb/jsplumb	Interactive/Node-base	jsplumb/jsplumb	7.8k / 1.4k [24]	Framework-agnostic, works with	Alternative to React Flow if a

	d			vanilla JS, React, Vue, etc.	non-React-specific solution is needed.
SuperViz/superviz	Collaboration Layer	(https://github.com/SuperViz/superviz)	381 / 2 [25]	SDK that integrates with other libraries like React Flow.	Future enhancement for real-time collaborative diagram editing.

Part III: Infographic & Data Charting Libraries

This section evaluates and catalogs libraries for creating data visualizations and infographics, from standard bar and line charts to complex, specialized diagrams like Sankey flows and geographical maps. The analysis provides a comparative framework for selecting the appropriate tool based on the desired level of abstraction and visual complexity.

3.1 Comparative Analysis: Chart.js vs. Recharts vs. D3.js

The open-source web charting landscape is dominated by three major players, each representing a different point on the spectrum of abstraction versus power.

- **Chart.js:** This library is best understood as the accessible, general-purpose solution for data visualization. It is easy to get started with, has clear documentation, and covers the most common chart types (bar, line, pie, etc.) out of the box.[32] Its primary strength, however, lies in a vast and modular plugin ecosystem that allows developers to extend its core capabilities to include highly specialized chart types without a steep learning curve.
- **Recharts:** This is the premier **React-native** charting library. Its fundamental advantage is its composable, component-based architecture, which feels natural to React developers.[5] Charts are built by assembling declarative components like <LineChart>, <CartesianGrid>, and <Tooltip>, enabling deep integration with the React lifecycle and state management.[34] This makes it the ideal choice for building highly polished, interactive dashboards and infographics directly within the "old.new" React

application.

- **D3.js:** D3.js is not a charting library in the traditional sense; it is a low-level **data visualization kernel**.[6] It provides the fundamental, data-driven building blocks—scales, axes, shapes, transitions—to create virtually any data visualization imaginable. Its power is unmatched, but it comes with the highest complexity. Many popular charting libraries, such as
  billboard.js [35] and
  britecharts [36], are built on top of D3, which underscores its foundational role in the ecosystem.

3.2 The Chart.js Plugin Ecosystem: Extending Core Capabilities

The true power of Chart.js for a platform like "old.new" lies not in its core library, but in its rich, modular plugin ecosystem. This ecosystem allows for the generation of highly specialized infographics that would otherwise require the complexity of D3. The chartjs/awesome repository serves as the central, curated index for discovering these plugins and is a critical resource for development.[4]

This plugin architecture allows for a highly scalable approach to infographic generation. Key plugin categories include:

- **New Chart Types:** These plugins add entirely new type options to the Chart.js configuration, enabling the creation of sophisticated visualizations with minimal effort.
  - chartjs-chart-boxplot: Adds support for statistical box and violin plots.[37]
  - chartjs-chart-geo: Enables the creation of choropleth and bubble maps for visualizing geographical data.[38]
  - chartjs-chart-matrix: Provides matrix and heatmap-style charts, useful for correlation matrices.[40]
  - chartjs-chart-sankey: Adds Sankey diagrams, which are excellent for illustrating flow, such as user journeys or budget allocations.[41]
  - chartjs-chart-funnel: Creates funnel charts, ideal for visualizing conversion processes or sales pipelines.[43]
  - chartjs-chart-graph: Renders force-directed graphs and tree diagrams for network visualization.[44]
- **Enhanced Features & Interactivity:** These plugins augment existing charts with new capabilities.
  - chartjs-plugin-annotation: Allows for drawing lines, boxes, and labels directly onto the chart area, which is crucial for calling out specific data points or trends.[45]
  - chartjs-plugin-datalabels: A highly popular plugin for displaying labels directly on

data elements (e.g., the value on top of a bar), greatly improving readability.[47]

By adopting Chart.js, "old.new" gains access to a massive library of pre-built, specialized infographic components. The AI can be trained to recognize when a user's text describes a flow (output a Sankey diagram), a geographical distribution (output a choropleth map), or a process pipeline (output a funnel chart), and then invoke the appropriate Chart.js plugin to render the visualization. This provides a scalable and efficient path to generating a wide variety of rich infographics.

3.3 High-Fidelity & Specialized Visualizations with D3.js

For visualizations that require maximum flexibility or a completely novel design, the D3.js ecosystem offers unparalleled power. The wbkd/awesome-d3 repository is a curated gateway to the vast collection of D3 modules and helper libraries.[6]

The D3 ecosystem can be broadly categorized into:

- **Chart Libraries Built on D3:** These libraries provide higher-level abstractions over D3's core functionality, offering pre-built charts with the power of D3 under the hood. Notable examples include billboard.js [35],
  britecharts [36],
  nvd3, and plotly.js.
- **Specialized D3 Plugins:** These are focused modules that solve specific visualization problems. This includes advanced mapping tools (d3-geo-projection), network graph layouts (d3-force), and utility helpers for common tasks like creating legends (d3-legend) or adding SVG patterns (textures).

Within the "old.new" platform, D3 should be reserved for "premium" or highly custom visualization types that cannot be achieved with the more abstract libraries. It represents the high-end of the platform's capabilities and could be used to generate unique, signature infographic styles that become a hallmark of the "old.new" brand, offering users visualizations that are not available from other tools.

3.4 Catalog of Modern Charting Libraries & Components

The following table provides a comprehensive, categorized catalog of charting libraries and plugins. It is structured around the three primary ecosystems—Chart.js, Recharts, and

D3.js—to allow the "old.new" team to easily find relevant tools based on their chosen integration strategy. This list forms the core of the asset discovery process for infographics.

Table 3: Comprehensive Catalog of Charting Libraries & Plugins

Library/Plugin Name	Core Technology	Repository Link	Community Metrics (Stars/Forks)	Supported Chart Types / Key Features
Chart.js Ecosystem				
chartjs/Chart.js	Core Library	chartjs/Chart.js	63.3k / 11.9k	Bar, Line, Pie, Doughnut, Radar, Polar Area, Bubble, Scatter
sgratzl/chartjs-chart-geo	Chart.js Plugin	sgratzl/chartjs-chart-geo	379 / 53 [39]	Choropleth Maps, Bubble Maps
kurkle/chartjs-chart-sankey	Chart.js Plugin	kurkle/chartjs-chart-sankey	169 / 35 [41]	Sankey Diagrams
sgratzl/chartjs-chart-graph	Chart.js Plugin	sgratzl/chartjs-chart-graph	203 / 25 [44]	Force-directed Graphs, Tree Diagrams, Dendrograms
chartjs/chartjs-plugin-datalabels	Chart.js Plugin	chartjs/chartjs-plugin-datalabels	908 / 501 [47]	Displays labels directly on data elements
chartjs/chartjs-plugin-annotation	Chart.js Plugin	chartjs/chartjs-plugin-annotation	619 / 370 [45]	Draws lines, boxes, points, and labels on chart area

sgratzl/chartjs-chart-boxplot	Chart.js Plugin	sgratzl/chartjs-chart-boxplot	121 / 29 [37]	Box Plots, Violin Plots
kurkle/chartjs-chart-matrix	Chart.js Plugin	kurkle/chartjs-chart-matrix	234 / 18 [40]	Matrix, Heatmap
sgratzl/chartjs-chart-funnel	Chart.js Plugin	sgratzl/chartjs-chart-funnel	26 / 15 [43]	Funnel Charts
Recharts & React Ecosystem				
recharts/recharts	Core Library	recharts/recharts	21.6k / 1.6k	Composable React components for all standard chart types
sriramveeraghanta/react-recharts-examples	Recharts Examples	sriramveeraghanta/react-recharts-examples	4 / 0 [5]	Example implementations of various Recharts charts
jacquelynmarcella/Project3	Recharts Examples	jacquelynmarcella/Project3	0 / 0 [5]	MERN stack application demonstrating data visualization
D3.js Ecosystem				
d3/d3	Core Library	d3/d3	107k / 23.3k	Low-level data visualization kernel for custom graphics

naver/billboard.js	D3 Chart Library	naver/billboard.js	5.9k / 360 [35]	Re-usable chart library based on D3.js
britecharts/britecharts	D3 Chart Library	britecharts/britecharts	3.7k / 217 [36]	Composable charting library based on D3 components
plottable/plottable	D3 Chart Library	plottable/plottable	2.9k / 380	Flexible, interactive charts for the web
dc-js/dc.js	D3 Chart Library	dc-js/dc.js	7.4k / 1.4k	Multi-dimensional charting built to work with crossfilter
d3-annotation/d3-annotation	D3 Utility	d3-annotation/d3-annotation	903 / 110	Add annotations to D3 charts
susielu/d3-legend	D3 Utility	susielu/d3-legend	974 / 179	Helper for creating legends for D3 charts
d3/d3-geo	D3 Module	d3/d3-geo	1.1k / 247	Geographic projections, shapes, and spherical math
d3/d3-force	D3 Module	d3/d3-force	1.3k / 274	Force-directed graph layout algorithm

Conclusion: A Unified Asset Strategy for "old.new"

This analysis of the open-source landscape reveals a rich and diverse toolkit available for building the "old.new" AI generation platform. The success of the platform will depend not just on selecting the correct individual libraries, but on implementing a robust, scalable architecture that can orchestrate these diverse assets to deliver a seamless and powerful user experience. The strategic recommendations for documents, diagrams, and infographics provide a clear path forward.

A phased implementation is recommended to manage complexity and accelerate time-to-market:

- **Phase 1 (MVP):** Focus on the highest value, lowest complexity integrations to establish a core product offering.
  - **Documents:** Implement the JSON Resume engine with a curated set of 10-15 top-tier themes, such as even, caffeine, and bluetime.
  - **Diagrams:** Integrate Mermaid.js for text-to-diagram generation, covering essential types like flowcharts and sequence diagrams.
  - **Infographics:** Integrate the Chart.js core library along with 5-7 of the most impactful plugins, including chartjs-plugin-datalabels, chartjs-plugin-annotation, chartjs-chart-sankey, and chartjs-chart-geo.
- **Phase 2 (Expansion):** Broaden capabilities and enhance the user experience with more advanced and interactive features.
  - **Documents:** Expand the theme library based on user feedback and usage metrics, potentially adding a theme marketplace or customization options.
  - **Diagrams:** Introduce a React Flow-based interactive editor as a premium feature, allowing users to modify AI-generated diagrams.
  - **Infographics:** Integrate Recharts to provide more polished, dashboard-like outputs. Begin development of a D3.js-based module to create a unique, signature infographic type that sets "old.new" apart from competitors.

By following this strategic roadmap, "old.new" can effectively leverage the power of the open-source community to build a feature-rich, scalable, and market-leading AI generation platform.

Works cited

1. jsonresume/jsonresume.org: The mono repo that builds the homepage, utils, ui components, registry and anything else - GitHub, accessed on October 6, 2025, https://github.com/jsonresume/jsonresume.org
2. mermaid-js/mermaid: Generation of diagrams like flowcharts or sequence diagrams from text in a similar manner as markdown - GitHub, accessed on October 6, 2025, https://github.com/mermaid-js/mermaid

3. reactflow · GitHub Topics · GitHub, accessed on October 6, 2025, https://github.com/topics/reactflow
4. chartjs/awesome: A curated list of awesome Chart.js ... - GitHub, accessed on October 6, 2025, https://github.com/chartjs/awesome
5. recharts · GitHub Topics · GitHub, accessed on October 6, 2025, https://github.com/topics/recharts?o=asc&s=updated
6. wbkd/awesome-d3: A list of D3 libraries, plugins and utilities - GitHub, accessed on October 6, 2025, https://github.com/wbkd/awesome-d3
7. JSON Resume - Brian Douglass, accessed on October 6, 2025, https://bhdouglass.com/blog/json-resume/
8. theme path jsonresume-theme-even could not be resolved from current working directory #626 - GitHub, accessed on October 6, 2025, https://github.com/jsonresume/resume-cli/issues/626
9. hacksalot/HackMyResume: Generate polished résumés and CVs in HTML, Markdown, LaTeX, MS Word, PDF, plain text, JSON, XML, YAML, smoke signal, and carrier pigeon. - GitHub, accessed on October 6, 2025, https://github.com/hacksalot/HackMyResume
10. JSON Resume Themes - Discover gists · GitHub, accessed on October 6, 2025, https://gist.github.com/asbjornu/7873be2713fcacc911be2035a482091d
11. erming/jsonresume-theme-flat: A minimalistic theme for http://jsonresume.org - GitHub, accessed on October 6, 2025, https://github.com/erming/jsonresume-theme-flat
12. mudassir0909/jsonresume-theme-elegant - GitHub, accessed on October 6, 2025, https://github.com/mudassir0909/jsonresume-theme-elegant
13. vkcelik/jsonresume-theme-onepage-plus: A compact theme for JSON Resume, designed for printing. - GitHub, accessed on October 6, 2025, https://github.com/vkcelik/jsonresume-theme-onepage-plus
14. rbardini/jsonresume-theme-even: A flat JSON Resume theme, compatible with the latest resume schema - GitHub, accessed on October 6, 2025, https://github.com/rbardini/jsonresume-theme-even
15. kelyvin/jsonresume-theme-caffeine: Caffeine theme for the JSON Resume project - GitHub, accessed on October 6, 2025, https://github.com/kelyvin/jsonresume-theme-caffeine
16. bluesialia/jsonresume-theme-bluetime 0.4.2 on npm - Libraries.io, accessed on October 6, 2025, https://libraries.io/npm/@bluesialia%2Fjsonresume-theme-bluetime
17. LinuxBozo/jsonresume-theme-kendall - GitHub, accessed on October 6, 2025, https://github.com/LinuxBozo/jsonresume-theme-kendall
18. Cannot use themes through CLI · Issue #408 · jsonresume/resume-cli - GitHub, accessed on October 6, 2025, https://github.com/jsonresume/resume-cli/issues/408
19. XuluWarrior/jsonresume-theme-kards - GitHub, accessed on October 6, 2025, https://github.com/XuluWarrior/jsonresume-theme-kards
20. XuluWarrior/jsonresume-theme-orbit - GitHub, accessed on October 6, 2025, https://github.com/XuluWarrior/jsonresume-theme-orbit

21. anthonyjdella/customized-jsonresume-theme-stackoverflow: I tweaked the 'Stack Overflow' theme for JSON Resume - GitHub, accessed on October 6, 2025, https://github.com/anthonyjdella/customized-jsonresume-theme-stackoverflow
22. Azim-Ahmed/Automation-workflow: React flow Examples with Workflow automations and others examples in one repo. - GitHub, accessed on October 6, 2025, https://github.com/Azim-Ahmed/Automation-workflow
23. xyflow/react-flow-example-apps - GitHub, accessed on October 6, 2025, https://github.com/xyflow/react-flow-example-apps
24. jsplumb/jsplumb: Visual connectivity for webapps - GitHub, accessed on October 6, 2025, https://github.com/jsplumb/jsplumb
25. SuperViz provides powerful SDKs and APIs that enable developers to easily integrate real-time features into web applications. Our platform accelerates development across various industries with robust, scalable infrastructure and a low-code approach. - GitHub, accessed on October 6, 2025, https://github.com/SuperViz/superviz
26. SuperViz - GitHub, accessed on October 6, 2025, https://github.com/superviz
27. mermaidjs/mermaidjs.github.io: Documentation has been moved to docs in https://github.com/mermaid-js/mermaid - GitHub, accessed on October 6, 2025, https://github.com/mermaidjs/mermaidjs.github.io
28. Edit, preview and share mermaid charts/diagrams. New implementation of the live editor. - GitHub, accessed on October 6, 2025, https://github.com/mermaid-js/mermaid-live-editor
29. erikbrinkman/d3-dag: Layout algorithms for visualizing directed acyclic graphs - GitHub, accessed on October 6, 2025, https://github.com/erikbrinkman/d3-dag
30. dagrejs/dagre-d3: A D3-based renderer for Dagre - GitHub, accessed on October 6, 2025, https://github.com/dagrejs/dagre-d3
31. forivall/react-d3-dag: :tokyo_tower: React component to create interactive D3 directed acyclic graphs - GitHub, accessed on October 6, 2025, https://github.com/forivall/react-d3-dag
32. Some quick chart.js examples - GitHub, accessed on October 6, 2025, https://github.com/sjmf/chartjs-examples
33. chartjs · GitHub Topics, accessed on October 6, 2025, https://github.com/topics/chartjs
34. Data Visualisation in React — Part I: An Introduction to Recharts | by Jack Rhodes - Medium, accessed on October 6, 2025, https://medium.com/swlh/data-visualisation-in-react-part-i-an-introduction-to-recharts-33249e504f50
35. naver/billboard.js: Re-usable, easy interface JavaScript chart library based on D3.js, accessed on October 6, 2025, https://github.com/naver/billboard.js/
36. Britecharts - GitHub, accessed on October 6, 2025, https://github.com/britecharts
37. sgratzl/chartjs-chart-boxplot: Chart.js Box Plots and Violin Plot Charts - GitHub, accessed on October 6, 2025, https://github.com/sgratzl/chartjs-chart-boxplot
38. Releases · sgratzl/chartjs-chart-geo - GitHub, accessed on October 6, 2025,

https://github.com/sgratzl/chartjs-chart-geo/releases
39. sgratzl/chartjs-chart-geo: Chart.js Choropleth and Bubble Maps - GitHub, accessed on October 6, 2025, https://github.com/sgratzl/chartjs-chart-geo
40. kurkle/chartjs-chart-matrix: Chart.js module for creating matrix charts - GitHub, accessed on October 6, 2025, https://github.com/kurkle/chartjs-chart-matrix
41. kurkle/chartjs-chart-sankey: Chart.js module for creating sankey diagrams - GitHub, accessed on October 6, 2025, https://github.com/kurkle/chartjs-chart-sankey
42. Releases · kurkle/chartjs-chart-sankey - GitHub, accessed on October 6, 2025, https://github.com/kurkle/chartjs-chart-sankey/releases
43. sgratzl/chartjs-chart-funnel: Chart.js Funnel chart - GitHub, accessed on October 6, 2025, https://github.com/sgratzl/chartjs-chart-funnel
44. sgratzl/chartjs-chart-graph: Chart.js Graph-like Charts (tree, force directed) - GitHub, accessed on October 6, 2025, https://github.com/sgratzl/chartjs-chart-graph
45. chartjs/chartjs-plugin-annotation: Annotation plugin for Chart.js - GitHub, accessed on October 6, 2025, https://github.com/chartjs/chartjs-plugin-annotation
46. Releases · chartjs/chartjs-plugin-annotation - GitHub, accessed on October 6, 2025, https://github.com/chartjs/chartjs-plugin-annotation/releases
47. chartjs/chartjs-plugin-datalabels: Chart.js plugin to display labels on data elements - GitHub, accessed on October 6, 2025, https://github.com/chartjs/chartjs-plugin-datalabels
48. Releases · chartjs/chartjs-plugin-datalabels - GitHub, accessed on October 6, 2025, https://github.com/chartjs/chartjs-plugin-datalabels/releases