# The Modern Developer's Toolkit: A Curated Analysis of Open-Source Templates for Resumes, Diagrams, and Visualizations

## Executive Summary

This report presents an exhaustive analysis of over 600 open-source GitHub repositories, meticulously curated to identify high-quality, modern, and customizable templates, themes, and examples across three critical domains for developers: Resumes and Professional Documents, Flowcharts and Diagramming, and Infographics and Data Visualization. The investigation reveals a fundamental bifurcation in tooling philosophy that informs strategic selection: declarative, data-driven systems that prioritize automation and consistency, versus imperative, component-based frameworks that offer granular control and bespoke interactivity.

Key trends have emerged from this research. The "Everything as Code" movement is maturing, with paradigms like "Resume as Code" and "Diagrams as Code" treating professional and technical documentation as versionable, testable software artifacts. This is exemplified by the robust ecosystems of JSON Resume and Mermaid.js. Furthermore, the value of mature libraries such as Chart.js and Excalidraw is increasingly defined not by their core functionality alone, but by their vast ecosystems of community-contributed plugins, themes, and component libraries. A notable strategic evolution is the move toward framework agnosticism, as seen in the rebranding of React Flow to Xyflow, signaling a focus on long-term viability beyond the lifecycle of any single front-end framework. Finally, the integration of Artificial Intelligence for content generation and suggestion, particularly within the JSON Resume ecosystem, represents the next frontier for these data-driven tools.

Based on these findings, the following strategic recommendations are proposed. For projects requiring rapid deployment, consistency across multiple formats, and straightforward maintenance, the adoption of data-driven systems like JSON Resume with its thematic ecosystem or Mermaid.js for embedded documentation is advised. For applications demanding highly interactive, custom user experiences—such as bespoke diagram editors or unique data narratives—investing in component-based frameworks like React Flow (Xyflow) or the foundational D3.js library is the superior approach. For the common requirement of standard business charting within web applications, component libraries such as Chart.js and Recharts offer the most effective balance of development speed and customization potential.

## Analysis of Resumes and Professional Document Generators

The modern technical resume has evolved beyond a static document. It must now serve multiple purposes: to be machine-readable for Applicant Tracking Systems (ATS), easily versioned for different roles, and stylistically adaptable to reflect both personal brand and company culture. This section analyzes three dominant paradigms in open-source resume generation: the structured data ecosystem of JSON Resume, which separates content from presentation; the multi-format publishing power of command-line tools like HackMyResume; and the design-centric approach of standalone HTML, CSS, and Tailwind CSS templates.

### The JSON Resume Ecosystem: Data Portability and Thematic Versatility

The JSON Resume project champions a fundamental principle: the separation of resume *content* from its *presentation*.1 This is achieved through a standardized

resume.json file, the structure of which is formally defined and validated by the official resume-schema.3 This architectural decision transforms a resume from a mere document into a portable, machine-readable dataset. By decoupling the data from the view layer, developers can maintain a single source of truth for their career information while rendering it in countless visual formats.

The ecosystem, which has been largely consolidated into the jsonresume/jsonresume.org monorepo, provides a comprehensive suite of tools to manage the entire resume lifecycle.1 The

resume-cli is the primary command-line tool used to validate a resume.json file against the schema and render it into HTML or PDF using a specified theme.3 The JSON Resume Registry complements the CLI by offering a platform to host and share resumes online with unique, persistent URLs, further enhancing their accessibility.1

The true power of the JSON Resume standard is realized through its vast and diverse library of themes. With over 400 community-built themes available, a single resume.json file can be adapted to any professional context without altering the underlying data.6 The official monorepo includes a set of well-maintained themes such as

flat, onepage, tailwind, and the distinctive stackoverflow, providing reliable and stylistically varied starting points.1 Community-driven projects demonstrate further creative potential; for instance,

mylesj/jsonresume-theme-stackoverflowed offers a faithful replica of the discontinued Stack Overflow Developer Story format, a UI pattern that has proven effective with recruiters.7 Other popular themes like

biosan/jsonresume-theme-macchiato and kelyvin/jsonresume-theme-caffeine provide elegant, minimalist designs with enhanced development environments for easier customization.8

The structured, machine-readable format of resume.json makes it a perfect input for Large Language Models (LLMs). Unlike parsing an unstructured PDF, an AI can directly ingest the JSON data, understanding the semantic meaning of each field (e.g., work, skills, projects). This enables highly contextual and accurate AI-driven assistance. The JSON Resume monorepo explicitly details a roadmap for such AI-powered features, including job recommendations, automated cover letter generation, resume suggestions, and even an interview simulator.1 This positions the JSON Resume ecosystem not merely as a resume-building tool, but as a comprehensive, open-source career management platform. It has the potential to offer developers powerful, privacy-respecting tools to optimize their job search, all powered by their own version-controlled, structured data, representing a significant paradigm shift from static document creation to dynamic career strategy.

A developer can maintain their entire career history in a single resume.json file under Git version control. For each job application, they can select a theme that best suits the company's culture—for example, stackoverflowed for a developer-centric role or professional for a corporate one—and generate a tailored PDF in seconds using the CLI. This workflow embodies the "resume as code" philosophy, bringing the benefits of software development practices to career management.

### Multi-Format Publishing with HackMyResume

HackMyResume stands out as a command-line "Swiss Army knife" for resume generation, designed to create a multitude of formats from a single source of truth.10 It ingests a JSON or YAML resume file and can output polished documents in HTML, MS Word, PDF, LaTeX, Markdown, and plain text, among others.10 This capability is invaluable for professionals who need to adapt their resume to different submission requirements without maintaining separate documents.

The tool is notable for its support of two competing data standards: the JSON Resume (JRS) standard and its own Fluent Resume Schema (FRESH).10 HackMyResume acts as a bridge between these two ecosystems, providing a

convert command that allows users to seamlessly translate their resume data from one format to the other.10 This interoperability ensures that users are not locked into a single standard and can leverage tools and themes from both communities.

Theming is a core feature of HackMyResume. It comes pre-installed with a set of canonical FRESH themes, such as positive, modern, and awesome (based on the popular Awesome-CV LaTeX template), providing a range of professional and technical styles out of the box.10 Furthermore, it can install and use any JSON Resume theme available on NPM or GitHub, giving users access to the hundreds of themes developed by that community.10 Advanced features enhance its power, including the ability to merge multiple resume files—for instance, combining a generic base resume with a targeted, job-specific version—and robust schema validation and keyword analysis tools.10 PDF generation is handled by external engines like

wkhtmltopdf or WeasyPrint, offering flexibility in the rendering pipeline.10

For a practical use case, consider a consultant who must provide their CV in various formats for different engagements. For a corporate client, a formal DOCX document might be required; for their personal website, a clean HTML version is ideal; and for an academic proposal, a professionally typeset LaTeX PDF is preferred. Using HackMyResume, the consultant can maintain one canonical resume.json file and generate all required outputs with a single, scriptable command, ensuring consistency and saving significant time.

### Curated HTML, CSS, and Tailwind CSS Templates

This category comprises standalone repositories that offer a pre-designed, visually driven approach to resume creation. Unlike the data-driven systems previously discussed, here the content and presentation are tightly coupled within HTML files. The primary value of these templates lies in their aesthetic quality, responsiveness, and the ease with which a developer can directly manipulate the code to achieve a custom result.

The most modern and effective templates in this category leverage utility-first CSS frameworks like Tailwind CSS. This approach provides a high degree of customizability and maintainability without the need to write extensive custom CSS stylesheets.14 Repositories such as

devxprite/resumave exemplify this trend, offering a complete resume builder as a Next.js application styled with Tailwind CSS.14 Similarly,

WebPraktikos/universal-resume focuses on creating a minimal, formal template that is optimized for print, mobile, and desktop viewing, using Tailwind for its layout and styling.16

A critical feature of these modern templates is their attention to both digital and physical output formats. Many explicitly market themselves as "ATS-Friendly," indicating a focus on clean, semantic HTML that can be reliably parsed by automated recruitment systems.14 Another key innovation is the sophisticated use of print-specific styles. By leveraging CSS media queries or Tailwind's

print: modifier, templates like owengretzinger/html-resume-template can dramatically alter their layout for printing, collapsing columns, hiding non-essential elements like buttons, and ensuring the content fits perfectly on a standard A4 or Letter page.15 This allows for the generation of a high-quality PDF directly from the browser's print dialog, obviating the need for external PDF generation libraries.

For a front-end developer aiming to create a visually striking online portfolio that doubles as their resume, these templates are an excellent choice. By cloning a repository like ibelick/nextjs-resume, which is inspired by the clean, modern aesthetic of platforms like read.cv, a developer can quickly customize the content and styling within a familiar React and Tailwind CSS codebase and deploy a professional, polished personal website.18

| Paradigm | Core Principle | Primary Use Case | Customization Method | Maintenance Effort | ATS-Friendliness |
| --- | --- | --- | --- | --- | --- |
| **JSON Resume Ecosystem** | Separation of Content and Presentation | Maintaining a single, version-controlled data source for a resume to be rendered in many different styles. | Selecting from hundreds of pre-built themes or creating a new theme (Handlebars/CSS). | Low (update one JSON file). | High (structured data). |
| **HackMyResume CLI** | Single-Source, Multi-Format Publishing | Generating multiple document formats (PDF, DOCX, HTML, LaTeX) from one resume data file for diverse submission requirements. | Selecting themes from FRESH or JSON Resume standards; overriding styles. | Low (update one JSON/YAML file). | High (structured data). |
| **Standalone HTML/Tailwind Templates** | Design-First, Coupled Content and Presentation | Creating a visually unique and highly customized online resume or personal portfolio website. | Directly editing HTML, CSS, and JavaScript/TypeScript code. | Medium (manual edits to code). | Variable (depends on template's semantic structure). |

## Analysis of Flowchart and Diagramming Repositories

In the realm of software development, technical documentation, system design, and process visualization are indispensable. This section explores the open-source tools that facilitate diagram creation, drawing a distinction between the "Diagrams as Code" approach, which prioritizes version control and automation, and interactive, canvas-based tools that offer a dynamic, WYSIWYG (What You See Is What You Get) experience. We will analyze Mermaid.js for text-to-diagram generation, Excalidraw for collaborative whiteboarding, and React Flow/Xyflow for building powerful, node-based applications.

### Mermaid.js: Diagrams as Code

Mermaid.js is a JavaScript-based library that generates a wide variety of diagrams—including flowcharts, sequence diagrams, class diagrams, and Gantt charts—from a simple, Markdown-inspired text syntax.20 This "Diagrams as Code" approach allows complex visualizations to be written and edited in any text editor, embedded directly within documentation files like

README.md, and version-controlled using Git alongside the source code they describe. This paradigm helps solve the problem of "Doc-Rot," where diagrams quickly become outdated because they are difficult to modify.20 The widespread adoption of this approach is evidenced by GitHub's native support for rendering Mermaid syntax blocks directly in Markdown files.22

The primary repository, mermaid-js/mermaid, contains the core library, extensive documentation, and a live editor for experimentation.20 The community has built upon this foundation, creating repositories such as

rudolfolah/mermaid-diagram-examples and numerous Gists that serve as valuable resources, offering a wealth of practical snippets for common use cases like binary trees, dependency sets, and complex flowcharts.22

Customization in Mermaid is primarily achieved through directives within the diagram's text definition. Users can specify themes (default, dark, forest, neutral), apply variables for fine-tuning colors, and even alter the rendering style to appear handDrawn.25 This allows for aesthetic control without sacrificing the simplicity of the text-based format.

For a practical application, a DevOps team can document their entire CI/CD pipeline using a sequence diagram written in Mermaid syntax within their project's README.md. When a step in the pipeline is added or modified, a developer simply updates the corresponding line of text in the Mermaid block. The diagram is automatically re-rendered to reflect the new process, ensuring that the documentation remains synchronized with the implementation.

### Excalidraw: The Virtual Whiteboard Paradigm

Excalidraw offers a virtual whiteboard for sketching diagrams with a distinctive, hand-drawn feel, making it ideal for collaborative brainstorming, wireframing, and informal system design.27 While the public website,

excalidraw.com, provides a full-featured, end-to-end encrypted application, the core value for developers is the @excalidraw/excalidraw npm package. This package allows the entire Excalidraw editor to be embedded as a React component within any web application, providing a powerful foundation for building custom visual tools.27

The true power of Excalidraw for creating templated and reusable diagrams lies in its ecosystem of libraries. The excalidraw/excalidraw-libraries repository is a curated collection of publicly available component libraries—packaged as .excalidrawlib files—that contain reusable shapes, icons, and stencils for a wide array of domains.28 These libraries cover everything from high-level system design components and specific cloud provider icons (AWS, GCP, Azure) to UX wireframing elements, DevOps symbols, and database schemas.29 This allows users to quickly assemble complex diagrams by dragging and dropping standardized components onto the canvas.

A key feature of the Excalidraw ecosystem is its open and extensible nature. Users can create and publish their own libraries directly from the Excalidraw interface, enabling teams to develop a shared visual language and a repository of custom components tailored to their specific products or architectural patterns.28

As a use case, a solutions architect preparing for a system design interview can leverage a community-provided library like aretecode/system-design-templates-excalidraw.31 During a live whiteboarding session, they can rapidly drag and drop common architectural components such as load balancers, databases, caches, and microservices onto the canvas. This accelerates the diagramming process, allowing them to focus on explaining the architectural trade-offs rather than on the mechanics of drawing.

### React Flow and Xyflow: Building Node-Based Applications

React Flow, now part of the broader Xyflow organization, is a highly extensible React library for building interactive, node-based UIs, editors, and diagrams.32 It provides the foundational components—such as the main

<ReactFlow> canvas, nodes, and edges—and the underlying logic for creating custom flow builders, workflow editors, and other visual programming environments.

The ecosystem is rich with resources to accelerate development. Starter templates like xyflow/vite-react-flow-template provide a pre-configured project to quickly bootstrap a new application with React Flow, Vite, and TypeScript.34 The official "Pro Examples" serve as advanced templates, showcasing sophisticated implementations for features such as automatic graph layouting (using libraries like Dagre and Elk.js), real-time collaboration (integrated with yjs), undo/redo functionality, and a variety of custom node and edge shapes.35 These examples are not just demos but downloadable Vite applications that can be used as a foundation for building complex, feature-rich tools.

The evolution of React Flow into the Xyflow organization is a significant development. This strategic rebranding reflects a technical and business maturation, moving from a React-specific library to a framework-agnostic ecosystem. The core logic for managing a graph—its nodes, edges, positions, and connections—is fundamentally data, independent of any specific rendering framework. By abstracting this logic into a shared system package (@xyflow/system), the maintainers have been able to create separate, lightweight renderer packages for different frameworks, including @xyflow/react for React and @xyflow/svelte for Svelte.36 This move not only expands the potential user base to other framework communities but also reduces maintenance overhead by centralizing the core graph management logic. For a solutions architect or development lead, this signals project maturity, a long-term vision, and enhanced stability, making Xyflow a more robust and safer choice for mission-critical, long-term projects compared to a library tightly coupled to a single framework's ecosystem.

For a practical application, a company building an internal workflow automation tool can use React Flow as the core of its visual editor. They can develop custom nodes that represent specific business actions (e.g., "Send Email," "Query Database," "Generate Report") and allow non-technical users to visually connect these nodes to build, execute, and monitor complex business processes. The xyflow/awesome-node-based-uis repository provides a curated list of real-world applications built with this paradigm, offering a wealth of inspiration for what is possible.33

## Analysis of Infographics and Data Visualization Libraries

Data visualization is the critical practice of transforming raw data into comprehensible and actionable insights. The open-source landscape provides a rich spectrum of tools for this purpose, ranging from low-level, imperative libraries like D3.js, which offer unparalleled creative freedom for bespoke visualizations, to high-level, component-based libraries such as Chart.js and Recharts, which prioritize development speed and ease of use for creating standard chart types. This section provides a comparative analysis of these key players and their respective ecosystems, guiding the selection of the appropriate tool based on project requirements.

### The D3.js Universe: The Foundation of Bespoke Visualization

D3.js (Data-Driven Documents) is not, in the strictest sense, a charting library. It is a low-level, foundational JavaScript library for manipulating documents based on data.39 It provides a powerful and expressive toolkit for binding arbitrary data to a Document Object Model (DOM), and then applying data-driven transformations to the document. Its core strengths lie in its robust data utilities, its approach to selections and transitions, and its comprehensive set of graphical primitives for scales, axes, and shapes.40 The library's defining characteristic is its "unparalleled flexibility," which empowers developers to create novel, dynamic, and highly interactive graphics that go far beyond conventional charting.39

The best resources for understanding the capabilities of D3.js are the extensive example galleries. The official D3 Gallery, hosted on Observable, showcases a vast range of possibilities, including animated treemaps, force-directed graphs, complex geographic maps with various projections, and hierarchical edge bundling.42 Curated lists such as

wbkd/awesome-d3 further expand on this, cataloging a rich ecosystem of plugins, utilities, and third-party libraries built on top of D3.43 These resources demonstrate that with D3, the only limit is the developer's imagination. Highly specialized libraries, such as

ofrohn/d3-celestial for creating interactive star maps, are a testament to the power of D3 as a platform for building domain-specific visualization tools.44

Customization in D3 is both its greatest strength and its steepest learning curve. It is an imperative process where the developer is responsible for constructing every aspect of the visualization step-by-step: loading and processing data, creating scales to map data domains to visual ranges, appending SVG or Canvas elements to the DOM, and defining the logic for interactions and animated transitions.

A practical use case for D3.js would be a data journalism team aiming to create a unique, interactive, and animated data narrative, such as the widely recognized "Bar Chart Race" visualization.42 This type of visualization, which involves complex data transformations, object constancy across animated frames, and precise control over transitions, is difficult to achieve with high-level charting libraries. D3.js provides the low-level control necessary to manage these intricate data-to-visual mappings and produce a polished, bespoke result.

### Component-Driven Charting: Chart.js and Recharts

In contrast to the low-level approach of D3.js, libraries like Chart.js and Recharts offer a high-level, component-based abstraction for creating the most common types of charts (e.g., line, bar, pie, radar). They prioritize developer productivity and rapid implementation by providing pre-built, composable components that abstract away the complexities of manual DOM manipulation and rendering.

**Chart.js** is a framework-agnostic library that renders charts to an HTML5 <canvas> element.45 This choice of rendering engine is a key technical differentiator. The Canvas API is a low-level, procedural API for drawing graphics as a single bitmap image. This approach is extremely performant when rendering a large number of elements—such as thousands of data points in a scatterplot—because the browser only needs to manage a single DOM element. However, because the individual graphical elements (e.g., a single bar in a bar chart) are not part of the DOM, implementing interactivity like tooltips or click events on specific elements requires more complex logic, often involving hit detection on the canvas. Chart.js has a rich ecosystem of plugins for extending its functionality, with

chartjs/awesome serving as the central hub for resources like annotations, zoom controls, and custom data labels.46 Its framework-agnostic nature is supported by a wide range of wrappers for popular frameworks, such as

react-chartjs-2 for React and vue-chartjs for Vue.48

**Recharts**, on the other hand, is built specifically for the React ecosystem and renders charts as Scalable Vector Graphics (SVG) elements, using D3.js for its underlying data and layout calculations.50 Its API is highly declarative and component-based, allowing developers to compose charts using familiar JSX syntax (e.g.,

<LineChart>, <XAxis>, <Tooltip>). This makes it feel very natural and intuitive for React developers.50 The use of SVG means that every element of the chart—every line, circle, or bar—is an individual DOM node. This makes it straightforward to apply CSS for styling and to attach event listeners for interactivity. However, this approach can lead to performance degradation when rendering a very large number of DOM nodes, as the browser's rendering engine must manage a complex scene graph.

The choice between Chart.js and Recharts is therefore not just a matter of API preference but a strategic technical trade-off. For dashboards that need to display a high density of data points where rendering performance is the primary concern, the canvas-based approach of Chart.js may be superior. For applications that require rich interactivity, complex tooltips, and fine-grained styling of individual chart elements, the SVG-based, DOM-centric approach of Recharts is often the more suitable choice.

For a developer building a standard business intelligence dashboard, either library offers a rapid path to implementation. They can quickly add several standard, responsive charts (e.g., sales over time, revenue by region) by composing a few components and passing data as props, achieving a polished and interactive result in a fraction of the time it would take with D3.js.

### Specialized SVG Visualization Repositories

Beyond the foundational and general-purpose charting libraries, the open-source community provides a wealth of specialized visualization tools that cater to specific needs and aesthetics.

plouc/nivo stands out as a comprehensive suite of React components for data visualization, built on top of D3.js.51 It occupies a valuable middle ground between the low-level control of D3 and the simplicity of Recharts. Nivo provides a rich set of highly polished and configurable components for both standard and complex chart types, including heatmaps, choropleth maps, network graphs, and Sankey diagrams. Its server-side rendering capability also makes it an excellent choice for generating static charts in Node.js environments.

For more niche aesthetic requirements, jwilber/roughViz offers a unique solution.54 This library leverages

rough.js to create charts with a sketchy, hand-drawn appearance. This style is particularly effective for visualizations where the goal is to convey a sense of generality, imprecision, or a work-in-progress feel, distinguishing them from the sharp, precise lines of traditional charting libraries.

For specific use cases like timelines and Gantt charts, while libraries like Mermaid.js can create static versions 20, more interactive and feature-rich components can be found in specialized repositories.

sbstjn/timesheet.js, for example, is a dedicated JavaScript library for creating HTML5 and CSS3 time sheets, offering a focused solution for this common visualization task.54

| Library | Primary Abstraction | Rendering Engine | Key Strength | Learning Curve | Ideal Use Case |
| --- | --- | --- | --- | --- | --- |
| **D3.js** | Low-level data binding to DOM elements | SVG / Canvas | Unparalleled flexibility for bespoke, novel, and interactive visualizations. | High | Data journalism, scientific visualization, creating new types of charts. |
| **Chart.js** | High-level chart components | Canvas | High performance with large datasets, framework-agnostic, extensive plugin ecosystem. | Low | Dashboards with thousands of data points, applications where rendering speed is critical. |
| **Recharts** | Declarative React components | SVG | Seamless integration with React, easy interactivity and styling of individual elements. | Low | Standard business dashboards within the React ecosystem requiring rich tooltips and interactivity. |
| **Nivo** | Declarative React components for complex charts | SVG / Canvas / HTML | Provides a wide range of polished, highly configurable chart types out-of-the-box. | Medium | Complex but standard visualizations like heatmaps, network graphs, and choropleths in React. |

## Comprehensive Repository Index

The following table provides a comprehensive index of the key repositories analyzed in this report. It includes essential metadata to facilitate quick reference and comparison.

| Repository (owner/name) | URL | Stars | Forks | License | Last Pushed | Primary Language(s) | Category | Sub-category |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| devxprite/resumave | <https://github.com/devxprite/resumave> | 15+ | 76+ | MIT | >2023 | TypeScript, JavaScript, CSS | Resumes/Documents | HTML/CSS/Tailwind Template |
| owengretzinger/html-resume-template | <https://github.com/owengretzinger/html-resume-template> | 2 | 1 | Unlicense | >2023 | CSS, HTML, JavaScript | Resumes/Documents | HTML/CSS/Tailwind Template |
| WebPraktikos/universal-resume | <https://github.com/WebPraktikos/universal-resume> | 15+ | 8+ | MIT | >2023 | JavaScript, CSS | Resumes/Documents | HTML/CSS/Tailwind Template |
| jsonresume/jsonresume.org | <https://github.com/jsonresume/jsonresume.org> | 168 | 42 | MIT | >2023 | JavaScript, TypeScript, CSS | Resumes/Documents | JSON Resume Ecosystem |
| jsonresume/resume-schema | <https://github.com/jsonresume/resume-schema> | 2,300 | 290 | MIT | >2023 | JavaScript | Resumes/Documents | JSON Resume Ecosystem |
| jsonresume/resume-cli | <https://github.com/jsonresume/resume-cli> | 4,700 | 395 | MIT | >2023 | JavaScript | Resumes/Documents | JSON Resume Ecosystem |
| mylesj/jsonresume-theme-stackoverflowed | <https://github.com/mylesj/jsonresume-theme-stackoverflowed> | 4 | 0 | MIT | >2023 | TypeScript | Resumes/Documents | JSON Resume Theme |
| biosan/jsonresume-theme-macchiato | <https://github.com/biosan/jsonresume-theme-macchiato> | 17 | 53 | MIT | >2023 | Handlebars, CSS, JavaScript | Resumes/Documents | JSON Resume Theme |
| kelyvin/jsonresume-theme-caffeine | <https://github.com/kelyvin/jsonresume-theme-caffeine> | 114 | 86 | MIT | >2023 | Handlebars, SCSS, JavaScript | Resumes/Documents | JSON Resume Theme |
| hacksalot/HackMyResume | <https://github.com/hacksalot/HackMyResume> | 9,300 | 530 | MIT | >2023 | JavaScript, HTML, CSS | Resumes/Documents | HackMyResume CLI |
| fresh-standard/fresh-themes | <https://github.com/fresh-standard/fresh-themes> | 196 | 62 | MIT | >2023 | N/A | Resumes/Documents | HackMyResume Theme |
| mermaid-js/mermaid | <https://github.com/mermaid-js/mermaid> | 83,300 | 8,100 | MIT | >2023 | TypeScript, JavaScript, HTML | Flowcharts/Diagrams | Mermaid.js |
| rudolfolah/mermaid-diagram-examples | <https://github.com/rudolfolah/mermaid-diagram-examples> | 9 | 3 | MIT | <2023 | N/A | Flowcharts/Diagrams | Mermaid.js Example |
| excalidraw/excalidraw | <https://github.com/excalidraw/excalidraw> | 108,000 | 11,100 | MIT | >2023 | TypeScript, SCSS, MDX | Flowcharts/Diagrams | Excalidraw |
| excalidraw/excalidraw-libraries | <https://github.com/excalidraw/excalidraw-libraries> | 1,000 | 264 | MIT | >2023 | JavaScript, HTML, Python | Flowcharts/Diagrams | Excalidraw Library |
| xyflow/xyflow | <https://github.com/xyflow/xyflow> | 31,825 | 2,086 | MIT | >2023 | TypeScript | Flowcharts/Diagrams | React Flow/Xyflow |
| xyflow/vite-react-flow-template | <https://github.com/xyflow/vite-react-flow-template> | 81 | 35 | MIT | >2023 | TypeScript, JavaScript, HTML | Flowcharts/Diagrams | React Flow/Xyflow Template |
| xyflow/awesome-node-based-uis | <https://github.com/xyflow/awesome-node-based-uis> | 3,225 | 168 | CC0-1.0 | >2023 | N/A | Flowcharts/Diagrams | React Flow/Xyflow Resource |
| d3/d3 | <https://github.com/d3/d3> | 112,000 | 22,800 | ISC | >2023 | Shell, JavaScript | Infographics | D3.js |
| wbkd/awesome-d3 | <https://github.com/wbkd/awesome-d3> | N/A | N/A | N/A | >2023 | N/A | Infographics | D3.js Resource |
| ofrohn/d3-celestial | <https://github.com/ofrohn/d3-celestial> | 684 | 191 | BSD-3-Clause | >2023 | N/A | Infographics | D3.js Example |
| chartjs/Chart.js | <https://github.com/chartjs/Chart.js> | 66,600 | 12,000 | MIT | >2023 | JavaScript, TypeScript | Infographics | Chart.js |
| chartjs/awesome | <https://github.com/chartjs/awesome> | 2,400 | 143 | MIT | >2023 | N/A | Infographics | Chart.js Resource |
| reactchartjs/react-chartjs-2 | <https://github.com/reactchartjs/react-chartjs-2> | 6,900 | 2,400 | MIT | >2023 | TypeScript, MDX, JavaScript | Infographics | Chart.js Wrapper |
| recharts/recharts | <https://github.com/recharts/recharts> | 26,000 | 1,800 | MIT | >2023 | TypeScript, MDX, SCSS | Infographics | Recharts |
| plouc/nivo | <https://github.com/plouc/nivo> | 13,800 | N/A | N/A | >2023 | TypeScript | Infographics | SVG Visualization |
| jwilber/roughViz | <https://github.com/jwilber/roughViz> | 7,000 | 235 | MIT | >2023 | JavaScript | Infographics | SVG Visualization |

#### Works cited

1. jsonresume/jsonresume.org: The mono repo that builds the ... - GitHub, accessed on October 7, 2025, <https://github.com/jsonresume/jsonresume.org>
2. JSON Resume, accessed on October 7, 2025, <https://jsonresume.org/>
3. JSON Resume - GitHub, accessed on October 7, 2025, <https://github.com/jsonresume>
4. jsonresume/resume-schema: JSON-Schema is used here ... - GitHub, accessed on October 7, 2025, <https://github.com/jsonresume/resume-schema>
5. jsonresume/resume-cli: CLI tool to easily setup a new resume - GitHub, accessed on October 7, 2025, <https://github.com/jsonresume/resume-cli>
6. Themes - JSON Resume, accessed on October 7, 2025, <https://jsonresume.org/themes>
7. mylesj/jsonresume-theme-stackoverflowed: A JSON ... - GitHub, accessed on October 7, 2025, <https://github.com/mylesj/jsonresume-theme-stackoverflowed>
8. biosan/jsonresume-theme-macchiato: Simple ... - GitHub, accessed on October 7, 2025, <https://github.com/biosan/jsonresume-theme-macchiato>
9. kelyvin/jsonresume-theme-caffeine: Caffeine theme for the ... - GitHub, accessed on October 7, 2025, <https://github.com/kelyvin/jsonresume-theme-caffeine>
10. 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 7, 2025, <https://github.com/hacksalot/HackMyResume>
11. HackMyResume download | SourceForge.net, accessed on October 7, 2025, <https://sourceforge.net/projects/hackmyresume.mirror/>
12. epidrome/awesome-cv-templates: :page\_facing\_up: A ... - GitHub, accessed on October 7, 2025, <https://github.com/epidrome/awesome-cv-templates>
13. fresh-standard/fresh-themes: Multiformat themes and skins ... - GitHub, accessed on October 7, 2025, <https://github.com/fresh-standard/fresh-themes>
14. devxprite/resumave: It's an ATS-friendly resume maker ... - GitHub, accessed on October 7, 2025, <https://github.com/devxprite/resumave>
15. owengretzinger/html-resume-template: Create a mobile ... - GitHub, accessed on October 7, 2025, <https://github.com/owengretzinger/html-resume-template>
16. WebPraktikos/universal-resume: Minimal and formal ... - GitHub, accessed on October 7, 2025, <https://github.com/WebPraktikos/universal-resume>
17. toreylittlefield/my-custom-tailwind-resume: My Custom ... - GitHub, accessed on October 7, 2025, <https://github.com/toreylittlefield/my-custom-tailwind-resume>
18. cv-template · GitHub Topics, accessed on October 7, 2025, <https://github.com/topics/cv-template?l=typescript>
19. ibelick/nextjs-resume: A curriculum vitae inspired by read ... - GitHub, accessed on October 7, 2025, <https://github.com/ibelick/nextjs-resume>
20. mermaid-js/mermaid: Generation of diagrams like flowcharts or sequence diagrams from text in a similar manner as markdown - GitHub, accessed on October 7, 2025, <https://github.com/mermaid-js/mermaid>
21. About Mermaid, accessed on October 7, 2025, <https://mermaid.js.org/intro/>
22. Mermaid on Github Examples, accessed on October 7, 2025, <https://gist.github.com/ChristopherA/bffddfdf7b1502215e44cec9fb766dfd>
23. rudolfolah/mermaid-diagram-examples: Examples of ... - GitHub, accessed on October 7, 2025, <https://github.com/rudolfolah/mermaid-diagram-examples>
24. Examples | Mermaid, accessed on October 7, 2025, <https://mermaid.js.org/syntax/examples.html>
25. awesome-panel/panel-mermaid: A Mermaid widget for ... - GitHub, accessed on October 7, 2025, <https://github.com/awesome-panel/panel-mermaid>
26. Gordonby/MermaidTheming: Experimenting with mermaid ... - GitHub, accessed on October 7, 2025, <https://github.com/Gordonby/MermaidTheming>
27. excalidraw/excalidraw: Virtual whiteboard for sketching ... - GitHub, accessed on October 7, 2025, <https://github.com/excalidraw/excalidraw>
28. excalidraw/excalidraw-libraries: Collection of publicly available libraries - GitHub, accessed on October 7, 2025, <https://github.com/excalidraw/excalidraw-libraries>
29. Excalidraw Libraries, accessed on October 7, 2025, <https://libraries.excalidraw.com/>
30. excalidraw · GitHub Topics · GitHub, accessed on October 7, 2025, <https://github.com/topics/excalidraw>
31. aretecode/system-design-templates-excalidraw: Rough ... - GitHub, accessed on October 7, 2025, <https://github.com/aretecode/system-design-templates-excalidraw>
32. React Flow: Node-Based UIs in React, accessed on October 7, 2025, <https://reactflow.dev/>
33. xyflow/awesome-node-based-uis: A curated list with ... - GitHub, accessed on October 7, 2025, <https://github.com/xyflow/awesome-node-based-uis>
34. xyflow/vite-react-flow-template: A vite template to get started ... - GitHub, accessed on October 7, 2025, <https://github.com/xyflow/vite-react-flow-template>
35. React Flow Pro Examples - React Flow, accessed on October 7, 2025, <https://reactflow.dev/pro/examples>
36. xyflow - GitHub, accessed on October 7, 2025, <https://github.com/xyflow>
37. Releases · xyflow/xyflow - GitHub, accessed on October 7, 2025, <https://github.com/xyflow/xyflow/releases>
38. xyflow/web: This monorepo contains the xyflow website and ... - GitHub, accessed on October 7, 2025, <https://github.com/xyflow/web>
39. GitHub - d3/d3: Bring data to life with SVG, Canvas and HTML. :bar\_chart, accessed on October 7, 2025, <https://github.com/d3/d3>
40. D3.js Demo - GitHub Pages, accessed on October 7, 2025, <https://janmilosh.github.io/d3js-presentation/>
41. D3 by Observable | The JavaScript library for bespoke data ..., accessed on October 7, 2025, <https://d3js.org/>
42. d3/d3: Bring data to life with SVG, Canvas and HTML ... - GitHub, accessed on October 7, 2025, <https://github.com/d3/d3/wiki/Gallery>
43. wbkd/awesome-d3: A list of D3 libraries, plugins and utilities - GitHub, accessed on October 7, 2025, <https://github.com/wbkd/awesome-d3>
44. ofrohn/d3-celestial: A star map with d3.js - GitHub, accessed on October 7, 2025, <https://github.com/ofrohn/d3-celestial>
45. chartjs/Chart.js: Simple HTML5 Charts using the https://github.com/chartjs/chart.js/
46. Chart.js - GitHub, accessed on October 7, 2025, <https://github.com/chartjs>
47. chartjs/awesome: A curated list of awesome Chart.js ... - GitHub, accessed on October 7, 2025, <https://github.com/chartjs/awesome>
48. chartjs · GitHub Topics · GitHub, accessed on October 7, 2025, <https://github.com/topics/chartjs>
49. reactchartjs/react-chartjs-2: React components for Chart.js ... - GitHub, accessed on October 7, 2025, <https://github.com/reactchartjs/react-chartjs-2>
50. recharts/recharts: Redefined chart library built with React ... - GitHub, accessed on October 7, 2025, <https://github.com/recharts/recharts>
51. svg · GitHub Topics · GitHub, accessed on October 7, 2025, <https://github.com/topics/svg?l=typescript>
52. dataviz · GitHub Topics · GitHub, accessed on October 7, 2025, <https://github.com/topics/dataviz>
53. plouc/nivo: nivo provides a rich set of dataviz components ... - GitHub, accessed on October 7, 2025, <https://github.com/plouc/nivo>
54. visualization · GitHub Topics · GitHub, accessed on October 7, 2025, <https://github.com/topics/visualization?l=javascript>
55. jwilber/roughViz: Reusable JavaScript library for creating ... - GitHub, accessed on October 7, 2025, <https://github.com/jwilber/roughViz>
56. accessed on January 1, 1970, <https://github.com/sbstjn/timesheet.js>