

Zirui Wen 个人品牌网站:赛博朋克风格 AI 工程师作品集设计与技术实现方案

I. 战略基础:“AI 架构师”叙事

本设计方案的核心战略目标是超越传统的“学生作品集”范畴,将 Zirui Wen 塑造为一名经过验证的、具备端到端系统构建能力的“AI 架构师”。赛博朋克美学在此并非单纯的视觉风格选择,而是一种叙事载体,旨在直观地传达其技术深度、系统思维能力和对前沿科技的掌控力。

定义核心叙事

本网站将通过用户体验讲述一个故事:用户访问网站的过程将模拟与一个复杂、高性能的计算系统进行交互的过程。这种叙事方式直接映射了 Zirui Wen 在其项目中展现出的从底层到应用层的系统构建能力,例如他独立设计的 C++ B+Tree 数据库和 CUDA K-Means 加速器项目¹。其目标是让招聘官在浏览网页时,感觉像是在探索一位系统构建者的思维逻辑,而非仅仅阅读一份履历。这种沉浸式体验旨在从一开始就建立起专业、深刻的第一印象,将访问者从被动的“阅读者”转变为主动的“探索者”。

赛博朋克美学的战略性应用

为了使视觉风格服务于核心叙事,我们将赛博朋克主题解构为与人工智能工程相关的核心概念,确保每一个设计元素都具有明确的象征意义。

- 数据即光影:霓虹色调、发光的线条和粒子流将被用来象征数据在复杂模型中的流动、处理与传输。这将使抽象的数据概念变得具体、可见,并充满动感。
- 计算即力量:故障艺术(Glitch)效果、扫描线纹理和仿终端界面,将象征着他所掌握的

原始计算能力和底层硬件控制技术。这直接关联到他在 CUDA 编程等项目中展现出的高性能计算技能¹。

- 智能即网络:赛博朋克作品中常见的、交织着混乱与秩序的未来城市景观,将被用作神经网络和知识图谱的视觉隐喻。这与他在研究项目中构建 Neo4j 知识图谱的经验直接呼应,展现了他对复杂智能系统结构的理解¹。

目标受众分析:招聘官与技术经理

设计必须在强烈的视觉冲击力与专业信息的清晰传达之间取得精妙平衡。网站的主要受众是技术招聘官和工程经理,他们时间有限,注重结果。因此,设计必须在第一时间高效地传递候选人的核心竞争力,并为他们提供一条清晰、无摩擦的路径来获取关键成就和联系信息。视觉上的“惊艳感”必须服务于、而非阻碍“获得面试”这一核心目标。

网站的构建本身即是一个元项目(meta-project),是候选人能力的活体展示。选择一个视觉复杂、交互丰富且技术要求高的赛博朋克主题,并非仅仅为了装饰。成功实现这样一个高性能、交互式的 WebGL 网站,本身就需要对前端框架(如 React)、3D 图形学(Three.js)、性能优化乃至后端服务有深入的理解和实践²。这恰好与 Zirui Wen 简历中所列举的全栈开发技能(包括 React, Next.js, C++, CUDA, Kubernetes)高度吻合¹。因此,这个作品集网站本身就成为了他的“零号项目”。在招聘官阅读项目描述之前,网站的流畅运行和精妙交互就已经证明了其作为一名全栈工程师的综合实力。媒介即信息,这使得作品集从一份被动的文档,转变为一场主动的能力演示。

II. 视觉识别系统:“科技黑色”设计语言

本章节将定义一套完整、可执行的设计规范,以确保网站整体视觉语言的统一性、专业性和主题契合度。我们将采用一种成熟的“科技黑色”(Tech Noir)美学风格,优先考虑信息的清晰度和视觉冲击力,而非混乱的噪音。

色彩搭配方案

色彩方案经过精心设计,旨在实现高对比度、优良的可读性以及强烈的主题共鸣。深色背

景基调营造出一种终端或未来派界面的氛围，而高能霓虹色则用于点缀交互元素和数据可视化，形成视觉焦点。

- 表格:核心色彩方案
 - 目的:建立一个严格、易于访问且与主题一致的色彩系统。
 - 说明:一个明确的调色板是品牌一致性和专业性的基石。所选颜色参考了对有效赛博朋克色彩方案的研究⁵，并特意选择了在深色背景下具有高对比度的颜色，以确保可访问性和视觉冲击力。

角色	颜色名称	HEX 色码	用途
主背景色	虚空黑	#0A0A14	页面主背景、模态框背景
次背景色	面板灰	#101424	项目卡片背景、时间线区块
主文本色	米白	#EAEAEA	正文、段落、描述文本
交互/强调色	电光青	#03D8F3	主要链接、按钮高亮、活动图表节点、标题
高亮/次强调色	霓虹品红	#FF007A	次要悬停效果、图表簇高亮
警告/行动号召色	故障黄	#FCEE0C	行动号召 (CTA) 按钮、特殊高亮

字体系统

字体选择将融合等宽字体的机械精准感与现代无衬线字体的可读性，构建清晰的视觉层级。

- 主字体(标题与UI): **Model Mono**⁹。其简洁、模块化的机械美学与赛博朋克主题完美契合，为标题和关键 UI 标签提供了强烈的结构感。
- 副字体(正文): **Neue Montreal Mono**⁹ 或 **Source Code Pro**¹⁰。这些是为开发者设计的高可读性等宽字体，确保详细的项目描述在保持终端风格的同时，依然易于阅读。
- 层级定义:

- H1 (英雄区标题): Model Mono, 4rem, 大写, 电光青 (#03D8F3), 并附带 text-shadow 以实现发光效果。
- H2 (章节标题): Model Mono, 2.5rem, 大写, 米白 (#EAEAEA)。
- 正文与段落: Neue Montreal Mono, 1.1rem, 米白 (#EAEAEA), 并设置较宽的行高以提升可读性。
- 代码片段: 使用专门的代码字体, 如启用连字功能的 **Fira Code**¹⁰, 并搭配合适的语法高亮主题。

UI 组件与图标

所有 UI 组件将采用硬朗的边缘、几何形状(如六边形、尖角矩形)以及与主题一致的交互反馈¹¹。

- 按钮: 将采用“故障”悬停效果。设计将借鉴 CodePen 上的示例¹³, 利用 CSS 伪元素 ::before 和 ::after 在悬停时创建分层、色差分离的视觉效果。
- 项目卡片: 悬停时将呈现微妙的全息光晕或扫描线叠加效果¹⁶。边框动画将沿着卡片边缘描绘, 以吸引用户注意力。
- 图标: 将使用一套定制的、具有锐利几何风格的 SVG 图标, 用于链接至 GitHub、LinkedIn 和电子邮箱。

III. 信息架构与页面蓝图

本章节详细阐述了单页应用(SPA)的结构, 设计为一次连续的、叙事性的滚动体验。信息流逻辑清晰, 引导用户从对 Zirui Wen 身份的宏观印象, 逐步深入到对其具体成就的详细探索。

“引导序列”: 英雄区与简介

- 概念: 首屏视图是一个全屏的沉浸式体验, 旨在确立“AI 架构师”的叙事基调。它不是一个静态横幅, 而是一个可交互的环境。
- 视觉呈现: 使用 react-three-fiber 创建一个动态的、生成式的粒子系统¹⁸。这些粒子将在背景中形成一个微妙的、流动的网格或神经网络结构, 并通过产生涟漪或轨迹来响

应鼠标的移动²⁰。这立即创造出一种“有生命的”数字空间感。

- 内容: 在此场景之上, 将叠加主标题: “ZIRUI WEN”, 其下是 “AI & MACHINE LEARNING ENGINEER”。这段文字将通过打字机动画与 CSS 故障效果结合的方式出现, 模拟终端启动过程¹²。一个微妙的、动画化的向下滚动指示器将引导用户进入下一部分。

“核心矩阵”: 交互式技能可视化

- 概念: 超越静态的技术列表, 展示各项技能之间的内在联系。本节将他的专业知识可视化为一个相互连接的系统, 从而证明他对整个技术栈的系统性理解。
- 实现方式: 使用 **D3.js**²⁴ 构建一个力导向图, 并将其集成到 React 应用中。该图表将是可交互的, 并在视觉上与赛博朋克美学保持一致。
- 数据结构: 节点将代表其简历中列出的技术¹, 如 Python, C++, PyTorch, LangChain, Neo4j, Kubernetes 等。连线代表它们之间的关联 (例如, Python 连接到 Pandas, PyTorch 和 Flask)。
- 交互性:
 - 聚类: 节点将通过聚类力 (clustering force) 按类别 (语言、深度学习、数据工程、DevOps) 分组²⁶。每个集群将有一个微弱的发光边界形状 (例如六边形)。
 - 悬停与高亮: 当鼠标悬停在一个节点 (如 “PyTorch”) 上时, 该节点及其直接连接会变亮, 而所有其他节点则会淡出。同时, 会出现一个工具提示, 列出使用了该技术的相关项目。
 - 筛选: 点击一个集群名称 (如 “DevOps”) 将使图表重新居中, 并高亮显示该集群内的所有节点。

“历程纪实”: 经验与研究时间线

- 概念: 以比标准简历列表更具吸引力的方式, 按时间顺序呈现其职业和学术经历。垂直时间线的格式直观, 并允许信息的渐进式披露。
- 实现方式: 一个响应式的垂直时间线组件²⁸。当用户向下滚动时, 每个条目将通过动画进入视野, 使用 **AOS.js**³⁰ 等库触发淡入或滑入动画。
- 内容: 每个时间线“区块”将代表一个关键事件: 研究助理职位、西门子实习经历以及发表的论文¹。区块将清晰地显示职位/标题、机构和日期。
- 可扩展细节: 每个区块都将是可点击的或带有一个“+”图标。交互后, 区块将平滑展开, 显示其简历中的要点, 特别是那些可量化的成就 (例如, “将答案的精确匹配准确率提

升了 32%”)。这既保持了初始视图的简洁性,又提供了按需深入了解的可能。

“成果展示”:深度项目档案

- 概念: 这是展示实践技能最关键的部分。它需要既有视觉吸引力,又能为希望深入研究的技术人员提供全面的细节。
- 布局: 一个响应式的项目卡片网格。每张卡片将包含项目标题、一句简短的标语以及所用关键技术的列表。
- 卡片悬停效果: 悬停时,卡片将展示多种效果的组合:微妙的 3D 倾斜、发光的边框,以及一个带有简短项目描述的叠加层¹⁶。
- 模态弹窗: 点击卡片将触发一个全屏模态窗口。这是一个至关重要的交互,必须专业地处理。
 - 可访问性: 模态窗口将遵循 WAI-ARIA 最佳实践进行构建³²,确保其可通过键盘导航、能够捕获焦点,并能被屏幕阅读器正确播报。将使用 `role="dialog"` 和 `aria-modal="true"` 等属性。
 - 内容: 模态窗口将包含详细的项目剖析:更长的描述、问题陈述、技术架构(可能附有简单的图表)、带有语法高亮的关键代码片段,以及指向 GitHub 仓库和在线演示的醒目链接(如果可用)。

招聘人员在筛选简历时会迅速寻找影响力。诸如“32%的准确率提升”、“60%的处理时间缩短”和“1,000倍的CPU加速”等数字是 Zirui Wen 简历中最有力的元素¹。传统的作品集布局常常将这些关键数字埋在段落文本中。因此,本设计方案必须在视觉上突出这些指标。在时间线和项目模态窗口中,这些数字将被赋予独特的样式——例如,使用“故障黄”强调色或稍大的字号——使其在信息流中脱颖而出,即刻吸引眼球。这一设计决策直接服务于候选人展示其高价值贡献的核心目标。

IV. 动态体验与微交互

本章详细说明了那些统一网站整体感官、创造精致沉浸式体验的微妙环境效果、过渡动画和交互细节。

全局氛围

- 动画扫描线叠加层: 一个持久的、半透明的叠加层将覆盖整个视口, 其上带有微妙且缓慢移动的水平扫描线。这种效果可通过纯 CSS 实现, 利用重复的线性渐变 (linear-gradient) 和关键帧动画 (@keyframes), 能即刻营造出复古 CRT 显示器或终端的质感³⁵。
- 交互式背景着色器: 英雄区的 WebGL 粒子系统可以以一种更柔和、更低调的形式延伸至整个页面背景。一个简单的低多边形网格或非常稀疏的粒子场可以对用户的滚动位置做出反应, 从而在不分散注意力或影响性能的前提下, 创造出深度感和动感。CodePen 上的一些示例展示了如何使用着色器创建交互式背景³⁸。

无缝导航与页面过渡

- 单页滚动: 网站将作为单页应用呈现。一个固定的导航栏 (可能采用垂直布局, 如¹¹所建议) 将允许用户在不同部分 (简介、技能、经验、项目、联系方式) 之间快速跳转。
- 平滑滚动: 点击导航链接将触发一个动画式的平滑滚动, 将用户带到相应的区域。
- 区域过渡: 当用户滚动时, 可以考虑使用如 **swup.js** 或 **Barba.js**⁴⁰ 这样的页面过渡库, 甚至是原生的 View Transition API⁴⁴, 来在不同主题区域之间创建微妙的动画过渡效果, 从而增强叙事的流畅性和电影感。

定制光标系统

- 概念: 在一个高度风格化的界面中, 默认的浏览器光标可能会破坏沉浸感。一个定制的光标不仅能强化主题, 还能提供更丰富的交互反馈。
- 实现方式: 我们将使用 CSS 和 JavaScript 来隐藏默认光标, 并渲染一个定制的 SVG 光标⁴⁵。光标本身可以是一个简约的未来派十字准星或几何图形。
- 状态变化: 光标的外观将根据其悬停在不同元素上而改变:
 - 在链接/按钮上: 它可能会放大、变为“电光青”色, 或显示一个“目标锁定”的微动画。
 - 在 D3 图表上: 它可能转变为一个“扫描”图标。
 - 在文本上: 它可能变回更传统的 I 型光标, 但其样式会与整体主题保持一致。

V. 技术实现蓝图

本章为开发工作提供了具体的实施计划，重点是采用一个现代、高性能且易于维护的技术栈。

推荐技术栈

- 表格:核心技术栈
 - 目的:为作品集项目提供一个清晰、合理且现代化的技术选型方案。
 - 说明:每一项选择都旨在平衡开发者体验、最终产品性能以及实现复杂视觉与交互需求的能力。选择 Next.js 是为了其卓越的性能优势;react-three-fiber 是在 React 中进行声明式 3D 开发的行业标准;而 Vercel 则为该技术栈提供了最无缝的部署体验。

类别	技术	理由与依据
前端框架	React with Next.js	提供了一个强大的基础，具备静态站点生成(SSG)和服务端渲染(SSR)等性能优化特性，这对于保证快速的初始加载时间至关重要 ⁴⁷ 。
3D/WebGL	Three.js + react-three-fiber (R3F)	R3F 是 Three.js 的声明式 React 渲染器，它简化了复杂 3D 场景的创建过程，并支持基于组件的架构 ¹⁸ 。
R3F 辅助库	@react-three/drei	这是一个为 R3F 提供的实用辅助工具和抽象集合，将极大地加速相机控制、模型加载等元素的开发 ¹⁸ 。
数据可视化	D3.js	数据驱动文档的黄金标准。对于创建定制化的、交互式的技能图谱至关重要 ²⁴ 。
动画库	GSAP (GreenSock Animation Platform)	一个功能强大的专业级动画库，用于对滚动动画、区域过渡和复

		杂的 UI 编排进行精细控制。
样式方案	Styled-components 或 Tailwind CSS	用于实现组件作用域内的样式, 确保 CSS 架构的可维护性和无冲突性。
部署平台	Vercel	与 Next.js 提供最佳的集成体验, 开箱即用提供自动构建、全球 CDN 和无服务器功能, 是追求性能的理想选择 ⁴⁷ 。

组件与项目结构

推荐采用基于功能的(feature-based)文件夹结构来组织 React 项目, 以促进可扩展性和可维护性。

- 结构示例⁵²:

```
/src
├── /components    # 可复用的、共享的 UI 组件 (Button, Modal, Card)
├── /features      # 网站中独立的特性区域
│   ├── /hero     # 英雄区的组件和逻辑
│   ├── /skills-graph # D3 图表组件和相关数据
│   └── /projects  # 项目网格、卡片和模态框内容
├── /hooks         # 全局自定义钩子
├── /lib           # 辅助函数、常量
└── /styles        # 全局样式、字体、主题变量
```

性能与可访问性指南

- 性能清单: 一份关键的最佳实践列表, 以确保这个重度依赖 WebGL 的网站保持高性能。
 - 最小化绘制调用 (**Draw Calls**): 对粒子使用实例化渲染(instanceing)⁴。
 - 优化几何体: 始终使用 BufferGeometry, 并保持较低的多边形数量⁵⁵。
 - 纹理管理: 使用压缩纹理和 Mipmaps⁴。
 - 懒加载: 异步加载 3D 模型和重型组件, 以加快初始页面加载速度。

- 代码分割: 利用 Next.js 的自动代码分割功能, 确保只加载当前视图所需的 JavaScript。
- 可访问性清单:
 - 确保所有交互元素(按钮、链接)都有清晰的焦点状态(focus states)³²。
 - 为 D3 图表和模态框正确使用 ARIA 属性, 以便为屏幕阅读器提供上下文³³。
 - 为所有有意义的视觉元素提供替代文本。
 - 确保颜色对比度符合 WCAG AA 标准。

部署策略

- 平台: **Vercel** 是首选的部署平台, 因为它与 Next.js(由同一团队创建)实现了无缝集成。这为 SSR/SSG 提供了优化的性能, 为每次 Git 推送提供即时预览, 并拥有强大的边缘网络以实现快速的全球内容分发⁴⁷。
- 流程: 将一个 GitHub 仓库连接到 Vercel, 以建立一个持续集成/持续部署(CI/CD)管道。使用环境变量来管理任何必要的 API 密钥或配置设置。最后, 设置一个自定义域名, 以呈现最终的专业形象。

引用的著作

1. Zirui Wen CV.pdf
2. 3d-portfolio-website · GitHub Topics, 访问时间为 八月 5, 2025, <https://github.com/topics/3d-portfolio-website>
3. Building Efficient Three.js Scenes: Optimize Performance While Maintaining Quality, 访问时间为 八月 5, 2025, <https://tympanus.net/codrops/2025/02/11/building-efficient-three-js-scenes-optimize-performance-while-maintaining-quality/>
4. How to Optimize WebGL for High-Performance 3D Graphics, 访问时间为 八月 5, 2025, <https://blog.pixelfreestudio.com/how-to-optimize-webgl-for-high-performance-3d-graphics/>
5. The Best 15 Cyberpunk Color Palette Combinations - Piktochart, 访问时间为 八月 5, 2025, <https://piktochart.com/tips/cyberpunk-color-palette>
6. How To Create a Cyberpunk Color Palette | Page Flows, 访问时间为 八月 5, 2025, <https://pageflows.com/resources/cyberpunk-color-palette/>
7. Cyberpunk Color Scheme - Palettes - SchemeColor.com, 访问时间为 八月 5, 2025, <https://www.schemecolor.com/cyberpunk.php>
8. Cyberpunk 2077 UI Colors Color Palette, 访问时间为 八月 5, 2025, <https://www.color-hex.com/color-palette/1041326>
9. Best Monospace Fonts for 2025 – Pangram Pangram Foundry, 访问时间为 八月 5, 2025, <https://pangrampangram.com/blogs/journal/best-monospace-fonts-2025>

10. 18 Best Monospace Fonts for Coding & Design, 访问时间为 八月 5, 2025, <https://octet.design/journal/best-monospace-fonts/>
11. The ONLY Guide to Cyberpunk UI Design - YouTube, 访问时间为 八月 5, 2025, https://www.youtube.com/watch?v=yLdJh2_-o8U
12. The Cyberpunk UI Principles Behind Cerebrasystem - DEV ..., 访问时间为 八月 5, 2025, <https://dev.to/mrscatalinajorquera/the-cyberpunk-ui-principles-behind-cerebrasystem-id7>
13. Pure CSS Cyberpunk 2077 Buttons - CodePen, 访问时间为 八月 5, 2025, <https://codepen.io/jh3y/pen/PoGbxLp>
14. CSS CyberPunk Buttons (:hover to glitch) - CodePen, 访问时间为 八月 5, 2025, <https://codepen.io/jh3y/pen/BajVmOg>
15. Pure CSS Cyberpunk 2077 Buttons - CodePen, 访问时间为 八月 5, 2025, <https://codepen.io/nquare/pen/abBQKad?anon=true&view=pen>
16. Awesome Looking CSS Hover Effects for You - Slider Revolution, 访问时间为 八月 5, 2025, <https://www.sliderrevolution.com/resources/css-hover-effects/>
17. CSS Hover Effects: 40 Engaging Animations To Try - Prismic, 访问时间为 八月 5, 2025, <https://prismic.io/blog/css-hover-effects>
18. Introduction - React Three Fiber, 访问时间为 八月 5, 2025, <https://r3f.docs.pmnd.rs/getting-started/introduction>
19. Build Website With Three JS and React Three Fiber. | by Samie Ullah Baig - Medium, 访问时间为 八月 5, 2025, <https://medium.com/@wuzsamie/build-website-with-three-js-and-react-three-fiber-488c73e982dd>
20. Interactive Particles with Three.js - Codrops, 访问时间为 八月 5, 2025, <https://tympanus.net/codrops/2019/01/17/interactive-particles-with-three-js/>
21. Particle system with Three.js - CodePen, 访问时间为 八月 5, 2025, <https://codepen.io/kelsS/pen/XBYMXY>
22. CSS Text Glitch Effects Examples 2025 - Avada | Blog, 访问时间为 八月 5, 2025, <https://blog.avada.io/css/text-glitch-effects>
23. Glitch Effect on Words - CodePen, 访问时间为 八月 5, 2025, <https://codepen.io/aquacode/pen/xxGGpEN>
24. GitHub - d3/d3: Bring data to life with SVG, Canvas and HTML. :bar_chart, 访问时间为 八月 5, 2025, <https://github.com/d3/d3>
25. D3 by Observable | The JavaScript library for bespoke data visualization, 访问时间为 八月 5, 2025, <https://d3js.org/>
26. Clustered Bubbles / D3 - Observable, 访问时间为 八月 5, 2025, <https://observablehq.com/@d3/clustered-bubbles>
27. d3-force-cluster - NPM, 访问时间为 八月 5, 2025, <https://www.npmjs.com/package/d3-force-cluster>
28. Vertical Timeline in CSS and JavaScript | CodyHouse, 访问时间为 八月 5, 2025, <https://codyhouse.co/gem/vertical-timeline/>
29. Vertical timeline (animated) - StackBlitz, 访问时间为 八月 5, 2025, <https://stackblitz.com/edit/stackblitz-starters-fvhzu7>
30. AOS - Animate on scroll library, 访问时间为 八月 5, 2025,

- <https://michalsnik.github.io/aos/>
31. michalsnik/aos: Animate on scroll library - GitHub, 访问时间为 八月 5, 2025, <https://github.com/michalsnik/aos>
 32. Technique: Accessible modal dialogs - Harvard's Digital Accessibility, 访问时间为 八月 5, 2025, <https://accessibility.huit.harvard.edu/technique-accessible-modal-dialogs>
 33. ARIA: aria-modal attribute - MDN Web Docs - Mozilla, 访问时间为 八月 5, 2025, <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-modal>
 34. ARIA: dialog role - MDN Web Docs - Mozilla, 访问时间为 八月 5, 2025, https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Roles/dialog_role
 35. Using CSS Animations To Mimic The Look Of A CRT Monitor | by Dovid Edelkopf | Medium, 访问时间为 八月 5, 2025, <https://medium.com/@dovid11564/using-css-animations-to-mimic-the-look-of-a-crt-monitor-3919de3318e2>
 36. Using CSS to create a CRT - Alec Lownes, 访问时间为 八月 5, 2025, <https://aleclownes.com/2017/02/01/crt-display.html>
 37. Retro CRT terminal screen in CSS + JS - DEV Community, 访问时间为 八月 5, 2025, <https://dev.to/ekeijl/retro-crt-terminal-screen-in-css-js-4afh>
 38. Interactive WebGL Image Effect - CodePen, 访问时间为 八月 5, 2025, <https://codepen.io/bubuiv/pen/bWJbMp>
 39. WebGL Interactive Background - CodePen, 访问时间为 八月 5, 2025, <https://codepen.io/moshimochii/pen/NPKXMMG>
 40. Page Transition Js Libraries, 访问时间为 八月 5, 2025, <https://mrinalcs.github.io/page-transition-js-libraries>
 41. Javascript libraries to create page transitions in your website - Denis Bouquet, 访问时间为 八月 5, 2025, <https://www.denisbouquet.com/javascript-libraries-create-page-transitions-web-site/>
 42. barba.js, 访问时间为 八月 5, 2025, <https://barba.js.org/>
 43. swup — Versatile and extensible page transition library for server-rendered websites, 访问时间为 八月 5, 2025, <https://swup.js.org/>
 44. View Transition API - MDN Web Docs - Mozilla, 访问时间为 八月 5, 2025, https://developer.mozilla.org/en-US/docs/Web/API/View_Transition_API
 45. Cyberpunk 2077 - Cursor - CodePen, 访问时间为 八月 5, 2025, <https://codepen.io/hipsterreed/pen/LYRKbeZ>
 46. Custom CSS Cursor - Coherent Labs, 访问时间为 八月 5, 2025, <https://coherent-labs.com/blog/uitutorials/custom-cursor/>
 47. Vercel vs Netlify: How to Pick the Right Platform - FocusReactive, 访问时间为 八月 5, 2025, <https://focusreactive.com/vercel-vs-netlify-how-to-pick-the-right-platform/>
 48. Vercel vs Netlify: Compare Jamstack Hosting and Deployment - Prismic, 访问时间为 八月 5, 2025, <https://prismic.io/blog/vercel-vs-netlify>
 49. pmndrs/react-three-fiber: A React renderer for Three.js - GitHub, 访问时间为 八

- 月 5, 2025, <https://github.com/pmndrs/react-three-fiber>
50. First React Three Fiber Application - Three.js Journey, 访问时间为 八月 5, 2025, <https://threejs-journey.com/lessons/first-r3f-application>
 51. Vercel vs Netlify: Choosing the right one in 2025 (and what comes next) | Blog - Northflank, 访问时间为 八月 5, 2025, <https://northflank.com/blog/vercel-vs-netlify-choosing-the-deployment-platform-in-2025>
 52. Folder Structures in React Projects - DEV Community, 访问时间为 八月 5, 2025, <https://dev.to/itswillt/folder-structures-in-react-projects-3dp8>
 53. Popular React Folder Structures and Screaming Architecture, 访问时间为 八月 5, 2025, <https://profy.dev/article/react-folder-structure>
 54. How can I optimise my THREE.JS rendering? - Questions, 访问时间为 八月 5, 2025, <https://discourse.threejs.org/t/how-can-i-optimise-my-three-js-rendering/42251>
 55. The Big List of three.js Tips and Tricks!, 访问时间为 八月 5, 2025, <https://discoverthreejs.com/tips-and-tricks/>
 56. WebGL Performance Optimization: Techniques and Tips - PixelFreeStudio Blog, 访问时间为 八月 5, 2025, <https://blog.pixelfreestudio.com/webgl-performance-optimization-techniques-and-tips/>