

李仁龙

个人信息

个人信息：男 | 22 | 广州 | 广东工贸职业技术学院 | 软件工程

求职意向：前端开发

联系方式：13798936727 | 3454515879@qq.com

Github：<https://github.com/LXRL>

技术栈

- **HTML + CSS**：编写语义化 HTML，掌握 Flex 布局、响应式设计、移动端适配方案（vw/rem、媒体查询），严格遵循 W3C 标准。
- **Vue/React**：熟练使用 Vue3（Vue Router/Pinia）和 React（Hooks/Redux），具备开发复杂 SPA 和封装高复用组件的能力。
- **uni-app**：基于 uni-app 开发多端应用（H5/小程序/App），熟悉 uni-app API（如路由、生命周期、原生模块调用），配合 uview UI 组件库快速实现高质量界面与交互。
- **JS/TS**：熟练 ES6+ 特性（模块化、解构等），熟悉 TypeScript 类型系统（接口、泛型类型），能通过高阶函数（map/filter）高效处理数据流。
- **网络协议**：熟练使用 Axios 进行网络请求，掌握其配置、拦截器、浏览器同源策略，解决跨域解决方案（CORS/代理）。
- **可视化开发**：熟练使用 Echarts 进行数据可视化。能够实现复杂的图形绘制和动画效果，定制化图表开发。
- **UI 框架**：熟练使用 Element Plus、Ant Design、Vant、uview 等组件库，具备组件二次封装能力。
- **Git 协作**：掌握 Git 进行代码版本控制，规范使用分支管理、代码合并及团队协作流程。

项目经历

SQL 代码生成器

【项目地址】<https://zhashut.top/>

【代码仓库】<https://github.com/LXRL/sql-generate-frontend>

【项目背景】

前端基于 Vue3 + Element Plus + Vite 的代码生成器共享平台。开发者可以在线制作、发布代码生成器，用户能够搜索、下载、在线使用生成器（如生成项目模板），管理员可集中管理所有用户和生成器。提供可视化 SQL 生成与数据模拟的一站式解决方案，降低数据库开发门槛。

【技术亮点】

- **多源数据导入**：实现 5 种模式（智能/预置表/JSON 配置/建表 SQL/Excel）导入，支持示例数据一键填充，通过 Pinia 管理解析与表单状态同步。
- **动态字段配置表单**：整合 Vue Draggable Next，实现字段的拖拽排序、重命名和实时预览，优化用户交互体验，做到所见即所得。
- **轻量化代码编辑器**：通过集成 vue-codemirror 提供 SQL / JSON / Go / Java 代码的语法高亮/智能提示/错误检测功能。

- **配置可移植性**: 采用 `navigator.clipboard` 实现表结构配置的 JSON 化复制/保存, 支持配置保存为可移植模板, 实现「设计-导出-复用」闭环。
- **状态分层管理**: 基于状态管理库实现前端状态的分层管理, 分别负责表单设计、生成结果存储以及公共/个人数据的管理, 确保状态职责清晰、数据流向可控。
- **数据生成与展示分离**: 通过中间存储层解耦表单设计与数据展示的逻辑, 确保生成结果能够独立存储并通过监听机制实时更新到视图层。
- **字段级动态编辑器**: 封装可复用的字段配置组件, 支持多模拟策略 (随机/递增/正则/词库/禁用), 通过条件渲染动态切换参数输入组件, 实现字段级数据规则的精细化配置。

Nexus Agent

【代码仓库】 <https://github.com/LXRL/nexus-agent-ui>

【项目背景】

一款对话式智能(BI)平台的前端应用。该平台旨在革新传统BI的交互模式, 通过自然语言驱动, 为用户提供从数据查询、分析洞察到资产沉淀的全链路闭环体验。

【技术亮点】

- **流式响应**: 基于 `EventSource` 实时处理SSE事件流, 通过渐进式地更新组件State, 将SQL、表格、图表等多模态 `payload` 动态映射并组合渲染至单一消息气泡中, 实现了AI思考过程的可视化。
- **上下文感知型交互**: 在AI消息中嵌入交互式“动作”按钮, 通过回调向父组件传递上下文ID以触发特定API调用, 并将返回的结构化报告渲染为新消息追加至对话流, 实现了探索式分析闭环。
- **认证与API服务层抽象**: 基于 `Axios` 拦截器构建了独立的API服务层, 在请求阶段自动从 `Redux` 状态中读取并注入 `X-User-ID` 认证头, 实现了认证逻辑与业务组件的完全解耦。
- **深度主题定制**: 基于CSS变量构建双色主题, 并通过 `ConfigProvider` 对 `Popconfirm`、`Tag` 等多个组件的Design Token进行深度覆盖, 解决了在复杂自定义主题下三方组件的视觉一致性难题。
- **复杂动态表单**: 运用 `Form.usewatch` 与 `Form.List` 实现表单项的条件渲染与动态增减 (如动态添加表结构列), 并将创建/编辑逻辑封装至单一可复用模态框中, 显著提高代码复用率。
- **无中断对话流优化**: 运用 `useRef` 与 `useEffect` 监听 `isLoading` 状态, 在每次AI异步响应结束后自动将焦点重新定位到输入框, 免去用户二次点击, 实现了流畅无缝的连续对话体验。
- **通用数据状态展示封装**: 封装通用的 `DataDisplayWrapper` 组件, 统一处理了异步数据的加载、空状态及错误 (含重试) 视图, 显著提升了代码复用与全应用UX的一致性。

仿 ElementPlus 组件

【项目地址】 <https://lxrl.github.io/element-plus-mock/>

【代码仓库】 <https://github.com/LXRL/element-plus-mock>

【项目背景】

基于 Vue3 和 TypeScript 开发的高仿 ElementPlus 组件库, 支持按需加载、主题定制, 集成 Vite 等工具链, 完成核心组件开发及单元测试, 确保高性能与稳定性。

【技术亮点】

- **复杂交互组件实现**: 实现了高复杂度的组件 (如 Select、Form、Message 等), 解决了跨组件通信、多层嵌套和异步更新等技术难点。
- **高性能组件库设计**: 结合虚拟 DOM 的 VNode (虚拟节点), 优化组件的渲染性能, 确保组件在高频交互场景下的流畅度与稳定性。
- **防抖优化用户输入**: 实现了防抖 (debounce) 功能, 有效解决用户输入时的性能问题, 减少了频繁触发的事件对性能的影响。

- **PostCSS 插件处理 CSS 变量**: 使用 PostCSS 插件优化 CSS 变量的管理，增强了样式的动态性和可维护性，实现了组件库灵活的主题定制功能。
- **组件库文档自动化生成**: 借助 Vitepress 构建高质量文档站点，实现了文档的自动化生成与部署，覆盖组件使用方法、Props、事件以及代码示例的说明。
- **单元测试覆盖与质量保障**: 使用 Vitest 和 Vue Test Utils 进行单元测试，为关键组件编写测试用例，覆盖组件的核心逻辑与边界场景。
- **严格的代码规范与工程化实践**: 使用 ESLint 和 Prettier 统一代码风格，结合 Git 提交规范 (Commitlint + Husky)，保证代码质量和团队协作效率。

技术栈

广东易联智慧科技有限公司 | 前端开发（实习生）| 2025.03 - 2025.06

- **云手机管理平台**

- 负责云手机管理平台前端视频素材管理模块的开发，实现视频上传、分类、预览等功能。
- 参与视频模板组件的设计与实现，支持模板的创建、编辑与复用，提升内容生产效率。
- 负责平台权限管理前端实现，支持角色分级、细粒度权限分配与操作可见性控制。

- **云手机管理平台APP端**

- 开发设备远程管理模块，支持设备分组、批量操作、远程控制与状态实时同步，提升设备运维效率。
- 实现素材上传、分发与移动端预览功能，支持多格式文件的高效流转与批量管理。
- 负责前端权限与角色控制，实现细粒度权限分配与操作可见性，保障数据安全与合规。