

甘智斌

☎ 19079602718 ✉ luli_0819@qq.com

🎓 本科 🌐 <https://new-resume-rho.vercel.app/>



教育经历

东华理工大学 软件工程 本科

2023-09 ~ 2027-07

- GPA: 专业前10% ; 入党积极分子 ; 大一曾参与老师团队项目开发,负责接入讯飞大模型 , 实现智能聊天
- 荣誉/奖项: 江西省蓝桥杯二等奖、信息系统创新大赛三等奖、 二等奖学金、东华理工ECPC银奖

个人作品

- 稀土掘金: 90+ 技术文 , 四级作者 , 人气作者排名52
- 算法训练: LeetCode 110+ , 附 掘金题解 ; 蓝桥 OJ250+ , 200+题解
- Gitee: <https://gitee.com/luli1314520> 记录自学过程

实习经历

北京爱乐尔科技公司(Airacle)

2025-03 ~ 至今

AI全栈开发实习生

AI科研聚合网站

简介:使用nextjs、react、shadcn、tailwindcss、Supabase、Auth.js、Stripe、aisdk等技术栈实现科研从idea到paper的工具聚合平台 , 旨在利用LLM革新下一代科研范式

- 利用next-intl全面国际化,通过AITDK插件走查网页SEO,并用prompt驱动cursor高效提升落地页的关键词密度
- 自主封装了AI Detector和Humanizer组件,并进行优化: 布局抽离为服务端组件,交互逻辑为客户端端组件,有效减少了bundle包的大小。同时,封装了ai-detector和ai-humanizer接口,进行AI查重测试
- 基于Auth.js框架完成Github、Google国外平台登录鉴权,并通过阅读Auth.js文档实现网页端微信扫码登录

DeepInsight

简介:一款基于nextjs、python、fastapi、deepseek-r1-250120技术栈开发的项目文档评审工具

- 结合甲方表格截图,利用Loveable快速实现简约现代化ui ,并使用cursor对细节优化
- 封装AI analysis 组件: 自定义hooks useSSE , 基于SSE技术分别对deepseek中content、reasoning_content 缓存过滤并流式渲染,添加样式到文档头保证渲染稳定性,通过js自定义算法控制文字流速均匀稳定
- 封装useReviewForm: 监听后端流式响应,获取到完整json数据结构后,使用jsonrepair修复并实现一系列容错机制 , 保证能正确映射到指定表格体中,增强程序鲁棒性
- python后端优化SSE响应速度,开三个线程并行启动每个任务,产生结果后立即放入共享队列,主异步函数从 队列中实时获取结果并立即返回给前端,响应速度优化30%
- 通过SSH登录远程服务器,打包构建前后端代码 , 部署上线交付第一版

专业技能

- 掌握HTML、CSS、JavaScript和ES6、TypeScript等基础技术 ; 熟练使用git指令进行版本控制
- 熟练使用React、Vue3 ,以及VantUI、Element plus、Antd等前端组件库
- 全流思维: 了解Nodejs , 能够利用Express、koa构建Restful API ; 了解python、gradio、fastapi,进行agent开发
- AIGC:使用过OpenAI Agent SDK以及向量数据库Faiss , 了解基于LangChain.js的RAG开发
- 擅长prompt技巧与cursor、lovable协作,以planner、debugger视角进行需求规划与多端调试
- 了解多种部署方式: 使用过devbox、阿里云ECS + 宝塔linux、vercel 部署过项目 , 了解docker

项目经历

墨香逸读

技术栈: Vue3、Pinia、Vant UI、Iconfont、Tailwind、TypeScript、CozeBot

- 使用keepalive缓存页面,通过路由和组件的懒加载等技术对项目进行了优化,提升了页面加载速度和用户体验。
- 封装了动态组件,根据对话的身份验证加载不同的组件,实现了AI对话的渲染效果。
- 借助Coze的API接入了为项目打造的Agent,根据用户的输入,能够根据用户需求回答问题。
- 根据需求自定义hooks,判断图片是否出现在视窗,并结合自定义指令v-lazy实现图片的懒加载功能,提升页面加载速度。
- 项目地址: <https://gitee.com/luli1314520/AI-full-stack/tree/master/project/books>

卷王笔记

技术栈: Vue3、Vant UI、Nest.js、TypeORM、LangChain、DeepSeek、Mysql

- 二次封装 axios 并利用 JWT 实现前后端双 token 无感刷新,优化用户体验
- 利用 SSE 服务端至客户端单向通信技术,实现流式输出
- 使用 TypeORM 框架封装 sql 操作,进行数据查询和密码校验
- 大文件上传操作:实现文件分片、断点续传、秒传
- 将文档实现加载、切割、并且使用Embedding模型向量化,Vector Store(Faiss)中持久化
- 将多个 prompt 控制的 chain组装为一条RAG chain,并结合 Memory拓展 llm上下文

项目地址: <https://gitee.com/luli1314520/AI-full-stack/tree/master/project/RAG-project>