

# 韩王宇

(+86) 158-9550-5220 | [hanwangyu@foxmail.com](mailto:hanwangyu@foxmail.com) | [github.com/Han-Wangyu](https://github.com/Han-Wangyu) | Web 前端开发

## 教育经历

南京信息工程大学 | 数据科学与大数据技术

2021.09—2025.06

中级软件资格(软件设计师) | CET-6 | 获二等奖学金, 江苏省蓝桥杯 Web 开发组省二等奖, 全国大学生算法设计与编程挑战赛铜奖, 中国高校大数据挑战赛一等奖。

## 专业技能

- 熟悉 HTML CSS3 JavaScript 及常用 ES6 编程语法, 可快速搭建界面, 有真实发票预览类界面开发的开源经历;
- 熟悉 React 框架生态, 包括 React Hook, Next.js, shadcn/ui, React Native, Expo Router;
- 熟悉浏览器渲染原理及计算机网络 HTTP 相关原理; 了解常用数据结构与算法;
- 了解 Next.js 及 Node.js 的全栈开发实践, 有 Prisma、服务端渲染及 Vercel 等 Serverless 平台使用经验;
- 了解 Webpack, Vite 打包工具的使用。

## 项目经历

Discord-clone: Next.js 全栈项目 | 全栈开发

2023.10—2023.12

- **技术栈:** Next.js 13 + React + TypeScript + Socket.io + Prisma + Tailwind CSS + React Hook Form
- **项目介绍:** 参考海外交流平台 Discord 的社区型聊天程序开发, 全面支持语音、视频和文字通信。允许用户新建或通过邀请链接加入群组中进行实时通信, 包括完整的权限管理机制和实时删除或编辑消息。
- **项目实施:**
  1. 使用 Tailwind CSS 和 shadcn/ui 无头组件库实现移动端响应式布局和深浅色模式切换。
  2. 使用 Socket.io 封装基于 WebSocket 的高可用 Hook 并绑定连接状态供全局使用, 设计呈现连接状态的 UI 组件。
  3. 使用 React Query 中的 useInfiniteQuery 封装消息分页查询 Hook; 监听 WebSocket 的连接状态配置 refetchInterval 参数实现自动回退 HTTP 长轮询提高可用性。
  4. 使用 React Hook Form 实现表单并配合 Zod 实现对表单信息的验证; 引入 Zustand 全局管理模态对话框的状态并封装 Hook 实现解耦。
  5. 基于 Next.js 的服务端组件完成项目的后端开发。使用 Prisma ORM 操作 CockroachDB 开源 Serverless 数据库; 利用开源云服务提高开发效率, 如使用 Clerk 实现包括 OAuth 和邮箱验证的登陆模块, LiveKit 实现基于 WebRTC 的语音和视频通话频道。

Apple-Website-clone: Three.js & GSAP 项目 | 前端开发

2024.01—2024.04

- **技术栈:** React + Three.js + React Three Fiber + React Three Drei + GSAP + Vite + Tailwind CSS
- **项目介绍:** 参考 iPhone 15 Pro 官网开发, 重点实现了复杂的动画效果和手机 3D 模型预览。
- **项目实施:**
  1. 使用 GSAP 实现视频轮播组件, 包括动态更新进度条和自动切换功能。
  2. 使用 Three.js 及 React 下的有关封装库实现从不同角度探索 3D 模型, 并提供了不同的颜色切换和带动画效果的尺寸切换选项。
  3. 接入开源监控平台 Sentry 并集成于 Vercel 部署实现性能监控和错误上报。对模型渲染部分通过 Suspense 进行优化, 监控得到 FID 指标的 75 分位值仅 10ms, 页面响应速度良好。

Diliveroo-clone: React Native 跨端项目 | 前端开发

2023.10—2023.11

- **技术栈:** React Native + Expo Router + Zustand + React Native Reanimated
- **项目介绍:** 参考海外知名订餐 App Diliveroo 界面的移动端开发项目。
- **项目实施:**
  1. 动画性能优化: 使用 React-Native-Reanimated 库替代 React Native 的原生动画实现, 将动画效果默认在 UI 线程运行, 从而减少 JS 线程计算和异步线程通信所带来的性能损耗。
  2. 长列表性能优化: 通过配置 initialNumToRender 等列表属性来提高首屏加载速度; 避免在 renderItem 函数中使用匿名函数, 优化创建匿名函数作为渲染项时额外内存开销和垃圾回收压力。