## ****DAY1学习总结 – Cashierly POS 项目初始化****

### 1. 技术选型与环境准备

**从 Vue 2 转向 Vue 3 + Vite**

原因：Vue 3 是现代项目的主流，性能更好，生态更新。

过程中升级了 Node.js 版本（从 v18 升级到 v20）以兼容 Vite 7。

**安装和配置依赖**

vue-router (路由管理)

pinia (状态管理)

element-plus (UI 组件库)

vue-i18n (国际化)

**可吹点**：

“这个项目我从 Vue 2 的 CLI 方案迁移到了 Vue 3 + Vite 的现代技术栈，解决了 Node 版本和依赖不兼容的问题，保证了项目更轻量和快速开发。”

### 2. 基础目录结构规划

cashierly-pos/

├─ src/

│ ├─ locales/ # 国际化文件

│ ├─ router/ # 路由配置

│ ├─ store/ # 状态管理

│ ├─ views/ # 页面组件

│ │ ├─ Login.vue

│ │ └─ Dashboard.vue

│ ├─ App.vue

│ └─ main.js

### 3. 登录页功能开发

**国际化切换**（中英文切换）

**Element Plus 表单校验**

**美观布局（左右分屏）**

**表单验证成功后跳转到 Dashboard**

**可吹点**：

“我自己手写了一个符合商业审美的登录界面，支持中英文切换，使用 Element Plus 的 Form 组件做了表单验证和跳转，整个流程前端独立完成。”

### 4. 路由逻辑

默认 / 跳转到 /login

登录成功后跳转 /dashboard

**可吹点**：

“我配置了基于 Vue Router 的路由守卫，让项目启动默认进入登录页，后续可以扩展权限控制。”

### 5. 遇到的典型问题与解决方案

**Node 版本问题**：升级到 Node 20 后解决 Vite 启动报错。

**路径别名 @ 不生效**：手动改为相对路径 ../views/Login.vue 或在 vite.config.js 里配置 alias。

**空白页调试**：确认 App.vue 使用 <router-view />。

## ****今天你掌握的内容****

**用 Vite 创建 Vue 3 项目的完整流程**

**如何配置国际化、多语言切换**

**Element Plus 的基本使用（布局、表单、校验）**

**路由的配置与默认重定向**

**常见错误排查（空白页、路径 alias、Node 版本）**

## ****面试可吹方向****

“我最近在独立开发一个收银系统 Demo（Cashierly POS），  
从技术栈选择、Node 环境处理、项目脚手架搭建、到多语言国际化、UI 框架整合、路由权限配置，都是自己手动完成的，  
这个项目后续会逐步完善订单管理、仪表盘、权限角色等功能，作为完整的可展示作品。

## 建议功能范围（适合作品集）

**必须完成（必做）**

**登录**（模拟 API，支持多语言）

**token 持久化 + 登录后跳转 Dashboard**

**Dashboard 首页简单两三块内容**

**API 文件夹结构化 + mock 模拟后台**

**加分项（建议做）**

注册和忘记密码页面可以加 **空界面**，展示路由和界面布局，但不必实现真正注册逻辑。  
（点击按钮可以弹出 “此功能暂未开放”）

登录时增加 **加载中状态** 和错误提示，显得专业。

Footer 版权信息、右上角语言切换等 UI 完善。

**不建议现在做**

完整注册逻辑（需要 mock 保存用户数据、密码加密）

忘记密码完整流程（需要邮件验证码等）