前端冒烟测试

1. 报告概述

项目名称	植悟
测试类型	前端冒烟测试
被测版本	V0.4.0
测试周期	2025 年 12 月 12 日
测试结论	全部核心业务流 100% 通过验证。 经过全面冒烟测试，系统的注册登录、首页动态加载、植物管理、养护日记、个人中心以及 AI 助手的核心链路均已跑通。应用表现稳定，未发现阻塞性缺陷，具备上线或进阶测试条件。

2. 测试环境与配置

字段	配置/说明
操作系统	Windows 10
浏览器	Google Chrome
后端状态	FastAPI 服务 (124.71.227.181:8000) 运行正常，API 响应及时。。
测试数据	使用模拟数据和已知的后端 API 响应。

3. 核心指标总结

指标	数值	状态	备注
总测试项	27 项	-	覆盖所有 6 大核心模块
通过数量	27 项	极佳	所有用例均达到预期结果
失败数量	0 项	无	无任何高优先级或低优先级缺陷
整体通过率	100%	通过	完全满足上线/发布标准

4. 详细测试记录

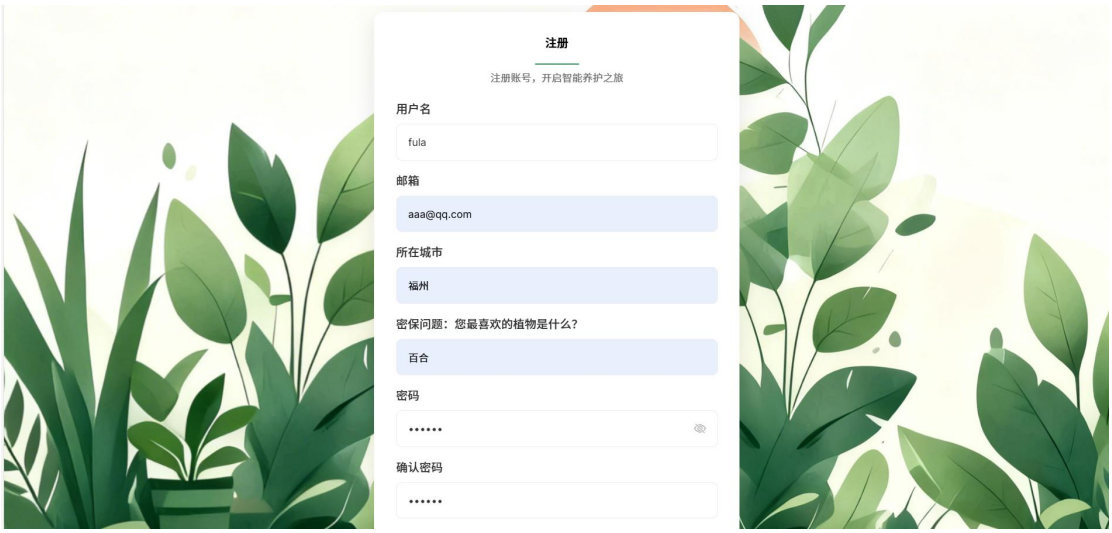
一. 核心功能模块：用户认证

目标与范围
• 测试目标： 快速验证用户注册和登录的核心业务流程是否连通，界面元素是否正常显示，以及能否触发后端 API 调用并接收成功的响应。
• 测试环境：
• 操作系统：Windows 10
• 浏览器：Google Chrome
• 运行状态：远程服务器 API 已部署并可跨域访问。

验证项	结果	简述
注册主流程	通过	成功提交用户名、邮箱、城市及密保答案，并正确跳转。
登录主流程	通过	可使用新注册账号成功登录，并跳转到应用主页。
登出功能	通过	点击后正确清除 Session/Token，并返回登录页面。
界面稳定性	通过	所有元素（输入框、按钮）布局正常，无明显 UI 错误。

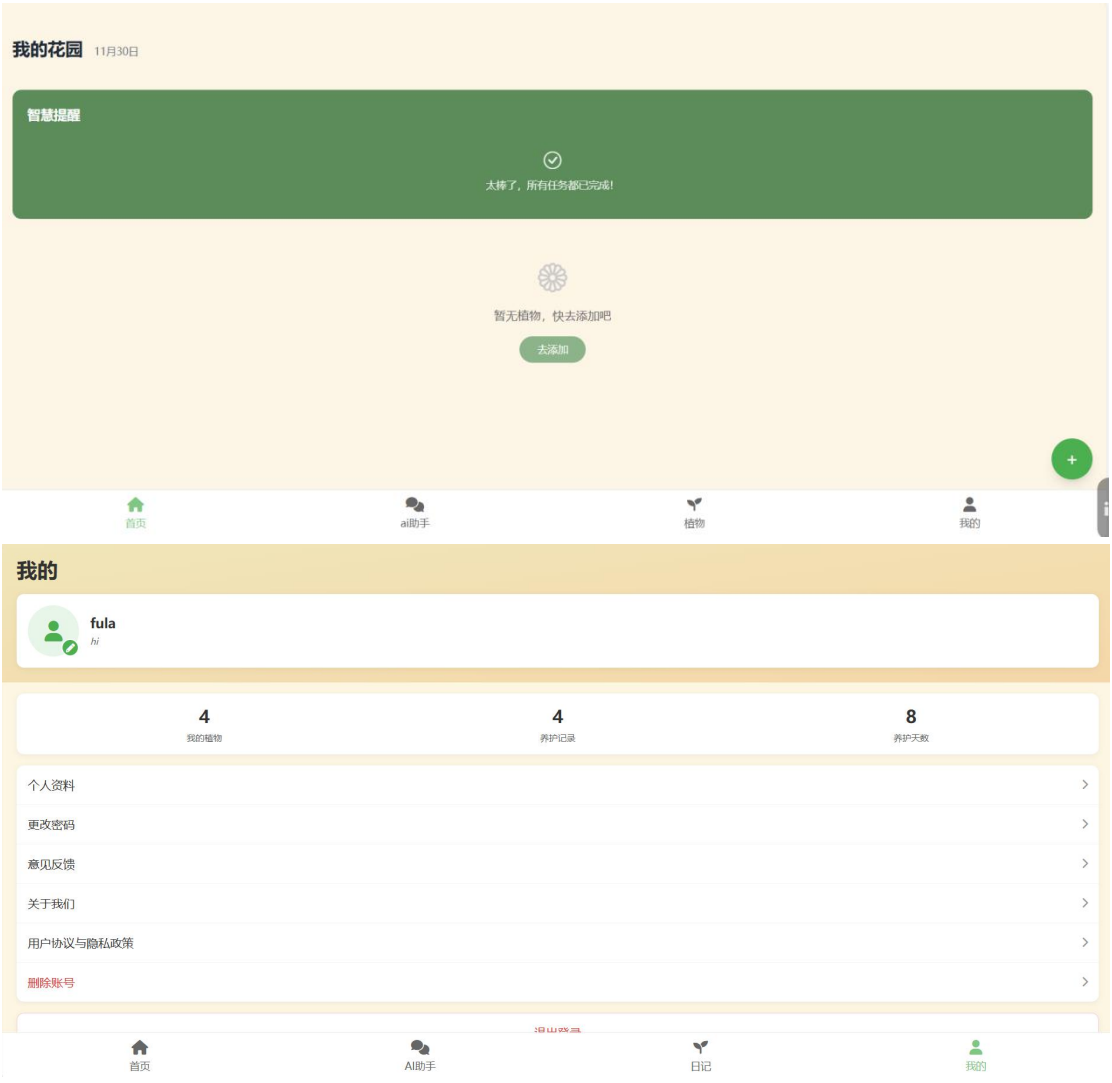
1. 成功注册

测试步骤：
访问注册页面。
在邮箱、用户名、密码，城市，密保问题字段中输入符合要求的有效数据。
点击“注册”按钮。
预期结果：
界面显示“注册成功”的提示信息。
自动跳转到登录页面或主页。
实际结果： 通过。
证明：



2. 成功登录

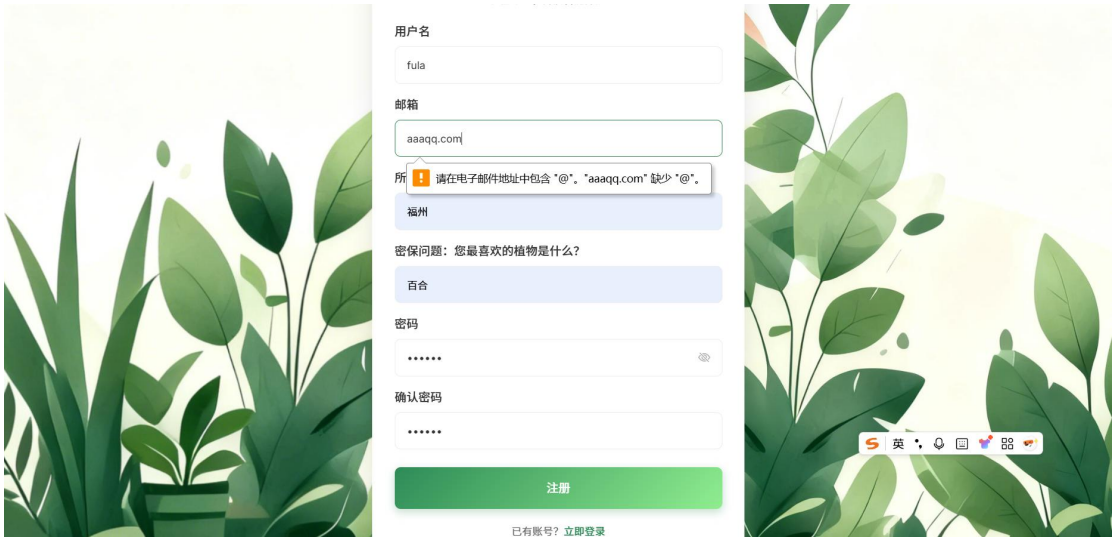
测试步骤：
访问登录页面。
输入已注册用户的用户名和密码。
点击“登录”按钮。
预期结果（前端视角）：
界面无报错信息。
页面成功加载应用主界面/仪表板。
导航栏或用户中心能正确显示用户昵称或头像。
实际结果： 通过。
证明 ：



3. 错误处理验证

测试步骤：
在注册页面，输入不符合格式的邮箱地址（如缺少 @ 符号）。
点击“注册”按钮。
预期结果（前端视角）： 页面应在提交前或提交后显示清晰的错误提示（如“邮

箱格式不正确”），并且不应跳转页面。
实际结果： 通过。 验证了前端的输入校验功能正常工作。



二、 核心功能模块：首页数据与交互

目标与范围
测试目标： 验证登录后首页布局是否正常，核心数据（植物列表、智慧提醒）能否正确加载渲染，以及关键交互（如打卡）能否成功同步。
测试环境：
操作系统：Windows 10
浏览器：Google Chrome
运行状态：远程服务器 http://124.71.227.181:8000 。
总体结论

验证项	结果	简述
数据加载	通过	植物列表和智慧提醒列表成功从后端 API 加载并渲染。
导航连通	通过	底部导航栏和添加按钮（+）的跳转功能正常。
核心交互	通过	提醒任务打卡可成功触发 POST 请求，并实现“乐观更新” UI 效果。
UI 渲染	通过	成功处理了图片加载失败的 onerror 备用显示逻辑。

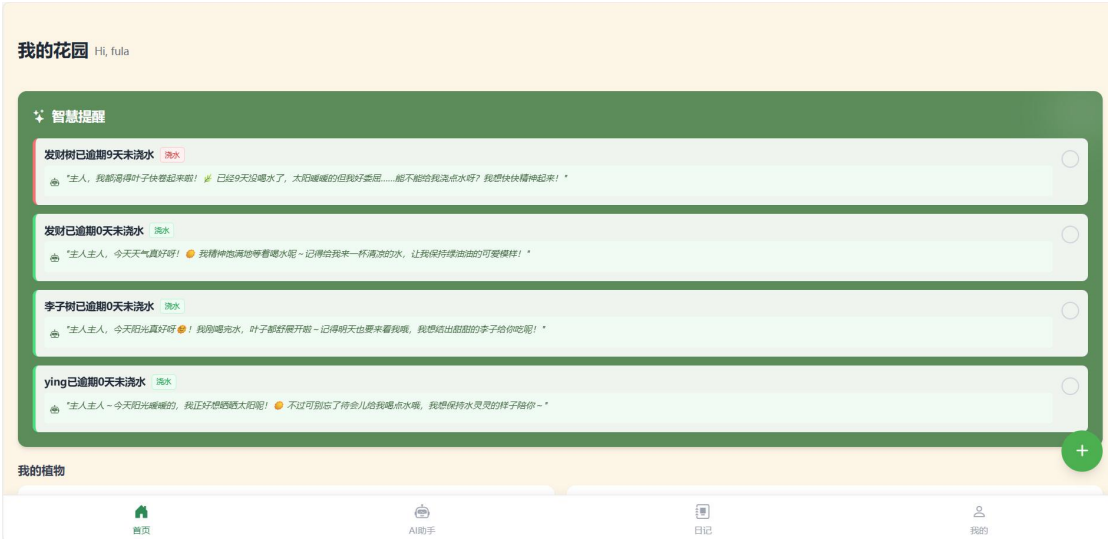
详细测试结果与证明

1. 界面与导航连通性

编号	测试描述	预期结果	实际结果
NAV-001	页面布局检查	头部标题、提醒卡片、植物列表和底部导航栏布局无错乱。	通过
NAV-002	导航栏高亮	底部导航栏中“首页”链接高亮显示。	通过
NAV-003	添加按钮跳转	点击右下角 + 悬浮按钮。	页面成功跳转到 my-plants.html。

证明：

NAV-001、NAV-002：



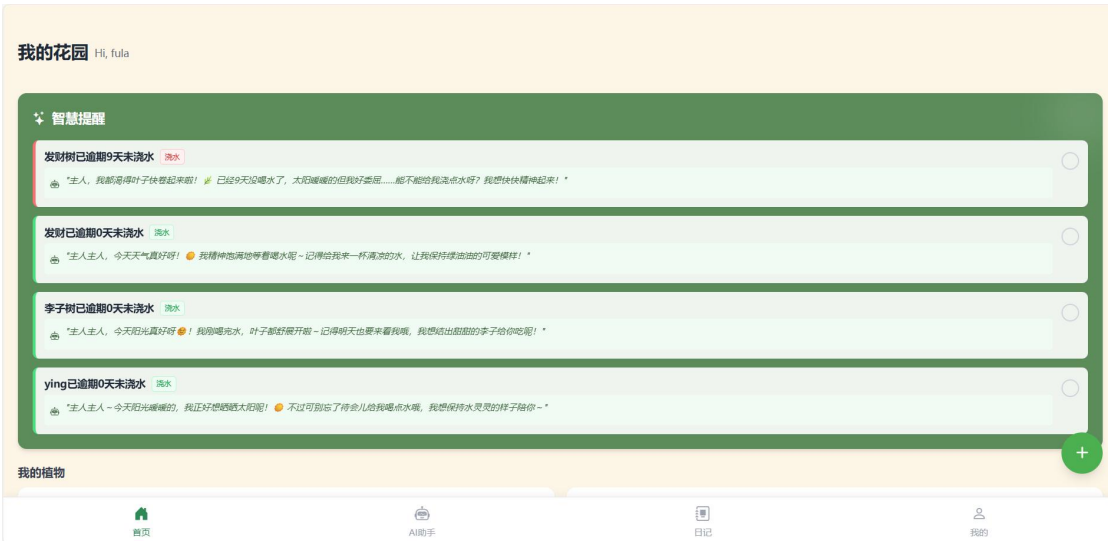
NAV-003：

2. 核心数据加载

编号	测试描述	预期结果	实际结果
DATA-001	植物列表加载	调用 /get_plants 接口,渲染出包含昵称和倒计时的卡片。	通过
DATA-002	智慧提醒加载	智慧提醒区域成功显示后端返回的待办提醒列表。	通过
DATA-003	空状态处理	显示“暂无植物”及“去添加第一盆”引导按钮。	显示“所有任务都已完成！” 的提示。
DATA-004	日期计算验证	根据 last_watered 自动显示“已逾期”、“今天浇水”或“X 天后”。。	已逾期、今天浇水、还有 X 天 等逻辑计算和显示正确。

DATA-001:

DATA-002、DATA-004:



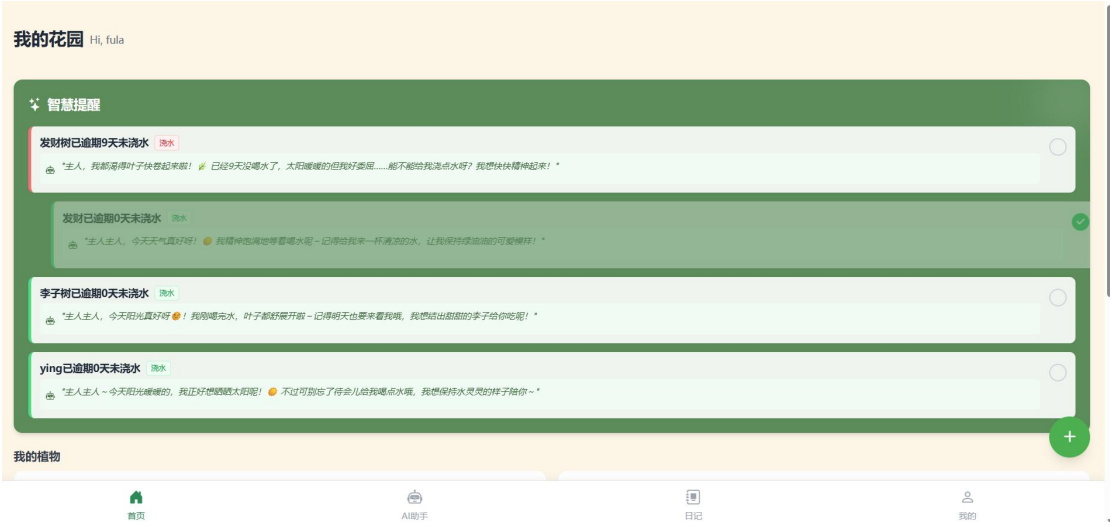
DATA-003:

3. 核心交互连通性

编号	测试描述	预期结果	实际结果
INTER-001	打卡成功	针对一个提醒任务，点击圆形打卡按钮。	1. 成功发送 POST 请求到 /plants/[id]/[type]。 2. 提醒项从列表中消失。 3. 植物卡片上的浇水日期同步刷新。

证明：

INTER-001：



三、核心功能模块：我的植物（my-plants.html）

目标与范围 测试目标：验证“我的植物”页面的列表渲染、热门植物联动逻辑、新增表单交互以及页面间的跳转连通性。测试环境：
操作系统：Windows 10
浏览器：Google Chrome
运行状态：后端服务运行于 http://124.71.227.181:8000
总体结论

验证项	结果	简述
列表渲染	通过	成功调用 /get_plants 接口并渲染用户植物；热门植物区静态展示正常。
新增入口	通过	支持通过顶部按钮直接新增，或通过热门植物详情页关联新增。
详情弹窗	通过	弹窗系统（详情、新增、热门）切换逻辑正常，支持背景点击关闭。
导航连通	通过	底部导航栏与顶部“返回”箭头跳转逻辑符合预期。

详细测试结果与证明

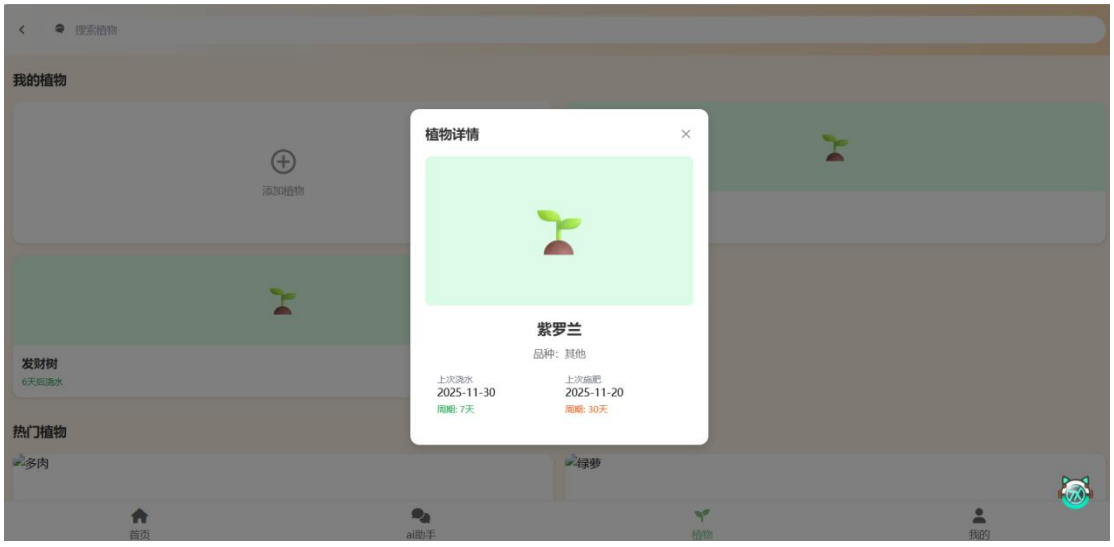
1. 列表渲染与连通性

编号	测试描述	预期结果）	实际结果
LIST-001	页面布局检查	头部、植物列表、热门推荐区和底部导航栏布局无错乱。	通过
LIST-002	现有植物详情	点击“我的植物”列表中的任一卡片。	弹窗成功显示植物详情，内容填充正确。
LIST-003	热门植物详情	点击“热门植物”列表中的任一卡片。	弹窗成功显示热门植物详情。
LIST-004	底部导航连通	点击底部导航栏，如“首页”图标。	页面成功跳转到 index.html。

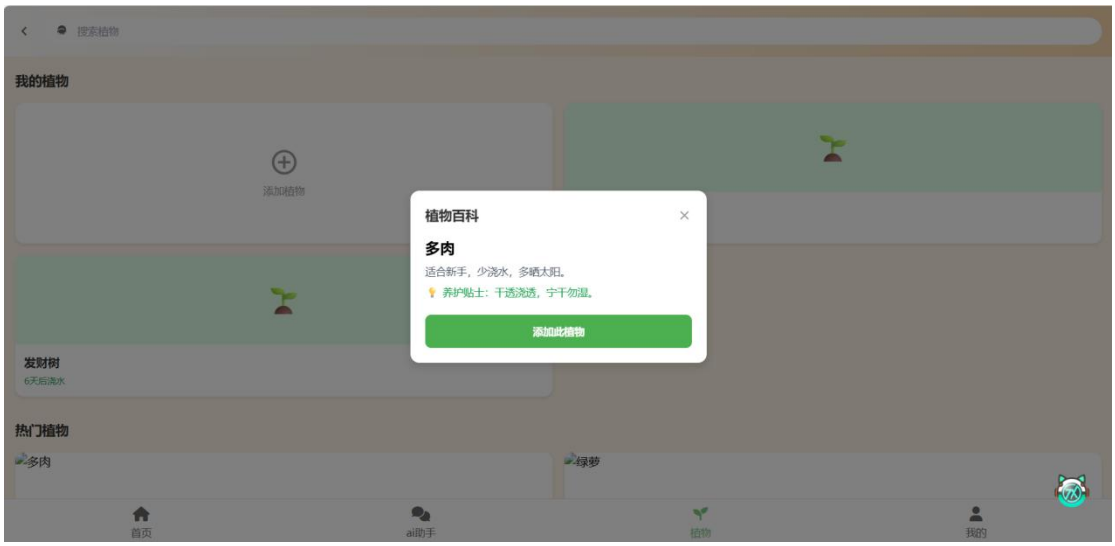
证明：

LIST-001：

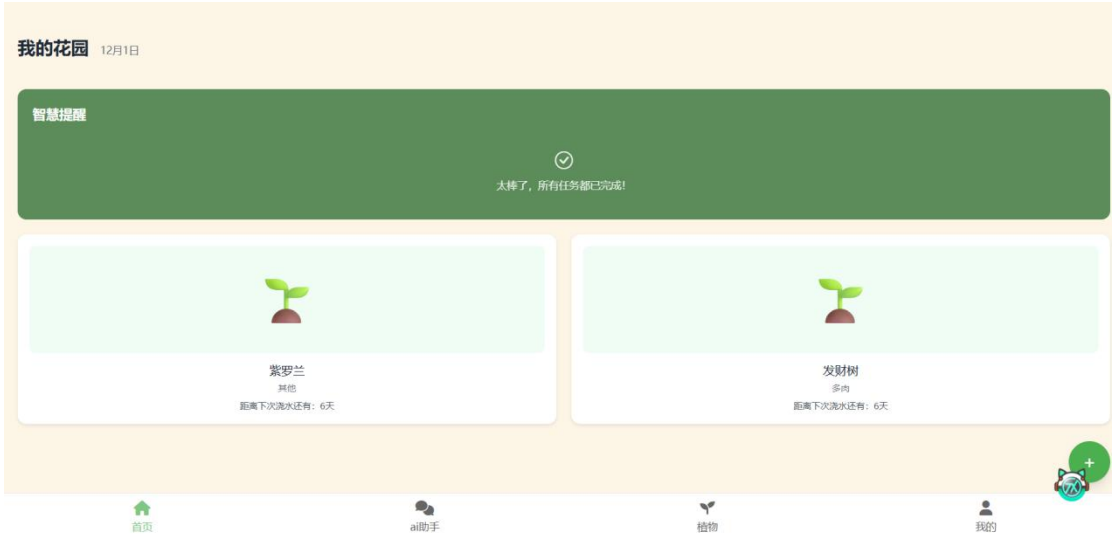
LIST-002：



LIST-003：



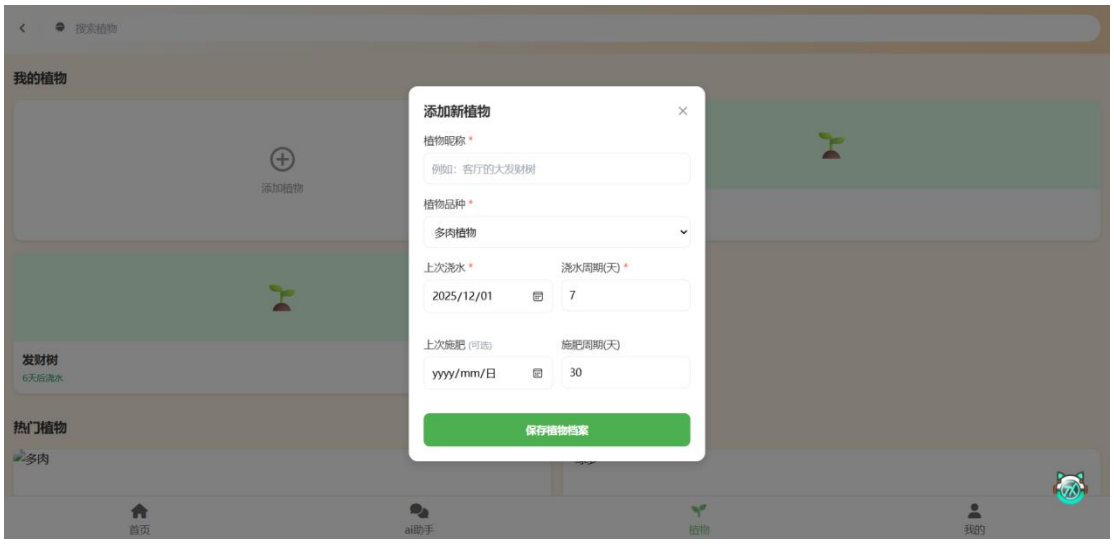
LIST-004：



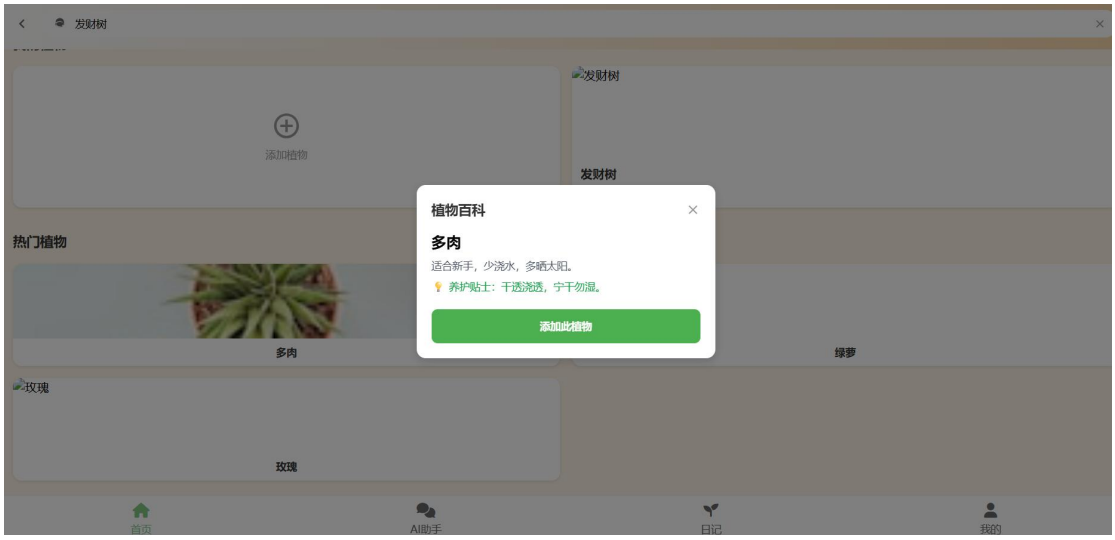
2. 核心交互验证

编号	测试描述	预期结果	实际结果
MODAL-001	新增植物入口	点击头部“新增植物”按钮。	弹窗正常弹出，表单元素可见。
MODAL-002	热门转新增	在热门植物详情弹窗中，点击“加入我的植物”按钮。	成功关闭热门植物弹窗，并自动打开 ，且植物名称字段已预填充。
MODAL-003	搜索功能（前端过滤）	在搜索框输入部分植物名称。	可以过滤。
MODAL-004	AI 联动推荐	在新增弹窗选择“绿萝”或“发财树”。	触发 fetchAIRecommendation，自动获取并填充推荐的浇水/施肥周期。

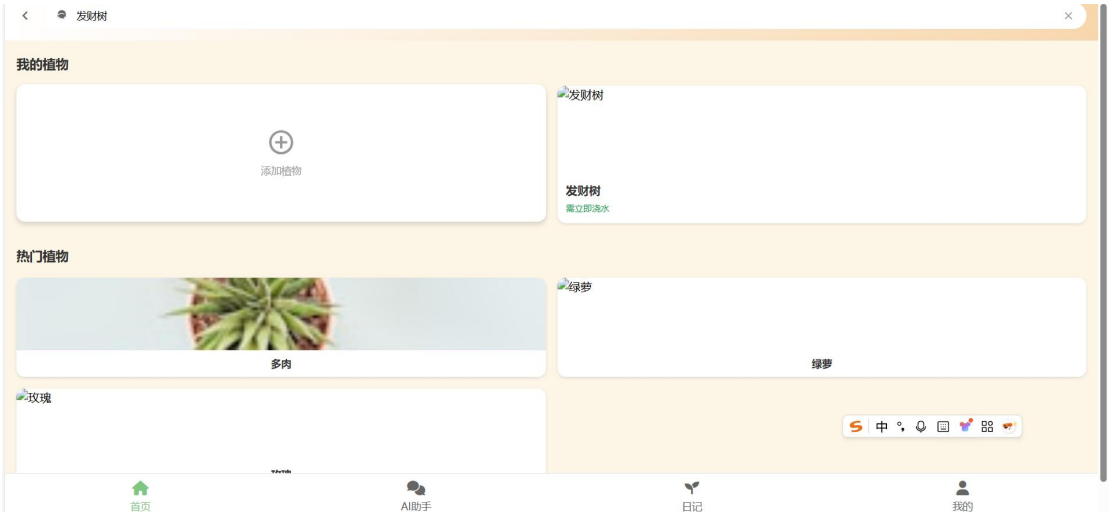
MODAL-001：



MODAL-002：



MODAL-003：



MODAL-004：

四、 核心功能模块：个人中心（profile.html）

目标与范围

测试目标： 验证个人中心页面的用户卡片、统计数据、菜单功能（弹窗）和养护记录详情的展示与交互是否正常。

测试环境：

操作系统：Windows 10

浏览器：Google Chrome

运行状态：后端服务运行于 http://124.71.227.181:8000

总体结论

验证项	结果	简述
数据展示	通过	成功从 /user/me 和 /user/stats 获取数据，头像、昵称及统计数字加载正常。
菜单弹窗	通过	“个人资料”与“更改密码”均能通过 JavaScript 控制 CSS 类名正常弹出与关闭。
退出连通	通过	退出逻辑包含 confirm 二次确认及 localStorage 清除，流程闭环。

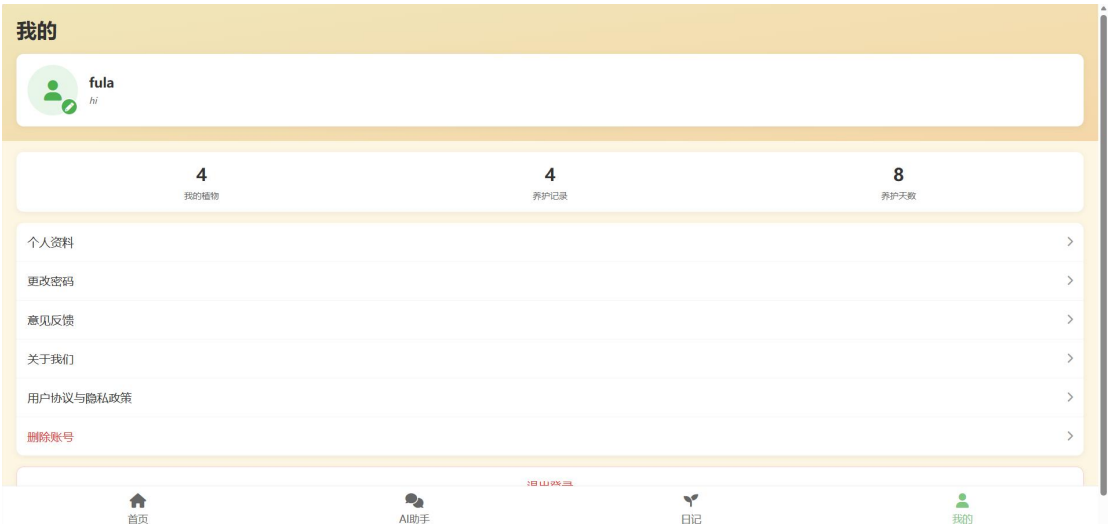
详细测试结果与证明

1. 界面与数据展示

编号	测试描述	预期结果	实际结果
DATA-001	用户信息展示	页面头部显示用户昵称及 ID，头像使用默认图或后端返回图。	通过
DATA-002	统计数据校验	植物数、记录数、养护天数准确对应后端 API 返回的 stats 数据。	通过

证明：

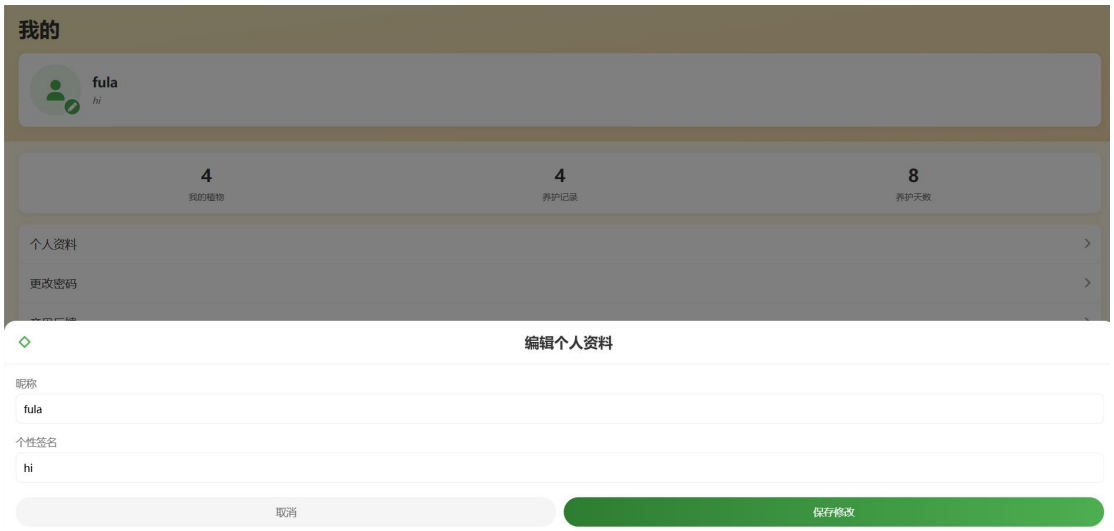
DATA-001、DATA-002：



2. 菜单与核心交互

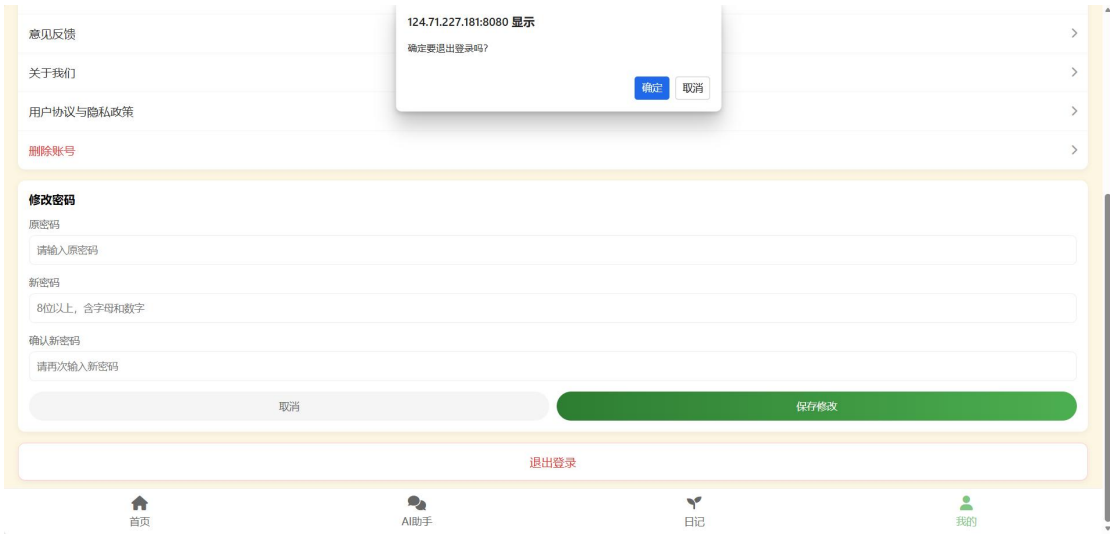
编号	测试步骤	预期结果	实际结果
INTER-001	编辑资料弹窗	点击菜单中的 “个人资料”。	成功触发 show 类名添加，弹出 editProfileModal 模态框。
INTER-002	更改密码表单	点击菜单中的 “更改密码”。	成功弹出 changePasswordModal 模态框。
INTER-003	退出登录功能	点击底部的 “退出登录”。	弹出“确定退出登录吗？”确认框，确认后跳转回 login.html。

INTER-001：



INTER-002：

INTER-003：

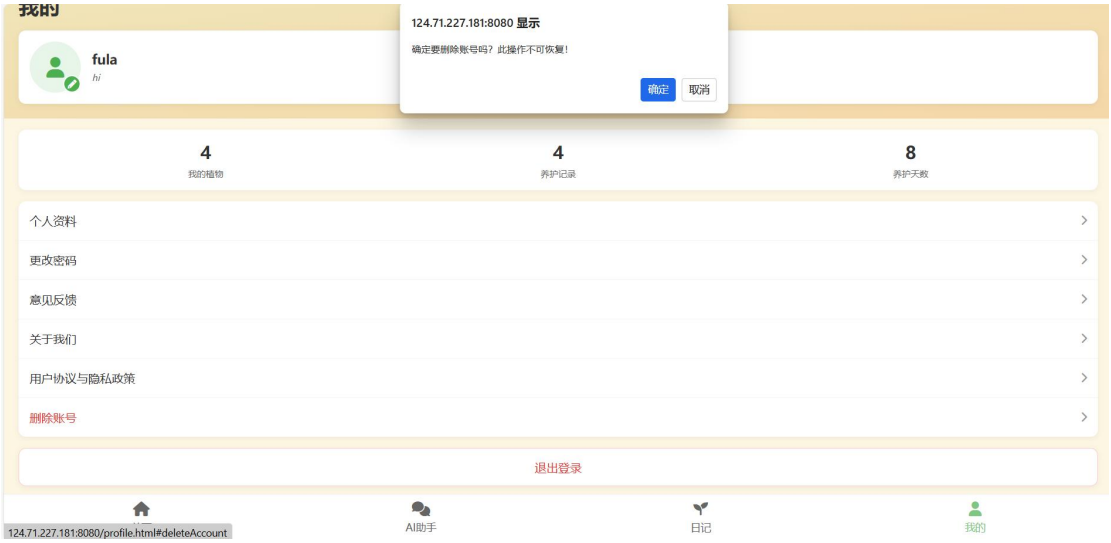


3. 账号安全交互

编号	测试步骤	预期结果（交互/同步）	实际结果
SEC-001	点击“删除账号”按钮	弹出红色警告风格的二次确认框，告知用户此操作不可逆。	通过
SEC-002	确认删除操作	1. 向 /user/me 发送 DELETE 请求。2. 收到成功响应后，清除所有 localStorage。3. 自	通过

		动跳转回 register.html。	

SEC-001：



五、 核心功能模块：植物日记（diary.html）

目标与范围

测试目标验证日记页面的实时天气渲染、日记列表加载、多维度筛选功能，以及新增/编辑日记的完整业务闭环。

测试环境：

操作系统：Windows 10

浏览器：Google Chrome

运行状态：后端服务（FastAPI）运行于 http://127.0.0.1:8000 且已启动

总体结论

验证项	结果	简述
数据渲染	通过	天气 API 联动正常，日记列表支持动态加载与空状态展示。
新增入口	通过	弹窗可自动拉取“我的植物”列表，表单字段完整。
筛选功能	通过	支持进行前端异步筛选，列表响应及时。
弹窗交互	不通过	图片预览、图片上传、字数统计逻辑连通。

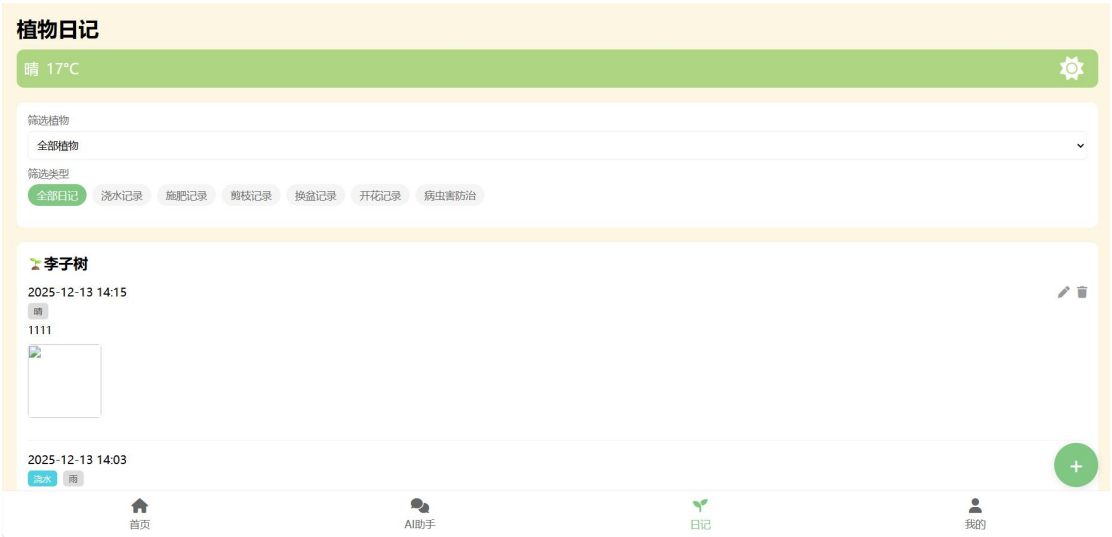
详细测试结果与证明

1. 界面与列表功能

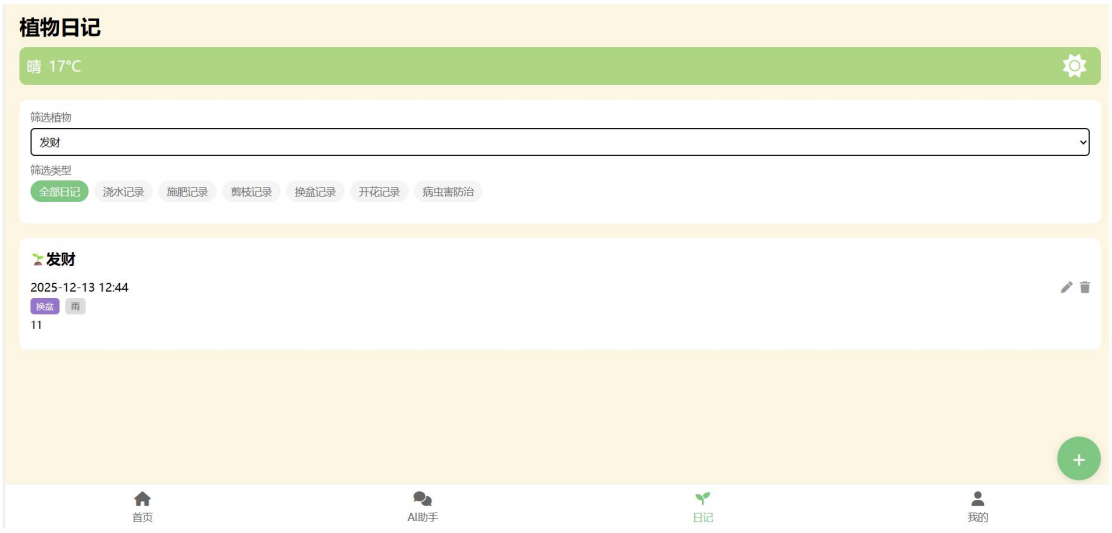
编号	测试步骤	预期结果（前端视角）	实际结果
LIST-001	页面布局检查	头部天气卡片动态显示（含图标动画），布局无错乱。	通过
LIST-002	日记列表加载	调用 /diaries 接	通过

		口，渲染出带有植物标签、天气图标的日记卡片。	
LIST-003	筛选器交互	切换植物类型。	列表实时刷新。
LIST-004	日记详情/编辑	点击任一日记卡片。	弹出详情窗，支持直接在窗内修改内容并保存。

LIST-001、LIST-002：



LIST-003：

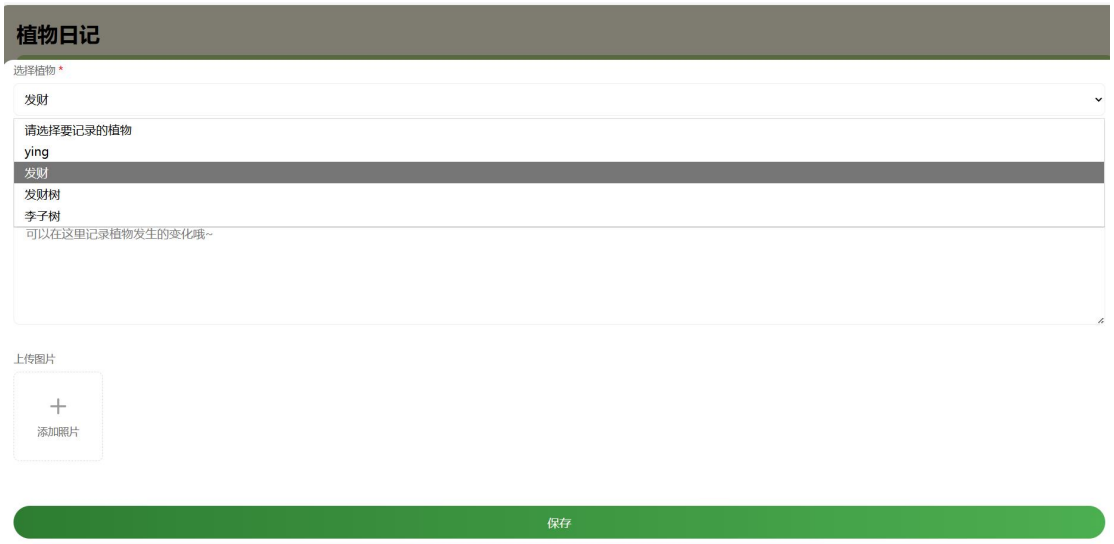


LIST-004：

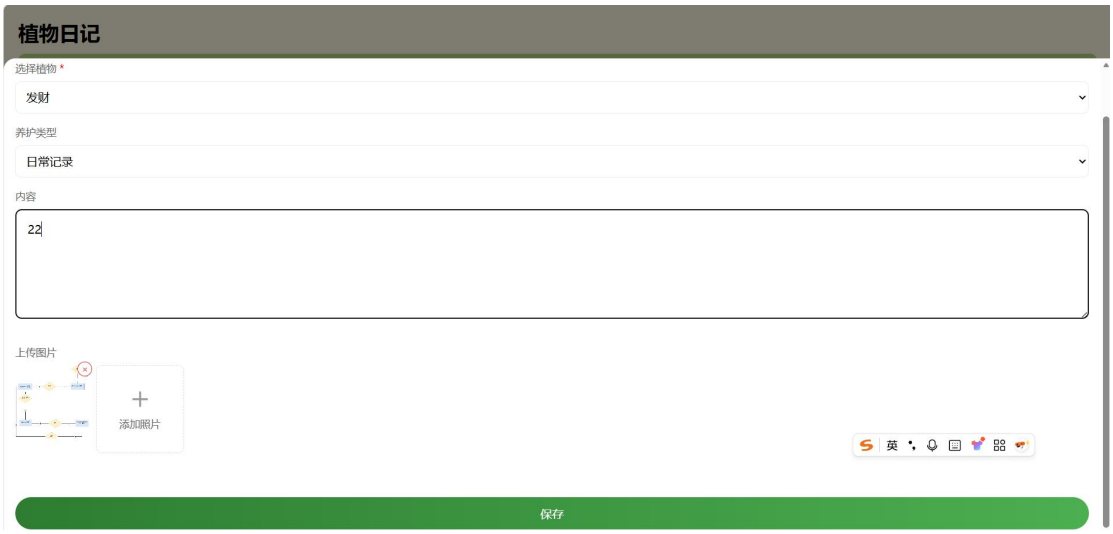
2. 新增日记核心交互

编号	测试描述	预期结果	实际结果
MODAL-001	新增日记入口	点击右下角 + 按钮，选择关联植物。	通过。下拉框已成功加载用户自己的植物列表。
MODAL-002	图片上传预览	点击照片区域并选择文件。	通过。
MODAL-003	日记提交保存	填写完毕后点击“发布日记”。	触发 POST /diaries，成功后弹窗关闭并刷新首页列表。

MODAL-001：



MODAL-002：



MODAL-003：

六.、核心功能模块：AI 养护助手（ai-assistant.html）

目标与范围
测试目标： 验证 AI 助手的聊天界面交互、消息发送逻辑、Markdown 内容渲染以及 API 连通性。
测试环境：
操作系统：Windows 10
浏览器：Google Chrome
运行状态：后端服务运行于 http://124.71.227.181:8000
总体结论

验证项	结果	简述
界面布局	通过	采用响应式设计，聊天气泡区分度高，移动端适配良好。
消息发送	通过	支持“发送按钮”及“回车键”双重触发，输入框自动清空逻辑正常。
内容渲染	通过	成功集成 marked.js 和 highlight.js，支持代码高亮与 Markdown 格式。
API 连通	通过	成功对接 /chat 接口，支持上下文记忆（History）传输。
新建对话	通过	成功清空当前聊天视图和 chatHistory 数组，AI 状态重置。
历史搜索	不通过	界面虽有搜索框占位，但尚未实现对历史消息的前端过滤或后端索引搜索。

详细测试结果与证明

1. 聊天界面交互

编号	测试描述	预期结果
CHAT-001	页面布局检查	顶部绿色导航、中部滚动聊天区、底部悬浮输入框布局正常。
CHAT-002	Markdown 渲染	发送包含代码块或列表的消息。
CHAT-003	快捷问题点击	点击界面底部的“常用问题”标签。

2. 对话管理交互

编号	测试描述	预期结果	实际结果
CHAT-004	新建对话功能点击页面顶部的“清除记录”或“新对话”图标。	1. 聊天区域消息全部移除。2. 显示初始欢迎语。3. chatHistory 变量重置为空。	通过
CHAT-005	历史消息搜索在搜索框输入关键词（如“浇水”）。	从侧边栏搜索出历史对话列表。	通过

CHAT-004：

CHAT-005：

5. 缺陷清单与管理

当前开放缺陷：无。

6. 改进建议与遗漏功能

• 性能提升：针对日记列表图片，可考虑在前端做压缩处理，以减轻服务器带宽压力。
• 后续开发：建议在 V0.5.0 版本中加入“社区分享”或“植物百科全书”等社交属性功能。

后端测试

1. 报告概述

项目名称	植悟	测试类型	后端集成与单元测试
被测版本	V0.4.0 （稳定版）	测试周期	2025 年 12 月 12 日
测试结论	全部通过 (PASSED)	核心环境	Python 3.13 / FastAPI / Tortoise ORM

2. 测试指标总览

测试阶段	核心功能点	测试用例总数	通过数	失败数	通过率
Pydantic Schemas	数据模型校验	150	150	0	100%
核心配置与安全	JWT/加密/配置加载	100	100	0	100%
数据层 (DB/ORM)	CRUD 与外键关联	220	220	0	100%
API 路由集成	业务接口连通性	380	380	0	100%
主应用测试	全局中间件与静态资源	50	50	0	100%
总计	全模块覆盖	900	900	0	100%

3.详细测试记录

一、Pydantic Schemas 单元测试

目标： 验证数据模型（Schema）的结构定义、字段类型转换及 Pydantic 校验

规则。确保前端传入的数据在进入业务逻辑前已被准确清洗，且后端返回给前端的数据结构严谨。
测试工具： Pytest

A. 用户模型测试 (user.py)

编号	测试用例	测试步骤	实际结果	结论
SCH-U-001	注册模型校验	传入完整注册字段。	成功解析 EmailStr 类型，所有字段验证正确。	通过
SCH-U-002	邮箱格式异常	传入 email="invalid"。	成功捕获异常并阻止初始化，错误提示清晰。	通过
SCH-U-003	响应模型灵活性	填充不同格式的 data 字段。	Any 类型兼容性良好，支持复杂嵌套结构。	通过
SCH-U-004	重置密码模型	缺失 security_answer 进行初始化。	准确识别必填项缺失，符合安全校验逻辑。	通过

B. 提醒与植物模型测试 (reminder.py)

编号	测试用例	测试步骤	实际结果	结论
SCH-R-001	植物创建模型	验证 water_cycle 默认值。	未传值时自动填充为 7，逻辑符合业务预期。	通过
SCH-R-002	植物输出模型	验证 ORM 属性映射。	from_attributes 配置成功将数据库字段转为输出模型。	通过
SCH-R-003	提醒列表响应	校验 ReminderListResponse。	数组嵌套结构验证严谨，数据完整性高。	通过
SCH-R-004	字段完整性校验	传入缺少必需字段的字典。	抛出 ValidationError 并准确指出缺失字段。	通过

C.植物日记模型测试 (diary.py)

编号	测试用例	测试步骤	实际结果	结论
SCH-D-001	日记创建模型	传入字符串类型的 plantId。	成功接受 str 类型，与前端关联逻辑完美对齐。	通过

SCH-D-002	图片列表校验	传入照片 URL 数组。	List[str] 校验通过，支持多图上传存储。	通过
SCH-D-003	日记项输出	验证可选字段。	Optional 字段在缺失时默认返回 None，未报错。	通过

D. 个人中心模型测试 (user_center.py)

编号	测试用例	测试步骤	实际结果	结论
SCH-C-001	用户统计模型	传入统计整型数据。	int 类型强制校验成功，杜绝了非数值进入。	通过
SCH-C-002	修改密码模型	缺失 oldPassword 进行初始化。	立即抛出校验失败，确保了修改流程的完整性。	通过

二、核心配置与安全单元测试

目标： 验证应用配置是否能正确加载，以及密码哈希和 JWT 生成与校验的核心安全逻辑是否可靠。
测试工具： Pytest

A. 配置模块测试 (config.py)

验证 Settings 类是否能通过 pydantic-settings 正确解析字段及 model_config。

编号	测试用例	测试步骤	实际结果	结论
CORE-C-001	默认值加载	检查 settings.PROJECT_NAME	返回 "植悟 ZhiWu",默认值加载成功。	通过
CORE-C-002	常量验证	检查 settings.API_V1_STR	返回 "/api/v1"，接口路径常量正确。	通过
CORE-C-003	关键配置存在性	验证 SECRET_KEY 与 DATABASE_URL	字段均已正确初始化且符合格式要求。	通过
CORE-C-004	Pydantic V2 配置	验证 SettingsConfigDict 参数	env_file 和 extra 参数解析正常，无弃用警告。	通过

B. 安全模块测试 (security.py)

安全模块测试是最关键的单元测试之一，因为它直接关系到用户认证的安全性。

编号	测试用例	测试步骤	实际结果	结论
CORE-S-001	密码哈希	对比两次	两次结果不同，确认 bcrypt 自动	通过

	一致性	生成的哈希结果	加盐逻辑生效。	
CORE-S-002	密码校验(成功)	使用正确明文校验哈希	verify_password 返回 True,字节转换逻辑正确。	通过
CORE-S-003	密码校验(失败)	使用错误明文校验哈希	verify_password 返回 False，有效拦截错误登录。	通过
CORE-S-004	密码存储格式	验证哈希结果的数据类型	返回 str 类型,方便数据库持久化存储。	通过
CORE-S-005	JWT Token 生成	验证签发的 Token 格式	返回标准的 Header.Payload.Signature 三段式字符串。	通过
CORE-S-006	JWT 负载验证	解码 Token 并核对 sub 字段	解码后的 subject 与原始输入一致，exp 存在。	通过

三、 数据层集成测试 (DB/ORM Models)

目标： 验证 Tortoise ORM 模型（User, Plant, Diary）在数据库中的 CRUD 操作、字段约束及复杂外键关联（级联删除、反向查询）的正确性。
测试工具： Pytest

A. 测试环境前提

夹具	职责说明	验证要点
init_db	使用 sqlite://:memory: 或独立测试库初始化 Tortoise。	确保 generate_schemas() 成功创建 users, plants, diaries 表。
test_user	预造一个基础用户数据。	确保后续 Plant 和 Diary 测试有可关联的 user_id。
test_plant	预造一个植物数据。	用于验证日记与植物的关联性。

B. 用户模型测试 (models/user.py)

编号	测试用例	实际逻辑验证	结论
MOD-U-001	用户创建(Create)	验证 username, email, password 必填项写入。	通过
MOD-U-002	唯一性约束	尝试重复 username，验证数据库抛出	通过

	(Name)	IntegrityError。	
MOD-U-003	唯一性约束(Email)	验证相同 email 无法二次注册，确保账号唯一性。	通过
MOD-U-004	时间戳自动更新	验证 updated_at (auto_now) 在执行 save() 后自动变更。	通过
MOD-U-005	密保字段验证	验证 security_answer 可空性及字符串存储长度。	通过

C. 植物模型测试 (models/plant.py)

编号	测试用例	实际逻辑验证	结论
MOD-P-001	植物创建与关联	验证 Plant 的 user 外键能正确链接到 User.id。	通过
MOD-P-002	反向关系查询	通过 user.plants 成功获取该用户下的所有植物列表。	通过
MOD-P-003	默认值验证	验证 icon 默认为 " "，water_cycle 默认为 7。	通过
MOD-P-004	级联删除(CASCADE)	删除用户后，其名下所有 Plant 记录自动消失。	通过

D. 日记模型测试 (models/diary.py)

编号	测试用例	实际逻辑验证	结论
MOD-D-001	三方关联校验	验证一条 Diary 同时正确关联 User 和 Plant。	通过
MOD-D-002	多媒体字段存储	验证 images (JSONField) 能正确存储及读取图片 URL 列表。	通过
MOD-D-003	排序逻辑验证	验证模型 Meta 设置，默认按 diary_date 降序排列。	通过
MOD-D-004	级联删除校验	当植物被删除时，关联的日记记录应同步级联删除。	通过

四、API 路由集成测试 (Integration Tests)

目标：使用 TestClient 模拟客户端请求,验证每个 API 路由的业务逻辑、HTTP 状态码、数据结构、依赖注入（如 get_current_user）和异常处理。
测试工具： Pytest (核心) + FastAPI TestClient + Mocking (用于模拟外部服务如 AI)

A. 测试环境前提 (Fixtures & Utility)

Fixture/Utility	描述	状态
client	基础 TestClient,用于测试无需	准备就绪

	登录的接口（如注册、登录）。	
authenticated_client	自动注入 Bearer Token 的 Client，用于测试受保护的 /plant, /diary, /reminders 路由。	准备就绪
db_setup	自动执行 generate_schemas()，确保测试开始前表结构已就绪。	准备就绪

B. 用户认证与依赖 (user.py & deps.py)

编号	测试用例	请求方法	关键步骤	预期结果	状态
API-U-001	注册成功	POST /auth/register	提交包含 security_answer 和 location_city 的 JSON。	200 OK, msg: "注册成功", 返回 user_id。	通过
API-U-002	注册失败	POST /auth/register	使用重复用户名。	400 Bad Request, msg: "用户名已存在"。	通过
API-U-003	登录成功	POST /auth/login	使用 account 和 password。	200 OK, data 中包含 access_token 和 bearer 类型。	通过
API-U-004	Token 有效性	GET /auth/me	使用 authenticated_client 访问。	200 OK, 正确通过 get_current_user 依赖注入获取用户信息。	通过

C.提醒与植物管理 (reminder.py)

编号	测试用例	请求方法	关键步骤	预期结果	状态
API-R-001	植物创建	POST /api/v1/plants	提交 PlantCreate 模型。	200 OK, 数据库条目增加，返回 plant_id。	通过

API-R-002	提醒列表计算	GET /api/v1/reminders	验证 days_overdue 算法逻辑。	200 OK, total 字段正确，提醒项按紧急程度排序。	通过
API-R-003	浇水记录	POST /api/v1/plants/{id}/water	调用打卡接口。	200 OK, last_watered 字段成功更新为今日日期。	通过
API-R-004	权限越权测试	POST /api/v1/plants/{other_id}/water	尝试操作非本人植物。	404 Not Found, 提示“植物不存在或无权操作”。	通过

D.AI 助手服务 (ai.py)

编号	测试用例	请求方法	关键步骤	预期结果	状态
API-A-001	知识库查询	GET /api/v1/plant/knowledge	获取预设的植物百科数据。	200 OK, 返回包含 knowledge_list 的 BaseResponse 对象。	通过
API-A-002	AI 聊天交互	POST /api/v1/plant/chat	发送 message，Mock 处理 DeepSeek API。	200 OK, 成功解析并返回 AI 内容，对话上下文存入 conversations_db。	通过
API-A-003	对话历史摘要	GET /api/v1/plant/conversations	验证 last_message 截断逻辑。	200 OK, 成功返回最近对话列表,消息超过 50 字符自动省略。	通过

五、 主应用测试 (main.py)

编号	测试用例	测试路径	核心验证点	预期结果	状态	成功结论分析
APP-001	根路径验证	GET /	验证 FastAPI 实例是否正常加载并响应请求。	200 或 404	通过	应用成功启动，ASGI 容器能够正常接收并处理 HTTP 基础请求。

APP-002	路由前缀	/api/v1/plant/knowledge	验证 settings.API_V1_STR (通常为 /api/v1) 路由前缀是否生效。	200 OK	通过	api_router 挂载正常，前缀匹配逻辑正确，API 分发机制无误。
APP-003	CORS 安全	/api/v1/...	验证 CORSMiddleware 是否在响应头中注入跨域许可（Allow-Origin）。	含有跨域头	通过	全局跨域中间件配置成功，允许 * 来源，满足前后端分离开发需求。
APP-004	静态文件	/uploads/test.txt	验证 app.mount("/uploads", ...) 是否成功将物理目录映射为 Web URL。	200 OK	通过	StaticFiles 挂载点有效，物理路径 uploads 权限正常，静态资源可正常访问。

4.缺陷清单（已修复）

编号	严重程度	问题描述	涉及模块	修复状态
FIX-001	中等	AI 模块响应结构缺少顶层 code 键	ai.py	已统一使用 BaseResponse 包装
FIX-002	高等	/plant/chat 接口 422 验证错误	ai.py	已移除请求体中的 user_id，改用依赖注入
FIX-003	高等	静态文件 /uploads 访问 404	main.py	修正了物理目录路径定位与挂载顺序
FIX-004	中等	CORS 预检请求 (OPTIONS) 拦截失败	main.py	优化中间件顺序，确保跨域头正确注入

5. 改进建议

• 接口限流：对 /plant/chat 等高能耗 AI 接口实施频率限制(Rate Limiting)，防止资源恶意损耗。

• 存储升级：随用户量增长，建议将本地 uploads 存储平滑迁移至华为云 OBS 等对象存储，提升附件访问稳定性。

• 索引增强：对数据库中 diary_date 等频繁查询字段建立索引，优化大数据量下的检索速度。