

玉米病虫害智能决策系统







基于 Dify AI 工作流的玉米病虫害防治策略推荐系统，提供条件选择和智能问答两种交互模式。

功能特色

双系统模式

- 条件选择系统:** 通过选择病虫害类型、生长期、地点等条件获取精准防治建议
- 智能问答系统:** 支持自然语言对话，可追问和深入讨论病虫害相关问题

核心功能

-  **用户登录系统:** 支持记住密码功能
-  **多会话管理:** 创建、切换、删除对话会话
-  **历史记录:** 自动保存查询历史和对话记录
-  **批量删除:** 支持单个删除、批量删除、清空全部历史
-  **系统切换:** 在两种模式间无缝切换
-  **响应式设计:** 适配桌面和移动设备

快速开始

环境要求

- Node.js 18+
- npm 或 yarn

安装步骤

1. 克隆项目

```
git clone https://github.com/your-username/corn-pest-decision-system.git
cd corn-pest-decision-system
```

2. 安装依赖

```
npm install
# 或
yarn install
```

3. 配置 Dify API Key

 **重要:** 需要配置您的 Dify API Key 才能正常使用系统

在以下文件中找到并替换 API Key:

- app/api/dify/route.ts (第6行)
- app/api/dify-blocking/route.ts (第6行)
- app/api/dify-query/route.ts (第15行)

将这行代码:

```
const API_KEY = "app-5m7qs7XWIS2p4DiG1LR6l8QF"
```

或者:

```
Authorization: "Bearer app-5m7qs7XWIS2p4DiG1LR6l8QF",
```

替换为您的 Dify API Key:

```
const API_KEY = "your-dify-api-key-here"
```

如何获取 Dify API Key:

1. 登录 [Dify 控制台](#)
2. 创建或选择一个应用
3. 在应用设置中找到 API 密钥
4. 复制 API Key (格式通常为 app-xxxxxxxxxx)
5. 启动开发服务器

```
npm run dev
# 或
yarn dev
```

5. 访问应用

打开浏览器访问 <http://localhost:3000>

使用指南

登录系统

- 可使用任意用户名和密码登录（演示用途）
- 支持"记住密码"功能
- 选择要使用的系统类型（条件选择或智能问答）

条件选择系统

1. 选择病虫害类型（玉米斑病、玉米锈病等）
2. 选择生长期（苗期、拔节期等）
3. 输入地点信息
4. 点击"获取防治建议"
5. 查看AI生成的专业防治策略

智能问答系统

1. 在输入框中输入问题
2. 支持追问和深入讨论
3. 系统会保持对话上下文
4. 可创建多个对话会话

历史管理

- **查看历史:** 在右侧面板查看所有历史记录
- **重新加载:** 点击历史记录重新加载之前的查询
- **删除记录:**
 - 单个删除：悬停显示删除按钮
 - 批量删除：点击"管理"进入选择模式
 - 清空全部：一键清空所有历史

技术栈

- **前端框架:** Next.js 14 (App Router)

- **UI 组件:** shadcn/ui + Tailwind CSS
- **状态管理:** React Hooks
- **数据存储:** localStorage (客户端)
- **AI 服务:** Dify API
- **开发语言:** TypeScript

项目结构

```
├─ app/
│   └─ api/                # API 路由
│       └─ dify/           # Dify 流式 API
│       └─ dify-blocking/  # Dify 阻塞式 API
│       └─ dify-query/     # Dify 查询 API
│       └─ weather/        # 天气 API (模拟)
│   └─ globals.css         # 全局样式
│   └─ layout.tsx          # 根布局
│   └─ page.tsx            # 主页面组件
├─ components/ui/         # UI 组件库
├─ public/                # 静态资源
└─ README.md
```

配置说明

API 配置

系统使用 Dify 的 `/v1/chat-messages` 端点，支持两种模式：

- **streaming:** 流式响应，实时显示
- **blocking:** 阻塞式响应，等待完整结果

环境变量（可选）

如果需要使用环境变量管理 API Key，可以创建 `.env.local` 文件：

```
DIFY_API_KEY=your-dify-api-key-here
```

然后在代码中使用：

```
const API_KEY = process.env.DIFY_API_KEY || "fallback-key"
```

自定义配置

修改病虫害类型

在 `app/page.tsx` 中找到 `SelectContent` 部分，可以添加或修改病虫害类型选项：

```
<SelectContent className="bg-slate-800 border-slate-600">
  <SelectItem value="玉米斑病">玉米斑病</SelectItem>
  <SelectItem value="玉米锈病">玉米锈病</SelectItem>
  // 添加更多选项...
</SelectContent>
```

修改生长期选项

同样在 `SelectContent` 中修改生长期选项：

```
<SelectItem value="苗期">苗期</SelectItem>
<SelectItem value="拔节期">拔节期</SelectItem>
// 添加更多选项...
```

部署

Vercel 部署（推荐）

1. 将代码推送到 GitHub
2. 在 [Vercel](#) 中导入项目
3. 配置环境变量（如果使用）
4. 部署完成

其他平台

项目是标准的 Next.js 应用，可以部署到任何支持 Node.js 的平台：

- Netlify
- Railway

- Heroku
- 自建服务器

故障排除

API 调用失败

1. 检查 Dify API Key 是否正确配置
2. 确认 Dify 应用是否正常运行
3. 查看浏览器控制台的错误信息

历史记录丢失

- 历史记录保存在浏览器的 localStorage 中
- 清除浏览器数据会导致历史记录丢失
- 可以考虑集成数据库进行持久化存储

样式问题

- 确保 Tailwind CSS 正确安装
- 检查 `globals.css` 是否正确导入

贡献

欢迎提交 Issue 和 Pull Request!

1. Fork 项目
2. 创建功能分支 (`git checkout -b feature/AmazingFeature`)
3. 提交更改 (`git commit -m 'Add some AmazingFeature'`)
4. 推送到分支 (`git push origin feature/AmazingFeature`)
5. 开启 Pull Request

许可证

本项目采用 MIT 许可证 - 查看 [LICENSE](#) 文件了解详情

致谢

- [Dify](#) - 提供强大的 AI workflow 平台
- [Next.js](#) - React 全栈框架
- [shadcn/ui](#) - 优秀的 UI 组件库
- [Tailwind CSS](#) - 实用的 CSS 框架