

- 郎家园枣园农业监控系统
 - 系统要求
 - 硬件要求
 - 软件要求
 - 快速开始
 - 1. 环境准备
 - Windows系统
 - macOS系统
 - Ubuntu/Linux系统
 - 2. 系统安装
 - 系统功能
 - 主要模块
 - 配置说明
 - 基础配置
 - 端口配置
 - 生产环境配置
 - 常见问题
 - 1. 端口占用
 - 2. 依赖安装失败
 - 3. 数据库错误
 - 4. 权限问题 (Linux/macOS)
 - 5. 虚拟环境问题
 - 系统维护
 - 数据备份
 - 日志管理
 - 性能监控
 - 系统停止
 - 正常停止
 - 强制停止
 - 技术栈
 - 项目结构
 - 更新日志
 - v1.0.0 (2024-07-10)
 - 支持与反馈
 - 许可证
 - 贡献指南

郎家园枣园农业监控系统

系统要求

硬件要求

- CPU:** 双核1.5GHz以上
- 内存:** 4GB RAM以上
- 存储:** 2GB可用空间
- 网络:** 稳定的网络连接

软件要求

- 操作系统:** Windows 10/11、macOS 10.14+、Ubuntu 18.04+
- Python:** 3.8或更高版本
- 浏览器:** Chrome 70+、Firefox 65+、Safari 12+、Edge 79+

快速开始

1. 环境准备

Windows系统

- 下载并安装Python 3.8+: <https://www.python.org/downloads/>
- 安装时勾选"Add Python to PATH"
- 验证安装:

```
python --version  
pip --version
```

macOS系统

```
# 安装Homebrew (如果没有)
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# 安装Python
brew install python
```

Ubuntu/Linux系统

```
sudo apt update
sudo apt install python3 python3-pip python3-venv
```

2. 系统安装

1. 解压系统文件

```
unzip 郎家园枣园农业监控系统_完整版.zip
cd agricultural_monitoring_system
```

2. 创建虚拟环境（推荐）

```
# Windows
python -m venv venv
venv\Scripts\activate

# macOS/Linux
python3 -m venv venv
source venv/bin/activate
```

3. 安装依赖

```
pip install -r requirements.txt
```

4. 初始化数据库

```
python init_database.py
```

5. 启动系统

```
python app.py
```

6. 访问系统

打开浏览器访问：<http://localhost:8080>

系统功能

主要模块

模块	功能	访问地址
首页	系统概览和快速导航	/
数据监控	实时环境数据监控	/monitoring
病虫害预测	AI智能预测分析	/predictions
预警系统	多渠道预警通知	/warnings
绿色防控	防治方案推荐	/pest-control
市场分析	价格趋势分析	/market-analysis
产品追溯	生产全流程追溯	/traceability
系统设置	配置管理	/settings

配置说明

基础配置

编辑 `config.py` 文件进行系统配置：

```
# 数据库配置
DATABASE_URL = 'sqlite:///agriculture.db'
```

```
# 邮件配置
```

```
MAIL_SERVER = 'smtp.gmail.com'
MAIL_PORT = 587
MAIL_USE_TLS = True
MAIL_USERNAME = 'your_email@gmail.com'
MAIL_PASSWORD = 'your_app_password'

# 短信配置 (Twilio)
TWILIO_ACCOUNT_SID = 'your_account_sid'
TWILIO_AUTH_TOKEN = 'your_auth_token'
TWILIO_PHONE_NUMBER = 'your_twilio_number'

# 系统配置
SECRET_KEY = 'your_secret_key'
DEBUG = True
```

端口配置

修改 `app.py` 文件中的端口设置：

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)
```

生产环境配置

1. 安装生产服务器

```
pip install gunicorn
```

2. 启动生产服务器

```
gunicorn -w 4 -b 0.0.0.0:8080 app:app
```

3. 配置反向代理（Nginx示例）

```
server {
    listen 80;
    server_name your_domain.com;

    location / {
        proxy_pass http://127.0.0.1:8080;
```

```
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```

常见问题

1. 端口占用

```
# 查看端口占用  
netstat -ano | findstr :8080 # Windows  
lsof -i :8080                # macOS/Linux  
  
# 终止进程  
taskkill /F /PID <进程ID>    # Windows  
kill -9 <进程ID>             # macOS/Linux
```

2. 依赖安装失败

```
# 升级pip  
pip install --upgrade pip  
  
# 清理缓存  
pip cache purge  
pip install -r requirements.txt
```

3. 数据库错误

```
# 重新初始化数据库  
rm agriculture.db  
python init_database.py
```

4. 权限问题（Linux/macOS）

```
chmod +x app.py
chmod +x init_database.py
```

5. 虚拟环境问题

```
# 重新创建虚拟环境
rm -rf venv
python -m venv venv
source venv/bin/activate # Linux/macOS
venv\Scripts\activate   # Windows
```

系统维护

数据备份

```
# 备份数据库
cp agriculture.db agriculture_backup_$(date +%Y%m%d).db

# 备份配置文件
cp config.py config_backup.py
```

日志管理

```
# 查看系统日志
tail -f logs/app.log

# 清理旧日志
find logs/ -name "*.log" -mtime +30 -delete
```

性能监控

```
# 查看系统资源使用
top
```

```
htop
```

```
# 查看Python进程
```

```
ps aux | grep python
```

系统停止

正常停止

在运行终端按 **Ctrl+C**

强制停止

```
# Windows
```

```
taskkill /F /IM python.exe
```

```
# macOS/Linux
```

```
pkill -f "python.*app.py"
```

技术栈

- **后端**: Python 3.8+, Flask, SQLAlchemy
- **前端**: HTML5, CSS3, JavaScript, Bootstrap
- **数据库**: SQLite
- **机器学习**: scikit-learn, pandas, numpy
- **图表**: Chart.js, Plotly
- **通信**: SMTP, Twilio
- **其他**: Gunicorn, Nginx

项目结构

```
agricultural_monitoring_system/
```

```
├─ app.py
```

```
# 主应用程序
```

```
├─ config.py
```

```
# 配置文件
```



```
├── init_database.py      # 数据库初始化
├── requirements.txt      # 依赖列表
├── README.md            # 项目说明
├── models/              # 数据模型
│   └── database.py
├── modules/             # 功能模块
│   ├── data_collection.py
│   ├── data_preprocessing.py
│   ├── ml_models.py
│   ├── warning_system.py
│   ├── pest_control.py
│   ├── market_analysis.py
│   └── traceability.py
├── static/              # 静态文件
│   ├── css/
│   ├── js/
│   └── images/
├── templates/           # 页面模板
└── logs/                # 日志文件
```

更新日志

v1.0.0 (2024-07-10)

- 初始版本发布
- 完整的监控系统功能
- 响应式界面设计
- 多模块集成

支持与反馈

如有问题或建议，请联系：

- 邮箱: 2842888888@qq.com
- 电话: 15887122055
- 微信: 15887122055

许可证

本项目采用 MIT 许可证。详情请参见 [LICENSE](#) 文件。

贡献指南

欢迎贡献代码和建议！请遵循以下步骤：

1. Fork 本项目
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. 创建 Pull Request

郎家园枣园农业监控系统 - 让农业生产更智能、更高效！