

# 闲鱼二手交易平台

一个基于Tornado框架开发的二手闲置物品交易平台，支持商品发布、实时聊天、订单管理、商品评价等功能。

## 功能特性



### 商品管理

- 商品发布和编辑
- 商品图片上传
- 商品分类和搜索
- 商品状态管理（在售、已售完等）
- 库存管理



### 实时聊天

- WebSocket实时通信
- 好友系统
- 私人聊天室
- 消息历史记录
- 未读消息提醒
- 离线消息推送



### 订单系统

- 订单创建和管理
- 订单状态跟踪
- 买家和卖家视角

- 订单历史查询

## ★ 评价系统

- 商品评价和评分
- 评价展示和统计
- 购买后评价权限

## 👤 用户系统

- 用户注册和登录
- 密码重置功能
- 用户个人主页
- 用户资料管理

## 技术栈

- **后端:** Python 3.11 + Tornado
- **数据库:** MySQL (用户和商品数据) + MongoDB (聊天消息)
- **缓存:** Redis
- **前端:** HTML + CSS + JavaScript
- **部署:** Docker + Nginx + Supervisor

# 项目结构

```
MVC/
├─ app.py                # 应用入口文件
├─ config.ini            # 配置文件
├─ requirements.txt      # Python依赖
├─ Dockerfile            # Docker配置
├─ docker-compose.yml    # Docker Compose配置
├─ nginx.conf            # Nginx配置
├─ supervisord.conf      # Supervisor配置
├─ start.sh              # 启动脚本
├─ README.md             # 项目文档
├─ base/
│   └─ base.py           # 数据库基础配置
├─ models/               # 数据模型
│   └─ user.py           # 用户模型
│   └─ product.py        # 商品模型
│   └─ comment.py        # 评价模型
│   └─ order.py          # 订单模型
│   └─ friendship.py     # 好友关系模型
├─ controllers/          # 控制器
│   └─ auth_controller.py # 认证控制器
│   └─ product_controller.py # 商品控制器
│   └─ chat_controller.py # 聊天控制器
│   └─ comment_controller.py # 评价控制器
│   └─ order_controller.py # 订单控制器
│   └─ search_controller.py # 搜索控制器
├─ templates/            # HTML模板
├─ mystatics/            # 静态资源
│   └─ css/              # 样式文件
│   └─ js/               # JavaScript文件
│   └─ images/           # 上传的图片
└─ utils/                # 工具函数
```

# 快速开始

## 方式一：Docker部署（推荐）

### 1. 克隆项目

```
bash git clone <repository-url> cd MVC
```

### 2. 修改配置

编辑 `docker-compose.yml` 文件，修改数据库密码等配置：

```
yaml      environment:      MYSQL_ROOT_PASSWORD:      your_secure_password
MYSQL_PASSWORD: your_secure_password
```

### 1. 启动服务

```
bash docker-compose up -d
```

### 2. 访问应用

打开浏览器访问 `http://localhost`

## 方式二：手动部署

### 环境要求

- Python 3.11+
- MySQL 8.0+
- MongoDB 6.0+
- Redis 7.0+
- Nginx

### 安装步骤

#### 1. 安装Python依赖

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

#### 2. 配置数据库

修改 `config.ini` 文件中的数据库连接信息：

```
```.ini
[mysql]
user = your_mysql_user
password = your_mysql_password
host = localhost
port = 3306
database = xianyu_db
charset = utf8mb4

[mongodb]
host = localhost
port = 27017
database = chat_db
```.
```

### 1. 创建数据库

```
sql CREATE DATABASE xianyu_db CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;
```

### 2. 启动应用

```
bash python app.py
```

### 3. 配置Nginx

将 `nginx.conf` 复制到Nginx配置目录，并重启Nginx。

## 微信小程序版本

项目已支持转换为微信小程序版本，详细转换步骤：

### 1. 准备工作

- 注册微信小程序账号
- 下载微信开发者工具
- 准备小程序AppID

## 2. 创建小程序项目结构

```
miniprogram/
├─ app.js
├─ app.json
├─ app.wxss
├─ pages/
│   ├─ index/          # 首页
│   ├─ product/        # 商品页面
│   ├─ chat/           # 聊天页面
│   ├─ order/          # 订单页面
│   ├─ profile/        # 个人中心
│   └─ login/          # 登录页面
├─ components/        # 组件
├─ utils/              # 工具函数
└─ images/             # 图片资源
```

## 3. API适配

小程序版本需要使用微信小程序的API：

- `wx.request()` 替代 `fetch()`
- `wx.connectSocket()` 实现WebSocket连接
- `wx.uploadFile()` 处理文件上传
- `wx.login()` 微信登录

## 4. 页面转换

将现有的HTML页面转换为小程序页面：

- `.html` → `.wxml`
- `.css` → `.wxss`
- `.js` → `.js` (适配小程序语法)

# API接口文档

## 认证接口

- `POST /login` - 用户登录
- `POST /register` - 用户注册
- `POST /forgot_password` - 忘记密码
- `POST /reset_password` - 重置密码

## 商品接口

- `GET /product_list` - 获取商品列表
- `GET /product/detail/{id}` - 获取商品详情
- `POST /product/upload` - 发布商品

## 订单接口

- `GET /orders` - 获取订单列表
- `POST /orders` - 创建订单
- `PUT /orders/{id}` - 更新订单状态
- `DELETE /orders/{id}` - 取消订单

## 评价接口

- `GET /api/comments/{product_id}` - 获取商品评价
- `POST /api/comments` - 发布评价
- `GET /api/product/{id}/rating` - 获取商品评分统计

## 聊天接口

- `GET /chat_room` - 聊天室页面

- `WebSocket /ws/chat_room` - WebSocket连接
- `POST /api/send_message` - 发送消息
- `GET /api/messages` - 获取消息历史

## 部署注意事项

### 生产环境配置

#### 1. 修改密钥

```
python settings = { 'cookie_secret': 'your_random_secret_key',  
'xsrp_cookies': True }
```

#### 2. 启用HTTPS

- 获取SSL证书
- 配置Nginx HTTPS
- 更新小程序域名配置

#### 3. 数据库优化

- 设置合适的连接池大小
- 启用慢查询日志
- 定期备份数据

#### 4. 监控和日志

- 配置日志轮转
- 设置监控告警
- 性能监控

### 安全注意事项

- 定期更新依赖包
- 设置防火墙规则
- 启用访问日志分析
- 实施速率限制



# 开发指南

## 添加新功能

1. 在 `models/` 中定义数据模型
2. 在 `controllers/` 中实现业务逻辑
3. 在 `templates/` 中创建页面模板
4. 在 `app.py` 中添加路由

## 数据库迁移

```
# 在模型文件中修改后执行
from base.base import Base, engine
Base.metadata.create_all(engine)
```

# 故障排除

## 常见问题

1. **数据库连接失败**
  - 检查配置文件中的连接信息
  - 确认数据库服务是否启动
2. **WebSocket连接失败**
  - 检查Nginx配置中的WebSocket代理
  - 确认防火墙设置
3. **图片上传失败**
  - 检查目录权限
  - 确认文件大小限制

# 贡献指南

1. Fork 项目
2. 创建功能分支
3. 提交更改
4. 推送到分支
5. 创建 Pull Request

# 许可证

MIT License

# 联系方式

如有问题或建议，请通过以下方式联系：

- GitHub Issues
- Email: [your-email@example.com](mailto:your-email@example.com)

---

感谢使用闲鱼二手交易平台！