

- 健身房会员系统配置与运维文档
 - 一、配置管理
 - 1.1 文件结构说明
 - 1.2 环境配置
 - 1.3 依赖管理
 - 二、部署流程
 - 2.1 PythonAnywhere 部署步骤
 - 2.2 详细部署流程
 - 2.3 数据库初始化说明
 - 三、版本控制策略
 - 3.1 Git分支管理
 - 3.2 文件版本管理规范
 - 四、持续集成与部署
 - 4.1 CI/CD 流水线
 - 4.2 自动化测试用例缺失情况
 - 五、运维计划
 - 5.1 定期查看日志
 - 5.2 维持托管服务
 - 5.3 安全措施

健身房会员系统配置与运维文档

一、配置管理

1.1 文件结构说明

```
/code
├── flask_app.py      # Flask主应用入口
├── gym.py            # 核心业务逻辑
├── mysite.wsgi       # WSGI部署入口文件
├── /static
│   ├── image.png    # 静态资源文件
├── /templates       # Jinja2模板
│   ├── login.html   # 登录界面
│   ├── logout.html  # 登出界面
│   ├── add_member.html # 添加会员界面
│   ├── add_courses.html # 添加课程界面
│   └── courses.html  # 课程管理/查询界面
```

```
| |— edit_courses.html # 编辑课程界面
| |— edit_member.html # 编辑会员界面
| |— edit_profile.html # 编辑个人资料界面
| |— profile.html      # 个人资料界面
| |— members.html      # 会员管理界面
| |— register.html     # 注册界面
| |— view_reservations_by_member.html # 会员课程信息界面
|— /database
|   — gym_management.sql # 数据库初始化脚本
|— requirements.txt
```

1.2 环境配置

开发环境配置（.env 文件）

```
# 开发环境变量
SECRET_KEY = "dev_secret_key_123"
DB_PASSWORD = "dev_db_password"
DEBUG = True
```

生产环境配置（PythonAnywhere）

```
# flask_app.py
app.secret_key = os.environ.get('SECRET_KEY') # 从环境变量获取密钥
app.config['DEBUG'] = False # 生产环境关闭调试

DB_CONFIG = {
    "user": "demonnn7",
    "password": os.environ.get('DB_PASSWORD'), # 安全获取密码
    "host": "demonnn7.mysql.pythonanywhere-services.com",
    "port": "3306",
    "database": "demonnn7$gym_system",
    "raise_on_warnings": True,
    "time_zone": "+08:00"
}
```

1.3 依赖管理

requirement.txt

```
Flask==2.0.3
mysql-connector-python==8.0.28
```

```
python-dotenv==0.20.0
```

安装依赖：

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

二、部署流程

2.1 PythonAnywhere 部署步骤

flowchart TD

A[准备代码] --> B[配置PythonAnywhere]

B --> C[设置虚拟环境]

C --> D[配置WSGI文件]

D --> E[设置环境变量]

E --> F[初始化数据库]

F --> G[重启应用]

2.2 详细部署流程

1. 进入网站

进入<https://www.pythonanywhere.com/login/>网站登录

2. 准备代码

```
cd home/demonnn7/code
pip install -r requirements.txt
```

3. 配置PythonAnywhere

- 创建Web应用
- Python版本选择3.9
- 虚拟环境路径：[/home/demonnn7/code/gym](#)

4. 配置WSGI文件

```
import os
import sys
from os.path import dirname, abspath

# 设置环境变量
os.environ['SECRET_KEY'] = '32b73c5e1c3f4a9a8e5d5e2c5a8b3d7'

# 项目根目录
project_path = '/home/demonnn7/code'

# 添加项目路径到系统路径
if project_path not in sys.path:
    sys.path.insert(0, project_path)

# 导入 Flask 应用实例
from gym import app as application
```

5. 设置环境变量

- 在Web应用配置页面添加：
 - SECRET_KEY
 - DB_PASSWORD

6. 初始化数据库

```
mysql -u demonnn7 -p gym_system < database/gym_management.sql
```

7. 重启应用

- 在PythonAnywhere控制台点击"Reload"按钮

2.3 数据库初始化说明

`gym_management.sql` 包含：

- 创建所有表结构
- 初始管理员账号（用户名/密码：sysu/8888）
- 基础数据样例

三、版本控制策略

3.1 Git分支管理

```
graph LR
    main[主分支] -->|稳定版本| prod[生产环境]
    dev[开发分支] -->|新功能| feature[功能分支]
    feature -->|测试完成| dev
    dev -->|测试通过| main
    main -->|紧急修复| hotfix[热修复]
    hotfix -->|修复验证| main
```

3.2 文件版本管理规范

1. 代码文件：

- flask_app.py - Flask应用配置
- gym.py - 核心业务逻辑
- 提交时需通过基础测试

2. 模板文件：

- 位于 /templates目录
- 命名需明确功能（如 edit_profile.html）
- 修改时需更新文档说明

3. 数据库脚本：

- /database/gym_management.sql
- 任何结构变更需更新此文件
- 重大变更需创建新版本脚本

四、持续集成与部署

4.1 CI/CD 流水线

```
sequenceDiagram
    participant Dev as 开发者
    participant Git as Git仓库
    participant CI as CI服务器
```

```
participant Test as 测试环境
participant Prod as 生产环境

Dev->>Git: 推送代码到dev分支
Git->>CI: 触发CI流水线
CI->>CI: 1. 安装依赖
CI->>CI: 2. 运行单元测试
alt 测试通过
    CI->>Test: 3. 部署到测试环境
    Test-->>CI: 自动化测试结果
    alt 测试通过
        CI->>Prod: 4. 手动触发生产部署
    else 测试失败
        CI-->>Dev: 通知修复
    end
else 测试失败
    CI-->>Dev: 通知修复
end
```

4.2 自动化测试用例缺失情况

由于项目仅使用黑盒测试且手动测试功能，因此在持续集成过程使用：

- **手动测试**：开发者需要在每次提交代码前进行手动测试，包括功能测试、回归测试等
- **代码审查**：强制进行代码审查，让其他团队成员检查代码质量，发现潜在的问题和错误，提高代码质量。

五、运维计划

5.1 定期查看日志

日志位置：在pythonanywhere主页中Web页面可以查看日志

- PythonAnywhere: `/var/log/demonnn7.pythonanywhere.com.server.log`
- 报错日志: `/var/log/demonnn7.pythonanywhere.com.error.log`
- Access Log: `/var/log/demonnn7.pythonanywhere.com.access.log`

5.2 维持托管服务

由于使用的是pythonanywhere的免费用户服务，所以网站无法永久长期运行，每三个月将会自动禁用，但可手动重新计时或重新启用。

在pythonanywhere主页中Web页面有运行截止日期（如：当前显示为 “This site will be disabled on **Monday 08 September 2025**”），点击下方按钮 **Run until 3 months from today**，即可继续运行三个月。此外，开启停止前一周邮件提醒，避免网页到期停止，维持网页的长期可用。

5.3 安全措施

1. 访问控制：

- 最小权限原则
- 定期审查账户权限
- 会话超时：30分钟

2. 数据安全：

```
# 参数化查询防止SQL注入
cursor.execute('SELECT * FROM Users WHERE username = %s', (username,))
```

3. 配置安全：

- 生产环境禁用DEBUG模式
- 敏感信息使用环境变量
- 定期更新依赖库