**数据库课程设计实验报告**

**总结报告**



**项目名称： 　　 机票预订系统**

**项目负责人：**

**项目组成员：**

**完成时间：　　　　　 2024.10.9**

项目组成员分工

组长学号及姓名：

分工：项目整体规划、任务分配与协调、进度监控、识别项目中的潜在风险，并制定应对措施。

成绩：A

组员1学号及姓名：

分工：环境搭建：配置数据库和开发环境，如安装数据库管理系统、设置开发工具等、编码实现、应用层开发。

成绩： A

组员2学号及姓名：

分工：概念与逻辑设计、设计文档编写、根据需求分析，进行数据库的ER图设计，定义实体、属性和关系。

成绩： A

组员3学号及姓名：

分工： 用户需求收集、需求分析文档、参与设计讨论、与项目组成员及用户确认需求分析的正确性和完整性。

成绩：A

组员4学号及姓名：

分工：制定测试计划、问题跟踪、技术文档整理、执行测试计划，包括功能测试、性能测试等，记录测试结果。

成绩：A

# 1引言

在现代航空业中，机票管理系统是航空公司和旅行社运营的核心组成部分。随着航空旅行需求的不断增长，传统的手工管理方式已无法满足高效、准确的票务处理需求。因此，开发一个功能全面、用户友好的机票管理系统显得尤为重要。本项目旨在设计和实现一个机票管理系统，以提高票务管理的效率，优化用户体验，并确保数据的安全性和准确性。通过本系统，用户可以方便地进行航班查询、机票预订、订单管理等操作，从而提升整体服务质量。

## 机票管理系统需求概述

本系统用于远程机票预订，包括远程航班信息查询、机票预订与确认等；主要分为四大功能：查询、订票、退票和管理。

管理员登录、注销到系统并进行插入、删除、更新以及查看机票后台数据库操作

插入：机票的插入可以按照航班号、班期、公司、座位号、起飞地以及抵达地等等插入数据库。

删除：机票可以按照航班号、起止城市、星期进行删除

# 2需求说明

（1）功能性需求

用户注册与登录：系统应提供用户注册和登录功能，以确保用户身份的合法性和安全性。

航班查询与展示：系统应能够从航空公司或其他数据源获取航班信息，并将其以清晰、易读的方式展示给用户。

机票预订：用户应能够根据自己的需求选择合适的航班，并进行机票预订操作，包括选择座位、填写乘客信息等。

支付与订单管理：系统应提供安全的支付接口，支持多种支付方式，并能够生成订单并进行订单管理，包括查看订单状态、取消订单等。

乘客信息管理：用户应能够管理自己的乘客信息，包括添加、编辑、删除乘客信息，以便在预订机票时快速选择乘客信息。

航班变更通知：系统应能够及时通知用户航班的变更情况，如航班延误、取消等，以便用户能够及时做出相应的调整。

（2）非功能性需求

用户友好性：系统应具有良好的用户界面设计，简洁明了、易于操作，提供良好的用户体验。

可靠性：系统应具备高可靠性，能够处理大量的用户请求，并保证数据的安全性和完整性。

性能：系统应具备良好的性能，能够快速响应用户的操作请求，保证系统的稳定性和高效性。

安全性：系统应采取必要的安全措施，保护用户的个人信息和支付信息的安全，防止未经授权的访问和数据泄露。

可扩展性：系统应具备良好的可扩展性，能够方便地进行功能扩展和性能优化，以适应未来的需求变化和用户增长。

# 3 系统概念结构设计

## 3.1 局部ER图

旅客订票：



旅客航班：



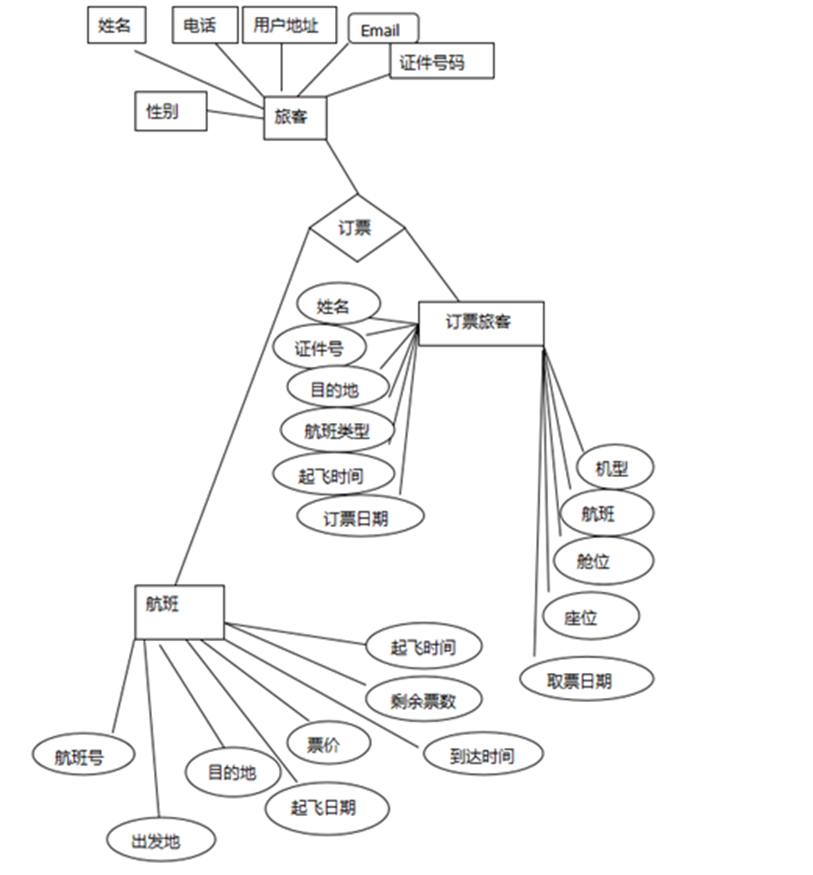
航班与旅客信息：



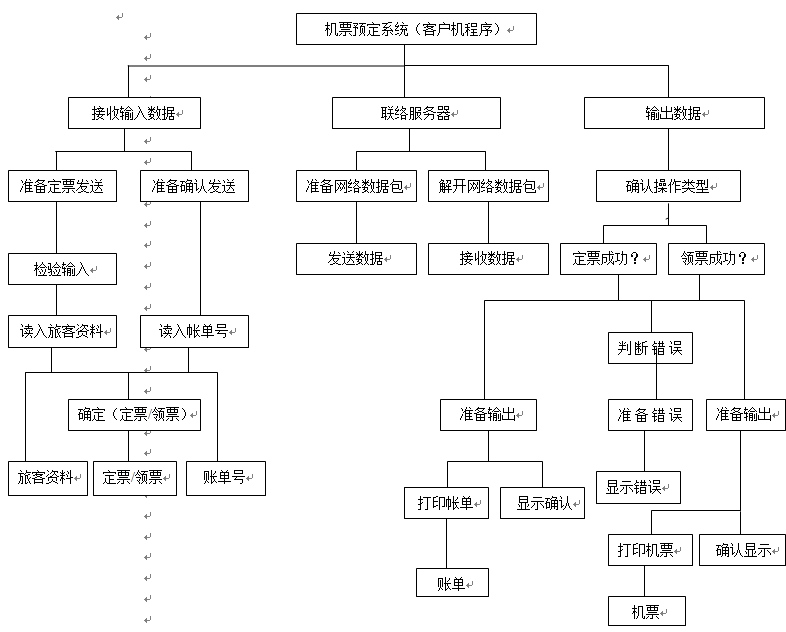
## 3.2 ER图集成



## 3.3 全局ER图



# 4逻辑设计

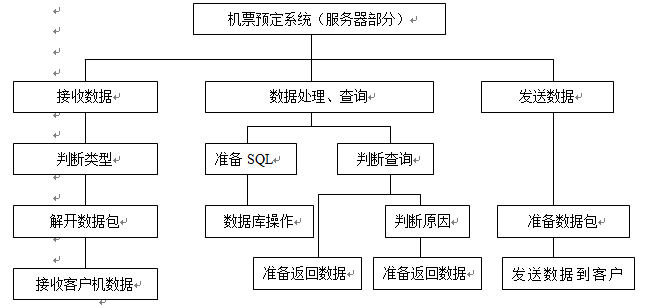


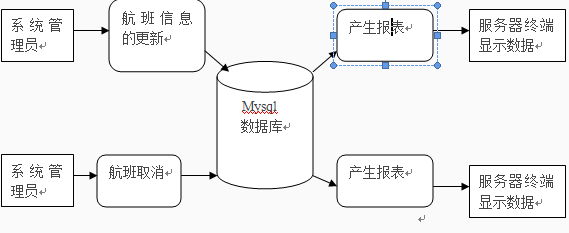
服务器端系统功能

查询：当有旅客查询机票时，接收其数据信息，并能返回查询结果

统计：对售出的机票作出统计

更新航班信息：当客户端发来请求信息，服务器进行接收，





1.用户表（users）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 记录内容 | 是否为空 | 备注 |
| u\_id | Varchar(20) | 用户名 | 0 | 主键 |
| u\_password | Varchar(20) | 密码 | 0 |  |
| u\_name | Varchar(20) | 用户姓名 | 1 |  |
| u\_telephone | Varchar(20) | 手机号码 | 1 |  |
| u\_address | Varchar(100) | 用户地址 | 1 |  |
| u\_email | Varchar(20) | Email | 1 |  |
| u\_idcard | Varchar(20) | 身份证 | 1 |  |
| u\_power | number | 权限 | 1 | 1表示机场人员  0表示旅客 |

2.登录表（login）（统计记录的人数，供服务器查看 共有多少人访问该网站）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 记录内容 | 是否为空 | 备注 |
| l\_uid | Varchar(20) | 用户名 | 0 |  |
| l\_password | Varchar(20) | 密码 | 0 |  |

3.机票订单表（order）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 记录内容 | 是否为空 | 备注 |
| o\_id | Varchar（100） | 订单编号 | 0 | 自增长 |
| o\_uid | Varchar(20) | 用户名 | 0 |  |
| o\_tid | Varchar(40) | 票编号 | 0 |  |
| o\_time | Varchar(40) | 下单时间 | 0 |  |

4.机票信息（tacket\_infor）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 记录内容 | 是否为空 | 备注 |
| t\_id | Varchar（100） | 机票编号 | 0 | 主键 |
| t\_f\_id | number | 航班id 号 | 0 |  |
| t\_uid | Varchar(20) | 用户Id | 0 |  |
| t\_seatId | Number | 座位号 | 0 |  |
| t\_date | Varchar(20) | 该票的日期 | 0 | 这是一个出发的日期 |
| t\_type | Varchar（30） | 座位类型 | 1 | 这是一个座位的类型 |
| t\_price | Number | 票价 | 1 | 这个舱位的价格 |

5.航班信息表（flight\_infor）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 记录内容 | 是否为空 | 备注 |
| f\_id | number | 航班id 号 | 0 | 航班起飞的ID号(主键) |
| f\_number | Vachar2(30) | 航班号 | 0 |  |
| f\_start\_time | Varchar(40) | 出发时间 | 0 |  |
| f\_end\_time | Varchar(40) | 到达时间 | 0 |  |
| f\_departure | Varchar(40) | 始发地 | 0 |  |
| f\_destination | Varchar(40) | 目的地 | 0 |  |
| f\_company | Varchar(30) | 航空公司 | 0 |  |

6.航空仓类型——座位表(seat)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 记录内容 | 是否为空 | 备注 |
| s\_id | Number | 座位唯一Id | 0 | 标识 能够以下得到信息 |
| f\_id | number | 航班id | 0 | 航班id(外键) |
| s\_ type | Varchar（30） | 舱位类型 | 0 | 舱位类型 |
| s\_number | Number | 舱位数量 | 0 | 舱位数量 |
| s\_price | Number | 舱位价格 | 0 | 舱位价格 |

7.用户操作记录表（deal\_log）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 记录内容 | 是否为空 | 备注 |
| d\_number | number | 用户编号 | 0 | 主键，自增长 |
| d\_uid | Varchar(30) | 用户名 | 0 |  |
| d\_opertype | Varchar(40) | 操作类型 | 0 | 订票，退票，改签 |
| d\_tid | Varchar(40) | 机票编号 | 0 |  |
| d\_time | Varchar(10) | 时间 | 0 | 操作的时间 |

8.航班动态表（state）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 记录内容 | 是否为空 | 备注 |
| fs\_number | Varchar(30) | 航班号 | 0 | 航班号 |
| fs\_date | Varchar(10) | 航班飞行日期 | 0 | 航班 |
| fs\_state | Varchar(300) | 航班动态 | 0 | 航班动态 |

# 5.物理设计

## 5.1 系统环境

系统版本：Linux Ubuntu 22.04

数据库版本：MySQL 8.0.39-0ubuntu0.22.04.1

网站服务器：Apache/2.4.52 (Ubuntu)

前端/客户端：HTML5，CSS

后端/服务端：PHP 8.1.2-1ubuntu2.18

## 5.2 硬件环境

1. 网站服务器

服务器类型：LightNode Virtual Private Server

公网Ipv4地址：38.60.212.7

地址：日本东京

CPU：2 intel vCPU

RAM：4 GB

系统硬盘：50 GB

2. 数据库服务器

服务器类型：579Yun Virtual Private Server

公网Ipv4地址：110.40.64.15

地址：中国成都

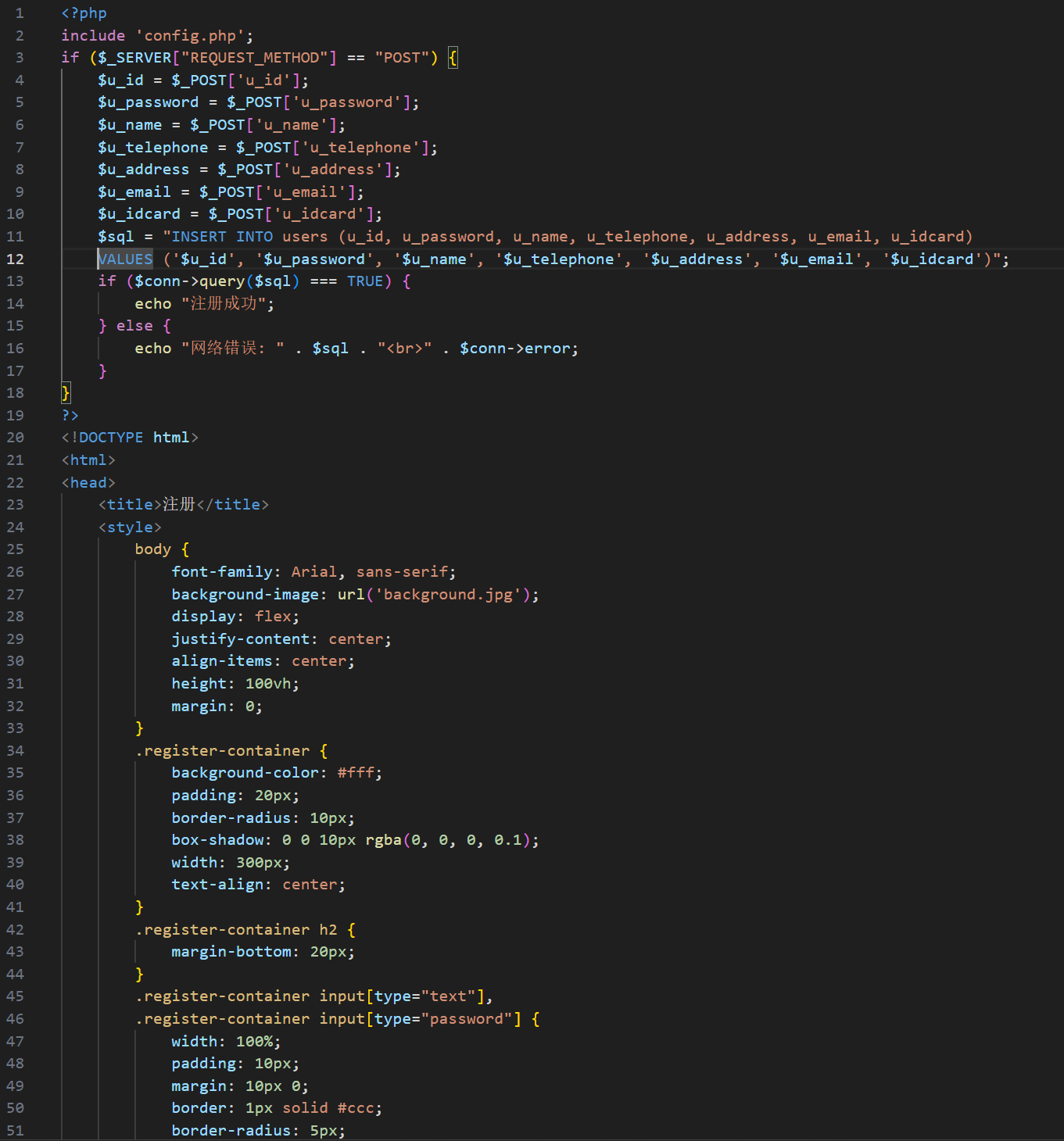
CPU：4 intel vCPU

RAM：4 GB

系统硬盘：40 GB

## 5.3 工程文件及项目资源

注册页面 register.php



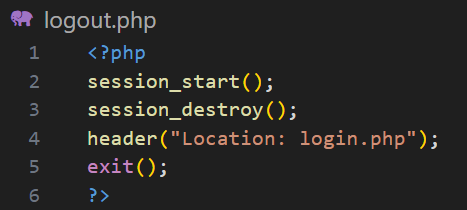


登录页面 login.php

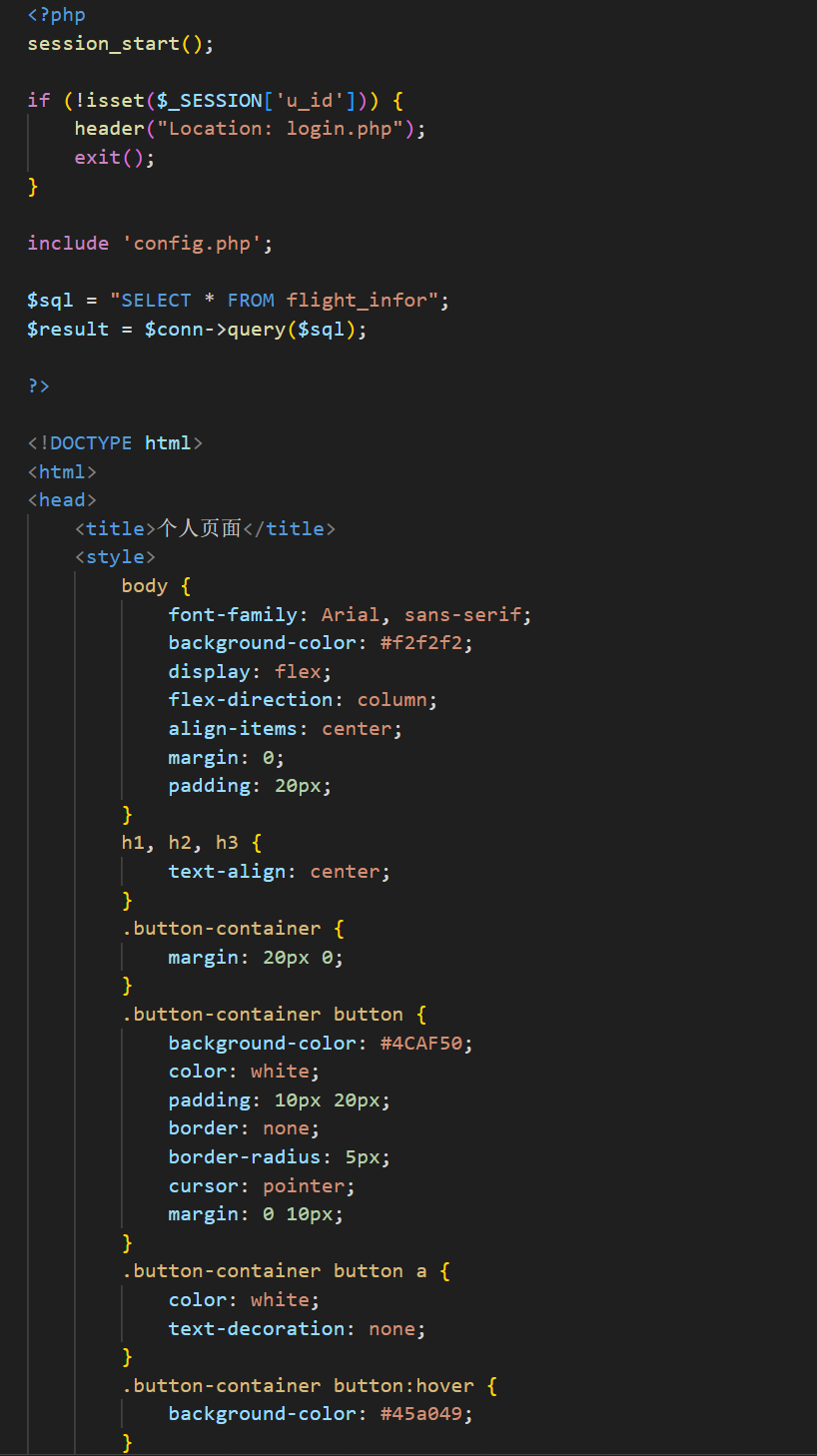




登出页面 logout.php



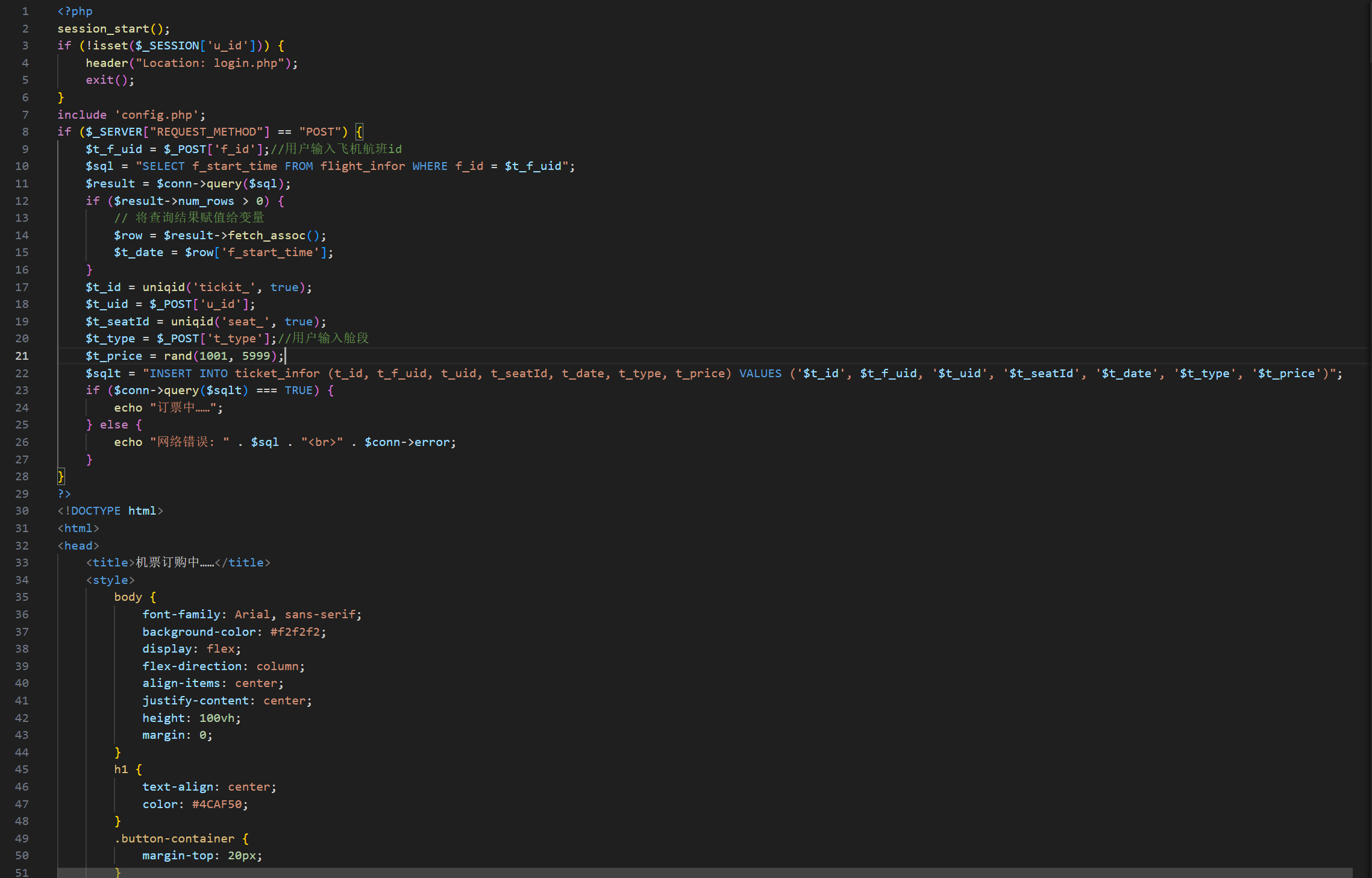
个人主页 welcome.php

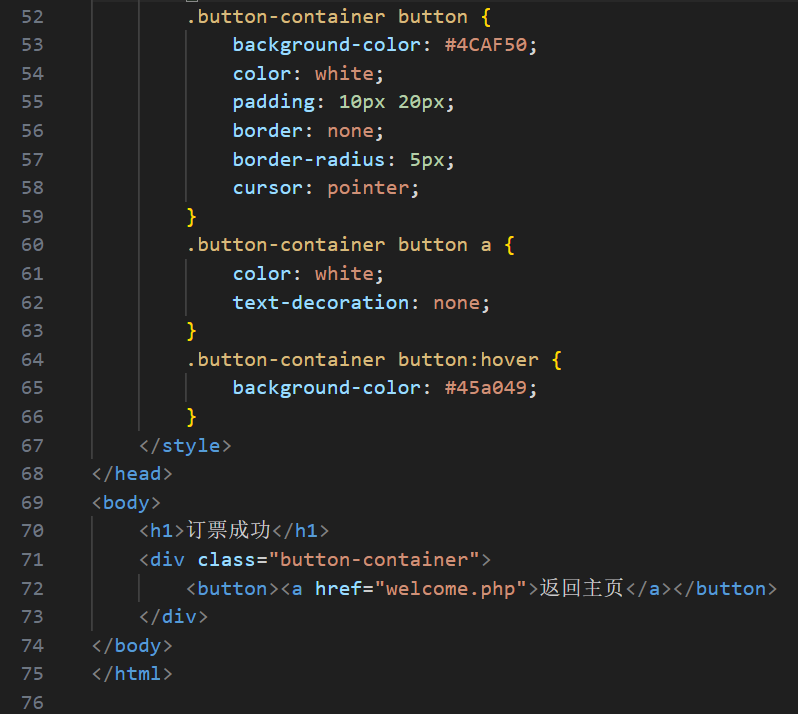




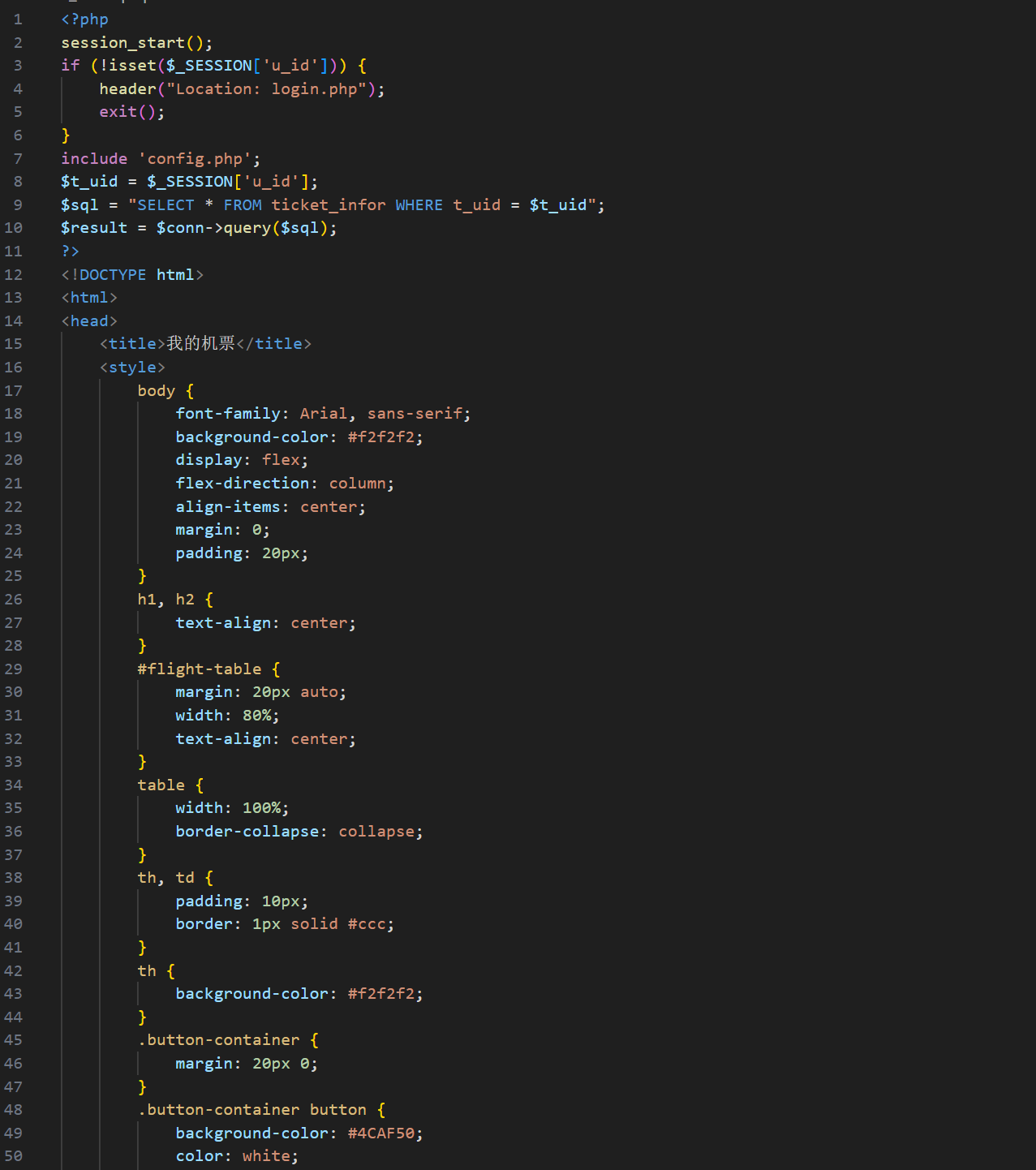


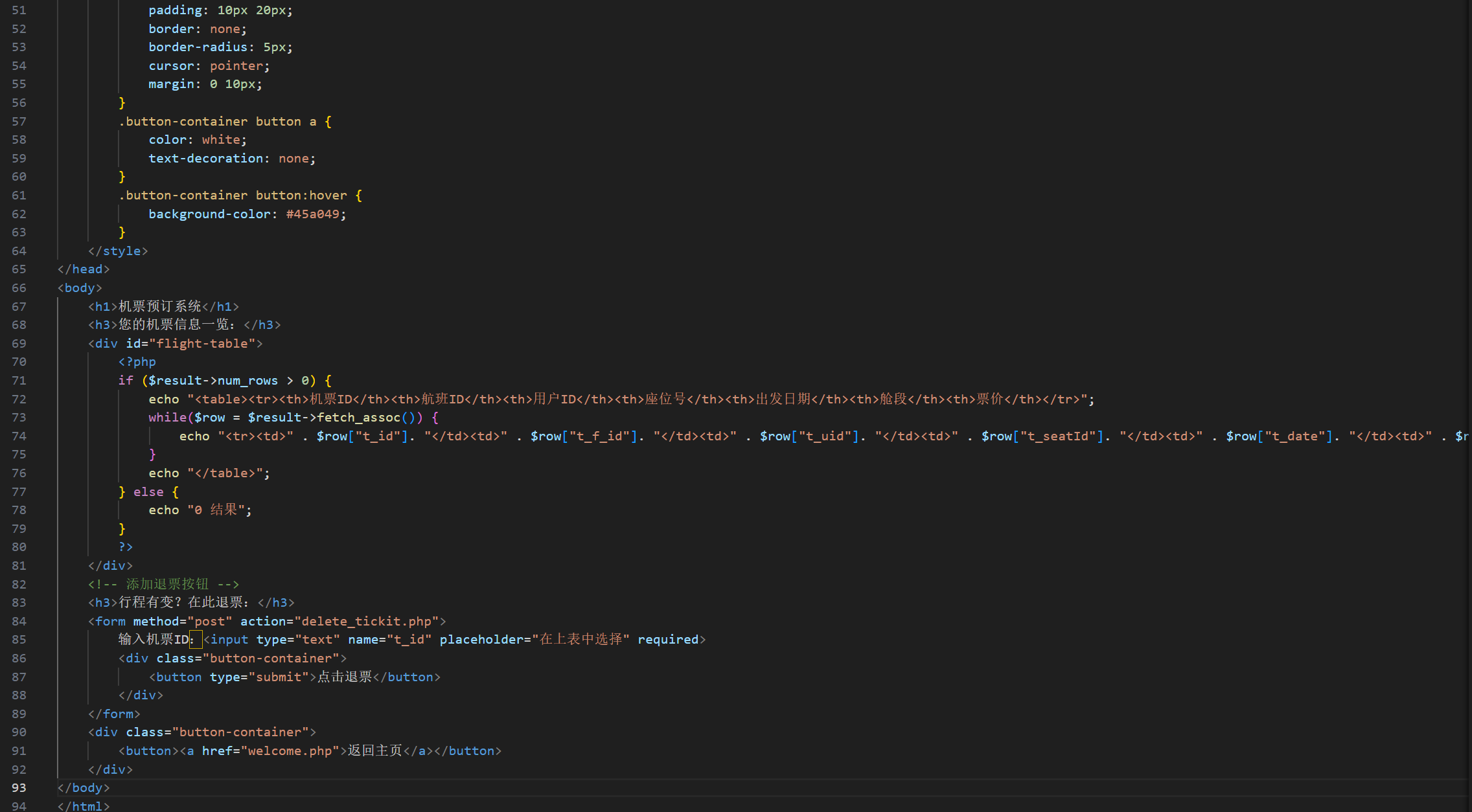
购票窗口 buy\_tickit.php



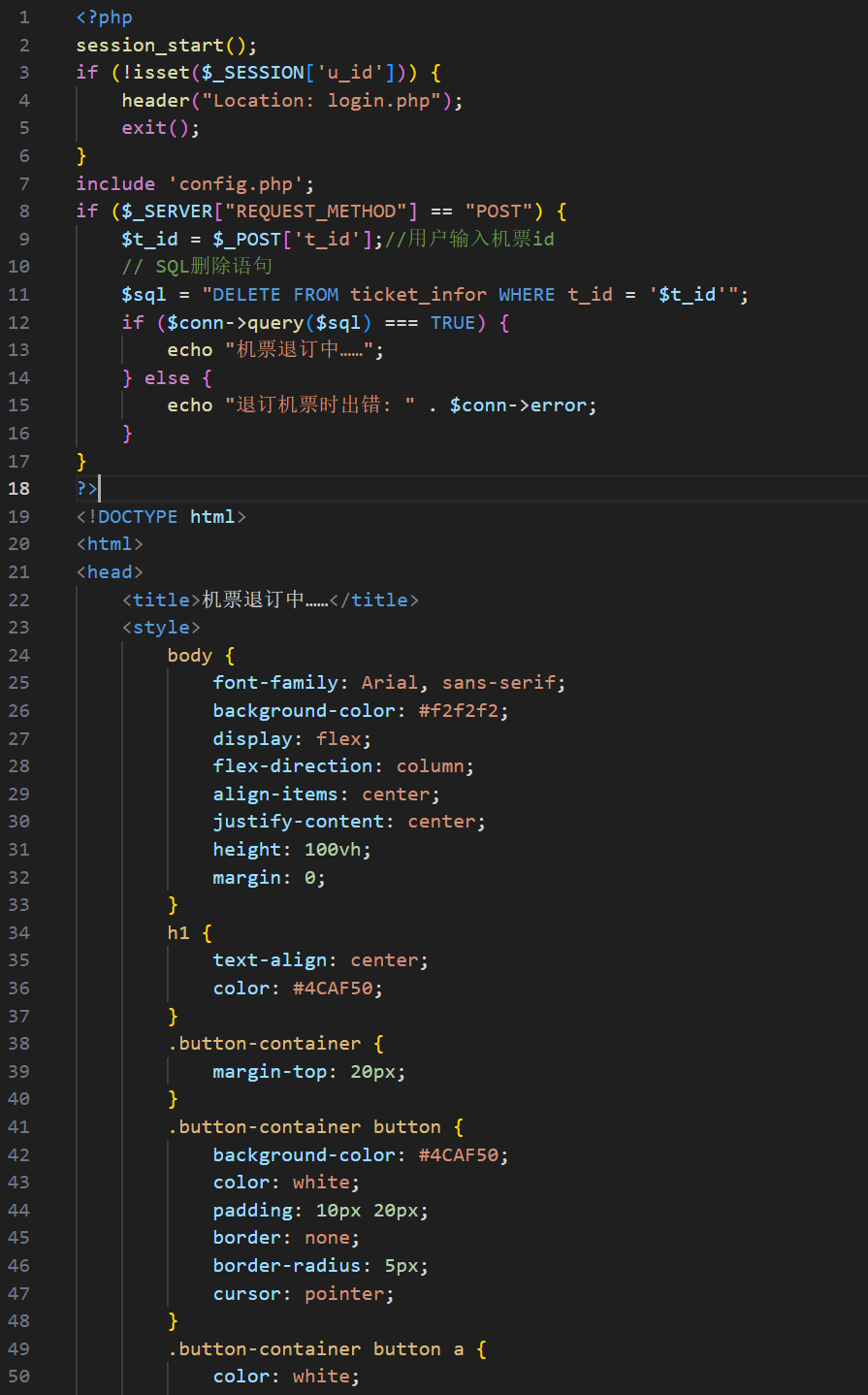


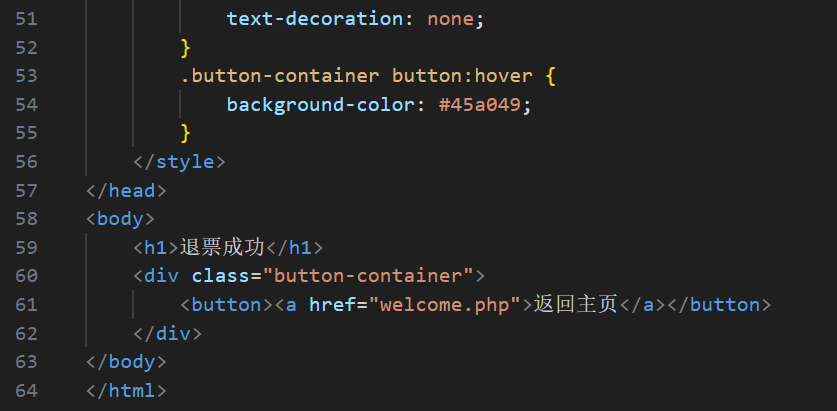
查票窗口show\_tickit.php



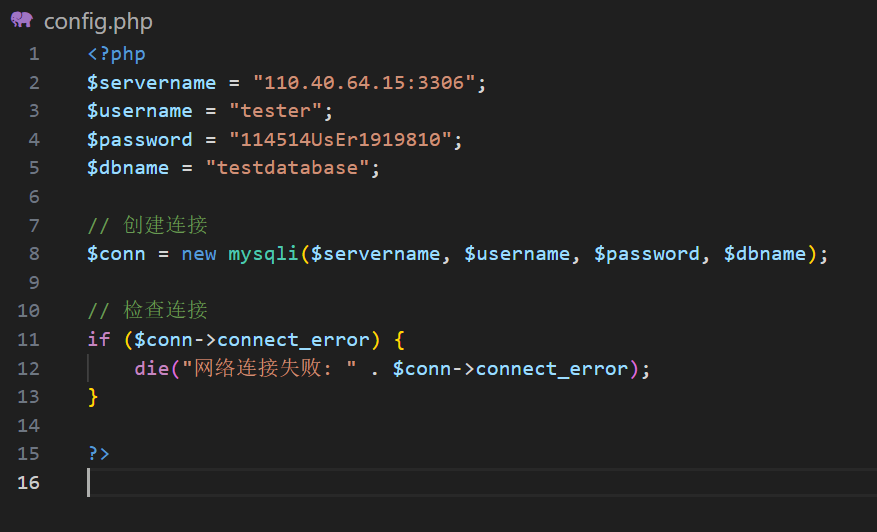


退票窗口delete\_tickit.php





数据库连接配置文件 config.php



背景图片 background.jpg

# 6.数据库实现

1. 创建数据库（如果尚未创建）：

o 使用数据库管理系统（如MySQL）创建一个新的数据库。

2. 创建表：

o 根据逻辑设计，为每个实体创建一个表。

o 在创建表时，指定每个字段的数据类型、是否允许为空、默认值等。

o 为每个表的主键字段添加PRIMARY KEY约束。

o 根据需要为外键字段添加FOREIGN KEY约束，以维护表之间的关系。

3. 调整和优化：

o 根据实际需求，对表的字段进行增加、删除或修改。

o 对表的索引进行优化，以提高查询性能。

4. 安全性考虑：

o 对敏感信息（如密码）进行加密存储。

o 设置适当的访问权限，确保只有授权用户才能访问和修改数据。

5. 建库指令：

deal\_log 表

CREATE TABLE deal\_log (

d\_number INT PRIMARY KEY AUTO\_INCREMENT,

d\_uid VARCHAR(30) NOT NULL,

d\_opertype VARCHAR(40) NOT NULL,

d\_tid VARCHAR(40), -- 这里可能是指向ticket\_infor或orders表的外键，具体取决于业务逻辑

d\_time VARCHAR(10) DEFAULT CURRENT\_TIMESTAMP

);

-- flight\_infor 表

CREATE TABLE flight\_infor (

f\_id INT PRIMARY KEY AUTO\_INCREMENT,

f\_number VARCHAR(30) NOT NULL,

f\_start\_time VARCHAR(40) NOT NULL,

f\_end\_time VARCHAR(40) NOT NULL,

f\_departure VARCHAR(40) NOT NULL,

f\_destination VARCHAR(40) NOT NULL,

f\_company VARCHAR(30) NOT NULL

);

-- login 表

CREATE TABLE login (

l\_uid VARCHAR(20) PRIMARY KEY,

l\_password VARCHAR(20) NOT NULL -- 建议使用更安全的密码存储方式，如哈希

);

-- orders 表

CREATE TABLE orders (

o\_id INT PRIMARY KEY AUTO\_INCREMENT,

o\_uid VARCHAR(20) NOT NULL,

o\_tid VARCHAR(40), -- 指向ticket\_infor表的外键

o\_time VARCHAR(40) DEFAULT CURRENT\_TIMESTAMP

);

-- seat 表

CREATE TABLE seat (

s\_id INT PRIMARY KEY AUTO\_INCREMENT,

f\_id INT NOT NULL, -- 指向flight\_infor表的外键

s\_type VARCHAR(30) NOT NULL,

s\_number INT NOT NULL,

s\_price INT NOT NULL,

FOREIGN KEY (f\_id) REFERENCES flight\_infor(f\_id)

);

-- state 表

CREATE TABLE state (

fs\_number VARCHAR(30) PRIMARY KEY,

fs\_data VARCHAR(10),

fs\_state VARCHAR(300) NOT NULL

);

-- ticket\_infor 表

CREATE TABLE ticket\_infor (

t\_id VARCHAR(100) PRIMARY KEY AUTO\_INCREMENT,

t\_f\_uid INT, -- 这里假设是指向flight\_infor表的外键，但也可能需要调整

t\_uid VARCHAR(20) NOT NULL, -- 购买机票的用户ID，可能指向users表

t\_seatld INT, -- 指向seat表的外键

t\_date VARCHAR(20) NOT NULL,

t\_type VARCHAR(30) NOT NULL,

t\_price INT NOT NULL,

FOREIGN KEY (t\_seatld) REFERENCES seat(s\_id),

-- 如果t\_f\_uid确实是指向flight\_infor的外键，则添加以下行

-- FOREIGN KEY (t\_f\_uid) REFERENCES flight\_infor(f\_id)

-- 如果t\_uid是指向users的外键，则添加以下行（但通常用户ID在ticket\_infor中不会直接指向users，而是通过orders等中间表关联）

-- FOREIGN KEY (t\_uid) REFERENCES users(u\_uid) -- 注意：这里可能需要根据实际业务逻辑调整

-- 由于直接添加t\_uid到users的外键可能导致多对一问题（一个用户可能有多个机票），因此通常不这样做，而是通过orders表来关联

);

-- 注意：由于orders和ticket\_infor之间通常存在多对一的关系（一个订单可能包含多个机票），

-- 因此通常不会在ticket\_infor中直接添加指向users的外键。相反，您可能会通过orders表来查找哪个用户购买了哪些机票。

-- 如果确实需要在ticket\_infor中直接记录用户信息，请考虑使用其他方法，如触发器或应用层逻辑。

-- users 表

CREATE TABLE users (

u\_uid VARCHAR(20) PRIMARY KEY,

u\_password VARCHAR(20) NOT NULL -- 同样建议使用更安全的密码存储方式

u\_name VARCHAR(20) NOT NULL,

u\_telephone VARCHAR(20),

u\_address VARCHAR(100),

u\_email VARCHAR(20),

u\_idcard VARCHAR(20),

u\_power INT NOT NULL -- 用户权限等

);

-- visit\_log 表

-- 注意：visit\_log表中通常还需要一个时间戳字段来记录访问时间，但您只提供了user\_ip

-- 如果您确实只需要记录IP地址，那么以下语句就足够了

CREATE TABLE visit\_log (

user\_ip VARCHAR(30) PRIMARY KEY,

visit\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP -- 添加时间戳字段以记录访问时间（可选）

);

# 7.实验项目成果验收

在本次数据库实践实验中，我们成功开发了一个功能完善的机票预订系统。该系统实现了用户注册、航班查询、机票预订、订单管理等核心功能，满足了预期的需求和目标。

通过团队的共同努力，我们在系统设计、数据库构建、前后端开发以及测试阶段都取得了显著的成果。系统不仅具备良好的用户体验，还在性能和安全性方面表现出色。所有功能模块经过严格测试，运行稳定，数据处理准确无误。

此次实验不仅提升了我们的技术能力，还增强了团队协作和项目管理的经验。我们对项目的最终成果感到非常满意，并相信该系统在实际应用中将发挥重要作用。¹²