YOYOTLX



網際網路系統設計期末專題

資管二 4112029008 蔡侑容



簡述

- 本專案模擬 KKTIX 購票平台,使用 Django MVC 架構實作。
- 提供會員註冊與登入、活動票券購買、訂單紀錄、客服留言系統等功能。



核心功能

系統

活動展示 主頁呈現所有活動,支援搜尋與分類

每場活動有獨立詳情頁,顯示圖片、時間、票價與剩餘票數

購票系統 支援開賣時間控制(尚未開賣會提示,且無法選擇)

每人購票上限控制(最多購買4張)

剩餘票數即時更新

完成前有資料確認頁面

會員系統 註冊/登入/編輯個人檔案

會員可查詢歷史訂單與票券狀態

客服留言使用者可選擇匿名或具名留言

支援「公開/非公開」分類,維持留言隱私



購票流程

使用者登

• 未登入者將被導向登入頁面

選擇活動

- 首頁 (index.html) 點選活動卡片
- 進入活動詳情頁 (/event/<evid>/detail)
- 點選「下一步」按鈕進入購票流程

• 購票頁面 (/event/<evid>/registration)

• 使用者可選擇:

選擇票種 與張數

- 票種(每筆訂單僅限一種票種)
- 購買張數 (1~4 張)
- 必須勾選「我已閱讀並同意服務條款與隱私權政策」



購票流程



- 進入確認頁面 (/event/<evid>/registration/check)
- 顯示購票資訊與張數
- 勾選「確認表單資料」才能繼續

導向結帳頁面 (/event/<evid>/checkout)選擇付款方式:

結帳付款

- 信用卡
- ATM轉帳
- 可由使用者訂單導回此頁

取票

• 於使用者訂單點選取票按鈕



專案架構

- 核心 Django App:
 - accounts/:使用者註冊、登入、個人資料管理
 - tickets/:活動、票券、訂單的建立與管理
 - support/: 客服中心,支援使用者留言(可匿名、公開/非公開)
- 靜態與媒體資源:
 - static/: 前端樣式 (CSS) 、圖片與 JavaScript
 - media/:活動封面圖與座位圖上傳儲存位置
- templates/ 模板頁面:
 - 包含註冊登入、活動詳情、購票流程、客服中心等頁面
- yoyotix/ 核心 Django 專案設定 :
 - 包含 settings.py、urls.py 等配置檔



專案架構

• 專案樹狀結構摘要圖

```
yoyotix/
   accounts/ # App 1:會員系統
    ── forms.py # 註冊、修改個人資料
      — models.py # 自訂    CustomUser
   tickets/ # App 2:活動票券系統
    --- models.py # Event \ Ticket \ Order
      forms.py # Form
      - signals.py
   support/ # App 3:客服中心
       models.py # Message 模型
      - forms.py # ModelForm
```



專案架構

• 專案樹狀結構摘要圖

```
media/ # 媒體資源 演出圖片資料夾(from admin)
L— events/
    ├--- # 主視覺
      seating_maps/ # 座位圖
static/ # 靜態資源
-- css/
 — image/
 — js/
templates/
 — accounts/ #登入 註冊 個人資料
  - support/ # 客服留言 常見問題
  — tickets/ # 票務相關頁面
  - includes/
    - footer.html
   L-- header.html
  — base.html # 所有頁面共用母模板
L— index.html #首頁
supportdb.sql # 資料庫
db.sqlite3
manage.py
yoyotix/
  - route.py # 資料庫路由
  — urls.py #網頁路徑
```



技術說明

使用者系統(accounts app)

會員相關邏輯

活動與購票系統(tickets app)

票務相關邏輯

客服留言系統 (support app)

留言板相關邏輯



User App 設計

- 技術說明
 - Custom User Model (擴充預設 User)
 - User 註冊 / 登入 / 登出 (表單驗證與訊息提示)
 - @login_required 權限控管
 - PasswordChangeForm 內建密碼變更表單
 - 表單應用
 - forms.Form表單 (登入)
 - Forms.Modelform (修改資料)
 - UserCreationForm (使用者註冊)
 - 雙次密碼驗證、Captcha驗證
- Model設計如右圖
 - 繼承內建 AbstractUser

```
# accounts/models.py
from django.contrib.auth.models import AbstractUser
from django.db import models

class CustomUser(AbstractUser):
    nickname = models.CharField(max_length=30, blank=True)
    phone = models.CharField(max_length=15, blank=True, unique=True)
    birthday = models.DateField(null=True, blank=True)

email = models.EmailField(unique=True)

def __str__(self):
    return self.username
```



- 技術說明
 - Session 儲存購票中間資料
 - @login_required 權限控管
 - 表單應用 使用Form
 - RadioSelect 與 ChoiceField 處理票種選擇
 - 表單驗證:判斷開賣時間、票數限制、餘票
 - 頁面動態設計
 - 使用qrcode[pil]
- Model四層關聯模型設計(於次頁)
 - Event:基本演唱會資訊與圖片
 - Ticket:同一個 Event 可有多種票種(名稱、票價、數量)
 - 未儲存剩餘票券(可用數量-售出座位計算) 符合第三正規化
 - Order: 連結 User 和 Ticket
 - Quantity 表示買多少張
 - Status 管理訂單狀態
 - Created_at + is_expired() 判斷訂單是否超時(保留15分鐘)
 - 超時自動取消訂單
 - TicketCode



```
| class Event(models.Model):
    title = models.CharField(max_length=200, verbose_name="名稱")
    description = models.TextField(verbose_name="描述")
    main_image = models.ImageField(
    upload_to='events/',
    verbose_name="主視覺圖",
    default='event/default.png'

| event_date = models.DateTimeField(verbose_name="演出日期")
    location = models.CharField(max_length=200, verbose_name="地點")
    organizer = models.CharField(max_length=200, verbose_name="主辦單位")
    seating_map = models.ImageField(upload_to='events/seating_maps/', null=True, blank=True, verbose_name="座位圖")

| def __str__(self):
    return self.title
```

- 圖片存於media資料夾
- 紀錄演出時間

```
class Ticket(models.Model):
   event = models.ForeignKey(Event, on delete=models.CASCADE, related name='tickets')
   name = models.CharField(max length=100, verbose name="票種名稱", default="全票") # 如「VIP票」
   price = models.DecimalField(max digits=8, decimal places=2, verbose name="票價")
   quantity = models.PositiveIntegerField(verbose name="座位數", default=0) # 座位數量
   sold = models.PositiveIntegerField(default=0, verbose name="已售張數")
   start time = models.DateTimeField(verbose name="開賣時間", default=timezone.now) # 新增欄位
   def str (self):
       return f"{self.event.title} - {self.name} (${self.price})"
   # 例如在 Ticket model 中的 remaining 屬性計算前加入:
   def get valid orders(self):
       for order in self.orders.filter(status='pending'):
           order.auto cancel if expired()
       return self.orders.exclude(status='cancelled')
   def remaining(self):
       if self.quantity is None or self.sold is None:
           return 0
       return self.quantity - self.sold
```

- Event(FK) 一場演出會有多種票種
- 記錄開賣時間
- 自動計算餘票



self.ticket.save()

```
class Order(models.Model):
   STATUS CHOICES = [
       ('pending', '未結帳'),
       ('paid', '已結帳'),
       ('collected', '已取票'),
       ('cancelled', '已取消'),
                                                                                                                為Unknown
   user = models.ForeignKey(settings.AUTH USER MODEL, on delete=models.SET DEFAULT, default=4)
   ticket = models.ForeignKey(Ticket, on delete=models.CASCADE, related name='orders')
   quantity = models.PositiveIntegerField(verbose name="張數") # 一次只能買一個價位的張數
   status = models.CharField(max length=10, choices=STATUS CHOICES, default='pending', verbose name="狀態")
   created at = models.DateTimeField(auto now add=True, verbose name="建立時間")
   updated at = models.DateTimeField(auto now=True, verbose name="更新時間")
                                                                                                                   消訂單
   def str (self):
       return f"訂單 #{self.id} - {self.user.username} - {self.ticket.name} x {self.quantity}"
   def is expired(self):
          if self.status == 'pending':
               expire time = self.created at + timedelta(minutes=15)
               return timezone.now() > expire time
           return False
                                                                       def cancel order(self):
   def auto cancel if expired(self):
                                                                           if self.status != 'cancelled':
       if self.is expired():
                                                                               self.status = 'cancelled'
           self.status = 'cancelled'
                                                                              self.ticket.sold = max(self.ticket.sold - self.quantity, 0)
           self.save()
                                                                               self.ticket.save()
                                                                              self.save()
           # 將訂單數量從票券的 sold 中扣回來
           self.ticket.sold = max(self.ticket.sold - self.quantity, 0)
```

User(FK)

使用者被删除將改

- Ticket(FK)
- 超時自動取



```
# 驗票碼

class TicketCode(models.Model):
    order = models.ForeignKey(Order, on_delete=models.CASCADE, related_name='ticket_codes')
    code = models.CharField(max_length=64, unique=True)
    created_at = models.DateTimeField(auto_now_add=True)

def __str__(self):
    return f"票碼:{self.code}"

@staticmethod
    def generate_code():
        raw_uuid = uuid.uuid4().hex
        return hashlib.sha256(raw_uuid.encode()).hexdigest()[:20] # 取前 20 碼即可
```

- Order(FK)
- 訂單被刪除將改為 Unknown
- 需pip3 install qrcode[pil]



Support App 設計

- 技術說明
 - 使用 MySQL 作為資料庫後端
 - User 註冊 / 登入 / 登出 (表單驗證與訊息提示)
 - @login_required 權限控管
 - 支援 匿名 / 具名留言,選擇是否公開
 - 設計 留言列表頁面,公開提問紀錄
 - 表單應用
 - forms. ModelForm 處理留言表單驗證與渲染
- Model設計如右圖



技術說明

- 專案架構與基礎設定 settings.py
 - 配置靜態檔(static)、媒體路徑(media)、APP 註冊、自訂資料庫(mysql、router)
- 前端設計與其他
 - Message Framework 顯示提示訊息 (success, error)
 - redirect() / HttpResponse() 導頁與警示
 - Html 迴圈、判別、 block area、form應用
 - staticfiles:整合CSS/JS/image
 - 使用 URL 反解 可避免硬編網址
 - reverse() 動態產生 URL (購票導頁)
 - {% csrf_token %} 保護表單
 - get_object_or_404() 防止無效或不存在的資料導致錯誤
 - timezone.now() 判斷票券開賣與活動時間
 - RWD 響應式網頁設計



關鍵頁面展示 YOYOTIX





使用者相關



XITOYOY

Index 一般使用者登入

顯示使用者暱稱/帳號



演出活動









Index 未登入



Index 一般使用者



XITOYOY

登入 /user/login



XITOYOY

註冊 /user/register

註冊新帳號 使用者名稱 請輸入使用者名稱 暱稱 請輸入暱稱 電子信箱 Phone Birthday 年/月/日 密碼 機器人驗證 密碼確認 Captcha XXXX 成功註冊將自動導向首頁 IG 連結 導向登入頁面 已經有帳號?登入 聯絡我們:94ruthtsai@gmail.com 0 YOYOTIX 客服時間:我醒著的時候 22 © 2025 YOYOTIX All Rights Reserved.

個人資料 /user/profile



		YOYOTIX 探索活動 購票流程	常見問題 我的票券	蔡侑容▼
			個人資料	
			登入成功,歡迎回來!	
			修改暱稱、生日與電話	
			壓稱: 蔡侑容	
			生日: 年/月/日	
			手機號碼: 0911111111	
			更新個人資料	
All share			變更密碼	
舊密碼	在認永久刪除帳號? ×		舊密碼:	
新密碼。此	上操作無法復原,確定要永久刪除您的帳號嗎?		新密碼:	
МУ БЫ АН.			你的密碼不能與其他個人資訊太相近。 你的密碼不能與其他個人資訊太相近。	
• 彼	取消 確認刪除		你的密碼必須包含至少8個字元。你不能使用常見的密碼。你的密碼不能完全是數字。	
	台端必須包含至少 8 個子元。 能使用常見的密碼。		新密碼確認:	
	容碼不能完全是數字。		更新密碼	
	 點擊後會彈出視窗,	次確認是否刪除	刪除帳號	
		YOYO	聯絡我們:94ruthtsai@gmail.com 客服時間:我醒著的時候	0



票務相關 YOYOTIX



活動頁面/event/<evid>/detail

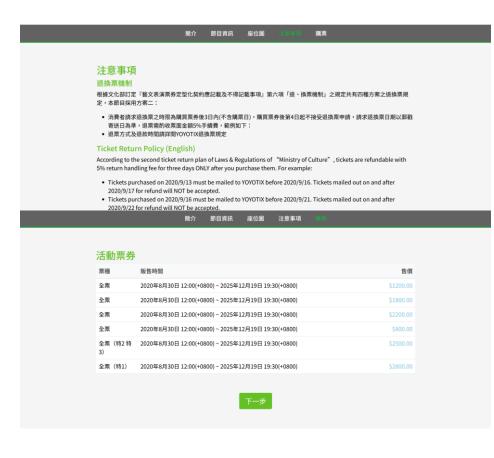






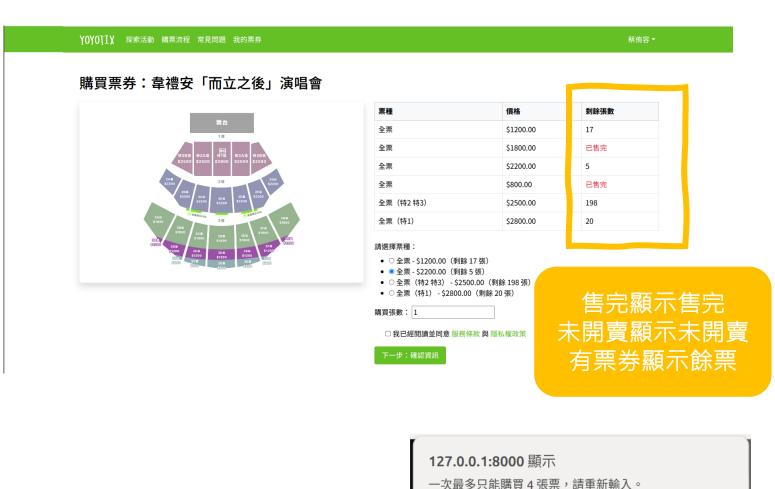
上圖為KKTIX 票價顏色是我的得意之作





購票頁面/event/<evid>/regrastration







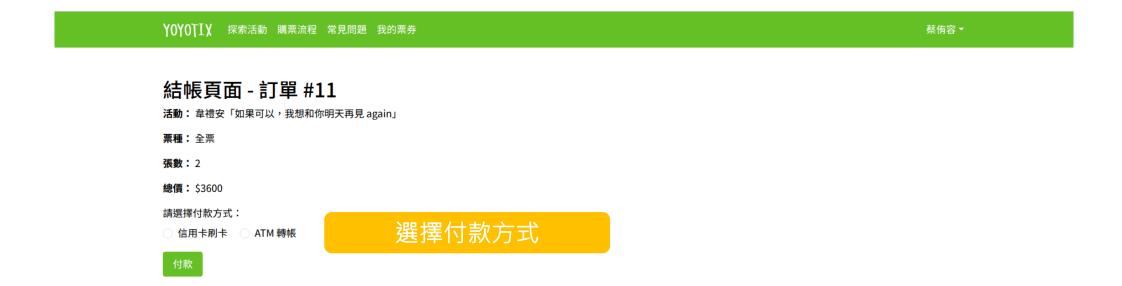
確認頁面/event/<evid>/regrastration/check





結帳頁面/user/order/<order_id>/checkout







XITOYOY

訂單頁面/user/order



我的票券頁面/user/tickets

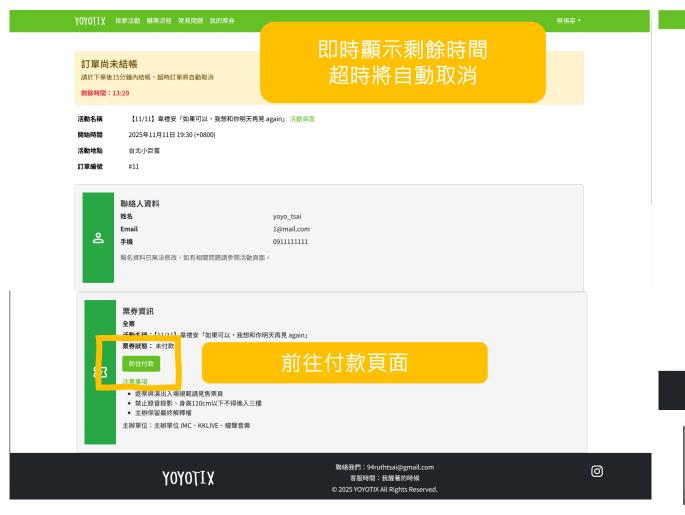




訂單頁面/user/order/<order_id>



未付款



已付款未取票



訂單頁面/user/order/<order_id>



已取票



已取消



購票流程頁面/events/strategy



(0Y0) 【X 探索活動 購票流程 常見問題 我的票券

蔡侑容▼

哈哈分享我喜歡的歌啦~

沒錯,你被騙進來了! 本來以為是什麼購票攻略對吧? 結果我只是要跟你分享這首歌而已,嘻嘻



嵌入Youtube

喜歡的話就加到歌單! 還是你也有歌要騙我點進去聽?我是不會被騙的

XITOYOY

聯絡我們:94ruthtsai@gmail.com 客服時間:我醒著的時候 © 2025 YOYOTIX All Rights Reserved.





客服相關 YOYOTIX



聯絡客服 /contact



Y0Y0TIX 探索活動 購票流程 常見問題 我的票券		蔡侑容▼
	聯絡客服	
暱稱(可留空):		
留言內容:		
類別: 是否公開:	購票 🗸	
	送出留言	

XITOYOY	聯絡我們:94ruthtsai@gmail.com 客服時間:我醒著的時候 © 2025 YOYOTIX All Rights Reserved.	0
---------	---	---

常見問題 /messages



