

รายงาน

เรื่อง Ticket Concert

โดย

กลุ่มห้อง sec102

นางสาวนนทิชา	สุขเจริญ	63010484
นางสาวนภสร	ชาลานุมาศ	63010492
นายปุณณวิษฐ์	พานิชผล	63010616
นางสาวพัฒนชิตา	ธีรพัฒน์โรจน์	63010665
นายพีระภัทร์	เศรษฐพรนรา	63010702

เสนอ

อาจารย์คณัฐ ตั้งติสานนท์

อาจารย์เกียรติณรงค์ ทองประเสริฐ

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา Data Structures and algorithm

รหัสวิชา 01076005

ภาคเรียนที่ 1 ปีการศึกษา 2564

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำนำ

รายงาน เรื่อง Ticket Concert มีวัตถุประสงค์จัดทำขึ้นเพื่อจำลองการซื้อขายบัตรเข้าชมคอนเสิร์ต โดยสามารถทำงานผ่านทางแพลตฟอร์มเว็บแอปพลิเคชันที่ได้จัดทำขึ้นมา

คณะผู้จัดทำได้รับมอบหมายงานให้จัดทำชิ้นงาน 1 ชิ้นงาน โดยงานมีความเกี่ยวข้องกับเนื้อหาวิชา Data Structures and algorithm จึงได้นำโครงสร้างของข้อมูลและอัลกอริทึมด้วยภาษาไพธอนมาประยุกต์ใช้กับชิ้นงานนี้

ตลอดระยะเวลาการจัดทำโครงงาน คณะผู้จัดทำได้นำความรู้ที่ได้เรียนและศึกษามาใช้ในการจัดทำชิ้นงาน ทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงานนี้จะมีประโยชน์ต่อผู้อ่านไม่มากนักน้อย

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1-2
ที่มาและความสำคัญ	
ข้อกำหนดของโปรแกรม	
ขอบเขตการใช้งาน	
บทที่ 2 ภาพรวมและการออกแบบโปรแกรม	3-7
การออกแบบโปรแกรม	
รายละเอียดข้อมูลที่เกี่ยวข้อง	
บทที่ 3 การทำงานของโปรแกรม	8-12
การทำงานของโปรแกรม	
โครงสร้างส่วนของข้อมูล	
โค้ดของการทำงานส่วนโปรแกรมหลัก	
บทที่ 4 ผลการทำงานของโปรแกรม	13-15
บทที่ 5 สรุปผลการทำงานของโปรแกรม	16
การมีส่วนร่วมของสมาชิกในกลุ่ม	17
เอกสารอ้างอิง	18

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากทางคณะผู้จัดทำได้เล็งเห็นถึงปัญหาในการซื้อบัตรเข้าชมงานคอนเสิร์ต เมื่อถึงช่วงเวลาขายบัตรจะเกิดความวุ่นวายในการจัดเก็บข้อมูลหรือบางครั้งเกิดเหตุการณ์ขายบัตรไม่หมด เพราะ ช่องทางการขายไม่สามารถเอื้อมไปถึงคนซื้อได้อย่างทั่วถึง ทางคณะผู้จัดทำจึงจัดทำเว็บไซต์จองบัตรคอนเสิร์ตขึ้นมา เพื่อให้ปัญหาเหล่านั้นหมดไป

1.2 ข้อกำหนดของโปรแกรม

โปรแกรมจะจำลองการซื้อบัตรเข้าชมงานคอนเสิร์ต โดยสามารถทำงานผ่านทางแพลตฟอร์มเว็บแอปพลิเคชัน ซึ่งใช้งานได้ผ่านโปรแกรมที่ผู้จัดทำได้จัดทำทำขึ้น

1.3 ขอบเขตการใช้งาน

โดยโปรแกรมจะประกอบไปด้วย

1. หน้าเว็บ

แสดงแถบเมนู เช่น แถบเมนูหลัก แถบค้นหาและช่องทางติดต่อ

2. หน้าทำการจอง

หน้านี้จะแสดงแผนผังคอนเสิร์ตสำหรับทำการจอง สามารถจองคอนเสิร์ตได้ 1 คอนเสิร์ต มีวันและช่วงเวลาเดียว แต่ผู้ใช้งาน 1 คนสามารถจองได้มากกว่า 1 โชน

3. หน้าโชน

แบ่งเป็น 2 โชน คือ โชนนั่งและโชนยืน เมื่อเข้าแต่ละโชน จะแสดงจำนวนโชนและจำนวนที่นั่งสำหรับการจอง

4. หน้าชำระเงิน

เมื่อทำการจองเรียบร้อยแล้ว จึงเข้าใช้งานหน้าชำระเงินได้

5. หน้าค้นหา

เมื่อชำระเงินเรียบร้อยแล้ว จึงสามารถค้นหาข้อมูลของผู้ที่ทำการซื้อบัตรเข้างานคอนเสิร์ตได้ ซึ่งจะแสดงผลผ่านทางหน้าเว็บ

6. ช่องทางติดต่อ

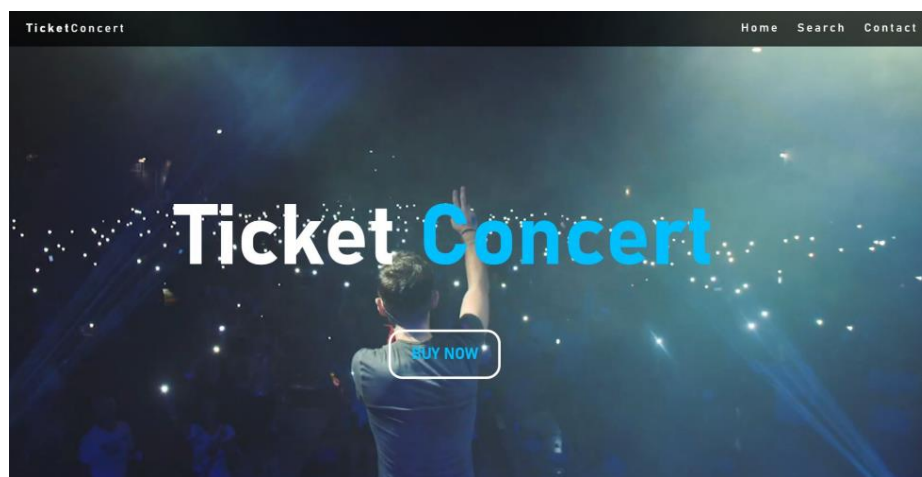
กรอกชื่อ นามสกุล อีเมลล์และข้อความที่ต้องการจะติดต่อ ซึ่งมีระบบแจ้งเตือนผ่านทางอีเมลล์

บทที่ 2

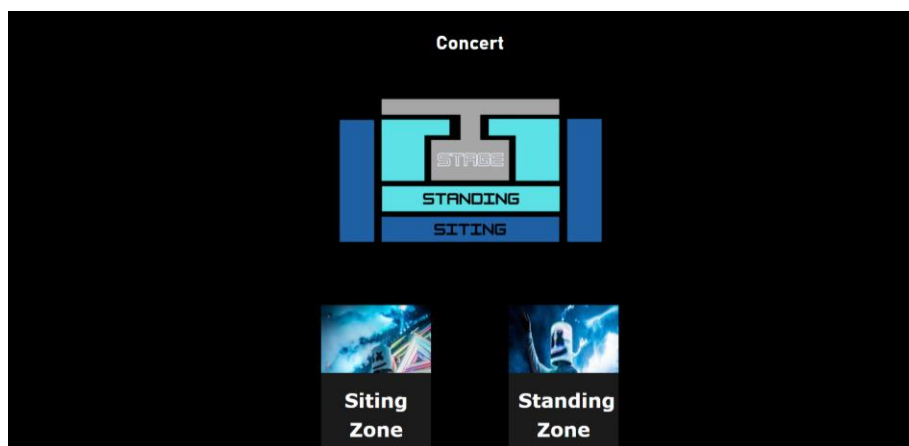
ภาพรวมและการออกแบบโปรแกรม

2.1 การออกแบบโปรแกรม

1. หน้าหลัก



2. หน้าทำการจอง



ในการทำการจองบัตรคอนเสิร์ต จะต้องทำการเลือกโซน ซึ่งมีทั้งหมด 2 โซน คือ โซนนั่งและโซนยืน โดยมีภาพแสดงแผนผังคอนเสิร์ตอยู่ข้างบน

3. หน้าของแต่ละโซน

- หน้าโซนนั่ง

Siting Zone

Buy Ticket

Zone
Zone A

Seat
1

Full name

Payment

ในส่วนของโซนนั่งจะมีทั้งหมด 3 โซน จะต้องทำการจองบัตรคอนเสิร์ต โดยการเลือกโซน เลขที่นั่งและกรอกชื่อ-นามสกุลให้เรียบร้อย เมื่อตรวจสอบความถูกต้องเรียบร้อยแล้วจึงให้กดปุ่ม payment เพื่อไปยังหน้าชำระเงินต่อไป

1. โซน A มี 20 ที่นั่ง
2. โซน B มี 20 ที่นั่ง
3. โซน C มี 20 ที่นั่ง

- หน้าโซนยืน

Standing Zone

Buy Ticket

Zone
Zone A

Full name

Payment

ในส่วนของโซนยืนจะมีทั้งหมด 3 โซน จะต้องทำการจองบัตรคอนเสิร์ต โดยการเลือกโซนและกรอกชื่อ-นามสกุลให้เรียบร้อย ส่วนเลขที่นั่งจะเรียงตามลำดับของผู้จอง เมื่อตรวจสอบความถูกต้องเรียบร้อยแล้วจึงให้กดปุ่ม payment เพื่อไปยังหน้าชำระเงินต่อไป

1. โซน A มี 20 ที่นั่ง
 2. โซน B มี 20 ที่นั่ง
 3. โซน C มี 20 ที่นั่ง
4. หน้าชำระเงิน

เมื่อทำการจองจากหน้าจอเรียบร้อยแล้วจะเข้าสู่หน้าชำระเงิน กรอกข้อมูลตามฟอร์มให้เรียบร้อย จากนั้นจะไปเข้าหน้าแปะสลิปโอนเงิน เมื่อกรอกข้อมูลครบแล้วให้กดปุ่ม submit หลังจากการดำเนินการดังกล่าวเรียบร้อยแล้ว จะถือว่าการจองบัตรคอนเสิร์ตของผู้จองนั้นสำเร็จ

The image displays two screenshots of the TicketConcert website's payment interface.

Top Screenshot: The page is titled "Payment" and shows a form for "Bank xxxxx-xxxxx". The form includes the following fields:

- Zone: Zone A (sit) (dropdown menu)
- Seat: 1 (text input)
- Full name: (text input)
- Email: example@gmail.com (text input)
- Phone number: (text input)

 A blue "Submit" button is at the bottom. A warning message at the bottom reads: "* Warning! Please send us the payment slip after remittance *".

Bottom Screenshot: The page is also titled "Payment" and shows a form for "Bank xxxxx-xxxxx". The form includes the following fields:

- Choose File: No file chosen (button)

 A blue "Payment" button is at the bottom.

5. หน้าค้นหา

เมื่อจ่ายเงินเรียบร้อยแล้ว จึงสามารถค้นหาข้อมูลของผู้ที่ทำการซื้อบัตรเข้าชมงานคอนเสิร์ตได้ ซึ่งจะแสดงผลผ่านทางหน้าเว็บ

TicketConcert Home Search Contact

Check Booking

Zone
Zone A (sit)

Full name

* upper case NOT equal lower case *

Search

6. หน้าช่องทางติดต่อ

กรอกชื่อและนามสกุล อีเมลล์ ซึ่งมีระบบแจ้งเตือนผ่านทางอีเมลล์

TicketConcert Home Search Contact

Name
First name
Last name

Email
example@gmail.com

Message

Send

2.2 รายละเอียดข้อมูลที่เกี่ยวข้อง

จำลองการซื้อบัตรเข้าชมคอนเสิร์ต สามารถทำงานผ่านทางแพลตฟอร์มเว็บแอปพลิเคชันที่ได้จัดทำขึ้นมา โดยพัฒนาผ่าน Django framework เขียนร่วมกับ HTML, CSS และ Javascript ในส่วนของ Frontend

2.2.1 Django framework

ใช้สำหรับพัฒนา Web application ซึ่งเขียนในภาษา Python โดยมีเป้าหมายหลักในการทำให้การสร้าง website ที่ทำงานร่วมกับ database และมีความซับซ้อนให้ง่ายขึ้น โดยจะมีการพัฒนาด้วยภาษา python ตลอดทั้ง application ซึ่งรวมถึงการ setting, files ต่างๆ และ Data model ด้วย

2.2.2 HTML

ภาษา HTML จะทำหน้าที่ในการจัดการโครงสร้างและรูปร่างของเว็บไซต์

2.2.3 CSS

ภาษา CSS จะทำหน้าที่ตกแต่งหน้าเว็บไซต์ให้สวยงาม

2.2.4 Javascript

ภาษา Javascript จะทำหน้าที่เพิ่มความสามารถหรือเพิ่มคุณสมบัติพิเศษให้กับเว็บไซต์ ทำให้เว็บไซต์สามารถ interactive กับผู้ใช้งานหรือ user ได้ดีขึ้น

บทที่ 3

การทำงานของโปรแกรม

3.1 การทำงานของโปรแกรม

3.1.1 ส่วนของ Frontend

ใช้ภาษา HTML , CSS และ Javascript

3.1.2 ส่วนของ Backend

ใช้ภาษา Python ในส่วนของโปรแกรมหลัก พัฒนาผ่าน Django framework

3.2 โครงสร้างส่วนของข้อมูล

3.2.1 วิธีการจัดเก็บข้อมูล

Doubly-linked list ในการจัดเก็บข้อมูลผู้ซื้อบัตรแบบนั่ง ซึ่งจะมีข้อมูลในส่วนของการตรวจสอบโซน ที่นั่งและชื่อผู้จองว่ามีข้อมูลซ้ำกันหรือไม่

Queue doubly-linked list ใช้ในการเก็บข้อมูลของผู้ซื้อบัตรคอนเสิร์ตแบบยืน ที่นั่งมีการเรียงตามลำดับของผู้จอง

3.2.2 วิธีการจัดเรียงข้อมูล

Bubble-Sort จะมีการเรียงข้อมูลของผู้ที่ซื้อบัตรแบบยืนให้มีการเรียงตามเลขที่นั่ง เพื่อใช้ในระบบของหลังบ้านที่จะนำไปตรวจบัตรคอนเสิร์ตในสถานที่จริง ซึ่งแสดงผลทาง Terminal

3.3.3 วิธีการค้นหาข้อมูล

Search to front เมื่อจ่ายเงินเรียบร้อยแล้ว จึงสามารถค้นหาข้อมูลของผู้ที่ทำการซื้อบัตรเข้างานคอนเสิร์ตได้ ซึ่งจะแสดงผลผ่านทางหน้าเว็บ

Sentinel search จะตรวจสอบข้อมูลของผู้ซื้อบัตรคอนเสิร์ตของหลังบ้าน ซึ่งแสดงผลผ่านทาง Terminal

3.3 โค้ดของการทำงานส่วนโปรแกรมหลัก

3.3.1 ฟังก์ชันในหน้าทำการจอง

มี Function Reserve data ในการบันทึกและรับข้อมูลจากผู้จองบัตรในตัว Doubly-linked list ที่ชื่อว่า ReserveQ

```
@csrf_exempt
def addresserve(request):
    c_F = open("pko_C.pkl", "rb")
    Whalndloph = pickle.load(c_F)
    c_F.close()

    c_P = open("pko_P.pkl", "rb")
    PaymentQ = pickle.load(c_P)
    c_P.close()

    c_R = open("pko_R.pkl", "rb")
    ReserveQ = pickle.load(c_R)
    c_R.close()

    if request.method == 'POST':
        full_name = request.POST.get('full_name')
        number = request.POST.get('seat')
        zoner = request.POST.get('zone')

        ReserveQ.receiveData(full_name, zoner, number)
        print(full_name, number, zoner)
```

```
c_P = open("pko_P.pkl", "wb")
pickle.dump(PaymentQ, c_P)
c_P.close()

c_R = open("pko_R.pkl", "wb")
pickle.dump(ReserveQ, c_R)
c_R.close()

return render(request, 'payment.html')
```

3.3.2 ฟังก์ชันในหน้าชำระเงิน

มี Function Reserve data จาก class Payment ซึ่ง class Payment เป็น class ลูกของ class ClientDoublyLinkedlist จะทำงานโดยมีการรับข้อมูลจากผู้จองบัตรและตรวจสอบความถูกต้อง โดยนำไปตรวจสอบผ่าน Function SelectSearchData ซึ่งเป็นของ ReserveQ จะมีหน้าที่ตรวจสอบข้อมูลว่าข้อมูลมีอยู่ในรายชื่อการจองหรือไม่ และ SelectFindSameChair ซึ่งเป็นของ WhalnDloph (object ของ class concert) จะมีหน้าที่ตรวจสอบว่าที่นั่งนั้นถูกจองไปแล้วหรือไม่ ถ้าหากที่นั่งถูกจองไปแล้วหรือไม่มีรายชื่ออยู่ในรายชื่อผู้จอง จะไม่สามารถจ่ายเงินเพื่อจองบัตรได้

```
@csrf_exempt
def addpayment(request):
    c_F = open("pko_C.pkl","rb")
    WhalnDloph = pickle.load(c_F)
    c_F.close()

    c_P = open("pko_P.pkl","rb")
    PaymentQ = pickle.load(c_P)
    c_P.close()

    c_R = open("pko_R.pkl","rb")
    ReserveQ = pickle.load(c_R)
    c_R.close()

    global PaymentResult
    if request.method == 'POST':
        full_name2 = request.POST.get('full_name')
        number2 = request.POST.get('seat')
        zone2 = request.POST.get('zone')

    if number2 == "":
        number2 = None
    print(full_name2,number2,zone2)

    if number2 is not None : #เงื่อนไข
        PaymentResult = PaymentQ.recvData(full_name2,zone2,number2,ReserveQ.SelectSearchData(full_name2,zone2,number2),WhalnDloph.SelectFindSameChair(zone2,number2),timedata)
    else :
        PaymentResult = PaymentQ.recvData(full_name2,zone2,number2,ReserveQ.SelectSearchData(full_name2,zone2,number2),None,timedata)

    if PaymentResult == True :
        print("inside is working !!!!!")
        WhalnDloph.recvData(full_name2,zone2,number2)
        print("Paymentresult is :",PaymentResult)
        print(WhalnDloph)
        adminGetData(WhalnDloph)
        c_F = open("pko_C.pkl","wb")
        pickle.dump(WhalnDloph,c_F)
        c_F.close()
```

```
c_P = open("pko_P.pkl","wb")
pickle.dump(PaymentQ,c_P)
c_P.close()

c_R = open("pko_R.pkl","wb")
pickle.dump(ReserveQ,c_R)
c_R.close()
return render(request,'paymentbill.html')
else:
    print("inside is44444 working !!!!!")
    c_F = open("pko_C.pkl","wb")
    pickle.dump(WhalnDloph,c_F)
    c_F.close()

    c_P = open("pko_P.pkl","wb")
    pickle.dump(PaymentQ,c_P)
    c_P.close()

    c_R = open("pko_R.pkl","wb")
    pickle.dump(ReserveQ,c_R)
    c_R.close()
    return render(request,'payment.html')
```

3.3.3 ฟังก์ชันในหน้า Search

มีการทำงานผ่านฟังก์ชัน SelectSearchDataToFront ของ WhalnDloph (Object ของ class concert) ซึ่งมีหน้าที่ในการค้นหาข้อมูลและเปลี่ยนตำแหน่งของข้อมูลนั้นมาไว้ด้านหน้าสุดของ Doubly-LinkedList

```
def search_user(request):

    c_F = open("pko_C.pkl", "rb")
    WhalnDloph = pickle.load(c_F)
    c_F.close()

    c_P = open("pko_P.pkl", "rb")
    PaymentQ = pickle.load(c_P)
    c_P.close()

    c_R = open("pko_R.pkl", "rb")
    ReserveQ = pickle.load(c_R)
    c_R.close()

    full_nameS = None
    zonerS = None

    if request.method == 'POST':
        full_nameS = request.POST.get('full_name')
        zonerS = request.POST.get('zone')
        print(full_nameS , zonerS)

    c_F = open("pko_C.pkl", "wb")
    pickle.dump(WhalnDloph, c_F)
    c_F.close()
```

```
c_P = open("pko_P.pkl", "wb")
pickle.dump(PaymentQ, c_P)
c_P.close()

c_R = open("pko_R.pkl", "wb")
pickle.dump(ReserveQ, c_R)
c_R.close()

try :
    h = WhalnDloph.SelectSearchDataToFront(full_nameS, zonerS)
    if type(h) == ClientNode :
        print("is ClientNode")
        full_nameS = None
        zonerS = None
        return render(request, 'searchsuccess.html', {"name": h.data, "zone": h.zonedata, "number": h.numberdata})
    elif type(h) == dict :
        print("is dict")
        full_nameS = None
        zonerS = None
        return render(request, 'searchsuccess.html', {"name": h['name'], "zone": h['zone'], "number": h['number']})
    else:
        messages.info(request, 'wtf')
        return redirect('/search')
except AttributeError :
    messages.info(request, 'wtf')
    return redirect('/search')
```

3.3.4 ฟังก์ชัน Bubblesortlist

มีการทำงานใน Terminal เพื่อใช้สำหรับ Admin หรือพนักงานในการตรวจสอบข้อมูล ณ สถานที่จริงโดยข้อมูลจะมีการเรียงจากค่าที่น้อยหรือเลขบัตรจากน้อยไปมาก

```
def bubblesortList(self):
    setLastGo = None
    swapped = False
    ''' Checking for empty list '''
    if (self.head == None):
        return
    while True:
        swapped = False
        h = self.head
        while (h.next != setLastGo):
            if (int(h.numberdata) > int(h.next.numberdata)):
                h.data , h.zonedata , h.numberdata , h.next.data , h.next.zonedata , h.next.numberdata = h.next.data , h.next.zonedata , h.next.numberdata , h.data , h.zonedata , h.numberdata
                swapped = True
            h = h.next
        setLastGo = h
        if swapped == False:
            break
```

3.3.5 ฟังก์ชัน BubblesortlistHtl

มีการทำงานใน Terminal เพื่อใช้สำหรับ Admin หรือพนักงานในการตรวจสอบข้อมูล ณ สถานที่จริงโดยข้อมูลจะมีการเรียงจากค่าที่น้อยหรือเลขบัตรจากมากไปน้อย

```
def bubblesortListHtl(self):
    setLastGo = None
    swapped = False
    ''' Checking for empty list '''
    if (self.head == None):
        return
    while True:
        swapped = False
        h = self.head
        while (h.next != setLastGo):
            if (int(h.numberdata) < int(h.next.numberdata)):
                h.data , h.zonedata , h.numberdata , h.next.data , h.next.zonedata , h.next.numberdata = h.next.data , h.next.zonedata , h.next.numberdata , h.data , h.zonedata , h.numberdata
                swapped = True
            h = h.next
        setLastGo = h
        if swapped == False:
            break
```

```
def bubblesortList(self):
    #Check whether list is empty
    if self.head is None:
        return
    else:
        h = self.head
        while h.next is not None:
            hN = h.next
            while(hN != None):
                if(int(h.numberdata) > int(hN.numberdata)):
                    numtemp = h.numberdata
                    nametemp = h.data
                    h.numberdata = hN.numberdata
                    h.data = hN.data
                    hN.numberdata = numtemp
                    hN.data = nametemp
                else :
                    break
            hN = hN.next
        h = h.next
```

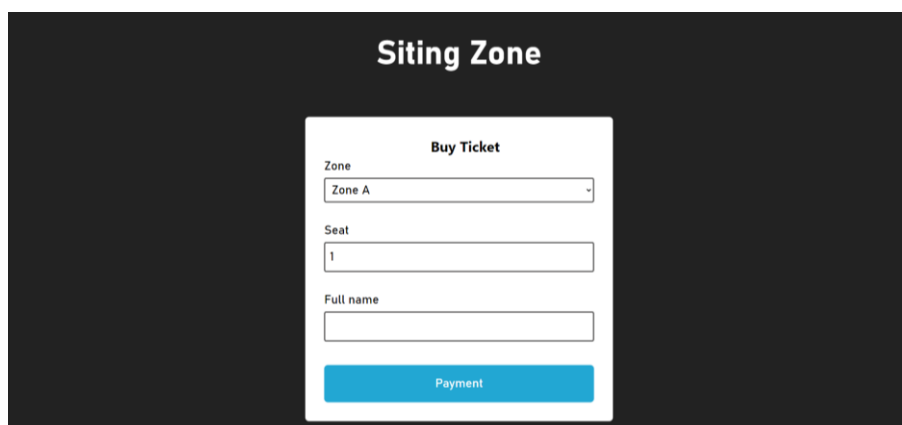
บทที่ 4

ผลการทำงานของโปรแกรม

ผลการทำงานของส่วนโปรแกรมหลัก

4.1 โซนนั่ง

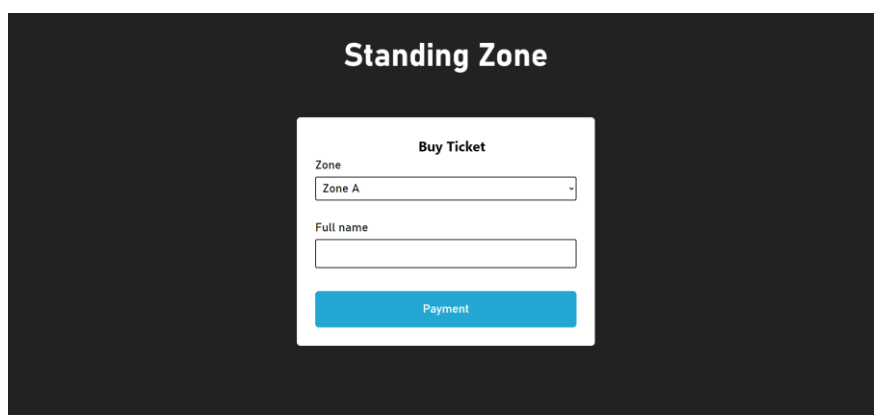
กรอกโซน ที่นั่งและชื่อนามสกุลให้เรียบร้อย



The screenshot shows a web form titled "Siting Zone" on a dark background. The form itself is white and titled "Buy Ticket". It contains three input fields: a dropdown menu for "Zone" with "Zone A" selected, a text box for "Seat" with the number "1" entered, and a text box for "Full name". Below these fields is a blue button labeled "Payment".

4.2 โซนยืน

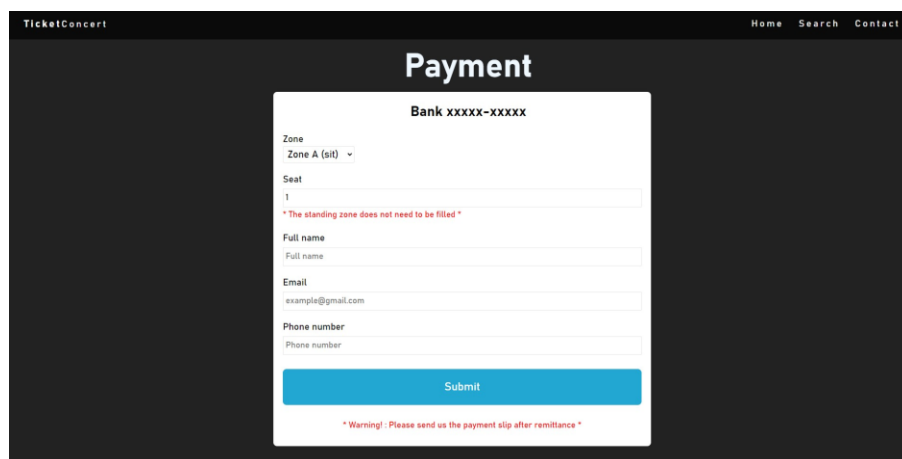
กรอกโซนและชื่อนามสกุลให้เรียบร้อย ส่วนที่นั่งจะเรียงตามลำดับของผู้จองบัตร



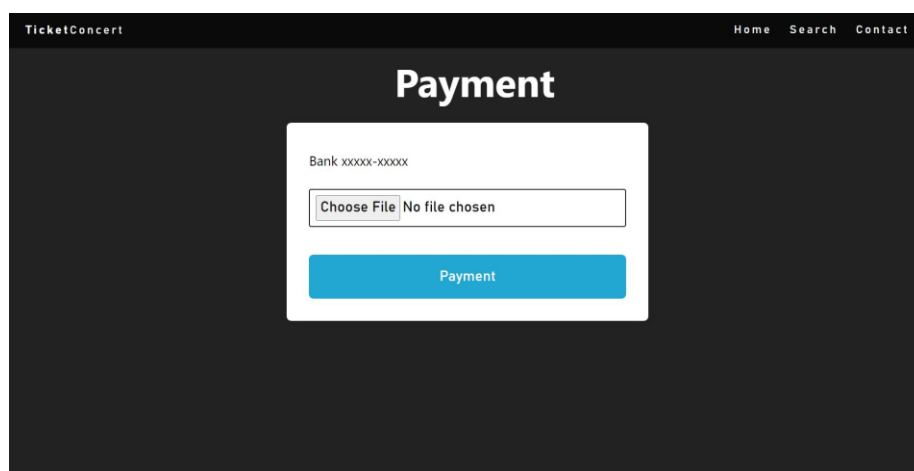
The screenshot shows a web form titled "Standing Zone" on a dark background. The form is white and titled "Buy Ticket". It contains two input fields: a dropdown menu for "Zone" with "Zone A" selected, and a text box for "Full name". Below these fields is a blue button labeled "Payment".

4.3 หน้าชำระเงิน

เมื่อทำการจองจากหน้าจองเรียบร้อยแล้วจะเข้าสู่หน้าชำระเงิน กรอกข้อมูลตามฟอร์มให้เรียบร้อย จากนั้นจะไปเข้าหน้าแปะสลิปโอนเงิน เมื่อกรอกข้อมูลครบแล้วให้กดปุ่ม submit หลังจากการดำเนินการดังกล่าวเรียบร้อยแล้ว จะถือว่าการจองบัตรคอนเสิร์ตของผู้จองนั้นสำเร็จ



The screenshot shows the 'Payment' page of the TicketConcert website. The page has a dark header with 'TicketConcert' on the left and 'Home Search Contact' on the right. The main heading is 'Payment'. Below it, the bank information is 'Bank xxxxx-xxxxx'. The form contains the following fields: 'Zone' with a dropdown menu showing 'Zone A (sit)', 'Seat' with the value '1', 'Full name' with the placeholder 'Full name', 'Email' with the placeholder 'example@gmail.com', and 'Phone number' with the placeholder 'Phone number'. A red warning message states: '* The standing zone does not need to be filled *'. At the bottom of the form is a blue 'Submit' button. Below the button, another red warning message states: '* Warning! : Please send us the payment slip after remittance *'.

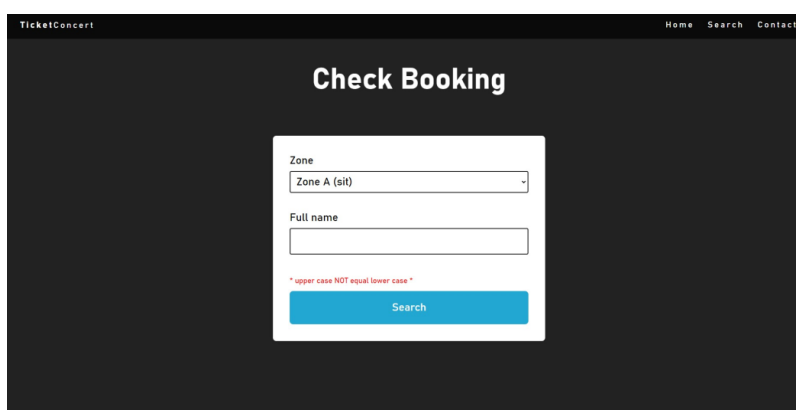


The screenshot shows the 'Payment' page of the TicketConcert website. The page has a dark header with 'TicketConcert' on the left and 'Home Search Contact' on the right. The main heading is 'Payment'. Below it, the bank information is 'Bank xxxxxx-xxxxx'. The form contains a file upload field with a 'Choose File' button and the text 'No file chosen'. At the bottom of the form is a blue 'Payment' button.

4.4 หน้าค้นหา

เมื่อจ่ายเงินเรียบร้อยแล้ว จึงสามารถค้นหาข้อมูลของผู้ที่ทำการซื้อบัตรเข้าชมงานคอนเสิร์ตได้ ซึ่งจะแสดงผลผ่านทางหน้าเว็บ

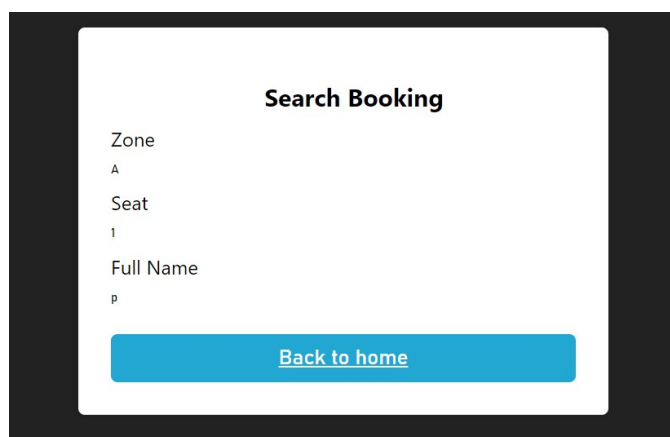
ช่องค้นหา



The screenshot shows a web interface for 'TicketConcert' with a dark header containing 'Home', 'Search', and 'Contact' links. The main heading is 'Check Booking'. Below it is a white form box with the following elements:

- A 'Zone' dropdown menu currently showing 'Zone A (sit)'.
- A 'Full name' text input field.
- A red error message: '* upper case NOT equal lower case *'.
- A blue 'Search' button.

แสดงผล



The screenshot shows a web interface with a dark header. The main heading is 'Search Booking'. Below it is a white box containing the search results:

- Zone: A
- Seat: 1
- Full Name: p

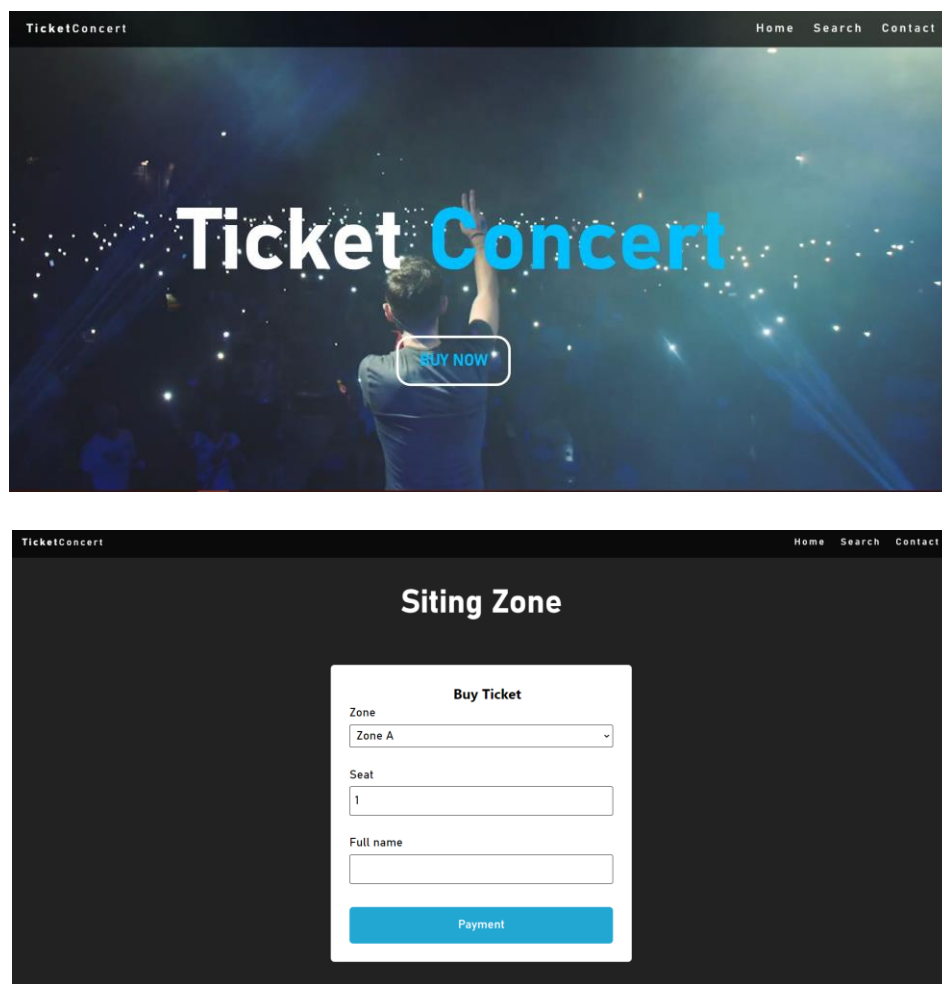
At the bottom of the white box is a blue button labeled 'Back to home'.

บทที่ 5

สรุปผลการทำงาน

สรุปผลการทำงานของโปรแกรม

โปรแกรมจำลองการซื้อขายบัตรเข้าชมคอนเสิร์ต โดยสามารถทำงานผ่านทางแพลตฟอร์มเว็บแอปพลิเคชัน ซึ่งโปรแกรมสามารถรับอินพุตจากผู้ใช้และแสดงผลผ่านจอภาพได้



ตัวอย่างหน้าเว็บ

การมีส่วนร่วมของสมาชิก

1. นางสาวนันทิชา สุขเจริญ 63010484

หน้าที่ : Frontend และคิดเป็นเปอร์เซ็นต์การมีส่วนร่วม 100%

2. นางสาวนภสร ชาลานุมาศ 63010492

หน้าที่ : Backend และคิดเป็นเปอร์เซ็นต์การมีส่วนร่วม 100%

3. นายปุณณวิชญ์ พานิชผล 63010616

หน้าที่ : Backend และคิดเป็นเปอร์เซ็นต์การมีส่วนร่วม 100%

4. นางสาวพัฒนัชดา อีร์พัฒนโรจน์ 63010665

หน้าที่ : Frontend และคิดเป็นเปอร์เซ็นต์การมีส่วนร่วม 100%

5. นายพีระภัทร์ เศรษฐพรนรา 63010702 (หัวหน้ากลุ่ม) B

หน้าที่ : Backend และคิดเป็นเปอร์เซ็นต์การมีส่วนร่วม 100%

เอกสารอ้างอิง

Codebee. (2559). เขียน HTML, CSS และ Javascript, , สืบค้นเมื่อ 26 ตุลาคม 2564. จาก.

<https://www.codebee.co.th/labs/%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B9%80%E0%B8%82%E0%B8%B5%E0%B8%A2%E0%B8%99-html-%E0%B9%81%E0%B8%A5%E0%B8%B0-css/>

KongRuksiam Official. (2564). เขียนเว็บภาษา Python ด้วย Django Framework เบื้องต้น, สืบค้นเมื่อ 26 ตุลาคม 2564. จาก. https://youtu.be/no1oR9_fc6s

KongRuksiam Official. (2564). พื้นฐานการสร้างเว็บด้วย HTML5 จบในคลิปเดียว, สืบค้นเมื่อ 27 ตุลาคม 2564. จาก. <https://youtu.be/0hfeNPM7piw>

GeeksforGeeks. (2564). Pass data to javascript in Django Framework, สืบค้นเมื่อ 15 พฤศจิกายน 2564. จาก. <https://www.geeksforgeeks.org/how-to-pass-data-to-javascript-in-django-framework/>

W3school. (2564). HTML, CSS และ Javascript, สืบค้นเมื่อ 1 พฤศจิกายน 2564. จาก. <https://www.w3schools.com/html/default.asp>