



MUICT Student Room Booking System

โปรแกรมสำหรับการจองห้องเรียนของนักศึกษาคณะ ICT

เสนอ

อาจารย์ ดร. จิตภา ไกรสังข์

อาจารย์ ดร. อัคร สุประทักษ์

อาจารย์ ดร. ทิพาจินต์ ไทยพิสุทธิกุล

จัดทำโดย

นาย	จตุโชติ	เพ็งปาน	รหัสนักศึกษา	6687010
นาย	นวดล	สมบุญกุล	รหัสนักศึกษา	6687026
นางสาว	รวีพลอย	เจริญชัยประกิจ	รหัสนักศึกษา	6687042
นาย	วัฒนชัย	บุญไชย	รหัสนักศึกษา	6687045

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา ทสวต 120 การเขียนโปรแกรมขั้นพื้นฐาน

หลักสูตรวิทยาการและเทคโนโลยีดิจิทัล คณะเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยมหิดล

ภาคเรียนที่ 1 ปีการศึกษา 2566

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา ทสวต 120 การเขียนโปรแกรมขั้นพื้นฐาน จัดทำขึ้นโดยมีวัตถุประสงค์เพื่ออธิบายการทำงานของโครงงาน “ MUICT Student Room Booking System โปรแกรมสำหรับการจองห้องเรียนของนักศึกษาคณะ ICT ” โดยจะประกอบไปด้วย รายละเอียดฟังก์ชันที่ใช้ในการทำงานและการทำงานของโปรแกรม เพื่อให้ผู้ที่มีความสนใจในโปรแกรมห่วงสามารถใช้งานและเข้าใจการทำงานของโปรแกรมนี้ได้

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงานฉบับนี้จะเป็นประโยชน์ต่อผู้อ่านในการใช้งานโปรแกรมนี้อย่างเข้าใจ หากมีข้อผิดพลาดประการใด ต้องขออภัยไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ

๒๔ พฤศจิกายน ๒๕๖๖

สารบัญ

เรื่อง	หน้า
คำนำ	ก
สารบัญ	๗
รายละเอียดการทำงานของฟังก์ชัน	1
1. Module / Library	1
2. ฟังก์ชันทั่วไป	1
- ฟังก์ชันการตรวจสอบไฟล์ Studentbooking.csv	1
- ฟังก์ชันแสดงผลหน้าจอ UI และอ่านไฟล์ CSV	2
- ฟังก์ชันการตรวจสอบรหัสนักศึกษา	2
- ฟังก์ชันการตรวจสอบวันที่	3
3. ฟังก์ชันหลัก	4
- ฟังก์ชันที่1 แสดงรายชื่อของนักศึกษาในโปรแกรม	4
- ฟังก์ชันที่2 ส่งคำร้องการจองห้องเรียน	5
- ฟังก์ชันที่3 ตรวจสอบว่าประวัติการจองของห้องเรียน	8
- ฟังก์ชันที่4 ตรวจสอบห้องเรียนที่ว่างอยู่โดยดูจากวันที่	9
- ฟังก์ชันที่5 ตรวจสอบการจองของห้องเรียนด้วยรหัสนักศึกษา	10
- ฟังก์ชันที่6 ตรวจสอบการจองของห้องเรียนด้วยชื่อจริง	12
- ฟังก์ชันที่7 แสดงสรุปการของห้องเรียนของนักศึกษา	13
4. หน้าจอแสดงผล	15

รายละเอียดการทำงานของฟังก์ชัน

“MUICT Student Room Booking System”

โปรแกรมสำหรับการจองห้องเรียนของนักศึกษาคณะ ICT

1. Module / Library

```
1 # Module
2 import os
3 import csv
4 import pandas as pd
5 import numpy as np
```

มีการเรียกใช้ Module OS และ CSV เพื่อนำไปใช้ในการตรวจสอบไฟล์ CSV และ เรียกใช้ Module ของ pandas เพื่อนำไฟล์ CSV มาแสดงข้อมูล ตรวจสอบข้อมูล และบันทึกข้อมูล

2. ฟังก์ชันทั่วไป

2.1 ฟังก์ชันการตรวจสอบไฟล์ Studentbooking.csv

```
7 # ตรวจสอบไฟล์ booking ไม่พบไฟล์ว่างใหม่
8 def check_file_exist():
9     global booking_df
10    header = ['id', 'fname', 'lname', 'roomtype', 'room', 'bookingdate', 'sorted_date']
11    if not os.path.exists('studentbooking.csv'):
12        with open("studentbooking.csv", "w", newline="") as file:
13            newfile = csv.writer(file)
14            newfile.writerow(header)
15    else:
16        booking_df = pd.read_csv("studentbooking.csv")
```

ฟังก์ชันจะตรวจสอบว่ามีไฟล์ studentbooking.csv อยู่หรือไม่ หากไม่มีไฟล์อยู่ฟังก์ชันจะทำการสร้างไฟล์ชื่อ studentbooking.csv ขึ้นมา แต่หากพบว่ามีไฟล์อยู่แล้ว ฟังก์ชันจะทำการอ่านไฟล์และนำข้อมูลมาเก็บไว้ในตัวแปร booking_df

2.2 ฟังก์ชันแสดงผลหน้าจอ UI และอ่านไฟล์ CSV

```

18 # ฟังก์ชันแสดงผลหน้าจอโปรแกรม '''
19 def UI():
20     global student_df
21     student_df = pd.read_csv("students.csv")
22     check_file_exist()
23     print("=====")
24     print("          Student Room Booking System ")
25     print("=====")
26     print(" 1  Print a list of students")
27     print(" 2  Submit a booking request")
28     print(" 3  Check the current booking via room number")
29     print(" 4  Check the available rooms via date")
30     print(" 5  Check booking with student ID")
31     print(" 6  Check booking with student first name")
32     print(" 7  Print booking summary")
33     print(" 0  Exit")
34     print("=====")

```

ฟังก์ชันจะทำการอ่านไฟล์students.csvและเรียกใช้ฟังก์ชันcheck_file_exist (สำหรับการตรวจสอบเงื่อนไขของไฟล์studentbooking.csv) หลังจากนั้นจะทำการแสดงผลUI สำหรับผู้ใช้โปรแกรม

2.3 ฟังก์ชันการตรวจสอบรหัสนักศึกษา

```

36 #ฟังก์ชันตรวจสอบรหัสนักศึกษา '''
37 def checkvalid_IDstudent(ID_student):
38     if ID_student in student_df["id"].values:
39         return True

```

ฟังก์ชันจะรับparameter จำนวน 1 ตัว ได้แก่ ID_student เพื่อนำมาเช็คว่ามีรหัสนักศึกษานี้อยู่ในไฟล์ student.csvหรือไม่ หากพบว่ามีรหัสนักศึกษา จะทำการส่งค่า True กลับไป

2.3 ฟังก์ชันการตรวจสอบวันที่

```

41 # ฟังก์ชันตรวจสอบวันที่ '''
42 def checkvalid_date(Bookingdate, month_names):
43     check = Bookingdate.split("-")
44     if len(check) == 3:
45         for items in check:
46             if not items.isdigit():
47                 print("Incorrect date format, should be DD-MM-YYYY")
48                 return True
49     else:
50         days, month, years = map(int, check)
51         if month < 1 or month > 12:
52             print("Please enter a month between 1-12 (January-December)")
53             return True
54         elif years > 9999 :
55             print("Incorrect date format, should be DD-MM-YYYY")
56             return True
57         elif month in [1, 3, 5, 7, 8, 10, 12] and days > 31:
58             print(f"On {month_names[month]} there are only 31 days")
59             return True
60         elif month in [4, 6, 9, 11] and days > 30:
61             print(f"On {month_names[month]} there are only 30 days")
62             return True
63         elif month == 2:
64             if (years % 400 == 0 or years % 4 == 0 and years % 100 != 0):
65                 if days > 29:
66                     print(f"Leap years have only 29 days")
67                     return True
68             elif days > 28:
69                 print(f"{month_names[month]} in this year has only 28 days")
70                 return True
71     else:
72         print("Incorrect date format, should be DD-MM-YYYY")
73     return True

```

ฟังก์ชันจะรับ parameter จำนวน 2 ตัว ได้แก่ Bookingdate และ month_names หลังจากนั้นจะนำค่า Bookingdate ไปเช็คว่าระบุวันที่ได้ถูกต้องหรือไม่ หากไม่ถูกต้องจะมีการแจ้งว่าระบุวันที่ผิดพลาดอย่างไรโดยการนำตัวแปร month_names มาอธิบาย หลังจากนั้นจึงจะทำการส่งค่า True กลับไป

2.3 ฟังก์ชันแปลงรูปแบบวันที่ (dd-mm-yyyy)

```

74 # สร้าง format dd-mm-yyyy
75 def make_ddmmyyyy_format(Bookingdate):
76     day,month,year = Bookingdate.split("-")
77     if len(day) == 1:
78         day = f"0{day}"
79     if len(month) == 1:
80         month = f"0{month}"
81     Bookingdate = f"{day}-{month}-{year}"
82     Sorted_date = f"{year}-{month}-{day}"
83     return Bookingdate,Sorted_date
84

```

3. ฟังก์ชันหลัก

3.1 ฟังก์ชันที่1 แสดงรายชื่อของนักศึกษาในโปรแกรม (print a list of students)

```

85 # ฟังก์ชันแสดงรายชื่อของนักศึกษา
86 def func1_liststudentname():
87     print(student_df)
88

```

ฟังก์ชันจะแสดงข้อมูลรายชื่อของนักศึกษาจากไฟล์student.csvที่เก็บไว้ในตัวแปรที่ชื่อว่า student_df

ตัวอย่าง Output :

```

Option: 1

```

	id	fname	lname
0	6687001	Keatikun	Komkeng
1	6687002	Techit	Thititammajariya
2	6687003	Taechinee	Ratanawimon
3	6687004	Krittanon	Chongklahan
4	6687005	Kongpop	Keattireangwattana
5	6687006	Kongphop	Kaochot
6	6687007	Kanyanat	Uetrakulchai
7	6687008	Kantanat	Chamnien
8	6687009	Jirayu	Saisawat
9	6687010	Jutichot	Phengpan
10	6687011	Chanathip	Chaowaphong
11	6687012	Chitipat	Pangma
12	6687013	Yatavee	Wariyot
13	6687014	Tapanut	Aunpikul
14	6687015	Thitaree	Kongkaew
15	6687016	Thitiporn	Panthanan
16	6687017	Nutt	Saipipan
17	6687018	Natnicha	Malailoy
18	6687019	Daraporn	Saepoo
19	6687020	Taned	Somchirvieng

3.2 ฟังก์ชันที่2 ส่งคำร้องการจองห้องเรียน (submit a booking request)

```

89 def func2_booking(LAB, LECTURE):
90     ID_student = int(input("ID: "))
91     if checkvalid_IDstudent(ID_student):
92         Roomtype = input("Roomtype (Lecture / Lab): ").upper()
93         #รับประเภทของห้องเรียนที่จอง
94         while True:
95             if Roomtype != 'LECTURE' and Roomtype != 'LAB':
96                 print("Please enter Roomtype Lecture or Lab")
97                 Roomtype = input("Roomtype (Lecture / Lab): ").upper()
98             else:
99                 break
100         if Roomtype == 'LECTURE':
101             print(f"Room: {LECTURE}")
102         elif Roomtype == 'LAB':
103             print(f"Room: {LAB}")
104         Room = input("Please select one room above: ").upper()
105         #รับห้องเรียนที่จอง
106         while True:
107             if (Roomtype == 'LECTURE') and Room not in LECTURE:
108                 Room = input("Please enter a room according to the Lecture Room: ").upper()
109             elif (Roomtype == 'LAB') and Room not in LAB:
110                 Room = input("Please enter a room according to the Lab Room: ").upper()
111             else:
112                 break

```

ฟังก์ชันนี้จะถูกเรียกใช้สำหรับจองห้องเรียน(Lecture) หรือห้องแล็บ(Lab) โดยฟังก์ชันจะรับค่ารหัสนักศึกษาจากก่อนโดยเอารหัสนักศึกษามาเช็กับไฟล์ student.csv ในตัวแปรที่มีชื่อว่า "ID_student"

ตัวอย่างการรับค่า :

```

Option: 2
ID: 

```

หากรหัสนักศึกษารับมามีค่าตรงกับข้อมูลที่มีอยู่ภายในไฟล์student.csvแล้ว โปรแกรมจะรับค่าประเภทของห้องที่ต้องการจองอย่างต่อเนื่อง ซึ่งค่าที่จะรับมีเพียง ห้องเรียน(Lecture) หรือ ห้องแล็บ(Lab) เท่านั้น

ตัวอย่างการรับค่า :

```

ID: 6687001
Roomtype (Lecture / Lab): 

```

แต่ถ้ารหัสนักศึกษาที่ระบุมานั้นไม่ตรงกับข้อมูลที่มีอยู่โปรแกรมจะวนรับค่ารหัสนักศึกษาใหม่จนกว่าข้อมูลที่ได้จะถูกต้องและตรงกับข้อมูลที่มีอยู่ภายในไฟล์

หลังจากนั้น โปรแกรมจะทำการแสดงชื่อของห้องประเภทที่เลือกและรับค่า เลขห้อง ที่นักศึกษาต้องการจอง

ตัวอย่างการรับค่า :

```
Roomtype (Lecture / Lab): Lab
Room: ['LAB103', 'LAB104', 'LAB105', 'LAB106']
Please select one room above: █
```

เมื่อรับค่า เลขห้อง ซึ่งถูกต้องตามชื่อที่โปรแกรมระบุไว้แล้ว โปรแกรมจะรับค่า วัน-เดือน-ปี ที่นักศึกษาต้องการจอง

ตัวอย่างการรับค่า :

```
Please select one room above: LAB103
Booking Date (DD-MM-YYYY): █
```

```
113     Bookingdate = input("Booking Date (DD-MM-YYYY): ")
114     #รับวันที่จนกว่าจะถูก
115     while True:
116         if checkvalid_date(Bookingdate, month_names):
117             print("Please try again!!!")
118             Bookingdate = input("Booking Date (DD-MM-YYYY): ")
119         else:
120             break
121     #แปลง format ของวันที่
122     if make_ddmmyyyy_format(Bookingdate):
123         Bookingdate, Sorted_date = make_ddmmyyyy_format(Bookingdate)
124     for _,items in booking_df.iterrows():
125         # ตรวจสอบว่าห้องในวันที่รับเข้ามาถูกจองไปหรือยัง
126         if Room == items['room'] and Bookingdate == items['bookingdate']:
127             print(f'''
128=====
129 {Room} is not available on {Bookingdate}
130=====
131             ''')
132     return
```

ถ้าในวัน เดือน ปีที่ระบุห้องที่นักศึกษาต้องการจองนั้นถูกจองแล้วโปรแกรมจะแสดงผลว่า “Room is not available on {วันที่จอง}”

ตัวอย่าง Output :

```
=====
LAB103 is not available on 01-12-2023
=====
```

```

134     else:
135         #บันทึกการจอง studentbooking.csv
136         for _, value in student_df.iterrows():
137             if ID_student == value['id']:
138                 fname = value['fname']
139                 lname = value['lname']
140                 newdata = {
141                     'id': ID_student,
142                     'fname': fname,
143                     'lname': lname,
144                     'roomtype': Roomtype,
145                     'room': Room,
146                     'bookingdate': Bookingdate,
147                     'sortingdate': Sorted_date
148                 }
149                 booking_data = pd.DataFrame(newdata, index=[0])
150                 booking_data.to_csv("studentbooking.csv", mode="a", header=False, index=False)
151                 print(f'''
152 =====
153 Booking Successfully
154 =====
155 ID:{ID_student}
156 Firstname: {fname}
157 Lastname: {lname}
158 Roomtype: {Roomtype}
159 Room: {Room}
160 Booking Date: {Bookingdate}
161 =====
162 ''')
163     else:
164         print("You can't booking")

```

หากห้องที่นักศึกษาต้องการจองในวันที่ระบุนั้นยังไม่ถูกจอง โปรแกรมจะแสดงผลเป็นรหัส
นักศึกษาชื่อของนักศึกษาและวันที่จอง

ตัวอย่าง Output :

```

=====
Booking Successfully
=====
ID:6687001
Firstname: Keatikun
Lastname: Komkeng
Roomtype: LAB
Room: LAB103
Booking Date: 24-11-2023
=====

```

3.3 ฟังก์ชันที่3 ตรวจสอบประวัติการจองด้วยห้องเรียน

(check the current booking via room number)

```

166 ฟังก์ชันตรวจสอบการจองห้องเรียน
167 def func3_checkwithroom(LECTURE, LAB):
168     booking_file = booking_df.sort_values(by='id')
169     Room = input("Room: ").upper()
170     # รับค่าห้องเรียนจากผู้ใช้
171     while Room not in LECTURE and Room not in LAB:
172         print(f"Room {Room} does not exist in the system.")
173         print(f"Lecture Rooms: {LECTURE}")
174         print(f"Lab Rooms: {LAB}")
175         print("Please select one room above.")
176         Room = input("Room: ").upper()
177     print("Current Booking: ")
178     for _, value in booking_file.iterrows():
179         if Room == value['room']:
180             print(f"    Date: {value['bookingdate']} Student ID: {value['id']}")
181     if Room not in booking_file['room'].values:
182         print("    No booking")

```

ฟังก์ชันสำหรับตรวจสอบการจองห้องด้วยเลขห้อง โดยโปรแกรมจะรับค่า “เลขห้อง” และนำค่านั้นไปตรวจสอบในไฟล์ student.csv ที่ตัวแปร “room” หากใส่เลขห้องไม่ถูกต้อง โปรแกรมจะทำการวนรับค่าใหม่

ตัวอย่างการรับค่า :

```

Option: 3
Room: 

```

ถ้าห้องที่ระบุมานั้นมีประวัติการจอง โปรแกรมจะแสดงวัน เดือน ปี ที่ถูกจองและรหัสนักศึกษาที่จองห้องดังกล่าว

ตัวอย่าง Output :

```

Current Booking:
Date: 01-12-2023 Student ID: 6687001
Date: 12-10-2023 Student ID: 6687004
Date: 16-07-2023 Student ID: 6687013

```

หากห้องที่ระบุมาไม่มีประวัติการจองใดๆ โปรแกรมจะแสดงผลว่า “No booking”

ตัวอย่าง Output :

```

Current Booking:
No booking

```

3.4 ฟังก์ชันที่4ตรวจสอบห้องเรียนที่ว่างอยู่โดยดูจากวันที่ (check the available rooms via date)

```

def func4_checkwithdate():
    lefLECTURE = np.array(['IT381', 'IT382', 'IT383', 'IT384'])
    lefLAB = np.array(['LAB183', 'LAB184', 'LAB185', 'LAB186'])
    #รับวันที่จองห้องเรียน
    Bookingdate = input("Booking Date (DD-MM-YYYY): ")
    while True:
        if checkvalid_date(Bookingdate, month_names):
            print("Please try again")
            Bookingdate = input("Booking Date (DD-MM-YYYY): ")
        else:
            break
    #ปรับ format วันที่
    if make_ddmmyyyy_format(Bookingdate):
        Bookingdate, Sorted_date = make_ddmmyyyy_format(Bookingdate)
    #ดึงข้อมูลห้องเรียนที่จอง (LECTURE) ตามวันที่จองเข้ามา
    booked_lecture_rooms = booking_df.loc[(booking_df['bookingdate'] == Bookingdate) & (booking_df['roomtype'] == 'LECTURE'), 'room'].values
    #ดึงข้อมูลห้องเรียนที่จอง (LAB) ตามวันที่จองเข้ามา
    booked_lab_rooms = booking_df.loc[(booking_df['bookingdate'] == Bookingdate) & (booking_df['roomtype'] == 'LAB'), 'room'].values
    #ตรวจสอบว่ามีห้องเรียนที่จองหรือไม่ ถ้ามีไปลบ lefLECTURE
    if len(booked_lecture_rooms) > 0:
        lefLECTURE = np.setdiff1d(lefLECTURE, booked_lecture_rooms, assume_unique=True)
    #ตรวจสอบว่ามีห้องเรียนที่จองหรือไม่ ถ้ามีไปลบ lefLAB
    if len(booked_lab_rooms) > 0:
        lefLAB = np.setdiff1d(lefLAB, booked_lab_rooms, assume_unique=True)
    print("Available Room : ")
    print("# Lecture: {lefLECTURE}")
    print("# Lab: {lefLAB}")

```

ฟังก์ชันสำหรับการตรวจสอบประวัติการจองห้องด้วยวัน เดือน ปี โดยโปรแกรมจะรับค่าวัน-เดือน-ปี และนำไปเช็คในไฟล์ studentbooking.csvก่อนจะทำการตรวจสอบรูปแบบการเรียงของวันเดือน ปี ที่ระบุมาว่าถูกต้องหรือไม่

ตัวอย่างการรับค่า :

```

Option: 4
Booking Date (DD-MM-YYYY): 

```

หากไม่ถูกต้องตามรูปแบบดังกล่าวโปรแกรมจะแสดงผลว่า “Please try again”และวนรับค่าใหม่อีกครั้ง

ตัวอย่างการรับค่า :

```

Option: 4
Booking Date (DD-MM-YYYY): 12-23-2022
Please enter a month between 1-12 (January-December)
Please try again
Booking Date (DD-MM-YYYY): 

```

ในกรณีที่ใส่ถูกต้องตามรูปแบบการเรียงที่กำหนดไว้แล้วโปรแกรมจะแสดงผลว่า “Available Room: {เลขห้องที่ว่างทั้งหมด}”

ตัวอย่าง Output :

```
Booking Date (DD-MM-YYYY): 12-10-2023
Available Room:
Lecture: ['IT302', 'IT304']
Lab: ['LAB103', 'LAB104', 'LAB105', 'LAB106']
```

3.5 ฟังก์ชันที่ 5 ตรวจสอบการจองของห้องเรียนด้วยรหัสนักศึกษา (check booking with student ID)

```
213# ตรวจสอบการจองด้วยรหัสนักศึกษา
214def func5_checkwithID():
215    booking_file = booking_df.sort_values(by='sorted_date')
216    ID_student = int(input("Student ID: "))
217    if checkvalid_IDstudent(ID_student):
218        print("Current Booking: ")
219        for _, values in booking_file.iterrows():
220            if ID_student == values['id']:
221                print(f"Room: {values['room']} Date: {values['bookingdate']}")
222            elif ID_student not in booking_file['id'].values:
223                print("No Booking")
224            return
225    else:
226        print("ID student not match on students.csv file")
```

ฟังก์ชันสำหรับการตรวจสอบประวัติการจองห้องด้วยรหัสนักศึกษา โดยโปรแกรมจะรับค่ารหัสนักศึกษา ก่อนจะนำไปตรวจสอบข้อมูลภายในไฟล์ studentbooking.csv ที่ตัวแปร "id"

ตัวอย่างการรับค่า :

```
Option: 5
Student ID: █
```

หากค่าที่นำไปตรวจสอบนั้นไม่ตรงกับข้อมูลภายในไฟล์ โปรแกรมจะแสดงผลว่า "ID student not match on students.csv file"

ตัวอย่าง Output :

```
ID student not match on students.csv file
```

ถ้ารหัสนักศึกษาที่รับมามีค่าตรงกับข้อมูลในไฟล์ แล้วโปรแกรมจะตรวจสอบประวัติการจองจากไฟล์“studentbooking.csv” ถ้ามีการจองโปรแกรมจะแสดงผลวันเดือน ปีและเลข-ห้องที่จอง

ตัวอย่าง Output :

```
Current Booking:
Room: LAB103 Date: 01-12-2023
Room: IT301 Date: 01-12-2023
Room: LAB105 Date: 02-01-2023
Room: LAB103 Date: 24-11-2023
Room: LAB103 Date: 25-11-2023
```

หากไม่มีการจองไว้โปรแกรมจะแสดงผลว่า“No booking”

ตัวอย่าง Output :

```
Current Booking:
No Booking
```

3.6 ฟังก์ชันที่ 6 ตรวจสอบการจองของห้องเรียนด้วยชื่อจริง (check booking with student first name)

```

228#ตรวจสอบการจองด้วยชื่อจริง
229def func6_checkwithFname():
230    data = []
231    Nostudent = True
232    booking_file = booking_df.sort_values(by='id')
233    student_file = student_df.sort_values(by='id')
234    firstname = input("Firstname: ").lower()
235    #เก็บข้อมูลของคนที่มีประวัติการจอง
236    for _, values in booking_file.iterrows():
237        name = values['fname'].lower()
238        if firstname in name:
239            data.append(values)
240    #เก็บข้อมูลของคนที่ไม่ใช่ประวัติการจอง
241    for _, values in student_file.iterrows():
242        name = values['fname'].lower()
243        if values['id'] not in booking_file['id'].values and firstname in name:
244            data.append(values)
245    #แสดงผล
246    for item in data:
247        if item['id'] in booking_file.values:
248            print(f"{item['id']} {item['fname']} {item['lname']}")
249            print("    Current Booking: ")
250            print(f"        Room:{item['room']} Date: {item['bookingdate']}")
251            print()
252        else:
253            print(f"{item['id']} {item['fname']} {item['lname']}")
254            print("    Current Booking: ")
255            print("        No bookings")
256            print()
257    Nostudent = False
258
259    if Nostudent:
260        print("There is no student founds.")

```

ฟังก์ชันสำหรับการตรวจสอบประวัติการจองห้องด้วยชื่อจริงของนักศึกษา โดยโปรแกรมจะรับค่า ชื่อจริง หรือ ตัวอักษร ก่อนจะนำค่าที่ได้ไปเปลี่ยนให้ตัวอักษรอยู่ในรูปของตัวพิมพ์เล็กทั้งหมดเพื่อให้ความสะดวกและแม่นยำของการตรวจสอบภายในไฟล์studentbooking.csv จากตัวแปร “fname”

ตัวอย่างการรับค่า :

Option: 6
Firstname: █

หากไม่ข้อมูลที่ได้รับมานั้นไม่ตรงกับข้อมูลใดภายในไฟล์เลยโปรแกรมจะแสดงผลว่า
“There is no student founds”

ตัวอย่าง Output :

There is no student founds.

เมื่อค่าที่รับมามีข้อมูลที่ตรงกับข้อมูลภายในไฟล์โปรแกรมจะแสดงข้อมูลของนักศึกษาที่ชื่อจริงหรือตัวอักษรตรงกับค่าที่รับมาทุกคนและประวัติการจองห้องของนักศึกษาเหล่านั้น

ตัวอย่าง Output :

```
Option: 6
Firstname: Th
6687016 Thitiporn Panthanan
    Current Booking:
        Room:LAB106 Date: 01-01-2023

6687011 Chanathip Chaowaphong
    Current Booking:
        No bookings

6687015 Thitaree Kongkaew
    Current Booking:
        No bookings
```

3.7 ฟังก์ชันที่ 7 แสดงสรุปการของห้องเรียนของนักศึกษา (print booking summary)

```
263. แสดงสรุปการของห้องเรียนของนักศึกษา
263 def func7_summary(LECTURE, LAB):
264     booking_file = booking_df.sort_values(by='sorted_date', ascending=True)
265     print("Roomtype: LECTURE")
266     for room in LECTURE:
267         no_booking = True
268         print(f"    {room}")
269         for _, values in booking_file.iterrows():
270             if values['roomtype'] == 'LECTURE' and values['room'] == room:
271                 print(f"        StudentID: {values['id']} Date: {values['bookingdate']}")
272                 no_booking = False
273         if no_booking:
274             print("        No booking")
275     print()
276     print("Roomtype: LAB")
277     for room in LAB:
278         no_booking = True
279         print(f"    {room}")
280         for _, values in booking_file.iterrows():
281             if values['roomtype'] == 'LAB' and values['room'] == room:
282                 print(f"        StudentID: {values['id']} Date: {values['bookingdate']}")
283                 no_booking = False
284         if no_booking:
285             print("        No booking")
```

ฟังก์ชันจะแสดงประวัติการจองห้องทั้งหมด โดยจะแสดงผลเรียงลำดับตามเลขห้อง หากมีการจอง จะแสดงรหัสนักศึกษาของผู้จอง และวันที่มีการจองไว้ หากเลขห้องใดไม่มีการจองเลย จะแสดงผลดังนี้ “No booking”

ตัวอย่าง Output :

```
Roomtype: LECTURE
IT301
    StudentID: 6687001 Date: 01-12-2023
    StudentID: 6687004 Date: 12-10-2023
    StudentID: 6687013 Date: 16-07-2023
IT302
    StudentID: 6687009 Date: 18-10-2023
IT303
    StudentID: 6687003 Date: 12-10-2023
    StudentID: 6687012 Date: 16-07-2023
IT304
No booking

Roomtype: LAB
LAB103
    StudentID: 6687001 Date: 01-12-2023
    StudentID: 6687001 Date: 25-11-2023
    StudentID: 6687001 Date: 24-11-2023
    StudentID: 6687002 Date: 12-04-2023
LAB104
    StudentID: 6687004 Date: 01-01-2023
    StudentID: 6687009 Date: 18-10-2023
LAB105
    StudentID: 6687001 Date: 02-01-2023
LAB106
    StudentID: 6687004 Date: 22-11-2023
```

4. หน้าจอแสดงผล

```

287UI()
288select = input("Option: ")
289while True:
290    if select == '1':
291        func1_liststudentname()
292    elif select == '2':
293        func2_booking(LAB, LECTURE)
294    elif select == '3':
295        func3_checkwithroom(LECTURE, LAB)
296    elif select == '4':
297        func4_checkwithdate()
298    elif select == '5':
299        func5_checkwithID()
300    elif select == '6':
301        func6_checkwithFname()
302    elif select == '7':
303        func7_summary(LECTURE, LAB)
304    elif select == '0':
305        print("Thank you and goodbye")
306        break
307    else:
308        print("Please enter function 1-7")
309    UI()
310    select = input("Option: ")

```

โปรแกรมในส่วนนี้จะเป็นการเรียกใช้ฟังก์ชันต่างๆเริ่มจาก UI ซึ่งเป็นรายละเอียดคำสั่งต่างๆ และ จะมีการรับค่า คำสั่ง ก่อนจะเข้าสู่ฟังก์ชันตาม คำสั่ง ที่ได้รับเข้ามา

ตัวอย่าง Output :

```
=====
                        Student Room Booking System
=====
1 Print a list of students
2 Submit a booking request
3 Check the current booking via room number
4 Check the available rooms via date
5 Check booking with student ID
6 Check booking with student first name
7 Print booking summary
0 Exit
=====
Option: 
```