```
from guizero import App, Window, PushButton, Text, TextBox, Picture, ListBox, info
#
#This is needed for the sql database
import sqlite3
from sqlite3 import Error
#import sql
import os
import os.path
#import datetime
import datetime


#Define the DDL SQL to make the tables
#Tables created in database with the following details below
sql = """
CREATE TABLE "User_Table" (
    "UserID"        INTEGER NOT NULL,
    "UserName"      TEXT,
    "UserPassword"  TEXT,
    "UserFirstName" TEXT,
    "UserSurname"   TEXT,
    "UserActive"    INTEGER,
    PRIMARY KEY("UserID" AUTOINCREMENT)
);

CREATE TABLE "Flight_Table" (
    "FlightNumber"  INTEGER NOT NULL,
    "FlightIdentifier" STRING,
    "FromAIRPORT"   STRING,
    "ToAIRPORT"     STRING,
    "DayofWeek"     STRING,
    PRIMARY KEY("FlightNumber")
);

CREATE TABLE "Booking_Table" (
    "BookingID"     INTEGER NOT NULL,
    "DateBooked"    TEXT,
    "DateRequired"  TEXT,
    "TimeRequired"  TEXT,
    "NumAdults"     INTEGER,
    "NumChildren"   INTEGER,
    "NumBags"       INTEGER,
    "NumMeals"      INTEGER,
    "UserID"        INTEGER,
    "FlightNumber"  INTEGER,
    PRIMARY KEY("BookingID"),
    CONSTRAINT "UserID_fk"  FOREIGN KEY("UserID") REFERENCES
      "User_Table"("UserID"),
    CONSTRAINT "FlightNumber_fk" FOREIGN KEY("FlightNumber") REFERENCES
      "Flight_Table"("FlightNumber")
);

insert into User_Table (UserName, UserPassword, UserFirstName, UserSurname,
  UserActive) values ('joehpr', 'Stockholm', 'Joe', 'Harper', 1);
insert into Flight_Table (FlightIdentifier, FromAIRPORT, ToAIRPORT, DayofWeek)
  values ('D82858', 'London Gatwick Airport', 'Stockholm Arlanda Airport',
```

```
  'Friday');
insert into Booking_Table(DateBooked, DateRequired, TimeRequired, NumAdults,
  NumChildren, NumBags, NumMeals, UserID, FlightNumber) values ('3/12/2021',
  '5/12/2021', '17:00', 2, 2, 4, 0, 1, 1);
"""
#
#
userHasLoggedIn = False
#
database_file = 'BookingFlights.db'
#Delete the database IF IT EXISTS
#
#
if os.path.exists(database_file):
  os.remove(database_file)

#Connect to the database
conn = sqlite3.connect(database_file) #My connection is called 'conn'
#Get a cursor pointing to the database
cursor = conn.cursor()
#Create the tables
cursor.executescript(sql)
#Commit to save everything
conn.commit()
# Close the connection to the database

#Queries the database using the database parameter as the database
#to query and the query parameter as the actual query to issue
def query_database(database, query):
    conn = sqlite3.connect(database)
    cur = conn.cursor()
    cur.execute(query)
    rows = cur.fetchall()
    cur.close()
    return rows

#Executes the sql statement
def execute_sql(database, sql_statement):
    conn = sqlite3.connect(database)
    cur = conn.cursor()
    cur.execute(sql_statement)
    conn.commit()
    return cur.lastrowid

#Opens the booking window
def openWindowBook():
    if userHasLoggedIn == True:
        windowBook.show()
    else:
        windowBook.hide()
#Opens the log in window
def open_windowLogIn():
    windowL.show()
    print("Button clicked")
#Opens the sign up window
def open_windowSignUp():
```

```
    windowS.show()

def open_windowC():
    windowC.show()

#Checks the login
def check_login():
    global User
    global userHasLoggedIn

    username = input_boxU.value
    password = input_boxP.value
    print(username + password)

    query = f"select * from User_Table where UserName = '{username}' and
      UserPassword = '{password}'"
    rows = query_database(database_file, query)
    print(rows)
    userNAME = input_boxU.value
    if len(rows) == 0:          #If there are no users then there are no rows
        info("Error", "Username or password incorrect")
        userHasLoggedIn = False
    else:
        info("You can login", "Valid details")
        User = rows[0]  #rows[0] shows that 0 is the first user
                        #Save the user details when they log in
        userHasLoggedIn = True
#
#
#Sign up function
def sign_up():
    username = input_boxUs.value
    password = input_boxPa.value
    userfirstname = input_boxFn.value
    usersurname = input_boxSn.value
    checkPassword = input_boxPasswordConfirm.value

    sql_query = f"SELECT COUNT(*) FROM User_Table WHERE UserName = '{username}';"
    userExists = query_database(database_file, sql_query)
    print(userExists)
    if userExists [0][0] > 0:   #If userExists[0][0] is the count of the users in
      the database
        print("Username already taken")     #Here, if the username is already taken
           there will be a message to show that the username is already taken
    else:
        sql_query = f"insert into User_Table (UserName, UserPassword,
          UserFirstName, UserSurname, UserActive) values ('{username}',
          '{password}', '{userfirstname}', '{usersurname}', 1);"
        if password != checkPassword:
            info("Error", "Passwords don't match")
        else:
            execute_sql(database_file, sql_query)
    if len(password) < 8:
        info("Password error", "Length of password is not 8 characters")
    else:
        info("Password accepted", "Password at acceptable length")
```

```
def userNotes():
    query = f"select * from User_Notes where UserID = '{User[0]}'"
    rows = query_database(database_file, query)
    print(rows)

#Closes the window
def close_window():
    windowS.hide()
    windowL.hide()
    windowC.hide()

def makeBooking():                                  #This makes a booking for the user
  for a flight to Stockholm
    global User
    print("I am now making a booking")
    print(textFlightDetails.value)
    print(textBoxAdults.value)
    NumAdults = textBoxAdults.value
    NumChildren = textBoxChild.value
    NumBags = textBoxBags.value
    NumMeals = textBoxMeals.value
    UserID = User[0]
    DateRequired = "21/10/21"
    TimeRequired = "19:50"
    DateBooked = "07/10/2021"
    FlightNumber = 1
    sqlInsert = f"insert into Booking_Table(DateBooked, DateRequired, TimeRequired,
       NumAdults, NumChildren, NumBags, NumMeals, UserID, FlightNumber) values
      ('{DateBooked}','{DateRequired}','{TimeRequired}','{NumAdults}','{NumChildren
      }','{NumBags}', '{NumMeals}', '{UserID}', '{FlightNumber}');"
    print(sqlInsert)
    execute_sql(database_file, sqlInsert)

app = App(title="Log in or sign up for flights with Norwegian to Stockholm (Logga
  in för flyg med Norwegian till Stockholm)")       #App title

windowS = Window(app, title="Sign Up")          #Sign up window
windowS.hide()

windowL = Window(app, title="Log in")           #Log in window
windowL.hide()

windowC = Window(windowS, title="Success")
windowC.hide()

windowBook = Window(app, title="Booking page", width=1000)          #Booking page
windowBook.hide()

Gologin_button = PushButton(app, text="Go to Log In", command=open_windowLogIn)

Gosignup_button = PushButton(app, text="Go to Sign Up", command=open_windowSignUp)
closeS_button = PushButton(windowS, text="Close", command=close_window)
  #command=close_window closes the window which is currently open
closeL_button = PushButton(windowL, text="Close", command=close_window)
```

```
#
#Set up log in
#
text = Text(windowL, text="Enter username: \nSkriv in ditt användarnamn: ")
input_boxU = TextBox(windowL)
text = Text(windowL, text="Enter password: \nSkriv in lösenord: ")
input_boxP = TextBox(windowL, hide_text=True)   #hide_text=True makes the password have **** and not the real word
login_button = PushButton(windowL, text="Log In To Your Account", command=check_login) # pass username to check exists
bookflights_button = PushButton(windowL, text="Proceed to book flights", command=openWindowBook)
windowL.hide()
close_window()
#
#Set up sign up
#
text = Text(windowS, text="We are so glad you have decided to fly with Norwegian! \nVi är så glada att du har bestämt dig för att flyga med Norwegian!")
text = Text(windowS, text="Enter a username: \nAnge ett användarnamn: ")
input_boxUs = TextBox(windowS)
text = Text(windowS, text="Enter a first name: \nAnge ett förnamn:")
input_boxFn = TextBox(windowS)
text = Text(windowS, text="Enter a surname: \nAnge ett efternamn:", width=45)
input_boxSn = TextBox(windowS)
text= Text(windowS, text ="Please enter a password: \nAnge ett lösenord: ")
input_boxPa = TextBox(windowS, hide_text=True)  #hide_text=True makes the password have **** and not the real word
text = Text(windowS, text ="Please re-enter your password: \nAnge ditt lösenord igen:")
input_boxPasswordConfirm = TextBox(windowS, hide_text=True) #hide_text=True makes the password have **** and not the real word
button = PushButton(windowS, text="Go to log in", command=open_windowLogIn)
button = PushButton(windowS, text="Save details", command=sign_up)
#
#Set up the confirmation for account
#
text = Text(windowC, text="You have successfully set up an account")
close_button = PushButton(windowC, text="Close", command=close_window)
#
#Booking the flights
#
textFlightDetails = ListBox(windowBook, items=["12:25-14:45 - D84452", "19:50-21:00 - D82858", "23:00-01:45 - D82415"], width = 1000, height = 100)         #This is the listbox to show the flights
textboxNumAdults = Text(windowBook, text="Number of adults")
textBoxAdults = TextBox(windowBook, text="2")
textboxNumChild = Text(windowBook, text="Number of children")
textBoxChild = TextBox(windowBook, text="1")
textboxNumBags = Text(windowBook, text="Number of bags")
textBoxBags = TextBox(windowBook, text="3")
textboxNumMeals = Text(windowBook, text="Number of meals")
textBoxMeals = TextBox(windowBook, text="0")
BookButton = PushButton(windowBook, text="Book Flight", command=makeBooking)

app.display()
```