```
"""Module name : Project.py
   Main project framework"""


import os                  # built-in module
import time                # built-in module
import booking             # built-in module
from review_mod import *   # program module
from dtbs_mod import *     # program module

welcome()             # Welcome message

print("Please Enter your Mysql Credintials to continue:")

while True:
    try:
        con = connectdb()  # Connect to Mysql
        break
    except Exception as e:
        print("Msg:", e)      # Display encountered error
        print("Something went wrong.Please try again.\n")

time.sleep(0.5)       # Display connection status
os.system("cls")      # Clear terminal screen

while True:
    print('\n\t\tMain Menu\n\nChoose an option from the below listed : \n')
    print(' 1. Reviews of places \n 2. Your Reviews ')
    print(' 3. Hotel Bookings \n 4. Attraction of the day ')
    print(' 5. Exit ')
    q = input("\n\nYour choice (1,2,3,4) : ")

    if q == '1':
        try:
            infoplace = input("Enter the place you would like to know about : ")
            info(con, infoplace)     # Show reviews
        except Exception as e:
            print("Error code:", e)
        time.sleep(0.5)

    elif q == '2':
        while True:
            print("\n\t\tReviews\n\nEnter an option from the below listed : \n")
            print("Enter a : To write review \nEnter b : To edit a review ")
            print("Enter c : To delete a review \nEnter d : To view previous reviews ")
            print("Enter e : Exit this option")
            op4 = input("\n\nYour choice (a,b,c,d,e) : ")

            if op4.lower() == 'a':
                p = input("Enter the Place: ")
                create(con, p)

            elif op4.lower() == "b":
                edit(con)
                time.sleep(0.5)

            elif op4.lower() == "c":
                delete(con)
```

```
                time.sleep(0.5)

            elif op4.lower() == "d":
                id = int(input("Enter the reference id: "))
                show_reviews(con, id)
                time.sleep(1)

            elif op4.lower() == 'e':
                break

            else:
                print("Kindly enter a valid option")
                time.sleep(0.5)

        time.sleep(0.7)

    elif q == '3':
        print("\n\nEnter the website you would like to do your hotel booking in : ")
        print(" 1. Yatra\n 2. Easemytrip\n 3. Exit \n\n")

        while True:
            choice = input("Your response :")
            if choice == "1":
                booking.yatra()
                break
            elif choice == "2":
                booking.easemytrip()
                break
            elif choice == "3":
                break
            else:
                print("Sorry , please enter a valid option")
                time.sleep(0.5)

    elif q == '4':
        attrofday(con)

    elif q == '5':
        break

    else:
        print("Please enter a valid choice.")

con.close()

print("Thank you for using our service")
time.sleep(1.5)
```

```
"""Module Name : dtbs_mod.py
   All functions related to Mysql connections
   and database creations."""

import os                          # built-in module
import pwinput.pwinput as pw       # pip install pwinput
import mysql.connector as mysql    # pip install mysql-connector-python


def create_rev(fname, curso):  # Create Tables in Mysql from dump file
    fd = open(fname, 'r')
    sqlFile = fd.read()
    fd.close()
    sqlCommands = sqlFile.split(';')

    for command in sqlCommands:
        try:
            if command.strip() != '':
                curso.execute(command)
        except IOError as msg:
            print("Command skipped: ", msg)


def connectdb():               # Connect to Mysql
    usr = input("Enter Username: ")
    psw = pw.pwinput("Enter Password: ")
    con = mysql.connect(host='localhost', user=usr, passwd=psw)
    curs = con.cursor()
    create_dbase(curs)
    print("\nMySQL connection established ✅")
    return con


def create_dbase(curs):  # Creating database in Mysql
    sql1 = "SELECT count(*) FROM information_schema.TABLES WHERE "
    sql2 = "(TABLE_SCHEMA = 'Review') AND (TABLE_NAME = 'Reviews')"
    sql = sql1 + sql2
    curs.execute(sql)
    c = curs.fetchone()[0]
    if c == 0:
        curs.execute("Create Database Review")
        curs.execute("Use Review")
        path = os.getcwd().replace('\\', '/') + "/review_reviews.sql"
        create_rev(path, curs)
    else:
        curs.execute("Use Review")
```

```
"""Module name : review_mod.py
   All functions related with accessing and manipulating
   reviews"""

import os       # built-in module
import sys      # built-in module
import random   # built-in module


def Wiki():  # Add wikipedia to Python path
    path = os.getcwd()
    wiki_path = path + r"\Wikipediam"
    sys.path.append(wiki_path)


Wiki()
import wikipedia     # Only import after appending to path


def welcome():  # welcome message
    print("\t\t\t\tWELCOME! \n")
    wel1 = "This program has been designed to help you find information "
    wel2 = "regarding places in Kerala.We hope this will be of help to you "
    wel3 = "in finding what you are looking for!\t\t\t\t\n"
    wel = wel1 + wel2 + wel3
    print(wel)

def info(conn, infoplace):   # Show Reviews
    result = wikipedia.summary(infoplace, sentences=5)
    print(result)
    p_wiki = result.split()[0]
    status(conn, infoplace, p_wiki)
    print("\nReviews:")
    show_reviews_info(conn, infoplace, p_wiki)
    ch = input("Do you want to add review?(y/n): ")
    if ch.lower() == "y":
        create(conn, infoplace)
    print()


def attrofday(conn):      # Attraction of the day
    curs = conn.cursor()
    query = 'SELECT DISTINCT Place FROM reviews'
    curs.execute(query)
    placesl = curs.fetchall()
    places = []
    for i in placesl:
        places.append(i[0])
    attraction = random.choice(places)
    info(conn, attraction)
    curs.close()


def create(conn, place):  # for writing reviews
    f = 2
    curs = conn.cursor()
    while f != 1:
        m = input('Write your review here: ')
        ctad = input("Current restrictions(Skip if not available): ")
```

```
        name = input("Enter Your name: ")
        curs.execute("SELECT max(rev_id) FROM Reviews")
        n = curs.fetchone()[0]
        print("Reference id for editing or deleting your review is: ", n + 1)
        t = (n + 1, name, place, m, ctad)
        sql1 = 'INSERT INTO Reviews(rev_id,usr_name,Place,Reviews,trvl_avl) '
        sql2 = "values(%s,%s,%s,%s,%s)"
        sql = sql1 + sql2
        curs.execute(sql, t)
        conn.commit()
        f = int(input('Enter zero to quit. '))
        f = f + 1
    curs.close()


def show_reviews_info(conn, place, pl_wiki):     # for displaying reviews
    curs = conn.cursor()
    sql = "SELECT usr_name,REVIEWS,revdate,trvl_avl FROM Reviews WHERE Place in (%s,%s)"
    curs.execute(sql, (place, pl_wiki))
    rev = curs.fetchall()
    if rev:
        for i in rev:
            print(i[0], '  ', i[2], '\n', i[1])
    else:
        print("No Reviews were Found.")
    print()
    curs.close()


def edit(conn):     # for editing reviews previously entered
    curs = conn.cursor()
    a = int(input("Enter Review Id for the review you would like to edit: "))
    show_reviews(conn, a, 'edit')
    b = input("Enter new review: ")
    ctad = input("Current restrictions(Skip if not available): ")
    s = ("Update Reviews set Reviews=%s, trvl_avl=%s where rev_id=%s")
    curs.execute(s, (b, ctad, a))
    if curs.rowcount == 0:
        print("The reference id entered does not exist.Enter a valid id")
    else:
        print("Review has been updated")
    conn.commit()
    curs.close()


def delete(conn):        # for deleting reviews previously entered
    curs = conn.cursor()
    d = int(input("Enter the reference id for the review you would like to delete: "))
    s = ('Delete from Reviews where rev_id=%s')
    curs.execute(s, (d,))
    if curs.rowcount == 0:
        print("The reference id entered does not exist.Enter a valid id.")
    else:
        print("Review has been deleted")
    conn.commit()
    curs.close()


def status(conn, pl, plw):  # show status of a place if available
    curs = conn.cursor()
```

```
    query1 = "SELECT trvl_avl,rev_id,usr_name,revdate FROM Reviews WHERE Place=%s "
    query2 = "or Place=%s GROUP BY revdate HAVING revdate=max(revdate)"
    query = query1 + query2            # query cut short to compensate hard copy
    curs.execute(query, (pl, plw))
    dat = curs.fetchall()
    if dat:
        sta = dat[0][0]
        date = dat[0][3]
        name = dat[0][2]
        if sta != "Data not available":
            print("Current Status ( last updated on", date, 'by', name, ') :', sta)
        else:
            print("Current Status: (Please add through reviews)")
    else:
        print("Current Status: (Please add through reviews)")
    curs.close()


def show_reviews(conn, id, mode="d"):  # show reviews according to revid
    curso = conn.cursor()
    curso.execute("SELECT Reviews,Place FROM Reviews WHERE rev_id=%s", (id,))
    r = curso.fetchone()
    if r:
        place, rev = r[1], r[0]
        if mode == 'edit':  # show reviews in edit op
            print("Old review of", place, ":", rev)
        elif mode == 'd':  # show reviews according to id
            print("Your review of", place, ":", rev)
    else:
        if mode == 'd':
            print("Review id doesn't exist")
    curso.close()
```

```
"""Module name : Booking.py
   Module containing booking related functions"""

import webbrowser     # built-in module


def yatra():
    print("\n\tYATRA")
    place = input("\nEnter Destination : ")
    adults = int(input("Enter total number of Adults : "))
    children = int(input("Enter total number of children : "))
    site1 = "https://www.yatra.com/pwa/hotels/srp?roomRequests[0].id=1&roomRequests[0]."
    site2 = "noOfAdults={}&roomRequests[0].noOfChildren={}&source=BOOKING_ENGINE&"
    site3 = "pg=1&tenant=B2C&isPersnldSrp=1&city.name={}&city.code={}&state.name="
    site4 = "KER&state.code=KER&country.name=India&country.code=IND"
    # Web Address reduced to compensate with Hard copy
    site = site1 + site2 + site3 + site4
    yatra = (site.format(adults, children, place, place))
    print("You are being redirected...")
    webbrowser.open(yatra, new=1)


def easemytrip():
    print("\n\tEASEMYTRIP")
    destination = input("Enter your Destination : ")
    checkin = input("Enter Check-in date in format DD/MM/YYYY : ")
    checkout = input("Enter Check-out date in format DD/MM/YYYY : ")
    pax = input("Enter number of adults : ")
    rooms = input("Enter number of rooms required : ")
    site1 = "https://hotels.easemytrip.com/newhotel/Hotel/HotelListing?e=202193214436&"
    site2 = "city={},%20India&cin={}&cOut={}&Hotel=NA&Rooms={}&pax={}"
    site = site1 + site2         # Web address reduced to compensate with Hard copy
    easemytrip = (site.format(destination, checkin, checkout, rooms, pax))
    print("You are being redirected...")
    webbrowser.open(easemytrip, new=1)
```