*# Importing necessary libraries***import** mysql.connector  
**import** pyfiglet  
**import** requests  
**import** wikipediaapi  
**from** datetime **import** datetime  
  
  
*# Connect to the MySQL database*db = mysql.connector.connect(  
 host=**"localhost"**,  
 user=**"root"**,  
 password=**"admin"**,  
 database=**"library"**,  
)  
c = db.cursor()  
  
  
*# Function to display the return policy information***def** returnPolicy():  
 print(**"Return Policy : "**)  
 print(**"The issued book should be returned within 14 days(2 weeks)."**)  
 print(  
 **"If the user kept the issued book for more than 14 days, then the user have to pay ₹5 as fine for each extra day the user kept the issued book."** )  
 print(**"--------------------------"**)  
  
  
*# Function to calculate the length of a given integer after converting it to a string***def** length(i):  
 s = str(i)  
 length = len(s) + 2  
  
 **return** length  
  
  
*# Function to display a message for an invalid option***def** validOption():  
 print(**"Please enter a valid option!"**)  
 print(**"--------------------------"**)  
  
  
*# Function to handle program exit***def** exiting():  
 print(**"\033[3;34m--------------------------\033[0;0m"**)  
 print(**"\033[3;33mExiting the program."**)  
 print(**"Thank You!\033[0;0m"**)  
 print(**"\033[3;34m--------------------------\033[0;0m"**)  
 exit()  
  
  
*# Function to display the user menu and handle user choices***def** userMenu():  
 *# Displaying options for the user* print(**"1. Add Note"**)  
 print(**"2. Home"**)  
 print(**"3. Back"**)  
 print(**"4. Exit"**)  
 *# Taking user choice as input* userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 addNote()  
 **elif** userChoice == 2:  
 home()  
 **elif** userChoice == 3:  
 user()  
 **elif** userChoice == 4:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to display information about the library***def** aboutLibrary():  
 *# Retrieve the name of the librarian who is also an admin* c.execute(**"SELECT userName FROM users WHERE adminStatus='admin'"**)  
 userName = c.fetchall()  
  
 *# Retrieve the total number of books and users in the library* c.execute(**"SELECT \* FROM books"**)  
 totalBooks = c.fetchall()  
  
 c.execute(**"SELECT \* FROM users"**)  
 totalUsers = c.fetchall()  
 db.commit()  
  
 print(**"--------------------------"**)  
 print(**"About Library"**)  
 print(**"--------------------------"**)  
 *# Display library information* print(**"Year of Library's Establishment : "**, 2023)  
 print(**"Name of the Librarian : "**, userName[0][0])  
 print(**"Total Number of Books Available in the Library : "**, len(totalBooks))  
 print(**"Total Number of Users Enrolled in the Library : "**, len(totalUsers))  
 print(**"--------------------------"**)  
 userMenu()  
  
  
*# Function to display the list of books in the library***def** displayBooks():  
 print(**"--------------------------"**)  
 print(**"Display Books"**)  
 print(**"--------------------------"**)  
 *# Retrieve all books from the database* c.execute(**"SELECT \* FROM books ORDER BY bookId"**)  
 result = c.fetchall()  
 db.commit()  
  
 *# Display books if available, otherwise notify the user* **if** result:  
 print(**"Books available in the Digital Library are :"**)  
 print(**"--------------------------"**)  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. Book ID : {**row[0]**}"**)  
 print(**" "** \* r + **f"Book Name : {**row[1]**}"**)  
 print(**" "** \* r + **f"Publication Year : {**row[2]**}"**)  
 print(**" "** \* r + **f"Author Name : {**row[7]**}"**)  
 print(**" "** \* r + **f"Issue Status : {**row[8]**}"**)  
 print(**"--------------------------"**)  
 userMenu()  
 **else**:  
 *# Notify the user if no books are found* print(**"No books found."**)  
 print(**"--------------------------"**)  
 userMenu()  
  
  
*# Search books menu options***def** searchBooksMenu():  
 print(**"1. Add Note"**)  
 print(**"2. Home"**)  
 print(**"3. Back"**)  
 print(**"4. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
  
 *# User choices handling* **if** userChoice == 1:  
 addNote()  
 **elif** userChoice == 2:  
 home()  
 **elif** userChoice == 3:  
 searchBooks()  
 **elif** userChoice == 4:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to search books by Book ID***def** searchBooksbyId():  
 print(**"--------------------------"**)  
 print(**"Search Books by Book ID"**)  
 print(**"--------------------------"**)  
 *# Get user input for Book ID* bookId = int(input(**"Enter the Book ID to search the Book : "**))  
 print(**"--------------------------"**)  
  
 *# Execute SQL query to retrieve book information by Book ID* c.execute(**"SELECT \* FROM books WHERE bookId=%s"**, (bookId,))  
 result = c.fetchall()  
 db.commit()  
  
 *# Display search results if books are found, otherwise notify the user* **if** result:  
 print(**f'Book available in the Digital Library with the Book ID "{**bookId**}" is :'**)  
 print(**"--------------------------"**)  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. Book ID : {**row[0]**}"**)  
 print(**" "** \* r + **f"Book Name : {**row[1]**}"**)  
 print(**" "** \* r + **f"Publication Year : {**row[2]**}"**)  
 print(**" "** \* r + **f"Author Name : {**row[7]**}"**)  
 print(**" "** \* r + **f"Issue Status : {**row[8]**}"**)  
 print(**"--------------------------"**)  
 searchBooksMenu()  
 **else**:  
 print(**f'No book found with the book id "{**bookId**}".'**)  
 print(**"--------------------------"**)  
 searchBooksMenu()  
  
  
*# Function to search books by keyword***def** searchBooksbyKeyword():  
 print(**"--------------------------"**)  
 print(**"Search Books by Keyword"**)  
 print(**"--------------------------"**)  
 *# Get user input for keyword* keyword = input(**"Enter a Keyword to search Books : "**)  
 print(**"--------------------------"**)  
  
 *# Execute SQL query to retrieve books by keyword* c.execute(  
 **"SELECT \* FROM books WHERE bookName LIKE '%{}%' ORDER BY bookId"**.format(keyword)  
 )  
 result = c.fetchall()  
 db.commit()  
  
 *# Display search results if books are found, otherwise notify the user* **if** result:  
 print(  
 **f'Books available in the Digital Library with the Keyword "{**keyword**}" are :'** )  
 print(**"--------------------------"**)  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. Book ID : {**row[0]**}"**)  
 print(**" "** \* r + **f"Book Name : {**row[1]**}"**)  
 print(**" "** \* r + **f"Publication Year : {**row[2]**}"**)  
 print(**" "** \* r + **f"Author Name : {**row[7]**}"**)  
 print(**" "** \* r + **f"Issue Status : {**row[8]**}"**)  
 print(**"--------------------------"**)  
 searchBooksMenu()  
 **else**:  
 print(**f'No books found with the keyword "{**keyword**}".'**)  
 print(**"--------------------------"**)  
 searchBooksMenu()  
  
  
*# Function to display search options for books***def** searchBooks():  
 print(**"--------------------------"**)  
 print(**"Search Books"**)  
 print(**"--------------------------"**)  
 print(**"1. Search by Book ID"**)  
 print(**"2. Search by Keyword"**)  
 print(**"3. Home"**)  
 print(**"4. Back"**)  
 print(**"5. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 searchBooksbyId()  
 **elif** userChoice == 2:  
 searchBooksbyKeyword()  
 **elif** userChoice == 3:  
 home()  
 **elif** userChoice == 4:  
 user()  
 **elif** userChoice == 5:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to display the add book menu and handle user choices***def** addBookMenu():  
 *# Add book menu options* print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 modifyBook()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to add a new book to the library***def** addBook():  
 print(**"--------------------------"**)  
 print(**"Add Book"**)  
 print(**"--------------------------"**)  
 *# Get user input for book details* bookId = int(input(**"Enter the Book ID : "**))  
 bookName = input(**"Enter the Book Name : "**)  
 publicationYear = int(input(**"Enter the Book Publication Year : "**))  
 author = input(**"Enter the Book Author Name : "**)  
 print(**"--------------------------"**)  
  
 c.execute(**"SELECT bookId FROM books"**)  
 result = c.fetchall()  
 db.commit()  
  
 **if** (bookId,) **in** result:  
 print(  
 **f'The book of book id "{**bookId**}" is already available in the digital library.'** )  
 print(**"--------------------------"**)  
 addBookMenu()  
 **else**:  
 *# Execute SQL query to insert the new book into the database* c.execute(  
 **"INSERT INTO books (bookId, bookName, publicationYear, author) VALUES (%s, %s, %s, %s)"**,  
 (bookId, bookName, publicationYear, author),  
 )  
 db.commit()  
  
 *# Notify the user that the book has been added successfully* print(**"Book added Successfully!"**)  
 print(**"--------------------------"**)  
 addBookMenu()  
  
  
*# Function to display the delete book menu and handle user choices***def** deleteBookMenu():  
 *# Delete book menu options* print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 admin()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to delete a book from the library***def** deleteBook():  
 print(**"--------------------------"**)  
 print(**"Delete Book"**)  
 print(**"--------------------------"**)  
 *# Get user input for the book ID to be deleted* bookId = int(input(**"Enter the Book ID : "**))  
 choice = input(**"Are you sure to delete the Book? (Yes/No) : "**)  
 print(**"--------------------------"**)  
  
 c.execute(**"SELECT bookId FROM books"**)  
 result = c.fetchall()  
 db.commit()  
  
 **if** choice.lower() **in** [**"yes"**, **"y"**]:  
 **if** (bookId,) **in** result:  
 *# Execute SQL query to delete the book from the database* c.execute(**"DELETE FROM books WHERE bookId=%s"**, (bookId,))  
 db.commit()  
  
 *# Notify the user that the book has been deleted successfully* print(**"Book deleted Successfully!"**)  
 print(**"--------------------------"**)  
 deleteBookMenu()  
 **else**:  
 print(  
 **f'The book of book id "{**bookId**}" does not available in the digital library.'** )  
 print(**"--------------------------"**)  
 deleteBookMenu()  
 **elif** choice.lower() **in** [**"no"**, **"n"**]:  
 print(**"--------------------------"**)  
 print(**"Book Not Deleted!"**)  
 print(**"--------------------------"**)  
 deleteBookMenu()  
 **else**:  
 validOption()  
  
  
*# Update book menu options***def** updateBookMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 updateUser()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
**def** notBook(bookId):  
 print(**f'The book of book id "{**bookId**}" does not available in the digital library.'**)  
 print(**"--------------------------"**)  
 updateBookMenu()  
  
  
*# Function to update book details***def** updateBook():  
 print(**"--------------------------"**)  
 print(**"Update Book Details"**)  
 print(**"--------------------------"**)  
 print(**"1. Update the Book ID"**)  
 print(**"2. Update the Book Name"**)  
 print(**"3. Update the Book Publication Year"**)  
 print(**"4. Update the Book Author Name"**)  
 print(**"5. Home"**)  
 print(**"6. Back"**)  
 print(**"7. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 c.execute(**"SELECT bookId FROM books"**)  
 result = c.fetchall()  
 db.commit()  
  
 *# User choices handling* **if** userChoice == 1:  
 currentBookId = int(input(**"Enter the Current Book ID : "**))  
 newBookId = int(input(**"Enter the New Book ID : "**))  
  
 **if** (currentBookId,) **in** result:  
 *# Execute SQL query to update the Book ID* c.execute(  
 **"UPDATE books SET bookId=%s WHERE bookId=%s"**, (newBookId, currentBookId)  
 )  
 db.commit()  
  
 print(**"Book ID changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateBookMenu()  
 **else**:  
 notBook(currentBookId)  
  
 **elif** userChoice == 2:  
 bookId = int(input(**"Enter the Book ID : "**))  
 newBookName = input(**"Enter the New Book Name : "**)  
  
 **if** (bookId,) **in** result:  
 *# Execute SQL query to update the Book Name* c.execute(  
 **"UPDATE books SET bookName=%s WHERE bookId=%s"**, (newBookName, bookId)  
 )  
 db.commit()  
  
 print(**"Book Name changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateBookMenu()  
 **else**:  
 notBook(bookId)  
  
 **elif** userChoice == 3:  
 bookId = int(input(**"Enter the Current Book ID : "**))  
 newPublicationYear = input(**"Enter the New Publication Year : "**)  
  
 **if** (bookId,) **in** result:  
 *# Execute SQL query to update the Publication Year* c.execute(  
 **"UPDATE books SET publicationYear=%s WHERE bookId=%s"**,  
 (newPublicationYear, bookId),  
 )  
 db.commit()  
  
 print(**"Book Publication Year changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateBookMenu()  
  
 **elif** userChoice == 4:  
 bookId = int(input(**"Enter the Current Book ID : "**))  
 newAuthor = input(**"Enter the New Author Name : "**)  
  
 **if** (bookId,) **in** result:  
 *# Execute SQL query to update the Author Name* c.execute(  
 **"UPDATE books SET author=%s WHERE bookId=%s"**,  
 (newAuthor, bookId),  
 )  
 db.commit()  
  
 print(**"Book Author Name changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateBookMenu()  
 **else**:  
 notBook(bookId)  
  
 **elif** userChoice == 5:  
 home()  
 **elif** userChoice == 6:  
 modifyBook()  
 **elif** userChoice == 7:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to display the issue book menu and handle user choices***def** issueBookMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 admin()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to issue a book***def** issueBook():  
 print(**"--------------------------"**)  
 print(**"Issue Book"**)  
 print(**"--------------------------"**)  
 bookId = int(input(**"Enter the Book ID to be Issued: "**))  
 userId = int(input(**"Enter the User ID to whom Book will be Issued: "**))  
  
 *# Execute SQL query to check the issue status of the book* c.execute(**"SELECT userId FROM users"**)  
 result1 = c.fetchall()  
 c.execute(**"SELECT bookId FROM books"**)  
 result2 = c.fetchall()  
 c.execute(**"SELECT issueStatus FROM books WHERE bookId=%s"**, (bookId,))  
 result3 = c.fetchall()  
 db.commit()  
  
 **if** (userId,) **in** result1:  
 **if** (bookId,) **in** result2:  
 *# Check if the book is not already issued* **if** result3[0][0] == **"not issued"**:  
 *# Execute SQL queries to update book details and mark it as issued* c.execute(  
 **"UPDATE books SET issueDate = CURRENT\_DATE WHERE bookId = %s"**,  
 (bookId,),  
 )  
 c.execute(  
 **"UPDATE books SET issueTime = CURRENT\_TIME WHERE bookId = %s"**,  
 (bookId,),  
 )  
 c.execute(  
 **"UPDATE books SET issueStatus = 'issued' WHERE bookId = %s"**,  
 (bookId,),  
 )  
 c.execute(  
 **"UPDATE books SET returnDate = NULL WHERE bookId = %s"**, (bookId,)  
 )  
 c.execute(  
 **"UPDATE books SET returnTime = NULL WHERE bookId = %s"**, (bookId,)  
 )  
 c.execute(  
 **"UPDATE books SET issuedUserId = %s WHERE bookId = %s"**,  
 (userId, bookId),  
 )  
 db.commit()  
 c.execute(  
 **"select issuedUserId,bookName,issueDate,issueTime from books where bookId=%s"**,  
 (bookId,),  
 )  
 result = c.fetchall()  
 c.execute(  
 **"INSERT INTO issuedBooksDetails (userId, bookId,bookName,issueDate,issueTime) VALUES (%s, %s, %s, %s, %s)"**,  
 (result[0][0], bookId, result[0][1], result[0][2], result[0][3]),  
 )  
 db.commit()  
  
 print(**"--------------------------"**)  
 print(  
 **f'Book of Book Id "{**bookId**}" is issued successfully to the User of User Id "{**userId**}".'** )  
 print(**"--------------------------"**)  
 returnPolicy()  
 issueBookMenu()  
 **else**:  
 *# Notify the user that the book is already issued* print(  
 **f'The book of book id "{**bookId**}" is already issued by another user.'** )  
 print(**"--------------------------"**)  
 issueBookMenu()  
 **else**:  
 print(  
 **f"Book with book id {**bookId**} does not available in the digital library."** )  
 print(**"--------------------------"**)  
 issueBookMenu()  
 **else**:  
 print(**f"User with user id {**userId**} does not exists in the digital library."**)  
 print(**"--------------------------"**)  
 issueBookMenu()  
  
  
*# Function to display the return book menu and handle user choices***def** returnBookMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 admin()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to return a book***def** returnBook():  
 print(**"--------------------------"**)  
 print(**"Return Book"**)  
 print(**"--------------------------"**)  
 bookId = int(input(**"Enter the Book ID to be Returned: "**))  
  
 *# Execute SQL query to check the issue status of the book* c.execute(**"SELECT bookId FROM books"**)  
 result1 = c.fetchall()  
 c.execute(**"SELECT issueStatus FROM books WHERE bookId=%s"**, (bookId,))  
 result2 = c.fetchall()  
  
 db.commit()  
  
 **if** (bookId,) **in** result1:  
 *# Check if the book is issued* **if** result2[0][0] == **"issued"**:  
 *# Execute SQL queries to update book details and mark it as returned* c.execute(  
 **"UPDATE books SET returnDate = CURRENT\_DATE WHERE bookId = %s"**,  
 (bookId,),  
 )  
 c.execute(  
 **"UPDATE books SET returnTime = CURRENT\_TIME WHERE bookId = %s"**,  
 (bookId,),  
 )  
 c.execute(  
 **"UPDATE books SET issueStatus = 'not issued' WHERE bookId = %s"**,  
 (bookId,),  
 )  
 db.commit()  
 c.execute(  
 **"select issuedUserId,returnDate,returnTime from books where bookId=%s"**,  
 (bookId,),  
 )  
 result = c.fetchall()  
 c.execute(  
 **"UPDATE issuedBooksDetails SET returnDate = %s, returnTime = %s WHERE userId = %s AND bookId = %s"**,  
 (result[0][1], result[0][2], result[0][0], bookId),  
 )  
  
 db.commit()  
 c.execute(  
 **"UPDATE books SET issuedUserId = NULL WHERE bookId = %s"**, (bookId,)  
 )  
 db.commit()  
  
 print(**f'The book of book id "{**bookId**}" is returned successfully.'**)  
  
 c.execute(**"select issueDate from books WHERE bookId = %s"**, (bookId,))  
 issueDate = c.fetchall()  
 c.execute(**"select returnDate from books WHERE bookId = %s"**, (bookId,))  
 returnDate = c.fetchall()  
 db.commit()  
  
 c.execute(**"UPDATE books SET issueDate = NULL WHERE bookId = %s"**, (bookId,))  
 c.execute(**"UPDATE books SET issueTime = NULL WHERE bookId = %s"**, (bookId,))  
 c.execute(**"UPDATE books SET returnDate = NULL WHERE bookId = %s"**, (bookId,))  
 c.execute(**"UPDATE books SET returnTime = NULL WHERE bookId = %s"**, (bookId,))  
 db.commit()  
  
 d1 = datetime.strptime(**f"{**issueDate[0][0]**}"**, **"%Y-%m-%d"**)  
 d2 = datetime.strptime(**f"{**returnDate[0][0]**}"**, **"%Y-%m-%d"**)  
 dateDifference = d1 - d2  
  
 **if** dateDifference.days > 14:  
 extraDays = dateDifference.days - 14  
 fine = extraDays \* 5  
 print(**"Fine(in Rs.) : "**, fine)  
 c.execute(  
 **"update issuedBooksDetails set fineInRs=%s where userId=%s and bookId=%s"**,  
 (fine, result[0][0], bookId),  
 )  
 db.commit()  
 **else**:  
 fine = 0 \* 5  
 print(**"Fine(in Rs.) : "**, fine)  
 c.execute(  
 **"update issuedBooksDetails set fineInRs=%s where userId=%s and bookId=%s"**,  
 (fine, result[0][0], bookId),  
 )  
 db.commit()  
  
 print(**"--------------------------"**)  
 returnBookMenu()  
 **else**:  
 *# Notify the user that the book is not issued* print(**f'The book of book id "{**bookId**}" is not issued by any user.'**)  
 print(**"--------------------------"**)  
 returnBookMenu()  
 **else**:  
 print(**f"Book with book id {**bookId**} does not available in the digital library."**)  
 print(**"--------------------------"**)  
 returnBookMenu()  
  
  
*# Function to display the add user menu and handle user choices***def** addUserMenu():  
 *# Add user menu options* print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 modifyUser()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to add a new user***def** addUser():  
 print(**"--------------------------"**)  
 print(**"Add User"**)  
 print(**"--------------------------"**)  
 *# Get user input for user details* userId = int(input(**"Enter the User ID : "**))  
 userName = input(**"Enter the User Name : "**)  
 userPhoneNumber = input(**"Enter the User Phone Number : "**)  
 userEmailId = input(**"Enter the User Email ID : "**)  
 password = input(**"Enter the User Password : "**)  
 print(**"--------------------------"**)  
  
 c.execute(**"SELECT userId FROM users"**)  
 result = c.fetchall()  
 db.commit()  
  
 **if** (userId,) **in** result:  
 print(  
 **f'The user of user number "{**userId**}" is already enrolled in the digital library.'** )  
 print(**"--------------------------"**)  
 addUserMenu()  
 **else**:  
 *# Execute SQL query to insert the new user into the database* c.execute(  
 **"INSERT INTO users (userId, userName, phoneNumber, emailId, password) VALUES (%s, %s, %s, %s, %s)"**,  
 (userId, userName, userPhoneNumber, userEmailId, password),  
 )  
 db.commit()  
  
 *# Notify the user that the user has been added successfully* print(**"--------------------------"**)  
 print(**"User added successfully!"**)  
 print(**"--------------------------"**)  
 addUserMenu()  
  
  
*# Function to display the delete user menu and handle user choices***def** deleteUserMenu():  
 *# Delete user menu options* print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 modifyUser()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to delete a user***def** deleteUser():  
 print(**"--------------------------"**)  
 print(**"Delete User"**)  
 print(**"--------------------------"**)  
 *# Get user input for the user ID to be deleted* userId = int(input(**"Enter the User ID : "**))  
 choice = input(**"Are you sure to delete the User? (Yes/No) : "**)  
  
 c.execute(**"SELECT userId FROM users"**)  
 result = c.fetchall()  
 db.commit()  
  
 **if** choice.lower() **in** [**"yes"**, **"y"**]:  
 **if** (userId,) **in** result:  
 c.execute(**"DELETE FROM users WHERE userId=%s"**, (userId,))  
 db.commit()  
  
 *# Notify the user that the user has been deleted successfully* print(**"User deleted successfully!"**)  
 print(**"--------------------------"**)  
 deleteUserMenu()  
 **else**:  
 print(  
 **f'The user of user id "{**userId**}" does not enrolled in the digital library.'** )  
 print(**"--------------------------"**)  
 deleteUserMenu()  
 **elif** choice.lower() **in** [**"no"**, **"n"**]:  
 print(**"--------------------------"**)  
 print(**"User Not Deleted!"**)  
 print(**"--------------------------"**)  
 deleteUserMenu()  
 **else**:  
 validOption()  
  
  
*# Function to display the update user menu and handle user choices***def** updateUserMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 updateUser()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
**def** notUser(userId):  
 print(**f'The user of user id "{**userId**}" does not enrolled in the digital library.'**)  
 print(**"--------------------------"**)  
 updateBookMenu()  
  
  
*# Function to update user details***def** updateUser():  
 print(**"--------------------------"**)  
 print(**"Update User Details"**)  
 print(**"--------------------------"**)  
 *# Display user update options* print(**"1. Update the User ID"**)  
 print(**"2. Update the User Name"**)  
 print(**"3. Update the User Phone Number"**)  
 print(**"4. Update the User Email ID"**)  
 print(**"5. Update the User Password"**)  
 print(**"6. Home"**)  
 print(**"7. Back"**)  
 print(**"8. Exit"**)  
 *# Get user choice* userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 c.execute(**"SELECT userId FROM users"**)  
 result = c.fetchall()  
 db.commit()  
  
 **if** userChoice == 1:  
 *# Update user ID* currentUserId = int(input(**"Enter the Current User ID : "**))  
 newUserId = int(input(**"Enter the New User ID : "**))  
  
 **if** (currentUserId,) **in** result:  
 c.execute(  
 **"update users set userId=%s where userId=%s"**, (newUserId, currentUserId)  
 )  
 db.commit()  
  
 print(**"User ID changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateUserMenu()  
 **else**:  
 notUser(currentUserId)  
  
 **elif** userChoice == 2:  
 *# Update user name* userId = int(input(**"Enter the User ID : "**))  
 newUserName = input(**"Enter the New User Name : "**)  
  
 **if** (userId,) **in** result:  
 c.execute(  
 **"update users set userName=%s where userId=%s"**, (newUserName, userId)  
 )  
 db.commit()  
  
 print(**"User Name changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateUserMenu()  
 **else**:  
 notUser(userId)  
  
 **elif** userChoice == 3:  
 *# Update user phone number* userId = int(input(**"Enter the Current User ID : "**))  
 newPhoneNumber = input(**"Enter the New Phone Number : "**)  
  
 **if** (userId,) **in** result:  
 c.execute(  
 **"update users set phoneNumber=%s where userId=%s"**,  
 (newPhoneNumber, userId),  
 )  
 db.commit()  
  
 print(**"User Phone Number changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateUserMenu()  
 **else**:  
 notUser(userId)  
  
 **elif** userChoice == 4:  
 *# Update user email ID* userId = int(input(**"Enter the Current User ID : "**))  
 newEmailId = input(**"Enter the New Email ID : "**)  
  
 **if** (userId,) **in** result:  
 c.execute(  
 **"update users set emailId=%s where userId=%s"**, (newEmailId, userId)  
 )  
 db.commit()  
  
 print(**"User Email ID changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateUserMenu()  
 **else**:  
 notUser(userId)  
  
 **elif** userChoice == 5:  
 *# Update user password* userId = int(input(**"Enter the Current User ID : "**))  
 newPassword = input(**"Enter the New Password : "**)  
 **if** (userId,) **in** result:  
 c.execute(  
 **"update users set password=%s where userId=%s"**, (newPassword, userId)  
 )  
 db.commit()  
  
 print(**"User Password changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateUserMenu()  
 **else**:  
 notUser(userId)  
  
 **elif** userChoice == 6:  
 *# Return to home* home()  
 **elif** userChoice == 7:  
 *# Go back to the previous menu* modifyUser()  
 **elif** userChoice == 8:  
 *# Exit the program* exiting()  
 **else**:  
 validOption()  
  
  
*# Function to modify user***def** modifyUser():  
 print(**"--------------------------"**)  
 print(**"Modify User"**)  
 print(**"--------------------------"**)  
 *# Display user modification options* print(**"1. Add User"**)  
 print(**"2. Delete User"**)  
 print(**"3. Update User Details"**)  
 print(**"4. Home"**)  
 print(**"5. Back"**)  
 print(**"6. Exit"**)  
 *# Get user choice* userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 *# Add a new user* addUser()  
 **elif** userChoice == 2:  
 *# Delete a user* deleteUser()  
 **elif** userChoice == 3:  
 *# Update user details* updateUser()  
 **elif** userChoice == 4:  
 *# Return to home* home()  
 **elif** userChoice == 5:  
 *# Return to the previous menu* admin()  
 **elif** userChoice == 6:  
 *# Exit the program* exiting()  
 **else**:  
 validOption()  
  
  
*# Display users menu options***def** displayUsersMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 admin()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to display all users***def** displayUsers():  
 print(**"--------------------------"**)  
 print(**"Display Users"**)  
 print(**"--------------------------"**)  
 *# Fetch all users from the database* c.execute(**"SELECT \* FROM users ORDER BY userId"**)  
 result = c.fetchall()  
 db.commit()  
  
 **if** result:  
 *# Display user information* print(**"Users enrolled in the Digital Library are :"**)  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. User ID : {**row[0]**}"**)  
 print(**" "** \* r + **f"User Name : {**row[1]**}"**)  
 print(**" "** \* r + **f"Phone Number : {**row[2]**}"**)  
 print(**" "** \* r + **f"Email ID : {**row[3]**}"**)  
 print(**" "** \* r + **f"Admin Status : {**row[5]**}"**)  
 print(**"--------------------------"**)  
 displayUsersMenu()  
  
 **else**:  
 print(**"No users found."**)  
 print(**"--------------------------"**)  
 displayUsersMenu()  
  
  
*# Search user menu options***def** searchUsersMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
  
 *# User choices handling* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 searchUsers()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to search users by ID***def** searchUsersbyId():  
 print(**"--------------------------"**)  
 print(**"Search Users by User ID"**)  
 print(**"--------------------------"**)  
 *# Get user ID to search* userId = int(input(**"Enter the User ID to search the User : "**))  
  
 *# Search for the user in the database* c.execute(**"SELECT \* FROM users WHERE userId=%s"**, (userId,))  
 result = c.fetchall()  
 db.commit()  
  
 **if** result:  
 *# Display user information if found* print(**f'User enrolled in the Digital Library with the User ID "{**userId**}" is :'**)  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. User ID : {**row[0]**}"**)  
 print(**" "** \* r + **f"User Name : {**row[1]**}"**)  
 print(**" "** \* r + **f"Phone Number : {**row[2]**}"**)  
 print(**" "** \* r + **f"Email ID : {**row[3]**}"**)  
 print(**" "** \* r + **f"Admin Status : {**row[5]**}"**)  
 print(**"--------------------------"**)  
 searchUsersMenu()  
  
 **else**:  
 *# Handle case when no user is found* print(**f'No user found with the user id "{**userId**}".'**)  
 print(**"--------------------------"**)  
 searchUsersMenu()  
  
  
*# Function to search users by keyword***def** searchUsersbyKeyword():  
 print(**"--------------------------"**)  
 print(**"Search Users by Keyword"**)  
 print(**"--------------------------"**)  
 *# Get keyword input from the user* keyword = input(**"Enter a Keyword to search Users : "**)  
  
 *# Search for users with the given keyword in their names* c.execute(  
 **"SELECT \* FROM users WHERE userName LIKE '%{}%' ORDER BY userId"**.format(keyword)  
 )  
 result = c.fetchall()  
 db.commit()  
  
 **if** result:  
 *# Display user information if users are found* print(  
 **f'Users enrolled in the Digital Library with the Keyword "{**keyword**}" are :'** )  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. User ID : {**row[0]**}"**)  
 print(**" "** \* r + **f"User Name : {**row[1]**}"**)  
 print(**" "** \* r + **f"Phone Number : {**row[2]**}"**)  
 print(**" "** \* r + **f"Email ID : {**row[3]**}"**)  
 print(**" "** \* r + **f"Admin Status : {**row[5]**}"**)  
 print(**"--------------------------"**)  
 searchUsersMenu()  
  
 **else**:  
 *# Handle case when no user is found* print(**f'No users found with the keyword "{**keyword**}".'**)  
 print(**"--------------------------"**)  
 searchUsersMenu()  
  
  
*# Function to search users***def** searchUsers():  
 print(**"--------------------------"**)  
 print(**"Search Users"**)  
 print(**"--------------------------"**)  
 *# User search menu* print(**"1. Search by User ID"**)  
 print(**"2. Search by Keyword"**)  
 print(**"3. Home"**)  
 print(**"4. Back"**)  
 print(**"5. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 searchUsersbyId()  
 **elif** userChoice == 2:  
 searchUsersbyKeyword()  
 **elif** userChoice == 3:  
 home()  
 **elif** userChoice == 4:  
 admin()  
 **elif** userChoice == 5:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to modify books***def** modifyBook():  
 print(**"--------------------------"**)  
 print(**"Modify Book"**)  
 print(**"--------------------------"**)  
 *# Book modification menu* print(**"1. Add Book"**)  
 print(**"2. Delete Book"**)  
 print(**"3. Update Book Details"**)  
 print(**"4. Home"**)  
 print(**"5. Back"**)  
 print(**"6. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# User choices handling* **if** userChoice == 1:  
 addBook()  
 **elif** userChoice == 2:  
 deleteBook()  
 **elif** userChoice == 3:  
 updateBook()  
 **elif** userChoice == 4:  
 home()  
 **elif** userChoice == 5:  
 admin()  
 **elif** userChoice == 6:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to manage notes***def** notes():  
 print(**"--------------------------"**)  
 print(**"Notes"**)  
 print(**"--------------------------"**)  
 *# Display menu options* print(**"1. Modify Note"**)  
 print(**"2. Display Notes"**)  
 print(**"3. Search Notes"**)  
 print(**"4. Home"**)  
 print(**"5. Back"**)  
 print(**"6. Exit"**)  
 *# Get user choice* userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 modifyNote()  
 **elif** userChoice == 2:  
 displayNotes()  
 **elif** userChoice == 3:  
 searchNotes()  
 **elif** userChoice == 4:  
 home()  
 **elif** userChoice == 5:  
 user()  
 **elif** userChoice == 6:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to display the add note menu and handle user choices***def** addNoteMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 *# Get user choice* userChoice = int(input(**"Enter your Choice to Continue : "**))  
  
 *# Handle user choices* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 modifyNote()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to add note***def** addNote():  
 print(**"--------------------------"**)  
 print(**"Add Note"**)  
 print(**"--------------------------"**)  
 *# Get note details from the user* noteNumber = int(input(**"Enter the Note Number : "**))  
 noteTitle = input(**"Enter the Note Title : "**)  
 noteDescription = input(**"Enter the Note Description : "**)  
 print(**"--------------------------"**)  
  
 c.execute(**"SELECT noteNumber FROM notes where userId=%s"**, (USERID,))  
 result = c.fetchall()  
 db.commit()  
  
 **if** (noteNumber,) **in** result:  
 print(  
 **f'The note of note number "{**noteNumber**}" is already exists in the digital library.'** )  
 print(**"--------------------------"**)  
 addNoteMenu()  
  
 **else**:  
 *# Execute SQL query to insert the note into the database* c.execute(  
 **"INSERT INTO notes (userId, noteNumber, noteTitle, noteDescription, updateDate, updateTime) VALUES (%s, %s, %s, %s, CURRENT\_DATE, CURRENT\_TIME)"**,  
 (USERID, noteNumber, noteTitle, noteDescription),  
 )  
 db.commit()  
  
 print(**f'The note of note number "{**noteNumber**}" is added successfully.'**)  
 print(**"--------------------------"**)  
 addNoteMenu()  
  
  
*# Function to display the delete note menu and handle user choices***def** deleteNoteMenu():  
 *# Display menu options after deleting the note* print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 *# Get user choice* userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 modifyNote()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to delete a note***def** deleteNote():  
 print(**"--------------------------"**)  
 print(**"Delete Note"**)  
 print(**"--------------------------"**)  
 *# Get note number to be deleted from the user* noteNumber = int(input(**"Enter the Note Number to Delete the Note : "**))  
 choice = input(**"Are you sure to delete the Note? (Yes/No) : "**)  
 print(**"--------------------------"**)  
  
 c.execute(**"SELECT noteNumber FROM notes where userId=%s"**, (USERID,))  
 result = c.fetchall()  
 db.commit()  
  
 **if** choice.lower() **in** [**"yes"**, **"y"**]:  
 **if** (noteNumber,) **in** result:  
 *# Execute SQL query to delete the note from the database* c.execute(  
 **"delete FROM notes WHERE userId=%s and noteNumber=%s"**,  
 (USERID, noteNumber),  
 )  
 db.commit()  
  
 print(**f'The note of note number "{**noteNumber**}" is deleted successfully.'**)  
 print(**"--------------------------"**)  
 deleteNoteMenu()  
  
 **else**:  
 print(  
 **f'The note of note number "{**noteNumber**}" does not exists in the digital library.'** )  
 print(**"--------------------------"**)  
 deleteNoteMenu()  
 **elif** choice.lower() **in** [**"no"**, **"n"**]:  
 print(**"--------------------------"**)  
 print(**"Note Not Deleted!"**)  
 print(**"--------------------------"**)  
 deleteNoteMenu()  
 **else**:  
 validOption()  
  
  
*# Function to display the update notes menu and handle user choices***def** updateNotesMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 *# Get user choice* userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 updateNotes()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
**def** notNote(noteNumber):  
 print(  
 **f'The note of note number "{**noteNumber**}" does not exists in the digital library.'** )  
 print(**"--------------------------"**)  
 updateNotesMenu()  
  
  
*# Function to update a note***def** updateNotes():  
 print(**"--------------------------"**)  
 print(**"Update Notes"**)  
 print(**"--------------------------"**)  
 *# Display update options* print(**"1. Update the Note Number"**)  
 print(**"2. Update the Note Title"**)  
 print(**"3. Update the Note Description"**)  
 print(**"4. Home"**)  
 print(**"5. Back"**)  
 print(**"6. Exit"**)  
 *# Get user choice* userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 c.execute(**"SELECT noteNumber FROM notes where userId=%s"**, (USERID,))  
 result = c.fetchall()  
 db.commit()  
  
 *# Handle user choices* **if** userChoice == 1:  
 *# Update Note Number* currentNoteNumber = int(input(**"Enter the Current Note Number : "**))  
 newNoteNumber = int(input(**"Enter the New Note Number : "**))  
  
 **if** (currentNoteNumber,) **in** result:  
 *# Update date and time* c.execute(  
 **"update notes set updateDate=CURRENT\_DATE where userId=%s and noteNumber=%s"**,  
 (USERID, currentNoteNumber),  
 )  
 c.execute(  
 **"update notes set updateTime=CURRENT\_TIME where userId=%s and noteNumber=%s"**,  
 (USERID, currentNoteNumber),  
 )  
 *# Update Note Number* c.execute(  
 **"update notes set noteNumber=%s where userId=%s and noteNumber=%s"**,  
 (newNoteNumber, USERID, currentNoteNumber),  
 )  
 db.commit()  
  
 print(**"Note Number changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateNotesMenu()  
 **else**:  
 notNote(currentNoteNumber)  
  
 **elif** userChoice == 2:  
 *# Update Note Title* noteNumber = int(input(**"Enter the Current Note Number : "**))  
 newTitle = input(**"Enter the New Note Title : "**)  
  
 **if** (noteNumber,) **in** result:  
 *# Update date and time* c.execute(  
 **"update notes set updateDate=CURRENT\_DATE where userId=%s and noteNumber=%s"**,  
 (USERID, noteNumber),  
 )  
 c.execute(  
 **"update notes set updateTime=CURRENT\_TIME where userId=%s and noteNumber=%s"**,  
 (USERID, noteNumber),  
 )  
 *# Update Note Title* c.execute(  
 **"update notes set noteTitle=%s where userId=%s and noteNumber=%s"**,  
 (newTitle, USERID, noteNumber),  
 )  
 db.commit()  
  
 print(**"Note Title changed Successfully!"**)  
 print(**"--------------------------"**)  
 updateNotesMenu()  
 **else**:  
 notNote(noteNumber)  
  
 **elif** userChoice == 3:  
 *# Update Note Description* noteNumber = int(input(**"Enter the Current Note Number : "**))  
 newDescription = input(**"Enter the New Note Description : "**)  
  
 **if** (noteNumber,) **in** result:  
 *# Update date and time* c.execute(  
 **"update notes set updateDate=CURRENT\_DATE where userId=%s and noteNumber=%s"**,  
 (USERID, noteNumber),  
 )  
 c.execute(  
 **"update notes set updateTime=CURRENT\_TIME where userId=%s and noteNumber=%s"**,  
 (USERID, noteNumber),  
 )  
 *# Update Note Description* c.execute(  
 **"update notes set noteDescription=%s where userId=%s and noteNumber=%s"**,  
 (newDescription, USERID, noteNumber),  
 )  
 db.commit()  
  
 print(**"Note Description changed successfully!"**)  
 print(**"--------------------------"**)  
 updateNotesMenu()  
 **else**:  
 notNote(noteNumber)  
  
 **elif** userChoice == 5:  
 home()  
 **elif** userChoice == 6:  
 modifyNote()  
 **elif** userChoice == 7:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to handle note modifications***def** modifyNote():  
 print(**"--------------------------"**)  
 print(**"Modify Notes"**)  
 print(**"--------------------------"**)  
 *# Display modification options* print(**"1. Add Note"**)  
 print(**"2. Delete Note"**)  
 print(**"3. Update Notes"**)  
 print(**"4. Home"**)  
 print(**"5. Back"**)  
 print(**"6. Exit"**)  
 *# Get user choice* userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 addNote()  
 **elif** userChoice == 2:  
 deleteNote()  
 **elif** userChoice == 3:  
 updateNotes()  
 **elif** userChoice == 4:  
 home()  
 **elif** userChoice == 5:  
 admin()  
 **elif** userChoice == 6:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to display the display notes menu and handle user choices***def** displayNotesMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 user()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to display notes***def** displayNotes():  
 *# Fetch all notes from the database* c.execute(**"SELECT \* FROM notes ORDER BY noteNumber"**)  
 result = c.fetchall()  
 db.commit()  
  
 *# Check if there are notes available* **if** result:  
 print(**f"Notes available in the Digital Library are :"**)  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. Note Number : {**row[1]**}"**)  
 print(**" "** \* r + **f"Note Title : {**row[2]**}"**)  
 print(**" "** \* r + **f"Note Description : {**row[3]**}"**)  
 print(**" "** \* r + **f"Update Date : {**row[4]**}"**)  
 print(**" "** \* r + **f"Update Time : {**row[5]**}"**)  
 print(**"--------------------------"**)  
 displayNotesMenu()  
  
 **else**:  
 *# If no notes are found* print(**"No notes found."**)  
 print(**"--------------------------"**)  
 displayNotesMenu()  
  
  
*# Function to display the search notes menu and handle user choices***def** searchNotesMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
  
 *# Handle user choices* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 searchNotes()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to search notes by note number***def** searchNotesbynoteNumber():  
 *# Get the note number to search* noteNumber = int(input(**"Enter the Note Number to search the Note : "**))  
  
 *# Execute SQL query to fetch notes with the given note number* c.execute(**"SELECT \* FROM notes WHERE bookId=%s"**, (noteNumber,))  
 result = c.fetchall()  
 db.commit()  
  
 *# Check if notes are found* **if** result:  
 print(  
 **f'Note available in the Digital Library with the Note Number "{**noteNumber**}" is :'** )  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. Note Number : {**row[1]**}"**)  
 print(**" "** \* r + **f"Note Title : {**row[2]**}"**)  
 print(**" "** \* r + **f"Note Description : {**row[3]**}"**)  
 print(**"--------------------------"**)  
 searchNotesMenu()  
  
 **else**:  
 *# If no notes are found with the given note number* print(**f'No note found with the note number "{**noteNumber**}".'**)  
 print(**"--------------------------"**)  
 searchNotesMenu()  
  
  
*# Function to search notes by keyword***def** searchNotesbyKeyword():  
 print(**"--------------------------"**)  
 print(**"Search Notes by Keyword"**)  
 print(**"--------------------------"**)  
 *# Get keyword from user* keyword = input(**"Enter a Keyword to search Notes : "**)  
  
 *# Execute SQL query to fetch notes with the given keyword in the title* c.execute(  
 **"SELECT \* FROM notes WHERE noteTitle LIKE '%{}%' ORDER BY noteNumber"**.format(  
 keyword  
 )  
 )  
 result = c.fetchall()  
 db.commit()  
  
 *# Check if notes are found* **if** result:  
 print(  
 **f'Notes available in the Digital Library with the Keyword "{**keyword**}" are :'** )  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. Note Number : {**row[1]**}"**)  
 print(**" "** \* r + **f"Note Title : {**row[2]**}"**)  
 print(**" "** \* r + **f"Note Description : {**row[3]**}"**)  
 print(**"--------------------------"**)  
 searchNotesMenu()  
  
 **else**:  
 *# If no notes are found with the given keyword* print(**f'No notes found with the keyword "{**keyword**}".'**)  
 print(**"--------------------------"**)  
 searchNotesMenu()  
  
  
*# Function to handle note searching***def** searchNotes():  
 print(**"--------------------------"**)  
 print(**"Search Notes"**)  
 print(**"--------------------------"**)  
 *# Display search options* print(**"1. Search by Note Number"**)  
 print(**"2. Search by Keyword"**)  
 print(**"3. Home"**)  
 print(**"4. Back"**)  
 print(**"5. Exit"**)  
 *# Get user choice* userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 searchNotesbynoteNumber()  
 **elif** userChoice == 2:  
 searchNotesbyKeyword()  
 **elif** userChoice == 3:  
 notes()  
 **elif** userChoice == 4:  
 modifyNote()  
 **elif** userChoice == 5:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to display the change admin menu and handle user choices***def** changeAdminMenu():  
 print(**"1. Home"**)  
 print(**"2. Back"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 home()  
 **elif** userChoice == 2:  
 admin()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to change the admin status***def** changeAdmin():  
 print(**"--------------------------"**)  
 print(**"Change Admin"**)  
 print(**"--------------------------"**)  
 *# Get new admin's ID and password from the user* newAdminId = int(input(**"Enter the New Admin's User ID : "**))  
 newAdminPassword = input(**"Enter the New Admin's Password : "**)  
 choice = input(**"Are you sure to change the Admin? (Yes/No) : "**)  
 print(**"--------------------------"**)  
  
 *# Check if the entered user ID exists* c.execute(**"SELECT password FROM users WHERE userId=%s"**, (newAdminId,))  
 result = c.fetchall()  
 db.commit()  
  
 *# Check the user's choice to proceed or cancel* **if** choice.lower() **in** [**"yes"**, **"y"**]:  
 *# If the user ID is not valid, print an error message* **if** len(result) == 0:  
 print(**"Please enter a valid user id!"**)  
 **else**:  
 *# If the entered password matches the user's password* **if** newAdminPassword == result[0][0]:  
 *# Update admin status for all users* c.execute(  
 **"UPDATE users SET adminStatus='not admin' WHERE adminStatus ='admin'"** )  
 c.execute(  
 **"UPDATE users SET adminStatus='admin' WHERE userId =%s"**,  
 (newAdminId,),  
 )  
 db.commit()  
  
 print(**"Admin Changed Successfully!"**)  
 print(**"--------------------------"**)  
 changeAdminMenu()  
  
 **else**:  
 print(**"Please enter a valid password!"**)  
 **elif** choice.lower() **in** [**"no"**, **"n"**]:  
 print(**"Admin Not Changed!"**)  
 print(**"--------------------------"**)  
 changeAdminMenu()  
 **else**:  
 validOption()  
  
  
*# Function to authenticate admin***def** authAdmin():  
 print(**"--------------------------"**)  
 print(**"Admin Authentication"**)  
 print(**"--------------------------"**)  
 adminId = int(input(**"Enter the Admin's User ID : "**))  
 adminPassword = input(**"Enter the Admin's User Password : "**)  
  
 *# Check if the entered admin ID exists* c.execute(**"SELECT password FROM users WHERE userId=%s"**, (adminId,))  
 result = c.fetchall()  
 db.commit()  
  
 *# If the entered admin ID is not valid, print an error message* **if** len(result) == 0:  
 print(**"--------------------------"**)  
 print(**"Please enter a valid user id!"**)  
 print(**"--------------------------"**)  
 **else**:  
 *# If the entered password matches the admin's password* **if** adminPassword == result[0][0]:  
 **global** USERID  
 USERID = adminId  
 print(**"\033[0;35m--------------------------\033[0;0m"**)  
 print(**"\033[0;36mAdmin is verified successfully.\033[0;0m"**)  
 print(**"\033[0;35m--------------------------\033[0;0m"**)  
 admin() *# Call the admin menu* **else**:  
 print(**"Please enter a valid password!"**)  
 print(**"--------------------------"**)  
  
  
*# Function to display the admin menu***def** admin():  
 print(**"--------------------------"**)  
 print(**"Admin"**)  
 print(**"--------------------------"**)  
 print(**"1. Login into User Panel"**)  
 print(**"2. Modify User"**)  
 print(**"3. Display Users"**)  
 print(**"4. Search Users"**)  
 print(**"5. Modify Book"**)  
 print(**"6. Issue Book"**)  
 print(**"7. Return Book"**)  
 print(**"8. Change Admin"**)  
 print(**"9. Home"**)  
 print(**"10. Back"**)  
 print(**"11. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 print(**"You are successfully login into user panel."**)  
 print(**"--------------------------"**)  
  
 user()  
 **elif** userChoice == 2:  
 modifyUser()  
 **elif** userChoice == 3:  
 displayUsers()  
 **elif** userChoice == 4:  
 searchUsers()  
 **elif** userChoice == 5:  
 modifyBook()  
 **elif** userChoice == 6:  
 issueBook()  
 **elif** userChoice == 7:  
 returnBook()  
 **elif** userChoice == 8:  
 changeAdmin()  
 **elif** userChoice == 9:  
 home()  
 **elif** userChoice == 10:  
 authAdmin()  
 **elif** userChoice == 11:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to authenticate a user***def** authUser():  
 print(**"--------------------------"**)  
 print(**"User Authentication"**)  
 print(**"--------------------------"**)  
 userId = int(input(**"Enter the User ID : "**))  
 password = input(**"Enter the User Password : "**)  
  
 *# Check if the entered user ID exists* c.execute(**"SELECT password FROM users WHERE userId=%s"**, (userId,))  
 result = c.fetchall()  
 db.commit()  
  
 *# If the entered user ID is not valid, print an error message* **if** len(result) == 0:  
 print(**"--------------------------"**)  
 print(**"Please enter a valid user id!"**)  
 print(**"--------------------------"**)  
 **else**:  
 *# If the entered password matches the user's password* **if** password == result[0][0]:  
 **global** USERID  
 USERID = userId  
 print(**"\033[0;35m--------------------------\033[0;0m"**)  
 print(**"\033[0;36mUser is verified successfully.\033[0;0m"**)  
 print(**"\033[0;35m--------------------------\033[0;0m"**)  
 user() *# Call the user menu* **else**:  
 print(**"Please Enter a Valid Password!"**)  
 print(**"--------------------------"**)  
  
  
*# Function to search & display the wikipedia articles***def** wikipediaArticles():  
 *# Function to fetch article details* **def** fetchingArticle(keyword, articleLength=1500):  
 *# Creating a Wikipedia API object* wiki = wikipediaapi.Wikipedia(language=**"en"**, user\_agent=**"digital-library/1.1"**)  
 *# Fetching the page for the given search query* page = wiki.page(keyword)  
  
 *# Checking if the page exists* **if not** page.exists():  
 print(  
 **f'Sorry, the Wikipedia Article for the keyword "{**keyword**}" does not exists.'** )  
 print(**"--------------------------"**)  
 **else**:  
 *# Displaying article title* print(**"Title : "**)  
 print(page.title)  
 print(**"URL : "**)  
 print(page.fullurl)  
 *# Displaying a summary of the article within the specified length* print(**"Summary : "**)  
  
 start = 0  
 end = 157  
 article = page.summary[:articleLength]  
  
 **while** end <= articleLength:  
 print(article[start:end])  
 start += 157  
 end += 157  
 **else**:  
 print(**"--------------------------"**)  
  
 print(**"--------------------------"**)  
 print(**"Search Articles"**)  
 print(**"--------------------------"**)  
 *# Taking user input for the keyword and article length* keyword = input(**"Enter the Keyword for searching the Wikipedia Article : "**)  
 articleLength = int(input(**"Enter the Article Length : "**))  
 print(**"--------------------------"**)  
  
 *# Calling the function to fetch and display the article* fetchingArticle(keyword, articleLength)  
  
 userMenu()  
  
  
*# Function to search & display the news***def** news():  
 **def** fetchNews(apiKey, country=**"in"**, category=**"science"**, numArticles=5):  
 url = **f"https://newsapi.org/v2/top-headlines"** params = {  
 **"apiKey"**: apiKey,  
 **"country"**: country,  
 **"category"**: category,  
 **"pageSize"**: numArticles,  
 }  
 response = requests.get(url, params=params)  
  
 **if** response.status\_code == 200:  
 news\_data = response.json()  
 articles = news\_data.get(**"articles"**, [])  
  
 **for** i, article **in** enumerate(articles, start=1):  
 print(**f"{**i**}. {**article[**'title'**]**}"**)  
 print(**f" Source: {**article[**'source'**][**'name'**]**}"**)  
 print(**f" URL: {**article[**'url'**]**}"**)  
 print(**"--------------------------"**)  
  
 **else**:  
 print(**f"Error {**response.status\_code**}: {**response.text**}"**)  
 print(**"--------------------------"**)  
  
 API\_KEY = **"YOUR\_API\_KEY"** print(**"--------------------------"**)  
 print(**"News"**)  
 print(**"--------------------------"**)  
 print(**"Country codes are : "**)  
 print(**"https://newsapi.org/sources"**)  
 print(**"Categories are : "**)  
 print(**"business, entertainment, general, health, science, sports, technology"**)  
 print(**"--------------------------"**)  
 country = input(**"Enter the Country Code : "**)  
 category = input(**"Enter the Category : "**)  
 numArticles = int(input(**"Enter the Number of Articles : "**))  
 print(**"--------------------------"**)  
  
 fetchNews(API\_KEY, country, category, numArticles)  
  
 userMenu()  
  
  
*# Function to display the issued books details of a user***def** issuedBooksDetails():  
 print(**"--------------------------"**)  
 print(**"Issued Books Details"**)  
 print(**"--------------------------"**)  
 returnPolicy()  
  
 c.execute(  
 **"SELECT \* FROM issuedBooksDetails WHERE userId=%s ORDER BY bookId"**, (USERID,)  
 )  
 result = c.fetchall()  
 db.commit()  
  
 **if** result == []:  
 print(**"No Books Issued!"**)  
 print(**"--------------------------"**)  
 userMenu()  
  
 **else**:  
 i = 0  
 **for** row **in** result:  
 i += 1  
 r = length(i)  
 print(**f"{**i**}. Book ID : "**, row[1])  
 print(**" "** \* r + **"Book Name : "**, row[2])  
 print(**" "** \* r + **"Issue Date : "**, row[3])  
 print(**" "** \* r + **"Issue Time : "**, row[4])  
 print(**" "** \* r + **"Return Date : "**, row[5])  
 print(**" "** \* r + **"Return Time : "**, row[6])  
 print(**" "** \* r + **"Fine(in Rs.) : "**, row[7])  
 print(**"--------------------------"**)  
 userMenu()  
  
  
*# Function to display the user menu***def** user():  
 print(**"--------------------------"**)  
 print(**"User"**)  
 print(**"--------------------------"**)  
 *# Check if the entered user ID exists* c.execute(**'SELECT userId FROM users WHERE adminStatus="admin"'**)  
 result = c.fetchall()  
 db.commit()  
  
 **if** result[0][0] == USERID:  
 print(**"1. Login into Admin Panel"**)  
 print(**"2. About the Library"**)  
 print(**"3. News"**)  
 print(**"4. Wikipedia Articles"**)  
 print(**"5. Display Books"**)  
 print(**"6. Search Books"**)  
 print(**"7. Issued Books Details"**)  
 print(**"8. Notes"**)  
 print(**"9. Home"**)  
 print(**"10. Back"**)  
 print(**"11. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 print(**"You are successfully login into admin panel."**)  
 print(**"--------------------------"**)  
  
 admin()  
 **elif** userChoice == 2:  
 aboutLibrary()  
 **elif** userChoice == 3:  
 news()  
 **elif** userChoice == 4:  
 wikipediaArticles()  
 **elif** userChoice == 5:  
 displayBooks()  
 **elif** userChoice == 6:  
 searchBooks()  
 **elif** userChoice == 7:  
 issuedBooksDetails()  
 **elif** userChoice == 8:  
 notes()  
 **elif** userChoice == 9:  
 home()  
 **elif** userChoice == 10:  
 authUser()  
 **elif** userChoice == 11:  
 exiting()  
 **else**:  
 validOption()  
 **else**:  
 print(**"1. About Library"**)  
 print(**"2. News"**)  
 print(**"3. Wikipedia Articles"**)  
 print(**"4. Display Books"**)  
 print(**"5. Search Books"**)  
 print(**"6. Issued Books Details"**)  
 print(**"7. Notes"**)  
 print(**"8. Home"**)  
 print(**"9. Back"**)  
 print(**"10. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 aboutLibrary()  
 **elif** userChoice == 2:  
 news()  
 **elif** userChoice == 3:  
 wikipediaArticles()  
 **elif** userChoice == 4:  
 displayBooks()  
 **elif** userChoice == 5:  
 searchBooks()  
 **elif** userChoice == 6:  
 issuedBooksDetails()  
 **elif** userChoice == 7:  
 notes()  
 **elif** userChoice == 8:  
 home()  
 **elif** userChoice == 9:  
 authUser()  
 **elif** userChoice == 10:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Function to display the main menu***def** home():  
 **while True**:  
 print(**"=========================="**)  
 print(**"\033[1;32m~~~~~~~~~~~~~~~~~~~~~~~~~~\033[0;0m"**)  
 print(  
 **"\033[1;31m"** + pyfiglet.figlet\_format(**"Welcome to the"**, font=**"banner3"**, width=1000)  
 )  
 print(  
 pyfiglet.figlet\_format(**"Digital Library"**, font=**"banner3"**, width=1000)  
 + **"\033[0;0m"** )  
 print(**"\033[1;32m~~~~~~~~~~~~~~~~~~~~~~~~~~\033[0;0m"**)  
 print(**"=========================="**)  
 print(**"--------------------------"**)  
 print(**"Home"**)  
 print(**"--------------------------"**)  
 print(**"1. Admin"**)  
 print(**"2. User"**)  
 print(**"3. Exit"**)  
 userChoice = int(input(**"Enter your Choice to Continue : "**))  
 print(**"--------------------------"**)  
  
 *# Handle user choices* **if** userChoice == 1:  
 authAdmin()  
 **elif** userChoice == 2:  
 authUser()  
 **elif** userChoice == 3:  
 exiting()  
 **else**:  
 validOption()  
  
  
*# Call the main menu function*home()