

pre code

```
from datetime import datetime, timedelta # for handling borrow times and reservations

# ----- DATA STORAGE -----
# storing all users; admin is built-in
users = {
    "04-2024-022": {"name": "Admin", "role": "Admin", "borrowed_books": [], "reservations": []}
}

borrowed_books = {} # track which books are borrowed
auditorium_reservations = [] # track auditorium reservations
announcements = ["LRC will be closed on Friday for maintenance."]

ADMIN_ID = "04-2024-022"

# ----- INTRO & OUTRO -----
def intro():
    # welcome message
    print("\n=====")
    print(" TAKDA - LRC MANAGEMENT SYSTEM")
    print(" Tulong sa Aklatan para sa Kaayusan,")
    print(" Dues, at Appointments")
    print("=====")
    print("Manage your library tasks easily.\n")

def outro():
    # exit message
    print("\n=====")
    print("Thank you for using TAKDA.")
    print("Have a productive day!")
    print("=====")

def back_to_menu():
    # simple pause before returning to main menu
    input("\nPress ENTER to return to the Main Menu...")

# ----- LOGIN / REGISTER -----
def register_user():
    # register new student or faculty
    print("\n--- REGISTER ---")
    role_input = input("Are you a Student or Faculty? (S/F): ").strip().upper()
    student_id = input("Enter ID: ").strip()

    if student_id in users:
        print("This ID already exists. Try logging in.")
        return None

    name = input("Enter Name: ").strip()
    section = ""
```

```

if role_input == "S":
    section = input("Enter Section: ").strip()

role = "Faculty" if role_input == "F" else "Student"

users[student_id] = {
    "name": name,
    "section": section,
    "role": role,
    "borrowed_books": [],
    "reservations": []
}
print("Registration successful!")
return student_id

def login_user():
    # login for existing users
    print("\n--- LOGIN ---")
    student_id = input("Enter ID: ").strip()
    if student_id in users:
        print(f"Welcome, {users[student_id]['name']} ({users[student_id]['role']}!)")
        return student_id
    else:
        print("User not found. Please register first.")
        return None

# ----- STUDENT / FACULTY FEATURES -----

def show_announcements():
    # display all announcements
    print("\n--- LRC ANNOUNCEMENTS ---")
    for a in announcements:
        print("-", a)
    back_to_menu()

def borrow_book(student_id):
    # borrow a book with 1-day return
    print("\n--- BORROW BOOK ---")
    book = input("Enter the book title: ").strip()

    if book in borrowed_books:
        print(f"Sorry! '{book}' is already borrowed.")
        back_to_menu()
        return

    borrow_time = datetime.now()
    return_time = borrow_time + timedelta(days=1)
    borrowed_books[book] = student_id
    users[student_id]["borrowed_books"].append({
        "book": book,
        "borrow_time": borrow_time,
        "return_time": return_time
    })

```

```

print(f"Book '{book}' borrowed! Return by {return_time.strftime('%Y-%m-%d %H:%M')}")
back_to_menu()

def reserve_auditorium(student_id):
    # request an auditorium reservation
    print("\n--- AUDITORIUM RESERVATION ---")
    date = input("Enter reservation date (YYYY-MM-DD): ").strip()
    start_time = input("Enter start time (HH:MM, 24-hour): ").strip()
    end_time = input("Enter end time (HH:MM, 24-hour): ").strip()
    purpose = input("Enter purpose of use: ").strip()

    auditorium_reservations.append({
        "student_id": student_id,
        "date": date,
        "start_time": start_time,
        "end_time": end_time,
        "purpose": purpose,
        "status": "Pending" # placeholder for future approval
    })
    print("Reservation request submitted!")
    back_to_menu()

```

```
# ----- ADMIN FEATURES -----
```

```

def admin_make_announcement():
    # admin can add announcements
    print("\n--- MAKE ANNOUNCEMENT ---")
    announcement = input("Enter announcement text: ").strip()
    if announcement:
        announcements.append(announcement)
        print("Announcement added!")
    else:
        print("No announcement entered.")
    back_to_menu()

def admin_approval():
    # simple display of pending reservations
    print("\n--- PENDING AUDITORIUM RESERVATIONS ---")
    for r in auditorium_reservations:
        if r["status"] == "Pending":
            user = users[r["student_id"]]
            print(f"Student: {user['name']} | Date: {r['date']} | Time: {r['start_time']} - {r['end_time']} |"
            f"Purpose: {r['purpose']}")
```

```
# ----- MAIN MENU -----
```

```

def main_menu(student_id):
    # show menu based on user role
    while True:
        print("\n--- MAIN MENU ---")
        if student_id == ADMIN_ID:
            print("1 - Make Announcement")
            print("2 - View Pending Reservations")

```

```

        print("0 - Logout / Exit")
else:
    print("1 - View Announcements")
    print("2 - Borrow Book")
    print("3 - Reserve Auditorium")
    print("0 - Logout / Exit")

choice = input("Choose an option: ")

if student_id == ADMIN_ID:
    if choice == "1":
        admin_make_announcement()
    elif choice == "2":
        admin_approval()
    elif choice == "0":
        print("Logging out...")
        break
    else:
        print("Invalid choice. Try again.")
else:
    if choice == "1":
        show_announcements()
    elif choice == "2":
        borrow_book(student_id)
    elif choice == "3":
        reserve_auditorium(student_id)
    elif choice == "0":
        print("Logging out...")
        break
    else:
        print("Invalid choice. Try again.")

```

----- RUN PROGRAM -----

```

intro()
while True:
    print("\n1 - Login")
    print("2 - Register")
    print("0 - Exit")
    choice = input("Choose an option: ")
    if choice == "1":
        user_id = login_user()
        if user_id:
            main_menu(user_id)
    elif choice == "2":
        user_id = register_user()
        if user_id:
            main_menu(user_id)
    elif choice == "0":
        outro()
        break
    else:
        print("Invalid choice. Try again.")

```