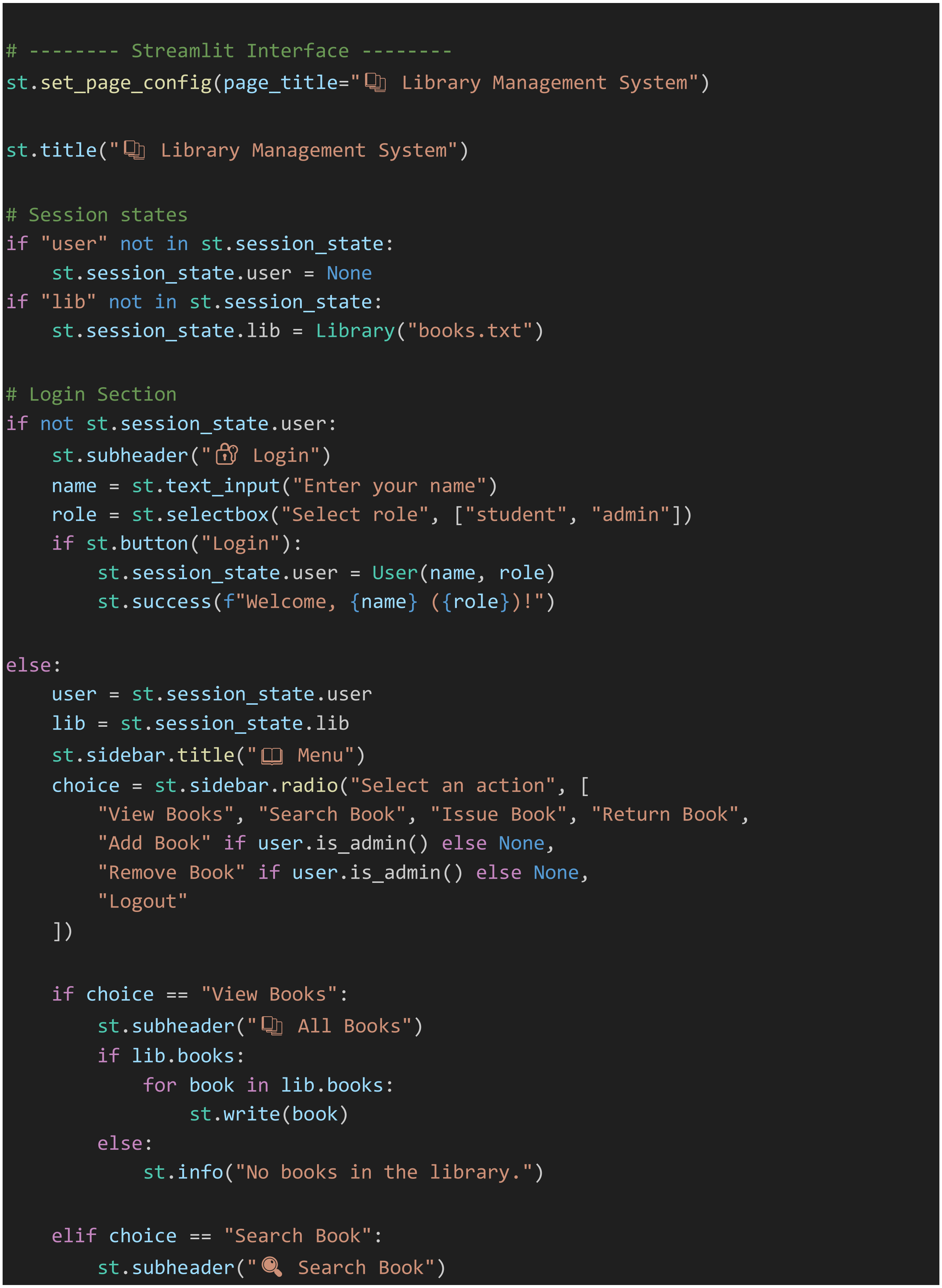
WEEK 1 TASKS (CONTINUE..)

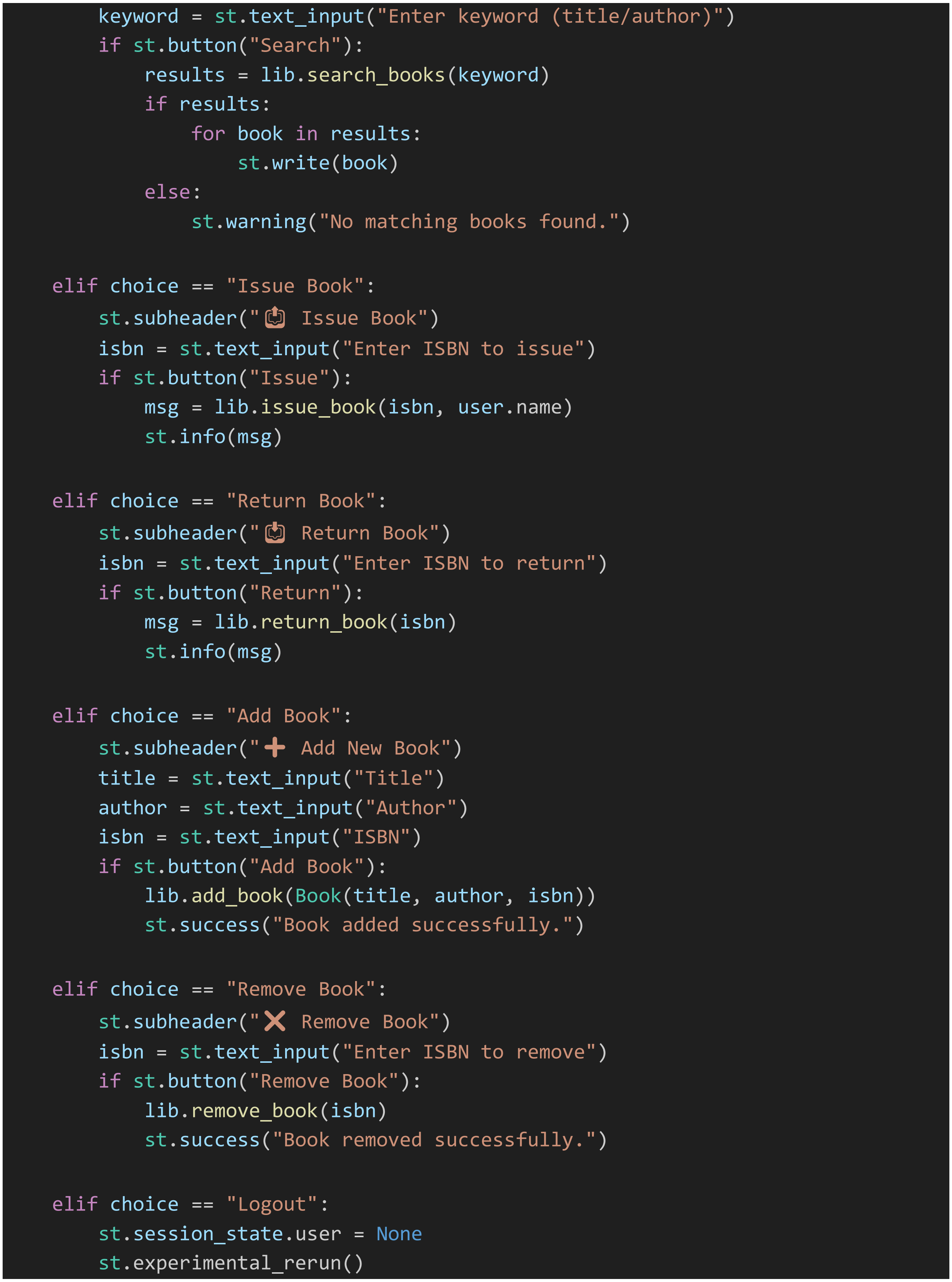
CODE:

PYTHON:

```
import streamlit as st
import os

# -------- Book Class --------
class Book:
    def __init__(self, title, author, isbn):
        self.title = title
        self.author = author
        self.isbn = isbn
        self.issued_to = None

    def __str__(self):
        status = f"Issued to {self.issued_to}" if self.issued_to else "Available"
        return f"{self.title} by {self.author} | ISBN: {self.isbn} | Status:
{status}"

# -------- User Class --------
class User:
    def __init__(self, name, role):
        self.name = name
        self.role = role

    def is_admin(self):
        return self.role == 'admin'

# -------- Library Class --------
class Library:
    def __init__(self, filename):
        self.books = []
        self.filename = filename
        self.load_books()

    def load_books(self):
        if not os.path.exists(self.filename):
            return
        with open(self.filename, 'r') as f:
            for line in f:
                title, author, isbn, issued_to = line.strip().split('|')
                book = Book(title, author, isbn)
                book.issued_to = None if issued_to == 'None' else issued_to
                self.books.append(book)

    def save_books(self):
        with open(self.filename, 'w') as f:
            for book in self.books:

f.write(f"{book.title}|{book.author}|{book.isbn}|{book.issued_to}\n")

    def add_book(self, book):
        self.books.append(book)
        self.save_books()

    def remove_book(self, isbn):
        self.books = [b for b in self.books if b.isbn != isbn]
        self.save_books()

    def search_books(self, keyword):
        return [book for book in self.books if keyword.lower() in
book.title.lower() or keyword.lower() in book.author.lower()]

    def issue_book(self, isbn, user_name):
        for book in self.books:
            if book.isbn == isbn:
                if book.issued_to:
                    return "Book already issued."
                else:
                    book.issued_to = user_name
                    self.save_books()
                    return "Book issued successfully."
        return "Book not found."

    def return_book(self, isbn):
        for book in self.books:
            if book.isbn == isbn:
                if book.issued_to:
                    book.issued_to = None
                    self.save_books()
                    return "Book returned successfully."
                else:
                    return "Book was not issued."
        return "Book not found."
```



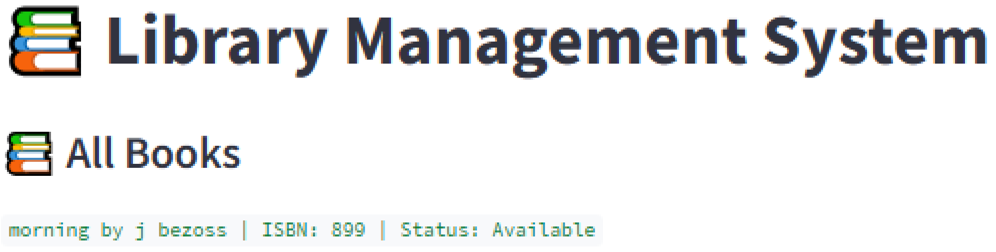
```python
# -------- Streamlit Interface --------
st.set_page_config(page_title="📚 Library Management System")

st.title("📚 Library Management System")

# Session states
if "user" not in st.session_state:
    st.session_state.user = None
if "lib" not in st.session_state:
    st.session_state.lib = Library("books.txt")

# Login Section
if not st.session_state.user:
    st.subheader("🔐 Login")
    name = st.text_input("Enter your name")
    role = st.selectbox("Select role", ["student", "admin"])
    if st.button("Login"):
        st.session_state.user = User(name, role)
        st.success(f"Welcome, {name} ({role})!")

else:
    user = st.session_state.user
    lib = st.session_state.lib
    st.sidebar.title("📖 Menu")
    choice = st.sidebar.radio("Select an action", [
        "View Books", "Search Book", "Issue Book", "Return Book",
        "Add Book" if user.is_admin() else None,
        "Remove Book" if user.is_admin() else None,
        "Logout"
    ])

    if choice == "View Books":
        st.subheader("📚 All Books")
        if lib.books:
            for book in lib.books:
                st.write(book)
        else:
            st.info("No books in the library.")

    elif choice == "Search Book":
        st.subheader("🔍 Search Book")
```



```
        keyword = st.text_input("Enter keyword (title/author)")
        if st.button("Search"):
            results = lib.search_books(keyword)
            if results:
                for book in results:
                    st.write(book)
            else:
                st.warning("No matching books found.")

    elif choice == "Issue Book":
        st.subheader("📤 Issue Book")
        isbn = st.text_input("Enter ISBN to issue")
        if st.button("Issue"):
            msg = lib.issue_book(isbn, user.name)
            st.info(msg)

    elif choice == "Return Book":
        st.subheader("📥 Return Book")
        isbn = st.text_input("Enter ISBN to return")
        if st.button("Return"):
            msg = lib.return_book(isbn)
            st.info(msg)

    elif choice == "Add Book":
        st.subheader("➕ Add New Book")
        title = st.text_input("Title")
        author = st.text_input("Author")
        isbn = st.text_input("ISBN")
        if st.button("Add Book"):
            lib.add_book(Book(title, author, isbn))
            st.success("Book added successfully.")

    elif choice == "Remove Book":
        st.subheader("❌ Remove Book")
        isbn = st.text_input("Enter ISBN to remove")
        if st.button("Remove Book"):
            lib.remove_book(isbn)
            st.success("Book removed successfully.")

    elif choice == "Logout":
        st.session_state.user = None
        st.experimental_rerun()
```

[redacted]

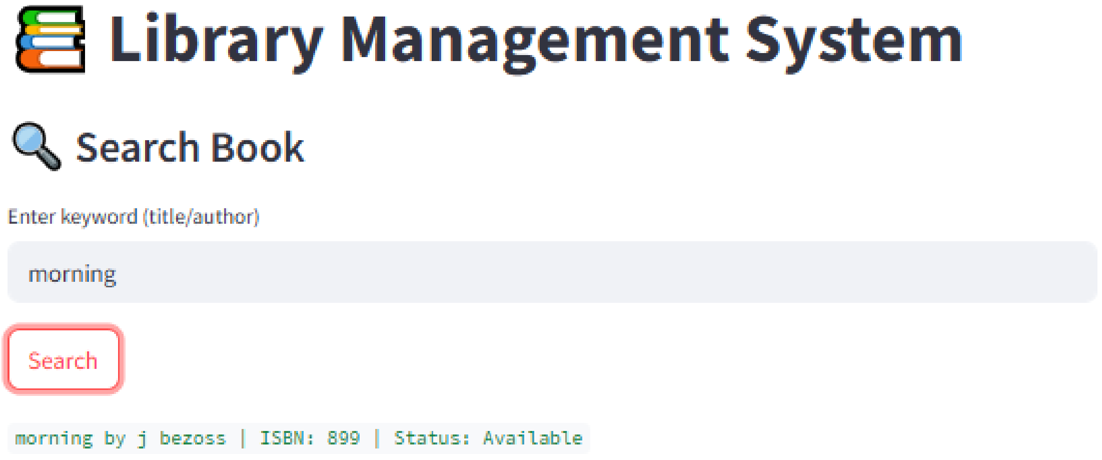
OUTPUT:



📚 Library Management System

📚 All Books

```
morning by j bezoss | ISBN: 899 | Status: Available
```



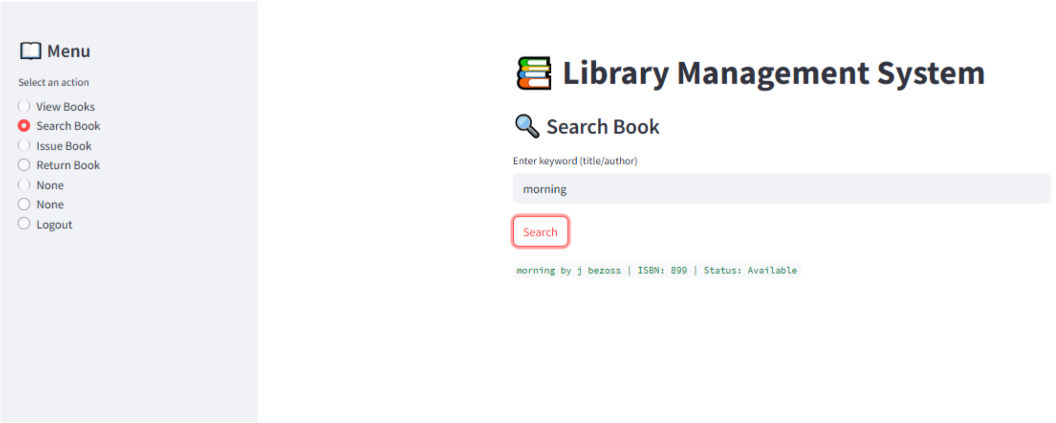
📚 Library Management System

🔍 Search Book

Enter keyword (title/author)

morning

```
morning by j bezoss | ISBN: 899 | Status: Available
```



📚 Library Management System

🔍 Search Book

Enter keyword (title/author)

morning

```
morning by j bezoss | ISBN: 899 | Status: Available
```