Group 7

Group members:

1. **Muhammad Firdaus Hakimi - A20MJ5014**
2. **Abdullah Aiman bin Jaafar – B21MJ0001**
3. **Syarif Hidayatullah bin Muhammad Nasir – A21MJ0093**

Pseudo-code:

1. Initialize Flask app and set a secret key.
   - Create a Flask instance.
   - Set a secret key for session security.

2. Define a catalog of books with title, author, genre, price, and cover image.
   - Create a list of dictionaries representing books.

3. Define routes for:
   - Home page ("/") to display the book catalog.
   - Search ("/search") to search for books based on title or author.
   - Add to Cart ("/add_to_cart/<int:book_index>") to add a book to the shopping cart.
   - View Cart ("/cart") to display the items in the shopping cart.
   - Clear Cart ("/clear_cart") to clear the entire shopping cart.
   - Sign Up ("/signup") to register a new user.
   - Login ("/login") to authenticate a user.
   - Logout ("/logout") to log out a user.
   - Purchase ("/purchase") to complete the purchase.

4. Implement functions for each route:
   4.1 home():
      - Render the 'index.html' template with the book catalog.

   4.2 search():
      - Retrieve the search query from the form.
      - Filter books based on the query.
      - Render the 'search_results.html' template with search results.

   4.3 add_to_cart(book_index):
      - Retrieve the existing cart from the session or initialize an empty list.
      - Append the book index to the cart.
      - Update the session with the modified cart.
      - Redirect to the home page.

   4.4 view_cart():
      - Retrieve cart indices from the session.
      - Create a list of unique books in the cart with quantities.

- Calculate the total price of items in the cart.
- Render the 'cart.html' template with cart details.

  4.5 clear_cart():
- Remove the 'cart' key from the session.
- Redirect to the home page.

  4.6 signup():
- Handle both GET and POST requests.
- If POST, validate and create a new user, then log in.
- If GET, render the 'signup.html' template.

  4.7 login():
- Handle both GET and POST requests.
- If POST, authenticate the user and log in.
- If GET, render the 'login.html' template.

  4.8 logout():
- Clear the username from the session.
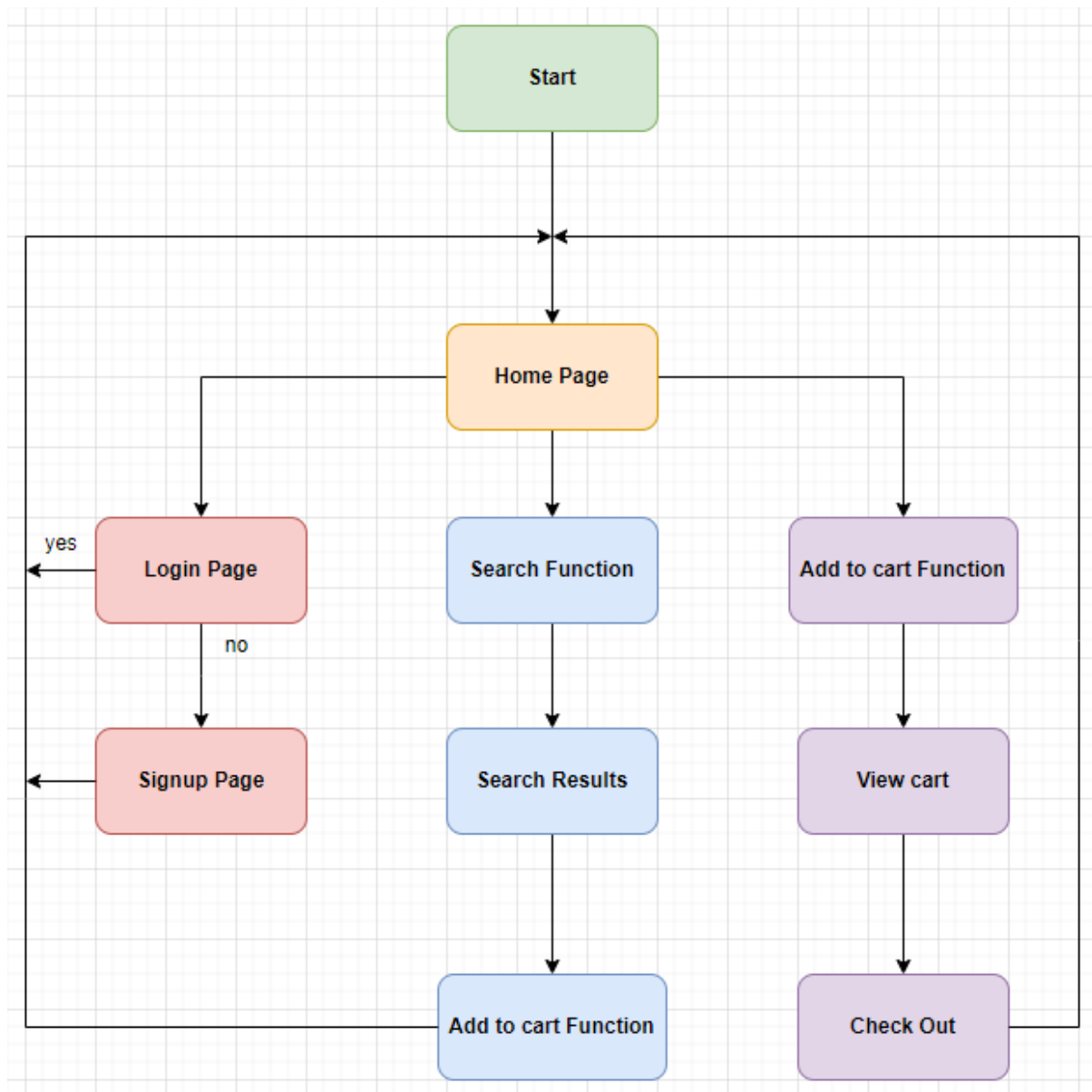- Render the 'logout.html' template.

  4.9 purchase():
- Retrieve cart indices from the session.
- Calculate the total price of items in the cart.
- Retrieve the current user.
- Check if the user has enough balance for the purchase.
- Deduct the total price from the user's balance and update the cart.
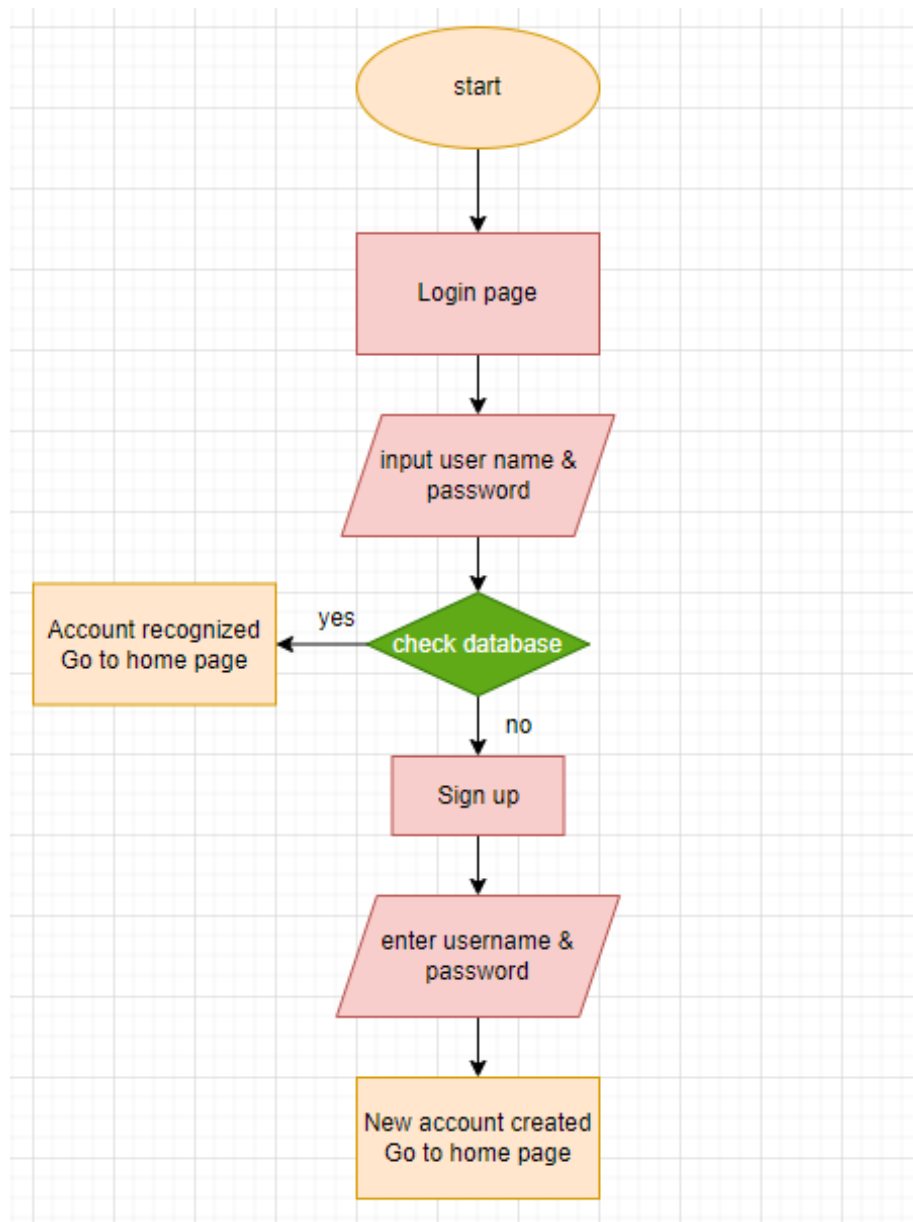- Render 'purchase_success.html' or 'purchase_error.html' templates accordingly.

5. Run the Flask app.
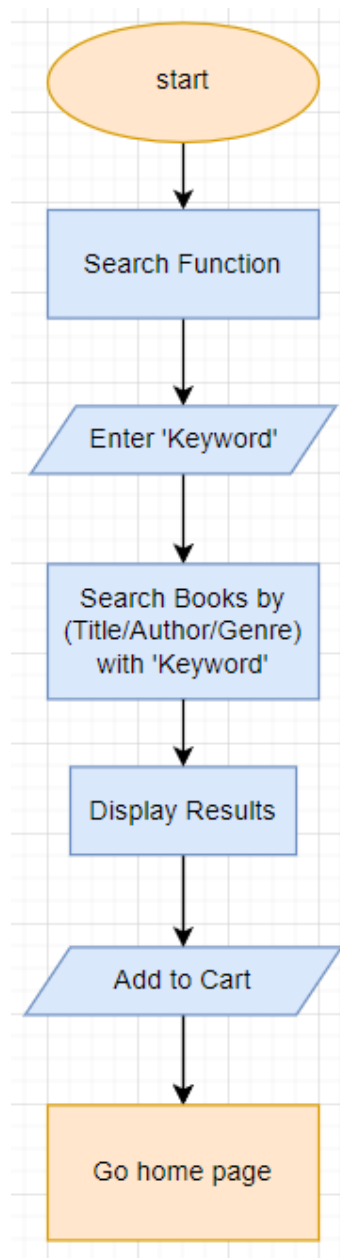 - Start the Flask app, listening for incoming requests.

Program Flowchart

Flowchart with the following nodes and connections:

- **Start** → **Home Page**
- **Home Page** branches to:
  - **Login Page** — yes → (loops back); no → **Signup Page** → (loops back)
  - **Search Function** → **Search Results** → **Add to cart Function**
  - **Add to cart Function** → **View cart** → **Check Out**

Login/Signup Function Flowchart

Search Function

Cart Function Flowchart