Streamlit tutorial part 2

> **Page 1:** Login/Register tabs → on successful login → go to **Dashboard page**.

1. Folder structure for a multi-page app
2. First page with **Login/Register tabs**
3. A **Dashboard page** that only logged-in users can see
4. How the redirect works with `st.switch_page` + `st.session_state`

---

1. Project structure

Create a folder like this:

```
my_app/
│
├─ Home.py                # First page with Login/Register tabs
└─ pages/
   └─ 1_Dashboard.py     # Dashboard shown after login
```

Streamlit automatically turns:

- `Home.py` into the **main page**
- every `.py` inside `pages/` into additional pages (ordered by prefix number). ([docs.streamlit.io][1])

Run the app from inside `my_app/`:

```
streamlit run Home.py
```

---

2. First page: Login / Register tabs

Home.py

```python
import streamlit as st

st.set_page_config(page_title="Login / Register", page_icon="🔑",
layout="centered")

# ---------- Initialise session state ----------
if "users" not in st.session_state:
    # Very simple in-memory "database": {username: password}
    st.session_state.users = {}

if "logged_in" not in st.session_state:
    st.session_state.logged_in = False

if "username" not in st.session_state:
```

File failed to load: file:///Users/anna29/Desktop/Streamlit_part2_files/extensions/MathZoom.js

```python
    st.session_state.username = ""

st.title("🔐 Welcome")

# If already logged in, go straight to dashboard (optional)
if st.session_state.logged_in:
    st.success(f"Already logged in as **{st.session_state.username}**.")
    if st.button("Go to dashboard"):
        # Use the official navigation API to switch pages
        st.switch_page("pages/1_Dashboard.py")  # path is relative to Home.py :contentReference[oaicite:1]{index=1}
    st.stop()  # Don’t show login/register again


# ---------- Tabs: Login / Register ----------
tab_login, tab_register = st.tabs(["Login", "Register"])

# ----- LOGIN TAB -----
with tab_login:
    st.subheader("Login")

    login_username = st.text_input("Username", key="login_username")
    login_password = st.text_input("Password", type="password", key="login_password")

    if st.button("Log in", type="primary"):
        # Simple credential check (for teaching only – not secure!)
        users = st.session_state.users
        if login_username in users and users[login_username] == login_password:
            st.session_state.logged_in = True
            st.session_state.username = login_username
            st.success(f"Welcome back, {login_username}! 🎉")

            # Redirect to dashboard page
            st.switch_page("pages/1_Dashboard.py")
        else:
            st.error("Invalid username or password.")


# ----- REGISTER TAB -----
with tab_register:
    st.subheader("Register")

    new_username = st.text_input("Choose a username", key="register_username")
    new_password = st.text_input("Choose a password", type="password", key="register_password")
    confirm_password = st.text_input("Confirm password", type="password", key="register_confirm")

        # Basic checks – again, just for teaching
```

File failed to load: file:///Users/anna29/Desktop/Streamlit_part2_files/extensions/MathZoom.js

```
        if not new_username or not new_password:
            st.warning("Please fill in all fields.")
        elif new_password != confirm_password:
            st.error("Passwords do not match.")
        elif new_username in st.session_state.users:
            st.error("Username already exists. Choose another
one.")
        else:
            # "Save" user in our simple in-memory store
            st.session_state.users[new_username] = new_password
            st.success("Account created! You can now log in from
the Login tab.")
            st.info("Tip: go to the Login tab and sign in with
your new account.")
```

What's happening here?

- We use **tabs** to toggle between Login and Register:

  ```
  tab_login, tab_register = st.tabs(["Login", "Register"])
  ```
- We store data in **`st.session_state`** so it persists across reruns/pages:
  - `users` : a dict of registered users (just in memory)
  - `logged_in` : `True/False`
  - `username` : current user's name
- When login is successful:
  - Set `logged_in = True`
  - Set `username`
  - Call `st.switch_page("pages/1_Dashboard.py")` to jump to the Dashboard page. ([docs.streamlit.io][2])

Important:

This is **not secure** and only for learning. For real apps, use proper authentication (e.g. OIDC with `st.login()` or libraries like `streamlit-authenticator`). ([docs.streamlit.io][3])

---

3. Dashboard page (only after login)

pages/1_Dashboard.py

```
import streamlit as st
import pandas as pd
import numpy as np

st.set_page_config(page_title="Dashboard", page_icon="📊",
layout="wide")

# Ensure state keys exist (in case user opens this page first)
[illegible]:
```

File failed to load: file:///Users/anna29/Desktop/Streamlit_part2_files/extensions/MathZoom.js

```python
    st.session_state.logged_in = False
if "username" not in st.session_state:
    st.session_state.username = ""

# Guard: if not logged in, send user back
if not st.session_state.logged_in:
    st.error("You must be logged in to view the dashboard.")
    if st.button("Go to login page"):
        st.switch_page("Home.py")   # back to the first page
    st.stop()

# If logged in, show dashboard content
st.title("📊 Dashboard")
st.success(f"Hello, **{st.session_state.username}**! You are 
logged in.")

# Example dashboard layout
st.caption("This is just demo content — replace with your own 
dashboard.")

# Sidebar filters
with st.sidebar:
    st.header("Filters")
    n_points = st.slider("Number of data points", 10, 200, 50)

# Fake data
data = pd.DataFrame(
    np.random.randn(n_points, 3),
    columns=["A", "B", "C"]
)

col1, col2 = st.columns(2)

with col1:
    st.subheader("Line chart")
    st.line_chart(data)

with col2:
    st.subheader("Bar chart")
    st.bar_chart(data)

with st.expander("See raw data"):
    st.dataframe(data)

# Logout button
st.divider()
if st.button("Log out"):
    st.session_state.logged_in = False
    st.session_state.username = ""
    st.info("You have been logged out.")
    st.switch_page("Home.py")
```

Key ideas in the dashboard

File failed to load: file:///Users/anna29/Desktop/Streamlit_part2_files/extensions/MathZoom.js **d in**:

```python
if not st.session_state.logged_in:
    st.error("You must be logged in...")
    st.switch_page("Home.py")
    st.stop()
```

- If logged in, we show charts and let them **log out** by resetting session_state and switching back to `Home.py`.

---

4. How this fits the Streamlit multipage model

You're using three core concepts from the official docs:

1. **Multipage apps** via main file + `pages/` directory ([docs.streamlit.io][1])
2. **Programmatic navigation** with `st.switch_page()` ([docs.streamlit.io][2])
3. **Session state** (`st.session_state`) to remember login info across pages (covered in the widgets/state documentation).

In [ ]:

File failed to load: file:///Users/anna29/Desktop/Streamlit_part2_files/extensions/MathZoom.js