

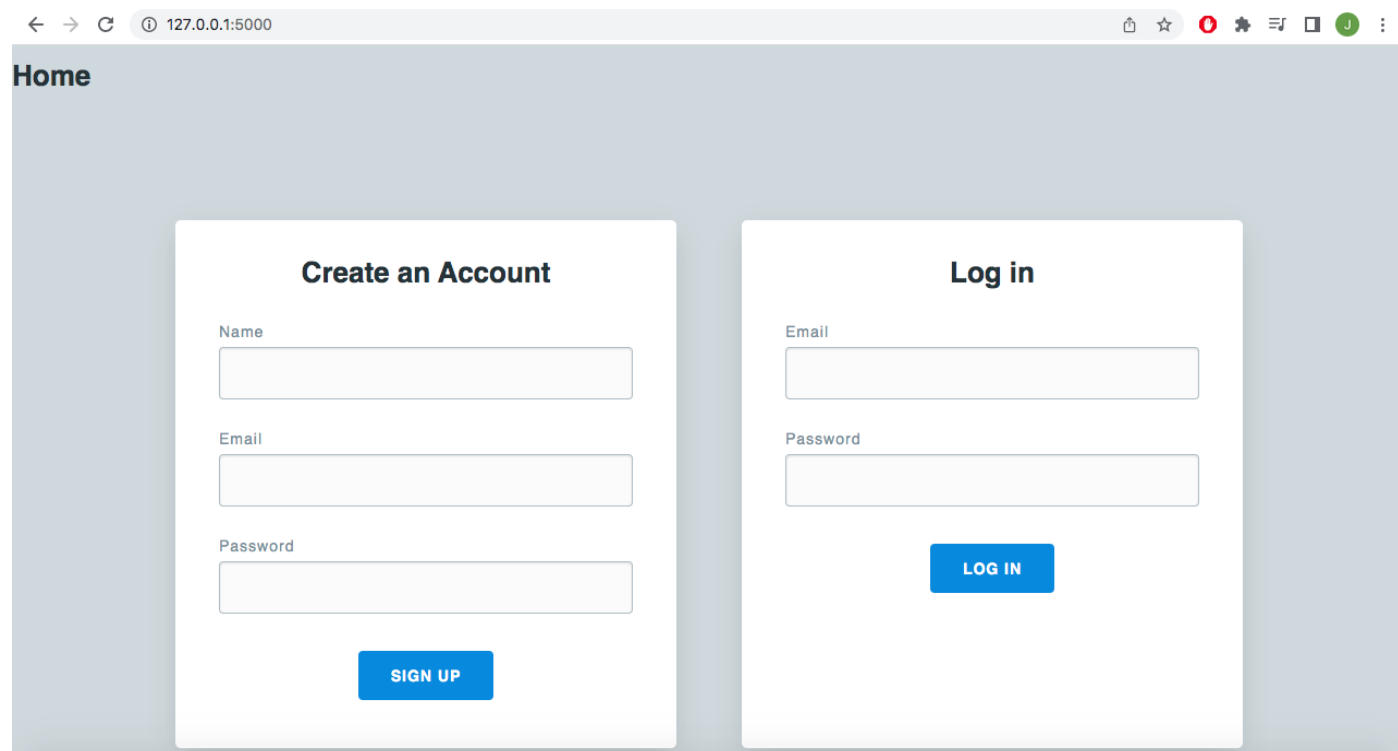
Login Project Using MongoDB and Flask

Description: In the first image we see the home page which features a signup form and log in form.

Users can create an account and their account is added to a database if the account doesn't already exist in the database. The database queries the account details based on email to determine if the account already exists; this occurs when signing up. The log in functionality queries for existing users based on email and uses additional logic to make sure the password matches the encrypted one in the database. If the passwords don't match, the user is met with red text stating an error. If the email doesn't exist the user is also met with an error. To prevent users from bypassing log in and manually entering the dashboard URL, a decorator for a login_required function has been added between the name of the dashboard route and the function for the dashboard route. This means that if a user manually types in the "/dashboard" URL they are redirected to the homepage since they haven't logged in. Users who are signed up are tracked using the MongoDB database, and their information is visualized in Studio 3T or, alternatively, MongoDB compass. After a single user signs in they can view their info on the dashboard page with their encrypted password. Finally, users can sign out and be redirected to the home page.

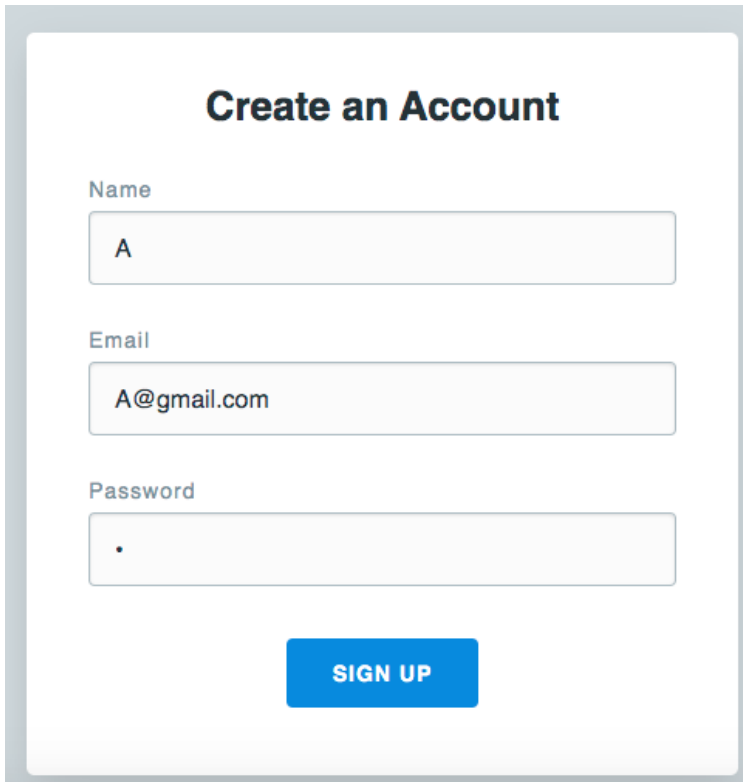
Link: https://github.com/NathanPhipps0/Login_Project

Image 1: Home Page: Sign Up and Login Page:



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000". The page has a light blue header with the word "Home" on the left. Below the header, there are two white rectangular forms side-by-side. The left form is titled "Create an Account" and contains three input fields labeled "Name", "Email", and "Password", followed by a blue "SIGN UP" button. The right form is titled "Log in" and contains two input fields labeled "Email" and "Password", followed by a blue "LOG IN" button.

Image 2: Creating an Example User to Display in Dashboard Image and Studio 3T, and MongoDB Compass:



A screenshot of a web form titled "Create an Account". The form is contained within a light gray border. It has three input fields: "Name" with the value "A", "Email" with the value "A@gmail.com", and "Password" with a single dot. Below the fields is a blue button labeled "SIGN UP".

Create an Account

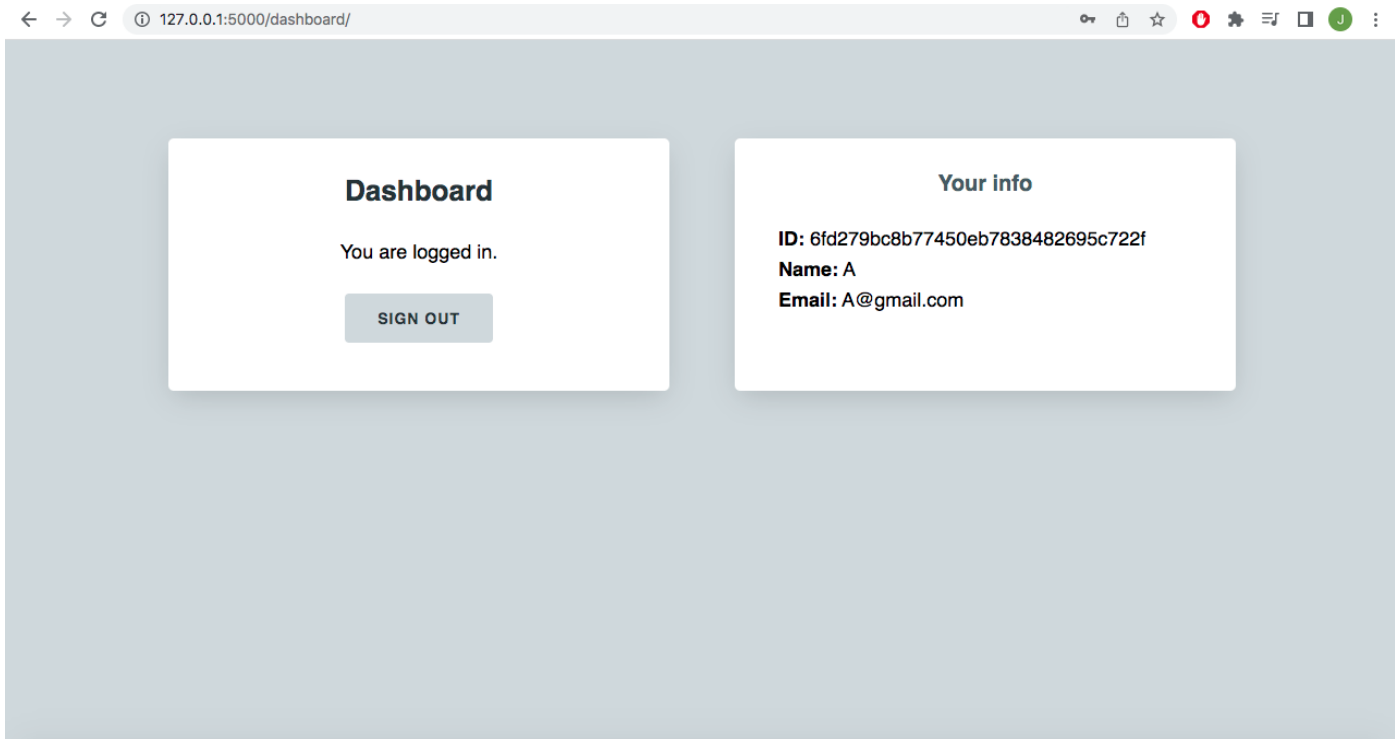
Name
A

Email
A@gmail.com

Password
•

SIGN UP

Image 3: Displaying Dashboard, Equipped with User Info, Encrypted Password, and Sign Out Functionality:



A screenshot of a web browser displaying a dashboard. The browser's address bar shows "127.0.0.1:5000/dashboard/". The dashboard has a light gray background and two white cards. The left card, titled "Dashboard", says "You are logged in." and has a "SIGN OUT" button. The right card, titled "Your info", displays user details: "ID: 6fd279bc8b77450eb7838482695c722f", "Name: A", and "Email: A@gmail.com".

← → ↻ ⓘ 127.0.0.1:5000/dashboard/ 🔑 📄 ☆ 🔴 ⚙️ 📄 📄 📄 📄

Dashboard

You are logged in.

SIGN OUT

Your info

ID: 6fd279bc8b77450eb7838482695c722f
Name: A
Email: A@gmail.com

Image 4: Displaying Invalid Login Credentials Error:

Home

Create an Account

Name

Email

Password

SIGN UP

Log in

Email

Password

LOG IN

Invalid login credentials

Image 5: Displaying Database of Signed Up Users using Studio 3T or MongoDB Compass:

Studio 3T is used to display users in the MongoDB database.

Id, name, email and an encrypted form of the password (using passlib.hash) are stored and viewable by establishing a connection to the database based on the localhost settings.

Studio 3T for MongoDB - Full product trial

Connect Collection IntelliShell SQL Aggregate Map-Reduce Compare Schema Reschema Tasks Export

You have 30 days remaining of your Studio 3T full product trial. You can find more information and manage your trial through the My License area in the Help menu.

Search Open Connections

localhost:27017 [direct]

- User_Login_System
 - Collections (1)
 - users
 - _id_
 - Views (0)
 - GridFS Buckets (0)
 - System (0)
 - admin
 - config
 - local

Quickstart Index Properties: _id_ users

(localhost:27017) > User_Login_System > users

Query Projection Sort Skip Limit

Result Query Code Explain

50 Documents 1 to 1

_id	name	email	password
17c7b82e0e62	A	A@gmail.com	\$pbkdf2-sha256

1 document selected

Count Documents 00:00:00.002

Image 6: MongoDB Compass: Storing Login Credentials using MongoDB, viewing with MongoDB Compass: Id, name, email and an encrypted form of the password (using passlib.hash) are stored and viewable by establishing a connection to the database based on the localhost settings.

