

PROJECT -
Secure Login System with User Role
Management.

Submitted by –
Rudrapratap Singh Bais

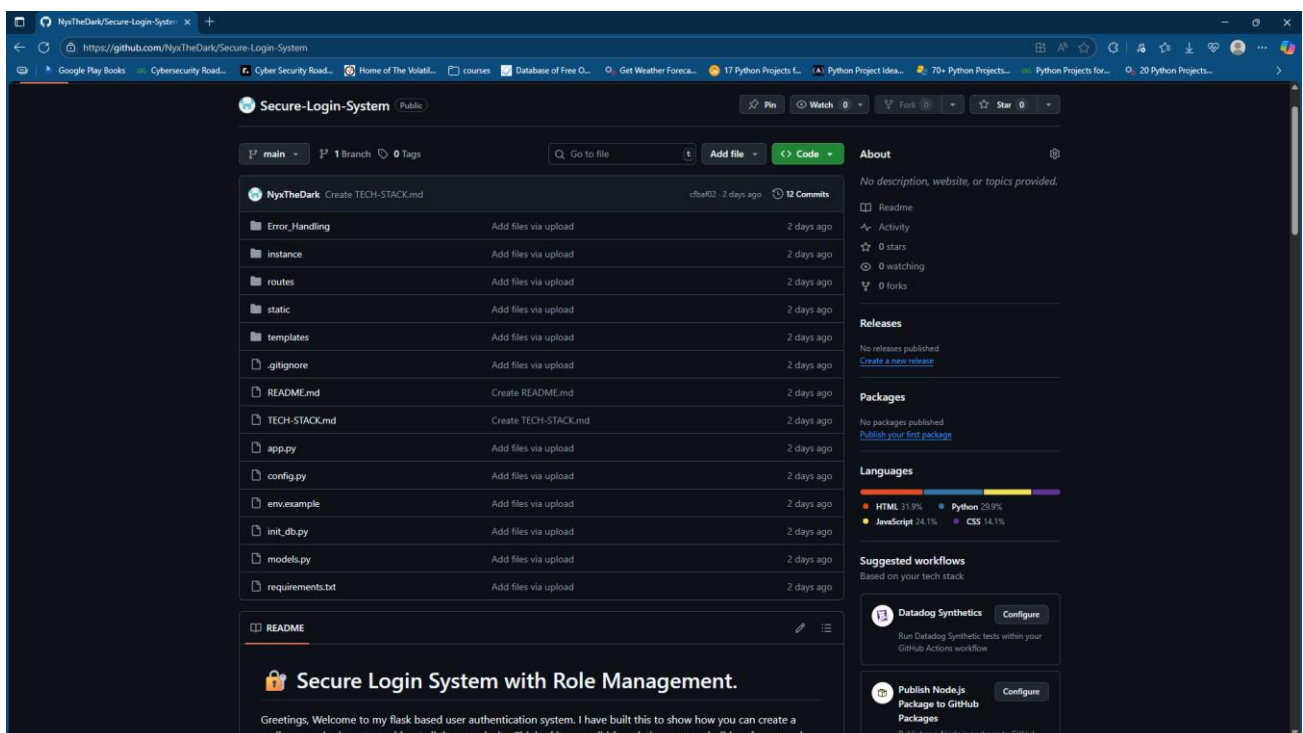
Submitted to –
Vault-Tec Security

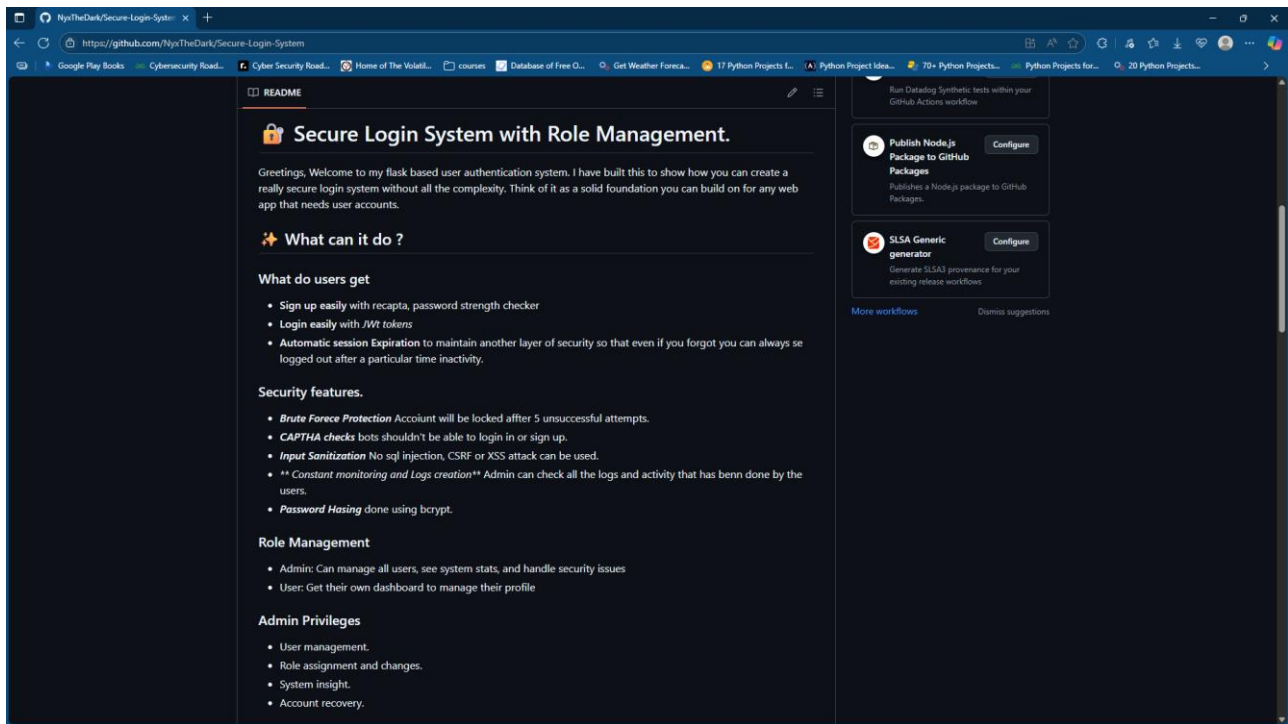
Introduction:

This project is comprised of a web-based secure login system that incorporate the user authentication and role-based access control.

The project is named as SecureAuth.

The project files can be visited at [NyxTheDark/Secure-Login-System](https://github.com/NyxTheDark/Secure-Login-System) and it can directly clones using HTTP at <https://github.com/NyxTheDark/Secure-Login-System.git> and using GitHub CLI at “ gh repo clone NyxTheDark/Secure-Login-System ”





Project Structure:

```
|— app.py                # Main application file
|— config.py            # Configuration settings
|— models.py           # Database models
|— requirements.txt     # Python dependencies
|— init_db.py          # Database initialization
|— HOW_TO_RUN.md       # This guide
|— routes/             # Route handlers
|   |— auth.py         # Authentication routes
|   |— admin.py        # Admin routes
|   |— user.py         # User routes
|— templates/          # HTML templates
|   |— base.html
|   |— index.html
|   |— login.html
|   |— register.html
|   |— dashboard.html
|— static/             # CSS, JS, images
|   |— css/style.css
|   |— js/
|— instance/           # Database files
|   |— user_auth.db
```

This is the basic structure of the project. Except for this the folder Error Handling contains the information error and bugs occurred during the making and there are files on how to solve them.

Starting The SecureAuth:

To start the login system, you can use any IDE by entering python app.py or run the file app.py directly or can use terminal (like command prompt).

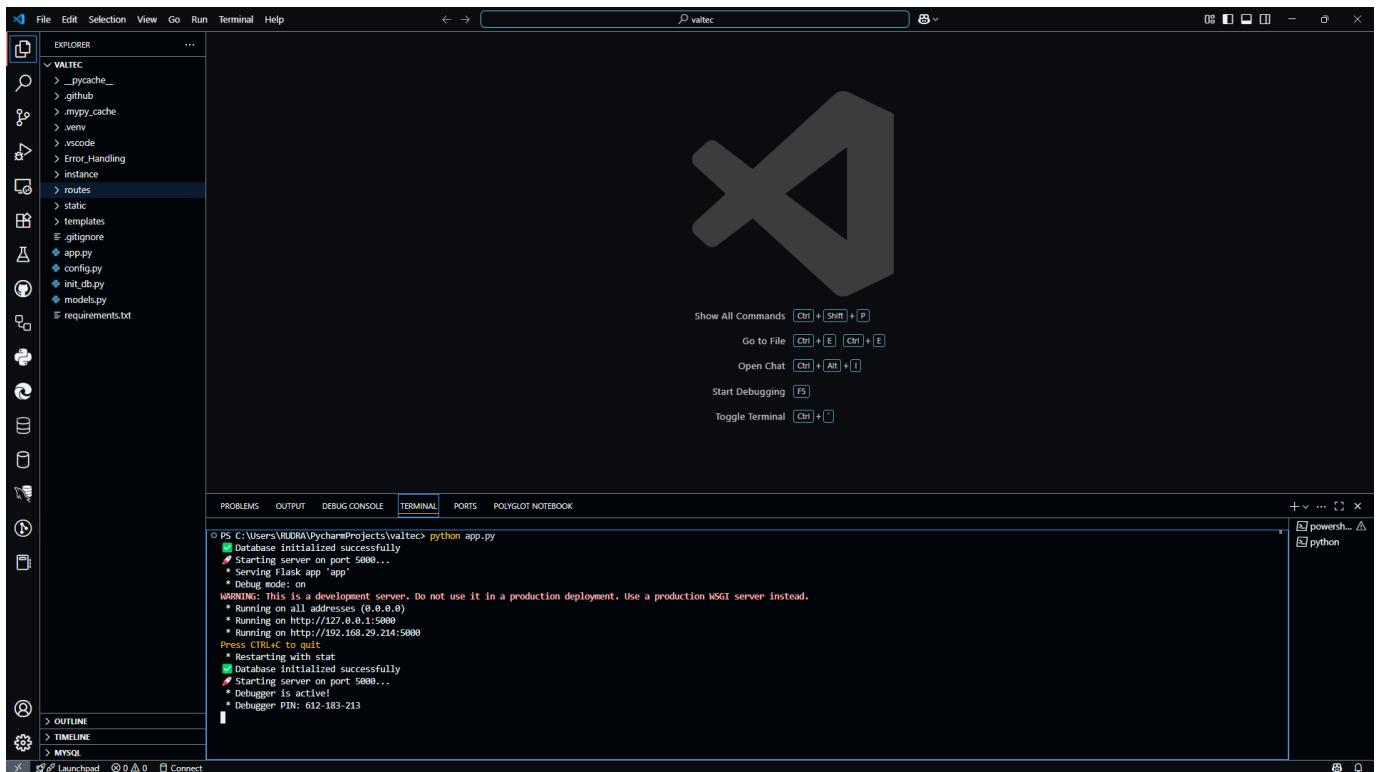


Figure 1 In case the file app.py is run in IDE then the terminal will provide output like the above image.

After the app.py is run the terminal will show the local host which can be used to access the SecureAuth. The system is host on the port 5000, for some reasons if the port is unavailable, the port can be manually changed to desired port no. from the file itself or through the terminal.

```
Command Prompt
C:\Users\RUDRA\PycharmProjects\valtec>python app.py
Database initialized successfully
Starting server on port 5000...
Serving Flask app 'app'
Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
Running on all addresses (0.0.0.0)
Running on http://127.0.0.1:5000
Running on http://192.168.29.214:5000
Press CTRL+C to quit
Restarting with stat
Database initialized successfully
Starting server on port 5000...
Debugger is active!
Debugger PIN: 602-247-801
192.168.29.214 - - [25/Jul/2025 15:35:22] "GET / HTTP/1.1" 200 -
192.168.29.214 - - [25/Jul/2025 15:35:22] "GET /static/css/style.css HTTP/1.1" 200 -
192.168.29.214 - - [25/Jul/2025 15:35:22] "GET /static/js/main.js HTTP/1.1" 200 -
Admin auth error: Signature has expired
192.168.29.214 - - [25/Jul/2025 15:35:22] "GET /admin/stats HTTP/1.1" 401 -
192.168.29.214 - - [25/Jul/2025 15:35:22] "GET /favicon.ico HTTP/1.1" 404 -
192.168.29.214 - - [25/Jul/2025 15:35:23] "GET /login HTTP/1.1" 200 -
192.168.29.214 - - [25/Jul/2025 15:35:23] "GET /static/css/style.css HTTP/1.1" 304 -
192.168.29.214 - - [25/Jul/2025 15:35:23] "GET /static/js/main.js HTTP/1.1" 304 -
Login successful for user: user, role: User
JWT token created with sub: 2, role: User, username: user
192.168.29.214 - - [25/Jul/2025 15:35:40] "POST /auth/login HTTP/1.1" 200 -
192.168.29.214 - - [25/Jul/2025 15:35:42] "GET /dashboard HTTP/1.1" 200 -
192.168.29.214 - - [25/Jul/2025 15:35:42] "GET /static/css/style.css HTTP/1.1" 304 -
192.168.29.214 - - [25/Jul/2025 15:35:42] "GET /static/js/main.js HTTP/1.1" 304 -
192.168.29.214 - - [25/Jul/2025 15:35:42] "GET /static/js/dashboard.js HTTP/1.1" 200 -
C:\Users\RUDRA\PycharmProjects\valtec>
```

Figure 2Running the program through command prompt.

The Home page gives visitors two basic options for sing-in and sign-up. It also highlights the key features of the SecureAuth.

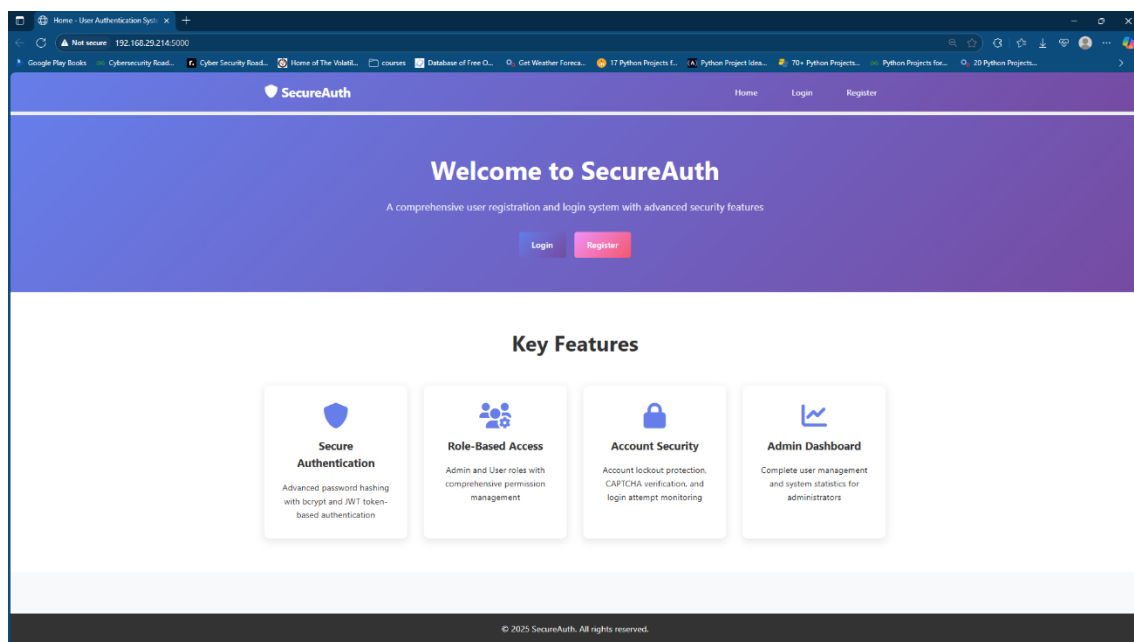


Figure 3Home page of SecureAuth (Secure Login System).

The Home page gives visitors two basic options for sing-in and sign-up. It also highlights the key features of the SecureAuth.

Registration Page:

User Registration Page contains the username, email id, password, CAPTHA and terms and condition box.

The screenshot shows a web browser window with the URL `192.168.29.214:5000/register`. The page is titled "SecureAuth" and has a navigation bar with "Home", "Dashboard", and "Logout" links. The main content area is a "Create your account" form. The form includes the following fields and validation messages:

- Username:** A text input field with the placeholder "Enter your username".
- Email Address:** A text input field with the placeholder "Enter your email".
- Password:** A text input field with the placeholder "Enter your password". Below it, a list of requirements is shown: "At least 8 characters", "One uppercase letter", "One lowercase letter", "One number", and "One special character".
- Confirm Password:** A text input field with the placeholder "Confirm your password".
- CAPTCHA Verification:** A CAPTCHA image showing the text "YNHbpx" and a text input field with the placeholder "Enter CAPTCHA".
- Terms and Conditions:** A checkbox labeled "I agree to the Terms of Service and Privacy Policy".
- Create Account:** A purple button with a user icon and the text "Create Account".
- Footer:** A link "Already have an account? Login here".

Each and every field has in validation built-in that make the page input sanitized so that attacks injection attacks cannot be used.

The sign-in cannot be done until all the security requirements are matched like password strength, captcha, and terms and condition, any remaining field will be highlighted and text will be prompted to ask user to fill it correctly.

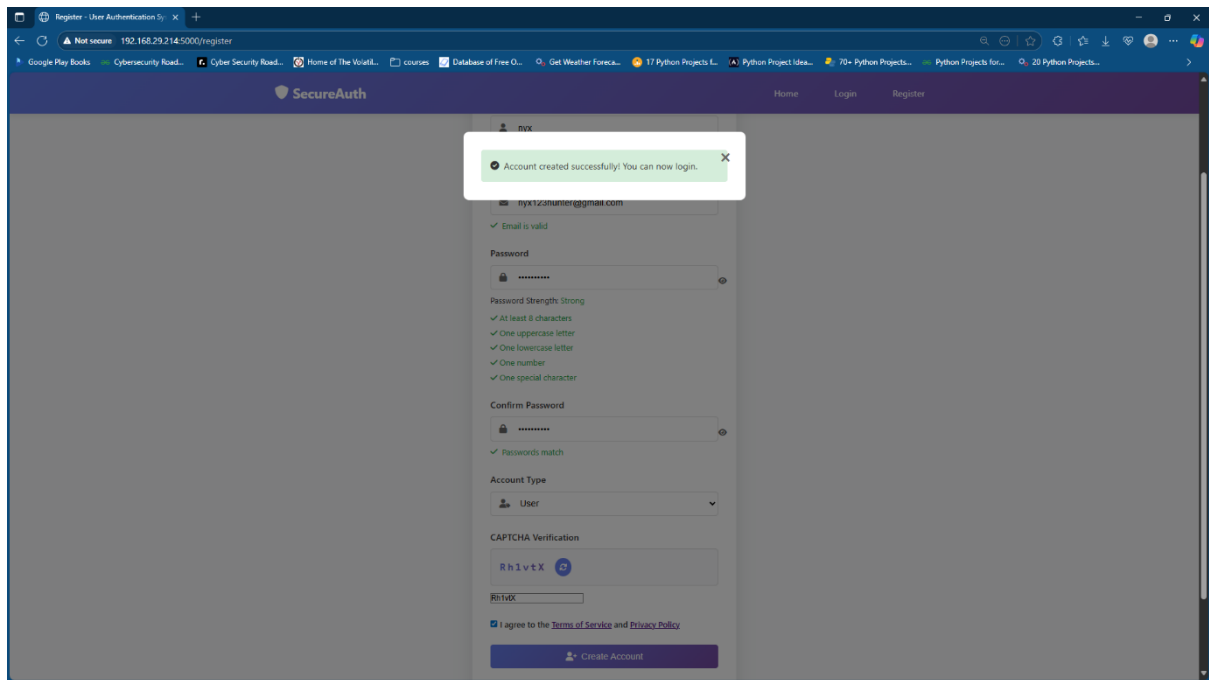
The screenshot shows the same "Create your account" form as the previous image, but with validation messages. The form is now filled with the following data:

- Username:** "nyx" (highlighted in blue).
- Email Address:** "nyx123hunter@gmail.com" (highlighted in blue).
- Password:** "*****" (masked).
- Confirm Password:** "*****" (masked).
- CAPTCHA Verification:** The CAPTCHA image shows "YNHbpx" and the text input field contains "Enter CAPTCHA".

Validation messages are displayed:

- Next to the username: "✓ Username is valid".
- Next to the email address: "✓ Email is valid".
- Below the password field: "Password Strengths Weak".
- Below the confirm password field: "✗ Passwords do not match".
- Below the CAPTCHA field: "Please fill out this field." (highlighted in orange).

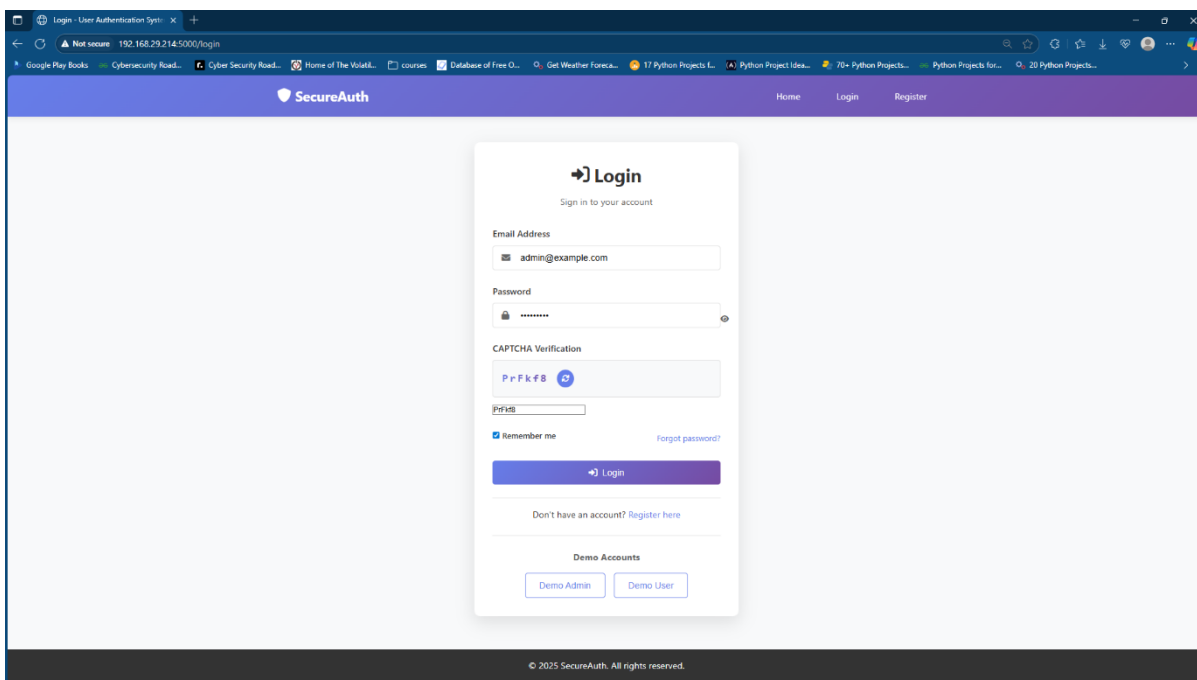
If all the criteria are fulfilled and if the user has registered successfully then a message will pop up at the centre top of the page.



Login Page:

The login page is simple and implemented with the features like input sanitization, Captcha. So that user can login securely.

Currently demo tabs had been added so that users can see the functionality of the site. These tabs can be disabled later.

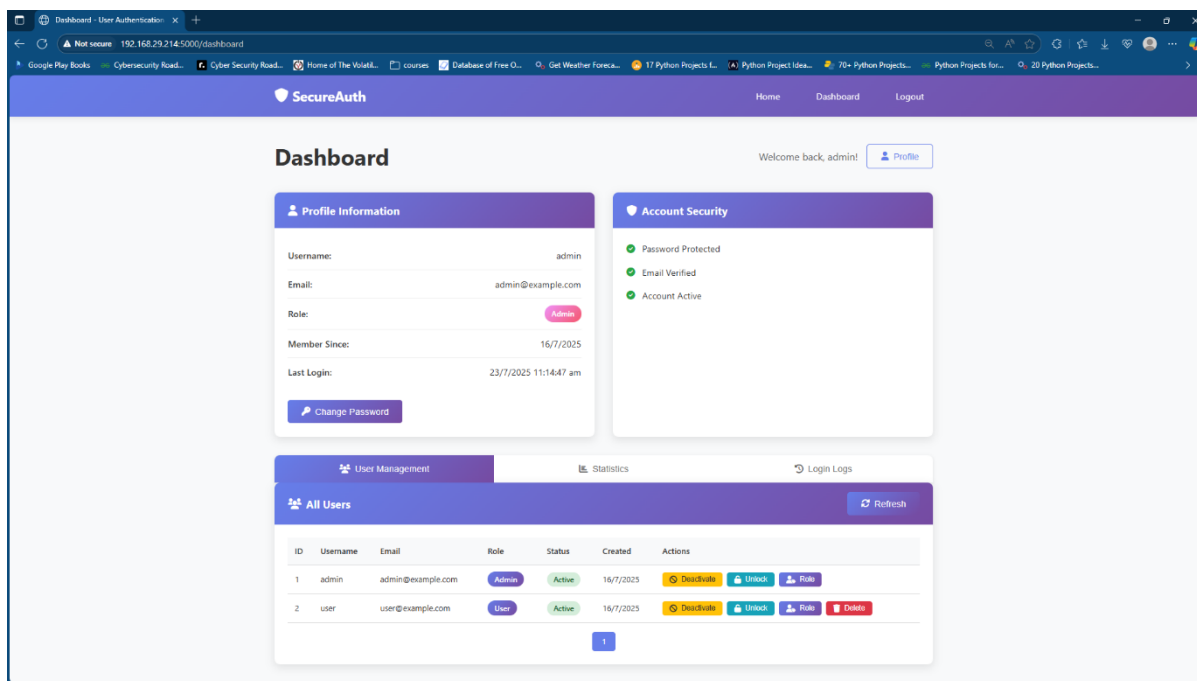


Admin Dashboard:

It showcases the admin profile with security features that are currently working and also shows the list of users.

The admin has the Privilege's to Change user role, lock and unlock their account, deactivate or delete their account.

Except this Admin also has the privileges of accessing the login logs, password change request (The passwords are not visible).



In case the user forgets the password of his account, the account can be reset by the IT department through the files provided in the folder Error Handling.

Troubleshooting:

Issues like

- port already in use
- database errors
- missing dependencies

- admin error or logout
- checking of databases content

are mentioned in the file [Secure-Login-System/HOW TO RUN.md at main · NyxTheDark/Secure-Login-System](#)

Apart from this any kind of debugging and test can be done with the modules provided in the Folder Error handling.

The **difference.txt** can be used to know which test file is used for which test or aspect of test, and **COMPLETE_Documentation.md** can be used for summary of overall project from installation to error handling and test summary and their ideal outputs.