# Initial Examination

* Can access admin page without logging in
* Can access the profile page without logging in
* Can create lottery draws without logging in
* Can submit winning draws without logging in
* Passwords are stored as plain text inside of the database
* No input validation on any fields, including register, and lottery fields, only type validation
* Cannot login or logout
* Method not allowed error when trying to access logs and then clicking login
* Can view all users when not logged in
* No server-side validation
* Logged in as admin by default
* Database has no role-based access control

After examining the provided project, I have found many security issues. I can access the admin page without logging in. I believe this is because it thinks I am logged in as the admin, as when I submit a draw, then make that draw the winner using the admin page, I can see the winning draw belongs to the admin.

I can also access the profile, and account pages, without being logged in, as well as submit draws and view all registered users. Although, this may be related to the fast it thinks I am logged in as the admin.

There is very little input validation on fields. The only input validation I was able to find was on the draw inputs, where it limits you to use only numbers, however, this can easily be overcome by editing the html of the webpage, which tells me there is no server-side input validation either.

A lot of features appear to be broken, and show no error message, so I could be using them incorrectly, and I would not know. An example of this would be the logs button in the admin menu. I also get a ‘Method Not Allowed’ error message when I click on the logs button, and then try to navigate to a broken page, such as the Login page.

The database has passwords stored as plain text inside, which is a huge problem, as if the database is compromised, the attackers will have easy access to them. Hashing the passwords would make the database a lot more secure in the scenario of a data breach, as they would have to De-hash the passwords, which is very difficult, and mostly involves guessing.

Text

Description automatically generated

The database does not have role -based access control, however, this is a limitation of SQLite, and should not be too much of a problem, as the python flask application should be the only thing accessing the database on behalf of the users using it.

There is very little to no error handling. An example of this is the 404 page not found error, and the ‘Method Not Allowed’ error. These take me to the default error page, with no way to go back without using the back button in the browser.