**“Spookie”: project documentation** Ronny Paz 313651416, Liron Glickman 203092630

**Introduction**

Our goal is to fake our identity using cookies that were obtained with brute force methods instead of actually getting the cookie of someone else’s session.

We will register a username and use the cookie given to us by the website to guess another cookie that identifies a different user.

**Plan**

We will create and host a simple login page with a *MySQL* database. The database will have a “company” table that holds the credentials of all registered users (username, password and id). Each registered user will enter his username and password in the main page and will be assigned a PHP session ID upon logging in (assuming they are on the database). That session information will be stored on the client as a cookie. The session expiration is dependent on the cookie expiration, meaning, the session on the server will stay active and the user can enter his profile directly as long as his cookie was not destroyed. The default expiration date for the cookie is set to *0,* meaning that it will be destroyed when the browser is closed – so an active session has to be open on a browsing while the user is logged in.

We will then use our tool to pull the initial cookie and use it as a starting point for a brute force attack – the goal is to find a session cookie that was given to a different user and use it to enter his profile page, without knowing his *username* or *password*.

**Built in functions**

There are several built in PHP functions that should be taken under consideration: [\* = session]

*\*.auto\_start* – starts a session automatically, before a login attempt was made.

*\*.use\_strict\_mode* – ignore uninitialized or deprecated session ID.

*\*.use\_cookies* – toggles whether our sessions will be stored as cookies on the client.

*\*.use\_only\_cookies* – the session will rely on client-side cookies exclusively.

*\*.cookies\_path*– controls whether the cookies is relevant to certain parts of the site or not.

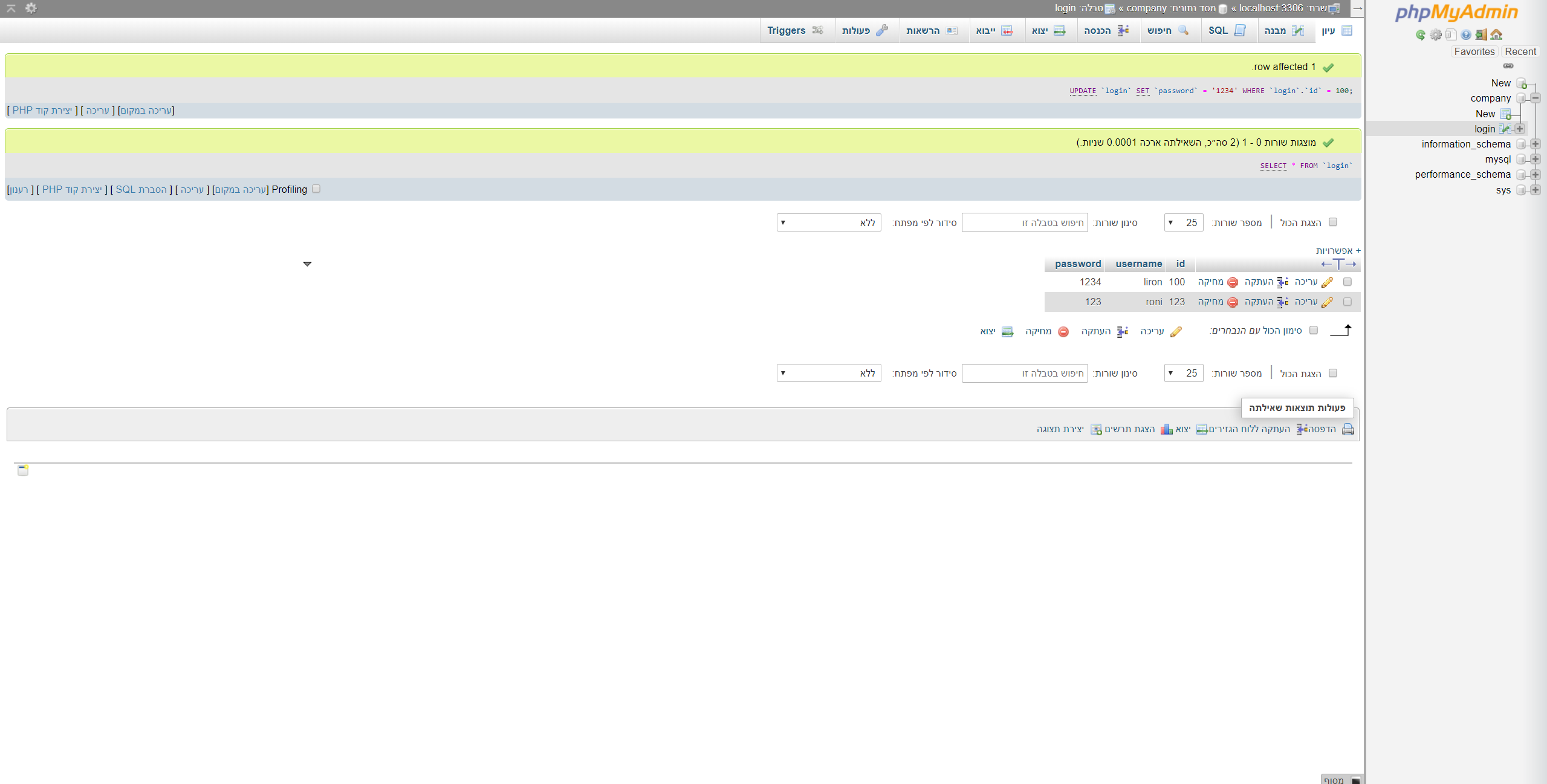
*\*.cookie\_lifetime* – determines the lifetime of the cookie.

\*.sid\_bits\_per\_character – limits the characters used in the session cookie string

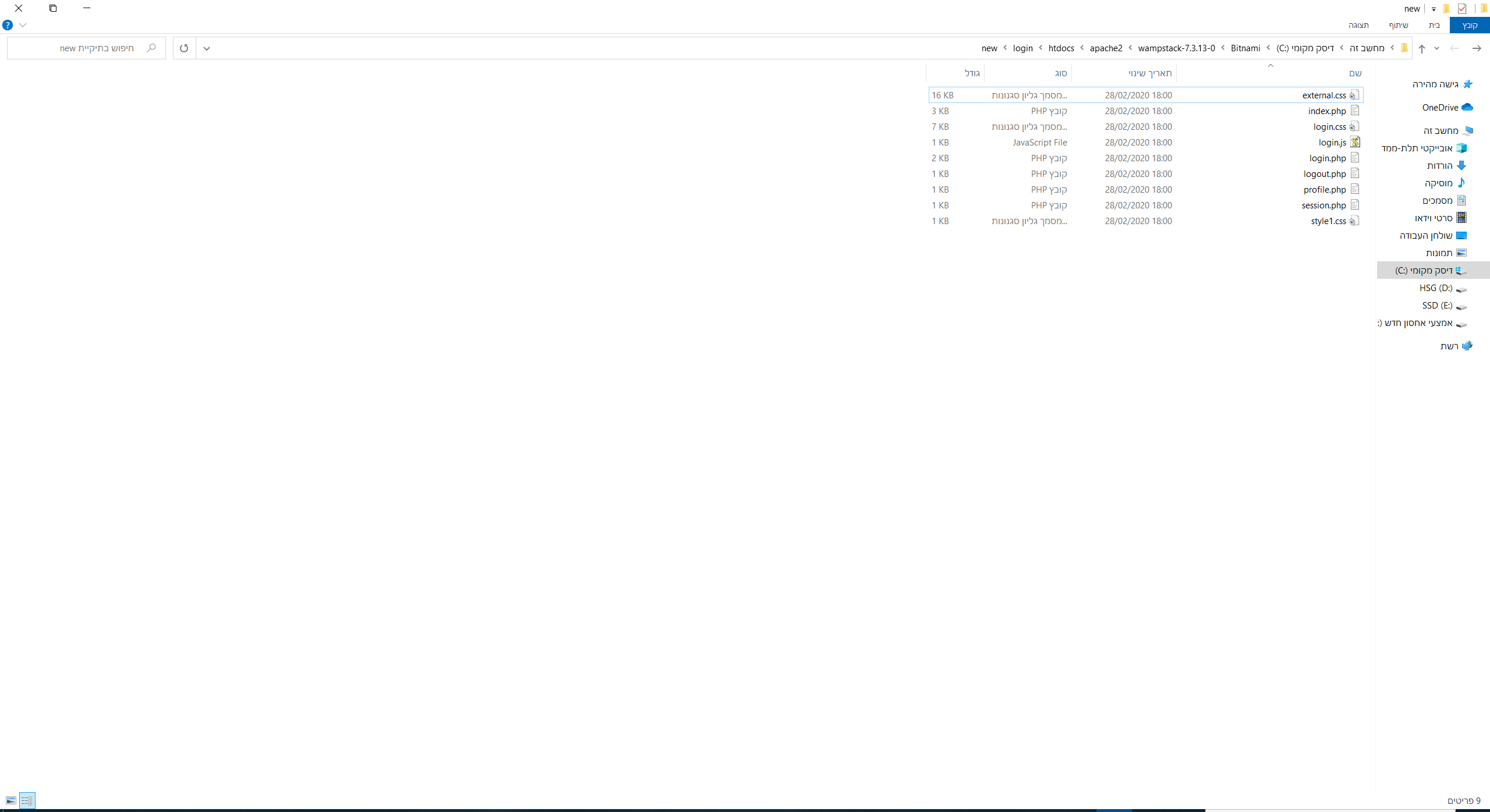
*\*\_start* – creation of a session.

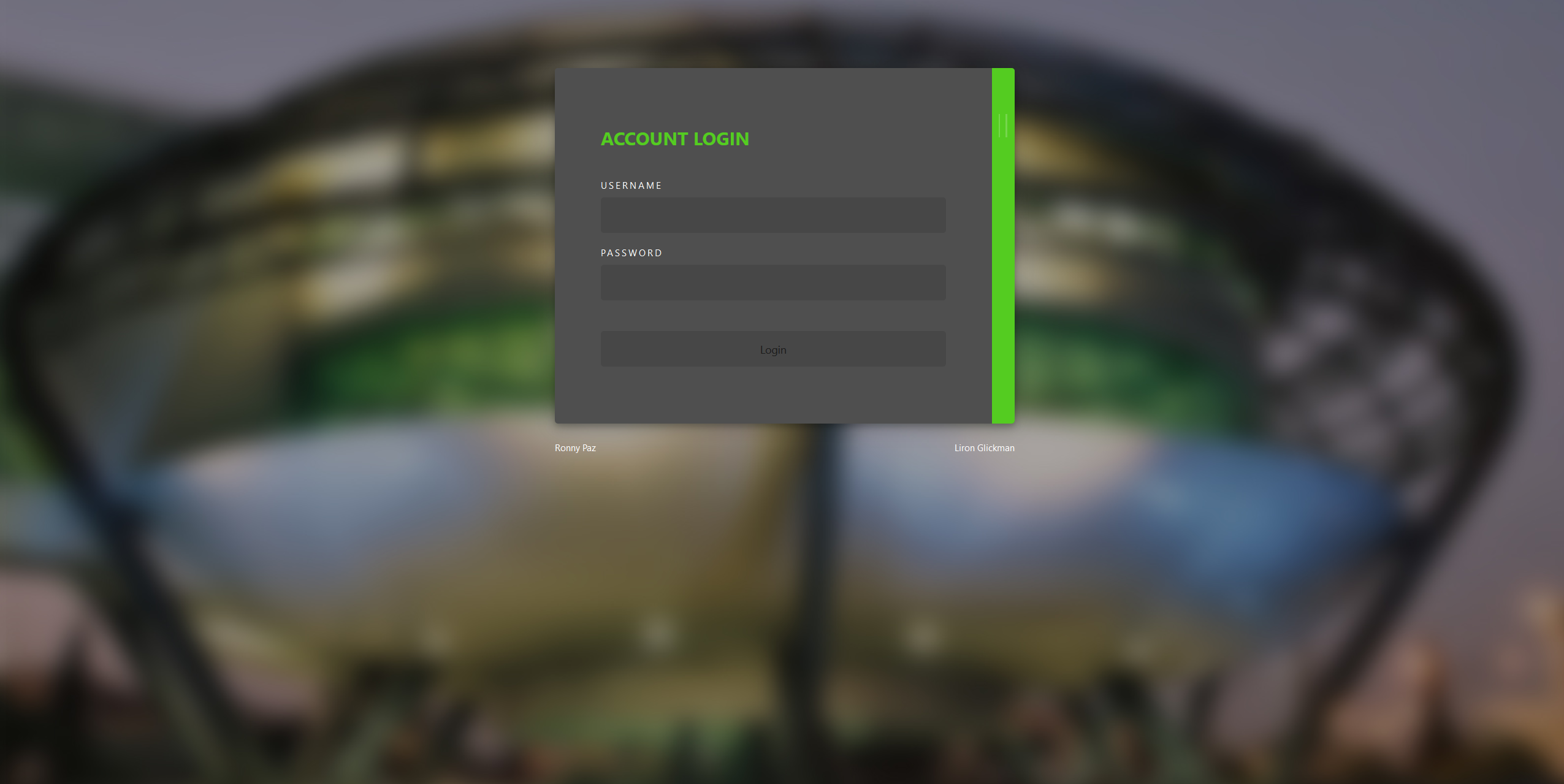
Example of usage:

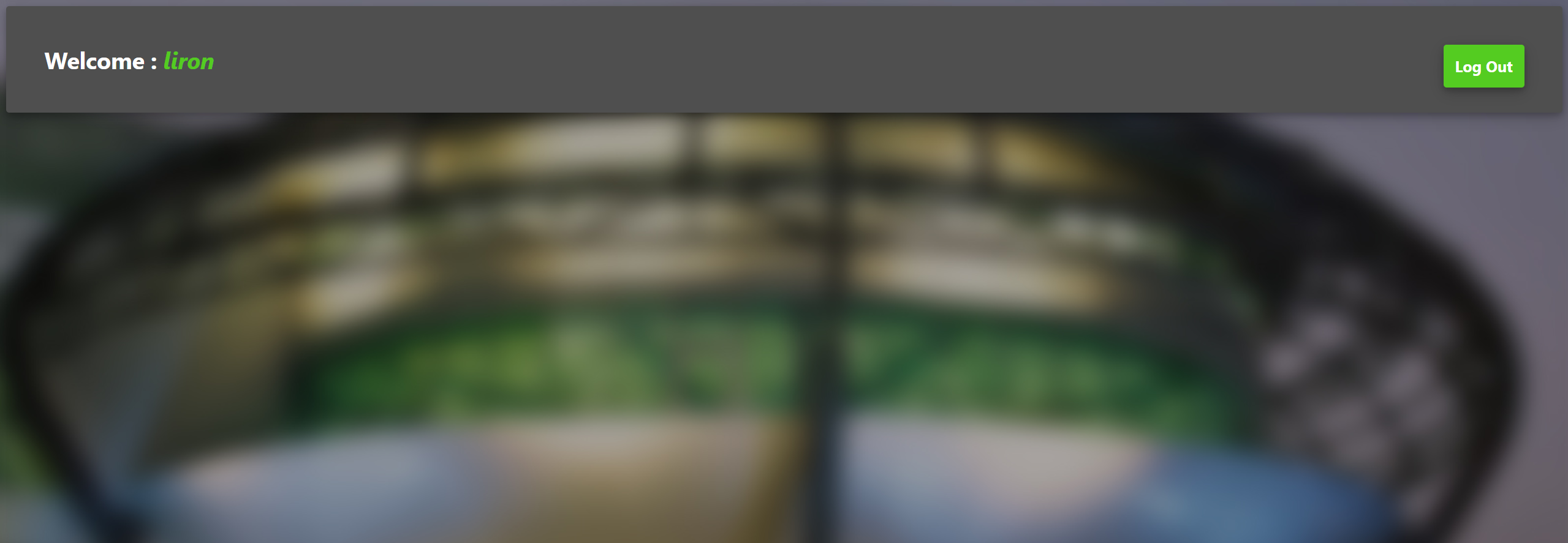
1) We install *Bitnami* services and make a database to manage the server side and store all the user data:



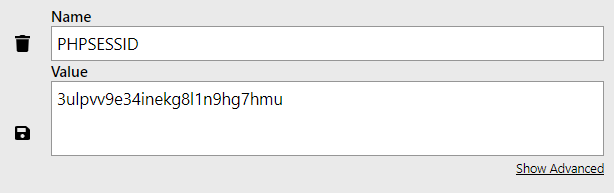
2) We set all the files for the system that we want to attack:



3) This is the example of the attacked system: 

4) This is when we login: 

5) This is the cookie of the system:

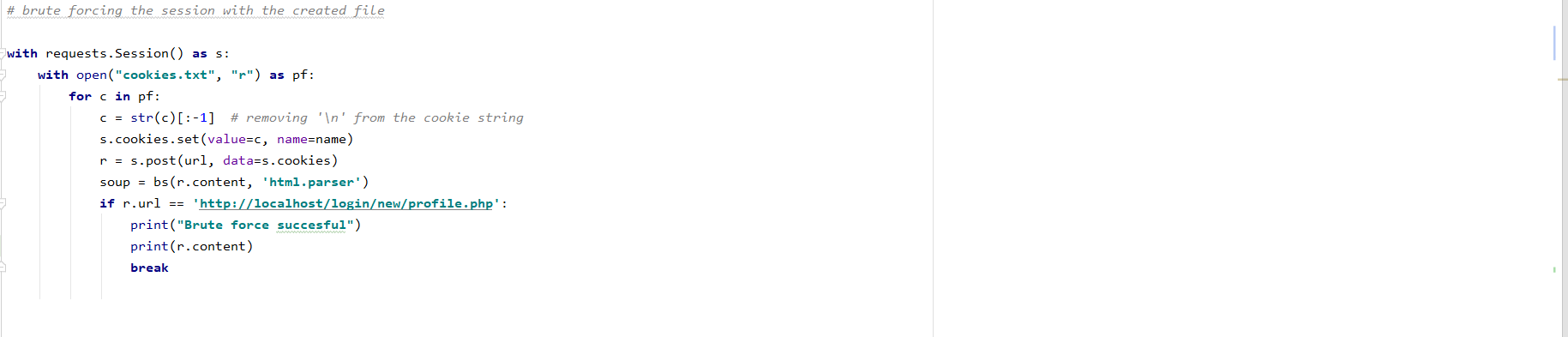


\*We know that the name is *PHPSESSID* and the string size is 26 characters combining integers (0-9) and letters (a-v).

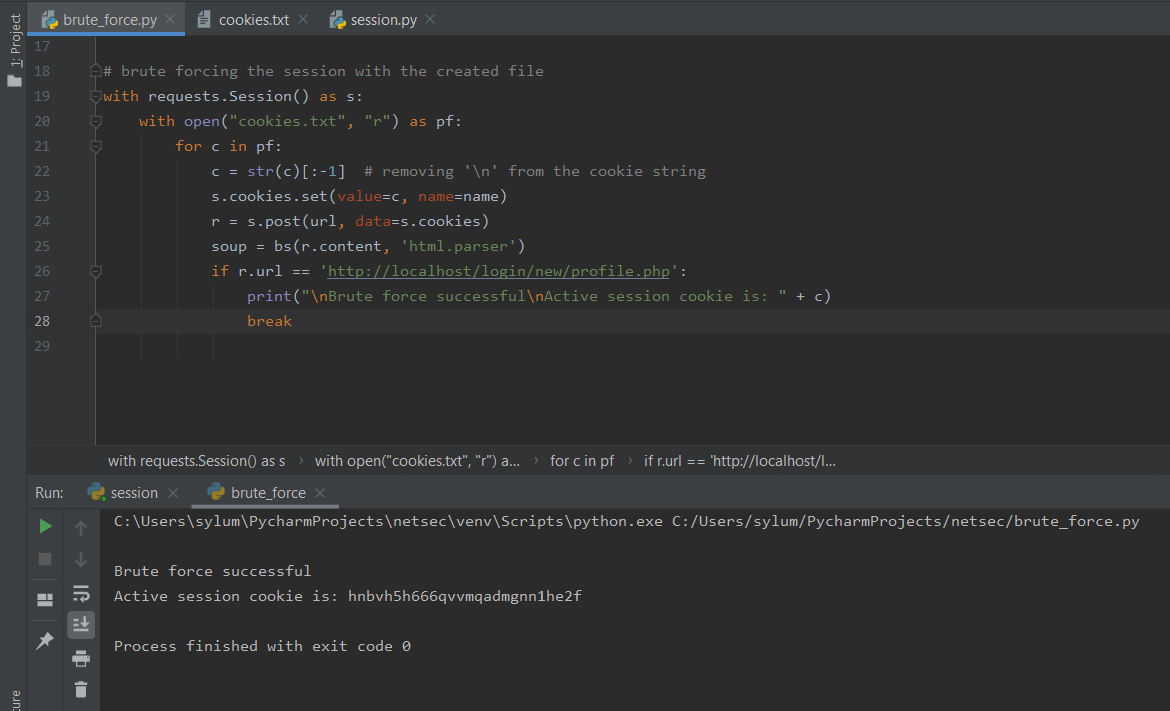
6) With the brute force tool we’ve made in python, we create a txt file with 32^26 different string values ([10 integers + 22 letters]^26 characters).



7) We use said file to try to enter an active session:



8) After several iterations, we find a string representing an active session:



**Libraries**

requests – used to send requests to the server and get responses from it

beautifulsoup4 – website information linting for scraping

strgen – A random string generator

**Tools**

Windows 10 – main workstation

WAMP stack – local website hosting

PHP – server-side scripting

MySQL – user database

Google chrome – browsing and presentation

Python 3.8 and PyCharm – tool development