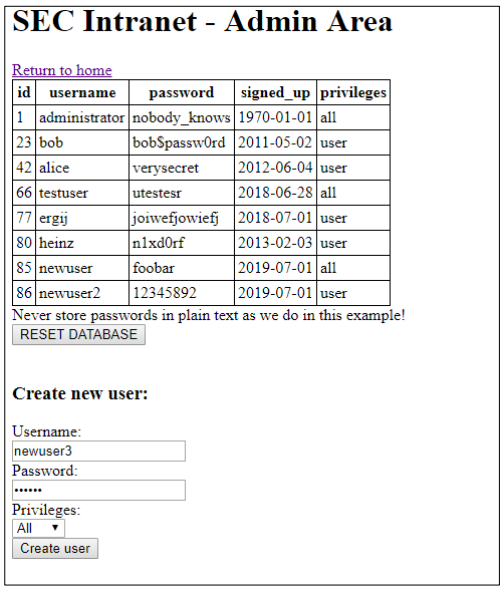
# System and Web Security – Homework 10

## Problem 1: SQL Injections, CSRF

### a) Adding new users

A form to add new users was added to the provided web application. The form is very basic and consists of fields to enter the new user's name and password, as well as a drop-down menu to select the new user's privilege level. The new GUI can be seen on the following screenshot:




Please refer to the source file `sec-intranet-upgraded.py` for implementation details.

Please also note that the database's structure has been changed slightly: instead of writing logic for creating the user ids manually, the database was altered to make the id column the user table's primary key, and the column set to auto-increment, which is a common pattern.

This has the distinct advantage of ids being unique in that table, as well as the database automatically providing new ids once a new user is created.

Please refer to the file `users-upgraded.sql` for the updated user table's structure.

### b) Protecting the SQL queries

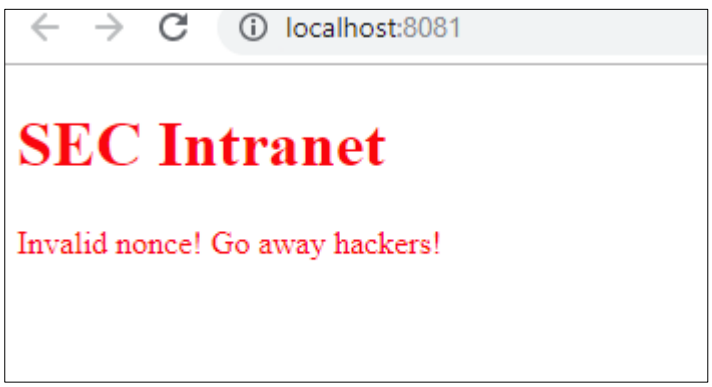
The SQL queries were changed to parameterized queries, please refer to `sec-intranet-upgraded.py` for the implementation.

### c) CSRF attack

The file `csrf-exploit.html` contains an attack that logs visitors to that page into the SEC intranet application as the user "heinz". This is done via a manipulated hidden form that is triggered by JavaScript, which causes a forged log in request from the user's browser.

### d) CSRF defense

A session-dependent nonce approach was implemented, which rejects requests with an invalid nonce in the form with the following error message:




Please again refer to the source file `sec-intranet-upgraded.py` for implementation details.