****

实验报告

|  |  |
| --- | --- |
| 姓名： | 强珂阳 |
| 学号： | 57118106 |

东南大学网络空间安全学院

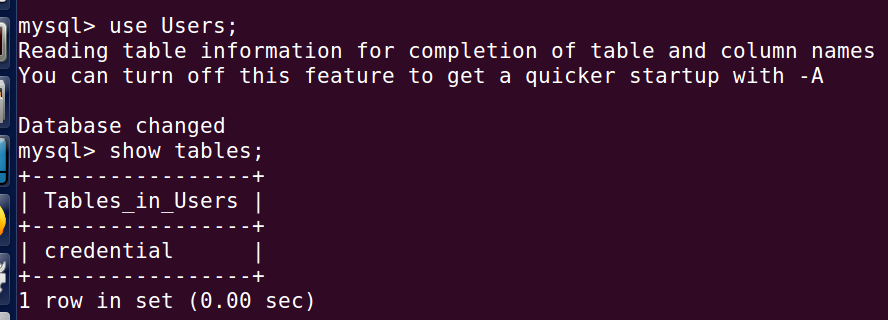
School of Cyber Science & Engineering

Southeast University

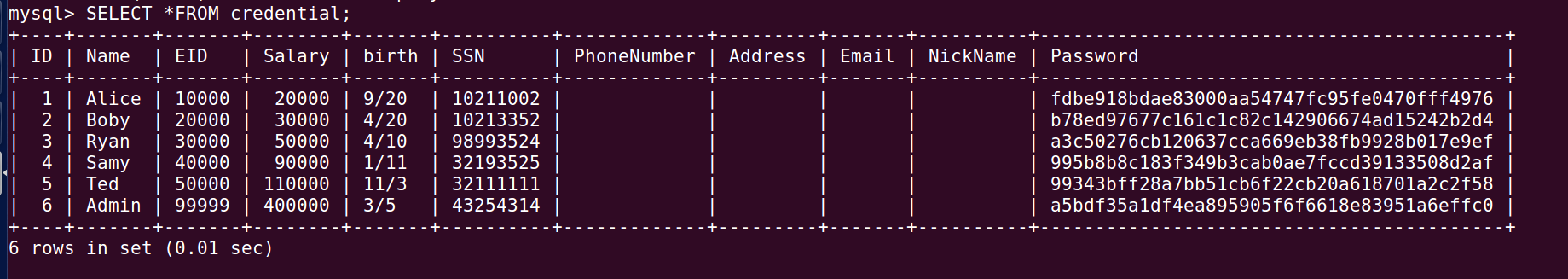
2020年9月

TASK1：**Get Familiar with SQL Statements**

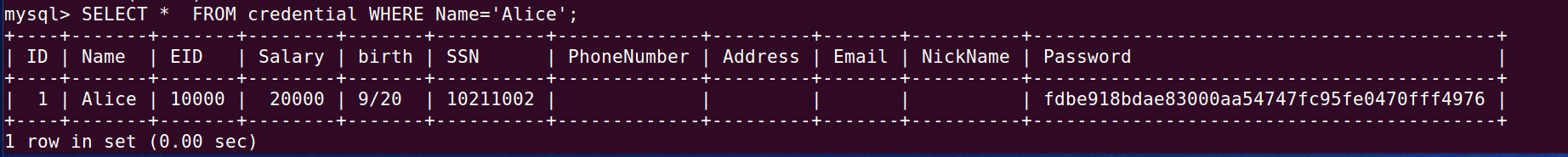
首先登录到MySQL控制台，加载数据库Users，显示Users中的表格：



再打印表格credential中的内容：



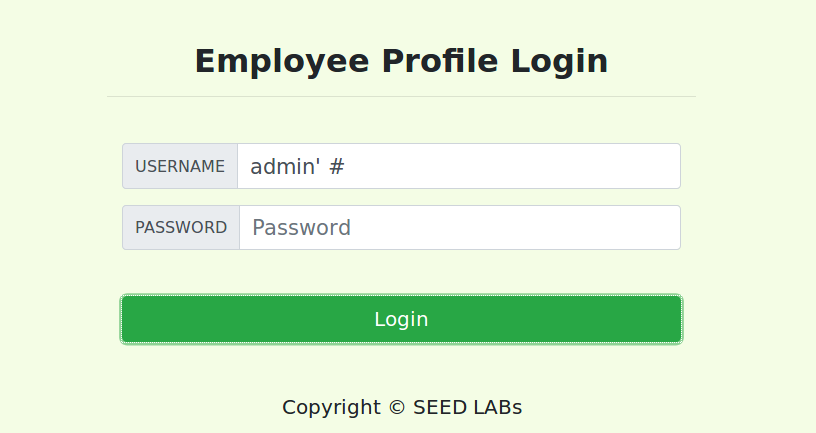
筛选出Alice的信息，可以通过WHERE语句来进行：

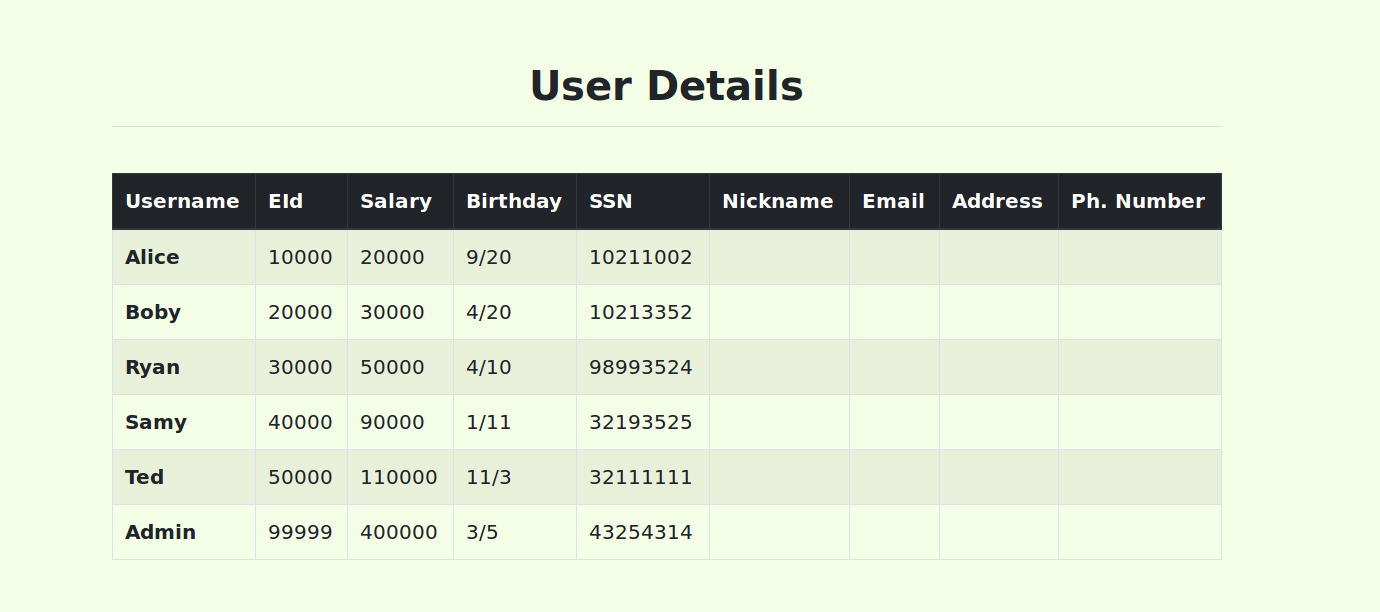


TASK2：SQL Injection Attack on SELECT Statement

**2.1 SQL Injection Attack from webpage**

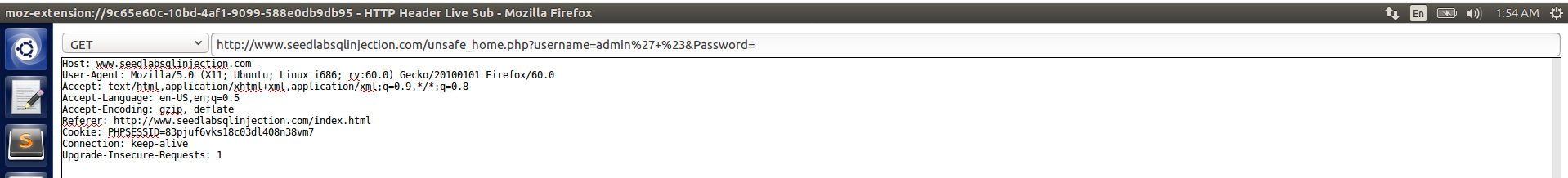
在username中输入admin’# 就可以注释掉验证代码中的and Password=’$hashed\_pwd’这样就只需要输入已知的admin用户名即可以登录：





**2.2 SQL Injection Attack from command line**

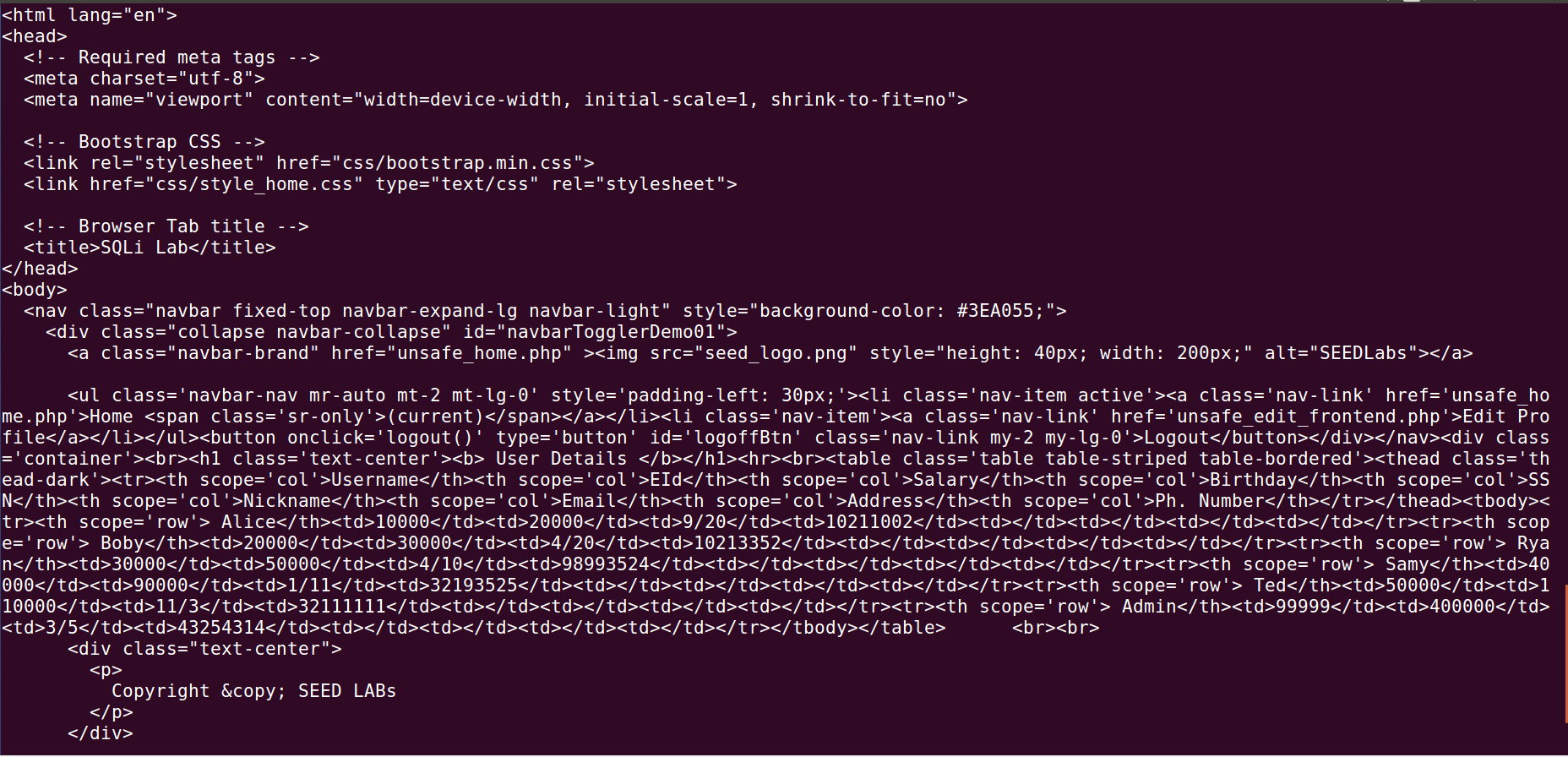
在用2.1的方法登录的同时，采用HTTP Header Live工具获取请求：



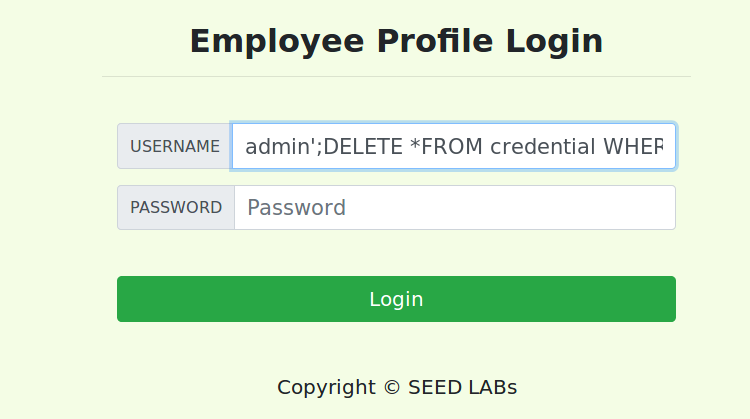
由此可以得知curl发送请求的内容：

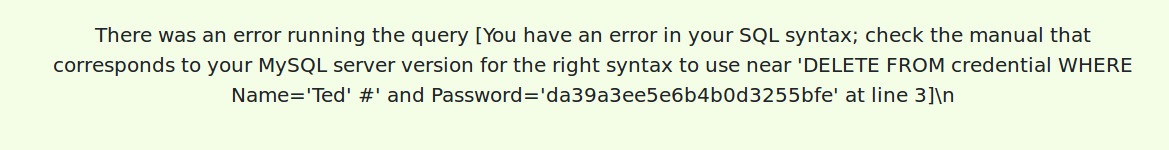


得到admin用户的权限：



**2.3 Append a new SQL statement**



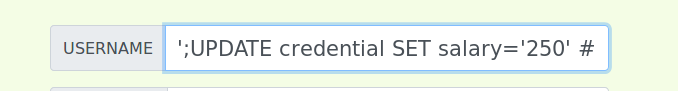


可以发现无法在其中插入两个SQL语句并成功执行。

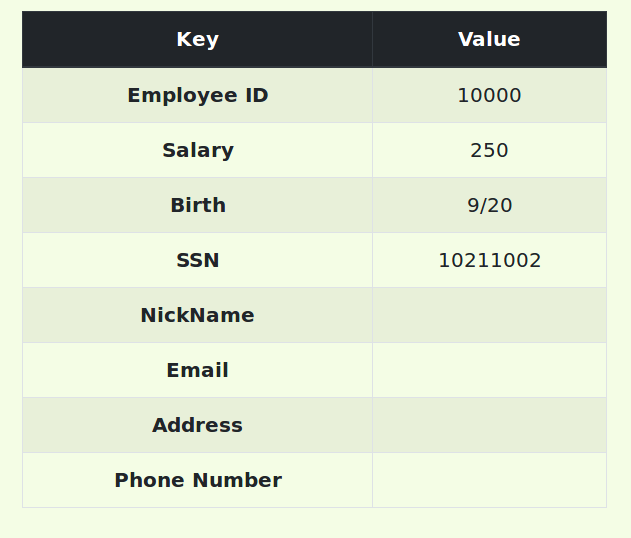
修改/var/www/SQLInjection中的unsafe\_home.php文件：



在登陆界面输入两条语句：



再将unsafe\_home.php修改回去后，登录Alice的账号：

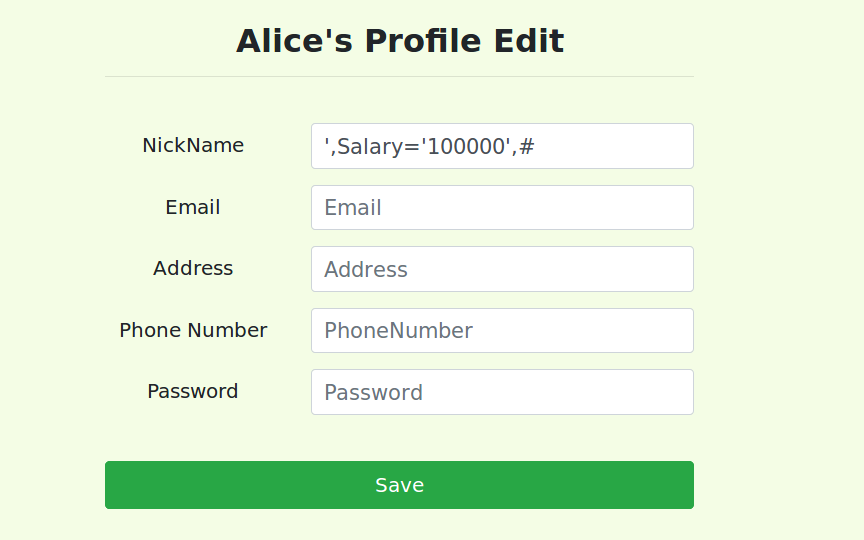


执行成功。

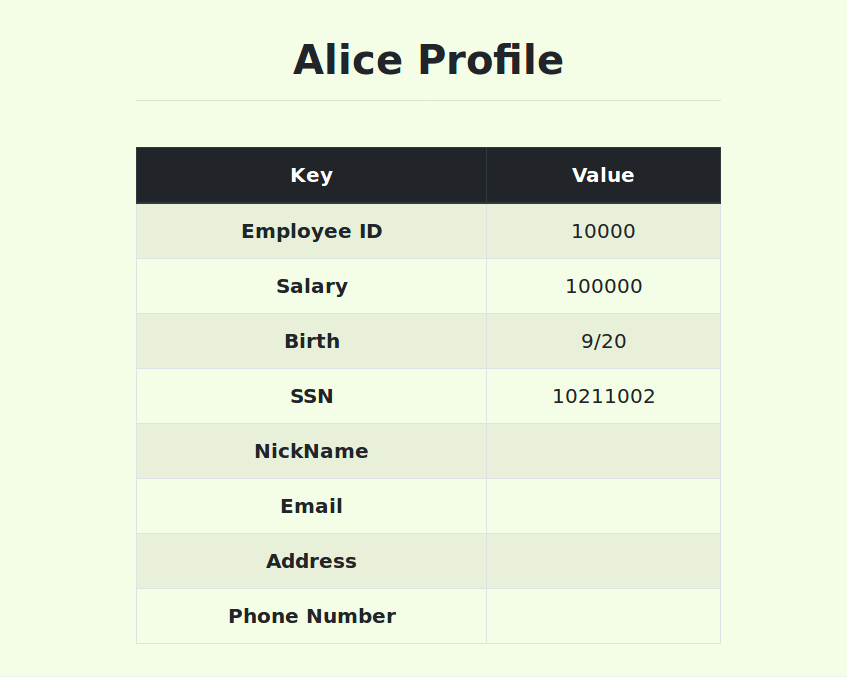
TASK3：**SQL Injection Attack on UPDATE Statement**

**3.1 Modify your own salary**

在Alice的帐户中，修改个人资料：

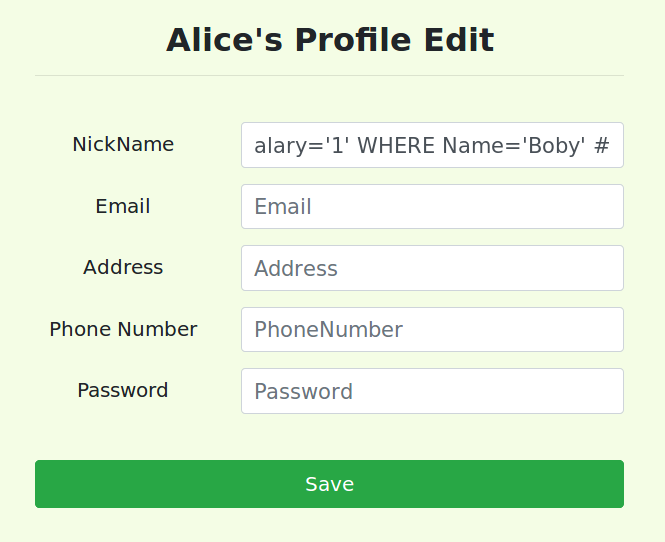


保存后发现Alice的工资已经被修改：

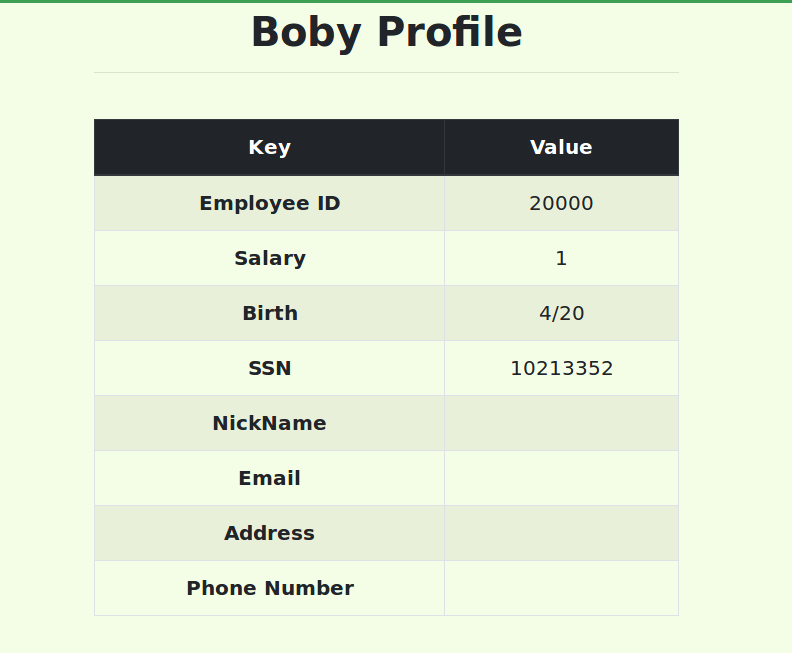


**3.2 Modify other people’salary**

运用上述手法，修改Alice的个人资料：

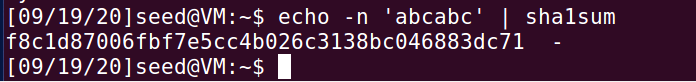


保存后登录Boby的账号：



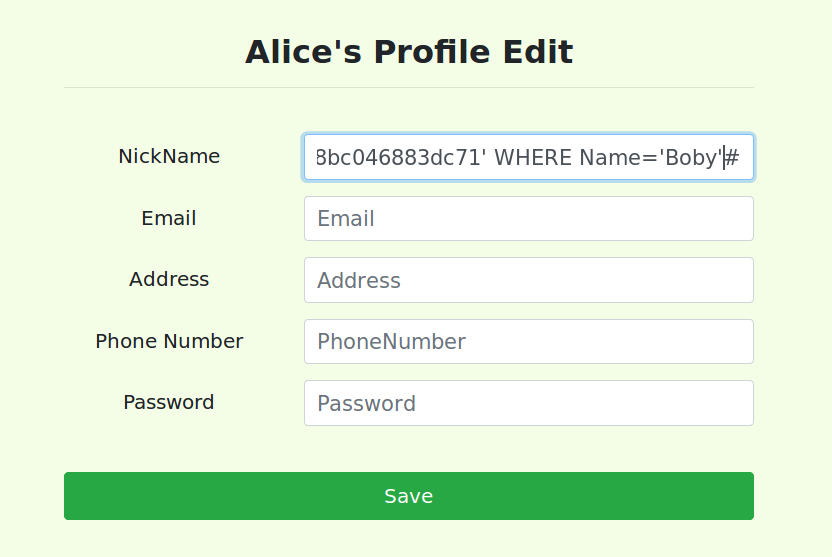
**3.3 Modify other people’password**

计算“abcabc”的SHA1加密后的值：

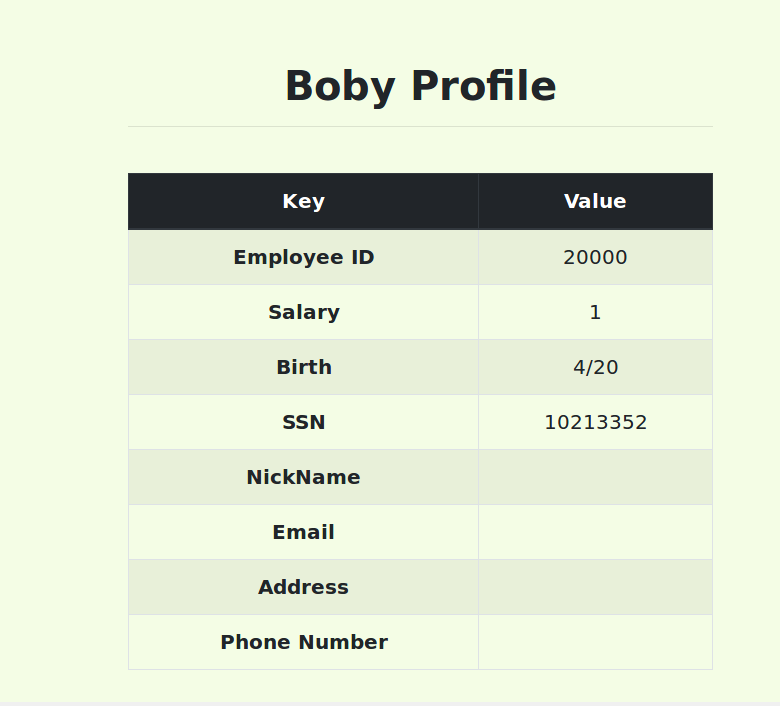


由于表格中存储的是密码的哈希值，所以要将对应的密文修改到Boby的密码中。

修改Alice的个人资料：



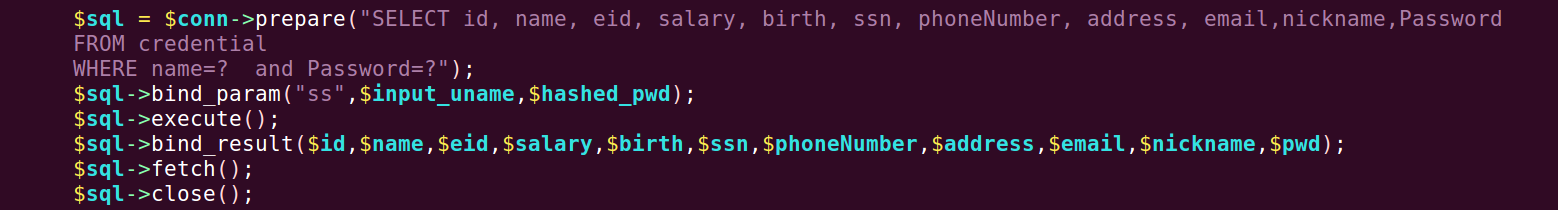
尝试用abcabc来登录Boby的账号：



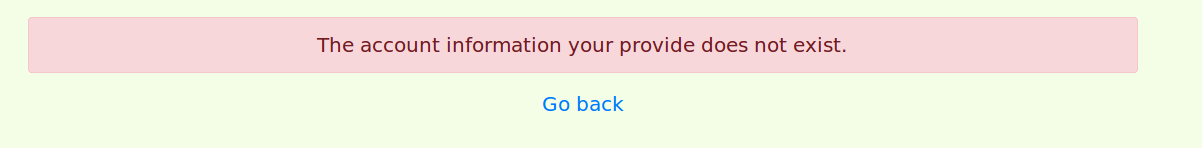
登录成功，说明Alice修改了Boby的账号。

TASK4：**Countermeasure——Prepared Statement**

修改/var/www/SQLInjection中的unsafe\_home.php文件：



再次用admin’# 尝试登录：



发现已经无法利用SQL注入攻击，说明漏洞已经修复。

**总结：**

SQL注入攻击也是利用程序无法辨别代码和数据的漏洞来达到攻击目的的方法。在SQL注入攻击中要善于利用注释符号#，使程序后续代码无法生效而执行了用户输入的恶意代码。

通过将代码和数据分开编译的方式可以有效抵御SQL注入攻击。