073 修改网站会员登录密码

* 核心技术

要实现修改密码功能，首先需要让用户输入原始密码，然后再使用原始密码与数据库中的密码进行对比，如果用户输入的原始密码正确，然后再将数据库中的密码更改为新密码。

Flask框架的werkzeug库为我们提供了密码生成函数generate\_password\_hash和密码验证函数check\_password\_hash。这两个函数的使用说明如下所示。

（1）密码生成函数：generate\_password\_hash

密码生成函数的定义如下：

werkzeug.security.generate\_password\_hash(password, method='pbkdf2:sha1', salt\_length=8)

generate\_password\_hash是一个密码加盐哈希函数，生成的哈希值可通过哈希之后的哈希字符串格式是这样的：

method$salt$hash

参数说明如下：

* password: 明文密码
* method: 哈希的方式（需要是hashlib库支持的），格式为:

pbpdf2:<method>[:iterations]。

参数说明：

* method：哈希的方式，一般为SHA1，
* iterations：（可选参数）迭代次数，默认为1000。
* slat\_length: 盐值的长度，默认为8。

密码生成示例如下：

1. **from** werkzeug.security **import** generate\_password\_hash
2. **print** generate\_password\_hash('123456')

运行结果如下：

'pbkdf2:sha1:1000$X97hPa3g$252c0cca000c3674b8ef7a2b8ecd409695aac370'

因为盐值是随机的，所以就算是相同的密码，生成的哈希值也不会是一样的。

（2）密码验证函数：check\_password\_hash

check\_password\_hash函数定义如下：

**werkzeug**.security.check\_password\_hash(**pwhash**, **password**)

其中，check\_password\_hash()函数用于验证经过generate\_password\_hash哈希的密码。若密码匹配，则返回真，否则返回假。

check\_password\_hash()函数的参数说明如下：

* pwhash: generate\_password\_hash生成的哈希字符串
* password: 需要验证的明文密码

密码验证示例如下：

1. **from** werkzeug.security **import** check\_password\_hash
2. pwhash = 'pbkdf2:sha1:1000$X97hPa3g$252c0cca000c3674b8ef7a2b8ecd409695aac370'
3. **print** check\_password\_hash(pwhash, '123456') # 输出为**True**

运行结果如下：

True