## Q1 SQL Injection

#### 2 Points

This homework has instant feedback. When you click "Save Answer," if the answer is correct, you will see an explanation. You can resubmit as many times as you want.

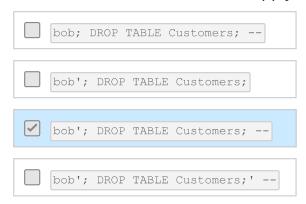
You find the following Java code in the client login section of an online banking website:

Assume that before issuing a request, the bank's server calls <a href="https://example.com/checkPassword">checkPassword</a> and ensures that the returned <a href="https://example.com/ResultSet">ResultSet</a> contains exactly one <a href="https://example.com/userID">userID</a>. If the set returns any other number of <a href="https://example.com/userID">userID</a> records, the bank fails the request. Otherwise the request is issued as the user represented by <a href="https://example.com/userID">userID</a>.

Assume execute queries can execute multiple SQL queries separated by a semicolon ;.

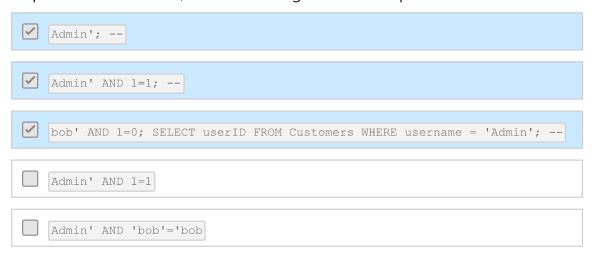
### Q1.1 1 Point

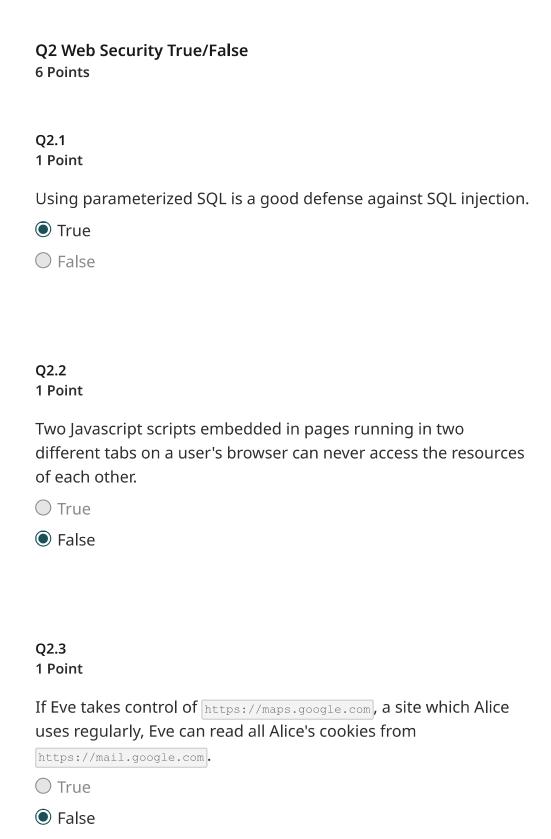
Which usernames could an attacker enter in order to delete the Customers table? Select all that apply.



## Q1.2 1 Point

What username could an attacker enter in order to issue a request as user Admin, without having to know the password?





Q2.4 1 Point
Browsers have a private browsing mode, which lets you visit websites without using any cookies.
○ True
False
Q2.5 1 Point
Because of the cookie policy, you cannot be tracked across domains by cookies.
○ True
<ul><li>False</li></ul>
03.6
Q2.6 1 Point
The same-origin policy prevents XSS (cross-site scripting) attacks
○ True

False

# **Q3 Snapitterbook**

3 Points

This question walks you through some common web security exploits and defenses, using a vulnerable social networking website called Snapitterbook.

For simplicity, throughout this question we define a helper function <code>render(endpoint, content)</code>, which displays <code>content</code> at the specified <code>endpoint</code> of the website. For example, <code>render("/hello", "Hello World")</code> will cause users to see <code>Hello World</code> when they visit <code>https://snapitterbook.com/hello</code>.

Snapitterbook displays a user's status on their wall. When a user updates their status with new\_status, the following Python code snippet runs:

```
query = "INSERT INTO statuses VALUES (?, ?);"

# add the status to the database using parameterized SQL
db.execute(query, username, new_status)
```

When someone visits a user's wall, the following Python code snippet runs:

```
query = "SELECT status from statuses WHERE user = ?"

# fetch the status from the database using parameterized SQL
status = db.execute(query, username)

# display the user's status on their wall
render("/{}/wall".format(username), status)
```

What type of attack is this code vulnerable to?

- SQL injection
- Stored XSS
- Reflected XSS
- CSRF
- O None of the above, the code is secure.

If a user tries to access the profile of a nonexistent user, Snapitterbook runs the following code snippet to display an error:

```
# assume username is read from a URL parameter
username = escape_sql(username)

# check if username exists
if not is_valid_username(username):
    render("/profile/{}".format(username), "{} does not exist".format(username))
```

# For example, if a user goes to

https://snapitterbook.com/profile/jason, they will see the message jason does not exist.

What type of attack is this code vulnerable to?

- SQL injection
- Stored XSS
- Reflected XSS
- CSRF
- O None of the above, the code is secure.

#### Q3.3 1 Point

Snapitterbook users can repost their friend's statuses by clicking a button on their friend's profile. When a user clicks the button, the following HTTP request is made:

https://snapitterbook.com/repost?post=<id of post>

The user's cookies are sent along with the request so that the server knows which user is reposting the status.

What type of attack is this code vulnerable to?

- SQL injection
- Stored XSS
- Reflected XSS
- CSRF
- O None of the above, the code is secure.

#### Q4 Feedback

#### **0** Points

Student Yiyun Chen

Total Points 11 / 11 pts

What's something we could do to make the class better? Or, what did you find most difficult or confusing from lectures or the rest of class, and what would you like to see explained better? If you have feedback, submit your comments here.

If you have feedback, submit your comments on <u>this form</u>. Your name will not be connected to any feedback you provide, and anything you submit here will not affect your grade.

Leaving feedback is completely optional! To encourage submissions, please grab the magic word from the feedback form (available in the first page) and enter it here.

# Homework 5 15 Minutes Late Select each question to review feedback and grading details.

Question 1				
SQL Injection	<b>2</b> / 2 pts			
1.1 (no title)	<b>1</b> / 1 pt			
1.2 (no title)	<b>1</b> / 1 pt			
Question 2				
Web Security True/False	<b>6</b> / 6 pts			
2.1 (no title)	<b>1</b> / 1 pt			
2.2 (no title)	<b>1</b> / 1 pt			
2.3 (no title)	<b>1</b> / 1 pt			
2.4 (no title)	<b>1</b> / 1 pt			
2.5 (no title)	<b>1</b> / 1 pt			
2.6 (no title)	<b>1</b> / 1 pt			
Question 3				
Snapitterbook	<b>3</b> / 3 pts			
3.1 (no title)	<b>1</b> / 1 pt			
3.2 (no title)	<b>1</b> / 1 pt			
3.3 (no title)	<b>1</b> / 1 pt			
Question 4				
Feedback	<b>0</b> / 0 pts			