1) Get all of the items listed at once via SQL injection:

```php
$v_cat = $_GET['category'];
$sqlstr = "SELECT * FROM product WHERE Category='" . $v_cat . "";
$rs = pg_query($sqlstr);
```

2) Find all inputs to drop users from the database (yes, there are many other things wrong with this):

```php
$user = $_GET['user'];
$password = $_GET['password'];
$prep = 'SELECT secretinfo FROM users WHERE ' .
    'uid = ' . $user . ' AND password = $1';
$rs = pg_query_params($prep, $password);
```
3) True or False and why, there is an XSS here:

```php
$name = $_GET['name'];
if (preg_match("<script>", $name)) {
    die;
}
echo("<p>Hello, " . $name . ".</p>");
```

4) Assume that an attacker from evil.com can control the JavaScript variable prevSite when a victim visits this code at profiles.com. Find an attack and write down a payload.

```html
<html>
<p><a id="myanchor">Return to Previous Website</a></p>
<script>
var prevSite = valueFromURL();
var elt = document.getElementById('myanchor');
elt.setAttribute('href', prevSite);
</script>
</p> </html>
```