Session Control

HTTP is a stateless protocol

What does this mean?

What this means

- HTTP has no built-in way of maintaining state between 2 transactions.

What this means

- HTTP has no built-in way of maintaining state between 2 transactions.

- I request a page from last.fm
- I request another page.
- last.fm doesn't know both requests are from the same user.

Session Control

track a user during a single session on a web site.

So when I request a page from a website, the php code on the server will know,

Hey, I've seen this guy before; his name is Ron; he has 12 MP3 tracks in his shopping cart.

It does this through session variables

It does this through session variables

- session variables are stored on the server.
- they persist across multiple pages.
- Unlike cookies, no expiration date. They die when the session ends.
  - one way a session ends is for the user to close the browser.

Implementing simple sessions

- starting a session
- using session variables
- destroying the session.

Starting a session

At the beginning of every PHP script
that needs state add the PHP function

```
session_start();
```

Store data on the server using superglobal variables

```
$_SESSION['username'] = 'raz';
$_SESSION['shoppingcart_id'] = '1290';
```

Using session variables

```
echo(“<p>You are logged in as $_SESSION[‘user’]</p>”);
```

Deleting session variables

```
unset($_SESSION[‘CART’]);
```

Closing a session

- session_destroy();
  - for ex., maybe want to close session when user logs out.
- user closes browser

</sessions>

<dangerous characters>

Search

For example, you can type 'movies', 'coffee', 'Mexican', or the name of a store like 'Starbucks

Search: Peet's go

movies

Movie	Theater
Avatar3D	Allen Cinema 4 Mesilla Valley
Dear John	Regal Fredericksburg 15
From Paris with Love	Marquee Cinemas Southpoint 9
The Book of Eli	Allen Cinema 4 Mesilla Valley
The Wolfman	Regal Fredericksburg 15
The Wolfman	Allen Cinema 4 Mesilla Valley
Valentine's Dat	Regal Fredericksburg 15

logout

Search

For example, you can type 'movies', 'coffee', 'Mexican', or the name of a store like 'Starbucks

Search: go

Peet's

Error Querying Database

mysqli_real_escape_string to the rescue

database connection

```
$movie = mysqli_real_escape_string($db, trim($_POST[‘movieName));

$synopsis = mysqli_real_escape_string($db, trim($_POST[‘movieName));

$query = “INSERT INTO movies VALUES ($movie, $synopsis)”;

$query2 = “SELECT rating FROM movies WHERE name = ‘$movie’”;
```

</mysqli_real_escape_string>

<password security>

Not a good idea to store password as clear text.

id	username	password
1	raz	p00d13
2	ann	changeme
3	lazy	qwerty

Solutions - MySQL fns.

sha

Terminal — mysql — 80×24

```
mysql>
mysql>
mysql>
mysql> SELECT SHA('p00d13');
+------------------------------------------+
| SHA('p00d13')                            |
+------------------------------------------+
| 44fdb64bf775e5c2dc37c3b43a74193519f03c96 |
+------------------------------------------+
1 row in set (0.00 sec)

mysql> DESC passwords;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(10) | YES  |     | NULL    |       |
| password | varchar(40) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> INSERT INTO passwords VALUES ('raz', SHA('p00d13'));
Query OK, 1 row affected (0.00 sec)

mysql>
```

Want to check if user entered correct password

```
$pw = $_POST[‘password’]
```

Want to check if user entered correct password

```
$pw = $_POST['password'];

SHA is one-way - we can encrypt but we cannot decrypt.

$query = "SELECT * FROM passwords WHERE password = SHA($pw)";
```

Terminal — mysql — 80×24

```
mysql>
mysql>
mysql>
mysql>
mysql> SELECT * FROM passwords WHERE password = SHA('p00d13');
+----------+------------------------------------------+
| username | password                                 |
+----------+------------------------------------------+
| raz      | 44fdb64bf775e5c2dc37c3b43a74193519f03c96 |
+----------+------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM passwords;
+----------+------------------------------------------+
| username | password                                 |
+----------+------------------------------------------+
| ann      | fa9beb99e4029ad5a6615399e7bbae21356086b3 |
| lazy     | 5eab7a25fdc1cbc959eb8378386b557adbb23265 |
| raz      | 44fdb64bf775e5c2dc37c3b43a74193519f03c96 |
+----------+------------------------------------------+
3 rows in set (0.00 sec)

mysql>
```

Not a good idea to store password as clear text.

id	username	password
1	raz	p00d13
2	ann	changeme
3	lazy	qwerty

Summary

saw how to implement sessions

saw how to escape dangerous characters

saw how to handle passwords

task

- clone session repository (see website)
- implement sessions. So when a user types in a zip code it is remembered on future searches.
- escape all strings sent to mysql
- implement create account
- implement secure login. (opt. logout)