Lecture #21: Building Twitter out of PHP and MySQL
Administrivia

- **Homework**
  - Homework 7 due TODAY, HW 8 Going out

- **Do Some Reading**
  - Chapters 9-11 PHP & MYSQL Book
On The Menu

- PHP And MYSQl Review
- Building Twitter
  - Database Schema
  - Forms To Build
PHP & MySQL

- connect
  • $database_connection = mysqli_connect($host, $username, $password, $database_name);
PHP Security

- Don't put sensitive (e.g. passwords) information in web-accessible folders.
- PHP on i6 can include files from non-web-accessible folders, put sensitive information there.
PHP Mysql

- Query
  - $results = mysqli_query(
      $database_connection,
      $sql
  );
PHP Mysql

- Handle Results
  - while ($row = $result->fetch_assoc()) {
    ...
  }

  - Iterates over all rows returned by the database as an associative array between column names and values.
PHP Query Security

- PHP will often take user inputs and place those inputs into a query.
- Malicious actors (bad hackers) can embed MySQL commands into their inputs that can affect the underlying database and inadvertently expose your private data.
PHP Query Security

- Suppose you have a form that takes a user name and then queries the database for the username.
- "SELECT * FROM user WHERE user_name ='".$user_name.";"
- What would happen if a bad actor submitted this as a user name:
  - '; DROP TABLES;
- Official Name: SQL Injection attack
PHP SQL Injection

Hi, this is your son's school. We're having some computer trouble.

Oh, dear - did he break something?

In a way-

Did you really name your son Robert'); DROP TABLE Students--;?

Oh, yes. Little Bobby Tables, we call him.

Well, we've lost this year's student records. I hope you're happy.

And I hope you've learned to sanitize your database inputs.
Protecting Against SQL Injection

- Use the `mysqli_real_escape_string` on ALL database inputs.
- E.G.
- $sql = "SELECT * FROM x WHERE y = '' . mysqli_real_escape_string($z).'''" ;
GET VS POST

- Generally speaking:
  - We use HTTP GET when we only want information
  - We use HTTP POST when we want to store information
    - Or there are security concerns
    - Or the amount of information passed along is too big for a get request

- HTTP GET parameters are passed in the URL not in the request body.
Parts Of A URL

- `http://<host>/<path>?param1=val&param2=val2`
- E.G.
  - `http://google.com/search?q=nyu`
GET In PHP

- URL Parameters get added to $_REQUEST array.
- If the request is an HTTP GET the request parameters get added to $_GET array.
Twitter!

user_follows
- user_id INT
- follow_user_id INT
- follow_time TIMESTAMP

Indexes

user
- user_id INT
- user_name VARCHAR(255)
- first_name VARCHAR(255)
- last_name VARCHAR(45)
- email_address VARCHAR(255)
- image_url VARCHAR(255)
- bio VARCHAR(140)

Indexes

retweet
- user_id INT
- tweet_id INT
- retweet_time DATETIME

Indexes

tweet
- tweet_id INT
- tweet_text VARCHAR(140)
- tweet_time TIMESTAMP
- user_id INT

Indexes
Let's Build

- **User View**
  - Show user photo, bio and username
  - List All Tweets for a Given User
  - List All People Followed By User
  - Tweet Stream
    - Show all tweets for the users followed by a user.
    - Show all tweets for a given user
  - Post a Tweet