MSCI 346 – Spring 2019

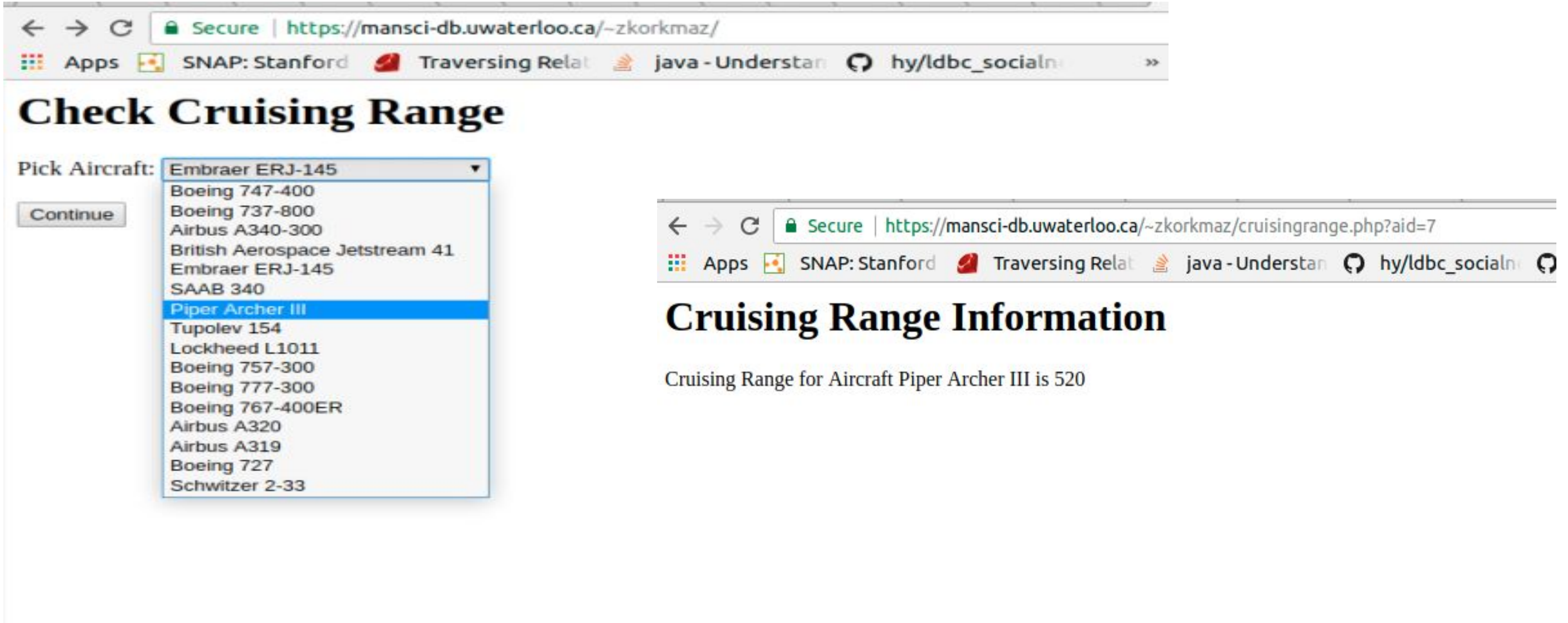
LAB #6 - PHP/SQL

Recap of Lab 5

- PhpMyAdmin
  - Advanced SQL queries
- PHP-MySQLi Connection
  - Running queries with user-supplied parameters and print query output in php

Lab 5: Exercise-3

Application: Check cruising range of an aircraft



Check Cruising Range

Pick Aircraft: Embraer ERJ-145

Cruising Range Information

Cruising Range for Aircraft Piper Archer III is 520

Lab 5: Exercise-4

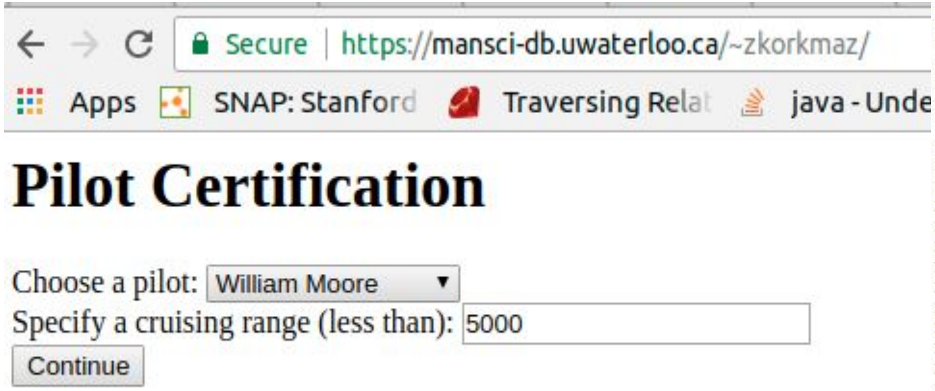
Application: Find the aircrafts that a pilot is certified for and have cruising ranges greater than a specified value.

- List all employees in drop-down list
- Input a cruising range

} You need to bind two parameters!!

- Use the selected pilot and cruising range to list the aircrafts that she or he is certified for.
  - If the result is empty -> print “No record found”

Lab 5: Exercise-4

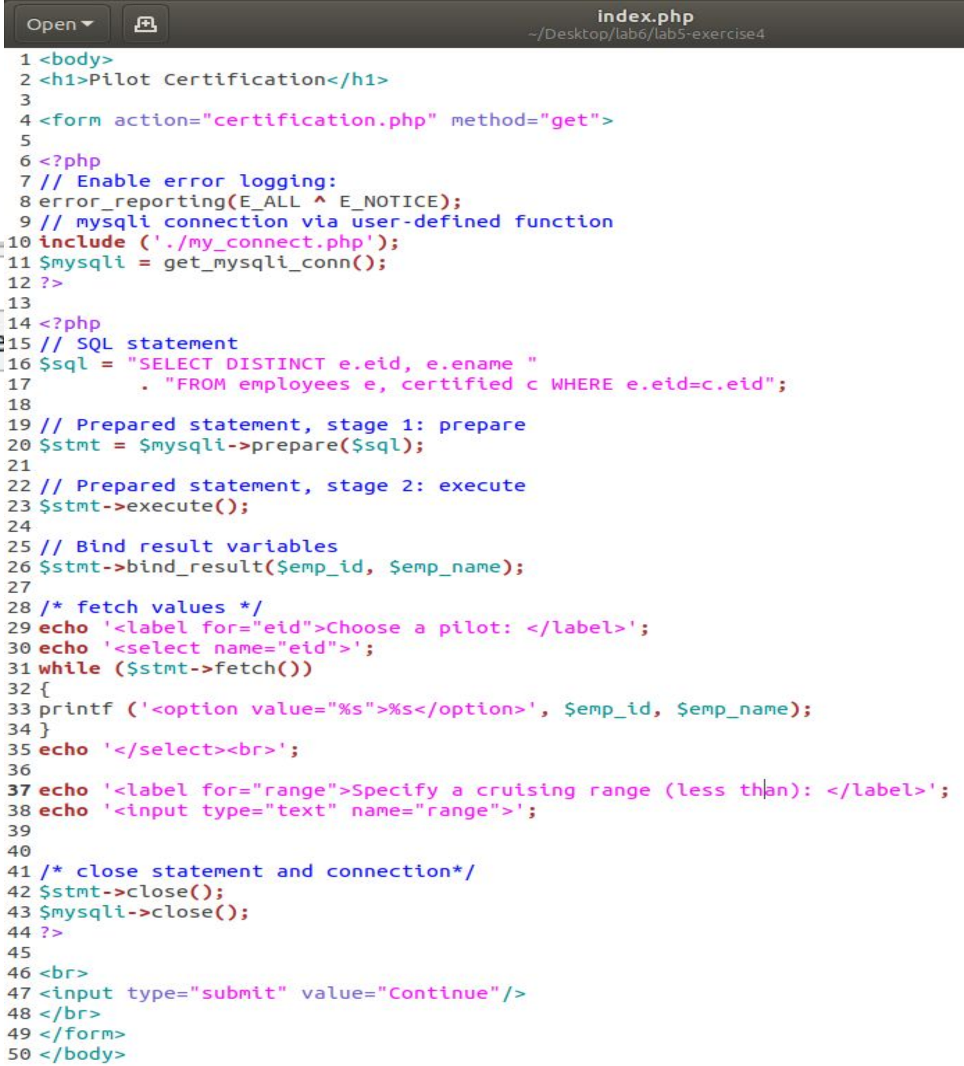


Pilot Certification

Choose a pilot: William Moore ▾

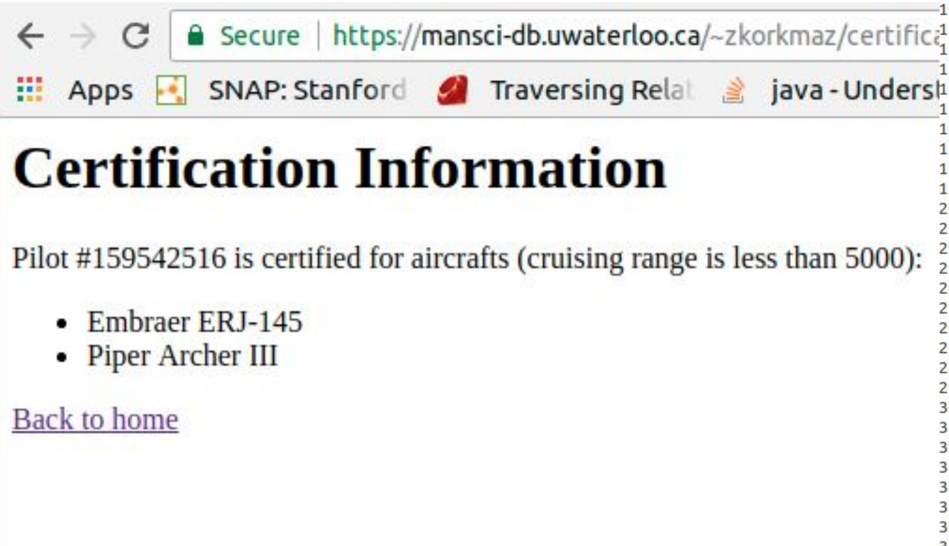
Specify a cruising range (less than): 5000

Continue



```php
<body>
<h1>Pilot Certification</h1>

<form action="certification.php" method="get">

<?php
// Enable error logging:
error_reporting(E_ALL ^ E_NOTICE);
// mysqli connection via user-defined function
include ('./my_connect.php');
$mysqli = get_mysqli_conn();
?>

<?php
// SQL statement
$sql = "SELECT DISTINCT e.eid, e.ename "
        . "FROM employees e, certified c WHERE e.eid=c.eid";

// Prepared statement, stage 1: prepare
$stmt = $mysqli->prepare($sql);

// Prepared statement, stage 2: execute
$stmt->execute();

// Bind result variables
$stmt->bind_result($emp_id, $emp_name);

/* fetch values */
echo '<label for="eid">Choose a pilot: </label>';
echo '<select name="eid">';
while ($stmt->fetch())
{
printf ('<option value="%s">%s</option>', $emp_id, $emp_name);
}
echo '</select><br>';

echo '<label for="range">Specify a cruising range (less than): </label>';
echo '<input type="text" name="range">';


/* close statement and connection*/
$stmt->close();
$mysqli->close();
?>

<br>
<input type="submit" value="Continue"/>
</br>
</form>
</body>
```

Lab 5: Exercise-4

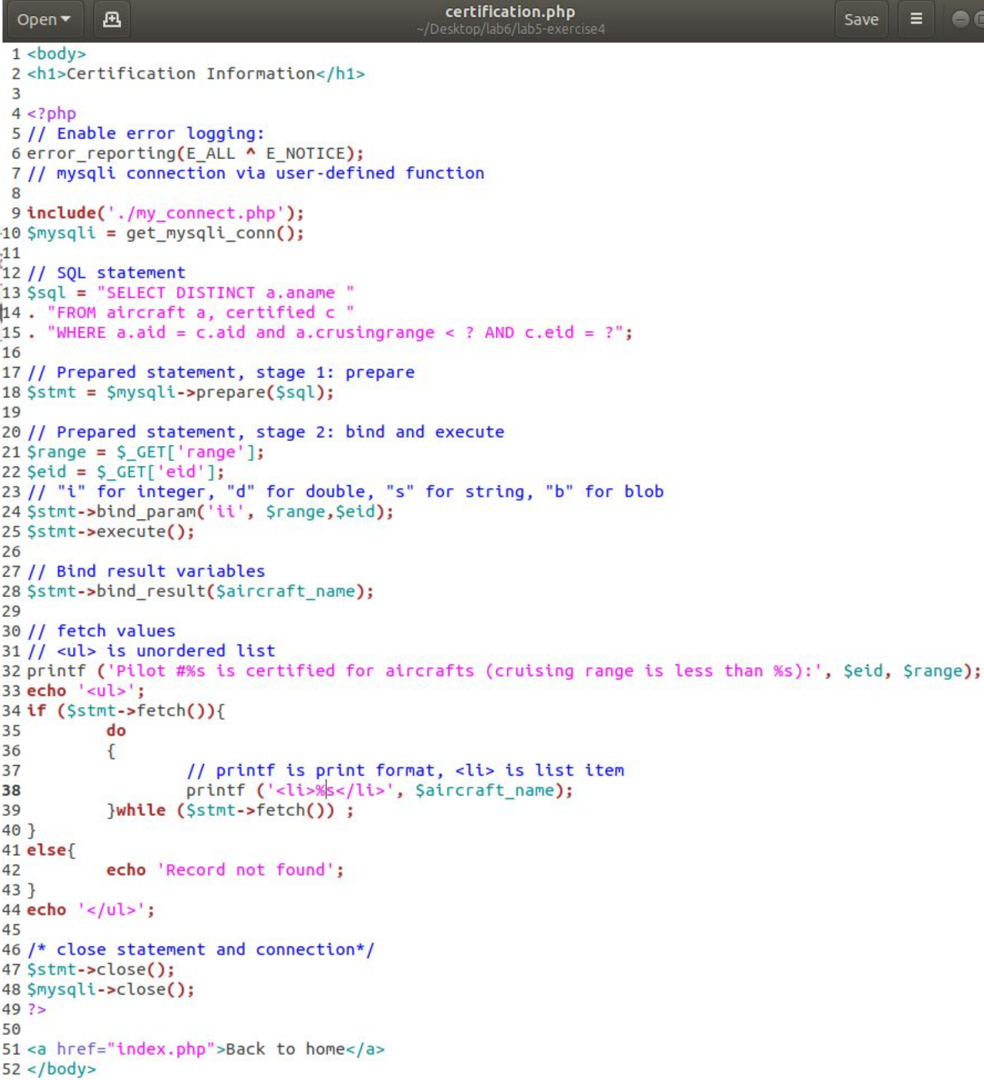


Certification Information

Pilot #159542516 is certified for aircrafts (cruising range is less than 5000):

- Embraer ERJ-145
- Piper Archer III

Back to home



```php
<body>
<h1>Certification Information</h1>

<?php
// Enable error logging:
error_reporting(E_ALL ^ E_NOTICE);
// mysqli connection via user-defined function

include('./my_connect.php');
$mysqli = get_mysqli_conn();

// SQL statement
$sql = "SELECT DISTINCT a.aname "
. "FROM aircraft a, certified c "
. "WHERE a.aid = c.aid and a.crusingrange < ? AND c.eid = ?";

// Prepared statement, stage 1: prepare
$stmt = $mysqli->prepare($sql);

// Prepared statement, stage 2: bind and execute
$range = $_GET['range'];
$eid = $_GET['eid'];
// "i" for integer, "d" for double, "s" for string, "b" for blob
$stmt->bind_param('ii', $range,$eid);
$stmt->execute();

// Bind result variables
$stmt->bind_result($aircraft_name);

// fetch values
// <ul> is unordered list
printf ('Pilot #%s is certified for aircrafts (cruising range is less than %s):', $eid, $range);
echo '<ul>';
if ($stmt->fetch()){
        do
        {
                // printf is print format, <li> is list item
                printf ('<li>%s</li>', $aircraft_name);
        }while ($stmt->fetch()) ;
}
else{
        echo 'Record not found';
}
echo '</ul>';

/* close statement and connection*/
$stmt->close();
$mysqli->close();
?>

<a href="index.php">Back to home</a>
</body>
```

Today..

PHP application needs to communicate with a database server

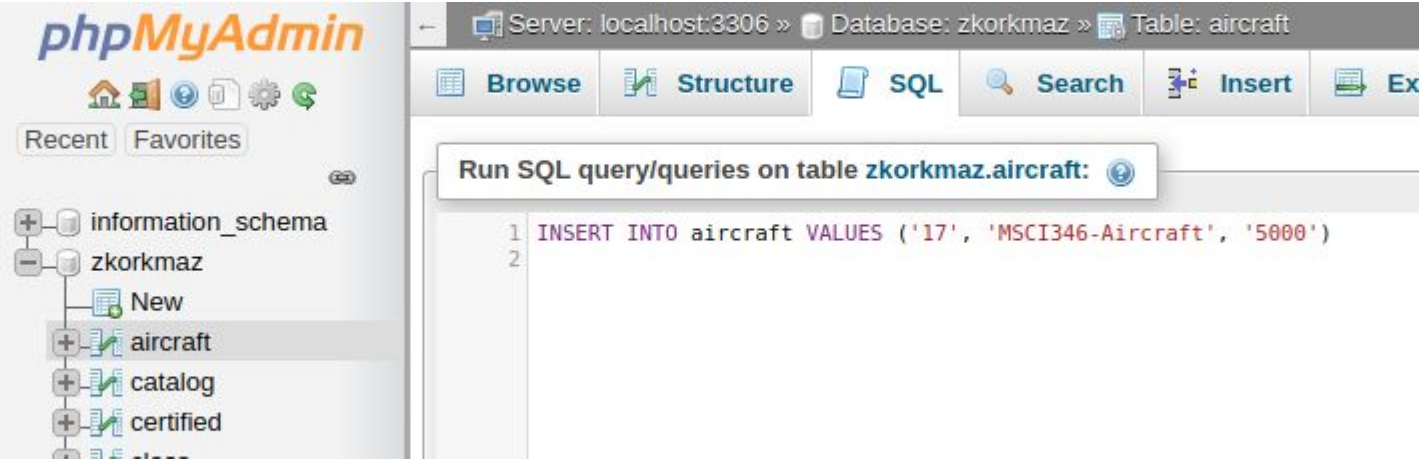
- Connecting to the database server
- Querying the database
- Displaying results on a web form
- Taking user input through a web form
- Querying the database using user-input as a parameter
- **Modify tables in the database**
  - **Inserting the user-input into the database**
  - **Updating/deleting records**

Possible Scenarios

A web application may offer various functions. For example;

- We bought a new aircraft. -> add a new aircraft to our fleet
- Stop flying long distances -> remove flights distance>?
- Somebody misspelled the name of aircraft ID#5 -> update the name of the aircraft

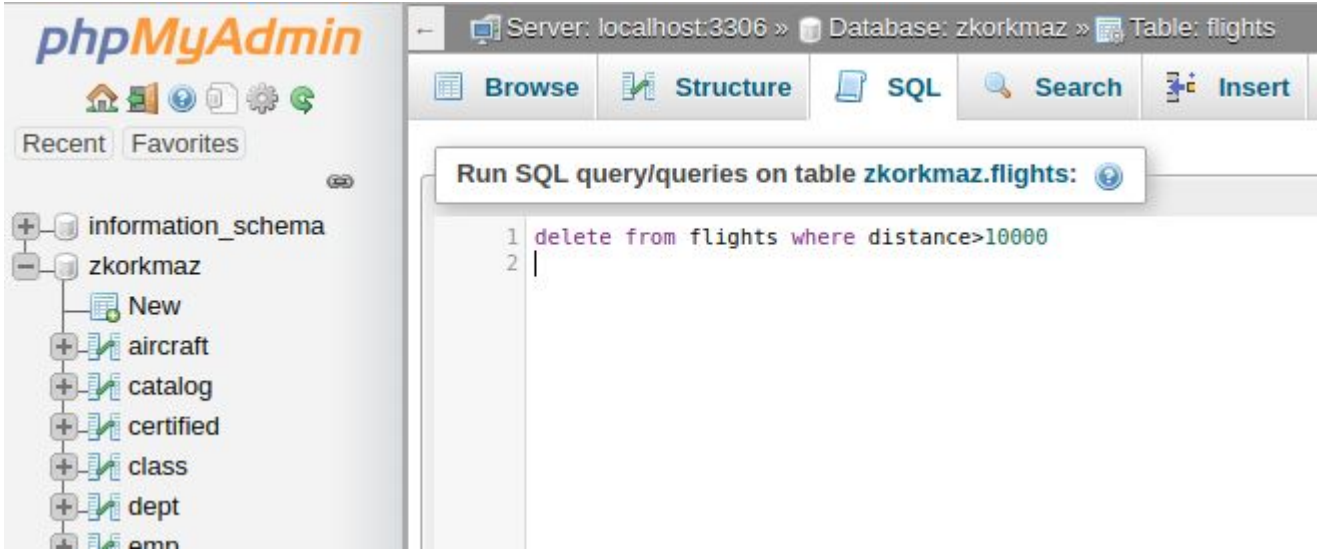
Example: We bought a new Aircraft



Run SQL query/queries on table zkorkmaz.aircraft:

```
INSERT INTO aircraft VALUES ('17', 'MSCI346-Aircraft', '5000')
```

Example: Stop flying long distances

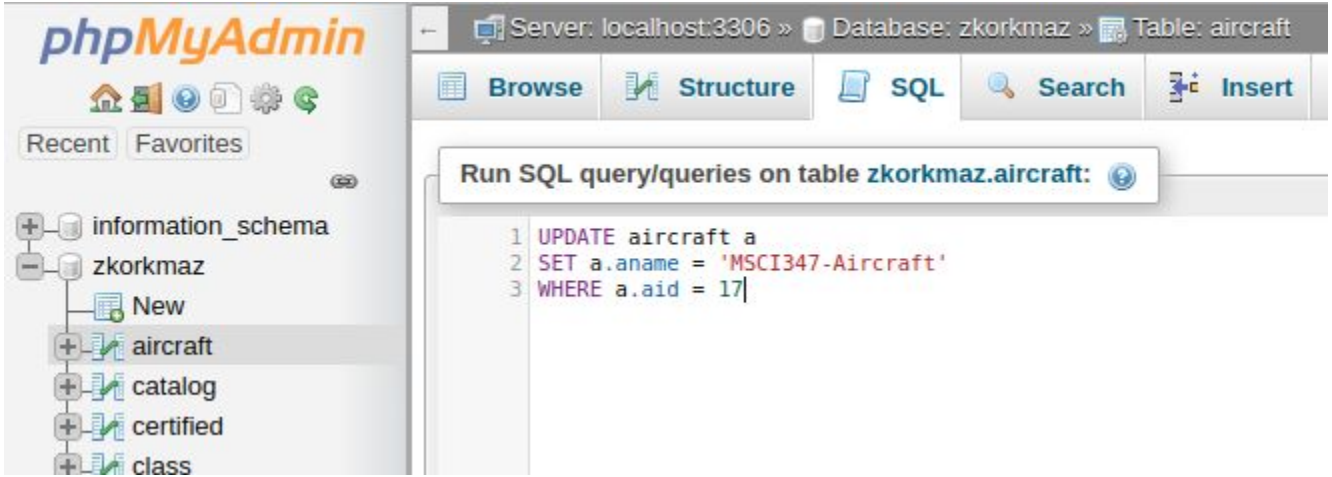


Run SQL query/queries on table zkorkmaz.flights:

```sql
delete from flights where distance>10000

```

Example: Misspelled name of aircraft ID#17

- Aircraft id#17 name is MSCI436-Aircraft
  - Change it as MSCI437-Aircraft



Run SQL query/queries on table zkorkmaz.aircraft:

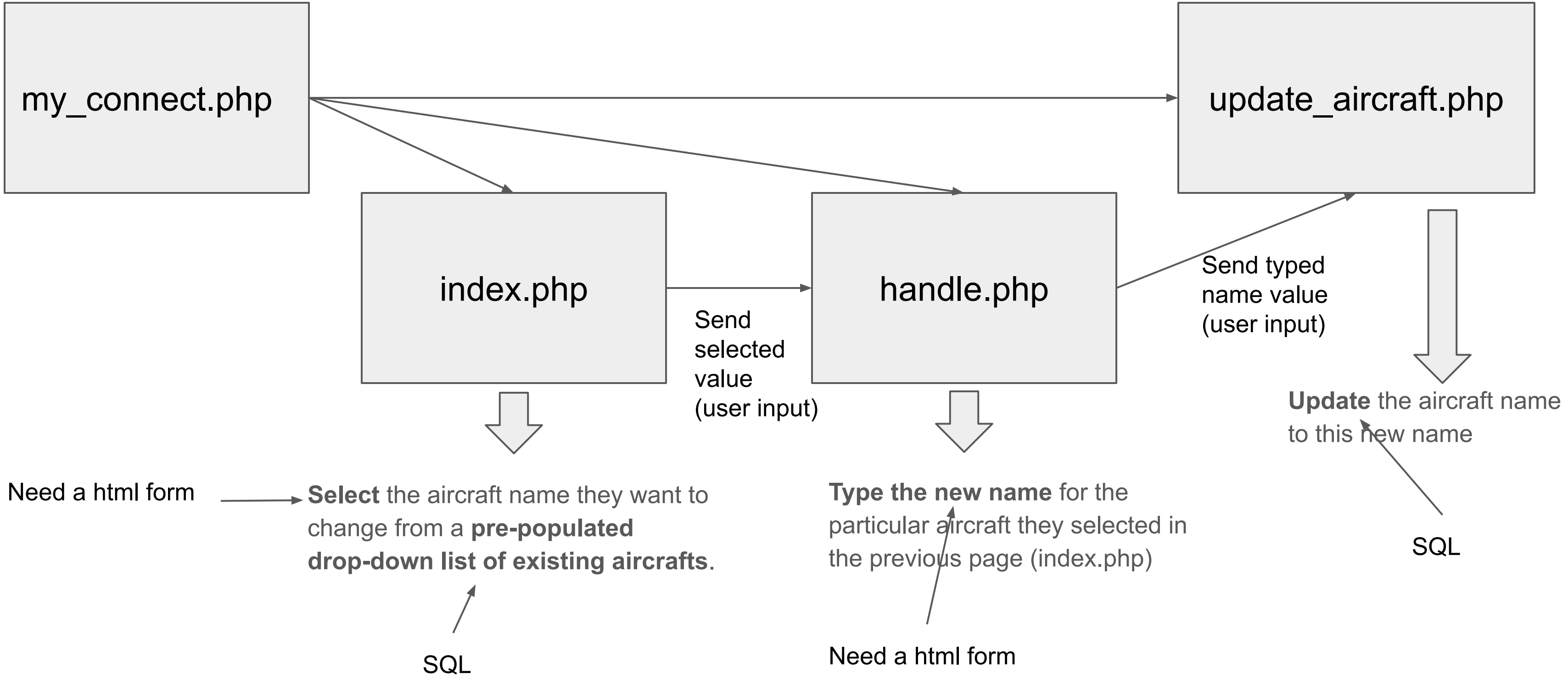
```sql
UPDATE aircraft a
SET a.aname = 'MSCI347-Aircraft'
WHERE a.aid = 17
```

A Scenario: Update aircraft name using web application

We will make a web application that will enable users to:

- Select the aircraft name they want to change from a pre-populated drop-down list of existing aircrafts.
- Type the new name for the particular aircraft they selected in the previous page.
- Update the aircraft name to this new name

Outline for the solution



my_connect.php

index.php

handle.php

update_aircraft.php

Send selected value (user input)

Send typed name value (user input)

Need a html form

Select the aircraft name they want to change from a **pre-populated drop-down list of existing aircrafts**.

SQL

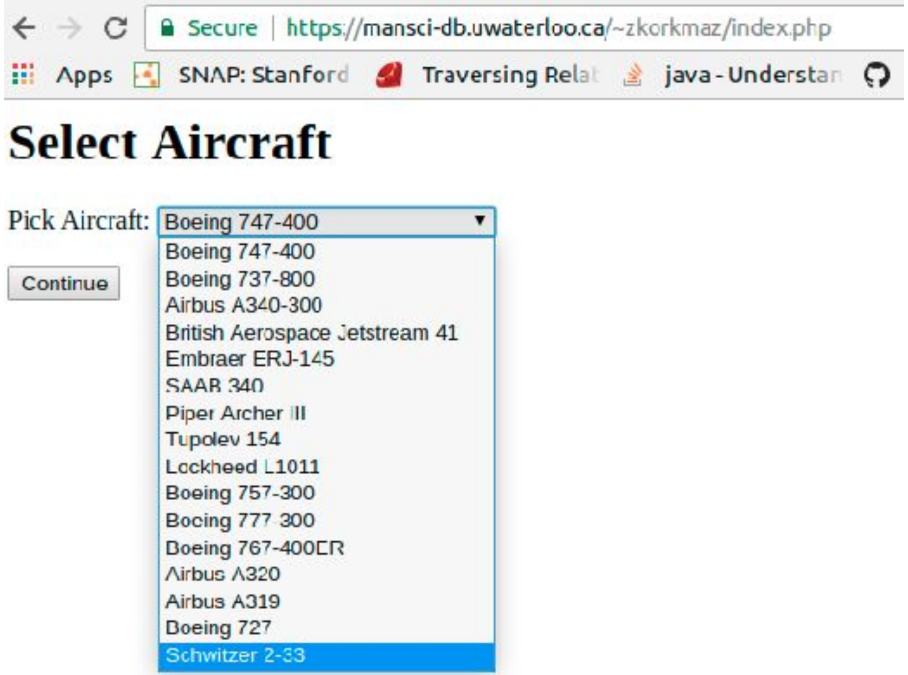
Type the new name for the particular aircraft they selected in the previous page (index.php)

Need a html form

Update the aircraft name to this new name

SQL

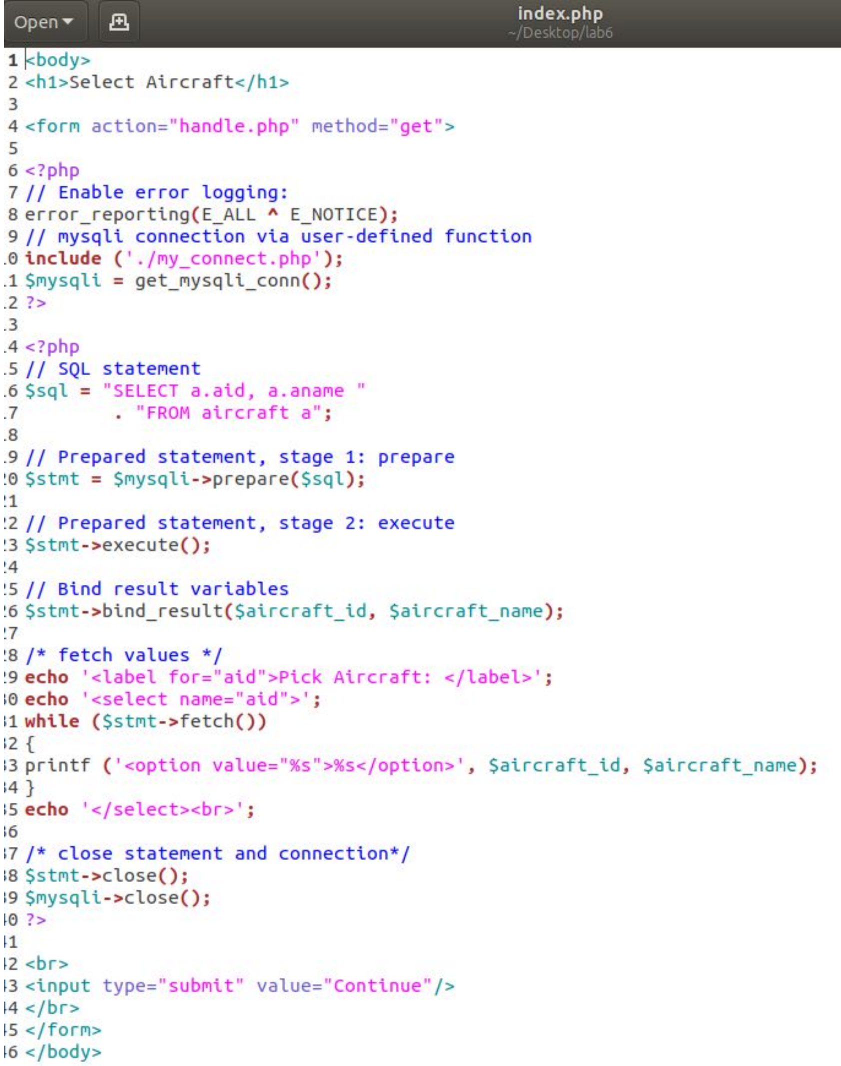
Index.php (Select the aircraft name)



Select Aircraft

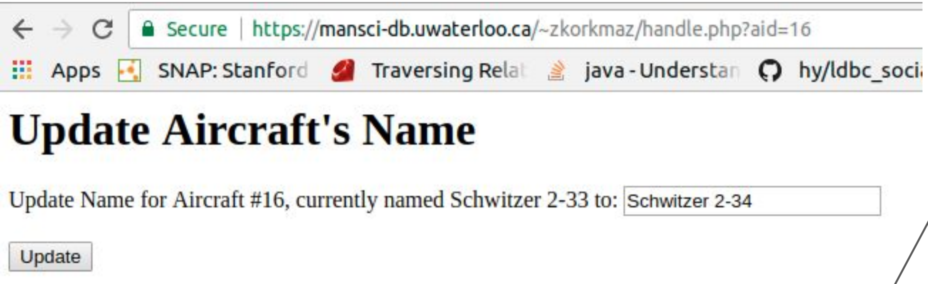
Pick Aircraft: Boeing 747-400

- Boeing 747-400
- Boeing 737-800
- Airbus A340-300
- British Aerospace Jetstream 41
- Embraer ERJ-145
- SAAB 340
- Piper Archer III
- Tupolev 154
- Lockheed L1011
- Boeing 757-300
- Boeing 777-300
- Boeing 767-400ER
- Airbus A320
- Airbus A319
- Boeing 727
- Schwitzer 2-33



```php
<body>
<h1>Select Aircraft</h1>

<form action="handle.php" method="get">

<?php
// Enable error logging:
error_reporting(E_ALL ^ E_NOTICE);
// mysqli connection via user-defined function
include ('./my_connect.php');
$mysqli = get_mysqli_conn();
?>

<?php
// SQL statement
$sql = "SELECT a.aid, a.aname "
        . "FROM aircraft a";

// Prepared statement, stage 1: prepare
$stmt = $mysqli->prepare($sql);

// Prepared statement, stage 2: execute
$stmt->execute();

// Bind result variables
$stmt->bind_result($aircraft_id, $aircraft_name);

/* fetch values */
echo '<label for="aid">Pick Aircraft: </label>';
echo '<select name="aid">';
while ($stmt->fetch())
{
printf ('<option value="%s">%s</option>', $aircraft_id, $aircraft_name);
}
echo '</select><br>';

/* close statement and connection*/
$stmt->close();
$mysqli->close();
?>

<br>
<input type="submit" value="Continue"/>
</br>
</form>
</body>
```

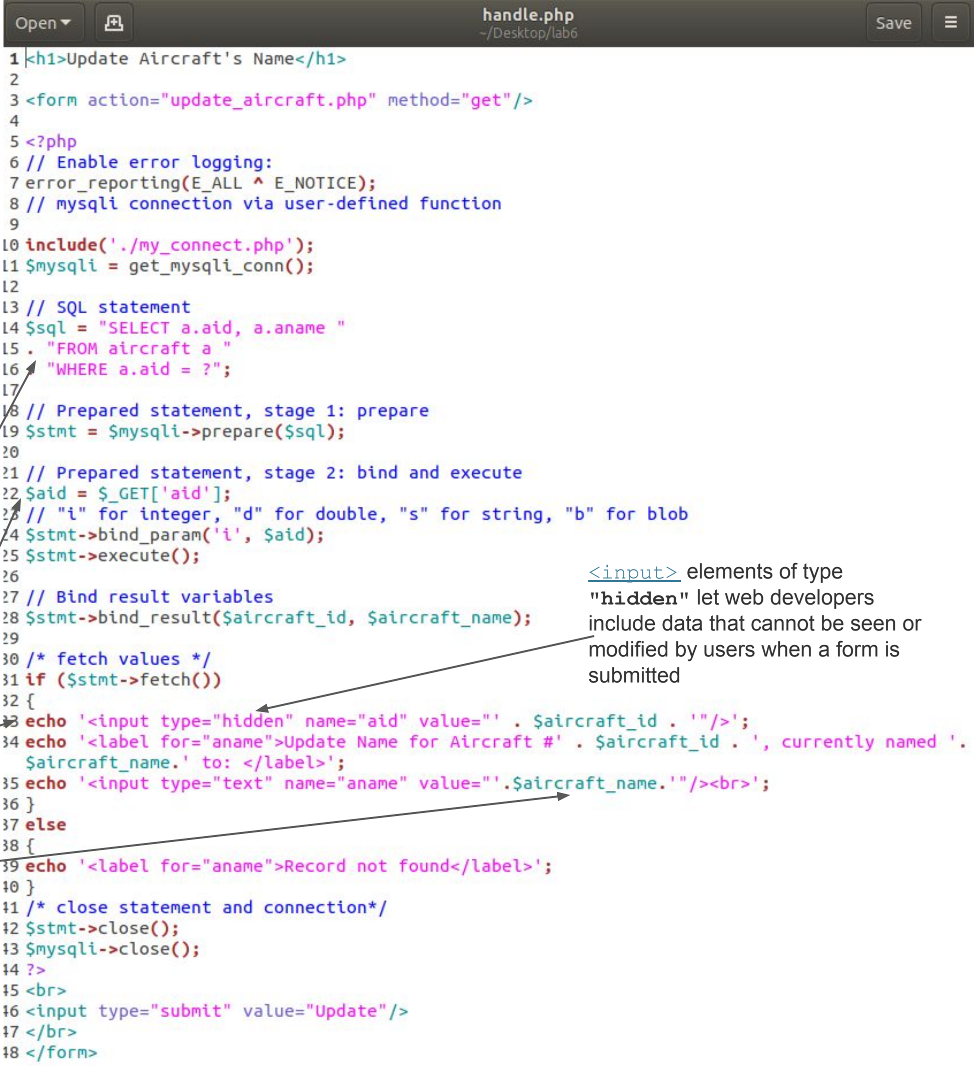
handle.php(Type the new name)



Update Aircraft's Name

Update Name for Aircraft #16, currently named Schwitzer 2-33 to: Schwitzer 2-34

- The aircraft name needs to be retrieved from the database again.
- Notice how the value of aid is retrieved from GET and stored in a local variable called $aid
- Notice that even though the aircraft id is displayed using a label, a hidden text box (input type) is used. This is done because the GET method passes only the values of input objects automatically.
- Notice that the current name is echoed into the text box for the new name. This is done for the case where the user accidently hits “Update”, changing the aircraft name to blank
- On clicking “Update”, the GET method caches the current value of text box “**aid**” and text box “**aname**”, passing them over to update_aircraft.php

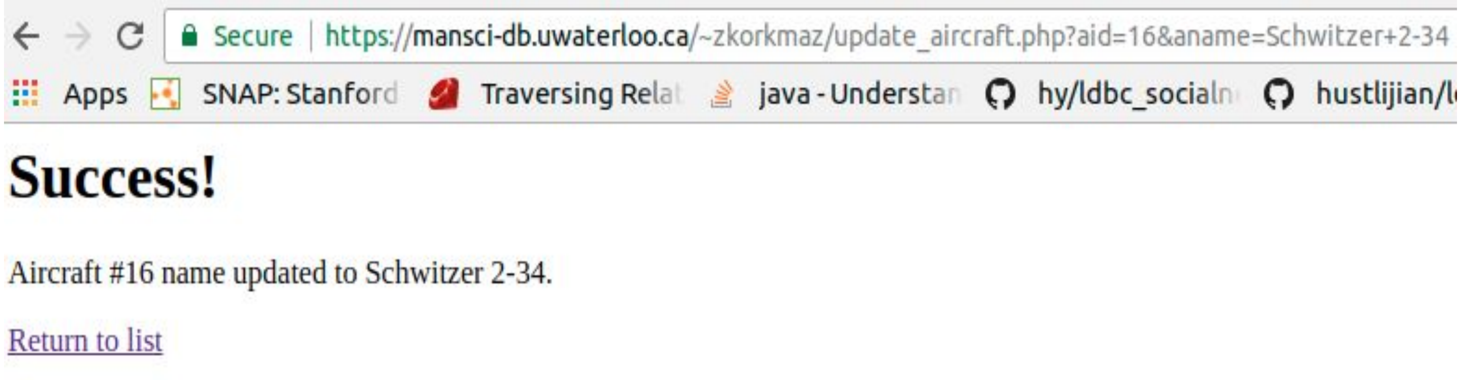


```
<h1>Update Aircraft's Name</h1>

<form action="update_aircraft.php" method="get"/>

<?php
// Enable error logging:
error_reporting(E_ALL ^ E_NOTICE);
// mysqli connection via user-defined function

include('./my_connect.php');
$mysqli = get_mysqli_conn();

// SQL statement
$sql = "SELECT a.aid, a.aname "
. "FROM aircraft a "
. "WHERE a.aid = ?";

// Prepared statement, stage 1: prepare
$stmt = $mysqli->prepare($sql);

// Prepared statement, stage 2: bind and execute
$aid = $_GET['aid'];
// "i" for integer, "d" for double, "s" for string, "b" for blob
$stmt->bind_param('i', $aid);
$stmt->execute();

// Bind result variables
$stmt->bind_result($aircraft_id, $aircraft_name);

/* fetch values */
if ($stmt->fetch())
{
echo '<input type="hidden" name="aid" value="' . $aircraft_id . '"/>';
echo '<label for="aname">Update Name for Aircraft #' . $aircraft_id . ', currently named '.
$aircraft_name.' to: </label>';
echo '<input type="text" name="aname" value="'.$aircraft_name.'"/><br>';
}
else
{
echo '<label for="aname">Record not found</label>';
}
/* close statement and connection*/
$stmt->close();
$mysqli->close();
?>
<br>
<input type="submit" value="Update"/>
</br>
</form>
```

<input> elements of type **"hidden"** let web developers include data that cannot be seen or modified by users when a form is submitted

update_aircraft.php()

- The final step updates the name of the aircraft.
- The status of the update is communicated to the user.
- A link is provided to update another aircraft’s name.
  - Hint for course project: In an application with many functions, links to both the update feature as well as the homepage (with links to all functions) must be provided.


Secure
https://mansci-db.uwaterloo.ca/~zkorkmaz/update_aircraft.php?aid=16&aname=Schwitzer+2-34
Apps
SNAP: Stanford
Traversing Relat
java - Understan
hy/ldbc_socialn
hustlijian/l

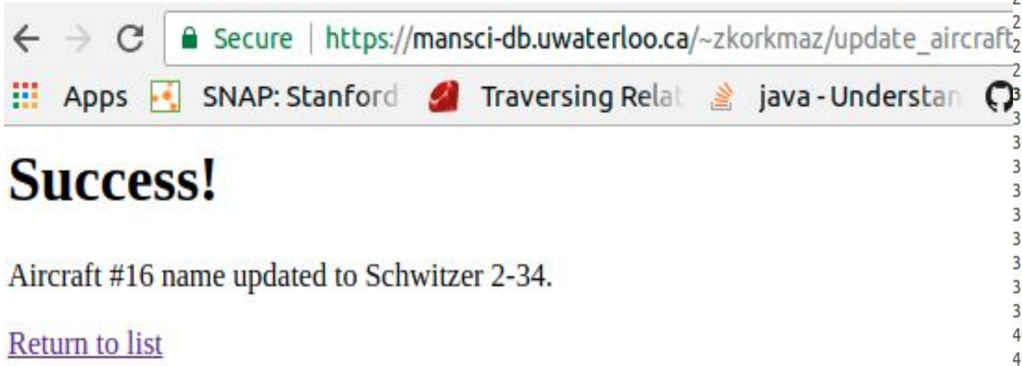
Success!

Aircraft #16 name updated to Schwitzer 2-34.

Return to list

Exercise: update_aircraft.php()

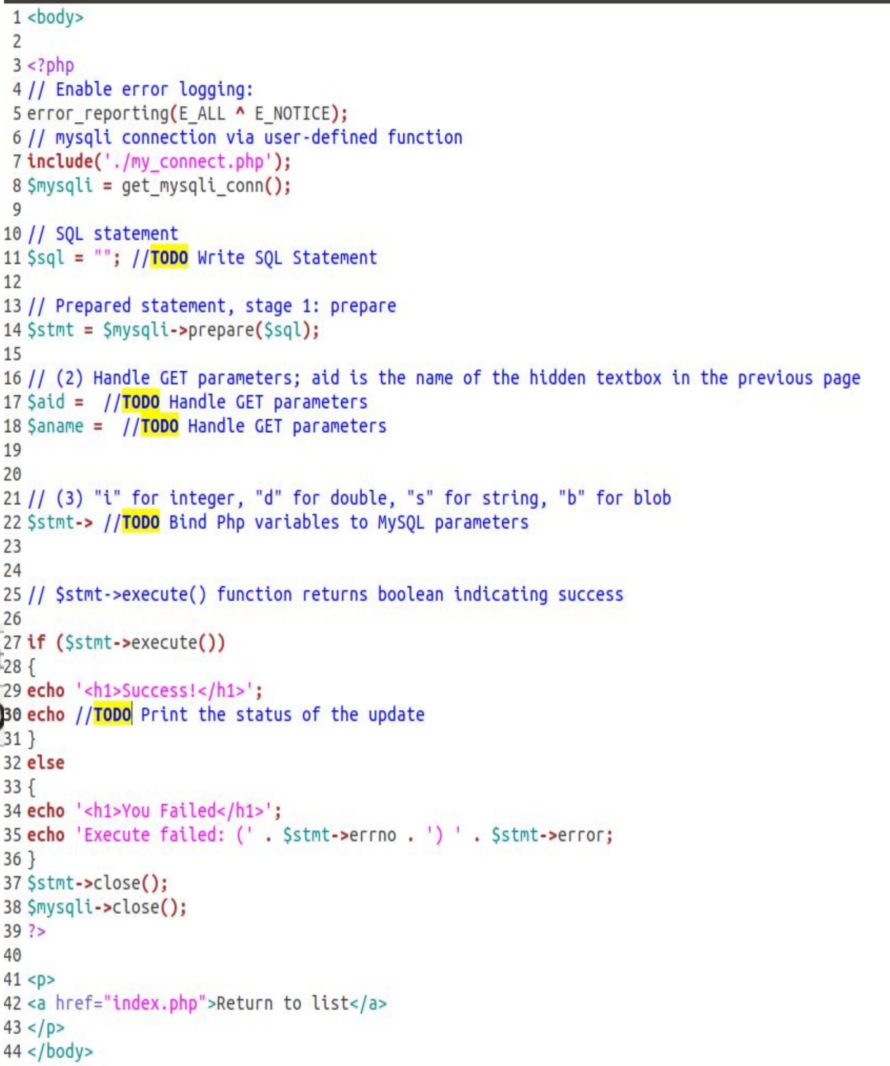
- Download & Modify update_aircraft.php file!
    - TODO comments



Success!

Aircraft #16 name updated to Schwitzer 2-34.

Return to list



```
<body>

<?php
// Enable error logging:
error_reporting(E_ALL ^ E_NOTICE);
// mysqli connection via user-defined function
include('./my_connect.php');
$mysqli = get_mysqli_conn();

// SQL statement
$sql = ""; //TODO Write SQL Statement

// Prepared statement, stage 1: prepare
$stmt = $mysqli->prepare($sql);

// (2) Handle GET parameters; aid is the name of the hidden textbox in the previous page
$aid =  //TODO Handle GET parameters
$aname =  //TODO Handle GET parameters


// (3) "i" for integer, "d" for double, "s" for string, "b" for blob
$stmt-> //TODO Bind Php variables to MySQL parameters


// $stmt->execute() function returns boolean indicating success

if ($stmt->execute())
{
echo '<h1>Success!</h1>';
echo //TODO Print the status of the update
}
else
{
echo '<h1>You Failed</h1>';
echo 'Execute failed: (' . $stmt->errno . ') ' . $stmt->error;
}
$stmt->close();
$mysqli->close();
?>

<p>
<a href="index.php">Return to list</a>
</p>
</body>
```