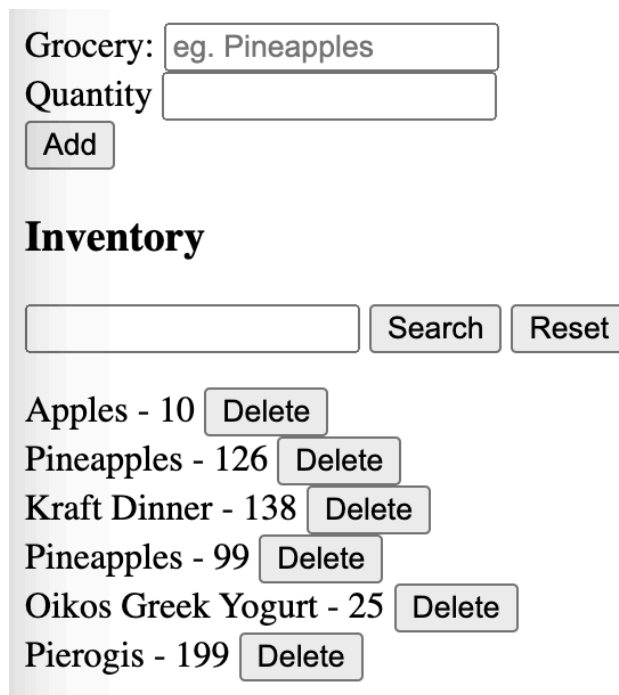# COMP 3015, BCIT Computing
Final Exam Part 2

Searchable Grocery Store Inventory Application

Write a script called **index.php** which provides the form(s) to add a grocery item and associated quantity, search the existing inventory and reset the search (show everything again). Each record can also be deleted from the database. You must use parameterized prepared statements in order to ensure your application is not vulnerable to SQL injection.
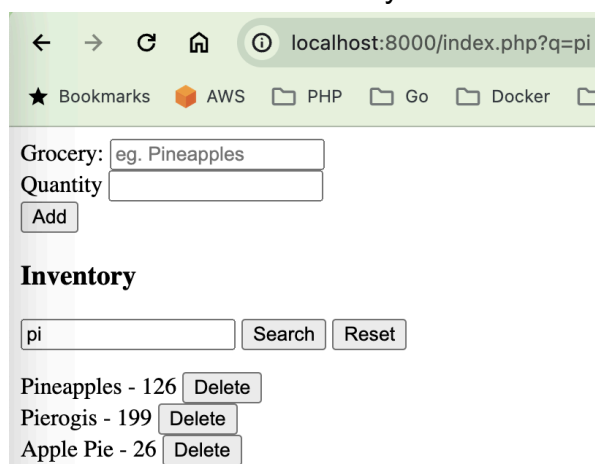

Fig 1. The grocery store inventory application with 6 records added.

When a new inventory item is added, an HTTP POST request containing the data should be made to the server. The implementation of deletion should also make a POST request to the server.

When a term is searched, an HTTP GET request should be made to the server. The list should be filtered to only contain matching items:

Fig 2. The application with a search term of "pi".

As this is not a SQL course, here's a hint on the search. You can use the MySQL "LIKE" operator in order to implement a simple search:

```
$dsn = "mysql:host=localhost;dbname=c3015_final";
$pdo = new PDO($dsn, "usernameHere", "passwordHere");
…
$pdoStatement = $pdo->prepare("SELECT * FROM inventory WHERE title LIKE ?");
$pdoStatement->execute(["%$query%"]);
```

When the "reset" button is clicked, the filter should be removed and all todo items shown again.

The SQL commands to create the database are as follows:

```
CREATE DATABASE c3015_final;
USE c3015_final;
CREATE TABLE inventory (
    id          INT UNSIGNED auto_increment,
    item_name   VARCHAR(255),
    quantity    INT UNSIGNED,
    PRIMARY KEY(id)
);
```

Some reminders to help...

If you need to drop your database for any reason:
```
DROP DATABASE c3015_final;
```

Reminder on SELECT statements and WHERE clauses:
```
SELECT * FROM inventory WHERE id = 1;
```

Reminder on INSERT statements:
```
INSERT INTO inventory (item_name, quantity) VALUES ("3% Milk", 2);
```

Reminder on DELETE statements:
```
DELETE FROM inventory WHERE id = 1;
```