CS1116/CS5018 Web Development II

Dr. D.G. Bridge

Answer all questions.
Silent non-programmable calculators may be used.

Time allowed: 90 minutes
(Suggestion: Q1 54 minutes; Q2 36 minutes)
1. (45 marks) A music venue uses the following database table (shown with sample data) to record past, present and future gigs:

<table>
<thead>
<tr>
<th>num</th>
<th>bandname</th>
<th>gig_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Monster Fetish</td>
<td>2015-04-12</td>
</tr>
<tr>
<td>2</td>
<td>Thirst Beside Dent</td>
<td>2015-05-29</td>
</tr>
<tr>
<td>3</td>
<td>Pig of the Pushy Trucker</td>
<td>2015-06-02</td>
</tr>
<tr>
<td>4</td>
<td>Monster Fetish</td>
<td>2015-06-15</td>
</tr>
</tbody>
</table>

The venue owner desires a Python server-side program called `band_gigs.py` that implements a self-processing page with a sticky form as follows:

- If the user submits neither a band name nor a date, the program outputs the form.
- If the user submits a band name but no date, the program outputs an HTML table of all of the band’s gigs — past, present and future.
- If the user submits a date but no band name, the program outputs an HTML table of all gigs on or after that date.
- If the user submits both a band name and a date, the program outputs an HTML table of all of the band’s gigs on or after that date.
- If the user submits the name of a band that is not in the database, or submits an illegal date, the program outputs an informative error message.
- If the program encounters problems communicating with the database management system, the program outputs a message of apology.

Complete `band_gigs.py`, which is shown on the next page, by giving Python statements to replace the comment in the middle of the program:

Notes:

- You can check for a legal date using the `strptime` function. The following statement will succeed if variable `gig_date` contains any legal date in the correct format, otherwise it will raise a `ValueError` exception:
  
  ```python
  valid_date = strptime(gig_date, '%Y-%m-%d')
  ```

- You may assume that the following will create a connection to the database management software:
  
  ```python
  connection = db.connect('my_server', 'me', 'my_password', 'my_db')
  ```

- There is extra credit for guarding against JavaScript injection attacks and SQL injection attacks.
#!/usr/local/bin/python3

from cgitb import enable
enable()

from cgi import FieldStorage, escape
from time import strptime
import pymysql as db

# Your code would appear here

print(""
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Our Gigs</title>
</head>
<body>
  <form action="band_gigs.py" method="get">
    <label>Band name: </label>
    <input type="text" name="bandname" value="%s"><br>
    <label>Gig date (YYYY-MM-DD): </label>
    <input type="text" name="gig_date" value="%s"><br>
    <input type="submit" value="Search for gigs">
  </form>

%s
</body>
</html>"
% (bandname, gig_date, result))
2. (30 marks) A tailor has created a Web page, shown below. She desires a JavaScript client-side program. The user enters his height in feet and inches into the Web page’s first two textfields. When the user presses the Submit button, the program converts his height to metres and displays the result in the third textfield.

Write the JavaScript program.

Notes:

i) One foot is 12 inches, and 1 inch is 0.0254 metres.

ii) There is extra credit for validating the user’s input, and displaying any error messages in the span elements next to the textfields.

iii) You may assume that the following will test whether the string in some variable s converts to an integer:

```javascript
var number = Number(s);
if (String(number) === s) {
    // s converts to an integer
}
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Inches to metres</title>
    <script src="conversion.js"></script>
</head>
<body>

<form>
    <label>Feet: </label>
    <input type="text" name="feet" value="0" id="feet">
    <span id="message1"></span>

    <label>Inches: </label>
    <input type="text" name="inches" value="0" id="inches">
    <span id="message2"></span>

    <label>Metres: </label>
    <input type="text" name="metres" value="0" id="metres" disabled>

    <input type="submit">
</form>
</body>
</html>
```