PennyFlask/runserver.py (Page 1 of 1)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# runserver.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import sys
import penny

def main():

    if len(sys.argv) != 2:
        print('Usage: ' + sys.argv[0] + ' port', file=sys.stderr)
        sys.exit(1)

    try:
        port = int(sys.argv[1])
    except Exception:
        print('Port must be an integer.', file=sys.stderr)
        sys.exit(1)

    try:
        penny.app.run(host='0.0.0.0', port=port, debug=True)
    except Exception as ex:
        print(ex, file=sys.stderr)
        sys.exit(1)

if __name__ == '__main__':
    main()
```

PennyFlask/penny.py (Page 1 of 2)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# penny.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import html # html_code.escape() is used to thwart XSS attacks
import flask
import common
import database

#-----------------------------------------------------------------------

app = flask.Flask(__name__)

#-----------------------------------------------------------------------

@app.route('/', methods=['GET'])
@app.route('/index', methods=['GET'])
def index():

    html_code = ''
    html_code += '<!DOCTYPE html>'
    html_code += '<html>'
    html_code += '<head>'
    html_code += '<title>Penny.com</title>'
    html_code += '</head>'
    html_code += '<body>'
    html_code += common.get_header()
    html_code += '<br>'
    html_code += 'Click here to <a href="/searchform">begin</a>.<br>'
    html_code += '<br>'
    html_code += common.get_footer()
    html_code += '</body>'
    html_code += '</html>'

    response = flask.make_response(html_code)
    return response

#-----------------------------------------------------------------------

@app.route('/searchform', methods=['GET'])
def search_form():

    prev_author = flask.request.cookies.get('prev_author')
    if prev_author is None:
        prev_author = '(None)'

    html_code = ''
    html_code += '<!DOCTYPE html>'
    html_code += '<html>'
    html_code += '<head>'
    html_code += '<title>Penny.com</title>'
    html_code += '</head>'
    html_code += '<body>'
    html_code += common.get_header()
    html_code += '<h1>Author Search</h1>'
    html_code += '<form action="/searchresults" method="get">'
    html_code += 'Please enter an author name:'
    html_code += '<input type="text" name="author" autofocus>'
    html_code += '<input type="submit" value="Go">'
    html_code += '</form>'
    html_code += '<br>'
    html_code += '<br>'
```

PennyFlask/penny.py (Page 2 of 2)

```
    html_code += '<strong>Previous author search:</strong> '
    html_code += html.escape(prev_author)
    html_code += '<br>'
    html_code += '<br>'
    html_code += common.get_footer()
    html_code += '</body>'
    html_code += '</html>'

    response = flask.make_response(html_code)
    return response

#---------------------------------------------------------------------

@app.route('/searchresults', methods=['GET'])
def search_results():

    author = flask.request.args.get('author')
    if author is None:
        author = ''
    author = author.strip()

    if author == '':
        prev_author = '(None)'
        books = []
    else:
        prev_author = author
        books = database.get_books(author) # Exception handling omitted

    html_code = ''
    html_code += '<!DOCTYPE html>'
    html_code += '<html>'
    html_code += '<head>'
    html_code += '<title>Penny.com</title>'
    html_code += '</head>'
    html_code += '<body>'
    html_code += common.get_header()
    html_code += '<h1>Author Search Results</h1>'
    html_code += '<h2>Books by ' + html.escape(prev_author) + ':</h2>'

    if len(books) == 0:
        html_code += '(None)<br>'
    else:
        pattern = '%s: <strong>%s</strong>: %s<br>'
        for book in books:
            html_code += pattern % (book['isbn'], book['author'],
                book['title'])

    html_code += '<br>'
    html_code += '<br>'
    html_code += 'Click here to do another '
    html_code += '<a href="/searchform">author search</a>.'
    html_code += '<br>'
    html_code += '<br>'
    html_code += '<br>'
    html_code += '<br>'
    html_code += common.get_footer()
    html_code += '</body>'
    html_code += '</html>'

    response = flask.make_response(html_code)
    response.set_cookie('prev_author', prev_author)
    return response
```

blank (Page 1 of 1)

```
1: This page is intentionally blank.
```

PennyFlaskJinja/header.html (Page 1 of 1)

```
1: <hr>
2: Good {{ampm}} and welcome to <strong>Penny.com</strong>
3: <hr>
```

PennyFlaskJinja/footer.html (Page 1 of 1)

```
1: <hr>
2: Today is {{current_time}}.<br>
3: Created by <a href="https://www.cs.princeton.edu/~rdondero">
4: Bob Dondero</a>
5: <hr>
```

PennyFlaskJinja/index.html (Page 1 of 1)

```
 1: <!DOCTYPE html>
 2:  <html>
 3:    <head>
 4:       <title>Penny.com</title>
 5:       </head>
 6:    <body>
 7:       {% include 'header.html' %}
 8:       <br>
 9:       Click here to <a href="/searchform">begin</a>.<br>
10:       <br>
11:       {% include 'footer.html' %}
12:    </body>
13:  </html>
```

PennyFlaskJinja/searchform.html (Page 1 of 1)

```
 1: <!DOCTYPE html>
 2: <html>
 3:    <head>
 4:       <title>Penny.com</title>
 5:    </head>
 6:    <body>
 7:       {% include 'header.html' %}
 8:       <h1>Author Search</h1>
 9:       <form action="/searchresults" method="get">
10:          Please enter an author name:
11:          <input type="text" name="author" autofocus>
12:          <input type="submit" value="Go">
13:       </form>
14:       <br>
15:       <br>
16:       <strong>Previous author search: {{prev_author}}</strong>
17:       <br>
18:       <br>
19:       {% include 'footer.html' %}
20:    </body>
21: </html>
```

PennyFlaskJinja/searchresults.html (Page 1 of 1)

```
<!DOCTYPE html>
<html>
   <head>
      <title>Penny.com</title>
   </head>
   <body>
      {% include 'header.html' %}
      <h1>Author Search Results</h1>
      <h2>Books by {{author}}:</h2>
      {% if books|length == 0: %}
            (None)
      {% else %}
         {% for book in books: %}
             {{book['isbn']}}:
             <strong>{{book['author']}}</strong>:
             {{book['title']}}<br>
         {% endfor %}
      {% endif %}
      <br>
      <br>
      Click here to do another
      <a href="/searchform">author search</a>.
      <br>
      <br>
      <br>
      <br>
       {% include 'footer.html' %}
   </body>
</html>
```

blank (Page 1 of 1)

```
This page is intentionally blank.
```

PennyFlaskJinja/penny.py (Page 1 of 2)

```
#!/usr/bin/env python

#-----------------------------------------------------------------------
# penny.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import time
import flask
import database

#-----------------------------------------------------------------------

app = flask.Flask(__name__, template_folder='.')

#-----------------------------------------------------------------------

def get_ampm():
    if time.strftime('%p') == "AM":
        return 'morning'
    return 'afternoon'

def get_current_time():
    return time.asctime(time.localtime())

#-----------------------------------------------------------------------

@app.route('/', methods=['GET'])
@app.route('/index', methods=['GET'])
def index():

    html_code = flask.render_template('index.html',
        ampm=get_ampm(),
        current_time=get_current_time())
    response = flask.make_response(html_code)
    return response

#-----------------------------------------------------------------------

@app.route('/searchform', methods=['GET'])
def search_form():

    prev_author = flask.request.cookies.get('prev_author')
    if prev_author is None:
        prev_author = '(None)'

    html_code = flask.render_template('searchform.html',
        ampm=get_ampm(),
        current_time=get_current_time(),
        prev_author=prev_author)
    response = flask.make_response(html_code)
    return response

#-----------------------------------------------------------------------

@app.route('/searchresults', methods=['GET'])
def search_results():

    author = flask.request.args.get('author')
    if author is None:
        author = ''
    author = author.strip()

    if author == '':
        prev_author = '(None)'
```

PennyFlaskJinja/penny.py (Page 2 of 2)

```
        books = []
    else:
        prev_author = author
        books = database.get_books(author) # Exception handling omitted

    html_code = flask.render_template('searchresults.html',
        ampm=get_ampm(),
        current_time=get_current_time(),
        author=prev_author,
        books=books)
    response = flask.make_response(html_code)
    response.set_cookie('prev_author', prev_author)
    return response
```