

PennyFlask/runserver.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # runserver.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import penny
10:
11: def main():
12:
13:     if len(sys.argv) != 2:
14:         print(f'usage: {sys.argv[0]} port', file=sys.stderr)
15:         sys.exit(1)
16:
17:     try:
18:         port = int(sys.argv[1])
19:     except Exception:
20:         print(f'{sys.argv[0]}: Port must be an integer.',
21:               file=sys.stderr)
22:         sys.exit(1)
23:
24:     try:
25:         penny.app.run(host='0.0.0.0', port=port, debug=True)
26:     except Exception as ex:
27:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
28:         sys.exit(1)
29:
30: if __name__ == '__main__':
31:     main()

```

PennyFlask/penny.py (Page 1 of 3)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # penny.py
5: # Author: Bob Dondero
6: #-----
7:
8: import html # html_code.escape() is used to thwart XSS attacks
9: import flask
10: import common
11: import database
12:
13: #-----
14:
15: app = flask.Flask(__name__)
16:
17: #-----
18:
19: @app.route('/', methods=['GET'])
20: @app.route('/index', methods=['GET'])
21: def index():
22:
23:     html_code = f'''
24:         <!DOCTYPE html>
25:         <html>
26:             <head>
27:                 <title>Penny.com</title>
28:             </head>
29:             <body>
30:                 {common.get_header()}
31:                 <br>
32:                 Click to <a href="/searchform">begin</a>. <br>
33:                 <br>
34:                 {common.get_footer()}
35:             </body>
36:         </html>
37:     '''
38:
39:     response = flask.make_response(html_code)
40:     return response
41:
42: #-----
43:
44: @app.route('/searchform', methods=['GET'])
45: def search_form():
46:
47:     prev_author = flask.request.cookies.get('prev_author')
48:     if prev_author is None:
49:         prev_author = '(None)'
50:
51:     html_code = f'''
52:         <!DOCTYPE html>
53:         <html>
54:             <head>
55:                 <title>Penny.com</title>
56:             </head>
57:             <body>
58:                 {common.get_header()}
59:                 <h1>Author Search</h1>
60:                 <form action="/searchresults" method="get">
61:                     Please enter an author name:
62:                     <input type="text" name="author" autofocus>
63:                     <input type="submit" value="Go">
64:                 </form>
65:                 <br>

```

PennyFlask/penny.py (Page 2 of 3)

```

66:         <br>
67:         <strong>Previous author search:</strong>
68:         {html.escape(prev_author)}
69:         <br>
70:         <br>
71:         {common.get_footer()}
72:     </body>
73: </html>
74: '''
75:
76: response = flask.make_response(html_code)
77: return response
78:
79: #-----
80:
81: def convert_to_html(books):
82:
83:     if len(books) == 0:
84:         return '(None)'
85:     html_code = ''
86:     for book in books:
87:         html_code += f'''
88:             <strong>{html.escape(book['isbn'])}</strong>:
89:             {html.escape(book['author'])}:
90:             {html.escape(book['title'])}<br>
91:             '''
92:     return html_code
93:
94: #-----
95:
96: @app.route('/searchresults', methods=['GET'])
97: def search_results():
98:
99:     author = flask.request.args.get('author')
100:     if author is None:
101:         author = ''
102:     author = author.strip()
103:
104:     if author == '':
105:         prev_author = '(None)'
106:         books = []
107:     else:
108:         prev_author = author
109:         books = database.get_books(author) # Exception handling omitted
110:
111:     html_code = f'''
112:     <!DOCTYPE html>
113:     <html>
114:         <head>
115:             <title>Penny.com</title>
116:         </head>
117:         <body>
118:             {common.get_header()}
119:             <h1>Author Search Results</h1>
120:             <h2>Books by {html.escape(prev_author)}:</h2>
121:             {convert_to_html(books)}
122:             <br>
123:             <br>
124:             Click here to do another
125:             <a href="/searchform">author search</a>.
126:             <br>
127:             <br>
128:             <br>
129:             <br>
130:             {common.get_footer()}

```

PennyFlask/penny.py (Page 3 of 3)

```

131:         </body>
132:     </html>
133:     '''
134:
135:     response = flask.make_response(html_code)
136:     response.set_cookie('prev_author', prev_author)
137:     return response

```

PennyFlaskTemplates/templates.py (Page 1 of 2)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # templates.py
5: # Author: Bob Dondero
6: #-----
7:
8: index = '''
9: <!DOCTYPE html>
10: <html>
11:   <head>
12:     <title>Penny.com</title>
13:   </head>
14:   <body>
15:     <hr>
16:     Good {ampm} and welcome to <strong>Penny.com</strong>
17:     <hr>
18:     <br>
19:     Click to <a href="/searchform">begin</a>. <br>
20:     <br>
21:     <hr>
22:     Today is {current_time}. <br>
23:     Created by <a href="https://www.cs.princeton.edu/~rdondero">
24:     Bob Dondero</a>
25:     <hr>
26:   </body>
27: </html>
28: '''
29:
30: searchform = '''
31: <!DOCTYPE html>
32: <html>
33:   <head>
34:     <title>Penny.com</title>
35:   </head>
36:   <body>
37:     <hr>
38:     Good {ampm} and welcome to <strong>Penny.com</strong>
39:     <hr>
40:     <h1>Author Search</h1>
41:     <form action="/searchresults" method="get">
42:       Please enter an author name:
43:       <input type="text" name="author" autofocus>
44:       <input type="submit" value="Go">
45:     </form>
46:     <br>
47:     <br>
48:     <strong>Previous author search: {prev_author}</strong>
49:     <br>
50:     <br>
51:     <hr>
52:     Today is {current_time}. <br>
53:     Created by <a href="https://www.cs.princeton.edu/~rdondero">
54:     Bob Dondero</a>
55:     <hr>
56:   </body>
57: </html>
58: '''
59:
60: searchresults = '''
61: <!DOCTYPE html>
62: <html>
63:   <head>
64:     <title>Penny.com</title>
65:   </head>

```

PennyFlaskTemplates/templates.py (Page 2 of 2)

```

66:   <body>
67:     <hr>
68:     Good {ampm} and welcome to <strong>Penny.com</strong>
69:     <hr>
70:     <h1>Author Search Results</h1>
71:     <h2>Books by {author}:</h2>
72:     {books}
73:     <br>
74:     <br>
75:     Click here to do another
76:     <a href="/searchform">author search</a>.
77:     <br>
78:     <br>
79:     <hr>
80:     Today is {current_time}. <br>
81:     Created by <a href="https://www.cs.princeton.edu/~rdondero">
82:     Bob Dondero</a>
83:     <hr>
84:   </body>
85: </html>
86: '''

```

PennyFlaskTemplates/penny.py (Page 1 of 2)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # penny.py
5: # Author: Bob Dondero
6: #-----
7:
8: import time
9: import flask
10: import html
11: import database
12: import templates
13:
14: #-----
15:
16: app = flask.Flask(__name__)
17:
18: #-----
19:
20: def get_ampm():
21:     if time.strftime('%p') == 'AM':
22:         return 'morning'
23:     return 'afternoon'
24:
25: def get_current_time():
26:     return time.asctime(time.localtime())
27:
28: #-----
29:
30: @app.route('/', methods=['GET'])
31: @app.route('/index', methods=['GET'])
32: def index():
33:
34:     html_code = templates.index.format(
35:         ampm=get_ampm(),
36:         current_time=get_current_time())
37:     response = flask.make_response(html_code)
38:     return response
39:
40: #-----
41:
42: @app.route('/searchform', methods=['GET'])
43: def search_form():
44:
45:     prev_author = flask.request.cookies.get('prev_author')
46:     if prev_author is None:
47:         prev_author = '(None)'
48:
49:     html_code = templates.searchform.format(
50:         ampm=get_ampm(),
51:         current_time=get_current_time(),
52:         prev_author=prev_author)
53:
54:     response = flask.make_response(html_code)
55:     return response
56:
57: #-----
58:
59: def convert_to_html(books):
60:
61:     if len(books) == 0:
62:         return '(None)'
63:     html_code = ''
64:     for book in books:
65:         html_code += f''

```

PennyFlaskTemplates/penny.py (Page 2 of 2)

```

66:         <strong>{html.escape(book['isbn'])}</strong>:
67:         {html.escape(book['author'])}:
68:         {html.escape(book['title'])}<br>
69:         '''
70:     return html_code
71:
72: #-----
73:
74: @app.route('/searchresults', methods=['GET'])
75: def search_results():
76:
77:     author = flask.request.args.get('author')
78:     if author is None:
79:         author = ''
80:     author = author.strip()
81:
82:     if author == '':
83:         prev_author = '(None)'
84:         books = []
85:     else:
86:         prev_author = author
87:         books = database.get_books(author) # Exception handling omitted
88:
89:
90:     html_code = templates.searchresults.format(
91:         ampm=get_ampm(),
92:         current_time=get_current_time(),
93:         author=prev_author,
94:         books=convert_to_html(books))
95:     response = flask.make_response(html_code)
96:     response.set_cookie('prev_author', prev_author)
97:     return response

```

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/~rdonero">
4: Bob Donero</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 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)

```
1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:     {% include 'header.html' %}
8:     <h1>Author Search Results</h1>
9:     <h2>Books by {{author}}:</h2>
10:    {% if books|length == 0: %}
11:      (None)
12:    {% else %}
13:      {% for book in books: %}
14:        {{book['isbn']}}:
15:        <strong>{{book['author']}}</strong>:
16:        {{book['title']}}<br>
17:      {% endfor %}
18:    {% endif %}
19:    <br>
20:    <br>
21:    Click here to do another
22:    <a href="/searchform">author search</a>.
23:    <br>
24:    <br>
25:    {% include 'footer.html' %}
26:  </body>
27: </html>
```

blank (Page 1 of 1)

1: This page is intentionally blank.

PennyFlaskJinja/penny.py (Page 1 of 2)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # penny.py
5: # Author: Bob Dondero
6: #-----
7:
8: import time
9: import flask
10: import database
11:
12: #-----
13:
14: app = flask.Flask(__name__, template_folder='.')
15:
16: #-----
17:
18: def get_ampm():
19:     if time.strftime('%p') == 'AM':
20:         return 'morning'
21:     return 'afternoon'
22:
23: def get_current_time():
24:     return time.asctime(time.localtime())
25:
26: #-----
27:
28: @app.route('/', methods=['GET'])
29: @app.route('/index', methods=['GET'])
30: def index():
31:
32:     html_code = flask.render_template('index.html',
33:         ampm=get_ampm(),
34:         current_time=get_current_time())
35:     response = flask.make_response(html_code)
36:     return response
37:
38: #-----
39:
40: @app.route('/searchform', methods=['GET'])
41: def search_form():
42:
43:     prev_author = flask.request.cookies.get('prev_author')
44:     if prev_author is None:
45:         prev_author = '(None)'
46:
47:     html_code = flask.render_template('searchform.html',
48:         ampm=get_ampm(),
49:         current_time=get_current_time(),
50:         prev_author=prev_author)
51:     response = flask.make_response(html_code)
52:     return response
53:
54: #-----
55:
56: @app.route('/searchresults', methods=['GET'])
57: def search_results():
58:
59:     author = flask.request.args.get('author')
60:     if author is None:
61:         author = ''
62:     author = author.strip()
63:
64:     if author == '':
65:         prev_author = '(None)'

```

PennyFlaskJinja/penny.py (Page 2 of 2)

```

66:     books = []
67:     else:
68:         prev_author = author
69:         books = database.get_books(author) # Exception handling omitted
70:
71:     html_code = flask.render_template('searchresults.html',
72:         ampm=get_ampm(),
73:         current_time=get_current_time(),
74:         author=prev_author,
75:         books=books)
76:     response = flask.make_response(html_code)
77:     response.set_cookie('prev_author', prev_author)
78:     return response

```