

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.escape() is used to thwart XSS attacks
9: import time
10: import flask
11: import database
12:
13: #-----
14:
15: app = flask.Flask(__name__)
16:
17: #-----
18:
19: def get_header():
20:
21:     if time.strftime('%p') == 'AM':
22:         ampm = 'morning'
23:     else:
24:         ampm = 'afternoon'
25:
26:     html_str = f'''
27:     <hr>
28:     Good {ampm} and welcome to <strong>Penny.com</strong>
29:     <hr>
30:     '''
31:
32:     return html_str
33:
34: #-----
35:
36: def get_footer():
37:
38:     current_time = time.asctime(time.localtime())
39:
40:     html_str = f'''
41:     <hr>
42:     Today is {current_time}<br>
43:     Created by <a href="https://www.cs.princeton.edu/~rdondero">
44:     Bob Dondero</a>
45:     <hr>
46:     '''
47:
48:     return html_str
49:
50: #-----
51:
52: @app.route('/', methods=['GET'])
53: @app.route('/index', methods=['GET'])
54: def index():
55:
56:     html_code = f'''
57:     <!DOCTYPE html>
58:     <html>
59:     <head>
60:     <title>Penny.com</title>
61:     </head>
62:     <body>
63:     {get_header()}
64:     <br>
65:     Click to <a href="/searchform">begin</a>.<br>

```

PennyFlask/penny.py (Page 2 of 3)

```

66:         <br>
67:         {get_footer()}
68:     </body>
69: </html>
70: '''
71:
72: response = flask.make_response(html_code)
73: return response
74:
75: #-----
76:
77: @app.route('/searchform', methods=['GET'])
78: def search_form():
79:
80:     html_code = f'''
81:     <!DOCTYPE html>
82:     <html>
83:         <head>
84:             <title>Penny.com</title>
85:         </head>
86:         <body>
87:             {get_header()}
88:             <h1>Author Search</h1>
89:             <form action="/searchresults" method="get">
90:                 Please enter an author name:
91:                 <input type="text" name="author" autofocus>
92:                 <input type="submit" value="Go">
93:             </form>
94:             <br>
95:             <br>
96:             {get_footer()}
97:         </body>
98:     </html>
99:     '''
100:
101:     response = flask.make_response(html_code)
102:     return response
103:
104: #-----
105:
106: def convert_to_html(books):
107:
108:     if len(books) == 0:
109:         return '(None)'
110:     html_code = ''
111:     for book in books:
112:         html_code += f'''
113:         <strong>{html.escape(book['isbn'])}</strong>:
114:         {html.escape(book['author'])}:
115:         {html.escape(book['title'])}<br>
116:         '''
117:     return html_code
118:
119: #-----
120:
121: @app.route('/searchresults', methods=['GET'])
122: def search_results():
123:
124:     author = flask.request.args.get('author')
125:     if author is None:
126:         author = ''
127:     author = author.strip()
128:
129:     if author == '':
130:         author = '(None)'

```

PennyFlask/penny.py (Page 3 of 3)

```

131:     books = []
132: else:
133:     books = database.get_books(author) # Exception handling omitted
134:
135: html_code = f'''
136: <!DOCTYPE html>
137: <html>
138:     <head>
139:         <title>Penny.com</title>
140:     </head>
141:     <body>
142:         {get_header()}
143:         <h1>Author Search Results</h1>
144:         <h2>Books by {html.escape(author)}:</h2>
145:         {convert_to_html(books)}
146:         <br>
147:         <br>
148:         Click here to do another
149:         <a href="/searchform">author search</a>.
150:         <br>
151:         <br>
152:         <br>
153:         <br>
154:         {get_footer()}
155:     </body>
156: </html>
157:     '''
158:
159: response = flask.make_response(html_code)
160: 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/~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 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:     {% include 'footer.html' %}
17:   </body>
18: </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:     html_code = flask.render_template('searchform.html',
44:         ampm=get_ampm(),
45:         current_time=get_current_time())
46:     response = flask.make_response(html_code)
47:     return response
48:
49: #-----
50:
51: @app.route('/searchresults', methods=['GET'])
52: def search_results():
53:
54:     author = flask.request.args.get('author')
55:     if author is None:
56:         author = ''
57:     author = author.strip()
58:
59:     if author == '':
60:         author = '(None)'
61:         books = []
62:     else:
63:         books = database.get_books(author) # Exception handling omitted
64:
65:     html_code = flask.render_template('searchresults.html',

```

PennyFlaskJinja/penny.py (Page 2 of 2)

```

66:         ampm=get_ampm(),
67:         current_time=get_current_time(),
68:         author=author,
69:         books=books)
70:     response = flask.make_response(html_code)
71:     return response

```