

PennyOnePage/penny.py (Page 1 of 1)

```

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:     author = flask.request.args.get('author')
33:     if author is None:
34:         author = ''
35:     author = author.strip()
36:
37:     if author == '':
38:         books = []
39:     else:
40:         books = database.get_books(author) # Exception handling omitted
41:
42:     html = flask.render_template('index.html',
43:         ampm=get_ampm(),
44:         current_time=get_current_time(),
45:         author=author,
46:         books=books)
47:     response = flask.make_response(html)
48:     return response

```

PennyOnePage/index.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <html>
3:
4:     <head>
5:         <title>Penny.com</title>
6:     </head>
7:
8:     <body>
9:         <hr>
10:        Good {{ampm}} and welcome to <strong>Penny.com</strong>
11:        <hr>
12:
13:        <h1>Author Search</h1>
14:        <form action="/index" method="get">
15:            Please enter an author name:
16:            <input type="text" name="author" value="{{author}}" autoFocus>
17:            <input type="submit" value="Go">
18:        </form>
19:        <hr>
20:        {% for book in books: %}
21:            {{book['isbn']}}:
22:            <strong>{{book['author']}}:</strong>
23:            {{book['title']}}<br>
24:        {% endfor %}
25:
26:        <hr>
27:        Date and time: {{current_time}}.<br>
28:        Created by <a href="https://www.cs.princeton.edu/~rdondero">
29:        Bob Dondero</a>
30:        <hr>
31:    </body>
32:
33: </html>

```

PennyJavaScript/penny.py (Page 1 of 1)

```
1:#!/usr/bin/env python
2:
3: #-----
4: # penny.py
5: # Author: Bob Dondero
6: #-----
7:
8: import flask
9: import database
10:
11: #-----
12:
13: app = flask.Flask(__name__, template_folder='.')
14:
15: #-----
16:
17: @app.route('/', methods=['GET'])
18: @app.route('/index', methods=['GET'])
19: def index():
20:
21:     author = flask.request.args.get('author')
22:     if author is None:
23:         author = ''
24:     author = author.strip()
25:
26:     if author == '':
27:         books = []
28:     else:
29:         books = database.get_books(author) # Exception handling omitted
30:
31:     html_code = flask.render_template('index.html',
32:                                     author=author,
33:                                     books=books)
34:     response = flask.make_response(html_code)
35:     return response
```

blank (Page 1 of 1)

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

PennyJavaScript/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:
7:   <body>
8:
9:     <hr>
10:    Good <span id="ampmSpan"></span> and welcome to
11:    <strong>Penny.com</strong>
12:    <hr>
13:
14:    <h1>Author Search</h1>
15:    <form action="/index" method="get">
16:      Please enter an author name:
17:      <input type="text" name="author" value="{author}" autoFocus>
18:      <input type="submit" value="Go">
19:    </form>
20:    <hr>
21:    {% for book in books: %}
22:      {{book['isbn']}}:
23:      <strong>
24:        {{book['author']}}:
25:      </strong>
26:      {{book['title']}}<br>
27:    {% endfor %}
28:
29:    <hr>
30:    Date and time: <span id="datetimeSpan"></span><br>
31:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
32:    Bob Dondero</a>
33:    <hr>
34:
35:    <script>
36:
37:      'use strict';
38:
39:      function getAmPm() {
40:        let dateTime = new Date();
41:        let hours = dateTime.getHours();
42:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
43:        let ampMspan = document.getElementById('ampmSpan');
44:        ampMspan.innerHTML = amPm;
45:      }
46:
47:      function getDateTime() {
48:        let dateTime = new Date();
49:        let datetimeSpan = document.getElementById('datetimeSpan');
50:        datetimeSpan.innerHTML = dateTime.toLocaleString();
51:      }
52:
53:      function setupHeader() {
54:        getAmPm();
55:        let apInterval = window.setInterval(getAmPm, 1000);
56:        window.addEventListener('beforeunload',
57:          function(e) {window.clearInterval(apInterval);})
58:      }
59:
60:      function setupFooter() {
61:        getDateTime();
62:        let dtInterval = window.setInterval(getDateTime, 1000);
63:        window.addEventListener('beforeunload',
64:          function(e) {window.clearInterval(dtInterval);})
65:      }

```

PennyJavaScript/index.html (Page 2 of 2)

```

66:
67:      function setup() {
68:        setupHeader();
69:        setupFooter();
70:      }
71:
72:      document.addEventListener('DOMContentLoaded', setup);
73:
74:    </script>
75:  </body>
76: </html>

```

PennyAjax1/penny.py (Page 1 of 1)

```
1: #!/usr/bin/env python
2:
3: #-----
4: # penny.py
5: # Author: Bob Dondero
6: #-----
7:
8: import json
9: import flask
10: import database
11:
12: #-----
13:
14: app = flask.Flask(__name__)
15:
16: #-----
17:
18: @app.route('/', methods=['GET'])
19: @app.route('/index', methods=['GET'])
20: def index():
21:
22:     return flask.send_file('index.html')
23:
24: #-----
25:
26: @app.route('/searchresults', methods=['GET'])
27: def search_results():
28:
29:     author = flask.request.args.get('author')
30:     if author is None:
31:         author = ''
32:     author = author.strip()
33:
34:     if author == '':
35:         books = []
36:     else:
37:         books = database.get_books(author) # Exception handling omitted
38:
39:     json_doc = json.dumps(books)
40:     response = flask.make_response(json_doc)
41:     response.headers['Content-Type'] = 'application/json'
42:     return response
```

blank (Page 1 of 1)

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

PennyAjax1/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:     <hr>
8:     Good <span id="ampmSpan"></span> and welcome to
9:     <strong>Penny.com</strong>
10:    <hr>
11:    <h1>Author Search</h1>
12:    Please enter an author name:
13:    <input type="text" id="authorInput" autoFocus>
14:    <hr>
15:    <div id="resultsDiv"></div>
16:    <hr>
17:    Date and time: <span id="datetimeSpan"></span><br>
18:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
19:    Bob Dondero</a>
20:    <hr>
21:    <script>
22:      'use strict';
23:
24:      function getAmPm() {
25:        let dateTime = new Date();
26:        let hours = dateTime.getHours();
27:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
28:        let ampMspan = document.getElementById('ampmSpan');
29:        ampMspan.innerHTML = amPm;
30:      }
31:
32:      function getDateTime() {
33:        let dateTime = new Date();
34:        let datetimeSpan =
35:          document.getElementById('datetimeSpan');
36:        datetimeSpan.innerHTML = dateTime.toLocaleString();
37:      }
38:
39:      function escape(s) {
40:        s = s.replace('&', '&amp;');
41:        s = s.replace('<', '&lt;');
42:        s = s.replace('>', '&gt;');
43:        s = s.replace('"', '&quot;');
44:        s = s.replace("'", '&apos;');
45:        return s;
46:      }
47:
48:      function convertToHtml(books) {
49:        let html = '';
50:        for (let book of books) {
51:          html += escape(book.isbn) + ' : ' ;
52:          html += '<strong>';
53:          html += escape(book.author);
54:          html += '</strong> : ' ;
55:          html += escape(book.title) + '<br>';
56:        }
57:        return html;
58:      }
59:    </script>
60:  </body>
61: </html>

```

PennyAjax1/index.html (Page 2 of 2)

```

62:    function handleResponse() {
63:      if (this.status !== 200) {
64:        alert('Error: Failed to fetch data from server');
65:        return;
66:      }
67:      let books = JSON.parse(this.response);
68:      let html = convertToHtml(books);
69:      let resultsDiv = document.getElementById('resultsDiv');
70:      resultsDiv.innerHTML = html;
71:    }
72:
73:    function handleError() {
74:      alert('Error: Failed to fetch data from server');
75:    }
76:
77:    function getResults() {
78:      let authorInput = document.getElementById('authorInput');
79:      let author = authorInput.value;
80:      let encodedAuthor = encodeURIComponent(author);
81:      let url = '/searchresults?author=' + encodedAuthor;
82:      let request = new XMLHttpRequest();
83:      request.onload = handleResponse;
84:      request.onerror = handleError;
85:      request.open('GET', url);
86:      request.send();
87:    }
88:
89:    function setupHeader() {
90:      getAmPm();
91:      let apInterval = window.setInterval(getAmPm, 1000);
92:      window.addEventListener('beforeunload',
93:        function(e) {window.clearInterval(apInterval);})
94:    }
95:
96:    function setupFooter() {
97:      getDateTime();
98:      let dtInterval = window.setInterval(getDateTime, 1000);
99:      window.addEventListener('beforeunload',
100:        function(e) {window.clearInterval(dtInterval);})
101:    }
102:
103:    function setup() {
104:      setupHeader();
105:      setupFooter();
106:      let authorInput = document.getElementById('authorInput');
107:      authorInput.addEventListener('input', getResults);
108:    }
109:
110:    document.addEventListener('DOMContentLoaded', setup);
111:  </script>
112: </body>
113: </html>

```

PennyAjax2/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:     <hr>
8:     Good <span id="ampmSpan"></span> and welcome to
9:     <strong>Penny.com</strong>
10:    <hr>
11:    <h1>Author Search</h1>
12:    Please enter an author name:
13:    <input type="text" id="authorInput" autoFocus>
14:    <hr>
15:    <div id="resultsDiv"></div>
16:    <hr>
17:    Date and time: <span id="datetimeSpan"></span><br>
18:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
19:    Bob Dondero</a>
20:    <hr>
21:    <script>
22:      'use strict';
23:
24:      function getAmPm() {
25:        let dateTime = new Date();
26:        let hours = dateTime.getHours();
27:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
28:        let ampMspan = document.getElementById('ampmSpan');
29:        ampMspan.innerHTML = amPm;
30:      }
31:
32:      function getDateTime() {
33:        let dateTime = new Date();
34:        let datetimeSpan =
35:          document.getElementById('datetimeSpan');
36:        datetimeSpan.innerHTML = dateTime.toLocaleString();
37:      }
38:
39:      function escape(s) {
40:        s = s.replace('&', '&amp;');
41:        s = s.replace('<', '&lt;');
42:        s = s.replace('>', '&gt;');
43:        s = s.replace('"', '&quot;');
44:        s = s.replace("'", '&apos;');
45:        return s;
46:      }
47:
48:      function convertToHtml(books) {
49:        let html = '';
50:        for (let book of books) {
51:          html += escape(book.isbn) + ' : ' ;
52:          html += ' <strong>';
53:          html += escape(book.author);
54:          html += ' </strong> : ' ;
55:          html += escape(book.title) + ' <br>';
56:        }
57:        return html;
58:      }
59:    </script>
60:  </body>
61: </html>

```

PennyAjax2/index.html (Page 2 of 2)

```

62:     function handleResponse() {
63:       if (this.status !== 200) {
64:         alert('Error: Failed to fetch data from server');
65:         return;
66:       }
67:       let books = JSON.parse(this.response);
68:       let html = convertToHtml(books);
69:       let resultsDiv = document.getElementById('resultsDiv');
70:       resultsDiv.innerHTML = html;
71:     }
72:
73:     function handleError() {
74:       alert('Error: Failed to fetch data from server');
75:     }
76:
77:     let request = null;
78:
79:     function getResults() {
80:       let authorInput = document.getElementById('authorInput');
81:       let author = authorInput.value;
82:       let encodedAuthor = encodeURIComponent(author);
83:       let url = '/searchresults?author=' + encodedAuthor;
84:       if (request !== null)
85:         request.abort();
86:       request = new XMLHttpRequest();
87:       request.onload = handleResponse;
88:       request.onerror = handleError;
89:       request.open('GET', url);
90:       request.send();
91:     }
92:
93:     function setupHeader() {
94:       getAmPm();
95:       let apInterval = window.setInterval(getAmPm, 1000);
96:       window.addEventListener('beforeunload',
97:         function(e) {window.clearInterval(apInterval);})
98:     }
99:
100:    function setupFooter() {
101:      getDateTime();
102:      let dtInterval = window.setInterval(getDateTime, 1000);
103:      window.addEventListener('beforeunload',
104:        function(e) {window.clearInterval(dtInterval);})
105:    }
106:
107:    function setup() {
108:      setupHeader();
109:      setupFooter();
110:      let authorInput = document.getElementById('authorInput');
111:      authorInput.addEventListener('input', getResults);
112:    }
113:
114:    document.addEventListener('DOMContentLoaded', setup);
115:  </script>
116: </body>
117: </html>

```

PennyAjax3/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:     <hr>
8:     Good <span id="ampmSpan"></span> and welcome to
9:     <strong>Penny.com</strong>
10:    <hr>
11:    <h1>Author Search</h1>
12:    Please enter an author name:
13:    <input type="text" id="authorInput" autoFocus>
14:    <hr>
15:    <div id="resultsDiv"></div>
16:    <hr>
17:    Date and time: <span id="datetimeSpan"></span><br>
18:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
19:    Bob Dondero</a>
20:    <hr>
21:    <script>
22:      'use strict';
23:
24:      function getAmPm() {
25:        let dateTime = new Date();
26:        let hours = dateTime.getHours();
27:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
28:        let ampSpan = document.getElementById('ampmSpan');
29:        ampSpan.innerHTML = amPm;
30:      }
31:
32:      function getDateTime() {
33:        let dateTime = new Date();
34:        let datetimeSpan =
35:          document.getElementById('datetimeSpan');
36:        datetimeSpan.innerHTML = dateTime.toLocaleString();
37:      }
38:
39:      function escape(s) {
40:        s = s.replace('&', '&amp;');
41:        s = s.replace('<', '&lt;');
42:        s = s.replace('>', '&gt;');
43:        s = s.replace('"', '&quot;');
44:        s = s.replace("'", '&apos;');
45:        return s;
46:      }
47:
48:      function convertToHtml(books) {
49:        let html = '';
50:        for (let book of books) {
51:          html += escape(book.isbn) + ' : ' ;
52:          html += ' <strong>';
53:          html += escape(book.author);
54:          html += ' </strong> : ' ;
55:          html += escape(book.title) + ' <br>';
56:        }
57:        return html;
58:      }
59:    </script>
60:  }
61: </body>
62: </html>

```

PennyAjax3/index.html (Page 2 of 2)

```

66:    function handleResponse() {
67:      if (this.status !== 200) {
68:        alert('Error: Failed to fetch data from server');
69:        return;
70:      }
71:      let books = JSON.parse(this.response);
72:      let html = convertToHtml(books);
73:      let resultsDiv = document.getElementById('resultsDiv');
74:      resultsDiv.innerHTML = html;
75:    }
76:
77:    function handleError() {
78:      alert('Error: Failed to fetch data from server');
79:    }
80:
81:    let request = null;
82:
83:    function getResults() {
84:      let authorInput = document.getElementById('authorInput');
85:      let author = authorInput.value;
86:      let encodedAuthor = encodeURIComponent(author);
87:      let url = '/searchresults?author=' + encodedAuthor;
88:      if (request !== null)
89:        request.abort();
90:      request = new XMLHttpRequest();
91:      request.onload = handleResponse;
92:      request.onerror = handleError;
93:      request.open('GET', url);
94:      request.send();
95:    }
96:
97:    let timer = null;
98:
99:    function debouncedGetResults() {
100:      clearTimeout(timer);
101:      timer = window.setTimeout(getResults, 500);
102:    }
103:
104:    function setupHeader() {
105:      getAmPm();
106:      let apInterval = window.setInterval(getAmPm, 1000);
107:      window.addEventListener('beforeunload',
108:        function(e) {window.clearInterval(apInterval);})
109:    }
110:
111:    function setupFooter() {
112:      getDateTime();
113:      let dtInterval = window.setInterval(getDateTime, 1000);
114:      window.addEventListener('beforeunload',
115:        function(e) {window.clearInterval(dtInterval);})
116:    }
117:
118:    function setup() {
119:      setupHeader();
120:      setupFooter();
121:      let authorInput = document.getElementById('authorInput');
122:      authorInput.addEventListener('input', debouncedGetResults);
123:    }
124:
125:    document.addEventListener('DOMContentLoaded', setup);
126:
127:  </script>
128: </body>
129: </html>

```

PennyAjax4/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:
7:   <body>
8:
9:     <hr>
10:    Good <span id="ampmSpan"></span> and welcome to
11:    <strong>Penny.com</strong>
12:    <hr>
13:
14:    <h1>Author Search</h1>
15:    Please enter an author name:
16:    <input type="text" id="authorInput" autoFocus>
17:    <hr>
18:    <div id="resultsDiv"></div>
19:
20:    <hr>
21:    Date and time: <span id="datetimeSpan"></span><br>
22:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
23:    Bob Dondero</a>
24:    <hr>
25:
26:    <script src=
27:      "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
28:    </script>
29:
30:    <!-- <script src="/static/mustache.min.js"></script> -->
31:
32:    <script>
33:
34:      'use strict';
35:
36:      function getAmPm() {
37:        let dateTime = new Date();
38:        let hours = dateTime.getHours();
39:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
40:        let ampmSpan = document.getElementById('ampmSpan');
41:        ampmSpan.innerHTML = amPm;
42:      }
43:
44:      function getDateTime() {
45:        let dateTime = new Date();
46:        let datetimeSpan =
47:          document.getElementById('datetimeSpan');
48:        datetimeSpan.innerHTML = dateTime.toLocaleString();
49:      }
50:
51:      function convertToHtml(books) {
52:        let template = `
53:          {{#books}}
54:            {{isbn}}:
55:            <strong>{{author}}</strong>;
56:            {{title}}
57:            <br>
58:          {{/books}}
59:        `;
60:        let map = {books: books};
61:        let html = Mustache.render(template, map);
62:        return html;
63:      }
64:
65:      function handleResponse() {

```

PennyAjax4/index.html (Page 2 of 2)

```

66:        if (this.status !== 200) {
67:          alert('Error: Failed to fetch data from server');
68:          return;
69:        }
70:        let books = JSON.parse(this.response);
71:        let html = convertToHtml(books);
72:        let resultsDiv = document.getElementById('resultsDiv');
73:        resultsDiv.innerHTML = html;
74:      }
75:
76:      function handleError() {
77:        alert('Error: Failed to fetch data from server');
78:      }
79:
80:      let request = null;
81:
82:      function getResults() {
83:        let authorInput = document.getElementById('authorInput');
84:        let author = authorInput.value;
85:        let encodedAuthor = encodeURIComponent(author);
86:        let url = '/searchresults?author=' + encodedAuthor;
87:        if (request !== null)
88:          request.abort();
89:        request = new XMLHttpRequest();
90:        request.onload = handleResponse;
91:        request.onerror = handleError;
92:        request.open('GET', url);
93:        request.send();
94:      }
95:
96:      let timer = null;
97:
98:      function debouncedGetResults() {
99:        clearTimeout(timer);
100:        timer = window.setTimeout(getResults, 500);
101:      }
102:
103:      function setupHeader() {
104:        getAmPm();
105:        let apInterval = window.setInterval(getAmPm, 1000);
106:        window.addEventListener('beforeunload',
107:          function(e) {window.clearInterval(apInterval);})
108:      }
109:
110:      function setupFooter() {
111:        getDateTime();
112:        let dtInterval = window.setInterval(getDateTime, 1000);
113:        window.addEventListener('beforeunload',
114:          function(e) {window.clearInterval(dtInterval);})
115:      }
116:
117:      function setup() {
118:        setupHeader();
119:        setupFooter();
120:        let authorInput = document.getElementById('authorInput');
121:        authorInput.addEventListener('input', debouncedGetResults);
122:      }
123:
124:      document.addEventListener('DOMContentLoaded', setup);
125:
126:    </script>
127:  </body>
128: </html>

```

PennyAjaxFetch1/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:     <hr>
8:     Good <span id="ampmSpan"></span> and welcome to
9:     <strong>Penny.com</strong>
10:    <hr>
11:    <h1>Author Search</h1>
12:    Please enter an author name:
13:    <input type="text" id="authorInput" autoFocus>
14:    <hr>
15:    <div id="resultsDiv"></div>
16:    <hr>
17:    Date and time: <span id="datetimeSpan"></span><br>
18:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
19:    Bob Dondero</a>
20:    <hr>
21:    <script src=
22:    "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
23:    </script>
24:    <script>
25:      'use strict';
26:
27:      function getAmPm() {
28:        let dateTime = new Date();
29:        let hours = dateTime.getHours();
30:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
31:        let ampMspan = document.getElementById('ampmSpan');
32:        ampMspan.innerHTML = amPm;
33:      }
34:
35:      function getDateTime() {
36:        let dateTime = new Date();
37:        let datetimeSpan =
38:        document.getElementById('datetimeSpan');
39:        datetimeSpan.innerHTML = dateTime.toLocaleString();
40:      }
41:
42:      function getText(response) {
43:        if (! response.ok)
44:          throw new Error();
45:        return response.text();
46:      }
47:
48:      function convertToHtml(books) {
49:        let template = `
50:          {{#books}}
51:            {{isbn}}:
52:            <strong>{{author}}</strong>:
53:            {{title}}
54:            <br>
55:          {{/books}}
56:          `;
57:        let map = {books: books};
58:        let html = Mustache.render(template, map);

```

PennyAjaxFetch1/index.html (Page 2 of 2)

```

59:        return html;
60:      }
61:
62:      function updateResultsDiv(text) {
63:        let books = JSON.parse(text);
64:        let html = convertToHtml(books);
65:        let resultsDiv = document.getElementById('resultsDiv');
66:        resultsDiv.innerHTML = html;
67:      }
68:
69:      function handleError(err) {
70:        if (err.name !== 'AbortError')
71:          alert('Error: Failed to fetch data from server');
72:      }
73:
74:      let controller = null;
75:
76:      function getResults() {
77:        let authorInput = document.getElementById('authorInput');
78:        let author = authorInput.value;
79:        let encodedAuthor = encodeURIComponent(author);
80:        let url = '/searchresults?author=' + encodedAuthor;
81:        if (controller !== null)
82:          controller.abort();
83:        controller = new AbortController();
84:        fetch(url, {signal: controller.signal})
85:          .then(getText)
86:          .then(updateResultsDiv)
87:          .catch(handleError);
88:      }
89:
90:      let timer = null;
91:
92:      function debouncedGetResults() {
93:        clearTimeout(timer);
94:        timer = setTimeout(getResults, 500);
95:      }
96:
97:      function setupHeader() {
98:        getAmPm();
99:        let apInterval = window.setInterval(getAmPm, 1000);
100:        window.addEventListener('beforeunload',
101:        function(e) {window.clearInterval(apInterval);})
102:      }
103:
104:      function setupFooter() {
105:        getDateTime();
106:        let dtInterval = window.setInterval(getDateTime, 1000);
107:        window.addEventListener('beforeunload',
108:        function(e) {window.clearInterval(dtInterval);})
109:      }
110:
111:      function setup() {
112:        setupHeader();
113:        setupFooter();
114:        let authorInput = document.getElementById('authorInput');
115:        authorInput.addEventListener('input', debouncedGetResults);
116:      }
117:
118:      document.addEventListener('DOMContentLoaded', setup);
119:
120:    </script>
121:  </body>
122: </html>

```

PennyAjaxFetch2/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:     <hr>
8:     Good <span id="ampmSpan"></span> and welcome to
9:     <strong>Penny.com</strong>
10:    <hr>
11:    <h1>Author Search</h1>
12:    Please enter an author name:
13:    <input type="text" id="authorInput" autoFocus>
14:    <hr>
15:    <div id="resultsDiv"></div>
16:    <hr>
17:    Date and time: <span id="datetimeSpan"></span><br>
18:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
19:    Bob Dondero</a>
20:    <hr>
21:    <script src=
22:    "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
23:    </script>
24:    <script>
25:      'use strict';
26:
27:      function getAmPm() {
28:        let dateTime = new Date();
29:        let hours = dateTime.getHours();
30:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
31:        let ampMspan = document.getElementById('ampmSpan');
32:        ampMspan.innerHTML = amPm;
33:      }
34:
35:      function getDateTime() {
36:        let dateTime = new Date();
37:        let datetimeSpan =
38:        document.getElementById('datetimeSpan');
39:        datetimeSpan.innerHTML = dateTime.toLocaleString();
40:      }
41:
42:      function convertToHtml(books) {
43:        let template = `
44:          {{#books}}
45:            {{isbn}}:
46:            <strong>{{author}}</strong>:
47:            <strong>{{title}}</strong>
48:            <br>
49:          {{/books}}
50:        `;
51:        let map = {books: books};
52:        let html = Mustache.render(template, map);
53:        return html;
54:      }
55:
56:      let controller = null;
57:
58:      async function getResults() {

```

PennyAjaxFetch2/index.html (Page 2 of 2)

```

59:        let authorInput = document.getElementById('authorInput');
60:        let author = authorInput.value;
61:        let encodedAuthor = encodeURIComponent(author);
62:        let url = '/searchresults?author=' + encodedAuthor;
63:        if (controller !== null)
64:          controller.abort();
65:        controller = new AbortController();
66:        try {
67:          let response =
68:            await fetch(url, {signal: controller.signal});
69:          if (!response.ok)
70:            throw new Error();
71:          let text = await response.text();
72:          let books = JSON.parse(text);
73:          let html = convertToHtml(books);
74:          let resultsDiv = document.getElementById('resultsDiv');
75:          resultsDiv.innerHTML = html;
76:        }
77:        catch (err) {
78:          if (err.name !== 'AbortError')
79:            alert('Error: Failed to fetch data from server');
80:        }
81:
82:        let timer = null;
83:
84:        function debouncedGetResults() {
85:          clearTimeout(timer);
86:          timer = setTimeout(getResults, 500);
87:        }
88:
89:        function setupHeader() {
90:          getAmPm();
91:          let apInterval = window.setInterval(getAmPm, 1000);
92:          window.addEventListener('beforeunload',
93:            function(e) {window.clearInterval(apInterval);})
94:        }
95:
96:        function setupFooter() {
97:          getDateTime();
98:          let dtInterval = window.setInterval(getDateTime, 1000);
99:          window.addEventListener('beforeunload',
100:            function(e) {window.clearInterval(dtInterval);})
101:        }
102:
103:        function setup() {
104:          setupHeader();
105:          setupFooter();
106:          let authorInput = document.getElementById('authorInput');
107:          authorInput.addEventListener('input', debouncedGetResults);
108:        }
109:
110:        document.addEventListener('DOMContentLoaded', setup);
111:
112:      </script>
113:    </body>
114:  </html>

```