

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

```

PennyJavaScript/index.html (Page 2 of 2)

```

60:
61:
62:
63:
64:
65:
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:
12:    <h1>Author Search</h1>
13:    Please enter an author name:
14:    <input type="text" id="authorInput" autoFocus>
15:    <hr>
16:    <div id="resultsDiv"></div>
17:
18:    <hr>
19:    Date and time: <span id="datetimeSpan"></span><br>
20:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
21:    Bob Dondero</a>
22:    <hr>
23:
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 escape(s) {
43:        s = s.replace('&', '&amp;');
44:        s = s.replace('<', '&lt;');
45:        s = s.replace('>', '&gt;');
46:        s = s.replace('"', '&quot;');
47:        s = s.replace("'", '&apos;');
48:        return s;
49:      }
50:
51:      function convertToHtml(books) {
52:        let html = '';
53:        for (let book of books) {
54:          html += escape(book.isbn) + ': ';
55:          html += '<strong>';
56:          html += escape(book.author);
57:          html += '</strong>: ';
58:          html += escape(book.title) + '<br>';
59:        }
60:        return html;
61:      }
62:
63:    </script>
64:  </body>
65: </html>

```

PennyAjax1/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:     function getResults() {
82:       let authorInput = document.getElementById('authorInput');
83:       let author = authorInput.value;
84:       let encodedAuthor = encodeURIComponent(author);
85:       let url = '/searchresults?author=' + encodedAuthor;
86:       let 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:
116:  </script>
117: </body>
118: </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:
12:    <h1>Author Search</h1>
13:    Please enter an author name:
14:    <input type="text" id="authorInput" autoFocus>
15:    <hr>
16:    <div id="resultsDiv"></div>
17:
18:    <hr>
19:    Date and time: <span id="datetimeSpan"></span><br>
20:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
21:    Bob Dondero</a>
22:    <hr>
23:
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 escape(s) {
43:        s = s.replace('&', '&amp;');
44:        s = s.replace('<', '&lt;');
45:        s = s.replace('>', '&gt;');
46:        s = s.replace('"', '&quot;');
47:        s = s.replace("'", '&apos;');
48:        return s;
49:      }
50:
51:      function convertToHtml(books) {
52:        let html = '';
53:        for (let book of books) {
54:          html += escape(book.isbn) + ': ';
55:          html += '<strong>';
56:          html += escape(book.author);
57:          html += '</strong>: ';
58:          html += escape(book.title) + '<br>';
59:        }
60:        return html;
61:      }
62:
63:    </script>
64:  }
65:

```

PennyAjax2/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:    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', getResults);
116:    }
117:
118:    document.addEventListener('DOMContentLoaded', setup);
119:
120:  </script>
121: </body>
122: </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:
12:    <h1>Author Search</h1>
13:    Please enter an author name:
14:    <input type="text" id="authorInput" autoFocus>
15:    <hr>
16:    <div id="resultsDiv"></div>
17:
18:    <hr>
19:    Date and time: <span id="datetimeSpan"></span><br>
20:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
21:    Bob Dondero</a>
22:    <hr>
23:
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 escape(s) {
43:        s = s.replace('&', '&amp;');
44:        s = s.replace('<', '&lt;');
45:        s = s.replace('>', '&gt;');
46:        s = s.replace('"', '&quot;');
47:        s = s.replace("'", '&apos;');
48:        return s;
49:      }
50:
51:      function convertToHtml(books) {
52:        let html = '';
53:        for (let book of books) {
54:          html += escape(book.isbn) + ': ';
55:          html += '<strong>';
56:          html += escape(book.author);
57:          html += '</strong>: ';
58:          html += escape(book.title) + '<br>';
59:        }
60:        return html;
61:      }
62:    </script>
63:  }
64: </body>
65: </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:   <body>
7:     <hr>
8:     Good <span id="ampmSpan"></span> and welcome to
9:     <strong>Penny.com</strong>
10:    <hr>
11:
12:    <h1>Author Search</h1>
13:    Please enter an author name:
14:    <input type="text" id="authorInput" autoFocus>
15:    <hr>
16:    <div id="resultsDiv"></div>
17:
18:    <hr>
19:    Date and time: <span id="datetimeSpan"></span><br>
20:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
21:    Bob Dondero</a>
22:    <hr>
23:
24:    <script src=
25:    "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
26:    </script>
27:
28:    <!-- <script src="/static/mustache.min.js"></script> -->
29:
30:    <script>
31:      'use strict';
32:
33:      function getAmPm() {
34:        let dateTime = new Date();
35:        let hours = dateTime.getHours();
36:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
37:        let ampMspan = document.getElementById('ampmSpan');
38:        ampMspan.innerHTML = amPm;
39:      }
40:
41:      function getDateTime() {
42:        let dateTime = new Date();
43:        let datetimeSpan =
44:        document.getElementById('datetimeSpan');
45:        datetimeSpan.innerHTML = dateTime.toLocaleString();
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);
59:        return html;
60:      }
61:
62:      function handleResponse() {

```

PennyAjax4/index.html (Page 2 of 2)

```

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:      let timer = null;
94:
95:      function debouncedGetResults() {
96:        clearTimeout(timer);
97:        timer = window.setTimeout(getResults, 500);
98:      }
99:
100:      function setupHeader() {
101:        getAmPm();
102:        let apInterval = window.setInterval(getAmPm, 1000);
103:        window.addEventListener('beforeunload',
104:          function(e) {window.clearInterval(apInterval);})
105:      }
106:
107:      function setupFooter() {
108:        getDateTime();
109:        let dtInterval = window.setInterval(getDateTime, 1000);
110:        window.addEventListener('beforeunload',
111:          function(e) {window.clearInterval(dtInterval);})
112:      }
113:
114:      function setup() {
115:        setupHeader();
116:        setupFooter();
117:        let authorInput = document.getElementById('authorInput');
118:        authorInput.addEventListener('input', debouncedGetResults);
119:      }
120:
121:      document.addEventListener('DOMContentLoaded', setup);
122:
123:    </script>
124:  </body>
125: </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:
12:    <h1>Author Search</h1>
13:    Please enter an author name:
14:    <input type="text" id="authorInput" autoFocus>
15:    <hr>
16:    <div id="resultsDiv"></div>
17:
18:    <hr>
19:    Date and time: <span id="datetimeSpan"></span><br>
20:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
21:    Bob Dondero</a>
22:    <hr>
23:
24:    <script src=
25:    "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
26:    </script>
27:
28:    <script>
29:
30:      'use strict';
31:
32:      function getAmPm() {
33:        let dateTime = new Date();
34:        let hours = dateTime.getHours();
35:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
36:        let ampMspan = document.getElementById('ampmSpan');
37:        ampMspan.innerHTML = amPm;
38:      }
39:
40:      function getDateTime() {
41:        let dateTime = new Date();
42:        let datetimeSpan =
43:        document.getElementById('datetimeSpan');
44:        datetimeSpan.innerHTML = dateTime.toLocaleString();
45:      }
46:
47:      function usingResponseGetText(response) {
48:        if (! response.ok)
49:          throw new Error();
50:        return response.text();
51:      }
52:
53:      function convertToHtml(books) {
54:        let template = `
55:          {{#books}}
56:            {{isbn}}:
57:            <strong>{{author}}</strong>:
58:            {{title}}
59:            <br>
60:          {{/books}}
61:        `;
62:        let map = {books: books};
63:        let html = Mustache.render(template, map);

```

PennyAjaxFetch1/index.html (Page 2 of 2)

```

64:        return html;
65:      }
66:
67:      function usingTextUpdateResultsDiv(text) {
68:        let books = JSON.parse(text);
69:        let html = convertToHtml(books);
70:        let resultsDiv = document.getElementById('resultsDiv');
71:        resultsDiv.innerHTML = html;
72:      }
73:
74:      function handleError(err) {
75:        if (err.name !== 'AbortError')
76:          alert('Error: Failed to fetch data from server');
77:      }
78:
79:      let controller = null;
80:
81:      function getResults() {
82:        let authorInput = document.getElementById('authorInput');
83:        let author = authorInput.value;
84:        let encodedAuthor = encodeURIComponent(author);
85:        let url = '/searchresults?author=' + encodedAuthor;
86:        if (controller !== null)
87:          controller.abort();
88:        controller = new AbortController();
89:        fetch(url, {signal: controller.signal})
90:          .then(usingResponseGetText)
91:          .then(usingTextUpdateResultsDiv)
92:          .catch(handleError);
93:      }
94:
95:      let timer = null;
96:
97:      function debouncedGetResults() {
98:        clearTimeout(timer);
99:        timer = setTimeout(getResults, 500);
100:      }
101:
102:      function setupHeader() {
103:        getAmPm();
104:        let apInterval = window.setInterval(getAmPm, 1000);
105:        window.addEventListener('beforeunload',
106:          function(e) {window.clearInterval(apInterval);})
107:      }
108:
109:      function setupFooter() {
110:        getDateTime();
111:        let dtInterval = window.setInterval(getDateTime, 1000);
112:        window.addEventListener('beforeunload',
113:          function(e) {window.clearInterval(dtInterval);})
114:      }
115:
116:      function setup() {
117:        setupHeader();
118:        setupFooter();
119:        let authorInput = document.getElementById('authorInput');
120:        authorInput.addEventListener('input', debouncedGetResults);
121:      }
122:
123:      document.addEventListener('DOMContentLoaded', setup);
124:
125:    </script>
126:  </body>
127: </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:
12:    <h1>Author Search</h1>
13:    Please enter an author name:
14:    <input type="text" id="authorInput" autoFocus>
15:    <hr>
16:    <div id="resultsDiv"></div>
17:
18:    <hr>
19:    Date and time: <span id="datetimeSpan"></span><br>
20:    Created by <a href="https://www.cs.princeton.edu/~rdondero">
21:    Bob Dondero</a>
22:    <hr>
23:
24:    <script src=
25:    "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
26:    </script>
27:
28:    <script>
29:
30:      'use strict';
31:
32:      function getAmPm() {
33:        let dateTime = new Date();
34:        let hours = dateTime.getHours();
35:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
36:        let ampMspan = document.getElementById('ampmSpan');
37:        ampMspan.innerHTML = amPm;
38:      }
39:
40:      function getDateTime() {
41:        let dateTime = new Date();
42:        let datetimeSpan =
43:        document.getElementById('datetimeSpan');
44:        datetimeSpan.innerHTML = dateTime.toLocaleString();
45:      }
46:
47:      function convertToHtml(books) {
48:        let template = `
49:          {{#books}}
50:            {{isbn}}:
51:            <strong>{{author}}</strong>:
52:            {{title}}
53:            <br>
54:          {{/books}}
55:        `;
56:        let map = {books: books};
57:        let html = Mustache.render(template, map);
58:        return html;
59:      }
60:
61:      async function fetchBooks(url) {
62:        try {
63:          let response =

```

PennyAjaxFetch2/index.html (Page 2 of 2)

```

64:          await fetch(url, {signal: controller.signal});
65:          if (!response.ok)
66:            throw new Error();
67:          let text = await response.text();
68:          let books = JSON.parse(text);
69:          let html = convertToHtml(books);
70:          let resultsDiv = document.getElementById('resultsDiv');
71:          resultsDiv.innerHTML = html;
72:        }
73:      }
74:    }
75:    catch (err) {
76:      if (err.name !== 'AbortError')
77:        alert('Error: Failed to fetch data from server');
78:    }
79:  }
80:
81:  let controller = 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 (controller !== null)
89:      controller.abort();
90:    controller = new AbortController();
91:    fetchBooks(url);
92:  }
93:
94:  let timer = null;
95:
96:  function debouncedGetResults() {
97:    clearTimeout(timer);
98:    timer = setTimeout(getResults, 500);
99:  }
100:
101:  function setupHeader() {
102:    getAmPm();
103:    let apInterval = window.setInterval(getAmPm, 1000);
104:    window.addEventListener('beforeunload',
105:      function(e) {window.clearInterval(apInterval);})
106:  }
107:
108:  function setupFooter() {
109:    getDateTime();
110:    let dtInterval = window.setInterval(getDateTime, 1000);
111:    window.addEventListener('beforeunload',
112:      function(e) {window.clearInterval(dtInterval);})
113:  }
114:
115:  function setup() {
116:    setupHeader();
117:    setupFooter();
118:    let authorInput = document.getElementById('authorInput');
119:    authorInput.addEventListener('input', debouncedGetResults);
120:  }
121:
122:  document.addEventListener('DOMContentLoaded', setup);
123:
124: </script>
125: </body>
126: </html>

```