PennyViewport/penny.py (Page 1 of 1)

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

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

import json
import flask
import database

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

app = flask.Flask(__name__)

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

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

    return flask.send_file('index.html')

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

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

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

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

    json_doc = json.dumps(books)
    response = flask.make_response(json_doc)
    response.headers['Content-Type'] = 'application/json'
    return response
```

blank (Page 1 of 1)

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

PennyViewport/index.html (Page 1 of 2)

```
<!DOCTYPE html>
<html>
   <head>
      <title>Penny.com</title>

      <meta name="viewport"
         content="width=device-width, initial-scale=1">
   </head>
   <body>
      <hr>
      Good <span id="ampmSpan"></span> and welcome to
      <strong>Penny.com</strong>
      <hr>

      <h1>Author Search</h1>
      Please enter an author name:
      <input type="text" id="authorInput" autoFocus>
      <hr>

      <div id="resultsDiv"></div>

      <hr>
      Date and time: <span id="datetimeSpan"></span><br>
      Created by <a href="https://www.cs.princeton.edu/~rdondero">
      Bob Dondero</a>
      <hr>

      <script src=
         "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
      </script>

      <script>

         'use strict';

         function getAmPm() {
            let dateTime = new Date();
            let hours = dateTime.getHours();
            let amPm = (hours < 12) ? 'morning' : 'afternoon';
            let ampmSpan = document.getElementById('ampmSpan');
            ampmSpan.innerHTML = amPm;
         }

         function getDateTime() {
            let dateTime = new Date();
            let datetimeSpan =
               document.getElementById('datetimeSpan');
            datetimeSpan.innerHTML = dateTime.toLocaleString();
         }

         function convertToHtml(books) {
            let template = `
               {{#books}}
                  {{isbn}}:
                  <strong>{{author}}</strong>:
                  {{title}}
                  <br>
               {{/books}}
            `;
            let map = {books: books};
            let html = Mustache.render(template, map);
            return html;
         }

         function handleResponse() {
```

PennyViewport/index.html (Page 2 of 2)

```
            if (this.status !== 200) {
               alert('Error: Failed to fetch data from server');
               return;
            }
            let books = JSON.parse(this.response);
            let html = convertToHtml(books);
            let resultsDiv = document.getElementById('resultsDiv');
            resultsDiv.innerHTML = html;
         }

         function handleError() {
            alert('Error: Failed to fetch data from server');
         }

         let request = null;

         function getResults() {
            let authorInput = document.getElementById('authorInput');
            let author = authorInput.value;
            let encodedAuthor = encodeURIComponent(author);
            let url = '/searchresults?author=' + encodedAuthor;
            if (request !== null)
               request.abort();
            request = new XMLHttpRequest();
            request.onload = handleResponse;
            request.onerror = handleError;
            request.open('GET', url);
            request.send();
         }

         let timer = null;

         function debouncedGetResults() {
            clearTimeout(timer);
            timer = setTimeout(getResults, 500);
         }

         function setupHeader() {
            getAmPm();
            let apInterval = window.setInterval(getAmPm, 1000);
            window.addEventListener('beforeunload',
               function(e) {window.clearInterval(apInterval);})
         }

         function setupFooter() {
            getDateTime();
            let dtInterval = window.setInterval(getDateTime, 1000);
            window.addEventListener('beforeunload',
               function(e) {window.clearInterval(dtInterval);})
         }

         function setup() {
            setupHeader();
            setupFooter();
            let authorInput = document.getElementById('authorInput');
            authorInput.addEventListener('input', debouncedGetResults);
         }

         document.addEventListener('DOMContentLoaded', setup);

      </script>
   </body>
</html>
```

PennyCss/static/penny.css (Page 1 of 1)

```
/*--------------------------------------------------------------------*/
/* penny.css                                                          */
/* Author: Bob Dondero                                                */
/*--------------------------------------------------------------------*/

/* Element rules */

body {font-family: Arial, Helvetica, sans-serif; font-size: 16px;}

input[type="text"] {font-size: 16px;}

/*--------------------------------------------------------------------*/

/* Class rules */

.header, .footer {
   background-color: #295078; /* Venice blue */
   text-align: center;
   color: white;
}

.headerlink, .footerlink {color: white;}

.grayborder {
   border-style: solid;
   border-color: gray;
   border-width: 1px;
}

.grid-container {
   display: grid;
   grid-template-columns: 1fr 3fr;
   gap: 16px;
}
```


PennyCss/index.html (Page 1 of 3)

```
<!DOCTYPE html>
<html>

   <head>
      <title>Penny.com</title>
      <meta name="viewport"
         content="width=device-width, initial-scale=1">
      <link rel="stylesheet" href="static/penny.css">
   </head>

   <body>
      <div class="header">
         <h2>Penny.com</h2>
         <h3>Good <span id="ampmSpan"></span> and welcome to
            Penny.com</h3>
      </div>

      <br>

      <div class="grid-container">
         <div><strong>Author name:</strong></div>
         <div><input type="text" id="authorInput" autoFocus></div>
         <div><strong>Books:</strong></div>
         <div class="grayborder" id="resultsDiv"></div>
      </div>

      <br>

      <div class="footer">
         Date and time: <span id="datetimeSpan"></span><br>
         Created by <a class="footerlink"
         href="https://www.cs.princeton.edu/~rdondero">
         Bob Dondero</a>
      </div>

      <script src=
         "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
      </script>

      <script>
         'use strict';

         function getAmPm() {
            let dateTime = new Date();
            let hours = dateTime.getHours();
            let amPm = (hours < 12) ? 'morning' : 'afternoon';
            let ampmSpan = document.getElementById('ampmSpan');
            ampmSpan.innerHTML = amPm;
         }

         function getDateTime() {
            let dateTime = new Date();
            let datetimeSpan =
               document.getElementById('datetimeSpan');
            datetimeSpan.innerHTML = dateTime.toLocaleString();
         }

         function convertToHtml(books) {
            let template = `
               {{#books}}
                  {{isbn}}:
                  <strong>{{author}}</strong>:
                  {{title}}
                  <br>
               {{/books}}
```

PennyCss/index.html (Page 2 of 3)

```
66:             `;
67:             let map = {books: books};
68:             let html = Mustache.render(template, map);
69:             return html;
70:          }
71:
72:          function handleResponse() {
73:             if (this.status !== 200) {
74:                alert('Error: Failed to fetch data from server');
75:                return;
76:             }
77:             let books = JSON.parse(this.response);
78:             let html = convertToHtml(books);
79:             let resultsDiv = document.getElementById('resultsDiv');
80:             resultsDiv.innerHTML = html;
81:          }
82:
83:          function handleError() {
84:             alert('Error: Failed to fetch data from server');
85:          }
86:
87:          let request = null;
88:
89:          function getResults() {
90:             let authorInput = document.getElementById('authorInput');
91:             let author = authorInput.value;
92:             let encodedAuthor = encodeURIComponent(author);
93:             let url = '/searchresults?author=' + encodedAuthor;
94:             if (request !== null)
95:                request.abort();
96:             request = new XMLHttpRequest();
97:             request.onload = handleResponse;
98:             request.onerror = handleError;
99:             request.open('GET', url);
100:            request.send();
101:         }
102:
103:         let timer = null;
104:
105:         function debouncedGetResults() {
106:            clearTimeout(timer);
107:            timer = setTimeout(getResults, 500);
108:         }
109:
110:         function setupHeader() {
111:            getAmPm();
112:            let apInterval = window.setInterval(getAmPm, 1000);
113:            window.addEventListener('beforeunload',
114:               function(e) {window.clearInterval(apInterval);})
115:         }
116:
117:         function setupFooter() {
118:            getDateTime();
119:            let dtInterval = window.setInterval(getDateTime, 1000);
120:            window.addEventListener('beforeunload',
121:               function(e) {window.clearInterval(dtInterval);})
122:         }
123:
124:         function setup() {
125:            setupHeader();
126:            setupFooter();
127:            let authorInput = document.getElementById('authorInput');
128:            authorInput.addEventListener('input', debouncedGetResults);
129:         }
130:
```

PennyCss/index.html (Page 3 of 3)

```
131:            document.addEventListener('DOMContentLoaded', setup);
132:         </script>
133:      </body>
134: </html>
```

PennyResponsive/static/penny.css (Page 1 of 1)

```
/*--------------------------------------------------------------------*/
/* penny.css                                                          */
/* Author: Bob Dondero                                                */
/*--------------------------------------------------------------------*/

/* Element rules */

body {font-family:Arial, Helvetica, sans-serif; font-size:16px;}

input[type="text"] {font-size:16px;}

/*--------------------------------------------------------------------*/

/* Class rules */

.header, .footer {
   background-color: #295078;
   text-align: center;
   color: white;}

.headerlink, .footerlink {color: white;}

.grayborder {
   border-style: solid;
   border-color: gray;
   border-width: 1px;
}

.grid-container {
   display: grid;
   grid-template-columns: 1fr 3fr;
   gap: 16px;
   color: blue;
}

/*--------------------------------------------------------------------*/

/* Media query */

@media (max-width: 600px) {
   h2 {display: none;}

   .grid-container {
      display: grid;
      grid-template-columns: 1fr;
      gap: 16px;
   }
}
```

blank (Page 1 of 1)

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

PennyBootstrap/index.html (Page 1 of 3)

```
<!DOCTYPE html>
<html>
   <head>
      <title>Penny.com</title>

      <meta name="viewport"
         content="width=device-width, initial-scale=1">

      <link rel="stylesheet" href=
         "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstra
p.min.css">

      <style>
         .header, .footer {background-color:#295078; color:white}
         .headerlink, .footerlink {color:white}
      </style>

   </head>

   <body>
      <div class="container-fluid sticky-top header">
         <center>
            <h2 class="d-none d-sm-block">Penny.com</h2>
            <h3>Good <span id="ampmSpan"></span> and welcome to
            Penny.com</h3>
         </center>
      </div>

      <br>

      <div class="container-fluid">
         <div class="row">
            <div class="col-sm-3"><h5>Author name:</h5></div>
            <div class="col-sm-9">
               <input type="text" class="form-control"
                  id="authorInput" autoFocus>
            </div>
         </div>
         <br>
         <div class="row">
            <div class="col-sm-3"><h5>Books:</h5></div>
            <div class="col-sm-9">
               <div class="border" id="resultsDiv"></div>
            </div>
         </div>
      </div>

      <br>

      <div class="container-fluid sticky-bottom footer">
         <center>
            Date and time: <span id="datetimeSpan"></span><br>
            Created by <a class="footerlink"
            href="http://www.cs.princeton.edu/~rdondero">
            Bob Dondero</a>
         </center>
      </div>

      <script src=
         "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
      </script>

      <script>

         'use strict';
```

PennyBootstrap/index.html (Page 2 of 3)

```

         function getAmPm() {
            let dateTime = new Date();
            let hours = dateTime.getHours();
            let amPm = (hours < 12) ? 'morning' : 'afternoon';
            let ampmSpan = document.getElementById('ampmSpan');
            ampmSpan.innerHTML = amPm;
         }

         function getDateTime() {
            let dateTime = new Date();
            let datetimeSpan =
               document.getElementById('datetimeSpan');
            datetimeSpan.innerHTML = dateTime.toLocaleString();
         }

         function convertToHtml(books) {
            let template = `
               {{#books}}
                  {{isbn}}:
                  <strong>{{author}}</strong>:
                  {{title}}
                  <br>
               {{/books}}
            `;
            let map = {books: books};
            let html = Mustache.render(template, map);
            return html;
         }

         function handleResponse() {
            if (this.status !== 200) {
               alert('Error: Failed to fetch data from server');
               return;
            }
            let books = JSON.parse(this.response);
            let html = convertToHtml(books);
            let resultsDiv = document.getElementById('resultsDiv');
            resultsDiv.innerHTML = html;
         }

         function handleError() {
            alert('Error: Failed to fetch data from server');
         }

         let request = null;

         function getResults() {
            let authorInput = document.getElementById('authorInput');
            let author = authorInput.value;
            let encodedAuthor = encodeURIComponent(author);
            let url = '/searchresults?author=' + encodedAuthor;
            if (request !== null)
               request.abort();
            request = new XMLHttpRequest();
            request.onload = handleResponse;
            request.onerror = handleError;
            request.open('GET', url);
            request.send();
         }

         let timer = null;

         function debouncedGetResults() {
            clearTimeout(timer);
```

PennyBootstrap/index.html (Page 3 of 3)

```
130:             timer = setTimeout(getResults, 500);
131:          }
132:
133:          function setupHeader() {
134:             getAmPm();
135:             let apInterval = window.setInterval(getAmPm, 1000);
136:             window.addEventListener('beforeunload',
137:                function(e) {window.clearInterval(apInterval);})
138:          }
139:
140:          function setupFooter() {
141:             getDateTime();
142:             let dtInterval = window.setInterval(getDateTime, 1000);
143:             window.addEventListener('beforeunload',
144:                function(e) {window.clearInterval(dtInterval);})
145:          }
146:
147:          function setup() {
148:             setupHeader();
149:             setupFooter();
150:             let authorInput = document.getElementById('authorInput');
151:             authorInput.addEventListener('input', debouncedGetResults);
152:          }
153:
154:          document.addEventListener('DOMContentLoaded', setup);
155:
156:       </script>
157:    </body>
158: </html>
```

blank (Page 1 of 1)

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

PennyModal/indexRaw.html (Page 1 of 3)

```
<!DOCTYPE html>
<html>
   <head>
      <title>Penny.com</title>

      <meta name="viewport"
         content="width=device-width, initial-scale=1">

      <script src=
         "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap
.bundle.min.js">
      </script>

      <link rel="stylesheet" href=
         "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstra
p.min.css">

      <style>
         .header, .footer {background-color:#295078; color:white}
         .headerlink, .footerlink {color:white}
      </style>

   </head>

   <body>

      <div class="modal" id="booksModal">
         <div class="modal-dialog">
            <div class="modal-content">
               <div class="modal-header">
                  <h4 class="modal-title">Books</h4>
                  <button class="btn-close" data-bs-dismiss="modal">
                  </button>
               </div>
               <div class="modal-body" id="booksModalBody">
                  Modal body; to be replaced.
               </div>
               <div class="modal-footer">
                  <button class="btn btn-danger"
                     data-bs-dismiss="modal">
                     Close
                  </button>
               </div>
            </div>
         </div>
      </div>

      <div class="container-fluid header">
         <center>
            <h2 class="d-none d-sm-block">Penny.com</h2>
            <h3>Good <span id="ampmSpan"></span> and welcome to
            Penny.com</h3>
         </center>
      </div>

      <br>

      <div class="container-fluid">
         <div class="row">
            <div class="col-sm-2"><h5>Author name:</h5></div>
            <div class="col-sm-8">
               <input type="text" class="form-control"
                  id="authorInput" autoFocus>
            </div>
            <div class="col-sm-2">
```

PennyModal/indexRaw.html (Page 2 of 3)

```
               <button id="submitbutton">Submit</button>
            </div>
         </div>
      </div>

      <br>

      <div class="container-fluid footer">
         <center>
            Date and time: <span id="datetimeSpan"></span><br>
            Created by <a class="footerlink"
            href="http://www.cs.princeton.edu/~rdondero">
            Bob Dondero</a>
         </center>
      </div>

      <script src=
         "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
      </script>

      <script>

         'use strict';

         function getAmPm() {
            let dateTime = new Date();
            let hours = dateTime.getHours();
            let amPm = (hours < 12) ? 'morning' : 'afternoon';
            let ampmSpan = document.getElementById('ampmSpan');
            ampmSpan.innerHTML = amPm;
         }

         function getDateTime() {
            let dateTime = new Date();
            let datetimeSpan =
               document.getElementById('datetimeSpan');
            datetimeSpan.innerHTML = dateTime.toLocaleString();
         }

         function convertToHtml(books) {
            let template = `
               {{#books}}
                  {{isbn}}:
                  <strong>{{author}}</strong>:
                  {{title}}
                  <br>
               {{/books}}
            `;
            let map = {books: books};
            let html = Mustache.render(template, map);
            return html;
         }

         function handleResponse() {
            if (this.status !== 200) {
               alert('Error: Failed to fetch data from server');
               return;
            }
            let books = JSON.parse(this.response);
            let html = convertToHtml(books);
            let booksModalBodyNode =
               document.getElementById('booksModalBody');
            booksModalBodyNode.innerHTML = html;
            let booksModalNode =
               document.getElementById('booksModal');
```

PennyModal/indexRaw.html (Page 3 of 3)

```
129:             let booksModal = new bootstrap.Modal(booksModalNode);
130:             booksModal.show();
131:          }
132:
133:          function handleError() {
134:             alert('Error: Failed to fetch data from server');
135:          }
136:
137:          let request = null;
138:
139:          function getResults() {
140:             let authorInput = document.getElementById('authorInput');
141:             let author = authorInput.value;
142:             let encodedAuthor = encodeURIComponent(author);
143:             let url = '/searchresults?author=' + encodedAuthor;
144:             if (request !== null)
145:                request.abort();
146:             request = new XMLHttpRequest();
147:             request.onload = handleResponse;
148:             request.onerror = handleError;
149:             request.open('GET', url);
150:             request.send();
151:          }
152:
153:          function setupHeader() {
154:             getAmPm();
155:             let apInterval = window.setInterval(getAmPm, 1000);
156:             window.addEventListener('beforeunload',
157:                function(e) {window.clearInterval(apInterval);})
158:          }
159:
160:          function setupFooter() {
161:             getDateTime();
162:             let dtInterval = window.setInterval(getDateTime, 1000);
163:             window.addEventListener('beforeunload',
164:                function(e) {window.clearInterval(dtInterval);})
165:          }
166:
167:          function setup() {
168:             setupHeader();
169:             setupFooter();
170:             let submitButton = document.getElementById('submitbutton');
171:             submitButton.addEventListener('click', getResults);
172:          }
173:
174:          document.addEventListener('DOMContentLoaded', setup);
175:
176:       </script>
177:    </body>
178: </html>
```

blank (Page 1 of 1)

1: This page is intentionally blank.

PennyModal/indexJQuery.html (Page 1 of 3)

```
<!DOCTYPE html>
<html>
   <head>
      <title>Penny.com</title>

      <meta name="viewport"
         content="width=device-width, initial-scale=1">

      <script src=
         "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap
.bundle.min.js">
      </script>

      <link rel="stylesheet" href=
         "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstra
p.min.css">

      <style>
         .header, .footer {background-color:#295078; color:white}
         .headerlink, .footerlink {color:white}
      </style>

   </head>

   <body>

      <div class="modal" id="booksModal">
         <div class="modal-dialog">
            <div class="modal-content">
               <div class="modal-header">
                  <h4 class="modal-title">Books</h4>
                  <button class="btn-close" data-bs-dismiss="modal">
                  </button>
               </div>
               <div class="modal-body" id="booksModalBody">
                  Modal body; to be replaced.
               </div>
               <div class="modal-footer">
                  <button class="btn btn-danger"
                     data-bs-dismiss="modal">
                     Close
                  </button>
               </div>
            </div>
         </div>
      </div>

      <div class="container-fluid, header">
         <center>
            <h2 class="d-none d-sm-block">Penny.com</h2>
            <h3>Good <span id="ampmSpan"></span> and welcome to
            Penny.com</h3>
         </center>
      </div>

      <br>

      <div class="container-fluid">
         <div class="row">
            <div class="col-sm-2"><h5>Author name:</h5></div>
            <div class="col-sm-8">
               <input type="text" class="form-control"
                  id="authorInput" autoFocus>
            </div>
            <div class="col-sm-2">
```

PennyModal/indexJQuery.html (Page 2 of 3)

```
               <button id="submitbutton">Submit</button>
            </div>
         </div>
      </div>

      <br>

      <div class="container-fluid, footer">
         <center>
            Date and time: <span id="datetimeSpan"></span><br>
            Created by <a class="footerlink"
            href="http://www.cs.princeton.edu/~rdondero">
            Bob Dondero</a>
         </center>
      </div>

      <script src=
         "https://cdn.jsdelivr.net/npm/jquery@3.7.1/dist/jquery.min.js">
      </script>

      <script src=
          "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
      </script>

      <script>

         'use strict';

         function getAmPm() {
            let dateTime = new Date();
            let hours = dateTime.getHours();
            let amPm = 'morning';
            if (hours >= 12)
               amPm = 'afternoon';
            $('#ampmSpan').html(amPm);
         }

         function getDateTime() {
            let dateTime = new Date();
            $('#datetimeSpan').html(dateTime.toLocaleString());
         }

         function convertToHtml(books) {
            let template = `
               {{#books}}
                  {{isbn}}:
                  <strong>{{author}}</strong>:
                  {{title}}
                  <br>
               {{/books}}
            `;
            let map = {books: books};
            let html = Mustache.render(template, map);
            return html;
         }

         function handleResponse(books) {
            let html = convertToHtml(books);
            $('#booksModalBody').html(html);
            $('#booksModal').modal('show');
         }

         function handleError(request) {
            if (request.statusText !== 'abort')
               alert('Error: Failed to fetch data from server');
```

PennyModal/indexJQuery.html (Page 3 of 3)

```
129:          }
130:
131:          let request = null;
132:
133:          function getResults() {
134:             let author = $('#authorInput').val();
135:             let encodedAuthor = encodeURIComponent(author);
136:             let url = '/searchresults?author=' + encodedAuthor;
137:             if (request != null)
138:                request.abort();
139:             let requestData = {
140:                type: 'GET',
141:                url: url,
142:                success: handleResponse,
143:                error: handleError
144:             };
145:             request = $.ajax(requestData);
146:          }
147:
148:          function setupHeader() {
149:             getAmPm();
150:             let apInterval = window.setInterval(getAmPm, 1000);
151:             $(window).on('beforeunload',
152:                function(e) {window.clearInterval(apInterval);})
153:          }
154:
155:          function setupFooter() {
156:             getDateTime();
157:             let dtInterval = window.setInterval(getDateTime, 1000);
158:             $(window).on('beforeunload',
159:                function(e) {window.clearInterval(dtInterval);})
160:          }
161:
162:          function setup() {
163:             setupHeader();
164:             setupFooter();
165:             $('#submitbutton').on('click', getResults);
166:          }
167:
168:          $('document').ready(setup);
169:
170:       </script>
171:    </body>
172: </html>
```

blank (Page 1 of 1)

1: This page is intentionally blank.

PennyModal/indexReact.html (Page 1 of 4)

```
<!DOCTYPE html>
<html>
   <head>
      <meta name="viewport"
         content="width=device-width, initial-scale=1">

      <!-- For development: -->
      <!--
      <script src=
         "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.development.js">
      </script>
      <script src=
         "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.development.js">
      </script>
      -->

      <!-- For production: -->
      <script src=
         "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production.min.js">
      </script>
      <script src=
         "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.production.min.js">
      </script>

      <script src=
         "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min.js">
      </script>

      <script src=
         "https://cdn.jsdelivr.net/npm/react-bootstrap@2.10.0/dist/react-bootstrap.min.js">
      </script>

      <link rel="stylesheet" href=
         "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">

      <style>
         .header, .footer {background-color:#295078; color:white}
         .headerlink, .footerlink {color:white}
      </style>

      <title>Penny.com</title>
   </head>

   <body>

      <div id="root"></div>

      <script type="text/babel">

         'use strict';

         //---------------------------------------------------------------

         function PennyHeader() {
            const [datetime, setDatetime] = React.useState(new Date());

            function updateHeader() {
               let apInterval = window.setInterval(
```

PennyModal/indexReact.html (Page 2 of 4)

```
                  () => {setDatetime(new Date());},
                  1000
               );
               return () => {window.clearInterval(apInterval);};
            }
            React.useEffect(updateHeader, []);

            let hours = datetime.getHours();
            let ampm = (hours < 12) ? 'morning' : 'afternoon';
            return (
               <ReactBootstrap.Container fluid className="header">
                  <center>
                     <h2>Penny.com</h2>
                     <h3>Good {ampm} and welcome to Penny.com</h3>
                  </center>
               </ReactBootstrap.Container>
            );
         }

         //---------------------------------------------------------------

         function BooksModal(props) {
            let books = props.b;
            let show = props.s;
            let showCallback = props.scb;

            if (books === null) return;

            function handleClose() {
               showCallback(false);
            }

            return (
               <ReactBootstrap.Modal show={show}
                  onHide={handleClose}>
                  <ReactBootstrap.Modal.Header closeButton>
                     <ReactBootstrap.Modal.Title>
                        Books
                     </ReactBootstrap.Modal.Title>
                  </ReactBootstrap.Modal.Header>
                  <ReactBootstrap.Modal.Body>
                     {books.map((book) => (
                        <div key={book.isbn}>
                           {book.isbn}: 
                           <strong>{book.author}</strong>: 
                           {book.title}<br />
                        </div>
                     ))}
                  </ReactBootstrap.Modal.Body>
                  <ReactBootstrap.Modal.Footer>
                     <ReactBootstrap.Button variant="danger"
                        onClick={handleClose}>
                        Close
                     </ReactBootstrap.Button>
                  </ReactBootstrap.Modal.Footer>
               </ReactBootstrap.Modal>
            );
         }

         //---------------------------------------------------------------

         function PennySearch() {
            const [books, setBooks] = React.useState(null);
            const [show, setShow] = React.useState(false);

```

PennyModal/indexReact.html (Page 3 of 4)

```
124:             const inputRef = React.useRef();
125:
126:             let request = null;
127:
128:             function fetchBooks(author) {
129:                function handleResponse() {
130:                   if (this.status !== 200) {
131:                      alert('Error: Failed to fetch data from server');
132:                      return;
133:                   }
134:                   let books = JSON.parse(this.response);
135:                   setBooks(books);
136:                   setShow(true);
137:                }
138:
139:                function handleError() {
140:                   alert('Error: Failed to fetch data from server');
141:                }
142:
143:                let encodedAuthor = encodeURIComponent(author);
144:                let url = '/searchresults?author=' + encodedAuthor;
145:                if (request !== null)
146:                   request.abort();
147:                request = new XMLHttpRequest();
148:                request.onload = handleResponse;
149:                request.onerror = handleError;
150:                request.open('GET', url);
151:                request.send();
152:             }
153:
154:             return (
155:                <div>
156:                   <BooksModal s={show} scb={setShow} b={books} />
157:                   <ReactBootstrap.Container fluid>
158:                      <ReactBootstrap.Row>
159:                         <ReactBootstrap.Col sm={2}>
160:                            <h5>Author name:</h5>
161:                         </ReactBootstrap.Col>
162:                         <ReactBootstrap.Col sm={8}>
163:                            <ReactBootstrap.Form.Control
164:                               type="text" ref={inputRef} autoFocus
165:                            />
166:                         </ReactBootstrap.Col>
167:                         <ReactBootstrap.Col sm={2}>
168:                            <button onClick={ (event) => {
169:                                fetchBooks(inputRef.current.value);
170:                            }} >
171:                            Submit
172:                            </button>
173:                         </ReactBootstrap.Col>
174:                      </ReactBootstrap.Row>
175:                   </ReactBootstrap.Container>
176:                </div>
177:             );
178:          }
179:
180:          //-------------------------------------------------------------
181:
182:          function PennyFooter() {
183:             const [datetime, setDatetime] = React.useState(new Date());
184:
185:             function updateFooter() {
186:                let dtInterval = window.setInterval(
187:                   () => {setDatetime(new Date());},
188:                   1000
```

PennyModal/indexReact.html (Page 4 of 4)

```
189:                );
190:                return () => {window.clearInterval(dtInterval);};
191:             }
192:             React.useEffect(updateFooter, []);
193:
194:             return (
195:                <ReactBootstrap.Container fluid className="footer">
196:                   <center>
197:                      Date and time: {datetime.toLocaleString()}
198:                      <br />
199:                      Created by 
200:                      <a class="footerlink"
201:                      href="https://www.cs.princeton.edu/~rdondero">
202:                      Bob Dondero</a>
203:                   </center>
204:                </ReactBootstrap.Container>
205:             );
206:          }
207:
208:          //-------------------------------------------------------------
209:
210:          function App() {
211:             return (
212:                <div>
213:                   <PennyHeader />
214:                   <br />
215:                   <PennySearch />
216:                   <br />
217:                   <PennyFooter />
218:                </div>
219:             );
220:          }
221:
222:          //-------------------------------------------------------------
223:
224:          let domRoot = document.getElementById('root');
225:          let reactRoot = ReactDOM.createRoot(domRoot);
226:          reactRoot.render(
227:             <React.StrictMode>
228:                <App />
229:             </React.StrictMode>
230:          );
231:
232:       </script>
233:    </body>
234: </html>
235:
```