PennyReact/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.
```

PennyReact/index.html (Page 1 of 3)

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

   <body>

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

      <!-- 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 type="text/babel">

         'use strict';

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

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

            function updateHeader() {
               let apInterval = window.setInterval(
                  () => {setDatetime(new Date());},
                  1000
               );
               return () => {window.clearInterval(apInterval);};
            }

            React.useEffect(updateHeader, []);

            let hours = datetime.getHours();
            let ampm = (hours < 12) ? 'morning' : 'afternoon';
            return (
               <div>
                  <hr />
                  Good {ampm} and welcome to Penny.com
                  <hr />
               </div>
            );
```

PennyReact/index.html (Page 2 of 3)

```
         }

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

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

            let request = null;

            function fetchBooks(author) {
               function handleResponse() {
                  if (this.status !== 200) {
                     alert('Error: Failed to fetch data from server');
                     return;
                  }
                  let books = JSON.parse(this.response);
                  setBooks(books);
               }

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

               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 debouncedFetchBooks(author) {
               clearTimeout(timer);
               timer = setTimeout(()=>{fetchBooks(author);}, 500);
            }

            return (
               <div>
                  <h1>Author Search</h1>
                  Please enter an author name: 
                  <input
                      type='text'
                      onInput={ (event) => {
                         debouncedFetchBooks(event.target.value);
                      }}
                      autoFocus
                  />
                  <hr />
                  {books.map((book) => (
                     <div key={book.isbn}>
                        {book.isbn}: 
                        <strong>{book.author}</strong>: 
                        {book.title}
                        <br />
                     </div>
                  ))}
               </div>
            );
         }

```

PennyReact/index.html (Page 3 of 3)

```
126:          //------------------------------------------------------------
127:
128:          function PennyFooter() {
129:             const [datetime, setDatetime] = React.useState(new Date());
130:
131:             function updateFooter() {
132:                let dtInterval = window.setInterval(
133:                   () => {setDatetime(new Date());},
134:                   1000
135:                );
136:                return () => {window.clearInterval(dtInterval);};
137:             }
138:
139:             React.useEffect(updateFooter, []);
140:
141:             return (
142:                <div>
143:                   <hr />
144:                   Date and time: {datetime.toLocaleString()}
145:                   <br />
146:                   Created by 
147:                   <a href="https://www.cs.princeton.edu/~rdondero">
148:                   Bob Dondero</a>
149:                </div>
150:             );
151:          }
152:
153:          //------------------------------------------------------------
154:
155:          function App() {
156:             return (
157:                <div>
158:                   <PennyHeader />
159:                   <PennySearch />
160:                   <PennyFooter />
161:                </div>
162:             );
163:          }
164:
165:          //------------------------------------------------------------
166:
167:          let domRoot = document.getElementById('root');
168:          let reactRoot = ReactDOM.createRoot(domRoot);
169:          reactRoot.render(
170:             <React.StrictMode>
171:                <App />
172:             </React.StrictMode>
173:          );
174:
175:       </script>
176:    </body>
177: </html>
178:
```

blank (Page 1 of 1)

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

PennyReactWebpack/pennyheader.jsx (Page 1 of 1)

```
 1: //--------------------------------------------------------------------------
 2: // pennyheader.jsx
 3: // Author: Bob Dondero
 4: //--------------------------------------------------------------------------
 5:
 6: import React from 'react';
 7:
 8: //--------------------------------------------------------------------------
 9:
10: function PennyHeader() {
11:    const [datetime, setDatetime] = React.useState(new Date());
12:
13:    function updateHeader() {
14:       let apInterval = window.setInterval(
15:          () => {setDatetime(new Date());},
16:          1000
17:       );
18:       return () => {window.clearInterval(apInterval);};
19:    }
20:
21:    React.useEffect(updateHeader, []);
22:
23:    let hours = datetime.getHours();
24:    let ampm = (hours < 12) ? 'morning' : 'afternoon';
25:    return (
26:       <div>
27:          <hr />
28:          Good {ampm} and welcome to Penny.com
29:          <hr />
30:       </div>
31:    );
32: }
33:
34: //--------------------------------------------------------------------------
35:
36: export default PennyHeader;
```

PennyReactWebpack/pennyfooter.jsx (Page 1 of 1)

```
 1: //--------------------------------------------------------------------------
 2: // pennyfooter.jsx
 3: // Author: Bob Dondero
 4: //--------------------------------------------------------------------------
 5:
 6: import React from 'react';
 7:
 8: //--------------------------------------------------------------------------
 9:
10: function PennyFooter() {
11:    const [datetime, setDatetime] = React.useState(new Date());
12:
13:    function updateFooter() {
14:       let dtInterval = window.setInterval(
15:          () => {setDatetime(new Date());},
16:          1000
17:       );
18:       return () => {window.clearInterval(dtInterval);};
19:    }
20:
21:    React.useEffect(updateFooter, []);
22:
23:    return (
24:       <div>
25:          <hr />
26:          Date and time: {datetime.toLocaleString()}
27:          <br />
28:          Created by 
29:          <a href="https://www.cs.princeton.edu/~rdondero">
30:          Bob Dondero</a>
31:       </div>
32:    );
33: }
34:
35: //--------------------------------------------------------------------------
36:
37: export default PennyFooter;
```

PennyReactWebpack/pennysearch.jsx (Page 1 of 1)

```
 1: //--------------------------------------------------------------------------
 2: // pennysearch.jsx
 3: // Author: Bob Dondero
 4: //--------------------------------------------------------------------------
 5: import React from 'react';
 6: //--------------------------------------------------------------------------
 7:
 8: function PennySearch() {
 9:    const [books, setBooks] = React.useState([]);
10:
11:    let request = null;
12:
13:    function fetchBooks(author) {
14:       function handleResponse() {
15:          if (this.status !== 200) {
16:             alert('Error: Failed to fetch data from server');
17:             return;
18:          }
19:          let books = JSON.parse(this.response);
20:          setBooks(books);
21:       }
22:       function handleError() {
23:          alert('Error: Failed to fetch data from server');
24:       }
25:       let encodedAuthor = encodeURIComponent(author);
26:       let url = '/searchresults?author=' + encodedAuthor;
27:       if (request !== null)
28:          request.abort();
29:       request = new XMLHttpRequest();
30:       request.onload = handleResponse;
31:       request.onerror = handleError;
32:       request.open('GET', url);
33:       request.send();
34:    }
35:
36:    let timer = null;
37:    function debouncedFetchBooks(author) {
38:       clearTimeout(timer);
39:       timer = setTimeout(()=>{fetchBooks(author);}, 500);
40:    }
41:
42:    return (
43:       <div>
44:          <h1>Author Search</h1>
45:          Please enter an author name: 
46:          <input
47:              type='text'
48:              onInput={ (event) => {
49:                 debouncedFetchBooks(event.target.value);
50:              }}
51:              autoFocus
52:          />
53:          <hr />
54:          {books.map((book) => (
55:             <div key={book.isbn}>
56:                {book.isbn}: 
57:                <strong>{book.author}</strong>: 
58:                {book.title}
59:                <br />
60:             </div>
61:          ))}
62:       </div>
63:    );
64: }
65: export default PennySearch;
```

PennyReactWebpack/app.jsx (Page 1 of 1)

```
 1: //--------------------------------------------------------------------------
 2: // app.jsx
 3: // Author: Bob Dondero
 4: //--------------------------------------------------------------------------
 5:
 6: import React from 'react';
 7: import PennyHeader from './pennyheader.jsx';
 8: import PennySearch from './pennysearch.jsx';
 9: import PennyFooter from './pennyfooter.jsx';
10:
11: //--------------------------------------------------------------------------
12:
13: function App() {
14:    return (
15:       <div>
16:          <PennyHeader />
17:          <PennySearch />
18:          <PennyFooter />
19:       </div>
20:    );
21: }
22:
23: //--------------------------------------------------------------------------
24:
25: export default App;
```

PennyReactWebpack/index.jsx (Page 1 of 1)

```
 1: //--------------------------------------------------------------------
 2: // index.jsx
 3: // Author: Bob Dondero
 4: //--------------------------------------------------------------------
 5:
 6: import React from 'react';
 7: import ReactDOM from 'react-dom';
 8: import App from './app.jsx';
 9:
10: //--------------------------------------------------------------------
11:
12: let domRoot = document.getElementById('root');
13: let reactRoot = ReactDOM.createRoot(domRoot);
14:
15: reactRoot.render(
16:    <React.StrictMode>
17:       <App />
18:    </React.StrictMode>
19: );
```

PennyReactWebpack/index.html (Page 1 of 1)

```
 1: <!DOCTYPE html>
 2: <html>
 3:    <head>
 4:       <title>Penny.com</title>
 5:    </head>
 6:
 7:    <body>
 8:       <div id="root"></div>
 9:    </body>
10:
11:    <script src="/static/index.bundle.js"></script>
12: </html>
```

PennyReactWebpack/package.json (Page 1 of 1)

```
{
   "name": "pennyreactwebpack",
   "version": "1.0.0",
   "description": "The Penny Application Using React",
   "scripts": {
      "builddev": "webpack --mode=development",
      "buildprod": "webpack --mode=production"
   },
   "author": "Bob Dondero",
   "dependencies": {
      "@babel/core": "7.25.9",
      "@babel/preset-env": "7.25.9",
      "@babel/preset-react": "7.25.9",
      "babel-loader": "9.2.1",
      "react": "18.3.1",
      "react-dom": "18.3.1",
      "webpack": "5.97.1",
      "webpack-cli": "6.0.1"
   }
}
```

PennyReactWebpack/webpack.config.js (Page 1 of 1)

```
//---------------------------------------------------------------------------
// webpack.config.js
// Author: Lucas Manning and Bob Dondero
//---------------------------------------------------------------------------

const path = require('path');

let config = {
   entry: {
      index: './index.jsx',
   },

   output: {
      // Store the output files in a subdirectory named static.
      path: path.resolve(__dirname, 'static'),

      // Name the output files.
      filename: '[name].bundle.js'
   },

   module: {
      rules: [{
         // Use babel-loader as the loader.
         loader: 'babel-loader',

         // Use these loader options.
         options: {
            presets: ['@babel/preset-env', '@babel/preset-react']
         },

         // Exclude files whose names match the specified pattern.
         exclude: /node_modules/
      }]
   }
};

module.exports = (env, argv) => {
   if (argv.mode === 'development') {
      config.devtool = 'source-map';
   }
   return config;
};

```

PennyReactViteBackend/.env (Page 1 of 1)

```
1: FRONTEND_URL="http://localhost:3000"
```

PennyReactViteBackend/runserver.py (Page 1 of 1)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # runserver.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import sys
 9: import penny
10:
11: PORT = 5000
12:
13: def main():
14:
15:     if len(sys.argv) != 1:
16:         print(f'usage: {sys.argv[0]}', file=sys.stderr)
17:         sys.exit(1)
18:
19:     try:
20:         penny.app.run(host='0.0.0.0', port=PORT, debug=True)
21:     except Exception as ex:
22:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
23:         sys.exit(1)
24:
25: if __name__ == '__main__':
26:     main()
```

PennyReactViteBackend/penny.sql (Page 1 of 1)

```
DROP TABLE IF EXISTS books;

CREATE TABLE books (isbn TEXT PRIMARY KEY, author TEXT, title TEXT);

INSERT INTO books (isbn, author, title)
   VALUES ('123', 'Kernighan','The Practice of Programming');
INSERT INTO books (isbn, author, title)
   VALUES ('234', 'Kernighan','The C Programming Language');
INSERT INTO books (isbn, author, title)
   VALUES ('345', 'Sedgewick','Algorithms in C');
```

PennyReactViteBackend/database.py (Page 1 of 1)

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

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

import sqlite3
import contextlib

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

_DATABASE_URL = 'file:penny.sqlite'

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

def get_books(author):

    books = []

    with contextlib.closing(
        sqlite3.connect(_DATABASE_URL + '?mode=ro',
        isolation_level=None, uri=True)) as connection:

        with contextlib.closing(connection.cursor()) as cursor:

            query_str = '''
                SELECT isbn, author, title FROM books
                WHERE author LIKE ?
                '''
            cursor.execute(query_str, [author+'%'])

            table = cursor.fetchall()
            for row in table:
                book = {'isbn': row[0], 'author': row[1],
                    'title': row[2]}
                books.append(book)

    return books

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

def _test():

    books = get_books('ker')
    for book in books:
        print(book['isbn'])
        print(book['author'])
        print(book['title'])
        print()

if __name__ == '__main__':
    _test()
```

PennyReactViteBackend/requirements.txt (Page 1 of 1)

```
1: Flask
2: gunicorn
3: flask-cors
4: python-dotenv
5: SQLAlchemy
```

PennyReactViteBackend/penny.py (Page 1 of 1)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # penny.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import os
 9: import json
10: import flask
11: import flask_cors
12: import dotenv
13: import database
14:
15: #-----------------------------------------------------------------------
16:
17: dotenv.load_dotenv()
18: FRONTEND_URL = os.environ['FRONTEND_URL']
19:
20: app = flask.Flask(__name__)
21:
22: flask_cors.CORS(app, resources={r'/api/*': {'origins': FRONTEND_URL}})
23:
24: #-----------------------------------------------------------------------
25:
26: @app.route('/api/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
```

PennyReactViteFrontend/src/main.jsx (Page 1 of 1)

```
 1: //---------------------------------------------------------------------
 2: // main.jsx
 3: // Author: Bob Dondero
 4: //---------------------------------------------------------------------
 5:
 6: import React from 'react'
 7: import ReactDOM from 'react-dom/client'
 8: import App from './app.jsx'
 9:
10: //---------------------------------------------------------------------
11:
12: ReactDOM.createRoot(document.getElementById('root')).render(
13:   <React.StrictMode>
14:      <App />
15:   </React.StrictMode>,
16: )
```

PennyReactViteFrontend/src/app.jsx (Page 1 of 1)

```
 1: //---------------------------------------------------------------------------
 2: // app.jsx
 3: // Author: Bob Dondero
 4: //---------------------------------------------------------------------------
 5:
 6: import React from 'react';
 7: import PennyHeader from './pennyheader.jsx';
 8: import PennySearch from './pennysearch.jsx';
 9: import PennyFooter from './pennyfooter.jsx';
10:
11: //---------------------------------------------------------------------------
12:
13: function App() {
14:    return (
15:       <div>
16:          <PennyHeader />
17:          <PennySearch />
18:          <PennyFooter />
19:       </div>
20:    );
21: }
22:
23: //---------------------------------------------------------------------------
24:
25: export default App;
```

PennyReactViteFrontend/src/pennyheader.jsx (Page 1 of 1)

```
 1: //--------------------------------------------------------------------------
 2: // pennyheader.jsx
 3: // Author: Bob Dondero
 4: //--------------------------------------------------------------------------
 5:
 6: import React from 'react';
 7:
 8: //--------------------------------------------------------------------------
 9:
10: function PennyHeader() {
11:    const [datetime, setDatetime] = React.useState(new Date());
12:
13:    function updateHeader() {
14:       let apInterval = window.setInterval(
15:          () => {setDatetime(new Date());},
16:          1000
17:       );
18:       return () => {window.clearInterval(apInterval);};
19:    }
20:
21:    React.useEffect(updateHeader, []);
22:
23:    let hours = datetime.getHours();
24:    let ampm = (hours < 12) ? 'morning' : 'afternoon';
25:    return (
26:       <div>
27:          <hr />
28:          Good {ampm} and welcome to Penny.com
29:          <hr />
30:       </div>
31:    );
32: }
33:
34: //--------------------------------------------------------------------------
35:
36: export default PennyHeader;
```

PennyReactViteFrontend/src/pennyfooter.jsx (Page 1 of 1)

```
 1: //--------------------------------------------------------------------------
 2: // pennyfooter.jsx
 3: // Author: Bob Dondero
 4: //--------------------------------------------------------------------------
 5:
 6: import React from 'react';
 7:
 8: //--------------------------------------------------------------------------
 9:
10: function PennyFooter() {
11:    const [datetime, setDatetime] = React.useState(new Date());
12:
13:    function updateFooter() {
14:       let dtInterval = window.setInterval(
15:          () => {setDatetime(new Date());},
16:          1000
17:       );
18:       return () => {window.clearInterval(dtInterval);};
19:    }
20:
21:    React.useEffect(updateFooter, []);
22:
23:    return (
24:       <div>
25:          <hr />
26:          Date and time: {datetime.toLocaleString()}
27:          <br />
28:          Created by 
29:          <a href="https://www.cs.princeton.edu/~rdondero">
30:          Bob Dondero</a>
31:       </div>
32:    );
33: }
34:
35: //--------------------------------------------------------------------------
36:
37: export default PennyFooter;
```

PennyReactViteFrontend/src/pennysearch.jsx (Page 1 of 2)

```
 1: //--------------------------------------------------------------------
 2: // pennysearch.jsx
 3: // Author: Bob Dondero
 4: //--------------------------------------------------------------------
 5:
 6: import React from 'react';
 7: const BACKEND_URL = import.meta.env.VITE_BACKEND_URL;
 8:
 9: //--------------------------------------------------------------------
10:
11: function PennySearch() {
12:    const [books, setBooks] = React.useState([]);
13:
14:    let request = null;
15:
16:    function fetchBooks(author) {
17:       function handleResponse() {
18:          if (this.status !== 200) {
19:             alert('Error: Failed to fetch data from server');
20:             return;
21:          }
22:          let books = JSON.parse(this.response);
23:          setBooks(books);
24:       }
25:
26:       function handleError() {
27:          alert('Error: Failed to fetch data from server');
28:       }
29:
30:       let encodedAuthor = encodeURIComponent(author);
31:       let url = BACKEND_URL +
32:          '/api/searchresults?author=' + encodedAuthor;
33:       if (request !== null)
34:          request.abort();
35:       request = new XMLHttpRequest();
36:       request.onload = handleResponse;
37:       request.onerror = handleError;
38:       request.open('GET', url);
39:       request.send();
40:    }
41:
42:    let timer = null;
43:    function debouncedFetchBooks(author) {
44:       clearTimeout(timer);
45:       timer = setTimeout(()=>{fetchBooks(author);}, 500);
46:    }
47:
48:    return (
49:       <div>
50:          <h1>Author Search</h1>
51:          Please enter an author name: 
52:          <input
53:              type="text"
54:              onInput={ (event) => {
55:                 debouncedFetchBooks(event.target.value);
56:              } }
57:              autoFocus
58:          />
59:          <hr />
60:          {books.map((book) => (
61:             <div key={book.isbn}>
62:                {book.isbn}: 
63:                <strong>{book.author}</strong>: 
64:                {book.title}
65:                <br />
```

PennyReactViteFrontend/src/pennysearch.jsx (Page 2 of 2)

```
66:             </div>
67:          ))}
68:       </div>
69:    );
70: }
71:
72: export default PennySearch;
```

PennyReactViteFrontend/.env (Page 1 of 1)

```
1: VITE_BACKEND_URL=http://localhost:5000
```

PennyReactViteFrontend/vite.config.js (Page 1 of 1)

```
 1: import { defineConfig } from 'vite'
 2: import { loadEnv } from 'vite'
 3: import react from '@vitejs/plugin-react'
 4:
 5: // https://vitejs.dev/config/
 6:
 7: export default defineConfig({
 8:    plugins: [react()],
 9:    preview: { port:3000 },
10:    server: { port:3000 }
11: })
```