

PennyReact/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.
```

PennyReact/index.html (Page 1 of 3)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:
8:     <div id="root"></div>
9:
10:    <script src=
11:      "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production
12:      .min.js">
13:    </script>
14:
15:    <script src=
16:      "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.pr
17:      oduction.min.js">
18:    </script>
19:
20:    <script src=
21:      "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min
22:      .js">
23:    </script>
24:
25:    <script type="text/babel">
26:      'use strict';
27:
28:      //-----
29:      function PennyHeader() {
30:        const [datetime, setDatetime] = React.useState(new Date());
31:
32:        function updateHeader() {
33:          let apInterval = window.setInterval(
34:            () => {setDatetime(new Date());},
35:            1000
36:          );
37:          return () => {window.clearInterval(apInterval)};
38:        }
39:
40:        React.useEffect(updateHeader, []);
41:
42:        let hours = datetime.getHours();
43:        let ampm = (hours < 12) ? 'morning' : 'afternoon';
44:        return (
45:          <div>
46:            <hr />
47:            Good {ampm} and welcome to Penny.com
48:            <hr />
49:          </div>
50:        );
51:      }
52:
53:      //-----
54:
55:      function PennySearch() {
56:        const [books, setBooks] = React.useState([]);
57:
58:        let request = null;
59:
60:        function fetchBooks(author) {
61:          function handleResponse() {
62:            if (this.status !== 200) {

```

PennyReact/index.html (Page 2 of 3)

```

63:            alert('Error: Failed to fetch data from server');
64:            return;
65:          }
66:          let books = JSON.parse(this.response);
67:          setBooks(books);
68:        }
69:
70:        function handleError() {
71:          alert('Error: Failed to fetch data from server');
72:        }
73:
74:        let encodedAuthor = encodeURIComponent(author);
75:        let url = '/searchresults?author=' + encodedAuthor;
76:        if (request !== null)
77:          request.abort();
78:        request = new XMLHttpRequest();
79:        request.onload = handleResponse;
80:        request.onerror = handleError;
81:        request.open('GET', url);
82:        request.send();
83:      }
84:
85:      let timer = null;
86:
87:      function debouncedFetchBooks(author) {
88:        clearTimeout(timer);
89:        timer = setTimeout(()=>{fetchBooks(author)}; 500);
90:      }
91:
92:      return (
93:        <div>
94:          <h1>Author Search</h1>
95:          Please enter an author name:&nbsp;&nbsp;&nbsp;
96:          <input
97:            type='text'
98:            onInput={ (event) => {
99:              debouncedFetchBooks(event.target.value);
100:            } }
101:            autoFocus
102:          />
103:          <hr />
104:          {books.map((book) => (
105:            <div key={book.isbn}>
106:              {book.isbn}&nbsp;&nbsp;&nbsp;
107:              <strong>{book.author}</strong>&nbsp;&nbsp;&nbsp;
108:              {book.title}
109:            <br />
110:          </div>
111:        ))}
112:        </div>
113:      );
114:    }
115:
116:    //-----
117:
118:    function PennyFooter() {
119:      const [datetime, setDatetime] = React.useState(new Date());
120:
121:      function updateFooter() {
122:        let dtInterval = window.setInterval(
123:          () => {setDatetime(new Date());},
124:          1000
125:        );
126:        return () => {window.clearInterval(dtInterval)};
127:      }

```


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&nbsp;&nbsp;&nbsp;
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:&nbsp;
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}&nbsp;
57:           <strong>{book.author}</strong>&nbsp;
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)

```

1: {
2:   "name": "pennyreactwebpack",
3:   "version": "1.0.0",
4:   "description": "The Penny Application Using React",
5:   "scripts": {
6:     "builddev": "webpack --mode=development",
7:     "buildprod": "webpack --mode=production"
8:   },
9:   "author": "Bob Dondero",
10:  "dependencies": {
11:    "@babel/core": "7.25.9",
12:    "@babel/preset-env": "7.25.9",
13:    "@babel/preset-react": "7.25.9",
14:    "babel-loader": "9.2.1",
15:    "react": "18.3.1",
16:    "react-dom": "18.3.1",
17:    "webpack": "5.97.1",
18:    "webpack-cli": "6.0.1"
19:  }
20: }

```

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

```

1: //-----
2: // webpack.config.js
3: // Author: Lucas Manning and Bob Dondero
4: //-----
5:
6: const path = require('path');
7:
8: let config = {
9:   entry: {
10:    index: './index.jsx',
11:  },
12:
13:   output: {
14:    // Store the output files in a subdirectory named static.
15:    path: path.resolve(__dirname, 'static'),
16:
17:    // Name the output files.
18:    filename: '[name].bundle.js'
19:  },
20:
21:   module: {
22:    rules: [{
23:      // Use babel-loader as the loader.
24:      loader: 'babel-loader',
25:
26:      // Use these loader options.
27:      options: {
28:        presets: ['@babel/preset-env', '@babel/preset-react']
29:      },
30:
31:      // Exclude files whose names match the specified pattern.
32:      exclude: /node_modules/
33:    }]
34:  }
35: };
36:
37: module.exports = (env, argv) => {
38:   if (argv.mode === 'development') {
39:     config.devtool = 'source-map';
40:   }
41:   return config;
42: };
43:

```

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)

```

1: DROP TABLE IF EXISTS books;
2:
3: CREATE TABLE books (isbn TEXT PRIMARY KEY, author TEXT, title TEXT);
4:
5: INSERT INTO books (isbn, author, title)
6:   VALUES ('123', 'Kernighan', 'The Practice of Programming');
7: INSERT INTO books (isbn, author, title)
8:   VALUES ('234', 'Kernighan', 'The C Programming Language');
9: INSERT INTO books (isbn, author, title)
10:  VALUES ('345', 'Sedgewick', 'Algorithms in C');

```

PennyReactViteBackend/database.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # database.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sqlalchemy
10: import sqlalchemy.orm
11: import dotenv
12:
13: #-----
14:
15: dotenv.load_dotenv()
16: _DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///penny.sqlite')
17: _DATABASE_URL = _DATABASE_URL.replace('postgres://', 'postgresql://')
18: _DATABASE_URL = _DATABASE_URL.replace(
19:     'postgresql://', 'postgresql+psycopg://')
20:
21: #-----
22:
23: class Base(sqlalchemy.orm.DeclarativeBase):
24:     pass
25:
26: class Book(Base):
27:     __tablename__ = 'books'
28:     isbn = sqlalchemy.Column(sqlalchemy.String, primary_key=True)
29:     author = sqlalchemy.Column(sqlalchemy.String)
30:     title = sqlalchemy.Column(sqlalchemy.String)
31:
32: _engine = sqlalchemy.create_engine(_DATABASE_URL)
33:
34: #-----
35:
36: def get_books(author):
37:
38:     books = []
39:
40:     with sqlalchemy.orm.Session(_engine) as session:
41:
42:         query = session.query(Book).filter(
43:             Book.author.ilike(author+'%'))
44:         table = query.all()
45:         for row in table:
46:             book = {'isbn': row.isbn, 'author': row.author,
47:                   'title': row.title}
48:             books.append(book)
49:
50:     return books
51:
52: #-----
53:
54: # For testing:
55:
56: def _test():
57:     books = get_books('ker')
58:     for book in books:
59:         print(book['isbn'])
60:         print(book['author'])
61:         print(book['title'])
62:         print()
63:
64: if __name__ == '__main__':
65:     _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&nbsp;&nbsp;&nbsp;
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:&nbsp;
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}&nbsp;
63:           <strong>{book.author}</strong>&nbsp;
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: })
```