

## PennyReactWebpack/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.
```

## 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;
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:
6: import React from 'react';
7:
8: //-----
9: function PennySearch() {
10:   const [books, setBooks] = React.useState([]);
11:
12:   function fetchBooks(author) {
13:     function handleResponse() {
14:       if (this.status !== 200) {
15:         alert('Error: Failed to fetch data from server');
16:         return;
17:       }
18:       let books = JSON.parse(this.response);
19:       setBooks(books);
20:     }
21:
22:     function handleError() {
23:       alert('Error: Failed to fetch data from server');
24:     }
25:
26:     let encodedAuthor = encodeURIComponent(author);
27:     let url = '/searchresults?author=' + encodedAuthor;
28:     let request = new XMLHttpRequest();
29:     request.onload = handleResponse;
30:     request.onerror = handleError;
31:     request.open('GET', url);
32:     request.send();
33:     return () => {request.abort();}
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/main.js (Page 1 of 1)

```
1: //-----
2: // main.js
3: // Author: Bob Dondero
4: //-----
5:
6: import React from 'react';
7: import ReactDOM from 'react-dom';
8: import App from './App.js';
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/app.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:     "build": "webpack"
7:   },
8:   "author": "Bob Dondero",
9:   "dependencies": {
10:    "@babel/core": "7.25.9",
11:    "@babel/preset-env": "7.25.9",
12:    "@babel/preset-react": "7.25.9",
13:    "babel-loader": "9.2.1",
14:    "react": "19.0.0",
15:    "react-dom": "19.0.0",
16:    "webpack": "5.97.1",
17:    "webpack-cli": "6.0.1"
18:  }
19: }

```

## 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: module.exports = {
9:
10:  mode: "production", // "production", "development", "none"
11:
12:  entry: './main.js',
13:
14:  output: {
15:    // Store the output file in a subdirectory named static.
16:    path: path.resolve(__dirname, 'static'),
17:
18:    // Name the output file app.bundle.js.
19:    filename: 'app.bundle.js'
20:  },
21:
22:  module: {
23:    rules: [{
24:      // Use babel-loader as the loader.
25:      loader: 'babel-loader',
26:
27:      // Use these loader options.
28:      options: {
29:        presets: ['@babel/preset-env', '@babel/preset-react']
30:      },
31:
32:      // Include files whose names match the specified pattern.
33:      test: /\.jsx?$/,
34:
35:      // Exclude files whose names match the specified pattern.
36:      exclude: /node_modules/
37:    }]
38:  }
39: };

```

## 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('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(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 sqlite3
9: import contextlib
10:
11: #-----
12:
13: _DATABASE_URL = 'file:penny.sqlite?mode=ro'
14:
15: #-----
16:
17: def get_books(author):
18:
19:     books = []
20:
21:     with sqlite3.connect(_DATABASE_URL, isolation_level=None,
22:         uri=True) as connection:
23:
24:         with contextlib.closing(connection.cursor()) as cursor:
25:
26:             query_str = "SELECT isbn, author, title FROM books "
27:             query_str += "WHERE author LIKE ?"
28:             cursor.execute(query_str, [author+'%'])
29:
30:             table = cursor.fetchall()
31:             for row in table:
32:                 book = {'isbn': row[0], 'author': row[1],
33:                     'title': row[2]}
34:                 books.append(book)
35:
36:     return books
37:
38: #-----
39:
40: def _test():
41:
42:     books = get_books('ker')
43:     for book in books:
44:         print(book['isbn'])
45:         print(book['author'])
46:         print(book['title'])
47:         print()
48:
49: if __name__ == '__main__':
50:     _test()

```

## 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

```

**PennyReactViteBackend/requirements.txt (Page 1 of 1)**

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

**blank (Page 1 of 1)**

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



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

```

## PennyReactViteFrontend/src/PennySearch.jsx (Page 2 of 2)

```

66:   );
67: }
68:
69: export default PennySearch;

```

**PennyReactViteFrontend/.env.development (Page 1 of 1)**

1: VITE\_BACKEND\_URL=http://0.0.0.0:5000

**PennyReactViteFrontend/.env.production (Page 1 of 1)**

1: VITE\_BACKEND\_URL=http://0.0.0.0:5000

**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(({ mode }) => {
8:
9:   const env = loadEnv(mode, process.cwd(), '');
10:
11:   return {
12:     plugins: [react()],
13:   };
14: })
```

**blank (Page 1 of 1)**

1: This page is intentionally blank.

## arrow1.js (Page 1 of 1)

```
1: //-----
2: // arrow1.js
3: // Author: Bob Dondero
4: //-----
5:
6: 'use strict';
7:
8: class Car {
9:
10:   constructor(color) {
11:     this._color = color;
12:   }
13:
14:   writeColor() {
15:     // Error: this is undefined.
16:     setTimeout(
17:       function () { process.stdout.write(this._color + '\n'); },
18:       2000
19:     );
20:   }
21: }
22:
23: function main() {
24:   let blueCar = new Car('blue');
25:   blueCar.writeColor();
26: }
27:
28: if (require.main === module)
29:   main();
```

## arrow2.js (Page 1 of 1)

```
1: //-----
2: // arrow2.js
3: // Author: Bob Dondero
4: //-----
5:
6: 'use strict';
7:
8: class Car {
9:
10:   constructor(color) {
11:     this._color = color;
12:   }
13:
14:   writeColor() {
15:     let self = this;
16:     setTimeout(
17:       function () { process.stdout.write(self._color + '\n'); },
18:       2000
19:     );
20:   }
21: }
22:
23: function main()
24: {
25:   let blueCar = new Car('blue');
26:   blueCar.writeColor();
27: }
28:
29: if (require.main === module)
30:   main();
```

## arrow3.js (Page 1 of 1)

```
1: //-----  
2: // arrow3.js  
3: // Author: Bob Dondero  
4: //-----  
5:  
6: 'use strict';  
7:  
8: class Car {  
9:  
10:   constructor(color) {  
11:     this._color = color;  
12:   }  
13:  
14:   writeColor() {  
15:     setTimeout(  
16:       () => { process.stdout.write(this._color + '\n'); },  
17:       2000  
18:     );  
19:   }  
20: }  
21:  
22: function main() {  
23:   let blueCar = new Car('blue');  
24:   blueCar.writeColor();  
25: }  
26:  
27: if (require.main === module)  
28:   main();
```