

## PennyReactBundled/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:     window.setInterval(
15:       () => {setDatetime(new Date());},
16:       1000
17:     );
18:   }
19:
20:   React.useEffect(updateHeader, []);
21:
22:   let hours = datetime.getHours();
23:   let ampm = (hours < 12) ? 'morning' : 'afternoon';
24:   return (
25:     <div>
26:       <hr />
27:       Good {ampm} and welcome to Penny.com
28:       <hr />
29:     </div>
30:   );
31: }
32:
33: //-----
34:
35: export default PennyHeader;

```

## PennyReactBundled/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:
12:   const [datetime, setDatetime] = React.useState(new Date());
13:
14:   function updateFooter() {
15:     window.setInterval(
16:       () => {setDatetime(new Date());},
17:       1000
18:     );
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;

```

## PennyReactBundled/PennySearch.jsx (Page 1 of 1)

```

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

```

## PennyReactBundled/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;

```

## PennyReactBundled/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: let domRoot = document.getElementById('root');  
11: let reactRoot = ReactDOM.createRoot(domRoot);  
12:  
13: reactRoot.render(  
14:   <React.StrictMode>  
15:     <App />  
16:   </React.StrictMode>  
17: );
```

## PennyReactBundled/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>
```

## PennyReactBundled/package.json (Page 1 of 1)

```

1: {
2:   "name": "pennyreactbundled",
3:   "version": "1.0.0",
4:   "description": "The Penny Application Using React",
5:   "main": "main.js",
6:   "scripts": {
7:     "build": "NODE_OPTIONS=--openssl-legacy-provider webpack"
8:   },
9:   "author": "Bob Dondero",
10:  "license": "ISC",
11:  "dependencies": {
12:    "@babel/core": "7.22.1",
13:    "@babel/preset-env": "7.22.4",
14:    "@babel/preset-react": "7.22.3",
15:    "babel-loader": "9.1.2",
16:    "react": "18.2.0",
17:    "react-dom": "18.2.0",
18:    "webpack": "5.85.0",
19:    "webpack-cli": "5.1.3"
20:  }
21: }

```

## PennyReactBundled/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: };

```

## PennyReactViteApi/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: def main():
12:
13:     if len(sys.argv) != 1:
14:         print('Usage: ' + sys.argv[0], file=sys.stderr)
15:         sys.exit(1)
16:
17:     PORT = 5000
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()
```

## PennyReactViteApi/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');
```

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

```

## PennyReactViteApi/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 flask_dotenv
13: import database
14:
15: #-----
16:
17: flask_dotenv.load_dotenv()
18: CLIENT_URL = os.environ['CLIENT_URL']
19:
20: app = flask.Flask(__name__)
21:
22: flask_cors.CORS(app, resources={r'/api/*': {'origins': CLIENT_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

```

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

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

**PennyReactVite/src/main.jsx (Page 1 of 1)**

```
1: import React from 'react'
2: import ReactDOM from 'react-dom/client'
3: import App from './App.jsx'
4:
5: ReactDOM.createRoot(document.getElementById('root')).render(
6:   <React.StrictMode>
7:     <App />
8:   </React.StrictMode>,
9: )
```

## PennyReactVite/src/App.jsx (Page 1 of 2)

```

1: //-----
2: // App.jsx
3: // Author: Bob Dondero
4: //-----
5:
6: import * as React from 'react';
7: const VITE_SERVER_URL = import.meta.env.VITE_SERVER_URL
8:
9: //-----
10:
11: function PennyHeader() {
12:   const [datetime, setDatetime] = React.useState(new Date());
13:
14:   function updateHeader() {
15:     window.setInterval(
16:       () => {setDatetime(new Date());},
17:       1000
18:     );
19:   }
20:   React.useEffect(updateHeader, []);
21:
22:   let hours = datetime.getHours();
23:   let ampm = (hours < 12) ? 'morning' : 'afternoon';
24:   return (
25:     <div>
26:       <hr />
27:       Good {ampm} and welcome to Penny.com
28:       <hr />
29:     </div>
30:   );
31: }
32:
33: //-----
34:
35: function PennySearch() {
36:   const [author, setAuthor] = React.useState('');
37:   const [books, setBooks] = React.useState([]);
38:   function fetchBooks() {
39:     function handleResponse() {
40:       if (this.status !== 200) {
41:         alert('Error: Failed to fetch data from server');
42:         return;
43:       }
44:       let books = JSON.parse(this.response);
45:       setBooks(books);
46:     }
47:     function handleError() {
48:       alert('Error: Failed to fetch data from server');
49:     }
50:     let encodedAuthor = encodeURIComponent(author);
51:     let url = VITE_SERVER_URL +
52:       '/api/searchresults?author=' + encodedAuthor;
53:     let request = new XMLHttpRequest();
54:     request.onload = handleResponse;
55:     request.onerror = handleError;
56:     request.open('GET', url);
57:     request.send();
58:     return () => {request.abort();}
59:   }
60:   React.useEffect(fetchBooks, [author]);
61:
62:   let timer = null;
63:   function debouncedSetAuthor(author) {
64:     clearTimeout(timer);
65:     timer = setTimeout(()=>{setAuthor(author);}, 500);

```

## PennyReactVite/src/App.jsx (Page 2 of 2)

```

66:   }
67:
68:   return (
69:     <div>
70:       <h1>Author Search</h1>
71:       Please enter an author name:&nbsp;&nbsp;&nbsp;
72:       <input
73:         type="text"
74:         onInput={ (event) => {
75:           debouncedSetAuthor(event.target.value);
76:         } }
77:         autoFocus
78:       />
79:       <hr />
80:       {books.map((book) => (
81:         <div key={book.isbn}>
82:           {book.isbn}&nbsp;&nbsp;&nbsp;
83:           <strong>{book.author}</strong>&nbsp;&nbsp;&nbsp;
84:           {book.title}
85:           <br />
86:         </div>
87:       ))}
88:     </div>
89:   );
90: }
91:
92: //-----
93:
94: function PennyFooter() {
95:   const [datetime, setDatetime] = React.useState(new Date());
96:
97:   function updateFooter() {
98:     window.setInterval(
99:       () => {setDatetime(new Date());},
100:      1000
101:    );
102:   }
103:   React.useEffect(updateFooter, []);
104:
105:   return (
106:     <div>
107:       <hr />
108:       Date and time: {datetime.toLocaleString()}
109:       <br />
110:       Created by&nbsp;&nbsp;&nbsp;
111:       <a href="https://www.cs.princeton.edu/~rdondero">
112:         Bob Dondero</a>
113:     </div>
114:   );
115: }
116:
117: //-----
118:
119: function App() {
120:   return (
121:     <div>
122:       <PennyHeader />
123:       <PennySearch />
124:       <PennyFooter />
125:     </div>
126:   );
127: }
128:
129: //-----
130: export default App;

```



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

1: VITE\_SERVER\_URL=http://localhost:5000

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

1: VITE\_SERVER\_URL=https://pennyreactviteback.onrender.com

## 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();
```