The Sound of Code

Group 5

brief walk through of the app and explanation of it

This program can convert js code into beautiful music. It will make different sounds according to different data types and structures. Even erroneous codes also have unique melodies.

Although the program is not finished, we will still be showing the completed parts of the project.

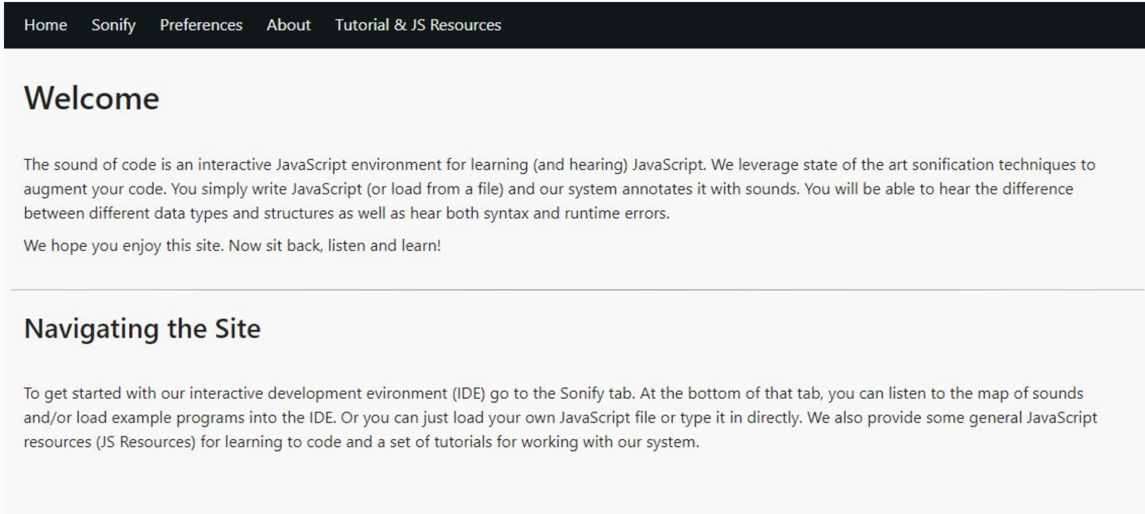
Use Case Diagram

For the "Home" interface

Learn about the program:

Users read introductory content to learn what the program is and how to use it.

The interface also provides new users with links to navigate to other features and resources.



Welcome

The sound of code is an interactive JavaScript environment for learning (and hearing) JavaScript. We leverage state of the art sonification techniques to augment your code. You simply write JavaScript (or load from a file) and our system annotates it with sounds. You will be able to hear the difference between different data types and structures as well as hear both syntax and runtime errors.

We hope you enjoy this site. Now sit back, listen and learn!

Navigating the Site

To get started with our interactive development evironment (IDE) go to the Sonify tab. At the bottom of that tab, you can listen to the map of sounds and/or load example programs into the IDE. Or you can just load your own JavaScript file or type it in directly. We also provide some general JavaScript resources (JS Resources) for learning to code and a set of tutorials for working with our system.

Fo

W

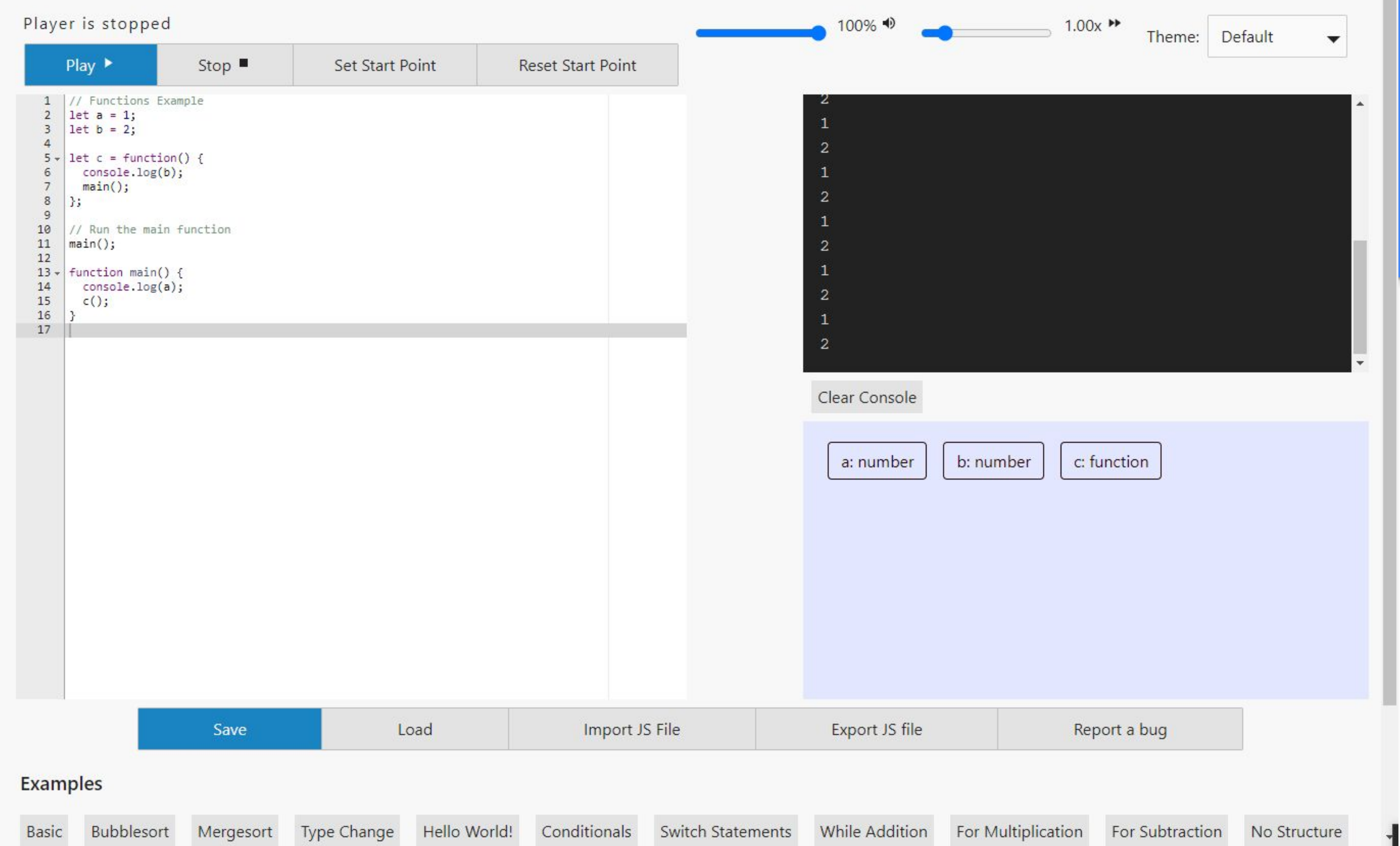
Ex

Sc

Vi
ite

Co

Vi



```
// Functions Example
let a = 1;
let b = 2;

let c = function() {
  console.log(b);
  main();
};

// Run the main function
main();

function main() {
  console.log(a);
  c();
}

```

```
2
1
2
1
2
1
2
1
2
1
2
```

Use Case Diagram

For the "Preferences" interface :

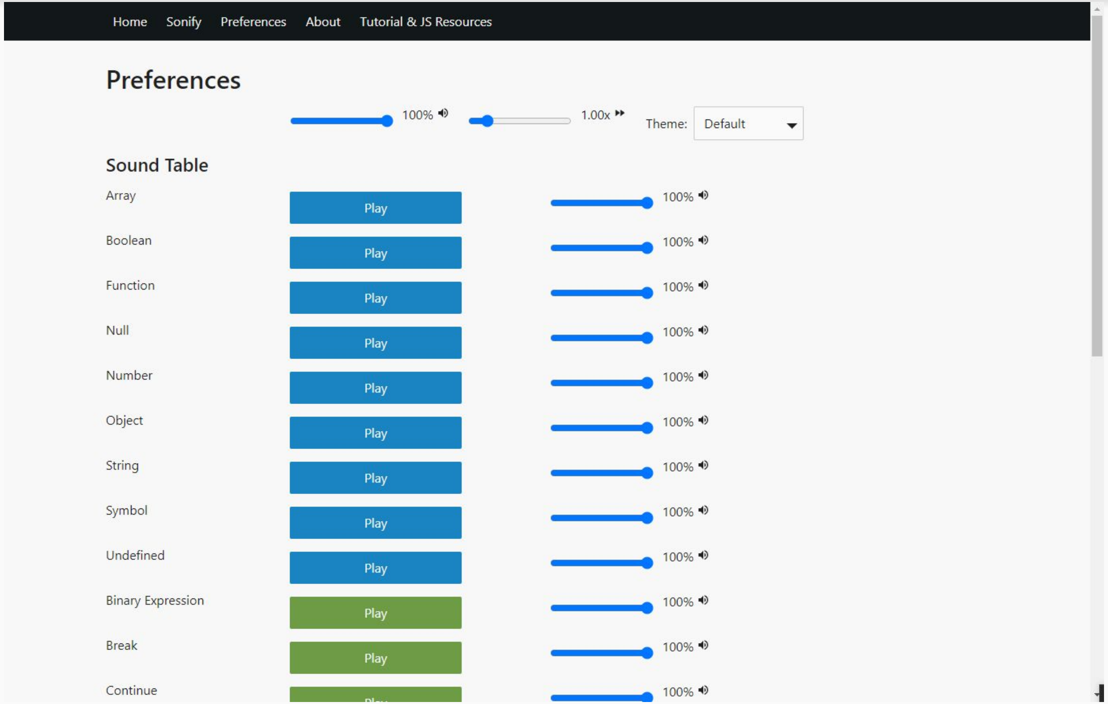
Preview sound effects:

Users can preview what sounds different types of code will produce.

For the "About" interface:

Understand the project background and team:

Users can learn about the project's history, sponsors, and project team.



Preferences

100% 1.00x Theme: Default

Sound Table

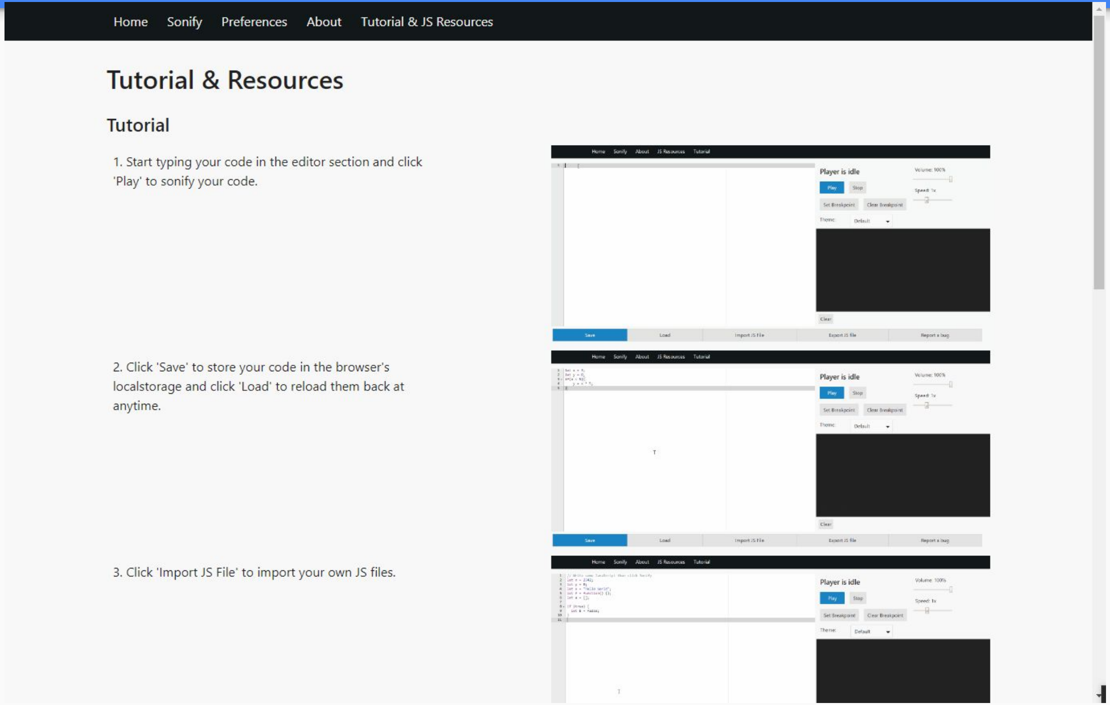
Array	Play	100%
Boolean	Play	100%
Function	Play	100%
Null	Play	100%
Number	Play	100%
Object	Play	100%
String	Play	100%
Symbol	Play	100%
Undefined	Play	100%
Binary Expression	Play	100%
Break	Play	100%
Continue	Play	100%

Use Case Diagram

For the "Tutorial & Resources" interface :

Study and review resources:

Users can learn how to use the platform, as well as JavaScript programming, through tutorials and resources.



Tutorial & Resources

Tutorial

1. Start typing your code in the editor section and click 'Play' to sonify your code.

2. Click 'Save' to store your code in the browser's localstorage and click 'Load' to reload them back at anytime.

3. Click 'Import JS File' to import your own JS files.

Why Containerize?

Consistent Environments

- Containers provide isolated, portable runtimes across environments

Scalability

- Kubernetes enables automatically scaling and load balancing

Resiliency

- Orchestration handles failures, restarts, replication

TimeLine of Containerization

Week 1: Evaluate and select target versions

Determine compatible Node.js and dependency versions

Select Kubernetes provider and tools

Week 2: Build containers

Configure Dockerfiles for app and database

Test locally, push images to registry

Week 3: Deploy to Kubernetes

Define Kubernetes config YAML

Set up deployment pipelines

Create Kubernetes cluster on cloud

Week 4: Validation

Smoke and integration testing

Performance benchmarking

Security auditing

Monitoring and logging

Where We Current Are?

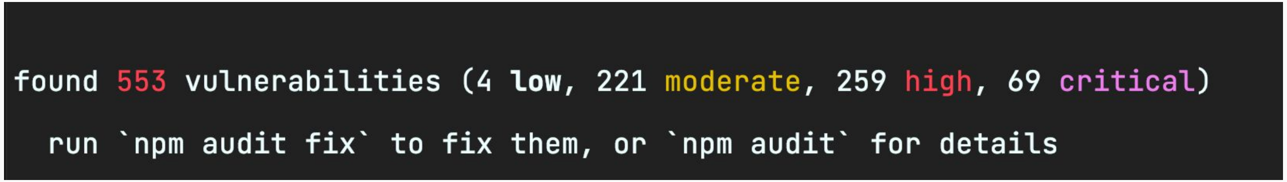
- Learning Docker & Kubernetes fundamentals
  - Covered Docker architecture, images, containers
  - Learned to manage containers and images from CLI
- Built sample Dockerfiles and images
  - Created containers for simple Node.js apps
  - Pushed images to DockerHub registry
  - Running containers with ports & volumes
- Run a local Kubernetes cluster with Minikube
  - Practice kubectl commands to manage cluster
  - Get hands-on with core Kubernetes objects

Next Steps

- Containerize current applications
- Build Dockerfiles tailored for apps
- Optimize containers for production
- Deploy apps to managed Kubernetes cluster
- Choose a managed Kubernetes provider
- Learn to operate apps on production-grade cluster

Initial State of the Project

- The application can be run from a server, but is insecure due to dependencies being outdated or deprecated
- The code can be run locally with node.js v12.22.12 and npm v6.14.16
- The application itself runs as it was designed to, there are no errors or bugs that occur from poorly implemented code



```
found 553 vulnerabilities (4 low, 221 moderate, 259 high, 69 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```

Timeline

Demo 1: explain the application, create documentation, upgrade npm and node.js

Demo 2: have the dependency errors resolved so the application can run, have the application in a docker so it is more secure, clean up pages on the application

Demo 3: develop and implement new features

New Features/Improvements

- Improving “layered” sound when loops are running
- Updated About section on the website
- New code examples, including ones with “bad” code noise
- Implementing a sound for recursion
- Verifying the licenses for all the packages are still open source
- Adding more resources for learning Javascript

JavaScript

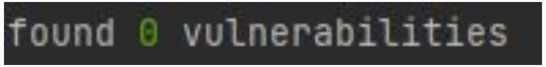
What we’ve accomplished

Forward progress:

Updated node.js to v20.6.1

Updated npm to v10.1.0

Reduced vulnerabilities to 0



```
found 0 vulnerabilities
```

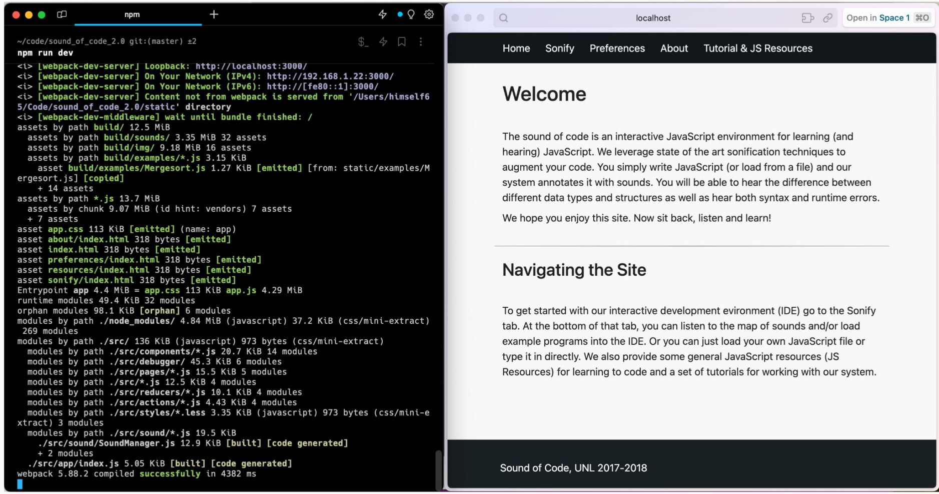
Issues created by that progress:

The application no longer launches when ran

The Webpack dependency now has an error

Upgrade build tools

- Webpack from 4 to 5
- Improve the build steps
- CI/CD



```
~/code/sound_of_code_2.0 git:(master) ±2
npm run dev
<i> [webpack-dev-server] Loopback: http://localhost:3000/
<i> [webpack-dev-server] On Your Network (IPv4): http://192.168.1.22:3000/
<i> [webpack-dev-server] On Your Network (IPv6): http://[fe80::1]:3000/
<i> [webpack-dev-server] Content not from webpack is served from '/Users/himself6
5/Code/sound_of_code_2.0/static' directory
<i> [webpack-dev-middleware] wait until bundle finished: /
assets by path build/ 12.5 MiB
  assets by path build/sounds/ 3.35 MiB 32 assets
  assets by path build/img/ 9.18 MiB 16 assets
  assets by path build/examples/*.js 3.15 KiB
    asset build/examples/Mergesort.js 1.27 KiB [emitted] [from: static/examples/M
ergesort.js] [copied]
    + 14 assets
assets by path *.js 13.7 MiB
  assets by chunk 9.07 MiB (id hint: vendors) 7 assets
  + 7 assets
asset app.css 113 KiB [emitted] (name: app)
asset about/index.html 318 bytes [emitted]
asset index.html 318 bytes [emitted]
asset preferences/index.html 318 bytes [emitted]
asset resources/index.html 318 bytes [emitted]
asset sonify/index.html 318 bytes [emitted]
Entrypoint app 4.4 MiB = app.css 113 KiB app.js 4.29 MiB
runtime modules 49.4 KiB 32 modules
orphan modules 98.1 KiB [orphan] 6 modules
modules by path ./node_modules/ 4.84 MiB (javascript) 37.2 KiB (css/mini-extract)
 269 modules
modules by path ./src/ 136 KiB (javascript) 973 bytes (css/mini-extract)
  modules by path ./src/components/*.js 20.7 KiB 14 modules
  modules by path ./src/debugger/ 45.3 KiB 6 modules
  modules by path ./src/pages/*.js 15.5 KiB 5 modules
  modules by path ./src/*.js 12.5 KiB 4 modules
  modules by path ./src/reducers/*.js 10.1 KiB 4 modules
  modules by path ./src/actions/*.js 4.43 KiB 4 modules
  modules by path ./src/styles/*.less 3.35 KiB (javascript) 973 bytes (css/mini-e
xtract) 3 modules
  modules by path ./src/sound/*.js 19.5 KiB
    ./src/sound/SoundManager.js 12.9 KiB [built] [code generated]
    + 2 modules
  ./src/app/index.js 5.05 KiB [built] [code generated]
webpack 5.88.2 compiled successfully in 4382 ms
```

Welcome

The sound of code is an interactive JavaScript environment for learning (and hearing) JavaScript. We leverage state of the art sonification techniques to augment your code. You simply write JavaScript (or load from a file) and our system annotates it with sounds. You will be able to hear the difference between different data types and structures as well as hear both syntax and runtime errors.

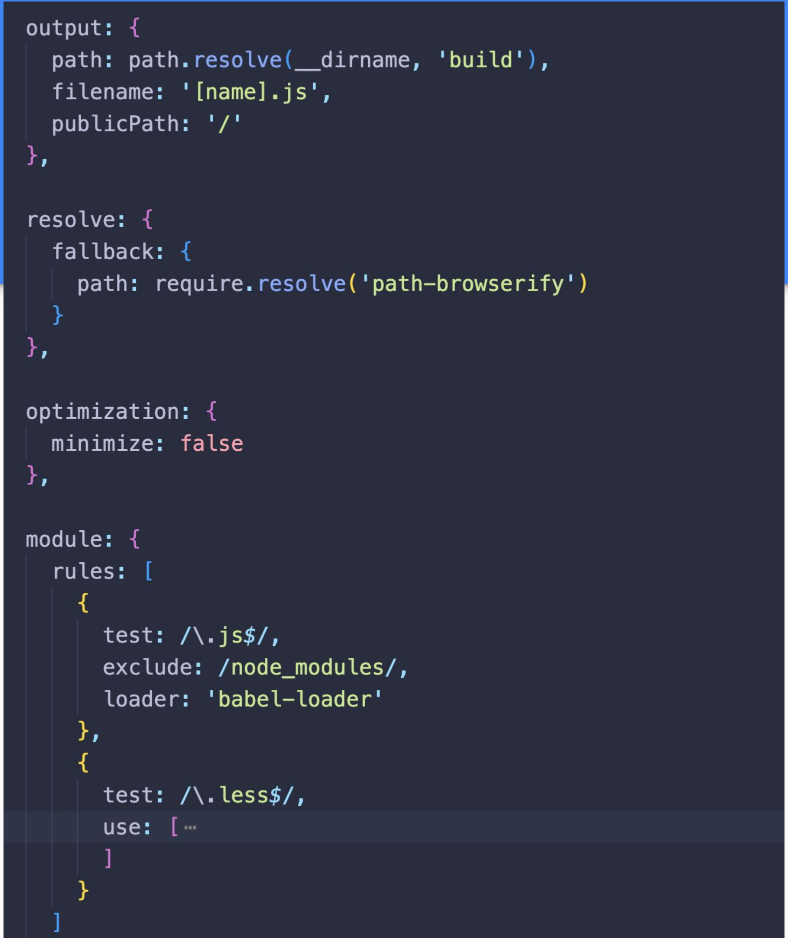
We hope you enjoy this site. Now sit back, listen and learn!

Navigating the Site

To get started with our interactive development evironment (IDE) go to the Sonify tab. At the bottom of that tab, you can listen to the map of sounds and/or load example programs into the IDE. Or you can just load your own JavaScript file or type it in directly. We also provide some general JavaScript resources (JS Resources) for learning to code and a set of tutorials for working with our system.

Sound of Code, UNL 2017-2018

Preview: https://sound-of-code.vercel.app/

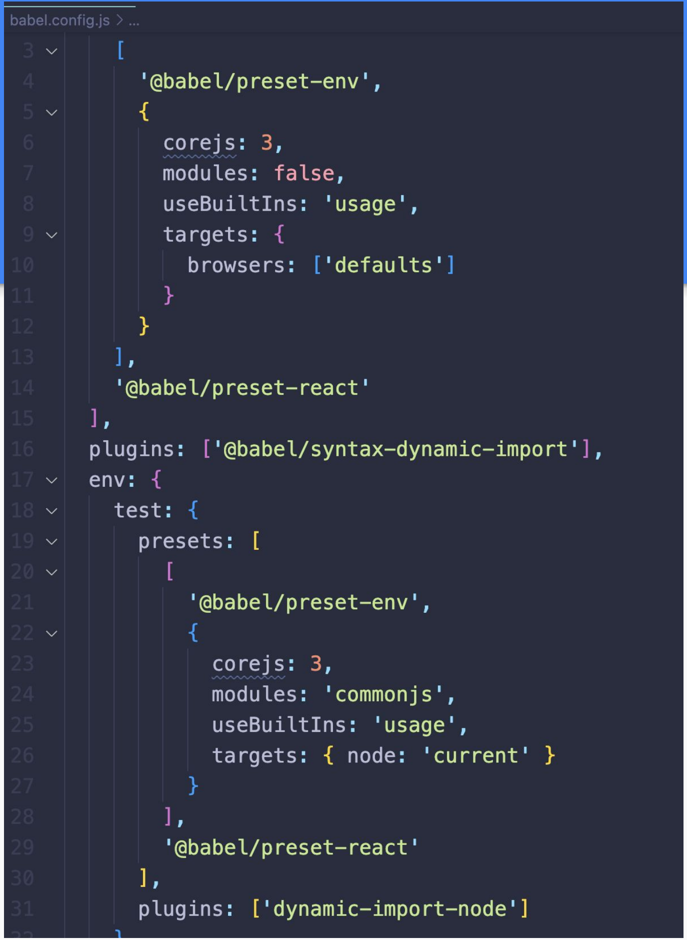


```
output: {
  path: path.resolve(__dirname, 'build'),
  filename: '[name].js',
  publicPath: '/'
},

resolve: {
  fallback: {
    path: require.resolve('path-browserify')
  }
},

optimization: {
  minimize: false
},

module: {
  rules: [
    {
      test: /\.js$/,
      exclude: /node_modules/,
      loader: 'babel-loader'
    },
    {
      test: /\.less$/,
      use: [⋯
      ]
    }
  ]
```



```
    [
      '@babel/preset-env',
      {
        corejs: 3,
        modules: false,
        useBuiltIns: 'usage',
        targets: {
          browsers: ['defaults']
        }
      }
    ],
    '@babel/preset-react'
  ],
  plugins: ['@babel/syntax-dynamic-import'],
  env: {
    test: {
      presets: [
        [
          '@babel/preset-env',
          {
            corejs: 3,
            modules: 'commonjs',
            useBuiltIns: 'usage',
            targets: { node: 'current' }
          }
        ],
        '@babel/preset-react'
      ],
      plugins: ['dynamic-import-node']
```

Goals for this project

- Update npm and node.js to the most recent versions
- Fix the deprecated libraries by replacing them with newer versions or refactoring the code
- Containerize the code using a docker
- Add new features

Two Future Directions

Education

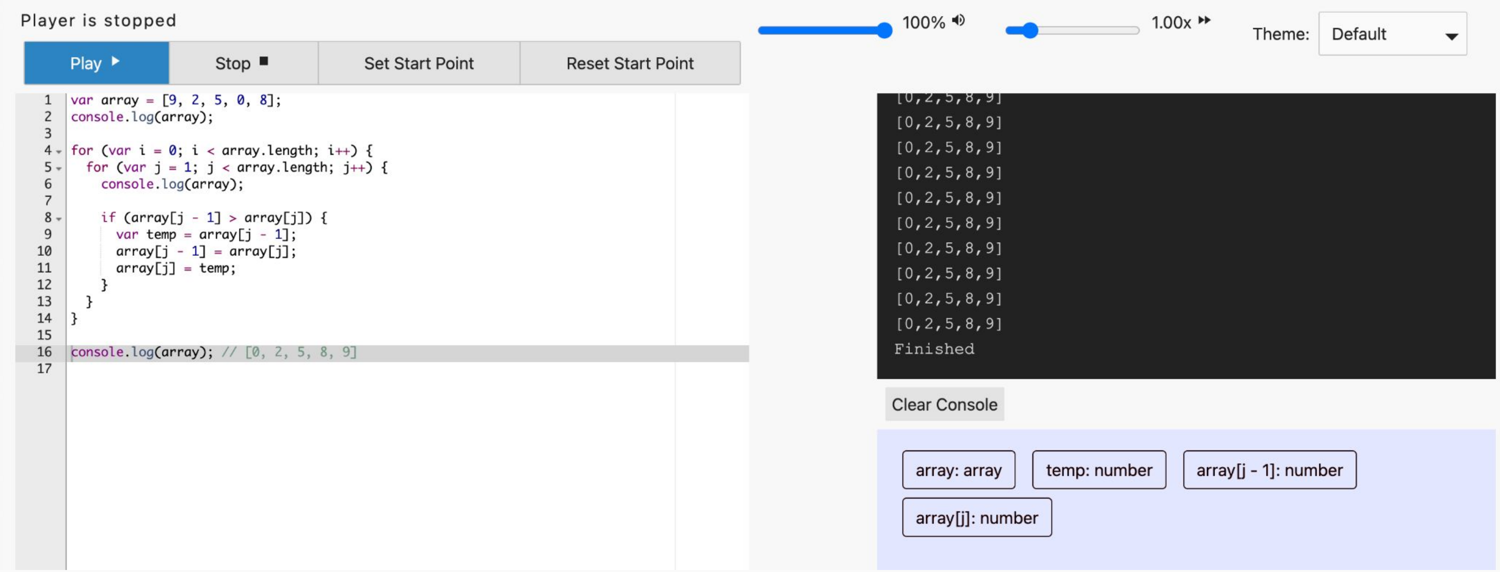
- Sonify is very useful to helping novice programmers understand how code is being ran
- There are several resources on Sonify's website to start programming in Javascript

Debugging

- Listening to Sonify helps programmers hear when something isn't running correctly
- Very helpful for the programmers with visual impairments to understand the code better

Use Case Demonstration

- Bubble Sort



```
var array = [9, 2, 5, 0, 8];
console.log(array);

for (var i = 0; i < array.length; i++) {
  for (var j = 1; j < array.length; j++) {
    console.log(array);

    if (array[j - 1] > array[j]) {
      var temp = array[j - 1];
      array[j - 1] = array[j];
      array[j] = temp;
    }
  }
}

console.log(array); // [0, 2, 5, 8, 9]
```

```
[0,2,5,8,9]
[0,2,5,8,9]
[0,2,5,8,9]
[0,2,5,8,9]
[0,2,5,8,9]
[0,2,5,8,9]
[0,2,5,8,9]
[0,2,5,8,9]
[0,2,5,8,9]
[0,2,5,8,9]
Finished
```

Conclude

- The Sounds of Code project has the potential to be incredibly valuable for educational purposes, offering a unique perspective on the debugging process.
- To enhance the project's security, we can fix vulnerabilities by updating the framework and eliminating outdated dependencies.
- To expand the project's capabilities, we aim to enhance the functionality of Sounds of Code, providing an improved educational and debugging experience.