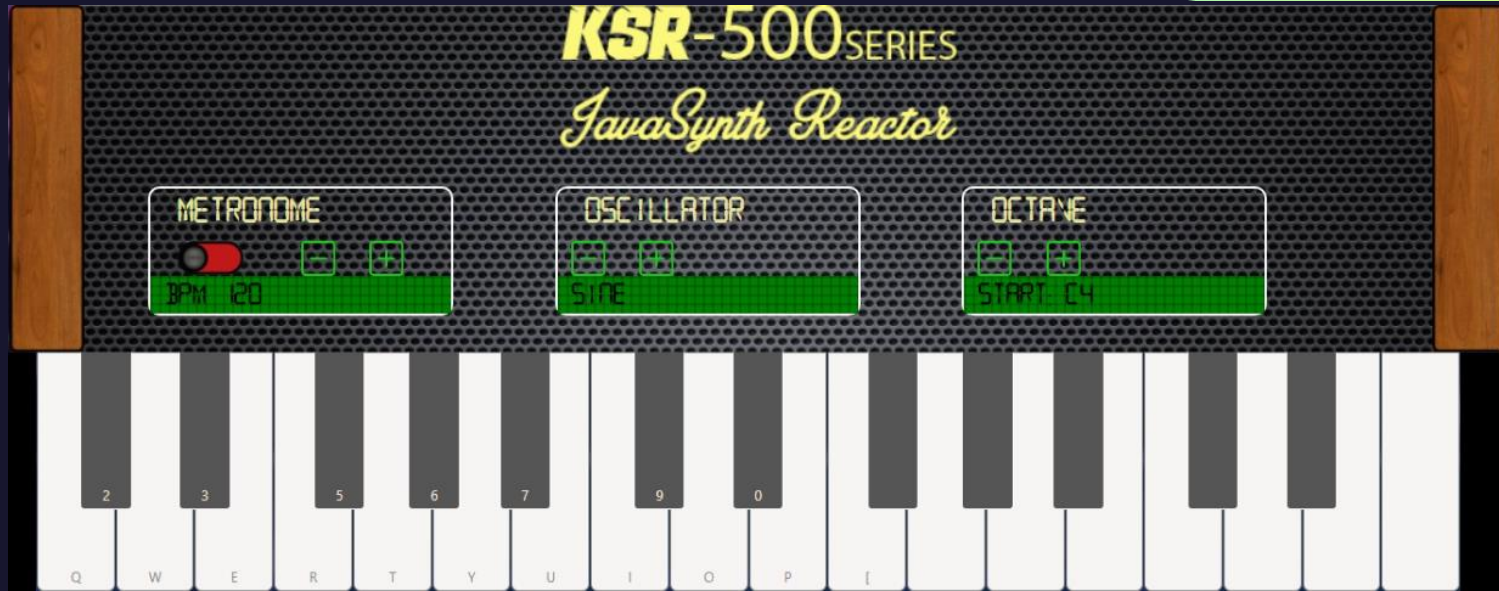


KSR-500

JavaSynth Reactor

Documentatie

Stamate Cezar Ionel





PAGINA PRINCIPALA



CUPRINS



CAPITOLE

01

*Prezentare si
Obiectiv*

02

*Tehnologii
utilizate*

03

*Functionalitati
curente*

04

*Functionalitati
viitoare si
imbunatatiri*



MULTUMESC!

CUPRINS

01

Prezentare si Obiectiv

Descrierea aplicatiei,
link-uri si obiectivul
aplicatiei

02

Tehnologii utilizate

Limbaje, framework-uri
si node packages
folosite.

03

Functionalitati curente

Descrierea
functionalitatilor si a
serviciilor

04

Functionalitati viitoare si imbunatatiri

Functionalitati viitoare,
known issues,
imbunatatiri.



Prezentare si Obiectiv



PAGINA PRINCIPALA



CUPRINS

Prezentare si Obiectiv

Aplicatia KSR-500 (JavaSynth Reactor) este o prezentare a capacitatilor JavaScript React library si Tone.js.

Este un exercitiu tehnic, menit sa exploreze tehnologiile mai sus mentionate, care mai tarziu sa fie dezvoltat intr-o aplicatie DAW (Digital Audio Workstation) pentru utilizatori.

Link aplicatie: [KSR-500 \(react-synth.netlify.app\)](https://react-synth.netlify.app)

Link GitHub: [kaiser899/react-synth \(github.com\)](https://github.com/kaiser899/react-synth)

Realizat de: Stamate Cezar Ionel

Sub indrumarea: Cristian Scarlat (SDA Academy)



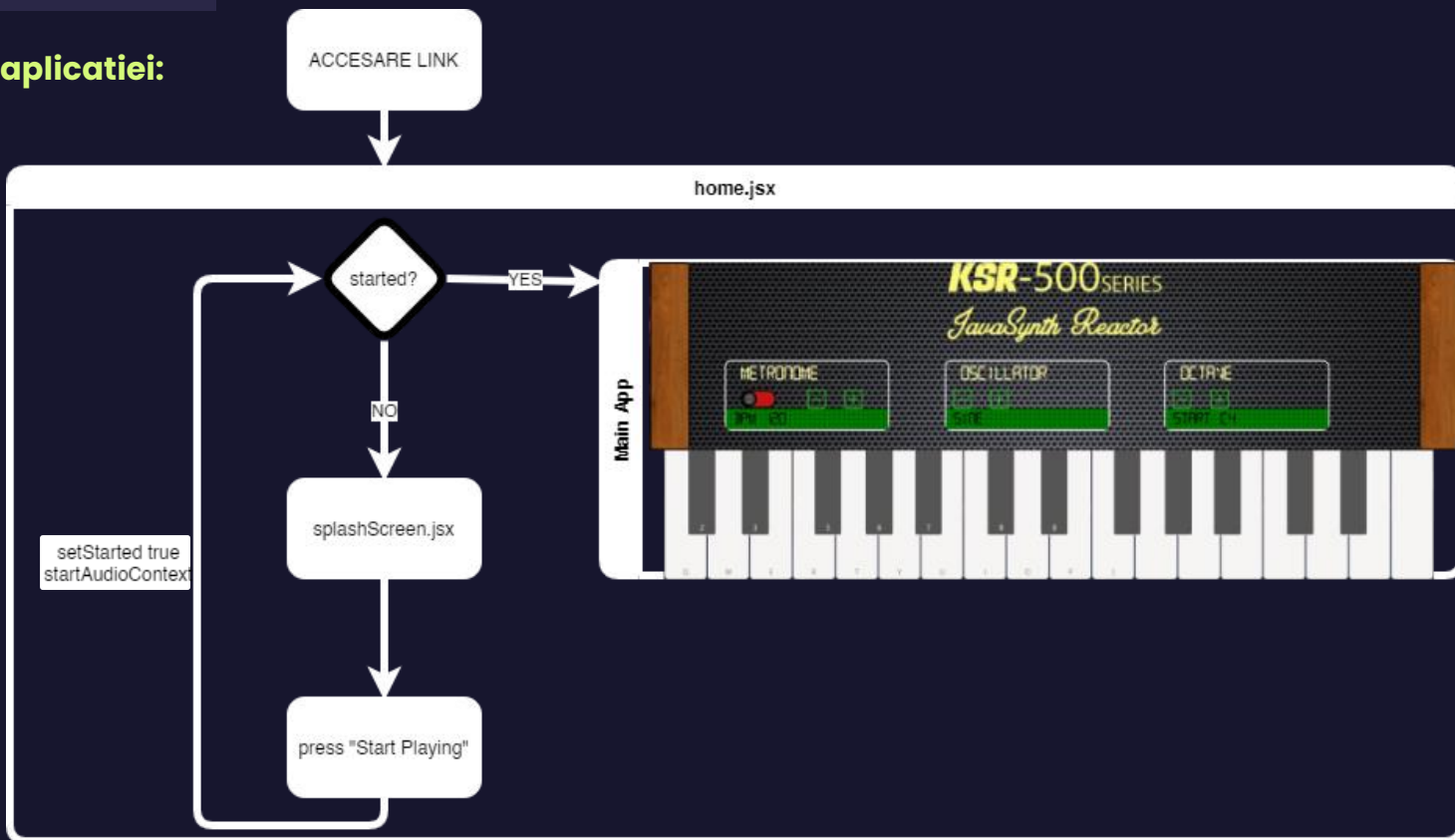
PAGINA PRINCIPALA



CUPRINS

Prezentare si Obiectiv

Structura aplicatiei:



02

Tehnologii utilizate



PAGINA PRINCIPALA



CUPRINS

Tehnologii utilizate

Aplicatia este construita folosind limbaj HTML, CSS, Javascript in libraria React.

Dependente:

- react : v17.0.2 ([react - npm \(npmjs.com\)](https://www.npmjs.com/package/react)) MIT license
utilizat pentru construirea unei aplicatii single-page.
- bootstrap: v5.1.3 ([bootstrap - npm \(npmjs.com\)](https://www.npmjs.com/package/bootstrap)) MIT license
- react-bootstrap: v2.0.3 ([react-bootstrap - npm \(npmjs.com\)](https://www.npmjs.com/package/react-bootstrap)) MIT license
utilizate pentru crearea interfetei de tip grid.
- react-piano: v3.1.3 ([react-piano - npm \(npmjs.com\)](https://www.npmjs.com/package/react-piano)) MIT license
utilizat pentru implementarea unei claviaturi interactive.
- tone : v14.7.77 ([tone - npm \(npmjs.com\)](https://www.npmjs.com/package/tone)) MIT license
utilizat pentru implementarea interfetei interactive Web Audio.

Software si tool-uri folosite:

- Node.js
- Visual Studio Code
- Git
- Adobe Photoshop



Functionalitati curente



PAGINA PRINCIPALA



CUPRINS

Functionalitati curente

Aplicatia se poate accesa folosind URL-ul [KSR-500 \(react-synth.netlify.app\)](https://KSR-500.netlify.app)

In splash screen-ul aplicatii se pot gasi:

- un screenshot al aplicatiei
- un buton de activare cu eticheta "Start Playing"
- o scurta descriere a aplicatiei
- link catre pagina de GitHub

Splash Screen-ul este necesar, datorita particularitatilor Web Audio, care necesita activarea manuala a contextului audio inainte de utilizare.

La incarcarea paginii home.jsx, se va verifica state-ul *started*.
Daca acesta este false (0), atunci se va afisa splash screen-ul.

```
return (  
  <Container fluid id="background-main">  
    {!started ? (  
      <Row>  
        <SplashScreen handleButtonStart={handleButtonStart} />  
      </Row>  
    ) : (  
      <Container>  
        <Container id="backdrop">
```



PAGINA PRINCIPALA



CUPRINS

Functionalitati curente

Dupa activarea butonului de "Start Playing", se va apela functia asincrona **handleButtonStart**, care are ca scop activarea contextului audio (**startAudioContext** – *audioFunctions.js*) si setarea state-ului **started** ca true (1).

Dupa schimbarea state-ului, pagina se va reincarca cu restul elementelor de design si cu componentele functionale ale aplicatiei.

Elemente de design:

- Backdrop : consta din imaginea retro prezenta pe fundalul aplicatiei.
- App Row: consta din corpul principal al sintetizatorului, compus la randul lui din:
 - ❖ Margine lemn stanga
 - ❖ Element central cu logo si comenzi (screen.jsx)
 - ❖ Margine lemn dreapta
- Piano Keyboard: consta din componenta responsabila cu afisarea claviaturii, scurtaturilor de tastatura si a feedback-ului la apasarea tastelor.





PAGINA PRINCIPALA



CUPRINS

Functionalitati curente

METRONOME – reprezentat prin `metronome.jsx`, incarcat in `screen.jsx`.
Componenta ofera posibilitatea de a avea un metronom cu toggle on/off, si BPM control.

La apasarea butonului de toggle, functia **`handleToggleMetronome`** va fi apelata, care seteaza state-ul **`isMetronomeOn`** true sau false, in functie de starea actuala.

Daca metronomul este oprit (**`isMetronomeOn`** false),
imaginea de metronome toggle se schimba in varianta
pentru "ON" si un metronom poate fi auzit in
boxe/casti.



Daca metronomul este pornit (**`isMetronomeOn`** true),
imaginea de metronome toggle se schimba in varianta
pentru "OFF" si metronomul nu se va mai auzi.



BPM-ul metronomului reprezinta tempo-ul pentru cele 4 batai (patrimi), care pot fi auzite. La apasarea butoanelor "-" sau "+" acest tempo poate fi scazut sau crescut de la 10 BPM la 120BPM. Valoarea BPM-ului va fi afisata pe ecran si salvata in `sessionStorage` ca **`bpm`** pentru mentinerea setarilor dupa reincarcarea aplicatiei. Aceste functionalitati sunt indeplinite de functiile **`increaseBPMByTen`** si **`decreaseBPMByTen`** din **`audioFunctions.js`**.



PAGINA PRINCIPALA



CUPRINS

Functionalitati curente

OSCILLATOR – reprezentat prin `oscillatorTypeSetter.jsx`, incarcat in `screen.jsx`. Componenta ofera posibilitatea de a schimba tipului oscilatorului folosit de sintetizator.

La apasarea butoanelor “-” sau “+”, functiile **`handleOscillatorDecrement`** sau **`handleOscillatorIncrement`** vor fi apelate, avand ca scop schimbarea setarilor sintetizatorului folosit pentru producerea sunetelor.



Aceste functii, trimise ca props componenteii **`oscillatorTypeSetter`**, seteaza obiectul **`synth`** cu setari luate din array-ul **`synthSettings`**, folosind incrementarea si decrementarea pe indexul setarii selectate din acest array.

Valorile pe care le poate schimba sunt (simplificat) **`[sine, triangle, square]`**. Numele acestor valori sunt preluate din `synthNames` si vor fi afisate pe ecran.

Indexul setat poate merge de la 0 pana la maxim 2, corespunzand celor 3 setari disponibile curent.

Valoarea indexului va fi salvata in `sessionStorage` ca **`synthIndex`** pentru mentinerea setarilor dupa reincarcarea aplicatiei.



PAGINA PRINCIPALA



CUPRINS

Functionalitati curente

OCTAVE – reprezentat prin `octaveSetter.jsx`, incarcat in `screen.jsx`.

Componenta ofera posibilitatea de a schimba octava de inceput folosita in producerea sunetelor de catre sintetizator.

La apasarea butoanelor “-” sau “+”, functiile **`handleOctaveDecrement`** sau **`handleOctaveIncrement`** vor fi apelate, avand ca scop schimbarea setarilor componentei Piano, folosita pentru transmiterea notelor catre sintetizator.



Aceste functii, trimise ca props componentei **`octaveSetter`**, seteaza state-ului **`octave`** cu valoarea setata, pentru a fi trimisa mai apoi ca props pentru componenta Piano.

Octava de start setata setat poate merge de la 0 pana la maxim 7, tinand cont de faptul ca pe claviatura pianului sunt doua octave, avand range efectiv de la C0 la C8, acoperind intreaga gama posibila a unui pian.

Octava de start va fi afisata pe ecran.

Valoarea octavei de start va fi salvata in `sessionStorage` ca **`octave`** pentru mentinerea setarilor dupa reincarcarea aplicatiei.



PAGINA PRINCIPALA



CUPRINS

Functionalitati curente

PIANO – reprezentat prin npm-ul react-piano, incarcat in home.jsx.

Componenta are ca functionalitate inregistrarea inputului de la utilizator, sub forma apasarii tastelor de claviatura.

Acestea pot fi apasate fie cu mouse-ul (left-click), fie prin tastarea scurtaturilor afisate pe ecran pentru taste.

Pianul are 18 taste albe si 12 taste negre (accidentale), definite prin range-ul primit ca props. Scurtaturile de taste pentru note pot fi schimbate, din props.

Latimea pianului este trimisa ca props. Pentru a permite un comportament relativ responsive, latimea este calculata diferit pentru doua breakpoint-uri, ecrane mai mari de 1400px si ecrane mai mici de 1400px.

Producerea sunetelor de la sintetizator se face prin doua functii:

- **playNote** activata la apasarea unei taste de pian, porneste producerea sunetului dorit.
- **stopNote** activata la oprirea apasarii unei taste, opreste producerea sunetului dorit.

```
let pianoSize;
const getSize = () => {
  if (window.innerWidth > 1400) {
    pianoSize = 0.67 * window.innerWidth;
  } else {
    pianoSize = 0.83 * window.innerWidth;
  }
  return {
    width: window.innerWidth,
  };
};
// eslint-disable-next-line no-unused-vars
const [windowSize, setWindowSize] = useState(getSize());

//getting the window size for piano keyboard resize
useEffect(() => {
  const handleResize = () => setWindowSize(getSize());
  window.addEventListener("resize", handleResize);
  return () => window.removeEventListener("resize", handleResize);
  // eslint-disable-next-line react-hooks/exhaustive-deps
}, []);
```

04

**Functionalitati viitoare
si imbunatatiri**



PAGINA PRINCIPALA



CUPRINS

Functionalitati viitoare si imbunatatiri

Oscillator

- Introducerea unor setari mai complexe pentru producerea sunetelor.
- Salvarea setarilor si posibilitatea crearii de oscilatoare custom.

Octave Setter

- La schimbarea octavei, daca anumite note sunt cantate, oscilatorul nu va opri notele activate si aceste note vor ramane pornite pana la refresh-ul paginii.

Piano

- Introducerea a mai multor scurtaturi de tastatura.
- Implementarea MIDI support.

Aplicatie

- Fortarea folosirii aplicatiei in portrait mode pentru dispozitive mobile.

Cod

- Refactorizare cod, in special componenta home.jsx.

MULTUMESC!

Stamate Cezar Ionel