

**Academia de Studii Economice, București**

**Facultatea de Cibernetică, Statistică și Informatică Economică**

**Specializarea: Informatică Economică**

Proiect de practică

***Tema proiectului: Website cu jocuri***

**Cadrul didactic coordonator**: Gheorghe Mihai

**Student**: Simionescu Corina Andreea

**București**

**2024**

**Cuprins**

[**Introducere** 4](#_Toc167628743)

[**1.** **Prezentarea generală a societații** 5](#_Toc167628744)

[**2.** **Tehnologiile folosite în cadrul cursurilor Chess Coders** 6](#_Toc167628745)

[**2.1.** **HTML (HyperText Markup Language)** 6](#_Toc167628746)

[**2.2.** **CSS (Cascading Style Sheets)** 7](#_Toc167628747)

[**2.3.** **JavaScript** 8](#_Toc167628748)

[**2.4.** **React** 9](#_Toc167628749)

[**2.5.** **Tailwind CSS** 10](#_Toc167628750)

[**2.6.** **Vite** 11](#_Toc167628751)

[**2.7.** **Node.js** 12](#_Toc167628752)

[**2.8.** **NPM** 13](#_Toc167628753)

[**2.9.** **MongoDB** 13](#_Toc167628754)

[**2.10.** **MongoDB Compass** 14](#_Toc167628755)

[**2.11.** **Git** 14](#_Toc167628756)

[**2.12.** **Visual Studio Code** 15](#_Toc167628757)

[**3.** **Prezentarea proiectului de practică** 16](#_Toc167628758)

[**3.1.** **Rularea proiectului** 17](#_Toc167628759)

[**3.2.** **Structura proiectului** 18](#_Toc167628760)

[**3.3.** **Header-ul** 19](#_Toc167628761)

[**3.4.** **Sectiunea ‘Choose a game’** 21](#_Toc167628762)

[**3.5.** **Secțiunea ‘Tic-Tac-Toe’ (X si 0)** 23](#_Toc167628763)

[**3.6.** **Sectiunea ‘Guess The Color’** 28](#_Toc167628764)

[**3.7.** **Secțiunile About si Contact** 34](#_Toc167628765)

[**3.8.** **Footer-ul** 34](#_Toc167628766)

[**4.** **Concluzii** 35](#_Toc167628767)

[**5.** **Bibliografie** 35](#_Toc167628768)

**Lista de figuri**

[Figură 1 - Logo ChessCoders 6](#_Toc167568273)

[Figură 2 - Logo HTML 7](#_Toc167568274)

[Figură 3 - Logo CSS 8](#_Toc167568275)

[Figură 4 - Logo JavaScript 9](#_Toc167568276)

[Figură 5 - Logo React: 10](#_Toc167568277)

[Figură 6 - Logo tailwind css 11](#_Toc167568278)

[Figură 7 - Logo Vite 12](#_Toc167568279)

[Figură 8 - Logo Node.js 13](#_Toc167568280)

[Figură 9 - Logo npm 13](#_Toc167568281)

[Figură 10 - Logo MongoDB 14](#_Toc167568282)

[Figură 11 - Logo git 15](#_Toc167568283)

[Figură 12 - Logo Visual Studio Code 16](#_Toc167568284)

[Figură 13 - Games website 18](#_Toc167568285)

[Figură 14 - Structura directoarelor 19](#_Toc167568286)

[Figură 15 - Header 20](#_Toc167568287)

[Figură 16 - Cod pentru header 20](#_Toc167568288)

[Figură 17 - functia scrollToSection 21](#_Toc167568289)

[Figură 18 - Choose a game 22](#_Toc167568290)

[Figură 19 - cod pentru 'Choose a game' 22](#_Toc167568291)

[Figură 20 - Tic Tac Toe 24](#_Toc167568292)

[Figură 21 - Cod pentru Tic Tac Toe din App.jsx 24](#_Toc167568293)

[Figură 22 - Cod pentru Tic Tac Toe din TicTacToe.jsx 25](#_Toc167568294)

[Figură 23 - Cod Square.jsx 25](#_Toc167568295)

[Figură 24 - Guess the color 30](#_Toc167568296)

[Figură 25 - Cod pentru 'Guess the color' din App.jsx 30](#_Toc167568297)

[Figură 26 - cod pentru 'Guess the color' din GuessTheColor.jsx 30](#_Toc167568298)

[Figură 27 - About si Contact 35](#_Toc167568299)

[Figură 28 - Footer 35](#_Toc167568300)

# **Introducere**

Stagiul meu de practică a fost susținut în cadrul companiei Chess Coders, companie specializată în dezvoltarea aplicațiilor web.

În timpul stagiului de practică am avut oportunitatea de a lucra într-un mediu profesional și dinamic, în care am avut șansa de a învăța și de a contribui la un proiect real. Am avut șansa să mă implic în dezvoltarea de aplicații web, să înțeleg procesele implicate în crearea unui site web. Experiența mea la Chess Coders mi-a consolidat abilitățile practice și mi-a oferit o perspectivă valoroasă asupra industriei dezvoltării web.

# **Prezentarea generală a societații**



Figură 1 - Logo ChessCoders: https://www.chesscoders.com/

Chess Coders este un startup, o companie românească fondată în urmă cu câțiva ani de către cei 3 fondatori Bogdan Posedaru, Tiberiu Georgescu si Victor Ocnărescu. Aceștia au fost jucători profesioniști de șah, dar au realizat că din păcate nu se pot face bani din șah și s-au apucat de programare. Și asa s-a nascut Chess Coders.

Serviciile oferite de Chess Coders sunt aplicații web, site-uri web, platforme de e-commerce, produse software, platforme de e-learning și inovatii.

Printre multele produsele create de Chess Coders se numara și decont.one.ro, superliga.ro, trimbitasu.ro, retur.fancourier.ro, flip.ro.

# **Tehnologiile folosite în cadrul cursurilor Chess Coders**

Pe parcursul cursurilor ne-au fost prezentate numeroase tehnologii folosite pentru crearea unei aplicatii web sau a unui site web.

## **HTML (HyperText Markup Language)**



Figură 2 - Logo HTML:https://en.wikipedia.org/wiki/HTML#/media/File:HTML5\_logo\_and\_wordmark.svg

HyperText Markup Language (HTML) este un [limbaj de marcare](https://ro.wikipedia.org/wiki/Limbaj_de_marcare) utilizat pentru crearea paginilor web ce pot fi afișate într-un [browser](https://ro.wikipedia.org/wiki/Browser) (sau navigator). Scopul HTML este mai degrabă prezentarea informațiilor – paragrafe, fonturi, tabele ș.a.m.d. – decât descrierea semanticii documentului. În cadrul [dezvoltării web](https://ro.wikipedia.org/wiki/Dezvoltare_web) de tip [front-end](https://ro.wikipedia.org/wiki/Front-end_%C8%99i_back-end), HTML este utilizat împreună cu [CSS](https://ro.wikipedia.org/wiki/CSS) și [JavaScript](https://ro.wikipedia.org/wiki/JavaScript).

HTML constă într-o serie de elemente, pe care le folosești pentru a închide, sau înveli, diferite părți ale conținutului pentru a le face să apară într-un anumit fel, sau să acționeze într-un anumit fel. Etichetele de închidere pot transforma un cuvânt sau o imagine într-un hyperlink către altă parte, pot pune cuvintele în italic, pot face fontul mai mare sau mai mic, și așa mai departe.

## **CSS (Cascading Style Sheets)**



Figură 3 - Logo CSS: https://en.wikipedia.org/wiki/CSS#/media/File:CSS3\_logo\_and\_wordmark.svg

Cascading Style Sheets (CSS) este un limbaj de stil utilizat pentru a specifica prezentarea și stilizarea unui document scris într-un limbaj de marcare, cum ar fi HTML sau XML (inclusiv dialecte XML precum SVG, MathML sau XHTML). CSS este o tehnologie de bază a World Wide Web, alături de HTML și JavaScript.

CSS este conceput pentru a permite separarea conținutului de prezentare, inclusiv layout-ul, culorile și fonturile. Această separare poate îmbunătăți accesibilitatea conținutului; oferă mai multă flexibilitate și control în specificarea caracteristicilor de prezentare; permite mai multor pagini web să partajeze formatarea prin specificarea CSS-ului relevant într-un fișier separat .css, ceea ce reduce complexitatea și repetiția în conținutul structural; și permite fișierului .css să fie memorat în cache pentru a îmbunătăți viteza de încărcare a paginii între paginile care partajează fișierul și formatarea acestuia.

## **JavaScript**



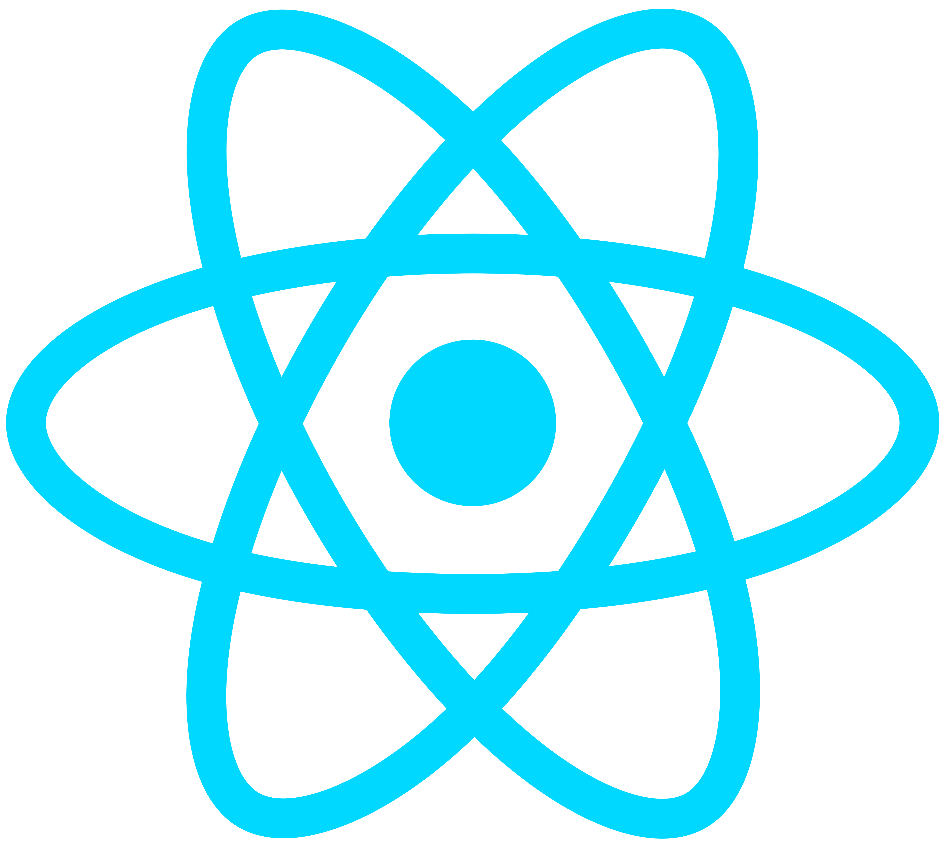
Figură 4 - Logo JavaScript: https://i.pinimg.com/originals/98/0b/a0/980ba07956ccfdf81b5060b1795cdcad.png

JavaScript (JS) este un [limbaj de programare](https://ro.wikipedia.org/wiki/Limbaj_de_programare) [orientat obiect](https://ro.wikipedia.org/wiki/Programare_orientat%C4%83_pe_obiecte) bazat pe conceptul [prototipurilor](https://ro.wikipedia.org/wiki/Prototip_(POO)). Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către [browser](https://ro.wikipedia.org/wiki/Browser_web). Limbajul este binecunoscut pentru folosirea sa în construirea [siteurilor web](https://ro.wikipedia.org/wiki/Website), dar este folosit și pentru accesul la obiecte încapsulate (embedded objects) în alte aplicații. A fost dezvoltat inițial de către [Brendan Eich](https://ro.wikipedia.org/wiki/Brendan_Eich) de la [Netscape Communications Corporation](https://ro.wikipedia.org/wiki/Netscape_Communications_Corporation) sub numele de Mocha, apoi LiveScript, și denumit în final JavaScript.

În ciuda numelui și a unor similarități în sintaxă, între JavaScript și limbajul Java nu există nicio legătură. Ca și [Java](https://ro.wikipedia.org/wiki/Java), JavaScript are o sintaxă apropiată de cea a [limbajului C](https://ro.wikipedia.org/wiki/Limbajul_de_programare_C), dar are mai multe în comun cu [limbajul Self](https://ro.wikipedia.org/w/index.php?title=Limbajul_Self&action=edit&redlink=1) decât cu [Java](https://ro.wikipedia.org/wiki/Limbajul_Java).

Până la începutul lui 2005, ultima versiune existentă a fost JavaScript 1.5, care corespunde cu Ediția a 3-a a ECMA-262, [ECMAScript](https://ro.wikipedia.org/wiki/ECMAScript), cu alte cuvinte, o ediție standardizată de JavaScript. Versiunile de [Mozilla](https://ro.wikipedia.org/wiki/Mozilla) începând cu 1.8 Beta 1 au avut suport pentru [E4X](https://ro.wikipedia.org/w/index.php?title=E4X&action=edit&redlink=1), care este o extensie a limbajului care are de a face cu [XML](https://ro.wikipedia.org/wiki/XML), definit în standardul ECMA-357. Versiunea curentă de Mozilla, 1.8.1 (pe care sunt construite [Firefox](https://ro.wikipedia.org/wiki/Firefox) și [Thunderbird](https://ro.wikipedia.org/wiki/Thunderbird) versiunile 2.0) suportă JavaScript versiunea 1.7.

## **React**



Figură 5 - Logo React: https://en.wikipedia.org/wiki/React\_(JavaScript\_library)#/media/File:React\_Logo\_SVG.svg

React.js este o bibliotecă JavaScript utilizată pentru crearea interfețelor utilizator web dinamice și interactive. A fost dezvoltată de Facebook și este folosită pentru construirea de aplicații web moderne. React.js este construită în jurul unui concept numit “componente”, care reprezintă bucăți de cod reutilizabile și independent gestionate, ceea ce face ca dezvoltarea aplicațiilor web să fie mai ușoară și mai eficientă.

Există mai multe motive pentru a folosi React.js în dezvoltarea aplicațiilor web. Iată câteva dintre cele mai importante:

* Reutilizarea codului: React.js încurajează dezvoltatorii să creeze componente reutilizabile, ceea ce poate reduce semnificativ timpul de dezvoltare și poate duce la cod mai curat și mai ușor de întreținut.
* Performanța: React.js are o performanță excelentă datorită faptului că utilizează Virtual DOM, o tehnică care optimizează modul în care sunt actualizate paginile web, fără a necesita o reîncărcare completă a paginii.
* Comunitatea mare: React.js are o comunitate mare și activă de dezvoltatori, care împărtășesc cunoștințe și dezvoltă instrumente și biblioteci suplimentare care pot fi utilizate împreună cu React.js.

React.js diferă față de alte framework-uri prin abordarea sa modulară și component-based. În timp ce alte framework-uri web se concentrează pe construirea de pagini web întregi, React.js se concentrează pe construirea de componente reutilizabile. Această abordare modulară face ca dezvoltarea să fie mai ușoară și mai eficientă, deoarece dezvoltatorii pot construi și testa componente independent, apoi le pot folosi în diferite aplicații web fără a fi nevoie să le rescrie. De asemenea, React.js utilizează Virtual DOM pentru a gestiona actualizările paginii, ceea ce face ca aplicațiile să fie mai rapide și mai eficiente decât cele construite folosind alte framework-uri.

## **Tailwind CSS**



Figură 6 - Logo tailwind css: https://en.wikipedia.org/wiki/Tailwind\_CSS#/media/File:Tailwind\_CSS\_logo.svg

Tailwind CSS este un framework utilitar (low level) creat pentru a construi mai rapid interfețe web, fără nevoia de a scrie cod CSS custom și fără a ieși din HTML. Mai exact, Tailwind folosește arhitectura Atomic CSS care presupune crearea de clase mici și simple al căror scop e sa faca un singur lucru.

Folosirea Tailwind CSS seamăna cu scrierea codului CSS inline, diferența fiind faptul că scrie clase, nu proprietati CSS. Acest lucru poate încărca destul de mult layout-ul HTML dar poate face dezvoltarea mai rapidă.

## **Vite**



Figură 7 - Logo Vite: https://en.wikipedia.org/wiki/Vite\_(software)#/media/File:Vitejs-logo.svg

Vite este un server de dezvoltare local scris de Evan You, creatorul Vue.js, și este folosit implicit de Vue și pentru șabloanele de proiect React. Are suport pentru TypeScript și JSX. Intern, folosește Rollup și esbuild pentru împachetare. Monitorizează fișierele pe măsură ce sunt editate și, la salvarea fișierului, browser-ul web reîncarcă codul editat printr-un proces numit înlocuire a modulului în timp real (HMR), care funcționează prin reîncărcarea fișierului specific modificat folosind module ES6 (ESM) în loc să recompileze întreaga aplicație. Vite oferă suport încorporat pentru redarea pe server (SSR). Implicit, ascultă pe portul TCP 5173.

## **Node.js**



Figură 8 - Logo Node.js: https://en.wikipedia.org/wiki/Node.js#/media/File:Node.js\_logo.svg

Node.js este un mediu de execuție al limbajului de programare JavaScript, care îți permite să rulezi aplicații JavaScript independent de un browser web. Inițial, JavaScript a fost conceput pentru a adăuga interactivitate paginilor web, rulând exclusiv în browserul utilizatorului. Cu apariția Node.js, a devenit posibil să scrii software de server sau aplicații care funcționează direct pe un sistem de operare, dezvoltând semnificativ capabilitățile JavaScript.

## **NPM**



Figură 9 - Logo npm: https://en.wikipedia.org/wiki/Npm#/media/File:Npm-logo.svg

npm este un manager de pachete pentru limbajul de programare JavaScript, întreținut de npm, Inc., o filială a GitHub. npm este managerul de pachete implicit pentru mediul de execuție JavaScript Node.js și este inclus ca o caracteristică recomandată în instalatorul Node.js. El constă dintr-un client de linie de comandă, numit și npm, și o bază de date online de pachete publice și plătite, numită registrul npm. Registrul este accesat prin intermediul clientului, iar pachetele disponibile pot fi navigabile și căutate prin intermediul site-ului npm. Managerul de pachete și registrul sunt gestionate de npm, Inc.

## **MongoDB**



Figură 10 - Logo MongoDB: https://en.wikipedia.org/wiki/MongoDB#/media/File:MongoDB\_Fores-Green.svg

MongoDB este o bază de date NoSQL open-source orientată pe documente. Acestă bază de date beneficiază de suport din partea companiei [10gen](https://ro.wikipedia.org/w/index.php?title=10gen&action=edit&redlink=1). MongoDB face parte din familia de sistemelor de baze de date [NoSQL](https://ro.wikipedia.org/w/index.php?title=NoSQL&action=edit&redlink=1). Diferența principală constă în faptul că stocarea datelor nu se face folosind tabele precum într-o [bază de date relațională](https://ro.wikipedia.org/w/index.php?title=Baz%C4%83_de_date_rela%C8%9Bional%C4%83&action=edit&redlink=1), MongoDB stochează datele sub formă de documente [JSON](https://ro.wikipedia.org/wiki/JSON) cu scheme dinamice.

## **MongoDB Compass**

MongoDB Compass este o interfață grafică utilizată pentru a interacționa cu baza de date MongoDB. Acesta oferă o modalitate vizuală și intuitivă de a explora și de a gestiona datele stocate în bazele de date MongoDB, inclusiv vizualizarea, interogarea și manipularea datelor. Compass poate fi folosit pentru a crea și a executa interogări, pentru a analiza structura colecțiilor de date, pentru a importa și exporta date, și pentru a gestiona aspecte precum indexarea și controlul accesului la date.

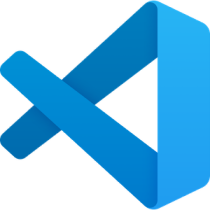
## **Git**



Figură 11 - Logo git: https://en.wikipedia.org/wiki/Git#/media/File:Git-logo.svg

Git este un sistem de control al versiunilor care rulează pe majoritatea platformelor, inclusiv Linux, POSIX, Windows și OS X. Ca și [Mercurial](https://ro.wikipedia.org/wiki/Mercurial), Git este un sistem distribuit și nu întreține o bază de date comună. Este folosit în echipe de dezvoltare mari, în care membrii echipei acționează oarecum independent și sunt răspândiți pe o arie geografică mare. Git este dezvoltat și întreținut de Junio Hamano, fiind publicat sub licență [GPL](https://ro.wikipedia.org/wiki/GPL_(licen%C8%9Ba)) și este considerat [software liber](https://ro.wikipedia.org/wiki/Software_liber).

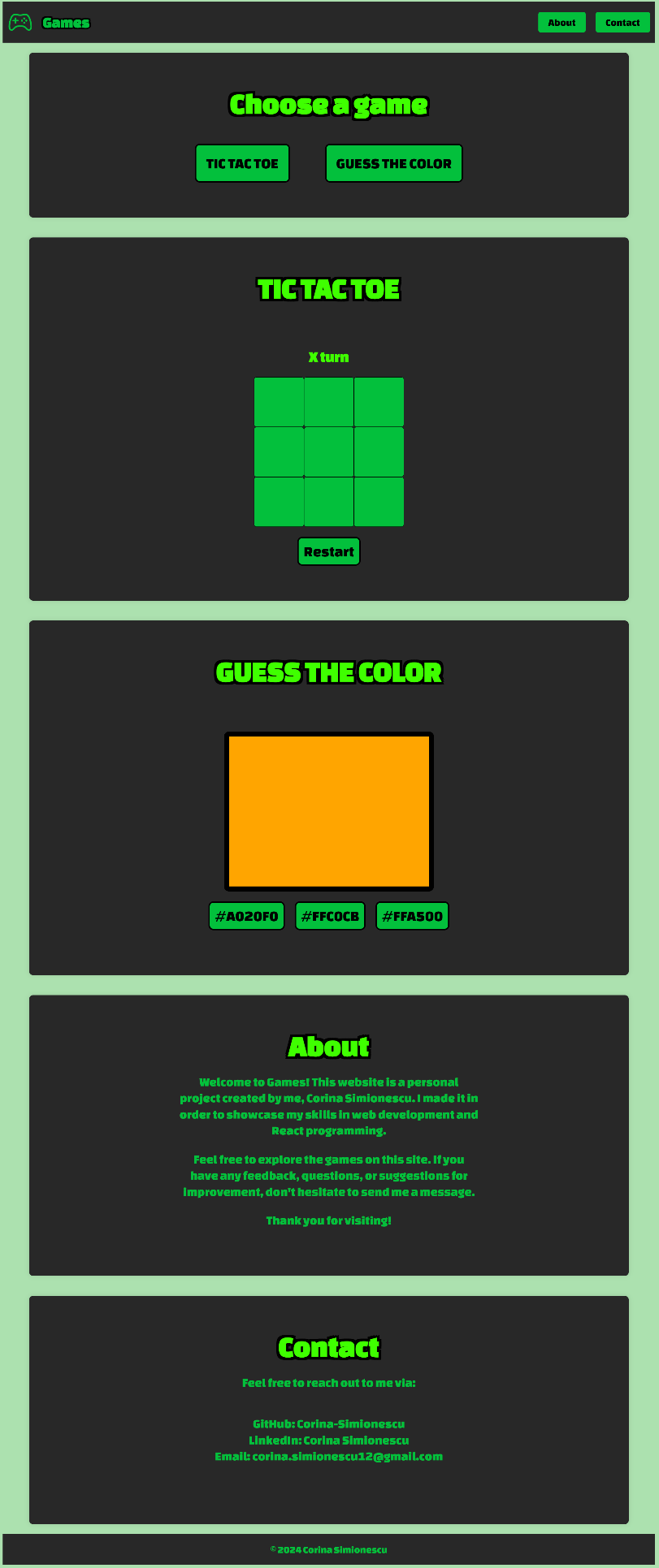
## **Visual Studio Code**



Figură 12 - Logo Visual Studio Code: https://en.wikipedia.org/wiki/Visual\_Studio\_Code#/media/File:Visual\_Studio\_Code\_1.35\_icon.png

Visual Studio Code, cunoscut și sub denumirea de VS Code, este un editor de cod sursă dezvoltat de Microsoft pentru Windows, Linux, macOS și browsere web. Funcționalitățile includ suport pentru depanare, evidențierea sintaxei, completarea inteligentă a codului, fragmente de cod, refacturare a codului și control înglobat al versiunilor cu Git. Utilizatorii pot schimba tema, scurtăturile de tastatură, preferințele și pot instala extensii care adaugă funcționalități.

# **Prezentarea proiectului de practică**



Figură 13 - Games website

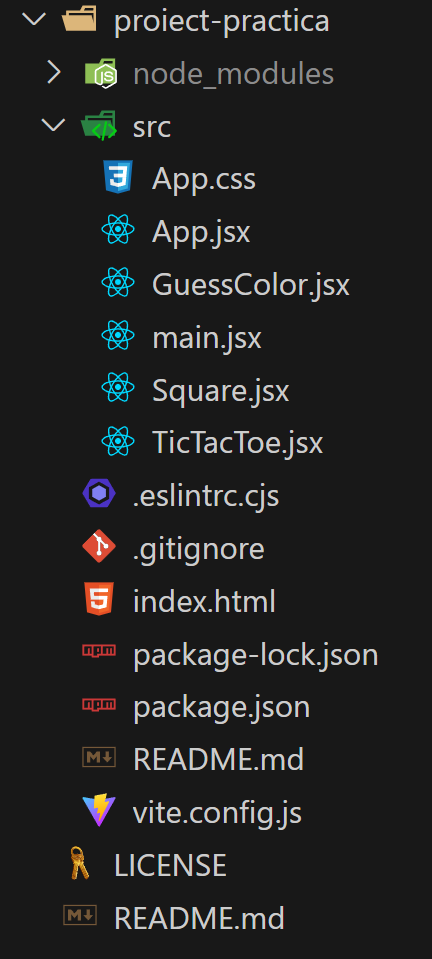
Pentru proiectul de practică am ales sa fac un website cu jocuri astfel încat să pun în practică o parte din lucrurile învățate de-a lungul stagiului de practică.

Tehnologiile folosite pentru realizarea acestui webstite sunt: HTML, CSS, JavaScript, React, Vite, npm, GIT, VS Code.

## **Rularea proiectului**

Se intră pe link-ul <https://github.com/Corina-Simionescu/proiect-de-practica>, se apasă pe butonul verde ‘Code’ si din secțiunea HTTPS se copiaza URL-ul. Se intră în folderul în care se dorește salvarea proiectului și se realizează următorii pași: click dreapta -> git bash here, se observă că se deschide terminalul si se scrie comanda ‘git clone <URL-ul copiat>’, se dă ENTER. Pentru a continua, se asigură faptul ca este instalată utlima versiune de node.js. Se deschide folderul proiectului in Visual Studio Code ( sau alt editor/IDE). Se deschide terminalul și se rulează comanda ‘npm i’ astfel încât să se instaleze toate dependințele necesare proiectului. Dupa care se rulează comanda ‘npm run vite’ și se accesează linkul de la Local. În final, ar trebui să se deschidă automat browserul cu linkul accesat și se poate oberva proiectul meu.

## **Structura proiectului**



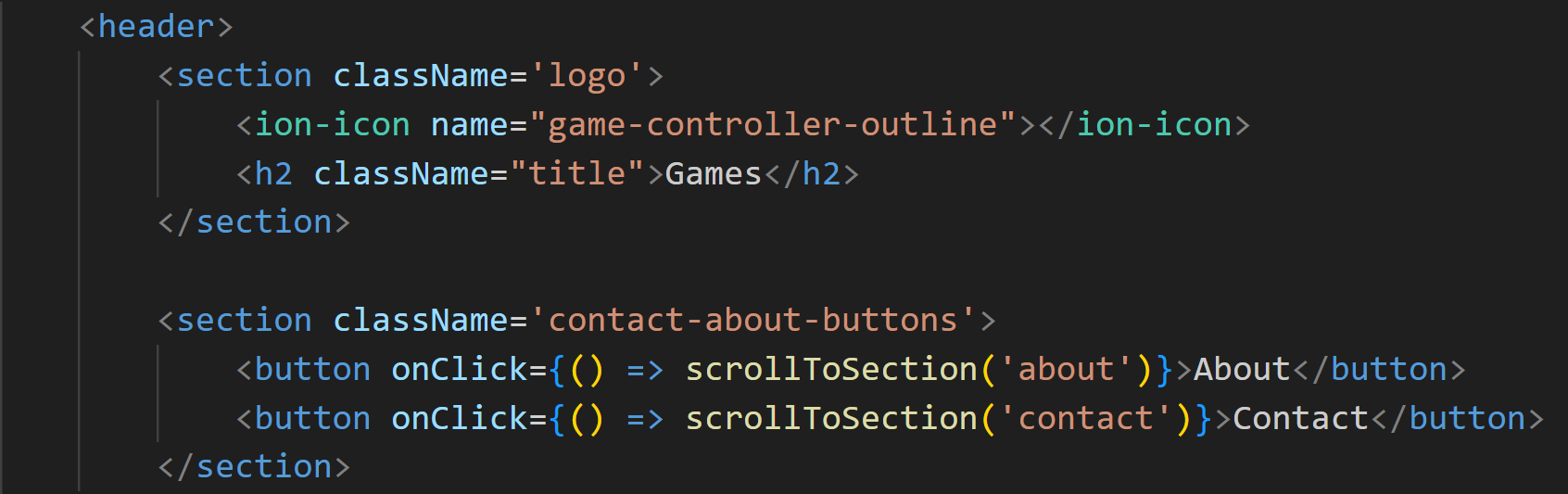
Figură 14 - Structura directoarelor

* package.json: este un fisier în care se află metadata despre proiect. Sunt informații precum numele proiectului, versiunea, dependințe. Atunci când se rulează comanda ‘npm i’, se instalează dependințele prezente în package.json .
* package-lock.json: este un fișier în care se află informațiile despre dependințele proiectului, precum versiunea, tree-ul dependinței (adică se văd toate dependințele care au format dependința care s-a instalat).
* node\_modules: este un folder in care Node.js stochează modulele si dependințele utilizate în proiect.
* .gitignore: este un fișier folosit în proiectele Git pentru a specifica fișierele și folderele care nu ar trebui incluse în repository.
* index.html: este fișierul HTML principal al proiectului. Scopul său este de a furniza structura de bază a paginii web și de a încărca aplicația React în browser.
* src: este folderul în care am scris efectiv codul pentru proiectul meu. Aici sunt plasate toate fișierele si componentele React.
  + main.jsx: este fișierul de intrare care încarcă aplicația în DOM.
  + App.jsx: este componenta principala și definește structura de bază a aplicației.
  + TicTacToe.jsx: este componenta în care am creat jocul X si 0.
  + Square.jsx: este componenta în care este reprezentat un patrat necesar jocului X si 0.
  + GuessColor.jsx: este componenta în care am creat jocul Guess The Color.
  + App.css: este fisierul în care se află tot codul CSS al proiectului. Acesta defineste aspectul vizual și prezentarea elementelor în pagina.

## **Header-ul**



Figură 15 - Header

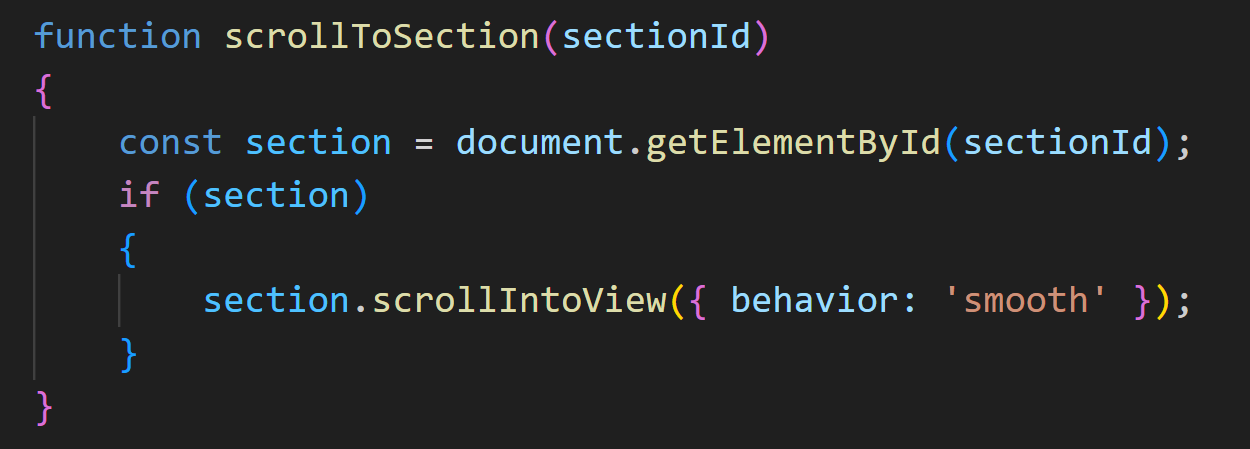


Figură 16 - Cod pentru header

Headerul este format din 4 elemente: pictograma, h2 si 2 butoane.

Pictograma am luat-o de pe site-ul <https://ionic.io/ionicons>.

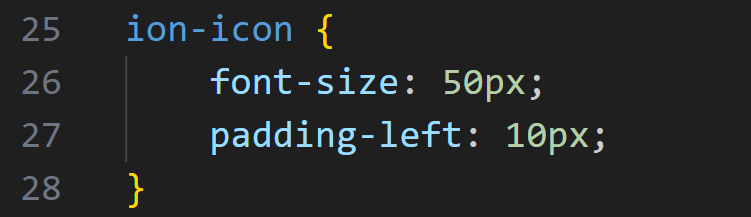
La cele doua butoane, se poate observa că am folosit event handler-ul onClick care primește ca parametru un arrow function anonim. Aceasta arrow function returnează funcția scrollToSection(‘id sectiune’). Daca aș fi scris direct: onClick={ scrollToSection(‘id sectiune’) }, ar fi fost greșit, deoarecec funcția s-ar fi apelat chiar dacă nu aș fi dat click pe buton. Asa că am adăugat acel arrow function astfel încat funcția să fie apelată numai atunci când se dă click pe butonul respectiv.

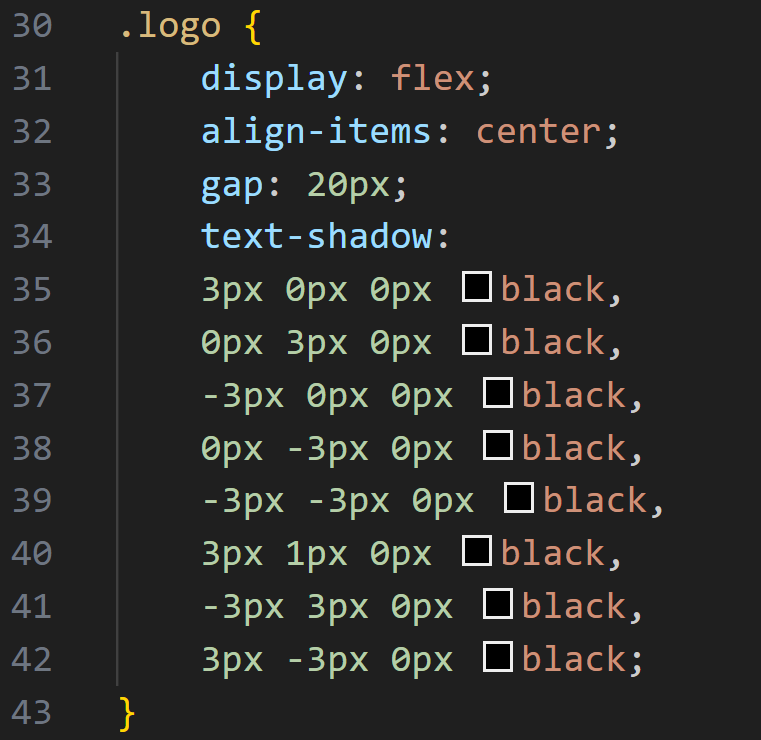
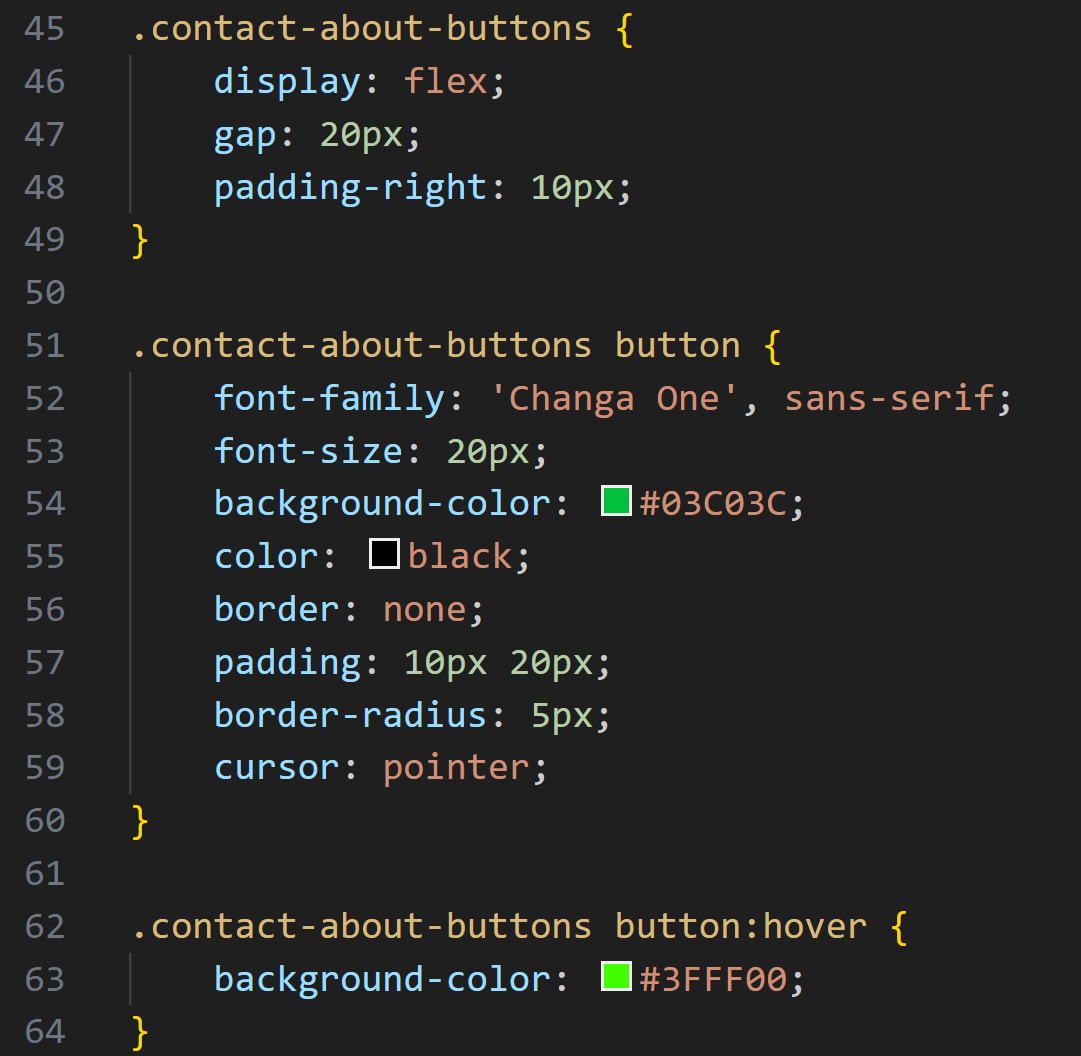


Figură 17 - functia scrollToSection

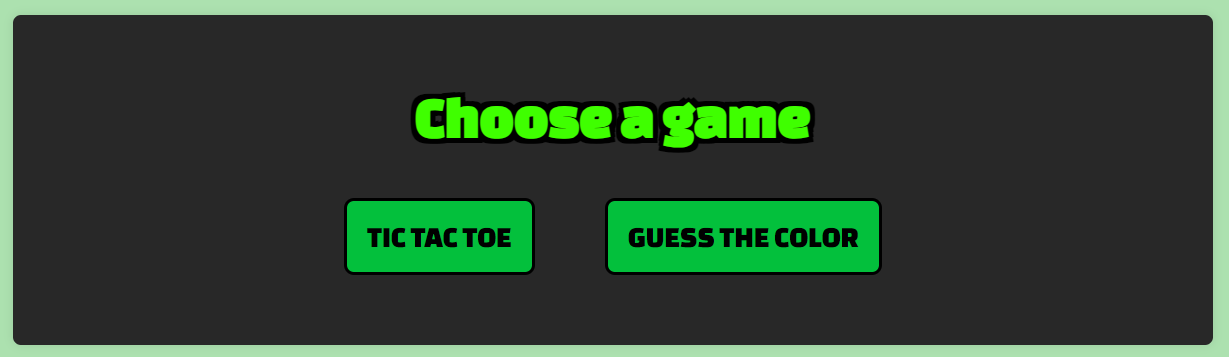
Scopul funcției scrollToSection este de a trimite utilizatorul la secțiunea dorită după apăsarea butonului. Adică dacă se apasă butonul ‘About’, utilizatorul să fie trimis la sectiunea ‘About’ a paginii.

* Stilizare

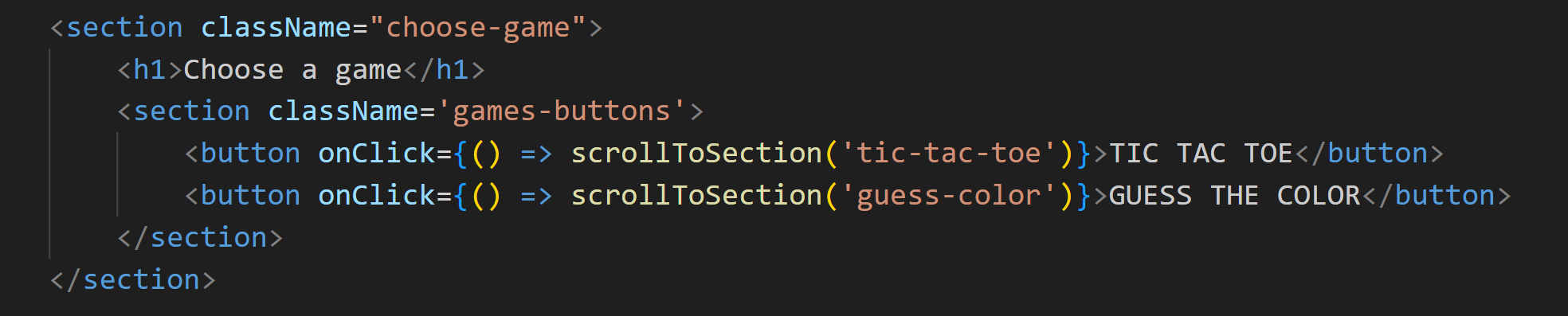
 

## **Sectiunea ‘Choose a game’**



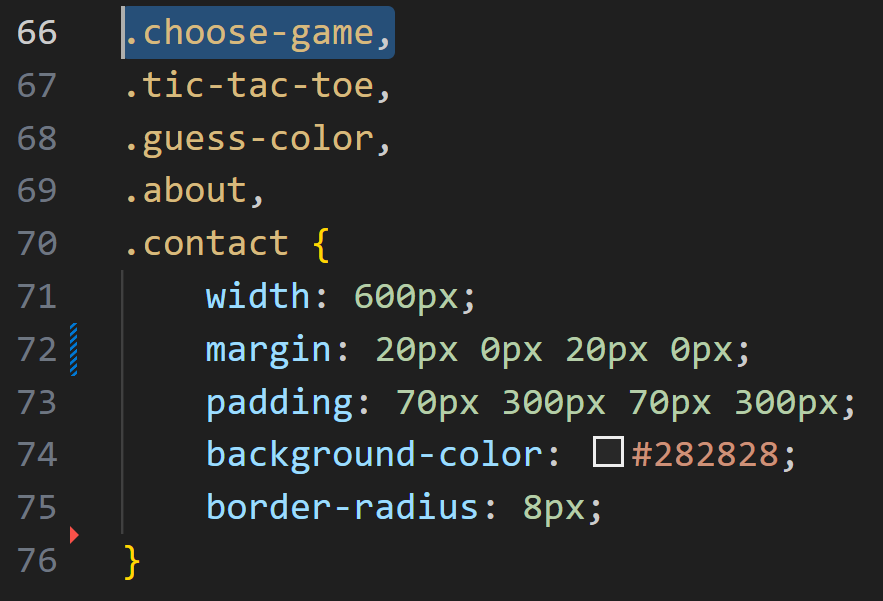
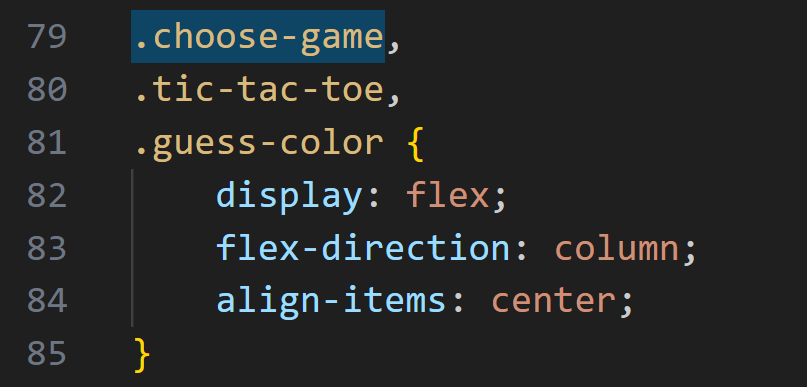
Figură 18 - Choose a game

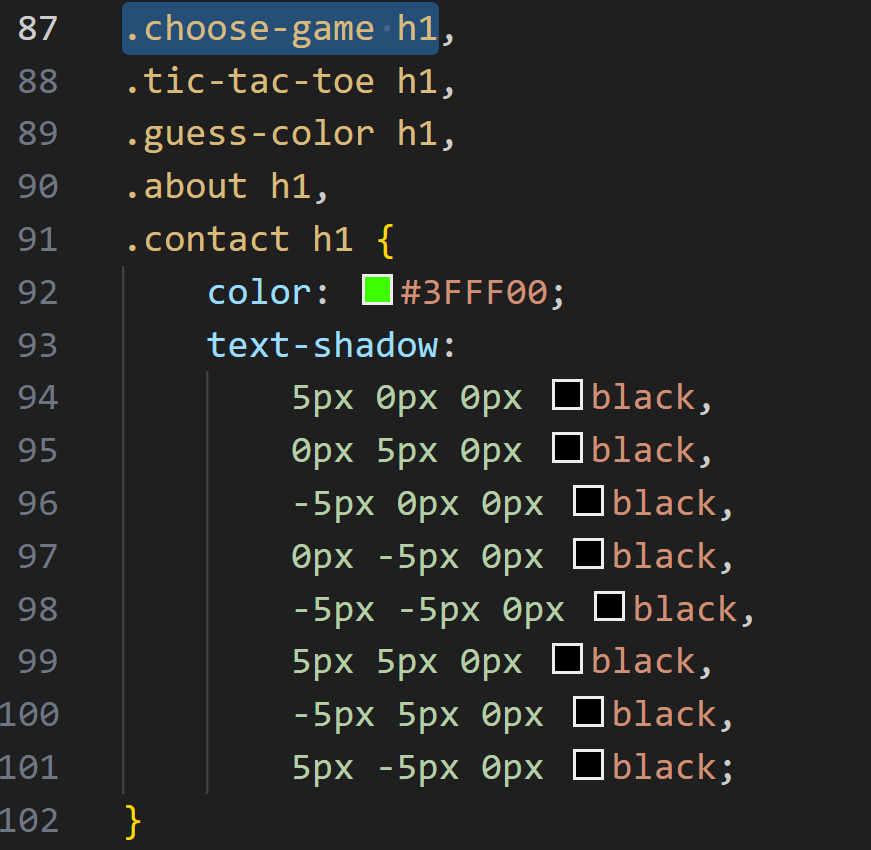
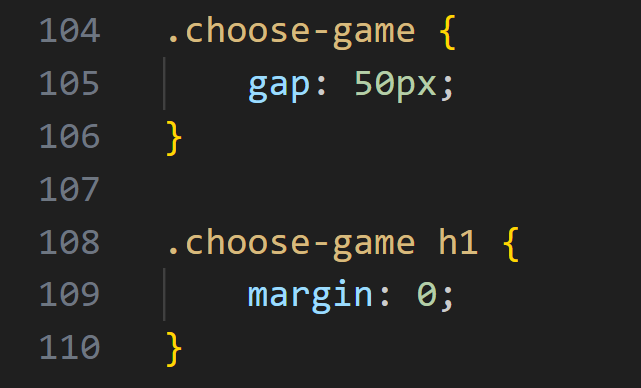
****

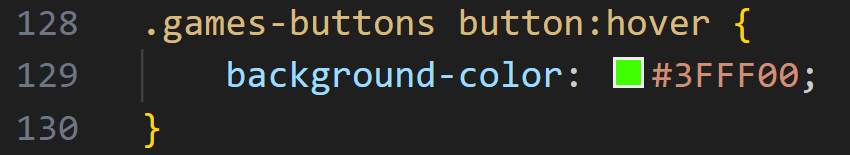
Figură 19 - cod pentru 'Choose a game'

Această sectiune este formată dintr-un header1 si doua butoane care dacă sunt apasate trimit utilizatorul catre secțiunea corespunzatoare jocului.

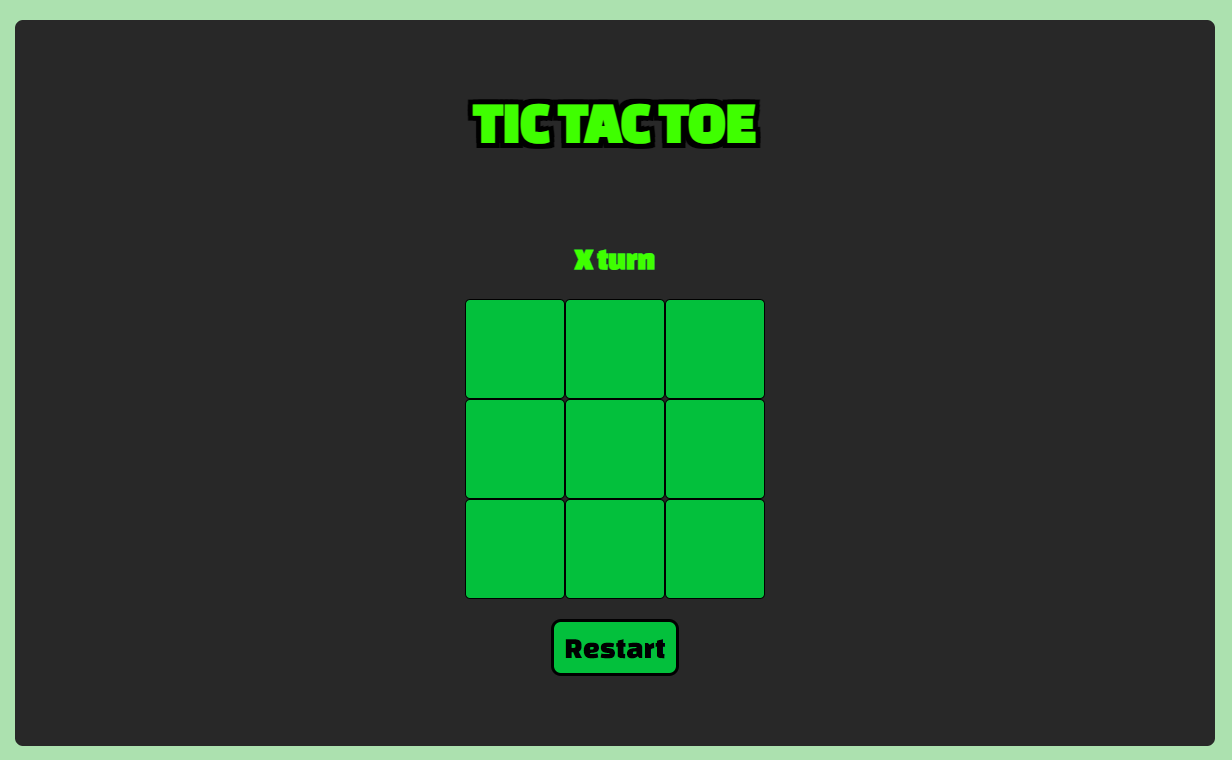
* Stilizare

## **Secțiunea ‘Tic-Tac-Toe’ (X si 0)**



Figură 20 - Tic Tac Toe



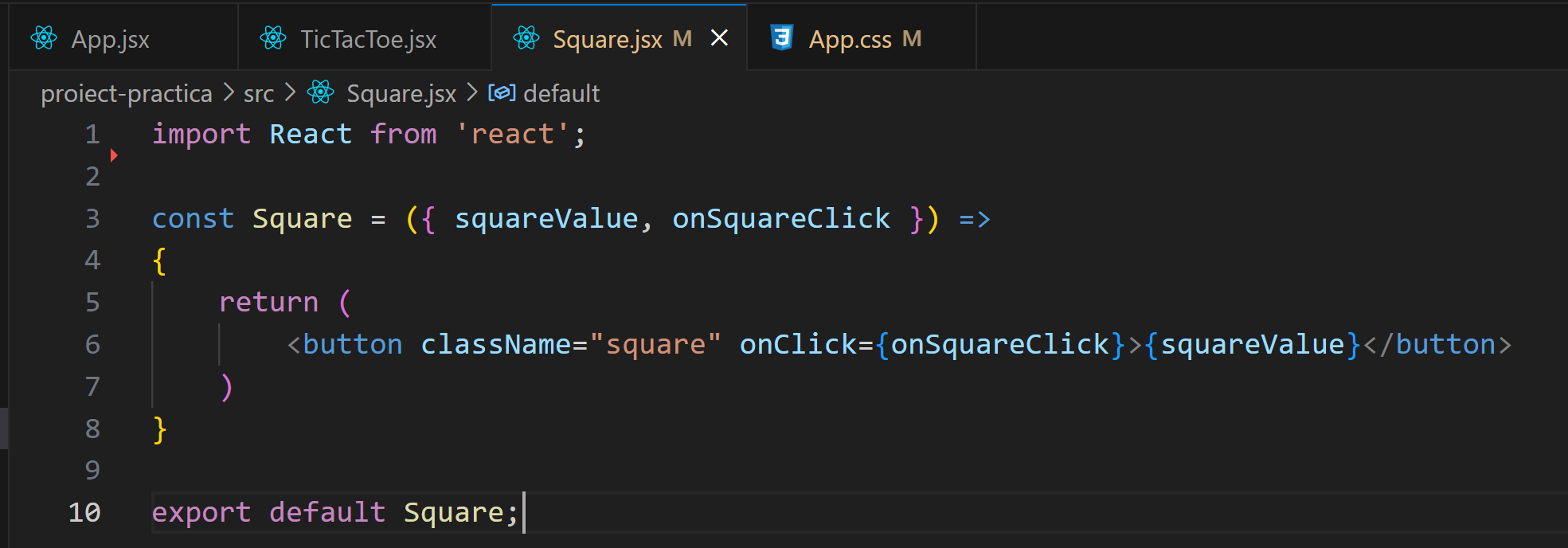
Figură 21 - Cod pentru Tic Tac Toe din App.jsx

In App.jsx doar este inserată componenta React TicTacToe prin intermediul sintaxei JSX.



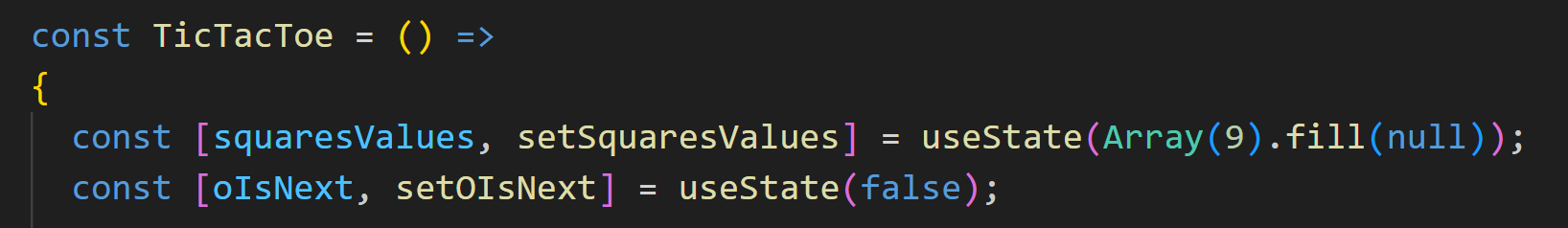
Figură 22 - Cod pentru Tic Tac Toe din TicTacToe.jsx

In TicTacToe.jsx se afla h1 (titlul ‘TIC TAC TOE’), h2 (mesajul care spune al cui este randul și cine a castigat), 9 inserări ale componentei Square, ce reprezintă un patrat în care poate fi inserat X sau 0 și se mai afla butonul de restart.

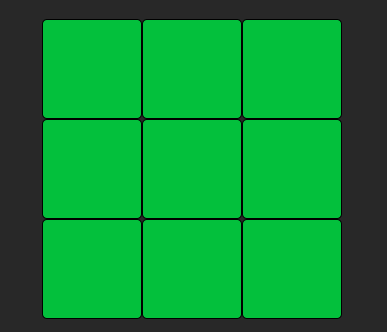


Figură 23 - Cod Square.jsx

squaresValues este o listă de 9 elemente (9 patrate) ale caror valori pot fi ‘null’,’X’ sau ‘O’.



Initial, lista squaresValues are cele 9 elemente egale cu ‘null’, adica patratele sunt goale.



Atunci cand dau click pe oricare din cele 9 patrate, se apelează funcția handleSquareClick(indexul elementului in listă).

Sa presupunem ca dau click pe patratul din stanga sus, primul element din lista squaresValues, cel de la indexul 0.

Se apelează handleSquareClick(0)



Mai întâi se verifică daca pătratul pe care am dat click este diferit de null (daca era deja X sau 0 pe acel patrat, nu ar fi avut sens să i se mai schimbe valoarea) sau daca a fost gasit deja un castigator și nu se returnează nimic daca una dintre cele doua situații este adevarată.

squaresValues: [null, null, null, null, null, null, null, null, null, null]

0 1 2 3 4 5 6 7 8 9

Pentru a respecta principiul imutabilitații, am creat o copie a listei squaresValues pe care voi face modificări și în final voi seta lista inițială squaresValues cu copia schimbată.

Verific al cui este rândul și în funcție de asta setez pătratul cu X sau cu O. În cazul presupus de mine, ar fi prima miscare din joc, adica valoarea lui oIsNext este false, ceea ce inseamna ca X este cel care va apărea pe pătrat => updatedSquaresValues[0] = ‘X’;. Prin linia setSquaresValues(updatedSquaresValues), squaresValues devine egal cu updatedSquaresValues, adică:

squaresValues: [‘X’, null, null, null, null, null, null, null, null, null]

0 1 2 3 4 5 6 7 8 9

De asemena, oIsNext devine true, ceea ce inseamnă că urmatorul pătrat pe care se va da click va lua valoarea ‘0’.

Ne putem imagina că în Square.jsx se întâmplă asta:

const Square = ({ squareValue, onSquareClick }) =>

{

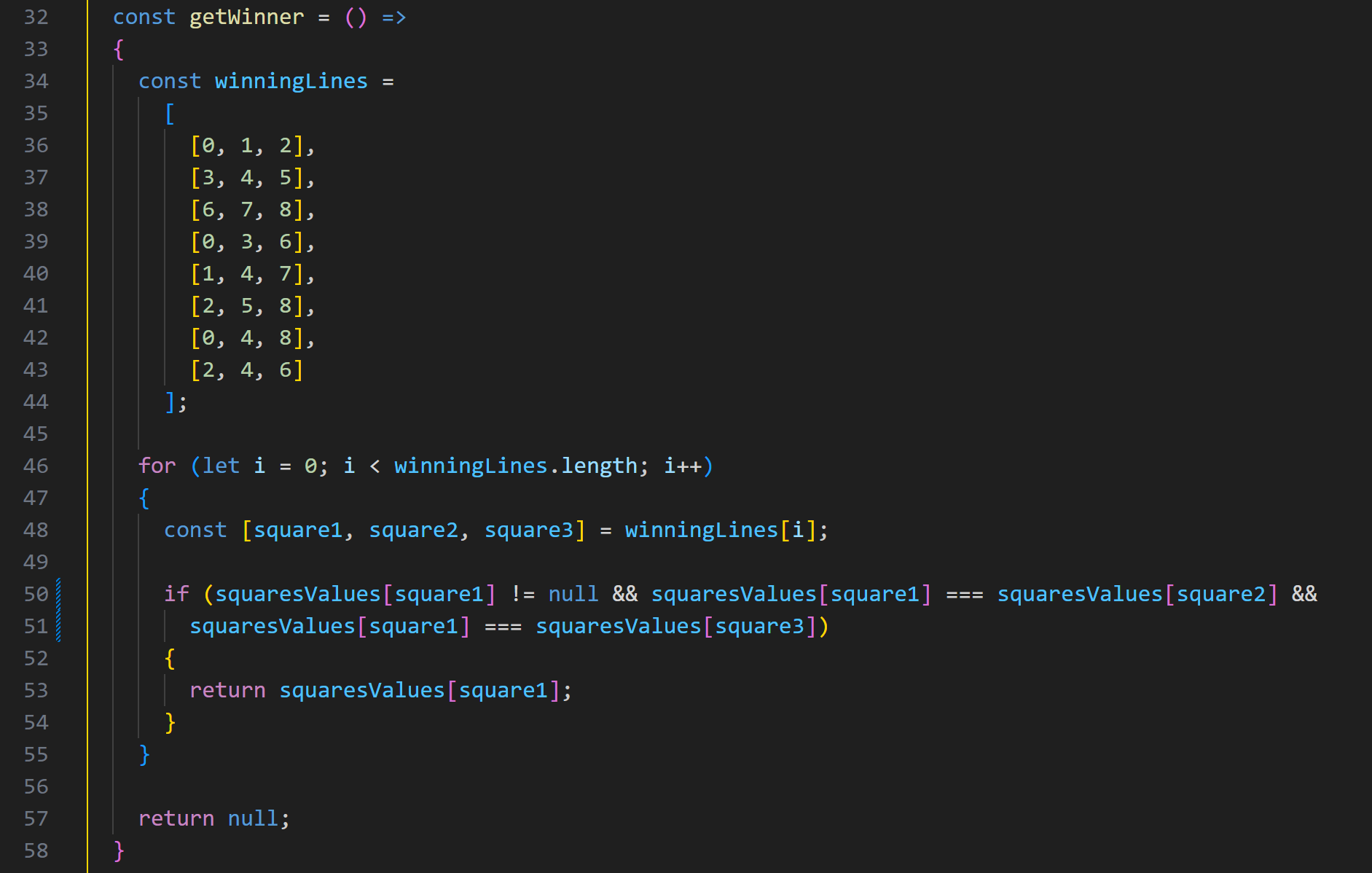
    return (

        <button className="square" onClick={onSquareClick}>X</button>

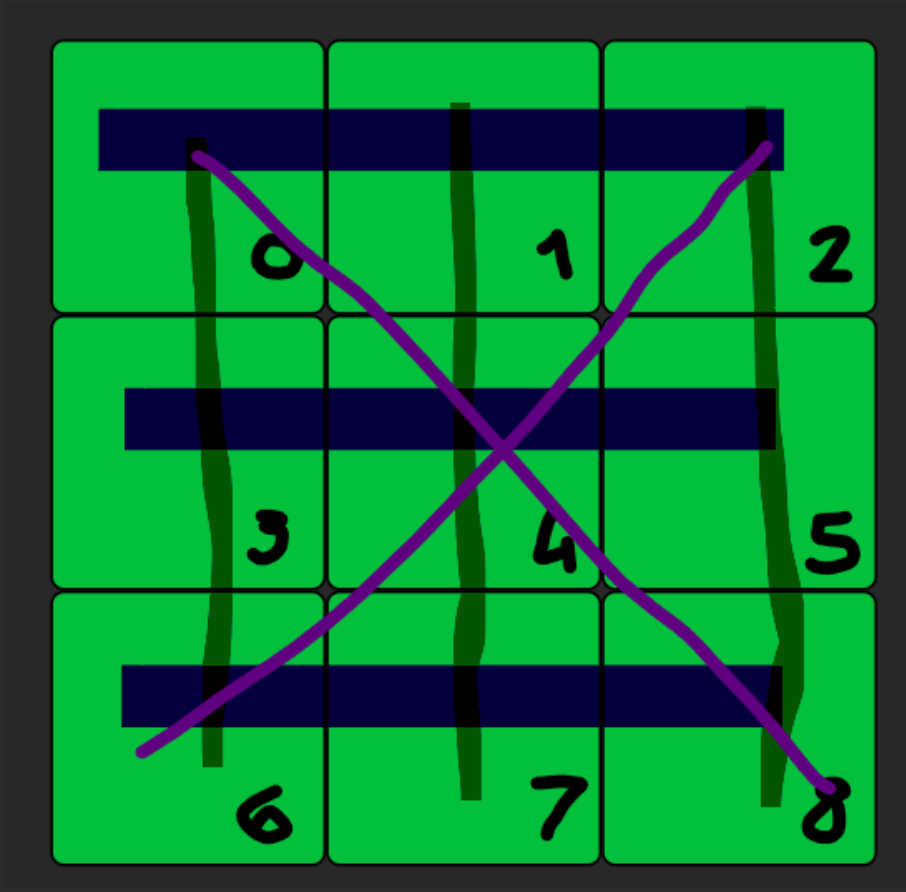
    )

}

Pentru calcularea caștigătorului am făcut funcția getWinner():

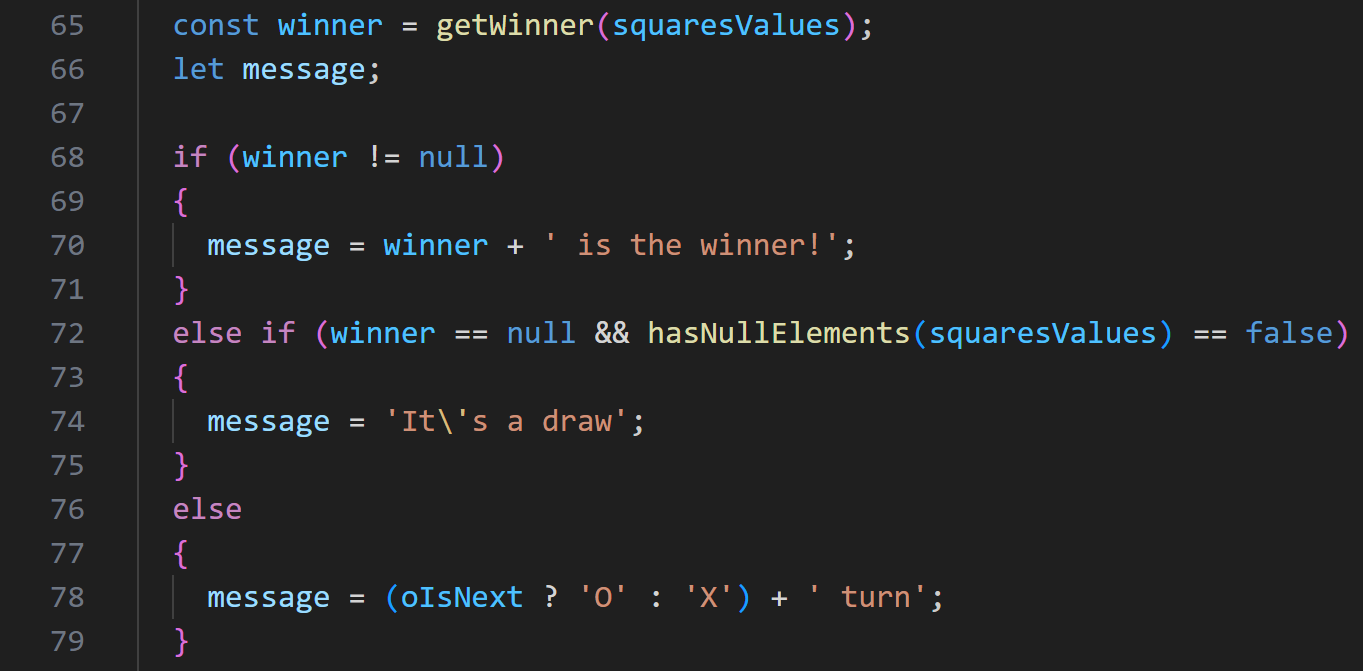


winningLines:

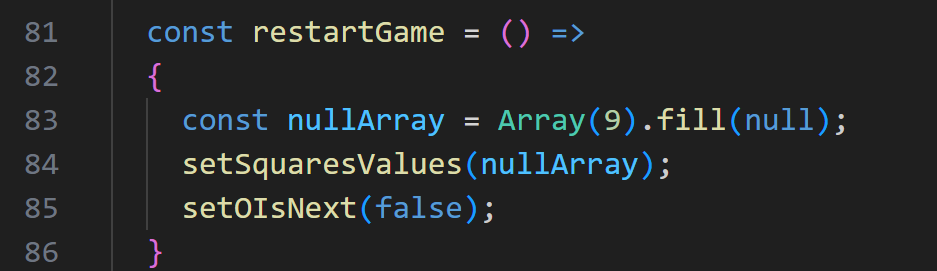


winningLines este o listă de liste care conține liniile caștigătoare. Pentru a verifica daca o linie este caștigătoare, parcurg un element al lui winningLines (adica o linie), verific dacă elementele liniei (patratele) sunt diferite de null și sunt egale, iar în caz afirmativ, returnez valoarea ce se afla pe linie, adica X sau O.

Setarea mesajului:



Restartul jocului:

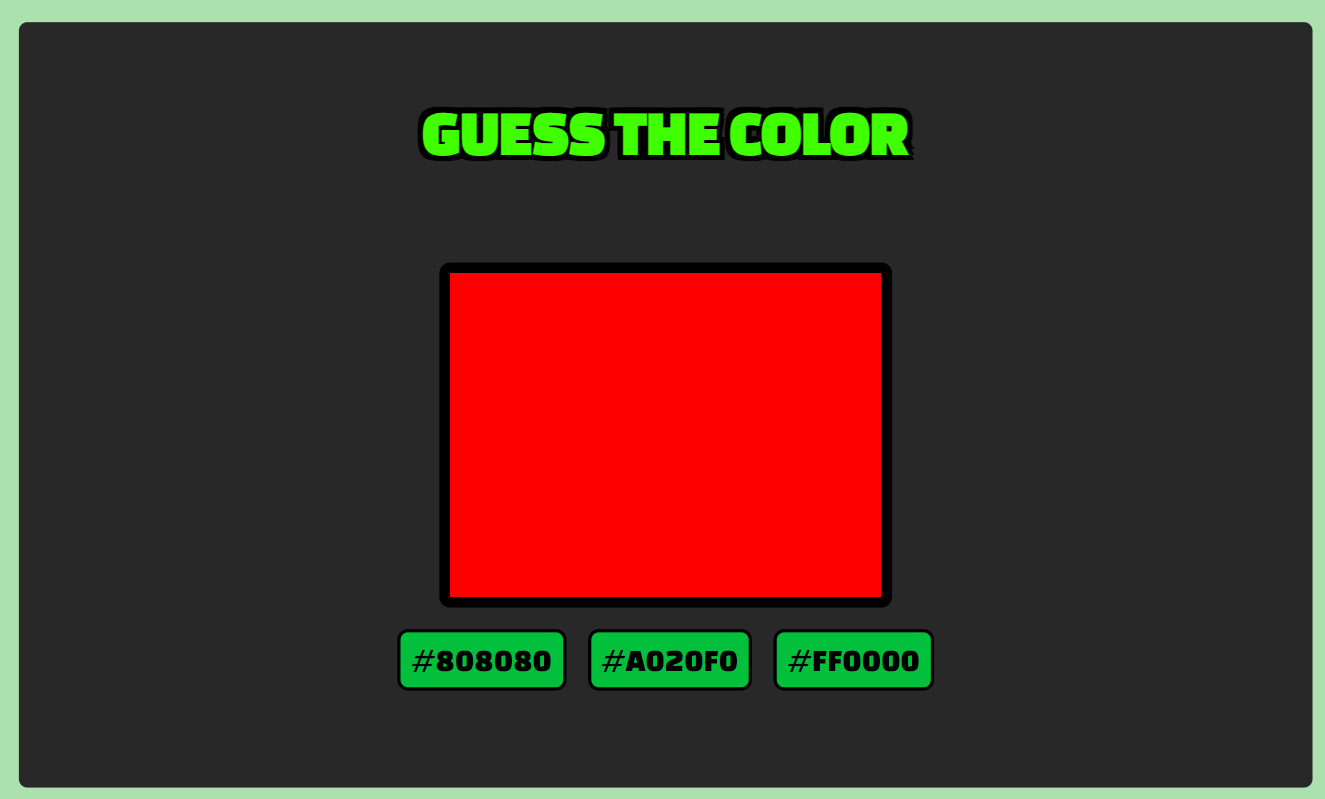


Stilizare:

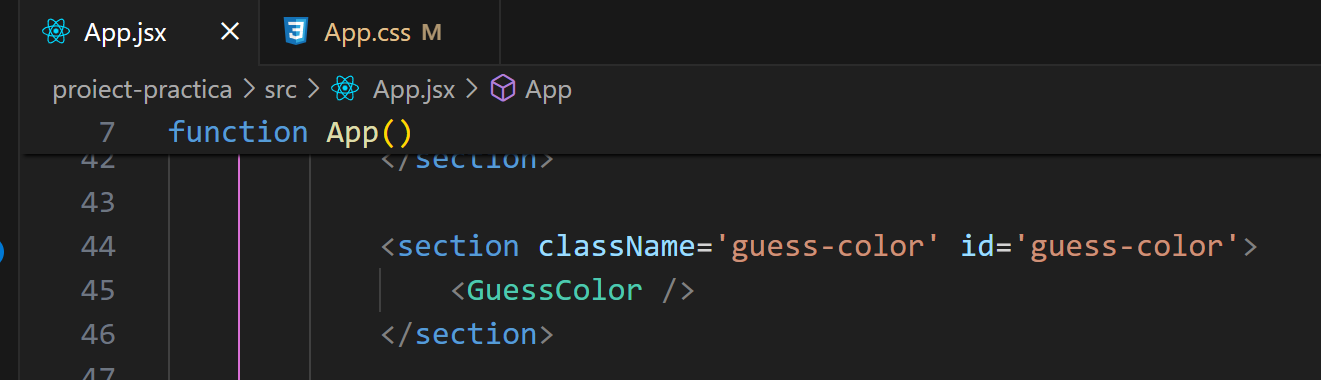
Pentru stilizare am aplicat flexbox astfel incat sa asez patratele si celelalte elemente frumos in pagina.

## **Sectiunea ‘Guess The Color’**

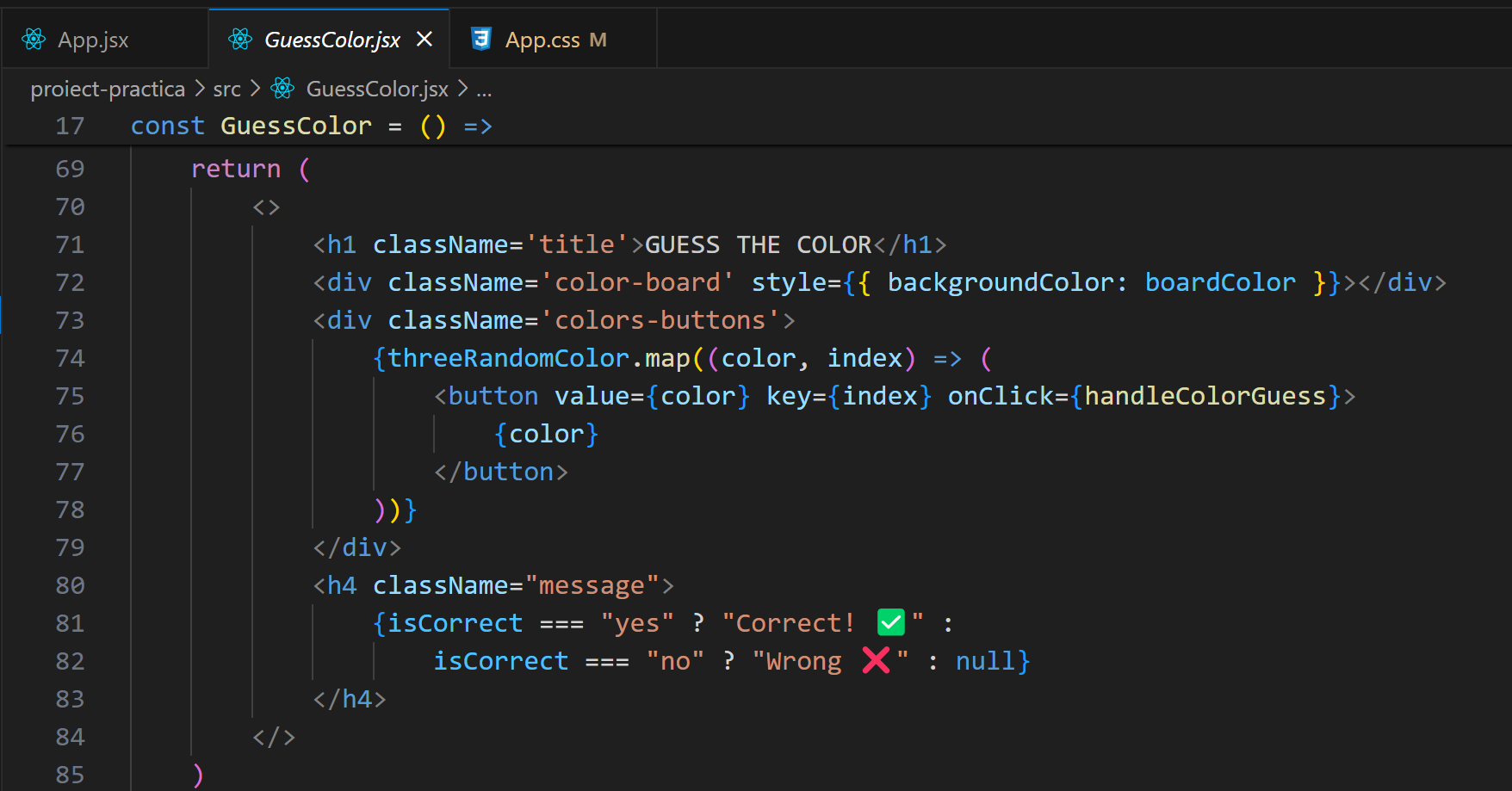
Regulile jocului sunt ca pe tablă o să apară o culoare la întamplare, iar utlizatorul trebuie să aleagă valoarea in hexadecimal corespunzatoare culorii de pe tabla. Dacă este corect, se afisează un mesaj și se continuă jocul (tabla și butoanele generează alte culori), dacă este greșit, se asteaptă până utilizatorul găsește culoarea corectă.



Figură 24 - Guess the color

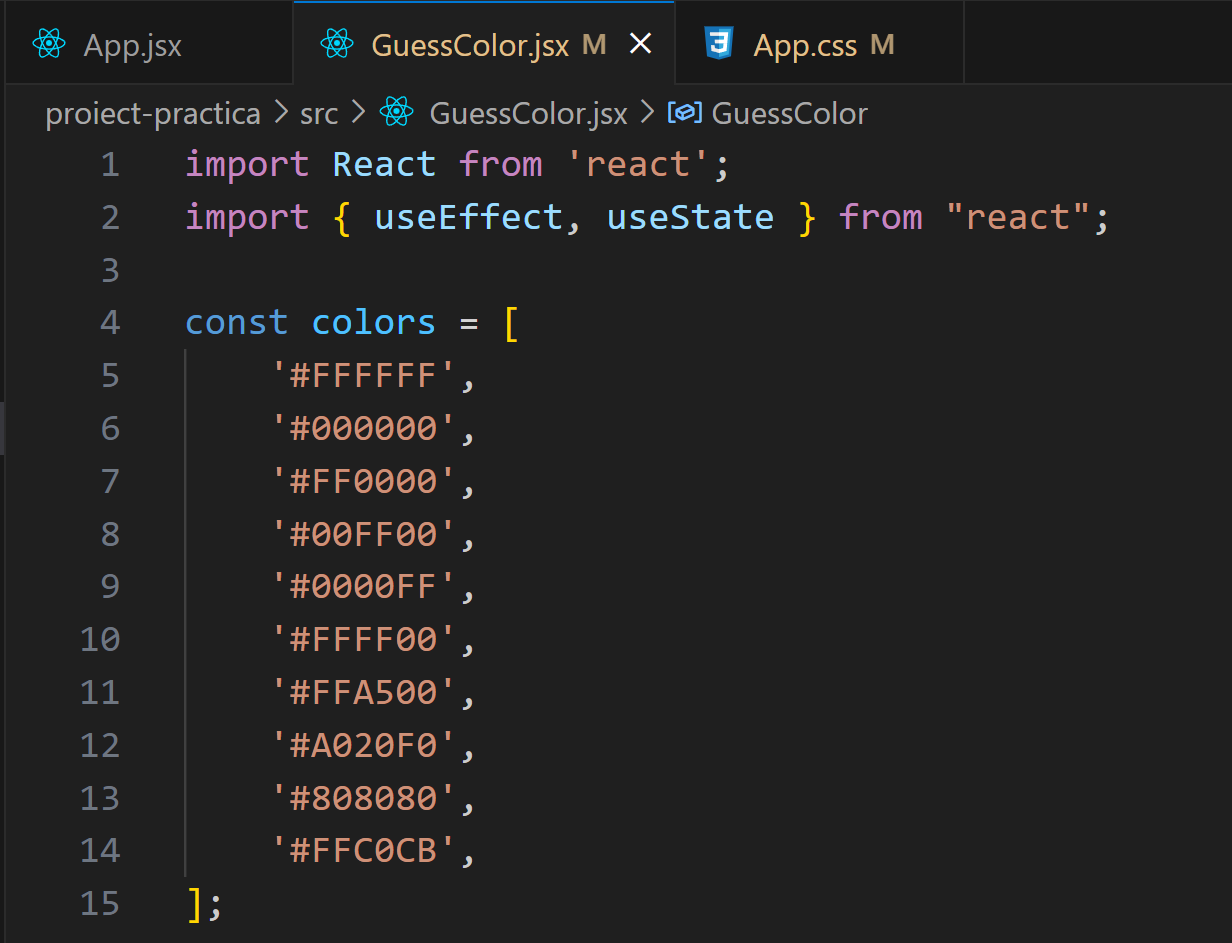


Figură 25 - Cod pentru 'Guess the color' din App.jsx

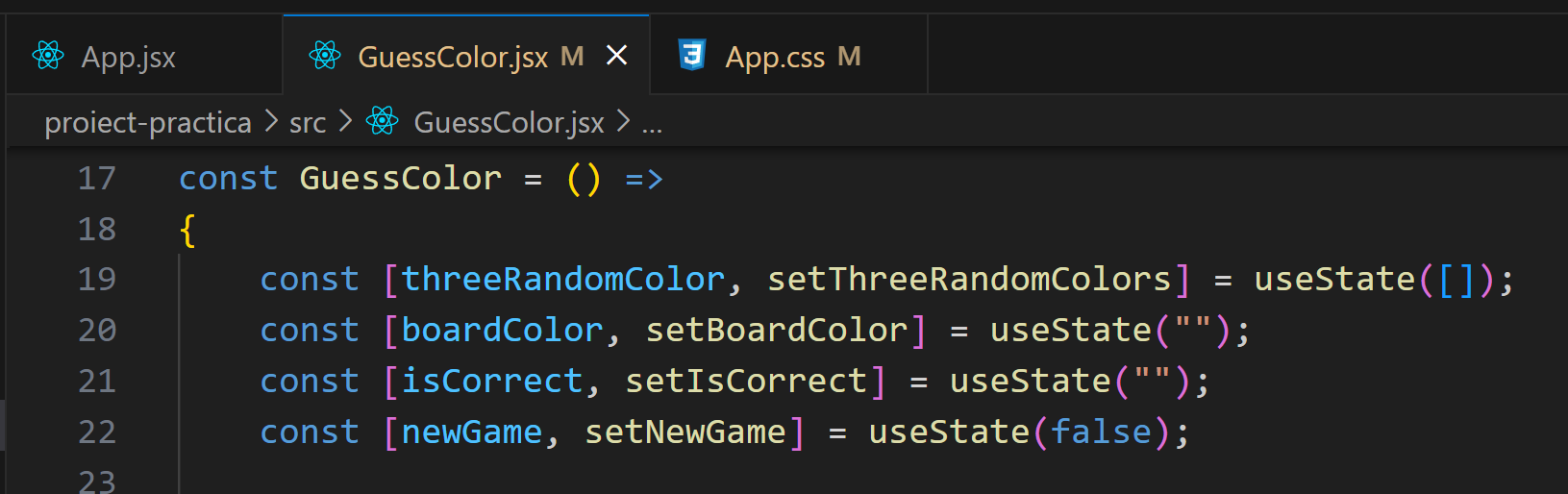


Figură 26 - cod pentru 'Guess the color' din GuessTheColor.jsx

In GuessColor.jsx se afla titlul ‘GUESS THE COLOR’, tabla in care este o culoarea, butoanele cu cele 3 culori si un mesaj care precizeaza daca butonul ales este corect sau nu.



Lista colors contine 10 culori alese de mine pentru a fi puse pe tabla si butoane.



threeRandomColor va contine 3 culori random alese din lista colors.

boardColor va contine o culoare random aleasă din lista threeRandomColor.

isCorrect va contine ‘yes’ sau ‘no’, in funcție de ce buton alege utilizatorului.

newGame este variabila care va declanșa începerea unui joc nou.

Simularea unui joc pentru explicarea codului:

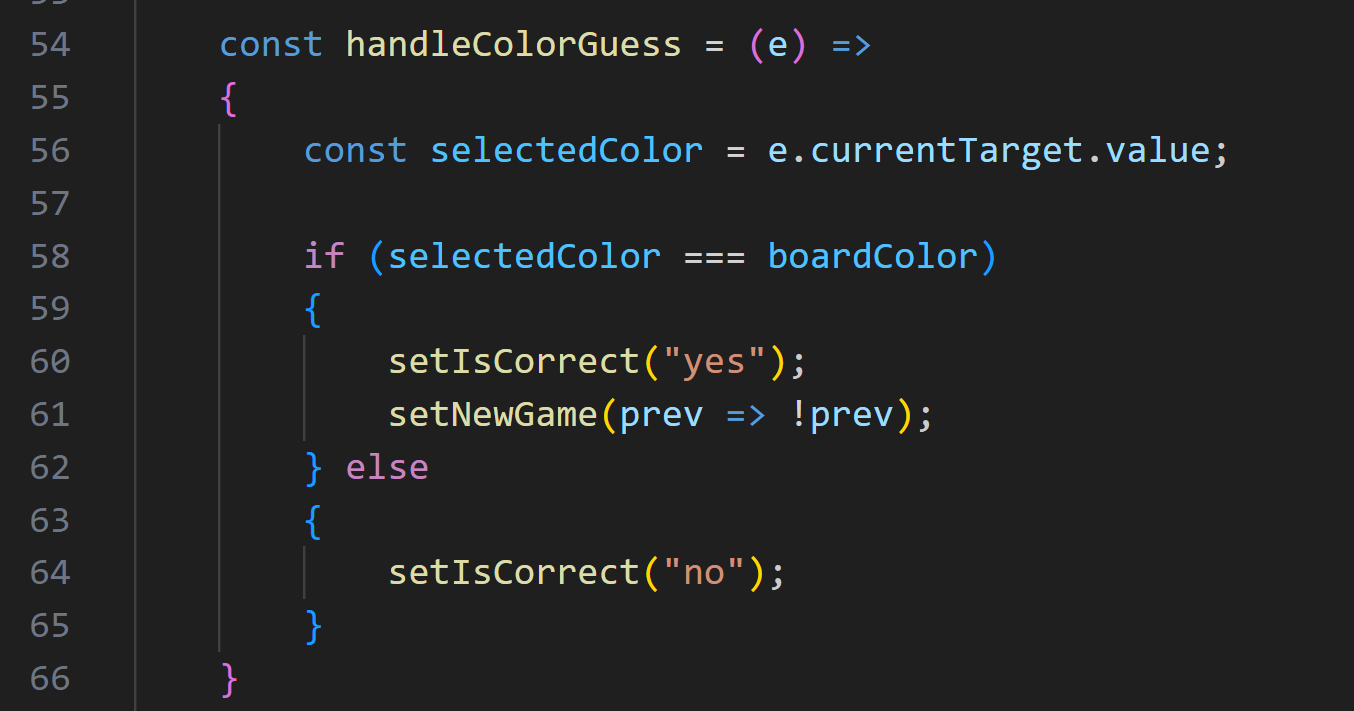


Valorea corectă este #000000, dar voi apasă butonul din mijloc pentru a înțelege ce se întamplă în cod.

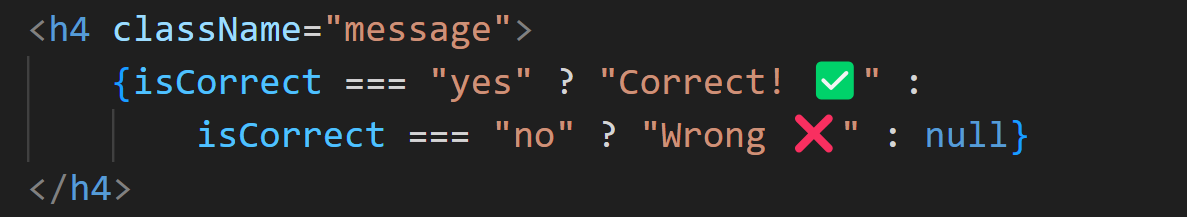


Atunci cand apas unul dintre cele 3 butoane se apeleaza functia handleColorGuess

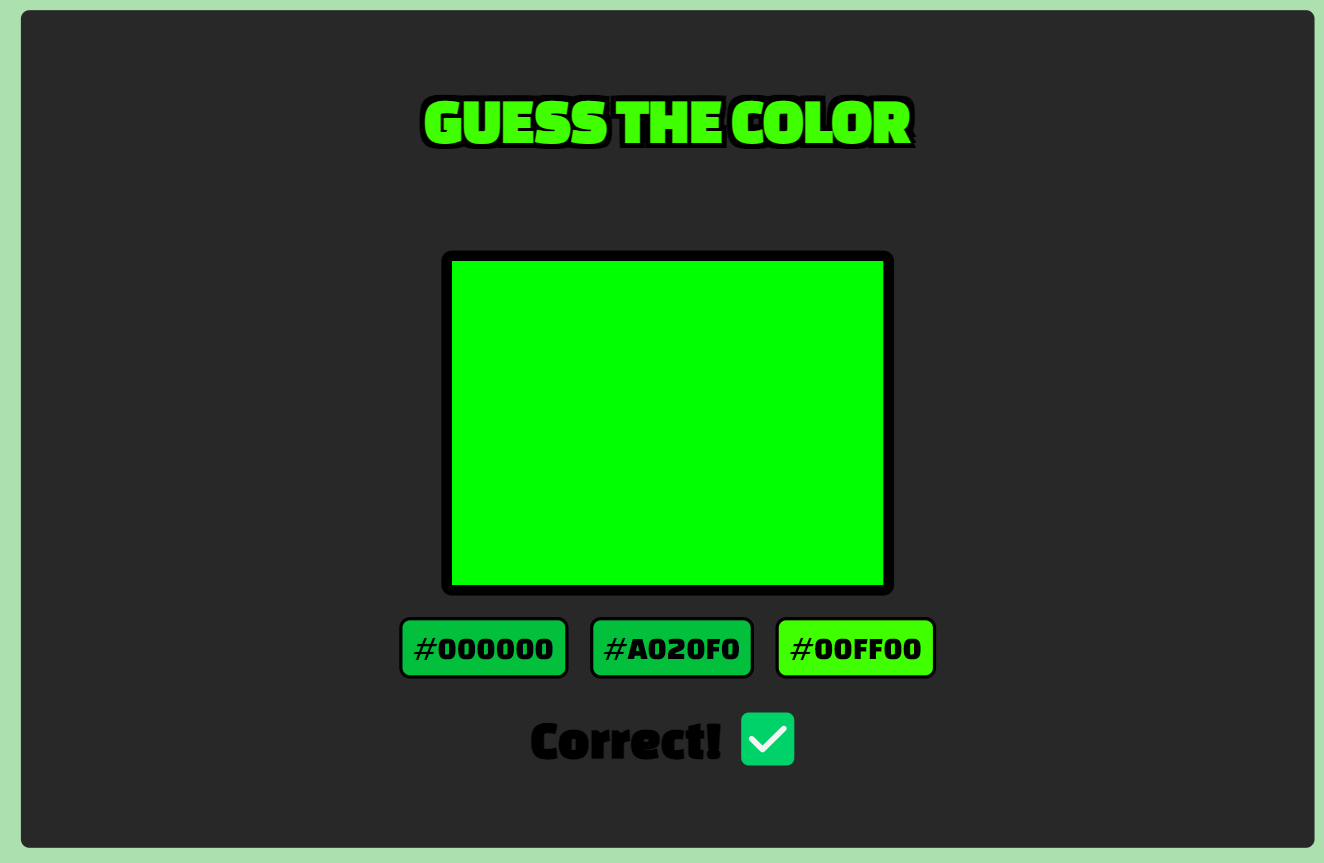




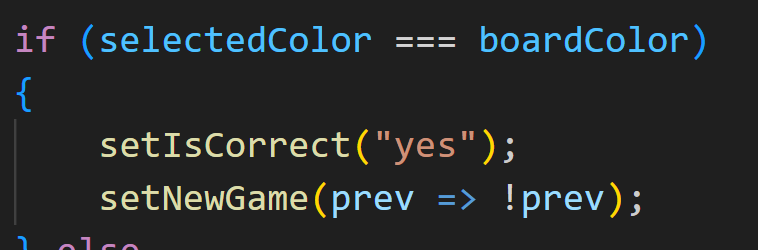
‘e’ (event) este un obiect care conține date despre eveniment (click-ul) cum ar fi și valoarea targetului care a declanșat evenimentul. În cazul nostru targetul este butonul din mijloc, și valoarea este ‘#FF0000’. Se verifică dacă ‘#FF00000’ (culoarea aleasă) = ‘#000000’ (culoarea corectă) => setIsCorrect(‘no’). Astfel, se actualizează doar mesajul:



Acum să presupunem că apăs butonul corect:



Pentru butonul din stanga se apeleaza functia handleColorGuess, setIsCorrect devine ‘yes’ si newGame isi schimba valoarea din false in true sau invers.



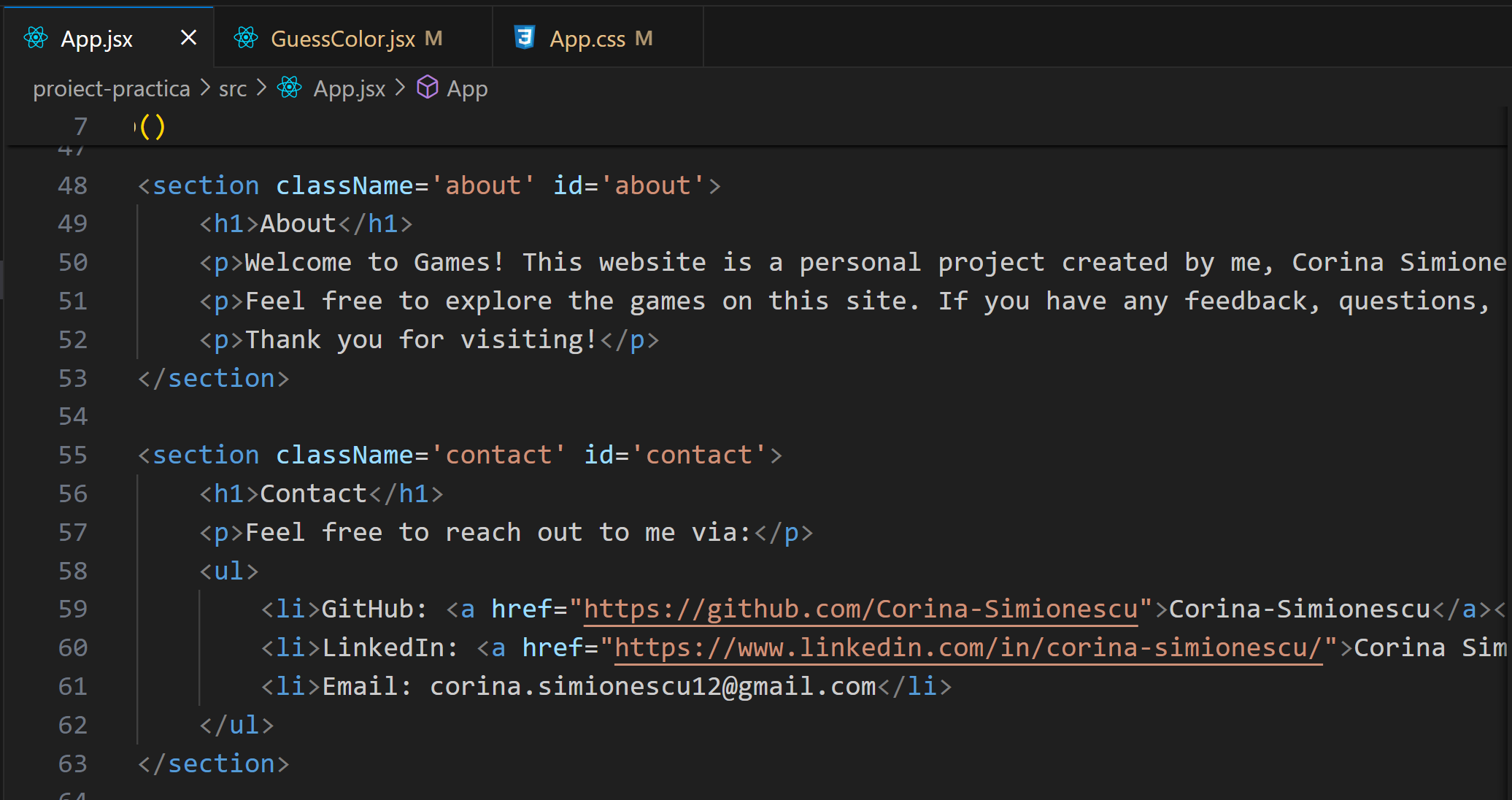
Atunci când newGame își schimbă valoarea, se declanșează useEffect-ul care incepe un nou joc. De asemenea, useEffect se execută automat și la montarea componentei GuessColor.jsx.



Stilizare:

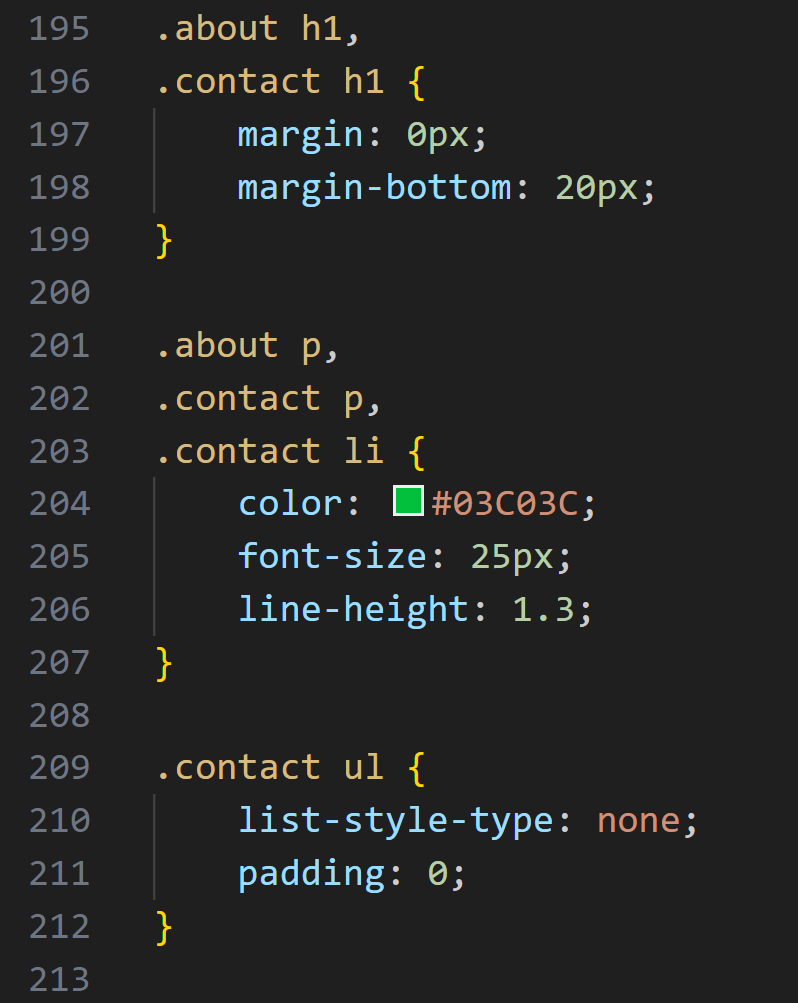
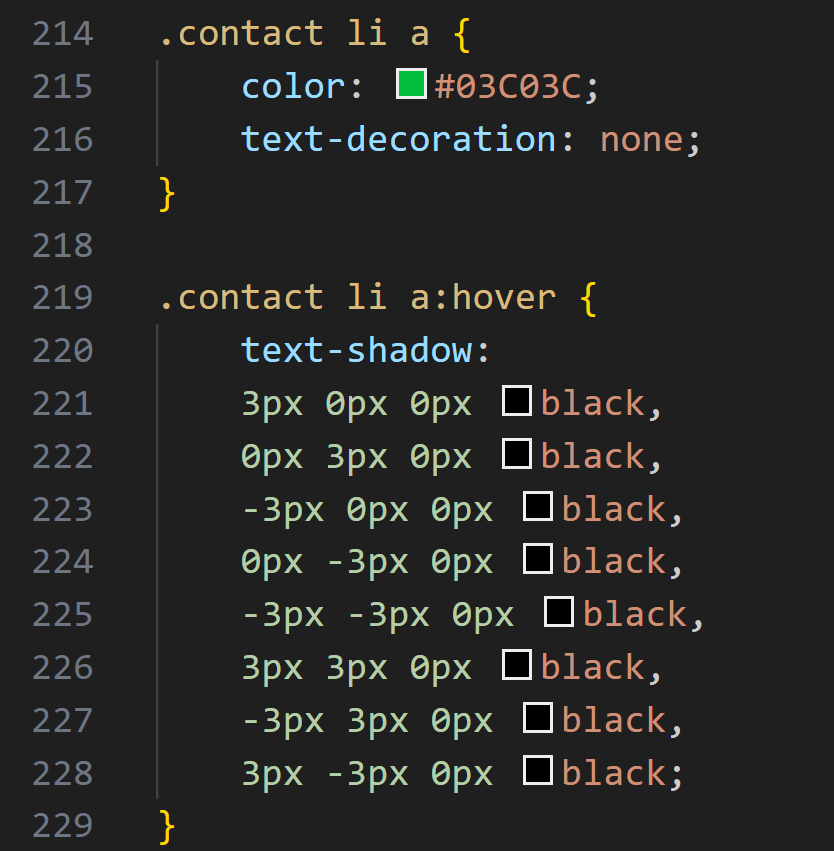
Am folosit flexbox pentru aranjarea elementelor în pagină și alte stiluri de CSS.

## **Secțiunile About si Contact**

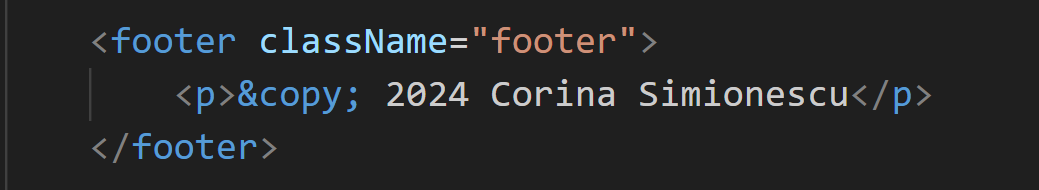


Figură 27 - About si Contact

Stilizare:

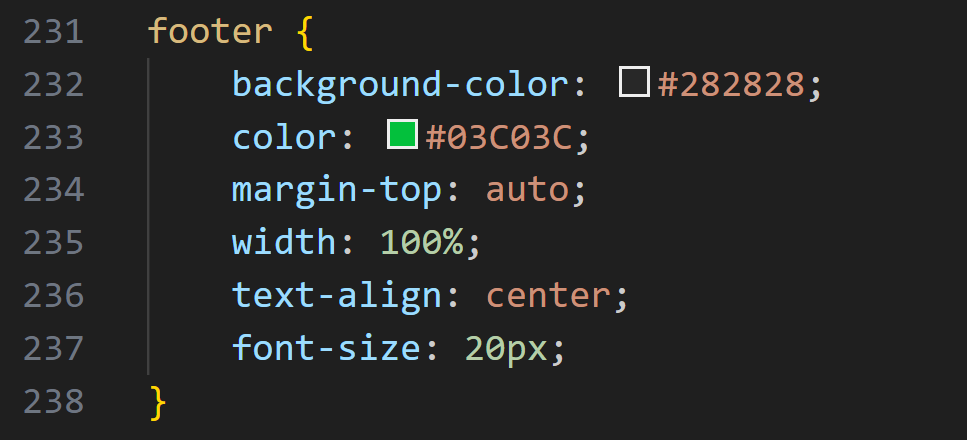
 

## **Footer-ul**



Figură 28 - Footer

Stilizare:



# **Concluzii**

Experiența mea de practică a fost una valoroasă și de neuitat. Am avut ocazia să întâlnesc oameni cu adevărat pasionați de programare care mi-au împărtășit cunoștințele lor și m-au inspirat. Am reușit să văd cum se lucrează într-o firmă adevarată, la un proiect adevărat și am avut șansa să aplic cunoștințele învățate într-un mediu practic. Această practică va rămâne cu siguranță un punct de referință în parcursul meu de dezvoltare profesională și mă motivează să continui să explorez și să mă perfecționez în domeniul IT.

# **Bibliografie**

* https://www.chesscoders.com/
* https://ro.wikipedia.org/wiki/HyperText\_Markup\_Language
* https://developer.mozilla.org/en-US/docs/Web/HTML
* https://en.wikipedia.org/wiki/CSS
* https://ro.wikipedia.org/wiki/JavaScript
* https://ocw.cs.pub.ro/courses/pw/laboratoare/04
* https://www.blogdeit.ro/ce-este-tailwind-css/
* https://en.wikipedia.org/wiki/Vite\_(software)
* https://www.newtech.ro/blog-ce-ese-node-js/
* https://en.wikipedia.org/wiki/Npm
* https://ro.wikipedia.org/wiki/MongoDB
* https://www.mongodb.com/docs/compass/current/
* https://en.wikipedia.org/wiki/Visual\_Studio\_Code