**Colegiul Național “Vasile Lucaciu”**

**Baia Mare**

*Bicicleta Potrivită*

**LUCRARE PENTRU ATESTAREA COMPETENŢELOR PROFESIONALE**

**Profesor Coordonator: Maidan Alin**

**Autor: Ignat Mihai**

**2020**

**CUPRINS**

**INTRODUCERE.............................................................................. 3**

**LIMBAJUL HTML.......................................................................... 4**

**LIMBAJUL CSS............................................................................... 6**

**LIMBAJUL JAVASCRIPT............................................................. 8**

**SOFTUL UTILIZAT........................................................................ 9**

**STRUCTURA APLICAȚIEI ......................................................... 11**

**COMENZILE APLICAȚIEI......................................................... 12**

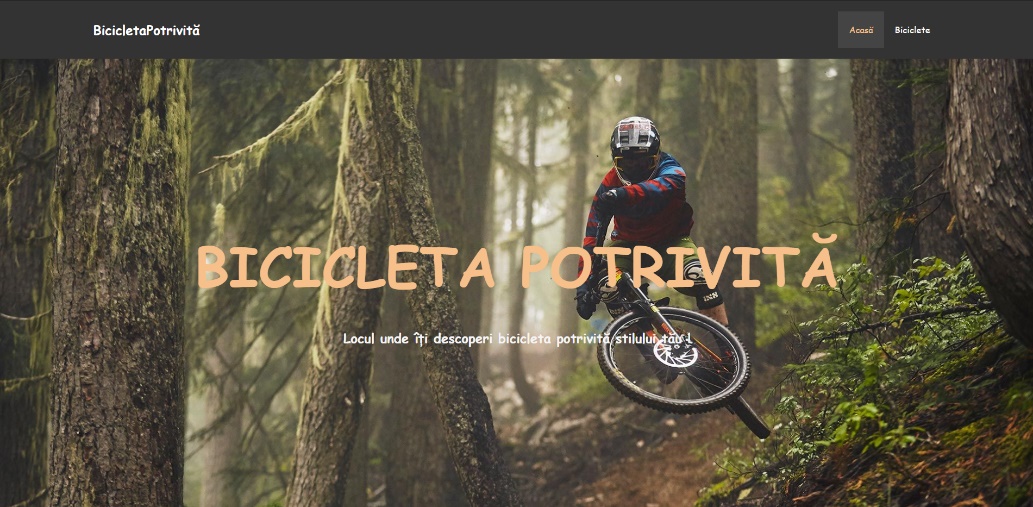
**DETALII TEHNICE...................................................................... 13**

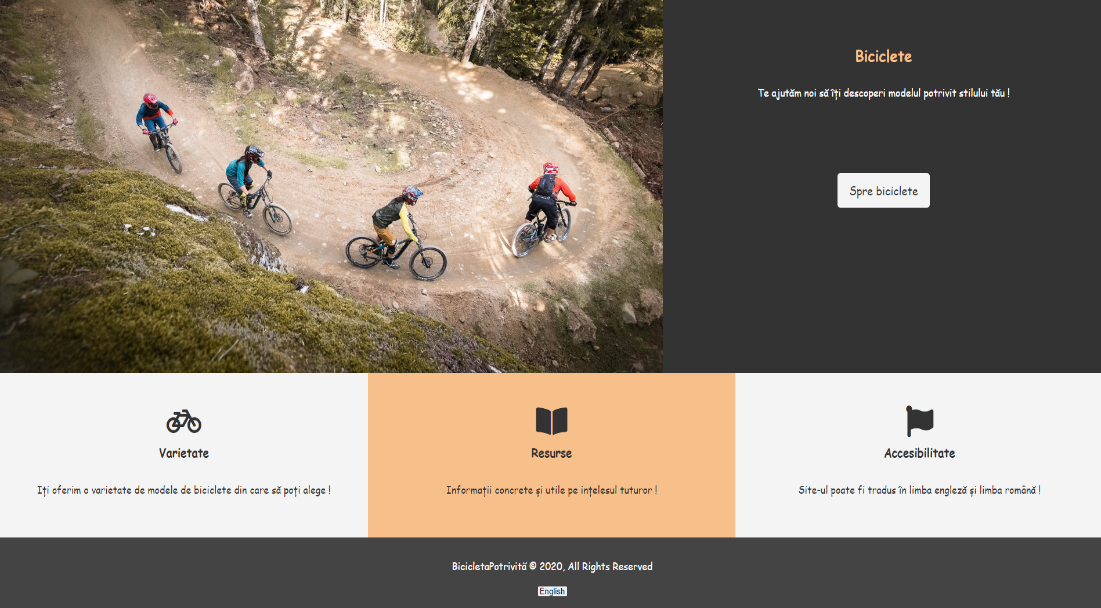
**RESURSE DE HARD SI SOFTWARE NECESARE…………. 17**

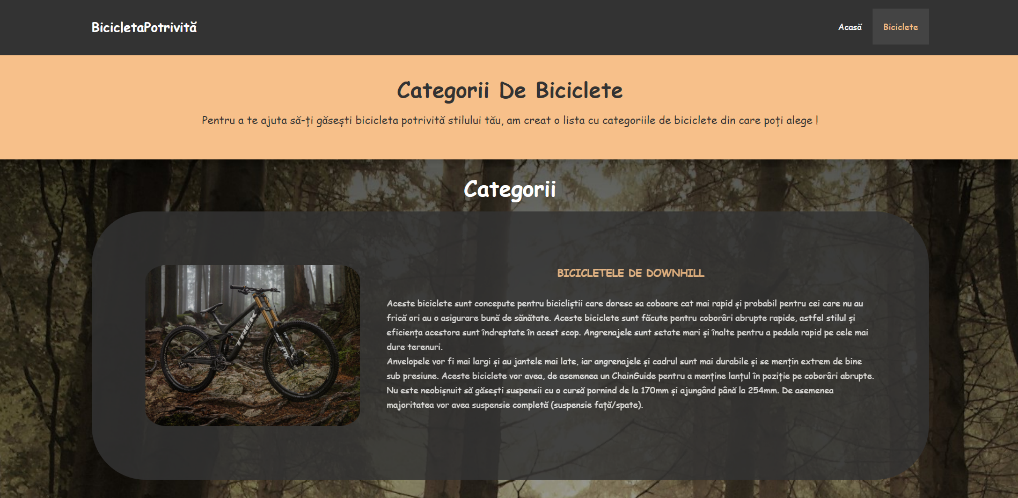
**BIBLIOGRAFIE…………………………………………………. 18**

**INTRODUCERE**

**BICICLETA POTRIVITĂ** este un produs soft care poate fi utilizat pentru a clasifica cu ușurință categoriile de biciclete. Motivul pentru care am ales această tema a fost pentru a fi folosit de către utilizatorii în căutare de informații concrete despre categoriile de biciclete putând astfel să creez un comfort datorită lizibilității site-ului și a modului lui ușor de operare. Aceste informații sunt însoțite de către imagini exemplificate. Site-ul dispune de traducere în engleză , astfel utilizatorul poate selecta limba dorită.







**LIMBAJUL HTML**

HTML (Hypertext Markup Language) este un limbaj creat în scopul de a descrie, în mod text, formatul paginilor Web; fisierele create în acest limbaj vor fi interpretate de navigatoare, care vor afisa paginile în forma doritã (cu texte formatate, liste, tabele, formule, imagini, hiperlegãturi, obiecte multimedia etc.). HTML a apãrut ca o aplicatie ISO standard (apartine standardului SGML - Standard Generalized Markup Language, specializat pentru hipertext si adaptat la Web).

Asa cum se poate deduce din numele limbajului, HTML descrie caracteristicile de format ale elementelor incluse prin procedee de marcare. Acestea pot fi asemãnate intuitiv cu marcajele folosite în tipografie pentru a indica scrierea unui text cu un anumit tip de caractere. Fiecare element va fi introdus între douã marcaje ("tags", în limba englezã) - de început si sfârsit - (uzual) de forma <marcaj> … </marcaj>. Caracterele speciale de delimitare a marcajelor "<", ">" permit deosebirea acestora de textul propriu-zis. De exemplu, pentru textele aldine (îngrosate), marcajul de început este <B> iar de sfârsit - </B>.

În informaticã, limbajele de marcare sunt foarte convenabile fiindcã comenzile lor pot fi interpretate simplu. LaTeX-ul, de exemplu, este tot un limbaj de marcare; prin interpretarea fisierelor .tex descrise în acest limbaj se va genera formatul dorit al documentelor pe diverse tipuri de sisteme de calcul (în cazul, LaTeX-ului, se obtine uzual format PostScript sau PDF). În schimb, procesoarele de documente uzuale nu au un limbaj de marcare standardizat, care sã ofere compatibilitate între diverse tipuri de calculatoare si sisteme de operare. Astfel, se poate spune cã avantajele aplicãrii limbajelor de marcare constau în portabilitate si flexibilitate: fisierele create cu ajutorul lor pot fi transferate pe orice tip de sistem, unde vor fi interpretate cu ajutorul unor programe specifice.

De fapt, procesoarele de texte uzuale folosesc adesea procedee de marcare pentru formatãri (de exemplu, formatãrile de tip caracter din Word); în acest caz însã, caracterele de control introduse sunt ascunse iar rezultatul editãrii este direct vizibil ("What You See Is What You Get"). În schimb, în limbajele de marcare - inclusiv HTML - marcajele sunt introduse în text, astfel încât acestea sunt exclusiv succesiuni de caractere (litere, cifre, caractere speciale) - fisiere de tip text.

Referitor la legãtura dintre procesoarele de documente uzuale si limbajul HTML, mai trebuie mentionat faptul cã ultimele versiuni ale editoarelor de documente oferã facilitãti de salvare în format HTML - de exemplu, Word, începând cu versiunea Microsoft Office '97. Mai mult, toate produsele incluse în aceastã gamã dedicatã biroticii (MS Office) oferã compatibilitate cu formatul HTML.

Procesele de standardizare si de includere a comenzilor de marcare în fisierele HTML permit navigatoarelor sã citeascã si sã formateze paginile Web, lucru foarte important în conditiile în care ele contin nu numai texte alb-negru, ci si culori, imagini, hiperlegãturi, diverse obiecte. Practic, marcajele HTML asigurã controlul asupra modului de afisare a obiectelor corespunzãtoare în cadrul programelor de vizualizare a documentelor HTML - navigatoarele.

Limbajul HTML a evoluat în versiuni succesive, odatã cu evolutia protocolului HTTP si a programelor de navigare. Astfel, HTML 1.0 era compatibil cu Mosaic, primul program de navigare, dar dupã aparitia unor navigatoare noi, a fost necesarã introducerea unui standard oficial Internet pentru construirea paginilor (HTML 2.0) si extinderea sa cu noi facilitãti: formule matematice, tabele, moduri avansate de descriere a organizãrii paginilor (începând cu HTML 3.0).

Standardizarea oficialã a limbajului HTML a fost realizatã de consortiul WWW si dezvoltatã de diversi producãtori de soft (unii dintre acestia urmãresc chiar promovarea navigatoarelor proprii prin introducerea unor particularitãti în formatele oficiale).

Paginile HTML se pot crea cu orice editor de texte de cãtre utilizatorii care cunosc limbajul HTML sau, mai simplu, se pot utiliza editoare speciale, în care obiectele se introduc interactiv iar codul HTML se genereazã automat. Având în vedere cã si în acest caz este utilã cunoasterea marcajelor generate pentru corectarea eventualelor erori (mai ales în cazul link-urilor), vom prezenta în continuare entitãtile care se pot introduce în paginile HTML si marcajele caracteristice acestora:

**Elementele limbajului HTML**

Toate obiectele HTML sunt introduse între marcaje care le definesc; majoritatea acestora sunt de forma <tip\_obiect> (la început) si </tip\_obiect> (la sfârsit). Tipul standard al obiectului poate fi specificat cu majuscule sau minuscule; totusi, se recomandã utilizarea majusculelor fiindcã astfel marcajele ies în evidentã.

Majoritatea navigatoarelor permit vizualizarea paginii curente în formatul sursã HTML (forma pe care o interpreteazã pentru afisarea paginii). La interpretare, programele de navigare ignorã spatiile si <Enter>-urile, aplicând formatarea specificatã.

Existã însã si marcaje cu parametri; acestea au forma <tip\_obiect param1=valoare1 param2=valoare2 …>. De exemplu, obiectele de tip imagine sunt introduse cu delimitatorul <IMG>, care are diversi parametri. Pentru definirea hiperlegãturilor se foloseste marcajul <A>, care are de asemenea parametri proprii.  
Codurile de marcare HTML pot fi clasificate în urmãtoarele categorii:

1. marcaje de bazã - cele care delimiteazã pagina / documentul HTML, titlul acesteia si corpul paginii;
2. marcaje pentru structurarea documentului - care permit introducerea de subtitluri, paragrafe, linii de delimitare;
3. marcaje pentru formatarea textului si crearea listelor;
4. marcaje pentru crearea hiperlegãturilor (hyperlinks);
5. marcaje pentru introducerea de obiecte - tabele, formule, imagini sau obiecte multimedia preluate din fisiere, formulare.

**LIMBAJUL CSS**

Css-ul este un limbaj de stilizare al elementelor html, al tagurilor html. Denumirea CSS provine din expresia Cascading Style Sheets. In Web Design-ul modern, pentru stilizarea paginilor web se foloseste numai CSS. Acest lucru inseamna ca de la culoarea literelor si a backgroundului pana si la pozitionarea elementelor de pe o pagina web, totul este stilizat prin CSS. Stilurile folosite pe o pagina pot fi incorporate in pagina respectiva sau pot fi chemate din fisiere externe, fisiere css.

CSS este un mijloc de stilizare a conținutului HTML. Dacă conținutul HTML este o casă proaspăt construită, CSS este mobilierul, decorul, facilitățile și orice altceva în plus, care îl transformă într-o casă. Creează conținut HTML pentru a-l face plăcut și atractiv din punct de vedere estetic. În CSS, aveți un selector, o declarație, o proprietate și o valoare.

Selectorul este numele etichetei elementului la care aplicați stilul: h1, p, img, a (hyperlink) sunt toate exemple de elemente pe care le puteți formato.

O declarație este parametrul de stil pe care l-ați setat. Dacă spuneți că etichetele <p> trebuie redate în culoarea albastru, aceasta este o declarație. Dacă spuneți etichete <table> pentru a utiliza 10px de spațiu celular, aceasta este și o declarație.

Proprietatea și valoarea merg mână în mână. Proprietatea este tipul de stil pe care îl declarați: familia fontului, dimensiunea fontului, culoarea, înălțimea liniei etc. Lista continuă și continuă. Valoarea este stilul specific, deci pentru familia de fonturi: Helvetica ;, valoarea este Helvetica. Pentru dimensiunea fontului: 20px; valoarea este de 20px.

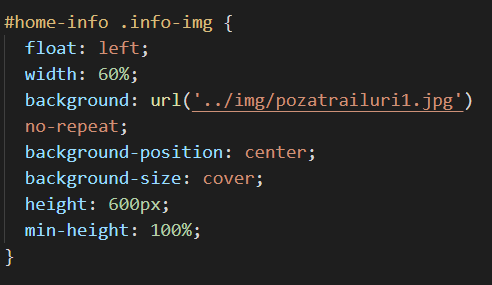
Practic, proprietatea se află în partea stângă a declarației: iar valoarea este în dreapta. Când aveți mai multe declarații, separați-le cu o;

CSS și HTML sunt două limbi diferite, dar ambele pot fi utilizate în același document. Există trei moduri de a încorpora CSS:

Extern, unde un fișier cu foi de stil (cel mai probabil cu o extensie .css) este legat aproape de începutul documentului HTML.

* Intern, în care o etichetă <style> este cuibărită în eticheta <head>.
* Inline, prin adăugarea atributului <style> sau <id> în interiorul unui element HTML individual. Inline aplică stiluri CSS doar unui anumit element.
* Extern și intern aplică stiluri CSS pentru întregul document.





**LIMBAJUL JAVASCRIPT**

**JavaScript** (**JS**) este un [limbaj de programare](https://ro.wikipedia.org/wiki/Limbaj_de_programare" \o "Limbaj de programare) [orientat obiect](https://ro.wikipedia.org/wiki/Programare_orientat%C4%83_pe_obiecte" \o "Programare orientată pe obiecte) bazat pe conceptul [prototipurilor](https://ro.wikipedia.org/w/index.php?title=Prototip_(POO)&action=edit&redlink=1" \o "Prototip (POO) — pagină inexistentă).[[5]](https://ro.wikipedia.org/wiki/JavaScript#cite_note-FOOTNOTEFlanaganFerguson20061-5) 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 [siturilor web](https://ro.wikipedia.org/wiki/Website" \o "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" \o "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" \o "Limbajul Self — pagină inexistentă) decât cu [Java](https://ro.wikipedia.org/wiki/Limbajul_Java).

Cea mai des întâlnită utilizare a JavaScript este în scriptarea [paginilor web](https://ro.wikipedia.org/wiki/Pagin%C4%83_web" \o "Pagină web). Programatorii web pot îngloba în paginile HTML script-uri pentru diverse activități cum ar fi verificarea datelor introduse de utilizatori sau crearea de meniuri și alte efecte animate.

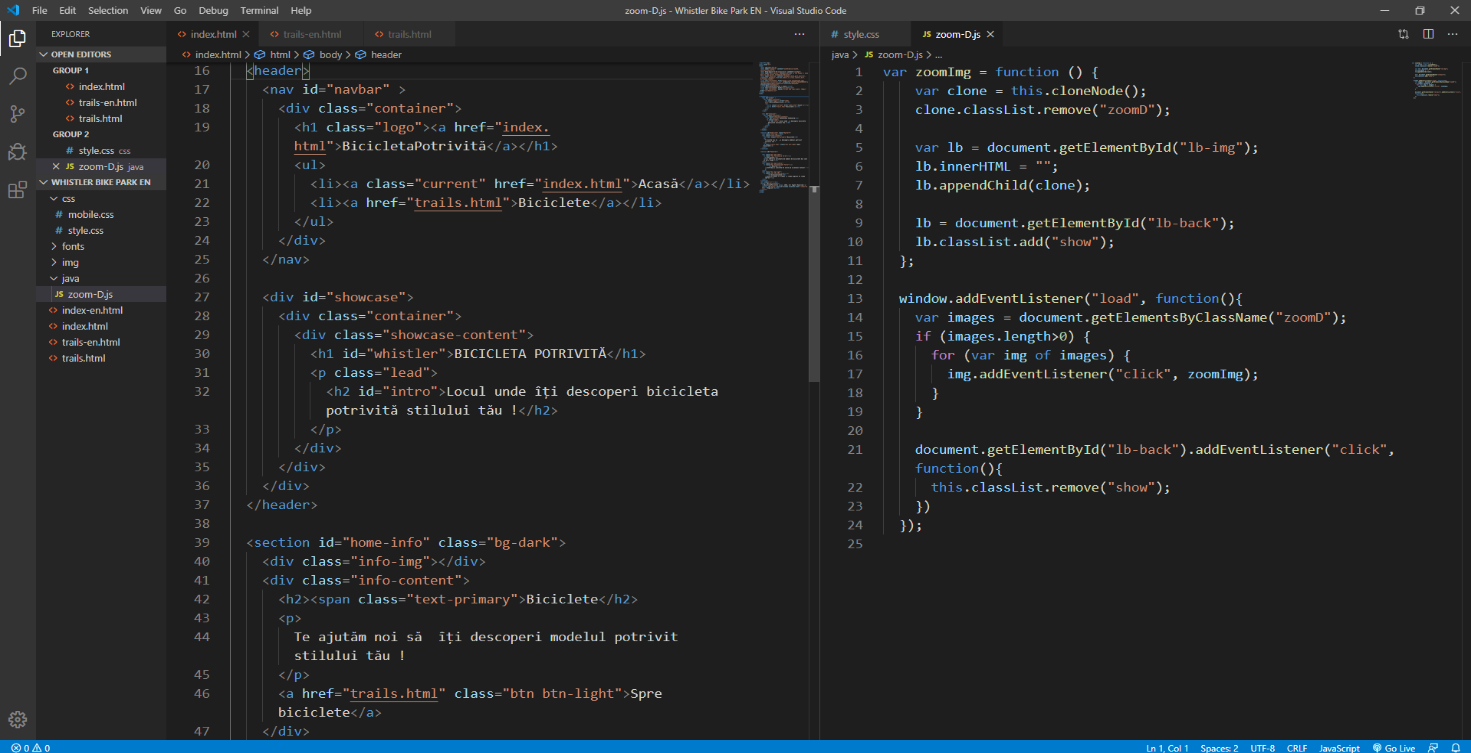
Browserele rețin în memorie o reprezentare a unei pagini web sub forma unui [arbore](https://ro.wikipedia.org/wiki/Arbore) de [obiecte](https://ro.wikipedia.org/wiki/Obiect_(informatic%C4%83)" \o "Obiect (informatică)) și pun la dispoziție aceste obiecte script-urilor JavaScript, care le pot citi și manipula. Arborele de obiecte poartă numele de [Document Object Model](https://ro.wikipedia.org/wiki/Document_Object_Model) sau [DOM](https://ro.wikipedia.org/wiki/DOM). Există un standard [W3C](https://ro.wikipedia.org/wiki/W3C) pentru DOM-ul pe care trebuie să îl pună la dispoziție un browser, ceea ce oferă premiza scrierii de script-uri [portabile](https://ro.wikipedia.org/w/index.php?title=Portabilitate&action=edit&redlink=1" \o "Portabilitate — pagină inexistentă), care să funcționeze pe toate browserele. În practică, însă, standardul W3C pentru DOM este incomplet implementat. Deși tendința browserelor este de a se alinia standardului W3C, unele din acestea încă prezintă incompatibilități majore, cum este cazul [Internet Explorer](https://ro.wikipedia.org/wiki/Internet_Explorer).

O tehnică de construire a paginilor web tot mai întâlnită în ultimul timp este [AJAX](https://ro.wikipedia.org/wiki/AJAX), abreviere de la „Asynchronous JavaScript and XML”. Această tehnică constă în executarea de cereri [HTTP](https://ro.wikipedia.org/wiki/HTTP) în fundal, fără a reîncărca toată pagina web, și actualizarea numai anumitor porțiuni ale paginii prin manipularea DOM-ului paginii. Tehnica AJAX permite construirea unor interfețe web cu timp de răspuns mic, întrucît operația (costisitoare ca timp) de încărcare a unei pagini HTML complete este în mare parte eliminată. JavaScript este cu adevărat un limbaj dynamic.



**SOFTUL UTILIZAT**

Programul utilizat pentru crearea paginii web utilizat de mine este Visual Studio Code. Visual Studio Code este un editor de coduri. La fel ca mulți alți editori de coduri, VS Code adoptă o interfață de utilizator comună care arată toate fișierele și folderele la care ai acces și un editor din dreapta, care arată conținutul fișierelor pe care le-ai deschis.



VS Code vine cu un aspect simplu și intuitiv, care maximizează spațiul oferit editorului, lăsând spațiu suficient pentru a răsfoi și a accesa contextul complet al folderului sau proiectului. Interfața de utilizare este împărțită în cinci domenii:

* Editor - zona principală pentru a vă edita fișierele. Puteți deschide oricât de mulți editori doriți cot la cot, vertical și orizontal.
* Bară laterală - conține diferite vizualizări precum Explorer pentru a vă ajuta în timp ce lucrați la proiectul dvs.
* Bara de stare - Informații despre proiectul deschis și despre fișierele pe care le editați.
* Bara de activități - amplasată pe partea stângă, vă permite să comutați între vizualizări și vă oferă indicatori suplimentari pentru context, cum ar fi numărul de modificări de ieșire atunci când Git este activat.
* Panouri - Puteți afișa diferite panouri sub regiunea editorului pentru informații de ieșire sau depanare, erori și avertismente sau un terminal integrat. Panoul poate fi, de asemenea, mutat la dreapta pentru mai mult spațiu vertical.

De fiecare dată când porniți VS Code, acesta se deschide în aceeași stare în care a fost când l-ați închis ultima dată. Dosarul, aspectul și fișierele deschise sunt păstrate. Fișierele deschise din fiecare editor sunt afișate cu anteturi cu file (filele) în partea de sus a regiunii editorului.

Puteți deschide oricât de mulți editori doriți cot la cot, vertical și orizontal. Dacă aveți deja un singur editor deschis, există mai multe modalități de deschidere a unui alt editor în partea celui existent: Apăsați Alt pe un fișier din Explorer. Ctrl + \ pentru a împărți editorul activ în două. Deschideți în lateral (Ctrl + Enter) din meniul contextual Explorer al unui fișier. Faceți clic pe butonul Split Editor din partea dreaptă sus a editorului. Trageți și plasați un fișier în orice parte a regiunii editorului.

Un Minimap (contur de cod) vă oferă o imagine de ansamblu la nivel înalt a codului sursă, care este util pentru navigarea rapidă și pentru înțelegerea codului. Minimizarea unui fișier este afișată în partea dreaptă a editorului. Puteți face clic sau trage zona umbrită pentru a trece rapid la diferite secțiuni ale fișierului dvs.

Editorul are o bară de navigare deasupra conținutului său numit Breadcrumbs. Afișează locația curentă și vă permite să navigați rapid între foldere, fișiere și simboluri.Explorer este utilizat pentru a naviga, deschide și gestiona toate fișierele și folderele din proiectul dvs. VS Code se bazează pe fișiere și folder - puteți începe imediat prin deschiderea unui fișier sau a unui folder din VS Code.

După deschiderea unui folder în VS Code, conținutul folderului este afișat în Explorer. De aici puteți face multe lucruri:

* Creați, ștergeți și redenumiți fișiere și foldere.
* Mutați fișierele și folderele cu drag and drop.
* Utilizați meniul contextual pentru a explora toate opțiunile.

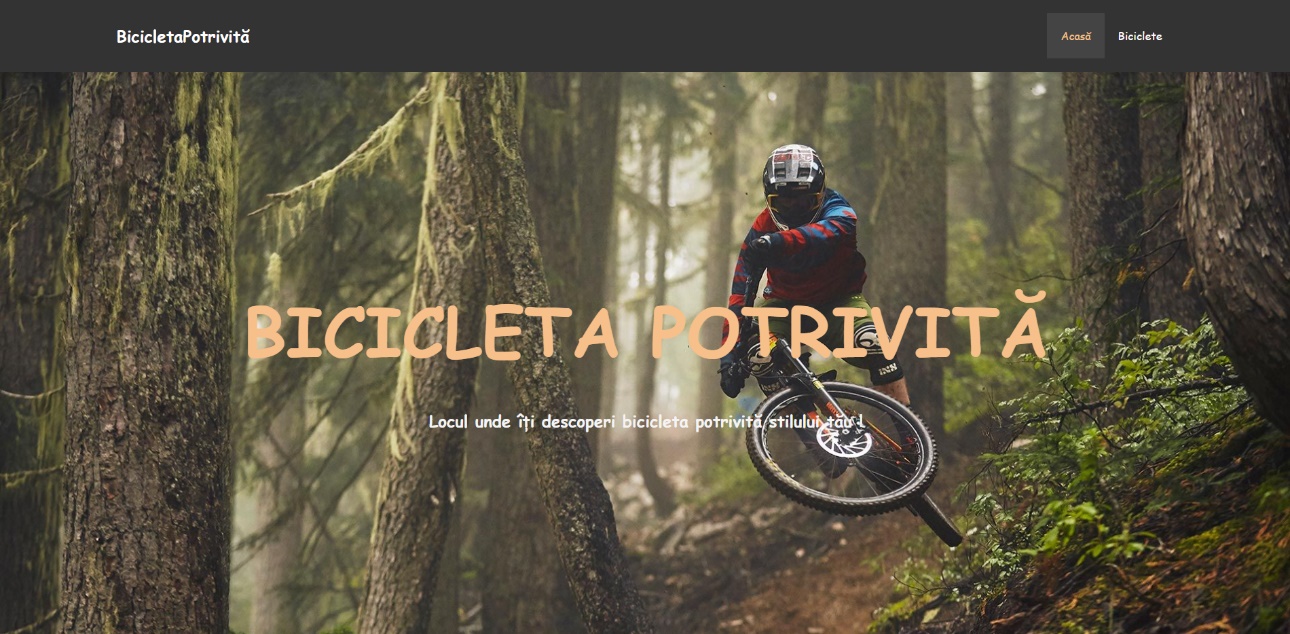
File Explorer este doar una dintre vizualizările disponibile în VS Code. Există, de asemenea, Vizualizări pentru: Căutare - oferă căutare globală și înlocuiește-l în folderul deschis. Control sursă - Codul VS include implicit controlul sursei Git. Debug - View Debug Code VS afișează variabile, stive de apeluri și puncte de întrerupere. Extensii - Instalați și gestionați extensiile dvs. în VS Code. Vizualizări personalizate - Vizualizări contribuite de extensii din vizualizările Explorator și Debug.

VS Code vă oferă multe opțiuni pentru a configura editorul. Din meniul Vizualizare, puteți ascunde sau comuta diverse părți ale interfeței cu utilizatorul, cum ar fi bara laterală, bara de stare și bara de activități.

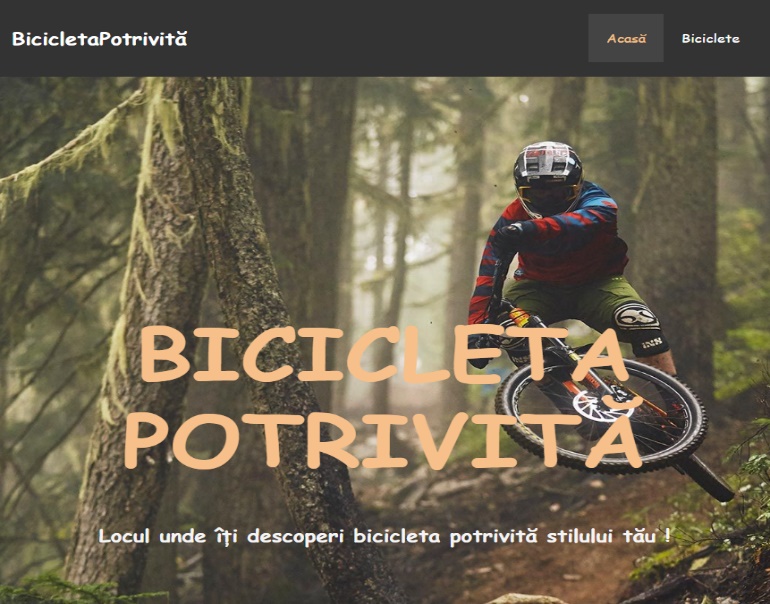
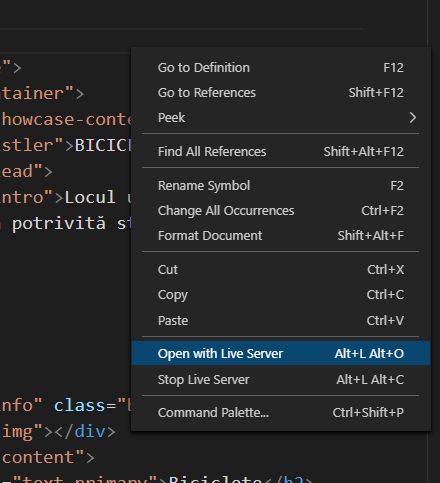
**STRUCTURA APLICAȚIEI**

Programul conține o forma principală alcătuită din mai multe butoane precum :

1. Pagina **ACASĂ/HOME** este pagina principală careconține:
   * Meniul de navigare in care este plasat in partea stângă logo-ul site-ului, iar în partea dreapta butoanele de navigare Acasă/Home și Biciclete/Bikes.
   * Conținutul paginii Acasă/Home în care se remarca imaginea de fundal alături de titlul site-ului ”Bicicleta Potrivită” și paragraful ”Locul unde îți descoperi bicicleta potrivită stilului tău” cu caracter de introducere în tema proiectului.
   * Un meniu care conține butonul ”Spre biciclete” care te redirecționează spre pagina Biciclete/Bikes.
   * O listă cu posibilitățile pe care site-ul ți le oferă.
   * Subsolul cu drepturile de autor și butonul de traducere a site-ului.
2. Pagina **BICICLETE/BIKES** este pagina secundară care conține:
   * Meniul de navigare in care este plasat in partea stângă logo-ul site-ului, iar în partea dreapta butoanele de navigare Acasă/Home și Biciclete/Bikes.
   * Un paragraf care conține un rezumat asupra conținutului ce va urma.
   * Categoriile de biciclete alături de poze și informații.
   * Subsolul cu drepturile de autor și butonul de traducere a site-ului.



**COMENZILE APLICAȚIEI**

Pentru a putea rula acest proiect vom avea nevoie de programul Visual Studio Code. Vom deschide Visual Studio Code. Odată aflați in codul sursă vom apăsa click dreapta + open with live server, astfel programul va deschide automat aplicația intr-o pagină web.

PAGINA ACASĂ/HOME:

1. Logo-ul aplicației – în partea stângă a meniului de navigație. Apăsând click stânga te va redirecționa pe pagina principală a site-ului Acasă/Home
2. Butoanele de navigare – în partea dreaptă a meniului de navigare. Apăsând click stânga te va redirecționa pe pagina respectivă.
3. Butonul ”Spre Biciclete” – în conținutul aplicației. Apăsând click stânga te va redirecționa pe pagina Biciclete/Bikes.
4. Butonul de traducere a paginii – în subsolul aplicației. Apăsând click stânga îți va traduce pagina in engleză/română.

PAGINA BICICLETE/BIKES

1. Logo-ul aplicației – în partea stângă a meniului de navigație. Apăsând click stânga te va redirecționa pe pagina principală a site-ului Acasă/Home
2. Butoanele de navigare – în partea dreaptă a meniului de navigare. Apăsând click stânga te va redirecționa pe pagina respectivă.
3. Imaginile – în conținutul paginii. Apăsând click stânga îți va redimensiona mărimea imaginii făcând mai vizibilă acea imagine. Apasând click inafara imaginii o va închide.
4. Butonul de traducere a paginii – în subsolul aplicației. Apăsând click stânga îți va traduce pagina in engleză/română.

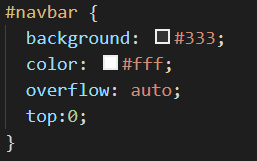
**DETALII TEHNICE**

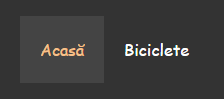
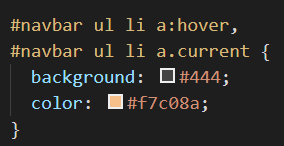
Proiectul a fost realizat cu ajutorul limbajelor HTML/CSS/JAVASCRIPT.

Pagina **ACASĂ/HOME:**

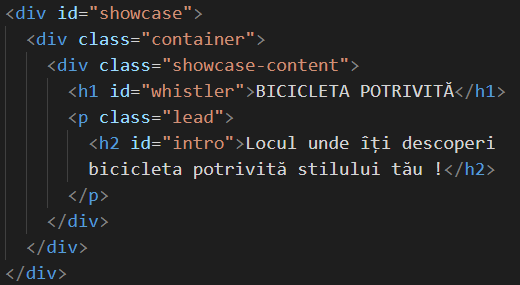
1. **Bara de navigare/Navbar**

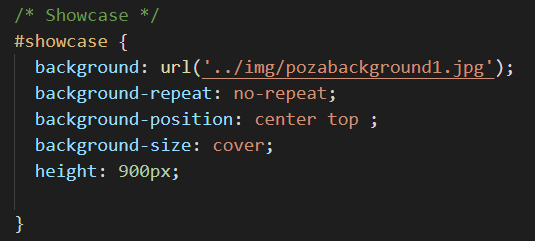
* Aceasta a fost creată cu ajutorul unei liste neordonate în ”<ul>” în care am plasat logo-ul site-ului urmat de două elemente listate ”<li>” reprezentând linkurile ”href=”…“ ” spre paginile site-uluiButoanelor de navigare le-am oferit un aspect “Hover”.



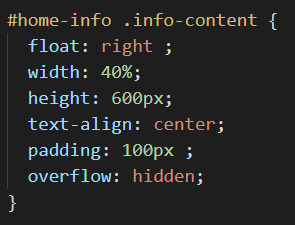
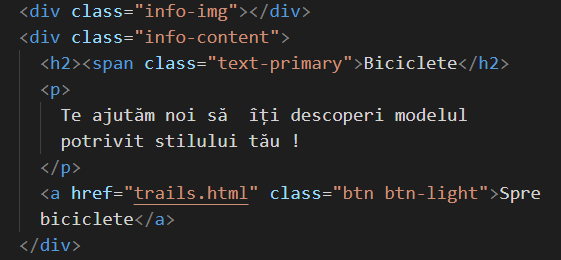


1. **Retrospectivă/Showcase**
   1. Reprezintă prima interfață cu utilizatorul . Aceasta conține o poza de fundal , un titlu “<h1>” reprezentând titlul site-ului , urmat de un paragraf “<p>” cu un mesaj introductiv în conținutul proiectului.

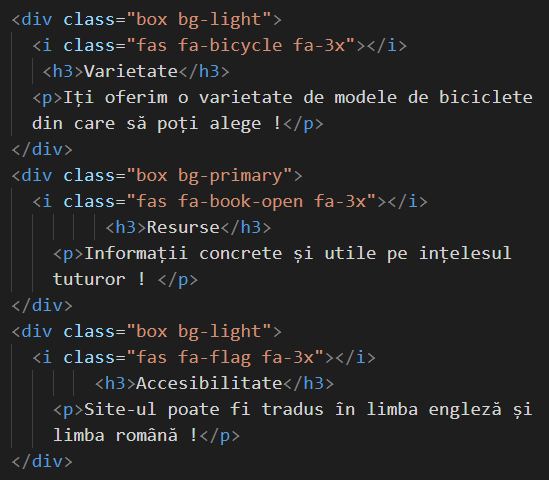
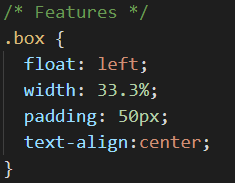




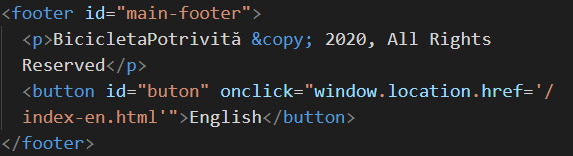
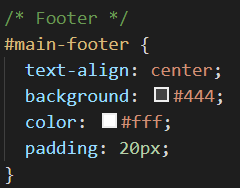
1. **Meniu Redirectionare/ Redirect Meniu**
   1. Conține o image in partea dreaptă și butonul “Spre biciclete” alături de un antet ”<h1>” si un paragraf ”<p>”.



1. **Meniul de informații/Information Meniu**
   1. Conține trei coloane fiecare coloană reprezentând o informație diferită create cu ajutorul unor elemente ”<i>”.
   2. M-am folosit de ajutorul site-ului ”TramIcon” pentru utilizarea unor iconițe, folosind un cod sursă oferit de aceștia ,folosind un ”stylesheet” diferit pentru aplicarea modificărilor.



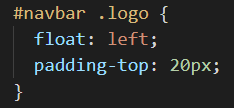
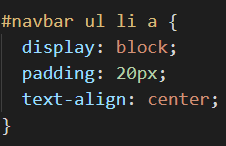
1. **Subsolul/Footer**
   1. Acesta conține drepturile de autor si butonul de traducere a paginii din limba română în limba engleză și viceversa, cu ajutorul unui buton “<button>” realizat cu ajutorul limbajului JavaScript.



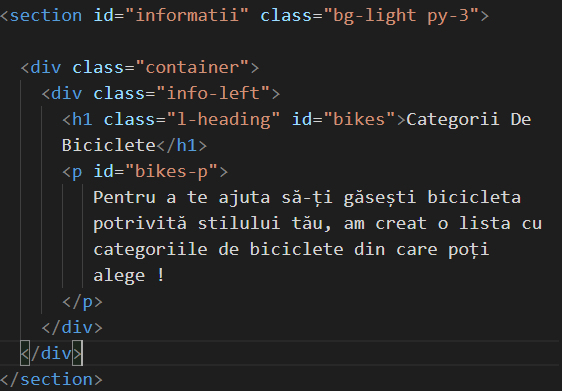
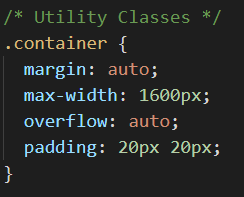
Pagina **BICICLETE/BIKES:**

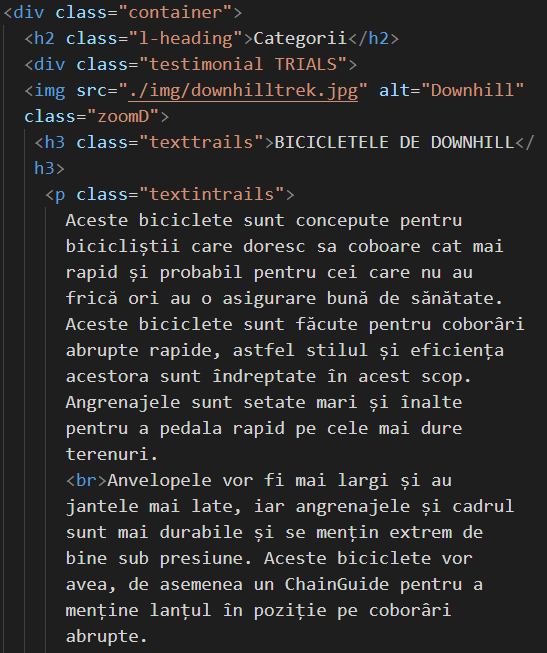
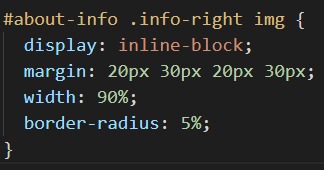
1. **Bara de navigare/Navbar**

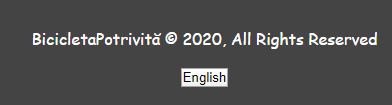
* Aceasta a fost creată cu ajutorul unei liste neordonate în ”<ul>” în care am plasat logo-ul site-ului urmat de două elemente listate ”<li>” reprezentând linkurile ”href=”…“ ” spre paginile site-uluiButoanelor de navigare le-am oferit un aspect “Hover”.



1. **Meniul De Informații/Information Meniu**
   1. Este o scurtă introducere în conținutul acestei pagini. A fost create cu ajutorul unui antet si al unui paragraf .



1. **Meniul De Categorii/Category meniu** 
   1. Acest meniu este alcatuit dintr-un antet care reprezintă titlul conținutului , urmat de cinci casete , fiecare casetă având o imagine în partea dreaptă , iar în partea stângă informațiile despre categoria de bicicletă reprezentată.
   2. Imaginile folosesc un cod sursă JavaScript astfel încât în momentul apasării click-stânga pe acestea, ele se vor mări astfel utilizatorului îi este mai ușor să distingă detaliile și să-și formeze o părere mai concretă asupra imaginii.
   3. Pentru crearea unui capăt de rând am folosit ”<br>”.
2. **Subsolul/Footer**
   1. Acesta conține drepturile de autor si butonul de traducere a paginii din limba română în limba engleză și viceversa, cu ajutorul unui buton “<button>” realizat cu ajutorul limbajului JavaScript.



**RESURSE DE HARD SI SOFTWARE NECESARE**

**Resurse minime :**

1) Hardware :

* Procesor (CPU) : 500 MHz +
* Memorie (RAM): 64 MB
* Spaţiu liber pe hard-disk : 30 MB
* Monitor :1024\*768
* Un browser oarecare: Internet explore ,Mozilla, Opera

2) Software :

* Sistem de operare : Windows 98/XP

**Resurse recomandate:**

1) Hardware :

* Procesor (CPU) : 1000 MHz +
* Memorie (RAM): 128 MB
* Spaţiu liber pe hard-disk : 30 MB
* Monitor : VGA 32-bit color (1024x768)
* Un browser oarecare: Internet explorer ,Mozilla

2) Software :

* Sistem de operare : Windows XP

**BIBLIOGRAFIE**

* <https://code.visualstudio.com/docs/getstarted/userinterface>
* <https://ro.wikipedia.org/wiki/HyperText_Markup_Language>
* <https://ro.wikipedia.org/wiki/JavaScript>
* <https://skillcrush.com/blog/css/>