

Liceul Național de Informatică Arad

LUCRARE DE ATESTAT
DISCIPLINA INFORMATICĂ

Candidat,

Clop Paul Eduard

Coordonator,

Prof. Pantea Bogdan

2023

Introducere

Lucrarea de față are ca și temă „Japonia”, mai precis prezentarea unor generalități despre modul de desfășurare a țării pe plan intern. Din meniul principal se pot selecta categoriile, prin care s-au grupat informațiile :

- Logo-ul site-ului situat în centru, prin care se poate accesa pagina Home
- Poze
- Economie
- Cultură
- Istorie
- Sport
- Bibliografie



Cuprins:

- 1. Motivul alegerii temei (pag. 4)**
- 2. Resurse de hard si soft (pag. 4)**
- 3. Programul utilizat (pag. 5-6)**
- 4. Algoritmul urmărit în realizarea proiectului
(pag. 7-18)**
- 5. Concluzie (pag. 19)**
- 6. Bibliografie (pag. 19)**

Motivul alegerii temei

Eu am ales ca și temă pentru atestat „Japonia”, pentru că sunt pasionat de această țară, prin cultura sa inedită și prin modul de construire a infrastructurii, iar prin acest site vreau să transmit mai departe informații de cultură generală despre această națiune, Japonia fiind o țară care a crescut considerabil în popularitate, în decursul ultimilor ani.

Resurse de hard si soft

Proiectul fiind realizat în limbajul de marcare HTML, adițional CSS, acesta nu are nevoie de un nivel ridicat de configurație pentru a funcționa în parametri normali.

Resurse hard

	Cerințe Windows	Cerințe Mac	Cerințe Linux
Sistem de operare	Windows 7 sau mai nou	macOS High Sierra 10.13 sau mai nou	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+
Procesor	Intel Pentium 4 sau mai nou	Intel, M1 sau mai nou	Intel Pentium 4 sau mai nou
Memorie	minim 2 GB, recomandat 4 GB		
Rezoluție ecran	1280x1024 px sau mai mare, recomandat 1920x1080 px		
Conexiune la internet	Recomandată, pentru a avea la dispoziție toate opțiunile site-ului		

Resurse soft

Browser	Chrome	Firefox	Edge	Safari
Versiune (minimă)	v112	v111	v112	v14.1

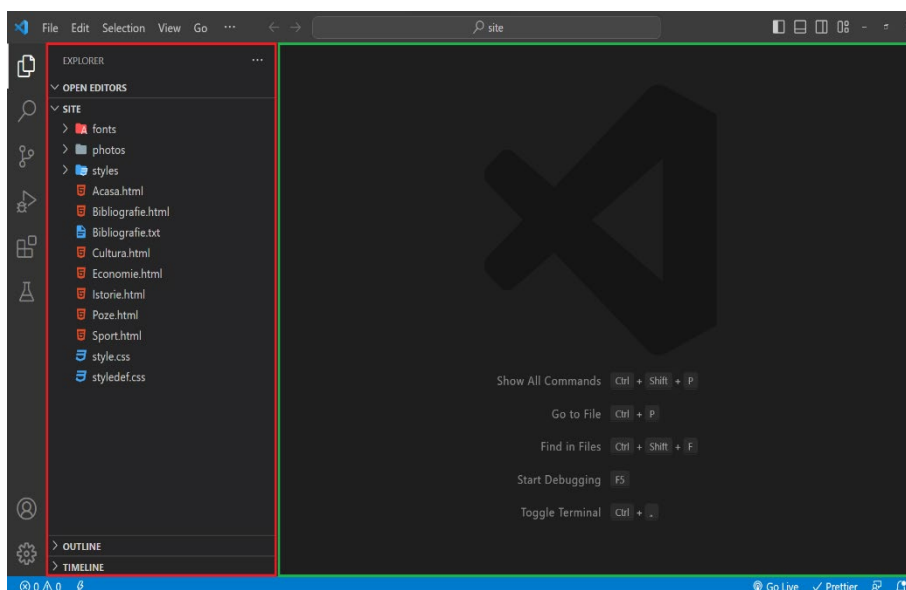
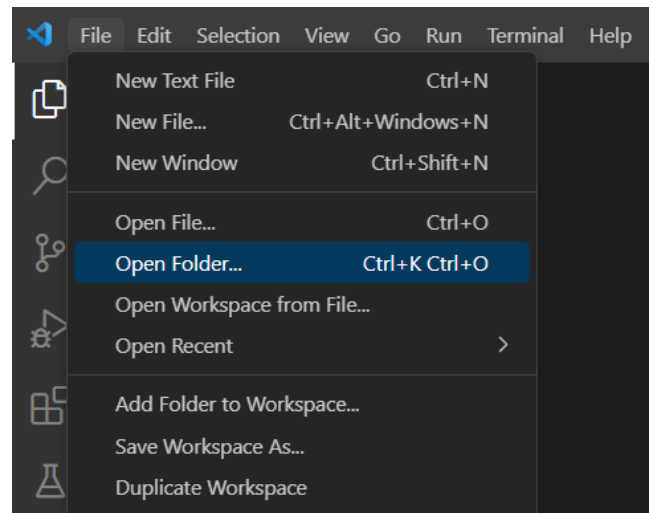
Programul utilizat

Pentru realizarea site-ului web am folosit aplicația **Visual Studio Code**.

Visual Studio Code, denumit și **VS Code**, este un **editor de cod sursă** realizat de Microsoft cu Electron Framework, pentru Windows, Linux și macOS. Caracteristicile includ suport pentru **depanare**, **evidențierea sintaxei**, **completarea inteligentă a codului**, **fragmente**, **refactorizarea codului** și **Git** încorporat. Utilizatorii pot modifica tema, comenzile rapide de la tastatură, preferințele și pot instala extensii care adaugă funcționalități suplimentare.

Mediul de lucru

- Pentru deschiderea fișierului, unde se află site-ul web, se accesează din bara de meniuri: „File – Open Folder...” .

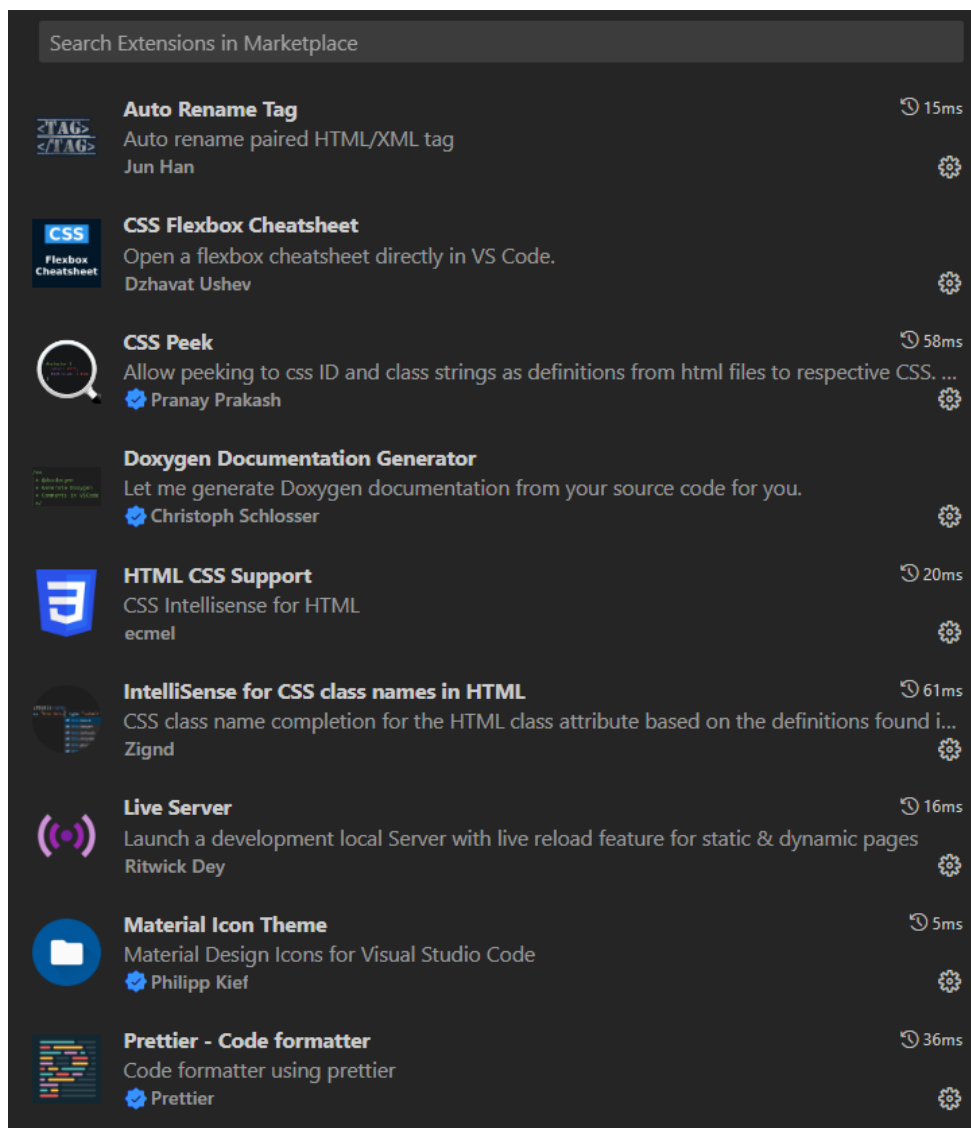


- în această căsuță se afișează conținutul fișierului și se pot accesa;

- în această căsuță se afișează conținutul documentului deschis

Extensii folosite

Extensiile VS Code permit adăugarea de limbi, programe de depanare și instrumente la instalare pentru a sprijini fluxul de lucru. Diversitatea și complexitatea extensiilor le permit autorilor să se conecteze direct la interfața de utilizare VS Code și să contribuie cu funcționalitate prin aceleași API-uri utilizate de VS Code.



- Una dintre cele mai folosite extensii este „**Live Server**”, care permite modificarea în timp real al site-ului după code.

Algoritmul urmărit în realizarea proiectului

HyperText Markup Language (HTML) este un **limbaj de marcare** utilizat pentru crearea **paginilor web** ce pot fi afișate într-un **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 de tip front-end, HTML este utilizat împreună cu **CSS**. HTML a fost dezvoltat inițial de **Tim Berners-Lee** la CERN în **1989**. Acesta a fost văzut ca o posibilitate pentru fizicienii care utilizează computer-e diferite și schimbă între ei informații, utilizând Internetul. Specificațiile HTML sunt dictate de **World Wide Web Consortium (W3C)**, care încurajează utilizarea CSS în locul HTML-ului explicit pentru prezentare.

Documentele HTML sunt documente în format **ASCII**, prin urmare pot fi create cu orice **editor de text**. Au fost însă dezvoltate editoare specializate, care cresc nivelul de productivitate. Au fost, de asemenea, dezvoltate convertoare care permit formatarea HTML a documentelor generate cu alte editoare.

Structura documentului HTML

Orice document HTML începe cu notația **<html>** și se termină cu notația **</html>**. Acestea se numesc în literatură de specialitate „**TAG-uri**”. Prin convenție, toate informațiile HTML încep cu o paranteză unghiulară deschisă "**<**" și se termină cu o paranteză unghiulară închisă "**>**".

Tag-urile între aceste paranteze transmit comenzi către browser pentru a afișa pagina într-un anumit mod. Unele blocuri prezintă delimitator de sfârșit de bloc, în timp ce pentru alte blocuri acest delimitator este opțional sau chiar interzis. Între cele doua marcaje **<html>** și **</html>** vom introduce doua secțiuni:

I. Secțiunea de antet <head>...</head>

```
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <link rel="stylesheet" href="styledef.css">
8   <link rel="stylesheet" href="style.css">
9   <link rel="stylesheet" href="styles/Acasa.css">
10  <title>Atestat</title>
11 </head>
```

linia 4 - definește codificarea caracterelor pe care browserul trebuie să o utilizeze atunci când afișează conținutul site-ului web;

linia 5 - oferă browser-ului instrucțiuni despre cum să afișeze dimensiunile și scalarea paginii;

linia 6 - se folosește pentru a preveni Internet Explorer 8 să afișeze pagina incorect;

linia 7-9 -specifică relația dintre documentul curent și documentul legat, în cazul acesta CSS;

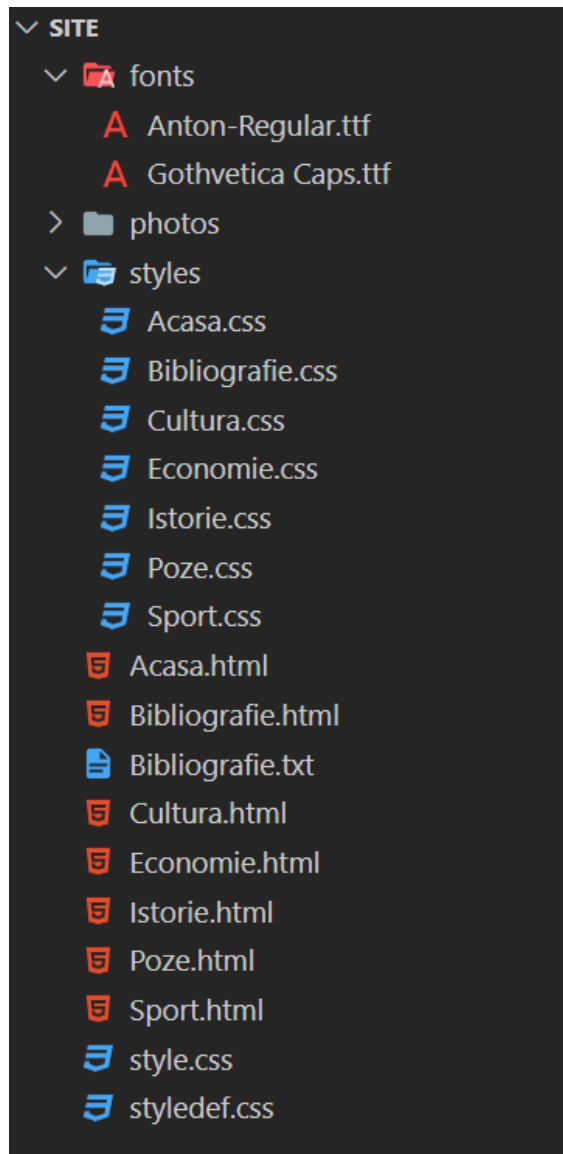
linia 10 - definește titlul site-ului web;

II. Corpul documentului <body>...</body>

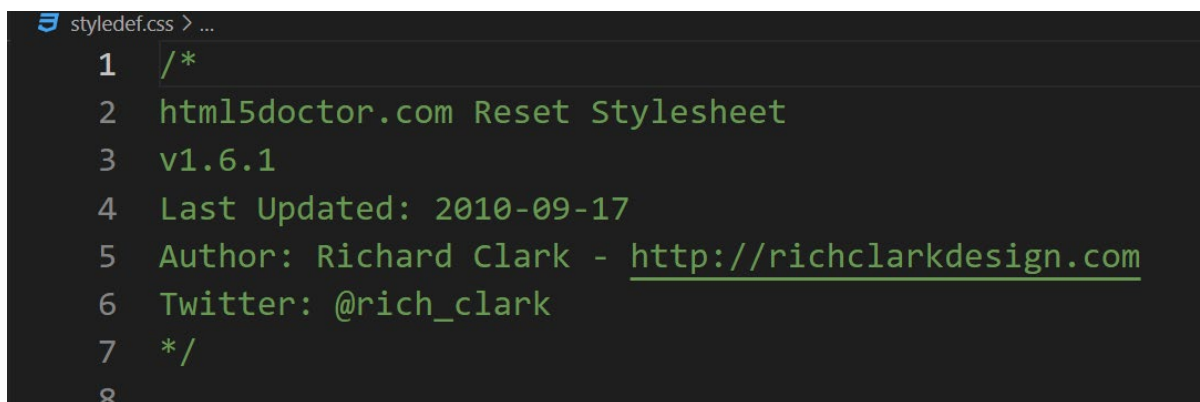
- Blocul <body>...</body> cuprinde conținutul propriu-zis al paginii HTML, adică ceea ce va fi afișat în fereastra browser-ului.

Atestatul a fost organizati în mai multe părți:

- în folder-ul **fonts** se păstrează fonturile personalizate, salvate local;
- în folder-ul **photos** sunt salvate imaginile folosite la crearea site-ului web;
- în folder-ul **styles** se află CSS-ul specific pentru fiecare pagină;
- în folder-ul principal „**site**” există paginile Web, un fișier text unde este scrisă Bibliografia, un **style** care se aplică pentru toate paginile la bara de navigație și un **styledef** care rescrie setările predefinite de la diferite browser-e;




Bara de navigație



styledef.css

CSS

```
18 .header{
19     background:  rgb(50,49,49,0.9);
20     position: fixed;
21     width: 100%;
22     height: 7vh;
23     z-index: 10;
24     align-content: center;
25     backdrop-filter: blur(4px);
26 }
```


Selectorul de clasă (.)

- selectorul CSS este folosit pentru a identifica la ce element se vor aplica regulile și este primul element al definiției unei reguli;
- de exemplu clasa „header” cu atributele ei;

@font-face

- regula **@font-face** CSS specifică un font personalizat pentru afișarea textului;
- fontul poate fi încărcat fie de pe un **server la distanță**, fie de pe un font **instalat local** pe computere-ul utilizatorului;

```
1 @font-face {
2     src: url(fonts/Gothvetica Caps.ttf);
3     font-family: GothCap;
4 }
5
6
7 @font-face {
8     src: url(fonts/Anton-Regular.ttf);
9     font-family: antone;
10 }
```

```
136 @media screen and (max-width: 900px) {
137     .headerli {
138         display: none;
139     }
140
141     .menu2{
142         display:  flex;
143         height: 0;
144     }
```

@media

- regula **@media** CSS poate fi utilizată pentru a aplica o parte a unei foi de stil pe baza rezultatului uneia sau mai multor interogări media;

- cu ajutorul acesteia, se specifică o interogare media și un bloc de CSS care se aplică documentului dacă și numai dacă interogarea media se potrivește cu dispozitivul pe care este utilizat conținutul;
- în cazul de față dacă lățimea ecranului este mai mică de 900px atunci se aplică setările din interiorul acoladelor;

Display

- proprietatea CSS de afișare stabilește dacă un element este tratat ca element de tip block sau inline și aspectul utilizat pentru copiii săi, cum ar fi aspectul de tip flow layout, grid sau flex.

FlexBox CSS

- este un modul CSS care definește un model CSS box optimizat pentru proiectarea interfeței cu utilizatorul și pentru dispunerea elementelor într-o singură dimensiune;
- în modelul de dispunere flex, copiii unui container flex pot fi

```

27 ~ .headerdiv{
28     display: flex;
29     align-items: center;
30     justify-content: center;
31     width: 100%;
32     margin: 0 auto;
33     height: 100%;
34 }
35

```

dispuși în orice direcție și își pot „flexa” dimensiunile, fie crescând pentru a umple spațiul nefolosit, fie micșorându-se pentru a evita depășirea părintelui;

- atât alinierea orizontală („justify”), cât și cea verticală („align”) a copiilor poate fi manipulată cu ușurință;

```

122 .headerli a::after{
123     content: "";
124     background:  red;
125     height: 3px;
126     width: 0;
127     display: block;
128     transition: 0.3s;
129 }
130
131 .headerli a:hover::after{
132     width: 1em;
133 }

```

::after

- în CSS, ::after creează un pseudo-element care este ultimul copil al elementului selectat;
- este adesea utilizat pentru a adăuga conținut cosmetic la un element cu proprietatea content;

:hover

- pseudo-clasa CSS :hover se potrivește atunci când utilizatorul interacționează


cu un element, cu un dispozitiv de indicare, dar nu îl activează neapărat;

- se declanșează atunci când utilizatorul trece cu cursorul (pointerul mouse-ului) peste un element;

::before

- în CSS ca și ::after, ::before creează un pseudo-element, dar care este primul copil al elementului selectat;

```

.hovering::before{
    content: "";
    opacity: 0;
    background-color:  rgba(0,0,0,0.5);
}

```

CSS Transforms

- este un modul al CSS care definește modul în care elementele stilizate cu CSS pot fi transformate în spațiul bidimensional sau tridimensional;
- funcția CSS translate() repoziționează un element în direcția orizontală și/sau verticală;

```

69 .hovering .para h1{
70     text-align: center;
71     transform: translate(0, -70%);
72 }

```

clamp()

- funcția CSS clamp() fixează o valoare medie într-un interval de valori cuprins între o limită minimă și o limită maximă definite (folosită în acest caz pentru modificarea mărimii textului);

```
74 .para p{
75     text-align: center;
76     font-size: clamp(1.1em, 1.2vw, 3em);
77 }
```

aspect-ratio

- proprietatea CSS aspect-ratio stabilește un raport de aspect preferat pentru casetă, care va fi utilizat la calcularea dimensiunilor automate și a altor funcții de aspect;

```
137 .whitezone iframe{
138     width: 80vw;
139     aspect-ratio: 16/9;
140 }
```

CSS Grid

- grid excelează la împărțirea unei pagini în regiuni majore sau la definirea relației în termeni de dimensiune, poziție și strat, între părțile unui control construit din primitive HTML;
- la fel ca tabelele, aspectul grilă permite autorului să alinieze elementele în coloane și rânduri;
- cu toate acestea, cu grila CSS sunt posibile sau mai ușor de realizat mai multe layout-uri decât cu tabelele;

```
40 ~ .grid{
41     display: grid;
42     grid-template-columns: repeat(10,10%);
43     grid-row-gap: 5vh;
44 }
```

grid-template-columns

- proprietatea CSS grid-template-columns definește numele liniilor și funcțiile de dimensionare coloanelor grilei;

grid-row-gap

- proprietatea CSS grid-row-gap stabilește dimensiunea spațiului dintre rândurile grilei;

```
79 ~ .whitetxt.r {  
80   grid-column-start: 7;  
81   grid-column-end: 9;  
82 }
```

grid-column-start

- proprietatea CSS grid-column-start specifică poziția de început a unui element

de grilă în cadrul coloanei, contribuind cu o linie, un interval sau nimic (automat) la plasarea sa în grilă;


- această poziție de start definește marginea de început a blocului din zona grilei;

grid-column-end

- ca și proprietatea grid-column-start specifică poziția elementului de grilă în cadrul coloanei, dar nu cea de început, ci cea finală, specificând marginea de sfârșit;

CSS Transitions

- este un modul de CSS care permite crearea de tranziții graduale între valorile anumitor proprietăți CSS;
- comportamentul acestor tranziții poate fi controlat prin specificarea funcției de sincronizare, a duratei și a altor atribute:

```
59   color:  #a2a2a2;  
60   transition: 0.3 ease;  
61   font-size: clamp(1.4em, 3vw, 3em);
```

transition

- proprietatea CSS de tranziție este o proprietate prescurtată pentru transition-property, transition-duration, transition-timing-function și transition-delay;

```
234 @keyframes fade-in {  
235     0% {  
236         max-height: 0;  
237         opacity: 0;  
238     }  
239     100% {  
240         max-height: 100vh;  
241         opacity: 1;  
242     }  
243 }
```

CSS Animations

- CSS Animations este un modul care permite animarea valorilor unor proprietăți CSS în timp, folosind cadre cheie;

@keyframes

- regula CSS @keyframes controlează etapele intermediare dintr-o secvență de animație

CSS prin definirea stilurilor pentru cadrele cheie (sau punctele de referință) de-a lungul secvenței de animație;

- acest lucru oferă mai mult control asupra etapelor intermediare ale secvenței de animație decât tranzițiile;

Pagina „Home”

- aceasta este construită în elementul <body>, care are ca și **background** steagul Japoniei (background-image);
- în <body> se află un element <header> și două elemente <div>;

Header

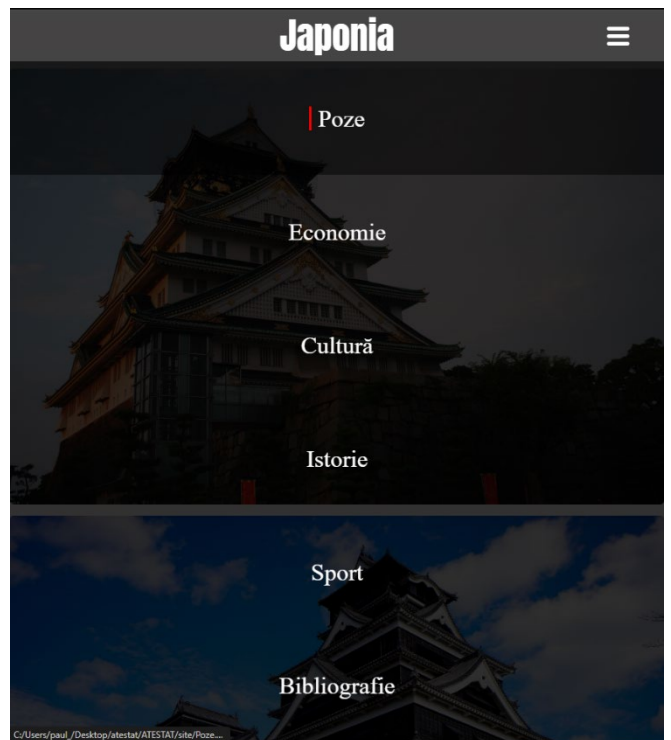
- în **header** s-a construit bara de navigație, comună pentru toate paginile;

- sub numele paginii active se află o bară roșie, construită cu pseudo-element-ul **::after**;
- când lățimea paginii scade sub o valoare specifică, link-urile de la pagini dispar (display: none), iar un **label** apare, cu rol de a deschide **noul meniu de navigare**;



Div-ul „menu2”

- este comun pentru toate paginiile;
- cu ajutorul elementului `<input>`, care se activează la apăsarea **label-ului** din **header**, se deschide **div-ul „menu2”**;
- în stânga paginii active se află o bară roșie, construită cu pseudo-element-ul **::before**;
- de asemenea, casuța respectivă paginii își schimbă **background-ul** când mouse-ul se află pe acesta;
- aceste efecte sunt valabile pentru toate paginiile în afară de pagina „Home”, care are ca și button logo-ul site-ului;



Div-ul „content”

- acesta prezintă informații generale despre „Japonia”;
- informațiile sunt structurate în două tipuri de `<div>`: unul cu text-ul centrat în mijloc, iar unul împărțit în două părți, una pentru imagine și cealaltă pentru text;
- când lățimea paginii scade sub o valoare specifică, imaginea se mută deasupra textului;

Pagina „Poze”

- aceasta este construită în elementul <body>;
- în <body> se află un element <header> și două elemente <div>;

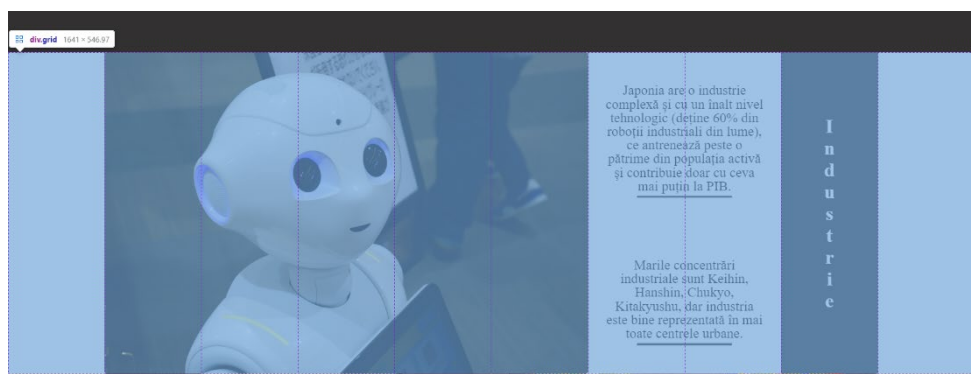
Div-ul „content”

- conține mai multe div-uri „**hovering**”, respectiv pentru fiecare imagine;
- în funcție de lățimea paginii, există între trei și o coloană pentru afișarea imaginii;
- când mouse-ul se află pe o poză, informații despre aceasta apar;



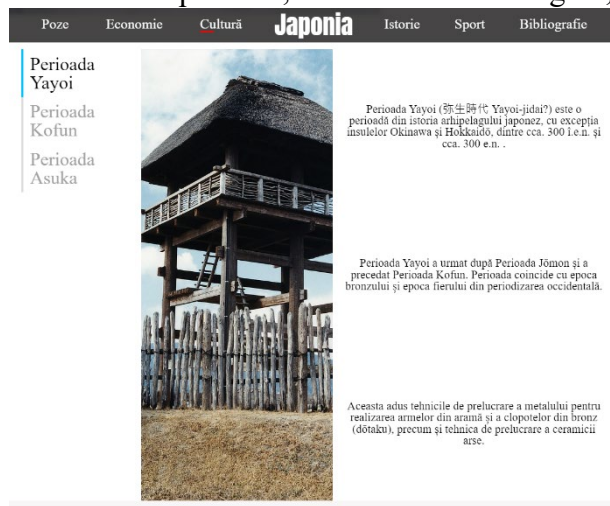
Pagina „Economie”

- div-ul „content” conține două tipuri de div: unul cu text-ul centrat în mijloc, iar unul sub formă de **grid**, împărțit în zece coloane;
- fiecare div de formă **grid** are trei părți: un div pentru poza, unul pentru text și un div pentru titlu domeniului;



Pagina „Cultură”

- div-ul „content” conține trei tipuri de div: un div cu text-ul centrat în mijloc, unul împărțit în două părți, una pentru imagine și cealaltă pentru text și un div care prezintă un meniu și o zonă cu informațiile specifice titlului selectat;
- când lățimea paginii scade sub o valoare specifică, meniul cu titlurile se mută deasupra, iar în celălalt tip de div, textul trece sub imagine;



- Paginile „Istorie” și „Sport” respectă același design ca și pagina „Cultură”.

Pagina Bibliografie

- conține două div-uri centrate pe mijloc, unde unul este pe post de **slideshow** cu logo-urile paginilor principale și un div prezintă **link-urile** tuturor paginilor, folosite în realizarea site-ului;



Concluzie

Japonia este o țară bogată în cultură și tehnologie, care merită vizitată și valorificată ca o experiență de neuitat, iar prezentarea acesteia ca și temă pentru proiectul de atestat a fost plăcere, din care am învățat cât de diversificată este lumea și că întotdeauna este loc de mai bine, de îmbunătățire a modului de lucru.

Bibliografie

- https://en.wikipedia.org/wiki/Visual_Studio_Code
- https://ro.wikipedia.org/wiki/Visual_Studio_Code
- <https://caniuse.com>
- <https://code.visualstudio.com/docs/editor/extension-marketplace>
- https://ro.wikipedia.org/wiki/HyperText_Markup_Language
- <https://web.ceiti.md/lesson.php?id=1>
- <https://devdocs.io>