

**Universitatea Tehnică a Moldovei**  
**Facultatea *Calculatoare, Informatică și Microelectronică***  
**Specialitatea *Tehnologii Informaționale***



# Raport

**la lucrarea de laborator nr. 1**

**Tema: “*Conceperea unei aplicatii WEB.*”**

***Crearea continutului Web utilizand HTML CSS”***

**Disciplina: “Tehnologii web”**

**A efectuat:**

Student grupa TI-231 FR

Apareci Aurica

**A verificat:**

Asistent universitar

Rusu Viorel

**Chișinău 2025**

# Cuprins

1. Cadru teoretic.....	3
2. Repere teoretice.....	4
3. Sarcini practice.....	6
3.1 Instalare server Web .....	6
3.2 HTML5/CSS .....	8
4. Listingul progmului.....	13
5. Concluzii.....	14
6. Webografie.....	15

## **1. Cadru teoretic**

**Tema lucrării:** Conceperea unei aplicații web. Crearea conținutului Web utilizând HTML, CSS.

**Scopul lucrării:** Crearea conținutului Web utilizând HTML, CSS.

**Sarcina practică:**

### **1.1 Instalare server Web**

- a) Se studiază diferite aplicații/pachete ce permit instalarea serverului web pe local (exemple: XAMPP, WAMP, EasyPHP, AMPPS, AppServ, etc.);
- b) Se motivează alegerea unuia din acestea și se instalează pe calculator;
- c) Studiați posibilitățile de a crea hosturi virtuale, creați câteva;
- d) Creați câteva pagini web pentru testarea serverului web.

### **1.2 HTML5/CSS**

- a) alegeți un subiect pentru a dezvolta un site;
- b) descrieți succint acesta;
- c) creați modelul funcțional al site-ului;
- d) creați paginile informative statice utilizând HTML/CSS (acestea trebuie să conțină meniuri derulante, blocuri dinamice, formulare, s.a.). Trebuie separat explicit conținutul HTML de CSS.

## 2. Repere teoretice

**Pagină web** - o unitate de conținut accesibilă pe internet sau intranet, construită folosind limbaje precum HTML, CSS și, uneori, JavaScript, care poate conține texte, imagini, videoclipuri și alte elemente interactive.

**HTML (HyperText Markup Language)** - limbajul standard folosit pentru a structura și prezenta conținutul pe paginile web. Folosește etichete pentru a defini structura documentului.

**CSS (Cascading Style Sheets)** - un limbaj utilizat pentru a stiliza elementele HTML, controlând aspectul vizual, cum ar fi culorile, fonturile, spațierile și layout-ul general.

**Browser** - o aplicație software care permite utilizatorilor să vizualizeze și să interacționeze cu paginile web. Exemple: Google Chrome, Mozilla Firefox, Microsoft Edge.

**Server web** - un software sau un sistem hardware care găzduiește și servește paginile web către utilizatori prin internet sau rețele locale, folosind protocoale precum HTTP sau HTTPS. Exemple: Apache, Nginx, IIS.

**Host virtual** - o configurație a unui server web care permite găzduirea mai multor site-uri web pe același server, fiecare cu propriul domeniu sau subdomeniu.

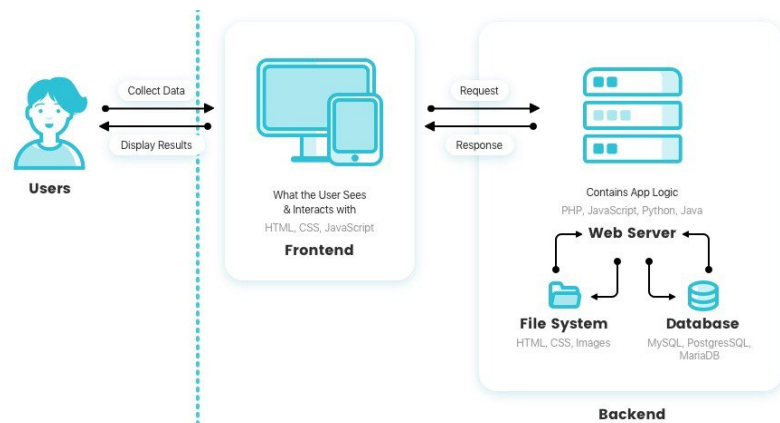


Figura 1 Arhitectura unei aplicații web

**Frontend** - reprezintă partea vizibilă și interactivă a unei aplicații web, accesibilă utilizatorilor finali prin intermediul unui browser. Este responsabil pentru design, structură și funcționalitate la nivel de interfață.

**Tehnologii utilizate:** HTML, CSS, JavaScript și framework-uri (React, Angular, Vue.js).

**Responsabilități principale:**

Crearea layout-ului și design-ului paginii.

Implementarea elementelor interactive (meniuri, animații, formulare).

Optimizarea pentru o experiență user-friendly.

**Backend** - reprezintă partea ascunsă a aplicației web, care gestionează logica aplicației, procesarea datelor și comunicarea cu baza de date. Acesta rulează pe un server.

**Tehnologii utilizate:** limbaje de programare (Python, PHP, C#), baze de date (MySQL).

**Responsabilități principale:**

Stocarea și gestionarea datelor în baza de date.

Implementarea business logicii.

Gestionarea cererilor primite de la frontend și trimiterea răspunsurilor.

**Client-side** - partea unei aplicații care rulează pe dispozitivul utilizatorului (în browser). Este responsabil pentru interacțiunile directe cu utilizatorul și reprezintă funcționalitatea implementată pe partea de frontend.

**Server-side** - partea aplicației care rulează pe server și gestionează cererile primite de la client (frontend). Este utilizată pentru operațiuni mai complexe, securizate și care necesită acces la baze de date sau alte resurse externe.

### 3. Sarcini practice

#### 3.1 Instalare server Web

a) Se studiază diferite aplicații/pachete ce permit instalarea serverului web pe local (exemplu: XAMPP, WAMP, EasyPHP, AMPPS, AppServ, etc.);

**XAMPP** este o soluție multiplatformă care include serverul web Apache, MariaDB, PHP și Perl, fiind ideală pentru dezvoltarea și testarea aplicațiilor web pe un mediu local.

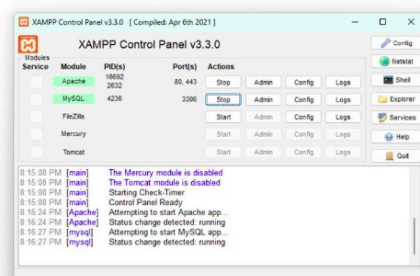
**WAMP** este un pachet pentru Windows care instalează Apache, MySQL și PHP, simplificând configurarea unui server local pentru dezvoltatori.

**EasyPHP** oferă o platformă ușor de utilizat pentru instalarea Apache, MySQL și PHP, fiind potrivită pentru testarea rapidă a aplicațiilor web.

**AMPPS** este un pachet cu o gamă extinsă de aplicații open-source care include Apache, MySQL, PHP, Perl și Python, având o interfață ușor de gestionat.

b) Se motivează alegerea unuia din acestea și se instalează pe calculator;

**XAMPP** reprezintă o soluție software open-source destinată dezvoltatorilor web care doresc să creeze, să testeze și să ruleze aplicații web într-un mediu local. Numele său provine de la componentele principale ale pachetului: **X: Cross-platform** (compatibil cu Windows, Linux și macOS), **A: Apache** (server web), **M: MariaDB** (bază de date), **P: PHP** (limbaj de programare server-side), **P: Perl** (limbaj de programare).

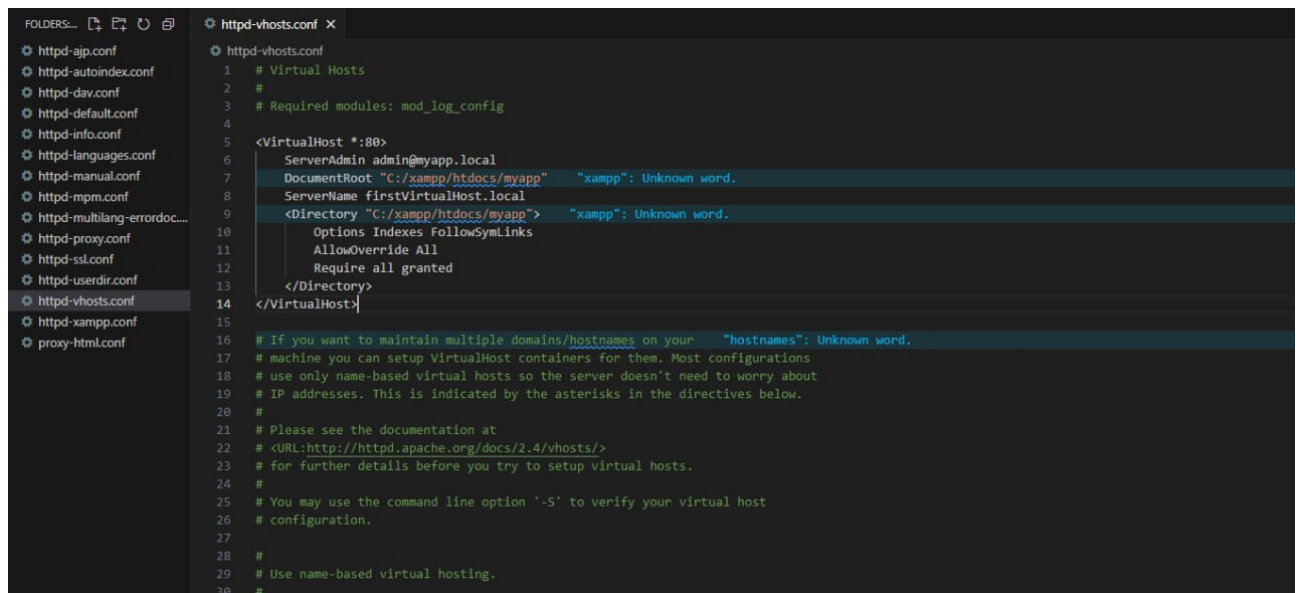


**XAMPP** este conceput pentru a elimina complexitatea configurării unui mediu de server local. Instalarea este rapidă și intuitivă, necesitând doar câțiva pași simpli pentru a fi funcțional. Datorită designului său cross-platform, XAMPP poate fi utilizat pe diverse sisteme de operare, ceea ce îl face o soluție versatilă pentru utilizatori.

c) Studiați posibilitățile de a crea hosturi virtuale, creați câteva;

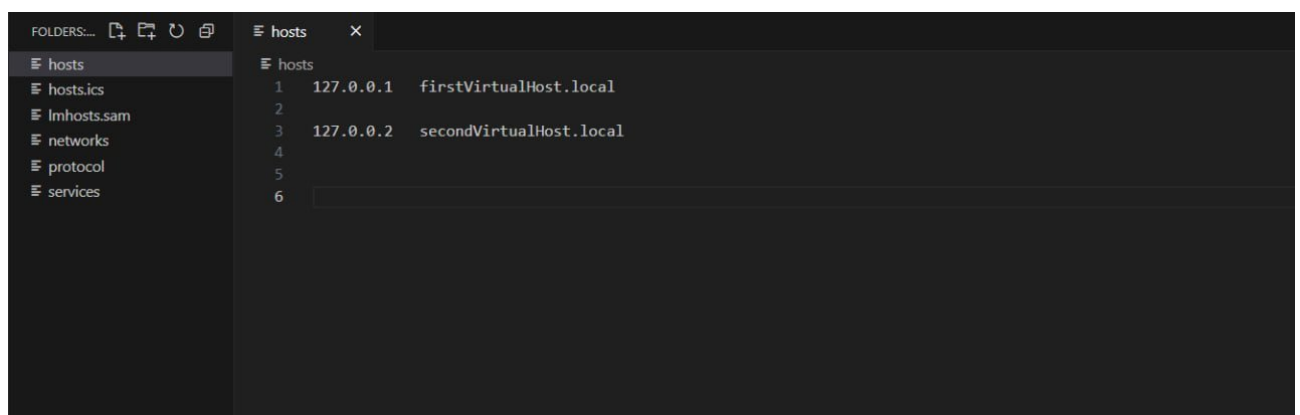
**Hosturile virtuale** permit rularea mai multor site-uri web sau aplicații pe același server local, utilizând domenii personalizate (test.dev). Acest lucru este util pentru separarea proiectelor și pentru simularea mediului live pe un server local.

Accesam `C:\xampp\apache\conf\extra\httpd-vhosts.conf`, adaugam configuratia pentru hostul virtual



```
1 # Virtual Hosts
2 #
3 # Required modules: mod_log_config
4
5 <VirtualHost *:80>
6     ServerAdmin admin@myapp.local
7     DocumentRoot "C:/xampp/htdocs/myapp" "xampp": Unknown word.
8     ServerName firstVirtualHost.local
9     <Directory "C:/xampp/htdocs/myapp"> "xampp": Unknown word.
10         Options Indexes FollowSymLinks
11         AllowOverride All
12         Require all granted
13     </Directory>
14 </VirtualHost>
15
16 # If you want to maintain multiple domains/hostnames on your "hostnames": Unknown word.
17 # machine you can setup VirtualHost containers for them. Most configurations
18 # use only name-based virtual hosts so the server doesn't need to worry about
19 # IP addresses. This is indicated by the asterisks in the directives below.
20 #
21 # Please see the documentation at
22 # <URL:http://httpd.apache.org/docs/2.4/vhosts/>
23 # for further details before you try to setup virtual hosts.
24 #
25 # You may use the command line option '-S' to verify your virtual host
26 # configuration.
27 #
28 #
29 # Use name-based virtual hosting.
30 #
```

Deschidem `C:\Windows\System32\drivers\etc\hosts` si adaugăm următoarea linie pentru a lega domeniul local de adresa IP. Dupa care repornim serverul. La accesarea, url-urilor configurate, din browserul web vom putea vizualiza paginile respective.



```
1 127.0.0.1 firstVirtualHost.local
2
3 127.0.0.2 secondVirtualHost.local
4
5
6
```

## 3.2 HTML5/CSS

a) alegeți un subiect pentru a dezvolta un site;

Pentru realizarea sarcinii am ales să dezvolt un site pentru rezervarea camerelor de hotel, un instrument digital indispensabil în era modernă, care oferă utilizatorilor posibilitatea de a planifica și gestiona călătoriile într-un mod comod și eficient. Acest site va reprezenta o platformă centralizată, unde utilizatorii pot găsi informații detaliate despre hoteluri, camere disponibile, prețuri și facilități, eliminând nevoia de a căuta informațiile pe mai multe surse.

b) descrieți succint acesta;

Prin integrarea unor funcționalități precum căutarea avansată, filtrarea rezultatelor, vizualizarea recenziilor altor utilizatori și plata online securizată, site-ul va transforma procesul de rezervare într-o experiență intuitivă și rapidă. De asemenea, platforma va permite utilizatorilor să își personalizeze căutările, ținând cont de criterii precum locația, bugetul, facilitățile necesare (sau preferințele privind tipul camerei). Acest proiect a fost planificat pentru a răspunde nevoilor diverse ale clienților și pentru a aduce valoare adăugată în domeniul turismului digital.

Cerinte functionale:

### **Sistem de căutare și filtrare a camerelor disponibile**

- Utilizatorii trebuie să poată introduce criterii de căutare (destinație, dată de check-in și check-out, nr persoane) pe pagina principală sau în secțiunea dedicată camerelor.
- Filtrele adiționale trebuie să includă opțiuni precum preț, facilități (Wi-Fi, piscină, parcare), clasificarea hotelului sau recenzii.

### **Formular de contact**

- Utilizatorii trebuie să poată completa un formular cu numele, adresa de email și un mesaj. Mesajul trimis trebuie să fie gestionat de un script care confirmă primirea acestuia. Formularul trebuie să valideze corect datele.

### **Pagini dinamice pentru camere**

- Pagina *rooms.html* trebuie să afișeze lista camerelor disponibile, iar fiecare cameră să includă detalii precum: preț pe noapte, descriere, imagini și recenzii.
- Blocuri dinamice trebuie să afișeze camere recomandate sau reduceri speciale în funcție de utilizator (ex. camere disponibile în destinațiile căutate recent).

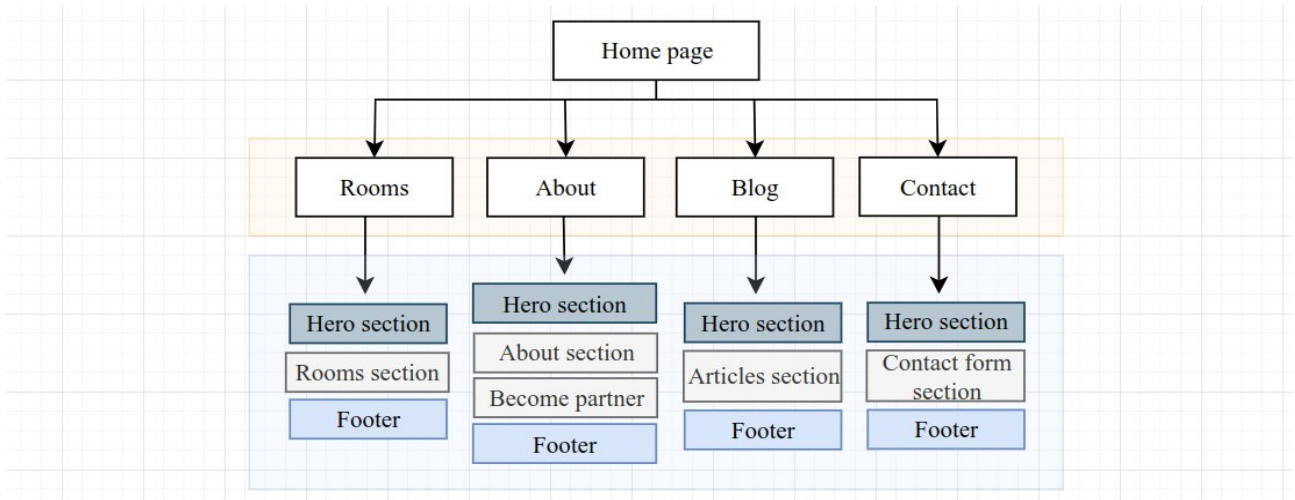
### **Meniu derulant pentru navigare între pagini**

- Toate paginile trebuie să includă un meniu derulant care să permită accesul rapid la secțiunile principale.
- Meniul trebuie să fie vizibil și funcțional atât pe desktop, și pe dispozitive mobile.

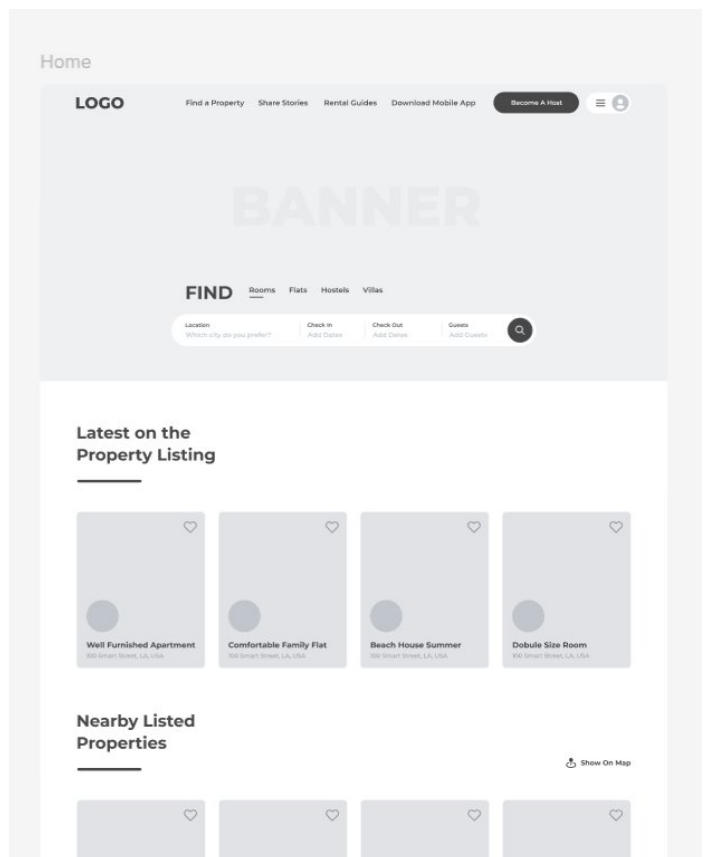
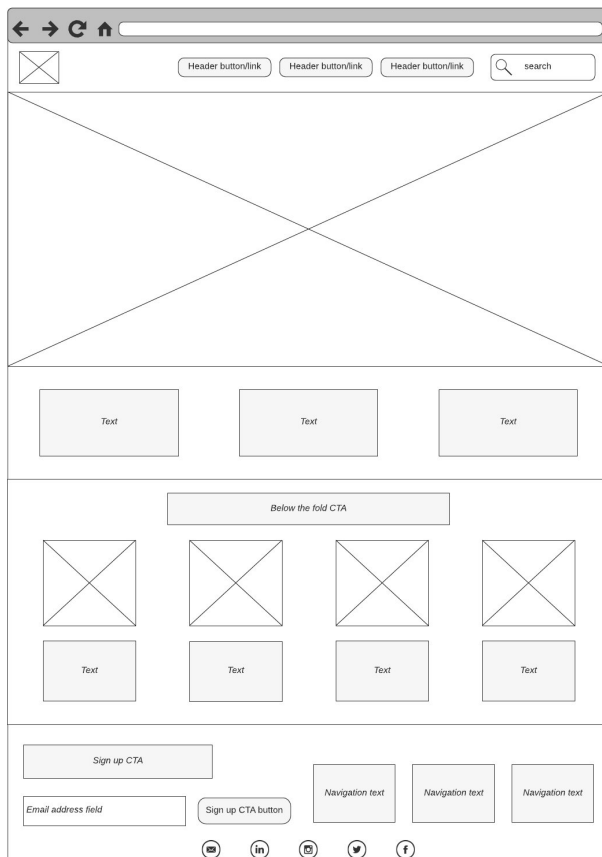


c) creati modelul functional al site-ului;

În planificarea unui site, **sitemap-ul** joacă un rol esențial deoarece oferă o structură clară și logică a paginilor, facilitând organizarea conținutului și navigația utilizatorilor. De asemenea, acesta ajută dezvoltatorii să vizualizeze fluxul de informații și relațiile dintre pagini, contribuind la o experiență de utilizare intuitivă și eficientă.

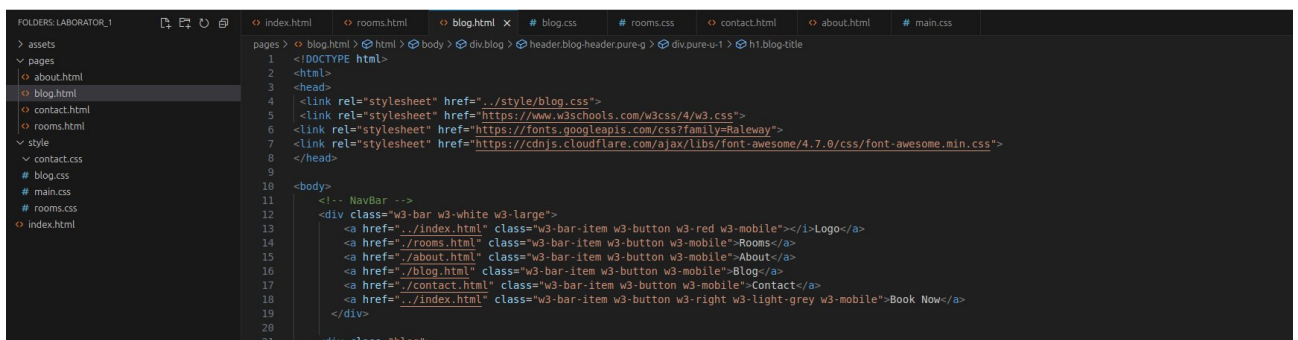


Un **wireframe** este esențial în procesul de planificare a unui site deoarece reprezintă o schemă vizuală simplificată care ilustrează structura și layout-ul paginilor. Acesta ajută echipa de dezvoltare și design să stabilească ierarhia informațiilor, poziționarea elementelor și fluxul utilizatorului, asigurând o aliniere clară între cerințele funcționale și design-ul final.



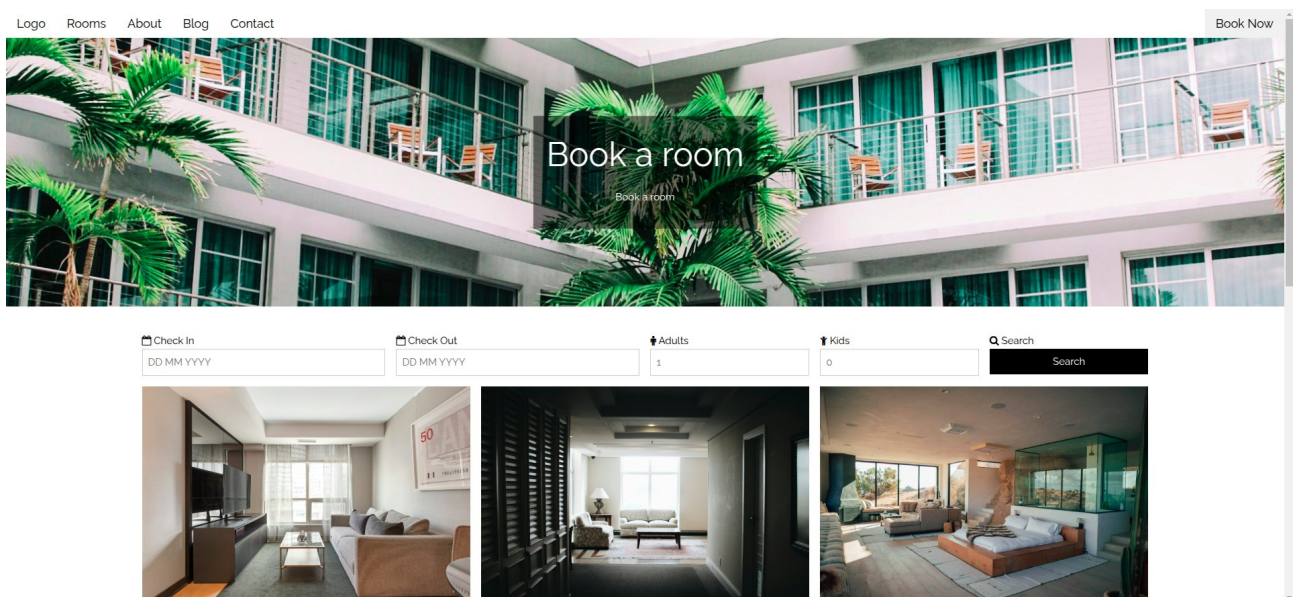
d) creați paginile informative statice utilizând HTML/CSS (acestea trebuie să conțină meniuri derulante, blocuri dinamice, formulare, ș.a.). Trebuie separat explicit conținutul HTML de CSS.

Pentru dezvoltarea aplicației web, am utilizat *Visual Studio Code* ca mediu principal de dezvoltare datorită funcționalităților sale avansate, precum suportul pentru HTML, CSS, integrarea extensiilor pentru dezvoltare web, și interfața intuitivă. Proiectul a fost organizat conform unei structuri bine definite a folderelor, asigurând o gestionare clară a fișierelor. Structura proiectului include un folder *pages/*, unde sunt stocate paginile web individuale (e.g., *about.html*, *contact.html*, *rooms.html*, *blog.html*), un folder *style/* pentru fișierele CSS (*main.css*, *rooms.css*, *contact.css*, *blog.css*), și un folder *assets/* pentru stocarea resurselor suplimentare, precum imagini sau alte fișiere media.

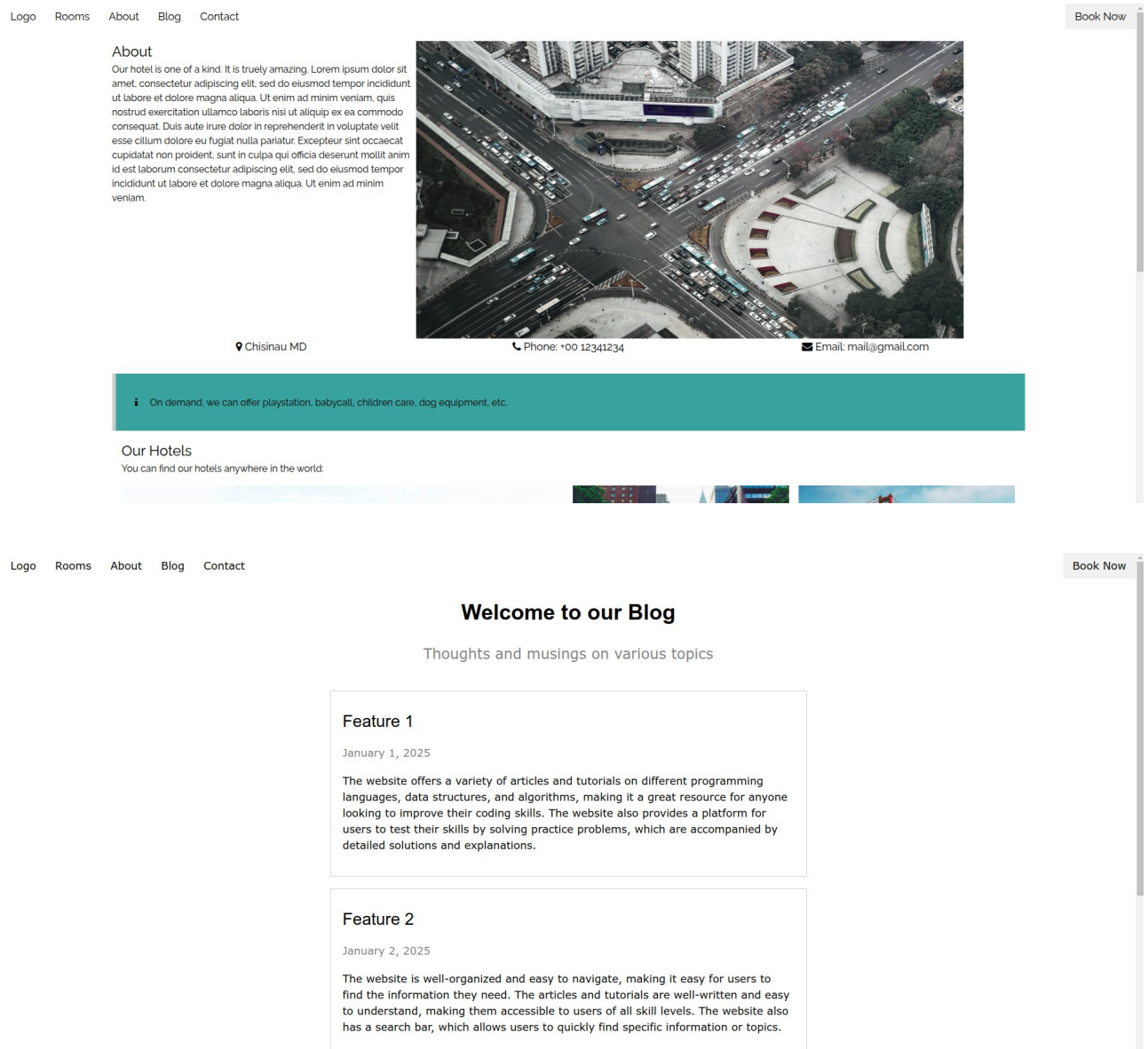


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <link rel="stylesheet" href="../style/blog.css">
5 <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
6 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway">
7 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
8 </head>
9
10 <body>
11 <!-- NavBar -->
12 <div class="w3-bar w3-white w3-large">
13 <a href="../index.html" class="w3-bar-item w3-button w3-red w3-mobile"></i>Logo</a>
14 <a href="../rooms.html" class="w3-bar-item w3-button w3-mobile">Rooms</a>
15 <a href="../about.html" class="w3-bar-item w3-button w3-mobile">About</a>
16 <a href="../blog.html" class="w3-bar-item w3-button w3-mobile">Blog</a>
17 <a href="../contact.html" class="w3-bar-item w3-button w3-mobile">Contact</a>
18 <a href="../index.html" class="w3-bar-item w3-button w3-right w3-light-grey w3-mobile">Book Now</a>
19 </div>
20
21 <div class="blog">
```

În cadrul lucrării de laborator, am realizat o aplicație web care include un *meniu interactiv* ce permite navigarea facilă între paginile componente ale site-ului. Meniul este structurat sub forma unor ancore HTML, fiecare dintre acestea fiind legată către o pagină specifică a aplicației. Această implementare asigură o experiență de utilizare intuitivă și eficientă, oferind utilizatorilor posibilitatea de a accesa rapid informațiile dorite. Designul meniului a fost realizat astfel încât să fie responsiv și accesibil, fiind testat pentru funcționalitate atât pe desktop, cât și pe dispozitive mobile.



Pentru a optimiza aplicația web din perspectiva SEO, am integrat și o pagină dedicată afacerii pe care o promovează afacerea și una specifică bloggingului. Aceasta permite adăugarea de conținut dinamic, actualizat periodic, ceea ce contribuie la creșterea relevanței site-ului în motoarele de căutare.



În cadrul dezvoltării aplicației web, instrumentele de bază utilizate au fost **HTML** și **CSS**, asigurând o structură solidă și un design atractiv. Structura paginilor a fost construită folosind elemente HTML comune, precum:

<header> – pentru bara de navigare și titlul principal al paginii;

<nav> – pentru meniul de navigare, care conține ancorele (<a>) către celelalte pagini;

<footer> – pentru informațiile de contact și link-urile către rețelele sociale;

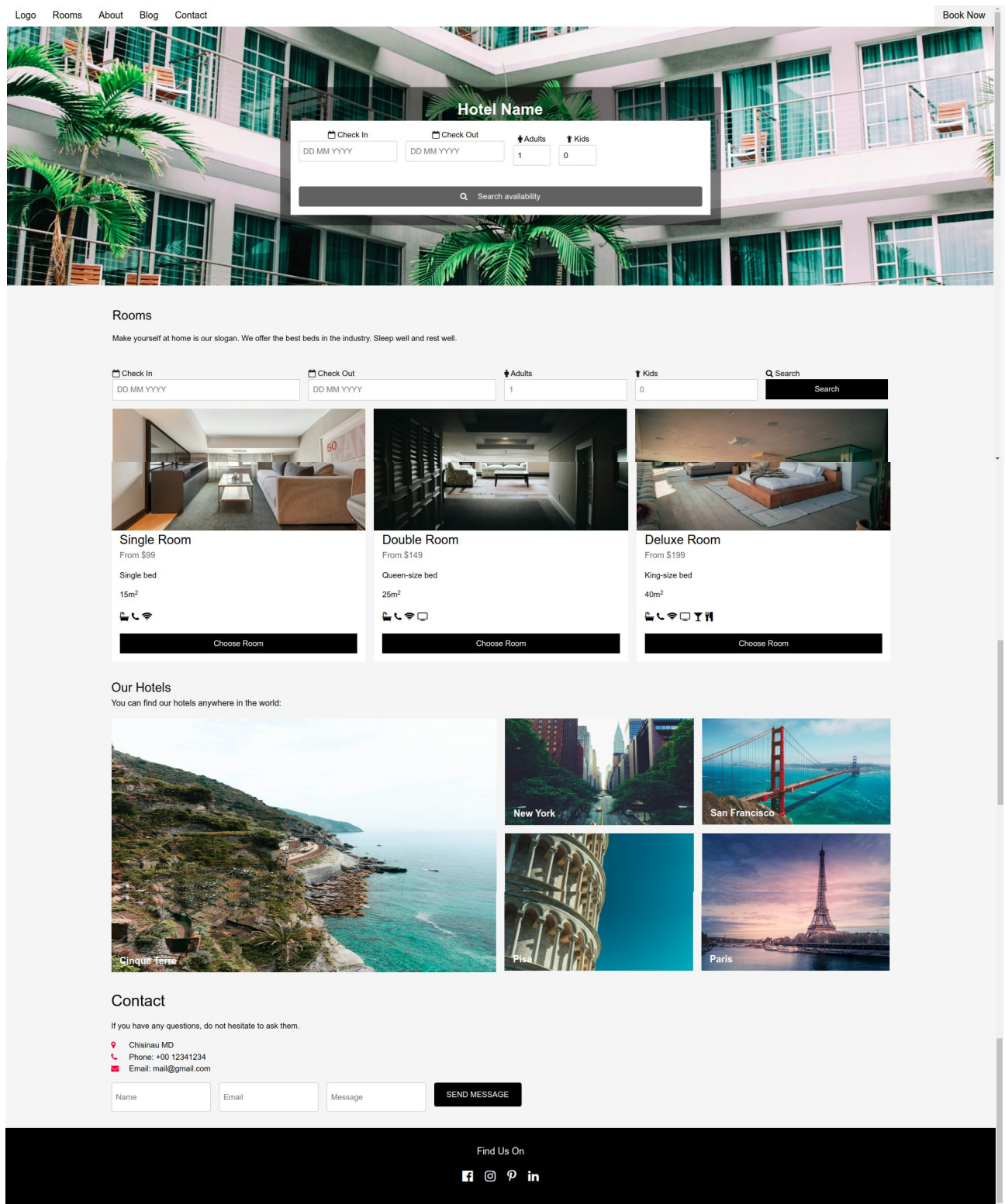
<img> – pentru includerea imaginilor, completate cu attribute alt pentru optimizare SEO;



<h1> - <h3> – pentru ierarhizarea titlurilor și subcapitolelor;

<div> – pentru gruparea elementelor și aplicarea de stiluri personalizate.

Stilizarea a fost realizată cu CSS, utilizând clase și ID-uri definite în mod clar pentru personalizarea aspectului fiecărui element. Clasele CSS au fost utilizate pentru elemente comune, cum ar fi butoanele, cardurile de articole și secțiunile de conținut. De asemenea, s-au folosit pseudoclasă precum :hover și :focus pentru a adăuga interactivitate, îmbunătățind experiența utilizatorului.



## 4. Listingul progmu-lui

```
<!-- NavBar -->
<div class="w3-bar w3-white w3-large">
  <a href="../index.html" class="w3-bar-item w3-button w3-mobile"></i>Logo</a>
  <a href="../rooms.html" class="w3-bar-item w3-button w3-mobile">Rooms</a>
  <a href="../about.html" class="w3-bar-item w3-button w3-mobile">About</a>
  <a href="../blog.html" class="w3-bar-item w3-button w3-mobile">Blog</a>
  <a href="../contact.html" class="w3-bar-item w3-button w3-mobile">Contact</a>
  <a href="../index.html" class="w3-bar-item w3-button w3-right w3-light-grey w3-mobile">Book Now</a>
</div>

<div class="hero-image">
  
  <div class="hero-text">
    <h1 style="font-size:50px">Book a room</h1>
    <p>Book a room</p>
  </div>
</div>

<!-- Page content -->
<div class="w3-content" style="max-width:1532px;">
  <div class="w3-row-padding">
    <div class="w3-col m3">
      <label><i class="fa fa-calendar-o"></i> Check In</label>
      <input class="w3-input w3-border" type="text" placeholder="DD MM YYYY">
    </div>
    <div class="w3-col m3">
      <label><i class="fa fa-calendar-o"></i> Check Out</label>
      <input class="w3-input w3-border" type="text" placeholder="DD MM YYYY">
    </div>
    <div class="w3-col m2">
      <label><i class="fa fa-male"></i> Adults</label>
```

```
.bar {
  width: 100%;
  overflow: hidden;
  background-color: #ffffff;
  border-bottom: 1px solid #ddd;
}

.bar-item {
  padding: 12px 16px;
  display: inline-block;
  text-align: center;
  color: #000;
  text-decoration: none;
  transition: background-color 0.3s ease;
}

.bar-item:hover {
  background-color: #f1f1f1;
  color: #333;
}

.bar-item.w3-red {
  background-color: #f44336;
  color: #fff;
}

header {
  position: relative;
  width: 100%;
  height: auto;
}
```

## 5. Concluzii

În cadrul acestei lucrări, am realizat cu succes toate sarcinile necesare pentru conceperea unei aplicații web utilizând HTML și CSS. Procesul de lucru a inclus pași esențiali pentru înțelegerea și implementarea tehnologiilor necesare, precum și pentru dezvoltarea unui site web care să îndeplinească cerințele funcționale și estetice.

Pentru realizarea sarcinii, am analizat mai multe aplicații pentru instalarea unui server web local, printre care XAMPP, WAMP și EasyPHP. După evaluarea funcționalităților și a ușurinței de utilizare, am ales să instalez XAMPP datorită compatibilității sale extinse cu diferite sisteme de operare, a interfeței intuitive. După instalare, am creat hosturi virtuale pentru gestionarea mai multor proiecte în mod simultan. Acestea mi-au permis să configurez și să accesez fiecare proiect ca pe un site separat, utilizând nume personalizate. De asemenea, am realizat și testat câteva pagini web simple pentru a mă asigura că serverul funcționează corect și răspunde cerințelor dezvoltării locale.

Subiectul ales pentru dezvoltarea site-ului a fost rezervarea camerelor de hotel, un proiect util și aplicabil în domeniul turismului digital. Am realizat o descriere detaliată a site-ului și s-a conceput un model funcțional clar, care include structura paginilor, funcționalitățile necesare și navigarea între secțiuni. Am dezvoltat pagini web statice utilizând HTML pentru structură și CSS pentru stilizare, respectând principiul separării conținutului de design. Site-ul include meniuri derulante, blocuri dinamice și formulare, oferind o interfață intuitivă și atractivă pentru utilizatori.

În concluzie, lucrarea a permis aplicarea practică a cunoștințelor despre HTML5 și CSS în realizarea unui site web funcțional, punând accent pe organizarea structurată a conținutului și pe utilizarea unui mediu de dezvoltare local eficient. Prin respectarea cerințelor sarcinii practice, am demonstrat înțelegerea conceptelor fundamentale și capacitatea de a implementa soluții web moderne, adaptabile cerințelor utilizatorilor și contextului tehnologic actual.

## 6. Webografie

1. <https://else.fcim.utm.md/course/view.php?id=2318>
2. <https://developer.mozilla.org/en-US/docs/Web/HTML>
3. <https://developer.mozilla.org/en-US/docs/Web/CSS>
4. <https://www.apachefriends.org/>
5. <https://www.figma.com/>
6. <https://moqups.com/>
7. <https://app.diagrams.net/>
8. <https://www.w3schools.com/>
9. <https://chatgpt.com/>