

# Manipulacija slikama na webu

---

**Kus, Mia**

**Undergraduate thesis / Završni rad**

**2015**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:216:745730>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-11-05**



*Repository / Repozitorij:*

[Faculty of Graphic Arts Repository](#)



SVEUČILIŠTE U ZAGREBU  
GRAFIČKI FAKULTET

# ZAVRŠNI RAD

Mia Kus



Sveučilište u Zagrebu  
Grafički fakultet

Smjer: tehničko tehnološki

# ZAVRŠNI RAD

## MANIPULACIJA SLIKAMA NA WEBU

Mentor:

doc.dr.sc. Tajana Koren Ivančević

Studentica:

Mia Kus

U Zagrebu, 2015.

## **SAŽETAK**

U završnom radu prikazat će se postupak uvođenja slika na web stranicu. Izradit će se galerije slika pomoću kojih se slike prikazuju na jedinstven način. Također, definirati se mjesto gdje se slike, određene galerije, prikazuju. Prikazat će se neke od mogućih funkcija slike, kao što su: cijela slika koja može biti link te dijelovi slike koji služe kao linkovi koji vode u definirano područje. Pridruživanje različitih linkova na različita područja jedne slike uz korištenje slikovnih mapa. To je moguće uz definiranje različitih područja dostupnih na slici. Kako bi se postigla dinamika na webu, upotrebljavaju se tranzicije. Neke od tranzicija koje se koriste su: promjena opaciteta, promjena pozicije određenog objekta, promjena boje te promjena dimenzija elementa. Cjelokupan eksperimentalni rad se izvodi uz pomoć HTML i CSS jezika. Kroz galerije slika prikazuje se potpun proces izrade povijesnog predmeta ("Ranokršćanska svjetiljka"). Cilj ovog završnog rada je napraviti funkcionalnu galeriju slika, namijenjenu za web stranicu, koja će doprinijeti u prepoznavanju i oživljavanju replike već spomenutog povijesnog predmeta, suvenira grada Siska.

Ključne riječi: uvođenje slika, galerije slika, slikovne mape, tranzicije.

## **ABSTRACT**

This bachelor thesis will show the process of introducing images on a website. Photo galleries will be made so that images can be shown in a unique way. Also, the place where the photos of a certain gallery are shown will be defined. It will show some of the possible functions of images, such as: full image that can be a link and parts of the image that serve as links which lead to defined area. Furthermore, this thesis will show the joining of different links with different areas of one image with the use of image maps. This is possible with defining different areas accessible on the image. In order to achieve dynamic on the website, it requires the use of transitions. Some of these transitions are: change of opacity, change of position of an object, change of color and change of dimension of an element. Complete experimental work is possible with the use of HTML and CSS language. Image galleries show the complete process of the production of a historical object („early Christian lamp“). The goal of this bachelor thesis is to make a functional image gallery which is intended for the website and which will contribute in recognition and revivification of the replica of the earlier mentioned historical object, souvenir of the city Sisak.

Keywords: image introduction, image gallery, image maps, transitions.

## SADRŽAJ

1. UVOD.....	1
2. TEORIJSKI DIO.....	2
2.1. HTML .....	2
2.1.1. Osnovna struktura HTML-a .....	3
2.1.2. Pravila HTML označavanja.....	3
2.1.3. Uvođenje multimedijских elemenata u HTML.....	5
2.2. SLIKE NA WEBU .....	8
2.2.1. Označavanje slika u web dokumentu .....	8
2.2.2. Najpoznatiji formati slika za web.....	9
2.2.3. Slika kao link.....	10
2.3. CSS.....	11
2.3.1. CSS sintaksa .....	11
3. EKSPERIMENTALNI DIO .....	14
3.1. Primjena slikovne mape .....	14
3.2. CSS animacije primijenjene u web galeriji .....	21
4. REZULTATI I RASPRAVA.....	29
5. ZAKLJUČAK.....	35
6. LITERATURA .....	36

## 1. UVOD

Multimedijski elementi na web stranici imaju veliku važnost. Jedan od elemenata je slika. Slika prenosi informacije korisnicima. Općenito, slike poboljšavaju izgled web stranice ukoliko se pravilno koriste. Uz korištenje cijele slike koja može predstavljati link, slici se mogu pridruživati različiti linkovi na različita područja jedne slike [1]. Takav način rada poznat je pod imenom slikovne mape. Slikovne mape pružaju jednostavan način povezivanja raznih dijelova slike, bez da se slika dijeli u odvojene slikovne datoteke. Također, moguće je definirati različite oblike područja na koje se može kliknuti. Takve površine mogu biti su kružne, pravokutne i poligonalne. Prilikom prikaza slike u pregledniku, označena područja su nevidljiva. Drugi važan multimedijski element jest animacija. Animacijom se postiže dinamika na web stranici. CSS animacije omogućuju prijelaz jednog CSS svojstva u drugo CSS svojstvo. Primjer animacije mogu biti: promjena boje, promjena oblika, promjena pozicije, prozirnost. Prikaz promjene animacije definira se na različite načine koje uključuju kontrolu brzine prijelaza nekog svojstva u određenoj fazi. Također, animaciji je moguće definirati završno stanje. Faze animacije mogu koristiti postotke (%) ili 'from' i 'to' *keyframeove* [2]. Cilj ovog rada jest HTML i CSS tehnologijom izraditi funkcionalnu web galeriju koja prikazuje rekonstrukciju svjetiljke.

## 2. TEORIJSKI DIO

### 2.1. HTML

HyperText Markup Language ili skraćeno HTML, jezik je za označavanje hipertekstualnih dokumenata. Oni se od običnih dokumenata razlikuju po tome što sadrže hiperveze (hiperlinkove) kojima su povezani s drugim (hipertekstualnim) dokumentima [3]. Svaku web stranicu koja sadrži barem jednu hipervezu ubrajamo u skupinu hipertekstualnih dokumenata. Hipertekst ili *web*-dokument je tekst koji sadrži veze (*linkove*) ka drugim dokumentima ili na samog sebe. Hipertekst je skup stranica, u obliku datoteka koje su međusobno povezane. Obično se vide kao veze (hiperveze) na koje se može *kliknuti*. Za razliku od običnog teksta, koji se čita linearno (s lijeva na desno, odozgora prema dolje), hipertekst se čita prateći hiperveze u tekstu, dakle, ne nužno na linearan način.

Izvorne datoteke s dokumentom opisanim u HTML-u imaju ekstenziju *.html* ili *.htm*, a nalaze se u određenom kazalu servera vezanog na Internet, što ih čini dostupnim (vidljivim) na *web*-u. Preglednik (*browser*) je tumač jezika za prikazivanje hipertekstualnih dokumenata. On omogućava njihov vizualni prikaz na ekranu. Najpoznatiji preglednici su: Mozilla Firefox, Internet Explorer, Safari, Google Chrome. Preglednici ne prikazuju HTML *tagove*, nego ih koriste za određivanje načina prikaza sadržaja HTML stranice. Kada preglednik učita HTML dokument, preglednik tumači sve oznake koje okružuju tekst.

Oznake (*tags*) su samo riječi ili slova smještena unutar šiljastih zagrada, kao na primjer `<head>`, `<p>`, `<h1>`, itd. Oznake daju informaciju pregledniku o strukturi i značenju teksta. HTML koristi oznake koje daju informaciju pregledniku koji tekst je u naslovu, koji dio teksta je odlomak, koji dio teksta mora biti naglašen, ili čak na kojoj poziciji se slike moraju prikazivati. HTML omogućuje označavanje teksta s oznakama (*tag*) koje govore pregledniku o strukturi teksta. Preglednici ignoriraju razmake u HTML dokumentu.

Što se tiče koordinatnog sustava HTML dokumenta, ishodište se nalazi u gornjoj lijevoj točki te su mjerne jedinice koje se koriste pri kodiranju pikseli. Kada govorimo o pozicioniranju, postoje dva načina, a to su: apsolutno i relativno pozicioniranje.

Apsolutno pozicioniranje podrazumijeva definiranje početne, gornje i lijeve točke nekog elementa. Pri ovom načinu pozicioniranja, navodimo točne koordinate u odnosu na ishodište. Za razliku od apsolutnog, relativno pozicioniranje podrazumijeva smještanje elemenata jedan ispod drugog ili jedan pokraj drugog uz korištenje atributa "*float:left*". Elementi se relativno pozicioniraju u odnosu na svoju prvobitnu poziciju.

### **2.1.1. Osnovna struktura HTML-a**

HTML koristi elemente kako bi opisao strukturu stranice, kao npr. naslovi, podnaslovi, odlomci. Osnovni kod HTML dokumenta mora sadržavati <html>, <head>, <title> i <body> elemente. Ovi elementi dolaze u paru što znači da početni element označava početak sadržaja, dok završni element označava kraj sadržaja.

```
<html>
<head>
<title> </title>
</head>
<body>
</body>
</html>
```

Početni *tag* HTML-a <html> , govori pregledniku da je sadržaj datoteke HTML kod, dok završni *tag* HTML-a </html>, daje informaciju pregledniku da je kraj HTML koda. <head> element označava zaglavlje dokumenta i sadržaj zaglavlja nije vidljiv u pregledniku. Unutar <head> elementa, obično se nalazi <title> element koji nema atributa već određeni tekst koji se definira kao naslov dokumenta prikazan na naslovnoj liniji preglednika. Sve što se prikazuje u glavnom prozoru preglednika, nalazi se između početnog i završnog <body> *taga* [4].

### **2.1.2. Pravila HTML označavanja**

Potrebno je pridržavati se pravila pisanja koda kako bi HTML kod bio podržan na različitim preglednicima.

HTML kôd sastoji se od oznaka (*tag*) i njihovih atributa, a piše se po pravilu:



```
<oznaka atribut="vrijednost">  
Tekst koji se prikazuje na stranici.  
</oznaka>
```

Oznake i atributi imaju unaprijed zadana imena koja se moraju naučiti. Na primjer, oznaka koja opisuje font kojim je pisan tekst nosi ime `<font>`, a atribut kojim definiramo boju tog fonta nosi naziv *color*. U web pregledniku prikazuje se samo tekst koji se nalazi unutar početnog i završnog dijela oznake `<font>`, dok sam kôd koji opisuje izgled teksta neće biti prikazan.

Sve što se nalazi unutar neke oznake poprima njezina svojstva. Većina oznaka ima svoj početni dio, npr. `<i>`, i završni dio `</i>`. Ova će oznaka, primjerice, ukositi sav tekst koji ona omeđuje (*italic*). U završnom dijelu oznake ispred naziva oznake stoji znak `" / "`.

Također, nije bitno piše li se kôd velikim ili malim slovima, no preporučljivo je pisati sve malim slovima. Kôd cijele stranice može se pisati i u jednom jedinom retku, nižući oznake jednu iza druge. No, bolje je pisati ih tako da kôd bude što pregledniji.

Vrijednosti atributa se uvijek stavljaju pod navodnike, npr. `color="red"`. Ako neka oznaka sadrži više atributa, nije važno kojim redoslijedom su oni napisani. Atributi se uvijek odvajaju razmakom. Primjerice,

```
<font size="2" color="red">
```

Oznake omeđuju sve elemente našeg dokumenta, pa tako i jedna drugu. Ako u kodu imamo dvije ili više oznaka, jedna se mora nalaziti unutar druge, moraju biti ugniježdene. Na primjer:

```
<i>Ovaj tekst je ukošen, a ovaj <u>i ukošen i podvučen.</u></i>
```

Što se tiče komentara, oni se navode pod određenim pravilom koje glasi `"<!--sadržaj komentara-->"`. Ako je komentar napisan po tom pravilu, preglednik ga u potpunosti zanemaruje i ne prikazuje na web stranici.

### 2.1.3. Uvođenje multimedijских elemenata u HTML

Multimedijске elemente čine: tekst, slika, zvuk, animacija, video.

#### *Tekst*

Tekst na web stranicama je vrlo važan multimedijски element. Postoji nekoliko načina uvođenja teksta. Najjednostavniji način je pomoću *paragraf taga* <p>. U takvom slučaju tekst se unosi između otvarajućeg <p> i zatvarajućeg *taga* </p>, dok se njegov stil definira u CSS-u. Drugi način jest preko div elementa. Ovakav način je poželjan ukoliko unutar HTML dokumenta postoji više tekstova kojima se određena svojstva razlikuju.

Primjer:

```
div
{
    position: absolute;
    width: 100px;
    height: 100px;
    border: 5px solid black;
    font-weight: bold;
    text-align: center;
    box-sizing: border-box;
}

.r1 {
    top: 0px;
    font-family: Calibri;
    font-size: 10px;
    padding-top: 35px;
}

.r2 {
    top: 110px;
    font-family: Times;
    font-size: 20px;
    padding-top: 30px; }
```

## ***Slika***

Slike imaju veliki značaj na webu, odnosno njihova funkcija je velika. Slike mogu prenositi razne informacije te mogu poboljšati izgled stranice. Slike pozivamo pomoću `<img>` taga koji sadrži atribut *src* (*source*) [5]. Atribut *src* sadrži ime slike koju je potrebno naći u dokumentu, odnosno putanju do te slike. Sintaksa `<img>` taga je sljedeća:

```

```

## ***Zvuk***

Zvuk je također bitan multimedijски element. Zvukom se prenosi određena informacija, ali na zanimljiviji način. Važno je spomenuti da se zvuk iz analogne informacije mora pretvoriti u digitalnu (binarnu) informaciju. Neki od najpoznatijih formata su: WAV, MP3 i WMA. Zvuk se može uvesti na različite načine u HTML dokument. Neki od njih su preko `<audio>` taga, `<a>` taga, `<object>` taga te `<embed>` taga.

Preko `<audio>` taga, zvuk se uvodi:

```
<audio src="audio1.format" type="audio/format" controls>  
</audio>
```

Primjer uvođenja preko `<a>` taga:

```
<a href="audio2.format"> Uvođenje zvuka uz pomoć a taga </a>
```

Preko `<object>` taga, uvođenje zvuka:

```
<object width="x" height="y" data="zvuk.format"> </object>
```

Te posljednji primjer uvođenja preko `<embed>` taga:

```
<embed width="x" height="y" src="zvuk.format">
```

## ***Animacija***

CSS animacije pružaju mogućnost animiranja prijelaza iz jednog CSS svojstva u drugo CSS svojstvo. Primjer animacije koje mogu biti: promjena boje, promjena oblika, promjena pozicije, prozirnost. Animacija se sastoji od dvije komponente, a to su: stil koji opisuje CSS animaciju i niz okvira (*keyframes*) koji određuju početno, međustanja i konačno stanje animacije. Svaka animacija mora imati ime (*animation-name*) i vrijeme

trajanja jednog ciklusa animacije (*animation-duration*). Osim tih, animacija može sadržavati i dodatna svojstva.

### ***Video***

Video ima mogućnost prenošenja najviše informacija u usporedbi s ostalim multimedijским elementima. Najčešći formati videa su: MP4, AVI, MPEG, WMW, FLV. Video se može uvesti preko različitih *tagova*, a to su `<video>` *tag*, `<object>` *tag* te `<embed>` *tag*.

Uvođenje preko `<video>` *taga*:

```
<video src="video1.format" type="video/format" controls>  
</video>
```

Uvođenje preko `<object>` *taga*:

```
<object width="x" height="y" data="video.format"> </object>
```

Uvođenje preko `<embed>` *taga*:

```
<embed width="x" height="y" src="video.format">
```

## 2.2. SLIKE NA WEBU

Slike imaju veliki značaj na webu. One prenose važne informacije korisnicima, te općenito poboljšavaju izgled stranice. Ako se slike ispravno i pažljivo koriste, one web stranici daju vizualno bolji izgled, no ako se koriste neispravno tada mogu stranicu učiniti nečitljivom.

### 2.2.1. Označavanje slika u web dokumentu

Preglednici obrađuju `<img>` elemente drugačije od ostalih elemenata. Kada preglednik vidi element poput `<h1>`, `<p>`; sve što treba učiniti jest prikazati ih. No, kada preglednik vidi `<img>` element, događa se nešto drugačije. Preglednik treba dohvatiti sliku prije nego što može biti prikazana na web stranici. Dakle, preglednik dohvati određenu HTML datoteku sa servera koju onda učita i prikaže. Ukoliko postoji slika, preglednik je preuzima sa web servera.

Element `<img>` posjeduje atribut *src* (*source*) te on sadrži ime slike koju je potrebno naći u dokumentu, odnosno putanju do te slike. Ukoliko se slika nalazi u istoj mapi kao i sam HTML dokument, sintaksa `<img>` elementa je sljedeća:

```

```

Za određivanje dimenzija slike mogu se koristiti atributi *height* i *width* s `<img>` elementom kako bi preglednik imao informaciju kolika je visina i širina slike. No, prije uvođenja slike na web, potrebno je podesiti rezoluciju za web koja iznosi 72 px/inch, te dimenzije slike, dakle, visinu i širinu slike.

Mjerna jedinica je piksel. Kada se određuju visina i širina slike koja se razlikuje od stvarne dimenzije slike, preglednik ih prilagodi na definirane dimenzije bez obzira na deformaciju slike. Kod ovakvog prikazivanja slike mora se paziti na proporcionalnost kako slika ne bi postala deformirana ili izgubila na kvaliteti.

Definiranje dimenzija *inline*, odnosno unutar samog HTML dokumenta:

```

```

Definiranje dimenzija *interno*, odnosno CSS-om:

```
<html>
<head>
<style>
img{
height:200px;
width:300px;}
</style>
</head>
<body>

</body>
</html>
```

### 2.2.2. Najpoznatiji formati slika za web

Slike se mogu spremiti i kreirati u mnogo različitih formata, ali svega nekoliko formata je prikladno koristiti za web i koje su kompatibilne sa svim operacijskim sustavima i računalima.

JPEG- Najbolje radi za kontinuirani ton slike, kao što su fotografije. Mogućnost prikaza slike i sa 16 milijuna različitih boja, podržavaju 24 bitne boje. Nedostatak je taj što ne podržava transparentnost i animacije.

PNG- Najbolje radi sa slikama koje sadrže samo nekoliko boja, primjer su logotipi, slike s malim tekstom u sebi. Pri mijenjanju veličine slike, ne dolazi do gubitka vrijednosti i detalja. Mogućnost transparentnosti svih boja te ostvaruje dobar prikaz poput JPEG formata. PNG format slike podržava 24 bitne i 32 bitne boje.

GIF- Najbolje radi sa slikama koje sadrže mali broj boja, dobar primjer su slike s malim tekstom u sebi, logotipi. Nedostatak je prikaz do 256 različitih boja. Poput PNG formata, prednost je ta što pri mijenjanju veličine ne dolazi do gubitka informacija i detalja. Transparentna boja može biti samo jedna. Podržava animaciju.

WEBP- Format slike koji pruža jednaku kvalitetu kao JPEG, ali sa 40% manjom veličinom datoteke. WebP format je predložen za animirane slike kao alternativa za GIF

format. WebP format također nudi i opciju transparentije, npr. izbijanje pozadine. U usporedbi s JPEG slikama, WebP slike imaju blaže prijelaze rubnih vrijednosti [6].

### 2.2.3. Slika kao link

Slike se mogu koristiti kao linkovi koji su povezani s drugim objektima ili drugim web stranicama. Slika kao link se obično koristi kada se želi prikazati više slika u istom prozoru preglednika. Dakle, ima smisla kreirati više malih slika koje vode na iste te slike samo većih dimenzija. Na takav način se mogu postaviti umanjene slike na web stranicu kako bi posjetitelji lakše uočili sadržaj.

Kako bi se stvorila veza sa slikom, koristi se `<img>` element koji je ugniježđen unutar `<a>` elementa [7]. Element `<a>` sadrži `href` atribut. U `href` atribut postavlja se link željenog odredišta. Gdje je moguće, tekst linka trebao bi objasniti gdje je odredište ukoliko se *klikne* na njega.

```
<a href="velika.jpg">  
  
</a>
```

## 2.3. CSS

*CascadingStyleSheets* ili skraćeno CSS, stilski je jezik, služi za uređivanje izgleda HTML elemenata. S CSS-om poboljšava se izgled i funkcionalnost web stranica . CSS dokument je najobičniji tekstualni dokument koji sadrži naredbe za web preglednike, koji definiraju stilove za određene HTML elemente na web stranici [8].

U početku, HTML je definirao kompletan izgled, strukturu i sadržaj web stranice. Uz dodatne *tagove* npr. `<font>` , dolazi do poteškoća jer HTML dokumenti postaju preveliki i teško se snalaziti u njima. CSS ima veće dizajnerske mogućnosti što znači da definira izgled web stranice uključujući boju, font, pozadinu, margine. CSS donosi puno prednosti, kao što su: brzina, jednostavnost, štednja prostora na hostu te se na takav način sigurnije održavaju HTML i CSS dokumenti.

### 2.3.1. CSS sintaksa

Osnovni element CSS-a je *selector* čija funkcija je definiranje izgleda svih elemenata na koje je primijenjen. Selector može biti: HTML *tag* (element), klasa HTML-a i identifikacijska vrijednost HTML elementa na koji će se opis stila odnositi.

Svaki HTML element ima odgovarajući *selector*, primjerice `<body>` (tijelo dokumenta), `<p>` (odlomak). Kada stil definiramo preko HTML taga, taj opis utječe na sve elemente u dokumentu koji imaju ovaj *tag*. Ime im je jednako HTML oznaci, ne uključujući šiljaste zagrade. Definiranje elemenata se vrši navođenjem naziva HTML elementa kojeg se želi selektirati (npr. `div` ):

```
div { atribut: vrijednost; }
```

Ako je *selector* klasa HTML-a tada se isti stil primjenjuje na više elemenata iste klase. *Class* selektori imenuju se tako da se stavi točka ispred imena.

```
.imeKlase { atribut: vrijednost; }
```

ID selektor zahtjeva da element koji odabiremo sadrži atribut *id* s nazivom



identifikatora. Vrijednost atributa *id* ne može dijeliti više HTML elemenata, već isključivo jedan element po stranici, čime je omogućeno korištenje *id* selektora kako bi se stilizacija primijenila na jedinstven element. *ID selector* zahtjeva da se ispred naziva identifikatora nalaze ljestve ( # ) nakon kojih slijedi naziv identifikatora.

```
#IDvrijednost { atribut: vrijednost; }
```

Također, element CSS-a je i atribut koji opisuje odgovarajuća svojstva elemenata kao što su: boja, okvir, font, visina, širina. Atributi se navode unutar vitičastih zagrada sa znakom dvotočke nakon naziva, dok iza naziva se poziva vrijednost. Pod „vrijednost“ spadaju podaci koji određuju izgled stranice npr. određena boja, definirano poravnavanje teksta, debljina okvira.

```
selector { atribut: vrijednost; }
```

CSS stilovi mogu biti interni, eksterni ili inline. Prioritet ima inline stil. Ako nema inline stila tada prioritet ima interni stil, osim ako je link na eksterni stil postavljen u <head> elementu nakon internog stila kao što prikazuje sljedeći kod:

```
<html>
<head>
<style>
h1 {
font-family:Arial; font-size:12px; color:red; }
</style>
<link rel="stylesheet" type="text/css" href="h1.css">
</head>
<body>
<h1> Naslov </h1>
</body>
</html>
```

Interni način se koristi kada pojedinačna stranica ima jedinstven stil, tj. razlikuje se od ostalih stranica. Interni stil se definira u zaglavlju (<head> element) unutar <style> taga. Nedostatak ovakvog načina je taj što se u svakom HTML dokumentu moraju navoditi svi stilovi.

```
<style type="text/css">
p { color: red;
font-size: 12pt;
background-color: green; }
</style>
```

Eksterni način je upisivanje stilova u jedan vanjski dokument s ekstenzijom .css. Ti stilovi se odnose na svaku HTML stranicu koja poziva .css dokument. Pozivanje .css dokumenta se radi tako da u HTML dokumentu, u zaglavlju (<head>) upišemo:

```
<link rel="stylesheet" type="text/css" href="h1.css" />
```

U atributu *href* upisuje se lokacija do .css dokumenta.

Treći način za unos CSS stilova jest *inline*. Ovakav stil se odnosi samo na jedan element koji se razlikuje po dizajnu od ostalih istoimenih tagova. Koristi se *style* atribut odgovarajućeg HTML elementa. *Style* atribut sadrži svojstva i vrijednosti i navode se pod navodnim znakovima. *Inline* stil ima najviši prioritet kod primjene. Nedostatak ovakvog pisanja koda je što uz svaki element moraju se navoditi sva svojstva i vrijednosti. Npr.:

```
<p style="color:yellow; font-style:italic;">
Ovo je odlomak
</p>
```

Ako nigdje nije definiran stil za određeni tag, preglednik (Chrome, Firefox, Opera, Internet Explorer) će primijeniti podrazumijevani stil za taj tag. Svaki preglednik ima podrazumijevani stil za bilo koji tag. Tako će preglednik, ako se definira kako neki tekst treba izgledati, ispisati taj tekst crnom bojom. U većini preglednika tekstualni link će biti ispisan plavom bojom i bit će podvučen, ako se kroz CSS ne definira drugačije.

### 3. EKSPERIMENTALNI DIO

#### 3.1. Primjena slikovne mape

Slikovne mape podrazumijevaju pridruživanje različitih linkova na različita područja jedne slike [9]. To se realizira kroz podjelu slike na oblike, odnosno određivanje područja koje će se moći pritisnuti mišem. Nakon što su različiti linkovi vezani za različita područja, moguće je kliknuti pojedine dijelove slike kako bi se otvorio željeni element ili dokument. Takav način rada je poznat pod imenom slikovne mape. Slikovne mape pružaju jednostavan način povezivanja raznih dijelova slike, bez da se slika dijeli u odvojene slikovne datoteke. Definirana područja nisu vidljiva na prikazanoj slici u pregledniku. Također, na jednoj slici je moguće koristiti više slikovnih mapa.

Slikovne mape se sastoje od dva dijela. Prvi dio je uvođenje stvarne slike u HTML dokument preko `<img>` elementa. Taj element mora imati atribut *usemap*. *Usemap* atribut imenuje slikovnu mapu koja se koristi za određenu sliku. Ako atribut *usemap* počinje s ljestvama (`#`), podrazumijeva se da se mapa nalazi u istom HTML dokumentu kao i `<img>` element.

```

```

*Usemap* atribut povezan je s `<map>` elementom i tako stvara odnos između objekta i mape.

Drugi dio je `<map>` element. Mapa se opisuje pomoću otvarajućeg elementa `<map>` i zatvarajućeg elementa `</map>`. Važno je da naziv mape (*map name*) odgovara nazivu za upotrebu mape unutar atributa *usemap* `<img>` elementa.

```
<map name="Map">
```

Unutar otvarajućeg i zatvarajućeg elementa `<map>`, nalazi se `<area>` element. Njegova funkcija je definiranje jednog područja koje se može kliknuti unutar slikovne mape.

Atributi `<area>` elementa su: *shape*, *href* i *coords*.

```
<map name="Map">
```

```
<area shape="circle" coords="60,190,30" href="rep.html" alt="Rep" target="iframe1"/>
```

```
<area shape="rect" coords="125,200,200,260" href="podnozje.html"
alt="Podnožje" target="iframe1"/>

<area shape="circle" coords="155,200,30" href="zdjela.html"
alt="Zdjela" target="iframe1"/>

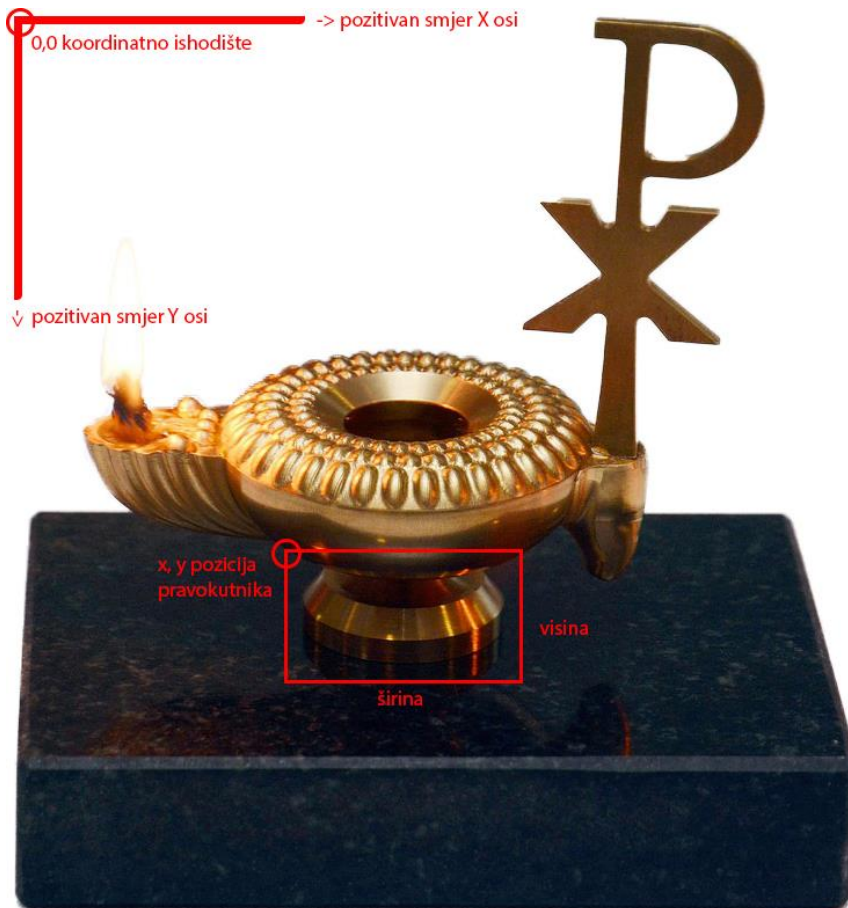
<area shape="circle" coords="255,170,40" href="glava.html"
alt="Glava" target="iframe1"/>

<area shape="circle" coords="265,100,30" href="xp.html" alt="XP"
target="iframe1"/>

<area shape="circle" coords="155,170,30" href="poklopac.html"
alt="Poklopac" target="iframe1"/>

</map>
```

Atribut *shape* određuje oblik površine koja se koristi. Taj atribut se upotrebljava zajedno s koordinatama (*coords*) kako bi se odredila veličina, oblik i položaj područja. Oblik može biti u obliku pravokutnika (*shape="rect"*), poligona (*shape="poly"*) ili kruga (*shape="circle"*). Svaki od navedenih oblika sadrži x i y koordinatne vrijednosti, a za oblik kruga se dodaje i vrijednost radijusa (r). Zadane vrijednosti se odvajaju zarezom. Koordinatno ishodište mape započinje u gornjem lijevom kutu slike, a x i y vrijednosti predstavljaju udaljenost u pikselima (px), u odnosu na koordinatno ishodište kao što je prikazano na slici 1.



Slika 1. Primjer koordinatnog sustava sa zadanim koordinatama pravokutne slikovne mape

Ako je željeno područje na slici u obliku pravokutnika tada se određuje udaljenost od gornjeg lijevo i donjeg desnog kuta pravokutnika u pikselima. U izrađenoj web galeriji, ovakav oblik je korišten za jedno područje na slici, prikaz na slici 1.

```
<area shape="rect" coords="125,200,200,260" href="podnozje.html" alt="Podnožje" target="iframe1"/>
```

Za kružno područje na slici, određuje se vrijednost centra kružnice, te radijus kružnice u odnosu na ishodište. Na ovakav način, u radu je definirano pet područja slike, primjer na slici 2.

```
<area shape="circle" coords="60,190,30" href="rep.html" alt="Rep" target="iframe1"/>
```

```
<area shape="circle" coords="155,200,30" href="zdjela.html" alt="Zdjela" target="iframe1"/>
```

```
<area shape="circle" coords="255,170,40" href="glava.html"
alt="Glava" target="iframe1"/>
```

```
<area shape="circle" coords="265,100,30" href="xp.html" alt="XP"
target="iframe1"/>
```

```
<area shape="circle" coords="155,170,30" href="poklopac.html"
alt="Poklopac" target="iframe1"/>
```

Kada bi područja bila vidljiva, izgledalo bi ovako:



Slika 2. Prikaz slikovne mape, kružni oblik

Ako je željeno područje na slici u obliku poligona tada se koordinatne vrijednosti povezuju redom, dok se posljednji par vrijednosti spaja s prvim. Ako bismo željeli stvoriti poligonalnu slikovnu mapu koristili bismo sljedeći kod:

```
<area shape="poly" coords="x1,y1,x2,y2,x3,y3..." href="x.html"
alt="tekst">
```

Atribut *href* određuje koji će se HTML dokument otvoriti pritiskom na određeno područje. U ovom radu, napravljeno je šest zasebnih HTML dokumenata za šest galerija slika. Dakle, za pojedini dio slike, odnosno pojedini dio svjetiljke, izrađen je HTML dokument koji sadrži slike i tekstualne informacije o navedenom objektu.

Atribut *alt* određuje alternativni tekst za područja ako se, iz nekog razloga, ne mogu prikazati. Razlog tome može biti, primjerice, spora veza ili pogreška u *src* atributu.

Atribut *target* određuje gdje će se otvoriti dokument. U izrađenoj web galeriji, *target* zasebnih HTML dokumenata je *iframe* pod nazivom "iframe1".

Općenito, *iframe* predstavlja točno definirano područje koje je postavljeno na web stranici. Može služiti kao ciljano odredište (*target*) gdje će se prikazivati željeni sadržaj. Odredište (*target*) linka mora sadržavati isti naziv kao što je zadano u *iframe* elementu. Npr.

```
<iframe name="iframe1" class="iframe1" scrolling="no"> </iframe>  
  
<area shape="circle" coords="155,170,30" href="poklopac.html"  
alt="Poklopac" target="iframe1"/>
```

Pri izradi web galerije, *iframe* je korišten za ugrađivanje drugog HTML dokumenta unutar trenutnog HTML dokumenta. Na taj način doprinijelo se kontinuiranom radu web galerije, bez potrebe da se u preglednik ponovno učitava cijela stranica. Element "*iframe1*" pozicioniran je apsolutno, te su njegove informacije definirane CSS-om, unutar <style> elementa. Pozicija "*iframe1*" je određena pomoću naredbi *top* i *left*, te su definirane dimenzije *iframe*-a preko naredbi *width* (širina) i *height* (visina). Navedenom *iframe*-u je dodana zaobljenost rubova (*border-radius*) kako bi se povećala usklađenost s postoljem svjetiljke. Da bi se *iframe* prikazao u pregledniku, potrebno ga je ispisati u <body> element.

```
.iframe1{  
position:absolute;  
top:150px;  
left:600px;  
width:450px;  
height:467px;  
opacity:0;  
border-radius:7px;  
}
```

```
<iframe name="iframe1" class="iframe1" scrolling="no"> </iframe>
```

Za potrebe web galerije, izrađeno je 6 zasebnih HTML dokumenata, po istom principu, od kojih svaki sadrži tri slike različitog dijela svjetiljke. Otvaranjem HTML dokumenta u "iframe1" elementu, prikazuju se tri uzastopne umanjene slike, opaciteta 0.5.

Pritiskom miša na umanjenu sliku koja ima funkciju *linka*, otvara se uvećana slika u definiranom okviru nazvanog "veliki". Uvećana slika ima vrijednost opaciteta 1, dakle potpuno je vidljiva.

Primjer uvođenja slike kao link:

```
<a href="1a.jpg" target="veliki">  
  
</a>
```

Definiranje CSS-om te uvođenje, u <body> element, okvira pod nazivom "veliki":

```
iframe{  
position:absolute;  
border-radius:7px;  
border:none;}
```



```
.veliki{  
position:absolute;  
top:190px;  
left:30px;  
width:390px;  
height:245px;  
}
```

```
<iframe name="veliki" class="veliki"> </iframe>
```

U svakoj galeriji također je prisutan tekst definiran CSS-om i ispisan u <body> element na način:

```
<div class="tekst">  
<p> Dio svjetiljke koji predstavlja otvor za stijenj (fitilj).  
</p>  
</div>
```

### 3.2. CSS animacije primijenjene u web galeriji

CSS animacije pružaju mogućnost animiranja prijelaza iz jednog CSS svojstva u drugo CSS svojstvo. Potrebno je definirati putanju animacije koja se odnosi na skup točaka, kod kojeg svaka točka ima svoja CSS svojstva. Između svake dvije susjedne točke različitih svojstava događa se prijelaz iz jednog u drugo svojstvo, te se tako stvaraju animacije. Neki od primjera animacija koje se koriste: promjena boje, oblika, pozicije te prozirnost.

Animacija se sastoji od dvije komponente, a to su: stil koji opisuje CSS animaciju i niz okvira (*keyframes*) koji određuju početno, međustanja i konačno stanje animacije. Svaka animacija mora imati ime (*animation-name*) i vrijeme trajanja jednog ciklusa animacije (*animation-duration*). Osim tih, animacija može sadržavati i dodatna svojstva.

Prikaz promjene svojstva animacije može se definirati na različite načine, pa tako razlikujemo: *ease-in*, *ease-out*, *ease-in-out* i *linear*. *Ease-in* označava sporiji prijelaz na početku, dok pri završetku ubrzava. *Ease-out* obrnuto od *ease-in*, dakle prijelaz je na početku brži te pri kraju usporava. *Ease-in-out* je prijelaz nekog svojstva koji ima postepeno usporavanje na početku i na kraju. *Linear* je linearni prijelaz svojstva što znači da je prijelaz jednak tijekom odvijanja animacije.

Uz prikaz promjene svojstva animacije, moguće je definirati i završno stanje animacije. Tako razlikujemo: *infinite*, *alternate*, *forwards* i *animation-delay* stanje. *Infinite* svojstvo daje naredbu da se animacija ponavlja beskonačno mnogo puta. Svojstvom *alternate* definiramo animaciju koja kada dođe do završnog stanja, ponavlja se unatrag. Svojstvo pri kojem animacija zadržava završno stanje jest *forwards*. Te posljednje, *animation-delay* svojstvo kojim se može kontrolirati početak animacije, dakle, moguće je zadati vrijeme nakon kojeg se animacija pokreće.

Svaki korak animacije moguće je kontrolirati putem *keyframeova*. *Keyframeovi* se sastoje od riječi *@keyframes*, imena animacije te niza pravila u vitičastim zagradama koja definiraju animaciju. Tako se definira početna, međufaza i završna faza animacije. Faze animacije mogu koristiti postotke (%), te se na taj način određuje vrijeme kada se

određeni dio animacije prikazuje. Također, uz postotke postoji još jedan način definiranja faza animacije, a to je sa 'from' i 'to'. U tom slučaju, animacija ima samo početno i završno stanje.

U ovom radu prikazuje se uvođenje šest animacija. Prva animacija koja se odvija jest promjena pozadinske boje. Kako bi se boja prikazala preko cijelog zaslona računala, definirane su širina i visina koje zauzimaju 100% površinu zaslona. Boja je ispisana heksadecimalnom metodom. Heksadecimalne boje imaju znak ljestve (#), nakon kojeg dolazi 6 ili 3 heksadecimalnih znamenki. Prve dvije znamenke se odnose na vrijednost crvene boje, druge dvije na vrijednost zelene, a zadnje dvije na vrijednost plave boje.

Prvi korak pri ispisivanju animacije jest davanje naziva animaciji, koji je u ovom slučaju "anim1". Važno je da se zadana animacija postavi unutar elementa kojeg želimo animirati.

Prijelaz pozadinske boje iz jedne u drugu podijeljen je u dvije faze. Takav prijelaz postigao se ispisivanjem početne boje (#be8b8b) u 'from' dio *keyframe*-a nakon kojeg se u vitičastim zagradama ispisuje svojstvo koje se želi promijeniti. Zatim se u 'to' dio *keyframe*-a ispisala boja (#C7AF AF) koja se prikazuje u završnoj fazi animacije.

Animacija pozadinske boje započinje otvaranjem HTML dokumenta u pregledniku. Prijelaz pozadinske boje traje 5s. Animacija se odvija linearno što znači da je prijelaz jednak tijekom animacije. Pozadinska boja zadržava svoje završno stanje, odnosno boju koja je zadana.

```
.pozadina{  
position: absolute;  
width: 100%;  
height: 100%;  
background: #be8b8b;  
animation: anim1 5s forwards 0s linear;}  
  
@keyframes anim1{  
from{background: #be8b8b;}  
to{background: #C7AF AF;}}
```

Za prikaz u pregledniku, ispisan je div element klase "pozadina" u <body> elementu.

```
<div class="pozadina">  
</div>
```

Druga animacija jest animacija glavne slike, svjetiljke. Prvi korak je definiranje dimenzija slike (500 x 500 px) te rezolucije koja iznosi 72 px/inch u Adobe Photoshopu. Nakon toga, potrebno je uvesti sliku u <body> element kako bi se slika prikazala u pregledniku.

```

```

U <style> elementu definiraju se informacije o željenoj slici. Slika je pozicionirana apsolutno, dakle, pozicija se računa s obzirom na točku ishodišta, gornji lijevi kut. Slika je smještena pomoću naredbi *top* i *left*.

Naziv animacije glasi "anim2". *Keyframeovi* su definirani s 'from' i 'to'. Unutar njih su zapisana svojstva s vrijednostima koja se mijenjaju. S obzirom na početno stanje mijenja se pozicija na x-osi (*left*), dok po y-osi ostaje iste vrijednosti. Također, mijenjaju se širina i visina slike. U završnoj fazi animacije, slika je umanjena za 150px.

Animacija slike pokreće se s 2s zakašnjenja. Prijelaz pozicije slike traje 4s. Animacija se odvija linearno što znači da je prijelaz jednak tijekom animacije. Slika zadržava svoje završno stanje, odnosno poziciju koja je posljednja zadana.

```
img{  
position:absolute;  
top:150px;  
left:335px;  
width:500px;  
height:500px;  
animation:anim2 4s forwards 2s linear;  
}
```

```
@keyframes anim2{
  from{left:335px;width:500px;height:500px;}
  to{left:150px;width:350px;height:350px;}
}
```

Sljedeća animacija se odnosi na promjenu opaciteta teksta koji se prikazuje. Prvo što je potrebno napraviti jest ispisati u `<p>` element tekst koji se želi prikazati u pregledniku.

```
<div class="iframe2">
  <p class="opis"> Ranokršćanska svjetiljka datira iz 5. stoljeća
  i pronađena je na lokalitetu Siska. <br>Ideja je povezati
  tehnologije izrade iz vremena u kojem je bila izrađena
  svjetiljka i moderne CNC tehnologije našeg vremena. </p>
</div>
```

Tekst se prikazuje u posebnom okviru, nazvanom "iframe2" čije su dimenzije i pozicija definirane u `<body>` elementu. Odlomku je dana klasa pod nazivom "opis" u kojem su ispisana svojstva. Odlomak je pozicioniran apsolutno, definiran je font, poravnavanje, boja, stil i veličina slova.

U ovoj animaciji dolazi do promjene opaciteta. Prijelaz se odvija u tri faze (0%, 50%, 100%). Animacija pod nazivom "anim3" u prve dvije faze mijenja opacitet odlomka iz prozirnog tj. nevidljivog vrijednosti 0 u neprozirno tj. vidljivog vrijednosti 1. U posljednjoj fazi opacitet ponovno poprima vrijednost 0 odnosno tekst postaje nevidljiv.

Animacija opaciteta odlomka pokreće se sa 7s zakašnjenja. Prijelaz opaciteta traje 15s. Animacija se odvija linearno. Opacitet zadržava svoje završno stanje, odnosno potpunu prozirnost odlomka.

```
.opis{
  position:absolute;

  font-family:Avdira;

  text-align:justify;

  color:black;

  font-size:20px;

  opacity:0;
```

```
font-weight:bold;

animation:anim3 15s forwards 7s linear;
}

@keyframes anim3{
0%{opacity:0;}
50%{opacity:1;}
100%{opacity:0;}
}
```

Četvrta animacija se odnosi na promjenu opaciteta i pozadinske boje. Element je apsolutno pozicioniran. Određene su dimenzije i pozicija elementa. Naredbom *border-radius* zaobljuju se rubovi elementa za 7px.

Prijelaz se odvija u dvije faze. U prvoj fazi, elementu je definirana boja koja je ista boja zaslona te se iz tog razloga čini kao da je element proziran. U završnoj fazi, elementu se mijenja početna boja (#be8b8b) u završnu boju (#2a2f35) opaciteta 0.8. Boja je zapisana heksadecimalnom metodom.

Animacija pod nazivom "anim4" pokreće prijelaz opaciteta odlomka sa 10s zakašnjenja. Prijelaz boje traje 2s. Animacija se odvija linearno što znači da je prijelaz jednak tijekom animacije. Opacitet i nova boja zadržavaju svoje završno vidljivo stanje.

```
.okvir{
position:absolute;
width:350px;
height:110px;
top:510px;
left:150px;
opacity:0;
border-radius:7px;
animation:anim4 2s forwards 10s linear;
}
```

```
@keyframes anim4{
  from{background-color:#be8b8b;}
  to{opacity:.8;background-color:#2a2f35;}
}
```

Kod pete animacije, naziva "anim5", dolazi do promjene opaciteta teksta. Tekst je pozicioniran apsolutno, zadane su vrijednosti poput fonta, poravnavanja, boje, veličine te stila teksta. Potrebno je ispisati u <p> element tekst koji se prikazuje u pregledniku. Tekst se prikazuje unutar elementa "okvir".

```
<div class="okvir">
<p> KLIKOM NA DIO SVJETILJKE OTVARA SE GALERIJA SLIKA POJEDINOG
DIJELA </p>
</div>
```

*Keyframeovi* animacije definirani su s 'from' i 'to'. Unutar njih zapisano je svojstvo s vrijednosti koja se mijenja. S obzirom na početno stanje mijenja se opacitet iz prozirnog stanja vrijednosti 0, u neprozirno stanje vrijednosti 1.

Animacija teksta pokreće se sa 9s zakašnjenja. Prijelaz opaciteta teksta traje 2s.

Animacija se odvija linearno. U završnoj fazi animacije, tekst ostaje vidljiv.

```
p{
  position:absolute;
  font-family:Avdira;
  text-align:center;
  color:#C7AFAF;
  font-size:20px;
  opacity:0;
  font-weight:bold;
  animation:anim5 2s forwards 9s linear;
}

@keyframes anim5{
  from{opacity:0;}
  to{opacity:1;}
}
```

Posljednja animacija se odnosi na promjenu opaciteta i boje koju poprima "iframe1". Element je apsolutno pozicioniran. Određene su dimenzije i pozicija elementa. Naredbom *border-radius* zaobljuju se rubovi elementa za 7px.

Prijelaz se odvija u dvije faze. U prvoj fazi, elementu definiran je opacitet vrijednosti 0, te ga to čini nevidljivim prije početka animacije. U završnoj fazi, elementu mijenja se rub debljine 2px, iz pozadinske boje (#be8b8b) u boju okvira (#2a2f35). Boja je zapisana u RGB sustavu heksadecimalno. Okvir postaje vidljiv tek nakon pokretanja animacije.

Animacija pod nazivom "anim6" pokreće prijelaz opaciteta okvira sa 22s zakašnjenja. Prijelaz boje traje 1s. Animacija se odvija linearno što znači da je prijelaz jednak tijekom animacije. Opacitet i boja zadržavaju vrijednosti završne faze.

```
.iframe1{  
position: absolute;  
top: 150px;  
left: 600px;  
width: 450px;  
height: 467px;  
opacity: 0;  
border-radius: 7px;  
animation: anim6 1s forwards 22s linear;  
}  
  
@keyframes anim6{  
from{opacity: 0;}  
to{opacity: .8; border: 2px solid #2a2f35;}  
}
```



Također, potrebno je u <body> elementu ispisati element koji se treba prikazati u pregledniku, dakle u ovom slučaju to je "iframe1".

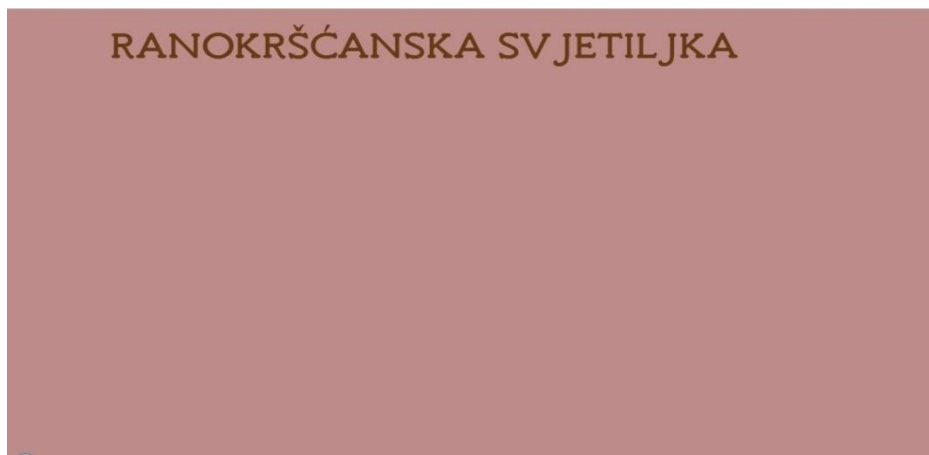
```
<iframe name="iframe1" class="iframe1" scrolling="no">  
</iframe>
```

#### 4. REZULTATI I RASPRAVA

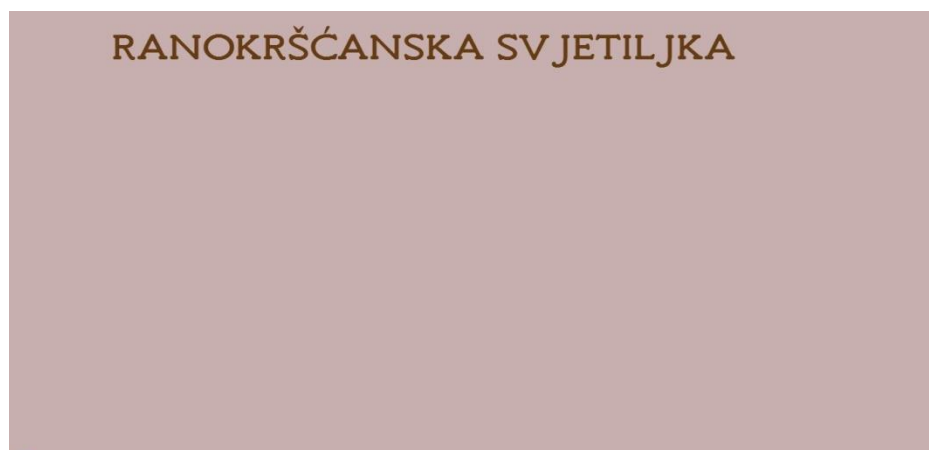
U eksperimentalnom dijelu prikazana je izrada slikovne mape i šest animacija koje su sadržane na web galeriji. Promatrajući rezultate može se zaključiti da su slikovne mape pogodne za tematiku koja se opisuje u ovom radu. HTML i CSS tehnologije pridonijele su u stvaranju web galerije te uz njih, uz pravilno korištenje ne bi trebalo biti poteškoća.

Kod animacija, pratio se prijelaz iz jedne boje u drugu, prijelaz slike s jedne pozicije na drugu poziciju te opacitet koji prelazi iz prozirnog stanja vrijednosti 0 u vidljivo stanje vrijednosti 1.

Uspješnost prijelaza iz jedne pozadinske boje u drugu možemo uočiti na slici 3 i slici 4.



Slika 3. Prikaz pozadinske boje prije početka animacije



Slika 4. Prikaz pozadinske boje nakon završetka animacije

Također, prijelaz glavne slike određene dimenzije koja mijenja svoj položaj i dimenzije u pregledniku, pokazao se uspješnim. Rezultat promjene pozicije i veličine slike vidljiv je na slikama 5. i 6.



Slika 5. Prikaz pozicije svjetiljke prije animacije



Slika 6. Prikaz pozicije svjetiljke nakon završetka animacije

Animacija pod nazivom "anim3" odvija se u 3 faze. Uz definiranje svojstava, animacija funkcionira. Faze animacije prikazane su na slici 7., slici 8. i slici 9. Slike prikazuju promjenu vrijednosti opaciteta koja se mijenja iz 0 u 1 te ponovno u 0.

## RANOKRŠĆANSKA SVJETILJKA



Slika 7. Prikaz opaciteta teksta, vrijednosti 0, prije animacije

## RANOKRŠĆANSKA SVJETILJKA



Ranokršćanska svjetiljka datira iz 5. stoljeća i pronađena je na lokalitetu Siska. Ideja je povezati tehnologije izrade iz vremena u kojem je bila izrađena svjetiljka i moderne CNC tehnologije našeg vremena.

Slika 8. Prikaz opaciteta teksta, vrijednosti 1, u drugoj fazi animacije

## RANOKRŠĆANSKA SVJETILJKA



Slika 9. Prikaz opaciteta teksta, vrijednosti 0, nakon završetka animacije

Animacije pod nazivom "anim4" i "anim5" prikazuju kako se uz definiranje *animation-delay* i opaciteta, animacija prikazuje u točno predviđeno vrijeme te se pokazuje uspješnom. Rezultat promjene opaciteta prikazan je na slikama 10. i 11.



Slika 10. Prikaz *box*-a prije same animacije, opacitet 0



Slika 11. Prikaz *box*-a nakon same animacije, opacitet 1

Animacija pod nazivom "anim6" pokazuje se uspješnom kao u prijašnjim primjerima kojima se mijenja opacitet. Primjer je prikazan na slikama 12. i 13.



Slika 12. Prikaz okvira, opaciteta vrijednosti 0 prije same animacije



Slika 13. Prikaz okvira, opaciteta vrijednosti 1 nakon animacije

Slikovne mape pridonose praktičnosti web galerije. Uz definirano odredište zasebnih HTML dokumenata, sadržanim slikama dijelova svjetiljke, povećala se funkcionalnost.

Pritiskom miša na određeno područje, otvara se HTML dokument u okviru pozicioniranom s desne strane svjetiljke. Svaki HTML dokument galerije slika izrađen je po istom principu.



Slika 14. Pritiskom miša na određeno područje otvara se galerija slika u okviru



Slika 15. Otvaranje uvećane slike u okviru

Slika 14. i slika 15. prikazuju okvir u kojem se otvaraju zasebni HTML dokumenti.

## 5. ZAKLJUČAK

U eksperimentalnom dijelu rada prikazan je postupak uvođenja slika na web stranicu. Izradila se galerija slika pomoću kojih se slike prikazuju na jedinstven način. Prikazan je postupak definiranja mjesta gdje se slike, određene galerije, prikazuju. Cjelokupan eksperimentalni rad izvodio se uz pomoć HTML i CSS jezika. Kroz galerije slika prikazuje se potpun proces izrade povijesnog predmeta ("Ranokršćanska svjetiljka"). Slikovne mape pokazale su se kao dobar odabir za izradu web galerije. Razlog tome je mogućnost određivanja područja koje pritiskom miša vodi na drugi HTML dokument. Korištena su dva oblika površine na koja se može kliknuti, a to su: kružni i pravokutni oblik. Prednost takvog načina rada jest funkcionalnost. Animacijom se povećala dinamika na webu. Animacije koje su korištene u radu su: promjena opaciteta, promjena pozicije određenog objekta, promjena boje, promjena dimenzija elementa. Određivanjem vremenskog perioda, nakon kojeg se nad određenim elementom vrši prijelaz svojstva, postignut je kontinuirani rad na webu. Prema tome, animacije su se također pokazale kao uspješan odabir u izradi web galerije.



## 6. LITERATURA

1. \*\*\*<http://dev.w3.org/html5/spec-preview/image-maps.html>, *Image maps*, rujan 2015.godine
2. Shenoy A., Guarini G., (2013). *HTML5 and CSS3 Transition, Transformation, and Animation*, Packt, Birmingham
3. Robson E., Freeman E., (2012). *Head First HTML and CSS-2nd edition*, O'Reilly Media, Sebastopol
4. Meloni J., Morrison M., (2010). *HTML and CSS in 24 Hours, Eighth Edition*, Sams, Indianapolis
5. Duckett J., (2011). *HTML and CSS - design and build websites*, John Wiley & Sons, Indianapolis
6. Mijić J.,(2014). *Stupanj kompresije i utjecaj na kvalitetu WebP formata slike*, diplomski rad, Grafički fakultet
7. \*\*\* <http://poincare.matf.bg.ac.rs/~jelenagr/op/sk6.htm>, *Uključivanje hiperveza*, kolovoz 2015.godine
8. Volarić T., (2010/2011). *Osnove CSS-a*, vježbe iz kolegija "Računalni praktikum 2", Mostar
9. \*\*\* <http://www.html-5-tutorial.com/map-and-area-elements.htm>, *The map and area elements and image maps*, kolovoz 2015.godine