



UNIVERZITET U  
ISTOČNOM SARAJEVU  
**PEDAGOŠKI FAKULTET**  
BIJELJINA



*SEMINARSKI RAD*  
**GENERISANJE SLIKA U PHP PROGRAMU**

Profesor:  
***Prof. dr Predrag Katanić***

Student:  
***Marko Ivanović***

Bijeljina, 2023.

# SADRŽAJ

UVOD .....	3
GENERISANJE PRAZNE SLIKE.....	4
SMANJIVANJE SLIKE KORISTEĆI RESAMPLING TEHNIKU .....	6
MODIFIKACIJA POSTOJEĆE SLIKE .....	8
KOMBINOVANJE DVIJE SLIKE.....	10
PRIKAZIVANJE SLIKA IZ TRENUTNOG DIREKTORIJUMA.....	12
ZAKLJUČAK .....	14

# UVOD

Generisanje slika je važan aspekt u razvoju web aplikacija i sistemima za obradu slika. PHP je popularan jezik za server-side razvoj koji pruža moćne funkcionalnosti za generisanje, modifikovanje i manipulaciju slikama. U ovom seminarskom radu ćemo istražiti pet primjera generisanja slika u PHP programu. Svaki primjer će demonstrirati različite tehnike za kreiranje i obradu slika koristeći ugrađene funkcije i biblioteke koje su dostupne u PHP-u. Kroz ove primjere, upoznaćemo se sa osnovnim konceptima generisanja slika, promjenom njihovih dimenzija, rotacijom, primjenom filtera i kombinovanjem slika. Saznaćemo kako PHP omogućava programerima da dinamički generišu slike i pružaju ih korisnicima ili ih koriste u različite svrhe kao što su prikaz na web stranicama ili kreiranje vizuelnih elemenata u aplikacijama. Uz pomoć ovih primjera, steći ćemo temeljno razumijevanje generisanja slika u PHP-u i njihove primjene u realnim scenarijima.

Uključeni primjeri su:

1. Prvi primjer generiše praznu sliku crvene boje u PNG formatu.
2. Drugi primjer smanjuje veličinu slike koristeći resampling i prikazuje je kao JPEG.
3. Treći primjer modifikuje postojeću sliku: skalira je, rotira za 90 stepeni i primjenjuje crno-bijeli filter.
4. Četvrti primjer kombinuje dvije slike u jednu kolaž sliku.
5. Peti primjer prikazuje sve PNG, JPG i WEBP slike iz trenutnog direktorijuma na web stranici.

Kroz ove primjere, istražićemo različite aspekte generisanja i obrade slika u PHP-u, pružajući praktične primjere i razumijevanje kako primijeniti ove tehnike u vlastitim projektima.

## GENERISANJE PRAZNE SLIKE

U ovom primjeru (slika 1), koristimo ugrađene funkcije PHP-a za generisanje i prikazivanje jednostavne slike. Prvo definišemo željenu širinu i visinu slike, nakon čega koristimo funkciju **imagecreatetruecolor()** za kreiranje prazne slike sa zadatim dimenzijama. Zatim postavljamo boju pozadine pomoću funkcije **imagecolorallocate()**, koja prihvata RGB vrijednosti boje. U ovom slučaju, postavljamo crvenu boju pozadine. Nakon toga, koristimo funkciju **imagefill()** da popunimo pozadinu slike sa definisanom bojom.

```
<?php
// Definisanje širine i visine nove slike
$sirina = 500;
$visina = 300;

// Kreiranje nove prazne slike
$novaSlika = imagecreatetruecolor($sirina, $visina);

// Definisanje boje pozadine
$bojaPozadine = imagecolorallocate($novaSlika, 255, 0, 0); // Ovdje je
// postavljena crvena boja pozadine (RGB: 255, 0, 0)

// Popunjavanje pozadine sa definisanom bojom
imagefill($novaSlika, 0, 0, $bojaPozadine);

// Postavljanje zaglavlja HTTP odgovora kako bi se prikazala slika
header('Content-type: image/png');

// Ispisivanje slike kao PNG datoteke na izlaz
imagepng($novaSlika);

// Oslobađanje resursa
imagedestroy($novaSlika);
<!-- ?> -->
```

Slika 1: Generisanje prazne slike

Da bismo prikazali sliku u internet pretraživaču, postavljamo zaglavlje HTTP odgovora koristeći funkciju **header()**. U ovom slučaju, postavljamo tip slike na image/png kako bismo prikazali sliku u PNG formatu. Konačno, koristimo funkciju **imagepng()** za ispisivanje slike kao PNG datoteke na izlaz. Na kraju, oslobađamo resurse koje smo koristili pomoću funkcije **imagedestroy()**.

Ovaj primjer ilustruje osnovne korake za generisanje i prikazivanje slika u PHP-u. Može se prilagoditi za kreiranje slika različitih dimenzija i boja pozadine, omogućavajući programerima da dinamički generišu slike koje odgovaraju specifičnim zahtjevima aplikacija.

## SMANJIVANJE SLIKE KORISTEĆI RESAMPLING TEHNIKU

U drugom primjeru (slika 2), istražujemo kako smanjiti veličinu slike koristeći *resampling tehniku* u PHP-u. Prvo, definišemo putanju do originalne slike koju želimo smanjiti. Zatim koristimo funkciju **imagecreatefromjpeg()** da učitamo originalnu sliku u memoriju.

```
<?php

// Putanja do ogromne slike koju želimo smanjiti
$putanjaOgromneSlike = 'lav_3944x2585.jpg';

// Dimenzije nove slike
$novaSirina = 500;
$novaVisina = 300;

// Učitavanje ogromne slike
$originalnaSlika = imagecreatefromjpeg($putanjaOgromneSlike);

// Dobijanje dimenzija originalne slike
$originalnaSirina = imagesx($originalnaSlika);
$originalnaVisina = imagesy($originalnaSlika);

// Stvaranje prazne slike nove veličine
$novaSlika = imagecreatetruecolor($novaSirina, $novaVisina);

// Promjena veličine slike
imagecopyresampled($novaSlika, $originalnaSlika, 0, 0, 0, 0, $novaSirina,
$novaVisina, $originalnaSirina, $originalnaVisina);

// Postavljanje zaglavlja HTTP odgovora kako bi se prikazala nova slika
header('Content-type: image/jpeg');

// Ispisivanje nove slike na izlaz
imagejpeg($novaSlika);

// Oslobađanje resursa
imagedestroy($originalnaSlika);
imagedestroy($novaSlika);
?>
```

Slika 2: Smanjivanje slike koristeći resampling tehniku

Nakon što dobijemo dimenzije originalne slike pomoću funkcija **imagesx()** i **imagesy()**, stvaramo praznu sliku nove veličine koristeći **imagecreatetruecolor()**. Nakon toga, primjenjujemo funkciju **imagecopyresampled()** kako bismo smanjili originalnu sliku na zadate dimenzije nove slike, koristeći resampling tehniku. Ova tehnika održava kvalitet slike tokom procesa smanjivanja.

Da bismo prikazali novu sliku, postavljamo zaglavlje HTTP odgovora na image/jpeg i koristimo funkciju **imagejpeg()** za ispisivanje nove slike na izlaz. Kao i u prethodnom primjeru, oslobađamo resurse korišćene za rad sa slikama koristeći funkciju **imagedestroy()**.

Ovaj primjer demonstrira kako prilagoditi veličinu slike u PHP-u koristeći resampling tehniku. To je korisno kada je potrebno optimizovati veličinu slika za brže učitavanje i prikaz na web stranicama ili prilagoditi sliku za specifične zahteve aplikacije.

## MODIFIKACIJA POSTOJEĆE SLIKE

Treći primjer u ovom radu (slika 3) prikazuje modifikaciju postojeće slike koristeći nekoliko tehnika. Počinjemo učitavanjem slike iz JPEG formata pomoću funkcije **imagecreatefromjpeg()**. Zatim, koristimo funkciju **imagescale()** kako bismo promijenili veličinu slike na željenu vrijednost. U ovom primjeru, nova veličina je postavljena na 500 piksela. Nakon toga, koristimo funkciju **imagerotate()** kako bismo rotirali sliku za 90 stepeni u smjeru suprotnom od kazaljke na satu. Na kraju, primjenjujemo filter na sliku koristeći funkciju **imagefilter()**. U ovom slučaju, primjenjujemo crno-bijeli filter kako bismo sliku pretvorili u crno-bijelu verziju.

```
<?php
// Putanja do postojeće slike koju želimo modifikovati
$putanjaSlike = 'tigar.jpg';

// Učitavanje postojeće slike
$originalnaSlike = imagecreatefromjpeg($putanjaSlike);

// Promjena veličine slike
$novaVelicina = 500; // Nova veličina slike (širina i visina)
$modificiranaSlike = imagescale($originalnaSlike, $novaVelicina);

// Rotacija slike za 90 stepeni
$rotiranaSlike = imagerotate($modificiranaSlike, 90, 0);

// Primjena filtera – u ovom slučaju, crno-bijeli filter
imagefilter($rotiranaSlike, IMG_FILTER_GRAYSCALE);

// Postavljanje zaglavlja HTTP odgovora kako bi se prikazala slika
header('Content-type: image/jpeg');

// Ispisivanje modifikovane slike na izlaz
imagejpeg($rotiranaSlike);

// Oslobađanje resursa
imagedestroy($originalnaSlike);
imagedestroy($modificiranaSlike);
imagedestroy($rotiranaSlike);
?>
```

Slika 3: Modifikacija postojeće slike



Nakon primjene svih ovih transformacija, koristimo funkciju **header()** za postavljanje zaglavlja HTTP odgovora kako bismo prikazali modificovanu sliku kao JPEG na web stranici. Konačno, koristimo funkciju **imagejpeg()** da bismo ispisali modificovanu sliku na izlaz, nakon čega slijedi **imagedestroy()** funkcija, kao i u dosadašnjim primjerima.

Ovaj primjer demonstrira kako PHP može biti korišten za manipulaciju slikama, uključujući promjenu veličine, rotaciju i primjenu filtera. Kroz ovaj primjer, možemo vidjeti kako PHP pruža moćne funkcionalnosti za modifikaciju slika i otvara vrata kreativnim i interaktivnim mogućnostima u razvoju aplikacija.

## KOMBINOVANJE DVIJE SLIKE

Navedeni primjer (slika 4) pokazuje kombinovanje dvije slike u jednu kolaž sliku. Prvo učitavamo dvije slike koje želimo kombinovati koristeći funkciju **imagecreatefromjpeg()**. Nakon toga, dobijamo dimenzije obje slike pomoću funkcija **imagesx()** i **imagesy()**.

```
<?php
// Putanje do slika koje želimo kombinovati
$slika1 = 'kamila.jpg';
$slika2 = 'tigar.jpg';

// Učitavanje slika
$slikaObjekat1 = imagecreatefromjpeg($slika1);
$slikaObjekat2 = imagecreatefromjpeg($slika2);

// Dimenzije prve slike
$sirina1 = imagesx($slikaObjekat1);
$visina1 = imagesy($slikaObjekat1);

// Dimenzije druge slike
$sirina2 = imagesx($slikaObjekat2);
$visina2 = imagesy($slikaObjekat2);

// Kreiranje nove prazne slike za kolaž
$kolaz = imagecreatetruecolor($sirina1 + $sirina2, max($visina1, $visina2));

// Kopiranje prve slike na kolaž
imagecopy($kolaz, $slikaObjekat1, 0, 0, 0, 0, $sirina1, $visina1);

// Kopiranje druge slike na kolaž
imagecopy($kolaz, $slikaObjekat2, $sirina1, 0, 0, 0, $sirina2, $visina2);

// Postavljanje zaglavlja HTTP odgovora kako bi se prikazao kolaž
header('Content-type: image/jpeg');

// Ispisivanje kolaža na izlaz
imagejpeg($kolaz);

// Oslobađanje resursa
imagedestroy($slikaObjekat1);
imagedestroy($slikaObjekat2);
imagedestroy($kolaz);
?>
```

Slika 4: Kombinovanje dvije slike

Zatim kreiramo novu praznu sliku za kolaž koristeći funkciju **imagecreatetruecolor()**. Njene dimenzije se postavljaju tako da su širina jednaka zbiru širina dvije slike, a visina je postavljena na veću visinu od dvije slike.

Nakon toga, koristimo funkciju **imagecopy()** kako bismo kopirali prvu sliku na kolaž sliku. Prilikom kopiranja, specificiramo odgovarajuće koordinate i dimenzije. Zatim koristimo istu funkciju **imagecopy()** da bismo kopirali drugu sliku na kolaž sliku, pri čemu postavljamo koordinate tako da druga slika bude postavljena odmah do prve slike.

Nakon što smo završili sa kombinovanjem slika, koristimo funkciju **header()** da bismo postavili zaglavlje HTTP odgovora i prikazali kolaž sliku kao JPEG na web stranici. Konačno, koristimo funkciju **imagejpeg()** da bismo ispisali kolaž sliku na izlaz, a zatim **imagedestroy()**, kao u ranijim primjerima.

Ovaj primjer ilustruje kako PHP može kombinovati slike kako bi stvorio novu kompoziciju. To može biti korisno u raznim scenarijima, kao što su izrada kolaža, spajanje nekoliko slika u jednu ili kreiranje vizuelno bogatih elemenata u aplikacijama. PHP pruža funkcionalnosti za manipulaciju slikama i daje programerima slobodu da eksperimentišu i kreiraju raznovrsne vizuelne efekte.

## PRIKAZIVANJE SLIKA IZ TRENUTNOG DIREKTORIJUMA

Peti primjer u ovom radu prikazuje kako PHP može prikazati sve PNG, JPG i WEBP slike iz trenutnog direktorijuma na web stranici. Prvo, koristimo funkciju **glob()** koja pronalazi sve slike u trenutnom direktorijumu sa odgovarajućim ekstenzijama (**.png**, **.jpg** i **.webp**). Koristimo petlju **foreach()** da bismo prošli kroz sve pronađene slike.

```
<?php

echo "<h3> PNG SLIKE </h3>";

// Pronalazi sve PNG slike u trenutnom direktoriju
foreach (glob("*.png") as $filename) {
    echo "<p>$filename</p>"; // Ispisuje naziv slike
    echo "<img src='$filename' alt='$filename' /><br><br>"; // Prikazuje sliku
}

echo "<h3> JPG SLIKE </h3>";

// Pronalazi sve JPG slike u trenutnom direktoriju
foreach (glob("*.jpg") as $filename) {
    echo "<p>$filename</p>"; // Ispisuje naziv slike
    echo "<img src='$filename' alt='$filename' /><br><br>"; // Prikazuje sliku
}

echo "<h3>WEBP SLIKE</h3>";

// Pronalazi sve WEBP slike u trenutnom direktoriju
foreach (glob("*.webp") as $filename) {
    echo "<p>$filename</p>"; // Ispisuje naziv slike
    echo "<img src='$filename' alt='$filename' /><br><br>"; // Prikazuje sliku
}

?>
```

Slika 5: Prikazivanje slika iz trenutnog direktorijuma

Unutar petlje, prikazujemo naziv svake slike koristeći element **\$filename** i dodajemo HTML oznake **<p>** kako bismo stvorili paragraf sa nazivom slike. Zatim koristimo element **\$filename** u atributu **src** kako bismo postavili putanju slike, tako da se slika prikazuje na web stranici. Nakon prikaza slike, dodajemo prazne linije kako bismo stvorili razmak između slika.

Ovaj primjer ilustruje kako PHP može dinamički prikazivati slike na web stranici. Koristeći funkciju **glob()**, možemo pronaći i prikazati slike sa određenim ekstenzijama, što omogućava fleksibilnost u upravljanju slikama na web stranicama. Ovo je korisno kada želimo prikazati galeriju slika, generisati izveštaje sa slikama ili prikazivati vizuelne elemente koji se mogu mijenjati dinamički.

## ZAKLJUČAK

U ovom seminarskom radu smo istražili pet primjera generisanja slika u PHP programu. Svaki primjer je pružio uvid u različite tehnike za kreiranje, modifikovanje i prikazivanje slika korišćenjem ugrađenih funkcija i biblioteka dostupnih u PHP-u.

Kroz ove primere smo se upoznali sa osnovnim konceptima generisanja slika u PHP-u. Naučili smo kako kreirati praznu sliku određene širine i visine, kako promeniti veličinu slike korišćenjem resamplinga, kako rotirati sliku, primeniti filtere i kombinovati slike u kolaž. Takođe smo videli kako prikazivati slike na web stranici dinamički, pronalazeći slike sa određenim ekstenzijama u trenutnom direktorijumu.

PHP pruža moćne funkcionalnosti za rad sa slikama, omogućavajući programerima da generišu, modifikuju i manipulišu slikama u svojim aplikacijama. Generisanje slika je posebno korisno u web razvoju za prikazivanje vizuelnih elemenata, galerija slika, generisanje izveštaja i još mnogo toga.

U zaključku, generisanje slika u PHP-u pruža programerima moćne alate za kreativno upravljanje slikama u svojim projektima. Uzimajući u obzir sve naučeno kroz ove primere, mogućnosti za dalje istraživanje i primenu PHP-a u generisanju slika su široke i obećavajuće.