

## 1. Opiši pojam MVC.

**LINK:**chrome-

extension://efaidnbmnnnibpcajpcgltclfindmkaj/http://www.efos.unios.hr/upravljanje-marketingom/wp-content/uploads/sites/228/2013/04/RPA\_P4\_MVC-web-aplikacije.pdf

MVC arhitektura dijeli aplikaciju u tri osnovne komponente **Model**, **Pogled** (*View*) i **Kontroler** (*Controller*).

MVC aplikacija komunicira s korisnikom putem **Pogleda** (eng. *View*), gdje korisnik unosom zahtjeva šalje naredbe **Kontroleru** (eng. *Controller*), dok **Kontroler** od **Modela** traži da obavi radnju i vrati mu rezultat. **Rezultat radnje** zatim **Kontroler** šalje u **Pogled** gdje ga korisnik može vidjeti.

## 2. Opiši proces zahtjeva unutar Laravela od ruta do pogleda.

**LINK:** <https://zir.nsk.hr/islandora/object/ffzg:2496/datastream/PDF/download>

Nakon što su postavljeni modeli, prikazi i upravljači, sve te komponente potrebno je povezati, i za to služi tok usmjeravanja. U Laravelu je to omogućeno korištenjem klase *Route*. Unutar mape *routes* nalaze se datoteke koje Laravel sam učitava, a to su *api.php*, *channels.php*, *console.php* i *web.php*. Pri izradi jednostavnijih aplikacija, najčešće se uređuje samo *web.php* datoteka. Rute koje se definiraju u toj datoteci dostupne su unosom definiranog URL-a u preglednik.

**Vrste tokova usmjeravanja** koje su dostupne su **rute preusmjeravanja** i **rute prikaza**. Rutom preusmjeravanja (engl. *redirect routes*) moguće je preusmjeriti jedan URI na drugi, a to se može napraviti koristeći *Route::redirect*. U slučaju da je potrebno samo vratiti neki prikaz, može se koristiti *Route::view* metoda i kao i ranija metoda ne zahtijeva dodatne informacije kao što je puna ruta ili upravljač koji će koristiti.

### 3. Opiši pojam ORM.

**LINK:** [https://en.wikipedia.org/wiki/Object%E2%80%93relational\\_mapping](https://en.wikipedia.org/wiki/Object%E2%80%93relational_mapping)

**Object relational mapping** (Objektno relacijsko preslikavanje) je tehnika programiranja za pretvorbu podataka između relacijske baze podataka i hrpe objektno orijentiranog programskog jezika. Time se zapravo stvara baza podataka virtualnih objekata koja se može koristiti unutar programskog jezika.

### 4. Što je Bootstrap?

**LINK:**

**Bootstrap** je besplatni CSS okvir otvorenog koda usmjeren na responzivni front-end web razvoj prvenstveno za mobilne uređaje. Sadrži HTML, CSS i predloške dizajna temeljene na JavaScriptu.

### 5. Napiši 3 komponente iz sustava Bootstrap po izboru.

**LINK:** <https://hr.education-wiki.com/3104974-bootstrap-components>

**Grupe gumba** - Pomoću ove komponente Bootstrap možete grupirati skup gumba zajedno u nizu jedan pored drugog.

**Padajuće tipke** - ova komponenta koristi se za korištenje gumba za aktiviranje padajućeg izbornika.

**Značke** - Ovo je komponenta koja označava nešto poput nepročitanih stavki dodavanjem klase `.badge` u DOM element, po mogućnosti raspon.

## 6. Koje su novosti (nove oznake) u HTML 5?

LINK: <https://www.webtech.com.hr/html5.php>

HTML5 je smanjio potrebu za dodatnim priključcima (poput Flash-a), tolerantniji je na pogreške, sadrži više **oznaka** (elemenata).

### Novi elementi

---

Novi elementi unutar HTML5:

Novi elementi	
Element (Tag)	Opis
<acronym>	Definira akronim
<applet>	Definira applet
<basefont>	Definira glavni font za stranicu
<big>	Definira veliki tekst
Novi elementi forme	
Element (Tag)	Opis
<datalist>	Definira listu predefiniranih opcija
<keygen>	Definira generator ključeva (za forme)
<output>	Definira rezultat računanja
Novi medijski elementi	
Element (Tag)	Opis
<audio>	Definira zvučnu datoteku
<video>	Definira video datoteku
<source>	Definira višestruki medijski izvor za <video> i <audio>
<embed>	Definira prostor za vanjsku aplikaciju ili dodatak (plug-in)
<canvas>	Definira prostor grafičkog elementa , točnije prostor za crtanje

## 7. Opiši zašto se koriste migracijske skripte? Koje su prednosti?

LINK:

<https://www.google.com/search?q=why+we+use+migrations+script&oq=why+we+use+migrations+script+&aqs=chrome..69i57j0i546l4.10816j0j7&sourceid=chrome&ie=UTF-8>

Svrha korištenja migracijskih skripti je osigurati da se promjene baze podataka mogu pravilno primijeniti bez narušavanja integriteta baze podataka, onemogućavaju ručne promjene konfiguracije i osiguravaju sprječavanja gubitka podataka.

## 8. U kojoj datoteci se pohranjuju podaci za spajanje na bazu?

U **.env** datoteci.

## 9. Koja php artisan naredba pokreće Laravel projekt?

**LINK:** <https://blog.devsense.com/2019/how-to-install-laravel-for-visual-studio-code>

php artisan serve

## 10. Koja naredba kreira Laravel projekt?

**LINK:** <https://blog.devsense.com/2019/how-to-install-laravel-for-visual-studio-code>

composer create-project laravel/laravel

## 11. Kako bi pronašli popis svih php artisan naredbi?

Upišemo u terminal php artisan list

## 12. Opiši primjerom PHP klasu? Što je nasljeđivanje?

**LINK:** [https://www.w3schools.com/php/php\\_oop\\_classes\\_objects.asp](https://www.w3schools.com/php/php_oop_classes_objects.asp)

Klasa ili razred je nacrt za jedan ili proizvoljni broj objekata koji imaju slične karakteristike definirane unutar te klase.

Deklariramo klasu pod imenom Fruit koja se sastoji od svojstva jednog svojstva (\$name) i dvije metode set\_name() i get\_name() za postavljanje i dobivanje svojstva \$name.

```
<?php
class Fruit {
    public $name;

    function set_name($name) {
        $this->name = $name;
    }
    function get_name() {
        return $this->name;
    }
}
?>
```

### Nasljeđivanje:

**LINK:** [chrome-extension://efaidnbmnnnibpcajpcgicfindmkaj/https://repozitorij.ffos.hr/islandora/object/fos:16/datastream/PDF](https://efaidnbmnnnibpcajpcgicfindmkaj/https://repozitorij.ffos.hr/islandora/object/fos:16/datastream/PDF)

Nasljeđivanje omogućuje da jedna klasa naslijedi sve metode, karakteristike i ponašanje neke druge klase. Klase nasljeđuju pomoću riječi `extends`.

