

3. [6] Potrebno je testirati deo programskog koda, tako što ćete za sledeću klasu realizovati jedinične testove (po želji uraditi u PHP ili Python Django):

Programski kod u PHP

(koristiti za testnu klasu sledeću klasu: *use PHPUnit\Framework\TestCase;*)

```
class Author {
    public $first_name;
    public $last_name;
    public $birth_year;
    public $sex;
    public $email;

    public function get_sex() {
        return $this->sex;
    }

    public function calculate_age() {
        $current_year = date('Y');
        $age = $current_year - $this->birth_year;
        return $age;
    }

    public function get_age_category() {
        $age = $this->calculate_age();
        if ($age < 18) {
            return "children";
        } elseif ($age >= 19 && $age <= 25) {
            return "student";
        } elseif ($age >= 26 && $age <= 64) {
            return "adult";
        } else {
            return "senior";
        }
    }
}
```

Programski kod u *Python/Django*

(koristiti za testnu klasu sledeću klasu: *from django.test import TestCase*)

```
from django.db import models
from datetime import date

class Author(models.Model):
    first_name = models.CharField(max_length=32)
    last_name = models.CharField(max_length=32)
    birth_year = models.IntegerField()
    sex = models.CharField(max_length=1, choices=[('M', 'Male'), ('F', 'Female')])
    email = models.EmailField(max_length=50)

    def get_sex(self):
        return self.sex

    def calculate_age(self):
        current_year = date.today().year
        age = current_year - self.birth_year
        return age

        def get_age_category(self):
            age = self.calculate_age()
            if age < 18:
                return "children"
            elif 19 <= age <= 25:
                return "student"
            elif 26 <= age <= 64:
                return "adult"
            else:
                return "senior"
```


Junski ispitni rok - praktični deo ispita -

Napomena: Upotreba štampane literature i komunikacionih uređaja je zabranjena i strogo kažnjiva. Dozvoljena je samo upotreba elektronskih materijala sa diska *Materijali*. Ispit traje 165 minuta.

Broj indeksa	Ime i prezime	Broj poena

4. [28] Posmatra se informacioni sistem za usluge rezervisanja termina za tehnički pregled i registraciju automobila. U sistemu se vodi evidencija o više agencija za tehnički pregled automobila. Za svaku od njih se prati naziv, adresa, cena tehničkog pregleda, radno vreme (smatrati da je isto radnim danima i vikendom) i već rezervisani termini. Termini traju sat vremena. Smatrati da radna vremena svih agencija počinju i završavaju se na pun sat i da svaka od agencija ima kapacitet da u jednom terminu vrši tehnički pregled i registraciju samo jednog vozila.

Sistem omogućava prijavu koja zahteva unos registarskih tablica i unos datuma kada korisnik želi da obavi tehnički pregled i registraciju. Pored toga, korisnik u ovom koraku može, ali ne mora izabrati agenciju za tehnički pregled u kojoj želi da obavi registraciju. Nakon unosa registarskih tablica, izbora datuma i potencijalno izbora agencije, pritiskom na odgovarajuće dugme prelazi se na narednu stranicu sistema ili se ispisuje greška (u slučaju da nisu unete tablice ili izabran datum ili je izabrani datum već prošao).

Na drugoj stranici sistema tabelarno se prikazuju dostupni termini za registraciju automobila. Ukoliko nema dostupnih termina, ispisuje se poruka. Svaki red tabele odgovara jednom dostupnom terminu i ima dugme kojim je moguće izvršiti rezervaciju odgovarajućeg termina. Pritiskom na dugme se u bazi evidentira rezervacija, a na trećoj stranici sistema korisniku ispisuje u kojoj je agenciji rezervisao registraciju, kao i datum i vreme termina koji je rezervisan. Ukoliko je neko drugi u međuvremenu rezervisao taj termin, na trećoj stranici se ispisuju informacije o neuspešnoj rezervaciji.

- a) [14] Realizovati MVC (*Model-View-Controller*) aplikaciju korišćenjem programskog jezika *PHP* ili *Python*. Implementaciju je moguće raditi korišćenjem radnih okvira *CodeIgniter* ili *Laravel* sa uzorkom MVC (*Model-View-Controller*), ili *Django* sa uzorkom MTV (*Model-Template-View*). Aplikacija je potrebno da sadrži isključivo deo sistema koji se odnosi na rezervaciju termina.

Očekivani rezultat zadatka 3a): čitav projekat veb aplikacije komprimovati u ZIP arhivu i iskopirati na disk Rad (L:).

[Pocetna](#)

WEB\src_project\app\Filters

Izbor termina:

Agencija	Datum	Vreme	Iznos	Rezervisi
Agencija 2	2023-06-15	10	5000	Rezervisi
Agencija 2	2023-06-15	11	5000	Rezervisi
Agencija 2	2023-06-15	14	5000	Rezervisi

Copyright 2023

[Pocetna](#)

Rezervacija termina za registraciju:

Agencija:

Registarske tablice:

Datum:

Copyright 2023

[Pocetna](#)

Uspešna rezervacija termina

Registarske tablice: BG135SA

Datum: 2023-06-16

Vreme: 12

Agencija: Agencija 2

Copyright 2023

- b) [7] Korišćenjem alata Selenium IDE, napisati test primere kojim ćete testirati registraciju termina, koja je opisana u tekstu zadatka i implementirana u tački a).

Očekivani rezultat zadatka 3b): projekat okruženja Selenium IDE, sa svim testovima, snimiti kao fajl sa ekstenzijom .side i prekopirati na disk Rad (L:) . Dovoljno je testove pokrenuti samo u okruženju Google Chrome ili Mozilla Firefox.

- c) [7] Nacrtati dijagram sekvence za funkcionalnost rezervacije termina za registraciju, prema opisanoj specifikaciji sistema.

Očekivani rezultat zadatka 3c): dijagrame nacrtati na ovom papiru.

Napomena:

Rešenja iskopirati na disk Rad (L:). Samo rešenja koja su predata na serveru biće ocenjena!