Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za elektronsko poslovanje



Web aplikacija za rezervaciju avio karti

Seminarski rad iz Internet Tehnologija

Profesor: Studenti:

Zorica Bogdanović Mutavdžić Miloš 88/16

Kostić Pavle 1026/17

Knežević Lazar 1047/17

Beograd, 2020.god

Sadržaj

[1.Korisnički zahtev 4](#_Toc30101678)

[1. Opis sistema 7](#_Toc30101679)

[1.1 Slučajevi korišćenja 7](#_Toc30101680)

[2.1.1. SK Prijava korisnika 7](#_Toc30101681)

[2.1.2. SK Unos leta 8](#_Toc30101682)

[2.1.3 SK Pretraživanje rezervacija 9](#_Toc30101683)

[2.1.4 SK Izmena leta 10](#_Toc30101684)

[2.1.5 SK Unos nove slike u galeriju 11](#_Toc30101685)

[2.1.6 SK Unos nove rezervacije 12](#_Toc30101686)

[2.1.7 SK Brisanje rezervacija 13](#_Toc30101687)

[2.1.8 Prikaz grafika 14](#_Toc30101688)

[2.1.9 SK Prikaz letova 15](#_Toc30101689)

[2.1.10 SK Registrovanje korisnika 16](#_Toc30101690)

[1.2 Opis strukture sistema 17](#_Toc30101691)

[17](#_Toc30101692)

[1.3 Opis procesa – dijagram sekvenci 18](#_Toc30101693)

[2.3.1 DS Prijava korisnika 18](#_Toc30101694)

[2.3.2 DS Unos leta 20](#_Toc30101695)

[2.3.3 DS Pretraživanje rezervacija 22](#_Toc30101696)

[2.3.4 DS Izmena leta 23](#_Toc30101697)

[2.3.5 DS Unos nove slike u galeriju 25](#_Toc30101698)

[2.3.6 DS Unos nove rezervacije 25](#_Toc30101699)

[2.3.7 DS Brisanje rezervacije 26](#_Toc30101700)

[2.3.8 DS Prikaz grafika 28](#_Toc30101701)

[2.3.9 DS Prikaz letova 29](#_Toc30101702)

[2.3.10 DS Registrovanje korisnika 29](#_Toc30101703)

[1.4 Model podataka 31](#_Toc30101704)

[2.5. Specifikacija REST APIja 32](#_Toc30101705)

[2.5.1 Funkcija sviLetovi() 32](#_Toc30101706)

[2.5.2 Funkcija sveKlase() 33](#_Toc30101707)

[2.5.3 Funkcija ubaciKorisnika(data) 34](#_Toc30101708)

[2.5.4 Funkcija unosLeta(data) 35](#_Toc30101709)

[3. Korišćene tehnologije 36](#_Toc30101710)

[4.Uputstvo za korisnike 38](#_Toc30101711)

[5. Delovi koda seminarskog rada “LetLine” 48](#_Toc30101712)

[5.1 Prikaz svih letova 48](#_Toc30101713)

[Prikaz vremenske prognoze 54](#_Toc30101714)

[Unos novog leta 56](#_Toc30101715)

[Unos nove slike u galeriju 56](#_Toc30101716)

[Prikaz grafika 58](#_Toc30101717)

[Registracija 60](#_Toc30101718)

[Login 63](#_Toc30101719)

[Rezervacije 64](#_Toc30101720)

[Prikaz svih rezervacija i brisanje rezervacije 68](#_Toc30101721)

[Sopstveni servisi 70](#_Toc30101722)

# 1.Korisnički zahtev

Tema seminarskog rada je kreiranje web aplikacije za rezervisanje avio karti. Korisnicima je potrebno omogućiti logovanje, a ukoliko korisnici ne postoje u bazi, omogućiti im registrovanje i upisivanje u bazu podataka.

Korisnici koji imaju ulogu administratora, imaju mogućnost da unose nove letove (odnosno destinacije putovanja sa dodatnim atributima o samom letu).

Pored navednog slučaja korišćenja, administratori imaju mogućnost i izmene podataka o prethodno unetom letu, omogućeno im je i brisanje rezervacija iz tabele rezervacije u bazi, kao i prikaz grafika, na kojem se prikazuju, procentualno, rezervacije po klasi leta.

Administratori, kao i obični korisnici imaju mogućnost rezervacije. Administratori mogu i da ubacuju nove slike u galeriju slika.Na početnoj strani je potrebno omogućiti prikaz svih dostupnih letova.

Pristup imaju **dve vrste korisnika:**

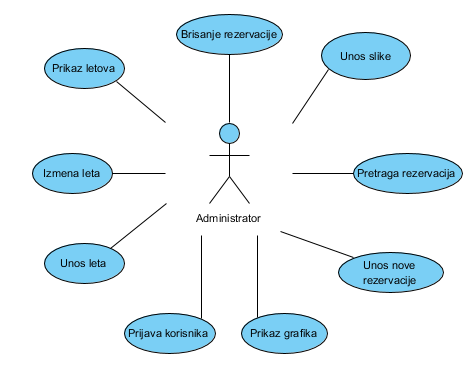
* administrator i
* korisnik

**Administratoru** omogućiti sledeće funkcionalnosti**:**

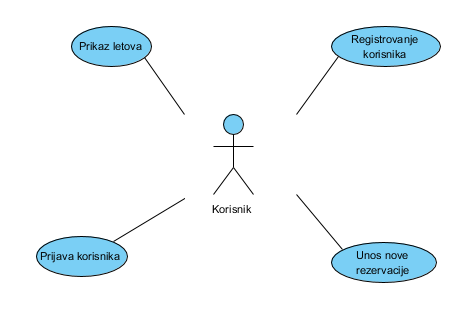
1. Prijava korisnika
2. Unos leta
3. Izmena leta
4. Pretraga rezervacija
5. Unos nove slike u galeriju
6. Unos nove rezervacije
7. Brisanje rezervacije
8. Prikaz grafika
9. Prikaz letova

**Korisniku** omogućiti sledeće funkcionalnosti:

1. Prijava korisnika
2. Registrovanje korisnika
3. Unos nove rezervacije
4. Prikaz letova



Slika 1 Slučajevi korišćenja za administratora



Slika 2 Slučajevi korišćenja za korisnika

# Opis sistema

## Slučajevi korišćenja

### SK Prijava korisnika

**Naziv SK**

*Prijavljivanje na sistem*

**Aktori Sk**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem*

**Preduslov**

*Sistem je uključen*

**Osnovni scenario SK**

1. *Korisnik unosi korisničko ime i lozinku(APUSO)*
2. *Korisnik poziva sistem da se uloguje (proveri podatke) (APSO)*
3. *Sistem proverava podatke o korisniku (SO)*
4. *Sistem prikazuje korisniku poruku „Uspešno ulogovan korisnik“(IA)*

**Alternativni scenario**

* 1. *Ako sistem ne može da nađe korisnika, prikazuje poruku „Neuspešno prijavljivanje.“(IA)*

### SK Unos leta

**Naziv SK**

*Unos leta*

**Aktori Sk**

*Administrator*

**Učesnici SK**

*Administrator i sistem*

**Preduslov**

*Sistem je uključen i administrator je ulogovan. Sistem prikazuje formu za rad sa letovima.*

**Osnovni scenario SK**

1. *Administrator unosi podatke o letu(APUSO)*
2. *Administrator poziva sistem da zapamti podatke o letu(APSO)*
3. *Sistem pamti podatke o letu (SO)*
4. *Sistem prikazuje administratoru poruku „Uspešno unet novi let.“(IA)*

**Alternativni scenario**

* 1. *Ako sistem ne može da zapamti podatke o letu, administratoru prikazuje poruku „Neuspešno unošenje novog leta.“ (IA)*

### 2.1.3 SK Pretraživanje rezervacija

**Naziv SK**

*Pretraživanje rezervacija*

**Aktori Sk**

*Administrator*

**Učesnici SK**

*Administrator i sistem*

**Preduslov**

*Sistem je uključen i administrator je ulogovan.*

**Osnovni scenario SK**

1. *Administrator unosi vrednost po kojoj pretražuje rezervacije(APUSO)*
2. *Administrator poziva sistem da nađe rezervacije po zadatoj vrednosti(APSO)*
3. *Sistem traži rezervacije po zadatoj vrednosti (SO)*
4. *Sistem prikazuje administratoru listu rezervacija (IA)*

### 2.1.4 SK Izmena leta

**Naziv SK**

*Izmena leta*

**Aktori Sk**

*Administrator*

**Učesnici SK**

*Administrator i sistem*

**Preduslov**

*Sistem je uključen i administrator je ulogovan.Učitana je lista letova(mesta polaska i mesta dolaska).*

**Osnovni scenario SK**

1. *Administrator bira destinacije leta čiju klasu ili cenu karte želi da menja (APUSO)*
2. *Administrator unosi novu klasu ili cenu leta (APUSO)*
3. *Administrator poziva sistem da zapamti nove podatke o letu (APSO)*
4. *Sistem pamti nove podatke o letu (SO)*
5. *Sistem prikazuje administratoru poruku „Uspešno izmenjen let“ (IA)*

**Alternativni scenario**

* 1. *Ako sistem ne može da zapamti podatke o letu, prikazuje poruku „Neuspešno izmenjen let. “(IA)*

### 2.1.5 SK Unos nove slike u galeriju

**Naziv SK**

*Unos nove slike*

**Aktori Sk**

*Administrator*

**Učesnici SK**

*Administrator i sistem*

**Preduslov**

*Sistem je uključen i administrator je ulogovan*.

**Osnovni scenario SK**

1. *Administrator bira novu sliku (APUSO)*
2. *Administrator poziva sistem da ubaci novu sliku u galeriju (APSO)*
3. *Sistem pamti novu sliku (SO)*
4. *Sistem prikazuje administratoru novu sliku u galeriji (IA)*

### 2.1.6 SK Unos nove rezervacije

**Naziv SK**

*Unos nove rezervacije*

**Aktori Sk**

*Korisnik(ili administrator)*

**Učesnici SK**

*Korisnik(ili administrator) i sistem*

**Preduslov**

*Sistem je uključen I korisnik je ulogovan. Sistem prikazuje formu za rad sa rezervacijama. Učitane su lista relacija i lista klasa .*

**Osnovni scenario SK**

1. *Korisnik unosi podatke o rezervaciji (APUSO)*
2. *Korisnik poziva sistem da zapamti podatke o rezervaciji (APSO)*
3. *Sistem pamti podatke o rezervaciji (SO)*
4. *Sistem prikazuje korisniku poruku „Uspešno rezervisano“(IA)*

**Alternativni scenario**

* 1. *Ako sistem ne može da zapamti podatke o rezervaciji, korisniku prikazuje poruku „Neuspešna rezervacija“ (IA)*

### 2.1.7 SK Brisanje rezervacija

**Naziv SK**

*Brisanje rezervacija*

**Aktori Sk**

*Administrator*

**Učesnici SK**

*Administrator i sistem*

**Preduslov**

*Sistem je uključen i administrator je ulogovan. Sistem prikazuje formu za rad sa rezervacijama.*

**Osnovni scenario SK**

1. *Administrator unosi vrednost po kojoj pretražuje rezervacije (APUSO)*
2. *Administrator poziva sistem da pronađe rezervacije po zadatoj vrednosti (APSO)*
3. *Sistem traži rezervacije po zadatoj vrednosti (SO)*
4. *Sistem prikazuje administratoru listu rezervacija (IA)*
5. *Administrator bira rezervaciju koju će obrisati (APUSO)*
6. *Administrator poziva sistem da obriše izabranu rezervaciju (APSO)*
7. *Sistem briše rezervaciju (SO)*

**Alternativni scenario**

* 1. *Ako sistem ne može da obriše rezervaciju, administratoru prikazuje poruku „Došlo je do greške prilikom brisanja“ (IA)*

### 2.1.8 Prikaz grafika

**Naziv SK**

*Prikaz grafika*

**Aktori Sk**

*Administrator*

**Učesnici SK**

*Administrator i sistem*

**Preduslov**

*Sistem je uključen i administrator je ulogovan*.

**Osnovni scenario SK**

1. *Administrator poziva sistem da prikaže grafik (APSO)*
2. *Sistem obrađuje podatke i kreira grafik (SO)*
3. *Sistem prikazuje administratoru grafik (IA)*

### 2.1.9 SK Prikaz letova

**Naziv SK**

*Prikaz letova*

**Aktori Sk**

*Korisnik(ili administrator)*

**Učesnici SK**

*Korisnik(ili administrator)i sistem*

**Preduslov**

*Sistem je uključen.*

**Osnovni scenario SK**

1. *Korisnik poziva sistem da učita listu letova (APSO)*
2. *Sistem učitava listu letova (SO)*
3. *Sistem prikazuje korisniku listu letova (IA)*

### 2.1.10 SK Registrovanje korisnika

**Naziv SK**

*Registrovanje korisnika*

**Aktori Sk**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem*

**Preduslov**

*Sistem je uključen.Sistem prikazuje formu za registrovanje novog korisnika.*

**Osnovni scenario SK**

1. *Korisnik unosi podatke (APUSO)*
2. *Korisnik poziva sistem da zapamti podatke (APSO)*
3. *Sistem pamti podatke (SO)*
4. *Sistem prikazuje korisniku poruku „Uspešno registrovan korisnik“(IA)*

**Alternativni scenario**

* 1. *Ako sistem ne može da zapamti podatke o krisniku, prikazuje poruku „Neuspešna registracija korisnika.“(IA)*

## Opis strukture sistema

## C:\Users\Milica\Desktop\seminarskiIteh\dijagramKlasa.png

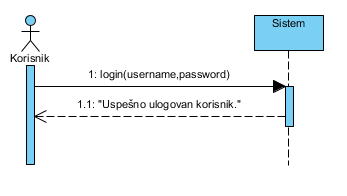
Slika 3 Dijagram klasa

## Opis procesa – dijagram sekvenci

### 2.3.1 DS Prijava korisnika

**Osnovni scenario SK**

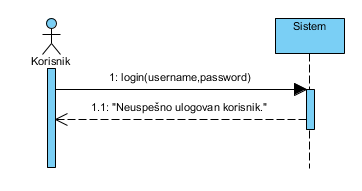
1. Korisnik poziva sistem da se uloguje (proveri podatke) (APSO)
2. Sistem prikazuje korisniku poruku „Uspešno ulogovan korisnik“ (IA)



Slika 4 DS uspešno logovanje

**Alternativni scenario**

* 1. Ako sistem ne može da nađe korisnika, prikazuje poruku „Neuspešno logovanje korisnika.“(IA)

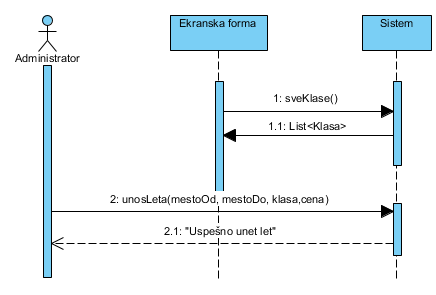


Slika 5 DS neuspešno logovanje

### 2.3.2 DS Unos leta

**Osnovni scenario SK**

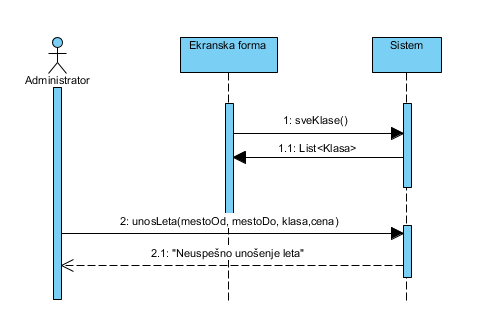
1. Administrator poziva sistem da zapamti podatke o letu(APSO)
2. Sistem prikazuje administratoru poruku „Uspešno unet novi let“(IA)



Slika 6DS unos leta

**Alternativni scenario**

* 1. Ako sistem ne može da zapamti podatke o letu, administratoru prikazuje poruku „Neuspešno unošenje novog leta“ (IA)

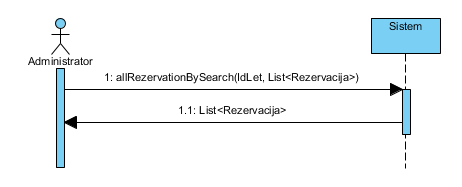
****

Slika 7 DS neuspesno unosenje leta

### 2.3.3 DS Pretraživanje rezervacija

**Osnovni scenario SK**

1. Administrator poziva sistem da nađe rezervacije po zadatoj vrednosti(APSO)
2. Sistem prikazuje administratoru listu rezervacija (IA)

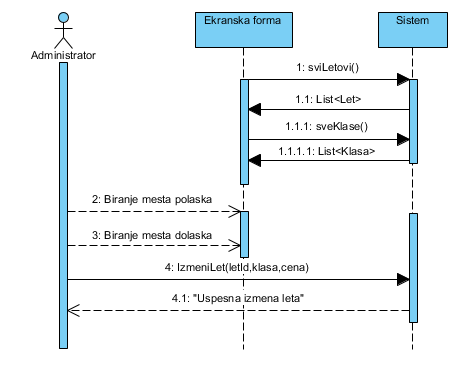


Slika 8 DS pretrazivanje rezervacija

### 2.3.4 DS Izmena leta

**Osnovni scenario SK**

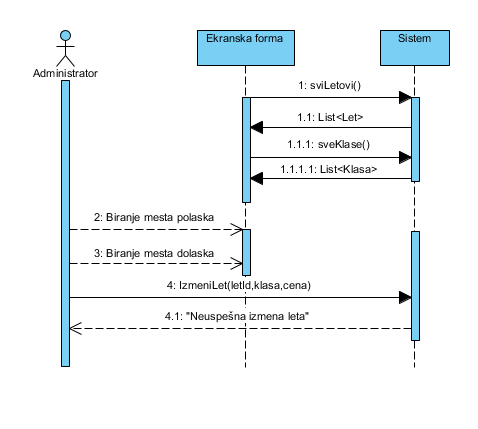
1. Ekranska forma poziva sistem da učita listu letova,tačnije njihovih destinacija i klasa (APSO)
2. Sistem prikazuje gore navedene liste (IA)
3. Administrator poziva sistem da prikaže izabranu relaciju (APSO)
4. Sistem prikazuje izabranu relaciju (IA)
5. Administrator poziva sistem da zapamti nove podatke o letu (APSO)
6. Sistem prikazuje administratoru poruku „Uspešna izmena leta“ (IA)



Slika 9 DS uspesna izmene leta

**Alternativni scenario**

* 1. Ako sistem ne može da zapamti podatke o letu, prikazuje poruku „Neuspešna izmena leta.“(IA)

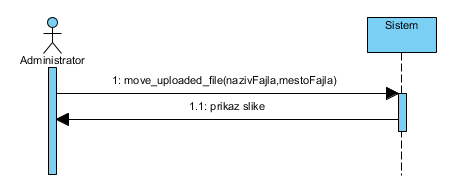
****

Slika 10 DS neuspesna izmena leta

### 2.3.5 DS Unos nove slike u galeriju

**Osnovni scenario SK**

1. Administrator poziva sistem da ubaci novu sliku u galeriju (APSO)
2. Sistem prikazuje administratoru novu sliku u galeriji (IA)

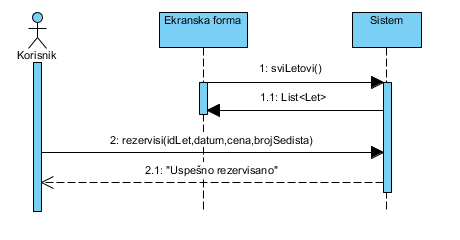


Slika 11 DS prikaz slike

### 2.3.6 DS Unos nove rezervacije

**Osnovni scenario SK**

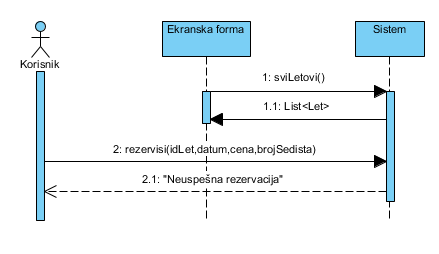
1. Korisnik poziva sistem da zapamti podatke o rezervaciji (APSO)
2. Sistem prikazuje korisniku poruku „Uspešno rezervisano“(IA)



Slika 12 DS uspesno rezervisano

**Alternativni scenario**

* 1. Ako sistem ne može da zapamti podatke o rezervaciji, korisniku prikazuje poruku „Neuspešna rezervacija“ (IA)

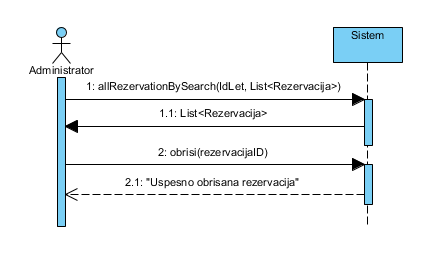
****

Slika 13 DS neuspesno rezervisano

### 2.3.7 DS Brisanje rezervacije

**Osnovni scenario SK**

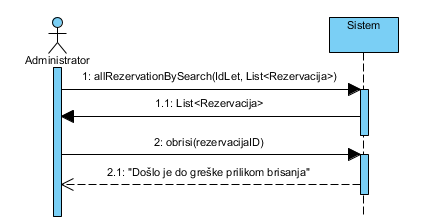
1. Administrator poziva sistem da pronađe rezervacije po zadatoj vrednosti (APSO)
2. Sistem prikazuje administratoru listu rezervacija (IA)
3. Administrator poziva sistem da obriše izabranu rezervaciju (APSO)



Slika 14 DS uspesno obrisana rezervacija

**Alternativni scenario**

* 1. Ako sistem ne može da obriše rezervaciju, administratoru prikazuje poruku „Došlo je do greške prilikom brisanja“ (IA)

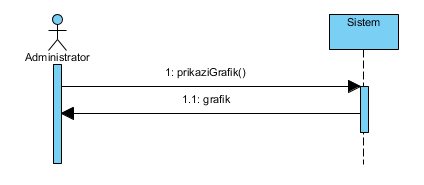
****

Slika 15 DS neuspesno brisanje

### 2.3.8 DS Prikaz grafika

**Osnovni scenario SK**

1. Administrator poziva sistem da prikaže grafik (APSO)
2. Sistem prikazuje administratoru grafik (IA)

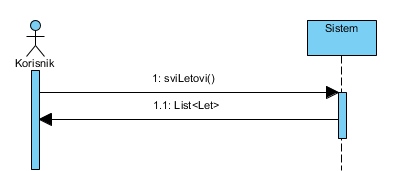


Slika 16 DS prikaz grafika

### 2.3.9 DS Prikaz letova

**Osnovni scenario SK**

1. Korisnik poziva sistem da učita listu letova (APSO)
2. Sistem prikazuje korisniku listu letova (IA)

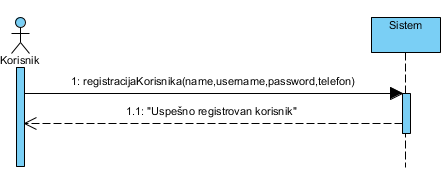


Slika 17 DS prikaz letova

### 2.3.10 DS Registrovanje korisnika

**Osnovni scenario SK**

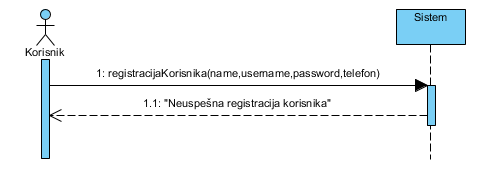
1. Korisnik poziva sistem da zapamti podatke (APSO)
2. Sistem prikazuje korisniku poruku „Uspešno registrovan korisnik“(IA)



Slika 18 DS uspesno registrovan korisnik

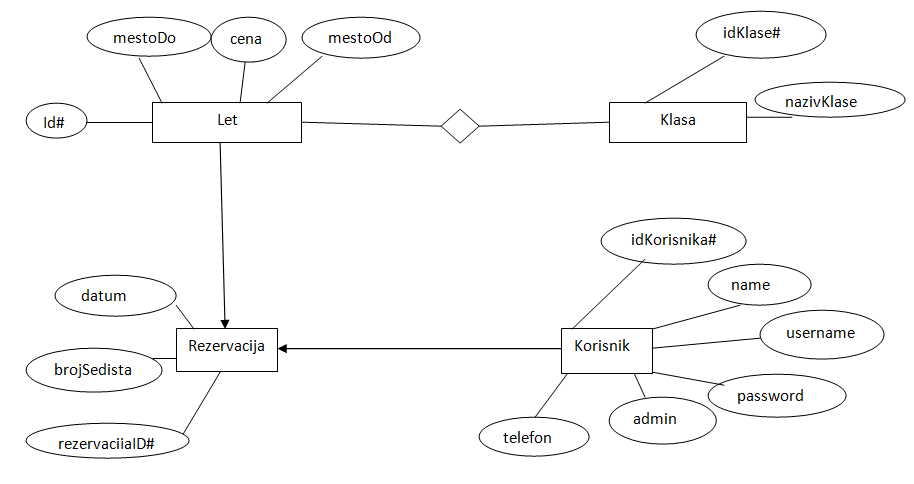
**Alternativni scenario**

* 1. Ako sistem ne može da zapamti podatke o krisniku, prikazuje poruku „Neuspešna registracija korisnika.“(IA)

****

Slika 19 DS neuspesna registracija korisnika

## Model podataka



Slika 20 PMOV

# 2.5. Specifikacija REST APIja

### 2.5.1 Funkcija sviLetovi()

|  |  |
| --- | --- |
| Opis funkcije | Vraćanje svih vrsta letova |
| HTTP metoda | GET |
| URL | /letovi.json |
| URL parametri | (nema) |
| HTTP body parametric | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametric | JSON objekat sa atributima id[int],mestoOs[varChar],mestoDo[varChar],klasa[int], cena[double]  Primer:  {  "id":"2",  "mestoOd":"Berlin",  "mestoDo":"Budimpesta",  "klasa":"3"  "cena":"3"  } |
| Format izlaznih parametara | application/json |

### 2.5.2 Funkcija sveKlase()

|  |  |
| --- | --- |
| Opis funkcije | Vraćanje svih tipova klasa |
| HTTP metoda | GET |
| URL | /klase.json |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | JSON objekat sa atributima idKlase[int],nazivKlase[string]  Primer:  {  "idKlase": "3",  "nazivKlase": "premium ekonmska"  } |
| Format izlaznih parametara | application/json |

### 2.5.3 Funkcija ubaciKorisnika(data)

|  |  |
| --- | --- |
| Opis funkcije | Dodavanje novog korisnika |
| HTTP metoda | POST |
| URL | /ubaciKorisnika.json |
| URL parametri | (nema) |
| HTTP body parametri | JSON objekat sa atributima idKorisnik[int], name[varchar], username[varchar], password[varchar], admin[int], telefon[varchar],  Primer:  {  "idKorisnik":"1",  "name":"Milica Karajovic",  "usename":"mica",  "password":"mica"  "admin":"1"  "telefon":"0640518912"  } |
| Format HTTP body parametara | application/json |
| Izlazni parametri | JSON objekat sa jednim atributom, poruka[string].  Primer:  {  "poruka":"Uspesno!"  } |
| Format izlaznih parametara | application/json |

### 2.5.4 Funkcija unosLeta(data)

|  |  |
| --- | --- |
| Opis funkcije | Dodavanje novog leta |
| HTTP metoda | POST |
| URL | /unosLeta.json |
| URL parametri | (nema) |
| HTTP body parametri | JSON objekat sa atributima id[int], mestoOd[varChar], mestoDo[varChar], klasa[int], cena[double]  Primer:  {  "id":"2",  "mestoOd":"Berlin",  "mestoDo":"Budimpesta",  "klasa":"3"  "cena":"3589"  } |
| Format HTTP body parametara | application/json |
| Izlazni parametri | JSON objekat sa jednim atributom, poruka[string].  Primer:  {  "poruka":"Uspesno!"  } |
| Format izlaznih parametara | application/json |

# 3. Korišćene tehnologije

Kod izrade sajta korišćene su tehnologije poput:

* PHPa,
* HTML-a,
* CSS-a,
* Java Script-a,
* Bootstrapa i
* AJAXa

U par rečenica, prokomentarisaćemo svaku od navedenih tehnologija.

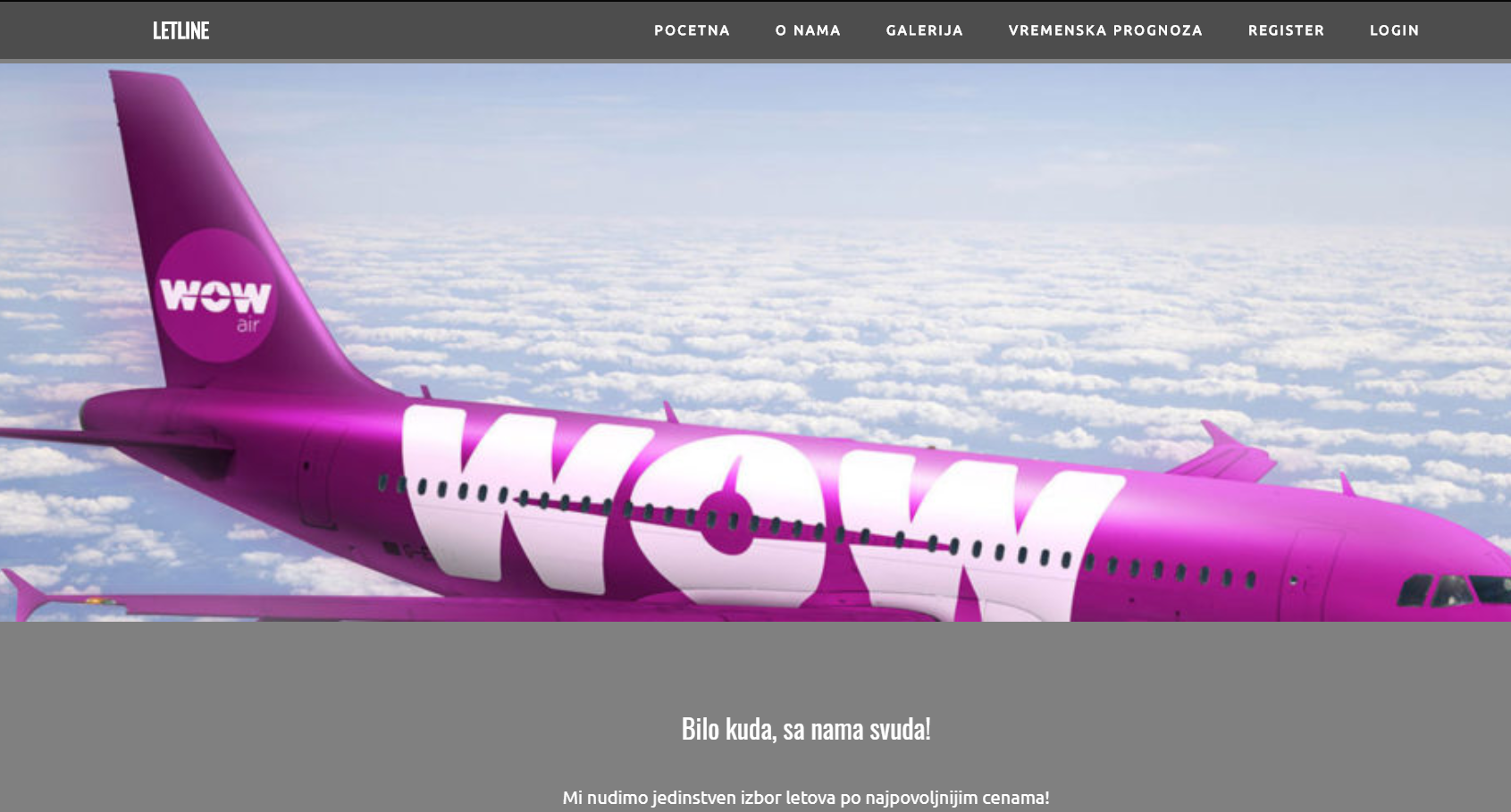
1. **PHP**je open source jezik koji se koristi za razvoj dinamičkog web sadržaja i izvodi se na strani servera. PHP se koristi za upravaljanje dinamičkim sadržajem, praćenje sesija, takođe tu je i interakcija se velikim brojem relacionih baza podatka, konkretno PHP vam omugućava da napravite krajnje e-commerce sajtove. Rasmus Lerdorf je izbacio prvu verziju PHP-a davne 1994. godine. PHP je stekao popularnost zbog svoje jednostavnosti i sintakse nasleđene iz programskog jezika C.
2. **HTML** je akronim za  **HyperText Markup Language** i predstavlja standardizovan jezik koji se koristi prilikom struktuiranja tekstova, medija i ugradjenih objekata u web stranice. Prikaz hiperteksta omogućava web pretraživač. Prvi HTML se spominje 1991, a originalna verzija je izašla juna 1993. i kreirana je od strane Tim Berners-Lee. Sam jezik sastoji se od tagova i teksta koji omogućavaju da internet pretraživač prepozna strukturu dokumenta i ispravno je prikaže korisniku.
3. **CSS -Cascading HTML StyleSheets**je jezik za formatiranje pomoću kog se definiše izgled samih elemenata web stranice. Konkretno HTML je služio da definiše kompletan izgled, strukturu i sadržaj web strnice, i to je činio sve do HTML verzije 4.0. Od ove verzije uveden je CSS koji bi definisao konretan izgled, dok je HTML ostao u funkciji definisanja strukture i sadržaja.
4. **JavaScript** je programski jezik HTML-a i web-a. Bitan deo je web aplikacija jer omogućava iteraktivnost aplikacija. Java Script je skriptni jezik, odnosno sastoji se od niza komandi koje se očitavaju u interepertoaru, bez prethodnog kompajliranja i podržava objektno-orjentisano programiranje.
5. **Boostrap**- Bootstrap je interfejs između serverske strane i korisnika, što ga čini front-end frejmvorkom. Trenutno je najpopularniji frejmvork za izradu web sajtova i web aplikacija, ispred Zurb Foundation-a. Sadrži HML i CSS šablone za dizajn, kao i JavaSCript ekstenzije. Jedna od najvažnijih uloga jeste omogućavanje responzivnosti sajta.
6. **AJAX** je akronim za**Asynchronous JavaScript And XML.** AJAX predstavlja web razvojnu tehniku za kreiranje interaktivnih web sadržaja. AJAX tehnologija se zasniva na razmeni malih količina podataka sa serverom, kako korisnik ne bi stalno morao da osvežava stranicu. Na taj način se omogućava potpuna interaktivnost, efikasnost i poboljšava funkcionisanje stranice uopšte. AJAX nije tehnologija sam po sebi, već termin koji se odnosi na korišćenje grupe tehnologija.
7. **REST web servisi:** REST (Representational State Transfer) predstavlja stil arhitekture aplikacije, koji se zasniva na često korišćenim tehnologijama i protokolima, od kojih je najvažniji HTTP u kombinaciji sa URL-om. Koristi osnovne HTTP-ove metode kao što su GET,POST,PUT,DELETE. Osnovni koncept REST-a je resurs, koji može biti informacija, dokument, slika, itd. Svaki resurs ima svoj identifikator, koji je URL, pomoću koga se pristupa i manipuliše tim resursom. Pri kreiranju sopstvenog REST servisa korišćen je framework Flight. Takođe korišćeni sui dostupni API-ji za prikazivanje vremenske prognoze i generisanje lozinke pri registraciji novog korisnika

# 4.Uputstvo za korisnike

Učitavanjem aplikacije, najpre nam se prikazuje početna strana. Ona sadrži meni, odakle korisnik može da izabere preostale stranice naše aplikacije, medju kojima su:

* POČETNA,
* O NAMA,
* GALERIJA,
* VREMENSKA PROGNOZA,
* REZERVISI LET,
* ADMINISTRATOR,
* ODJAVA

Kartica O NAMA nudi prikaz svih dodatnih informacija ukoliko se neko od naših potencijalnih korisnika odluči da nama ukaže poverenje. Kartica GALERIJA sadrži vizuelni prikaz naših aviona. U kartici VREMENSKA PROGONOZA dobićete informacije o tome kakvo nas vreme očekuje u Beogradu u narednih 10 dana. Ukoliko se odlučite da preko nas rezervišete karte za vase omiljene destinacije ono što je neophodno je da se najpre registrujete na našu aplikaciju u kartici REGISTER. Kada se jednom registrujete na našu aplikaciju, svaki sledeći put našoj ponudi moći ćete da pristupite tako što se ulogujete na kratici LOGIN.



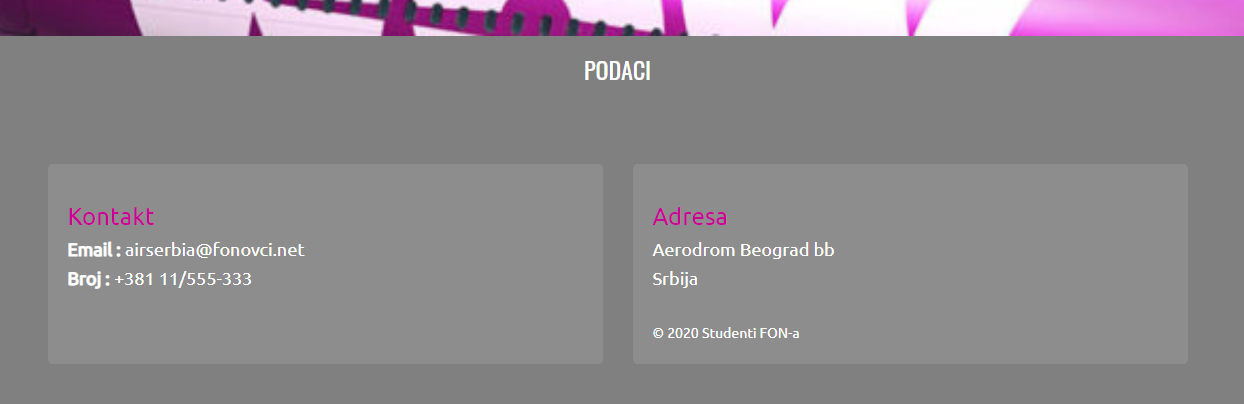
Slika 21 - Prikaz početne stranice HOME

U okviru početne stranice imamo još i ponudu trenutno dostupnih letova.



Slika 22 – Ponuda letova

Kao i osnovne podatke vezane za našu kompaniju:



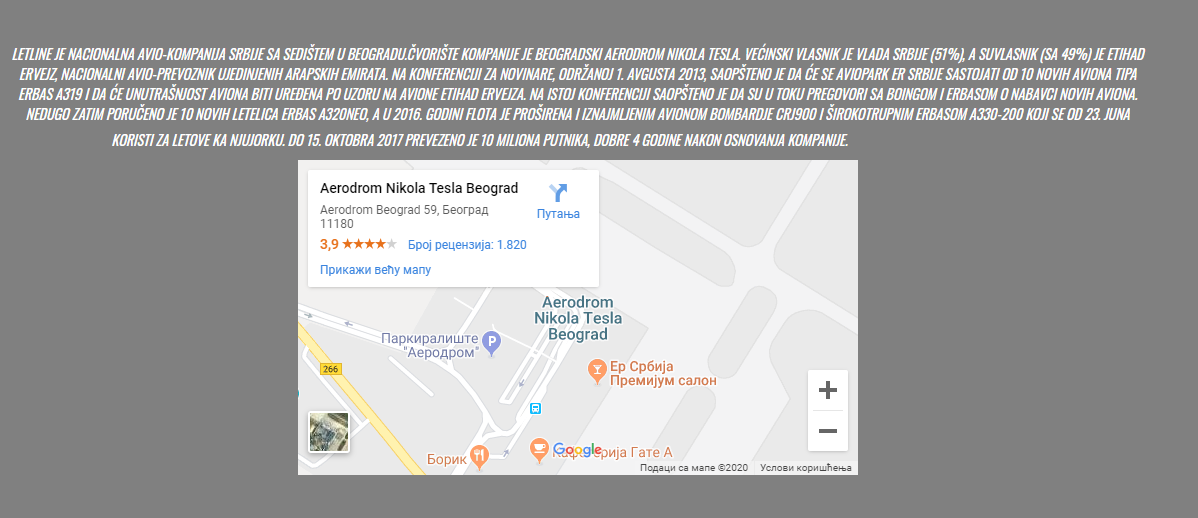
Slika 23– Osnovni podaci

Odabirom kartice O NAMA, na ekranu nam se pored već objašnjenog footera i ponude prikazati i dodatne informacije.

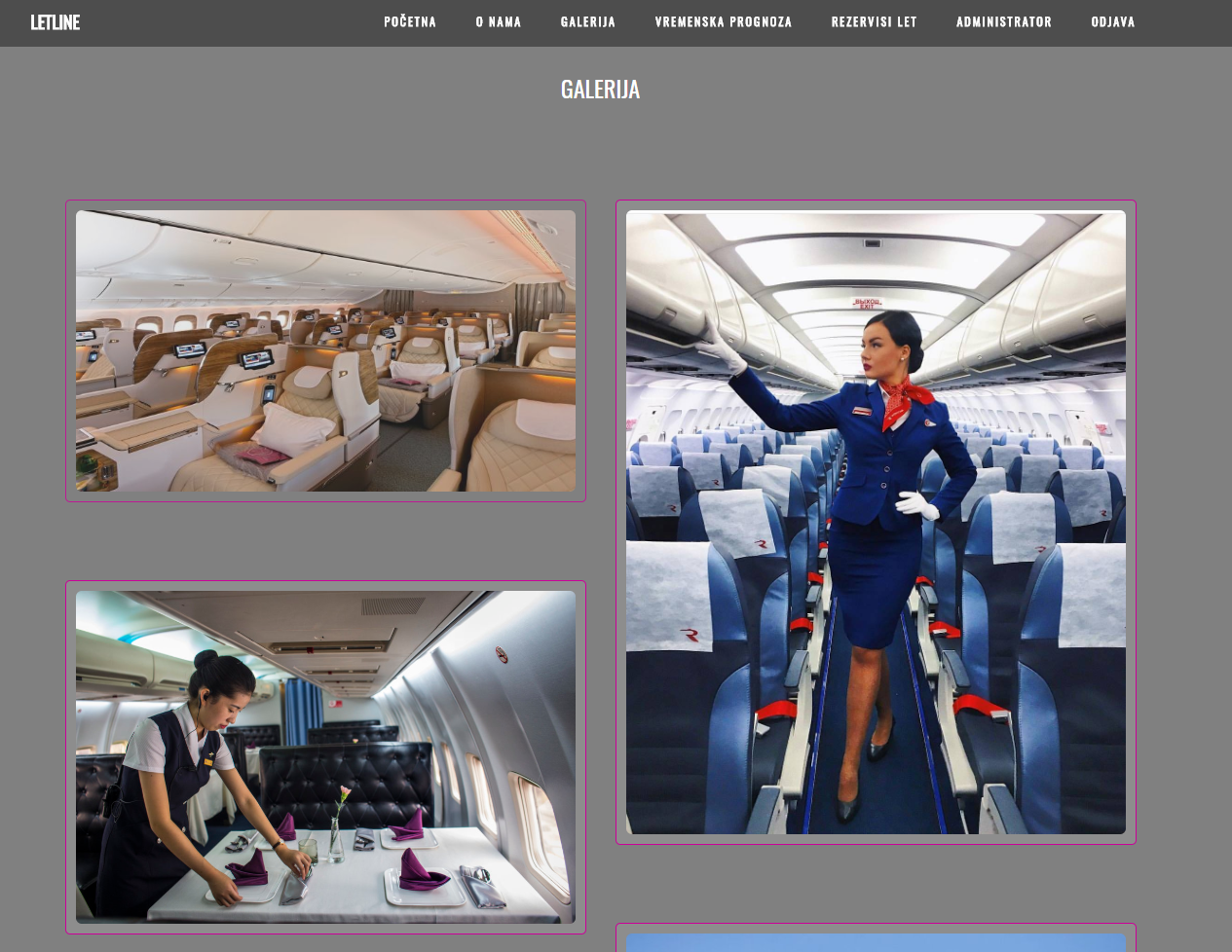


Slika 24 – Prikaz stranice O NAMA

Klikom na dugme “Klikni za prikaz mape aerodroma” dobijate tačnu lokaciju naše kompanije.

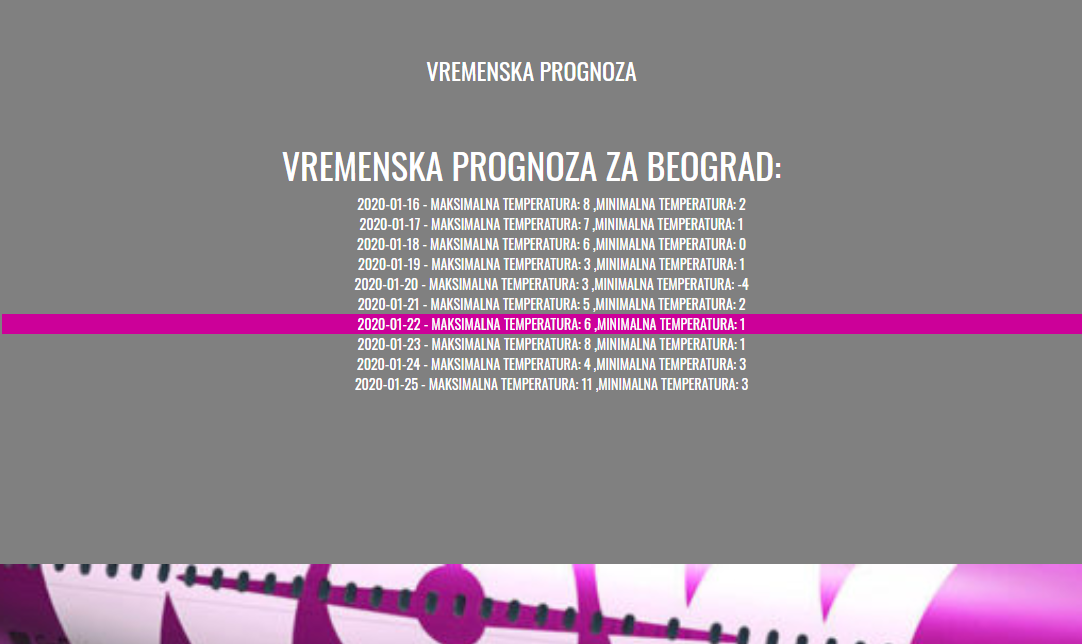


Klikom na GALERIJA u okviru menija dobijamo izgled prozorao kao na slici:



Slika 25 – Prikaz stranice GALERIJA

VREMENSKA PROGNOZA u okviru menija prikazuje vremensku prognozu za Beograd u narednih 10 dana.

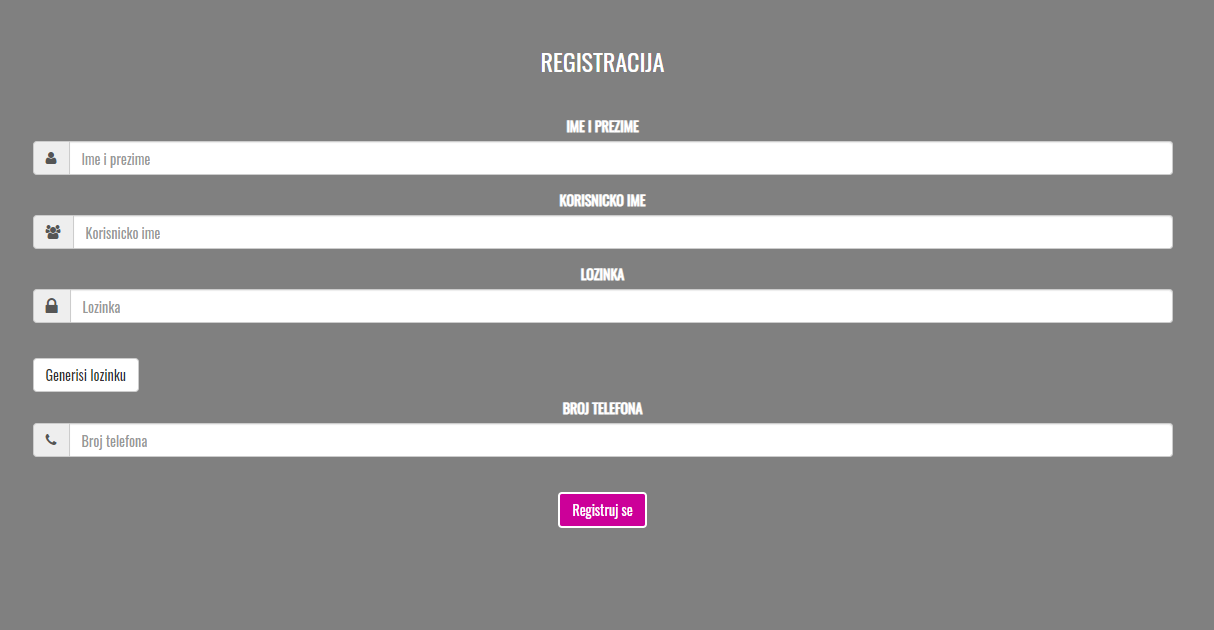


Slika 26- Prikaz stranice VREMENSKE PROGNOZE

Kartica REZERVISI LET omogućava novim korisnicima da se prijave i pamti ih.

Korisnik unosi svoje ime i prezime, korisnicko ime i šifru koje će koristiti za logovanje. Korisnik može izabrati svoju šifru ili kliknuti na dugme *Generiši lozinku* koja im kreira novu lozinku.

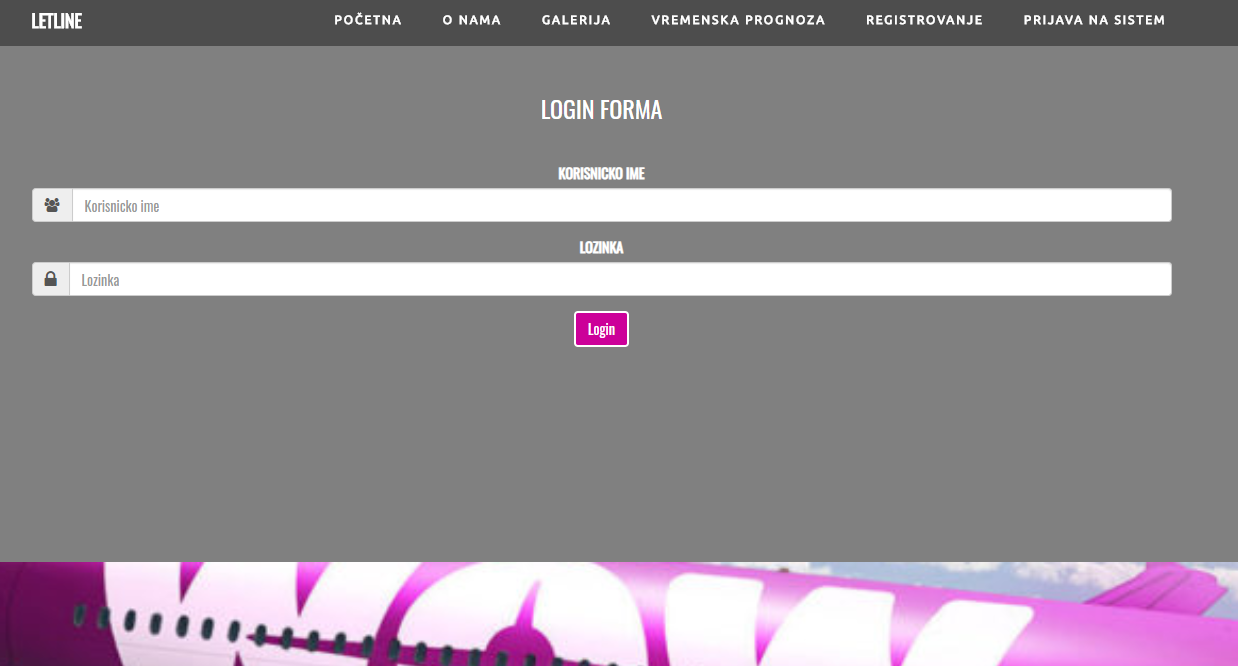
Klikom na dugme *Registruj se* pamti se novi korisnik.



Slika 27 – Prikaz strnice REGISTROVANJE

Kartica PRIJAVA NA SISTEM omogućava korisnicima koji su u bazi da se prijave, bilo da su administratori ili obični korisnici.

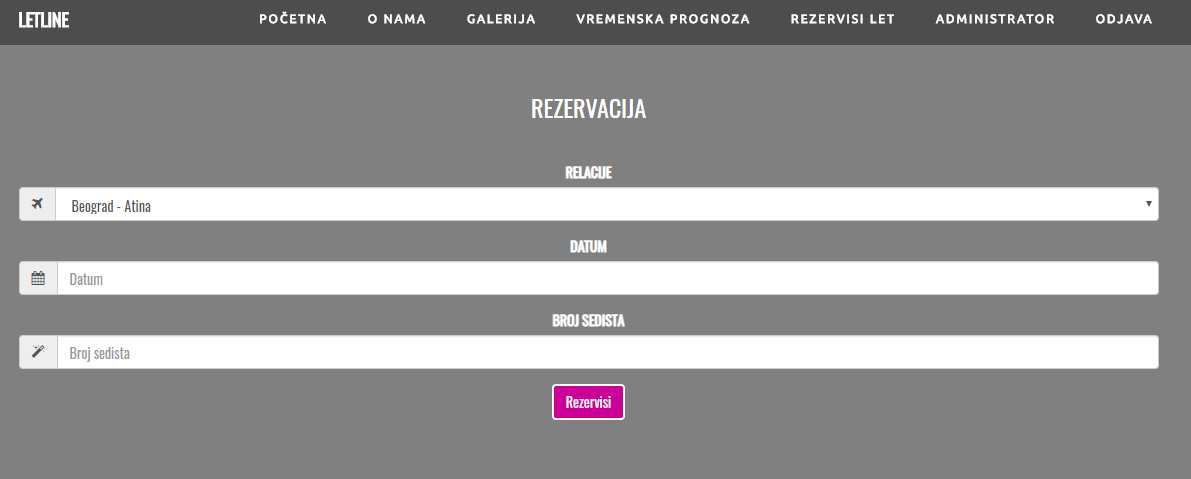
Potrebno je upisati korisničko ime i šifru i kliknuti dugme *Login*.



Slika 28 – Prikaz stranice PRIJAVA NA SISTEM

Ulogovanom korisniku se prikazuju kartica ODJAVA za odjavljivanje, nakon čega se ponovo prikazuje početna strana,i kartica REZERVISI LET, koja omogućava kreiranje nove rezervacije. Ukoliko je korisnik i administrator, prikazuje mu se i kartica ADMINISTRATOR.

Ako korisnik želi da rezerviše, potrebno je da iz padajuće liste izabere mesto poletanja i mesto sletanja, datum i broj sedišta. Klikom na dugme ***Rezervisi*** pamti se nova rezervacija.

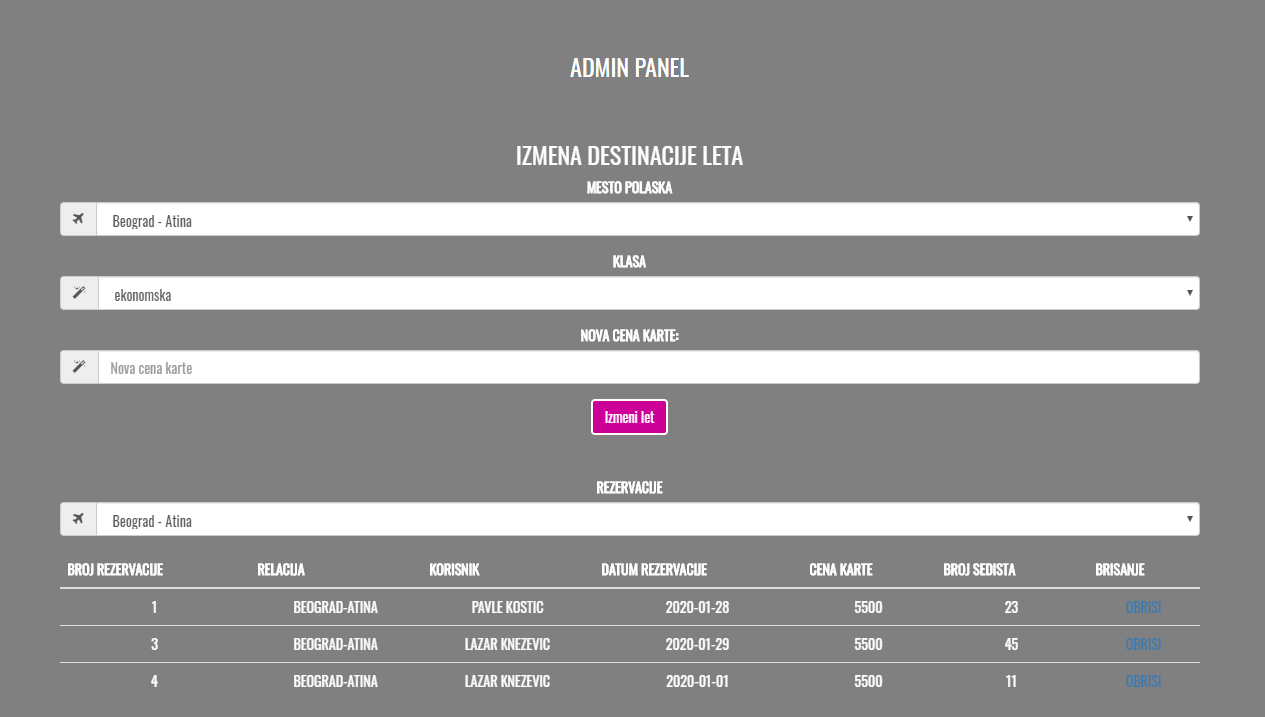


Slika 30 – Prikaz stranice REZERVACIJA LETA

Kartica ADMINISTRATOR omogućava administratorima da izmene postojeću destinaciju za neki let, prikaz svih rezervacija za jedan let, unos nove slike u galeriju, unos novog leta i prikaz grafika rezervisanosti za određenu klasu.

U slučaju da želimo da izmenimo destinaciju, administrator bira iz padajuće liste mesto poletanja i mesto sletanja, a zatim unosi klasu i novu cenu karte.

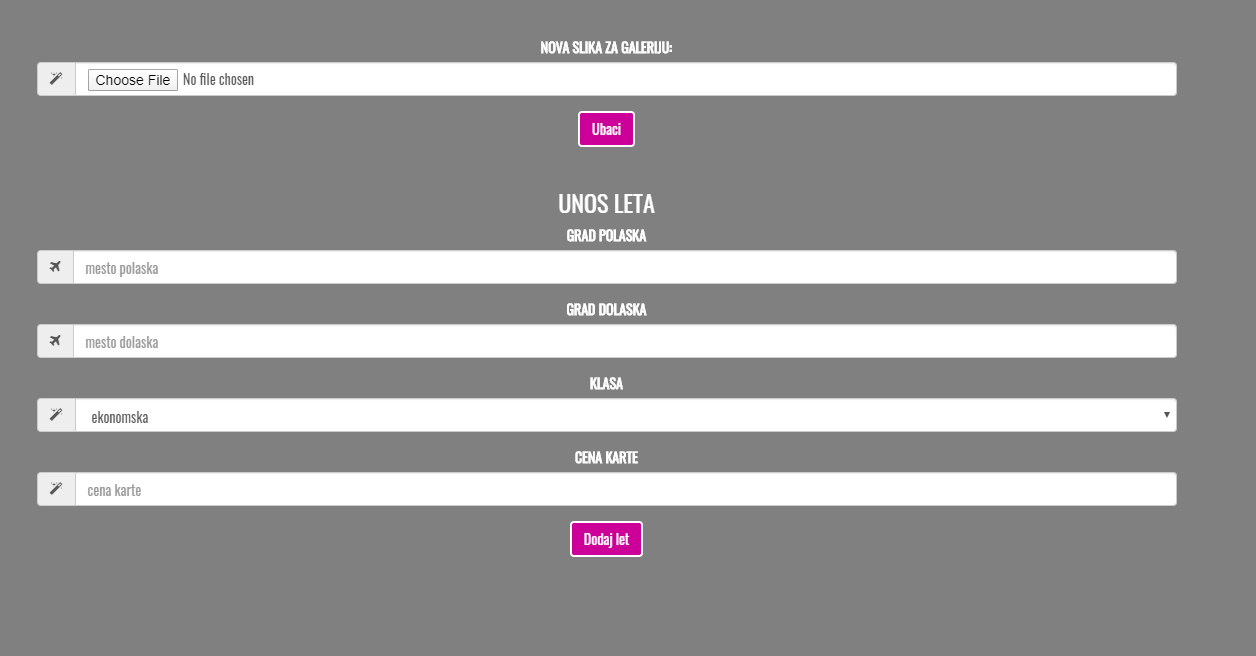
Klikom na dugme ***Izmeni*** *leta* pamti se nova destinacija.



Slika 31- Prikaz strnaice IZEMNA DESTINACIJE LETA



Slika 33 – Prikaz svih REZERVACIJA



Slika 34 – Prikaz preostalih funkcija sa ADMINISTRATORA

Koristeći rezervacije dobijamo grafik broja rezervacija po klasama:



Slika 35- Prikaz grafika - Broj rezervacija po klasi

# 5. Delovi koda seminarskog rada “LetLine”

## 5.1 Prikaz svih letova

Za prikaz letova korišćeni su PHP fajlovi:

* index.php koji prikazuje podatke,
* init.php koji omogućava konekciju na bazu odakle se čitaju podaci i
* letClass.php koji predstavlja klasu i sadrži atribute leta i funkcije.

<?php

include("letClass.php");

$let = new Let($mysqli);

$let->sviLetovi();

$result = $let->getResult();

if(count($result) >0) {

?>

<table class="table">

<thead>

<tr>

<th class="text-center">ID</th>

<th class="text-center">Mesto polaska</th>

<th class="text-center">Mesto dolaska</th>

<th class="text-center">Klasa</th>

<th class="text-center">Cena</th>

</tr>

</thead>

<tbody>

<?php

foreach ($result as $red ) {

$id = $red['id'];

$mestoOd = $red['mestoOd'];

$mestoDo= $red['mestoDo'];

$klasa= $red['nazivKlase'];

$cena= $red['cena'];

?>

<tr style="color: #fff; background: black;">

<td><?php echo $id; ?></td>

<td><?php echo $mestoOd; ?></td>

<td><?php echo $mestoDo; ?></td>

<td><?php echo $klasa; ?></td>

<td><?php echo $cena; ?></td>

</tr>

<?php

}

?>

* U ovom delu koda smo koristili DataTables za prikaz podataka u tabeli i pretragu.
* DataTables koristi AJAX tehnlogiju

<script type="text/javascript" src="https://cdn.datatables.net/v/dt/dt-1.10.16/datatables.min.js"></script>

<script>

$(document).ready(function() {

$('.table').DataTable();

} );

</script>

</body>

* init.php fajl za konekciju i pokretanje sesije

<?php

error\_reporting(E\_ALL | E\_STRICT);

ini\_set("display\_errors", 0);

ini\_set("log\_errors", 1);

ini\_set("error\_log", "logs.log");

//instace of mysqli

$mysqli = mysqli\_connect("localhost", "root", "", "letovi");

//sesija korisnik sistema, inicijalizovanje user sesije

session\_start();

if(!isset($\_SESSION['user'])){

$\_SESSION['user'] = array();

}

?>

* **letClass.php**, za prikaz svih letova poziva se funkcija sviLetovi()

<?php

class Let {

private $conn;

private $result;

public function \_\_construct($conn) {

$this->conn = $conn;

}

public function getResult(){

return $this->result;

}

public function setResult($res){

$this->result = $res;

}

public function sviLetovi() {

$curl\_zahtev = curl\_init("http://localhost/rezervacije/api/letovi.json");

curl\_setopt($curl\_zahtev, CURLOPT\_RETURNTRANSFER, 1);

$curl\_odgovor = curl\_exec($curl\_zahtev);

$json\_objekat=json\_decode($curl\_odgovor, true);

curl\_close($curl\_zahtev);

$this->setResult($json\_objekat);

}

public function sveKlase() {

$curl\_zahtev = curl\_init("http://localhost/rezervacije/api/klase.json");

curl\_setopt($curl\_zahtev, CURLOPT\_RETURNTRANSFER, 1);

$curl\_odgovor = curl\_exec($curl\_zahtev);

$json\_objekat=json\_decode($curl\_odgovor, true);

curl\_close($curl\_zahtev);

$this->setResult($json\_objekat);

}

public function allReservationBySearch($id) {

$id = mysqli\_real\_escape\_string($this->conn,$id);

$q = mysqli\_query($this->conn, "SELECT \* FROM rezervacija as r join let as l on (r.let=l.id ) join klasa as k on (l.klasa=k.idKlase) join korisnik as u on (u.idKorisnik=r.korisnik) where r.let = $id");

$this->setResult($q);

}

public function rezervacija($letId,$datum,$brSedista) {

$letId = mysqli\_real\_escape\_string($this->conn,$letId);

$datum = mysqli\_real\_escape\_string($this->conn,$datum);

$brSedista = mysqli\_real\_escape\_string($this->conn,$brSedista);

$userID=$\_SESSION["user"]["idKorisnik"];

$timestamp = date('Y-m-d H:i:s', strtotime($datum));

if($brSedista<0 or $brSedista>60)

{

$this->setResult(false);

return;

}

$sql = "INSERT INTO rezervacija (datum,let,korisnik,brojSedista) VALUES ('$timestamp',$letId,$userID,$brSedista)";

if(mysqli\_query($this->conn, $sql)){

$this->setResult(true);

}else{

$this->setResult(false);

};

}

public function unosLeta($gradOd,$gradDo,$klasa,$cena) {

$gradOd = mysqli\_real\_escape\_string($this->conn,$gradOd);

$gradDo = mysqli\_real\_escape\_string($this->conn,$gradDo);

$klasa = mysqli\_real\_escape\_string($this->conn,$klasa);

$cena = mysqli\_real\_escape\_string($this->conn,$cena);

$data = Array (

"gradOd" => $gradOd,

"gradDo" => $gradDo,

);

$zaSlanje = json\_encode($data);

$curl\_zahtev = curl\_init("http://localhost/rezervacije/api/unosLeta.json");

curl\_setopt($curl\_zahtev, CURLOPT\_POST, TRUE);

curl\_setopt($curl\_zahtev, CURLOPT\_POSTFIELDS, $zaSlanje);

curl\_setopt($curl\_zahtev, CURLOPT\_RETURNTRANSFER, 1);

$curl\_odgovor = curl\_exec($curl\_zahtev);

$curl\_odgovor = curl\_exec($curl\_zahtev);

$json\_objekat=json\_decode($curl\_odgovor, true);

curl\_close($curl\_zahtev);

if($json\_objekat == "Uspesno!") {

$this->setResult(true);

}

else {

$this->setResult(false);

}

}

public function izmenaLeta($mestoOd,$klasa,$cena) {

$mestoOd = mysqli\_real\_escape\_string($this->conn,$mestoOd);

$mestoDo = mysqli\_real\_escape\_string($this->conn,$mestoDo);

$klasa = mysqli\_real\_escape\_string($this->conn,$klasa);

$cena = mysqli\_real\_escape\_string($this->conn,$cena);

if(mysqli\_query($this->conn, "UPDATE let SET klasa=$klasa,cena='$cena' WHERE id=$mestoOd")){

$this->setResult(true);

}else{

$this->setResult(false);

};

}

}

?>

### 

### Prikaz vremenske prognoze

Za prikaz vremenske prognoze, kod koji smo pisali nalazi se u fajlu prognoza.php, a sam fajl koristi javni API. U nastavku **sledi deo koda prognoza.php koji omogućava korišćenje API-ja**.

<script>

$( document ).ready(function() {

$.getJSON('http://api.worldweatheronline.com/premium/v1/weather.ashx?key=611b337e360348da9ff120030191811&q=Belgrade&num\_of\_days=10&format=json')

.done(function(data) {

var text = '<ul>';

$.each(data.data.weather, function(i, val) {

text += '<li>'+val.date+' - maksimalna temperatura: '+val.maxtempC+' ,minimalna temperatura: '+val.mintempC+'</li>';

});

text+='</ul>';

$('#podaci').html(text);

})

.error(function(err) { alert('Greska');

});

});

</script>

*Izmena podataka o letu*

Za izmenu podataka o letu potrebni su:

* admin.php i
* letClass.php.

Admin.php fajl prikuplja podatke i kreira novu klasu leta koja omogućava pamćenje izmene.

Iz letClass.php fajla pozivaju se funkcije sviLetovi() i izmenaLeta(mestoOd, mestoDo, klasa, novaCena).

**Delovi koda admin.php za izmenu podataka o letu**

<select class="form-control" name="letSearch" id="letSearch" onchange="search()">

<?php

$let = new Let($mysqli);

$let->sviLetovi();

$result = $let->getResult();

foreach ($result as $red ) {

$id = $red['id'];

$mestoOd = $red['mestoOd'];

$mestoDo = $red['mestoDo'];

?>

<option value="<?php echo $id;?>"><?php echo $mestoOd.' - '.$mestoDo;?></option>

<?php

}

?>

</select>

if(isset($\_POST['izmeni'])) {

$let = new Let($mysqli);

$let->izmenaImena(trim($\_POST['mestoOd']),trim($\_POST['type']),trim($\_POST['novacena']));

if($let->getResult()){

$msg="Uspesno izmena leta";

}else{

$msg="Neuspesna izmena leta , takav let ne postoji u bazi.";

}

}

### Unos novog leta

Za unos leta potrebni su takođe:

* admin.php i
* letClass.php fajl.

Iz letClass.php fajla se poziva funkcija unosLeta(gradOd, gradDo, tip, cena)

**Deo admin.php fajla koji omogućava unos novog leta**

* if(isset($\_POST['unosLeta'])) {

$let = new Let($mysqli);

$let->unosLeta(trim($\_POST['gradOd']),trim($\_POST['gradDo']),trim($\_POST['type']),trim($\_POST['cena']));

if($let->getResult()){

$msg="Uspesno unet novi let ";

}else{

$msg="Neuspesno unosenje novog leta ";

}

}

### Unos nove slike u galeriju

Za unos nove slike potrebni sun am sledeći fajlovi admin.php, novaSlikaGalerija.php i fajl koji služi za prikaz slika.

**Deo koda admin.php**

<form name="galerija" method="post" action="novaSlikaGalerija.php" enctype="multipart/form-data">

<div class="form-group">

<label for="file" class="cols-sm-2 control-label">Nova slika za galeriju:</label>

<div class="cols-sm-10">

<div class="input-group">

<span class="input-group-addon"><i class="fa fa-magic fa" aria-hidden="true"></i></span>

<input type="file" class="form-control" name="file" placeholder="Ubacite sliku za galeriju"/>

</div>

</div>

</div>

<div class="form-group ">

<input type="submit" name="file" class="btn btn-danger btn-lg " value="Ubaci">

</div>

</form>

**novaSlikaGalerija.php**

<?php

ini\_set('display\_errors', 1);

include("init.php");

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$name = $\_FILES['file']['name'];

$tmpName = $\_FILES['file']['tmp\_name'];

$error = $\_FILES['file']['error'];

$size = $\_FILES['file']['size'];

$ext = strtolower(pathinfo($name, PATHINFO\_EXTENSION));

$targetPath = dirname( \_\_FILE\_\_ ) . DIRECTORY\_SEPARATOR. 'assets'. DIRECTORY\_SEPARATOR.'galerija' . DIRECTORY\_SEPARATOR. $name;

move\_uploaded\_file($tmpName,$targetPath);

header( 'Location: galerija.php' ) ;

exit;

}

?>

### 

### Prikaz grafika

Grafik treba procentualno da prikaže broj rezervacija i popunjenost svakog leta pojedinačno. Za to su nam potrebni fajlovi:

* admin.php i
* podaciGrafik.php i
* ajax tehnologija.

**U nastavku sledi deo koda iz fajla admin.php**

<script>

$( function() {

$( "#datepicker" ).datepicker({ dateFormat: 'yy-mm-dd' });

} );

</script>

<script type="text/javascript" src="https://www.google.com/jsapi"></script>

<script type="text/javascript">

google.load('visualization', '1', {'packages':['corechart']});

google.setOnLoadCallback(grafik);

function grafik() {

var jsonData = $.ajax({

url: "podaciGrafik.php",

dataType:"json",

async: false

}).responseText;

var data = new google.visualization.DataTable(jsonData);

var options = {'title':'Broj rezervacija po klasi',

backgroundColor: { fill:'transparent' },

titleTextStyle: {

textAlign: 'center',

color: 'white',

fontSize: 18},

'width':800,

'height':500,

is3D:true,

legend: {

textStyle: {

color: 'white'

}

},

};

var chart = new google.visualization.PieChart(document.getElementById('divGrafik'));

function selectHandler() {

var selectedItem = chart.getSelection()[0];

if (selectedItem) {

alert( data.getValue(selectedItem.row,0));

}

}

google.visualization.events.addListener(chart, 'select', selectHandler);

chart.draw(data, options);

}

</script>

**Kod iz fajla podaciGrafik.php**

<?php

include("init.php");

$array['cols'][] = array('label' => 'Kalsa','type' => 'string');

$array['cols'][] = array('label' => 'Broj letova po klasi', 'type' => 'number');

$sql = "SELECT nazivKlase, COUNT(l.klasa) as broj FROM let l JOIN klasa k ON (l.klasa=k.idKlase) join rezervacija r on (r.let=l.id) GROUP BY k.idKlase";

if (!$q=$mysqli->query($sql)){

echo '{"greska":"Nastala je greška pri izvršavanju upita."}';

exit();

} else {

if ($q->num\_rows>0){

$niz[] = array();

while ($red=$q->fetch\_object()){

$array['rows'][] = array('c' => array( array('v'=>$red->nazivKlase),array('v'=>(int)$red->broj)) );

}

$niz\_json = json\_encode ($array);

print ($niz\_json);

} else {

//ako nema rezultata u bazi

echo '{"greska":"Nema rezultata."}';

}

}

?>

### Registracija

Registracija nam pruža mogućnost da upamtimo novog korisnika. Fajlovi koji su nam potrebni su :

* register.php,
* userClass.php koji predstavlja klasu User-a i
* lozinka.php koji omogućava korišćenje javnog API-ja za generisanje lozinke.

**U nastavku sledi deo fajla register.php**

$msg = '';

if(isset($\_POST['register'])) {

require('userClass.php');

$user = new User($mysqli);

$user->registracijaKorisnika(trim($\_POST['name']),trim($\_POST['username']),trim($\_POST['password']),trim($\_POST['telephone']));

if($user->getResult()){

$msg="Uspesno registrovan korisnik";

}else{

$msg="Neuspesna registracija korisnika ";

}

}

//generisanje lozinke

<script>

$(document).ready(function() {

$('#generisiLozinku').click(function() {

$.ajax({

url: 'lozinka.php',

dataType: 'json',

success: function(json) {

$('#password').val(json);

},

error: function(json) {

console.log(json,password +' error');

}

});

});

});

</script>

**Kod iz fajla lozinka.php**

<?php

$ch = curl\_init();

curl\_setopt($ch, CURLOPT\_URL, "https://helloacm.com/api/random/?n=10");

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

curl\_setopt($ch, CURLOPT\_HEADER, 0);

curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

$output = curl\_exec($ch);

$password ="";

for($i=1;$i<9;$i++){

$password[$i]=$output[$i];

}

curl\_close($ch);

echo json\_encode($password);

?>

**Kod iz fajla userClass.php**

<?php

class User {

private $conn;

private $result;

public function \_\_construct($conn) {

$this->conn = $conn;

}

public function getResult(){

return $this->result;

}

public function setResult($res){

$this->result = $res;

}

public function registracijaKorisnika($name,$username,$password,$telephone) {

$name = mysqli\_real\_escape\_string($this->conn,$name);

$username = mysqli\_real\_escape\_string($this->conn,$username);

$password = mysqli\_real\_escape\_string($this->conn,$password);

$telephone = mysqli\_real\_escape\_string($this->conn,$telephone);

$data = Array (

"name" => $ime,

"username" => $username,

"password" => $password,

"telephone"=> $telephone

);

$zaSlanje = json\_encode($data);

$curl\_zahtev = curl\_init("http://localhost/rezervcije/api/ubaciKorisnika.json");

curl\_setopt($curl\_zahtev, CURLOPT\_POST, TRUE);

curl\_setopt($curl\_zahtev, CURLOPT\_POSTFIELDS, $zaSlanje);

curl\_setopt($curl\_zahtev, CURLOPT\_RETURNTRANSFER, 1);

$curl\_odgovor = curl\_exec($curl\_zahtev);

$json\_objekat=json\_decode($curl\_odgovor, true);

curl\_close($curl\_zahtev);

$this->setResult($json\_objekat);

}

public function login($username,$password) {

$username = mysqli\_real\_escape\_string($this->conn,$username);

$password = mysqli\_real\_escape\_string($this->conn,$password);

$q = mysqli\_query($this->conn, "select \* from korisnik where username='$username' and password='$password' limit 1");

if(mysqli\_num\_rows($q)>0){

while($red = mysqli\_fetch\_assoc($q)) {

$\_SESSION['user'] = $red;

}

$this->setResult(true);

}else{

$this->setResult(false);

};

}

}

?>

### Login

Omogućava prijavu korisnika koji postoje u bazi. Fajlovi koji su nam potrebni su:

* login.php i
* userClass.php koji poziva funkciju login(username, password).

**Deo koda iz fajla login.php**

$msg = '';

if(isset($\_POST['login'])) {

require('userClass.php');

$user = new User($mysqli);

$user->login(trim($\_POST['username']),trim($\_POST['password']));

if($user->getResult()){

$msg="Uspesno ulogovan korisnik";

}else{

$msg="Neuspesan login korisnika ";

}

}

### Rezervacije

Rezervacije omogućavaju korisnicima kreiranje nove rezervacije. Potrebni fajlovi su:

* rezervacija.php i
* letClass.php koji poziva funkcije sviLetovi() za prikaz ponuđenih letova i sveKlase() za prikaz svih klasa ponudi.

Na kraju poziva se funkcija iz letClass.php rezervacija(letId, datum i brSedista) koja pamti novu rezervaciju.

**Kod iz fajla rezervacija.php**

<?php

include("init.php");

require('letClass.php');

$msg = '';

if(isset($\_POST['rezervisi'])) {

$let = new Let($mysqli);

$let->rezervacija(trim($\_POST['letId']),trim($\_POST['datum']),trim($\_POST['brSedista']));

if($let->getResult()){

$msg="Uspesno rezervisano";

}else{

$msg="Neuspesna rezervacija ! Broj sedista mora biti u opsegu 0-80";

}

}

?>

<!DOCTYPE html>

<html lang="en" class="no-js" >

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />

<meta name="description" content="" />

<meta name="author" content="" />

<title>Flightl booking</title>

<link href="assets/css/bootstrap.css" rel="stylesheet" />

<link href="assets/css/ionicons.css" rel="stylesheet" />

<link href="assets/css/font-awesome.css" rel="stylesheet" />

<link href="assets/js/source/jquery.fancybox.css" rel="stylesheet" />

<link href="assets/css/animations.min.css" rel="stylesheet" />

<link href="assets/css/style-solid-black.css" rel="stylesheet" />

<link href="assets/css/jquery-ui.css" rel="stylesheet" />

</head>

<body data-spy="scroll" data-target="#menu-section">

<?php

include("meni.php");

?>

<?php

include("slider.php");

?>

<!-- Glavna sekcija -->

<section>

<div class="container">

<div class="row text-center header">

<h3>Rezervacija</h3>

<hr />

<?php

if($msg!=''){

echo("<h3> ".$msg."</h3>");

}

?>

<div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">

<div class="services-wrapper">

<form name="rezervacija" method="post" action="">

<div class="form-group">

<label for="let" class="cols-sm-2 control-label"> Relacije </label>

<div class="cols-sm-10">

<div class="input-group">

<span class="input-group-addon"><i class="fa fa-plane fa" aria-hidden="true"></i></span>

<select class="form-control" name="letId" >

<?php

$let = new Let($mysqli);

$let->sviLetovi();

$result = $let->getResult();

foreach ($result as $red ) {

$id = $red['id'];

$mestoOd = $red['mestoOd']; $mestoDo = $red['mestoDo'];

?>

<option value="<?php echo $id;?>"><?php echo $mestoOd.' - '.$mestoDo;?></option>

<?php

}

?>

</select>

</div>

</div>

</div>

<div class="form-group">

<label for="date" class="cols-sm-2 control-label">Datum</label>

<div class="cols-sm-10">

<div class="input-group">

<span class="input-group-addon"><i class="fa fa-calendar fa" aria-hidden="true"></i></span>

<input id="datepicker" type="text" class="form-control" name="datum" id="datum" placeholder="Datum"/>

</div>

</div>

</div>

<div class="form-group">

<label for="drzava" class="cols-sm-2 control-label">Broj sedista</label>

<div class="cols-sm-10">

<div class="input-group">

<span class="input-group-addon"><i class="fa fa-magic fa" aria-hidden="true"></i></span>

<input type="text" class="form-control" name="brSedista" id="brSedista" placeholder="Broj sedista"/>

</div>

</div>

</div>

<div class="form-group ">

<input type="submit" name="rezervisi" class="btn btn-danger btn-lg " value="Rezervisi">

</div>

</form>

</div>

</div>

</div>

</div>

</section>

<?php

include("footer.php");

?>

<script src="assets/js/jquery-1.11.1.js"></script>

<script src="assets/js/bootstrap.js"></script>

<script src="assets/js/vegas/jquery.vegas.min.js"></script>

<script src="assets/js/jquery.easing.min.js"></script>

<script src="assets/js/source/jquery.fancybox.js"></script>

<script src="assets/js/jquery.isotope.js"></script>

<script src="assets/js/appear.min.js"></script>

<script src="assets/js/animations.min.js"></script>

<script src="assets/js/custom-solid.js"></script>

<script src="assets/js/jquery-ui.js"></script>

<script>

$( function() {

$( "#datepicker" ).datepicker({ dateFormat: 'yy-mm-dd' });

} );

</script>

</body>

</html>

### Prikaz svih rezervacija i brisanje rezervacije

U okviru prikaza svih rezervacija I brisanja imamo:

* admin.php,
* letClass.php fajl koji poziva funkciju allReservationBySearchId) koji pronalazi rezervacije za određeni let,
* generišiPodatkeSearch.php koji kreira klasu leta i prikuplja podatke i
* obriši.php koji omogućava asinhrono brisanje rezervacije.

**Deo koda iz fajla admin.php**

<script>

function search(){

var event = $("#letSearch").val();

$.ajax({

url: "generisiPodatkeSearch.php",

data: "id="+event,

success: function(result){

var text = '<table class="table"><thead><tr><th>Broj rezervacije</th><th>Relacija</th><th>Korisnik</th><th>Datum rezervacije</th><th>Cena karte</th><th>Broj sedista</th><th>Brisanje</th></tr></thead><tbody>';

$.each($.parseJSON(result), function(i, val) {

text += '<tr>';

text += '<td>'+val.rezervacijaID+'</td>';

text += '<td>'+val.mestoOd+'-'+val.mestoDo+'</td>';

text += '<td>'+val.name+'</td>';

text += '<td>'+val.datum+'</td>';

text += '<td>'+val.cena+'</td>';

text += '<td>'+val.brojSedista+'</td>';

text += '<td><a href="obrisi.php?id='+val.rezervacijaID+'">Obrisi</a></td>';

text += '</tr>';

});

text+='</tbody></table>';

$('#tabela').html(text);

}});

}

</script>

**Kod iz fajla generisiPodatke.php**

<?php

include("init.php");

require('letClass.php');

$id=$\_GET['id'];

$let = new Let($mysqli);

$let->allReservationBySearch($id);

$result = $let->getResult();

$niz = array();

$iterator = 0;

while($red = mysqli\_fetch\_assoc($result)) {

$niz[$iterator] = $red;

$iterator++;

}

echo(json\_encode($niz));

?>

**Kod iz fajla obrisi.php**

<?php

include("init.php");

$id=mysqli\_real\_escape\_string($mysqli,$\_GET["id"]);

if(mysqli\_query($mysqli, "delete from rezervacija where rezervacijaID=$id")){

header("Location: admin.php");

}else{

echo("Doslo je do greske prilikom brisanja");

};

?>

### Sopstveni servisi

**Kod iz fajla database.php**

<?php

class Database {

private $hostname = "localhost";

private $username = "root";

private $password = "";

private $dbname = "letovi";

private $dblink;

private $result = true;

private $records;

private $affectedRows;

function \_\_construct($dbname)

{

$this->$dbname = $dbname;

$this->Connect();

}

public function getResult()

{

return $this->result;

}

function \_\_destruct()

{

$this->dblink->close();

}

function Connect()

{

$this->dblink = new mysqli($this->hostname, $this->username, $this->password, $this->dbname);

if($this->dblink->connect\_errno)

{

printf("Konekcija neuspesna: %s\n", $mysqli->connect\_error);

exit();

}

$this->dblink->set\_charset("utf8");

}

function noviLet($data) {

$mysqli = new mysqli("localhost", "root", "", "letovi");

$mestoOd = mysqli\_real\_escape\_string($mysqli,$data["mestoOd"]);

$mestoDo = mysqli\_real\_escape\_string($mysqli,$data["mestoDo"]);

$klasa = mysqli\_real\_escape\_string($mysqli,$data["klasa"]);

$cena = mysqli\_real\_escape\_string($mysqli,$data["cena"]);

$sql = "INSERT INTO let (mestoOd,mestoDo,klasa,cena) VALUES ('$mestoOd','$mestoDo','$klasa','$cena')";

if($mysqli->query($sql))

{

$this ->result = true;

}

else

{

$this->result = false;

}

$mysqli->close();

}

function ubaciKorisnika($data) {

$mysqli = new mysqli("localhost", "root", "", "letovi");

$cols = '(name, username, password, admin, telephone)';

$name = mysqli\_real\_escape\_string($mysqli,$data['name']);

$username = mysqli\_real\_escape\_string($mysqli,$data['username']);

$password = mysqli\_real\_escape\_string($mysqli,$data['password']);

$telephone = mysqli\_real\_escape\_string($mysqli,$data['telephone']);

$query = "INSERT into user (name, username, password, admin, telefon) VALUES ('$name','$username','$password','0','$telephone')";

if($mysqli->query($query))

{

$this ->result = true;

}

else

{

$this->result = false;

}

$mysqli->close();

}

function sviLetovi() {

$mysqli = new mysqli("localhost", "root", "", "letovi");

$q = 'SELECT \* FROM let l join klasa k on (l.klasa=k.idKlase)';

$this ->result = $mysqli->query($q);

$mysqli->close();

}

function sveKlase() {

$mysqli = new mysqli("localhost", "root", "", "letovi");

$q = 'SELECT \* FROM klasa';

$this ->result = $mysqli->query($q);

$mysqli->close();

}

function ExecuteQuery($query)

{

if($this->result = $this->dblink->query($query)){

if (isset($this->result->num\_rows)) $this->records = $this->result->num\_rows;

if (isset($this->dblink->affected\_rows)) $this->affected = $this->dblink->affected\_rows;

return true;

}

else{

return false;

}

}

}

?>

**Kod iz fajla index.php**

<?php

require 'flight/Flight.php';

require 'jsonindent.php';

Flight::route('/', function(){

echo('Popis ruta: <br>');

echo('------------ <br>');

echo('GET rooms.json <br>');

echo('GET letovi.json <br>');

echo('POST ubaciKorisnika.json <br>');

echo('POST unosLeta.json <br>');

});

Flight::register('db', 'Database', array('niz'));

Flight::route('GET /klase.json', function()

{

header("Content-Type: application/json; charset=utf-8");

$db = Flight::db();

$db->sveKlase();

$niz = array();

$iterator = 0;

while ($red = $db->getResult()->fetch\_object())

{

$niz[$iterator] = $red;

$iterator += 1;

}

echo indent(json\_encode($niz));

});

Flight::route('GET /letovi.json', function()

{

header("Content-Type: application/json; charset=utf-8");

$db = Flight::db();

$db->sviLetovi();

$niz = array();

$iterator = 0;

while ($red = $db->getResult()->fetch\_object())

{

$niz[$iterator] = $red;

$iterator += 1;

}

echo indent(json\_encode($niz));

});

Flight::route('POST /ubaciKorisnika.json', function()

{

header("Content-Type: application/json; charset=utf-8");

$db = Flight::db();

$post\_data = file\_get\_contents('php://input');

$json\_data = json\_decode($post\_data,true);

$db->ubaciKorisnika($json\_data);

if($db->getResult())

{

$response = "Uspesno!";

}

else

{

$response = "Greska!";

}

echo indent(json\_encode($response));

});

Flight::route('POST /unosLeta.json', function()

{

header("Content-Type: application/json; charset=utf-8");

$db = Flight::db();

$post\_data = file\_get\_contents('php://input');

$json\_data = json\_decode($post\_data,true);

$db->noviLet($json\_data);

if($db->getResult())

{

$response = "Uspesno!";

}

else

{

$response = "Neuspesno";

}

echo indent(json\_encode($response));

});

Flight::start();

?>