

Fakulteta za elektrotehniko, računalništvo in informatiko

*Praktikum I*

1. *Letnik UNI (ITK)*

Mestni utrip



**Avtorji:**

Primož Stopar

Uroš Zagoranski **Mentor:**

Aljoša Sikošek Mitja Gradišnik

Nejc Vnuk

Junij 2018

VSEBINA

[Navodila naloge 3](#_Toc516746891)

[Entitetno relacijski Model 4](#_Toc516746892)

[Opis rešitve problema 5](#_Toc516746893)

[Glavne komponente, metode 7](#_Toc516746894)

[1. Index.jsp (Glavna stran) : 7](#_Toc516746895)

[2. Registracija : 7](#_Toc516746896)

[3. Strani, dostopne preko filtrov na glavni strani : 7](#_Toc516746897)

[4. Podrobnosti : 8](#_Toc516746898)

[5. Dodajanje dogodka : 8](#_Toc516746899)

[6. Moji dogodki : 8](#_Toc516746900)

[7. Urejanje dogodkov : 8](#_Toc516746901)

[Uporabniški vmesnik aplikacije 9](#_Toc516746902)

[Pomembnejši izseki kode 13](#_Toc516746903)

[Poročilo testiranja 16](#_Toc516746904)

KAZALO SLIK

[Slika 1: Navodila naloge 3](#_Toc516746872)

[Slika 2: Entitetno relacijski model 4](#_Toc516746873)

[Slika 3: Odstotki jezikov 5](#_Toc516746874)

[Slika 4: Lent 6](#_Toc516746875)

[Slika 5: Pozdravna stran Index.jsp 9](#_Toc516746876)

[Slika 6: Galerija filtriranih seznamov 9](#_Toc516746877)

[Slika 7: Dodajanje dogodka 10](#_Toc516746878)

[Slika 8: Urejanje željenega dogodka 10](#_Toc516746879)

[Slika 9: Aktualni dogodki 11](#_Toc516746880)

[Slika 10: Podrobnosti izbranega dogodka 11](#_Toc516746881)

[Slika 11: Galerija slik pri podrobnostih 12](#_Toc516746882)

[Slika 12: Zemljevid pri podobnostih 12](#_Toc516746883)

[Slika 13: Komentarji 13](#_Toc516746884)

[Slika 14: Controller Dodajanje dogodka 13](#_Toc516746885)

[Slika 15: Sortirano branje dogodka iz baze 14](#_Toc516746886)

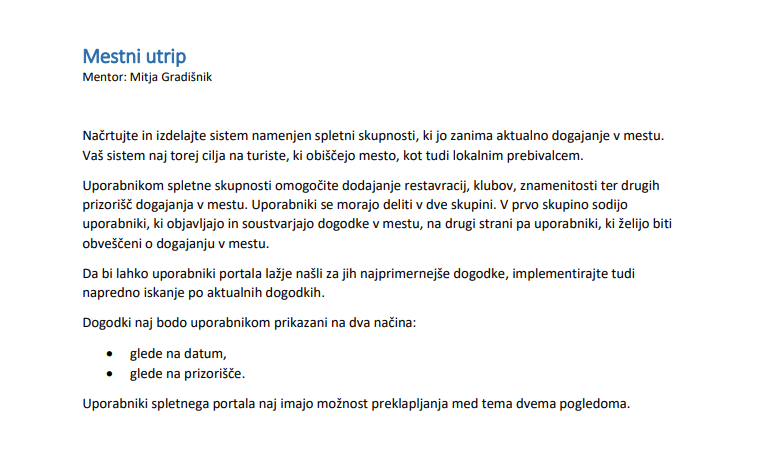
[Slika 16: jQuery pri registraciji 14](#_Toc516746887)

[Slika 17: JSTL 15](#_Toc516746888)

[Slika 18: Javanski razred 15](#_Toc516746889)

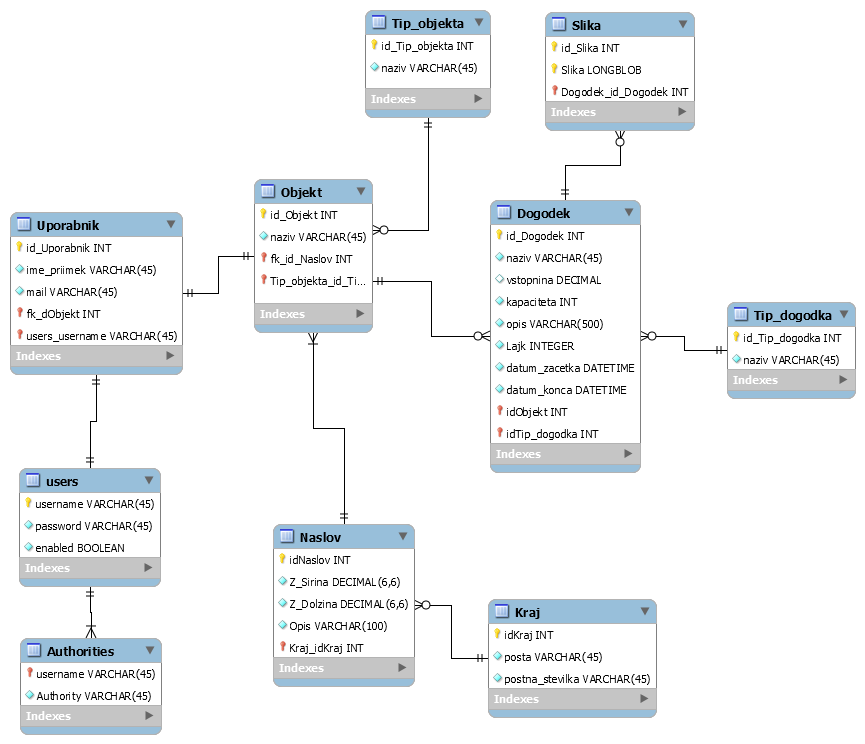
[Slika 19: Testiranje 16](#_Toc516746890)

# Navodila naloge

****

Slika 1: Navodila naloge

# Entitetno relacijski Model



Slika 2: Entitetno relacijski model

# Opis rešitve problema

Aplikacija je zasnovana tako, da je uporabniku kar se da prijazna. Na zgornjem delu ekrana je konsistentno oblikovan meni, v katerem se nahajajo pomembnejše komponente. Celotna stran je oblikovana z dodatkom Bootstrap-a, zaradi česar je prilagodljiva za vse naprave. Osnova aplikacije je podatkovna baza v MySQL-u, celotna rešitev pa temelji na frameworku SpringBoot. Za nemoteno delovanje aplikacije je potreben vtičnik Maven. Za prikaz podatkov smo uporabljali tehnologijo JSP, ki za delovanje uporablja jezik Java.

Naš izbor integriranega razvojnega okolja je IntelliJ Idea, ki nudi največ funkcionalnosti in je za naše pojme najbolj pregleden. Celotna spletna stran je po statistiki GitHuba sestavljena iz: Jave (83.2%), CSS (11.9%) in JavaScripta (4.9%). Torej smo v rešitev vključili večji del jezikov, ki smo jih spoznali tekom prvega letnika na smeri Informatika in tehnologije komuniciranja.



Slika 3: Odstotki jezikov

Arhitekturna zasnova aplikacije je razvidna iz E-R modela. Ključni entiteti sta torej dogodek in objekt. Primarno se spletna stran odpre na povezavi index.jsp, kjer so pregledno podane splošne informacije o problemu, naši podatki, nekaj povezav na zunanje strani, povezane z našo tematiko, izjemno pregledna »galerija« lokacij, ki po kliku na določeno sliko preusmeri na stran s filtriranimi dogodki, forme za prijavo, registracijo in kontakt.

Po kliku na željeno skupino dogodkov (restavracije, klubi, narava, kultura, šport), se izpiše pregleden seznam vseh dogodkov, ki ustrezajo filtru (naslovi povezav: restavracije.jsp ipd.). Po kliku na skupino dogodkov AKTUALNO, se izpišejo vsi dogodki vseh skupin. Vsaka skupina dogodkov vsebuje 3 poglede (sortirano po datumu, objektu ali strnjen pogled objektov na zemljevidu).

Po kliku na naziv željenega dogodka se odpre še podstran s podrobnostmi, ki v parameter prejme id izbranega dogodka (naslov povezave: podrobnosti.jsp/?id=x). Tako se s preglednim »gridom« izpišejo podrobnosti posameznega dogodka, kot so: daljši opis, naziv objekta, datum in ura začetka, datum in ura konca, kapaciteta in vstopnina, zanimanje za dogodek, galerija slik, lokacija na zemljevidu in komentarji).

V osnovi sta možna 2 tipa uporabnikov: prvi je neprijavljen uporabnik, ki ima osnovni pregled nad dogodki in stranjo, drugi pa uporabnik administrator, ki lahko dodaja dogodke, hkrati pa ureja le tiste dogodke, ki jih je dodal sam.

Do administratorja strani uporabnika vodi gumb »registracija« na dnu prve strani. Po kliku na gumb, je uporabnik preusmerjen na stran s formo (registracija.jsp), ki od uporabnika zahteva osnovne podatke. Tako se preko imena podjetja unikatno ustvari uporabniško ime, ki se nato posreduje uporabniku na mail. S tem mu je omogočen dostop do strani »dodajanje dogodka«, »moji dogodki« in »urejanje dogodka«.

Dodajanje dogodka je implementirano z osnovno formo, ki sprejme podatke o dogodku ter fotografije le-tega.

Stran z mojimi dogodki je praktično identična ostalim seznamom dogodkov, le da so na seznam vključeni samo dogodki, ki jih je dodal uporabnik sam. Po kliku na željenega se odpre stran za urejanje, ki je praktično identična strani »dodajanje dogodka«, le da ima v vnosnih poljih že vključene podatke izbranega dogodka.

S temi lastnostmi in funkcionalnostmi je uporaba strani izjemno poenostavljena in omogoča preprosto razumevanje klikov ter urejanj posameznih komponent.



Slika 4: Lent

# Glavne komponente, metode

## Index.jsp (Glavna stran) :

* Glavna stran, po kateri je omgočeno brskanje po strani s pomočjo hash-ov, prav tako pa vsebuje tudi povezave do ostalih spletnih strani. Z uporabo JSTL-ja se stran in povezave rahlo spremenijo, glede na avtoriteto (če je uporabnik prijavljen ali pa je anonimen).
* Vsebuje »portfolio«, na katerem so razdeljeni dogodki po kategorijah, torej filter, preko katerih se dostopa do seznama dogodkov, glede na tip objekta.
* Prijava je narejena s pomočjo SpringSecurity-a. S konfiguracijo EmployeeSecurityConfiguration, so določeni jsp-ji dostopni vsem, za nekatere pa je potrebna avtoriteta. Po registraciji, ki se nahaja na svoji strani, se pošlje mail tistemu, ki se je ravnokar registriral. Potem se lahko prijavi v sistem na koncu strani. Odjava je implementirana v menijski vrstici, ali pa na indexu, na mestu, kjer se je prej nahajala prijava.

## Registracija :

* Vsebuje enostavno formo, v katero se vnesejo podatki uporabnika in podatki o objektu.
* Če uporabnik uspešno izpolni vsa potrebna polja, potem se njegovi podatki shranijo v našo bazo podatkov, iz katere se kasneje bere pri prijavi.

## Strani, dostopne preko filtrov na glavni strani :

* Na tej strani se odprejo seznami dogodkov, glede na izbran tip objekta, ali pa vsi, če je izbrana opcija »aktualno«. Te strani so drugače identične, in imajo iste funkcionalnosti. Za odpiranje seznama se uporabni metoda GetDogodekByFK(x), ta bere iz baze, in najde tip objekta po tujem ključu v tabeli.
* Odpre se seznam dogodkov, kjer pa je napisano ime dogodka, kraj, datum in slika dogodka, za lepši izgled seznama.
* Na tej strani sta tudi opciji za razvrščanje dogodkov, po datumu in po imenu objekta. To se pa implementira z identično metodo prejšnji, vendar se za to uporabi drugačen sql stavek (ORDER BY), ki razvrsti seznam po določenem parametru.
* Klik na dogodek v seznamu te poveze s stranjo »podrobnosti«.
* Opcija pogleda vseh objektov, ki gostijo dogodke na zemljevidu

## Podrobnosti :

* Na strani podrobnosti, je napisan kratek opis dogodka, lokacija, datum, vstopnina in kapaciteta.
* Implementiran je »Like« gumb, kateri pokaže zanimanje za dogodek, ki je implmentiran z jQuery-jem.
* Odpre se Galerija slik, ki so povezana na dogodek. Berejo se iz baze, odprejo pa se z JSTL stavkom »forEach«, ki prebere iz baze byte-array-e. Preko base64 encoder-ja pa se ta array spremeni v človeku vidno sliko.
* Glede na objekt, v katerem se dogodek izvaja, se na zemljevidu kreira marker, ki označuje lokacijo objekta
* Na tej strani so tudi implementirani komentarji, ki se shranijo v bazo, za vsak komentar pa tudi všeček, možnost odgovora in pa prijava zlorabe.

## Dodajanje dogodka :

* Po uspešni prijavi se odpre stran za dodajanje dogodka. Spet normalna forma, preko katere se dogodek shrani v bazo. Dogodek ima tuj ključ objekta in uporabnika, tako da, je vsak dogodek vezan na določen objekt oziroma uporabnika.
* Implementirano je tudi dodajanje slik na dogodek.

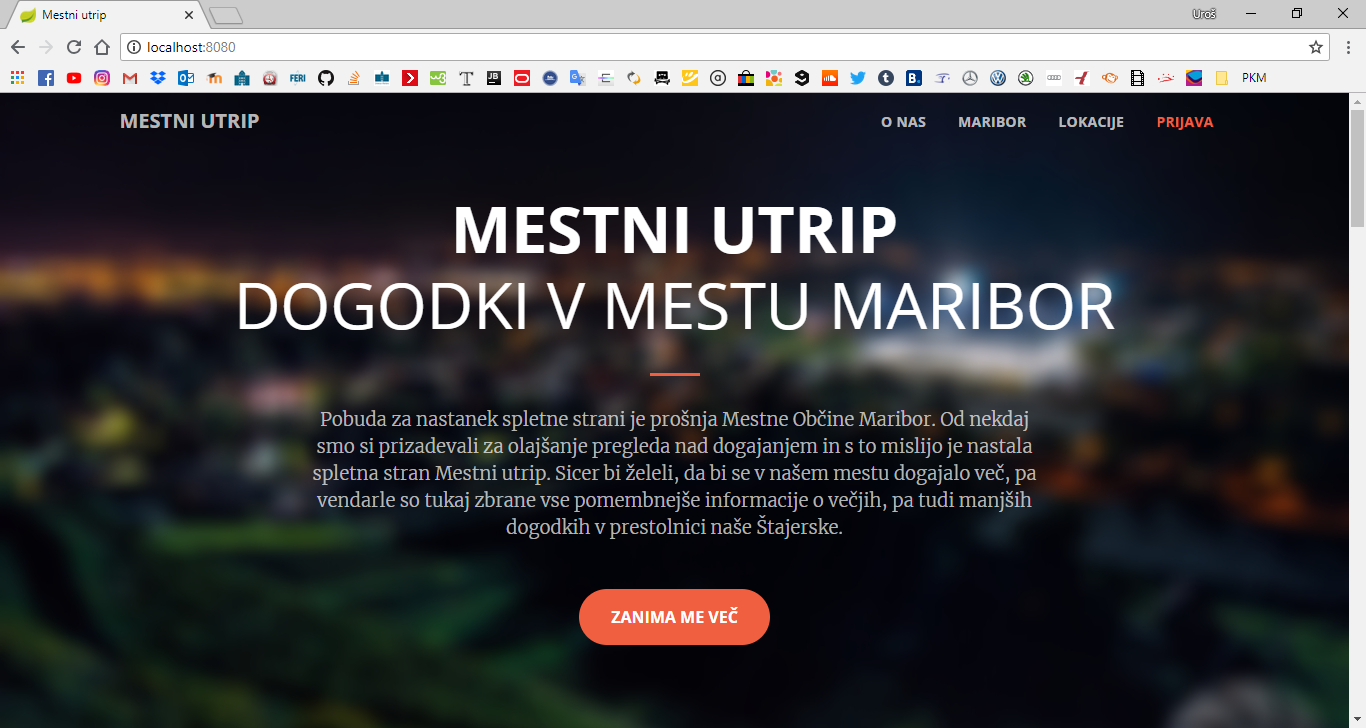
## Moji dogodki :

* Odpre se seznam dogodkov, ki jih je prijavljen uporabnik objavil. Uporabljena je zelo podobna metoda, kot na prejšnjih seznamih, vendar, se tu išče po tujem ključu uporabnika.

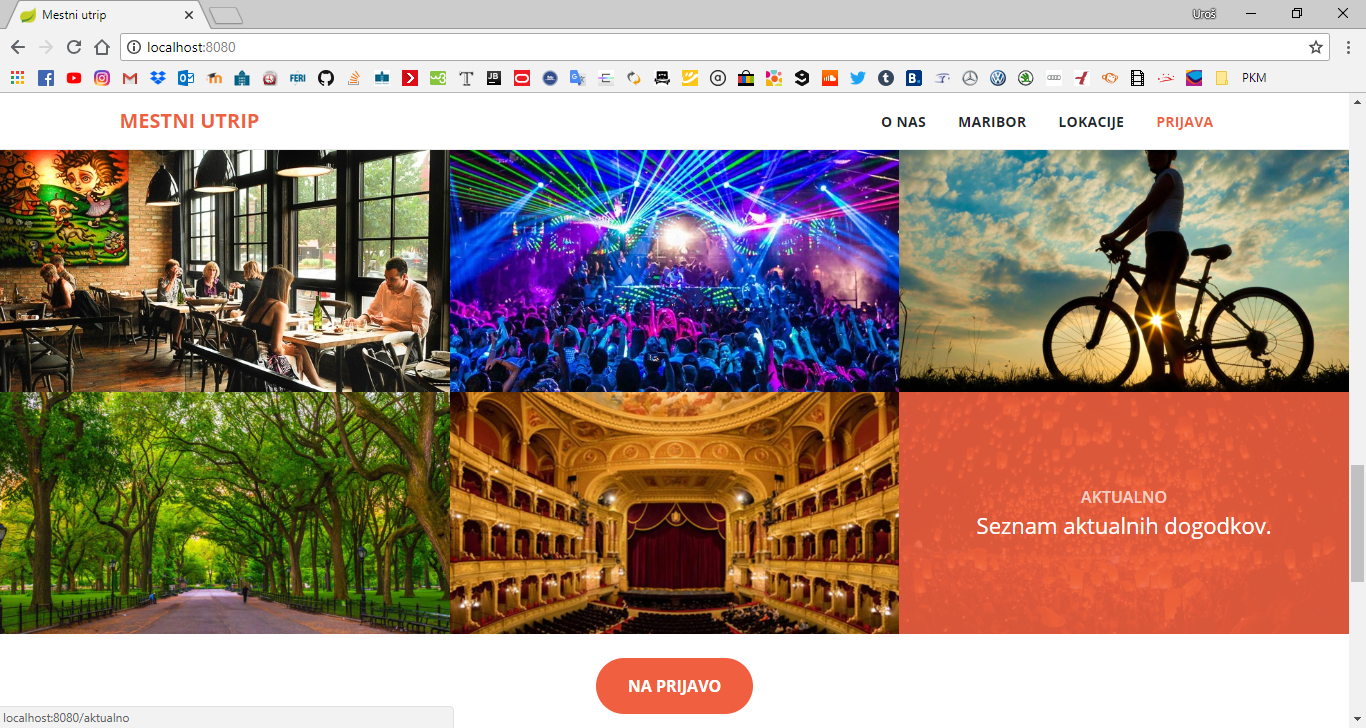
## Urejanje dogodkov :

* Ko se izbere dogodek, na strani mojiDogodki, se odpre forma, ki je identična formi za dodajanje, vendar se na tej le ureja, že obstoječi dogodek.
* Dogodek se lahko na tej strani tudi izbriše.

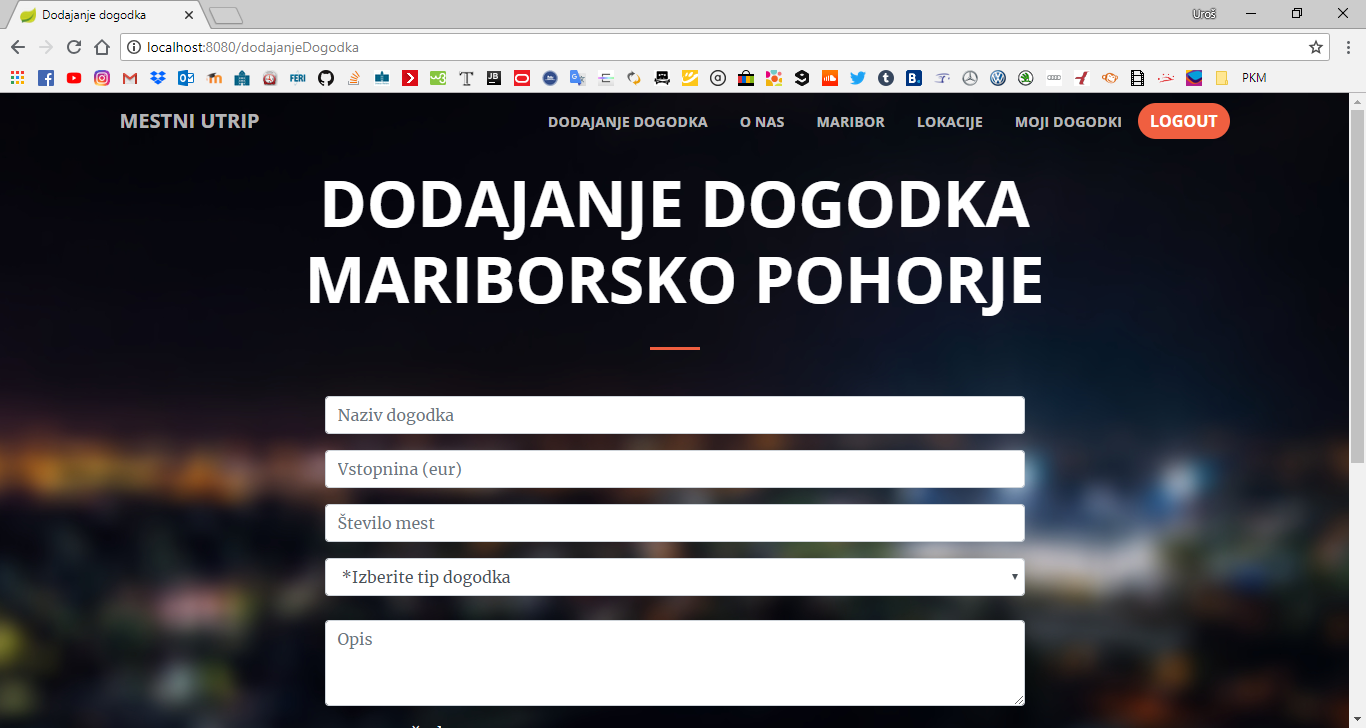
# Uporabniški vmesnik aplikacije



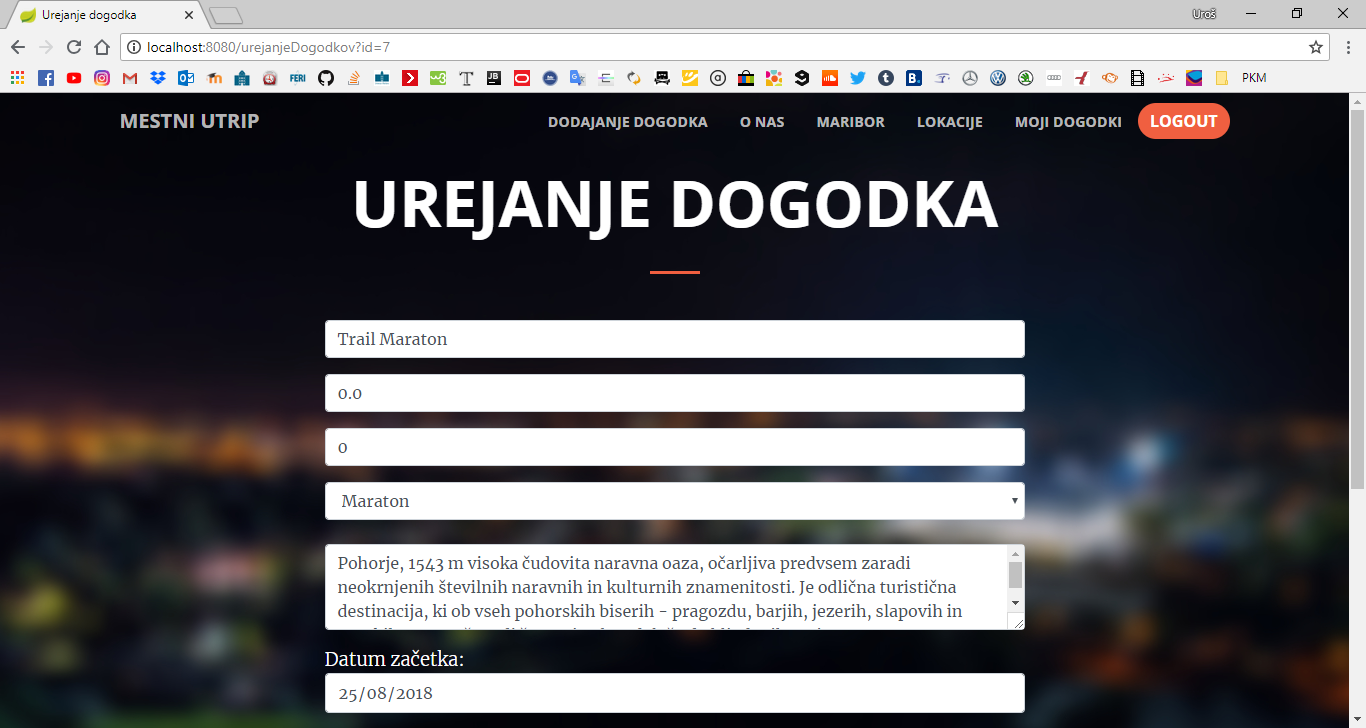
Slika 5: Pozdravna stran Index.jsp



Slika 6: Galerija filtriranih seznamov



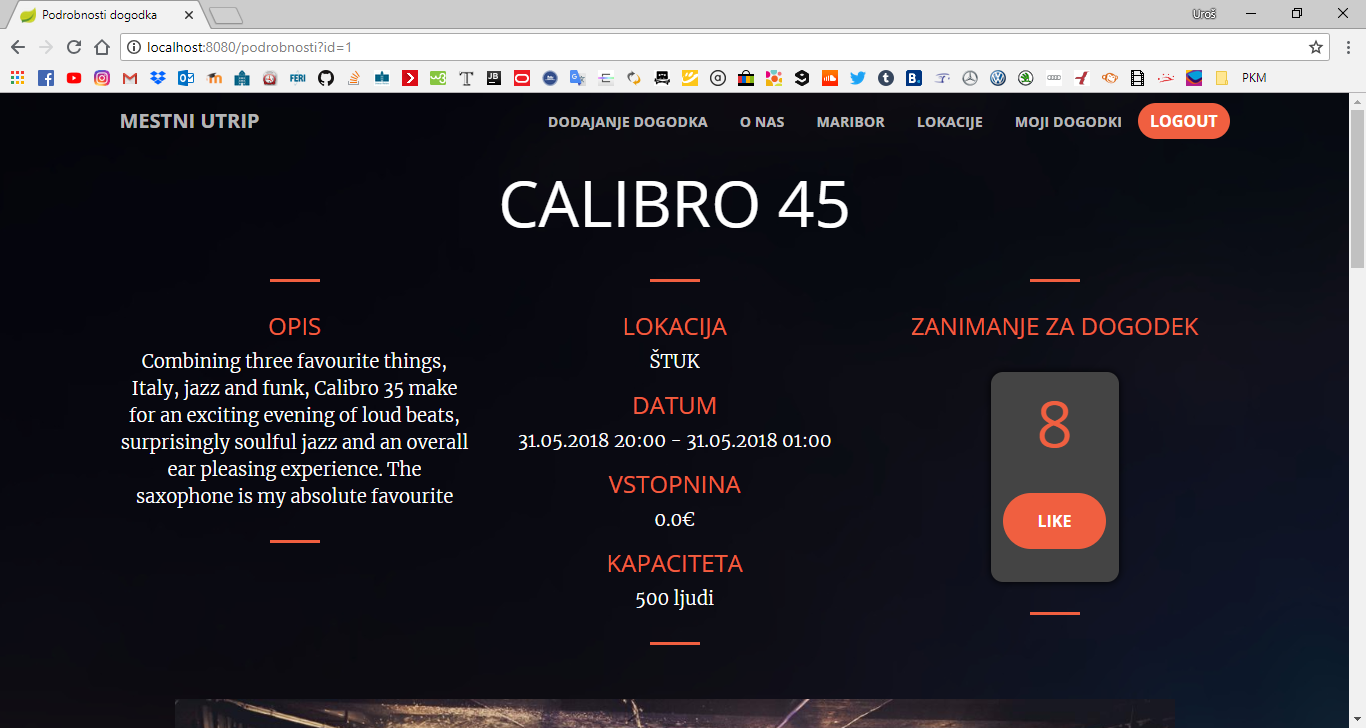
Slika 7: Dodajanje dogodka



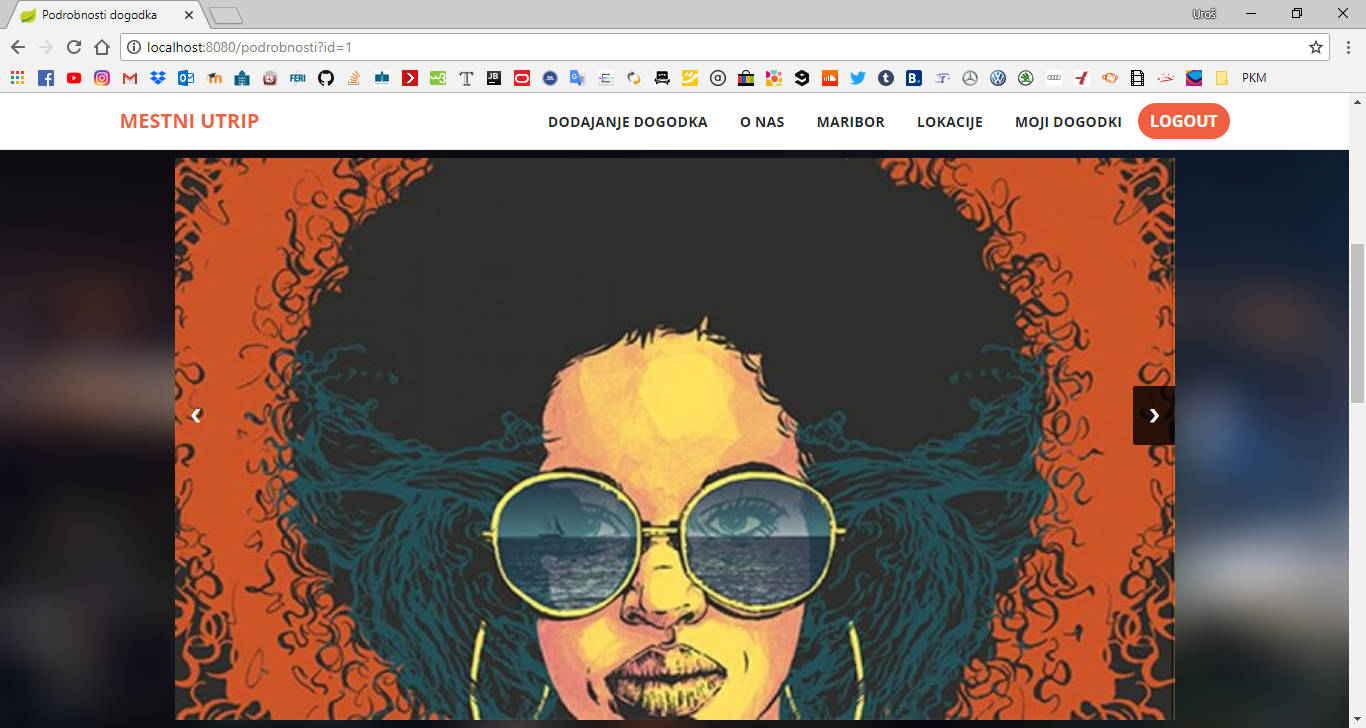
Slika 8: Urejanje željenega dogodka



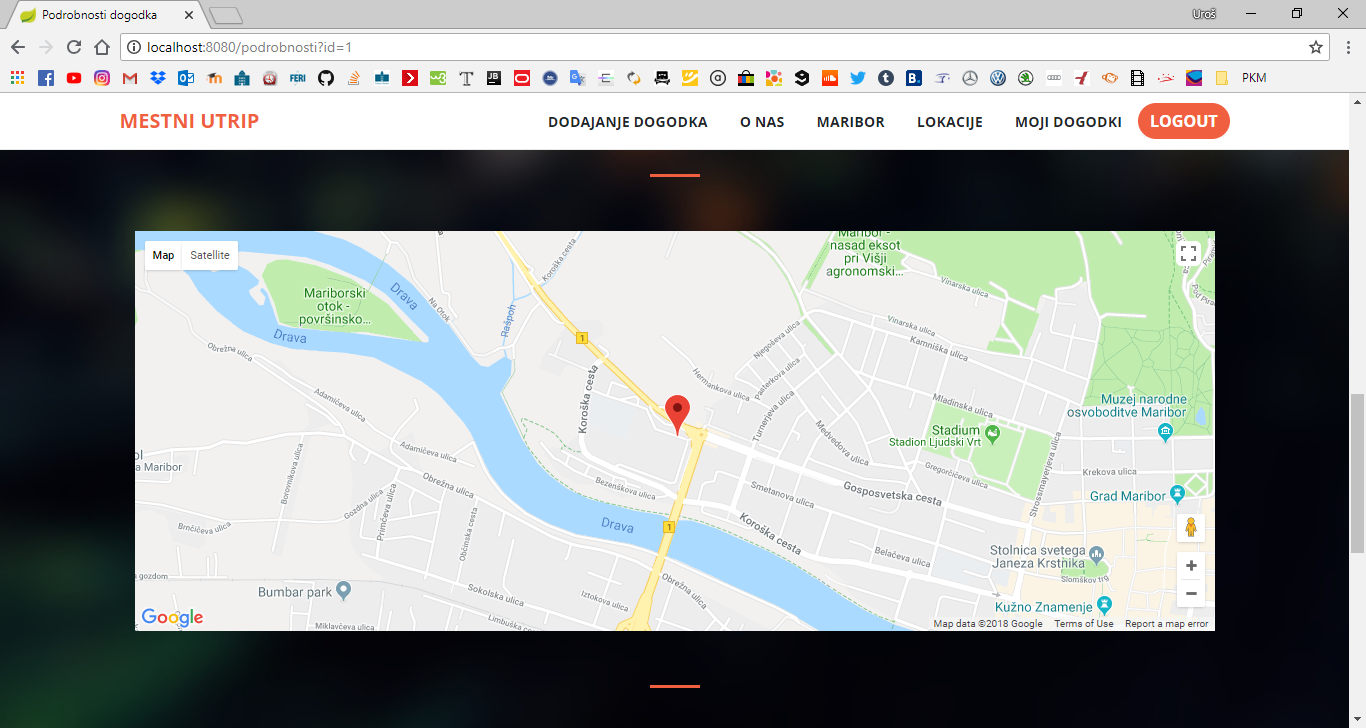
Slika 9: Aktualni dogodki



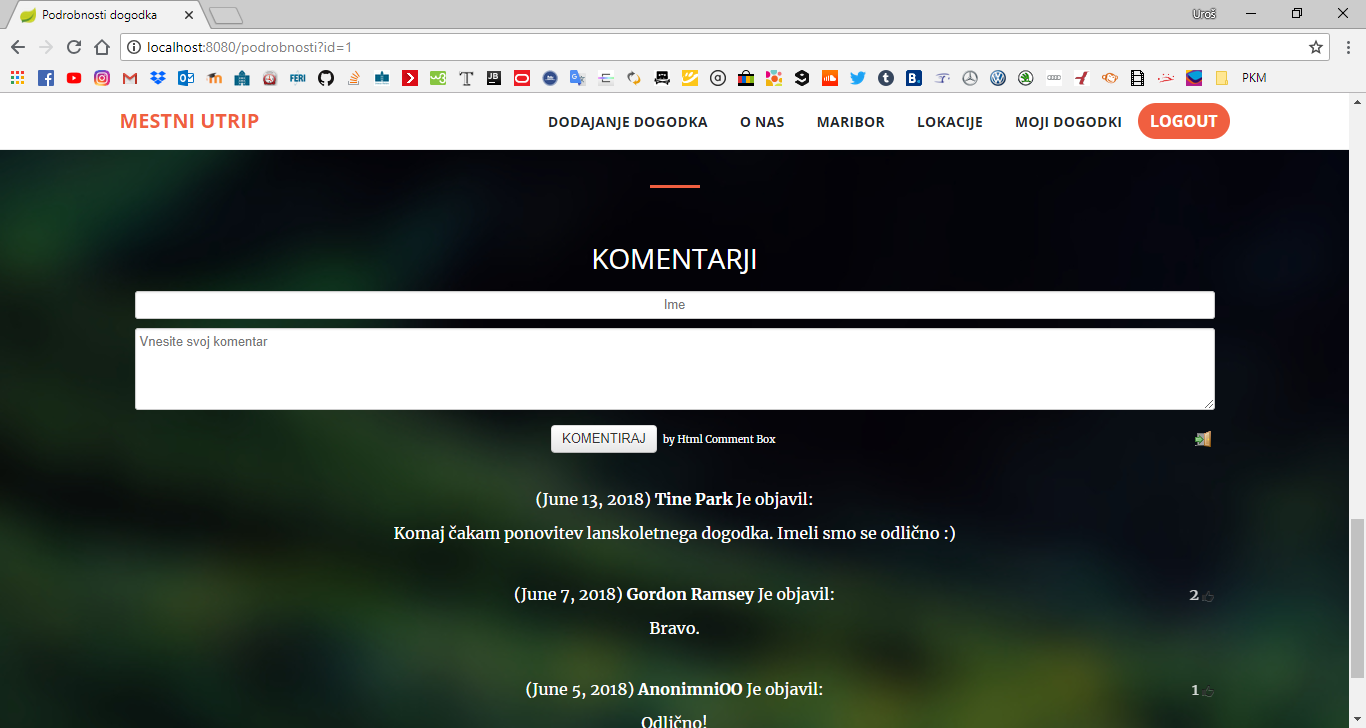
Slika 10: Podrobnosti izbranega dogodka



Slika 11: Galerija slik pri podrobnostih

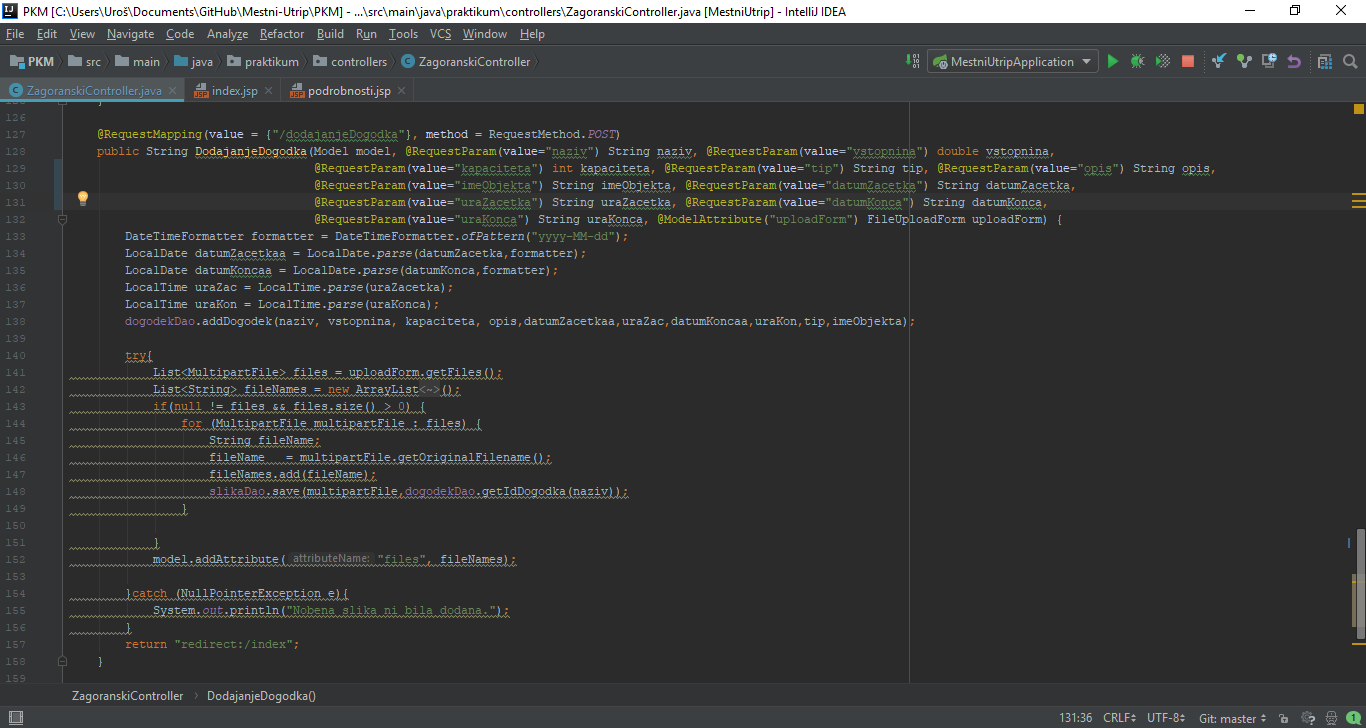


Slika 12: Zemljevid pri podobnostih

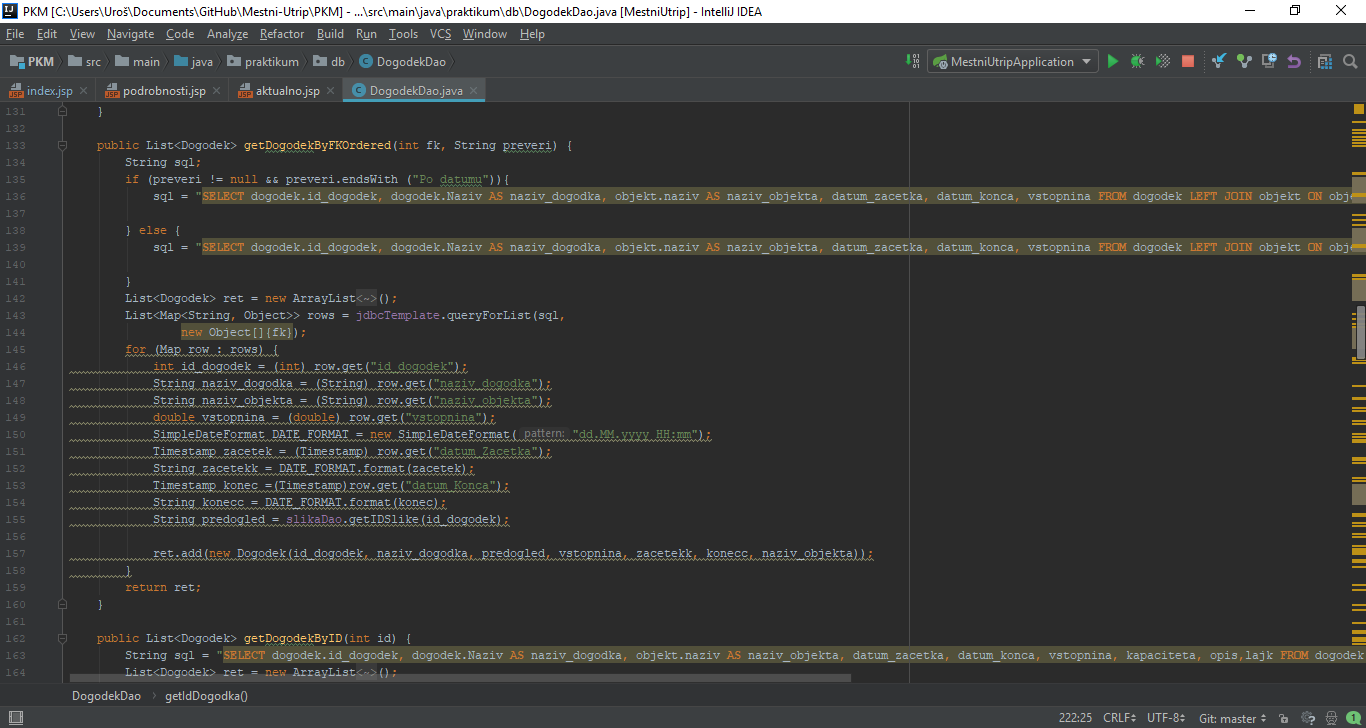


Slika 13: Komentarji

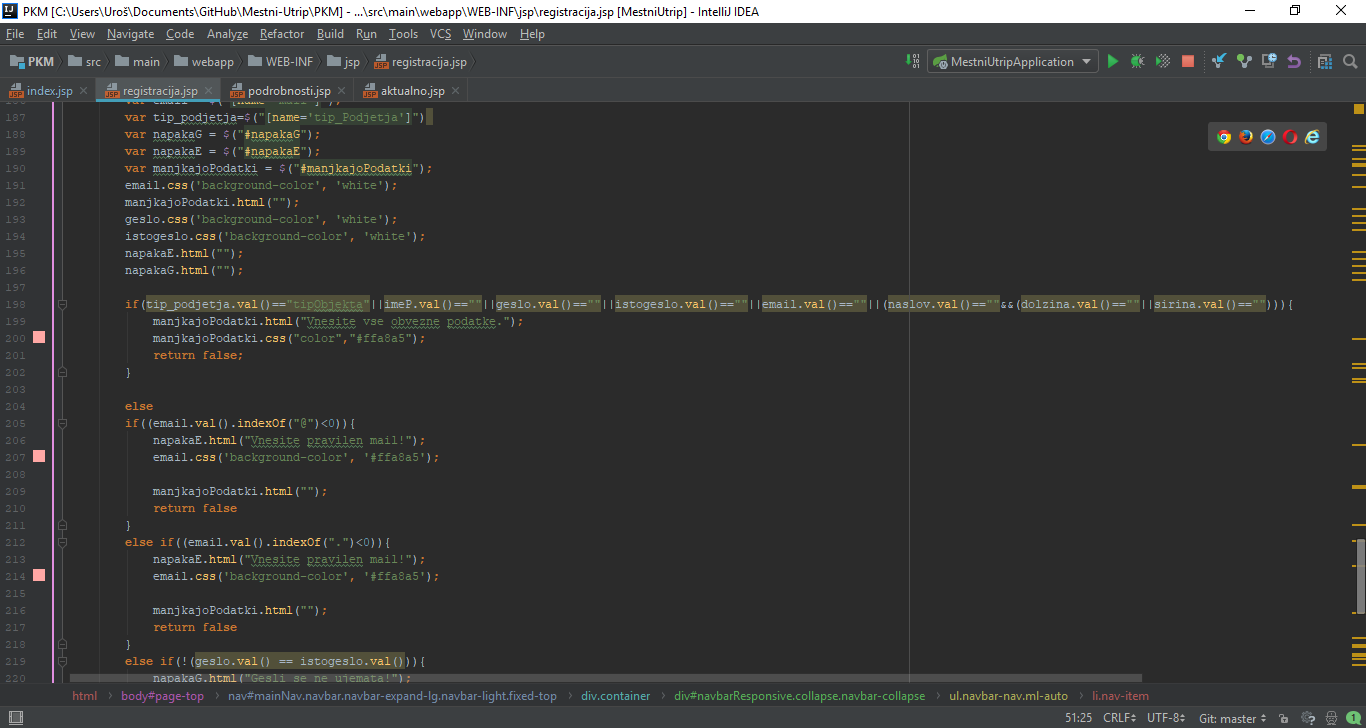
# Pomembnejši izseki kode



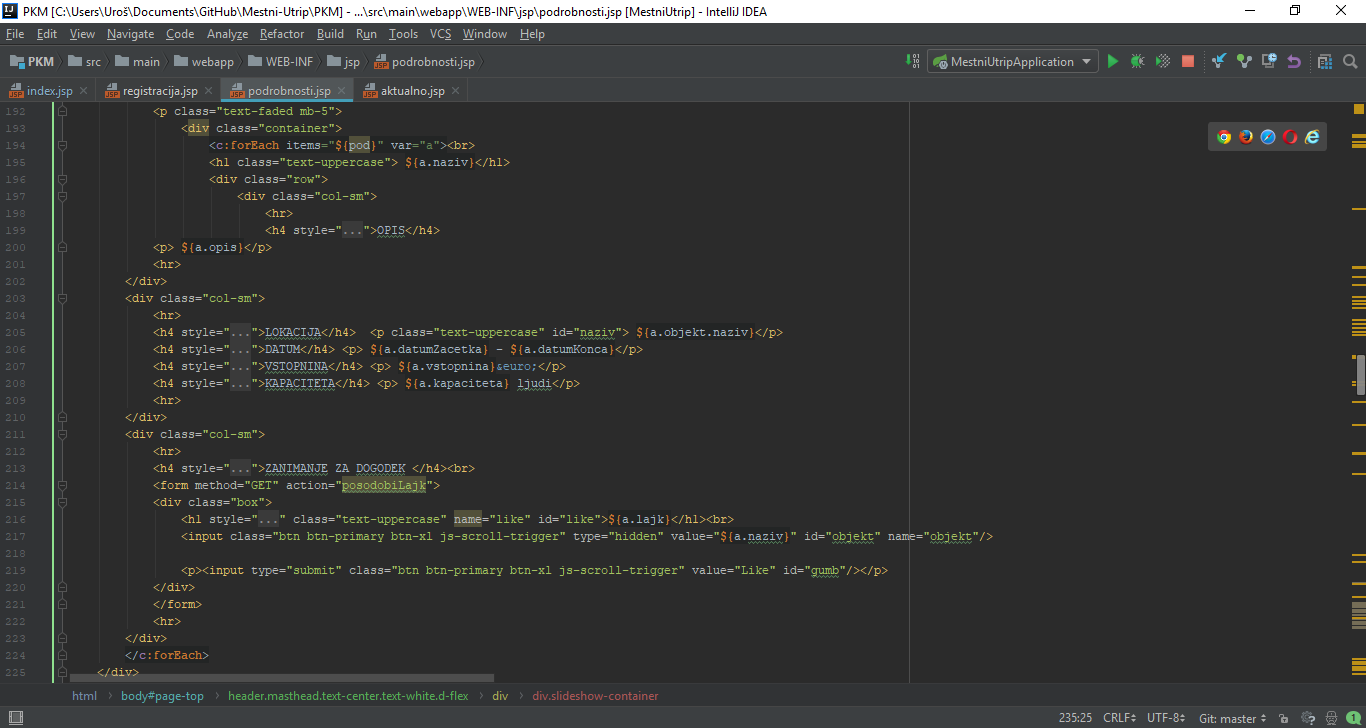
Slika 14: Controller Dodajanje dogodka



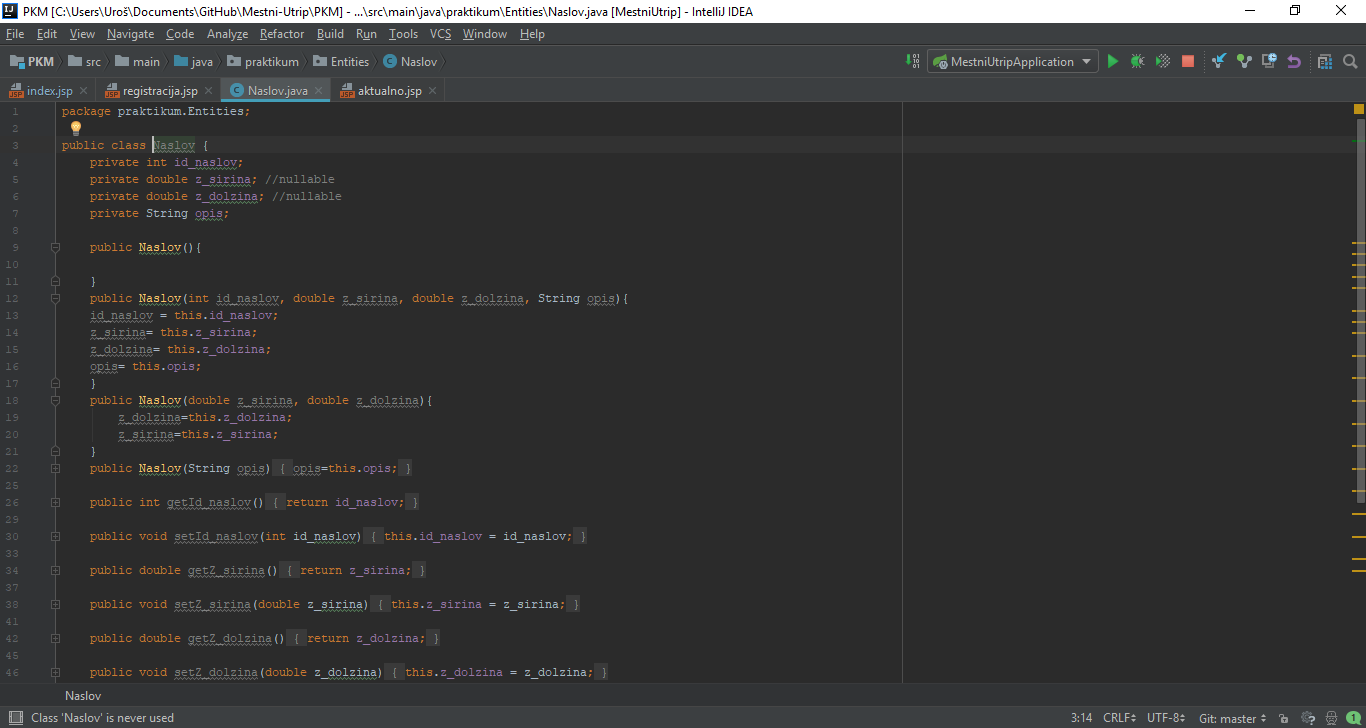
Slika 15: Sortirano branje dogodka iz baze



Slika 16: jQuery pri registraciji



Slika 17: JSTL



Slika 18: Javanski razred

# Poročilo testiranja

Testiranje smo izvedli tekom izdelave aplikacije, zato posebno testiranje na koncu ni bilo potrebno. Preizkusili smo vse mogoče funkcionalnosti in klike na naši strani, zato smo sto odstotno prepričani, da spletna stran deluje tako kot mora.



Slika 19: Testiranje