Praktikum 1:

**Dogodki**

Avtorji: Evgen Tušek, Toni Žunec, Marko Pavičevič

Mentor: Luka Pavlič

Kazalo:

Contents

[1. Uvod 4](#_Toc524535616)

[1.1 Besedilo naloge 4](#_Toc524535617)

[Specifikacije in zahteve 4](#_Toc524535618)

[1.2 Izbrano ogrodje in jeziki 5](#_Toc524535619)

[2. Uporabljena orodja 5](#_Toc524535620)

[2.1 Eclipse Java Oxygen 5](#_Toc524535621)

[slika 1: eclipse dokumentacija 6](#_Toc524535622)

[2.2 WAMP 6](#_Toc524535623)

[slika 2:WAMP 6](#_Toc524535624)

[3. Podatkovna baza 6](#_Toc524535625)

[3.1 Načrtovanje baze 7](#_Toc524535626)

[slika 3: prva baza 7](#_Toc524535627)

[slika 4: Končna baza 8](#_Toc524535628)

[3.2 Ustvarjanje podatkovne baze 8](#_Toc524535629)

[slika 5: Podatkovna baza v phpMyAdmin 8](#_Toc524535630)

[Slika 5 – Vnašanje v podatkovno bazo z ukazom za pridobitev vseh podatkov iz entitete Uporabniki. 9](#_Toc524535631)

[4. Uporabniški vmesnik 9](#_Toc524535632)

[5. Komponenti spletne strani 9](#_Toc524535633)

[5.1 Kontrolerji 10](#_Toc524535634)

[Slika 8 – Kontroler za dodajanje dogodka. 11](#_Toc524535635)

[Slika 9 - Kontroler pošiljanja e-pošte. 12](#_Toc524535636)

[Slika 10-Kontroler prijave na dogodek. 12](#_Toc524535637)

[5.2 Model 12](#_Toc524535638)

[Slika 11-SQL povpraševanje za dodajanjedogodka v bazo. 12](#_Toc524535639)

[Slika 12-SQL povpraševanje za pridobitev tipa uporabnika. 13](#_Toc524535640)

[5.3 View 13](#_Toc524535641)

[5.4 E-pošta 13](#_Toc524535642)

[Slika 13-Metoda za pošiljanje sporočila. 14](#_Toc524535643)

[Slika 14 – Prikaz izgleda poslanega besedila. 14](#_Toc524535644)

[Ocenjevanje 14](#_Toc524535645)

[Nadzor dostopa 14](#_Toc524535646)

[Slika 15 – Preverjanje tipa uporabnika. 15](#_Toc524535647)

[Testiranje 15](#_Toc524535648)

[Zaključek 15](#_Toc524535649)

# Uvod

## Besedilo naloge

Soočate se z organizacijo skupinskih dogodkov in sestankov. Razvijte orodje, ki bo pomagalo pri razpisovanju, vodenju prisotnosti in ocenjevanju dogodkov. Organizacija dogodka se običajno začne z vpisom dogodka, ki ga ponudimo določenim uporabnikom ali skupini le-teh. Dogodek označujejo termin, trajanje, prostor in ostali podatki. Omogočite tudi možnost določitve skrajnega termina prijave in odjave od dogodka ter določitev minimalnega in maksimalnega števila udeležencev. Vabljenim na dogodek omogočite enostavno prijavljanje na dogodke, organizator na naj vidi, koliko in kdo vse je prijavljen na določen dogodek. Po dogodku lahko organizator tudi označi, kdo se je dogodka udeležil, udeleženci pa lahko dogodek tudi ocenijo. Celotno organizacijo podprite tudi s sporočanjem preko e-pošte. Predvideni uporabniki Organizatorji dogodkov, udeleženci dogodkov.

## Specifikacije in zahteve

• Razpis dogodka

• Prijava na dogodek

• Odjava od dogodka

• Obveščanje o dogodku

• Označevanje prisotnosti

• Ocenjevanje dogodka

• … Omejitve

• Organizator bo uporabljal primarno osebni računalnik, udeleženci pa mobilne telefone.

## Izbrano ogrodje in jeziki

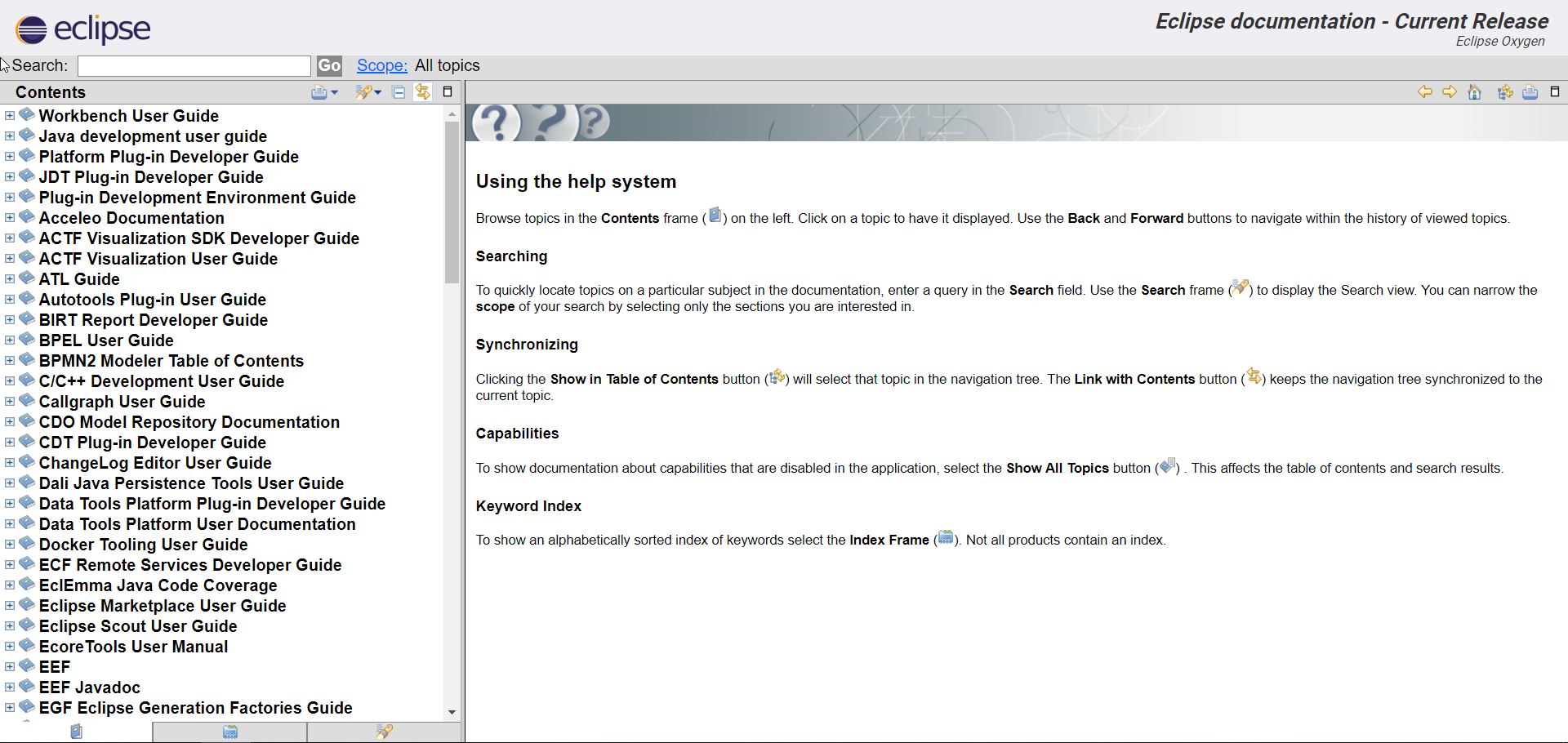
Za ogrodje (''framework'') smo uporabljali CodeIgniter. Zdelo se nam je primerno saj je ugodno orodje, ki pušča malo odtisov in ima enostaven in močan toolkit za ustvarjanje web aplikacij. Za razvojno okolje za pisanje kode smo uporabljali Eclipse, za podatkovno bazo in dostop do strežnika pa smo uporabljali WAMP, saj smo delali v programskem jeziku PHP. Sprva smo delali za JSP in H2 bazo ampak to nam po več poskusih ni uspelo, tako da smo našli alternativo v PHPju.

Za ustvarjanje spletnih strani smo uporabili HTML, za stilsko oblikovanje pa CSS in Bootstrap.

# Uporabljena orodja

## Eclipse Java Oxygen

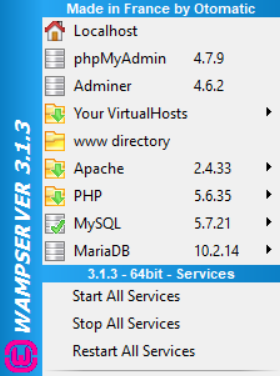
Eclipse Java Oxygen je programsko orodje in okolica za ustvarjanje Java aplikacij in spletnih aplikacij. Je orodje katero smo uporabljali pri izdelavi projekta, saj smo bli z njim najbolj seznanenji. Poleg tega smo rabli plugin PHP Development Tools da smo lahko obravnavali PHP kodo. Ima tudi Git iontegracijo, ki omogoča vse funkcije le tega.



### slika 1: eclipse dokumentacija

## WAMP

WAMP je orodje, ki omogoča pisanje dinamičnih spletnih aplikacij na Windowsu z Apache2,PHP,MySQL in MariaDB. Izbrali smo ga zato, ker nam avtomatsko inštalira vse kaj rabimo za intuitivno razvijanje spletnih aplikacij brez da bi spreminjali nastavitvene datoteke.



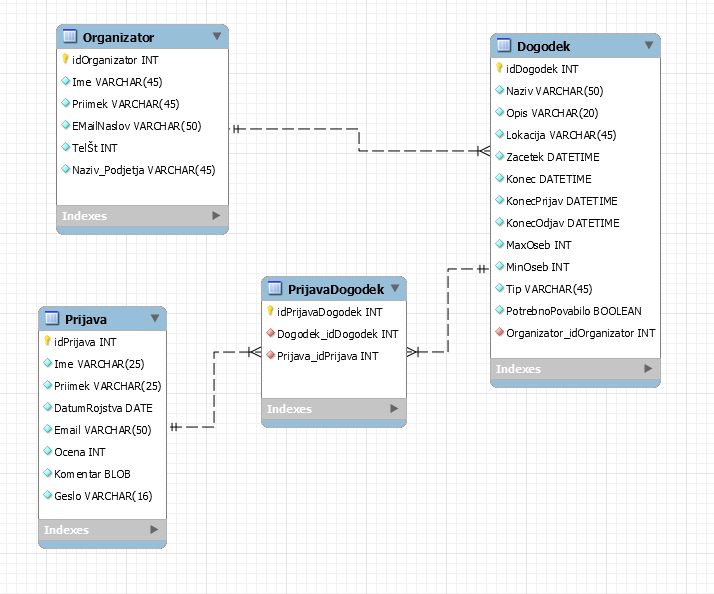
### slika 2:WAMP

# Podatkovna baza

Za podatkovno bazo smo sprva naredili E-R diagram. Po tem smo ustvarili skripto za ustvarjanje tabel. Sprva smo probali sami narediti z pomočjo MySQL in nam ni šlo, ker smo imeli problem z vmesno bazo. Potem smo probali z H2 bazo, ki je sprva delala, ampak smo naletili na težavo pri JSP kodi in prikazu podatkov. Na koncu smo z pomočjo WAMP-a naredili bazo v MySQL.

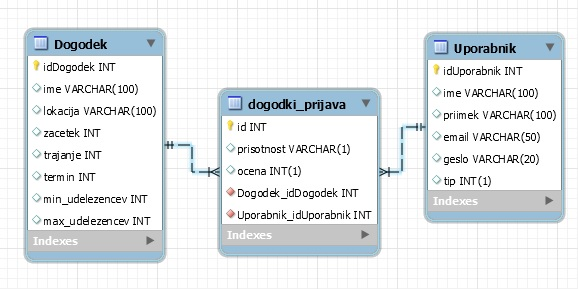
## Načrtovanje baze

Spva smo se osredotočili na zahteve naloge in ustvarili entitete . Te smo smiselno opredelili glede na navodila.



### slika 3: prva baza

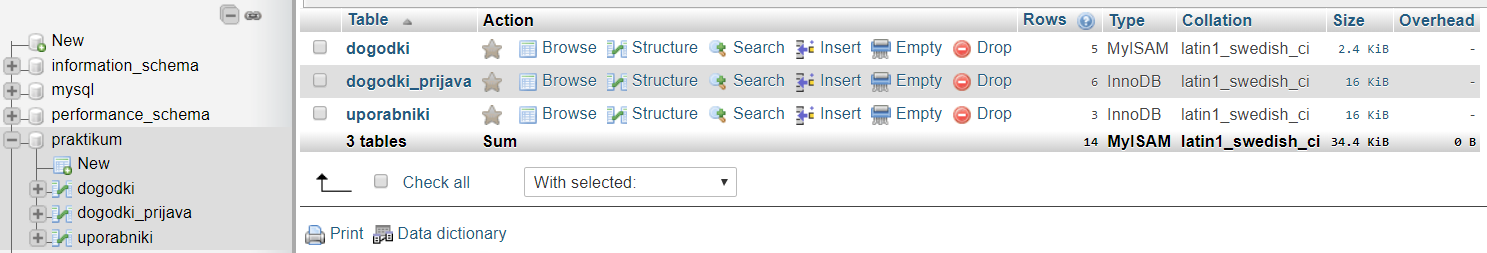
Nato smo izločili nepotrebne podatke in združili Pijava in Organizator, ter dodali tip, da se vidi, če je organizator ali uporabnik. Vmes smo še odstranili telefonsko številko.



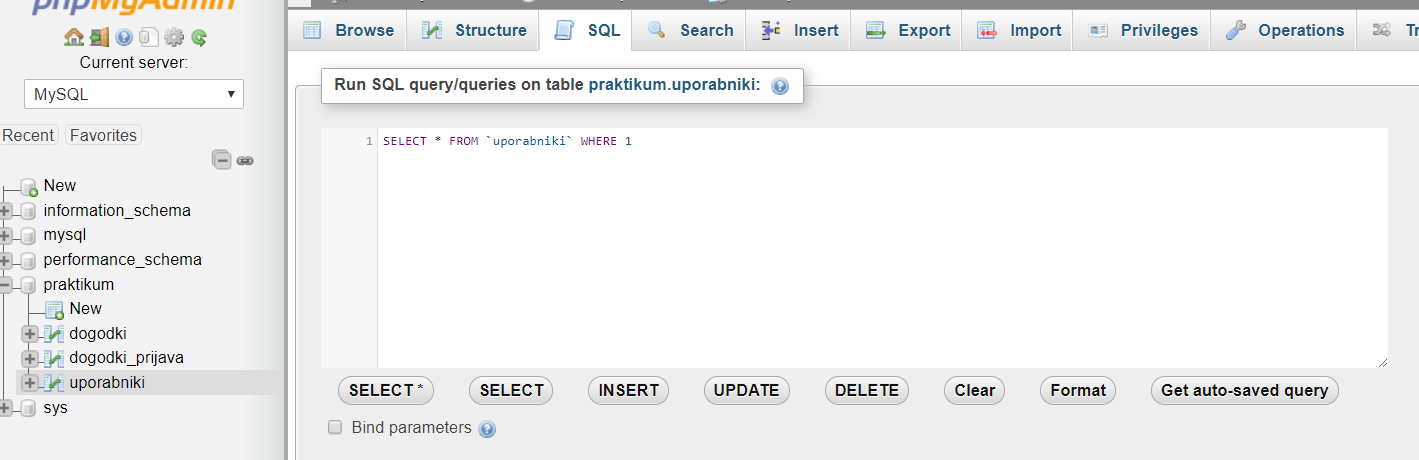
### slika 4: Končna baza

## Ustvarjanje podatkovne baze

Zaradi zapletov pri uporabi MySQL smo se odločili da delamo z H2 bazo. To nam je sprva bilo uspešno ampak zaradi zapletu pri JSP nam ni pokazalo podatke iz baze. Alternativo smo našli v PHPju, bolj podrobno v WAMP (phpMyAdmin), na katerem smo napisali tabele v jeziku SQL.



### slika 5: Podatkovna baza v phpMyAdmin



### Slika 5 – Vnašanje v podatkovno bazo z ukazom za pridobitev vseh podatkov iz entitete Uporabniki.

# Uporabniški vmesnik

Pri izdelavi uporabniškega vmesnika smo uporabljali CSS in Bootstrap. CSS nam nudi ogromno količino stilov in oblikovanja, Bootstrap smo uporabljali zgolj za lep izgled na vseh napravah. Odločili smo se za ta orodja zato, ker so nam najbližja, saj smo jih redno uporabljali pri predmetu Osnove Spletnih Tehnologij.

# Komponenti spletne strani

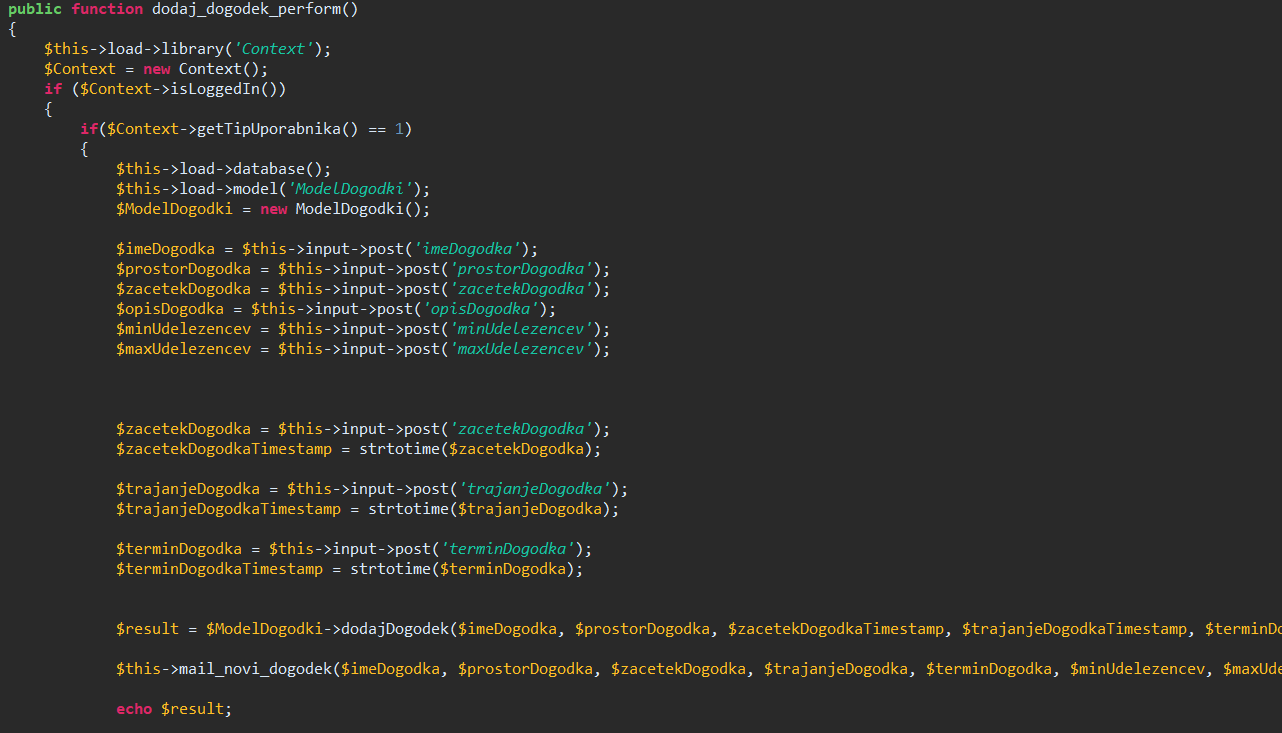
To poglavje je namenjeno predstavitvi komponent spletne strani naše aplikacije. Obrazložene so z izsekami kode.

Komponenti so:

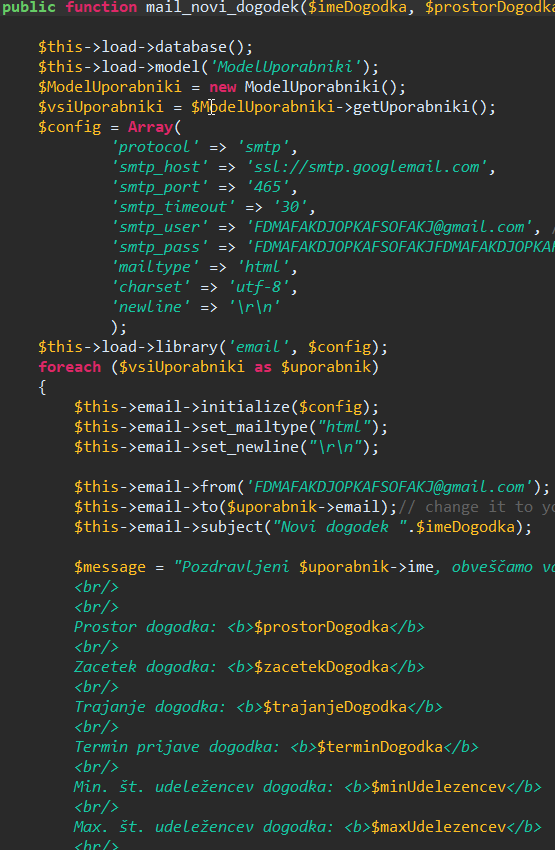
* Kontrolerji
* View
* Prijava in registracija
* E – pošta
* Modeli
* Objavlanje in ocenjevanje

## Kontrolerji

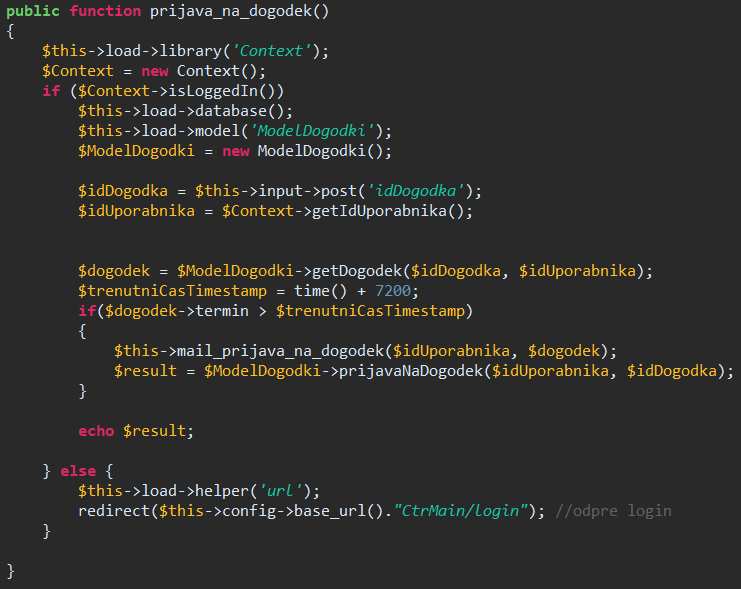
Kontrolerji upravljajo (''kontrolirajo''), kaj mi vidimo na spletni strani. V njih vključujemo skupek metod in fukcij, ki se odzivajo na določene zahteve. So pomembni za delovanje spletne strani.



### Slika 8 – Kontroler za dodajanje dogodka.



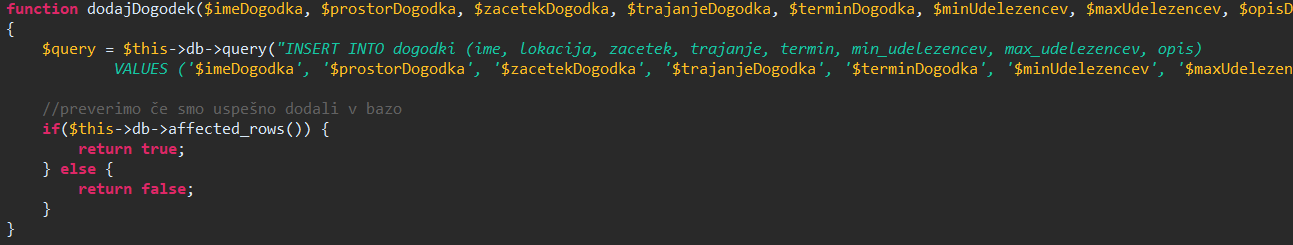
### Slika 9 - Kontroler pošiljanja e-pošte.



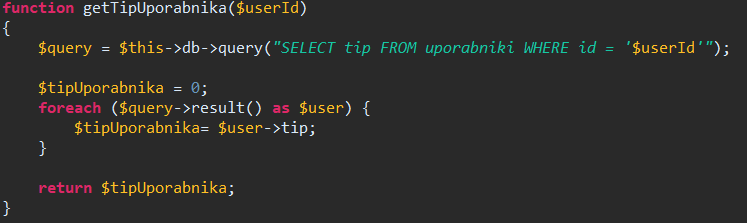
### Slika 10-Kontroler prijave na dogodek.

## Model

Modeli zajemajo metode, preko katerih se spreminja podatkovna baza.



### Slika 11-SQL povpraševanje za dodajanjedogodka v bazo.



### Slika 12-SQL povpraševanje za pridobitev tipa uporabnika.

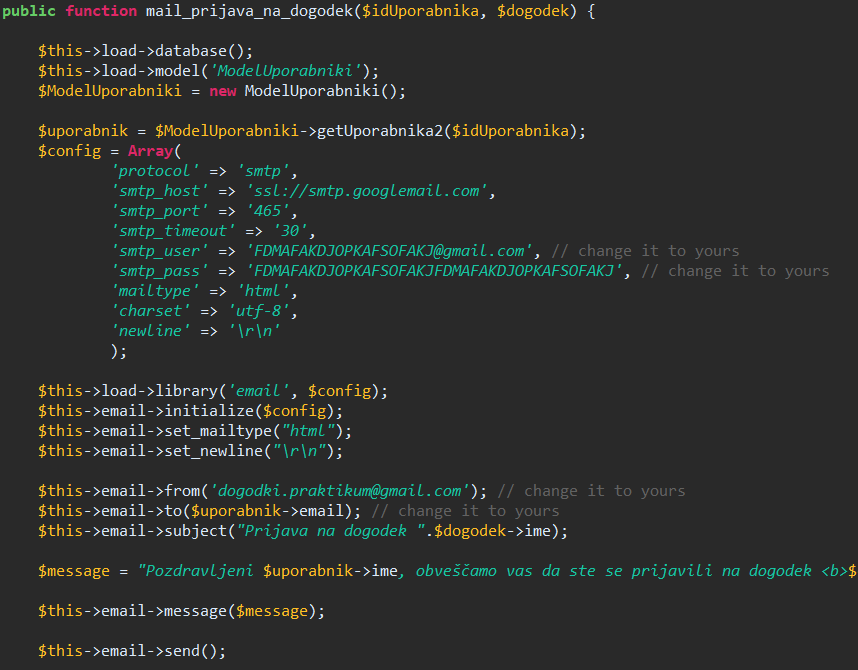
## View

V View komponento našega projekta spada vse, kar uporabnik vidi. V to spada naše HTML in PHP strani.

## E-pošta

V projekt smo vključili e-poštno sporočanje, kjer se uporabnikom pošlje sporočilo ob prijavi in odjavi na dogodek.

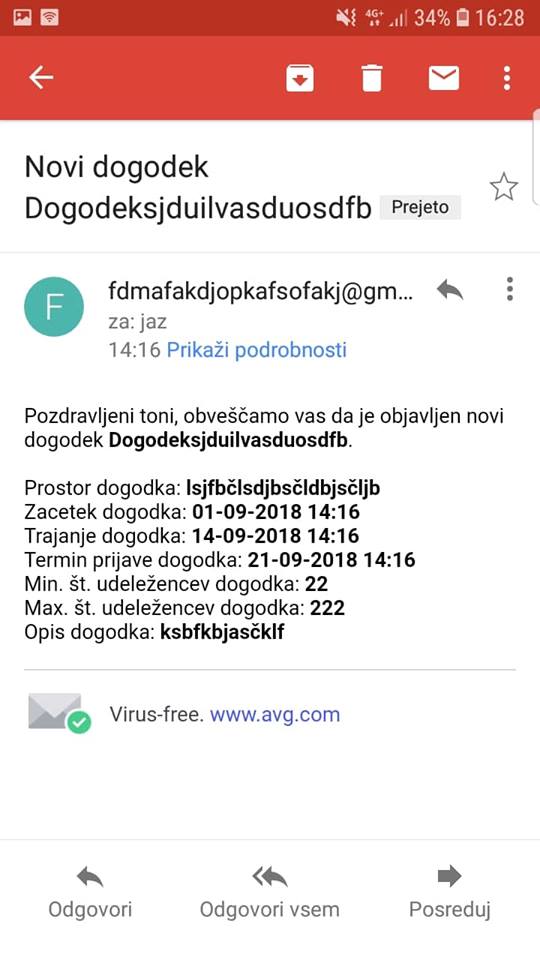
Omogočeno je tudi obveščanje organizatorjev o novo objavljenih dogodkih.



### Slika 13-Metoda za pošiljanje sporočila.

Preko metode na sliki se pošlje e-mail prijavljenemu uporabniku z vnešene e-pošte.

Uporabnik dobi sledeči mail.



### Slika 14 – Prikaz izgleda poslanega besedila.

## Ocenjevanje

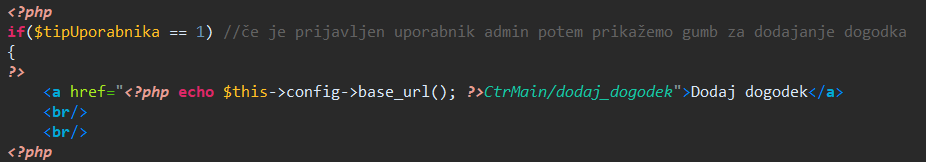
Med zahtevanimi funkcionalnostmi je bilo tudi ocenjevanje dogodkov.

To se omogoči, kadar organizator potrdi prisotonost prijavljenim uporabnikom na dogodku, kateremu je ze potekel dan odjav.

## Nadzor dostopa

Uporabnikom je omogočena registracija samo kot tipa uporabnik.

Če se želi oseba registrirati kot organizator, se mora to izvesti prek direktnega dodajanja v podatkovno bazo. S tem se rešimo nevarnosti, da bi se objavlali neresnični dogodki.



### Slika 15 – Preverjanje tipa uporabnika.

## Testiranje

Testiranje je potekalo preko profila organizatorja in uporabnika.

Predvsem je bilo na začetku zagotoviti, da uporabnik nima dostopa do funkcij organizatorja.

Pomembno je bilo tudi, da so zahtevana vsa polja pri vnosti podatkov uporabnika in dogodka.

Nekaj težav nam je povzročalo pri javljanu napake med odjavo in prijavo. Problem smo našli v povezavi s funkcijo pošiljanja e-maila, katero smo reševali s pomočjo poročila o napaki v brskalniku.

# Zaključek

Zaradi nekaj neuspehov z drugimi tehnologijami, nam je projekt predstavljal ogromen izziv.

Preizkusili smo več, še ne poznanih tehnologij in nam je na koncu uspelo.

Zaradi neuspehov z JSP-ji in Servleti, smo se odličili poiskusiti s PHP jezikom v povezavi s CodeIgniter frameworkom, kateri nam je bil v ogromno korist pri začetku. Povezava s WAMP serverjem si bila težka, imelo smo le nekatere težave pri instaliranju tega.

Sam WAMP ponuja več podatkovnih baz s katerimi bi lahko delali.

Zaradi ogromno zapletov pri projektu nam je šlo veliko časa.

Zajeli smo vse funckionalnosti, katere smo si zadali, le za lepše oblikovanje spletne strani nam je zmanjkalo časa.