

Podporni informacijski sistem Strokovno poročilo

Mentor: Darjan Toth, prof. Avtor: Nejc Vrčon Zupan, R 4. B

Ljubljana, marec 2020

**Povzetek**

V poročilu bom predstavil podatkovno bazo in spletno aplikacijo, katera zagotavlja informacijsko podporo sejmu rabljenih izdelkov. Poročilo zajema podroben opis podatkovne baze, spletne aplikacije in njene uporabnosti, kot rešitev težav v praksi. Prav tako bom omenil težave, na katere sem naletel pri razvijanju podatkovne baze in aplikacije, ter možnih izboljšavah končnega izdelka.

**Ključne besede:** Firebird, PHP, SQL, HTML, PSQL

**Abstract**

The purpose of this paper is to report on the development of my database and web application which together create my support information system. The report describes the database and web application as an example of a useful solution that can be used in the real world. I also talk about the problems, which I encountered while developing my information system and possible improvements one could make to the end product.

**Key words**: Firebird, PHP, SQL, HTML, PSQL

Kazalo

[**1. UVOD** 5](#_Toc37887698)

[**2. CILJI** 5](#_Toc37887699)

[**2.1 Uporabljene tehnologije** 5](#_Toc37887700)

[**2.1.1 PHP** 5](#_Toc37887701)

[**2.1.2 HTML** 6](#_Toc37887702)

[**2.1.3 CSS** 6](#_Toc37887703)

[**2.1.4 Firebird** 7](#_Toc37887704)

[**2.1.5 XAMPP** 7](#_Toc37887705)

[**3. SHEMA POTEKA SEJMA** 8](#_Toc37887706)

[**4. SHEMA PODATKOVNE BAZE IN TABEL** 10](#_Toc37887707)

[**4.1 Tabela Member** 10](#_Toc37887708)

[**4.2 Tabela Staff** 10](#_Toc37887709)

[**4.3 Tabela Ptype** 10](#_Toc37887710)

[**4.4 Tabela Product** 10](#_Toc37887711)

[**4.5 Tabela Event** 11](#_Toc37887712)

[**4.6 Shema podatkovne baze** 11](#_Toc37887713)

[**4.7 Shranjene procedure** 12](#_Toc37887714)

[**4.7.1 Procedura CASH\_LOG** 12](#_Toc37887715)

[**4.7.2 Procedura CASH\_SUM** 12](#_Toc37887716)

[**4.7.3 Procedura FIND\_PRODUCT** 12](#_Toc37887717)

[**4.7.4 Procedura GEN\_MEM** 12](#_Toc37887718)

[**4.7.5 Procedura INPUT\_EVENT** 12](#_Toc37887719)

[**4.7.6 Procedura INPUT\_PRODUCT** 13](#_Toc37887720)

[**4.7.7 Procedura INPUT\_PTYPE** 13](#_Toc37887721)

[**4.7.8 Procedura INPUT\_STAFF** 14](#_Toc37887722)

[**4.7.9 Procedura PP\_FLOW** 14](#_Toc37887723)

[**4.7.10 Procedura PTYPE\_OUT** 14](#_Toc37887724)

[**4.7.11 Procedura OUT\_FLOW** 14](#_Toc37887725)

[**4.7.12 Procedura STAFF\_LOGIN** 16](#_Toc37887726)

[**4.7.13 Procedura FIND\_STAFF** 16](#_Toc37887727)

[**4.7.14 Procedura SHOW\_BASKET** 16](#_Toc37887728)

[**4.8 Sprožilci** 18](#_Toc37887729)

[**4.8.1 Sprožilec IP\_LOG** 18](#_Toc37887730)

[**4.8.2 Sprožilec E\_CHK** 18](#_Toc37887731)

[**4.9 Pomožna skripta** 20](#_Toc37887732)

[**5. SPLETNA APLIKACIJA** 20](#_Toc37887733)

[**5.1 Zakaj spletna aplikacija** 20](#_Toc37887734)

[**5.2 Povezovanje z bazo v PHP** 21](#_Toc37887735)

[**5.3 Videz spletne aplikacije** 21](#_Toc37887736)

[**5.4 Generiranje dokumentov** 22](#_Toc37887737)

[**6. UPORABA INFORMACIJSKEGA SISTEMA** 23](#_Toc37887738)

[**7. MOŽNE IZBOLJŠAVE** 24](#_Toc37887739)

[**8. ZAKLJUČEK** 24](#_Toc37887740)

[**9. VIRI** 25](#_Toc37887741)

[**9.1 Citati** 25](#_Toc37887742)

# **1. UVOD**

V poročilu je opisan izdelek, ki sem ga izdelal za poklicno maturo. Izdelek je namenjen informacijski podpori sejma rabljenih izdelkov. Realiziran je s podatkovno bazo Firebird in spletno aplikacijo, napisano v jeziku PHP. Izdelek ima poudarek predvsem na podatkovni bazi, natančneje shranjenih procedurah in sprožilcih. Idejo za izdelek sem dobil med opravljanjem PUD, saj sem v sklopu PUD pomagal pri razvoju informacijskega sistema.

Opisal in predstavil bom shemo podatkovne baze, spletno aplikacijo, procedure, ki so potrebne za delovanje oz. praktično uporabo izdelka in težave na katere sem naletel pri razvijanju podatkovne baze in aplikacije, ter možne izboljšave končnega izdelka.

# **2. CILJI**

Pred samim pisanjem programske kode in načrtovanjem baze sem si postavil nekaj ciljev. To so:

- zbrati podatke za artikel in jih vpisati v podatkovno bazo,

- ustvariti račune in dokazila, s katerimi udeleženci dokažejo lastništvo,

- omogočiti samodejni izračun razlike v ceni pri košarici udeleženca,

- beležiti vse pomembne dogodke v bazo in tako ustvariti dnevnik oz. log,

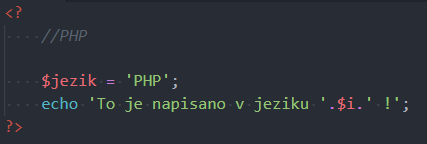
- izdelati spletno aplikacijo, ki bo služila kot vmesnik med bazo in uporabnikom,

- izdelati uporaben in intuitiven izdelek, ki bo služil svojemu namenu.

## **2.1 Uporabljene tehnologije**

### **2.1.1 PHP**

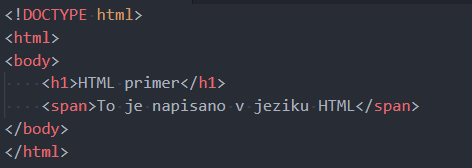
PHP je programski jezik, ki ga uporabljamo za pisanje spletnih aplikacij. PHP je skriptni jezik, ki se izvaja na strežniški strani in za njegovo funkcionalnost potrebujemo spletni strežnik. Strežnik ima vgrajen tolmač, ki prevede kodo in jo pošlje brskalniku v obliki HTML kode. S PHP-jem lahko pregledujemo podatke, jih spreminjamo, šifriramo, se povežemo na podatkovno bazo s spletno stranjo ter oblikujemo dinamično spletno aplikacijo.



Slika 1: Primer kode programskega jezika PHP (lasten vir)

### **2.1.2 HTML**

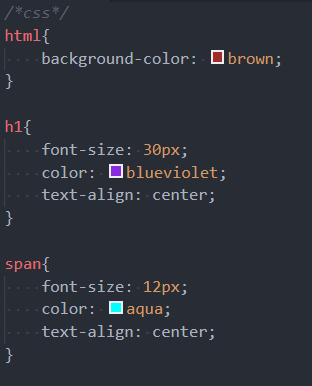
HTML je označevalni jezik, ki se uporablja za izdelovanje spletnih strani. Pogosto ga najdemo uporabljenega v kombinaciji z PHP-jem oziroma CSS-om, ki skrbi za postavitev in izgled HTML – elementov. HTML ukaze prepoznamo po elementih: večje, manjše in poševnica (primeri: <p> </p>).



Slika 2: Primer kode jezika HTML

### **2.1.3 CSS**

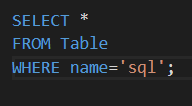
CSS je jezik, ki ga uporabljamo v kombinaciji z HTML. Je jezik oblikovanja spletnih strani, torej HTML elementom doda oblikovne lastnost.



Slike 3: Primer kode jezika CSS (lasten vir)

### **2.1.4 Firebird**

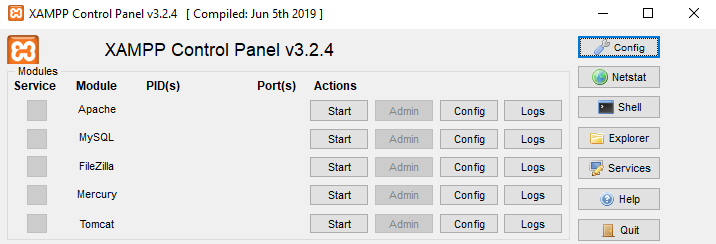
Firebird je odprtokodni sistem za upravljanje s podatkovnimi bazami. Jezik SQL je programski jezik 4. generacije, ki se uporablja za delo s podatki.



Slika 4: Primer kode jezika SQL (lasten vir)

### **2.1.5 XAMPP**

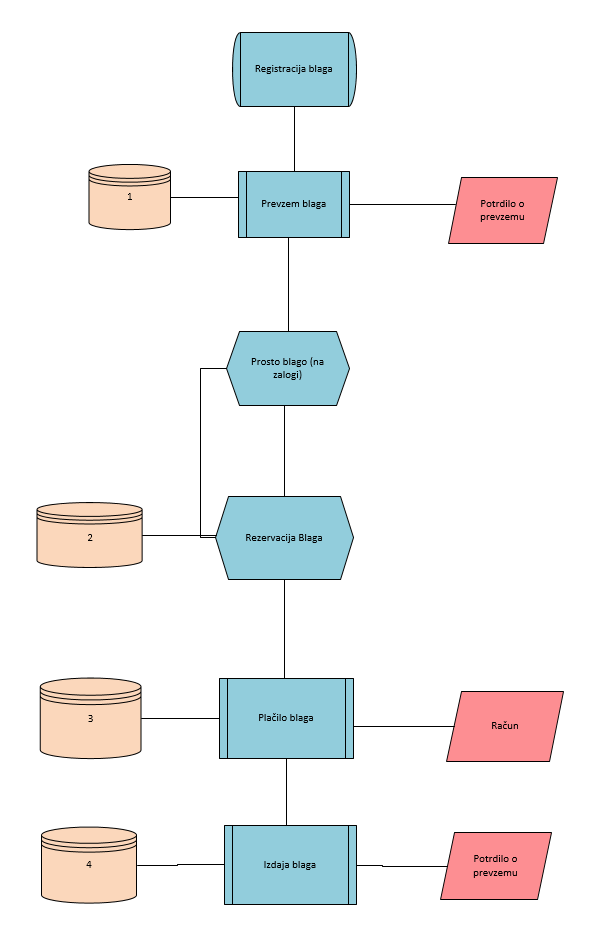
Xampp je brezplačno in odprtokodno programsko – razvojno okolje, ki je zaradi svoje kompleksnosti in preprostega postopka namestitve zelo razširjen med PHP razvijalci in razvijalci podatkovnih baz. Del orodja Xampp vsebuje Apache server in programske tolmače (angl. Interpreter), ki kodo prevedejo in jo pošljejo brskalniku v HTML obliki.



Slika 5: Nadzorna plošča orodja Xampp (lasten vir)

# **3. SHEMA POTEKA SEJMA**

Pred načrtovanjem podatkovne baze in spletne aplikacije, sem se najprej lotil analiziranja problema. Ravnal sem se po citatu: **"Prvo reši problem. Nato napiši kodo. "** (Johnson, 2009).  
Modri liki predstavljajo pomemben dogodek na sejmu. Rdeči liki predstavljajo dokument, ki se generira, ko artikel doseže ta dogodek. Oranžni liki predstavljajo zapis v podatkovno bazo (tabelo Event). Shema mi je pomagala pri načrtovanju baze in načrtovanju spletne aplikacije.

****

Slika 6: Shema poteka sejma (lasten vir)

# **4. SHEMA PODATKOVNE BAZE IN TABEL**

Podatkovno bazo, katera je napisana v jeziku SQL sem se odločil goniti v orodju Firebird, saj je sintaksa proceduralnega SQL-a veliko boljša v primerjavi z MySQL.

## **4.1 Tabela Member**

Ena vrstica v tabeli Member predstavlja enega udeleženca. Tabela vsebuje eno polje MemberID tipa varchar(5). Identifikacijska številka (ID), ki jo naključno generira baza, služi sledenju lastništva oz. interakcij z bazo.

## **4.2 Tabela Staff**

Ena vrstica v tabeli Staff predstavlja enega blagajnika oz. osebo z dostopom do baze in spletne aplikacije. Tabela vsebuje polja:

* StaffID varchar(5), ID, ki ga naključno generira baza služi sledenju interakcij,
* S\_name varchar(45), ime blagajnika,
* S\_surname varchar(200), priimek blagajnika.

## **4.3 Tabela Ptype**

Ena vrstica v tabeli Ptype predstavlja en tip izdelka. Tabela vsebuje polja:

* PtypeID varchar(5), ID, ki ga naključno generira baza služi sledenju interakcij,
* T\_Name varchar(50), ime tipa izdelka,
* T\_Descr varchar(200), opis tipa izdelka,
* T\_commision float, stopnja provizije,
* Tax float, davčna stopnja.

## **4.4 Tabela Product**

Ena vrstica v tabeli Product predstavlja en artikel. Tabela vsebuje polja:

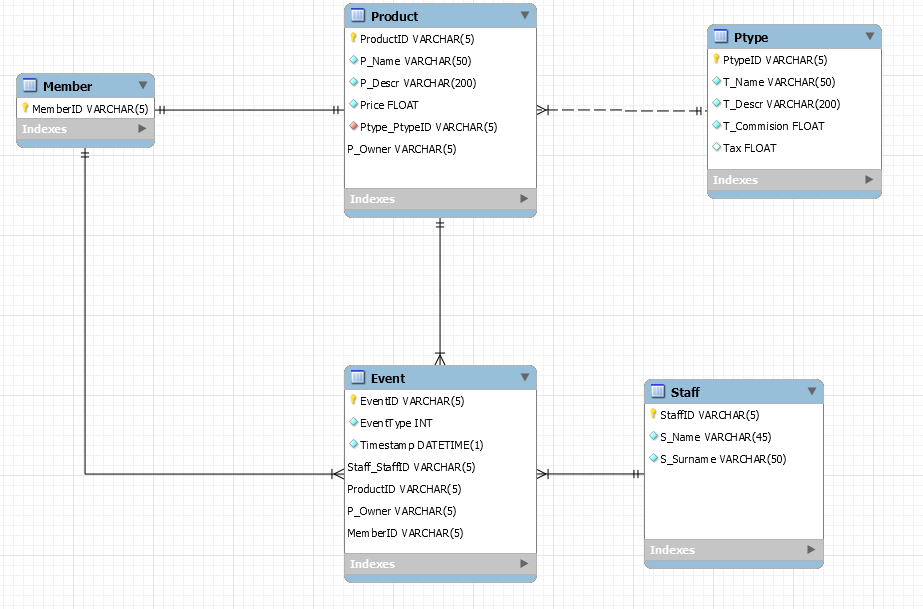
* ProductID varchar(5), ID, ki ga naključno generira baza služi sledenju interakcij,
* P\_Name varchar(50), ime izdelka,
* P\_Descr varchar(200), opis izdelka,
* Price float, cena izdelka,
* PtypeID varchar(5), ID tipa izdelka (tabela Ptype),
* P\_Owner varchar(5), ID lastnika izdelka (tabela Member).

## **4.5 Tabela Event**

Ena vrstica v tabeli Event predstavlja eno beleženje dogodka. Tabela vsebuje polja:

* EventID varchar(5), ID, ki ga naključno generira baza (številka transakcije).
* Eventtype int, tip transakcije.
* Time\_Stamp TIMESTAMP, čas transakcije.
* StaffID varchar(5), ID uslužbenca (tabela Staff).
* ProductID varchar(5), ID izdelka (tabela Product).
* P\_Owner varchar(5), ID lastnika izdelka (tabela Member).
* MemberID varchar(5), ID kupca (tabela Member).

## **4.6 Shema podatkovne baze**



Slika 7: ER model podatkovne baze (lasten vir)

## **4.7 Shranjene procedure**

Podatkovna baza ima 14 shranjenih procedur. Shranjene procedure zagotavljajo pravilne izpise podatkov, ki jih potrebuje uporabnik, večjo varnost in hitrejše delovanje baze. Skrbijo tudi za pravilne vnose v podatkovno bazo, ter izvajanje poslovnih pravil.

### **4.7.1 Procedura CASH\_LOG**

Procedura CASH\_LOG izpiše vse denarne transakcije in jih poveže s fizičnimi osebami, lastnik, kupec in blagajnik. Služi predvsem v analitične namene.

### **4.7.2 Procedura CASH\_SUM**

Procedura CASH\_SUM izpiše, koliko je bilo na sejmu skupno denarnega prometa, prihodkov (iz provizij) in plačanega davka. Služi predvsem v analitične namene.

### **4.7.3 Procedura FIND\_PRODUCT**

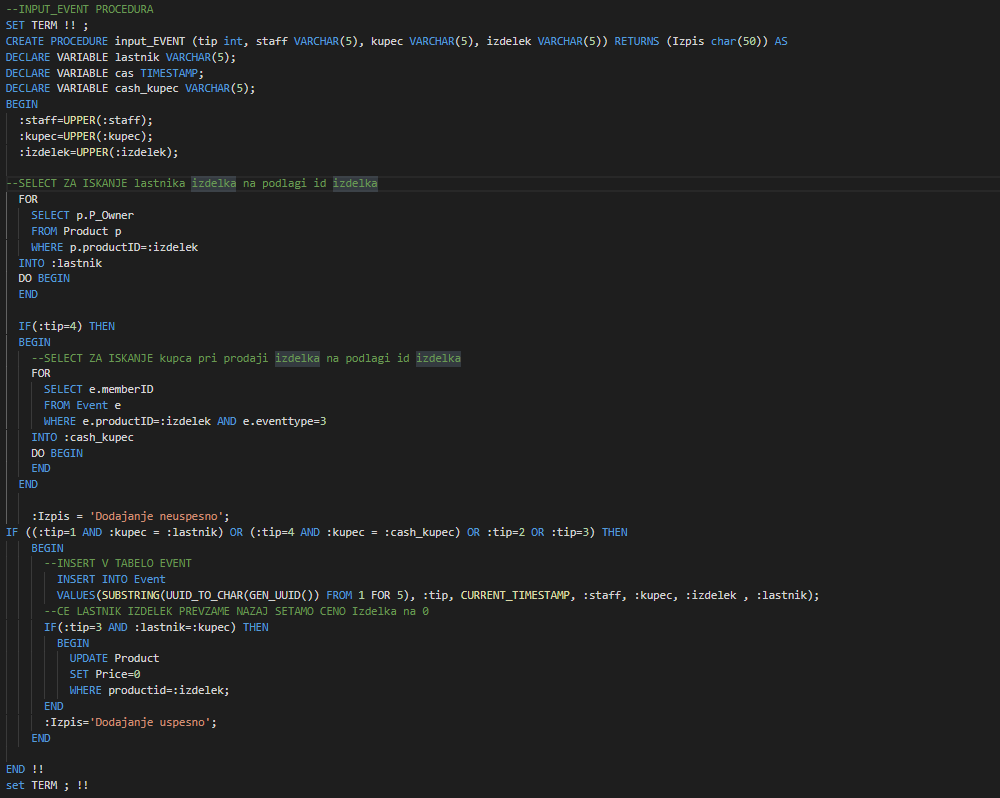
Procedura FIND\_PRODUCT izpiše ID izdelka, ime izdelka, ceno izdelka in ID lastnika izdelka na podlagi ID-ja izdelka. Ima splošen namen in se veliko uporablja.

### **4.7.4 Procedura GEN\_MEM**

Procedura GEN\_MEM naključno generira x število vpisov v tabelo Member. Ker udeležence na sejmu zaznamuje samo MemberID v tabeli Member jih lahko vnaprej naključno generiramo in tako zmanjšamo čakalne vrste.

### **4.7.5 Procedura INPUT\_EVENT**

Procedura INPUT\_EVENT dodaja zapise v tabelo Event. V tabeli Event se beležijo vsi pomembni dogodki.



Slika 8: Procedura INPUT\_EVENT (lasten vir)

### **4.7.6 Procedura INPUT\_PRODUCT**

Procedura INPUT\_PRODUCT dodaja izdelke v podatkovno bazo (tabela Product).

### **4.7.7 Procedura INPUT\_PTYPE**

Procedura INPUT\_PTYPE dodaja tipe izdelkov v podatkovno bazo (tabela Ptype).

### **4.7.8 Procedura INPUT\_STAFF**

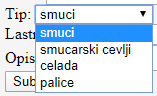
Procedura INPUT\_STAFF dodaja uporabnike v podatkovno bazo (tabela Staff).

### **4.7.9 Procedura PP\_FLOW**

Procedura PP\_FLOW izpiše število aktivnih udeležencev in število izdelkov na sejmu. Služi predvsem v analitične namene.

### **4.7.10 Procedura PTYPE\_OUT**

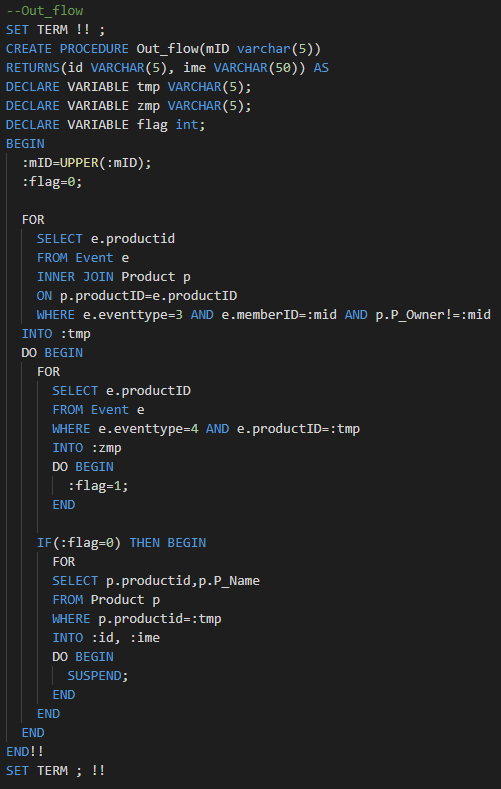
Procedura PTYPE\_OUT izpiše vse tipe izdelkov. Uporablja se pri vnašanju izdelkov in omogoča uporabniku izbiranje med tipi izdelkov.



Slika 9: Uporaba procedure PTYPE\_OUT (lasten vir)

### **4.7.11 Procedura OUT\_FLOW**

Procedura OUT\_FLOW izpiše izdelke udeleženca sejma (MemberID), ki so pripravljeni na izdajo.



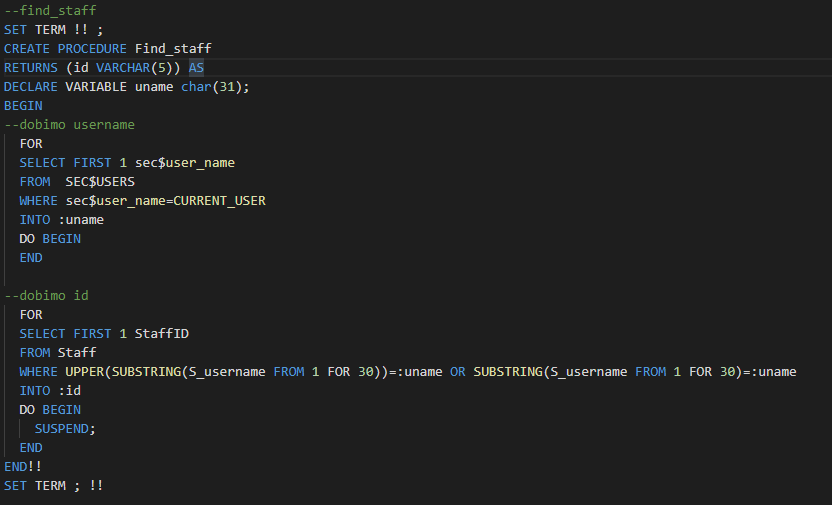
Slika 10: Procedura OUT\_FLOW (lasten vir)

### **4.7.12 Procedura STAFF\_LOGIN**

Procedura STAFF\_LOGIN vrne ID, ime in priimek blagajnika s pomočjo uporabniškega imena. Procedura se uporablja pri prijavi uporabnika v sistem.

### **4.7.13 Procedura FIND\_STAFF**

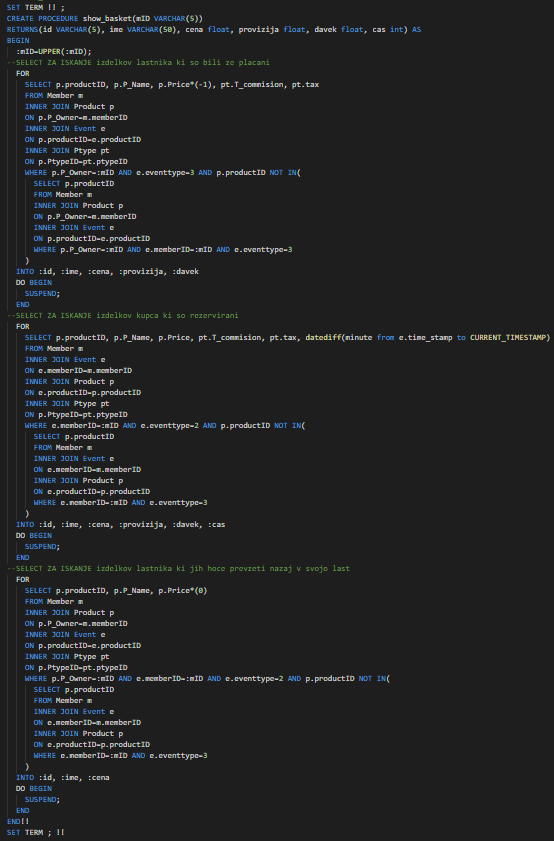
Procedura FIND\_STAFF vrne ID blagajnika oz. uslužbenca. S pomočjo sistemske spremenljivke v bazi CURRENT\_USER izpiše ID trenutno prijavljenega uporabnika oz. blagajnika.



Slika 11: Procedura FIND\_STAFF (lasten vir)

### **4.7.14 Procedura SHOW\_BASKET**

Procedura SHOW\_BASKET prikaže košarico udeleženca sejma. V košarici so artikli, ki jih je udeleženec rezerviral in jo mora plačati. V primeru, da je rezerviral svoj artikel se cena postavi na 0. V košarici so prisotni artikli, katerih udeleženec je lastnik in so bili že prodani. V tem primeru je udeležencu potrebno izplačati denar.­



Slika 12: Procedura SHOW\_BASKET (lasten vir)



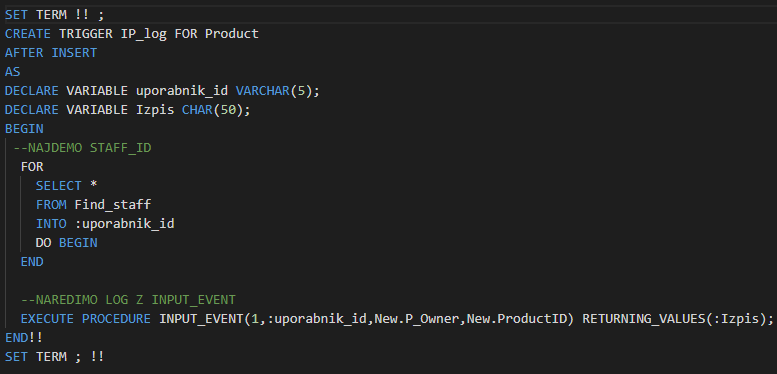
Slika 13: Uporaba procedure SHOW\_BASKET (lasten vir)

## **4.8 Sprožilci**

Podatkovna baza ima 2 sprožilca. Sprožilci preverjajo na pravilnost vnosov, ter zagotavljajo, da se poslovna pravila pravilno izvajajo.

### **4.8.1 Sprožilec IP\_LOG**

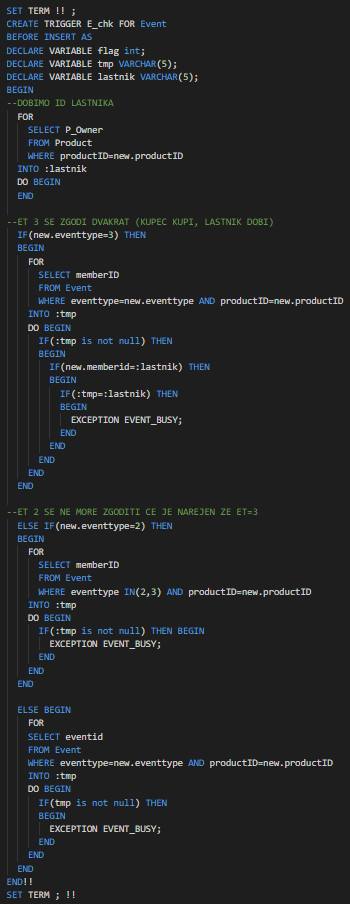
Sprožilec IP\_LOG, skrbi da se ob vnosu izdelka v tabelo Product izvede procedura INPUT\_EVENT, ki v tabelo Event zabeleži dogodek.



Slika 14: Sprožilec IP\_LOG (lasten vir)

### **4.8.2 Sprožilec E\_CHK**

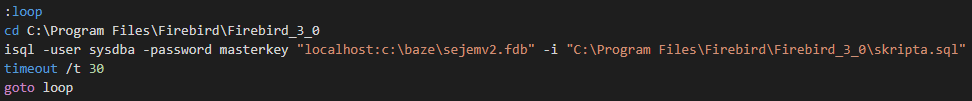
Sprožilec E\_CHK, skrbi za pravilne vnose v tabelo Event in s tem zagotovi pravilno beleženje pomembnih dogodkov.



Slika 15: Sprožilec E\_CHK (lasten vir)

## **4.9 Pomožna skripta**

Pri implementaciji rezervacije izdelka, sem naletel na manjšo težavo. Izdelke, ki so rezervirani več kot 20 min je potrebno sprostiti nazaj v prodajo. Podatkovna baza, ne omogoča klica neke skripte v obliki sprožilca, funkcije, procedure itd. na neko časovno obdobje. Problem sem rešil tako, da sem napisal .bat skripto, ki se poveže na podatkovno bazo in zažene SQL skripto vsakih 30 sekund.



Slika 16: Bat skripta (lasten vir)

# **5. SPLETNA APLIKACIJA**

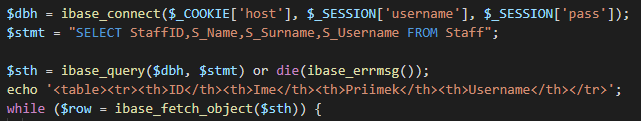
Spletna aplikacija služi, kot vmesnik med uporabnikom in podatkovno bazo. Spletno aplikacijo sem se odločil napisati v jeziku PHP, saj smo se o njem učili pri predmetu NSA in mi je zato najbolj poznan. Pri pisanju spletne aplikacije sem naletel na težavo pri povezovanju z mojo podatkovno bazo v Firebirdu. Moral sem uporabiti gonilnik in dodati sistemsko spremenljivko v okolje Windows. PHP aplikacijo sem poganjal v Apache strežniku, znotraj orodja Xampp.

## **5.1 Zakaj spletna aplikacija**

Za spletno aplikacijo sem se odločil, ker ima mnogo prednosti pred programom, ki se izvaja na lokalnem računalniku. Spletna aplikacija je neodvisna od operacijskega sistema in nastavitev operacijskih sistemov. Prav tako zahteva manj administracijskega dela, saj jo lahko enostavno posodobimo.

## **5.2 Povezovanje z bazo v PHP**

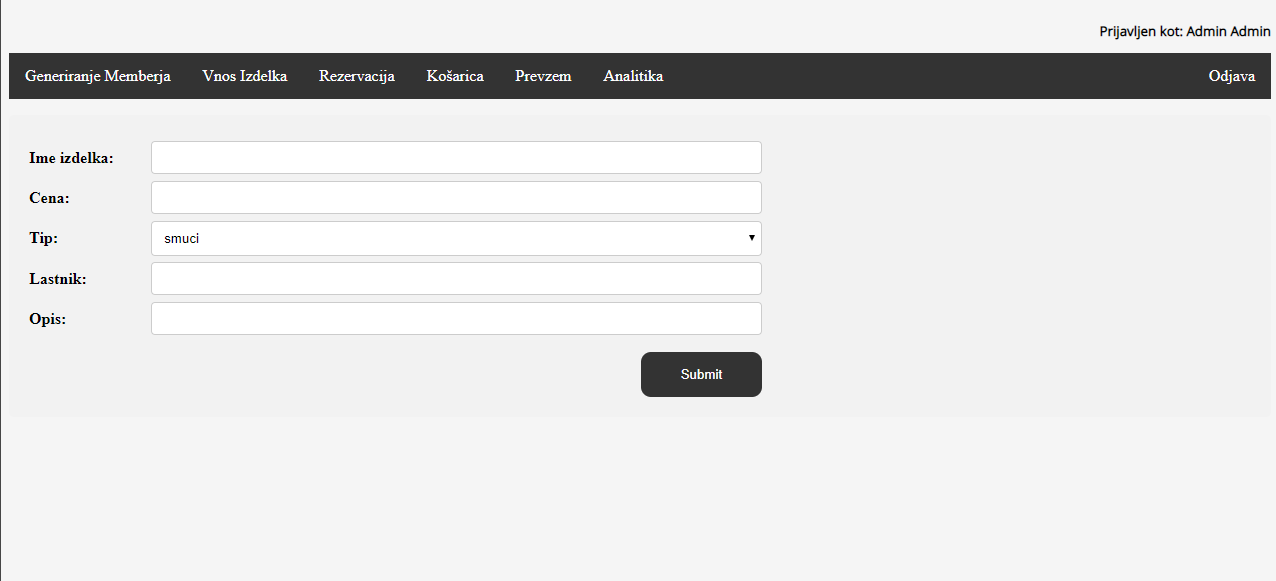
Za povezovanje sem uporabil knižnjico ibase. V spremenljivko $dbh sem shranil parametre s katerimi se bom uspešno povezal na bazo. Uporabniško ime in geslo sem shranil v spremenljivko seje ($\_SESSION), lokacijo strežnika pa v piškotek ($\_COOKIE). V spremenljivko $stmt sem sestavil in shranil SQL stavek. Nato sem ga izvedel s funkcijo ibase\_query($dbh, $stmt) in v while zanki pridobil podatke iz baze. Podatke iz baze prejmem v $row spremenljivko.



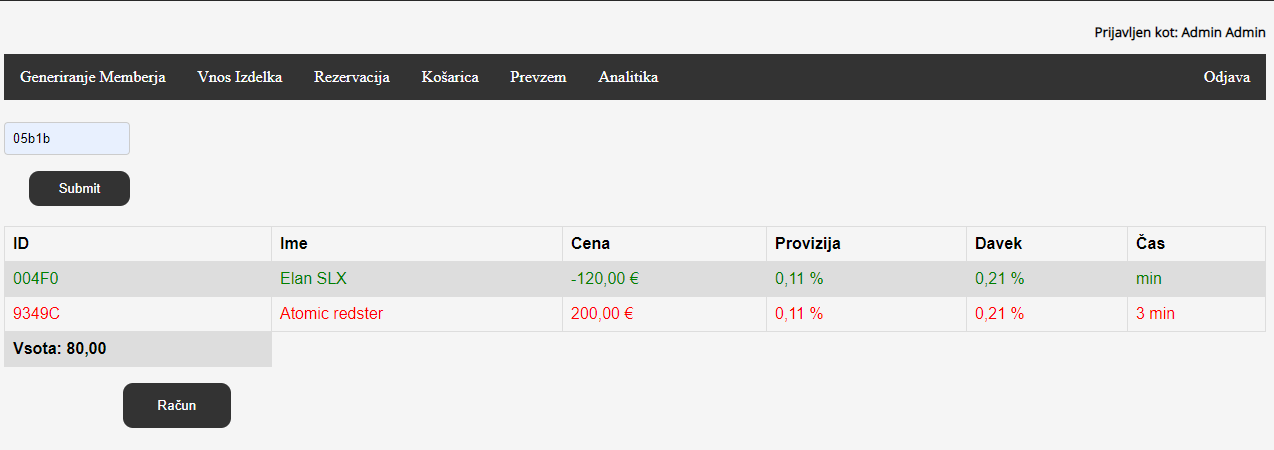
Slika 17: Povezava s podatkovno bazo (lasten vir)

## **5.3 Videz spletne aplikacije**

Celotna spletna aplikacija je osnovana na celostni grafični podobi. Njen videz je preprost in minimalističen. Spletna aplikacija je lahka in intuitivna za uporabo.



Slika 18: Forma za vnašanje izdelka (lasten vir)



Slika 19: Izpis košarice udeleženca (lasten vir)

## **5.4 Generiranje dokumentov**

Generiranje dokumentov, je bil eden izmed večjih ciljev izdelka. Dokumenti so izjemno pomembni, saj udeleženci z njimi dokazujejo lastništvo nad izdelkom. Dokumente sem generiral s pomočjo PHP-ja in procedur iz katerih sem črpal podatke iz podatkovne baze.

Ko blagajnik vpiše identifikacijsko številko udeleženca, se na ekran izriše njegova košarica, ter gumb za generiranje računa. Ob kliku na gumb se izvede plačilo in izriše račun, kateri se natisne in služi, kot dokaz o plačilu.



Slika 20: Račun (lasten vir)

# **6. UPORABA INFORMACIJSKEGA SISTEMA**

Uporabnik spletne aplikacije se prijavi z podanim uporabniškim imenom in geslom v začetnem oknu spletne aplikacije in se tako prijavi v spletno aplikacijo in podatkovno bazo. Pričaka ga spletna stran z navigacijsko ploščo. Ta nato izbere dogodek, ki ga želi opraviti. Odpre se mu forma v katero izpolni ustrezne podatke. Če se pri dogodku generira dokument se mu ta odpre in ga lahko natisne in izda udeležencu sejma. Ko uporabnik konča z delom pritisne gumb, ki ga odjavi.

Navigacijska plošča:

* **Vpis** – če uporabnik želi uporabljati informacijski sistem se potrebuje prijaviti. Okno za vpis je vstopno okno.
* **Generiranje udeleženca** – uporabnik vnese število udeležencev, ki jih želi generirati in izbiro potrdi z gumbom potrdi. Na zaslon se izpišejo identifikacijske številke udeležencev.
* **Vnos izdelka** – uporabnik vnese v formo podatke in potrdi izbiro z gumbom potrdi. Če je bil vnos uspešen se uporabniku nato odpre dokazilo o prevzemu katerega natisne.
* **Rezervacija** – uporabnik vnese v formo identifikacijsko številko kupca in izdelka, ter potrdi izbiro z gumbom potrdi. Na zaslon se izpiše sporočilo, če je bila rezervacija uspešna.
* **Košarica** – Uporabnik vnese identifikacijsko številko udeleženca in izbiro potrdi z gumbom potrdi. Na zaslon se izriše tabela z uporabnikovimi izdelki za katere mu je potrebno izplačati ali pa zaračunati določeno vsoto denarja. Izriše se tudi gumb z imenom račun, kateri ob uporabnikovem kliku opravi transakcijo in uporabniku odpre račun katerega natisne in izda udeležencu.
* **Prevzem** – Uporabnik vnese identifikacijsko številko udeleženca in izbiro potrdi z gumbom potrdi. Na zaslon se mu izpiše seznam izdelkov udeleženca, katere mu je potrebno izdati. Izriše se tudi gumb izdaja, kateri ob uporabnikovem kliku opravi transakcijo in izriše potrdilo o prevzemu.
* **Odjava** – ob kliku na gumb odjava, se uporabnik odjavi. Odpre se vstopno okno za prijavo.

# **7. MOŽNE IZBOLJŠAVE**

Podatkovna baza je zasnovana dobro in deluje optimalno. Procedure in sprožilci prav tako delujejo pravilno in optimalno.

Izboljšati bi bilo potrebno spletno aplikacijo, ki jo bi lahko prilagodili za mobilne telefone ter omogočili generiranje QR kod, s katerimi bi bilo lažje in hitrejše opravljati transakcije, saj bi se tako izognili tipkanju identifikacijskih številk. Izdelali bi lahko tudi spletno aplikacijo za udeležence, kjer bi lahko beležili svoje rezervacije, nakupe in spremljali njihove artikle.

# **8. ZAKLJUČEK**

S končnim izdelkom sem zadovoljen. Izpolnjuje vse cilje, ki sem si jih zastavil in odgovarja na prvoten problem. Izdelek predstavlja praktičen primer uporabe baze podatkov, ter črpanje uporabnih podatkov iz nje. Uporabil sem veliko programskih jezikov in orodij. Pri načrtovanju in izdelavi sem se veliko zabaval, ter tudi ogromno naučil.. Izdelek, bi lahko nadgradil z spletno aplikacijo za udeležence sejma.

# **9. VIRI**

* PHP dokumentacija: <https://www.php.net/manual/en/language.references.php> (26. 3. 2020)
* HTML dokumentacija: <https://www.w3schools.com/tags/> (24. 3. 2020)
* CSS dokumentacija: <https://www.w3schools.com/cssref/> (26. 3. 2020)
* Firebird dokumentacija:

<https://firebirdsql.org/file/documentation/reference_manuals/fblangref25-en/html/fblangref25.html> (26. 3. 2020)

## **9.1 Citati**

* John Johnson. (2009) Pridobljeno s strani: <https://www.defprogramming.com/quotes-by/john-johnson/> (28. 3. 2020)

**Izjava o avtorstvu**

Izjavljam, da je strokovno poročilo Podporni informacijski sistem v celoti moje avtorsko delo, ki sem ga izdelal samostojno s pomočjo navedene literature in pod vodstvom mentorja.

28. 3. 2020 Nejc Vrčon Zupan