Sustav daljinskog nadzora i upravljanja ribnjakom - programska podrška

Crnjac, Filip

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:453697

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni diplomski studij

Sustav daljinskog nadzora i upravljanja ribnjakom – programska podrška

Diplomski rad

Filip Crnjac

Mentor: Prof. Dr. Sc. Željko Hocenski

Osijek, 2016.

Sadržaj

1. UVOD 1
2. RIBOGOJSTVO 2
3. SCADA SUSTAV 4
4. TEHNOLOGIJE 9
4.1MySQL 9
4.1.1 Vrste podataka u MySQL-u 9
4.1.2 Osnovni pojmovi u projektiranju baze podataka 12
4.1.3 Osnovne naredbe za upravljanje podacima u SQL-u 12
4.2 PHP 14
5. PROGRAMSKO RJEŠENJE 17
5.1 PROGRAMSKA IZVEDBA U PHP-u 17
5.1.1.Login.php 18
5.1.2 Data.php 19
5.1.3 Ispis podataka 23
5.2 STRUKTURA BAZE PODATAKA 25
5.2.1 Spajanje na bazu podataka 28
5.3 TESTIRANJE PROGRAMA 29
6. ZAKLJUČAK 32
LITERATURA 33
SAŽETAK 34
ABSTRACT 35
PRILOZI NA CD-U 37

1. UVOD

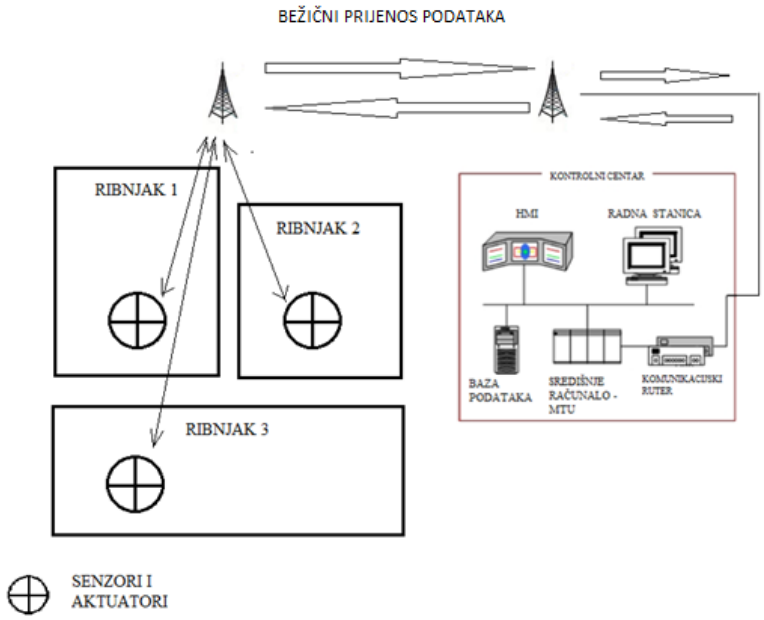
Tisućama godina riba i plodovi mora važan su i neizostavan dio ljudske prehrane. Riblje meso obiluje bjelančevinama koje sadrže esencijalne aminokiseline potrebne organizmu za normalno funkcioniranje. Jedan od najznačajnijih sastojaka ribe su i masti, od kojih valja izdvojiti omega-3 masne kiseline koje pozitivno djeluju na krvožilni sustav te su bitne za zdrav razvoj djeteta u majčinoj utrobi. Visoka nutritivna vrijednost razlog je tome da se riba lovi i jede još od razdoblja prapovijesti, a najstarija zabilješka o ribi kao hrani Homo sapiensa stara je više od 380 000 godina. Ribolovna kultura bila je razvijena uz Nil i obalu Sredozemnog mora u Egiptu, o čemu svjedoče brojni reljefi iz 2 500 p.n.e. Rimljani su prvi počeli graditi ribnjake u svrhu čuvanja ulovljene ribe, no primijetili su da se ribe u tim ribnjacima razmnožavaju, pa time započinje uzgoj ribe. U srednjem vijeku ribogojstvo se usavršava gradnjom ribnjaka uz dvorce za privilegirane društvene staleže te se prvenstveno uzgaja šaran. Razvoj tehnike omogućuje i razvoj ribogojstva te se početkom 20.-tog stoljeća grade tzv. moderni ribnjaci koji zahtijevaju visokostručne kadrove.

U ovom radu ribnjak ćemo promatrati kao spoj proizvodnog pogona (sa svim elementima koji to podrazumijevaju: nadzor, upravljanje, obrada i spremanje podataka, alarmi) i ekosustava (kao kompleksnog jedinstva živih bića i prostora u kojem žive). Upravljanjem i kontroliranjem pojedinih elemenata želimo omogućiti postizanje optimalnih uvjeta za uzgoj riba. Pomoću takvog sustava također želimo optimizirati troškove proizvodnje u cilju povećanja efikasnosti i učinkovitosti koji bi kao krajnji rezultat imali kvalitetan proizvod. Sustav kontrole i nadzora realizirat ćemo kao interaktivnu web stranicu koristeći PHP programski jezik za prikaz sustava i MySQL kao bazu za pohranu podataka. Takva programska podrška se naziva SCADA koju ćemo kasnije detaljnije opisati.

2. RIBOGOJSTVO

Ribnjaci su umjetno izgrađena ili prilagođena jezera koja se po želji mogu prazniti ili puniti vodom. Postoje razne kvalifikacije ribnjaka, no obično se dijele na hladnovodne (pastrva) ili toplovodne (šaran). Tip ribnjaka također određuju klimatski uvjeti i kvaliteta vode. Kvaliteta vode u ribnjacima se dijeli na tri dijela: fizičko, kemijsko i biološko stanje kvalitete. Svi ti čimbenici direktno utječu na tempo i intenzitet rasta ribe. Osnovna mjerenja se odnose na temperaturu vode, dubinu i prozirnost. Optimalna temperatura za uzgoj se kreće od 20 do 27 stupnjeva Celzijevih. U kemijskom sastavu vode najbitnija je količina otopljenog kisika i pH vrijednost vode. Količine ostalih kemijskih elemenata (ugljik, sumpor, kalij, kalcij, magnezij...) se vrši povremeno *(tablica 2.2).*

U ribnjaku kao proizvodnom pogonu najvažniju ulogu ima produktivnost od koje zavisi ukupno proizvedena količina ribe po hektaru. U tu analizu također ulaze drugi parametri kao što su prosječna masa ribe, kvaliteta mesa i postotni sadržaj masti. Ribnjake tako možemo podijeliti i na: ribnjake visoke produktivnosti i ribnjake niske produktivnosti. Produktivnost je izražena kao proizvedena količina biomase (ribe) na jedinici površine (hektar).



Slika 2.1 – Blok shema sustava ribnjaka kojima se upravlja

Kiselost	pH 6-9
Arsen	<440 µg/L
Alkalnost	>20 mg/L (as $CaCO_3$)
Aluminij	<0.075 mg/L
Amonijak	<O.O2mg/L
Kadmij	<0.005 mg/L
Kalcij	>5 mg/L
Ugljkov dioksid	<5–10 mg/L
Klorid	>4.0 mg/L
Klor	<0.003 mg/L
Bakar	<0.03
Sumporovodik	<0.003 mg/L
Željezo	<0.1 mg/L
Olovo	<0.02 mg/L
Živa	<0.0002 mg/L
Dušik	<1.0 mg/L
Nitrit	<0.1 mg/L
Kisik	4 mg/L
Selen	<0.01 mg/L
Cink	<0.005 mg/L

Tablica 2.2 Optimalni parametri vode u ribnjaku

3. SCADA SUSTAV

SCADA (Supervisory Control and Data Acquisition) je programska podrška za prikupljanje, nadzor i upravljanje podataka. Ovi sustavi u različitim oblicima postoje još od 60-tih godina 20. stoljeća, a od 90-tih godina 20. stoljeća doživljavaju veliku ekspanziju sa pojavom sve bržih i efikasnijih računalnih i mikrokontrolnih uređaja. Mogu se koristiti za jednostavan nadzor (npr. temperature, vlažnosti zraka, tlaka) ili za kompleksan nadzor i upravljanje (npr. proizvodni procesi u tvornicama ili regulacija prometa). SCADA sustavi se sastoje od uređaja za primanje i slanje signala, kontrolnih uređaja, sučelja čovjek-stroj (engl. HMI - Human Machine Interface), komunikacije, baze podataka i programske podrške.
Osnovne funkcije SCADA sustava su:

- Prikupljanje mjernih i drugih podataka
- Baza podataka – pohrana i čuvanje pohranjenih vrijednosti procesnih veličina
- Obrada mjernih podataka
- Registriranje alarma i događaja – npr. registriranje važnih promjena u procesu i akcija operatera
- Slanje određenih informacija na zadana odredišta
- HMI (Human Machine Interface) - grafički prikaz stanja procesa i procesnih zbivanja te rukovanja operaterskim naredbama
- Različite dopunske funkcije

Razvojem telekomunikacijske infrastrukture omogućava se prikupljanje podataka sa širokog područja te slanje podataka na vrlo udaljena odredišta. Komunikacija je moguća i preko Interneta.
U osnovi postoje tri vrste SCADA sustava:

- Osnovna SCADA – jedan namjenski (ograničen) proces s pripadnom upravljačkom jedinicom (npr. upravljanje temperature u prostoriji, robotski manipulator)
- Integrirana SCADA – složen proces sa višestrukim priključenim jedinicama SCADA-e i jak komunikacijski sustav za povezivanje s upravljačkim jedinicama
- Umrežena SCADA – složeni tehnički proces s distribuiranom upravljačkom i nadzornom jedinicom, složen komunikacijski sustav za povezivanje SCADA-e unutar nadzorne jedinice.

SCADA program se odvija na središnjem (master/server) računalu kod osnovne i integrirane SCADA-e, ali moguće je odvijanje i u više računala, što je slučaj kod umrežene SCADA-e.

SCADA komponente:

MTU (Master Terminal Unit)

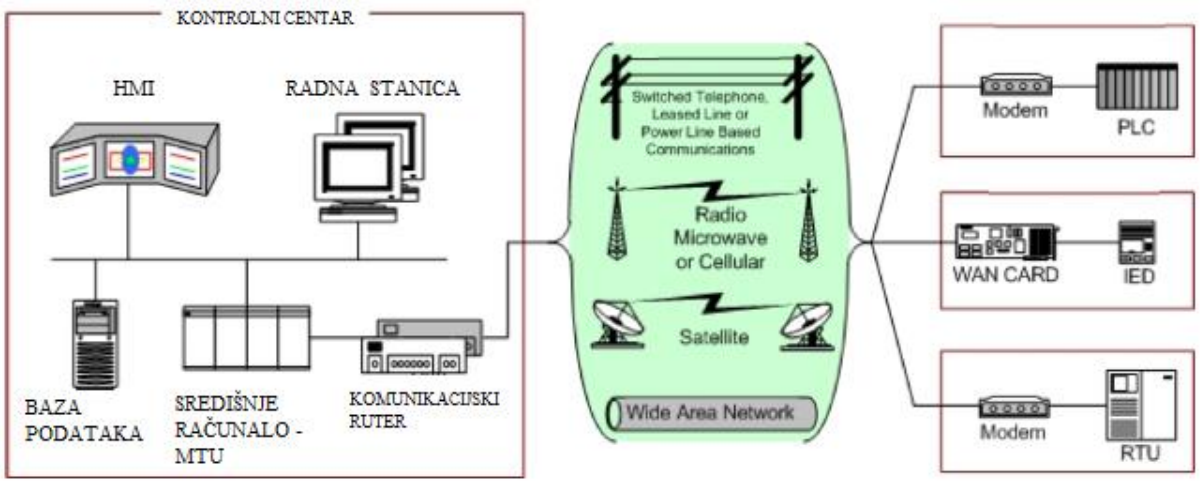
- Središnje računalo SCADA sustava (obično smješteno u središnjoj kontrolnoj dvorani)
- Preuzima sve prikupljene podatke, arhivira ih, obrađuje te prosljeđuje drugim sustavima
- Nadgleda sve druge uređaje preko kojih se prikupljaju podaci

RTU (Remote Terminal Unit)

- Pomoćni uređaji distribuirani po postrojenju, namijenjeni prikupljanju podataka sa IED-ova

IED (Intelligent Electronic Device)

- Uređaj koji uz izvođenje zadanog programa ujedno pruža i sučelje u podatkovnoj komunikaciji sa SCADA-om
- Raspodijeljeni po postrojenju (PLC, industrijski PC, digitalni regulatori, senzori)



Slika 3.1 – Blok shema SCADA sustava

U SCADA sustavima se jako vodi računa o skalarnosti (mogućnosti proširenja sustava) i zalihosti (pojava koja se javlja u sustavima za upravljanje podatkovnim bazama koje sadrže polje koje se ponovilo u dvjema ili većem broju tablica). Kako je fokus ovog rada sučelje, spomenut ćemo dva glavna sučelja koja susrećemo u SCADA sustavima:

- HMI-P (HMI za proces) – operatersko sučelje – sučelje namijenjeno tehnolozima i operaterima pogona. Prikazuje stanje pogona (instrumenti, ekrani, prikaz poruka održavanja), omogućuje direktno upravljanje (pritiskom na tipke, dodirnik), administraciju (sigurnosne razine, pristup) i snimanje svih manipulacija.
- HMI-E (HMI za inženjere) – inženjersko sučelje – omogućava konfiguraciju pogona (povezivanjem novih uređaja, programa), definiranje protokola i upravljačkih sekvenci, postavljanje referentnih vrijednosti, limita i koeficijenata. Ovo sučelje se najviše koristi tijekom izgradnje pogona, podešavanja i puštanja u rad te naknadnih modifikacija procesa.

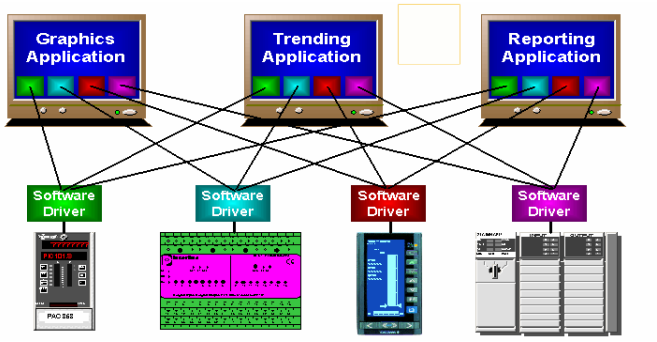
U programskom sučelju najvažniji je **OPC (Open Process Control).**OPC temelji se na OLE (Object Linking and Embedding) standardu kojeg je izradio Microsoft. Dizajniran je tako da bude proširiv drugim komponentama što je, uz kompatibilnost sa Windows okruženjem te već postojeće baze korisnika OLE standarda dovelo do popularizacije OPC standarda. OPC koristimo kako bismo uskladili uređaje raznih proizvođača i omogućili jednostavnu upotrebu i kontrolu procesa. Svaki element u sustavu povezan je jednim jedinstvenim pravilom.
Ovim putem uklanja se niz problema:

- Zbog nekompatibilnosti svaka aplikacija zahtijevala je svoj driver što je znatno usporavalo i otežavalo povezivanje klijenata i poslužitelja
- Proizvođači su zasebno trebali razvijati drivere
- Svaka promjena u sklopovlju značila je veliki napor pa i nemogućnost osposobljavanja sustava
- Dva različita klijenta ne mogu pristupiti istom uređaju zbog različitih drivera

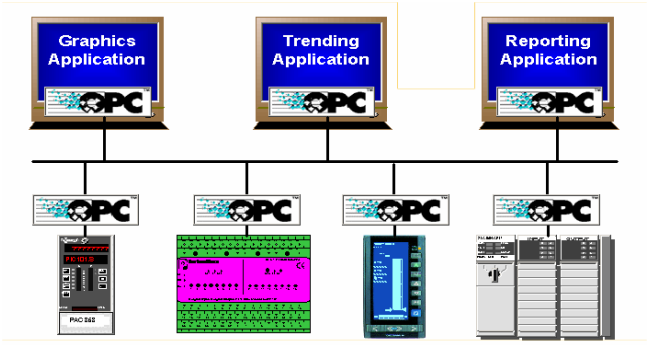
OPC je skup detaljno razrađenih protokola koji su odnedavno postali nezaobilazni u proizvodnji softvera za kontrolu najsuvremenijih industrijskih procesa. Glavna karakteristika

specifikacija OPC-a je da svaki poslužitelj treba pružati mogućnost razvoja tri osnovna sučelja koja su temelj cijelog sustava SCADA, a to su:

- **OPC Dana Access Server –** adresira prikupljene procesne varijable, a glavni klijenti su vizualizacija i posluživanje te registracija podataka
- **OPC Alarm & Event Server**– specificira informacije o alarmima i događajima te uvjete za njihovo slanje, a glavni klijenti su zapisivači alarma i događaja
- **OPC Historical Data Server –** specificira način na koji se povijesni podaci mogu dohvatiti iz zapisa u memoriji, prikupiti i obraditi, a glavni klijenti su prikazi trendova i histograma
- **OPC-DX - Data eXchange –** definira kako OPC server razmjenjuje podatke sa drugim OPC serverima
- **OPC-XML - XML Data Access –** omogućava razmjenu i obradu podataka kroz različite operacijske sustave (Windows, Unix, Solaris...)



Slika 3.2. - dosadašnji način povezivanja uređaja



Slika 3.3. - OPC struktura povezivanja uređaja

Na gornjim slikama prikazana je razlika stare komunikacijske strukture (*Slika 3.2*) i novog koncepta OPC strukture (*Slika 3.3*). Aplikacijski programi moraju omogućiti komunikaciju između temeljnih uređaja na razini automatizacije u pogonu i ostalih aplikacija. Problemi takvih povezivanja često postaju aktualniji ukoliko je sustav koncipiran kao decentralizirani. Ključ problema leži u činjenici da različiti proizvođači isporučuju uređaje s programskim sučeljima koji nisu standardizirani. U nedostatku standarda isporučitelji opreme isporučivali su opise komunikacijskih protokola, i API programske pakete (Application Programming Interfaces). Cijene integracija različitih podsustava znatno su rasle zbog činjenice da je za svaki uređaj bilo potrebno izrađivati posebno programsko sučelje. Statistike pokazuju da je za programski razvoj tipične nadzorno – kontrolne aplikacije bilo potrebno utrošiti oko 25-30% inženjerskog razvojnog vremena za pisanje programskog sučelja. Integratori sustava trošili su vrlo mnogo razvojnog vremena za takve poslove te je konačna cijena sustava bila skupa kao i održavanje jer je i ono zahtijevalo specijalistička znanja. Krajnji kupac kupovao je skup sustav koji se veoma skupo održava i proširuje.

4. TEHNOLOGIJE

Tehnologije koje su korištene u izradi ovo projekta su:

-MySQL

-PHP

4.1MySQL

MySQL (skraćenica za SQL: Structured Query Language) je sustav za upravljanje relacijskim bazama podataka. Originalno je razvijen 70-tih godina u *IBM Research Laboratory, San Jose, California* u okviru relacijskog sustava za upravljanje bazama podataka *System R*. SQL je nakon toga standardiziran te je postao referentni jezik za relacijske baze podataka.

MySQL može:

- izvršavati upite u bazi podataka,
- upisivati, brisati i ažurirati podatke
- kreirati, brisati i ažurirati tablice i baze podataka
- stvarati dozvole pristupa za pojedine tablice ili baze
- stvarati procedure (potprograme) unutar baze podataka

Jedna od mogućih definicija baze podataka glasi da je to zbirka zapisa pohranjenih u računalu na sustavan način, tako da joj se računalni program može obratiti prilikom odgovaranja na problem. Svaki se zapis za bolji povratak i razvrstavanje obično prepoznaje kao skup elemenata (činjenica) podataka. Predmeti vraćeni u odgovoru na upitnike postaju informacije koje se mogu koristiti za stvaranje odluka koje bi inače mogle biti mnogo teže ili nemoguće za stvaranje. Računalni program korišten za upravljanje i ispitivanje baze podataka nazvan je sustav upravljanja bazom podataka (SUBP). Svojstva i dizajn sustava baze podataka uključeni su u proučavanje informatičke znanosti [2].

4.1.1 Vrste podataka u MySQL-u

MySQL sadrži tri osnovne vrste podataka:

- Tekstualni
- Numerički
- Datum/vrijeme

Tekstualni tipovi podataka

CHAR()	String fiksne dužine od 0 do 255 znakova.
VARCHAR()	String promjenjive dužine od 0 do 255 znakova
TINYTEXT	String maksimalne dužine 255 znakova.
TEXT	String maksimalne dužine 65535 znakova.
BLOB	String maksimalne dužine 65535 znakova.
MEDIUMBLOB	String maksimalne dužine 16777215 znakova.
MEDIUMTEXT	String maksimalne dužine 16777215 znakova.
LONGTEXT	String maksimalne dužine 4294967295 znakova.
LONGBLOB	String maksimalne dužine 4294967295 znakova

Tablica 4.1.1. tekstualni tipovi podataka u MySQL-u

Dužina stringa se upisuje u zagrade, npr. tip CHAR(100) može primiti maksimalno 100 znakova, dok tip VARCHAR(100) također može primiti 100 znakova, ali kreira polje dužine unesenog stringa za razliku od CHAR-a kojem je polje uvijek iste (maksimalne duljine). Zbog toga je tip CHAR malo brži, dok tip VARCHAR zauzima manje memorije. Isto se odnosi na tipove TEXT i BLOB.

Datum i vrijeme

DATE	YYYY-MM-DD
DATETIME	YYYY-MM-DD HH:MM:SS
TIMESTAMP	YYYYMMDDHHMMSS
TIME	HH:MM:SS

Tablica 4.1.2. vremenski tipovi podataka

DATETIME tip podataka prikazuje određene datume i vrijeme s točnošću u sekundama. Dozvoljeni raspon je „1000-01-01 00:00:00 do 9999-12-31 23:59:59“. TIMESTAMP upisuje zadnju izmjenu sadržaja tablice, dok DATE pohranjuje podatke o datumu, a TIME o vremenu. Formati zapisa su prikazani u tablici 4.1.2.

Numerički tipovi podataka

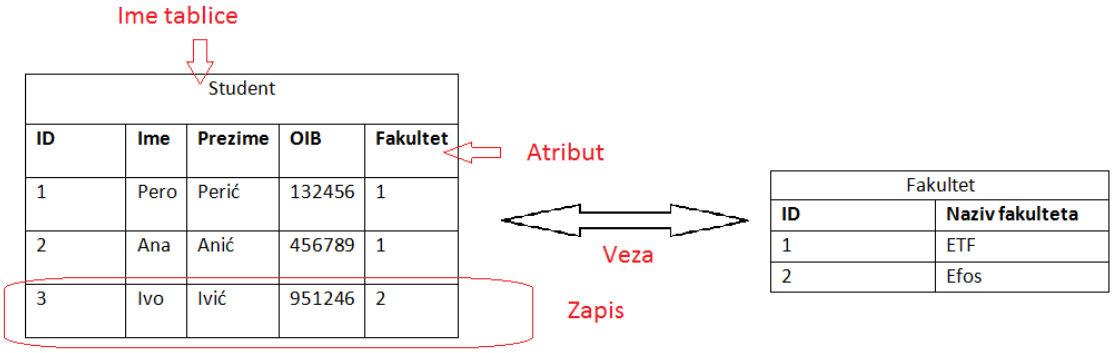
Numerički tip podataka INT pohranjuje cijele negativne i pozitivne brojeve u različitim rasponima (TINYINT, MEDIUMINT, SMALLINT i BIGINT – vidi tablicu 4.1.3). Tip FLOAT je tip podatka koji pohranjuje brojeve s decimalnom točkom, dok je DOUBLE inačica FLOAT tipa podatka sa većim rasponom.

TINYINT	-128 do 127 normal
	0 do 255 UNSIGNED
SMALLINT	-32768 do 32767 normal
	0 do 65535 UNSIGNED
MEDIUMINT	-8388608 do 8388607 normal
	0 do 16777215 UNSIGNED
INT	-2147483648 do 2147483647 normal
	0 do 4294967295 UNSIGNED
BIGINT	-9223372036854775808 do 9223372036854775807 normal
	0 do 18446744073709551615 UNSIGNED
FLOAT	3.402823466E+38 do -1.175494351E-38, 0 i od 1.175494351E-38 do 3.402823466E+38
DOUBLE	1.7976931348623157E+308 do -2.2250738585072014E-308, 0 i od 2.2250738585072014E-308 do 1.7976931348623157E+308
DECIMAL	DOUBLE spremljen kao string i to točno određene dužine..

Tablica 4.1.3. numerički tipovi podataka

4.1.2 Osnovni pojmovi u projektiranju baze podataka

Prije kreiranja bilo kakve baze podataka, potrebno je dizajnirati njen izgleda i napraviti shemu baze. Nju kasnije prevodimo u određeni broj baza i tablica gdje spremamo podatke. Osnovni element u koji pohranjujemo podatke je **entitet**. Svaki entitet je opisan **atributima**, npr. entitet *osoba* je opisan atributima: *ime, visina, težina, OIB*... Između entiteta mogu postojati **veze**. Veza se može objasniti matematičkim pojmom pridruživanja elementa jednog skupa elementima drugog skupa.



Tablica 4.1.4 Primjer jednostavne baze

4.1.3 Osnovne naredbe za upravljanje podacima u SQL-u

CREATE DATABASE – služi za kreiranje nove baze podataka.

Oblik naredbe:

```
CREATE DATABASE  ime_baze;
```

CREATE TABLE – služi za kreiranje tablice u bazi podataka.

Oblik naredbe:

```
CREATE TABLE ime_tablice (
ime_kolone1   tip_podataka,
ime_kolone2   tip_podataka,
.....
);
```

Primjer:

```
CREATE TABLE korisnici (
Ime      varchar(15),
Prezime varchar(15),
Godište int
);
```

DROP DATABASE – briše podatke i strukturu baze podataka.

Oblik naredbe:

```
DROP DATABASE ime_baze;
```

DROP TABLE – briše podatke i strukuturu tablice.

Oblik naredbe:

```
DROP TABLE ime:tablice;
```

INSERT– unosi zapis u relaciju

Oblik naredbe:

```
INSERT INTO student (prezime, ime, oib, fakultet)
    VALUES ('perić','pero','123456','etf');
```

Moguć je i kraći zapis ako lista vrijednosti koji slijedi ide točno po slijedu definicija relacije.

```
INSERT INTO student VALUES ('perić' , 'pero' , '123456' , 'etf' );
```

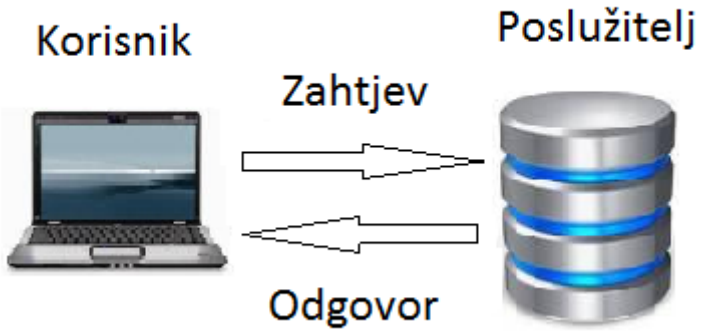
SELECT– postavlja SQL upit nad bazom podataka

Oblik naredbe:

```
SELECT * FROM [ime tablice];
```

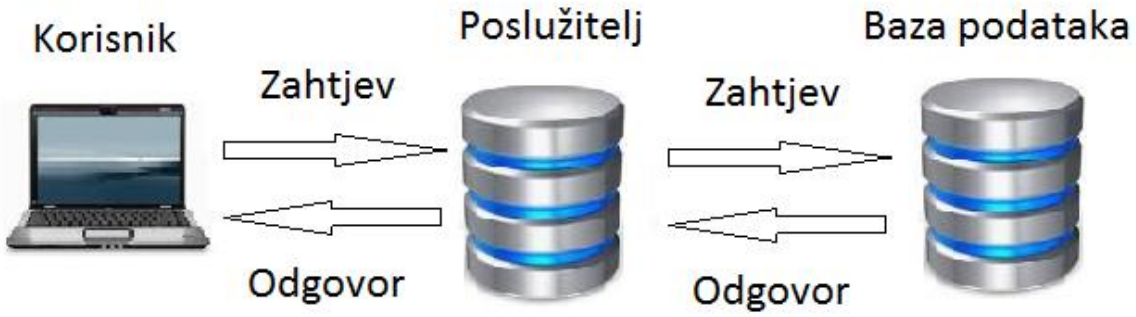
4.2 PHP

PHP (skraćenica za PHP: Hyper text Preprocessor) je open source skriptni jezik namijenjen web developerima pomoću kojeg je moguće kreirati dinamičke i interaktivne HTML stranice na serveru. PHP se ističe širokom podrškom raznih baza podataka i internet protokola kao raspoloživosti brojnih programskih knjižnica. Ono je serverska tehnologija što znači da se gotovi programi izvršavaju na web serveru kojima korisnici pristupaju u vidu web stranice koja je vidljiva na njihovim preglednicima. PHP također omogućava interakciju sa web stranicama, tj. omogućava dvosmjernu komunikaciju. Kada korisnik želi pristupiti stranici on šalje zahtjev poslužitelju koji mu odgovara slanjem HTML datoteke. Tada preglednik dobivenu datoteku grafički oblikuje i prikazuje korisniku. Te datoteke su smještene na poslužitelju što znači kada je korisnik upisivao adresu stranice, zapravo je dobio adresu na fizičkom disku poslužitelja. Takav način spremanja, gdje su sve datoteke na jednom mjestu se naziva dvoslojna arhitektura (eng. Two-tier architecture, *slika 4.2.1.*).



Slika 4.2.1. Dvoslojna arhitektura

Danas je češća troslojna arhitektura*(slika 4.2.2.)*. Ona se razlikuje od dvoslojne po tome što su u njoj podaci i datoteke smješteni na odvojenim bazama podataka koje su u pravilu na različitim poslužiteljima. Što je veći broj upita, veći je i broj poslužitelja koji se zajedno koriste kao jedna baza podataka. PHP programski jezik je dio troslojne arhitekture jer u većini slučajeva dio podataka pohranjuje u bazi podataka.



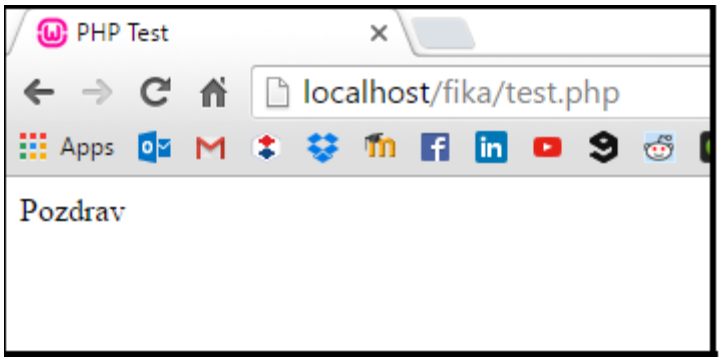
Slika 4.2.2.Troslojna arhitektura

Programe pisane u programskom jeziku PHP nije potrebno prevoditi u izvršni oblik (eng. Compile), oni se izvode prilikom pokretanja u interpreteru. Programi se pišu kao dio HTML stranice. Sam program se nalazi unutar HTML oznake koji počinje sa'<html>' izavršava sa '</html>'. Sve unutar ove oznake se smatra PHP programom i ako je na poslužitelju instaliran i ispravno podešen PHP interpreter, a datoteka završava s .php tada će se taj dio programa automatski izvršiti [2].

Primjer jednostavnog programa pisanog u PHP-u je sljedeći:

```
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<?phpecho '<p>Pozdrav</p>'; ?>
</body>
</html>
```

Slika 4.2.3. jednostavan PHP kod



Slika 4.2.4. Ispis koda u pregledniku

HTML kôd koji je preglednik dobio tada se može vidjeti tako da je dio u kojem je bio napisan PHP kôd zamijenjen rezultatom njegovog izvođenja:



Slika 4.2.5. Rezulat izvedbe koda u pregledniku

PHP možemo koristiti na većini operacijskih sustava, od Unix-a, Windowsa i Mac-a.

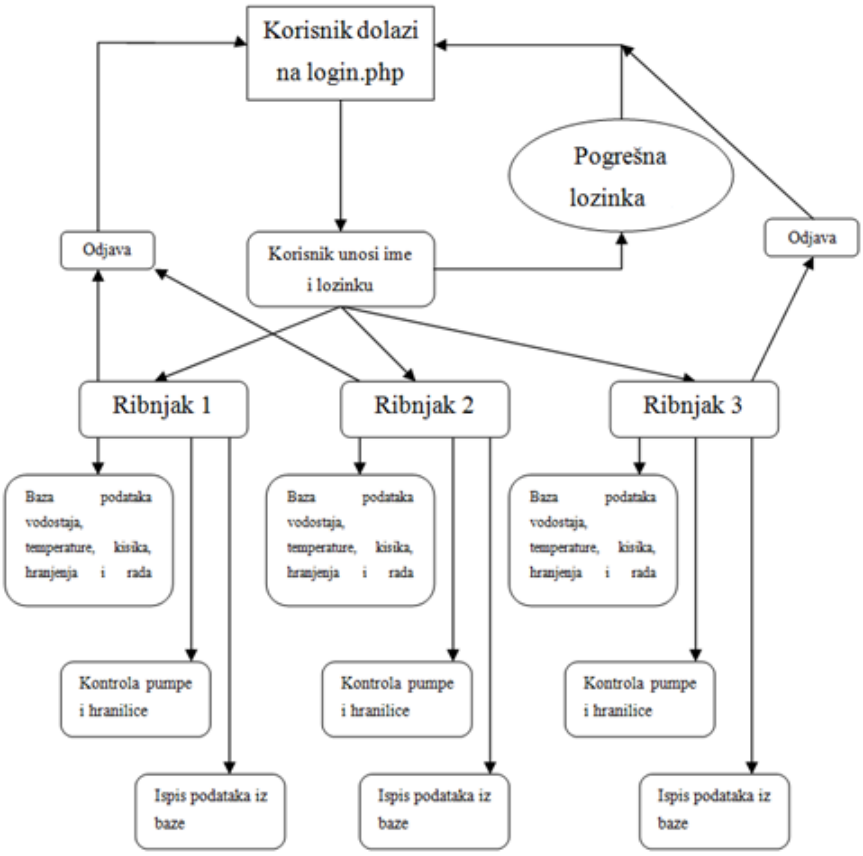
5. PROGRAMSKO RJEŠENJE

U ovom poglavlju je predstavljeno programsko rješenje sustava daljinskog nadzora i upravljanja za tri ribnjaka. Objašnjena je struktura programa, prikazane su baze podataka te upravljanje istima, a na kraju testiranje i rad samog programa.

Program je instaliran i pokrenut na osobnom računalu pomoću WAMP servera. Taj programski paket omogućuje simuliranje rada web servera na osobnom računalu. U svom sastavu sadrži HTTP server (Apache), server za bazu podataka (mySQL) i interpretator (PHP). Programski paket također sadrži i phpMyAdmin, alat za upravljanje bazama podataka.

5.1 PROGRAMSKA IZVEDBA U PHP-u

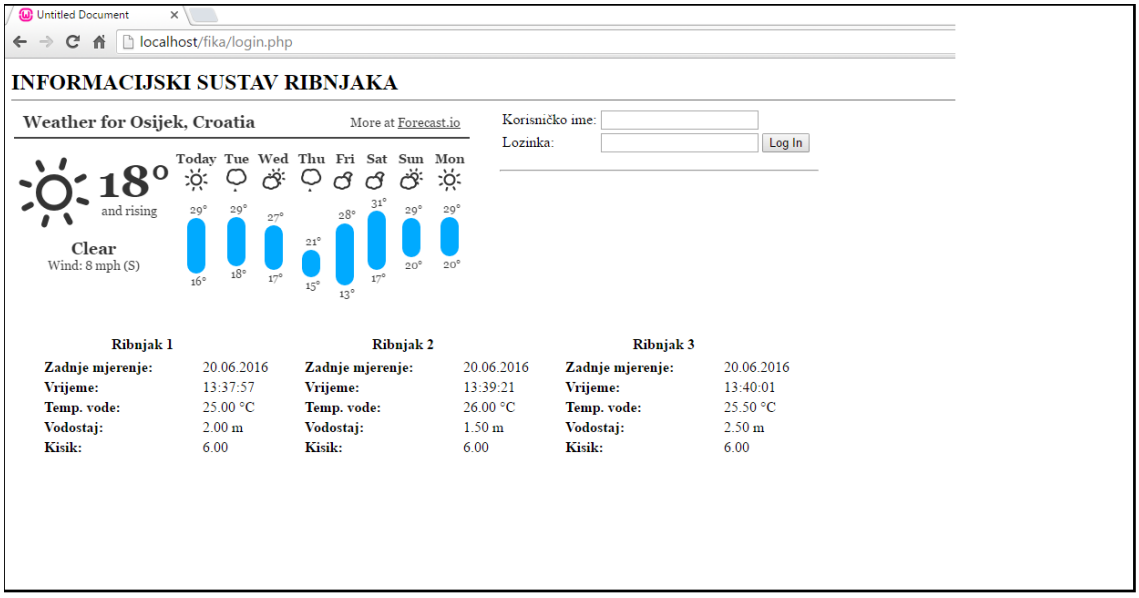
Struktura programa za daljinski nadzor i upravljanje ribnjakom prikazana je sljedećim dijagramom:



Slika 5.1 Struktura programa za upravljanje ribnjakom

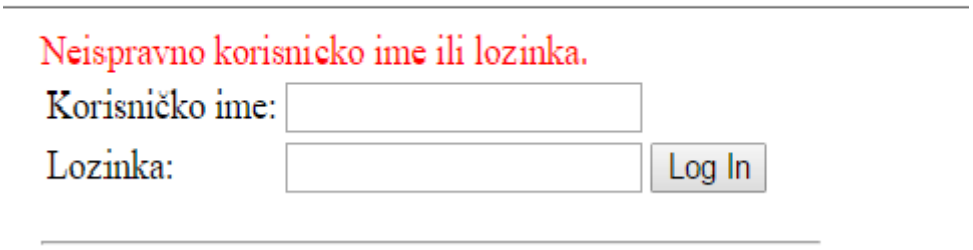
5.1.1.Login.php

Program je zamišljen kao PHP stranica kojom se pristupa putem internetskog preglednika. Početna stranica (*slika 5.1.1.1*) pruža korisniku uvid u podatke prikupljene na zadnjem mjerenju za sva 3 ribnjaka. To su temperatura vode, vodostaj, razina kisika, vremena hranjenja i pumpanja vode. Nakon prijave korisnik će imati mogućnost biranja između ručnog i automatskog pokretanja hranilice i pumpe na pojedinim ribnjacima. U gornjem lijevom kutu prikazani su trenutni vremenski uvjeti s kratkom vremenskom prognozom za sljedećih sedam dana.



Slika 5.1.1.2 Izgled početne stranice

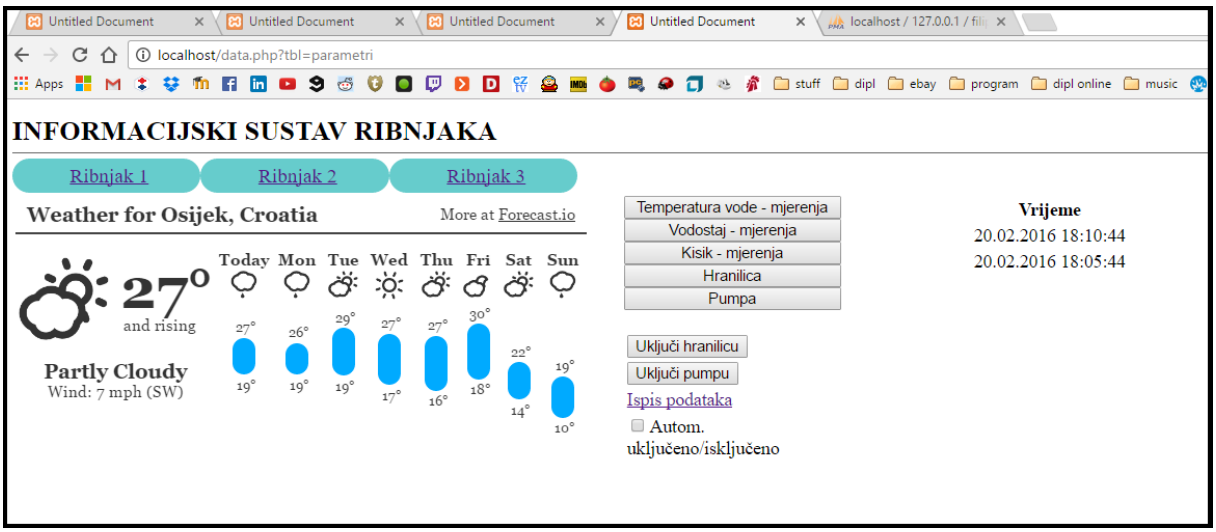
Za pristup mogućnostima upravljanja ribnjakom korisnik se mora logirati svojim korisničkim imenom i lozinkom. Unošenjem pogrešnog korisničkog imena ili lozinke sustav ispisuje poruku: „Neispravno korisničko ime ili lozinka“ *(slika 5.1.1.2).*



Slika 5.1.1.2 Prikaz nakon pogrešnog unosa imena ili lozinke

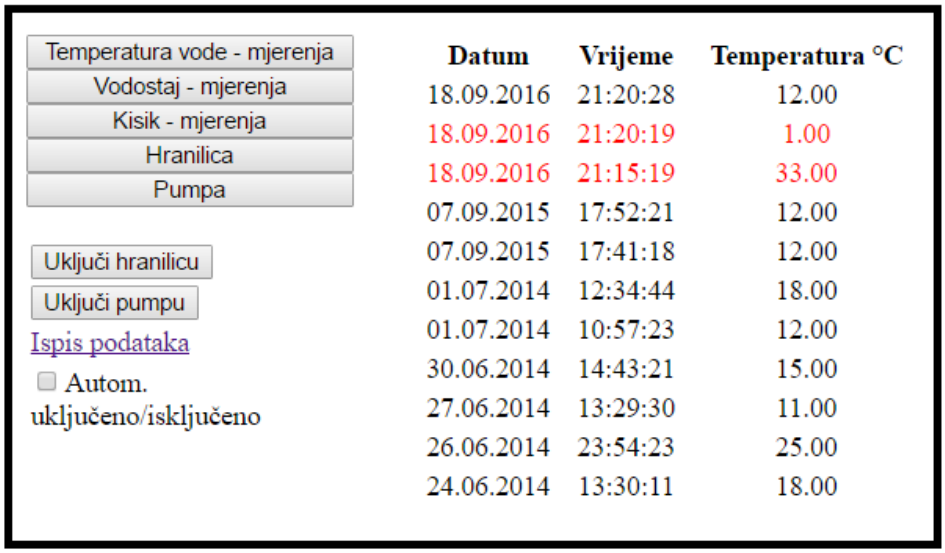
5.1.2 Data.php

Nakon uspješnog unošenja korisničkog imena i lozinke dolazimo na stranicu *data.php*. Izbor željenog ribnjaka se vrši pritiskom na pojedine kartice (Ribnjak 1, Ribnjak 2, Ribnjak 3)gdje dobivamo uvid u sve prikupljene podatke i mogućnosti upravljanja ribnjacima (*Slika 5.1.2.1).*



Slika 5.1.2.1 Izgled stranice data.php

Pritiskom na karticu „*Temperatura vode – mjerenja*“ *(Slika 5.1.2.2), „Vodostaj – mjerenja“ i „Kisik - mjerenja“* možemo vidjeti vrijeme, datum i iznos svakog pojedinog mjerenja.



Datum	Vrijeme	Temperatura °C
18.09.2016	21:20:28	12.00
18.09.2016	21:20:19	1.00
18.09.2016	21:15:19	33.00
07.09.2015	17:52:21	12.00
07.09.2015	17:41:18	12.00
01.07.2014	12:34:44	18.00
01.07.2014	10:57:23	12.00
30.06.2014	14:43:21	15.00
27.06.2014	13:29:30	11.00
26.06.2014	23:54:23	25.00
24.06.2014	13:30:11	18.00

Slika 5.1.2.2 Primjer ispisa svih temperatura mjerenja na pojedinom ribnjaku

Vrijednosti pojedinih mjerenja su poredane po datumu. Crveno ispisane vrijednosti ukazuju na mjerenje čiji iznos nije unutar optimalnih parametara. U ovom slučaju je to iznos koji je manji od 10 i veći od 30. Time dobivamo jednostavan sustav upozoravanja korisnika o potencijalnom problemu.

```
<?phpforeach($dataAll as $row): ?>

<trstyle="color:<?phpecho $row['temperatura'] > 30 || $row['temperatura'] < 10  ? 'red' :
"; ?>">

<tdalign="center"><?phpecho date("d.m.Y", strtotime($row['vrijeme']));?></td>

<tdalign="center"><?phpecho date("H:i:s", strtotime($row['vrijeme']));?></td>

<tdalign="center"><?phpecho $row['temperatura'];?></td>

</tr>

<?phpendforeach; ?>
```

Slika 5.1.2.3 Dio koda za ispis temperature

Ukoliko jedan od parametara ne zadovoljava kriterije, dodan je sustav upozoravanja korisnika i putem elektroničke pošte. Na taj način korisnik konstantno može imati uvid u biološke procese ribnjaka i brzo može otkloniti potencijalno neželjene posljedice. Slanje elektroničke pošte se izvršava putem SMTP protokola (Simple Mail Transfer Protocol) preko *Google* računa. Da bi proces slanja bio uspješan potrebno je pravilno konfigurirati protokol. Adresa poslužitelja je *'ssl://smtp.gmail.com'*, a port (pristupnu točka) mora biti postavljen na 465. Time je omogućena sigurna komunikacija među računalima. Adresa primatelja može biti proizvoljna (u ovom slučaju je *fcrnjac13@hotmail.com*) kao i naslov te sadržaj poruke *(slika 5.1.2.4)*.

```
if($temperatura>30 or $vodostaj<1 or $kisik<5){

		$message = "Alarm";
		$message = wordwrap($message, 70, "\r\n");
		$from = '<crnjac.filip@gmail.com>';
		$to = '<fcrnjac13@hotmail.com>';
		$subject = 'Upozorenje-ribnjak1!';
		$body = "Jedan od parametara na ribnjaku 1 nije unutar optimalnih granica! \n\ ?";

		$headers = array(
			'From' => $from,
			'To' => $to,
			'Subject' => $subject
		);

		$smtp = Mail::factory('smtp', array(
				'host' => 'ssl://smtp.gmail.com',
				'port' => '465',
				'auth' =>true,
				'username' => 'crnjac.filip@gmail.com',
				'password' => 'hfwctdyfcvwmfgqw'
			));

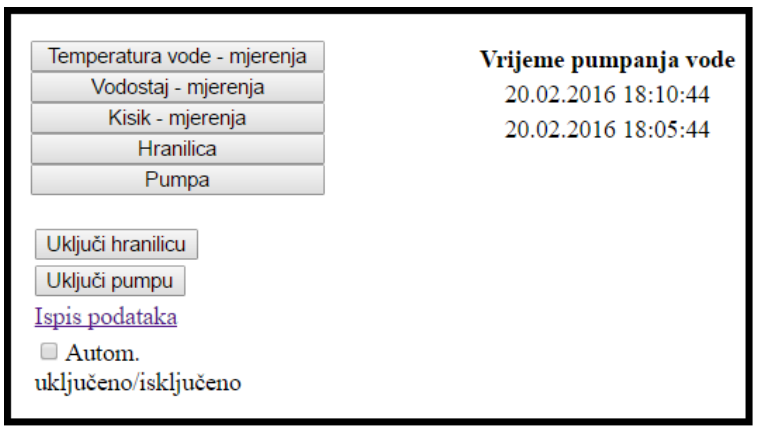
		$mail = $smtp->send($to, $headers, $body);

		if (PEAR::isError($mail)) {
			echo('<p>' . $mail->getMessage() . '</p>');
		} else {
			echo('<p>Poruka uspješno poslana!</p>');
		}
	}

	$msg = "Podaci su spremljeni u bazu";
```

Slika 5.1.2.4 Dio koda za slanje elektroničke pošte

Kartica „Hranilica“ i „Pumpa“ nam pokazuju datum i vrijeme izvršenog ciklusa hranjenja i pumpanja vode *(slika 5.1.2.5).*

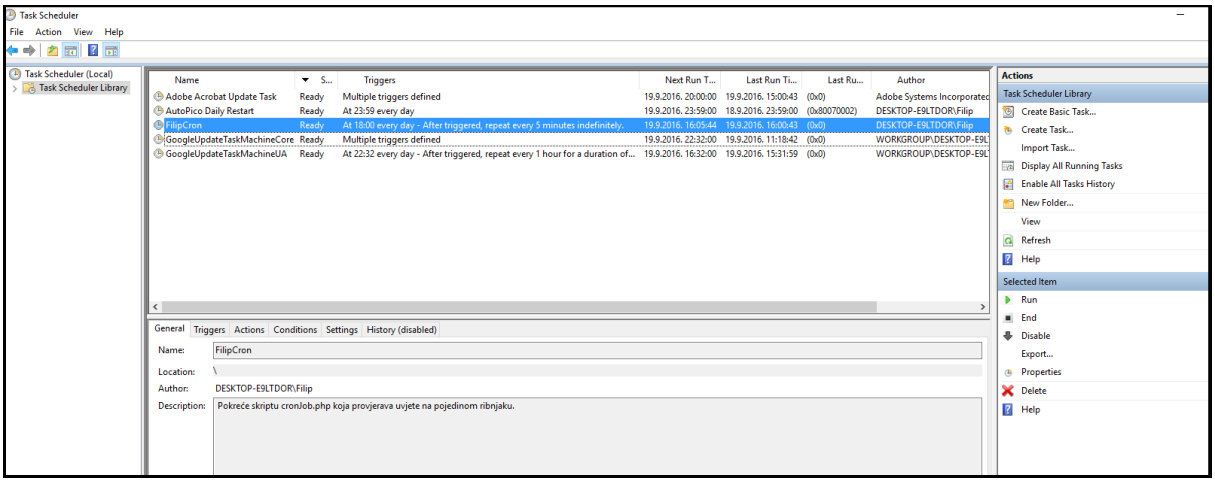


Slika 5.1.2.5 Primjer ispisa vremena pumpanja vode u ribnjak

Ciklusom hranjenja i pumpanja vode možemo upravljati automatski i ručno. Ručno upravljanje se vrši pritiskom na kraticu *„Uključi pumpu“* ili „*Uključi hranilicu“* čime se izvršava jedan ciklus hranjenja (ili pumpanja vode) u trajanju od 5 sekundi. Automatsko hranjenje i pumpanje vode se uključuje obilježavanjem kvačice na prozoru *„Autom. Uključeno/isključeno“*. Taj sustav radi tako da kontinuirano provjerava dali kvačica na prozoru *„Autom. uključeno/isključeno“* obilježena za svaki pojedini ribnjak. Na ribnjacima gdje je kvačica obilježena u odvojenim vremenskim intervalima se vrši proces hranjenja i pumpanja vode. Program koji pokreće taj ciklus se zove *„Task Scheduler“* (ili upravitelj zadataka) koji je dio *Windows 10 Pro* operacijskog sustava. Upravitelju zadataka je potrebna skripta (u ovom slučaju je nazvana *cronJobs.php*) koja provjerava uvjet jeli kvačica označena.

```
if ($ribnjak1 == "Y") {doJobs("parametri");}
if ($ribnjak2 == "Y") {doJobs("parametri2");}
if ($ribnjak3 == "Y") {doJobs("parametri3");}
        functiondoJobs($param){
                $urlHranilica = "http://localhost/fika/hranilica.php";
                $ch = curl_init();
                curl_setopt($ch, CURLOPT_URL, $urlHranilica);
                curl_setopt($ch, CURLOPT_POST, true);
                curl_setopt($ch, CURLOPT_POSTFIELDS, " param=".$param);
                $result = curl_exec($ch);
                curl_close($ch);
```

Slika 5.1.2.6 Dio koda koji provjerava uvjete u skripti cronJobs.php

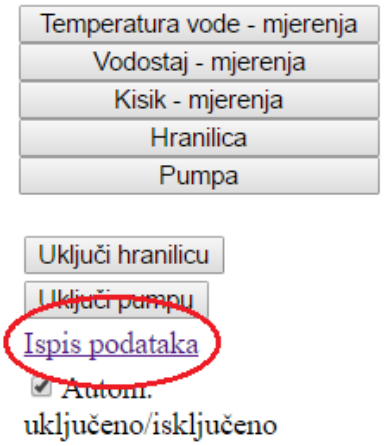


Slika 5.1.2.7 Izgled prozora TaskScheduler sa skriptom cronJob.php

TaskScheduler(slika 5.1.2.7) ima više mogućnosti izvršavanja željene skripte. U ovom slučaju zadatak (nazvan FilipCron) izvršava skriptu svaki dan u vremenskim intervalima od 5 minuta sve dok poslužiteljsko računalo radi.

5.1.3 Ispis podataka

Program posjeduje mogućnost ispisa svih podataka iz baze u PDF formatu *(eng. Portable Document Format)*. To je univerzalni format za zapis dvodimenzionalnih dokumenata koji se mogu prenositi s računala na računalo pri čemu se zadržava originalni izgled stranice sa svim fontovima, slikama i tablicama. Ispis se vrši pritiskom na tipku *„Ispis podataka"(slika 5.1.3.1)*



Slika 5.1.3.1 Tipka za ispis podataka u pdf formatu

Pritiskom na tipku *„Ispis podataka“* dobiva se ispis u tabličnom obliku *(slika 5.1.3.2).*

Ribnjak 1

Vrijeme	Temperatura	Vodostaj	Kisik
2014-07-01 10:57:23	12.00	99.99	12.00
2014-07-01 12:34:44	18.00	4.00	15.00
2015-09-07 17:41:18	12.00	2.00	23.00
2015-09-07 17:52:13	12.00	5.00	22.00
2015-09-11 21:24:19	1.00	0.00	0.00
2016-02-19 22:13:28	22.00	22.00	22.00

Ribnjak 1

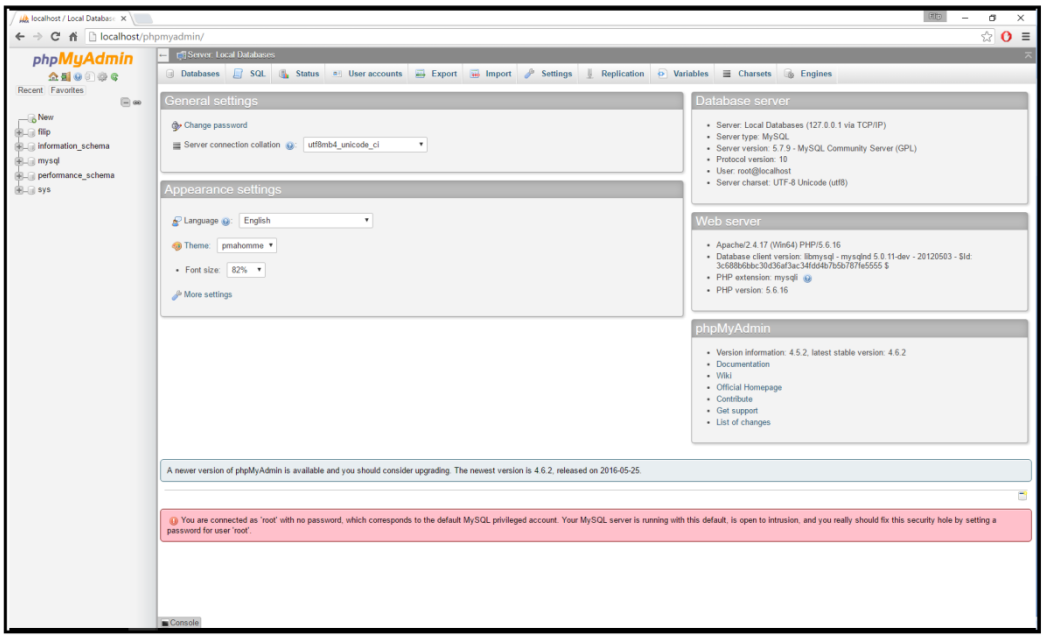
Vrijeme Hranjenja
2016-02-20 18:10:44
2016-02-20 18:05:44
2016-09-19 15:26:05

Vrijeme Pumpanja
2016-02-20 18:10:44
2016-02-20 18:05:44
2016-09-19 15:48:34

Slika 5.1.3.2 Ispis podataka u PDF formatu za ribnjak 1

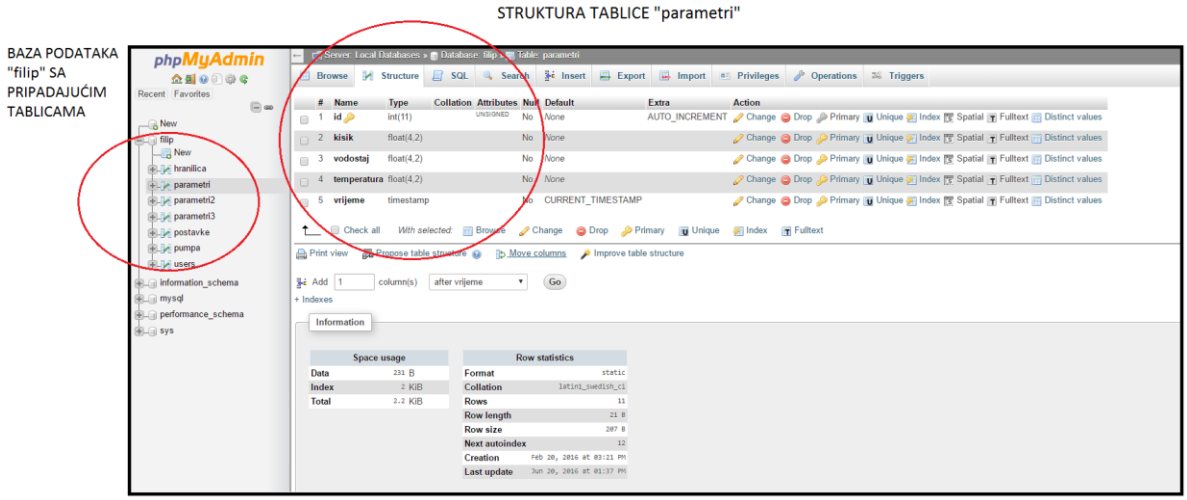
5.2 STRUKTURA BAZE PODATAKA

Za kreiranje i upravljanje bazom podataka koristili smo *phpMyAdmin*. To je programski paket koji daje mogućnost upravljanja bazama podataka putem grafičkog sučelja. Pristupa mu se putem web preglednika upisujući adresu: *http://localhost/phpmyadmin/*(*slika 5.2.1*).



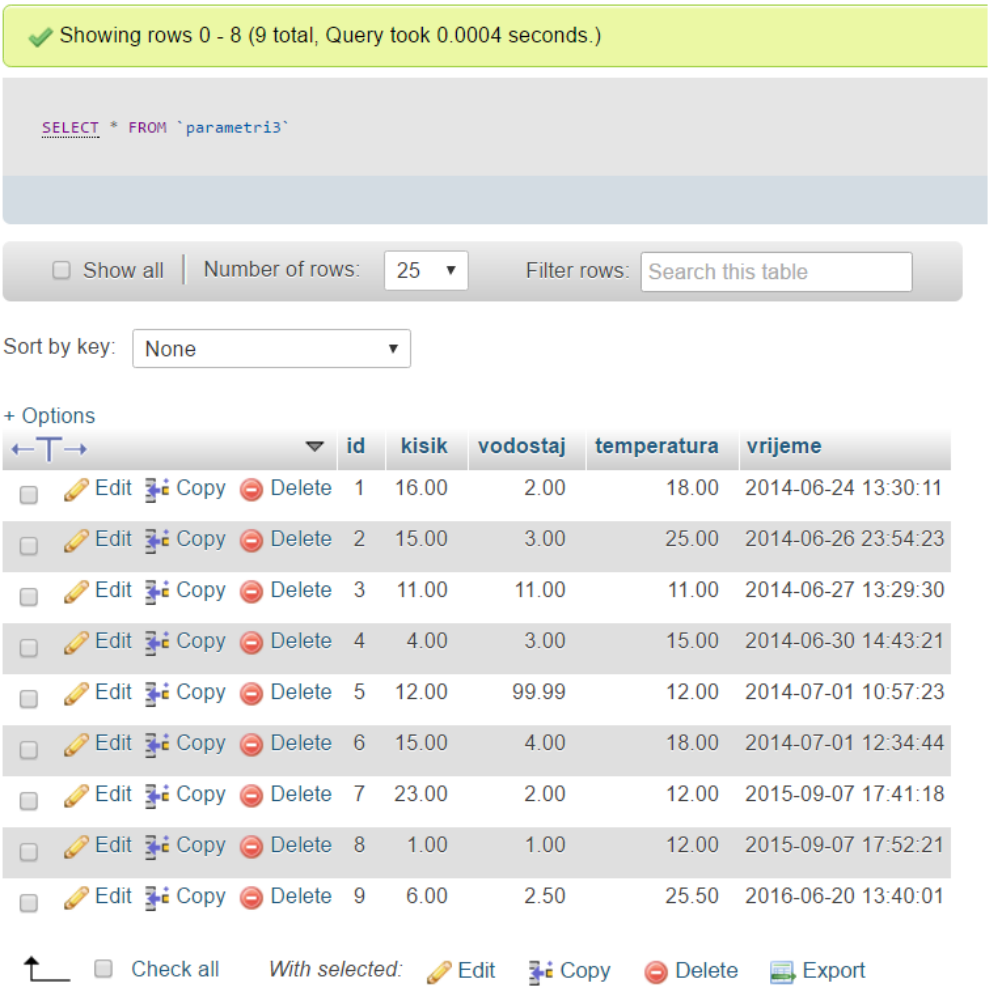
Slika 5.2.1 Osnovno grafičko sučelje phpMyAdmin programskog alata

Bazu podataka nazvana je *„filip“*. Sadrži sedam tablica: *users, parametri, parametri2, parametri3, hranilica, pumpa i postavke*. Tablica *parametri* ima pet atributa: *id, kisik, vodostaj, temperatura i vrijeme.*



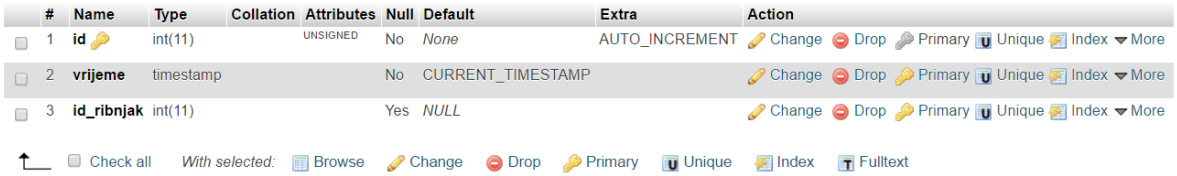
Slika 5.2.2 Struktura baze podataka i pripadajućih tablica

Tablica *parametri* (kao i *parametri2* i *parametri3*) se sastoji od pet atributa: *id, kisik, vodostaj, temperatura i vrijeme*. Atributi *kisik, vodostaj i temperatura* su postavljeni kao float tip podataka, tj. kao numerički tip podataka. Float (x,y) koristimo za pohranu realnih brojeva sa manjom preciznošću. X označava ukupan broj znakova a y broj decimalnih znakova. Parametar *id* je integer tip podataka, tj. cjelobrojni tip podataka. Postavljena mu je karakteristika AUTO_INCREMENT koju koristimo za izradu jedinstvenih indentifikatora u tablici. Svi brojevi se generiraju automatski i uvijek su veći za prvi veći cijeli broj. U tablici može postojati samo jedan jedinstveni identifikator i ne smije biti nula. Ako upisano polje ostane prazno automatski mu se dodjeljuje vrijednost. Atribut *vrijeme* koristi TIMESTAMP tip podatka koji zapisuje vrijeme i datum zadnje promjene podataka, tj. zapisa podataka u tablicu. Naredbom SELECT * FROM 'parametri3' u MySQL-u dobivamo sljedeći ispis (*slika 5.2.3*):



Slika 5.2.3 Primjer zapisa podataka iz tablice parametri3

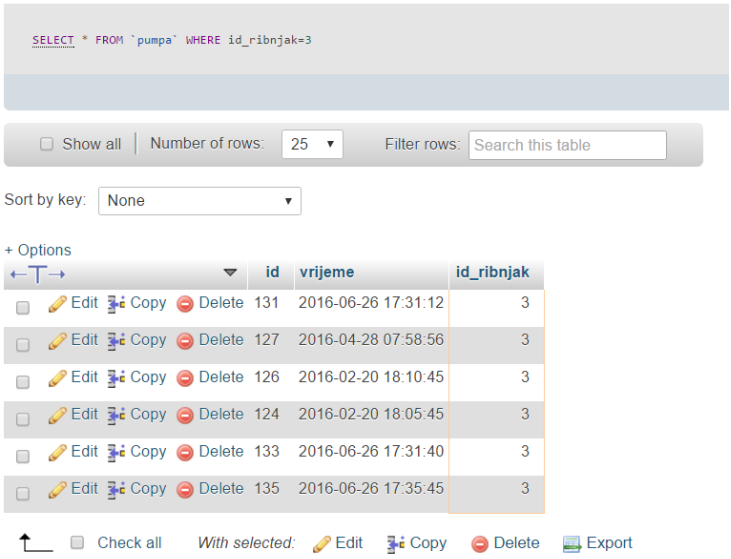
Tablice *hranilica i pumpa* su jednake po strukturi. Sadrže tri atributa: *id, vrijeme i id_ribnjaka*(*slika 5.2.4*). Atributi *id* i *vrijeme* su jednaki kao u prethodno opisanoj tablici (parametri), dok *id_ribnjaka* označava kojem je ribnjaku pridružen zapis. Tako se se*id_ribnjak = 1* odnosi na na parametre ribnjaka 1, *id_ribnjak = 2* na parametre ribnjaka 2, a*id_ribnjak = 3*na parametre ribnjaka 3.



Slika 5.2.4 Struktura tablice hranilica (pumpa)

Ispisom iz takve baze dobiva se vrijeme jednog ciklusa hranjenja (ili pumpanja vode) na pojedinom ribnjaku.

Primjerice, naredbom SELECT * FROM 'pumpa' WHERE id_ribnjak=3 dobivamo ispis svih vremena pumpanja vode na ribnjaku 3 (*slika 5.2.5*)



Slika 5.2.5 Ispis iz tablice pumpa za ribnjak 3

Tablica *users* sadrži popis korisnika koji imaju pristup informacijskom sustavu. Svaki korisnik ima svoje korisničko ime i zaporku.

5.2.1 Spajanje na bazu podataka

Spajanje na MySQL server je prvi korak koji se mora obaviti u svakoj skripti koja ovisi o podacima iz baze podataka. Glavni preduvjet za spajanje je taj da je baza kreirana.

```
<?php

        $host = "localhost";
        $user = "root";
        $pass = "";
        $db = "filip";

        // Connecting to theMySQLdatabase
        $conn = mysqli_connect($host, $user, $pass, $db);
        // Checkconnection
        if (!$conn) {
        die("Connectionfailed: " . mysqli_connect_error());
        }
```

Spajanje na server se obavlja pomoću mysql_connect naredbe. Funkciji prosljeđujemo četiri argumenta: *host, user, pass* i *db*. *Host* je mjesto gdje se nalazi server. U našem slučaju je to *localhost*, tj. server je lokalno računalo. Argument *user* je korisničko ime kojim je spojeno na bazu a *pass* je zaporka tog istog korisnika. *Db* je odbrana baza podataka u kojoj se nalaze željene tablice.

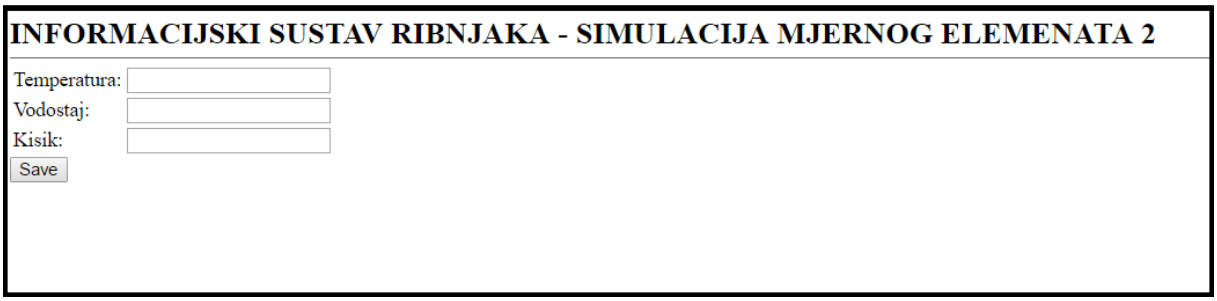
5.3 TESTIRANJE PROGRAMA

Za testiranje programa je kreirana php stranicu *mjerni-element (te mjerni-element2 i mjerni-element3)*, vidljivu na slici 5.3.2, koja simulira mjerne elemente u pojedinim ribnjacima. Upisivanjem vrijednosti u tu stranicu podaci su upisani u našu bazu podataka i prikazani na glavnoj stranici sustava.

```
include 'db.php';
$msg = null;
if (isset($_POST['temperatura'])) {
        $temperatura = $_POST['temperatura'];
        $vodostaj = $_POST['vodostaj'];
        $kisik = $_POST['kisik'];

        $sql = "INSERT INTO parametri (kisik, vodostaj, temperatura) VALUES ('{$kisik}','{$vodostaj}','{$temperatura}')";
        sql($conn, $sql);
```

Slika 5.3.1 Dio koda koja upisuje simulirane podatke u bazu



Slika 5.3.2 Simulacija mjernog elementa ribnjaka 3

Kako bi se testirali upis podataka prvo ćemo obrisati sve postojeće podatke u bazi. Zatim se na stranici '*simulacija mjernog elementa'* u tražena polja upisuju proizvoljne vrijednosti. Za prvi slučaj će se upisati vrijednosti temperature 18, vodostaja 2 i kisika 6 preko mjernog elementa 2 u ribnjak 2. Pritiskom na tipku *Save* dobivena je poruka: „Podaci su spremljeni u bazu“ *(slika 5.3.3)* i vidljivi u sustavu (*slika 5.3.4*).



Slika 5.3.3 Prikaz nakon uspješnog upisa u bazu podataka

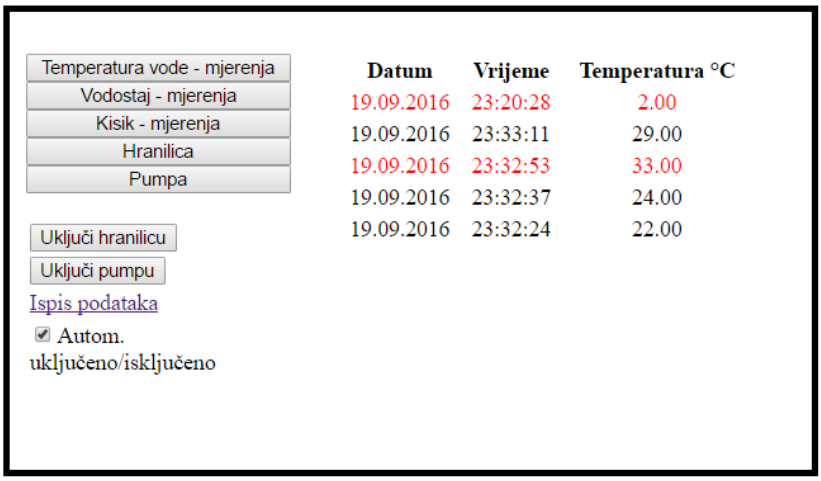


Ribnjak 2

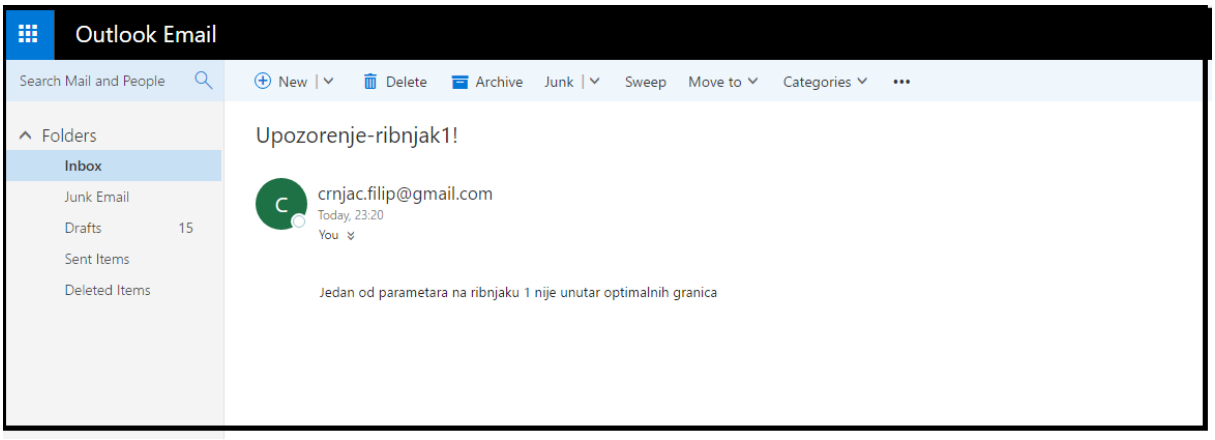
Vrijeme	Temperatura	Vodostaj	Kisik
2016-09-19 23:15:05	18.00	2.00	6.00

Slika 5.3.4 Ispis podataka iz baze u pdf formatu

U drugom primjeru testiramo sustav tako da se upisuje različite vrijednosti temperature: 2, 29, 33, 24 i 22 u sustav. Sve vrijednosti su uspješno upisane ali su vrijednosti 2 i 33 obojane crvenom bojom *(slika 5.3.5)* jer nisu unutar optimalnog područja. Osim toga, putem elektroničke pošte zaprimljeno je upozorenje o nepovoljnim vrijednostima unutar ribnjaka *(slika 5.3.6).*

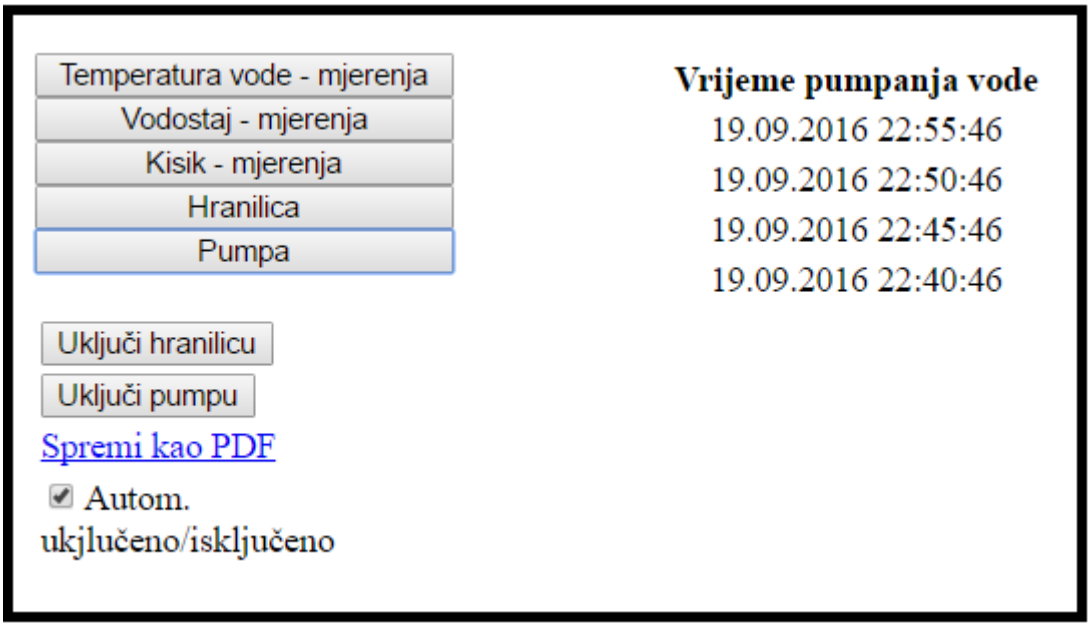


Slika 5.3.5 Ispis unesenih temperatura u ribnjaku 1



Slika 5.3.6 Zaprimljena elektronička pošta sa upozorenjem

Testiranje automatskog ciklusa hranjenja i pumpanja vode vršit će se na ribnjaku 3. Obilježit će se kvačicom prozor *Autom. uključeno/isključeno* i pustiti program da radi 15 minuta. Time dobivamo ispisane pravilne intervale pumpanja vode od 5 minuta (s*lika 5.3.7)*.



Slika 5.3.7 Ispis vremena pumpanja vode u sustavu

6. ZAKLJUČAK

Ribogojstvo je uzgoj ribe koji uključuje sve korake u ciklusu odrastanja ribe. Cilj ove privredne grane je u što boljim uvjetima proizvesti što veću količinu kvalitetne ribe za ljudsku ishranu. Riba se uzgaja u ribnjacima gdje se prate svi parametri bitni za optimalan rast ribe.
Koristeći PHP programski jezik u radu je izrađeno sučelje gdje se ti parametri mogu pratiti. Također se daju mogućnosti upravljanja parametrima te upozoravanje korisnika sustava ako parametri prekorače određene vrijednosti. Korisnik putem elektroničke pošte dobiva upozorenja (alarme) čime se želi pružiti mogućnost kontinuiranog praćenja rada sustava i izbjegavanje svih potencijalno neželjenih negativnih bioloških procesa. Svi podaci su spremljeni u MYSQL bazi podataka i dostupni putem web stranice na koju se treba prijaviti putem korisničkog imena i zaporke. Sustav također nudi ispis svih podataka u pdf formatu i mogućnost ručnog ili automatskog upravljanja pumpama vode i hranilicama. Za automatski način rada koristimo aplikaciju *Task Scheduler* koja jedio Windows operacijskog sustava.
Program je izrađen na hrvatskom jeziku i radi na Windows 10 operacijskom sustavu. Tijekom testiranja nisu primijećene greške pri ispisu i slanju podataka. Također program šalje uspješno upozorenja putem elektroničke pošte čime je zaključeno da programsko rješenje radi stabilno.

LITERATURA

[1] L. Welling, L. Thomson, *PHP andMySQL Web Development (4th Ed.)*, Addison-Wesley, Boston, 2009.

[2] V. Paunović, S. Tomić, *PHP priručnik uz seminar*, Hrvatska udruga za otvorene sustave i Internet, 2006.

[3] Wikipedia, s interneta,https://hr.wikipedia.org/wiki/Baza_podataka

[4] Wikipedia, s interneta, https://en.wikipedia.org/wiki/Fish_farming

[5] Ž. Cvrtila, L. Kozačinski, *Kemijski sastav mesa ribe*, Meso, Vol VII (2006.) studeni-prosinac br. 6

[6] D. Slišković, Procesna *Automatizacija - predavanja* , Osijek, 2009.

SAŽETAK

Tradicionalni ribolov polako prestaje biti najveći opskrbljivač ribom u svijetu. Ribogojstvo iz godine u godinu kontinuirano raste i uskoro će postati glavni izvor ribe u svijetu. Nudi mnoge prednosti u odnosu na tradicionalni ribolov. Na relativno maloj jedinici površine donosi velik i kontinuirani prinos proteina uz minimalan utjecaj na okoliš. Riba je važan dio uravnotežene ljudske prehrane. Bogata je Omega 3 masnim kiselinama i proteinima i kao takva će biti vrlo važna u prehrani stanovništva planete.

Diplomski rad daje rješenje za sustav nadzora nad ribnjakom prateći parametre bitne za ekosustav ribnjaka kao što su temperatura, razina kisika i vodostaj. Također nam daje uvid i kontrolu nad vremenima hranjenja i pumpanja vode te biranjem između automatskog i ručnog načina rada. Time se želi postići maksimalan prinos uz minimalne troškove. Osim toga, program nam daje mogućnost ispisa podataka u pdf formatu za dodatan način arhiviranja podataka, dok alarm obavještava korisnika sustava putem elektroničke pošte o nepravilnostima u radu sustava. Sustav je testiran na osobnom računalu koristeći WAMP server koji simulira rad web servera i utvrđeno je da nema problema pri slanju podataka i stabilnosti sustava.

Ključne riječi: php, MySQL, SCADA, ribnjak, ribogojstvo

ABSTRACT

Traditional fishing will soon be the largest supplier of fish in the world. Aquaculture year after year has been growing constantly and will soon become a major source of fish in the world. It offers many advantages over traditional fishing. On a relatively small area unit brings a large and sustained yield of protein with minimal environmental impact. Fish is an important part of a balanced human diet. It is rich in Omega 3 fatty acids and proteins, and as such it will be very important in the diet of the population of the planet.

This thesis provides a solution for system control over the pond following parameters relevant to the ecosystem of the pond such as temperature, oxygen levels and water levels. It also gives us the insight and control over the times of feeding and pumping water and selecting between automatic and manual modes. We want to achieve the maximum yield at minimum cost. In addition the program gives us the ability to print data as a PDF file to an alternate way to archiving data. An alarm system informs the user via e-mail about the malfunctioning of the system .The system has been tested on the PC using WAMP server that simulates a web server and it was found that no problems sending data and system stability.

Key words: php, MySql, Scada, fish farm, fisihing

ŽIVOTOPIS

Filip Crnjac rođen je 17.6.1987. godine u Vukovaru. Osnovnu školu Dobriše Cesarića završava u Požegi. 2001. godine upisuje se u Srednju tehničku školu Požega – smjer tehničar za računarstvo koju završava 2005. sa odličnim uspjehom. Na Elektrotehničkom fakultetu u Osijeku, 2005.godine upisuje se na sveučilišni preddiplomski studij računarstva koji završava 2010.g. te dobiva naziv sveučilišni prvostupnik računarstva. Iste godine upisuje se na sveučilišni diplomski studij računarstva - smjer procesno računarstvo. Aktivno se služi engleskim i njemačkim jezikom.

PRILOZI NA CD-U

Prilog	Vrsta	Opis
1.	Diplomski rad Filip Crnjac.pdf	Diplomski rad u PDF dokumentu
2.	Diplomski rad Filip Crnjac.docx	Diplomski rad u MS Word dokumentu
3.	Diplomski rad.ppt	Prezentacija diplomskog rada
4.	Informacijski sustav ribnjaka	Programski kod