

TEHNIČKA ŠKOLA DARUVAR

PAMETNI PARKING ZA BICIKE



UČENICI

Antonio Butorac, Ivan Sekereš, Vinko Sigurnjak

MENTORI

Krešimir Ećimović, Miro Kligl, Alen Klubička

Daruvar, svibanj 2021.

Sadržaj:

1.	Uvod.....	- 2 -
2.	Teoretska podloga rada	- 3 -
2.1	Arduino UNO	- 4 -
2.2	Ethernet proširenje (Shield).....	- 5 -
2.3	RFID čitač.....	- 6 -
2.4	Dvokanalni relej	- 7 -
2.5	WS2812B adresabilna RGB LED dioda	- 8 -
2.6	Brava 12V	- 9 -
3.	Izvedba rada.....	- 10 -
3.1	Potrebni materijali	- 10 -
3.2	Shema.....	- 11 -
3.3	Arduino kôd.....	- 12 -
3.4	Python kôd	- 15 -
3.5	Kôd za Web stranicu	- 17 -
3.6	Baza podataka	- 30 -
3.7	Sučelje za prijavu, registraciju i ispis	- 31 -
4.	Zaključak.....	- 34 -
5.	Literatura.....	- 35 -

1. Uvod

Pametni sustav za parkiranje bicikala namijenjen je prvenstveno učenicima. Cilj je potaknuti učenike na dolazak biciklom u školu. Baziran je na RFID tehnologiji koja služi za identifikaciju.

Za komunikaciju između čitača i poslužitelja koristi se Arduino koji je spojen ethernet kabelom na kontrolno računalo. Na računalu se vrti Python skripta koja vrši provjeru ID-a kartice te upisivanja u MySQL bazu koja se nalazi u cloud-u.

Za registraciju korisnika postoji Web stranica pomoću koje se, osim registracije, korisnici mogu prijaviti te vidjeti popis prethodnih korištenja kartice.



*Slika 1. Parking za bicikle
(Izrađeno programom Fusion360)*

2. Teoretska podloga rada

Konstrukcija parkinga sastoji se od više pojedinačnih stupića u kojima su RFID čitači i brave. U stupiću se nalazi Arduino mikroupravljač s Ethernet proširenjem koji je povezan UTP kabelom na mrežni preklopnik. Preklopnik je povezan s poslužiteljem (računalom) na kojem se vrti Python skripta koja vrši provjeru podataka u bazi te na temelju njih šalje Arduinu zahtjev za zaključavanjem ili otključavanjem brave. Skripta također stvara zapis u bazi podataka kada je koji bicikl i na kojoj stanici otključan ili zaključan.

Za otključavanje brave koristi se relej koji je upravljan Arduino mikroupravljačem. Na stupiću je postavljena RGB dioda koja signalizira da li je stanica zauzeta te kada treba prisloniti karticu na čitač.



Slika 2. Izgled stupca za prijavu korisnika

2.1 Arduino UNO

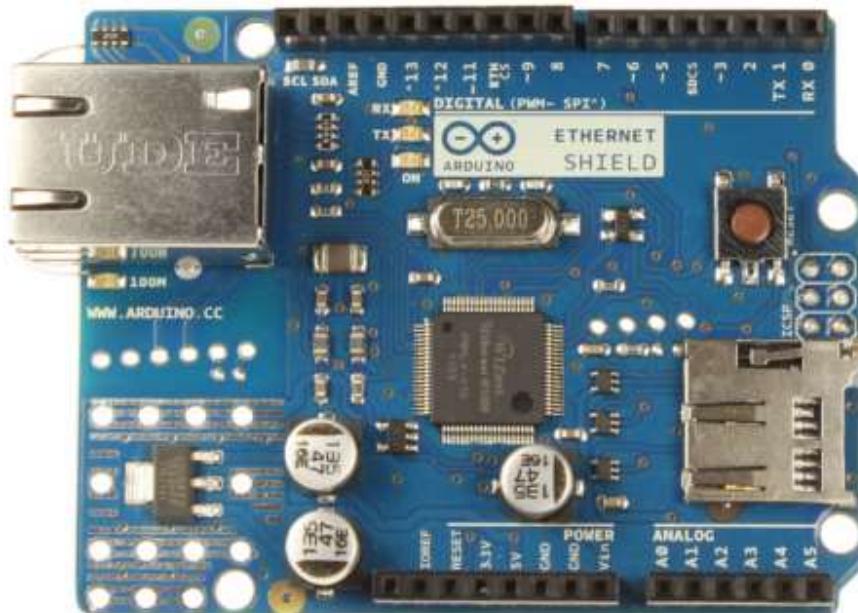
Arduino UNO je mikroupravljač koji se sastoji od 14 digitalnih ulazno/izlaznih pinova (6 njih može biti iskorišteno kao PWM izlaz), 6 analognih ulaza, 16Mhz kvarjni kristal, USB priključak za prenošenje podataka i kao izbor napona, ICSP konektore, konektor za bateriju i gumb za reset. Za programiranje koristi se Arduino IDE koji je baziran na C i C++ programskim jezicima.



Slika 3. Arduino UNO
(<https://bit.ly/2S3uGmj>)

2.2 Ethernet proširenje (Shield)

Arduino Ethernet proširenje V1 dozvoljava Arduino pločici spajanje na internet. Baziran je na Wiznet W5100 ethernet čipu koji podržava TCP i UDP protokole. Podržava do četiri istodobne konekcije. Za korištenje shield-a potreban je Ethernet library. Na sebi ima micro-SD utor za SD kartice koje se mogu koristiti za spremanje datoteka za slanje preko mreže. Shield podržava Power over Ethernet(PoE).

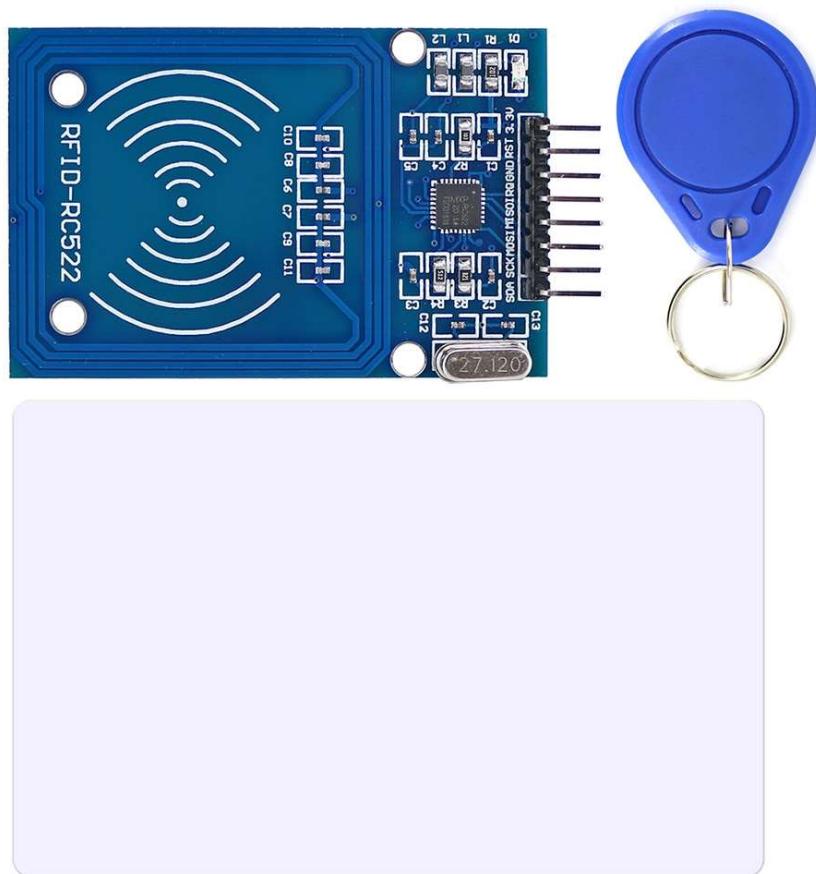


Slika 4. Ethernet Shield V1
(<https://bit.ly/2UtZWwp>)

2.3 RFID čitač

RFID (radio-frequency identification) čitač koristi elektromagnetsko polje za prijenos podataka preko kratkih udaljenosti. Modul radi na frekvenciji od 13,56MHz. Napaja se s 3,3V, ali pinovi podržavaju do 5V napona.

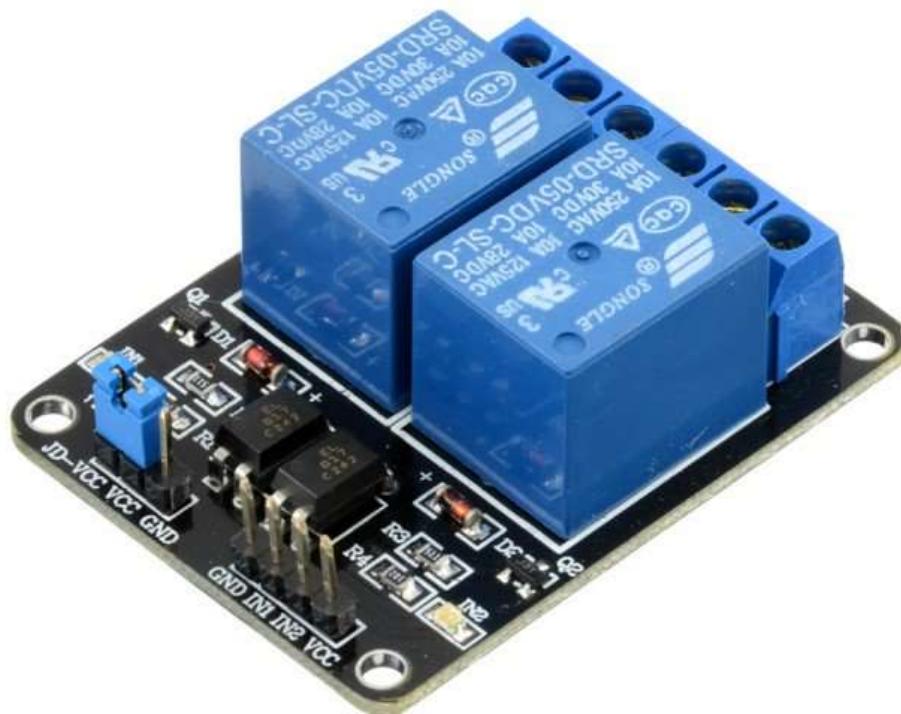
Modul komunicira s mikrokontrolerom pomoću SPI, I2C ili UART protokola.



Slika 5. RFID čitač, RFID kartica i token
(<https://bit.ly/31Eo6FZ>)

2.4 Dvokanalni relej

Dvokanalni relej modul jest relej koji je u mogućnosti biti upravljan pomoću mikroupravljača. Modul se sastoji od logičkih ulaza koji rade na naponu od 3,3V do 5V, dvije signalne LED diode, relejima koji podržavaju 250V izmjenične ili 30V istosmjerne struje snage 10A te izlaza. Veličina tiskane pločice modula iznosi 50x45mm.



Slika 6. Dvokanalni relej
(<https://bit.ly/2Uz16XE>)

2.5 WS2812B adresabilna RGB LED dioda

WS2812B adresabilna RGB LED dioda koristi se za signalizaciju stanja pojedinačne stanice za parking.



Slika 7. Adresabilna RGB LED dioda
[\(https://amzn.to/31yJ5Kh\)](https://amzn.to/31yJ5Kh)

2.6 Brava 12V

Elektronička brava napaja se naponom od 12V. Zbog zahtjevnog solenoida koji pomiče okidni mehanizam, prilikom otpuštanja brava povlači oko 2A struje. Brava ima povratni signal koji šalje vrijednost kada je u bravi nešto zaključano i kada nije.



Slika 8. Elektronička brava 12V
<https://bit.ly/2S3xcZN>

3. Izvedba rada

3.1 Potrebni materijali

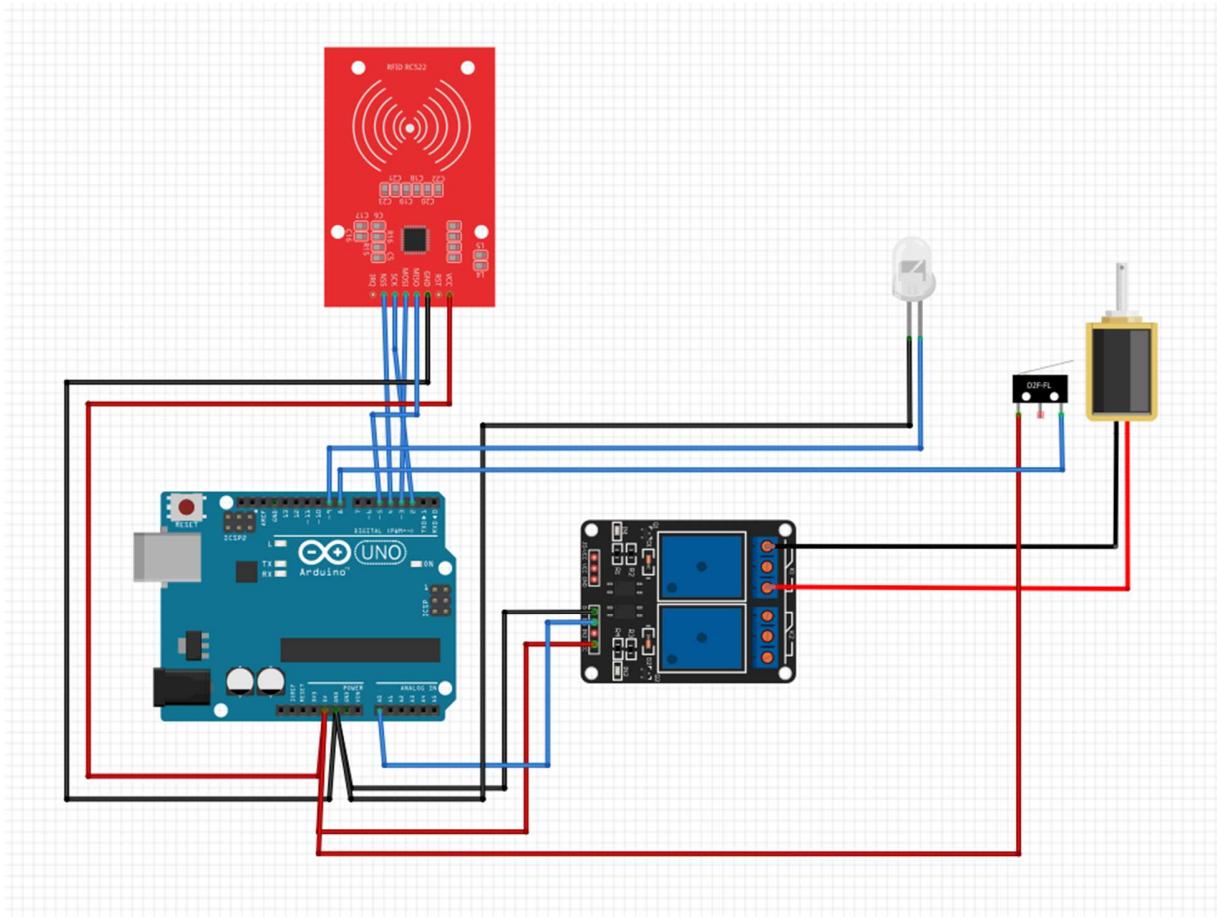
- Arduino UNO
- Ethernet Shield V1
- RFID čitač i kartica
- Dvokanalni relej
- Elektronička brava
- Žice
- WS2812B adresabilne RGB LED diode
- Prusa i3 MK3 3D printer
- PLA i PETG plastični filament
- Lemilica
- Napajanje
- Mrežni preklopnik
- PC
- Ethernet kabel



*Slika 9. Glava stupića za parking
(Izrađeno programom Fusion360)*

Pametni parking za bicikle

3.2 Shema



Slika 10. Shema

(Izrađeno programom Fritzing)

3.3 Arduino kôd

```
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>
#include <Wire.h>
#include <Adafruit_PN532.h>
#include <FastLED.h>

#define PN532_SCK (2)
#define PN532_MOSI (3)
#define PN532_SS (4)
#define PN532_MISO (5)

#define NUM_LEDS 1
#define DATA_PIN 9

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xEE};
IPAddress ip(192, 168, 0, 24);
unsigned int localPort = 23;
char packetBuffer[UDP_TX_PACKET_MAX_SIZE];
String datReq;
int packetSize;
EthernetUDP Udp;

Adafruit_PN532 nfc(PN532_SCK, PN532_MISO, PN532_MOSI, PN532_SS);

int relay = A0;
int brava = 8;
int b = 0;

CRGB leds[NUM_LEDS];

uint8_t success;
uint8_t uid[] = {0, 0, 0, 0};
uint8_t uidLength;

void setup()
{
    FastLED.addLeds<WS2812, DATA_PIN, RGB>(leds, NUM_LEDS);
    pinMode(brava, INPUT_PULLUP);
    pinMode(relay, OUTPUT);
    //pinMode(buzzer, OUTPUT);
    digitalWrite(relay, LOW);
    Serial.begin(115200);
    Ethernet.begin(mac, ip);
    Udp.begin(localPort);
    delay(500);
```

```
nfc.begin();

nfc.SAMConfig();

Serial.println("Waiting for a card ...");

delay(5000);

packetSize = Udp.parsePacket();

if(packetSize > 0)
{
    Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
    String datReq(packetBuffer);
    Serial.println("Ocitavanje paketa: ");
    Serial.println(datReq);
}

void loop()
{
    leds[0] = CRGB::Green;
    FastLED.show();

    while(digitalRead(brava) == 0)
    {
        if(b == 0)
        {
            leds[0] = CRGB::Blue;
            FastLED.show();
        }
    }

    success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uidLength);

    if(success)
    {
        Serial.println("");
        Serial.println("Found an ISO14443A card");
        Serial.print(" UID Length: ");
        Serial.print(uidLength, DEC);
        Serial.println(" bytes");
        Serial.print(" UID Value: ");
        nfc.PrintHex(uid, uidLength);
        Serial.println(uid[0]);
        Serial.println(uid[1]);
        Serial.println(uid[2]);
        Serial.println(uid[3]);
    }
}
```

Pametni parking za bicikle

```
Serial.println("");
Serial.println("Slanje paketa");
Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
Udp.print(uid[0]);
Udp.print(uid[1]);
Udp.print(uid[2]);
Udp.print(uid[3]);
Udp.endPacket();
Serial.println("Zavrsetak slanja paketa");

delay(500);

memset(packetBuffer, 0, UDP_TX_PACKET_MAX_SIZE);
packetSize = Udp.parsePacket();
Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
String datReq(packetBuffer);
Serial.println("Ocitavanje paketa:");
Serial.println(datReq);

if(datReq == "zakljucaj")
{
    b = 1;
    Serial.println("Card accepted!");
    leds[0] = CRGB::Red;
    FastLED.show();
    success = 0;
} else if(datReq == "otkljucaj")
{
    leds[0] = CRGB::Green;
    FastLED.show();
    digitalWrite(relay, HIGH);
    delay(200);
    digitalWrite(relay, LOW);
    b = 0;
    success = 0;
} else
{
    Serial.println("Card denied");
    success = 0;
}
memset(packetBuffer, 0, UDP_TX_PACKET_MAX_SIZE);
}

delay(3000);

}
```

3.4 Python kôd

```
from socket import *
import mysql.connector
import datetime

address1 = ("192.168.0.24", 23)
client_socket = socket(AF_INET, SOCK_DGRAM)
client_socket.settimeout(1)

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="lozinka123",
    database="baza"
)

mycursor = mydb.cursor()
station = mydb.cursor()

d = "rfid"
client_socket.sendto(d.encode(), address1)
print(d)

while(1):
    try:
        mycursor.execute("SELECT * FROM korisnik")
        myresult = mycursor.fetchall()
        rec_data1, address1 = client_socket.recvfrom(2048)
        t1 = str(rec_data1.decode())
    
```

Pametni parking za bicikle

```
print(t1)

for x in myresult:

    if t1 == x[8] and x[7] == 0:

        mail = x[4]

        data = b"zakljucaj"

        client_socket.sendto(data, address1)

        print(data)

        mycursor.execute("UPDATE `korisnik` SET `Stanje` = 1 WHERE Kartica = %s", [x[8]])

        mydb.commit()

        mycursor.execute("UPDATE `korisnik` SET `Stanica` = 1 WHERE uid = %s", [x[6]])

        mydb.commit()

        station.execute("INSERT INTO `zapis` (`Email`, `Vrijeme`, `Kartica`, `Stanje`, `Stanica`)
VALUES (%s, %s, %s, 'zaključano', 1)", [mail, datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S"), t1])

        mydb.commit()

    elif t1 == x[8] and x[7] == 1:

        mail = x[4]

        data = b"otkljucaj"

        client_socket.sendto(data, address1)

        print(data)

        mycursor.execute("UPDATE `korisnik` SET `Stanje` = 0 WHERE Kartica = %s", [x[8]])

        mydb.commit()

        mycursor.execute("UPDATE `korisnik` SET `Stanica` = 0 WHERE uid = %s", [x[6]])

        mydb.commit()

        station.execute("INSERT INTO `zapis` (`Email`, `Vrijeme`, `Kartica`, `Stanje`, `Stanica`)
VALUES (%s, %s, %s, 'otključano', 1)", [mail, datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S"), t1])

        mydb.commit()

    print("")

except:

    pass
```

3.5 Kôd za Web stranicu

connect.php

```
<?php  
$server = 'localhost';  
$username = 'root';  
$password = 'lozinka123';  
$database = 'baza';  
  
$db = mysqli_connect($server, $username, $password, $database);  
  
//$db_selected = mysqli_select_db($db, $database);  
?>
```

user_login.php

```
<!DOCTYPE html>  
<html lang="en" dir="ltr">  
  <head>  
    <meta charset="utf-8">  
    <title>User login</title>  
    <link href="login.css?v=<?php echo time(); ?>" rel="stylesheet" type="text/css"/>  
  </head>  
  <body>  
    <script>  
      function reg()  
      {  
        document.getElementById("reg").style.display = "block";  
        document.getElementById("log").style.display = "none";  
      }  
    </script>
```

Pametni parking za bicikle

```
function log()
{
    document.getElementById("reg").style.display = "none";
    document.getElementById("log").style.display = "block";
}

</script>

<div class="odabir">
    <button type="button" name="registracija" class="gumb" onclick="reg()">Registriraj se</button>
    <button type="button" name="login" class="gumb" onclick="log()">Prijavi se</button>
</div>

<div class="registracija" id="reg">
    <form class="form1" action="" method="post">
        <p>Unesite ime:</p>
        <input class="input" type="text" name="ime" value="" maxlength="20" required><br><br>
        <p>Unesite prezime:</p>
        <input class="input" type="text" name="prezime" value="" maxlength="25" required><br><br>
        <p>Email:</p>
        <input class="input" type="email" name="email" value="" pattern=".+@skole.hr" placeholder="adresa@skole.hr" required><br><br>
        <p>Lozinka:</p>
        <input class="input" type="password" name="lozinka" value="" maxlength="16" required><br><br><br>
    <div class="r">
        <label class="rad">
            <input class="radio" type="radio" name="bike" value="0" required>Električni bicikl<br>
        </label>
        <label class="rad">
            <input class="radio" type="radio" name="bike" value="1">Obični bicikl<br>
        </label>
        <label class="rad">
```

Pametni parking za bicikle

```
<input class="radio" type="radio" name="bike" value="2">Romobil<br>
</label>
</div><br>
<p>Unesite ID kartice:</p>
<input class="input" type="text" name="kartica" value="" maxlength="12" required><br><br>
<input class="input buton" type="submit" name="submit" value="Registriraj se">
</form>
</div>
<div class="prijava" id="log" hidden>
<form class="" action="user_pregled.php" method="post">
<p>Email:</p>
<input class="input" type="email" name="email" value="" pattern=".+@skole.hr" placeholder="adresa@skole.hr" required><br><br>
<p>Lozinka:</p>
<input class="input" type="password" name="lozinka" value="" maxlength="16" required><br><br>
<input class="input buton" type="submit" name="submit" value="Prijavi se">
</form>
</div>
</body>
</html>

<?php
include("connect.php");

if(isset($_POST['submit']))
{
$Ime = $_POST["ime"];
$Prezime = $_POST["prezime"];
$Lozinka = $_POST["lozinka"];
```

```
$Email = $_POST["email"];
$Bicikl = $_POST["bike"];
$Kartica = $_POST["kartica"];

$sql = "INSERT INTO korisnik";
$sql .= "(Ime, Prezime, Lozinka, Email, Bicikl, Kartica_ID)";
$sql .= "VALUES ('$Ime', '$Prezime', '$Lozinka', '$Email', '$Bicikl', '$Kartica')";

mysqli_query($db, $sql);
}
```

user_pregled.php

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="utf-8">
<title>Prikaz prijava</title>
<link href="login.css?v=<?php echo time(); ?>" rel="stylesheet" type="text/css"/>
</head>
<body>
<button type="button" name="button" class="gumb prk" onclick="window.location.href =
'user_login.php';">
    Povratak na prijavu
</button>
<table class="tablica">
<tr>
    <th>Vrijeme</th>
    <th>ID kartice</th>
    <th>Stanje</th>
    <th>Stanica</th>
```

Pametni parking za bicikle

```
</tr>
<?php
include("connect.php");

$Email = $_POST['email'];
$Lozinka = $_POST['lozinka'];

$sql = "SELECT * FROM korisnik WHERE Email = '$Email'";
$result = mysqli_query($db, $sql);
$i = 0;

while($row = mysqli_fetch_assoc($result))
{
    if($row['Lozinka'] == $Lozinka)
    {
        $i++;
        if($i == 1)
        {
            $sql = "SELECT * FROM zapisi WHERE Email = '$Email'";
            $result = mysqli_query($db, $sql);

            while($row = mysqli_fetch_assoc($result))
            {
                echo
'<tr><td>'.$row['Vrijeme'].'.</td><td>'.$row['Kartica'].'.</td><td>'.$row['Stanje'].'.</td><td>'.$row['Stanica'].'.</td></tr>';
            }
        }
    }
}

else
{
    $message = "Netočni podatci";
```

Pametni parking za bicikle

```
echo "<script type='text/javascript'>alert('$message');</script>";  
}  
}  
}  
?  
</table>  
</body>  
</html>
```

login.css

```
.registracija  
{  
margin-left: auto;  
margin-right: auto;  
text-align: center;  
border: 2px solid red;  
width: 35%;  
padding-top: 15px;  
padding-bottom: 30px;  
border-radius: 8px;  
background-color: #a5a5a5;  
font-family: Arial;  
}
```

```
p  
{  
font-size: 16px;  
}
```

```
.input
```

```
{  
    font-size: 14px;  
    border-radius: 8px;  
    padding: 5px;  
    border: 1px solid black;  
}
```

```
.buton  
{  
    padding: 10px;  
    font-size: 16px;  
    background-color: #d9d9d9;  
}
```

```
.rad input  
{  
    height: 20px;  
    width: 20px;  
    background-color: #eee;  
    vertical-align: middle;  
}
```

```
.r  
{  
    width: 50%;  
    margin: auto;  
    padding-left: 25%;  
    text-align: left;  
}
```

Pametni parking za bicikle

```
.odabir  
{  
    margin-left: auto;  
    margin-right: auto;  
    text-align: center;  
    padding-top: 40px;  
    padding-bottom: 40px;  
    border: 0px solid black;  
    width: 40%;  
    font-family: Arial;  
}
```

```
.gumb  
{  
    width: 170px;  
    height: 60px;  
    margin-left: 5px;  
    margin-right: 5px;  
    text-align: center;  
    font-size: 20px;  
    border-radius: 8px;  
    background-color: white;  
    color: black;  
    border: 2px solid red;  
    transition-duration: 0.5s;  
}
```

```
.gumb:hover  
{
```

Pametni parking za bicikle

```
background-color: #555555;  
color: white;  
}
```

```
.prijava  
{  
margin-left: auto;  
margin-right: auto;  
text-align: center;  
border: 2px solid red;  
width: 35%;  
padding-top: 15px;  
padding-bottom: 30px;  
border-radius: 8px;  
background-color: #a5a5a5;  
font-family: Arial;  
}
```

```
table  
{  
border-collapse: collapse;  
margin: auto;  
margin-top: 30px;  
border-radius: 8px;  
overflow: hidden;  
box-shadow: 0 0 0 2px red;  
}
```

```
table, th, td  
{
```

Pametni parking za bicikle

```
border: 2px solid black;  
padding: 10px;  
font-family: Arial;  
text-align: center;  
font-size: 16px;  
}  
  
th
```

```
{  
background-color: #4d4d4d;  
color: white;  
}
```

```
.prk  
{  
text-align: center;  
margin: auto;  
display: block;  
margin-top: 40px;  
}
```

baza.sql

```
-- phpMyAdmin SQL Dump  
-- version 4.8.5  
-- https://www.phpmyadmin.net/  
--  
-- Host: localhost  
-- Generation Time: Feb 05, 2020 at 07:25 PM  
-- Server version: 8.0.13-4  
-- PHP Version: 7.2.24-0ubuntu0.18.04.2
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

-- 
-- Database: `baza`
-- 
-- -----
-- 
-- Table structure for table `korisnik`
-- 

CREATE TABLE `korisnik` (
  `ID` int(11) NOT NULL,
  `Ime` varchar(20) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,
  `Prezime` varchar(25) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,
  `Lozinka` varchar(16) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,
  `Email` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,
  `Bicikl` int(1) NOT NULL,
  `Stanica` int(1) DEFAULT NULL,
  `Kartica_ID` varchar(12) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

-- 
-- Dumping data for table `korisnik`
```

```
--  
  
INSERT INTO `korisnik` (`ID`, `Ime`, `Prezime`, `Lozinka`, `Email`, `Bicikl`, `Stanica`, `Kartica_ID`)  
VALUES  
(4, 'Ivan', 'Seki', 'bicikl1234', 'ivanseki@skole.hr', 1, NULL, '135344416'),  
(7, 'neki', 'tamo', 'bicikl4321', 'ivanseki@skole.hr', 2, NULL, '21354855');  
  
-----  
  
--  
-- Table structure for table `zapis`  
  
--  
  
CREATE TABLE `zapis` (  
    `ID` int(11) NOT NULL,  
    `Email` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,  
    `Vrijeme` varchar(30) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,  
    `Kartica` varchar(12) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,  
    `Stanje` varchar(10) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,  
    `Stanica` int(1) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;  
  
--  
-- Dumping data for table `zapis`  
  
--  
  
INSERT INTO `zapis` (`ID`, `Email`, `Vrijeme`, `Kartica`, `Stanje`, `Stanica`) VALUES  
(1, 'ivanseki@skole.hr', '2019-06-18 12:22:08', '135344416', 'zaključano', 1),  
(2, 'ivanseki@skole.hr', '2019-01-18 10:34:09', '135344416', 'otključano', 1),  
(3, 'ivanseki@skole.hr', '2020-02-03 7:55:10', '135344416', 'zaključano', 1),  
(4, 'ivanseki@skole.hr', '2020-02-04 15:35:40', '21354855', 'zaključano', 2);  
  
--  
-- Indexes for dumped tables  
  
--  
  
--  
  
-- Indexes for table `korisnik`  


---


```

```
--  
ALTER TABLE `korisnik`  
ADD PRIMARY KEY (`ID`);  
  
--  
-- Indexes for table `zapis`  
  
--  
ALTER TABLE `zapis`  
ADD PRIMARY KEY (`ID`);  
  
--  
-- AUTO_INCREMENT for dumped tables  
  
--  
--  
-- AUTO_INCREMENT for table `korisnik`  
  
--  
ALTER TABLE `korisnik`  
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;  
  
--  
-- AUTO_INCREMENT for table `zapis`  
  
--  
ALTER TABLE `zapis`  
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;  
COMMIT;  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

3.6 Baza podataka

The screenshot shows two tables in a database:

5Az6B1YddY korisnik	
ID : int(11)	(Primary Key)
Ime : varchar(20)	
Prezime : varchar(25)	
Lozinka : varchar(16)	
Email : varchar(30)	
Bicikl : int(1)	
Stanica : int(1)	
Stanje : int(1)	
Kartica_ID : varchar(12)	

5Az6B1YddY zapisi	
ID : int(11)	(Primary Key)
Email : varchar(30)	
Vrijeme : varchar(30)	
Kartica : varchar(12)	
Stanje : varchar(10)	
Stanica : int(1)	

Slika 11. Relacijski dijagram baze podataka

(izrađeno u PHPMyAdmin-u)

3.7 Sučelje za prijavu, registraciju i ispis

The image shows a user interface for a login and registration system. At the top, there are two buttons: "Registriraj se" (Register) on the left and "Prijavi se" (Login) on the right. Below these buttons is a large rectangular form with a red border. Inside the form, there is a label "Email:" followed by a text input field containing the placeholder "adresa@skole.hr". Below this, there is a label "Lozinka:" followed by a password input field. At the bottom of the form is a "Prijavi se" button.

Slika 12. Sučelje za prijavu

Registriraj se

Prijavi se

Unesite ime:

Unesite prezime:

Email:

Lozinka:

Električni bicikl
 Obični bicikl
 Romobil

Unesite ID kartice:

Registriraj se

Slika 13. Sučelje za registraciju

Povratak na
prijavu

Vrijeme	ID kartice	Stanje	Stanica
2019-06-18 12:22:08	135344416	zaključano	1
2019-01-18 10:34:09	135344416	otključano	1
2020-02-03 7:55:10	135344416	zaključano	1
2020-02-04 15:35:40	21354855	zaključano	2
2020-02-06 17:21:08	21354855	otključano	2

Slika 14. Prikaz zapisa

4. Zaključak

Pametni sustav za parkiranje za bicikala koristan je sustav ne samo u rekreativne svrhe već i kao i primjer korištenja moderne tehnologije u stvarnom životu. Koristi svakidašnja programska rješenja i tehnologije koje su zastupljene danas u svijetu. Namijenjen je učenicima no ne nužno samo za parkiranje bicikala nego i na dalnjem razvoju i unaprjeđivanju.

5. Literatura

<https://forum.arduino.cc/>

<https://www.instructables.com/>

<https://www.youtube.com/>

<https://www.w3schools.com/>

<https://docs.python.org/3/>

<https://stackoverflow.com/>