SZABADKAI MŰSZAKI SZAKFŐISKOLA

PROJEKTUM
Elektronikus ügyvitel tantárgyból

JELÖLT **MENTOR**

Laci Zoltán,Horváth Valentin **dr Zlatko Čović**

SZABADKA, 2020.

A web oldal müködése

Egy olyan webes applikáció-t hoztunk létre, amely segít felhasználóinak tájékozódni a világ bármely városában.Az oldal főbb funkciójai közé tartozik a városon belüli helyek feltüntetése,valamint a közösségi aktivitás listázása abból a városból ahonnan a felhasználó bejelentkezett.A felhasználó regisztrálás és a fiók aktiválása után betud lépni a rendszerbe, el kell fogadnia a böngésző helymeghatározását ,és így tudja használni a web oldal funkcióit.Belépéskor a rendszer a felhasználó koordinátái alapján betölti a mappát ahol megjelöli a pontos helyzetét, ezek után a felhasználó kategória szerint tud keresni helyszíneket.A kategóriákat 4 fő csoportba definiáltuk "Going out","Shopping","Transport","Museum", szóval lehetőség van rákeresni taxi és busz állomásokra, plázákra, múzeumokra, kávézókra bulizó helyekre.Amikor a felhasználó rákattint a meghatározott kategóriára a mappa betölti az összes találatot és feltünteti ajax technológiával,valamint feltünteti azoknak a felhasználóknak a posztjait akik már jártak azon a helyen.Az oldalra lehetőségünk van képet feltölteni, például ha jártunk egy múzeumban készitünk egy képet feltöltsük az oldalra, a rendszer pedig megőrzi nekünk emlékben.A web oldal struktúrájához bootstrap 4 keretrendszert használtunk,ennek köszönhetően az oldal reszponziv és bármely eszközről kényelmesen használható.PHP és MySQL technológiát alkalmaztunk a szerver oldali müködéshez, valamint JavaScript-et a klisen oldali müködéshez, a térkép megjelenítésre pedig HERE MAPS API-t.A biztonsági tényezőket is figyelembe vettük, regisztrálásnál kötelező létező email-t megadni a felhasználónak,hiszen nem tudná aktiválni a fiókját.A jelszavakat MD5-ös karakterkódolással tároljuk és "SALT" technikát alkalmazzuk rajtuk,valamint a bevitelő mezőket mysql_real_escape_string(); módszerrel védjük.

Regisztráció

Az összes regisztrációs mezőre létrehoztunk egy JavaScript függvényt, ami ellenőrzi ,a felhasználó álltal bevitt adatokat.

Az alábbi kód a vezetéknév beviteli mezőt ellenőrzi :

```
function lastName () {



    var lastname = document.getElementById("form_lastname_input");
    var lastname_span=document.getElementById("lastname_span");
    var numbers = /^[a-zA-Z\u00C0-\u00FF]+$/;


    if( (lastname.value.length<2) || (!numbers.test(lastname.value))){
        lastname_span.innerHTML="A vezetékneved nem tartalmazhat betüt
és speciális karaktert(Minimum:2 karakter)" +
            "";
       // lastname_span.style.backgroundColor="red";
        lastname_span.style.color="red";
       return true;


    }else{
        lastname_span.innerHTML="Megfelelő név";
       lastname_span.style.color="darkgreen";
       return false;
```

Ezt a függvény-t onchange eseménnyel folyamatosan meghívjuk ezálltal a regisztráció kényelmes lesz a felhasználó számára, hiszen a hibaüzeneteket valós időben tudjuk feltüntetni

```
document.getElementById("form_name_input").onchange = function()
{firstName();check();};
```

Amennyiben az adatok megfelelőek a check függvény engedélyezni fogja a regisztráció gomb megnyomását

```
function check(){
    if(firstName(),lastName(),emailAdress(),password()){
        document.getElementById("submit_button").style.pointerEvents    =
"auto";
    }else{
        document.getElementById("submit_button").style.pointerEvents    =
"none";
    }
}
```

Az adatokat ajax technológiával továbbitja a register.php oldalnak

Miután ellenőriztük az adatokat real escape string módszerrel, és beállítottuk a jelszó-ra az md5-ös karakterkódolást, az oldal átadja az adatokat az insert_user() függvénynek, amely behelyezi az adatokat az adatbázisba.

```
$firstname=mysqli_real_escape_string($con,$_POST['form_name_input']);
$password=md5($password_one);
insert_User($firstname,$lastname,$username,$password,$email);
```

A térkép betöltése

Az alábbi kód lekéri a böngészőtől a felhasználó pontos koordinátáit,majd átadja őket a map() függvénynek.

```
function getCoords(){


navigator.geolocation.getCurrentPosition(map);

}
```

A map függvényt onload eseménnyel hivjuk meg és a here maps könyvtárat és api technikát alkalmazva beállítjuk a térkép paramétereit

```
window.onload=map(position);

var map = new H.Map(

document.getElementById('map'),

defaultLayers.vector.normal.map,

{

pixelRatio: window.devicePixelRatio || 1,

zoom: 15,


center: { lat: position.coords.latitude, lng:
position.coords.longitude }

});

var ui = H.ui.UI.createDefault(map, defaultLayers, 'en-US');


var mapEvents = new H.mapevents.MapEvents(map);



var icon2 = new H.map.Icon('../home/images/human.png');

coords2 = {lat: position.coords.latitude,
lng:position.coords.longitude};

marker2 = new H.map.Marker(coords2, {icon: icon2});
```

```
map.addObject(marker2);
```

A helyszínek betöltése

A helyszínek betöltésére REST API technikát alkalmazunk és onclick eseményt.

Amikor a felhasználó rákattint a feltüntett kategória egyikére akkor létrehozunk egy függvényt ami lekéri a here maps-tól az adatokat,majd ezeket az adatokat feltüntetjük a térképen.Mivel az adatok lekérésére AJAX és rest api technikát alkalmazunk ezért frissítés nélkül fog megváltozni a térkép kinézete és a rajta feltüntett helyszínek.

```
document.getElementById("goingout").addEventListener("click",
function(){

var http=new XMLHttpRequest();

var
url='https://places.ls.hereapi.com/places/v1/discover/explore?at='+posi
tion.coords.latitude+'%2C'+position.coords.longitude+'&cat=going-
out&apiKey=HQCRAxCAMwEmKNbhSrEW4JE5Vye_YbczGs0qx4zBfsM';

http.open("GET",url);

http.send();

console.log(url);


http.onreadystatechange=function (){


var result=JSON.parse(this.responseText);

console.log(result.results.items[1].title);

for(var i=0; i<result.results.items.length; i++){

// console.log(result.results.items[i].title)

// console.log(result.results.items[i].position)


nearby_going_out[i]=result.results.items[i].title;

//console.log(result.results.items[i].title);


var iko=result.results.items[i].icon;

var icon = new H.map.Icon('../home/images/goingout.png');

console.log(icon);

coords = {lat:result.results.items[i].position[0] ,
lng:result.results.items[i].position[1]};
```

```
                marker1 = new H.map.Marker(coords, {icon: icon});

                map.addObject(marker1);

            }

            for(var i=0; i<nearby_going_out.length; i++){

document.getElementById("places").innerHTML+=result.results.items[i].ti
tle;

            }
```

Közösségi aktivitás listázása

Miután a felhasználó belépett a rendszerbe, az users táblába beleirjuk a jelenlegi tartózkodási helyét,ezálltal tudni fogjuk,hogy melyik városban van és ,hogy melyik városból kell kilistázni a feltöltött képeket amelyek más felhasználók töltöttek fel.A posztok listázása is frissítés nélkül megjelennek, hiszen amikor feltöltünk egy képet az onclick esemény létrehoz egy ajax kérést a szervernek ami azonnal feltünteti nekünk a kért adatokat.

```
window.addEventListener('load',show_post);
```

```
document.getElementById("upload_photo").onclick=function(){


    const url = '../functions/insertpost.php'
    const form = document.querySelector('form')

   //var image=document.getElementById("imgupload");

   const files = document.querySelector('[type=file]').files;
   const formData = new FormData();


   for (let i = 0; i < files.length; i++) {
    let file = files[i]

    formData.append('files[]', file)
  }


  fetch(url, {
    method: 'POST',
    body: formData,
  }).then(response => {
    console.log(response)
  })
  setTimeout(show_post, 2000);


}
```

```
    function show_post(){
        var xmlhttp=new XMLHttpRequest();
        xmlhttp.onreadystatechange=function(){

document.getElementById("posts").innerHTML=xmlhttp.responseText;
        }

        xmlhttp.open("POST","../functions/showpost.php");
        xmlhttp.send();


    }
```

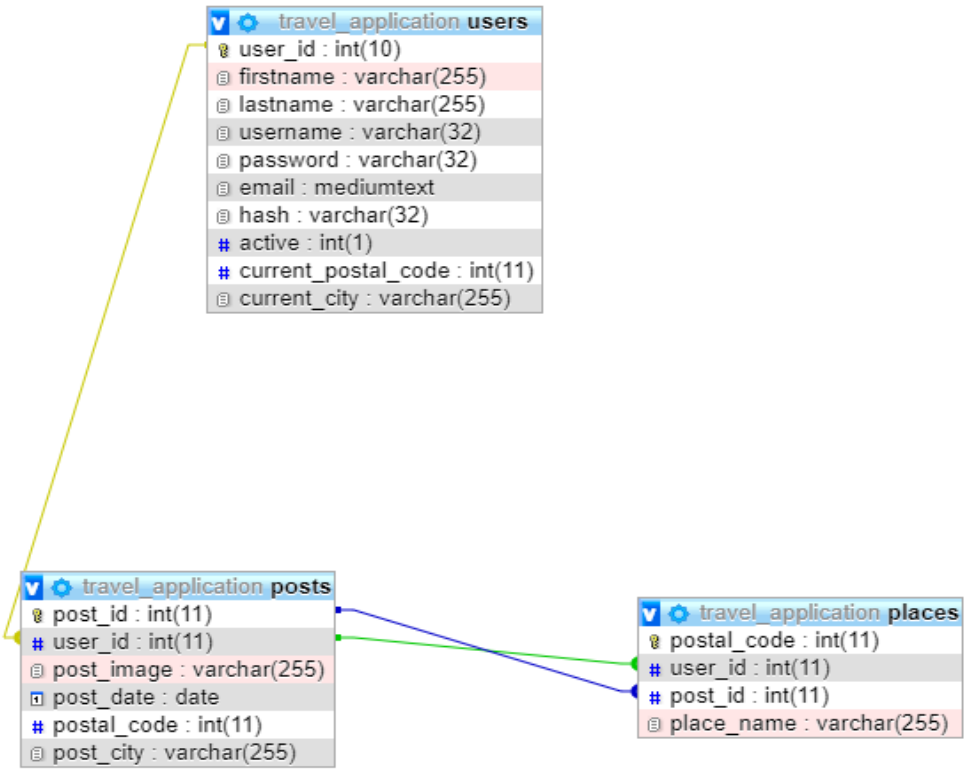
A showpost.php oldal megkeresi nekünk az adatbázisban azokat posztokat amelyek a mi jelenlegi tartózkodási helyünkön készültek és kilistázza őket :

```
$sql="SELECT * from  posts where postal_code='$postalcode'";
  $run=mysqli_query($con,$sql);
  while($row2=mysqli_fetch_assoc($run)){
     $image=$row2['post_image'];
      $posted_city=$row2['post_city'];
      $date=$row2['post_date'];
      $posted_user_id=$row2["user_id"];
      $query="SELECT * from users where user_id='$posted_user_id'";
      $run2=mysqli_query($con,$query);
      $row3=mysqli_fetch_array($run2);
      $posted_user_name=$row3['firstname'];
      $posted_user_lastname=$row3['lastname'];

      echo  "<div><p>$posted_user_name  $posted_user_lastname  <b>posted
image near you in $posted_city</b> $date</p>";

      echo            "<img            src='../functions/uploads/$image'
width='75%'></div>";
  }
```

AZ ADATBÁZIS STRUKTÚRÁJA (MySQL)



FELHASZNÁLT IRODALOM

A felhasznált irodalomban a szerzők nevét ábécé sorrendben kell feltüntetni. Példa:

1. Danny Goodman: „JavaScript Biblija”, Mikro knjiga, Beograd, 2000.

2. David Flanagan: „JavaScript: sveobuhvatni vodič” Mikro knjiga, Beograd, 2011.

W1. http://www.w3schools.com/

W2. http://www.jquery.com