**SZABADKAI MŰSZAKI SZAKFŐISKOLA**

**PROJEKTUM**

**Elektronikus ügyvitel tantárgyból**

**JELÖLT MENTOR**

**Tóth Dániel dr Zlatko Čović**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SZABADKA, 2018.**

# TARTALOM

Tartalom

[TARTALOM 2](#_Toc532134145)

[A FELADATKÖR MEGHATÁROZÁSA 3](#_Toc532134146)

[Kidolgozás 4](#_Toc532134147)

[Admin 4](#_Toc532134148)

[Felhasználói oldal 6](#_Toc532134149)

[Funkciók 7](#_Toc532134150)

[SEO linkek 7](#_Toc532134151)

[AZ ADATBÁZIS STRUKTURÁJA 8](#_Toc532134152)

[A PROJEKT MŰKÖDÉSÉNEK LEÍRÁSA 9](#_Toc532134153)

# A FELADATKÖR MEGHATÁROZÁSA

A projekt feladata, hogy egy weboldalt hozzunk létre, arra a célra, hogy a felhasználók kezelni, ellenőrizni, szabályozni tudják a pénzügyeiket. Az alkalmazás lehetőséget nyújt különböző „virtuális pénztárcák“ létrehozásában. A pénztárcákba különböző értékeket tudunk bevinni, amelyek a pénz áramlását jelzik. Létrehozhatunk különböző kategóriákat, amelyek a kiadásainkat vagy a bevételeinket tartalmazzák. Az oldalhoz tartozik egy adminisztrációs felület, ahol lehetőség van a felhasználók kezelésére, valamint, hogy admin jogosultságot adjunk.

# Kidolgozás

## Admin

A feladatot az adminisztrációs felület létrehozásával kezdtem el mivel ez a legfontosabb része egy weblapnak. Az adminisztrációs oldal 3 fő részből áll. A kezdőoldal, ahol egy statisztikát láthatunk, arról, hogy hányan jelentkeznek be naponta. A statisztika az elmúlt 20 nap adatait mutatja. A statisztikák kimutatásához a chart.js plugint használtam, amelyet a következő kódrészlet ábrázol

HTML:

<canvas id="myChart"></canvas>

JS:

**<script>**

**Var ctx = document.getElementById("myChart").getContext('2d');**

**var myChart = new Chart(ctx, {**

**type: 'line',**

**data: {**

**labels: [<?php echo implode(',', $all\_date); ?>],**

**datasets: [{**

**label: 'Page visitors (Last 20 day)',**

**data: [<?php echo implode(',', $all\_count); ?>],**

**backgroundColor: [**

**<?php echo "'rgba(" . rand(0, 255) . ", " . rand(0, 255) . ", " . rand(0, 255) . ", 0.4)'"; ?>**

**],**

**borderColor: [**

**<?php echo "'rgba(" . rand(0, 255) . ", " . rand(0, 255) . ", " . rand(0, 255) . ", 0.4)'"; ?>**

**],**

**borderWidth: 1**

**}]**

**},**

**options: {**

**scales: {**

**yAxes: [{**

**ticks: {**

**beginAtZero:true**

**}**

**}]**

**}**

**}**

**});**

**</script>**

Ezután a felhasználók kezelését készítettem el. Amikor kattintunk a felhasználó hozzáadása, avagy a módosítás gombra, akkor egy modál jelenik meg. A háttérben elindul egy ajax folyamat és ennek a végeredménye lesz a megjelenő form.

$('body').on('click', '.add-user', function() {

var type = 'new';

$.ajax({

url: "../ajax/get\_user.php",

data: {type: type},

method: "POST",

success: function(response) {

if (response.status == 'success') {

$('#edit-modal').modal('show');

$('#edit-modal .modal-title').html('Add user');

$('#edit-body').html(response.data);

}

},

error: function() {

},

dataType: "json"

});

});

Amikor a formot elküljük, akkor php segítségével ellenőrizzük a bevitt, adatokat. Amennyiben egy kötelezően kitöltendő mező üresen maradt, illetve helytelenül lett kitöltve, akkor egy hibaüzenet jelenik meg a képernyőn.

Az adminok hozzáadása és törlése végtelenül egyszerű. Egy listában megjelenik az összes aktív felhasználó. Ha egy felhasználónak admin jogosultságot akarunk adni, csupán annyi a dolgunk, hogy kiválasztjuk a listából és kattintunk a save gombra.

## Felhasználói oldal

Az adminisztrációs oldal után következett a felhasználói oldal. Az első lépés volt, hogy megalkossam a html kódot, amely az oldal kinézetéhez szükséges. Meta jelölőelemeket is használtam. Az oldalt Bootsrap segítségével készítettem el.

A html kód után a bejelentkezés és a regisztrációs kódot írtam meg. Regisztrációs és bejelentkezés során figyletem a biztonságra. A kötezelő mezőket mindíg ki kell tölteni, valamint helyes adatokat kell megadni. A jelszót md5 ös karatkerkódolással valamint SALT-al láttam el.

// Bejelentkezés

if (isset($\_POST['login']) && !isset($\_SESSION['user\_id'])) {

// Validáljuk az email és password értékét

$email = mysqli\_real\_escape\_string($connection, $\_POST['email']);

$password = mysqli\_real\_escape\_string($connection, $\_POST['password']);

// Salt hozzáadása a jelszóhoz, azután titkosítás

$password = md5(SALT . $password);

$sql\_string = "SELECT id, first\_name FROM users WHERE email = '{$email}' AND password = '{$password}'";

$user\_data = mysqli\_query($connection, $sql\_string);

$user\_data = mysqli\_fetch\_assoc($user\_data);

if (!empty($user\_data)) {

// Sikeres bejelentkezés -> session létrehozása -> logolás

$\_SESSION['user\_id'] = $user\_data['id'];

$\_SESSION['first\_name'] = $user\_data['first\_name'];

$messages['success'][] = 'Successfull login!';

$sql\_string = "INSERT INTO login\_statistics SET logged\_at = '{$now}', added\_at = '{$now}'";

mysqli\_query($connection, $sql\_string);

}

else {

$messages['error'][] = 'This user does not exist!';

}

}

### Funkciók

1. Elektronikus pénztárcák megjelenítése: A legelső dolog, az hogy lekérjük az összes aktív pénztárcát. Az pénztárcákat egy listában jelenítsük meg. Ha a lista eleme megváltozik, akkor egy ajax kérvény segítségével lekérjük az új pénzátviteli adatokat és megjelenítjük a táblázatban. Továbbá lehetőségünk van a pénztárcák szerkesztésére és újakat is létrehozhatunk.
2. Pénz átvitelek kezelése: Lehetőségünk van új átvitelt létrehozni, valamint módosítani az adatot, amennyiben véletlenül félreírtunk valamit
3. Kategóriák: Lehetőségünk van kategóriák létrehozására és szerkesztésére. A kategóriáknak van nevük, valamint egy érték amely azt jelzi, hogy a pénzátvitel kimenő, vagy bemenő típusú.
4. Az oldal alján egy diagram (chart.js) mutatja be nekünk a meglévő pénztárcáinkat és a tartalmukat.

### SEO linkek

Az oldal seo linkeket használ. Ezek arra szolgálnak, hogy a felhasználók jobban boldoguljanak az oldalon és könnyebben eligazodjanak rajta.

Íme a .htaccess file tartalma:

RewriteEngine on

RewriteRule ^home/$ ./index.php

RewriteRule ^logout/$ ./index.php?logout

RewriteRule ^transfer/([0-9]+)/$ ./index.php?transfer&id=$1

RewriteRule ^transfer/$ ./index.php?transfer

RewriteRule ^editwallet/([0-9]+)/$ ./index.php?wallet=$1

RewriteRule ^addwallet/$ ./index.php?wallet=new

RewriteRule ^categories/$ ./index.php?category=show

RewriteRule ^editcategory/([0-9]+)/$ ./index.php?category=$1

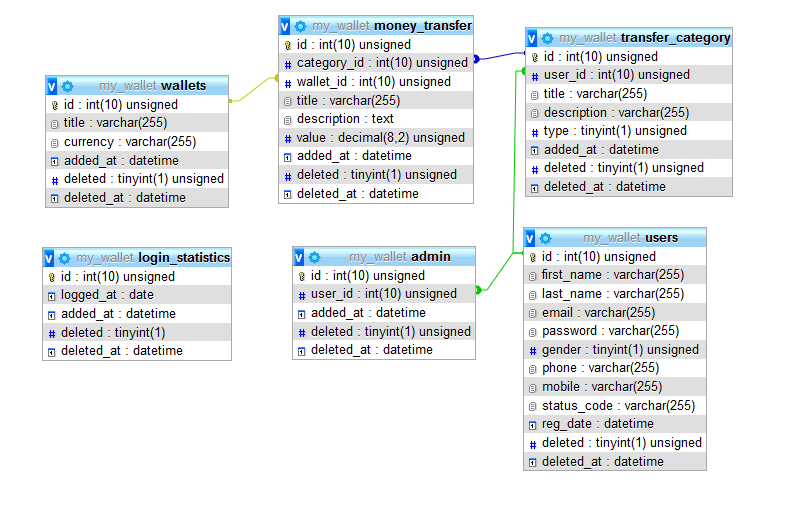
RewriteRule ^addcategory/$ ./index.php?category=new

RewriteRule ^delete/([0-9]+)/$ ./index.php?del=$1

RewriteRule ^deletewallet/([0-9]+)/$ ./index.php?del\_wallet=$1

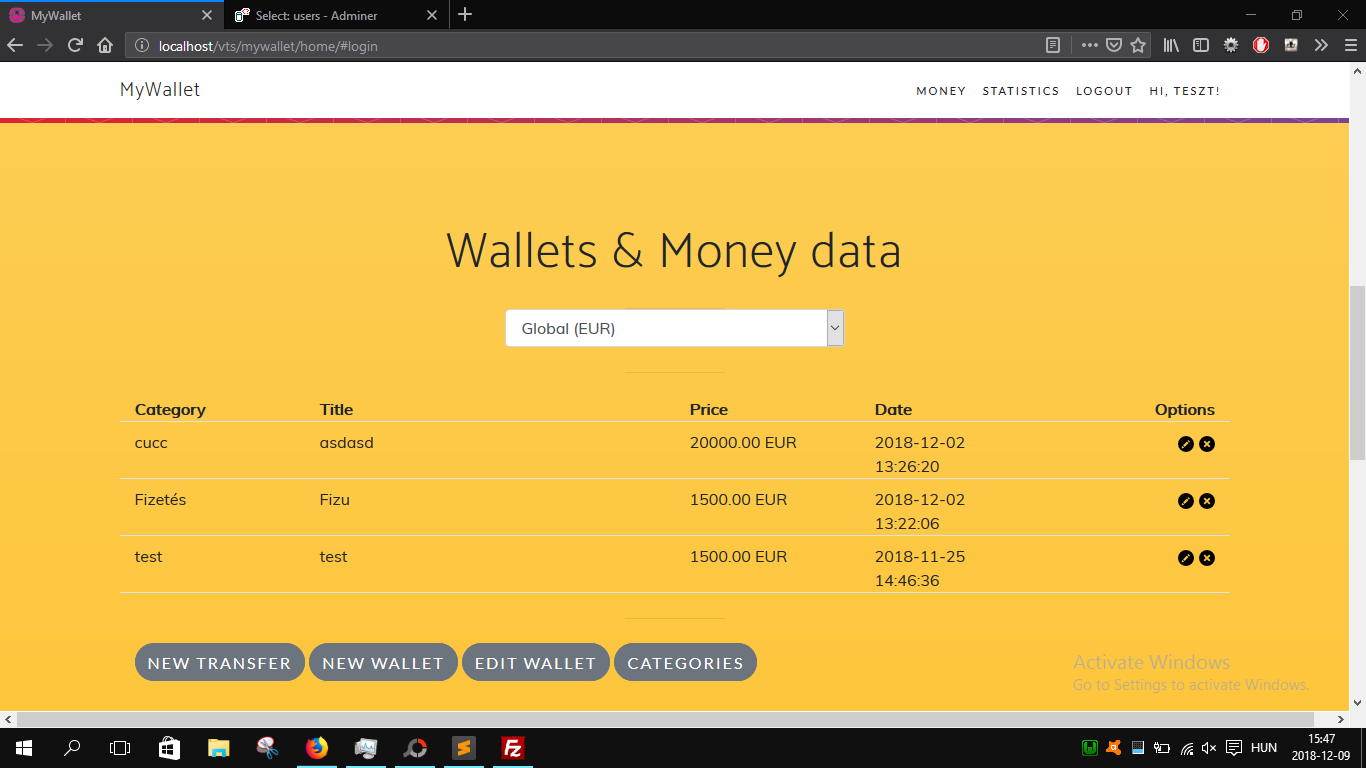
RewriteRule ^categories/delete/([0-9]+)/$ ./index.php?category=show&del\_cat=$1

# AZ ADATBÁZIS STRUKTURÁJA

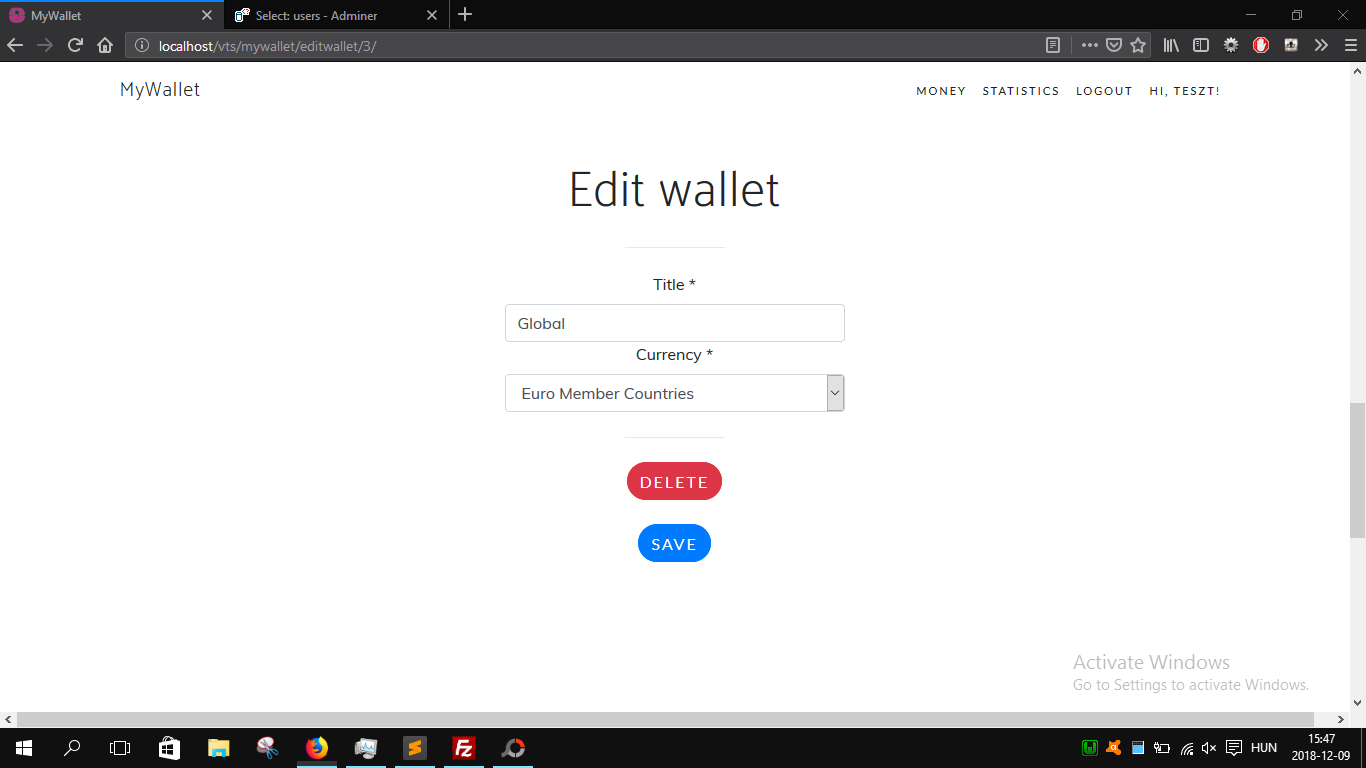
****

# A PROJEKT MŰKÖDÉSÉNEK LEÍRÁSA

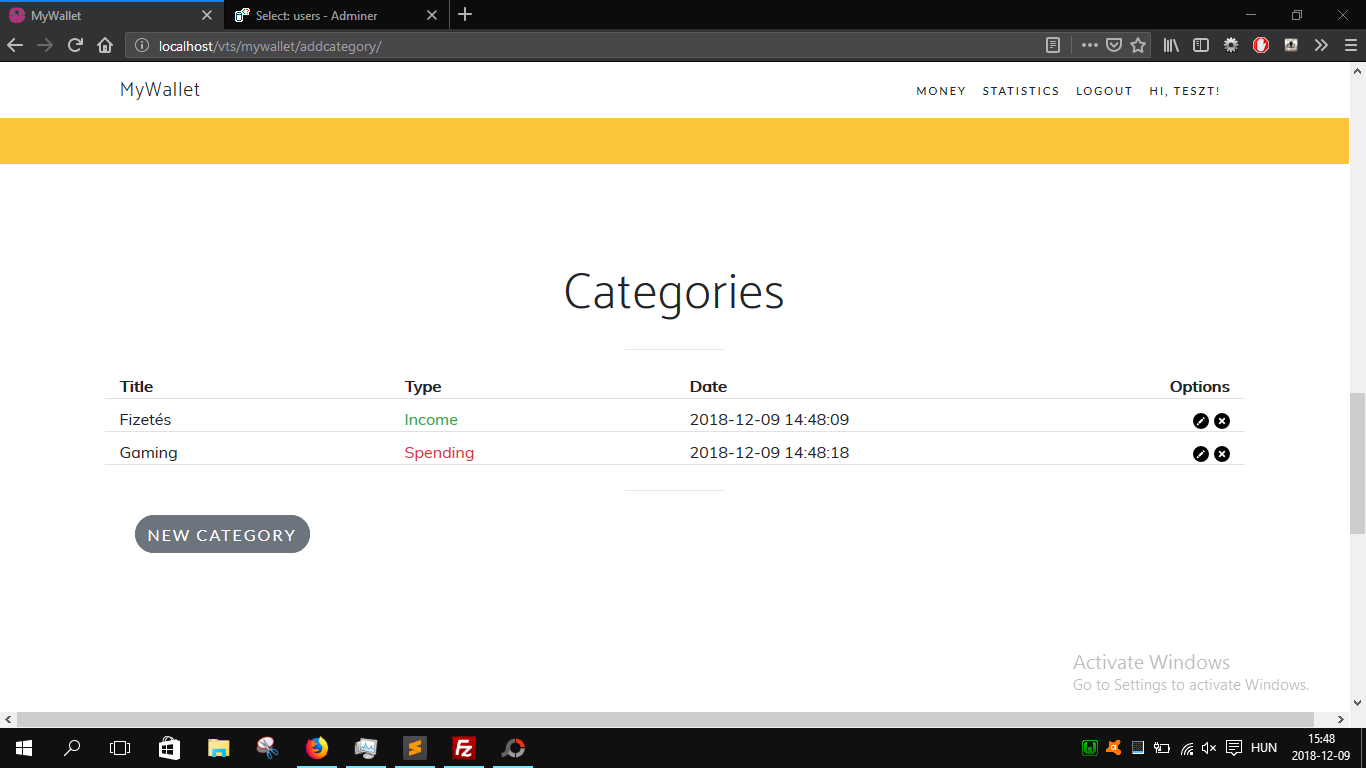
A pénzátvitelek listája és kezelése:



Pénztárcák kezelése:

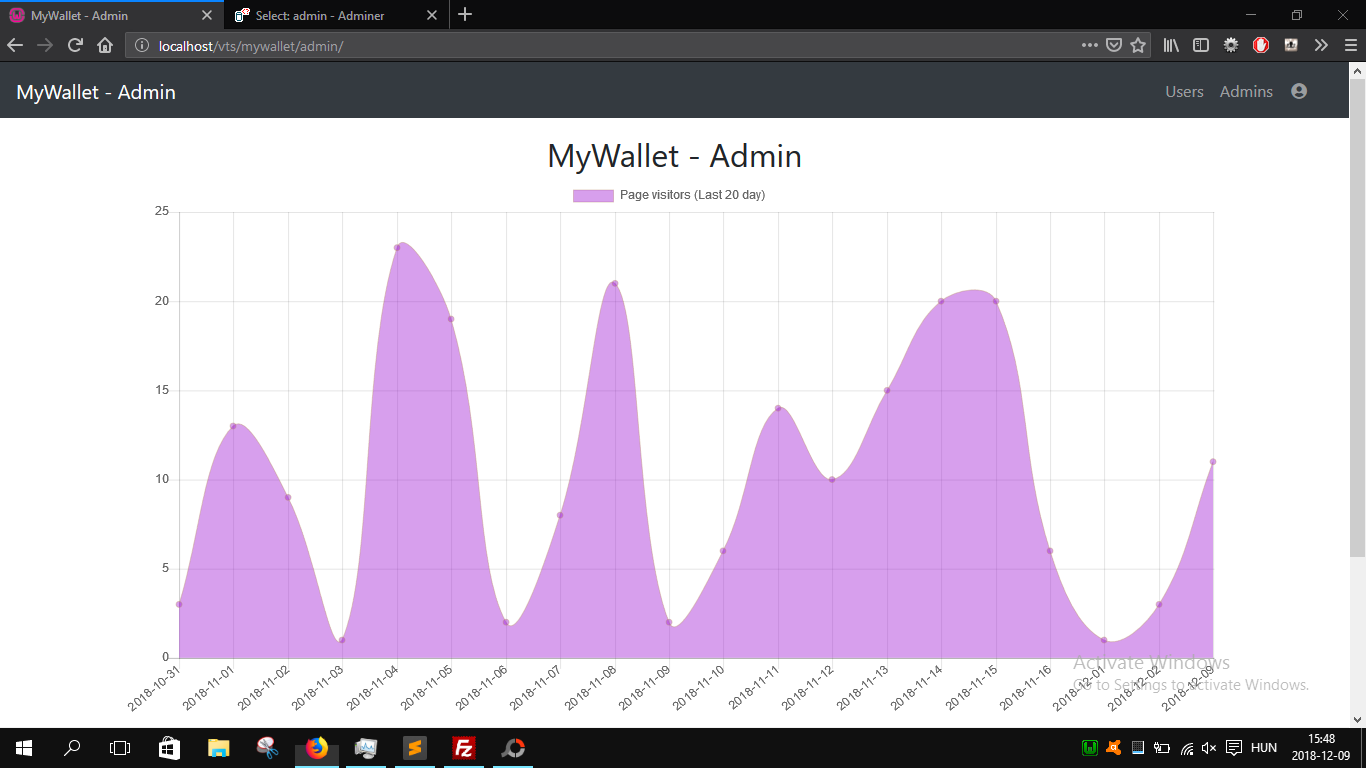


Kategóriák kilistázása:

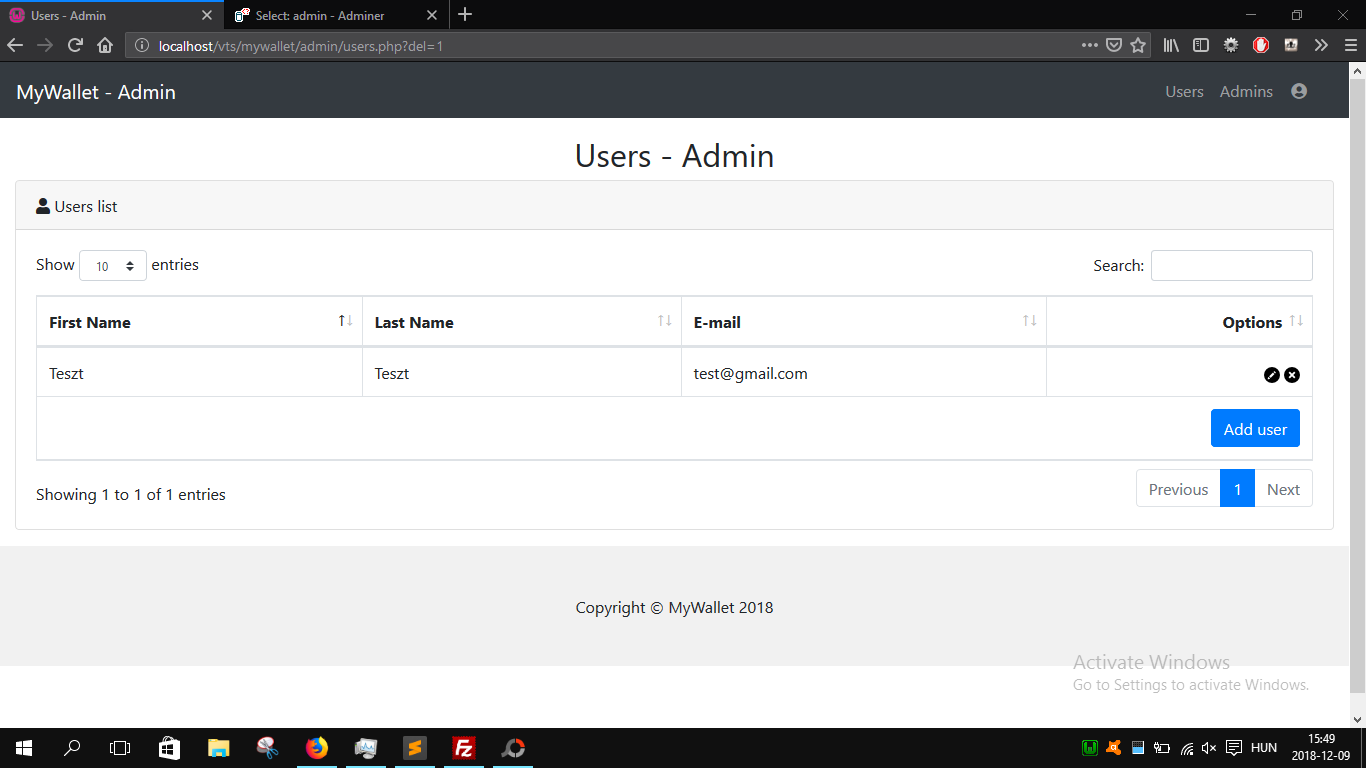


Adminisztrációs oldal:

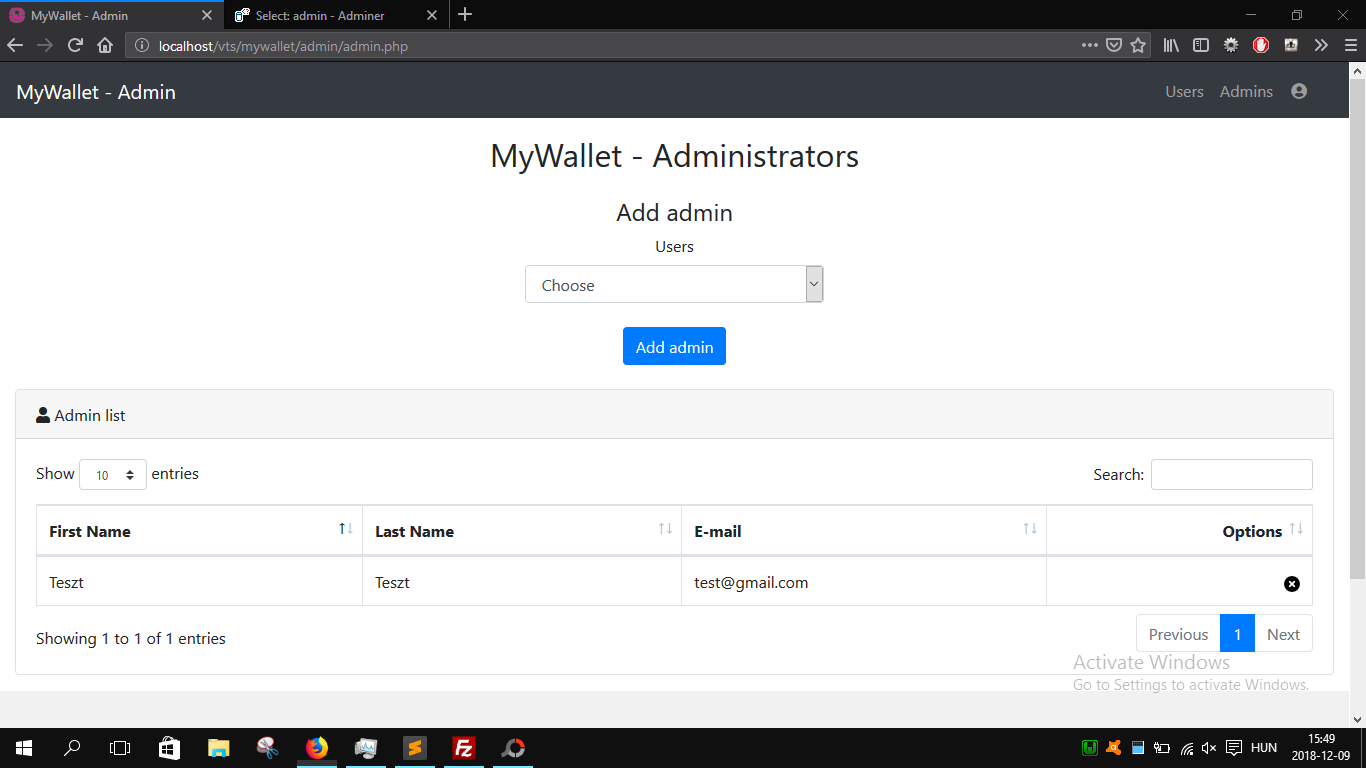
Statisztika:



Felhasználók kezelése:



Adminisztrátorok kezelése:



**FELHASZNÁLT IRODALOM**

A felhasznált irodalomban a szerzők nevét ábécé sorrendbenkell feltüntetni.Példa:

1. James Steele, Nelson To, “Izrada aplikacija pomoću paketa Android SDK”, Mikro knjiga, Beograd, 2011.
2. Bruce Lawson, Remy Sharp, “Uvod u HTML5 za programere”, Mikro knjiga, Beograd, 2012.

W1. <http://appinventor.mit.edu/>

W2. <http://puravidaapps.com/snippets.php>