**Debreceni Szakképzési Centrum**

**Baross Gábor Középiskolája és Kollégiuma**

**Debrecen, 4030**

**Budai Ézsaiás u. 8/A.**

**A szakképesítés azonosító Száma:54 213 05**

**Szakképesítés megnevezése: Szoftverfejlesztő**

**Záródolgozat címe:**

„Mit üzen neked?” mobil app és weblap fejlesztés

**Záródolgozat készítője:**

Mező Tamás Zsolt

**Konzulens: Dr. Várbíróné Nahaji Anikó**

**Konzultációs időpontok:**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Debrecen, 2023.**

Tartalomjegyzék

[Bevezetés 3](#_Toc32482713)

[Felhasználói dokumentáció 4](#_Toc32482714)

[Feladatspecifikáció 4](#_Toc32482715)

[Rendszerkövetelmények 4](#_Toc32482716)

[Program telepítése, elérése 4](#_Toc32482717)

[Program használatának leírása 5](#_Toc32482718)

[Fejlesztői dokumentáció 6](#_Toc32482719)

[Témaválasztás indoklása 6](#_Toc32482720)

[Alkalmazott fejlesztői eszközök 6](#_Toc32482721)

[Rendszerterv 6](#_Toc32482722)

[Adatmodell leírása 6](#_Toc32482723)

[Program leírása 7](#_Toc32482724)

[Részletes feladatspecifikáció, algoritmusok 12](#_Toc32482725)

[Tesztelés 16](#_Toc32482726)

[Továbbfejlesztési lehetőségek 16](#_Toc32482727)

[Összegzés 16](#_Toc32482728)

[Irodalomjegyzék 17](#_Toc32482729)

[Eredetiségnyilatkozat 18](#_Toc32482730)

# Bevezetés

A záródolgozatom egy vallással kapcsolatos alkalmazás, melyben Bibliai igéket lehet olvasni, leírni, hogy mit üzen számodra. Azért választottam ezt a témát, mert szeretek templomba járni.

Az applikációban és a weblapon is igéket lehet olvasni, és ha rányomtam a szövegre, akkor az ige jelenik meg, és szövegdobozba be kell írni, hogy mit jelent számodra a választott ige és mi az üzenete és el lehet küldeni azt. Mindkettőben van keresés funkciója. Menük vannak benne. Admin felületen lehet rendezni, ellenőrizni, hogy minden rendben van-e, statisztika nyomon követése, a nem odaillő tartalom törlése, a rossz kategóriában szereplő tartalom áthelyezése a megfelelőbe. Itt is lehet igéket írni. Nem bejelentkezett felhasználók nem törölhetnek. Az igék alatt megjelenik az időpont, a Bibliai igehely (könyv, fejezet és vers) és hogy milyen kategóriában van. Az idézetek és vélemények felett a sorszáma létható. Ez lehet akár az azonosítója. Kategóriákat a legördülő listából lehet kiválasztani (pl: ünnep, tanítások). Lapozni lehet, és egy oldalon 20 elem szerepelhet. Ezt lehet állítani például 10. El lehet mondani, hogy mit üzen számodra, hol hallottad stb. Új igét fel lehet vinni a rendszerbe a felvitel menüpont segítségével. Mobilon megosztási lehetőség, fájlba mentés. Többet is lehet egyszerre, akkor jelölő nézetek segítségével.

Az alkalmazást azoknak ajánlom, akik szeretnek Bibliát olvasni. Meg akarják osztani másokkal a gondolataikat és üzenetüket az igéről.

# Felhasználói dokumentáció

## Feladatspecifikáció

**Mobil applikáció funkciói:**

**Admin:**

* Keresés a vélemények és igék között
* Vélemény írása
* Könyv, fejezet és vers megjelenítése az Igénél
* Statisztika nyomon követése (vélemény, idézet) külön-külön
* Kategória
* Ige felvitele
* Az idézetek és vélemények felett azonosító.

## Rendszerkövetelmények

Mobil applikáció:

* Internet szükséges
* Android
* Bármelyik népszerű böngésző (Chrome, Firefox)

## Program telepítése, elérése

A programot a githubon lehet elérni, melynek elérhetőségei:

Mobil:

<https://github.com/mezotamas/mobilfrontend>

Weblap:

<https://github.com/mezotamas/webfrontend>

<https://github.com/mezotamas/webbackend>

Ezek működéséhez kell a nodejs.

A nodejs telepítése után mindhárom esetben npm i parancsot kell kiadni.

utána npm start.

## Program használatának leírása

**Mobil applikáció leírása**

A mobil applikációban 6 menüpont van:

Kezdőlap, Idézet keresése, Felvitel, Vélemények, Statisztika, Névjegy

A kezdőlap menüben üdvözlőszöveg, alatta a lehetőségek gombjai találhatóak. Ebben a menüpontban lehet ugrani a menükben, a vissza gombbal lehet a kezdőlapra térni.  
Az Idézet keresésben meg lehet tekinteni az eddigi idézeteket időponttal, kategóriával, Bibliai hellyel ellátva, a szövegbeviteli mezőbe be lehet írni a kulcsszót, például szeretet. Ha megnyomják a gombot, akkor csak azokat az igéket jeleníti meg amiben szerepel a keresett szó.  
A Felvitel menüpontban adatokat lehet felvinni a rendszerbe a következő sorrendben: idézet szövege könyv, fejezet és vers, a lenyílóban pedig a kategóriát. A Felvitel gomb megnyomásával megtörténik az adatok felvitele.  
A Vélemények menüpontban le lehet írni, hogy mit üzen számunkra az ige. A fent található szövegbeviteli mezőben le lehet írni az üzenetet, a lenyílóban pedig a kiválasztott ige.   
A küldés gomb megnyomásával lehet elküldeni az üzenetet. Az üzenet alább jelennek meg.

A statisztika menüben meg lehet tekinteni a rendszerben található idézetek és vélemények számát külön-külön. A frissítéshez meg kell nyomni a Frissítés gombot, és frissülnek az adatok.

A névjegy menüpontban pedig a készítő neve.

# Fejlesztői dokumentáció

## Témaválasztás indoklása

## Alkalmazott fejlesztői eszközök

**Mobil applikáció:**

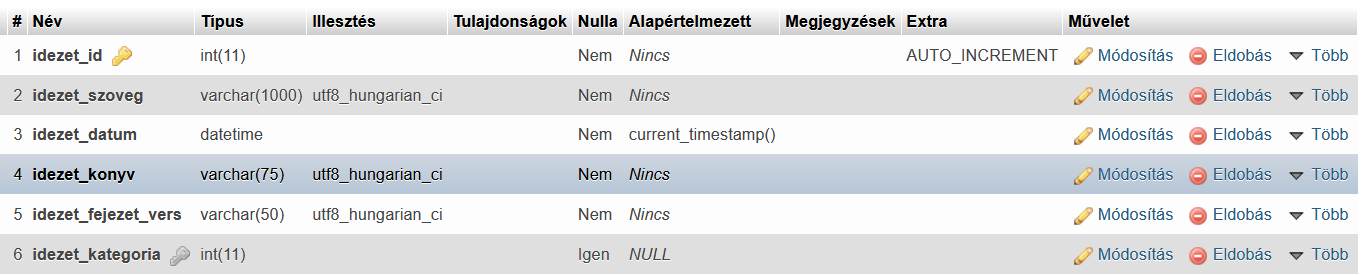
Ezt a programot react native nyelvben írtam(honlapja: https://reactnative.dev/) és a feljesztői környezet Expo Go(honlapja: https://expo.dev/), Javascript alapú. gépen kell hozzá nodejs(honlapja: https://nodejs.org/en/) fordító és terminál, és mobilon kell az Expo Go applikáció. A mobillal le kell olvasni a QR kódot amit kihoz a gépen a program. Ennek az a feltétele hogy azonos hálózaton kell lenni. A node-modulok az npmjs.com oldalról lehet telepíteni. Az adatbázist Mysql és PhpMyAdminban készítettem el. A kódokat VSCode nevű programban szerkesztettem.

## Rendszerterv

### Adatmodell leírása

Adatbázis: vallas

## idezet tábla:



idezet\_id: auto\_increment, egyedi azonosítót kap.

idezet\_szoveg: szöveg (max 50 karakter), ige szövegét tarolja

idezet\_datum: datetime, current\_timestamp(), az idézet felvitelének az időpontját tárolja

idezet\_konyv: szöveg (max 75 karakter), a könyv címét tárolja

idezet\_fejezet\_vers: szöveg(max 50 karakter), a fejezetet és a verset tárolja

idezet\_kategoria: szám típusú, idegen kulcs a kategoria tábla azonosítóját tárolja

## kategoria tábla:



kategoria\_id: auto\_increment, egyedi azonosítót kap

kategoria\_nev: szöveg(max 50 karakter), magát a kategória nevét tárolja

## velemeny tábla:



velemeny\_id: auto\_increment: egyedi azonosítót kap

velemeny\_szoveg: szöveg (max 50 karakter), a vélemény szövegét tárolja

velemeny\_idopont: datetime current\_timestamp(), a vélemény felvitelének az időpontját tárolja

velemeny\_idezet\_id: szám tipusú, idegen kulcs, az idezet táblának az azonosítóját tárolja

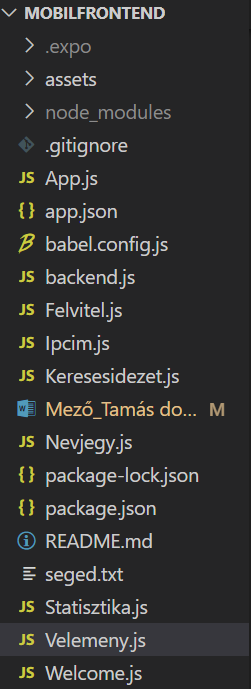
## kapcsolatok:

### Program leírása

használt php és egyéb js fájlok és feladataik leírása, hierarchia bemutatása rajzzal, kikopizhatod pl. commanderből, vscode-ból, hogy egy-egy mappádon belül milyen fájlok vannak, rövid leírás.

## Mobil applikació:

Az mappában 9 darab .js fájl található:



Fájlok:

App.js: Ez az indítófájl, ebben vannak a menük.

Felvitel.js: Ez a fájl az idézetek felvitelét végzi el.

Ipcim.js: Az távoli backend szerver címét tárolja. Kapcsolatot létesít.

Keresesidezet.js: Az idézetek között való keresést oldja meg.

Nevjegy.js: Ez csak a névjegy.

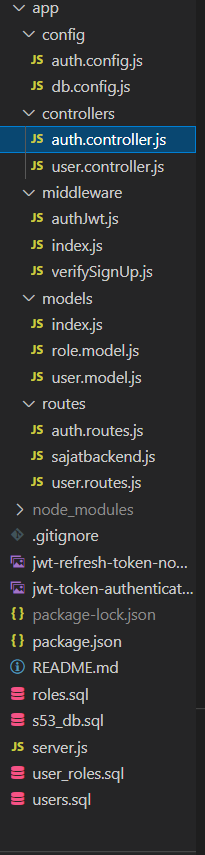
Statisztika.js: Statisztikai adatok lekérése.

Velemeny.js: Vélemények írását és keresését teszi lehetővé.

Welcome.js: Kezdőlap, itt van az üdvözlőszöveg és a gombok a menübe.

# Weblap:

### Backend:

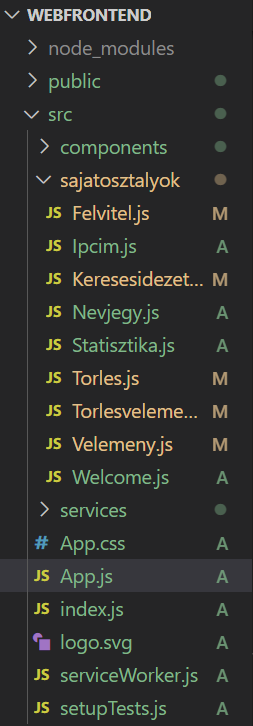


routes/sajatbackend.js: Itt vannak a backend-végpontok.

server.js: A web backend indítófájlja.

### Frontend:

A src/sajatosztalyok mappában 10 darab .js fájl található.



App.js: a web frontend indítóféjlja.

src/sajatosztalyok mappa:

Felvitel.js: Az idézetek felvitelét teszi lehetővé.

Ipcim.js: A szerver címét tárolja.

Keresesidezet.js: Az idézetek között való keresést teszi lehetővé.

Nevjegy.js: Névjegyet tartalmazza.

Statisztika.js: Statisztikai adatok lekérését teszi lehetővé.

Torles.js: Az idézetek törlését és írását teszi lehetővé weben keresztül csak admin számára.

Torlesvelemeny.js: Az vélemények törlését és írását teszi lehetővé weben keresztül csak admin számára.

Velemeny.js: A vélemények írását és keresését teszi lehetővé.

Welvome.js: Kezdőlap, és üdvözlés.

## Részletes feladatspecifikáció, algoritmusok

## Mobil applikáció:

### App.js:

    <NavigationContainer>

      <Drawer.Navigator useLegacyImplementation initialRouteName="Home">

        <Drawer.Screen name="Kezdőlap" component={Kezdo\_lap} />

        <Drawer.Screen name="Idézet keresése" component={Keresidezet\_lap} />

        <Drawer.Screen name="Felvitel" component={Felvitel\_lap} />

        <Drawer.Screen name="Vélemények" component={Velemeny\_lap} />

        <Drawer.Screen name="Statisztika" component={Stats\_lap} />

        <Drawer.Screen name="Névjegy" component={Nevjegy\_lap} />

      </Drawer.Navigator>

    </NavigationContainer>

Ez a kód a menük hivatkozásai. Ezzel lehet hivatkozni más js fájlokra. A name a menü címe.

### Felvitel.js:

 <Picker

                style={{backgroundColor:"#42adf5",color:"white",marginTop:10, marginBottom:10}}

                selectedValue={this.state.valaszto}

                onValueChange={(ertek) =>

this.setState({valaszto:ertek})

              }>

                  {this.state.dataSource.map(item=>

                <Picker.Item label={item.kategoria\_nev} value={item.kategoria\_id} />

          )}

              </Picker>

Leírása:

Ez a kód a lenyíló listát ami picker névre hallgat.

A „{this.state.dataSource.map(item=>

                <Picker.Item label={item.kategoria\_nev} value={item.kategoria\_id} />

          )}

” rész létrehoz egy tömböt amibe az elemeket belepakolja.

Velemeny.js:

import moment from 'moment';

import 'moment/locale/hu'

moment.locale('hu')

.  
.  
.

<Text style={{color:"black",fontSize:20,textAlign:"center",marginTop:15,marginBottom:5}}   >Időpont: {moment(item.idezet\_datum).format('YYYY MMMM DD, H:mm:ss')}  </Text>

A moment egy olyan modul amely lehetővé teszi, hogy a dátumot egyedi, olvasható formátumban való megjelenítését. Telepítése az npm i moment paranccsal lehetséges.

Ez szerepel a Keresesidezet.js fájban meg a weben, ahol a dátumot meg kell jeleníteni.

### Ipcim.js

module.exports = {

    ipcim:'http://nodejs.dszcbaross.edu.hu:22007/'

};

A távoli backend szerver címet tárolja, ahonnan az adatok érkeznek.

### Keresesidezet.js:

<FlatList

data={this.state.dataSource}

renderItem={({item}) =>

<View style={{borderWidth:2,borderColor:"blue", borderRadius:7, marginLeft:10, marginRight:10, marginTop:10}}>

<Text style={{marginRight:"auto",marginLeft:"auto",color:"blue",fontSize:20,textAlign:"center",marginLeft:10, marginRight:10, marginTop:10}} >Sorszám: #{item.idezet\_id} </Text>

<Text style={{marginRight:"auto",marginLeft:"auto",color:"blue",fontSize:20,textAlign:"center",marginLeft:10, marginRight:10, marginTop:10}} >{item.idezet\_szoveg} </Text>

<Text style={{fontStyle:"italic", fontSize:20,textAlign:"left",marginLeft:10, marginRight:10, marginTop:10}} >{item.idezet\_konyv} {item.idezet\_fejezet\_vers} </Text>

<Text style={{color:"black",fontSize:20,textAlign:"center",marginTop:15,marginBottom:5}} >Időpont: {moment(item.idezet\_datum).format('YYYY MMMM DD H:mm:ss')} </Text>

<Text style={{color:"blue",fontSize:20,textAlign:"center",marginTop:15,marginBottom:5}}>Kategória: {item.kategoria\_nev}</Text>

</View>

}

.

./>

A FlatList for ciklusként funkcionál, amely végigmegy a DataSource-on. Az item.valami hivatkozik az adatra amely az adatbázis oszlopának a nevével azonos. Ez felelős az összes idézet megjeleítéséhez.

### Statisztika.js:

 frissit=()=>{

    fetch(IP.ipcim+"statisztika1" )

    .then((response) => response.json())

    .then((responseJson) => {

      this.setState({

        isLoading: false,

        dataSource: responseJson,

      }, function(){

      });

    })

    .catch((error) =>{

      console.error(error);

    });

    fetch(IP.ipcim+"statisztika2" )

    .then((response) => response.json())

    .then((responseJson) => {

      this.setState({

        isLoading: false,

        dataSource2: responseJson,

      }, function(){

      });

    })

    .catch((error) =>{

      console.error(error);

    });

}

.

.

.

<TouchableOpacity

           style={styles.kekgomb}

           onPress={async ()=>this.frissit()}

         >

           <Text style={{color:"white", fontWeight:"bold",fontSize:15}}  >Frissítés</Text>

         </TouchableOpacity>

Ez felelős a statisztikai adatok frissítéséért. Amikor rányomnak a gombra, akkor meghívja a frissít függvényt, amit a fenti feladatok oldja meg.

## Weblap

### Felvitel.js

if(this.state.bevitel1=="" || this.state.bevitel2==""||this.state.bevitel3=="")

alert("Egyetlen sor sem maradhat üresen!")

else{

.  
.  
.

}

### Keresesidezet.js, Torles.js, Velemeny.js, Torlesvelemeny.js

if(this.state.bevitel1=="" )

      alert("Nem maradhat üresen!")

      else{

.  
.  
.

}

Ez a kód megakadályozza, hogy üres adatokat lehessen felvinni.

## Tesztelés

## Továbbfejlesztési lehetőségek

### Mobil applikáció:

A következő fejlesztések, funkciók hiányoznak még:

* Fájlba mentés.
* Megosztás.
* Oldalak hozzáadása, hogy ne legyen végtelen hosszú a menü.

### Weblap:

* Több elem törlése admin.
* Oldalak hozzáadsasa.

# Összegzés

A szakdogám azoknak lehet hasznos, akik érdeklődnek a Biblia iránt, és szeretnének másokkal megosztani a gondolataikat. A munkámat jónak értékelem és sok hasznos dolgot tanultam, amit máshol is használhatok.

# Irodalomjegyzék

tankönyvek php, css, html, sql, javascript

netes forrasok, lap címe es megnevezése, linkeket nyugodtan kopizz ide!!!

React native:

<https://reactnative.dev/docs/environment-setup>

Node modulok:

<https://www.npmjs.com/package/>

# Eredetiségnyilatkozat

Alulírott Mező Tamás Zsolt nyilatkozom, hogy záró dolgozatom a saját szellemi termékem.

Debrecen, 2018. márc. 28

.

aláírás