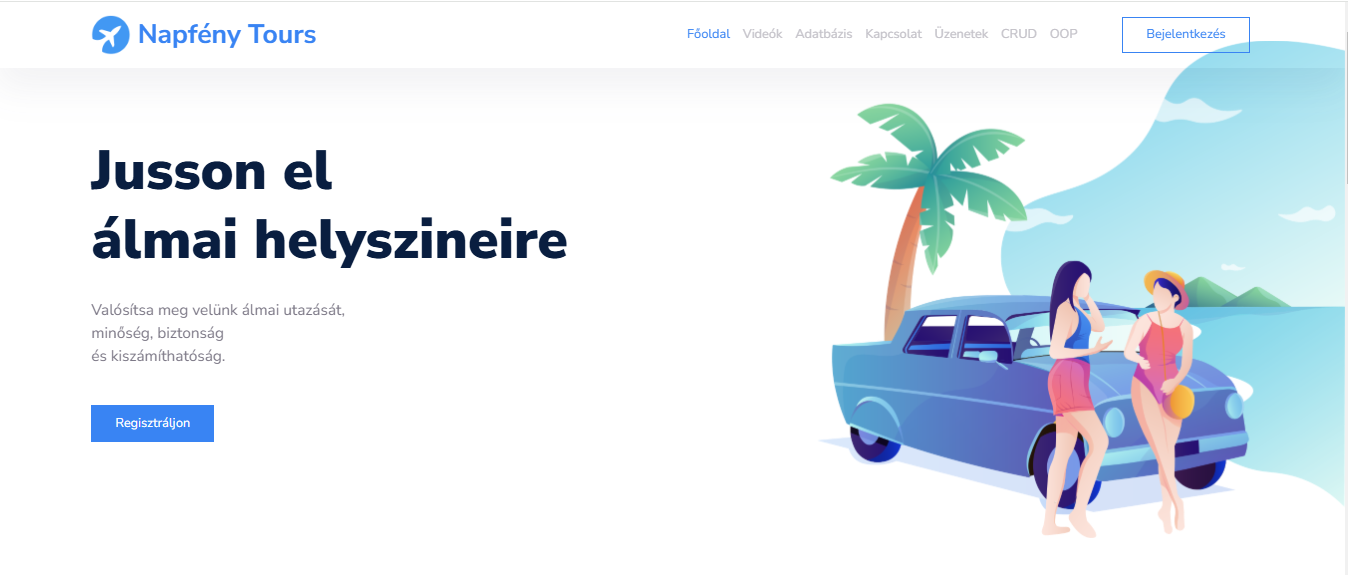
# Utazás Webalkalmazás Dokumentáció

**Készítette:**   
Czentye Györgyi Tímea (AX9NJQ) &  
Fekete Pál (PIXUNC)

**WEB-PROGRAMOZÁS II – Beadandó Feladat – NodeJS**

****

Dátum: 2024. 11. 24.

# Tartalomjegyzék

1. 1. Bevezetés .............................................. 3

2. 2. Reszponzív téma kiválasztása .............................................. 4

3. 3. Főoldal menü .............................................. 5

4. 4. Adatbázis menü .............................................. 6

5. 5. Kapcsolat menü .............................................. 8

6. 6. Üzenetek menü .............................................. 9

7. 7. CRUD menü .............................................. 10

8. 8. OOP-JavaScript menü .............................................. 12

9. 9. Szerver beállítás .............................................. 13

10. 10. Verziókövetés .............................................. 14

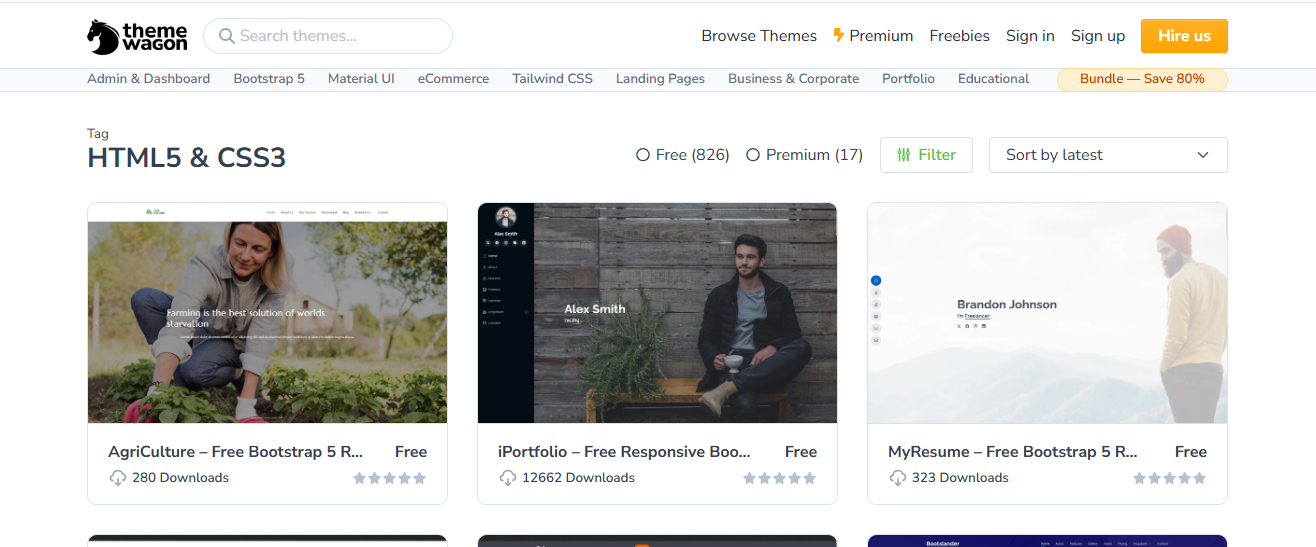
11. 11. Adatok .............................................. 15

# 1. Bevezetés

**Projekt neve:** Utazás  
**Fejlesztők:** Czentye Györgyi Tímea (AX9NJQ) és Fekete Pál (PIXUNC)  
**Cél:** Egy interaktív, reszponzív utazási webalkalmazás létrehozása adatbázis integrációval, kapcsolati lehetőségekkel és CRUD funkcionalitással.  
  
**Főbb technológiák:**- Backend: Node.js, Express.js  
- Adatbázis: MySQL  
- Frontend: HTML, CSS, JavaScript (OOP)  
- Verziókövetés: GitHub  
- Szerver: Linux-alapú tárhely

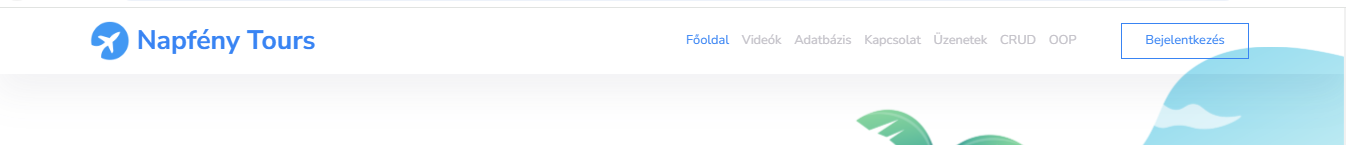
# 2. Reszponzív téma kiválasztása

**Kiválasztott téma:** Rhea téma   
Forrrás: <https://themewagon.com/theme-tag/html5-css3/>   
**Indoklás:** Ez a téma modern dizájnt, reszponzív megjelenést és könnyű testreszabhatóságot biztosít, amely megfelel az utazási oldal céljainak.



**Implementáció:**  
- A Rhea téma letöltése és beillesztése a projekt public könyvtárába.  
- A HTML sablon átalakítása dinamikus tartalmak betöltésére EJS segítségével.

# 3. Főoldal menü



**Funkciók:**  
- A vállalat rövid bemutatása szöveges és vizuális elemekkel.  
- Navigációs menü az oldalak közötti egyszerű váltáshoz.

**Megvalósítás:**  
- HTML sablon alapú főoldal, amely az EJS templating motorral dinamikusan betöltődő tartalmakat jelenít meg.  
- A stílus és layout a kiválasztott reszponzív téma alapján került kialakításra.

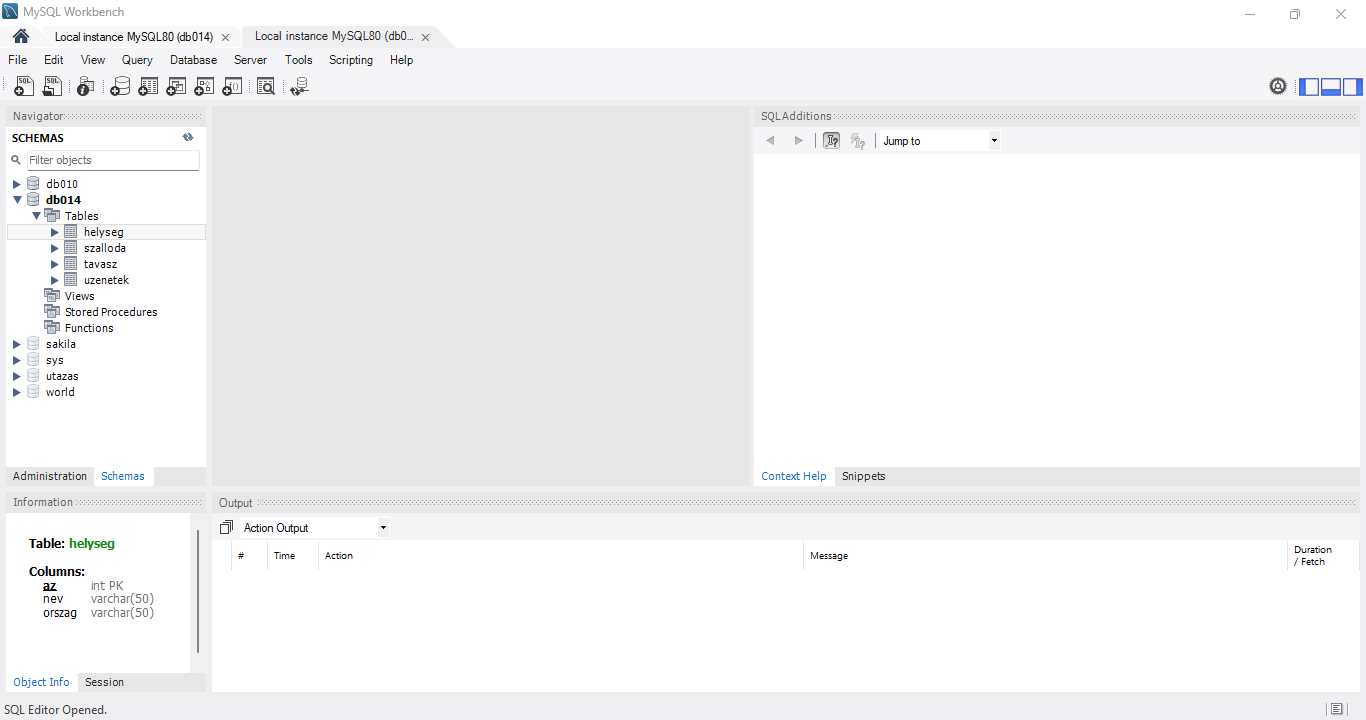
# 4. Adatbázis menü

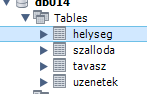
**Adatbázis kapcsolat:**  
- A Node.js MySQL modult használtuk az adatbázis kapcsolódáshoz és a lekérdezések futtatásához.

**Konfigurációs fájl**: **adatbazis.js** , amely tartalmazza az adatbázis csatlakozási adatokat.  
  
Kód példa:

**Megjelenített táblák az oldalon:**   
- utazasok: Utazási csomagok adatai  
- ugyfelek: Ügyfelek adatai  
- foglalasok: Foglalási információk

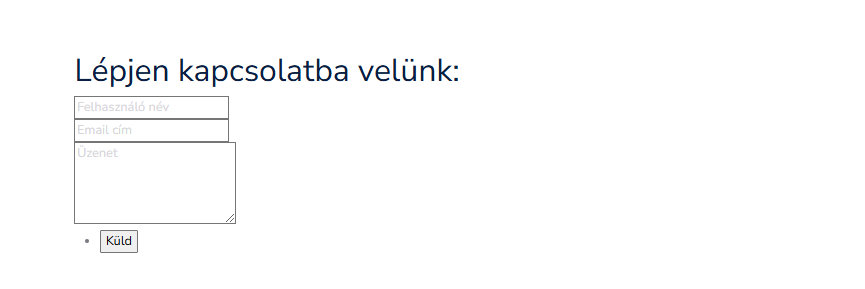
**Adatbázis:** db0XY / it jelen estben vagy db010 vagy db014/   
4 táblával megjelenítve benne:



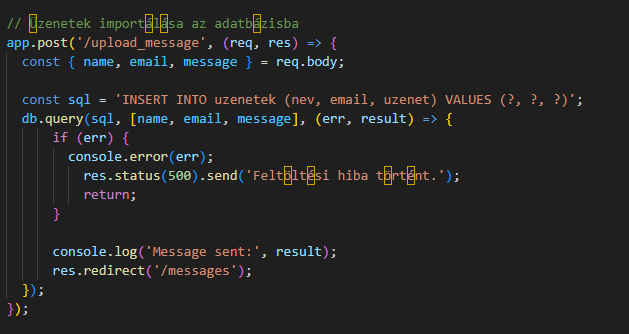
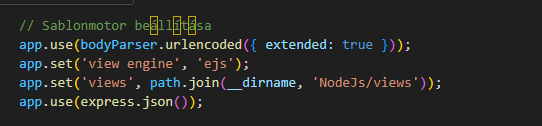


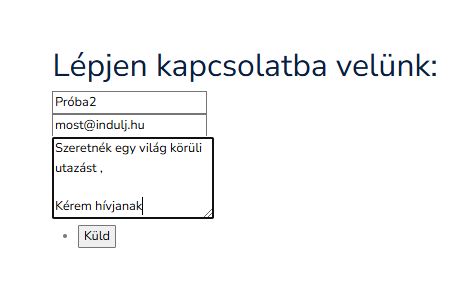
Az üzenetek tábla lásd, lásd 9. oldalon

# 5. Kapcsolat menü

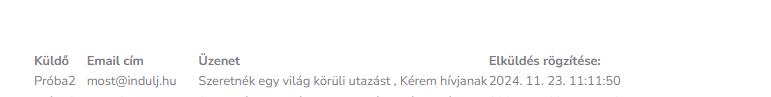


**Űrlap funkció**: Lehetővé teszi az üzenetek küldését az oldal tulajdonosának.

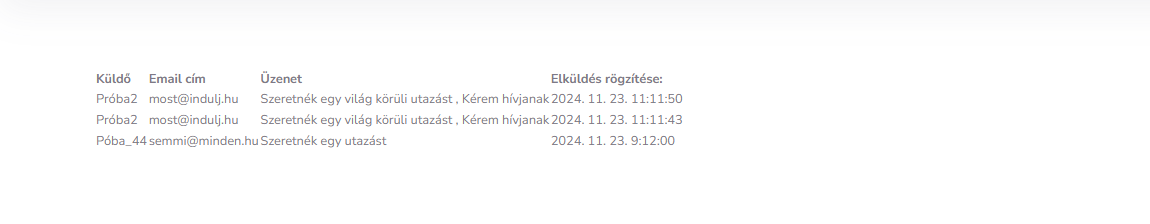
**Konfigurációs fájl: az indito.js és contact.ejs** kapcsolata  
**indito.js ben található a sablon motor beállításai között :**  
**Backend feldolgozás:**  
- Az űrlap adatait egy POST kéréssel küldjük a szerverre.  
- A szerver az adatokat az uzenetek táblába menti.  
**Kód példa: indito.js-ből** vett:

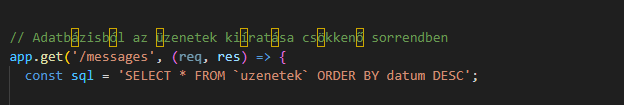
Működés közben:   
  
1. Kapcsolat menüpontból: - üzenet írás: majd Küld gomb megnyomása:

2. Megjelenítés az oldalon :



# 6. Üzenetek menü

A kapcsolatok Küld gomb lenyomásával az Üzenteknél megjelenik az összes küldött üzenetek:   
  
  
  
  
  
**Funkció: Az elküldött üzenetek fordított időrendben való megjelenítése.**  
  
**SQL lekérdezés:**



**Megvalósítás:**

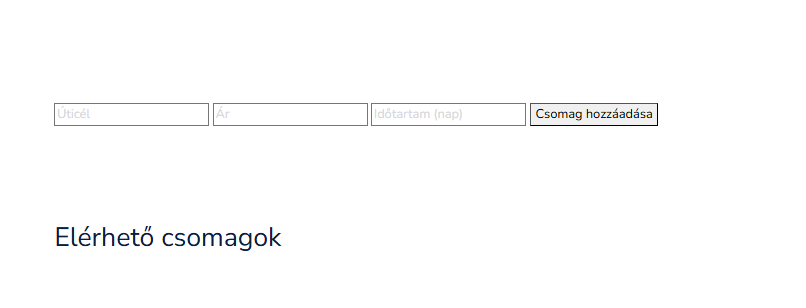
- Az adatokat az `uzenetek` táblából lekérdezzük, és táblázatos formában megjelenítjük.  
- Az EJS sablon segítségével a megjelenítés dinamikusan frissül.

# 7. CRUD menü

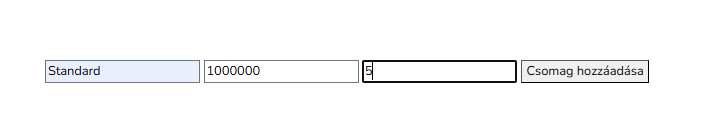
Funkció: Teljes CRUD műveletek az utazasok táblához.  
Példák:  
Adatok lekérése:  
```javascript  
app.get('/utazasok', (req, res) => {  
 db.query('SELECT \* FROM utazasok', (err, results) => {  
 if (err) throw err;  
 res.render('utazasok', { utazasok: results });  
 });  
});  
```  
Adatok módosítása:  
```javascript  
app.post('/utazasok/modosit/:id', (req, res) => {  
 const { nev, ar } = req.body;  
 const query = 'UPDATE utazasok SET nev = ?, ar = ? WHERE id = ?';  
 db.query(query, [nev, ar, req.params.id], (err) => {  
 if (err) throw err;  
 res.redirect('/utazasok');  
 });  
});  
```  
Adatok törlése:  
```javascript  
app.post('/utazasok/torles/:id', (req, res) => {  
 const query = 'DELETE FROM utazasok WHERE id = ?';  
 db.query(query, [req.params.id], (err) => {  
 if (err) throw err;  
 res.redirect('/utazasok');  
 });  
});  
```

# 8. OOP-JavaScript menü

**Feladat leírása szerint egy :**

OOP-Javascript menü: Az oldalon készítsen egy kis alkalmazást, amiben alkalmazza az objektum-orientált JavaScript elveit.

Megvalósítás: egy oop.ejs file létrehozása és annak konfigurálása  
Működés : Úticél, ár és időtartam adatok megadásával lehet csomagot hozzáadni , és törölni is

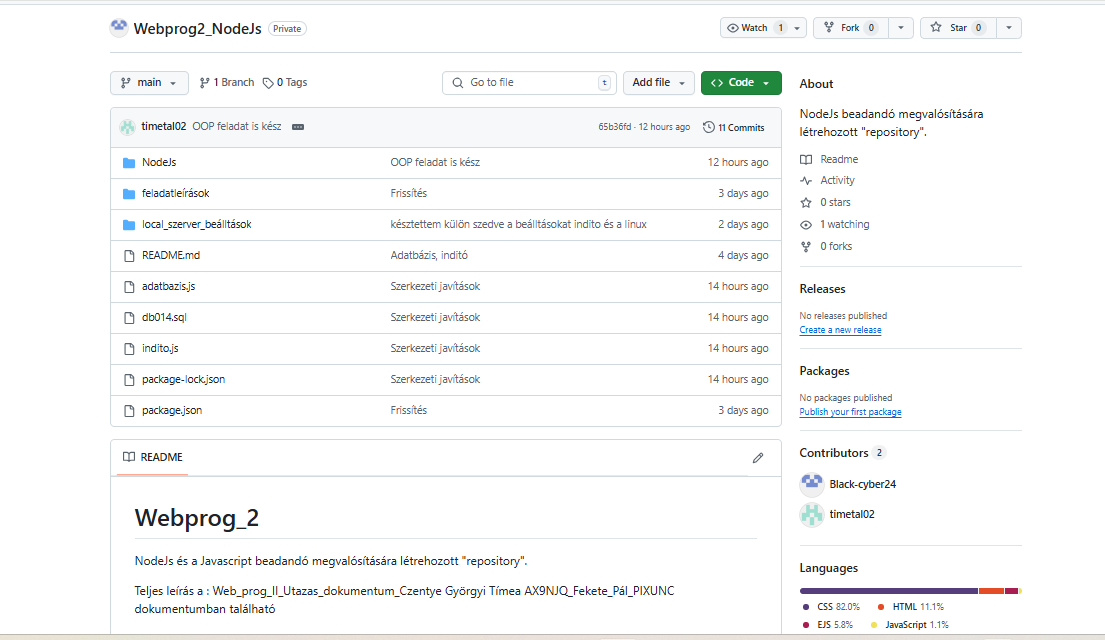
  
majd Csomag hozzáadása:

# 9. Szerver beállítás

Szerver belépési adatok:  
- IP-cím: [Megadandó]  
- Felhasználónév: [Megadandó]  
- Jelszó: [Megadandó]

Két helyen ment a fejlesztés Local és Linux szerver oldalon:

# 10. Verziókövetés

**GitHub repository:** <https://github.com/Black-cyber24/Webprog2_NodeJs.git>   
  


**t**[**imetal02**](https://github.com/timetal02) **:**  Czentye Györgyi Tímea AX9NJQ &

**Black-cyber24:** Fekete Pál PIXUNC

**Funkció:**   
- A verziókövetés lehetővé tette a csapatmunka megszervezését.

# 11. Adatok

GitHub projektjének URL címe: [Adja meg az URL-t]

Czentye Györgyi Tímea AX9NJQ : Linux adatok:

Linux belépési adatok:  
- IP-cím: [Megadandó]  
- Felhasználónév: [Megadandó]  
- Jelszó: [Megadandó]

**Black-cyber24:** Fekete Pál PIXUNC