**SZABADKAI MŰSZAKI SZAKFŐISKOLA**



**PROJEKTUM**

**Integrált web rendszerek tantárgyból**

Weather-App

**JELÖLT MENTOR**

**Borics Jonathan 26221084 dr Zlatko Čović**

**Konc Roland 26221111 dr Zlatko Čović**

**SZABADKA, 2025**

Tartalom

[Bevezetés: 2](#_Toc207689526)

[Rendszeráttekintés 3](#_Toc207689527)

[Architektúra: 3](#_Toc207689528)

[Adatbázis 4](#_Toc207689529)

[Táblák: 4](#_Toc207689530)

[Users 4](#_Toc207689531)

[Oszlopok: 4](#_Toc207689532)

[cities 5](#_Toc207689533)

[Oszlopok: 5](#_Toc207689534)

[favorite\_cities (kapcsolótábla) 6](#_Toc207689535)

[Oszlopok: 6](#_Toc207689536)

[weather\_archives 7](#_Toc207689537)

[Oszlopok: 7](#_Toc207689538)

[password\_resets 8](#_Toc207689539)

[Oszlopok: 8](#_Toc207689540)

[Kapcsolatok összefoglaló 9](#_Toc207689541)

[Integritások: 9](#_Toc207689542)

[API tesztek 10](#_Toc207689543)

[Auth 10](#_Toc207689544)

[Piackutatás 12](#_Toc207689545)

[Vizsgált rendszerek 12](#_Toc207689546)

[Felhasznált irodalom / Források 14](#_Toc207689547)

Bevezetés:  
Az Időjárás App célja, hogy több platformon (web, mobil – Expo/React Native) egyszerűen elérhető, valós idejű és rövid távú előrejelzést adjon, kedvencek kezelésével, és alap archívummal. A háttérrendszer PHP-ben készült, MySQL adatbázissal, a külső időjárás-adatforrás az **OpenWeather**.

# Rendszeráttekintés

## Architektúra:

**Frontend**: React Native + Expo (és Expo Web), Axios kliens.

**Backend**: PHP 8 + Composer csomagok (vlucas/phpdotenv, firebase/php-jwt, phpmailer/phpmailer, guzzlehttp/guzzle).

**Adatbázis**: MySQL (táblák: users, cities, favorite\_cities, weather\_archives, password\_resets).

**Külső szolgáltatás**: OpenWeather API (Current Weather + Geocoding).

**E-mail**: PHPMailer + Mailtrap (teszt).

***Frontend ↔ Backend ↔ MySQL, és Backend → OpenWeather, Backend → Mailtrap***

# Adatbázis

## Táblák:

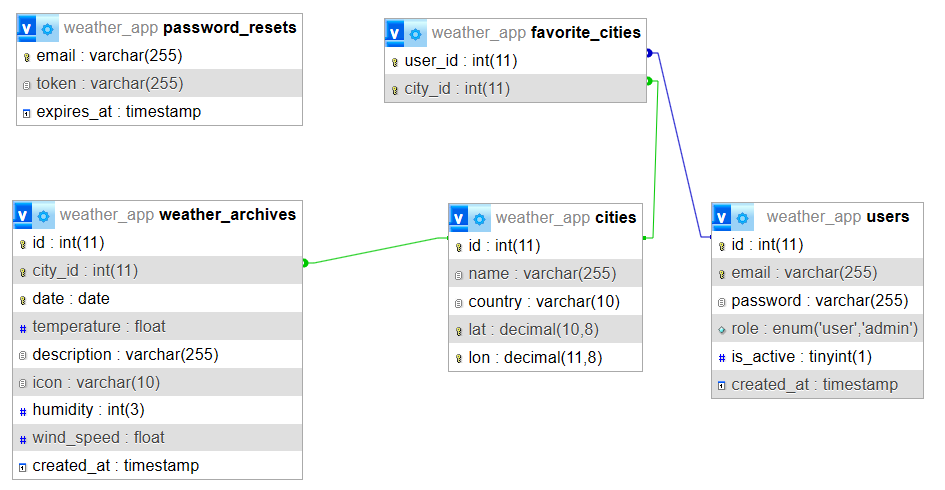
**users**: id, email (unique), password (bcrypt), role (user|admin), is\_active (0/1), created\_at

**cities**: id, name, country, lat, lon

**favorite\_cities**: user\_id, city\_id, created\_at (PK: összetett, vagy egyedi index)

**weather\_archives**: id, city\_id, temp, condition, description, icon, humidity, wind\_speed, collected\_at

**password\_resets**: email, token, expires\_at



# Users

## Oszlopok:

id (PK, INT AI), email (VARCHAR, UNIQUE), password (VARCHAR, bcrypt hash),

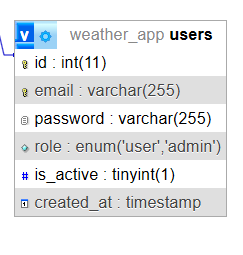
role (ENUM/user|admin vagy VARCHAR CHECK), is\_active (TINYINT 0/1), created\_at (TIMESTAMP)

Cél: regisztrált felhasználók nyilvántartása és jogosultságok.

Kulcsok & indexek:

PRIMARY KEY (id)

UNIQUE (email) – e-mail egyediség.



# cities

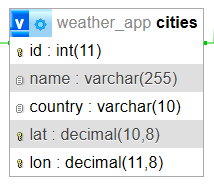
## Oszlopok:

id (PK, INT AI), name (VARCHAR), country (VARCHAR), lat (DECIMAL), lon (DECIMAL)

Cél: város törzsadatok, koordináták (OpenWeather hívásokhoz).

Kulcsok & indexek:

PRIMARY KEY (id)



# favorite\_cities (kapcsolótábla)

## Oszlopok:

user\_id (FK → users.id), city\_id (FK → cities.id)

Cél: felhasználó–város kedvencek N:M reláció.

Kulcsok & indexek:

Összetett elsődleges kulcs: PRIMARY KEY (user\_id, city\_id)

(így nem lehet duplikált kedvenc)

INDEX (city\_id) – népszerűségi statisztikákhoz gyors csoportosítás.

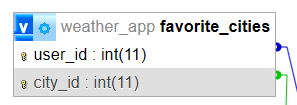
Kapcsolatok & szabályok:

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE

FOREIGN KEY (city\_id) REFERENCES cities(id) ON DELETE CASCADE

Kardinalitás:

users (1) — (N) favorite\_cities (N) — (1) cities → összességében N:M a users és cities között.



# weather\_archives

## Oszlopok:

id (PK, INT AI), city\_id (FK → cities.id),

temp (FLOAT/DECIMAL), condition (VARCHAR), description (VARCHAR), icon (VARCHAR),

humidity (INT), wind\_speed (FLOAT), collected\_at (TIMESTAMP)

Cél: adott városhoz időjárási pillanatképek archiválása (amikor felveszi kedvencnek, illetve ütemezetten is gyűjthető).

Kulcsok & indexek:

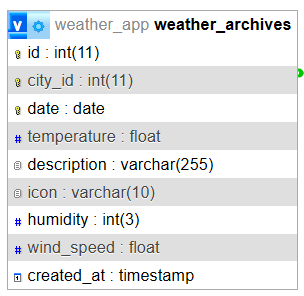
PRIMARY KEY (id)

INDEX (city\_id, collected\_at) – városonként, időrendben lekérdezéshez.

Kapcsolat & szabályok:

FOREIGN KEY (city\_id) REFERENCES cities(id) ON DELETE CASCADE

Kardinalitás: cities (1) — (N) weather\_archives



# password\_resets

## Oszlopok:

email (VARCHAR), token (VARCHAR, UNIQUE), expires\_at (DATETIME)

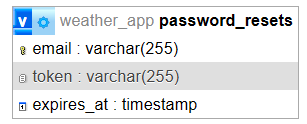
Cél: jelszó-visszaállítás időzített tokenekkel.

Kulcsok & indexek:

UNIQUE (token) – a token egyedi.

INDEX (email) – gyors törlés/lookup az e-mail alapján.

Kapcsolat: nincs kötelező FK a users.email-re (szándékos: biztonság/anonimitás, és ha nincs ilyen user, akkor is 200 OK).



# Kapcsolatok összefoglaló

users (1) — (N) favorite\_cities (N) — (1) cities

→ Minden user több várost is kedvenchez adhat, és egy városnak sok kedvelője lehet.

cities (1) — (N) weather\_archives

→ Egy városhoz tetszőleges számú időbélyeges mérés/pillanatkép.

password\_resets lazán kapcsolódik a users-höz az e-mail mezőn keresztül (FK nélkül), kizárólag a reset folyamat idejére.

# Integritások:

Egyediség: users.email UNIQUE; városok duplikációjának megakadályozása cities-ben (legalább UNIQUE(lat,lon)).

Kedvencek: PRIMARY KEY (user\_id, city\_id) → ugyanazt a várost nem lehet többször felvenni.

FK-k és CASCADE:

favorite\_cities.user\_id → users.id (ON DELETE CASCADE)

favorite\_cities.city\_id → cities.id (ON DELETE CASCADE)

weather\_archives.city\_id → cities.id (ON DELETE CASCADE)

Jogosultság-ellenőrzés: archívum olvasás csak olyan city\_id-re, ami az adott user kedvence.

Jelszókezelés: users.password bcrypt hash; reset token érvényesség és törlés felhasználás után.

Indexek: megfelelő indexek a gyakori műveletekhez (kedvencek listázása, archívum lekérés, népszerűség).

Időbélyegek: created\_at / collected\_at a visszakereshetőséghez.

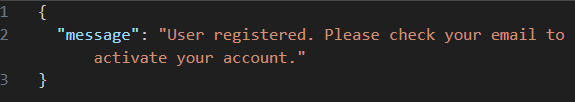
# API tesztek

## Auth

POST /auth/register

URL: <https://boys.stud.vts.su.ac.rs/weather-app/backend/api/auth/register>

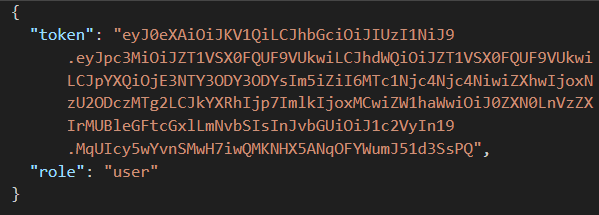
Json body: {"email":"test.user+1@example.com","password":"Asdasd123!"}



POST /auth/login

URL: <https://boys.stud.vts.su.ac.rs/weather-app/backend/api/auth/login>

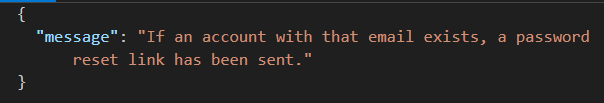
Json body: {"email":"test.user+1@example.com","password":"Asdasd123!"}



POST /auth/forgot-password

URL: <https://boys.stud.vts.su.ac.rs/weather-app/backend/api/auth/forgot-password>

Json body: {"email":"test.user+1@example.com"}



# Piackutatás

## Vizsgált rendszerek

Időkép (idokep.hu)

Funkciók: részletes előrejelzés, radarképek, riasztások, közösségi fotók.

Előnyök: nagyon részletes, széles körben ismert; valós idejű radarkép.

Hátrányok: reklámokkal túlzsúfolt, lassabb betöltés mobilon.

Köpönyeg (koponyeg.hu)

Funkciók: rövid és hosszú távú előrejelzés, hőtérképek.

Előnyök: egyszerű kereső, gyors alapadatok.

Hátrányok: kevesebb testreszabás, kevésbé modern mobilos UX.

AccuWeather (accuweather.com, mobilapp)

Funkciók: részletes, óránkénti előrejelzés, push értesítések, radar.

Előnyök: megbízható, több nyelv, profi mobilapp.

Hátrányok: reklámok; angol nyelv domináns, kevés magyar lokalizáció.

Yr.no (Norvég Meteorológiai Intézet)

Funkciók: tiszta, reklámmentes felület; precíz modellek.

Előnyök: gyors, egyszerű, nincs reklám.

Hátrányok: nincs magyar nyelv; városok néha nehezen kereshetők.

10.2 Kutatási eredmények (hallgatói megfigyelés)

Felhasználók elsődleges igényei: gyorsan elérhető alapadatok (hőmérséklet, csapadék), kedvenc városok mentése, reklámmentes megjelenítés.

Hiányosságok a vizsgált appokban:

Túl sok hirdetés (Időkép, AccuWeather).

Lassú betöltés gyengébb mobilokon.

Nincs egyszerű személyes időjárás-archívum.

Kevesen kínálnak offline gyorsítótárat vagy könnyű PWA telepítést.

További funkciók a saját projektben

A piackutatás alapján három olyan funkció, amit beépítünk:

Saját archívum per város

– A felhasználó kedvenceiről automatikusan tároljuk a napi előrejelzést / mérést, és azt bármikor visszanézheti.

Minimalista felület, reklámmentesen

– Kifejezetten diákoknak, oktatási célra: egyszerű, gyors, letisztult design, alacsony adatforgalommal.

Kedvenc városok központi szerepe

– Nem menük mélyén elrejtve, hanem a főképernyőn. Ez eltér a legtöbb konkurens megoldásától.

# Felhasznált irodalom / Források

Időkép – Magyarország időjárása

https://www.idokep.hu

Köpönyeg – Országos időjárás előrejelzés

https://www.koponyeg.hu

AccuWeather – Worldwide Weather Forecasts

https://www.accuweather.com

Yr.no – Weather forecast from the Norwegian Meteorological Institute

https://www.yr.no

OpenWeather – Weather API Documentation

https://openweathermap.org/api

PHPMailer – GitHub repository

https://github.com/PHPMailer/PHPMailer

Firebase PHP-JWT – GitHub repository

https://github.com/firebase/php-jwt

Dotenv (vlucas/phpdotenv) – GitHub repository

https://github.com/vlucas/phpdotenv

Stack Overflow – How to handle CORS preflight in PHP

https://stackoverflow.com/questions/8719276/cross-origin-request-headerscors-with-php-headers

(A CORS beállítás kódrészletének inspirációjához)

Stack Overflow – How to build a simple PHP router

https://stackoverflow.com/questions/17385864/php-simple-router

(Az index.php router logikájának kialakításához ötletként szolgált)