

NotNeptun Dokumentáció

Előszó

A program Apache XAMP-ot használ, mysql adatbázissal, nimble programozási nyelvben írva. Elméletileg teljesen cross platform (Windows, Linux, macOS), viszont Ubuntu alapú rendszeren való tesztelés után kiderült egy openssl hiba, emiatt Windows rendszeren ajánlott a futtatása.

Indítás

Indítás lehetséges egyenesen a NotNeptun mappájában lévő bináris fájlal, de ha valamiért nem működne akkor szükségünk lesz nim-re és nimble-re: [nim](#)

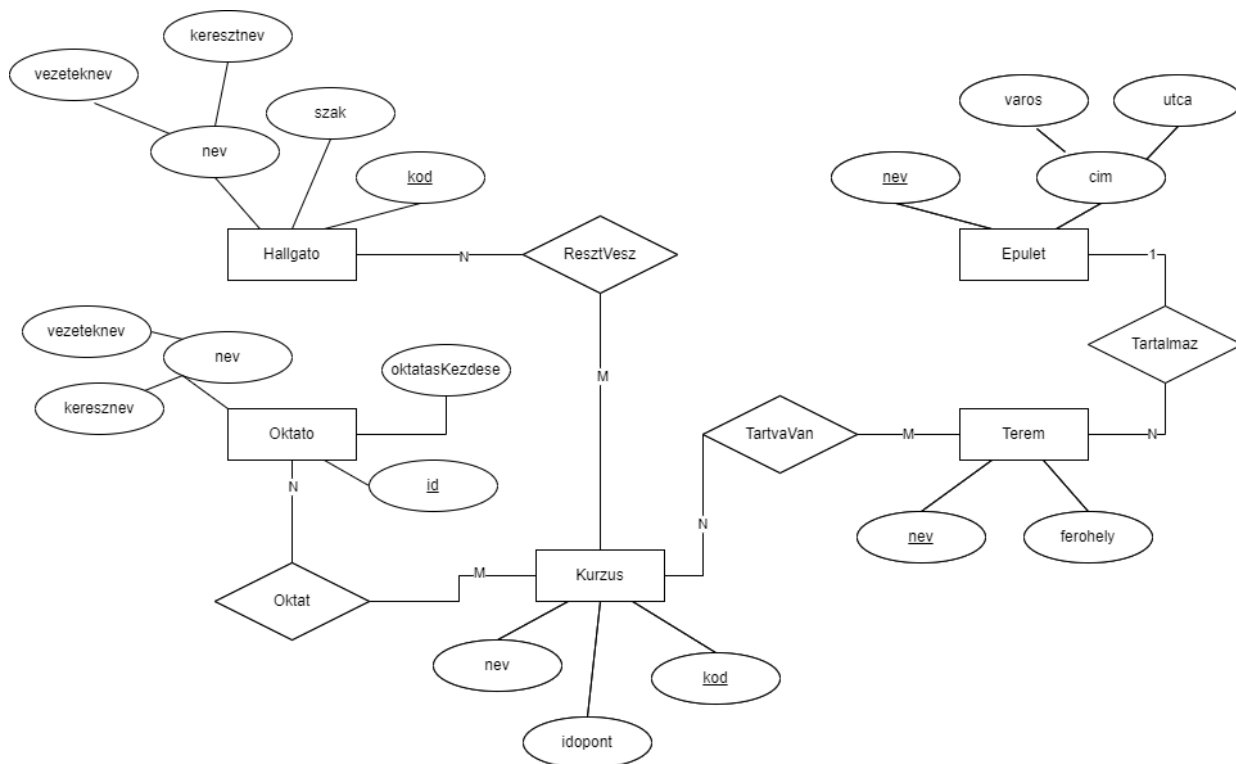
Emellett a fidget gui csomagra, amelyet `nimble install fidget` paranccsal lehet telepíteni.

Fordítani és indítani egyszerre a `nim c -r ./NotNeptun.nim -d:release --threads:on` paranccsal lehet. Utána pedig a létrehozott bináris fájlal.

Csatlakozni az adatbázisba a programon belül lehet. A kiexportált adatbázis alap adatai:

- host: localhost
- user: root
- password: root
- database: etr

Egyed-Kapcsolati Diagram



Táblák

Hallgato(kod, vezeteknev, keresztnev, szak)

- kod: a hallgató monogrammjából, a lista hosszából és véletlen számokból generált egyedi azonosító, legfeljebb 10 karakter, elsődleges kulcs

- vezeteknev: a hallgató vezetékeve, legfeljebb 255 karakter
- keresztnév: a hallgató keresztnéve, legfeljebb 255 karakter
- szak: hallgató szakja, legfeljebb 10 karakter

Oktató(kód, vezeteknev, keresztnév, kezdes)

- kód: az oktató monogrammjából, a lista hosszából és véletlen számokból generált egyedi azonosító, legfeljebb 10 karakter, elsődleges kulcs
- vezeteknev: az oktató vezetékeve, legfeljebb 255 karakter
- keresztnév: az oktató keresztnéve, legfeljebb 255 karakter
- szak: az oktató szakja, legfeljebb 10 karakter
- kezdes: mikor kezdett oktatni, évszám

Kurzus(kód, nev, idopont)

- kód: kurzus kódja, amely a nevének a rövidítése, legfeljebb 10 karakter, elsődleges kulcs
- nev: kurzus neve, legfeljebb 50 karakter
- idopont: nap amikor van a kurzus, datetime

Terem(nev, ferohely)

- nev: terem neve, legfeljebb 255 karakter, elsődleges kulcs
- ferohely: terem befogadó képessége, 11 hosszú szám

Epület(nev, varos, utca)

- nev: épület neve, legfeljebb 50 karakter, elsődleges kulcs
- varos: város ahol az épület található, legfeljebb 255 karakter
- utca: az utca a városban ahol található az épület, legfeljebb 255 karakter

ResztVesz(Hallgato.kód, Kurzus.kód): N:N kapcsolat

Oktat(Oktato.kód, Kurzus.kód) N:N kapcsolat

TartvaVan(Kurzus.kód, Terem.epuletnev) N:1 kapcsolat

Tartalmaz(epulet.nev, Terem.epuletnev) 1:N kapcsolat

Relációs Adatbáziséma

Hallgato(kód, vezeteknev, keresztnév, szak)

Oktató(kód, vezeteknev, keresztnév, kezdes)

Kurzus(kód, *terem.nev*, nev, idopont)

Terem(nev, *epuletnev*, ferohely)

Epület(nev, varos, utca)

ResztVesz(Hallgato.kód, Kurzus.kód)

Oktat(Oktato.kód, Kurzus.kód)

Funkcionális függőségek

Hallgato(kód, vezeteknev, keresztnév, szak)

{kód} -> { vezeteknev, keresztnév, szak }

Oktató(kód, vezeteknev, keresztnév, szak, kezdes)

{kód} -> { vezeteknev, keresztnév, szak, kezdes }

Kurzus(kod, terem.nev, nev, idopont)

{kod} -> { terem.nev, nev, idopont }

Terem(nev, epulet.nev, ferohely)

{ nev } -> { epulet.nev, ferohely }

Epulet(nev, varos, utca)

{ nev } -> { varos, utca }

2NF és 3NF

Hallgato(kód, vezeteknev, keresztnév, szak)

Oktato(kód, vezeteknev, keresztnév, szak, kezdes)

Kurzus(kod, terem.nev, nev, idopont)

Terem(nev, epulet.nev, ferohely)

Epulet(nev, varos, utca)

ResztVesz(Hallgato.kod, Kurzus.kod)

Oktat(Oktato.kod, Kurzus.kod)

Használt lekérdezések:

- SELECT * FROM hallgato;
- SELECT * FROM terem;
- SELECT * FROM kurzus;
- SELECT * FROM epulet;
- SELECT * FROM oktato;
- SHOW TABLES FROM etr;
- SHOW COLUMNS FROM [table] IN etr; ahol table egy változó
- INSERT INTO [table] VALUES ...; ahol table a különböző táblák az adatbázisban, viszont felsorolásuk nagyon hosszú lenne
- DELETE FROM [table] WHERE hallgato.kod = [key]; ahol table a különböző táblák, a key pedig a kulcs ami alapján törölünk
- SELECT * FROM [table] WHERE hallgato.kod = [key]; ahol table a különböző táblák, a kulcs pedig ami alapján keresünk
- UPDATE [table] SET ... WHERE [key]; ahol table a tábla amit frissítünk, ... az adatok, key pedig a kulcs, ami alapján frissítünk

Megvalósítás

A szoftver nim-ben volt írva, és mivel a fidget ui library nem támogatja erősen az objektum orientált programozást, ezért teljesen imperatív módon van megírva.

Kettő fontos fájl van, a NotNeptun.nim és a db.nim. A NotNeptun a megjelenítésért és adatok felviteléért felel, a db pedig az adatbázis műveletekért.

Különböző funkciók, fájlanként:

NotNeptun.nim:

- viewHallgatok
A hallgato tábla megjelenését biztosítja
- viewKurzusok
A kurzus tábla megjelenését biztosítja

- viewTermek
A terem tábla megjelenését biztosítja
- viewEpuletek
Az epulet tábla megjelenését biztosítja
- viewOktatok
Az oktato tábla megjelenését biztosítja
- hozzaAd
A táblákhoz való hozzáadást biztosítja
- torles
A táblákból való törlést biztosítja
- modositas
A táblákon a modositást biztosítja
- connect
Az adatbázis felé a csatlakozást továbbítja
- drawMain
A megjelenítésért felelős főleg, menü és háttér rajzolásáért
- startFidget
Ez indítja el az egész applikációt, kommunikál a fidget gui könyvtárral

db.nim:

- listHallgato
SQL üzenetet küld a szerver fele, nem vár adatot, visszaküld az összes hallgató adatait
- listTermek
SQL üzenetet küld a szerver fele, nem vár adatot, visszaküld az összes terem adatait
- listKurzusok
SQL üzenetet küld a szerver fele, nem vár adatot, visszaküld az összes kurzus adatait
- listEpulet
SQL üzenetet küld a szerver fele, nem vár adatot, visszaküld az összes épület adatait
- listOktato
SQL üzenetet küld a szerver fele, nem vár adatot, visszaküld az összes oktató adatait
- getTableNames
SQL üzenetet küld a szerver fele, nem vár adatot, visszaküld az összes tábla nevét ami az adatbázisban van
- getColumnTable
SQL üzenetet küld a szerver fele, egy tábla nevet vár, visszaküld az összes oszlop nevét egy táblában
- insertData
SQL üzenetet küld a szerver fele, egy tábla nevet és egy kulcsot vár, beilleszt adatot egy táblába
- deleteData
SQL üzenetet küld a szerver fele és töröl egy adott adatot a táblából
- changeData
SQL üzenetet küld a szerver fele, egy tábla nevet, egy kulcsot és az új adatokat várja, megváltoztat egy adott adatot a táblában
- searchKey
SQL üzenetet küld a szerver fele, egy tábla nevet és egy kulcsot vár, visszaküld azt a sort amely

megegyezik a várt kulccsal

- dbConn

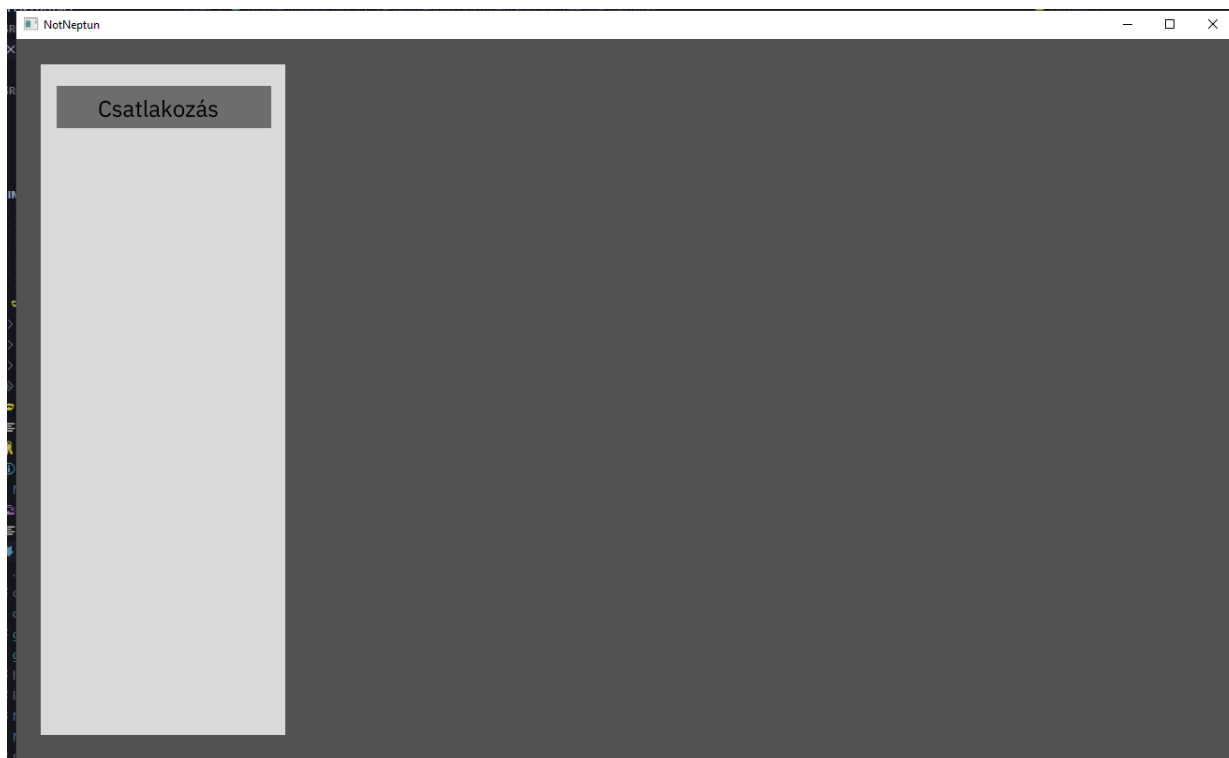
A szerverhez való csatlakozási adatokat állítja be, a csatlakozás adatait várja

Funkciók

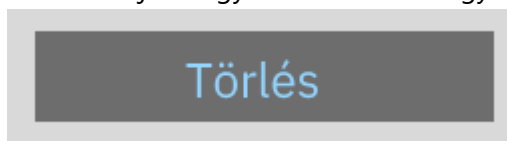
A programban különböző menükön keresztül lehetőségünk van csatlakozni az adatbázishoz, adatokat megnézni és módosítani. Ezekre a képernyőképeket lentebb láthatjuk.

Felhasználói útmutató

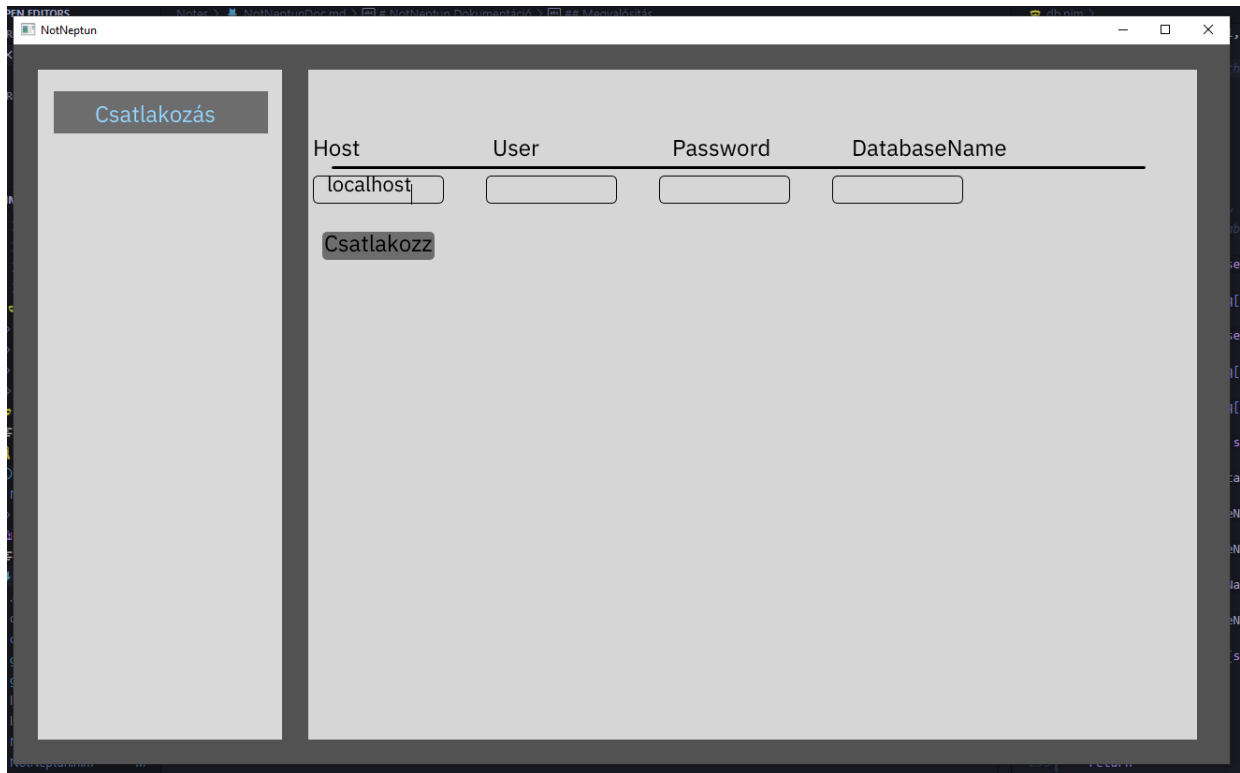
Miután elindítottuk a programot, csakis egy lehetőségünk lesz, kiválasztani az adatbázishoz való csatlakozás menűt:



Onnan tudjuk hogy kiválasztottunk egy menűt, hogy kékre fog váltani a betű színe a gombban.



Ha rámentünk a csatlakozás menűre akkor meg kell adni az adatbázishoz való csatlakozási adatokat. Ezek után a csatlakozás gombra kattintva tovább tudunk lépni. A mezők **nem** lehetnek



Továbblépésnél a főmenüben találjuk magunkat. Itt egy opciót kiválasztva léphetünk tovább különböző részekre.



Ha kiválasztottunk egy menüt akkor meg fog jelenni a hozzá tartozó funkciók, példa az adatok megjelenítésére és a különböző funkciókra:

The screenshot shows the NotNeptun application interface. On the left is a vertical sidebar with buttons: Hallgatók, Oktatók, Kurzusok, Termek, Épületek, Hozzáadás, Módosítás, and Törlés. The 'Hozzáadás' button is highlighted in blue. The main area on the right displays a table with the following data:

| Kód | Vezetéknév | Keresztnév | Szak |
|-------|------------|------------|------|
| TE487 | Teszt | Elek | BTK |

A különböző adat manipulációs mezőknél ki kell választani egy leeső menűn keresztül a megváltoztatni kívánt táblát, fel kell vinni az kért adatokat, majd a gomb megnyomására az adatok változtatásra kerülnek, amiket a listázó menűkben megtekinthetünk.

The screenshot shows the NotNeptun application with the 'Hozzáadás' button selected in the sidebar. The main area displays a form for adding new data. At the top, there are two tabs: 'Táblák' and 'terem', with 'terem' being the active tab. Below the tabs, there are three input fields labeled 'nev', 'ferohely', and 'epuletnev'. A 'Csinál' button is located below the input fields.

NotNeptun

Táblák terem

nev

Csinál

NotNeptun

Táblák terem

nev

Módosít

Adatok:

nev ferohely epuletnev