

Įžanga 'backend'

Užduotį galima atlikti pagal skirtingus sudėtingumo lygius (level 1, level 2, level 3 ir Bonus). Kiekviename lygmenyje gaunami taškai už atliktus reikalavimus. Taškai sumuojami į bendrą balą pagal kurį sprendžiamas tavo stojimas į NFQ Akademiją. Geriausiai atlikusiems užduotį dar bus 30 min pokalbis susipažinimui (telefonu).

Užduotį atlikti ir atsiųsti iki rugsėjo 23-tos vidurnakčio. Siųsti čia - <https://forms.gle/cUpGNWfi991iicidA>

Norint įgyvendinti level 2 ar level 3 reikės įgyvendinti level 1

Backend užduoties kontekstas

Ligoninėse, bankuose, pašte, pasų išdavimo skyriuose ir pan. galima matyti ekranus su skaičiukais. Ateini, gauni lapuką pas pasirinktą specialistą/darbuotoją/langelį ir lauki savo eilės.

Ar kada pagalvojote, kaip būtų faina, jei žinotumėte kiek maždaug dar reikia laukti eilėje ir atitinkamai susiplanuoti savo darbus.

Bonus - papildomi balai

- ☐ Projektas įkeltas į public GitHub (tvarkingai, o ne zip failas) ir duotas veikiantis GitHub URL
- ☐ Stengiamasi darbus skaidyti į logiškas dalis ir juos įkelti kaip atskirus commit į versijų valdymo sistemą
- ☐ Projektas yra live - t.y. patalpintas viešai ir prieinamas, ir duotas veikiantis URL
- ☐ Projekto aprašymas - trumpai pagrįsk, kodėl pasirinkai tokį sprendimą (3 – 5 sakiniai)
- ☐ Nenaudojant framework (kaip Laravel, Symfony ir pan) bus skiriami papildomi taškai
- ☐ Kodas rašomas tvarkingai (naudojamas PSR-2 ar pan. kodo stilius, linkas apačioje)

- ☐ Projektas veikia be fatal, parse error ir pan. notice
- ☐ Panaudotas objektinis programavimas
- ☐ README failas yra

Siūloma (papildoma) literatūra:

GitHub instrukcijos

- <https://www.codecademy.com/courses/learn-git/lessons/git-workflow>
- <https://help.github.com/en/articles/adding-a-file-to-a-repository-using-the-command-line>
- <https://help.github.com/en/articles/adding-an-existing-project-to-github-using-the-command-line>

Svetainės talpinimas į serverį

- <https://scotch.io/@phalconVee/deploying-a-php-and-mysql-web-app-with-heroku>
- <https://www.youtube.com/watch?v=LXb6f8GJ0qs>

Nemokami serveriai patalpinti svetainę

- <https://www.awardspace.com>
- <https://infinityfree.net>
- <https://www.000webhost.com/free-php-hosting>
- <https://www.heroku.com>

README file

- <https://medium.com/@latoyazamill/how-to-create-a-readme-md-file-37cfa2d7ab4>

Kita

- <https://www.php-fig.org/psr/psr-2/>
- https://en.wikipedia.org/wiki/Atomic_commit
- https://en.wikipedia.org/wiki/Object-oriented_programming

Minimalus užduoties įgyvendimas (Level 1)

Turi būti 3 puslapiai:

- ☐ Administravimo puslapis, skirtas įvesti naują klientą į eilę (Klientas turi įvesti bent jau savo vardą)
- ☐ Švieslentės puslapis skirtas rodyti greitai sulauksiančius klientus
- ☐ Specialisto puslapis, kur jis gali pažymėti, kad aptarnavo klientą

Techniniai kriterijai:

- ☐ Yra failas duomenų bazės struktūrai
- ☐ Yra failas prisijungimui prie duomenų bazės (config file)
- ☐ Panaudotas POST HTTP metodas
- ☐ Panaudotas GET HTTP metodas
- ☐ Įrašoma į duomenų bazę
- ☐ Skaitoma iš duomenų bazės (su rikiavimu)
- ☐ Ištrinimas iš duomenų bazės (WHERE sąlyga)
- ☐ Keli būdai atvaizduoti tuos pačius duomenis (LIMIT sąlyga)
- ☐ Panaudotas sudėtingesnė SQL struktūra (kelios lentelės susietos ryšiais (1:daug))

Siūloma literatūra:

- <https://www.w3schools.com/php/>
- https://www.w3schools.com/sql/sql_ref_mysql.asp
- <https://www.w3schools.com/html/>

Rekomenduojamas užduoties įgyvendinimas (Level 2)

Vertinama, kai yra padarytos pirmos dalys

Extra puslapis

- ☐ Lankytojo puslapis, kur jis mato laiką iki savo eilės (nebūtina žiūrėti į švieslentę)

Techniniai kriterijai:

- ☐ Specialistui aptarnavus klientą, vietoj duomenų ištrynimo, pažymima, kad klientas aptarnautas
- ☐ Švieslentėje rodomi tik neaptarnauti klientai
- ☐ Yra funkcija ar SQL užklausa apskaičiuoti, kiek truko apsilankymas (galima apsilankymų laikus saugoti atskiroje lentelėje)
- ☐ Švieslentėje rodoma, kiek laiko liko klientui laukti (vidurkis pagal laukimo laiką per specialistą)
- ☐ Lankytojas gavęs nuorodą (ar įvedęs savo numerį kažkokioje formoje) mato tik jam skirtą laukti laiką
- ☐ Lankytojo puslapyje numatomas laikas patikslinamas kas 5s (JavaScript arba HTML meta)
- ☐ Užregistravus naują klientą turi išvesti Užregistruota sėkmingai arba Įvyko klaida, kreipkitės telefonu

Siūloma (papildoma) literatūra:

- <https://www.w3schools.com/js/>
- <https://www.w3schools.com/css/>

Galimas užduoties praplėtimas (Level 3)

Vertinama, kai yra padarytos pirmos dalys

Extra puslapiai

- ☐ Lankytojo puslapyje pridėti galimybę pavėlinti apsilankymą
- ☐ Puslapis su statistika, kada labiausiai verta ateiti pas specialistą

Techniniai kriterijai:

- ☐ Apsilankymų laikas grupuojamas pagal dienas (SQL)
- ☐ Statistikoje galima uždėti filtrą konkreitiems specialistams
- ☐ Lankytoji sugeneruojama unikali nuoroda į lankytojo puslapį (kad apsaugoti URL atspėjimą)
- ☐ Lankytojas mato mygtuką pavėlinti (sukeičia su už juo esančiu žmogumi)
- ☐ Pavėlinimas negalimas, jei lankytojas yra paskutinis eilėje
- ☐ Lankytojas gali atšaukti susitikimą pas specialistą (PHP + SQL)
- ☐ Įvedamų duomenų validavimas PHP pusėje vardenis(data validation)
- ☐ Po duomenų validavimo PHP pusėje forma turi išlaikyti užpildytą informaciją

Siūloma (papildoma) literatūra:

- <https://dev.mysql.com/doc/>
- <https://www.php.net/manual/en/index.php>
- <https://getcomposer.org/>
- <https://getbootstrap.com/>
- <https://www.w3schools.com/w3css/>
- <https://phpunit.de/>

=====

Atmink tavo geriausias pagalbininkas yra Google.

Sėkmės :)