Testa uzdevums kandidātam

**Uzdevuma būtība:**

Nepieciešams uzprogrammēt vienkāršu testu sistēmu, kurā lietotājs ievada savu vārdu, izvēlās testu, izpilda to, un beigās redz savu rezultātu.

Risinājumā jāizmanto PHP, MySQL, HTML, CSS, JavaScript. PHP kodam jābūt objektorientētam. Dizainiskais risinājums ir katra paša ziņā - var brīvi izpausties.

Tests sastāv no 3 dažādiem skatiem:

1. Sākumlapas - lietotājs ievada savu vārdu un izvēlas kādu no datubāzē esošajiem testiem.
2. Testa jautājumu skats - katram jautājumam ir n atbilžu varianti. Viens no tiem ir pareizs.
3. Rezultāta skats - lietotājs redz savu rezultātu.

**Detalizēts skatu apraksts:**

**1) Sākumlapa**

Lietotājs ievada savu vārdu un izvēlās testu. Ja nav ievadīts vārds vai nav izvēlēts tests, uz nākamo skatu tikt nevar - jāparāda kļūdas paziņojums.

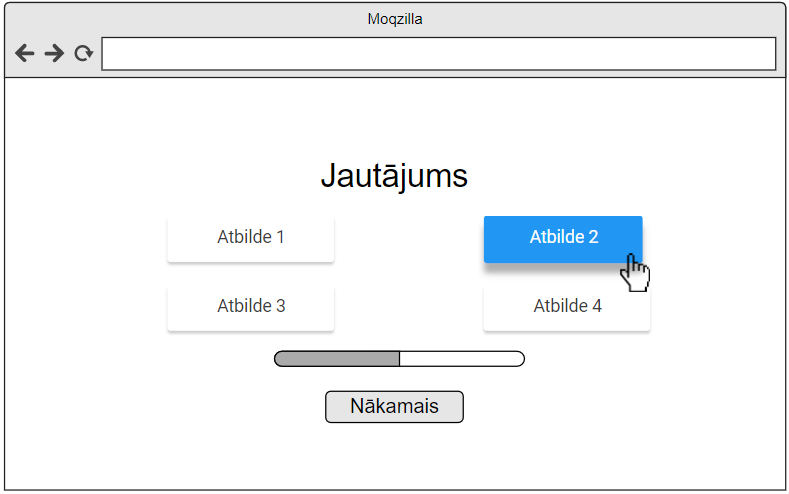


**2) Testa jautājumu skats**

Lietotājs pilda testu. Izvēlas vienu atbildes variantu un dodas uz nākamo jautājumu. Kamēr nav izvēlēts neviens atbildes variants, uz nākamo jautājumu tikt nevar.

Svarīgi: jautājumam var būt mainīgs atbilžu variantu skaits. Vienam jautājumam var būt 2 atbilžu varianti, citam, piemēram, 11.

Atbilstoši jautājuma kārtas numuram, tiek parādīta dinamiska progresa josla (*progressbar*).



**3) Rezultāta skats**

Šajā skatā nepieciešams vēlreiz parādīt lietotāja vārdu, kuru viņš ievadīja sākumlapā, kā arī kopējo rezultātu - cik kopā bija jautājumu un uz cik no tiem lietotājs atbildēja pareizi.



**Tehniskie nosacījumi:**

Dizains

1. Tam ir jābūt responsīvam
   1. Ja ekrāna izmērs ir pietiekami plats, tad atbilžu varianti jārāda 2 kolonnās (kā augstāk redzamajā skicē)
   2. Ja ekrāna izmērs nav tik plats, lai parādītu 2 kolonnas (piemēram, ja lapu skatās ar mobilo telefonu), tad visus atbilžu variantus jāparāda vienā kolonnā vienu zem otra.
2. Izmēri, fonti, krāsas, attālumi - brīva iespēja improvizācijai. Iepriekšējos soļos redzamie vizuālie materiāli ir tikai shematiskas dizaina izkārtojuma skices. Kandidātam pašam jāizvēlas teksta krāsa, fonts, dokumenta fonu krāsa, pogu krāsas utt.

PHP

1. Kodam jābūt objektorientētam.
2. Komentē savu kodu.
3. Nav atļauts izmantot PHP ietvarus (framework). Koda struktūra jāveido pašam.
4. Papildus punkti būs par to, ja katrs nākamais jautājums tiks parādīts dinamiski bez pilnas lapas pārlādes.
5. Jāpievērš uzmanība arī koda formatējumam, stilam un labajai programmēšanas praksei un drošībai.

Datubāze

1. Jāizmanto MySQL.
2. Vienlaicīgi datubāzē var būt vairāki testi.
3. Katram testam ir iespējams neierobežots skaits jautājumu.
4. Katram jautājumam ir 2 - n atbilžu varianti. Respektīvi, maksimālais atbilžu variantu skaits nav stingri definēts. Viens no atbilžu variantiem ir pareizais. Pārējie ir nepareizie.
5. Katru reizi, kad lietotājs atbild uz jautājumu, šis fakts ir jāsaglabā datubāzē - kurš lietotājs, kuram testam, uz kuru jautājumu, kādu atbildi atbildēja.
6. Datubāzē jāsaglabā fināla rezultāts par katru reizi, kad tests tiek pabeigts - ir jāzina, kurš lietotājs kuru testu izpildīja, un uz cik jautājumiem atbildēja pareizi.
7. Jāpieņem, ka datubāze laika gaitā “izaugs” ļoti liela, tādēļ jāsaliek attiecīgie indeksi, lai SQL vaicājumi būt optimāli.

**Nododamais materiāls:**

Jānodod viss kods ar pavadošo dokumentāciju par sagatavošanās soļiem, kuri jāveic uz servera, kurā šis kods tiks testēts (kādi ir konfigurējamie parametri utt.). Datubāzes tabulas jāsaglabā mysql dump failā.

Papildus punkti tiks doti tad, ja uzdevums tiks iesūtīts nevis saarhivētā (piemēram, *.zip*) veidā, bet gan novietots uz kāda GIT repozitorija (piemēram, gitlab.com, github.com utt., absolūti brīva izvēle). Šajā gadījumā ir jāiesūta links uz GIT repozitoriju. Atkarībā no izvēlētā repozitorija - ja tas ir nepieciešams, jānorāda arī pieejas dati, kā tam var piekļūt.