Testa uzdevums kandidātam

**Uzdevuma būtība:**

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

Risinājums jābūvē par pamatu ņemot JavaScript. Realizācija var būt plain JS, tomēr ieteicams izmantot *Vue.js vai React.* Izvēle ir kandidāta ziņā. Dizainiskais risinājums nav limitēts - var brīvi izpausties ar mērķi nodemonstrēt savas zināšanas.

Tests sastāv no 3 dažādiem skatiem:

1. Sākumlapas - lietotājs ievada savu vārdu un izvēlas kādu no testiem.
2. Testa jautājumu skats - katram jautājumam ir vismaz 2 atbilžu varianti (bet maksimālais skaits nav ierobežots). Viens no tiem ir pareizs.
3. Rezultāta skats - lietotājs redz savu rezultātu.

**Detalizēts skatu apraksts:**

**1) Sākumlapa**

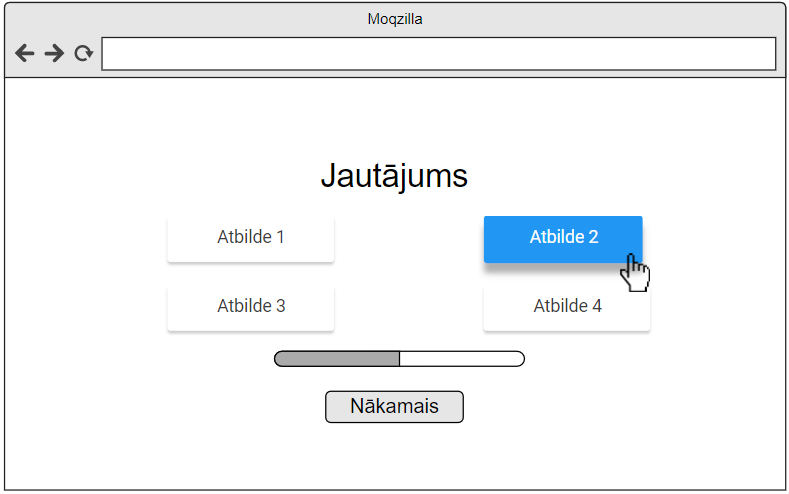
Lietotājs ievada savu vārdu un izvēlās testu. Ja nav ievadīts vārds vai nav izvēlēts tests, uz nākamo skatu tikt nevar - jāparāda kļūdas paziņojums.



**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.

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:**

Frontend

1. Risinājumam jābūt bāzētam uz JavaScript. Ne obligāti, bet ieteicams izmantot *Vue.js* vai *React*.
2. Dizainam 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.
3. 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.

Backend

1. Testiem, jautājumiem un atbilžu variantiem ir jāglabājas backend pusē.
2. Šos datus ir atļauts hardcodēt PHP failos. Respektīvi, datus var glabāt arī datubāzē, bet šis nosacījums nav obligāts.
3. Pēc katras atbildes iesniegšanas, ir jāveic pieprasījums uz backend, kur arī tiek noteikts - vai atbilde bija pareiza, vai nē.

**Nododamais materiāls:**

Jānodod viss kods ar pievienotu readme failu, kur īsumā aprakstīta izvēlētā risinājuma arhitektūra. Ja tiek veidota datubāze, tad tās 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.