Veebiarendaja proovitöö

Eesti keeles

TAUST: SPORDIVÕISTLUSE AJAVÕTUSÜSTEEM

Finišikoridoris on 2 automaatset digitaalset ajavõtu punkti:

1. Finišikoridori sisenemisel

2. Finišijoone ületamisel

Ajavõtu punktist saadetakse reaalajas serverisse järgmine info:

1. Sportlase külge kinnitatud ajavõtu kiibi kood (identifikaator)

2. Ajavõtu punkti identifikaator

3. Kellaaeg sekundi murdosa täpsusega, millal sportlane (antud kiip) selle punkti läbis.

Ürituse eelselt on andmebaasi kantud tabel järgmise infoga üritusel osalevate sportlaste kohta:

1. Antud sportlase külge kinnitatud ajavõtu kiibi kood/identifikaator

2. Sportlase stardinumber

3. Sportlase täisnimi (Eesnimi Perenimi)

ÜLESANNE

Teostada/implementida järgmised asjad:

1. Server/service, mis võtab reaalajas ajavõtu infot vastu. Protokoll ei ole hetkel teada -- mõelda see ise oma suva järgi välja. Testimiseks reaalset ajavõtu süsteemi kasutada ei saa -- teha selle asemel dummy andmeid saatev test-klient

2. Veebi kasutajaliides, kus kuvatakse reaalajas tabeli kujul järjest stardikoridori sisenenud sportlasi järgmiselt:

2.1. Finišikoridori sisenemisel lisatakse kasutajaliideses kuvatavasse tabelisse uus rida, kus on näha sportlase stardinumber ja nimi.

2.2. Finišijoone ületamisel lisatakse sportlase reale tema finišeerumise aeg.

2.3. Kujundada UI selliselt, et viimati stardikoridori tulnud oleksid ekraanil nähtavad ilma, et kasutaja selleks vaeva peaks nägema, vanemad kirjed liiguvad nähtavast osas järgemööda välja.

2.4. Demonstreerida süsteemi toimimist dummy andmeid saatva test-kliendi abil.

2.5 Püüda kasutajaliides disainida selliselt, et kasutaja ei peaks jälgimise ajal millegagi vaeva nägema / mingeid lisaliigutusi tegema (nt. pildi värskendamiseks või scrollimiseks vms. tüütuseks), et ta toimuvast adekvaatselt aru saaks ja segadusse ei satuks.

MITTEKOHUSTUSLIKUD LISAÜLESANDED

3. Jälgida, et finišijoone ületanud oleksid õiges järjekorras kuvatud -- näiteks, kui võistleja A siseneb finišikoridori enne võistlejat B, siis kuvatakse nad koridori sisenemisel ka vastavas järjekorras. Kui aga võistleja B läheb võistlejast A finišikoridoris mööda (ehk ületab finišijoone enne võistlejat A), siis parandada kuvatavat järjekorda vastavalt. Demonstreerida dummy test-kliendiga.

4. Teha nii, et veebi kasutajaliideses toimuks reaalajas suhtlus serveriga ainult siis, kui see brauseri aken esiplaanil/aktiivne on. Kui kasutaja mingi teise akna ette võtab, siis peab veebi kasutajaliides serveriga suhtluse peatama. Kui kasutaja uuesti brauseri akna aktiveerib, peab reaalajas serveriga suhtlemine ja info kuvamine jätkuma.

5. Kui brauseri aken on vahepeal olnud mitteaktiivne ja kasutaja selle uuesti esiplaanile võtab/aktiveerib, siis sõltuvalt tehnilisest lahendusest võib potentsiaalselt olla tekkinud olukord, kus osa infot on nö. "vahelt puudu" (kuna serveriga suhtlust ei toimunud ja vahepealset infot ei saadud). Sellisel juhul mõelda / pakkuda välja, kuidas seda kasutajaliideses käsitleda võiks selliselt, et kasutaja situatsioonist adekvaatselt aru saaks ja segadusse ei satuks.

LAHENDUSE ESITAMINE

1. Backend nõuded:

1.1. Ootame PHP-põhist lahendust.

1.2. Kogu UI tuleb renderdada brauseripoolel -- backendis UI-d renderdada ei tohi, front-endi ja backendi suhtlus peab toimuma üle Ajax JSON REST API.

1.3. PHP -- mingi tuntud raamistiku kasutamine on KOHUSTUSLIK. Tungivalt soovitatav kas Laravel või Slim (mõlemad toetavad JSON REST API arendust).

2. Front-end nõuded:

2.1. JavaScript:

2.1.1. AngularJS -- KOHUSTUSLIK!

2.1.2. jQuery -- KEELATUD!

2.2. CSS:

2.2.1. Bootstrap -- KOHUSTUSLIK!

2.2.2. Vajaliku CSS genereerimiseks Less või Sass preprotsessori kasutamine -- KOHUSTUSLIK!

3. Ülesande lahenduse lähetkood saata kokku pakitud ZIP failina koos juhtnööridega paigaldamiseks ja vaatamiseks/testi käivitamiseks.

Inglise keeles

BACKGROUND: SPORTS EVENT TIMING

There are 2 automatic digital timing points/locations in the finish corridor:

1. Entering into finish corridor

2. Crossing the finish line

From the timing point the following information is sent to our server:

1. The code of the chip that is attached to the athlete/sportsmen (identifier)

2. The identifier of the timing point

3. The (wall)clock time with a precision of a fraction of second, when the athlete (the given chip) passed this timing point.

The database, that is prepared before the sports event, has a table with the following information about the athletes participating the sports event:

1. The code/identifier of the chip that is attached to this athlete

2. The start number of the athlete

3. The full name (First name, Surname) of the athlete

THE TASK

Implement the following software:

1. Server/service, that receives the timing information in real-time. Protocol is not known at the moment -- just design it yourself in the way you wish it. You can't use the real timing system for testing -- create a test-client that sends some dummy data instead.

2. Web user interface, that displays in real-time in table form the athletes who have entered the finish corridor in the following way:

2.1. When athlete enters the finish corridor, a corresponding row is added to the table,

where the athlete's start number and name is displayed.

2.2. When the athlete crosses the finish line, the finish time is added to the athlete's row.

2.3. Design the UI in the way, that the athlete's who entered to the finish corridor last, would be visible to the user without any effort from the user -- the older rows/records just move out of the visible area, sequentially.

2.4. Demonstrate the functioning of the system with the test-client that sends the dummy data.

2.5 Try to design the user interface in the way, that user don't have to put any effort or do any additional moves in order to see something (for example, no need to "refresh" or scroll the page or do any other annoyances), so that he/she would understand adequately what is happening and won't get confused.

NON-MANDATORY ADDITIONAL TASKS

3. Make sure that the athletes who cross the finish-line would be displayed in the correct order -- for example, if athlete A enters the finish corridor before athlete B, then they are displayed in the order of entering the corridor. But if athlete B passes the athlete A in the finish corridor (i.e. the athlete B crosses the finish line before the athlete A), then adjust the displayed order accordingly. Demonstrate it with the dummy test-client.

4. Do so that the web user interface would interact with the server in real-time only, when the browser window is in foreground / active. If user brings some other application window into foreground, then the web user interface has to stop the communication with the server. I the user activates the web browser window again, the real-time communication with the server must be resumed.

5. If the browser window has been deactivated meantime and the user brings it to foreground again, then, depending on the technical solution, there might be situation where there is a "gap" in the information that has been received from the server (because the communication with the server didn't happen and the information was not sent). In that case, think / propose, how it could be handled in the user interface so, that user would understand it adequately and won't get confused.

SOLUTION

1. Backend requirements:

1.1. We expect a PHP-based solution.

1.2. The entire UI must be rendered in the browser -- rendering UI on the back-end is forbidden, front-end and backend communication must happen over Ajax JSON

REST API.

1.3. PHP -- using a well-known framework is MANDATORY. It is strongly recommended to use either Laravel or Slim (both support the development of JSON REST API).

2. Front-end requirements:

2.1. JavaScript:

2.1.1. AngularJS -- MANDATORY!

2.1.2. jQuery -- FORBIDDEN!

2.2. CSS:

2.2.1. Bootstrap -- MANDATORY!

2.2.2. Using Less or Sass preprocessor for generating the needed CSS -- MANDATORY!

3. Send us the solution as a ZIP file that contains source code together with guidelines for setting it up and observing / running the test.