

Allpool on kolm ülesannet, iga ülesande jaoks on ette nähtud 1,5h, ehk maksimaalselt **kokku 4,5 tundi**. Soovitus on alustada ülesannetest, mis tunduvad sulle lihtsamad! Tagasi saada töötavad failid, mida saab kohe testida. Failid nimeta stiilis PHP\_1.1\_Perenimi. Kui midagi jääb segaseks või ei jõua valmis, siis palun helista (vastasel juhul ei pruugi garanteerida piisavalt kiiret reageerimist).

Arvesta, et esimene proovitöö on erinevate oskustega kandidaatidele sama ja eesmärgiks on saada ülevaade hetkeolukorrast - ära heitu kui see tundub liiga raske ega pettu kui liiga lihtne.

Vastus saada: leili.mard@lumav.com etteantud tähtajaks! Hilinenud vastused loetakse mitteläbinuks!

Edu!

---

## 1. PHP ja XML

On kaks XML struktuuri teksti kujul (struktuurides on N rida, read on järjestatud kasvavalt):

On kaks XML struktuuri teksti kujul (struktuurides on N rida, read on järjestatud kasvavalt):

A.     <data>  
          <row><value>1</value></row>  
          <row><value>2</value></row>  
          ...  
          <row><value>N</value></row>  
      </data>

B.     <data>  
          <row><attr>X1</attr></row>  
          <row><attr>X2</attr></row>  
          ...  
          <row><attr>XN</attr></row>  
      </data>

1.1 . Kirjuta PHP funktsioon, mis ühendaks kaks XML struktuuri nii, et tulemus jääks kujul:

```
<data>  
  <row><value>1</value><attr>X1</attr></row>  
  <row><value>2</value><attr>X2</attr></row>  
  ...  
  <row><value>N</value><attr>XN</attr></row>  
</data>  
function do1 ($a, $b) {  
  ...
```

```

        return $c;
    }

```

**1.2.** Kirjuta PHP funktsioon, mis ühendaks kaks XML struktuuri nii, et tulemus jääks kujul:

```

    <data>
        <row attr="X1">1</row>
        <row attr="X2">2</row>
        ...
        <row attr="XN">N</row>
    </data>
function do2 ($a, $b) {
    ...
    return $c;
}

```

**1.3.** Kirjuta PHP funktsioon, mis ühendaks kaks XML struktuuri nii, et tulemus jääks massiivina kujul:

```

    array(N) { [X1]=> int(1) [X2]=> int(2) ... [XN]=> int(N) }

function do3 ($a, $b) {
    ...
    return $c;
}

```

## 2. SQL (MySQL näitel)

On kaks tabelit:

A. user

id (int) (varchar)	name
1	Kasutaja1
2	Kasutaja2
3	Kasutaja3

b. task

id (int)	user_id (int)	content (text)
1	1	Task1
2	2	Task2
3	2	Task3
4	2	Task4
5	3	Task5

**2.1.** Kirjuta SQL laused user ja task tabelite struktuuri loomiseks. Arvestada, et "id" väljad on automaatselt kasvavad ja user\_id on välisvõti. Lisa vajalikud indeksid. Täida tabelid Andmetega.

**2.2.** Kirjuta SQL lause, mis mõlemale juba olemasolevale tabelile lisab juurde veerud, mis salvestaks automaatselt kirjade lisamise ja muutmise ajad.

**2.3.** Kirjuta SQL lause, mis tagastaks kahe tabeli (user, task) andmed grupeeritult järgmisel kujul (tulemuses peab olema sama palju ridu kui on erinevaid kasutajaid user tabelis ja kõik kasutajatega seotud ülesanded välja toodud):

user_name	user_tasks
Kasutaja1	Task1
Kasutaja2	Task2, Task3, Task4
Kasutaja3	Task5

## 3. JS ja jQuery

**3.1.** Kirjuta Javascripti funktsioon ImageClick(image), mis jätab pildile klikkides pildi kursori külge, st liiguks kursori liigutamisega kaasa. Teine klikk pildil viiks pildi tagasi algsesse asendisse. Kasuta ainult Javascripti (mitte jQuery't).

```

```

**3.2.** Kirjuta jQuery kood, mis peale lehe laadimist teeb html-is kõik pildid klikitavaks nii, et pilt avaneks eraldi aknas. Jäta vahele pildid, mis on juba lingitud (topelt linke ei tohi tekkida).

**3.3.** Kirjuta jQuery funktsioon, mis teeb AJAX päringu etteantud urlile, success funktsioon peab serverist vastuvõetud html koodist kuvama elemendi id-ga "content" sisu elemendi id- ga "ajax-content" sisse.