

HTML5

Detyra 1:

- a) Shkruani kodin në HTML i cili implementon ueb formën e cila përmban këto ueb kontrollat: Numrin e indeksit (tekst), emrin e studentit (tekst), mbiemrin (tekst), datëlindjen (tekst), gjininë (radio-butona për të zgjedhur gjininë 'F' ose 'M'), email-in (tekst), departamentin (listë drop-down me departamentet e fakultetit), aftësitë profesionale (checkbox-a), komenti (textarea) butonin për aprovimin e të dhënave (submit) dhe butonin për pastrimin e të dhënave (reset).
Me anë të funksionit ValidoFusha në JavaScript-sintaksë të validohen fushat: Numri i indeksit, emri dhe mbiemri. Pra, me ndihmën e këtij funksioni duhet të mundësohet që nëse ndonjëra nga fushat e më sipërme nuk plotësohen me të dhëna, të paraqitet mesazhi: "Plotësoni fushat e kërkuara!", dhe fushat e tilla të ndryshojnë ngjyrën në "salmon" ose sipas dëshirës.
- b) Në formën e implementuar nga detyra paraprake, tek fusha 'Ditëlindja' të përdoret kontrolla 'date' për zgjedhjen e datës (date picker) nga HTML5.
- c) Duke përdorur input type 'email' nga HTML5, të bëhet validimi i email-it në formën e krijuar nga detyra 'a'.
- d) Në formën e implementuar nga detyra 'a' të shtohet fusha për regjistrimin e url adresës së studentit, duke përdorur 'url' tipin nga HTML5 që validon automatikisht url adresën e dhënë.
- e) Në formën e implementuar nga detyra 'a' të shtohet fusha për regjistrimin e notës mesatare të studentit. Për këtë fushë të përdoret tipi 'number' nga HTML5. Në këtë fushë të lehet mundësia e paraqitjes së vlerave në mes 6.0 (min=6) dhe dhe 10.0 (max=10). Hapi për rritje ose zbritje të jetë 0.1 (step=0.1), ndërsa default vlera e kësaj fushe të jetë 7.5 (value=7.5).
- f) Në formën e implementuar nga detyra 'a' të shtohet fusha për zgjedhjen e ngjyrës së preferuar të studentit, duke përdorur 'color' tipin (color picker) nga HTML5.

Zgjidhja për (a):

Kodi:

```
<html>
  <head>
    <title>Regjistrimi i studenteve...</title>
    <script type="text/javascript">
      function ValidoFushat()
      {
        try
        {
          var nr_indeksit =
document.forms["frmStudent"]["nrIndeksit"].value;
          var emri = document.forms["frmStudent"]["Emri"].value;
          var mbiemri =
document.forms["frmStudent"]["Mbiemri"].value;
          var validimi = true;
          if(nr_indeksit==null || nr_indeksit=="")
          {
            validimi = false;
```

```

document.forms["frmStudent"]["nrIndeksit"].style.backgroundColor = "salmon";
    }
    if(emri==null || emri=="")
    {
        validimi = false;
    }

document.forms["frmStudent"]["Emri"].style.backgroundColor = "salmon";
    }
    if(mbiemri==null || mbiemri=="")
    {
        validimi = false;
    }

document.forms["frmStudent"]["Mbiemri"].style.backgroundColor = "salmon";
    }
    if(!validimi)
    {
        alert("Plotësoni fushat e kërkuara!");
    }
    return validimi;
}
catch(err)
{
    alert(err.message);
    return false;
}
}
</script>
</head>
<body>
    <form name="frmStudent" action="index.php" method="post"
onsubmit="return ValidoFushat()">
        <table align="center">
            <tr><br/>
                <td>Nr. Indeksit:</td>
                <td><input name="nrIndeksit" type="text" size="10"
maxlength="10"/></td>
            </tr>
            <tr>
                <td>Emri:</td>
                <td><input name="Emri" type="text" size="30"
maxlength="30"/></td>
            </tr>
            <tr>
                <td>Mbiemri:</td>
                <td><input name="Mbiemri" type="text" size="30"
maxlength="30"/></td>
            </tr>
            <tr>
                <td>Ditë e lindja:</td>
                <td><input name="Ditelindja" type="text"/></td>
            </tr>
        </table>
    </form>
</body>
</html>

```

```

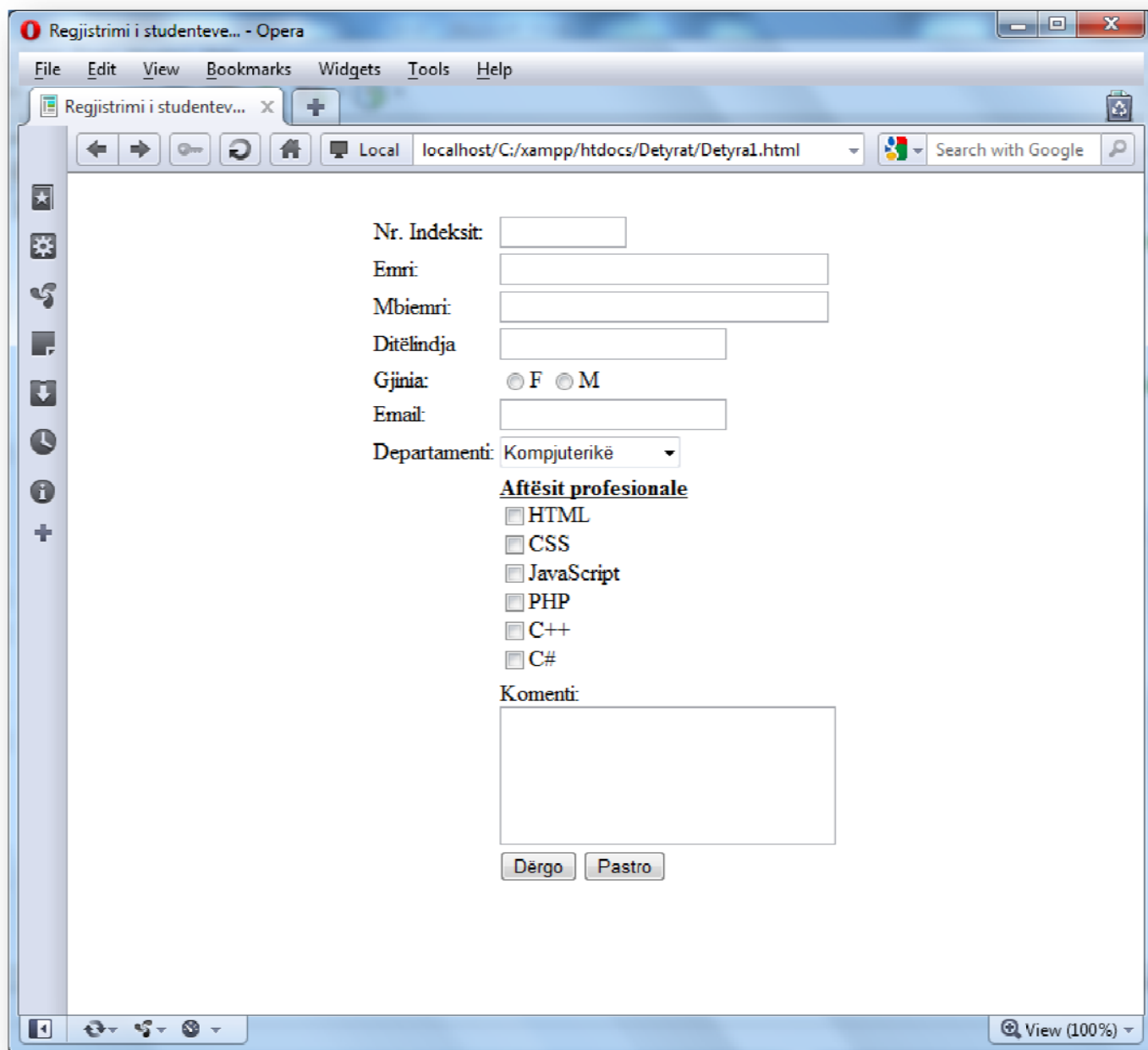
        </tr>
        <tr>
            <td>
                Gjinia:
            </td>
            <td>
                <input type="radio" name="Gjinia" value="F"/>F
                <input type="radio" name="Gjinia" value="M"/>M
            </td>
        </tr>
        <tr>
            <td>Email:</td>
            <td><input name="Email" type="text"/></td>
        </tr>
        <tr>
            <td>Departamenti:</td>
            <td>
                <select name="Departamenti">
                    <option value="K">Kompjuterik&euml;</option>
                    <option value="T">Telekomunikacion</option>
                    <option value="A">Automatik&euml;</option>
                    <option value="E">Elektronik&euml;</option>
                </select>
            </td>
        </tr>
        <tr>
            <td></td>
            <td>
                <b><u> Aft&euml;sit profesionale</u></b><br/>

                <label><input name="HTML" type="checkbox">HTML</label><br/>
                <label><input name="CSS" type="checkbox">CSS</label><br/>
                <label><input name="JavaScript"
type="checkbox">JavaScript</label><br/>
                <label><input name="PHP" type="checkbox">PHP</label><br/>
                <label><input name="Cpp" type="checkbox">C++</label><br/>
                <label><input name="Csharp" type="checkbox">C#</label>
            </td>
        </tr>
        <tr>
            <td></td>
            <td>Komenti:<br/>
                <textarea name="komenti" cols="30" rows="6"></textarea>
            </td>
        </tr>
        <tr>
            <td>
            </td>
            <td>
                <input name="btnDergo" type="submit" value="D&euml;rgo"/>
                <input name="btnPastro" type="reset" value="Pastro" />
            </td>
        </tr>
    </table>

```

```
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

Ekzekutimi (në browser Opera, version 11.01):



The screenshot shows the Opera browser window titled "Regjistrimi i studenteve... - Opera". The address bar shows the local file path: "localhost/C:/xampp/htdocs/Detyrat/Detyra1.html". The form contains the following fields and options:

- Nr. Indeksit:
- Emri:
- Mbiemri:
- Ditëlindja:
- Gjinia: ☐ F ☐ M
- Email:
- Departamenti:
- Aftësit profesionale**
 - ☐ HTML
 - ☐ CSS
 - ☐ JavaScript
 - ☐ PHP
 - ☐ C++
 - ☐ C#
- Komenti:

At the bottom of the form are two buttons: "Dërgo" and "Pastro". The status bar at the bottom right shows "View (100%)".

Zgjidhja për (b):

Për të shfaqur kalendarin tek fusha 'Ditëlindja', zëvendësojmë kodin:

```
<input name="Ditelindja" type="text"/>
```

Me këtë kod:

```
<input name="Ditelindja" type="date"/>
```

Ekzekutimi (në browser Opera, version 11.01):

Nr. Indeksit:

Emri:

Mbiemri:

Ditëlindja: 2011-02-23 ▼

Gjinia:

Departamenti:

Mon	Tue	Wed	Thu	Fri	Sat	Sun
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6
7	8	9	10	11	12	13

Today

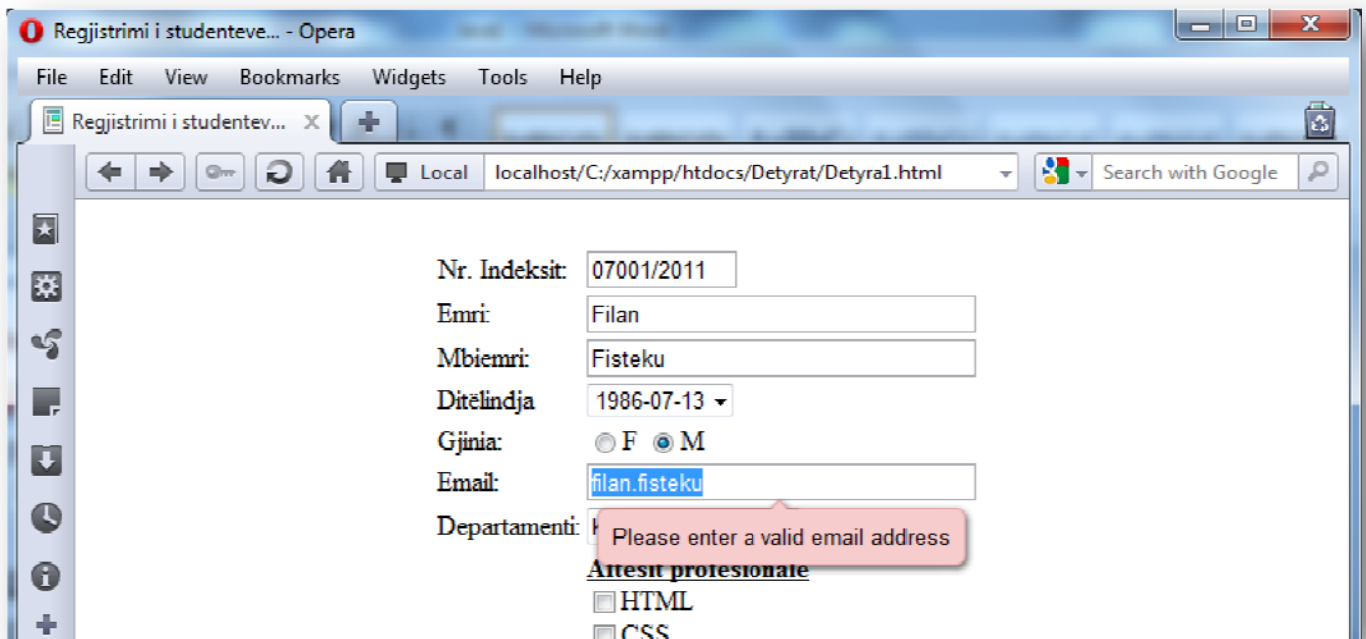
Zgjidhja për (c):

Për të validuar automatikisht email-in e studentit duke përdorur tipin 'email' nga HTML5, zëvendësojmë këtë kod nga detyra 'a':

```
<input name="Email" type="text"/>
```

Me këtë kod:

```
<input name="Email" type="email"/>
```

Ekzekutimi (në browser Opera, version 11.01):**Zgjidhja për (d):**

Për të shtuar fushën për regjistrimin e url-së së studentit duke përdorur kontrollen 'url' nga HTML5, në formën e koduar nga detyra 'a' shtojmë këtë pjesë të kodit:

```
Url-ja e studentit:
<input name="Url_e_studentit" type="url"/>
```

Zgjidhja për (e):

Për të shtuar fushën për regjistrimin e notës mesatare të studentit duke përdorur kontrollen 'number' nga HTML5, në formën e koduar nga detyra 'a' shtojmë këtë pjesë të kodit:

```
Nota mesatare:
<input name="Nota_Mesatare" type="number" min="6" max="10" step="0.1"
value="7.5"/>
```

Ekzekutimi (në browser Opera, version 11.01):

Nota mesatare:

Nëse shkruajmë numër më të vogël ose më madh:

Nota mesatare:

Ngjyra:

Gjinia:

Please enter a number greater than or equal to 6

Nota mesatare:

Ngjyra:

Gjinia:

Please enter a number less than or equal to 10

Zgjidhja për (f):

Për të shtuar fushën për zgjedhjen e ngjyrës së preferuar të studentit duke përdorur kontrollen 'color' nga HTML5, në formën e koduar nga detyra 'a' shtojmë këtë pjesë të kodit:

Ngjyra e preferuar:

```
<input name="Ngjyra_e_preferuar" type="color"/>
```

Ekzekutimi (në browser Opera, version 11.01):

Ngjyra e preferuar:

#22b14c

Other...

Detyra 2:

Ne këtë ushtrim është treguar struktura e një HTML5 dokumenti:

```
<!DOCTYPE html>
```

```
<html>
  <head><title>Fiek</title></head>
<body>
  <!--
    Tagu <header> specifikon një hyrje, ose një grup elementesh navigimi
    për dokument.
  -->
  <header>
    <h1>Programimi në Internet</h1>
  </header>
  <!--
    Tagu <nav> definon një seksion (paragraf) të navigimit.
  -->
  <nav>
    <ul>
      <li><a href="/html5">HTML5</a></li>
      <li><a href="/xml">XML</a></li>
      <li><a href="/rss">RSS</a></li>
      <li><a href="/php">PHP</a></li>
      <li><a href="/ajax">AJAX</a></li>
    </ul>
  </nav>
  <!--
    Tagu <article> specifikon pavarësinë, vetë përmbajtjen. Një artikull
    duhet të ketë kuptim në vetë e tij dhe duhet të ketë mundësin të
    shpërndaj atë pavarësisht pjesës tjetër të sajtit.
  -->
  <article>
    <!--
      Tagu <section> definon sektorët (paragrafet) në një dokument. Si
      kapitujt, hederat, futulerat, ose ndonjë sektorët tjetër të
      dokumentit.
    -->
    <section>
      <h1>HTML5</h1>
      <p>HTML5 &euml;sht&euml; gjenerata e ardhshme e HTML...</p>
    </section>
  </article>
  <!--
    Tagu <aside> definon një pjesë të përmbajtjes nga përmbajtja në të
    cilin është i vendosur. Përmbajtja anash duhet të jetë e lidhur me
    përmbajtjen më të afërt.
  -->
  <aside>
    <h1>Arkivi</h1>
    <ul>
      <li><a href="/2009/">2009</a></li>
      <li><a href="/2010/">2010</a></li></ul>
  </aside>
</body>
```



```
</html>
```

Detyra 3:

Ne këtë ushtrim është treguar se si përdoret atributi placeholder nga HTML5:

```
<form>
  <input name="q" type="search" placeholder="Kërko...">
  <input type="submit" value="Kërko">
</form>
```

Ekzekutimi (në browser Opera, version 11.01):**Detyra 4:**

Ne këtë ushtrim është treguar se si përdoret elementi video në HTML5:

```
<video src="video1.ogg" width="320" height="240" controls="controls">
  Browser-i juaj nuk e përkrah tag-un video.
</video>
```

- Atributi 'controls' është përdorur për të shtuar kontrollat: play, paues, dhe volume.
- Teksti në mes tagjeve <video> dhe </video> paraqitet nëse browser-i nuk e përkrah elementin video.
- Atributet 'width' dhe 'height' janë përdorur për të caktuar dimensionet (gjerësinë dhe lartësinë) e videos.

Ekzekutimi (në browser Opera, version 11.01):



Gjithashtu elementi video lejon shumë source elemente, që mund të linkojnë video fajlla të ndryshëm. Browser-i do të përdor formatin e parë që e përkrahë:

```
<video width="320" height="240" controls="controls">
  <source src="video1.ogg" type="video/ogg" />
  <source src="video1.mp4" type="video/mp4" />
  <source src="video1.webm" type="video/webm" />
  Browser-i juaj nuk e përkrah tag-un video.
</video>
```

Disa attribute shtesë të elementit video:

- autoplay – nëse ky atribut është prezent, video do të filloj posa është gati.
- loop – nëse ky atribut është prezent, video do të startohet përsëri çdo herë kur të përfundoj.
- preload – nëse ky atribut është prezent, video do të ngarkohet (load) përgjatë ngarkimit të faqës.

Detyra 5:

Ne këtë ushtrim është treguar se si përdoret elementi audio në HTML5:

```
<audio src="audio1.ogg" controls="controls">
</audio>
```

Ekzekutimi (në browser Opera, version 11.01):



Disa attribute shtesë të elementit audio:

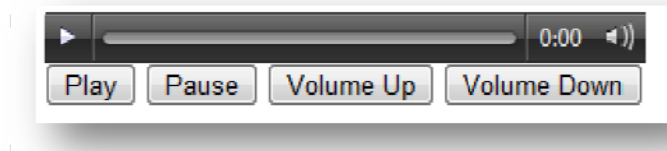
- autoplay – nëse ky atribut është prezent, audio do të filloj posa është gati.
- loop – nëse ky atribut është prezent, audio do të startohet përsëri çdo herë kur të përfundoj.
- preload – nëse ky atribut është prezent, audio do të ngarkohet (load) përgjatë ngarkimit të faqës.
- controls – përdoret për të shtuar kontrollat: play, pauses, dhe volume.

Detyra 6:

Ne këtë ushtrim është treguar se si mund të shtohen butonat për kontrollen e audio në HTML5:

```
<audio id="player" src="audio1.ogg" controls>
</audio>
<div>
  <button onclick="document.getElementById('player').play()">
    Play
  </button>
  <button onclick="document.getElementById('player').pause()">
    Pause
  </button>
  <button onclick="document.getElementById('player').volume += 0.1">
    Volume Up
  </button>
  <button onclick="document.getElementById('player').volume -= 0.1">
    Volume Down
  </button>
</div>
```

Ekzekutimi (në browser Opera, version 11.01):



XML & DTD

Detyra 1:

Informatat në vijim të paraqiten në XML:

Një person me emrin "Filan Fisteku" dhe datëlindje 13.07.1986.

```
<?xml version="1.0" encoding="iso-8859-1"?>
  <person>
    <emri>Filan</emri>
    <mbiemri>Fisteku</mbiemri>
    <datelindja>
      <dita>13</dita>
      <muaji>07</muaji>
      <viti>1986</viti>
    </datelindja>
  </person>
```

Detyra 2:

- Të shqyrtohet dokumenti XML i dhënë më poshtë dhe XML pema përkatëse e tij.
- Për dokumentin XML të dhënë me poshtë, shkruani **DTD** (Document Type Definition) përkatëse.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<librat>
  <libri isbn="1-904811-82-5">
    <titulli>
      Timoni nga Athina
    </titulli>
    < autori viti_lindjes="1564">
      <emri>William</emri>
      <mbiemri>Shakespeare</mbiemri>
    </ autori>
    <viti>
      1607
    </viti>
  </libri>
  <libri isbn="1-59059-392-8">
    <titulli>
      Lahuta e Malsis
    </titulli>
    < autori viti_lindjes="1871">
      <emri>Gjergj</emri>
      <mbiemri>Fishta</mbiemri>
    </ autori>
  </libri>
</librat>
```

```

        </autori>
    </libri>
</librat>

```

Detyra 3:

Për dokumentin e dhënë "email.xml" të shkruhet DTD-ja:

```

<?xml version="1.0" encoding="UTF-16"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
  <head>
    <from name="Besmir Sejdiu" address="besmir.sejdiu@uni-pr.edu"/>
    <to name="Emri Mbiemri" address="shembull@shembull.com"/>
    <subject>Përshëndetje</subject>
  </head>
  <body>
    <text>Text, ...</text>
  </body>
</email>

```

DTD-ja:

```

email.dtd
<?xml version="1.0" encoding="UTF-16"?>
<!ELEMENT email (head, body) >
<!ELEMENT head (from, to+,cc*, subject)>
<!ELEMENT from EMPTY>
<!ATTLIST from
  name CDATA #IMPLIED
  address CDATA #REQUIRED>
<!ELEMENT to EMPTY>
<!ATTLIST to
  name CDATA #IMPLIED
  address CDATA #REQUIRED>
<!ELEMENT cc EMPTY>
<!ATTLIST cc
  name CDATA #IMPLIED
  address CDATA #REQUIRED>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT body (text, attachment*)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT attachment EMPTY>

```

```
<!ATTLIST attachment
encoding (mime|binhex) "mime"
file CDATA #REQUIRED>
```

RSS (Really Simple Syndication)

Historia e RSS?

Pse të përdoret RSS?

Kush do të përdor RSS?

E ardhmja e RSS?

Avantazhet e RSS?

Versionet e RSS?

Sintaksa e RSS?

RSS Publish Feed?

A mund të menaxhoni RSS feed-in e juaj vet?

RSS automatik?

Lexuesit RSS?

Detyra 1:

Ne këtë ushtrim është treguar se si përdoret RSS:

```
//deklarimi XML-tregon versionin dhe enkodimin e karaktereve
<?xml version="1.0" encoding="ISO-8859-1" ?>
//deklarimi i RSS dokumentit - versioni 2.0
<rss version="2.0">
//deklarimi i channel elementit - është përdorur për të
//përshkruar furnizuesin RSS (feed) -ka tri elemente fëmijë të rekomanduara
<channel>
//definon titullin e kanalit
<title>Programimi ne internet- Home Page</title>
//definon hyperlink-un e kanalit
<link>http://www.fiek.ProgInternet.com</link>
```

```
//Përshkruan kanalin
<description>Universiteti i Prishtinës</description>

//Secili channel element mund te kete nje ose me shume item element
//Secili item element definon nje artikull ose "stoje" in RSS feed
<item>
  //Definon titullin e item-it
  <title>RSS shembuj</title>
  //Definon hyperlink-un e item-it
  <link>http://www.fiek.ProgInternet.com/rss</link>
  <description>Te rejat e RSS ne fiek-ProgInternet</description>
</item>
<item>
  <title>XML shembuj</title>
  <link>http://www.fiek.ProgInternet.com/xml</link>
  <description>Te rejat e XML ne fiek-ProgInternet</description>
</item>
</channel>
</rss>
```

- Elementi **<channel>** përveç nën elementeve të domosdoshme ka edhe disa nën elemente opcionale:
 - **<category>** - përdoret për të specifikuar kategorinë e furnizimit (feed). Në bazë të këtij elementi RSS aggregator-et grupojnë sajtet bazuar në këtë kategori.
p.sh: **<category>Web development</category>**
 - **<copyright>2011 Të gjitha të drejtat e autorit janë të rezervuara</copyright>**
 - **<image>** - shfaq imazhin kur aggregator-et prezantojnë feed, p.sh:


```
<image>
  <url>http://www.fiek.ProgInter.com/images/logo.gif</url>
  <title>Fiek.ProgInter.com </title>
  <link>http://www.fiek.ProgInter.com</link>
</image>
```
- Elementi **<item>** përveç nën elementeve të domosdoshme mund të ketë edhe disa nën elemente opcionale:
 - **<author>** - është përdorur për të specifikuar e-mail adresën e autorit për atë 'artikull'
p.sh: **<author>autori@atuori.test</author>**
 - **<comments>** - mundëson që një artikull të linkohet të një koment rreth atij artikulli
p.sh: **<comments>http://www.fiek.ProgInter/comments</comments>**
 - **<enclosure>** - lejon një media-fajll të përfshihet me një artikull. Ky element ka tri attribute të domosdoshme:
 - url – definon URL-në e media fajllit
 - length – definon length (në bajta) e medja fajllit
 - type – definon tipin e media fajllit
 p.sh: **<enclosure url="http://www.fiek.ProgInter/rss/rss.mp3" length="5000" type="audio/mpeg" />**

Konceptet themelore në PHP

Cka është PHP?

PHP – është gjuhë skriptuese e cila ekzekutohet në anën e serverit (server-side)

PHP – qëndron për **PHP: Hypertext Preprocessor**

PHP – është skriptuese nga ana e serverit, sikurse ASP

PHP- përkrahin shumë sisteme të menaxhimit të bazave të të dhënave (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etj.)

PHP-është një softuer open source

PHP-është pa para për tu shkarkuar dhe për tu përdorur

Fajllat PHP?

PHP fajllat mund të përmbajnë tekst, HTML tag-e dhe skripta

PHP fajllat nga browser-et shfaqen si HTML

PHP fajllat përfundojnë me **“.php”**, **“.php3”**, ose **“.phtml”**

Pse PHP?

PHP ekzekutohet në platforma të ndryshme (Windows, Linux, Unix, etj.)

PHP është kompatibil pothuajse me të gjithë serverët që përdoren sot (Apache, IIS, etj.)

PHP është ‘free’ për tu shkarkuar (<http://www.php.net>)

PHP është e lehtë për tu mësuar dhe ekzekutuar në anën e serverit.

Avantazhet e PHP?

Disa nga konkurrentet kryesore të PHP-së janë: **Perl**, **Microsoft ASP.NET**, **JavaServer Pages (JSP)**, dhe **ColdFusion**.

Në krahasim me këto produkte PHP-ja ka shumë të mira përfshirë:

- Performancë të lartë
- Lidhje me sisteme të ndryshme për menaxhimin e bazave të të dhënave
- Çmimi i ulët
- E lehtë për tu mësuar dhe shfrytëzuar
- Përkrah orientimin në objekte
- Transportueshmëri
- Disponueshmeria e source kodit
- Disponueshmeria e përkrahjes

Detyra 1:

Ne këtë ushtrim janë treguar tri stilet e përdorimit të sintaksës së PHP-së:

a) XML style

```
<html>
  <head>
    <title>PHP</title>
  </head>
  <body>
    <?php
      echo 'Mirësevini në PHP!';
    ?>
  </body>
</html>
```

b) Short style

```
<html>
  <head>
    <title>PHP</title>
  </head>
  <body>
    <?
      echo 'Mirësevini në PHP!';
    ?>
  </body>
</html>
```

c) Script style

```
<html>
  <head>
    <title>PHP</title>
  </head>
  <body>
    <SCRIPT LANGUAGE="php">
      echo "Mirësevini në PHP!";
    </SCRIPT>
  </body>
</html>
```

Detyra 2:

Ne këtë ushtrim është treguar përdorimi i komenteve në PHP:

```
<html>
  <head>
    <title>PHP Komentet</title>
  </head>
  <body>
    <?php
      //Ky është një koment brenda një rreshti
      #Ky është një koment brenda një rreshti
      /*
        Ky
        është
        një koment
        brenda bllokut
      */
      echo "Mirësevini në PHP!";
    ?>
  </body>
</html>
```

Variablat në PHP

Disa tipe të variablave në PHP:

- **Integer**—Used for whole numbers
- **Float** (also called **double**)—Used for real numbers
- **String**—Used for strings of characters
- **Boolean**—Used for true or false values
- **Array**—Used to store multiple data items
- **Object**—Used for storing instances of classes

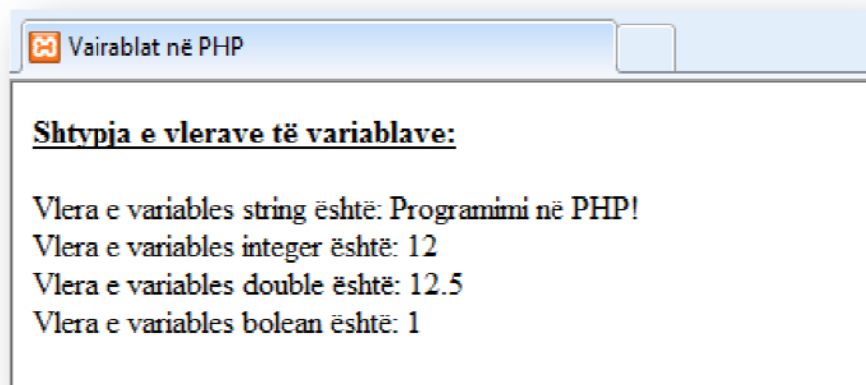
Detyra 3:

Ne këtë ushtrim është treguar përdorimi i variablave në PHP:

```
<html>
  <head><title>Vairablat në PHP</title></head>
  <body>
    <?php
      $var_String = "Programimi në PHP!"; //variabël e tipit string
      $var_Ineger = 12;                  //variabël e tipit integer
      $var_Double = 12.5;
      $var_Boolean = true;

    ?>
    <p><b><u>Shtypja e vlerave të variablave:</u></b></p>
    Vlera e variables string është: <?php echo($var_String) ?> <br/>
    Vlera e variables integer është: <?php echo($var_Ineger) ?> <br/>
    Vlera e variables double është: <?php echo($var_Double) ?> <br/>
    Vlera e variables boolean është: <?php echo($var_Boolean)?> <br/>
  </body>
</html>
```

Ekzekutimi (në browser Internet Explorer 8, version 8.0.7600.16385):



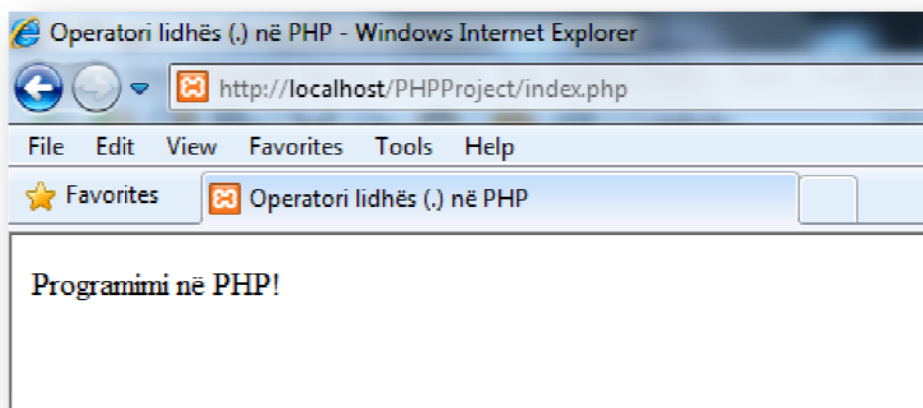
Operatori lidhës (.)

Detyra 4:

Ne këtë ushtrim është treguar përdorimi i operatorit pikë (.) në PHP:

```
<html>
  <head><title>Operatori lidhës (.) në PHP</title></head>
  <body>
    <?php
      $txt1="Programimi në";
      $txt2="PHP!";
      echo $txt1 . " " . $txt2;
    ?>
  </body>
</html>
```

Ekzekutimi (në browser Internet Explorer 8, version 8.0.7600.16385):



Operatorët në PHP**Operatorët aritmetikë**

Operatori	Përshkrimi	Shembull	Rezultati
+	Addition	$x=2$ $x+2$	4
-	Subtraction	$x=2$ $5-x$	3
*	Multiplication	$x=4$ $x*5$	20
/	Division	$15/5$ $5/2$	3 2.5
%	Modulus (division remainder)	$5\%2$ $10\%8$ $10\%2$	1 2 0
++	Increment	$x=5$ $x++$	$x=6$
--	Decrement	$x=5$ $x--$	$x=4$

Operatorët assignment

Operatori	Shembull	Është e njëjtë si
=	$x=y$	$x=y$
+=	$x+=y$	$x=x+y$
-=	$x-=y$	$x=x-y$
=	$x=y$	$x=x*y$
/=	$x/=y$	$x=x/y$
.=	$x.=y$	$x=x.y$
%=	$x\%=y$	$x=x\%y$

Operatorët relacionale

Operatori	Përshkrimi	Shembull
==	is equal to	$5==8$ returns false
!=	is not equal	$5!=8$ returns true
<>	is not equal	$5<>8$ returns true
>	is greater than	$5>8$ returns false
<	is less than	$5<8$ returns true
>=	is greater than or equal to	$5>=8$ returns false
<=	is less than or equal to	$5<=8$ returns true

Operatorët logjikë

Operatori	Përshkrimi	Shembull
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
	or	x=6 y=3 (x==5 y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

Detyra 5:

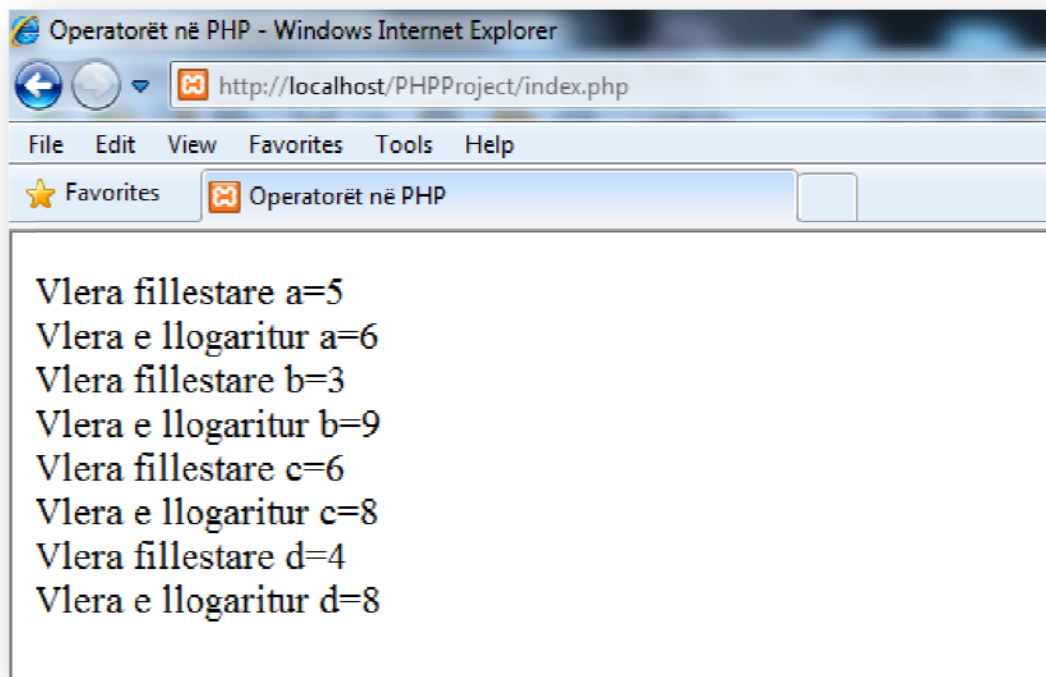
Ne këtë ushtrim është treguar përdorimi i operatorëve në PHP:

```

<html>
  <head><title>Operatorët në PHP</title></head>
  <body>
    <?php
      $a=5; $b=3; $c=6; $d=4; $x=7;
      echo "Vlera fillestare a=" . $a . "<br/>"; //5
      $a++;
      echo "Vlera e llogaritur a=" . $a . "<br/>"; //6
      echo "Vlera fillestare b=" . $b . "<br/>"; //3
      $b += 6;
      echo "Vlera e llogaritur b=" . $b . "<br/>"; //9
      echo "Vlera fillestare c=" . $c . "<br/>"; //6
      $c = ++$x;
      echo "Vlera e llogaritur c=" . $c . "<br/>"; //8
      echo "Vlera fillestare d=" . $d . "<br/>"; //4
      $d = $x++;
      echo "Vlera e llogaritur d=" . $d . "<br/>"; //8
    ?>
  </body>
</html>

```

Ekzekutimi (në browser Internet Explorer 8):

**Detyra 6:**

Ne këtë ushtrim është treguar përdorimi i degëzimeve (**if - if else - else**) në PHP:

```
<html>
  <head><title>Degëzimet në PHP</title></head>
  <body>
    <?php
      $x=3;
      $y=4;
      if($x>$y)
        echo "x është me i madh se y";
      else if($x==$y)
        echo "x është i barabartë me y";
      else
        echo "x është me i vogël se y";
    ?>
  </body>
</html>
```

Detyra 7:

Në këtë ushtrim është treguar se si përdoret **switch** komanda në PHP:

```
<html>
  <head><title>Switch komanda në PHP</title></head>
  <body>
    <?php
      $x=2;
      switch ($x) {
        case 1:
          echo "Numri 1";
          break;
        case 2:
          echo "Numri 2";
          break;
        case 3:
          echo "Numri 3";
          break;
        default:
          echo "Vlera e dhënë nuk është në mes 1 dhe 3";
      }
    ?>
  </body>
</html>
```

Detyra 8:

Në këtë ushtrim është treguar se si bëhet deklarimi dhe përdorimi i **variable variables** në PHP:

```
<?php
  $varStdenti = "K70001";
  $$varStdenti = "Filan Fisteku"; //Njëjtë me $K70001 = "Filan Fisteku";
  echo $K70001;
?>
```

Detyra 9:

Në këtë ushtrim është treguar se si bëhet deklarimi dhe përdorimi i konstanteve në PHP:

Konstantat ruajnë një vlerë sikur një variabël, por vlera e vendosur njëherë, nuk mund të ndërrohet më në skriptë. Dallimi tjetër në mes të variablave dhe konstantave është që konstantat ruajnë vetëm të dhëna boleanë, integer, float, ose string. Këto tipe bashkërisht janë të njohura si vlera skalare.

Definimi i këtyre konstantave mund të bëhet duke përdorur funksionin **define**:

```
<?php
define("Cmimi_PC",750);
define("Cmimi_Printer", 125.5);
define("Cmimi_HD", 75.05);
define("Cmimi_PC",25); //nuk mund të ndërrohet vlera e konstantës Cmimi_PC

echo "PC ". Cmimi_PC ."<br />";
echo "Printer ". Cmimi_Printer ."<br />";
echo "HD ". Cmimi_HD ."<br />";
?>
```

Detyra 10:

Në këtë ushtrim është treguar se si behët përdorimi i funksioneve **gettype** dhe **settype** në PHP:

string **gettype**(mixed *var*);

int **settype**(mixed *var*, string *type*);

```
<?php
$a=15;
echo gettype($a). '<br />'; // provo edhe për NULL
settype($a, 'string');
echo gettype($a). '<br />';
?>
```

Detyra 11:

Në këtë ushtrim është treguar se si behët përdorimi i disa funksioneve për të testuar tipin e variablave (**type-testing**).

Këto funksione pranojnë një variabël si argument dhe kthejnë **true** ose **false**, varësisht se a ë

- **is_string()** - kontrollon nëse variabla është string,
- **is_array()** - kontrollon nëse variabla është varg,
- **is_double()** - kontrollon nëse variabla është double,
- **is_float()** - kontrollon nëse variabla është float,
- **is_real()** - kontrollon nëse variabla është real,
- **is_long()** - kontrollon nëse variabla është long,
- **is_int()** - kontrollon nëse variabla është int,
- **is_integer()** - kontrollon nëse variabla është integer,
- **is_object()** - kontrollon nëse variabla është object,

- **is_resource()**- kontrollon nëse variabla është resource,
- **is_null()**- kontrollon nëse variabla është null,
- **is_scalar()**-kontrollon nëse variabla është një scalar, si një p.sh integer, boolean, string, ose float.
- **is_numeric()**-kontrollon nëse variabla është ndonjë lloj I numrit ose numër string.
- **is_callable()**-kontrollon nëse variabla është emër valid I funksionit.

```
<?php
    $a='84.25';
    if(is_numeric($a))
        echo 'Variabla a ka vlerë numerike<br/>';
    if(is_string($a))
        echo 'Variabla b ka vlerë string<br/>';
    if(!is_integer($a))
        echo 'Variabla a nuk ka vlerë integer';
?>
```

Detyra 12:

Në këtë ushtrim është treguar se si bëhet përdorimi I disa funksioneve për të testuar statusin e variablave (*isset*, *empty*, *unset*):

```
<?php
    $a="php";

    if(isset($a))
        print "Variabla 'a' ka vlerë! <br/>";
    else
        print "Variabla 'a' nuk ka vlerë! <br/>";

    unset($a);

    if(isset($a))
        print "Variabla 'a' ka vlerë! <br/>";
    else
        print "Variabla 'a' nuk ka vlerë! <br/>";

?>
```

Detyra 13:

Në këtë ushtrim është treguar se si behët përdorimi I komandës **while**.

Struktura e përgjithshme është:

```
while (kushti)
    shprehja;
```

```

<html>
  <head><title>While komanda në PHP</title></head>
  <body>
    <table border="1">
      <tr bgcolor="yellow">
        <td>Distanca</td>
        <td>Çmimi</td>
      </tr>
      <?php
        $distanca = 50;
        while ($distanca<250)
        {
          print "<tr> <td> $distanca</td><td>" . $distanca/10;
          print "</td></tr>";
          $distanca+=50;
        }
      ?>
    </table>
  </body>
</html>

```

Detyra 14:

Në këtë ushtrim është treguar se si behët përdorimi I komandës **do..while**.

Struktura e përgjithshme është:

```

do
    shprehja;
while (kushti);

```

```

<?php
    $distanca = 50;
    do
    {
        print "<tr> <td> $distanca</td><td>" . $distanca/10;
        print "</td></tr>";
        $distanca+=50;
    }
    while ($distanca<250)
    ?>

```

Detyra 15:

Në këtë ushtrim është treguar se si behët përdorimi I komandës **for** dhe **foreach**.
(foreach-me vonë në array)

```
for( shprehja1; kushti; shprehja2)
    shprehja3;
```

```
<table border="1">
  <tr bgcolor="yellow">
    <td>Distanca</td>
    <td>Çmimi</td>
  </tr>
  <?php
    for($distanca=50; $distanca<250; $distanca+=50)
    {
      print "<tr><td>$distanca</td><td>". $distanca/10;
      print "</td></tr>";
    }
  ?>
</table>
```

Përdorimi i vargjeve (Arrays)

Detyra 16:

Defino vargun (*array*)?

Variablat e përdorura me lartë ishin variabla skalare, në të cilat ruhet vetëm një vlerë e vetme, ndërsa variablat vargje (*array*) ruan një bashkësi (sekuencë) të vlerave.

Një varg (*array*) mund të ketë shumë elemente, dhe secili element mund të ketë një vlerë të vetme, p.sh një tekst ose numër. Vargu (*array*) mund të përmbaj vargje (*arrays*) - varg shumëdimensional (*multidimensional array*)



Fig. 1-Gjuhët programuese mund të ruhen në një varg (*array*) GjuheProg.

Detyra 17:

Në këtë ushtrim është treguar se si behët inicializimi i vargjeve (*arrays*).

```
<?php
    $GjuheProg = array("C++","C#","VB","Java");
?>
```

Vargu me lartë mund të krijohet edhe në këtë menyrë:

```
<?php
    $GjuheProg[0]="C++";
    $GjuheProg[1]="C#";
    $GjuheProg[2]="VB";
    $GjuheProg[3]="Java";
?>
```

Detyra 18:

Në këtë ushtrim është treguar se si behët përdorimi i llupave (**for**, **foreach**) për tu qasur vargut (array).

```
<?php
    $GjuheProg = array("C++","C#","VB","Java");

    //shtypja e antarëve te vargut me anë të llupes (for)
    print "shtypja e antareve te vargut me ane te llupes (for)<br>";
    for($i=0; $i<4; $i++)
    {
        echo "$GjuheProg[$i] <br>";
    }

    //shtypja e antarëve te vargut me anë të llupes(foreach)
    print "shtypja e antareve te vargut me ane te llupes (foreach)<br>";
    foreach($GjuheProg as $gjuhe_prog)
        print $gjuhe_prog . "<br>";
?>
```

Detyra 19:

Në këtë ushtrim është treguar se si behët inicializimi i vargjeve (arrays) me tregues të ndryshëm si dhe qasja në elemente të vargut.

```
<?php
$cmimet = array( "Libëri për C++"=>100,
                 "Libëri për C#"=>120,
                 "Libëri për Java"=>80
                );

foreach ($cmimet as $celesi => $vlera)
    echo $celesi."=>".$vlera."<br />";

//ose kjo mund të realizohet edhe në këtë menyre
//$cmimet = array("Libëri për C++"=>100);
//$cmimet["Libëri për C#"]=120;
//$cmimet["Libëri për Java"]=80;
?>
```

Detyra 20:

Në këtë ushtrim është treguar se si behët sortimi i vargut (array).

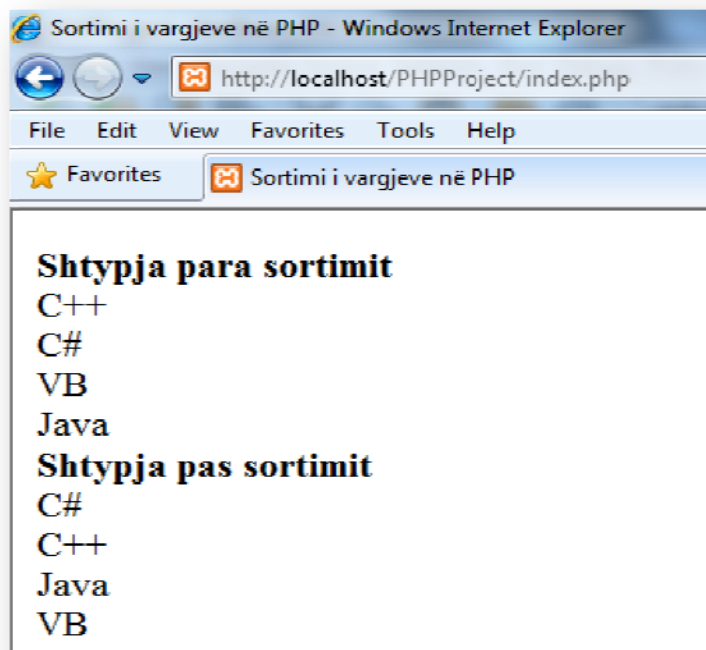
```
<html>
  <head><title>Sortimi i vargjeve në PHP</title></head>
  <body>
    <?php
      $GjuheProg = array("C++","C#","VB","Java");

      echo "<b>Shtypja para sortimit</b><br>";
      for($i =0; $i<4; $i++) {
        echo $GjuheProg[$i]."<br/>";
      }

      //sortimi i vargut
      sort($GjuheProg);

      echo "<b>Shtypja pas sortimit</b><br/>";
      for($i =0; $i<4; $i++) {
        echo $GjuheProg[$i]."<br/>";
      }
    ?>
  </body>
</html>
```

Ngjashëm behet sortimi edhe kur kemi të bëjmë me vlera numerike. Gjithashtu kemi edhe funksionin **rsort()**.



Detyra 21:

Në këtë ushtrim është treguar deklarimi i vargjeve multidimensional.

```
<?php
$products = array(array( 'Saab', 'Tires', 100 ),
                   array( 'Volvo', 'Oil', 10 ),
                   array( 'BMW', 'Spark Plugs', 4 ) );

for($i=0;$i<3;$i++)
{
    for($j=0; $j<3; $j++)
        print $products[$i][$j].['.$i.$j.'];
    print '<br>';
}
?>
```


Përdorimi i funksioneve në PHP

Kufizimet për emërtimin e funksioneve:

- Funksioni nuk mund të ketë emër të njëjtë me ndonjë funksion ekzistues.
- Emri i funksionit mund të përmbaj vetëm shkronja, numra, dhe nënvija (_ undersores).
- Emri i funksionit nuk mund të filloj më numër.

Detyra 1:

Në këtë ushtrim është treguar deklarimi dhe thirrja e funksioneve në PHP.

```
<?php
//deklarimi i funksionit 'emri_i_funksiont'
function emri_i_funksiont()
{
    //komandat ne trupin e funksionit
    echo "Është thirrur funksioni 'emri_i_funksiont'.";
}
//thirrja e funksionit
emri_i_funksiont();
?>
```

Detyra 2:

Të definohet funksioni **shuma** i cili pranon dy parametra hyrës (x dhe y), dhe kthen shumën e tyre ($z = x + y$).

```
<?php
function shuma($x, $y) {
    $z = $x + $y;
    return $z;
}
$a = 2;
$b = 3;
echo "Shuma e $a dhe $b është: ". shuma($a, $b);
?>
```

Funksioni **shuma** mund të definohet edhe në këtë mënyrë:

```
function shuma($x, $y) {
return $x + $y; }
```

Detyra 3:

Të definohet funksioni ***create_table*** i cili merr një parametër hyrës varg njëdimensional dhe shfaq e shfaqë si tabelë.

```
<?php
function create_table($data)
{
    echo '<table border=1>';
    reset($data);
    $value = current($data);
    while ($value)
    {
        echo "<tr><td>$value</td><tr>\n";
        $value=next($data);
    }

    //ose kjo mund te realizohet edhe ne kete menyre
    //foreach($data as $t)
    //    echo "<tr><td>$t.a</td><tr>\n";

    echo '</table>';
}
$vargu = array ('HTML5','JavaScript','CSS','PHP');
create_table($vargu);
?>
```

Detyra 4:

Në këtë ushtrim është treguar përdorimi i variablave **global**.

```
<?php

function fn()
{
    global $var;
    $var = 'vlera';
    echo 'brenda funksionit, $var = '.$var.'<br />';
}

fn();
echo 'jashtë funksionit, $var = '.$var.'<br />';
?>
```

String Manipulation

Këtu do të diskutojmë për:

- Formatimin e stringjeve
- Bashkimin dhe ndarjen e stringjeve
- Krahasimin e stringjeve
- Zgjedhjen dhe zëvendësimin e nënstringjeve me string funksionet

Formatimi I Stringjeve: **trim(), ltrim(), dhe rtrim()**

Detyra 1:

Në këtë ushtrim është treguar se si bëhet përdorimi I funksionit **trim()**.

Funksioni **trim()** heq zbrastirat nga fillimi dhe fundi I stringut dhe kthen rezultatin një string:

```
<?php
$var = " ne ";
print "Programimi".$var."Internet ";
print "<br/>";
$var = trim($var);
print "Programimi".$var."Internet ";
?>
```

Programimi ne Internet
ProgramimineInternet

Detyra 2:

Në këtë ushtrim është treguar se si bëhet përdorimi I funksionit **ltrim()**.

```
<?php
$var = " në ";
print "Programimi".$var."Internet ";           // Programimi në Internet
print "<br/>";
$var = ltrim($var);
print "Programimi".$var."Internet ";           //Programiminë Internet
?>
```

Detyra 3:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksionit **rtrim()**.

```
<?php
$var = " në ";
print "Programimi".$var."Internet ";
print "<br/>";
$var = rtrim($var);
print "Programimi".$var."Internet ";    //Programimi nëInternet
?>
```

Formatimi i stringut për printim

Për të printuar stringjet në browser mund të përdoret komanda **echo**.

PHP gjithashtu përkrahë komandën **print()** që është e njëjtë me **echo**, por kthen një vlerë (true ose false).

Detyra 4:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksioneve **echo** dhe **print()** :

```
<?php
    echo("Printimi me komandën echo<br/>");

    if(print("Printimi me komandën print<br/>"))
    {
        print "Printimi është bërë me sukses!";
    }
    else
    {
        print "Printimi ka dështuar!";
    }
?>
```

Detyra 5:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksioneve **printf()** :

Mund të aplikohen edhe disa funksione për formatime me të ndërlikuara duke përdorur **printf()** dhe **sprintf()**.

```
<?php
$totali = 12.4;
echo "Shuma totale është $totali";
print "<br>";
printf ("Shuma totale është %s", $totali);
```

```
$tvsh = 16;
print "<br>";
printf ("Shuma totale eshte %.2f (tvsh eshte %.2f perqind)", $totali, $tvsh);    //%%
?>
```

%s në formatin e stringut është quajtur shndërrim specifik (*conversion specification*). Kjo zëvendëson me string, në këtë rast kjo zëvendësohet me \$totali të interpretuar si një string.

Detyra 6:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksioneve **strtoupper()**, **strtolower()**, **ucfirst()** dhe **ucwords()**:

```
<?php
$var = "Programimi në internet";
print strtoupper($var);    //rez: PROGRAMIMI NË INTERNET
print "<br/>";
print strtolower($var);    //rez: programimi në internet
print "<br/>";
print ucfirst($var); //Make a string's first character uppercase //rez: Programimi në internet
print "<br/>";
print ucwords($var); //Uppercase the first character of each word in a string //rez: Programimi Në Internet
?>
```

Detyra 7:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksioneve **addslashes()** dhe **stripslashes()**.

```
<?php
$var = "Program'imi ne internet";
print "<br />";
print addslashes($var);    //Program\'imi ne internet
print "<br />";

$var = "Program\'imi ne inte\"rnet"; //Program\'imi ne inte"rnet    \\\
print $var;
print "<br />";
print stripslashes($var);    //Program'imi ne inte"rnet
?>
```

Bashkimi dhe ndarja e stringjeve me string funksione

Detyra 8:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksioneve **explode()**.

```
<?php
    $email = "filan@gmail.com";
    $email_array = explode("@",$email);
    print $email_array[0];
    print "<br/>";
    print $email_array[1];
?>
```

Detyra 9: Në këtë ushtrim është treguar se si bëhet përdorimi i funksioneve **implode()** dhe **join()**.

```
<?php
    $email_array[0]="filan";
    $email_array[1]="gmail.com";
    $email = implode("@", $email_array);
    //ose $email = join("@", $email_array);
    print $email;
?>
```

Detyra 10:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksionit **strtok()**.

Përndryshe nga funksioni **explode()**, i cili ndan stringun në pjesë përnjëherë, **strtok()** bashkon pjesët (të quajtura *tokens*) nga një string në të njëjtën kohë.

```
<?php
    $var = "Prog ne internet";
    $strvar = strtok($var, ' ');
    echo $strvar.'<br />';
    while ($strvar!="")
    {
        $strvar = strtok(' ');
        echo $strvar.'<br />';
    }
?>
```

Detyra 11:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksionit **substr()**.

```
<?php
    $var = "Programimi ne internet";
    echo substr($var, 1); //rogramimi ne internet
    echo "<br />";
    echo substr($var, -8); //internet
```

```
echo "<br />";  
echo substr($var, 0, 10); //Programimi  
echo "<br />";  
echo substr($var, 4, -3); //ramimi ne inter  
?>
```

Krahasimi I stringjeve

Detyra 12:

Në këtë ushtrim është treguar se si bëhet krahasimi I stringjeve duke përdorur ==.

```
<?php  
$str1="abc";  
$str2="abc";  
if($str1==$str2)  
{  
    print "Stringjet janë të barabarta!";  
}  
else  
{  
    print "Stringjet nuk janë të barabarta";  
}  
?>
```

Detyra 13:

Në këtë ushtrim është treguar se si bëhet përdorimi I funksionit **strcmp()**.

Këto funksione (**strcmp()**, **strcasecmp()**, and **strnatcmp()**) mund të përdoren për të renditur stringjet (ideale për sortimin e të dhënave).

Ky funksion është *case sensitive*.

```
<?php  
$str1="a";  
$str2="a";  
print strcmp($str1, $str2);    //0  
  
print "<br />";  
  
$str1="a";  
$str2="b";  
print strcmp($str1, $str2);    //-1  
  
print "<br />";
```

```

    $str1 ="b";
    $str2 ="a";
    print strcmp($str1, $str2);    //1
?>

```

Binary safe string comparison

Return Value

Type: int

< 0 if str1 is less than str2; > 0 if str1 is greater than str2, and 0 if they are equal.

Detyra 14:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksionit **strcasecmp()**.

Ky funksion është identik me funksionin **strcmp()** në përjashtim që ky nuk është *case sensitive*.

Detyra 15:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksionit **strlen()**.

```

<?php
    $email = "a@a.to";
    if(strlen($email)<6)
    {
        echo 'Kjo email adrese nuk eshte valide.';
        exit;
    }
? >

```

Zgjedhja dhe zëvendësimi i substringjeve me string funksione

Gjetja e stingjeve në stringje

Detyra 15:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksioneve **strstr()**, **strchr()**, **strrchr()**, dhe **stristr()**.

```

<?php
    $str = "Programimi ne internet";
    $str_ker = "net";
    if(strstr($str,$str_ker))
    {
        print "Stringu '$str_ker' është gjetur.";
    }

```



```
    else
    {
        print "Stringu '$str_ker' nuk është gjetur.";
    }
?>
```

Detyra 16:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksioneve **strpos()** dhe **strrpos()**.

Parametri opsional **offset** specifikon pikën brenda stringut kërkues për të filluar kërkimin.

```
<?php
    $var = "Programimi në internet";
    print strpos($var, "m");           //6
    print "<br />";
    print strrpos($var, "m", 7);       //8
?>
```

Funksioni **strrpos()** pothuajse është identik, mirëpo kthen pozitën e fundit të *needle* në *haystack*.

Nëse fjala kërkuese nuk është në string, atëherë funksionet **strpos()** dhe **strrpos()** kthejnë vlerën *false*.

```
<?php
    $var = "efiek";
    $rez = strpos($var, 'e');
    if ($rez === false)
        echo 'Stringu nuk është gjetur';
    else
        echo "Stringu është gjetur në pozitën $rez";
?>
```

Detyra 17:

Në këtë ushtrim është treguar se si bëhet përdorimi i funksioneve **str_replace()** dhe **substr_replace()**.

```
<?php
    $var = "programimi ne internet";
    print $var;
    print "<br />";
    $var = str_replace("ne", "xx", $var);
    print $var;
?>
```

Funksioni **substr_replace()** gjenë dhe zëvendëson një substring të veçantë të stringut bazuar në pozitën e tij. Ka prototipin si në vijim:

string substr_replace(string *string*, string *replacement*, int *start*, int [*length*])

P.sh

```
$test= substr_replace($test, 'X', -1);
```

Dërgimi i email-it përmes PHP

Detyra 1:

Ne këtë ushtrim është treguar përdorimi i funksionit mail në PHP për dërgimin e email-ave:

mail(\$to, \$subject, \$message, \$additional_headers = null, \$additional_parameters = null)

Funksioni mail është funksion boolean, i cili kthen true nëse email është dërguar me sukses, përndryshe false.

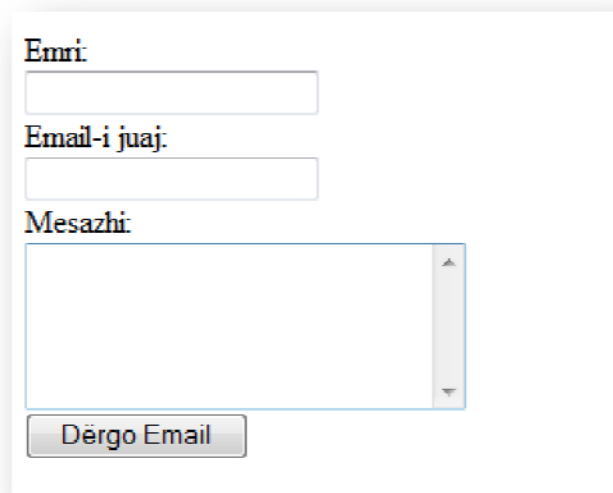
EmailForm.html

```
<html>
  <head>
    <title>Forma për dëgimin e email-it</title>
  </head>
  <body>
    <form name="fsendmail" action="SendMail.php" method="post">
      <label for="_emri">Emri:</label><br/>
      <input type="text" name="emri" id="_emri" /><br/>

      <label for="_email">Email-i juaj:</label><br/>
      <input type="email" name="email" id="_email"/><br/>

      <label for="_msg">Mesazhi:</label><br/>
      <textarea name="msg" id="_msg" cols="25" rows="5"></textarea><br/>

      <input type="submit" id="btndergo" value="Dërgo Email"/>
    </form>
  </body>
</html>
```



Emri:

Email-i juaj:

Mesazhi:

Dërgo Email

SendMail.php

```
<?php

$name = $_POST["emri"];
$email = $_POST["email"];
$msg = $_POST["msg"];

$to = "feedback@example.com";
$subject = "Feedback-u nga ueb sajti! ";
$mailcontent = 'Emri i dërguesit: '. $name . "\n"
               .'Email-i i dërguesit: '. $email . "\n"
               ."Komenti i dërguesit: \n".$msg ."\n";

$headers = 'from: webserver@example.com';

if(mail($to, $subject, $coments, $headers)) {
    echo 'Email-i është dërguar me sukses!';
}
else {
    echo 'Dërgimi i email-it ka dështuar!';
}
?>
```

Gjithashtu, nëse është e nevojshme duhet të konfigurohet **smtp**-ja dhe **porti** në fajllin 'php.ini'.

Detyra 2:

Modifikimi i detyrës paraprake për dërgimin e email-it përmes gmail smtp-së:

SendMail.php

```
<?php
include "Mail.php";
$name = $_POST["emri"];
$email = $_POST["email"];
$msg = $_POST["msg"];

$from = "besmir.sejdiu@gmail.com";
$to = "filan.fisteku@gmail.com";

$subject = "Feedback-u nga ueb sajti! ";

$mailcontent = 'Emri i dërguesit: '. $name . "\n"
               .'Email-i i dërguesit: '. $email . "\n"
               ."Komenti i dërguesit: \n".$msg ."\n";
```

```
$host = "ssl://smtp.gmail.com";
$port = "465";
$username = "besmir.sejdiu@gmail.com";
$password = "password";

$headers = array ('From' => $from,
                  'To' => $to,
                  'Subject' => $subject);

$smtp = Mail::factory('smtp',
    array ('host' => $host,
          'port' => $port,
          'auth' => true,
          'username' => $username,
          'password' => $password));

$mail = $smtp->send($to, $headers, $mailcontent);

if($mail) {
    echo 'Email-i është dërguar me sukses!';
}
else {
    echo 'Dërgimi i email-it ka dështuar!';
}

?>
```

Puna me fajlla në PHP

Detyra 1:

Ne këtë ushtrim është treguar se si punohet me fajlla në PHP:

RegjistrimiKlienteve.html

```
<html>
  <head>
    <title>Regjistrimi i klientav</title>
  </head>
  <body>
    <p>Të dhënat e klientit</p>
    <form name="frmKlientet" action="Klientet.php" method="POST">
      Id Klienti: <br/>
      <input type="text" name="IdKlienti" /><br/>
      Klienti: <br/>
      <input type="text" name="Klienti"/><br/>
      Adresa:<br/>
      <input type="text" name="Adresa"/><br/>
      <input type="submit" name="btnDergo" value="Dergo"/>
    </form>
  </body>
</html>
```

Klientet.php

```
<?php
$IdKlienti = $_POST['IdKlienti'];
$Klienti = $_POST['Klienti'];
$Adresa = $_POST['Adresa'];
$path = $_SERVER['DOCUMENT_ROOT'];

$regDate = date("d.m.y H:i");
$str = "\n". $regDate.", ".$IdKlienti. ", ". $Klienti. ", ". $Adresa ;

//shkruarja në fajll
$fp = fopen('$path/../../test.txt', "a");
fwrite($fp, $str);

//leximi nga fajlli
$fr = fopen('$path/../../test.txt', "r");

echo "<table border=1>";
while (!feof($fr)) {
    $row = explode(',', fgetss($fr));
    echo "<tr>
      <td>".$row[0]."</td>
```

```
        <td>"$.row[1]."</td>  
        <td>"$.row[2]."</td>  
        <td>"$.row[3]."</td>";  
    }  
    echo "</table>";  
?>
```

Object-Oriented PHP

Detyra 1:

Në këtë ushtrim është treguar krijimi i klasave, attributeve dhe funksioneve në PHP:

Përmes klasave jepet një mundësi e shkruarjes së programeve, të cilët sipas nevojës, lehtë ndryshohen, duke i ndryshuar vetëm klasat. Gjatë kësaj, problemi që zgjidhet copëtohet në klasa dhe në deklarimin e objekteve përkatëse, gjë që ka një rëndësi të veçantë kur kemi të bëjmë me programe komplekse, me çka zvogëlohet mundësia e gabimeve.

Kur e krijojmë një klasë në PHP duhet të përdoret fjala kyçe **class**.

Struktura e klasës

Definimi i një klase minimale duket si ne vijim:

```
class classname
{
}
```

Për të qenë të vlefshme klasat ju nevojiten attribute dhe funksione. Atributet krijohen duke deklaruar variablat brenda definimit të klasës duke përdorur fjalën kyçe **var**. Kodi ne vijim krijon një klasë me emrin **classname** me dy attribute, **\$attribute1** dhe **\$attribute2** :

```
class classname
{
    var $attribute1;
    var $attribute2;
}
```

Funksionet krijohen duke deklaruar funksionet brenda definimit të klasës. Kodi në vijim krijon një klasë me emrin **classname** me dy funksione që nuk bëjnë asgjë. Funksioni **operation1()** nuk ka parametra, dhe funksioni **operation2()** ka dy parametra:

```
class classname
{
    var $attribute1;
    var $attribute2;

    function operation1()
    {
    }
    function operation2($param1, $param2)
    {
    }
}
```


Detyra 2:

Në këtë ushtrim është treguar përdorimi i konstruktorëve në PHP:

Me qëllim të inicializimit të variablave të cilat përfshihen në komponentët e klasës, brenda saj definohen funksione të veçanta, të cilat njihen si *konstruktorë*.

Këto funksione ekzekutohen automatikisht, kur deklarohen objektet përkatëse të klasave në fjalë. Konstruktorët dallohen nga funksionet e zakonshme brenda klasës. Konstruktorët mund të kenë ose mund të mos kenë parametra formalë.

Pra, shumica e klasave kanë tip të veçantë të funksionit të quajtur **constructor**. Konstruktori thirret kur të krijohet një objekt dhe gjithashtu kryen normalisht punët e inicializimit si vendosjen e attributeve vlerave fillestare ose krijimin e objekteve tjera të nevojshme nga ky objekt.

Konstruktorët deklarohen sikurse funksionet tjera por kanë emër të veçantë **__construct()**. Ky ndryshim është në PHP5. Në versionet e më hershme funksionet e konstruktorit kanë pasur emrin e njëjtë sikurse klasa.

Megjithëse konstruktorin mund të thirrni në mënyrë manuale, qëllimi i tij kryesor është të thirret automatikisht kur një objekt të krijohet. Kodi në vijim deklaron një konstruktor në klasën **classname**:

```
class classname
{
    function __construct($param)
    {
        echo "Thirrja e konstruktorit me parametër $param <br />";
    }

    var $attribute1;
    var $attribute2;

    function operation1()
    {
    }
    function operation2($param1, $param2)
    {
    }
}
```

Detyra 3:

Çka janë destruktort? Si deklarohen destruktoret në PHP?

E kundërta e konstruktorit është destructor-i. Këta ekzekutohen para se klasa të shkatërrohet, gjë e cila ndodh automatikisht kur të gjitha referencat në klasë bëhen unset.

Ngjashëm emërohen edhe destruktoret **__destruct()**. Destruktorët nuk mund të kenë parametra.

Me qëllim të lirit të hapësirës memoruese, të cilën e shfrytëzon një objekt, pasi objekti më nuk nevojitet, brenda klasës definohet një funksion i veçantë, i cili njihet si *destruktor*.

Destruktori mund të shfrytëzohet vetëm për fshirje të objekteve të deklaruara.

Kur përfundon ekzekutimi i programit, nëse nuk shfrytëzohen destruktore, kompjuteri automatikisht i fshin objektet e shfrytëzuara brenda tij. Prandaj, edhe shfrytëzimi i destruktoreve të programet e zakonshme nuk ka ndonjë rëndësi të veçantë.

Detyra 4:

Në këtë ushtrim është treguar si bëhet instancimi i klasave në PHP:

Pasi që të deklarohet klasa, duhet të krijohet një object- individ i veçantë që është anëtar i klasës-për të punuar më të. Kjo është e njohur edhe si *krijimi i një instance* ose *instancimi i klasës*.

Një object krijohet duke përdorur fjalën kyçe **new**. Kur ta bëni këtë duhet ta specifikoni se cilës klasë ose instance e cilës klasë është objekti i juaj dhe të ofroni ndonjë parametër të kërkuar nga konstruktori. Kodi në vijim deklaron një klasë të quajtur **classname** me një konstruktor dhe mandej krijon dy objekte të tipit **classname** :

```
class classname
{
    function __construct($param)
    {
        echo "Thirrja e konstruktorit me parametër $param <br />";
    }
}

$a = new classname('First');
$b = new classname('Second');
```

Për arsye që konstruktori thirret sa herë që ta krijoni një object, ky kod paraqet rezultatin si në vijim:

```
Thirrja e konstruktorit me parametër First
Thirrja e konstruktorit me parametër Second
```

Detyra 5:

Në këtë ushtrim është treguar përdorimi i attributeve të klasës në PHP:

Brenda klasës, ju keni qasje tek pointeri special **\$this**. Nëse një atribut i klasës aktuale quhet **\$attribute**, ju i referoni atij si **\$this->attribute** kur i qaseni ose i vendosni një variabël nga një funksion brenda klasës. Kodi në vijim demonstroi vendosjen dhe qasjen e një atributi brenda klasës.

```
class classname
{
    var $attribute;

    function operation($param)
    {
```

```

        $this->attribute = $param;
        echo $this->attribute;
    }
}
$a = new classname();
$a->operation("value");

```

Ky shembull nuk e kufizon qasjen në attribute andaj ju mund ti qaseni atyre nga jashtë klasës :

```

class classname
{
    var $attribute;
}

$a = new classname();
$a->attribute = 'value';
echo $a->attribute;

```

Në përgjithësi nuk është ide e mirë të ju qasemi attributeve nga jashtë klasës. Një avantazh i orientimit në objekte është se inkurajon encapsulimin. Ju mund ta detyroni këtë duke përdorur funksionet **`__get`** dhe **`__set`**. Nëse, në vend të qasjes se attributeve të klasës direkt shkruani **accessor** funksionet, ju mund ti bëni qasjet përmes një sektori të vetëm të kodit. Kur fillimisht e shkruani një **accessor functions**, ata mund te duken si në vijim:

```

class classname
{
    var $attribute;

    function __get($name)
    {
        return $this->$name;
    }

    function __set ($name, $value)
    {
        $this->$name = $value;
    }
}

```

Ky kod ofron funksionet minimale të qasjes se atributit \$attribute. Funksioni **`__get()`** thjesht kthen vlerën e \$attribute, dhe funksioni **`__set()`** i'a shoqëron një vlerë të re \$attribute.

`__get()` merr një parametër – emrin e atributit- dhe rikthen vlerën e atij atributi. Ngjashëm funksioni **`__set()`** merr dy parametra: emrin e atributit dhe vlerën që doni t'ia vendosni. Këto funksione nuk thirren direkt. Nënviija e dyfishtë para emrit tregon që këto funksione kanë një kuptim të veçantë në PHP, sikurse funksionet **`__construct()`** dhe **`__destruct()`**.

```

<?php
class person
{
    var $emri, $mbiemri;

    function __get($name)
    {
        return $this->name;
    }
    function __set($name,$value)
    {
        $this -> name = $value;
    }
}
$personi = new person();
$personi->emri = "filan";
print $personi->emri;
?>

```

Si punojnë mandej këto? Nëse e instanconi klasën **\$a=new classname;** I përdorni funksionet **__get()** dhe **__set()** për të kontrolluar dhe vendosur vlerën e ndonjë atributi. Nëse shtypni **\$a->attribute=5;** kjo deklaratë në mënyrë **implicite** thërret funksionin **__set()** me vlerë të \$name të vendosur në “attribute” dhe vlerën e \$value të vendosur 5. Ju duhet ta shkruani funksionin **__set()** për me kontrollua ndonjë gabim. Funksioni **__get()** punon njëjtë. Nëse në kodin e juaj I referoheni **\$a->attribute** kjo shprehje në mënyrë implicite thërret funksionin **__get()** me parametër \$name të vendosur në “attribute”. Varet nga ju nëse doni të shkruani funksioni **__get()** për me kthi vlerë.

Detyra 6:

Në këtë ushtrim është treguar kontrollimi I qasjes me “private” dhe “public”

PHP5 përkrah tre qasje të ndryshme të modifikatorëve:

- Opcioni default është **public**, që d.m.th nëse nuk e specifikoni një modifikator qasjeje për një atribut ose metodë, ai do të jetë **public**. Anëtarët që janë public mund të jenë të qasshëm nga jashtë ose brenda klasës.
- Modifikatori I qasjes **privat (private access)** d.m.th që anëtari (item-i) I dukshëm mund të qaset vetëm nga brenda klasës. Mund ta përdorni tek të gjitha atributet nëse nuk jeni duke I përdorur **__get()** dhe **__set()**. Mund ti bëni disa metoda private, p.sh nëse janë funksione në dobi vetëm për përdorim brenda klasës.
- Modifikatori I qasjes së mbrojtur (**protected access**) d.m.th që anëtari I dukshëm mund të qaset vetëm nga brenda klasa. Gjithashtu ekziston në ndonjë nënklasë, pra mund ta mendoni në mes të private dhe public.

```

<?php

class Studenti

```

```

{
    public $NrIndeksit = "12345";
    var $emri= "Filan";
    private $mbiemri = "Fisteku";
    protected $add = "Prishtine";
}
$objStudenti = new Studenti();
print $objStudenti->NrIndeksit;
print '<br />';
print $objStudenti ->emri;
print '<br />';
print $objStudenti->mbiemri;
print '<br />';
print $objStudenti->add;

```

?>

<?php

```

class Studenti
{
    public function funk1()
    {
        return "Është thirru funksioni funk1";
    }
    function funk2()
    {
        return "Është thirru funksioni funk2";
    }
    private function funk3()
    {
        return "Është thirru funksioni funk3";
    }
}
$objStudenti = new Studenti();
print $objStudenti->funk1();
print '<br />';
print $objStudenti ->funk2();
print '<br />';
print $objStudenti->funk3();
print '<br />';

```

?>

Secili anëtar i klasës prezantohet me një modifikator të qasjes për të treguar nëse është privat ose public. Mund të mos ta shkruani fjalën kyçe public, sepse është default, por kodi me lehet kuptohet me të nëse jeni duke përdorur modifikatorët tjerë.

Fjalën kyçe **var** e kemi larguar para atributit; është zëvendësuar me modifikatorin **public**. Në këtë rast çdo gjë është bërë **publike**.

Detyra 7:

Në këtë ushtrim është treguar se si mund të behët thirrja e funksioneve te klasës:

Mund t'i thërrisni funksionet e klasës sikurse atributet e klasës. P.sh kemi klasën:

```
class classname
{
    function operation1()
    {
        print "funksion1<br/>";
    }
    function operation2($param1, $param2)
    {
        print $param1." ". $param2;
    }
}
```

Dhe krijo një object të tipit **classname** të quajtur \$a si në vazhdim:

```
$a=new classname();
```

Mandej funksionet thirren në mënyrën e njëjtë sikurse funksionet: duke përdorur emrin e tyre dhe duke vendosur ndonjë parametër që ju nevojitet në kllapa. Për shkak se këto funksione (operations) i takojnë një objekti me tepër se sa funksionet normale, duhet të specifikoni se cilit objekt i takojnë. Emri i objektit përdoret në mënyrë të njëjtë me atributet e objekteve, si në vazhdim :

```
$a->operation1();
$a->operation2(12, 'test');
```

Nëse funksionet (operations) kthejnë diçka, ju mund të tërhiqini të dhënat që kthehen si në vazhdim:

```
$x = $a->operation1();
$y = $a->operation2(12, 'test');
```

Detyra 8:***Në këtë ushtrim është treguar implementimi i trashëgimeve në PHP:***

Nëse klasa duhet të jetë nënklasë e tjerave, ju mund të përdorin fjalën kyçe ***extends*** për të specifikuar atë. Kodi në vazhdim krijon një klasë të quajtur B që trashëgohet nga klasa e definuar me herët e quajtur A:

```
<?php

class A
{
    var $atributi_A;
    function funk_A()
    {
        print "<br/>". $this->atributi_A;
    }
}
class B extends A
{
    var $atributi_B;
    function funk_B()
    {
        print "<br/>". $this->atributi_B;
    }
}
$Obj_B = new B();
$Obj_B ->atributi_A =2;
$Obj_B->funk_A();
$Obj_B->atributi_B=3;
$Obj_B->funk_B();

?>
```

Detyra 9:

Në këtë ushtrim është treguar kontrollimi i dukshmërisë përgjatë trashëgimit me private dhe protected:

Mund të përdorim modifikuesit e qasjes **private** dhe **protected** për të kontrolluar çfarë është trashëgimia. Nëse një atribut ose metodë është specifikuar si **private**, kjo nuk do të trashëgohet. Nëse një atribut ose metode është specifikuar si **protected**, kjo nuk do të duket jashtë klasës (sikurse një **private** element) por do të trashëgohet. Shqyrtojmë shembullin në vazhdim:

```
<?php
class A
{
    private function operation1()
    {
        echo "operation1 called";
    }
    protected function operation2()
    {
        echo "operation2 called";
    }
    public function operation3()
    {
        echo "operation3 called";
    }
}
class B extends A
{
    function __construct()
    {
        $this->operation1();
        $this->operation2();
        $this->operation3();
    }
}

$b = new B;

?>
```

Ky kod krijon një funksion (operator) të secilit tip në klasën A: **public**, **protected**, dhe **private**. B trashëgohet nga A. Në konstruktorin e B, mund të provojmë të thërrasim operatorët nga prindi.

Linja

```
$this -> operation1();
```

Paraqet një **fatal error** si në vazhdim:

Fatal error: Call to private method A::operation1() from context 'B'

Ky shembull tregon që operatoret privat nuk mund të thirren nga klasa fëmijë. Nëse komentojmë këtë rresht, thirrja e dy funksioneve të tjera do të punoj. Funksioni **protected** është trashëguar por mund të përdoret vetëm nga ana e klasës fëmijë. Nëse provojmë të shtojmë rreshtin:

```
$b->operation2();
```

Në fund të fajllit, do të marrim error-in në vazhdim:

```
Fatal error: Call to protected method A::operation2() from context ``
```

Sidoqoftë, mund të thirrjmë **operation3()** nga jashtë klasës, si ne vazhdim:

```
$b->operation3();
```

Mund ta bëjmë këtë thirrje sepse kjo është deklaruar si **public**.

Detyra 10:

Overriding

Në këtë kapitull, pamë deklarimet e attributeve të reja dhe operacioneve të një nënklase. Vlen dhe ndonjëherë është e dobishme të rideklorojmë atributet dhe funksionet e njëjta. Ju mund ta bëni këtë duke i dhënë një atribut në nënklasë vlerë të ndryshme default, atributit të njëjtë në superklasën e saj ose ti jepni një funksioni nënklasë funksionalitet të ndryshëm të funksionit të njëjtë në superklasën e tij. Ky aksion quhet **overriding**.

Për instancën, kemi një klasë A:

```
class A
{
    var $attribute = 'default value';
    function operation()
    {
        echo 'Something<br />';
        echo "The value of \$attribute is $this->attribute<br />";
    }
}
```

Nëse dëshironi të ndryshoni vlerën default të **\$attribute** dhe të ofroni funksionalitet të rij për **operation()**, mund ta krijoni klasën B, e cila e mbishkruan **\$attribute** dhe **operation()**:

```
class B extends A
{
    var $attribute = 'different value';
    function operation()
    {
        echo 'Something else<br />';
        echo "The value of \$attribute is $this->attribute<br />";
    }
}
```

```
}
}
```

Deklarimi i **B** nuk ka efekt tek definimi origjinal i **A**. shqyrtojmë dy linjat e kodit:

```
$a= new A();
$a -> operation();
```

Këto linja krijojnë një objekt të tipit **A** dhe thirrin funksionin e saj **operation()**. Këto produkte

Something

The value of \$attribute is default value

Sigurojnë që krijimi i **B** nuk ka ndryshuar **A**. Nëse krijojmë një objekt të tipit **B**, mund të marrim output-e të ndryshme.

Ky kod

```
$b = new B();
$b -> operation();
```

Prodhon

Something else

The value of \$attribute is different value

Në mënyrë të njëjtë sigurojmë që atributet ose funksionet e reja në një nënklasë nuk kanë efekt tek superklasa, atributet e mbishkruara (overriding) ose funksionet në një nënklasë nuk kanë efekt tek superklasa.

Detyra 11:

Në këtë ushtrim është treguar parandalimi i trashëgimit dhe mbishkruajtjes me final:

PHP5 ka fjalën kyçe **final**. Kur përdorim këtë fjalë kyçe para deklarimit të funksionit, funksioni nuk mund të mbishkruhet në ndonjë nënklasë. Për shembull, mund të shtojmë këtë në klasën **A** nga shembulli paraprak, si në vazhdim:

```
class A
{
    var $attribute = 'default value';

    final function operation()
    {
```

```
        echo 'Something<br />';  
        echo "The value of \$attribute is \$this->attribute<br />";  
    }  
}
```

Duke përdorur këtë trajtim parandalojmë juve nga mbishkruaja **operation()** në klasën **B**. Nëse ju tentoni të bëni këtë, ju do të merrni këtë error:

Fatal error: Cannot override final method A::operation()

Mundeni gjithashtu të përdorni fjalën kyçe **final** që të parandaloni një klasë nga krijimet e të gjitha nënklasave. Të parandaloni klasën A nga ekzistenca e nënklasave, ju mund të shtoni këtë si në vazhdim:

```
final class A  
{...}
```

Nëse tentoni të trashëgoni nga **A**, do të merrni këtë error :

Fatal error: Class B may not inherit from final class (A)

Detyra 12:

Trashëgimia e shumëfishtë në PHP:

Pak gjuhë të orientuara në objekte përkrahin trashëgimin e shumëfishtë, por si shumica as PHP nuk e përkrahë. Kjo do të thotë se secila klasë mund të trashëgoj nga vetëm një prind. Nuk ekziston ndonjë kufizim se për sa fëmijë mund të ndajë një prind i vetëm. Figura 6.1 tregon tri mënyra të ndryshme që klasat A, B dhe C mund të trashëgojnë.

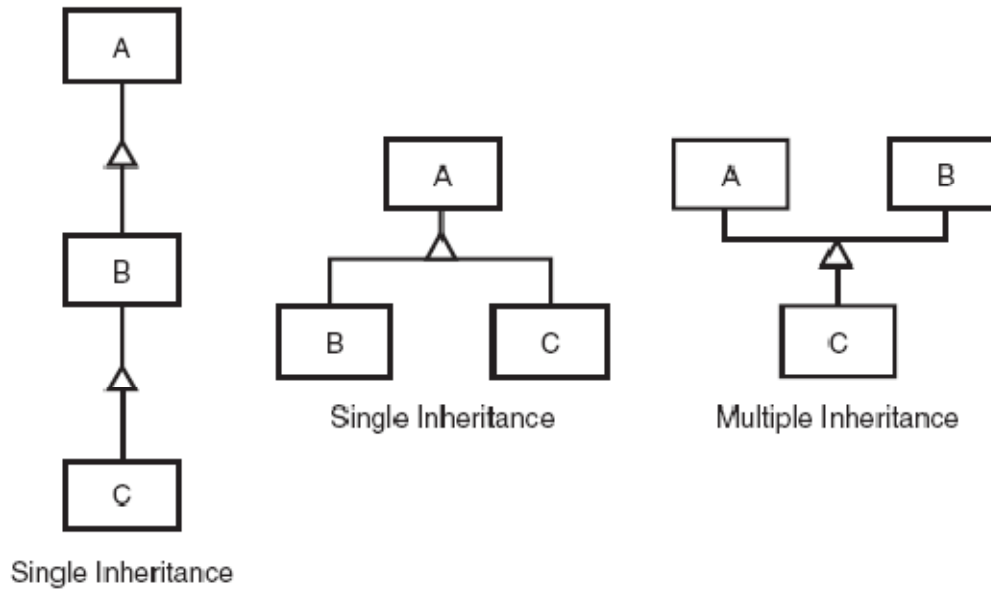


Figure 6.1 PHP does not support multiple inheritance.

Detyra 13:

Në këtë ushtrim është treguar implementimi i Interfasit:

PHP5 përkrahë **interfaces**. Ideja e interfeisit është që specifikon një bashkësi funksioneve që duhet të implementohet në klasa që implementojnë atë interfaces. P.sh, mund të vendosni të keni një bashkësi të klasave që duhet të jenë në gjendje ta shfaqin vetveten. Në vend që të kenë një klasë prind me funksionin **display()** prej të cilës trashëgojnë dhe mbishkruajnë, ju mund të implementoni një interface si ne vijim:

```

interface Displayable
{
    function display();
}
class webPage implements Displayable
{
    function display()
    {
        // ...
    }
}

```

Ky shembull tregon një lloj të tërthort të trashëgimisë së shumëfishtë sepse klasa **webPage** mund të trashëgoj nga një klasë dhe të implementoj një ose më shumë interface.

Nëse nuk e implementoni metodat e specifikuar në interface (në këtë rast, **display()**) do të keni një gabim fatal.

Regular Expressions (shprehjet e rregullta)

Mund të përdorim karakterin **.** (**pikë**) si një zëvendësim për ndonjë karakter tjetër të vetëm, përveç newline (\n).
P.sh shprehja e rregullt **.at** përshtat (barazon) string-et **'cat'**, **'sat'**, **'mat'**, **'#at'** midis tjerash.

[a-z]at –limitojnë karakteret mes a dhe z.

Shprehja në **[]** barazon (përputh) **vetëm një karakter të vetëm** (është një *character class*).

[a-zA-Z] –ndonjë shkronjë e madhe ose e vogël

[aeiou] –d.m.th ndonjë zanore

[^a-z] –përputh ndonjë karakter **që nuk është** në mes a dhe z

Karakteri **^** kur është brenda kllapave nënkupton (nuk, jo), ndërsa kur është jashtë kllapave ka d.m.th tjetër.

Përsëritjet (Repetition)

Simboli ***** d.m.th që modeli (mostra) i dhënë mund të përsëritet 0 ose më shumë herë.

Simboli **+** d.m.th që mostra e dhënë mund të përsëritet 1 ose më shumë herë. Simboli duhet të paraqitet direkt pas pjesës së shprehjes në të cilën aplikohet.

[[a-zA-Z]] + d.m.th së paku një karakter alfanumerik

Nënshprehjet (Subexpressions)

(very)*large dmth barazon: **'large'**, **'very large'**, **'very very large'**, e kështu me radhë.

Detyra 1:

```
<?php
$var = "shume mire";
if(ereg("shume )*mire",$var)           // if(ereg("(shume )+mire",$var)) //ereg - case insensitive
{
    print "Stringu është përputhur me modelin";
}
else
{
    print "Stringu nuk është përputhur me modelin";
}
?>
```

Nënshprehjet e numëruara

Ju mund të specifikoni sa here doni që diçka të përsëritet, duke e përdorur shprehjen numerike në kllapa gjarpërore ({}). Mund të tregoni saktë numrin e përsëritjeve.

{{3}} –d.m.th saktësisht tri përsëritje

{{2,4}} –nga dy deri në katër përsëritje

{{2, }} –së paku dy përsëritje

p.sh: **{very} {1,3}**

'very', 'very very', 'very very very'

Simboli ^ përdoret në fillim të shprehjes së rregullt të tregoi që duhet të paraqitet në fillim të stringut që kërkohet ose nën shprehja e veçanët duhet të paraqitet në fillim të stringut që kërkohet.

Simboli \$ -analoge, por në fund të string-ut

P.sh: **^bob** – përputh bob në fillim të string-ut (bobinë)

com\$ – përputh com në fund të string-ut (bercom)

^[a-z]\$ - përputh (gjen) stringjet që përmbajnë një karakter të vetëm nga **a** në **z**

'|' – paraqet një zgjidhje në shprehjen e rregullt

(com)|(edu)|(net) –përputh njëren nga stringjet brenda kllapave

Karakteret speciale

Karakteret speciale, sikurse '.', '{', ose '\$', për të përputhur, një backslash '\' duhet të vendoset para,

p.sh: **\.**, ose **\{**, ose **\\$**

Shifro (kodo) formatin standard të një e-mail adrese

^[a-zA-Z0-9_\\-\\.]+@[a-zA-Z0-9\\-\\.]+[a-zA-Z0-9\\-\\.]+\$

Detyra 2:

Shembulli në vazhdim validon email-in duke përdorur funksionin **ereg()** :

```
<?php
$email = "test@hotmail.com";
print "Email-i testues: '$email'";
print "<br>";
if (ereg("^[a-zA-Z0-9 _\-\.\.]+@[a-zA-Z0-9\-\.\.]+[a-zA-Z0-9\-\.\.]+$", $email))
{
    echo "Email adresa është valide.";
}
else
{
    echo "Email adresa nuk është valide.";
}
?>
```

Detyra 3:

Në këtë ushtrim është treguar kërkimi i substring-eve me regular expressions:

Dy funksionet e përdorshme në PHP për stilin POSIX të shprehjeve të rregullta janë **ereg()** dhe **eregi()**.

ereg() -ka këtë prototip: **int ereg(string pattern, string search, array [matches]);**

ose

int **ereg** (string \$pattern , string \$string , array \$regs)

Kërkon *\$string* për të krahasuar me shprehjen e rregullt të dhënë në *\$pattern* në mënyrën 'case-sensitive'

pattern – shprehja e rregullt 'case sensitive'

string- stringu hyrës

regs – nëse modeli është përputhur me stringun hyrës dhe funksioni thirret me argumentin e tretë *regs*, përputhjet- matches do të ruhen në elementet e vargut *regs*.

\$regs[1] – do të përmban substringun me të cilin fillon shprehja e parë në kllapa nga ana e majt;

\$regs[2] – do të përmban substringun e dytë, e kështu me radhë.

\$regs[0] – do të përmban një kopje të komplet stringut që është përputhur.

Detyra 4:

Shembulli në vazhdim e merr datën në formatin ISO (YYYY-MM-DD) dhe e shfaq at në formatin DD.MM.YYYY

```
<?php
$date = "2010-04-13";
print $date;
print "<br>";
if (ereg ("([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $date, $regs)) {
    echo "$regs[3].$regs[2].$regs[1]";
} else {
    echo "Format invalid i datës: $date";
}
?>
```

Detyra 5:

Shembulli në vazhdim kontrollon validimin e emrit të user-it.

```
<?php
$var = "filan123";

if (ereg ("^[a-zA-Z0-9_]{6,16}$", $var))
{
    print "User-i '$var' është valid.";
}
else
{
    print "User-i '$var' nuk është valid.";
}
?>
```

Detyra 6:

Shembulli në vazhdim kontrollon validimin e numrit të telefonit.

```
<?php
$var = "044/111-222";
if (ereg ("^(04){1}[459]/[0-9]{3}\-[0-9]{3}$", $var))
{
    print "Numri tel: '$var' është valid.";
}
else
{
    print "Numri tel: '$var' nuk është valid.";
}
?>
```


Detyra 7:

Shembulli në vazhdim shfrytëzon përdorimin e funksionit **eregi()**.

```
<?php
$string = 'XYZ';
if (eregi('z', $string))
{
    echo "Stringu '$string' përmban ndonjë 'z' or 'Z'!";
}
else
{
    echo "Stringu '$string' nuk përmban ndonjë 'z' or 'Z'!";
}
?>
```

Detyra 8:

Shembulli në vazhdim shfrytëzon përdorimin e **[[:alpha:]]** që bashkon karaktere alfabetike.

```
<?php
$string = 'test';
if (eregi("[[:alpha:]]", $string))
{
    echo "Stringu '$string' përmban ndonjë shkronjë!";
}
else
{
    echo "Stringu '$string' nuk përmban ndonjë shkronjë!";
}
?>
```

Detyra 9:

Shembulli në vazhdim shfrytëzon përdorimin e klases **[[:alnum:]]** që bashkon numra ose karaktere alfanumerike.

```
<?php
$string = '123';
if (eregi("[[:alnum:]]", $string))
{
    echo "Stringu '$string' përmban ndonjë numër ose shkronjë!";
}
else
{

```

```
echo "Stringu '$string' nuk përmban ndonjë numër ose shkronjë!";
}
?>
```

Detyra 10:

Shembulli në vazhdim tregon përdorimin e funksionit **ereg_replace()**.

string **ereg_replace** (string \$pattern , string \$replacement , string \$string)

```
<?php
$string = "Ky është një test!";
print $string;
print "<br />";
echo str_replace(" është", " ishte", $string);
print "<br />";
echo ereg_replace(" është", " ishte", $string);
?>
```

Detyra 11:

Shembulli në vazhdim tregon përdorimin e funksionit **ereg_replace()**, i cili do të interpretoj numrin si një vlerë origjinale të karakterit:

```
<?php
/* Nuk do të funksionoj. */
$num = 4;
$string = "Ky string ka kater fjale.";
$string = ereg_replace('kater', $num, $string);
echo $string;

print "<br />";

/* Kjo do të funksionoj. */
$num = '4';
$string = "Ky string ka kater fjale.";
$string = ereg_replace('kater', $num, $string);
echo $string;
?>
```

Detyra 12:

Cilat nga shprehjet e rregullta në vijim përputh stringjet “zoot” dhe “zot”?

- A. z(oo)+t

- B. zo*t\$
- C. \$zo*t
- D. ^(zo)+t

Detyra 13:

Në këtë ushtrim është treguar se si punohet me shprehjet e rregullta në PHP.

```

<htm>
<head>
<title>Shprehjet e Rregullta</title>
</head>
<body>
<?php
$str = "Sot eshte dite me diell.";
print( "Test stringu është: '$str'<br /><br />" );

// thirrja e funksionit ereg te kerkimi për modelin 'Sot' në variablën str
if (ereg( "Sot", $str ) )
print( "Stringu 'Sot' është gjetur.<br />" );

// kerkimi për shprehjen 'Sot' në fillim të stringut
if ( ereg( "^Sot", $str ) )
print( "Stringu 'Sot' është gjetur në fillim të rreshtit.<br />" );

// kerkimi për modelin 'Sot' tek fundi i stringut
if (ereg( "Sot$", $str ) )
print( "Stringu 'Sot' është në fund të rreshtit.<br />" );

// kerkimi për ndonjë fjalë që mbaron me 'll'
if ( ereg( "[[:<:]]([a-zA-Z]*ll)[[:>:]]", $str, $match ) )
print( "Është gjetur fjala që mbaron me 'll': " . $match[1] . "<br />" );

// kerkimi për fjalët që fillojnë me 'd'
print( "Fjalët që fillojnë me 'd' janë gjetur: " );
while ( ereg( "(d[[:alpha:]]+)", $str, $match ) )
{
    print( $match[1] . " " );
    $str = ereg_replace( $match[1], "", $str );
}
print( "<br />" );
?>
</body>
</html>

```

Session Control në PHP

Detyra 1:

Çka është Cookie ?

Cookie shpesh përdoret të identifikoj një shfrytëzues. Cookie është një fajll i vogël që serveri e mbërthen në kompjuterin e shfrytëzuesit. Secilën herë që kompjuteri i njëjtë kërkon një faqe me browser, do të dërgohet edhe cookie gjithashtu. Me PHP, mund të krijoni dhe të rifitoni vlerat cookie.

Cookie është një pjesë e vogël e informacionit që skripta mund ta ruaj në makinën e klientit (client-side). Një Cookie mund të vendoset në makinën e shfrytëzuesit duke dërguar një header HTTP që përmban të dhëna si në vijim:

Set-Cookie: **NAME=VALUE**; [**expires=DATE**]; [**path=PATH**];

[**domain=DOMAIN_NAME**]; [**secure**]

expires = `time()+60*60*24*30` will set the cookie to expire in 30 days. If set to 0, or omitted, the cookie will expire at the end of the session (when the browser closes).

Kjo krijon një Cookie të quajtur **NAME** më vlerë **VALUE**. Të gjithë parametrat tjerë janë opcional. Fusha expires vendos një datë përtej secilës Cookie më nuk është valid. Path dhe Domain shfrytëzohen për me specifik URL-në ose URL-të për të cilat Cookie është i përshtatshëm, secure nënkupton që Cookie nuk do të dërgohet përmes një lidhje të thjeshtë HTTP. Kur browser-i lidhet në një URL, së pari kërkon cookie-et e ruajtura në lokacionin e browser-it. Nëse ndonjëra prej tyre janë relevante më URL-në me të cilën është e lidhur, ato do të transmetohen prapa tek serveri.

Detyra 2:

Në këtë ushtrim është treguar vendosja e Cookie-ve:

Vendosja e Cookie-ve në PHP mund të bëhet duke e shfrytëzuar funksionin **setcookie()**.

Sintaksa:

```
bool setcookie(string name [, string value [, int expire [,  
string path [, string domain [, int secure]]]])
```

pra:

```
setcookie(emri, vlera, skadimi, pathi, domeni);
```

Funksioni `setcookie()` duhet të paraqitet para tagut `<html>`.

Nëse vendosim një Cookie si:

```
setcookie('mycookie','value');
```

kur shfrytëzuesi viziton faqen tjetër në adresën tuaj (ose ringarkon faqen aktuale), ju do të keni qasje në cookie përmes `$_COOKIE['mycookie']` ose `$HTTP_COOKIE_VARS['mycookie']`.

Detyra 3:

Si bëhet fshirja e Cookie-es?

Fshirja e cookie-es bëhet duke thirrur funksionin **setcookie()** prapë me emër të njëjtë të cookie-et dhe më një datë që është skaduar.

Detyra 4:

Cilët janë hapat bazikë të përdorimit të sesionit?

1. Start a session (Fillo sesionin)
2. Register session variables (Regjistro variablat e sesionit)
3. Use session variables (Shfrytëzo variablat e sesionit)
4. Deregister variables and destroy a session (ç'regjistro variablat dhe përfundoje(shkatërroje) sesionin)

1. Start a session

Para se të filloni përdorimin e funksionalitetit të sesionit, ju nevojitet të filloni një sesion.

Fillimi i sesionit mund të bëhet në këto mënyra:

session_start(); -Ky funksion kontrollon nëse tashmë ekziston një sesion. Nëse jo, ai do ta krijoj një duke ofruar qasje në vargun **\$_SESSION**. Nëse sesioni ekziston, **session_start()** i lancon(loads) variablat e regjistruara të sesionit për përdorim.

Pra para se të ruani informacionet e shfrytëzuesit në sesioni PHP, së pari pra duhet të filloni sesionin. Funksioni **session_start()** duhet të paraqitet para tagut **<html>**

```
<?php
session_start();
?>
<html>
<body>
</body>
</html>
```

Kodi më lartë do të regjistrojë sesionin e shfrytëzuesit më server, ku ju lejon të filloni të ruani informacionet e shfrytëzuesit, dhe të caktoni një UID për atë session të shfrytëzuesit.

Mënyra e dytë, për me e fillua një sesion, është që PHP ta filloj automatikisht kur dikush viziton faqen tuaj. Kjo mund të bëhet duke përdorur aplikacionin **session.auto_start** në fajllin **php.ini**. Kjo metodë ka një disavantazh: Kur **auto_start** është aktivizuar, nuk mund ti përdorë objektet si variabla të sesioneve.

2. Register session variables (Regjistro variablat e sesionit)

Për me e krijua një variabël të sesionit, thjesht e vendosim një element në varg, si në vijim:

```
$_SESSION['myvar']=5;
```

Nëse përdorim versionin e vjetër të PHP, për një variabël që kalon nga një skript tek tjetra, do ta regjistrojmë atë me thirrjen: **session_register()**.

Variablen e sesionit të krijuar në këtë mënyrë do të ketë jetëgjatësi derisa sesioni të përfundoj ose ne vetë e ç'vendosim atë. P.sh për me e regjistru variablen **\$myvar**: **\$myvar=5; session_register("myvar");**

3. Use session variables (Shfrytëzo variablat e sesionit)

Për ti sjellur variablat e sessionit deri të rasti që ato mund të shfrytëzohen, së pari fillojmë sesionin duke përdorur **session_start()**. Madje mund ti qasemi variablave përmes **\$_SESSION**, p.sh **\$_SESSION['myvar']**.

Kur jemi duke e përdorur një objekt si variabël të sesionit, është e rëndësishme të përfshihet definimi i klasës para thirrjes **session_start()** për ti ringarkuar variablat e sesionit. Në këtë mënyrë PHP din si ta rindërtoj (rivendos) objektin e sesionit. Variablat e sesionit nuk mund të refuzohen nga të dhënat **GET** ose **POST** e cila është një veti e mirë sigurie.

Në anën tjetër duhet të jemi të kujdesshëm kur kontrollojmë nëse variablat e sesionit janë vendosur (për mes **isset()** ose **empty()**). Variablat mund të vendosen nga shfrytëzuesi përmes **GET** ose **POST**. Mund ta kontrollojmë variablen për të parë nëse është variabël e regjistruar e sesionit duke kontrolluar në **\$_SESSION**.

P.sh **if (isset(\$_SESSION['myvar']))...**

4. Deregister variables and destroy a session

Kur të përfundojmë punën me variabël të sesionit, mund ta çregjistrojmë atë. Kjo mund të bëhet direkt duke ç'vendosur elementin e duhur të vargut **\$_SESSION**, si në vijim:

```
unset($_SESSION['myvar']);
```

çregjistrimi i një variablave të vetme të sesionit:

Thërrasim funksionin **session_unregister()**

```
session_unregister(['myvar']);
```

Për ti çregjistruar të gjitha variablat përkatëse të sesionit **session_unset()**

Fshirja e session ID: **session_destroy()**

Detyra 5:

Në këtë ushtrim është treguar puna me sesione në PHP:

Page1.php

```
<?php
session_start();
?>
<html>
  <head>
    <title>Page 1...</title>
  </head>
  <body>
    <p>Puna me sesione: Page 1</p>
    <br>
    <?php
      $str_var = "Programimi ne Internet";
      print "Vlera e variables str_var: ".$str_var."<br>";
      $_SESSION['var'] = $str_var;
    ?>
    <br>
    <a href="Page2.php">Page 2</a>
  </body>
</html>
```

Page2.php

```
<?php
session_start();
?>
<p>Puna me sesione: Page 2</p><br/>

<?php
print "Vlera e var se sessionit eshte:". $_SESSION['var']."<br/>";
?>
<a href="Page3.php">Page 3</a>
```

Page3.php

```
<?php
session_start();
?>
<p>Puna me sesione: Page 3</p><br/>

<?php
print "Vlera e var se sessionit eshte: ". $_SESSION['var']."<br/>";
unset($_SESSION['var']);
session_destroy();
print "Vlera e var se sessionit eshte: ". $_SESSION['var']."<br/>";
?>
```

```
<a href="Page1.php">Page 1</a>
```

Detyra 6:

Në këtë ushtrim është treguar se si mund të kyçemi (login) në aplikacion dhe se si mund të ç'kyçemi (logout) duke përdorur sesionet në PHP:

HomePage.php

```
<?php
session_start();
?>
<html>
<head>
<title>Home Page...</title>
</head>
<body>
<h2>Miresevini ne Faqen kryesore...</h2>
<p>
<a href="Page1.php">Page1</a>
<a href="Page2.php">Page2</a>

</p>
<p>
<a href="LogIn.php">Kycuni</a>
</p>
</body>
</html>
```

Login.php

```
<?php
session_start();
include 'ValidateUser.php';
?>
<html>
<head>
<title>Login page...</title>
</head>
<body>
<p>Jepni te dhenat</p>

<form action="LogIn.php" name="frmLogIn" method="Post">
  User: <input type="text" name="user" /><br>
  Pass: <input type="password" name="pass" /><br>
  <input type="submit" value="Kycu" />
```



```

</form>
<?php
    $user = $_POST['user'];
    $pass = $_POST['pass'];
    control_User($user, $pass);
    if(isset($_SESSION['ValidUser']))
    {
        print '<br>Jeni kycur si: ' . $_SESSION['ValidUser'] . '<br/>';
        print "<a href='Page1.php'>Page1</a><br>";
        print "<a href='Page2.php'>Page2</a><br>";
        print "<a href='Logout.php'>Logout</a><br>";
    }
    else
    {
        if(isset($user))
        {
            print 'Nuk mund te kyceni...';
        }
    }
?>
</body>
</html>

```

ValidateUser.php

```

<?php
session_start();
function control_User($user, $pass)
{
    $u = 'user1';
    $p = '123';
    if(($u == $user) && ($p == $pass))
    {
        $_SESSION['ValidUser'] = $user;
    }
}
?>

```

Page1.php

```

<?php
session_start();
if(isset($_SESSION['ValidUser']))
{
    Print '<H1>Page1.....</H1>';
    print '<br/>';
    print 'Permbajta e faqes PAGE1...';
}

```

```
}  
else  
{  
    print 'Nuk mund te shihni permbajten e kesaj faqje.<br/>Se pari duhte te kyceni.<br/>';  
    print '<a href="login.php">Kycuni</a>';  
}  
?>
```

Page2.php

```
<?php  
session_start();  
  
    if(isset($_SESSION['ValidUser']))  
    {  
        Print '<H1>Page2.....</H1>';  
        print '<br/>';  
        print 'Permbajta e faqes PAGE2...';  
    }  
    else {  
        print 'Nuk mund te shihni permbajten e kesaj faqje.<br/>Se pari duhte te kyceni.<br/>';  
        print '<a href="login.php">Kycuni</a>';  
    }  
?>
```

Logout.php

```
<?php  
session_start();  
  
unset($_SESSION['ValidUser']);  
session_destroy();  
print 'Jeni ckycur me sukesë...<br/>';  
print '<a href="Homepage.php">Kthehuni ne faqen kryesore...</a>';  
?>
```

Users...

```
<?php  
session_start();  
function control_User($user, $pass)  
{  
  
    $u = 'user1';
```

```
$p = '123';
$con = mysql_connect("localhost","root","123");
mysql_select_db("bs");
$sql = "select Firstname, Lastname, Address, Username, Password from users where
        Username='$user' and Password='$pass'";
$res = mysql_query($sql);
$row = mysql_fetch_array($res);
print $sql."<br>";
if (mysql_num_rows($res) >0 )
{
    $_SESSION['ValidUser'] = $row["Firstname"]." ".$row["Lastname"];
}
}
?>
```

PHP dhe MySQL

Detyra 1:

Në këtë ushtrim është treguar konektimi i PHP-se me bazën e të dhënave në MySQL, regjistrimi i studentëve, lista e studentëve, editimi dhe fshirja e studentëve.

Krijimi bazës së të dhënave në MySQL me emrin **dbStudentet**: **CREATE DATABASE dbStudentet;**

Krijimi i tabelës **Studentet**:

```
CREATE TABLE Studentet (  
  NrIndeksit VARCHAR(10),  
  Emri VARCHAR(50),  
  Mbiemri VARCHAR(50),  
  Adresa VARCHAR(150),  
  PRIMARY KEY(NrIndeksit))
```

Krijoni **Menu.php** për paraqitjen e liqeve për navigim:

```
<nav>  
  <h1 align="center">Miresevini...</h1>  
  <a href="Add.php">Shto</a>  
  ||  
  <a href="List.php">Lista</a>  
  <br><br><br>  
</nav>
```

Krijoni **DBConfig.php** për konektim me MySQL:

```
<?php  
class DBConfig  
{  
  private $ServerName = "localhost";  
  private $DbName = "db";  
  private $Username = "root";  
  private $Password = "123";  
  
  function get_ServerName()  
  {  
    return $this->ServerName;  
  }  
  
  function get_DbName()  
  {  
    return $this->DbName;  
  }  
}
```

```

}

function get_Username()
{
    return $this->Username;
}

function get_Password()
{
    return $this->Password;
}
}
$db_config = new DBConfig();

$con = mysql_connect($db_config->get_ServerName(),
                    $db_config->get_Username(),
                    $db_config->get_Password());

mysql_select_db($db_config->get_DbName(), $con);
?>

```

Krijë **Add.php** për faqen e regjistrimit të studentëve:

```

<html>
<head>
    <title>Regjistrimi i studenteve</title>
</head>
<body>
<?php
include 'Menu.php';
?>
    <b>Te dhenat e studentit:</b>
    <form name="frmstd" action="Insert.php" method="post">
        <table>
            <tr>
                <td>NrIndeksit</td>
                <td><input name="nrIndeksit" type="text"/></td>
            </tr>
            <tr>
                <td>Emri</td>
                <td><input name="emri" type="text"/></td>
            </tr>
            <tr>
                <td>Mbiemri</td>
                <td><input name="mbiemri" type="text"/></td>
            </tr>
        </table>
    </form>

```

```
<td>Adresa</td>
<td><input name="adresa" type="text"/></td>
</tr>
</table>
<input type="submit" value="Dergo"/>
</form>
</body>
</html>
```

Krijë **Insert.php** për insertimin e të dhënave në MySQL:

```
<?php
include 'Menu.php';

$nrIndeksit = $_POST["nrIndeksit"];
$emri = $_POST["emri"];
$mbiemri = $_POST["mbiemri"];
$adresa = $_POST["adresa"];

include 'DBConfig.php';

if(isset($nrIndeksit) && $nrIndeksit!="")
{
    $sql_Insert = "INSERT INTO studentet
        (NrIndeksit,
        Emri,
        Mbiemri,
        Adresa)
        VALUES
        (
        '$nrIndeksit',
        '$emri',
        '$mbiemri',
        '$adresa'
        )";

    $res_Insert = mysql_query($sql_Insert);
    mysql_close();
}
echo "Studenti është regjistruar me sukses...";

?>
```

Krijë **List.php** për paraqitjen e listës së studentëve:

```
<?php
include 'Menu.php';
include 'DBConfig.php';

$sql_Select = "select NrIndeksit, Emri, Mbiemri, Adresa from studentet";
$res_Select = mysql_query($sql_Select);
$numRows = mysql_num_rows($res_Select);
print '<b>Numri total i studenteve: ' . $numRows . "</b><br/><br/>";

if($numRows)
{
    echo "<table>";
    echo "<tr style='color:white; background-color:#7AC234;'>
        <th>NrIndeksit</th>
        <th>Emri</th>
        <th>Mbiemri</th>
        <th>Adresa</th>
    </tr>";
    while($row = mysql_fetch_array($res_Select))
    {
        echo "<tr>
            <td>$row[NrIndeksit]</td>
            <td>$row[Emri]</td>
            <td>$row[Mbiemri]</td>
            <td>$row[Adresa]</td>
            <td><a href='Delete.php?id=$row[NrIndeksit]'>Delete</a></td>
            <td><a href='Edit.php?id=$row[NrIndeksit]'>Edit</a></td>
        </tr>";
    }
    echo "</table>";
}
echo '<br/>';
mysql_close();
?>
```

Krijë **Delete.php** për fshirjen e studentit të zgjedhur nga lista e studentëve:

```
<?php
include 'Menu.php';
$nrIndeksit = $_REQUEST['id'];
if(isset($nrIndeksit))
{
    $sql_Delete = "DELETE FROM studentet WHERE NrIndeksit ='$nrIndeksit'";
```

```

include 'DBConfig.php';

$res_Delete = mysql_query($sql_Delete);

echo "Studenti eshte fshij me sukses.<br/>";
//header("Location: List.php");
}
?>

```

Krijo **Edit.php** për shfaqjen e të dhënave të studentit për editim:

```

<?php
include 'DBConfig.php';
$nrIndeksit = $_GET['id'];

$sql_select = "select NrIndeksit, Emri, Mbiemri, Adresa from studentet where NrIndeksit='$nrIndeksit'";

$sql_rez = mysql_query($sql_select);
$row = mysql_fetch_array($sql_rez);
?>
<html>
<head>
<title>Editimi i studentit</title>
</head>
<body>
<?php
include 'Menu.php';
?>
<b>Te dhenat e studentit per editim:</b>
<form name="frmstd" action="Update.php" method="post">
<table>
<tr>
<td>NrIndeksit</td>
<td><input name="nrIndeksit" readonly type="text" value="<?php echo $row[NrIndeksit];?>" /></td>
</tr>
<tr>
<td>Emri</td>
<td><input name="emri" type="text" value="<?php echo $row[Emri]; ?>" /></td>
</tr>
<tr>
<td>Mbiemri</td>

```



```

        <td><input name="mbiemri" type="text" value="<?php echo $row[Mbiemri]; ?>" /></td>
    </tr>
    <tr>
        <td>Adresa</td>
        <td><input name="adresa" type="text" value="<?php echo $row[Adresa]; ?>" /></td>
    </tr>
</table>
<input type="submit" value="Dergo" />
</form>
</body></html>

```

Krijò **Update.php** për update të studentit të zgjedhur:

```

<?php
include 'Menu.php';
$nrIndeksit = $_POST["nrIndeksit"];
$emri = $_POST["emri"];
$mbiemri = $_POST["mbiemri"];
$adresa = $_POST["adresa"];

include 'DBConfig.php';

if(isset($nrIndeksit) && $nrIndeksit!="")
{
    $sql_Insert = "UPDATE studentet
        SET
            Emri ='$emri',
            Mbiemri ='$mbiemri',
            Adresa ='$adresa'
        WHERE NrIndeksit = '$nrIndeksit'";

    $res_Insert = mysql_query($sql_Insert);
    mysql_close();
    echo "Studenti eshte edituar me sukesse...";
}
?>

```

Detyra 2:

Në këtë ushtrim është treguar se si mund të konfigurohen të dhënat për konektim me MySQL, në fajllin **php.ini**

Hapë fajllin **php.ini** dhe konfiguro të dhënat:

mysql.default_host = localhost	-konfigurimi i serverit (hostname)
mysql.default_user = root	-konfigurimi i user-it për
mysql.default_password = 123	-konfigurimi i password-it
mysql.connect_timeout = 60	-konfigurimi i timeout për lidhje me MySQL

Pastaj ristarto Apache.

Testo lidhjen me MySQL duke përdorur funksionin `mysql_connect()`:

```
<?php
$link = mysql_connect() or die(mysql_error());
$result = mysql_query("SELECT SESSION_USER(), CURRENT_USER();");
$row = mysql_fetch_row($result);
echo "SESSION USER: ", $row[0], "<br>\n";
echo "CURRENT USER: ", $row[1], "<br>\n";
?>
```

Detyra 3:

Me poshtë janë paraqitur disa MySQL funksione:

- [mysql_affected_rows](#) — Get number of affected rows in previous MySQL operation
- [mysql_client_encoding](#) — Returns the name of the character set
- [mysql_close](#) — Close MySQL connection
- [mysql_connect](#) — Open a connection to a MySQL Server
- [mysql_create_db](#) — Create a MySQL database
- [mysql_data_seek](#) — Move internal result pointer
- [mysql_db_name](#) — Get result data
- [mysql_db_query](#) — Send a MySQL query
- [mysql_drop_db](#) — Drop (delete) a MySQL database
- [mysql_errno](#) — Returns the numerical value of the error message from previous MySQL operation
- [mysql_error](#) — Returns the text of the error message from previous MySQL operation
- [mysql_escape_string](#) — Escapes a string for use in a `mysql_query`
- [mysql_fetch_array](#) — Fetch a result row as an associative array, a numeric array, or both
- [mysql_fetch_assoc](#) — Fetch a result row as an associative array
- [mysql_fetch_field](#) — Get column information from a result and return as an object
- [mysql_fetch_lengths](#) — Get the length of each output in a result
- [mysql_fetch_object](#) — Fetch a result row as an object
- [mysql_fetch_row](#) — Get a result row as an enumerated array
- [mysql_field_flags](#) — Get the flags associated with the specified field in a result
- [mysql_field_len](#) — Returns the length of the specified field
- [mysql_field_name](#) — Get the name of the specified field in a result
- [mysql_field_seek](#) — Set result pointer to a specified field offset
- [mysql_field_table](#) — Get name of the table the specified field is in
- [mysql_field_type](#) — Get the type of the specified field in a result
- [mysql_free_result](#) — Free result memory
- [mysql_get_client_info](#) — Get MySQL client info
- [mysql_get_host_info](#) — Get MySQL host info
- [mysql_get_proto_info](#) — Get MySQL protocol info
- [mysql_get_server_info](#) — Get MySQL server info

[mysql_info](#) — Get information about the most recent query
[mysql_insert_id](#) — Get the ID generated in the last query
[mysql_list_dbs](#) — List databases available on a MySQL server
[mysql_list_fields](#) — List MySQL table fields
[mysql_list_processes](#) — List MySQL processes
[mysql_list_tables](#) — List tables in a MySQL database
[mysql_num_fields](#) — Get number of fields in result
[mysql_num_rows](#) — Get number of rows in result
[mysql_pconnect](#) — Open a persistent connection to a MySQL server
[mysql_ping](#) — Ping a server connection or reconnect if there is no connection
[mysql_query](#) — Send a MySQL query
[mysql_real_escape_string](#) — Escapes special characters in a string for use in an SQL statement
[mysql_result](#) — Get result data
[mysql_select_db](#) — Select a MySQL database
[mysql_set_charset](#) — Sets the client character set
[mysql_stat](#) — Get current system status
[mysql_tablename](#) — Get table name of field
[mysql_thread_id](#) — Return the current thread ID
[mysql_unbuffered_query](#) — Send an SQL query to MySQL without fetching and buffering the result rows.

Ajax

Detyra 1:

Në këtë ushtrim është treguar se si mund të kërkojmë lëndet që janë ruajtur në një varg të definuar në PHP duke përdorur ajax.

Krijë fajllin **KerkoLendet.php** që duhet të përmbaj një fushë tekst për të shkruar emrin e lëndës:

```
<html>
<head>
  <title>Ajax dhe PHP</title>
  <script type="text/javascript" src="jsLendet.js">
  </script>
</head>
<body>
  Lenda:<input type="text" id="txtLenda" onKeyUp="kerkoLendet(this.value)"/>
  <p>
    <br/>
    <br/>
    <br/>
    Rezultati: <div id="txtRez"></div>
  </p>
</body>
</html>
```

Krijë javascript fajllin me emër **jsLendet.php**:

```
// JavaScript Document
var xmlhttp;

//Funksioni kerkoLendet() ekzekutohet cdo herë që karakteri futet në "txtLenda".
function kerkoLendet(str)
{
  //Thërret f-onin createRequest() të krijoj një object XMLHttpRequest
  xmlhttp = createRequest();
  if(xmlhttp==null)
  {
    alert("Nuk mund te krijoni kerkesen");
    return;
  }

  //Definon URL (emrin e fajllit) të dërgoj në server
  var url = "http://localhost/Detyrat/Ajax/LendetServer.php";
  //Shton një parametër (q) tek URL me përmbajtje të fushës hyrëse
```

```
url = url+"?q="+str;

//Hap objektin XMLHttpRequest me URL-në e dhënë
xmlhttp.open("GET",url,true);
//Dërgon një kërkesë HTTP tek server
xmlhttp.send(null);
//Sa herë që ndryshon vetia readyState, f-oni stateChange() do të ekzekutohet
xmlhttp.onreadystatechange = displayRes;
}

//Ky funksion ekzekutohet sa herë që gjendja e objektit XMLHttpRequest ndryshon.
function displayRes()
{
    //alert(xmlhttp.readyState);
    //Kur gjendja ndryshon në 4("komplet") dhe status=200,
    //përmbajtja e txtRez mbushet me tekstin përgjigje.
    if(xmlhttp.readyState==4)
    {
        if(xmlhttp.status==200)
        {
            document.getElementById("txtRez").innerHTML = xmlhttp.responseText;
        }
    }
}

//ky funksion krijon objekte të ndryshme XMLHttpRequest për browser të ndryshëm
function createRequest()
{
    try
    {
        // kodi për IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    }
    catch(ex)
    {
        try
        {
            // kodi për IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(ex)
        {
            xmlhttp = null;
        }
    }
}
```

```
return xmlhttp;  
}
```

Krijo php fajllin me emër **LendetServer.php** për kërkim të lendëve:

```
<?php  
//Mbush vargun me emra  
$a[]="Gjuhe Programuese";  
$a[]="Bazat e te Dhenave";  
$a[]="Programimi ne Internet";  
$a[]="Programimi i Orientuar ne Objekte";  
$a[]="Programimi i avancuar";  
$a[]="Matematika 1";  
$a[]="Matematika 2";  
$a[]="Matematika 3";  
//merr parametrin q nga URL-ja  
$q=$_GET["q"];  
  
//shiko per te gjitha informatat nga vargu nese gjatesi e 'q>0'  
if (strlen($q) > 0)  
{  
    $informata="";  
    for($i=0; $i<count($a); $i++)  
    {  
        if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q))))  
        {  
            if ($informata=="")  
            {  
                $informata=$a[$i];  
            }  
            else  
            {  
                $informata=$informata.", ".$a[$i];  
            }  
        }  
    }  
}  
  
if ($informata == "")  
{  
    $rezultati=" ska propozime...";  
}  
else  
{  
    $rezultati=$informata;  
}
```

```
//paraqit rezultatin  
echo $rezultati;  
?>
```

Detyra 2:

Në këtë ushtrim është treguar se si përmes ajax-it duke e ekzekutuar php skriptën mund të kërkojmë studentët që janë të regjistruar në **MySQL**:

Krijohet fajllin **KerkoStudentin.php** që duhet të përmbajë një fushë tekst për të shkruar numrin e indeksit të studentit:

```
<html>  
  <head>  
    <title>Ajax dhe PHP</title>  
    <script type="text/javascript" src="jsStudentet.js">  
    </script>  
  </head>  
  <body>  
    NrIndeksit:<input type="text" id="txtNrIndeksit" onKeyUp="kerkoStudentin(this.value)"/>  
    <p>  
      <br/>  
      <br/>  
      <br/>  
      Rezultati: <div id="txtRez"></div>  
    </p>  
</body>  
</html>
```

Krijohet javascript fajllin me emër **jsStudentet.php**:

```
// JavaScript Document  
var xmlhttp;  
  
//Funksioni kerkoLendet() ekzekutohet cdo herë që karakteri futet në "txtLenda".  
function kerkoStudentin(str)  
{  
  //Thërret f-onin createRequest() të krijoj një object XMLHttpRequest  
  xmlhttp = createRequest();  
  if(xmlhttp==null)  
  {  
    alert("Nuk mund të krijoni kërkesën");  
    return;  
  }  
  
  //Definon URL (emrin e fajllit) të dërgoj në server  
  var url = "http://localhost/Detyrat/Ajax_1/Studentet.php";
```

```
//Shton një parametër (q) tek URL me përmbajtje të fushës hyrëse
url = url+"?q="+str;

//Hap objektin XMLHttpRequest me URL-në e dhënë
xmlhttp.open("GET",url,true);
//Dërgon një kërkesë HTTP tek server
xmlhttp.send(null);
//Sa herë që ndryshon vetia readyState, f-oni stateChange() do të ekzekutohet
xmlhttp.onreadystatechange = displayRes;
}

//Ky funksion ekzekutohet sa herë që gjendja e objektit XMLHttpRequest ndryshon.
function displayRes()
{
    //alert(xmlhttp.readyState);
    //Kur gjendja ndryshon në 4("komplet") dhe status=200,
    //përmbajtja e txtRez mbushet me tekstin përgjigje.
    if(xmlhttp.readyState==4)
    {
        if(xmlhttp.status==200)
        {
            document.getElementById("txtRez").innerHTML = xmlhttp.responseText;
        }
    }
}

//ky funksion krijon objekte të ndryshme XMLHttpRequest për browser të ndryshëm
function createRequest()
{
    try
    {
        // kodi për IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    }
    catch(ex)
    {
        try
        {
            // kodi për IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(ex)
        {
            xmlhttp = null;
        }
    }
    return xmlhttp;
}
```



```
}
```

Krijohet php fajllin me emër **Studentet.php** për kërkim të studentëve në bazën e të dhënave:

```
<?php
$con = mysql_connect("localhost","root","123");
mysql_select_db("db",$con);
$sql_Select = "select NrIndeksit, Emri, Mbiemri, Adresa from studentet where NrIndeksit like '$_GET[q]%'";
print $sql_Select;
$res_Select = mysql_query($sql_Select);
$num_rows = mysql_num_rows($res_Select);

if($num_rows)
{
    echo "<table>";
    echo "<tr style='color:white; background-color:#7AC234;'>
        <th color='white'>NrIndeksit</th>
        <th>Emri</th>
        <th>Mbiemri</th>
        <th>Adresa</th>
    </tr>";
    while($row = mysql_fetch_array($res_Select))
    {
        echo "<tr>
            <td>$row[NrIndeksit]</td>
            <td>$row[Emri]</td>
            <td>$row[Mbiemri]</td>
            <td>$row[Adresa]</td>
        </tr>";
    }
    echo "</table>";
}
echo '<br/>';
mysql_close();

?>
```

Detyra 3:

Në këtë ushtrim është treguar se si mund të lexohen të dhënat nga XML fajlli përmes Ajax-it:

Krijo html fajllin **librat.html**:

```
<html>
<head>
  <title>AJAX dhe XML</title>
  <script type="text/javascript" src="librat.js"></script>
</head>
<body>

  Server, shfaq librat me te pelqyer!
  <br/>
  <input type="button" name="btnShfaqLibrat" value="Shfaq librat" onclick="lexoXML()"/>
  <br/>
  <div id="myDivElement" />
</body>
</html>
```

Krijo javascript fajllin **librat.js**:

```
var xmlhttp;
// krjo nje instance XMLHttpRequest
//ky funksion krijon objekte të ndryshme XMLHttpRequest për browser të ndryshëm
function createRequest()
{
  try
  {
    // kodi per IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp = new XMLHttpRequest();
  }
  catch(ex)
  {
    try
    {
      // kodi per IE6, IE5
      xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    catch(ex)
    {
      xmlhttp = null;
    }
  }
  return xmlhttp;
}
```

```
// lexo fajllin nga serveri
function lexoXML()
{
    xmlHttp = createRequest();
    // vazhdo vetem nese xmlHttp nuk eshte i zbrazet
    if (xmlHttp)
    {
        // provo konektimin me server
        try
        {
            // fillo leximin e fajllit nga server-i
            xmlHttp.open("GET", "http://localhost/Detyrat/java10/librat.xml", true);
            xmlHttp.onreadystatechange = shfaqLibrat;
            xmlHttp.send(null);
        }
        // paraqit error-in ne rast te deshtimit
        catch (e)
        {
            alert("Nuk mund te lidheni me server:\n" + e.toString());
        }
    }
    else
    {
        alert("Nuk mund te krijoni kerkesen");
    }
}

// funksioni thirret kur gjendja e kerkeses HTTP nderrohet
function shfaqLibrat()
{
    // kur readyState eshte 4, jemi gati te lexojme pergjigjen nga serveri
    if (xmlHttp.readyState == 4)
    {
        // vazhdo vetem nese HTTP statusi eshte "OK"
        if (xmlHttp.status == 200)
        {
            try
            {
                // thirr funksioni handleServerResponse per pergjigje nga server-i
                Server_Response();
            }
            catch(e)
            {
                // paraqit error mersazhin
                alert("Gabim gjate leximit te pergjigjes: " + e.toString());
            }
        }
        else
    }
}
```

```

        {
            // parait statusin
            alert("Eshte shfaqur nje problem gjate nxjerrjes se te dhenave:\n" +
                xmlHttp.statusText);
        }
    }
}
// merr pergjigjen e pranuar nga server-i
function Server_Response()
{
    // lexo mesazhin nga server-i
    var xmlResponse = xmlHttp.responseXML;

    // merr elementet e XML dokumentit
    xmlRoot = xmlResponse.documentElement;

    //alert(xmlResponse.documentElement.getElementsByTagName("libri")[1].attributes["isbn"].nodeValue);
    //alert(xmlResponse.documentElement.getElementsByTagName("titulli").item(0).firstChild.data);
    //
    // merr vargjet me titujt dhe isbn e libres
    titleArray = xmlRoot.getElementsByTagName("titulli");
    isbnArray = xmlRoot.getElementsByTagName("libri");

    //gjeneron daljen HTML
    var html = "";

    // perseritet pergjat vargjeve dhe krjon nje strukture HTML
    for (var i=0; i<titleArray.length; i++)
        html += titleArray.item(i).firstChild.data +
            ", " + isbnArray[i].attributes["isbn"].nodeValue + "<br/>";

    // merr permbajtjen e elementit <div> ne faqe
    myDiv = document.getElementById("myDivElement");

    //paraqit daljen HTML
    myDiv.innerHTML = "Pergjigja e Server-it: <br />" + html;
}

```

Krijo xml fajllin **librat.xml** në të cilin ruhen librat:

```

<?xml version="1.0" encoding="iso-8859-1"?>
<librat>
  <libri isbn="1-904811-82-5">
    <titulli>
      Timoni nga Athina
    </titulli>
  </libri>

```

```

</titulli>
< autori viti_lindjes="1564">
  <emri>William</emri>
  <mbiemri>Shakespeare</mbiemri>
</ autori>
<viti>
  1607
</viti>
</libri>
<libri isbn="1-59059-392-8">
  <titulli>
    Lahuta e Malsis
  </titulli>
  < autori viti_lindjes="1871">
    <emri>Gjergj</emri>
    <mbiemri>Fishta</mbiemri>
  </ autori>
  <viti>
    1902
  </viti>
</libri>
</librat>

```

Detyra 4:

Në këtë ushtrim është treguar se si mund t'a gjenerojmë një RSS dokument përmes PHP, ku të dhënat për item-at lexohen nga baza e të dhënave MySQL:

Krijoni tabelën në MySQL për ruajtjen e artikujve:

```

create table lajmet (
  id int auto_increment,
  titulli text not null,
  autori text not null,
  permbajtja text not null,
  primary key(id))

```

Krijohet php fajllin **RSS.php**:

```

<? header('Content-type: text/xml'); ?>
<?php
$dbhost = "localhost"; //Server Name.
$dbname = "rss"; // Database Name
$dbuser = "root"; // Database Username
$dbpass = "123"; // Database Password

$conn = mysql_connect("$dbhost","$dbuser","$dbpass");// Connecting to Database
mysql_select_db($dbname) or die (mysql_error()); // Selecting Database

```

```

?>

<rss version="2.0">
//deklarimi i channel elementit - është përdorur për të
//përshkruar furnizuesin RSS (feed) -ka tri elemente fëmijë të rekomanduara
<channel>
    //definon titullin e kanalit
    <title>Programimi ne internet- Home Page</title>
    //definon hyperlink-un e kanalit
    <link>http://www.fiek.ProgInternet.com</link>
    //Përshkruan kanalën
    <description>Universiteti i Prishtinës</description>
    //Secili channel element mund të ketë një ose më shumë item element
    //Secili item element definon një artikull në RSS feed
<?
$sql = "SELECT * FROM lajmet order by id desc limit 5";
$result = mysql_query($sql);
while($row = mysql_fetch_assoc($result))
{
?>
<item>
    <title><?php echo $row['titulli']; ?></title>
    <author><?php echo $row['autori']; ?></author>
    <link>http://fiek.ProgInternet.com/lajmet.php?id=<?php echo $row['id'];
?></link>
</item>
<?php    }    ?>
</channel>
</rss>

```

Detyra 5:

Në këtë ushtrim është treguar se si mund të modifikohet një **xml** dokument përmes **php**-së, duke shtuar elemente të reja ose duke i modifikuar ato ekzistuese:

Krijoni xml fajllin **librat.xml**:

```

<?xml version="1.0" encoding="iso-8859-1"?>
<librat>
    <libri isbn="1-904811-82-5">
        <titulli>
            Timoni nga Athina
        </titulli>
        <autori viti_lindjes="1564">
            <emri>William</emri>
            <mbiemri>Shakespeare</mbiemri>

```

```

</autori>
<viti>
  1607
</viti>
</libri>
<libri isbn="1-59059-392-8">
  <titulli>
    Lahuta e Malsis
  </titulli>
  <autori viti_lindjes="1871">
    <emri>Gjergj</emri>
    <mbiemri>Fishta</mbiemri>
  </autori>
  <viti>
    1902
  </viti>
</libri>
</librat>

```

Krijoni html fajllin **AddHTML.html**:

```

<html>
<head>
  <title>Regjistro libra ne XML</title>
  <script type="text/javascript" src="ajax.js">
  </script>
</head>
<body>
  ISBN:<input type="text" id="isbn"/><br/>
  Titulli: <input type="text" id="titulli"/><br/>
  Emri Autorit:<input type="text" id="autori_emri"/><br/>
  Mbiemri Autorit:<input type="text" id="autori_mbiemri"/><br/>
  Viti Lindjes:<input type="text" id="autori_viti"/><br/>
  Viti:<input type="text" id="viti"/><br/>
  <input type="button" name="btnSaveData" value="Ruaj" onclick="SaveData()"/>
  <br/>
  <label id="lblmsg"/>
</body>
</html>

```

Krijoni javascript dokumentin **ajax.js**:

```

var xmlhttp;
function SaveData()
{
  var isbn = document.getElementById('isbn').value;

```

```
var titulli = document.getElementById('titulli').value;
var autori_emri = document.getElementById('autori_emri').value;
var autori_mbiemri = document.getElementById('autori_mbiemri').value;
var autori_viti = document.getElementById('autori_viti').value;
var viti = document.getElementById('viti').value;

//Thërret f-onin createRequest() të krijoj një object XMLHttpRequest
xmlhttp = createRequest();
if(xmlhttp==null)
{
    alert("Nuk mund te krijoni kerkesen");
    return;
}

//Definon URL (emrin e fajllit) të dërgoj në server
var url = "http://localhost/Detyrat/XMLUpdate/Add.php";
//Shton një parametër (q) tek URL me përmbajtje të fushës hyrëse
url = url+"?isbn="+isbn+"&titulli="+titulli+"&autori_emri="+autori_emri
   +"&autori_mbiemri="+autori_mbiemri+"&autori_viti="+autori_viti+"&viti="+viti;

//Hap objektin XMLHttpRequest me URL-në e dhënë
xmlhttp.open("GET",url,true);
//Dërgon një kërkesë HTTP tek server
xmlhttp.send(null);
//Sa herë që ndryshon vetia readyState, f-oni stateChange() do të ekzekutohet
xmlhttp.onreadystatechange = displayRes;
}
//ky funksion krijon objekte të ndryshme XMLHttpRequest për browser të ndryshëm
function createRequest()
{
    try
    {
        // kodi per IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    }
    catch(ex)
    {
        try
        {
            // kodi per IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch(ex)
        {
            xmlhttp = null;
        }
    }
    return xmlhttp;
}
```



```

}

//Ky funksion ekzekutohet sa herë që gjendja e objektit XMLHTTP ndryshon.
function displayRes()
{
    //Kur gjendja ndryshon në 4("komplet") dhe status=200,
    //përmbajtja e txtRez mbushet me tekstin përgjigje.
    if(xmlhttp.readyState==4)
    {
        if(xmlhttp.status==200)
        {
            document.getElementById("lblmsg").innerHTML = xmlhttp.responseText;
        }
    }
}
}

```

Krijoni php fajllin **Add.php**:

```

<?php
$isbn = $_GET['isbn'];
$titulli = $_GET['titulli'];
$ autori_emri = $_GET['autori_emri'];
$ autori_mbiemri = $_GET['autori_mbiemri'];
$ autori_viti = $_GET['autori_viti'];
$viti = $_GET['viti'];
// Interprets an XML file into an object
$xml = simplexml_load_file("librat.xml"); //Returns an object of class SimpleXMLElement with properties
//containing the data held within the XML document. On errors, it will return FALSE.
$xml = new SimpleXMLElement($xml->asXML()); //Represents an element in an XML document.
$libri = $xml->addChild("libri");
$libri->addAttribute("isbn",$isbn);
$libri->addChild("titulli",$titulli);
$libri->addChild("autori");
$libri->autori->addChild("emri",$autori_emri);
$libri->autori->addChild("mbiemri",$autori_mbiemri);
$libri->autori->addAttribute("viti_lindjes",$autori_viti);
$libri->addChild("viti",$viti);
// asXML -If the filename isn't specified, this function returns a string on success and FALSE on error. If the parameter is
//specified, it returns TRUE if the file was written successfully and FALSE otherwise
$xml -> asXML("librat.xml"); //Return a well-formed XML string based on SimpleXML element
echo '<br/>';
echo '<b>Te dhenat jane ruajtur me sukses!</b>';
echo '<br/>';
include 'LexoXML.php'; ?>

```

Krijioni php fajllin **LexoXML.php**

```
<?php
$file = "librat.xml";
$xml = simplexml_load_file($file);

for($i=0; $i<count($xml->libri); $i++)
{
echo "ISBN: ". $xml->libri[$i]->attributes()."<a href='Edit.php?isbn=". $xml->libri[$i]->attributes()->isbn.'">
Edito</a><br>";
echo "Titulli: ". $xml->libri[$i]->titulli."<br>";
echo "Autori: ". $xml->libri[$i]->autori->emri." ". $xml->libri[$i]->autori->mbiemri."<br>";
echo "Viti i lindjes se atuari: ". $xml->libri[$i]->autori->attributes()->viti_lindjes."<br>";
echo "Viti: ". $xml->libri[$i]->viti."<br>";
echo "-----<br/>";
}

?>
```

Krijioni php fajllin **Edit.php**

```
<?php
$file = "librat.xml";
$isbn = $_GET['isbn'];
$xml = simplexml_load_file($file) or die ("Unable to load XML file!");
//count($xml->libri);
for($i=0; $i<sizeof($xml->libri); $i++) {
    if($xml->libri[$i]->attributes()==$isbn)
    {
        $xml->libri[$i]->titulli="Titulli1";
        $xml->libri[$i]->autori->emri="autoriemri";
        $xml->libri[$i]->autori->mbiemri="autorimbiemri";
        $xml->libri[$i]->autori->attributes()->viti_lindjes="2018";
        $xml->libri[$i]->viti="2015";
    }
    file_put_contents($file, $xml->asXML() ); //Write a string to a file (identical to calling fopen(), fwrite() and
    //fclose()). If filename does not exist, the file is created. Otherwise, the existing file is overwritten,
    //unless the FILE_APPEND flag is set.
} ?>
```

API-të

Google Maps JavaScript API V3

V3: Zgjidhja për aplikacionet e hartave për desktop dhe paisjet mobile

Google Maps Javascript API ju lejon ta mbërtheni Google Maps në web faqet e juaja. Versioni i 3-të i këtij API është dizajnuar në mënyrë specifike të jetë më i shpejtë dhe më i aplikueshëm tek paisjet mobile, si dhe aplikacionet shfletues të desktopëve tradicional.

API ofron një numër shërbimesh për manipulimin e hartave (sikurse në web faqen <http://maps.google.com>) dhe shtimin e përmbajtjes në hartë përmes një shumëllojshmërie të shërbimeve, duke ju lejuar të krijoni aplikacione të hartave të fuqishme në web sajtin tuaj.

JavaScript Maps API V3 është shërbim falas në dispozicion për çfarëdo web sajti që është falas për konsumatorë.

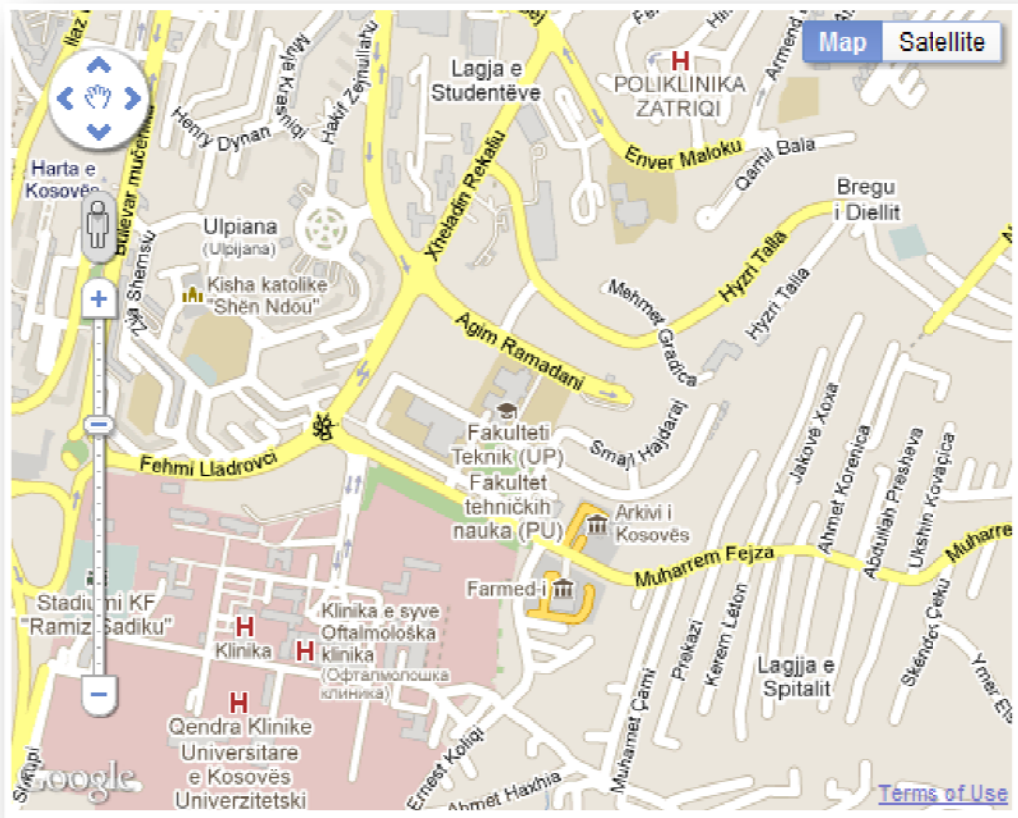
Detyra 1:

Më poshtë është dhënë një shembull i Google Maps API:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<style type="text/css"> html { height: auto } body { height: auto } </style>

<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false">
  //gjuha: src="http://maps.google.com/maps/api/js?sensor=false&language=ja" ose ar
</script>

<script type="text/javascript">
function showMap() {
  var latlng = new google.maps.LatLng(42.648117,21.166599);
  var myOptions = {
    zoom: 15,
    center: latlng,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  var map = new google.maps.Map(document.getElementById("div_map"),
    myOptions);
}
</script>
</head>
<body onload="showMap()">
  <div id="div_map" style="width: 500px; height: 400px;"></div>
</body>
</html>
```

Rezultati:**Detyra 2:**

Krijoni një marker duke përdorur **google.maps.Marker**:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<style type="text/css"> html { height: auto } body { height: auto } </style>

<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false">
</script>

<script type="text/javascript">
function showMap() {
    var latLng = new google.maps.LatLng(42.648117,21.166599);
    var myOptions = {
        zoom: 15,
        center: latLng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }
    var map = new google.maps.Map(document.getElementById("map"), myOptions);
    var marker = new google.maps.Marker({
        position: latLng,
        title: "Marker"
    });
    map.addMarker(marker);
}
```

```

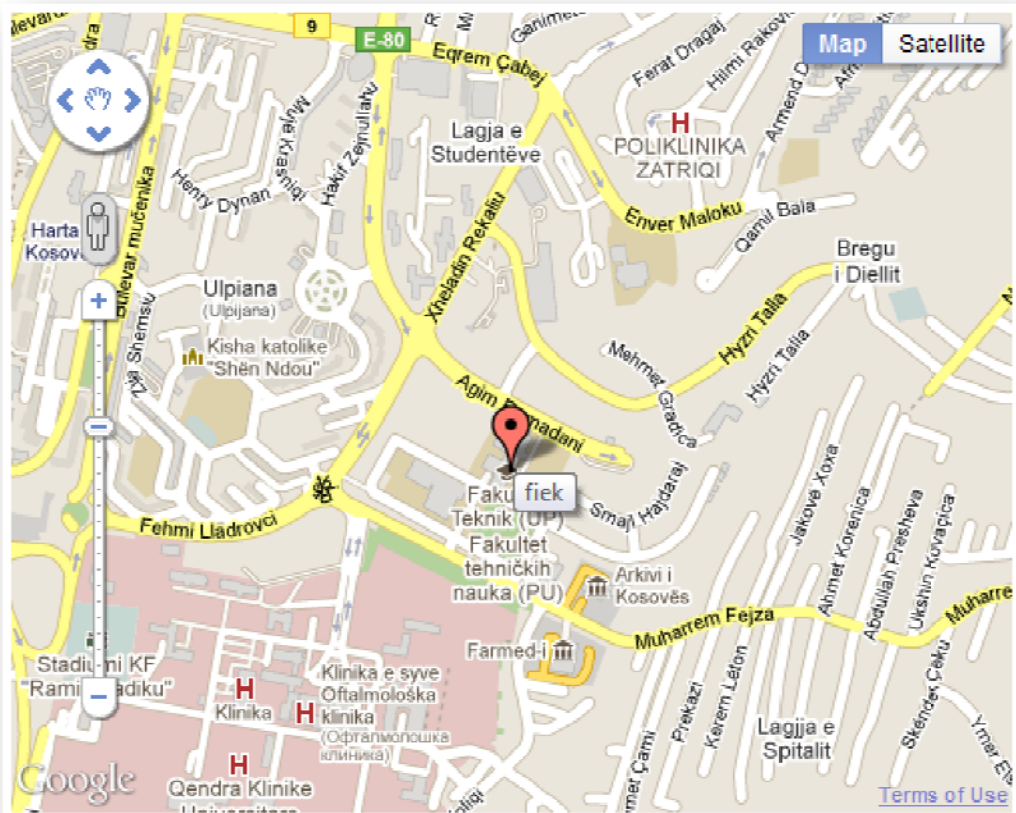
});
var map = new google.maps.Map(document.getElementById("div_map"),
    myOptions);

var marker = new google.maps.Marker({
    map: map,
    position: map.getCenter(),
    title:"fiek"
});

}
</script>
</head>
<body onload="showMap()">
  <div id="div_map" style="width: 500px; height: 400px;"></div>
</body>
</html>

```

Rezultati:



Detyra 3:

Krijoni një info window duke përdorur **google.maps.InfoWindow**:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css"> html { height: auto } body { height: auto } </style>

    <script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false">
    </script>

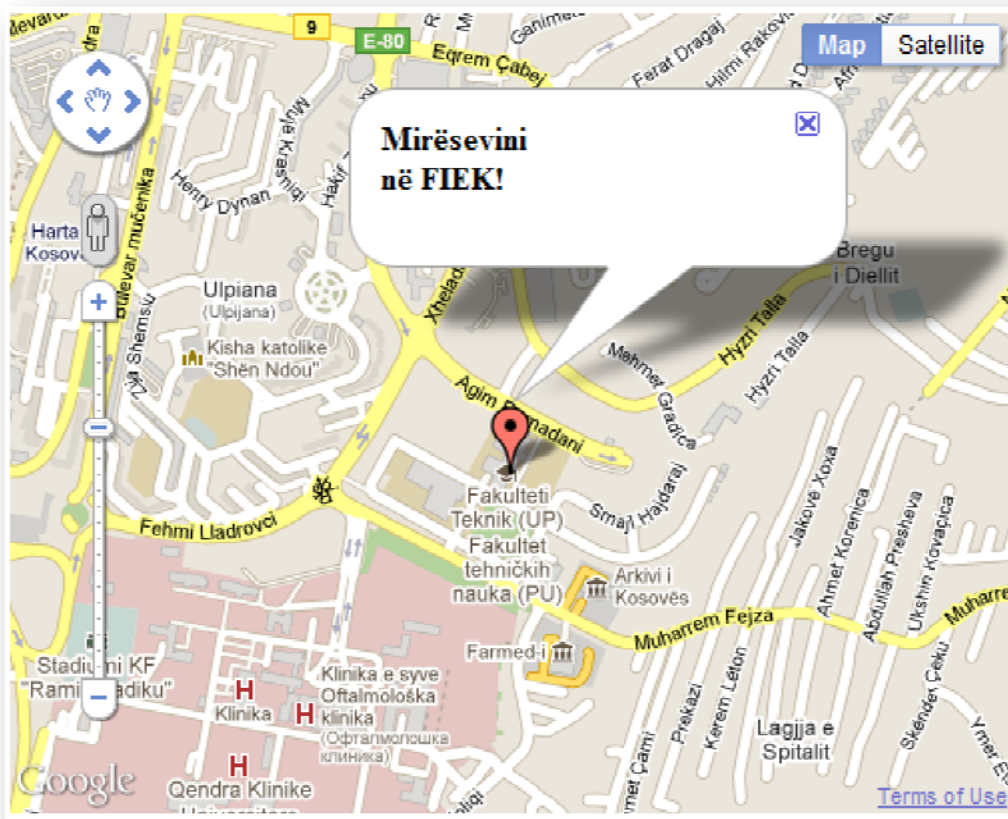
    <script type="text/javascript">
      function showMap() {
        var latlng = new google.maps.LatLng(42.648117,21.166599);
        var myOptions = {
          zoom: 15,
          center: latlng,
          mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var map = new google.maps.Map(document.getElementById("div_map"),
          myOptions);

        var marker = new google.maps.Marker({
          map: map,
          position: map.getCenter(),
          title:"fiek"
        });

        var infowindow = new google.maps.InfoWindow();

        infowindow.setContent('<b>Mir&euml; sevini <br/> n&euml; FIEK!</b>');

        google.maps.event.addListener(marker, 'click', function()
        {
          infowindow.open(map, marker);
        } )
      }
    </script>
  </head>
  <body onload="showMap()">
    <div id="div_map" style="width: 500px; height: 400px;"></div>
  </body>
</html>
```

Rezultati:**Detyra 4:**

Duke përdorur **google.maps.Polyline** dizajnoni një linjë me ngjyrë të kuqe prej koordinatave (42.648117, 21.166599)-FIEK deri te koordinatat (42.651779,21.165226) - Qendra e Studentëve:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css"> html { height: auto } body { height: auto } </style>

    <script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false">
    </script>

    <script type="text/javascript">
      function showMap() {
        var latlng = new google.maps.LatLng(42.648117,21.166599);
        var myOptions = {
```



```
        zoom: 15,
        center: latlng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    });
    var map = new google.maps.Map(document.getElementById("div_map"),
    myOptions);

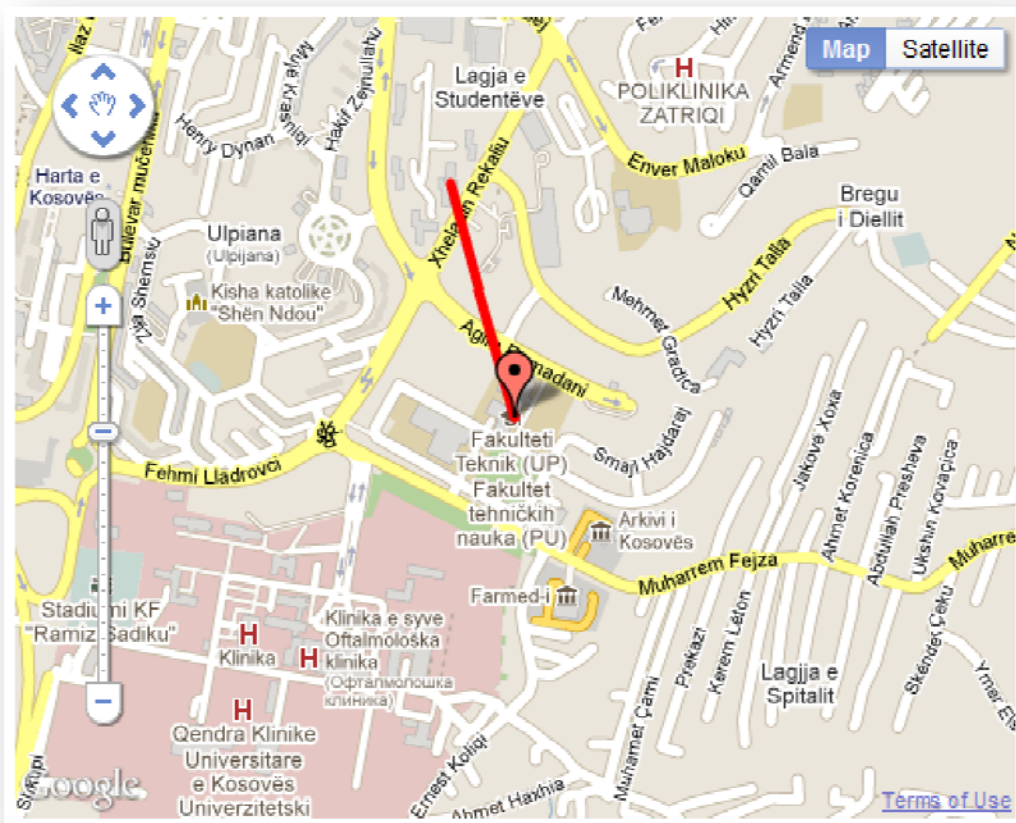
    var marker = new google.maps.Marker({
        map: map,
        position: map.getCenter(),
        title:"fiek"
    });

    var infowindow = new google.maps.InfoWindow();

    infowindow.setContent('<b>Mir&euml; sevini <br/> n&euml; FIEK!</b>');

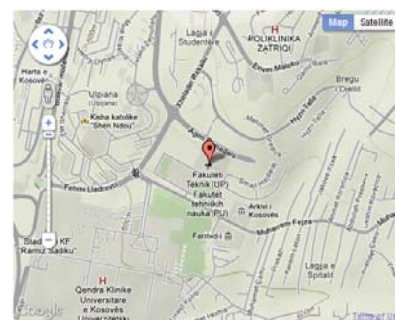
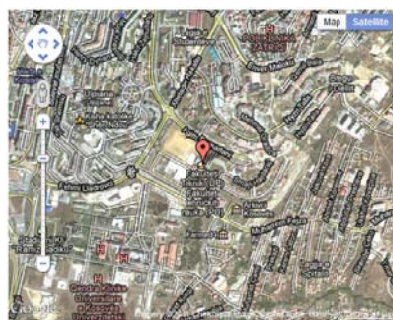
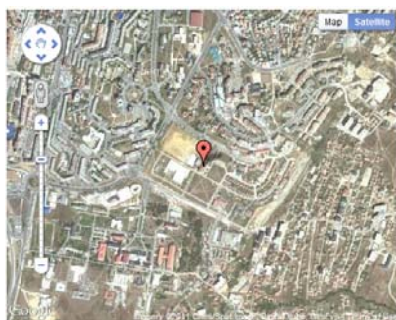
    google.maps.event.addListener(marker, 'click', function()
    {
        infowindow.open(map, marker);
    } )
    var koordinatat = [
        new google.maps.LatLng(42.648117,21.166599),
        new google.maps.LatLng(42.651779,21.165226)
    ];
    var _path = new google.maps.Polyline({
        path: koordinatat,
        strokeColor: "#FF0000",
        strokeWeight: 5
    });
    _path.setMap(map);

    }
</script>
</head>
<body onload="showMap()">
    <div id="div_map" style="width: 500px; height: 400px;"></div>
</body>
</html>
```


Rezultati:**Detyra 5:**

Provoni tipe të hartave të ndryshme:

- **ROADMAP**
- **SATELLITE**
- **HYBRID**
- **TERRAIN**



LinkedIn

LinkedIn – është një rrjet social që kryesisht përdoret për rrjete profesionale, i themeluar në Dhjetor 2002 dhe i lancuar në Maj 2003. LinkedIn raporton më shumë se 100 milion shfrytëzues të regjistruar (05.2011).

Hapat e implementimit të LinkedIn në ueb sajtin tonë:

1. Caktoni emrin e domenit

Për arsye të sigurisë, ju duhet të informoni LinkedIn për emrin e domenit që do ta përdorni. Nëse jeni duke përdorur një port jo-standard (tjetër prej 80 për http dhe https 443) ju duhet të shkruani: `http://www.example.com:8080`.

2. Merreni API key

Pas zgjedhjes së domenit të juaj, regjistroni një API key. Kur të regjistroheni për çelës, sigurohuni të specifikoni domenin e zgjedhur më lartë për fushën '**JavaScript API Domain**'. API secret key- nuk do të ju nevojitet për **JavaScriptAPI**.

3. Shto framework-un në faqen tuaj

Tani mund të importojmë JavaScript framework-un në aplikacion. Krijë një HTML fajll dhe shto skriptën e mëposhtme në **<head>** tagun.

```
<script type="text/javascript" src="http://platform.linkedin.com/in.js">
  api_key: bKje-vlCxmKh7nPRKli9Bv40Se10s4hnE8GuEBGXtSxAOI7dLa7GELIf5zMn21-H
</script>
```

Detyra 1:

Në ueb sajtin e juaj shtoni butonin për kyçje në LinkedIn:

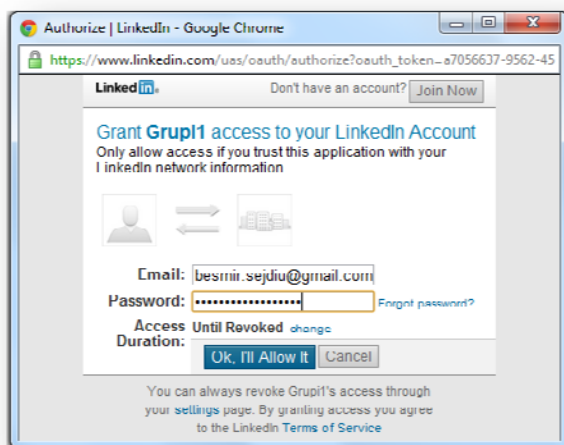
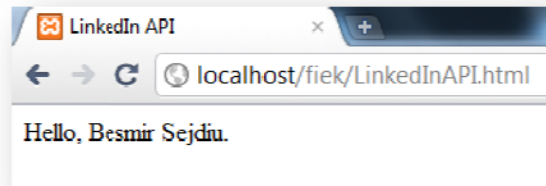
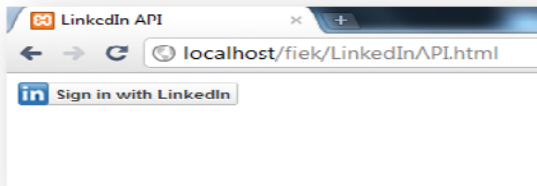
```
<html>
<head>
<title>LinkedIn API</title>

<script type="text/javascript" src="http://platform.linkedin.com/in.js">
  api_key: bKje-vlCxmKh7nPRKli9Bv40Se10s4hnE8GuEBGXtSxAOI7dLa7GELIf5zMn21-H
</script>

</head>
<body>
  <script type="in/Login">
    Hello, <?js= firstName ?> <?js= lastName ?>.
  </script>
</body>
```

</html>

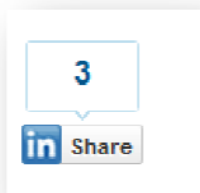
Rezultati:



Detyra 2:

Në ueb sajtin e juaj shtoni Share Button:

```
<script src="http://platform.linkedin.com/in.js" type="text/javascript"></script>
<script                                type="IN/Share"                                data-
url="http://www.linkedin.com/shareArticle?mini=true&url=http://localhost/Detyrat/API/Det1.html&title=Progr
amimi%20Ne%20Internet&summary=Programimi%20Ne%20Internet&source=" data-counter="top">
</script>
```



The Share URL

All parameter values must be URL encoded.

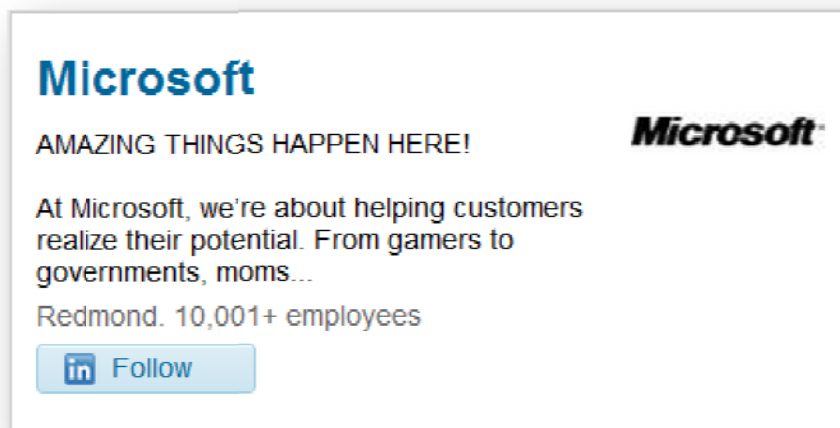
`http://www.linkedin.com/shareArticle?mini=true&url={articleUrl}&title={articleTitle}&summary={articleSummary}&source={articleSource}`

Parameter	Required	Recommended	Character Limit	Description
mini	Yes	Yes	4	Must always be <i>true</i>
url	No	Yes	1024	The permanent link to the article. Must be URL encoded
title	No	Yes	200	The title of the article. Must be URL encoded
source	No	Yes	200	The source of the article. Must be URL encoded. Example: <i>Wired Magazine</i>
summary	No	Yes	256	A brief summary of the article. Must be URL encoded. Longer titles will be truncated gracefully with ellipses.

Detyra 3:

Build a Company Profile plugin:

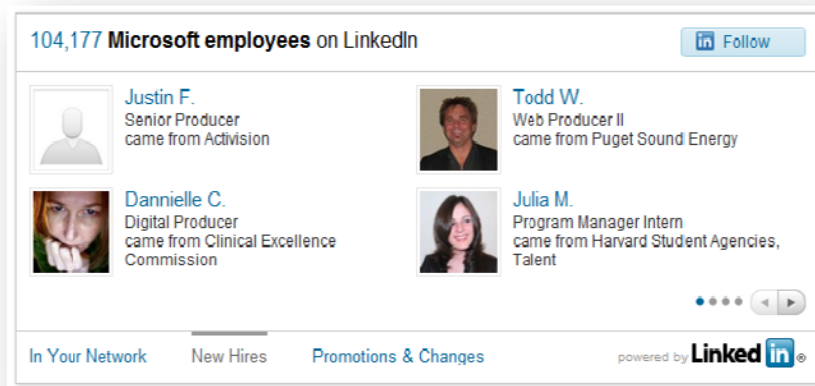
```
<script src="http://platform.linkedin.com/in.js" type="text/javascript"></script>
<script type="IN/CompanyProfile" data-id="1035" data-format="inline"></script>
```



Detyra 4:

Build a Company Insider plugin:

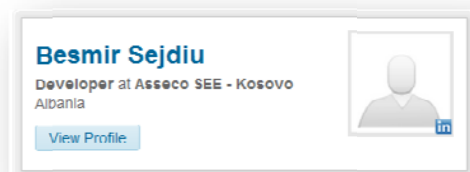
```
<script src="http://platform.linkedin.com/in.js" type="text/javascript"></script>
<script type="IN/CompanyInsider" data-id="1035"></script>
```



Detyra 5:

Build a Member Profile plugin:

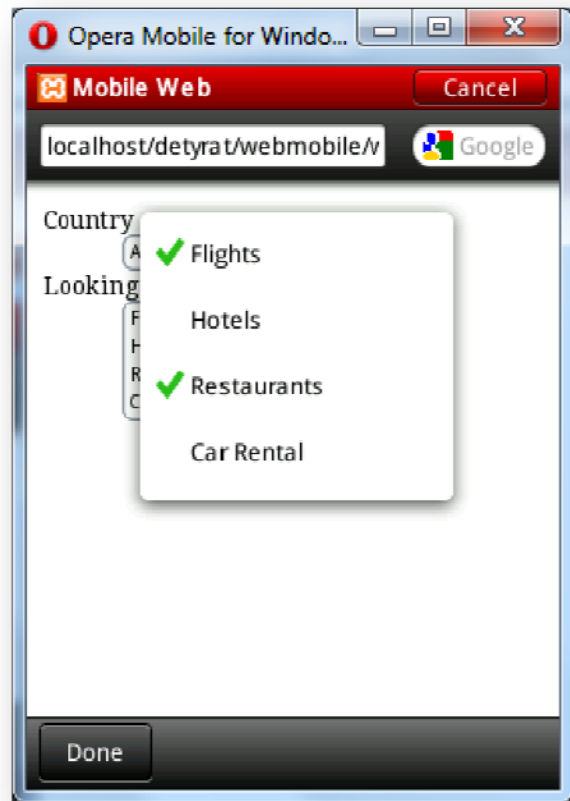
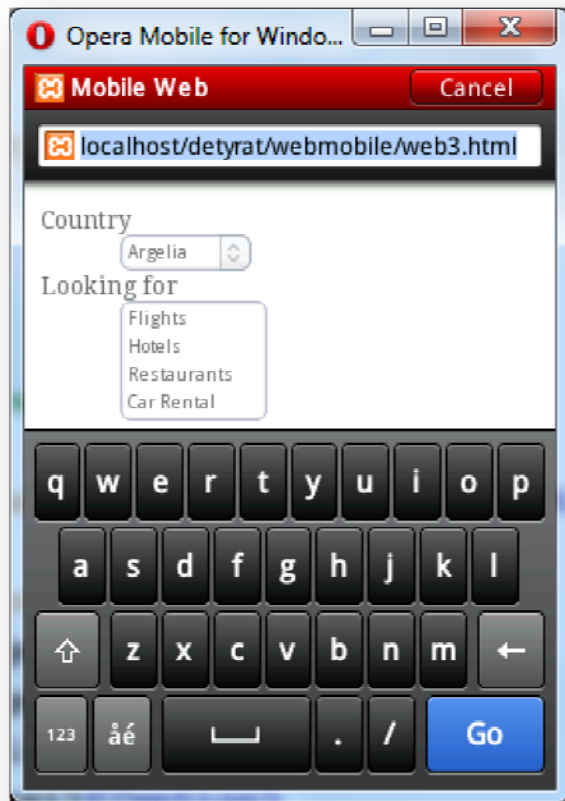
```
<script src="http://platform.linkedin.com/in.js" type="text/javascript"></script>
<script type="IN/MemberProfile" data-id="http://www.linkedin.com/pub/besmir-sejdiu/30/a59/64b" data-format="inline"></script>
```



Mobile Web

Detyra 1:

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Mobile Web</title>
  </head>
  <body>
    <form action="formAction" method="post">
      <dl>
        <dt><label for="country">Country</label></dt>
        <dd>
          <select name="country">
            <option>Argelia</option>
            <option>Argentina</option>
            <option>Bolivia</option>
            <option>Brazil</option>
          </select>
        </dd>
        <dt><label for="filter">Looking for</label></dt>
        <dd>
          <select name="filter" multiple="multiple">
            <option>Flights</option>
            <option>Hotels</option>
            <option>Restaurants</option>
            <option>Car Rental</option>
          </select>
        </dd>
      </dl>
    </form>
  </body>
</html>
```



Detyra 2

```
<?xml version="1.0" ?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.2//EN"
"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile12.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>First Template</title>
  </head>
  <body>
    <h1>First Template</h1>
    <h2>Programming the Mobile Web</h2>
    <p>Welcome to the first template of this book</p>
    <p>It <strong>should work</strong> in every mobile browser in the market</p>
    <ol>
      <li><a href="http://m.yahoo.com" accesskey="1">Yahoo!</a></li>
      <li><a href="http://m.google.com" accesskey="2">Google</a></li>
      <li><a href="http://m.bing.com" accesskey="3">Bing</a></li>
    </ol>
    <p></p>
```

```
</body>  
</html>
```

