

SARE ZERBITZUAK ETA APLIKAZIOAK

3. PRAKTIKA OROKORRA: WEB APLIKAZIO BATEN GARAPENA: LIBURUTEGIA

Beñat Sarasua, Iñaki Urdangarin eta Manex Atxa.

AURKIBIDEA

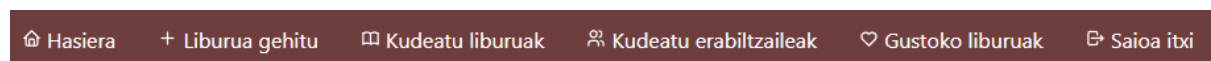
- Sarrera eta webaren deskribapena
- Irakaslearentzako argibideak
- Teknologia eta lengoaiak
 - HTML
 - CSS
 - JavaScript
 - PHP
 - DTD
 - XML
 - Datubasea
- Ondorioak

SARRERA ETA WEBAREN DESKRIBAPENA

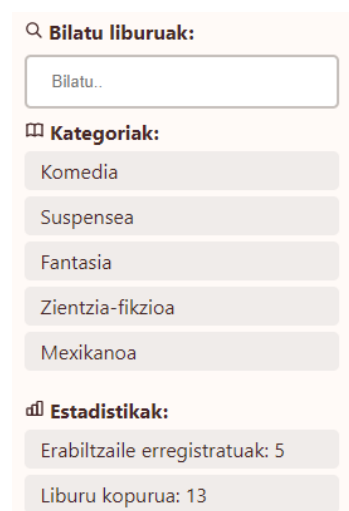
Hirugarren praktika honetan, web aplikazio bat garatzea egokitu zaigu. Web aplikazioaren gaia eta izango dituen funtzionalitateak guk hautatu ditugu, beti ere kontuan izanik erabiltzaileak aukera izan behar duela zerbitzariari informazioa bidaltzeko formularioak erabiliz. Ondoren, bidalitako informazio hori kontsultatzeko eta aldatzeko aukera ere eman behar dugu.

Oinarri horiek kontuan izanda, liburutegi bat garatzea otu zaigu. Liburutegian erabiltzaileak zerbitzarian gordeta dauden liburuak eta hauen informazioa lortu ditzake eta erabiltzailea administratzailea izanez gero, liburuak igotzeko aukera ere izango du liburu zehatz horren informazioa eskatuko dion formulario bat beteta (egilea, agitaratze-data, hizkuntza...).

Formularioen erabileraz gain, beste aukera batzuez ere osatu dugu orrialdea. Bi erabiltzaile mota izanda (erabiltzaile arrunta eta adminstratzailea), erregistratzeko eta logeatzeko aukerak sortu ditugu. Behin logeatuta, bi erabiltzaile motek gustoko liburuaren zerrenda bat izango dute eta, administratzaileek soilik, liburu berriak gehitzeko, igota dauden liburuaren informazioa editatzeko eta erregistratuta dauden erabiltzaileak kudeatzeko (administratzaile egin, kontuak ezabatu...) aukerak izango dituzte.



Azkenik, orrialdea gehiago osatze aldera, liburuaren bilaketa egiteko modu desberdinak implementatu ditugu. Batetik, kategoriak bilatu daitezke liburuak eta, bestetik, hitzak bilatu daitezke eta tituluaren hitz hori duten liburuak agertuko dira. Gainera, orrialdean erregistratuta dauden erabiltzaile kopurua eta igota dauden liburu kopurua agertzen dira “Estadistikak” atalean.



Nola ez, funtzionalitate hauek guztiak inplementatzeko hainbat teknologia desberdinez baliatzea eskatzen zen eta, behin orrialdearen oinarrizko funtzionamendua azalduta, orain banan-banan teknologiak non eta nola erabili ditugun argituko dugu.

IRAKASLEARENTZAKO ARGIBIDEAK

Gure webaren funtzionalitatea osotasunean ikusteko bi kontu sortu ditugu liburutegirako, bata erabiltzaile arrunta eta bestea administratzailea. Noski, kontu berriak ere erregistratu daitezke, baina hauek erabil ditzakezue ere. Beraz, hona hemen logeatzeko eposta eta pasahitzak.

- **Erabiltzaile arrunta.** Eposta: irakurlea@ehu.eus. Pasahitza: irakurlea.
- **Administratzailea.** Eposta: admin@ehu.eus. Pasahitza: admin000.

000webhost.com orrialdean erabilgarri jarri dugu web aplikazioa:
<https://sza3proiek.000webhostapp.com/index.php>

GitHub-eko repositorioaren linka:

https://github.com/bsarasuspe/SZA_3PraktikaOrok

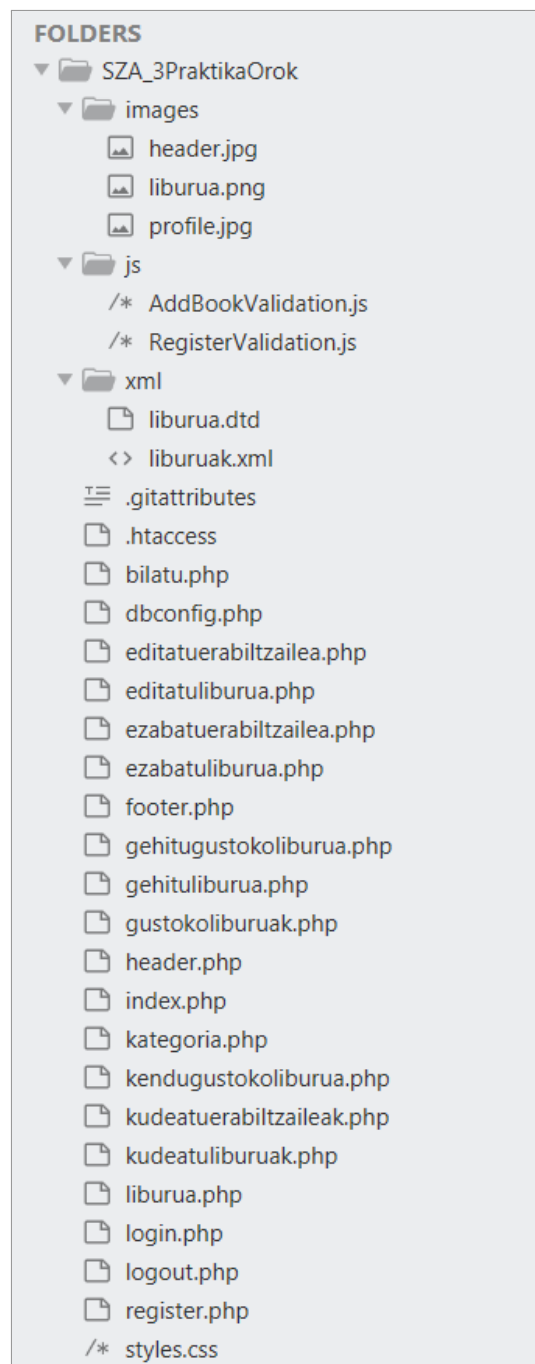
Bestalde, komenigarria da Google-ko kontua sartuta edukita izatea; izan ere, liburu batzuk Google Drive-ra igo ditugu eta partekatzeko linkaz baliatu gara online edukitzeko. Beraz, **Google-ko kontu bat izan behar da linkean sartu ahal izateko** (ez guztietan, adibidez, The Great Gatsby liburua zuzenean online aurkitu dugu link batean).

Azkenik, gure web-aplikazioaren fitxategiak honela daude antolatuta:

- **images karpeta:** bertan erabili ditugun irudiak gordetzen dira.
- **js karpeta:** erabili ditugun javascript fitxategiak kokatzen dira bertan.
 - **AddBookValidation.js:** liburuak gehitu formularioaren balidazioa.
 - **RegisterValidation.js:** erregistro formularioaren balidazioa.
- **xml karpeta:** hemen kokatzen da liburuak gordetzen ditugun xml fitxategia, baita ere bere dtd egitura.
 - liburuak.xml
 - liburua.dtd
- **index.php:** webgunearen hasierako orria.
- **header.php:** webgunearen header atala, bertan kokatzen da nabigazio barra (orri guztietan ez errepikatzeko, include eginez).

- **footer.php:** webgunearen footer atala (orri guztietan ez errepikatzeko, include eginez).
- **liburua.php:** liburu konkretu baten informazioa bistaratu.
- **kategoria.php:** kategoria konkretu batean dauden liburuak bistaratu.
- **bilatu.php:** bilatzailean egindako bilaketaren emaitza erakutsi.
- **register.php:** erabiltzaileen erregistro orria.
- **login.php:** erabiltzaileak login egiteko orria.
- **logout.php:** erabiltzaileak logout egiteko orria.
- **gehituliburua.php:** liburu berri bat gehitzeko orria.
- **kudeatuliburuak.php:** liburu zerrenda erakutsi, liburuak editatzeko edo ezabatzeko.
- **editatuliburua.php:** liburua editatzeko orria.
- **ezabatuliburua.php:** liburua ezabatzen du (SQL aginduarekin), liburuaren id-a POST moduan pasata.
- **gehitugustokoliburua.php:** liburua gustoko jartzen du (SQL aginduarekin), liburuaren id-a eta erabiltzailearen eposta POST moduan pasata.
- **kendugustokoliburua.php:** liburua gustoko kentzen du (SQL aginduarekin), liburuaren id-a eta erabiltzailearen eposta POST moduan pasata.
- **gustokoliburuak.php:** erabiltzailearen gustoko liburu zerrenda erakutsi.
- **kudeatuerabiltzaileak.php:** erabiltzaile zerrenda erakutsi, liburuak editatzeko edo ezabatzeko.
- **ezabatuerabiltzailea.php:** erabiltzailea ezabatzen du (SQL aginduarekin) erabiltzailearen eposta POST moduan pasata.
- **editatuerabiltzailea.php:** erabiltzailea editatzeko orria.
- **dbconfig.php:** datu-basearekin konexioa egiteko beharrezko datuak daude bertan.
- **styles.css:** web aplikazioari itxura ematen dion css fitxategia.

Laburbilduz, hau da gure web aplikazioaren fitxategi zuhaitza:



TEKNOLOGIA eta LENGOAIAK

Arestian aipatu bezala, klasean ikusitako teknologia inplementatu behar genituen eta banaka aztertuko ditugu non eta nola erabili ditugun. Zehazki, HTML, CSS, JavaScript, PHP, XML eta datu-baseez baliatu gara. Azken hau klasean ikusi ez dugun teknologia izanik, azalpen bat ere emango dugu.

1. HTML

HTML erabiliz garatu dugu gure orrialdearen hezurdura garatzeko. “index.php” orri nagusia izanik, hemen bila ditzakegu gure webaren atal nagusiak. Hasteko, ‘head’ atalean zehazten ditugu ondoren azalduko dugun CSS fitxategiarekiko konexioa, Google-tik hartu ditugun letra-motak edota erlaitzean agertzen den orriaren ikonoa.

```
<head>
  <title>Liburutegia</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="styles.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Lato:wght@700&family=Roboto&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Jost:wght@300;400;500;600;700;800;900&display=swap" rel="stylesheet">
  <link rel="icon" href="images/liburua.png" type="image/x-icon">
</head>
```

‘body’ atalean, berriz, lehenengo titulua agertzen den burukoa dago zeinak link bat duen hasierako orrira eramaten duena zauden lekuan zaudela. Bigarrenik, burukoaren azpiko menua definitu dugu eta, horretarako, lista bat sortu dugu zeinak logeatzeko eta erregistratzeko orrietara eramaten duten. Hirugarrenik, ezkerreko zutabea dago, zeinetan liburu bilatzailea, kategoriak eta estadistikak dauden. Liburu bilatzaileak bertan idatzitakoa aldagai bezala pasatzen du geroago azalduko dugun PHP martxan jartzeko eta, kategoria bakoitzean klikatzea, dagokion aldagaia pasatzen da ere. Hona hemen “Komedia” kategoriaren adibidea:

```
<ul>
  <li>
    <a href="index.php">...</a>
  </li>
  <li>
    <a href="login.php">...</a>
  </li>
  <li>
    <a href="register.php">...</a>
  </li>
</ul>
```



```
<div id="kategoria">
  <a href="kategoria.php?kategoria=Komedia">Komedia</a>
</div>
```

Gainera, bigarren zutabe bat ere badago, lehenengoa baino zabalagoa, zeinetan liburuak ageri diren eta, liburu bakoitzak id bat izanik, liburuan klik egitean id hori pasatzen da PHP-rentzako aldagai bezala liburuaren datuak lortzeko.

Hori da HTML-rekin egin dugun egitura nagusia. Esan bezala, beste orrialde batzuetarako estekak jarri ditugu, adibidez, logeatzeko eta erregistratzeko zeinetan formulario bat betetzen den. Zehazki, erregistratzeko formularioa hainbat etiketa eta text motako inputez osatzen da, horrela, izen-abizenak, e-posta, pasahitza eta pasahitza errepikatua jaso eta formularioaren bidez bidali ditzake. Halere, ia orri guztietan hasieran azaldutako HTML egitura nagusia mantentzen da.

2. CSS

HTML-rekin egindakoari itxura deigarria emateko erabili dugu CSS, zehazki, 'styles.css' izeneko beste fitxategi batean definitu dugu itxura. Burukoarekin hasita, atzeko irudia eta letren itxura definitu ditugu HTML-an ezarritako identifikatzaileekin.

```
#header{
  width: 100%;
  height: 180px;
  background-image: url('images/header.jpg');
}

#headertext{
  font-size: 40px;
  font-weight: bold;
  color: white;
  text-align: center;
  padding-top: 75px;
  text-shadow: 2px 2px 4px #000000;
}
```

Zerrendetarako, ordea, etiketez baliatu gara eta ez identifikatzaileen ordeaz, alegia, etiketa dutenei CSS bera aplikatuko zaie eta etiketadunei ere bai. Aipatzekoa da ere hover-en erabilera, zeinak listako elementuen gainetik sagua pasatzean kolorez aldarazten dituen.

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  overflow: hidden;  
}  
  
li {  
  float: left;  
}
```

```
li a:hover {  
  background-color: #442727;  
}
```

Aipatzekoa da ere, CSS aplikatzeko bi modu horiez gain, class atributuaz ere baliatu garela. Hain zuzen ere, webean etiketa batzuen ondoan jarri ditugun irudi txikiei (ikonoiei) class = 'svg-icon' ezarri diegu, horrela, 'svg-icon' erabiliz CSS fitxategian hauek guztiak modifikatu ditugu.

```
.svg-icon {  
  width: 1em;  
  height: 1em;  
}
```

Modifikatzeko hiru modu hauekin eta CSS-k eskaintzen dituen etiketekin eraldatu dugu gainontzeko guztia: liburuazalen itxura, formularioen tamaina eta koloreak, orriaren atzeko koloreak... Guztiari itxura erakargarria eta homogeneoa emateko.

3. JavaScript

JavaScript formularioak balidatzeko erabili dugu, bai erregistro formularioan, baita liburuak gehitzeko eta editatzeko formularioetan ere. Horrela, formulario bakoitzean errore kontrol bat eginez, sarrerei baldintza batzuk betetzea exigitzen dizkiegu.

Hasteko, liburuak gehitzean egiten dugun JavaScript errore kontrola aztertuko dugu. Kasu honetan, errore kontrola erraza da, izan ere, eremu batuzetan bete beharreko baldintza eremua hutsik ez uztea da. Hona hemen bete daitezkeen eremuak eta bete ezean agertzen diren errore mezuak:

Derrigorrez bete behar diren eremuak titulua, egilea, sinopsia, irudiaren url-a eta deskarga url-a dira. Hizkuntza eta kategoria atalek defektuzko balioa dute, beraz, ez dira errore kontrolean sartzen, baina balio egokiak ematea komeni da. Bestalde, argitaratze-data ez da beharrezkoa. Hona hemen baldintza horiek betetzeko kodea (ez guzita, titulua eta egilearen adibideak):

```
function AddBookValidation(){
    var titulua = document.forms["form"]["titulua"].value;
    var egilea = document.forms["form"]["egilea"].value;
    var sinopsia = document.forms["form"]["sinopsia"].value;
    var irudia = document.forms["form"]["irudia"].value;
    var deskarga = document.forms["form"]["deskarga"].value;

    if (titulua.length == 0){
        document.forms["form"]["titulua"].style.border='2px solid red';
        document.getElementById("titulua-error").innerHTML = "Titulua falta da.";
    }else{
        document.forms["form"]["titulua"].style.border='2px solid #c6c0bc';
        document.getElementById("titulua-error").innerHTML = "";
    }

    if (egilea.length == 0){
        document.forms["form"]["egilea"].style.border='2px solid red';
        document.getElementById("egilea-error").innerHTML = "Egilea falta da.";
    }else{
        document.forms["form"]["egilea"].style.border='2px solid #c6c0bc';
        document.getElementById("egilea-error").innerHTML = "";
    }
}
```

Azken finean, hau da bezeroan egiten dugun balidazioa JavaScriptekin, hain zuzen ere, “AddBookValidation()” .js fitxategi batean definituta dago eta

funtzioari dei egiten diogu formularioari submit egitean (onsubmit=return“AddBookValidation()”).

Bigarrenik, erregistratzean erabilitako errore kontrola aztertuko dugu. Orain, derrigorrez bete beharreko eremuak guztiak dira, baina hutsa ez uzteaz gain beste baldintza batzuk bete behar dira. Batetik, pasahitzak gutxienez 8 karaktereko luzera izan behar du eta pasahitza errepikatuaren berdina izan behar da.

```
if (pasahitza.length == 0){
    document.forms["form"]["pasahitza"].style.border='2px solid red';
    document.forms["form"]["pasahitza2"].style.border='2px solid red';
    document.getElementById("pasahitza-error").innerHTML = "Pasahitzak ezin du hutsa izan.";
}else{
    document.forms["form"]["pasahitza"].style.border='2px solid #c6c0bc';
    document.forms["form"]["pasahitza2"].style.border='2px solid #c6c0bc';
    document.getElementById("pasahitza-error").innerHTML = "";
    if(pasahitza.length<9){
        document.forms["form"]["pasahitza"].style.border='2px solid red';
        document.forms["form"]["pasahitza2"].style.border='2px solid red';
        document.getElementById("pasahitza-error").innerHTML = "Pasahitzak gutxienez 8 karaktere izan behar ditu.";
    }else{
        document.forms["form"]["pasahitza"].style.border='2px solid #c6c0bc';
        document.forms["form"]["pasahitza2"].style.border='2px solid #c6c0bc';
        document.getElementById("pasahitza-error").innerHTML = "";
        if(pasahitza2 != pasahitza){
            document.forms["form"]["pasahitza"].style.border='2px solid red';
            document.forms["form"]["pasahitza2"].style.border='2px solid red';
            document.getElementById("pasahitza-error").innerHTML = "Pasahitzak ez dira berdinak.";
        }else{
            document.forms["form"]["pasahitza"].style.border='2px solid #c6c0bc';
            document.forms["form"]["pasahitza2"].style.border='2px solid #c6c0bc';
            document.getElementById("pasahitza-error").innerHTML = "";
        }
    }
}
```

Bestetik, *pattern* edo eredu bat jarraitu dugu eposta egiaztatzeko, alegia, karaktere kate bat, '@' ikurra, beste karaktere kate bat, puntua eta puntu ostean bi edo hiru letra izan behar dira. Hona hemen pattern-a, eremuak eta eremuak bete ezean agertzen diren erroreak:

```
var pattern = /^w+@[a-zA-Z_]+?\.[a-zA-Z]{2,3}$/;
```

Bestalde, zerbitzarian ere egiten ditugu errore kontrolak, baina PHP bidez egiten dira eta datorren atalean azalduko ditugu.

4. PHP

PHP-ez baliatu gara zerbitzari aldeko kodea programatzeko, hain zuzen ere, sesioak eta bere aldagaiak kudeatzeko, XML fitxategitik (liburuak.xml) eta datu-basetik (erabiltzaileak) datuak atzitzeko, gehitzeko eta aldatzeko, eta azkenik, erregistro garaian eta liburuak gehitzean zerbitzari aldean datuen balidazioa egiteko.

- Sesioak:

Sarreran aipatu dugun bezala, gure web aplikazioan erabiltzaileek logeatzeko aukera dute, eta hau guztia kudeatu ahal izateko PHP sesioak erabili ditugu.

Erabiltzailea logeatzen denean, `$_SESSION["kautotua"] = "BAI"` gordetzen da eta `$_SESSION` aldagaietan erabiltzailearen eposta, izena eta mota gordetzen dira.

```

if (($stabladatauak = $sema->fetch_row()) != null) {
    if ($datuak["eposta"] == $stabladatauak[0] && $datuak["pasahitza"]==$stabladatauak[2]) {
        $_SESSION["kautotua"] = "BAI";
        $_SESSION["eposta"] = $stabladatauak[0];
        $_SESSION["izena"] = $stabladatauak[1];
        $_SESSION["mota"] = $stabladatauak[3];
        echo "<script> window.location.href = 'index.php';</script>";
    }
}

```

login.php

Era horretan, erabiltzailea kautotuta dagoela adierazten da eta erabiltzaile bakoitzari dagokion menua erakutsi ahal diogu motaren arabera. Administratzailea bada, liburua gehitu, liburuak kudeatu eta erabiltzaileak kudeatu ahal izango du. Jarraian ikus daiteken bezala:

```

if(isset($_SESSION['kautotua']) && ($_SESSION['kautotua'] == "BAI")){
    if($_SESSION['mota'] == 1){
        echo '<li><a href="gehituliburuak.php"><svg xmlns="http://www.w3.org/2000/svg" class="icon icon-tabler icon-tabler-book" width="15" height="15"><path stroke="none" d="M0 0h24v24H0z" fill="none"></path><path d="M3 19a9 9 0 0 1 9 0a9 9 0 0 1 9 0"></path><path d="M3 6a9 9 0 0 1 9 0a9 9 0 0 1 9 0"></path><line x1="3" y1="6" x2="3" y2="19"></line><line x1="12" y1="6" x2="12" y2="19"></line><line x1="21" y1="6" x2="21" y2="19"></line></path></svg> Liburuak gehitu</a></li>';
    }
    echo '<li><a href="kudeatuliburuak.php"><svg xmlns="http://www.w3.org/2000/svg" class="icon icon-tabler icon-tabler-book" width="15" height="15"><path stroke="none" d="M0 0h24v24H0z" fill="none"></path><path d="M3 19a9 9 0 0 1 9 0a9 9 0 0 1 9 0"></path><path d="M3 6a9 9 0 0 1 9 0a9 9 0 0 1 9 0"></path><line x1="3" y1="6" x2="3" y2="19"></line><line x1="12" y1="6" x2="12" y2="19"></line><line x1="21" y1="6" x2="21" y2="19"></line></path></svg> Liburuak kudeatu</a></li>';
}

```

header.php

Baina menuan estekak ez agertzearekin ez da nahikoa, izan ere, zuzenean url-a jartzen bada, liburuak eta erabiltzaileak kudeatu litezke nahiz eta administratzailea ez izan. Hori dela eta, segurtasun minimo bat gehitu beharra izan diogu. Administratzailea ez den norbait (erabiltzaile arrunta edo ez kautotua) administratzaileen funtzioetako bat erabiltzen saiatzen bada, hasierako orrira joaten da zuzenean.

```

<?php
if(isset($_SESSION["kautotua"])){
    if (($_SESSION["kautotua"] != "BAI") || ($_SESSION["mota"] != 1)) {
        echo "<script> window.location.href = 'index.php';</script>";
        exit();
    }
}
else{
    echo "<script> window.location.href = 'index.php';</script>";
}

```

Liburuen deskarga url-a ikusi eta liburuak gustoko jartzeko aukera erabiltzaile logeatuek soilik dute, beraz, aurrekoaren antzera, `$_SESSION["kautotua"] == "BAI"` baldin bada soilik erakutsiko da.

```
if(isset($_SESSION['kautotua']) && ($_SESSION['kautotua'] == "BAI")){
    echo "<a href='\$liburua->deskarga' target='_blank'><button action='\$liburua->deskarga'>Deskargatu liburua</button></a>";
}else{
    echo "<div id='errorbox'>Liburua deskargatzeko erabiltzaile erregistratua izan behar zara! <a href='register.php'>Erregi
}
```

liburua.php

- **Liburuak atzitu/gehitu/editatu:**

Liburuak gehitu, editatu eta ezabatu PHP-ri esker egiten dugu. Azken finean, dokumentu honetako PHP atalean egiten den guztia XML fitxategia osatzeko da, beraz, XML atalean azalduko dugu sakonago.

- **Erabiltzaileak atzitu/gehitu/editatu:**

Erabiltzaileen kudeaketa ere PHP-ri esker egiten da. Lehenik eta behin, erregistroan erabiltzaileak gehitzen dira, beti ere bezero aldean JavaScript erabilia eta zerbitzari aldean php-rekin datuen balidazioa bete bada. Datu-basean MySQL agindu bat erabilia gortzen dira erabiltzaileak, baina hau Datu Basearen atalean sakonago azalduko dugu.

Ondoren, erabiltzaileak kudeatu atalean datu-basea atzitzen da, erregistratutako erabiltzaile zerrenda lortzeko (berriz ere PHP eta MySQL erabilia). Erabiltzaileak editatu eta ezabatzerako garaian ere PHP eta MySQL erabiltzen dugu.

- **Formularioetako datuen balidazioa:**

Erregistroan eta liburuak gehitzean, nahiz eta JavaScript bidez balidazioa egin, zerbitzari aldean ere datu horien balidazioa egitea beharrezkoa da. Izan ere, JavaScript kentzea oso erraza da nabigatzailean eta era horretan datu desegokiak sartzeko aukera dago datu-basean eta xml fitxategian.

Adibidez, register.php-an honela egin dugu datuen balidazioa PHP-rekin:

```
if (isset($_POST['eposta'])) {
    if(strlen($datuak["eposta"])>0 && strlen($datuak["izena"])>0 && strlen($datuak["pasahitza"])>0 && strlen($datuak["pasahitza2"])>0){
        if(preg_match("/^\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,3}$/", $datuak["eposta"])){
            if(strlen($datuak["pasahitza"]) > 8){
                if($datuak["pasahitza"] == $datuak["pasahitza2"]){
```

5. DTD

DTD egitura ere definitu dugu liburuak definitzeko, XML fitxategi batean gordetzen baitira liburuak. Hona hemen DTD fitxategian egiturak nola definitu ditugun.

```
<!ELEMENT liburuak (liburua)*>
<!-- liburuak azkenid IDREF #REQUIRED -->
<!-- liburuak (titulua, egilea, hizkuntza, kategoria, data, sinopsia, irudia, deskarga)>
<!-- liburuak id ID #REQUIRED -->
<!-- liburuak (titulua) (#PCDATA) -->
<!-- liburuak (egilea) (#PCDATA) -->
<!-- liburuak (hizkuntza) (#PCDATA) -->
<!-- liburuak (kategoria) (#PCDATA) -->
<!-- liburuak (data) (#PCDATA) -->
<!-- liburuak (sinopsia) (#PCDATA) -->
<!-- liburuak (irudia) (#PCDATA) -->
<!-- liburuak (deskarga) (#PCDATA) -->
```

Esan bezala liburuak jasotzen dituen egitura bat dugu zeinak ‘liburuak’ izena duen eta ‘liburua’ bezala definitu dugun egitura asko jaso ditzaken. Zerrenda honek atributu bakarra izango du: jaso den azken liburuaren ID-a (‘azkenid’), beraz, liburu bakoitzak ere bere ‘id’ propioa izan behar du derrigorrez. Horrez gain, hainbat string edukiko ditu: titulua, egilea, hizkuntza, kategoria, data (string moduan ere), sinopsia, azaleko irudiaren linka eta deskarga linka. Egitura hau baliatuz gordetzen dira XML fitxategian.

6. XML

Azaldu berri dugun DTD egiturak (liburuak), esan bezala, XML fitxategi batean gordetzen dira, zehazki, ‘liburuak.xml’ fitxategian. Hemen ikus dezakegu lortzen den egitura adibide baten bitartez:


```

<?xml version="1.0"?>
<!DOCTYPE liburuak SYSTEM "liburuak.dtd">
<liburuak azkenid="21">
  <liburua id="1">
    <titulua>Harry Potter y la piedra filosofal</titulua>
    <egilea>J.K. Rowling</egilea>
    <hizkuntza>Gaztelera</hizkuntza>
    <kategoria>Fantasia</kategoria>
    <data>1997-06-26</data>
    <sinopsia>Harry Potter se ha quedado hu&#xE9;rfano y vive en casa de sus abominables t&#xED;os y el insuportable primo Dudley. Harry se siente muy
    <irudia>https://i.kinja-img.com/gawker-media/image/upload/s--AUsfSKba--/c_scale,f_auto,fl_progressive,q_80,w_800/18eqlyw2pur67jpg.jpg</irudia>
    <deskarga>https://www.amazon.es/Viajo-sola-SELLER-Samuel-Bjork/dp/8490629161</deskarga>
  </liburua>

```

Horrelako egitura bat lortzeko 'gehituliburua.php' fitxategitik egitura bat osatzen dugu eremuetan lortutako parametroekin. Zehazki, .xml fitxategia erreferentziatu eta azaldutako DTD egitura jarraitu nahi dugunez, *addAttribute()* eta *addChild()* metodoen bidez parametroak egituratzen ditugu. Adibidez, irudian ikus daiteke, '\$xmlliburua' aldagaia lortzean, addAttribute('id', \$azkenid) egiten dugula id-a ezartzeko.

```

if (isset($_POST['titulua'])) {
    if($_POST['titulua'])
        $fitxategia = "xml/liburuak.xml";
    $xml = simplexml_load_file($fitxategia);
    $azkenid = $xml["azkenid"];
    $azkenid = $azkenid + 1;
    $xml["azkenid"] = $azkenid;
    $xmlliburua = $xml->addChild('liburua');
    $xmlliburua->addAttribute('id', $azkenid);
    $xmlliburua->addChild('titulua', $_POST['titulua']);
    $xmlliburua->addChild('egilea', $_POST['egilea']);
    $xmlliburua->addChild('hizkuntza', $_POST['hizkuntza']);
    $xmlliburua->addChild('kategoria', $_POST['kategoria']);
    $new_date = date('Y-m-d', strtotime($_POST['date']));
    $xmlliburua->addChild('data', $new_date);
    $xmlliburua->addChild('sinopsia', $_POST['sinopsia']);
    $xmlliburua->addChild('irudia', $_POST['irudia']);
    $xmlliburua->addChild('deskarga', $_POST['deskarga']);
    $xml->asXML('xml/liburuak.xml');
    echo "<script> window.location.href = 'gehituliburua.php?gehitu=true';</script>";
}

```

7. Datubasea

Lehenago aipatu dugun bezala, gure web aplikazioan erabiltzaile bat erregistratzeko aukera dago. Erabiltzaile hauen datuak (eposta, izena, pasahitza eta mota) datu basean gordetzen dira, baita ere erabiltzaile bakoitzak dituen gustoko liburuak.

Gure kasuan MySQL erabili dugu, datu-baseak administratzeko open source sistema bat da, Oraclek garatutakoa. Datu-base batean datuak biltegitatu nahi direnean, hainbat ikuspegi erabil ditzakezu. MySQL-ek datu-base erlazional izeneko ikuspegi bat aukeratzen du. Datu-base erlazional batekin, datuak hainbat biltegitatze-eremutan zatitzen dira (tauletan).

Adibidez, gure kasuan bi taula ditugu:

- **erabiltzaileak**: bertan lau atributu desberdin ditugu eta erabiltzaile bakoitzaren informazioa gordetzen da (eposta, izena, pasahitza eta mota).

- **Taularen egitura:**

- eposta VARCHAR(254) PRIMARY KEY
- izena TEXT
- pasahitza TEXT
- mota INT(11)

Datu-baseetan taula guztiek dute gako bat. Gakoa atributu batek edo bat baino gehiagok osatu dezake eta ezin du inoiz errepikatuta egon taula batean. Taula honetan eposta da gakoa, ezin baita erabiltzaile bat baino gehiago egon eposta berdinarekin.

Aipatu beharra dago gakoak ongi definitzea oso garrantzitsua dela, ondoren SELECT, INSERT, UPDATE eta DELETE modu egokian egin ahal izateko.

eposta	izena	pasahitza	mota
admin@ehu.eus	Admin Admin	admin000	1
bsarasua@ehu.eus	Beñat Sarasua	bsarasua1	0
inakiurdan@gmail.com	Iñaki Urdangarin	inakiurdan	1
irakurlea@ehu.eus	Irakurlea Irakurlea	irakurlea	0
manex@ehu.eus	Manex Atxa Landa	manexatxa	1

Taulan gordetzen diren datuak

#	Nombre	Tipo
1	eposta 	varchar(254)
2	izena	text
3	pasahitza	text
4	mota	int(11)

Taularen egitura



- **gustoko_liburuak:** taula honetan aldiz, bi atributu ditugu eta erabiltzaile bakoitzak zein liburu gustoko dituen gordetzen da (erabiltzaile bakoitzak gustoko duen liburu bakoitzarentzat tupla bat, erabiltzailearen eposta eta liburuaren id-arekin)

- **Taularen egitura:**

- user_eposta VARCHAR(254) PRIMARY KEY
- liburu_id INT(11) PRIMARY KEY

Kasu honetan aldiz, gakoa bi atributuk osatzen dute (user_eposta eta liburu_id), izan ere, erabiltzaile batek liburu bat baino gehiago izan dezake gustoko eta aldi berean liburu bat erabiltzaile batek baino gehiago izan dezake gustoko.

user_eposta	liburu_id
admin@ehu.eus	1
admin@ehu.eus	2
admin@ehu.eus	5
admin@ehu.eus	13
admin@ehu.eus	21
bsarasua@ehu.eus	2
elorrira14@gmail.com	13
elorrira14@gmail.com	16

#	Nombre	Tipo
1	user_eposta 	varchar(254)
2	liburu_id 	int(11)

Taularen egitura

Taulan gordetzen diren datuak

Datu-basean datuak (tuplak) gehitzeko eta atzitzeko SQL aginduak erabiltzen dira: INSERT, UPDATE eta DELETE. Gure web aplikazioan erabili ditugun aginduak azalduko ditugu labur-labur.

- **Erregistroa:**

Erregistro formularioaren bidez eskuratzen ditugun datuak (JavaScript bidez eta baita zerbitzariaren aldean balidatuak) datu basean sartu behar dira. Horretarako, honako SQL agindu hau erabili dugu:

```
INSERT INTO erabiltzaileak(eposta, izena, pasahitza, mota)
VALUES ('$_POST[eposta]', '$_POST[izena]',
'$_POST[pasahitza]', '0')
```

Aipatu beharra dago mota atributuan 0 balioa sartzen da, erregistratzen den erabiltzailea erabiltzaile arrunta delako defektuz. Administratzaileek mota 1 dute, baina hori erregistratu ondoren aldatu behar da.

- **Logina:**

Formularioaren bidez login egin ondoren, datu-basea atzitu behar da erabiltzaile eposta eta bere pasahitza egokiak direla konprobatzeko eta egokiak badira login egiteko. Erabili dugun agindua honakoa izan da:

```
SELECT eposta, izena, pasahitza, mota FROM erabiltzaileak
WHERE eposta = '".$_POST["eposta"]."'
```

Honekin erabiltzailea eskuratzen dugu eta ondoren konprobatu beharko da pasahitza egokia dela. Hori horrela bada, sesioa hasi eta erabiltzailearen datuak (eposta, izena eta mota) \$_SESSION aldagaietan gordeko dira.

- **Gustoko liburuak:**

Liburuen atalean liburu bat gustoko jartzeko edo kentzeko aukera dago. Erabiltzaileak liburua gustoko nahi badu jarri, “gustoko jarri” botoia agertzen da. Erabiltzaileak liburua gustoko badu dagoeneko aldiz, “gustoko kendu” botoia agertzen da. Hau egiteko, lehenik datu-basetik atzitu behar da dagoeneko gustoko duen ala ez jakiteko. Horretarako, honako SQL agindua erabili dugu:

```
SELECT * FROM gustoko_liburuak WHERE user_eposta =
'$_SESSION[eposta]' AND liburu_id = '$_GET[id]'
```

SQL agindu horren emaitzaren arabera botoi bat edo bestea agertuko da, beti ere erabiltzailea logeatuta badago.

Gustoko liburua gehitu nahi badugu, “Gustoko jarri” sakatzean honako SQL agindua exekutatzen da:

```
INSERT INTO gustoko_liburuak VALUES('$_POST[eposta]',  
'$_POST[id]')
```

“Gustoko kendu” botoia sakatzen badugu aldiz, honakoa:

```
DELETE FROM gustoko_liburuak WHERE user_eposta =  
'$_POST[eposta]' AND liburu_id = '$_POST[id]'
```

Azkenik, “Gustoko liburua” atalean erabiltzaileak gustoko dituen liburuen zerrenda erakusteko datu-basetik atzitzen da liburuen id-ak lortzeko eta ondoren XML fitxategitik liburu horien informazioa esakuratzen da. Hau da SQL agindua:

```
SELECT liburu_id FROM gustoko_liburuak WHERE user_eposta =  
'$_SESSION[eposta]'
```

- **Kudeatu erabiltzaileak:**

Kudeatu erabiltzaileak orrian, erabiltzaileen zerrenda bat erakusten da. Horretarako, datu-basetik eskuratu behar ditugu erabiltzaileak honako SQL aginduarekin:

```
SELECT eposta, izena FROM erabiltzaileak
```

Erabiltzaile bat editatutakoan, datu-basea eguneratzeko honako SQL agindu hau erabiltzen dugu:

```
SELECT eposta, izena, pasahitza, mota FROM erabiltzaileak  
WHERE eposta='$_GET[eposta]'
```

Azkenik, erabiltzailea ezabatzeko honako SQL agindua exekutatzen da:

```
DELETE FROM erabiltzaileak WHERE eposta =  
'".$_POST["eposta"].''
```

ONDORIOAK

Web orrialdea garatu ahala eta amaitu ostean hainbat ondoriotara iritsi gara. Hasteko, HTML bezalako teknologia sinple samar batekin jada egitura bat sortu dezakegula dakusagu, baina geroz eta teknologia eta lengoaia gehiagoz baliatuz lortzen den garapen maila asko hazten dela eta weba erakargarriago ez ezik, eraginkorrago bihurtzen dela ere ondorioztatu dugu.

Bestetik, teknologia desberdin hauek funtzio desberdinak burutzeko balio dutela ikasi dugu, baina baita funtzio berdin bat burutzeko teknologia desberdinak erabili daitezkeela ere. Esaterako, informazioa zerbitzarian gordetzeko XML erabili daiteke, baina datu-base baten bidez ere lortu daiteke. Gure kasuan, XML erabili dugu liburuak gordetzeko eta datu-basea erabiltzaileak gordetzeko.

Azkenik, ikusi dugu teknologia bat besteekin osatzea progresiboki eginda hobeto lortzen dela guztiak kolpetik aplikatzean baino, alegia, esaterako, HTML-rekin hasi, CSS aplikatu, JavaScriptekin formularioak osatu... bezalako ordena bat eramanda hobeto barneratzeaz gain, errazago da web orria osatzea zuzenean hainbat teknologiarekin batera lanean hastean baino.