

BASIC IDEA:


The project goal was to build a simple website which interconnect knowledge of HTML, CSS, JavaScript, and PHP. The basic idea was to build an online store to buy and sell products. We further developed this idea and we wanted to add some features as online academy where you can register on different courses and take lessons online. Then we implemented a section where users that are already registered in the database could post, comment and edit content of their profile. It is a dynamic website, which besides a nice and interesting style it has a solid database and it has a lot of functionalities waiting to be explored.

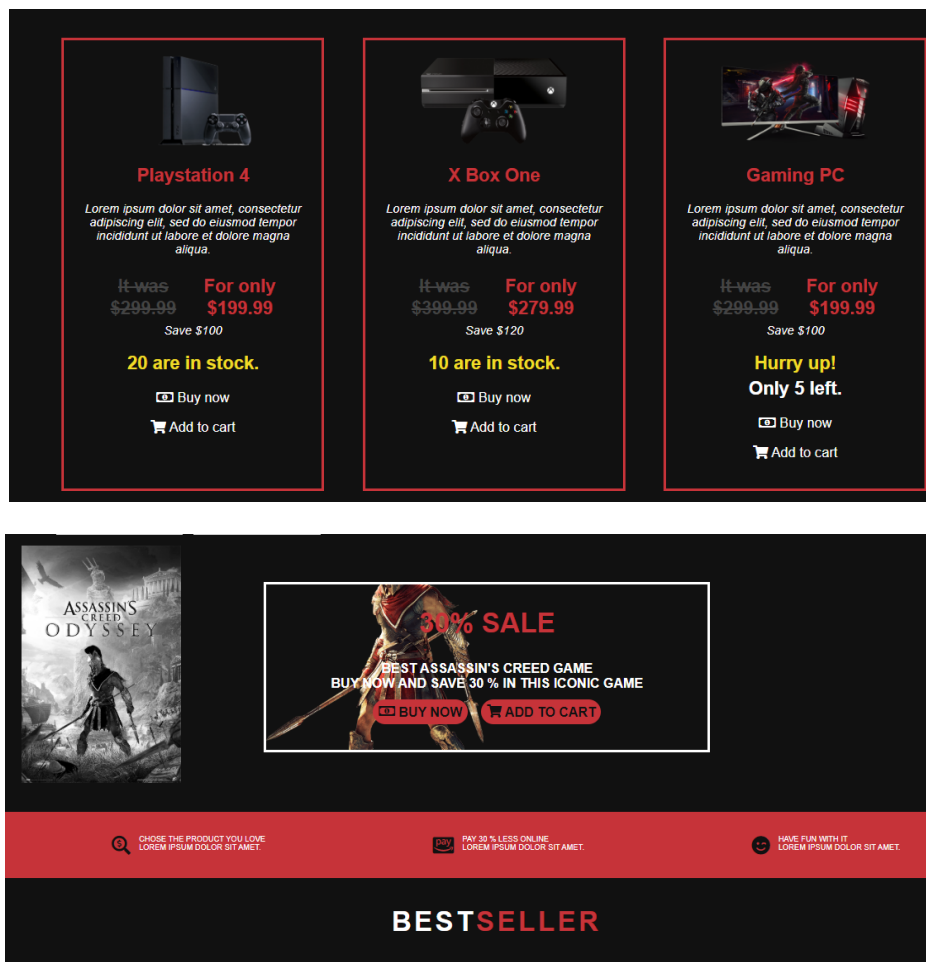
STRUCTURE OF THE PROJECT:

The project contains a large number of files and pages. Even though some of them might not be visible, they are working under the hood to keep the website “server” always working. In this section you can find main pages and files which handle the most important features of our website. You can find the database files attached with the submissions of this project.

- Login/Register – connected to database they allow us to register new user and to login. They are two other files that deal with data validation and updating the database such as login.inc.php and signup.inc.php.
- User Interface/Admin Interface – we created the project such that there would be two level of users. One will be the administrator of the network whose responsibility is to manage users data, add/remove products on the store, add new available courses and responds to incoming messages coming from visitors of our website.
- Index, About, Shop, Academy, Blog – are some of the main pages that will be explained in details in the subsequent sections. Index is just a static page styled with CSS and introduce the visitors with what they would expect to find in our site. The other pages are more dynamic, and they might affect the database.

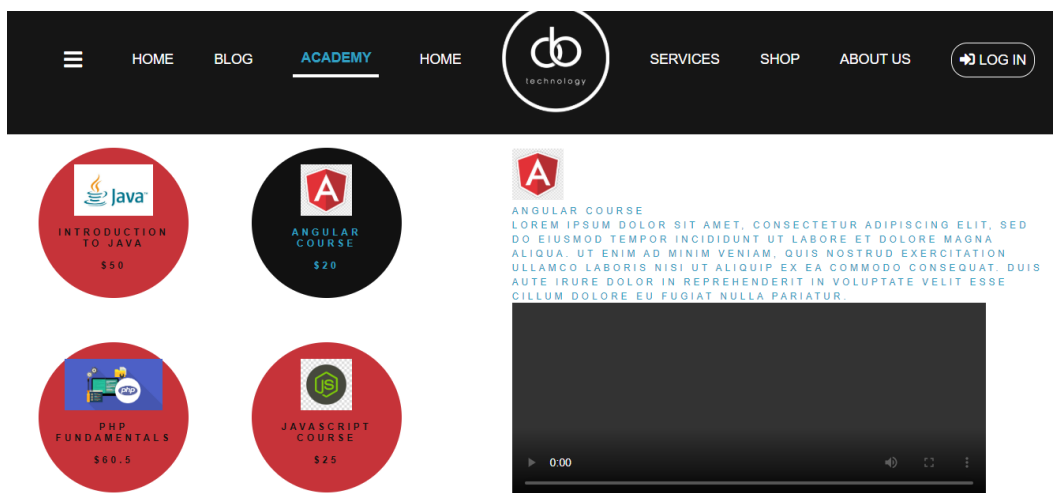
STRUCTURE OF MAIN PAGES

-  **Shop Page:** This is the page that deals with an online store where the user may buy some products and enjoy the offers of our website. Besides the style and effects there is something related with database. When user click “Buy Now” button, he/she should first provide some information such as username, email and card number which are validated in another file. If all data are valid, then the user completes the payment successfully. After the user purchases the product, the quantity of that product is decreased by 1. The stock quantity is saved in the database, and the invoice is printed in a file. The administrator of the network could find a detailed description of all the invoices in his profile whenever he logs in.



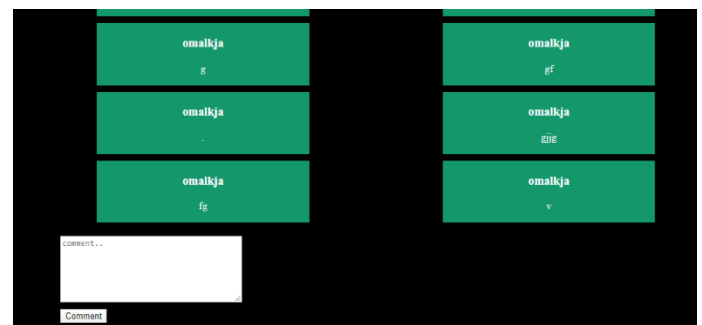
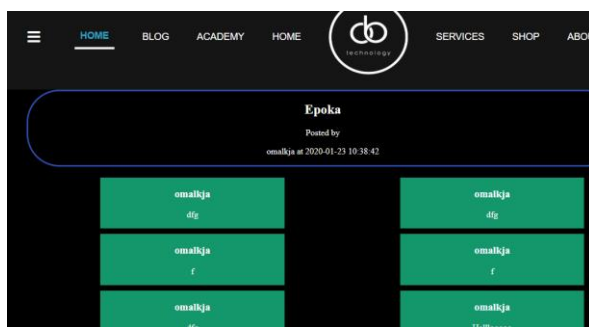
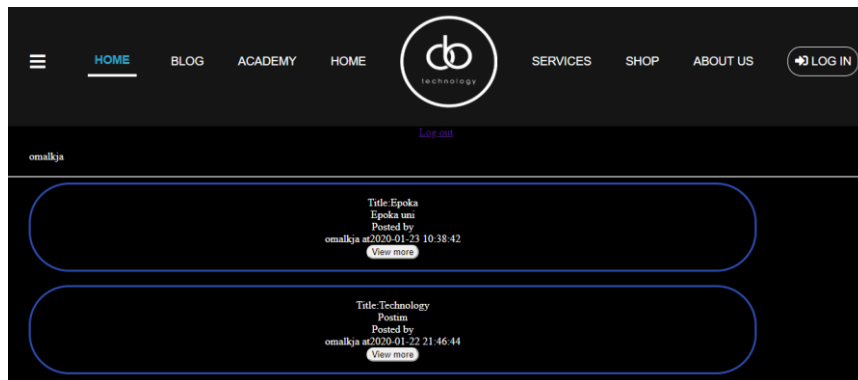
The PHP script that echo the containers where product are displayed, it as an if statement that if the product quantity is less than 5 it should echo “Hurry up” as you can see from the above pictures.

🚦 **Academy Page:** This page in this section is only to introduce to the visitors the opportunity to take courses online. Courses are all stored in the database and are displayed in this page. To register in one of them you should first need to log in in your profile and there you can find a section where you can add new courses or remove the current. The user may have a look at all the courses which are on sale. The courses information found in the database are displayed in the page. If the user clicks one of them, the details for only that course are displayed on the right.



On the right, you can see the information for only one course at time. This is an interesting feature achieved with a static container that is always there even when you scroll down the website. It contains the course name, course description, and a short introductory video.

🚦 **Blog Page:** At this page users are acquainted with the latest topics regarding to technology. The user can see all the posts which are shared from other users briefly. The user can view each post and the comments which are made for a particular post, by clicking **View more**. In the new page that displays the post and the comments for that post, the user of the page (who is not already logged in) has to previously login, in order to add a comment to that post.



You can post new contents if you are logged in and, in your profile will appear a section where you can add a new post which will be inserted into database and later will be shown into the blog page, where other users can comment and share their opinions about this topic. The posts are stored in database at the exact date and time using **TIMESTAMP** and the username of the person that posted that information.

User Interface Page: This is the user profile where he can manage his account information including the password, username and even the profile picture which he can change or delete. He may create a new post which will be shared with other users in blog page, he may view the courses he has purchased and may buy a new one. he may edit his account by changing his personal information or changing his profile picture.

WHAT DO YOU HAVE IN MIND?

Title:

Share your thoughts...

POST

Edit profile and add a new post.

EDIT ACCOUNT

Old Pass:

New Pass:

omalkja

New User:

Save changes

Save changes

localhost/Project/Project/userInterface.php

BUY A NEW COURSE

Course title	Price	Image	
Introduction to Java	50		Buy now
Angular Course	20		Buy now
Php fundamentals	60.5		Buy now

The section where you can register in a new course.

The section where you can check the purchased courses and their information

ject/Project/userInterface.php

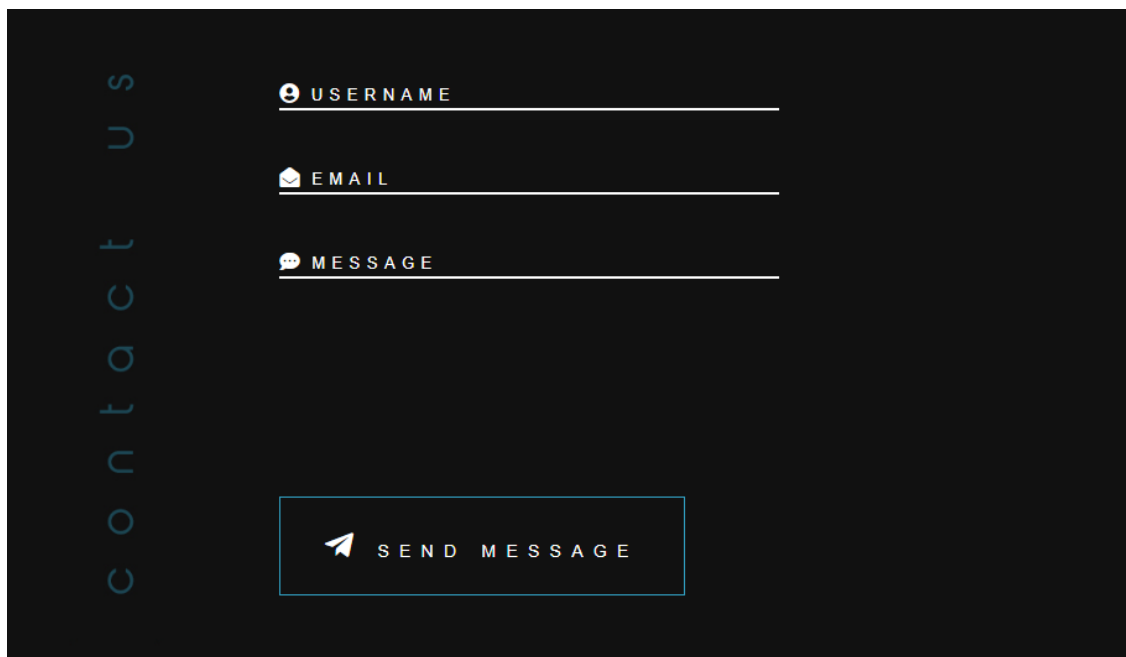
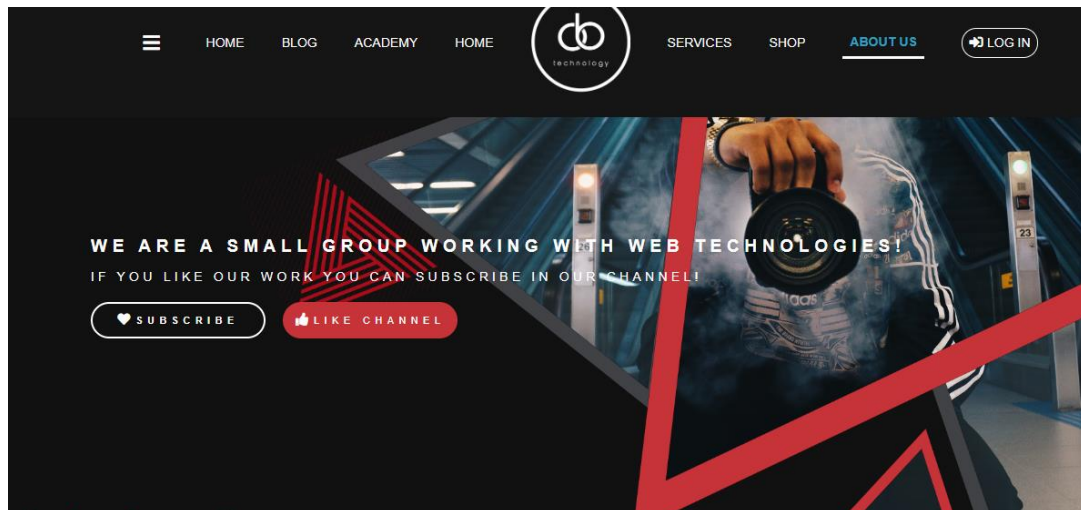
MY COURSES

INTRODUCTION TO JAVA

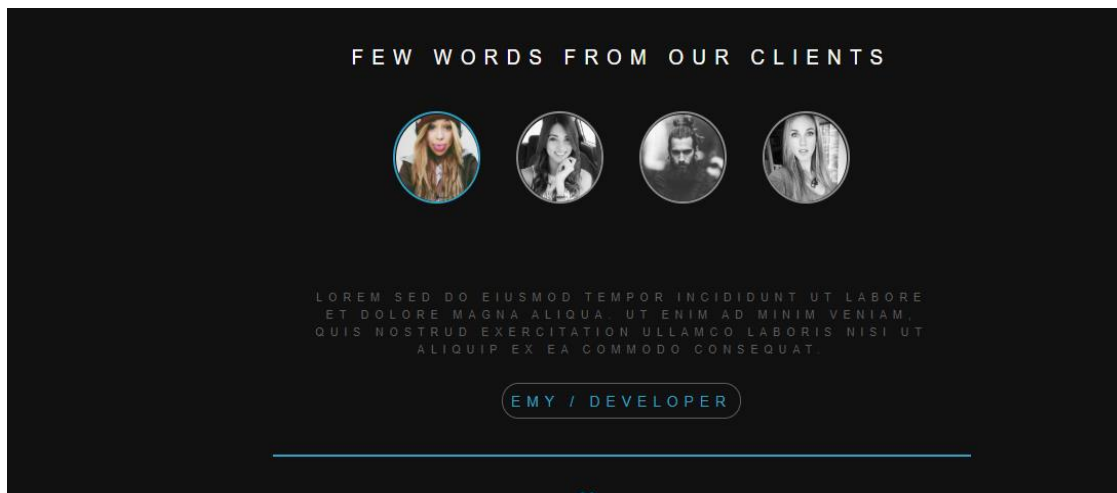
ELITE TECHNOLOGY

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullam laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

About Us page: An introduction of our tech company is made. Someone who visits our page may send a message to us, and the message will be saved in the database. The admin of the page will view the messages and reply to each later. Also, some opinions are shared from our satisfied “customers”.



The following screenshot is a section in this page where visitors can see and review what some ‘clients’ had witnessed about us. By using JavaScript (jQuery Library) you can change the content the description as you can see in the image below.

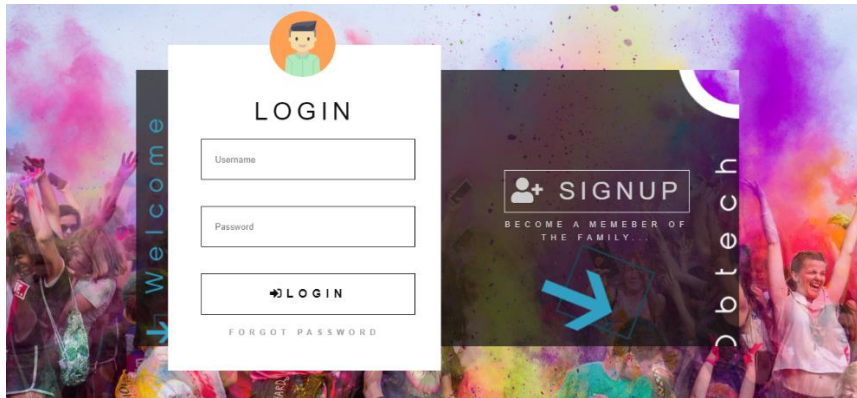


The content of the container changes any time you click an image.

Login/Register Page – in collaboration with some hidden files, they make possible the registration and logging of users.

Before you register you have to accept our term and conditions, privacy issues and other stuffs. When you register you have to complete the fields like username, email, password. You need to confirm the password and if it is not correct we have make a validation either with JavaScript that will alert a message saying your password does not match or with a php script which will not be executed in case there is an error in completing the fields.

A registration form titled "REGISTER". It contains two input fields: "Email" and "mm/dd/yyyy". Below these fields is a checkbox labeled "TERMS & CONDITIONS". At the bottom, there are two buttons: "BACK" and "SIGN UP".

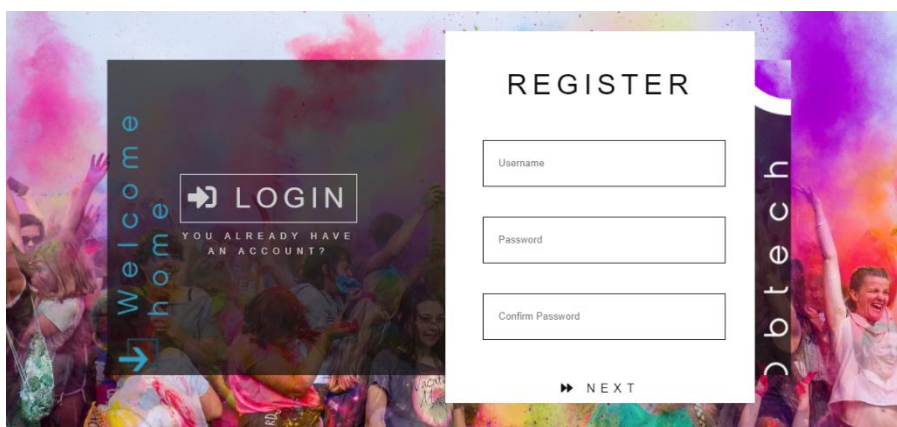


When you press login, you are redirected to login.inc.php which makes a quick validation to find if the credentials provided satisfy and user profile or not. If no user is found you are then redirected to login again. If the user exists than its profile is displayed on the screen. If any of the fields is empty than an alert box is displayed showing some error messages. The validation is provided even in login.inc.php where the fields are check if they are empty or not.

When the user is logged in his data are stored into session. When a new user is trying to register or login his data are stored in cookies and are displayed whenever an error occurred and the page is reloaded so the client does not lose any of the previously written information. The cookies are destroyed in the moment you login or complete a successful registration. The session is destroyed when you log out.

When you register you are redirected to signup.inc.php which handle inserting new user to database. In this file it is checked if the username already exists or email is taken, or password should be 8 characters long etc... If all data are valid a new user is inserted to database.

Along with the user table that is inserted a new row another table is affected. Every new user in the database has a profile image which should be initialized in the moment of registration. (a default image is provided in case of any registration).



Accept our terms and condition before registering

Terms and Conditions

Welcome to OBTechnology!

These terms and conditions outline the rules and regulations for the use of Besjon & Orlin shpk.'s Website, located at <http://localhost/Project/login.php#>.

By accessing this website we assume you accept these terms and conditions. Do not continue to use OBTechnology if you do not agree to take all of the terms and conditions stated on this page. Our Terms and Conditions were created with the help of the [Terms And Conditions Generator](#) and the [Terms & Conditions Generator](#).

The following terminology applies to these Terms and Conditions, Privacy Statement and Disclaimer Notice and all Agreements: "Client", "You" and "Your" refers to you, the person log on this website and compliant to the Company's terms and conditions. "The Company", "Ourselves", "We", "Our" and "Us", refers to our Company. "Party", "Parties", or "Us", refers to both the Client and ourselves. All terms refer to the offer, acceptance and consideration of payment necessary to undertake the process of our assistance to the Client in the most appropriate manner for the express purpose of meeting the Client's needs in respect of provision of the Company's stated services, in accordance with and subject to, prevailing law of Netherlands. Any use of the above terminology or other words in the singular, plural, capitalization and/or he/she or they, are taken as interchangeable and therefore as referring to same.

Cookies

We employ the use of cookies. By accessing OBTechnology, you agreed to use cookies in agreement with the Besjon & Orlin shpk.'s Privacy Policy.

Most interactive websites use cookies to let us retrieve the user's details for each visit. Cookies are used by our website to enable the functionality of certain areas to make it easier for people visiting our website. Some of our affiliate/advertising partners may also use cookies.

License

ADMIN INTERFACE

EDIT YOUR PROFILE IMAGE

YOU ARE LOGGED IN AS USER #1

Choose File No file chosen

U P L O A D

D E L E T E

WELCOME TO YOUR ACCOUNT: ADMIN

EMAIL ADDRESS IS: BCIFLIKU17@EPOKA.EDU.AL

YOUR BIRTHDATE: 1999-09-06

Here you can edit your profile image. Your account information is displayed at the bottom. Chose a file from your folder and place it so anyone can see it.

MANAGE ACCOUNTS

USERNAME	EMAIL	BIRTHDATE	MANAGE
ADMIN	BCIFLIKU17@EPOKA.EDU.AL	1999-09-06	DELETE
ORLIN	OMALKJA17@EPOKA.EDU.AL	1998-12-05	DELETE
SONI	SONI@YAHOO.COM	1985-12-12	DELETE

Manage user account and you can even delete any account that you want.

Add new products to the store. A php script will insert the product in the database.

ADD PRODUCTS

CHOOSE AN IMAGE

No file chosen

Product Name

Description

Price

Quantity

INBOX

NO#	NAME	EMAIL	MESSAGE
BESJON CIFLIKU	BCIFLIKU17@EPOKA.EDU.AL	1	ALREADY READ
BESJON	BESJON@EMAIL.COM	2	READ IT
BESI	BBESI@EPLKA.COM	3	READ IT
BESJON CIFLIKU	BCIFLIKU17@EPOKA.EDU.AL	4	ALREADY READ
BESJON CIFLIKU	BCIFLIKU17@EPOKA.EDU.AL	5	ALREADY READ
BESJON CIFLIKU	BCIFLIKU17@EPOKA.EDU.AL	6	READ IT

You can even have an inbox where messages come from contact form found at about us page. Each visitor can send you a message and you can read and edit the messages.

VALIDATION

Below you can find a form of validation using JavaScript in a sign-up form where each element is checked if it is empty. The value on the password field are checked if they are same and if a checkbox that represents our terms and conditions is selected or not. This approach uses DOM methods such as *document.getElementById()* etc.

```
function fireSignup(){
    if(document.getElementById('namereg').value=== "" || document.getElementById('pw-reg').value=== "" || document.getElementById('email-reg').value=== "" || document.getElementById("bdate-reg").value==='')
    {
        alert('You have to complete all fields');
    }else{
        if(document.getElementById("pwreg").value===document.getElementById("cpw-reg").value){
            if(document.getElementById("term").checked){
                var button = document.getElementById("sing");
                button.click();
            }else{
                alert('Accept terms and condition in order to continue!');
            }
        }else{
            alert('The password field should match!');
        }
    }
}
```

In addition, you can find many other validations form and below we will give some other ways to validate data using PHP. First method is by using ready made methods such as *isset()* or *empty()* which check if data exist or not.

```
if(empty($title) || empty($desc) || empty($username) ){
    header("Location: ../userInterface.php?post=failed");
}else{
    //continue the process
}
```

Moreover, you can validate data by using regular expression or some other php functions.

```
if(!filter_var($email,FILTER_VALIDATE_EMAIL)){  
    header('Location: ../login.php?signup=invalidemail');  
}
```

Filter check if email is a valid format or not. This could be achieved even by usage of regular expression. Check the example below to see how we validate the format of email using regex. There are plenty more regex(check name,a text field to be only character and not to contain any digits) in the project but there is not enough space in this short documentation.

```
$regex = '/^[a-z0-9-]+(.[a-z0-9-]+)*@[a-z0-9-]+(.[a-z0-9-]+)*(.a  
z]{2,3})$/';  
if (!preg_match($regex, $email)) {  
    array_push($errors, "is not a valid email. We can not accept it.");  
    echo $user['email'] . " is not a valid email. We can not accept it.";  
}
```

A regular expression to check the password length and content. It should be at least 8 character long and it contains letters and numbers.

```
if (!preg_match("/^[a-zA-Z0-9 ]{8,}$/",$pass)) {  
    array_push($errors, "Only letters,number and white  
space allowed");  
    echo 'Password is not valid format';  
}
```

Making the password encrypted: By using hash function and a string with random characters (after that inserting it into database).

```
$password = $_POST['pw-reg'];  
$salt = "d5f332312e3e390c81f6ef9f242c21bf9e472d6296ddd4bebddd0f54eb576f14";  
$hpassword = hash('sha256', $salt . $password);
```

Another way to validate is to check if email exists or not, or username is already taken. First we construct a sql query and then execute that query. If the query response return something it means a user with that username or that email already exist in the database so you need to try another username.

```
$user_check_query = "SELECT * FROM users WHERE username='$uname' OR email='$email' LIMIT 1";

$result = mysqli_query($conn, $user_check_query);
$user = mysqli_fetch_assoc($result);

if ($user) { // if user exists
    //check username if it is unique
    if ($user['username'] === $uname) {
        array_push($errors, "Username already exists");
        echo 'Username exists<br>';
    }

    // Check email if it is unique
    if ($user['email'] === $email) {
        array_push($errors, "email already exists");
        echo 'Email exist<br>';
    }
}
```

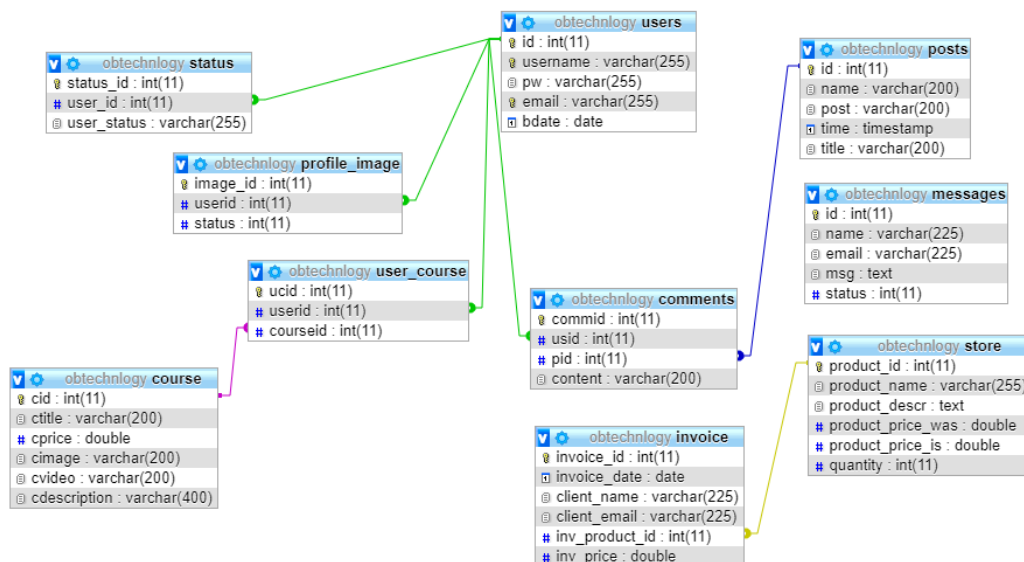
DATABASE

We have created a database, which stores the following tables:

1. Users
2. Course
3. User_course
4. Posts
5. Comments
6. Profile_image
7. Status
8. Store .
9. Invoice
10. Messages

- **Users** table will contain information regarding to a user such as : username , password, email and birthday.
- **Course** table will contain information of courses which are sold online to users. It has columns such as course title, course price , the path of the image of that course, a description and a video.
- **User_course** makes the connection of user and course, whenever a certain user buys a course , we need to keep track of the user and the course which was bought by the user . It holds the userid and courseid that each user has bought.
- **Posts** stores information related to posts that users create. It stores information such as (username of the user which made a post , time when the post was created , title and the post(the content of the post)).
- **Comments like table User_course** stores information that shows which user commented in which post. It has the following attributes(primary key , userid , postid and the content of the comment
- **Store table** stores information of a certain product (product name ,description , a previous price, current price and its quantity)
- **Messages table** stores information of a message send at contact form in about us page (name ,email , message, status) where message is the content you want to send and the status is a variable that check if the message is read or not.
- **Invoice table** stores information of a purchase made by someone visiting our website. Invoice (inv_date, client name ,client email , product id, product price)

Table	Action	Rows	Type	Collation	Size	Overhead
comments		11	InnoDB	utf8_general_ci	48 KiB	-
course		7	InnoDB	utf8_general_ci	16 KiB	-
invoice		7	InnoDB	utf8_general_ci	32 KiB	-
messages		6	InnoDB	utf8_general_ci	16 KiB	-
posts		13	InnoDB	utf8_general_ci	16 KiB	-
profile_image		3	InnoDB	utf8_general_ci	32 KiB	-
status		1	InnoDB	utf8_general_ci	32 KiB	-
store		6	InnoDB	utf8_general_ci	16 KiB	-
users		3	InnoDB	utf8_general_ci	48 KiB	-
user_course		6	InnoDB	utf8_general_ci	48 KiB	-
10 tables	Sum	63	InnoDB	utf8_general_ci	304 KiB	0 B



The store and invoice tables are not connected with user as you do not need to be logged in to access our store. You have just to complete some fields and then enjoy your order. The invoice is stored as a text file and a record is kept in database and displayed in administrator profile. Even the messages are not connected with users table as every one including visitors of the webpage can send a message to the administrator to suggest an improvement or to request a change.

PHP script to make a connection to the database:

```
<?php
    $dbServername = "localhost";
    $dbUsername = 'root';
    $dbPassword = '';
    $dbName = 'obtechnology';

    $conn = mysqli_connect($dbServername, $dbUsername, $dbPassword, $dbName);

?>
```

Another example on how we used prepared statements to insert a new user or a new product:

```
$sql2 = "INSERT INTO profile_image(userid,status) VALUES (?,?)";
$stmt = mysqli_stmt_init($conn);
if(!mysqli_stmt_prepare($stmt,$sql2)){
    echo "SQL error 2";
}else{
    echo 'inserting profile image of user';
    $zero = 0;
    mysqli_stmt_bind_param($stmt,"ss", $data['id'],$zero);
    mysqli_stmt_execute($stmt);
}
```


REFERENCES:

HTML & CSS

Design and build websites by John Ducket

Published by

John Wiley & Sons, Inc.

10475 Crosspoint Boulevard

Indianapolis, IN 46256

www.wiley.com

©2011 by John Wiley & Sons, Inc., Indianapolis, Indiana

ISBN: 978-1-118-00818-8

Manufactured in the United States of America

Published simultaneously in Canada

Learning PHP, MySQL & JavaScript

With jQuery, CSS & HTML5 by Robin Nixon

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

December 2014

W3School HTML CSS JAVASCRIPT PHP

https://www.w3schools.com/w3css/w3css_references.asp

MMUTS Channel - Introduction to PHP Programming

<https://www.youtube.com/watch?v=qVU3V0A05k8&list=PL0eyrZgxdwhwBTowajm9faF1ixePexft->

Designing input fields – Dev Ed. Channel on December 10,2018

<https://www.youtube.com/watch?v=IxRJ8vplzAo>

YouTube – Easy tutorials Channel

<https://www.youtube.com/channel/UCkjoHfkLEy7ZT4bA2myJ8xA>