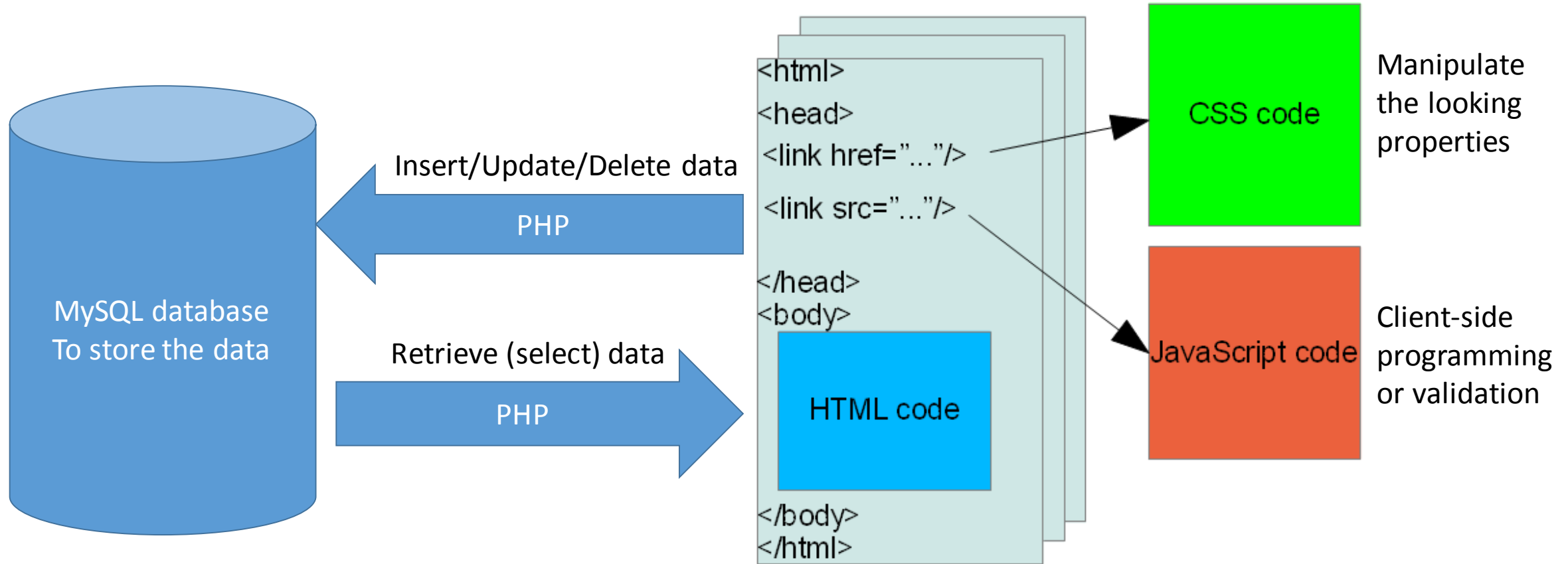


Profile Project: Putting the pieces together

0360-270 Advanced Website Design

Abed Alkhateeb

Project structure

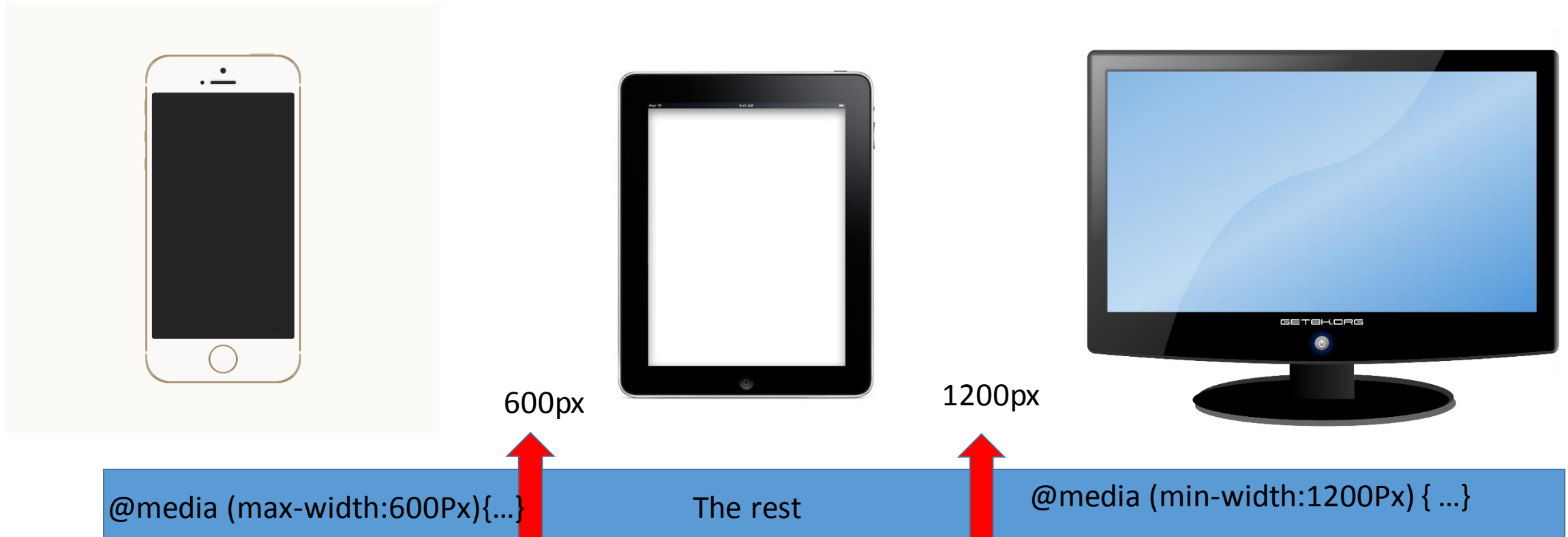


Profiles

- In this presentation , we will enhance the Profiles_form project to be responsive with more features.
- As stated earlier, you may use the profiles and user_codes structures and webpages to implement your website users system.
- Before we start, In 60270 folder, create profile_project folder.
- In profile_project folder create css, html, and js folder (all small letters).

Responsive form.css

- We will implement a responsive form, with breaking points 600px and 1200px as following:



60270/profile_project/css/form.css

- First implementing "the rest" in the previous figure for tablets
- For this part, it will work on the tablets and the not specified properties in the other sections(smartphones and widerscreens > 1200px)

Tablet and common features form.css

- Common features

```
body {padding: 20px 15%;}  
input[type=submit] {  
  width: 100%;  
  background-color: #4CAF50;  
  color: white;  
  padding: 14px 20px;  
  margin: 8px 0;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;}  
input[type=submit]:hover {  
  background-color: #45a049;}
```

- Tablet features

```
input[type=text],  
input[type=email],  
input[type=url],  
input[type=password],  
textarea,  
select {  
  height: 50px;  
  width: 80%;  
  margin: 8px 0;  
  display: inline-block;  
  border: 1px solid #ccc;  
  border-radius: 4px;  
  box-sizing: border-box;}
```

```
input[type=text]:focus,  
input[type=email]:focus,  
input[type=url]:focus,  
input[type=password]:focus,  
textarea:focus {  
  outline: 0;  
  border-color: #4697e4;  
}  
form > div > label,  
legend {  
  font-size: 1.3em;  
  float: left;  
  padding-right: 10px;}
```

Specifying smartphone properties form.css

```
@media (max-width: 600px){
```

```
  form > div > label,  
  legend {  
color:red;  
  width: 100%;  
  float: none;  
  margin: 0 0 5px 0;  
  text-align: right;  
font-size: 1.2em;  
  }
```

```
form > div{  
  padding: 5px 5%;  
border-radius: 5px;  
  background-color: #f2f2f2;}
```

```
input[type=text],  
input[type=email],  
input[type=url],  
input[type=password],  
textarea,  
select {  
  height: 50px;  
  width: 100%;  
margin: 8px 0;  
  display: inline-block;  
  border: 1px solid #ccc;  
  border-radius: 4px;  
  box-sizing: border-box; }  
}
```

Specifying tablet properties form.css

```
@media (min-width: 1200Px) {  
  form > div > label{  
    text-align: right;  
    font-size: 1.5em;  
padding-top: 7px;  
}  
  form > div{  
    padding: 5px 5%;  
}  
  
  input[type=text],  
  input[type=email],  
  input[type=url],  
  input[type=password],  
  textarea,  
  select {  
height: 30px;  
width: 50%;  
text-align: left;  
border: 1px solid #58595b;}  
}
```


Database changes

- Adding 2 columns to profiles table, email and password.

The screenshot shows the phpMyAdmin interface for the 'alkhate_db' database, specifically the 'Profiles' table structure. The table has 7 columns: id, fname, lname, picture, usercode, DOB, and created_date. The 'Add' button at the bottom is highlighted, and the 'Go' button is also highlighted.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(10)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	fname	varchar(30)	utf8_unicode_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	lname	varchar(30)	utf8_unicode_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	picture	blob			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	usercode	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
6	DOB	date			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
7	created_date	timestamp			No	CURRENT_TIMESTAMP		Change Drop Primary Unique Index Spatial Fulltext Distinct values

At the bottom of the interface, the 'Add' button is highlighted, and the 'Go' button is also highlighted.

Database changes

The screenshot shows the phpMyAdmin interface for a database named 'alkhate_db'. The 'Profiles' table is selected, and its structure is displayed. The table has two columns: 'email' and 'password'. Both columns are of type 'VARCHAR'. The 'email' column has a length of 100, and the 'password' column has a length of 30. Both columns have a default value of 'None'. The 'email' column is highlighted with a red box, and the 'password' column is also highlighted with a red box. The 'Save' button at the bottom right is also highlighted with a red box.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments
email	VARCHAR	100	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
password	VARCHAR	30	None			<input type="checkbox"/>	---	<input type="checkbox"/>	

Preview SQL Save

profile_form.php changes

- The main changes are :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta name="viewport" content="width=device-width, initial-  
scale=1.0">
```

// the previous command tells the device to treat the actual device width as the screen width so pages you design for a certain size will render as expected.

profile_form.php changes

```
<link rel="stylesheet" type="text/css" href="../css/form.css" >
```

```
// link to external css file, where we implement css changes
```

```
<script src="../js/validation.js"> </script>
```

```
//link to external js file, where we script the browser-side functions,
```

```
// like validating matching password
```

```
</head>
```

profiles_form.php changes

- Associate each form element with it's label inside one <div>, this will unite them both in one structure so we can divide/control this structure in flexible way as in the previous css code.

```
<div>
```

```
<label for="fname">First Name</label>
```

```
<input type="text" required name="fname"><br>
```

```
</div>
```

profiles_form.php changes

- Adding new 3 forms elements for email, password, and confirm password fields:

```
<div>
```

```
<label for="email">Email</label>
```

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

```
</div>
```

```
<div>
```

```
<label
for="pwd">Password</label><input
type="password" id="pwd"
name="pwd">
```

```
</div>
```

```
<div>
```

```
<label for="cpwd">Confirm
password</label>
```

```
<input type="password" id="cpwd"
name="cpwd">
```

```
</div>
```

profiles_form.php changes

- Calling js function to check if password and confirm password fields match, and/ or any, both are not empty.
- `<form action="insert_into_profiles.php" onsubmit="return validate_password()" method="post" enctype="multipart/form-data">`

Implement js/validation.js

```
function validate_password(){  
    var p1 = document.getElementById("pwd").value;  
    var p2 = document.getElementById("cpwd").value;  
    if (p1 == "" || p2 == ""){  
        alert("password or confirm password field is not entered");  
        return false;  
    }  
    if (p1 == p2){  
        return true;  
    } else {  
        alert("password and confirm password fields are mismatch");  
        return false;  
    }  
}
```


insert_into_profiles.php changes

- // receiving 2 form elements sent by profiles_form

```
$lname = $_REQUEST['lname'];
```

```
$email = $_REQUEST['email'];
```

```
$pwd = $_REQUEST['pwd'];
```

insert_into_profiles.php changes

- Adding the received 2 elements to sql statement
- `$sql_insert = "INSERT INTO Profiles (id, fname, lname, DOB, picture, usercode, email, password)`
The 2 elements which are created in profile table
- `VALUES (". $id.", " ". $fname.", " ". $lname.", STR_TO_DATE(" ". $dob.", '%Y-%m-%d'), " ". $data.", " ". $user_code.", " ". $email.", " ". $pwd."");`
The 2 variables where the 2 form elements values are recieved
-

Full code

- Try it first
, http://alkhate.myweb.cs.uwindsor.ca/profile_project/html/profiles_form.php
- You may
use https://www.mobilemoxie.com/tools/mobile_handset_emulator/
to try it on different devices (screen sizes)
- Full code can be downloaded from 270 blackboard site > resources menu.