

RICHFIELD
FACULTY OF INFORMATION TECHNOLOGY
INTERNET PROGRAMMING 622
4TH SEMESTER ASSIGNMENT

Name & Surname: Lourens Van Der Bijl ICAS No: 140111

Qualification: BSCIT Semester: 4 Module Name: Internet Programming 622

Date Submitted: 2020-10-09

ASSESSMENT CRITERIA	MARK ALLOCATION	EXAMINER MARKS	MODERATOR MARKS
MARKS FOR CONTENT			
QUESTION ONE	80		
QUESTION TWO	20		
TOTAL MARKS	100		
Examiner's Comments:			
Moderator's Comments:			
Signature of Examiner:		Signature of Moderator:	

Table Of Content

QUESTIONS ONE	3
<i>Social Media application visual view</i>	4
<i>Social Media application Database</i>	28
The Database	28
The tables	29
The messaging table	30
The email password recovery table	31
The posts table	33
The users table	35
<i>Social media app CODE</i>	37
index.php	37
db.php (database connection)	38
registration.php	39
login.php	43
passwordRec.php	46
reset.php	50
user.php	52
.htaccess	54
header.php	54
dashboard.php	57
profile.php	60
messageSystem.php	64
update.php	68
upload.php	70
logout.php	73
style.css	74
hstyle.css	78
QUESTIONS TWO	87
<i>References</i>	90

QUESTIONS ONE

[80 MARKS]

Social media refers to websites and applications that are designed to allow people to share content quickly, efficiently, and in real-time. Many people define social media as apps on their smartphone or tablet.

The ability to share photos, opinions, events, etc in real-time has transformed the way we live and, also, the way we do business.

Based on the existing social media (Facebook, Twitter, Instagram) create your own social media to connect people and businesses.

Your website should do the following:

1. Have a login system that allow a user to login or register (login will only be granted if the user credential were found in the database)

Login system located on login.php

2. On the register form get the user basic details and store them in the database.

Registration located on registration.php

3. Allow user to login in the system based on their email and password Note:(Email should be unique, if the user is already registered in the database provide option such as login or forget password to allow him to login)

On the Login form an option has been provides as to recover a password and to redirect you to the registration page. Users register based on a unique username. If the same username already exist the system will assign a value on the end of the user name to make it unique

4. Once logged in create a user timeline where the user can share post, share picture.

User timeline is located on the dashboard.php

5. The user should also have a profile page (profile pictures, account holder name, etc)

**When a User register a default profile picture is being assign to them the user can then go to the settings page and update their profile picture to whatever they want it to be
Located on profile.php & upload.php**

6. Provide an option for the user to see another user profile (profile picture, timeline....)

Users can use the search function in the header.php to locate other users and view their profiles

7. Finally allow the user to send a message to another user who is using your social media (Allow them to send messages to each other hint: user a search button to search for a particular user)

When the user search for another user in the search bar in will view the searched users profile from there the logged in user can sent a message to the user being search. The messaging system is located on messageSystem.php

8. Using your personal discretion make the social media beautiful by enhancing it with CSS3, Javascript or JQuery.

The Login and register page CSS can be located on the style.css the rest of the website styling was done on hstyle.css

Social Media application visual view

	Name	Date modified	Type	Size
	css	2020/09/10 12:48 PM	File folder	
	images	2020/10/08 4:12 PM	File folder	
	include	2020/09/19 6:46 AM	File folder	
	uploads	2020/10/08 9:29 PM	File folder	
	.htaccess	2020/09/18 8:49 PM	HTACCESS File	1 KB
	dashboard	2020/10/08 4:18 PM	PHP File	6 KB
	db	2020/10/07 12:12 AM	PHP File	1 KB
	header	2020/10/08 1:23 PM	PHP File	4 KB
	index	2020/09/27 2:56 PM	PHP File	2 KB
	login	2020/09/26 12:21 AM	PHP File	4 KB
	logout	2020/09/08 2:53 PM	PHP File	1 KB
	messageSystem	2020/10/06 9:18 PM	PHP File	6 KB
	passwordRec	2020/10/06 10:04 PM	PHP File	6 KB
	profile	2020/10/06 9:17 PM	PHP File	5 KB
	register	2015/07/30 7:57 AM	SQL File	2 KB
	registration	2020/10/07 1:33 AM	PHP File	6 KB
	reset	2020/09/27 12:25 PM	PHP File	3 KB
	update	2020/10/08 12:10 PM	PHP File	3 KB
	upload	2020/10/07 1:23 AM	PHP File	3 KB

Social Degree Home

localhost/SocialMedaiApp/

Social Degree

Welcome to Social Degree. This is a social network platform based on Facebook, LinkedIn.
The idea of this platform is to motivate and aid students in their studies.
Students will be able to share ideas and study material that is needed to perform
in one's career

Already have an account? [Log in](#)

Create a new account? [Register](#)

6 default users has been created

Username	Password
Bart	Bart
Faith	Faith
Lilo	Lilo
Lisa	Lisa
Lourens	Lourens
Michayla	Mickhayla

The first page the user will see when clicking on the website is
the index page. Where there is a short introduction, and two
button links asking a user whether he want to login as an existing
user or create a new account

Please notice the hyperlink if a user miss clicked and meant to click login instead

Registration

Already a member click here!

Username
Email
Password
Name
Surname
Age
Gender
Country
City

Add a profile picture on settings Tab

Register

For the purpose of this demonstration we will create a new user with the following information

Username:	Moodle
Email:	richfield@gmail.com
Password:	InternetProgramming
Name:	Richfield
Surname:	Institution
Age:	2years
Gender:	Male
Country:	South Africa
City:	Durban

When a user register a new account a default profile pic will be assign to him. This profile pic can be change in the update information tab

Registration

Already a member click here!

Moodle

richfield@gmail.com

.....

Richfield

Institution

2

Male

South Africa

Durban|

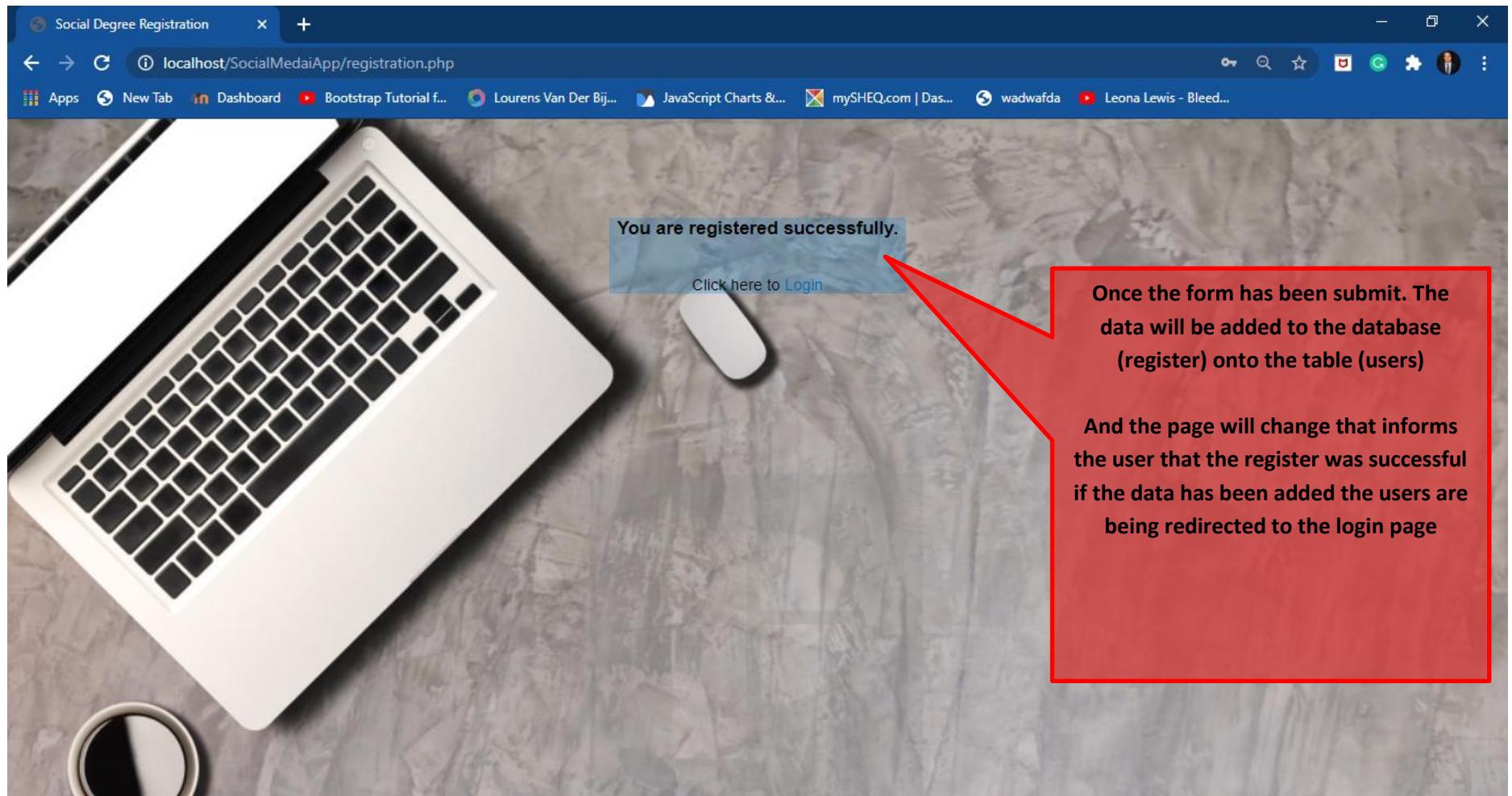
Add a profile picture on settings Tab

Register

With the HTML type and required field we put on some validations to this form. The form will not allow to be submit if these rules has not been met or if some information is empty. For example the email field need to have an @ symbol and have text in front and at the back of the symbol

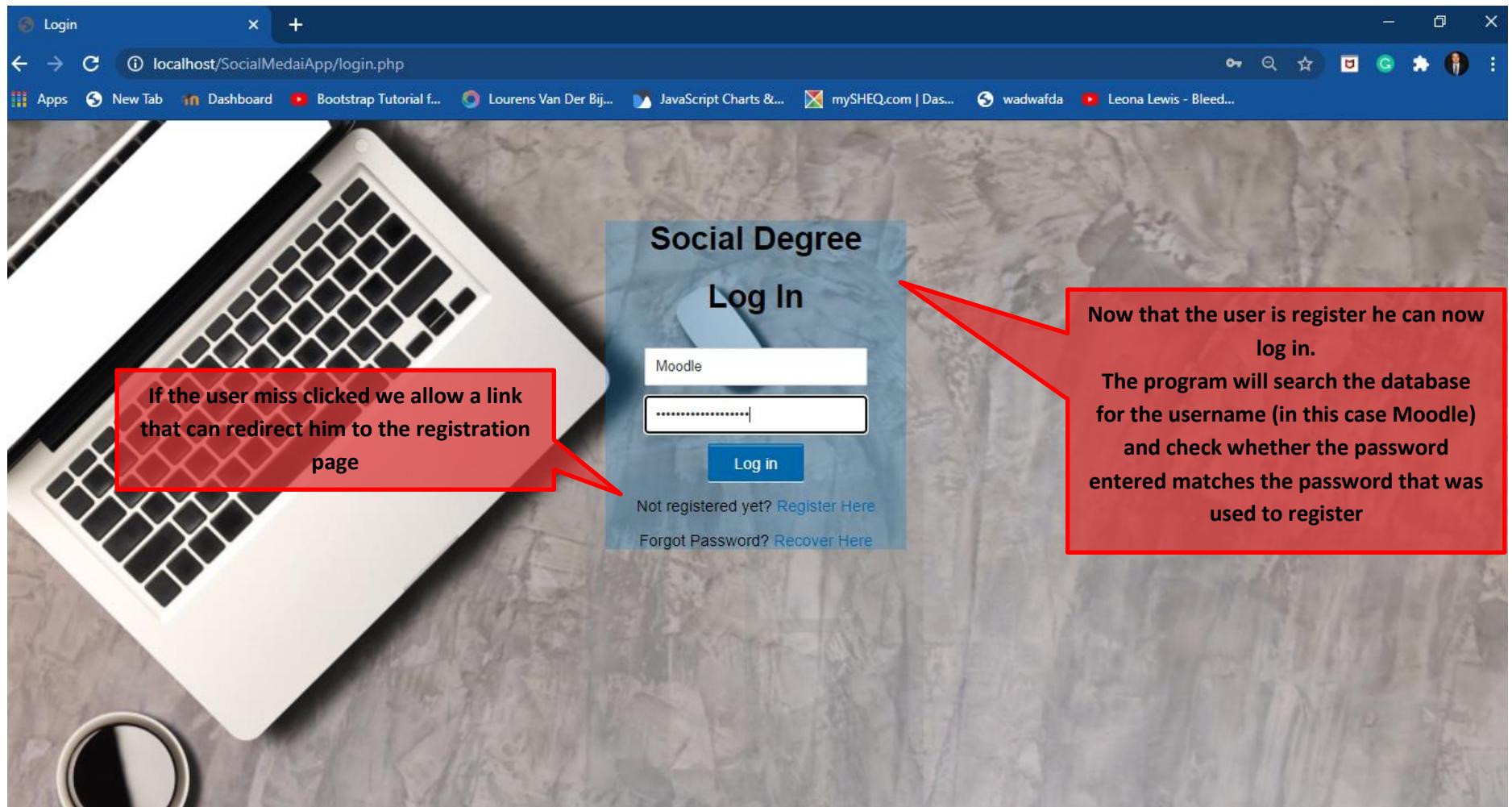
If the username Moodle already exist an integer value will be added to the username

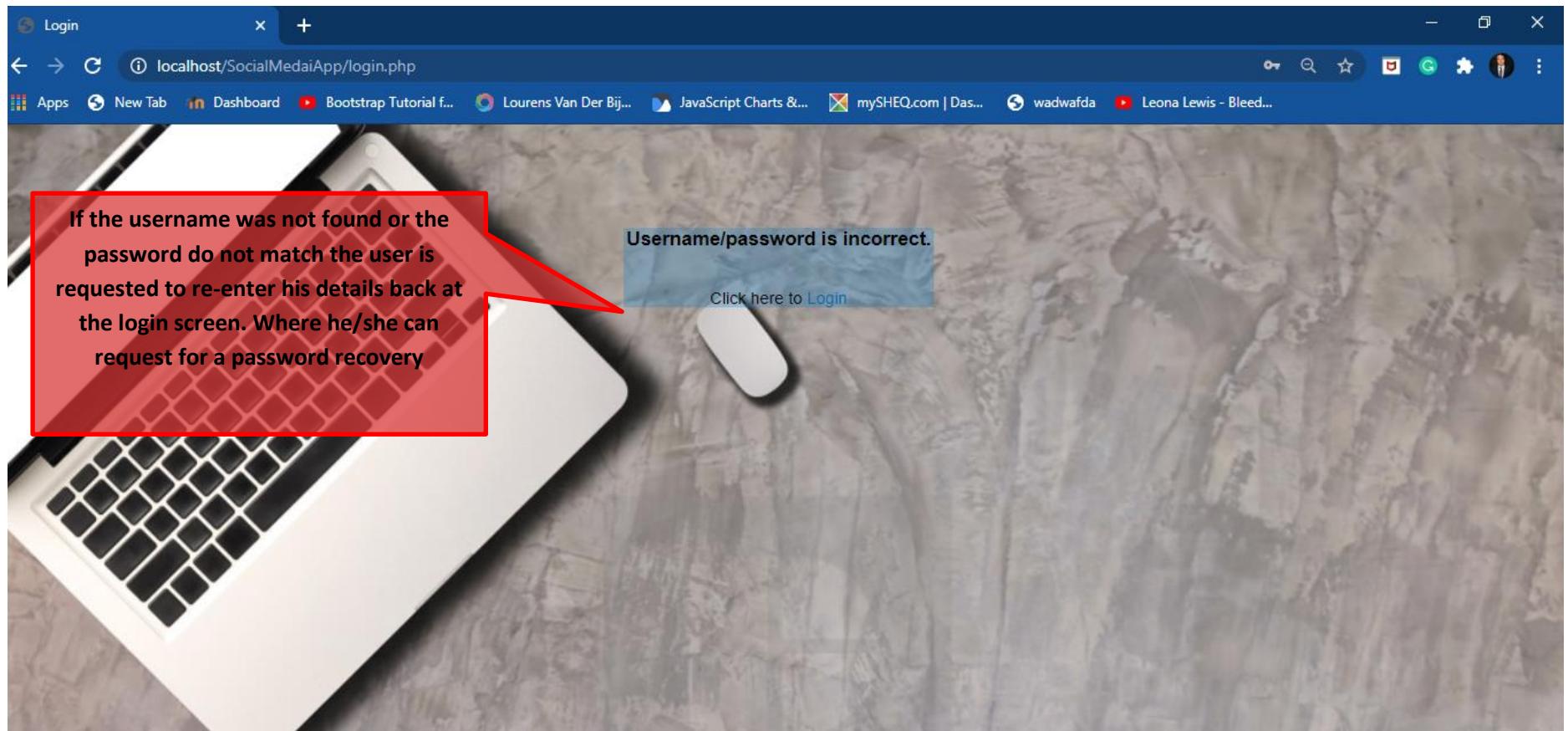
The age is a scrollable number field and will only except integer values



Once the form has been submit. The data will be added to the database (register) onto the table (users)

And the page will change that informs the user that the register was successful if the data has been added the users are being redirected to the login page





Social Degree Password Recovery

Enter Your Email Address:

username@email.com

Reset Password

This is a feature I added that was not part of the assignment but was done anyway.

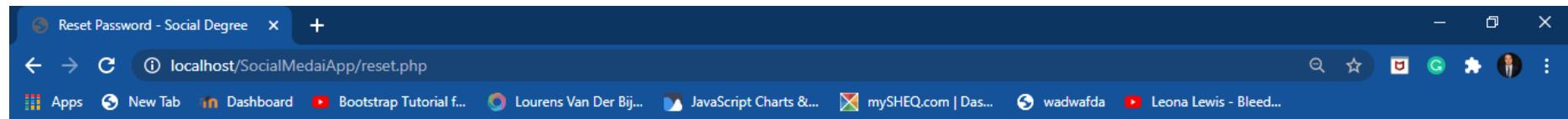
This page will request the user to enter his email address then a Temp password and link to another page will be sent to him via email. Code can be found on the passwordRec.php page.

With this code the application automatically creates an email and sent it to the user

This feature is currently avoided by the use of a hyperlink that direct the user to the page where he can change his password due to the fact that I do not have a mail server and the time consuming for lectures to watch their email when testing this code

If you wish to test this code simply remove the hyperlink and add your mail server details in the code where the comments request it

Hyperlink set up on the button



When the hyperlink is clicked. (Or if you used the mail system the link sent via mail) you will be redirected to this page where the application will search for the email in the users table and update the password

This code can be found on reset.php

Social degree Reset Password

Account Email

New Password

Change Password

Not registered yet? [Register Here](#)

Back to Log in? [Log in Here](#)

A default profile picture has been assigned to him that he/she can change in the update.php or simply clicking on the tools icon or on the change button below the profile picture

Want to post something??

posting

Once the user manages to log in his personal information is pulled from the database to display as his profile

The screenshot shows a web browser window titled "Welcome to Social Degree" with the URL "localhost/SocialMedaiApp/dashboard.php". The dashboard has a blue header with the "Social Degree" logo, search bars for "Search by name" and "Search user", and a user account section for "Richfield Institution". On the left, there's a profile box for a user named "Richfield Institution" (Gender: Male, Age: 2) with a placeholder "Default" profile picture. A red callout box points to this area with the text: "A default profile picture has been assigned to him that he/she can change in the update.php or simply clicking on the tools icon or on the change button below the profile picture". In the center, there's a box asking "Want to post something??". To its right, a blue button says "posting". Below these, two recent posts are displayed in blue boxes. The first post is by "Bart Simpson" (Profile pic: Bart Simpson, Date: 2020-10-07 00:27:08) with the message "How are you?". The second post is by "Mickayla Greed" (Profile pic: Mickayla Greed, Date: 2020-09-27 15:43:47) with the message "This is such a cool app made me wish that I am studying something". A red callout box points to the user information in the first post with the text: "Once the user manages to log in his personal information is pulled from the database to display as his profile".

Welcome to Social Degree

localhost/SocialMedaiApp/update.php

Richfield Institution

User Details update

Change Password: [Click here](#)

Change Name: Richfield

Change Surname: Institution

Change Age: 2

Change Gender: Male

Change Country: South Africa

Change City: Durban

Update

Please refresh to see updates

Change Profile Picture: Choose File | No file chosen

Update

If the user wish to change his password he will be redirected to the reset.php page where the password recovery can be done

As you can see the current values of the user is currently displaying in the fields. If a user wishes to change these values he can simply replace these values with those he want and hit update

Here the user can browse for a picture file validation is located in upload.php and then hit update to save it to the database

When a user click on the tools icon or on the change button below his profile picture he will be directed here were he can change some of the information that is not sensitive to the program. For example we do not allow a user to change his username or Email since it can become a security risk to the application

Please note that a refresh might be necessary to display the new values in the fields although it is already updated in the database

Welcome to Social Degree

localhost/SocialMedaiApp/update.php

Richfield Institution

User Details update

Change Password: [Click here](#)

Change Name:

Change Surname:

Change Age:

Change Gender:

Change Country:

Change City:

Update

Please refresh to see updates

Change Profile Picture: No file chosen

For this example we will change the:

Age:	4
Gender:	Female
Country	International

Please note that this application will allow you to change all the values at the same time or you can pick a few and it will only update those you have change

Press this update button to update the personal information

For this example we will change the:

Age:	4
Gender:	Female
Country	International

Please note that this application will allow you to change all the values at the same time or you can pick a few and it will only update those you have change

Press this update button to update the personal information

Welcome to Social Degree

localhost/SocialMediaApp/dashboard.php

Richfield Institution

Want to post something??

posting

As you can see your information has been updated if you wish to see whether country has also been update please visit your update page again

Default

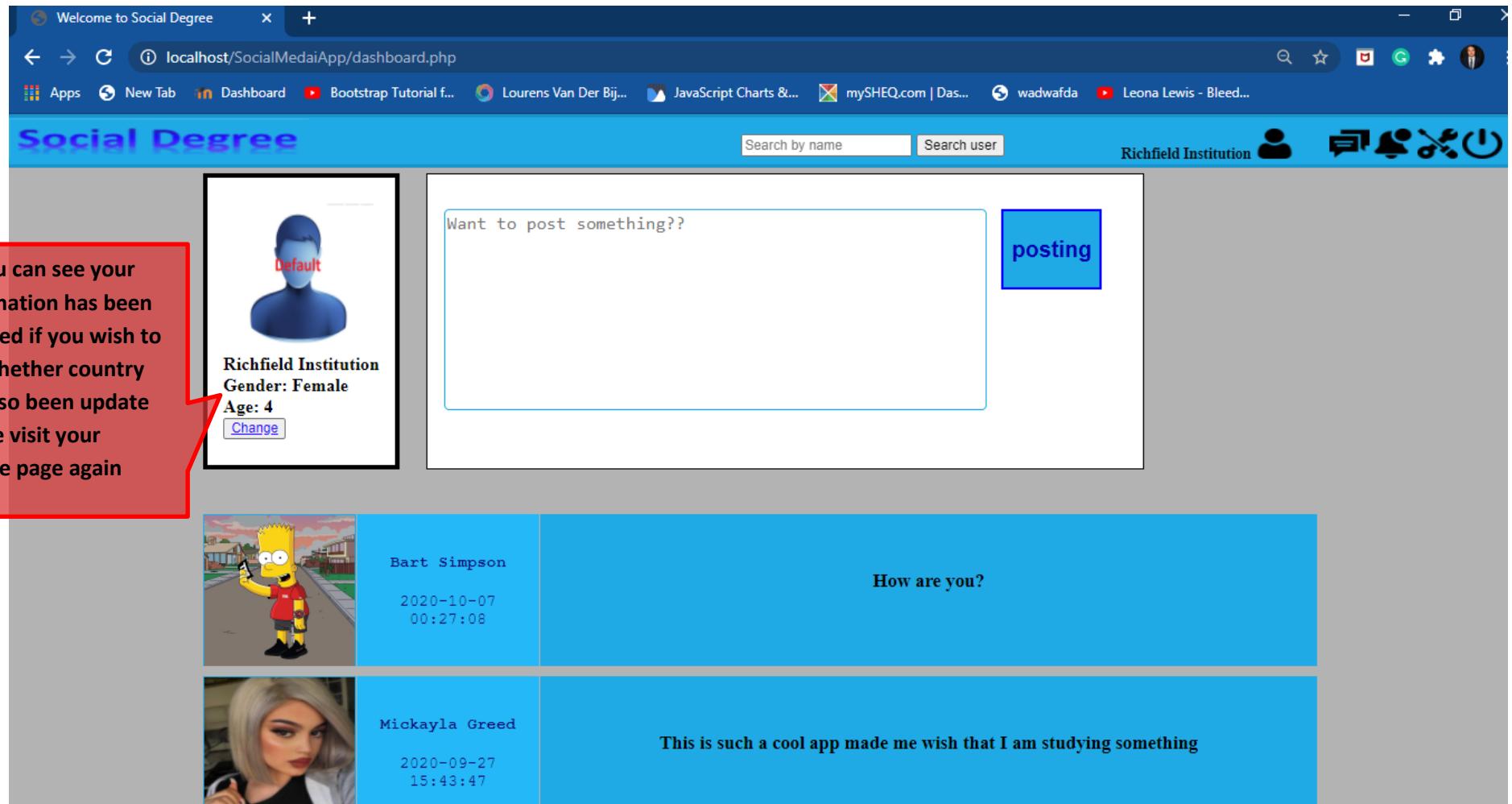
Richfield Institution
Gender: Female
Age: 4
[Change](#)

Bart Simpson
2020-10-07
00:27:08

How are you?

Mickayla Greed
2020-09-27
15:43:47

This is such a cool app made me wish that I am studying something



The screenshot shows a web-based social media application named "Social Degree". At the top, there's a navigation bar with links like "Welcome to Social Degree", "localhost/SocialMediaApp/dashboard.php", "Richfield Institution", and various system icons. Below the header, the main interface features a user profile section on the left with a placeholder image labeled "Default", the name "Richfield Institution", gender "Female", age "4", and a "Change" button. To the right is a large white area with a blue border containing the text "Want to post something??". A blue rectangular box with the word "posting" is overlaid on the right side of this area. On the far left, a red rectangular callout box contains the text: "As you can see your information has been updated if you wish to see whether country has also been update please visit your update page again". Below the profile section, there are two recent post cards. The first post is from "Bart Simpson" (image of Bart Simpson) with the timestamp "2020-10-07 00:27:08" and the message "How are you?". The second post is from "Mickayla Greed" (image of Mickayla Greed) with the timestamp "2020-09-27 15:43:47" and the message "This is such a cool app made me wish that I am studying something".

I will now download an image from google as the Richfield profile picture and save it in my downloads folder. The idea of the file system is to save the picture a user upload on to the upload folder via the program

Google search results for "richfield." showing various Richfield-related images. A red callout box points to the first image result.

Windows File Explorer showing a download of a Richfield logo file (5862ab4b9c34806f).

The screenshot shows a web browser window with the title "Welcome to Social Degree". The address bar displays "localhost/SocialMedaiApp/update.php". The page header includes a search bar, a user profile for "Richfield Institution", and various icons. The main content area is titled "User Details update". It contains fields for changing name, surname, age, gender, country, and city, each with a corresponding input field. Below these fields is a red "Update" button. A note below the buttons says "Please refresh to see updates". At the bottom, there is a section for changing the profile picture, featuring a "Choose File" button and a file path "5862ab4b9c34806f.jpg", followed by another red "Update" button. A red callout box on the left side of the page contains the following text:

We can now upload the previous picture using the file system. The application will save the name of the file in the database and create a copy of the image in the uploads folder where my application is located

As you can see the profile picture has now been updated and is now save in the database and in the applications uploads folder

The screenshot shows a web application interface titled "Social Degree". At the top, there's a navigation bar with links like "Welcome to Social Degree", "localhost/SocialMedaiApp/dashboard.php", "Apps", "New Tab", "Dashboard", "Bootstrap Tutorial f...", "Lourens Van Der Bij...", "JavaScript Charts &...", "mySHEQ.com | Das...", "wadwafda", and "Leona Lewis - Bleed...". Below the navigation bar, the main content area has a blue header with "Social Degree" and search bars for "Search by name" and "Search user". A user profile for "Richfield Institution" is shown, with gender "Female" and age "4". To the right, a box contains the text "Want to post something??" with a blue box labeled "posting" next to it. Below this, there are two user posts: one from "Bart Simpson" dated "2020-10-07 00:27:08" and another from "Mickayla" dated "2020-09-15:43". At the bottom, a file explorer window titled "uploads" shows a folder structure under "Acer (C) > MAMP > htdocs > SocialMedaiApp > uploads". It lists several files and folders, including "5862ab4b9c34806f", "Bart", "default", "download", "Faith", "Lito", "Lisa", "Mickayla", "profile_Pic1", and "unnamed".

Welcome to Social Degree

localhost/SocialMedaiApp/dashboard.php

Richfield Institution

Search by name | Search user

This is a post from Richfield Showing that this is working

posting

Now that our profile page is correct we can start posting. For this example we will post "This is a post from Richfield to show that this is working" and hit posting to post

Bart Simpson
2020-10-07 00:27:08
How are you?

Mickayla Greed
2020-09-27 15:43:47
This is such a cool app made me wish that I am studying something

The screenshot shows a web application interface titled "Social Degree". At the top, there's a navigation bar with tabs like "Welcome to Social Degree", "localhost/SocialMedaiApp/dashboard.php", and various browser icons. Below the navigation is a header with the "Social Degree" logo and search bars for "Search by name" and "Search user". The main content area is titled "Richfield Institution". On the left, there's a sidebar with the Richfield logo and text: "Richfield Institution", "Gender: Female", "Age: 4", and a "Change" button. The main feed area contains a post from "Bart Simpson" dated "2020-10-07 00:27:08" with the message "How are you?". Below it is a post from "Mickayla Greed" dated "2020-09-27 15:43:47" with the message "This is such a cool app made me wish that I am studying something". A red callout box points to the "posting" button next to Bart's post, and another red callout box on the right provides instructions for posting.

Social Degree

Welcome to Social Degree

localhost/SocialMedaiApp/dashboard.php

Richfield Institution

Want to post something??

posting

As you can see the post appears on the dashboard where all the users of this applications post are from the newest to the oldest

Informing the user that the post has been added to the database on table (posts)

Post added to your profile

This is a post from Richfield Showing that this is working

Richfield Institution

2020-10-09
11:32:46

Bart Simpson

2020-10-07
00:27:08

How are you?

The screenshot shows a web browser window with the title 'Welcome to Social Degree' and the URL 'localhost/SocialMedaiApp/dashboard.php'. The dashboard has a blue header with the 'Social Degree' logo and a search bar. Below the header, there's a profile section for 'Richfield Institution' showing a placeholder profile picture, gender (Female), and age (4). A button labeled 'Change' is present. To the right, there's a large text input field with the placeholder 'Want to post something??' and a blue button labeled 'posting'. A red callout box with a black border and white text points to this area, containing the text 'Informing the user that the post has been added to the database on table (posts)' and 'Post added to your profile'. Below this, there are two posts listed in a table format. The first post is from 'Richfield Institution' with the timestamp '2020-10-09 11:32:46'. The second post is from 'Bart Simpson' with the timestamp '2020-10-07 00:27:08'. A red callout box points to the newest post from 'Richfield Institution', containing the text 'This is a post from Richfield Showing that this is working'. A large red arrow points from the text 'As you can see the post appears on the dashboard where all the users of this applications post are from the newest to the oldest' towards the newest post.

Welcome to Social Degree

localhost/SocialMedaiApp/dashboard.php

Richfield Institution

Want to post something??

posting

Post added to your profile

Richfield Institution

2020-10-09
11:32:46

This is a post from Richfield Showing that this is working

Bart Simpson

2020-10-07
00:27:08

How are you?

Although this is all very cool what if we want to see only the post made by a certain user.

Easy we simply search the user in the search box to view his profile. In this example we will search for user Bart Simpson

Search by name Search user

Social Degree

Richfield Institution

Want to post something??

posting

Post added to your profile

Richfield Institution

2020-10-09
11:32:46

This is a post from Richfield Showing that this is working

Bart Simpson

2020-10-07
00:27:08

How are you?

Welcome to Social Degree

localhost/SocialMedaiApp/profile.php

Richfield Institution

This is the profile page of:

Bart Simpson

Gender: Male
Age: 17
Country: America
City: Springfield
User since: 2020-09-27 14:58:17

Type message to Bart

Sent Message

We are now viewing the profile page of Bart Simpson and only displaying the post that he made

Bart Simpson
2020-09-27 15:05:56
Hello Social Nerd.

Bart Simpson
2020-10-07 00:27:08
How are you?

Let's say we really enjoy Bart's post and would like to send him a private message say "Keep up the funny posts"

We can simply enter it in the sent message field..

Once the user has enter a message (will not allow empty) and hit Sent Message he will be directed to his inbox and outbox page

Welcome to Social Degree

localhost/SocialMedaiApp/messageSystem.php

Message sent

Inbox

Outbox

To: Bart Simpson
From: Richfield Institution
Date: 2020-10-09 11:43:39

Keep up the funny posts

Currently Richfield has one outbox message and no Inbox message let us now log on to Bart's Profile and response to Richfield's message

The screenshot shows a web browser window titled "Welcome to Social Degree" with the URL "localhost/SocialMedaiApp/dashboard.php". The dashboard has a blue header with the title "Social Degree" and a search bar. On the left, there's a sidebar with a profile picture of Bart Simpson, his gender (Male), age (17), and a "Change" button. A red callout box points to this area with the text: "First of all now that we are on the log in of Bart Simpson notice also how Richfield's post is also display on his profile". In the center, there's a posting area with a placeholder "Want to post something??". To the right, there's a blue button labeled "posting". Below this, there are two posts. The first post is from "Richfield Institution" dated "2020-10-09 11:32:46" with the content "This is a post from Richfield Showing that this is working". The second post is from "Bart Simpson" dated "2020-10-07 00:27:08" with the content "How are you?". A red callout box points to the messaging icon in the top right corner with the text: "Bart can view his inbox by clicking on the messaging icon".

First of all now that we are on the log in of Bart Simpson notice also how Richfield's post is also display on his profile

Bart can view his inbox by clicking on the messaging icon

As you can see the message from Richfield has been added to Bart's Inbox

Inbox		Outbox	
From: Richfield Institution To: Bart Simpson Date: 2020-10-09 11:43:39	Keep up the funny posts	To: Mickayla Greed From: Bart Simpson Date: 2020-09-27 15:09:54	Hey hot stuff you want to come study with me?
From: Lourens Van Der Bijl To: Bart Simpson Date: 2020-10-08 13:21:02	kbku	To: Lourens Van Der Bijl From: Bart Simpson Date: 2020-09-27 15:09:11	Can we please get more people on this app!!
From: Mickayla Greed To: Bart Simpson Date: 2020-09-27 15:44:58	Hello Cuttie	To: Lisa Simpson From: Bart Simpson Date: 2020-09-27 15:08:37	Hey sis your profile pic looks like me!!!
From: Lourens Van Der Bijl To: Bart Simpson Date: 2020-09-27 15:41:20	As we are entering our examination season I trust that a Social Degree will make a difference on your way forward. As the developer of this social application I would just like to wish you all good luck. Please feel free to share your experience with the examinations with the other users Kind regards Lourens Van Der Bijl	To: Lilo Stich From: Bart Simpson Date: 2020-09-27 15:07:56	You are a cartoon character. Fake profile alert.
From: Lisa Simpson To: Bart Simpson Date: 2020-09-27 15:27:53	Don't talk to me!! Don't let any one know that we are related.	To: Faith Marura From: Bart Simpson Date: 2020-09-27 15:06:58	Hello I am Bart Simpson. What are you studying?

Welcome to Social Degree

localhost/SocialMedaiApp/dashboard.php

Richfield Institution

Want to post something??

posting

Richfield Institution

2020-10-09
11:32:46

This is a post from Richfield Showing that this is working

Bart Simpson

2020-10-07
00:27:08

How are you?

Back on the Richfield page the last thing we need to address is the log out.. The Log out button will redirect the user to the index page and destroy all the Session variables for security purposes

Richfield Institution

Gender: Female

Age: 4

Change

Want to post something??

posting

Richfield Institution

2020-10-09
11:32:46

This is a post from Richfield Showing that this is working

Bart Simpson

2020-10-07
00:27:08

How are you?

Back on the Richfield page the last thing we need to address is the log out.. The Log out button will redirect the user to the index page and destroy all the Session variables for security purposes

Social Media application Database

The Database

The screenshot shows the phpMyAdmin interface on a web browser. The title bar indicates the URL is `localhost / localhost | phpMyAdmin`. The main navigation bar includes links for Apps, New Tab, Dashboard, Bootstrap Tutorial f..., Lourens Van Der Bij..., JavaScript Charts &..., mySHEQ.com | Das..., wadwafda, and Leona Lewis - Bleed... The top menu bar has tabs for Databases, SQL, Status, User accounts, Export (which is selected), Import, Settings, Replication, Variables,Charsets, Engines, and Plugins. On the left, a tree view lists databases: New, information_schema, mysql, performance_schema, register (circled in red), NEW, messaging, password_reset_temp, posts, users, social, sys, and test. The central content area is titled "Exporting databases from the current server". It contains sections for "Export method:" (with "Quick - display only the minimal options" selected) and "Format:" (set to "SQL"). A "Go" button is at the bottom.

```
-- Database: `register`  
--  
CREATE DATABASE IF NOT EXISTS `register` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;  
USE `register`;
```

The tables

The screenshot shows the phpMyAdmin interface for the 'register' database. The left sidebar lists databases, and the main area shows the structure of the 'register' database with four tables: messaging, password_reset_temp, posts, and users.

Table	Action	Rows	Type	Collation	Size	Overhead
messaging	Browse Structure Search Insert Empty Drop	32	InnoDB	utf8_general_ci	16.0 Kib	-
password_reset_temp	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 Kib	-
posts	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8_general_ci	16.0 Kib	-
users	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8_general_ci	32.0 Kib	-
4 tables	Sum	48	InnoDB	utf8_general_ci	80.0 Kib	0 B

Below the table list, there are buttons for 'Print' and 'Data dictionary', and a 'Create table' form with fields for 'Name:' (empty) and 'Number of columns:' (set to 4). A 'Go' button is located at the bottom right of the form.

The messaging table

This Table is used to store the messages of all the users

The screenshot shows the phpMyAdmin interface for the 'register' database. The left sidebar lists databases and tables, with 'messaging' selected. The main area shows the 'Structure' tab for the 'messaging' table. The table has 8 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)	utf8_general_ci		No	None		AUTO_INCREMENT	Change Drop More
2	name_from	varchar(25)	utf8_general_ci		No	None			Change Drop More
3	surname_from	varchar(45)	utf8_general_ci		No	None			Change Drop More
4	name_to	varchar(25)	utf8_general_ci		No	None			Change Drop More
5	surname_to	varchar(45)	utf8_general_ci		No	None			Change Drop More
6	message_sent	text	utf8_general_ci		No				Change Drop More
7	date_send	datetime			No	None			Change Drop More
8	Sender_pic	varchar(255)	utf8_general_ci		No	None			Change Drop More

Below the table structure, there is a section for indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	25	A	No	

-- Table structure for table `messaging`

```
--  
CREATE TABLE `messaging` (  
  `id` int(11) NOT NULL,  
  `name_from` varchar(25) NOT NULL,  
  `surname_from` varchar(45) NOT NULL,  
  `name_to` varchar(25) NOT NULL,
```

```
'surname_to` varchar(45) NOT NULL,  
 `message_sent` text NOT NULL,  
 `date_send` datetime NOT NULL,  
 `Sender_pic` varchar(255) NOT NULL  
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

The email password recovery table

Please note this table is only usable if the lecture decide he want to use it and add a mail server to the code where the comments request it.

The code will generate a new password and a link for the user to be mailed to his computer with an expiring date of one day meaning after one day the password will be un-useable

localhost / localhost / register / [P](#) X +

localhost/phpMyAdmin/tbl_structure.php?db=register&table=password_reset_temp

Apps New Tab Dashboard Bootstrap Tutorial f... Lourens Van Der Bij... JavaScript Charts &... mySHEQ.com | Das... wadwafda Leona Lewis - Bleed...

phpMyAdmin

Recent Favorites

New

- information_schema
- mysql
- performance_schema
- register
 - New
 - messaging
 - password_reset_temp
 - posts
 - users
- social
- sys
- test

Server: localhost:3306 » Database: register » Table: password_reset_temp

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	email	varchar(250)	latin1_swedish_ci	No	None		Change Drop More	
<input type="checkbox"/>	2	key	varchar(250)	latin1_swedish_ci	No	None		Change Drop More	
<input type="checkbox"/>	3	expDate	datetime		No	None		Change Drop More	

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after expDate Go

Indexes

No index defined!

Create an index on 1 columns Go

Partitions

No partitioning defined!

```
CREATE TABLE `password_reset_temp` (
  `email` varchar(250) NOT NULL,
  `key` varchar(250) NOT NULL,
  `expDate` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The posts table

The Post table is where I store all the information about post that the users made

The screenshot shows the phpMyAdmin interface for the 'posts' table in the 'register' database. The table has 9 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)	utf8_general_ci		No	None	AUTO_INCREMENT		Change Drop More
2	username	varchar(25)	utf8_general_ci		No	None			Change Drop More
3	body	text	utf8_general_ci		No				Change Drop More
4	added_by_name	varchar(60)	utf8_general_ci		No	None			Change Drop More
5	added_by_surname	varchar(60)	utf8_general_ci		No	None			Change Drop More
6	date_added	datetime			No	None			Change Drop More
7	user_closed	varchar(3)	utf8_general_ci		No	None			Change Drop More
8	deleted	varchar(3)	utf8_general_ci		No	None			Change Drop More
9	post_pic	varchar(255)	utf8_general_ci		No	None			Change Drop More

Below the table structure, there are buttons for 'Check all', 'With selected:', and various actions like 'Browse', 'Change', 'Drop', etc. There is also a 'Indexes' section at the bottom.

```
CREATE TABLE `posts` (
  `id` int(11) NOT NULL,
```

```
'username` varchar(25) NOT NULL,  
`body` text NOT NULL,  
`added_by_name` varchar(60) NOT NULL,  
`added_by_surname` varchar(60) NOT NULL,  
`date_added` datetime NOT NULL,  
`user_closed` varchar(3) NOT NULL,  
`deleted` varchar(3) NOT NULL,  
`post_pic` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

The users table

The screenshot shows the phpMyAdmin interface for the 'register' database. The left sidebar lists databases and tables, with 'users' selected. The main area displays the 'Structure' tab for the 'users' table, showing 13 columns: id, username, email, password, name, surname, age, gender, country, city, profile_pic, post, and trn_date. Below the table structure, the 'Indexes' section shows two indexes: 'PRIMARY' (id) and 'email'. Navigation buttons like 'Check all', 'With selected:', and 'Go' are visible at the bottom.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)	utf8_general_ci		No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(50)	utf8_general_ci		No	None			Change Drop More
3	email	varchar(50)	utf8_general_ci		No	None			Change Drop More
4	password	varchar(50)	utf8_general_ci		No	None			Change Drop More
5	name	varchar(25)	utf8_general_ci		No	None			Change Drop More
6	surname	varchar(30)	utf8_general_ci		No	None			Change Drop More
7	age	int(3)			No	None			Change Drop More
8	gender	varchar(6)	utf8_general_ci		No	None			Change Drop More
9	country	varchar(25)	utf8_general_ci		No	None			Change Drop More
10	city	varchar(25)	utf8_general_ci		No	None			Change Drop More
11	profile_pic	varchar(255)	utf8_general_ci		No	None			Change Drop More
12	post	int(25)			No	None			Change Drop More
13	trn_date	datetime			No	None			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	6	A	No	
Edit Drop	email	BTREE	Yes	No	email	6	A	No	

```
-- Table structure for table `users`
```

```
--  
CREATE TABLE `users` (  
  `id` int(11) NOT NULL,  
  `username` varchar(50) NOT NULL,  
  `email` varchar(50) NOT NULL,  
  `password` varchar(50) NOT NULL,  
  `name` varchar(25) NOT NULL,  
  `surname` varchar(30) NOT NULL,  
  `age` int(3) NOT NULL,  
  `gender` varchar(6) NOT NULL,  
  `country` varchar(25) NOT NULL,  
  `city` varchar(25) NOT NULL,  
  `profile_pic` varchar(255) NOT NULL,  
  `post` int(25) NOT NULL,  
  `trn_date` datetime NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Social media app CODE

index.php

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Social Degree Home</title>
        <link rel="stylesheet" href="css/style.css" /> <!--connected to the style.css styling sheet-->
    </head>

    <body class="wrapper"><!--class name that is being style on my style.css page-->

        <table class = center>
            <tr>
                <td colspan="2" class="welcome_note">
                    <h1>Social Degree</h1>
                    <p>Welcome to Social Degree.
                    This is a social network platform based on Facebook, LinkedIn.<br>
                    The idea of this platform is to motivate and aid students in their studies.<br>
                    Students will be able to share ideas and study material that is needed to perform<br> in one's career</p>
            >
                </td>
            </tr>

            <tr>
                <br>
            </tr>
        </table>
    </body>
</html>
```

```

<tr>
    <td class="index_login">Already have an account?<br>
        <button class="button_login"><a href="login.php">Log in</a> </button>
    </td>

    <td class="index_Regis">Create a new account?<br>
        <button class="button_Regis"><a href="registration.php">Register</a> </button>
    </td>

<!--This is my welcome page. This page contains the welcoming paragraph and
the description of what the social app is all about.
This page has two button with a href that connects it to the login and registration
page-->
</table>
</body>
</html>

```

db.php (database connection)

```

<?php

//The connection is to localhost username is root and password is root and connecting to register database
$con = mysqli_connect("localhost","root","root","register");
if (mysqli_connect_errno()){ //connecting to the database if fail is will echo the error
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    die(); // if the connection fail it will also die all the attempted connections
}

```

```

date_default_timezone_set('Africa/Harare');
$error=""; //setting the time zone to south africa time and creating an error variable if error happens
?>

<!--This page is intended as a class so that I do not have to write this code on every page I can just include it-->

```

registration.php

```

<?php
    session_start();

    //connecting this page to the database via a separate connection page db.php
    require('db.php');
?>

<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8">
        <title>Social Degree Registration</title>
        <link rel="stylesheet" href="css/style.css" /><!--connecting to the style.css stylesheet-->
    </head>

    <body class="wrapper"><!--this wrapper is being styled in the style.css stylesheet-->

    <?php // starting my php in the body to get the details in the post form

```

```
// If form submitted, insert values into the database.
if (isset($_REQUEST['username'])){

    $username = stripslashes($_REQUEST['username']); // removes backslashes
    $username = mysqli_real_escape_string($con,$username); //escapes special characters in a string
    // the username variable is connected to my con (connection) variable as declared in the db.php page

    $email = stripslashes($_REQUEST['email']); // removes backslashes
    $email = mysqli_real_escape_string($con,$email); //escapes special characters in a string
    // the email variable is connected to my con (connection) variable as declared in the db.php page

    $password = stripslashes($_REQUEST['password']); // removes backslashes
    $password = mysqli_real_escape_string($con,$password); //escapes special characters in a string
    // the password variable is connected to my con (connection) variable as declared in the db.php page

    $name = stripslashes($_REQUEST['name']); // removes backslashes
    $name = mysqli_real_escape_string($con,$name); //escapes special characters in a string
    // the name variable is connected to my con (connection) variable as declared in the db.php page

    $surname = stripslashes($_REQUEST['surname']); // removes backslashes
    $surname = mysqli_real_escape_string($con,$surname); //escapes special characters in a string
    // the surname variable is connected to my con (connection) variable as declared in the db.php page

    $age = stripslashes($_REQUEST['age']); // removes backslashes
    $age = mysqli_real_escape_string($con,$age); //escapes special characters in a string
    // the age variable is connected to my con (connection) variable as declared in the db.php page

    $gender = stripslashes($_REQUEST['gender']); // removes backslashes
    $gender = mysqli_real_escape_string($con,$gender); //escapes special characters in a string
    // the gender variable is connected to my con (connection) variable as declared in the db.php page
```

```

$country = stripslashes($_REQUEST['country']); // removes backslashes
$country = mysqli_real_escape_string($con,$country); //escapes special characters in a string
// the country variable is connected to my con (connection) variable as declared in the db.php page

$city = stripslashes($_REQUEST['city']); // removes backslashes
$city = mysqli_real_escape_string($con,$city); //escapes special characters in a string
// the city variable is connected to my con (connection) variable as declared in the db.php page

$trn_date = date("Y-m-d H:i:s"); //Automatic date function by php that get the current date and time

//This is my SQL query to add the values to the column in my users table note that ID is not included due to primary key and auto increasement..
//The md5 infront of password is to encrypt the password
$query = "INSERT into users (username, password, email, name, surname, age, gender, country, city, profile_pic, post, trn_date)
VALUES ('$username', '".md5($password)."', '$email', '$name', '$surname', '$age', '$gender', '$country', '$city', 'default.JPG','0', '$trn_date')";

// with result I combine the SQL query with the connection variable as defined in db.php
$result = mysqli_query($con,$query);
if($result){
    echo "<div class='form'><h3>You are registered successfully.</h3><br/>Click here to <a href='login.php'>Login</a></div>";
}
}else{
?>

<!-- This is the html registration form.
That is being used by the post method to obtain the information and insert it
</pre>

```

into the database.

By carefully declaring the input type field and marking it as required,
I also added some form validation features. -->

```
<div class="form">
    <h1>Registration</h1>
    <a href="login.php">Already a member click here!</a> <!--allowing the user to return to the login form-->

    <form name="registration" action="" method="post">
        <input type="text" name="username" placeholder="Username" required />
        <input type="email" name="email" placeholder="Email" required />
        <input type="password" name="password" placeholder="Password" required />
        <input type="text" name="name" placeholder="Name" required />
        <input type="text" name="surname" placeholder="Surname" required /> <br><br>
        <input type="number" name="age" placeholder="Age" required /> <br> <br>

        <select name="gender" id="gender" required>
            <option value="Defualt">Gender</option>
            <option value="Male">Male</option>
            <option value="Female">Female</option>
        </select> <br>

        <input type="text" name="country" placeholder="Country" required />
        <input type="text" name="city" placeholder="City" required/><br><br>
        <label>Add a profile picture on settings Tab</label>

        <input type="submit" name="submit" value="Register" />
    </form>
</div>

<?php } ?>
```

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

login.php

```
<?php  
//starting a session variable to keep the username to use it in the dashboard.php  
session_start();  
?>  
  
<!DOCTYPE html>  
<html>  
  
    <head>  
        <meta charset="utf-8">  
        <title>Login</title>  
        <link rel="stylesheet" href="css/style.css" /><!--connecting to the style.css stylesheet-->  
    </head>  
  
    <body class = "wrapper">  
  
        <?php //starting php commands in my body tag to obtain details about existing users  
  
            //connecting this page to the database that is on db.php (db = database)  
            require('db.php');
```

```

$_SESSION['username'] = $username; //creating the username session for my dashboard.php

// If form submitted, start the query to see if the user exist
if (isset($_POST['username'])){

    $username = stripslashes($_REQUEST['username']); // remove spaces
    $username = mysqli_real_escape_string($con,$username); //remove special characters in a string
    // combining the con variable on db.php with the username variable

    $password = stripslashes($_REQUEST['password']); // remove spaces
    $password = mysqli_real_escape_string($con,$password); //remove special characters in a string
    // combining the con variable on db.php with the password variable

    //Checking is user existing in the database or not
    $query = "SELECT * FROM `users` WHERE username='".$username' and password='".md5($password)."'";
    //with the md5 I am encrypting the password entered also to match it with the already encrypted one in the database

    //result is the SQL query with the connection variable (db.php) and the query above
    $result = mysqli_query($con,$query) or die(mysql_error());

    //using a SQL fuction that will count the rows where the query applies
    $rows = mysqli_num_rows($result);

    // if the rows return is 1 then the user exist and we will proceed to the dashboard.php
    if($rows==1){
        $_SESSION['username'] = $username; //recalling the session variable of the text in the username field, and re-declaring it
        header("Location: dashboard.php"); // Redirect user to dashboard.php
    }else{ //if the variable rows return 0 is means that the user does not exist
}
}

```

```

        echo "<div class='form'><h3>Username/password is incorrect.</h3><br/>Click here to <a href='login.php'>Login
</a></div>";
    }
}else{

?>
<div class="form">
    <h1>Social Degree</h1>
    <h1>Log In</h1>

    <form action="" method="post" name="login">
        <!--required field has been added to ensure that the user do not submit and empty request -->
        <input type="text" name="username" placeholder="Username" required />
        <input type="password" name="password" placeholder="Password" required />
        <input name="submit" type="submit" value="Log in" />
    </form>

    <p>Not registered yet? <a href='registration.php'>Register Here</a></p>
    <p>Forgot Password? <a href='passwordRec.php'>Recover Here</a></p>
    <!--Form design with the post method to hide any information in the case of attackers.
    With the options for users to register if they do not have an account and also if they had forgotten their p
asswords.

    On both a href had been design that will take you to the page to -->
</div>

<?php } ?>
</body>
</html>
```

passwordRec.php

(This code has been avoided with an HREF due to the fact that I do not have a mail server and the time consuming factor that lectures might encounter when testing the code to check their emails, if you wish to test the code simply remove the HREF on the button and add your mail server info into the code where the comments request it. The idea of this code is that the system construct an email to send to the user with a temporary password and a link to allow them to change their passwords).

```
<?php

?>
<!--With this code I create an automatic email reply once a user
click on thie forgot password.
But this code is being by-passed at the bottom of the code to allow easy access to lectures.
And because no email server set up. if you want it to work actively yo can just enter your email server details
in the code below --&gt;

&lt;html&gt;

    &lt;head&gt;
        &lt;title&gt;Password Recovery - Socail Degree&lt;/title&gt;
        &lt;link rel='stylesheet' href='css/style.css' type='text/css' media='all' /&gt;
    &lt;/head&gt;

    &lt;body&gt;
        &lt;div style="width:700px; margin:50 auto;"&gt;
            &lt;h2&gt;Social Degree Password Recovery&lt;/h2&gt;
    &lt;?php
    include('db.php'); //including the database connection file</pre>
```

```

//if the textbox has been filled make the following variables
if(isset($_POST["email"]) && (!empty($_POST["email"])))
{
    $email = $_POST["email"];
    $email = filter_var($email, FILTER_SANITIZE_EMAIL); //this will remove all illegal characters from an email address
    $email = filter_var($email, FILTER_VALIDATE_EMAIL); //this will remove all illegal characters from an email address

    if (!$email) { //if the email variable is incorrect the user made a type error
        $error .= "<p>Invalid email address please type a valid email address!</p>";
    }else{ //if no type errors we will start searching the database for the email typed by the user
        $sel_query = "SELECT * FROM `users` WHERE email='".$email."'";
        $results = mysqli_query($con,$sel_query);
        $row = mysqli_num_rows($results); //using a php row function to determine how many rows was found with the query result
        if ($row==""){ // if no rows was found it means that this user does not exist or email does not exist
            $error .= "<p>No user is registered with this email address!</p>";
        }
    }

    // creating another table to create a new temporary password for the user
    if($error!=""){
        echo "<div class='error'>".$error."</div>
        <br /><a href='javascript:history.go(-1)'>Go Back</a>";
    }else{
        // the created password will only be useble for 1 day
        $expFormat = mktime(date("H"), date("i"), date("s"), date("m") , date("d")+1, date("Y"));
        $expDate = date("Y-m-d H:i:s",$expFormat); //the date that the request has been made
        $key = md5(2418*2+$email); // creating an unique key and combining it to create an unique password
    }
}

```

```

$addKey = substr(md5(uniqid(rand(),1)),3,10);
$key = $key . $addKey;
// Insert into the password_reset_temp table
mysqli_query($con,
"INSERT INTO `password_reset_temp` (`email`, `key`, `expDate`)
VALUES ('".$email."', '".$key."', '".$expDate."');

//With this code we are creating the email that is going to be sent to the users email account
$output='<p>Dear user,</p>';
$output.=<p>Please click on the following link to reset your password.</p>;
$output.=<p>-----</p>;
$output.= '<p><a href="C:\MAMP\htdocs\SocialMedaiApp</a></p>'; //need to be change to a password recovery page
$output.=<p>-----</p>;
$output.= '<p>Please be sure to copy the entire link into your browser.
The link will expire after 1 day for security reason.</p>'; //with the time declaration this link or actaully the pa
ssword will only be available for 1 day
$output.= '<p>If you did not request this forgotten password email, no action
is needed, your password will not be reset. However, you may want to log into
your account and change your security password as someone may have guessed it.</p>';
$body = $output; // defining it as the body of the email
$subject = "Password Recovery -Social Degree"; // the subject line

$email_to = $email; //the email address that was found in the database
$fromserver = "noreply@noHost.com"; //enter a host server email
require("PHPMailer/PHPMailerAutoload.php");
$mail = new PHPMailer();
$mail->IsSMTP();
$mail->Host = "mail.visionweb.co.za"; // Enter your host here
$mail->SMTPAuth = true;
$mail->Username = "lourens@visionweb.co.za"; // Enter your email here
$mail->Password = "Middelburg"; //Enter your password here

```

```

$mail->Port = 25; //the port of your host server
$mail->IsHTML(true);
$mail->From = "noreply@noHost.com"; // changing the sender information
$mail->FromName = "Social Degree"; // changing the sender information
$mail->Sender = $fromserver; // indicates ReturnPath header
$mail->Subject = $subject; // creating the subject variable to enter into the email
$mail->Body = $body; // creating the email body variable
$mail->AddAddress($email_to); // ensuring that the email is sent
if(!$mail->Send()){
    echo "Mailer Error: " . $mail->ErrorInfo;
} else{
    echo "<div class='error'>
        <p>An email has been sent to you with instructions on how to reset your password.</p>
    </div><br /><br /><br />";
}
}

} else{
?>

<!--As you will see the code above enter all into the password_reset_temp table in the database and will
also sent the email perfectly. But due to the fact that I have no email server and I do not want to make marking my
assignment time consuming I have created a by-pass method to the reset.php page where user can change the email without
using the email instructions -->
<form method="post" action="" name="reset">

    <label><strong>Enter Your Email Address:</strong></label><br /><br />
    <input type="email" name="email" placeholder="username@email.com" />
    <br /><br />
    <button><a href="reset.php">Reset Password</a></button>

```

```
<!-- With the href above i will automatically move the user to the reset page by-passing the email sent -->
</form>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<?php } ?>

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

reset.php

The link that the email will sent or the HREF on the button will redirect the user to this page that allows password change

```
<?php

?>
<html>
    <head>
        <title>Reset Password - Social Degree</title>
        <link rel='stylesheet' href='css/style.css' type='text/css' media='all' />
    </head>

    <body>
        <?php
            require('db.php'); //including the connection to the database on db.php
            session_start();
```

```

// If form submitted, insert values into the database.
if (isset($_POST['email'])){

    $email = stripslashes($_REQUEST['email']); // remove spaces
    $email = mysqli_real_escape_string($con,$email); //remove special characters in a string
    $NewPassword = stripslashes($_REQUEST['NewPassword']); // remove spaces
    $NewPassword = mysqli_real_escape_string($con,$NewPassword); //remove special characters in a string

    //Checking is user existing in the database or not
    $query = "UPDATE users SET password='".md5($NewPassword)' WHERE email = $email";
    $result = mysqli_query($con,$query) or die(mysql_error()); //if the querry suceed .. proceed if not die the
connection
    $rows = mysqli_num_rows($result); //using the php row function to determine whether there is a record with thi
s email address

    //if row is equal to 1 it means that a record has been found and has been updated
    if($rows==1){
        $_SESSION['email'] = $email;
        header("Location: login.php"); // Redirect user to login.php
        //if the email is found and has been updated the user will be redirected to the login.php page

    }else{ // if no rows has been found then nothing will be updated cause the email does not exist
        echo "<div class='form'><h3>User Not found.</h3><br/>Click here to <a href='login.php'>Login</a></div>";
    }
}else{

    ?>
<div class="form">

```

```

<h1>Social degree</h1>
<h1>Reset Password</h1>

<form action="" method="post" name="login">
    <input type="text" name="email" placeholder="Account Email" required />
    <input type="password" name="NewPassword" placeholder="New Password" required />
    <input name="submit" type="submit" value="Change Password" />
</form>

    <p>Not registered yet? <a href='registration.php'>Register Here</a></p>
    <p>Back to Log in? <a href='login.php'>Log in Here</a></p>

</div>
<?php } ?>

</body>
</html>

```

user.php

Once the user has logged in I am creating a user class so that access to the information of the user that is logged in (saved with a session variable) can be easily obtain throughout the complete application this file is located in the includes folder

```

<!-- with this code I created a user class where I can pull the user information from
any of my pages where it is included on this page -->
<?php
class User { //the user class
    private $user; //private class variable user

```

```

private $con; //the con variable as declare in db.php

//This is the contructor that save all information on the user so that i can use it when I include the user file
public function __construct($con, $user){
    $this->con = $con;
    $query = "SELECT * FROM `users` WHERE username='$user'";
    $user_details_query = mysqli_query($con,$query) or die(mysql_error());
    $this->user = mysqli_fetch_array($user_details_query);

}

public function getUsername() { //part of the class I use this fuction to save the user log in instead of the session
    return $this->user['username'];
}

//This fuction is not part of the assignment but can be use to get the number of post that the user have post
public function getNumPosts() {
    $username = $this->user['username'];
    $query = "SELECT post FROM `users` WHERE username='$username'";
    $query1 = mysqli_query($this->con,$query);
    $row = mysqli_fetch_array($query1);
    return $row['post'];
}

public function getFirstAndLastName() { //easy accessible fucntion that return the name and the surname of the user
    $username = $this->user['username'];
    $query = "SELECT name, surname FROM `users` WHERE username='$username'";
    $query_getUserFL = mysqli_query($this->con,$query);
    $row = mysqli_fetch_array($query_getUserFL);
}

```

```
        return $row[ 'name' ] . " " . $row[ 'surname' ];
    }
}

?>
```

.htaccess

This is also not part of the assignment but is coded to allow easy access to user's details and information and also aid in the process of unique usernames (this is a type hidden file)

```
RewriteEngine On
RewriteRule ^([a-zA-Z0-9_-]+)$ profile.php?profile_username=$1
RewriteRule ^([a-zA-Z0-9_-]+)/$ profile.php?profile_username=$1
```

header.php

This code is use for the top navigation bar using the user logged in information to personalize it

```
<!--This is my header file in also form a sort of class meaning i can use this code on
multiple pages. on every page I will include the header that indicate the user loged in
and the available menu bar for the user-->

<?php
session_start(); //pulling the username for the session variable on log in to save it as the user loged in
require('db.php'); // using my connection page to connect to the database

if(isset($_SESSION[ 'username' ])){ //Sesion obtain from the login.php page
    $userLoggedIn = $_SESSION[ 'username' ]; // user that is not log in will be redirected

    //with this query I used the sesion variable to search for the users details in the database
```

```

$query = "SELECT * FROM `users` WHERE username='$userLoggedIn'";
$user_details = mysqli_query($con,$query) or die(mysql_error());
$user = mysqli_fetch_array($user_details, MYSQLI_ASSOC);

// if not found redirect the user to the login page (maybe when the session has cleared)
} else {
    header("Location: login.php");
}
// creating a search option to view other users or his own page
if (isset($_GET['search'])){
    header("Location: profile.php");

}
?>

<html>
<head>
    <title>Welcome to Social Degree</title>
    <link rel="stylesheet" href="css/hstyle.css"/>
</head>
<body>
    <div class = "top-nav-bar">
        <div class="logo">
            <a href = "dashboard.php">  </a>

        </div>

        <nav>
            <!--form for the search box allowing users to search for one another-->

```

```

<form action="profile.php" method="post" class = "topnav">
    <input type="text" name="search" placeholder="Search by name" >
    <input type= "submit" value = "Search user">

</form>

<!--The navigation bar menu's--&gt;
&lt;a href = "#" class = "user"&gt; &lt;?php echo $user[ 'name' ]." ". $user[ 'surname' ]; ?&gt; &lt;img src="images/user.png" alt="Logo" width = "40" height = "30" class="icons"&gt; &lt;/a&gt;
    &lt;a href = "messageSystem.php"&gt; &lt;img src="images/mesage.png" alt="mesenger" width = "40" height = "30" class="icons"&gt; &lt;/a&gt;
        &lt!-- will take yhe user to its inbox and outbox--&gt;

        &lt;a href = "#"&gt; &lt;img src="images/notification.png" alt="notification" width = "40" height = "30" class="icons"&gt; &lt;/a&gt;
            &lt!--This was not part of the assignment but give the pages more look--&gt;

            &lt;a href = "update.php"&gt; &lt;img src="images/maintenance.png" alt="setings" width = "40" height = "30" class="icons"&gt; &lt;/a&gt;
                &lt!-- will take the user to his setings page to update his details--&gt;

                &lt;a href = "logout.php"&gt; &lt;img src="images/on-off-button.png" alt="Logout" width = "40" height = "30" class="icons"&gt; &lt;/a&gt;
                    &lt!-- User Logout and also destroy the session--&gt;
&lt;/nav&gt;

&lt;!--
- please notice that the div class is starting on the header.php and ending on the dashboard.php same goes for the messaging pages
</pre>

```

This is to connect the header page with all the other pages-->

```
</div>
<div class="wrapper">
```

dashboard.php

This codes fall below the header.php file that will complete the dashboard or full view of posts for all the users

```
<?php

require('db.php'); //including my connection to the database page
include("header.php"); //including the header that will appear on the top of the page
include("include/classes/User.php"); //including the user.php to allow easy access for user information
?>

<!--
Take note that because this section falls below the header.php the styling of this page will also be done in hstyle.css
-->
<div class="user_detail_column">

    <!--creating the table that allows a user to write posts-->
    <table class="profile_table">
        <tr>
            <td class = "user-
column"><a href = "<? echo $userLoggedIn; ?>" class="user_pic"> </a><br>
```

```

<?php echo $user['name']." ".$user['surname'];?><br> <!--obtaining this information from the user class-->
Gender: <?php echo $user['gender'];?> <br> <!--obtaining this information from the user class-->
Age: <?php echo $user['age'];?> <br> <!--obtaining this information from the user class-->
<button><a href = "update.php" class = "changelink">Change</a></button>

</td>
<td class = "user-post"><br>
<!--
Using the post method to save the text that the user input in the text field to insert it into the database -->
<form class = "post_form" action = " " method="POST">
    <textarea name="post_text" id = "post_text" placeholder = "Want to post something??" class="post_textarea" required></textarea>
    <input type="submit" name="submit_post" value="posting" class="post_button">
</form>

<?php //starting the php code to input the post into the database
require('db.php'); //connecting to the database via db.php

//if the textbox for post is filled out(dont allow users to post nothing)
if(isset($_REQUEST['post_text'])){
    $userPost = $_POST['post_text'];
    $userPost = stripslashes($_REQUEST['post_text']); // remove backslashes
    $userPost = mysqli_real_escape_string($con,$userPost); //remove all special characters
    $datePost = date("Y-m-d H:i:s"); //php function to save the date and time of the post to display it from newest to oldest
    $added_by_name = $user['name']; //adding the user that made the post via the user class
    $added_by_surname = $user['surname']; //adding the user surname that made the post via the user classes
    $user_pic = $user['profile_pic']; // adding the profile pic of the user that made the post via the user class
}

```

```

        //inserting all the information regarding the post into the database under the post table
        $query = "INSERT into posts (username, body, added_by_name, added_by_surname, date_added, user_closed, deleted, post_pic)
                  VALUES ('$userLoggedIn','$userPost', '$added_by_name', '$added_by_surname', '$datePost', 'no', 'no',
'${user_pic}');
        $result = mysqli_query($con,$query);
        if($result){
            echo "<h3>Post added to your profile";
        }
        }else{
        }
    ?>
    </td>
</tr>
</table><br><br>
</div>

<!-- creating the table and div class where all the post of the entire database will be displayed--&gt;
&lt;div class = the_posts&gt;
    &lt;table class = "post_table"&gt;
&lt;?php
    //no need to include the database connection cause it has already been included above
    $sql = "SELECT * FROM posts ORDER BY date_added DESC;";
    $result = mysqli_query($con, $sql);
    $resultCheck = mysqli_num_rows($result);//using php row fuction to obtain the amount of row and their data in it

    //The row fuction basically makes a class of each row of the data obtain
    if($resultCheck &gt; 0) { //if the rows are more than 1 their is post that can be recalled if 0 there is no posts in the database
        while ($row = mysqli_fetch_assoc($result)) {
</pre>

```

```

        echo "<tr class='post_row'>
            <td class = 'post_pic' >
                <img src='uploads/".$row['post_pic']."' height=150 width=150 /></td>
            <td class= 'post_user'>
                <h4>".$row['added_by_name']." ".$row['added_by_surname']."</h4>".$row['date_added']."
            <td class='post_text'>
                <h3>".$row['body']."</h3></td></tr>";

        // because I want to print all the post I will use a while statement, while there still is some post kee
p printing
        //while row is more than 0 print each line in its own row in this table
        //and using the small post class that I created with the php row fucntion
    }

}

?>
    </table>
</div>
</div> <!--closing the div that started in my header.php file-->

</body>
</html>

```

profile.php

When a user search for another in the search box located in the header the following code is being executed

```

<!--When a user search for another user we will be displaying all the post made by that
user. In the header.php file we offer a search that can be used to be redirected to this page
where the user being search information and posts will be displayed.
also on this page we will allow a user to search for another user and sent him / her an online
message-->
<?php
    session_start(); //start the sesion of the user loged in we wil be using it to sent messages
    require('db.php'); //including the database connection page
    include("header.php"); //including the header so we can see to person currently loged in
    include("include/classes/User.php"); //including the user class (user.php) to use the information of the user loged
    in
    $userProfile = $_POST['search']; //obtaining the information that was entered in the search box
?>

<!-- The table for the details of the user being search-->
<div class="user_detail_column">
<body>

<div class = "viewProfile">
    <table class = "beingView" >
        <?php
            //searching the database for the users being search for
            $sql = "SELECT * FROM users WHERE name= '$userProfile'";
            $result = mysqli_query($con, $sql);
            //obtaining all that information with the php row function (we will be using it as a class also)
            $resultCheck = mysqli_num_rows($result);

            if($resultCheck > 0) {
                while ($row = mysqli_fetch_assoc($result)) {
                    //while row is equal to data from the query print the following table rows and columns

```

```

//we are using the class of the php row function to obtain the data in the rows $row['the column we are looking for']

echo "<h1 class='private_profile_heading'>This is the profile page of:</h1>
<tr>
    <td class='private_profile_IMG'>
        <img src='uploads/".$row['profile_pic']."' height=150 width=150 /></td>
    <td class='private_profile_details'>
        <h2>".$row['name']." ".$row['surname']. "</h2>
        Gender: <b>".$row['gender']. "</b><br>
        Age: <b>".$row['age']. "</b><br>
        Country: <b>".$row['country']. "</b><br>
        City: <b>".$row['city']. "</b><br>
        User since: <b>".$row['trn_date']. "</b></td>

<td class='private_profile.messaging'>
<form action='messageSystem.php' method='post' class = 'messenger'>
    <textarea name='messageSent' placeholder = 'Type message to ".$row['name']."' class = 'messageSent'>
</textarea>
    <input type = 'hidden' name='sent_to_name' value ='".$row['name']."'>
    <input type = 'hidden' name='sent_to_surname' value ='".$row['surname']."'><br>
    <input type= 'submit' value = 'Sent Message'>
</form></td>";

//with the form above I added the feature to do online messages where a user can search for a profile and sent them a message
//This form is of post method where the information will be sent to the messageSystem.php page where all the messages will be displayed
}
}

```

```

    ?>
</table>
</div>

<!--
once the user have search for another users I will be querying al the post from the user that is being searched and disp
lay it on this page--&gt;
&lt;div class = "viewProfile_Post"&gt;
&lt;table class="beingView_Post"&gt;

&lt;?php
    //Search for all the post from the table post where the user being search is the user that added the post
    $sql = "SELECT * FROM posts WHERE added_by_name = '$userProfile'";
    $result = mysqli_query($con, $sql);
    $resultCheck = mysqli_num_rows($result);
    //The results must be pulled with the php row fuction so that the data can be displayed in a class manner

    if($resultCheck &gt; 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            echo "&lt;tr class='post_row'&gt;
                &lt;td class = 'post_pic' &gt;
                    &lt;img src='uploads/".$row['post_pic']."' height=150 width=150 /&gt;&lt;/td&gt;
                &lt;td class= 'post_user'&gt;
                    &lt;h4&gt;".$row['added_by_name']. " ".$row['added_by_surname']."&lt;/h4&gt;".$row['date_added']. "&lt;/td&gt;
                &lt;td class='post_text'&gt;
                    &lt;h3&gt;".$row['body']. "&lt;/h3&gt;&lt;/td&gt;&lt;/tr&gt;";
        }
    }
}
</pre>

```

```
?>
</div>
</div>
</div>

</body>
</html>
```

messageSystem.php

```
<!--when a user sent a message or click on the message icon he should be redirected to this page  
on this page I will be displaying the inbox and the outbox of the user the message he has sent and  
the message he has received-->  
<?php  
    session_start(); //sesion to know what user is currently logged in  
    require('db.php'); //connecting this page to the database via db.php  
    include("header.php"); // adding the header file  
    include("include/classes/User.php"); //including the user.php file to use all the information about the user loged in  
  
    if(isset($_REQUEST['messageSent'])){ //before I add a message to the database insure that there is a message to  
add  
        $userMessage = $_POST['messageSent']; //via the post method obtain the data in the text area  
        $userMessage = stripslashes($_REQUEST['messageSent']); // remove backslashes  
        $userMessage = mysqli_real_escape_string($con,$userMessage); //remove all special characters  
        $dateSent = date("Y-m-  
d H:i:s"); //declare a variable for the current date and time to know when the message was sent  
        $name_from = $user['name']; //variable for the sender of the message using the user.php class  
        $surname_from = $user['surname']; //variable for the sender of the message using the user.php class  
        $name_to = $_POST['sent_to_name']; //declaring a variable for the receiver of the message using the post met  
hod in the search box
```

```

    $surname_to = $_POST['sent_to_surname']; //declaring a variable for the receiver of the message using the post method in the search box
    $sender_pic = $user['profile_pic']; //adding the senders profile pic

    //the insert query to insert the message into the database
    $query = "INSERT into messaging (name_from, surname_from, name_to, surname_to, message_sent, date_send, Sender_pic)
VALUES ('$name_from','$surname_from', '$name_to', '$surname_to', '$userMessage', '$dateSent', '$sender_pic')
";
    $result = mysqli_query($con,$query);
    if($result){ // if sent was successfull add a message to inform the user
        echo "<h3>Message sent</h3>";

    }
}else{
}
?>

<div class="user_detail_column">

<body>
    <div class = "split left"> <!--method use to split the screen as styled on hstyle.css-->

    <table class = 'messaging_system'><!--Table to display the messages-->
        <tr>
            <td class='inbox_heading' colspan = "3"><h2>Inbox</h2></td><!--
creating a heading that span across 3 columns-->
        </tr>
        <?php
            $userLoggedIn = $user['name']; //variable for who is currently loged in

```

```

// SQL query to retrieve the message where the message belongs to the user that is loged in
$sql = "SELECT * FROM messaging WHERE name_to = '$userLoggedIn' ORDER BY date_send DESC;";
$result = mysqli_query($con, $sql); //connecting the connection to the query
$resultCheck = mysqli_num_rows($result);//displaying it with the php row function

if($resultCheck > 0) {
    while ($rows = mysqli_fetch_assoc($result)) {
        echo "<tr>
            <td class= 'inbox_details'>
                From: ".$rows['name_from']." ".$rows['surname_from']."<br>
                To: ".$rows['name_to']." ".$rows['surname_to']."<br>
                Date: ".$rows['date_send']."
            <td class='inbox_message'>
                ".$rows['message_sent']."<br></td>
            <td class='inbox_pics'>
                <img src='uploads/".$rows['Sender_pic']."' height=80 width=80 /> </td>";
    }
}
?>
</table>
</div>

<!--split screen method for the outbox side of the screen--&gt;
&lt;div class = "split right"&gt;
    &lt;table class = 'messaging_system'&gt;
        &lt;tr&gt;
            &lt;td class='inbox_heading' colspan = "3"&gt;&lt;h2&gt;Outbox&lt;/h2&gt;&lt;/td&gt; &lt;!--
creating a heading that span across 3 columns--&gt;
        &lt;/tr&gt;
        &lt;?php
</pre>

```

```

$userLoggedIn = $user['name']; //variable for who is currently logged in

// SQL query to retrieve the message where the message belongs to the user that is logged in
$sql = "SELECT * FROM messaging WHERE name_from = '$userLoggedIn' ORDER BY date_send DESC;";
$result = mysqli_query($con, $sql); //connecting the connection to the query
$resultCheck = mysqli_num_rows($result); //displaying it with the php row function

if($resultCheck > 0) {
    while ($rows = mysqli_fetch_assoc($result)) {
        echo "<tr>
            <td class= 'outbox_details'>
                To: ".$rows['name_to']." ".$rows['surname_to']."<br>
                From: ".$rows['name_from']." ".$rows['surname_from']."<br>
                Date: ".$rows['date_send']."
            <td class='outbox_message'>
                ".$rows['message_sent']."
            <td class='outbox_pics'>
                <img src='images/mail.png' height=80 width=80 /> </td>";
    }
}
?>

</div>
</div>
</div>

</body>
</html>

```

update.php

```
<?php
    session_start();
    include("header.php"); //including the header that will appear on the top of the page
    include("include/classes/User.php");

    //connecting this page to the database via a seperate connection page db.php
    require('db.php');

?>

<body ><!--this wrapper is being styled in the style.css stylesheet-->
<div class = "form-update">
<h2>User Details update</h2>

<Label>Change Password: </Label> <a href = "reset.php">Click here </a>
<br><br> <!--Redirect the user to the password recover page that allows a user change his/her password-->

<div class = "personal_info">
<form method="post" action="update.php" name="UpdateName">
<br>
<Label>Change Name: </Label> <input type="text" name="UName" value = "<?php echo $user['name']; ?>"> <br><br>
<Label>Change Surname: </Label> <input type="text" name="USurname" value = "<?php echo $user['surname']; ?>"><br><br>
<Label>Change Age: </Label> <input type="number" name="UAge" value = "<?php echo $user['age']; ?>"><br><br>
<Label>Change Gender: </Label> <input type="text" name="UGender" value = "<?php echo $user['gender']; ?>"><br><br>
<Label>Change Country: </Label><input type="text" name="UCountry" value = "<?php echo $user['country']; ?>"><br><br>
<Label>Change City: </Label> <input type="text" name="UCity" value = "<?php echo $user['city']; ?>"><br><br>
```

```

<br><input type ="submit" name="UpSurname" value = "Update" class="updateInfo" >
<h6>Please refresh to see updates</h6>

<!--<input type ="submit" name="UpSurname" value = "Update" class="updateInfo">-->
</form><br>
<?php
require ('db.php'); //connecting to the database with the db.php page
$updateName = $_POST['UName']; //declaring a variable from the POST method
$updateSurname = $_POST['USurname']; //declaring a variable from the POST method
$updateAge = $_POST['UAge']; //declaring a variable from the POST method
$updateGender = $_POST['UGender']; //declaring a variable from the POST method
$updateCountry = $_POST['UCountry']; //declaring a variable from the POST method
$updateCity = $_POST['UCity']; //declaring a variable from the POST method

//my SQL statement to update the database
$sqlName = "UPDATE users SET name='".$updateName."', surname = '".$updateSurname."' , age = '".$updateAge."', gender = '".$updateGender."', country = '".$updateCountry."', city = '".$updateCity."' WHERE username='$userLoggedIn'";
//because I used value instead of placeholders the statement will run even if the user only update one of the fields

//connecting the database con variable on the db.php page to the sql query above
if ($con->query($sqlName) === TRUE) {
    echo "";
    $con->close();
} else {
    echo "" . $conn->error;
}

```

```

    $con->close(); //closing the connection to allow another update
}
?>
</div>

<form method="post" action="upload.php" name="UpdateProfilePicture" enctype="multipart/form-data">
<br> <!--creating a form to upload image enctype ="multipart/form-data" that allows any form of file in this form -->
<Label>Change Profile Picture: </Label> <input type="file" name="fileToUpload" id="fileToUpload"> <input type ="submit"
name="UpProfilePic" value = "Update" class="updateInfo">
<!--hidden value to save to who the profile pic belongs to -->
<input type = "hidden" name= "uploadingUser" value = "<?php echo $user['username']; ?>">
</form><br>

</div>
</body>

</html>

```

upload.php

This page is used to upload profile pictures to the database and to the file uploads (folder in my project)

```

<?php
session_start();
include("include/classes/User.php"); //including the user.php file to use all the information about the user loged in

```

```
$target_dir = "uploads/"; //providing the path where the picture needs to be save
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]); //creating a new picture file with the above val
ues
$uploadOk = 1; // if the upload is success this should remain 1 if fail go to 0
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

// Validating the image file
if(isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if($check !== false) {
        echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        echo "File is not an image. <br><br> <a href = 'dashboard.php'> Back to Site </a>";
        $uploadOk = 0;
    }
}

// Check if file already exists
if (file_exists($target_file)) {
    echo "Sorry, file already exists. <br><br> <a href = 'dashboard.php'> Back to Site </a>";
    $uploadOk = 0;
}

// Check file size
if ($_FILES["fileToUpload"]["size"] > 500000) {
    echo "Sorry, your file is too large. <br><br> <a href = 'dashboard.php'> Back to Site </a>";
    $uploadOk = 0;
}

// Allow certain file formats
```

```

if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
    echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed. <br><br> <a href = 'dashboard.php'> Back to Site </a>";
    $uploadOk = 0;
}

// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
// if everything is ok, try to upload file
} else {
    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
        echo "The file ". htmlspecialchars( basename( $_FILES["fileToUpload"]["name"])). " has been uploaded. ";
        $servername = "localhost";
        $username = "root";
        $password = "root";
        $dbname = "register";
        $profile_pic = htmlspecialchars( basename( $_FILES["fileToUpload"]["name"]));
        $uploadingUser = $_POST['uploadingUser'];
//This code was used to demostrate my ability to write the code with out the use of the db.php file (inheritance)

        // Create connection
        $conn = new mysqli($servername, $username, $password, $dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }
        // The update query where the picture name is being saves
        $sql = "UPDATE users SET profile_pic='".$profile_pic."' WHERE username='$uploadingUser'";
    }
}

```

```
if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully <br><br> <a href = 'dashboard.php'> Back to Site </a>";

} else {
    echo "Error updating record: " . $conn->error;
}

$conn->close();

} else {
    echo "Sorry, there was an error uploading your file.";
}
?>
```

logout.php

This code is destroying all the session variables for security purposes

```
<?php

session_start();
if(session_destroy()) // Destroying All Sessions
{
header("Location: login.php"); // Redirecting To Login page
}
?>
```

style.css

With this style sheet I am styling the index, Login and Registration page

```
body {  
    font-family: Arial, Sans-Serif;  
}  
  
.clearfix:before,  
.clearfix:after {  
    content: "";  
    display: table;  
}  
  
.clearfix:after {  
    clear: both;  
}  
  
a {  
    color: #0067ab;  
    text-decoration: none;  
}  
  
a:hover {  
    text-decoration: underline;  
}  
  
.form {
```

The longer numbers allow
me to also set the
transparency

```
width: 300px;
margin: 0 auto;
background-color: #027bcc44;
text-align: center;
margin-top: 100px;
}

input[type='text'],
input[type='email'],
input[type='password'],
input[type='number'],
input[type='file'] {
    width: 200px;
    border-radius: 2px;
    border: 1px solid #CCC;
    padding: 10px;
    color: #333;
    font-size: 14px;
    margin-top: 10px;
}

input[type='submit'] {
    padding: 10px 25px 8px;
    color: #fff;
    background-color: #0067ab;
    text-shadow: rgba(0, 0, 0, 0.24) 0 1px 0;
    font-size: 16px;
    box-shadow: rgba(255, 255, 255, 0.24) 0 2px 0 0 inset, #fff 0 1px 0 0;
    border: 1px solid #0164a5;
    border-radius: 2px;
    margin-top: 10px;
}
```

```
        cursor: pointer;
    }

input[type='submit']:hover {
    background-color: #024978;
}

.wrapper {
    background-image: url("../images/background_log_reg.jpg");
    background-size: 100%;
    width: 100%;
    height: 100%;
    min-width: 950px;
    margin: 0 auto;
}

.center {
    margin-left: auto;
    margin-right: auto;
}

.welcome_note {
    text-align: center;
    border: 0px solid rgb(19, 19, 92);
    position: relative;
    width: 100%;
    margin: 0 auto;
    background-color: #027bcc70;
    text-align: center;
    margin-top: 100px;
    font-weight: bolder;
}
```

```
    font-size: 22px;
    color: rgb(0, 0, 0);
}

.index_login {
    background-color: #67f32779;
    text-align: center;
}

.index_Regis {
    background-color: #27f3d879;
    text-align: center;
}

.button_login {
    width: 100px;
    height: 50px;
    font-weight: bolder;
    font-size: 23px;
    background-color: rgba(4, 175, 4, 0.603);
    border: 3px solid rgb(1, 75, 1);
}

.button_login:hover {
    background-color: rgb(1, 75, 1);
    border: 3px solid rgba(4, 175, 4, 0.603);
}

.button_Regis {
    width: 150px;
    height: 50px;
```

```
font-weight: bolder;
font-size: 23px;
background-color: rgba(4, 135, 175, 0.603);
border: 3px solid rgb(0, 2, 128);
}

.button_Regis:hover {
    background-color: rgb(0, 2, 128);
    border: 3px solid rgba(4, 135, 175, 0.603);
}
```

hstyle.css

This styling sheet is used to style the rest of my social media application

```
body {
    background-color: #b3b3b3;
}

.top-nav-bar {
    width: 100%;
    height: 45px;
    background-color: #20aae5;
    border-bottom: 0.25em solid #1894ca;
    border-top: 0.25em solid #1894ca;
    margin-top: -8px;
    margin-left: -8px;
    position: fixed;
    display: inline-flex;
```

```
    z-index: 10;
}

.logo {
    background-color: #20aae5;
}

nav {
    float: right;
    right: 1px;
    position: absolute;
    margin: 10px;
}

.icons:hover {
    background-color: #011bad7c;
    color: #011bad7c;
}

nav a {
    position: relative;
    color: #20aae5;
    text-decoration: none;
    font-size: 17px;
}

.user {
    color: black;
    padding-right: 30px;
    font-size: 15px;
    position: relative;
```

```
    text-align: center;
    font-weight: bolder;
}

.user_pic {
    padding-top: 100px;
}

.wrapper {
    margin: 0 auto;
    top: 50px;
    position: relative;
    padding: 0 10px;
    width: 75%;
    min-width: 950px;
}

.user-column {
    border: 5px solid black;
    margin: 2px;
    padding: 15px;
    background-color: rgb(255, 255, 255);
    font-size: 18px;
    font-weight: bold;
}

.user-post {
    float: right;
    border: 2px solid black;
    box-shadow: 2px rgb(66, 66, 66);
    height: 260px;
```

```
padding: 15px;
width: 680px;
z-index: -1;
margin-left: 25px;
background-color: white;
}

.post_tarea {
    width: 540px;
    height: 200px;
    border-radius: 5px;
    resize: none;
    border: 1px solid #20aae5;
    font-size: 18px;
}

.post_tarea:focus {
    outline-color: #0300b4;
    background-color: #04adf5a4;
}

.post_button {
    width: 100px;
    height: 80px;
    position: absolute;
    margin-left: 15px;
    font-size: 22px;
    font-weight: bolder;
    border: 3px solid blue;
    background-color: #20aae5;
    color: #0300b4;
}
```

```
}

.the_posts {
    text-align: center;
}

.post_pic {
    width: 10%;
    border-top: 5px solid #b3b3b3;
    border-bottom: 5px solid #b3b3b3;
    background-color: #20aae5;
}

.post_user {
    width: -10px;
    border-top: 5px solid #b3b3b3;
    border-bottom: 5px solid #b3b3b3;
    background-color: #24b8f8;
    width: 180px;
    text-align: center;
    font-family: 'Courier New', Courier, monospace;
    color: #100e7c;
}

.post_text {
    text-align: center;
    font-family: 'Times New Roman', Times, serif;
    background-color: #20aae5;
    border-top: 5px solid #b3b3b3;
    border-bottom: 5px solid #b3b3b3;
}
```

```
.topnav {  
    float: left;  
    padding: 6px 10px;  
    margin-top: 0px;  
    margin-right: 106px;  
    background: #20aae5;  
    font-size: 17px;  
    border: none;  
    cursor: pointer;  
}  
  
.beingView {  
    background-color: #1894ca;  
    width: 100%;  
}  
  
.private_profile_heading {  
    text-align: center;  
    color: #0300b4;  
}  
  
.private_profile_details {  
    color: #061b94;  
    width: 400px;  
}  
  
.private_profile_mBox {  
    width: 100%;  
    height: 120px;  
    background-color: #62e0ffa9;
```

```
}

.inbox_heading {
    font-size: 24px;
    text-align: center;
}

.inbox_details {
    color: #0300b4;
    font-size: 12px;
    width: 200px;
    border-top: 2px solid rgb(0, 73, 0);
    background-color: rgba(89, 241, 89, 0.589);
}

.inbox_message {
    border-top: 2px solid rgb(0, 73, 0);
    background-color: rgba(62, 247, 62, 0.589);
    border-left: rgba(62, 247, 62, 0.589);
}

.inbox_pics {
    border-top: 2px solid rgb(0, 73, 0);
    border-right: 8px solid rgb(0, 73, 0);
}

.split {
    width: 50%;
    position: absolute;
    z-index: 1;
    top: 0;
```

```
    overflow-x: hidden;
    padding-top: 20px;
}

.left {
    left: 0;
    background-color: rgba(62, 247, 62, 0.589);
}

.right {
    right: 0;
    background-color: rgb(4, 143, 207);
}

.outbox_details {
    color: #0300b4;
    font-size: 12px;
    width: 200px;
    border-top: 2px solid rgb(3, 74, 141);
    background-color: rgba(6, 226, 197, 0.589);
    border-left: 8px solid rgb(3, 74, 141);
}

.outbox_message {
    border-top: 2px solid rgb(3, 74, 141);
    background-color: rgba(6, 226, 197, 0.589);
    border-left: rgb(3, 74, 141);
}

.outbox_pics {
    border-top: 2px solid rgb(3, 74, 141);
```

```
}

.messageSent {
    width: 350px;
    height: 150px;
    background-color: #01ecfd;
    resize: none;
}

.update_user {
    text-align: center;
    background-color: #00b2be83;
    position: fixed;
    width: 450;
    margin-top: 25px;
}

.form-update {
    text-align: center;
    width: 550px;
    background-color: #0300b44f;
    margin-left: 350px;
}

.updateInfo {
    margin-left: 30px;
}

.personal_info {
    background-color: #01ecfd;
}
```

```
.changelink {  
    font-weight: 400;  
    color: rgb(37, 0, 250);  
}
```

QUESTIONS TWO

[20 MARKS]

What is the difference between cookies and session? In what circumstance should someone use cookies or session? Support your answer with practical Examples.

Cookies and session are used to store information a cookie is a piece of information stored on the clients URL where a session is information that is stored on the server side. A session variable in some cases sometimes uses a cookie variable to obtain the information on the server

Cookies	Sessions
Data is stored in the users browser	Data is stored on the Server
Cookies can be viewed by users or attackers (high risk)	Sessions cannot be viewed by the user or an attacker
Can only store string values	Can store object values as well

Cookies data is available until it reaches its expiring date	Session data is only available while the session is live (while the browser is open)
--	--

Cookies can be used to store information that is not sensitive that cannot cause harm if an attacker obtain it or if it is changed. We can use cookies to identify users save the personal choices when it comes to web site appearance etc.

For Example

```
<!DOCTYPE html>
<?php
$cookie_name = "user"; //creating the cookie variable named "user"
$cookie_value = "John Doe"; //Giving the cookie a value
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
// Setting the cookie and declaring its expiring date 86400 = 1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) { //checking that the cookie is set
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
</body>
</html>
```

With this code I have created a cookie to be stored on the users URL (in this case the user is John Doe) so if the user re-log on this page I pull the cookie variable to obtain the information stored in it

Session can be used to store bit more sensitive data that the developer does not want people to change or obtain for instance a log in system. We do not want a user to simply change the information in this variable and be logged in as somebody else. Sessions can be created at the start of the web page and be reused throughout the complete website

For example

Let's state the we are selling computers and cars all to many we get client request asking which is the best car or the best computer to have we decided to add that to our web site so that the clients can see for them self the best car/computer of this time.

So on the first page we first need to declare these variables

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["best_car"] = "Aston Martin"; //providng fixed default values for the sake of this example
$_SESSION["best_computer"] = "Dell"; //providng fixed default values for the sake of this example
echo "Session variables are set."; //inform the user that the sessoin is set
?>
```

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

Now that we have set these session on another page of the website we want to show these values to the users that wish to view it we can simply pull these session variables

```
<?php
session_start(); //new page so we start the session again
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "The best car is " . $_SESSION["best_car"] . ".<br>";
echo "The best computer is " . $_SESSION["best_computer"] . ".";
?>

</body>
</html>
```

On some sort of logging system or social app it is important to know which user is currently logged in even when they browse through your website so we save the current logged in user in a session variable and recall the variable all over our website if the session is destroyed don't let the user keep browsing

References

<https://www.youtube.com/watch?v=JrTQWgj5b0g>

<https://www.w3schools.com/php/default.asp>

https://www.tutorialspoint.com/php/php_sessions.htm

<https://stackoverflow.com/questions/tagged/php>

Internet Programming 622 learners guide

TOTAL: 100 MARKS