CMS Presentation

Marco Ghibaudi

Student Number 94238

The Website

This is the project for an electronic music label, the main objective is to gather all the information regarding the events of the party.

Admin will be able to add posts and manage other accounts, while regular users, upon a registration through the website, will be allowed to comment the admin's post.

In the event post the admins will have to fill all the details relevant to the specific party.

FRONT END

General Design

The general style of the website will remain quite simple and minimal.

The focus will stay in the content

The design will stay in black and white and the main colours will be brought by high quality images of event's flyers
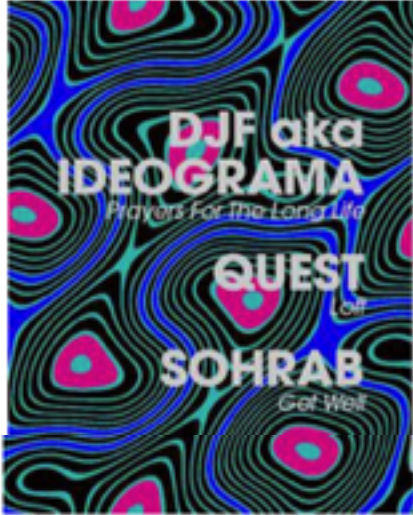
Home Page

The Home page will present a showcase of all the events submitted by the admin, from the last added.

The space will be organised in three columns, with the event's image, all the details

Event page

In the event page, the user will see just the event selected.

Admins can see the delete button, to delete the post and all his comments.

Logged in users, if not banned, have the possibility to leave a comment
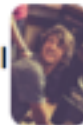
Edit user

Admins have access to profile's page where they have the options to ban/unban, mod/unmod or delete users

Profile page

Clicking on the profile_pic on the top right corner, the user can go on his personal profile page where he can change some of his details

BACK END

Website built with :

- HTML

- CSS

- JavaScript

- PHP

- MySQLi

Database

login_system_undersound account
- user_id : int(10)
- username : varchar(100)
- profilepic : varchar(50)
- status : varchar(10)
- banned : varchar(3)

login_system_undersound user
- user_id : int(10)
- firstname : varchar(100)
- lastname : varchar(100)
- password : varchar(200)
- email : varchar(100)
- activation : varchar(40)

login_system_undersound post
- post_id : int(10)
- user_id : int(10)
- title : varchar(100)
- postpic : varchar(100)
- content : varchar(10000)
- data : varchar(10)
- venue : varchar(100)
- lineup : varchar(100)

login_system_undersound comment
- comment_id : int(10)
- post_id : int(10)
- user_id : int(10)
- comment : text
- date : timestamp

user

-user_id (primary key)-> Auto-generated, scaling

Main index of the website, used in most of the sql to track other columns.

-first name, last name, email ->

Input by user through a registration form

-password -> Input by user through a registration form, inserted into DB with a sha1()

-activation -> auto-generated, grants the login only is in the database is set to NULL.

activation

Upon registration, an email is sent to the user with this link:

Click Here

$email = Set by the user in the registration form

$activation = Random generated

When the user visits email_activation.php a $_GET variable takes both query_string

With a sql it checks if at that specific email is assigned that specific activation key, if so, activation is set to NULL.

account

user_id -> derivate from 'user'

username, profilepic -> set by the user with forms

status -> set NULL by default

banned -> set NO by default

profilpic

The file is uploaded with a form and moved to the assets folder:

$move_file = move_uploaded_file($file_tmp_location, $file_path);

$file_path = assets/$file_name

$file_tmp_location = assets folder

The sql saves in the database the path to get to it:

$sql = "UPDATE account set profilepic = '$file_path' where user_id = '$user_id'";

$file_path = set with a $_FILES var, inserted by user with a form, input type='file'

$user_id = set by $_SESSION['user_id']

status

It can be set to NULL or admin, only admin can change this row in 'new_admin.php'

if($status != 'admin'){

$sql= "UPDATE account set status = 'admin' where user_id = '$user_id'";}

if($status = 'admin'){

$sql= "UPDATE account set status = '' where user_id = '$user_id'"}

$user_id = from a $_GET var

An user with banned=YES cannot be admin

if($banned == 'yes'){

    echo " A banned user cannot be admin , GO BACK";

banned

It can be set to YES or NO, only admin can change this row in 'ban_user.php'

if($banned = 'NO'){

$sql= "UPDATE account set banned = 'YES' where user_id = '$user_id'";}

if($banned = 'YES'){

$sql= "UPDATE account set banned = 'NO' where user_id = '$user_id'"}

$user_id = from a $_GET var

An user with status=admin cannot be banned

if($status == 'admin'){

    echo " An admin cant be banned , GO BACK";

admin_check.php

ban_user.php and new_admin.php can be accessed only after the admin_check.php

```
if(!$_SESSION){
    header('location:index.php');
    }else{
        if(!$_SESSION['status']){
            header('location:index.php');
    }
}
```

If the user is not logged in or not an admin, it heads to index.php

post

post_id-> Auto-generated, scaling

user_id -> derivate from 'user'

title, content, data, venue, lineup, postpic
-> set by the user with forms

post pic * same mechanics of profilepic

login_system_undersound **post**
- post_id : int(10)
- # user_id : int(10)
- title : varchar(100)
- postpic : varchar(100)
- content : varchar(10000)
- data : varchar(10)
- venue : varchar(100)
- lineup : varchar(100)

comment

comment_id-> Auto-generated, scaling

post_id-> derivate from 'post'

user_id -> derivate from 'user'

comment-> set by the user with forms

date-> Auto-generated

delete_process.php

Admin and Authors can delete post or comments

Next to posts and comments there is an image with anchor in it, the link sends to the same post page but with a query string, different for post or comment

<img>

<img>

The post page then recalls the PHP delete_process.php:

If isset($_GET['delete_post']) - > The post and all his comments are deleted

If isset($_GET['delete_comment']) - > The comment is deleted

```php
$delete_post
    $sql =  "DELETE FROM post WHERE post_id='$delete_post'";
        $result = mysqli_query($conn,$sql);
     $sql =  "DELETE FROM comment WHERE post_id='$delete_post'";
        $result = mysqli_query($conn,$sql);




$delete_comment
    $sql =  "DELETE FROM comment WHERE comment_id='$delete_comment'";
        $result = mysqli_query($conn,$sql);
```
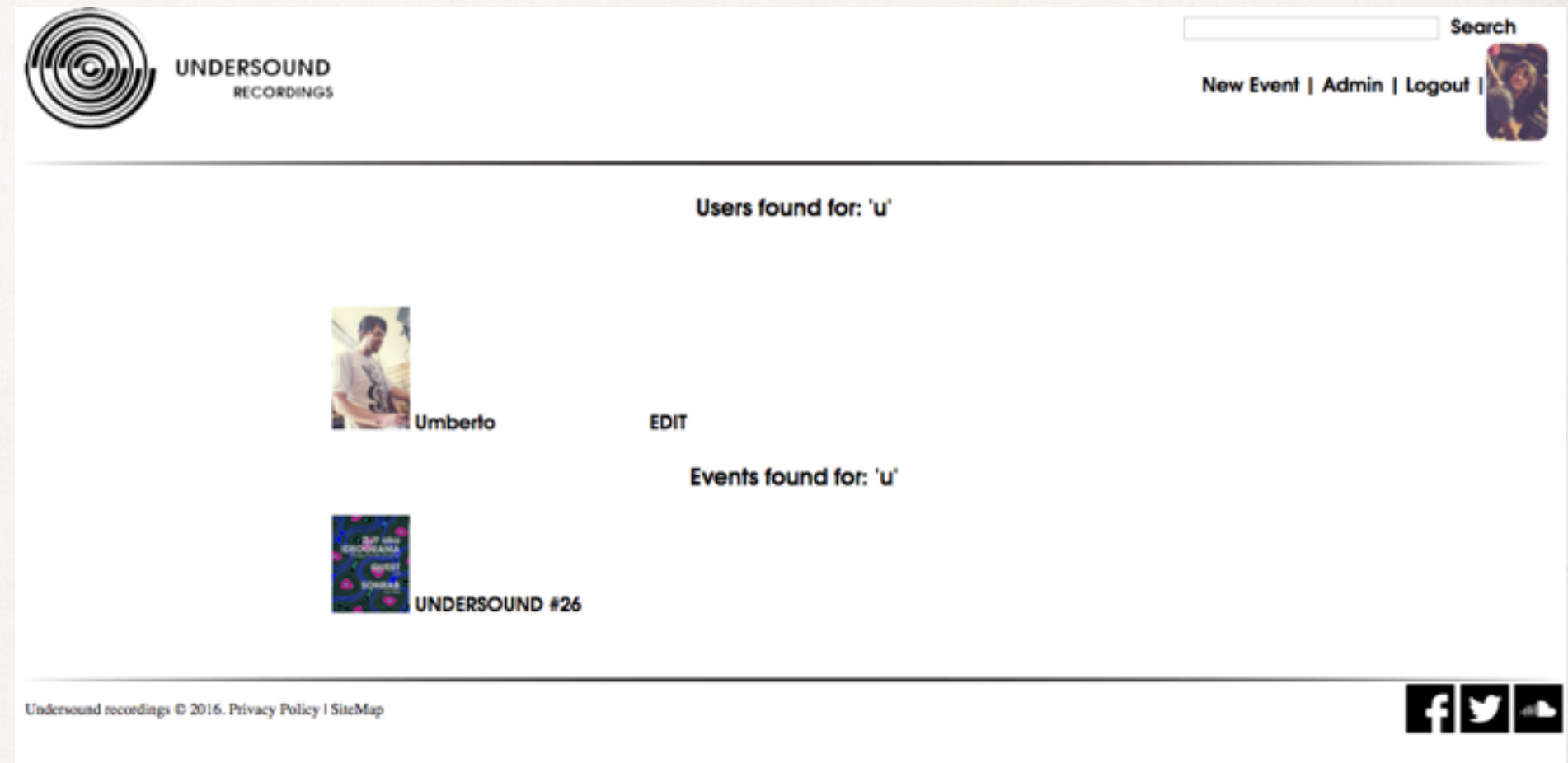
SEARCH

All the users will be able to to search between all the accounts and the events with key word input by a simple type form

search.php

Submitting the search form, sends to the page search.php with a query staring where the search key is the value of the input.

$search = $_POST['search'];

header('Location: http://localhost/undersound/search.php?value='.$search);

With a sql, the php finds all the usernames in 'account' and all the titles in 'post', similar to the var $search, takes from the $_GET

$sql = "select * from account where username LIKE '%{$search}%' ";

    $result = mysqli_query($conn, $sql);

$sql = "select * from post where title LIKE '%{$search}%' ";

    $result = mysqli_query($conn, $sql);