

ELABORATO ESAME DI STATO

TRACCIA DI SVOLGIMENTO:

-GESTIONE SITO DI TESINE-

“Un gruppo di studenti ha deciso di mettere su un sito per permettere a studenti come loro di pubblicare e così condividere le proprie tesine. Ogni tesina è caratterizzata da un titolo e da un autore, ma l'autore può essere visualizzato a meno a seconda che abbia voluto rimanere anonimo o no. Inoltre vengono anche indicate le materie coinvolte dalla trattazione e i relativi argomenti. È possibile segnalare l'anno di composizione e la scuola che lo studente frequentava quando ha composto la tesina.”

IPOTESI AGGIUNTIVE

qui elencate in modo da rendere chiaro ogni punto svolto nell'elaborato

- Aggiunta della funzione “elimina utente” per evitare sprechi di spazio e per garantire il rispetto dell'utente e dei suoi dati personali.
- Gestione admin da parte del webmaster -> attraverso l'uso della funzione “contattami” sarà possibile inviare una mail di richiesta dei permessi di amministrazione .
- Gestione lamentele/bug con la funzione “contattami”.
- Accettazione privacy policy in fase di registrazione.
- Eliminando una tesina vengono eliminati anche i relativi log in modo da liberare spazio.
- In un futuro il sito potrebbe necessitare di un certificato SSL
- Una tesina è scritta su una sola materia e tratta un solo argomento

SISTEMI E RETI

- Descrizione dei protocolli di rete:

TCP: In forma estesa “Transmission Control Protocol”, è un accordo standardizzato per la trasmissione dei dati tra diversi utenti di una rete informatica. TCP permette a due punti terminali all'interno di una rete informatica comune di realizzare una connessione attraverso cui può avvenire una trasmissione bidirezionale dei dati. Nell'ambito di questa connessione, le

eventuali perdite di dati vengono riconosciute e corrette automaticamente; per questo motivo TCP è considerato un protocollo affidabile. Prima di effettuare una connessione, i due estremi devono dichiarare apertamente il proprio indirizzo IP e la porta desiderata per la connessione. A questo punto è possibile cominciare la procedura denominata “Three way handshake” che viene così espletata :

1-> Il client invia un pacchetto SYN per sincronizzarsi contenente un numero casuale

2-> Il server risponde con un pacchetto SYN-ACK (da acknowledgment, conferma)

3-> Il client risponde nuovamente con un pacchetto ACK

Allo stesso modo la chiusura si effettua in con uno scambio di pacchetti denominati FIN e ACK

IPSEC: In forma estesa “IP security” è un insieme di protocolli che fornisce sicurezza al protocollo IP e altri livelli garantendo una connessione su un percorso sicuro. Uno dei principali protocolli che fanno parte di IPSEC è :

AH: In forma estesa “Authentication header“, è il protocollo che si occupa di verificare e garantire l'autenticità del mittente. Il mittente applica una funzione di tipo crittografico per rilasciare una sorta di “firma digitale” che sarà in un momento successivo interpretata dal destinatario. Questo protocollo permette anche di verificare l'integrità dei dati

HTTP: In forma estesa “Hypertext Transfer Protocol”, è un protocollo basato su linguaggio di testo che consente la comunicazione tra client e server attraverso internet. Il protocollo si basa sulla meccanica request-response che viene avviata dopo la fase iniziale di connessione garantita dal Three way handshake operata dal protocollo TCP. Ogni conversazione tra client e server parte con una richiesta da parte del client nella forma sotto indicata, la quale comprende il necessario per stabilire la connessione:

[method] [URI] [version]

[headers]

[body]

Method -> metodo di richiesta (es. GET)

Uri-> percorso univoco della risorsa richiesta

Version -> versione HTTP

Headers -> righe di valori singoli (esclusa la prima riga)

Body -> corpo del messaggio (i byte trasmessi)

A questo punto il server conosce l'obiettivo che deve raggiungere, viene elaborato perciò un messaggio di risposta così composto:

[version] [status code] [reason]

[headers]

[body]

Version -> versione HTTP

Status code-> codice generato dopo l'elaborazione della richiesta

Reason-> Ragione per la quale viene generato il codice di stato

Headers e body-> variano seconda dell'obiettivo da raggiungere

Il protocollo HTTP viene reso oggi più sicuro attraverso SSL o TLS, ovvero due meccanismi di crittografia asimmetrica. Grazie a questa implementazione nasce il protocollo HTTPS.

SSL: In forma estesa "Secure Socket Layer", è un protocollo crittografico posto tra livello applicazione e livello trasporto che ha il compito di assicurare la segretezza della comunicazione (tramite crittografia asimmetrica), l'integrità del messaggio (tramite funzioni di hash) e l'autenticazione di client e server (tramite scambio di certificati e chiave pubblica). SSL inizializza la connessione tramite l'SSL Handshake e si differenzia dal TSL in quanto suo predecessore (è leggermente meno sicuro a causa delle evoluzioni delle tecnologie di questi anni)

- Gestione della privacy :

In europa, la normativa di riferimento sull'argomento è il GDPR (General data protection regulation). Essendo un regolamento ha come caratteristica il fatto che non necessita di recepimento; ogni titolare di "trattamento di dati personali" ha dovuto perciò adeguarsi al testo di riferimento pubblicato sulla Gazzetta ufficiale europea a partire da maggio del 2016, con termine ultimo in maggio 2018. Il GDPR è stato redatto seguendo il principio fondamentale della tutela dei dati dei cittadini, diventato un effettivo diritto dopo il trattato di Lisbona. Grazie a questa normativa l'utente diventa "vero proprietario" del dato che fornisce comunicando il proprio consenso al trattamento ed è tutelato in caso di esportazione di dati all'estero o violazioni. In questo modo vengono rafforzati i diritti dell'interessato, il quale deve poter sapere se i suoi dati sono usati e come vengono usati per tutelare lui e l'intera collettività dai rischi insiti nel trattamento dei dati. All'interno della normativa sono indicati i codici di condotta e le procedure utili a certificare i trattamenti; vengono

inoltre previste le sanzioni previste per gli enti o le imprese che non offrono all'utente adeguata trasparenza e protezione. Il regolamento viene applicato sia nei processi di trattamento dei dati automatizzati o parzialmente tali sia nei procedimenti cosiddetti non automatizzati (es .archivi cartacei). Il GDPR non è applicabile in caso di persona deceduta, di trattamento a fini domestici/privati e in caso di utilizzo dati per un indagine.

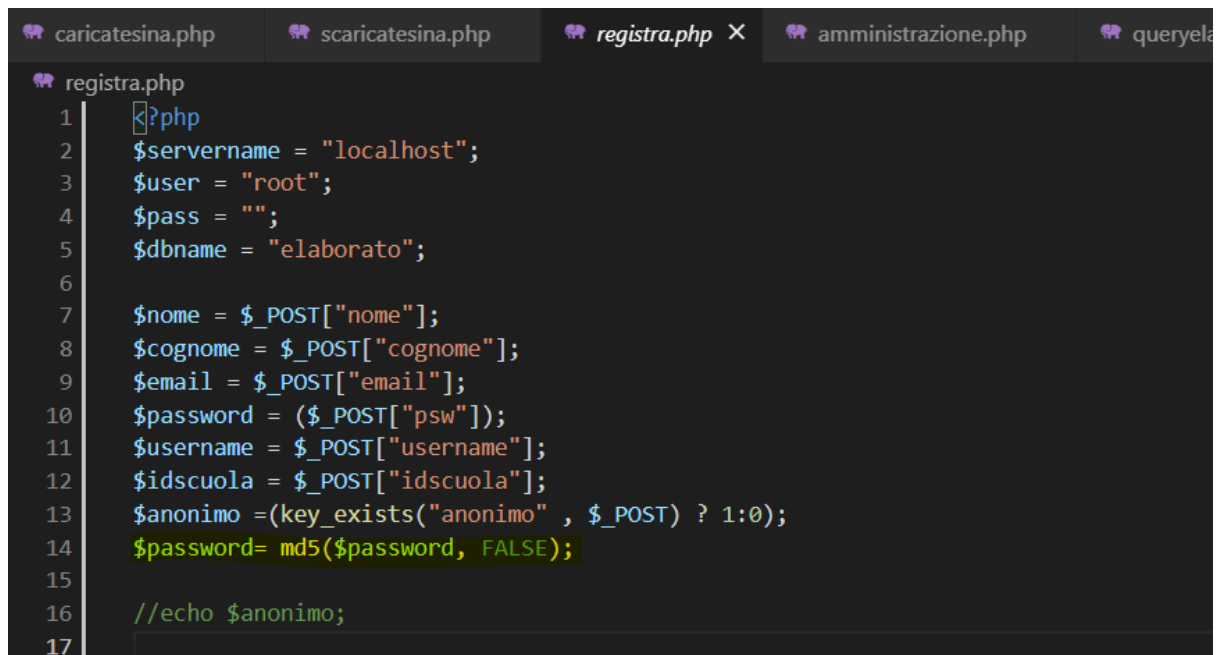
“Affermare che non si è interessati al diritto alla privacy perché non si ha nulla da nascondere è come dire che non si è interessati alla libertà di parola perché non si ha nulla da dire.”

Questa frase di Edward Snowden permette di capire quanto difenderci (e difendere l'utente) sia importante anche nei progetti in cui non sia presente un utilizzo di dati personali legati alla sfera dei pagamenti o dell'anagrafica. Analizzando il sito web secondo quest'ottica, ho potuto scegliere quali dati proteggere e quali funzionalità aggiungere per far sì che ogni utente possa sentirsi al sicuro durante la navigazione. Nel particolare ho scelto di concentrarmi su:

- L'anonimato : L'utente può scegliere di rimanere anonimo durante il processo di registrazione in modo tale da evitare che il suo nome e il cognome vengano esposti nel sito nella schermata di download. Questa scelta viene effettuata tramite una checkbox che immette nel database il valore 1 oppure 0 a seconda della scelta.
- Il diritto all'oblio : Nell'ottica di rispettare il diritto all'oblio è stata prevista una funzione che permette all'utente di eliminare il proprio account e i relativi dati collegati. In questo modo gli utilizzatori saranno in grado di eliminare le proprie informazioni in ogni momento consapevolmente, senza ricorrere a tempi di attesa o e-mail di contatto.

- **Hash functions or cryptographic algorithms:**

The MD5 is a cryptographic hash function created by Ronald Rivest in 1991. It is an irreversible unidirectional function and takes into input a string producing another long 128 bits. The process takes place very quickly and returned is such that it is highly unlikely to get with two different strings in the input one same output. The cryptography through an algorithm is applied in those situations where the digital signature is required or that still process sensitive data: with this function, we can check if the content of a file has changed (the function used by search engines to understand if a page needs to be indexed again). It is also widespread as support for authenticating users through server-side Web scripting languages (PHP in particular). In particular, on my website, I used this feature to ensure that passwords are safe in case of unauthorized database attack or access. In case of data theft, all the criminal has is a list of unreadable passwords.

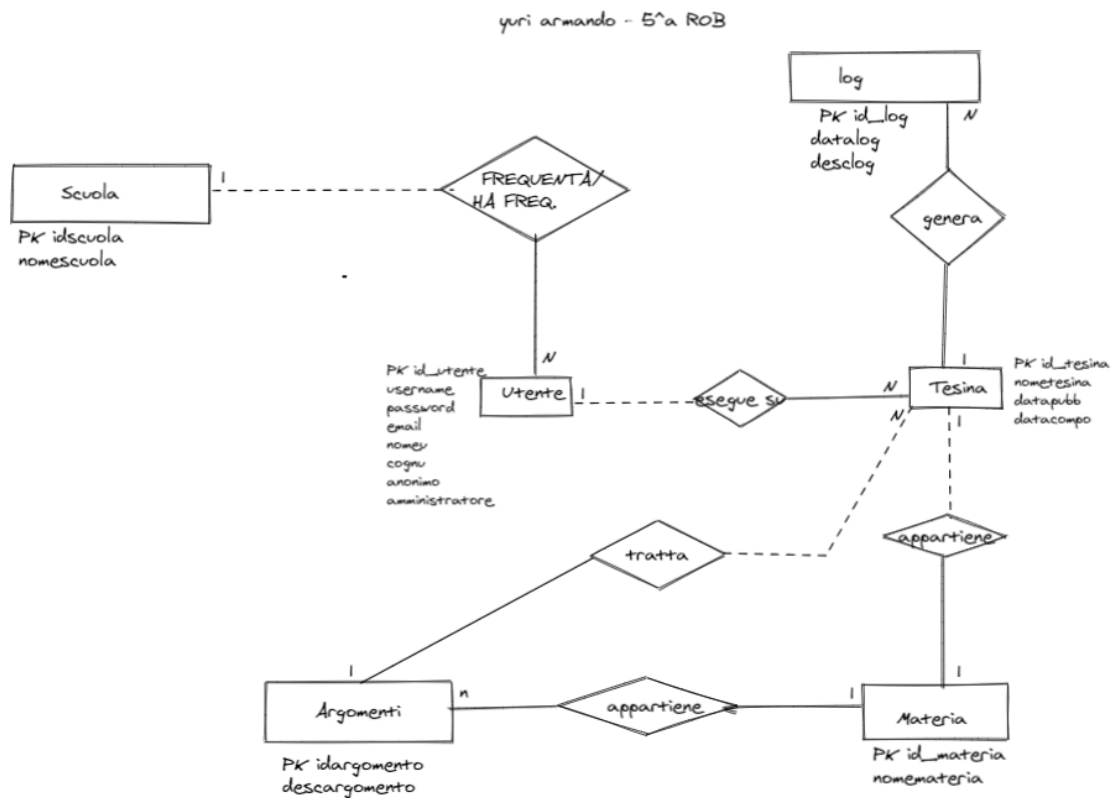


```
caricatesina.php  scaricatesina.php  registra.php X  amministrazione.php  queryela
registra.php
1  <?php
2  $servername = "localhost";
3  $user = "root";
4  $pass = "";
5  $dbname = "elaborato";
6
7  $nome = $_POST["nome"];
8  $cognome = $_POST["cognome"];
9  $email = $_POST["email"];
10 $password = ($_POST["psw"]);
11 $username = $_POST["username"];
12 $idscuola = $_POST["idscuola"];
13 $anonimo =(key_exists("anonimo" , $_POST) ? 1:0);
14 $password= md5($password, FALSE);
15
16 //echo $anonimo;
17
```

Example of the md5 function (line 14)

INFORMATICA -Progettazione base di dati-

1. Progettazione modello logico e concettuale



MAPPING:

Utente(IDutente, username, password, email, nome, cognome, anonimo, amministratore, idscuola*)

Tesina(IDtesina, nometesina, idautore*, datapubb, datacomp, idmateria*, idargomento*)

Materia(idmateria, nomemateria)

Argomenti(idargomento, descargomento, idMateria*)

Log(idlog, datalog, desclog, idtesina*)

Scuola(idscuola, nomescuola)

2. Progettazione modello logico e concettuale

Creazione tabella tesina:

```
CREATE TABLE `tesina` (  
  `idtesina` int(11) NOT NULL,  
  `nometesina` varchar(100) NOT NULL,  
  `idautore` int(11) NOT NULL,  
  `idmateria` int(11) NOT NULL,  
  `datapubb` date NOT NULL,  
  `datacomp` date NOT NULL,  
  `idArgomento` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Creazione tabella utenti:

```
CREATE TABLE `utenti` (  
  `IDutente` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(20) NOT NULL,  
  `password` char(32) NOT NULL,  
  `nome` varchar(20) NOT NULL,  
  `cognome` varchar(20) NOT NULL,  
  `email` varchar(30) NOT NULL,  
  `anonimo` tinyint(1) NOT NULL,  
  `amministratore` tinyint(1) DEFAULT 0,  
  `idscuola` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Creazione tabella scuola:

```
CREATE TABLE `scuole` (  
  `idscuola` int(11) NOT NULL,  
  `nomescuola` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Creazione tabella log:

```
CREATE TABLE `log` (  
  `idlog` int(11) NOT NULL,  
  `datalog` date NOT NULL,  
  `desclog` varchar(50) NOT NULL,  
  `idtesina` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Creazione tabella materie:

```
CREATE TABLE `materie` (  
  `idmateria` int(11) NOT NULL,  
  `nomemateria` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Creazione tabella argomento:

```
CREATE TABLE `argomento` (  
  `idargomento` int(11) NOT NULL,  
  `idmateria` int(11) NOT NULL,  
  `descargomento` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Per creare i vincoli di associazione e le chiavi primarie tra tabella e tabella la funzione utilizzata è stata ALTER TABLE.

Chiavi e limiti della tabella tesina:

```
ALTER TABLE `tesina`  
  ADD PRIMARY KEY (`idtesina`),  
  ADD KEY `idautore` (`idautore`),  
  ADD KEY `idmateria` (`idmateria`),  
  ADD KEY `idArgomento` (`idArgomento`);  
ALTER TABLE `tesina`  
  ADD CONSTRAINT `tesina_ibfk_1` FOREIGN KEY (`idautore`) REFERENCES `utenti` (`IDutente`),  
  ADD CONSTRAINT `tesina_ibfk_4` FOREIGN KEY (`idmateria`) REFERENCES `materie` (`idmateria`),  
  ADD CONSTRAINT `tesina_ibfk_5` FOREIGN KEY (`idArgomento`) REFERENCES `argomento` (`idargomento`);  
COMMIT;
```

Chiavi e limiti della tabella utenti:

```
ALTER TABLE `utenti`  
  ADD PRIMARY KEY (`IDutente`),  
  ADD UNIQUE KEY `username` (`username`),  
  ADD KEY `idscuola` (`idscuola`);  
ALTER TABLE `utenti`  
  ADD CONSTRAINT `utenti_ibfk_1` FOREIGN KEY (`idscuola`) REFERENCES `scuole` (`idscuola`);  
COMMIT;
```

Chiavi e limiti delle tabelle argomento e materie :

```
ALTER TABLE `argomento`  
  ADD PRIMARY KEY (`idargomento`),  
  ADD KEY `idmateria` (`idmateria`);  
ALTER TABLE `argomento`
```



```
ADD CONSTRAINT `argomento_ibfk_1` FOREIGN KEY (`idmateria`)
REFERENCES `materie` (`idmateria`);
COMMIT;

ALTER TABLE `materie`
  MODIFY `idmateria` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=4;
COMMIT;
```

Chiavi e limiti della tabella log :

```
ALTER TABLE `log`
  ADD PRIMARY KEY (`idlog`),
  ADD KEY `log_ibfk_1` (`idtesina`);
ALTER TABLE `log`
  ADD CONSTRAINT `log_ibfk_1` FOREIGN KEY (`idtesina`) REFERENCES
`tesina` (`idtesina`) ON DELETE CASCADE;
COMMIT;
```

Chiavi e limiti della tabella scuole:

```
ALTER TABLE `scuole`
  ADD PRIMARY KEY (`idscuola`);
COMMIT;
```

-Sviluppo del sito Web-

Il sito web è stato realizzato mediante l'utilizzo di HTML, CSS, PHP e PDO. Per quanto riguarda la parte grafica ho costruito lo scheletro del sito dopo aver realizzato uno schema dell' UI a mano. In una prima fase ho creato la pagina "template" in modo che riuscisse ad avere elementi che rimanevano tali in ogni pagine del sito (in modo da evitare stili diversi e differenze troppo evidenti nell'interfaccia). Eseguita questa prima parte e completato il "template" ho effettuato su internet una ricerca volta a trovare la libreria css più consona allo stile del template, in questo modo ho potuto selezionare i vari elementi utilizzati e le relative classi (es. bottoni, form) e aggiungerli man mano che le pagine prendevano forma . La libreria utilizzata è [W3.css](#) ; è stata utilizzata solamente per lo stile generale del sito mentre per il login e per le pagine di upload, download e amministrazione ho aggiunto linee di codice alla libreria sopracitata e creato un secondo file denominato "csslogin.css".

Terminata l'ideazione della parte grafica ho utilizzato php e la sua estensione pdo per le funzioni necessarie al funzionamento del sito.

NB: Per comodità, l'HTML è inserito quasi totalmente nelle pagine.php in modo da avere meno file e meno passaggi durante la navigazione tra le varie pagine.

SCHERMATA DI LOGIN + SCHERMATA DI REGISTRAZIONE

The image displays two screenshots of a web application interface, both set against a black background. The top screenshot shows a 'Login' form with a title 'Login' in bold. It features a text input field containing 'yuriarmando', a password input field with masked characters, and an orange 'Avanti' button. Below the button is a link: 'Non sei ancora registrato? Fallo ora'. The bottom screenshot shows a 'Crea account' form with a title 'Crea account' in bold. It includes input fields for 'yuriarmando', 'Cognome', 'E-mail', and 'Username', followed by a masked password field. There is a dropdown menu for 'Scegli la scuola' with 'ITIS M. DELPOZZO' selected. A checkbox labeled 'Vuoi rimanere anonimo?' is present. An orange 'Registrati' button is at the bottom, with a link below it: 'Possiedi già un account? Vai al login'.

Attraverso la homepage, utilizzando il bottone login/registrazione, l'utente viene indirizzato alla pagina logins.php . Questa pagina gestisce sia il login che la registrazione mediante l'utilizzo di un script che cambia la card. L'utente sarà quindi in grado di effettuare login e sign up dalla stessa pagina (immettendo informazioni essenziali quali la scelta di essere anonimo e la scuola frequentata).

Script "card changer" (richiamato da logins.php e presente in jslogin.js):

Si occupa essenzialmente di gestire un "Onclick" per cambiare il form. Form-hidden indica la parte nascosta che viene mostrata o tagliata a seconda del link che compare a fine form.

#login, #linkCreateAccount sono i nomi richiamati nella pagina php utili a compiere il compito.

```
document.addEventListener("DOMContentLoaded", () => {
    const loginForm = document.querySelector("#login");
    const createAccountForm = document.querySelector("#createAccount");

    document.querySelector("#linkCreateAccount").addEventListener("click", e
=> {
        e.preventDefault();
        loginForm.classList.add("form--hidden");
        createAccountForm.classList.remove("form--hidden");
    });

    document.querySelector("#linkLogin").addEventListener("click", e => {
        e.preventDefault();
        loginForm.classList.remove("form--hidden");
        createAccountForm.classList.add("form--hidden");
    });
});
```

Vuoi rimanere anonimo?

La gestione dell'anonimato viene eseguita mediante la spunta di una checkbox in fase di registrazione. Il valore immesso varia da 0 a 1 e viene elaborato nel file .php (seconda immagine) grazie alla funzione `key_exists`. A questo punto il valore viene caricato nel database per essere gestito successivamente nella schermata di download.

```
<input type="checkbox" id="c1" name="anonimo" value="anonimo" />
<label for="c1"><span></span>Vuoi rimanere anonimo?</label>
```

```
$nome = $_POST["nome"];
$cognome = $_POST["cognome"];
$email = $_POST["email"];
$password = ($_POST["psw"]);
$username = $_POST["username"];
$idscuola = $_POST["idscuola"];
$anonimo =(key_exists("anonimo" , $_POST) ? 1:0);
$password= md5($password, FALSE);

//echo $anonimo;
```

SCHERMATA UPLOAD TESINA

Carica la tua tesina

Scegli file | Nessun file selezionato

Data di creazione gg/mm/aaaa

Scegli la materia
italiano

Scegli l'argomento
Il neorealismo e i suoi autori

carica

Il funzionamento di questo elemento del sito web è reso possibile da due file .php . Il primo contiene il codice necessario alla raccolta dei dati e all'invio degli stessi, il secondo si occupa di eseguire le query sul database e di fare l'upload del file. L'upload dei file è perfettamente funzionante così come la categorizzazione e collegato al download anche se sarà in futuro necessario implementare il controllo sulle estensioni e sulla dimensione per avere un prodotto più completo.

Upload.php:

Panoramica delle query necessarie all'upload e della funzione move_uploaded_file() che si occupa di prendere il nome del file e il path di destinazione per poi spostarlo nella directory.

```
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $user, $pass);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO tesina (idtesina, nometesina, datacomp, datapubb, idautore, idmateria, idargomento)
    VALUES ('', '$nometesina', '$datacomp', '$datapubb', (SELECT idutente FROM utenti WHERE username = '$autore'), '$materia', '$argomento')";
    $sqllog= "INSERT INTO log (datalog, desclog, idtesina)
    VALUES ('$datapubb', 'file caricato', (SELECT idtesina FROM tesina t WHERE idtesina = (SELECT idtesina FROM tesina WHERE nometesina = '$nometesina'))";
    // Check if file already exists
    if (file_exists($target_file)) {
        $msg="IL FILE E' GIA' PRESENTE";
        echo $msg;
    }else{
        if(move_uploaded_file($_FILES['file']['tmp_name'], $_upload_path.$_FILES['file']['name'])){
        }else{
            echo "File vuoto";
        }
    }
}
```

W.I.P.: Controllo estensioni e collegamento tra i due menù dropdown in modo da associare materie e argomento anche nel frontend e non solo sul database.

Schermata di caricamento non riuscito (al momento è presente solo un messaggio sotto la schermata di caricamento riuscito nel caso l'upload dia esito negativo)

SCHERMATA DOWNLOAD TESINA

Questa schermata presenta la funzione di ricerca con filtro eseguita tramite un menù a tendina i cui argomenti vengono aggiornati ad ogni refresh della pagina. Alla pressione del tasto cerca viene eseguita la query responsabile della ricerca (legata poi a codice php per il download). Essenziale il foreach finale in modo da avere tutti i risultati riga per riga.

```

if($_SERVER["REQUEST_METHOD"]=="GET" && key_exists("argomento" , $_GET)){
    $argomento = $_GET["argomento"];
    $dbh = new PDO('mysql:host=localhost;dbname=elaborato', $user, $pass);
    $row = [];

    $arr_txt = explode(" ", $argomento);
    $sql = "SELECT * FROM tesina WHERE ";
    for ($i=0; $i<count($arr_txt); $i++)
    {
        if ($i > 0)
        {
            $sql .= " AND ";
        }
        $sql .= "(idargomento LIKE '" . $arr_txt[$i] . "%')";
    }
    $sql .= " ORDER BY idtesina ";
    $query = $dbh -> prepare($sql);
    $query -> execute();
    $res = $query->fetchAll(PDO::FETCH_ASSOC);
    if (count($res) == 0)
    {
        echo "nessun risultato <br>";
    }
    else
    {
        foreach($res as $row){
            echo "<br>";
            echo '<li><a href="download.php?nometesina='.$row['nometesina'].' " target="_blank">'.$row['nometesina'].' </a></li>';
            echo "<br>";
        }
    }
}

```

Nel div in bianco si trova invece la funzione che ci permette di scaricare ogni tesina presente sul sito.

```

<?php

$query = $dbh->prepare("SELECT * FROM `tesina` ORDER BY datapubb");
$query->execute();
$sqlutente = $dbh->prepare("SELECT * FROM utenti u, tesina t, scuole s WHERE u.IDutente = t.idautore AND s.idscuola = u.idscuola");
$sqlutente->execute();
$resutente = $sqlutente -> fetch(PDO::FETCH_ASSOC);

while($fetch = $query->fetch(PDO::FETCH_ASSOC)){
?>
<tr>
<td><?php echo $fetch['nometesina'];></td>
<td><?php echo $resutente["anonimo"]==1?"anonimo":$resutente['username'];></td>
<td><?php echo $fetch['datacomp'];></td>
<td><?php echo $resutente['nomescuola'];></td>
<td><a href="download.php?nometesina='.$fetch['nometesina'].' " class="btn btn-primary">Download</a></td>
</tr>
<?php

```

Si può notare come il valore anonimo citato in precedenza venga utilizzato ora per garantire l'anonimato: se in fase di registrazione l'utente sceglie di rimanere anonimo, il valore nel database del campo anonimo sarà settato a 1. Questo valore viene recuperato eseguendo la query sqlutente e utilizzato per mostrare o meno il nome dell'autore della tesina.

SCHERMATA DI AMMINISTRAZIONE (W.I.P. PARZIALE)

Id log	Data log	Id tesina	Desc log
27	2030-05-21	86	file caricato

ELIMINA/MODIFICA UNA TESINA

id tesina	nometesina	id materia	id argomento	Azione
86	provaneorealismo2.txt	3	2	Modifica Elimina

La schermata di amministrazione viene mostrata all'utente solo se esso è effettivamente amministratore. L'utente viene elevato a amministratore soltanto tramite l'intervento del webmaster (questo è possibile mandando una mail con il bottone "contattami"). L'admin check viene effettuato con un if che controlla se l'utente nella sessione corrente ha il campo amministratore con valore 1 e carica il body html di conseguenza.

```
<body>
<?php
session_start();
if($_SESSION["admin"]==1){
|
?>
```

Nel caso opposto viene mostrata la scritta "non amministratore".

La gestione dei log viene effettuata tramite una tabella la cui query collegata presenta tutti i dati presenti (`$sql= ("SELECT * FROM log") ;`)

Nell'immagine successiva è presente il codice eseguito alla pressione del bottone modifica relativo a ogni tesina. Questo codice si occupa di aggiornare il database e di creare una pagina dedicata all'inserimento dei nuovi dati.

```
<?php
$pdo = new PDO('mysql:host=localhost;dbname=elaborato', "root", "");
$msg = '';
// Check if the contact id exists, for example update.php?id=1 will get the contact with the id of 1
if (isset($_POST['nometesina'])) {
    if (empty($_POST)) {
        print_r($_POST);
        $datapubb = date('d-m-y');
        $nometesina = isset($_POST['nometesina']) ? $_POST['nometesina'] : NULL;
        $sqllog = $pdo->prepare("INSERT INTO log (datalog, desclog, idtesina)
VALUES ('$datapubb', 'file modificato', (SELECT idtesina FROM tesina t WHERE idtesina = (SELECT idtesina FROM tesina WHERE idtesina = ?)))");
        // This part is similar to the create.php, but instead we update a record and not insert
        $sqllog -> execute();
        $idmateria = isset($_POST['idmateria']) ? $_POST['idmateria'] : NULL;
        $idargomento = isset($_POST['idargomento']) ? $_POST['idargomento'] : NULL;
        $idtesina = isset($_POST['idtesina']) ? $_POST['idtesina'] : NULL;
        if($nometesina && $idmateria && $idargomento && $idtesina){
            // Update the record
            $stmt = $pdo->prepare('UPDATE tesina SET nometesina = ?, idmateria = ?, idargomento = ? WHERE idtesina = ?');
            $stmt->execute([$nometesina, $idmateria, $idargomento, $idtesina]);
            $msg = 'Tesina aggiornata!';
            header("Location: amministrazione.php");
        }
    } else {
        $msg2 = "Servono tutti e 4 i parametri";
        function EchoMessage($msg, $redirect){
            echo '<script type="text/javascript">
            alert("'" . $msg2 . "'");
            window.location.href = "'" . $redirect . "'";
            </script>';
        }
    }
}
},
```

nometesina=86

Aggiorna Tesina :

Nome tesina idArgomento idMateria

La pagina è ora in fase di WIP per quanto riguarda lo stile ma è perfettamente funzionante.

W.I.P.: Eliminazione utente per garantire il diritto all'oblio, dati utente a schermo.

-QUERY SQL-

Queste le interrogazioni da fornire in linguaggio SQL e da implementare in PHP oltre a quelle richieste per la realizzazione del sistema:

1. Per ogni scuola, il numero di tesine presenti per ogni materia
2. La materia che è trattata di più in assoluto
3. le tesine presentati da studenti della stessa scuola dell'utente

```
1: "SELECT s.nomescuola, m.nomemateria, COUNT(*) AS tesinettrattate
    FROM materie m, tesina t, scuole s
    WHERE m.idmateria = t.idmateria
    GROUP BY s.idscuola, m.nomemateria"
```

2:

```
(SELECT MAX(tesinettrattate) AS tesinet
    FROM (SELECT COUNT(*) AS tesinettrattate
          FROM tesine t, materie m
          WHERE m.idmateria == t.idmateria)
    )";
```

3:

```
"SELECT t.nometesina
FROM tesina t, utenti u
WHERE t.idautore = u.IDutente AND u.idscuola = (
SELECT DISTINCT idscuola
FROM utenti WHERE username= 'admin')");
```