

Online notes submission

Sommario

1	Analisi della consegna	4
1.1	La consegna.....	4
1.2	La consegna ampliata	5
1.3	Panoramica delle funzionalità da implementare	6
1.4	Suddivisione in fasi	7
2	Implementazione della base di dati	7
2.1	Costruzione del schema ER	7
2.1.1	L'individuazione delle entità	7
2.1.2	La realizzazione “entità per entità”	8
2.1.2.1	Utente	8
2.1.2.2	Topic.....	8
2.1.2.3	Skin.....	9
2.1.2.4	Documento	9
2.1.3	Schema ER finale	11
2.2	Ristrutturazione dello schema ER	11
2.3	Traduzione in schema relazionale	11
2.3.1	La traduzione delle entità	12
2.3.1.1	Skin.....	12
2.3.1.2	Utente	12
2.3.1.3	Documento	13
2.3.1.4	Topic.....	13
2.3.2	La traduzione delle relazioni	14
2.3.2.1	Utilizzo.....	14
2.3.2.2	Upload.....	14
2.3.2.3	Inclusione	15
2.3.3	Schema relazionale finale	16
2.4	La creazione della base di dati con PHPMyAdmin	17
2.4.1	La creazione del DB.....	17

2.4.2	L'ordine di creazione delle tabelle.....	19
2.4.3	La creazione delle tabelle.....	19
2.4.3.1	Skin.....	19
2.4.3.2	Topic.....	20
2.4.3.3	Utente	20
2.4.3.4	Documento	21
2.4.4	Schema di base di dati finale.....	21
3	La realizzazione del sito.....	22
3.1	Pagine web necessarie.....	22
3.2	Lo script sul sito principale.....	24
3.2.1	Il sito principale	24
3.2.2	Google Sites.....	26
3.2.3	Il form per la registrazione	27
3.2.4	Il form per il login.....	28
3.3	L'autenticazione	28
3.3.1	Le funzioni principali	28
3.3.2	Le variabili superglobali utilizzate.....	31
3.3.3	Registrazione.php	31
3.3.4	Login.php.....	32
3.4	Le pagine del sito.....	33
3.4.1	Lemiepubblicazioni.php	33
3.4.1.1	L'interfaccia con l'utente.....	34
3.4.1.2	Lo script	35
3.4.1.3	La stampa della tabella.....	36
3.4.1.4	L'eliminazione dei documenti.....	37
3.4.2	Carica.php	40
3.4.2.1	Lo script	40
3.4.2.2	Upload.php	40
3.4.3	Menu.php.....	43
3.4.3.1	Cambio_username.php.....	43
3.4.3.2	Cambio_password.php.....	45
3.4.3.3	Cambia_skin.php	45
3.4.3.4	Elimina_profilo.php.....	48
3.4.3.5	Logout.php	49
4	La pubblicazione online.....	50

4.1	Occorrente	53
4.2	La richiesta dell'IP pubblico statico	53
4.3	Il port mapping.....	54
4.4	Settaggio dell'IP privato statico.....	55
4.5	Configurazione web server	56
4.6	Creazione sottodomainio	57
5	La sicurezza.....	59
5.1	L'aggiunta del servizio HTTPS.....	59
5.1.1	Ottenere un certificato	60
5.1.2	Aggiunta del certificato.....	61
5.2	La prevenzione dagli SQL injection.....	62
5.2.1	Le SQL injection	62
5.2.2	Multiple query injection.....	63
5.2.3	Boolean value injection.....	64
5.2.3.1	Registrazione → username	65
5.2.3.2	Registrazione → password.....	66
5.2.3.3	Accedi → username	66
5.2.3.4	Accedi → password.....	66
5.2.4	Soluzioni.....	67
6	Considerazioni finali	67
6.1	Ulteriori considerazioni sulla sicurezza.....	67
6.2	Fonti.....	68

1 Analisi della consegna

1.1 La consegna

Il candidato, implementi un sito che permetta agli utenti di effettuare la registrazione per accedere ad un'area riservata dove sia possibile inviare file inerenti a topic su cui gli autori del sito stanno attualmente cercando testi e visualizzarne il loro giudizio.

Illustrare:

- la progettazione concettuale e logica della base di dati;
- le pagine web del sito e le modalità di interazione con la base di dati;
- le tecniche di invio, memorizzazione e lettura dei file caricati;
- i metodi per la pubblicazione online del sito utilizzati.

Ipotesi al contorno:

- decidiamo di includere il sito da realizzare come parte di un altro sito più grande che abbiamo già sviluppato durante l'anno tramite un CMS (Content Management System);
- la traccia non specificati requisiti sulla sicurezza delle comunicazioni o tecniche per prevenire possibili attacchi alla base di dati. Decidiamo quindi di ampliare la traccia inserendo:
 - la prevenzione delle SQL injection;
 - l'aggiunta di un certificato SSL per la crittazione asimmetrica delle comunicazioni.

Queste aggiunte, ovviamente, non consentono una protezione da qualsiasi attacco possibile, ma rappresentano comunque un'importante aggiunta al nostro applicativo;

- possedendo già un dominio di secondo livello, utilizzeremo il record DNS già in nostro possesso per creare un dominio di terzo livello da assegnare all'indirizzo IP pubblico che utilizzeremo per la pubblicazione;
- diamo anche la possibilità all'utente di cambiare le proprie credenziali, la propria skin (template realizzato fogli di stile) e di cancellare la propria utenza.

1.2 La consegna ampliata

In verde abbiamo evidenziato le parti aggiunte dalle ipotesi al contorno:

Il candidato, implementi un sito che, **accessibile da un sito già realizzato**, permetta agli utenti di effettuare la registrazione per accedere ad un'area riservata dove sia possibile inviare file inerenti a topic su cui gli autori del sito stanno attualmente cercando testi e visualizzarne il loro giudizio.

Illustrare:

- la progettazione concettuale e logica della base di dati;
- le pagine web del sito e le modalità di interazione con la base di dati;
- le tecniche di invio, memorizzazione e lettura dei file caricati;
- i metodi per la pubblicazione online del sito utilizzati **e per la personalizzazione dell'URL;**
- **la prevenzione dagli attacchi SQL di tipo injection;**
- **l'aggiunta del servizio HTTPS;**
- **la possibilità per l'utente di cambiare le proprie credenziali, la propria skin e cancellare la propria utenza;**

1.3 Panoramica delle funzionalità da implementare

Funzionalità	Occorrente	Mezzo utilizzato	Materia coinvolta	Argomento trattato
Base di dati	Applicativo per creare la base di dati e gestirla	phpMyAdmin con MySQL	Informatica teoria e laboratorio	Realizzazione dello schema relazionale e comandi DDL per la creazione delle tabelle e la definizione dei vincoli
Script lato server per l'interazione con la base di dati	Web server e linguaggio di script lato client	Apache (tramite usbwebserver) e PHP	Informatica laboratorio e TPSI teoria	I web server nelle architetture 3-tier e i linguaggi di scripting lato server
Aggiunta al sito già realizzato	CMS che permetta l'aggiunta di codice HTML/Javascript	Google Sites	TPSI laboratorio	I CMS e le loro funzionalità avanzate
Invio, eliminazione e lettura dei file caricati	Funzioni realizzate per questo scopo	Invio: move_uploaded_file() Eliminazione: unlink()	Approfondimento di informatica laboratorio	Approfondimento sulle funzioni del linguaggio PHP
Pubblicazione del sito	Indirizzo IP pubblico statico e port mapping	Router Fastweb (Fastgate)	Approfondimento di sistemi e reti teoria	Approfondimento sui servizi integrati dei gateway
Aggiunta del sottodomini	Programma/sito per la gestione del record DNS	Sito web di Godaddy	Sistemi e reti teoria	Il DNS ed i Resource Record
HTTPS	Certificate Authority che rilasci un certificato SSL	SSL For Free	Sistemi e reti teoria	La crittografia a chiave asimmetrica, il servizio HTTPS ed i certificati digitali
La protezione dalle SQL Injection	Funzioni per il controllo delle query	mysqli_query() mysqli_real_escape_string()	TPSI teoria e laboratorio (argomento non trattato nel corso dell'anno)	L'attacco SQL di tipo injection e la sua prevenzione

1.4 Suddivisione in fasi

Considerando l'importanza dei punti che la traccia prevede di sviluppare, procediamo nel seguente ordine:

1. Creazione della base di dati
 - 1.1. Individuazione delle entità
 - 1.2. Progettazione concettuale
 - 1.3. Progettazione logica
 - 1.4. Creazione della base di dati con phpMyAdmin
2. Realizzazione del sito
 - 2.1. Progettazione dello schema del sito
 - 2.2. Incorporazione degli script nel sito CMS
 - 2.3. Realizzazione delle pagine web
3. La pubblicazione del sito
 - 3.1. Ottenimento dell'IP pubblico
 - 3.2. Configurazione del port mapping sul proprio gateway
 - 3.3. Creazione ed aggiunta del sottodomini
4. Considerazioni ulteriori sulla sicurezza
 - 4.1. Implementazione del servizio HTTPS
 - 4.2. Prevenzione delle SQL injection

2 Implementazione della base di dati

2.1 Costruzione del schema ER

2.1.1 L'individuazione delle entità

Il candidato, implementi un sito che permetta agli **utenti** di effettuare la registrazione per accedere ad un'area riservata dove sia possibile inviare **file** inerenti a **topic** su cui gli autori del sito stanno attualmente cercando testi e visualizzarne il loro giudizio.

- la possibilità per l'utente di cambiare le proprie credenziali, la propria **skin** e cancellare la propria utenza;

Possiamo individuare 4 **entità**:

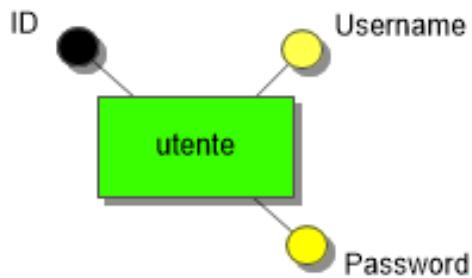
1. Utente;
2. File (documento);
3. Topic;
4. Skin.

2.1.2 La realizzazione “entità per entità”

2.1.2.1 Utente

Un utente sappiamo possedere:

- ID (identificativo univoco);
- Username;
- Password.

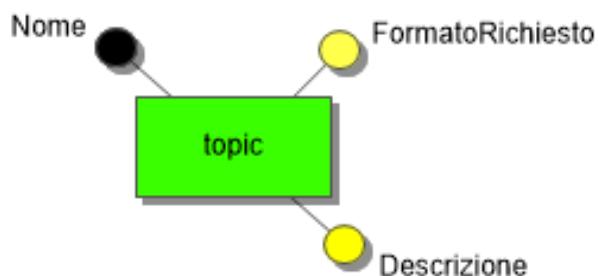


Nonostante lo Username possa essere utilizzato come identificativo, l'utente può cambiarlo. Perciò è preferibile utilizzare come identificativo un ID “permanente”.

2.1.2.2 Topic

Un topic deve necessariamente possedere:

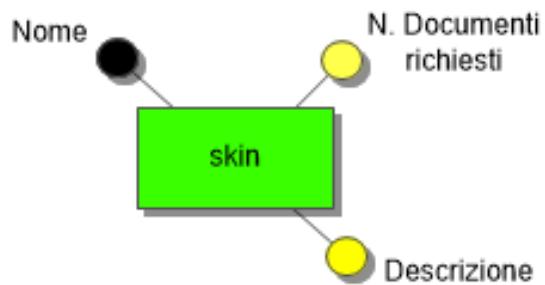
- Un nome (che farà da identificativo);
- Un attributo che descriva il formato in cui il file deve essere caricato;
- Una descrizione sul tipo di documento che gli autori stanno cercando (ad esempio: “breve e sintetico, che ponga l'enfasi su... ”).



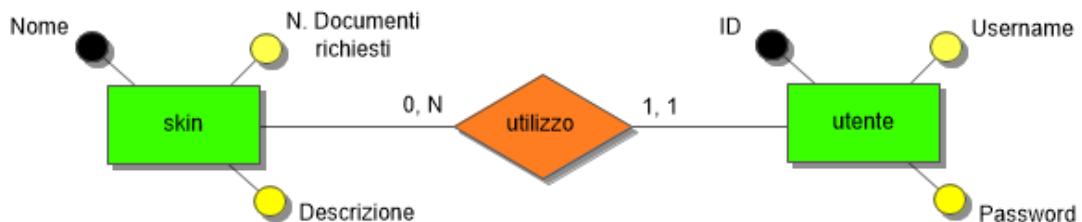
2.1.2.3 Skin

Una skin deve possedere:

- Un nome;
- Un numero di documenti richiesti per essere utilizzata (alcune skin possono essere premium);
- Una descrizione.



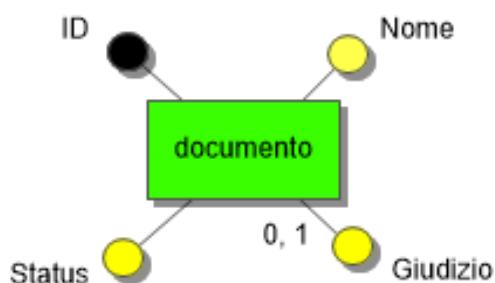
Ad un utente corrisponde una ed una sola skin, mentre una skin può essere assegnata da 0 ad N utenti:



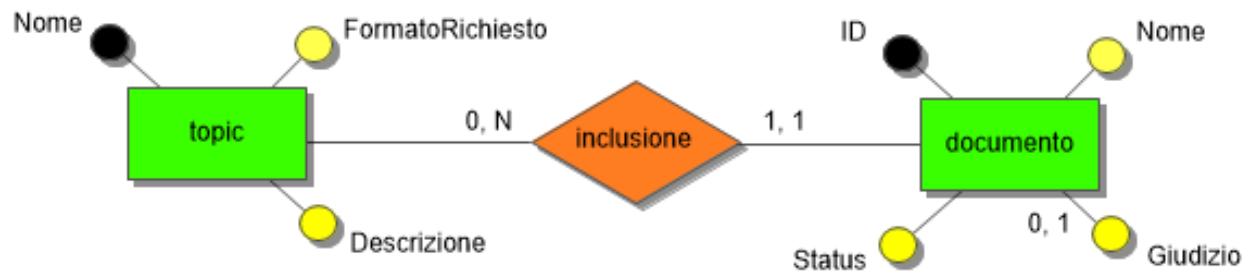
2.1.2.4 Documento

Un documento possiede:

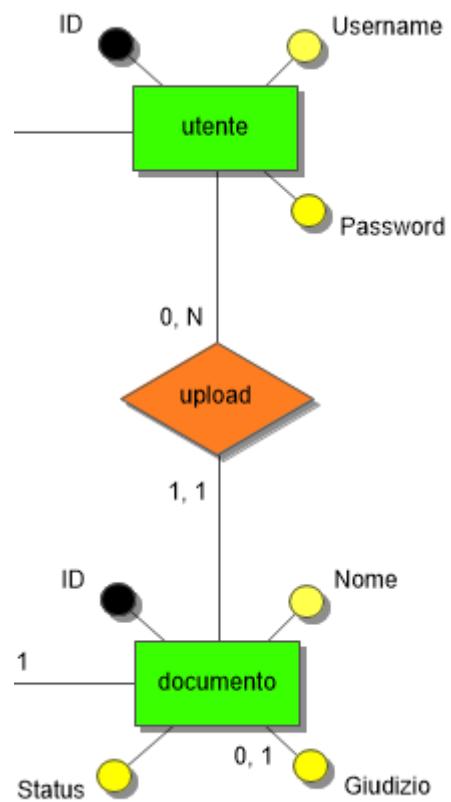
- Un ID;
- Un nome;
- Un giudizio (opzionale);
- Uno status (per rappresentare se è stato approvato, se non è per nulla conforme a ciò che gli autori cercano...).



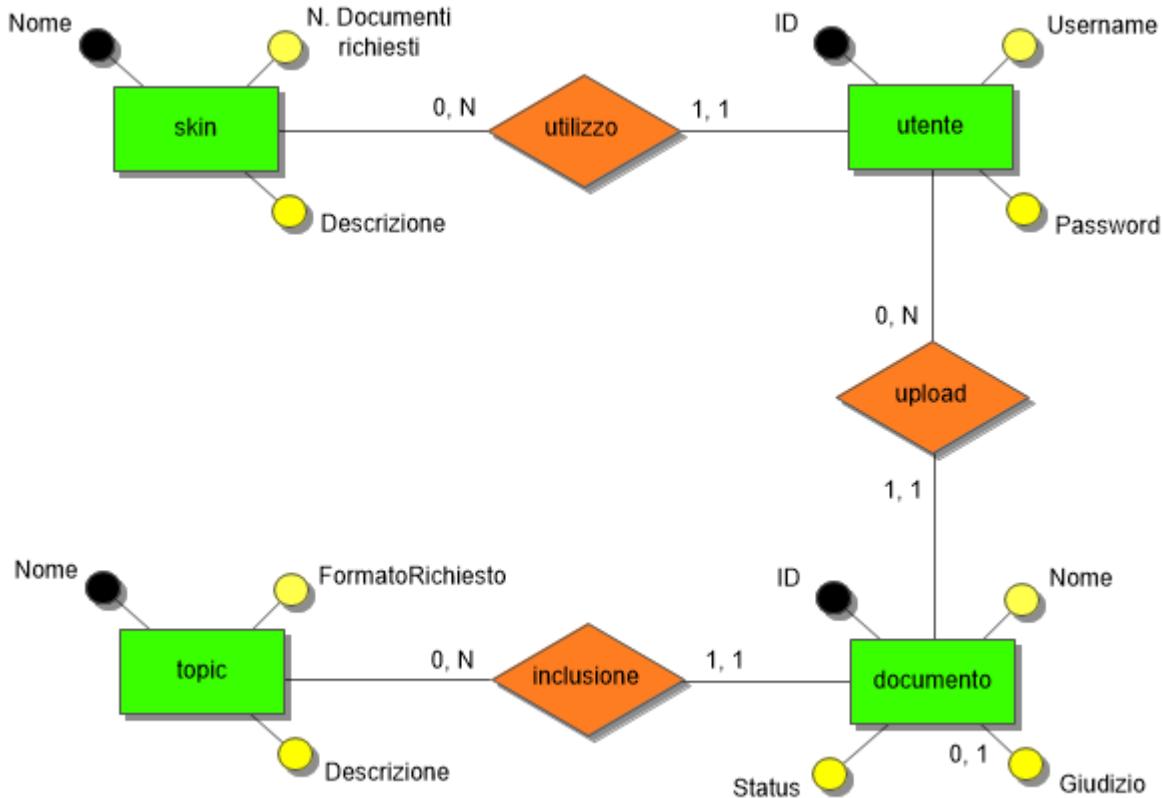
Ad un documento corrisponde uno e un solo topic (a cui possono corrispondere da 0 a N documenti):



Un documento è scritto da uno e un solo utente, che può aver caricato da 0 a N documenti:



2.1.3 Schema ER finale



2.2 Ristrutturazione dello schema ER

Lo schema non presenta elementi/configurazioni per cui si veda necessaria una ristrutturazione.

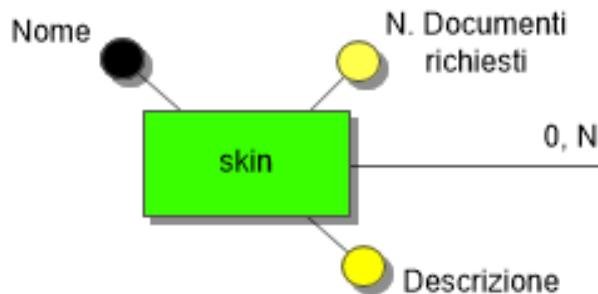
2.3 Traduzione in schema relazionale

Per la traduzione seguiamo il seguente “modus operandi”:

1. Traduzione delle entità;
2. Traduzione delle relazioni “molti a molti”;
3. Traduzione delle relazioni “uno a molti”;
4. Traduzione delle relazioni “uno a uno”.

2.3.1 La traduzione delle entità

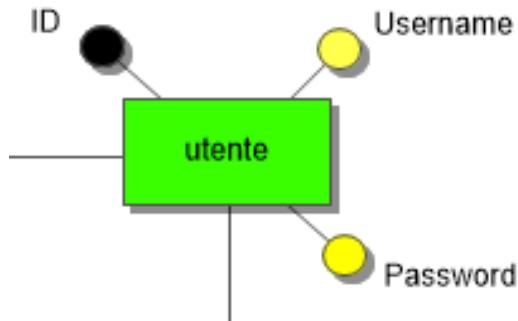
2.3.1.1 Skin



skin (Nome, DocumentiRichiesti, Descrizione)

Attributo	Dominio	Vincoli
Nome	varchar(255)	PRIMARY KEY
DocumentiRichiesti	int	NOT NULL DocumentiRichiesti >= 0
Descrizione	varchar(255)	NOT NULL

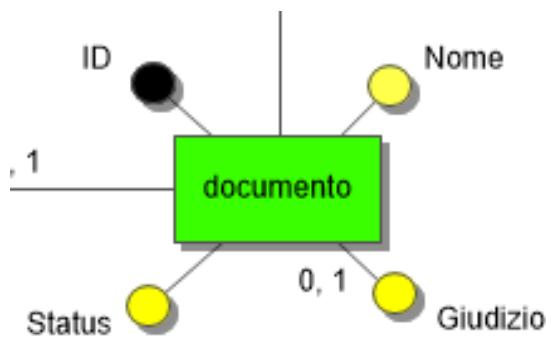
2.3.1.2 Utente



utente (ID, Username, Password)

Attributo	Dominio	Vincoli
ID	int	PRIMARY KEY
Username	varchar(255)	NOT NULL UNIQUE
Password	varchar(255)	NOT NULL

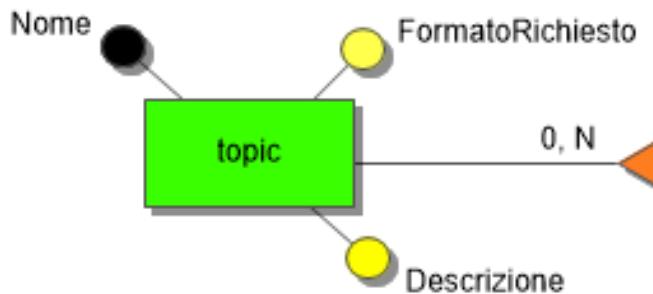
2.3.1.3 Documento



documento (ID, Nome, Giudizio, Status)*

Attributo	Dominio	Vincoli
ID	int	PRIMARY KEY
Nome	varchar(255)	NOT NULL
Giudizio	varchar(255)	--
Status	int	NOT NULL

2.3.1.4 Topic



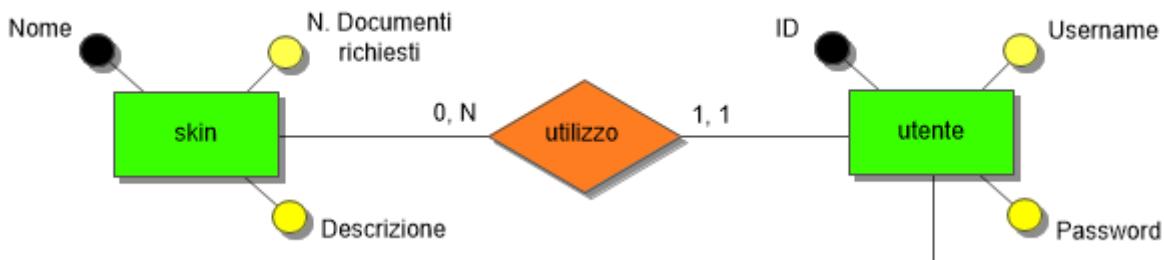
topic (Nome, FormatoRichiesto, Descrizione)

Attributo	Dominio	Vincoli
Nome	varchar(255)	PRIMARY KEY
FormatoRichiesto	varchar(255)	NOT NULL
Descrizione	varchar(255)	NOT NULL

2.3.2 La traduzione delle relazioni

Tutte e tre le relazioni sono di tipo “uno a molti” con partecipazione obbligatoria del lato “a molti”.

2.3.2.1 Utilizzo

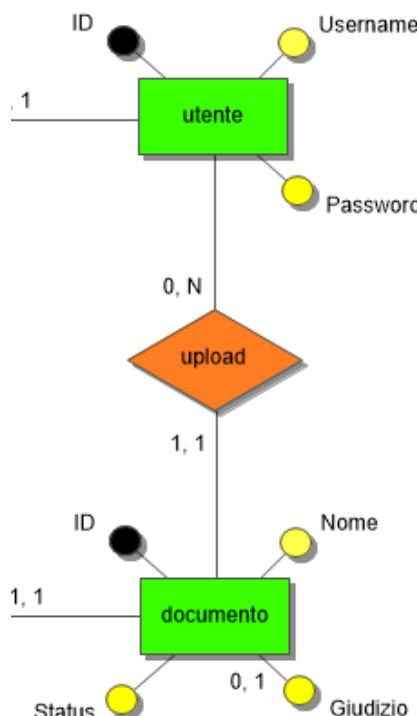


La tabella derivante dalla traduzione del lato “a molti” (ovvero “utente”) importa la chiave primaria dell’altra entità (ovvero “skin”) tramite il **vincolo di chiave esterna**:

utente (ID, Username, Password, FKNomeSkin)

Attributo	Dominio	Vincoli
FKNomeSkin	varchar(255)	NOT NULL FOREIGN KEY (FKNomeSkin) REFERENCES skin(Nome)

2.3.2.2 Upload

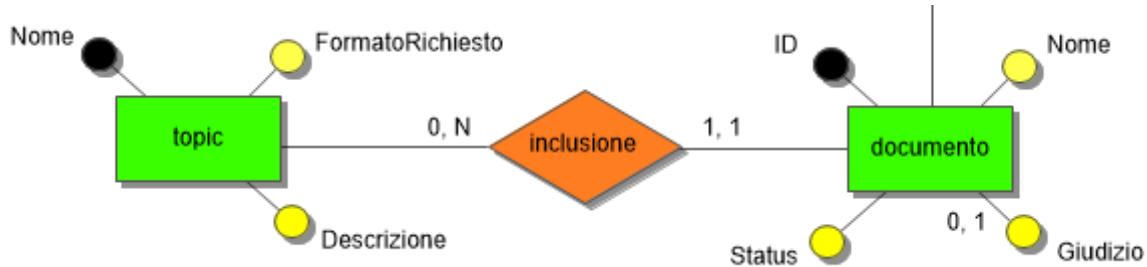


La tabella derivante dalla traduzione del lato “a molti” (ovvero “documento”) importa la chiave primaria dell’altra entità (ovvero “utente”) tramite il vincolo di chiave esterna:

documento (ID, Nome, Giudizio, Status, FKIDUtente)*

Attributo	Dominio	Vincoli
FKIDUtente	int	NOT NULL FOREIGN KEY (FKIDUtente) REFERENCES utente(ID)

2.3.2.3 Inclusione

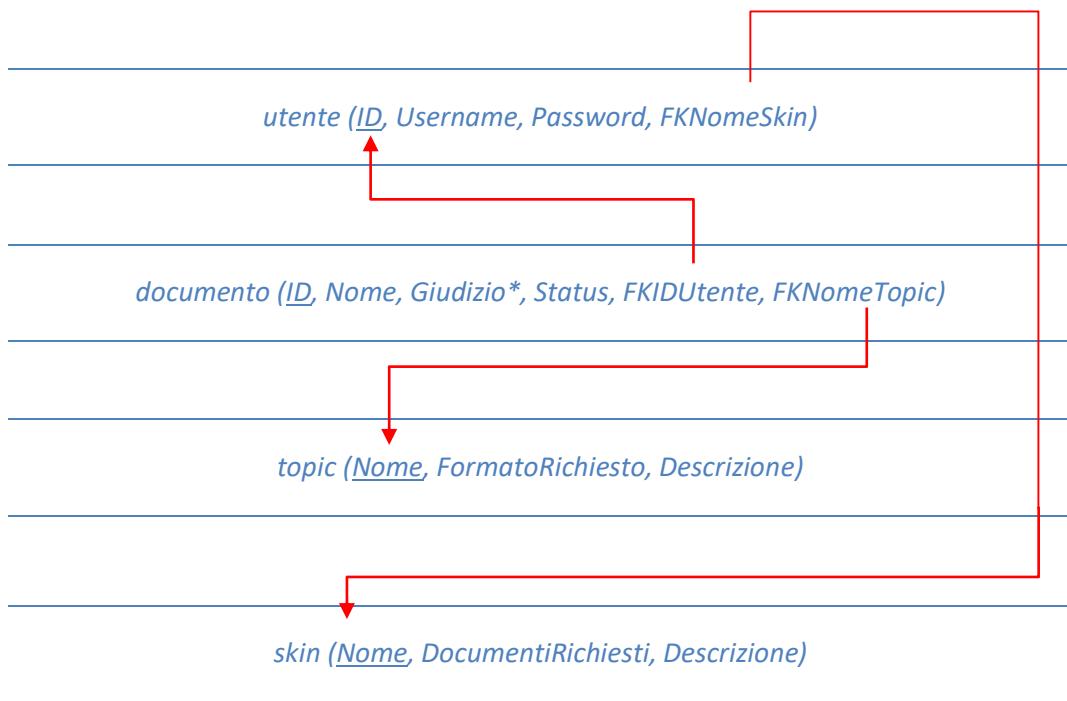


La tabella derivante dalla traduzione del lato “a molti” (ovvero “documento”) importa la chiave primaria dell’altra entità (ovvero “topic”) tramite il vincolo di chiave esterna:

documento (ID, Nome, Giudizio, Status, FKIDUtente, FKNomeTopic)*

Attributo	Dominio	Vincoli
FKNomeTopic	varchar(255)	NOT NULL FOREIGN KEY (FKNomeTopic) REFERENCES topic(Nome)

2.3.3 Schema relazionale finale



Attributo	Dominio	Vincoli
utente		
ID	int	PRIMARY KEY
Username	varchar(255)	NOT NULL UNIQUE
Password	varchar(255)	NOT NULL
FKNomeSkin	int	NOT NULL FOREIGN KEY (FKNomeSkin) REFERENCES skin(Nome)
documento		
ID	int	PRIMARY KEY
Nome	varchar(255)	NOT NULL
Giudizio*	varchar(255)	--
Status	int	NOT NULL
FKIDUtente	int	NOT NULL FOREIGN KEY (FKIDUtente) REFERENCES utente(ID)
FKNomeTopic	varchar(255)	NOT NULL FOREIGN KEY (FKNomeTopic) REFERENCES topic(Nome)
topic		
Nome	varchar(255)	PRIMARY KEY
FormatoRichiesto	varchar(255)	NOT NULL
Descrizione	varchar(255)	NOT NULL

skin

Nome	varchar(255)	PRIMARY KEY
DocumentiRichiesti	int	DocumentiRichiesti >= 0 NOT NULL
Descrizione	varchar(255)	NOT NULL

2.4 La creazione della base di dati con PHPMyAdmin

2.4.1 La creazione del DB

Tramite USBwebserver accediamo a PHPMyAdmin:



This screenshot shows the 'Connetti' (Connect) page of PHPMyAdmin. It has fields for 'Nome utente:' (root) and 'Password:' (left blank). At the bottom right is a red-bordered 'Esegui' (Execute) button. Below this, a section titled 'Default USBWebserver settings' lists 'Nome utente:' as root, 'Password:' as usbw, and 'Mysql port' as 3307.

Accediamo alla sezione dov'è possibile scrivere in linguaggio SQL:

The screenshot shows the PHPMyAdmin interface for a MySQL connection to 'localhost'. The top navigation bar has tabs for 'Database', 'SQL' (which is highlighted with a red box), 'Stato', 'Processi', 'Privilegi', 'Esporta', and 'Importa'. Below the tabs, there are two main sections: 'Impostazioni Generali' and 'Impostazioni di Presentazione'. The 'General Settings' section contains a 'Cambia password' link and a dropdown for 'Collation della connessione di MySQL' set to 'utf8_general_ci'. The 'Presentation Settings' section includes a 'Lingua - Language' dropdown set to 'Italiano - Italian', a 'Tema / Stile' dropdown set to 'pmahomme', a 'Dimensione font' dropdown set to '82%', and a 'Ulteriori impostazioni' link.

Per creare un database si utilizza il comando DDL “CREATE DATABASE”:

The screenshot shows the SQL query editor window. The query text area contains the command 'CREATE DATABASE account;'. Below the text area is a 'Cancella' button. At the bottom, there are fields for 'Delimitatori' (with a semicolon selected) and 'Mostra di nuovo questa query' (checked). On the far right, there is an 'Esegui' button, which is highlighted with a red box.

A questo punto il database è stato creato.

Per creare le tabelle potremmo utilizzare l'interfaccia grafica messa a disposizione da PHPMyAdmin, tuttavia preferiamo utilizzare solo ed esclusivamente la shell di inserimento dei comandi SQL.

Creare le tabelle, una per una, senza tornare su una appena creata, prevede di seguire un preciso ordine in base ai vincoli di chiave esterna definiti nello schema relazionale (si parte dalle tabelle senza vincoli di chiave esterna e si procede in una modalità quasi “gerarchica”).

2.4.2 L'ordine di creazione delle tabelle

	Tabella	Tabelle referenziate
1	skin	--
2	topic	--
3	utente	skin
4	documento	topic, utente

2.4.3 La creazione delle tabelle

2.4.3.1 Skin

skin		
<u>Nome</u>	varchar(255)	PRIMARY KEY
DocumentiRichiesti	int	DocumentiRichiesti >= 0 NOT NULL
Descrizione	varchar(255)	NOT NULL

```
CREATE TABLE skin
(
    Nome varchar(255) PRIMARY KEY,
    DocumentiRichiesti int NOT NULL CHECK(DocumentiRichiesti>=0),
    Descrizione varchar(255) NOT NULL
);
```

2.4.3.2 Topic

topic		
<u>Nome</u>	varchar(255)	PRIMARY KEY
<u>FormatoRichiesto</u>	varchar(255)	NOT NULL
<u>Descrizione</u>	varchar(255)	NOT NULL

```
CREATE TABLE topic
(
    Nome varchar(255) PRIMARY KEY,
    FormatoRichiesto varchar(255) NOT NULL,
    Descrizione varchar(255) NOT NULL
);
```

2.4.3.3 Utente

utente		
<u>ID</u>	int	PRIMARY KEY
<u>Username</u>	varchar(255)	NOT NULL UNIQUE
<u>Password</u>	varchar(255)	NOT NULL
<u>FKNomeSkin</u>	int	NOT NULL FOREIGN KEY (FKNomeSkin) REFERENCES skin(Nome)

```
CREATE TABLE utente
(
    ID varchar(255) PRIMARY KEY,
    Username varchar(255) NOT NULL UNIQUE,
    Password varchar(255) NOT NULL,
    FKNomeSkin varchar(255) NOT NULL,
    FOREIGN KEY (FKNomeSkin) REFERENCES skin(Nome)
);
```

2.4.3.4 Documento

documento		
ID	int	PRIMARY KEY
Nome	varchar(255)	NOT NULL
Giudizio*	varchar(255)	--
Status	int	NOT NULL
FKIDUtente	int	NOT NULL FOREIGN KEY (FKIDUtente) REFERENCES utente(ID)
FKNomeTopic	varchar(255)	NOT NULL FOREIGN KEY (FKNomeTopic) REFERENCES topic(Nome)

```
CREATE TABLE documento
(
    ID int PRIMARY KEY,
    Nome varchar(255) NOT NULL,
    Giudizio varchar(255),
    Status int NOT NULL,
    FKIDUtente varchar(255) NOT NULL,
    FKNomeTopic varchar(255) NOT NULL,
    FOREIGN KEY (FKIDUtente) REFERENCES utente(ID),
    FOREIGN KEY (FKNomeTopic) REFERENCES topic(Nome)
);
```

2.4.4 Schema di base di dati finale

Tabella	Azione	Righe	Tipo	Collation	Dimensione	Overhead
documento	Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	48.0 KiB	-
skin	Mostra Struttura Cerca Inserisci Svuota Elimina	1	InnoDB	latin1_swedish_ci	16.0 KiB	-
topic	Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
utente	Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	48.0 KiB	-
4 tabelle	Totali	1	InnoDB	latin1_swedish_ci	128.0 KiB	0 B

Selezione tutti / Deseleziona tutti Se selezionati: ▾

3 La realizzazione del sito

3.1 Pagine web necessarie

Il candidato, implementi un sito che, accessibile da un sito già realizzato, permetta agli utenti di effettuare la registrazione per accedere ad un'area riservata dove sia possibile inviare file inerenti a topic su cui gli autori del sito stanno attualmente cercando testi e visualizzarne il loro giudizio.

Illustrare:

- la progettazione concettuale e logica della base di dati;
- le pagine web del sito e le modalità di interazione con la base di dati;
- le tecniche di invio, memorizzazione e lettura dei file caricati;
- i metodi per la pubblicazione online del sito utilizzati e per la personalizzazione dell'URL;
- implementare una modalità di protezione dagli attacchi SQL di tipo injection;
- abilitare il servizio https;
- l'utente può cambiare le proprie credenziali, la propria skin e cancellare la propria utenza;

In rosa abbiamo evidenziato le funzionalità che la traccia prevede vengano messe a disposizione tramite il sito.

Queste funzionalità implicano la realizzazione di:

1. Pagina web per la registrazione;
2. Pagina web per il login;
3. Pagina web per il caricamento del file;
4. Pagina web per la visualizzazione dello storico degli upload;
5. Pagina web per la gestione delle credenziali, delle skin e che accolga la funzionalità di cancellazione della propria utenza.

Le pagine di registrazione e log-in saranno collocate sul sito già realizzato e possiederanno un collegamento al sito web “vero e proprio”.



Abbiamo 4 “categorie” di pagine:

- **Script HTML** sulle pagine del sito principale realizzato con il CMS;
- **Pagine di autenticazione;**
- **Pagine principali;**
- **Sottopagine.**

3.2 Lo script sul sito principale



Per prima cosa, implementiamo il codice HTML da incorporare alle due pagine web sul nostro sito già realizzato con il CMS “Google Sites”.

Il sito principale è accessibile [qui](#).

Iniziamo illustrando il sito principale ed il CMS che abbiamo utilizzato.

3.2.1 Il sito principale

Nel corso degli anni presso l’istituto, ho cercato di affinare sempre di più il metodo di studio e rendere accessibile il materiale da me prodotto a chiunque ne facesse richiesta.

Dopo aver provato a condividerlo sulla piattaforma di Google Drive, all’inizio di quest’anno ho pensato fosse più comodo e fruibile caricare il materiale, prodotto durante il corso dell’anno, su un sito web.

In realtà, al fine di permettere l’accesso ad appunti e dispense ai soli autorizzati, quasi tutti i documenti ed i file multimediali presenti nel sito sono caricati in Drive, ma accessibili solo a determinati account mail.

Se un estraneo tentasse di aprire uno dei documenti, dovrebbe aspettare la mia autorizzazione prima che il suo account mail possa avere accesso al file richiesto.

Questa restrizione sarà temporaneamente disabilitata durante il colloquio orale, al fine di evitare ambiguità sul concetto di “registrazione”.

Registrarsi infatti non significa aver accesso ai file, bensì avere accesso alla piattaforma per l’invio dei file agli autori dei documenti che vengono pubblicati sul mio sito.

Sul sito principale è possibile:

- Consultare appunti su tutti gli argomenti svolti di tutte le materie (fatta eccezione per GPOI laboratorio, IRC e scienze motorie);
- Consultare le dispense e scaricare i programmi che i professori mettono a disposizione;
- Guardare le lezioni “peer to peer” svolte con la classe (funzione particolarmente utile durante la didattica a distanza);
- Guardare tutorial ed ascoltare podcast (nel tempo, questi contenuti sono stati tutti sostituiti dai peer to peer);
- Visionare le circolari di interesse per la classe ed il calendario con le interrogazioni e le verifiche in programma.

The screenshot displays the main interface of the Andrea Bellani online notes submission system. At the top, there's a navigation bar with links for 'Appunti', 'Spiegazioni', 'Comunicazioni', 'Programmi', 'Registrati', and 'Accedi'. Below the navigation, there are several sections representing different courses or topics:

- Il livello application**: Shows a thumbnail of a presentation slide titled 'Andrea Bellani | FI - 09/2019'.
- Come funzionano**: Shows a thumbnail of a presentation slide titled 'Come funzionano'.
- Le reti wireless (introduzione)**: Shows a thumbnail of a presentation slide titled 'Le reti wireless (introduzione)'.
- Il routing**: Shows a thumbnail of a presentation slide titled 'Il routing'.
- La crittografia**: Shows a thumbnail of a presentation slide titled 'La crittografia'.
- Le reti wireless**: Shows a thumbnail of a presentation slide titled 'Le reti wireless'.
- Reti IP e reti cellulari**: Shows a thumbnail of a presentation slide titled 'Reti IP e reti cellulari'.
- Le VPN**: Shows a thumbnail of a presentation slide titled 'Le VPN'.
- Il DHCP**: Shows a thumbnail of a presentation slide titled 'Il DHCP'.
- La gestione delle reti**: Shows a thumbnail of a presentation slide titled 'La gestione delle reti'.
- Il calcolo dei volumi**: Shows a thumbnail of a presentation slide titled 'Il calcolo dei volumi'.
- Gli integrali impropri**: Shows a thumbnail of a presentation slide titled 'Gli integrali impropri'.
- Informatica**: A large section containing three sub-thumbs:
 - La traduzione dello schema ER**: Shows a screenshot of a database diagram and its English translation.
 - Ristrutturazione e traduzione dello schema ER**: Shows a screenshot of a database diagram and its restructuring and translation.
 - Ristrutturazione e traduzione dello schema ER (Esercizio 2)**: Shows a screenshot of a database diagram and its restructuring and translation for exercise 2.
 - Ristrutturazione e traduzione dello schema ER (Esercizio 3)**: Shows a screenshot of a database diagram and its restructuring and translation for exercise 3.

3.2.2 Google Sites

Google Sites è un CMS (Content Management System), ovvero una piattaforma per lo sviluppo di siti web di varia natura.

Tramite un'interfaccia grafica WYSIWYG (What You See Is What You Get), chiunque può creare un sito professionale e sovente configurare sul proprio sito diverse funzionalità aggiuntive anche molto complesse da programmare.

La maggior parte dei CMS permette all'utente anche di:

- Installare plug-in sul proprio sito: i plug-in sono in genere degli script inseribili in una determinata zona o pagina e permettono il supporto di funzionalità aggiuntive (come le traduzioni o una piccola base di dati per la memorizzazione dei dati acquisiti da un form);
- Pubblicare il proprio sito ed aggiungere il proprio URL personalizzato;
- Aggiungere il proprio codice HTML alle pagine.

Ciò che a noi interessa sono gli ultimi due punti.

Per quanto riguarda la pubblicazione online, tratteremo approfonditamente l'argomento nella sezione dedicata, mentre l'ultimo punto è ciò che dobbiamo realizzare in questa fase del progetto.

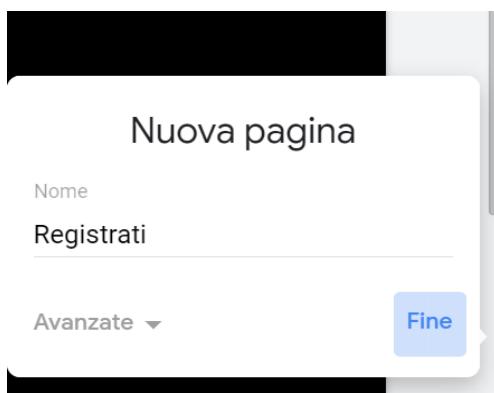
Prima di iniziare, però, vorrei chiarire la scelta di Google Sites.

Durante l'anno, ho provato diversi CMS (alcuni più approfonditamente altri meno), per cercare quali fossero i migliori e che, allo stesso tempo, permettessero l'aggiunta del proprio dominio personalizzato gratuitamente. L'unico tra questi che lo permetteva fu proprio Google Sites (normalmente l'aggiunta del proprio dominio personalizzato è disponibile fino a quando l'utente è abbonato al piano pro di circa 9-10 euro/dollari mensili; una volta disdetto il piano, il dominio personalizzato viene automaticamente rimosso).

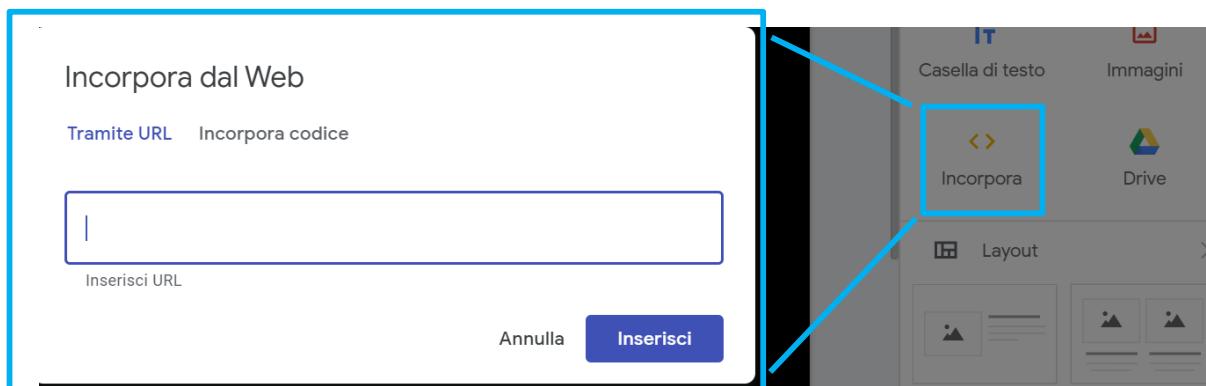
Google Sites è un CMS estremamente semplice e standard in tutte le funzionalità che mette a disposizione.

I plug-in quasi non esistono, essendo possibile solo incorporare dello script e le pagine non sono responsive (ovvero non si adattano alle dimensione del dispositivo che le visualizza).

Procediamo ora con la creazione della pagina che ospiterà il form per la registrazione:



Aggiungiamo l'oggetto grafico per incorporare il codice HTML:



Scriviamo il codice HTML per la creazione del form:



3.2.3 Il form per la registrazione

Di seguito, lo script commentato del form della seguente [pagina](#):

```

<br>
<!--
una volta cliccato il pulsante di invio, il form invia i dati inseriti ad
"https://account.andreabellani.com/andreabellani/registrazione.php"
con il metodo HTTP "POST".
Preferiamo utilizzare "POST" perché stiamo inserendo dei dati sensibili,
l'alternativa sarebbe l'utilizzo del metodo di default "GET", che utilizza l'URL
per il passaggio dei parametri.
-->
<form action="https://account.andreabellani.com/andreabellani/registrazione.php" method="post">
    Il tuo username:
    <input type="text" name="username">
    La tua password:
    <input type="password" name="password0">
    Reinserisci la password
    <input type="password" name="password">
    <input type="submit">
</form>

```

3.2.4 Il form per il login

Il form per il login, accessibile da [qui](#), è realizzato con gli stessi tag del form per la registrazione.

La pagina invocata sarà quella per l'autenticazione di un utente già registrato ed è presente un solo campo per l'inserimento della password:

```
<form action="https://account.andreabellani.com/andreabellani/login.php" method="post">
    Il tuo username:
    <input type="text" name="username">
    La tua password:
    <input type="password" name="password">
    <input type="submit">
</form>
```

3.3 L'autenticazione

Ora vediamo le prime due pagine PHP per l'autenticazione/registrazione dell'utente.



3.3.1 Le funzioni principali

Prima di illustrare il codice specifichiamo le seguenti funzioni (che saranno utilizzate anche in quasi tutte le altre pagine):

Nome	Linguaggio	Prototipo	Scopo
session_start()	PHP	bool session_start([array]): <ul style="list-style-type: none"> • True: la sessione è stata creata od aperta. • False: non è stato possibile creare od aprire la sessione 	Crea una sessione PHP oppure ne apre una già creata. Va messa sempre come prima istruzione di tutto il codice PHP.
session_unset()	PHP	bool session_start(void): <ul style="list-style-type: none"> • True: le variabili della sessione sono state deallocate; • False: non è stato possibile deallocare le variabili della sessione. 	Dealloca tutte le variabili superglobali memorizzate nella sessione (svuota l'array \$_SESSION[...]).
session_destroy()	PHP	bool session_start(void): <ul style="list-style-type: none"> • True: la sessione è stata 	Elimina la sessione PHP corrente.

		<p>eliminata;</p> <ul style="list-style-type: none"> • False: non è stato possibile eliminare la sessione. 	
print()	PHP	<pre>int print(string STRINGA1)</pre> <ul style="list-style-type: none"> • Valore di ritorno: sempre 1. 	Data una stringa, ne stampa a video il contenuto (se la stringa sono tag HTML, i tag verranno eseguiti dal browser).
alert()	JavaScript	<pre>void alert (string STRINGA)</pre>	Stampa in un pop up il valore di "STRINGA".
window.close()	JavaScript	<pre>void window.close()</pre>	Chiude la finestra corrente.
window.open()	JavaScript	<pre>* window.open(string URL, string NAME, string SPECS, string REPLACE):</pre> <ul style="list-style-type: none"> • URL: URL della pagina da aprire; • NAME: opzioni (noi usiamo “_self” per aprire la nuova finestra nella stessa scheda). 	Apre una nuova pagina nel browser.
mysqli_connect()	PHP	<p>Versione procedurale:</p> <pre>object mysqli_connect(string host, string username, string password, string dbname, string port)</pre> <ul style="list-style-type: none"> • host: nome dell'host oppure il suo IP (noi usiamo “localhost”, essendo sulla stessa macchina del web server); • username: MySQL username per accedere al DBMS; • password: password dello username; • dbname: nome del database; • port: specifica la porta dove è in ascolto MySQL (non usato in questo 	Apre una nuova connessione con un MySQL server.

		progetto).	
mysqli_connect_errno()	PHP	<p>Versione procedurale:</p> <pre>int mysqli_connect_errno()</pre> <p>Valore di ritorno:</p> <ul style="list-style-type: none"> • Ritorna l'errore dell'ultima <code>mysqli_connect()</code> effettuata 	Sapere se la connessione con il DBMS è andata a buon fine.
mysqli_query()	PHP	<p>Versione procedurale:</p> <pre>bool/object mysqli_query(object CON, string QUERY)</pre> <ul style="list-style-type: none"> • CON: oggetto restituito da una <code>mysqli_connect()</code>; • QUERY: stringa contenente la query da inviare al database. <p>Valore di ritorno:</p> <ul style="list-style-type: none"> • True: query andata a buon fine, ma che ha restituito un insieme vuoto di valori; • False: errore nell'invio della query; • Oggetto: oggetto per accedere all'insieme di valori restituiti dalla query. 	Inviare una query (non multipla) al server MySQL.
mysqli_fetch_array()	PHP	<p>Versione procedurale:</p> <pre>string[...] mysqli_fetch_array(object QUERY)</pre> <ul style="list-style-type: none"> • QUERY: oggetto restituito da una <code>mysqli_query()</code>. <p>Valore di ritorno:</p> <ul style="list-style-type: none"> • Array contenente i risultati che la query SQL ha restituito (ogni elemento è una cella). 	Trasformare i risultati della query in valori utilizzabili nello script PHP.
rand()	PHP	<pre>int rand (int MIN, int MAX)</pre> <ul style="list-style-type: none"> • MIN: estremo sinistro dell'intervallo chiuso dove sarà estratto il numero; • MAX: estremo destro 	Dati 2 estremi, genera un numero pseudo-casuale “n”, tale che: $MIN \leq n \leq MAX$.

		<p>dell'intervallo chiuso dove sarà estratto il numero.</p> <p>Valore di ritorno:</p> <ul style="list-style-type: none"> • Intero pseudo-casuale generato. 	
--	--	---	--

3.3.2 Le variabili superglobali utilizzate

Facciamo anche largo utilizzo delle seguenti variabili PHP superglobali:

Nome array	Linguaggio	Valori memorizzati
<code>\$_POST['...']</code>	PHP	Array superglobale al cui interno sono memorizzati i valori dei campi inseriti nel form che ha contattato la pagina PHP
<code>\$_SESSION ['...']</code>	PHP	Array superglobale al cui interno sono memorizzate le variabili “di sessione”. Questa tecnica ci permette di memorizzare variabili visibili e modificabili da qualsiasi pagina

3.3.3 Registrazione.php

Il flow chart dell'algoritmo usato dalla pagina PHP è troppo esteso per essere inserito nel documento: è accessibile [qui](#).

Ogni volta che si dovrà cambiare pagina oppure chiudere quella corrente, si utilizzerà sempre:

1. `Session_unset()`: dealloca le variabili superglobali presenti in `$_SESSION[]` (negli script per l'autenticazione questo comando viene omesso dato che la sessione è iniziata ma non sono ancora state allocate variabili in essa);
2. `session_destroy()`: termina la sessione;
3. `print()`: scrive sullo schermo un testo (che, se HTML, sarà tradotto in quello che deve rappresentare);
4. tag HTML: per includere lo script Javascript;
5. `alert()`: per inviare il messaggio all'utente tramite una finestra di dialogo;
6. `window.close()` o `window.open()`.

```
session_destroy();
print("
<script>
    alert('le password non corrispondono, reinserirle uguali per continuare');
    window.close();
</script>
");
```

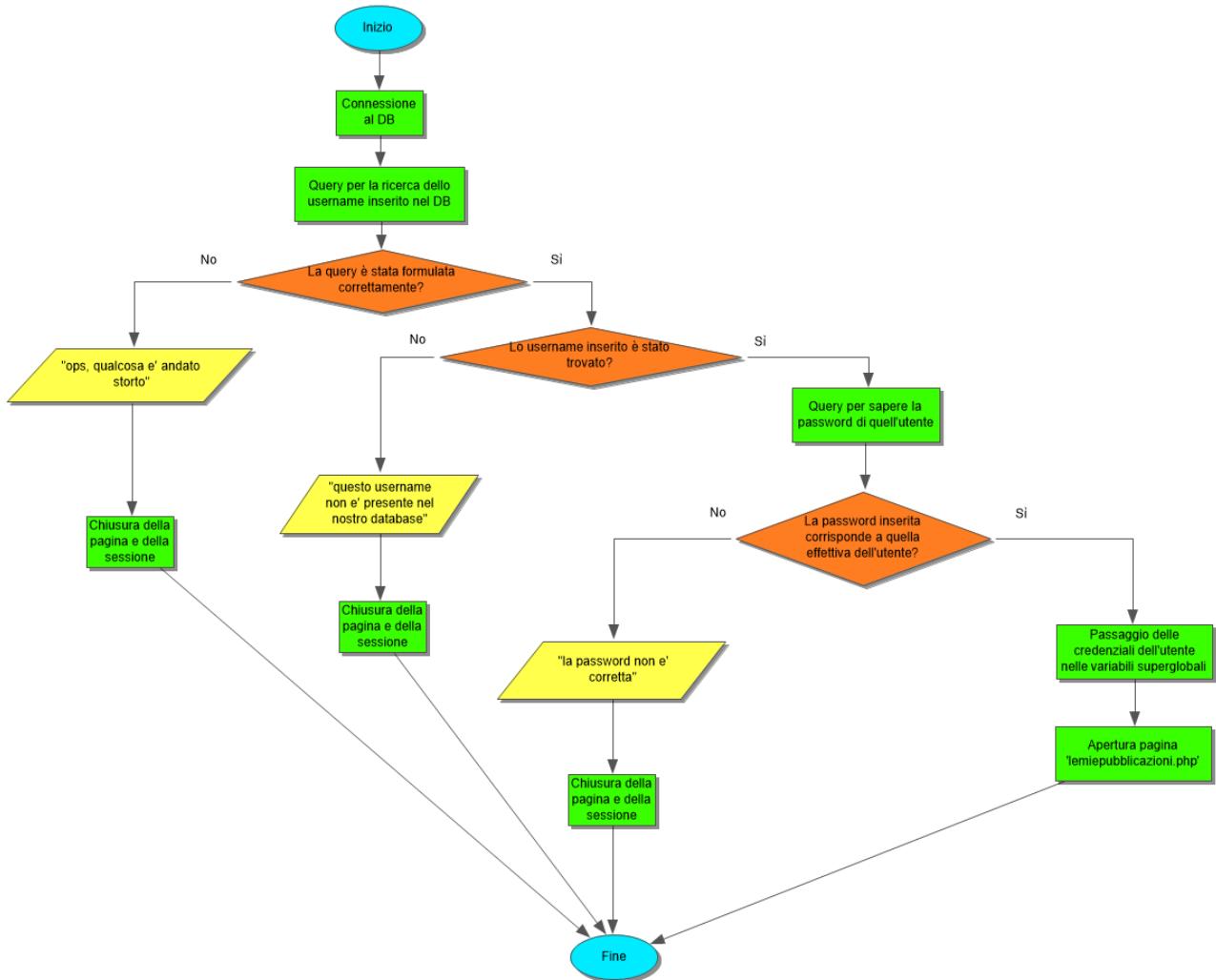
Se la registrazione va a buon fine dobbiamo salvare le credenziali dell'utente nelle variabili superglobali e aprire la nuova pagina:

```

print("
    <script>
        alert('la registrazione e\' andata a buon fine');
        window.open('".$host."/lemiepubblicazioni.php');
    </script>
");
$_SESSION['username'] = $_POST['username'];
$_SESSION['password'] = $_POST['password'];
$_SESSION['id'] = $random;

```

3.3.4 Login.php

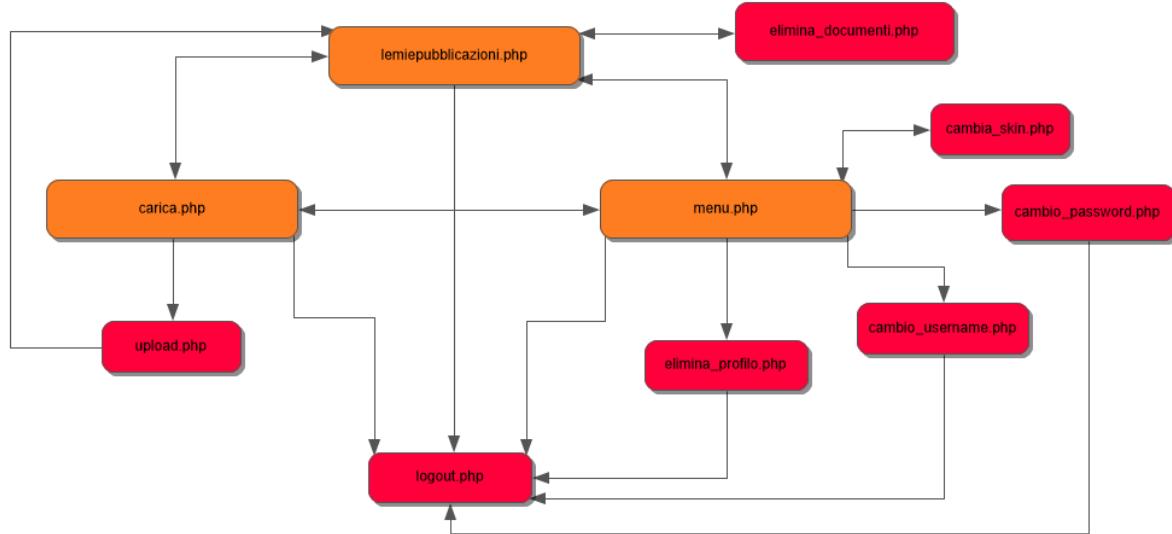


Per quanto riguarda lo script, non sono presenti istruzioni differenti da "registrazione.php".

3.4 Le pagine del sito

Vediamo ora le pagine “vere e proprie”:

1. Lemiepubblicazioni
 - a. Elimina_documenti
2. Carica
 - a. Upload
3. Menu
 - a. Cambio_username
 - b. Cambio_password
 - c. Elimina_profilo
 - d. Cambio_skin



3.4.1 Lemiepubblicazioni.php

Questa è la pagina principale del sito ed è perciò quella che viene aperta subito dopo la registrazione od il login.

Questa pagina mette a disposizione le funzionalità di:

- Visualizzazione dei documenti caricati dall’utente (topic, nome, giudizio e status);
- Download dei file caricati;
- Eliminazione dei file caricati (tramite “elimina_documenti.php”);
- Logout (tramite “logout.php”).

Decidiamo di mostrare prima tutte le funzionalità e poi la loro implementazione.

3.4.1.1 L'interfaccia con l'utente

The screenshot shows a user interface for managing documents. At the top right is a 'Log out' button. Below it is a row of three icons: a green eye (View), a purple download (Download), and a red trash can (Delete). A blue arrow points from the text 'Pulsante per il download' to the download icon. Another blue arrow points from 'Pulsante per eliminare i file selezionati' to the trash can icon. A third blue arrow points from 'Checkbox per selezionare il file' to the checkbox in the bottom right corner of the document list. Labels with arrows point to specific elements: 'Topic del documento' to the first column, 'Nome del documento' to the second column, 'Giudizio' to the third column, and 'Status' to the fourth column.

Topic del documento	Nome del documento	Giudizio	Status
Eugenio Montale	Montale.pptx	L'idea c'è, ma dovresti indicare le fonti ed approfondire i momenti salienti della vita	<input type="checkbox"/>
DNS	Appunti DNS.pdf	Mi dispiace, non siamo interessati	<input type="checkbox"/>
DNS	DNS - Seconda prova.pdf		<input type="checkbox"/>
Giuseppe Ungaretti	Ungaretti-Dокументo.pdf	Questo documento è stato approvato	<input checked="" type="checkbox"/>

Carica un tuo documento!

Lo status:

- Status = 2 : il documento può interessarci ma va migliorato;
- Status = 0 : il documento non ci interessa;
- Status = 1 : il documento non è ancora stato guardato;
- Status = 3 : il documento è stato approvato.

Questa feature è molto utile: chiunque voglia realizzare un applicativo col nostro DB può stabilire che valore numerico dare ai vari stati e cosa fare in base al loro valore (indipendenza tra base di dati ed applicativo che la utilizza).

Per eliminare i documenti basta spuntare le checkbox dei documenti che vogliamo eliminare e poi cliccare sul pulsante con l'icona del cestino.

The screenshot shows a user interface for managing documents. At the top right is a 'Log out' button. Below it is a row of three icons: a green eye (View), a purple download (Download), and a red trash can (Delete). A blue arrow points from the text 'Pulsante per il download' to the download icon. Another blue arrow points from 'Pulsante per eliminare i file selezionati' to the trash can icon. A third blue arrow points from 'Checkbox per selezionare il file' to the checkbox in the bottom right corner of the document list. A tooltip 'Cliccami per eliminarli!' is shown over the trash can icon. Labels with arrows point to specific elements: 'Topic del documento' to the first column, 'Nome del documento' to the second column, 'Giudizio' to the third column, and 'Status' to the fourth column.

Topic del documento	Nome del documento	Giudizio	Status
Eugenio Montale	Montale.pptx	L'idea c'è, ma dovresti indicare le fonti ed approfondire i momenti salienti della vita	<input type="checkbox"/>
DNS	Appunti DNS.pdf	Mi dispiace, non siamo interessati	<input type="checkbox"/>
DNS	DNS - Seconda prova.pdf		<input type="checkbox"/>
Giuseppe Ungaretti	Ungaretti-Dокументo.pdf	Questo documento è stato approvato	<input checked="" type="checkbox"/>

Cliccami per eliminarli!

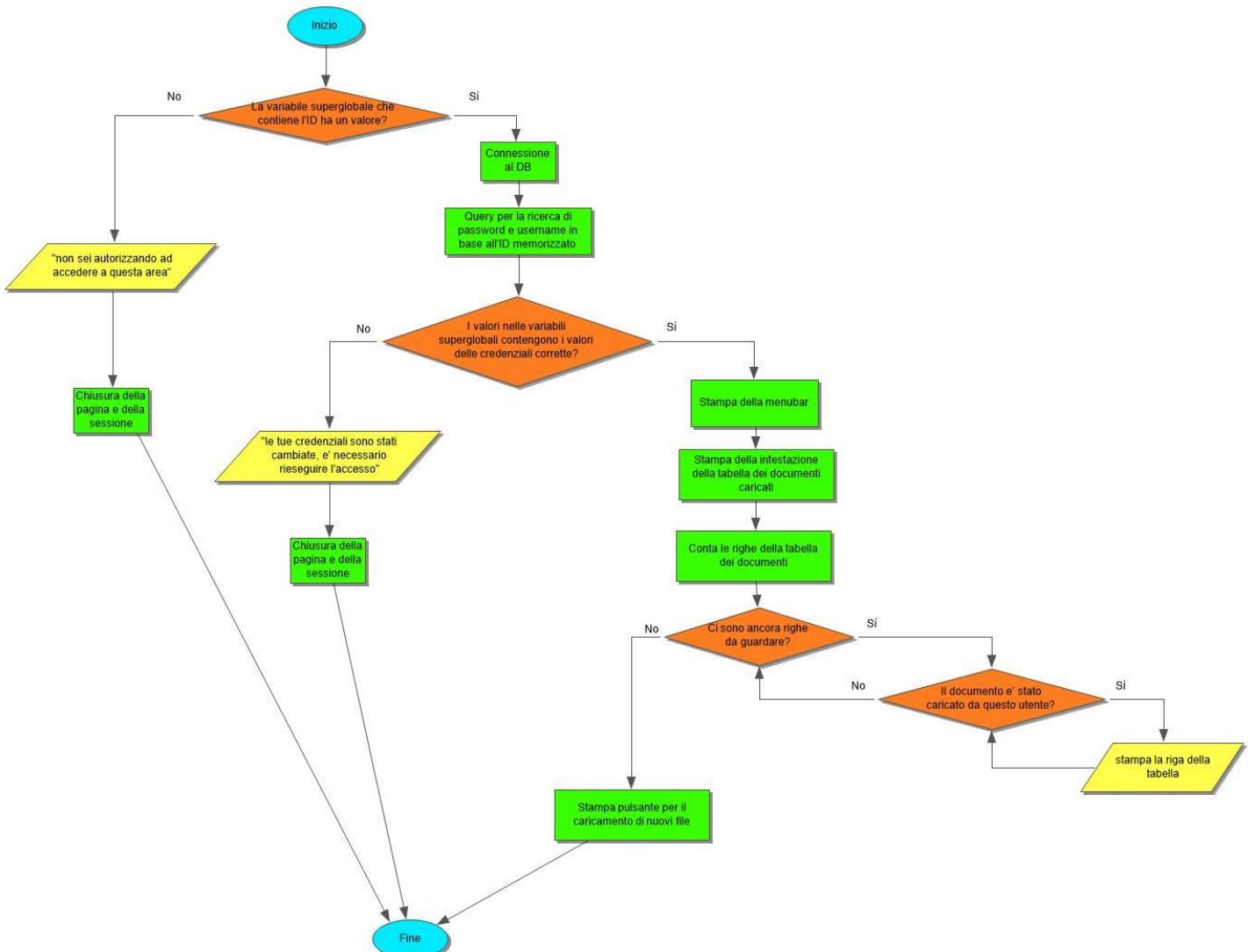
3.4.1.2 Lo script

Questa pagina è la più complessa di tutto il sito, ma l'algoritmo possiede una complessità ciclomatica (possibili scelte) molto bassa.

Durante i test, ci siamo resi conto che, se un utente avesse aperto contemporaneamente due schede ed in una di esse avesse cambiato le proprie credenziali, sull'altra scheda avrebbe potuto usare le credenziali vecchie per confermare un nuovo cambio di credenziali od addirittura l'eliminazione dell'account.

Da quel momento, ogni pagina controlla sempre se: prima di tutto nelle variabili superglobali sono presenti dei valori (così da negare l'accesso a chi semplicemente scrive un URL nella barra degli indirizzi senza aver eseguito l'accesso); in secondo luogo, controlla se i valori presenti nelle variabili superglobali corrispondono a quelli realmente presenti nella entry della tabella che possiede il dato ID.

Il flow chart dell'algoritmo è il seguente:



3.4.1.3 La stampa della tabella

Le operazioni precedenti non sono particolarmente interessanti, possono essere trovate commentate qui.

La stampa della tabella avviene tramite una tecnica che prevede di scrivere HTML in caso in cui vengano soddisfatte le condizioni previste da dei costrutti scritti in PHP.

La stampa della tabella è quindi automatica ogni qual volta la condizione nel ciclo viene rispettata:

```

File Modifica Formato Visualizza ?
$n_righe=mysqli_fetch_array(mysqli_query($con, "SELECT COUNT(*) FROM documento;")); //conteggio righe
//creazione della tabella
for ($i=0; $i<$n_righe[0]; $i++) //per ogni riga
{
    $risultato = mysqli_fetch_array(mysqli_query($con, "SELECT * FROM documento LIMIT ".$i.", 1;")); //seleziona tutti i campi della riga "i"
    $token = strtok($risultato[1], "$"); //estrai il suo nome
    $token = strtok("$"); //rimuovi il carattere separatore "$" (così da non stampare anche l'ID)
    if ($risultato[4]==$id[0]) //se l'ID dell'utente corrisponde con quello memorizzato nell'attributo col vincolo di foreign key
    {
        /**
         * in base allo status del documento, imposta una faccina,
         * un colore ed un tooltip al passaggio del mouse sopra la faccina
        */
        switch ($risultato[3])
        {
            case 0:
                $icon = "far fa-frown-o";
                $color = "crimson";
            ...
        }
    }
}

print("
<tr style='border: 2px solid black; background-color: ".$color."'> <!-- stampa la riga del colore in base allo status -->
<th class='cells'>".$risultato[5]."</th> <!-- stampa il topic del documento -->
<th class='cells' style='text-align: right'>".$token."</th> <!-- stampa il nome senza il suo ID -->
<th class='cells'>".$risultato[2]."</th> <!-- stampa il giudizio -->
<th class='cells' title='".$tooltip."'><i class='".$icon."'></i></th> <!-- stampa la faccina in base allo status -->
<th class='cells' style='text-align: center; background-color: blueviolet'> <!-- stampa la cella per il download -->
    <a href='".$host."/Documenti/".$risultato[1]."'> <!-- link al file sul server -->
        <i class='fa fa-download'></i>
    </a>
</th>
<th class='cells' style='background-color: red'> <!-- stampa la checkbox per l'eliminazione -->
    <input name='".$risultato[0]."' type='checkbox'> <!-- la checkbox avrà come nome l'ID del documento -->
</th>
</tr>
");

```

La lettura/download dei file non quindi altro che un link al file sulla directory del server.

Funzioni particolari utilizzate:

Nome	Linguaggio	Prototipo	Scopo
strtok()	PHP	<p>Overloading:</p> <ul style="list-style-type: none"> string strtok(string STRINGA, string TOKEN): data una stringa “STRINGA”, la divide in parti per ogni stringa “TOKEN” trovata; string strtok(string TOKEN): restituisce la parte di STRINGA preceduta da TOKEN (ogni volta che viene chiamata, passa al token successivo). 	Serve a segmentare una stringa in token ed a selezionare i token generati.

Nota bene “il problema dell’accesso ai file”:

Il download dei file, quindi, non è altro che un link al file all’interno del web server. Questo dunque significa che potenzialmente chiunque abbia il link di un file può scaricarlo. A livello di sicurezza questo rappresenterebbe un grave errore, tuttavia abbiamo deciso di lasciare questa tecnica per non allungare ulteriormente l’elaborato e perché abbiamo ritenuto fosse una modifica troppo slegata da qualsiasi argomento affrontato durante l’anno.

3.4.1.4 L’eliminazione dei documenti

Tramite il pulsante con l’icona del cestino è possibile eliminare i documenti selezionati.

Il collegamento alla pagina PHP “elimina_documenti.php” è realizzato mediante un form che incorpora tutta la tabella e che ha il seguente tag di apertura nella pagina “lemiepubblicazioni.php”:

```
<form action='".$host."/elimina_documenti.php' method='post'>
<th>
    <div class='delete_buttons' id='download' title='download'>
        <i class='fa fa-download'></i>
    </div>
</th>
<th>
    <button type='submit' class='delete_buttons' id='delete' title='Cliccami per eliminarli!'>
        <i class='fa fa-trash'></i>
    </button>
</th>
```

Il seguente tag form viene chiuso a tabella completamente stampata.

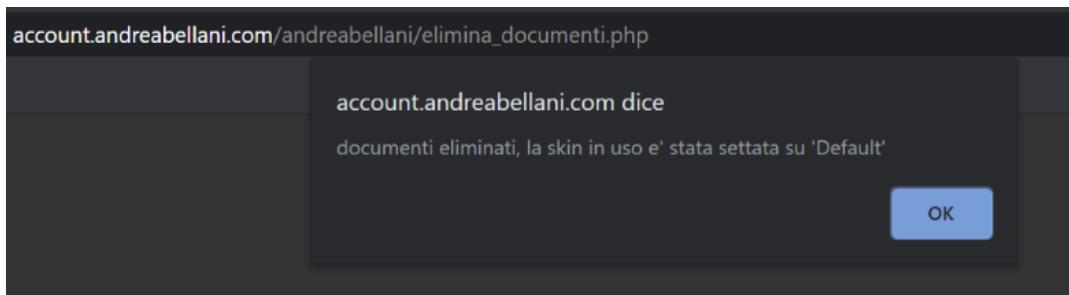
Il “trucco” sta nell’includere anche in questo tag “gigantesco” un bottone solo, ovvero quello con l’icona del cestino e tutte le checkbox dei documenti.

Preferiamo non mostrare il flow chart questa volta, vista la semplicità dello script, ma mostrare direttamente il codice commentato:

```

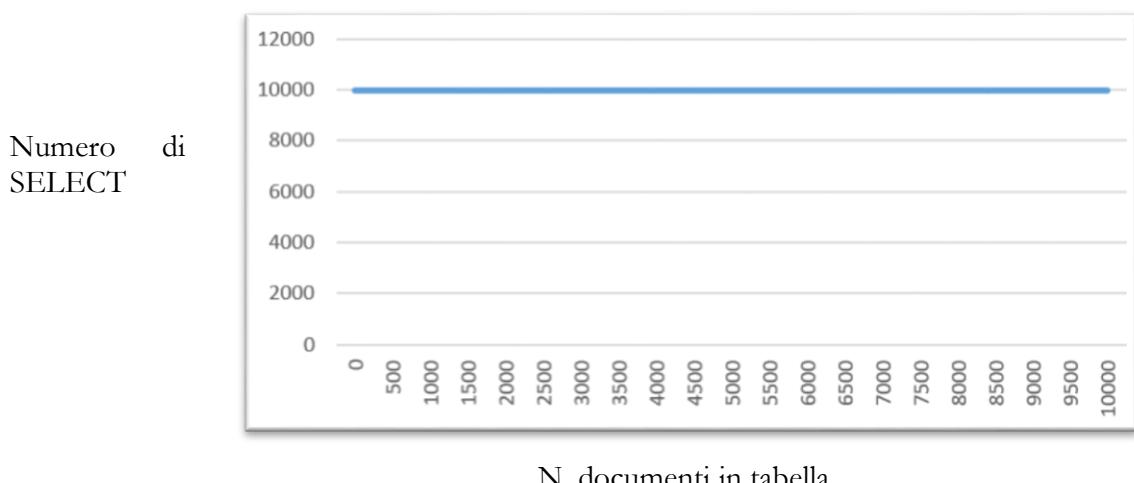
for ($i=0; $i<$NUM_MAX_DOCUMENTI_TOTALI; $i++) //si scorrono tutti i possibili valori per l'ID
{
    if (isset($_POST[$i])) //se è stata selezionata un campo nel form precedente con un nome uguale ad "i"
    {
        $query = "SELECT Nome FROM documento WHERE ID='".$i."'"; //formula la query per ottenerne il nome
        $files = mysqli_fetch_array(mysqli_query($con, $query)); //invia la query
        //elimina il documento dalla cartella
        unlink("C:\\\\Users\\\\Andrea\\\\Desktop\\\\Andrea\\\\Scuola\\\\Informatica\\\\Laboratorio\\\\PHP\\\\USBWebserver v8.5\\\\8.5\\\\root
        mysqli_query($con, "DELETE FROM documento WHERE ID='".$i."'"); //elimina la entry dalla tabella
    }
}
//setta la skin a quella di default (evitare che un utente possa tenersi una skin premium anche quando
//ha meno di tot. documenti caricati)
mysqli_query($con, "UPDATE utente SET FKNomeSkin='Default' WHERE Username='".$username."");
//comunica all'utente quanto è accaduto
print("
<script>
    alert('documenti eliminati, la skin in uso e\' stata settata su \'Default\'');
    window.open(\"$host./lempubblicazioni.php\", '_self');
</script>
");

```

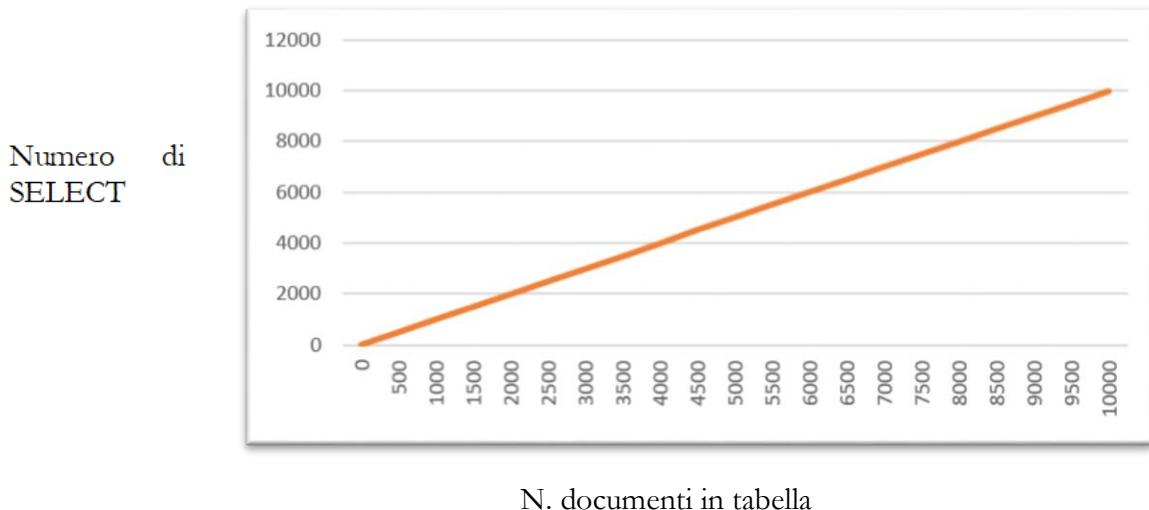


Nota bene “considerazioni sull’efficienza”:

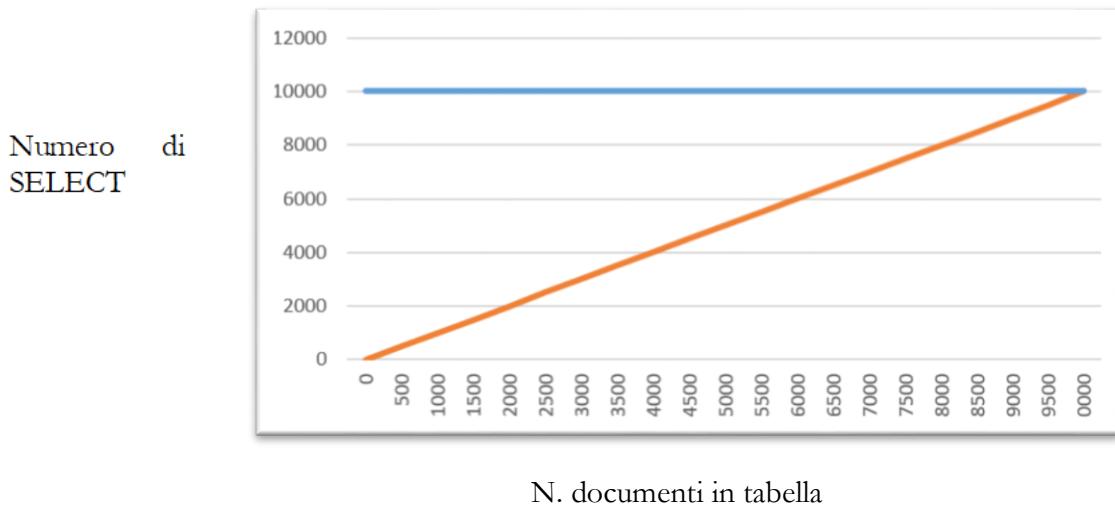
Si potrebbe obiettare che l’algoritmo non è abbastanza efficiente, infatti, al crescere del numero di documenti, l’algoritmo eseguirebbe sempre il massimo delle operazioni (complessità costante, in base al numero massimo di documenti):



Un algoritmo che conti prima il numero di documenti attuali e poi controlli per ogni riga se è presente una checkbox selezionata con nome uguale a quell'ID, avrebbe una complessità minore (lineare), dato che eseguirebbe un numero di SELECT proporzionale al numero di documenti (più una iniziale per il conteggio delle righe):



Confrontandole:



Come mai allora non utilizziamo la seconda versione?

- 10000 query sembrano tante a dirsi, ma in realtà sono eseguibili in pochi decimi di secondo dal nostro hardware.
- Per selezionare una sola riga, dovremmo utilizzare una query del tipo “SELECT Nome FROM documento LIMIT “.\$i.”, 1;”, che non sempre è andata a buon fine. Abbiamo perciò preferito, al fine di evitare spiacevoli inconvenienti, di utilizzare la prima versione.

3.4.2 Carica.php

Veniamo ora al “cuore” di tutto il sito: la pagina per l’upload dei file.

Anche in questo caso non ci troviamo davanti ad algoritmi complessi, addirittura in questo caso, oltre alla menubar, è presente un semplice form dove scegliere il topic e caricare il file.

3.4.2.1 Lo script

```
<!-- form per l'upload-->
<form method='post' action='".$host."/upload.php' method='post' enctype='multipart/form-data' class='menu_form'>
<br>
Scegli il topic su cui vuoi caricare il file:
<br>
<br>
<select name='topic' required>
<br>
<option disabled selected value> seleziona il topic </option>); //prima voce senza nessun topic
$query = "SELECT COUNT(*) FROM topic;"; //numero di topic
$n_topic=mysqli_fetch_array(mysqli_query($con, $query));
for ($i=0; $i<$n_topic[0]; $i++) //fino a quando ci sono topic da inserire nel menù a tendina
{
    $query = "SELECT * FROM topic LIMIT ".$i.", 1"; //inseriscilo
    $topic=mysqli_fetch_array(mysqli_query($con, $query));
    print
    (
        "
            <option value='".$topic[0]."'> <!-- agli il valore del nome del topic -->
            ".$topic[0]." | ".$topic[1]." | ".$topic[2]." <!-- scrivi nome, formato e descrizione -->
        </option>
    ");
}
print
(
"
</select>
<br>
<br>
<input type='file' name='file1' class='file_sub'> <!-- inserisci l'oggetto per caricare il file -->
<br>
<br>
<button type='submit' name='submit' class='generic_buttons'> <!-- bottone per l'invio -->
    <i class='fas fa-upload'></i>
    Carica!
</button>
```

3.4.2.2 Upload.php

Questa pagina si occuperà di:

- Controllare la correttezza dei campi inseriti nel form di “carica.php”;
 - Inviare il file una volta che questo è conforme alle caratteristiche richieste;
 - Creare una nuova entry per il file nella tabella “documento”.

Le variabili:

```
$uploads_dir = 'C:\Users\Andrea\Desktop\Andrea\Scuola\Informatica\Laboratorio\PHP\USBWebserver v8.5\8.5\root\andreambellani\Documenti';
$nomeUSB = "localhost";
$userUSB = "root";
$passUSB = "Rorapandi";
$dbUSB = "account";
$NUM_MAX_USER = 1000;
$NUM_MAX_DOCUMENTI_PER_USER = 10;
$NUM_MAX_DOCUMENTI_TOTALI = $NUM_MAX_USER*$NUM_MAX_DOCUMENTI_PER_USER;
```

- \$uploads_dir: destinazione dell'upload sul web server;
- \$NUM_MAX_DOCUMENTI_PER_USER: il numero massimo di documenti che un singolo utente può caricare;
- \$NUM_MAX_DOCUMENTI_TOTALI: \$NUM_MAX_DOCUMENTI_PER_USER * \$NUM_MAX_USER.

Le parti più importanti dello script:

```

if ($_FILES["file1"]["name"]=="") //il file caricato ha nome vuoto? (ovvero: "non è stato caricato nulla?")
{
    print("
        <script type='text/javascript'>
            alert('non hai caricato nessun file!');
            window.open("'" . $host . "/carica.php', '_self');
        </script>
    ");
}
else
{
    $query = "SELECT FormatoRichiesto FROM topic WHERE Nome='".$_POST["topic"].""; //query per sapere quale sia il formato richiesto
$formato_richiesto=mysqli_fetch_array(mysqli_query($con, $query));
if (pathinfo($_FILES["file1"]["name"], PATHINFO_EXTENSION) == $formato_richiesto[0]) //il formato corrisponde
{
    //conteggio del numero di documenti caricati dall'utente
    $query = "SELECT COUNT(*) FROM documento WHERE FKIDUtente='".$id."'";
    $actual_doc_num=mysqli_fetch_array(mysqli_query($con, $query)); //invio della query
    if ($actual_doc_num[0] >= $NUM_MAX_DOCUMENTI_PER_USER) //l'utente ha raggiunto il limite di file caricati?
    {
        print("
            <script type='text/javascript'>
                alert('ci dispiace ma hai raggiunto il limite di upload');
                window.open("'" . $host . "/lemiepubblicazioni.php', '_self');
            </script>
        ");
    }
}

if (strpos($_FILES["file1"]["name"], "$") !== false) //il file contiene il carattere speciale "$"?
{
    print("
        <script>
            alert('\'$\' non e\' un carattere ammissibile nel titolo del file');
            window.open("'" . $host . "/carica.php', '_self');
        </script>
    ");
}
else
{
    do //generazione del numero random per l'ID
    {
        $random = rand(1, $NUM_MAX_DOCUMENTI_TOTALI);
        $num=mysqli_fetch_array(mysqli_query($con, "SELECT ID FROM documento WHERE ID=".$random.""));
    }
    while($num[0]=="");

    $pname = $random."".$_FILES["file1"]["name"]; //in "$pname" viene memorizzato il nuovo nome del file [ID]+[$]+[NOME ORIGINALE]
    //invio il file alla cartella e, controllo se l'upload è andato a buon fine
    if (move_uploaded_file($_FILES["file1"]["tmp_name"], $uploads_dir.'/'.$pname))
    {
        //aggiungi la nuova entry a 'documento'
        $query = "INSERT INTO documento VALUES (".$random.", '".$pname."', 'Non abbiamo ancora controllato questo file', 1, ".$id.", mysqli_query($con, $query));
        print("
            <script>
                alert('il file e\' stato caricato');
                window.open("'" . $host . "/lemiepubblicazioni.php', '_self');
            </script>
        ");
    }
}

```

Variabili superglobali:

Nome	Linguaggio	Utilità	Modalità di utilizzo
<code>\$_FILES[...][...]</code>	PHP	Memorizza i file che sono stati caricati con il metodo POST.	Nella prima parentesi quadra mettiamo il nome del file, nella seconda il tipo di informazione che vogliamo su quel file, ad esempio: <ul style="list-style-type: none"> ▪ “tmp_name”: il file temporaneo che rappresenta il file caricato; ▪ “name”: il nome del file.

Le funzioni:

Nome	Linguaggio	Prototipo	Scopo
<code>strpos()</code>	PHP	<code>strpos(STRINGA1, STRINGA DA TROVARE, PUNTO DI INIZIO RICERCA (opzionale)) : int, boolean</code> <ul style="list-style-type: none"> ▪ Int: se il carattere viene trovato restituisce la posizione della prima occorrenza della seconda stringa nella prima ▪ Boolean: se non è stato trovato un carattere, restituisce false. 	Serve a cercare un stringa in un'altra.
<code>move_uploaded_file()</code>	PHP	<code>move_uploaded_file(file temporaneo, destinazione) : boolean</code> <ul style="list-style-type: none"> ▪ True: l'upload è andato a buon fine. ▪ False: per qualche motivo, non è stato possibile caricare il file nella cartella 	Dato un file caricato col metodo POST, lo carica sulla cartella specificata dal secondo parametro.
<code>pathinfo()</code>	PHP	<code>pathinfo(stringa, opzione) : vario, in base alle opzioni</code> Opzioni possibili: <ul style="list-style-type: none"> ▪ PATHINFO_DIRNAME. ▪ PATHINFO_BASENAME. ▪ PATHINFO_EXTENSION: ritorna l'estensione del file ▪ PATHINFO_FILENAME. 	Conoscere informazioni su un file (ad esempio la sua estensione).

3.4.3 Menu.php

Questa è l'ultima delle pagine principali e contiene tutti i form per il cambio delle credenziali, della skin e l'eliminazione del profilo:

The screenshot shows a dark-themed web interface with a top navigation bar containing four buttons: "Le mie pubblicazioni" (green), "Carica un tuo documento!" (green), "Elimina profilo" (red), and "Log out" (red). Below the navigation, there are four input fields arranged in pairs. The first pair consists of "Il tuo vecchio username:" (old username) and "Il tuo nuovo username:". The second pair consists of "La tua vecchia password:" (old password) and "La tua nuova password:". Below these are two more input fields: "Reinserisci il nuovo username:" and "Reinserisci la nuova password:". Each input field has a corresponding green "Cambia" button below it. At the bottom center is a dropdown menu labeled "Scegli la tua skin:" with options "Default | documenti richiesti: 0 | Skin base" and a "Cambia" button.

Dopo aver stampato, come di consueto, la menubar, vengono stampati i 2 form per il cambio delle credenziali e il menù a tendina per la scelta delle skin.

Rispetto alle altre pagine, nella menubar è presente un pulsante per l'eliminazione del profilo.

3.4.3.1 Cambio_username.php

The screenshot shows the "Cambio_username.php" page with three input fields: "Il tuo vecchio username:", "Il tuo nuovo username:", and "Reinserisci il nuovo username:". A green "Cambia" button is at the bottom. To the right, a block of PHP code is displayed with red arrows pointing from each input field to its corresponding line in the code. The code is as follows:

```

<th colspan='2'>
    <form action='".$host."/cambio_username.php' class='menu_form' method='post'>
        <br>Il tuo vecchio username:<br>
        <input type='text' name='username0' class='actual_username'><br>
        Il tuo nuovo username:<br>
        <input type='text' name='username1' class='new_username'><br>
        Reinserisci il nuovo username:<br>
        <input type='text' name='username2' class='new_username'><br>
        <button type='submit' class='generic_buttons'>
            <i class='fa fa-refresh'></i>
            Cambia
        </button>
    </form>
</th>

```

Quando l'utente clicca il pulsante, sulla pagina “cambio_username.php” verrà gestito tutto ciò che è stato inserito nel form.

Le variabili:

```
$vecchio_username = $_POST['username0'];
$nuovo_username1 = $_POST['username1'];
$nuovo_username2 = $_POST['username2'];
```

```
if (strcmp($username, $vecchio_username)!=0) //è stato inserito lo username attuale?
{
    print("
        <script>
            alert('devi inserire il tuo username attuale!');
            window.open('".$host."/menu.php', '_self');
        </script>
    ");
}
else
{
    if (strcmp($nuovo_username1, $nuovo_username2)!=0) //i 2 username sono uguali?
    {
        print("
            <script>
                alert('l\' inserimento del nuovo username non corrisponde');
                window.open('".$host."/menu.php', '_self');
            </script>
        ");
    }
    else
    {
        if ($nuovo_username1=="") //è stato inserito un nuovo username diverso da ""
        {
            print("
                <script>
                    alert('questo username non e\' ammissibile');
                    window.open('".$host."/menu.php', '_self');
                </script>
            ");
        }
    }
}

else
{
    //ricerca del nuovo username nel database
    $query = "SELECT Username FROM utente WHERE Username='".$nuovo_username1."'";
    $is_equal=mysqli_fetch_array(mysqli_query($con, $query));
    if ($is_equal[0] != "") //se qualcuno ha già preso quello username
    {
        print("
            <script>
                alert('questo username e\' già presente');
                window.open('".$host."/menu.php', '_self');
            </script>
        ");
    }
    else
    {
        //cambio username
        $query = "UPDATE utente SET Username='".$nuovo_username1."' WHERE ID='".$id."'";
        mysqli_query($con, $query);
        print("
            <script>
                alert('lo username e\' stato cambiato, e\' necessario rieseguire l\'accesso');
                window.open('".$host."/logout.php');
                window.close();
            </script>
        ");
    }
}
```

Le funzioni:

Nome	Linguaggio	Prototipo	Scopo
strcmp()	PHP	int strcmp(STRINGA1, STRINGA2) Valore di ritorno, intero “x”: <ul style="list-style-type: none"> ▪ x=0: le 2 stringhe sono uguali; ▪ x<0: la prima precede la seconda; ▪ x>0: la seconda precede la prima. 	Confrontare due stringhe e sapere quale delle due viene prima in ordine alfabetico dell'altra.

3.4.3.2 Cambio_password.php

Questa pagina è identica a quella per il cambio dello username, se non che, al posto di controllare se lo username è già stato assegnato, controlla se la password nuova è diversa da quella vecchia:

```
if (strcmp($nuova_password1, $vecchia_password)==0) //se la password nuova è uguale a quella vecchia
{
    print("
        <script>
            alert('questa e\' già la tua password');
            window.open('".$host."/menu.php', '_self');
        </script>
    ");
}
else
{
    //cambio password
    $query = "UPDATE utente SET Password='".$nuova_password1."' WHERE ID='".$id."'";
    mysqli_query($con, $query);
    print("
        <script>
            alert('la password e\' stata cambiata, e\' necessario rieseguire l'accesso');
            window.open('".$host."/logout.php');
            window.close();
        </script>
    ");
}
```

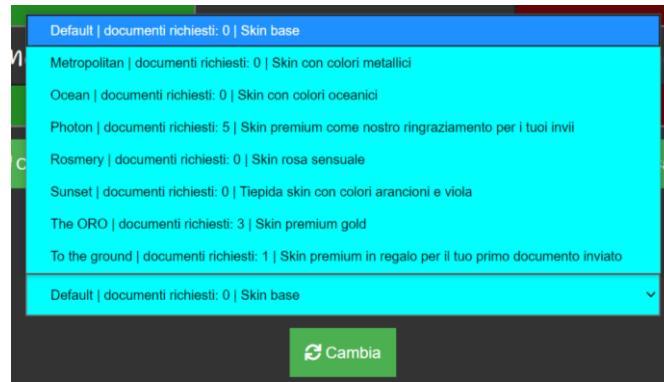
3.4.3.3 Cambia_skin.php

Le skin non sono altro che dei fogli di stile che, una volta scelti, vanno a modificare l'attributo di “utente” “FKNameSkin”.

In base al suo valore, la pagina importa un diverso file “.css”.

```
$r_color = mysqli_fetch_array(mysqli_query($con, "SELECT FKNameSkin FROM utente WHERE Username='".$username."'"));
$skin = $r_color[0];
<link rel='stylesheet' type='text/css' href='".$host."/Skin/".$skin.".css'>
```

Le skin sono selezionabili nel menu:



La scrittura del form HTML è identica a quella utilizzata per la stampa del form per la selezione del topic.

Lo script per il cambio di skin risulta quindi piuttosto banale: in sostanza, non fa altro che cambiare il valore dell'attributo “FKNomeSkin” dopo essersi assicurato che l'utente possa “permettersi” la skin:

```
//quanti documenti richiede la skin selezionata?
$n_doc_richiesti = mysqli_fetch_array(mysqli_query($con, "SELECT * FROM skin WHERE Nome='".$_POST['skin']."'"));
//quanti documenti sono associati a questo utente?
$n_doc_utente = mysqli_fetch_array(mysqli_query($con, "SELECT COUNT(*) FROM documento WHERE FKIDUtente='".$id."'"));
//l'utente può permettersi la skin?
if ($n_doc_utente[0]>=$n_doc_richiesti)
{
    //settaggio della skin
    mysqli_query($con, "UPDATE utente SET FKNomeSkin='".$_POST['skin']."' WHERE ID='".$id."'");

    print("
        <script>
            window.open('".$host."/menu.php', '_self');
        </script>
    ");
}
else
{
    print("
        <script>
            alert('non hai caricato abbastanza documenti per sbloccare '".$_POST['skin']."' ');
            window.open('".$host."/menu.php', '_self');
        </script>
    ");
}
```

Skin “Photon”:

Skin “Ocean”:

Skin “Metropolitan”:



Skin “The ORO”:



Organizzare tutta la parte del “look and feel” tramite i fogli di stile CSS, consente (anche nel caso in cui fosse presente una sola skin) di:

- Scrivere nelle pagine PHP solo il codice strettamente legato alle funzionalità della pagina;
- Utilizzare una formattazione standard per tutti i tag HTML;
- Sapere subito dove trovare le espressioni per cambiare i colori.

Qui alcuni esempi di fogli di stile per la skin (da sinistra: “To the ground”, “Sunset”, “Rosmery”):

```
.menubar
{
    background-color: #291d0c;
}
.delete_buttons
{
    background-color: #122e1f;
    border: none;
    color: white;
    padding: 15px 15px;
    text-align: center;
    display: inline-block;
    margin: 4px 2px;
    cursor: pointer;
    font-size: 16px;
}
.delete_buttons:hover
{
    background-color: #2d543f;
}
.menu_form
{
    text-align: center;
    font-family: Verdana;
    color: white;
    font-size: 20pt;
}
.actual_username
{
    width: 50%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;
    background-color: #2fde7f;
}
```

```
.menubar
{
    background-color: #550663;
}
.delete_buttons
{
    background-color: #96055a;
    border: none;
    color: white;
    padding: 15px 15px;
    text-align: center;
    display: inline-block;
    margin: 4px 2px;
    cursor: pointer;
    font-size: 16px;
}
.delete_buttons:hover
{
    background-color: #c9107d;
}
.menu_form
{
    text-align: center;
    font-family: Kristen ITC;
    color: white;
    font-size: 20pt;
}
.actual_username
{
    width: 50%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;
    background-color: #f55433;
}
```

```
.menubar
{
    background-color: #be03fc;
}
.delete_buttons
{
    background-color: #fc03df;
    border: none;
    color: white;
    padding: 15px 15px;
    text-align: center;
    display: inline-block;
    margin: 4px 2px;
    cursor: pointer;
    font-size: 16px;
}
.delete_buttons:hover
{
    background-color: #b504a0;
}
.menu_form
{
    text-align: center;
    font-family: Lucida Handwriting;
    color: white;
    font-size: 18pt;
}
.actual_username
{
    width: 50%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;
    background-color: #ed13a8;
}
```

Di contro, però, ogni oggetto deve avere un suo ID od una sua classe conforme alle definizioni dei fogli di stile.

3.4.3.4 Elimina_profilo.php

Il processo di eliminazione del profilo è composto da due fasi:

1. Autenticazione dell'utente;
2. Eliminazione di credenziali, documenti e variabili di sessione.

La prima parte è collocata in “menu.php”, in una funzione javascript chiamata quando il pulsante “elimina profilo” è stato cliccato:

```
function elimina ()
{
    if (prompt('inserisci \"ELIMINA\"') == 'ELIMINA') //l'utente deve scrivere 'ELIMINA'
    {
        if (prompt('inserisci la tua password') == ".$password.") //l'utente deve inserire la sua password
        {
            window.open('".$host."/elimina_profilo.php', '_self'); //processo di eliminazione avviato
        }
        else
        {
            alert('questa non e\' la tua password');
        }
    }
}
```

A questo punto è chiaro che l'utente vuole davvero eliminare il proprio profilo e possiamo procedere con:

1. L'eliminazione di tutti i suoi file dalla directory;
2. L'eliminazione di tutte le entry della tabella che hanno “FKIDUtente” uguale all'ID dell'utente che vuole eliminarsi;
3. Eliminazione della sua entry nella tabella utente;
4. Logout (con “logout.php”).

Questa sequenza è importante sia per i vincoli di chiave esterna (non è possibile eliminare prima un utente e poi i suoi documenti dal database), che per questioni legate all'efficienza (vediamo prima lo script):

```
for ($i=0; $i<$NUM_MAX_DOCUMENTI_TOTALI; $i++) //per ogni ID possibile di un documento
{
    //si controlla se il suo FKIDUtente è uguale all'ID dell'utente che vuole eliminarsi
    $query = "SELECT Nome FROM documento WHERE ID=\"$i.\" AND FKIDUtente=\"$id.\";";
    $nome_file = mysqli_fetch_array(mysqli_query($con, $query));
    if ($nome_file[0]!="") //se quel file corrisponde all'utente che vuole eliminarsi
    {
        unlink($uploads_dir."\\".$nome_file[0]); //eliminazione del file dalla directory
    }
}
//eliminazione di tutte le entry dei suoi documenti
$query = "DELETE FROM documento WHERE FKIDUtente=\"$id.\";";
mysqli_query($con, $query);
//eliminazione dell'utente
$query = "DELETE FROM utente WHERE ID=\"$id.\";";
mysqli_query($con, $query);
print("
<script>
    alert('la tua utenza e\' ed i tuoi file sono stati eliminati con successo, speriamo di rivederti presto');
    window.open('".$host."/logout.php');
    window.close();
</script>
");

```

Potremmo, come abbiamo fatto per “elimina_documenti.php”, formulare delle considerazioni sull’efficienza.

I motivi per cui, anche in questo caso, non abbiamo implementato un algoritmo di complessità “lineare”, sono gli stessi visti per la pagina sopracitata.

Rimuovere i documenti dalla tabella dopo la loro eliminazione dalla directory, ci consente di utilizzare un’unica istruzione per la cancellazione delle entry (raddoppieremmo il numero di query se volessimo eliminarle in contemporanea con l’eliminazione dei file dalla directory).

3.4.3.5 Logout.php

Questa pagina inizialmente non era stata scritta, le istruzioni in essa contenute erano eseguite in ogni pagina quando il pulsante per il logout veniva cliccato.

Viste le complicazioni nel momento in cui le istruzioni PHP di chiusura della sessione venivano eseguite ogni volta che la pagina veniva ricaricata, la soluzione fu chiamare queste istruzioni con una funzione javascript: “window.open()”.

Lo script è molto semplice e consiste nella chiusura della sessione:

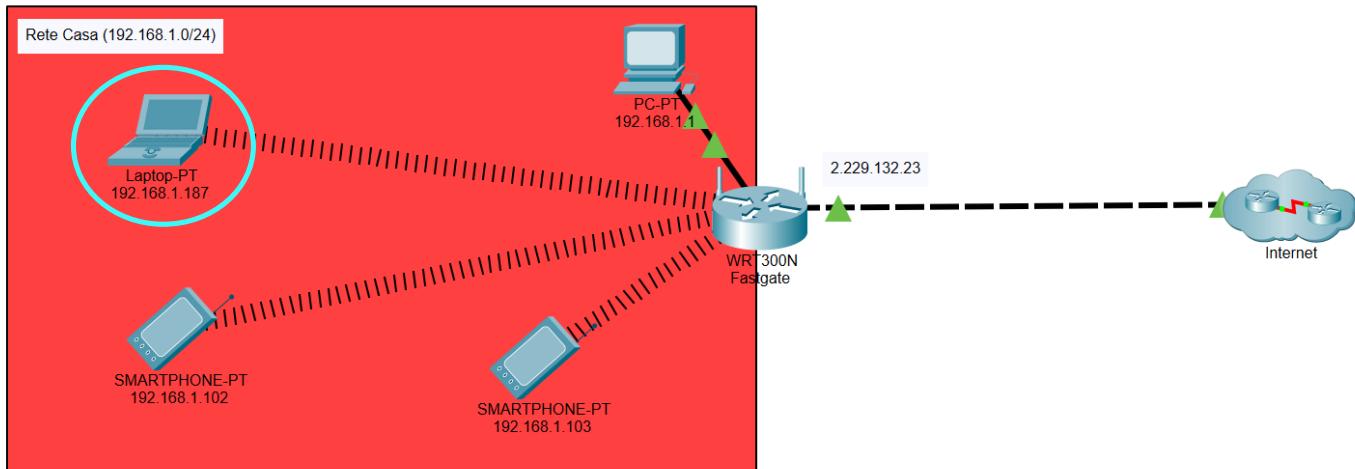
```
<?php
    session_start(); //entra nella sessione corrente
    session_unset(); //metti a null tutte le variabili della sessione
    session_destroy(); //dealloca la sessione
    print("
        <script>
            window.close();
        </script>
    ");
?>
```

4 La pubblicazione online

A questo punto, il nostro sito è realizzato e funzionante.

Dobbiamo predisporre l'occorrente per la pubblicazione online del web server e configurare il router.

Contestualizziamo l'ambiente dove si trova il nostro web server non ancora pubblicato (ovvero la nostra rete di casa):



In questo momento, il sito (escluse le pagine coi form di accesso incluse del sito realizzato con Google Sites), risiede sul nostro **portatile** (IP privato: 192.168.1.187).

All'interno di una rete (più precisamente all'interno di uno stesso dominio di broadcast) tutti i dispositivi devono avere IP differenti (come se fossero dei numeri di telefono).

Gli indirizzi IP sono suddivisi in pubblici e privati e le comuni reti domestiche (in generale qualsiasi rete privata) posseggono un router che, con un indirizzo IP pubblico “dinamico”, si connette ad Internet.

Gli indirizzi IP privati sono utilizzati solo ed esclusivamente all'interno di una rete privata; al di fuori di essa, i dispositivi comunicano con indirizzi pubblici (univoci).

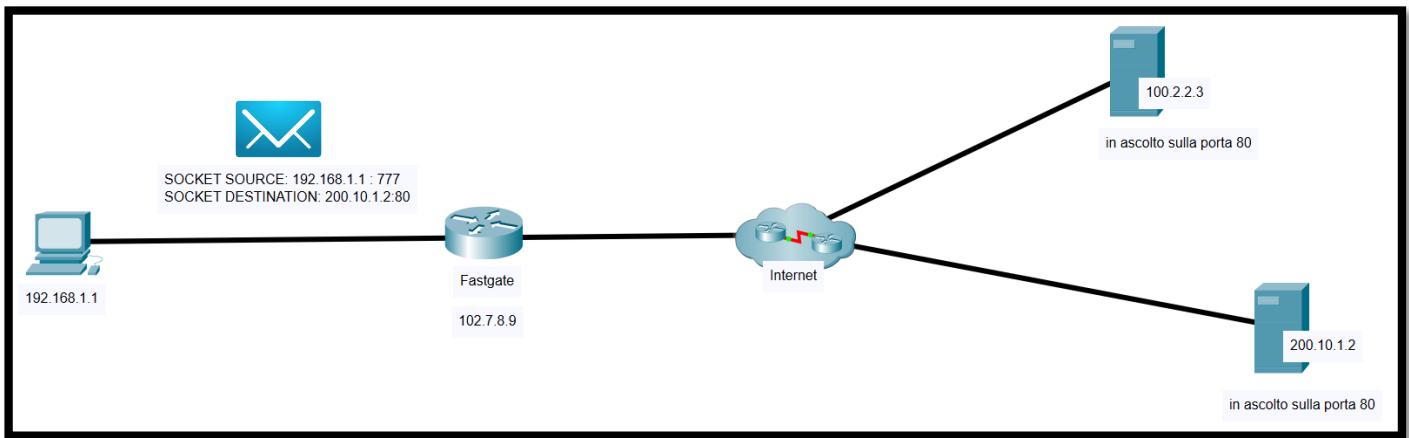
Private IP address space	
From	To
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

“192.168.1.187” è un indirizzo privato, eppure il nostro dispositivo può comunicare attraverso Internet. Questo è possibile perché il nostro router (che da qui in poi, per comodità, chiameremo “fastgate”) esegue la tecnica del PAT (Port Address Translation).

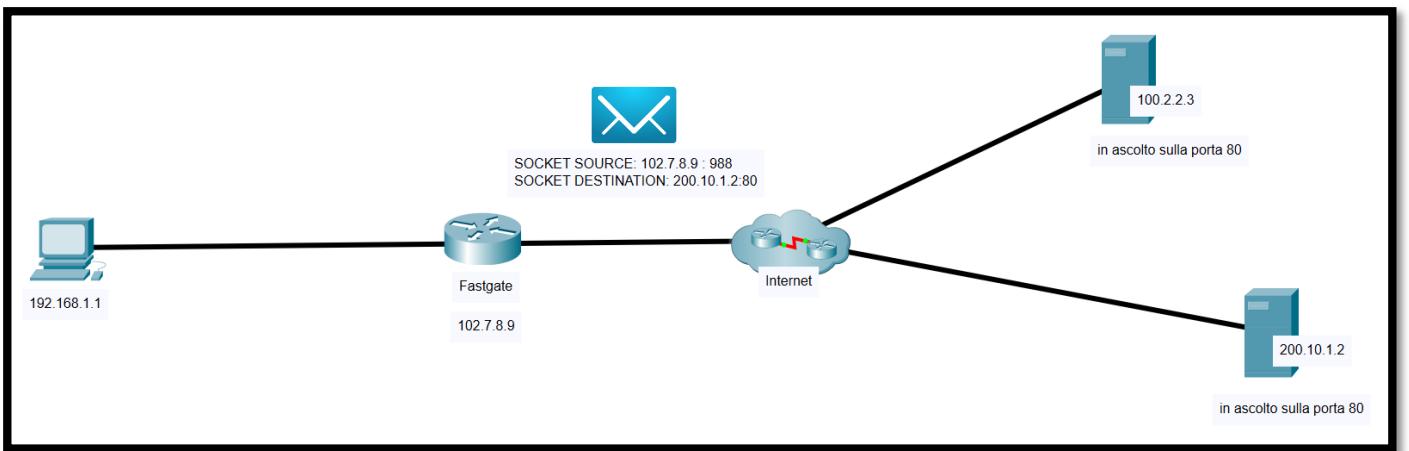
Con questa tecnica, grazie ad un solo IP pubblico (assegnato all' interfaccia verso Internet del fastgate), possiamo instaurare 65535 connessioni contemporaneamente (ognuna su una porta diversa).

Ad ogni nuova richiesta di connessione verso l'esterno, tramite una tabella, il router associa ogni porta ad una connessione. Quando riceverà un messaggio su quella porta, saprà a quale connessione corrisponde:

1. 192.168.1.1 vuole comunicare con 200.10.1.2 sulla porta 80:



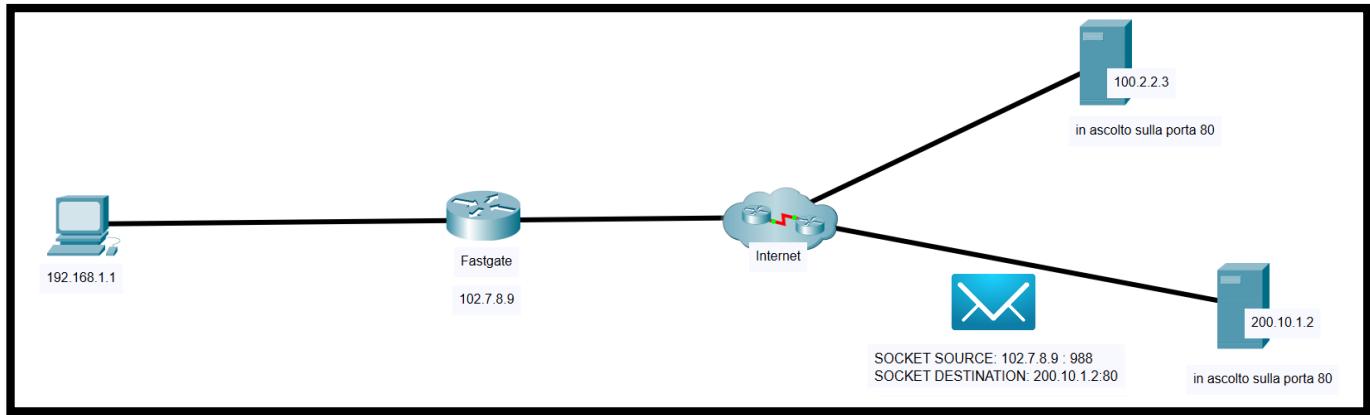
2. Fastgate applica il PAT ed assegna la connessione alla porta 988:



Nel frattempo il router ha memorizzato la corrispondenza:

Protocol	Source socket inside	Destination socket inside	Source outside	socket	Destination socket outside
TCP	192.168.1.1:777	200.10.1.2:80	102.7.8.9:988		200.10.1.2:80

3. Tramite Internet, 200.10.1.2 riceve il messaggio sulla porta 80:



Quando arriva il messaggio il router, consultando la tabella, saprà che quando un messaggio è arrivato sulla porta 988, è destinato a 192.168.1.1 e dovrà rispondere sulla porta 80:

Protocol	Source socket inside	Destination socket inside	Source outside socket	Destination socket outside
TCP	192.168.1.1:777	200.10.1.2:80	102.7.8.9:988	200.10.1.2:80

In caso di altre connessioni, anche allo stesso Server, le corrispondenze tra porta e indirizzo saranno sempre univoche:

Protocol	Source socket inside	Destination socket inside	Source outside socket	Destination socket outside
TCP	192.168.1.1:777	200.10.1.2:80	102.7.8.9:988	200.10.1.2:80
TCP	192.168.1.1:777	100.2.2.3:80	102.7.8.9:1000	100.2.2.3:80
TCP	192.168.1.2:777	100.2.2.3:80	102.7.8.9:2356	100.2.2.3:80

Ma “pubblicare” in Internet è un concetto differente dal comunicare attraverso di essa:

- Per essere raggiunti, chiunque deve sapere come ci chiamiamo (il nostro IP pubblico, oppure l'accoppiata IP pubblico del gateway ed una porta);
- L'IP pubblico che il nostro Internet Service Provider (Fastweb) ha assegnato al nostro router è dinamico; vedremo che avremo bisogno un indirizzo ip pubblico statico.

Dobbiamo, in sostanza, applicare una modifica al PAT e fare in modo che l'host con IP privato “192.168.1.187” abbia associata sempre la stessa porta (es. la 443).

In questo modo:

Protocol	Source socket inside	Destination socket inside	Source socket outside	Destination socket outside
TCP	192.168.1.1:777	200.10.1.2:80	102.7.8.9:443	200.10.1.2:80
TCP	192.168.1.1:777	100.2.2.3:80	102.7.8.9:443	100.2.2.3:80
TCP	192.168.1.2:777	100.2.2.3:80	102.7.8.9:2356	100.2.2.3:80

Il router avrà sempre scritto (dobbiamo specificare sia la porta che da fuori chiunque userà per contattarci, che quella su cui il nostro host si aspetta di ricevere i messaggi provenienti dall'esterno):

Protocol	Source socket inside	Destination socket inside	Source socket outside	Destination socket outside
--	192.168.1.1:443	--	102.7.8.9:443	--
--	--	--	--	--
--	--	--	--	--

4.1 Occorrente

- Router che permetta il port mapping;
- Indirizzo privato statico (per la verità non è un requisito indispensabile, ma lo consigliamo caldamente);
- Indirizzo IP pubblico statico.

Opzionalmente:

- Dominio di secondo livello dove sia possibile modificare il record DNS.

4.2 La richiesta dell'IP pubblico statico

Contattando telefonicamente il nostro ISP, in questo caso “Fastweb”, è possibile richiedere l'assegnazione di un IP pubblico statico al nostro router che ci verrà assegnato nel giro di un paio di giorni.

Da quel momento il port mapping sarà abilitato e funzionante.

4.3 Il port mapping

Con questa tecnica, possiamo associare ad una delle 65535 porte, un indirizzo privato.

In questo modo, qualsiasi pacchetto con:

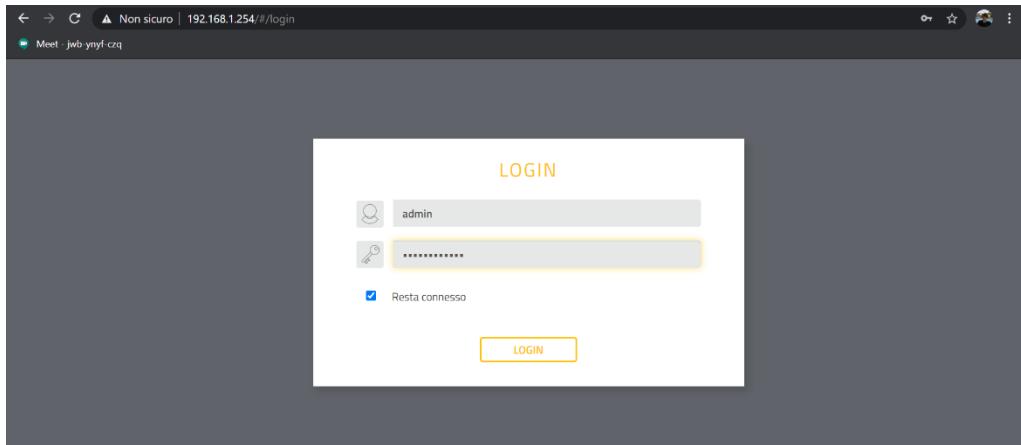
- Destination IP address: 2.229.132.23 (indirizzo pubblico statico del nostro router);
- Destination port: 443 (porta che abbiamo scelto);

sarà indirizzato all'indirizzo ip specificato nel port mapping.

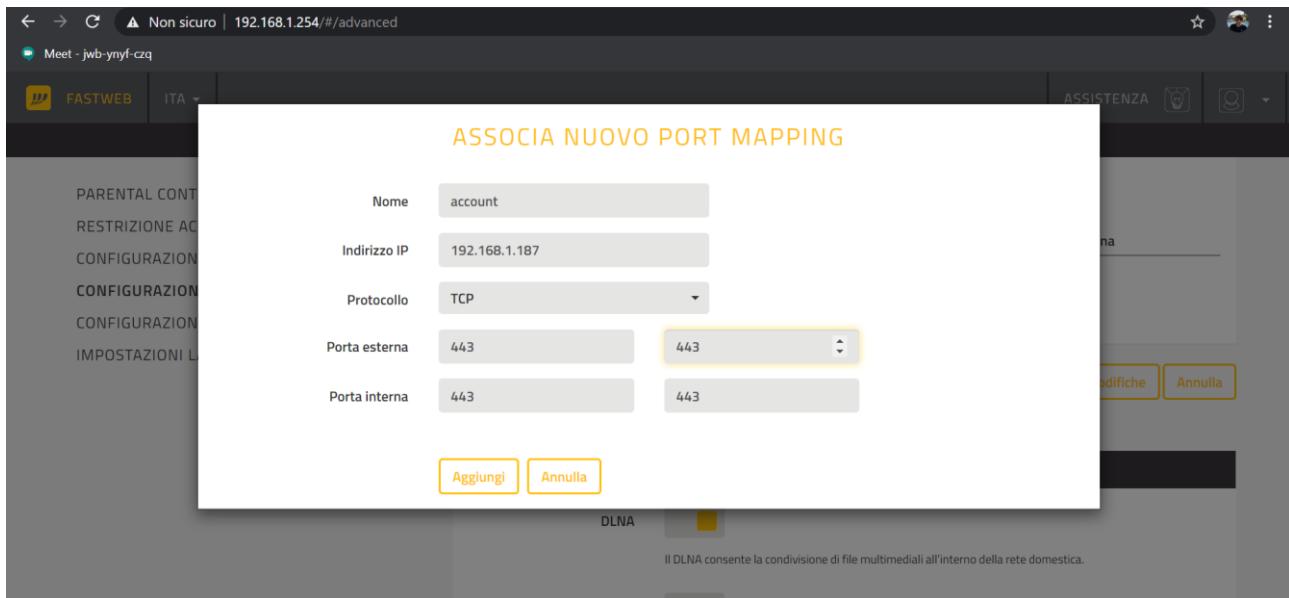
Riprendendo il discorso del PAT, significa avere una entry “statica” nella NAT table del fastgate:

Protocol	Source socket inside	Destination socket inside	Source socket outside	Destination socket outside
--	192.168.1.1:443	--	2.229.132.23:443	--

Dopo aver eseguito l'accesso al fastgate:



Nella sezione “avanzate”:



4.4 Settaggio dell'IP privato statico

Questo passaggio non è obbligatorio, ma può essere utile e comodo.

Fastgate fa anche da server DHCP, ovvero si occupa anche dell'assegnazione dinamica degli IP all'interno della rete.

Questo ci consente di:

- Non preoccuparci della configurazione degli host;
- Non correre il rischio della presenza di IP duplicati.

Avere assegnato un IP dinamicamente, significa che il tempo di utilizzo di quell'IP è limitato da un tempo di lease al termine del quale l'IP andrà restituito e servirà effettuare una nuova richiesta per un nuovo IP.

A quel punto, dovremmo cambiare anche la configurazione del port mapping, oppure escludere il nostro IP (192.168.1.187) dal pool di indirizzi a cui attinge il fastgate per assegnare gli IP agli host che ne fanno richiesta.

Sempre nelle impostazioni “avanzate” è possibile assegnare un IP privato ad un MAC address in modo permanente:

4.5 Configurazione web server

Infine, dobbiamo configurare la porta su cui sarà in ascolto Apache:



4.6 Creazione sottodomainio

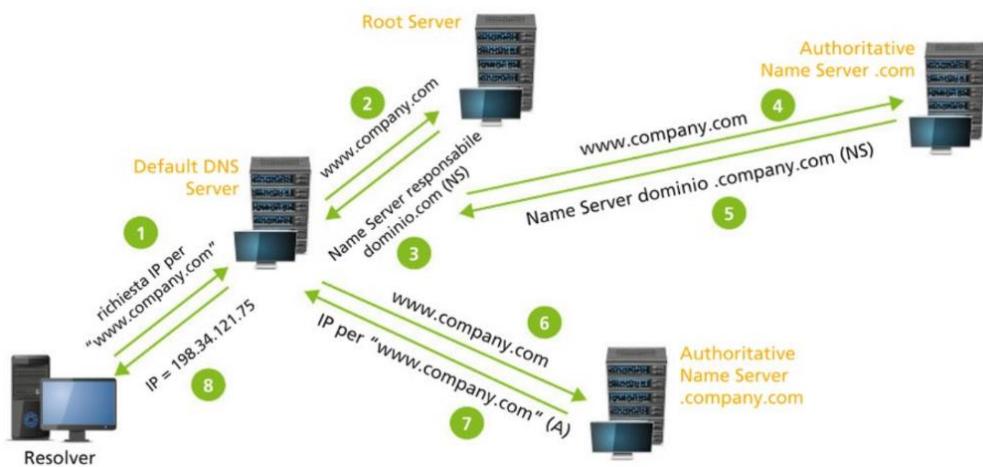
A questo punto, il nostro sito è pubblicato ma raggiungibile solo tramite “2.229.132.23:443”.

Sarebbe più elegante e pratico utilizzare un nome mnemonico.

A questo scopo, esiste il DNS.

Il DNS è un servizio messo a disposizione da una rete di database distribuiti che permette di associare un indirizzo IP ad un nome mnemonico (es. “account.andreabellani.com”).

La risoluzione di un nome avviene nel seguente modo:



Come vediamo, nel momento in cui scriviamo www.andreabellani.com (nell'immagine: www.company.com, ma il procedimento è lo stesso):

1. Il nostro DNS Server di default controlla se conosce a quale indirizzo corrisponde www.andreabellani.com. Se non lo sa:
2. Chiede al Root Server dove si trova il name server che contiene i riferimenti per “.com”;
3. Il Root Server comunica al nostro DNS Server la posizione di “.com”;
4. A “.com” viene chiesto dove si trova il Name Server di “.andreabellani.com”;
5. “.com” lo sa e lo comunica al nostro DNS Server;
6. Una volta raggiunto, esso saprà quale sarà l'IP per www.andreabellani.com (web server che ospita il sito accessibile da www.andreabellani.com).

Comprando un dominio di secondo livello come “andreabellani.com” su godaddy.com, il provider del dominio ci permette anche di modificare i campi (resource record) presenti nel record DNS del Name Server “.andreabellani.com”.

Ecco come si presenta il record DNS:

The screenshot shows a web-based DNS management interface. At the top, there's a navigation bar with 'I miei domini / Impostazioni dominio' and the title 'Gestione DNS' followed by the domain name 'ANDREABELLANI.COM'. Below this, a section titled 'Record' displays a table of current DNS entries. The table has columns for 'Tipo', 'Nome e Cognome', 'Value', and 'TTL'. The records listed are:

Tipo	Nome e Cognome	Value	TTL
CNAME	www	ghs.googlehosted.com	1 ora
MX	@	aspmx.l.google.com (Priorità: 1)	1 ora
MX	@	aspmx2.googlemail.com (Priorità: 10)	1 ora
MX	@	aspmx3.googlemail.com (Priorità: 10)	1 ora
MX	@	alt1.aspmx.l.google.com (Priorità: 5)	1 ora
MX	@	alt2.aspmx.l.google.com (Priorità: 5)	1 ora
NS	@	ns71.domaincontrol.com	1 ora

Esistono diversi tipi di resource record, quelli che a noi interessano per questo progetto sono:

- A: crea la corrispondenza tra nome mnemonico ed indirizzo IPv4;
- CNAME: permette la creazione di “alias”, ovvero “soprannomi”.

Il record CNAME presente indica che, nel momento in cui scriveremo www.andreabellani.com (e non solo “andreabellani.com”), verremo indirizzati a ghs.googlehosted.com (sarà poi Google a preoccuparsi di connettere il nostro sito realizzato con Google Sites).

Sapendo che con un record “A” possiamo assegnare un nome mnemonico ad un indirizzo IPv4 (chiaramente pubblico), ciò che dobbiamo inserire è abbastanza chiaro:

The dialog box for adding a new DNS record has the following fields:

- Tipo ***: A (selected)
- Host ***: account
- Punta a ***: 2.229.132.23
- TTL ***: 30 minuti
- Buttons**: Salva (Save) and Annulla (Cancel)

Da questo momento, chiunque scriva sul proprio browser “account.andreabellani.com” sarà indirizzato al fastgate e, se metterà come destination port “443”, tramite il port mapping riuscirà a contattare il server web installato sul nostro laptop (in ascolto sulla porta “443”).

5 La sicurezza

L'aspetto della sicurezza sappiamo essere un punto critico, soprattutto in progetti “casalinghi” come questo.

Chiariamo subito che la sicurezza che noi possiamo fornire per mezzo dei dispositivi e delle conoscenze che abbiamo è assai limitata (vedremo infatti che saremo ancora vulnerabili ad alcuni specifici attacchi).

Il massimo che possiamo ancora fare però, non è da sottovalutare:

- Servizio HTTPS per la crittazione delle comunicazioni tra i client ed il nostro web server;
- Uso di funzioni PHP particolari per prevenire gli attacchi SQL di tipo injection (oppure, come vedremo, controlli con le variabili PHP superglobali della sessione).

5.1 L'aggiunta del servizio HTTPS

HTTP (HyperText Transfer Protocol) è il protocollo standard per l'invio delle pagine web, tuttavia non è l'unico.

HTTPS è la sua versione “secure” che, tramite i protocolli di livello presentation SSL e TLS (al giorno d'oggi si usa sempre TLS), consente l'invio di comunicazioni sicure.

HTTPS prevede che:

1. un client col proprio browser si colleghi al sito web di account.andreabellani.com (HTTPS well-known port: 443);
2. account.andreabellani.com invii al browser il suo certificato pubblico SSL/TLS;
3. il browser, con la chiave pubblica di SSL For Free, verifichi la firma di SSL For Free sul certificato SSL/TLS di account.andreabellani.com;
4. il browser generi una chiave segreta (simmetrica) e utilizzi la chiave pubblica di account.andreabellani.com, per crittografarla;
5. venga inviata la chiave segreta crittografata a account.andreabellani.com;
6. account.andreabellani.com la decritti con la propria chiave privata e la trattenga;
7. il browser e account.andreabellani.com utilizzeranno la chiave segreta condivisa per crittografare le comunicazioni.

I motivi per cui non si utilizza la crittografia asimmetrica durante tutta la comunicazione sono:

- Il client normalmente non possiede certificati;
- La crittografia simmetrica è molto più rapida di quella asimmetrica e meno laboriosa.

5.1.1 Ottenere un certificato

I certificati sono rilasciati dalle Certificate Authority (CA), ovvero degli enti certificati che rilasciano dei certificati digitali.

Possiamo riassumere il procedimento in fasi:

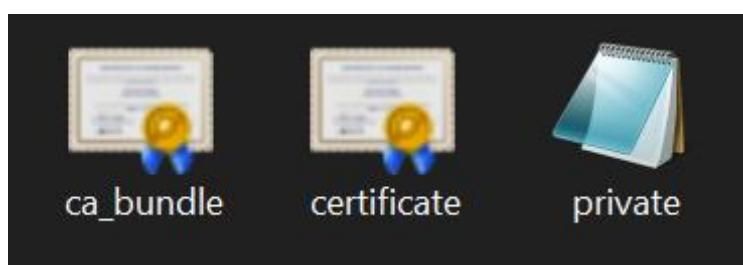
1. Individuazione delle ente certificatore;
2. Richiesta del certificato;
3. Ottenimento del certificato;
4. Download dei file che compongono il certificato.

L'ente certificatore presso cui ci siamo rivolti è [SSL For Free](#), ovvero un sito che rilascia certificati gratuitamente validi per 90 giorni.

The screenshot shows the SSL For Free website. At the top, there's a navigation bar with links for Login, Register, Need Help?, and a language selection dropdown set to Italian. Below the header, a large green button says "Create Free SSL Certificate". The main content area features three circular icons: a dollar sign for "100% Free Forever", a checkmark for "Widely Trusted", and a globe for "Enjoy SSL Benefits". Below each icon is a brief description and a small note at the bottom right of the section.

Una volta effettuata la registrazione, aver inserito il dominio che utilizzerà il certificato ed esserci convalidati, il certificato sarà generato e potremo scaricare il “.zip” contenente i 3 file che compongono il nostro certificato:

- Certificate.crt : file contenente la chiave pubblica;
- Ca_bundle.crt : file “catena”, contenente i riferimenti per contattare altre CA (che il browser già conosce come affidabili) che possano garantire SSL For Free come nuova CA autentica;
- Private.key : file contenente la chiave privata.



5.1.2 Aggiunta del certificato

Le modalità di aggiunta del certificato variano da web server a web server e da versione a versione.

Su Apache v. 2.4.29 questa è una procedura funzionante.

Nel file “httpd.conf” (ovvero il file di configurazione del web server), “scommentiamo” la riga evidenziata:

```
#LoadModule spelling_module modules/mod_speling.so
#LoadModule ssl_module modules/mod_ssl.so
#LoadModule status_module modules/mod_status.so
```

Per “scommentare” basta eliminare il “#” davanti.

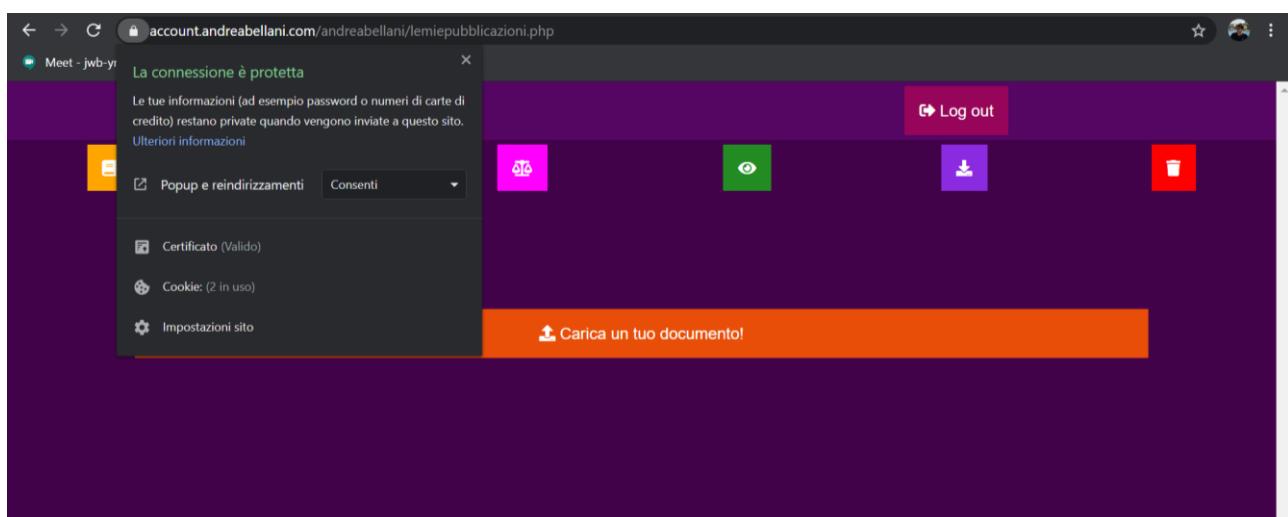
A questo punto dobbiamo indicare la posizione del file che conterrà le informazioni per trovare i file del certificato:

```
LoadModule ssl_module modules/mod_ssl.so
Include conf/extra/httpd-ssl.conf
```

Nella cartella “conf” (cartella di default che contiene i file di configurazione come “httpd.conf”), creiamo una nuova cartella “extra” al cui interno creiamo il file “httpd-ssl.conf” che dovrà contenere le seguenti righe di testo (può essere tranquillamente editato col blocco note):

```
<VirtualHost 192.168.1.187:443>
SSLEngine On
SSLCertificateFile "C:\Users\Andrea\Desktop\Andrea\Scuola\Informatica\Laboratorio\PHP\usbwebserver\apache2\conf\Certificato\certificate.crt"
SSLCertificateChainFile "C:\Users\Andrea\Desktop\Andrea\Scuola\Informatica\Laboratorio\PHP\usbwebserver\apache2\conf\Certificato\ca_bundle.crt"
SSLCertificateKeyFile "C:\Users\Andrea\Desktop\Andrea\Scuola\Informatica\Laboratorio\PHP\usbwebserver\apache2\conf\Certificato\private.key"
ServerName account.andreabellani.com
</VirtualHost>
```

Dopo aver riavviato il web server, il risultato sarà il seguente:



Nota bene “troubleshooting”:

Realizzare l’intero sito, ha richiesto un’ottima capacità di individuazione dei problemi e formulazione delle soluzioni, in quanto abbiamo dovuto configurare e modificare file di configurazione (mai fatto prima) e seguire le loro regole.

In questa sezione, il troubleshooting ci ha permesso di capire quale delle molteplici soluzioni trovate online fossero le più opportune.

Abbiamo proceduto col seguente schema:

1. Controllo di correttezza dei campi di “httpd.conf”;
2. Controllo di correttezza dei campi “httpd-ssl.conf”.

Nel momento in cui usbwebserver non riusciva a connettersi ad Apache, commentavamo la riga di “httpd.conf” per includere “httpd-ssl.conf”.

Se l’errore persisteva, sostituivamo “httpd.conf” con il file di backup tenuto dall’inizio.

Se l’errore scompariva, eliminavamo una per una le istruzioni scritte su “httpd-ssl.conf” cercando online diversi esempi.

5.2 La prevenzione dagli SQL injection

Inizialmente, avevamo programmato di mostrare come il nostro sito venisse prima attaccato da un SQL injection e poi come la stessa query fosse inefficace.

Con nostro stupore, ci siamo accorti che i molteplici controlli e le funzioni che stavamo utilizzando, in qualche modo riuscivano già a proteggerci da questo attacco. Al posto, quindi, di semplificare il codice per riuscire ad attaccare a tutti i costi la base di dati, abbiamo pensato di spiegare come siamo riusciti, inconsapevolmente, a proteggerci da questi attacchi.

5.2.1 Le SQL injection

Gli attacchi SQL di tipo injection sono considerati tra gli attacchi informatici più diffusi al mondo ed eseguirli tramite un applicativo che non controlli i dati inseriti in input è molto semplice.

In sostanza, tutto quello che noi scriviamo nei campi di testo di un form, poi verrà inserito in una query e sarà inviato al database; in altre parole stiamo quasi permettendo a chiunque di scrivere sul nostro database inconsapevolmente.



```
$query = "SELECT Username FROM utente WHERE Username='".$_POST['username']."'";  
$check = mysqli_query($con, $query);
```

Questo tipo di attacco può essere usato per:

- Modificare il database;
- Produrre crash ed errori per riuscire a capire la struttura della base di dati;
- Rubare i dati di altri utenti.

Esistono diversi tipi di attacchi SQL che spesso risultano sufficientemente complessi per i nostri mezzi e le nostre conoscenze. Tra quelli più semplici, abbiamo constatato che i due più diffusi sono:

1. Multiple query injection: si usa la query di partenza per inviare un'altra query fraudolenta;
2. Boolean value injection: si scrivono delle espressioni booleane in modo che i valori ritornati dalle selezioni (SELECT ... FROM ... WHERE ...) non contengano i dati solo dell'utente interessato, ma di tutti gli utenti.

Nota bene: i nomi delle tipologie di injection sono stati pensati da noi per identificare questi attacchi nella spiegazione seguente.

5.2.2 Multiple query injection

In SQL è possibile scrivere più query alla volta, ad esempio:

Il tuo username:

```
a'; TRUNCATE TABLE documento; SELECT Username FROM utente WHERE Username='
```

Quando questa stringa viene inclusa già nella prima query che incontriamo:

```
$query = "SELECT Username FROM utente WHERE Username='".$_POST['username']."'";  
$check = mysqli_query($con, $query);
```

Il risultato è il seguente:

```
SELECT Username FROM utente WHERE Username='a'; TRUNCATE TABLE documento;
SELECT Username FROM utente WHERE Username='';
```

1. `SELECT Username FROM utente WHERE Username=''` : parte della query già scritta dello script;
2. `'a'` : parte di query per terminare la query iniziale;
3. `TRUNCATE TABLE documento;` : query fraudolenta per svuotare la tabella da tutte le entry (volendo potremmo scrivere qualsiasi query conforme ai permessi dell'utente che usiamo per interrogare il DBMS: ovvero tutte a parte “DROP DATABASE”);
4. `SELECT Username FROM utente WHERE Username=''` : nuova query per concludere l'ultimo apice;
5. `''` : chiusura della stringa nello script.

Per fare questo, l'hacker avrebbe dovuto solo sapere:

- Nome di una tabella non referenziata (una tabella referenziata, non verrà eliminata da MySQL);
- Nome di un attributo per la query di “chiusura”.

Per prevenire questo attacco, basta utilizzare la funzione `mysqli_query()`.

Proprio allo scopo di prevenire attacchi di questo tipo, questa funzione non permette l'invio di query multiple, inviabili solo con `mysqli_multi_query()`.

5.2.3 Boolean value injection

Rispetto al tipo precedente, queste query sono utilizzate non per recare danni al database, bensì per visionare dati di altri utenti; in generale per rendere i predicati di “WHERE” sempre veri.

Per realizzare questa query, non si fa altro che aggiungere alla query un'espressione booleana sempre verificata (es. “Username = ‘andrea’ OR ‘ciao’=‘ciao’”):



```
$query = "SELECT Username FROM utente WHERE Username='".$POST['username']."'";  
$check = mysqli_query($con, $query);
```

Il risultato è il seguente:

```
SELECT Username FROM utente WHERE Username='andrea' OR 'ciao'='ciao';
```

- `SELECT Username FROM utente WHERE Username='`: parte della query già scritta dello script;
- `andrea'`: username qualsiasi solo per completare il primo WHERE;
- `OR 'ciao'='ciao'`: predicato sempre vero;
- `'`: chiusura della stringa nello script.

I valori restituiti dalla query saranno tutti quelli della tabella.

Il vantaggio di questo tipo di attacco sta nella sua semplicità, ma questo è anche un punto a sfavore.

Cerchiamo di capire quali siano i campi dove è possibile scrivere dei valori che saranno poi inseriti nelle query, tenendo presente che all'interno del sito sono presenti dei controlli che, tramite l'ID (quindi non un valore inseribile dall'utente), controllano se le password presenti nella sessione PHP corrispondono con quelle memorizzate nel proprio record del database.

Gli unici possibili punti deboli sarebbero:

- Registrati.html:
 - Campo username;
 - Uno dei campi password.
- Accedi.html:
 - Campo username;
 - Campo password.

Vediamo come ognuno di questi campi è protetto:

5.2.3.1 Registrazione → username

La prima query che utilizza un campo inserito nel form, è la ricerca di un utente con username uguale a quello inserito.

La query vista prima, restituirà sempre un valore, dunque non c'è pericolo che qualcuno possa ottenere quello username:

```
$is_in=mysqli_fetch_array($check);
//controllo se il risultato della query è vuoto (se è vuoto, significa che nessuno ha quello username)
if ($is_in[0]!="")
{
    session_destroy();
    print("
        <script>
            alert('lo username inserito non e\' disponibile');
            window.close();
        </script>
    ");
}
```

L'hacker potrebbe comunque completare l'inserimento:

- Se la tabella era già vuota (in quel caso la query sarebbe comunque vuota perché equivalebbe ad una proiezione di una tabella senza entry);
- Se al posto di inserire una query sempre vera, se ne inserisce una sempre falsa (es. “AND ‘ciao’<>’ciao’”).

In entrambi i casi il database memorizzerebbe nella tabella i valori 1 o 0 (vero o falso), mentre nella variabile di sessione sarà memorizzato lo username realmente inserito.

Al primo controllo, all'utente sarà comunicato che le sue credenziali sono state cambiate e che dovrà rieseguire l'accesso. Qui il controllo in “lemiepubblicazioni.php”:

```
$username = $_SESSION['username'];
$password = $_SESSION['password'];
$id = $_SESSION["id"];
$con = mysqli_connect($nomeUSB,$userUSB,$passUSB, $dbUSB);
$query = "SELECT Username FROM utente WHERE ID='".$id."'";
$actual_username=mysqli_fetch_array(mysqli_query($con, $query));
$query = "SELECT Password FROM utente WHERE ID='".$id."'";
$actual_password=mysqli_fetch_array(mysqli_query($con, $query));
if (($actual_password[0] == $password) AND ($actual_username[0] == $username))
{
    $r_color = mysqli_fetch_array(mysqli_query($con, "SELECT FKNomeSkin
$skin = $r_color[0];
print("
```

5.2.3.2 Registrazione → password

In questo caso, ciò che blocca l'attacco è il controllo delle variabili di sessione (che appunto non avranno 0 od 1, ma ciò che l'utente ha realmente scritto nel campo password).

5.2.3.3 Accedi → username

Anche in questo caso, qualsiasi cosa scriveremo nelle variabili di sessione corrisponderà a ciò che abbiamo realmente scritto nel form e non corrisponderà ad un utente od a una password.

5.2.3.4 Accedi → password

```
$query = "SELECT Password FROM utente WHERE Username='".$_POST['username']."'"; /
$my_query = mysqli_query($con, $query); //questa query è la stessa di prima, ma se
$pass = mysqli_fetch_array($my_query);
if ($pass[0] != $_POST['password']) //se le password non corrispondono
{
    session_destroy();
    print("
        <script>
            alert('la password non e\' corretta');
            window.close();
        </script>
    ");
}
```

Essendo la password selezionata tramite lo username, non sarebbe possibile, se non nel caso in cui lo username sia stato “iniettato”, ma a quel punto ricadremmo nel punto “5.2.3.3”.

5.2.4 Soluzioni

Concludiamo dicendo che:

- Le SQL injection:
 - Sono strettamente legate ai dati ed all'applicativo, pertanto non esistono le injection che funzioneranno sempre o che daranno sempre i comportamenti che ci si aspetta.
 - È opportuno conoscere sempre le motivazioni di riuscita o meno delle injection sul proprio applicativo.
- I punti dove sono possibili le injection sono quelli dove l'utente può inserire campi di testo.
- Esistono diversi modi per preverirle e, nel nostro caso, utilizziamo dei controlli serrati anche per mezzo di variabili memorizzate in PHP (e quindi con contenuto diverso da quello che la query in SQL processa).
- Esistono funzioni che bloccano le injection “a priori”:
 - mysqli_query(): per le query multiple;
 - mysqli_real_escape_string(): elimina alcuni caratteri speciali dalle stringhe (\n ‘ \ “), prima che esse vengano usate dentro mysqli_query() (non abbiamo avuto bisogno di utilizzare questa funzione);
 - Altre tecniche più avanzate come i PDO (PHP Data Objects).

6 Considerazioni finali

6.1 Ulteriori considerazioni sulla sicurezza

In conclusione, per aver realizzato un applicativo “casalingo” siamo riusciti mettere in campo misure di sicurezza abbastanza efficaci.

Tuttavia, esistono ancora dei rischi a cui siamo esposti. Noi abbiamo individuato:

- Possibilità per un utente di scaricare file altrui nel momento in cui ne conosce l'ID;
- SQL injection più sofisticate (alcune prevedono di eseguire delle UNION che, in caso di tabelle con numerose entry, provocherebbero un crash dell'applicativo);
- Visibilità da Internet del nostro router e della nostra macchina;
- Visibilità da Internet delle pagine di gestione della base di dati che, nonostante siano protette da password, rimangono comunque vulnerabili ad attacchi di forza bruta;
- Rischi di manipolazione delle variabili delle sessioni PHP.

6.2 Fonti

Realizzazione della base di dati:

- Dispense di Antonio Albunia del corso Base di dati.
- www.w3schools.com
- Paolo Camagni, Riccardo Nikolassy – Corso di informatica Linguaggio C e C++

Realizzazione del sito web (principalmente):

- www.php.net
- www.stackoverflow.com
- www.w3schools.com

Pubblicazione del sito:

- PAT:
 - Simulazioni NAT e PAT con Packet Tracer
 - Elena Baldino, Renato Rondano, Antonio Spano, Cesare Iacobelli – Internetworking
 - Romano Fantacci - Reti di Telecomunicazioni: Fondamenti e Tecnologie Internet
- DNS:
 - Elena Baldino, Renato Rondano, Antonio Spano, Cesare Iacobelli – Internetworking
- Port mapping:
 - www.fastweb.it
 - www.navigaweb.net

La sicurezza:

- HTTPS:
 - www.geekflare.com
- SQL injection:
 - www.w3schools.com
 - Elena Baldino, Renato Rondano, Fausto Beltramo, Cesare Iacobelli – Internetworking
 - Cesare Iacobelli, Marialaura Ajme, Velia Marrone – Eprogram