

IIS Einaudi-Scarpa, Montebelluna

IL MUSEO DELLA MATURITA'



Progetto di maturità

By: Ionuț Suciuc Augustin

Classe: 5B INF

A.S. 2019-2020

DESCRIZIONE

Il progetto del “museo della maturità” comprende la creazione del **sito web** di un museo che espone opere e sculture da tutto il mondo, e la creazione della **rete interna** del museo.

Il sito web del museo permette di vedere le opere digitalmente da remoto dopo essersi registranti sul sito avendo acquistato il biglietto all'ingresso (la registrazione richiede il codice di 10 caratteri presente sul biglietto cartaceo). Gli utenti che non hanno un account sono liberi di visitare tutte le pagine del museo, ma non quella delle opere, solo dopo aver fatto il login con il proprio nickname e password l'utente può vedere anche la pagina delle “opere”. Per ogni opera è disponibile una pagina personalizzata con un'immagine e con le informazioni dettagliate sull'autore che l'ha creata e informazioni dettagliate sull'opera stessa (prese dal database), le opere possono essere specializzazioni di tipo scultura oppure pittura dell'entità padre opera, è inoltre presente un filtro dove l'utente può scegliere quale tipo di opera visualizzare. L'admin del sito può fare diverse operazioni CRUD tramite i bottoni visibili solo a lui che lo portano alle diverse pagine di gestione. Può inserire nuove opere cliccando su “inserisci una nuova opera”, da qui può scegliere di inserire una nuova pittura oppure una nuova scultura, può selezionare l'autore che l'ha creata ed anche la corrente artistica a cui appartiene. Può scegliere di inserire nuove immagini cliccando su “inserisci una nuova immagine”. Può scegliere di inserire nuovi autori cliccando su “inserisci una nuova autore” da qui potrà inserire i dati anagrafici dell'autore, può inoltre modificare il piano e la stanza dove si trovano le opere e volendo può scegliere di eliminarle. Le altre pagine che compongono il sito sono: la pagina “presentazione” dove c'è una breve descrizione del museo, e delle immagini che sono scorribili del suo interno. La pagina “info” dove c'è l'orario di apertura del museo in formato .pdf , creato con la libreria php “fpdf” e scaricabile sul proprio dispositivo, la mappa embedded di GoogleMaps che mostra la posizione del museo, e le tariffe dei biglietti. Oltre a questo un utente può scegliere di disconnettersi cliccando su “logout” oppure fare il login o la registrazione cliccando su “login”.

La rete interna permette ai visitatori di accedere alla rete Wifi tramite un router wireless dopo essere stati autenticati dal server Radius. Grazie al proprio smartphone possono avere la connessione ad internet mentre stanno visitando il museo. La rete è inoltre protetta da un packet-filtering firewall. Sono presenti anche dei PC usati per l'amministrazione interna, come i pc della lobby del museo ed il pc del negozio di accessori che sono separati dalla resto della rete con una VLAN. Il sito web è hostato su cloud grazie ai vantaggi di questo, e non su un server locale.

Per questo progetto per la creazione del back-end ho utilizzato il linguaggio di scripting PHP, per la manipolazione dei dati SQL e per il front-end ho usato HTML, CSS e Javascript.

OBIETTIVI E PRIORITA'

LINK PER ANDARE AL SITO

<https://ilmuseodellamaturita.000webhostapp.com/index.php>

OBIETTIVI

GESTIRE I DATI:

- I dati degli account degli utenti (username, password e privilegi)
- I dati delle opere pitture/sculture
- I dati degli autori
- I dati delle correnti artistiche
- Le immagini
- i biglietti comprati

RIUSCIRE A FARE OPERAZIONI DI:

- Inserimento/cancellazione/modifica (se l'utente è amministratore)
- Sola lettura se l'utente non è amministratore

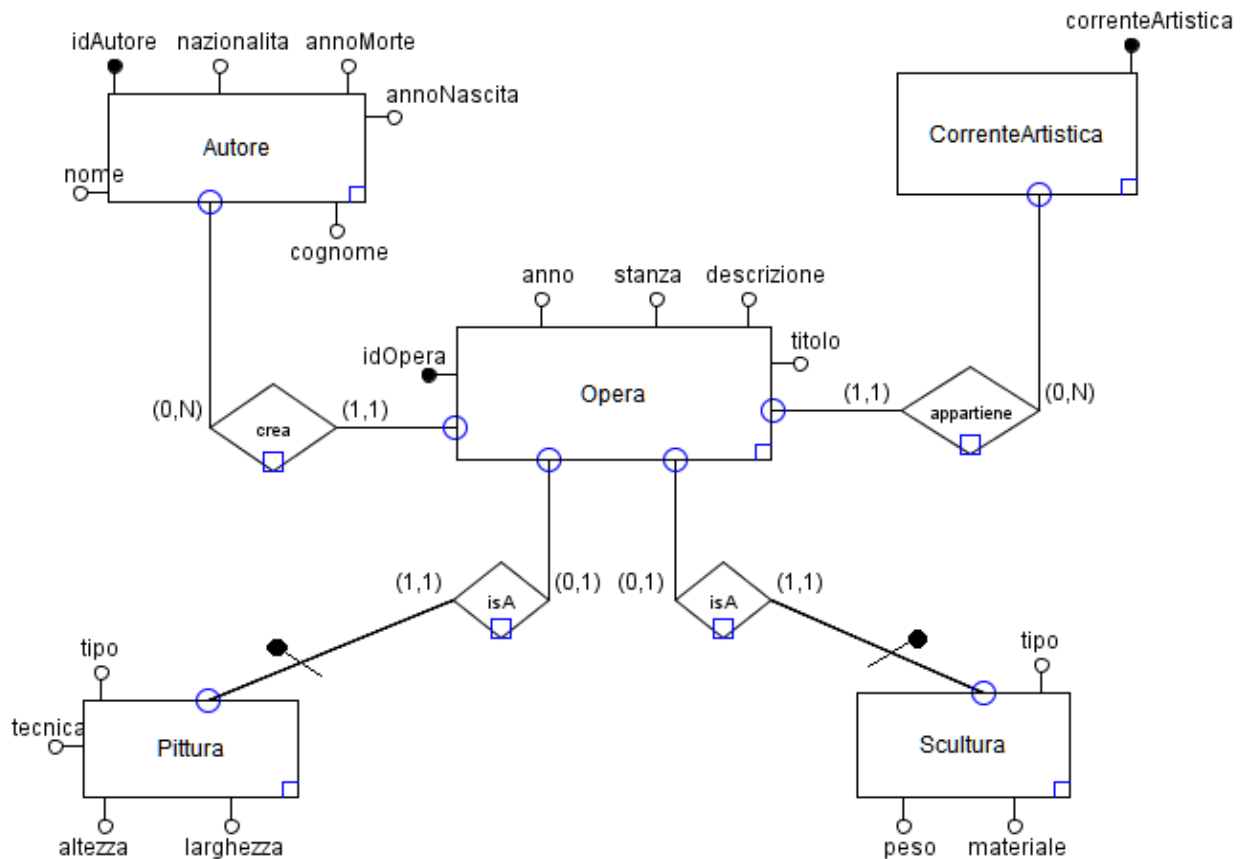
PROGETTARE:

- La rete interna del museo che permette ai visitatori di avere la connessione ad internet
- la gestione della sicurezza della rete.
- Un sito visivamente bello e interattivo che possa essere visualizzato anche su smartphone

PRIORITA'

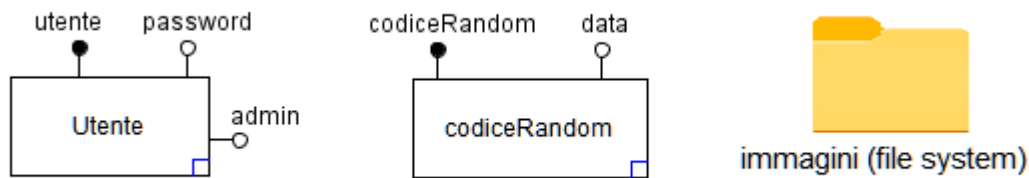
- 1) La creazione dello schema concettuale / logico / fisico
- 2) La creazione del database con gli utenti
- 3) La creazione della pagina di registrazione
- 4) La gestione dei biglietti
- 5) La creazione della pagina di login
- 6) Il popolamento del database
- 7) La creazione delle pagine statiche "presentazione" e "info".
- 8) La creazione delle pagine dinamiche dell'admin come ad esempio:
"inserimento opera", "inserimento immagine", "cancellazione opera",
"modifica opera" ecc...
- 9) L'implementazione dei privilegi e la protezione della pagine
- 10) La rete interna del museo
- 11) La sicurezza della rete interna
- 12) La grafica (che è stata sviluppata un pò alla volta, quando ho implementato grandi cambiamenti o ho creato nuove pagine)

SCHEMA CONCETTUALE



MOTIVAZIONI:

- Scultura e Pittura sono delle specializzazioni dell'entità padre "opere", ognuna di queste ha dei propri attributi di specializzazione ed ereditano la chiave primaria dal padre.
- Un'opera non deve per forza essere una Scultura (0,1) e nemmeno una Pittura (0,1), ma ogni Pittura o Scultura deve per forza essere un'opera (1,1).
- Una opera deve appartenere ad una corrente artistica (1,1) ma una corrente artistica non deve per forza riferire un'opera (perché se una corrente viene inserito nel DataBase, e poi l'opera che riferisce viene cancellata, questa può rimanere inserita per poi essere riusata in seguito).
- Una corrente artistica inoltre può appartenere a più opere (0,N).
- Un'opera deve avere un autore che l'ha creata (1,1) ma un autore non deve per forza riferire un'opera (perché se l'autore viene inserito nel DataBase, e poi l'opera che riferisce viene cancellata, questo può rimanere inserito per poi essere riusato in seguito).
- Un'autore inoltre può appartenere a più opere (0,N).
- L'Anno di morte dell'autore può essere null perché magari questo non è ancora morto.
- La lunghezza e l'altezza sono espresse in centimetri, mentre il peso il kilogrammi.



- la chiave primaria dell'utente è il suo nickname, che deve essere univoco
- Ogni tabella ha attributi non obbligatori, perché potrebbero non essere conosciuto come ad esempio il nome e cognome di chi ha creato l'opera, oppure l'anno in cui è stato realizzata ecc...
- L'utente ha un attributo "admin" che lo caratterizza come amministratore (se il suo valore è 1) oppure utente normale (se il suo valore è 0)
- La tabella "codiceRandom" contiene il codice univoco necessario per la registrazione, il codice si trova sul biglietto comprato all'entrata del museo ed è composto da 10 caratteri casuali. Oltre a questo c'è anche una data che permette al database di eliminare ogni 24h i codici, perciò una volta comprato l'utente ha 24h per registrare un account.
- Le immagini ho deciso di salvarle su file system del webserver invece che sul DataBase perché ogni opera ha soltanto 2 immagini, perciò durante la progettazione ho pensato fosse più facile inserirle e prelevarle da filesystem piuttosto che creare nuovi file con operazioni di inserimento e prelevamento da DataBase.

SCHEMA LOGICO

Utente(utente(PK),password,admin)

codiceRandom(codiceRandom(PK),data)

Opera(idOpera(PK),stanza,descrizione,titolo,correnteArtistica(FK),idAutore(FK))

CorrenteArtistica(correnteArtistica(PK))

Autore(idAutore(PK),nazionalita,annoMorte,annoNascita,nome,cognome)

Pittura(idOpera(FK)(FK),altezza,larghezza,tecnica,tipo)

Scultura(idOpera(FK)(FK),peso,materiale,tipo)

SCHEMA FISICO

```
CREATE TABLE `opera` (  
  `idOpera` int,  
  `titolo` varchar(50),  
  `piano` varchar(50),  
  `stanza` varchar(50),  
  `anno` int DEFAULT NULL,  
  `descrizione` varchar(3000),  
  `correnteartistica` varchar(50),  
  `idAutore` int,  
  PRIMARY KEY (idOpera),  
  FOREIGN KEY (idAutore)  
REFERENCES autore(idAutore),  
FOREIGN KEY (correnteArtistica)  
REFERENCES CorrenteArtistica(  
correnteArtistica ),  
);  
  
CREATE TABLE `codicerandom` (  
  `codicerandom` varchar(12),  
  `data` timestamp NOT NULL  
DEFAULT current_timestamp(),  
  PRIMARY KEY(codicerandom),  
);  
  
CREATE TABLE `correnteartistica` (  
  `nomeCorrente` varchar(50),  
  PRIMARY KEY(nomeCorrente),  
);  
  
CREATE TABLE `scultura` (  
  `peso` int DEFAULT NULL,  
  `materiale` varchar(50)  
DEFAULT NULL,  
  `idOpera` int,  
  `tipo` varchar(50) DEFAULT  
`scultura`  
  PRIMARY KEY(idOpera),  
  FOREIGN KEY(idOpera)  
REFERENCES opera(idOpera);  
);
```

```
CREATE TABLE `pittura` (  
  `tecnica` varchar(50) DEFAULT  
NULL,  
  `larghezza` int DEFAULT NULL,  
  `altezza` int DEFAULT NULL,  
  `idOpera` int NOT NULL,  
  `tipo` varchar(50) DEFAULT  
`pittura`,  
  PRIMARY KEY(idOpera),  
);  
  
CREATE TABLE `utente` (  
  `utente` varchar(50),  
  `password` varchar(300) NOT  
NULL,  
  `admin` int NOT NULL DEFAULT  
`0`,  
  PRIMARY KEY(utente),  
);  
  
CREATE TABLE `autore` (  
  `idAutore` int NOT NULL,  
  `nome` varchar(50) DEFAULT  
NULL,  
  `cognome` varchar(50) DEFAULT  
NULL,  
  `nazionalita` varchar(50)  
DEFAULT NULL,  
  `dataNascita` date DEFAULT  
NULL,  
  `dataMorte` date DEFAULT NULL  
CHECK `dataMorte` < Date();  
  PRIMARY KEY(idAutore),  
);
```

ESEMPIO DI UN'ISTANZA

La tabella “**opera**” (generalizzazione)

	idOpera	titolo	piano	stanza	anno	descrizione	correnteartistica	idAutore
<input type="checkbox"/> Edit Copy Delete	1	Il figlio dell'uomo	1	1	1964	l'opera raffigura, in primo piano, un uomo il cui v...	Surrealismo	12
<input type="checkbox"/> Edit Copy Delete	2	Il Bacio	1	1	1859	La scena è ambientata in un vago interno medievale...	Romanticismo	15
<input type="checkbox"/> Edit Copy Delete	3	Persistenza della memoria	1	2	1931	La persistenza della memoria raffigura un paesaggi...	Surrealismo	3
<input type="checkbox"/> Edit Copy Delete	4	David	1	2	1504	Largamente considerato un capolavoro della scultur...	Rinascimento	4

La tabella “**pittura**” (entità figlia)

	tecnica	larghezza	altezza	idOpera	tipo
<input type="checkbox"/> Edit Copy Delete	olio su tela	116	89	1	pittura
<input type="checkbox"/> Edit Copy Delete	olio su tela	88	112	2	pittura
<input type="checkbox"/> Edit Copy Delete	olio su tela	33	24	3	pittura

La tabella “**scultura**” (entità figlia)

	peso	materiale	idOpera	tipo
<input type="checkbox"/> Edit Copy Delete	5000	marmo	4	scultura
<input type="checkbox"/> Edit Copy Delete	3000	acciaio inossidabile	6	scultura
<input type="checkbox"/> Edit Copy Delete	20	Calcare	11	scultura

La tabella “**corrente artistica**”

	nomeCorrente
<input type="checkbox"/> Edit Copy Delete	Astrattismo
<input type="checkbox"/> Edit Copy Delete	Cubismo

La tabella “**autore**”

	idAutore	nome	cognome	nazionalita	dataNascita	dataMorte
<input type="checkbox"/> Edit Copy Delete	3	Salvador	Dali	Spagnola	1904-05-11	1989-01-23
<input type="checkbox"/> Edit Copy Delete	7	Pablo	Picasso	Spagnola	1881-10-25	1973-04-08
<input type="checkbox"/> Edit Copy Delete	12	Rene	Margitte	Francese	1898-11-21	1967-08-15

La tabella “**utente**”

	utente	password	admin
<input type="checkbox"/> Edit Copy Delete	aaaa	f6dd5e69e7b909f6b2447810b475a35a66630bc720cefe2ca3...	0
<input type="checkbox"/> Edit Copy Delete	adminnn	6426676144ad8dc5d7a79b4acfa0a292a36f41c9ee4f5420df...	1
<input type="checkbox"/> Edit Copy Delete	ciao	18901c8dd7498df86446b8616802c5d46f8c1e84c36d7bc759...	0
<input type="checkbox"/> Edit Copy Delete	marco	7a0c6fc328e202da0501d6451b5127ec4b1d7235a02651a13c...	0

La tabella “**codicerandom**” (biglietto)

	codicerandom	data
<input type="checkbox"/> Edit Copy Delete	Bq1xkoSrN5	2020-06-11 19:39:28
<input type="checkbox"/> Edit Copy Delete	FiuRYxTr3t	2020-06-11 19:39:26
<input type="checkbox"/> Edit Copy Delete	iPvpzod2o4	2020-06-11 19:39:27

FUNZIONALITA' SIGNIFICATIVE

IL "CODICE BIGLIETTO"

GENERA CODICE BIGLIETTO

CODICE: cbon5gTYYm

REGISTRAZIONE

UTENTE:
PASSWORD:
CODICE BIGLIETTO:

REGISTRAZIONE



COMPRA

Per potersi registrare l'utente ha bisogno di un "codice biglietto" che si ottiene comprando fisicamente il biglietto all'entrata del museo, nel mio sito ho simulato questo acquisto che si presenta in questo modo, con un codice di 10 caratteri casuali. Il codice viene anche generato anche sotto forma di Qrcode (che ad esempio potrà essere stampato sul biglietto cartaceo) e può essere visualizzato scansionandolo con la fotocamera del proprio telefono.

Questo è il codice che genera la stringa casuale, un ciclo For che prende per 10 volte casualmente una lettera da quelle elencate nella stringa \$characters. Per generare poi il codice QR ho usato l'API di Google, che restituisce un'immagine che poi inserisco nel sito con il tag .

```
<?php
$length = 10;
$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
$charactersLength = strlen($characters);
$randomString = '';
for ($i = 0; $i < $length; $i++) {
    $randomString .= $characters[rand(0, $charactersLength - 1)];
}
echo "<h1 style='color:green'>CODICE: ".$randomString."</h1>";
echo "<img
src='https://chart.googleapis.com/chart?chs=300x300&cht=qr&chl=http%3A%2F%2Ffilmuseodellamaturita.000webhostapp.com//qrcode.php?codice=".$randomString."&choe=UTF-8' title='Codice Biglietto' />";
$sql = "INSERT INTO codicerandom (codicerandom) VALUES ('".$randomString."')";
$result = $mysqli->query($sql);
...
```

Ho creato inoltre un evento che cancella ogni 24h i codici generati dalla tabella, questo perché non c'è bisogno di tenerli memorizzati e si evita di generare codici duplicati a lungo andare.

```
CREATE EVENT cancellacodici
ON SCHEDULE EVERY 24 HOUR
DO delete from codicerandom
WHERE TIMESTAMPDIFF(hour,
codicerandom.data, NOW()) >= 24;
```

Event name	cancellacodici
Status	ENABLED
Event type	RECURRING
Execute every	24 HOUR
Start	2020-06-04 19:12:35
End	
1 delete from codicerandom where TIMESTAMPDIFF(hour, codicerandom.data, NOW()) >= 24;	

I PRIVILEGI DEGLI UTENTI E DELL'AMMINISTRATORE



Quello che vede l'utente

Quello che vede l'amministratore

Per determinare se l'utente che ha fatto il login è admin oppure no, ho utilizzato una variabile di sessione che diventa 1 se dopo aver fatto il login nel database l'attributo "Admin" di quell'utente è uguale a 1, oppure 0 se l'attributo "Admin" di quell'utente è uguale a 0:

```
if($row['admin'] == '1') {  
    $_SESSION['admin'] = 1;  
} else {  
    $_SESSION['admin'] = 0;  
}
```

All'inizio di ogni pagina controllo se l'utente è admin oppure no prendendo questa variabile e di conseguenza visualizzo gli appositi bottoni e pagine utili a interfacciarsi con il Database per dare all'amministratore la possibilità di fare le operazioni CRUD come l'inserimento di un nuovo autore, di una nuova immagine, di una nuova corrente artistica, di una nuova opera e la modifica o eliminazione

```
$admin = $_SESSION['admin'];  
if($admin==0) {  
    ...  
} else if($admin==1) {  
    ...  
}
```

Per evitare accessi indesiderati a pagine protette, se un utente non ha i privilegi oppure non ha fatto il login, viene ridirezionato alla pagina iniziale:

```
if($_SESSION['admin']=="" ) {  
    echo "<script type='text/javascript'>  
window.top.location='https://ilmuseodellamaturita.000webhostapp.com/opere.php';  
</script>";  
}
```



Se l'utente non ha fatto il login oppure non ha un account non potrà visualizzare le opere presenti all'interno del museo, tutti possono accedere al sito web ma vedranno questa scritta quando andranno sulla sezione "opere", le altre sezioni del sito web sono invece libere e aperte a tutti.

LA CRIPTAZIONE DELLA PASSWORD

Sha-256 è una funzione dell'algoritmo Sha-2. Questo algoritmo accetta come input un messaggio di lunghezza massima 2^{64} e genera un hash a 256 bit. L'ho usato per sostituire la vecchia funzione hash md5. Sha-256 è unilaterale, quindi non si può trovare il testo in chiaro solo con l'hash. Per rubare una password bisogna confrontare questo hash con un database online e sperare che questo contenga la stessa password che stiamo cercando. Sha-256 è un buon modo per memorizzare le password degli utenti, poiché per esempio è molto più sicuro di Md5. Anche se è più sicuro, ho considerato l'uso di un "salt" per migliorare la sicurezza. Un salt è una stringa che si aggiunge alla password dell'utente per allungarla e aggiungerci nuovi caratteri. Ciò renderà molto più difficile la forza bruta e molto probabilmente la password non verrà trovata nei database online.

Il SALT del mio sito è la parola = "kappa"

```
<?php
...
$password = $_POST['password'];
$password = "". $password. "kappa";
$password = hash('sha256', $password);
...
?>
```

LA PAGINA PERSONALIZZATA

HAI FATTO IL LOGIN COME: marco

IL MUSEO DELLA MATURITA'

Progetto di: [nome] [cognome] [dataNascita] [dataMorte]

Il figlio dell'uomo



Descrizione:

L'opera affigura, in primo piano, un uomo il cui volto è nascosto quasi completamente da una mela verde sospesa in aria. Sullo sfondo è visibile un oceano sovrastato da un cielo nuvoloso. Riferendosi al dipinto, Magritte dichiarò: «Ebbene, qui abbiamo qualcosa di apparentemente visibile poiché la mela nasconde ciò che è nascosto e visibile allo stesso tempo, ovvero il volto della persona. Questo processo avviene infinitamente. Ogni cosa che noi vediamo ne nasconde un'altra; noi vogliamo sempre vedere quello che è nascosto da ciò che vediamo. Proviamo interesse in quello che è nascosto e in ciò che il visibile non ci mostra. Questo interesse può assumere la forma di un sentimento letteralmente intenso, un tipo di disputa, potrei dire, fra ciò che è nascosto e visibile e l'apparentemente visibile». Inoltre, l'opera esprime una forte critica rivolta alla classe borghese (simbologizzata dall'abito formale del soggetto), che il pittore reputava meschina/infelice e ipocrita (fingendo virtù, buone qualità, buoni sentimenti che non ha).

L'autore:

L'autore di questa opera è **René Magritte** di nazionalità **Francese**, nato il **1898-11-21** e deceduto il **1967-08-15**

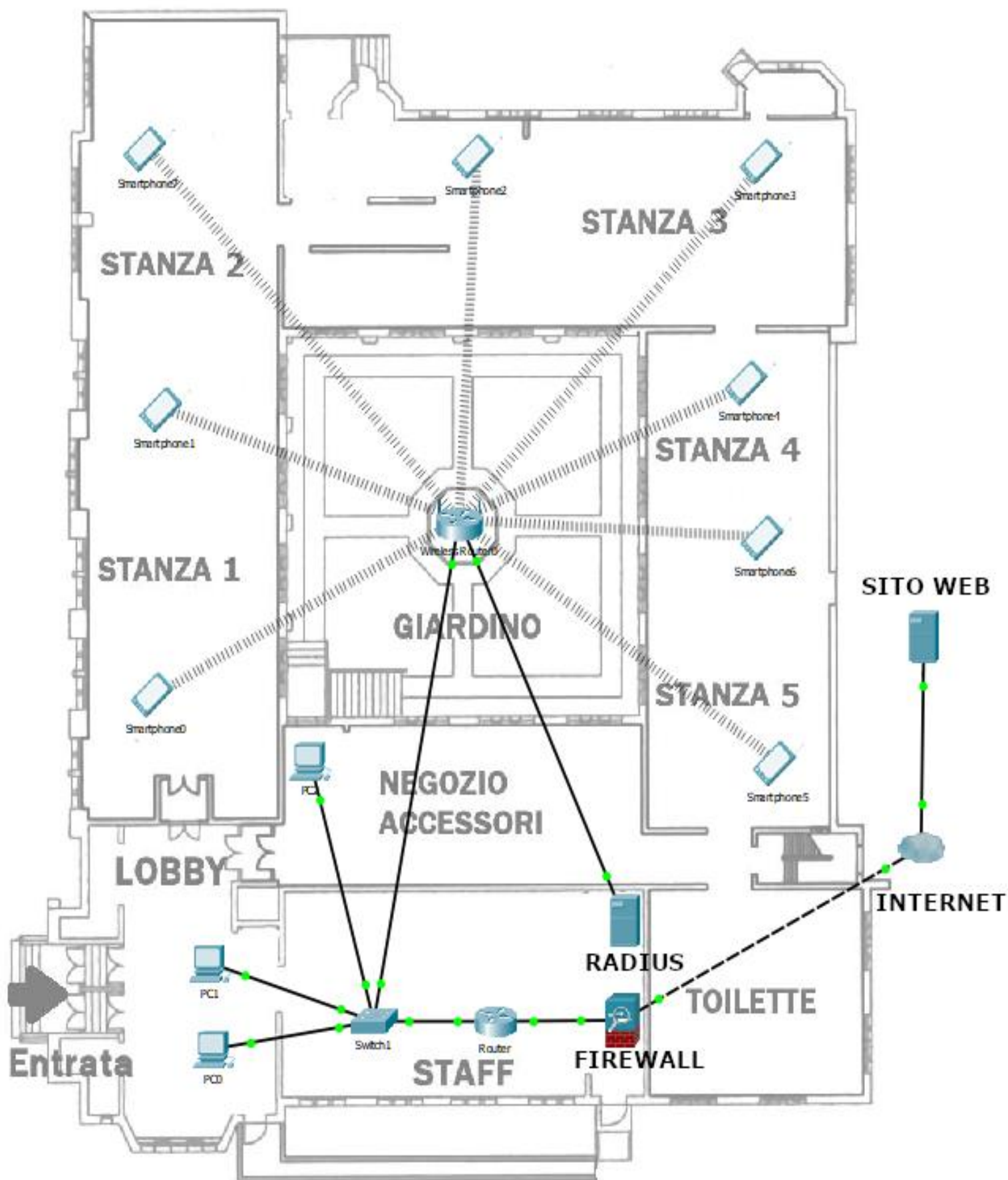
L'opera:

Questa pittura è stata realizzata nel **1964** e segue la corrente artistica del **Surrealismo**, ha una dimensione di **116 x 89** e la tecnica utilizzata è **olio su tela**. La si può trovare nella stanza **N° 1** del nostro museo.

Se l'utente ha fatto il login, potrà visualizzare anche una pagina personalizzata per ogni opera, questa pagina contiene dati presi dal Database come ad esempio l'immagine dell'opera, le informazioni dell'autore, dell'opera e la sua descrizione.

```
<?php
ecc...
$descrizione = $row['descrizione'];
$nome = $row['nome'];
$cognome = $row['cognome'];
$nazionalita = $row['nazionalita'];
$dataNascita = $row['dataNascita'];
$dataMorte = $row['dataMorte'];
echo "<p>".$descrizione."</p>";
echo "<h2
style='color:white;'>L'autore:</h2>";
echo "<p>L'autore di questa opera è
<span style='color:yellow;'>".$nome."
".$cognome."</span> di nazionalita
<span
style='color:yellow;'>".$nazionalita."
</span>, nato il <span
style='color:yellow;'>".$dataNascita."
</span> e deceduto il <span
style='color:yellow;'>".$dataMorte."
</span></p>";
ecc...
?>
```

LA RETE DEL MUSEO



LA CONNESSIONE AD INTERNET DEI VISITATORI

Invece del vecchio protocollo WEP nelle reti di oggi, si usano protocolli più sicuri come ad esempio il WPA e WPA2 perché forniscono funzioni di autenticazione e crittografia più affidabili. Poiché WPA2 è affidabile si potrebbe pensare di utilizzarlo nella rete del museo impostando la modalità WPA2 Personal. Questo si basa su un'autenticazione con chiave pre-condivisa e possiamo usare la crittografia TKIP o AES. La configurazione è semplice basta andare al menu Wireless dell'interfaccia Web del WirelessRouter, selezionando il menu Sicurezza wireless e selezionando WPA2 Personal come modalità di sicurezza ed inserendo una password, come ad esempio "passwordMuseo123" (è necessario utilizzare una password con una lunghezza minima di 8 caratteri). Ora sui client, dovremmo vedere la Sicurezza di WPA2-PSK e sarà necessario compilare il campo "password" con la stessa chiave per avere la connettività.

Questo metodo è adatto per ambienti domestici, ma in una rete aziendale dovremmo utilizzare un metodo più sicuro. Possiamo usare vari metodi per autenticare i nostri utenti. Il più comune usa il protocollo RADIUS, in cui viene utilizzato un server esterno per autenticare gli utenti. Per configurarlo bisognerà inserire un server nella topologia, collegandolo al WirelessRouter ed assegnandogli un indirizzo IP. Bisognerà poi configurare il servizio AAA(un acronimo di autenticazione, autorizzazione e affidabilità). Attiviamo il servizio e impostiamo i dati del client. In questo caso, il client sarà il WirelessRouter, gli forniamo un nome, un indirizzo IP e scegliamo una chiave pre-condivisa (RadiusPass) che verrà utilizzata tra il WirelessRouter e il server RADIUS. Creiamo poi un profilo utenti per i visitatori del museo, in modo che ogni utente possa disporre di nome utente e password per accedere alla rete.

Così facendo la connessione sarà autenticata dal server RADIUS e questo è più sicuro rispetto all'utilizzo di una chiave condivisa conosciuta da molte persone.

Per aumentare ulteriormente la sicurezza della rete la funzione di "Session Timeout" consente ad un amministratore di configurare il periodo di tempo in cui una determinata sessione può rimanere attiva. Il timeout della sessione può essere configurato sul server RADIUS in modo che ciascun client possa avere un valore di timeout diverso. L'attributo Session-Timeout è incluso nel messaggio di Access-Accept e imposta il numero massimo di secondi di servizio che può essere fornito all'utente prima della conclusione della sessione, nella mia rete un utente rimane connesso per al massimo 8 ore. Inoltre l'estensione del segnale del router è stato ridotto così si evitano connessioni da parte di persone che non si trovano nelle vicinanze del museo.

IL FIREWALL

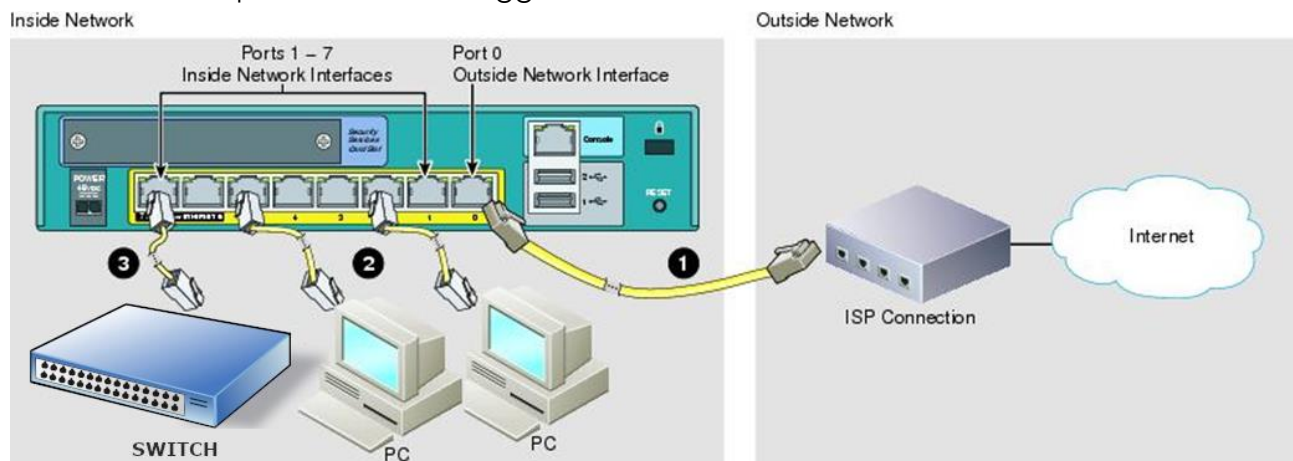
Il firewall regola e limita il traffico tra la rete interna e la rete esterna (internet), è un sistema hardware-software dedicato al filtraggio dei pacchetti ed è il punto di contatto della rete interna con quella esterna, solo il traffico autorizzato può attraversare il firewall. Il firewall che ho scelto per la mia rete funge da "network firewall", la sua caratteristica è che non ho bisogno di configurare ogni singolo host manualmente. Il modello è il "Cisco ASA5505" e filtra i pacchetti permettendo di bloccare o abilitare selettivamente il traffico che lo attraversa tramite le ACL perché è un "packet-filtering firewall".

Le ACL sono un insieme di condizioni di autorizzazione e negazione, che forniscono sicurezza bloccando gli utenti non autorizzati e consentendo agli utenti autorizzati di accedere a risorse specifiche. Normalmente le ACL risiedono in un router firewall.

Il Cisco ASA5505 utilizza i cosiddetti "livelli di sicurezza" che indicano l'attendibilità di un'interfaccia rispetto a un'altra. Un'interfaccia con un livello di sicurezza elevato può accedere ad un'interfaccia con un livello di sicurezza basso ma viceversa non è possibile a meno che non configuriamo una ACL che consenta questo traffico.

- Livello di sicurezza 0 : questo è il livello di sicurezza più basso presente sul firewall ed è assegnato all'interfaccia "esterna". Poiché non esiste un livello di sicurezza inferiore, ciò significa che il traffico dall'esterno non è in grado di raggiungere nessuna delle nostre interfacce a meno che non lo consentiamo in una ACL.
- Livello di sicurezza 100: questo è il livello di sicurezza più elevato sul firewall ed è assegnato all'interfaccia "interna", può raggiungere quindi tutte le altre interfacce.

Un "packet-filtering firewall" ha i vantaggi di essere trasparente cioè l'utente non si accorge della sua presenza, è veloce e immediato ed è "gateway-only" perciò non c'è bisogno di fare ulteriori configurazioni sugli host, per sicurezza però oltre a questo, sui PC di amministrazione cioè quelli che si trovano nella lobby del museo e quello del negozio degli accessori è installato anche un personal firewall cioè un programma software installato sui PC per avere una maggiore sicurezza.



esempio che mostra il funzionamento del firewall

CLOUD COMPUTING PER HOSTARE IL SITO WEB

Ho deciso di hostare il sito web sul cloud piuttosto che usare un server locale nella rete del museo, grazie ai vantaggi di questo:

- 1) Flessibilità: Grazie a una solida infrastruttura cloud, la potenza della macchina, sia essa fisica o virtuale, aumenta all'esigenza, senza che l'utente se ne accorga.
- 2) Disponibilità: Le risorse cloud sono sempre disponibili e possono essere attivate in pochi minuti. L'infrastruttura cloud è accessibile da remoto in qualunque momento, da qualsiasi luogo e qualunque dispositivo con connessione ad internet.
- 3) Sicurezza del sistema: Le infrastrutture cloud sono pensate per garantire al cliente la totale sicurezza, attraverso:
 - Sicurezza del Data Center
 - Backup e Disaster Recovery
- 4) Riduzione dei costi: Si paga solo ciò che si utilizza (pay-per-use): il Cloud è infatti cucito su misura per il cliente. La manutenzione dell'hardware richiede molte risorse economiche ed umane che solo le grandi multinazionali possono permettersi, affidarsi ad un Cloud porta all'abbattimento dei costi di gestione dell'hardware.

CONCLUSIONI

Generalmente sono abbastanza soddisfatto del prodotto realizzato, Il sito funziona bene, e offre tutte le principali funzionalità che mi ero imposto di implementare inizialmente questo è dovuto principalmente al fatto che ho iniziato a svilupparlo fin da subito. Si perde però a volte nei dettagli, questo è dovuto al poco tempo e la pressione del dover dedicare tempo a studiare anche le altre materie per l'esame orale.

- Il CSS è un po' sparso e poco organizzato, inizialmente avevo usato un file "style.css" che conteneva tutto il codice ma poi piano piano ho iniziato ad introdurre codice CSS anche all'interno del codice HTML per essere più veloce nella creazione e modifica della grafica.
Ora che la grafica è finita potrei spostare le modifiche sul file txt, ma è un processo lungo e macchinoso da fare per tutte le pagine del database
- Le immagini sono caricate su filesystem e non sul database, questo significa che se elimino un'opera l'immagine di questa rimane memorizzata e non viene eliminata. Inizialmente pensavo fosse una buona idea vista la semplicità di caricamento e lettura dei file, l'unico problema è che le immagini vanno chiamate con un nome specifico per essere individuate ad esempio:
"idOpera + 'icona.png'" per impostare l'immagine come icona dell'opera
"idOpera + '.png'" per impostare l'immagine come immagine di presentazione nella pagina personalizzata.
Questo si risolve mappando il percorso del file sul database oppure inserendola direttamente nel database su un campo blob. Oppure quando elimino un'opera dovrei chiamare una funzione javascript che elimina anche l'immagine dalla cartella oltre che i dati dell'opera dal database.
- Inizialmente ho messo l'attributo "tipo" all'interno delle entità figlie, cosa che è sbagliata e dovrei averlo messo nell'entità padre
- Non c'è un tasto "ricordami" per memorizzare i dettagli di login per un lungo periodo. Usando dei cookie si possono infatti memorizzare i dettagli dell'utente per giorni o mesi. All'inizio non avevo previsto questa funzione e mi sono accorto della sua mancanza soltanto alla fine dello sviluppo, integrare i cookie all'interno delle pagine di login al posto delle variabili di sessione avendo un progetto finito mi è sembra una mossa azzardata che mi avrebbe portato a complicazioni inutili e perdite di tempo
- Il codice inizialmente è stato sviluppato per funzionare online caricato sul Web Server di un host che offre questo servizio gratuitamente (000webhost.com), mi sono accorto però che i percorsi devono essere specificati come
" / paginaDiDestinazione.php" mentre su Xampp-Apache il percorso deve essere senza lo " / " davanti al percorso, questo significa che ho dovuto manualmente modificare tutti i percorsi di tutti i file per farlo funzionare anche in locale, questo è più un problema dell'ambiente di sviluppo piuttosto che mio

Per il futuro se riprenderò in mano questo progetto sistemerò questi problemi che non sono complicati, richiedono soltanto più tempo (circa una decina di ore).

Sono contento di aver realizzato questo progetto che mi ha dato la possibilità di creare da zero un sito web ed una rete che potrebbero esistere e funzionare veramente nella "vita reale". Forse con un preavviso maggiore avrei creato qualcosa di più complicato e completo ma sono pienamente soddisfatto di quello che sono riuscito a realizzare in questo tempo.

IONUT SUCIU AUGUSTIN