



iis
MarzottoLuzzatti
Valdagno

W

Sito Internet: www.iisvaldagno.it

E-Mail: viis022004@istruzione.it

E-Mail Certificata: viis022004@pec.istruzione.it



Classe V^a sez. D1 – sede ITI

Indirizzo: INFORMATICA E TELECOMUNICAZIONI

Articolazione: INFORMATICA

ESAME DI STATO A.S. 2020/2021

ELABORATO DI INFORMATICA E SISTEMI E RETI

Cognome e nome del candidato: Cornale Nicolò

Traccia

Si consideri la seguente realtà di riferimento: Un nuovo bar sta aprendo a Valdagno. Esso però, vuole differenziarsi dai classici bar volendo diventare un Internet Caffè. Il cliente deve poter prenotare un computer per un periodo di tempo limitato per giocare ai videogiochi e/o navigare in Internet. Ci sarà bisogno di una rete per i computer locali ed un sistema di prenotazione online. I clienti devono poter vedere, nel sito web, le proprie prenotazioni, gli orari disponibili e la bacheca delle news. Il proprietario, invece, deve riuscire anche a vedere la parte di amministrazione, dove verificare le varie prenotazioni e/o aggiungerne. Si progetti il servizio web e si descriva la rete interna ipotizzando una planimetria del ristorante.

Si specifichi, in particolare, come si intende dare ai clienti presenti in sala l'accesso al servizio anche in BYOD e come offrire, contemporaneamente, connessione ad Internet nel rispetto degli obblighi relativi alla privacy.

Il candidato, fatte le opportune ipotesi aggiuntive, sviluppi tutti i seguenti punti:

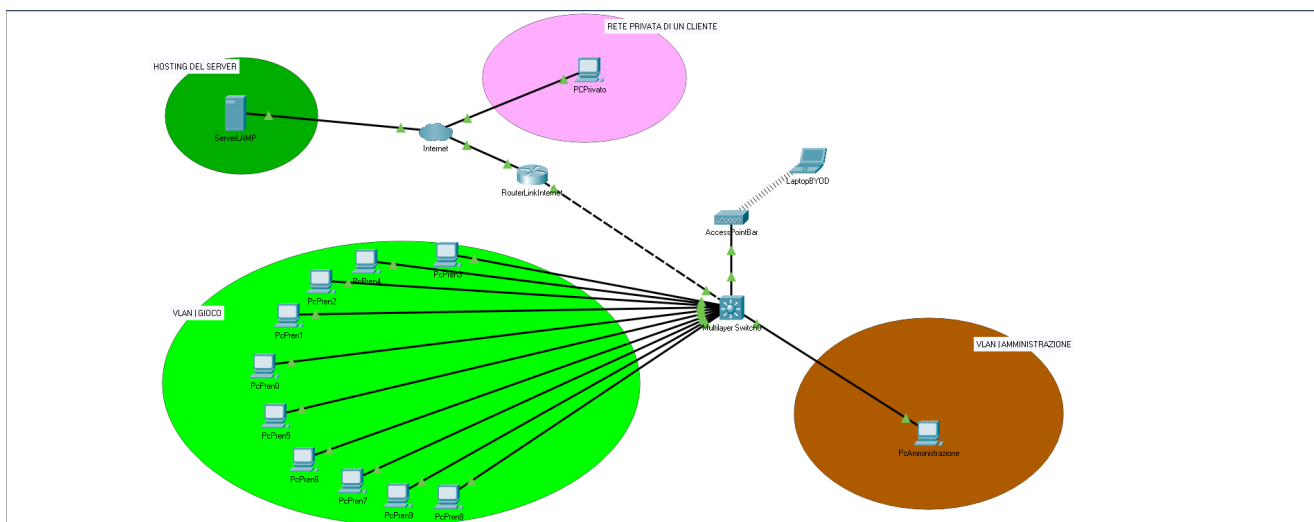
1. un'analisi della realtà di riferimento ipotizzata descrivendo, per la soluzione proposta:

- architettura di rete e caratteristiche dei sistemi server;
- modalità di comunicazione tra server e dispositivi, protocolli e servizi software per gestire la rete e fornire le pagine;
- gestione della sicurezza dei sistemi informatici realizzati o utilizzati;
- modello concettuale e logico del database;
- implementazione dello schema logico mediante linguaggio SQL.

2. Implementi una parte significativa del progetto sia per quanto riguarda l'applicazione web (o App per dispositivi mobili Android o iOS), sia per quanto riguarda la configurazione dei servizi e dispositivi di rete.

Architettura Di Rete e Modalità di Comunicazione

L'architettura di rete dell'elaborato può essere descritta efficacemente dallo schema qua sotto rappresentato con il programma "Cisco Packet Tracer":



La rete locale del negozio è una LAN, suddivisa a sua volta in 2 VLAN principali:

- La VLAN "Gioco", la quale è la principale poiché è dove sono installate le macchine.
- La VLAN "Amministrazione", che è quella dove il proprietario del locale o un suo delegato, possono controllare e registrare all'interno del sito i vari dati richiesti.

Poi, sempre all'interno del locale, è presente un access-point per permettere la comunicazione tra la rete locale e gli utenti BYOD (Bring Your Own Device, cioè che hanno portato uno strumento da casa) che sono presenti per usufruire del servizio.

Per connettersi alla rete ci sono due modi per gestire gli utenti. Il primo si basa su una pre-shared key, la quale sarebbe una password che il gestore del locale sceglierebbe da condividere con i propri clienti, in modo da permettergli un accesso alla mia rete tramite l'access point (ricordo che in questa modalità, la key sarebbe da cambiare almeno una volta a settimana, o si rischia che gente all'esterno del locale abbiano l'accesso al wi-fi).

La seconda opzione, invece, si basa sull'usare il computer amministratore come un server RADIUS per l'accounting. Questo permetterebbe di registrare l'utilizzo della rete in modo più efficace e automatico, però richiederebbe un costo maggiore e delle spese di manutenzioni eventuali.

Sono tutte e due valide scelte, lascerei la decisione al proprietario del locale.

Per quanto invece riguarda la connessione al server web su cui si basa il sito, esso è stato affidato ad un servizio esterno, poiché a mio avviso, il prezzo di mantenimento, acquisto e spesa energetica di un eventuale server in locale è potenzialmente molto più vasto di un affidamento su una macchina esterna, considerando anche la quantità di prestazioni che richiede il server, il quale è impostato come un server LAMP (Linux, Apache, MySQL, PHP, maggiori informazioni al link "<https://it.wikipedia.org/wiki/LAMP>"); è anche visibile, da questo schema, un'eventuale utente esterno alla rete locale, che potrà connettersi al server tramite una semplice connessione internet.

Riguardo invece ai protocolli utilizzati, quelli usati sono quelli dello stack TCP/IP. In particolare, viene usato HTTPS per la connessione sicura al server.

Si potrebbe rendere il progetto più ridondante e scalabile aggiungendo uno o due multilayer switch con le stesse impostazioni di quello centrale, in modo da aggiungere più fault-tolerance alla LAN. Essendo che comunque il caffè non si basa solo ed unicamente sulle macchine e ci sono diversi giorni liberi alla settimana in cui si possono sistemare eventuali problematiche, io non lo consiglierei, finché non si espande il business.

Il gateway posto prima dell'accesso ad internet sarebbe poi da configurare con i vari filtri a cui non si vuole fare accedere le macchine. Inoltre, il PC di Amministrazione è altamente consigliato farlo diventare un PDC (Primary Domain Controller), in modo da poter aver il totale controllo delle azioni svolte dall'utente che ha prenotato una delle macchine messe da noi a disposizione

Specifiche dei Computer

Le macchine che le persone useranno saranno in totale 11, le quali 10 sono per il noleggio, mentre una è dedicata per l'utente amministratore.

Per le macchine a noleggio il loro utilizzo è soprattutto ludico, quindi avere dei computer a basse prestazioni potrebbe influire sulla qualità del servizio che si offre e perciò sono richieste delle prestazioni medio/alte, mentre per il computer di amministrazione si può tranquillamente ragionare su una fascia medio bassa.

Per rispettare le richieste minime degli utenti e quindi offrire un servizio qualitativo e piacevole, proporrei di avere dell'hardware che è lo stesso per tutte le macchine di gioco, seguendo uno di queste tre proposte qua sotto elencate:

Fascia Media, prestazioni richieste soddisfatte, prezzo medio-basso (CONSIGLIATO):

- GPU di fascia media (Pixel Rate: 80.00 Gpixel/s, Texture Rate: 120 Gtexel/s, Shading Unit: 1200)
 - 4/8 GB di RAM
 - Processore fascia media (circa 3GHz, minimo dual-core (2 core), consigliato 4)
-

Fascia Medio-Alta, buone prestazioni, prezzo medio-alto:

- GPU di fascia medio-alta (Pixel Rate: 80.00 Gpixel/s, Texture Rate: 201.6 Gtexel/s, Shading Unit: 1920)
- 8/16 GB di RAM
- Processore fascia medio-alta (4.4 GHz clock, consigliato octa-core (8 core))

Fascia Alta, prestazioni praticamente impeccabili, prezzo alto (CON UN BUDGET MOLTO VASTO):

- GPU di fascia alta, ultima generazione delle schede
- 32/64 GB di RAM
- Processore fascia alta, ultima generazione delle schede

Per l'utente Amministratore va benissimo un computer di fascia medio bassa, potenzialmente avendo la possibilità di tralasciare anche la Scheda Grafica (GPU).

Gestione Della Sicurezza dei Sistemi Informatici

All'interno della LAN, le due VLAN sono tra loro regolate da delle ACL (Access-Control List), le quali svolgono la funzione di controllare tutte le comunicazioni che potrebbero avvenire tra una macchina prenotata e quella dell'utente amministratore, impedendo ogni sorta di comunicazione da parte della VLAN gioco, ma tralasciando invece la comunicazione della VLAN Amministrazione, la quale potrebbe in un futuro voler fare appoggio su qualche programma di monitoraggio per controllare il corretto uso dei computer noleggiati.

- Esempio Pratico -

VLAN Amministrazione → 192.168.1.0

VLAN Gioco → 192.168.10.0

configurazione della VLAN Gioco (usando Cisco PT):

```
access-list 10 deny 192.168.1.0 0.0.0.255
int vlan 10
ip access-group 10 out
```

configurazione della VLAN Amministrazione (usando Cisco PT):

```
access-list 20 deny 192.168.10.0 0.0.0.255
int vlan 10
ip access-group 10 in
```

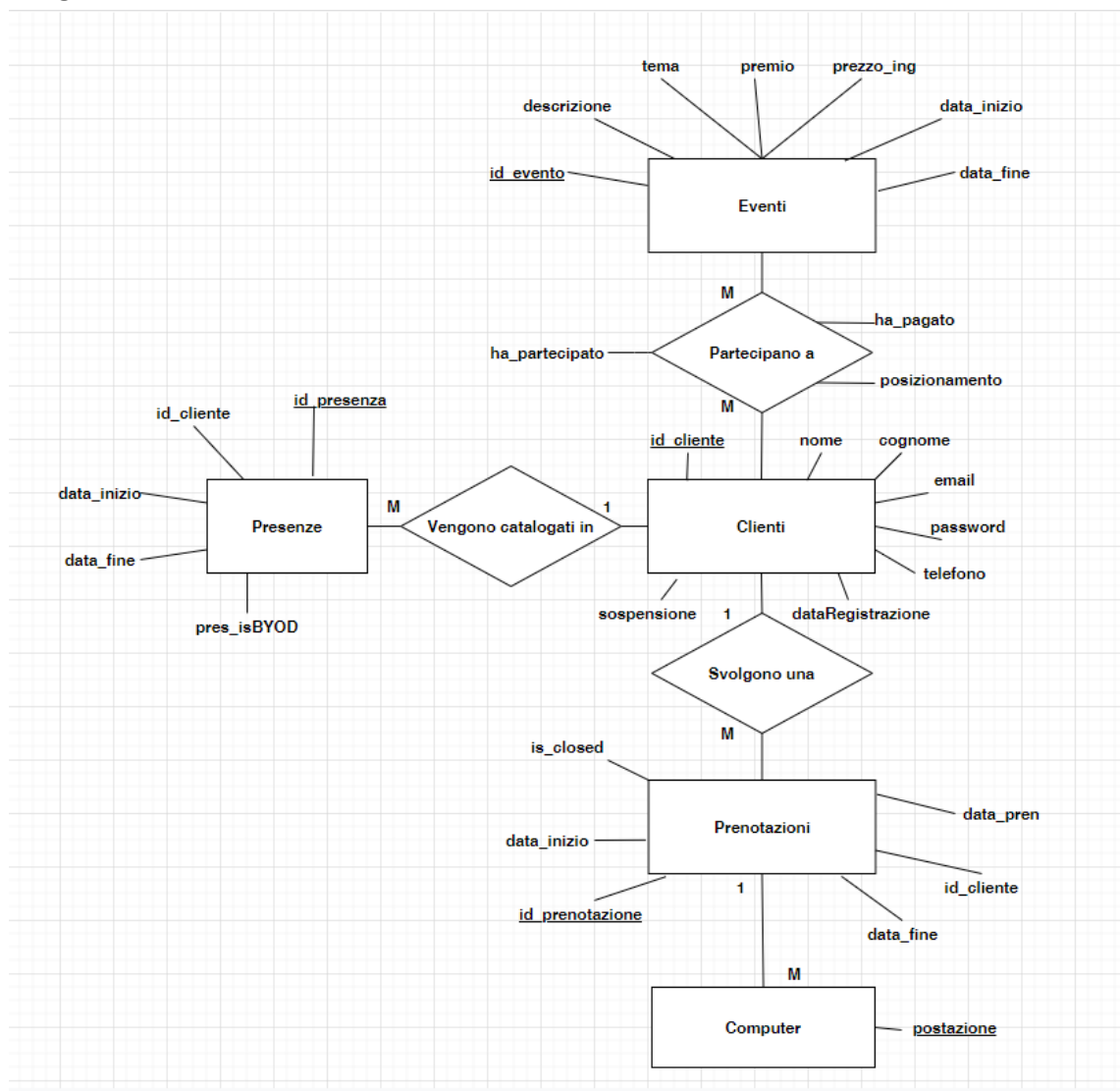
- Fine Esempio -

Per proteggere eventualmente da un attacco esterno, nel multilayer-switch è anche presente un firewall software. In caso si volesse aumentare ancora il livello di sicurezza, si potrebbe anche mettere una macchina firewall di rete per avere una sicurezza extra. Non lo ho inserito poiché è un comune un costo che credo possa essere parzialmente arginato o comunque extra per la realtà di riferimento.

Riguardo al server, le questioni relative alla sicurezza sono gestite dal gestore del servizio stesso, quindi eventuali backup e la ridondanza del sistema sono affidati a loro; sotto personale consiglio, però, visto il fatto che vengono conservati dei dati molto sensibili (come numero di telefono, data di nascita....), darei un'indicazione di effettuare un backup incrementale giornalmente alle 00:00, più un eventuale backup mensile svolto per la chiarificazione dei dati e un'eventuale corruzione dei dati precedentemente salvati, contenuto nella macchina server e anche in una macchina locale (preferibilmente quella dell'amministratore).

Modello Concettuale e Logico Del Database

- Diagramma ER -



Clients(id_cliente, INT; nome, STRING; cognome, STRING; email, STRING; password, STRING; telefono, STRING; dataRegistrazione, TIMESTAMP; sospensione, BOOLEAN);

{id_cliente} Primary Key

{email} Chiave Candidata basata sugli input degli utenti e sulle richieste del sito.

Presenze(id_presenza, INT; id_cliente, INT; data_inizio, TIMESTAMP; data_fine, TIMESTAMP; pres_isBYOD, BOOLEAN);

{Id_presenza} Primary Key

{id_cliente} Foreign Key riferita a {Clients}

Prenotazioni(id_prenotazione, INT; data_inizio, TIMESTAMP; data_fine, TIMESTAMP; is_closed, BOOLEAN; data_pren, TIMESTAMP; id_cliente, INT);

{id_prenotazione} Primary Key

{id_cliente} Foreign Key riferita a {Clients}

Computer(postazione, INT);

{postazione} Primary Key

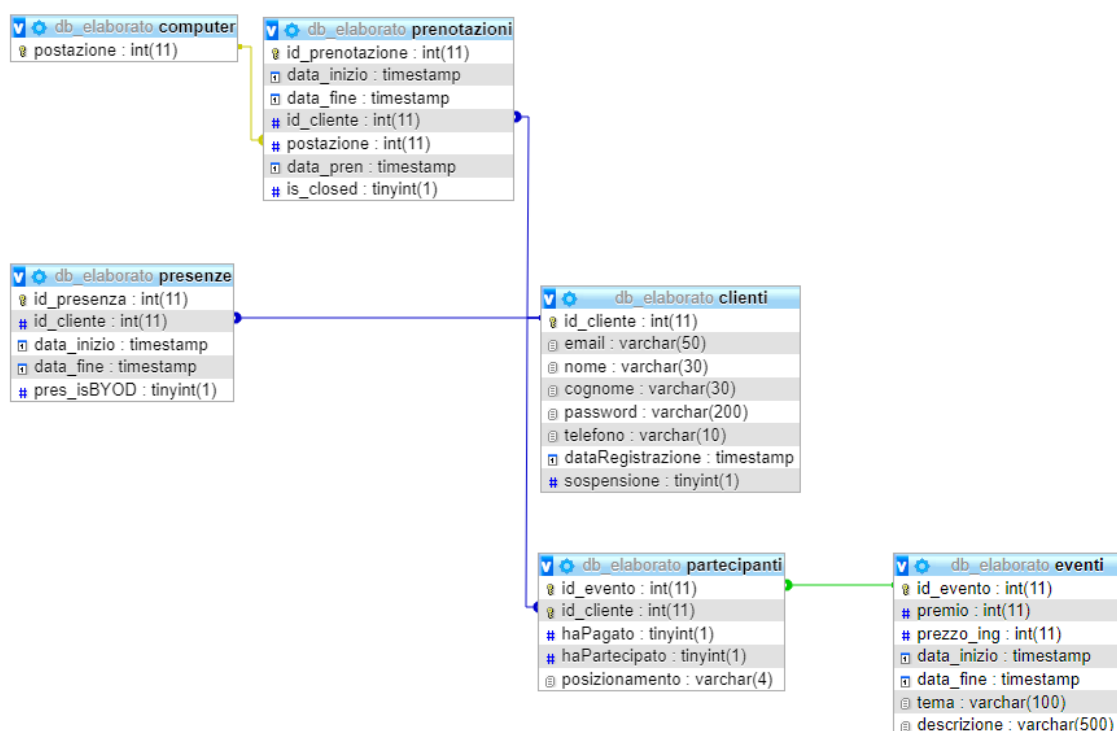
Eventi(id_evento, INT; descrizione, STRING; tema, STRING; premio, INT; prezzo_ing; data_inizio, TIMESTAMP; data_fine, TIMESTAMP);

{id_evento} Primary Key

Partecipanti(id_evento, INT; id_cliente, INT; haPagato, BOOLEAN; haPartecipato, BOOLEAN; posizionamento, INT);

{id_evento} Foreign Key riferita a {Eventi}

{id_cliente} Foreign Key riferita a {Clients}



Qui sopra si possono osservare anche le tabelle relazionali create dal designer di "phpmyadmin".
Questo è un grafico che spiega in maniera visiva quello che è stato fatto nelle info scritte sopra.
Lo scopo delle varie tabelle e i vari collegamenti è spiegato con dettaglio nei commenti dello schema SQL sottostante.

Descrizione del Database tramite "SQL"

In questa sezione sottostante si può osservare tutto il codice SQL che forma il database. Ho commentato in modo adeguato il codice per evitare fraintendimenti.

--- Creazione della tabella "Clienti" - Contiene tutti i dati di accesso di un utente, insieme a tutte le altre informazioni di cui il gestore ha bisogno;

```
create table CLIENTI
(
  id_cliente INT AUTO_INCREMENT NOT NULL primary key,
  email VARCHAR(50) NOT NULL,
  nome VARCHAR(30) NOT NULL,
  cognome VARCHAR(30) NOT NULL,
  password VARCHAR(200) NOT NULL,
  telefono VARCHAR(10) NOT NULL,
```

[illegible]

```
insert into COMPUTER values();
insert into COMPUTER values();
---
```

--- Creazione della tabella "Prenotazioni" - Essa contiene tutte le prenotazioni effettuate dai vari utenti. Essenziale per poter gestire con efficienza le cose fatte dagli utenti sul mio sito;

```
---
```

```
create table PRENOTAZIONI(
    id_prenotazione INT AUTO_INCREMENT NOT NULL primary key,
    data_inizio TIMESTAMP DEFAULT NOW(),
    data_fine TIMESTAMP DEFAULT NOW(),
    id_cliente INT NOT NULL,
    postazione INT NOT NULL,
    data_pren TIMESTAMP DEFAULT NOW(),
    is_closed BOOLEAN,

    foreign key(id_cliente) references CLIENTI (id_cliente),
    foreign key(postazione) references COMPUTER (postazione)
);
---
```

--- Creazione della tabella "Presenze" - Contiene tutte le presenze, le quali saranno inserite manualmente dal pannello di amministrazione, oppure confermate seguendo i dati della tabella prenotazioni;

```
---
```

```
create table PRESENZE(
    id_presenza INT AUTO_INCREMENT NOT NULL primary key,
```

```
id_cliente INT NOT NULL,
data_inizio TIMESTAMP NOT NULL DEFAULT NOW(),
data_fine TIMESTAMP NOT NULL DEFAULT NOW(),
pres_isBYOD BOOLEAN NOT NULL,
foreign key(id_cliente) references CLIENTI (id_cliente)
);
---
--- Creazione della tabella "Eventi" - Contiene gli Eventi, anche questi
inseriti manualmente dal pannello di amministrazione. Servono soprattutto
per attirare eventuali clienti nel caffè, infatti ci sono premi in denaro
o comunque premi utili;
---
create table EVENTI(
    id_evento INT AUTO_INCREMENT NOT NULL primary key,
    premio INT NOT NULL,
    prezzo_ing INT,
    data_inizio TIMESTAMP NOT NULL DEFAULT NOW(),
    data_fine TIMESTAMP NOT NULL DEFAULT NOW(),
    tema VARCHAR(100),
    descrizione VARCHAR(500)
);
---
--- Crea la tabella di raccordo "Partecipanti" - Essa serve per regolare
la relazione Molti a Molti della tabella "Eventi" - "Clienti" ;
---
create table PARTECIPANTI(
```

```
id_evento INT NOT NULL,  
id_cliente INT NOT NULL,  
haPagato BOOLEAN,  
haPartecipato BOOLEAN,  
posizionamento VARCHAR(4),  
primary key(id_evento, id_cliente),  
foreign key (id_evento) references EVENTI (id_evento),  
foreign key (id_cliente) references CLIENTI (id_cliente)  
);
```

Implementazione Significativa di una parte di Codice

Come fatto precedentemente, il codice è commentato, quindi non mi dilungherò troppo a spiegare, poiché è commentato in se.

Il codice che allego qua sotto è quello che gestisce le prenotazioni al sito. Esso è fatto principalmente in PHP, essendo che interagisce col database e ha bisogno di fare delle query di domanda per carpire delle informazioni.

```
<!-- pagina usata per permettere ad un utente di fare una prenotazione -->  
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">  
  <link rel="stylesheet" href="https://cdn.datatables.net/1.10.24/css/  
jquery.dataTables.min.css">  
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></  
script>  
  <script src="https://cdn.datatables.net/1.10.24/js/jquery.dataTables.min.js"></  
script>  
  <script src="help.js"></script>
```

```
<link rel="stylesheet" href="generalStyles.css">
<title>Coffee And Games - Prenota</title>
</head>

<body class="w3-sand">
  <!-- login standard header -->
  <?php require("loginHeader.php");?>
  <div>
    <a style="border:1px solid black;" type="button" class="w3-button" href="index.php">Ritorna Alla Home</a>
    <a style="margin-right: 10px;border:1px solid black;" type="button" class="w3-button" href="logout.php">Logout</a>
  </div>
</header>

<!-- contenuto principale della pagina -->
<hr style="border: 1px solid black;">
<div class="contenitore">
  <h1 style="margin-left: 10px;">
    Prenota Una Postazione
  </h1>
  <h3 style="margin-left: 10px;">
    Assicuriamo la nostra disponibilità!
  </h3>
  <div id="errore">
  </div>
  <!-- form con tutti gli input che l'utente ha da inserire -->
  <form id="registra" action="" method="POST" style="margin-left: 10px; margin-top: 50px;">
    <div>
      <label style="margin-top: 50px;" class="w3-label">Data Iniziale</label>
      <input id="data" class="w3-text" type="date" name="data" placeholder="Inserisci una Data" required>
      <label style="margin-top: 50px;" class="w3-label">Ora Inizio</label>
      <input id="ora" class="w3-text" type="time" name="ora" placeholder="Inserisci L'Ora" required>
    </div>
    <div style="margin-top: 50px;">
      <label style="margin-top: 50px;" class="w3-label">Data Finale</label>
      <input id="dataFine" class="w3-text" type="date" name="dataFine" placeholder="Inserisci una Data" required>
    </div>
  </form>
</div>
```

```
<label style="margin-top: 50px;" class="w3-label">Ora Fine</label>
<input id="oraFine" class="w3-text" type="time" name="oraFine" placeholder="Inserisci L'Ora" required>
</div>

<input type="submit" value="Effettua la Prenotazione" class="w3-button" style="border:1px solid black; margin-top: 50px;">
</form>

<?php

//vari controlli per assicurarsi che i dati inseriti siano coerenti
if(!isset($_POST["data"]) && !isset($_POST["ora"]) && !isset($_POST["dataFine"]) && !isset($_POST["oraFine"])){
    return;
}

//sanitizzazione degli input
$dataInizio = filter_var($_POST["data"], FILTER_SANITIZE_STRING);
$dataFine = filter_var($_POST["dataFine"], FILTER_SANITIZE_STRING);
$oraInizio = filter_var($_POST["ora"],FILTER_SANITIZE_STRING);
$oraFine = filter_var($_POST["oraFine"],FILTER_SANITIZE_STRING);

//comparazione della data di inizio con la data odierna (essendo che non si possono fare prenotazioni nel passato)
if(strcmp($dataInizio,date("Y-m-d")) < 0){
    echo '<h5 style="margin-top: 50px;">Non hai Inserito una Data Iniziale Adatta!</h5>';
    return;
}

//comparazione della data di inizio con la data di fine (essendo che non puoi finire la prenotazione prima di iniziandola)
if(strcmp($dataFine,$dataInizio) < 0){
    echo '<h5 style="margin-top: 50px;">Non hai Inserito una Data Finale Adatta!</h5>';
    return;
}

$timestamp = strtotime($dataInizio);
```

```
$day = date('l', $timestamp);

//comparazione della data con le stringhe 'Monday', 'Wednesday', per
assicurarsi che non siano uguali (essendo che il Lunedì e il Mercoledì il Locale è
chiuso
if(strcmp($day, 'Monday') == 0 || strcmp($day, 'Wednesday') == 0){
    echo '<h5 style="margin-top: 50px;">Spiacente, siamo chiusi il Lunedì e Mercoledì</h5>';
    return;
}

//in caso il giorno sia lo stesso, controlla che l'ora di fine non sia minore
dell'ora di inizio (sempre perché non si può finire la prenotazione nel passato)
if(strcmp($oraInizio, $oraFine) > 0 && strcmp($dataInizio, $dataFine) == 0){
    echo '<h5 style="margin-top: 50px;">Hai selezionato un\'ora di Inizio maggiore
dell\'ora di Fine</h5>';
    return;
}

$timestamp2 = strtotime($dataFine);

$day = date('l', $timestamp2);

//comparazione della data con le stringhe 'Monday', 'Wednesday', per assicurarsi
che non siano uguali (essendo che il Lunedì e il Mercoledì il Locale è chiuso).
Questa volta per la data di fine
if(strcmp($day, 'Monday') == 0 || strcmp($day, 'Wednesday') == 0){
    echo '<h5 style="margin-top: 50px;">Spiacente, siamo chiusi il Lunedì e Mercoledì</h5>';
    return;
}

//controlla se nella fascia di orario c'è un qualche computer libero
if(($appoggio = checkIfPcAvailable($dataInizio, $oraInizio, $dataFine, $oraFine)
) == NULL){
    echo '<h5 style="margin-top: 50px;">Spiacente, per questa ora tutte le postazioni
sono Prenotate</h5>';
    return;
}

//inserisce la prenotazione nel database
```

```
insertPren($dataInizio,$oraInizio,$dataFine,$oraFine,$appoggio);

?>
</div>
</body>
</html>

//funzione che controlla se i computer sono disponibili.
function checkIfPcAvaialbe($data,$ora,$dataF,$oraF){

    require("config.php");
    $mysqli = new mysqli($host, $username, $password, $db_name);
    $mysqli->set_charset("utf8");
    $dataInizio = $data." ".$ora;
    $dataFine = $dataF." ".$oraF;
    //query che ritorna la prima postazione libera dato un time-slice
    $risultato = $mysqli->query("select postazione from COMPUTER where postazione NOT
IN(select postazione from PRENOTAZIONI where ((data_inizio >= '$dataInizio' AND dat
a_inizio < '$dataFine') OR (data_fine <= '$dataFine' AND data_fine >= '$dataInizio'
)) OR (data_inizio < '$dataInizio' AND data_fine > '$dataFine')) order by postazion
e limit 1");

    $ritorno = ($risultato -> fetch_row())[0];
    $risultato->close();
    $mysqli->close();

    return $ritorno;

}

//inserisce la prenotazione nel database
function insertPren($dataInizio,$oraInizio,$dataFine,$oraFine,$postazione){
    require("config.php");
    //crea gli input timestamp, in modo da poterli inserire correttamente nel
database
    $id_cliente = getIdBySession();
    $dataInizioC = $dataInizio." ".$oraInizio;
    $dataFineC = $dataFine." ".$oraFine;
    $mysqli = new mysqli($host, $username, $password, $db_name);

    if (!$mysqli->connect_error)
```

```
{  
    $mysqli->set_charset("utf8");  
  
    $sql = "insert into PRENOTAZIONI (data_inizio,data_fine,id_cliente,postazione,d  
ata_pren,is_closed) values ('$dataInizioC','$dataFineC','$id_cliente','$postazione'  
, NOW(), FALSE)";  
  
    $mysqli->query($sql);  
  
    echo $dataInizioC."|".$dataFineC."|".$id_cliente."|".$postazione;  
  
    $mysqli->close();  
  
}  
  
header("Location: endPrenotazione.html");  
}
```

Il Progetto In Se e Cosa si può Aggiungere

Il Progetto in sé è disponibile al 100% su GitHub all'indirizzo

- <https://github.com/Kami-Yama/Elaborato2k21Informatica>

Allego anche la documentazione, la quale potete sempre trovare su GitHub.

Parlando delle varie aggiunte che si potrebbero fare, ce ne sono diverse, ma che sarebbero da discutere col gestore del locale.

1 Aggiunta

Permettere nel menu delle azioni di vedere gli eventi che sono in corso e a cui ci si può iscrivere. Questo permetterebbe di automatizzare anche le iscrizioni, ma si mette a rischio anche di inserire persone che non centrano nulla, essendo online. Personalmente metterei una possibilità di visionare gli eventi disponibili, ma mantenendo le iscrizioni agli stessi tramite un contatto al negozio, per evitare spiacevoli inconvenienti.

2 Aggiunta

Aggiungere le varie bevande etc... nella parte informatica. Essendo che comunque è un "internet CAFFE", ci dovrebbe essere anche la possibilità di comprare delle bevande o degli snack. Non è impensabile di aggiungere una tabella con gli snack e le bevande, per fare vedere anche online il menu ai clienti.

Questa sotto invece, come accennato prima, è la documentazione del sito. Essa è scritta in inglese poiché essendo una lingua internazionale, rende di più facile comprensione la documentazione.

Documentation of The Site

PS: The visuals of the site AREN'T DEVELOPED! This is just a prototype of the site just for making the client understand how it will work.

`index.html`: Home of the site. In here you can start seeing the various buttons and interact with the site.

`img`: Folder where images are held.

`azione.php`: Php page where the user can choose their action. They will be able to logout from their account or to start a prenotation for a date in the future.

`config.php`: Help cheatsheet for logging in as a root user in the DB.

`dbCrea.sql`: Code for creating the MySQL Database.

`endRegister.html`: Page that redirects you to the `index.html`.

`functions.php`: PHP Scripts which interacts with the database and sends data across all the scripts.

`help.js`: JavaScript Scripts which are helpful for the correct usage of the site.

`login.php`: PHP page used for the users to Login in their accounts.

`registrazione.php`: PHP page used to create an account. All the inputs are correctly filtered, so it's impossible to use SQL Injections or other mechanisms for hacking the site.

`logout.php`: Function for logging out.

`prenota.php`: Page of related to the prenotations of the site.

`endPrenotazione.html`: Redirect page at the ending of a prenotation.

`rules.php`: simple page that could eventually be filled with the rules of the caffè.

`about.php`: simple page that could eventually be filled with the info about the managers of the caffè.

`generalStyles.css`: general collection of the styles used in all the pages.

`loginHeader.php`: login header that can be used wherever you want, used to recycle a lot of code.

What do i have to Know About the Code

This specific code has been done without caring for the aesthetics of the site. In fact, i am a Developer, not a Designer. I inserted only the basic aesthetics to not make the site figure out so bad. It's normal to find some strange things, some elements put in the wrong order etc... But the important thing it's to see the work done behind and the various scripts and queries done with the database and the relative interaction that it has led to. I repeat it once more, it is not the work of a designer, but of a Developer.
