



**Mai più liti in classe
tra professori
e alunni**

**E' arrivato
SchoolSocial**



Alessandro Maggio
1996-01-09
Via Nino Bixio 9
alex.tcd.alex@gmail.com
5B Informatica
ITI e LSSA
'N Copernico'
Alunno

**Grafica semplice ed
intuitiva**

**Impostazioni facili
e veloci**

Dati di nascita: 09/01/1996
Seleziona una regione: Lazio
Seleziona una provincia: Latina
Seleziona un comune: 4021 - Castelforte
Via: Via Nino Bixio 9
E-Mail: alex.tcd.alex@gmail.com
Ruolo: Alunno
Seleziona la scuola: ITI e LSSA "N Copernico"
Seleziona l'indirizzo: Informatica
Seleziona la sezione: B
Seleziona la classe: 5
Perfetto, puoi salvare le modifiche.
Salva
Seleziona l'avatar dal tuo PC: Scegli file Nessun file selezionato
Salva
Eccella password:
La password non coincide con quella attuale.
Nuova password:
Conferma password:
Inserisci la password in entrambi i campi.
Salva

Francesco Maggio
1994-04-10
NuovaVia
francesco@gmail.com
IT Turistico I.T.C.G.T. "E Fermi" Altro...
Docente

Alessandro Maggio
1996-01-09
Via Nino Bixio 9
alex.tcd.alex@gmail.com
5B Informatica
ITI e LSSA
'N Copernico'
Alunno

Via Roma 250
all@aaamil.it
Tefono: 0909797980 | Fax: 77777777
<http://www.istitutocopernico.gov.it/>

Alessandro Maggio
1996-01-09
Via Nino Bixio 9
alex.tcd.alex@gmail.com
5B Informatica
ITI e LSSA
'N Copernico'
Alunno

**Integrazione a 360°
tra amici e gruppi
classe**



Powered by Alessandro Maggio

È scientificamente provato che i ragazzi d'oggi trovano noiose le lezioni scolastiche e preferiscono passare la maggior parte del loro tempo sui social network o giocare con il proprio smartphone. I professori invece, imbestialiti da questa situazione onorano i ragazzi con molteplici discorsi filosofici nella quale continuano a ripetere che non esiste più vita sociale, che oramai si è dipendenti dal cellulare, dai mezzi di comunicazione online, etc. Proprio per questo nasce **SchoolSocial**, un Social Network per le scuole, un diario virtuale dove è possibile condividere lezioni, compiti e quant'altro. Offre, inoltre, la possibilità di inviare richieste d'amicizia, visualizzare i profili, cambiare il proprio avatar, scrivere nelle bacheche altrui e altre funzione principali. A renderlo particolare è l'automatizzazione del gruppo classe; conclusa la registrazione il Social Network chiederà all'utente il ruolo che compie (Docente, Alunno) e la classe di appartenenza legato alla sezione, indirizzo e scuola per facilitare la condivisioni di materiale didattico tra alunni e prof.

Attori e relative funzioni:

- **Alunno:**
 - Inviare richieste d'amicizie a qualunque utente.
 - Gestire le varie richieste d'amicizia.
 - Scrivere sulla bacheca degli amici.
 - Scrivere sulla bacheca personale.
 - Registrarsi all'interno di una determinata classe, filtrando scuola e specializzazione.
 - Scrivere sul gruppo classe alla quale si è registrati.
 - Modificare i propri dati anagrafici.
 - Modificare il proprio avatar.
 - Aggiungere nuove classi, scuole, specializzazioni e sezioni nel caso in cui non siano già state inserite da un altro utente.
- **Docente:**
 - Inviare richieste d'amicizie a qualunque utente.
 - Gestire le varie richieste d'amicizia.
 - Scrivere sulla bacheca degli amici.
 - Scrivere sulla bacheca personale.
 - Registrarsi all'interno di una o più determinate classi, filtrando scuole e specializzazioni.
 - Scrivere nel gruppi classe alla quale si è registrati.
 - Modificare i propri dati anagrafici.
 - Modificare il proprio avatar.
 - Aggiungere nuove classi, scuole, specializzazioni e sezioni nel caso in cui non siano già state inserite da un altro utente.

Contenuti:

Nella home page del sito è posto centralmente il logo di quest'ultimo e in basso il menù contenente un piccolo trailer, il box per il login e per la registrazione. Dopo aver effettuato il primo accesso si verrà automaticamente reindirizzati alla pagina contenente le impostazioni per completare tutti i dati anagrafici e informazioni obbligatori mancanti. Nei prossimi accessi invece il sito reindirizza automaticamente alla pagina home costituita dalla parte destra dedicata all'utente che ha effettuato l'accesso, contenente l'avatar, il nome e cognome, la data di nascita, la via di residenza (con link automatico a Google Maps), l'email, il gruppo di appartenenza (nel caso



l'utente sia un docente verrà visualizzato solamente il primo gruppo alla quale si è registrati e di seguito la voce "Altro...") e il tipo di ruolo che svolge. Nella barra in alto è presente il logo con collegamento alla home, l'icona con conteggio delle richieste d'amicizie, l'icona con conteggio delle notifiche ricevute, la barra di ricerca, il tasto per il log out e il tasto per le impostazioni (la grafica descritta fin ora sarà il template standard dell'intero sito). Nella parte centrale invece è presente il box per condividere una qualunque notizia sul proprio profilo e i vari post scritti da amici o per gli amici e scritti all'interno del gruppo classe di appartenenza. Nelle pagine user e gruppoclasse invece è possibile visualizzare le varie informazioni o dati anagrafici riguardanti il gruppo classe o l'utente selezionato, leggere i vari post e nel caso degli utenti inoltrare una richiesta d'amicizia. Poi sono presenti le varie pagine per la creazione del gruppo classe, gestire le richieste d'amicizia, leggere le notifiche e per rendere SchoolSocial facile ed intuitivo il più possibile.

Le entità che possono essere individuate nel problema sono:

- **Utente**, per rappresentare le informazioni riguardanti gli iscritti al sito.
- **Post**, per i vari post scritti dagli utenti per gli utenti o per i gruppi classe.
- **Ruolo**, per i diversi tipi di ruoli all'interno della scuola (Docente, Alunno).
- **Scuola**, per individuare l'istituto di appartenenza degli utenti.
- **Classe**, per individuare classe di appartenenza degli utenti.
- **Sezione**, per individuare la sezione di appartenenza degli utenti.
- **Indirizzo**, per individuare l'indirizzo specialistico di appartenenza degli utenti
- **Comune**, per individuare il comune di residenza degli utenti o delle scuole.
- **Provincia**, per individuare la provincia di residenza degli utenti o delle scuole.
- **Regione**, per individuare la regione di residenza degli utenti o delle scuole.

Gli attributi di **Utente** sono: id_utente, nome, cognome, data_nascita, via, mail, password, avatar, ultimo_accesso.

Gli attributi di **Post** sono: id_post, data_ora, contenuto.

Gli attributi di **Ruolo** sono: id_ruolo, tipo.

Gli attributi di **Scuola** sono: id_scuola, nome, telefono, fax, mail, sitoweb, via.

Gli attributi di **Classe** sono: id_classe, classe.

Gli attributi di **Sezione** sono: id_sezione, nome_sezione.

Gli attributi di **Indirizzo** sono: id_indirizzo, indirizzo.

Gli attributi di **Comune** sono: id_com, cap, comune.

Gli attributi di **Provincia** sono: id_pro, nome_provincia.

Gli attributi di **Regione** sono: id_reg, nome_regione.



Tra l'entità *Utente* e l'entità *Post* esiste un'associazione uno a molti perché un utente può pubblicare più post e ogni post appartiene a un unico utente.

Tra l'entità *Classe* e l'entità *Post* esiste un'associazione uno a molti perché una classe può avere più post ma un post può avere una sola classe.

Tra l'entità *Utente* e l'entità *Ruolo* esiste un'associazione molti a uno perché un utente può avere un solo ruolo e lo stesso ruolo po' essere svolto da più utenti.

Tra l'entità *Utente* e l'entità *Utente* esiste un'associazione molti a molti perché un utente può avere più utenti amici e vice versa.

Tra l'entità *Utente* e l'entità *Classe* esiste un'associazioni molti a molti perché un utente può essere presente in più classi e una classe può avere più utenti.

Tra l'entità *Utente* e l'entità *Comune* esiste un'associazione molti a uno perché un utente può risiedere in un solo comune e in un comune possono risiedere più utenti.

Tra l'entità *Utente* e l'entità *Provincia* esiste un'associazione molti a uno perché un utente può risiedere in una sola provincia e in una provincia possono risiedere più utenti.

Tra l'entità *Utente* e l'entità *Regione* esiste un'associazione molti a uno perché un utente può risiedere in una sola regione e in una regione possono risiedere più utenti.

Tra l'entità *Scuola* e l'entità *Comune* esiste un'associazione molti a uno perché una scuola può risiedere in un solo comune e in un comune possono risiedere più scuole.

Tra l'entità *Scuola* e l'entità *Provincia* esiste un'associazione molti a uno perché una scuola può risiedere in una sola provincia e in una provincia possono risiedere più scuole.

Tra l'entità *Scuola* e l'entità *Regione* esiste un'associazione molti a uno perché una scuola può risiedere in una sola regione e in una regione possono risiedere più scuole.

Tra l'entità *Indirizzo* e l'entità *Scuola* esiste un'associazione molti a uno perché una scuola può avere più indirizzo ma lo stesso indirizzo può essere presente in una sola scuola.

Tra l'entità *Sezione* e l'entità *Indirizzo* esiste un'associazione molti a uno perché un indirizzo può avere più sezione ma una determinata sezione può avere un solo indirizzo.

Tra l'entità *Classe* e l'entità *Sezione* esiste un'associazione molti a uno perché una sezione può avere più classi e una determinata classe può avere una sola sezione.

Tra l'entità *Provincia* e l'entità *Regione* esiste un'associazione molti a uno perché una regione può avere più provincie ma una provincia può avere una sola regione.

Tra l'entità *Comune* e l'entità *Provincia* esiste un'associazione molti a uno perché una provincia può avere più comuni e una comune può avere una sola provincia.



Sulla base dell'analisi effettuata si può disegnare il modello Entità/Associazioni:

MODELLO E/R

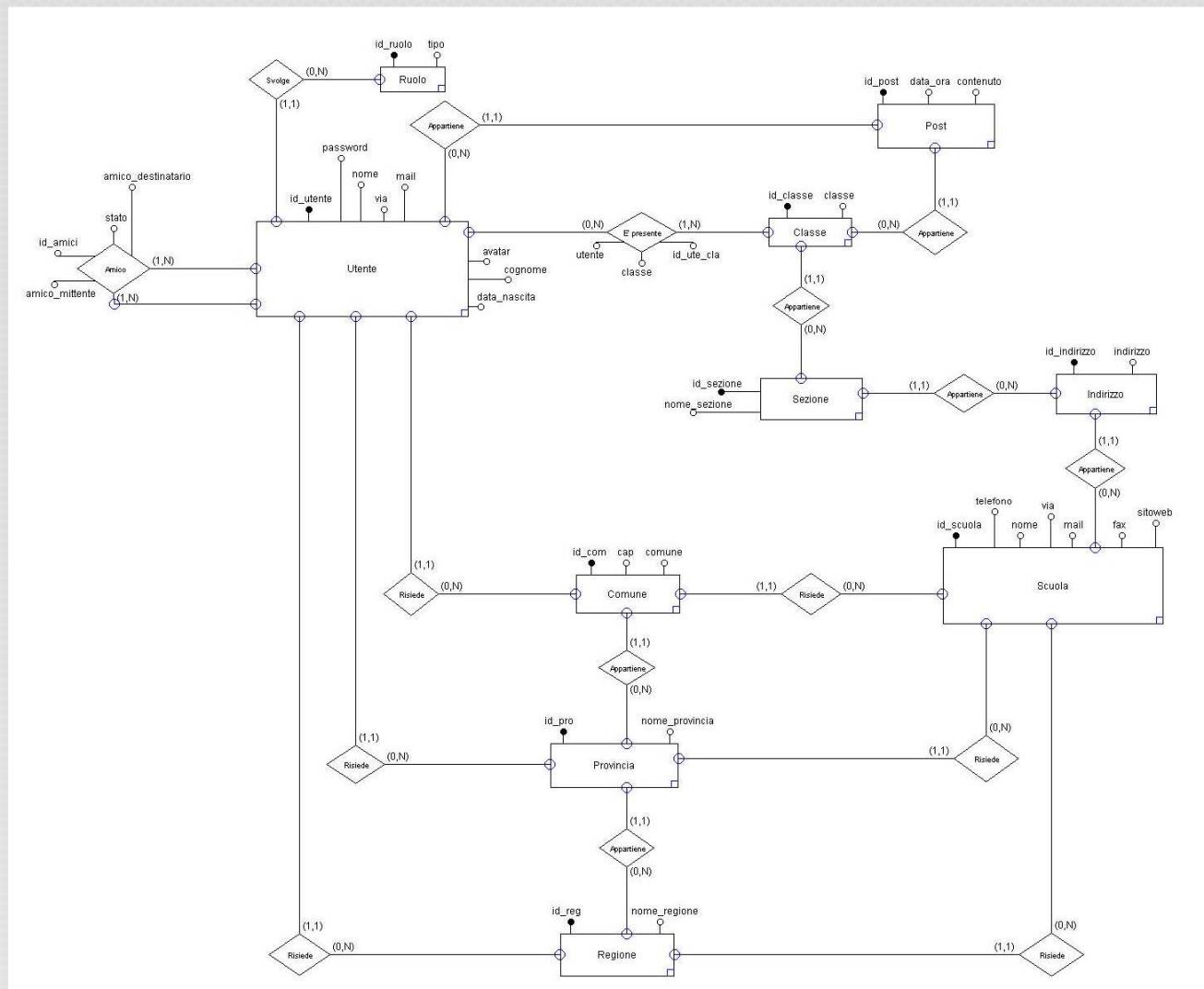


Tabella	Campo	Chiave	Formato	Dimensione
Utenti	id_utente	PK (Primaria)	Numerico	11
	nome		Carattere	30
	cognome		Carattere	30
	data_nascita		Data	
	via		Carattere	30
	regione	FK (Esterna)	Numerico	11
	provincia	FK (Esterna)	Numerico	11
	comune	FK (Esterna)	Numerico	11
	mail		Carattere	30
	password		Carattere	30
	avatar		Carattere	30
Post	ruolo	FK (Esterna)	Numerico	11
	ultimo_accesso		Data/Time	
	id_post	PK (Primaria)	Numerico	11
	data_ora		Data/Time	
	contenuto		Testo	
	scritto_per_utente	FK (Esterna)	Numerico	11



	scritto_per_classe	FK (Esterna)	Numerico	11
	scritto_da_utente	FK (Esterna)	Numerico	11
Ruoli	id_ruolo	PK (Primaria)	Numerico	11
	tipo		Carattere	30
Scuole	id_scuola	PK (Primaria)	Numerico	11
	nome		Carattere	30
	telefono		Carattere	30
	fax		Carattere	30
	mail		Carattere	30
	sitoweb		Carattere	30
	via		Carattere	30
	regione	FK (Esterna)	Numerico	11
	provincia	FK (Esterna)	Numerico	11
	comune	FK (Esterna)	Numerico	11
Classi	id_classe	PK (Primaria)	Numerico	11
	classe		Numerico	11
	sezione	FK (Esterna)	Numerico	11
Sezioni	id_sezione	PK (Primaria)	Numerico	11
	sezione		Numerico	11
	id_classe	FK (Esterna)	Numerico	11
Amici	id_amici	PK (Primaria)	Numerico	11
	amico_mittente	FK (Esterna)	Numerico	11
	amico_destinatario	FK (Esterna)	Numerico	11
	stato		Numerico	11
Comuni	id_com	PK (Primaria)	Numerico	11
	id_pro	FK (Esterna)	Numerico	11
	cap		Numerico	11
	comune		Carattere	30
Indirizzi	id_indirizzo	PK (Primaria)	Numerico	11
	indirizzo		Carattere	30
	scuola	FK (Esterna)	Numerico	11
Provincie	id_pro	PK (Primaria)	Numerico	11
	id_reg	FK (Esterna)	Numerico	11
	nome_provincia		Carattere	30
Regioni	id_reg	PK (Primaria)	Numerico	11
	nome_regioni		Carattere	30
Sezioni	id_sezione	PK (Primaria)	Numerico	11
	nome_sezione		Carattere	30
	indirizzo	FK (Esterna)	Numerico	11

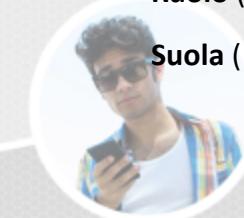
Tabelle:

Utente (id_utente, nome, cognome, data_nascita, via, *regione*, *provincia*, *comune*, mail, password, avatar, ultimo_accesso)

Post (id_post, data_ora, contenuto, *scritto_per_utente*, *scritto_per_classe*, *scritto_da_utente*)

Ruolo (id_ruolo, tipo)

Scuola (id_scuola, nome, telefono, fax, mail, sitoweb, via, *regione*, *provincia*, *comune*)



Classe (id_classe, classe, sezione)

Sezione (id_sezione, nome_sezione, indirizzo)

Indirizzo (id_indirizzo, indirizzo, scuola)

Amici (id_amici, amico_mittente, amico_destinatario, stato)

Comune (id_com, id_pro, cap, comune)

Provincia (id_pro, id_reg, nome_provincia)

Regione (id_reg, nome_regione)

Utente_Classe (id_ute_cla, utente, classe)

Esempi di Query utilizzati all'interno del sito:

- Crea database:

```
CREATE DATABASE IF NOT EXISTS schoolsocial;
```

- Crea tabella scuole:

```
CREATE TABLE IF NOT EXISTS scuole (  
    id_scuola int(11) primary key auto_increment not null,  
    nome char(30) not null,  
    telefono char(30),  
    fax char(30),  
    mail char(30),  
    sitoweb char(50),  
    via char(30),  
    regione int(11),  
    FOREIGN KEY (regione) REFERENCES regioni(id_reg),  
    provincia int(11),  
    FOREIGN KEY (provincia) REFERENCES province(id_pro),  
    comune int(11),  
    FOREIGN KEY (comune) REFERENCES comuni(id_com)  
);
```

- Crea tabella utenti:



```

CREATE TABLE IF NOT EXISTS utenti (
    id_utente int(11) primary key auto_increment not null,
    nome char(30) not null,
    cognome char(30) not null,
    data_nascita date,
    via char(30),
    regione int(11),
    FOREIGN KEY (regione) REFERENCES regioni(id_reg),
    provincia int(11),
    FOREIGN KEY (provincia) REFERENCES province(id_pro),
    comune int(11),
    FOREIGN KEY (comune) REFERENCES comuni(id_com),
    mail char(30) not null,
    password char(30) not null,
    avatar char(30),
    ruolo int(11),
    ultimo_accesso datetime,
    FOREIGN KEY (ruolo) REFERENCES ruoli(id_ruolo)
);

```

- Crea tabella relazione tra UTENTE (Amici):

```

CREATE TABLE IF NOT EXISTS amici (
    id_amici int(11) primary key auto_increment not null,
    amico_mittente int(11),
    amico_destinatario int(11),
    stato boolean,
    FOREIGN KEY (amico_mittente) REFERENCES utenti(id_utente),
    FOREIGN KEY (amico_destinatario) REFERENCES utenti(id_utente)
);

```

- Crea tabella post:



```

CREATE TABLE IF NOT EXISTS post (
    id_post int(11) primary key auto_increment not null,
    data_ora datetime,
    contenuto text not null,
    scritto_per_utente int(11),
    scritto_per_classe int(11),
    scritto_da_utente int(11) not null,
    FOREIGN KEY (scritto_da_utente) REFERENCES utenti(id_utente),
    FOREIGN KEY (scritto_per_utente) REFERENCES utenti(id_utente),
    FOREIGN KEY (scritto_per_classe) REFERENCES classi(id_classe)
);


```

Esempi di codice CSS utilizzati all'interno del sito:

The screenshot shows three separate code editors, each with a tab labeled "style.css". The code is presented in a numbered format, likely indicating line numbers.

- Editor 1 (Left):** Contains CSS for various elements including #header2, #logopic, and #menu. It includes properties like width, height, position: fixed, and box-shadow.
- Editor 2 (Middle):** Contains CSS for #utente_avatar, @font-face, and #utente_nome. It includes properties like min-height, max-height, background-color, and font-family: AGENCYB.
- Editor 3 (Right):** Contains CSS for .bubble and .bubble:after. It includes properties like background-size, position: absolute, border-radius, and border-style: solid.



Esempi di codice PHP/HTML utilizzati all'interno del sito:

```
25 if($check_classe_utente){ //Se l'array è pieno
26
27     //Controllo se l'utente ha ricevuto richieste d'amicizia
28     $query = "SELECT `amico_destinatario`, `stato` FROM `amici` WHERE amico_destinatario = ".$_SESSION["utente"]." AND stato =0"; //Query
29     $n_richieste_ris =mysql_query($query); //Esegua
30     $n_richieste = mysql_num_rows($n_richieste_ris); //Conto le rich.
31
32     //Ricavo le informazioni riguardanti l'indirizzo dell'utente che ha effettuato l'accesso
33     $query = "SELECT regioni.nome_regione, comuni.comune, province.nome_provincia
34     FROM regioni
35     INNER JOIN province ON (province.id_reg = regioni.id_reg)
36     INNER JOIN comuni ON (comuni.id_pro = province.id_pro)
37     WHERE comuni.id_com = ".$info_utente["comune"].""; //Query
38     $comune_utente_ris = mysql_query($query); //Esegua
39     $comune_utente = mysql_fetch_array($comune_utente_ris); //Crea l'array
40
41     //Ricavo il tiro di ruolo dell'utente loggato
42     $query = "select * from ruoli where id_ruolo='".$info_utente["ruolo"].""; //Query
43     $ruolo_utente_ris = mysql_query($query); //Esegua
44     if($ruolo_utente_ris>0) //Controllo se il risultato è positivo
45         $ruolo_utente = mysql_fetch_array($ruolo_utente_ris); //Crea l'array
46
47     //Ricavo le informazioni della classe, scuola, sezione, indirizzo dell'utente loggato
48     $query = "SELECT scuole.nome, sezioni.nome_sezione, indirizzi.indirizzo, classi.classe
49     FROM indirizzi
50     INNER JOIN scuole ON ( indirizzi.scuola = scuole.id_scuola )
51     INNER JOIN sezioni ON ( sezioni.indirizzo = indirizzi.id_indirizzo )
52     INNER JOIN classi ON ( classi.sezione = sezioni.id_sezione ) WHERE classi.id_classe = ".$check_classe_utente["classe"].""; //Query
53     $info_scuola_utente_ris = mysql_query($query); //Esegua
54     if($info_scuola_utente_ris>0) //Controllo se il risultato è positivo
55         $info_scuola_utente = mysql_fetch_array($info_scuola_utente_ris); //Crea l'array
56
57     //Controllo se sono arrivati nuovi post solo per l'utente
58     $query = "SELECT * FROM `post` WHERE `scritto_per_utente` = ".$_SESSION['utente']."' AND data_ora > '".$info_utente["ultimo_accesso"]."' ORDER BY data_ora DESC" ; //Query
59     $n_new_post_ris = mysql_query($query); //Esegua
```

```
332 <br>
333
334     Selezione un comune:
335     <select id="comuni" name="comuni">
336         <?php echo $opt->ShowComuni($info_utente["provincia"]); ?>
337     </select>
338     <br>
339
340     Via : <input type="text" id="via" name="via" value="<?php echo $info_utente["via"] ;?>"> <br>
341
342     E-Mail : <input type="email" id="mail" name="mail" value="<?php echo $info_utente["mail"] ;?>"> <br>
343
344     Ruolo: <select name="ruolo" id="ruolo">
345     <?php
346
347     while($ruoli = mysql_fetch_array($ruoli_ris)){
348         echo '<option value='.$ruoli['id_ruolo'].'>'.$ruoli['tipo'].'</option>';
349     }
350
351     <br>
352
353     </select><br>
354
355     <?php if($info_utente["ruolo"]==2){
356         echo "<ul type='square'>";
357         $query="SELECT * FROM ute_cla WHERE utente=".$_SESSION["utente"]."";
358         $check_classe_utente_ris = mysql_query($query); //Esegua
359         if($check_classe_utente_ris>0) //Controlla se il risultato è positivo
360             $check_classe_utente = mysql_fetch_array($check_classe_utente_ris); //Crea l'array
361
362         while($check_classe_utente){ //Ciclo le classi dell'utente
363             $query = "SELECT scuole . * , sezioni . * , indirizzi . * , classi.*"
364             FROM indirizzi
365             INNER JOIN scuole ON (indirizzi.scuola = scuole.id_scuola )
366             INNER JOIN sezioni ON (sezioni.indirizzo = indirizzi.id_indirizzo )
367             INNER JOIN classi ON (classi.sezione = sezioni.id_sezione ) WHERE classi.id_classe = ".$check_classe_utente['classe']."'";
```



Esempi di codice JQuery/Ajax utilizzati all'interno del sito:

```
84 //Crea link per Google Maps
85 //Funzione di ricerca
86 //Controlli sulla finestra
87 //Registrazione
88 //Autocompleta
89 //Validazione dei campi
90 //Salvataggio dei dati
91 //Modifica dei dati
92 //Eliminazione dei dati
93 //Visualizzazione dei dati
94 //Stampa dei dati
95 //E-mail dei dati
96 //Aggiornamento dei dati
97 //Cerca
98 //Ricerca
99 //Visualizza
100 //Stampa
101 //E-mail
102 //Aggiorna
103 //Cerca
104 //Ricerca
105 //Visualizza
106 //Stampa
107 //E-mail
108 //Aggiorna
109 //Cerca
110 //Ricerca
111 //Visualizza
112 //Stampa
113 //E-mail
114 //Aggiorna
115 //Cerca
116 //Ricerca
117 //Visualizza
118 //Stampa
119 //E-mail
120 //Aggiorna
121 //Cerca
122 //Ricerca
123 //Visualizza
124 //Stampa
125 //E-mail
126 //Aggiorna
127 //Cerca
128 //Ricerca
129 //Visualizza
130 //Stampa
131 //E-mail
132 //Aggiorna
133 //Cerca
134 //Ricerca
135 //Visualizza
136 //Stampa
137 //E-mail
138 //Aggiorna
139 //Cerca
140 //Ricerca
141 //Visualizza
142 //Stampa
143 //E-mail
144 //Aggiorna
145 //Cerca
146 //Ricerca
147 //Visualizza
148 //Stampa
149 //E-mail
150 //Aggiorna
151 //Cerca
152 //Ricerca
153 //Visualizza
154 //Stampa
155 //E-mail
156 //Aggiorna
157 //Cerca
158 //Ricerca
159 //Visualizza
160 //Stampa
161 //E-mail
162 //Aggiorna
163 //Cerca
164 //Ricerca
165 //Visualizza
166 //Stampa
167 //E-mail
168 //Aggiorna
169 //Cerca
170 //Ricerca
171 //Visualizza
172 //Stampa
173 //E-mail
174 //Aggiorna
175 //Cerca
176 //Ricerca
177 //Visualizza
178 //Stampa
179 //E-mail
180 //Aggiorna
181 //Cerca
182 //Ricerca
183 //Visualizza
184 //Stampa
185 //E-mail
186 //Aggiorna
187 //Cerca
188 //Ricerca
189 //Visualizza
190 //Stampa
191 //E-mail
192 //Aggiorna
193 //Cerca
194 //Ricerca
195 //Visualizza
196 //Stampa
197 //E-mail
198 //Aggiorna
199 //Cerca
200 //Ricerca
201 //Visualizza
202 //Stampa
203 //E-mail
204 //Aggiorna
205 //Cerca
206 //Ricerca
207 //Visualizza
208 //Stampa
209 //E-mail
210 //Aggiorna
211 //Cerca
212 //Ricerca
213 //Visualizza
214 //Stampa
215 //E-mail
216 //Aggiorna
217 //Cerca
218 //Ricerca
219 //Visualizza
220 //Stampa
221 //E-mail
222 //Aggiorna
223 //Cerca
224 //Ricerca
225 //Visualizza
226 //Stampa
227 //E-mail
228 //Aggiorna
229 //Cerca
230 //Ricerca
231 //Visualizza
232 //Stampa
233 //E-mail
234 //Aggiorna
235 //Cerca
236 //Ricerca
237 //Visualizza
238 //Stampa
239 //E-mail
240 //Aggiorna
241 //Cerca
242 //Ricerca
243 //Visualizza
244 //Stampa
245 //E-mail
246 //Aggiorna
247 //Cerca
248 //Ricerca
249 //Visualizza
250 //Stampa
251 //E-mail
252 //Aggiorna
253 //Cerca
254 //Ricerca
255 //Visualizza
256 //Stampa
257 //E-mail
258 //Aggiorna
259 //Cerca
260 //Ricerca
261 //Visualizza
262 //Stampa
263 //E-mail
264 //Aggiorna
```

Enjoy It!

Maggio Alessandro (Tk-Alex)

<http://www.tkdalex.altervista.org/>

ITT e LSSA "Copernico"

5B Informatica 2013/2014

<http://www.istitutocopernico.gov.it/>



Powered by Alessandro Maggio