Sito web con pagina di registrazione e verifica dati

Titolo del progetto: Sito web con pagina di registrazione e verifica dati
Alunno/a: Luca Di Bello
Classe: Info 3AC
Anno scolastico: 2018/2019
Docente responsabile: Luca Muggiasca, Francesco Mussi, Adriano Barchi, Elisa Nannini

1 Introduzione

1.1 Informazioni sul progetto

Il progetto sarà gestito e realizzato dal alievo Luca Di Bello (studente di informatica all'Arti e Mestieri di Trevano) sotto la supervisione del professor Luca Muggiasca, Adriano Barchi, Francesco Mussi e dalla professoressa Elisa Nannini.
La realizzazione del progetto prenderà piede il 5 settembre 2018 (05/09/18) e dovrà essere consegnato entro il 1 novembre 2018 (01/11/18).

1.2 Abstract

Nowadays, different products used for organizing giveaways are very cumbersome, slow and not very user-friendly. With this open-source product we offer a simple, fast and lightweight solution for organizing contests registrations, giveaways, surveys and much more. All data are saved on the webserver in several csv files, one containing all registered users while at the same time the various daily registrations are divided into different files to make statistics and management easier.
We have implemented a complex system for the validation of the data entered by the user, to ensure that the data collected are as truthful and certified as possible.
With this product contest organizers, giveaways and much more will be facilitated in multiple operations, saving time and much money.

1.3 Scopo

Lo scopo di questo progetto è di insegnarci come gestire interamente, quindi la gestione dei diari dei diari, della documentazione e dell'implementazione.

Analisi

1.4 Analisi del dominio

Il prodotto verrà utilizzato sul web, quindi esso sarà accessibile da tutti i dispositivi (sia mobile sia fissi) che dispongono di una connessione internet. La soluzione che offre questo prodotto è già disponibile in rete, esso infatti si inspira a molti di essi e rispetta le convenzioni definite da essi.

1.5 Analisi e specifica dei requisiti

Il progetto sarà strutturato su più pagine (come richiesto nella specifica): la pagina iniziale deve contenere un pulsante che porterà alla pagina di registrazione, all'interno di essa vi sarà un form che servirà a gestire i dati per la registrazione.
I campi saranno 12

- Nome (Testo) - Obbligatorio
- Cognome (Testo) - Obbligatorio
- Data Nascita (Data) – Obbligatorio
- Via (Testo) - Obbligatorio
- No. Civico (Numerico, 3 cifre) - Obbligatorio
- Città (Testo) - Obbligatorio
- Nap (Numerico, 5 cifre) - Obbligatorio
- No. Telefono (solo cifre, spazi e/o trattini - Obbligatorio
- E-Mail (Testo + Controllo formato e-mail testo@testo.testo) - Obbligatorio
- Sesso (Checkbox F/M) - Obbligatorio
- Hobby (Testo)
- Professione (Testo)

Sul fondo del form ci saranno due bottoni, uno intitolato "Cancella" che si occuperà di "pulire" il form rimuovendo tutti i dati immessi ed il tasto "Avanti".
Tutti i campi avranno un validatore che controllerà se il dato è valido o meno. Se tutti i dati sono corretti l'utente può cliccare su il tasto "Avanti" che lo porterà ad una pagina riassuntiva per il controllo dati mentre se i validatore hanno trovato uno o più dati invalidi l'utente sarà obbligato a modificarli per registrarsi.
Sul fondo della pagina di controllo ci saranno due tasti, uno intitolato "Correggi" che permetterà di modificare i dati immessi (senza doverli reinserire tutti) ed il pulsante "Registra" che permetterà all'utente di portare a termine la registrazione. Dopo la registrazione l'utente verrà portato in una pagina home dove verranno mostrati tutti i dati della registrazione nello stesso ordine in cui sono stati inseriti.
I dati delle registrazioni verranno salvati in due file CSV, uno contenente le registrazioni odierne e l'altro che conterrà le registrazioni globali (tutte le registrazioni effettuate) con l'aggiunta di un campo che conterrà la data e l'ora della registrazione (lato server). Durante la registrazione i dati verranno scritti in parallelo sia nel file CSV giornaliero sia in quello locale. Se si effettua la prima registrazione del giorno verrà creato un nuovo file CSV giornaliero dove verranno scritti i parallelo i nuovi dati giornalieri nel nuovo file e nel file globale.

ID: REQ-001	
Nome	Pagina web iniziale
Priorità	3
Versione	1.0
Note	
Sotto requisiti	
001	Necessita di un tasto “registrati” che porta al link di registrazione

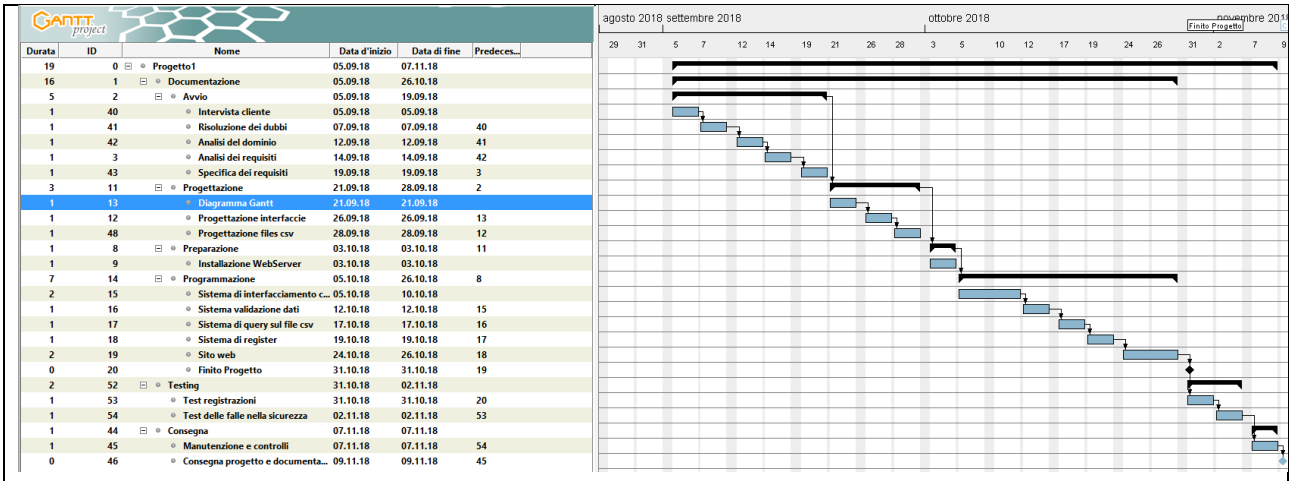
ID: REQ-002	
Nome	Pagina di registrazione tramite form
Priorità	1
Versione	1.0
Note	Pagina nella quale l’utente può fare la registrazione
Sotto requisiti	
001	Form da compilare con questra struttura e con queste restrizioni: • Nome (campo testo) - Obbligatorio • Cognome (campo testo – Obbligatorio • Data di nascita (campo data) - Obbligatorio • Numero Civico (campo numerico, massimo 3 cifre) – Obbligatorio • Città (campo testo) – Obbligatorio • Nap (campo numerico, massimo 5 cifre) – Obbligatorio • Numero di telefono (campo testo, ammessi solo cifre, spazi e/o trattini) - Obbligatorio • E-Mail (campo testo, controllo formato “testo@testo.testo”) – Obbligatorio • Genere (campo testo F/M) – Obbligatorio • Hobby (campo testo) • Professione (campo testo)
002	Pulsante cancella che permette di eliminare tutti i dati contenuti nel form
003	Pulsante avanti che permette di continuare nel percorso di registrazione
004	Vi saranno dei validatori che controlleranno i dati inseriti nei campi
005	I validatori devono assicurarsi che il dato sia consono al campo in cui è inserito
006	I validatori devono assicurarsi se il dato inserito abbia un senso logico
007	La validazione dei dati viene svolta sia in locale sia dal server

ID: REQ-003	
Nome	Pagina di conferma dati
Priorità	1
Versione	1.0
Note	Questa pagina si occupa di mostrare all'utente tutti i dati che ha inserito all'interno del form.
Sotto requisiti	
001	Tabella che mostra all'utente tutti i dati che ha inserito all'interno del form
002	Pulsante correggi che permette alll'utente di tornare indietro al form di registrazione, esso sarà già compilato con i dati precedentemente inseriti al fine di velocizzare la loro modifica
003	Pulsante registra con il quale l'utente potrà registrarsi definitivamente
004	Al momento della registrazione i dati dell'utente verranno scritti su entrambi i file csv in parallelo

ID: REQ-004	
Nome	File csv per conserva dati
Priorità	1
Versione	1.0
Note	I dati contenuti in questi file vengono scritti dopo la conferma nella pagina di conferma dati
Sotto requisiti	
001	File csv chiamato "Registrazioni_tutte.csv" che tiene in memoria tutti i dati di tutti gli utenti
002	File csv chiamato "Registrazione_aaaa-mm-gg.csv" che contiene i dati degli utenti creati in quel determinato giorno
003	I file verranno generati automaticamente dalla pagina web senza bisogno di nessun intervento da parte del utente, quindi durante la registrazione del primo utente della giornata verrà pure creato un nuovo file csv. Gli altri utenti della giornata andranno ad aggiungere i loro dati ai file senza sovrascriverli
004	Questi file si troveranno nella cartella "Registrazioni" che è posta sotto la cartella principale del sito
005	Ci possono essere utenti doppi, non c'è nessun controllo

ID: REQ-005	
Nome	Pagina Home
Priorità	1
Versione	1.0
Note	Mostra tutti i dati di tutti gli utenti registrati nella giornata

1.6 Pianificazione



Durata	ID	Nome	Data d'inizio	Data di fine	Predeces...
19	0	Progetto1	05.09.18	07.11.18	
16	1	Documentazione	05.09.18	26.10.18	
5	2	Avvio	05.09.18	19.09.18	
1	40	Intervista cliente	05.09.18	05.09.18	
1	41	Risoluzione dei dubbi	07.09.18	07.09.18	40
1	42	Analisi del dominio	12.09.18	12.09.18	41
1	3	Analisi dei requisiti	14.09.18	14.09.18	42
1	43	Specifica dei requisiti	19.09.18	19.09.18	3
3	11	Progettazione	21.09.18	28.09.18	2
1	13	Diagramma Gantt	21.09.18	21.09.18	
1	12	Progettazione interfaccie	26.09.18	26.09.18	13
1	48	Progettazione files csv	28.09.18	28.09.18	12
1	8	Preparazione	03.10.18	03.10.18	11
1	9	Installazione WebServer	03.10.18	03.10.18	
7	14	Programmazione	05.10.18	26.10.18	8
2	15	Sistema di interfacciamento c...	05.10.18	10.10.18	
1	16	Sistema validazione dati	12.10.18	12.10.18	15
1	17	Sistema di query sul file csv	17.10.18	17.10.18	16
1	18	Sistema di register	19.10.18	19.10.18	17
2	19	Sito web	24.10.18	26.10.18	18
0	20	Finito Progetto	31.10.18	31.10.18	19
2	52	Testing	31.10.18	02.11.18	
1	53	Test registrazioni	31.10.18	31.10.18	20
1	54	Test delle falle nella sicurezza	02.11.18	02.11.18	53
1	44	Consegna	07.11.18	07.11.18	
1	45	Manutenzione e controlli	07.11.18	07.11.18	54
0	46	Consegna progetto e documenta...	09.11.18	09.11.18	45

Figura 1: Diagramma di Gantt preventivo del progetto

1.7 Analisi dei mezzi

1.7.1 Software

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.
Per la realizzazione di questo progetto verranno utilizzati i sequenti software:

- GanttProject 2.8.5
- PhpStorm 2018.1 e Visual Studio Code 1.2.8
- Webserver tramite Apache 2.4.27 e Php 7.2.10 VC15 x64 Thread Safe
- XDebug per facilitarmi i processi di debugging del codice PHP
- MaterializeCSS e Jquery per la grafica e le animazioni

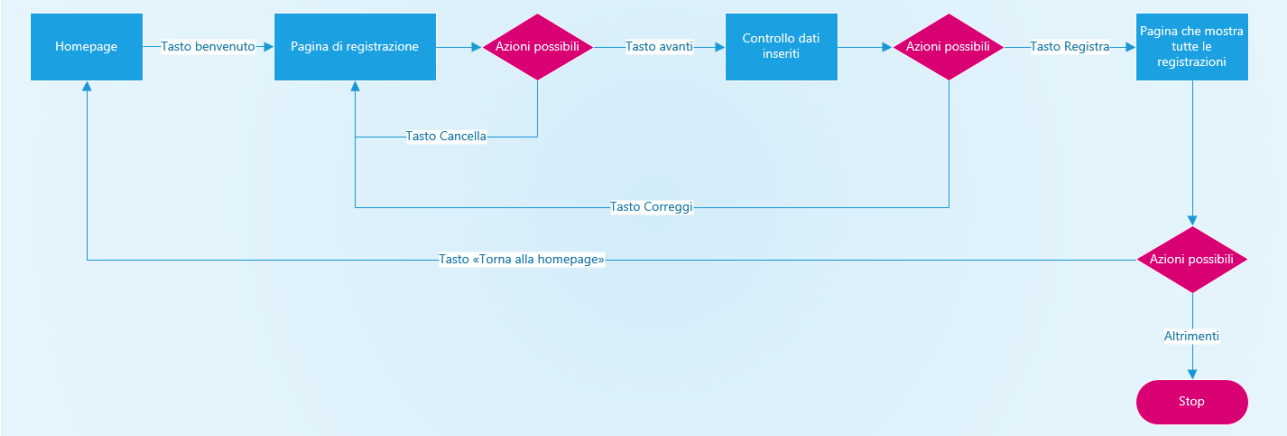
1.7.2 Hardware

Il prodotto potrà essere utilizzabile su qualsiasi dispositivo abile nella connessione ad una pagina web.
La macchina sulla quale sarà sviluppato il prodotto avrà queste specifiche:

- GPU: Nvidia GeForce GTX 1060 6GB
- RAM: 16 GB
- CPU: Intel Core i7 7700HQ 2.8GHz

2 Progettazione

2.1 Design dell'architettura del sistema



2.2 Design dei dati e database

I dati verrano salvati all'interno dei file csv globale con questo schema:
reg_date;nome;cognome;data di nascita;numero civico;città;NAP;numero di telefono;email;genere;hobby;professione

Invece i dati del file csv giornaliero verranno salvato con quest'altro schema molto simile:
nome;cognome;data di nascita;numero civico;città;NAP;numero di telefono;email;genere;hobby;professione

I dati all'interno del file csv saranno separati dal carattere ';'. Inoltre gli id dei dati salvati all'interno del file csv globale avranno il classico id incrementale e quindi non avranno nessuna relazione con gli id del file csv locale.

Nel file globale i dati avranno quindi questo aspetto:

```
reg_date;first_name;last_name;data_nascita;sesso;email;citta;cap;via;numero_civico;numero_telefono;work;hobby
2018-11-01 17:29:25;luca;di bello;2001-10-25;maschio;luca.dibello@samtrevano.ch;arzo;6864;cave di marmo;4;+41765969984;student
```

Mentre nel file csv delle registrazioni giornaliere i dati avranno questo aspetto:

```
first_name;last_name;data_nascita;sesso;email;citta;cap;via;numero_civico;numero_telefono;work;hobby
luca;di bello;2001-10-25;maschio;luca.dibello@samtrevano.ch;arzo;6864;cave di marmo;4;+41765969984;studente;testo sttrano {prc
```

2.3 Design delle interfacce

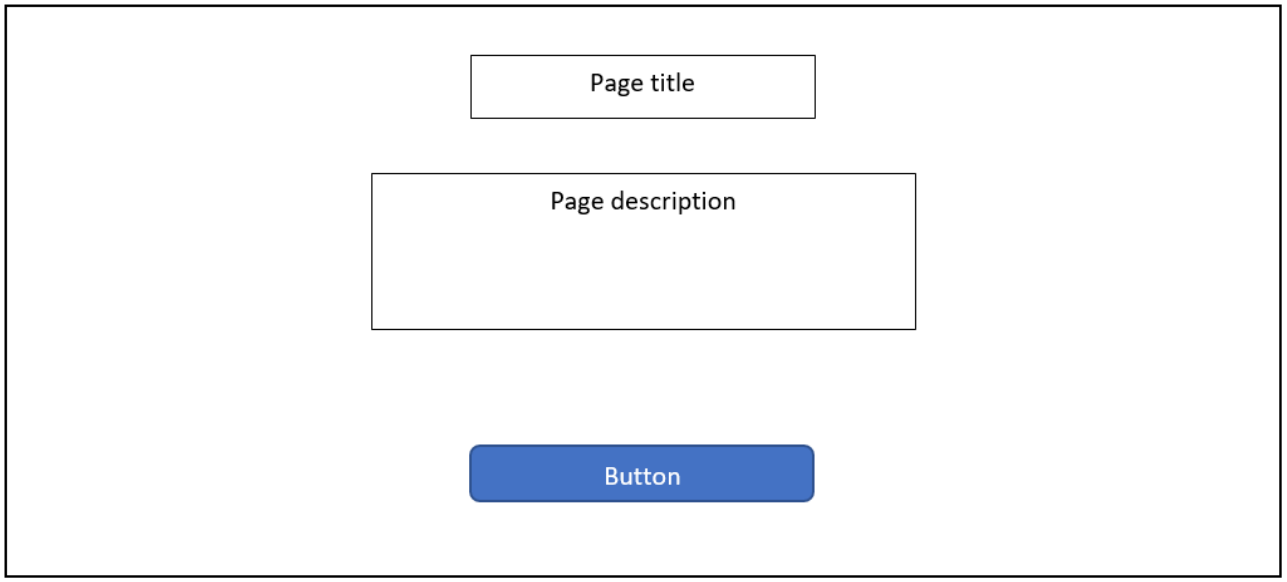


Figura 2 - Pagina iniziale

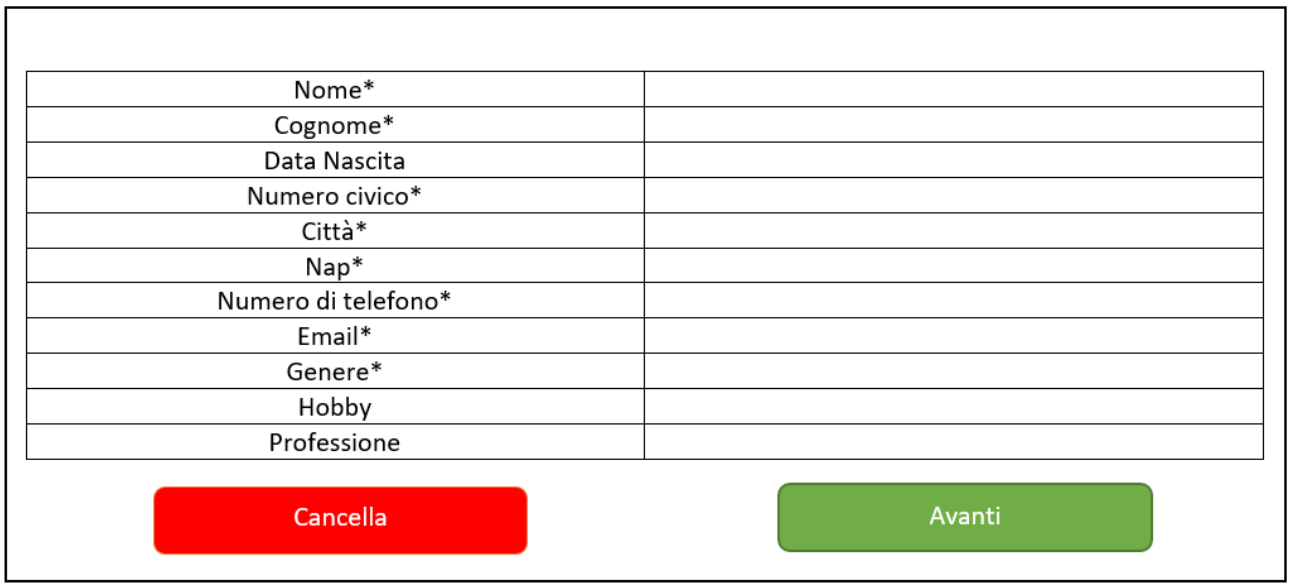


Figura 3 - Pagina di registrazione

Nome*	
Cognome*	
Data Nascita	
Numero civico*	
Città*	
Nap*	
Numero di telefono*	
Email*	
Genere*	
Hobby	
Professione	

Correggi

Registrati

Figura 4 - Pagina di controllo

Nome*	
Cognome*	
Data Nascita	
Numero civico*	
Città*	
Nap*	
Numero di telefono*	
Email*	
Genere*	
Hobby	
Professione	

Torna alla home

Figura 5 - Pagina di riassunto

2.4 Design procedurale

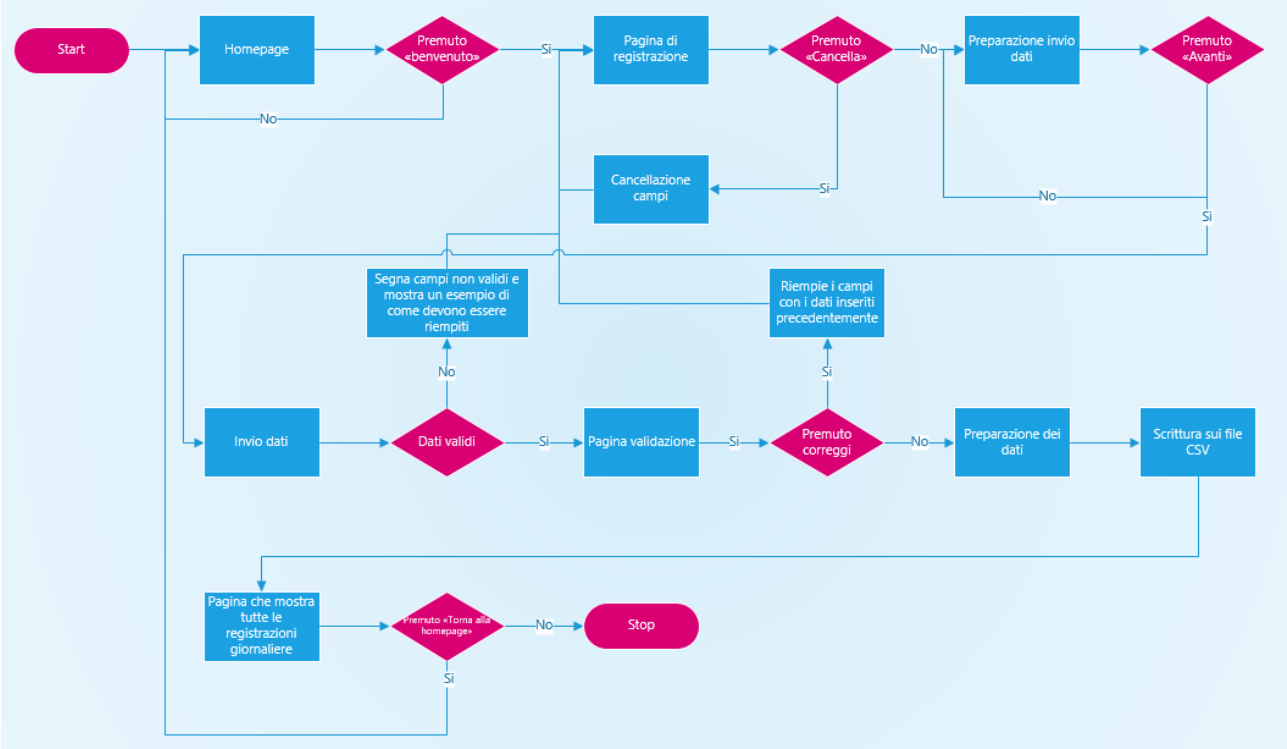


Figura 6 - Design procedurale

3 Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.
Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.
In questa parte è richiesto l'inserimento di codice sorgente/print screen di maschere solamente per quei passaggi particolarmente significativi e/o critici.
Inoltre dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.
Non deve apparire nessuna forma di guida d'uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.
Per eventuali dettagli si possono inserire riferimenti ai diari.

Questo progetto è stato scritto con HTML e CSS per dare lo stile alle interfaccie mentre è stato utilizzato JavaScript per la programmazione front-end e php per la programmazione back-end.
Per aiutarmi nello stile delle pagine ho utilizzato un framework abbastanza famoso chiamato "MaterializeCss", esso permette di semplificare ed automatizzare molte operazioni. Il download e la documentazione di esso è trovabile a questo link: https://materializecss.com/.
Il progetto è costituito di 4 pagine:

1. Pagina di benvenuto, la quale spinge l'utente a registrarsi
2. Pagina di registrazione, dove l'utente può inserire i suoi dati personali
3. Pagina di controllo, dove l'utente può controllare i dati inseriti prima della registrazione effettiva e se desidera ha la possibilità di cambiarli senza dovreli riscriverli tutti a mano
4. Pagina di riassunto, dove verranno mostrati tutti i dati di tutte gli utenti registrati quel giorno

Per semplificarmi la programmazione di codice in JavaScript ho scaricato la libreria chiamata Jquery, la quale consente una minore scrittura di codice, una scrittura più veloce ed un codice semanticamente più comprensibile. Essa è trovabile a questo indirizzo: https://jquery.com/.

La prima pagina presenta un semplicissimo carousel di immagini a scomparsa (uno ‘show’ di immagini, una dopo l’altra), le immagini cambiano ogni 5 secondi:

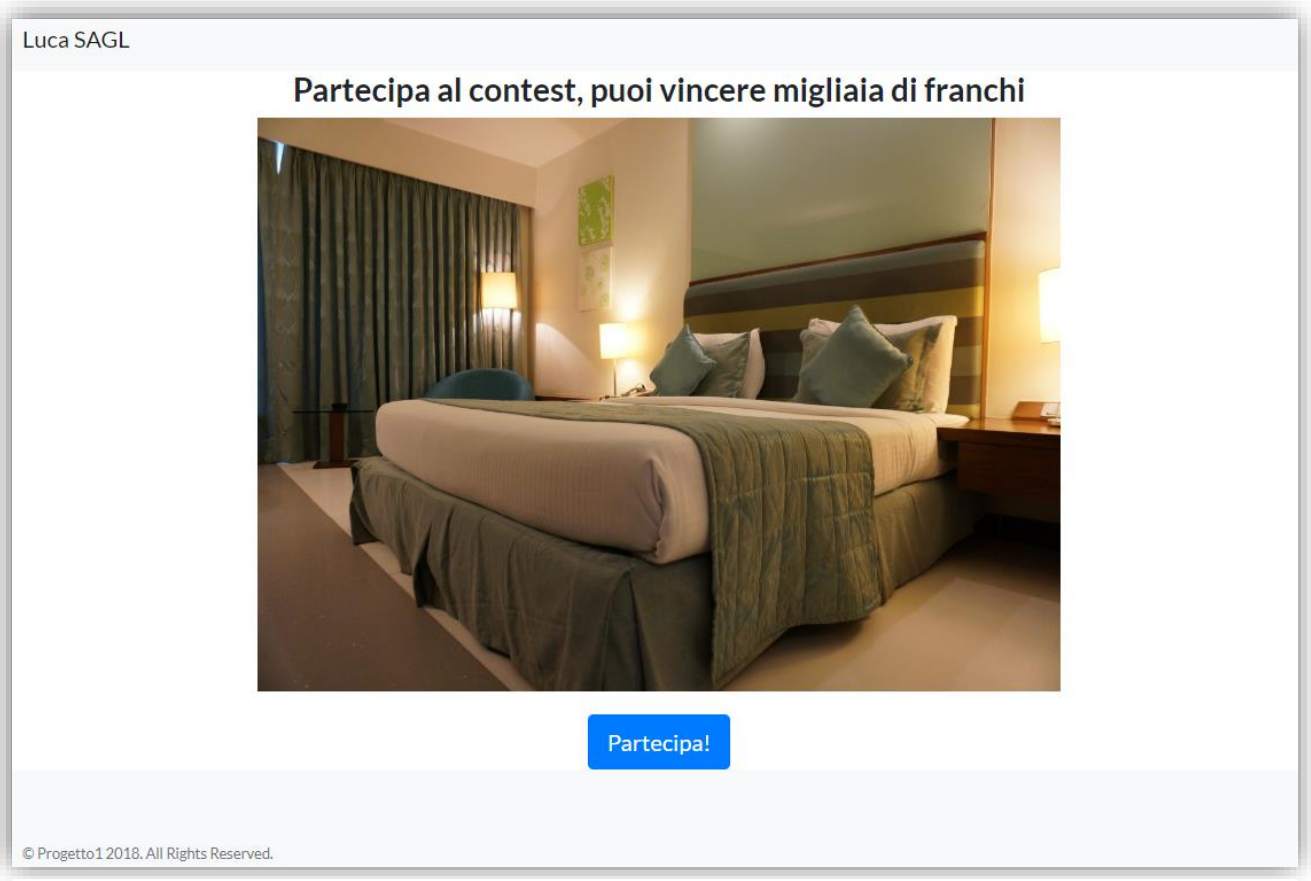


Figura 7 - Index.html

Per fare questa animazione ho scritto questo codice:

```
var intervall = 5000;

//Cambia l'immagine prima della fine del caricamento della pagina
switchImage();

//Quando la finestra è carica (loaded) inizia il carousel
$(document).ready(function () {
  setInterval(function () {
    //Nasconde l'immagine, arrivando fino ad un opacità di 0.15
    $("#showcase").fadeTo("slow", 0.15, function () {
      //Quando arriva ad un opacità di '0.15' cambia immagine
      switchImage();
      //Dopo aver nasconsto l'immagine cambia nuovamente l'opacità a '1' (quindi totalmente visibile)
      $("#showcase").fadeTo("slow", 1);
    });
    //Tutto questo viene eseguito ogni 'intervall' millisecondi
  }, intervall);

});
```

La pagina di registrazione invece comprende un form con all'interno tutti i campi descritti nella specifica, ovvero: Nome, Cognome, Data Nascita, Via, No. Civico, Città, Nap, No. Telefono, E-Mail, Sesso, Hobby e Professione.
Questi campi presentano due tipi di validazioni:

- Validazione dati lato client in tempo reale (quindi validazione che avviene subito durante l'inserimento del dato) tramite degli eventi keyUp e keyDown
- Validazione dati lato server tramite l'utilizzo di Php che avviene quando si tenta di continuare nel processo di registrazione

Per svolgere le validazioni ho creato due classi: una classe è chiamata "validator.js", la quale comprende tutti i metodi utili per la validazione dei campi del form lato client, mentre la classe per la validazione lato server è chiamata 'Validator.php' (la quale non è una vera e propia classe ma solamente un file dal quale ereditiamo i metodi scritti al suo interno):

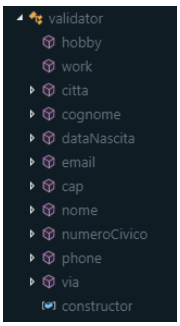


Figura 9 - Classe validazione lato client

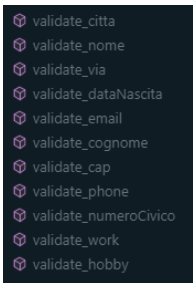


Figura 8 - Classe per la validazione lato server

Tutti i validatori di entrambe le classi validano il dato tramite delle espressioni regolari (regex), le quali controllano che il dato rispetti un certo formato. Il validatore del campo 'data' controllano pure il senso del dato, quindi controllano pure se il dato è verosimile oppure no. Infatti questo validatore controlla pure che la data di nascita non sia nel futuro (quindi non ancora passata) e che non sia posta prima di 120 anni dalla data corrente (quindi l'utente che si vuole iscrivere non può avere più di 120 anni).
I validatori ritornano 'true' se il dato risulta valido e rispettivamente 'false' se il dato non è valido.

Nella classe per la validazione lato server sono presenti delle variabili globali che hanno la funzione di 'impostazioni' per il validatore, le impostazioni possibili sono queste:

- $global_length_min, la quale descrive la lunghezza minima (di caratteri) del dato
- $global_length_max, la quale descrive la lunghezza massima (di caratteri) del dato
- $work_length_max, la quale descrive la lunghezza massima (di caratteri) per il campo 'professione'
- $hobby_length_max, la quale descrive la lunghezza massima (di caratteri) per il campo 'hobby'

Mentre per la classe 'validator.js' è possibile passare le impostazioni direttamente quando si richiama il metodo della validazione.
Esempio di chiamata del metodo per la validazione nome:

```
nome(testo, lunghezza minima, lunghezza massima)
```

Questo è un esempio di un validatore della classe “Validator.php” per la validazione del nome:

```php
function validate_nome($name){
    if(strlen($name) > $GLOBALS["global_length_min"] && strlen($name) < $GLOBALS["global_length_max"]){
        //THIS REGEX LOOK FOR ILLEGAL CHARACTERS
        $rexSafety = "/[\^<,\"@\/\{\}\(\)\*\$%\?=>:\|;#0-9]+/i";
        return !preg_match($rexSafety,$name);
    }
    return false;
}
```

Figura 10 - Validatore lato server del campo nome

Il regex (contenuto nella valiabile $rexSafety) funziona nell’esatto contrario dei regex classici. Lui controlla che il dato non contenga i caratteri:

- ^, <,(carattere virgola), “, @, /, {, }, (,), *, $, %, ?, =, >, :, |, ;, #, (numeri dallo 0 al 9).

Se li contiene ritorna un flag ‘false’ mentre se non li contiene ritorna un flag ‘true’.

Mentre questo è un esempio di un validatore della classe ‘validator.js’ per la validazione del numero civico:

```javascript
numeroCivico(numeroCivico) {
    var re = /^[0-9]{1,3}(|[a-zA-Z]){1}$/;
    if (numeroCivico.length > 0) {
        return re.test(numeroCivico);
    }
    return false;
}
```

Figura 11 - Validatore lato client del campo numero civico

Questo validatore permette l’inserimento di un numero civico che va dal numero ‘0’ al numero ‘999’ e permette l’aggiunta di una singola lettera (sia uppercase che lowercase), quindi permette l’inserimento anche di dati come ‘12B’ oppure ‘12b’.

Tramite i risultati dei validatori lato client e tramite gli eventi keyUp e keyDown attribuiti a tutti gli input del form è possibile cambiare il colore del campo in tempo reale per comunicare in modo visivo all’utente che certi campi contengono dati non validi.

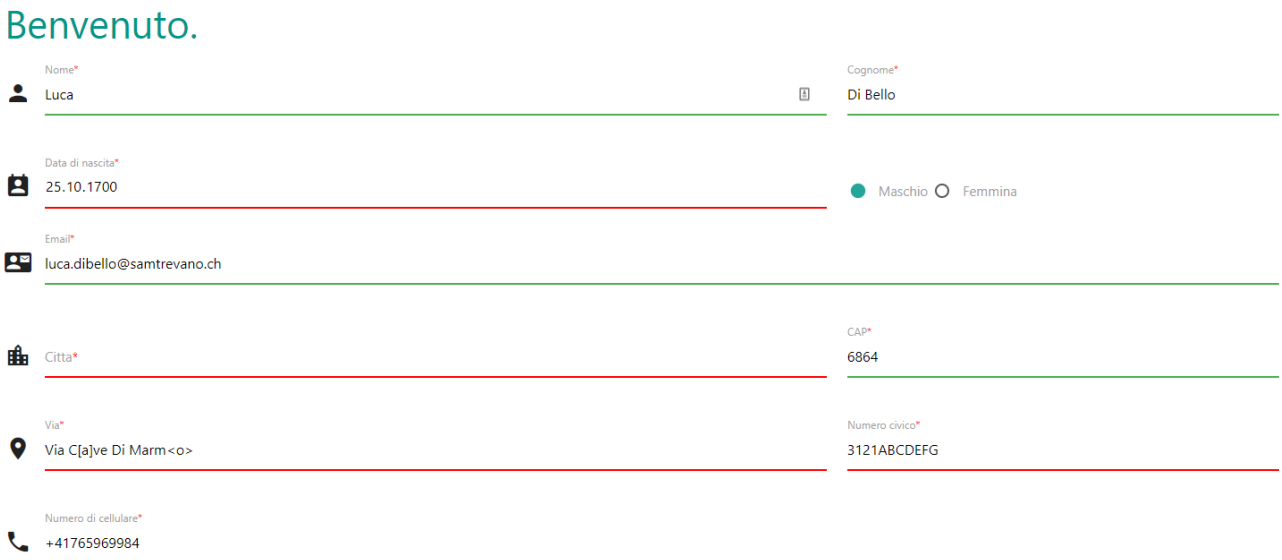


Figura 12 - Form di registrazione con validatori in tempo reale

Continuando nella registrazione, si passerà poi nella pagina di controllo dati dove (per accedervi) i dati verranno validati nuovamente dalla classe 'Validator.php', se tutti i dati risultano validi mostrerà la pagina di riassunto dei dati inseriti, mentre se i dati non sono validi eseguirà un redirect alla pagina contenente il form di registrazione aggiungendo però una variabile GET chiamata 'error' la quale dirà alla pagina di andare a leggere il contenuto della variabile $_SESSION['errors'], la quale contiene i nomi dei campi risultati non validi dal validatore lato server. Questi dati verranno mostrati all'utente tramite un alert-box.
Quando l'utente clicca il pulsante "Correggi" verrà aperta una sessione che si occuperà di tenere in memoria (tramite delle variabili) tutti i dati inseriti dall'utente.
Per far capire alla pagina 'register.php' che voglio eseguire un restore dei dati utilizzano quelli già presenti nelle variabili della sessione, io non faccio altro che eseguire una richiesta di tipo post a quella pagina passando un flag chiamato 'restore' (fatto tramite un input di tipo hidden) una volta cliccato il bottone, ecco il codice del form:

```
<form method="post" action="register.php" style="display:inline;">
    <input type="hidden" name="restore" value="true">
    <button class="btn waves-effect waves-light teal" style="background-color:rgb(211, 21, 21) !important;
</form>
```

Figura 13 - Pulsante 'correggi'

Per inserire i dati dell'utente all'interno della sessione utilizzo un metodo chiamato 'create_restore_session()'. Esso non fa altro che aggiungere una variabile di sessione chiamata $_SESSION['restore'] ed impostare il suo valore a 'true' ed inserisce ogni dato in una variabile differente, quindi la variabile $_POST['nome'] (il dato inviato dal form della pagina 'register.php') finirà in una variabile della sessione $_SESSION['nome'] e così via per tutti i dati inseriti dall'utente. La variabile chiama 'restore' serve soltanto come una flag per controllo almeno al posto di fare un controllo su tutti i dati della sessione mi basta controllare se il flag è stato impostato.

Se l'utente clicca sul pulsante 'Registrati' sul fondo della pagina si esegue un altra richiesta post alla pagina 'register_user.php' con una strategia molto simile a quella di prima, ma dove il campo di tipo hidden è chiamato 'register'.
La pagina 'register.user.php' controlla prima di tutto se la richiesta POST è una richiesta valida (controllando se esiste la variabile $_POST["register"], se è valida la pagina si occupa di prendere tutti i dati della sessione e passarli ad un metodo chiamato 'add_user', il quale si occupa (appoggiandosi alla classe CSVisualizer) di scrivere i dati nella maniera corretta in entrambi i file csv (quello giornaliero e quello globale). Questo metodo si occupa pure del controllo del file, se il file csv al quale si sta cercando di accedere non esiste ne crea uno nuovo con nome corretto ed aggiunge automaticamente l'header corretto alla sua cima.
La classe sulla quale si appoggia (ovvero CSVisualizer) è una classe che si occupa di gestire i file csv di questo progetto in modo semplificato e veloce. Infatti essa presenta molti metodi utili per ricavare i dati e per aggiungere dati, ecco com'è strutturata la classe:

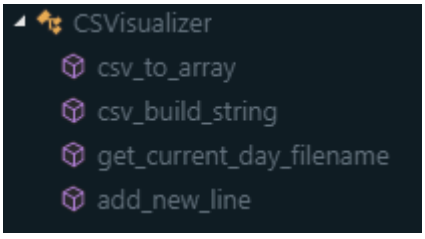


Figura 14 - Classe CSVisualizer.php

Nella pagina di riassunto, mostrata una volta terminata la registrazione, non faccio altro che costruire dinamicamente una tabella utilizzando l'array multi-dimensionale che mi ritorna il metodo di questa classe chiamato 'csv_to_array'.

Ecco un esempio di valore che ritorna questo metodo:

Figura 15 - Valore di ritorno del metodo 'csv_to_array'

Da questa immagine si può vedere che il metodo ritorna un array multi-dimensionale, dove ad ogni indice numerico sono presenti tutti i dati (sotto forma di un array associativo) dell'utente.
Una volta eseguita la registrazione creo una nuova variabile di sessione chiamata $_SESSION["registered"], la quale dice che l'utente della sessione si è già registrato. Questo permette all'utente di accedere alla pagina di riassunto senza dover per forza crearsi nuovamente l'account.
Per chiudere la sessione difinitivamente ho aggiunto un pulsante sul fondo della pagina 'riassunto.php' il quale permette di chiudere (quindi eliminare) la sessione e tutti i dati contenuti in essa, ho svolto questo tramite il metodo di php chiamato "session_destroy".

4 Test

4.1 Protocollo di test

Test Case: **Riferimento**:	TC-001 REQ-002	**Nome:**	Controllo dei caratteri nei campi di testo
Descrizione:	Questo controllo si occupa di testare il funzionamento della validazione dei campi di testo nel form di registrazione tranne del campo hobby. Il validatore ritiene valide le stringhe che hanno meno di 50 caratteri e che non contengono cifre numeriche.		
Prerequisiti:	Il server ed i relativi servizi di comunicazione devono essere attivi.		
Procedura:	1. Recarsi sulla pagina di registrazione ("register.php") 2. Inserire del testo con più di 50 caratteri e con delle cifre numeriche in tutti i campi di testo (tranne il campo hobby) Es: *questaèunaprova123questaèunaprova123questaèunaprova123* 3. Riempire gli altri campi con valori plausibili 4. Provare a continuare con il processo di registrazione schiacciando il bottone intitolato "Avanti"		
Risultati attesi:	I campi di testo (tranne il campo "hobby") saranno segnalate all'utente per renderlo consapevole di aver inserito un valore non valido e cliccando sul bottone "Avanti" verranno mostrati a schermo quali campi non sono stati riempiti correttamente		

Test Case: TC-002 **Riferimento**: REQ-002		**Nome:**	Controllo delle notifiche a schermo
Descrizione:	Questo controllo si occupa di testare il funzionamento delle notifiche a schermo generate nella pagina di registrazione quando c'è la presenza di uno o più errori.		
Prerequisiti:	Il server ed i relativi servizi di comunicazione devono essere attivi.		
Procedura:	1. Recarsi sulla pagina di registrazione ("register.php") 2. Provare a cliccare sul tasto avanti senza aver inserito nessun dato in alcun campo 3. Osservare notifiche toast generate 4. Provare ad inserire dei dati nel campo 'professione' e 'hobby' 5. Osservare notifiche toast generate		
Risultati attesi:	I campi del form contenenti dei dati non validi saranno mostrati a schermo sotto forma di notifica toast intitolata "Error"(ogni errore corrisponde ad una notifica differente). Se è presente almeno un errore nei campi obbligatori e c'è almeno un campo facoltativo (ovvero il campo 'hobby' ed il campo 'professione') vuoto verranno mostrate delle notifiche toast (il numero di notifiche dipende dal numero dei campi facoltativi lascitati vuoti) intitolate "Warning" che fanno notare all'utente che ha lasciato vuoto il campo facoltativo.		

Test Case: TC-003 **Riferimento**: REQ-005		**Nome:**	Controllo della creazione dei file csv
Descrizione:	Questo controllo si occupa di testare il funzionamento della creazione automatica dei file csv (quindi creazione file con nome adatto ed aggiunta header).		
Prerequisiti:	Il server ed i relativi servizi di comunicazione devono essere attivi.		
Procedura:	1. Eliminare i file csv posti nella cartella "registrazioni" 2. Eseguire la registrazione di un utente 3. Andare a controllare la presenza dei nuovi file csv nella cartella "registrazione"		
Risultati attesi:	Il sistema dovrebbe creare automaticamente il file giornaliero ed il file globale delle registrazioni. Se il sistema non trova i file ricrea automaticamente il file csv giornaliero ed il file csv globale con i loro giusti header.		

Test Case: TC-004 **Riferimento**: REQ-004		**Nome:**	Accedere alla pagina controllo.php oppure alla pagina riassunto.php
Descrizione:	Questo controllo si occupa di testare il corretto funzionamento del redirect svolto dalla pagina di controllo dati (controllo.php) e dalla pagina di riassunto finale (riassunto.php).		
Prerequisiti:	Il server ed i relativi servizi di comunicazione devono essere attivi.		
Procedura:	1. Senza aver creato alcun utente recarsi nella pagina di benvenuto 2. Aggiungere all'url già scritto questo testo: "pages/controllo.php" 3. Una volta che la pagina ha caricato cambiare il testo appena inserito con "pages/riassunto.php"		
Risultati attesi:	Entrambe le pagine dovrebbero eseguire dei redirect (ovvero cambiare la pagina richiesta) alla pagina "register.php" (la pagina contenente il form di registrazione).		

Test Case:	TC-005	**Nome:**	Rimpiazzamento doppio spazio e ‘;’
Riferimento:	REQ-002		
Descrizione:	Questo controllo si occupa di testare il corretto funzionamento della correzione durante la scrittura del dato dei doppi spazi con uno spazio singolo ed la rimozione del carattere ‘;’ (anch’essa in tempo reale).		
Prerequisiti:	Il server ed i relativi servizi di comunicazione devono essere attivi.		
Procedura:	1. Recarsi nella pagina di registrazione (register.php) 2. Cliccare il campo nome 3. Tenere premuto spazio per 5 secondi e poi digitare il carattere ‘;’		
Risultati attesi:	Nel campo ‘nome’ dovrebbe rimanere soltanto un singolo spazio.		

Test Case:	TC-006	**Nome:**	Funzionamento tasto cancella
Riferimento:	REQ-002		
Descrizione:	Questo controllo si occupa di testare il corretto funzionamento del bottone “cancella” posto nella pagina di registrazione (register.php).		
Prerequisiti:	Il server ed i relativi servizi di comunicazione devono essere attivi.		
Procedura:	1. Recarsi nella pagina di registrazione (register.php) 2. Riempire tutti i campi con dei valori 3. Cliccare il pulsante ‘cancella’		
Risultati attesi:	Tutti i valori, i colori e le animazioni dei campi dovrebbero venir resettate.		

Test Case:	TC-007	**Nome:**	Funzionamento tasto correggi
Riferimento:	REQ-003		
Descrizione:	Questo controllo si occupa di testare il corretto funzionamento del bottone “correggi” posto nella pagina di controllo dati (controllo.php).		
Prerequisiti:	Il server ed i relativi servizi di comunicazione devono essere attivi.		
Procedura:	1. Recarsi nella pagina di registrazione (register.php) 2. Riempire tutti i campi con dei valori validi 3. Procedere nel processo di registrazione cliccando ‘Avanti’ 4. Una volta aver eseguito l’accesso alla pagina di controllo dati cliccare sul pulsante ‘correggi’		
Risultati attesi:	Tutti i dati inseriti precedentemente dovrebbero essere ancora nei propri input.		

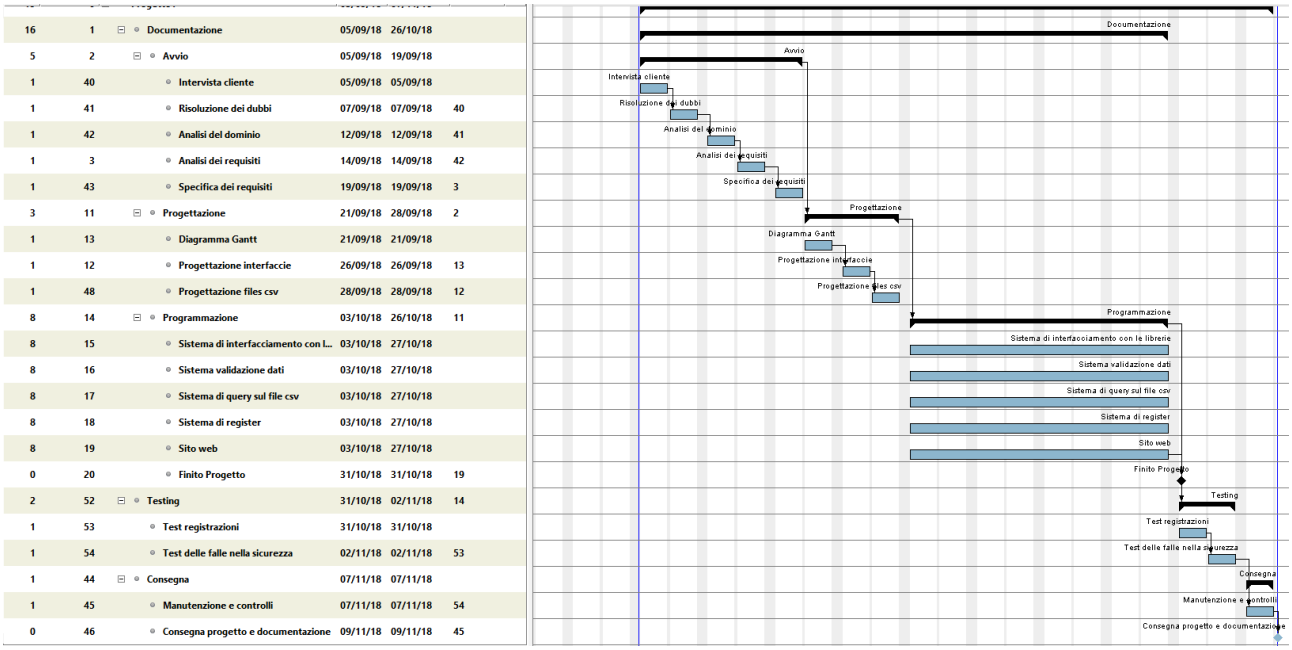
4.2 Risultati test

Test case	Stato test
TC-001	**Passato**
TC-002	**Passato**
TC-003	**Passato**
TC-004	**Passato**
TC-005	**Passato**
TC-006	**Passato**
TC-007	**Passato**

4.3 Mancanze/limitazioni conosciute

Il sito non è molto responsive, esso ha diversi problemi di visualizzazione se si utilizza su un dispositivo mobile (smartphone). Un'altra limitazione è che la mia soluzione sul browser Internet Explorer non è funzionante. Questo perchè le classi di JavaScript (come per esempio la classe 'validator.js') in Internet Explorer non sono supportate.

5 Consuntivo



16	1	Documentazione	05/09/18	26/10/18	
5	2	Avvio	05/09/18	19/09/18	
1	40	Intervista cliente	05/09/18	05/09/18	
1	41	Risoluzione dei dubbi	07/09/18	07/09/18	40
1	42	Analisi del dominio	12/09/18	12/09/18	41
1	3	Analisi dei requisiti	14/09/18	14/09/18	42
1	43	Specifica dei requisiti	19/09/18	19/09/18	3
3	11	Progettazione	21/09/18	28/09/18	2
1	13	Diagramma Gantt	21/09/18	21/09/18	
1	12	Progettazione interfaccie	26/09/18	26/09/18	13
1	48	Progettazione files csv	28/09/18	28/09/18	12
8	14	Programmazione	03/10/18	26/10/18	11
8	15	Sistema di interfacciamento con l..	03/10/18	27/10/18	
8	16	Sistema validazione dati	03/10/18	27/10/18	
8	17	Sistema di query sul file csv	03/10/18	27/10/18	
8	18	Sistema di register	03/10/18	27/10/18	
8	19	Sito web	03/10/18	27/10/18	
0	20	Finito Progetto	31/10/18	31/10/18	19
2	52	Testing	31/10/18	02/11/18	14
1	53	Test registrazioni	31/10/18	31/10/18	
1	54	Test delle falle nella sicurezza	02/11/18	02/11/18	53
1	44	Consegna	07/11/18	07/11/18	
1	45	Manutenzione e controlli	07/11/18	07/11/18	54
0	46	Consegna progetto e documentazione	09/11/18	09/11/18	45

Figura 16 - Diagramma di Gantt consuntivo del progetto

Si può notare che la parte relativa all'implementazione del progetto è cambiata radicalmente. Questo perchè diverse task le ho svolte a sbalzo e non ho finito del tutto prima un attività per poi passare alla prossima.

6 Conclusioni

Lo scopo di questo progetto era soltanto di far capire ai professori le conoscenze che avevamo ed il come le mettavamo in pratica al fine di avere un mandato consono alle nostre possibilità come prossimo progetto.

6.1 Sviluppi futuri

Uno sviluppo futuro molto utile sarebbe il salvare i dati in un database al posto che salvarli all'interno di file csv. Per collegarsi al server MySql basterebbe usare la classe *PDO*[1]oppure la classe *mysqli*[2].
Un altra feature molto utile sarebbe poter fare il login per poter visualizzare più informazioni relative a questo concorso, come per esempio i giorni mancanti, il numero di partecipanti ed il/i vincitori quando è terminato.

6.2 Considerazioni personali

In questo progetto ho imparato molte cose relative al linguaggio di programmazione server-side chiamato 'Php': Ho imparato bene ad usare le sessioni ($_SESSION), adesso (grazie alla pratica) mi destreggio bene con l'invio e la validazione dei dati sia da lato client sia da lato server. Ho anche imparato ad utilizzare jQuery (libreria che semplifica e velocizza di molto la scrittura di codice JavaScript) per la creazione di animazioni e per lo scripting in generale.

[1] http://php.net/manual/en/class.pdo.php
[2] http://php.net/manual/en/class.mysqli.php

Credo che la mia classe (I3AC) sia stata molto più avvantaggiata rispetto all'altra metà grazie al modulo 133 (ovvero programmazione in Php) dove abbiamo svolto diversi esercizi con *features* simili a quelle necessarie in questo progetto.
Questo progetto inoltre mi ha aiutato rinfrescarmi la memoria sulla lettura diagramma di Gantt e sulla sua relizzazione tramite dei programmi appositi (nel mio caso ho utilizzato GanttProject).

7 Bibliografia

7.1 Sitografia

- https://stackoverflow.com/questions/3797239/insert-new-item-in-array-on-any-position-in-php, *Insert new item in array on any position in php,*01-11-2018
- https://stackoverflow.com/questions/53105054/javascript-regex-nothing-to-repeaterror/53105153#53105153, *Javascript regex do nothing to repeat error,* 01-11-2018
- https://regex101.com/, *Online regex tester and debugger,* Consultato più volte durante tutto l'arco dell'implementazione progetto
- http://php.net/, *PHP: Hypertext Preprocessor,* Consultato più volte durante tutto l'arco dell'implementazione del progetto
- https://materializecss.com/, *Documentation – Materialize,* Consultato più volte durante tutto l'arco dell'implementazione del progetto

8 Allegati

- Diari di lavoro
- Codici sorgente
- Prodotto