Mattia Ruberto

Scuola Arti e mestieri  Via Trevano

Sito Web con pagina registrazione e verifica dati

# 1 Introduzione

## Informazioni sul progetto

Allievo: Mattia Ruberto I3AA

Docenti: Luca Muggiasca, Adriano Barchi, Francesco Mussi e Misha Cattaneo

Scuola: Arti e Mestieri Trevano Sezione Informatici

Materia: Modulo 306

Data inizio: 05.09.2018

Data fine: 07.11.2018

## Abstract

This project is a four page website, the welcome page, the registration page, the data control page and the final page of all the registrations made during the day. The user can register, on the first page there is a registrati nutton that when pressed brings to the registration page, here is the form to fill out and two buttons correggi and registra. Correggi takes you to the previous page and allows you to edit the form while registra recording automatically saves the data on the csv files and takes you to the final page with all recordings.

## Scopo

Lo scopo principale di questo progetto è quello di introdurci nel mondo dei progetti facendo sì che iniziamo a capire come condurre un progetto i tutti i dettagli dall’analisi alla consegna e alla presentazione, questo progetto ci consente di imparare come funzionano le varie fasi di un progettazione e ci consentirà di essere indipendenti nei prossimi progetti. Per imparare come condurre un progetto dovremo creare un sito internet che permette all’utente di registrarsi.

## Analisi del dominio

Questo progetto fa parte del modulo 306 della scuola arti e mestieri Trevano sezione informatica e viene assegnato come primo assaggio alla progettazione di un lavoro su lungo periodo. La richiesta da parte del cliente, in questo caso i docenti è stata quella di sviluppare un sito web con quattro pagine il cui era possibile registrarsi. È un progetto il cui scopo principale è quello di permettere di imparare e destreggiarsi meglio con le tecniche di progettazione e tutte le categorie riguardanti questo campo come l’analisi dei requisiti, pianificazione del lavoro, test di funzionamento e ecc,… Progetti simili a questo ne esistono a migliaia molto migliori ma l’obbiettivo di questo non è di rivoluzionare niente ma di capire le tecniche di progettazione. Il prodotto non deve avere un grado di sicurezza molto alto dato che le informazioni trattate non sono private e anche per quanto riguarda le prestazioni minime non ha bisogno di tanto

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| ID | REQ-001 |
| Nome | Pagina di benvenuto |
| Priorità | 2 |
| Versione | 1.0 |
| Note |  |
| Sub-ID | Requisito |
| Sotto Requisiti |  |
| 001 | All’entrata dovrà esserci scritto un benvenuto. |
| 002 | Bottone “registrati” che ti porta alla pagina per registrarsi. |

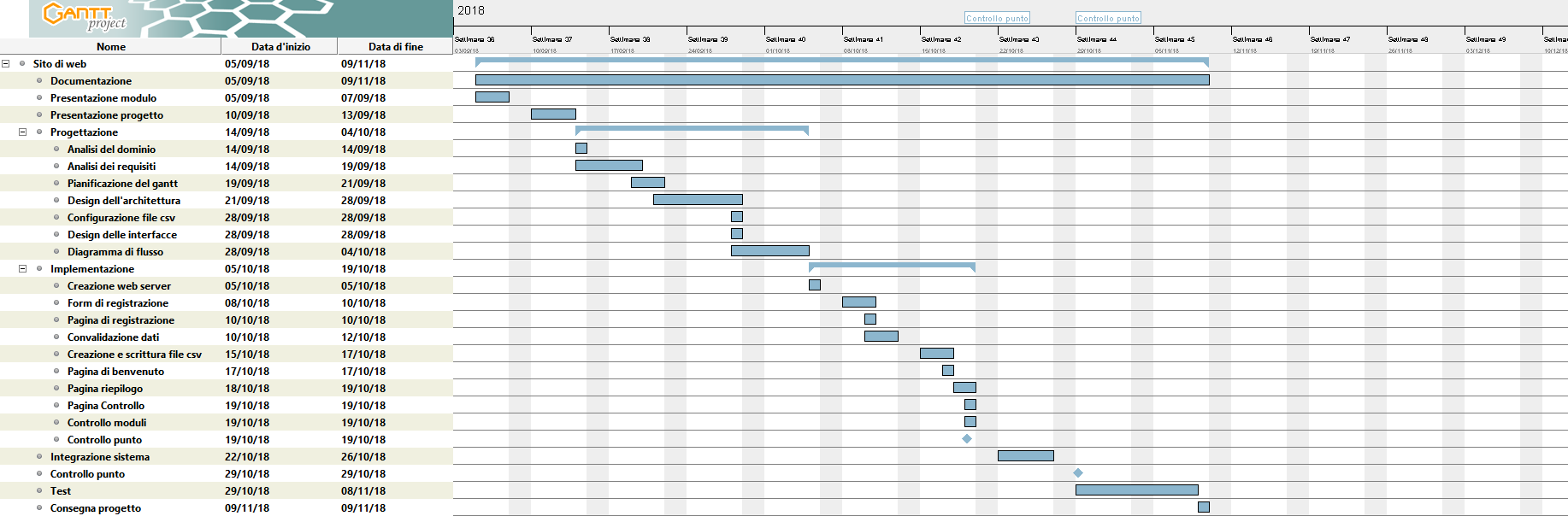
|  |  |
| --- | --- |
| ID | REQ-002 |
| Nome | Pagina di Registrazione |
| Priorità | 1 |
| Versione | 1.0 |
| Note |  |
| Sub-ID | Requisito |
| Sotto Requisiti |  |
| 001 | Form con i dati da inserire per registrarsi, vedi tabella. |
| 002 | Tutti i dati devono essere convalidati, senza però utilizzare tool predefiniti dal browser. |
| 003 | I dati obbligatori sono segnati con un \*, mentre quelli non obbligatori non hanno l’asterisco. |
| 004 | Bottone “avanti” che permette di proseguire con la pagina di controllo dei dati se tutti i campi sono stati riempiti e convalidati. |
| 005 | Bottone “cancella” che permette di cancellare tutti i dati inseriti nel form. |

|  |  |
| --- | --- |
| ID | REQ-003 |
| Nome | Pagina di controllo dei dati inseriti nel form |
| Priorità | 1 |
| Versione | 1.0 |
| Note |  |
| Sub-ID | Requisito |
| Sotto Requisiti |  |
| 001 | Form con dati inseriti che si possono solo leggere. |
| 002 | Bottone “correggi” che permette di tornare indietro alla pagina iniziale e modificare i dati. |
| 003 | Bottone “Registra” che registra l’utente scrivendo i dati sui file CSV inoltre aggiunge i campi data e ora insieme ai dati. |

|  |  |
| --- | --- |
| ID | REQ-004 |
| Nome | Salvataggio dati sui due file CSV |
| Priorità | 1 |
| Versione | 1.0 |
| Note |  |
| Sub-ID | Requisito |
| Sotto Requisiti |  |
|  |  |
| 001 | Controllare se i file CSV con tutte le registrazioni e quello con le registrazioni giornaliere esiste, se non ci sono crearli. |
| 002 | I due file CSV dovranno essere nella cartella “registrazione” sotto la cartella principale. |
| 003 | Primo file con tutte le registrazione effettuate che si chiama Registrazioni\_tutte.csv. |
| 004 | Secondo file con tutte le registrazioni giornaliere, uno per ogni giorno che si chiama Registrazione\_aaaa-mm-gg.csv. |
| 005 | Deve essere possibile creare i file csv. |
| 006 | Deve essere possibile ricavare i dati dal csv. |

|  |  |
| --- | --- |
| ID | REQ-005 |
| Nome | Pagina che mostra dati finali registrati |
| Priorità | 2 |
| Versione | 1.0 |
| Note |  |
| Sub-ID | Requisito |
| Sotto Requisiti |  |
|  |  |
| 001 | Viene mostrato il form con tutti i dati che sono stati salvati |
| 002 | I dati mostrati vengono letti dal file CSV |

## Pianificazione



## 1.8 Analisi dei mezzi

### 1.8.1 Software

I software che vengono utilizzati per questo progetto sono:

* GanttProject-2.8
* Sublime Text 3
* Word 2013
* PowerPoint 2013
* Apache-2.4.27
* Php-7.1.9
* Notepad++
* StarUML

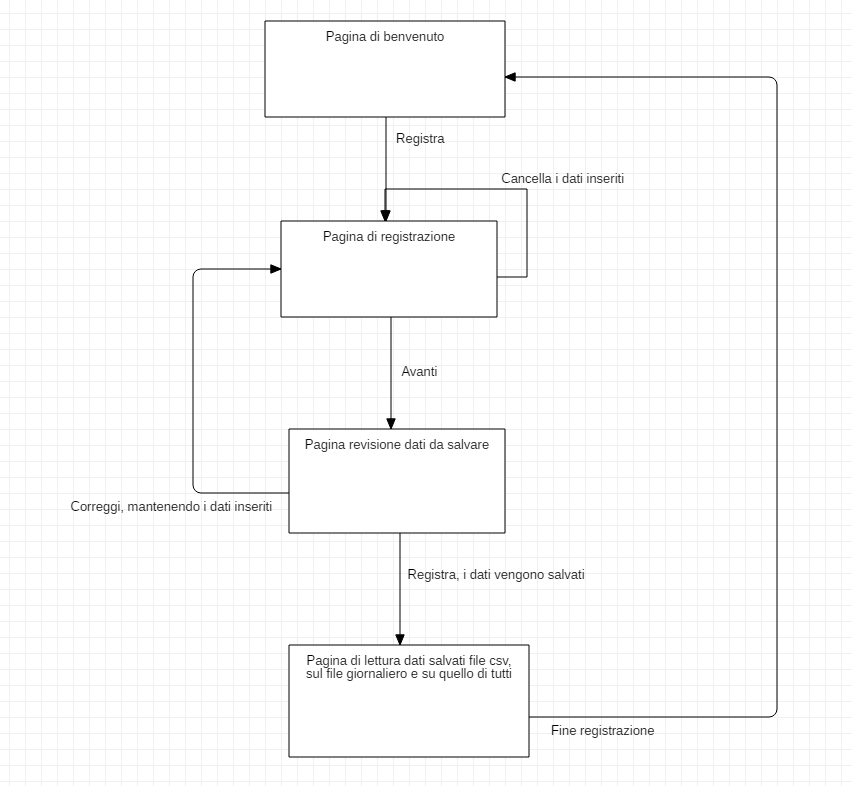
### 1.8.2 Hardware

L’hardware che viene utilizzato per la creazione di questo progetto sono:

* Intel® Core™ i7-6700HQ CPU 2.60GHz
* 16.0 GB DDR4
* 64bit, processore x64
* Windows 10

# 2.0 Progettazione

## 2.1 Design dell’architettura del sistema



## 2.2 Configurazione file csv

Ora ---> hh:mm:ss

Data ---> yyyy-mm-dd

Nome ---> stringa da massimo 50 caratteri

Cognome ---> stringa da massimo 50 caratteri

Data nascita ----> yyyy-mm-dd

Indirizzo ----> stringa da massimo 50 caratteri

Città ----> stringa da massimo 50 caratteri

No. Telefono ----> intero da un minimo di 10 a 13

No. Civico ---> Intero di massimo 4 cifre

Nap ---->

Email ----> minimo un punto e una @

Genere -----> radio button, M o F

Professione ---> stringa da massimo 50 caratteri

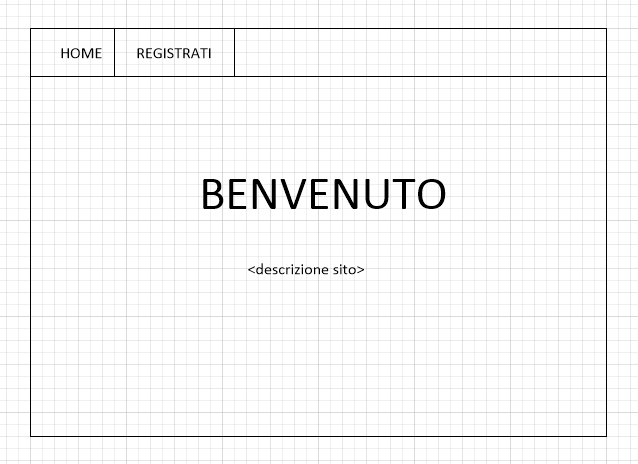
Hobby ----> stringa da massimo 500 caratteri

Configurazione del file csv:

< Ora>;< Data>;< Nome>;<Cognome>;<Data di nascita>;<Indirizzo>;<Città>;<No. Telefono>;<No. Civico>;<Nap>;<Email>;<Genere>;<Professione>;<Hobby>

## 2.3 Design delle interfacce

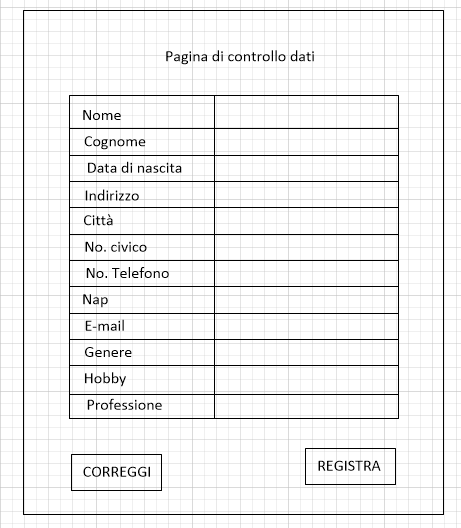
Pagina di benvenuto.



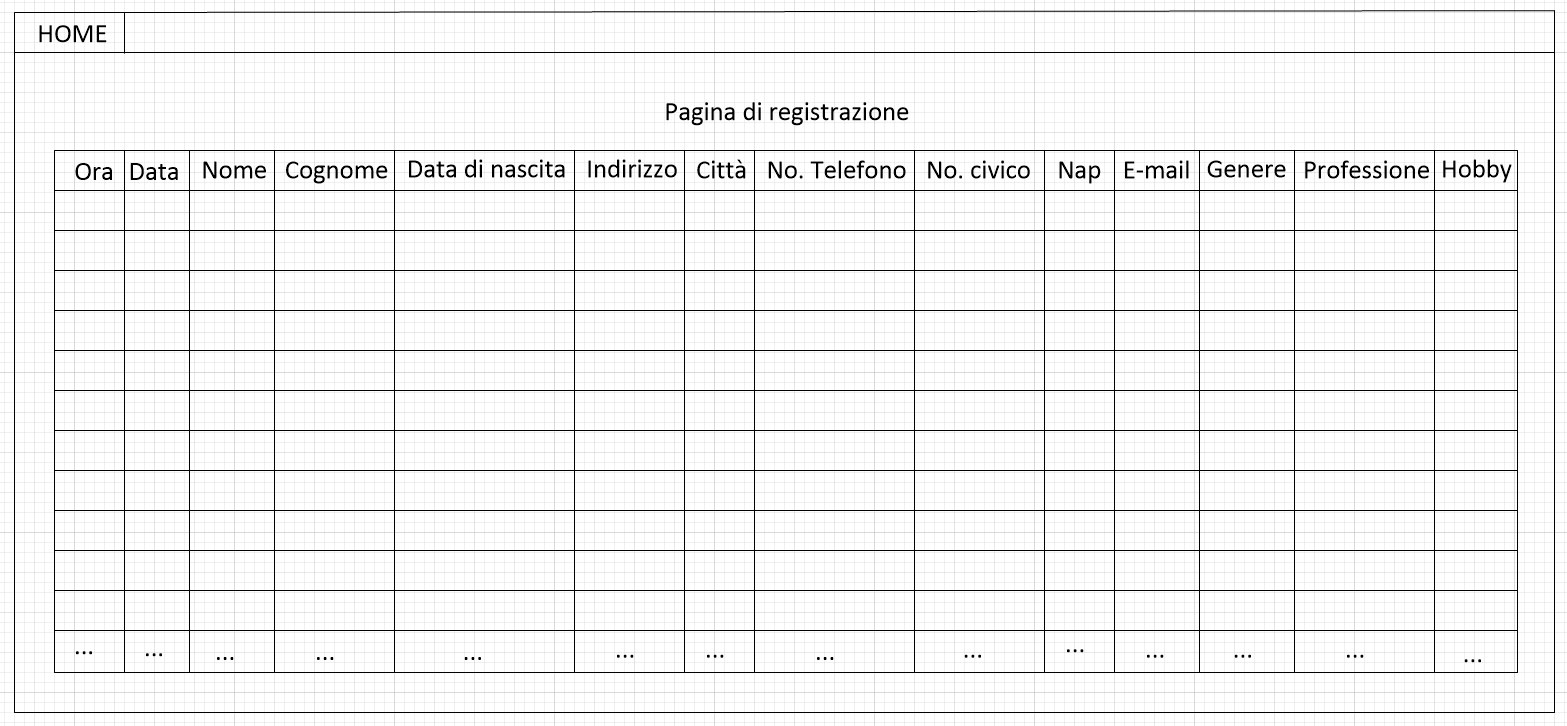
Pagina di registrazione



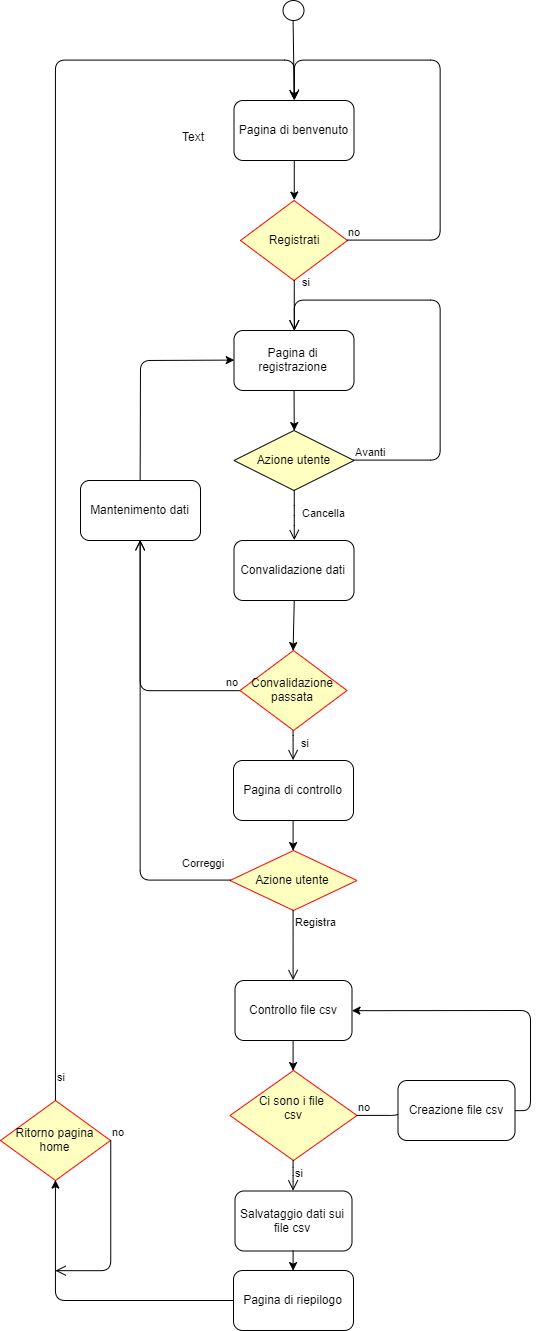
Pagina di controllo dei dati



Pagina riepilogo dati salvati



## 2.4 Diagramma di flusso



# 3.0 Implementazione

## 3.1 Pagina di benvenuto

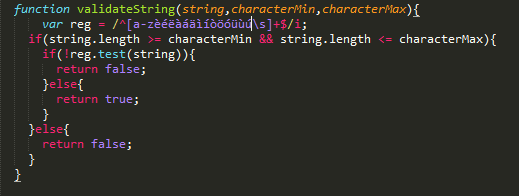
La pagina di benvenuto del mio sito è molto semplice infatti si presenta solo con il titolo “Benvenuto”, una piccola spiegazione del sito e il bottone registrati che ti porta alla pagina di registrazione.

## 3.2 Pagina di registrazione

Nella pagina di registrazione le parte più significativa dell’implementazione sono le funzioni base utilizzate per la convalidazione dei dati.

La convalidazione dei campi avviene interamente lato client tramite javascript, in questo sito vengono eseguite due convalidazioni, una è istantanea al cambiamento del contenuto del campo mentre l’altra avviene quando viene premuto il tasto avanti. Per fare ciò sono state implementate delle funzioni di base.

## **Validate String**



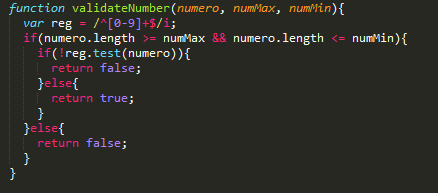
Questa funzione controlla che la stringa che gli viene passate contenga solo lettere, riceve tre parametri, il primo parametro è la stringa che deve essere controllata, il secondo e il terzo parametro sono il numero minimo e il numero massimo di caratteri che la stringa può possedere.

La funzione usa la funzione usa il **regex** per controllare che all’interno della stringa ci siano solo lettere, il regex in questione è questo:

**var reg = /^[a-zèéëàáäìíòöóüùú\s]+$/i;**.

La funzione si chiama **test** e controlla che la stringa passata rispetta il regex. Se la funzione conclude che la stringa è corretta ritorna “true” altrimenti ritorna “false”.

**Validate Number**



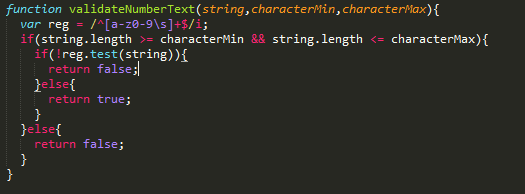
Questa funzione esegue lo stesso compito della prima funzione solo che invece che controllare se la stringa contiene solo lettere controlla che nella stringa siano presenti solamente numeri, anche questa funzione contiene li stessi parametri e esegue lo stesso procedimento della funzione **validate string** ma il **regex** utilizzato è differente:

**var reg = /^[a-z0-9\s]+$/i**

**Validate Number Text**

La funzione **validateNumberText** è un misto delle funzioni **validateNumber** e **validateString**, infatti la stringa che viene inserita in questa funzione può contenere sia numeri che lettere semplici, questa funzione viene utilizzata in particolare per l’indirizzo, il resto della funzione è identico alle altre ma il regex è differente:

**var reg = /^[0-9a-z]+$/i**

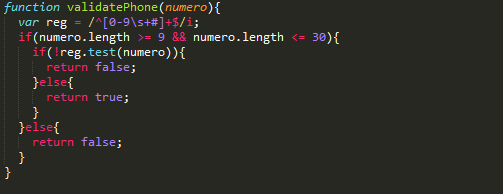


**Validate Phone**

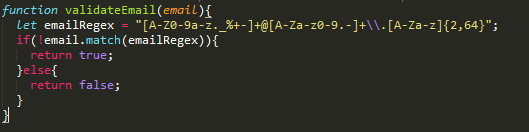
Quest’ultima funzione si occupa di controllare il numero di telefono, infatti oltre ad accettare tutti i numeri accetta anche il “+” e il “#” e la grandezza del numero di telefono può variare da 9 a 30 caratteri, la funzione è praticamente uguale alle altre ma anche in questo caso il **regex** cambia:

**var reg = /^[0-9\s+#]+$/i**

Alla fine se il numero è corretto ritorna “true” altrimenti ritorna false “false”.



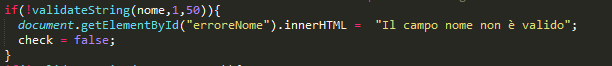
**Validate Email**



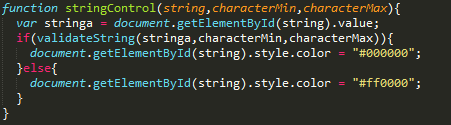
Quest’ultima funzione viene utilizzata per convalidare l’email, esegue le stesse funzioni delle precedenti ma in questo caso la funzione utilizzata è il **match** il **regex** è differente:

**let emailRegex = "[A-Z0-9a-z.\_%+-]**[**+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,64}**](mailto:+@[A-Za-z0-9.-]+\\.%5bA-Za-z%5d%7b2,64%7d)**"**

**Utilizzo delle convalidazioni base**



Questo è un pezzo di codice che viene eseguito dalla funzione che parte quando il bottone avanti viene premuto, la funzione fa passare una serie di if simile a questo per ogni campo. L’if si occupa tramite una funzione di convalidazione di base in questo cado **validateString()**, se la convalidazione non è corretta viene stampato il **messaggio di errore** del campo in questione e la variabile **ceck** viene settata a **false** in questo modo il programma non va avanti.



Questo è un altro esempio di come vengono utilizzate le funzioni di base, la funzione **stringControl** viene eseguita quando avviene un **onchange** su un campo come il nome ad esempio e se **validateString** ritorna true il colore resta nero altrimenti il testo viene colorato di rosso con il codice **document.getElementById(string).style.color = “#ff0000”.** Oltre a questa funzione c’è anche la funzione stringTextNumber, numberControl, phoneControl e controlEmail.

**Reset Form**



Questa funzione si occupa di resettare il form quando viene cliccato il bottone cancella, la funzione va a prendere l’id del form e la funzione **reset()** si occupa di svuotare tutti i campi.

**Submit**



La seguente funzione in **javascript** si occupa di richiamare il form di php tramite il suo id e di fare partire il **submit**.

## 3.3 Pagina di controllo

Nella pagina di controllo le parti di implementazione più interessanti sono quella di scrittura dei dati sul file csv e la funzione che permette di tornare in dietro quando viene schiacciato il bottone correggi.

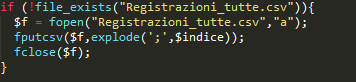
**Back to Page**



Questa funzione parte quando il **bottone** viene premuto, la funzione è molto semplice infatti tramite **javascript** con la funzione **window.location.href** che si sposta nel percorso che gli viene messo ma in questo caso non gli viene messo un percorso ma un comando che va a vedere la **cronologia** delle pagine viste torna in dietro a quella precedente.

Da adesso in poi tutti questi passaggi vengono effettuati per poter scrivere i dati sui due file csv, per poter scrivere sui file csv non bisogna scordarsi di controllare se i file **esistono**, se no di **crearli** e dopo aver preparato tutti i dati da inserire si può proseguire. Tutto questo viene eseguito con **php** quando viene schiacciato il bottone **registra** che è di tipo **submit**.

**Creazione file csv**



Con questo **if** il codice di php va a controllare se il file **csv** esiste tramite il metodo **file\_exixsts(<nome file>)**,

se non esiste tramite il metodo **fopen(<nome file>,<funzione>)** che va aprire il file, il secondo parametro sta per l’azione che si andrà a fare sul file, si potrebbe mettere anche **w** che sta per scrittura e **r** che sta per lettura mentre con **a** si possono fare entrambi.

Il metodo **fopen()** apre il file che se non esiste lo va a creare automaticamente, poi viene inserito l’indice del file csv tramite il metodo **fputcsv(<file aperto>, explode(<carattere>,<indice>) )** che dopo avergli passato il file appena aperto come primo parametro e i dati con il carattere che divide i dati come secondo parametro nel metodo **explode(<carattere>,<indice>)** in fine si chiude il file con la funzione **fclose(<file aperto>)**. Questa azione viene ripetuta anche per il file giornaliero.

**Scrittura sul file csv**



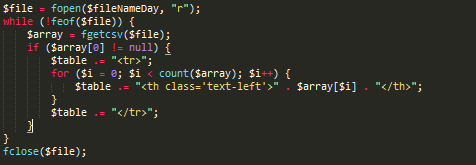
Per andare a scrivere i campi sul file csv è lo stesso procedimento che si è andato a fare per mettere l’indice nel file csv solo che questa volta nei dati da inserire non verrà messo l’indice ma una stringa con al suo interno tutti i campi separati dal punto e virgola.

**Cambiamento di pagina**



Tramite la funzione header si va a cambiare la pagina, questa funzione è molto semplice infatti basterà richiamarla e al suo interno scriver **“Location: <Percorso Pagina>”**.

## 3.4 Pagina registrazioni giornaliere



Per andare a leggere il file csv viene eseguito un while che passerà il file riga per riga e si ferma quando la funzione **feof(<file>)** che ritorna true quando il file è finito, all’interno del file viene associata una variabile chiama array che avrà la funzione di un array attraverso il metodo **fgetcsv(<file>)** che ritorna una matrice con al suo interno i campi del csv. Dopodiché viene controllato se il primo valore dell’array non è vuoto e se non è vuoto verrà eseguito un **for** grande come la variabile array e questo for inserirà ogni campo in una cella della tabella. Alla fine viene chiuso il file e stampata la tabella.

# 4.0 Test

## 4.1 Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-01-002 | **Nome:** | Bottone registrati. |
| **Descrizione:** | Verificare che il bottone registrati funziona correttamente. | | |
| **Prerequisiti:** | Aver aperto il sito alla pagina di benvenuto. | | |
| **Procedura:** | 1. Aprire il sito alla pagina di i benvenuto. 2. Premere il bottone registrati. | | |
| **Risultati attesi:** | Dopo aver schiacciato il bottone il sito deve aprire la pagina di registrazione. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-02-004 | **Nome:** | Compilazione corretta senza hobby e professione. |
| **Descrizione:** | Verificare che quando il bottone avanti viene premuto il sito prosegua se tutti i campi sono stati compilati correttamente tranne hobby e professione. | | |
| **Prerequisiti:** | Trovarsi nella pagina di registrazione con il form da compilare. | | |
| **Procedura:** | 1. Compilare campo nome: Mattia 2. Compilare campo cognome: Ruberto 3. Compilare campo data di nascita: 11.04.2001 4. Compilare campo indirizzo: Via Castello 5. Compilare campo città: New York 6. Compilare campo numero civico: 77a 7. Compilare campo numero di telefono: +41 69 456 00 76 8. Compilare campo cap: 2332 9. Compilare campo email: [mattia@gmail.ch](mailto:mattia@gmail.ch) 10. Compilare campo genere: F 11. Premere il bottone avanti | | |
| **Risultati attesi:** | Il sito deve portare l’utente alla pagina di controllo. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-02-004 | **Nome:** | Compilazione scorretta senza hobby e professione. |
| **Descrizione:** | Verificare che quando il bottone avanti viene premuto il sito non prosegua se tutti i campi non sono stati compilati tranne hobby e professione vengono compilati in modo scorretto, per ogni campo deve uscire un messaggio di errore. | | |
| **Prerequisiti:** | Trovarsi nella pagina di registrazione con il form da compilare. | | |
| **Procedura:** | 1. Compilare campo nome: Mattia23123 2. Compilare campo cognome: Ruberto2332 3. Compilare campo data di nascita: 11.04.20013 4. Compilare campo indirizzo: Via Castello 3232 5. Compilare campo città: New York33 6. Compilare campo numero civico: 77a33 7. Compilare campo numero di telefono: +41 69 456 00 76!! 8. Compilare campo cap: 2332dfs 9. Compilare campo email: [ciao.ciao](mailto:mattia@gmail.ch) 10. Compilare campo genere: F 11. Premere il bottone avanti | | |
| **Risultati attesi:** | Il sito va avanti ma sotto i bottoni esce un avviso di errore per ogni campo compilato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-02-004 | **Nome:** | Compilazione corretta di tutti i campi. |
| **Descrizione:** | Verificare che quando il bottone avanti viene premuto il sito non prosegua se tutti i campi sono stati compilati in modo errato, per ogni campo deve uscire un messaggio di errore. | | |
| **Prerequisiti:** | Trovarsi nella pagina di registrazione con il form da compilare. | | |
| **Procedura:** | 1. Compilare campo nome: Mattia 2. Compilare campo cognome: Ruberto 3. Compilare campo data di nascita: 11.04.2001 4. Compilare campo indirizzo: Via Castello 5. Compilare campo città: New York 6. Compilare campo numero civico: 77a 7. Compilare campo numero di telefono: +41 69 456 00 76 8. Compilare campo cap: 2332 9. Compilare campo email: [mattia@gmail.ch](mailto:mattia@gmail.ch) 10. Compilare campo genere: F 11. Compilare campo hobby: Tennis 12. Compilare campo professione: Informatico 13. Premere il bottone avanti | | |
| **Risultati attesi:** | Il sito non cambia pagina e sotto i bottoni viene stampato un messaggio di errore per ogni campo compilato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-02-004 | **Nome:** | Compilazione scorretta di tutti i campi. |
| **Descrizione:** | Verificare che quando il bottone avanti viene premuto il sito non prosegua se tutti i campi vengono compilati in modo scorretto, per ogni campo deve uscire un messaggio di errore. | | |
| **Prerequisiti:** | Trovarsi nella pagina di registrazione con il form da compilare. | | |
| **Procedura:** | 1. Compilare campo nome: Mattia23123 2. Compilare campo cognome: Ruberto2332 3. Compilare campo data di nascita: 11.04.20013 4. Compilare campo indirizzo: Via Castello 3232 5. Compilare campo città: New York33 6. Compilare campo numero civico: 77a33 7. Compilare campo numero di telefono: +41 69 456 00 76!! 8. Compilare campo cap: 2332dfs 9. Compilare campo email: [ciao.ciao](mailto:mattia@gmail.ch) 10. Compilare campo genere: F 11. Compilare campo hobby: Tennis&/\*/\*Q% 12. Compilare campo professione: +\*çInformatico 13. Premere il bottone avanti | | |
| **Risultati attesi:** | Il sito non va avanti ma sotto i bottoni esce un avviso di errore per ogni campo compilato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-02-005 | **Nome:** | Bottone cancella. |
| **Descrizione:** | Verificare che quando il bottone cancella viene premuto, il form compilato viene cancellato. | | |
| **Prerequisiti:** | Trovarsi nella pagina di registrazione con il form da compilare | | |
| **Procedura:** | 1. Compilare campo nome: Mattia 2. Compilare campo cognome: Ruberto 3. Compilare campo data di nascita: 11.04.2001 4. Compilare campo indirizzo: Via Castello 5. Compilare campo città: New York 6. Compilare campo numero civico: 77a 7. Compilare campo numero di telefono: +41 69 456 00 76 8. Compilare campo cap: 2332 9. Compilare campo email: [mattia@gmail.ch](mailto:mattia@gmail.ch) 10. Compilare campo genere: F 11. Compilare campo hobby: Tennis 12. Compilare campo professione: Informatico 13. Premere il bottone cancella | | |
| **Risultati attesi:** | Il contenuto di tutti i campi del form vengono cancellati completamente. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-03-002 | **Nome:** | Bottone correggi. |
| **Descrizione:** | Verificare che quando il bottone correggi viene premuto nella pagina di registrazione è possibile correggere i campi. | | |
| **Prerequisiti:** | Trovarsi nella pagina di controllo e voler modificare i propri campi. | | |
| **Procedura:** | 1. Premere il bottone correggi | | |
| **Risultati attesi:** | Il sito torna in dietro alla pagina di registrazione per dare la possibilità all’utente di correggere i campi. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-03-003 | **Nome:** | Bottone registra. |
| **Descrizione:** | Verificare che quando il bottone registra viene premuto il sito si sposta sulla pagina finale. | | |
| **Prerequisiti:** | Trovarsi nella pagina di controllo e voler registrare il proprio utente. | | |
| **Procedura:** | 1. Premere il bottone registra | | |
| **Risultati attesi:** | Il sito si sposta sulla pagina finale e mostra tutte le registrazione effettuate nell’arco del giorno. | | |

## 4.2 Risultati test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Nome** | **Descrizione** | **Risultato** |
| TC**-**001 | Bottone registrati | Verificare che il bottone registrati funziona correttamente. | **CORRETTO** |
| TC-002 | Compilazione corretta senza hobby e professione | Verificare che quando il bottone avanti viene premuto il sito prosegua se tutti i campi sono stati compilati correttamente tranne hobby e professione. | **CORRETTO** |
| TC-003 | Compilazione scorretta senza hobby e professione | Verificare che quando il bottone avanti viene premuto il sito non prosegua se tutti i campi non sono stati compilati tranne hobby e professione vengono compilati in modo scorretto, per ogni campo deve uscire un messaggio di errore. | **CORRETTO** |
| TC-004 | Compilazione corretta di tutti i campi | Verificare che quando il bottone avanti viene premuto il sito non prosegua se tutti i campi sono stati compilati in modo errato, per ogni campo deve uscire un messaggio di errore. | **CORRETTO** |
| TC-005 | Compilazione scorretta di tutti i campi | Verificare che quando il bottone avanti viene premuto il sito non prosegua se tutti i campi vengono compilati in modo scorretto, per ogni campo deve uscire un messaggio di errore. | **CORRETTO** |
| TC-006 | Bottone cancella | Verificare che quando il bottone cancella viene premuto, il form compilato viene cancellato. | **CORRETTO** |
| TC-007 | Bottone correggi | Verificare che quando il bottone correggi viene premuto nella pagina di registrazione è possibile correggere i campi. | **CORRETTO** |
| TC-008 | Bottone registra | Verificare che quando il bottone registra viene premuto il sito si sposta sulla pagina finale. | **CORRETTO** |

# 5.0 Consuntivo

# 6.0 Conclusioni

In conclusione questo progetto non cambierà sicuramente il mondo infatti sono tutte cose viste e riviste ma non è inutile e non è stata sicuramente una perdita di tempo dato che a livello personale è ha permesso di crescere personalmente nella gestione di un progetto. Grazie a questo progetto ho potuto entrare nell’ottica progettazione, vedere ad esempio che l’implementazione non è l’unica parte importante di un progetto ma al contrario solo con essa non si va da nessuna parte, ho iniziato a capire come programmare i propri tempi di lavoro e soprattutto nella gestione di essi.

## 6.1 Sviluppi futuri

Uno sviluppo futuro potrebbe innanzitutto di aggiungere un campo password al form, fare in modo che non possa essere usata più una email per ogni registrazione effettuata, aver cambiato il metodo di passaggio dei dati tra una pagina e l’altra in un modo più sicuro e dopo di che implementare una pagina di login dove ogni utente può fare il login del proprio account. Quando hai un’applicazione del genere puoi implementarla in quasi qualsiasi contesto tecnologico dato che al giorno d’oggi si ci registra per tutto.

## 6.2 Considerazioni personali

# 7.0 Bibliografia

## 7.1 Sitografia

<https://www.w3schools.com/php/func_filesystem_fputcsv.asp>

<https://www.w3schools.com/php/func_filesystem_fgetcsv.asp>

<https://www.w3schools.com/php/php_form_complete.asp>

<https://www.w3schools.com/php/php_forms.asp>

<https://www.mrwebmaster.it/php/variabili-get-post_9489.html>

<https://www.w3schools.com/jsref/jsref_regexp_test.asp>

<https://www.freeformatter.com/css-beautifier.html>

<https://www.w3schools.com/php/func_http_header.asp>

<https://www.w3schools.com/php/func_filesystem_feof.asp>

<https://www.w3schools.com/jsref/jsref_obj_regexp.asp>