

Progetto 1

Titolo del progetto: Progetto 1
Alunno/a: Gabriele Alessi
Classe: SAM I3AA
Anno scolastico: 2018/2019
Docenti responsabili: Adriano Barchi, Luca Muggiasca, Francesco Mussi

1	Introduzione	3
1.1	Informazioni sul progetto	3
1.2	Abstract	3
1.3	Scopo	3
1.4	Analisi del dominio	3
1.5	Analisi e specifica dei requisiti	4
1.6	Pianificazione	5
1.7	Analisi dei mezzi.....	6
1.7.1	Software	6
1.7.2	Hardware	6
2	Progettazione	7
2.1	Design dell'architettura del sistema	7
2.2	Design dei dati e database.....	7
2.3	Design delle interfacce	8
2.4	Design procedurale	10
3	Implementazione	11
3.1	Pagina di benvenuto	11
3.2	Pagina con form di registrazione.....	12
3.3	Pagina di controllo.....	17
3.4	Pagina di controllo del file	18
4	Test.....	21
4.1	Protocollo di test.....	21
4.2	Risultati test.....	23
5	Consuntivo.....	24
6	Conclusioni	24
6.1	Sviluppi futuri.....	24
6.2	Considerazioni personali.....	24
7	Bibliografia	24
7.1	Sitografia	24
8	Allegati.....	24

1 Introduzione

1.1 Informazioni sul progetto

- Allievo: Gabriele Alessi
Docenti e supervisori: Adriano Barchi, Luca Muggiasca, Francesco Mussi
- Scuola d'Arti e Mestieri di Trevano , Sezione informatica, Modulo 306
- Data inizio: 05.09.2018
Data fine: 09.11.2018

1.2 Abstract

The target of this project is to preserve a user's data simply and fast. This project will also be useful for learning the basics on how to manage a project. Nowadays it is important to be able to store data in a safe place, so it is a useful product that can be used in various situations. Everything will be done mainly through the use of a PC and a software to write the code. The final product consists in creating a website with a registration form and data storage.

1.3 Scopo

L'obiettivo del progetto è realizzare un sito web che utilizza un form di registrazione con vari campi per memorizzare i dati di un utente in un file. L'intera progettazione del mandato sarà fatta insieme ai colleghi della SAM I3, quindi risulterà anche un'esperienza utile per imparare a fare un lavoro del genere in ambito professionale.

1.4 Analisi del dominio

Un prodotto di questo tipo viene utilizzato spessissimo al giorno d'oggi. Basti pensare ai social network, ai moduli virtuali da compilare o alla creazione di un nuovo profilo per un servizio online. Questo servizio funziona esclusivamente nella rete internet e per realizzarlo è necessario solamente un computer con uno spazio web. Ovviamente esistono infinite soluzioni e prodotti utili per memorizzare dei dati, ad esempio si usano dai più comuni pezzi di carta per passare alle grandi banche dati. Solitamente gli utenti sono persone che hanno bisogno di un posto sicuro in cui conservare delle informazioni per tanto tempo ma essere comunque capaci di reperirle senza difficoltà (alberghi, ristoranti, utilizzatori di un qualche servizio, ...). Questi utenti sono in cerca della soluzione più efficiente e conveniente in base al loro bisogno, quindi qualcuno può semplicemente utilizzare un software come Microsoft Excel se riesce a starci dietro. Per concludere si può dire che un prodotto di questa categoria si vede ovunque e in varie forme, quindi non c'è una soluzione definitiva che è migliore di un'altra, visto che tutto è basato dalle proprie esigenze.

1.5 Analisi e specifica dei requisiti

ID: 01	
Nome	Pagina di benvenuto
Priorità	2
Versione	1.0
Note	
Sotto requisiti	
001	Presenza di testo di benvenuto con un pulsante: "Registrati"

ID: 02	
Nome	Pagina con form di registrazione
Priorità	1
Versione	1.0
Note	Necessità di implementare la validazione di ogni campo prima della registrazione (la validazione dei campi non può essere lasciata ai meccanismi del browser ma dovrà essere implementata), e in caso di dati non validi o campi obbligatori non compilati bisognerà segnalarli e sarà impossibile proseguire
Sotto requisiti	
001	Presenza di campi obbligatori contrassegnati con un asterisco e campi opzionali
002	Presenza di un pulsante "Cancella" che azzerà tutti i campi
003	Presenza di un pulsante "Avanti" per proseguire (in caso di dati idonei)

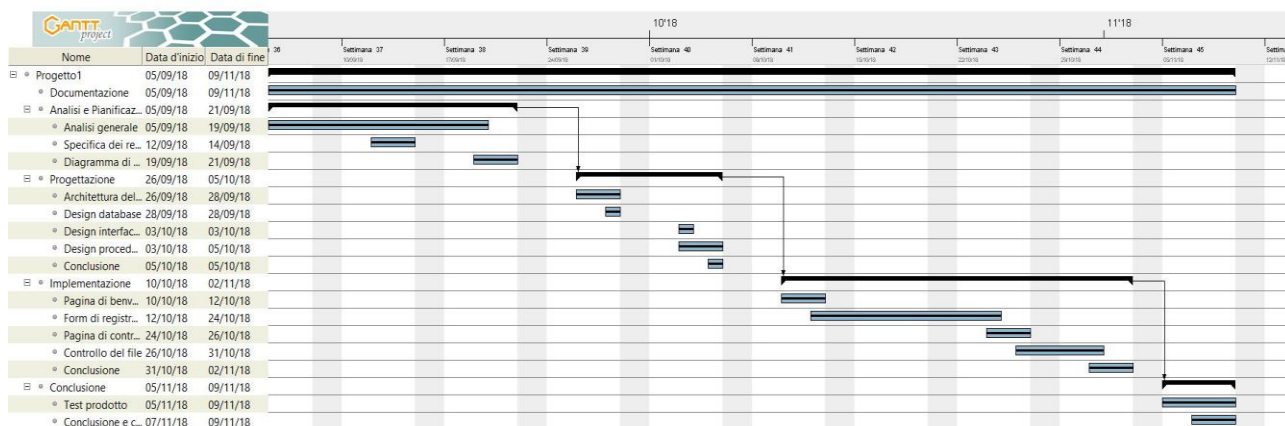
ID: 03	
Nome	Pagina per controllo dei dati
Priorità	2
Versione	1.0
Note	
Sotto requisiti	
001	Presenza di un pulsante "Correggi" che riporta al form di registrazione per correggere di dati
002	Presenza di un pulsante "Registra" che memorizza i dati

ID: 04	
Nome	Memorizzazione dei dati
Priorità	1
Versione	1.0
Note	L'applicazione valuterà se sovrascrivere i file (quindi se sono già creati) oppure crearli per poi scriverci
Sotto requisiti	
001	Presenza di una cartella "Registrazioni" sotto la cartella principale in cui situano i files
002	Aggiunta di un campo ai dati con data e ora al momento della registrazione
003	Presenza di un file con tutte le registrazioni: "Registrazioni_tutte.csv"
004	Presenza di un file con le registrazioni del giorno corrente: "Registrazione_aaaa-mm-gg.csv"

ID: 05	
Nome	Pagina che controlla i dati dal file
Priorità	2
Versione	1.0
Note	Questa pagina viene presentata alla fine della registrazione e mostra i dati letti dal file "Registrazione_aaaa-mm-gg.csv"
Sotto requisiti	
001	Presenza di un pulsante "Home" per tornare alla schermata di benvenuto

1.6 Pianificazione

La pianificazione può essere rappresentata mediante un diagramma di Gantt:



1.7 Analisi dei mezzi

In generale i mezzi utilizzati per lavorare a questo progetto sono quelli di base per sviluppare una piccola applicazione web. Quindi si parla di qualche software per disegnare l'analisi, la progettazione e scrivere il codice sorgente. Tutto ciò sarà fatto tramite un computer portatile e uno spazio web.

1.7.1 Software

La lista dei software utilizzati:

- GanttProject 2.8.9 Pilsen
- Sublime Text 3
- Microsoft Visio 2010
- Microsoft Word 2013
- Google Chrome 70.0.3538.77
- GitHub Desktop 1.4.3

1.7.2 Hardware

Personal Computer: HP ENVY Notebook

- Sistema operativo: Windows 10 Home 64 bit
- Microprocessore: Intel® Core™ i7-6500U CPU @ 2.50GHz
- Memoria di sistema: 16GB
- Scheda di sistema: 81CC KBC Version 60.26
- BIOS di sistema: F.21
- Dispositivo grafico: Intel® HD Graphics 520
- Memoria grafica: 4168 MB

2 Progettazione

La progettazione descrive l'idea della struttura del progetto, del database, il design delle interfacce web e il design procedurale.

2.1 Design dell'architettura del sistema

Il sistema è composto principalmente da:

- La cartella principale. (Progetto1);
- I file delle pagine web (HTML, PHP);
- La cartella contenente i file di stile (style: CSS);
- La cartella contenente i file di script (script: JS);
- La cartella contenente i file media (media: immagini, audio, video);
- La cartella contenente i file in cui sono conservati i dati (Registrazioni: CSV).

Sotto la cartella principale ci sono i file delle pagine web e la prima è la pagina di benvenuto, che presenta solamente un pulsante per proseguire. Proseguendo si troverà la pagina con il form di registrazione, in cui l'utente inserirà i dati nei rispettivi campi. Quindi si troverà una pagina di riepilogo per controllare i dati inseriti e decidere se si vuole andare avanti oppure tornare indietro e correggere un eventuale svista.

Finalmente i dati inseriti verranno memorizzati nei file che si trovano sotto la cartella "Registrazioni". Le registrazioni del giorno corrente vengono salvate nel file "Registrazione_aaaa-mm-gg.csv", mentre il file "Registrazioni_tutte.csv" contiene tutte le registrazioni effettuate.

In conclusione, ci sarà una pagina di controllo che legge le informazioni dal file delle registrazioni giornaliere per fare un'ultima verifica.

2.2 Design dei dati e database

La validazione dei dati verrà trattata su lato client per motivi di performance e semplicità.

Essi verranno conservati in due file CSV sotto la cartella "Registrazioni" che si trova nella cartella principale:

- Uno per le registrazioni giornaliere;
- Uno completo per le tutte le registrazioni.

Entrambi i file verranno creati in caso di necessità, quindi quando si effettuerà la primissima registrazione o la prima registrazione del giorno.

L'ordine dei dati all'interno dei file è il seguente:

Data;Nome;Cognome;Data_Nascita;Indirizzo;Num_Civico;Citta;NAP;Num_Telefono;Email;Genere;Hobby;Professione.

Guardando il prodotto finale, ci si trova inizialmente di fronte a una semplice pagina di benvenuto senza nessun tipo di input o login, bensì un pulsante che porta al form di registrazione.

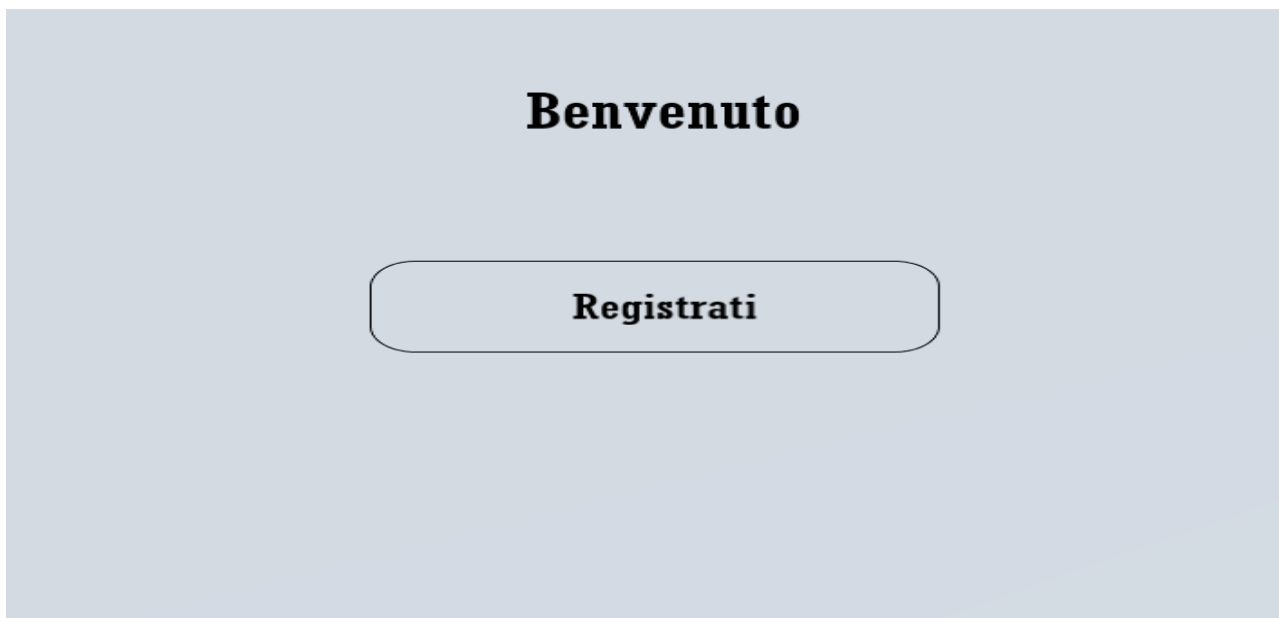
Questo form presenta i seguenti campi da compilare:

- Nome, Cognome, Indirizzo, Città:
campo di testo, obbligatorio, si accettano lettere con accento e caratteri speciali.
- Data di nascita:
campo di tipo data, obbligatorio, si accettano date che non siano successive al giorno corrente.
- Numero civico:
campo di testo, obbligatorio, si accettano fino a 4 cifre che con eventuale lettera finale.
- NAP:
campo numerico, obbligatorio, si accettano da 4 a 5 cifre.
- Numero di telefono:
campo di testo, obbligatorio, si accettano cifre, trattini e spazi con un eventuale "+" iniziale.
- E-mail
campo di testo, obbligatorio, si accetta solo il formato "testo@testo.testo".
- Genere (sesso):
campo di testo, obbligatorio, si accettano solo le lettere "M" o "F".
- Hobby, Professione:
campo di testo, opzionale, si accettano lettere con accento e caratteri speciali.

In caso di dati non validi o campi obbligatori non compilati, l'applicazione segnalerà l'errore senza proseguire al punto successivo.

2.3 Design delle interfacce

In generale, il layout del prodotto sarà pensato in modo semplice e intuitivo e pulito. Il design delle pagine web può essere rappresentato dalle seguenti immagini:



Si parte con una semplicissima pagina di benvenuto con un pulsante per iniziare la registrazione.

A screenshot of a registration form on a light blue background. The form consists of a list of labels on the left, each followed by a white rounded rectangular input field on the right. The labels are: **Nome***, **Cognome***, **Data di nascita***, **Indirizzo***, **Numero civico***, **Città***, **NAP***, **Numero di telefono***, **E-mail***, **Genere (sesso)***, **Hobby**, and **Professione**. Below the input fields, on the left, is the text ***campi obbligatori**. At the bottom left is a white rounded rectangular button with a thin black border containing the word **Cancella** in a bold, black serif font. At the bottom right is a white rounded rectangular button with a thin black border containing the word **Avanti** in a bold, black serif font.

Il form di registrazione presenta i vari campi da compilare scritti in modo allineato e semplice. Quindi si trovano due pulsanti utili per azzerare tutti i campi o proseguire. In caso di dati non validi o campi obbligatori vuoti, l'errore sarà segnalato nella parte bassa della pagina.

Nome: Mario
Cognome: Rossi
Data di nascita: 01.01.1970
Indirizzo: Via Trevano
Numero civico: 10
Città: Lugano
NAP: 6900
Numero di telefono: 0761234567
E-mail: mario.rossi@email.ch
Genere (sesso): M
Hobby: Cucina
Professione: Docente

Correggi

Registra

Successivamente si trova la pagina di controllo dei dati appena inseriti, in cui vengono semplicemente scritte le informazioni. Il pulsante “Correggi” permette di tornare indietro (al form di registrazione) e cambiare i dati, mentre il pulsante “Registra” memorizza i dati nei file CSV e prosegue alla prossima pagina di controllo.

Registrazione_aaaa-mm-gg.csv

<dati letti dal file>

<dati letti dal file>

<dati letti dal file>

<dati letti dal file>

<dati letti dal file>

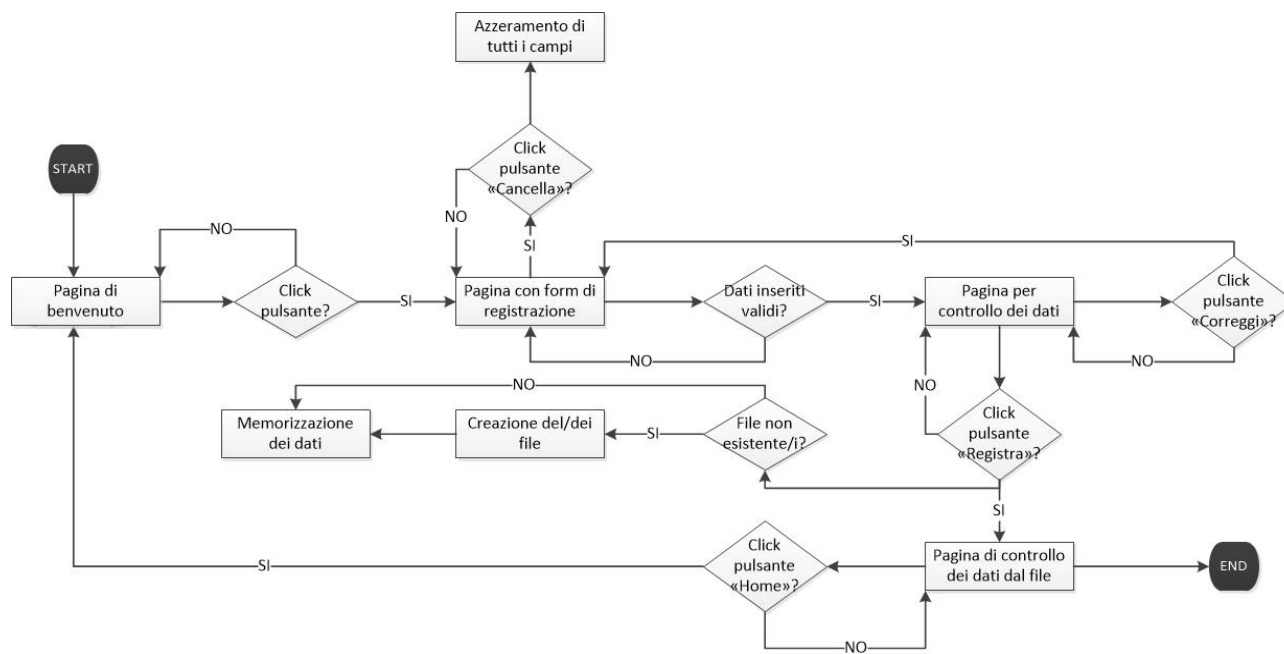
Home

Infine si trova la pagina che controlla il file CSV giornaliero, quindi che ci siano i dati i dati appena inseriti. In fondo c'è il pulsante che permette di ritornare alla pagina di benvenuto e quindi fare eventualmente un'altra registrazione.

2.4 Design procedurale

Questo capitolo descrive cosa succede dopo ogni passaggio effettuato nel sistema delle pagine web. In base a ogni azione dell'utente si scatena un evento preciso finché non è tutto formato nel modo corretto e si può procedere con la memorizzazione dei dati.

Tutto ciò è rappresentato da uno schema:



3 Implementazione

La parte di implementazione del progetto è stata concepita in modo da essere più semplice possibile senza andare a perdersi in dettagli superficiali che avrebbero creato solo problemi inutili.

3.1 Pagina di benvenuto

Questa pagina web introduttiva è semplice e non presenta parti “complicate” per cui sarebbe necessario dilungarsi a spiegare. È formata da un titolo e un pulsante e la pagina presenta un proprio file CSS. Qui di seguito lo screen della pagina:



C'è una piccola animazione che fa apparire una sottolineatura mentre il cursore è sul pulsante “Registrati”:



Questo è il codice CSS del pulsante, che presenta anche l'animazione del bordo inferiore:

```
/* Pulsante */
#register {
    cursor: pointer;
    font-size: 50px;
    background: transparent;
    border: none;
    display: block;
    margin: 0 auto; /* Centatura */
    -webkit-transition: border-bottom .5s;
    transition: border-bottom .5s; /* Animazione */
}
```

3.2 Pagina con form di registrazione

È la pagina principale del sistema, in cui l'utente inserisce i dati che verranno poi memorizzati. Essa ha il proprio file di stile CSS e un file JavaScript utile per la validazione dei campi.

Il layout della pagina si presenta in questo modo:

Si può dire che ci sono principalmente tre parti:

- Il titolo;
- La zona centrale formata da due colonne con campi e input;
- La parte in fondo con il testo che segnala un errore (appare in caso di errore), il testo per segnalare i campi obbligatori e infine il pulsante che azzerà gli input e quello che manda alla pagina successiva.

Parlando di codice sorgente, posso dire di aver lavorato nel modo più compatto possibile senza scrivere inutilità. Il seguente è il codice HTML delle parti principali:

```
<body>

  <h1>Registrazione</h1>

  <!-- Form di registrazione -->
  <form id="reg" method="POST" action="check.php">

    <!-- Contenitore delle colonne -->
    <div class="row">

      <!-- Colonna sinistra -->
      <div class="leftCol">

      </div>

      <!-- Colonna destra -->
      <div class="rightCol">

      </div>

    </div>

    <!-- Testo di errore -->
    <p id="error"></p>

    <!-- Contenitore dei pulsanti -->
    <div class="row">

    </div>

  </form>

</body>
```

Il contenitore principale è il form, che invia i dati con il metodo POST alla pagina di controllo “check.php”. Quindi c’è il contenitore delle colonne con i campi e gli input e si presentano così:

```
<!-- Colonna sinistra -->
<div class="leftCol">

    <!-- Campi -->
    <p>Nome*</p>
    <p>Cognome*</p>
    <p>Data di nascita*</p>
    <p>Indirizzo*</p>
    <p>Numero civico*</p>
    <p>Città*</p>
    <p>NAP*</p>
    <p>Numero di telefono*</p>
    <p>E-mail*</p>
    <p>Genere (sesso)*</p>
    <p>Hobby</p>
    <p>Professione</p>

</div>

<!-- Colonna destra -->
<div class="rightCol">

    <!-- Input nome -->
    <p><input id="nome" type="text" name="nome"></p>

    <!-- Input cognome -->
    <p><input id="cognome" type="text" name="cognome"></p>

    <!-- Input data di nascita -->
    <p><input id="nascita" type="date" name="nascita"></p>

    <!-- Input indirizzo -->
    <p><input id="indirizzo" type="text" name="indirizzo"></p>

    <!-- Input numero civico -->
    <p><input id="civico" type="text" name="civico"></p>

    <!-- Input città -->
    <p><input id="città" type="text" name="città"></p>

    <!-- Input NAP -->
    <p><input id="nap" type="number" name="nap"></p>

    <!-- Input numero di telefono -->
    <p><input id="telefono" type="text" name="telefono"></p>

    <!-- Input e-mail -->
    <p><input id="email" type="text" name="email"></p>

    <!-- Input genere (sesso) -->
    <p>
        <input id="m" type="radio" name="sesso" value="M" checked> M
        <input id="f" type="radio" name="sesso" value="F"> F
    </p>

    <!-- Input hobby -->
    <p><input id="hobby" type="text" name="hobby"></p>

    <!-- Input professione -->
    <p><input id="prof" type="text" name="prof"></p>

</div>
```

Successivamente si trova un semplice tag che segnala un’eventuale errore e infine un altro contenitore simile a quello precedente che contiene i due pulsanti:

```
<!-- Testo di errore -->
<p id="error"></p>

<!-- Contenitore dei pulsanti -->
<div class="row">

    <!-- Campi obbligatori -->
    <p id="obl">*campi obbligatori</p>

    <!-- Pulsante cancella -->
    <div class="leftBtn">
        <p><a onclick="deleteInputs()">Cancella</a></p>
    </div>

    <!-- Pulsante avanti -->
    <div class="rightBtn">
        <p><a onclick="validate()">Avanti</a></p>
    </div>

</div>
```

Per quanto riguarda lo stile CSS, le generalità (font, sfondo, margini, ...) sono simili a quelle della pagina di benvenuto e alle altre. Ma è importante il layout delle due colonne per campi e pulsanti, che sono formate in questo modo:

```
/* Colonna sinistra con i campi */
.leftCol {
    float: Left;
    width: 50%;
    text-align: right;
    padding-right: 5px;
}

/* Colonna destra con gli input */
.rightCol {
    float: Left;
    width: 50%;
    text-align: left;
    padding-left: 5px;
}

/* Pulsante sinistro cancella */
.leftBtn {
    float: Left;
    width: 50%;
    text-align: left;
    padding-left: 20px;
}

/* Pulsante destro avanti */
.rightBtn {
    float: Left;
    width: 50%;
    text-align: right;
    padding-right: 20px;
}

/* Cancello il float per terminare il contenitore */
.row:after {
    content: "";
    display: table;
    clear: both;
}
```

L'ultimo componente di questa pagina è il file JavaScript che presenta la cancellazione dei dati inseriti:

```
/* Azzeramento degli input */
function deleteInputs() {

    /* Prendo gli elementi */
    var inputs = document.getElementsByTagName('input');

    /* Cancello il loro valore */
    for (var i = 0; i < inputs.length; i++) {
        inputs[i].value = null;
    }
}
```

La funzione principale è quella della validazione ed è formata da:

- Una funzione di aiuto per controllare il valore vuoto:

```
/* Controllo che l'input non sia vuoto */
function isEmpty(value) {

    return value === null || value.match(/^ *$/) !== null;

}
```

- Il testo di errore (inizio della funzione principale):

```
/* Validazione */
function validate() {

    /* Testo di errore */
    var error = document.getElementById('error');
    error.innerHTML = "";

}
```

- I valori inseriti:

```
/* Valori inseriti */
var nome = document.getElementById('nome').value;
var cognome = document.getElementById('cognome').value;
var nascita = document.getElementById('nascita').value;
var indirizzo = document.getElementById('indirizzo').value;
var civico = document.getElementById('civico').value;
var citta = document.getElementById('citta').value;
var nap = document.getElementById('nap').value;
var telefono = document.getElementById('telefono').value;
var email = document.getElementById('email').value;
var hobby = document.getElementById('hobby').value;
var prof = document.getElementById('prof').value;
```

- Espressioni regolari:

```
/* RegEx */
var letters = /^[a-zA-Z\u00C0-\u017F\-\ ' ]*$/;
var digits = /^[0-9]*$/;
var alpha = /^[a-zA-Z]*$/;
var telex = /^[0-9\-\ ]*$/;
var mailer = /^[a-zA-Z0-9\-\ ' ]*@[a-zA-Z0-9\-\ ' ]*\.[a-zA-Z0-9\-\ ' ]*$/;
```

- Il controllo dei campi obbligatori:

```
/* Controllo che nessuno degli obbligatori sia vuoto */
if (isEmpty(nome) || isEmpty(cognome) || isEmpty(nascita) || isEmpty(indirizzo) || isEmpty(civico) || isEmpty(citta) || isEmpty(nap) || isEmpty(telefono) || isEmpty(email)) {

    error.innerHTML = "Compilare tutti i campi obbligatori";
    return;

}
```

- Il controllo di ogni campo:

```

/* Validazione nome */
if (!letters.test(nome) || nome.length > 50) {
    error.innerHTML = "Nome non valido";
    return;
}

/* Validazione cognome */
if (!letters.test(cognome) || cognome.length > 50) {
    error.innerHTML = "Cognome non valido";
    return;
}

/* Validazione data di nascita */
var dob = document.getElementById('nascita').value;
var today = new Date().toISOString().slice(0,10);
if (dob > today) {
    error.innerHTML = "Data di nascita non valida";
    return;
}

/* Validazione indirizzo */
if (!letters.test(indirizzo) || indirizzo.length > 50) {
    error.innerHTML = "Indirizzo non valido";
    return;
}

/* Validazione numero civico */
var last = civico.charAt(civico.length - 1);
if (alpha.test(last)) {
    var num = civico.slice(0, -1);
    if (!digits.test(num) || num.length > 3 || num.length < 1) {
        error.innerHTML = "Numero civico non valido";
        return;
    }
} else if (civico.length > 4 || !digits.test(civico)) {
    error.innerHTML = "Numero civico non valido";
    return;
}

/* Validazione città */
if (!letters.test(citta) || citta.length > 50) {
    error.innerHTML = "Città non valida";
    return;
}

/* Validazione NAP */
if (!digits.test(nap) || nap.length < 4 || nap.length > 5) {
    error.innerHTML = "NAP non valido";
    return;
}

/* Validazione Numero di telefono */
telefono = telefono.replace(/\s+/g, '');
if (telefono.charAt(0) == "+") {
    var num = telefono.substr(1);
    if (!telex.test(num) || num.length < 11 || num.length > 50) {
        error.innerHTML = "Numero di telefono non valido";
        return;
    }
} else if (!telex.test(telefono) || telefono.length < 10 || telefono.length > 50) {
    error.innerHTML = "Numero di telefono non valido";
    return;
}

/* Validazione e-mail */
if (!mail.test(email) || email.length > 50) {
    error.innerHTML = "E-mail non valida";
    return;
}

/* Validazione hobby */
if (!isEmpty(hobby) && (!letters.test(hobby) || hobby.length > 50)) {
    error.innerHTML = "Hobby non valido";
    return;
}

/* Validazione professione */
if (!isEmpty(prof) && (!letters.test(prof) || prof.length > 50)) {
    error.innerHTML = "Professione non valida";
    return;
}

```

- E infine l'invio dei dati:

```

/* Eseguo il submit dopo i controlli */
document.getElementById('reg').submit();

```


3.3 Pagina di controllo

È una pagina che ha lo scopo di riscrivere i dati appena inseriti dall'utente in modo da avere una conferma e memorizzare le informazioni oppure tornare indietro per avere la possibilità di correggere qualcosa.

Lo stile è praticamente identico alla pagina con il form, ma al posto degli input ci sono i dati appena inseriti.

Per fare il suo lavoro, questa pagina utilizza delle variabili di sessione:

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    /* Variabile di sessione */
    $_SESSION["dati"] = $_POST;
}

/* Prendo i valori inseriti */
$nome = printLine($_SESSION["dati"]["nome"]);
$cognome = printLine($_SESSION["dati"]["cognome"]);
$nascita = printLine($_SESSION["dati"]["nascita"]);
$indirizzo = printLine($_SESSION["dati"]["indirizzo"]);
$civico = printLine($_SESSION["dati"]["civico"]);
$citta = printLine($_SESSION["dati"]["citta"]);
$nap = printLine($_SESSION["dati"]["nap"]);
$telefono = printLine($_SESSION["dati"]["telefono"]);
$email = printLine($_SESSION["dati"]["email"]);
$sex = printLine($_SESSION["dati"]["sex"]);
$hobby = printLine($_SESSION["dati"]["hobby"]);
$prof = printLine($_SESSION["dati"]["prof"]);
```

Per usare queste variabili, è necessario inizialmente avviare la sessione con un semplice comando:

```
<?php session_start(); ?>
```

Infine c'è una funzione che scrive una riga con il dato inserito prima nel form:

```
/* Funzione per creare una riga nel HTML */
function printLine($value) {
    /* Se il valore è vuoto metto uno spazio vuoto */
    if (empty($value)) {
        $value = "<br>";
    }

    /* Ritorno la riga */
    return "<p>" . $value . "</p>";
}
```

Anche in questo caso si utilizza la strategia delle due colonne:

```
<!-- Colonna destra -->
<div class="rightCol">

    <!-- Dati inseriti precedentemente -->
    <?php echo $nome; ?>
    <?php echo $cognome; ?>
    <?php echo $nascita; ?>
    <?php echo $indirizzo; ?>
    <?php echo $civico; ?>
    <?php echo $citta; ?>
    <?php echo $nap; ?>
    <?php echo $telefono; ?>
    <?php echo $email; ?>
    <?php echo $sex; ?>
    <?php echo $hobby; ?>
    <?php echo $prof; ?>

</div>
```

3.4 Pagina di controllo del file

Questa è la pagina finale ed è formata da una tabella che scrive nuovamente i dati dell'utente, ma questa volta le informazioni sono lette dal file CSV giornaliero mostrando tutto il contenuto del documento.

Il codice PHP è un po' più complesso visto che è necessario lavorare con i file scrivendoli per poi leggerli e scriverli nella pagina web.

Come nella pagina di controllo, anche qui si utilizzano le variabili di sessione:

```
/* Prendo i valori inseriti */
$nome = $_SESSION["dati"]["nome"];
$cognome = $_SESSION["dati"]["cognome"];
$nascita = $_SESSION["dati"]["nascita"];
$indirizzo = $_SESSION["dati"]["indirizzo"];
$civico = $_SESSION["dati"]["civico"];
$citta = $_SESSION["dati"]["citta"];
$nap = $_SESSION["dati"]["nap"];
$telefono = $_SESSION["dati"]["telefono"];
$email = $_SESSION["dati"]["email"];
$sex = $_SESSION["dati"]["sex"];
$hobby = $_SESSION["dati"]["hobby"];
$prof = $_SESSION["dati"]["prof"];
```

Quindi si impostano dei valori costanti:

```
/* Valori generali */
date_default_timezone_set('Europe/Zurich');
define("DATA", date("Y-m-d_H:i:s"));
define("TUTTE", "Registrazioni/Registrazioni_tutte.csv");
define("GIORNO", "Registrazioni/Registrazione_" . date("Y-m-d") . ".csv");
define("HEADER", ["Data", "Nome", "Cognome", "Data_Nascita", "Indirizzo", "Num_Civico", "Citta", "NAP", "Num_Telefono", "Email", "Genere", "Hobby", "Professione"]);
define("SEP", ";");
```

Successivamente inizio a lavorare con i file inserendo i dati:

```
/* Formo i dati */
$dati = [DATA, $nome, $cognome, $nascita, $indirizzo, $civico, $citta, $nap, $telefono, $email, $sex, $hobby, $prof];

/* Formo il file (Registrazioni_tutte) */
$file = fopen(TUTTE, "a");
if (filesize(TUTTE) == 0) {
    fputcsv($file, HEADER, SEP);
}

/* Inserisco i dati */
fputcsv($file, $dati, SEP);
fclose($file);

/* Formo il file (Registrazione_aaaa-mm-gg) */
$file = fopen(GIORNO, "a");
if (filesize(GIORNO) == 0) {
    fputcsv($file, HEADER, SEP);
}

/* Inserisco i dati */
fputcsv($file, $dati, SEP);
fclose($file);
```

L'ultimo passaggio è la lettura del file e trasformare le informazioni in righe di tabella HTML:

```
/* Leggo il file e lo divido in righe */
$lines = file(GIORNO);

/* Risultato da inserire nel HTML */
$table = "";

/* Leggo ogni riga */
for ($i = 0; $i < count($lines); $i++) {

    /* Aggiungo una riga */
    $table .= "<tr>";

    /* Creo una riga e la divido con il separatore */
    $line = explode(SEP, $lines[$i]);

    /* Leggo ogni dato */
    for ($j = 0; $j < count($line); $j++) {
        $table .= "<td>" . $line[$j] . "</td>";
    }

    /* Chiudo la riga */
    $table .= "</tr>";
}
}
```

Infine, nel HTML, c'è la tabella generata e il pulsante per tornare alla pagina di benvenuto:

```
<!-- Tabella contenente i dati del file -->
<table><?php echo $table; ?></table>

<!-- Pulsante home -->
<div class="btn">
    <p><a href="welcome.html">Home</a></p>
</div>
```

Lo stile CSS anche qui è simile alle pagine precedenti per quanto riguarda testo, sfondo e allineamenti. Il seguente è lo stile della tabella:

```
/* Tabella */
table {
    width: 100%;
}

/* Riga della tabella */
td {
    font-size: 10px;
    text-align: center;
    border: 1px solid #fff;
    white-space: nowrap;
    max-width: 0px;
    overflow: hidden;
    text-overflow: ellipsis;
}
```

In questo modo appaiono dei puntini quando il testo è troppo grande per le dimensioni della pagina:

Controllo file												
Data	Nome	Cog...	Data...	Indiriz...	Num...	Citta	NAP	Num...	Email	Genere	Hobby	Profe...
2018-...	Mario	Rossi	1970-...	"Via T...	10	Luga...	6900	0761...	mari...	M		Doce...
												Home

4 Test

4.1 Protocollo di test

Test Case:	TC-01	Nome:	Pulsante di benvenuto
Riferimento:	REQ-01		
Descrizione:	Andare nella schermata del form di registrazione dopo aver premuto il pulsante di benvenuto.		
Procedura:	1. Cliccare il pulsante al centro della pagina.		
Risultati attesi:	Arrivo nella pagina con il form di registrazione dopo aver cliccato il pulsante.		

Test Case:	TC-02	Nome:	Validazione input del form di registrazione
Riferimento:	REQ-02		
Descrizione:	Controllare che nei campi del form vengano inseriti solo valori validi e tutti i campi obbligatori.		
Prerequisiti:	Inserire dei valori negli input.		
Procedura:	1. Compilare i campi con dei dati non idonei. 2. Cliccare il pulsante per proseguire. 3. Non compilare un campo obbligatorio. 4. Cliccare il pulsante per proseguire.		
Risultati attesi:	Indicare l'errore in caso di valori non validi o campi obbligatori non compilati.		

Test Case:	TC-03	Nome:	Pulsanti del form di registrazione
Riferimento:	REQ-02		
Descrizione:	Controllare il corretto funzionamento dei pulsanti "Cancella" e "Avanti".		
Prerequisiti:	Inserire dei valori idonei negli input.		
Procedura:	1. Cliccare il pulsante "Cancella". 2. Reinserire dei dati idonei. 3. Cliccare il pulsante "Avanti".		
Risultati attesi:	Il primo pulsante azzera i campi. In caso di valori validi e campi obbligatori compilati si proseguirà alla pagina di controllo.		

Test Case:	TC-04	Nome:	Pulsanti della pagina di controllo
Riferimento:	REQ-03		
Descrizione:	Controllare il corretto funzionamento dei pulsanti "Correggi" e "Registra".		
Prerequisiti:	Inserire dati idonei nel form e cliccare il pulsante per avanzare.		
Procedura:	1. Cliccare il pulsante "Correggi". 2. Cambiare alcuni dati nel form e avanzare. 3. Cliccare il pulsante "Registra".		
Risultati attesi:	Ritorno nella pagina con il form di registrazione. Dopo la correzione nel form c'è il cambiamento dei dati nella pagina di controllo. Il pulsante "Registra" porta alla pagina di controllo del file.		

Progetto 1

Test Case:	TC-05	Nome:	Memorizzazione dei dati
Riferimento:	REQ-04		
Descrizione:	Controllare la corretta memorizzazione dei dati.		
Procedura:	1. Cliccare il pulsante “Registra” nella pagina di controllo. 2. Attendere il giorno successivo e rifare il procedimento di registrazione.		
Risultati attesi:	Si creano i file “Registrazione_aaaa-mm-gg.csv” e/o il file “Registrazioni_tutte” con il corretto header se non sono già presenti sotto la cartella “Registrazioni”. Nel file completo ci sono tutte le registrazioni senza contare la data. Nel file giornaliero ci sono solo le registrazioni del giorno corrente.		

Test Case:	TC-06	Nome:	Pagina di controllo del file
Riferimento:	REQ-05		
Descrizione:	Controllare che la tabella con i dati letti dal file sia generata correttamente.		
Prerequisiti:	Scrivere il file giornaliero correttamente.		
Procedura:	1. Cliccare il pulsante “Registra” nella pagina di controllo.		
Risultati attesi:	Si genera una tabella che ricorda il contenuto di un file CSV che legge le righe dal file giornaliero e le riscrive in modo più chiaro e esplicito.		

Test Case:	TC-07	Nome:	Pulsante “Home”
Riferimento:	REQ-06		
Descrizione:	Controllare che il pulsante nell'ultima pagina riporti alla pagina di benvenuto.		
Procedura:	1. Cliccare il pulsante “Home” nella pagina di controllo del file.		
Risultati attesi:	Si ritorna nella pagina di benvenuto così da poter fare una nuova registrazione.		

4.2 Risultati test

Test Case:	TC-01
Risultati:	Il pulsante porta alla pagina con il form di registrazione.
Test Case:	TC-02
Risultati:	Non è possibile proseguire in caso di dati non validi o campi obbligatori non compilati. Viene indicato il primo errore riscontrato.
Test Case:	TC-03
Risultati:	I campi vengono azzerati tramite il pulsante “Cancella” e il pulsante “Avanti” porta alla pagina di controllo in caso di dati validi.
Test Case:	TC-04
Risultati:	Il pulsante “Correggi” porta alla pagina con il form senza cancellare i dati. Il pulsante “Registra” porta alla pagina di controllo del file.
Test Case:	TC-05
Risultati:	Il file vengono creati e scritti correttamente con la giusta formattazione e i dati vengono memorizzati nel file in base al giorno corrente.
Test Case:	TC-06
Risultati:	Il file vengono creati e scritti correttamente con la giusta formattazione e i dati vengono memorizzati nel file in base al giorno corrente.
Test Case:	TC-07
Risultati:	Il pulsante “Home” porta alla pagina di benvenuto iniziale.

5 Consuntivo

Il prodotto è stato concluso più o meno come pianificato. È stato necessario lavorare su alcune cose al di fuori delle ore di lezione e alcune attività sono durate di più o di meno di quello pianificato. Quindi nei prossimi lavori potrò organizzarmi meglio grazie a questa esperienza.

6 Conclusioni

La soluzione che ho adottato mi ha permesso di sapere ciò che faccio in ogni singolo punto del progetto. Ovviamente sono già presenti soluzioni di questo tipo e di sicuro migliori della mia, ma per iniziare posso dire di essere soddisfatto del risultato finale. È anche utile per la mia formazione in quanto mi ha portato nel mondo della registrazione dei dati e la programmazione lato server. Inoltre non lo ritengo un lavoro inutile perché i primi progetto di questo tipo vanno fatti in modo e non partire pensando di fare un prodotto che cambierà il mondo e sarà rivoluzionario. Infine posso dire che il lavoro che ho concluso può essere facilmente riutilizzato in futuro come spunto oppure semplicemente migliorarlo. In generale è stata una bella esperienza in cui ho imparato e sono migliorato in vari campi.

6.1 Sviluppi futuri

Un prodotto del genere può generare varie possibilità di sviluppo futuri. Ad esempio si potrebbe creare uno spazio di login in modo da usare in modo concreto i dati registrati. Quindi questo login porta ad uno spazio unico per ogni utente in cui si può sviluppare qualsiasi cosa: un gioco, un social network, ...

6.2 Considerazioni personali

Questo progetto è stato molto utile per la mia crescita in questo lavoro. Ho imparato per la prima volta come si possono prendere dei dati di utenti e vedere con i propri occhi come il prodotto funzioni. Il lavoro in generale non è stato pesante perché il metodo che ho utilizzato è stato il più semplice possibile e capire ogni cosa che faccio così da poter dire che è una cosa completamente mia ed essere “fiero”.

7 Bibliografia

7.1 Sitografia

- <https://www.flaticon.com/authors/simpleicon>, 7 novembre 2018
- <https://stackoverflow.com>, 4 novembre 2018
- <https://www.w3schools.com>, 4 novembre 2018

8 Allegati

Elenco degli allegati:

- I file delle pagine web (HTML, PHP);
- I file contenenti le registrazioni in giorni diversi (CSV);
- Il file con l'implementazione della validazione dei dati inseriti (form.JS);
- I file di stile delle pagine web (CSS);
- Gran parte del lavoro è anche conservato in GitHub: <https://github.com/gabrielessi/Progetto1>.
- I diari delle giornate di lavoro;