

Documentazione

Sito web con pagina di registrazione e verifica dati

Titolo del progetto:	Sito web con pagina di registrazione e verifica dati
Alunno/a:	Filippo Finke
Classe:	Info I3AC
Anno scolastico:	2018/2019
Docente responsabile:	Francesco Mussi, Adriano Barchi, Luca Muggiasca, Elisa Nannini

1	Introduzione	3
1.1	Informazioni sul progetto	3
1.2	Abstract	3
1.3	Scopo	3
	Analisi	3
1.4	Analisi del dominio	3
1.5	Analisi e specifica dei requisiti	3
1.6	Analisi dei costi	6
1.7	Pianificazione	6
1.8	Analisi dei mezzi	7
1.8.1	Software	7
1.8.2	Hardware	7
2	Progettazione	7
2.1	Design dell'architettura del sistema	7
2.2	Design dei dati e database	8
2.3	Design delle interfacce	9
2.4	Design procedurale	12
3	Implementazione	13
3.1	PreventBackHistory	13
3.2	Pagina di benvenuto	13
3.3	Pagina di registrazione	14
3.4	Pagina di controllo	15
3.5	Pagina finale	16
4	Test	17
4.1	Protocollo di test	17
4.2	Risultati test	21
4.3	Mancanze/limitazioni conosciute	21
5	Consuntivo	22
6	Conclusioni	22
6.1	Sviluppi futuri	22
6.2	Considerazioni personali	22
7	Bibliografia	23
7.1	Sitografia	23
8	Allegati	23

1 Introduzione

1.1 Informazioni sul progetto

Titolo del progetto: Sito web con pagina di registrazione e verifica dati

Alunno: Filippo Finke

Classe: I3AC

Docenti responsabili: Francesco Mussi, Adriano Barchi, Luca Muggiasca, Elisa Nannini

Data d'inizio: 07.09.2018

Data di fine: 09.11.2018

1.2 Abstract

We see a lot of websites that request user's data and store it for personalized service. This web application allow users to compile a form with personal information and send these information to the owner of the application that store it and can use them for internal and commercial purposes, like survey, contest,...

1.3 Scopo

Il prodotto deve poter ricavare delle informazioni inserite da un utente attraverso un sito web e poterle salvare all'interno di due file di formato csv. Le informazioni possono poi essere utilizzate dal gestore del sito web.

Analisi

1.4 Analisi del dominio

Questo prodotto è stato ideato con l'intento di avere un sito web per raccogliere in modo molto semplice di dati comuni di utenti. Il contesto di questo prodotto può essere vago, come per esempio invio di newsletter, statistica,... Attualmente il problema viene risolto da siti web molto avanzati (Come per esempio cognitofoms), esistono già alcuni prodotti simili. Gli utenti sono possibili acquirenti di prodotti che non per forza hanno un lavoro. Gli utenti non devono avere delle competenze avanzate per l'utilizzo del prodotto, servono solamente delle nozioni base che permettono all'utente di collegarsi al prodotto in rete. Per operare e eseguire cambiamenti al prodotto si devono avere delle conoscenze mediocri di vari linguaggi di programmazione che vengono utilizzati, fra qui: html, css, php.

1.5 Analisi e specifica dei requisiti

Il committente ha bisogno di avere una pagina web tramite il quale ricavare dati specifici e poterli salvare all'interno di file csv in modo automatico. Il prodotto dovrà guidare l'utente al completamento di questi campi passando da una pagina nel quale sarà possibile compilare i campi richiesti dal committente nel quale verranno eseguite varie validazioni su di essi in seguito una pagina di conferma dei dati ed in fine ad una registrazione che salverà il tutto. L'interazione dell'utente deve avvenire attraverso un browser ovvero un programma che permette la navigazione su internet. Il prodotto deve avere un interfaccia semplice ed intuitiva in modo da non confondere l'utente. Il prodotto deve poter gestire il tutto senza un consumo elevato di risorse, inoltre non sono richieste delle sicurezze oltre alla validazione dei campi all'interno del dominio. Il software sarà ideato in modo da poter funzionare su Chrome, Firefox, Opera e Edge.

ID: REQ-1	
Nome	Gestione file csv
Priorità	1
Versione	1.0
Note	Si necessita di permessi per lettura e scrittura.
Sotto requisiti	
001	I file devono essere salvati all'interno della cartella Registrazioni.
002	Scrittura dei dati in un file con "Registrazione_tutte.csv".
003	Scrittura dei dati per giorno corrente in un file chiamato "Registrazione_aaaa-mm-gg.csv".

ID: REQ-2	
Nome	Classi per la validazione dei dati lato server
Priorità	1
Versione	1.0
Note	Si necessita di un account amministratore per modificare i servizi. / Dipende dal requisito REQ-1.
Sotto requisiti	
001	Dovrà essere possibile controllare i campi.
002	Dovrà essere possibile controllare dati obbligatori.

ID: REQ-3	
Nome	Validazione campi lato client
Priorità	1
Versione	1.0
Note	Dipende dal requisito REQ-5.
Sotto requisiti	
001	Dovrà essere possibile controllare i campi prima dell'invio dei dati.
002	Si dovrà avvisare l'utente per la mancanza di campi obbligatori.
003	Dovrà essere possibile controllare i campi obbligatori.

ID: REQ-4	
Nome	Pagina di benvenuto
Priorità	3
Versione	1.0
Note	Pagina principale del sito con la presenza di un tasto “Registrati” che porta al form di registrazione. Dipende dal requisito REQ-0.

ID: REQ-5	
Nome	Interfaccia di registrazione
Priorità	2
Versione	1.0
Note	Dipende dal requisito REQ-1, REQ-2, REQ-3.
Sotto requisiti	
001	Possibilità di resettare i dati immessi nell'interfaccia con un tasto “Cancella” .
002	Quando i dati vengono inviati si viene reindirizzati ad una pagina nel quale confermare i dati.

ID: REQ-6	
Nome	Pagina di controllo
Priorità	1
Versione	1.0
Note	Pagina che permette all'utente di verificare i dati immessi. Dipende dal requisito REQ-5.
Sotto requisiti	
001	Possibilità di correggere i dati inseriti con un tasto “Correggi”.
002	Possibilità di eseguire la registrazione dell'utente con un tasto “Registrazione”

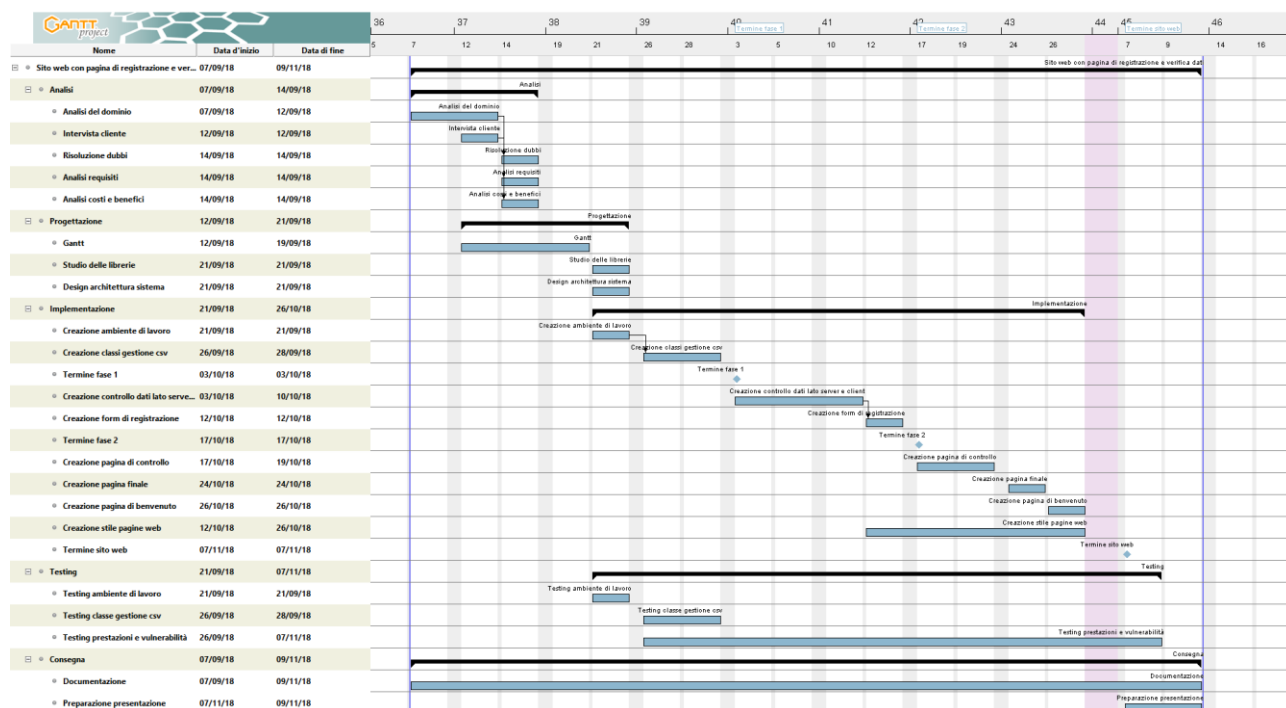
ID: REQ-7	
Nome	Pagina finale con visualizzazione dati
Priorità	1
Versione	1.0
Note	Pagina che visualizza i dati immessi dall'utente. Dipende dal requisito REQ-6.
Sotto requisiti	
001	I dati devono venire disposti come del form di registrazione.
002	I dati devono essere letti dal file csv.

1.6 Analisi dei costi

Categoria	Costo	Ore totali	Costo Totale
Mano d'opera	62 CHF/h	38h	2356CHF

1.7 Pianificazione

La fase che richiede maggior tempo per lo sviluppo del progetto è l'implementazione, nel quale ci sono le parti fondamentali per lo sviluppo di esso. Inoltre come punto fondamentale del progetto è segnata la fine del sito web con software lato server e client funzionanti.



1.8 Analisi dei mezzi

1.8.1 Software

Oracle VM VirtualBox 5.2.4 r119785 (Qt5.6.2)

SublimeText 3.0

Ubuntu Server 18.04.1 LTS

Apache 2.4

PHP 7.0

GanttProject 2.8.9

Microsoft Word 2016

Draw.io

Notify.js

jQuery-3.3.1.js

MaterializeCSS 1.00

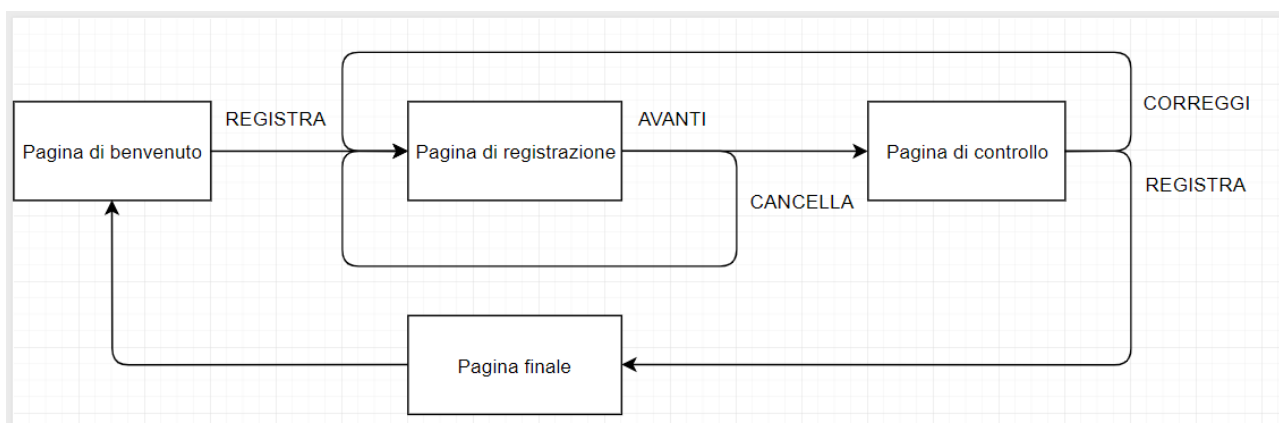
1.8.2 Hardware

Computer portatile (Utilizzato per lo sviluppo del progetto)

- ASUS GL502VSK
- 16GB Ram
- i7-7700HQ
- GTX 1070 GDDR5
- Windows 10 Home

2 Progettazione

2.1 Design dell'architettura del sistema

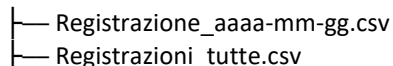


Quando l'utente accede all'applicativo dal WEB si ritroverà alla pagina di benvenuto dal quale può decidere se procedere a registrarsi. In caso volesse registrarsi verrà reindirizzato ad una pagina di registrazione nel quale l'utente può inserire dati specifici, una volta completato il form può decidere se cancellare il form che rimanderà l'utente alla pagina di registrazione vuota, oppure procede. In caso l'utente abbia scelto di procedere verrà mandato ad una pagina di controllo nel quale può decidere di correggere i dati immessi, quindi tornare al form di registrazione oppure di confermare la registrazione. In caso di registrazione l'utente verrà rimandato ad una pagina finale nel quale potrà vedere tutti gli utenti registrati nel suo stesso giorno.

2.2 Design dei dati e database

I dati degli utenti devono essere salvati all'interno dei file con la seguente gerarchia:

Registrazioni



I file verranno salvati utilizzando il seguente formato:

`date;name;lastname;birthdate;gender;street;civicnumber;nap;city;telephone;email;hobby;occupation`

Il campo date viene assegnato in modo automatico dall'applicativo. Corrisponde alla data e ora di registrazione dell'utente.

Campo	Obbligatorio	Limite	Descrizione
Nome	Si	50 caratteri	Può contenere solamente lettere dell'alfabeto, lettere accentate e i caratteri "." o "-" una sola volta all'interno della stringa. Non può essere nullo.
Cognome	Si	50 caratteri	Può contenere solamente lettere dell'alfabeto, lettere accentate e i caratteri "." o "-" una sola volta all'interno della stringa. Non può essere nullo.
Data di nascita	Si		Deve essere una data nel formato: aaaa-mm-gg ES: 2018-10-13 La data non può essere nel futuro. Non può essere nullo.
Via	Si	50 caratteri	Può contenere solamente lettere dell'alfabeto, lettere accentate e i caratteri "." o "-" una sola volta all'interno della stringa. Non può essere nullo.
Numero civico	Si	4 caratteri	Può contenere numeri e opzionalmente una sola lettera dell'alfabeto. ES: 12C Non può essere nullo.
Citta	Si	50 caratteri	Può contenere solamente lettere dell'alfabeto, lettere accentate e i caratteri "." o "-" una sola volta all'interno della stringa. Non può essere nullo.
Nap	Si	5 caratteri	Può contenere solamente numeri. Ha un minimo di 4 caratteri fino ad un massimo di 5.
Telefono	Si	13 caratteri	Può contenere solamente numeri e il carattere "+". Deve avere un minimo di 10 caratteri numerici fino ad un massimo di 13.
Email	Si		Non può contenere caratteri speciali e deve avere al suo interno una "@" e un ".". Deve rispettare il formato email. Non può essere nullo.
Genere	Si	1 carattere	Può contenere solamente "M" o "F". Non può essere nullo.
Hobby	No	500 caratteri	Ha un limite di 500 caratteri e può essere nullo.
Professione	No	500 caratteri	Ha un limite di 500 caratteri e può essere nullo.

2.3 Design delle interfacce

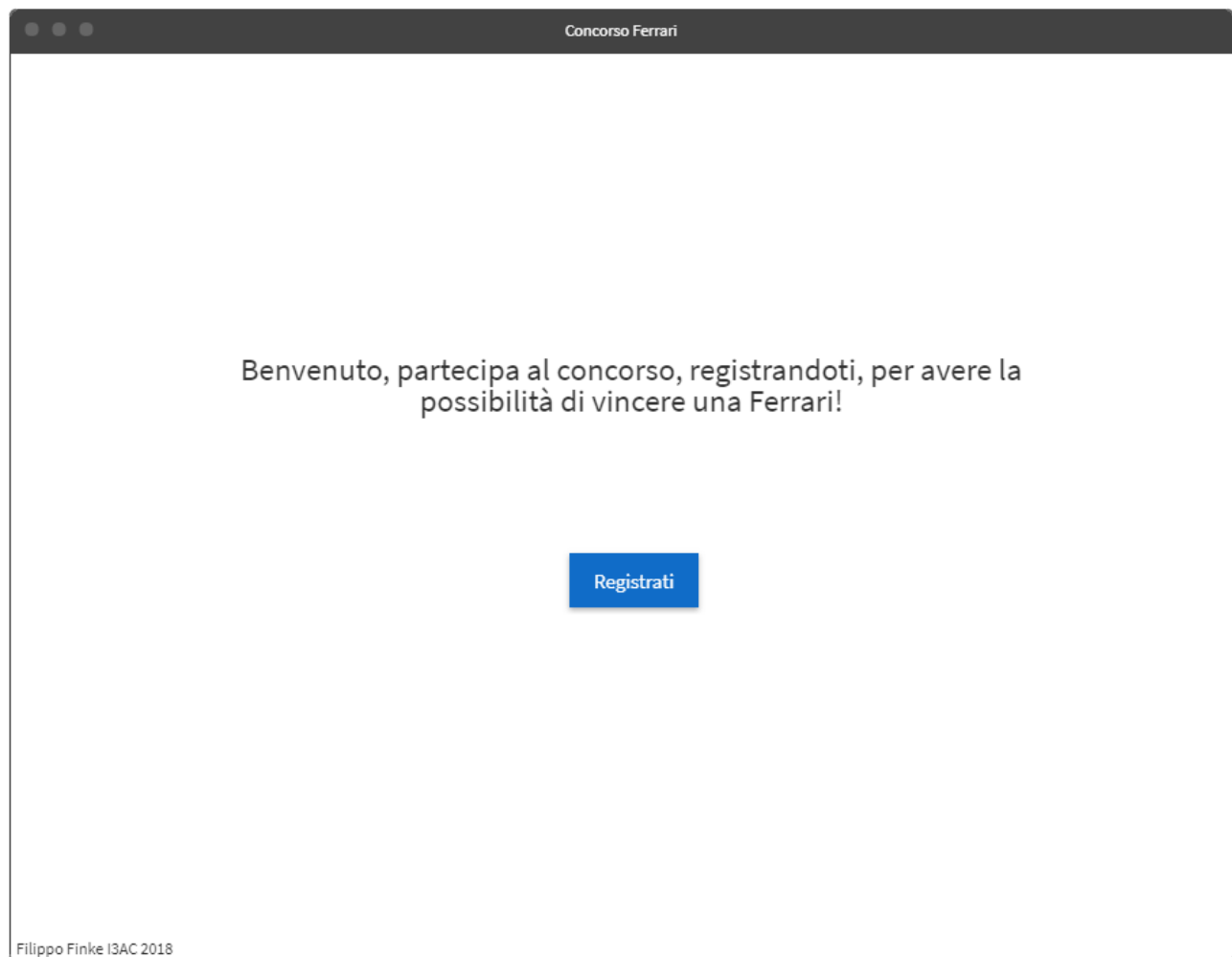


Figura 1 Pagina di benvenuto

Concorso Ferrari

Completa il modulo per partecipare al concorso!

Nome*
Cognome*

Filippo

Finke

Data di nascita*
Genere*

15

12 May 2016

☒ Maschio
☒ Femmina

Via*
Numero civico*

Via Monte
12C

NAP*
Città*

6828
Balerna

Numero di telefono*

Indirizzo email*

Hobby

Professione

I campi con * sono obbligatori.

Cancella

Avanti

Filippo Finke I3AC 2018

Figura 2 Pagina di registrazione

Concorso Ferrari

Verifica i tuoi dati

Nome	Cognome	Data di nascita	Via	Numero civico	Città	NAP	Numero di telef...	Email	Genere	Hobby	Professione
Filippo	Finke	22-12-2001	Via Monte	12C	Balerna	6828	+41091999999	filippo@finke.c...	Maschio	Programmazione	Apprendista

Correggi

Registra

Filippo Finke I3AC 2018

Figura 3 Pagina di controllo

Concorso Ferrari

Registrazioni giornaliere

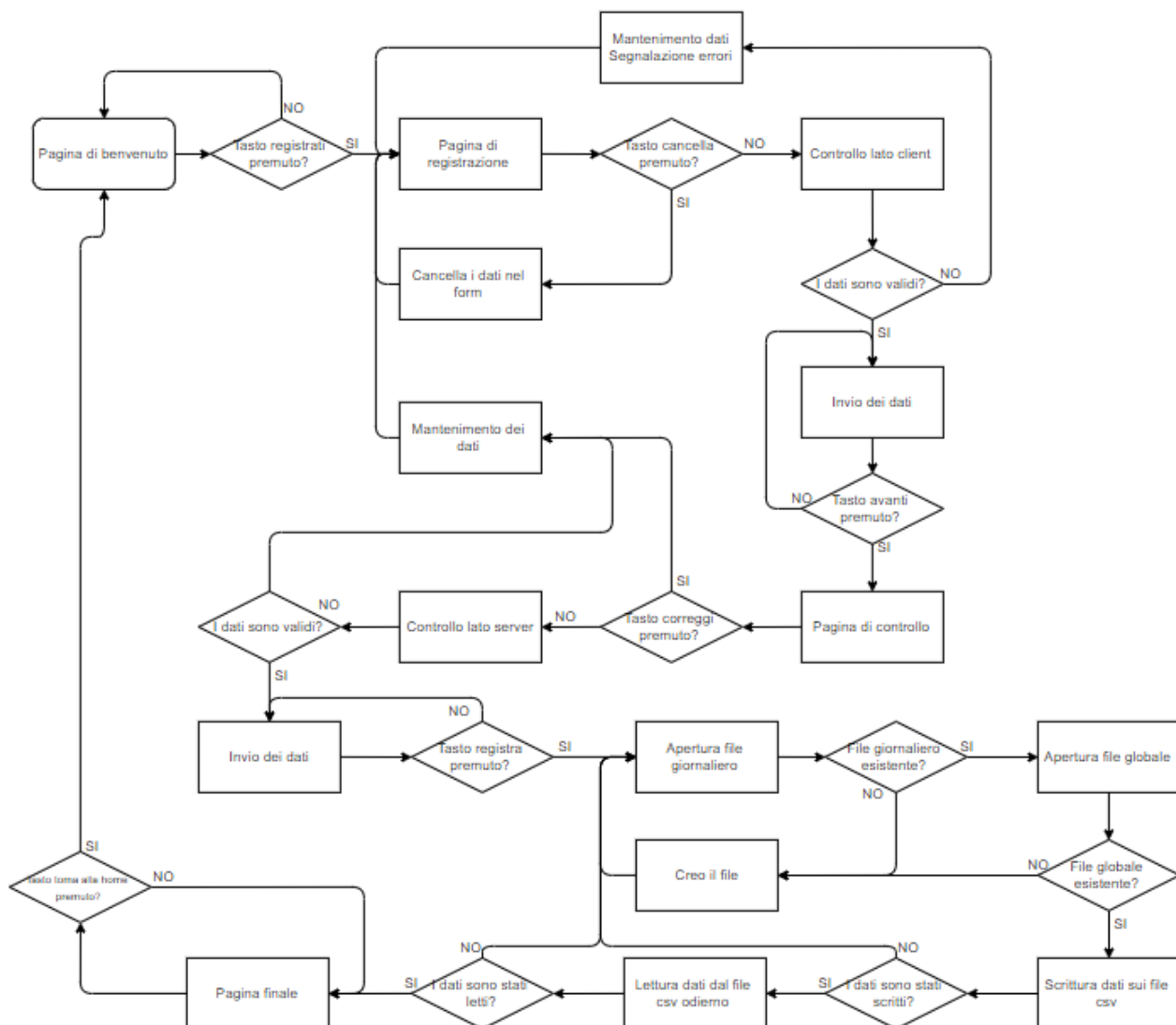
Nome	Cognome	Data di nascita	Via	Numero civico	Città	NAP	Numero di telef...	Email	Genere	Hobby	Professione
Filippo	Finke	22-12-2001	Via Monte	12C	Balerna	6828	+41091999999	filippo@finke.c...	Maschio	Programmazione	Apprendista
Filippo	Finke	22-12-2001	Via Monte	12C	Balerna	6828	+41091999999	filippo@finke.c...	Maschio	Programmazione	Apprendista

Torna alla home

Filippo Finke I3AC 2018

Figura 4 Pagina finale

2.4 Design procedurale



CsvManager
- path: string - delimiter: char
+ __construct(string path, string delimiter): + writeLine(array line):bool + readAll():array

Validator
- delimiter: char - disabledCharsAtStart: string[]
+ containChar(string, char): bool + startsWith(string, char): bool + basic(string): bool + gender(string): bool + telephone(string): bool + nap(string): bool + birthDate(string, string): bool + general(string): bool + civicnumber(string): bool + email(string): bool + textArea(string): bool

3 Implementazione

3.1 PreventBackHistory

All'interno delle pagine registrazione.php, controllo.php e resoconto.php utilizzo un file javascript esterno chiamato preventBackHistory.js che permette di bloccare l'utente di poter tornare alla pagina precedente e rieseguire delle richieste già completate in modo autonomo dal browser. All'interno di questo file si trova il seguente codice:

```
history.pushState(null, null, location.href);
window.onpopstate = function () {
    history.go(1);
};
```

Questo blocco di codice imposta a null la cronologia della pagina in modo da non permettere all'utente di tornare indietro.

3.2 Pagina di benvenuto

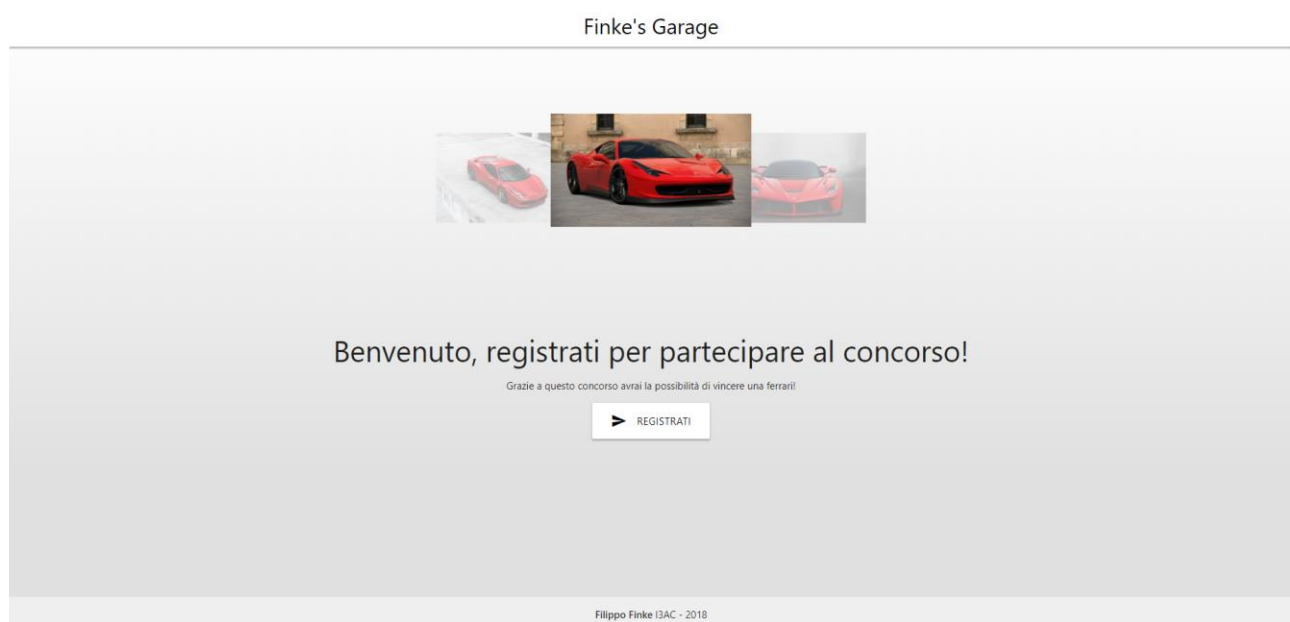


Figura 5 Pagina di benvenuto

Questa è la pagina di benvenuto dell'applicativo web. All'interno di questa pagina vi è uno slider gestito dalla libreria Materialize e viene creato ed applicato utilizzando il seguente blocco di codice:

```
$('.carousel').carousel({
    duration: 250
});
var inst = M.Carousel.getInstance($('.carousel'));
setInterval(function() {
    inst.next();
}, 2000);
```

In questo modo viene creato uno slider di immagini all'oggetto con la classe carousel che scorre ogni 2 secondi in modo automatico.

3.3 Pagina di registrazione

Figura 6 Pagina di registrazione

Questo è il formulario che l'utente deve riempire per procedere nell'utilizzo dell'applicazione.

All'interno di questa pagina utilizzo principalmente due classi, Validator.js e Notify.js.

All'interno della classe Validator ho creato tutti i metodi utilizzati per la validazione di ogni singolo campo, questo è un esempio di un validatore:

```
basic(string) {
    if (string.trim() == "" || string.trim() != string) {
        return false;
    }
    if (this.containsChar(string, this.delimiter)) {
        return false
    } else {
        this.disabledCharsAtStart.forEach(function(char) {
            if (string.startsWith(char)) {
                return false;
            }
        });
        return true;
    }
}
```

Questo metodo permette di controllare che una stringa non sia nulla, che non contenga il delimitatore del file csv e che non contenga dei caratteri specifici come inizio di stringa ("@", "+", "=", "-") in modo da prevenire csvInjection.

Per eseguire il controllo una volta premuto il tasto "Avanti" viene richiamato il metodo submitCheck che esegue i vari metodi di validazione sui campi sugli input.

Esempio di validazione del campo nome:

```
var name = validator.general($('#name').val());
if (!name) {
    $('#name').css("border-color", "red");
    filled = false;
    $.notify("Inserisci un nome valido!", "error");
}
```

In caso il validator del nome ritorni falso il bordo del campo viene colorato di rosso e mostrata una notifica a schermo come si vede nella figura numero 6.

3.4 Pagina di controllo

Finke's Garage

Nome	Filippo
Cognome	Finke
Data di nascita	2001-12-22
Genere	M
Via	Monte Generoso
Numero civico	12C
NAP	6828
Città	Balerna
Telefono	410916788827
Email	filippo.finke@samtrevano.ch
Hobby	Videogames
Professione	Informatico

Filippo Finke |3AC - 2018

Figura 7 Pagina di controllo

Questa è la pagina di controllo, all'interno di questa pagina ritengo fondamentale il seguente blocco di codice:

```
<form id="hiddenForm" action="registrazione.php" method="post">
  <?php
    foreach ($_POST as $field => $value) {
      echo '<input type="hidden" name="' . htmlspecialchars($field) . '"
value="' . htmlspecialchars($value) . '">';
    }
  ?>
</form>
```

Questo blocco di codice permette di generare un form invisibile, in modo dinamico, con i campi che sono stati passati dalla pagina di registrazione. In questo modo posso trasferire i dati dalla pagina di controllo a quella di registrazione in caso di correzione dei dati oppure passarli alla pagina finale nel quale eseguirne la scrittura.. Questa tecnica la utilizzo anche in altre pagine.

3.5 Pagina finale

Finke's Garage

Data di registrazione	Nome	Cognome	Data di nascita	Genere	Via	Numero civico	NAP	Città	Telefono	Email	Hobby	Professione
2018-10-26 15:55:53	Filippo	Finke	2001-12-22	M	Testing	12A	6900	Lugano	410910000000	test@gmail.com		

[🏠 TORNA ALLA PAGINA DI BENVENUTO](#)

Filippo Finke (IAC - 2018)

Figura 8 Pagina finale

Questa è la pagina finale nel quale vengono stampate tutte le registrazioni del giorno corrente. All'interno di questa classe utilizzo principalmente la classe CsvManager per gestire i file csv e Validator che è la traduzione della classe creata in JavaScript tradotta in PHP per la validazione dei dati lato server.

All'interno di questa pagina utilizzo questo codice per prevenire la scrittura in caso di ricaricamento della pagina e di redirect alla home del sito in caso di accesso senza una richiesta di tipo post.

```
$filled = false;
if(isset($_SESSION["filled"])){
    session_destroy();
    $filled = true;
}
else if ($_SERVER["REQUEST_METHOD"] != "POST") {
    header("Location: index.html");
    exit();
}
```

Mentre con questo blocco di codice scrivo i dati, che ricavo grazie alla richiesta POST, all'interno del file giornaliero e file globale ed una volta scritti ricarico la pagina ed assegno un flag nella sessione dell'utente. Ricarico la pagina in modo da eliminare la richiesta POST salvata in automatico dal browser. In questo modo non ho problemi di doppia scrittura dei dati.

```
$globalCsv = new CsvManager(PATH."/".$globalFile, ";");
$dailyCsv = new CsvManager(PATH."/".$dailyFile, ";");
if ($globalCsv -> writeLine($dataArray) && $dailyCsv -> writeLine($dataArray))
{
    $_SESSION["filled"] = true;
    header("Location: resoconto.php");
    exit();
} else {
    $errorMessage = "C'è stato un errore nel salvataggio dei dati,
riprova!<br>Redirezionamento in 3 secondi.";
}
```

Inoltre i dati vengono comunque validati lato server per prevenire modifiche lato client.

Per eseguire la validazione di un dato utilizzo la classe Validator.

Esempio di validazione del campo nome:


```
if (!$v->general($name))
```

```
    $errors[] = array("name", "Inserisci un nome valido!");
```

In questo caso chiamo il metodo di validazione generale utilizzato per Nome, Cognome, Via e Città attraverso l'oggetto \$v ed in caso il risultato sia falso aggiungo un messaggio di errore da stampare all'utente.

Anche all'interno di questa pagina utilizzo la tecnica di creare un form invisibile per rimandare i dati dell'utente alla pagina di registrazione in caso di errore di validazione. Con il blocco seguente di codice controllo se sono stati generati degli errori e creo un form invisibile contenente le informazioni riguardanti tutti i campi inseriti dall'utente e aggiungo i messaggi di errore che verranno interpretati dalla pagina di registrazione sotto forma di notifica. Inoltre stampo un messaggio di errore nella pagina e attraverso JavaScript eseguo il submit del form dopo 3 secondi.

```
<?php
if(isset($errors) && count($errors) > 0 || !$filled) {
?>
<form id="hiddenForm" action="registrazione.php" method="post">
    <?php
        foreach ($_POST as $a => $b) {
            echo '<input type="hidden" name="' . htmlspecialchars($a) . '" value="'
. htmlspecialchars($b) . '">';
        }
        foreach ($errors as $error) {
            echo '<input type="hidden" name="errors[]" value="' .
htmlspecialchars($error[1]) . '">';
        }
    ?>
</form>
<script>
    $('#container').html("<h5 class='red-text'><?php echo $errorMessage;
?></h5>");
    setTimeout(function(){document.getElementById('hiddenForm').submit()},3000);
</script>
<?php
}
?>
```

4 Test

4.1 Protocollo di test

Test Case:	TC-001	Nome:	Test campi semplici
Riferimento:	REQ-02/03		
Descrizione:	Validazione dei campi semplici (Nome, Cognome, Via e Città) di tipo stringa. La stringa deve avere una lunghezza massima di cinquanta caratteri. Non può essere nulla e può contenere caratteri alfabetici, spazi, un . o un – seguiti e preceduto da dei caratteri.		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Inserire tutti i campi con almeno un carattere. 2. All'interno del campo Nome, Cognome, Via e Città inserire dei dati non validi. Es: 1239123, Filippo1231, Filippo!@1 3. Premere il tasto "Avanti" 		
Risultati attesi:	La pagina di registrazione deve ritornare un messaggio di errore che segnala i campi inseriti.		

Test Case:	TC-002	Nome:	Test campi di tipo text area
Riferimento:	REQ-02/03		
Descrizione:	Validazione dei campi di tipo text area (Hobby e Professione) di tipo stringa. La stringa deve avere una lunghezza massima di 500 caratteri. Può essere nulla e può contenere qualsiasi carattere al di fuori di testo tra i caratteri "<" e ">".		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Inserire tutti i campi con almeno un carattere 2. All'interno del campo Hobby e Professione inserire dei dati non validi. Es: <TestString>, <script src=">, </777s> 3. Premere il tasto "Avanti" 		
Risultati attesi:	La pagina di registrazione deve ritornare un messaggio di errore che segnala i campi inseriti.		

Test Case:	TC-003	Nome:	Test campo email
Riferimento:	REQ-02/03		
Descrizione:	Validazione del campo email di tipo stringa. La stringa non può essere nulla, non deve contenere caratteri speciali e deve contenere almeno una @ e un .		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Inserire tutti i campi con almeno un carattere 2. All'interno del campo Email inserire dei dati non validi. Es: 123123w@ @test.com, 123123, <as@gmail.com 3. Premere il tasto "Avanti" 		
Risultati attesi:	La pagina di registrazione deve ritornare un messaggio di errore che segnala il campo inserito.		

Test Case:	TC-004	Nome:	Test campo telefono
Riferimento:	REQ-02/03		
Descrizione:	Validazione del campo telefono di tipo stringa. La stringa non può essere nulla, non deve contenere caratteri speciali. Il formato del numero di telefono può essere con o senza prefisso e può contenere solamente numeri, "+" e spazi. Ci devono essere un minimo di 10 fino ad un massimo di 13 cifre.		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Inserire tutti i campi con almeno un carattere 2. All'interno del campo telefono inserire dei dati non validi. Es: Numero123123, +411234567886868688, 123uiejwjdj923 3. Premere il tasto "Avanti" 		
Risultati attesi:	La pagina di registrazione deve ritornare un messaggio di errore che segnala il campo inserito.		

Test Case:	TC-005	Nome:	Test campo NAP
Riferimento:	REQ-02/03		
Descrizione:	Validazione del campo NAP di tipo numerico. Ha un limite di 5 caratteri e può contenere solamente numeri. Minimo 4 caratteri.		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Inserire tutti i campi con almeno un carattere 2. All'interno del campo NAP inserire dei dati non validi. Es: 123456, 1234, campo vuoto 3. Premere il tasto "Avanti" 		
Risultati attesi:	La pagina di registrazione deve ritornare un messaggio di errore che segnala il campo inserito.		

Test Case:	TC-006	Nome:	Test campo numero civico
Riferimento:	REQ-02/03		
Descrizione:	Validazione del campo numero civico di tipo stringa. Non può essere nullo ha un massimo di 4 caratteri. Può contenere numeri ed un solo carattere dell'alfabeto. Il carattere alfabetico può essere scritto solamente dopo i caratteri numerici ed è opzionale.		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Inserire tutti i campi con almeno un carattere 2. All'interno del campo Numero civico inserire dei dati non validi. Es: A123, 123AA, *123 3. Premere il tasto "Avanti" 		
Risultati attesi:	La pagina di registrazione deve ritornare un messaggio di errore che segnala il campo inserito.		

Test Case:	TC-007	Nome:	Test campo data di nascita
Riferimento:	REQ-02/03		
Descrizione:	Validazione del campo data di nascita di tipo data. Non può essere nullo. La data ha il formato "aaaa-mm-gg". La data non può essere in futuro.		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Inserire tutti i campi con almeno un carattere 2. All'interno del campo data premere con il mouse e selezionare una data nel passato. 3. Premere il tasto "Avanti" 		
Risultati attesi:	La pagina di registrazione deve mostrare un popup dal quale è possibile selezionare una data e sottolineare il campo di verde.		

Test Case:	TC-008	Nome:	Test campo genere
Riferimento:	REQ-02/03		
Descrizione:	Validazione del campo genere di tipo stringa. Non può essere nullo e può essere solamente "F" oppure "M"		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Inserire tutti i campi con almeno un carattere 2. Lasciare vuoto il campo che richiede il genere. 3. Premere il tasto "Avanti" 		
Risultati attesi:	La pagina di registrazione deve mostrare un errore che segnala il campo da completare.		

Test Case:	TC-009	Nome:	Test classe di gestione file csv
Riferimento:	REQ-01		
Descrizione:	Controllare che la classe di gestione del file csv crei i file se essi sono inesistenti.		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Eliminare la cartella Registrazioni 2. Recarsi nella pagina di registrazione e compilare il formulario. 3. Premere il tasto "Avanti", premere il tasto "Registra" 		
Risultati attesi:	I file csv mancanti devono essere stati creati in modo automatico.		

Test Case:	TC-010	Nome:	Test pagina di registrazione
Riferimento:	REQ-05		
Descrizione:	Controllare che il comando di eseguire un reset dei dati del form sia correttamente funzionante. Il reset deve cancellare tutti i dati negli input, i contatori ed eventuali bordi colorati.		
Prerequisiti:	La pagina registrazione.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Recarsi nella pagina di registrazione e compilare il formulario. 2. Premere il tasto "Cancella", premere il tasto "Ok" nel popup. 		
Risultati attesi:	Il formulario deve venire resettato completamente. I valori devono essere vuoti, i contatori dei caratteri impostati a zero e il colore dei bordi a grigio.		

Test Case:	TC-011	Nome:	Test pagina di controllo
Riferimento:	REQ-06		
Descrizione:	Controllare che il comando di correzione riporti alla pagina di registrazione con i campi già compilati in base al formulario precedenti.		
Prerequisiti:	La pagina registrazione.php e controllo.php deve essere attiva.		
Procedura:	<ol style="list-style-type: none"> 1. Recarsi nella pagina di registrazione e compilare il formulario. 2. Premere il tasto "Avanti" 3. Premere il tasto "Correggi" 		
Risultati attesi:	La pagina deve reindirizzare l'utente alla pagina di registrazione.php con i campi già compilati in base al formulario precedente.		

4.2 Risultati test

Codice test	Risultato	Note
TC-001	Passato	Nessuna
TC-002	Passato	Nessuna
TC-003	Passato	Nessuna
TC-004	Passato	Nessuna
TC-005	Passato	Nessuna
TC-006	Passato	Nessuna
TC-007	Passato	Nessuna
TC-008	Passato	Nessuna
TC-009	Passato	Nessuna
TC-010	Passato	Nessuna
TC-011	Passato	Nessuna

4.3 Mancanze/limitazioni conosciute

L'applicativo non può essere visualizzato in modo corretto dal browser "Internet Explorer".
"Internet Explorer" non supporta l'utilizzo di classi da parte di JavaScript quindi l'applicativo non può essere caricato.

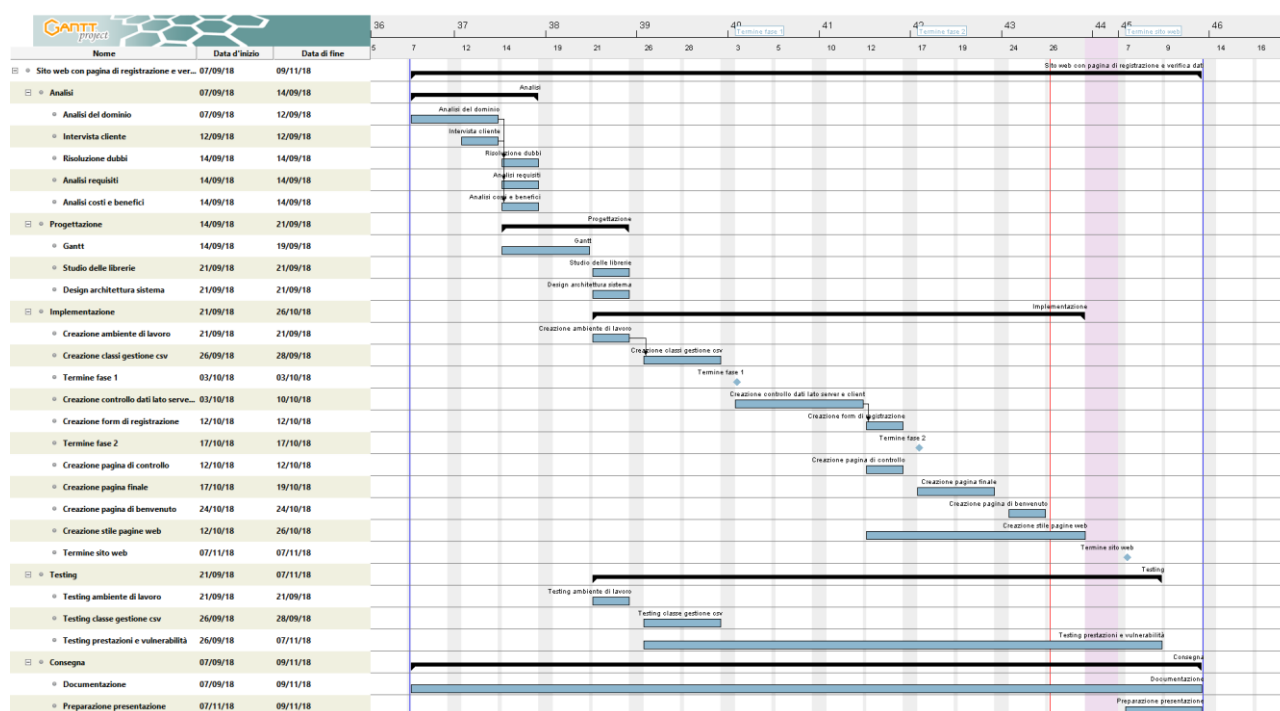
```

i HTML1300: È stata eseguita la navigazione.
  registrazione.php
x SCRIPT1002: Errore di sintassi
  validator.js (6,17)
x SCRIPT5009: 'Validator' non è definito
  registrazione.js (1,1)

```

Inoltre non ritengo essenziale lo sviluppo di una versione personalizzata per questo browser poiché vecchio ed obsoleto.

5 Consuntivo



Nel gantt consuntivo si possono notare diversi cambiamenti:

- L'attività di sviluppo del gantt si è accorciata di una settimana.
- Lo sviluppo della pagina di controllo e di registrazione sono state unite nello stesso giorno.
- Lo sviluppo della pagina finale è stato prolungato di una settimana.
- Lo sviluppo della pagina di benvenuto è stato anticipato di una settimana.

6 Conclusioni

L'applicativo è stato sviluppato con lo scopo di apprendere la gestione di un progetto. Esistono molti tipi di applicativi che permettono personalizzazioni, modifiche e molto altro rispetto a ciò che ho sviluppato. Quindi ritengo che il mio progetto non sia stato sviluppato con lo scopo di essere utilizzato al di fuori del campo scolastico. L'applicativo è stato sviluppato interamente come richiesto dalla specifica.

6.1 Sviluppi futuri

Su questo prodotto possono essere sviluppati molti componenti aggiuntivi, come per esempio:

- Conferma d'iscrizione al modulo tramite posta elettronica.
- Termini e Condizioni
- Verifica telefonica
- Verifica del NAP tramite il campo Città.
- Maggiore sicurezza nel salvataggio dei dati.

...

Alcune migliorie si potrebbero fare all'interno della gestione delle pagine web, in altre parole utilizzare delle sessioni. Gestire in maniera migliore i file csv in modo da supportare dati multi linea, ...

6.2 Considerazioni personali

Grazie a questo progetto ho imparato a sviluppare del codice orientato agli oggetti utilizzando il linguaggio JavaScript.

7 Bibliografia

7.1 Sitografia

- <https://www.ganttproject.biz/>, *GanttProject: free desktop project management app*, 14-09-2018.
- <https://www.draw.io/>, *Flowchart Maker Online Diagram Software*, 21-09-2018
- <https://www.ubuntu.com/download/server>, *Download Ubuntu Server | Download | Ubuntu*, 21-09-2018
- <http://php.net/manual/en/language.oop5.php>, *PHP: Classes and Objects - Manual*, 26-09-2018
- <https://www.phpliveregex.com/>, *PHP Live Regex*, 03-10-2018
- <https://payatu.com/csv-injection-basic-to-exploit/>, *CSV injection: Basic to Exploit!!!!* - payatu, 03-10-2018
- <https://materializecss.com/>, *Documentation - Materialize*, 05-10-2018
- <https://developer.mozilla.org/it/docs/Web/JavaScript/Reference/Classes>, *Classes - JavaScript | MDN*, 05-10-2018
- <https://materializecss.com/pickers.html>, *Pickers - Materialize*, 10-10-2018
- <https://stackoverflow.com/>, *Stack Overflow*, 10-10-2018
- <https://notifyjs.jpillora.com/>, *Notify.js*, 12-10-2018
- https://rextester.com/l/php_online_compiler, *compile php online*, 17-10-2018
- https://it.wikipedia.org/wiki/Cross-site_scripting, *Cross-site scripting - Wikipedia*, 19-10-2018

8 Allegati

- Diari di lavoro
- Quaderno dei compiti
- Prodotto
- Codice sorgente su GitHub <https://github.com/filippofinke/Progetto-1>