



---

# Sleek Paradise

Progetto di Tecnologie Web A.A. 2019/20

---

## Membri del gruppo

Alessio Bettarello - 1227736  
Mattia Gottardello - 1142520  
Olivier Utshudi - 1143556  
Nicoló Sartor - 1143715

**e-mail referente del gruppo:** [alessio.bettarello@studenti.unipd.it](mailto:alessio.bettarello@studenti.unipd.it)

**Indirizzo web homepage del sito:**

<http://tecweb.studenti.math.unipd.it/abettare>

**Link privato homepage amministrazione:**

<http://tecweb.studenti.math.unipd.it/abettare/admin.php>

## Account

Classe	Username	Password
Amministratore	admin	admin

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Abstract . . . . .	2
<b>2</b>	<b>Analisi</b>	<b>2</b>
2.1	Utente non autenticato . . . . .	2
2.2	Utente autenticato . . . . .	2
2.3	Conclusione . . . . .	3
<b>3</b>	<b>Sviluppo</b>	<b>4</b>
3.1	Progettazione . . . . .	4
3.2	Design . . . . .	4
3.3	Layout . . . . .	4
3.3.1	Desktop . . . . .	4
3.3.2	Mobile . . . . .	5
3.4	Linguaggi e tecnologie . . . . .	6
3.4.1	HTML e CSS . . . . .	6
3.4.2	Comportamento . . . . .	6
3.4.3	Dati . . . . .	7
3.4.4	Contenuto dinamico . . . . .	9
3.4.5	Accessibilità . . . . .	10
3.4.6	Accesso alle pagine . . . . .	10
3.4.7	Versioni software . . . . .	11
3.5	Criticità . . . . .	11
3.5.1	XHTML e CSS . . . . .	11
3.5.2	Dati . . . . .	11
3.5.3	JavaScript . . . . .	12
3.5.4	PhP . . . . .	12
<b>4</b>	<b>Testing</b>	<b>13</b>
<b>5</b>	<b>Installazione</b>	<b>13</b>
<b>6</b>	<b>Organizzazione del lavoro</b>	<b>14</b>

# 1 Introduzione

## 1.1 Abstract

Il sito web **Sleek Paradise** nasce per necessità in seguito a un periodo di pandemia globale. L'omonima azienda ha potuto così aumentare la propria visibilità, oltre ad offrire un servizio veloce e diretto. Nel sito è possibile trovare informazioni generali sull'azienda, i contatti principali, le news relative ai defunti, ma soprattutto richiedere un preventivo personalizzato.

## 2 Analisi

Questo sito fornisce informazioni chiare e comprensibili. L'utente deve poter trovare ciò che cerca in modo semplice e diretto, utilizzando sia la versione mobile che quella desktop. Il sito verrà realizzato per un bacino di utenza ampio, dall'anziano con poche competenze del computer, al giovane che si interfaccia tramite smartphone.

### 2.1 Utente non autenticato

L'utente non autenticato è, tipicamente, il cliente che necessita di informazioni riguardo l'impresa funebre. Ha la necessità di ottenere rapidamente quello che cerca navigando tra le pagine come orari, modalità di contatto e storico dell'impresa. Inoltre vuole poter richiedere un preventivo o comunque poter visualizzare i prodotti che l'impresa offre. Per fare ciò, è plausibile che non debba effettuare nessun tipo di autenticazione ma semplicemente compilare un form, attività più intuitiva e semplice.

### 2.2 Utente autenticato

Un utente autenticato, detto *amministratore* potrà:

- visualizzare la lista dei preventivi richiesti dagli utenti non autenticati e modificarne l'importo applicando eventuali sconti o maggiorazioni;
- gestire i prodotti offerti dall'impresa, aggiungerne di nuovi o rimuovere quelli non più disponibili;
- aggiornare la sezione *News* tramite accettazione del preventivo, in seguito al consenso da parte del cliente.

## 2.3 Conclusione

Il sito avrà quindi una sezione dove l'utente potrà richiedere un nuovo preventivo e una sottosezione dove potrà visualizzare nome e data del defunto. Dovrà avere inoltre la possibilità di gestire tali preventivi e aggiornare il proprio catalogo in base alla disponibilità accedendo alla partizione *gestionale* del sito tramite link non pubblico che verrà fornito al titolare dell'impresa. L'elenco delle pagine stilato a conclusione dell'analisi, organizzato gerarchicamente è il seguente:

- Homepage
- Azienda
- Contatti
- Preventivatore
  - Vista preventivo
- Log-in
  - Vista Preventivi
    - \* Vista/modifica singolo preventivo
  - Gestione credenziali utente
  - Gestione preventivatore
    - \* Rimozione elementi
    - \* Aggiunta bare
    - \* Aggiunta urne
    - \* Aggiunta auto
    - \* Aggiunta composizioni
    - \* Aggiunta cerimonie

## 3 Sviluppo

### 3.1 Progettazione

In una fase iniziale si è deciso la tipologia del sito web, il layout e le varie funzionalità da implementare. Successivamente sono state definite le tecnologie da applicare e, con l'ausilio di GitHub e GitKraken, è stato gestito il versionamento e sono stati assegnati i vari compiti per i singoli membri. A seguire è stato progettato il database con la finalità di gestire i preventivi richiesti dall'utente. In una fase successiva, si è passati allo sviluppo del front-end e del back-end e, una volta effettuato il merge del codice, è stata eseguita la fase di test, per verificare la presenza di eventuali bug, per verificare l'accessibilità e la compatibilità con diversi dispositivi e browser.

### 3.2 Design

Il gruppo ha posto la massima attenzione nella **separazione tra struttura, presentazione e comportamento**. Nello specifico è stato utilizzato XHTML Strict 1.0 per la struttura, così da garantire una maggiore retrocompatibilità con i vecchi browsers. Per la presentazione sono stati impiegati dei fogli di stile CSS linkati, mentre per il comportamento sono stati realizzati degli script Javascript. La generazione delle pagine web a contenuto dinamico avviene invece tramite il pre-processamento di fogli di codice PHP che, sfruttando dei template nel formato XHTML sopra indicato, forniscono tale contenuto. Il supporto mobile è stato reso disponibile tramite opportuni fogli di stile, così come i vari layout di stampa.

### 3.3 Layout

#### 3.3.1 Desktop

Si è deciso di strutturare il layout a colonna doppia con larghezza fissa. Nella colonna di sinistra è presente il menù con i link di navigazione, mentre nella colonna di destra è presente il contenuto della pagina web. Ciò permette una buona mantenibilità, in quanto eventuali modifiche non incidono sul layout del sito. È inoltre presente il *breadcrumb* che aiuta l'utente a non perdersi durante la navigazione.

L'*header* contiene il nome dell'azienda e varia in larghezza a seconda della risoluzione.

Il *footer* contiene la validazione delle pagine XHTML Strict 1.0 e del codice CSS.

### **3.3.2 Mobile**

La differenza sostanziale con la versione Desktop, riguarda il menù di navigazione. Per una migliore interazione si è passati ad una struttura a colonna singola, con i link posti sotto l'header. è stato cambiato il colore dei pulsanti clickabili, al fine di renderli maggiormente visibili, così come il titolo della pagina web in cui ci si trova.

## 3.4 Linguaggi e tecnologie

Per lo sviluppo del sito, sono state impiegate e integrate diverse tecnologie di seguito illustrate.

### 3.4.1 HTML e CSS

Il codice XHTML Strict 1.0 è stato templatizzato, ove possibile, per agevolare il riutilizzo in maniera efficace o per eventuali modifiche future. Sono state seguite alcune linee guida per ottenere codice pulito:

- mantenere la separazione tra contenuto e presentazione, utilizzando file linkati separati;
- i tag di intestazione vengono usati in base all'importanza e non in base all'estetica;
- le sezioni *header*, *footer* e *breadcrumb* vengono riutilizzate per ogni pagina del sito web;
- le grandezze sono state espresse in unità relative (%), ad eccezione dei bordi dei vari *div*.

Per quanto riguarda il codice CSS, è stato utilizzato un foglio di stile di base per gestire le dimensioni della pagina e i vari font. Ad ogni pagina è stato poi linkato un'ulteriore foglio di stile in CSS, con un'identificativo univoco, al fine di personalizzare al meglio la stessa.

**CSS per il mobile** Per gestire il sito nella sua versione mobile, è stato implementato un foglio di stile CSS, chiamato *mobile\_base*, il quale è stato importato in tutti i fogli di stile di ogni singola pagina web. Successivamente sono state effettuate alcune modifiche nei vari fogli di stile, per avere un miglior layout della pagina.

**CSS di stampa** Le pagine di stampa sono state gestite attraverso un foglio di stile generico, in cui sono state rimosse le immagini, il menù e il footer. Poi ad ogni singola pagina web, è stato associato un foglio CSS per la personalizzazione.

### 3.4.2 Comportamento

Il comportamento delle pagine web è stato gestito tramite il linguaggio JavaScript utilizzato nel progetto principalmente per effettuare controlli nelle form presenti nelle varie pagine. In ogni pagina contenente form, se il browser lo supporta, viene ritardato il comportamento atteso della form intercettando l'operazione di *submit*. Essa verrà effettuata in maniera automatica solo nel momento in cui sia stata verificata la qualità dei dati immessi. I principali controlli sono:

- Avvenuta compilazione dei campi della form
- Qualità della compilazione dei campi ovvero se il loro contenuto sia del tipo atteso per quel campo (e.g. numerico, email, data)
- Coerenza nella compilazione ovvero se nella form siano stati inseriti dati contrastanti fra di loro

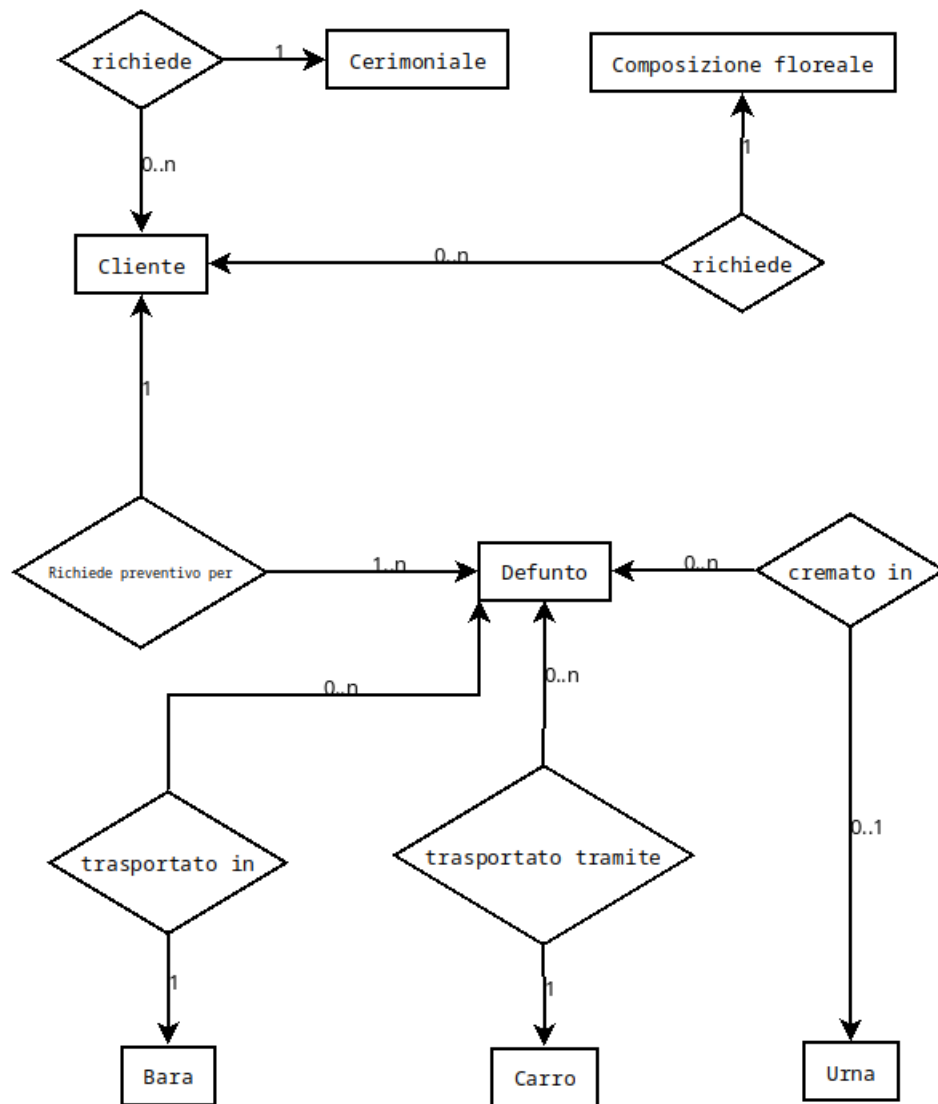
Qualora non si verifichi una delle precedenti condizioni, viene creato e visualizzato un elemento informativo contenente un messaggio di errore e delle informazioni che possono aiutare a correggerlo. Il messaggio di errore inoltre, funge da collegamento al campo da correggere di modo da permettere all'utente di accedere rapidamente al contenuto da modificare.

### 3.4.3 Dati

Il sito possiede del contenuto dinamico che va immagazzinato in una struttura dati. Per questo abbiamo impiegato un *database MariaDb*. Tramite esso vengono rappresentate tutte le informazioni utili alla corretta esecuzione delle operazioni effettuabili sul sito.



**Struttura** Il team ha puntato sulla semplicità per il salvataggio dei dati rappresentandoli con il seguente schema *Entity-Relationship*:



Vista l'entità informativa e non precisa e puntuale della sezione preventivatore e per questioni di budget di tempo a disposizione, il team ha preferito evitare di configurare un accesso cliente per effettuare multiple operazioni in favore di una singola richiesta di preventivo. Per questo sono state accorpate le entità cliente e defunto per poi andare a produrre lo schema logico-relazionale definitivo qui rappresentato:

- **USERS**( id:INT, username:VARCHAR(30)unique, enc\_password:VARCHAR(64))
- **CERIMONIE**( id: INT, tipologia:TEXT, costoBase:DECIMAL(13,2))
- **URNE**( id:INT, versione:VARCHAR(50), materiale:VARCHAR(50), costoBase:DECIMAL(13,2))

- **AUTO**( id:INT, marca:VARCHAR(40), modello:VARCHAR(40), costoBase:DECIMAL(13,2), cilindrata:INT)
- **BARE**( id:INT, versione:VARCHAR(50), materiale:VARCHAR(50), costoBase:DECIMAL(13,2))
- **COMPOSIZIONI**( id:INT, nome:VARCHAR(50), costoBase:DECIMAL(13,2))
- **DEFUNTI** ( cf:VARCHAR(16), nomeDefunto:TEXT, cognomeDefunto:TEXT, dataNascita: DATE, dataDecesso:DATE, residenza:VARCHAR(100), nomeCliente:TEXT, cognomeCliente:TEXT, numeroTelefono:VARCHAR(10), data:TIMESTAMP, isPublic:TINYINT(1), proposta:DECIMAL(13,2), idCerimonia:INT, idBara:INT, idUrna:INT, idAuto:INT, idFiori:INT)

Come si può notare è presente una relazione per la gestione degli amministratori e delle credenziali d'accesso.

#### 3.4.4 Contenuto dinamico

Per preprocessare le pagine contenenti form o contenuto dinamico, abbiamo impiegato il linguaggio PHP. Viene utilizzato nella pagina principalmente per effettuare il controllo dei dati inseriti nelle form lato back-end o per la generazione delle pagine a contenuto dinamico come la homepage contenente i necrologi dei clienti. Il processo di generazione viene effettuato caricando i template di pagine leggendoli dal filesystem. In essi vengono sostituiti dei placeholder specifici con il contenuto dinamico.

**Forms** Ogni dato inoltrato da una form viene verificato lato back-end all'invio. Questo è utile soprattutto nel caso il browser da cui si naviga, non supporti JavaScript oppure ci siano stati dei problemi con l'esecuzione degli script nel corso dell'invio oppure i dati siano stati corrotti o peggio ancora modificati per tentare un attacco potenzialmente dannoso. Nei casi indicati in sezione "Comportamento", viene riprodotto un messaggio d'errore identico al precedente. Inoltre, nel caso del preventivatore presente nel sito, tramite PHP è possibile rendere disponibile sempre i prodotti aggiornati a catalogo non appena vengano inseriti dagli amministratori non necessitando di modificare manualmente le form. PHP si occupa inoltre del salvataggio a database dei dati immessi una volta validati, sfruttando il driver MySQLi.

**Amministrazione** Tramite il preprocessing, è possibile anche gestire molteplici aspetti della partizione gestionale del sito. Lo status di autenticazione di un utente e crittografia delle credenziali, viene effettuato tramite PHP come i controlli sulle form di gestione dei prodotti offerti. Per il primo vengono impiegate delle variabili di sessione dotate di una scadenza che impedisce eventuali utilizzi impropri. Inoltre, tramite esse, è

possibile verificare se un utente non autenticato tenta di accedere ad una pagina amministrativa rimandandolo alla pagina di log-in. La loro forza è di risiedere *server-side* e perciò godono dell'immutabilità da parte dell'utente. Per quanto riguarda le form di amministrazione, il comportamento di PHP, è lo stesso illustrato per la form cliente.

### 3.4.5 Accessibilità

Al fine di rendere il sito più accessibile, sono stati utilizzati alcuni attributi HTML.

- **label:** usato nei form per facilitare la selezione del campo input;
- **alt:** utilizzato per fornire una descrizione dell'immagine;
- **tabindex:** per garantire uno spostamento corretto tra i vari link;
- **lang:** utilizzato per indicare in che lingua deve essere letta la frase o la parola.

### Altri elementi di accessibilità

- Sono stati evitati i link circolari per evitare il problema del "*Lost in navigation*" mettendo l'utente non autenticato in condizione di muoversi orizzontalmente sulla struttura del sito e dato la possibilità all'utente autenticato di accedere a livelli nelle sezioni;
- in alcune pagine del sito web sono state rimosse le immagini nella versione mobile, per rendere il caricamento della pagina più veloce;
- è stato aggiunto un link non visibile, il quale viene letto dallo screen reader e permette all'utente di saltare il menù di navigazione e accedere direttamente al contenuto;
- non sono state utilizzate **accesskey** in quanto divenute obsolete e rischiano di creare disagio all'utente, soprattutto perché si rischia di andare in conflitto con altre shortcut già esistenti.

### 3.4.6 Accesso alle pagine

Per una corretta navigazione e per evitare eventuali furti di credenziali, è stato regolato l'accesso alle pagine con Apache tramite il file *".htaccess"*. Tramite la sua configurazione, Apache segue le seguenti regole per erogare il servizio:

- Se viene richiesta la root del sito, Apache restituisce il file *"index.php"* dopo averlo processato;
- Se viene richiesta una pagina inesistente, Apache risponde con una pagina 404 personalizzata che contiene un collegamento alla homepage del sito.

- Se, vengono richiesti files di configurazione contenenti credenziali con estensione ".ini", Apache restituisce un codice d'errore di accesso negato

### 3.4.7 Versioni software

Il sito è stato sviluppato utilizzando determinate versioni di software di seguito elencate:

- Apache V2.4
- PHP V7.2
- MySQL MariaDB V10.1

## 3.5 Criticità

Durante lo svolgimento del progetto sono emerse alcune criticità. Questa sezione espone i problemi e le soluzioni individuate.

### 3.5.1 XHTML e CSS

Le principali difficoltà relative a questa parte sono legate alla gestione stessa del layout. Il menù ed il contenuto sono stati difficili da gestire soprattutto per colpa della lunghezza della pagina e per il ridimensionamento della stessa. Il primo passo è stato quello di creare delle pagine CSS dedicate ad ogni pagina ma dopo esserci accorti che la struttura era ridondante abbiamo deciso di creare un file di base. Questo però ha portato ad una conseguente modifica e creazione di pagine dedicate che modificavano ad hoc gli elementi che differivano. Al loro interno sono stati inseriti tutti i file CSS di base che però hanno portato ad un leggero "lazy loading" delle pagine.

Di notevole importanza è stato il controllo effettuato sulla "rottura" del layout che in alcune pagine ha comportato la spesa di molto tempo per l'aggiustamento dovuto ad errori legati a margin e padding. La struttura mobile è stata decisa dopo una riunione ed è stata la più pulita e semplice possibile ma comunque ha comportato difficoltà nella creazione corretta del menù.

Un'ultimo problema è stato riscontrato nell'allineamento del footer e delle relative immagini, risolto dopo arduo lavoro di test e prove sul CSS di base.

### 3.5.2 Dati

Durante la modellazione dei dati, abbiamo pensato di gestire l'autenticazione cliente dando la possibilità di registrarsi e tenere la corrispondente entità separata rispetto a quella del defunto nella produzione dello schema logico-relazionale. La scelta di unificare le due entità è stata guidata dai seguenti due parametri:

- **Fattore tempo:** nonostante la ridotta dimensione dell'effort di gestione di due categorie di utente server-side, il corrispondente sforzo dal punto di vista della creazione del contenuto e dello stile per un eventuale altro tipo di utente avrebbe superato la disponibilità oraria del progetto.
- **Tecnologie:** per produrre un prodotto di qualità, sorgeva l'esigenza di implementare un meccanismo di verifica dell'utente tramite indirizzo email o numero di telefono. Questa operazione sarebbe risultata difficile da implementare lasciando incompleto il sito.
- **User experience:** nonostante anche le generazioni non native digitali ormai abbiamo acquisito una certa manualità con operazioni di registrazione e accesso, abbiamo preferito lasciare all'utente la sottomissione di un preventivo nella maniera più rapida possibile considerando anche le difficoltà derivate dallo stress che può colpire un cliente che ha appena subito un lutto.

Un ulteriore problema consisteva nella rimozione dei prodotti: sempre pensando alla natura informativa del preventivatore, la rimozione di un elemento dalla lista prodotti, comporta la rimozione di tutti i preventivi contenenti quell'elemento.

### 3.5.3 JavaScript

Nello sviluppo del sito, il JavaScript viene utilizzato in maniera sporadica per la validazione del contenuto delle form usando i comandi più diffusi e supportati da tutti i browser. Unica criticità degna di nota è la presenza di istruzioni ridondanti per il focusing dei campi interessati dagli errori: alcuni browser, in seguito ad un click su un ancora con un riferimento ad un id di un tag input, non lo focalizzano e rendono perciò impossibile effettuare la compilazione di tale campo se non tramite una doppia operazione di tabulazione in due direzioni differenti. Per risolvere ciò, tramite una libreria creata ad-hoc dal team, vengono intercettati gli eventi di click e di pressione del tasto enter sul messaggio di errore, focalizzano il campo interessato.

### 3.5.4 PHP

Una prima difficoltà riscontrata dal team nella produzione di codice per il preprocessamento, è la ridondanza di codice di utilizzo del driver del database. Per questo, è stata creata una classe *wrapper*, contenente le primitive di gestione della connessione. Altra leggera ma non insormontabile difficoltà è stata trovata nel riprodurre server-side i messaggi di errore in caso di mancata validazione client-side.

## 4 Testing

Per la fase di test sono stati utilizzati alcuni strumenti per la validazione del codice e per verificare l'accessibilità del sito web. Nello specifico:

- **W3C CSS Validator** con cui sono stati validati tutti i file CSS;
- **W3C HTML Validator** con cui sono state validate le pagine HTML;
- **TotalValidator** tramite cui é stato validato il codice HTML nella sua interezza;
- **NVDA**, il quale ha permesso di effettuare alcuni controlli sull'accessibilità del sito;
- **Google Chrome Vox**, il quale ha permesso di effettuare alcuni controlli sull'accessibilità del sito.

Tutti i test HTML vengono effettuati dopo il pre-processamento (ove effettuato) in quanto nei template XHTML sono presenti dei tag *segnaposto* che non appartengono allo standard XHTML 1.0 Strict ma che vengono sostituiti con elementi standard dal pre-processore.

## 5 Installazione

Per la realizzazione del sito sono stati impiegati solo path relativi perciò esso può venire trasferito su qualsiasi hosting. Unica attenzione va prestata al file di indicazioni per Apache "*htaccess*" che va modificando inserendo il path assoluto della pagina 404 altrimenti irraggiungibile. Per quanto riguarda il database, viene fornito un file di creazione del database nella subdirectory del progetto *sql*. L'utente, le credenziali, il nome database e l'indirizzo sono configurabili nel file *credential.ini* presente nella root del progetto.

## 6 Organizzazione del lavoro

Lo sviluppo del sito web é stato organizzato in modo tale che ogni membro del gruppo potesse creare separatamente alcune pagine HTML cosí come i fogli di stile CSS e i codici sorgenti JavaScript e PhP seguendo le linee guida prese dal team. Tramite periodici brainstorming, una volta individuata la struttura, abbiamo effettuato la stesura dei task da effettuare per portare a termine il progetto inserendoli nel gestionale di progetto "*Boards*" fornito da GitKraken. É possibile visualizzare lo sviluppo nella repository di GitHub:

<https://github.com/bettaz/TecWeb2019>

- Referente: Alessio Bettarello
- Struttura HTML: Mattia Gottardello, Nicolás Sartor
- CSS desktop/mobile: Mattia Gottardello, Nicolás Sartor
- CSS stampa: Mattia Gottardello
- Database: Mattia Gottardello
- PHP: Alessio Bettarello, Olivier Utshudi
- Accessibilità: Olivier Utshudi
- Javascript: Olivier Utshudi, Alessio Bettarello
- Contenuti: Mattia Gottardello, Olivier Utshudi
- Relazione: Nicolás Sartor, Olivier Utshudi
- Testing & Validazione: Alessio Bettarello, Mattia Gottardello, Olivier Utshudi, Nicolás Sartor