



# UNIVERSITÀ DI TORINO

## Progetto T-Web

Milani Alessandro 947200

Anno Accademico 2021/2022

### Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Tema del sito	2
1.2	Sezioni principali	2
<b>2</b>	<b>Funzionalità</b>	<b>2</b>
2.1	Registrazione	2
2.1.1	Login/Logout	2
2.2	Contenuto generato dall'utente	2
2.3	Ricerca distribuzioni	2
<b>3</b>	<b>Caratteristiche</b>	<b>3</b>
3.1	Usabilità	3
3.2	Animazioni	3
3.3	Sessioni	3
3.4	Database	3
3.5	Validazione input e sicurezza	3
3.6	Presentazione	4
<b>4</b>	<b>Front-end</b>	<b>4</b>
4.1	Separazione presentazione/contenuto/comportamento	4
4.2	Gestione Cartelle	4
<b>5</b>	<b>Back-end</b>	<b>5</b>
5.1	Architettura back-end	5
5.2	Schema DB	5
5.3	Descrizione funzioni backend	5

# 1 Introduzione

## 1.1 Tema del sito

Il sito prende ispirazione da DistroWatch.com.

Permette di ricercare tra un gran numero di distribuzioni Linux, visualizzarne dettagli e descrizioni, lasciare commenti e crearsi una propria lista di preferiti.

## 1.2 Sezioni principali

Il sito è diviso in 5 pagine principali (più la pagina admin)

- **Homepage:** pagina di benvenuto, con carosello con 3 distribuzioni random e qualche info sul sito
- **Distro Finder:** Pagina di ricerca delle distribuzioni, che mostra i dati essenziali
  - **Linux:** pagina con descrizione completa, accessibile dalle card singole tramite pagina finder/favorite o cliccando Tux nella navbar per visualizzarne una random
- **Favorite:** Pagina con le distribuzioni aggiunte ai preferiti
- **Profilo:** Permette di modificare la password e l'immagine di profilo
- **Admin Page:** Permette di caricare una distribuzione e modificarne i dati nel DB (essendo una pagina a uso esclusivo del/degli admin è stata lasciata volutamente più spoglia dal lato estetico)

# 2 Funzionalità

## 2.1 Registrazione

Per accedere al sito è necessario registrarsi e per la precisione saranno richiesti all'utente un nome (solo lettere e numeri), una password (di minimo 8 caratteri) e la conferma della password.

In caso di eventuali errori verrà visualizzato un messaggio di errore tramite un div che verrà reso visibile al momento opportuno.

Se i dati rispettano le condizioni sopra citate, si fa un ulteriore check per controllare che il nome non sia già in uso. Per finire si eseguono salting e hashing della password per poi salvare il tutto nel db e far partire la sessione (reindirizzando direttamente l'utente alla homepage)

### 2.1.1 Login/Logout

Per il Login si controlla l'esistenza dell'utente e la correttezza della password. In caso di errore (user non esiste o password sbagliata) verrà visualizzato un messaggio di errore simile a quello presente nella pagina di registrazione.

Per il logout, l'utente può effettuarlo da qualsiasi pagina si trovi premendo sulla sua icona visibile nella navbar e successivamente sulla voce relativa, la quale porterà una chiamata al backend per chiudere la sessione utente.

## 2.2 Contenuto generato dall'utente

L'utente può generare 2 tipi di contenuti:

- **Lista Preferiti:** dalla pagina del finder l'utente può aggiungere delle distribuzioni tra i preferiti premendo sulla classica icona a forma di cuore (che diventerà verde in caso di successo, ciano se già presente o giallo in caso di problemi)  
Successivamente dalla pagina preferiti, sarà possibile visualizzare tutte le distribuzioni salvate ed eliminarle prendendo la card in questione e spostandola nella "drag-zone" (creata con JQuery UI) facilmente individuabile
- **Commenti:** L'utente può anche lasciare dei commenti nelle pagine con le info complete delle distribuzioni. Saranno poi visualizzabili dal più vecchio al più recente

## 2.3 Ricerca distribuzioni

La ricerca delle distribuzioni presenta 3 campi compilabili a propria scelta:

- **Distro Name:** campo libero, in cui l'utente può scrivere il nome della distribuzione che cerca o parte di esso
- **Base:** Se non si cerca una distribuzione nello specifico, si può filtrare per base della distribuzione (es. tutte le derivate di debian)
- **Region:** Filtro di ricerca simile al precedente, ma si basa sulla regione di origine della distribuzione

## 3 Caratteristiche

### 3.1 Usabilità

Per garantire l'usabilità del sito, la quasi totalità delle azioni fornisce un feedback (principalmente cambio colore di un elemento o comparsa di uno nuovo che attira l'attenzione), in più si sono mantenuti colori che non affatichino la vista e un più generale tema scuro per il medesimo motivo.

### 3.2 Animazioni

Le animazioni presenti nel sito sono il carosello nella homepage e il "drag and drop" delle distribuzioni nella pagina dei preferiti, le quali torneranno alla loro posizione definita in automatico se rilasciate al di fuori della drop-zone.

### 3.3 Sessioni

All'interno di ogni pagina php, come prima cosa, viene verificata l'esistenza di una sessione attiva. Se esiste, si recuperano i dati necessari e si prosegue con caricare il resto della pagina, altrimenti si viene reindirizzati sulla pagina di login.

I dati recuperabili dalla sessione sono:

- **username:** autoesplicativo
- **Role:** indica se l'utente è un admin o no
- **img:** Indirizzo dell'immagine di profilo
- **logged:** booleano per conferma che si abbia effettuato il login

In più la sessione viene anche recuperata all'inizio dei file php di backend nel caso sia necessario operare con alcuni dei dati sopraindicati all'intero delle funzioni del file.

### 3.4 Database

Durante la navigazione vengono eseguite le seguenti query:

- **Login:** richiesta al db per vedere se l'utente esiste e la password è corretta
- **Registrazione:** si controlla che l'utente non sia già registrato e se va bene si esegue l'INSERT
- **Ricerca distro:** Si esegue il select delle distro in base agli input dell'utente
- **aggiungi/rimuovi distro:** rispettivamente dalla pagina finder e favorite si eseguono le query per aggiungere e rimuovere la distro dai preferiti
- **Caricamento info distro:** quando si va alla pagina linux.php, se è presente un messaggio di get si fa il select della distro a cui si riferisce, altrimenti ne viene presa una random
- **Caricamento commenti:** Select dei commenti della distro che si sta visualizzando
- **Caricamento preferiti:** Select nella table favs con l'utente come filtro e join con distro per avere tutte le info
- **Modifica profilo:** Update della tupla dell'utente con i nuovi dati inseriti

### 3.5 Validazione input e sicurezza

Per minimizzare eventuali problemi di sicurezza, ho optato per ridurre i campi a inserimento libero per gli utenti. Per i pochi rimasti (pagina admin, login e finder per il nome distro) si sono prese le misure necessarie con la chiamata di htmlspecialchars() per evitare XSS.

L'esecuzione di ogni query è comunque preceduta da unprepare() di PDO per evitare casi di SQL Injection.

Infine, per evitare accesso alle pagine senza prima aver effettuato il login viene sempre richiamata la sessione php così, nel caso non risulti una sessione in corso, si viene reindirizzati alla pagina di login (mentre se si tenta di accedere a un file di backend, viene restituito un errore).

### 3.6 Presentazione

Le pagine del sito, sono divise in 3 parti principali:

- **Navbar:** contiene, in ordine: l'icona del sito per distribuzione random, i pulsanti per la navigazione tra le pagine principali e l'immagine profilo per pagina profilo/pagina admin/logout
- **Body:** Corpo principale del sito, costruito principalmente con componenti di Bootstrap 5
- **Footer:** semplice footer che comprende: "Nome cognome matricola" e il collegamento a gli account Twitter e GitHub

## 4 Front-end

### 4.1 Separazione presentazione/contenuto/comportamento

Per mantenere il più possibile separate le parti del progetto scritte in php/js/html, grossa parte del carico di lavoro è stata caricata alle funzioni JavaScript il quale fa da tramite tra il frontend e il backend. Per esempio quando si vuole far partire la ricerca di una distribuzione, JS recupera i dati dal form, li gira al backend che esegue la query, la quale risponde con un oggetto json che contiene le varie distribuzioni con le proprio info, a questo punto JS controlla la presenza d'immagini sul server (altrimenti carica quelle di default), impacchetta il tutto in una card di Bootstrap e ne esegue l'append alla lista.

Oltre questo, JavaScript si occupa anche di:

- fare da event listener per i form
- fare un primo check dei dati che l'utente inserisce nei campi di testo e visualizzare eventuali messaggi di errore

### 4.2 Gestione Cartelle

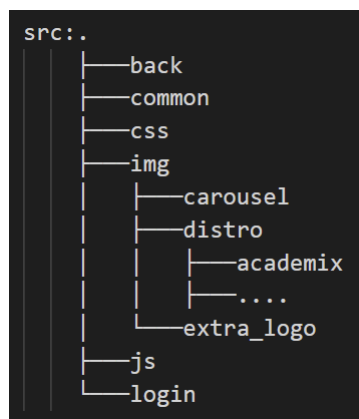


Figura 1: albero cartelle

- **back:** Contiene tutti i file di backend/API
- **common:** File html che vengono riutilizzati in più pagine (header, navbar, footer)
- **img:** contiene immagini e icone di tutte le distro + delle immagini predefinite per quelle sprovviste
- **css:** Contiene i fogli di stile di ogni pagina + un file common che raccoglie dettagli presenti in 2 o più pagine
- **JS:** contiene tutti i file JavaScript
- **login:** presenta i file html per login/signup
- **root:** Il resto delle pagine navigabili risiedono direttamente nella root del sito (insieme all'favicon.ico)

## 5 Back-end

### 5.1 Architettura back-end

Le funzioni del backend sono raggruppate per file in base alla loro pagina di riferimento con 3 eccezioni: Il file per il logout, quello per la connessione al database e la gestione dei commenti alle distro.

Per chiamare una funzione precisa da JS, basta includere un *action*: "*funzione*" nella sezione data **della** funzione ajax (mentre l'url punterà al file che la contiene)

### 5.2 Schema DB

Il DB mysql è composto da 4 tabelle dai nomi autoesplicativi per semplificarne l'uso

- **User**: username, password, salt, role, img
- **distro**: name, based, region, arch, desc, web, version
- **favs**: username, distro
- **comments**: id, username, distro, comment, date

### 5.3 Descrizione funzioni backend

Per chiarezza, la seguente lista è in ordinata in ordine alfabetico usando i nomi dei file e tutte le chiamate saranno POST (a meno di ulteriore specifica).

- **back/admin\_manager.php**:
  - *loadDistroData*:
    - \* action: loadDistroData
    - \* distro= nome della distribuzione
    - \* restituisce un json con un campo status che può avere i valori success/error, in caso di success sono presenti i campi della distro, altrimenti un campo message
    - \* Chiamata dal pulsante load nella admin page
  - *saveDistroData*
    - \* action: saveDistroData
    - \* data = json con i dati modificati della distribuzione
    - \* Restituisce un json con un campo status che può avere i valori success/error e un messaggio relativo
    - \* chiamata dal pulsante edit nella admin page
- **back/comments\_manager.php**:
  - *loadComments*:
    - \* action: loadComments
    - \* distro = nome della distro di cui devo caricare i commenti
    - \* Restituisce un json con un campo status che può avere i valori success/error, in caso di success, anche un campo con i vari messaggi, altrimenti un messaggio con l'errore
    - \* chiamata automatica al caricamento della pagina
  - *addComment*:
    - \* action: addComment
    - \* distro = nome della distribuzione | comment = contenuto del commento
    - \* Restituisce un json con un campo status che può avere i valori success/error, in caso di success anche i campi con i dati inviati arrivati di timestamp e username, altrimenti un campo message con l'errore
- **back/favs\_manager.php**:
  - *removeFav*:
    - \* action: remove

- \* distro = nome della distro da rimuovere dai preferiti
- \* Restituisce un json con i campi status che può avere i valori success/error, il primo accompagnato dal campo empty che indica se sono rimaste altre distro tra le preferite, il secondo da un messaggio
- \* Chiamata con il drop delle card distro nella drop-zone (pagina favorite)
- *getFavs*
  - \* action: getFavs
  - \* ritorna un json con campo status dal valore success/empty/error, il primo accompagnato dalle distro da mostrare, gli altri 2 da eventuali messaggi
  - \* chiamata al caricamento della pagina favorite
- **back/finder\_manager.php:**
  - *getDistro:*
    - \* action: getDistro
    - \* name = nome distro | base = famiglia della distro | region = paese della distro
    - \* Ritorna un json contenente le varie distribuzioni che passano i filtri dell'utente
    - \* chiamato dal pulsante search nella pagina finder
  - *addFav:*
    - \* action: addFav
    - \* distro = nome della distribuzione
    - \* ritorna un json con i campi status (success/present/error) e message (eventuale messaggio correlato)
    - \* chiamata dal pulsante presente nella card di ogni distro nella pagina finder
- **back/linux\_manager.php**
  - *getFullDistro*
    - \* chiamata GET
    - \* name = nome distro
    - \* ritorna le informazioni relative alla distribuzione se esiste, altrimenti un json con campo status = error e un messaggio
    - \* viene chiamata direttamente all'apertura della pagina
  - *randomDistro:*
    - \* chiamata se getFullDistro ritorna un array vuoto
    - \* recupera i dati di una distribuzione random
- **back/login\_manager.php:**
  - chiamata con il tasto login nella relativa pagina
  - username, password = campi inseriti dall'utente
  - restituisce un json con i campi status (success/error) e message (tipo di errore)
- **back/profile\_manager.php:**
  - *changePswd:*
    - \* action: changePswd
    - \* pswd1, pswd2 = i due campi per il cambio password
    - \* se le due password coincidono viene aggiornato il database. Ritorna sempre un json con i campi status/message, il contenuto varia in base al risultato
  - *changeUrl:*
    - \* action: changeUrl
    - \* url = nuovo url dell'immagine
    - \* tentato l'aggiornamento del db, ritorna un json con campi status e message
- **back/signup\_manager.php**
  - username, password = campi inseriti dall'utente e per il nuovo account
  - controlla che l'utente non esista, e in caso positivo prende le misure di sicurezza necessarie e salva i dati nel db per poi postare l'utente alla homepage. Ritorna sempre un json con i campi status (success/error) e message (eventuale descrizione dell'errore)