

- Tema del sito

SSNPlatform è un sistema integrato in grado di offrire supporto alla prescrizione, al tracciamento e al pagamento dei farmaci.

La piattaforma è indirizzata ai clienti del Sistema Sanitario Nazionale, ai medici e ai farmacisti. E' in grado di effettuare controlli automatici per facilitare l'immissione dei dati ed evitare possibili incongruenze.

- Sezioni principali

- **Pagina iniziale:**

Fornisce alcune informazioni di carattere generale all'utente, notificandolo del possibile utilizzo di cookies da parte del sito ed eventualmente indirizzandolo verso il form di login e di registrazione.

- **Form di login e di registrazione**

- **Selezione del ruolo:**

Qualora l'utente si sia registrato come farmacista o come medico, potrà selezionare il ruolo di cliente del SSN o come personale sanitario. A seconda del ruolo selezionato l'utente verrà indirizzato al cruscotto di visualizzazione adeguato.

- **Cruscotto di visualizzazione:**

Permette l'immissione e la visualizzazione dei dati relativi al ruolo precedentemente selezionato.

- Funzionalità

- **Login:**

L'utente dopo aver inserito le credenziali necessarie tenta di effettuare il login (viene eseguito un controllo sulla correttezza di queste ultime per mezzo di AJAX). Qualora il controllo delle credenziali avesse successo l'utente verrebbe indirizzato alla pagina dedicata alla selezione del ruolo la quale inizializzerà una sessione, in caso contrario verrebbe notificata all'utente l'errata compilazione delle credenziali.

- **Logout:**

Chiude la sessione corrente e indirizza l'utente alla pagina di login.

- **Registrazione:**

L'utente è tenuto a compilare diversi campi che rappresentano le proprie generalità e il tipo di ruolo che potrà esercitare all'interno del sistema (alcuni campi da compilare sono indicabili dipendentemente dal ruolo, il tutto è stato gestito client side con jQuery). La registrazione avviene per mezzo di email e password associati all'account. Sono stati implementati tutti i controlli necessari sia lato client (per mezzo di jQuery e javascript) sia lato server (per mezzo di PHP). Qualora l'utente riuscisse a generare input invalidi per mezzo di una modifica alla form il server restituirebbe (per mezzo di AJAX) un elenco con tutti gli input non validi che verrebbe poi mostrato nella stessa form di registrazione.

- **Contenuto immesso dall'utente:**

Gli unici ruoli in grado di immettere dati di dominio all'interno del sistema sono il medico (in grado di prescrivere ricette agli utenti) e il farmacista (in grado di vendere agli utenti farmaci prescritti da medici)

- **Contenuto fornito all'utente:**

Gli utenti sono in grado di visualizzare tutti i farmaci a loro prescritti e/o venduti, i medici sono in grado di visualizzare tutte le prescrizioni erogate, e i farmacisti sono in grado di visualizzare tutti gli ordini effettuati.

- Caratteristiche

- **Usabilità:**

- Lo stile con cui il sito è stato concepito è sufficientemente intuitivo, spesso accompagnato da immagini
- Ogni azione non corretta viene notificata all'utente

- **Interazione/animazione:**

L'elemento di interazione/animazione è stato inserito nel form di login e registrazione.

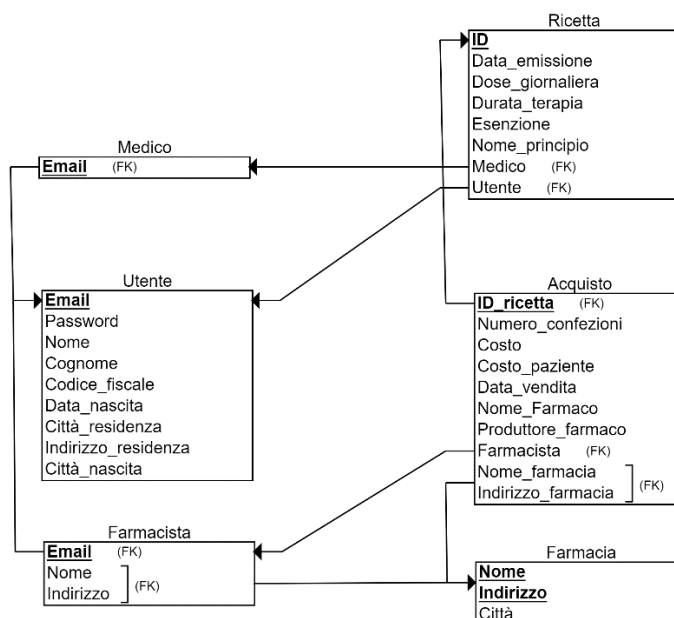
Per mezzo di un apposito bottone il form è in grado di transizionare graficamente dalla modalità di login a quella di registrazione sfruttando regole CSS (in particolare gli attributi 'overflow', 'z-index' e 'transition').

- **Sessioni:**

Le sessioni vengono aperte dalla pagina php incaricata di selezionare il ruolo, alla quale si accede immediatamente dopo l'operazione di login.

- **Interrogazioni del DB:**

Viene riportato di seguito lo schema relazionale della base di dati utilizzata per il progetto con alcune query SQL significative:



- (*)Query di visualizzazione delle ricette prescritte ad un utente:

```
select * from utente join ricetta r on (utente.Email = r.Medico) where Utente =
'{$SESSION[session_id()][0]}';
(in {$SESSION[session_id()][0]} è memorizzata l'email dell'utente associato alla sessione
corrente)
```

- **(**)Query di visualizzazione degli acquisti effettuati da un utente:**

```
select a.ID_ricetta, a.Numero_confezioni, a.Costo, a.Costo_paziente, a.Data_vendita,
a.Nome_Farmaco, a.Produttore_farmaco, u.Nome, u.Cognome, u.Email,
a.Nome_farmacia, a.Indirizzo_farmacia from acquisto a join utente u on (a.Farmacista = u.Email)
join ricetta r on (a.ID_ricetta = r.ID) where Utente = '{$ SESSION[session_id()][0]}';
```

- **(***)Query per l'aggiunta delle prescrizioni da parte dei medici:**

```
INSERT INTO ricetta (Dose_giornaliera, Durata_terapia, Esenzione, Nome_principio, Utente,
Data_emissione, Medico) VALUES ('{$ POST["Dose_giornaliera"]}', '{$ POST["Durata_terapia"]}',
'{$ POST["Esenzione"]}', '{$ POST["Nome_principio"]}', '{$ POST["Utente"]}', '{$currentDate}',
'{$ SESSION[session_id()][0]}');
```

- **Validazione dati input:**

La validazione dei dati è stata eseguita sia client-side (HTML, JS e jQuery) sia server-side(PHP), spesso con l'ausilio di espressioni regolari.

- **Sicurezza:**

- Sono stati evitati attacchi di tipo SQL Injection utilizzando la funzione 'real_escape_string()'
- E' stata scelta la funzione di hash resistente alle collisioni 'SHA-256' per memorizzare le password inserite dagli utenti all'interno della base di dati
- In ogni pagina a cui si potrebbe accedere solo dopo il login sono stati fatti numerosi controlli per evitare che l'utente riesca ad accedere a funzioni del sistema senza aver effettuato il login, oppure avendo falsificato l'ID della sessione corrente con uno non esistente
- Vengono fatti controlli sui ruoli per evitare che un utente che non possiede un determinato ruolo acceda a regioni del sito che non gli competono, in questo caso l'utente verrebbe indirizzato automaticamente alla pagina di selezione ruolo

- **Relazione:**

- **Front-end:**

(Tutte le pagine tolta 'LoginSignupForm.php' includono una barra di navigazione (implementata per mezzo di Bootstrap) in grado di indirizzare l'utente a LoginSignupForm.php oppure a Logout.php)

- **Index.html:**

E' contenuta al suo interno una breve descrizione del funzionamento del sito e un avviso che notifica l'utente del possibile utilizzo dei cookies (conformemente al GDPR). Per l'avviso è stato utilizzato codice js per evitare che in futuro, se l'utente accetta, l'avviso venga mostrato nuovamente e per l'aspetto sono state utilizzare regole CSS specifiche.

- **LoginSignupForm.php:**

La pagina contiene due pannelli principali associati rispettivamente al form di login e a quello di registrazione, mutualmente visibili. Per cambiare il pannello visibile sono stati introdotti dei bottoni che fungono da trigger per l'animazione implementata per mezzo di CSS (come descritto precedentemente nella sezione 'Interazioni/Animazioni'). Nella form riservata alla registrazione è possibile selezionare il ruolo dell'utente da registrare per mezzo di radiobuttons, a seconda della scelta appariranno o scompariranno all'interno del form degli input dipendenti dal ruolo. Nel caso esistesse già una sessione in corso la pagina indirizzerebbe automaticamente l'utente a ChooseRole.php evitando così di ripetere il login.

- **ChooseRole.php:**

La pagina viene generata dinamicamente in relazione ai ruoli disponibili per l'account associato alla sessione. Al centro della pagina vengono caricati i loghi rappresentanti i ruoli disponibili, così facendo l'utente può cliccarli per venire indirizzato ai cruscotti di visualizzazione

(inizialeUtente.php, inizialeMedico.php o inizialeFarmacista.php). Si occupa anche di creare le sessioni in corso.

- **Funzioni utente:**

(Tutte le pagine includono almeno una tabella (implementata per mezzo di Bootstrap) in grado di visualizzare il contenuto delle query DML elencate in precedenza.

Inoltre appare in ogni pagina un logo rappresentante il ruolo selezionato e all'interno della barra di navigazione appare un bottone che permette di riaggiornare i risultati della pagina)

- **inizialeUtente.php:**

La barra di navigazione permette di visualizzare la tabella in grado di mostrare il risultato della query (*) oppure il risultato della query (**)

- **inizialeMedico.php:**

La tabella inclusa all'interno della pagina mostra il risultato della query necessaria per la visualizzazione delle prescrizioni da parte del medico. La barra di navigazione permette l'esecuzione di uno script JS che rende il form di compilazione delle nuove ricette visibile. Per l'implementazione del form è di particolare importanza impostare l'attributo di stile 'z-index', in modo tale che il form venga sovrapposto alla tabella mostrata in precedenza. Il form è dotato dei controlli generali esposti in precedenza, più in dettaglio le regole di business impongono che non possa essere associata una ricetta ad un account inesistente, qualora si verificasse un problema del genere (per mezzo di AJAX) il riquadro del form lo farebbe presente all'utente. All'interno del riquadro è presente un bottone che (sempre per mezzo di JS) rende possibile la scomparsa del form stesso così da poter tornare alla configurazione iniziale. Un form simile è presente per la cancellazione delle ricette, la cancellazione può andare a buon fine solo se non c'è stata già una vendita da parte di un farmacista, in caso contrario il medico verrà notificato dell'errore (sempre per mezzo di AJAX).

- **inizialeFarmacista.php:**

La struttura della pagina è molto simile a inizialeMedico.php, cambia soltanto la query mostrata dalla tabella in primo piano. Inoltre anche i controlli sono differenti, il farmacista non può vendere farmaci per i quali esiste una prescrizione già espletata. Anche in questo caso, se si verificasse un errore verrebbe visualizzato un messaggio in maniera simile a come esposto in precedenza.

- **Back-end:**

- **SignUp.php:**

Prende come parametri i dati immessi dal form di registrazione, effettua controlli di consistenza (tra cui quello di non aver utilizzato un'email già utilizzata nel sistema per creare il nuovo account) e restituisce un output. L'output vale '1' se la registrazione è andata a buon fine, altrimenti restituisce un elenco di tutti i parametri che presentano errori nella compilazione.

- **Login.php:**

Implementa la funzione in grado di prendere in input email e password dell'utente e restituisce se il login è valido o meno rispetto al sistema.

- **LoginEvaluator.php:**

Sfrutta la funzione principale di Login.php e restituisce 1 se il login è valido altrimenti restituisce 0, la pagina verrà poi chiamata per mezzo di una call POST.

- **Logout.php:**

Si occupa di distruggere le connessioni aperte e indirizzare gli utenti al form di login.

- **connectdb.php:**

Implementa l'accesso alla base di dati, contiene anche username, password e nome della base di dati relativi al dbms da utilizzare.

- **addPrescription.php:**
Esegue controlli sui parametri passati per mezzo della chiamata POST, se non ci sono errori esegue la query (***) restituendo 1, altrimenti fornisce in output l'elenco dei parametri che generano gli errori.
- **removePrescription.php:**
Esegue controlli sui parametri passati per mezzo della chiamata POST, se non ci sono errori esegue la query adatta alla rimozione della prescrizione selezionata restituendo 1, altrimenti fornisce in output l'elenco dei parametri che generano gli errori.
- **addOrder.php:**
Esegue controlli sui parametri passati per mezzo della chiamata POST, se non ci sono errori esegue la query adatta all'aggiunta degli ordini restituendo 1, altrimenti fornisce in output l'elenco dei parametri che generano gli errori.
- **removeOrder.php:**
Esegue controlli sui parametri passati per mezzo della chiamata POST, se non ci sono errori esegue la query adatta alla rimozione degli ordini restituendo 1, altrimenti fornisce in output l'elenco dei parametri che generano gli errori.