CSS E JS

Le due cartelle contengono i file .css e .js del progetto.

* **commonStyle.css**: definisce lo stile condiviso tra le pagine e le variabili :root.
* **formStyle.css**: gestisce lo stile delle pagine di login, registrazione e del modal "Nuovo Viaggio" nella dashboard.
* **formScript.js**: contiene le funzioni JavaScript per login e registrazione.

Gli altri file .css e .js sono associati alle rispettive pagine HTML.

PHP

La cartella contiene file PHP condivisi tra più pagine:

* **emailUtente**: utilizzato in tutte le pagine con la barra superiore; imposta l'email dell’utente nel pannello di logout.
* **logout**: presente in ogni pagina con barra superiore; gestisce il logout della sessione corrente.
* **infoViaggio**: restituisce i dettagli del viaggio tramite ID; usato nel modal di modifica nella dashboard, nella valigia per aggiornare il banner laterale e nel planner per aggiornare dati e calendario.
* **aggiornaViaggio**: aggiorna il database con le nuove informazioni del viaggio fornite in input.

**login.html + login.php**

La pagina *login.html* include i campi email e password, che vengono inviati via POST al file *login.php*, che si occupa dell'autenticazione sul server. Se le credenziali sono corrette (confrontate con il database), viene creata una sessione utente e si viene reindirizzati a *dashboard.html*.

**registrazione.html + registrazione.php**

La pagina *registrazione.html* include i campi nome, cognome, email password e ripeti password. All’invio del form viene verificato che le password corrispondano e i dati vengono inviati via POST al file *registrazione.php*, che verifica che l’email non sia già registrata, quindi salva il nuovo utente nel database usando un hash per la password. Se la registrazione ha successo, viene creata una sessione utente e si viene reindirizzati a *dashboard.html*.

DASHBOARD

Contiene il file **dashboard.html** e i relativi file PHP:

* **crea\_viaggio**: crea un nuovo viaggio nel database.
* **dashboard**: recupera dal database la lista dei viaggi in base al filtro selezionato nella dashboard.
* **eliminaViaggio**: elimina dal database il viaggio selezionato.

**crea\_viaggio.php**

Gestisce l’inserimento di un nuovo viaggio nel database. Riceve destinazione, date e nome del viaggio e li inserisce nel database. Successivamente, salva l’ID del viaggio in sessionStorage e reindirizza l’utente ad *attivita.html*.

**attivita.html + attivitaScript.js**

Questa pagina permette all’utente di selezionare attività, mezzi di trasporto e tipologie di alloggio legati a un viaggio. Ogni opzione è rappresentata con un’immagine e una checkbox. Quando almeno una checkbox è selezionata, il pulsante “Crea la valigia” si attiva. Al clic su questo pulsante, lo script *attivitaScript.js* seleziona le checkbox scelte e le invia tramite una fetch POST a *crea\_lista.php*, includendo l'ID del viaggio memorizzato nel sessionStorage. Il server elabora queste scelte e, se la risposta è positiva, l’utente viene reindirizzato alla pagina *valigia.php* corrispondente allo specifico viaggio.

**crea\_lista.php**

Questo script riceve in POST un viaggio\_id e un array selectedActivities (in formato JSON). Ad ogni attività selezionata in *attivita.html* corrisponde una lista di oggetti. Per ogni valore di selectedActivities vengono inseriti gli oggetti corrispondenti nel database, ognuno associato all’id del viaggio e alla sua categoria (il nome dell’attività).

**valigia.php**

Questa è la pagina di visualizzazione e modifica della valigia. Recupera il viaggio\_id dall’URL (GET) e carica dal database gli oggetti collegati a quel viaggio, organizzati per categoria. I dati vengono poi mostrati in HTML e ogni oggetto è reso interattivo tramite *valigiaScript.js* (modifica quantità, aggiunta, eliminazione, ecc.). Inoltre, a sinistra è possibile visualizzare i dettagli del viaggio e passare al *planner.html* tramite un bottone.

**valigiaScript.js**

Questo script gestisce in modo dinamico la modifica della lista oggetti da mettere in valigia. Consente di:

* Modificare la quantità di ogni oggetto con i pulsanti +/−;
* cambiare lo stato di un oggetto con la checkbox;
* aggiungere nuove sezioni e oggetti personalizzati;
* eliminare elementi;
* espandere o comprimere sezioni.

Tutte queste azioni inviano richieste JSON al backend (*aggiornaValigia.php*) per aggiornare i dati del database.

**aggiornaValigia.php**

Si occupa di ricevere richieste dal file *valigiaScript.js* in formato JSON. Ogni richiesta contiene un’azione (add, updateQuantity, updateStatus, delete) e i relativi dati (ID viaggio, oggetto, quantità, ecc.). Lo script decodifica il JSON, esegue l’azione sul database e risponde con un messaggio di conferma o errore.

PLANNER

Contiene **planner.html** e i relativi file PHP:

* **caricaEventi**: recupera dal database tutti gli eventi relativi al viaggio fornito in input.
* **gestisciEvento**: in base al parametro ‘action’,consente di aggiungere, modificare o eliminare un evento nel calendario.
* **infoEvento**: restituisce tutte le informazioni associate all’evento richiesto.