

MODEL

Per quanto riguarda l'accesso ai dati necessari è stata implementata una connessione remota ad un database Apache Derby in host locale. Sono presenti 6 tabelle.

La tabella "Utenti" contiene i dati personali degli utenti, la tabella "Attività" contiene i titoli delle attività presenti, e grazie alla tabella "Partecipa" è possibile vedere quali utenti partecipano a quale attività. Le tabelle "Visite" e "Donazioni" contengono la quantità finale di, rispettivamente, visite alle pagine del sito e donazioni divise nei 12 mesi. Nel Database Management System sono presenti le funzioni necessarie per poter manipolare e visualizzare questi dati.

VIEW

Il package "Frammenti" contiene frammenti di codice html e jsp che vengono poi assemblati per creare le effettive pagine del sito (nel package "Pagine"). Il package "Javascript" contiene gli script utilizzati che si occupano di validare i form, modificare la citazione presente nello Header, far funzionare correttamente la dashboard, mandare una mail, aumentare i contatori delle visite al sito e gestire la comparsa del popup per la richiesta dei cookies. Alcune funzioni javascript comunicano con il database. In "Css" sono presenti quattro fogli di stile: un foglio comune a tutto il sito e tre specifici per le aree personali delle tre categorie di utenti. Il package "Immagini" contiene le figure mostrate nel sito. Sempre in webapp sono contenuti due file json utilizzati nella lettura dei dati per gli istogrammi.

CONTROLLER

La classe "Sessione" contiene un metodo per creare una nuova sessione e riportare l'utente ad una pagina di errore nel caso la sessione risulti invalida. La servlet "CookieStuff" si occupa di verificare se esiste già un cookie e di crearne di nuovi. "LogoutServlet" esegue il logout dell'utente. "Index" è la servlet di partenza che si occupa dell'url encoding.

Esistono quattro filtri: "SessionFilter" crea una nuova sessione, se necessario, o rimanda ad una pagina di errore se la sessione risulta nulla; "CookieFilter" controlla se sia necessario chiedere all'utente di accettare i cookie; "LoginFilter" gestisce l'accesso all'area personale; "SignInFilter" gestisce l'accesso alla pagina per il signIn e alla relativa pagina di conferma.

CONTRIBUTI DEGLI STUDENTI

Inizialmente, abbiamo definito insieme l'organizzazione del lavoro. Abbiamo cercato di fare in modo che ognuno lavorasse a diversi aspetti del progetto, per ottenere una maggiore comprensione d'insieme. Tuttavia, ognuno si è concentrato su degli aspetti particolari. Nel momento in cui sono stati riscontrati problemi, è stato utile confrontarsi con gli altri membri.

- Alberto Cimmino ha lavorato principalmente al database e alle funzioni necessarie per interagire con esso. Ha scritto l'HTML per le pagine "Homepage", e "Attività", e assieme a Tait, il codice Javascript per le funzioni di amministratore. Si è occupato, spensieratamente, della stesura dei testi e delle immagini e del volantino. Ha portato

a termine le proprie task rapidamente avendo così modo di prestare aiuto ai compagni ove necessario.

- Aurora Ottaviani si è occupata del css e ha uniformato l'html ove necessario. Ha realizzato i frammenti di "Contatti" e "emailConfermata". Ha realizzato i file javascript di Frasi.js e Mail.js.
- Sara Tait si è occupata della business logic, implementando le classi che gestiscono i filtri, la sessione e i cookie. Ha scritto insieme a Cimmino il codice JavaScript per le funzionalità relative alla pagina "Amministratore", e assieme a Zyryanova le funzioni per il popup dei cookies. Si è inoltre occupata delle funzioni per la validazione dei form. Ha realizzato i frammenti "descrizioneA1", "descrizioneA2", "descrizioneA3", "SignIn", "registrazioneConfermata" e "Amministratore". Si è poi occupata di riassemblare i frammenti nelle varie pagine jsp.
- Anastasia Zyryanova ha scritto l'HTML per le pagine "Chi siamo", "Login" e i frammenti "Intestazione", "Menu" e "Footer". Ha scritto il codice HTML e JavaScript per le pagine "Simpatizzante" e "Aderente" ed insieme a Tait le funzioni per il popup dei cookies.

PARTI NON SVILUPPATE

Non è possibile visualizzare le visite e le donazioni effettive del sito (i dati mostrati nell'istogramma sono fittizi). Questo è dovuto ad un problema di lettura e scrittura su file.

Per incrementare le visite (e le donazioni), una servlet scrive i dati su un file, mentre una funzione javascript legge gli stessi dati dal file.

Nel nostro caso, la scrittura e la lettura funzionano correttamente, ma avvengono su file diversi. JavaScript legge il file presente all'interno del progetto, ma Java scrive su un file all'interno della cartella "bin" relativa a Tomcat.