

REPORT GRUPPO 05

Strutturazione del progetto: Il progetto è strutturato seguendo il pattern MVC (Model-View-Controller). La cartella *webapp* contiene la **View**, suddivisa in diverse sezioni come la *dashboard* per la parte privata, *webapp_resources* per le risorse come immagini e i pdf, e *styles* per i fogli di stile. I file *.jsp*, *.html* e *.js* per la parte pubblica si trovano sotto la cartella principale *webapp*. I **Model** e i **Controller** sono situati nella cartella *src > java > com.solidisitiweb.tum4world*. La cartella *model* contiene i file Java per definire il modello dei dati dell'applicazione, mentre le *Servlet* e i filtri esterni rappresentano i *controller* e sono sotto la cartella principale.

Descrizione dei file:

- **webapp**
 - **dashboard**
 - *aderente.jsp*, *simpatizzante.jsp* e *amministratore.jsp* → pagine per la visualizzazione delle pagine private dei diversi livelli di utente
 - *footer.html* e *navbar.jsp* → il primo è il footer comune alle pagine private, il secondo è il menù generale comune sempre alle pagine private
 - **webapp_resources**
 - **img** → tutte le immagini utilizzate per il sito
 - *VolantinoA5.pdf* → volantino del sito in formato A5
 - **styles**
 - *global.css* → foglio di stile delle pagine pubbliche
 - *aderente.css*, *simpatizzante.css*, *amministratore.css* → fogli di stile delle pagine private (per ogni livello di utente)
 - *home.jsp*, *attivit .jsp*, *chisiamo.jsp*, *contatti.jsp*, *login.jsp*, *signin.jsp*, *paginaattivit _1.jsp*, *paginaattivit _2.jsp*, *paginaattivit _3.jsp* → queste sono le pagine principali (pubbliche del sito)
 - *error.jsp*, *invioconfermato.jsp*, *cancellazioneconfermata.jsp*, *registrazioneconfermata.jsp* → pagine chiamate in caso di successo/insuccesso di altre operazioni
 - *header.html* → header della tab del browser
 - *cookie.jsp* → banner dei cookies
 - *footer.html*, *navbar.jsp* → footer e menù principale presenti in tutte le pagine
 - *notify.js*, *notify-box.html* → file utili alla visualizzazione di un pop-up in caso di errori di compilazione di alcuni form
- **java>com.solidisitiweb.com**
 - **model**
 - *Account.java*, *Aderente.java*, *Amministratore.java*, *simpatizzante.java* → "account" è una classe astratta, da cui *aderente*, *simpatizzante* e *amministratore* derivano.
 - *FraseMotivazionale.java* → modello dati della frase che cambia periodicamente
 - *Visita.java* → utile per gestire le visite di ogni pagina
 - *CookieAccettati.java* → *Servlet* invocata se l'utente abilita i cookie che modifica la sessione
 - *DeleteAccount.java* → *Servlet* che elimina a database un utente registrato
 - *DeleteVisit.java* → *Servlet* che resetta i contatori delle visite di ogni pagina (dal database)
 - *Donazioni.java* → *Servlet* per inserire a database le donazioni ricevute
 - *FilterLogged.java* → filtro che controlla i permessi per le aree private in base all'utente
 - *FilterVisited.java* → filtro per ogni pagina che incrementa a database il numero di visite
 - *GetDati.java* → *Servlet* che ritorna i dati dell'utente che li richiede
 - *GetDonazioni.java* → *Servlet* che ottiene le donazioni mensili da database
 - *GetFrase.java* → *Servlet* che ritorna una frase motivazionale
 - *GetVisite.java* → *Servlet* che ottiene le visite per ogni pagina da database
 - *LoginUtente.java* → *Servlet* che gestisce il login
 - *Logout.java* → *Servlet* che "slogga" l'utente dalla sessione
 - *RegistraUtente.java* → *Servlet* per inserire l'utente a database
 - *SendEmail.java* → *Servlet* per inviare una mail fittizia
 - *SetAttivit .java* → *Servlet* per gestire l'iscrizione alle attivit  dei vari utenti
 - *UtentiAderenti.java*, *UtentiSimpatizzanti.java*, *UtentiRegistrati.java* → *Servlet* utili all'amministratore per vedere le varie tipologie di utenti registrati

Suddivisione compiti: fermo restando che ogni componente ha partecipato allo sviluppo di ogni parte del progetto, le aree in cui ognuno si è concentrato maggiormente sono:

- **Minatel:** gestione visite, gestione frasi motivazionali, realizzazione di alcune pagine pubbliche, file annex
- **Cappellato:** creazione database e uso libreria Gson, cancellazione utente, invio mail, lista utenti
- **Zini:** realizzazione dei fogli di stile, libreria highcharts.com, realizzazione pagine, navbar, footer, banner cookie
- **Zanibellato:** login, sign in, filtro accessi, validazione campi form, gestione attivit 

Il progetto è stato sviluppato in tutte le sue funzionalità richieste, seguendo le indicazioni ulteriori ricevute via e-mail.