Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAMT |
| Data | 06.11.2018 |

|  |
| --- |
| Lavori svolti |
| Questa mattina inizialmente ho controllato il funzionamento della pagina di richiesta dei corsi da parte degli allievi, e ho ottimizzato il codice in modo da correggere eventuali problemi e rendere più leggibile e “snello” il codice in sé.    Questa è la parte che è cambiata del codice, la parte finale.  Se si paragona alla versione precedente si può notare che è molto più piccola ed è stato aggiunto un controllo. Il controllo aggiunto è quello dello spazio libero nel corso, che viene controllato con una variabile booleana, se non c’è spazio viene portata a “false” in caso contrario registra l’utente e smette di controllare. Alla fine del ciclo controlla de nessun corso era libero ne crea uno e inserisce l’utente in una funzione esterna, come fa anche se non esistono corsi come quello richiesto.  Durante i controlli mi sono accorto di un errore nei controlli degli input della pagina, spiegati nella sezione “problemi riscontrati e soluzioni adottate.  Una volta finito questo problema ho messo a posto la pagina del docente di creazione dei corsi, in modo che al cambio della materia vengano mostrati a fianco le categorie annesse, come nella pagina di ricerca dei corsi dell’allievo. Nel farlo però ho riscontrato un problema, spiegato ella parte seguente.  Dopodiché ho iniziato a lavorare alle pagine dell’amministratore.  Inizialmente ho creato l’instestazione molto simile a quelli degli altri utenti ma con differenze nei nomi delle pagine, poi ho creato la classe model e il caso nel login per accedere alla pagina principale, quella dei coach, al login, e le funzioni nel “model” che fanno accedere alle relative pagine.    Questa è la funzione che va ad aprire le pagine per mostrare la lista dei coach.    Questa è la funzione che apre le pagine per mostrare le informazioni degli allievi.    Questa è la funzione che apre tutto il necessario per mostrare la pagina di tutti i corsi.  Una volta finito con la creazione dell’apertura delle pagine ho creato la grafica delle pagine, essendo che devono essere tutte molto simili.    Questa è la prima pagina che appare al login dell’amministratore, la pagina che mostra tutti i coach e permette di accederci e modificarla, l’intestazione delle pagine è la stessa per tutte l’unica differenza è il titolo che varia a dipendenza della pagina aperta.    Questa è la pagina che mostra tutti gli allievi registrati e da la possibilità d accedervi per modificare le impostazioni o aggiungere corsi.    Questa è la pagina che mostra tutti i corsi esistenti.    Questa è la funzione presente nel file javascript che viene richiamata dalle pagine “corsi” e “coach” dopo aver creato tutto il resto, si occupa di, con l’ausilio di ajax, fare una richiesta http alla funzione che prende tutti gli utenti e le relative informazioni, in base al tipo che gli viene passato tramite il POST.    Questa funzione viene richiamata dal POST nella funziona JavaScript mostrata sopra, e si occupa semplicemente di controllare chi ci accede e prendere i dati degli utenti che prende dal database la funzione in “Connection” codificarli in JSON e stamparli, in modo che Ajax riesca ad interpretarli come visto precedentemente.    Nella classe connection per prendere gli utenti ho semplicemente modificato la già esistente funzione “getUser” in modo che controllasse se viene passato un tipo, in caso positivo fa la “SELECT” in base ad esso invece che in base alla email.    Questa funzione di JavaScript fa la stessa cosa dell’altra ma prende i corsi e imposta in modo diverso i dati che inserisce all’interno del select, perché ne riceve altri, e in più non manda niente quando fa il “send()”.    Questa è la funzione della model dell’amministratore che si occupa di prendere i dati dalla funzione nella classe “Connection” di tutti i corsi esistenti.    Questa funzione è quella che si occupa di prendere tutti i dati di tutti i corsi e inserirli all’interno di un array per poi ritornarli. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Un primo problema che ho riscontrato è stato nella pagina della richiesta dei corsi, perché nel controllo dell’input numerico non veniva controllato che non c’erano scritte lettere, come si poteva fare in Edge, questo perché il nome della funzione dichiarata al click di un bottone all’interno dell’input e il nome della funzione effettiva erano diversi, per mettere a posto questo mi è bastato scriverle uguali.  Mentre mettevo la visualizzazione delle categorie all’interno della pagina di creazione dei corsi ho incontrato un errore, dato dai controlli che facevo, in modo che poteva accedere alla funzione per prendere le categorie solo l’allievo, mentre dovrebbero poterci accedere tutti, quindi una volta tolto quel controllo tutto è tornato a posto.    Questa è la funzione attuale che prende le categorie dalla funzione che si collega al database, senza controlli.  Ho avuto alcuni problemi con JSON, quando dovevo parsare ciò che mi ritornava il server in ajax, perché non dovevo fare lo stesso passaggio dell’altro ma uno in meno essendo che passavo già un array, e mi bastava parsarlo e non ritrasformarlo in un array. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono in linea con il gantt, ho fatto alcune cose oggi che avrei dovuto iniziare più avanti ma ho spostato altre cose per le prossime lezioni. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Iniziare l’implementazione della connessione ai coach dell’amministratore. |