Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Scuola d’arti mestieri Trevano |
| Data | 10.09.2020 |

|  |
| --- |
| Lavori svolti |
| Oggi ho iniziato ad implementare la struttura MVC dell’applicazione web. Per realizzare questa struttura ho utilizzato un template MVC che utilizzavo nel modulo 133 (Realizzare delle pagine interattive con php). La struttura cartelle dell’applicativo:  application: contiene tutte le cartelle per realizzare l’applicazione   * config: contiene il file config.php che permette di configurare URL della pagina web * controller: contiene i controllers che permettono di comunicare le views e i models (contiene tutte le azioni). * libs: contiene il file application.php che contiene tutte le classi dell’applicazione web. In questo file viene diviso le parti dell’URL in base ai controller, azioni e 3 parametri o delle azioni nella barra dell’URL. * models: contiene tutti i models (contengono tutte le funzioni che servono alle views) * views: contiene tutte le views (parte grafica)   Per questo progetto utilizzo una programmazione ad oggetti.  Nella cartella config ho creato il file config.php che contiene la configurazione del URL del progetto.  Per creare il percorso URL dell’applicativo utilizzo un define invece di una costante. Questo perché la variabile URL la utilizzo in tutta la struttura MVC e define viene generalmente utilizzato in un ambito globale. Ci sono anche dei vantaggi di scosti, la const viene definita durante la compilazione, invece define durante l’esecuzione dell’applicazione.  Esempio:    Nella cartella libs ho creato il file application.php che contiene le classi dell’applicazione web.  Nel URL possono essere aggiunti dei parametri, azioni o controller manualmente dall’utente o cliccando dei link di navigazione tra le pagine. Quest’ultimi corrispondono alle varie cartelle views (sono le diverse pagine web), funzioni e parametri. Se viene aggiunto manualmente dall’utente bisogna mantenere sempre la struttura corretta dell’URL (http:://localhost:porta/nome\_cartella\_progetto/controller/param1/action). Per mantenere questa struttura devo dividere in più parti l’URL e fare anche dei controlli (se l’utente inserisce qualcosa di sbagliato). Per realizzare questo ho creato il metodo splitURL che contiene il seguente codice:    Nel costruttore della classe Application creo l’URL in base ai parametri, controller e azioni.  In caso che non mi trovo in un’altra pagina devo impostare una pagina di partenza e quindi ho implementato un controller home che contiene un index che corrisponde alla views home.  Immagine che contiene coltello  Descrizione generata automaticamente  Nella cartella controller ho creato il file home.php che implementa la classe Home quest’ultimo corrisponde alla classe della pagina iniziale. Ho creato una funzione index che implementa la views della pagina home.  Immagine che contiene uccello  Descrizione generata automaticamente  Nella cartella models ho creato un’altra cartella home (views home) che contiene un file index.php nel quale viene implementato in HTML, CSS e php la pagina web. Per potere avviare l’applicazione devo creare nella cartella del progetto il file index.php che carica il file di configurazione e il file delle classi dell’applicazione.  Immagine che contiene tavolo  Descrizione generata automaticamente  Per verificare che tutto funziona ho aperto un browser e ho scritto nella barra di ricerca il seguente URL: http://localhost:8080/gestione\_backup  La finestra che deve comparire è la seguente:  Immagine che contiene screenshot  Descrizione generata automaticamente  Se clicco sul link Home mi ritorna la stessa pagina e nella barra di ricerca che c’è scritto il seguente URL:    Se clicco il link Login mi richiama sempre la stessa pagina perché non esiste la views login e nella barra di ricerca c’è scritto il seguente URL: |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Non ho riscontrato dei problemi perché per realizzare la struttura MVC ho utilizzato un template che utilizzavo già nel modulo 133 (Realizzare pagine web interattive). |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono ad un buon punto della pianificazione perché oggi dovevo iniziare ad implementare già qualcosa. Per ora non ho iniziato con script dei backup perché mi sto ancora documentando, quindi ho iniziato ad implementare il login. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Per la prossima giornata devo implementare il CSS del login e provo anche già a creare un utente amministratore con MySQl in modo da creare una sessione dell’utente. |