Diario di lavoro

|  |  |  |  |
| --- | --- | --- | --- |
| Luogo | Trevano | | |
| Data | 15.05.2023 | | |
| Mattino | 09:05 – 12:20 | Pomeriggio | 13:15 – 16:30 |

|  |
| --- |
| Lavori svolti |
| 09:05 – 10:00 🡪 Implementazione pagina inserimento dati generali  10:00 – 11:05 🡪 Implementazione pagina dati dell’aula e metodo per inserimento date  11:05 – 11:10 🡪 Diario  11:10 – 12:00 🡪 Salvataggio dei dati dell’aula con le disponibilità  12:00 – 12:05 🡪 Diario  12:05 – 12:20 🡪 Commenti codice scritto  13:15 – 14:10 🡪 Correzioni stile in salvataggio dati aula e salvataggio dati classe  14:10 – 14:15 🡪 Diario  14:15 – 14:35 🡪 Salvataggio dati docente e correzione per $conection  14:35 – 14:50 🡪 Salvataggio dati docenti  15:00 – 15:50 🡪 Salvataggio dati studente (drop down)  15:50 – 16:00 🡪 Diario  16:00 – 16:05 🡪 Salvataggio dati studente  16:05 – 16:15 🡪 Correzione ER e SQL  16:15 – 16:25 🡪 Revisione codice con commenti e todo  16:25 – 16:30 🡪 Diario e chiusura lavori |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Sono riuscito finalmente a gestire correttamente l’inserimento delle date di inizio e fine nel DB. L’ho corretto grazie ad una ricerca su Stackoverflow e questo codice:  $startDate = DateTime::createFromFormat('m/d/Y', $startDate);  $endDate = DateTime::createFromFormat('m/d/Y', $endDate);  $startDate = $startDate->format('Y-m-d');  $endDate = $endDate->format('Y-m-d');  fatto questo sono passato alla gestione delle aule. Anche qui ho dovuto trovare un metodo, aiutandomi online, per convertire gli intervalli orari dati da JS in oggetti *datetime* di PHP. L’ho fatto trovando e modificando un metodo che mi ritornasse, dato un intervallo orario e un giorno della settimana, tutti i giorni corrispondenti in quell’intervallo. In questo modo posso ottenere, per esempio, tutti i lunedì nell’intervallo specificato. [Link alla pagina di stackoverflow](https://stackoverflow.com/questions/7061802/php-function-for-get-all-mondays-within-date-range)  Nel salvare le aule nel DB non ho avuto particolari problemi, avvalendomi molto della classe “error\_log” ho loggato piano piano tutto arrivando la soluzione.  I passi sono: tradurre il formato “1.08:20 – 09:05” in “19.03.2023 – 09:05”, dove 19.03.2023 è il primo lunedì della pianificazione. Con il metodo citato sopra posso trovare questo formato per tutte le date e per tutti i giorni. Infine posso creare l’array di dati da inserire nel DB [startDateTime,endDateTime,aulaId].  Ho poi fatto la stessa cosa per i dati della classe, cercando di generalizzare il più possibile inserendo un altro metodo nella classe Model. Sto cercando di rendere i controller il più simile possibile. Ho anche cambiato tutti i *requrie* e *require\_once*, dato che dava un errore di dichiarazione. Ho anche inserito il parametro di connessione $connection come attributo di ogni model, e lo imposto solo nel costruttore.  Per lo studente ho incontrato come prima cosa il problema di gestire la visualizzazione tramite *dropdown* dei dati di classi e docenti. Ho dovuto girare online e per un passaggio, quello della comunicazione generale, chiedere a chatGPT. Sono però riuscito a finire e capire il codice.  VIEW   require\_once 'application/controller/StudentController.php';  $studController = new StudentController();  $select = $studController->populateSlectClass();  echo $select;  ?>  In questa parte di codice prima definisco il controller di studente, in modo da poter eseguire la funzione PopulateSelectClass e ottenere il risultato nella varabile select. Che poi stampo.  CONTROLLER  $selectData = $model->getData("nome,cognome", "docente");          $select = '<select name="option" class="form-select" aria-label="Default select example"> <option selected>Seleziona un docente</option>';  foreach ($selectData as $el) {  $select .= '<option value="' . $el['cognome'] . '">' . $el['nome'] . " " . $el["cognome"] . '</option>';  }  $select .= '</select>';  Nel controller, come prima cosa prendo i dati con il metodo nel Model.  Poi tramite il foreach ed accedendo alle chiavi (che poi sono i campi di mysql) creo tutte le opzioni. Alla fine del ciclo chiudo il tag select.  MODEL  $sql = "SELECT " . $field . " FROM " . $table;  $query = $this->connection->prepare($sql);  $query->execute();  $result = $query->fetchAll(PDO::FETCH\_ASSOC);  return $result;  nel model prendo i dati da sql come ho fatto in altri punti, li salvo in un array e li ritorno al contoller.  Nel salvare i dati dello studente mi sono accorto che nel DB la chiave “progettoId” è al posto sbagliato. Io l’ho messa su allievo, ma invece è la chiave dell’allievo che va su progetto. Guaradno la mia progettazione originale l’avevo pensata giusta, è quindi una svista imputabile al momento in cui ho convertito la relazione tra allievo e progetto da n-n a 1-n. Ho sistemato il tutto |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| In ritardo. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Finire pagina salvataggio studenti  Inizio motore di pianificazione |