Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 23.10.2018 |

|  |
| --- |
| Lavori svolti |
| Questa giornata di progetti l’ho passata per cercare di fare funzionare la parte di invio delle “informazioni” dei checkbox, presenti nella tabella stampata sulla pagina web principale, ad un secondo file. Le informazioni che non riuscivo a passare erano il nome ed il cognome dell’alunno su cui si era cliccato il checkbox. Come primo passo, come già citato nel diario precedente, essendo una tabella dinamica sono andato ad impostare come valore dell’input il nome concatenato al cognome dell’alunno in considerazione tramite php:  value="<?php echo $blockedResult[$i]['Nome']. "\_" . $blockedResult[$i]['Cognome']?>"  Una volta fatto ciò, ho sistemato le parti “ridondanti” di codice php presenti nel file principale andando a creare un blocco php unico all’inizio del file dove gestisco, in base a quali dati l’utente immette/sceglie, quale file esterno andare a richiamare:  <?php  $blockedResult = null;  $unBlockedResult = null;  $change = null;  //Eseguo il require dei file esterni necessari.  require\_once('php/db\_connection.php');  require\_once('php/data\_change.php');  //Creao gli oggetti delle classi necessarie;  $conn = new connection();  $gestions = new gestions();  //Controllo se é stato eseguito il POST  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  //Eseguo il controllo su quale bottone submit viene premuto, in base ai dati con cui l'amministratore lavora.  if (isset($\_POST["anno"]) && isset($\_POST["id"])){  //Richiamo la funzione desiderata passandogli i parametri che necessita.  $blockedResult = $conn->getClassBlocked($\_POST["anno"], $\_POST["id"]);  $unBlockedResult = $conn->getClassUnblocked($\_POST["anno"], $\_POST["id"]);  }  if (isset($\_POST["start"]) && isset($\_POST["end"])) {  if (isset($\_POST["studentBlocked"])) {  $change = $gestions->getCheckboxState($\_POST["studentBlocked"]);  }  if (isset($\_POST["YouTubeBlocked"])) {  $change = $gestions->getCheckboxYouTube($\_POST["YouTubeBlocked"]);  }  /\*\*var\_dump($\_POST["studentBlocked"]);  echo "<br>";  var\_dump($\_POST["YouTubeBlocked"]);\*/  }  else {  if (isset($\_POST["studentBlocked"])) {  $change = $gestions->getCheckboxState($\_POST["studentBlocked"]);  }  if (isset($\_POST["YouTubeBlocked"])) {  $change = $gestions->getCheckboxYouTube($\_POST["YouTubeBlocked"]);  }  }  }  ?>  Facendo in questo modo, però, ho dovuto creare due variabili (*$blockedResult* e *$unBlockedResult)* che andassero a sostituire la variabile precedente *$result.*  Dopo questo era sorto un problema, non riuscivo ad inviare i dati. Per controllare che tipo di dati dovesse gestire la mia funzione nel file esterno ho semplicemente stampato a schermo cosa “ritornasse” il metodo *$\_POST* tramite un var\_dump(), una volta visto che tipo di dato dovessi gestire (array) ho dovuto inserire nel *name* dell’input le parentesi quadre alla fine del name in modo tale da “fare capire” che stessi gestendo un array. Non mi restava altro che creare le mie funzioni di ricezione dati nel file esterno.  function getCheckboxState($state){  $resState = array();  if(count($state) != 0){  for ($i=0; $i < count($state); $i++) {  $stateId = explode("\_", $state[$i]);  array\_push($resState, $stateId);  }  }else{  echo "Non funziona";  }  echo "<pre>";  echo "Stato";  print\_r($resState);  echo "</pre>";  }  La funzione soprastante riceve la variabile stato contenente il nome ed il cognome dell’allievo scelto, la divide e la inserisce in un array multidimensionale. Subito dopo questo ne ho creata un’altra simile che gestisce la ricezione dei checkbox di YouTube. Come test facevo stampare a schermo il nuovo array che conteneva al primo posto il nome mentre al secondo il cognome.  In seguito ho implementato, tramite javascript e php una funzione che permettesse il “salvataggio” nel menu a tendina iniziale l’anno e l’id scelto, in moto tale da non dovere ogni volta andare a scegliere la classe.  <input type="hidden" name="anno" value="<?php echo $\_POST["anno"] ?>">  <input type="hidden" name="id" value="<?php echo $\_POST["id"] ?>">  Inizialmente ho creato questi due input nascosti nel secondo form in modo tale da poter tenere salvata la classe scelta anche quando clicco I submit degli altri form.  var year = null;  var id = null;  function getData(){  year = "<?php echo $\_POST["anno"]; ?>";  id = "<?php echo $\_POST["id"]; ?>";  document.getElementById("id").value = id;  document.getElementById("year").value = year;  }  Tramite il seguente script leggo la classe scelta inizialmente e, tramite un evento onload presente nel tag body, la “slavo” in moto tale che una volta che la pagina si è ricaricata vado a riscrivere la classe scelta. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| In questa lezione ho riscontrato multi problemi sull’invio dei dati da un file all’altro anche se alla fine della giornata sono riuscito a risolvere tutti i problemi, anche se con l’aiuto di alcuni compagni. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Un po’ in ritardo rispetto alla pianificazione. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Eseguire la scrittura e la stampa a schermo dei link nella pagina delle white-list.  Continuare con l’interazione tra l’index e il database. |