Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Canobbio |
| Data | 26.11.2019 |

|  |
| --- |
| Lavori svolti |
| 13h15 – 14h00  Corretto errori di ortografia presenti nella documentazione  14h00 – 14h20  Corretto formattazione della documentazione mettendo il tutto in tipo di testo giustificato.  14h20 – 14h45  Approfondito la documentazione e i commenti del codice.  15h00 – 16h20  Ho eseguito una pulizia del codice risolvendo piccoli bug documentati nella sezione problemi riscontrati.  Eseguito le seguenti modifiche:   * Rinominato i pulsanti delle pagine * Pulsante per tornare indietro da un articolo   Il pulsante per tornare alla pagina precedente è stato implementato nel seguente modo:  <button type="button" class="btn btn-secondary" onclick="history.go(-1);">Torna indietro</button>  Quando il pulsante viene premuto l’utente verrà rimandato di una pagina indietro rispetto alla sua cronologia.  16h20 – 16h30  Stesura diario. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Risolto problema di impaginazione, il numero di pagine massimo era calcolato in modo errato. Ho risolto utilizzando la funzione ceil al posto della funzione round per arrotondare il numero della pagina.  In questo modo se per esempio gli articoli sono 10 e il massimo per pagina è 3:  10 / 3 = 3.33 -> 4  4 sono le pagine  Mentre prima era:  10 / 3 = 3.33 -> 3  3 sono le pagina  Il problema di paginazione era presente anche quando un utente eseguiva la ricerca all’interno del sito, ho risolto implementando una funzione che si occupa di ricavare tutti i risultati della ricerca dell’ utente.  */\*\**  *\* Metodo che permette di ricavare la pagina massima di una ricerca.*  *\**  *\* @param String $search La ricerca.*  *\* @return Integer La pagina massima.*  *\*/*  public static function getSearchMaxPage($search)  {  $search = '%'.$search.'%';  $query = Database::get()->prepare("SELECT COUNT(\*) FROM articles WHERE title LIKE :title");  $query->bindParam(":title", $search, PDO::PARAM\_STR);  $query->execute();  $maxPage = ceil($query->fetch(PDO::FETCH\_NUM)[0] / self::$limit) - 1;  if ($maxPage < 0) {  $maxPage = 0;  }  return $maxPage;  } |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Mi trovo molto avanti rispetto alla pianifica. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| - |