Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Trevano |
| Data | 17.02.2020 |

|  |
| --- |
| Lavori svolti |
| Ho continuato la parte delle notifiche, utilizzando ajax ho creato vado a richiedere le ultime tre notifiche ad un metodo del controller notification.php che ritorna il tutto in formato json.  Una volta ricevuto il json aggiungo dei div contenenti le informazioni delle notifiche e, dopo aver calcolato il tempo, viene visualizzato a schermo la data della notifica (dopo le 23 ore viene mostrata la data e l’ora, dopo 59 minuti vengono mostrate le ore).    Premendo sul link “Visualizza tutte le notifiche” si viene indirizzati ad una pagina in cui sono mostrate tutte le proprie notifiche.  Durante lo sviluppo della pagina di notifiche mi sono accorto che dalla tabella di log non ho nessun riferimento agli ordini, quindi non posso recuperare le informazioni di quest’ultimi. Ho dovuto effettuare molte modifiche siccome nella tabella ordine non avevo un id ma utilizzavo 3 campi come primary key. |
| Per evitare di utilizzare 3 campi anche nelle altre tabelle come chiave primaria ho dovuto aggiunger un id e modificare le parti di codice dove utilizzavo gli altri 3 campi.  Questa parte di modifica del codice mi ha fatto perdere un bel po’ di tempo. |
| Ho deciso di mostrare tutte le notifiche dell’utente nell’apposita pagina. Ho utilizzato un modal per visualizzare i dettagli dell’ordine che è stato rifiutato/accettato. |
| Una volta fatto ciò mi sono dedicato al report del magazzino (pagina in cui è possibile monitorare la situazione del magazzino). Ho creato una pagina simile alle altre in cui vengono mostrate tutti i record preseti nella tabella di log.  Successivamente ho creato una stored procedura nella banca dati che controlla, tramite un evento, una volta al giorno se stanno per scadere articoli. L’idea è che se l’articolo è scaduto viene settato il campo eliminato della tabella articoli a 1, viene inserito nella tabella di log e tutti gli ordini per quell’articolo vengono rifiutati. Inoltre gli articoli che stanno per scadere vengono aggiunti alla tabella di log in modo da poter essere visualizzati dagli admin e gli operatori.  Ho utilizzato il campo facoltativo utente\_richiedente della tabella log per controllare se è un campo che può essere eliminato o meno. Se non è settato significa che il record non è riferito ad una richiesta d’ordine e nemmeno ad un articolo che è stato eliminato. Così ogni giorno i record che non hanno questo campo settato vengono eliminati automaticamente. Se un articolo è scaduto non viene eliminato il record dal file di log in modo da poter sempre visualizzare gli articoli scaduti.  Procedura che elimina i record degli articoli che stanno per scadere:   |  | | --- | | DELIMITER //  CREATE PROCEDURE deleteOldRecords()  BEGIN  DELETE FROM LOG WHERE utente\_richiedente IS NULL AND id\_tipo\_operazione != 4;  END //  DELIMITER ; |   Procedura che controlla ed inserisci gli articoli scaduti/in scadenza:   |  | | --- | | DELIMITER //  CREATE PROCEDURE checkExpiredArticles()  BEGIN  DECLARE finished int DEFAULT 0;  DECLARE article\_id int;  DECLARE expire\_date date;    -- declare cursor article  DECLARE curArticle  CURSOR FOR  SELECT id, data\_scadenza FROM articoli WHERE eliminato = 0 AND in\_magazzino = 1;    -- declare NOT FOUND handler  DECLARE CONTINUE HANDLER  FOR NOT FOUND SET finished = 1;    -- delete old record  call deleteOldRecords();    OPEN curArticle;    getArticleId: LOOP  FETCH curArticle INTO article\_id, expire\_date;  IF finished = 1 THEN  LEAVE getArticleId;    -- article expired  ELSEIF expire\_date <= date(now()) THEN    -- insert into log  INSERT INTO LOG(id\_articolo, id\_tipo\_operazione) VALUES (article\_id, 4);    -- delete article from storage  UPDATE articoli SET eliminato = 1 WHERE id = article\_id;    -- article almost expired  ELSEIF expire\_date <= date(now() + INTERVAL 2 day) THEN    -- insert into log  INSERT INTO LOG(id\_articolo, id\_tipo\_operazione) VALUES (article\_id, 3);  END IF;  END LOOP getArticleId;  CLOSE curArticle;    END //  DELIMITER ; |   Evento che viene chiamato una volta al giorno:   |  | | --- | | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Automatic event \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  SET GLOBAL event\_scheduler = ON;  CREATE EVENT daily\_articles\_check  ON SCHEDULE EVERY 1 day  DO  call checkExpiredArticles(); |   Link sintassi stored procedure: <https://www.quackit.com/mysql/tutorial/mysql_stored_procedures.cfm>  Link cursori: <https://www.mysqltutorial.org/mysql-cursor/> |

|  |
| --- |
| Una volta fatta questa parte ho mostrato a schermo, nella pagina di report, la situazione del magazzino: |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| - |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| In anticipo rispetto la pianificazione. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| La prossima lezione devo sviluppare la pagina in cui ogni utente base (cliente) può visualizzare la cronologia delle sue richieste d’ordine. |