

Diario di lavoro

Luogo	Mendrisio
Data	18.05.2020

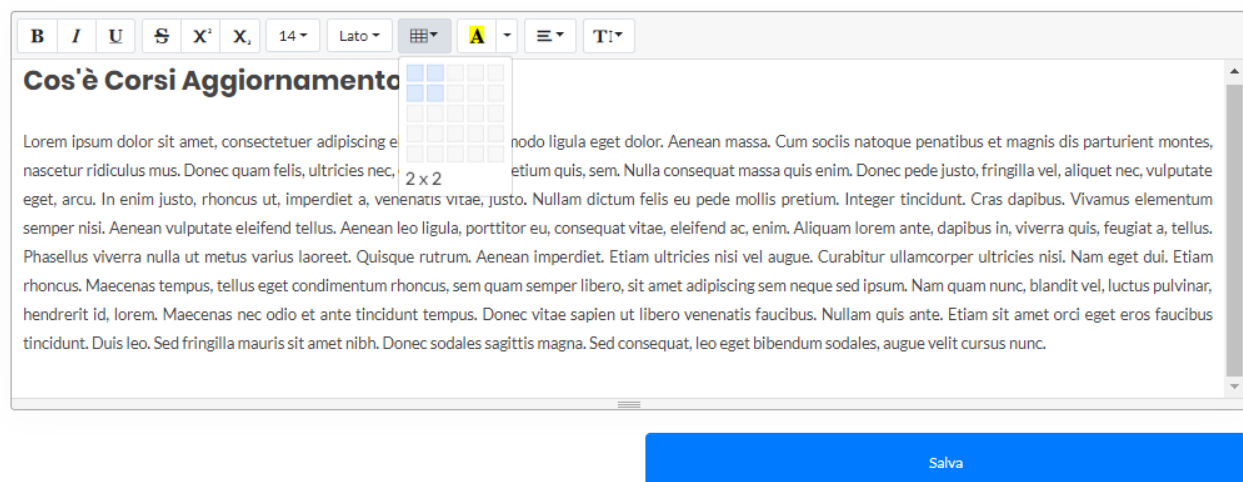
Lavori svolti

Durante la giornata di oggi ho concluso le funzionalità di gestione del sito. Più precisamente ho aggiunto la possibilità di creare delle tabelle nell'editor html per la modifica delle pagine e ho aggiunto la possibilità di modificare le impostazioni della pagina.

Per aggiungere la creazione delle tabelle ho direttamente cambiato libreria. Questa scelta è stata fatta in quanto la libreria trumbowyg, quando si andavano ad aggiungere dei plugin extra, rimuoveva tutti quelli di base. La libreria che ho scelto si chiama summernote (<https://summernote.org/>).

L'editor si presenta così:

Descrizione



Una volta fatto ciò ho aggiunto una tabella per mostrare tutti gli utenti registrati al sito. Per avere una tabella organizzata e responsive ho utilizzato la libreria datatables (<https://datatables.net/>).

La tabella si mostra così:

Utenti registrati

Mostra 10 utenti per pagina

Cerca:

NEWSLETTER	↑↓	NOME	↑↓	COGNOME	↑↓	EMAIL	↑↓	DATA DI NASCITA	↑↓	VIA	↑↓	CITTÀ	↑↓	CAP	↑↓
	✓	Mattia		Toscanelli		gestionecorsi@yopmail.com		2001-03-25		Via rime 10		Mendrisio		6850	
	✓	Mattia		Toscanelli		gestionecorsi3@yopmail.com		2001-03-25		Via rime 10		Mendrisio		6850	
	✗	Mattia		Toscanelli		gestionecorsi2@yopmail.com		2001-03-25		Via rime 10		Mendrisio		6850	
	✗	Mattia		Toscanelli		gestionecorsi4@yopmail.com		2001-03-25		Via rime 10		Mendrisio		6850	

Pagina 1 di 1

Precedente **1** Successivo

Infine ho creato la gestione delle impostazioni della pagina principale. In questa parte di pagina si possono modificare i dati di pagamento, modificare i giorni rimanenti per iscriversi ad un corso prima del suo inizio e aggiungere/rimuovere tipologie di corsi.

The screenshot shows the 'Corsi Aggiornamento' admin interface. It has a top navigation bar with links: Home, Pannello Admin, Contatti, Ricarica Corsi, and Login. The main content area is divided into three sections:

- Pagamento:** Contains a 'Modifica pagamento / CF' button and a table with two columns: TIPO and VALORE. The table has three rows: IBAN (CH8901244907444344457), Banca (UBS), and Beneficiario (Mattia Toscanelli).
- Giorni scadenza corso:** Contains a 'Modifica giorni / CF' button and a table with two columns: TIPO and VALORE. The table has one row: Giorni scadenza (3).
- Tipologie:** Contains an 'Aggiungi tipologia +>' button and a table with two columns: TIPOLOGIA and AZIONI. The table has three rows: Auto, Canion, and Moto. Each row has a red 'Elimina' button with a trash icon.

La gestione dei dati di pagamento e dei giorni di scadenza è molto simile. Il metodo per modificare i dati di pagamento è il seguente:

```
public function modifyPayment($iban, $bank, $beneficiary)
{
    $data = array();
    $checkIBAN = $checkBank = $checkBeneficiary = $checkAll = SUCCESSFUL;
    if(!$this->validator->checkIBAN($iban)){
        $checkIBAN = ERROR;
        $checkAll = ERROR;
    }
    if(!$this->validator->checkNameNumber($bank)){
        $checkBank = ERROR;
        $checkAll = ERROR;
    }
    if(!$this->validator->checkName($beneficiary)){
        $checkBeneficiary = ERROR;
        $checkAll = ERROR;
    }
    if($checkAll == SUCCESSFUL){
        $updateSettings = "UPDATE settings SET iban_number=%s, bank=%s,
beneficiary=%s";
        $this->connection->query($updateSettings,$iban,$bank,$beneficiary,$iban);
    }
    $data[] = array(
        SATUTS => $checkAll,
        CHECK_IBAN => $checkIBAN,
        CHECK_BANK => $checkBank,
        CHECK_BENEFICIARY => $checkBeneficiary
    );
    echo json_encode($data);
}
```

Questo metodo si occupa di verificare tutti i dati passati tramite il form e ritornare un json che contiene lo stato dell'operazione (fallito/bocciato) e i vari risultati dei controlli degli input. Grazie a questo json posso mostrare all'utente delle notifiche di successo/errore. Lo stesso procedimento è stato fatto per la gestione dei giorni di scadenza, ma naturalmente con un solo dato da controllare.

Invece per quanto riguarda la gestione delle tipologie ho implementato l'aggiunta e l'eliminazione. Per aggiungere una tipologia utilizzo questo metodo:

```
function addTypology($typology){
    if($this->validator->checkNameNumber($typology)){
        $selectTypology = "SELECT * FROM typology WHERE name=%s";
        $typologies = $this->connection->query($selectTypology,$typology);
        if(count($typologies) == 0) {
            $insertTypology = "INSERT INTO typology (name) VALUES (%s)";
            $this->connection->query($insertTypology,$typology);
            echo SUCCESSFUL;
        }else{
            echo ERR_DUP;
        }
    }else{
        echo ERROR;
    }
}
```

Esso verifica che la tipologia non sia già stata aggiunta in precedenza e che abbia un nome valido. In caso che entrambi i controlli vadano a buon fine la tipologia viene aggiunta al database e la tabella html viene aggiornata tramite ajax.

Per rimuovere una tipologia utilizzo questo metodo:

```
public function delTypology($name)
{
    $selectTypology = "SELECT * FROM typology WHERE name=%s";
    $typologies = $this->connection->query($selectTypology, $name);
    if(count($typologies)>0){
        $deleteTypology= "DELETE FROM typology WHERE name=%s";
        $this->connection->query($deleteTypology, $name);
        echo SUCCESSFUL;
    } else {
        echo ERROR;
    }
}
```

Esso verifica che la tipologia sia presente nel database e nel caso la elimina.

Infine ho documentato tutta la parte di login, recupero password, reimposto password e gestione della pagina principale.

Problemi riscontrati e soluzioni adottate

Durante la lezione di oggi non ho avuto particolari problemi.

Punto della situazione rispetto alla pianificazione

Sono in linea rispetto la mia pianificazione in quanto oggi avrei dovuto concludere la gestione delle impostazioni dell'applicazione.

Programma di massima per la prossima giornata di lavoro

La prossima giornata di lavoro farò l'aggiunta dei corsi di aggiornamento e de resta tempo la modifica.