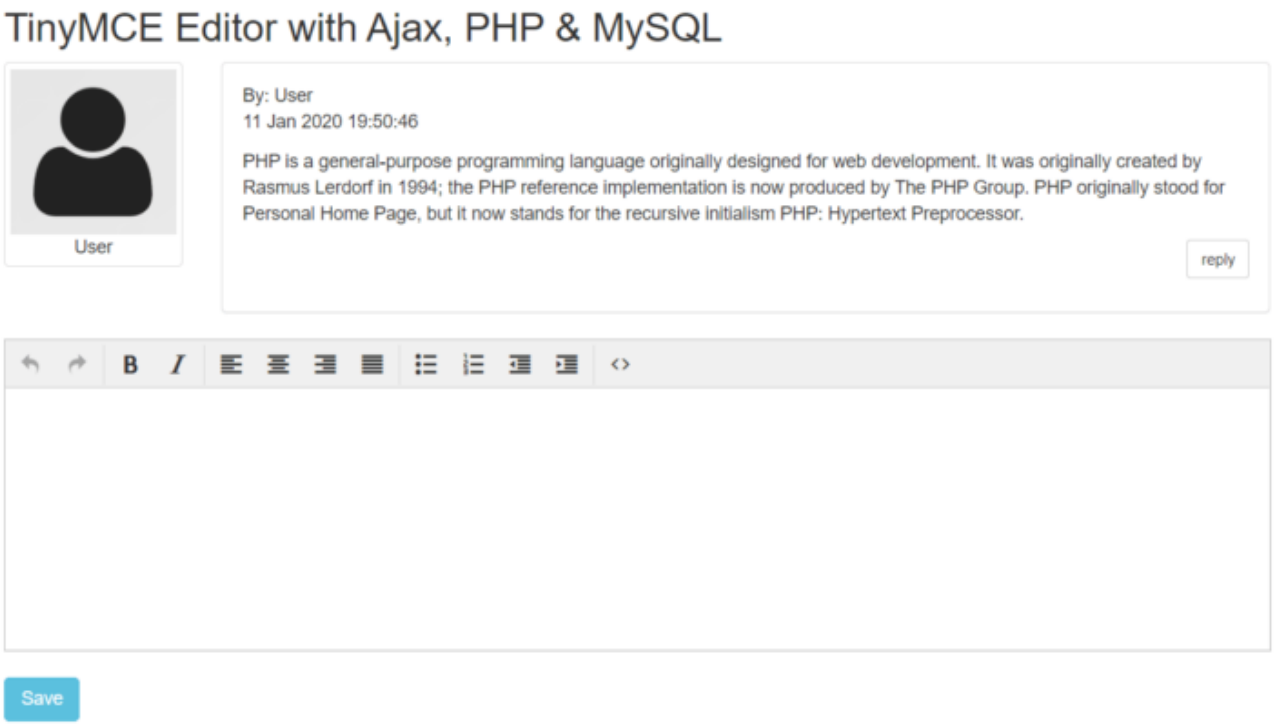
Сохранение содержимого редактора TinyMCE с помощью PHP и MySQL

В предыдущем уроке мы объяснили, как разработать систему учета посещаемости студентов с использованием PHP и MySQL . В этом уроке мы расскажем, как использовать редактор TinyMCE для замены текстового поля и сохранения контента с помощью Ajax, PHP и MySQL.

TinyMCE — это самый продвинутый JavaScript WYSIWYG HTML-редактор для создания контента на веб-сайтах. Редактор легко интегрируется в любой веб-сайт, преобразуя текстовые поля в HTML-редактор для создания HTML-контента.

Если вы работаете над сайтом и хотите заменить обычное текстовое поле на продвинутый WYSIWYG HTML-редактор, то вы попали по адресу.

В этом пошаговом руководстве мы рассмотрим, как интегрировать редактор TinyMCE и сохранять контент в базу данных MySQL с помощью Ajax и PHP.



Основные файлы:

- **index.php** : PHP-файл для загрузки редактора TinyMCE для сохранения контента.
- **tinymce_editor.js** : JavaScript-код для инициализации редактора TinyMCE.
- **action.php** : Файл действий для обработки AJAX-запросов.
- **Post.php** : Класс для хранения методов.

Шаг 1: Создание таблицы в базе данных MySQL

Сначала мы создадим таблицу в базе данных MySQL `posts` для хранения содержимого,

полученного из редактора TinyMCE.

MySQL

```
CREATE TABLE `posts` (
  `id` int(11) NOT NULL,
  `message` text NOT NULL,
  `user` varchar(255) NOT NULL,
  `edited` int(11) NOT NULL DEFAULT 0,
  `created` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Шаг 2: Подключите файлы Bootstrap, jQuery и TinyMCE Editor.

Поскольку мы будем создавать страницы с использованием Bootstrap, мы включим файлы Bootstrap. Также мы включим файлы jQuery и TinyMCE в `index.php` файл.

index.php

```
<link rel = "stylesheet" href = <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min
.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.j
s"></script>
<script src="tinymce/tinymce.min.js"></script>
<script src="js/tinymce_editor.js"></script>
```

Шаг 3: Создайте форму с помощью редактора TinyMCE.

В `index.php` файле мы создадим HTML-форму с текстовым полем и кнопкой отправки для сохранения данных формы.

index.php

```
<form id="posts" name="posts" method="post">
    <textarea name="message" id="message"></textarea><br>
    <input type="hidden" name="action" id="action" value="save" />
    <button type="submit" id="save" name="save" class="btn btn-info
```

```
saveButton">Save</button>
</form>
```

Шаг 4: Инициализация редактора TinyMCE.

В `tinymce_editor.js` файле мы инициализируем редактор TinyMCE. Мы будем использовать текстовое поле для преобразования в расширенный редактор TinyMCE.

tinymce_editor.js

```
tinymce.init({
    selector: "textarea",
    plugins: [
        "code ",
        "paste"
    ],
    toolbar: "undo redo | bold italic | alignleft aligncenter
alignright alignjustify | bullist numlist outdent indent | link code ",
    menubar:false,
    statusbar: false,
    content_style: ".mce-content-body {font-size:15px;font-
family:Arial,sans-serif;}",
    height: 200
});
```

Шаг 5: Обработка отправки формы и сохранение содержимого.

Мы будем обрабатывать отправку формы с помощью jQuery и выполнять JAX-запрос `action.php` с действием `save`для сохранения значений формы в таблицу базы данных и отображения сохраненного контента в случае успешного выполнения JAX-запроса.

action.php

```
$(document).on('submit','#posts', function(event){
    var formData = $(this).serialize();
    $.ajax({
            url: "action.php",
            method: "POST",
            data: formData,
            dataType:"json",
            success: function(data) {
                var html = $("#postHtml").html();
                html = html.replace(/USERNAME/g, data.user);
```

```
                html = html.replace(/POSTDATE/g, data.post_date);
                html = html.replace(/POSTMESSAGE/g, data.message);
                $("#postLsit").append(html).fadeIn('slow');
                tinymce.get('message').setContent('');
            }
    });
    return false;
});
```

В `action.php` файле мы обработаем AJAX POST-запрос и вызовем метод insert()для вставки содержимого в базу данных.

action.php

```
include_once 'config/Database.php';
include_once 'class/Post.php';
$database = new Database();
$db = $database->getConnection();
$post = new Post($db);
if(!empty($_POST['message']) && $_POST['message'] && $_POST['action']
== 'save') {
    $post->message = $_POST['message'];
    $post->user = "User";
    $post->insert();
}
```

`insert()` В классе будет реализован метод `Post.php` для сохранения содержимого в таблицу базы данных MySQL и возврата сохраненных данных в виде JSON-ответа для отображения записи содержимого.

Post.php

```
public function insert(){

    if($this->message) {

        $stmt = $this->conn->prepare("
            INSERT INTO ".$this->postsTable."(`message`, `user`)
            VALUES(?, ?)");

        $stmt->bind_param("ss", $this->message, $this->user);

        if($stmt->execute()){
            $lastPid = $stmt->insert_id;
            $sqlQuery = "
                SELECT id, message, user, DATE_FORMAT(created,'%d %M %Y
%H:%i:%s') AS post_date
                FROM ".$this->postsTable." WHERE id = '$lastPid'";
```

```
            $stmt2 = $this->conn->prepare($sqlQuery);
            $stmt2->execute();
            $result = $stmt2->get_result();
            $record = $result->fetch_assoc();
            echo json_encode($record);
        }
    }
}
```

Шаг 6: Отображение сохраненных записей

Мы отобразим сохраненные записи в `index.php` файле, вызвав соответствующий `getPost()` метод.

index.php

```
<div id="postLsit">
<?php
$result = $posts->getPost();
while ($post = $result->fetch_assoc()) {
    $date = date_create($post['created']);
?>
    <article class="row">
        <div class="col-md-2 col-sm-2 hidden-xs">
          <figure class="thumbnail">
            <img class="img-responsive" src="images/user-avatar.png" />
            <figcaption class="text-center"><?php echo
ucwords($post['user']); ?></figcaption>
          </figure>
        </div>
        <div class="col-md-10 col-sm-10">
          <div class="panel panel-default arrow left">
            <div class="panel-body">
              <header class="text-left">
                <div class="comment-user"><i class="fa fa-user"></i>
By: <?php echo $post['user']; ?></div>
                <time class="comment-date" datetime="16-12-2014
01:05"><i class="fa fa-clock-o"></i> <?php echo date_format($date, 'd M
Y H:i:s'); ?></time>
              </header>
              <div class="comment-post">
                <p>
                <?php echo $post['message']; ?>
                </p>
              </div>
              <p class="text-right"><a href="#" class="btn btn-default
btn-sm"><i class="fa fa-reply"></i> reply</a></p>
            </div>
```

```
            </div>
        </div>
    </article>
<?php }   ?>
</div>
```

Мы реализуем `getPost()` в классе метод `Post.php` для получения записей.

index.php

```
public function getPost(){
    $sqlQuery = "
        SELECT *
        FROM ".$this->postsTable." ORDER BY id DESC";

    $stmt = $this->conn->prepare($sqlQuery);
    $stmt->execute();
    $result = $stmt->get_result();
    return $result;
}
```

Дополнения и Файлы

- Ссылка на оригинальную статью
    - Сохранение содержимого редактора TinyMCE с помощью PHP и MySQL

—-