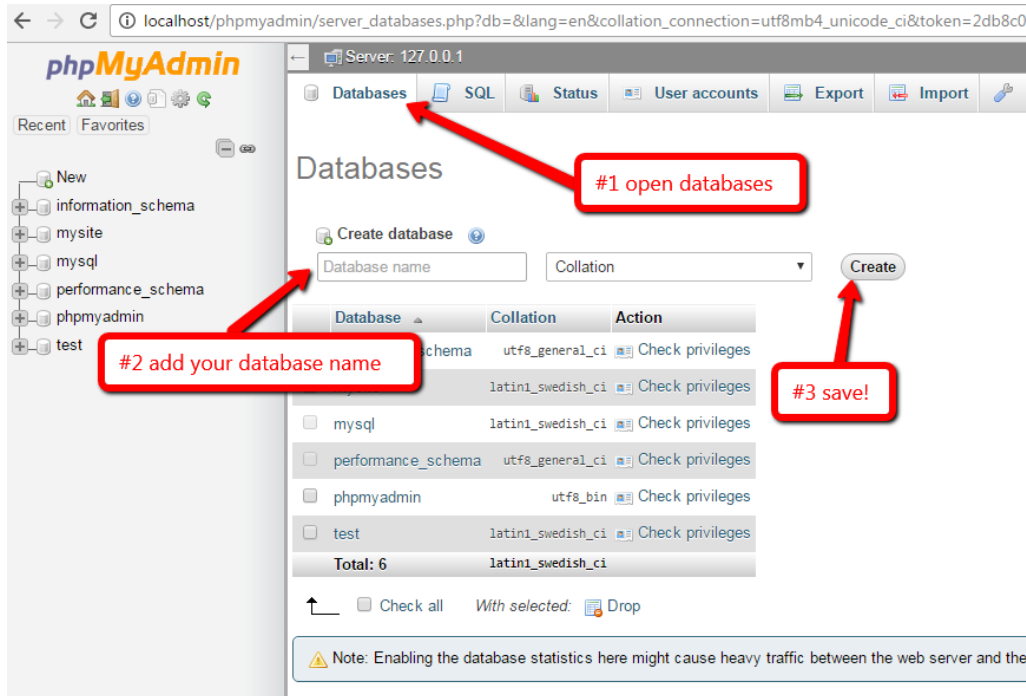


## Lorem Framework Tutorials – Part 2

### VI - Setting up Database and Configuration file

First, we need to have our database to connect with. Open *localhost/phpmyadmin*:



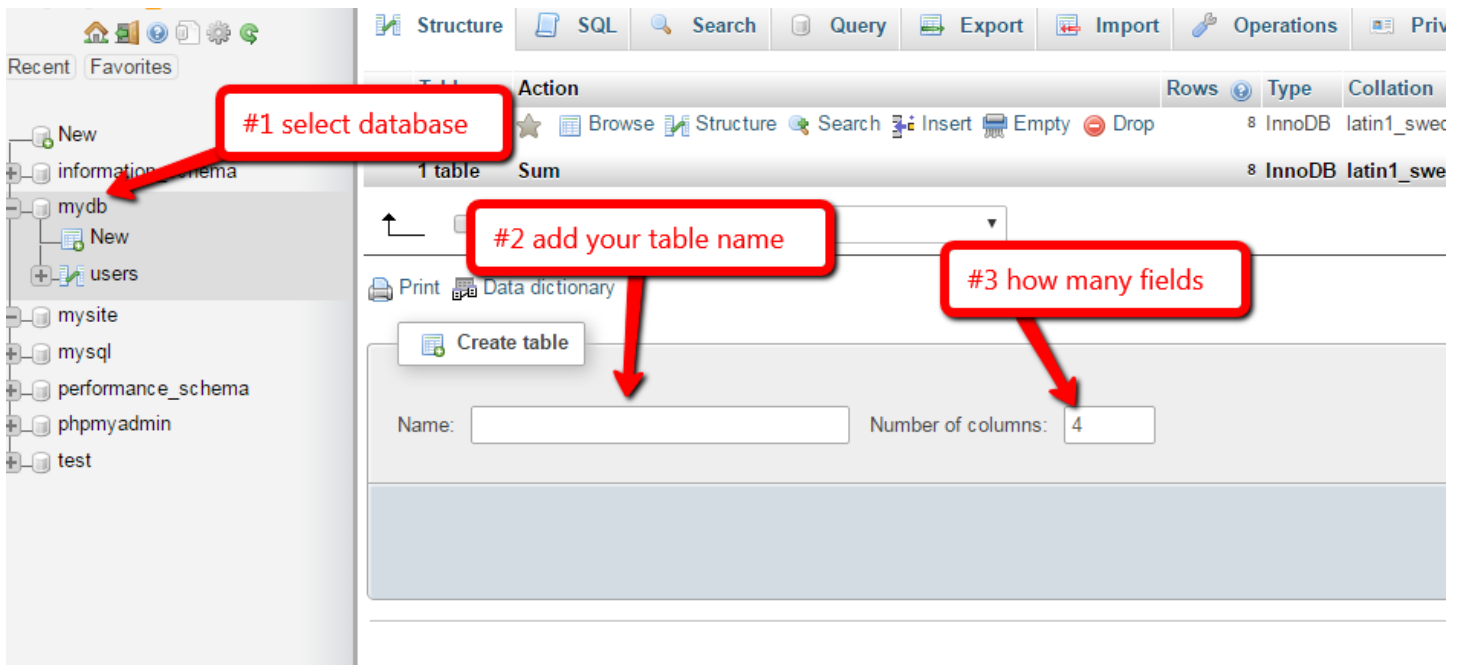
For this exercise, let's name our database "mydb". After you created the database, open config.php and update the file to this:

```
define( 'DBHOST', 'localhost' );
define( 'DBUSER', 'root' );
define( 'DBPASS', '' );
define( 'DBNAME', 'mydb' );
```

Let's leave the values by default except for the constant variable **DBNAME**, set the value to the name of the database you just created.

Next, let's add a new table in your database called 'users' and has this structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(10)	UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop
<input type="checkbox"/>	2	username	varchar(30)		No	None			Change  Drop
<input type="checkbox"/>	3	password	char(32)		No	None			Change  Drop
<input type="checkbox"/>	4	utype	varchar(20)		No	None			Change  Drop



## VII - Adding new records

Create a new page called **“add-new-user.php”** then save it in the **“pages”** folder then add the following codes:

```
<?php element( 'header' ); ?>
<form method="post">
    <input type="hidden" name="action" value="add_new_user">
    <label>Username</label>
    <input type="text" name="username"><br>
    <label>Password</label>
    <input type="text" name="password"><br>
    <label>Type</label>
    <input type="text" name="utype"><br>
    <input type="submit" value="Add new user">
</form>
<?php element( 'footer' ); ?>
```

In this code, you’ll see there is a *hidden* input, the value corresponds to a file in the **“actions”** folder.

```
<input type="hidden" name="action" value="add_new_user">
```

We are going to send the data to an action called **“add\_new\_user.php”** which can be found in the **“actions”** folder.

Next, let’s create the action **“add\_new\_user.php”** and save it in the **“actions”** folder then add the following code:

```
<?php
    $username    = $_POST[ 'username' ];
    $password    = md5( $_POST[ 'password' ] );
    $utype       = $_POST[ 'utype' ];
    $DB->query( "INSERT INTO users ( username, password, utype ) VALUES( '$username',
    '$password', '$utype' )" );
```

Now, open your browser and enter **http://localhost/mysite/?page=add-new-user** then new users.

---

Username

Password

Type

## VIII - Selecting all records

Since we can add our users, let's try to display them all. Create a new page called **"users.php"** and save it in your **"pages"** folder and add the following codes:

```
<?php element( 'header' ); ?>
<h1>Users</h1>
<?php
    $users = $DB->query( "SELECT * FROM users" );
?>
<a href="<?php echo SITE_URL ?>/?page=add-new-user">Add New User</a>
<table class="table table-bordered">
    <tr>
        <th>ID</th>
        <th>Username</th>
        <th>Password</th>
        <th>Type</th>
        <th></th>
    </tr>
    <?php if( $users->num_rows ) { ?>
    <?php while( $user = $users->fetch_object() ) { ?>
    <tr>
        <td><?php echo $user->id ?></td>
        <td><?php echo $user->username ?></td>
        <td><?php echo $user->password ?></td>
        <td><?php echo $user->utype ?></td>
        <td>
            <a href="<?php echo SITE_URL ?>/?page=edit_user&id=<?php echo $user->id
?>">Edit</a>
            <a href="<?php echo SITE_URL ?>/?action=delete_user&id=<?php echo
$user->id ?>">Delete</a>
        </td>
    </tr>
    <?php } ?>
    <?php } else { ?>
    <tr>
        <td colspan="4">No current record.</td>
    </tr>
    <?php } ?>
</table>
<?php element( 'footer' ); ?>
```

Now, open your browser and enter this URL: **http://localhost/mysite/?page=users**. Click the "Add New User" link to add new user.

Let's update **"add-new-user.php"** and make it redirect to the **"users"** page after adding a new user. Add this line of code at the end:

```
header( "Location: " . SITE_URL . "/?page=users" );
```

The constant variable **SITE\_URL** is available only in the framework, it automatically holds the current installation URL of your site. This means that if you installed the framework in folder **htdocs/new\_site\_url** then **SITE\_URL** will have the value: **“localhost/new\_site\_url”**.

TASK:

You will have to create a link in the main menu called “Users” and it should link to the users page.

## IX - Editing records

Create a new file called **“edit\_user.php”** and save it in the **“pages”** folder then add the following codes:

```
<?php element( 'header' ); ?>
<?php
    $id = $_GET[ 'id' ];
    $result = $DB->query( "SELECT * FROM users WHERE id = $id" );
    $user = $result->fetch_object();
?>
<form method="post">
    <input type="hidden" name="action" value="update_user">
    <input type="hidden" name="id" value="<?php echo $id ?>">
    <label>Username</label>
    <input type="text" name="username" value="<?php echo $user->username ?>"><br>
    <label>Password</label>
    <input type="text" name="password" value="<?php echo $user->password ?>"><br>
    <input type="submit" value="Update User">
</form>
<?php element( 'footer' ); ?>
```

Create a new file called **“update\_user.php”** and save it in the **“actions”** folder.

```
<?php
    $id = $_POST[ 'id' ];
    $username = $_POST[ 'username' ];
    $password = $_POST[ 'password' ];
    $DB->query( "UPDATE users SET username = '$username', password = '$password' WHERE
id = $id" );
    header( "Location: " . SITE_URL . "/?page=users" );
```

Try and see if the editing works!

## X - Delete records

Create a new file called **“delete\_user.php”** and save it in the **“actions”** folder then add the following codes:

```
<?php
    $id = $_GET[ 'id' ];
    $DB->query( "DELETE FROM users WHERE id = $id" );
    header( "Location: " . SITE_URL . "/?page=users" );
```

If you open **“users.php”** in the **“pages”** folder, you’ll see this line of code:

```
<a href="<?php echo SITE_URL ?>/?action=delete_user&id=<?php echo $user->id ?>">Delete</a>
```

You'll notice `?action=delete_user`, this means that we will call the action `"delete_user.php"` in the `"actions"` folder. Also we add another variable `id` to send the user's id, whatever value is passed and it exists in the users table will get deleted.

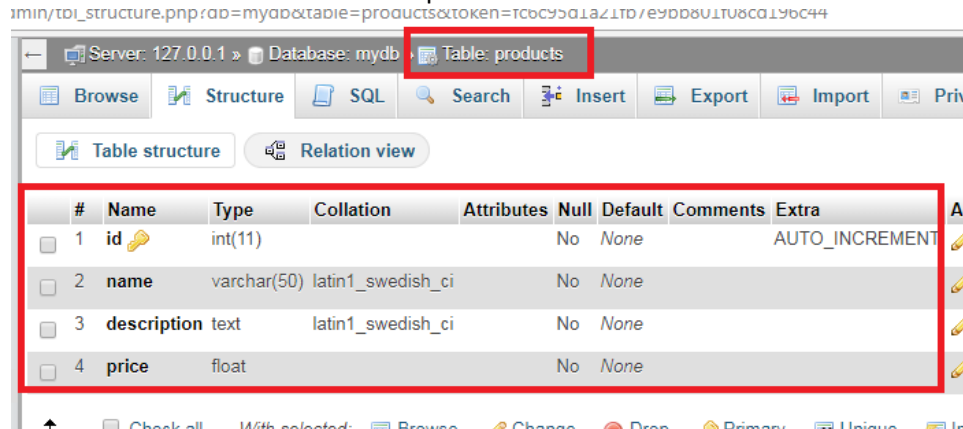
Try clicking on the "Delete" link to test it out, you'll see the record is deleted.

TASK:

You will need to create a new page called "Products" you will need to do the following to setup the page:

1. Add a menu called "Products" in the main navigation
2. Create a new table in the database called "products" with this structure:

min/tbl\_structure.php?db=mydb&table=products&token=TC6C95D1aZ1TD/e9DD8U1TU8CQ19bC44



The screenshot shows the phpMyAdmin interface for a database named 'mydb'. The 'Table: products' tab is selected. The 'Table structure' view is active, showing a table with 4 columns: id, name, description, and price. The 'id' column is an integer (11) with a primary key icon and is set to AUTO\_INCREMENT. The 'name' column is a varchar(50) with a latin1\_swedish\_ci collation. The 'description' column is a text field with a latin1\_swedish\_ci collation. The 'price' column is a float field. All columns are set to 'No' for null and 'None' for default. The table structure is highlighted with a red box.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	name	varchar(50)	latin1_swedish_ci		No	None		
3	description	text	latin1_swedish_ci		No	None		
4	price	float			No	None		

3. Same from what we did in the Users page, we should be able to add, edit, update and delete product records.