**NAME:-GUDUGUNTLA VENKATA SAITHARUN**

**Email id:-saitharungst@gmail.com**

**Mobile NO:-9705552386.**

-----------------------------------------------------------------------------------------------------------

## SOLUTIONS

Write a REST API with following features

1 .user registration.

Registration fields

-email

-phone

-password

-confirm password

**Source code:-**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"><html><head><META http-equiv="Content-Type" content="text/html; charset=utf-8"></head><body>

<form action="http://cookie.php" method="POST" target="\_blank" onsubmit="try {return window.confirm(&quot;This form may not function properly due to certain security constraints.\nContinue?&quot;);} catch (e) {return false;}">

<div bgcolor="Orange">

<div style="margin-left:400px;border:2px dashed red;width:400px;height:300px;float:left">

<table align="center" width="50%" border="0" cellspacing="3" cellpadding="5">

<th align="center" colspan="2"> User Registration </th>

<tr><td align="center">Email</td>

<td align="center"><input type="text" name="uname" size="15"></td>

</tr>

<tr><td align="center">Phone</td>

<td align="center"><input type="number" name="number" size="10"></td>

</tr>

<tr><td align="center">Password</td>

<td align="center"><input type="password" name="psw" size="15"></td>

<tr><td align="center"> Confirm Password</td>

<td align="center"><input type="password" required name="psw" size="15"></td>

<tr><td align="center"><input type="reset" value="RESET"></td>

<td align="center"><input type="submit" name="submit" value="SUBMIT"></td>

</tr></tr></table>

</div>

</div></form>

</body></html>

2. Registered user can create, read and update their profile

Profile fields

-First name

-Last Name

-role (editor/writer)

**Source code:-**

<html>

<body>

<form action="Welcome.php" method="POST">

firstName:<input type="text"name="firstname"<br>

lastname:<input type="text"name="lastname"<br>

Role:<input type="text"name="Editor/role"<br>

<input type="submit">

</form>

</body>

</html>

<!php

echo"<html>";

echo"<body bgcolor=$\_POST['color']>";

echo<h1>Welcome<pstyle="color:green;">$\_POST['name']</p></h1>

echo"</body>";

echo"</html>";

?>

3. User with writer role can create posts and read, update and delete only their posts

Article fields

-title

-post

**Sourcecode:-**

<?php

if(empty($\_POST['fname'])||empty($\_POST['lname']||empty($\_POST['role'])

empty($\_POST['uname'])||empty($\_POST['psw'])||empty($\_POST['cpsw'])

empty($\_POST['pcu'])||empty($\_POSt['em'])||empty($\_POST['br'])||(!isset($\_POST['la'])||(isset($\_POST['lan']))||(isset($\_POST['lang']))))))

{

echo"<html>";

echo"<body>";

echo"<h3 align='center'>Some credentials are missing</h3>";

echo"<span style='color:red'>please fill all the credentials which are marked</span>";

echo"<br><button onclick='goback()'>GoBack</button>";

echo"<script>";

echo"function goback(0);

echo"</script>";

echo"</html>";

}

else

{

$fn=$ln=$d=$b=$g=$un=$pa=$cpa=$ad=$p=$e=" ";

$fn=$\_POST['fname'];

$ln=$\_POST['lname'];

$r=$\_POST['role'];

}

4. User with writer role can read comments on his posts.

**USER ROLE-**

Users with the author role can write, edit, and publish their own posts. They can also delete their own posts, even if they are already published. When writing posts, authors cannot create new categories, but they can choose from existing ones. They can also add tags to their posts.

Authors are similar to contributors, but with more autonomy. They have full control over their own content, including the ability to create, publish, edit, and delete their own posts. Unlike contributors, authors can also add files to their content, and edit comments left on their posts.

Users can only be assigned to one role. This makes security cumbersome. For example, If a member of staff fills in for someone else, then there is no way to assign them to their main role and their temporary role at the same time.

5. User with editor role can read all posts and comment on posts.

**Editor Role:-**

delete\_others\_pages

delete\_others\_posts

delete\_pages

delete\_posts

delete\_private\_pages

delete\_private\_posts

delete\_published\_pages

delete\_published\_posts

delete Reusable Blocks

edit\_others\_pages

edit\_others\_posts

edit\_pages

edit\_posts

edit\_private\_pages

edit\_private\_posts

edit\_published\_pages

edit\_published\_posts

create Reusable Blocks

edit Reusable Blocks

manage\_categories

manage\_links

moderate\_comments

publish\_pages

publish\_posts

read

read\_private\_pages

read\_private\_posts

unfiltered\_html (not with Multisite)

upload\_files

For example, a user role includes capabilities such as:

* Managing comments;
* Writing blog posts;
* Adding pages;
* Installing or updating plugins or themes.
* Adding new users.

User roles are easy to use. They make sites accessible for multi-author concepts.

The super admin role has full access to all areas of a site, whereas the subscriber role has the least access. The subscriber role is only allowed to read posts but not to write or edit them. The contributor role can write posts but is not allowed to publish them. The author and editor roles are almost the same with a small difference, while the administrator has access to all of the features which are included in administration for a single website.