plant.php

```php
<!--  Multipart form -->
<form method="post" enctype="multipart/form-data">

        <!--  Get user's name from get_currenuserinfo(), a Wordpres function, then set it to a variable -->
        <?php
                global $current_user; get_currentuserinfo();
                $current_user=$current_user->display_name
        ?>

        <!--  Select a unit to plant, we need to get units from our database -->
        <h6>Available Units and Levels</h6>
        <select name="unitlevel_name" class="form-control" id="exampleFormControlSelect1">
        <!--  Only units shown are the empty ones -->
        <?php
        $sql = "SELECT * FROM `unitlevel_table` WHERE stage='empty'";
        $result = $wpdb->get_results($sql, ARRAY_A) ;
        foreach($result as $row){
                $unitlevel_id = $row['unitlevel_id'] ;
                $unit_name = $row['unit_name'];
                $unit_color = $row['unit_color'];
                $unit_seed = $row['unit_seed'];
        ?>
        <option value="<?php echo $unitlevel_id; ?>"><?php echo $unit_name; ?></option>
        <?php
        }
```

The code above is used to display a list of available units and levels for the user to choose from. The units shown are only those that are currently empty.

When the user submits the form, the selected unit and level will be inserted into the database as a new row. The unit_name, unit_color, and unit_seed columns will be populated with the values chosen by the user. The stage column will be set to 'planting'.

The code also sets a variable ($current_user) with the username of the currently logged-in user. This variable will be used when inserting the new row into the database.

```
</select>
<br>

<!--  Select seed type to plant, we need to get seeds from our database -->
<!--  Also we added Javascript code in onchange, it changes input text's placeholder -->
<h6>Seed Type</h6>
<select name="seed_name" class="form-control" id="exampleFormControlSelect1"
onchange='
let plantId = getElementsByName("seed_name")[0].value ;
if(plantId == 26 || plantId ==27 || plantId == 28)
        getElementsByName("planted_plugs")[0].placeholder = "Enter weight in ounces."
else
        getElementsByName("planted_plugs")[0].placeholder = "Enter planted plugs."
'>

<!-- To get data we used PHP before closing the select and 2 option HTML tags -->
<?php
//Query Operations
$sql = "SELECT * FROM `seed_table` ORDER BY type";
$result = $wpdb->get_results($sql, ARRAY_A) ;
foreach($result as $row){
        $seed_id = $row['seed_id'];
        $seed_name = $row['seed_name'];
        $seed_expected_date= $row['expected_time'];
        $type=$row['type'];
?>

<option value="<?php echo $seed_id; ?>"
<?php
// Make the line option unclickable
if ($type=="ds"){
        echo " disabled";
}
?>
>
```

The code above is used to create a drop-down menu of seed types that can be planted. The onchange event is used to change the placeholder text depending on the seed type that is selected. The PHP code is used to retrieve data from a database and populate the drop-down menu. The foreach loop is used to iterate through the results of the database query and create an option tag for each seed type. The seed_id and seed_name variables are used to store the id and name of each seed type. The expected_time variable is used to store the expected time it will take for the seed to germinate. The type variable is used to store the type of

seed (ds, cs, or ls). The disabled attribute is used to make the line option unclickable if the type is ds.

```
<!-- Getting user specifics from DB -->
<?php
//Querry Operations
if(isset($_POST['plant'])){

        //set variables from the form
        $unitlevel_name = $_POST['unitlevel_name'];
        $seed_name = $_POST['seed_name'];
        $color = $_POST['color'];
        $planted_plugs = $_POST['planted_plugs'];
        $planted_weight = $planted_plugs.' oz';

        // add date
        date_default_timezone_set('America/New_York');
        $t=time();
        $post_date = date("m-d-Y h:i A",$t);

        // use seed_name to acces type from seed_table
        $sql = "SELECT * FROM `seed_table` WHERE seed_id=$seed_name ";
        if($result = $wpdb->get_results($sql, ARRAY_A)){
                foreach($result as $row){
                        $seed_expected_date = $row['expected_time'];
                        $type = $row['type'];
                }
        }

        //fix might be needed later on, error is caused when there is 0 seeds in the seed_table
        else{
                $seed_expected_date='error';
        }

        // string concatenation to show better date
        $x='+'.$seed_expected_date;
        $expected_date = date("m/d/y", strtotime($x));

        // increment seed id by +1 for the next seed
        $sql = "SELECT * FROM `seed_id_table`";
        if($result = $wpdb->get_results($sql, ARRAY_A)){
                foreach($result as $row){
                        $seed_id = $row['seed_id'];
                        $seed_id += 1;
```

The code is a PHP script that queries a database for information about a particular plant. The script then prints out the expected date of harvest for the plant.

The first part of the script sets up variables from form input. These include the name of the plant, the type of seed used to plant it, and the number of seeds planted.

The second part of the script calculates the expected date of harvest for the plant based on the type of seed planted. This is done by querying the database for information about the type of seed and then using that information to calculate the expected date.

The third part of the script increments the seed id by one for future use. This ensures that each plant has a unique id.

The script is designed to help farmers keep track of their plants and to ensure that they are harvesting them at the correct time. This is important for both maximizing yield and ensuring that the plants are healthy.

farmers can use this code to help them keep track of their plants and ensure that they are harvesting them at the correct time. This is important for both maximizing yield and ensuring that the plants are healthy. The script can also be used to help farmers choose the best time to plant their crops based on the expected date of harvest.

The code is a PHP script that queries a database for information about a particular plant. The script then prints out the expected date of harvest for the plant. The first part of the script sets up variables from form input. These include the name of the plant, the type of seed used to plant it, and the number of seeds planted. The second part of the script calculates the expected date of harvest for the plant based on the type of seed planted. This is done by querying the database for information about the type of seed and then

using that information to calculate the expected date. The third part of the script increments the seed id by one for future use. This ensures that each plant has a unique id.