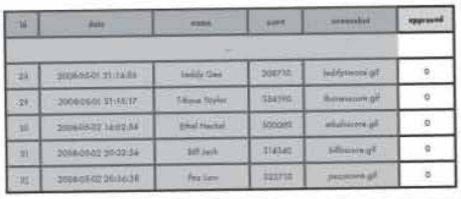
Plan for moderation in Guitar Wars

Adding a human moderation feature to Guitar Wars is significant because it affects several parts of the application. The database must change, a new script must be created to carry out an approval, the Admin page must add an "Approve" link to each score, and finally, the main page must change to only show approved scores. With this many changes involved, it's important to map out a plan and carry out each change one step at a time.

1 Use `ALTER` to add an `approved` column to the table.

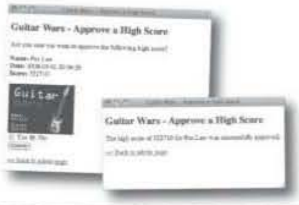
Let's start with the database, which needs a new column for keeping up with whether or not a score has been approved.



id	date	name	score	screenshot	approved
24	2008-05-01 21:14:0[illegible]	Teddy Gee	308710	teddyscore.gif	0
29	2008-05-01 21:15:17	T-Bone Taylor	354190	tbonescore.gif	0
30	2008-05-02 14:02:54	Ethel Heckel	500000	ethelscore.gif	0
31	2008-05-02 20:32:54	Biff Jeck	314340	biffscore.gif	0
32	2008-05-02 20:36:38	Pez Law	322710	pezscore.gif	0

2 Create an Approve Score script that handles approving a new high score (sets the `approved` column to 1).

With the database ready to accommodate high score approvals, you need a script to actually handle approving a score. This Approve Score script is responsible for looking up a specific score in the database and changing the `approved` column for it.

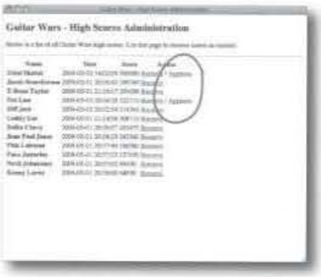


Guitar Wars - Approve a High Score

Guitar Wars - Approve a High Score

3 Modify the Admin page to include an "Approve" link for scores that have yet to be approved.

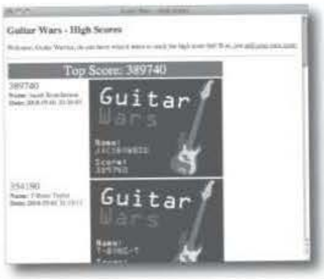
The Approve Score script is a back-end script that shouldn't normally be accessed directly. Instead, it is accessed through "Approve" links generated and displayed on the Admin page—only unapproved scores have the "Approve" link next to them.



Guitar Wars - High Scores Administration

4 Change the query on the main page to only show approved scores.

The last step is to make sure all this approval stuff gets factored into the main high score view. So the main page of the application changes to only show high scores that have been approved—without this change, all the other approval modifications would be pointless.



Guitar Wars - High Scores

Top Score: 389740

389740

354190

Make room for approvals with ALTER

Adding the new `approved` column to the `guitarwars` table involves a one-time usage of the `ALTER TABLE` statement, which is an SQL statement we've used before.

```
ALTER TABLE guitarwars
ADD COLUMN approved TINYINT
```

The MySQL data type BOOL is an alias for TINYINT, so you can use either one.

The new `approved` column is a `TINYINT` that uses `0` to indicate an unapproved score, or `1` to indicate an approved score. So all new scores should start out with a value of `0` to indicate that they are **initially unapproved**.

DONE

1 **Use `ALTER` to add an `approved` column to the table.**

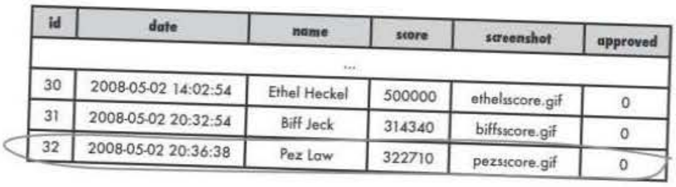
Wait a minute. I don't think you can just go adding a column to the database without changing the Add Score script—shouldn't it INSERT data into the new column?

It's true, a new column means a new value in the `INSERT` query in the Add Score script.

It's important to not lose sight of the fact that a PHP application is a careful orchestration of several pieces and parts: a database consisting of tables with rows and columns, PHP code, HTML code, and usually CSS code. It's not always immediately apparent that changing one part requires changing another. Adding the new `approved` column in the `guitarwars` table for the sake of the new Approve Score script also requires modifying the `INSERT` query in the Add Score script:

All newly inserted high score rows have approved set to 0... unapproved!

```
INSERT INTO guitarwars
VALUES (0, NOW(), '$name', '$score','$screenshot', 0)
```



id	date	name	score	screenshot	approved
...					
30	2008-05-02 14:02:54	Ethel Heckel	500000	ethelsscore.gif	0
31	2008-05-02 20:32:54	Biff Jeck	314340	biffsscore.gif	0
32	2008-05-02 20:36:38	Pez Law	322710	pezsscore.gif	0

When a new column is added, its approved column is set to 0 so that it starts out unapproved.

Sharpen your pencil

The Approve Score script is similar in structure to the Remove Score script except that its job is to approve a score. Finish the missing code for the Approve Score script, making sure to secure the page and only approve the appropriate score based on score data passed through a URL.

```
<?php
......................................................................;
?>
...
<?php
  require_once('appvars.php');
  require_once('connectvars.php');
  ...
  if (isset($_POST['submit'])) {
    if (..........................................................) {
      // Connect to the database
      $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

      // Approve the score by setting the approved column in the database
      $query = "UPDATE guitarwars SET ..................................................";
      mysqli_query($dbc, $query);
      mysqli_close($dbc);

      // Confirm success with the user
      echo ..........................................................................
    }
    else {
      echo ..........................................................................
    }
  }
  ...
  echo '<p><a href="....................">&lt;&lt; Back to admin page</a></p>';
?>
...
```