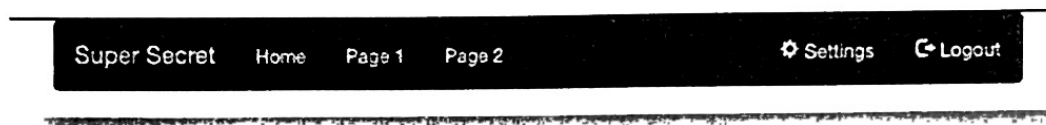
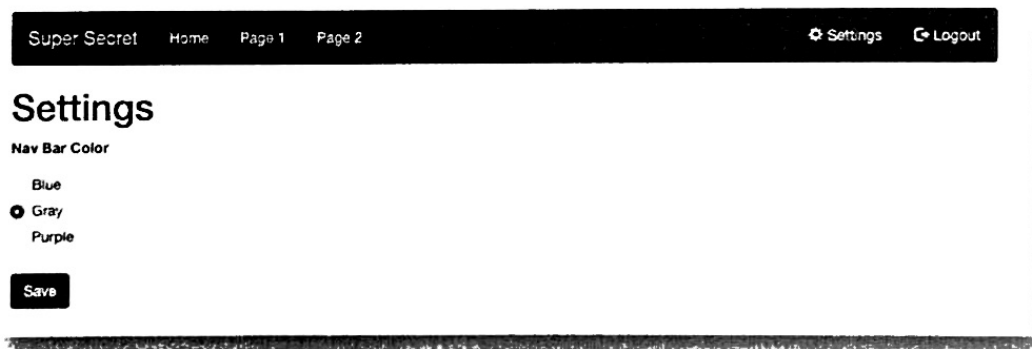


In this pair program, you will add a Settings page to the "login" application we created in class. The code is available in Canvas if you need it.

1. Add a Settings icon to your navigation bar. The icon shown here is glyphicon-cog, but you may use any image you wish. The Settings icon will link to a page called settings.php. (You'll create the page in step 3 below.)



2. Create an include file called navbar.html that contains your navigation bar code. Include the file on every page in your app.
3. Create a settings.php page, which will include a form that sets the color of the navigation bar. Use an array to populate the list of colors. Make sure that the Settings page is not accessible to users who are not logged in.



4. When the user submits the form, store the selected color in the session array. You'll need to start the session before you store the color in it. Therefore, the logic for your settings page will be:
 - a. Require the check-login.php script (this will start the session)
 - b. If the form has been submitted, assign the selected color to the session array
5. We will now create another include file called header.php that will use PHP to print the page title and use our session array to determine the color of the nav bar. Test header.php on one page, and then require it on every page in your app. Your code should look something like this?

header.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```

<title><?php echo $title; ?></title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
<style>
    .navbar-inverse {
        background-color: <?php echo $_SESSION['color']; ?>;
    }
</style>
</head>

```

index.php

```

<?php
    $title = "Admin Page";
    require('includes/check-login.php');
    require('includes/head.php');
?>

```

Note that in the settings file, we will process the form between the two require statements. Again, we need to start the session (check-login.php) before we process the form, and we need to process the form (which sets the color) before we display the navbar (head.php). That way, any changes the user makes will be visible on the page.

6. Modify your code so that when you visit the settings page, the color that is currently selected (and stored in the session array) should be checked. Make sure any variables you use are always initialized.
7. Add at least one additional form element (drop-down list, checkboxes, etc.) to your Settings page. You can decide what it controls... font style, font size, font color, or something else. Make sure that the form element on the Settings page displays the current setting, and that any changes get saved to the session array, and are reflected throughout the application.