COIS 3420
Winter 2020
Final Project

Name: Nikhil Pai Ganesh, Matthew Corr & Harrison Leitch

Live Web Link:
https://loki.trentu.ca/~nikhilpaiganesh/3420/project/dist/

Open Server (No Server Credentials Required)

Credentials for User Account in your application with Data:
Username: testuser
Password: test@user123

Functionality	/20
Design	/10
Complexity	/10
Code Quality	/5
Documentation & Testing	/ 5
Deductions (readability, submission guidelines)	
Total	/50

Table of Contents

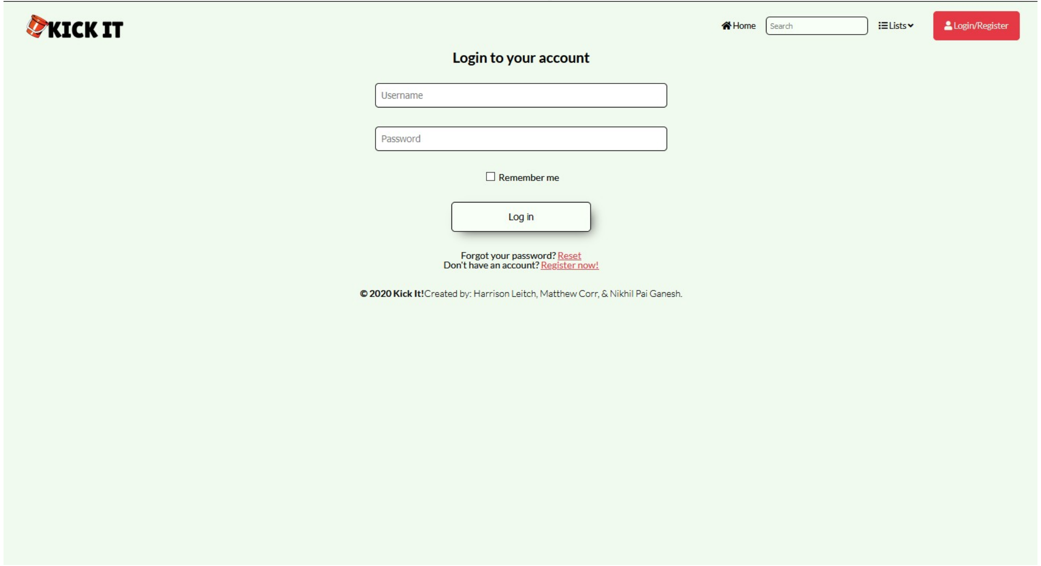
Table of Contents [Continued]

SCREENSHOTS [PAGES]:

login.php:

This page is only visible for users who have not yet logged in to an account, from here you can access the register account and reset password pages.

Windows 10 on Firefox Developer Edition



Login to your account

Username

Password

□ Remember me

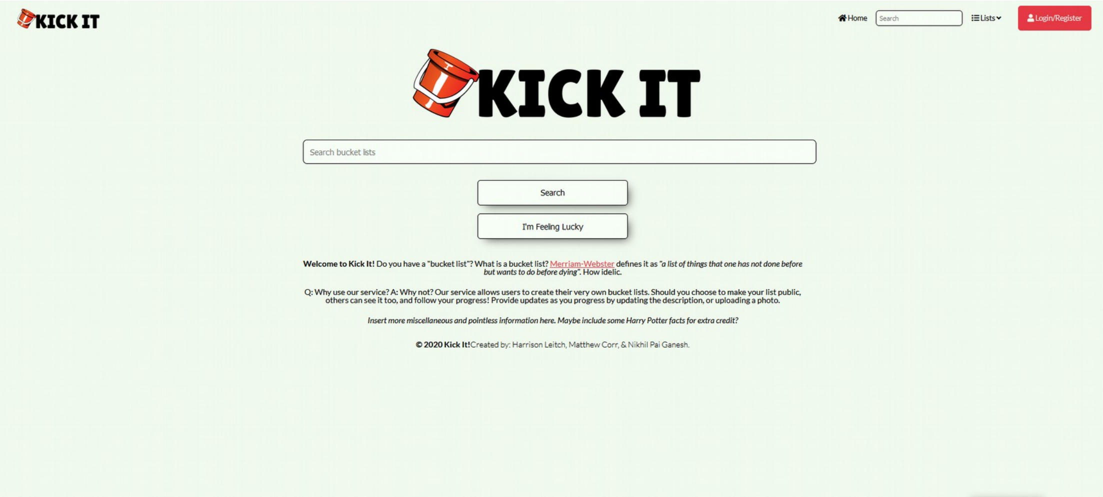
Log in

Forgot your password? Reset
Don't have an account? Register now!

© 2020 Kick It!Created by: Harrison Leitch, Matthew Corr, & Nikhil Pai Ganesh.

index.php:

The central landing page for our website. It displays Kick It’s logo and a search bar which will let you discover items on other public facing bucket lists. This page is also home to the Im Feeling Lucky button



Welcome to Kick It! Do you have a "bucket list"? What is a bucket list? Merriam-Webster defines it as *"a list of things that one has not done before but wants to do before dying"*. How idelic.

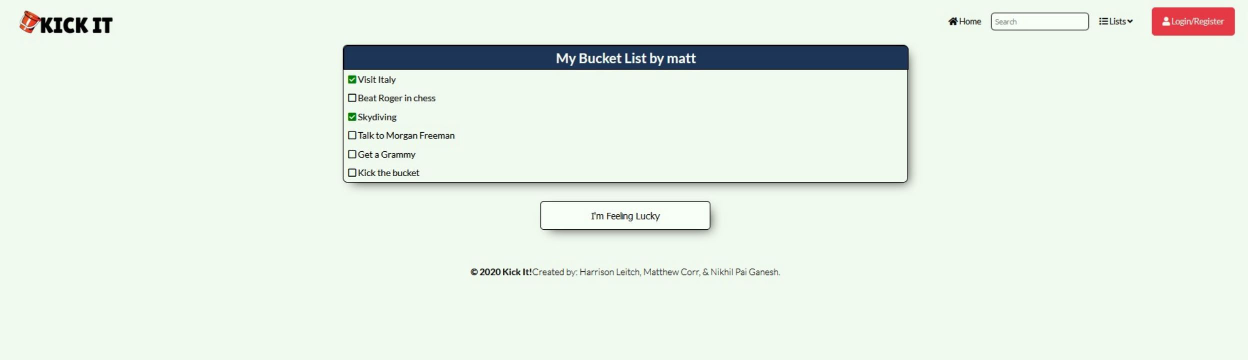
Q: Why use our service? A: Why not? Our service allows users to create their very own bucket lists. Should you choose to make your list public, others can see it too, and follow your progress! Provide updates as you progress by updating the description, or uploading a photo.

Insert more miscellaneous and pointless information here. Maybe include some Harry Potter facts for extra credit?

© 2020 Kick It!Created by: Harrison Leitch, Matthew Corr, & Nikhil Pai Ganesh.

lucky.php:
Displays a random public bucket list and contains a button that allows users to reroll for a different list

Windows 10 on Firefox Developer Edition



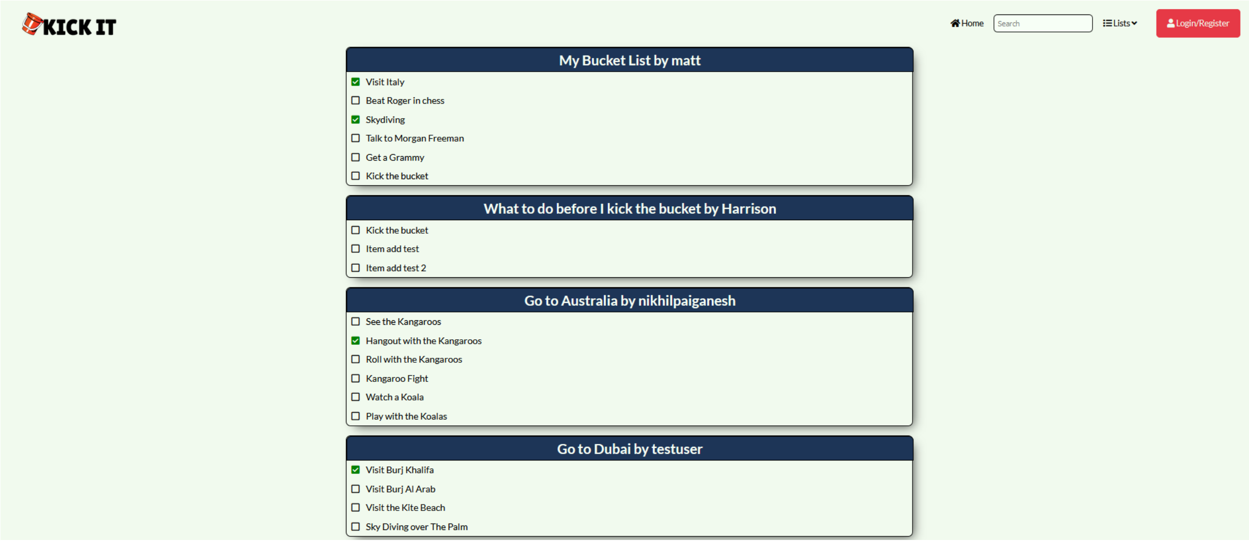
My Bucket List by matt

- ☑ Visit Italy
- □ Beat Roger in chess
- ☑ Skydiving
- □ Talk to Morgan Freeman
- □ Get a Grammy
- □ Kick the bucket

© 2020 Kick It!Created by: Harrison Leitch, Matthew Corr, & Nikhil Pai Ganesh.

public-lists.php:
This page displays all public facing bucket lists and will allow you to view uneditable details on each item.

Windows 10 on Firefox Developer Edition



My Bucket List by matt

- ☑ Visit Italy
- ☐ Beat Roger in chess
- ☑ Skydiving
- ☐ Talk to Morgan Freeman
- ☐ Get a Grammy
- ☐ Kick the bucket

What to do before I kick the bucket by Harrison

- ☐ Kick the bucket
- ☐ Item add test
- ☐ Item add test 2

Go to Australia by nikhilpaiganesh

- ☐ See the Kangaroos
- ☑ Hangout with the Kangaroos
- ☐ Roll with the Kangaroos
- ☐ Kangaroo Fight
- ☐ Watch a Koala
- ☐ Play with the Koalas

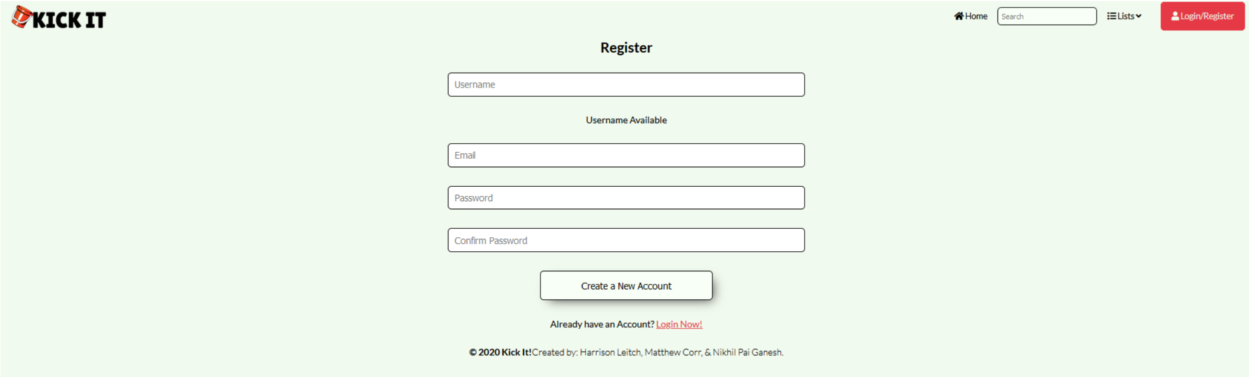
Go to Dubai by testuser

- ☑ Visit Burj Khalifa
- ☐ Visit Burj Al Arab
- ☐ Visit the Kite Beach
- ☐ Sky Diving over The Palm

register.php:

This is the registration page, it will display if your chosen username is available as well as providing a rough estimation to the security of your provided password (JQuery plugin).

Windows 10 on Firefox Developer Edition



Username Available

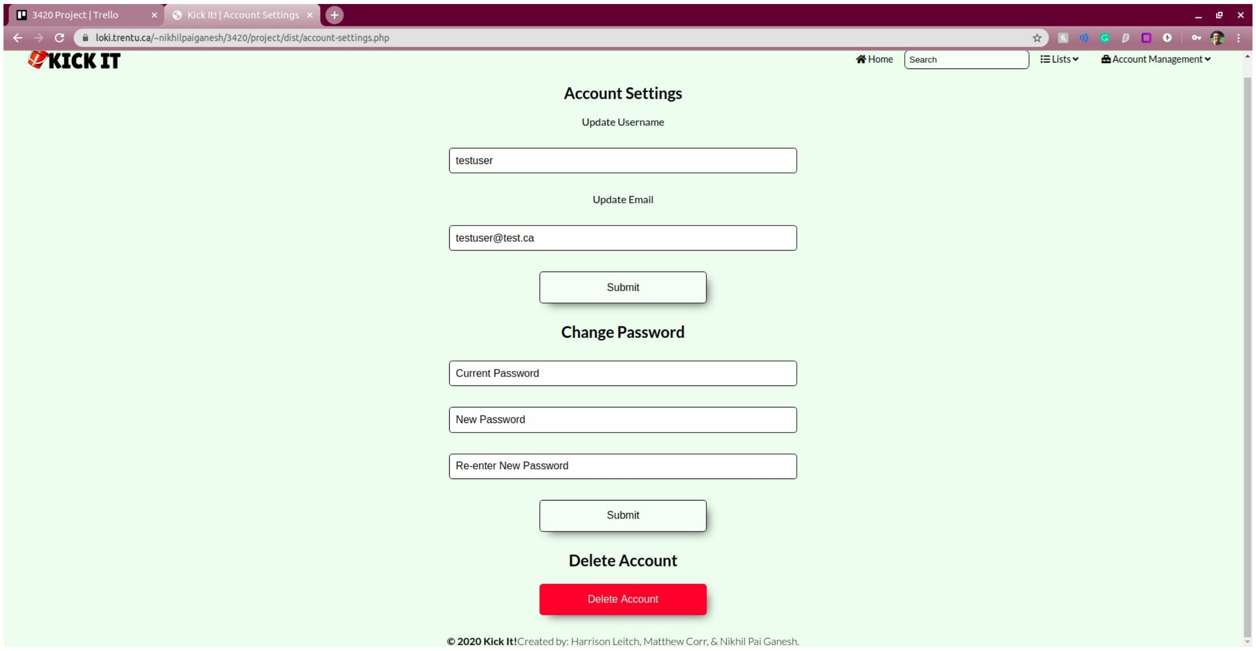
Already have an Account? Login Now!

© 2020 Kick It!Created by: Harrison Leitch, Matthew Corr, & Nikhil Pai Ganesh.

account-settings.php:

Settings page to update your login details such as username and email, change password and delete your account.

Ubuntu 18.4 on Chromium


3420 Project | Trello
Kick It! | Account Settings
loki.trentu.ca/~nikhilpaiganesh/3420/project/dist/account-settings.php
KICK IT
Home
Search
Lists
Account Management

Account Settings

Update Username

testuser

Update Email

testuser@test.ca

Submit

Change Password

Current Password

New Password

Re-enter New Password

Submit

Delete Account

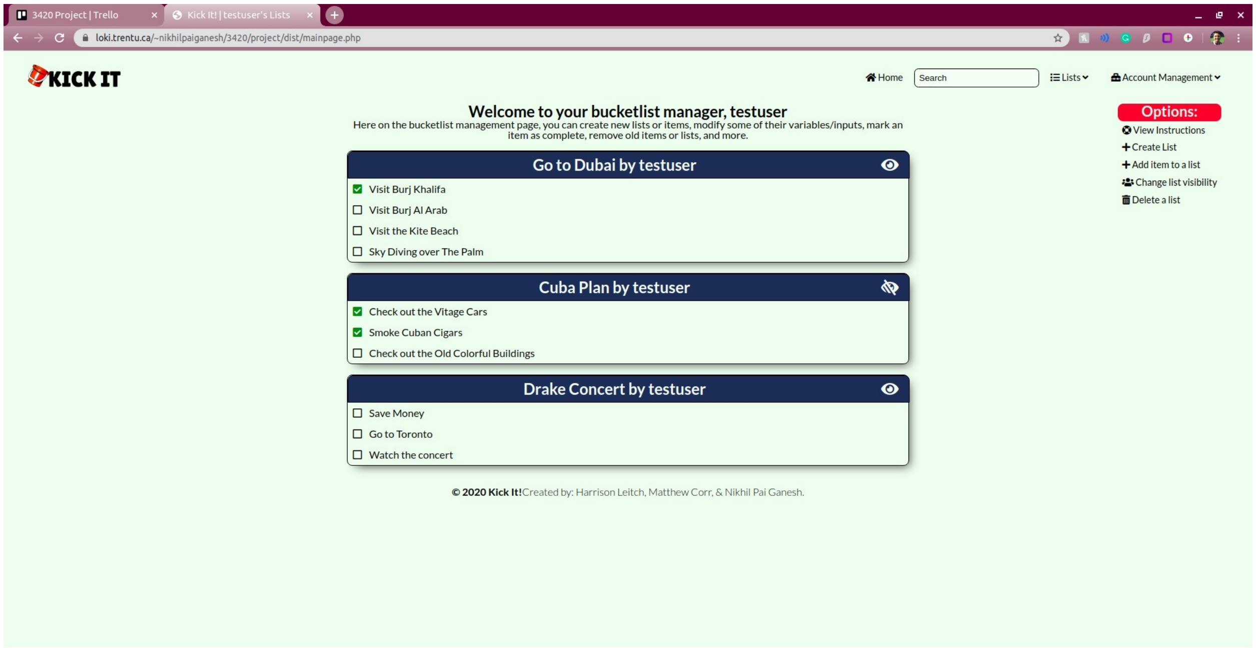
Delete Account

© 2020 Kick It!Created by: Harrison Leitch, Matthew Corr, & Nikhil Pai Ganesh.

mainpage.php:

Main page of Kickit! where you can manage and view your lists. This is where all the actions happen.

Ubuntu 18.4 on Chromium



Welcome to your bucketlist manager, testuser

Here on the bucketlist management page, you can create new lists or items, modify some of their variables/inputs, mark an item as complete, remove old items or lists, and more.

Go to Dubai by testuser

- ☑ Visit Burj Khalifa
- ☐ Visit Burj Al Arab
- ☐ Visit the Kite Beach
- ☐ Sky Diving over The Palm

Cuba Plan by testuser

- ☑ Check out the Vitage Cars
- ☑ Smoke Cuban Cigars
- ☐ Check out the Old Colorful Buildings

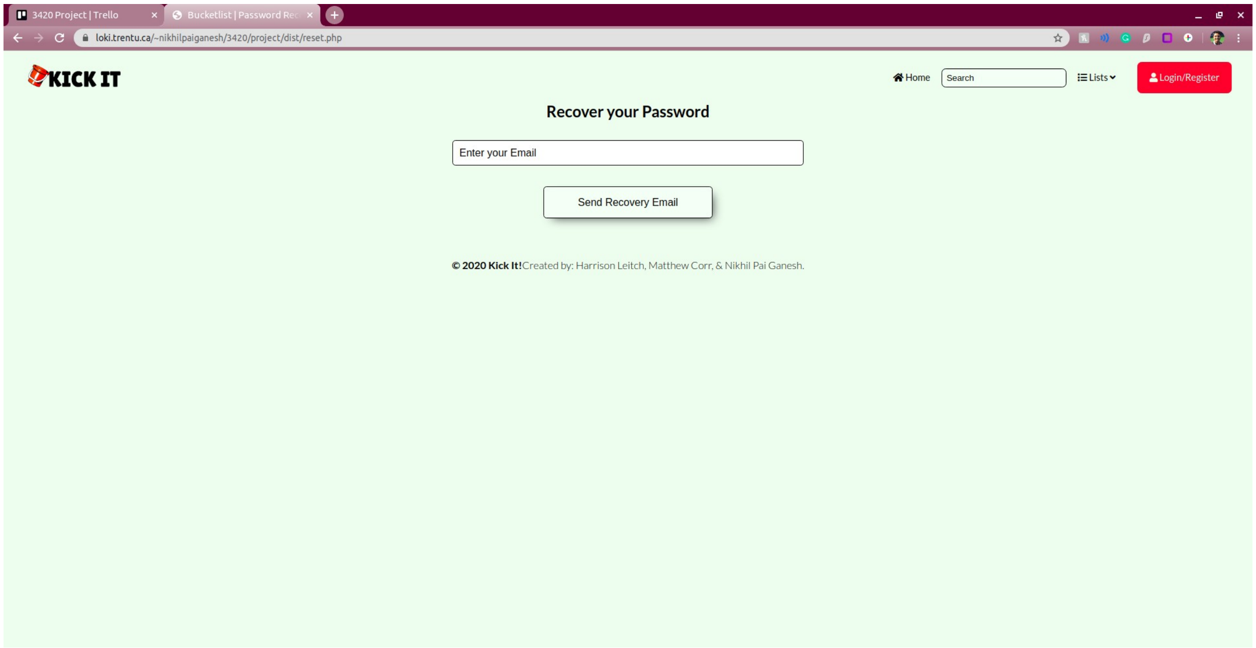
Drake Concert by testuser

- ☐ Save Money
- ☐ Go to Toronto
- ☐ Watch the concert

© 2020 Kick It!Created by: Harrison Leitch, Matthew Corr, & Nikhil Pai Ganesh.

reset.php:
Enter Email to send an one-time-password.

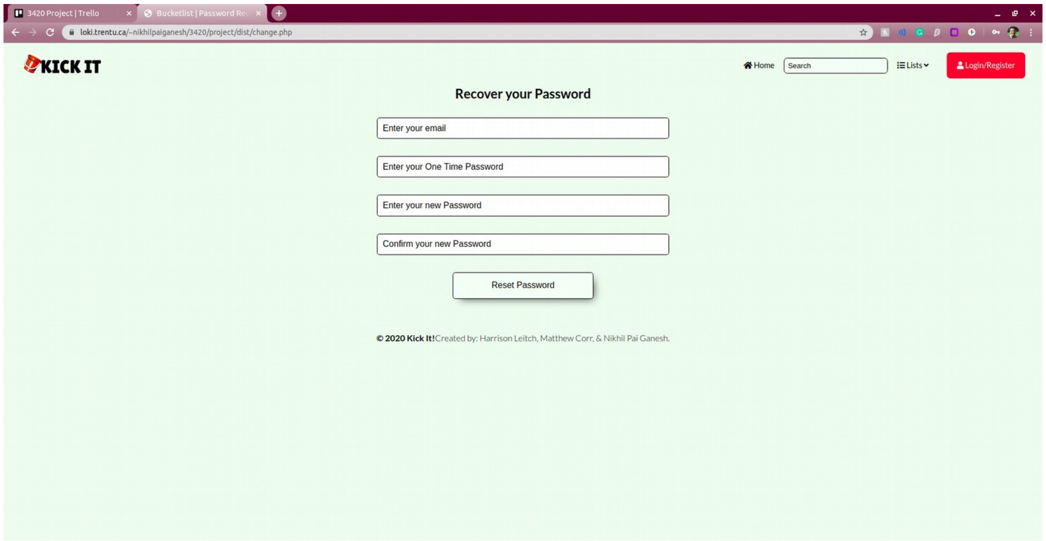
Ubuntu 18.4 on Chromium



© 2020 Kick It!Created by: Harrison Leitch, Matthew Corr, & Nikhil Pai Ganesh.

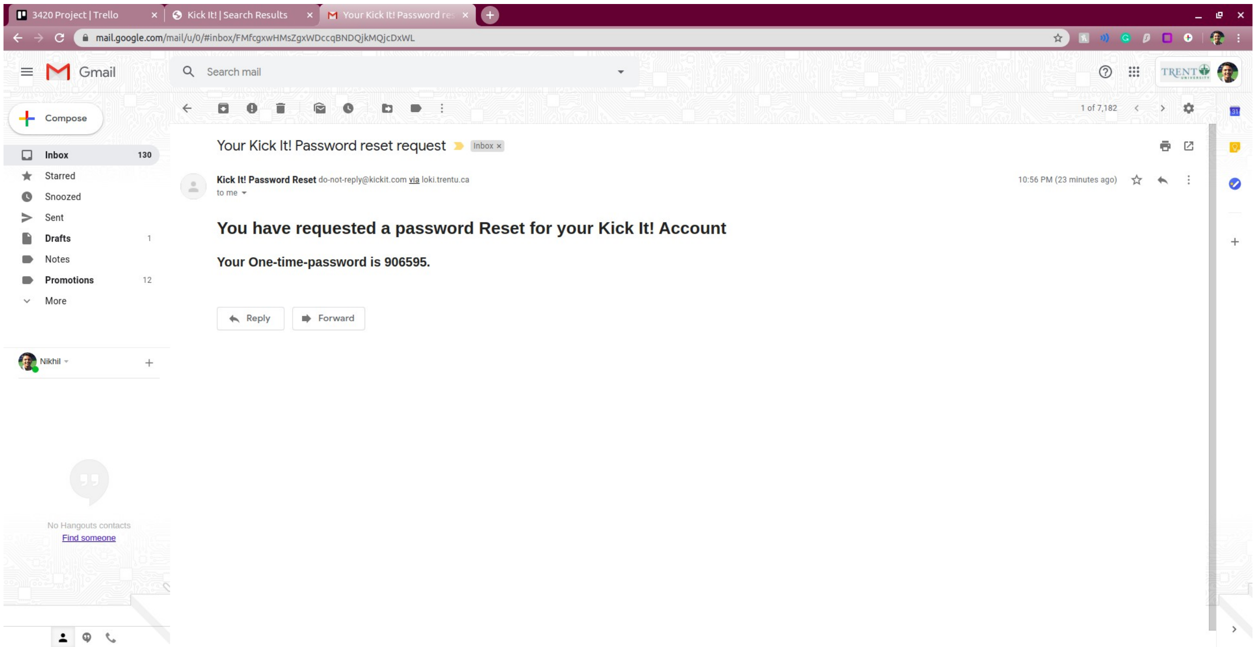
change.php:
Enter one time password to recover your account by changing password

Ubuntu 18.4 on Chromium



© 2020 Kick It!Created by: Harrison Leitch, Matthew Corr, & Nikhil Pai Ganesh.

Password reset email:



Your Kick It! Password reset request

Kick It! Password Reset do-not-reply@kickit.com via loki.trentu.ca

to me

10:56 PM (23 minutes ago)

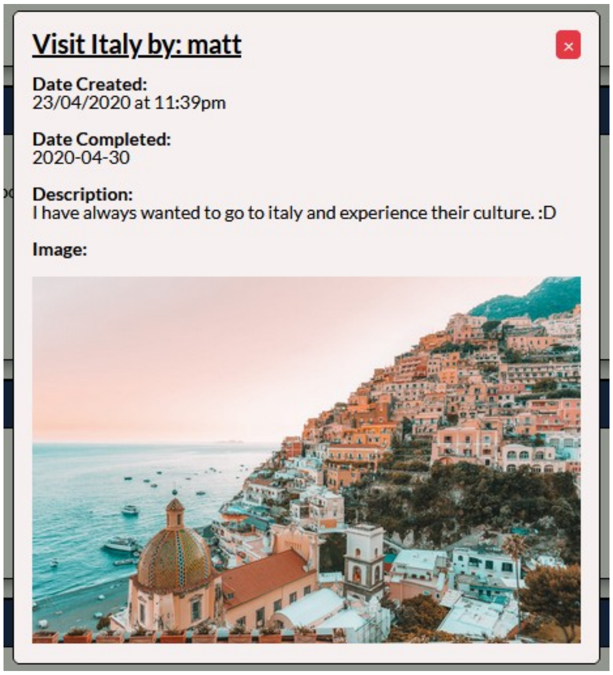
You have requested a password Reset for your Kick It! Account

Your One-time-password is 906595.

SCREENSHOTS [MODALS]:

View Item:

This is the “view details” modal when viewing someone else's list item, it cannot be edited unless you are the owner and on manage-lists.php.



Visit Italy by: matt

Date Created:
23/04/2020 at 11:39pm

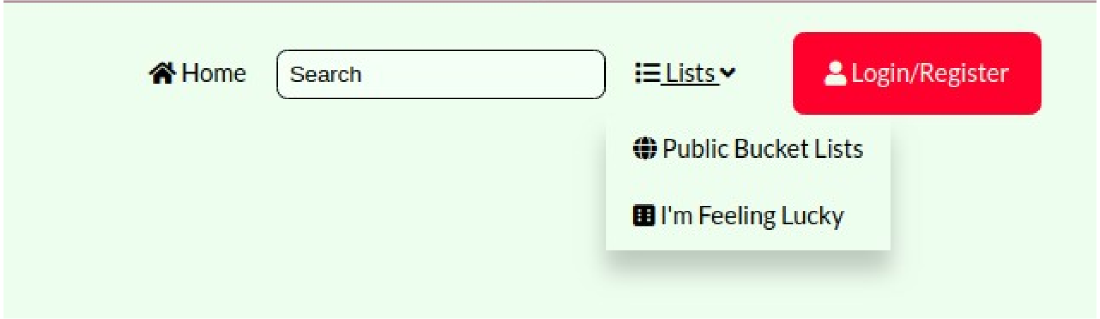
Date Completed:
2020-04-30

Description:
I have always wanted to go to italy and experience their culture. :D

Image:

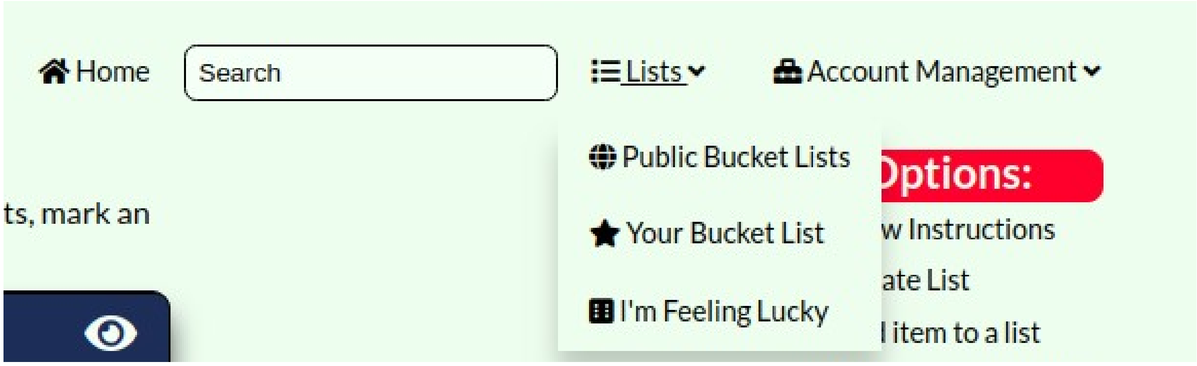
Navbar - Signed out

Navbar when user is not logged in



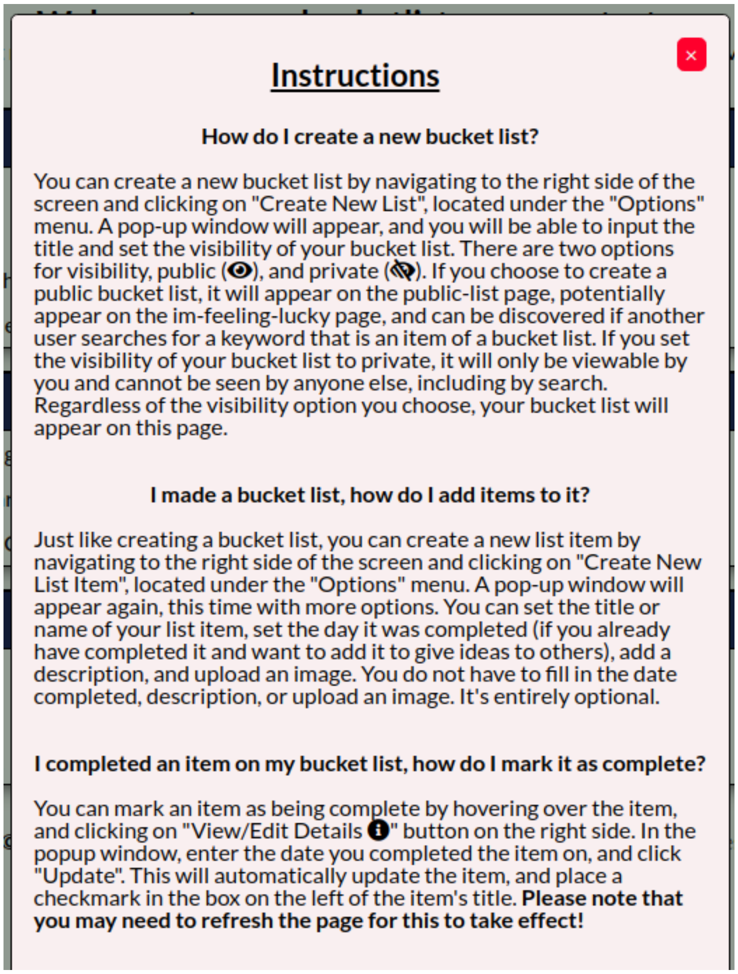
Navbar - Signed in

Navbar when user is logged in



View Instructions

Shows instructions



Instructions

×

How do I create a new bucket list?

You can create a new bucket list by navigating to the right side of the screen and clicking on "Create New List", located under the "Options" menu. A pop-up window will appear, and you will be able to input the title and set the visibility of your bucket list. There are two options for visibility, public (👁), and private (🚫). If you choose to create a public bucket list, it will appear on the public-list page, potentially appear on the im-feeling-lucky page, and can be discovered if another user searches for a keyword that is an item of a bucket list. If you set the visibility of your bucket list to private, it will only be viewable by you and cannot be seen by anyone else, including by search. Regardless of the visibility option you choose, your bucket list will appear on this page.

I made a bucket list, how do I add items to it?

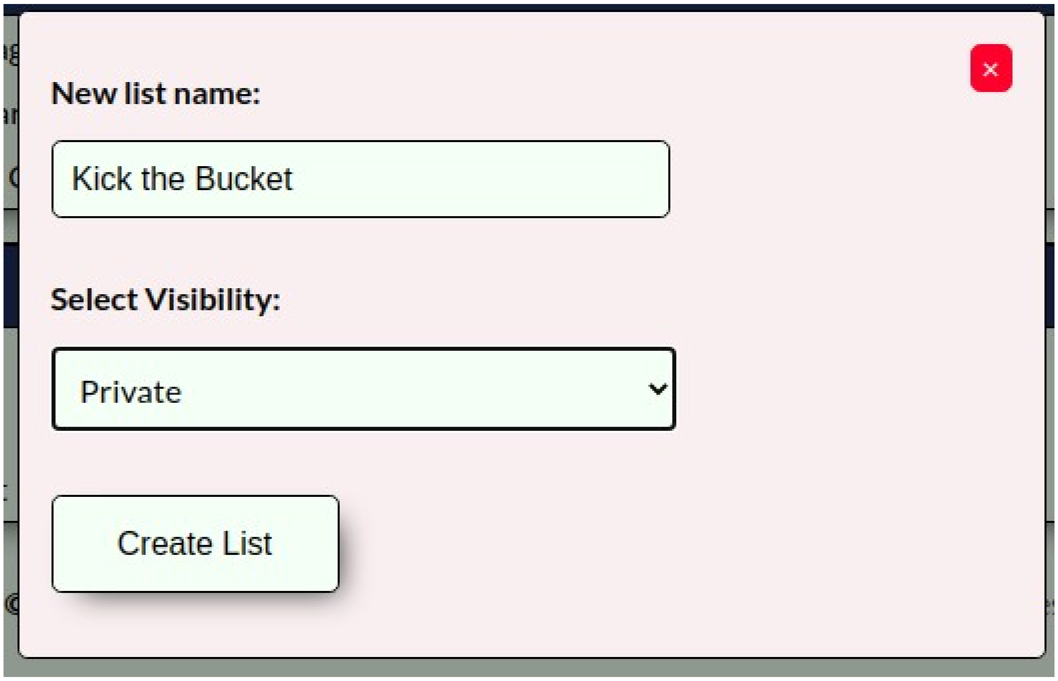
Just like creating a bucket list, you can create a new list item by navigating to the right side of the screen and clicking on "Create New List Item", located under the "Options" menu. A pop-up window will appear again, this time with more options. You can set the title or name of your list item, set the day it was completed (if you already have completed it and want to add it to give ideas to others), add a description, and upload an image. You do not have to fill in the date completed, description, or upload an image. It's entirely optional.

I completed an item on my bucket list, how do I mark it as complete?

You can mark an item as being complete by hovering over the item, and clicking on "View/Edit Details ℹ" button on the right side. In the popup window, enter the date you completed the item on, and click "Update". This will automatically update the item, and place a checkmark in the box on the left of the item's title. **Please note that you may need to refresh the page for this to take effect!**

Create List

Creates a new list

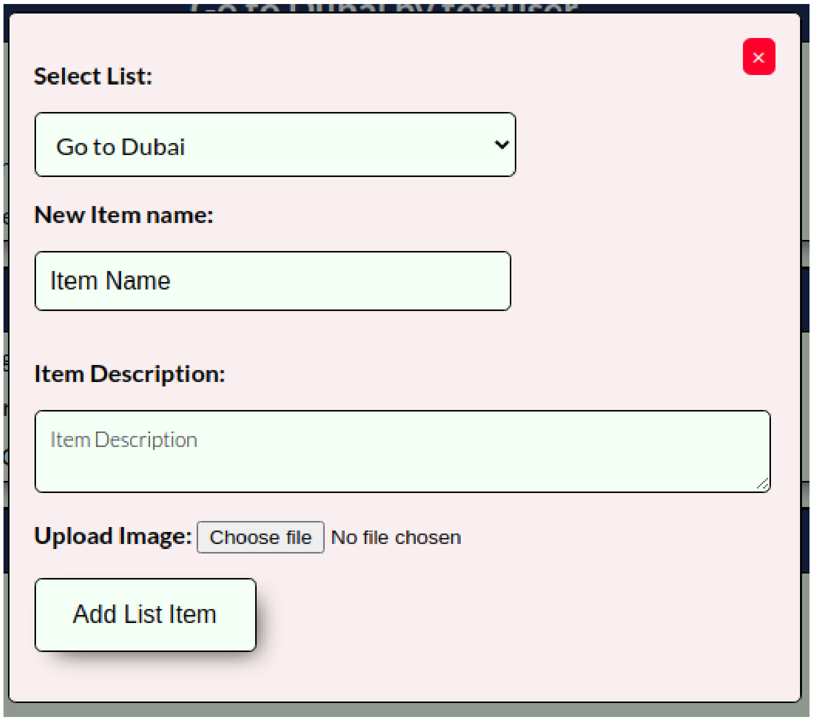


New list name: Kick the Bucket

Select Visibility: Private

Add new item to the list

Adds item to a list



Select List: Go to Dubai

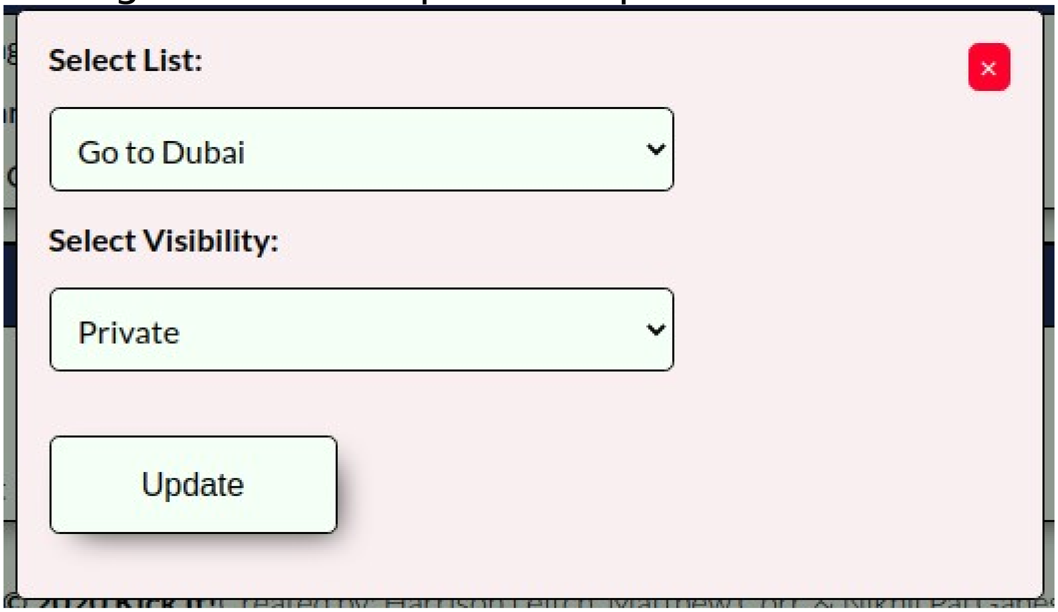
New Item name: Item Name

Item Description: Item Description

Upload Image: No file chosen

Change list visibility

Changes the list to public or private

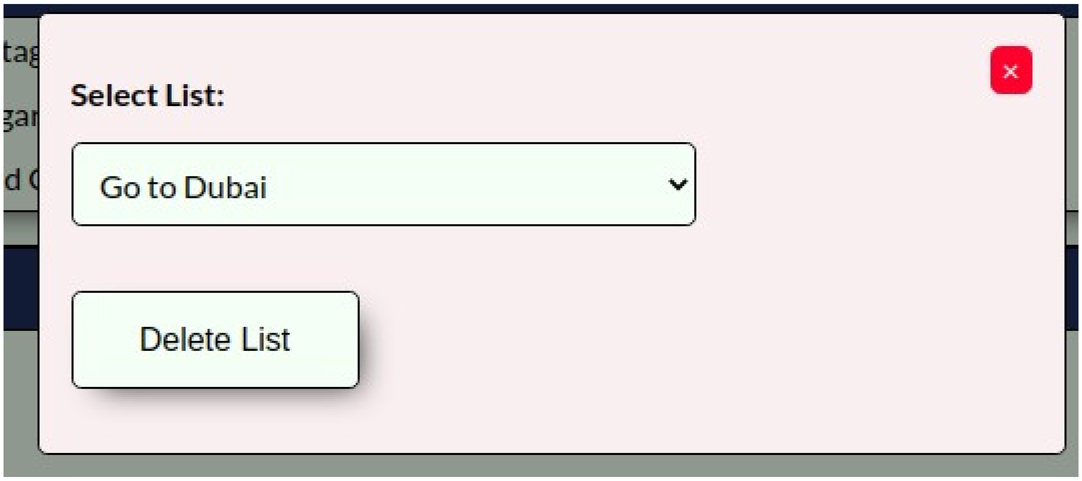


Select List: Go to Dubai

Select Visibility: Private

Deletes a list

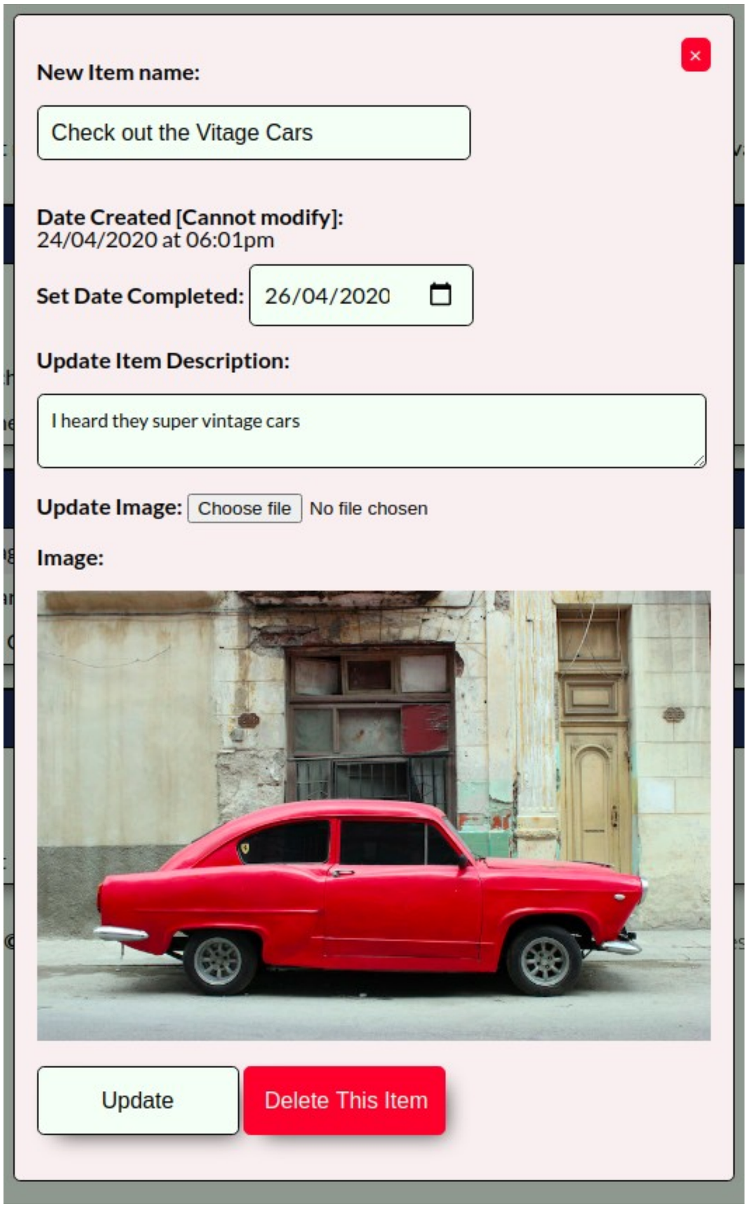
Deletes a bucket list



Select List: Go to Dubai

View/Edit Details

Found on your list items in mainpage.php. You can view, edit and delete your item in the list



New Item name:

Check out the Vitage Cars

Date Created [Cannot modify]:
24/04/2020 at 06:01pm

Set Date Completed: 26/04/2020

Update Item Description:

I heard they super vintage cars

Update Image: Choose file No file chosen

Image:

SOURCE CODE [HTML/PHP]:

account-settings.php

```
<?php
require "./includes/library.php"; // Include DB library
$pdo = connectDB();
```

```
session_start();

// If user has not logged in
if (!$_SESSION['loggedin']) {
  // Redirect them to the login page
  header("location:login.php");
  exit();
}
?>


<!DOCTYPE html>
<html lang="en-CA">

<head>
  <?php
  $PAGE_TITLE = "Kick It! | Account Settings";
  include "./includes/metadata.php";
  ?>

  <script src="https://kit.fontawesome.com/2cbf705ee6.js"
crossorigin="anonymous"></script>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></
script>
</head>

<body id="account-settings-page">
  <?php include("./includes/header.php"); ?>

  <?php
  // Grab the username and email using the session store variable
  $query = "SELECT username, email FROM project_users WHERE username = ?";
  $statement = $pdo->prepare($query);
  $statement->execute([$_SESSION['username']]);
  $result = $statement->fetch();
  ?>

  <!-- Change username / email -->
  <main>

    <h2> Account Settings </h2>

    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
      <label for="user">Update Username</label>
      <input class="text" type="text" name="user" value="<?=
$result['username']; ?>">
```

```
      <label for="email">Update Email</label>
      <input class="text" type="email" name="email" value="<?= $result['email']; ?>">
      <input type="submit" name="edit" value="Submit">

      <div class="form-message">
        <?php
        // User requests change of information
        if (isset($_POST['edit'])) {
          // Gather information from form
          $username = $_POST['user'];
          $email = $_POST['email'];

          // Update DB information
          $query = "UPDATE project_users SET username = ?, email = ? WHERE username = ?";
          $statement = $pdo->prepare($query);

          if ($statement->execute([$username, $email, $_SESSION['username']])) {
            // Update cookie and session variables
            setcookie('cookie', $username);
            $_SESSION['username'] = $username;

            // Log the user out so they can log back in with new information
            echo ("Information updated, please log back in");
            header("refresh:2;url=logout.php");
          } else {
            echo ("Couldn't update information");
          }
        }
        ?>
      </div> <!-- end form-message -->
    </form> <!-- end change-username/email -->

    <!-- Change password -->
    <h2>Change Password</h2>

    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
      <input class="text" type="password" name="current" placeholder="Current Password" required="">
      <input class="text" type="password" name="new" placeholder="New Password" required="">
      <input class="text" type="password" name="confirm" placeholder="Re-enter New Password" required="">
      <div class="form-message">
        <?php
        // Gather password hash for current user
        $query = "SELECT password FROM project_users WHERE username=?";
        $statement = $pdo->prepare($query);
        $statement->execute([$_SESSION['username']]);
        $result = $statement->fetch();

        // When change of password from is submitted
        if (isset($_POST['change'])) {
          $current = $_POST['current'];

          // Verify current password matches user supplied current password
          if (password_verify($current, $result['password'])) {
            $password = $_POST['new'];
            $confirm = $_POST['confirm'];

            // If both new passwords match
            if ($password == $confirm) {
              // Hash the new password
              $password = password_hash($password, PASSWORD_DEFAULT);

              // Prepare password update query
              $query = "UPDATE project_users SET password = ? WHERE username
= ?";

              $statement = $pdo->prepare($query);

              // If query was successful
              if ($statement->execute([$password, $_SESSION['username']])) {
                echo ('Password Changed, please log back in');
                header("Refresh:2;URL=logout.php");
              } else
                echo ('Could not update password');
            } else
              echo ('Passwords do not match');
          } else
            echo ("Current password was incorrect");
        }

        ?>
      </div> <!-- end form-message -->

      <input type="submit" name="change" value="Submit">
    </form> <!-- end change-password -->

    <!-- Delete Account -->
    <h2>Delete Account</h2>

    <form action="./delete.php" method="post">
      <button name="delete-account" id="delete-account"
          onclick="return confirm('Are you sure you want to delete your
account? You will not be able to recover your bucket list(s)!');">
        Delete Account
      </button>
    </form>

  </main>
  <?php include "./includes/footer.php"; ?>

</body>

</html>
```

change.php (password recovery)

```
<!DOCTYPE html>
<html lang="en-CA">
  <head>
    <?php
      $PAGE_TITLE = "Bucketlist | Password Recovery";
      include "./includes/metadata.php";
    ?>

    <script src="https://kit.fontawesome.com/2cbf705ee6.js"
crossorigin="anonymous"></script>
  </head>

  <body id="password-recovery-page">

    <?php include "./includes/header.php"; ?>
    <main>
      <div class="form">
        <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
          <h2>Recover your Password</h2>

          <input class="text" type="text" name="email" placeholder="Enter
your email" require>
          <input class="text" type="text" name="otp" placeholder="Enter your
One Time Password" require>
          <input class="text" type="password" name="password"
placeholder="Enter your new Password" require>
          <input class="text" type="password" name="confirm_password"
placeholder="Confirm your new Password" require>
```

```
     <input type="submit" name="reset" value="Reset Password">

     <?php
      require "./includes/library.php";
      $pdo = connectDB();

      // If reset submit button has been pressed
      if (isset($_POST['reset'])) {
       // Retrieve variables from POST array
       $otp = $_POST['otp'];
       $email = $_POST['email'];
       $password = $_POST['password'];
       $confirm = $_POST['confirm_password'];

       // Verify that the email and one time password exist
       $check = $pdo->prepare("SELECT COUNT(ID) FROM project_users WHERE email ='$email' AND token = '$otp'");
       $check->execute();
       $row = $check->fetchColumn();
       if ($row > 0) {
        // If the supplied passwords match
        if ($confirm == $password) {
         // Hash the password and update the database
         $password = password_hash($password, PASSWORD_DEFAULT);
         $query = "UPDATE project_users SET password = '$password' WHERE email = '$email'";
         $statement = $pdo->prepare($query);
         if ($statement->execute([])){
          $tokenDel = $pdo->prepare("UPDATE project_users SET token = NULL WHERE email = '$email'");
          $tokenDel->execute();
          echo ('Password Changed');
          header('Location: login.php');
         }
        } else {
         echo ('Passwords do not match');
        }
       } else {
        echo "Email and One Time Password do not match";
       }
      }
     ?>
     </form>
    </div> <!-- end form -->
  </main>
  <?php include "./includes/footer.php"; ?>
 </body>
```

```
</html>
```

delete.php (delete an account)

```php
<?php
    require "./includes/library.php"; // Include database connection library
    $pdo = connectDB(); // Connect to database

    session_start(); // Start $_SESSION

    // Prepare query
    $query = "DELETE FROM project_users WHERE username = ?";
    $statement = $pdo->prepare($query);

    //Fetches id from username
    $username = $_SESSION['username'];
    $query1 = "SELECT * FROM project_users WHERE username = '$username'";
    $statement1 = $pdo->prepare($query1);
    $statement1->execute([]);
    $lists = $statement1->fetch(PDO::FETCH_ASSOC);
    $userID = $lists['ID'];

    // If query was successful
    if($statement->execute([$_SESSION['username']])){
        session_destroy();
        setcookie('cookie', '', time()-1000); // Erase user cookie
    } else {
        header("location:settings.php");
    }

    //Fetches listid
    $query2 = "SELECT * FROM project_lists WHERE userID = $userID";
    $statement3 = $pdo->prepare($query2);
    $statement3->execute([]);
    $list_ID = $statement3->fetch(PDO::FETCH_ASSOC);
    $listID = $list_ID['ID'];

    //Deletes list items associated with the list
    $sql1 = "DELETE FROM project_list_items WHERE listID = $listID";
    $statement4 = $pdo->prepare($sql1);
    $statement4->execute([]);

    //Deletes Lists associated with the username
    $sql = "DELETE FROM project_lists WHERE userID = $userID";
    $statement2 = $pdo->prepare($sql);
    $statement2->execute([]);
    header("location:login.php");
?>
```

index.php (landing page)

```
<!DOCTYPE html>
<html lang="en-CA">
  <head>
    <?php
      $PAGE_TITLE = "Kick It! | Home";
      include "./includes/metadata.php";
    ?>
  </head>

  <body id="index-page"> <!-- Create unique ID for index page -->
    <?php include "./includes/header.php"; ?>
    <main>
      <img class="index-logo" src="./img/bucketlist.png" alt="Kick It!"/>

      <!-- self-processing form -->
      <form method="post" action="./list-search.php">
        <div>
          <label for="index-search-string"></label>
          <input id="index-search-string" name="index-search-string"
type="text" placeholder="Search bucket lists">
        </div>

        <button id="query-search-button" name="query-search-button"
type="submit">Search</button>
      </form>

      <!-- I'm Feeling Lucky Button -->
      <button id="search-lucky-button" name="search-lucky-button"
type="submit" onclick="window.location.href='./lucky.php'">I'm Feeling
Lucky</button>

      <div id="index-information-blurb">
        <p>
          <strong>Welcome to Kick It!</strong>
          Do you have a "bucket list"? What is a bucket list?
          <a href="https://www.merriam-webster.com/dictionary/bucket%20list"
target="_blank">Merriam-Webster</a> defines it as <em>"a list of things that
one has not done before but wants to do before dying".</em>
          How idelic.
        </p>
        <p>
          Q: Why use our service?
          A: Why not?
          Our service allows users to create their very own bucket lists.
          Should you choose to make your list public, others can see it too, and follow your progress!
          Provide updates as you progress by updating the description, or uploading a photo.
        </p>
        <p>
          <em>Insert more miscellaneous and pointless information here. Maybe include some Harry Potter facts for extra credit?</em>
        </p>
      </div> <!-- end index-information-blurb -->
    </main>
    <?php include "./includes/footer.php"; ?>
  </body>
</html>
```

list-search.php (query results using searchbars)

```
<?php
require "./includes/library.php"; // Include database connection library
$pdo = connectDB(); // Connect to database

session_start(); // Start $_SESSION

/* This is to determine which input form the user has used
   * Without it, an undefined index error is thrown, or one is overwritten by the other.
   */
if (isset($_POST['index-search-string'])) {
  $search = $_POST['index-search-string'] ?? NULL;
} else {
  $search = $_POST['nav-search-string'] ?? NULL;
}
?>

<!DOCTYPE html>
<html lang="en-CA">

<head>
  <?php
  $PAGE_TITLE = "Kick It! | Search Results";
  include "./includes/metadata.php";
  ?>
  <script defer src="scripts/modal-view-list-item.js"></script>
</head>

<body id="seach-results-page">
  <!-- Create unique ID for index page -->
  <?php include "./includes/header.php"; ?>
  <main>
    <h2>Searching for: "<?php echo $search; ?>"</h2>
    <?php
    // Select all public lists
    $query = "SELECT * FROM `project_lists` WHERE visibility = 1";
    $statement = $pdo->prepare($query);
    $statement->execute([]);
    $publicLists = $statement->fetchAll();
    ?>

    <!-- For each potential list -->
    <?php foreach ($publicLists as $potentialMatch) : ?>

      <!-- Select items in the list where the name matches the search query
-->
      <?php
      $query = "SELECT * FROM `project_list_items` WHERE (name LIKE '%" .
$search . "%') AND listID = ?";
      $statement = $pdo->prepare($query);
      $statement->execute([$potentialMatch['ID']]);
      $matchingListItems = $statement->fetchAll();
      ?>

      <!-- For every matching list item -->
      <?php foreach ($matchingListItems as $matchingItem) : ?>
        <div>
          <!-- Output the list containing the item -->
          <?php
          $query = "SELECT * FROM project_lists WHERE ID = ?;";
          $statement = $pdo->prepare($query);
          $statement->execute([$matchingItem['listID']]);
          $lists = $statement->fetchAll();
          ?>

          <?php foreach ($lists as $list) : ?>
            <div class="bucket-list">
              <?php
              // Find the owner of the current list
              $query = "SELECT username FROM project_users WHERE ID = ?";
              $statement = $pdo->prepare($query);
              $statement->execute([$list['userID']]);
              $listOwner = $statement->fetch();
              ?>

              <!-- Display list title -->
              <div class="list-header">
                <h2><?= $list['name']; ?> by <?= $listOwner['username'] ?></
h2>
              </div>

              <ul>
                <?php
                $query = "SELECT * FROM project_list_items WHERE listID
= ?;";
                $statement = $pdo->prepare($query);
                $statement->execute([$list['ID']]);
                $itemsInList = $statement->fetchAll();
                ?>

                <?php foreach ($itemsInList as $item) : ?>
                  <!-- HTML for displaying list item -->
                  <li>
                    <div class="list-item <?php if(stripos($item['name'],
$search) !== false){echo("selected");}?>">
                      <?php
                      if ($item['itemCompletion'] == 0) {
                        echo '<i class="far fa-square"></i> ';
                      } else {
                        echo '<i class="fa fa-check-square"></i> ';
                      }
                      ?>
                      <span id="list-item-main"> <?= $item['name']; ?
></span>

                      <div class="list-item-buttons list-item-view-button">
                        View Details <i class="fas fa-info-circle"></i>
                      </div> <!-- end list-item-buttons -->
                    </div> <!-- end list-item -->

                    <!-- Modal to view list item -->
                    <div class="modal-view-list-item modal">
                      <div class="modal-content">
                        <span class="close-view-list-item modal-
close">&times;</span>
                        <h3><?=$item['name'];?> by: <?
=$listOwner['username'];?></h3>
```

```
                        <div>
                          <h4>Date Created: </h4>
                          <p><?=$item['date']?></p>
                        </div>

                        <div>
                          <h4>Date Completed: </h4>
                          <?php
                            // Check to see if there is a date for
completion.
                            // If there is, show the date and time it was
marked as being complete.
                            if ($item['dateCompleted'] != NULL) {
                              echo '<p>';
                              echo ($item['dateCompleted']);
                              echo '</p>';
                            } else {
                              echo '<p>This item is not yet complete.</p>';
                            }
                          ?>
                        </div>

                        <div>
                          <h4>Description:</h4>
                          <p><?=$item['description'];?></p>
                        </div>

                        <div>
                          <h4>Image:</h4>
                          <?php
                            // Check to see if there is a date for
completion.
                            // If there is, show the date and time it was
marked as being complete.
                            if ($item['imagepath'] != NULL) {
                              echo '<img src="../../../www_data/';
                              echo ($item['imagepath']);
                              echo '" alt="Image uploaded by user">';
                            } else {
                              echo '<p>No image was provided.</p>';
                            }
                          ?>
                        </div>
                      </div> <!-- end modal-content -->
                    </div> <!-- end modal-view-list-item modal -->
                  </li>
```

```
                <?php endforeach ?>
              </ul>
            </div>
          <?php endforeach ?>


        </div>
      <?php endforeach //($matchingListItems as $matchingItem)?>
    <?php endforeach //($publicLists as $potentialMatch)?>
  </main>
  <?php include "./includes/footer.php"; ?>
</body>

</html>
```

login.php (login page)

```
<?php
  require "./includes/library.php"; // Include database connection library
  $pdo = connectDB(); // Connect to database

  session_start(); // Start $_SESSION

  // This is to catch any possible errors when the user logs out.
  // If this statement is not present, it throws an undefined index error.
  if(!isset($_SESSION["loggedin"])){
    $_SESSION["loggedin"] = false;
  }

  // If user is already logged in, redirect them to user settings
  if ($_SESSION["loggedin"] == true){
    header("Location:account-settings.php");
  }
?>

<!DOCTYPE html>
<html lang="en-CA">
  <head>
    <?php
      $PAGE_TITLE = "Kick It! | Login";
      include "./includes/metadata.php";
    ?>

    <script src="https://kit.fontawesome.com/2cbf705ee6.js"
crossorigin="anonymous"></script>
  </head>
  <body id="login-page">
    <?php include "./includes/header.php"; ?>

    <main>
    <!--div class="form"-->
      <!-- self-processing form -->
      <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
        <h2>Login to your account</h2>
        <input class="text" type="text" name="username"
placeholder="Username" <?php if (isset($_COOKIE['cookie'])) {
                                              echo
"value='{$_COOKIE['cookie']}'";
                                            } ?> require>
        <input class="text" type="password" name="password"
placeholder="Password" require/>
        <input type="checkbox" id="remember" name="remember"
value="remember"/>
        <label for="remember">Remember me</label>

        <!-- Contains output for form validation -->
        <div class="form-message">
          <?php
          // User has submitted the login form
          if (isset($_POST["login"])) {

            // Either the username or password was left blank
            if (empty($_POST["username"]) || empty($_POST["password"])) {
              echo "You must enter a username AND password";
            }

            else {
              // Gather user supplied username and password
              $username = $_POST["username"];
              $password = $_POST["password"];

              // Select the user's profile from the DB
              $query = "SELECT * FROM project_users WHERE username =
'$username'";
              $statement = $pdo->prepare($query);
              $statement->execute([]);
              $result = $statement->fetch();

              // If the result returned a user
              if ($result != null) {
                // If provided password matches the selected users hashed
password
                if(password_verify($password, $result['password'])){
                  session_start();
                  $_SESSION["loggedin"] = true;
                  $_SESSION["username"] = $username;
                  $_SESSION["userID"] = $result['ID'];
                  header("location:mainpage.php");
                }
                else{
                  echo('Incorrect Password!');
                }
              }
              else{
                echo "User", "$username", "does not exist!";
              }
            }
          }
          ?>
        </div>
        <input type="submit" name="login" value="Log in"/>
      </form>
      <p>Forgot your password? <a href="reset.php">Reset</a></p>
      <p>Don't have an account? <a href="register.php">Register now!</a></p>
    </main>
    <?php include "./includes/footer.php"; ?>
  </body>
</html>
```

logout.php (logout page)

```
<?php
   session_start();

   // Delete all session variables
   if(session_destroy()) {
      // Redirect user to login page
      header("location: login.php");
   }
?>
```

lucky.php (I’m Feeling Lucky page)

```
<?php
  require "./includes/library.php"; // Include database connection library
  $pdo = connectDB(); // Connect to database

  session_start(); // Start $_SESSION
?>
```

```
<!DOCTYPE html>
<html lang="en-CA">
  <head>
    <?php
      $PAGE_TITLE = "Kick It! | I'm Feeling Lucky";
      include "./includes/metadata.php";
    ?>
    <script defer src="scripts/modal-view-list-item.js"></script>
    <script src="https://kit.fontawesome.com/2cbf705ee6.js"
crossorigin="anonymous"></script>
  </head>

  <body id="lucky-page">
    <?php include "./includes/header.php"; ?>

    <main>
      <?php
        $query = "SELECT * FROM project_lists WHERE visibility = 1 ORDER BY
RAND() LIMIT 1;";
        $statement = $pdo->prepare($query);
        $statement->execute([]);
        $lists = $statement->fetchAll();
      ?>

      <?php foreach ($lists as $list) : ?>
        <div class="bucket-list">
        <?php
          // Find the owner of the current list
          $query = "SELECT username FROM project_users WHERE ID = ?";
          $statement = $pdo->prepare($query);
          $statement->execute([$list['userID']]);
          $listOwner = $statement->fetch();
        ?>

        <div class="list-header">
          <h2><?= $list['name']; ?> by <?= $listOwner['username'] ?></h2>
        </div>

        <ul>
          <?php
            $query = "SELECT * FROM project_list_items WHERE listID = ?;";
            $statement = $pdo->prepare($query);
            $statement->execute([$list['ID']]);
            $itemsInList = $statement->fetchAll();
          ?>
          <?php foreach ($itemsInList as $item) : ?>
            <!-- HTML for displaying list item -->
            <li>
              <div class="list-item">

                <?php
                  // Check to see if there is a date for completion.
                  // If there is, show the item has been completed.
                  if ($item['dateCompleted'] != NULL) {
                    echo '<i class="fa fa-check-square"></i> ';
                  } else {
                    echo '<i class="far fa-square"></i> ';
                  }
                ?>
                <span id="list-item-main"> <?= $item['name']; ?></span>

                <div class="list-item-buttons list-item-view-button">
                  View Details <i class="fas fa-info-circle"></i>
                </div> <!-- end list-item-buttons -->
              </div> <!-- end list-item -->

              <!-- Modal to view list item -->
              <div class="modal-view-list-item modal">
                  <div class="modal-content">
                    <span class="close-view-list-item modal-close">&times;</span>

                    <h3><?=$item['name'];?> by: <?=$listOwner['username'];?></h3>

                    <div>
                      <h4>Date Created: </h4>
                      <p><?=$item['date']?></p>
                    </div>

                    <div>
                      <h4>Date Completed: </h4>
                      <?php
                        // Check to see if there is a date for completion.
                        // If there is, show the date and time it was marked as being complete.
                        if ($item['dateCompleted'] != NULL) {
                          echo '<p>';
                          echo ($item['dateCompleted']);
                          echo '</p>';
                        } else {
                          echo '<p>This item is not yet complete.</p>';
                        }
                      ?>
                    </div>

                    <div>
                      <h4>Description:</h4>
                      <p><?=$item['description'];?></p>
                    </div>

                    <div>
                      <h4>Image:</h4>
                      <?php
                        // Check to see if there is a date for completion.
                        // If there is, show the date and time it was
marked as being complete.
                        if ($item['imagepath'] != NULL) {
                          echo '<img src="../../../www_data/';
                          echo ($item['imagepath']);
                          echo '" alt="Image uploaded by user">';
                        } else {
                          echo '<p>No image was provided.</p>';
                        }
                      ?>
                    </div>
                  </div> <!-- end modal-content -->
                </div> <!-- end modal-view-list-item modal -->

            </li>
          <?php endforeach ?>
          </ul>
        </div>
      <?php endforeach ?>

      <button name="feeling-lucky-btn" type="submit"
onclick="window.location.href='./lucky.php'">I'm Feeling Lucky</button>

    </main>
    <?php include "./includes/footer.php"; ?>
  </body>
</html>
```

mainpage.php (personal list management page)

```
<?php
session_start();

require "./includes/library.php";
```

```
$pdo = connectDB();

// Logged in
if ($_SESSION['loggedin']) {
  $username = $_SESSION['username'];
  $userID = $_SESSION['userID'];
}
// Not logged in, redirect to login page
else {
  header("Location:login.php");
}
?>

<!DOCTYPE html>
<html lang="en" dir="ltr">

<head>
  <!-- Title with Username -->
  <?php
  $PAGE_TITLE = "Kick It! | $username's Lists";
  include "./includes/metadata.php";
  ?>

  <!-- JS Scipts for all Modals  -->
  <script defer src="scripts/modal-new-list.js"></script>
  <script defer src="scripts/modal-new-list-item.js"></script>
  <script defer src="scripts/modal-visibility.js"></script>
  <script defer src="scripts/modal-view-list-item.js"></script>
  <script defer src="scripts/modal-delete-list.js"></script>
  <script defer src="scripts/modal-view-instructions.js"></script>
  <script defer src="scripts/list-selector.js"></script>
  <script src="https://kit.fontawesome.com/2cbf705ee6.js"
crossorigin="anonymous"></script>
</head>

<body>
  <?php include "./includes/header.php"; ?>

  <!-- User Option Navbar -->
  <aside id="user-options">
    <h3>Options:</h3>
    <ul>
      <li id="view-instructions-button"><i class="fas fa-life-ring"></i>
View Instructions</li>
      <li id="modal-button-new-list"><i class="fas fa-plus "></i> Create
List</li>
      <li id="modal-button-new-list-item"><i class="fas fa-plus "></i> Add
```

```
item to a list</li>
      <li id="modal-button-visibility"><i class="fas fa-users "></i> Change
list visibility</li>
      <li id="modal-button-delete-list"><i class="fas fa-trash-alt "></i>
Delete a list </li>
    </ul>
  </aside>

  <!-- Modals -->
  <!-- Modal to create new list -->
  <div id="modal-new-list" class="modal">
    <div class="modal-content">
      <span class="modal-close-new-list modal-close">&times;</span>
      <form method="post">

        <!-- list name -->
        <div>
          <label for="list-name">New list name:</label>
          <input class="text" type="text" name="list-name" placeholder="Kick
the Bucket">
        </div>

        <!-- privacy -->
        <div>
          <label for="list-privacy">Select Visibility:</label>
          <div>
            <select id="list-privacy" name="list-privacy">
              <option value="0">Private</option>
              <option value="1">Public</option>
            </select>
          </div>
        </div>

        <input type="submit" name="add-list" value="Create List">
      </form>
    </div>
  </div>

  <!-- PHP for Create List  -->
  <?php
  if (isset($_POST['add-list'])) {
    $listname = $_POST["list-name"];
    $listprivacy = $_POST["list-privacy"];
    $query = "INSERT INTO project_lists(userID, name, visibility)
VALUES('$userID','$listname', $listprivacy);";
    $statement = $pdo->prepare($query);
    if ($statement->execute([])) {
      echo ('New List Added');
    } else {
      echo ('Invalid Characters');
    }
  }
  ?>

  <!-- Gather bucket lists -->
  <?php
  // Gather lists owned by user
  $query = "SELECT * FROM project_lists WHERE userID = '$userID'";
  $statement = $pdo->prepare($query);
  $statement->execute([]);
  $lists = $statement->fetchAll();
  ?>

  <!-- Modal to add items to the list -->
  <div id="modal-new-list-item" class="modal">
    <div class="modal-content">
      <span class="modal-close modal-close-new-list-item">&times;</span>
      <form method="post" enctype="multipart/form-data">

        <!-- Select which list the user owns and would like to modify -->
        <div>
          <label for="list-name">Select List:</label>
          <div>
            <select id="list-name" name="list-name">
              <?php foreach ($lists as $list) : ?>
                <option value="<?= $list['ID'] ?>"><?= $list['name'] ?
></option>
              <?php endforeach ?>
            </select>
          </div>
        </div>

        <!-- new list item name -->
        <div>
          <label for="item-name">New Item name:</label>
          <input class="text" type="text" name="item-name" placeholder="Item
Name">
        </div>

        <!-- new list item description -->
        <div>
          <label for="item-desc">Item Description:</label>
          <textarea class="text" name="item-desc" placeholder="Item
```

```
Description"></textarea>
        </div>

        <!-- new list item image -->
        <div>
          <label for="item-desc">Upload Image:</label>
          <input type="file" name="file" class="file-upload-field" />
        </div>

        <input type="submit" name="add-item" value="Add List Item">
      </form>
    </div> <!-- end modal-content -->
  </div> <!-- end modal-new-list-item -->

  <!-- Php Script to add list item -->
  <?php
  if (isset($_POST['add-item'])) {
    // Gather information from POST array
    $itemname = $_POST["item-name"];
    $list_id = $_POST["list-name"];
    $item_desc = $_POST["item-desc"];

    $file_name = uniqid() . "-" . time();
    $extension  = pathinfo($_FILES["file"]["name"], PATHINFO_EXTENSION);
    $basename   = $file_name . '.' . $extension;
    $file_type = $_FILES['file']['type'];
    $file_size = $_FILES['file']['size'];
    $file_temp_loc = $_FILES['file']['tmp_name'];
    $file_store = "/home/nikhilpaiganesh/public_html/www_data/" . $basename;
    if (move_uploaded_file($file_temp_loc, $file_store)) {
      echo 'Image Uploaded Sucessfully';
    }


    // Create the date string to be inserted
    date_default_timezone_set("America/Toronto");
    $date = date("d/m/Y") . " at " . date("h:ia");

    $query = "INSERT INTO project_list_items(listID, name, description,
date, imagepath)
VALUES('$list_id','$itemname','$item_desc','$date','$basename');";
    $statement = $pdo->prepare($query);
    if ($statement->execute([])) {
      echo 'New Item added';
    } else {
      echo "No Special Character Allowed";
    }
  }
  ?>


  <!-- Change list visibility modal -->
  <div id="modal-visibility" class="modal">
    <div class="modal-content">
      <span class="modal-close-visibility modal-close">&times;</span>
      <form method="post">
        <label for="list-name1">Select List:</label>
        <div>
          <select id="list-name1" name="list-name1">
            <?php foreach ($lists as $list) : ?>
              <option value="<?= $list['ID'] ?>"><?= $list['name'] ?
></option>
            <?php endforeach ?>
          </select>
        </div>

        <!-- update list selection-->
        <div>
          <label for="list-privacy1">Select Visibility:</label>
          <div>
            <select id="list-privacy1" name="list-privacy1">
              <option value="0">Private</option>
              <option value="1">Public</option>
            </select>
          </div>
        </div>

        <!-- update list button -->
        <input type="submit" name="public" value="Update">
      </form>
    </div> <!-- end modal-content -->
  </div> <!-- end modal-visibility -->

  <!-- Php Script to make list public -->
  <?php
  if (isset($_POST['public'])) {
    $list_id1 = $_POST["list-name1"];
    $list_privacy1 = $_POST["list-privacy1"];
    $query = "UPDATE project_lists SET visibility = '$list_privacy1' WHERE
ID = '$list_id1';";
    $statement = $pdo->prepare($query);
    if ($statement->execute([])) {
      echo 'Visibility updated';
    } else {
      echo "Couldn't update visibility";
    }
  }
  ?>

  <!-- Delete List Modal -->
  <div id="modal-delete-list" class="modal">
    <div class="modal-content">
      <span class="modal-close-delete-list modal-close">&times;</span>
      <form method="post" action="<?= $_SERVER['PHP_SELF']; ?>">

        <div>
          <label for="list-name1">Select List:</label>
          <div>
            <select id="list-name1" name="list-name1">
              <?php foreach ($lists as $list) : ?>
                <option value="<?= $list['ID'] ?>"><?= $list['name'] ?
></option>
              <?php endforeach ?>
            </select>
          </div>
        </div>

        <!--Delete List Submit -->
        <input type="submit" onclick="return confirm('Are you sure you want
to delete this list?');" name="delete-list" value="Delete List">
      </form>
    </div> <!-- end modal-content -->
  </div> <!-- end modal-delete-list -->

  <!-- PHP for Delete List -->
  <?php
  if (isset($_POST['delete-list'])) {
    $listID = $_POST["list-name1"];

    //Deletes a list with a given list ID
    $query = "DELETE FROM project_lists WHERE ID = $listID";
    $statement = $pdo->prepare($query);

    // Delete the list
    if($statement->execute([])){

      // Deletes all items which belong to that list ID
      $query = "DELETE FROM project_list_items WHERE listID = $listID";
      $statement = $pdo->prepare($query);
      if($statement->execute([])){
        echo("List and items deleted.");
      }
      else{
        echo("No list items to delete");
      }
    }
    else{
      echo("Couldn't delete list");
    }
  }
  ?>

  <main>

  <h2>Welcome to your bucketlist manager, <?= $username ?></h2>
    <p>
      Here on the bucketlist management page, you can create new lists or
items, modify some of their variables/inputs, mark an item as complete,
remove old items or lists, and more.
    </p>

    <?php
      $query = "SELECT * FROM project_lists WHERE userID = ?";
      $statement = $pdo->prepare($query);
      $statement->execute([$userID]);
      $lists = $statement->fetchAll();
    ?>

    <?php foreach ($lists as $list) : ?>
      <div class="bucket-list">

        <?php
        // Find the owner of the current list
        $query = "SELECT username FROM project_users WHERE ID = ?";
        $statement = $pdo->prepare($query);
        $statement->execute([$list['userID']]);
        $listOwner = $statement->fetch();
        ?>

        <div class="list-header">
          <h2><?= $list['name']; ?> by <?= $listOwner['username'] ?>
            <?php
            if ($list['visibility'] == 0) {
              echo '<i class="fas fa-low-vision"></i></h2>';
            } else {
              echo '<i class="fas fa-eye"></i></h2>';
            }
            ?>
        </div>

        <ul>
          <?php
          $query = "SELECT * FROM project_list_items WHERE listID = ?;";
          $statement = $pdo->prepare($query);
          $statement->execute([$list['ID']]);
          $itemsInList = $statement->fetchAll();
          ?>

          <?php foreach ($itemsInList as $item) : ?>
            <!-- HTML for displaying list item -->
            <li>
              <div class="list-item">
                <?php
                  // Check to see if there is a date for completion.
                  // If there is, show the item has been completed.
                  if ($item['dateCompleted'] != NULL) {
                    echo '<i class="fa fa-check-square"></i> ';
                  } else {
                    echo '<i class="far fa-square"></i> ';
                  }
                ?>

                <a href="" id="list-item-main"> <?= $item['name']; ?></a>

                <div class="list-item-buttons list-item-view-button">
                  View/Edit Details <i class="fas fa-info-circle"></i>
                </div> <!-- end list-item-buttons -->
              </div> <!-- end list-item -->

              <!-- Modal to view list item -->
              <div class="modal-view-list-item modal">
                <div class="modal-content">
                  <span class="close-view-list-item modal-
close">&times;</span>
                  <form method="post" enctype="multipart/form-data">
                    <div>
                      <label for="item-name-u">New Item name:</label>
                      <input class="text" type="text" name="item-name-u"
value="<?= $item['name']; ?>">
                    </div>

                    <div>
                      <h4>Date Created [Cannot modify]: </h4>
                      <p><?=$item['date']?></p>
                    </div>
                    <div>
                      <label for="dateCompleted">Set Date Completed:</label>
                      <input type="date" name="dateCompleted" value="<?=
$item['dateCompleted']; ?>" min="<?= $item['date']; ?>">
                    </div>

                    <div>
                      <label for="item-desc-update">Update Item
Description:</label>
                      <textarea class="text" name="item-desc-update"><?=
$item['description']; ?></textarea>
                    </div>

                    <div>
                      <label for="item-desc">Update Image:</label>
                      <input type="file" name="file-update" class="file-
upload-field" />
                    </div>

                    <div>
                      <h4>Image:</h4>
                      <?php
                        // Check to see if there is a date for completion.
                        // If there is, show the date and time it was
marked as being complete.
                        if ($item['imagepath'] != NULL) {
                          echo '<img src="../../../www_data/';
                          echo ($item['imagepath']);
                          echo '" alt="Image uploaded by user">';
                        } else {
                          echo '<p>No image was provided.</p>';
                        }
                      ?>
                    </div>

                    <input type="submit" name="update-item" value="Update">
                    <input type="submit" name="delete-item"  id="delete-
account" value="Delete This Item" onclick="return confirm('Are you sure you
want to delete this item?');">

                    <input class="number" type="number" name="item-id-u"
value="<?= $item['ID']; ?>" readonly>
                  </form>

                </div> <!-- end modal-content -->
              </div> <!-- end list-item -->
              <!-- Script to update item -->
              <?php
              if (isset($_POST['update-item'])) {
                // Gather information from POST array
                $itemnameu = $_POST["item-name-u"];
                $item_descu = $_POST["item-desc-update"];
                $item_dateComp = $_POST["dateCompleted"];
                $id = $_POST['item-id-u'];

                $file_name_u = uniqid() . "-" . time();
                $extension_u  = pathinfo($_FILES["file-update"]["name"],
PATHINFO_EXTENSION);
                $basename_u = $file_name_u . '.' . $extension_u;
                $file_type_u = $_FILES['file-update']['type'];
                $file_size_u = $_FILES['file-update']['size'];
                $file_temp_loc_u = $_FILES['file-update']['tmp_name'];
                $file_store_u =
"/home/nikhilpaiganesh/public_html/www_data/" . $basename_u;

                $file_upload = $file_name_u . '.' . $extension_u;
                if (move_uploaded_file($file_temp_loc_u, $file_store_u)) {
                  echo 'Image Uploaded Sucessfully';
                }

                if(isset($_POST['file-update'])){
                  $query_uf = "UPDATE project_list_items SET name =
imagepath = '$basename_u' WHERE ID = $id;";
                  $statement_uf = $pdo->prepare($query_uf);
                  if ($statement_u->execute([])) {
                    echo 'Image uploaded';
                  } else {
                    echo "Failed to upload image to", $file_store_u;
                  }
                }


                $query_u = "UPDATE project_list_items SET name =
'$itemnameu', description = '$item_descu', dateCompleted= '$item_dateComp'
WHERE ID = $id;";
                $statement_u = $pdo->prepare($query_u);
                if ($statement_u->execute([])) {
                  echo 'New Item added';
                } else {
                  echo "No Special Character Allowed";
                }
              }
              if(isset($_POST['delete-item'])){
                $id = $_POST['item-id-u'];
                $delete = $pdo->prepare("DELETE FROM project_list_items
WHERE id = $id");
                 $delete->execute();
              }
              ?>

            </li>
          <?php endforeach ?>
        </ul>
      </div> <!-- end bucket-list -->
    <?php endforeach ?>

    <div id="modal-view-instructions" class="modal">
      <div class="modal-content">
        <span class="close-view-instructions modal-close">&times;</span>
        <div class="mainpage-instructions">
          <h3>Instructions</h3>
            <div>
              <h4>How do I create a new bucket list?</h4>
              <p>
                You can create a new bucket list by navigating to the right
side of the screen and clicking on "Create New List", located under the
"Options" menu.
                A pop-up window will appear, and you will be able to input
the title and set the visibility of your bucket list.
                There are two options for visibility, public (<i class="fas
fa-eye"></i>), and private (<i class="fas fa-low-vision"></i>).
                If you choose to create a public bucket list, it will appear
on the public-list page, potentially appear on the im-feeling-lucky page,
and can be discovered if another user searches for a keyword that is an item
of a bucket list.
                If you set the visibility of your bucket list to private, it
will only be viewable by you and cannot be seen by anyone else, including by
search.
                Regardless of the visibility option you choose, your bucket
list will appear on this page.
              </p>
            </div>
            <div>
              <h4>I made a bucket list, how do I add items to it?</h4>
              <p>
                Just like creating a bucket list, you can create a new list
item by navigating to the right side of the screen and clicking on "Create
New List Item",
                located under the "Options" menu. A pop-up window will
appear again, this time with more options.
                You can set the title or name of your list item, set the day
it was completed (if you already have completed it and want to add it to
give ideas to others),
                add a description,  and upload an image.
                You do not have to fill in the date completed, description,
or upload an image. It's entirely optional.
              </p>
            </div>
            <div>
              <h4>I completed an item on my bucket list, how do I mark it as
complete?</h4>
              <p>
                You can mark an item as being complete by hovering over the
item, and clicking on "View/Edit Details <i class="fas fa-info-circle"></i>"
button on the right side.
                In the popup window, enter the date you completed the item
on, and click "Update".
                This will automatically update the item, and place a
checkmark in the box on the left of the item's title.
                <strong>Please note that you may need to refresh the page
for this to take effect!</strong>
              </p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </main>
  <?php include "./includes/footer.php"; ?>
</body>

</html>
```

public-lists.php (displays all public lists)

```
<?php
require "./includes/library.php"; // Include database connection library
$pdo = connectDB(); // Connect to database

session_start(); // Start $_SESSION
?>

<!DOCTYPE html>
<html lang="en-CA">
```

```
  <head>
    <?php
    $PAGE_TITLE = "Kick It! | Public Lists";
    include "./includes/metadata.php";
    ?>

    <script defer src="scripts/modal-view-list-item.js"></script>
    <script src="https://kit.fontawesome.com/2cbf705ee6.js"
crossorigin="anonymous"></script>
  </head>

  <body id="public-lists-page">
    <?php include "./includes/header.php"; ?>

    <main>
       <?php
        $query = "SELECT * FROM project_lists WHERE visibility = 1";
        $statement = $pdo->prepare($query);
        $statement->execute([]);
        $lists = $statement->fetchAll();
      ?>

      <?php foreach ($lists as $list) : ?>
        <div class="bucket-list">

          <?php
            // Find the owner of the current list
            $query = "SELECT username FROM project_users WHERE ID = ?";
            $statement = $pdo->prepare($query);
            $statement->execute([$list['userID']]);
            $listOwner = $statement->fetch();
          ?>

        <div class="list-header">
          <h2><?= $list['name']; ?> by <?= $listOwner['username'] ?></h2>
        </div>

          <ul>
            <?php
              // Find all items in that list
              $query = "SELECT * FROM project_list_items WHERE listID = ?;";
              $statement = $pdo->prepare($query);
              $statement->execute([$list['ID']]);
              $itemsInList = $statement->fetchAll();
            ?>

            <?php foreach ($itemsInList as $item) : ?>
```

```
              <!-- HTML for displaying list item -->
              <li>
                <div class="list-item">
                  <?php
                    // Check to see if there is a date for completion.
                    // If there is, show the item has been completed.
                    if ($item['dateCompleted'] != NULL) {
                      echo '<i class="fa fa-check-square"></i> ';
                    } else {
                      echo '<i class="far fa-square"></i> ';
                    }
                  ?>
                  <a href="" id="list-item-main"> <?= $item['name']; ?></a>

                  <div class="list-item-buttons list-item-view-button">
                    View Details <i class="fas fa-info-circle"></i>
                  </div> <!-- end list-item-buttons -->
                </div> <!-- end list-item -->

                <!-- Modal to view list item -->
                <div class="modal-view-list-item modal">
                  <div class="modal-content">
                    <span class="close-view-list-item modal-close">&times;</
span>
                    <h3><?=$item['name'];?> by: <?=$listOwner['username'];?
></h3>

                    <div>
                      <h4>Date Created: </h4>
                      <p><?=$item['date']?></p>
                    </div>

                    <div>
                      <h4>Date Completed: </h4>
                      <?php
                        // Check to see if there is a date for completion.
                        // If there is, show the date and time it was
marked as being complete.
                        if ($item['dateCompleted'] != NULL) {
                          echo '<p>';
                          echo ($item['dateCompleted']);
                          echo '</p>';
                        } else {
                          echo '<p>This item is not yet complete.</p>';
                        }
                      ?>
                    </div>
                    <div>
                      <h4>Description:</h4>
                      <p><?=$item['description'];?></p>
                    </div>

                    <div>
                      <h4>Image:</h4>
                      <?php
                        // Check to see if there is a date for completion.
                        // If there is, show the date and time it was
marked as being complete.
                        if ($item['imagepath'] != NULL) {
                          echo '<img src="../../../www_data/';
                          echo ($item['imagepath']);
                          echo '" alt="Image uploaded by user">';
                        } else {
                          echo '<p>No image was provided.</p>';
                        }
                      ?>
                    </div>
                  </div> <!-- end modal-content -->
                </div> <!-- end modal-view-list-item modal -->

              </li>
            <?php endforeach ?>
          </ul>
        </div> <!-- end bucket-list -->
      <?php endforeach ?>
    </main>
    <?php include "./includes/footer.php"; ?>
  </body>
</html>
```

register.php (registration page)

```
<?php
  require "./includes/library.php"; // Include PDO DB library
  $pdo = connectDB(); // Connect to DB
?>

<!DOCTYPE html>
<html lang="en-CA">
  <head>
    <?php
      $PAGE_TITLE = "Kick It! | Register New Account";
      include "./includes/metadata.php";
    ?>

    <script src="https://kit.fontawesome.com/2cbf705ee6.js"
crossorigin="anonymous"></script>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></
script>
  </head>

  <script src="./scripts/register.js"></script> <!-- should this not go in
the <head> ? (H)-->

  <body id="register-page">
    <?php include "./includes/header.php"; ?>

    <main>
        <form method="POST" name="form" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
          <h2>Register</h2>
          <input class="text" id="username" type="text" name="username"
placeholder="Username" required="">
          <div id="user-feedback" class="form-message"></div>
          <input class="text" id="email" type="email" name="email"
placeholder="Email" required="">
          <input class="text" id="password" type="password" name="password"
placeholder="Password" onKeyUp="checkPasswordStrength();" required="">

          <div id="password-strength-status" class="form-message"></div>
          <input class="text" id="confirm_password" type="password"
name="confirm_password" placeholder="Confirm Password" required="">

          <div class="form-message">
            <?php
            // Wait for form submission
            if (isset($_POST['submit'])) {

              // If any fields have been left empty
              if (empty($_POST["username"]) || empty($_POST["password"]) ||
empty($_POST["email"]) || empty($_POST["confirm_password"])) {
                echo ('All Fields should be filled!');
              }

              // All fields have been filled
              else {

                // Gather user supplied form data
                $username = $_POST["username"];
                $email = $_POST["email"];
                $password = $_POST["password"];
                $confirm = $_POST["confirm_password"];

                // Perform a search to determine if a user with that
username already exists
                $query = "SELECT * FROM project_users WHERE
username='$username'";
                $statement = $pdo->prepare($query);
                $statement->execute([]);
                $result = $statement->fetch();

                // No user exists with username
                if(($result == null)){
                  // Both user supplied passwords match
                  if (($confirm == $password)){

                    // Hash the users password
                    $password = password_hash($password, PASSWORD_DEFAULT);

                    // Insert the user into the database
                    $query = "INSERT INTO project_users(username, password,
email) VALUES ('$username', '$password', '$email')";
                    $statement = $pdo->prepare($query);

                    // Query successful
                    if ($statement->execute([])) {
                      // Redirect to login
                      echo ('Registration Complete');
                      header('Location: login.php');
                    }
                    else{
                      echo('Unexpected Error: Please try again');
                    }
                  }
                  // Passwords do not match
                  else {
                    echo ('Passwords do not match');
                  }
                }
                // Username has been taken
                else{
                  echo('Account with that username exists');
                }
              }
            }
            ?>
```

```
          </div> <!-- end div form-message -->

          <input type="submit" id="submit" name="submit" value="Create a New
Account"/>
        </form>
        <p>Already have an Account? <a href="login.php">Login Now!</a></p>

    </main>
    <?php include "./includes/footer.php"; ?>
  </body>
</html>
```

reset.php (reset password)

```
<!DOCTYPE html>
<html lang="en-CA">
  <head>
    <?php
      $PAGE_TITLE = "Bucketlist | Password Recovery";
      include "./includes/metadata.php";
    ?>

    <script src="https://kit.fontawesome.com/2cbf705ee6.js"
crossorigin="anonymous"></script>
  </head>


  <body id="password-recovery-page">
    <?php include("./includes/header.php"); ?>

    <main>
      <div class="form">
        <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
          <h2> Recover your Password </h2>
          <input type="email" name="email" placeholder="Enter your Email"
require>
          <input type="submit" name="forgot-password" value="Send Recovery
Email">
          <?php
            require "./includes/library.php";
            $pdo = connectDB();

            if (isset($_POST['forgot-password'])) {
              $token = rand(100000, 999999);
              $email = $_POST['email'];
              $query = $pdo->prepare("SELECT COUNT(ID) FROM project_users
WHERE email ='$email'");
```

```
              $query->execute();
              $row = $query->fetchColumn();
              if ($row > 0) {
                $sql = $pdo->prepare("UPDATE project_users SET token = '$token' WHERE email = '$email'");
                if ($sql->execute()) {
                $subject = "Your Kick It! Password reset request";
                $headers = "From: Kick It! Password Reset <do-not-reply@kickit.com>\r\n";
                $headers .= "Content-type: text/html\r\n";
                $message = '<h1> You have requested a password Reset for your Kick It! Account </h1> <h2> Your One-time-password is ' . $token . '.</h2>';

                mail($email, $subject, $message, $headers);

                header('location:change.php');
              } else {
                echo ("Something Went Wrong!");
              }
            }else{
              echo ("Email not found, Enter the email registered with your account");
            }
          }
          ?>
        </form>
      </div> <!-- end form -->
    </main>
    <?php include "./includes/footer.php"; ?>
  </body>
</html>
```

validate.php (check if username is taken or not)

```
<?php
    require "./includes/library.php";
    $pdo = connectDB();

    error_reporting(0);

    $value = false;
    $username = $_POST['username'];

    $query = "SELECT * FROM project_users WHERE username = ?";
    $statement = $pdo->prepare($query);
    $statement->execute([$username]);
    $results = $statement->fetch();

    if($results == TRUE){
        echo('Username Taken');
    } else {
        echo('Username Available');
    }
?>
```

SOURCE CODE [JAVASCRIPT]:

modal-delete-list.js (modal to delete a list)

```
var modal_delete_list = document.getElementById("modal-delete-list");
var modalButton_delete_list = document.getElementById("modal-button-delete-list");
var modalClose_delete_list = document.getElementsByClassName("modal-close-delete-list")[0];

modalButton_delete_list.onclick = function(){
  modal_delete_list.style.display = "block";
}

modalClose_delete_list.onclick = function(){
  modal_delete_list.style.display = "none";
}

window.onclick = function(event){
  if (event.target == modal_delete_list){
    modal_delete_list.style.display = "none";
    }
}
```

modal-new-item.js (modal to add a new list item)

```
var modal_new_list_item = document.getElementById("modal-new-list-item");
var modalButton_new_list_item = document.getElementById("modal-button-new-list-item");
var modalClose_new_list_item = document.getElementsByClassName("modal-
close-new-list-item")[0];

modalButton_new_list_item.onclick = function(){
  modal_new_list_item.style.display = "block";
}

modalClose_new_list_item.onclick = function(){
  modal_new_list_item.style.display = "none";
}

window.onclick = function(event){
  if (event.target == modal_new_list_item){
    modal_new_list_item.style.display = "none";
  }
}
```

modal-new-list.js (modal to create a new list)

```
var modal_new_list = document.getElementById("modal-new-list");
var modalButton_new_list = document.getElementById("modal-button-new-list");
var modalClose_new_list = document.getElementsByClassName("modal-close-new-list")[0];

modalButton_new_list.onclick = function(){
  modal_new_list.style.display = "block";
}

modalClose_new_list.onclick = function(){
  modal_new_list.style.display = "none";
}

window.onclick = function(event){
  if (event.target == modal_new_list){
    modal_new_list.style.display = "none";
  }
}
```

modal-view-instructions.js (modal to view instructions)

```
// View instructions buttons
var modal_view_instructions = document.getElementById("modal-view-
instructions");
var modalButton_view_instructions = document.getElementById("view-
instructions-button");
var modalClose_view_instructions =
document.getElementsByClassName("close-view-instructions");

// Attach a click event listener to the button
modalButton_view_instructions.addEventListener('click', function (i) {
    // Display the modal when clicked
    modal_view_instructions.style.display = "block";
}.bind(null, i));

modalClose_view_instructions[0].addEventListener('click', function (i) {
    modal_view_instructions.style.display = "none";
}.bind(null, i));
```

modal-view-item-list.js (modal to view item lists)

```
// View List Item buttons
var modal_view_list_item = document.getElementsByClassName("modal-view-
list-item");
var modalButton_view_list_item = document.getElementsByClassName("list-
item-view-button");
var modalClose_view_list_item = document.getElementsByClassName("close-
view-list-item");

// For each view details button
for (var i = 0; i < modalButton_view_list_item.length; i++) {

    // Attach a click event listener to the button
    modalButton_view_list_item[i].addEventListener('click', function (i)
{
        // Display the modal when clicked
        modal_view_list_item[i].style.display = "block";
    }.bind(null, i));

    modalClose_view_list_item[i].addEventListener('click', function (i)
{
        modal_view_list_item[i].style.display = "none";
    }.bind(null, i));
}
```

modal-visibility.js (modal to adjust list visibility)

```
var modal_visibility = document.getElementById("modal-visibility");
var modalButton_visibility = document.getElementById("modal-button-visibility");
var modalClose_visibility = document.getElementsByClassName("modal-close-visibility")[0];

modalButton_visibility.onclick = function(){
  modal_visibility.style.display = "block";
}

modalClose_visibility.onclick = function(){
  modal_visibility.style.display = "none";
}

window.onclick = function(event){
  if (event.target == modal_visibility){
    modal_visibility.style.display = "none";
  }
}
```

register.js (registration)

```
$(document).ready(function() {
    $('#user-feedback').load('validate.php').show();
    $('#username').keyup(function(){
        $.post('validate.php', { username: form.username.value },
        function(result){
            $('#user-feedback').html(result).show();
        })
    });
});


// JQuery Plugin
// Plugin: https://phppot.com/jquery/jquery-password-strength-checker/

function checkPasswordStrength() {
    var number = /([0-9])/;
    var alphabets = /([a-zA-Z])/;
    var special_characters = /([~,!,@,#,$,%,^,&,*,-,_,+,=,?,>,<])/;

    if($('#password').val().length<8) {
        $('#password-strength-status').removeClass();
        $('#password-strength-status').addClass('weak-password');
        $('#password-strength-status').html("Weak (should be atleast 8 
characters.)");
    } else {
        if($('#password').val().match(number) && $
('#password').val().match(alphabets) && $
('#password').val().match(special_characters)) {
            $('#password-strength-status').removeClass();
            $('#password-strength-status').addClass('strong-password');
            $('#password-strength-status').html("Strong");
        } else {
            $('#password-strength-status').removeClass();
            $('#password-strength-status').addClass('medium-password');
            $('#password-strength-status').html("Medium (should include 
a combinations of 8 characters, numbers and special characters.)");
        }
    }}
```

delete-list.js (confirmation to delete a list)

```
function confirmDeleteList(id)
{
    var ok = confirm("Are you sure you would like to delete this list? 
This action is irreversible!");
    if (ok)
    {
        if (window.XMLHttpRequest)
        {// code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp = new XMLHttpRequest();
        }
        else
        {// code for IE6, IE5
            xmlhttp = new ActiveXObject('Microsoft.XMLHTTP');
        }

        xmlhttp.onreadystatechange = function()
        {
            if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
            {
                 window.location = "delete-list.php";  // self page
            }
        }
        xmlhttp.open('GET', 'delete-list.php');
        xmlhttp.send();
    }
    }
```

SOURCE CODE [CSS]:

header.css (exclusively for nav bar)

```css
/* Navigation Bar */
/* Code partially sourced from:
https://www.w3schools.com/css/css_dropdowns.asp */
nav {
  overflow: hidden;
  display: flex;
  justify-content: flex-end;
  flex-direction: row;
  align-items: baseline;
  width: auto;
  margin: 1rem 2rem;
}

/* Style anchors */
nav a {
  font-size: 0.9rem;
  color: #000;
  text-align: center;
  padding: 1rem 1.1rem;
  text-decoration: none;
}

/* Float logo left */
.nav-logo{
  margin-left: 2rem;
  float: left;
}

/* Unique style for search bar */
.nav-search-bar {
  display: inline-block;
}

input[id="nav-search-string"] {
  width: auto;
  height: auto;

  padding: 0.4rem;
  background-color: #F7FFF6;
  border: 1px solid #000;
  border-radius: 0.4rem;
}
```

```css
/* Button styling */
.nav-button-login {
  border: 1px solid #E63946;
  border-radius: 0.4rem;
  margin-left: 1rem;
  background-color: #E63946;
  color: #F1FAEE;
}

.nav-button-login:hover {
  border: 1px solid #E63946;
  border-radius: 0.4rem;
  background-color: #E63946;
  color: #F1FAEE;
}

.nav-dropdown-lists, .nav-dropdown-account {
  overflow: hidden;
}

.nav-dropdown-lists .nav-button-lists, .nav-dropdown-account .nav-button-account {
  font-size: 0.9rem;
  border: none;
  outline: none;
  color: #000;
  padding: 1rem 1.1rem;
  background-color: inherit;
  font-family: inherit;
  margin: 0;
}

nav a:hover, .nav-dropdown-lists:hover .nav-button-lists, .nav-dropdown-account:hover .nav-button-account {
  cursor: pointer; /* For consistency between anchors and dropdown buttons */
  transition: ease-in 0.15s;
  text-decoration: underline;
}

.nav-dropdown-lists-content, .nav-dropdown-account-content {
  display: none;
  position: absolute;
  background-color: #F1FAEE;
  min-width: 10rem;
  box-shadow: 0 0.8rem 0.8rem 0 rgba(0,0,0,0.2);
  z-index: 1;
}

.nav-dropdown-lists-content a, .nav-dropdown-account-content a {
  color: black;
  padding: 0.8rem 1rem;
  text-decoration: none;
  display: block;
  text-align: left;
}

.nav-dropdown-lists-content a:hover, .nav-dropdown-account-content a:hover {
  background-color: #E63946;
  color: #F1FAEE;
}

.nav-dropdown-lists:hover .nav-dropdown-lists-content, .nav-dropdown-
account:hover .nav-dropdown-account-content {
  display: block;
}
```

reset.css (clear cache)

```
/* http://meyerweb.com/eric/tools/css/reset/
   v2.0 | 20110126
   License: none (public domain)
*/

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
body {
    line-height: 1;
}
ol, ul {
    list-style: none;
}
blockquote, q {
    quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
    content: '';
    content: none;
}
table {
    border-collapse: collapse;
    border-spacing: 0;
}
```

style.css (master stylesheet)

```
@import url('https://fonts.googleapis.com/css2?
family=Lato:wght@300;400;700&display=swap');

:root{
  background-color: #F1FAEE;
}

body {
  font-family: 'Lato', sans-serif;
  background-size: cover;
  background-color: #F1FAEE;
}

main {
  width: 45%;
  margin: auto;
  text-align: center;
}
```

```css
/* Generic headers */
h1 {
  font-size: 2rem;
  text-align: left;
  margin-left: 0.5rem;
  margin-top: 0.5rem;
  color: #000;
  font-weight: 100;
  font-family: 'Lato', sans-serif;
}

h2 {
  font-size: 1.5rem;
  text-align: center;
  color: #000;
  font-weight: bold;
  font-family: 'Lato', sans-serif;
}

h3 {
  font-weight: bold;
  text-decoration: underline;
  font-size: 1.4rem;
}

/* Non-headers */
p {
  font-size: 1rem;
  color: #000;
  text-align: center;
}

p a {
  color: #E63946;
  text-decoration: underline;
  font-weight: 400;
}

/* Used on the index page & in the footer */
strong {
  font-weight: bold;
}
em {
  font-style: italic;
}
```

```css
/* Used on most pages to display list content */
li {
  text-align: left !important;
}

/* Text inputs */
input[class="text"], input[type="email"],
input[type="password"]url('https://fonts.googleapis.com/css2?
family=Lato:wght@300;400;700&display=swap');

:root{
  background-color: #F1FAEE;
}

body {
  font-family: 'Lato', sans-serif;
  background-size: cover;
  background-color: #F1FAEE;
}

main {
  width: 45%;
  margin: auto;
  text-align: center;
}

/* Generic headers */
h1 {
  font-size: 2rem;
  text-align: left;
  margin-left: 0.5rem;
  margin-top: 0.5rem;
  color: #000;
  font-weight: 100;
  font-family: 'Lato', sans-serif;
}

h2 {
  font-size: 1.5rem;
  text-align: center;
  color: #000;
  font-weight: bold;
  font-family: 'Lato', sans-serif;
}

h3 {
  font-weight: bold;
  text-decoration: underline;
  font-size: 1.4rem;
}

/* Non-headers */
p {
  font-size: 1rem;
  color: #000;
  text-align: center;
}

p a {
  color: #E63946;
  text-decoration: underline;
  font-weight: 400;
}

/* Used on the index page & in the footer */
strong {
  font-weight: bold;
}
em {
  font-style: italic;
}

/* Used on most pages to display list content */
li {
  text-align: left !important;
}

/* Text inputs */
input[class="text"], input[type="email"], input[type="password"] {
  margin: auto;
  display: flex;
  flex-direction: column;
  justify-content: center;
  margin-top: 2rem;
  margin-bottom: 2rem;
  width: 60%;
  font-size: 1em;
  color: #000000;
  font-weight: 400;
  text-align: left;
  border: 1px solid black;
  border-radius: 0.3rem;
  padding: 0.6rem;
}
```

```
::-webkit-input-placeholder {
  color: #000;
  font-weight: 300;
}

/* Button Types */
button[type="submit"], input[type="submit"] {
  font-size: 1em;
  color: #000;
  background-color: #F7FFF6;
  border: 1px solid #000000;
  border-radius: 0.3rem;
  cursor: pointer;
  padding: 0.9em;
  width: 30%;
  margin-bottom: 2rem;
  -webkit-box-shadow: 7px 7px 15px -7px rgba(0,0,0,0.75);
  -moz-box-shadow: 7px 7px 15px -7px rgba(0,0,0,0.75);
  box-shadow: 7px 7px 15px -7px rgba(0,0,0,0.75);
}

/* Effect on cursor hover */
button[type="submit"]:hover, input[type="submit"]:hover {
  transition: .5s all;
  background: #1D3557;
  color: #F1FAEE;
}

/* Form notifications */
.form-message{
  text-align: center;
  color: #000;
}

/* Global footer styling */
footer {
  margin-top: 2rem;
  display: flex;
  justify-content: center;
  align-items: flex-end;
  font-weight: 300;
}

/***********************
 *                      *
 * ==== INDEX.PHP ==== *
 *                      *
************************/

/* Searchbar */
#index-search-string {
  text-align: left;
  font-size: 1em;
  color: #000;
  font-weight: 100;
  width: 100%; /* 100% width of main element */
  display: block;
  border: 1px solid #000;
  background-color: #F7FFF6;
  border-radius: 0.5rem;
  padding: 0.8em;
  margin-bottom: 2rem;
}

/* Override default styling */
#query-search-button, #search-lucky-button {
  margin-bottom: 1rem;
}

/* Override default styling */
#index-page p {
  margin-top: 1.5rem;
}

/* Center the logo and set a max width */
#index-page .index-logo {
  text-align: center;
  width: auto;
  max-width: 37.5rem; /* equivalent to 600px */
}

/************************************
 *                                  *
 * ==== LOGIN.PHP/REGISTER.PHP ==== *
 *                                  *
*************************************/

/* Give some spacing for the login button */
input[name="login"] {
  margin-top: 2rem;
}

/**********************************
```

```
 *                                  *
 * ==== ACCOUNT-SETTINGS.PHP ==== *
 *                                  *
***********************************/
/* Fix strange spacing */
#account-settings-page h2 {
  margin-bottom: 1.5rem;
}

/* Delete Account Button */
#delete-account {
  font-size: 1em;
  color: #F1FAEE;
  background-color: #E63946;
  border: 1px solid #E63946;
  border-radius: 0.3rem;
  cursor: pointer;
  padding: 0.9em;
  width: 30%;
  -webkit-box-shadow: 7px 7px 15px -7px rgba(0,0,0,0.75);
  -moz-box-shadow: 7px 7px 15px -7px rgba(0,0,0,0.75);
  box-shadow: 7px 7px 15px -7px rgba(0,0,0,0.75);
}

/**************************
 *                      *
 * ==== MAINPAGE.PHP ==== *
 *                      *
***************************/
/* Display aside as a block floated right */
#user-options {
  display: block;
  float: right;
  margin-right: 3rem;
}

/* Option text at top */
#user-options h3 {
  text-align: center;
  padding-bottom: 0.2rem;
  color: #F1FAEE;
  background-color: #E63946;
  border: 1px solid #E63946;
  border-radius: 0.5rem;
  text-decoration: none;
}
```

```
/* Miscellaneous padding */
#user-options li {
  padding: 0.4rem;
  font-size: 0.9rem;
}

/* Make it clear it is a clickable option */
#user-options li:hover {
  cursor: pointer;
}

/* Add a small effect and colour the option icon red */
#user-options li:hover > i {
  color: #E63946;
}

.list-privacy-icon {
  float: right;
}

/*********************************************************************
 *                                                               *
 * ==== PUBLIC-LISTS.PHP / SEARCH-LISTS.PHP / FEELING-LUCKY.PHP ==== *
 *                                                               *
**********************************************************************/

/* Fix margin for I'm Feeling Lucky Button */
button[name="feeling-lucky-btn"]{
  margin-top: 2rem;
}

/* On the search-lists page, highlight the item(s) the user queried */
.selected {
  background-color: #a8dadc;
}

/* Unique styling for bucket list div */
.bucket-list {
  border: 1px solid #000;
  border-radius: 0.5rem;
  margin-top: 1rem;
  -webkit-box-shadow: 7px 7px 15px -7px rgba(0,0,0,0.75);
  -moz-box-shadow: 7px 7px 15px -7px rgba(0,0,0,0.75);
  box-shadow: 7px 7px 15px -7px rgba(0,0,0,0.75);
}

/* Visibility indicator icon for user's lists on mainpage */
```

```css
.list-header i {
  float: right;
  padding-right: 1rem;
  color: #F7FFF6;
}

/* Unique styling for list name */
.list-header h2 {
  border: 1px solid #000;
  width: 100%;
  font-size: 1.5rem;
  font-weight: bold;
  padding: 0.5rem 0;
  text-align: center;
  border-radius: 0.5rem 0.5rem 0 0;
  background-color: #1D3557;
  color: #F7FFF6;
}

/* Padding */
.list-item, .list-item a {
  padding: 0.5rem;
  text-decoration: none;
  color: #000;
}

/* Add a nice, subtle contrast if a user hovers over a list item */
li:hover {
  background-color: #E4E7E4;
  transition: ease-in 0.2s;
}

/* Add a nice small effect to make the icon of the object they are hovering over red */
li:hover > i, div:hover > i {
  transition: ease-in 0.2s;
  color: #E63946;
}

/* Keep from turning checkboxes red */
div.list-item:hover
{
  transition: ease-in 1ms;
  color: #000 !important;
}

/* Float the 'view details' button right */
```

```
.list-item-buttons {
  float: right;
  opacity: 0;
}

/* If someone hovers over a list item, show the 'view details' button */
li:hover .list-item-buttons {
  opacity: 100;
  transition: ease-in 0.2s;
  cursor:pointer;
}

/* If an item is completed, set the checkbox colour to green */
.fa-check-square {
  color: green;
}

/********************
 *                *
 * ==== MODALS ==== *
 *                *
*********************/

/* Base modal code */
.modal{
  display: none;
  position: fixed;
  z-index: 1;
  left: 0;
  top: 0;
  width: 100%;
  height:100%;
  overflow: auto;
  background-color: rgb(0,0,0);
      background-color: rgba(0,0,0,0.4);
}

/* Modal header styling */
.modal h4, .modal form label {
  margin-top: 1rem;
  font-size: 1rem;
  font-weight: bold;
}

/* Div containing all user inputs */
.modal-content {
  display: block;
  font-size: 1rem;
  background-color: #F7F0F0;
  margin: 20% auto;
  padding: 1rem;
  border: 1px solid black;
  border-radius: 0.3rem;
  width: 30rem;
}

/* Text inside modals */
.modal-content p {
  text-align: start;
  margin-bottom: 0.6rem;
}

/* User image styling */
.modal-content img {
  margin-top: 1rem;
  width: 100%;
}

/* Close icon for modal */
.close-view-list-item, .modal-close{
  font-size: 1rem;
  float: right;
  cursor: pointer;
  color: #F7FFF6;
  background-color: #E63946;
  border: 1px solid #E63946;
  border-radius: 0.3rem;
  padding: 0.2rem 0.3rem;
}

/* Fix strange spacing issues */
.modal-content div:first-of-type {
  margin-top: 1rem;
}

/* Partially override original input styling to justify left */
.modal input[type="text"] {
  margin-top: 1rem;
  margin-left: 0;
  background-color: #F7FFF6;
}

/* Unique styling for select and input form options */
.modal select, .modal input[type="date"] {
  width: 65%;
  font-size: 1em;
  font-family: inherit;
  color: #000;
  font-weight: 400;
  text-align: left;
  border: 1px solid black;
  border-radius: 0.3rem;
  padding: 0.6rem;
  background-color: #F7FFF6;
  margin-bottom: 1rem;
}

/* Necessary override or else the date input is larger than everything else */
.modal input[type="date"]{
  width: 29% !important;
}

/* Fix strange spacing issues */
.modal-content input[type="submit"] {
  margin: 1rem 0;
}

/* Styling for description input fields */
textarea {
  margin: 1rem 0;
  font-family: inherit;
  resize: vertical;
  word-wrap: break-word;
  width: 95%;
  height: auto;
  text-align: start;
  border: 1px solid black;
  border-radius: 0.3rem;
  padding: 0.6rem;
  background-color: #F7FFF6;
}

/* Hide item id */
.modal input[name="item-id-u"] {
  display: none;
}

/* Paragraphs of instructions on the mainpage */
.mainpage-instructions{
  margin-top: 2rem;
```

```css
}

/* Style question to stand out */
.mainpage-instructions h4 {
  padding: 0.5rem 0;
  font-weight: bold;
}

.mainpage-instructions p {
  padding: 0.5rem 0;
}
```