

# SUBMITTING FORM DATA TO SERVER

---

## Goal(s)

- ☐ Designing a HTML form that submits data to server.
- ☐ Receiving form data with PHP using super global variables (get, post, request)

## What You Should Already Know

Before starting this assignment, you should have the basic knowledge of the followings:

- HTML
- PHP
- FILE
- SESSION

## Academic Honesty

All the work that you do toward the fulfilment of this assignment must be your own unless collaboration is explicitly allowed (e.g., by some problem set or the final project). Viewing or copying another individual's work (even if left by a printer, stored in an executable directory, or accidentally shared in the course's virtual classroom) or lifting material from a book, magazine, website or from other sources, even partially, and presenting it as your own, constitutes academic dishonesty, as does showing or giving your work, even in part, to another student.

Similarly, dual submission academic dishonesty: you may not submit the same or similar work to this course that you have submitted or will submit to another. You should not provide or make available your assignments to individuals who have taken this course or may take this course in the future.

You are welcome to discuss the course's material with others in order to better understand it. You may even discuss problem sets with classmates, but you should not share code. You may also turn to the Web for instruction beyond the course's lectures, for references, and for solutions to technical difficulties, but not for outright solutions to problems on projects. However, failure to cite (as with comments) the origin of any code or technique that you do discover outside of the course's lectures and (even while respecting these constraints) and then integrate into your own work may be considered academic dishonesty.

All forms of academic dishonesty will be dealt very harshly.

## Grades

Your work will be evaluated along the following axes.

**Correctness:** Consistency with the given specifications and free of bugs.

**Design:** Well written code (i.e., clearly, efficiently, elegantly, and/or logically)

**Style:** Readable and modifiable code (commented, aptly named variables etc.)

## Form Data Submission

An HTML form is used to submit information to a server page defined in the **action** attribute of the form tag. The button that is to be clicked to send data to server must be '**submit**' in type to perform this action. At the server side, data is received using **\$\_REQUEST** or **\$\_POST** server variable with the corresponding HTML control name as the **key** of that server variable. For example, **\$\_REQUEST['myname']** or **\$\_POST['myname']** where *myname* is the value of the **name** attribute of a textbox.

## Required Features

- In the form there will be input field for
  - **First Name (Textbox)**
  - **Last Name (Textbox)**
  - **DOB**
    - **Day (Dropdown)**
    - **Month (Dropdown)**
    - **Year (Dropdown)**
  - **Gender (Radio)**
  - **Phone (Textbox)**
  - **Email ID (Textbox)**
  - **Password (password)**
  - **Confirm password (password)**
  - **Profile Picture**
- Any of the inputs cannot be empty.
- The email id should be valid according to the structure of an email address i.e. any@thing.
- The password fields should match with each other.
- The password fields must contain one Special Character and greater than 8 char in length
  - [http://www.w3schools.com/php/php\\_ref\\_string.asp](http://www.w3schools.com/php/php_ref_string.asp)
- If any of the conditions does not match, show proper message box/alert to notify the user.
- **If all the info is valid then append all of them into a file named : "record.txt"**
- Hash of password should be stored into file

**Note:** *If not valid, return user with corrected info they already provide, so that they do not need to re-entry the corrected fields once again.*

## Technical Requirements

- Your HTML should be well-formed that follows the basic syntax of HTML. Explain any intentional invalidity in details in HTML comments.
- You should not use an intelligent text editor to generate HTML.
- **JavaScript and CSS prohibited.**

## Exit Interview

Once done with your site, put together a text file **readme.txt**. Treat this readme as your opportunity not only to explain but to justify your design decisions. Keep this readme to just a few paragraphs in length.

## Additional Tasks

### login.html

1. User should get a form to provide login credentials
2. Login info provided by the user will be submitted to checker.php

### checker.php

- This page will check if login data received from login.html are matched with the data stored in the above mentioned file then page will be redirected to home.php
- When redirecting to home.php , this file will update a flag into session, only if the user is valid(based on user data stored in the file) and store user email into session for further identification

### home.php

- When loading the home.php , your script in the page must check whether the current user is validated by checker.php or not (based on the data in **record.txt**). Only if the user is, validated by checker.php then user can view the content of home.php. If not a valid user then page will be redirected to signout.php
- signout.php should unset all the values stored in the session and destroy the current session, then redirects to index.html. So user gets out of his/her home page
- Content of the home.php page should be the stored info of the current user, so your script will search the data associated with the current user and show them into the home.php. Remember, current user email is stored in the session by checker.php
- In the home.php there will be several links like : sign out , edit info, change password. Again, the profile picture of the user will be visible in the home.php page.
- The image URL for profile picture is stored in the file, only use that URL (string value) with an <img> tag. That will help to show an image.

### signout.php

- This page will unset all the variables stored in the session and destroys the current session
- Redirects to the index.html, so user gets out of the portal

### edit.php

- It will provide facility to edit all the info except the password for the current user
- For identification email can be found in the session, it can be used for searching current user info from file
- User can change all of his/her info and your script will update the file, thus newly edited data is updated in the file

### changepass.php

- This page will help user to change current password to newer one
- For identification email can be accessed from session

**Upload Date**

**Saturday, 11 PM Night, 22 October 2016**