Input from the user

Textfield

Textarea

List RED GREE YELL BLUE

Checkbox green red blue

Buton FM AM

Dropdown Wyatt Figueroa

Submit

```
<form method="post" action="showWorksOn.php">
    Manager:
    <select NAME="EmpID">
      <option VALUE="9">Wyatt Figueroa</option>
      <option VALUE="7">Ursula Stewart</option>
      <option VALUE="6">Odette Espinoza</option>
    </select>
    <input TYPE="submit" NAME="Request" VALUE="Go" />
</form>
```

showWorksOn.php

```php
<?php

  require_once 'connDB.php';
  require_once 'queryWorksOnByEmpID.php';

  if( !isset ($_POST['EmpID']) )
  {
     die("ERROR: No EmpID");
  }

  $EmpID = $_POST['EmpID'];

  $dbh = db_connect();
  $data = getWorksOnByEmpID($dbh, $EmpID);

  // display data in table

?>
```

Other Input Types

```
<input TYPE="submit" NAME="Request" VALUE="Go" />
```

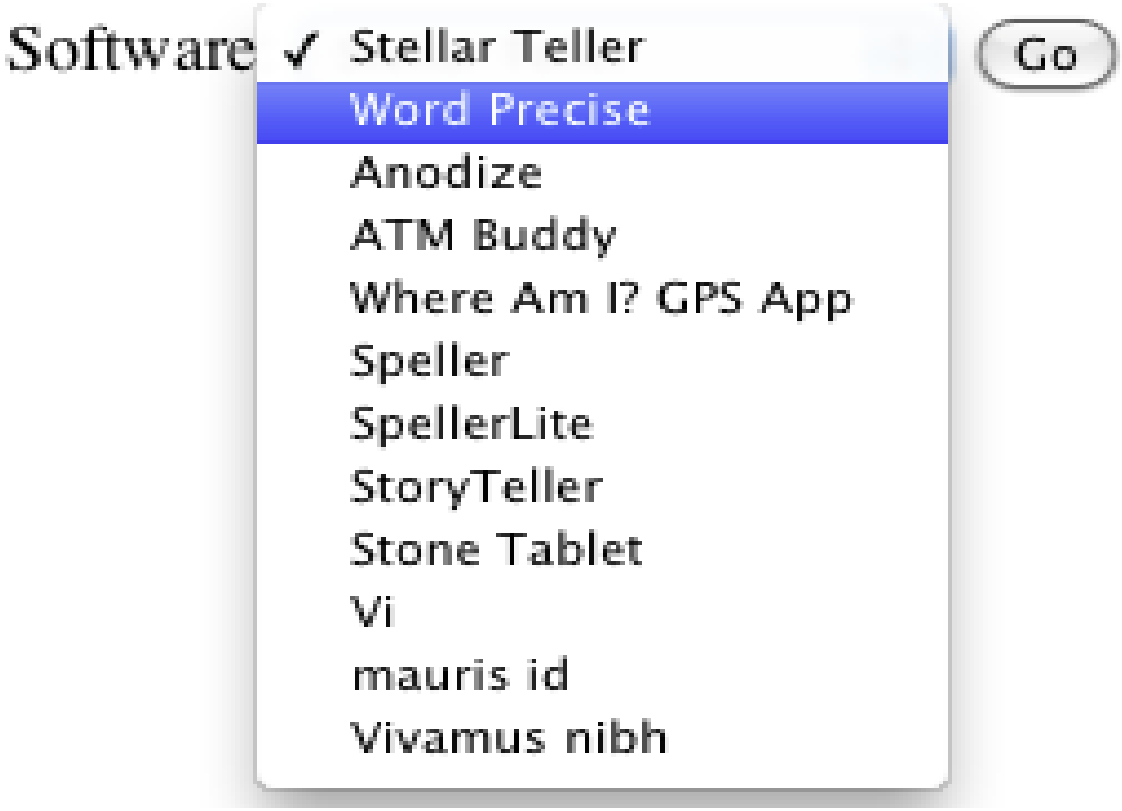
- **TYPE=”text”**
- **TYPE=”password”**
- **TYPE=”radio”**
- **TYPE=”checkbox”**
- **TYPE=”textarea”**

http://www.w3schools.com/html/html_forms.asp

Exercises

- Build a page to show all client information

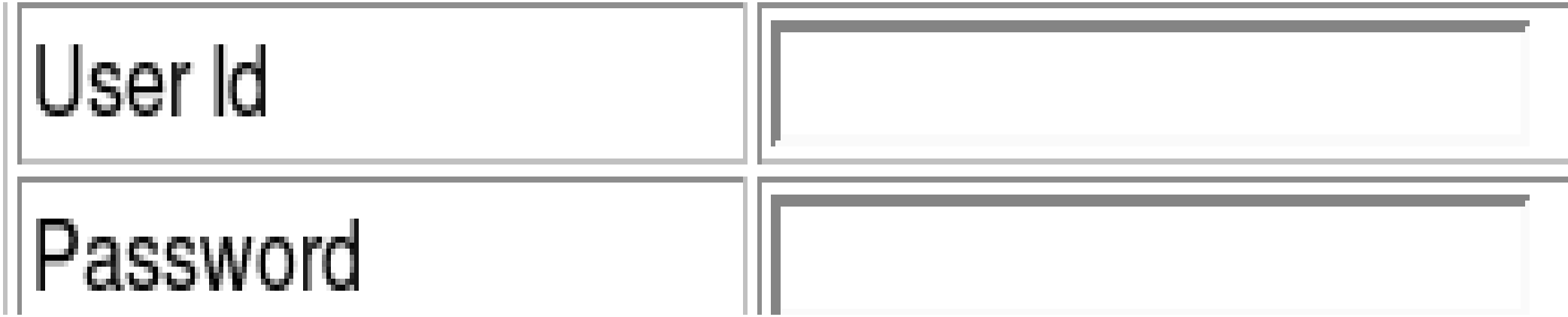
- Build a page to auto-populate a select box with software and then show all the software the chosen software directly depends on.



User Authentication

- Store usernames and passwords in the DB
  - Don't make a MySQL account for every user!
  - Securely store the passwords!

```
create table users (username varbinary(25),
                    passwd varbinary(XX),
                    salt varbinary(YY),
                    Primary Key (username));
```



Password Security

- Threats:
  - Intercept in flight
    - Solution: SSL/https
  - Brute force attack (external)
    - Solution: strong passwords, limited login failures
  - Brute force attack (internal)
    - someone stole your database and has the users table!
    - Solution: store hashed passwords
      - salted passwords
      - choose a good hash algorithm

Pseudo-code

$salt = generateRandomString();

$hashedPwd = somehash($passwd . $salt);

“Insert into table users values ($user, $hashedPwd, $salt);”

Job of the salt:

Job of the hash:

Other Resources

http://www.php.net/manual/en/faq.passwords.php

http://www.w3schools.com/php/func_string_crypt.asp

http://www.ibm.com/developerworks/opensource/library/os-php-encrypt/

http://stackoverflow.com/questions/1581610/how-can-i-store-my-users-passwords-safely

http://php.net/manual/en/function.crypt.php

http://www.openwall.com/phpass/

userAuth.php

```php
<?php
$_SESSION['VALID'] = 0;

if(  isset($_POST['txtUser']) &&
     isset($_POST['txtPassword']))
{
   $userID = $_POST['txtUser'];
   $passwd = $_POST['txtPassword'];

   $result = queryValidUser($dbh, $userID, $passwd);

   if( TRUE == $result )
   {
      $_SESSION['VALID'] = 1;
      header('Location: loggedIn.php');
   }
   else
   {
      header('Location: login.html');
```

queryValidateUser.php

```
<?php
function queryValidateUser($dbh, $user, $passwd)
{
   $retVal = FALSE;
   $salt = queryGetSalt($dbh, $user);

   $hashedPW = crypt($passwd.$salt,
      '$2y$07$8d88bb4a9916b302c1c68c$');

   $sth = $dbh->prepare("SELECT * FROM users WHERE
        username = :user and passwd = :pass");
    $sth->bindValue(“:user”,$user);
    $sth->bindValue(“:pass”,$hashedPW);
   $sth->execute();

    if( 1 == $sthWhereName -> columnCount()  )
    {
      $retVal = TRUE;
    }
   return $retVal;
}
```

login.html

```
<body>

  <form method="post" name="frmLogin" action=
  “authUser.php”>

  Username:
  <input name="txtUserId" type="text" >

  Password:
  <input name="txtPassword" type="password">

  <input type="submit" name="btnLogin" value="Login">

  </form>

</body>
```

authHelper.php

```
<?php
   // include this code at the top of each
   // php file that requires the user to
   // have already been authenticated

   if( !isset($_SESSION['VALID']) ||
       $_SESSION['VALID'] != 1 )
   {
      header('Location: login.html');
   }
?>
```

Binary Data

```
CREATE TABLE pictures (
    `PicID` int(11) NOT NULL auto_increment,
    `image` mediumblob NOT NULL,
    `type` varchar(255) NOT NULL,
   PRIMARY KEY  (`PicID`)) ENGINE=InnoDB;
```

For binary data, we need to track the type of data we have stored.

Usually the MIME type.

image/gif

image/png

binaryDataInput.php

```
<body>

    <form method="post"
      action=binaryDataInput.php
      enctype="multipart/form-data">


      <input type="hidden" name="MAX_FILE_SIZE"
        value="1000000">


      <br>File to upload/store in database:<br>
      <input type="file" name="datafile"  size="40">


     <p>
      <input type="submit" name="submit"
        value="submit">
    </form>

</body>
```

binaryDataInput.php

```
<?php
if(isset($_POST['submit']) ) {

  $filename = $_FILES['datafile']['tmp_name'];
  $filesize = $_FILES['datafile']['size'];
  $filetype = $_FILES['datafile']['type'];

  $data = fread( fopen($filename, "r"),
                      filesize($filename));
  $sth = $dbh->prepare("INSERT INTO pictures
     VALUES (null, :data , :filetype)");
  $sth->bindValue(“:data”, $data);
  $sth->bindValue(“:filetype”,$filetype);

  $sth->execute();

  print “We just added PicID:”. $dbh->lastInsertId();;
  print “{$filetype} {$_FILES['datafile']['name']}”;

}
?>
```

http://www.phpbuilder.com/columns/florian19991014.php3?page=2

getData.php

```
<?php
  if( isset($_GET['id']) ) {


     $id = $_GET['id'];
     $sth = $dbh->prepare(“select image, type from
        pictures where PicID=:picid";
     $sth->bindValue(“:picid”,$id);

     $sth->execute();

     $row = $sth->fetch(); // typo on handouts!
     $data = $row['image'];
     $type = $row['type'];

     Header( "Content-type: $type");
     print $data;


  }else{
      print “FILE NOT FOUND”;
  }
?>
```

showImage.html

https://64.59.233.246/chadd/getData.php?id=1

```
<html>
   <body>
      Image: <img src="getData.php?id=24">
   </body>
</html>
```

```
<html>
   <body>
      Image: <img src="getData.php?id=
   <?php $_SESSION['PicID'] ?>">
   </body>
</html>
```

Practice Exercise

- Add an Editor field to the user table
  - only allow people marked as editors to insert data in the queries below
- Build a webpage to create a new user
- Build a webpage that allows a user to enter a new Student
  - provide a drop down box listing all majors
- Build a webpage that allows the user to search for Students that received a specific final grade
  - provide a drop down box listing grades (A,A-,B+,B,...)