1. PHP Validation: 18%

   Outstanding

   i. The web application performs server-side validation using PHP on all fields (i.e., mandatory and optional) included in the form(s) used to add and/or update records in the database.

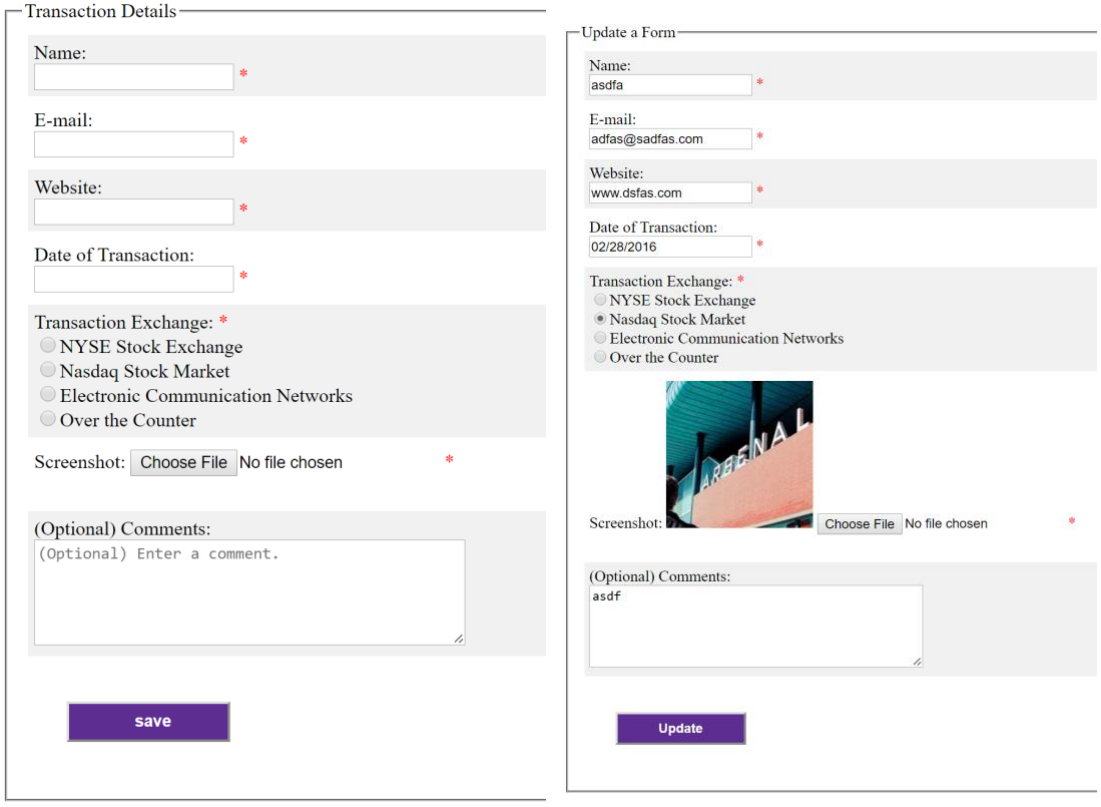
   ii. Validation both verifies whether required fields have been filled in or not and the integrity of the data entered by the user (e.g., a Name field does not accept numeric characters; an image upload field does not accept a .docx file).

   iii. Error messages regarding data integrity issues are very clear (e.g., “Phone number should have xxx-xxx-xxxx format” –telling the user what the expected format is).

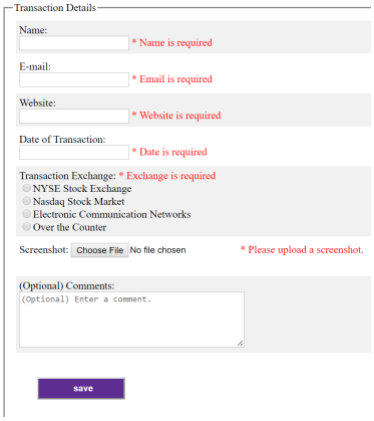
   My solutions:

   There are 7 fields included. 4 text boxes with different format, 1 radio input, 1 file upload, 1 textfield. All validations are implemented with PHP codes written in the same .php file the inputs are entered.

   i.

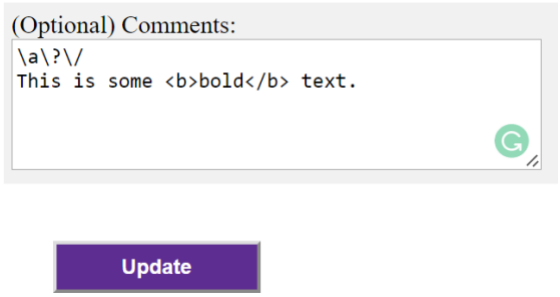


Above is my input and update page, I have implemented checks on all required fields, if required fields are left empty, error message will appear after the “*” symbol, as is shown below:



For the optional comment field, despite there isn't a required validation, I have added function where unnecessary special characters are removed to improve system security.

Input:



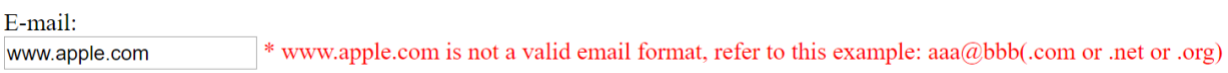
Output:

ii.

As required validation has been explained in the above part, the following part will explain how and where I implemented format evaluation.

Name field required input to be characters only, in incorrect format is entered, an error message will appear to tell user what is accepted format.

Email field will examine whether format of email is valid, if it doesn't meet requirement, an error message will appear and show a few valid examples to follow.



Website field will perform validations that are similar to Email field, error message and examples will appear if format is not correct.

Website:
john@gwu.edu * john@gwu.edu is not a valid URL, refer to this example: (http:// or ftp://)www.aaa(.com or .net or .org)

Date of Transaction field performs validations that are more complex than other fields. It will examine format, time span and provide guidelines for user to input valid date values.

Date of Transaction:
2012/02/17 * Invalid date format, follow example mm/dd/yyyy.

Date of Transaction:
02/17/2012 * Only records after year 2015 is accepted

Date of Transaction:
02/30/2017 * Invalid date format, follow example mm/dd/yyyy.

Date of Transaction:
02/12/2018 * Date should not later than today

Transaction Exchange field uses radio buttons to input values, therefore, only required validation is performed.

File upload field will only accept picture format as screenshot of the transaction. If input is a picture, the field will examine whether the file exceeds maximum allowed size (maximum 5 MB).

Screenshot: Choose File desktop.css * Sorry, only JPG, JPEG, PNG & GIF files are allowed.

Screenshot: Choose File No file chosen * Sorry, your file is too large.Maximum 5 MB allowed

iii.

As is shown in the above examples, guidelines were provided for user to input valid values.

Name:
12??xxxx * Only letters and white space allowed

Date of Transaction:
2012/02/17 * Invalid date format, follow example mm/dd/yyyy.

Date of Transaction:
02/17/2012 * Only records after year 2015 is accepted

Screenshot: Choose File desktop.css * Sorry, only JPG, JPEG, PNG & GIF files are allowed.

Screenshot: Choose File No file chosen * Sorry, your file is too large.Maximum 5 MB allowed

2. Store Record: 18%

Outstanding

i. The web application stores all valid data entered by the user in the MySQL database.
ii. Data that has not been validated is not stored in the database.
iii. Data is collected from more than five form fields.
iv. The form fields exhibit variety (i.e., at least two different types of input fields).
v. The application demonstrates use of advanced functionality. For example, file upload or AJAX.

My solutions:

i. I have set check variable to each field and only after variables of all fields are set to “1”, the upload and database operation can be performed. (for the screenshot upload field, I used $errMSG when there are mismatches in formats, thereby, the variable $errMSG, rather than “checkImage” is used here.)

```php
if(empty($username)){
    $user_nameErr = "Name is required";
} else {
    $username = test_input($_POST["user_name"]);
    // check if name only contains letters and whitespace
    if (!preg_match( pattern: "/^[a-zA-Z ]*$/",$username)) {
        $user_nameErr = "Only letters and white space allowed";
    } else {
        $checkname = "1";
    }
}
```

```php
if(empty($imgFile)){
    $errMSG = "Please Select Image File.";
    $imageErr = "Please upload a screenshot.";
} else
{
    $upload_dir = 'user_images/'; // upload directory

    $imgExt = strtolower(pathinfo($imgFile, options: PATHINFO_EXTENSION)); // get image extension

    // valid image extensions
    $valid_extensions = array('jpeg', 'jpg', 'png', 'gif'); // valid extensions

    // rename uploading image
    $userpic = rand(1000,1000000).".".$imgExt;

    // allow valid image file formats
    if(in_array($imgExt, $valid_extensions)){
        // Check file size '5MB'
        if($imgSize < 5000000)              {
            move_uploaded_file($tmp_dir, destination: $upload_dir.$userpic);
        } else{
            $imageErr = "Sorry, your file is too large.Maximum 5 MB allowed";
            $errMSG = "Sorry, your file is too large.Maximum 5 MB allowed";
        }
    } else{
        $imageErr = "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
        $errMSG = "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
```

```php
// if no error occured, continue ....
if(!isset($errMSG)&& $checkdate==1 && $checkexchange==1 && $checkemail==1 &&$checkname==1&&$checkwebsite==1)
{
    $stmt = $DB_con->prepare( statement: 'INSERT INTO tbl_users(userName,userPic,email, exchange, comment,date, website )
```

ii. As is explained in the above example, only the input has passed all validations, the follow up actions of file and database operations can be performed.

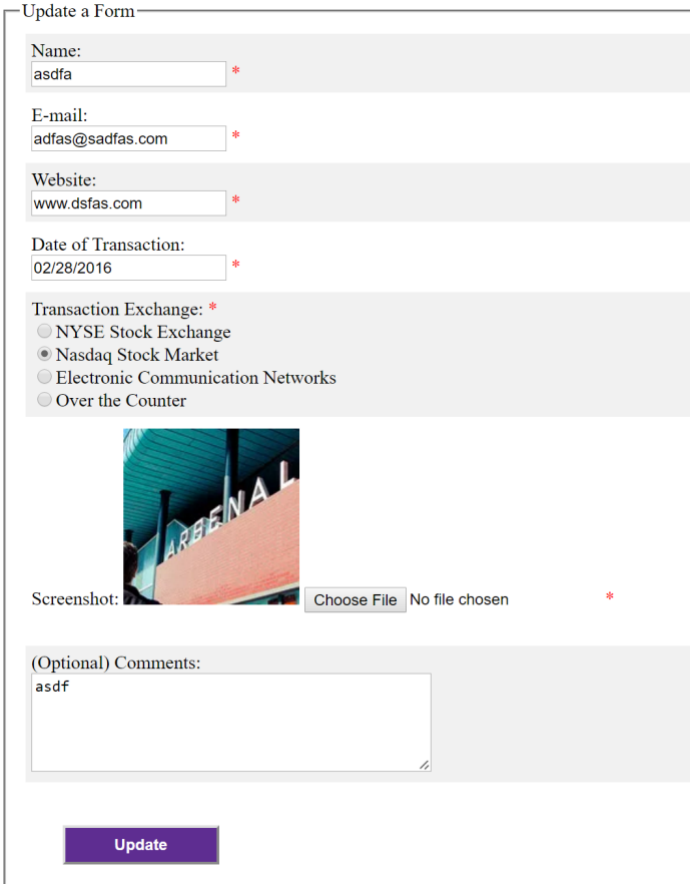
iii. There are 7 fields included. 4 text boxes with different format, 1 radio input, 1 file upload, 1 textfield.

iv. For the text fields, I have implemented validations for name, website, email, and date, each one have a different validation mechanism from another. I also have radio button, textfield, and file upload input fields.

v. I have applied AJAX extensively as well as implemented a picture upload feature for my project. Here are a few examples:

```
><a class="btn"  href="update.php?edit_id=<?php echo $row['userID']; ?>" title="click for edit" onclick="return confirm('Sure to edit ?')"><
 <a class="btn" href="?delete_id=<?php echo $row['userID']; ?>" title="click for delete" onclick="return confirm('Sure to delete ?')"><span
```

Two buttons request for confirm and get userID to delete and stay in the same page or update information on another page.



Above is an example of update page where each field is prefilled with information regarding userID retrieved from ID of the row the button clicked. Validations are performed, error messages are displayed if update is not in valid format on the same page. This is also an example where file upload feature is used.

3. Retrieve results:
   i. The web application is able to retrieve all records in the MySQL database.
   ii. Changes to the database records (i.e., add, update, or delete records) are reflected when the user retrieves all records currently in the database.
   iii. Records retrieved from the database are displayed in a table format.
   iv. The application demonstrates use of advanced functionality. For example, file retrieval or AJAX.

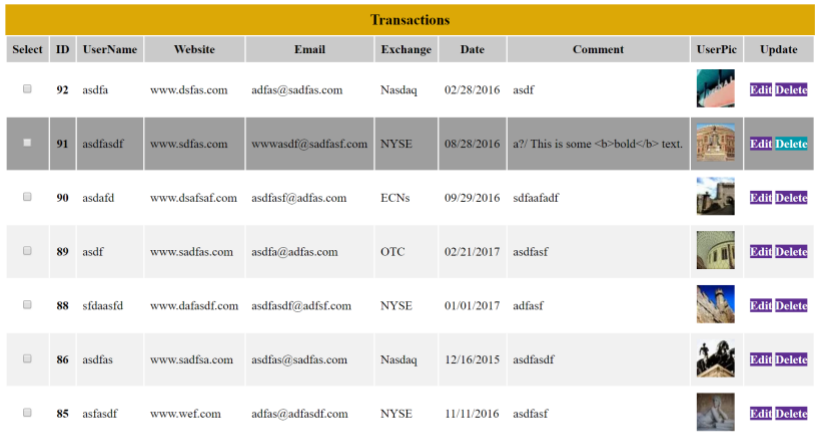
   My solutions:

   i. All valid results can be retrieve from database. See the following example:

Transactions

ID	UserName	Website	Email	Exchange	Date	Comment	UserPic
92	asdfa	www.dsfas.com	adfas@sadfas.com	Nasdaq	02/28/2016	asdf	
91	asdfasdf	www.sdfas.com	wwwasdf@sadfasf.com	NYSE	08/28/2016	a?/ This is some bold text.	
90	asdafd	www.dsafsaf.com	asdfasf@adfas.com	ECNs	09/29/2016	sdfaafadf	
89	asdf	www.sadfas.com	asdfa@adfas.com	OTC	02/21/2017	asdfasf	
88	sfdaasfd	www.dafasdf.com	asdfasdf@adfsf.com	NYSE	01/01/2017	adfasf	
86	asdfas	www.sadfsa.com	asdfas@sadfas.com	Nasdaq	12/16/2015	asdfasdf	
85	asfasdf	www.wef.com	adfas@adfasdf.com	NYSE	11/11/2016	asdfasf	

   ii. When the delete link is clicked, result is deleted and user will see the change. In this example, I deleted the record, highlighted with ID of “91”. The link will refresh the page and see the result, because I assigned a same page after action is perform. The function can still work, as have been tested, if the redirect code is removed.



Transactions

Select	ID	UserName	Website	Email	Exchange	Date	Comment	UserPic	Update
☐	92	asdfa	www.dsfas.com	adfas@sadfas.com	Nasdaq	02/28/2016	asdf		Edit Delete
☐	91	asdfasdf	www.sdfas.com	wwwasdf@sadfasf.com	NYSE	08/28/2016	a?/ This is some bold text.		Edit Delete
☐	90	asdafd	www.dsafsaf.com	asdfasf@adfas.com	ECNs	09/29/2016	sdfaafadf		Edit Delete
☐	89	asdf	www.sadfas.com	asdfa@adfas.com	OTC	02/21/2017	asdfasf		Edit Delete
☐	88	sfdaasfd	www.dafasdf.com	asdfasdf@adfsf.com	NYSE	01/01/2017	adfasf		Edit Delete
☐	86	asdfas	www.sadfsa.com	asdfas@sadfas.com	Nasdaq	12/16/2015	asdfasdf		Edit Delete
☐	85	asfasdf	www.wef.com	adfas@adfasdf.com	NYSE	11/11/2016	asdfasf		Edit Delete

Delete Multiple * Select Records to Delete

Transactions

Select	ID	UserName	Website	Email	Exchange	Date	Comment	UserPic	Update
☐	92	asdfa	www.dsfas.com	adfas@sadfas.com	Nasdaq	02/28/2016	asdf		Edit Delete
☐	90	asdafd	www.dsafsaf.com	asdfasf@adfas.com	ECNs	09/29/2016	sdfaafadf		Edit Delete
☐	89	asdf	www.sadfas.com	asdfa@adfas.com	OTC	02/21/2017	asdfasf		Edit Delete
☐	88	sfdaasfd	www.dafasdf.com	asdfasdf@adfsf.com	NYSE	01/01/2017	adfasf		Edit Delete
☐	86	asdfas	www.sadfsa.com	asdfas@sadfas.com	Nasdaq	12/16/2015	asdfasdf		Edit Delete
☐	85	asfasdf	www.wef.com	adfas@adfasdf.com	NYSE	11/11/2016	asdfasf		Edit Delete

Delete Multiple * Select Records to Delete

```
require_once 'dbconfig.php';
if(isset($_GET['delete_id']))
{

   // select image from db to delete
   $stmt_select = $DB_con->prepare( statement: 'SELECT userPic FROM tbl_users WHERE userID =:uid');
   $stmt_select->execute(array(':uid'=>$_GET['delete_id']));
   $imgRow=$stmt_select->fetch( fetch_style: PDO::FETCH_ASSOC);
   unlink( filename: "user_images/".$imgRow['userPic']);

   // it will delete an actual record from db
   $stmt_delete = $DB_con->prepare( statement: 'DELETE FROM tbl_users WHERE userID =:uid');
   $stmt_delete->bindParam( parameter: ':uid',$_GET['delete_id']);
   $stmt_delete->execute();
   ?>
   <script>
       alert('Successfully Deleted ...');
       //window.location.href='editlist.php';
   </script>
```

iii. As is shown in above examples, the records are organized in table format.

iv. As is shown in above examples, AJAX is reflected in codes and file (the pictures in table) retrieve and delete are demonstrated. Evidence is shown in the code below.

```
<th>
    <input type="checkbox" name="checked_id[]" id= "checked_id[]" class="checkbox" value="<?php echo $row['userID']; ?>"/>
</th>
<th scope="row"><?php echo $userID ?></th>

<td><?php echo $userName ?></td>
<td><?php echo $website ?></td>
<td><?php echo $email ?></td>
<td><?php echo $exchange ?></td>
<td><?php echo $date ?></td>
<td><?php echo $comment ?></td>
<td><img src="user_images/<?php echo $row['userPic']; ?>" class="img-rounded" width="50px" height="50px" /></td>
<td><a class="btn"  href="update.php?edit_id=<?php echo $row['userID']; ?>" title="click for edit" onclick="return confirm('S
    <a class="btn" href="?delete_id=<?php echo $row['userID']; ?>" title="click for delete" onclick="return confirm('Sure to
```

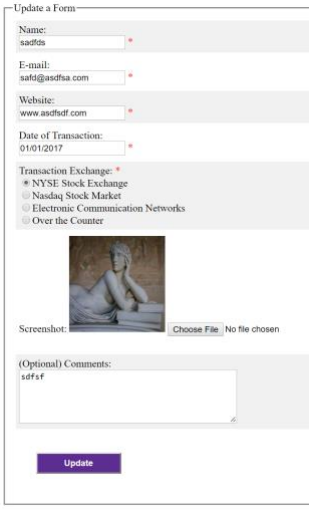
4. Update Record: 18%
   i. The web application allows the user to select a record to be updated. Once the user has selected a record to be updated, a form pre-filled with the data corresponding to the chosen record is displayed to the user.
   ii. After the user has changed the pertinent fields, the record is successfully updated.
   iii. The application demonstrates use of advanced functionality. For example, file update or AJAX.

   My solutions:

   i. User can click the" edit" link on the end of row of each record to update the record.

Transactions									
Select	ID	UserName	Website	Email	Exchange	Date	Comment	UserPic	Update
☐	104	sadfds	www.asdfsdf.com	safd@asdfsa.com	NYSE	01/01/2017	sdfsf		Edit Delete

   A new page will open and all fields are filled with data retrieved. After changes are made, click Update button to finish.



   ii. I entered in the comment field "My test message!" and clicked the Update button. The data was updated, reflected in the table.

(Optional) Comments:

My test message!

Update

Transactions

ID	UserName	Website	Email	Exchange	Date	Comment	UserPic
104	sadfds	www.asdfsdf.com	safd@asdfsa.com	NYSE	01/01/2017	My test message!	

iii. AJAX features have been discussed in previous Update examples. To illustrate the file update feature, I updated the picture in the above example.

Transactions

Select	ID	UserName	Website	Email	Exchange	Date	Comment	UserPic	Update
☐	104	sadfds	www.asdfsdf.com	safd@asdfsa.com	NYSE	01/01/2017	My test message!		Edit Delete

5. Delete records: 18%
   i. The web application allows the user to select multiple records to be deleted at a time (i.e., the user has the option to select a single or many records to be deleted).
   ii. Once the user has selected the pertinent records, those records are successfully deleted.
   iii. The application demonstrates use of advanced functionality. For example, file deletion or AJAX.

   My solutions:

   i. Shown in the example, I deleted records with ID of 94 and 99. User can also select one record or click the “Delete” link to delete a specific record.

☑	99	sadf	www.sdfas.com	asdfsadf@sadfasdf.com	ECNs	01/01/2017	fasdfsadf		Edit Delete
☐	98	asdfsaf	www.sadfsaf.com	asdfasf@asdfasf.com	NYSE	01/01/2015	sadfsadf		Edit Delete
☐	97	sadf	www.afdsaf.com	fsadf@sadfasf.com	NYSE	01/01/2015	asfdsafd		Edit Delete
☐	96	sdfsaf	www.wfadfa.com	asdfas@asdfasf.com	Nasdaq	01/01/2017	asdfasfd		Edit Delete
☐	95	sadfsa	www.sfasdf.com	asdfsaf@sadfsaf.com	NYSE	01/01/2015	sadfsf		Edit Delete
☑	94	sadfsa	www.sdfas.com	asdfasf@fsaf.com	NYSE	01/01/2017	sdfasf		Edit Delete
☐	93	sadfsafd	www.sfsadf.com	sadfsa@sadfasf.com	NYSE	01/01/2016	asdf		Edit Delete
☐	89	asdf	www.sadfas.com	asdfa@adfas.com	OTC	02/21/2017	asdfasf		Edit Delete
☐	88	sfdaasfd	www.dafasdf.com	asdfasdf@adfsf.com	NYSE	01/01/2017	adfasf		Edit Delete

Delete Multiple * Select Records to Delete

☐	102	sadfsa	www.dfsafc.com	sadf@wfsad.com	NYSE	01/01/2017	sdfsafd		Edit Delete
☐	101	sadfsa	www.sdfasf.com	sfa@safdasf.com	NYSE	01/01/2017	sdfsaf		Edit Delete
☐	100	sadfsa	www.sdfsa.com	sadfsa@sdfasf.com	OTC	01/01/2017	sadfsafd		Edit Delete
☐	98	asdfsaf	www.sadfsaf.com	asdfasf@asdfasf.com	NYSE	01/01/2015	sadfsadf		Edit Delete
☐	97	sadf	www.afdsaf.com	fsadf@sadfasf.com	NYSE	01/01/2015	asfdsafd		Edit Delete
☐	96	sdfsaf	www.wfadfa.com	asdfas@asdfasf.com	Nasdaq	01/01/2017	asdfasfd		Edit Delete
☐	95	sadfsa	www.sfasdf.com	asdfsaf@sadfsaf.com	NYSE	01/01/2015	sadfsf		Edit Delete
☐	93	sadfsafd	www.sfsadf.com	sadfsa@sadfasf.com	NYSE	01/01/2016	asdf		Edit Delete
☐	89	asdf	www.sadfas.com	asdfa@adfas.com	OTC	02/21/2017	asdfasf		Edit Delete
☐	88	sfdaasfd	www.dafasdf.com	asdfasdf@adfsf.com	NYSE	01/01/2017	adfasf		Edit Delete

Delete Multiple * Select Records to Delete

ii. The records were deleted, as is shown in the database query.

```
mysql> mysql> Select* from tbl_users;
+--------+----------+----------------+---------------------+----------+------------+------------------+------------+
| userID | userName | website        | email               | exchange | date       | comment          | userPic    |
+--------+----------+----------------+---------------------+----------+------------+------------------+------------+
|     96 | sdfsaf   | www.wfadfa.com | asdfas@asdfasf.com  | Nasdaq   | 01/01/2017 | asdfasfd         | 979633.jpg |
|     88 | sfdaasfd | www.dafasdf.com | asdfasdf@adfsf.com | NYSE     | 01/01/2017 | adfasf           | 236198.jpg |
|    100 | sadfsa   | www.sdfsa.com  | sadfsa@sdfasf.com   | OTC      | 01/01/2017 | sadfsafd         | 549057.jpg |
|     97 | sadf     | www.afdsaf.com | fsadf@sadfasf.com   | NYSE     | 01/01/2015 | asfdsafd         | 675903.jpg |
|     98 | asdfsaf  | www.sadfsaf.com | asdfasf@asdfasf.com | NYSE     | 01/01/2015 | sadfsadf         | 637365.jpg |
|     95 | sadfsa   | www.sfasdf.com | asdfsaf@sadfsaf.com | NYSE     | 01/01/2015 | sadfsf           | 881047.jpg |
|     93 | sadfsafd | www.sfsadf.com | sadfsa@sadfasf.com  | NYSE     | 01/01/2016 | asdf             | 504622.jpg |
|     89 | asdf     | www.sadfas.com | asdfa@adfas.com     | OTC      | 02/21/2017 | asdfasf          | 104396.jpg |
|    101 | sadfsa   | www.sdfasf.com | sfa@safdasf.com     | NYSE     | 01/01/2017 | sdfsaf           | 953448.jpg |
|    102 | sadfsa   | www.dfsafc.com | sadf@wfsad.com      | NYSE     | 01/01/2017 | sdfsafd          | 220523.jpg |
|    103 | sadfsa   | www.asdfasd.com | adfas@fasf.com     | Nasdaq   | 01/01/2017 | sadfsa           | 951022.jpg |
|    104 | sadfds   | www.asdfsdf.com | safd@asdfsa.com    | NYSE     | 01/01/2017 | My test message! | 496297.jpg |
+--------+----------+----------------+---------------------+----------+------------+------------------+------------+
12 rows in set (0.00 sec)

mysql>
```

iii. AJAX feature has been shown in the delete feature discussed in part 2 “store record” where I can delete/ update through a link and see result in the same page. Here is my code of delete a record with picture file in database.

```php
require_once 'dbconfig.php';
if(isset($_GET['delete_id']))
{

   // select image from db to delete
   $stmt_select = $DB_con->prepare( statement: 'SELECT userPic FROM tbl_users WHERE userID =:uid');
   $stmt_select->execute(array(':uid'=>$_GET['delete_id']));
   $imgRow=$stmt_select->fetch( fetch_style: PDO::FETCH_ASSOC);
   unlink( filename: "user_images/".$imgRow['userPic']);

   // it will delete an actual record from db
   $stmt_delete = $DB_con->prepare( statement: 'DELETE FROM tbl_users WHERE userID =:uid');
   $stmt_delete->bindParam( parameter: ':uid',$_GET['delete_id']);
   $stmt_delete->execute();
   ?>
   <script>
      alert('Successfully Deleted ...');
      window.location.href='editlist.php';
   </script>
```

6. Semantic Markup: 2.5%
   Outstanding
   i. The HTML markup has minimal inclusion of CSS and PHP.
   ii. In the case of CSS, external style sheets are used to include CSS as opposed to inline or embedded CSS.
   iii. In the case of PHP files, PHP script is interwoven with the HTML markup minimally.
   iv. Use of include statements in the PHP files to separate markup from script is extensive. HTML markup in PHP files is easily distinguishable.
   My solutions:
   i. The project created 6 php files, 1 html file and 3 css files. The homepage.html hosts links to other php files. Each html or php file have html markups and css styles included.
   ii. Desktop.css is created for desktop styles, mobile.css is created for mobile styles, historylist.css is created for tables in the view all records feature.
   iii. In this example, php code is located amid html code to retrieve data for html table

```
<table id= "list">
    <caption>Transactions</caption>
    <thead>
    <tr>
        <th scope="col">Select</th>
        <th scope="col">ID</th>
        <th scope="col">UserName</th>
        <th scope="col">Website</th>
        <th scope="col">Email</th>
        <th scope="col">Exchange</th>
        <th scope="col">Date</th>
        <th scope="col">Comment</th>
        <th scope="col">UserPic</th>
        <th scope="col">Update </th>
    </tr>
    </thead>

    <tbody>


    <?php

    $stmt = $DB_con->prepare( statement: 'SELECT userID, userName, email, website, exchange, date, comment, userPic FROM tbl_users ORDER BY userID DE
    $stmt->execute();

    if($stmt->rowCount() > 0)
    {
        while($row=$stmt->fetch( fetch_style: PDO::FETCH_ASSOC))
        {
            extract($row);
            ?>

    <tr>
            <th>

                <input type="checkbox" name="checked_id[]" id= "checked_id[]" class="checkbox" value="<?php echo $row['userID']; ?>"/>
            </th>
            <th scope="row"><?php echo $userID ?></th>
```

   Another example here is a piece of script code in the middle of php code to pop message box.
   iv. The dbconfig.php file is created to setup database connections and initialize the variables, this file in included every time when database operation is required in a php file. HTML structure are keep the same as created in Project 1, which used HTML markups extensively, because Project is created on basis of Project1. Shown in examples below.

```
require_once 'dbconfig.php';
if(isset($_GET['edit_id']) && !empty($_GET['edit_id'])) {
    $id = $_GET['edit_id'];
    $stmt_edit = $DB_con->prepare( statement: 'SELECT userName,userPic,email, exchange, comment,date, website FROM tbl_users WHERE userID =:uid');
    $stmt_edit->execute(array(':uid' => $id));
    $edit_row = $stmt_edit->fetch( fetch_style: PDO::FETCH_ASSOC);
    extract($edit_row);
```

```
        else{
            $errMSG = "Sorry Data Could Not Updated !";
        }

    }
}
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
```

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="utf-8">
    <title>Stocks Catalog</title>
    <link rel="stylesheet" href="mobile.css" media="screen and (max-width:480px)" />
    <link rel="stylesheet" href="desktop.css" media="screen and (min-width:481px)" />
    <script type="text/javascript" language="javascript" src="inputform.js"></script>
    <style>

    </style>
</head>
<body>
<main>
    <header>
        <h1 id="pagetitle">Stock Transaction System</h1>
        <div>
            <nav id="top">
                <ul>
```

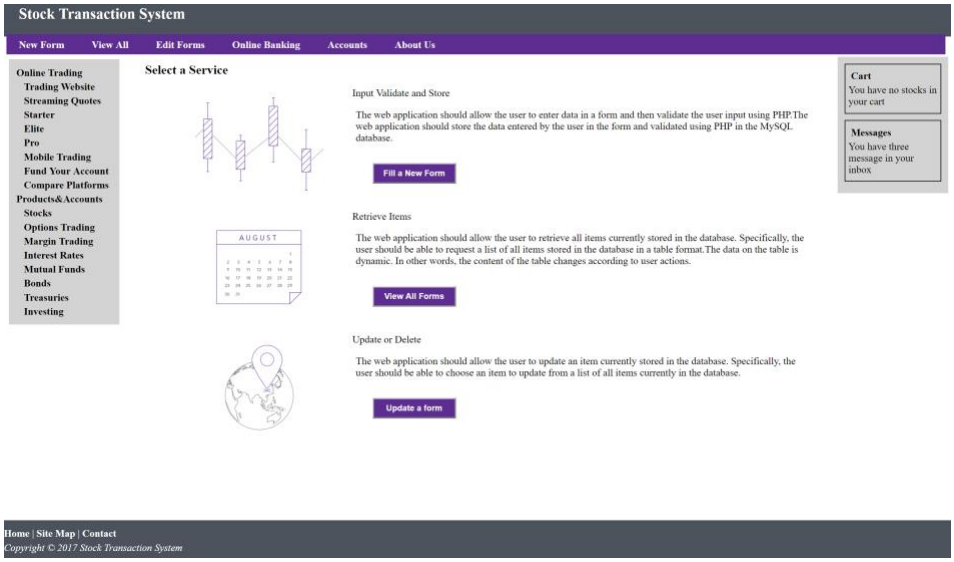
```
        $stmt->bindParam(parameter:':uname', $userName);
        //$stmt->bindParam(':ujob',$userjob);
        $stmt->bindParam(parameter:':upic', $userpic);
        $stmt->bindParam(parameter:':em', $email);
        $stmt->bindParam(parameter:':ex', $exchange);
        $stmt->bindParam(parameter:':cm', $comment);
        $stmt->bindParam(parameter:':dt', $date);
        $stmt->bindParam(parameter:':wb', $website);
        $stmt->bindParam(parameter:':uid',$id);


        if($stmt->execute()){
            ?>
            <script>
                alert('Successfully Updated ...');
                window.location.href='editlist.php';
            </script>
            <?php
        }
        else{
            $errMSG = "Sorry Data Could Not Updated !";
        }
```

7. Responsive Design: 2.5%
   Outstanding:
   i. The entire web application (i.e., all web pages) has a general design (e.g., top navigation bar, bottom navigation bar, side items).

ii. All web pages are responsive to different screen sizes. All the functionalities of the web application are available on devices of different screen sizes. The HTML markup changes according to the screen size when needed.

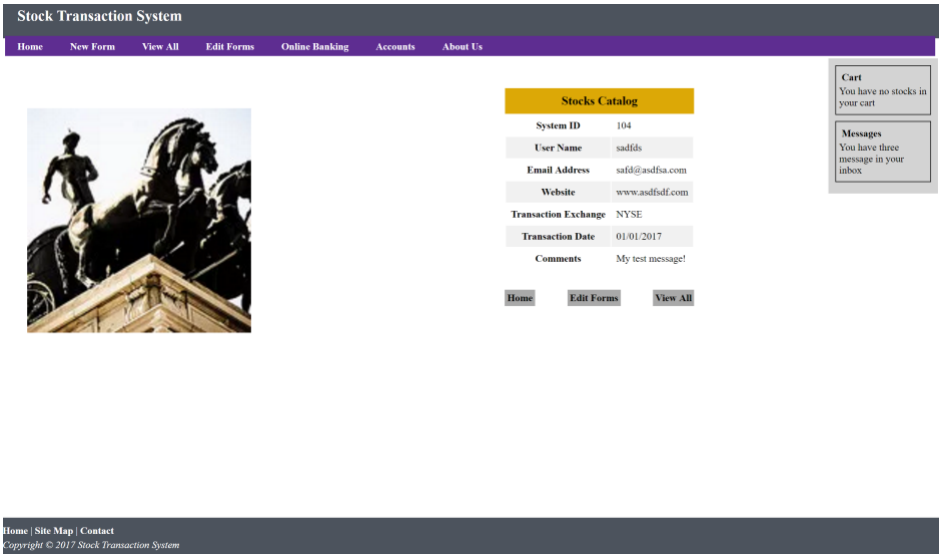
My Solutions:

i. Shown in the examples, header, top and bottom navigation bars, footer and side items are implemented.

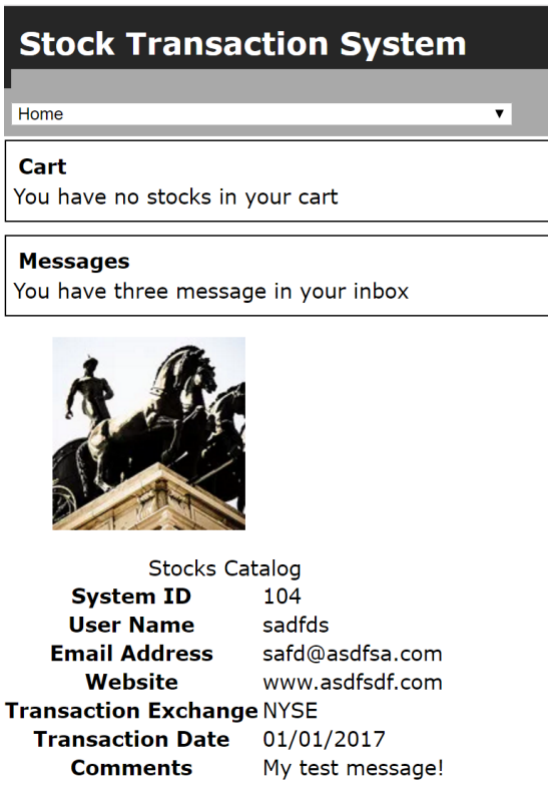


ii. Desktop.css and mobile.css are created to allow the pages to change style according to different screen sizes. Header, Navigation bars and styles changed to adapt to mobile screens.

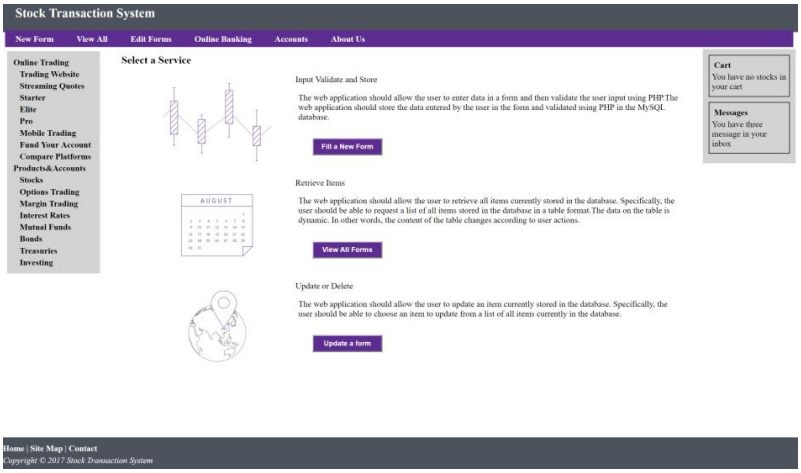
Desktop:

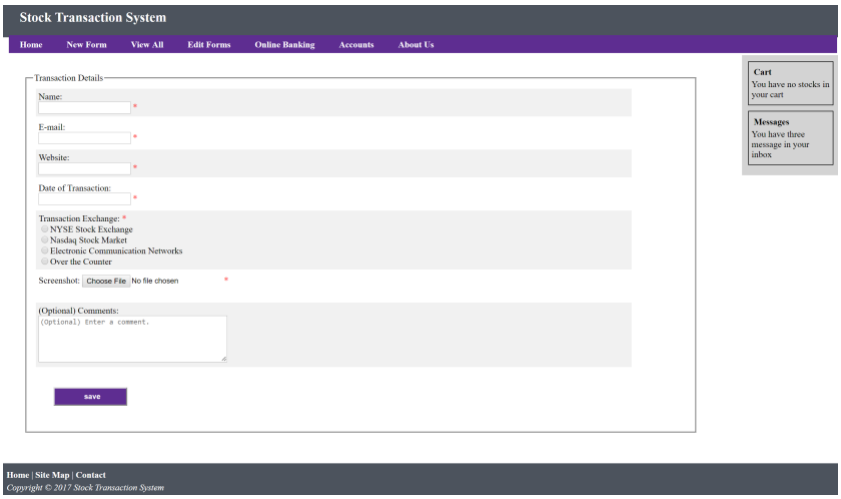


Mobile:



8. Navigation: 5%
   Outstanding:
   i. There are enough links to allow the user to easily access all the web pages associated with the various functionalities of the web application.
   ii. Excessive, duplicate, and unnecessary links are avoided (i.e., web pages are not crowded with links).
   My solutions:
   i. I have created links on the top navigation bars, which appears in every page, as “Home”, “View all”, “Edit Forms”, “New Form” to help user navigate through different functions. The bottom navigation bar also has a “home” link in each page. For some pages, I created links that navigate directly to another function. See the following example:





ii. Only links that are functionally necessary are included in the project, I include top and bottom bar with “home” link to help user to navigate back to home page, which have access to all functions, as well as a brief description of the requirement and feature of each function.