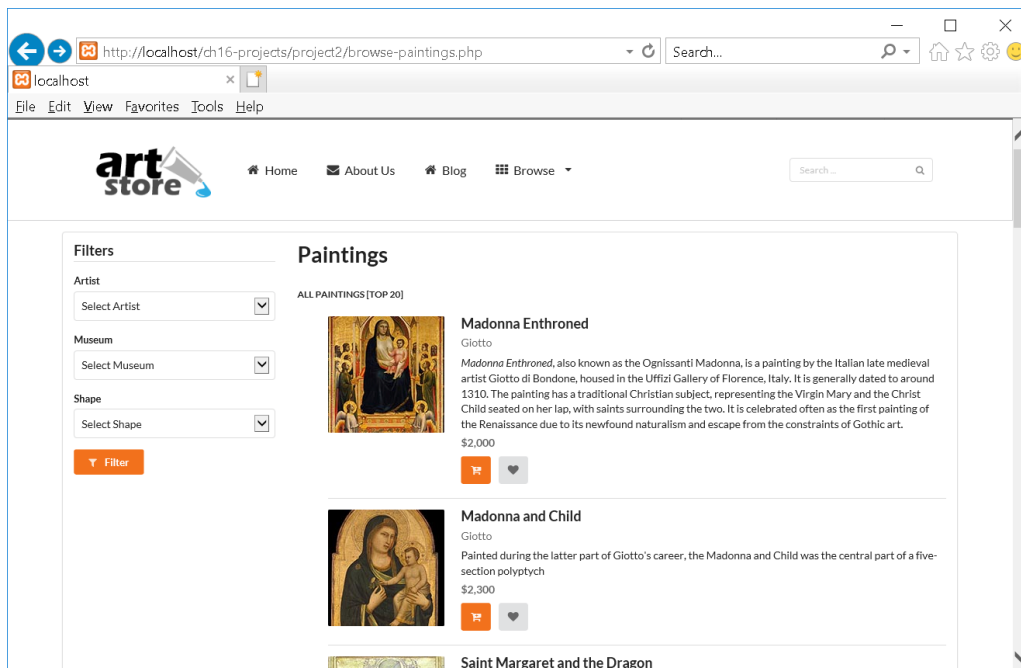


Ch 16 Project 2. Art Store

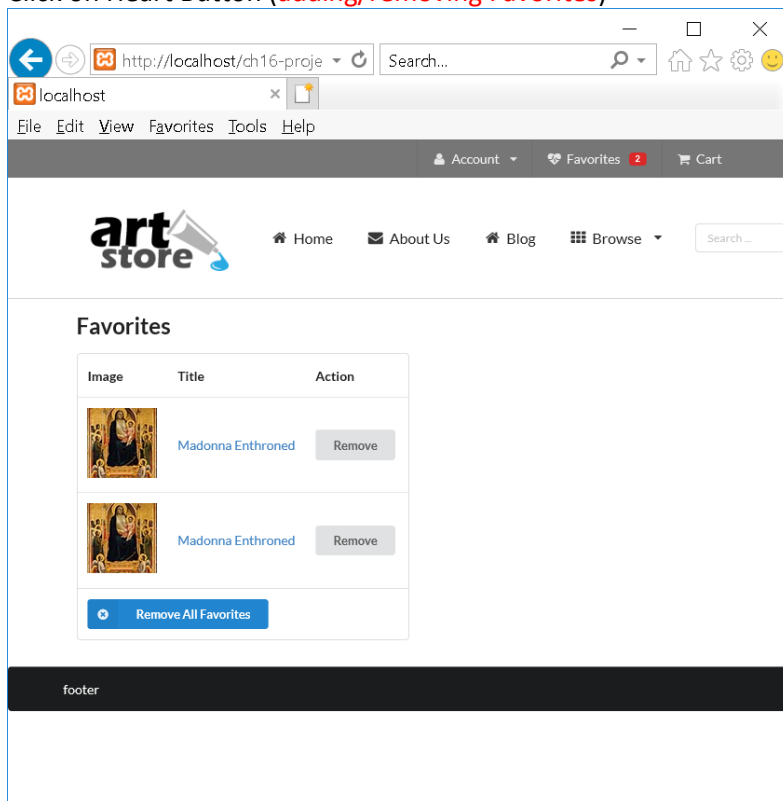
Add feature to implement a “session Variable”

- Session integration requires adding function call to all pages that will use session data.
 - **session_start()**
 - <http://php.net/manual/en/function.session-start.php>
- **browse-painting.php** and **single-painting.php** contain
 - **Add to Favorites** link styled as buttons.
 - Modify these links so that clicking on them will take the user to **addToFavorites.php**
 - May consider to include the following field in the query string
 - **PaintingID**
 - **ImageFileName**
 - **Title**
- Create a new blank page, **addToFavorites.php**, which will handle
 - a “**Get**” request to add a painting to the favorites list
 - This file will contain no markup; it will check for the existence of the relevant query string fields, and then add the painting information to session state.
- The favorites list will be represented as an array of arrays. Each favorite item will be an array that contains the following fields for the painting
 - **PaintingID, ImageFileName, Title**
 - You will need to retrieve the favorites array from “**Session State**” (or create it as a blank array if it doesn’t exist)
 - Then add the array for the new favorite item to the favorite array.
 - You must then store the modified favorite array back in “**Session State**.”
 - After this, redirect to **view-favorites.php** using the **header()** function.
- Modify the **view-favorites.php** page so that it displays the content of the favorite list in a table.
 - For each painting in the favorite list, display a small version of the painting (from the **images/art/works/small-square** folder) and its title.
 - Make the title a link to **single-painting.php** with the appropriate query string.
- Change the button links that will remove each painting from the favorites list as well as the button link to empty all the favorites from the list. These will be links to **remove-favorites.php**;
 - For the remove single painting links, the **PaintID** of the painting to remove will be provided as a query string parameter.
- Create a new blank page, **remove-favorites.php**, which will handle a “**Get**” request to remove a single painting to the favorites list (or remove all paintings)
 - This file will contain no markup.
 - It will check for the existence of the relevant query string fields, and then remove the specific paintings from the favorite array in “**Session State**.” After removing, redirect back to **view-favorites.php** page.
- Modify the **art-header.inc.php** file to display a count of items in the favorite list.
 - Use the class “**ui red mini label**”.

<http://localhost/ch16-projects/project2/browse-paintings.php>



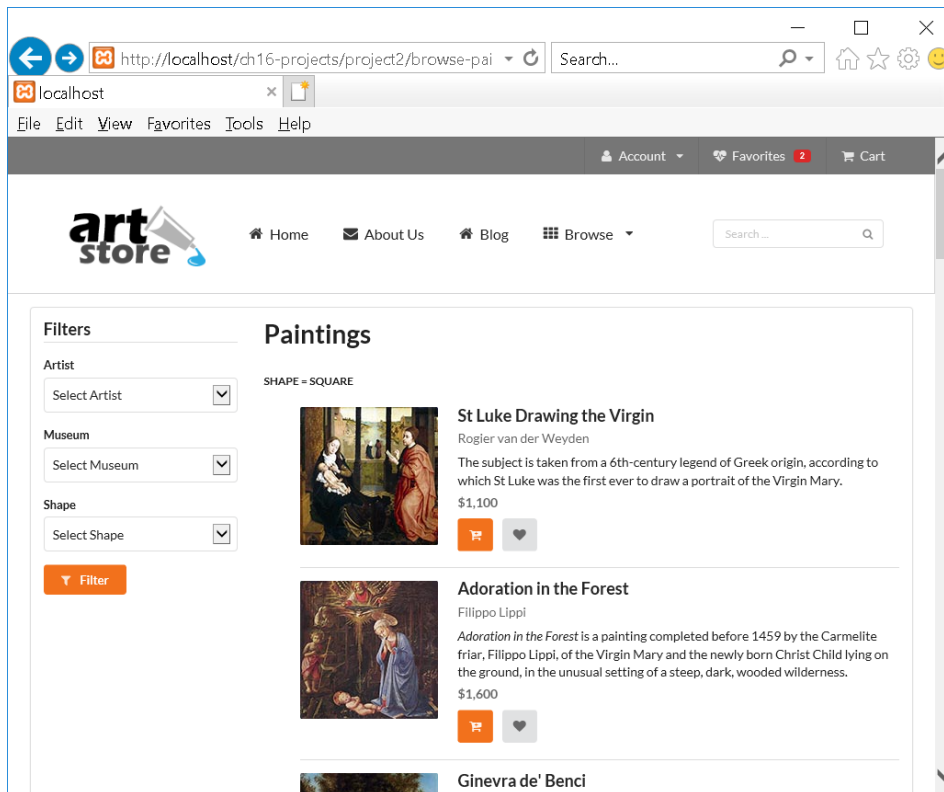
Click on Heart Button (adding/removing Favorites)



Filters (using Filter)

- Artist => Balla

- Museum => Bames Foundation
- Shape => Square
- Click on Filter



```
<?php
//art-config.inc.php

define('DBCONNECTION', 'mysql:host=localhost;dbname=art;charset=utf8');
define('DBUSER', 'testuser');
define('DBPASS', 'secret');

spl_autoload_register(function ($class) {
    $file = 'lib/' . $class . '.class.php';
    if (file_exists($file))
        include $file;
});
/*
$ip = getenv('IP');
$port = '3306';
$user = getenv('C9_USER');
```

```

$connection = "mysql:host=$ip;port=$port;dbname=art";
$pdo = DatabaseHelper::setConnectionInfo(array(DBCONNECTION, $user, ""));
*/

/* localhost connection */
$pdo = DatabaseHelper::setConnectionInfo(array(DBCONNECTION, DBUSER, DBPASS));
?>

```

```

<?php
//art-functions.inc.php
function generateReviewStars($rating) {
    $output = "";
    for ($i=0; $i<$rating; $i++) {
        $output .= '<i class="star icon"></i>';
    }
    return $output;
}

function generateRatingStars($rating) {
    $output = "";
    for ($i=0; $i<$rating; $i++) {
        $output .= '<i class="orange star icon"></i>';
    }
    for ($i=$rating; $i<5; $i++) {
        $output .= '<i class="empty star icon"></i>';
    }
    return $output;
}

function generateLink($url, $label) {
    return "<a href='$url'>$label</a>";
}

function outputFilterOptions($data, $valueField, $dataField) {
    while ($single = $data->fetch()) {
        echo '<option value=' . $single[$valueField] . '>';
        echo utf8_encode($single[$dataField]);
        echo '</option>';
    }
}

function makeArtistName($first, $last) {
    return utf8_encode($first . ' ' . $last);
}
?>

```

```

<?php
/* Handles database access for the Artist table. */
// ArtistDB.class.php
class ArtistDB
{
    private $pdo = null;
    private static $baseSQL = "SELECT * FROM Artists";
    private static $constraint = ' order by LastName';

    public function __construct($connection) {
        $this->pdo = $connection;
    }

    public function findById($id)
    {
        $sql = self::$baseSQL . ' WHERE ArtistID=? ';
        $statement = DatabaseHelper::runQuery($this->pdo, $sql, Array($id));
        return $statement->fetch();
    }
    public function getAll()
    {
        $sql = self::$baseSQL . self::$constraint;
        $statement = DatabaseHelper::runQuery($this->pdo, $sql, null);
        return $statement;
    }
}
?>

```

```

<?php
/* Handles database access for the galleries table. */
//GalleryDB.class.php
class GalleryDB
{
    private static $baseSQL = "SELECT GalleryID, GalleryName, GalleryNativeName, GalleryCity,
GalleryCountry, Latitude, Longitude, GalleryWebSite FROM Galleries ";

    private static $constraint = ' order by GalleryName';
    private $pdo = null;

    public function __construct($connection) {
        $this->pdo = $connection;
    }
    public function getAll()
    {

```

```

        $sql = self::$baseSQL . self::$constraint;
        $statement = DatabaseHelper::runQuery($this->pdo, $sql, null);
        return $statement;
    }
    public function findById($id)
    {
        $sql = self::$baseSQL . " WHERE GalleryID=?";
        $statement = DatabaseHelper::runQuery($this->pdo, $sql, Array($id));
        return $statement->fetch();
    }
}
?>

```

```

<?php
/* Handles database access for the Shapes table. */
//ShapeDB.class.php

class ShapeDB
{
    private static $baseSQL = "SELECT ShapeID, ShapeName FROM Shapes ";
    private $pdo = null;

    public function __construct($connection) {
        $this->pdo = $connection;
    }

    public function getAll()
    {
        $statement = DatabaseHelper::runQuery($this->pdo, self::$baseSQL, null);
        return $statement;
    }

    public function findById($id)
    {
        $sql = self::$baseSQL . " WHERE ShapeID=?";
        $statement = DatabaseHelper::runQuery($this->pdo, $sql, Array($id));
        return $statement->fetch();
    }
}
?>

```

```

<?php
/* Handles database access for the Painting table. */
//PaintingDB.class.php
class PaintingDB
{

```

```

private static $baseSQL = "SELECT PaintingID, Paintings.ArtistID AS ArtistID, FirstName, LastName,
Paintings.GalleryID AS GalleryID, Galleries.GalleryName AS GalleryName, ImageFileName, Title, ShapeID,
MuseumLink, AccessionNumber, CopyrightText, Description, Excerpt, YearOfWork, Width, Height,
Medium, Cost, MSRP, GoogleLink, GoogleDescription, WikiLink FROM Paintings INNER JOIN Artists ON
Paintings.ArtistID = Artists.ArtistID INNER JOIN Galleries ON Paintings.GalleryID = Galleries.GalleryID ";

private static $constraint = ' order by YearOfWork limit 20';

private $pdo = null;

public function __construct($connection) {
    $this->pdo = $connection;
}

public function getAll()
{
    $sql = self::$baseSQL . self::$constraint;
    $statement = DatabaseHelper::runQuery($this->pdo, $sql, null);
    return $statement;
}

public function findById($id)
{
    $sql = self::$baseSQL . " WHERE PaintingID=?" ;
    $statement = DatabaseHelper::runQuery($this->pdo, $sql, Array($id));
    return $statement->fetch();
}

public function findByArtist($artistID)
{
    $sql = self::$baseSQL . " WHERE Paintings.ArtistID=?" . self::$constraint;
    $statement = DatabaseHelper::runQuery($this->pdo, $sql, Array($artistID));
    return $statement;
}

public function findByGallery($galleryID)
{
    $sql = self::$baseSQL . " WHERE GalleryID=?" . self::$constraint;
    $statement = DatabaseHelper::runQuery($this->pdo, $sql, Array($galleryID));
    return $statement;
}

public function findByShape($shapeID)
{
    $sql = self::$baseSQL . " WHERE ShapeID=?" . self::$constraint;
    $statement = DatabaseHelper::runQuery($this->pdo, $sql, Array($shapeID));
    return $statement;
}

```

```

public function findByGenre($genreID)
{
    $sql = "SELECT Paintings.PaintingID as ID, PaintingGenres.GenreID, ImageFileName, Title FROM
Paintings INNER JOIN PaintingGenres ON Paintings.PaintingID = PaintingGenres.PaintingID WHERE
PaintingGenres.GenreID=? ORDER BY YearOfWork" ;
    $statement = DatabaseHelper::runQuery($this->pdo, $sql, Array($genreID));
    return $statement;
}
}
?>

```

<?php

//browse-paintings.php

include 'includes/art-config.inc.php';

include 'includes/art-functions.inc.php';

session_start(); //http://php.net/manual/en/function.session-start.php

try {

// connect and retrieve data for filters

\$artistDB = new ArtistDB(\$pdo); **//ArtistDB.class.php**

\$artists = \$artistDB->getAll();

\$galleryDB = new GalleryDB(\$pdo); **//GalleryDB.class.php**

\$galleries = \$galleryDB->getAll();

\$shapeDB = new ShapeDB(\$pdo); **//ShapeDB.class.php**

\$shapes = \$shapeDB->getAll();

// now retrieve paintings ... either all or a subset

\$paintDB = new PaintingDB(\$pdo); **//PaintingDB.class.php**

// filter by artist?

if (isset(\$_GET['artist']) && ! empty(\$_GET['artist'])) {

 \$paintings = \$paintDB->findByArtist(\$_GET['artist']);

 \$artist = \$artistDB->findById(\$_GET['artist']);

 \$filter = 'Artist = ' . makeArtistName(\$artist['FirstName'],\$artist['LastName']) ;

}

// filter by museum?

if (isset(\$_GET['museum']) && ! empty(\$_GET['museum'])) {

 \$paintings = \$paintDB->findByGallery(\$_GET['museum']);

 \$museum = \$galleryDB->findById(\$_GET['museum']);

 \$filter = 'Museum = ' . utf8_encode(\$museum['GalleryName']);


```

}

// filter by shape?
if (isset($_GET['shape']) && ! empty($_GET['shape'])) {
    $paintings = $paintDB->findByShape($_GET['shape']);

    $shape = $shapeDB->findById($_GET['shape']);
    $filter = 'Shape = ' . $shape['ShapeName'];
}

if (! isset($paintings) || $paintings->rowCount() == 0) {
    $paintings = $paintDB->getAll();
    $filter = "All Paintings [Top 20]";
}
}
catch (PDOException $e) {
    die( $e->getMessage() );
}
?>

<!DOCTYPE html>
<html lang=en>
<head>
    <meta charset=utf-8>
    <link href='http://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet' type='text/css'>
    <link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>
    <script src="css/semantic.js"></script>
    <script src="js/misc.js"></script>

    <link href="css/semantic.css" rel="stylesheet" >
    <link href="css/icon.css" rel="stylesheet" >
    <link href="css/styles.css" rel="stylesheet">
</head>
<body >

<?php include 'includes/art-header.inc.php'; ?>

<main class="ui segment doubling stackable grid container">
    <section class="four wide column">
        <?php include 'includes/browse-filters.inc.php'; ?>
    </section>
    <section class="twelve wide column">
        <h1 class="ui header">Paintings</h1>
        <h3 class="ui sub header"><?php echo $filter; ?></h3>
        <ul class="ui divided items" id="paintingsList">

```

```

<?php while ($work = $paintings->fetch()) { ?>
<li class="item">
    <a class="ui small image" href="single-painting.php?id=<?php echo $work['PaintingID']; ?>"></a>
    <div class="content">
        <a class="header" href="single-painting.php?id=<?php echo $work['PaintingID']; ?>"><?php
echo utf8_encode($work['Title']); ?></a>
        <div class="meta"><span class="cinema"><?php echo
makeArtistName($work['FirstName'],$work['LastName']); ?></span></div>
        <div class="description">
            <p><?php echo utf8_encode($work['Excerpt']); ?></p>
        </div>
        <div class="meta">
            <strong><?php echo '$' . number_format($work['MSRP'],0); ?></strong>
        </div>
        <div class="extra">
            <a class="ui icon orange button" href="cart.php?id=<?php echo $work['PaintingID']; ?>"><i
class="add to cart icon"></i></a>
            <a class="ui icon button" href="addToFavorites.php?id=<?php echo $work['PaintingID']
?>&path=<?php echo $work['ImageFileName'] ?>&title=<?php echo urlencode($work['Title']) ?>"><i
class="heart icon"></i></a>
        </div>
    </div>
</li>

<?php } ?>

</ul>
</section>

</main>

<footer class="ui black inverted segment">
    <div class="ui container">footer for later</div>
</footer>
</body>
</html>

```

```

//browse-filters.inc.php
<form class="ui form" method="get" action="browse-paintings.php">
  <h3 class="ui dividing header">Filters</h3>

  <div class="field">
    <label>Artist</label>
    <select class="ui fluid dropdown" name="artist">
      <option value='0'>Select Artist</option>
      <?php
        outputFilterOptions($artists, 'ArtistID', 'LastName');
      ?>
    </select>
  </div>
  <div class="field">
    <label>Museum</label>
    <select class="ui fluid dropdown" name="museum">
      <option value='0'>Select Museum</option>
      <?php
        outputFilterOptions($galleries, 'GalleryID', 'GalleryName');
      ?>
    </select>
  </div>
  <div class="field">
    <label>Shape</label>
    <select class="ui fluid dropdown" name="shape">
      <option value='0'>Select Shape</option>
      <?php
        outputFilterOptions($shapes, 'ShapeID', 'ShapeName');
      ?>
    </select>
  </div>
  <button class="small ui orange button" type="submit">
    <i class="filter icon"></i> Filter
  </button>
</form>

```