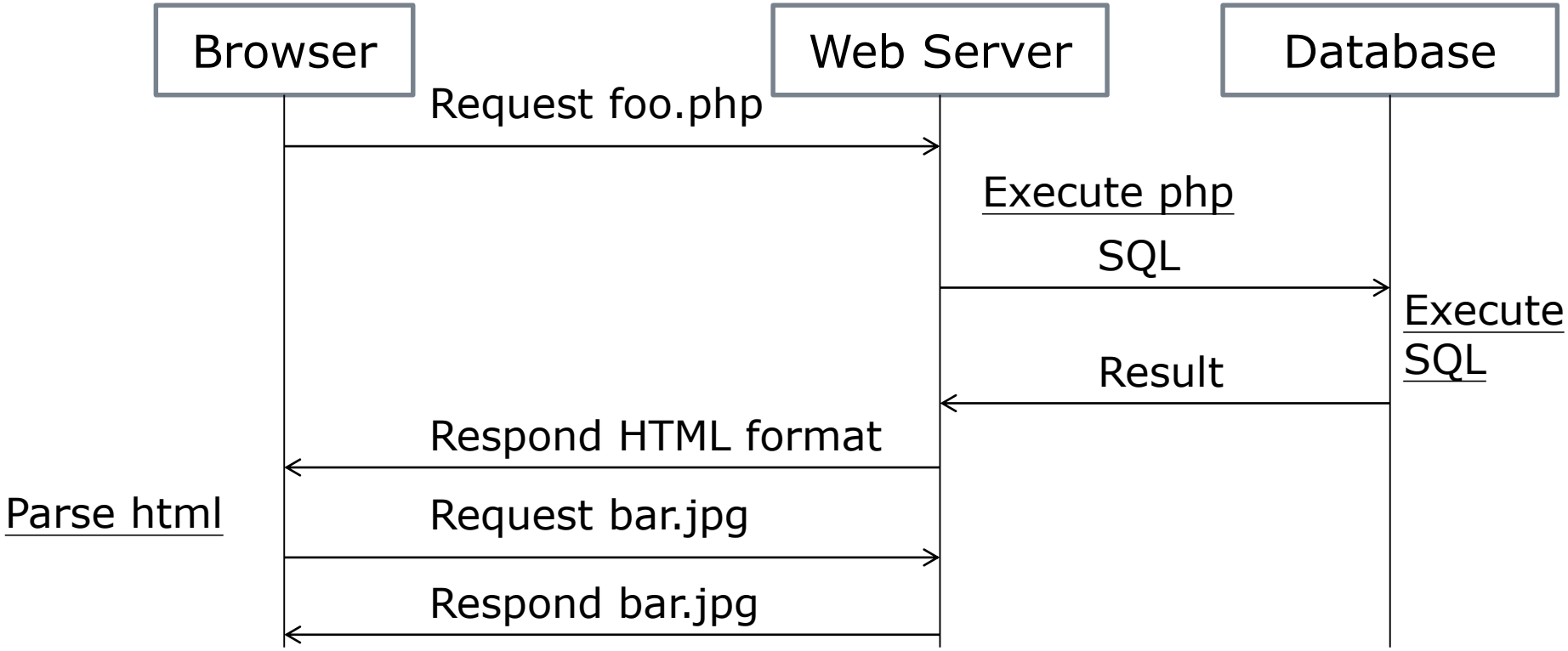
A Very Brief Introduction to Web Application Development (II)

Jiun-Long Huang (黃俊龍)
Department of Computer Science
National Chiao Tung University

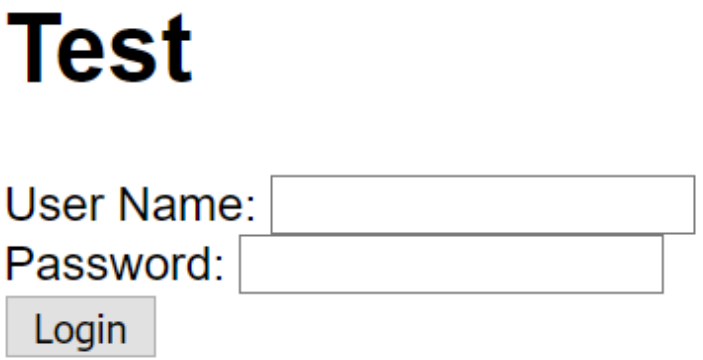
Dynamic Web Application 1.0: Back End Approach

Control Flow



- Data transfer between php files
  - GET
  - POST
  - Session
- SQL
  - Prepared statement

HTML Form




```
<!DOCTYPE html>
<html>
<body>
<h1>Test</h1>
<form action="page1.php" method="get">
  User Name:
  <input type="text" name="uname"><br>
  Password:
  <input type="password" name="pwd"><br>
  <input type="submit" value="Login">
</form>
</body>
</html>
```

Get and Post

- Get
  - After the button has been clicked, the browser will request the URL https://localhost/exam_test/page1.php?uname=jlhuang&pwd=jlhuang
- Post
  - uname=jlhuang&pwd=jlhuang appears in the HTTP request message

When to use GET?

- Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL).
- GET also has limits on the amount of information to send.
  - The limitation is about 2000 characters.
- GET is used in some comic web sites.

- However, because the variables are displayed in the URL, it is possible to bookmark the page.
- Note: GET should NEVER be used for sending passwords or other sensitive information!

When to use POST?

- Information sent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send.
- Moreover POST supports advanced functionality such as support for multi-part binary input while uploading files to server.
- However, because the variables are not displayed in the URL, it is not possible to bookmark the page.

- Use $_SERVER['REQUEST_METHOD'] to get the request method
  - Which request method was used to access the page; e.g. 'GET', 'HEAD', 'POST', 'PUT’.
- $_GET: HTTP GET variables

```
<?php
echo 'Hello ' . $_GET["name"]). '!';
?>
```

- $_POST: HTTP POST variables

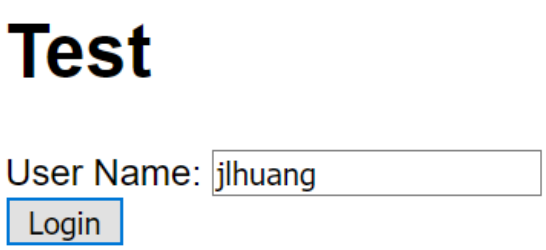
```
<?php
echo 'Hello ' . $_POST["name"] . '!';
?>
```

- $_REQUEST — HTTP Request (GET and POST) variables

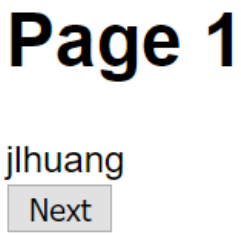
```
<?php
echo 'Hello ' . $_REQUEST["name"] . '!';
?>
```

Example

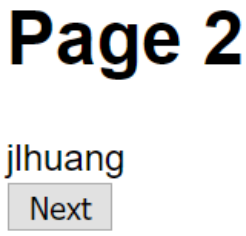
- index.php



- page1.php



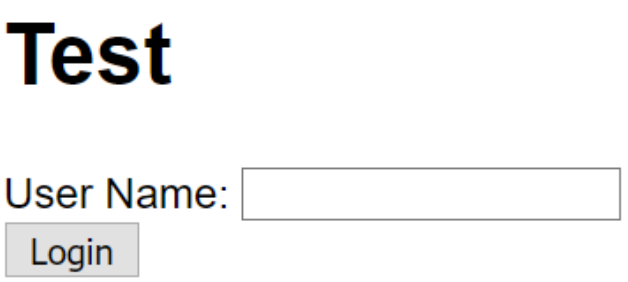
- page2.php



Transmitting Data via URL

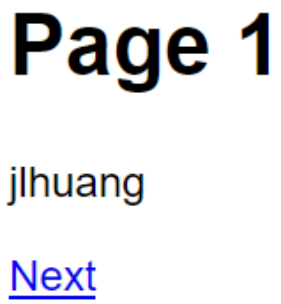
index_url.php

```
<!DOCTYPE html>
<html>
<body>
<h1>Test</h1>
<form action="page1_url.php"
method="get">
  User Name:
  <input type="text" name="uname"><br>
  <input type="submit" value="Login">
</form>
</body>
</html>
```



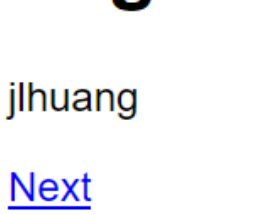
page1_url.php

```
<?php
  $uname=$_REQUEST['uname'];
  echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
        <h1>Page 1</h1>
        <p>$uname</p>
          <a href="page2_url.php?uname=$uname">
            Next</a>
        </form>
       </body>
     </html>
EOT;

?>
```



page2_url.php

```
<?php
  $uname=$_REQUEST['uname'];
  echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
        <h1>Page 1</h1>
        <p>$uname</p>
          <a href="page3_url.php?uname=$uname">
            Next</a>
        </form>
       </body>
     </html>
EOT;

?>
```



Transmitting Data via Hidden Input

index_hidden.php

```
<!DOCTYPE html>
<html>
<body>
<h1>Test</h1>
<form action="page1_hidden.php" method="get">
  User Name:
  <input type="text" name="uname"><br>
  <input type="submit" value="Login">
</form>
</body>
</html>
```

page1_hidden.php

```
<?php
  $uname=$_REQUEST['uname'];
  echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
        <h1>Page 1</h1>
           <form action="page2_hidden.php" method="get">
        <label>$uname</label>
          <input type="hidden" name="uname"
             value="$uname"><br>
          <input type="submit" value="Next">
        </form>
         </body>
       </html>
EOT;

?>
```

page2_hidden.php

```
<?php
  $uname=$_REQUEST['uname'];
  echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
        <h1>Page 2</h1>
           <form action="page3.php" method="get">
        <label>$uname</label>
          <input type="hidden" name="uname"
            value="$uname"><br>
          <input type="submit" value="Next">
        </form>
         </body>
       </html>
EOT;

?>
```

Keeping Data in Session

Session

- Problems of the above methods:
  - Inefficient:
    - Data are transmitted between browser and web server back and forth
  - Insecure:
    - Some sensitivity data are sent to browser
    - E.g., the user level (Normal, VIP…)
    - Users can modify them by the development tools provided by browser
- Session variables are stored in web server

$_SESSION — Session variables

```
<?php
session_start();
$_SESSION['foo']='bar';
echo $_SESSION['foo'];
session_destroy();
?>
```

index_session.php

```
<!DOCTYPE html>
<html>
<body>
<h1>Test</h1>
<form action="page1_session.php" method="get">
  User Name:
  <input type="text" name="uname"><br>
  <input type="submit" value="Login">
</form>
</body>
</html>
```

page1_session.php

```
<?php
  session_start();
  $uname=$_REQUEST['uname'];
  $_SESSION['uname']=$uname;
  echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
        <h1>Page 1</h1>
        <p>$uname</p>
          <a href="page2_session.php>Next</a>
        </form>
         </body>
       </html>
EOT;

?>
```

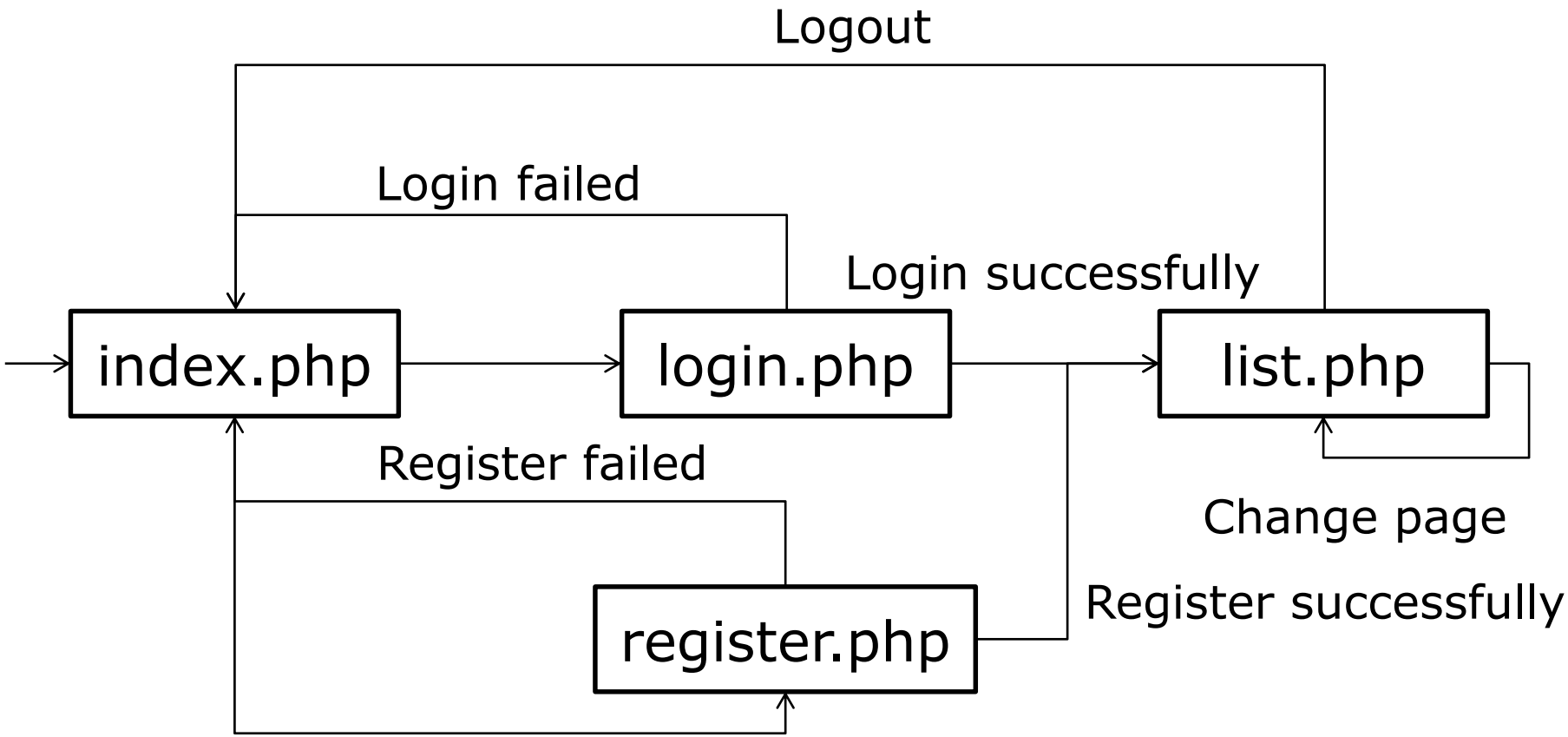
page2_session.php

```
<?php
  $uname=$_SESSION['uname'];
  echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
        <h1>Page 2</h1>
        <p>$uname</p>
        <a href="page3_session.php>Next</a>
        </form>
         </body>
       </html>
EOT;

?>
```

PHP Database Access

- PDO
- MySQLi
  - 'i' means improvement
- MySQL (deprecated in PHP 5.5.0)
  - mysql_connect, mysql_select_db
- PDO is better

An Example


Logout
Login failed
Login successfully
index.php
login.php
list.php
Register failed
Change page
Register successfully
register.php

Hash

- Do not use MD5 and SHA-1 for authentication
  - MD5 has been broken by Tao Xie and Dengguo Feng in 2009
  - SHA-1 has been broken by Google in 2017
- SHA-2 is preferred
  - SHA-224、SHA-256、SHA-384、SHA-512、SHA-512/224、SHA-512/256
- salt

Table users

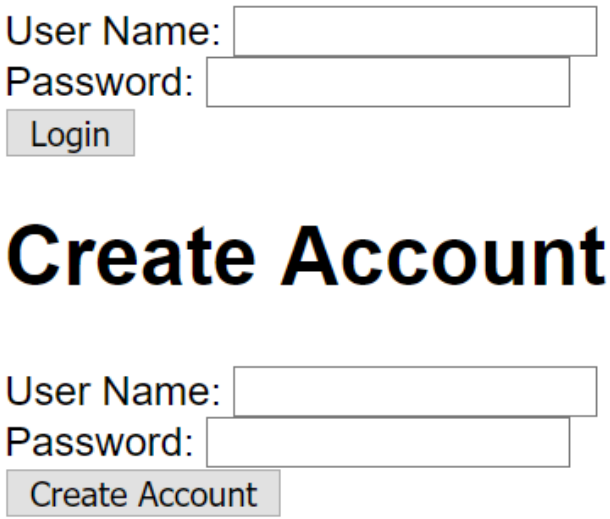
- Column
  - user_id: int(11), AUTO_INCREMENT, Primary Key
  - username: char(40) , Unique
  - password: char(64)
    - The length of the result of shar256 is 32 bytes (64 characters in hex mode)
  - salt: char(4)
- Index
  - user_id, username

Table posts

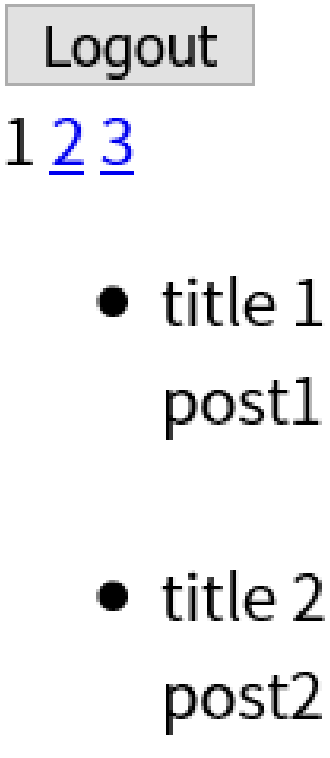
- Column
  - post_id: int(11), AUTO_INCREMENT, Primary Key
  - title: char(30)
  - content: char(100)
- Index
  - post_id

Screenshots

- Login/Create Account



- Browse



index.php

```php
<?php
  session_start();
  # remove all session variables
  session_unset();
  # destroy the session
  session_destroy();
  $_SESSION['Authenticated']=false;
?>

<!DOCTYPE html>
<html>
<body>
<h1>Login</h1>
<form action="login.php" method="post">
  User Name:
  <input type="text" name="uname"><br>
  Password:
  <input type="password" name="pwd"><br>
  <input type="submit" value="Login">
</form>
```

```
<h1>Create Account</h1>
<form action="register.php" method="post">
  User Name:
  <input type="text" name="uname"><br>
  Password:
  <input type="password" name="pwd"><br>
  <input type="submit" value="Create Account">
</form>

</body>
</html>
```

register.php

```
<?php
session_start();
$_SESSION['Authenticated']=false;

$dbservername='localhost';
$dbname='examdb';
$dbusername='examdb';
$dbpassword='examdb';

try {
  if (!isset($_POST['uname']) || !isset($_POST['pwd']))
  {
    header("Location: index.php");
    exit();
  }
  if (empty($_POST['uname']) || empty($_POST['pwd']))
    throw new Exception('Please input user name and
password.');
```

```
  $uname=$_POST['uname'];
  $pwd=$_POST['pwd'];
  $conn = new PDO("mysql:host=$dbservername;dbname=$dbname",
                  $dbusername, $dbpassword);
  # set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE,
                      PDO::ERRMODE_EXCEPTION);

  $stmt=$conn->prepare("select username from users where
username=:username");
  $stmt->execute(array('username' => $uname));

  if ($stmt->rowCount()==0)
  {
    $salt=strval(rand(1000,9999));

    $hashvalue=hash('sha256', $salt.$pwd);
```

```
    $stmt=$conn->prepare("insert into users (username,
      password, salt) values (:username, :password, :salt)");
    $stmt->execute(array('username' => $uname,
      'password' => $hashvalue, 'salt' => $salt));
    $_SESSION['Authenticated']=true;
    $_SESSION['Username']=$uname;
    echo <<<EOT
      <!DOCTYPE html>
      <html>
        <body>
            <script>
            alert("Create an account successfully.");
            window.location.replace("list.php");
          </script>  </body>  </html>
EOT;
    exit();
  }
  else
    throw new Exception("Login failed.");
}
```

```
catch(Exception $e)
{
  $msg=$e->getMessage();
  session_unset();
  session_destroy();

  echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
          <script>
          alert("$msg");
          window.location.replace("index.php");
        </script>
      </body>
      </html>
EOT;
}
?>
</body>
</html>
```

login.php

```
<?php
session_start();

$_SESSION['Authenticated']=false;

$dbservername='localhost';
$dbname='examdb';
$dbusername='examdb';
$dbpassword='examdb';

try
{
  if (!isset($_POST['uname']) || !isset($_POST['pwd']))
  {
    header("Location: index.php");
    exit();
  }
  if (empty($_POST['uname']) || empty($_POST['pwd']))
    throw new Exception('Please input user name and
password.');
```

```
  $uname=$_POST['uname'];
  $pwd=$_POST['pwd'];
  $conn = new PDO("mysql:host=$dbservername;dbname=$dbname",
                  $dbusername, $dbpassword);
  # set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE,
                      PDO::ERRMODE_EXCEPTION);
  $stmt=$conn->prepare("select username, password, salt
from users where username=:username");
  $stmt->execute(array('username' => $uname));

  if ($stmt->rowCount()==1)
  {
    $row = $stmt->fetch();
```

```
    if ($row['password']==
        hash('sha256',$row['salt'].$_POST['pwd']))
    {
      $_SESSION['Authenticated']=true;
      $_SESSION['Username']=$row[0];
      header("Location: list.php?page=1");
      exit();
    }
    else
      throw new Exception('Login failed.');
  }
  else
     throw new Exception('Login failed.');
}
```

```
catch(Exception $e)
{
  $msg=$e->getMessage();
  session_unset();
  session_destroy();
  echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
          <script>
          alert("$msg");
               window.location.replace("index.php");
        </script>
         </body>
       </html>
EOT;
}
?>
```

list.php

```
<?php
  session_start();

  $dbservername='localhost';
  $dbname='examdb';
  $dbusername='examdb';
  $dbpassword='examdb';

try
{
  if (!isset($_SESSION['Authenticated']) ||
              $_SESSION['Authenticated']!=true)
  {
    header("Location: index.php");
    exit();
  }

  if (isset($_GET['page']))
    $page=$_GET['page'];
  else
    $page=1;
```

```
  $postperpage=2;
  $conn = new PDO("mysql:host=$dbservername;dbname=$dbname",
                  $dbusername, $dbpassword);
  $conn->setAttribute(PDO::ATTR_ERRMODE,
                      PDO::ERRMODE_EXCEPTION);
  $stmt=$conn->prepare("select count(*) from posts");
  $stmt->execute();
  $row=$stmt->fetch();
  $totalpage=ceil($row[0]/$postperpage);

  echo <<<EOT
    <!DOCTYPE html>
      <html>
      <body>
        <button type="button"
           onClick="window.location.replace('index.php');">
           Logout</button><br>
EOT;
```

```
  echo <<<EOT
    <!DOCTYPE html>
      <html>
      <body>
        <button type="button"
          onClick="window.location.replace('index.php');">
          Logout</button><br>
EOT;

  if ($totalpage>0)
  {
    for($i=1;$i<=$totalpage;$i++)
    {
      if ($i==$page)
        echo "$i ";
      else
        echo "<a href='list.php?page=$i'>$i</a> ";
    }
    echo '<br>';
    $startrow=($page-1)*$postperpage;
```

```
  $stmt=$conn->prepare("select title, content from posts
limit $startrow,2");
    $stmt->execute();
    echo '<ul>';
    while($row=$stmt->fetch())
      echo '<li> ' . $row['title'] . '<br>' .
$row['content'] . '<br><br> </li>';
  }
  echo '</ul></body></html>';
}
```

```
catch (PDOException $e)
{
  session_unset();
  session_destroy();

  echo <<<EOT
    <!DOCTYPE html>
      <html>
        <body><script>
          alert("Internal Error. $msg");
          window.location.replace("index.php");
        </script></body>
      </html>
EOT;
}
?>
```

Transactions in PHP

- Table accounts
  - AccountName: char(10) not null
  - Amount: int(11) not null

index.php

```
<!DOCTYPE html>
<html>
<body>
<h1>Transaction Test</h1>
<form action="test.php" method="post">
  <input type="submit" value="Transfer">
</form>
</body>
</html>
```

test.php

```
<?php

$dbservername='localhost';
$dbname='examdb';
$dbusername='examdb';
$dbpassword='examdb';

try
{
  $conn = new PDO("mysql:host=$dbservername;dbname=$dbname",
                  $dbusername, $dbpassword);
  # set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE,
                      PDO::ERRMODE_EXCEPTION);
  $conn->beginTransaction();
  $stmt=$conn->prepare("insert into accounts (AccountName,
Amount) values ('Alice', 100)");
  $stmt->execute();
  $stmt=$conn->prepare("insert into accounts (AccountName,
Amount) values ('Bob', null)");
```

```
 $stmt->execute();
  $conn->commit();
    echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
          <script>
            alert("Insertion OK");
            window.location.replace("index.php");
          </script>
      </body>
    </html>
EOT;
}
```

```
catch(Exception $e)
{
  if ($conn->inTransaction())
    $conn->rollBack();
  $msg=$e->getMessage();
  echo <<<EOT
    <!DOCTYPE html>
    <html>
      <body>
        <script>
          alert("$msg");
          window.location.replace("index.php");
        </script>
      </body>
    </html>
EOT;
}
?>
```

Web Development Framework

- Framework
  - CodeIngiter
  - Laravel
  - …