

# Web Engineering: Miniproject

Jesper Riemer Andersen

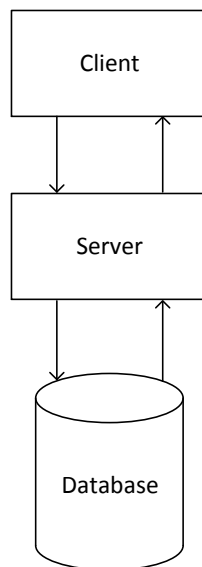
Nicklas Andersen

Simon Reedtz Olesen

September 19, 2013

## 1 Block 1: Introduction

We have simple 3-tier architecture which allows us to communicate with the server and extract data from our database. This protects the database from unwanted queries from the client.



1. The client opens his web browser and sends a HTTP request to the server.
2. The server runs the PHP script which queries data from the database.

3. The database returns the result according to the query.
4. Then the server returns the HTTP response to the client.
5. The response is then output as HTML.

```

1 <?php
2 $connection = mysql_connect("localhost", "root", "");
3 mysql_select_db("webengi", $connection);
4
5 $query = mysql_query("SELECT testtext FROM hello", $connection);
6
7 while ($row = mysql_fetch_assoc($query)) {
8     echo $row['testtext']."<br>";
9 }
10 ?>

```

This code shows the steps explained above. We chose to run the server on localhost using WampServer, which also comes bundled with a MySQL database.

We start by connecting to the localhost with the username *root* and empty password. Then we select the database called *webengi*, and then query the database. For each row that is returned by the query, we echo the content of the column called *testtext*.

## 2 Block 2: Data I: XML etc.

We found an XML document containing a small set of computers for sale on eBay, which we will display a subset of. We will be using the following:

- An XML file containing data on computers for sale.
- An XSL file with our XSL transformations (XSLT).
- A HTML file which contains only a little javascript to load the other files and display the result.

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL
  /Transform">
3
4 <xsl:template match="/">
5   <h2>Computers for sale on eBay</h2>

```

```

6      <table border="1">
7          <tr bgcolor="#9acd32">
8              <th>Current bid</th>
9              <th>Memory</th>
10             <th>Hard drive</th>
11             <th>CPU</th>
12         </tr>
13         <xsl:for-each select="root/listing">
14             <tr>
15                 <td><xsl:value-of select="auction_info/current_bid"/></td>
16                 <td><xsl:value-of select="item_info/memory"/></td>
17                 <td><xsl:value-of select="item_info/hard_drive"/></td>
18                 <td><xsl:value-of select="item_info/cpu"/></td>
19             </tr>
20         </xsl:for-each>
21     </table>
22 </xsl:template>
23 </xsl:stylesheet>

```

This is the XSL file with our transformation. The following box is copied from w3schools.com [1].

Since an XSL style sheet is an XML document, it always begins with the XML declaration: `<?xml version="1.0"encoding="ISO-8859-1"?>`.

The next element, `<xsl:stylesheet>`, defines that this document is an XSLT style sheet document (along with the version number and XSLT namespace attributes).

The `<xsl:template>` element defines a template. The `match="/"` attribute associates the template with the root of the XML source document.

The content inside the `<xsl:template>` element defines some HTML to write to the output.

## References

- [1] w3schools. Xslt tutorial, 2013. [http://www.w3schools.com/xsl/xsl\\_templates.asp](http://www.w3schools.com/xsl/xsl_templates.asp).