The revised The HitServlet.java (included in the zip file) demonstrates how you can create an image gallery. It expects you to have a “c:\tomcat\webapps\midp\images” folder with two images named  Spider-Man.png and blah.jpg in that folder.  These could be any images.  You should save your old HitServlet.java before you compile this one in classes directory.  Once compiled successfully, restart tomcat and access the servlet from the browser as before.  invoke the servlet using the same URL that you have been using e.g. <https://localhost:8081/midp/hits>  You will see a form show up asking for userid and password.  Just type whatever on both text boxes and hit the submit button.  You will see an image show up.  If you press either next or prev buttons, you will see that image is replaced.

I have put together some code to illustrate how AJAX could be integrated in your web application.  You can copy and paste the code given below to any w3school “Run” environment to try it.  You will see that when you click “Try It” button of this code, the “compman.gif” image will be replaced with “smiley.gif” image.  There is an Ajax request that does a GET from w3schools server and fetches some text.  The fetched text is discarded as it is not needed in this particular example but note that the source of the image tag is replaced in the same handler function.

The Auto button should periodically call a method similar to the loadDoc given below.  This method should make a similar Ajax call to a Servlet invoking its doGet method.  The doGet method should return only the name of an image file as a text/plain. The returned image filename then should be assigned to the src of the image tag.

<!DOCTYPE html>  
<html>  
<body>  
<div id="demo">  
<h1>The XMLHttpRequest Object</h1>  
</div>  
<img id="myImg" src="/jsref/compman.gif" width="107" height="98">  
 <p>Click the button to change image.</p>  
 <button onclick="loadDoc()">Try it</button>  
 <script>  
function loadDoc() {  
  var xhttp = new XMLHttpRequest();  
  xhttp.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
      document.getElementById("demo").innerHTML = this.responseText;  
      document.getElementById("myImg").src =  "<https://www.w3schools.com/jsref/smiley.gif>";  
    }  
  };  
  xhttp.open("GET", "/xml/ajax\_info.txt", true);  
  xhttp.send();  
}  
</script>  
 </body>  
</html>

Also, for future needs, given below is how you invoke a file uploader.  If you put this code any any HTML page, you will be able to see the file uploader functionality supported by HTML.

<form method="post" enctype="multipart/form-data">  
<div>  
<label for="file">Choose file to upload</label>  
<input type="file" id="file" name="file" multiple>  
</div>  
<div>  
<button>Submit</button>  
</div>  
</form>

Top of Form

I have also included a folder “upload” in the zip file.  Copy this folder into tomcat\webapps (just like you did to midp project).   Start tomcat and then on the browser enter <http://localhost:8081/upload> .  This will bring a form.  Just hit submit and you will now see a file upload form.  Select an image from you file system and press submit.  If browser says successfully uploaded, you should be able to see the file in the c:\tomcat\webapps\upload\images folder. You can see the listing of this folder on the browser by typing <http://localhost:8081/upload> /upload

Note the technique used in the index.html and index2.html to navigate from one HTML page to another.  Check out the following for other possible alternatives where server side assists in navigation:

<https://www.javatpoint.com/sendRedirect()-method>

Also compile and run JDBC\_ComplexData.java file that demonstrates how to insert and retrieve records that are composed of UUID, Date and BLOB types in Oracle database. The code expects you to have an image file of a specific name. Once you have that file in the folder, the code should run as I showed you in the lab. Adapt the code for mysql if you are using mysql instead.