SESSION: **DIET: FIRST**

Module Title Rich Internet Applications

(Module Code) MHG412774

LEVEL: H
MODULE LEADER: Martin L Gallacher

JANUARY 2017 **DURATION: 2 Hours**

CANDIDATES SHOULD ATTEMPT 3 QUESTIONS
PLEASE READ THE QUESTIONS CAREFULLY

MATERIALS TO BE SUPPLIED/ALLOWED:

Lined Examination Script Books ☐

Unlined Examination Script Books ☐

Other Materials, e.g. Graph paper, statistical tables (please specify)

_____

(Exemplars)
Question Paper (Supplied)
Calculator (Allowed)

Q.1 **(a)** With reference to the sites listed below describe the use of characteristics & features of Web 2.0 and Rich Internet Applications:

(i) Reddit
(ii) FaceBook
(iii) Twitter **[12]**

(b)

(i) With reference to the code & screenshots in Appendix A, explain the purpose and operation of the JavaScript function **`getArtistAlbumsCallback()`**. **[8]**

(ii) Write JavaScript code to implement the **`check()`** functionality as illustrated in the screenshots in Appendix A. **[5]**

Q.2 **(a)** With reference to the scenario described in Appendix B select a functional requirement and critically appraise the suitability of each Ajax message pattern option. **[8]**

(b) It has been decided to use XML instead of JSON for formatting the data returned from both PHP scripts depicted in Appendix C. Write the same information as shown in the example formatted using XML and outline the changes required to the script in Appendix A. **[9]**

(c) Both PHP scripts return related data; produce a single JSON document and a single XML document which combines the data from the two JSON excerpts depicted in Appendix C. **[8]**

PTO

Q.3 **(a)** Identify and explain suitable RESTful URLs for the following functionality:

(i) return a list of all artists;
(ii) create a new artist record;
(iii) delete the fifth artist;
(iv) update the fourth artist. **[8]**

(b) Explain the HTTP Access Control (CORS) giving examples of situations where it applies. **[8]**

(c) Explain how Amazon Developer or Google Developer Services could be used by a small business web site. **[9]**

Q.4 **(a)** Explain how the JavaScript functions **`getArtists()`** and **`getArtistsCallback()`** in Appendix A could be rewritten using jQuery. **[9]**

(b) With reference to the scenario in Appendix B, suggest and justify your choice of jQuery plugins to implement a functional requirement. **[8]**

(c) Produce a report highlighting FOUR new features to ECMAScript 6. **[8]**

PTO

Q.5 **(a)** With reference to the scenario described in Appendix B justify your selection of THREE features of HTML 5 to implement. [**9]**

(b) With reference to the scenario described in Appendix B, evaluate, using examples, the benefits of TWO Ajax Functionality and Usability patterns. **[6]**

(c) With reference to the scenario described in Appendix B, discuss Rich Internet Application usability and accessibility challenges. **[10]**

END OF EXAM PAPER

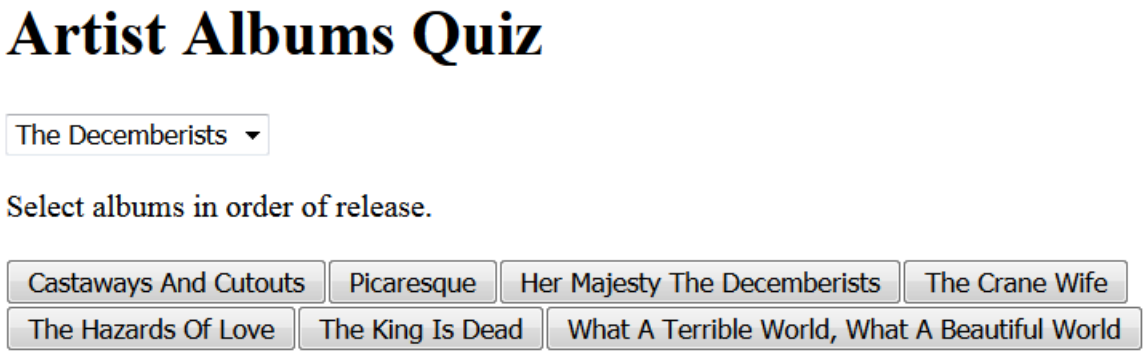
Appendix A

```
<html>
<head>
   <title>Music Collection</title>
   <script>
      var req;

      function getArtists()
      {
         var url = "artists.php";
         req = new XMLHttpRequest();

         req.open("GET", url, true);
         req.setRequestHeader("Accept",
                              "application/json; charset=utf-8");
         req.onreadystatechange = getArtistsCallBack;
         req.send(null);
      }

      function getArtistsCallBack()
      {
         if (req.readyState == 4) {
            if (req.status == 200) {
               var response = req.responseText.parseJSON();
               var artistSelect =
                   document.getElementById("artistslist");
               for (i=0;i<response.artistsarray.length;i++){
                  var artist_id =
                        response.artistsarray[i].artist_id;
                  var artist_name =
                        response.artistsarray[i].artist_name;
                  artistSelect.options[artistSelect.options.length]=
                     new Option(artist_name, artist_id,
                                false, false);
               }
            }
         }
      }

      function getArtistAlbums()
      {
         var selectedArtist =
               document.getElementById("artistslist");
         var url = "artistalbums.php?artist_id=" +
                                 escape(selectedArtist.value);
         req = new XMLHttpRequest();
         req.open("GET", url, true);
         req.setRequestHeader("Accept",
                              "application/json; charset=utf-8");
         req.onreadystatechange = getArtistAlbumsCallBack;
         req.send(null);
```

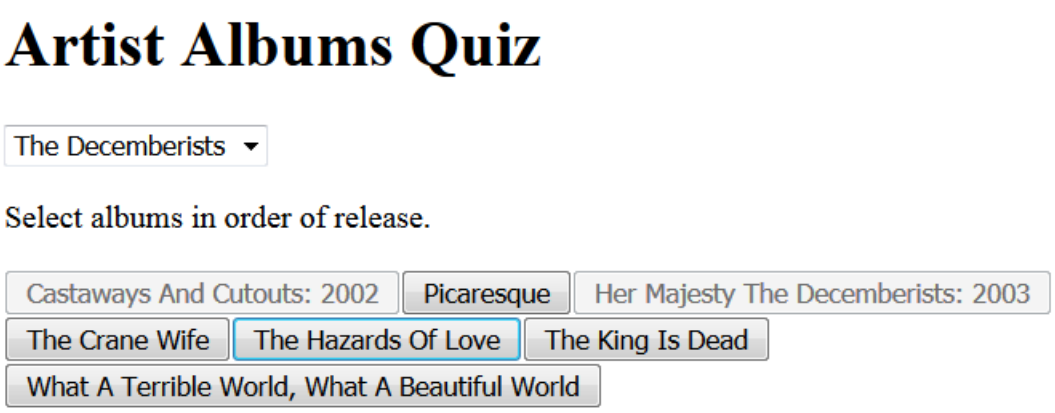
```
      }

      var albumsarray;
      var nextPosition;

      function getArtistAlbumsCallBack()
      {
         if (req.readyState == 4) {
            if (req.status == 200) {
               document.getElementById("albums").innerHTML="";
               nextAlbum=1;
               var response = req.responseText.parseJSON();
               albumsarray = response.albumsarray;
               for (i=0;i<albumsarray.length;i++){
                  var album_title = albumsarray[i].album_title;
                  var year_released = albumsarray[i].year_released;
                  var release_order = albumsarray[i].release_order;
                  var btn = document.createElement("BUTTON");
                  var t = document.createTextNode(album_title);
                  btn.appendChild(t);
                  btn.setAttribute("id", "button" + i);
                  var functionClick = "check('" + i + "')";
                  btn.setAttribute("onclick", functionClick);
                 document.getElementById("albums").appendChild(btn);
               }
            }
         }
      }

      function check(number) {

         // Test to see whether this album is the next album
         // in release order
         // If it is:
         //    Adjust next album value
         //    Disable button
         //    Update button text to include year of release
         // If not:
         //    Pop up wrong message
      }
   </script>
</head>
```

```
<body onLoad="getArtists()">
   <h1>Artist Albums Quiz</h1>
   <form id="artistform" action="">
      <p>
         <select id="artistslist" onChange="getArtistAlbums()">
            <option selected="selected">Pick artist</option>
         </select>
      </p>
   </form>
   <p>Select albums in order of release.</p>
   <div id="albums"></div>
</body>
</html>
```

Screenshot 1



Screenshot 2



Appendix B

Ticket Agency website

A Single Page Application is required for a ticket agency to provide information on artist tours including venues where a tour is visiting, artists and artist albums.

The following basic functional requirements are required:

1. Search facilities by tour, artist, venue with drill down through initial results

2. Map based browsing with interaction on selecting a venue to view images and information.

Appendix C

Response from artists.php


```
{
   "artistsarray":[
                    {"artist_id":"1", "artist_name":"The Decemberists"},
                    {"artist_id":"2", "artist_name":"Kate Bush"},
                    {"artist_id":"3", "artist_name":"Josh Rouse"},
                    …
                   ]
}
```


Response from artistalbums.php?artist_id=1


```
{
   "albumsarray":[
                  {"album_title":"Castaways And Cutouts",
                   "year_released":"2002","release_order":"1"},
                  {"album_title":"Picaresque",
                   "year_released":"2005","release_order":"3"},
                  {"album_title":"Her Majesty The Decemberists",
                   "year_released":"2003","release_order":"2"},
                   …
                  ]
}
```