Advanced Internet Technologies Project Report

Group members: Mercy Lio, Alexandru Stan

Abstract — The project as required for the completion of the Advanced Internet Technologies course is to develop a web application using the concepts learned during the lectures. The application is to focus on the presentation of open data to create an execution in which we can view certain data through the use of Databases, and web development languages such as PHP/MySQL, or Java and MS SQL-Server platforms as part of the implementation solution. The aim of the project is such that the data sets as provided by the open data websites (<http://daten.buergernetz.bz.it> , <http://dati.retecivica.bz.it> , <http://data.gov.uk> , etc.) should be visualized using a website as the interface to the users. And this website shall permit sending of messages by the users to a server which will process these requests. This server should establish a connection to a database which in turn contains tables for the data and the users.

Index Terms— Advanced Internet Technologies, HTML, PHP, CSS, JS, Open Data.

INTRODUCTION

The goals as developed from the brief abstract are:

To visualize data sets using a website as the users’ interface.

* This will contain a start page that gives an overview of the sets offered. There should be a possibility to request details of a particular set (simple off-set-shelf visualization should be sufficient).
* A shopping basket feature should be implemented that allows the users to select data for later export in a format like csv.
* A registration feature should be provided and allow the user to enter the name, address, and other contact details. Inputs should be checked for consistency.

The data presentation and registration should be on separate HTML pages. The solution should be responsive, using CSS and JQuery.

There should exist a server that receives messages from the user and processes them. The server has to establish connections to the database.

The database should contain tables holding information concerning the data as well as information concerning the users.

CONCEPTS AND INNOVATION

The web application created for this project is implemented using PHP/MySQL. This application will permit users to request data concerning the people who have had honours bestowed upon them by the queen.

The implementation of this system will use HTML, CSS and JavaScript for the interface of the website in order to make it attractive as well as responsive.

1. HTML

This is a markup language for describing web documents/web pages. It is an abbreviation for Hyper Text Markup Language. It is written using HTML tags which describe different document content. These tags are keywords surrounded by angle brackets: <tagname>content</tagname>.

Tags always come in pairs i.e you always have a start tag/opening tag: <tagname> and an end tag/closing tag: </tagname>. The end tag has a slash at the beginning of the tag name.

HTML is read and rendered by web browsers whose purpose is to use the HTML tags to determine how to display the document.

1. CSS

This is a stylesheet language that describes the presentation of a HTML document. It helps the web page developer to beautify and position elements on the screen.

In a CSS document, a developer would specify the tag (as in the HTML file) that they wish to present as follows: body{background-color:#d0e4fe;}. The html tag here is ‘body’ and the presentation will be whatever is in the curly brackets.

1. JavaScript

This is the programming language of HTML and the Web. Programming makes computers do what you want them to do. Hence, JavaScript is used to program the behaviour of web pages.

1. JQuery

This is a JavaScript Library. It aims to simplify JavaScript programming.

It will also implement a user login and registration capability, brought to life by PHP. This will connect to a database with a table that stores user information. The user will then be able to register and login/logout of the website.

1. PHP

This is a server scripting language. It is used for making dynamic and interactive web pages according to w3schools.com

1. MySQL

SQL is a standard language for accessing databases. There are many database systems such as Access, MySQL, Oracle, SQL Server, among others.

For our project, we decided to use MySQL as it is better known to us as a group.

1. localhost

This is a hostname that means this current working computer. It is used to access the network services that are running on the host through a network interface.

1. REST services

Representational State Transfer (REST) is an architectural style (of the world wide web) that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, simplicity, visibility, portability, reliability and modifiability which enable services to work best on the Web according to the Oracle Java EE 6 Tutorials.

To the extent that a system conforms to the constraints of REST, they can be called RESTful. RESTful systems usually communicate over Hypertext Transfer Protocol (HTTP) that use the verbs GET, POST, PUT, DELETE, etc. that web browsers use to retrieve web pages and to send data to remote servers.

1. Charts.js – this is
2. Shopping Carts - this

From all these concepts, the project meets the requirements set. These are:

* To have a responsive interface.
* To have a nice interface layout.
* To have input checks.
* To have bidirectional data transfer between the user interface (UI) and the database (DB).

This project will be created using some of the above mentioned concepts to create a website that will provide data pertaining to the Queen’s Honours List. The website is created as follows:

* The initial page (the index page), provides basic information about what data sets are being provided on the site. It provides some insight into the website and what it’s about, among other things.
* On this page, there is also the ability to login or register in order to actually get the data the site provides.
* Once logged in, the user is able to access the page with the data set information.
* This page calls information that is stored in a database and renders it in form of responsive charts, as well as a brief description of the data provided.
* The user is also able to add the information that they need to a shopping cart which then allows them to ‘buy’ the data set and then later export it into a csv file.

CONCLUSION AND FUTURE WORK

Requirements

* Responsive interface
* Nice interface layout **(fancy? with flowers & rainbows? =) )**
* Input data checks:
* Bidirectional data transfer between the UI and the DB
* Open Data - source: can be anything we like (e.g. UK Open Data - on a quest to be cool and interesting)

Functional Requirements

* Overview of data sets
* Request details of sets
* shopping basket
* registration
* DB with tables for data & users
  + important: show a good DB design! NOT EVERYTHING IN 1 TABLE!!!!!

REPORT:

* **Security**: just a consideration about the secirity at the 3 tiers of our system.
  + - DOESN’T HAVE TO BE IMPLEMENTED IN OUR SYSTEM (although, some little little things might help =))
  + Write our analysis of what happens when some data is sent from client to DB (i.e. use encryption), access control, etc.
  + Cases of firewalls, network stuff
  + Include session management considerations
  + USER REGISTRATION: shall a user confirm its accout?
    - avoid unwanted registration with some other users’ data

Specifications

* Platform: PHP & MySQL
* Hosting: Linux server; not important: any capable hosting is ok (even localhost:3000?)
* Client: HTML5 (HTML,CSS,JavaScript)
  + must be responsive
  + (OPTIONAL) cross-browser compatibility

Quality Analysis

|  |  |  |
| --- | --- | --- |
| Quality attribute | Priority | Comment |
| **Durability**: *how long a software solution will function effectively and meet business requirements* | Low |  |
| **Portability**: *the ability of a* [*program*](https://en.wiktionary.org/wiki/program) *to* [*execute*](https://en.wiktionary.org/wiki/execute) *properly on multiple* [*hardware*](https://en.wiktionary.org/wiki/hardware) [*platforms*](https://en.wiktionary.org/wiki/platform)*.* | Medium |  |
| **Interoperability**:  *is a property of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, without any restricted access or implementation* | **High** |  |
| **Usability**: *is the ease of use and learnability of a software.* | **High** | Veeeeeery important |
| **Testability**:  *is the degree to which a software artifact supports testing in a given test context.* | Low |  |
| **Adaptability**:  *refers to a* [*process*](https://en.wiktionary.org/wiki/process)*, in which an* [*interactive system*](https://en.wikipedia.org/wiki/Interactivity) *adapts its behaviour to individual users based on* [*information*](https://en.wikipedia.org/wiki/Information) *acquired about its user(s) and its environment.* | Low/Medium |  |
| **Security**: *comprehends measures taken throughout the code's life-cycle to prevent gaps in the* [*security policy*](https://en.wikipedia.org/wiki/Security_policy) *of an* [*application*](https://en.wikipedia.org/wiki/Application_software) *or the underlying* [*system*](https://en.wikipedia.org/wiki/Operating_system) *(*[*vulnerabilities*](https://en.wikipedia.org/wiki/Vulnerability_(computer_science))*) through flaws in the* [*design*](https://en.wikipedia.org/wiki/Software_design)*,* [*development*](https://en.wikipedia.org/wiki/Software_engineering)*,* [*deployment*](https://en.wikipedia.org/wiki/Software_deployment)*,* [*upgrade*](https://en.wikipedia.org/wiki/Software_upgrading)*, or* [*maintenance*](https://en.wikipedia.org/wiki/Software_maintenance) *of the application.* | Medium |  |
| **Flexibility**: *it refers to designs that can adapt when external changes occur.* | **High** |  |
| **Modifiability**: *it refers to the ability of the user to either perform some changes on the system or request them.* | Medium |  |
| **Performance**: *is characterized by the amount of useful work accomplished by a computer system or computer network compared to the time and resources used.* | Medium |  |
| **Reliability**: *describes the ability to function at a specified moment or interval of time* | Medium |  |
| **Recoverability**: *ability to recover from a failure without loss of data.* | Low |  |
| **Resilience**: *the ability to provide and maintain an acceptable level of* [*service*](https://en.wikipedia.org/wiki/Service_(systems_architecture)) *in the face of* [*faults*](https://en.wikipedia.org/wiki/Fault_(technology)) *and challenges to* [*normal operation*](https://en.wikipedia.org/w/index.php?title=Normal_operation&action=edit&redlink=1)*.* | Low/Medium |  |
| **Maintainability**: *the probability of performing a successful repair action within a given time* | Medium |  |
| **Documentability**: *the quality of being* [*documentable*](https://en.wiktionary.org/wiki/documentable)*.* | **High** | Important |
| **Availability**: *the system being able to perform & accomplish user’s/system’s activities at any time, preventing faults and detecting/recovering from failures through failure management.* | **High** | Important |

Application Design

Web Application Stucture

* INDEX PAGE:
  + provides overview of the available data sets. Users can request some particular data sets, but they have to be registered in order to do it
* REGISTRATION:
  + ~~separated from INDEX PAGE~~ it can be a pop-up
  + you can register to the page (to request data sets) and there are input checks for consistency and correctness of data on the registration form
* DATA SETS PAGE (private):
  + Details of data can be viewed on a separate page, once the user is logged-in
  + Shopping basket: “buy” data sets you’re interested in. In other words, select data you want to export later in different formats, e.g. CSV

ARCHITECTURE

* Single-page application:
  + for the display of data sets to the registered users.
* Stateless thick server:
  + the client page sends data representing its current state to the server, usually through **Ajax** requests. Using this data, the server is able to reconstruct the client state of the part of the page which needs to be modified and can generate the necessary data or code (for instance, as JSON or JavaScript), which is returned to the client to bring it to a new state, usually modifying the page DOM tree according to the client action which motivated the request. This approach requires that more data be sent to the server and may require more computational resources per request to partially or fully reconstruct the client page state in the server. At the same time, this approach is more easily scalable because there is no per-client page data kept in the server and, therefore, Ajax requests can be dispatched to different server nodes with no need for session data sharing or server affinity.
* DB:
  + <https://www.phpmyadmin.net/>

Layout

INDEX PAGE

Components

* LOGIN/REGISTER button
* ~~Search bar~~

REGISTRATION / LOGIN FORM

* Registration form
  + Username
  + Email
  + Password
* Login Form
  + username
  + password

CONFIRMATION PAGE (OPTIONAL)

* another login

SHOPPING PAGE

* List of items
  + Select button (check/uncheck item)
  + Name (which is also a link)
  + Hidden Description (on “hover” it reveals)
  + Data format
  + Individual download button
  + Remove button
* items are stored on user’s DB table
* Check-out (batch download of everything in the shopping cart)
  + ~~PAYPAL (donation to us =) )~~

ALL DATA SETS PAGE (must be logged in)

* List of data sets, in the following tree structure:
  + First level categories (YEAR - or all time)
    - 2nd level categories (NY or BD or Both)
      * 3rd level categories (ORDER aka Ranking)
* Display of data sets (
  + CENTRAL PART of the page
  + Button to display small details
* DataSet container:
  + Name (which is also link)
  + Small description: expanded on request
  + Category
  + Add to shopping cart

SINGLE DATA SET PAGE (must be logged in)

* details
* chart
* add to cart
  + BEHAVIOUR: after adding to shopping cart, with CSS & JS we should change the