**Aston Events Website**

Entry page on host server: <http://www.razamahj.eas-cs2410-1617.aston.ac.uk/CRW/Home.html>

Organiser Username: abdul

Organiser Password: 123

In this report I will provide a brief description of the structure of my system and its depths, a hyperlink to the entry page of my system on the host server which is named Home.html. I will also provide the username and password of one organiser along with my database schema specifying my table structure, relations as well as constraints. I will also describe any assumptions I have made in terms of using the system.

1 Introduction

For this piece of coursework I was asked to demonstrate my ability to design and develop a three-tier dynamic dataset-driven website. The surrounding theme was to be an Aston events website for Aston students, used to promote the events organised within university. The events would be classified into 3 types being sport, culture and others. Each event is described by its name, description, time, picture, organiser, contact and venue.

2 Functional requirements

In order to gain a high-level mark in this project I was required to implement a set of functional requirements for both students and organisers. Students were able to:

* List the basic event information (name, description, time) for all events (4 marks)
* List the basic event information according to the three catalogues. (4 marks)
* List the basic event information based on the likeness ranking. (4 marks)
* List the basic event information according to dates. (4 marks)
* Click an event to view more details of the event including a picture, person to contact venue etc. and showing your interest to the event by clicking the Like button (8 marks)
* Register to become an event organiser (4 marks)

Event organisers were able to:

* Login / out the system. (6 marks)
* E2. Add new events to the system. (6 marks)
* E3. List all events organised by you (the current organiser). (4 marks)
* E3. Update an event organised by you (the current organiser). (6 marks)

3 Databases

In terms of designing and creating a database I needed to ensure that the database was able to store the event organiser and event information with proper relations and constraints. An event organiser was required to have a name and contact information (e.g. email, phone, etc.).

An event was required to include the event category, name, time, date, description, organiser, place, picture, likeness ranking and other information I deemed suitable. It was clear that in order to create such a database I needed two table’s one named organiser which stored the organiser info and the other named event which stored the event information. It was apparent that one organiser could organise multiple events but an event only has one organiser. For this reason I had set up an id column which stored the organiser’s id. This column was the primary key in the organisers table. This primary key was set as the foreign key in the event table referencing the organiser table.

The events table was to have its own primary key named id. Both primary keys for the organiser and events table were set to AUTO increment whist the foreign key in the events table which referenced the primary key from the organiser table was set to NOT NULL with a foreign key constraint however was not set to auto increment. This was not set to auto increment due to the fact that it was referencing another key and therefore there was no need to increment its value. Through creating such relationships and constraints I was able to successfully query both tables and retrieve information efficiently.

4 Structure of System

My system consists of several php/html combined webpages. The first webpage in my named home.html is in other words the index page. Through good interface design the user is able to effectively navigate through multiple pages when at the home page. The homepage consists of a navigation bar which enables the user to switch between all the pages that they have permission to access. Through the navigation bar the user can direct themselves to the login, register, all events, sport, culture, other, events by date and events by ranking page. Through the login page the user is able to log in if he or she is registered to the system, if not then they are required to create a username and password after which they will be able log into the system.

When a new user registers to the system they are required to enter all details in the right format, once they are registered they will be appointed an organiser id and they can then log into the system user the newly created username and password.

After a user logs in they are ten given organiser privileges, the logged in page then welcomes the user. On this page the navigation bar changes as event organisers have different functional requirements. After logging in the navigation bar allows the event organiser to view all the events organised by them as well as register and update events on the system. Through good coding practice I was able to make sure all users of the system are aware of what kind of information is being displayed and where exactly they are in the website. If a user of the system is not a registered event organiser they are not able to register, view and modify their events.

The source code file named regForm.php refers to the registration page where the user is able to register to the system. Through updateevent.php file the registered user can update their events, the file named yourevents.php is where registered users can view the events they have registered, the logout.php file allows the logged in user to log out, this redirects them to the login page.

The page named sports.php, culture.php, other.php are accessible to all students via the homepage and they list all events according to the category, if the user clicks on the sport tab in the navigation bar they will be redirected to a page which lists all events by the category sport, it will also do the same for the culture tab and others tab if the user clicks on them in the navigation bar.