AEM Project

Team Name: Data Demons (Database Developer in canvas)

Group Members: Justin Patrick Rodriguez, Ashley Rus, Sebastian Serrano, Dylan Werelius

Link to Github: https://github.com/nokk3n-dev/CPSC-332-Project/

The goal of this project was to create a database that would be able to handle the backend of an event planning website. The database would need to be able to store users and all the information relevant to the users such as phone number, email address, name, etc, as well as information relating to the events such as venue, list of speakers, list of presenter, name of the event, description, and more. The website would need to connect the users with the events so that they could sign up and register for them. The user would need to sign up on the website and then they would be free to register for any event that they wanted to attend, so long as the event was not full. If a user was an admin, they could sign up as one which would allow them special privileges like ability to create and manage events, as well as add new speakers, sponsors, and presenters. All of these elements combined would serve as the final product that we would deliver to the client so that they could use the service to reach customers.

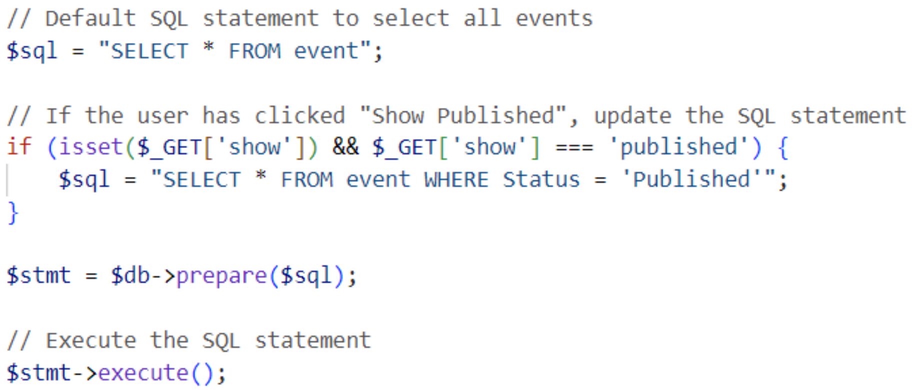
Database Design Process

When creating the database our group first created an ER diagram and later a Relational Diagram which you can find attached in the zip file. The ER diagram was created by Dylan Werelius and the Relational Diagram was created by Ashley Rus. Once there was a general understanding of the schema, Dylan transformed the diagrams into the sql code which would create the physical model as intended. Dylan also added some default users and a mock event titled LA Expo 2024 which would be hosted at USC on the football field.

Application Design

The basic model of the website would be a login page, a homepage, and then action pages which would be viewable from the homepage. Each of the pages had options which could be used to expand the user experience. For example, if the user did not have an account, then they would be able to navigate to the registration page from a link in the login page. If the user was a plain user, then they would be taken to the regular homepage, but if the user was signed up as an admin, then they would be taken to the admin home page.

From the regular homepage, a user can view events (active or inactive), sign up for events, and view the events that they are signed up for. The event viewer uses SQL queries that call the database and return all the events which are active or inactive depending on what the user wants. See Figure 01 below



```
// Default SQL statement to select all events
$sql = "SELECT * FROM event";

// If the user has clicked "Show Published", update the SQL statement
if (isset($_GET['show']) && $_GET['show'] === 'published') {
    $sql = "SELECT * FROM event WHERE Status = 'Published'";
}

$stmt = $db->prepare($sql);

// Execute the SQL statement
$stmt->execute();
```

Figure 01

From the admin homepage, an admin can view events, create new events, add universities/venues, manage events that they are the owner of, and add special guests (speakers, sponsors, and presenters). The admins are also able to sign up for events just like normal users.

The design is intended to be very simple with as few buttons as possible and a simple layout. The design uses basic elements of CSS in order to make the site look modern and robust.

Summary and Known Errors

Unfortunately this project is not complete, as not every requirement has been implemented, there are still some bugs that were not fixed. There are issues with the back buttons taking an admin user to the default user home page (simply an href that was never changed), and some buttons that take the user to old pages that were never updated.

All that in mind, the website and database are not very far off of completion. Most of the features that are missing do not need brand new code to be written, just old code to be updated and copied. The login page and registration page work perfectly, password checking works well, calls to the database are working well, and so much more.

Peer Reviews:

- In order to allow for anonymous peer reviews and feedback, I am asking the group members to write their reviews and send them via email. I apologize if this is inconvenient, but I thought it was important for the reviews to be truthful and insightful into each member's participation.