Assignment 5 Report

Andrew Murphy

This is a picture of my home page. It has six buttons displayed on it that all lead to different web pages. Each web page has a different functionality. The functionalities of the web pages are displayed in text on each of the six buttons.

Graphical user interface

Description automatically generated with low confidence

This is a picture of my insert student web page. It prompts the user to input information about a student to add to the database. There is a textbox for student id, student name, and major. Once the user types in all of the information they need to click the submit button and it sends the data that is inputted to the database. It then outputs a data from the query from before the new student was added. Below that it has the result of the query with the new student added. If the user clicks on the home page button it redirects them back to the main page.  
Graphical user interface, text, application

Description automatically generated

This is a picture of my insert job web page. It displays text prompting the user to input information about a job. There are textboxes for job id, company name, job title, salary, and desired major. Once the information is inputted the user clicks submit and it displays a query from before the new data was added and then below it shows the query with the new job added. If the user clicks home page it brings them back to the home page.Graphical user interface, text, application, email

Description automatically generated

This is a picture of my insert application web page. It has text guiding the user to insert information about an application. There are two text boxes. One for student id and one for job id. Once the user inputs the information, they should click submit to send the information to the database. Then a query from before the new application was added is displayed and then below it a query with the newly added application is displayed. If the user clicks the home page button, it redirects them to the home page.Graphical user interface, text, application

Description automatically generated

This is a picture of the view student web page. It has text prompting the user to select a major to view all students in that major. If the user selects an individual major it will query for all the students in that major and display it below. If the user selects all then it queries and finds all students by major and displays, it below. If the user clicks on the home page button it redirects them to the homepage. Text

Description automatically generated with medium confidence

This is a picture of the view jobs page. It has text prompting the users to select a major to view all jobs for that major. The user then uses the dropdown to select a major or all majors and the page does a query to find whatever was inputted. As you can see below it is searching for all jobs and displaying them below. If the user clicks on the home page button it redirects them to the home page.Graphical user interface, text, application

Description automatically generated

This is a picture of my view applications web page. It has text prompting the user to select an option to view all applications for a choice in the drop down menu. Once the user selects an option from the drop down menu and clicks submit it does a query for whatever the user selected. The image below shows results for a query of CSCE applications. If the user clicks the home page button it redirects the user to the home page.Graphical user interface, text, application, email

Description automatically generated

The url for my homepage is <http://www.csce.uark.edu/~ajm057/Assignment_5/homepage.html>. I kept it plain and simple with it referencing assignment 5 and its called homepage.html. I used turing as my host server and mysql also on turing to create my data base. In this project I used java, javascript, html, and php. I was able to combine them all to make a function webpage with six links that link to web pages based on different functionalities. My database design is here.

CREATE TABLE STUDENTS(

STUDENT\_ID int NOT NULL,

STUDENT\_NAME char(50) NOT NULL,

MAJOR char(50) NOT NULL,

PRIMARY key (STUDENT\_ID)

);

CREATE TABLE JOBS(

JOB\_ID int NOT NULL,

COMPANY\_NAME char(50) NOT NULL,

JOB\_TITLE char(50) NOT NULL,

SALARY int NOT NULL,

DESIRED\_MAJOR char(50) NOT NULL,

PRIMARY key (JOB\_ID)

);

CREATE TABLE APPLICATIONS(

STUDENT\_ID int NOT NULL,

JOB\_ID int NOT NULL,

PRIMARY key (STUDENT\_ID, JOB\_ID)

);

INSERT INTO STUDENTS VALUES (1234, 'Andrew Murphy', 'CSCE');

INSERT INTO STUDENTS VALUES (5678, 'John Smith', 'BIOL');

INSERT INTO JOBS VALUES (1111, 'JBHT', 'Software Engineer', 70000, 'CSCE');

INSERT INTO JOBS VALUES (2323, 'University of Arkansas', 'Teacher', 50000, 'MATH');

INSERT INTO JOBS VALUES (5656, 'Google', 'Software Developer', 90000, 'CSCE');

INSERT INTO APPLICATIONS VALUES (1234, 1111);

INSERT INTO APPLICATIONS VALUES (5678, 2323);

I create the three tables students, jobs, and applications. In students there are three attributes student\_id, student\_name, and major. The primary key is student\_id. In jobs there are 5 attributes job\_id, company\_name, job\_title, salary, desired\_major. The primary key is job\_id. In applications there are 2 attributes student\_id and job\_id. Both are primary keys. Some error checking that I did was using switch statements to determine what queries are executed. I also printed out the tables before the query to make sure they were able to be reached and then I printed out the updated tables with the new data. I used if statements to ensure that two inputs were given when selecting either job\_id or student\_id and then getting the actual id numbers. All in all I think this project was a success.