Database Management Systems
Authors: Bailey Grimes & Hayden Prater
Date: 4/20/2023

FINAL PROJECT

BAILEY'S URL: HTTPS://BAILEYG.APPS.DHAKA.CF-APP.COM/
HAYDEN'S URL: HTTPS://HBPRATER.APPS.DHAKA.CF-APP.COM/

INTRODUCTION

The objective of this project was to create a public website that utilizes a MySQL database in order to manipulate and display data. Specifically, our project is using Spring Boot as its backend, while the frontend is using Bootstrap and Thymeleaf frameworks to avoid having to use JavaScript. With this, we decided to develop our project off Turing to allow for a wider variety of available frameworks, server-hosts, and database management systems. Spring io was used to generate the initial project files in order to get started with the Spring Boot framework.

DESIGN METHODOLOGY

The first step in designing the project was to identify the requirements of the system. We wanted to create a web application that allows users to easily add/view students, jobs, and applications. With both of us being familiar in Java, we decided to use Spring Boot and Java as our backend framework. HTML, CSS, Bootstrap, and Thymeleaf were used to implement our frontend framework. To store our data we used MariaDB MySQL as our DBMS and Cloud Foundry as our server host.

First, we set up our development environment and created the database schema. We created a database called "service_instance_db" and tables called "students", "jobs", "job_majors", "applications", and "majors". We added the table “job_majors” to make a connection between jobID and majorID. We did this because it was in unnormalized form, having a many-to-many relationship between the major table and jobs table.

Next, we created the backend of the application using Spring Boot. We implemented this using maven that is our build tool, which complies and packages our source code/dependencies into an executable. The pom.xml describes all our dependencies/builds and maven reads this to see what dependencies need to be compiled and packaged in the executable that is deployed on the server. Spring Boot allows for “@Getmapping and @Postmapping”, in which you do a Getmapping when you want to access a particular page. When something is submitted on the page a Postmapping is executed, and a form is sent with the post. This form is what is used to access the variables to execute the correct queries.

Finally, we created the frontend of the application using HTML, CSS, Bootstrap, and Thymeleaf. We created a user interface that allows users to enter their ID, name, and current major to be added to the database. They are also allowed to create new jobs by entering in the company name, title, salary. All other inputs were allowed via a drop-down menu that was populated by the database. This ensured that the user would/could not input any invalid responses while also showing the user the valid choices.

In brief, the organization of the source code files were split up into certain package directories. Listed below is the path to certain files.

NOTE: Queries are contained in the repositories of each class, i.e. ApplicationRepository, JobRepository, MajorReposity, StudentRepository. The path to each of these will be in the backend work.

To View all Frontend Work:

DBfinalProject/src/main/resources/templates

To View all Backend Work:

DBfinalProject/src/main/java/edu.uark.csce.databasehb

To View DB table initialization:

DBfinalProject/src/main/java/edu.uark.csce.databasehb/data/Initialize

TESTING

During testing we used the Junit Jupiter framework, which is integrated with the Spring Boot framework, to allow us to use test driven development on certain user forms. During these unit tests we were able to determine the areas in the program where user input caused fatal errors. Each time a problem area was discovered we added another check in each of the isValid methods contained in each user form class, such as JobForm, StudentForm, etc.

In addition, we used Thymeleaf in our HTML to do checks on model variables like lists and single values to ensure the user enters valid input. One Thymeleaf example is the use of th:if which allows conditional checking in our HTML to avoid hassling with JavaScript. Changing input types from text to numeric also helped in instances where only number input was desired. With both HTML Thymeleaf and the backend checking for errors, messing up the program is significantly more difficult.

Using these two frameworks created ease of use during error checking which allowed us to focus on more resource intensive areas of the project. Also, with Bootstrap being integrated in our HTMLs we were able to create "Toast" messages which let the user know whether the action they were attempting was successful or not. For example, if the user tried to enter a new student and that student already exists in the database the toast message would notify them that it was a duplicate and will not act on the user's request. Granted, this was only possible by combining both the Bootstrap toast message and backend error checking in order to have functionality like this.

RESULTS AND PICTURES

The 7 functions implemented were successful and aligned with expectations. Any unexpected or known errors were handled with error checking using test driven development. The results displayed on each page are dynamic to user input and will alert the user if something went wrong. Below are pictures of each page as well as a picture for each result page.

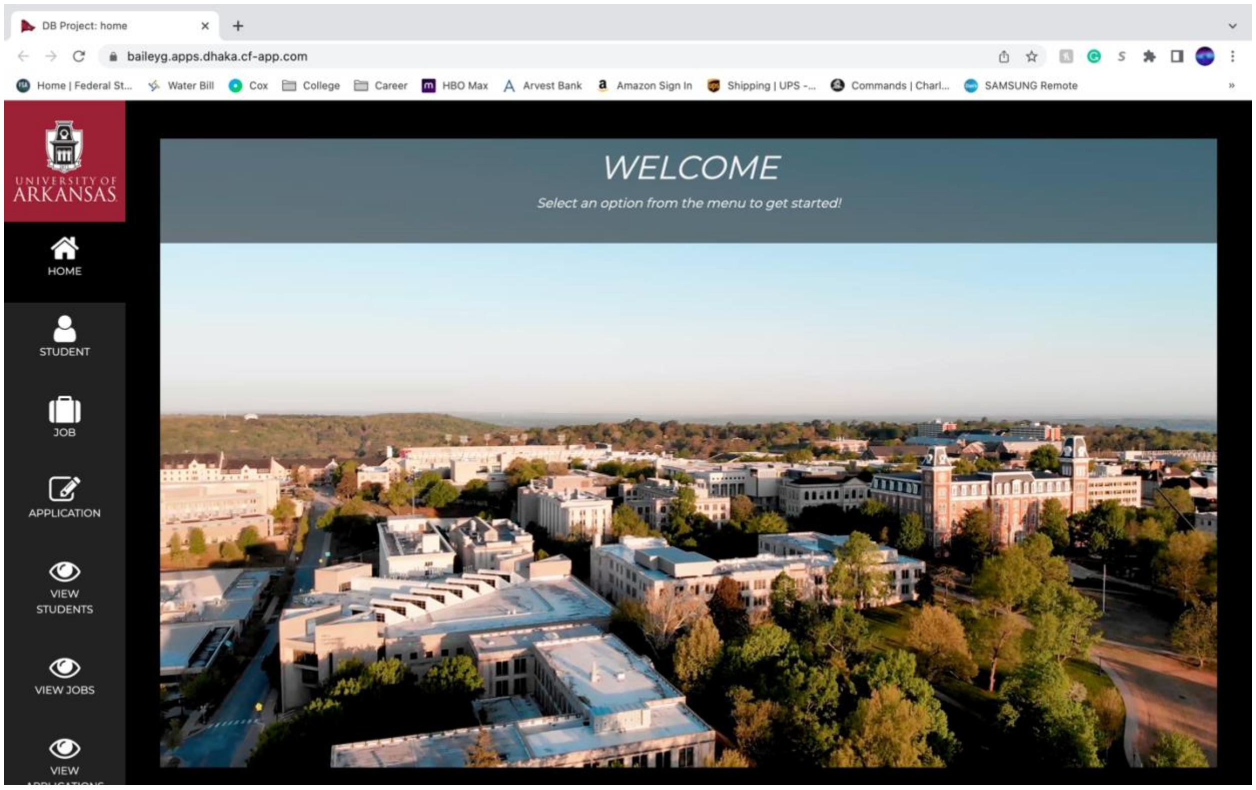


Figure 1: Home Page

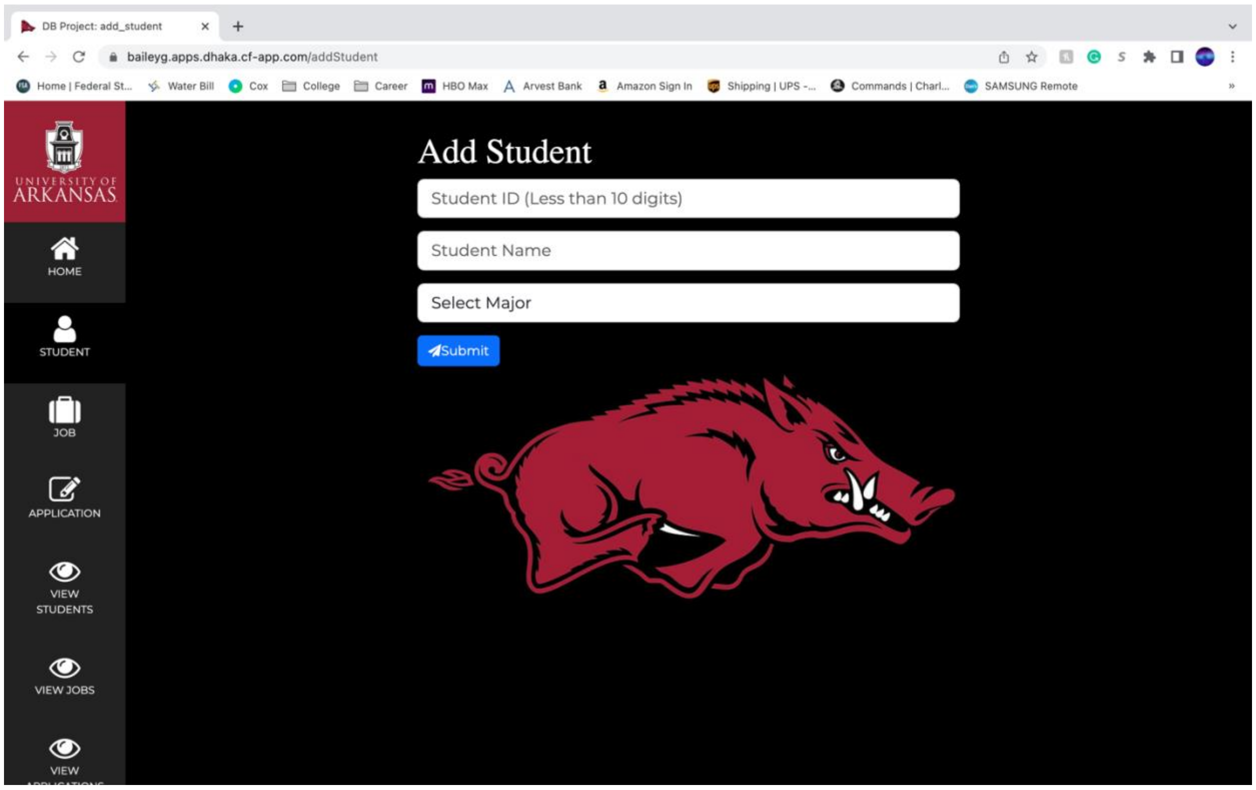
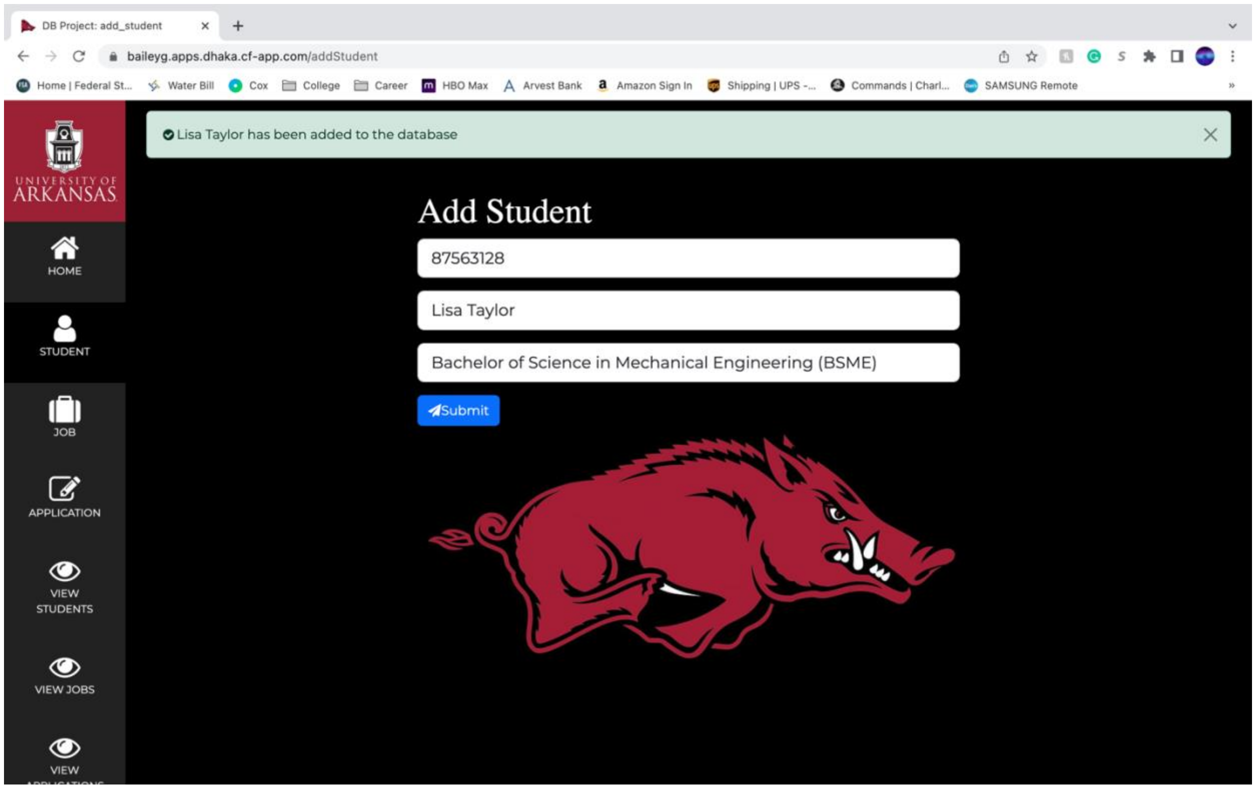


Figure 2: Add Student Page



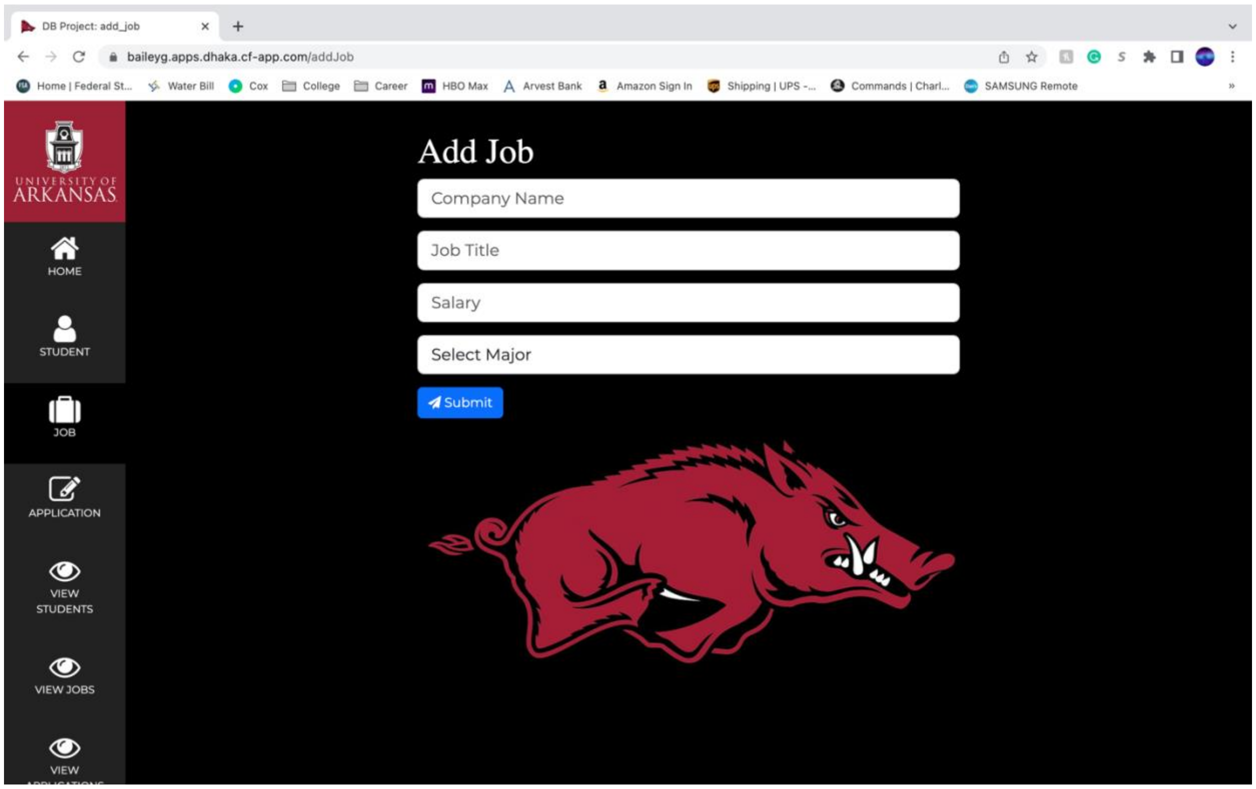
Lisa Taylor has been added to the database

87563128

Lisa Taylor

Bachelor of Science in Mechanical Engineering (BSME)

Figure 3: Add Student Result Page



Add Job

Company Name

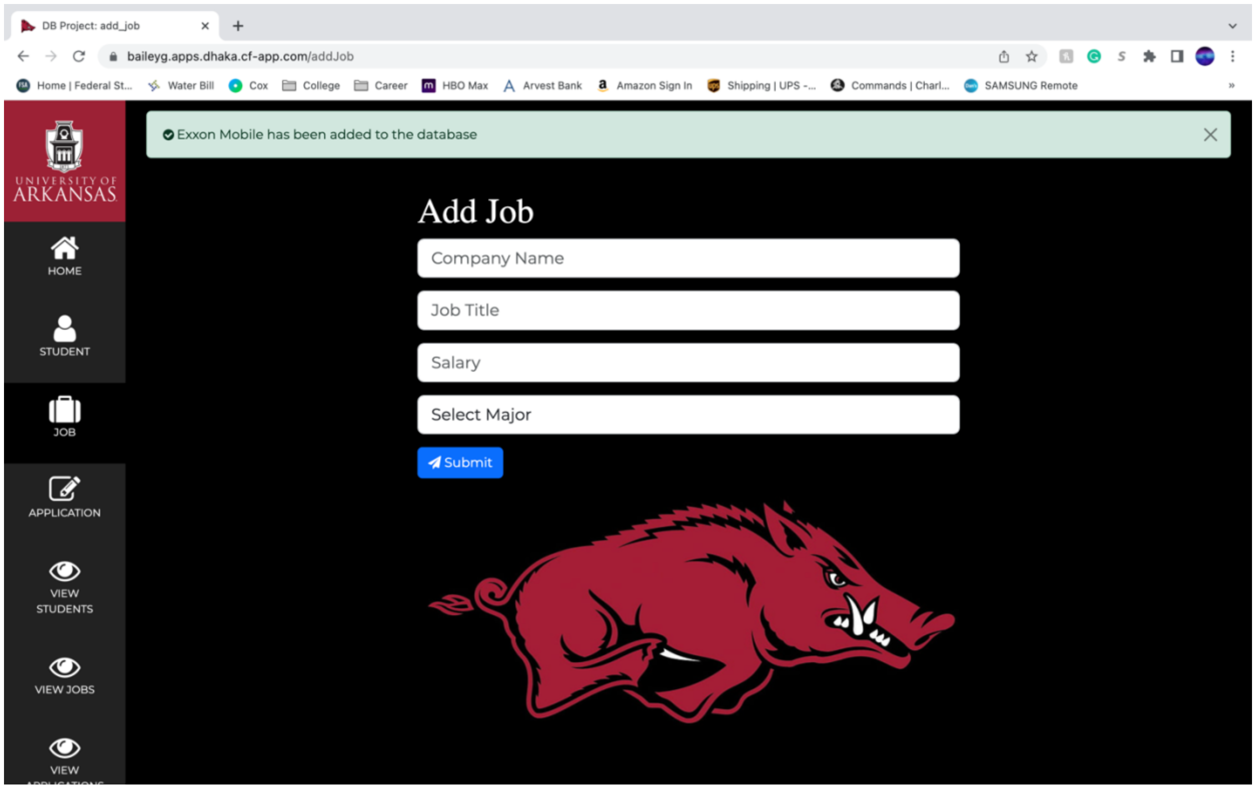
Job Title

Salary

Select Major

Submit

Figure 4: Add Job Page



Exxon Mobile has been added to the database

Add Job

Company Name

Job Title

Salary

Select Major

Submit

Figure 5: Add Job Result Page

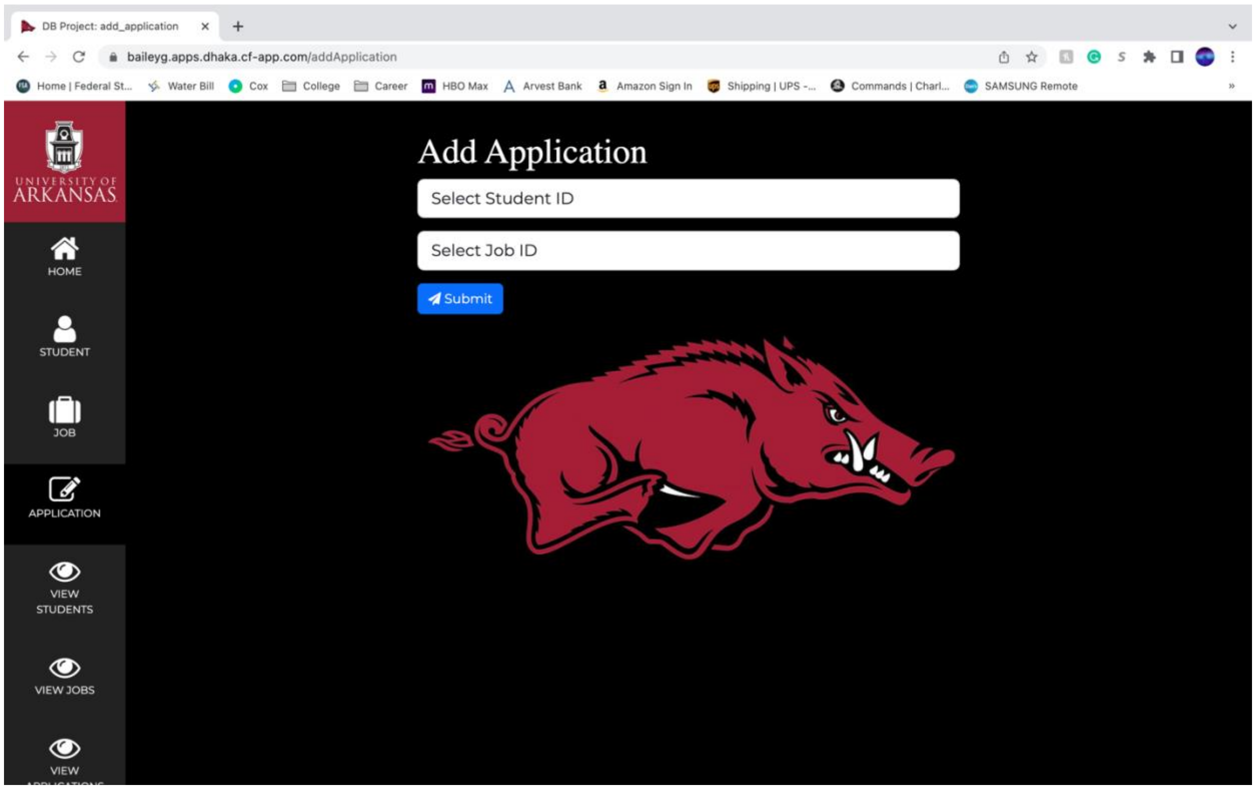
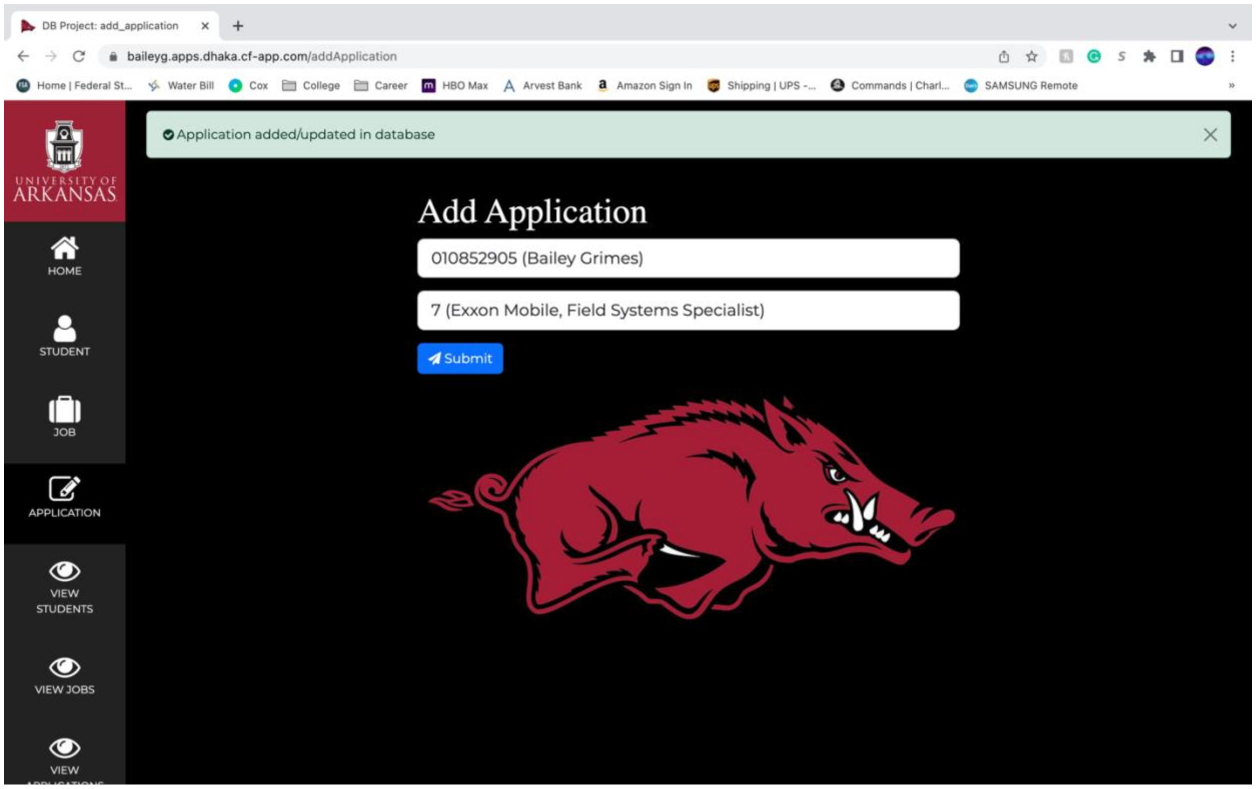


Figure 6: Add Application Page



Application added/updated in database

Add Application

010852905 (Bailey Grimes)

7 (Exxon Mobile, Field Systems Specialist)

Figure 7: Add Application Result Page

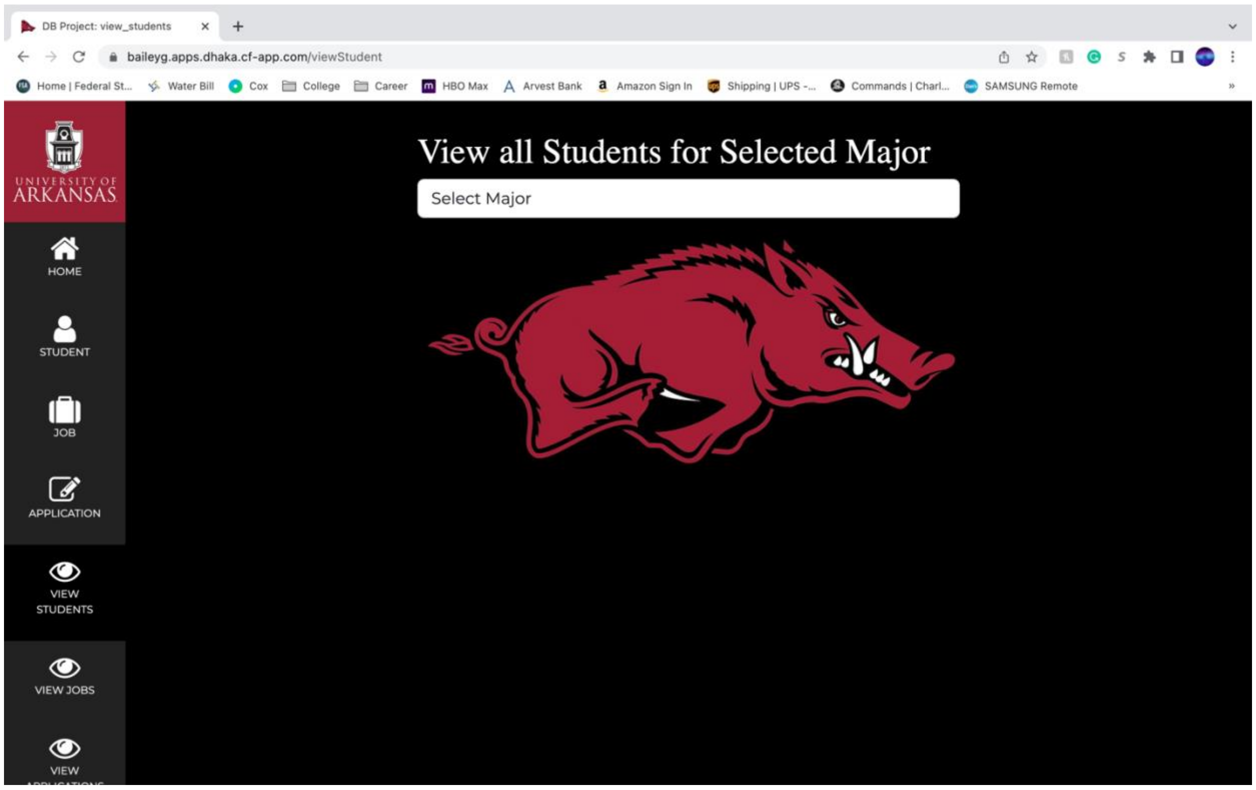
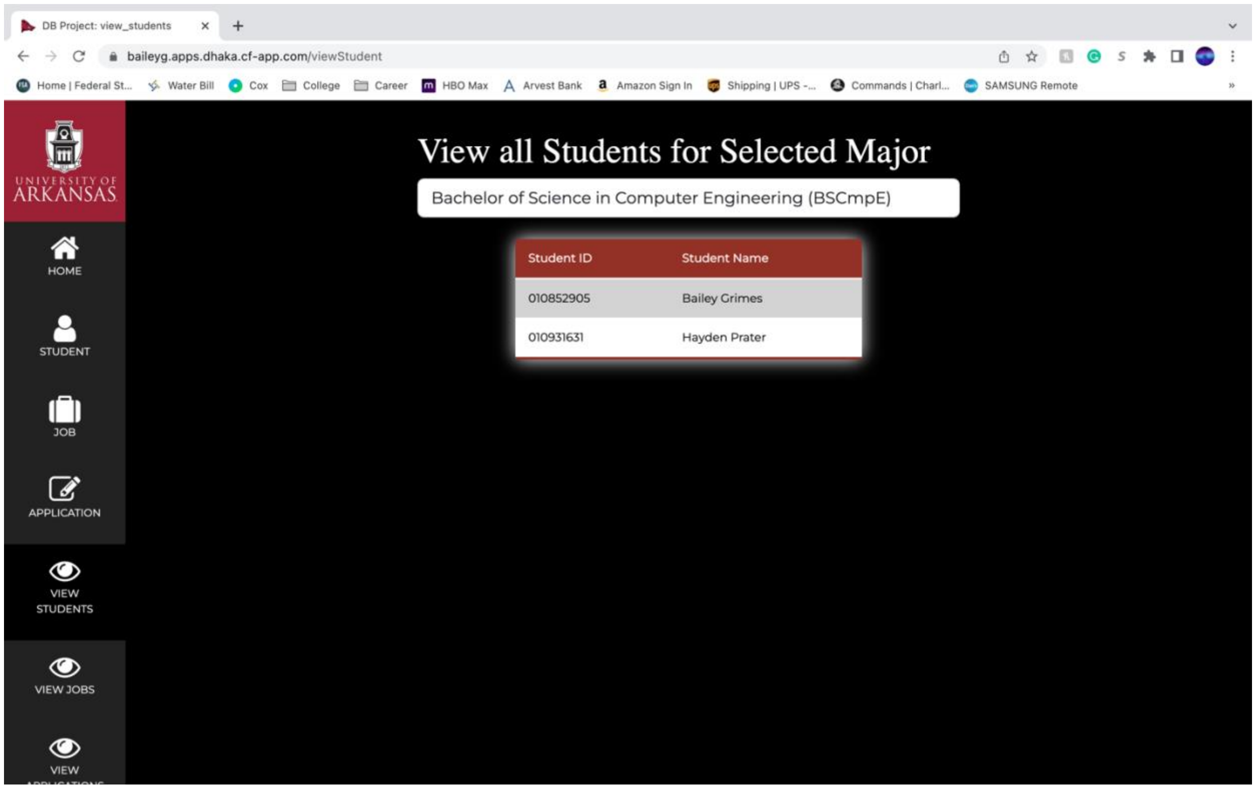
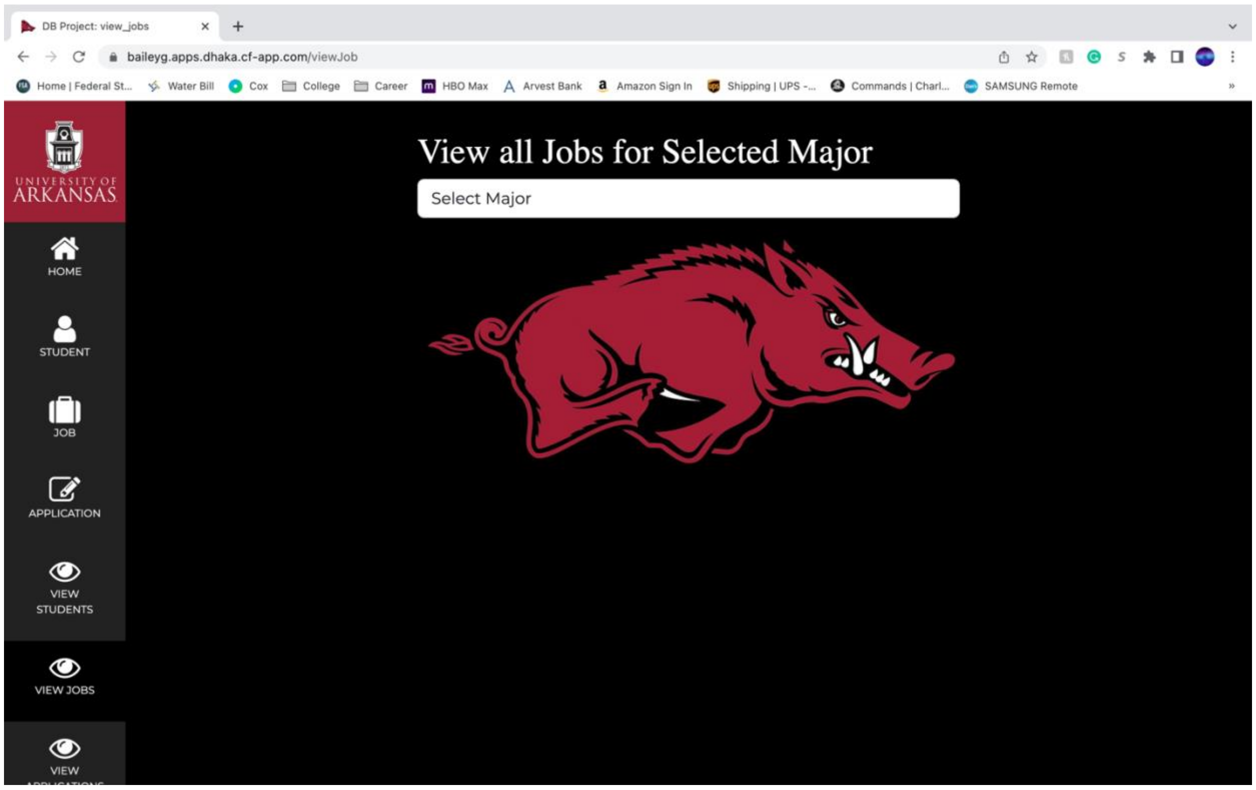

DB Project: view_students
baileyg.apps.dhaka.cf-app.com/viewStudent
UNIVERSITY OF ARKANSAS
HOME
STUDENT
JOB
APPLICATION
VIEW STUDENTS
VIEW JOBS
VIEW
View all Students for Selected Major
Select Major

Figure 8: View Students Page


DB Project: view_students
baileyg.apps.dhaka.cf-app.com/viewStudent
UNIVERSITY OF ARKANSAS
HOME
STUDENT
JOB
APPLICATION
VIEW STUDENTS
VIEW JOBS
VIEW
View all Students for Selected Major
Bachelor of Science in Computer Engineering (BSCmpE)

Student ID	Student Name
010852905	Bailey Grimes
010931631	Hayden Prater

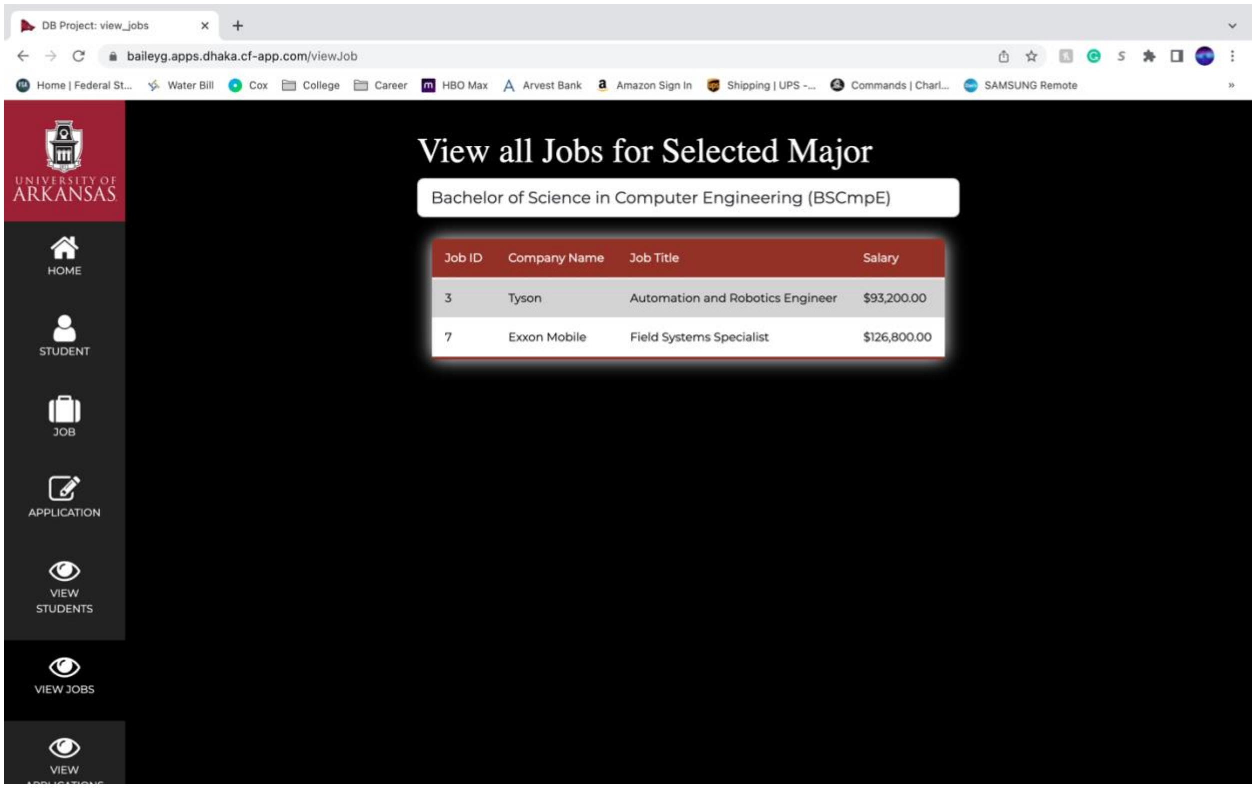
Figure 9: View Students Result Page



View all Jobs for Selected Major

Select Major

Figure 10: View Jobs Page



View all Jobs for Selected Major

Bachelor of Science in Computer Engineering (BSCmpE)

Job ID	Company Name	Job Title	Salary
3	Tyson	Automation and Robotics Engineer	$93,200.00
7	Exxon Mobile	Field Systems Specialist	$126,800.00

Figure 11: View Jobs Result Page

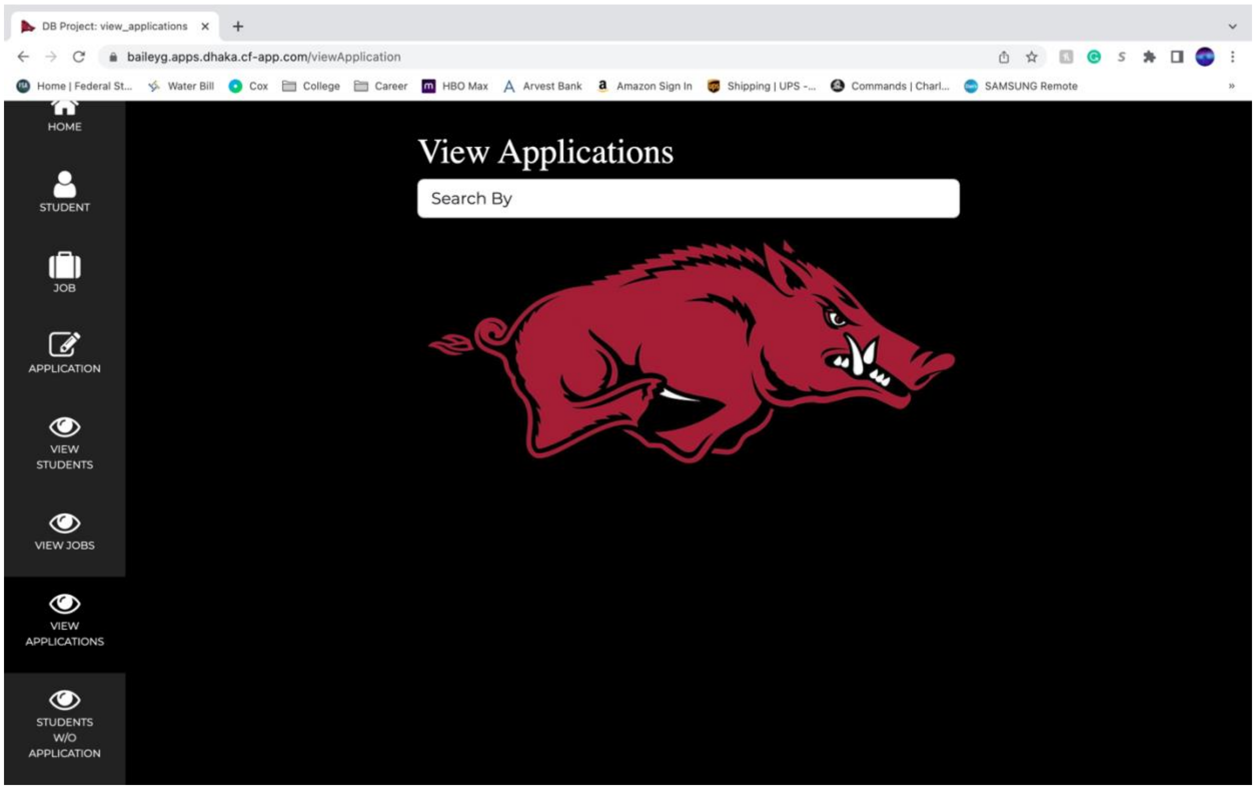
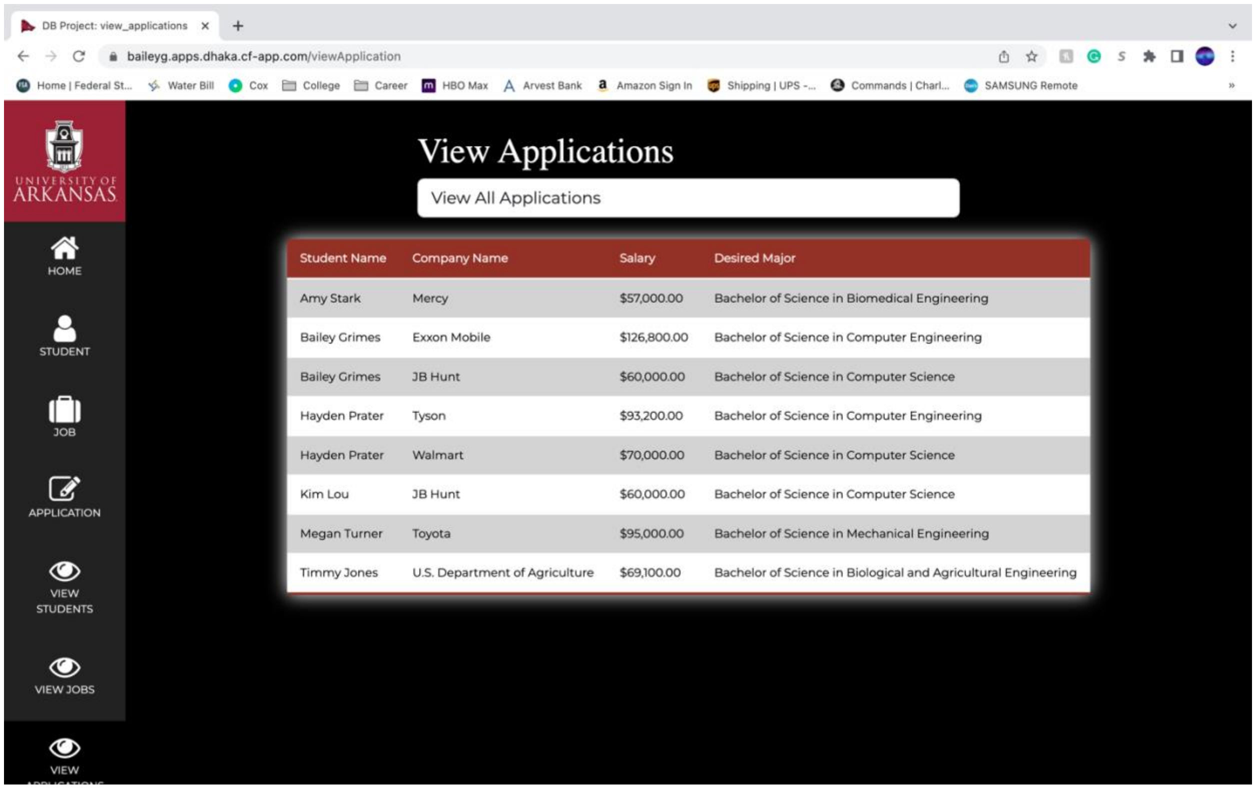


Figure 12: View Applications Page



Student Name	Company Name	Salary	Desired Major
Amy Stark	Mercy	$57,000.00	Bachelor of Science in Biomedical Engineering
Bailey Grimes	Exxon Mobile	$126,800.00	Bachelor of Science in Computer Engineering
Bailey Grimes	JB Hunt	$60,000.00	Bachelor of Science in Computer Science
Hayden Prater	Tyson	$93,200.00	Bachelor of Science in Computer Engineering
Hayden Prater	Walmart	$70,000.00	Bachelor of Science in Computer Science
Kim Lou	JB Hunt	$60,000.00	Bachelor of Science in Computer Science
Megan Turner	Toyota	$95,000.00	Bachelor of Science in Mechanical Engineering
Timmy Jones	U.S. Department of Agriculture	$69,100.00	Bachelor of Science in Biological and Agricultural Engineering

Figure 13: View Applications Result Page

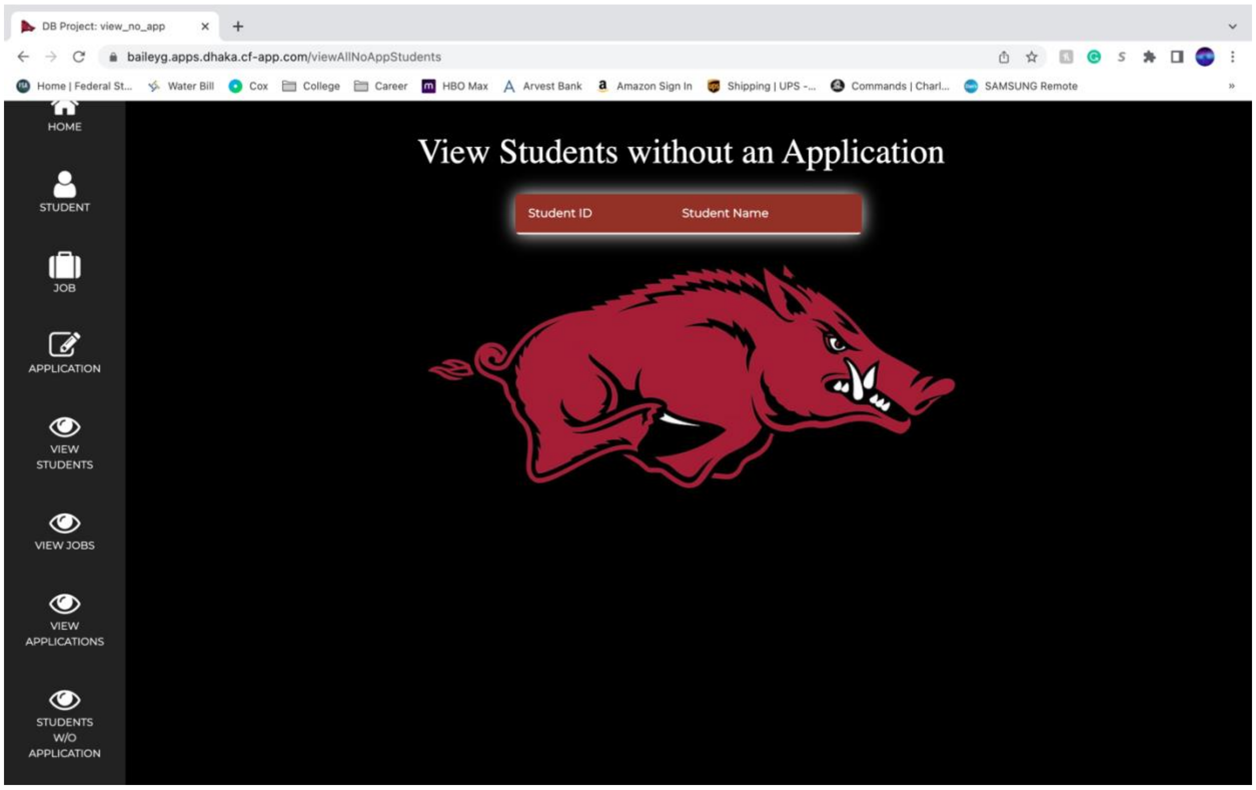
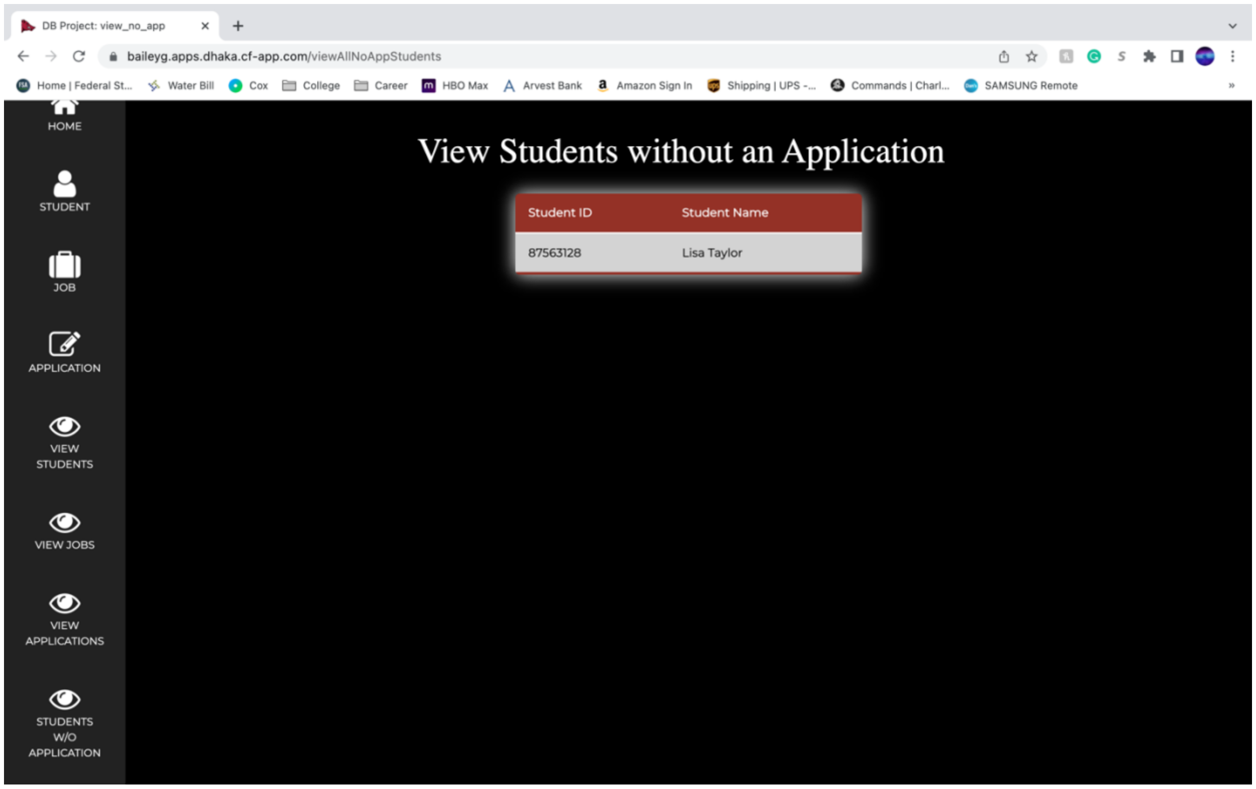


Figure 14: View No Application Students Page



Student ID	Student Name
87563128	Lisa Taylor

Figure 15: View No Application Students Result Page