Nicolas Lai

Abdullaah Robins

EK 125 C2

University of Statistics

By signing our names below, we acknowledge the work done in this project is original work.

Project Description

In this project, we aimed to help people learn more about schools in the United States in certain years so that they can have a better understanding about these schools when looking for a suitable school to apply to. The user can choose a year between 2009 and 2015 and enter an ID of the school of their choice. To find this ID the user just has to look through the data and match the ID with the school. They can then select the features they wish to see. When the click the button process then the data of the features they selected will show up inside the display box. Now when someone wants to know the key features about the school they were thinking about, they can see the features that play a big role in deciding if a school is suitable. The years that were provided in the database are also relevant years, so the data is reliable information. With this code, finding information about a school and comparing them with other schools is a lot easier than having to go online and searching each individual feature through each schools website.

Citation of the big data set: http://api.data.gov/ed/collegescorecard/

Project Solution

This project solves the issue of students not being sure which college is best fit for them. As college students, or students that have applied to college, we were not sure of any tools that would have all of the key information that would help us decide which college was best suited for us. While going through the process it would be a lot more helpful to have a tool that's sole purpose is to provide key features of different colleges and being able to easily compare between schools.

Folder name: “Abdullaah_Robins_Nicolas_Lai_Project_EK125_C2”

Filename: “FinalProjectx.mlapp”

In our code we have the reading and analyzing of our data in the file “FinalProjectx.mlapp” on lines 65-141

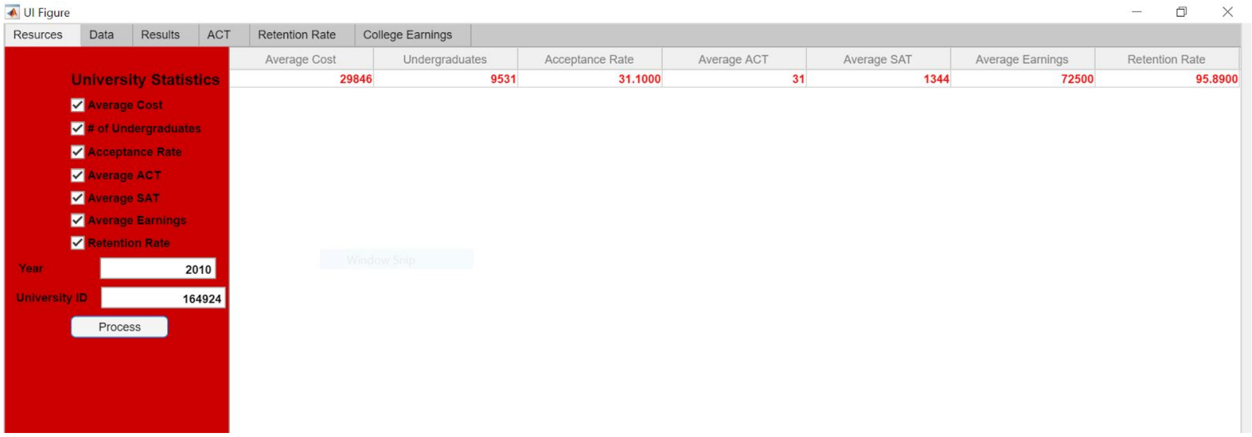
We have the writing to a file using fprintf in the file “FinalProjectx.mlapp” on lines 161-187

And plotting of the meaningful data in the file “FinalProjectx.mlapp” on lines 71-73/140-141/188-205

Instructions:

After running the app, the user should enter a university ID for a university of their choice. (Sample IDs can be found in the ‘Data’ tab) The user should then enter a year of which the data should represent and also click the boxes of which data they would like to see. The user should then click process.

Following the click a table should appear. The user should then go to the results tab and click ‘Display’. The program will save the data to a file and continue to write to it as many times as the user wants. The user can then look at the last three tabs for useful graphs comparing values to the national averages.

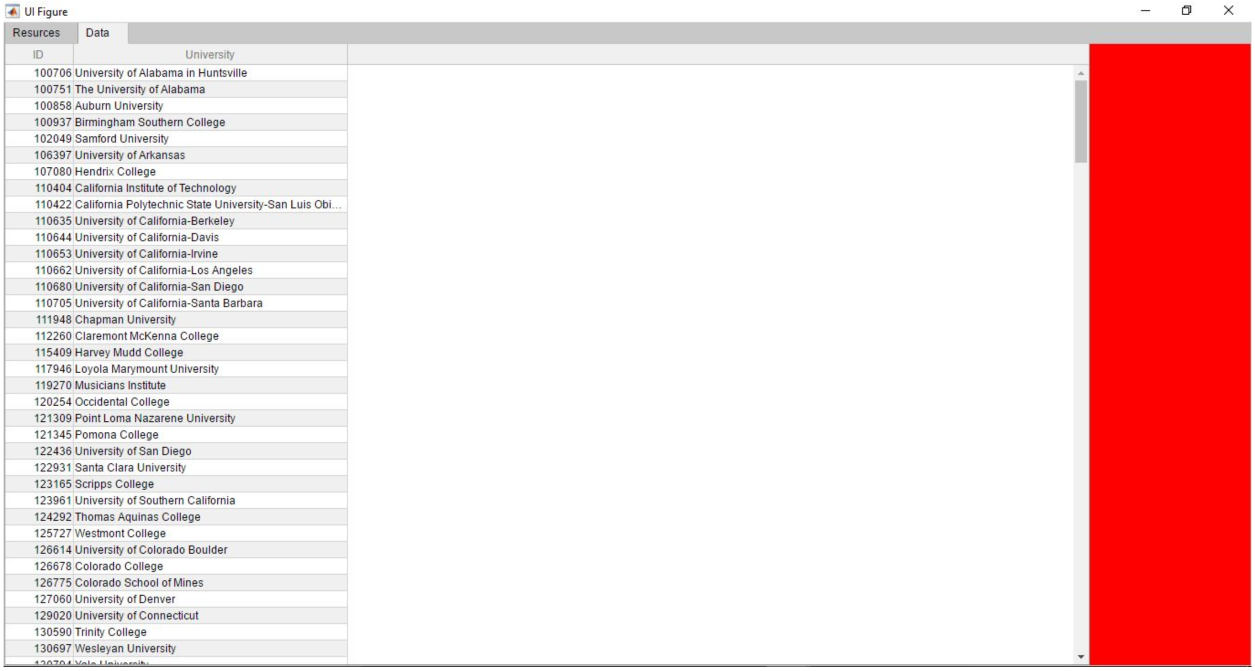


University Statistics

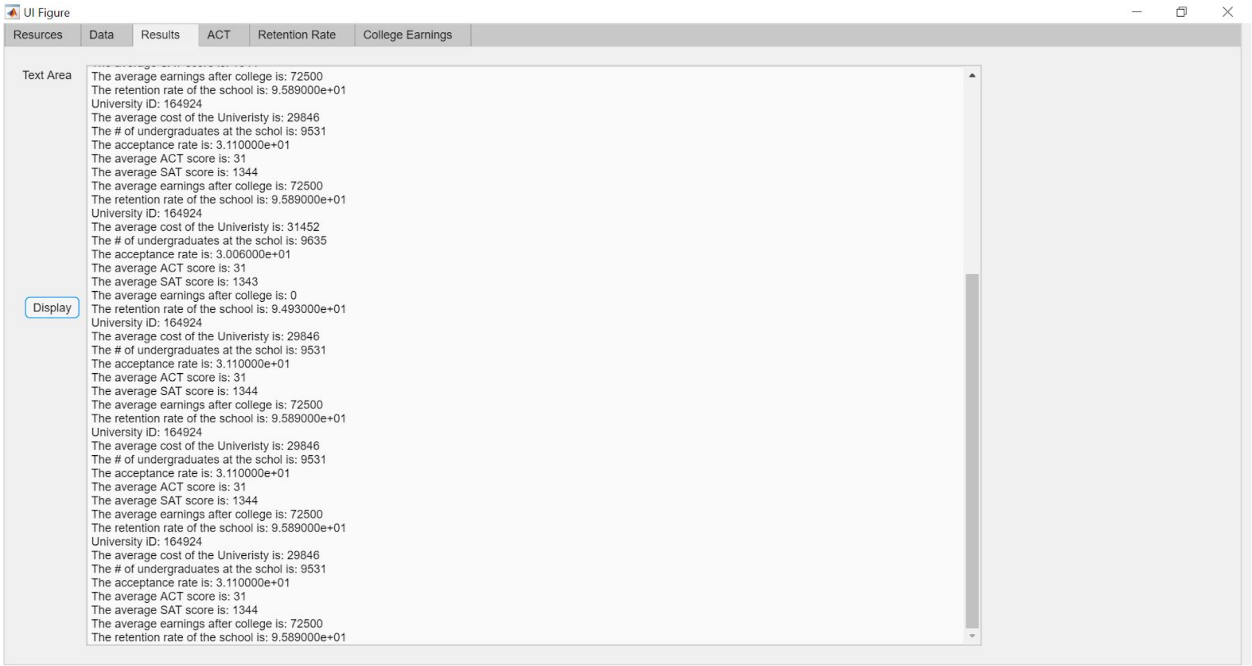
☑ Average Cost
☑ # of Undergraduates
☑ Acceptance Rate
☑ Average ACT
☑ Average SAT
☑ Average Earnings
☑ Retention Rate

Year: 2010
University ID: 164924

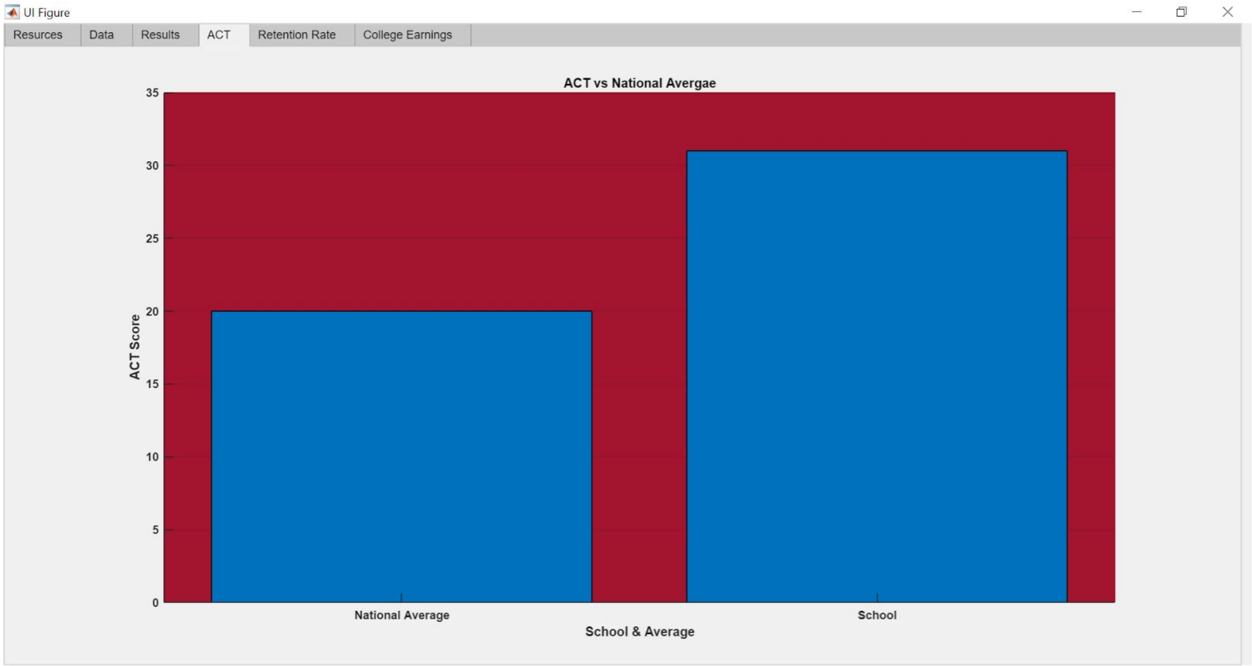
Average Cost	Undergraduates	Acceptance Rate	Average ACT	Average SAT	Average Earnings	Retention Rate
29846	9531	31.1000	31	1344	72500	95.8900

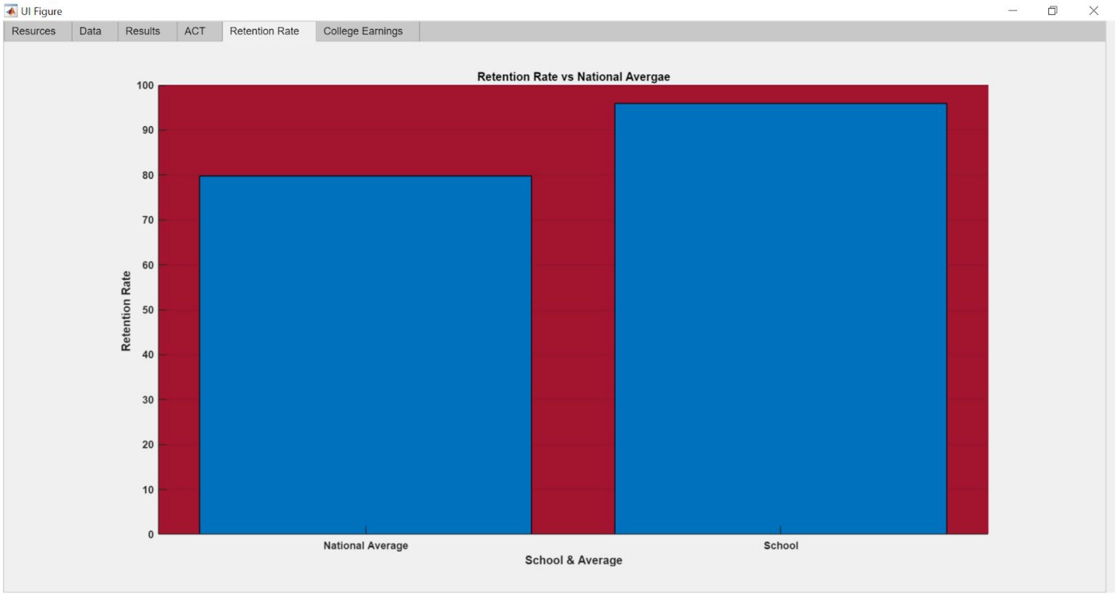


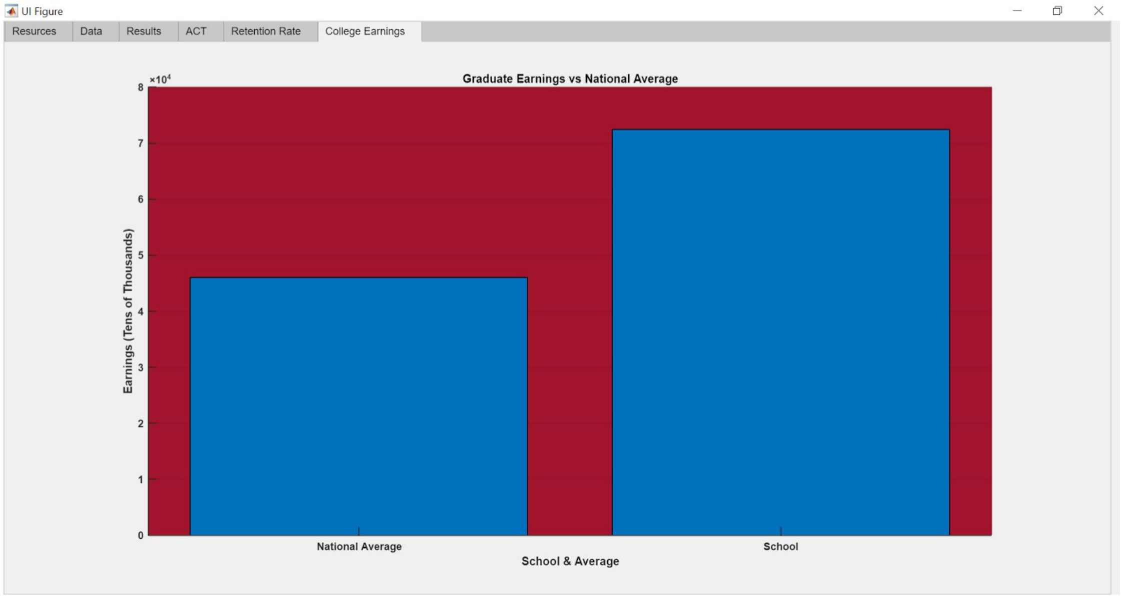
ID	University
100706	University of Alabama in Huntsville
100751	The University of Alabama
100858	Auburn University
100937	Birmingham Southern College
102049	Samford University
106397	University of Arkansas
107080	Hendrix College
110404	California Institute of Technology
110422	California Polytechnic State University-San Luis Obi…
110635	University of California-Berkeley
110644	University of California-Davis
110653	University of California-Irvine
110662	University of California-Los Angeles
110680	University of California-San Diego
110705	University of California-Santa Barbara
111948	Chapman University
112260	Claremont McKenna College
115409	Harvey Mudd College
117946	Loyola Marymount University
119270	Musicians Institute
120254	Occidental College
121309	Point Loma Nazarene University
121345	Pomona College
122436	University of San Diego
122931	Santa Clara University
123165	Scripps College
123961	University of Southern California
124292	Thomas Aquinas College
125727	Westmont College
126614	University of Colorado Boulder
126678	Colorado College
126775	Colorado School of Mines
127060	University of Denver
129020	University of Connecticut
130590	Trinity College
130697	Wesleyan University
130794	Yale University

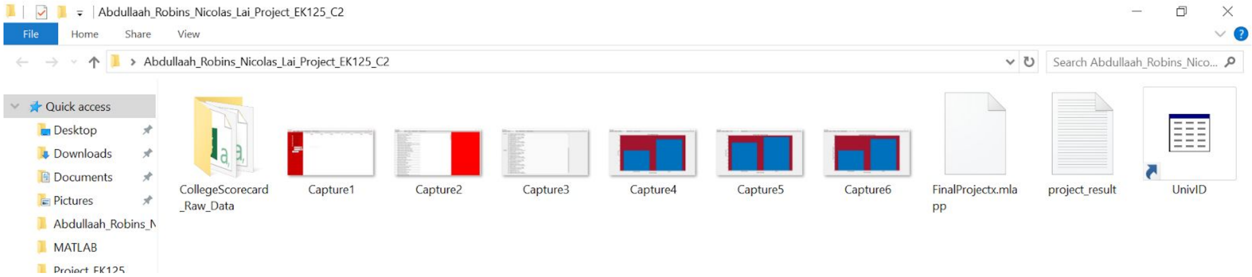


```
The average earnings after college is: 72500
The retention rate of the school is: 9.589000e+01
University ID: 164924
The average cost of the Univeristy is: 29846
The # of undergraduates at the schol is: 9531
The acceptance rate is: 3.110000e+01
The average ACT score is: 31
The average SAT score is: 1344
The average earnings after college is: 72500
The retention rate of the school is: 9.589000e+01
University ID: 164924
The average cost of the Univeristy is: 31452
The # of undergraduates at the schol is: 9635
The acceptance rate is: 3.006000e+01
The average ACT score is: 31
The average SAT score is: 1343
The average earnings after college is: 0
The retention rate of the school is: 9.493000e+01
University ID: 164924
The average cost of the Univeristy is: 29846
The # of undergraduates at the schol is: 9531
The acceptance rate is: 3.110000e+01
The average ACT score is: 31
The average SAT score is: 1344
The average earnings after college is: 72500
The retention rate of the school is: 9.589000e+01
University ID: 164924
The average cost of the Univeristy is: 29846
The # of undergraduates at the schol is: 9531
The acceptance rate is: 3.110000e+01
The average ACT score is: 31
The average SAT score is: 1344
The average earnings after college is: 72500
The retention rate of the school is: 9.589000e+01
University ID: 164924
The average cost of the Univeristy is: 29846
The # of undergraduates at the schol is: 9531
The acceptance rate is: 3.110000e+01
The average ACT score is: 31
The average SAT score is: 1344
The average earnings after college is: 72500
The retention rate of the school is: 9.589000e+01
```




UI Figure
Resources
Data
Results
ACT
Retention Rate
College Earnings
Retention Rate vs National Avergae
100
90
80
70
60
50
40
30
20
10
0
Retention Rate
National Average
School
School & Average


UI Figure
Resources
Data
Results
ACT
Retention Rate
College Earnings
Graduate Earnings vs National Average
×10^4
8
7
6
5
4
3
2
1
0
Earnings (Tens of Thousands)
National Average
School
School & Average


Abdullaah_Robins_Nicolas_Lai_Project_EK125_C2
File
Home
Share
View
Abdullaah_Robins_Nicolas_Lai_Project_EK125_C2
Search Abdullaah_Robins_Nico...
Quick access
Desktop
Downloads
Documents
Pictures
Abdullaah_Robins_N
MATLAB
Project_EK125
CollegeScorecard_Raw_Data
Capture1
Capture2
Capture3
Capture4
Capture5
Capture6
FinalProjectx.mlapp
project_result
UnivID

CODE

```
properties (Access = private)
Data % Description
Second_Data
Selected_Prop
ID
Year
Resp
URL % Description
TableData % Description
cost % resulting cost
act %act score from school
sat %sat score from school
size %school undergraduate size
acc_rate %acceptance rate
earning %average earnings
ret_rate %retention rates
Text % Data in text file
National_act % national average
National_earning %national earnings average
National_ret %national retention rate
End

function startupFcn(app)
% Load the data.Takes data from school data base to give user a list of schools and their IDs.
load UnivID.mat
name1 = xTry.INSTNM;
Name = cellstr(name1);
UID = xTry.UNITID;
% Store the data in a table and display it in one of the App's tabs.
app.Data = table(UID,Name);
app.UITable.Data = table2cell(app.Data);
app.National_act = 20; % national average
app.National_earning = 46000; %national earnings average
app.National_ret = 79.8; %national retention rate
end

function ProcessButtonPushed(app, event)
%Takes the ID from user input and the Year from user input and extracts the data from the API
key = 'zh3FrSr8h2G3o30rRTFfK9IlY8Rixy7g5jV3eDue';
app.ID = app.univ_id.Value;
app.Year = app.Years.Value;
app.URL =
sprintf('https://api.data.gov/ed/collegescorecard/v1/schools?id=%d&api_key=%s',app.ID,key);
```

```
app.Resp = webread(app.URL);
app.Second_Data = eval(sprintf('app.Resp.results.x%d',app.Year));
%Checks to see if Earning is checked & error checks to make sure the data isn't empty
if (app.EarningsBox.Value == 1) &&
(any(app.Second_Data.earnings.x7_yrs_after_entry.mean_earnings) == 1)
app.earning = app.Second_Data.earnings.x7_yrs_after_entry.mean_earnings;
else
app.earning = [0];
end
%Checks to see if Cost is checked & error checks to make sure the data isn't empty
if (app.CostBox.Value == 1) && (any(app.Second_Data.cost.avg_net_price.overall) == 1)
app.cost = app.Second_Data.cost.avg_net_price.overall;
else
app.cost = {'NULL'};
end
%Checks to see if Retention is checked & error checks to make sure the data isn't empty
if (app.RetentionRateCheckBox.Value == 1) &&
(any(app.Second_Data.student.retention_rate.overall.full_time) == 1)
app.ret_rate = app.Second_Data.student.retention_rate.overall.full_time;
app.ret_rate = app.ret_rate*100;
else
app.ret_rate = [0];
end
%Checks to see if Acceptance is checked & error checks to make sure the data isn't empty
if (app.RateBox.Value == 1) && (any(app.Second_Data.admissions.admission_rate.overall) ==
1)
app.acc_rate = app.Second_Data.admissions.admission_rate.overall;
app.acc_rate = app.acc_rate * 100;
else
app.acc_rate = {'NULL'};
end
%Checks to see if Size is checked & error checks to make sure the data isn't empty
if (app.UndergradBox.Value == 1) && (any(app.Second_Data.student.size) == 1)
app.size = app.Second_Data.student.size;
else
app.size = {'NULL'};
end
%Checks to see if SAT is checked & error checks to make sure the data isn't empty
if (app.SATBox.Value == 1) && (any(app.Second_Data.admissions.sat_scores.average.overall)
== 1)
app.sat = app.Second_Data.admissions.sat_scores.average.overall;
else
app.sat = [0];
end
%Checks to see if ACT is checked & error checks to make sure the data isn't empty
```

```
if (app.ACTBox.Value == 1) &&
(any(app.Second_Data.admissions.act_scores.midpoint.cumulative) == 1)
app.act = app.Second_Data.admissions.act_scores.midpoint.cumulative;
else
app.act = [0];
end
%Takes the data and turns it into a table and the stores the data in the UItable
app.TableData = table(app.cost,app.size,app.acc_rate,app.act,app.sat,app.earning,app.ret_rate);
app.UITable2.Data = table2cell(app.TableData);
End

function DisplayButtonPushed(app, event)
%Creates a file that will contain the information processed on the school.
fid = fopen('project_result.txt', 'at');
fprintf(fid,"University iD: %d\n",app.ID);
 %Writes to the file the information as long as the information is processed through the api.
if isnumeric(app.cost) == 1
fprintf(fid,"The average cost of the Univeristy is: %d\n",app.cost);
end
if isnumeric(app.size) == 1
fprintf(fid,"The # of undergraduates at the schol is: %d\n", app.size);
end
if isnumeric(app.acc_rate) == 1
fprintf(fid,"The acceptance rate is: %.2f\n", app.acc_rate);
end
if isnumeric(app.act) == 1
fprintf(fid,"The average ACT score is: %d\n", app.act);
end
if isnumeric(app.sat) == 1
fprintf(fid,"The average SAT score is: %d\n", app.sat);
end
if isnumeric(app.earning) == 1
fprintf(fid,"The average earnings after college is: %d\n", app.earning);
end
if isnumeric(app.ret_rate) == 1
fprintf(fid,"The retention rate of the school is: %.2f\n", app.ret_rate);
end
fclose(fid);
app.Text = fileread('project_result.txt');
app.TextArea.Value = app.Text;
fclose('all');
%creates bar graph for ACT
actcat = categorical({'School','National Average'});
scores = [app.act,app.National_act];
if app.act > 0
```

```
bar(app.ACT_Bar,actcat,scores);
end
%Creates bar graph for retention rate
retcat = categorical({'School','National Average'});
scores1 = [app.ret_rate,app.National_ret];
if app.ret_rate > 0
bar(app.Ret_Bar,retcat,scores1);
end
%Creates bar graph for earnings
earncat = categorical({'School','National Average'});
scores2 = [app.earning,app.National_earning];
if app.earning > 0
bar(app.Earn_Bar,earncat,scores2);
end
```