AVA An Automated Voice Activated Advisement System

(Code and Test images)

By: The Software Gurus

Team Members:

Omar Obidat

Jesse Parron

Allen Asencio

Dan Ferdetta

Sumit Mistry

Software Engineering 515

Fall 2022

Professor Johnson

December 19, 2022

Table of Contents

1. Table of Contents

[2.](#_heading=h.lnxbz9) Introduction 2

[3.](#_heading=h.35nkun2) Outputs 3

[3.1.](#_heading=h.1ksv4uv) The sign-in module that will be connected to the schools database:(below): 3

[3.2.](#_heading=h.44sinio) The test profile used to login (below): 4

[3.3.](#_heading=h.2jxsxqh) The home screen that connects with each module (below): 4

[3.4.](#_heading=h.z337ya) The current GPA module (below): 5

[3.5.](#_heading=h.z337ya) The what-if GPA module (below): 6

[3.6.](#_heading=h.3j2qqm3) The module for the students transcript (below): 7

[3.7.](#_heading=h.z337ya) The course catalog module (below): 8

[3.8.](#_heading=h.z337ya) The academic advisement module (below): 10

[3.9.](#_heading=h.2bn6wsx) The table that has the student profile that will be connected to the login page(below): 11

[3.10.](#_heading=h.qsh70q) The table that will connect each student to the classes that they have taken(below) 12

[3.11.](#_heading=h.1pxezwc) This table will show the grades of each student for GPA calculation (below): 13

[3.12.](#_heading=h.49x2ik5) This table shows all available courses and the professors teaching them (below): 13

[4.](#_heading=h.2p2csry) Main Code 14

[5.](#_heading=h.147n2zr) Catalog code 23

[6.](#_heading=h.3o7alnk) Current GPA code 26

7[.](#_heading=h.3o7alnk) What-If GPA code 28

8[.](#_heading=h.23ckvvd) Home interface code 29

9[.](#_heading=h.ihv636) Login code 33

10[.](#_heading=h.32hioqz) Transcript code 37

11[.](#_heading=h.3o7alnk) Advising code 39

# Introduction

This document shows some images of the modules being implemented and tested individually. So far, implementation is going well, and we are getting some bugs with the language recognition, but that will be sorted out by the next deliverable. We noticed when testing that sometimes based on internet strength the language recognition does not process speech in a quick manner. Due to this, when testing and screenshotting results the team could not include the catalog output.

Advising process has been implemented and functioning as per the specification provided. The team is also working on making the front end more pleasing for the overall user experience. Other functions that will be implemented by the next deliverable is the admin portion of the system.

# Outputs

# Once opened, the mandatory sign in page will allow users to sign in using their active Montclair State University MSU-CWID and password to enter the system. This sign-in module will be connected to the school’s database with information of students recorded within.

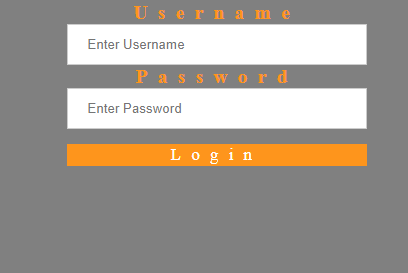


Figure 1: This is what the Log-In screen looks like

## The test profile used to login (below):

Graphical user interface, application

Description automatically generated

## Once successfully authenticated, the user is redirected to the below Dashboard. The home screen that connects with each module. Dashboard screen which user can vocally access any module.

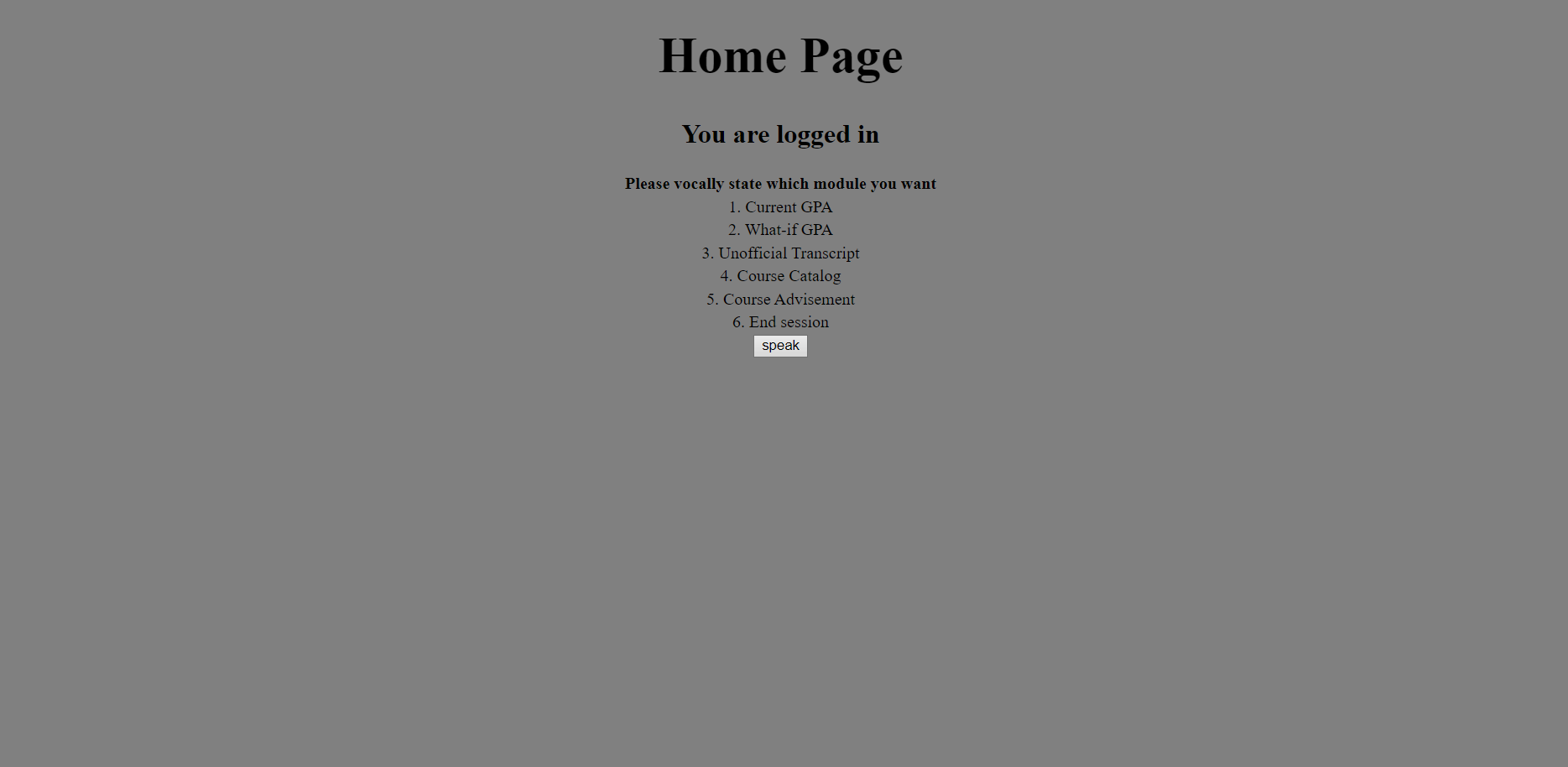


Figure 2: This is what the home screen looks like after a successful login

## Upon clicking the “speak” button, the user may say something along the lines of “Current GPA”. The user can access the GPA module allowing the student to view their current GPA. The user can then click the “speak” button, say “home”, then they will be redirected to the home screen.

Text

Description automatically generated with medium confidence

Figure 3 : This is the “Current GPA” screen

## From the home screen, upon clicking the “speak” button, the user may say something along the lines of “What-If GPA”. The user can access the What-If GPA module allowing the student to view their current GPA. The user can then click the “speak” button, say a whole number to indicate how many classes they would like to add to their “what-if gpa” calculation, followed by the grade points of each class, then the user will be redirected to the screen listed in Figure 4. The results are listed in Figure 5. From either screen, the user can then click the “speak” button, say “home”, then they will be redirected to the home screen.

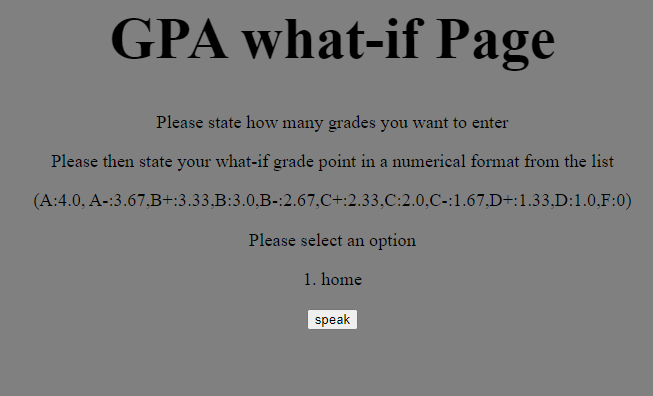


Figure 4 : This is the “What-If GPA” screen

A picture containing chart

Description automatically generated

Figure 5 : This is the “Calculated What-If GPA” screen

## From the home page the user can click the “speak” button and the user can say “Unofficial Transcript” to access their current transcript. After the Transcript is displayed, the user can then click the “speak” button, say “home”, then they will be redirected to the home screen.

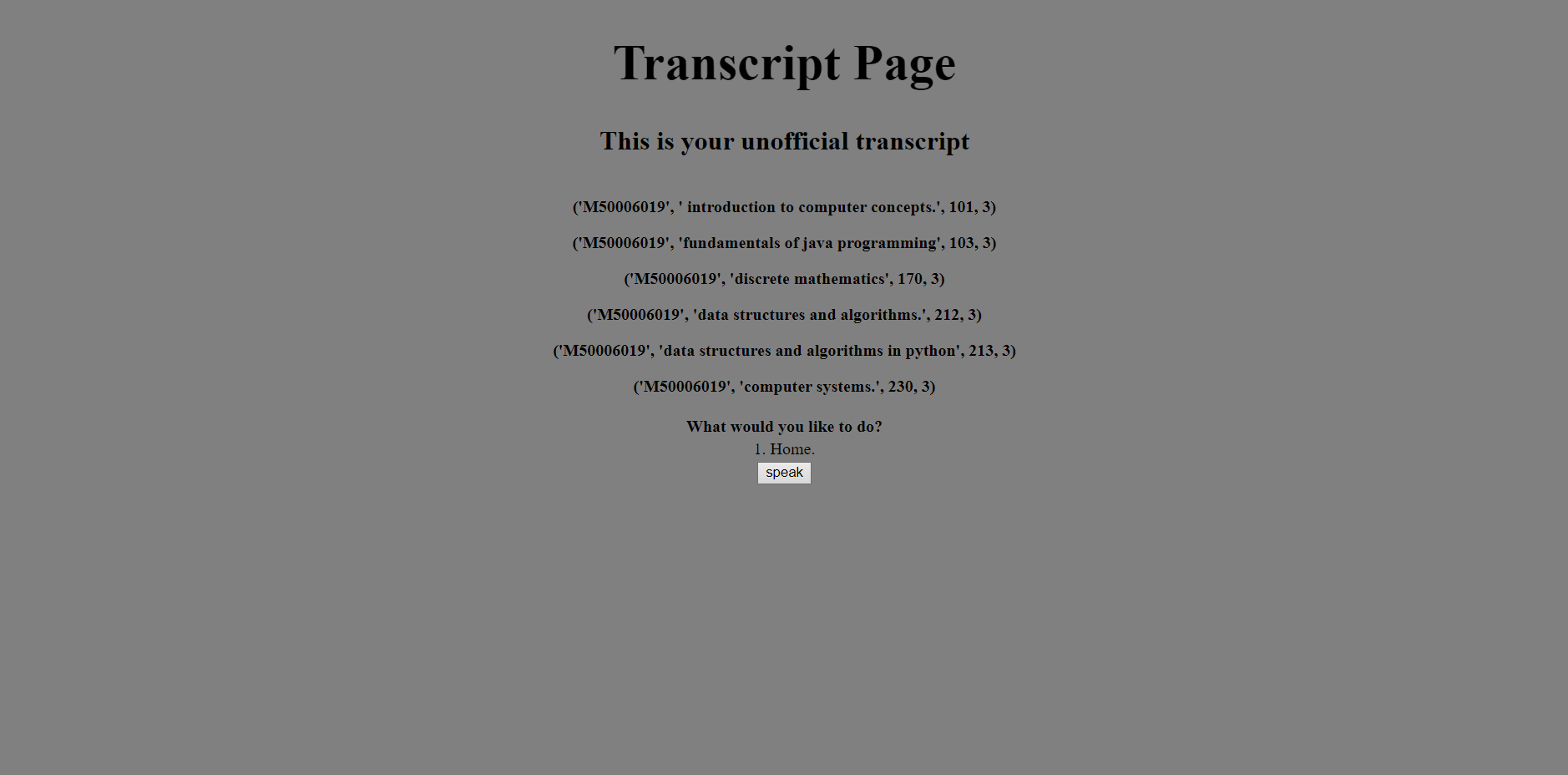


Figure 6 : This is the “Unofficial Transcript” screen

## From the home page the user can click the “speak” button and the user can say “Course Catalog” to bring them to the course catalog screen pictured in Figures 7 and 8. After the course catalog is displayed, the user has a few options. They can click the “speak” button, say “course information on (course name)”, then they will be redirected to a screen similar to Figure 9. They can click the “speak” button, say “home”, then they will be redirected to the home screen.

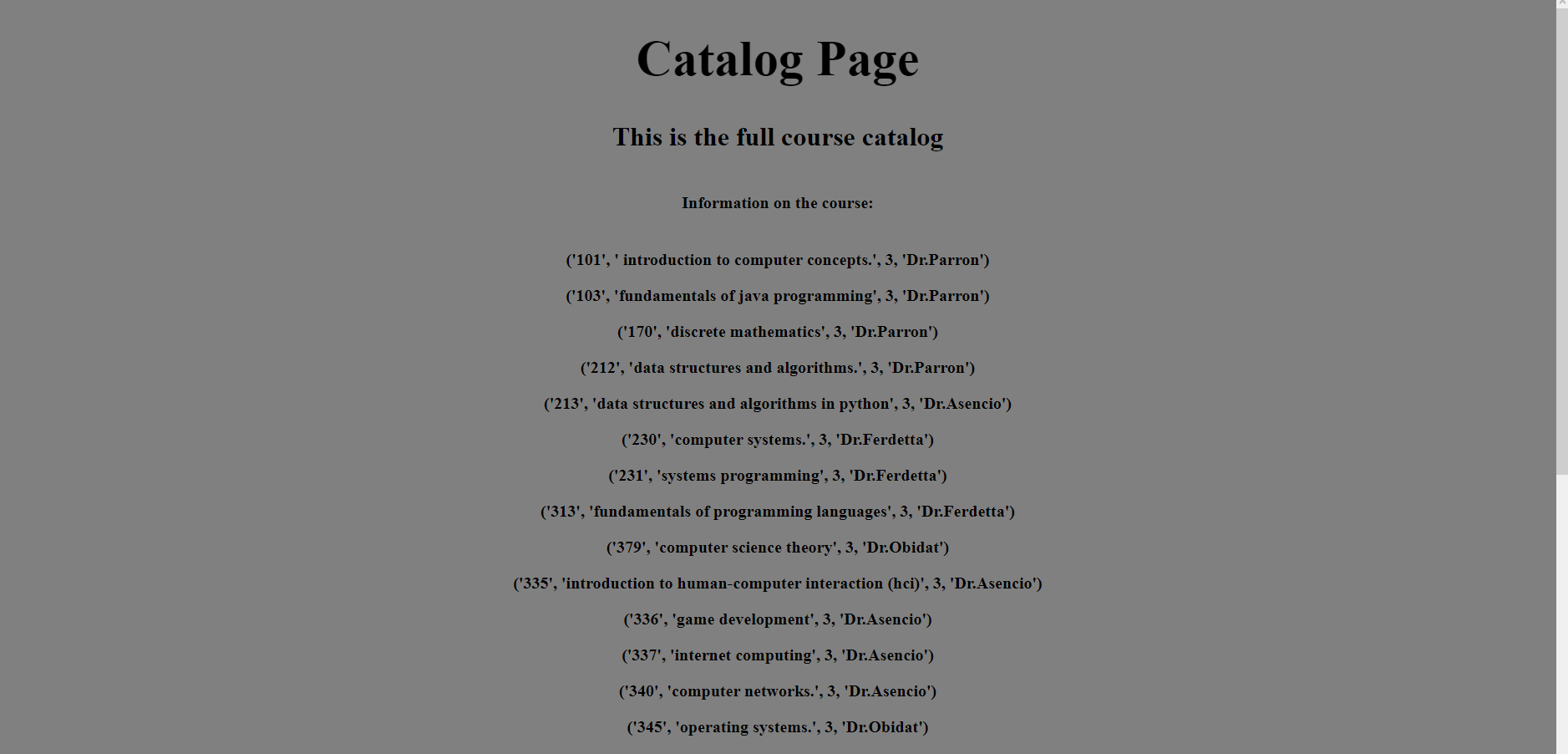


Figure 7 : This is the top of the “Course Catalog” screen



Figure 8 : This is the bottom of the “Course Catalog” screen

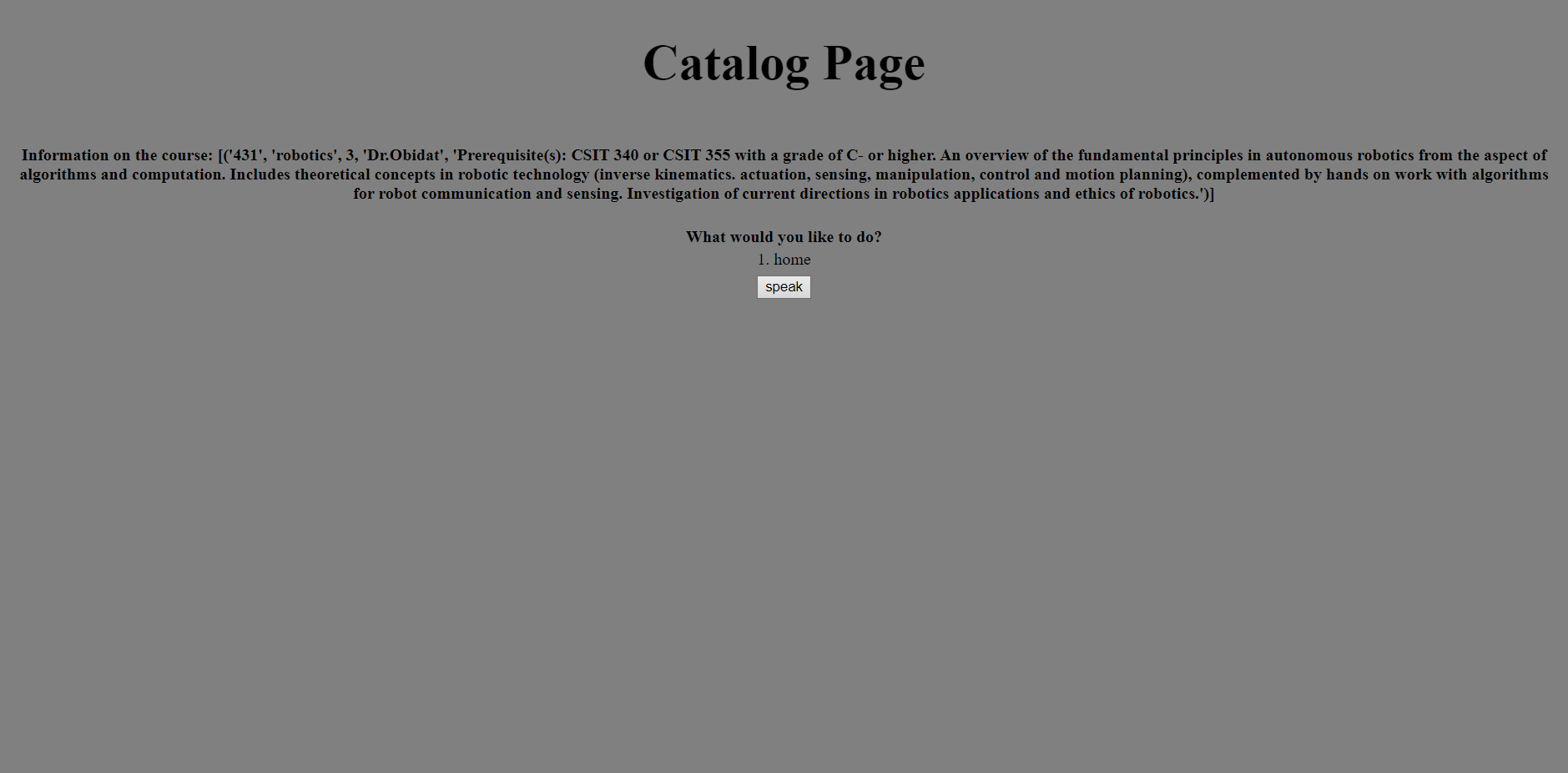


Figure 9 : This is “Course Information” screen for Robotics

## From the home page the user can click the “speak” button and the user can say “Advisement” to bring them to the academic advisement screen pictured in Figure 10. After the course catalog is displayed, the user has a few options. They can click the “speak” button, say “course information on (course name)”, then they will be redirected to a screen similar to Figure 9. They can click the “speak” button, say “home”, then they will be redirected to the home screen.

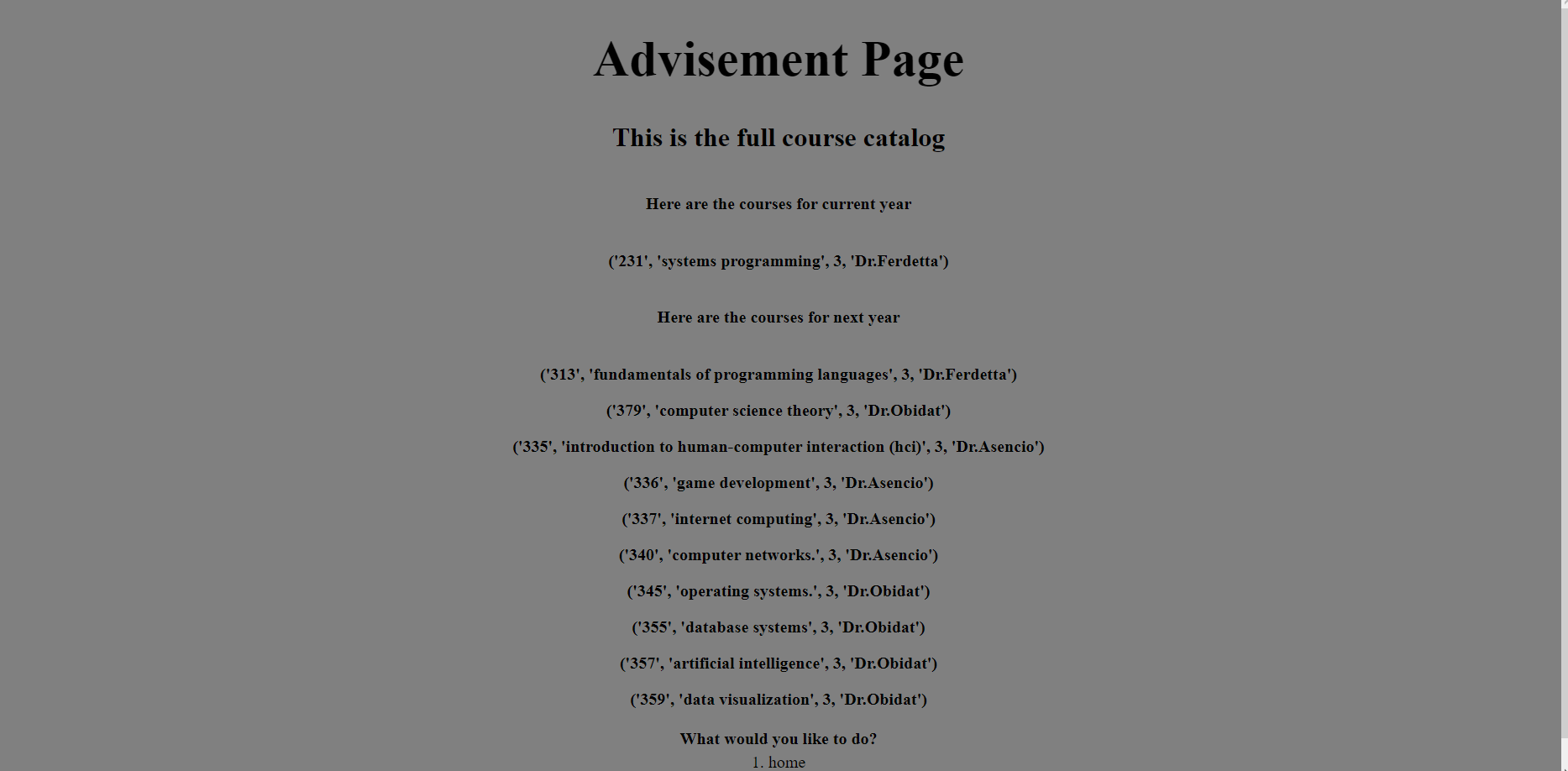
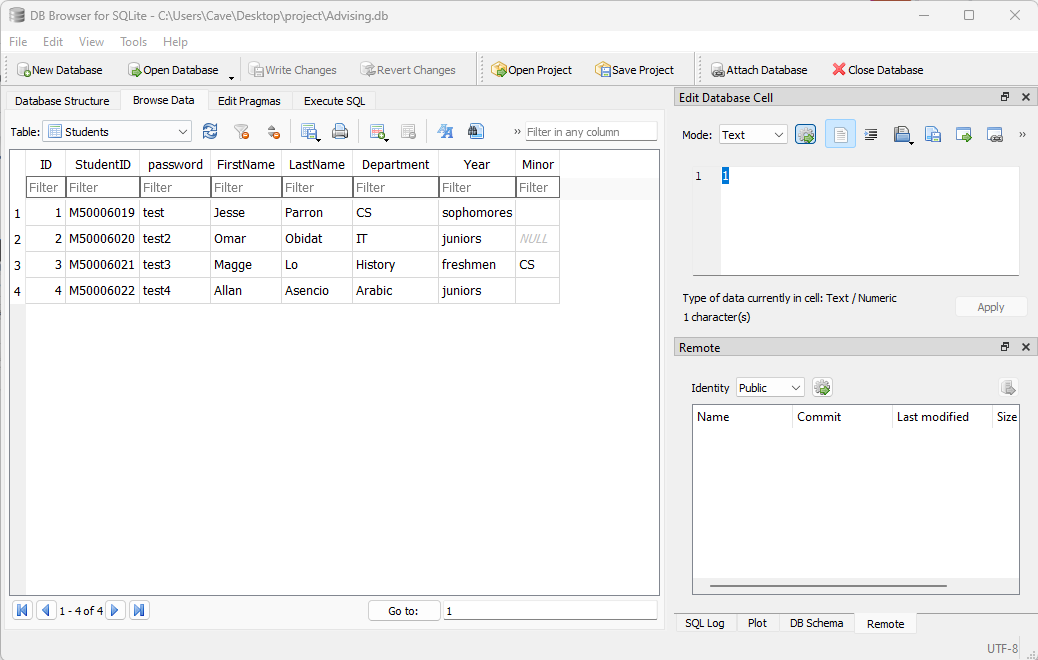
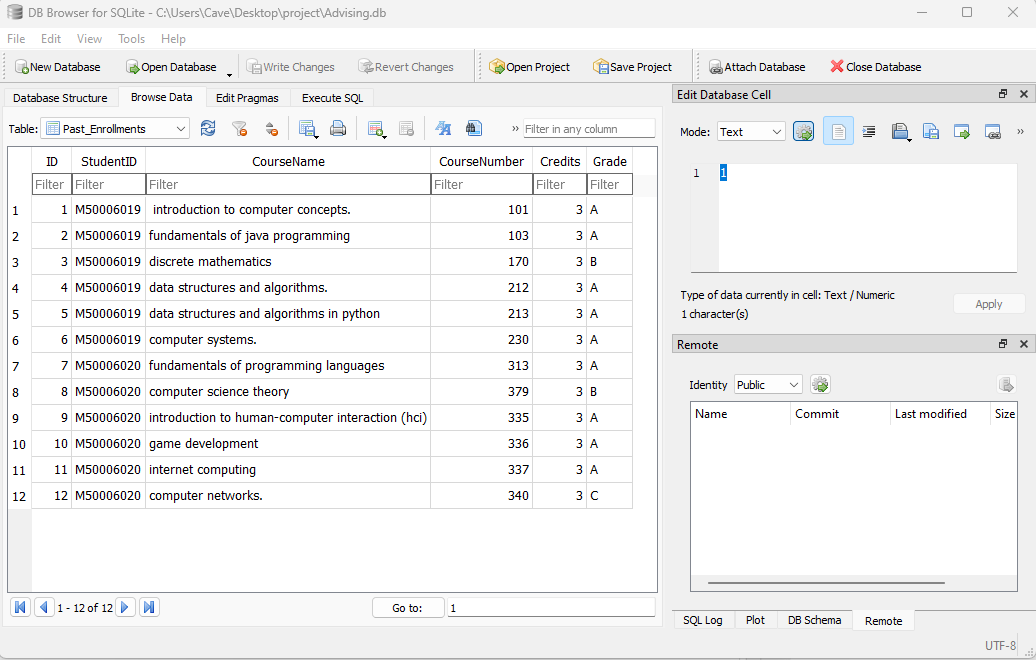


Figure 10 : This is the “Academic Advisement” screen

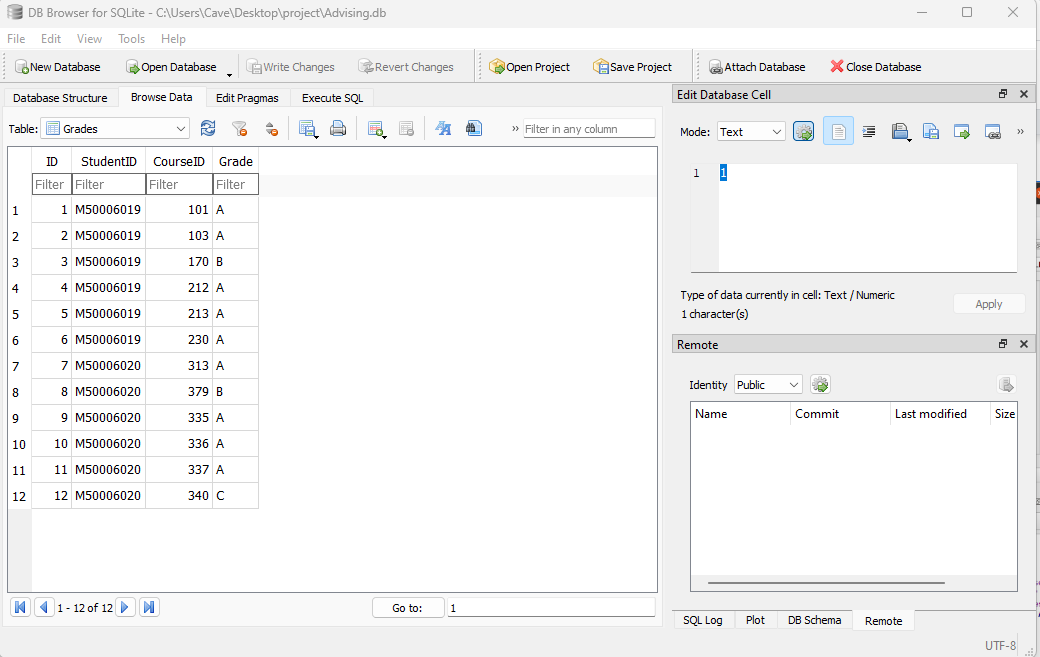
## The table that has the student profile that will be connected to the login page(below):



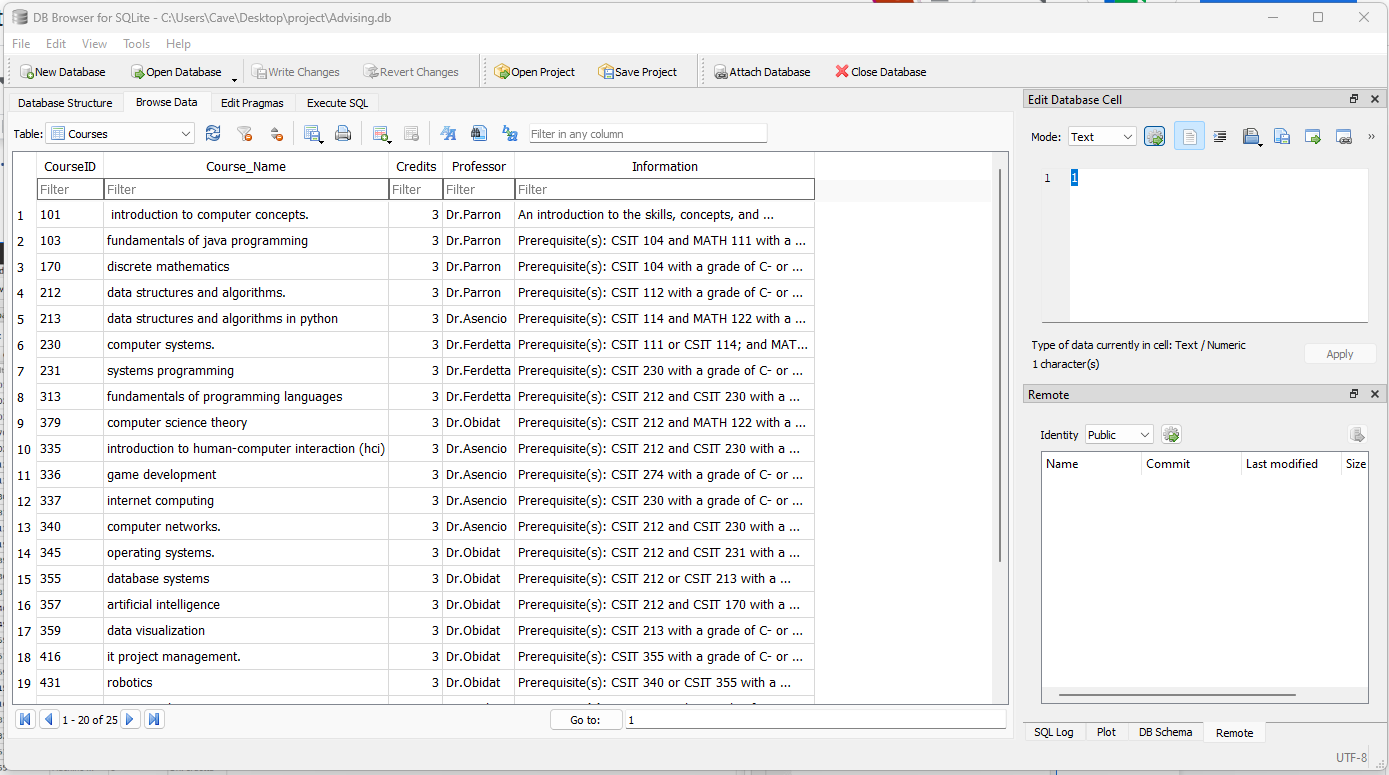
## The table that will connect each student to the classes that they have taken(below)



## This table will show the grades of each student for GPA calculation (below):



## This table shows all available courses and the professors teaching them (below):



# Main Code

The attached main code creates a session, connect to the server, and speech recognition.

from flask import Flask, render\_template, request, session

import sqlite3 as sql

import speech\_recognition as sr

import os

import webbrowser

import pyttsx3

dirname = os.path.dirname(\_\_file\_\_)

app = Flask(\_\_name\_\_, template\_folder = os.path.join(dirname,'pages'))

app.config['SESSION\_TYPE'] = 'filesystem'

app.config["SESSION\_PERMANENT"] = False

import pyttsx3

url = "http://127.0.0.1:5000"

webbrowser.open\_new\_tab(url)

def SpeakText(command):

# Initialize the engine

engine = pyttsx3.init()

voices = engine.getProperty('voices')

engine.setProperty('voice', voices[1].id)

engine.say(command)

engine.runAndWait()

def speak():

# get audio from the microphone

r = sr.Recognizer()

with sr.Microphone() as source:

#SpeakText("Please chose an option:")

audio1 = r.listen(source, timeout = 3, phrase\_time\_limit = 3)

audio = r.recognize\_google(audio1)

return audio

def current\_GPA():

points = 0

grade\_c = {"A":4,"A-":3.67

,"B+":3.33,"B":3.0

,"B-":2.67, "C+":2.33

,"C":2.0,"C-":1.67

,"D+":1.33,"D":1.0

,"F":0}

studentID = session['username']

with sql.connect("Advising.db") as con:

cur = con.cursor()

cur.execute('SELECT StudentID, Grade FROM Grades WHERE StudentID = ?', (studentID,))

con.commit

Student = cur.fetchall()

length = len(Student)

grades = []

for x in range(length):

grades.append(Student[x][1])

print(grades)

for grade in grades:

points += grade\_c[grade]

gpa = points / len(grades)

print(gpa)

return gpa

def transcript():

studentID = session['username']

with sql.connect("Advising.db") as con:

cur = con.cursor()

cur.execute('SELECT StudentID, CourseName, CourseNumber, Credits FROM Past\_Enrollments WHERE StudentID = ?', (studentID,))

con.commit

Student = cur.fetchall()

return Student

def catalog():

with sql.connect("Advising.db") as con:

cur = con.cursor()

cur.execute('SELECT CourseID, Course\_Name, Credits, Professor FROM Courses')

con.commit

courses = cur.fetchall()

return courses

def course\_Information(Course\_Name):

with sql.connect("Advising.db") as con:

cur = con.cursor()

cur.execute('SELECT \* FROM Courses WHERE Course\_Name = ?', Course\_Name)

con.commit

courses = cur.fetchall()

if courses == None:

courses = "There is no course with that ID"

return courses

else:

return courses

def what\_if\_GPA(what\_if\_grades):

points = 0

grade\_c = {"A":4.0,"A-":3.67

,"B+":3.33,"B":3.0

,"B-":2.67, "C+":2.33

,"C":2.0,"C-":1.67

,"D+":1.33,"D":1.0

,"F":0}

studentID = session['username']

with sql.connect("Advising.db") as con:

cur = con.cursor()

cur.execute('SELECT StudentID, Grade FROM Grades WHERE StudentID = ?', (studentID,))

con.commit

Student = cur.fetchall()

length = len(Student)

grades = []

i = 0

for x in range(length):

grades.append(Student[x][1])

while i < len(what\_if\_grades):

grades.append(what\_if\_grades[i])

i+=1

print(grades)

for grade in grades:

points += grade\_c[grade]

gpa = points / len(grades)

print(gpa)

return gpa

def Fcourses(courses):

finalcourselist =[]

for ID in courses:

if int(ID[0]) < 200:

finalcourselist.append(ID)

return finalcourselist

def Scourses(courses):

finalcourselist =[]

for ID in courses:

if int(ID[0]) > 200 and int(ID[0]) <300:

finalcourselist.append(ID)

return finalcourselist

def Jcourses(courses):

finalcourselist =[]

for ID in courses:

if int(ID[0]) > 300 and int(ID[0]) <400:

finalcourselist.append(ID)

return finalcourselist

def Secourses(courses):

finalcourselist =[]

for ID in courses:

if int(ID[0]) > 400:

finalcourselist.append(ID)

return finalcourselist

def filtercourses(courses):

studentID = session['username']

with sql.connect("Advising.db") as con:

cur = con.cursor()

cur.execute('SELECT CourseID FROM Grades WHERE StudentID = ?', (studentID,))

con.commit

courseids = cur.fetchall()

i = 0

j = 0

while i < len(courses):

while j < len(courseids):

if int(courses[i][0]) == int(courseids[j][0]):

courses.pop(i)

j = j +1

i = i +1

j = 0

return courses

@app.route('/')

def login():

return render\_template('login.html')

@app.route('/home', methods = ['POST'])

def loginVerification():

if request.method == 'POST':

ID = request.form['ID']

password = request.form['password']

with sql.connect("Advising.db") as con:

cur = con.cursor()

cur.execute('SELECT \* FROM Students WHERE StudentID = ? AND password = ?', (ID, password,))

con.commit

Student = cur.fetchone()

if Student == None:

return render\_template('login.html')

department = Student[5]

minor = Student[7]

global year

year = Student[6]

if department == 'CS' or department == 'IT' or minor == 'IT' or minor == 'CS':

if Student[1] == ID:

firstName = Student[3]

lastName = Student[4]

session['username'] = ID

question = 'Please vocally state which module you want'

SpeakText("Hello welcome to our new system")

return render\_template('home.html', question=question, firstName=firstName, lastName = lastName)

else:

return render\_template('login.html')

else :

SpeakText("Sorry currently this system is only for Computer science and Information Technology department")

return render\_template('login.html')

@app.route('/speak', methods = ['POST'])

def speakHomeModule():

home = "1. home"

q2 = "2. State the full course name you want information on (ex. Robotics)"

if request.method == 'POST':

value = request.form['button']

if value == '1':

audio = speak()

print("you said", audio)

if "transcript" in audio:

trans = transcript()

statement = "This is your unofficial transcript"

return render\_template('transcript.html', trans=trans, statement=statement)

elif "current GPA" in audio:

gpa = current\_GPA()

return render\_template("GPA.html", gpa=gpa, home=home)

elif "what if GPA" in audio:

what\_if\_statement = "Please state how many grades you want to enter"

statement\_whatif = "Please then state your what-if grade point in a numerical format from the list "

newstatement = "(A:4.0, A-:3.67,B+:3.33,B:3.0,B-:2.67,C+:2.33,C:2.0,C-:1.67,D+:1.33,D:1.0,F:0)"

return render\_template("GPA2.html", what\_if\_statement=what\_if\_statement, statement\_whatif=statement\_whatif, home=home,newstatement=newstatement)

elif "catalog" in audio:

courses = catalog()

statement = "This is the full course catalog"

return render\_template('catalog.html', courses=courses, statement=statement, home=home, q2=q2)

elif "advisement" in audio:

courses = catalog()

statement = "This is the full course catalog"

courses = filtercourses(courses)

if year == 'freshmen':

currentsemester = Fcourses(courses)

nextsemester = Scourses(courses)

elif year == 'sophomores':

currentsemester = Scourses(courses)

nextsemester = Jcourses(courses)

elif year == 'juniors':

currentsemester = Jcourses(courses)

nextsemester = Secourses(courses)

elif year == 'seniors':

currentsemester = Secourses(courses)

return render\_template('advising1.html', courses=courses, statement=statement, home=home, currentsemester=currentsemester)

return render\_template('advising.html', courses=courses, statement=statement, home=home, nextsemester=nextsemester ,currentsemester=currentsemester)

elif "home" in audio:

return render\_template('home.html')

elif "end session" in audio:

return render\_template('login.html')

else:

question = 'you didnt say a choice, please make sure to say a choice'

return render\_template('home.html', question=question)

if value == '2':

audio = speak()

print("you said", audio)

if "home" in audio:

return render\_template('home.html')

else:

statement = 'You didnt say a choice, please make sure to say a choice'

return render\_template('GPA.html', statement=statement, home=home)

if value == '3':

audio = speak()

print("you said", audio)

if "home" in audio:

return render\_template('home.html')

else:

statement = 'You didnt say a choice, please make sure to say a choice'

return render\_template('transcript.html', statement=statement)

if value == '4':

with sql.connect("Advising.db") as con:

cur = con.cursor()

cur.execute('SELECT Course\_Name FROM Courses')

con.commit

courses = cur.fetchall()

audio = speak()

audio = audio.lower()

course\_select = (audio,)

print("you said", audio)

print(course\_select)

x = 0

if "home" in audio:

return render\_template('home.html')

elif course\_select in courses:

while x < len(courses):

if course\_select == courses[x]:

print(courses[x])

course\_info = course\_Information(courses[x])

x +=1

break

else:

x +=1

return render\_template('catalog.html', course\_info=course\_info, home=home) #needs to be worked on

else:

statement = 'You didnt say a choice, please make sure to say a choice'

return render\_template('catalog.html', statement=statement, home=home)

if value == '5':

audio = speak()

print("you said", audio)

number = audio

if "home" in audio:

return render\_template('home.html')

number = int(number)

grades = []

k = 0

while k < number:

SpeakText("Please state the next what-if grade you want to enter")

#implement text to speech to state to enter the grade in a numerical float format from the list of grade points up in the defined function current\_GPA

audio = speak()

print("you said", audio)

if audio == '4.0':

temp = "A"

grades.append(temp)

elif audio == '3.67':

temp = "A-"

grades.append(temp)

elif audio == '3.33':

temp = "B+"

grades.append(temp)

elif audio == '3.0':

temp = "B"

grades.append(temp)

elif audio == '2.67':

temp = "B-"

grades.append(temp)

elif audio == '2.33':

temp = "C+"

grades.append(temp)

elif audio == '2.0':

temp = "C"

grades.append(temp)

elif audio == '1.67':

temp = "C-"

grades.append(temp)

elif audio == '1.33':

temp = "D+"

grades.append(temp)

elif audio == '1.0':

temp = "D"

grades.append(temp)

elif audio == '0.0':

temp = "F"

grades.append(temp)

else:

SpeakText("Grade point is not accepted, please try again")

#text to speech that, the grade point is not accepted, please try again

k=k-1

k += 1

#grades = [float(j) for j in grades]

print(grades)

final\_what\_if = what\_if\_GPA(grades)

if "home" in audio:

return render\_template('GPA2.html')

elif final\_what\_if != None:

statement = "this is your what-if GPA"

return render\_template('GPA2.html', statement=statement, home=home, final\_what\_if=final\_what\_if,temp = temp)

else:

statement = 'You didnt say a choice, please make sure to say a choice'

return render\_template('GPA2.html', statement=statement, home=home)

if value == '6':

audio = speak()

print("you said", audio)

courses = catalog()

statement = "This is the full course catalog"

courses = filtercourses(courses)

print( year == "sophomores")

if year == "freshmen":

currentsemester = Fcourses(courses)

nextsemester = Scourses(courses)

elif year == "sophomores":

currentsemester = Scourses(courses)

nextsemester = Jcourses(courses)

elif year == "juniors":

currentsemester = Jcourses(courses)

nextsemester = Secourses(courses)

elif year == "seniors":

currentsemester = Secourses(courses)

if "home" in audio:

return render\_template('home.html')

else:

statement = 'You didnt say a choice, please make sure to say a choice'

return render\_template('advising.html', courses=courses, statement=statement, home=home, nextsemester=nextsemester ,currentsemester=currentsemester)

if value == '7':

audio = speak()

print("you said", audio)

courses = catalog()

statement = "This is the full course catalog"

courses = filtercourses(courses)

if year == "freshmen":

currentsemester = Fcourses(courses)

nextsemester = Scourses(courses)

elif year == "sophomores":

currentsemester = Scourses(courses)

nextsemester = Jcourses(courses)

elif year == "juniors":

currentsemester = Jcourses(courses)

nextsemester = Secourses(courses)

elif year == "seniors":

currentsemester = Secourses(courses)

if "home" in audio:

return render\_template('home.html')

else:

statement = 'You didnt say a choice, please make sure to say a choice'

return render\_template('advising1.html', courses=courses, statement=statement, home=home, currentsemester=currentsemester)

if \_\_name\_\_ == '\_\_main\_\_':

app.secret\_key = 'super secret key'

app.run(debug=True)

# Catalog code

This is the html code that will be displayed for the user code:

<!DOCTYPE html>

<style>

h1{

text-align: center;

font-size: 3.2vw;

}

h2{

text-align:center;

font-size: 1.7vw;

}

.center {

margin-left: auto;

margin-right: auto;

}

td{

text-align: center;

}

.button{ text-align: center; }

</style>

<html>

<head>

<head>

<meta charset = "utf-8">

<title>Catalog</title>

</head>

</head>

<body style="background-color: grey;">

<h1> Catalog Page </h1>

<h2>{{statement}}</h2>

<table class = "center">

<tr>

<th>

<p>Information on the course: {{course\_info}}</p>

</th>

</tr>

<tr>

<th>

{% for course in courses %}

<p>{{course}}</p>

{% endfor %} </th>

</tr>

<tr>

<th>

What would you like to do?

</th>

</tr>

<tr>

<td>

{{home}}

</td>

</tr>

<tr>

<td>

{{q2}}

</td>

</tr>

</table>

<form action="speak" method="post">

<div class="button">

<button name="button" value="4">speak</button>

</div>

</form>

</body>

</html>

# Current GPA module:

<!DOCTYPE html>

<style>

h1{

text-align: center;

font-size: 3.2vw;

}

h2{

text-align:center;

font-size: 1.7vw;

}

.center {

margin-left: auto;

margin-right: auto;

}

td{

text-align: center;

}

.button{ text-align: center; }

</style>

<html>

<head>

<head>

<meta charset = "utf-8">

<title>GPA</title>

</head>

</head>

<body style="background-color: grey;">

<h1> GPA Page </h1>

<h2>{{statement}}</h2>

<table class = "center">

<tr>

<th>

This is your current GPA: {{gpa}}

</th>

</tr>

<tr>

<th>

What would you like to do?

</th>

</tr>

<tr>

<td>

{{home}}

</td>

</tr>

</table>

<form action="speak" method="post">

<div class="button">

<button name="button" value="2">speak</button>

</div>

</form>

</body>

</html>

# GPA what-if calculator

<!DOCTYPE html>

<style>

body{ margin:0 auto; }

h1{

text-align: center;

font-size: 3.2vw;

}

h2{

text-align:center;

font-size: 1.7vw;

}

p{

text-align:center;

font-size: 1vw;

}

body {

max-width: max-content;

margin: auto;

}

.center {

margin-left: auto;

margin-right: auto;

}

td{

text-align: center;

}

form{ text-align: center; }

</style>

<html>

<head>

<meta charset = "utf-8">

<title>What if GPA</title>

</head>

</head>

<body style="background-color: grey;" class = "center">

<h1>GPA what-if Page</h1>

<p>{{what\_if\_statement}}</p>

<p>{{statement\_whatif}}</p>

<p>{{newstatement}}</p>

<p>{{statement}}</p>

<p>{{final\_what\_if}}</p>

<p>{{temp}}</p>

<p>Please select an option</p>

<p>{{home}}</p>

<form action="speak" method="post">

<button name="button" value="5">speak</button>

</form>

</body>

</html>

# Home interface code

The interface for system’s home screen as described below is developed using html

<!DOCTYPE html>

<style>

h1{

text-align: center;

font-size: 3.2vw;

}

h2{

text-align:center;

font-size: 1.7vw;

}

.center {

margin-left: auto;

margin-right: auto;

}

td{

text-align: center;

}

.button{ text-align: center; }

</style>

<html>

<head>

<head>

<meta charset = "utf-8">

<title>Home</title>

</head>

</head>

<body style="background-color: grey;">

<h1>Home Page</h1>

<h2>You are logged in</h2>

<table class = "center">

<tr>

<th>

{{question}}

</th>

</tr>

<tr>

<td>

1. Current GPA

</td>

</tr>

<tr>

<td>

2. What-if GPA

</td>

</tr>

<tr>

<td>

3. Unofficial Transcript

</td>

</tr>

<tr>

<td>

4. Course Catalog

</td>

</tr>

<tr>

<td>

5. Course Advisement

</td>

</tr>

<tr>

<td>

6. End session

</td>

</tr>

</table>

<form action="speak" method="post">

<div class="button">

<button name="button" value="1">speak</button>

</div>

</form>

</body>

</html>

# Login code

This lists system’s login page as described below is developed using CSS computer language.

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

<style>

.imgcontainer {

text-align: center;

padding-top: 0px;

padding-bottom: -100px;

}

img {

margin-top: 15px;

width: 400px;

height: auto;

}

/\*@font-face {

font-family: "Shenttpuro";

src: url(./shenttpuro/ShenttpuroFont.ttf);

}\*/

label {

text-align: center;

color: rgba(254, 149, 28, 255);

font-family: "Shenttpuro";

font-size: 15pt;

letter-spacing: 10px;

}

.container {

text-align: center;

}

input[type=text],

input[type=password] {

width: 300px;

padding: 12px 20px;

margin: 0px 10px;

display: inline-block;

border: 1px solid #ccc;

box-sizing: border-box;

}

.button1 {

background-color: rgba(254, 149, 28, 255);

color: white;

border: none;

cursor: pointer;

width: 300px;

font-size: 13pt;

margin-top: 15px;

margin-bottom: 5px;

font-family: "Shenttpuro";

letter-spacing: 10px;

}

.button2 {

background-color: rgba(254, 149, 28, 255);

color: white;

border: none;

cursor: pointer;

width: 300px;

font-size: 13pt;

margin-top: 10px;

margin-bottom: 5px;

font-family: "Shenttpuro";

letter-spacing: 10px;

}

.a2 {

text-decoration:none;

color: white;

}

.a1 {

text-decoration:none;

color: rgba(254, 149, 28, 255);

}

.p1 {

text-align: center;

font-family: "Shenttpuro";

font-size: 15pt;

letter-spacing: 10px;

color: rgba(254, 149, 28, 255);

}

</style>

</head>

<body style="background-color: grey;">

<form action = "/home" method = "POST">

<div class="container">

<label for="username"><b>Username</b></label>

</div>

<div class="container">

<input type="text" placeholder="Enter Username" name="ID" required>

</div>

<div class="container">

<label for="password" style="font-family: Shenttpuro; text-align: center" ;><b>Password</b></label>

</div>

<div class="container">

<input type="password" placeholder="Enter Password" name="password" required>

</div>

<div class="container">

<button class="button1" type="submit" name="Login">Login</button>

</div>

</form>

</body>

</html>

# Transcript code

The code described below is developed to generate the user's transcript.

<!DOCTYPE html>

<style>

h1{

text-align: center;

font-size: 3.2vw;

}

h2{

text-align:center;

font-size: 1.7vw;

}

.center {

margin-left: auto;

margin-right: auto;

}

td{

text-align: center;

}

.button{ text-align: center; }

</style>

<html>

<head>

<head>

<meta charset = "utf-8">

<title>Transcript</title>

</head>

</head>

<body style="background-color: grey;">

<h1> Transcript Page </h1>

<h2>{{statement}}</h2>

<table class = "center">

<tr>

<th>

{% for tran in trans %}

<p>{{tran}}</p>

{% endfor %} </th>

</tr>

<tr>

<th>

What would you like to do?

</th>

</tr>

<tr>

<td>

1. Home.

</td>

</tr>

</table>

<form action="speak" method="post">

<div class="button">

<button name="button" value="3">speak</button>

</div>

</form>

</body>

</html>

# Advising code

<!DOCTYPE html>

<style>

h1{

text-align: center;

font-size: 3.2vw;

}

h2{

text-align:center;

font-size: 1.7vw;

}

.center {

margin-left: auto;

margin-right: auto;

}

td{

text-align: center;

}

.button{ text-align: center; }

</style>

<html>

<head>

<head>

<meta charset = "utf-8">

<title>Catalog</title>

</head>

</head>

<body style="background-color: grey;">

<h1> Advisement Page </h1>

<h2>{{statement}}</h2>

<table class = "center">

<tr>

<th>

<p>Here are the courses for current year</p>

</th>

</tr>

<tr>

<th>

{% for course in currentsemester %}

<p>{{course}}</p>

{% endfor %} </th>

</tr>

<tr>

<th>

<p>Here are the courses for next year</p>

</th>

</tr>

<tr>

<th>

{% for course in nextsemester %}

<p>{{course}}</p>

{% endfor %} </th>

</tr>

<tr>

<th>

What would you like to do?

</th>

</tr>

<tr>

<td>

{{home}}

</td>

</tr>

</table>

<form action="speak" method="post">

<div class="button">

<button name="button" value="6">speak</button>

</div>

</form>

</body>

</html>