# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CSC174 server-side javascript**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Unit 04 LAB: a voice driven customer feedback form using

# google web speech api

# Objectives

In this lab assignment, students will learn:

* To read and write more advanced JavaScript programs with third-party scripts, anonymous functions, objects, and dynamic DOM element creation
* To debug JavaScript code in a browser or IDE of the student’s choice
* To understand JSON and DOM
* To integrate different technologies using JavaScript

# COURSE PREPARATION

You should have done your reading assignment listed under “Reading Assignment” and “Video Assignment” sections in BlackBoard. You should also have reviewed the lecture slides in BlackBoard. There is an optional section called “In Case You Don’t Know” in BlackBoard for those who have limited exposure to JavaScript language.

# WHat to submit

For this lab you need to submit the following files:

* **index.html (Part One)**
* **Short answer questions sheet (Part Two)**

# grading rubric:

Be sure to follow the Coding Standard Guidelines. You must properly indent and comment your code. This assignment is worth 100 points. (50 points from Part One and 50 points from Part Two)

* Indent code and insert comments to document your program. [5 pts]
* Program must be implemented and run with no syntax errors. [20 pts]
* Program must be implemented and run with no logic errors. [20 pts]
* Required source files should be zipped and uploaded to BlackBoard assignment drop box before the deadline. [5 points]

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part One: A Voice Driven Customer Feedback**

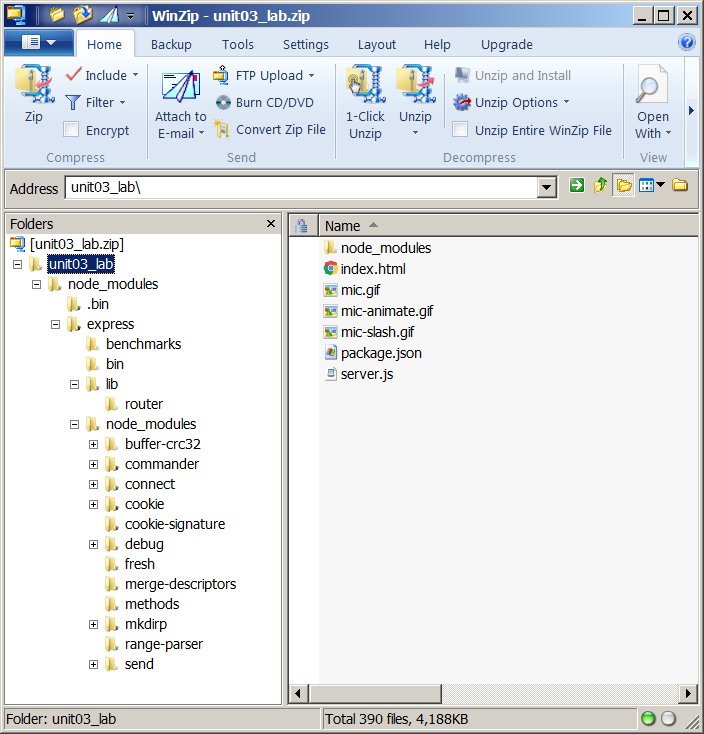
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Description:**

You are given a Node.js project that hosts a web page that allows the user to email some text using voice recognition technology created by Google engineer, Glen Shires. The detailed information of this technology can be found here: <https://developers.google.com/web/updates/2013/01/Voice-Driven-Web-Apps-Introduction-to-the-Web-Speech-API?hl=en>.

Your job is to modify this web page (index.html) so that it can be served as a customer feedback form for an imaginary institution called Wake Tech Credit Union (WTCU). Users can speak to the browser in various languages and then click “Send” button to send the feedback to the client-side JavaScript code (server.js), which logs the feedback entries on the server. The back-end code has been done by the instructor. You only need to do the front-end code (index.html).

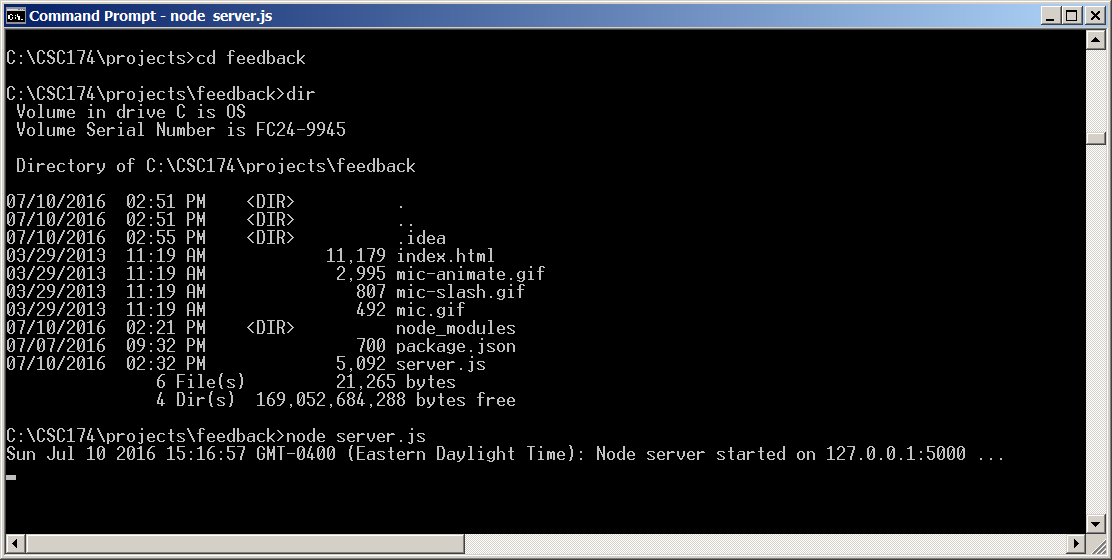
The instructor should provide you the source code in ***unit03\_lab.zip*** file in BlackBoard Unit 03 Lab section. There are various files in this folder that looks like below, but the most important files are ***index.html*** and ***server.js***. The former contains the JavaScript code that runs in your browser (client-side) and the latter contains the JavaScript code that runs in your Node.js engine (V8 engine).



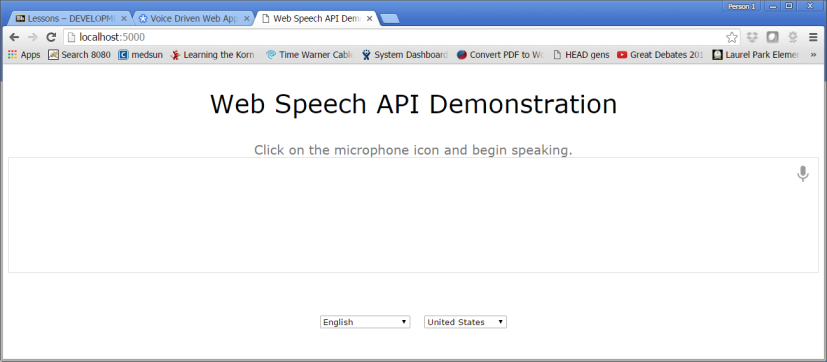
You can run this web site by either a Windows’ command window or MacOS’ terminal window by typing:

***node server.js***

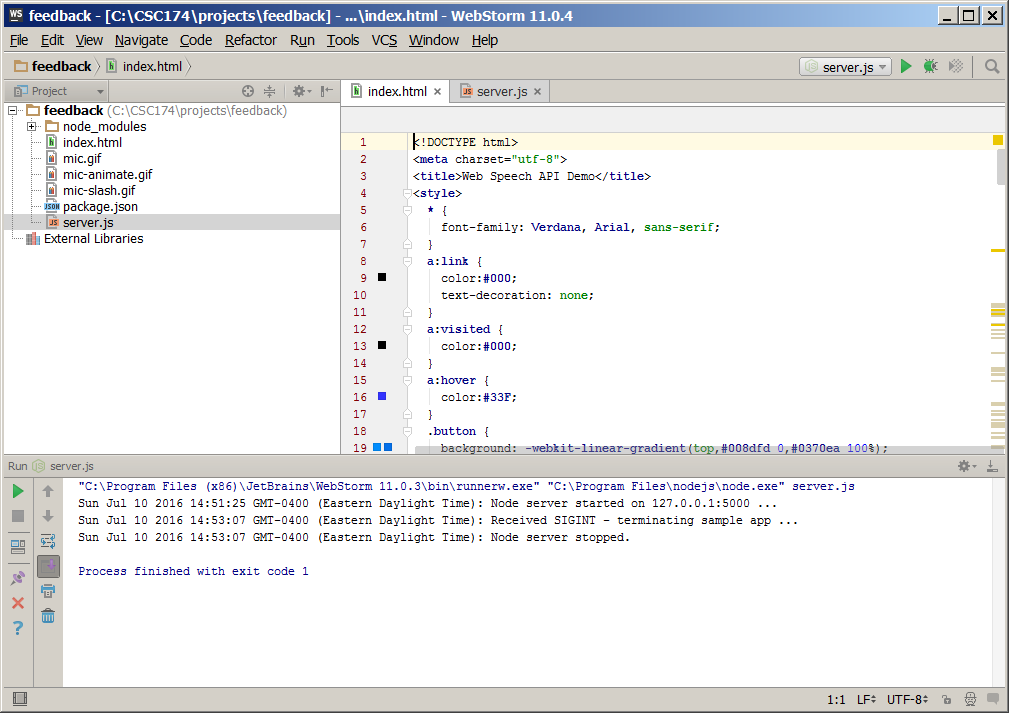
For example, I unzipped the project and put all the files in a folder under C:\CSC174\projects\feedback. Then I navigated to this directory using Windows ‘cd’ command. After I ran the node command above (in bold and italic), I got the following screenshot:



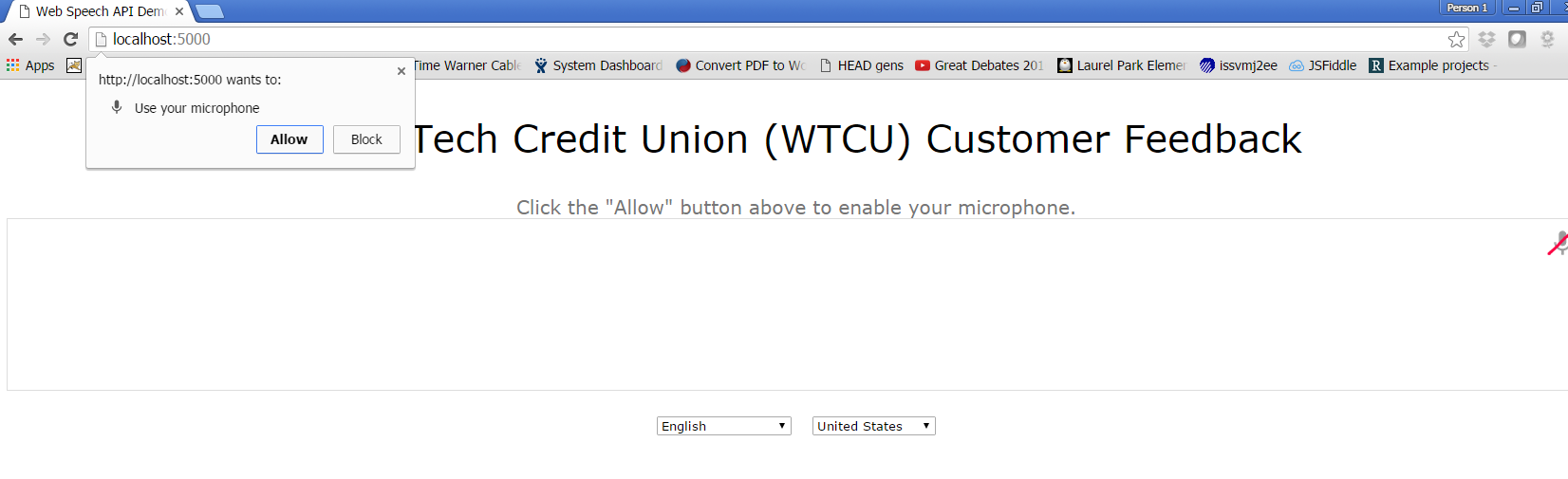
It tells us that the Node server is up running on localhost (IP: 127.0.0.1) port 5000. You can verify it by typing this <http://localhost:5000/> in your browser and you will get the following Google Web Speech demo page.

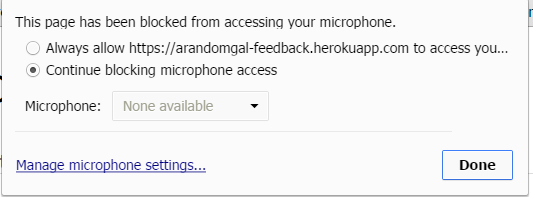
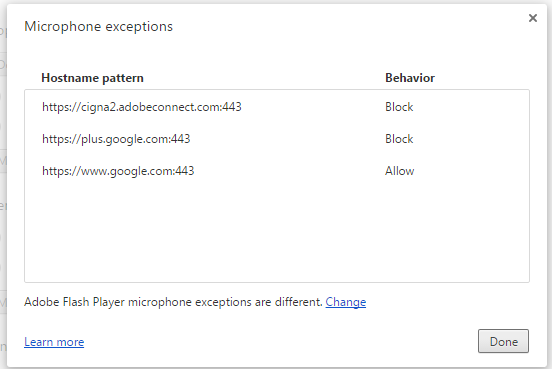


To stop this Node server, you can press <ctrl> + C in the command window or MacOS terminal window.

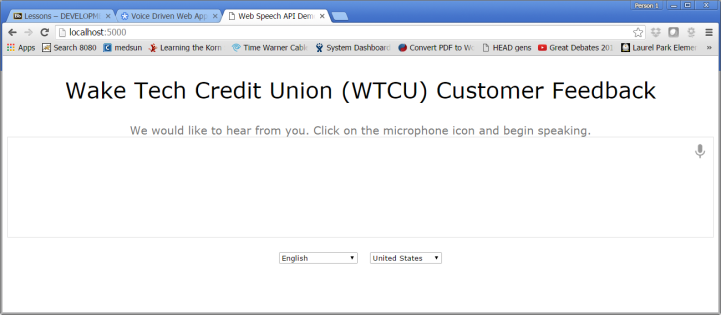
Alternatively you can run this project inside your favorite IDE. Below is a screenshots from WebStorm. I created a new general project in WebStorm and copied all the files into the project folder. I named my project ‘feedback’.

You can play around with this web page. You may have to modify your Chrome settings to allow Chrome browser have access to your microphone. I have seen some messages like below:

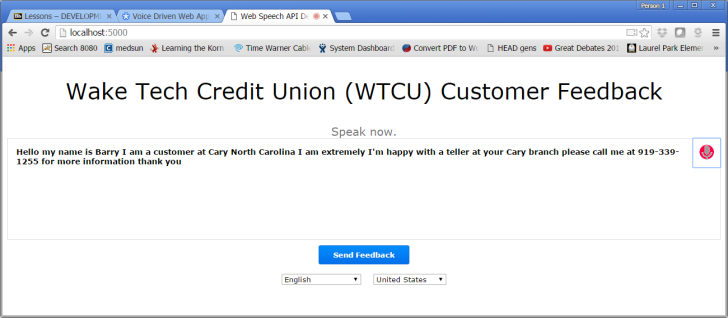


After you modify index.html, your web page should look like this:

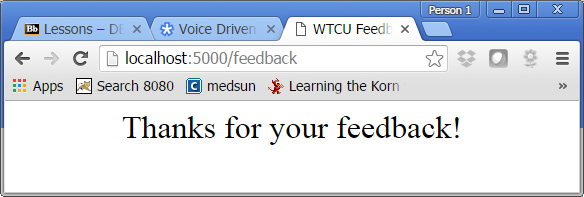


After you start speaking, you should see only one button “Send Feedback” appears.

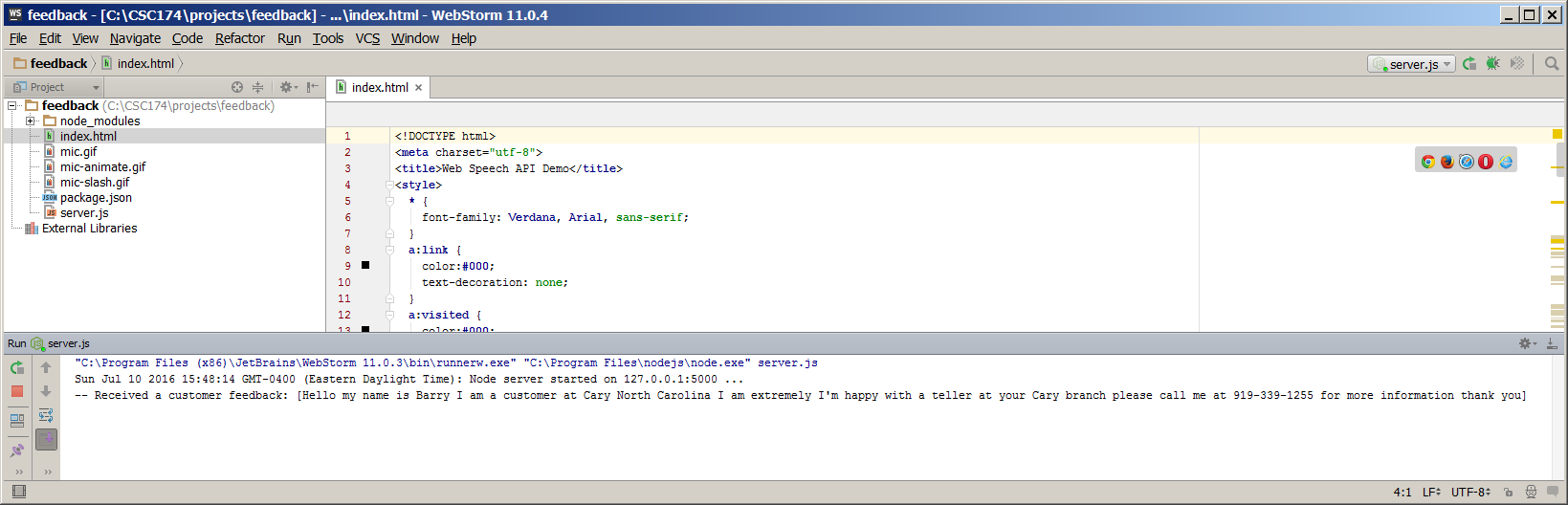


After you click “Send Feedback” button two things should happen:

1. You should get a confirmation page like this.



1. You should see the feedback message being logged on the Node server like this:



You only need to submit the modified index.html file.

**Hints:**

1. Locate the HTML code that you will put “Wake Tech Credit Union (WTCU) Customer Feedback” heading on the top.
2. Locate the HTML code that you will put “We would like to hear from you.”
3. Locate the HTML code that you will replace the “Copy/Paste” and “Email” buttons with just one button “Send Feedback”.
4. Modify the Javascript code which handles the onclick event for this new button you have added.
5. There is no form elements defined in this example by Google, but you can dynamically create and submit a form using DOM. Here is a sample solution. You can come up with your own:

<button id="send\_button" class="button" onclick="sendButton()">Feedback</button>

function sendFeedback() {

var feedbackform = document.createElement("FORM");

feedbackform.name="feedbackform";

feedbackform.method = "POST";

feedbackform.action = "/feedback";

var forminput = document.createElement("INPUT");

forminput.type = "TEXT";

forminput.name = "feedback";

forminput.value = final\_transcript;

feedbackform.appendChild(forminput);

feedbackform.submit();

}

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part Two: Short answer questions**

\*Questions 1 - 3 are based on the original index.html source code provided by the instructor. Questions 4 – 5 are based on the server.js source code.

Question 1: What type of variable is “**langs**”, is it an integer, a float, a boolean, a string, or an array?

Array

Question 2: What do you think ‘**webkitSpeechRecognition’** is on line 225? Do you think this mysterious thing exists only in Chrome web browser? Why?

webkitSpeechRecognition is an API that is available only in Chrome, until the API has been extended to also work in other browsers

Question 3: What do you think the ‘**recognition’** variable is? How many events and their corresponding handlers are registered with this object? What are they?

Recognition is the object of type webkitSpeechRecognition.

4 events – onstart, onerror, onend, onresult

Question 4: Can you point out where the in the server.js file the feedback is logged in the Node server console? (You can type the code out or give the line number.)

self.createRoutes = **function**() {  
 self.routes = { };  
  
self.routes['/feedback'] = **function**(req, res) {  
 console.log("-- Received a customer feedback: [" + req.body.feedback + "]");  
 res.send("<html><head><title>WTCU Feedback</title></head><body><div style='text-align:center; font-size:24pt'>Thanks for your feedback!</div></body></html>");  
 };  
  
 self.routes['/'] = **function**(req, res) {  
 res.setHeader('Content-Type', 'text/html');  
 res.send(self.cache\_get('index.html') );  
 };  
 };

Question 5: What do I need to change if I want this web application to run on localhost port 3333 instead of port 5000? (Which line to change? How will you change it?)

Change line 115, where the value is 5000 to 3333.

self.app.set('port', process.env.PORT || 5000);