CS122 Final Presentation

By: Frank Mock, Eddie Broeder, Sukanya Thakur

Contents

- Trip Advisor
- Nuance
- Intel Corporation
- Hewlett Packard Enterprise
- Brocade Communication Systems, Inc.

TripAdvisor Interview Results

TripAdvisor is a website that helps users find the best deals on anything related to travel such as hotels, restaurants and flights. It also provides user reviews so that travelers can be better informed.

How does it do this?

By linking with hundreds of other travel related websites by using data aggregation algorithms and services and then presenting meaningful results to the user.

TripAdvisor Interview Results

- Interviewed TripAdvisor employee Chuck Mock
- Works in Hotel Solutions dept. as Senior Software Engineer
- Provides the website and support for Hotel owners

TripAdvisor Interview Results (continued)

- Enterprise website written in Java and JavaScript
- Java - Backend and Frontend
- JavaScript - Frontend

TripAdvisor Interview Results (continued)

tripadvisor®

Why Java and JavaScript?

- Efficiency of lower level language is not necessary for building web UI
- Speed of lower level language not necessary "Java's fast enough"
- Low level details handled by: bash script ←→ Java program
- Abundance of JavaScript libraries to make developing web UI easier
- Java files work well with build software (Gradle → Groovy → JVM)

TripAdvisor Interview Results (continued)

- Servers run Java for DB work creating JSON and XML data packages
- Javascript consumes the JSON/XML for display
- Velocity - Java based template engine - used to make web pages

TripAdvisor Interview Results (continued)

- Python and Perl used for ETL (extract-transform-loads) data
- Engineers use linux development servers running bash to process data
- Gradle is used to automate the build process
- Python is used by Machine Learning and Analytics team

Nuance

Develop speech and imaging software and technologies for people, business and the healthcare markets.

“...Our people and our technology have pioneered the highest functioning speech software in the world, ceaselessly perfecting the ability for machines to recognize and emulate the human voice. We design and deliver technologies that intuitively link man and machine…”

(source: http://www.nuance.com/)

Nuance

“… there was heavy use of C, C++, Assembly and Python by our software engineers”

Why?

Nuance

C, C++, Assembly

- C, C++ used with heavy computational algorithms for speed and efficiency

(Examples) Pointer operations allow for great control and speed

Converting audio to a data model using Fourier transform algorithms

Quick data model access for fast language and acoustic comparison

- These languages blend well with Assembly code
- Assembly allows access to special chip set instructions.

(Example) Speech recognition on phones make use of SIMD to speed up calculations

Nuance

Python

- Python used for text/character identification and manipulation
- Model and analyze data using NumPy and SciPy

  Multi-dimensional models, C and C++ integration “glue”

- Also used for Quick proof-of-concept or implementation of ideas

  (Like an all purpose Swiss Army Knife language)

Nuance

Other Programming Languages Used?

Of course!

With inter-department work engineers often use Bash scripts or Perl

Intel Corporation

“You may know us for our processors. But we do so much more. Through computing innovation, we push the boundaries of smart and connected technology to make amazing experiences possible for every person on Earth. From powering the latest devices and the cloud you depend on, to driving policy, diversity, sustainability, and education, we create value for our stockholders, customers, and society.”

(source: http://www.intel.com/)

Intel Corporation

Multiple engineers in the Engineering Computing department were interviewed, with responsibilities ranging from web development, Windows and Linux system administration for tens of thousands of servers, and various other development and engineering roles.

Languages Used:

- Perl
- JavaScript
- Python
- Shell Scripting
- Windows Batch Scripting and VBScript

Intel Corporation

- Perl is probably the most commonly used language (in the Engineering Computing department), and will continue to be, as there is a huge amount of legacy code in the environment. There is automation that keeps tens of thousands of Linux servers running, nearly all of it written in Perl.
- Perl is also used for some (old) internal websites, via cgi scripts.

Intel Corporation

- JavaScript is widely used for newer internal websites, with AngularJS being the most common framework. It allows developers to easily bring up websites that are consistent with the company and department standards, while providing vast libraries of functionality, to get the most productivity out of the website.

Intel Corporation

- Python is becoming more common within Engineering Computing, but will not replace Perl anytime soon. This is due to the large amount of legacy Perl code, and the fact that many developers are very experienced with Perl, and do not have much Python experience.

- Shell scripting is very common, and is used often when engineers need to write some quick automation that doesn’t require all of the features of Perl, but interacts very closely with Linux OS features.

- Windows Batch file scripting is used for the majority of the automation of Windows system administration tasks, as is VBscript

Intel Corporation

Disclaimer: This information is based on interviews with Intel developers and engineers within a **single** department. Therefore, it does not provide any sort of accurate picture of the languages used by the majority of the company. i.e. if you don’t like any of the languages previously mentioned, you can probably find a job at Intel writing just about any language you’d like.

Brocade Communication Systems, Inc.

Brocade has a wide range of products for computer storage and networking. In addition to manufacturing its well known Fiber Channel and Network switches, Brocade makes Routers and software products for Software Defined Networking (SDN)

I interviewed Jimmy Yam who works as a development engineer for Fiber Channel switches. His main role is to develop switch diagnostic software.

Brocade Communication Systems, Inc.

BROCADE

Interview Results:

- C/C++ as the primary language for development
- Python and Bash for developing test scripts
- Use Eclipse as IDE for development of Diagnostic tools
- Use PyCharm as IDE for development of test scripts

Brocade Communication Systems, Inc.

BROCADE

Interview Results:

- Why C/C++ for Diagnostic Tools?
  - Allows to mix Assembly with the high level language
  - Pointers are used to interface with the switch hardware and volatile memory
  - C++ is used to implement top level architecture and code reuse as it helps design the tools in object oriented manner.
  - C/C++ give a wide range of control for developing software starting from top level design in objects to all the way up to low level memory pointers

Brocade Communication Systems, Inc.

BROCADE

Interview Results:

- Why Python for test scripting?
    - Development is easy and fast
    - Object oriented support is built-in and natural the Python
    - The language is simple without much syntactical complexity
    - Rich support for dealing with natural data structures such as List, Tuple, Set, Strings and Maps or dictionary
- Why not Perl?
    - Too confusing in syntax
    - Too many of %@$_
    - Hidden variables
    - Poor support for object oriented architecture

Hewlett Packard Enterprise

Hewlett Packard Enterprise makes hardware for mid range to high range enterprise servers. In addition the company has hardware and software products for storage and networking.

I interviewed *Tilak Sarkar* who is a Test Automation Engineer. He explained to me Python and Perl are the two most prominently used languages in test automation. His group has recently migrated from Perl to Python as test automation language.

Hewlett Packard Enterprise

Interview Results

Tilak’s group recently migrated from Perl to Python for developing test automation.

Why Python?

- Becoming the choice of industry, Perl is losing to it
- Python is simpler than Perl in syntax and style
- Python's duck typing is simple, in the same time Python is not quiet about type conflicts in the run time
- Dealing with errors and logs are simple and efficient
- Regex is not be as powerful as Perl, but is powerful enough to remain simple

Hewlett Packard Enterprise

Interview Results

Why Perl was dropped

- Perl is silent about *type* errors
- Not very readable
- Too many choices to do a thing makes the code hard to maintain
- No good support for OO

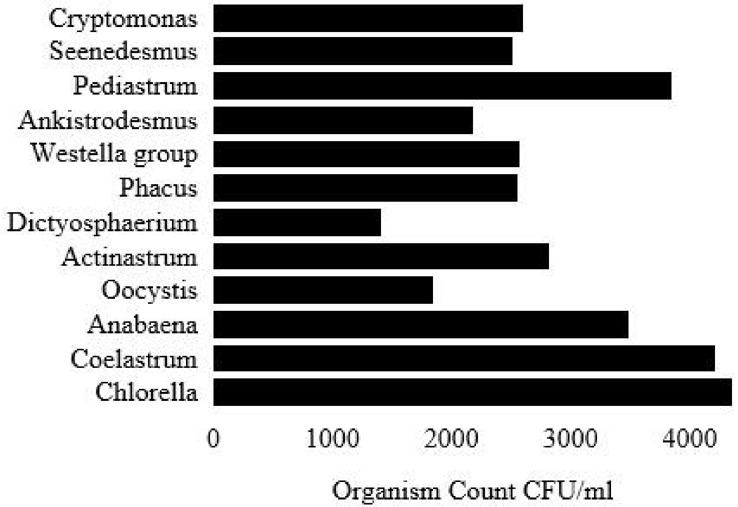
Final Project

Project that integrates Python, Perl and JavaScript

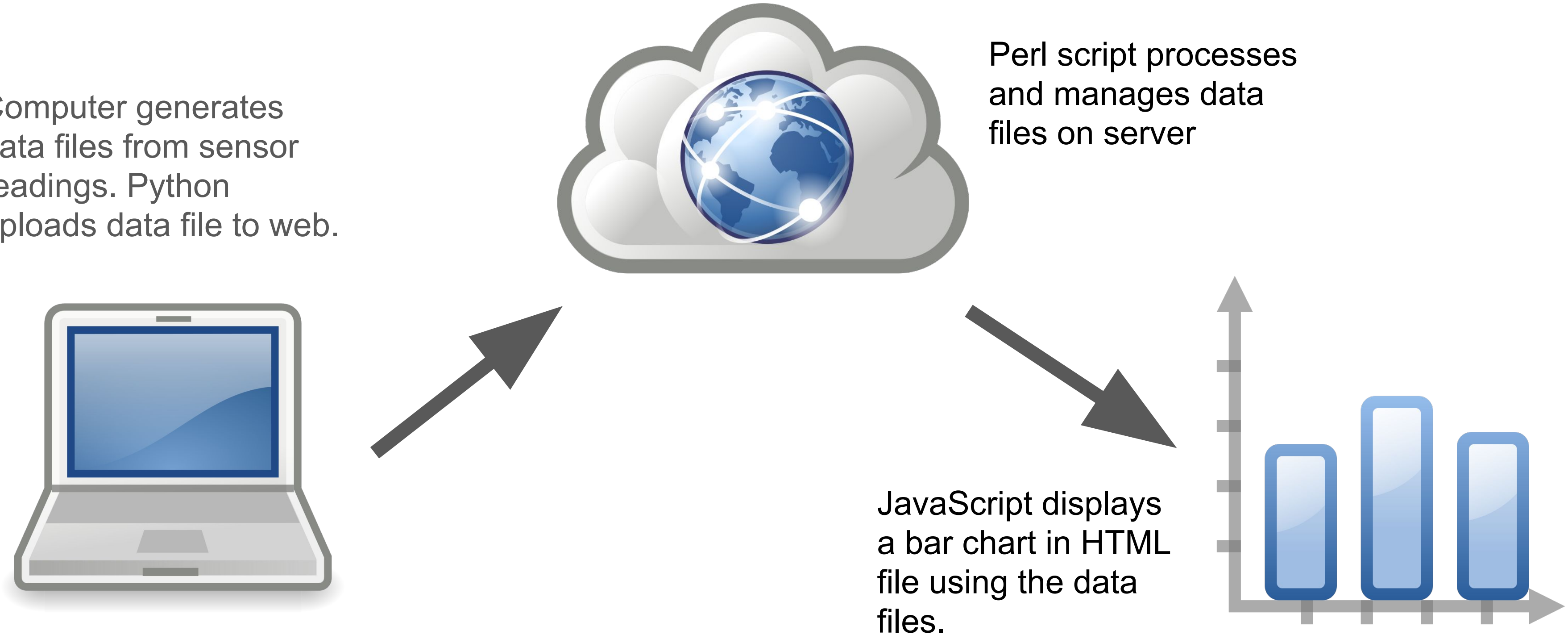
Demonstrates the programming languages that could be used on a biology computer network system.

A biologist installs computer controlled test equipment at pond or lake to monitor daily micro-organism count.

Data uploaded to web daily for analysis.



Python, Perl, JavaScript Application



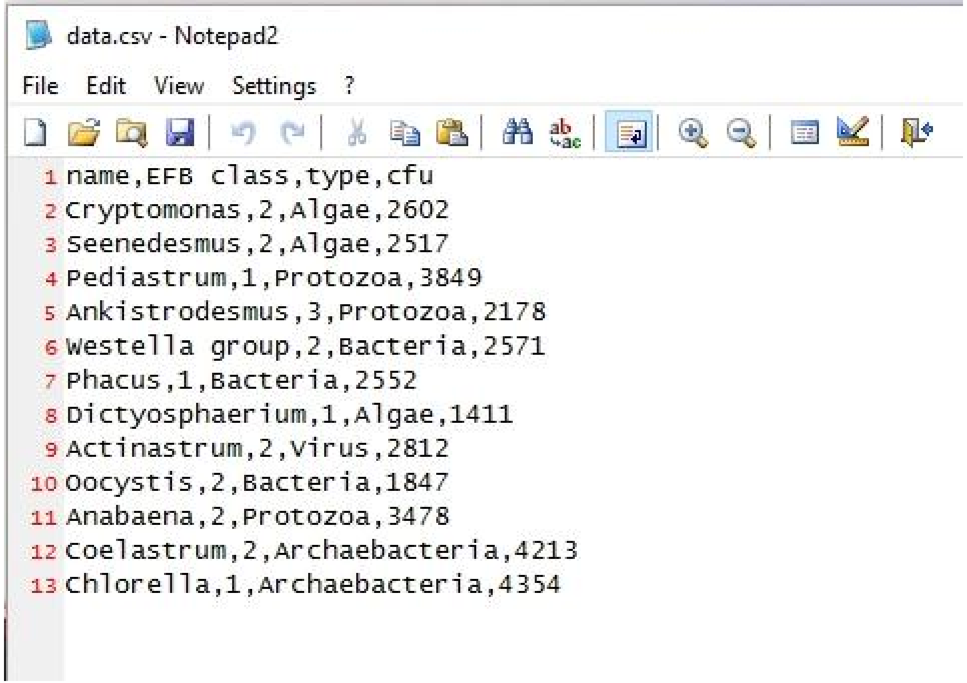
Computer generates data files from sensor readings. Python uploads data file to web.

Perl script processes and manages data files on server

JavaScript displays a bar chart in HTML file using the data files.

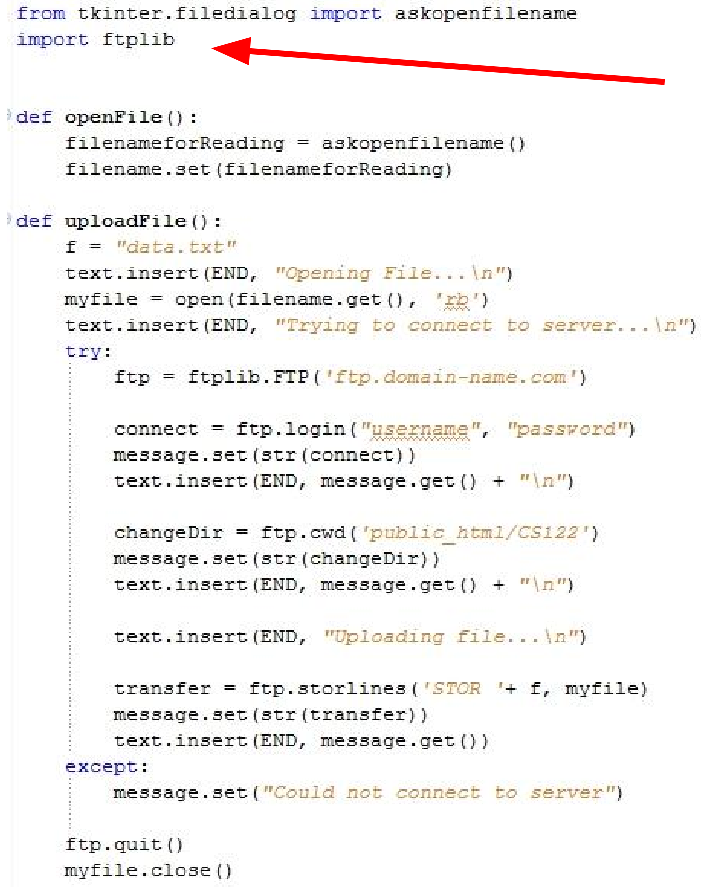
Python, Perl, JavaScript Application (continued)

Field equipment collects water sample, processes sample, and generates data file.



```
name,EFB class,type,cfu
Cryptomonas,2,Algae,2602
Seenedesmus,2,Algae,2517
Pediastrum,1,Protozoa,3849
Ankistrodesmus,3,Protozoa,2178
Westella group,2,Bacteria,2571
Phacus,1,Bacteria,2552
Dictyosphaerium,1,Algae,1411
Actinastrum,2,Virus,2812
Oocystis,2,Bacteria,1847
Anabaena,2,Protozoa,3478
Coelastrum,2,Archaebacteria,4213
Chlorella,1,Archaebacteria,4354
```

Python, Perl, JavaScript Application (continued)

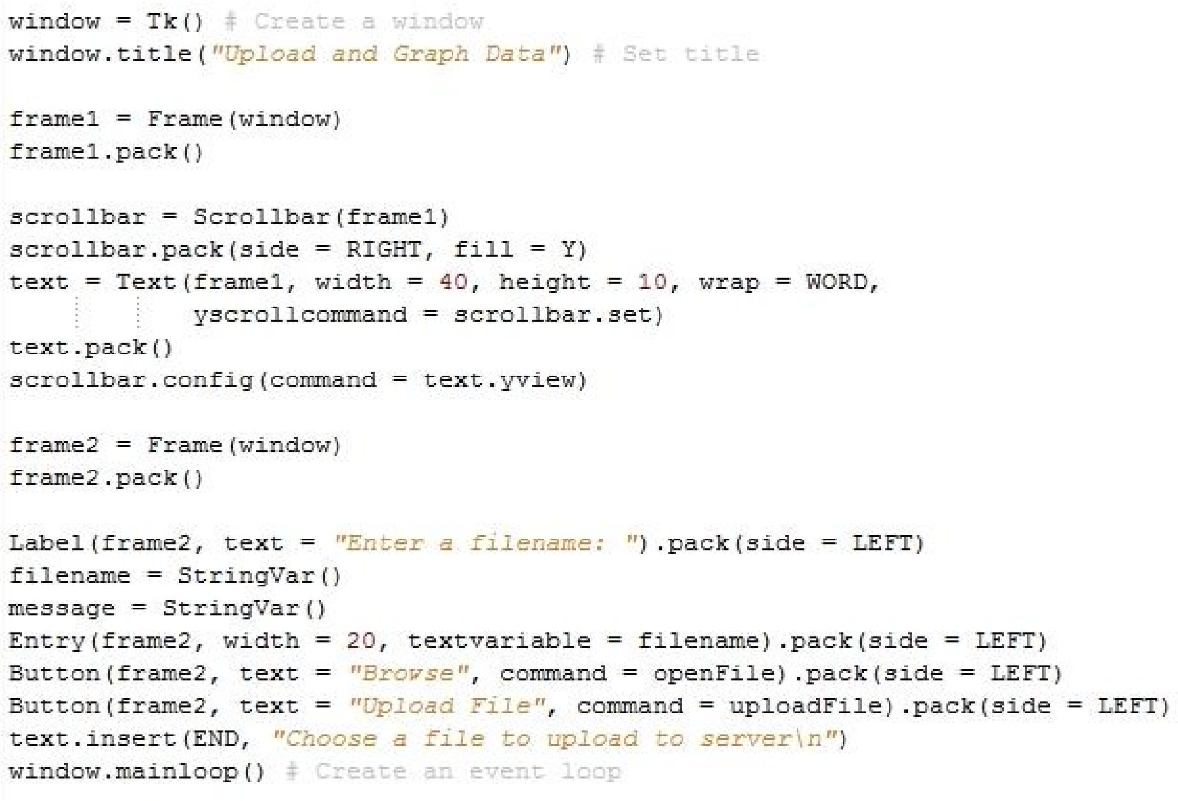


```
from tkinter.filedialog import askopenfilename
import ftplib


def openFile():
    filenameforReading = askopenfilename()
    filename.set(filenameforReading)

def uploadFile():
    f = "data.txt"
    text.insert(END, "Opening File...\n")
    myfile = open(filename.get(), 'rb')
    text.insert(END, "Trying to connect to server...\n")
    try:
        ftp = ftplib.FTP('ftp.domain-name.com')

        connect = ftp.login("username", "password")
        message.set(str(connect))
        text.insert(END, message.get() + "\n")

        changeDir = ftp.cwd('public_html/CS122')
        message.set(str(changeDir))
        text.insert(END, message.get() + "\n")

        text.insert(END, "Uploading file...\n")

        transfer = ftp.storlines('STOR '+ f, myfile)
        message.set(str(transfer))
        text.insert(END, message.get())
    except:
        message.set("Could not connect to server")

    ftp.quit()
    myfile.close()
```

Main.py
Python program that uses the built-in module ftplib to upload data file to web.

Python, Perl, JavaScript Application (continued)

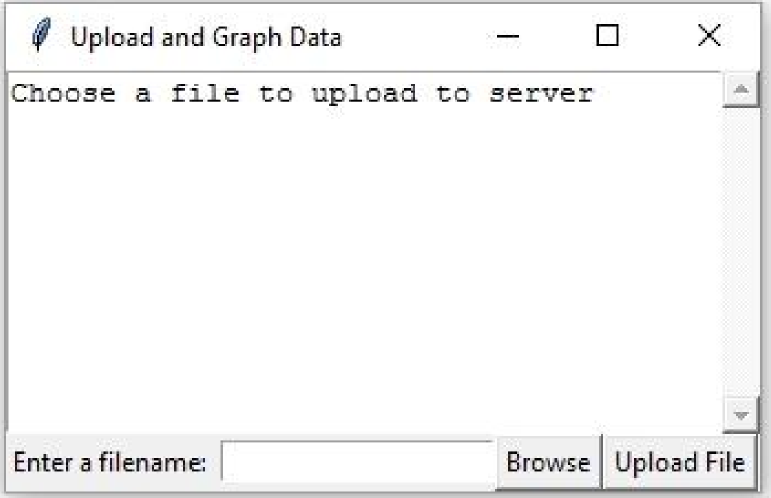
Main.py (continued)



```
window = Tk() # Create a window
window.title("Upload and Graph Data") # Set title

frame1 = Frame(window)
frame1.pack()

scrollbar = Scrollbar(frame1)
scrollbar.pack(side = RIGHT, fill = Y)
text = Text(frame1, width = 40, height = 10, wrap = WORD,
            yscrollcommand = scrollbar.set)
text.pack()
scrollbar.config(command = text.yview)

frame2 = Frame(window)
frame2.pack()

Label(frame2, text = "Enter a filename: ").pack(side = LEFT)
filename = StringVar()
message = StringVar()
Entry(frame2, width = 20, textvariable = filename).pack(side = LEFT)
Button(frame2, text = "Browse", command = openFile).pack(side = LEFT)
Button(frame2, text = "Upload File", command = uploadFile).pack(side = LEFT)
text.insert(END, "Choose a file to upload to server\n")
window.mainloop() # Create an event loop
```

Python, Perl, JavaScript Application (continued)

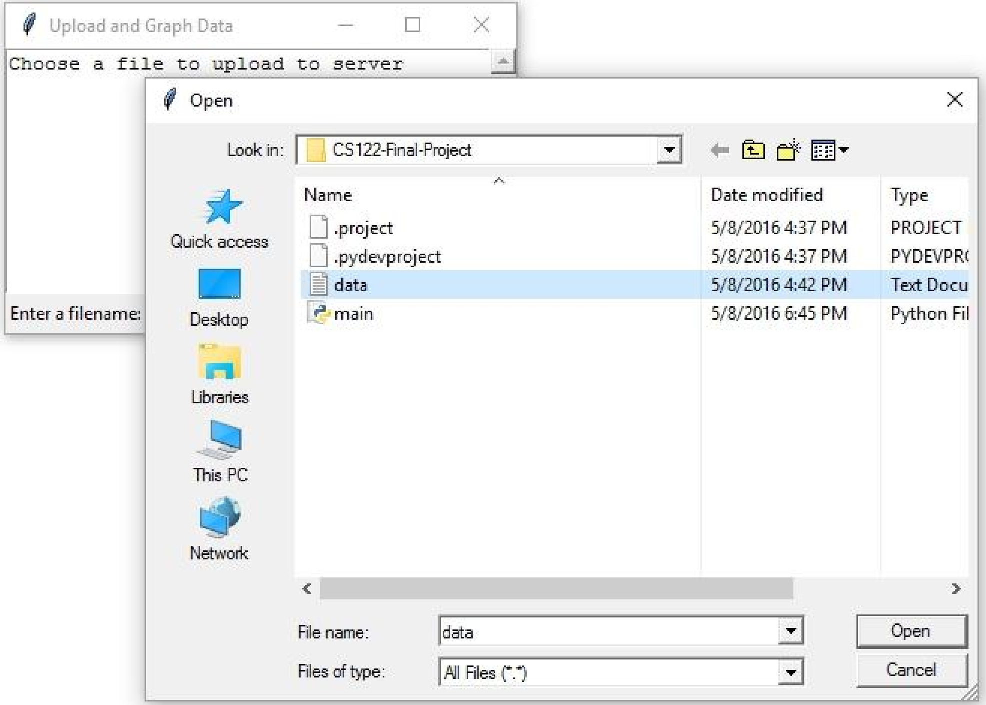
The Python GUI app used to upload data file.



```
Choose a file to upload to server
```

Python, Perl, JavaScript Application (continued)

Browse to select a file to upload.



```
Choose a file to upload to server
```

Enter a filename:

Look in: CS122-Final-Project

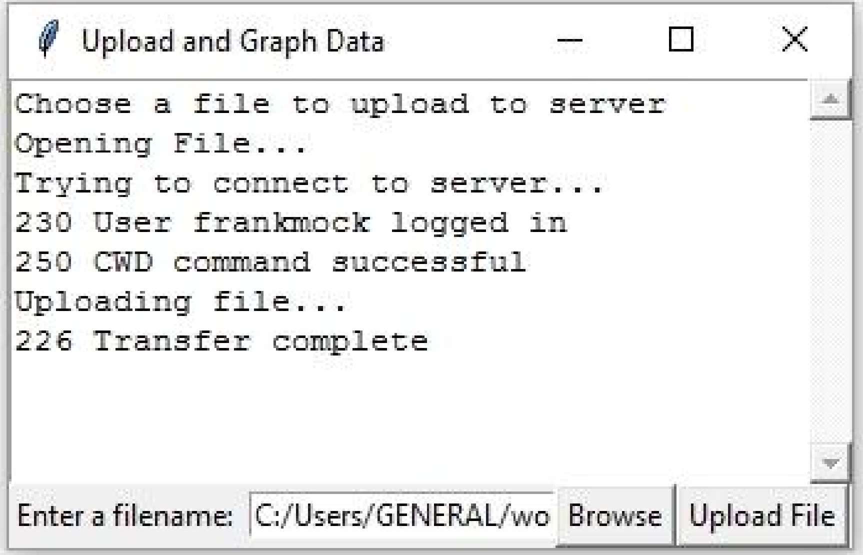
Name	Date modified	Type
.project	5/8/2016 4:37 PM	PROJECT
.pydevproject	5/8/2016 4:37 PM	PYDEVPR(
data	5/8/2016 4:42 PM	Text Docu
main	5/8/2016 6:45 PM	Python Fil

File name: data

Files of type: All Files (*.*)

Python, Perl, JavaScript Application (continued)

Successful upload. Python built-in ftplib library used for FTP



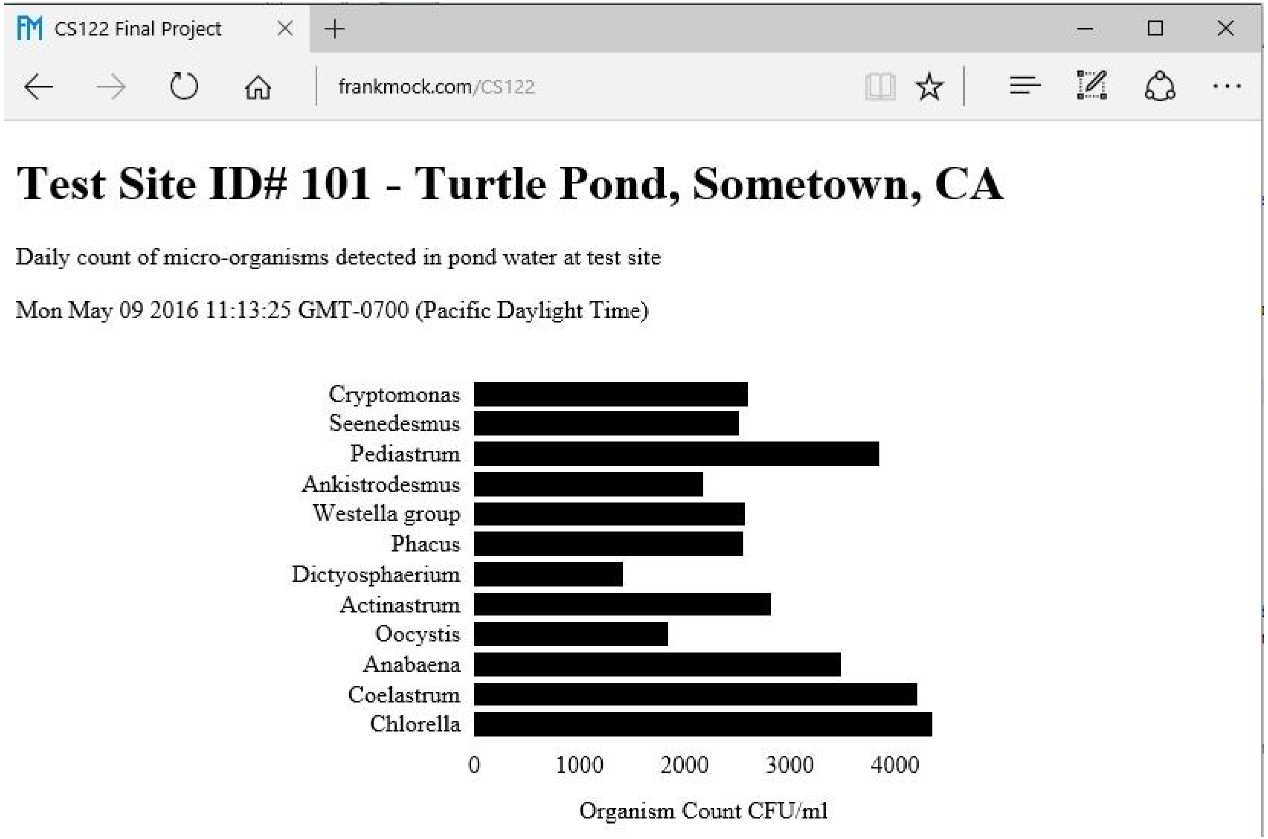
```
Choose a file to upload to server
Opening File...
Trying to connect to server...
230 User frankmock logged in
250 CWD command successful
Uploading file...
226 Transfer complete
```

Enter a filename: C:/Users/GENERAL/wo

Python, Perl, JavaScript Application (continued)

index.html

D3.js uses the data in the data.csv file to display a graph in a web page.



Test Site ID# 101 - Turtle Pond, Sometown, CA

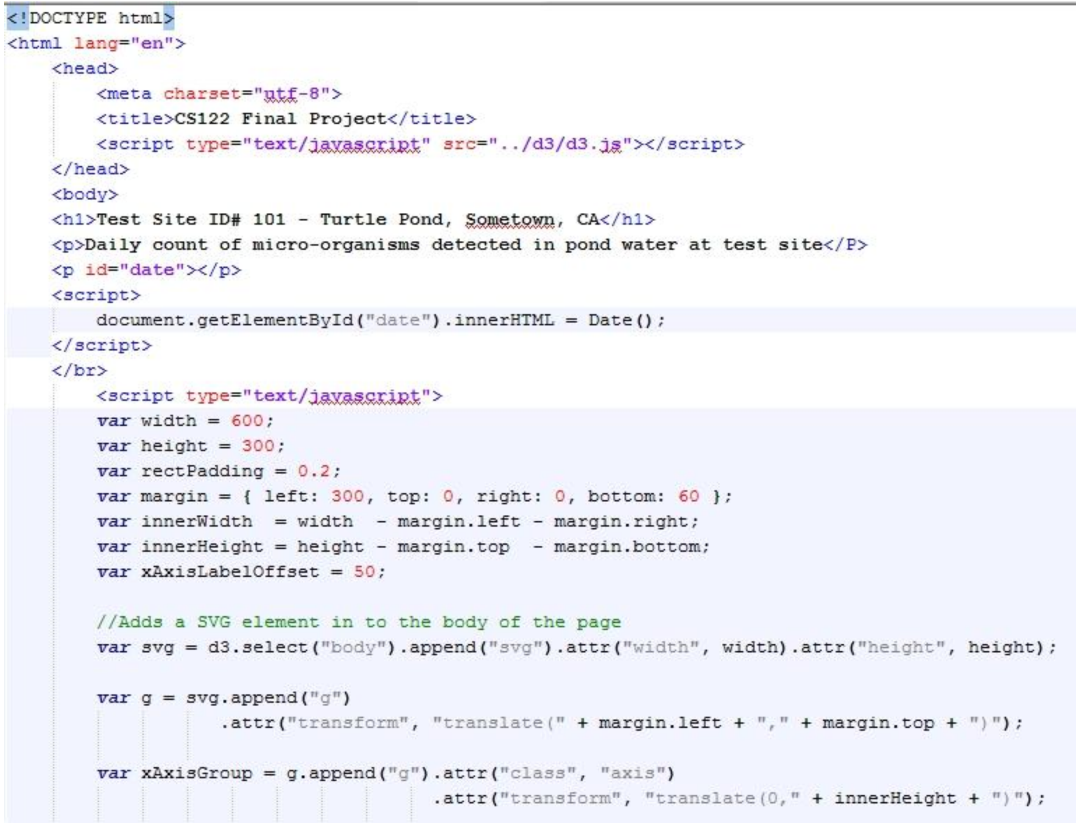
Daily count of micro-organisms detected in pond water at test site

Mon May 09 2016 11:13:25 GMT-0700 (Pacific Daylight Time)

Python, Perl, JavaScript Application (continued)

index.html

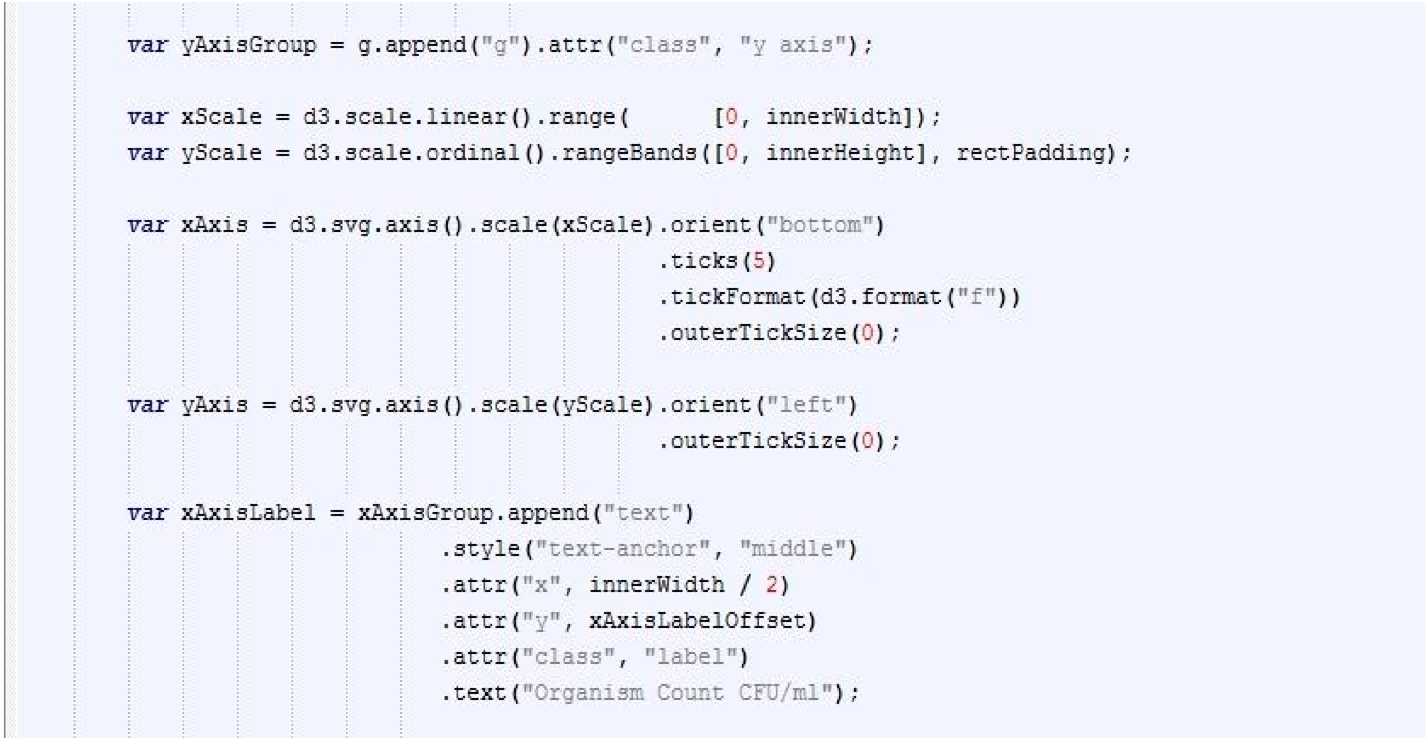
Web page with JavaScript used to display graph.



```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>CS122 Final Project</title>
        <script type="text/javascript" src="../d3/d3.js"></script>
    </head>
    <body>
    <h1>Test Site ID# 101 - Turtle Pond, Sometown, CA</h1>
    <p>Daily count of micro-organisms detected in pond water at test site</P>
    <p id="date"></p>
    <script>
        document.getElementById("date").innerHTML = Date();
    </script>
    </br>
        <script type="text/javascript">
        var width = 600;
        var height = 300;
        var rectPadding = 0.2;
        var margin = { left: 300, top: 0, right: 0, bottom: 60 };
        var innerWidth  = width  - margin.left - margin.right;
        var innerHeight = height - margin.top  - margin.bottom;
        var xAxisLabelOffset = 50;

        //Adds a SVG element in to the body of the page
        var svg = d3.select("body").append("svg").attr("width", width).attr("height", height);

        var g = svg.append("g")
                   .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

        var xAxisGroup = g.append("g").attr("class", "axis")
                                      .attr("transform", "translate(0," + innerHeight + ")");
```

Python, Perl, JavaScript Application (continued)

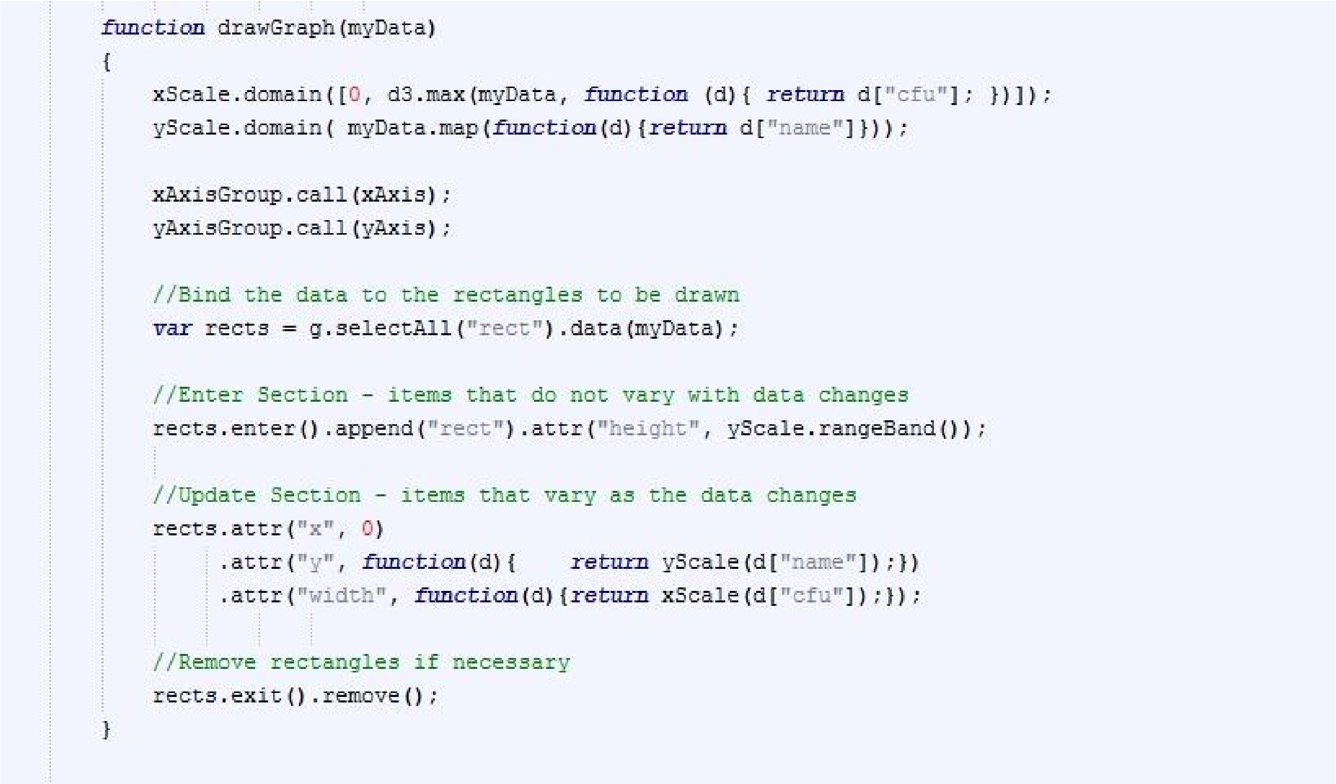
index.html
(continued)



```
        var yAxisGroup = g.append("g").attr("class", "y axis");

        var xScale = d3.scale.linear().range(      [0, innerWidth]);
        var yScale = d3.scale.ordinal().rangeBands([0, innerHeight], rectPadding);

        var xAxis = d3.svg.axis().scale(xScale).orient("bottom")
                                               .ticks(5)
                                               .tickFormat(d3.format("f"))
                                               .outerTickSize(0);

        var yAxis = d3.svg.axis().scale(yScale).orient("left")
                                               .outerTickSize(0);

        var xAxisLabel = xAxisGroup.append("text")
                               .style("text-anchor", "middle")
                               .attr("x", innerWidth / 2)
                               .attr("y", xAxisLabelOffset)
                               .attr("class", "label")
                               .text("Organism Count CFU/ml");
```

Python, Perl, JavaScript Application (continued)

index.html
(continued)



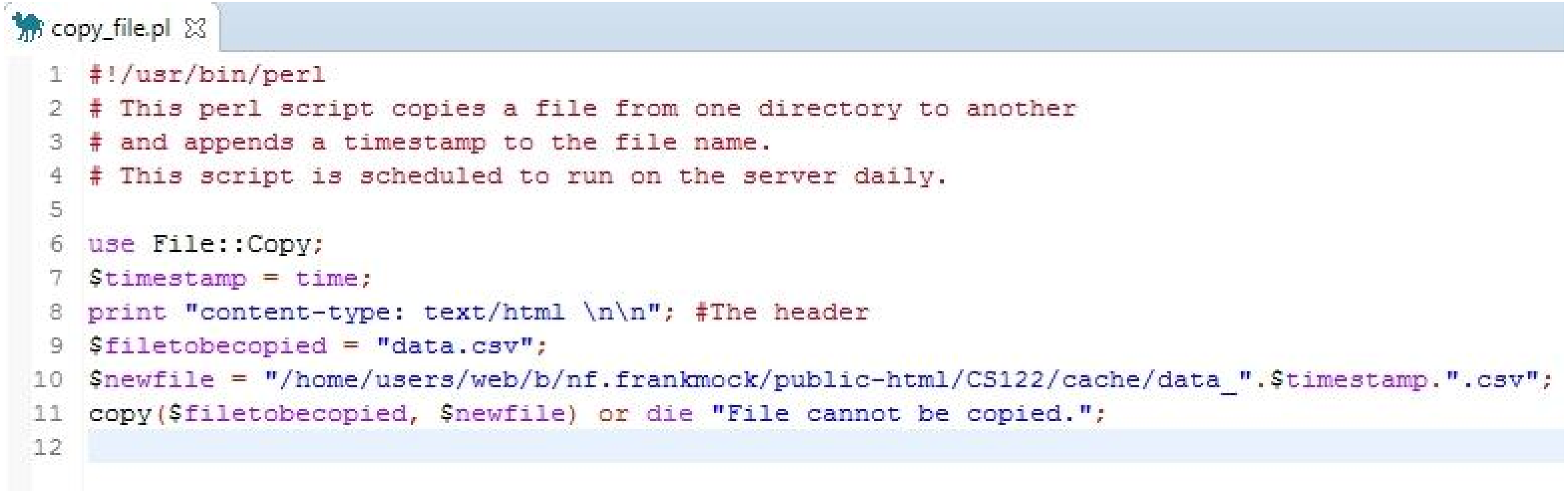
```
function drawGraph(myData)
{
    xScale.domain([0, d3.max(myData, function (d){ return d["cfu"]; })]);
    yScale.domain( myData.map(function(d){return d["name"]}));

    xAxisGroup.call(xAxis);
    yAxisGroup.call(yAxis);

    //Bind the data to the rectangles to be drawn
    var rects = g.selectAll("rect").data(myData);

    //Enter Section - items that do not vary with data changes
    rects.enter().append("rect").attr("height", yScale.rangeBand());

    //Update Section - items that vary as the data changes
    rects.attr("x", 0)
         .attr("y", function(d){    return yScale(d["name"]);})
         .attr("width", function(d){return xScale(d["cfu"]);});

    //Remove rectangles if necessary
    rects.exit().remove();
}
```

Python, Perl, JavaScript Application (continued)

copy_file.pl The server is scheduled to run this script automatically.

Perl script runs daily saving previous days data file to a data cache directory.

Perl Script



```perl
#!/usr/bin/perl
# This perl script copies a file from one directory to another
# and appends a timestamp to the file name.
# This script is scheduled to run on the server daily.

use File::Copy;
$timestamp = time;
print "content-type: text/html \n\n"; #The header
$filetobecopied = "data.csv";
$newfile = "/home/users/web/b/nf.frankmock/public-html/CS122/cache/data_".$timestamp.".csv";
copy($filetobecopied, $newfile) or die "File cannot be copied.";
```