

Research Topics in CS

The is a group project. You will work in your table groups of 4 for this project

Research an open source project, library, framework, or unique language or toolkit of your choice. Create a working demo from your toolset, and a 7-10 minute presentation based on your tool or toolset. Provide your well documented demo code via github.

This assignment is an open ended opportunity to explore a technology related to computer science that is of interest to you that will add to the collective knowledge of your classmates. The focus must be software related, but your choice of software language is up to you. The only requirement is that you will have to tie your presentation back to AP CSA in some way, and you will have to create and present a working prototype or demo code.

You must create something unique with the tools of choice, not simply walk through an existing demo that you find online!

You may tie the presentation back to AP CSA in one or more of the following ways:

- 1) Contrast the language of your topic to Java in as many ways related to AP CSA as possible. (since this is a unit 2 project, comparing the PLTW unit 2 topics to your application should be included in your discussion. You can go further, and talk about additional course topics In the context of your application)
- 2) Demonstrate how your tool, (library or language can be integrated with Java, or if the tool is a java toolkit such as Swing (for instance), demonstrate in sufficient depth, how to develop using the toolkit.
- 3) If your project contains multiple layers, such as the LAMP stack, or a software design pattern like a model, view, controller design pattern, discuss in reasonable depth your specific technology and problem space and suggest similar ways to solve the problem using Java tools. (If options 1-3 do not fit your specific project, discuss with the instructor other options for tying your toolset to AP CSA)

You will work in groups for this project so once you select a topic/toolset, you should create a design document for the following purposes:

- 1) Brainstorm of at least 3 ideas for a demo application of sufficient effort and complexity for A four week project.
- 2) Create a task list of steps to complete your project. Use the following format:

Task #	Task Title	Task Description,	Task Owner	Estimated time required	Actual time to complete	Status	Notes
--------	------------	-------------------	------------	-------------------------	-------------------------	--------	-------

Status: NS=Not Started, IP=In Progress, OH=On Hold, C=Completed, X=Cancelled

Since this is a four week project, each team member should have tasks that add up to 20-30 hours. Log estimated and actual hours to the half hour.

3) diagram in either wireframe, flowchart, or some other appropriate format an initial design for your demo prior to beginning your coding.

Notes:

- You may use more than one library, but must explain each one used, and why it was necessary
- If your application requires a complete stack (OS, Web/APP Server, DB, Programming language, etc) you will be responsible for setting up, and explaining all parts. (AWS, Repl.it, Local Linux, OSX, or Windows)

Presentation

Prepare a Google slides presentation that discusses your work. Your presentation should include the following:

- 1) Why you chose the toolset that you did
- 2) How did you decide upon the demo application
- 3) What is the architecture of your application - this may include the complete stack if some kind of web app, or it may be a wireframe or some other visual representation of the demo.
- 4) Walk through of task list including estimates and actual time for each task
- 5) Code walkthrough of program
- 6) An actual demo of application as a screencast, with as much discussion from 1-5 above as part of the screencast that you think is necessary.

Documentation/Source Code

All source code for your demo application should be well documented at the class or library level, as well as at the method level, and inline in code where appropriate. Your git repository must include a README.txt or README.md file that fully explains the project, all dependencies,

and the actual application. You must also maintain author and contributors in each class header or library header, as well as a revision date and version number. Try to make sure that there is some correspondence between version numbers and git commit comments.

Possible Topics or Research Resource Sites

Javascript Libraries

React, Angular, View.js, Bootstrap, JQuery, etc (See <https://www.javascripting.com/>)

Java Libraries

<https://dzone.com/articles/20-useful-open-source-libraries-for-java-programme>

<https://jaxenter.com/top-10-java-libraries-137587.html>

Java Frameworks (GUI)

<https://stackoverflow.com/questions/7358775/java-gui-frameworks-what-to-choose-swing-swt-awt-swingx-jgoodies-javafx/>

Java Frameworks (App)

<https://raygun.com/blog/popular-java-frameworks/>

Python Libraries

<https://pythontips.com/2013/07/30/20-python-libraries-you-cant-live-without/>

Python Web Frameworks

<https://stackify.com/python-frameworks/>

Gui Frameworks

Other Languages:

C++, C#, Perl, Ruby, PHP, R,

Articles:

http://www.mnl.com/ourideas/tech/fun_with_programming_languages.php
<https://medium.com/@triconinfotech/top-programming-languages-for-machine-learning-in-2019-48d4640a4f37>

Big Data (General)

<https://blog.eduonix.com/bigdata-and-hadoop/10-popular-open-source-big-data-tools/>

Java Big Data

<https://www.whizlabs.com/blog/big-data-tools-for-java-developers/>

Cool DB tools to look at:

<https://neo4j.com/>

The above list is only meant to be a starting point for your project