Project 4 – Personal Expression Project

Document Revision 1.1

Expectations

- Instructor Guidance in Class – Low
- Independent Effort – High
- Originality – High
- Teamwork – High

Overview

In this project students are expected to work with a peer to create a project that is unique and distinct from all other projects in past semesters or the current semester. This is the only project where students are expected to work closely with a project partner. The personal expression project grade will be based on the solution's functionality, sophistication, innovation, and appearance. While each final personal project grade is determined by the instructor, personal projects will be reviewed / commented on by your peers.

Usually, project partners receive the same grade, but that is not a guarantee. The instructor reserves the right to adjust the project grade of individual students in cases where the contributions to the project are inequitable. To assist and inform the instructor in these matters, students will be asked to evaluate their project partner at the end of the semester.

To be clear, students should have ***no more than one*** project partner.

The goal of Project 4 is to create a working prototype of a web application. It is OK for your application to have some elements that remain "under construction" and/or features that need to be added at a future date.

Your goal is to create enough functionality to convince an audience of your peers and MIS faculty that what you are building is both possible and that you could bring your project to a more complete / refined state if you had the necessary time.

Instructions

Preamble

The world is full of fascinating problems waiting to be solved.

Working in Information Technology is lots of fun, but it's a kind of fun that takes lots of effort. The effort takes motivation. Successful athletes get their motivation from a kind of physical delight in making their bodies perform, in pushing themselves past their own physical limits. Similarly, happy I.T. professionals get a basic thrill from solving problems, sharpening their skills, and exercising their intelligence.

You must develop a kind of faith in your own learning capacity — a belief that even though you may not know all of what you need to solve a problem, if you tackle just a piece of it and learn from that, you'll learn enough to solve the next piece — and so on, until you're done.

Categories

Student projects should fall into one of three categories.

Category	Notes
Generation of Revenue	Goal: Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all. See: https://sdgs.un.org/goals/goal8 Notes: Operating within the bounds of the law, come up with a revenue-generating business. Your imaginary business can be a small side hustle, a start up with big ambitions, or somewhere in between.
Improving Education	Goal: Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all. See: https://sdgs.un.org/goals/goal4 Notes: It would be smart to reflect on your own learning experiences. What works for you as a student? What doesn't? What would motivate you to learn? Try to draw a distinction between *learning* and getting good *grades*. Sadly, the two don't always go together. Education can be at *any level* (grade school, high school, college, vocational training , etc.)
Improving Physical Health	Goal: Ensure healthy lives and promote well-being for all at all ages. See: https://sdgs.un.org/goals/goal3 Notes: Health includes preventative medicine (i.e. diet & exercise), access to medical care, vaccinations, etc. etc. Please limit yourself to the arena of *physical* health. Solutions that claim to improve a patient's mental health by having them spend *even more time* online are not welcome. You might want to consider the needs of a specific group of people that are currently under-served.

Generate a project idea from one of the above categories. The categories are adaptations from the United Nations list of 17 Sustainable Development Goals. The original U.N. list of goals can be found here: https://sdgs.un.org/goals

Technology Requirements

Student work must use ***all*** the technologies presented in MIS3502 (HTML, JavaScript, jQuery, NodeJS/Lambda, MySQL). The solution must use one or more Web Services created by the student team. ***The solution must use one or more APIs provided by a third party***. The solution must be hosted on one or more S3 buckets.

Peer review and presentations

In the last week of the semester, students will "pitch" their idea and prototype to a subset of their peers. Student feedback will be collected by the instructor. The best (peer-selected) student presentations will be invited to present their work to the whole class. Finally, the very best student presenters will be invited to demonstrate their work at the MIS3502 Project Showcase.

Please be advised, if you are not selected to present your work at some point in this process, ***it does not mean that you will receive a failing grade on the project***. The actual grade for your project will be determined by your instructor.

Deliverables

Student teams are responsible for:

- Producing a working prototype
- A simple (but professional) PowerPoint presentation.

The presentation should contain these elements (not necessarily in this order):

1. Introduction – This should be concise. You must quickly convey what your idea is.
2. Demonstration - The PowerPoint should contain a link to the student work. No screenshots are expected. Use screenshots only if some technical limitation makes them necessary.
3. Why is your idea innovative/original?
4. Why is it sophisticated?
5. What APIs did you use?
6. An Architecture Diagram – this diagram should represent your client, your lambda resource(s), your database, your API choices, and suggest how they interact.
7. What are/would your next steps? The focus here is on new features (i.e. I don't care about next steps like "secure funding" or "start a marketing campaign".)

Banned Content

Here are things that I don't want to see.

1. ***Do not attempt*** to construct a financial plan as part of your presentation.
2. **You are welcome to *start with* the class templates**, but your web interface ***should not look like Project 2 or Project 3*** when you are done. The appearance of your solution should be unique, distinct, and professional.
3. Animated gifs and similar "busy" visual content are usually not very professional looking. If you use them at all, then they must be subtle and professional.
4. Project ideas that purport to (somehow) improve mental health.
5. I am sorely tempted to outlaw the use of maps. I have explained this in class. But to reiterate …
    a. ***If I can quickly get the answer to a question using Google maps, then I do not need your new application to answer the same question.***

    b. If you do use a map as your third-party API, it must provide some essential value to your solution.
    c. If you do use a map as your third-party API, it must be dynamic. That is, it must change relative to where the user is, and it must have data points (or "pins") that can change in response to underlying data changes.
    d. Students who use maps in their projects often neglect important questions like:
        i. Where will the data for the map pins come from? What's the process for getting that data?
        ii. How will the map pins be maintained? (And it is not acceptable to make this the job of some developer or technician unless your solution is generating a ton of revenue!)
        iii. What new data is your application collecting? How is it that valuable and/or unique?
6. ***Do not make a static (unchanging) "landing page" the focal point of your application.*** The focus of this class is not on making WordPress ***sites*** or heavily themed web ***sites***. The focus of this class is on making dynamic (changing, data-driven) web ***applications*** using web services.
7. ***Do not expect solutions that are mired in past technologies to be praised as "innovative".*** I do not need help getting my Kodak film developed, or my VHS tapes returned to Blockbuster, or turn-by-turn directions printed from MapQuest, or a way for me to buy/sell/trade textbooks with my classmates.
8. Do not claim to have improved an existing process by ***making it simpler***, unless you can really explain ***how*** you made it simpler. If you really didn't do anything innovative, the I will assume that what you are *really* saying is "we didn't know how to make it any better than this."

Sophistication

Here are the criteria I will use to determine if/your project is sophisticated or not. It is roughly the same list as you saw at the beginning of the semester, with one clarification, and one addition.

Remember ***there is no one magic component of a good, sophisticated, idea.***

Instead, we have "Shafer's Nine" questions. Recall that if you can reasonably answer yes to ***three or more*** of those questions when describing your project idea, then it is ***probably*** a good, sophisticated idea. And, honestly, you shouldn't have to ask your instructor if your idea is good or not. Answering these questions should give you all the confidence you need.

Shafer's Nine questions to think about when evaluating a new IS idea:

1. **Does it require roles?** Does your project involve users with different roles? For example: if you request an Uber ride there are at least two different roles in that process. The roles are *driver* and *passenger*. Users of the Uber system have different duties depending on their role, and different features of the Uber system are exposed to users based on their role.

2. **Does it leverage mobile technology?** Will your application be used primarily on a smartphone, tablet, laptop, or some combination of those things? Applications that run on smartphones can make use of geolocation, cameras, and other features that desktop applications can't. It is usually smart to design mobile user interfaces first and think about the desktop/laptop experience later.

3. **Do you need to model it?** Recall that an Information System can be thought of in terms of people, process, and technology. Does your project idea have enough steps that it inspires the need for a swim lane diagram or a flow chart?

4. **Is it self-perpetuating? And/or does it have a network effect?** Does your project idea support or create a virtuous cycle? (If you don't know what a "virtuous cycle" is you really should Google that term!) Does your project have a network effect – that is each new participant ***dramatically*** improves the value of the solution.

5. **Does the solution require the use of status codes?** Data that appears, disappears and/or seems to move from one location to another based on a status code is often an indicator of sophisticated solution.

6. **Does it collect new data?** If your solution collects data that was previously unavailable or very difficult to obtain, then you are probably on to something good. But be careful, it's not enough to just collect data, you should also be able to summarize it in some way that is either prescriptive, predictive or (at the very least) engaging.

7. **Are APIs used in a series of steps?** For example: a Google map that always shows the exact same location is almost entirely useless. A map that *changes* location and displays information relative to the user's current location can be more useful. To get that effect, you must use API data *in steps*. First, use an API to get the user's geolocation (latitude and longitude). Second, call *another* API to render a map using that geolocation. That's (at least) a two-step process for the developer, but it looks like a single feature to the non-technical user.

8. **Is it disruptive?** A business with a disruptive business model attempts to address consumer demands that the current industry cannot (or will not) provide. Disruptive Business models often employ innovations in IS not used elsewhere. Disruptive systems often bypass entire processes that were constructed around legacy technologies (phone, fax, human clerks, paper forms, etc.).

9. **Does it use gamification well?** Does your application take a difficult or boring chore and make it fun? Remember simply assigning points, and proclaiming “people will do this for fun” does not equal gamification. If you are going to gamify a task or process, then you must make an engaging interface and think about incentivized behavior. Never in my life have I been incentivized by an animated gif.

The next (and final) page contains the project rubric.

Rubric

All grades are assigned at the discretion of the instructor.

Category	Max Points
Functionality - Does your solution work? 0 – Solution crashes or is noticeably incomplete. 10 – Solution gives the illusion of functionality, but it is not effectively using a Web Service and/or a Third-Party API 15 – An effective, functional prototype of an application that uses a student created web service and a Third-Party API. The solution might have a minor bug. 20 – A convincing and polished solution that runs without error.	20
Sophistication 0 – One or fewer of Shafer's Nine criteria were met. 10 – Only two of Shafer's Nine criteria were met. 20 – Three or more of Shafer's Nine criteria were met.	20
Innovation 0 – The solution merely reproduces a well-established, existing, service. Students will often proclaim that "we made it simpler" … if that is true, then you must be able to demonstrate that you did more than simply take away/neglect useful features that exist in competing solutions. You can't convince me that your product is better because you left out a bunch of features that you couldn't build. 15 – The idea is adequate. 20 – The idea is creative, thoughtful, and inspires further discussion.	20
Appearance Only the appearance of the web application itself will be evaluated. (I have not asked you to create a static web site.) 0 – The interface is nothing but a repeat of the class template. 10 – A token effort was made to change the class template. 15 – A thorough effort was made to change the class template. 20 – The appearance is creative, thoughtful, and shows attention to detail.	20
Accessibility I will scan your web application for critical accessibility errors using WAVE. 0 – More than 1 critical error 5 – One critical error 10 – No critical A11y errors	10
HTML I will check your web application's HTML using the Firefox View Page Source feature. 0 – More than 1 syntax error 5 – One HTML syntax error 10 – No critical HTML syntax errors	10
Total points	100