Two potential projects

2)Planting Guide

--Thought: This is fun and all people who might want to grow something can relate. Genesis was about planning for blooms on flowers, but it was easier to find API about harvesting fruits/vegetables. I like the API idea.

Scope/Description/Purpose:

* People have some entry forms including a date field and a days tolerance field
* Python would calculate days from today to that date, and using API data which gives median time to harvest would return the crops that would be ready for harvest in the indicated time frame
* Enhancements include storing for the user…after having them register…taking location into account…I found historical weather APIs, but so far nothing on crops per location.

Wes: might be worthwhile to keep looking for APIs, but okay scope

Wes: does not need to be related to field…just show polished project…focus one aspect of project that make shine…front end….or something else…what do you want to learn…first project..external APIs is good…

IT could take two weeks and half projects are not impressive

1)IT web pages:

--A friend in IT said this was a real thing and would look interesting to employers, but I kind of wonder if departments with rogue web sites would even respond, and IT knowing could make them more culpable

Scope/Description/Purpose:

* IT departments of large corporations need to ensure they have good records of their web apps for reasons including security, and compliance.
* In many cases departments outside of IT have authority to create and modify their own web apps
* This program provides a simple and effective mechanism for IT heads to poll/survey various departments by sending an email with a link.
* Data
  + Users table will hold the various users of the app including viewers only, respondents, and IT who may modify associated/related data
  + website attributes, general type
    - those entered by respondents
    - fields updated by IT based on their view of the response and further processing such as marking the site as reviewed, approved, or such
  + list of expected or possible fields
    - Users:  standard such as id, (which depending on the org is all needed to get most of other fields), fname, lname, email, dept, is\_web\_owner, etc
    - website related
      * Type backend database (null for none)
      * payment info types (one for many if we want to allow multipe) such as PayPall, Cash, Venmo, etc
      * PCI (Payment Card Industry) compliance. It's just pass or fail, ,but there are levels of compliance for banks/issuers versus just merchants
      * Server OS
      * Contains PII (HR Data is a good example)
      * Name of app
      * Purpose of app
      * Level of criticality of app (business impact if down)
      * Dept owner
      * Dept rep who entered data/responded to survey
      * Hosting service (AWS, etc)
    - Other fields by IT or such
      * survey status...acccepted, returned, rejected, etc This 'survey'?
      * Audit status..accepted, rejected, etc.
      * Security approval status? If applicable, often InfoSec doesn't even know about some of these sites out there. A formal policy would likely have a Security Acceptance or approval process though.
      * Last vulnerability scan score and date A vulnerability scan is an assessment of the server and the application and whether or not it is susceptible to know vulnerabilities which may be exploited to hack either server os or app. Virtually all hacks come from exploiting a known vulnerability.
      * Maybe some historical fields… Date launched. End date? (Marketing may use a site for an event like the Super Bowl and then kill it)
      * Uptime support level appropriate for business criticality Could ask the respondents to provide a criticality rating (1-10) but they'd always say 10.