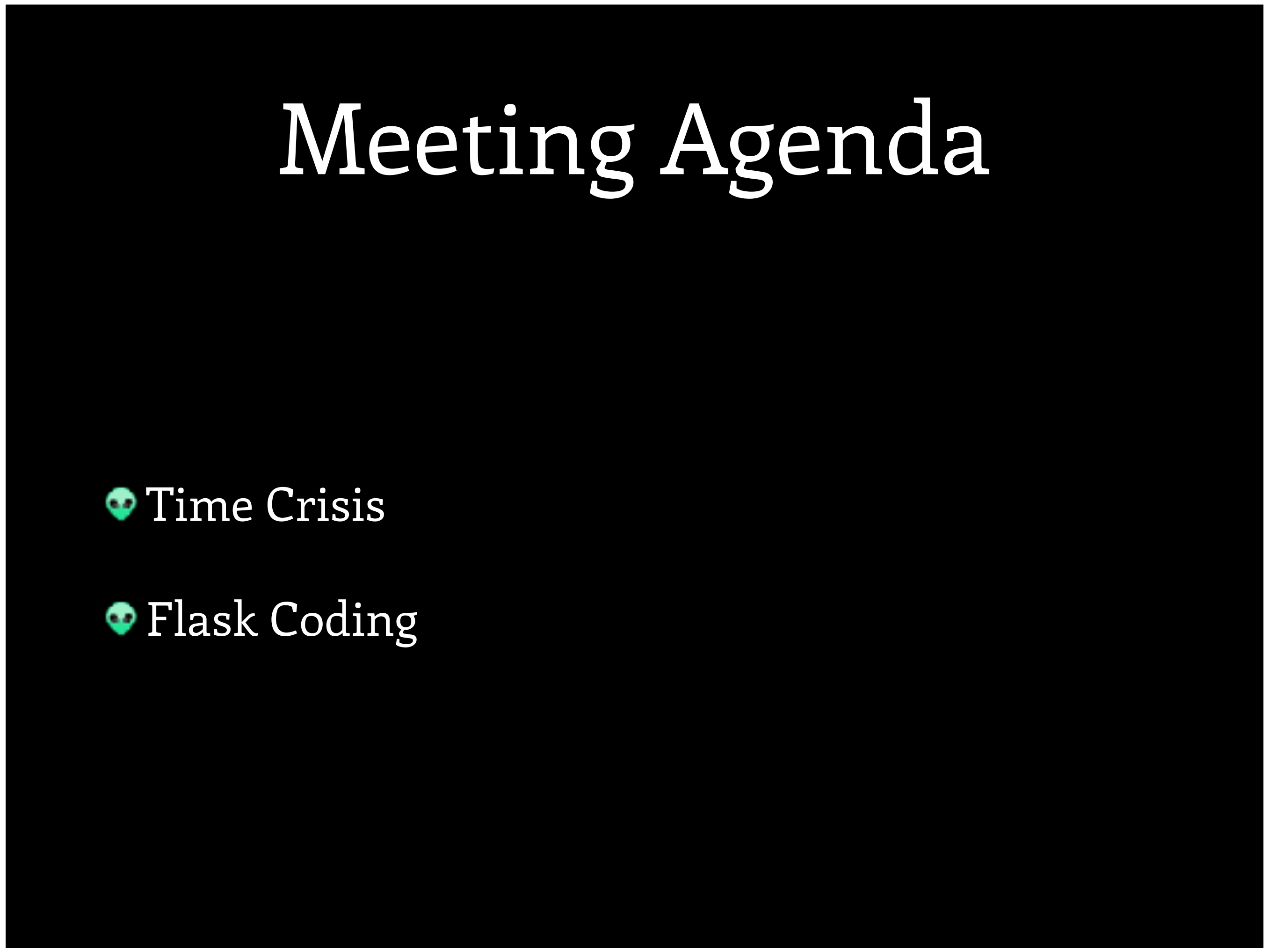
OFFICE OF STRATEGIC NATIONAL ALIEN PLANNING

Working Session

Jan 26



Meeting Agenda

- 👽 Time Crisis
- 👽 Flask Coding

We’re up against some time issues on the instructional side as well…

A Couple Dailies…

- I did not complete any additional work on the assignment. For today, I have developed a plan for how I am going to attack this project and I have a few getting started questions lined up for office hours tomorrow
- Also… I just want to double check. The database that we were creating for assignment 2, will that be used in the final term project or will a more complete database be found? Or, if we are to use it, will the customer be able to supply us with updated records?

The assignment 2 scores were much better than I had expected given the pain I'd been hearing. Well done all.

A couple other notes:
Developing a plan of attack is work on the assignment… It might feel less fulfilling since it isn't visibly part of the tangible deliverable, but it is critically important to efficiently implementing a solution.

A more complete database is hoped for when we get to the final term project… Making that happen depends on getting some extra time on the instructional side to generate the data. I would like to provide that data as well structured and normalized data files, so that you are all able to independently make changes to the database schema as you uncover new requirements for implementing LOST. The alternative would be a DB dump, but that forces us to all use the exact same schema for the project.

We need your help…

- Andy and I are students too
- We want our hours to go to you… not exceptions grading
  - Directory structure
  - Configurable Configuration
  - Code should run, even if the execution is incomplete

The gap week is good for you and good for the instructional team. Also, the due date for assignment 3 has been pushed a week. We want to be available and here to contribute to your success. There are about 30 of you versus 2 of us though. My ideal team size is 8, so we are at 4x of the management problem I like to solve and I have half the number of hours to solve it in, plus I have a bunch of instructional design work to do.

We are way over hours on the instructional side… Andy crossed over 20 hours for the week on Wednesday and I'm not even going to talk about my hours. I will pick up Andy's Friday office hours. I'm hoping this time problem will naturally correct itself; by the end of assignment 3 you will have demonstrated that you can use the core technologies for the project. The help hopefully will move rapidly from individual tactical technical debugging to design.

One of the big potential time sinks, that will just get worse as the project moves forward will be the grading. Grading this week was 2x our estimate; the time we spend grading is lost to other activities like office hours, piazza, and improving the instructional material. There are some things you can do that would help us out a lot in grading.

The directory structure and specified file names need to match. Looking through the directories and trying to guess which file and how to call the file is a huge time sink for grading. I'm not ready to make this defect class result in failing an assignment yet, but lets try to get our source trees structured the same. If you work on a project with a team, everyone needs to agree on where the files are going to be.

If the assignment spec calls for the script to take in configuration parameters those need to be respected. Overriding with hardcoded values can cause some trouble. For example in assignment 2 the createdb was to happen outside of your code, failure to correctly use the argument for the db name requires a manual step to correct. Relative rather than fully qualified paths should also be used to help increase portability… basically, if a fully qualified hardcoded path is in your code, something has gone wrong. I'm also not ready to fail assignments for this yet, but that position will likely change rapidly. If you work on a project that someone else has to deploy, expect their environment to be different than yours. Code defensively for that.

We want to give partial credit for partial solutions. A partial solution should still run. There are some solutions that fail to run but look right and appear close to working (syntax errors and typos in keywords). We all have a copy of the same environment for execution, everyone should be able to test their submissions. Debugging your code should be out of our scope.

I need the last one fixed immediately… You are not a programmer if your code doesn't run and your not a good programmer if you don't test your work during development. Wether the code is complete/correct is a different mater; the code in your master branch should always run. Broken code can and often should be checked in, but that code belongs in a different branch.

"Don't break the build!" is the number one rule when working with others using a shared repository.

Management Lobotomy

- Technical skills are hard and take a lot of work to keep up
- Management is time consuming and distracting
- Grow myself vs my team
- Your peers will always be your best technical resource

I've been thinking about the feedback that we spend a lot of time in lecture introspecting on the shape of the course and management practice. Why am I doing that?

Early on in the course, there was agreement that running the course like a project was desirable. I've never had a project meeting where we talked in depth about technology, that was always done in breakouts or vender presentations. There are a ton of resources (documentation, tutorial, etc) out there for learning technical tools. Meetings add value by communicating vision and synchronizing the understanding of what we are all working toward.

Academia is a strange place… this is the last time in your life where it is fair to expect your management to be an expert in what you are being asked to do. Forever after, you will be the expert. Management hired you because they can't solve the problem; they lack the skill or they lack the time.

The assent to management requires letting go of the deep technical skills. There's not time to focus. Tracking progress, selling work, putting out fires, estimating timelines, documenting/organizing information, and endless meetings is a ton of work. Getting time to really learn a new technology or even write a little bit of code can be nearly impossible.

If your management is smart, they will be able to keep up with you as you describe problems and suggest general solutions. You may get a former technical resource as a manager who can sit with you; which can be good but that kind of manager usually come with other challenges. If you have deep technical problems that you can't solve management is likely suggest peers for you to confer with or hire a consultant to solve the problem.

With all that said, the rest of lecture is probably going to be live coding with Flask… Which is going to be extra fun because I don't really have any Flask experience. From the dailies, I suspect at least some of you understand Flask a lot better than I do already.

Who's on which platform? Who has a strategy they are happy with for editing files? Who has had success connecting to the database? Connecting to the web server? Your peers are the experts.

Web Tech is Messy

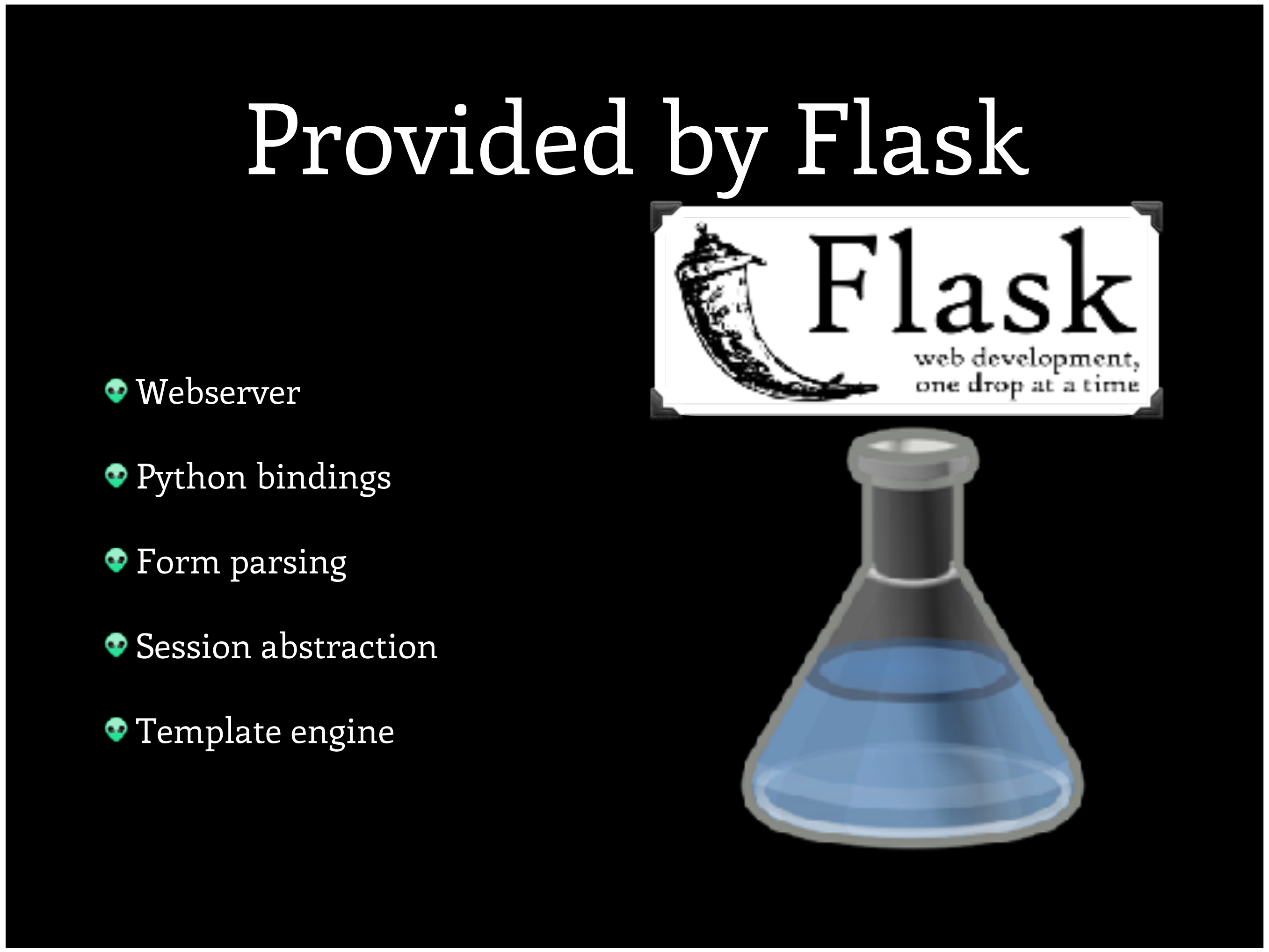
- Recall that network problems are thought of in layers
- Recall that web applications are layers of hacks
    - Different hacks use different languages
    - Web application architectures are convoluted

Flask
web development, one drop at a time

https://pl.wikipedia.org/wiki/Plik:Piersi%C3%B3wka_ubt.jpeg

Even though I’ve made a good amount of money doing web development, I passionately hate web applications. The inconsistencies and random ways the technologies are hacked on top of one another grate against my design sensibilities.

If you feel that web is confusing and terrible, I understand and agree… and will add that from my perspective it has only ever gotten worse. The only consolation I can provide is that the complexity is tractable.


Flask
web development, one drop at a time

Provided by Flask

- Webserver
- Python bindings
- Form parsing
- Session abstraction
- Template engine

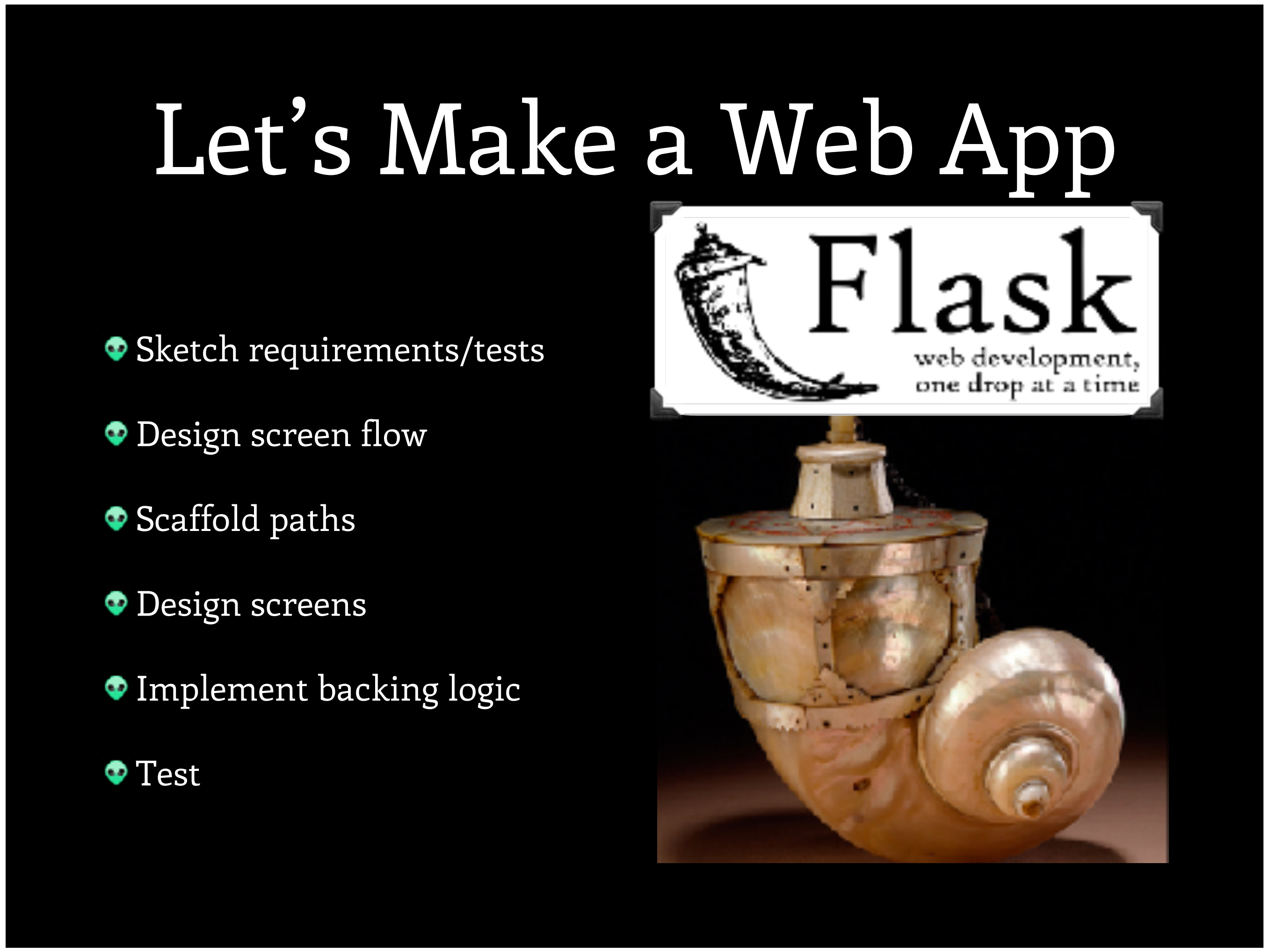
Flask has a built in web server, this is the part of the stack that connects to the network port, reads requests, and writes responses in the correct format.

Unlike a regular web server that selects and servers content from the filesystem, Flask uses python to select and serve content. In Flask we will use decorators to map from the path part of a URL to a function to be run.

Forms are one of the early hacks onto HTML… which was backed by hacks on the server… which relied on the way UNIX-like OSes manage the environment processes run in… I wrote some web applications in these bad old days, they were dark times. Hopefully you will never need to know how dark. Living in the future, most of that complexity will be handled by any web framework.

Sessions are an affront to the statelessness of the web. In the dark ages of the web, we embedded nonces in URLs and hidden web page form fields to help flag requests as related. These were terrible solutions.
Eventually cookies became widely supported, they are also a terrible solution but are so much better than the older style. Cookies should also be nonces, but they are handled automatically by the browser.
On the server side, the nonce can be used to check persistent storage for information about previous requests. Coding all this yourself is a lot of work that Flask helps to take care of for you.

Mixing a programming language and markup language is like a peanut butter cup. Do I embed the programming language in the markup language (e.g. PHP, python server pages) or do I write a program that outputs markup (e.g. old fashioned CGI scripts, J2EE servlets).
A lot of modern frameworks don't really make us choose. The program could do computation including outputting markup directly or make a call directly to a templating engine that executes code in the template or anything in the spectrum between these extremes.



Let’s Make a Web App

- Sketch requirements/tests
- Design screen flow
- Scaffold paths
- Design screens
- Implement backing logic
- Test

The first challenge is determining what we are trying to build. What we want sets the project requirements. Like our goals, each of our requirements should be testable and we should at least think about how we might test the requirement.

Most applications are made of screens that the user interacts with. Our web app is no different. What are the screens we think we need, how are they related.

I would scaffold the paths at this point. This will start to setup the functions that will be needed.

From here, it alternating between implementing a screen and checking that the screen does what I want. Eventually all the screens will be finish and I’ll have some reason to think they work due to the checks I’ve been doing during development.

After everything is suspected to be working, I need go back through and check that I didn’t break anything along the way. I know I’m done when I can pass the tests that match with my requirements.