

# HW11: Technical Prototype

Warm up due Wednesday at 4pm (grace period until 11:59pm)

Main due Monday at 4pm on Courseworks (no grace period – feedback will be given in class)

Note: if your TA feedback meetings are Tuesdays, then you may have slightly different deadlines. Please turn in assignments according to instructions given by your TA.

## Warm-up:

1. **Group.** Which prototype will you continue to iterate on for the rest of the semester?  
Note: You must get this approved by your TA.
2. **Group.** Do any final iterations on your prototype that you need to before you start implementing it. Make sure you have all the media/data you need to complete it.
  - Include a PDF of the final Google Slides
3. **Group.** Set up a github repo for your project. Include all your group members and your TA.
  - Submit a screenshot of the name and members of the repo.
4. **Group.** Job assignments. List the roles you plan to have and the names of the people assigned to them. Since you have less than 6 people in your group some people will have to be named twice. See below.

In every software project, dividing work into individual responsibilities is challenging. However, it is extremely important that everyone contribute to the code base every week. Thus, every week, each team member has to decide on what their contribution will be. For this week, we strongly suggest the assigning different people to the two main parts of the app:

1. Part 1. Learning portion (including the home page).
2. Part 2. Quiz portion of the app (including the quiz end page)

Within both of these two parts of the app, there are three main tasks

1. Part 1. Learning portion (including the home page). Jobs:
  - a. Architecting the data
  - b. Implementing the UI
  - c. Testing that you can click through the app. (this cannot be done by the learning UI implementer)
2. Part 2. Quiz portion of the app (including the quiz end page) Jobs:
  - a. Architecting the data
  - b. Implementing the UI
  - c. Testing that you can click through the app. (this cannot be done by the quiz UI implementer)

Although you don't have to use these exact roles, you do have to establish roles and have everybody assigned to at least one role.

What to turn in:

- A PDF with the answers to #1, #3, and #4
- The PDF with the slides to #2.

Although this work is all being done collectively, everyone in the group must turn in the work individually. (You may turn in identical work)

Note: You may not turn in a link of a google Slide – because then you could edit it after turning it.

## Main Assignment

This is the first of three weeks where you will work on implementation and testing. The goal of this week is to have something functional where the user can at least click through each state of the app – including the learning portion and the quiz. For HW 11, you will focus on the interactivity and usability, and HW12 will focus on graphic design. You will then have another week to clean stuff up before the final project is due.

### Requirements:

1. Everyone in the group must check something in to the git repo
2. You must have a Flask back end and a HTML/JS/JQuery/Bootstrap front end.
3. You must have a home screen with some kind of “start” button so you know when a new user has started the learning process.
4. On the back end, you must store important information about users’ choices on every page. For the quiz, you should store their quiz answers. For the learning activity, you should store important selections they make (or at least what time they enter the page).
5. You can’t hard-code the design of the pages into the HTML, you must represent the data in a JSON object, then render each page with the correct data. This is sort of like having pages of IMDB being generated dynamically from a template. You don’t necessarily have to use Jinja (all those curly braces). I personally don’t use Jinja – I just use JQuery to render everything from the data object in the `$(document).ready()` function.
6. Although your implementation may vary, you will probably need ~4 routes.
  - a. A home page (with a start button)
  - b. A learning route that takes a variable for what number lesson the user is on.
    - i. E.g. `/learn/1`
  - c. A quiz route that takes a variable for what number quiz question the user is on.
    - i. E.g. `/quiz/1`
  - d. You might also have a quiz end page.
7. Each page should show some data, have a few instructions, record some user data, and at least be able to go to advance to the next page.
  - a. You should be able to enter your quiz information and receive a score at the end that reflects your correct/incorrect answers.
  - b. You do not have to implement interactive components – If your prototype uses drag and drop, you don’t need to implement that yet, or if it has an image of a piano that you can press on to hear intervals, you can save that for next time.
8. Your app does not need to work for more than one person at a time. You can assume you only have one user at a time who ever uses the app. In real life, you wouldn’t do this – you’d use a package that implements user accounts and then store all the user data on the user object.
9. During your TA feedback session, every team member needs to have the app running on their laptop.

What to turn in:

- **Group.** A list of the responsibilities of each group member. It's okay if this deviated from what you planned in the warm up.
- **Group.** A short (~1 min video) of you clicking through the prototype and reaching every screen. Please provide a YouTube Link.
- **Individual.** Describe what you did for the project this week. A short paragraph or bullet points is fine. Show images if they help.
  - Show a screen shot of a commit you made to the github repo.