Time Matters

Week 7
<table>
<thead>
<tr>
<th>Week 6</th>
<th>Prototyping + Needfinding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 7</td>
<td></td>
</tr>
<tr>
<td>Week 8</td>
<td>Implementation</td>
</tr>
<tr>
<td>Week 9</td>
<td></td>
</tr>
<tr>
<td>Week 10</td>
<td>User Testing + Debugging + Final Prep</td>
</tr>
</tbody>
</table>
User Experience
<table>
<thead>
<tr>
<th>Date</th>
<th>Lecture</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue May 17</td>
<td>Design Critique: UX Design (Groups A-H)</td>
<td>8</td>
</tr>
<tr>
<td>Thu May 19</td>
<td>Design Critique: UX Design (Groups IN-W)</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

[http://cogs121.ucsd.edu/schedule/](http://cogs121.ucsd.edu/schedule/)
UX: Interaction Cost

- **Interaction cost**: any efforts required for users to attain their goal
- The goal is to **minimize** following interaction costs:
  - Reading
  - Scrolling
  - Looking for relevant information
  - Clicking, touching, typing (and **pogo sticking**)  
  - Page load and waiting times
  - Attention switches
  - Memory load

- [https://www.nngroup.com/articles/interaction-cost-definition/](https://www.nngroup.com/articles/interaction-cost-definition/)
Rules of Thumb

● Minimize Visual Path (mini-IA)
  ○ Users do not need to traverse entire page to reach their goal

● Prioritize visual components
  ○ Usage of font size, colors, and placement

● There are no universal icons
  ○ The symbolism of icons is based on user’s previous experience
  ○ Icons need texts as labels

● The navigation organization matters
  ○ Related components should be grouped and these groups are distinguishable
  ○ Keep navigation design consistent
UX: Golden Triangle
UX: F-Shape

https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/
UX: Response Time Limits

- How long can users wait or search?
  - **0.1 s**: “the limit for having the user feel that the system is reacting instantaneously”
  - **1.0 s**: “the limit for the user’s flow of thought to stay uninterrupted”
  - **10 s**: “the limit for keeping the user’s attention focused on the dialogue”

- [https://www.nngroup.com/articles/response-times-3-important-limits/](https://www.nngroup.com/articles/response-times-3-important-limits/)
Choosing Tools
Project Report - System Development

• Architecture: explain the general idea of the system such as client/server, inputs, outputs, sensors, information flow.
  • Add at least one image

• Technology used: describe what APIs did you use and what other web services or data sources you have been integrating
Performance

● In technical perspective, how can we improve time for better UX?

● Technical considerations:
  ○ **Network Travel**: HTTP request/response can affect the delays in network
    ■ **External dependencies**: using libraries referring to other server can create delays
  ○ **Parsing**: Browser/interpreters may read/parse your code character by character
    ■ Reading each character can take certain amount of time
    ■ **Multimedia**: images/videos tend to take more time to parse than text

● How can you minimize time with our technical knowledge?
Bottleneck

- 5 **bottlenecks** of web application
  1. Too many third-party widgets, ads and analytics tools.
  2. Too much server-side processing.
  3. DNS is taking too long.
  4. Failure to compress, concatenates, and combine.
  5. Poor choice of hosting.
Third-Party Tools

- Third-party tools (e.g. node_modules) can facilitate development process
- However, it is **highly unlikely** for you to use every feature of a tool.
- Each tool takes up time and space
  - Some tools might be slower and larger than your own application
  - Unnecessary codes that will kill your performance
- **Conclusion:**
  - Choose your tools wisely
  - Know the specs of your app
Depending on your code density, it can affect your performance.

<table>
<thead>
<tr>
<th></th>
<th>Client-Side</th>
<th>Server-Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Faster</td>
<td>Slower</td>
</tr>
<tr>
<td>Security</td>
<td>Less secure</td>
<td>More secure</td>
</tr>
<tr>
<td>Others</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Conclusion:**
- Simple algorithms → client-side
- Sensitive information → server-side
- And more ...
Compress + Concatenation

- Browser parses HTML, CSS, and Javascript code every time a user opens the your application.
  - Suppose parse time is 45 seconds.
  - 100 users opening application → 4500 s → 75 minutes
- Parsing
  - Program reading strings (by characters)
  - Unnecessary spaces and new lines will take extra time.
- In production level, your CSS and client-side JS should be:
  - Combined into minimal number of files (preferably one)
  - Minified or compressed (remove spaces and new lines)
Compress + Concatenation

- There are tools for compression:
  - https://github.com/fmarcia/UglifyCSS
  - https://github.com/mishoo/UglifyJS

- But we recommend one of the following:
  - Grunt
  - Gulp

- Grunt and gulp are task managers for your app:
  - Any steps needed before production, we can assign it as a task.

- Example of compressed JS: d3.v3.min.js
Questions?