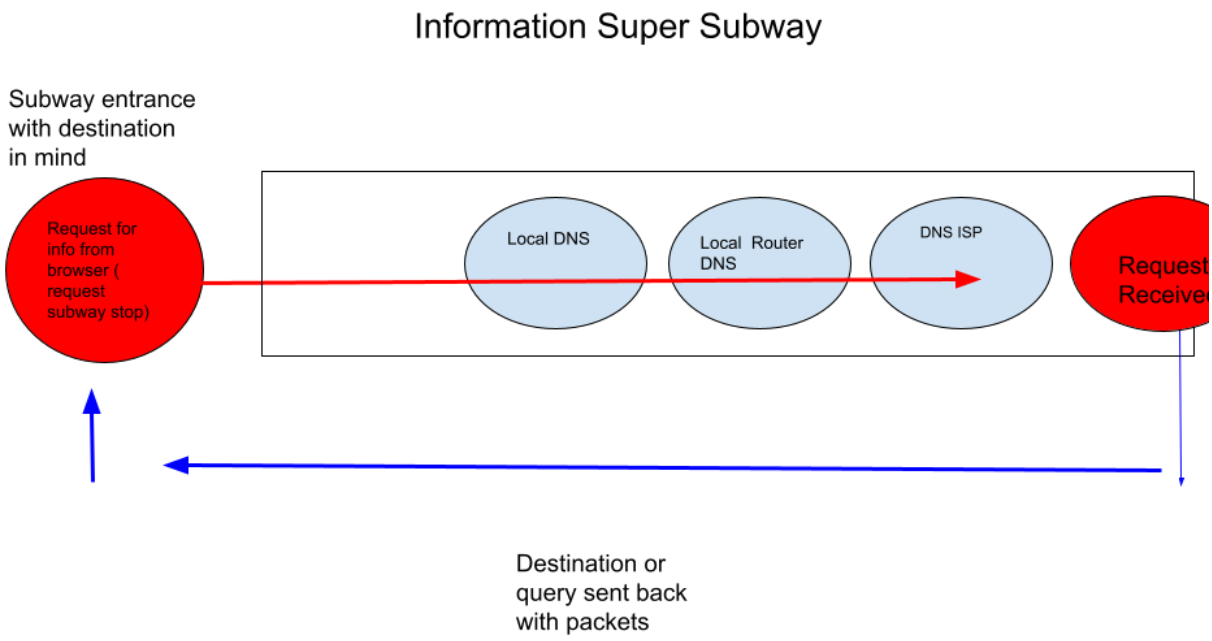
How the Web Works

In this lab, you'll be working with a partner to explore a little more about the internet, the web, requests, responses and more. You'll be reading and writing about concepts as well as practicing some of the commands that we saw during the lecture earlier.

Topic 1: The Internet and the World Wide Web

1. What is the internet? (hint: here) - The internet is a worldwide network of networks that uses the internet protocol suite(TCP/IP)
2. What is the world wide web? (hint: here) - an interconnected system of webpages accessible through the internet. The world wide web is one of many applications built on top of internet
3. Partner One: read this page on how the internet works, Partner Two: read this page on how the world wide web works. When you're done reading, come back together and and answer the following questions
    a. What are networks? Two or more computers that can intercommunicate
    b. What are servers? - Servers are computers that store web pages, sites, or apps. When a client device wants to access a webpage, a copy of the webpage is downloaded from the server onto the client machine to be viewed.
    c. What are routers? - Relays messages from one computer to another.
    d. What are packets? - When data is sent across the web, it is sent in thousands of small chunks called packets. If it was sent in only one big chunk, only one person would be able to visit a website at a time.
4. Come up with a metaphor for the internet and the web, you can do a single one if you think of one that puts them together or two separate ones (feel free to use one you've heard today or read about if you can't think of a new one, but spend at least 10 minutes trying to think of something different before you resort to that) - **Information super subway.**
5. Draw out a diagram of the infrastructure of the internet and how a request and response travel using your metaphor (like the map and letters we saw during the lecture). Insert the drawing into this document (can be a picture of a physical drawing, a Google Drawing, a Figma drawing, etc)



Topic 2: IP Addresses and Domains

1. What is the difference between an IP address and a domain name? - The IP address is a set of numerical instructions. It communicates information that is useful to computers. The domain name functions as a link to an IP address.
2. What's devmountain.com's IP address? (Hint: use 'ping' in the terminal) - 104.22.12.35
3. Try to access devmountain.com by its IP address. It shouldn't work because we have our sites protected by a service called CloudFlare. Why might it be important to not let users access your site directly at the IP address? - because some sites use shared hosting and have the same IP address. If you use the IP address it could confuse the server if there are multiple sites with the same IP address.
4. How do our browsers know the IP address of a website when we type in its domain name? (If you need a refresher, go read this comic linked in the handout from this lecture) - It uses a DNS.

Topic 3: How a web page loads into a browser

The steps of how a web page is requested and sent are in the table below. However, **they are out of order**. Unscramble them and explain your thinking/reasoning in the second two columns of the table.

Steps Scrambled	Steps in Correct Order	Why did you put this step in this position?
Example: Here is an example step	*Here is an example step*	*- I put this step first because ____* *- I put this step before/after ____ because ____*
Request reaches app server	2	When the server receives the request it sends the code to be processed
HTML processing finishes	4	Before #5 since the HTML is code that needs to finish being executed
App code finishes execution	5	Before #6 since the code needs to finish being executed before it can be rendered in the browser
Initial request (link clicked, URL visited)	1	Clicking the link is the first thing you do
Page rendered in browser	6	The last thing because the page cant be rendered until the code has been processed
Browser receives HTML, begins processing	3	Its before #4 because the HTML can't finish processing if it hasn't been received

Nmp iTopic 4: Requests and Responses

Setup

- Download the folder for this exercise from Frodo.
- Make sure you unzip it.
- Open it in VS Code
- Run `npm i` in the terminal (make sure you're in the web-works folder you just downloaded).
    - You'll know it was successful if you see a node_modules folder in the web-works folder.
- Run `node server.js` in the terminal (also in the web-works folder) and you should see a log to the terminal saying 'serving up port 4500'
- You'll be using this file to figure out what will happen when you make requests to this server, so read it over to see what's going on. We'll be getting into the two GET functions and the POST function.

Part A: GET /

- You'll start by looking at the function that runs when we make a get request to /, which looks like this: http://localhost:4500 or http://localhost:4500/
- You'll use the curl command to make a request and read the response in your terminal

1. Predict what you'll see as the body of the response: It will display the date and content entries
2. Predict what the content-type of the response will be: HTML

- Open a terminal window and run `curl -i http://localhost:4500`

1. Were you correct about the body? If yes, how/why did you make your prediction? If not, what was it and why? No. I assumed that it would display the entries but it did not. I didn't notice that you need to enter the "/entries" to display that
2. Were you correct about the content-type of the response? If yes, how/why did you make your prediction? If not, what was it and why? - It was text! All the javascript there seemed to point to it displaying text.

Part B: GET /entries

- Now look at the next function, one that runs on get requests to /entries.
- You'll use the curl command again. This time, you'll need to figure out how to modify it to get the response that you need.

1. Predict what you'll see as the body of the response: similar to what we saw before, but the object entries will be displayed.
2. Predict what the content-type of the response will be:

- In your terminal, run a curl command to get request this server for /entries

1. Were you correct about the body? If yes, how/why did you make your prediction? If not, what was it and why? - yes. Because you can look at the entries object.
2. Were you correct about the content-type of the response? If yes, how/why did you make your prediction? If not, what was it and why? - Wrong. It changed to application/json from the HTML that it was before.

Part C: POST /entry

- Last, read over the function that runs a post request.

1. At a base level, what is this function doing? (There are four parts to this)
    a. Adding a new entry to /entries
    b. Increasing the globalId
    c. Returns date and content values
    d. Returns 200 status
2. To get this function to work, we need to send a body object with our request. Looking at the function in server.js, what properties do you know you'll need to include on that body object? And what data types will they be (hint: look at the objects in the entries array)?
    a. Date
    b. Content
3. Plan the object that you'll send with your request. Remember that it needs to be written as a JSON object inside strings. JSON objects properties/keys and values need to be in **double quotes** and separated by commas.
4. What URL will you be making this request to?
    a. http://localhost:4500/entry
5. Predict what you'll see as the body of the response:
    a. It will look similar to the entries in /entries, but it will have the values that we input
6. Predict what the content-type of the response will be: application/json

- In your terminal, enter the curl command to make this request. It should look something like the example below, with" the information you decided on in steps 3 and 4 instead of the ALL CAPS WORDS.
    - curl -i -X POST -H 'Content-type: application/json' -d JSONOBJECT URL

1. Were you correct about the body? If yes, how/why did you make your prediction? If not, what was it and why? Yes. Because it says in the entry section that it will display the date and content. Just like the entries

2. Were you correct about the content-type of the response? If yes, how/why did you make your prediction? If not, what was it and why? Yes. Also because of its similarities to the entries section.

Submission

1. Save this document as a PDF
2. Go to Github and create a new repository. (Click the little + in the upper right hand corner.)
3. Name your repository “web-works” (or something like that).
4. Click “uploading an existing file” under the “Quick setup heading”.
5. Choose your web works PDF document to upload.
6. Add “commit message” under the heading “Commit changes”. A good commit message would be something like “Adding web works problems.”
7. Click commit changes.

Further Study: More curl

Visit this link and do the exercises using the website provided. Keep track of the commands you used in this document. (Don’t forget to resubmit to GitHub when you complete this section)