

[1. General Guidelines 4](#_Toc4579482)

[1.1 General Expectations 4](#_Toc4579483)

[1.2 Timebox 4](#_Toc4579484)

[1.3 Timeline 4](#_Toc4579485)

[1.4 Sending Your Completed Response 4](#_Toc4579486)

[1.5 Language/Environment 5](#_Toc4579487)

[1.6 Online Resources 5](#_Toc4579488)

[1.7 Asking For Help 5](#_Toc4579489)

[2. Requests & Debugging 6](#_Toc4579490)

[2.1 Request Lifecycle 6](#_Toc4579492)

[2.2 Debugging: Page Doesn’t Load 6](#_Toc4579493)

[3. Development 7](#_Toc4579494)

[3.1 Data Types 7](#_Toc4579496)

[3.2 Recursion 7](#_Toc4579497)

[3.3 Code Quality 8](#_Toc4579498)

[3.4 Code Structure 8](#_Toc4579499)

[4. Process 9](#_Toc4579500)

[4.1 Starting a Project 9](#_Toc4579501)

[4.2 Software Development Methodologies 9](#_Toc4579502)

[4.3 Testing 10](#_Toc4579503)

[5. Past Projects 11](#_Toc4579504)

[5.1 Wrong Approach 11](#_Toc4579505)

[5.2 Completed Project 11](#_Toc4579506)

[6. Work Style 12](#_Toc4579507)

[6.1 Work Environment 12](#_Toc4579509)

[6.2 Existing Systems 12](#_Toc4579510)

[7. Additional Comments 13](#_Toc4579511)

[7.1 Is there anything else you think we should know or you’d like to share? 13](#_Toc4579513)

[8. Coding Samples 14](#_Toc4579514)

[8.1 Custom Sorting 14](#_Toc4579516)

[8.2 Data Validation 15](#_Toc4579517)

[8.3 Regular Expressions 16](#_Toc4579518)

[8.4 Simple Web Form 17](#_Toc4579519)

1. General Guidelines
   1. General Expectations

We consider this to be a “full stack” developer test to help us get to know you, your skillsets, and your experience level as well as possible. We hire team members ranging from entry-level to seasoned developers across various areas of the stack, so our goal with this process is simply to determine if we’re a good fit to work together.

You may be comfortable answering all of the questions and if so that’s great! If you feel like some are just not “your thing” that’s fine too. Not every role on our team needs to know *everything* on this test, so just do your best and focus on showing us who *you* are!

* 1. Timebox

This test is intended to be something you can complete in approximately 4 hours depending on skill level. Everyone works at a different pace, so it’s completely acceptable if you want to spend more time, but please don’t spend all day on it.

* 1. Timeline

Generally, we expect to receive your completed response for this test within a week of us sending it to you. If you feel you need more time or something comes up, please let us know.

Once our team receives your completed test, we will generally review and respond within two weeks. At that point, we will either schedule an interview (via phone or in-person) or let you know that we have determined we aren’t the best fit for each other.

* 1. Sending Your Completed Response

For essay questions, please type your written response in the space provided. Each question has at least a half-page allotted for consistency. This simplifies formatting but is not an expectation of how much space each individual question should or shouldn’t take, please take as much or little space as you need.

For the questions that require code as part of the response, please send your completed code as a separate code file and not embedded in the Word document. Hosting on a public GitHub repo is also acceptable. Please be sure to note either the filename or the URL in your written answer to the question.

If you are sending completed code through email and not via a URL for a public repository, please:

1. **Do not attach zip files** as they are blocked by our email filters.
2. **Append “.txt” to the filename** of each code file before attaching to the email. (E.g., “validation.js” becomes “validation.js.txt” so it isn’t filtered by our email rules.)
3. **List the filenames of all attachments** so we can verify nothing was blocked by our email filters.

Name your completed Word document in the format of:

**MetroNet Developer Test - Your Name (YYYY-mm-dd)**

Name any file attachments (for code samples) in the format of:

**MetroNet Developer Test - Your Name (YYYY-mm-dd) - Question Name**

(If a question requires multiple files, please name them accordingly.)

Unless otherwise specified, you may simply reply to the email in which we sent you the test and attach the file containing your completed response.

* 1. Language/Environment

We believe that good developers can learn multiple programming languages, so our goal is to test your general ability to code and reason through problems, not quiz you on a specific syntax or library. Unless otherwise specified, feel free to use whatever programming language you feel most comfortable with to complete this test.

* 1. Online Resources

We believe that searching for answers in Google, Stack Overflow, etc. is part of a developer’s daily workflow and knowing where to find an answer is often as important if not more important than knowing it.

If you use any online resources to help you complete an answer please indicate which ones and a short description of your thought process, including any information you feel relevant such as search terms, what syntax you wanted to verify, etc.

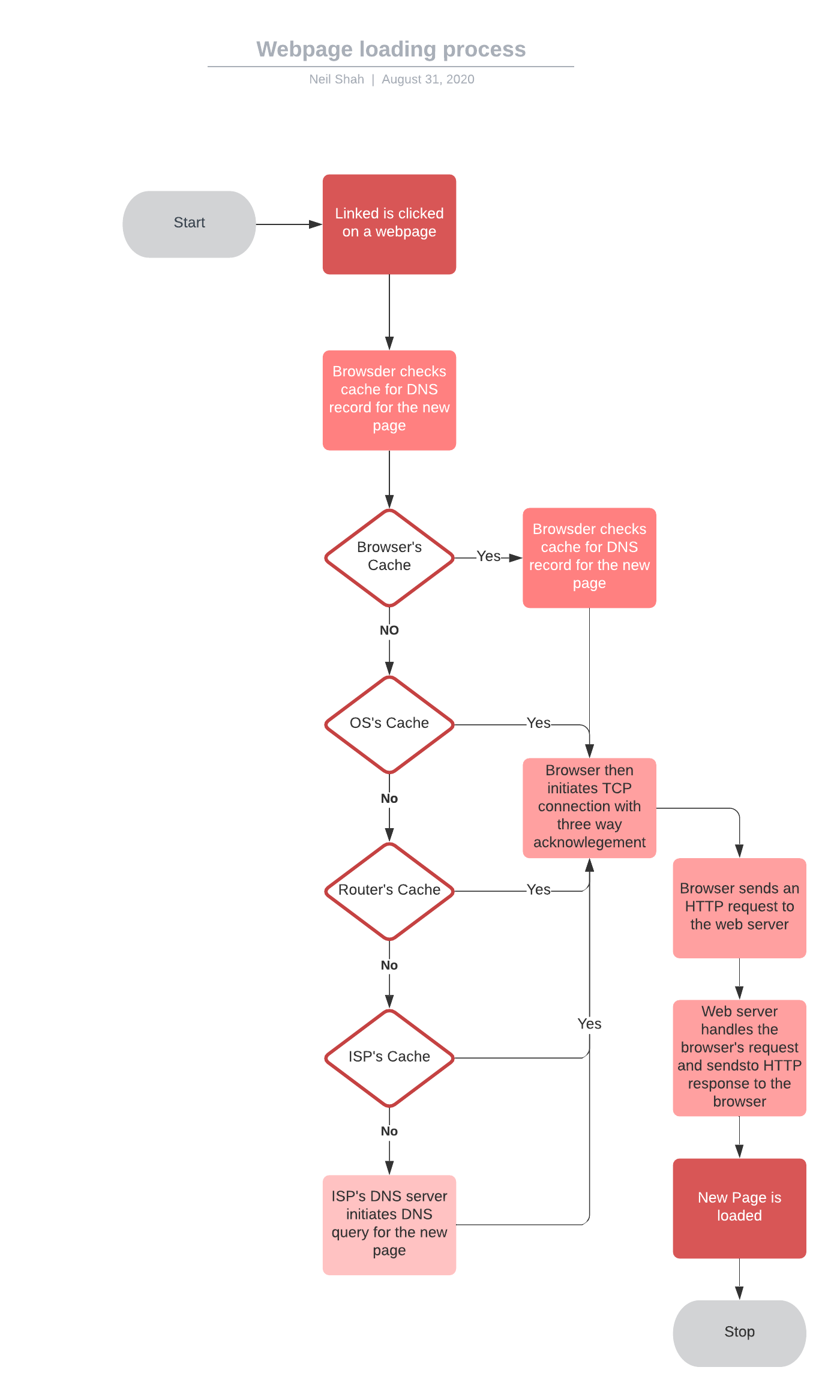
We reserve the right to verify your answers with online tools to check for plagiarism, so be sure not to just copy/paste something from Wikipedia.

* 1. Asking For Help

We believe that helping other team members and being willing to ask questions is an important part of a being a developer. If you get stuck or something doesn’t make sense, please reach out! Just email your hiring contact explaining what’s up and we’ll do our best to accommodate.

1. Requests & Debugging
2. 1. Request Lifecycle

Describe what happens in between clicking on a link on one web page and the new page being loaded and rendered by the browser. Be as detailed as you can.



* 1. Debugging: Page Doesn’t Load

You’ve developed a **(known working)** web application that, when accessed, will dynamically retrieve a set of data from an SQL database and display it as HTML. When you visit the URL of the application, the page is blank. Describe your troubleshooting steps.

I would first open the developer console tools and inspect the console, followed by this I would go the Network tab, and see if my request is failing and not giving a response code of “200”, based on the code I would debug it further such as if it’s a “502” code, I would look into why I cannot access the database and it is a bad gateway, looking at if my connection to the SQL server is down or the front end is not connecting properly. If I get a “200” code and the data is not displaying I would look to console log the data and see how I am processing it in the front end. And for other response codes I would look to resolve it looking at both the backend and frontend.

1. Development
2. 1. Data Types

Describe the differences, similarities, and common development pitfalls between: null, undefined, an empty string, the number 0, a string containing only the number 0, and false.

All these data types and points are similar in the sense that they represent nothingness or lack thereof. They differ in implementation however such as an empty string, the number 0, false and others have memory allocated for them and are stored, whereas undefined or null can refer to the lack of that object or data being stored in memory.

* 1. Recursion

Explain recursion. What are some common pitfalls when using it? Is it ever necessary? If so, what are some ways to determine whether or not a specific situation calls for it?

Recursion is the process of repeating a method or operation until a base case is met, and linearly processing the data passed to the recursive operation. Some pitfalls when implementing recursion, is large memory allocation, hard understanding conceptionally for others. Sometimes recursion may not be necessary as there is always a brute force way to solve it, but recursion could be the more optimal way to solve the problem at hand. Recursion is best in situations where one operation needs to be completed again but the data also can be truncated and the same operation can be applied to it.

* 1. Code Quality

How would you describe "good code"? What role (if any) do comments, refactoring, and code reviews play?

“Good code” to me is code that is detailed, easy to follow, and well commented, someone should be able to take a glance over the code, such as a Senior developer and know what your approach and thinking to solving a problem or implementing a feature is. Comments, refactoring, and code reviews are an important and core part of the SDLC, comments all for refreshing yourself when you come back to that code to know what it is doing and helps others when they read your code, refactoring helps keep the code base clean. Lastly, code reviews help us understand how to improve and write good code and to improve our code.

* 1. Code Structure

What criteria do you have for deciding when a single method/function does "too much" and needs to be broken down into multiple methods/functions?

The criteria I have when deciding to break down a single function or method, is typically if the code is trying to do much or portions of the function could also be used outside of that method or function, it becomes easier to debug and solve when you have functions doing smaller things, and built out a class or object to utilize those functions, it also allows for better code reusability and robustness.

1. Process
   1. Starting a Project

Explain your approach to a new project. Where do you start?

My approach to a new project is first to always design and research what stack, languages, and resources I would need based on the overall specifications of the project. Followed by this I would create tasks and subtasks if need be of core features and or system(s) I would need to add to the project.

* 1. Software Development Methodologies

Describe the differences and similarities between common software development methodologies. Which do you think is better and why?

The difference between Scrum and Kanban, is that scrum creates sprint cycles in which we have a set time to develop core features and fix issues, whereas in Kanban its more of a day to day progression where each day your complete tasks. I personally have great experience in Scrum as well as great success in scrum compared to Kanban, and I prefer it because over the course of a week or so, I can focus my resources on what needs to be done based on my high level estimates, with Kanban, it seems more lackluster to me as I would go at things day to day.

* 1. Testing

What role (if any) do you think testing plays in the development process? When would you use manual testing vs automated testing?

Testing plays a critical role in the development process, it is one of the most important processes as Testing could lead to finding issues with code prior to pushing to production and live users can see, this is why we must have integration, regression, and unit testing on our code base. I would chose to manual test when it is something that just needs to be tested a few times and not with each or majority of releases, automation is a core testing process as if something is being repeatedly tested or needs to be repeatedly tested, it makes far more sense to automate it then to manual do it.

1. Past Projects
   1. Wrong Approach

Have you ever spent all day (or even several days/weeks) working on a problem, only to find the solution you implemented was the wrong approach? What did you do?

Unfortunately, sometimes as a developer we think something will work in theory, but it does not end up working. I have also faced it while working on MoreTodos, I worked on creating the deployment pipeline for some time before I realized that I would need to get some perspective about why my way of thinking had a problem. I reached out to the other two members on my team and asked for their input. As soon as I asked for help, they made me realize what I was doing wrong and how I could fix it. I asked for help on an issue I was facing and instead of spending more hours progressing with not the most optimal outcome.

* 1. Completed Project

Describe the project you have worked on that you are most proud of. What was your part in the project that worked out particularly well?

The project I am most proud of is MoreToDos. It is something that my colleagues and I started when the pandemic forced us to stay indoors. We all wanted to sharpen our minds and learn something while staying safe and so we decided to make an application where users can access their todo lists. I worked on setting up the and authoring the backend, connecting the middleware and test the functionality of the application, also deploying it to Heroku, and creating the CI/CD system to deploy should any changes be pushed to the master branch, and all test cases in the pipeline pass.

1. Work Style
2. 1. Work Environment

Describe your ideal work environment.

My ideal work environment would be where I can grow my knowledge as well as grow as a person. I want to be in an environment where my voice and opinion are valued.

* 1. Existing Systems

Have you ever worked with existing legacy or third-party systems? If so, what were some of the challenges you faced and how did you solve them?

I’ve worked with third-party systems, some challenges I have faced where lack of documentation, deprecation, and large learning scope. To solve this issues, I would reach out to the author or senior developers who had experience in these systems, should I not be able to resolve it through my own means, also I would take the time to review and take notes on the available code base as well as the documentation and resources available online or offline.

1. Additional Comments
2. 1. Is there anything else you think we should know or you’d like to share?
3. Coding Samples
4. 1. Custom Sorting

**Task:** Write a function to sort a hand of cards.

**Input Parameters:** a list/vector/array of Card objects

**Return Parameter:** a list/vector/array of Card objects that are sorted

**Assumptions you don’t have to code:**

* Each card object has an attribute called **suit** that returns the suit of the card as a string: “Hearts”, “Spades”, “Clubs”, “Diamonds”
* Each card object has an attribute called **value** that returns a character representing their value: 2, 3, 4, 5, 6, 7, 8, 9, J, Q, K, A

**Requirements:**

* The cards should be sorted in **ascending** order unless you implement the bonus flag below. In that case, the default should be ascending unless overridden by the flag.
* **Bonus:** Add a flag to say whether to sort in ascending or descending order.

**Domain Knowledge:**

* Playing cards are ordered by **value** then by **suit**.
* For values, assume: 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < J < Q < K < A
* For suits, assume: Hearts < Diamonds < Clubs < Spades

**Example of Sorted Cards:**

1. 3 of Clubs
2. 7 of Hearts
3. Ace of Hearts
4. Ace of Spades

**Context/Hints:**

* Try to treat this like you would any other real-world sorting problem you may encounter in normal business logic.
* Feel free to write additional helper functions or other functions associated with the Card object to help accomplish the task.
  1. Data Validation

**Task:** Write a simple script to validate a set of contact records and report on any errors.

**Given:**

* A list of 20 contact objects (full names, city, phone number, and email address)

**Step 1: List all contact records with the following output:**

* Full name
* Whether the phone and email fields are "valid":
  + Output "Valid" if both email and phone are valid.
  + Output "Email is invalid." if email is invalid and phone is valid.
  + Output "Phone is invalid." if phone is invalid and email is valid.
  + Output "Email and Phone are invalid." if both phone and email are invalid.

**Step 2: List each city and report the following output:**

* Name of city
* Number of validation errors

**Requirements:**

* Contacts should be sorted alphabetically in **ascending** order.
* Cities should be sorted by number of validation errors in **descending** order.

**Validation criteria:**

* Email field: has exactly one @ symbol with data on each side
* Phone field: is numeric with only digits, dashes, and spaces allowed

**Data Set:** Use the records in Contacts.json, which are based off of U.S. census data via Wikipedia’s list of [given names](https://en.wikipedia.org/wiki/List_of_most_popular_given_names) and [surnames](https://en.wikipedia.org/wiki/List_of_most_common_surnames_in_North_America).

* 1. Regular Expressions

**Task:** Write a regular expression to parse timestamps.

**Requirements:**

* Extract the year/month/date and hours/minutes/seconds from a date timestamp.
* **Bonus:** Write the expression so that it will work if the timestamp is not included.

**Example Data**

With timestamps:

1. 2014-08-18T13:03:25Z
2. 2014/08/18T13:03:25Z

Without timestamps:

1. 2014-08-18
2. 2014/08/18
   1. Simple Web Form

**Requirements:**

* Page title should be “Team Introduction”.
* Page should have a simple HTML form that requests two fields of input: your name and a fun fact about yourself.
* The form should have a button to introduce yourself.
* When the submit button is clicked, both input fields should be validated and an alert message shown if there is an error.
* If there are no errors when the form is submitted:
  1. The validated data should be logged to the browser’s developer console.
  2. The form should be hidden and replaced with a new box displaying the input (name and fun fact).
* The “introduction box” should have a link or button to reset the form and allow “introducing” a different team member.
* **Bonus:** Use SCSS or describe how you’d refactor your CSS if SCSS was available in the build environment.