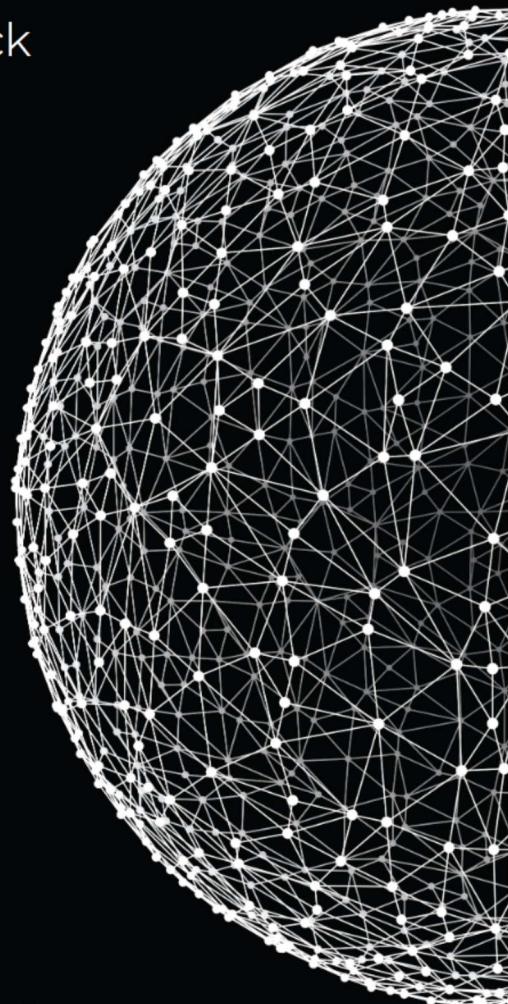


# Sprint 07

## Half Marathon Full Stack

November 15, 2021



**ucode connect**

## Contents

Engage . . . . .	2
Investigate . . . . .	3
Act Basic: Task 00 > Session for new . . . . .	5
Act Basic: Task 01 > Hack it! . . . . .	7
Act Basic: Task 02 > Charset . . . . .	9
Act Basic: Task 03 > Files . . . . .	11
Act Basic: Task 04 > Data to XML . . . . .	15
Act Basic: Task 05 > Show other sites . . . . .	16
Act Basic: Task 06 > Marvel API . . . . .	17
Act Advanced: Task 07 > Cookie counter . . . . .	20
Act Advanced: Task 08 > WhoIsWho . . . . .	21
Act Advanced: Task 09 > Notepad mini . . . . .	23
Act Advanced: Task 10 > Make square image . . . . .	25
Share . . . . .	27

**ucode connect**

# Engage

## DESCRIPTION

Awesome, we are glad you were able to get here!

Guess what? This **Sprint** is all about Node.js. Again. Because there are so many things you need to learn before you start creating your first web service.

Passwords hashing, for example. Have you thought about how Facebook or Google store your passwords? Have you heard anything about password encryption? Password protection is the most important aspect of any user account system. User account databases are hacked frequently, so you absolutely must do something to protect passwords of your users in case your web service is ever breached. One of the rules is to treat your password like a toothbrush: never share it and change it every couple of months.

You will also work more closely with the HTTP protocol and its request methods. In short, HTTP is what makes the World Wide Web work, giving browsers a language to communicate to remote servers that host web pages. The thing that is closely related to HTTP is XML - a tool for storing and transporting data. When you understand the basics of HTTP, you will be able to operate with files on your computer from the browser, namely create, update, and delete files.

And in the end, a more creative task will be waiting for you. Believe us, you want to see it.

## BIG IDEA

Basics for web service development.

## ESSENTIAL QUESTION

How to establish communication between a web page and a computer's local storage?

## CHALLENGE

Learn to operate with data received from a web page.



**Sprint 07 | Half Marathon Full Stack > 2**

## Investigate

### GUIDING QUESTIONS

We invite you to find answers to the following questions. By researching and answering them, you will gain the knowledge necessary to complete the challenge. To find solutions, ask the students around you and search the internet. We encourage you to ask as many questions as possible. Note down your findings and discuss them with your peers.

- Which HTTP request methods do you know?
- How to communicate with an HTML form using Node.js?
- What are the advantages of using the `express` framework?
- What is the difference between `GET` and `POST` methods?
- How to work with sessions?
- What is a password hash? How can you create a hash of the password?
- How to work with files in Node.js?
- Are there some special methods or modules to work with CSV files?
- How to use the `Fetch API` for Node.js?
- What is XML?
- What is a `character set`? Which character sets do you know?
- How to work with images using Node.js?

### GUIDING ACTIVITIES

Complete the following activities. Don't forget that you have limited time to overcome the challenge. Use it wisely. Distribute tasks correctly.

- Research various methods of making and processing requests in Node.js.
- Discover how to work with XML in Node.js.
- Find out what are the most popular and secure hashing algorithms these days.
- Learn how to add salt to hashing.
- Read about `peppered` passwords.
- Let your brain relax from programming and watch [this video](#). Watch it till the end, it may be useful.
- Clone your git repository issued on the challenge page in the LMS.
- Brainstorm with other students and find more than one solution.
- Try to implement your thoughts in code.

### ANALYSIS

Analyze your findings. What conclusions have you made after completing guiding questions and activities? In addition to your thoughts and conclusions, here are some more analysis results.

- Be attentive to all statements of the story. Examine the given examples carefully. They may contain details that are not mentioned in the task.



Sprint 07 | Half Marathon Full Stack > 3

- All tasks are divided into **Act Basic** and **Act Advanced**. This means that the complexity of the tasks increases gradually. Try to complete all tasks to get maximum points and more knowledge.
- Analyze all the information you have collected during the preparation stages.
- Perform only those tasks that are given in this document.
- Submit only the specified files in the required directory and nothing else. Garbage shall not pass.
- Pay attention to what is allowed. Use of forbidden stuff is considered a cheat, and a reason to fail the challenge.
- Complete tasks according to the rules specified in the following style guides:
  - HTML and CSS: [Google HTML/CSS Style Guide](#). As per section [3.1.7 Optional Tags](#), it doesn't apply. Do not omit optional tags, such as `<head>` or `<body>`
  - JavaScript:
    - \* [JavaScript Style Guide and Coding Conventions](#)
    - \* [JavaScript Best Practices](#)
    - \* [Node.js Best Practices](#)
- The solution will be checked and graded by students like you. [Peer-to-Peer learning](#).
- Your work may also be graded by your mentor. So, be ready for that.
- Also, the challenge will pass automatic evaluation which is called [Oracle](#).
- If you have any questions or don't understand something, ask other students or just google it.

## Act Basic: Task 00

### NAME

Session for new

### DIRECTORY

t00\_session\_for\_new/

### SUBMIT

index.js, \*.js, [optionally:] \*.html, package.json

### ALLOWED

JS, HTML

### DESCRIPTION

Create a web page with the form you have created earlier (information about a hero). See the [EXAMPLE](#). This time, your page must save the data submitted to the session:

- if the data wasn't saved at the session earlier - show the form with a `SEND` button
- if some data is contained in the session - show only the data and a `FORGET` button, which clears the session. See the [EXAMPLE](#)

Note that the data can be sent via GET and via POST.

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 5

**EXAMPLE****Session for new**

About HERO

Real Name: Carol Danvers	Current Alias: Captain Marvel	Age: 29
Carol Danvers, also known as Captain Marvel, is a former United States Air Force pilot who, upon being exposed to the energy of the Tesseract via the destruction of the Light Speed Engine, obtained cosmic powers.		
About: <input type="text"/>		
Photo: <input type="file"/> No file chosen.		
Powers:		
<input checked="" type="checkbox"/> Strength <input type="checkbox"/> Speed <input type="checkbox"/> Intelligence <input type="checkbox"/> Teleportation <input type="checkbox"/> Immortal <input type="checkbox"/> Another		
Level of control: <input type="range"/>		
Publicity:		
<input type="radio"/> UNKNOWN <input type="radio"/> LIKE A GHOST <input checked="" type="radio"/> I AM IN COMICS <input type="radio"/> I HAVE FUN CLUB <input type="radio"/> SUPERSTAR		
<input type="button"/> CLEAR <input type="button"/> SEND		

**Session for new**

name : Carol Danvers  
alias : Captain Marvel  
age : 29  
description : Carol Danvers, also known as Captain Marvel, is a former United States Air Force pilot who, upon being exposed to the energy of the Tesseract via the destruction of the Light Speed Engine, obtained cosmic powers.  
photo :  
experience : 2  
level : 10  
purpose : 2

 FORGET**ucode connect****Sprint 07 | Half Marathon Full Stack > 6**

## Act Basic: Task 01

### NAME

Hack it!

### DIRECTORY

t01\_hack\_it/

### SUBMIT

index.js, \*.js, [optionally:] \*.html, package.json

### ALLOWED

JS, HTML

### LEGEND

"I should probably look into that once my decryption program finishes breaking into all of S.H.I.E.L.D.'s secure files."

### BEFORE YOU BEGIN

Read about the role of `salt` in the context of password encryption on the internet.

### DESCRIPTION

Create a web page with input fields for a password and for salt.

Also, add a `Save` button to save data at a session. If the data was saved, show the hashed with salt password on the page and add a field to guess the password.

Add a `Clear` button that allows to clear the session.

If the user enters the correct password, display a message `Hacked!`, clean the previous session, and go to the page with the password input field for new password saving.

If the entered password is incorrect, show a message `Access denied!` with red color on the same page.

See the `EXAMPLE`.

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 7

**EXAMPLE**

# Password

Password not saved at session.

Password for saving to session

Salt for saving to session

# Password

Password saved at session.

Hash is \$1\$In5NIpON\$tkfxewLIL9afdiafn5RkM.

Try to guess:

# Password

Hacked!

Password not saved at session.

Password for saving to session

Salt for saving to session

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 8

## Act Basic: Task 02

### NAME

Charset

### DIRECTORY

t02\_charset/

### SUBMIT

index.js, \*.js, [optionally:] \*.html, package.json

### ALLOWED

JS, HTML

### LEGEND

"HYDRA, S.H.I.E.L.D., two sides of a coin that's no longer currency."

### DESCRIPTION

Just like in our legend, where HYDRA and S.H.I.E.L.D. are two sides of a coin, the same characters may differ depending on the encoding as well.

Create a web page for string conversion with three different charsets.

The web page must have:

- a field for an input string to convert
- a field with three selection options (three charset types)
- a `Change charset` button to change the charset
- a `Clear` button to return the page to initial state
- inactive output fields with converted strings  
(can be several if more than one charset is selected)

See the [EXAMPLE](#).

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 9

**EXAMPLE**

## Charset

Change charset:

UTF-8  
ISO-8859-1  
Windows-1252

Select charset or several charsets:  Change charset

## Charset

Change charset:

UTF-8  
ISO-8859-1  
Windows-1252

Change charset

Select charset or several charsets:

\$ and € are a currency  
 Input string:  
 \$ and € are a currency  
 UTF-8  
 \$ and EUR are a currency  
 ISO-8859-1  
 \$ and ₣ are a currency  
 Windows-1252

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 10

## Act Basic: Task 03

### NAME

Files

### DIRECTORY

t03\_files/

### SUBMIT

index.js, File.js, fileList.js, \*.js, [optionally:] \*.html, package.json

### ALLOWED

JS, HTML

### LEGEND

"Tony Stark: J.A.R.V.I.S., are you up?

J.A.R.V.I.S.: For you sir, always.

Tony Stark: I'd like to open a new project file, index as: Mark 2.

J.A.R.V.I.S.: Shall I store this on the Stark Industries' central database?

Tony Stark: I don't know who to trust right now. 'Til further notice, why don't we just keep everything on my private server.

J.A.R.V.I.S.: Working on a secret project, are we, sir?

Tony Stark: I don't want this winding up in the wrong hands. Maybe in mine it could actually do some good."

### DESCRIPTION

Create a web page with three sections:

- file creation form that contains
  - a required input field for a file name
  - a required input field for file content
  - a button for creating a file in the `tmp` directory with the name and content submitted in the form
- list of files that displays all files in the directory `tmp` as links
- current file section (visible only if a file is selected) that contains
  - the name of the file
  - the content of the file
  - a button for deleting the file from the `tmp` directory

By clicking on a filename from the file list, the file becomes selected, and the current file section displays its info. Notes:

- if the user tries to create a file that already exists, append the content to the end of the file
- if, when creating a file, the `tmp` directory doesn't exist, create it



Sprint 07 | Half Marathon Full Stack > 11

See the **CONSOLE OUTPUT** and the **EXAMPLE** screenshots.

Create classes for file manipulations:

- **FileList** - all manipulations with a list of files
- **File** - all manipulations with files (create, write, read, etc.)

#### CONSOLE OUTPUT

```
>node t04/test.js
Some text.
[ 'example_file.txt', 'list.txt', 'SHIELD_info' ]
true
<ul><li><a href="/select-file?file=example_file.txt">example_file.txt</a></li><li><a href="/select
    -file?file=list.txt">list.txt</a></li><li><a href="/select-file?file=SHIELD_info">SHIELD_info
    </a></li></ul>
[ 'list.txt', 'SHIELD_info' ]
```

#### EXAMPLE

```
/*
 * Sprint 07
 * Task name: Files
 */

const File = require('./File.js');
const fileList = require('./fileList.js');

let file = new File('example_file.txt');
file.write('Some text.');
const content = file.read();
console.log(content);

const fileList = new fileList();
console.log(fileList.getList());
console.log(fileList.hasFiles());
console.log(fileList.getHTMLList());

file.delete();
console.log(fileList.getList());
```



Sprint 07 | Half Marathon Full Stack > 12

**File manager**

Create a file

Filename

Content

Create file

Files:

**File manager**

Create a file

Filename  SHIELD\_info

The Strategic Homeland Intervention, Enforcement, and Logistics Division, better known by its acronym S.H.I.E.L.D., is an American extra-governmental military counter-terrorism and intelligence agency, tasked with maintaining both national and global security.

Content

Create file

Files:

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 13

**File manager**

Create a file

Filename

Content

Create file

**Files:**

- [list.txt](#)
- [SHIELD\\_info](#)
- [super\\_secret\\_passwords](#)

**Selected file:**

**SHIELD\_info**

Content:

The Strategic Homeland Intervention, Enforcement, and Logistics Division, better known by its acronym S.H.I.E.L.D., is an American extra-governmental military counter-terrorism and intelligence agency, tasked with maintaining both national and global security.

Delete file

**File manager**

Create a file

Filename

Content

Create file

**Files:**

- [list.txt](#)
- [super\\_secret\\_passwords](#)

## Act Basic: Task 04

### NAME

Data to XML

### DIRECTORY

t04\_data\_to\_xml/

### SUBMIT

index.js, AvengerQuote.js, ListAvengerQuotes.js, Comment.js, \*.js  
[optionally:] \*.html, package.json

### ALLOWED

JS, HTML

### LEGEND

"If you're nothing without this suit then you shouldn't have it."

### BEFORE YOU BEGIN

Do you know what XML is? Do you know for which purposes it can be used?  
In simple words - HTML is a "suit" for XML.  
Find out more by yourself.

### DESCRIPTION

Create a class `AvengerQuote` with such properties:

- id
- author - name of the Avenger who said it
- quote
- photo - array of links to images from a movie scene
- publishDate
- comments - object date, comment

You may also need `ListAvengerQuotes` and `Comment` classes.

Create functions to convert arrays of `AvengerQuote` to and from XML:

- `toXML(fileName)` - save XML to file `avenger_quote.xml`
- `fromXML(fileName)` - read XML from file `avenger_quote.xml`

Create an array of four `AvengerQuote` objects with at least one photo and one comment.

Show on page arrays of `AvengerQuote` before and after XML conversion in two columns.

### SEE ALSO

Parsing and serializing XML



Sprint 07 | Half Marathon Full Stack > 15

## Act Basic: Task 05

### NAME

Show other sites

### DIRECTORY

t05\_show\_other\_sites/

### SUBMIT

index.js, \*.js, [optionally:] \*.html, package.json

### ALLOWED

JS, HTML

### LEGEND

"Spider-Man: [referring to Captain America's shield]  
That thing doesn't obey the laws of physics at all!"

### DESCRIPTION

Create a web page with a form:

- input for a url address
- a go button

Upon pressing the button:

- load the page from the url address
- get content between <body> tags

On the page, show the HTML code of content in <div> tags.

### EXAMPLE

```
Show other sites
 BACK
url: https://apply.ucode.world

<body>
<!-- Google Tag Manager (noscript) -->
<noscript><iframe src="https://www.googletagmanager.com/ns.html?id=GTM-XH4BMB" height="0" width="0" style="display:none;visibility:hidden"></iframe></noscript>
<!-- End Google Tag Manager (noscript) -->
<script>Please enable Javascript to continue using this application.</script>
<script src="polyfills-es5.99ca0d99b7f7aefc10.js" nomodule defer></script><script src="polyfills-es2015.d4f99e4027bf5a14be99.js" type="module"></script><script src="runtime-es2015.35bdc9e9994edede711f.js"></script>
```

Show other sites

BACK

Type an URL...

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 16

## Act Basic: Task 06

**NAME**

Marvel API

**DIRECTORY**

t06\_marvel\_api/

**SUBMIT**

index.js, \*.js, \*.css, [optionally:] \*.html, package.json

**ALLOWED**

JS, HTML, CSS

**LEGEND**

"The Collector: Before creation itself, there were six singularities. Then the universe exploded into existence, and the remnants of these systems were forged into concentrated ingots... Infinity Stones."

**DESCRIPTION**

Did you know that Marvel has its own API? They do. Maybe you can even find Infinity Stones there... Who knows. Your mission for this task is to explore the features of their API.

Go to the [link](#), figure out what you can get from there and how.

Create a web page that displays the API request output in a structured format (not JSON). You can choose any endpoint that you want. But a heading on your page must be modified in accordance with the name of this endpoint. Also, you can create the same design as in the example or create your own. It's up to you.

See the [EXAMPLE](#).

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 17

**EXAMPLE****Comics from Marvel API**

```
code: 200
status: Ok
copyright: © 2019 MARVEL
attributionText: Data provided by Marvel, © 2019 MARVEL
attributionHTML: Data provided by Marvel, © 2019 MARVEL
etag: 7876897caa5d6cafdf605905b8cb5f0659be6f1f
```

**data:**

```
offset: 0
limit: 20
total: 45380
count: 20
```

**results:****0:**

```
id: 1332
digitalId: 0
title: X-Men: Days of Future Past (Trade Paperback)
issueNumber: 0
variantDescription:
```

```
description: "Re-live the legendary first journey into the dystopian future of 2013 - where Sentinels stalk the Earth, and the X-Men are humanity's only hope...until they die! Also featuring the first appearance of Alpha Flight, the return of the Wendigo, the history of the X-Men from Cyclops himself...and a demon for Christmas!?"
```

**Sprint 07 | Half Marathon Full Stack > 18**

## Comics from Marvel API

```
code: 200
status: Ok
copyright © 2019 MARVEL
attributionText: Data provided by Marvel. © 2019 MARVEL
attributionHTML: Data provided by Marvel. © 2019 MARVEL
etag: 7876897caa5d6caf4f605905b8ch5f0659be6ff1f

data:
  offset: 0
  limit: 20
  total: 45380
  count: 20

results:
  0:
    id: 1332
    digitalId: 0
    title: X-Men: Days of Future Past (Trade Paperback)
    issueNumber: 0
    variantDescription:
      description: "Re-live the legendary first journey into the dystopian future of 2013 - where Sentinels stalk the Earth, and the X-Men are humanity's only hope...until they die! Also featuring the first appearance of Alpha Flight, the return of the Wendigo, the history of the X-Men from Cyclops himself...and a demon for Christmas!?"
```

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 19

## Act Advanced: Task 07

**NAME**

Cookie counter

**DIRECTORY**

t07\_cookie\_counter/

**SUBMIT**

index.js, \*.js, [optionally:] \*.html, package.json

**ALLOWED**

JS, HTML

**LEGEND**

"[after witnessing Scott transform into Giant-Man] Iron Man: Okay, anybody on our side hiding any shocking and fantastic abilities they'd like to disclose?  
I'm open to suggestions."

**DESCRIPTION**

Create a web page that shows the count of page loads for the last minute using cookies.

**EXAMPLE**

# Cookie counter

This page was loaded 7 time(s) in last minute

**SEE ALSO**

Express middleware

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 20

## Act Advanced: Task 08

**NAME**  
WhoIsWho

**DIRECTORY**  
t08\_whoiswho/

**SUBMIT**  
index.js, \*.js, [optionally:] \*.html, package.json

**ALLOWED**  
JS, HTML

**LEGEND**  
"Pepper Potts after destroying Killian: Oh my God, that was really violent."

**DESCRIPTION**  
That is really violent - you may say while developing your own small CSV reader. But still, it will be very useful for your skills. Your reader must have the following features:
 

- open a file with a path in your file system
- filter by every column with all existing values in this column

 See the [EXAMPLE](#). Also, you can use `marvel_characters_info.csv` from resources for testing.

**EXAMPLE**

**Parsing CSV data**

Upload file:  No file chosen

ID	Name	Alignment	Gender	EyeColor	Race	HairColor	Publisher	SkinColor	height	Weight
0	A-Bomb	good	Male	yellow	Human	No Hair	Marvel Comics	-	203	441
1	Aine Sapien	good	Male	blue	Tenthyo Sapien	No Hair	Dark Horse Comics	blue	191	65
2	Abin Sur	good	Male	blue	Unsparan	No Hair	DC Comics	red	185	90
3	Abomination	bad	Male	green	Human / Radiation	No Hair	Marvel Comics	-	203	441
4	Abrahexes	bad	Male	blue	Comic Entity	Black	Marvel Comics	-	199	39
5	Absorbing Man	bad	Male	blue	Human	No Hair	Marvel Comics	-	193	122

**Parsing CSV data**

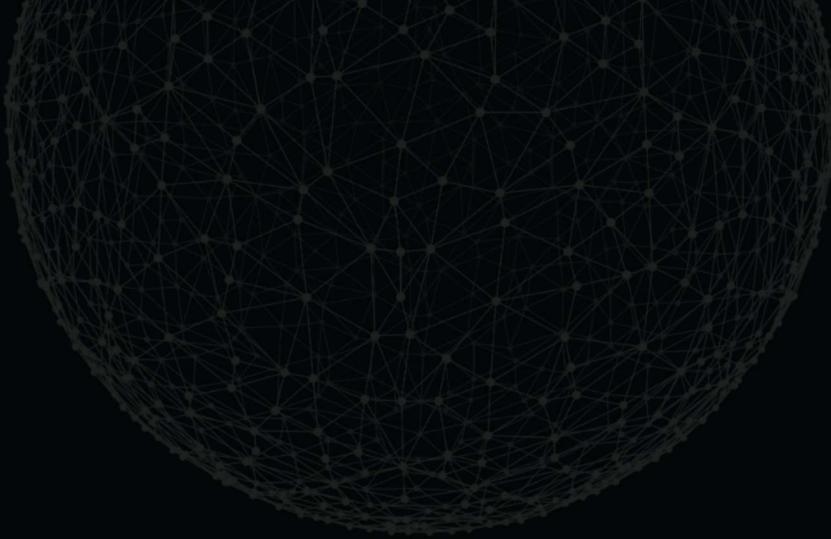
Upload: NOT SELECTED No file chosen Upload  
Filter: good APPLY

ID	Name	Alignment	Gender	EyeColor	Race	HairColor	Publisher	SkinColor	Height	Weight
0	neutral	good	Male	yellow	Human	No Hair	Marvel Comics	-	203	441
1	Abe Sapien	good	Male	blue	Ictyo Sapien	No Hair	Dark Horse Comics	blue	191	65
2	Abin Sur	good	Male	blue	Ungaran	No Hair	DC Comics	red	185	90
3	Abomination	bad	Male	green	Human / Radiation	No Hair	Marvel Comics	-	203	441
4	Ahraxas	bad	Male	blue	Cosmic Entity	Black	Marvel Comics	-	99	99
5	Abundant Man	bad	Male	blue	Human	No Hair	Marvel Comics	-	103	127

**Parsing CSV data**

Upload file: Choose file No file chosen Upload  
Filter: good APPLY

ID	Name	Alignment	Gender	EyeColor	Race	HairColor	Publisher	SkinColor	Height	Weight
0	A-Bomb	good	Male	yellow	Human	No Hair	Marvel Comics	-	203	441
1	Abe Sapien	good	Male	blue	Ictyo Sapien	No Hair	Dark Horse Comics	blue	191	65
2	Abin Sur	good	Male	blue	Ungaran	No Hair	DC Comics	red	185	90
6	Adam Monroe	good	Male	blue	-	Blond	NBC - Heroes	-	99	99
7	Adams Strange	good	Male	blue	Human	Blond	DC Comics	-	185	88
8	Aazat 13	good	Female	blue	-	Blond	Marvel Comics	-	173	61



**ucode connect**

Sprint 07 | Half Marathon Full Stack > 22

## Act Advanced: Task 09

### NAME

Notepad mini

### DIRECTORY

t09\_notepad\_mini/

### SUBMIT

index.js, Note.js, NotePad.js, \*.js, [optionally:] \*.html, package.json

### ALLOWED

JS, HTML

### LEGEND

"Jane Foster: Years of research. Gone.  
Darcy: They even took my iPod.  
Dr. Erik Selvig: What about the backups?  
Jane Foster: They took our backups. They took the backups of our backups. They were extremely thorough.  
Darcy: Just downloaded like thirty songs onto there."

### DESCRIPTION

Help Jane to restore data.

Create a web form for notes with:

- note name
- note importance - low, medium, high
- note content

By clicking a button:

- add the current date/time
- serialize data to JSON
- save data in the file

Under the form show a list of saved notes:

- "date > name" - link to see the details of the notes
- "delete" - link to delete the current note

Use a single page. See the EXAMPLE.

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 23

**EXAMPLE**

## Notepad mini

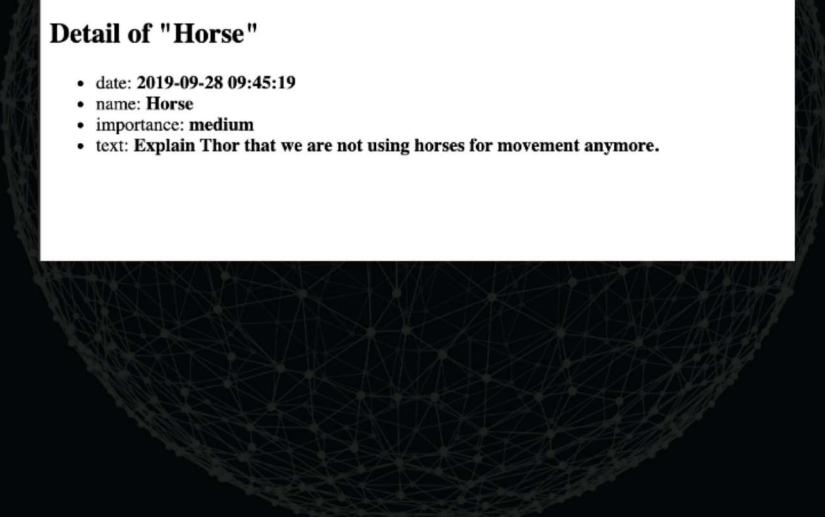
  
 low   
  

### List of notes

- [2019-09-28 09:14:08 > Read about Thor](#) [DELETE](#)
- [2019-09-28 09:44:15 > Find Thor on Facebook](#) [DELETE](#)
- [2019-09-28 09:45:19 > Horse](#) [DELETE](#)

### Detail of "Horse"

- date: **2019-09-28 09:45:19**
- name: **Horse**
- importance: **medium**
- text: **Explain Thor that we are not using horses for movement anymore.**



## Act Advanced: Task 10

**NAME**

Make square image

**DIRECTORY**

t10\_make\_square\_image/

**SUBMIT**

index.js, \*.js, \*.css, [optionally:] \*.html, package.json

**ALLOWED**

JS, HTML, CSS

**LEGEND**

"Tony Stark: Dude, you're embarrassing me in front of the wizards."

**DESCRIPTION**

Create a web page with the form on which you must place:

- an input field for a url to an image
- a button

When the form receives a url, the page will display the image.

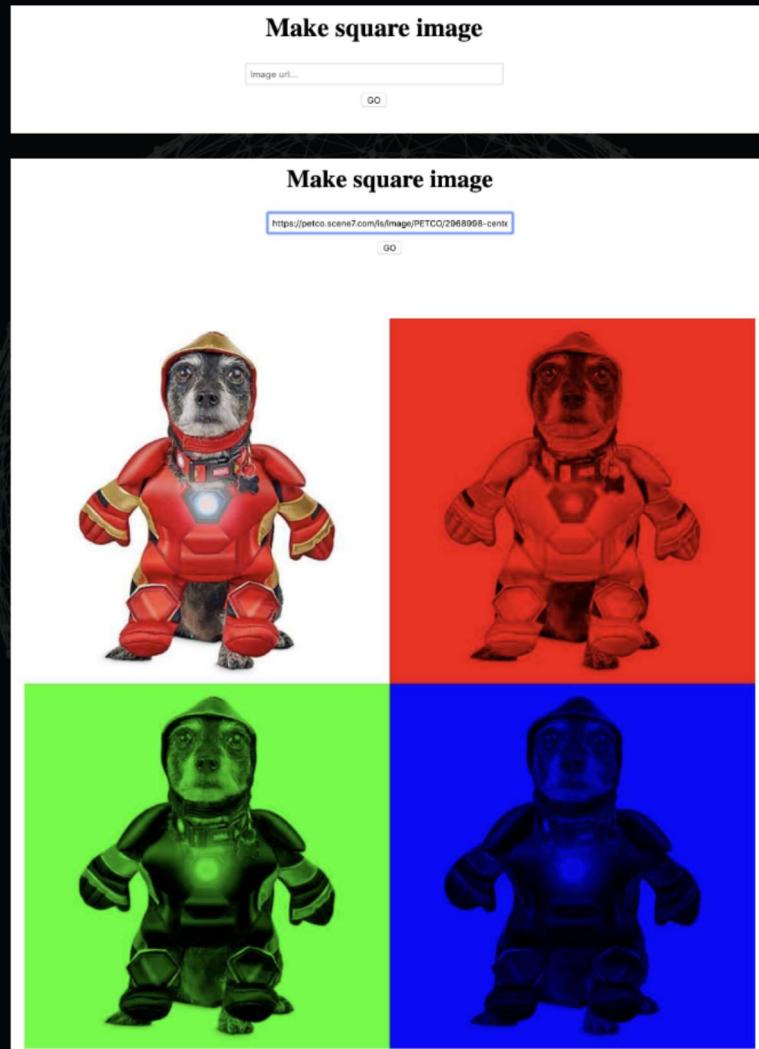
Upon clicking the button:

- load the image from the link
- cut the bigger side to create a square image
- create a new square image from:
  - prepared image
  - R channel
  - G channel
  - B channel
- show the obtained image on the page

See the **EXAMPLE**.

**ucode connect**

Sprint 07 | Half Marathon Full Stack > 25

**EXAMPLE****ucode connect****Sprint 07 | Half Marathon Full Stack > 26**

## Share

### PUBLISHING

Last but not least, the final stage of your work is to publish it. It allows you to share your challenges, solutions, and reflections with local and global audiences. During this stage, you will discover ways of getting external evaluation and feedback on your work. As a result, you will get the most out of the task and better understand your achievements and missteps.

To share your work, you can create:

- a text post, as a summary of your reflection
- charts, infographics or other ways to visualize your information
- a video, either of your work, or a reflection video
- an audio podcast. Record a story about your experience
- a photo report with a small post

Helpful tools:

- [Canva](#) – a good way to visualize your data
- [QuickTime](#) – an easy way to capture your screen, record video or audio (macOS)
- [ScreenToGif](#) – screen, webcam, and sketchboard recorder with an integrated editor (Windows)

Examples of ways to share your experience:

- [Facebook](#) – create and share a post that will inspire your friends
- [YouTube](#) – upload an exciting video
- [GitHub](#) – share and describe your solution
- [Instagram](#) – share photos and stories from ucode. Don't forget to tag us :)

Share what you've learned and accomplished with your local community and the world. Use [#ucode](#) and [#CBLWorld](#) on social media.



Sprint 07 | Half Marathon Full Stack > 27