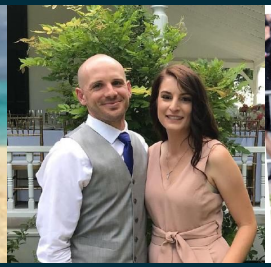
Welcome to...

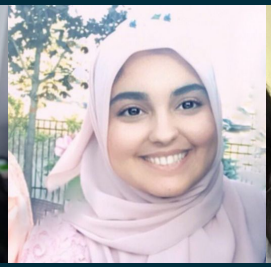
LIGHTHOUSE LABS

Our Team

Here are some soon-to-be familiar faces!





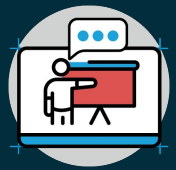




Taiwo Pedro Eric Sakhia Chetna Hajrah Kristen

Instructors

Education Manager

Student Success Coordinators



LIGHTHOUSE LABS

BE NOT AFRAID

It is easy to freeze up when approaching a problem.

When it comes to programming, the most important part is *trying*.

Write some code. It might not work... but we can change it and try, try again! Experiment. Break stuff. Fix it again! Share what you learn!

LIGHTHOUSE LABS

IS THERE ANYTHING...

you're *worried* about in your Lighthouse Labs adventure?

We're all in this together!

THE LIGHTHOUSE LABS CURRICULUM

Our curriculum is composed of 10 modules.

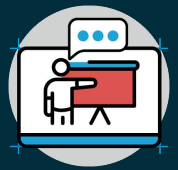
We explore essentials ranging from theory to practical, and from front-end to back-end!

LIGHTHOUSE LABS

MODULE 1 (Weeks 1 - 4) Programming Fundamentals with Javascript

FOCAL: Functions, Objects, Conditionals, Arrays, Loops.

Dev Approach: Code Style & Quality, Testing, Debugging, Problem Solving



LIGHTHOUSE LABS

MODULE 2 (Week 5) Networking and HTTP for Web Developers

Asynchronous Control Flow (Callbacks, Promises). Networking, HTTP & APIs. NPM and Packages. Unit Testing with Mocha & Chai

LIGHTHOUSE LABS

MODULE 3 (Week 6 - 7)
Intro to Web Server Development with Node

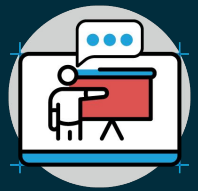
Your First Web App -- HTTP Servers, Express.js, Cookies, Basic HTML & Forms

Programming Test #3 & Data Structures (Mostly: Trees)

LIGHTHOUSE LABS

MODULE 4 (Weeks 8 - 11)
Intro to Front - End Development

Front-end -- Client-side JS. Browsers. jQuery, HTML, CSS, Box Model. AJAX



LIGHTHOUSE LABS

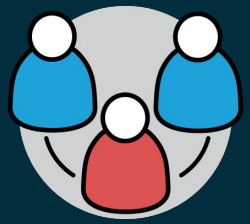
MODULE 5 (Weeks 12 - 13)
Relational Databases and SQL

Data -- Relational Databases, SQL, Data Design. Postgres.

LIGHTHOUSE LABS

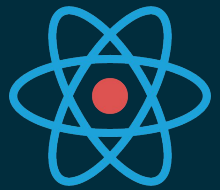
MODULE 6 (Weeks 14 - 15)
Mid-Term Project

We choose the groups, you pick from a list of possible projects.



LIGHTHOUSE LABS

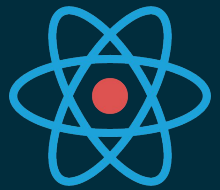
MODULE 7 (Weeks 16 - 19)
React



LIGHTHOUSE LABS

MODULE 8 (Weeks 20 - 21)

Automated Testing in React



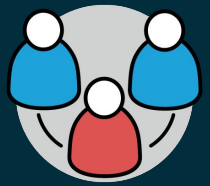
LIGHTHOUSE LABS

MODULE 9
Ruby on Rails
(Weeks 22 - 26)

LIGHTHOUSE LABS

MODULE 10 (Weeks 27 - 30)
Final Projects

You choose the group, you choose the project!



LIGHTHOUSE LABS

Major Solo Projects

1: Due Week 4 - Lotide due
2: Due Week 5 - Snek due
3: Due Week 7 - TinyApp due
4: Due Week 10 - Tweeter due
5: Due Week 13 - LighthouseBnB due
6: Due Week 19 - Scheduler due

LIGHTHOUSE LABS

PROGRAMMING TESTS

First one focused on FOCAL, not building apps.

Just as important as the projects. No more. No less.

Mock test first.

LIGHTHOUSE LABS

Tech Interviews.

Week 7,
Week 18

LIGHTHOUSE LABS

Quizzes (Multiple choice)

LIGHTHOUSE LABS

```
assessment = {
  completion:     [],
  codeReviews:    [],
  techInterviews: [],
  projectEvals:   [],
  quizAnswers:    [],
  testAnswers:    [],
    assistances:      []
};
```

LIGHTHOUSE LABS

LEARNING TO PROGRAM

Learning doesn't happen without failure. Try stuff, break stuff, fix stuff.

People don't often enough speak on the fifty-six times they failed... they brag about the one time that worked!

LIGHTHOUSE LABS

YOUR BIGGEST ENEMY?

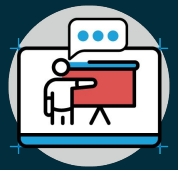
Don’t freeze in the face of a problem. Break it into small pieces... write pseudo-code... and try something!

Look at previous examples from class, or from your own experiments. There are often pieces you can carry to new challenges!

LIGHTHOUSE LABS

Approach to Lectures

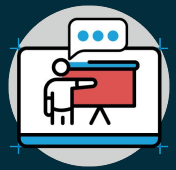
- Lectures are offered over Zoom; invites sent via Discord
- Tuesday and Thursday @ 1:00 PM ET / 10:00 AM PT
- Approximately 2 hours with a break near the middle
- Keep your camera on so we can see and engage with each other
- Ask questions (via chat, or put your hand up *(ALT+Y)*)
- Take notes (don't code every line the instructor types)



LIGHTHOUSE LABS

Lectures are Not

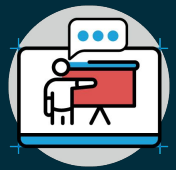
- ...time to work on your exercises—be present to make the most of each lecture!
- ...code-along sessions.
  - Feel free to write small experiments!
  - Feel free to peruse the example using the provided GitHub link!
  - Don't fall into the trap of trying to type *everything* and not having time to build *understanding*.



LIGHTHOUSE LABS

Visual Studio Code

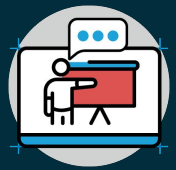
- We recommend using the free and powerful: Visual Studio Code
- Get familiar with the shortcuts, they save a *lot* of time!
  Go to **Help→Keyboard Shortcuts Reference** for your OS' instructions.
  - Linux
  - MacOS
  - Windows



LIGHTHOUSE LABS

VSCode Extensions

- When getting started, avoid using tools that write a lot of code for you (like GitHub Copilot); while powerful, they often make it difficult for you to learn *how* your code works, and it discourages essential repetition when engaging with new concepts
- **Do use** extensions that make your code more readable, and help you maintain a high standard in your code formatting:
  - ESLint
  - Rainbow Brackets
  - Prettier (wait a few weeks, then use this to save some time)

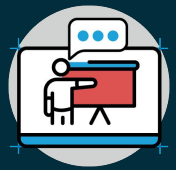


LIGHTHOUSE LABS

Approaching Problems

How to approach problem solving?

- List the steps in order to solve a problem (don't think about syntax)
- Step-by-step process
  a. State hypothesis
  b. Verify the hypothesis
  c. Make changes
- We express ourselves through code (like an author with a book)
- Make sure your book can be understood!



LIGHTHOUSE LABS

Thank you.

LIGHTHOUSE LABS