avion

Lesson 1: Introduction to Web Development

FRONT-END COURSE (V1.0.1)

What does it mean to be a...

Full Stack? Back End? Front End? Senior? Mid-level? Junior? Entry-Level? Associate?
Software Architect? Solutions Architect? Consultant? Tech Lead? Team Lead?
Database? API? Application? Mobile App? Tester? QA? Analyst? Dev Ops? SEO Specialist?

...

C? C#? Objective C? .NET? Java? Spring Boot? Grails? PHP? Symfony? CodeIgniter? Python? Django? Flask? Golang?
Ruby on Rai [illegible] e? Docker?
Jenkins? MySQL? Oracle PL/SQL Postgres? MongoDB? NoSQL? HTML? CSS? Javascript? ES6? React? XML? AJAX? SASS? JQuery?
TypeScript? Redux? Relay? Webpack? Gulp? Angular? Vue? Ember? Backbone? Gatsby? NextJS? Nuxt JS? CMS? WordPress?
Shopify? Laravel? Android? IOS? React Native? Flutter? Web Designer? Interaction? UI? UX? Product? Figma? CI / CD? SVN? Git?
Kafka? Lambda? Agile? Scrum? Kanban? OOP? SOLID? TDD? Unit-testing? Integration? End-to-End? Media Query? Responsive
Design? Serverless? Mobile-First? PWA? Rockstar? Ninja? Webmaster?

Software Engineer? Web Developer? Programmer?

Software Engineer? Web developer?

SOFTWARE ENGINEER

- A software engineer is someone who **writes computer code in order to build or maintain software.**

SKILLS THAT MAKE A GOOD SOFTWARE ENGINEER

- Knowledge of computer systems and programming languages
- Ability to work and learn autonomously
- Ability to solve unique unfamiliar problems.
- Ability to stay calm in the face of a complex problem.

WEB DEVELOPER

- A Web developer is a kind of **programmer who specializes in the development of applications** relating to the World Wide Web

DOMAINS

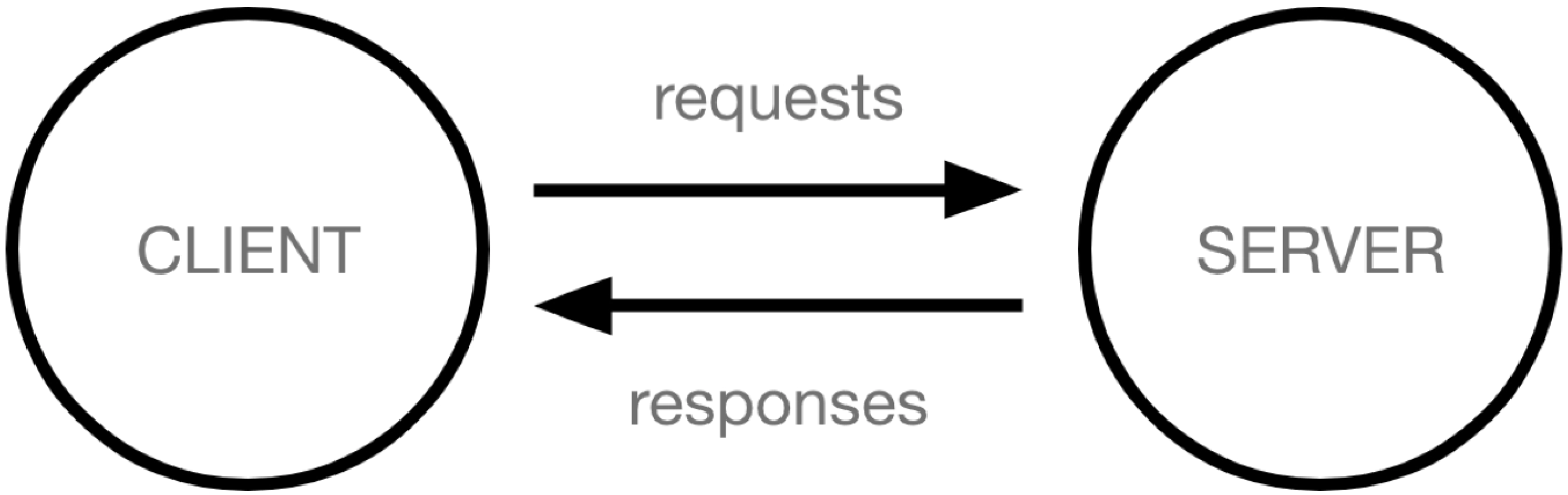
- Full-stack, Backend, Frontend, Mobile, DevOps, UI / UX

Brief History of the World Wide Web or Web

...30 years...

https://appinstitute.com/history-of-the-web-infographic/

How The Web works

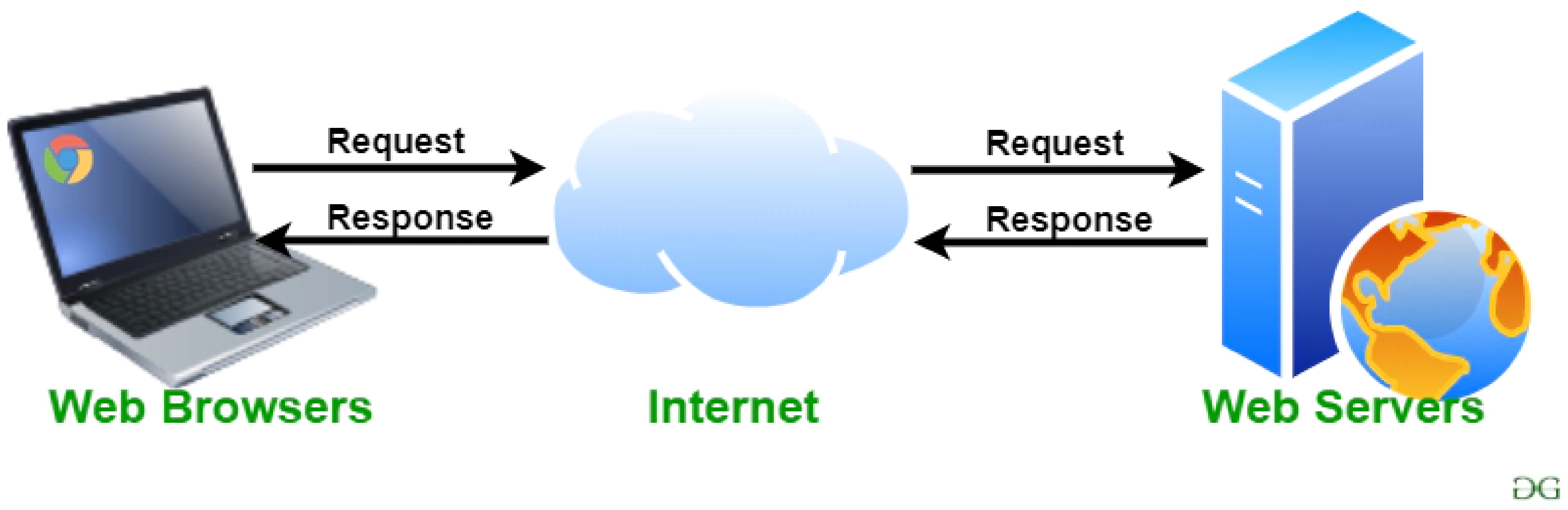


CLIENT

requests

responses

SERVER



How The Web works

CLIENTS

- User's internet-connected devices
  - For example, your computer connected to your Wi-Fi, or your phone connected to your mobile network.
  - **Web-accessing software** is available on those devices (i.e. Firefox, Safari)

SERVERS

- Computers that store webpages, sites, or apps.

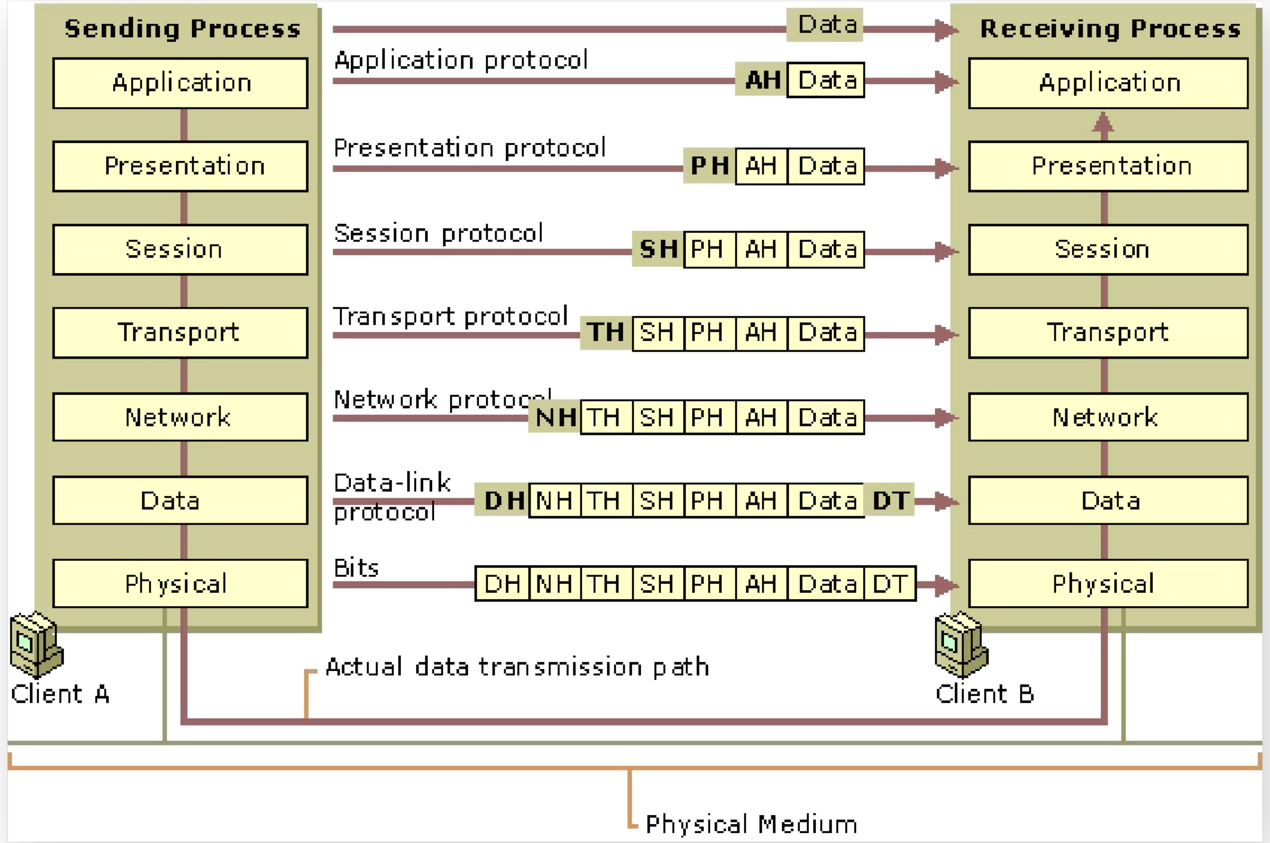
When a client device wants to access a webpage:

1. Copy of the webpage is downloaded from the server onto the client machine
2. Displayed in the user's web browser.

How the Web works

YOUR INTERNET CONNECTION

- Allows you to **send and receive data on the web**.



TCP/IP

- "Transmission Control Protocol and Internet Protocol"
- **Communication protocols** that define how data should travel across the internet.

DNS

- **Domain Name Servers** are like an address book for websites.
- When you type a web address in your browser, the browser looks at the DNS to find the website's real address before it can retrieve the website.
- The browser needs to find out which server the website lives on, so it can send HTTP messages to the right place.

How the Web works

HTTP/S

Hypertext Transfer Protocol is an application protocol that defines a language for clients and servers to speak to each other. This is like the language you use to order your goods.

COMPONENT FILES

- A website is made up of many different files, which are like the different parts of the goods you buy from the shop. These files come in two main types:

1. CODE FILES

- Websites are built primarily from HTML, CSS, and JavaScript, though you'll meet other technologies a bit later.

2. ASSETS

- This is a collective name for all the other stuff that makes up a website, such as images, music, video, Word documents, and PDFs.

The Technology Stack: Front-end

FRONT-END

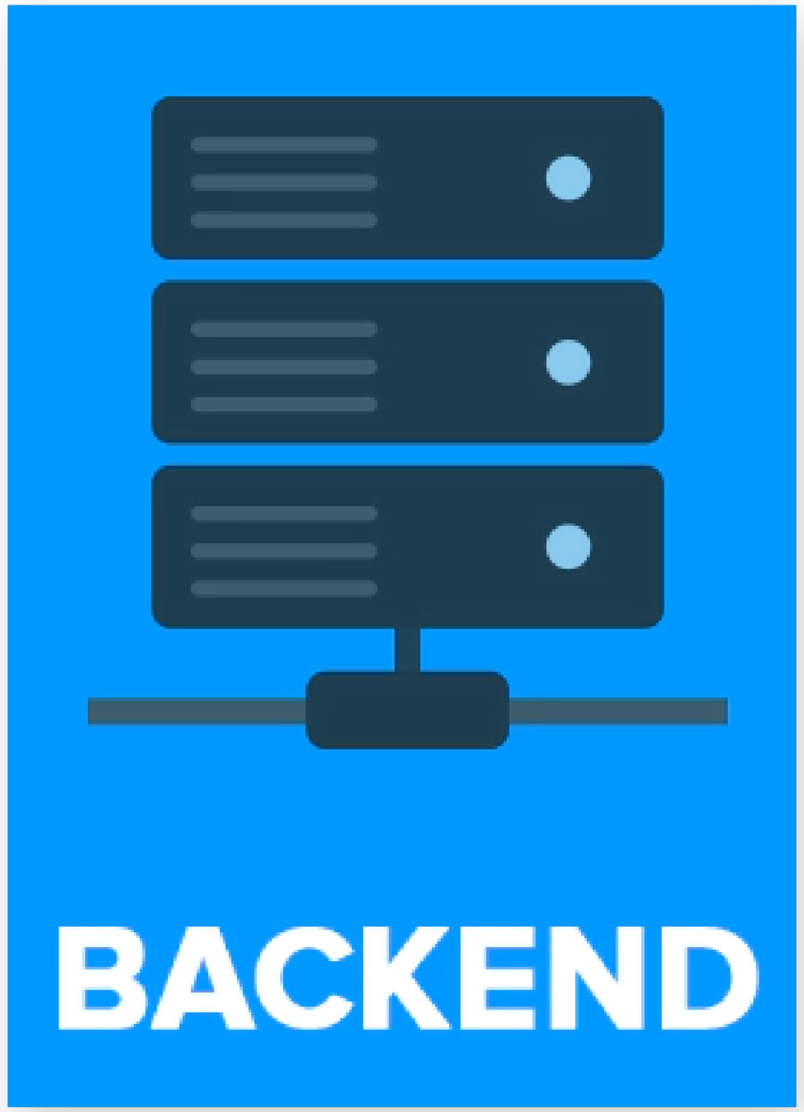
- Interface that a developer creates for **interaction** with the main application.
- Can be as simple as a **static web page** designed in HTML or a **complex implementation** done using a technology based on virtual dom (such as React.js)
- For browsers or a mobile application developed natively using android and iOS technologies or a hybrid platform such as React Native.

- i.e. HTML, CSS, JS, React.js, Vue.js, React Native

The Technology Stack: Back-end

BACK-END

- Server-side master program running on single or distributed topology and responsible for catering to the stateful and stateless requests sent from the front-end.
- i.e. Node.js, Python, PHP, Ruby, Java, Elixir



Programming

- Just like learning another language: knowing the syntax or grammar, and learning by doing.

HOW TO LEARN

- Docs, books, videos
- Tools: VSCode, Git, Browsers,
- Community: stackoverflow, reddit, avion people

MDN web docs moz://a

stackoverflow

YouTube

avion

Medium

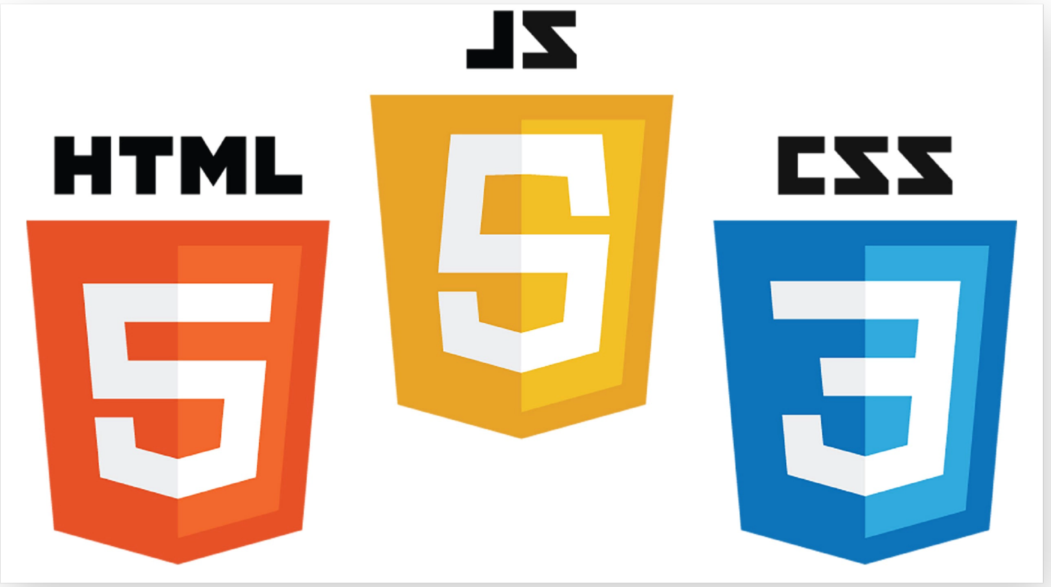
What will we learn

FRONT-END

- HTML5
- CSS3
- JavaScript

BACK-END

- Ruby on Rails

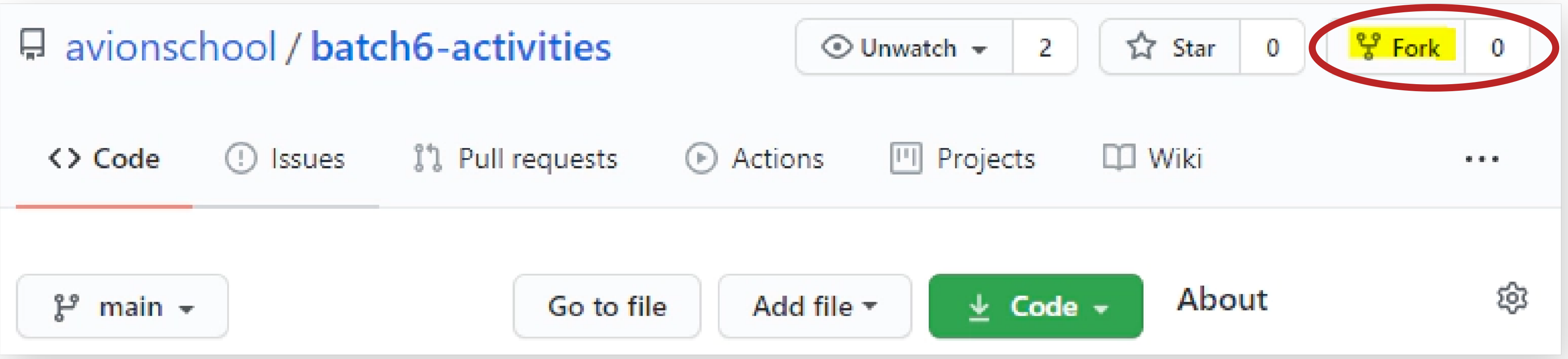


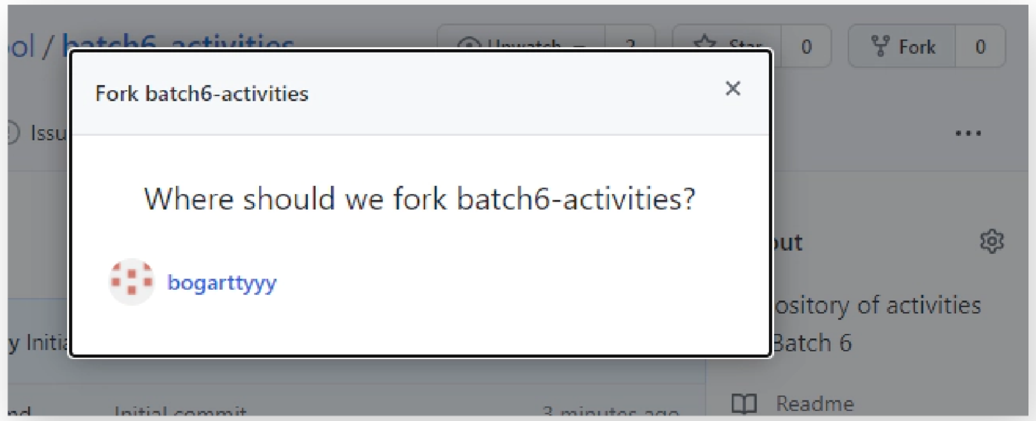
Github Pages

How to setup Github Pages

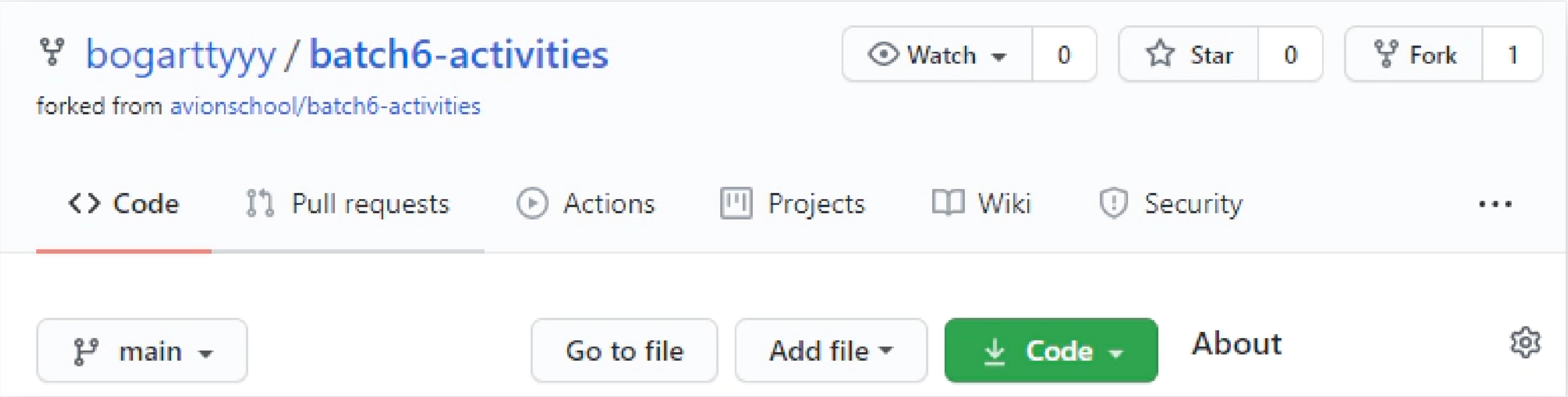
FORK EXISTING REPOSITORY

- Fork the **batch-activities** repository to your account
- You will be sent to the forked repository under your username. (i.e. bogarttyyy)
- *Link: https://github.com/avionschool/batch8-activties*




Fork batch6-activities

Where should we fork batch6-activities?

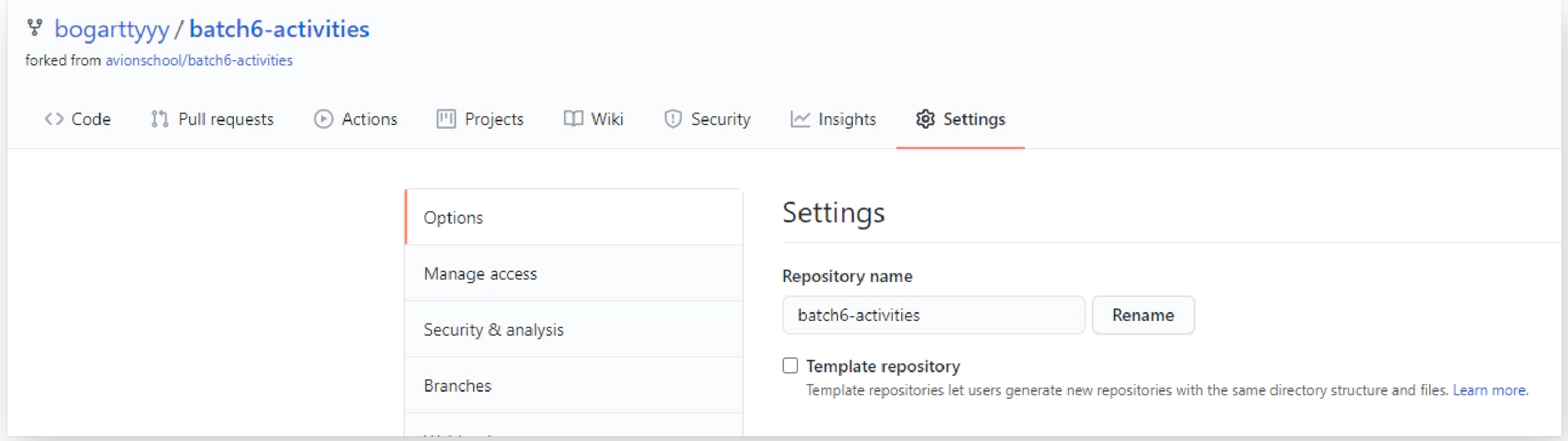


How to setup Github Pages

SETUP WITH EXISTING REPOSITORY

- Go to Settings in your forked repository
- Scroll down to Github Pages section
- Set the source from none to main and press Save
- Your Github Page is now accessible via: **<username>.github.io/batch8-activities**

Note: Github Pages automatically searches for an index.html or a README.md in your source branch (e.g. main) for it to display something when we access your github



bogarttyyy / batch6-activities

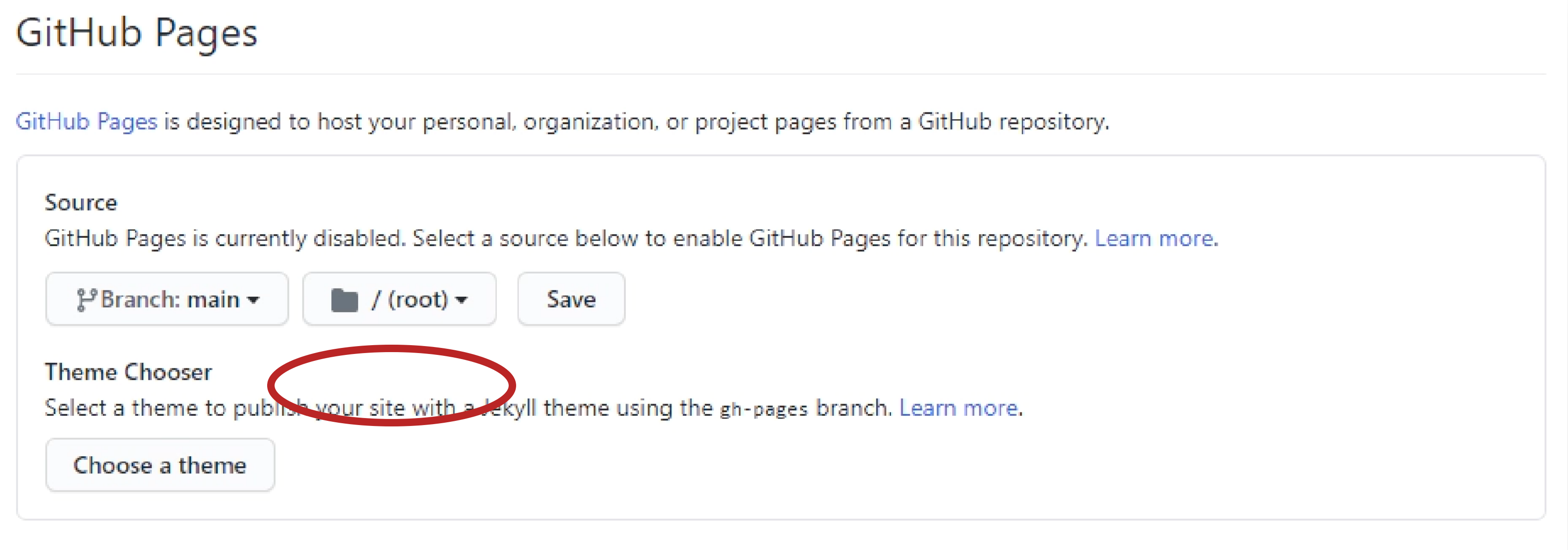
forked from avionschool/batch6-activities

Settings

Repository name: batch6-activities

□ Template repository

Template repositories let users generate new repositories with the same directory structure and files. Learn more.



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

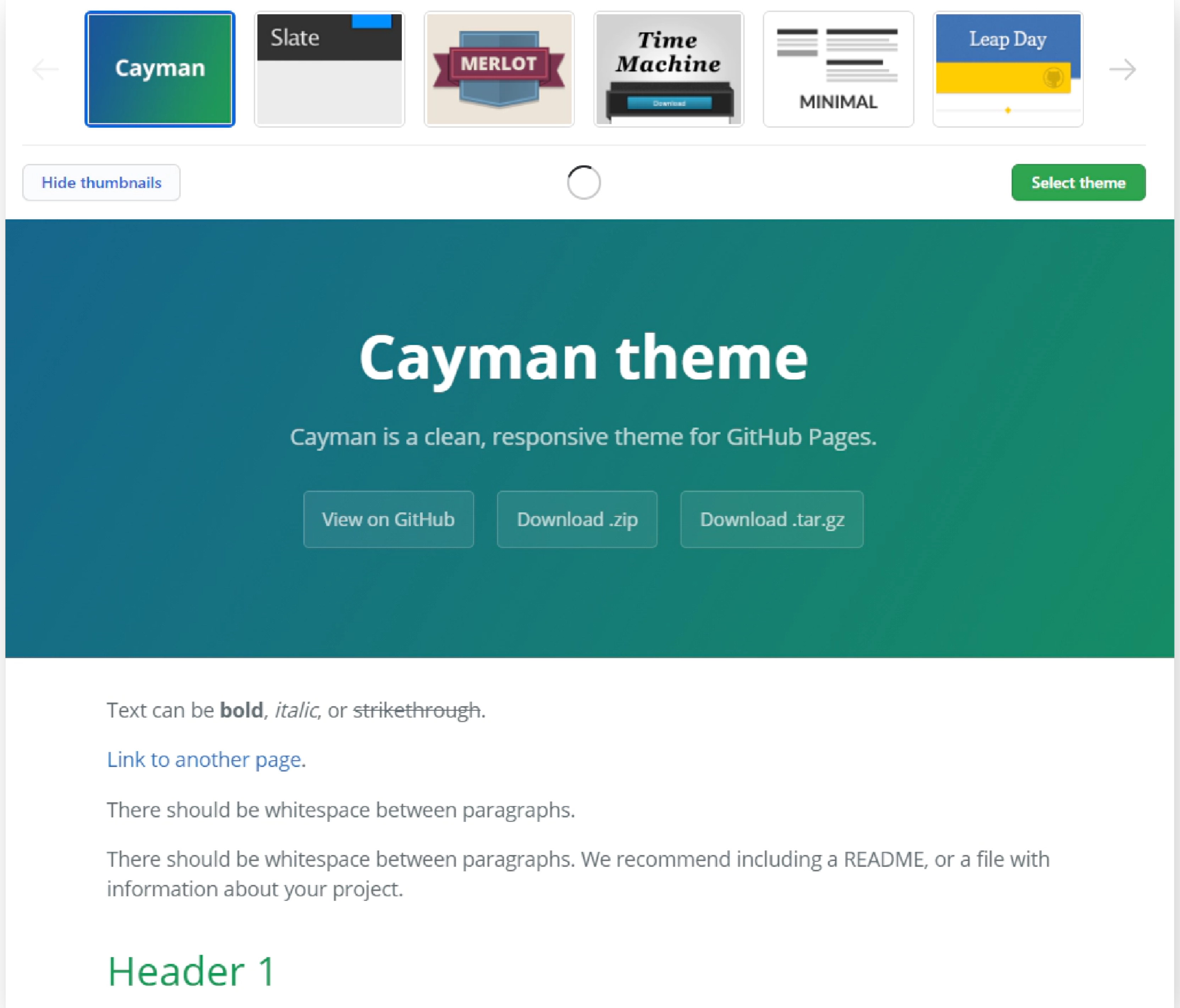
Theme Chooser

Select a theme to publish your site with a Jekyll theme using the gh-pages branch. Learn more.

How to setup Github Pages

CHOOSING A THEME

- If design isn't your strong suit, you may want to choose a theme for your Github page!
- We do recommend, and highly encourage you to create your own design
- After all, this can easily serve as your portfolio in landing that dream web developer job 😉



Cayman theme

Cayman is a clean, responsive theme for GitHub Pages.

Text can be **bold**, *italic*, or ~~strikethrough~~.

Link to another page.

There should be whitespace between paragraphs.

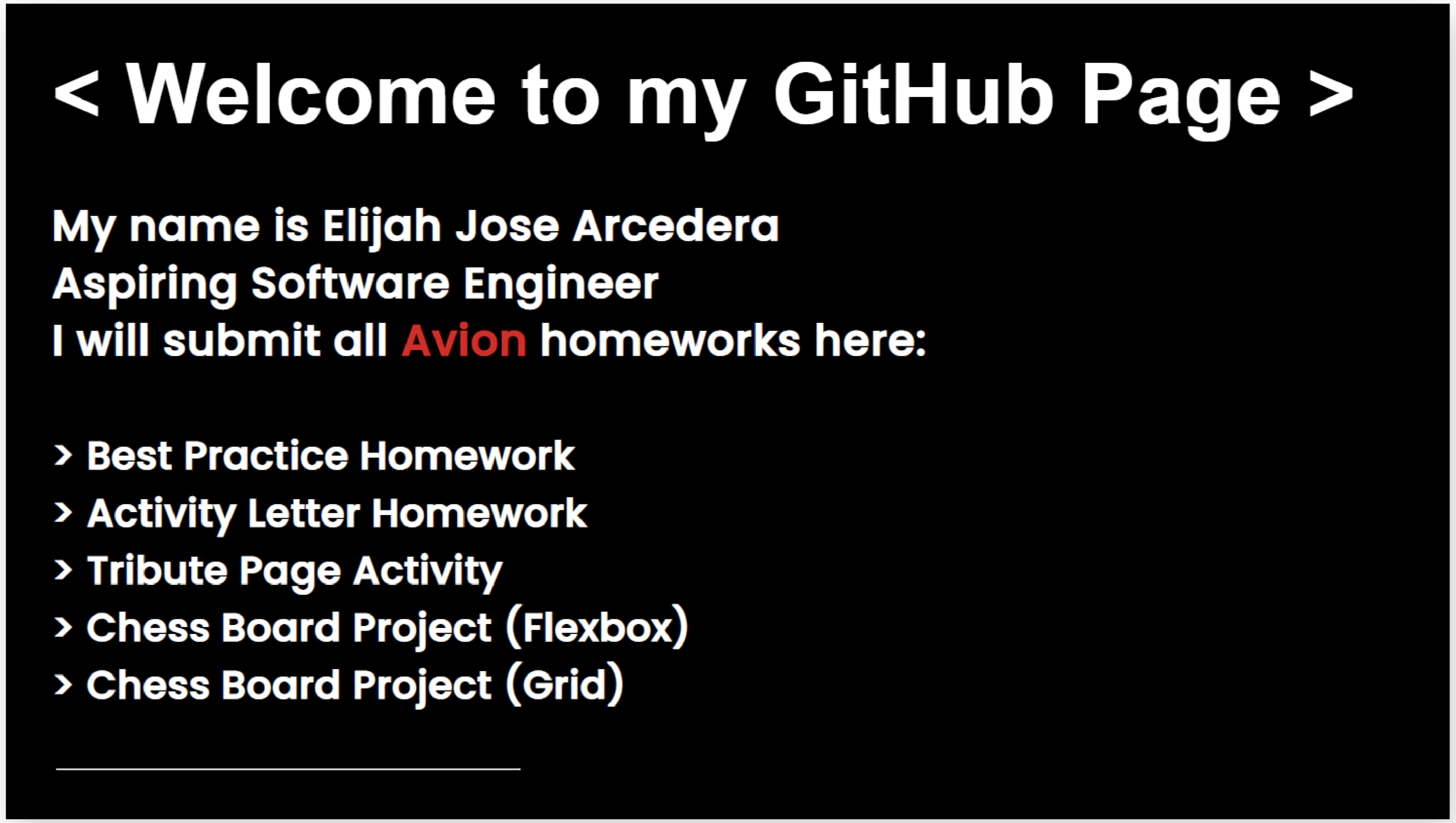
There should be whitespace between paragraphs. We recommend including a README, or a file with information about your project.

Header 1

How to setup Github Pages

DESIGN YOUR OWN PAGE!

- Here are some examples of custom designed pages from students here at Avion!



< Welcome to my GitHub Page >

My name is Elijah Jose Arcedera
Aspiring Software Engineer
I will submit all Avion homeworks here:

> Best Practice Homework
> Activity Letter Homework
> Tribute Page Activity
> Chess Board Project (Flexbox)
> Chess Board Project (Grid)

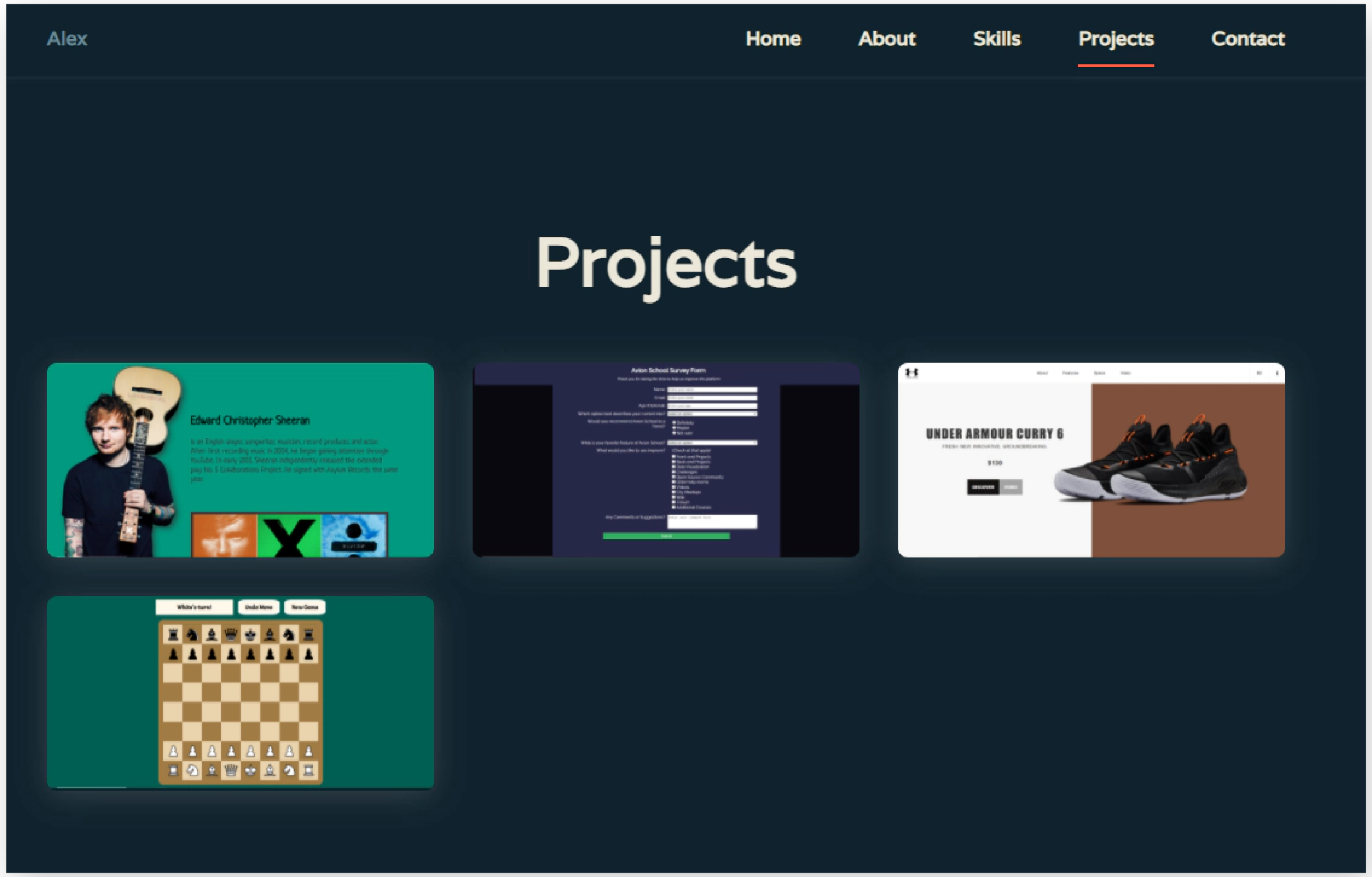
Elijah Arcedera - Batch 5

How to setup Github Pages

WHY YOU NEED A PORTFOLIO

- By setting this up, you'll be able to build your portfolio for hiring managers to review once you graduate.

- At the same time, your instructors can track your progress and provide feedback for improvements.



Alex Pasoquin - Batch 5

Assignment - Best Practice Page

1. Find a concept, an idea, or a thought online about a best practice related to web development or software engineering that can be summarized into a statement.
   a. Sources can range from a tweet, or a whole blog post, or from a book, etc.
2. Use the statement as a title, then add a description explaining it in your own words.
3. Present this using HTML & CSS on an **HTML page.**
4. Include the link of the source in the HTML page (however you choose to cite it).
5. Push your work to a new repo in your Github account, then send me the link in our Slack channel.
6. This is your chance to get adventurous and creative so we expect you to immerse yourself in HTML & CSS alongside web development concepts.

- An example: https://react-file-structure.surge.sh/