Senior Frontend Engineer Assignment (Commit Log)

Overview

For this assignment, your task is to build a web application screen that provides a visual GUI for software engineers who want to review the commit log for a given Git repository. Our design team has already created a design specification for this feature, and you can assume a backend engineer will build an API that will serve the necessary data. However, we need your help to build the frontend for this feature.

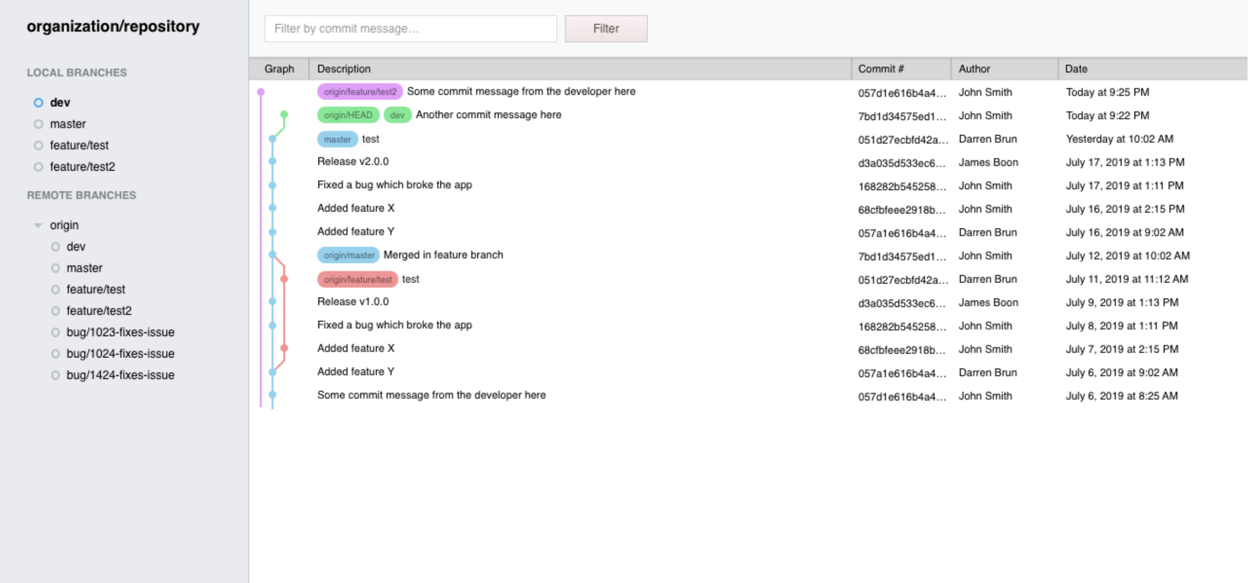
Requirements

- This will be a public web application. Steps should be taken to make sure it works in all modern web browsers, and at various screen resolutions.
- The app should follow the design guidelines below as closely as possible.
- You are free to use any technologies or third-party libraries you want. Describe your choices and decisions in a README file along with your submission.
- You are free to use existing boilerplates such as create-react-app, Next.js, or Gatsby however an important skill for this role is to have confidence setting up and maintaining things like Webpack and Babel. If you decide to use an existing framework please justify it in your README file and describe what (if any) configuration changes you had to make.
- You do NOT need to have your application connect to a real Git repository for its data. You can simulate or fake the commit log messages, branch names, authors, and other data. Focus on building the UI only. Assume there will be some API endpoint that will provide the necessary data. You can mock an API if you want.
- Most of the application is static with the exception of the filter toolbar, which should allow the user to filter the visible commit history items based on what

the user enters into the input field.

- You might have questions or concerns about some areas of this assignment. This is expected. Rather than reaching out to your recruiter for help, document your questions in your README file as well as how you decided to address or ignore those concerns.

Design Guidelines



organization/repository

LOCAL BRANCHES

- **dev**
- master
- feature/test
- feature/test2

REMOTE BRANCHES

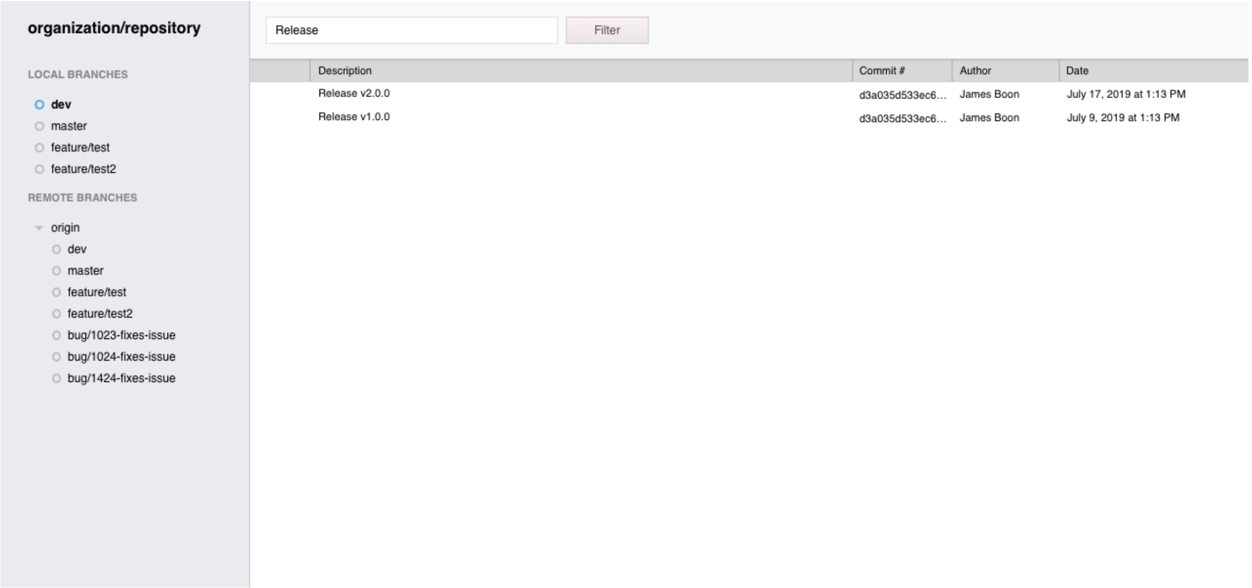
- origin
  - dev
  - master
  - feature/test
  - feature/test2
  - bug/1023-fixes-issue
  - bug/1024-fixes-issue
  - bug/1424-fixes-issue

Graph	Description	Commit #	Author	Date
	origin/feature/test2 Some commit message from the developer here	057d1e616b4a4…	John Smith	Today at 9:25 PM
	origin/HEAD dev Another commit message here	7bd1d34575ed1…	John Smith	Today at 9:22 PM
	master test	051d27ecbfd42a…	Darren Brun	Yesterday at 10:02 AM
	Release v2.0.0	d3a035d533ec6…	James Boon	July 17, 2019 at 1:13 PM
	Fixed a bug which broke the app	168282b545258…	John Smith	July 17, 2019 at 1:11 PM
	Added feature X	68cfbfeee2918b…	John Smith	July 16, 2019 at 2:15 PM
	Added feature Y	057a1e616b4a4…	Darren Brun	July 16, 2019 at 9:02 AM
	origin/master Merged in feature branch	7bd1d34575ed1…	John Smith	July 12, 2019 at 10:02 AM
	origin/feature/test test	051d27ecbfd42a…	Darren Brun	July 11, 2019 at 11:12 AM
	Release v1.0.0	d3a035d533ec6…	James Boon	July 9, 2019 at 1:13 PM
	Fixed a bug which broke the app	168282b545258…	John Smith	July 8, 2019 at 1:11 PM
	Added feature X	68cfbfeee2918b…	John Smith	July 7, 2019 at 2:15 PM
	Added feature Y	057a1e616b4a4…	Darren Brun	July 6, 2019 at 9:02 AM
	Some commit message from the developer here	057d1e616b4a4…	John Smith	July 6, 2019 at 8:25 AM

The screen contains three sections:

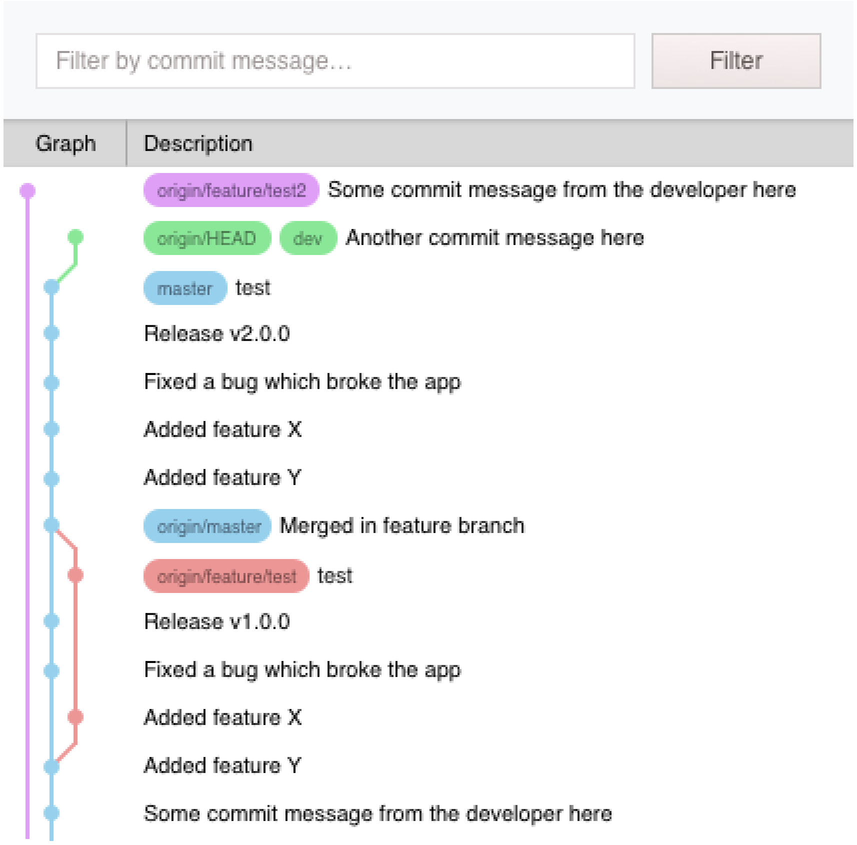
1. A left-hand sidebar which allows you to see a list of local and remote Git branches, as well as the name & organization of your repository. This area does not need to be interactive.
2. A top-head toolbar which contains a filter input and button which would allow a user to filter the commit history by a text string.
3. A commit table which shows a list of Git commits for that project with their description, commit hash number, author and date. A graph component shows a visual representation of the commits and branches in the project.

When a user types text into the filter input, it should immediately show the filtered results and hide the graph header and content, like this:



Description	Commit #	Author	Date
Release v2.0.0	d3a035d533ec6…	James Boon	July 17, 2019 at 1:13 PM
Release v1.0.0	d3a035d533ec6…	James Boon	July 9, 2019 at 1:13 PM

When the width of the browser screen is less than 800 pixels, the sidebar should disappear from view, like this:



Graph	Description
	origin/feature/test2 Some commit message from the developer here
	origin/HEAD dev Another commit message here
	master test
	Release v2.0.0
	Fixed a bug which broke the app
	Added feature X
	Added feature Y
	origin/master Merged in feature branch
	origin/feature/test test
	Release v1.0.0
	Fixed a bug which broke the app
	Added feature X
	Added feature Y
	Some commit message from the developer here

Note: All of the designs above are provided to you via Figma as well. Visit this link and create a Figma account (if you don't already have one). After you've logged in, you should have full visibility of the design assets. This will allow you to inspect sizes, colors and positions of all elements.

Submission

- Include a README file with your solution that describes:
    - All necessary instructions to install and run your application
    - What technologies and third-party libraries you decided to use any why
    - What improvements you could have made to your solution if given more time
    - *(Optional)* Any questions you had along the way and how you decided to address them
    - *(Optional)* What was something new you learned while doing the assignment
    - *(Optional)* Any other notes you want to share with the review team
- Upload your submission as a new GitHub repository in the ZenHub organization. Your recruiter will provide instructions on how you can get access to it. Send your recruiter an email with a link the repository.
- You have one week to work on the assignment as you see fit. You can work as much or as little as you feel is necessary.
- You will be asked to walk through your solution and code in your next interview, please have it ready to be run prior to the interview.

Evaluation

You will be judged on:

- Completeness of all requirements and design specifications
- Documentation (the attached README file)
- Support for desktop and mobile browsers (IE support is not needed)

- General code quality and clarity
- *Bonus points:* Include tests to ensure functionality works. 100% coverage is not required or expected.

These instructions do not specify every possible detail you should take into consideration. This is done intentionally to test your ability to analyze the problem and come up with a creative solution. Thoroughness and attention to detail are two of the most important qualities for this role.