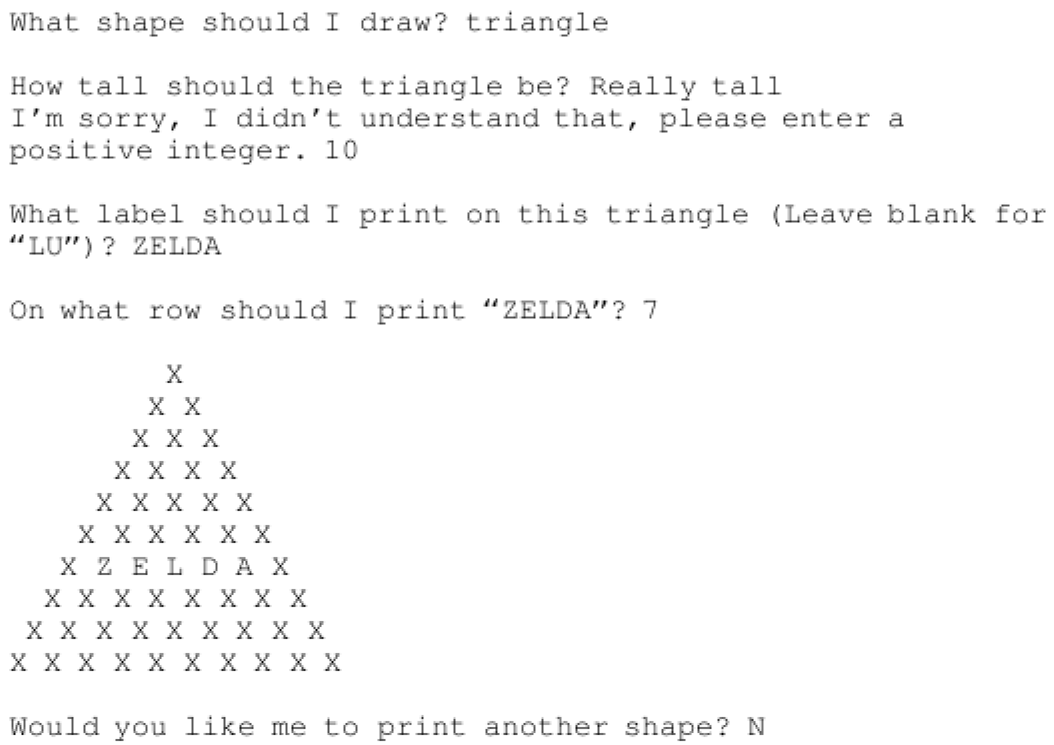
In order for us to get a better grasp on your programming abilities, please choose and **complete two of the four tasks** on the following pages. They’re fairly simple, but they allow our team to get a glimpse of how you solve problems and implement solutions. Feel free to add additional features. “Scope creep” is encouraged if it enables you to better highlight your coding proficiency. Have fun with this, and show us what you can do.

1. Write a simple Java or C# program that draws ASCII shapes
    a. It should take at least these inputs, but it could take more. It should be almost impossible to crash the program through bad input.
        i. Which shape to draw – the choices should include triangle, square, and diamond.
        ii. Height of the shape
        iii. Text label to display
            1. Choosing the label should be optional for the user.
            2. Default: “LU”
        iv. Row on which to display text label
            1. Choosing the label row should be optional for the user.
            2. Default: 4
        v. The option to repeat at the end.
    b. A sample run might go something like this:



```
What shape should I draw? triangle

How tall should the triangle be? Really tall
I’m sorry, I didn’t understand that, please enter a
positive integer. 10

What label should I print on this triangle (Leave blank for
“LU”)? ZELDA

On what row should I print “ZELDA”? 7

         X
        X X
       X X X
      X X X X
     X X X X X
    X X X X X X
   X Z E L D A X
  X X X X X X X X
 X X X X X X X X X
X X X X X X X X X X

Would you like me to print another shape? N
```

    c. Possible Bonus Tasks:
        i. Allow multiple text labels on multiple different lines.
        ii. Add more options for shapes. (square, circle, rectangle, smiley face ☺, etc.)

2. Design a brand new database that would store information about people. The expected number of people to be stored in this database is over 1,000,000. You need to store the following information:
    a. SSN
    b. Name
    c. Birth date
    d. Ethnicity
    e. US Address Info (needs to store several: home, work, etc.)
    f. Phone Info (needs to store several: home, work, etc.)
    g. Donation Type (Cash, Check, Credit Card, etc)
    h. DonationDate
    i. DonationAmount
    j. DonationMemo

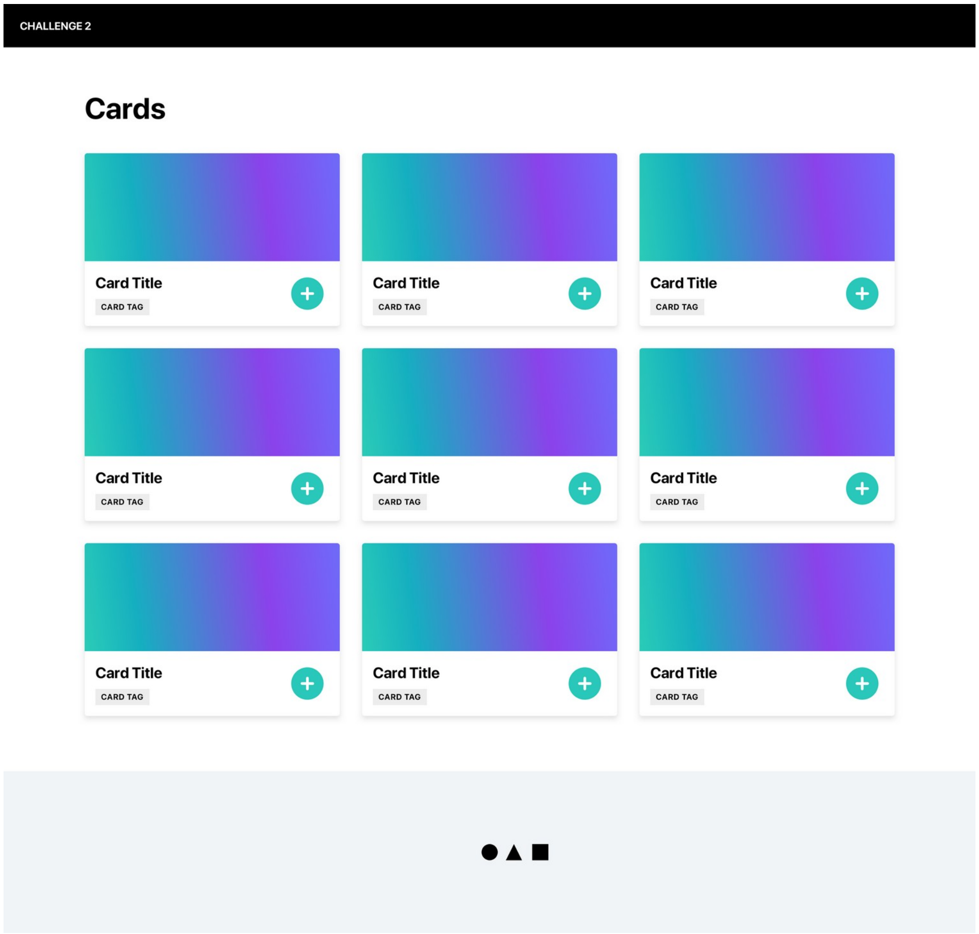
    Note: A single person may have zero, one, or several donations; but each donation has only one Type, Date, Amount, and Memo.

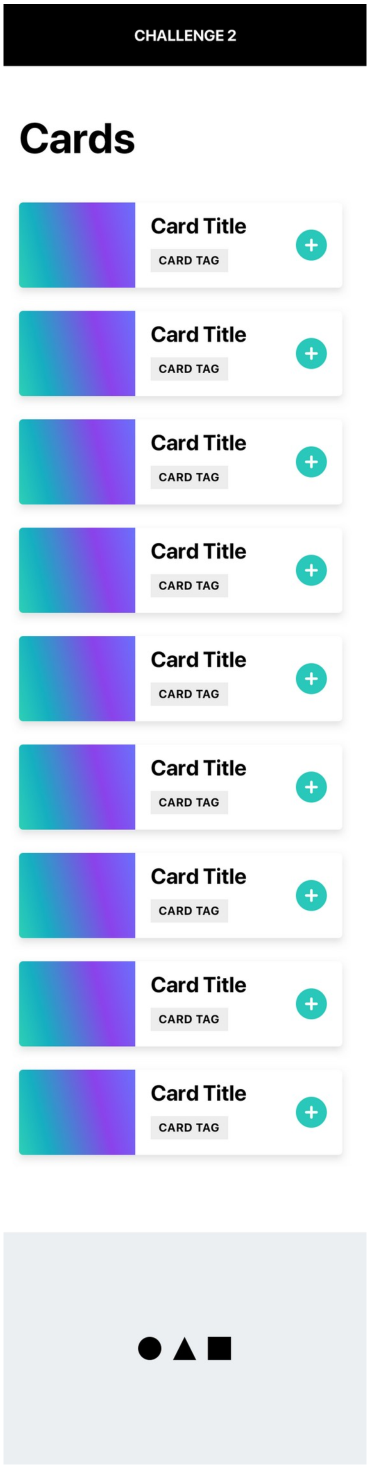
    Bonus: write the SQL to create these tables in Oracle or MS SQL Server.

3. Implement a responsive UI based on the designs below. This should follow the designs as close as possible, while fluidly adjusting through a range of viewport sizes from phone to a large desktop view. You do not need to interact with any APIs for data. You should use placeholder text and images where needed.

Desktop View:

Mobile View:





Each card on the screen contains the following:

a) a gradient image (you can use any image you'd like)

b) a title

c) a tag

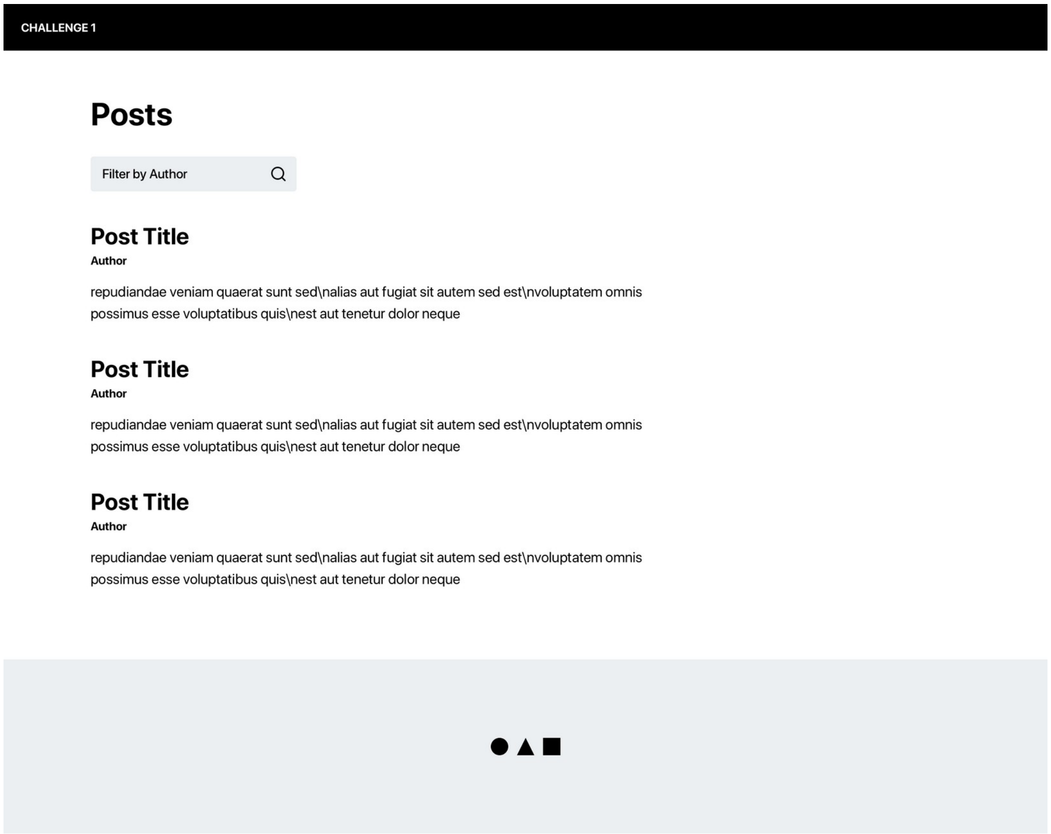
d) a plus button on the right

Also, be sure to include the header and footer on the page as well.

4. Write a web application that displays data from https://jsonplaceholder.typicode.com. This application may be written using JavaScript or TypeScript, HTML, and CSS/less/SCSS. You may use any web framework you would like, or none at all.

The application should display various posts from the documented API, showing the title, author, and body. If the user enters a numeric user id into a filter field, the displayed posts will be filtered down to those with that user id. Use the sample Desktop view below as a guideline of what your app should look like.

Desktop view:



Posts

Post Title

Author

repudiandae veniam quaerat sunt sed\nalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dolor neque

Post Title

Author

repudiandae veniam quaerat sunt sed\nalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dolor neque

Post Title

Author

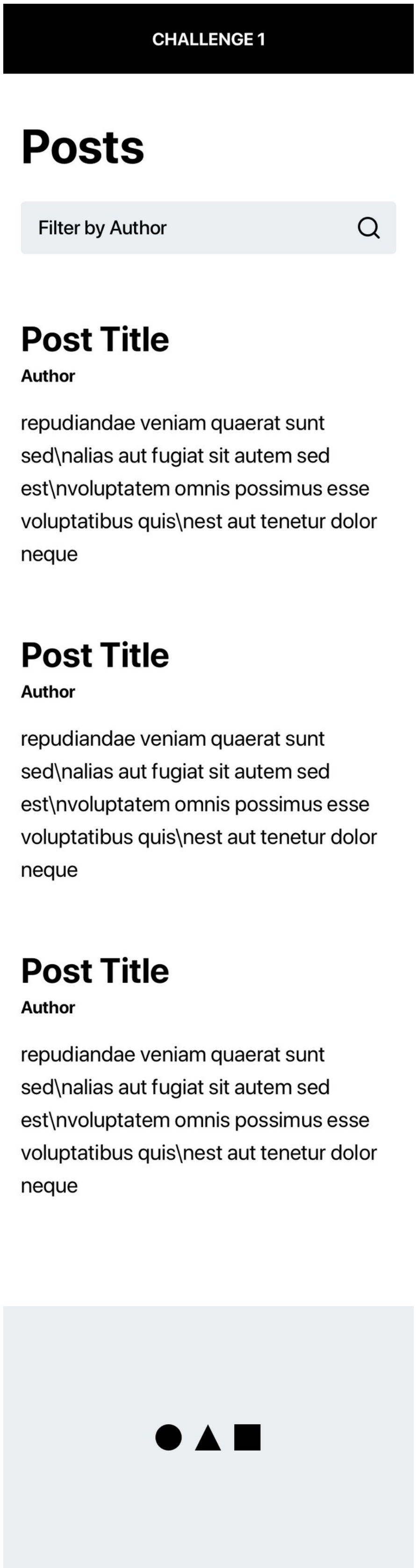
repudiandae veniam quaerat sunt sed\nalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dolor neque

Possible bonus tasks:

a) Initially fetch only 10 posts. Load more either with a button that the user can click, or as the use scrolls down the page.

b) Style the page in a better way that can showcase your experience with design and styling. This is similar to the request in problem #3, so **if you choose to implement the same requirements here, this will count as the two problems required for your submission.** You can reference the mobile view on the next page as an option for responsive design.

Mobile view (bonus):



CHALLENGE 1

Posts

Filter by Author

Post Title

Author

repudiandae veniam quaerat sunt sed\nalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dolor neque

Post Title

Author

repudiandae veniam quaerat sunt sed\nalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dolor neque

Post Title

Author

repudiandae veniam quaerat sunt sed\nalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dolor neque

● ▲ ■