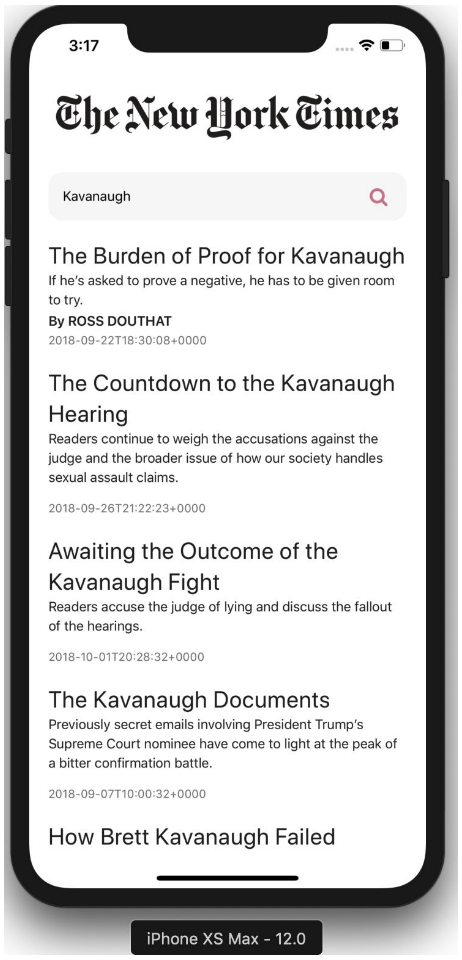
Assignment 5: New York Times 2.0

Due Tuesday, November 6th, at 11:59 PM.

Overview

After implementing the simple New York Times app, you found that there's actually a demand for such a news browser. You meet up with your design team and together you decide that you want to make your app more user friendly. As a result, you add an onboarding screen, and a screen to open the article within the app instead of Linking to the default browser. Your design team challenges themselves by building the initial screens; however, they need your help connecting and making them work as expected. It is now your job to take their project files and turn them into a functional single app using `AsyncStorage` and a `StackNavigator`.

your final product will look something like this:



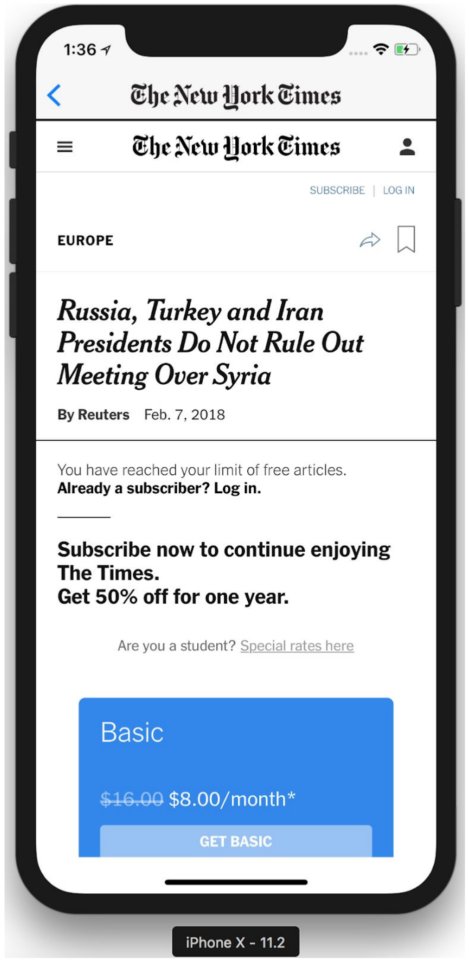
The Burden of Proof for Kavanaugh
If he's asked to prove a negative, he has to be given room to try.
By ROSS DOUTHAT
2018-09-22T18:30:08+0000

The Countdown to the Kavanaugh Hearing
Readers continue to weigh the accusations against the judge and the broader issue of how our society handles sexual assault claims.
2018-09-26T21:22:23+0000

Awaiting the Outcome of the Kavanaugh Fight
Readers accuse the judge of lying and discuss the fallout of the hearings.
2018-10-01T20:28:32+0000

The Kavanaugh Documents
Previously secret emails involving President Trump's Supreme Court nominee have come to light at the peak of a bitter confirmation battle.
2018-09-07T10:00:32+0000

How Brett Kavanaugh Failed



Russia, Turkey and Iran Presidents Do Not Rule Out Meeting Over Syria
By Reuters Feb. 7, 2018

You have reached your limit of free articles.
Already a subscriber? Log in.

Subscribe now to continue enjoying The Times.
Get 50% off for one year.

Are you a student? Special rates here

Design Details

You've seen the screenshots above. You probably already started thinking about how to layout most of the particular components. For this assignment, you won't be implementing much of the visual elements. However, feel free to change things to look however you would like, just make sure to preserve the general feel of the starter code. Here's a breakdown of what we want to see:

Feature	Description
Visual Requirements	● Lean towards matching the content in the screenshots.
Navigation Bar	● You must use a `StackNavigator` to go from the main screen (which shows the category icons) to the specific article viewing screen (which appears once you tap on a news item). Your navigation bar will show up automatically once you do that, though you might have to change your navigator's `headerMode`. ● Your navigation bar must have the New York Times logo in both screens. Make sure that its dimensions are the same in both screens.
Article Screen	● The article viewing screen only has a single `WebView`. In the starter code, we have provided an example of how to open a specific webpage using a URL.
Tap to View	● Tapping on an article should *navigate* the user to the article viewing screen and pass down the article's URL as a prop to be viewed in a `WebView`.
Onboarding Screen	● Build your own onboarding screen that provides a bit of info about the app. ● The onboarding screen should be shown only the first time that the user opens the app. Make use of `AsyncStorage` here.

Implementation Details

Here's a step-by-step guide of your app's implementation. You will mainly be dealing with 3 parts: the onboarding screen, navigation, and with saving the last screen that the user had open.

1. Onboarding Screen

In terms of design, any simple screen would suffice. However, the app always loads the onboarding screen upon opening it. The desired functionality is for the onboarding screen to load only the first time that the user opens the app. You will find it useful to check where we use `AsyncStorage` in the first navigation lecture.

2. Navigation:

You will need to implement navigation to go from the main screen (which shows the category icons) to the specific article viewing screen (which appears once you tap on a news item) and back. The gist is that when you tap on a news item, a separate dedicated screen must process the information for that article. This separate screen will be an article viewing screen. For our purposes, opening the article URL in a `WebView` is enough. Use a `StackNavigator` and note the following when achieving the desired functionality:

a. `this.props.navigation.navigate(nextScreen)` is the function that triggers a switch from the current screen to the given next screen. This function is part of the `props` of the screen that is defined in the `StackNavigator`, and not of its children. For instance, if I have a screen called `Main` and I use its render() function to render a `<News>` component within, then ONLY the `Main` component will have access to the navigation `props`, NOT the `News` component. As a result, calling it using the same syntax from the `News` will cause your app to crash. You will need your `News` component to tell its parent, `Main`, to change the screen to the next screen.

b. (This will be shown in lecture on Thursday) To pass data from a screen to another, you can use the function above in the following way: `this.props.navigation.navigate(nextScreen, data)`. Where `data` is a standard JS object. Once this function is executed, you can access your `data` by using the following line in the new screen: `this.props.navigation.state.params` — from there, you can access the keys of your `data` object, and their assigned values.

Grading

Grading is based on the specifications above. Like last time, we will be rewarding bonus points for novel solutions and extensions.

Onboarding Screen	3 points
Navigation	4 points
Total	**7 points**

Extensions

- Create a custom onboarding screen with your own style! (+1 point)
- Create a custom articles viewing screen. As per the current spec, you only have to implement a WebView. You can get +2 extra points if you implement a custom component, instead, that displays the article's information. You might have to tweak the `APIRequest` class to obtain more details from the NYT API.
- Preserve last screen before app closes. Combine the use of componentWillUnmount and AsyncStorage to get the user immediately to the last screen they were on last time. If they were reading a specific article, you should open the article view to that article as soon as they launch the app again. +3 points

Resources

Start the assignment from your finished assignment 3.

Submission

When you have completed your assignment, please remove the `node_modules` folder and package the entire project that you worked on in a .zip file. Please submit through Canvas.