Web Developer Interview Task #013 SimilarGroup

WEBSITES ANALYSIS APP

Create a single-page web application, for displaying websites information.

The app should contain an **input field for URL**, a **list of previous URL searches (history)** and the **main view**.
The **main view** is a visualization of the data provided by SimilarWeb's API and should be consisted of the following:

a. **Header** with website's Title, fav-icon and its URL address, the website's category, its global rank and its rank within that category.
b. **Similar sites section** – a list of websites, displaying their fav-icons and URLs. Initially, only the first 12 items should be displayed, with a "show more" link/button to expand the section and reveal the full list (when there are additional results).
c. **An IFrame element**, showing the selected website itself (see notes below).

This is also visualized on the attached wireframes.

Additional Requirements

1. The application should consume data using AJAX requests to URLs of the scheme - http://api.similarweb.com/site/www.website.com/rankoverview?userkey=a6fd04d833f2c28ce7c30dc957bf481e
   It responds with an "`access-control-allow-origin: *`" HTTP header
2. Your webapp layout should match the attached wireframes. *Initial state* (before any website search) and *empty state* (when website is unknown or not exist) wireframe are also available.
3. It should be possible to navigate through the recent website's searches (on the same visit), *using the back/forwards browsers' buttons*.
4. URLs history list
   a. When searching for a website, its URL should be inserted to the top of the history list in an "active" (selected) state.
   b. When clicking on an item (URL) from that list, it should get an "active" (selected) state, and the application should load and display its website's information in the main view.
   c. The list items should be persistent, should stay on page refresh.
   d. Each search history item should also contain an "x" button, to remove that item from the list.

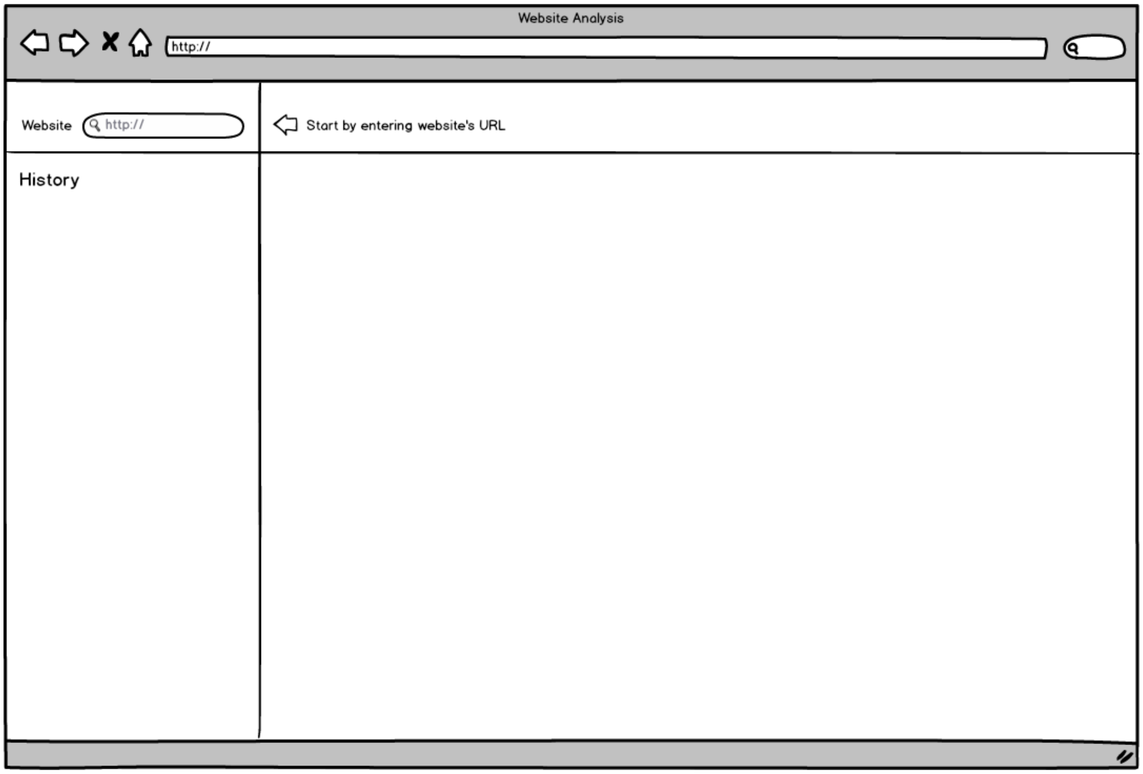
Notes

1. Some website (including most of the big ones – google.com, facebook.com, etc) *won’t load in an IFrame* due to their “`x-frame-options`” header limitations (on supported browsers) and some others will “*unframe*” themselves by navigating the top frame (your webapp) away to theirs site. *Ignore these behaviors and try other sites.*
2. Though the webapp itself is fully front-end, *you’ll may need to load it via HTTP protocol* in order for things to work as expected. To do that, you can use any HTTP server you like (node-static is a simple option).
3. No need to especially support old browsers (< IE9).

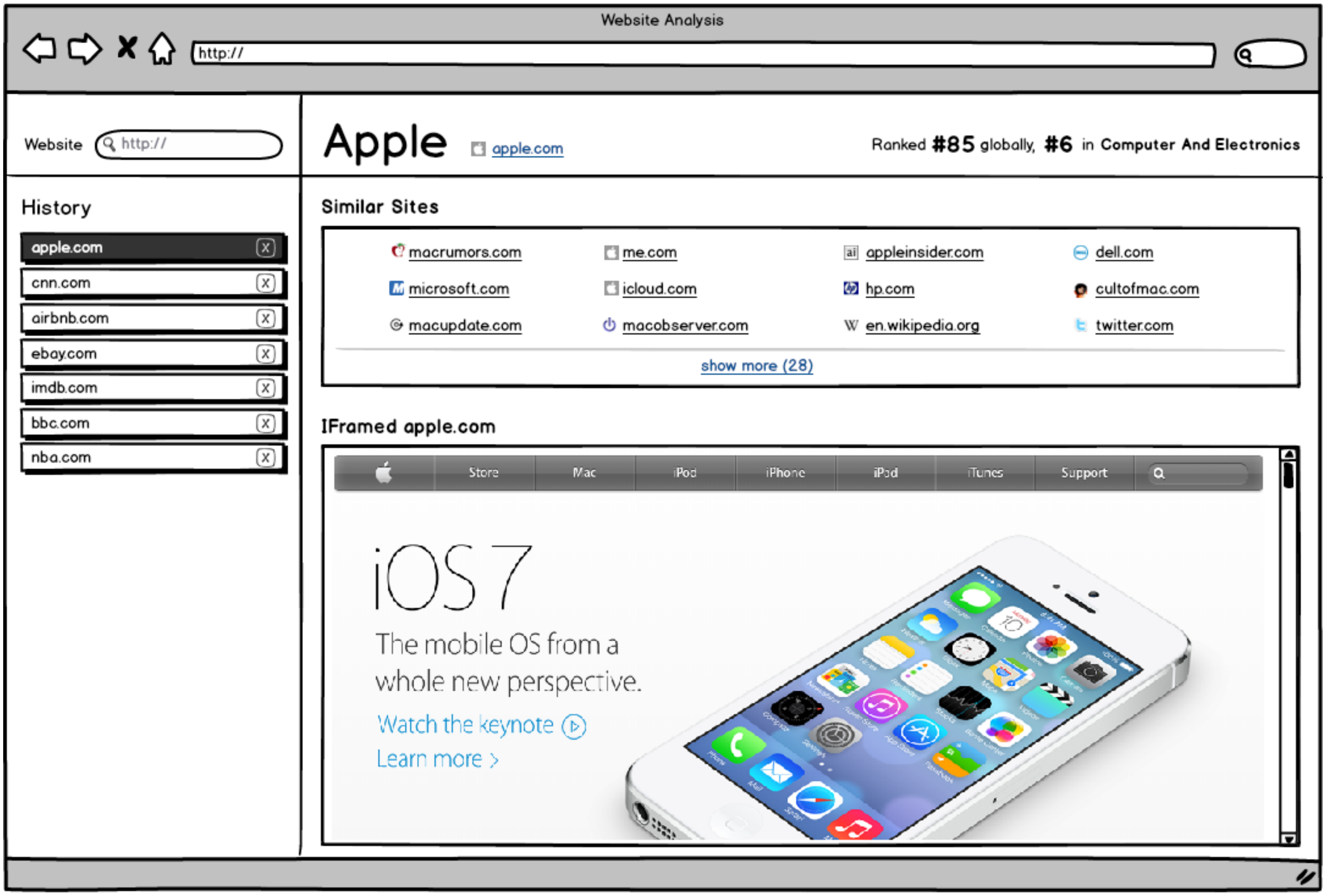
Time matters, start with the requirements before doing optional improvements.
Use any tools, frameworks, libraries and plugins of your choice, as you find suitable for this kind of project. Keep in mind it is a small “project”, therefore heavy infrastructure and some optimizations are not a must.

Wireframes

Initial state



Apple.com’s website information (with history)



History

- apple.com (X)
- cnn.com (X)
- airbnb.com (X)
- ebay.com (X)
- imdb.com (X)
- bbc.com (X)
- nba.com (X)

Apple

apple.com

Ranked **#85** globally, **#6** in **Computer And Electronics**

Similar Sites

macrumors.com	me.com	appleinsider.com	dell.com
microsoft.com	icloud.com	hp.com	cultofmac.com
macupdate.com	macobserver.com	en.wikipedia.org	twitter.com

show more (28)

IFramed apple.com

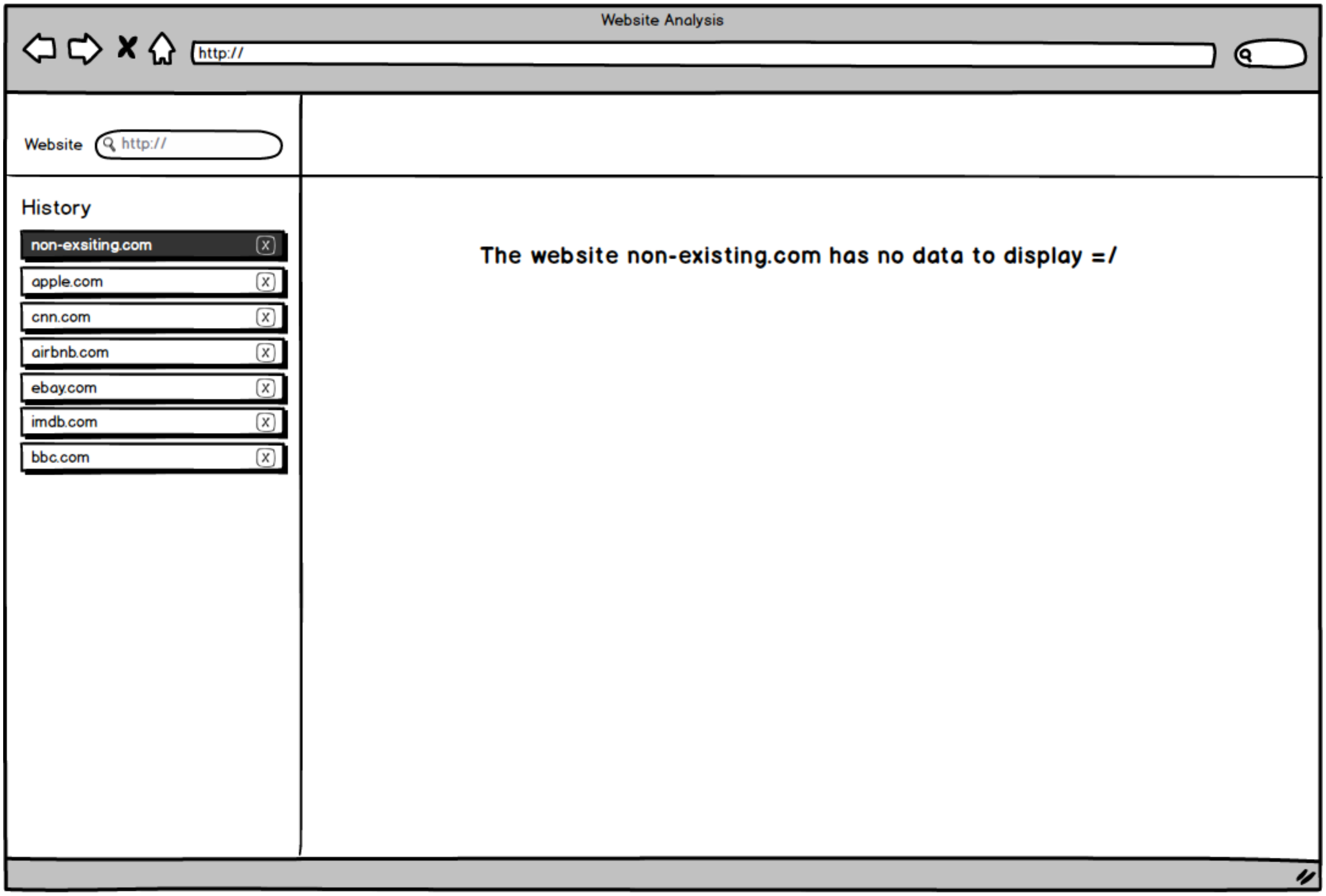
iOS 7

The mobile OS from a whole new perspective.

Watch the keynote

Learn more

Empty state (when there’s no data / getting bad server response)



The website non-existing.com has no data to display =/

- Please keep this document confidential -