

Develop a pseudo Front End Application which would let the users help list and browse top Hacker News articles conveniently.

- Use of Web API to fetch news details.
- Utilise Response Parameters [ id,title, url, num\_points, num\_comments, author, created\_at]
- Implement functionality to list articles in a web page.
- Visually interactive responsive design listing all the articles.
- A feature to Search by name. Store the API response in local storage and make queries to that.
- Implement Sort feature (to filter result through 'number of points' in ascending and descending order). Again, you should be searching local storage for this and not making fresh API calls.
- href url parameter. All headings should be links which should open in a new window. (Should open in new window | target="\_blank")

#### Plus Point (Good to have)

- Implement features to sort articles based on dates/time. (Sort based on old-to-new and new-to-old based on time).
- Add autocomplete features to *Search* articles conveniently . Search local storage where you have stored the API Response.
- Custom elegant design, fonts and icons to make web app more user-friendly.
- You may add portfolio activity comprising awesome work you have done on web application(s).
- Use your imagination and add features which would make things easier for end users

#### Guide

- Articles API: <https://github.com/HackerNews/API>
- HTML5 Local Storage: <http://html5demos.com/storage>
- IndexedDB API  
[https://developer.mozilla.org/en/docs/Web/API/IndexedDB\\_API](https://developer.mozilla.org/en/docs/Web/API/IndexedDB_API)
  - [https://www.tutorialspoint.com/html5/html5\\_indexeddb.htm](https://www.tutorialspoint.com/html5/html5_indexeddb.htm)
- For icons and subtle texture
  - <http://glyphsearch.com/>
  - <http://subtlepatterns.com/thumbnail-view/>
  - <http://codebeautify.org/jsonviewer>
  - <http://www.flaticon.com/categories>
- Structure / Data Parameters
  - id,title, url, num\_points, num\_comments, author, created\_at

- id — 'id'
- title — title of the article.
- url — url of the article.
- Num\_points — number of points of an article.
- num\_comments — number of comments on an article.
- created\_at — time of the article posted

## Deliverables

- Implementation should be functional and without errors. Please test it before submission.
- Assignment will be reviewed on both the UI design and implementation.
- Coding aspect that will be looked into: indented, commented, and documented code.
- Make sure input/form validations are done when required.
- Zip all your Source Code, Deployment Instructions, Screenshots and upload.