**The Technologies Plan**

For our app, we will be coding in…

* **React** because
  + We want our users to use our app on all the platforms that support web app
  + It’s easy to make React app a PWA, making it easier for mobile users
  + Our service is highly based on online communication with our backend server, so users should have a network connection when using the app, which combined with the reasons above means a JavaScript-based web app is the best choice
* **Node.js** because
  + Our service is heavily based on the communication between the front-end and back-end server. This is a web-app
  + Users should be able to store and sync their data across multiple devices.
  + Since we have already decided to learn React.js, which is written in JavaScript, it takes less time for us to pick up another thing in the same language
  + Node js is famous for its rich ecosystem, which means we can almost find all the packages we need, as will reduce the efforts and speed up the development
  + Based on the APIs we have chosen, the majority of them have clear and compelling official docs for Node.js
  + Users do not get exposure directly to the backend, so choosing the way that is easiest to develop is the right choice
  + If in the future, we find a way to eliminate the need for a backend-server, we can easily migrate those services to **Firebase Cloud Function**, which is also in JavaScript
* **MySQL** because
  + We would like to provide our users with the features including but not limited to the online storage for the documents they uploaded, the flashcards generated in the previous sessions, associated tags with each flashcard-set, sharing and accessing data with other users, looking up the generated timeline and so on
  + We are planning to use a relational database due to the strong relations among our data, e.g. the ownership of flashcards to each user, the relationships of flashcards to the documents they belong to etc.
  + And MySQL, as one of the most famous such databases, becomes the best choice without questions
* Packages
  + **Wikijs**
    - <https://www.npmjs.com/package/wikijs>
    - We need to communicate with Wikipedia to fetch the definition of the term, which is the selling point of this app (flashcards for difficult terms in uploaded documents)
    - Easy to use and popular and is updated frequently
    - MIT License
  + **Natural**
    - <https://www.npmjs.com/package/natural>
    - We need to use natural language processing packages to process the documents uploaded by the users, such as picking up the difficult terms by removing “stop-words” and words with very high frequency and so on
    - This package is very popular (around 7k downloads per week) and it’s using MIT License
  + **rc-calendar**
    - <https://www.npmjs.com/package/rc-calendar>
    - React calendar components
    - We would like to give the users some options to retrieve the events on a specific date
    - And we are using React
  + **React timeline**
    - <https://www.npmjs.com/package/react-time-line>
    - The users should be able to see the events on a timeline
    - We are using React
  + **Pdf2json**
    - <https://www.npmjs.com/package/pdf2json>
    - We will use this package to parse the pdf documents uploaded by the users
* APIs
  + **pusher.com**
    - We would like our users to share and thus it would be great if they can chat with each other
    - pusher.com has in-app chatting APIs we can use
    - And pusher.com offers free plans that match our needs at this stage
  + **DictService API**
    - <http://services.aonaware.com/DictService/DictService.asmx>
    - We will lookup terms using this API and return the definition to the users
  + **Oxford dictionaries API**
    - <https://developer.oxforddictionaries.com/>
    - We will lookup terms using this API and return the definition to the users
  + **Merriam-Webster Dictionary API**
    - <https://www.dictionaryapi.com/>
    - We will lookup terms using this API and return the definition to the users
  + **Google Analytics API**
    - <https://developers.google.com/analytics/?hl=en>
    - The usage statistics will be collected and analyzed using this API
    - We will use the results to improve our service to the users
  + **Ink-recognizer from Microsoft Azure**
    - <https://docs.microsoft.com/en-us/azure/cognitive-services/ink-recognizer/>
    - Hand-written recognition
    - We also plan to support processing the written notes uploaded by the users, like a photo of a paper-pen notes
    - And this API from Microsoft seems to be a perfect match
  + **Translator Text from Microsoft Azure**
    - <https://docs.microsoft.com/en-us/azure/cognitive-services/translator/>
    - If the user needs translations of the terms, we can use this API to get the results
  + **Language Understanding (LUIS) from Microsoft Azure**
    - <https://docs.microsoft.com/en-us/azure/cognitive-services/luis/>
    - The flashcards will be tagged with the “topics” of the documents it generated from
    - We will use this API to get those tags to tag our flashcards
  + **Content Moderator from Microsoft Azure**
    - <https://docs.microsoft.com/en-us/azure/cognitive-services/content-moderator/>
    - Users should not be able to upload offensive and inappropriate content to our database and share with other people
  + **Video Indexer from Microsoft Azure**
    - <https://docs.microsoft.com/en-us/azure/media-services/video-indexer/video-indexer-overview>
    - Users will be able to get information from the videos they upload to the web app
    - We can use this API from Microsoft to process the videos uploaded
  + **Bing Custom Search from Microsoft Azure**
    - <https://docs.microsoft.com/en-us/azure/cognitive-services/bing-custom-search/>
    - Users will be able to search for the flashcards or notes published in our domain using this bing API easily
  + **Bing Spell Check from Microsoft Azure**
    - <https://docs.microsoft.com/en-us/azure/cognitive-services/bing-spell-check/>
    - Will use this API to spell check all the possible forms where the users will input, like search bars and content on their profile pages and so on
  + **CalendarIndex**
    - [https://calendarific.com/](https://calendarific.com/?PageSpeed=noscript)
    - We will use the data from here to mark the important event on the user's calendar/timeline, so that users can organize and make better scheduling decisions.
  + [letsencrypt](https://letsencrypt.org/)
    - We will use letsencrypt to add SSL support to our website
  + <https://auth0.com/>
    - We will employ auth0 as our login system
  + **Twitter API**
    - Users will be able to post their studying progress on their favorite social media
  + **Linkedin API**
    - Users will be able to post their studying progress on their favorite social media
  + **Facebook API**
    - Users will be able to post their studying progress on their favorite social media