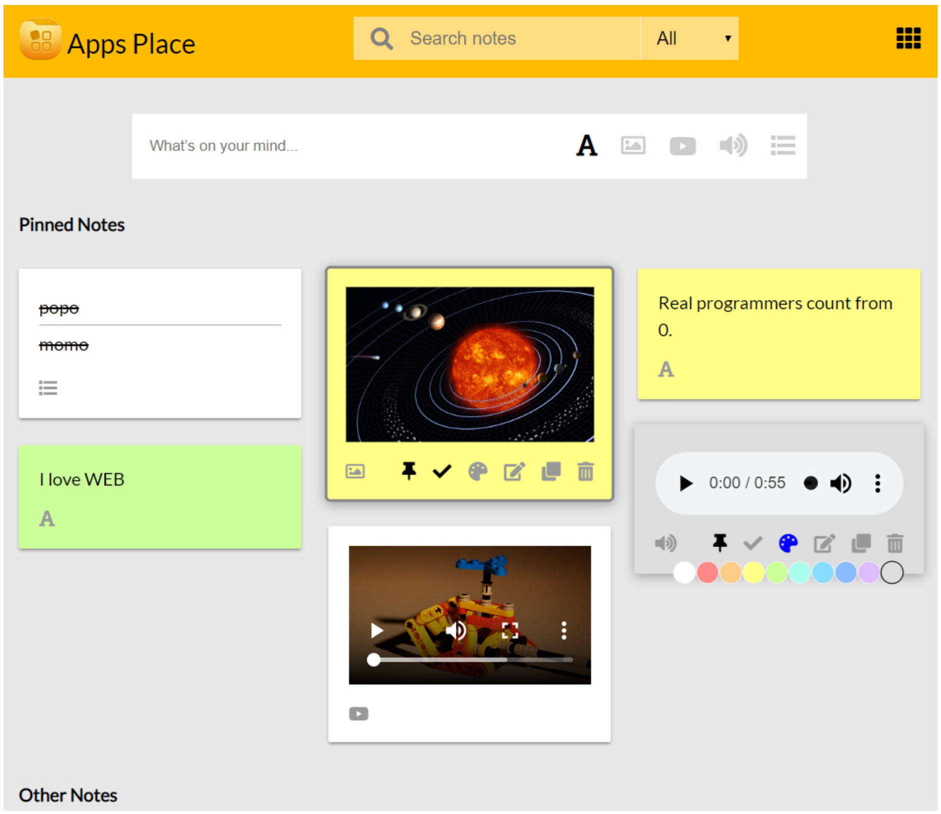
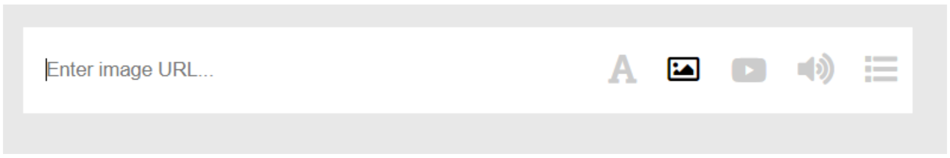
MissKeep *App*

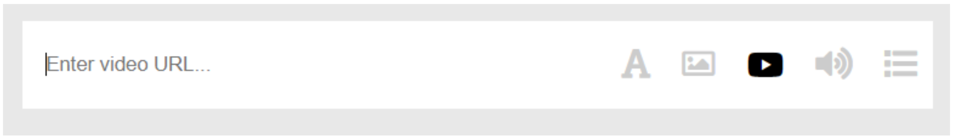
Build an app for keeping things (reference app "google keep")

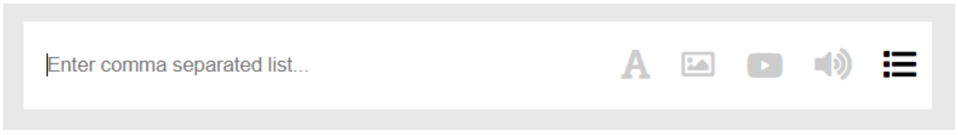
UX (hover states)



Notes Variations:







Build the root component: <MissKeep>

Render a list of components, use dynamic components to show different types of notes:

- <NoteTxt>
- <NoteImg>
- <NoteTodos>
- <NoteVideo>
- <NoteAudio>: bonus
- <NoteMap>: bonus
- Here is an array of some notes:

```
var notes = [
  {
    type: "NoteText",
    isPinned: true,
    info: {
      txt: "Fullstack Me Baby!"
    }
  },
  {
    type: "NoteImg",
    info: {
      url: "http://some-img/me",
      title: "Me playing Mi"
    },
    style: {
      backgroundColor: "#00d"
    }
  },
  {
    type: "NoteTodos",
    info: {
      label: "How was it:",
      todos: [
        { txt: "Do that", doneAt: null },
        { txt: "Do this", doneAt: 187111111 }
      ]
    }
  }
];
```

Support the following features:

- Allow creating, updating and deleting notes (CRUD)
- Support setting the note's background color and other styles
- Support searching notes
- Pin a note to the top of the list
- **Apps Integration**
    - Allow sending a note content straight into the compose-message page in misterEmail (use queryString Params)
    - Use a <LongText> component - gets the text to format as a **prop**. (this component is used by both apps)
    - Use a <UserMsg> component for showing success / error messages (this component is used by both apps)

Learn about dynamic components

- For VueJS, here is an article about Dynamic components, and here is a video - Hebrew, English.
- For React, here is the basic idea of implementing Dynamic components:

```
const DynamicCmp = (props) => {
          switch (currView) {
              case 'Hello':
                  return <Hello {...props} />
              case 'GoodBye':
                  return <GoodBye {...props} />
              case 'WelcomeBack':
                  return <WelcomeBack {...props} />
              default:
                  return //...some default error view
          }
      }
```