**What I did**

* Created the react-redux environment
* Created table framework for status pages (pages.js)
  + Table framework containing the column name and respective column span
  + This way new status pages can be added to pages.js irrespective of number of columns
* I called Datadog Api to get list of services and filtered the array for the required service of interest
  + Event pipeline and Alerting engine
* I called Azure api to get xml data and with xml-js library, I converted it to JSON object
  + I made a default Azure object containing Virtual machines and Cloud service keys with "operational" values
  + My objective is that if xml data reports faults relating to cloud services and virtual machines, I update the default Azure object.
  + This objective was not completed because I was not able to get any faults relating to these services properly parse the xml to the needed json object
* Added component for creating new services for status pages
  + Component contains new service button to toggle the form or not
  + Form contains the service name and other fields required by the status page
  + On save, updated the redux date with the form data

**What I did not do**

* I did not create the component for adding new status page.
* Like adding new services to the status page, adding new status pages will add new object to pages.js data
  + This will require another redux state that contains new status pages called newStatusPages.
  + With Object.assign, I will combine the object data from pages.js with the newStatusPages object from the redux state
  + Another way will be to move the pages.js data to the redux state and directly update it with new status pages data
* Create a form for collecting new status page data
  + Required form data is status page name (pageName) along with number to specify column span,
  + Optional data are other column names with their column span

**Libraries used (available in package.json)**

* React,
* React redux,
* Redux,
* Redux thunk
* Axios
* Eslint
* Other libraries are
  + React-time-ago for getting time from last refresh
  + Redux-devtools for redux debugging
  + shortId for unique id characters
  + xml-js for converting xml data to json object