# INTRODUCTION:

Following are the scenarios for which the script is write;

* 1. [Hobsons.com](http://hobsons.com/) home page renders as expected.
  2. On the home screen there is a Hero graphic with the text “**We help students** across the journey of a lifetime.” Click the down arrow. Assert the page scrolls the next sections “How can we help **your students?**” into the viewport window. Assert that it is correctly aligned with the top of the visible screen
  3. When user clicks the “hamburger” menu at the top of the screen then the items the list drops down as expected with the sub-menu items. The menu I’m referring to is Solutions, Services, Resources, About & Blog. Assert that the “*Resources*” menus contain a list of child links including “E*vents*”.
  4. Navigate to Resources > Events. Assert that all events on the page occurs in the future. Note you do NOT need to apply the filters, just assert on the default list that is displayed.

# APPROACH:

There is a **describe** function (as we are using Mocha framework) which work as a main and every test flow (mentioned in previous section) has its own **it** functions.

# TOOLS:

The above scenarios are automated on **Cypress**.

# FILES

1. Gtag exception has been written in index.js file under Support. Gtag basically sends data to Google Analytics showing how many visitors have visited the website and as we are using an automated browser so Gtag is preventing it.
2. The script is written in **automation.js** file under **integration**.

# Git REPOSITORY:

https://github.com/umair-hussain-10p/Hobsons---coding-challenge.-

# LIMITATIONS:

There is a limitation on the last point ‘Assert that all events on the page occurs in the future’. The problem with it is all the dates are in different formats like [Month. Days range, Year] OR [Month. Days Range] OR Year. We can get the dates by using following code;

var TextInsideLi = Cypress.$('p').text();

console.log(TextInsideLi) ;

The above code will give us the text in the tag and then we can parse the date from it. But After parsing it is impossible to compare it with all the other dates.

Another approach is we write regular expression for the dates but it would be impossible to get which date format is using in the particular scenario.

Therefore, we have only assert the future function in the current list.