Title

Entities component in Newzera website

Description of the problem

Develop the Entity components (Entity Card, Similar entity card, thumbnails, Entity in activity feed and list of entities) for the newzera website and test them if they are being rendered correctly to the display screen.

Functional requirements

We have to design the frontend UI for the Entity Components for Newzera Website. It Includes developing 5 different types of Entity Components. The Entity Component types are as follows:

1. Entity Card
2. Similar Entity Card
3. Thumbnails
4. Entity in Activity feed
5. Entity List

The designs should be responsive according to different viewport sizes and pass the required tests.

Non-functional requirements

The User interface is intended to be smooth and user friendly.

Non-goals

1. Will not handle the change of follow - > following and vice versa for buttons.
2. Will not pass the array of props (EntityNames, EntityImages etc) through the components. Would rather declare them in default props.

Assumptions

1. Assumed all the default boolean props to be false.
2. Created Carousel for similar entity card.
3. Assumed the number of cards to be scrolled each time in the carousel when the user clicks the left/right button.
4. Assumed some images’ border radius as it was not mentioned in the design doc (Confirmed this with Shruti).

Impact

The Entity components developed in this project would be later used in the entity flow main page. While creating an Entity Flow main page, the person will just need to call the particular entity component he/she wants to use along with the props he/she wants to change.

Solutions - all including considered and discarded

All the solutions to the problem go here.

1. **Have leftStartPoint and topStartPoint as props for all the components.**

This will help a lot whenever a person will try to take/call any one of the entity components. He/She will just need to mention in the props at what height from top and what length from left, he/she wants to place the entity component.

1. **Create an array for the props.**

With this implementation you just need to pass the array of props. The code will take care of positioning the inputs accordingly. For example, on passing the array of Entity images, Entity names and following/not following status for that Entity, each card with mentioned input will be displayed in the carousel appropriately.

1. **Handled how top line in “Entity in activity feed” section should look**

Using mathematical expressions, got the string to be shown at the top for any number of “Following Entity Names” passed.For example, if a user follows {n} entities, the top line in activity feed card will be “[UserName] started following [EntityName1], [EntityName2], [EntityName3], and [n-3] more”. The string will be generated for any value of n.

1. **Created a carousel for the similar entity cards.**

In the entity flow page, there is a section for similar entities with carousel for it. Therefore there is a carousel for the similar entity cards in this project. It will make it easier for a person developing the entity flow page as the carousel is created here.

There are several ways to create a carousel - >

* Flickity carousel -  **Discarded -** Used flickity dependency (Not free for private organizations) hence discarded
* Pure React Carousel - **Discarded -** Lots of debugging needed in the function handling scrolling. Hence discarded.
* Owl Carousel - **Discarded -** Unable to figure out working of certain functions and had a complex css file. Hence discarded.
* **Worked -** Using functions created from plain vanilla javascript. This method had 3 simple functions and did not involve any complex styles. Hence it worked.

1. **Rendering buttons from the button components.**

Instead of using the button assets in “Entity Card”, “Similar Entity Card”, and “List of entities”, I rendered the buttons from the button-component.

External dependencies

No external dependencies were used.

Internal dependencies

1. Used Buttons from the buttons-component
2. Assets downloaded from the design xd link
3. Functions in obi/web/newzera-web/src/transform folder to convert px -> vw/vh.

Planning

1. Start with basic setting up of project-

* Clone the main repo(obi) in local PC
* Create a new branch where entity components will be developed
* Push the new branch to the github

1. Go through the design XD link and download all assets, create appropriate files and

directories for the project components and get an idea of where to begin with.

1. Start developing the 5 entity components step by step.

* Entity Card - <https://newzera.monday.com/boards/611894737/pulses/649352394>
* Similar Entity Card - <https://newzera.monday.com/boards/611894737/pulses/649352653>
* Thumbnails - <https://newzera.monday.com/boards/611894737/pulses/649352938>
* Entity In Activity Feed - <https://newzera.monday.com/boards/611894737/pulses/649353771>
* Entity List - <https://newzera.monday.com/boards/611894737/pulses/653480860>

1. Keep below mentioned points in mind while developing entity components.

* Make the page responsive.
* Discuss with shruti if there are any design inconsistencies.
* Make buttons toggle wherever necessary.
* Import buttons from button-component.

1. Create and run tests for the entity components -

* Entity Card - <https://newzera.monday.com/boards/611894737/pulses/649352501>
* Similar Entity Card - <https://newzera.monday.com/boards/611894737/pulses/649353237>
* Thumbnails - <https://newzera.monday.com/boards/611894737/pulses/649353080>
* Entity in Activity feed - <https://newzera.monday.com/boards/611894737/pulses/649353869>
* Entity List - <https://newzera.monday.com/boards/611894737/pulses/653481070>

1. Two types of tests would be implemented.

* Snapshot test for each component
* Unit tests to check if props are being rendered correctly

1. Resolve “Eslint” and “flow” errors if any.
2. Push the changes to the created github branch if there are no errors.
3. Make the PR if there are no conflicting files with the master branch. If there are any

conflicting files then resolve them on VS code and then proceed with making PR..

1. Once all the checks pass in PR and it is rebased with the current master branch, the

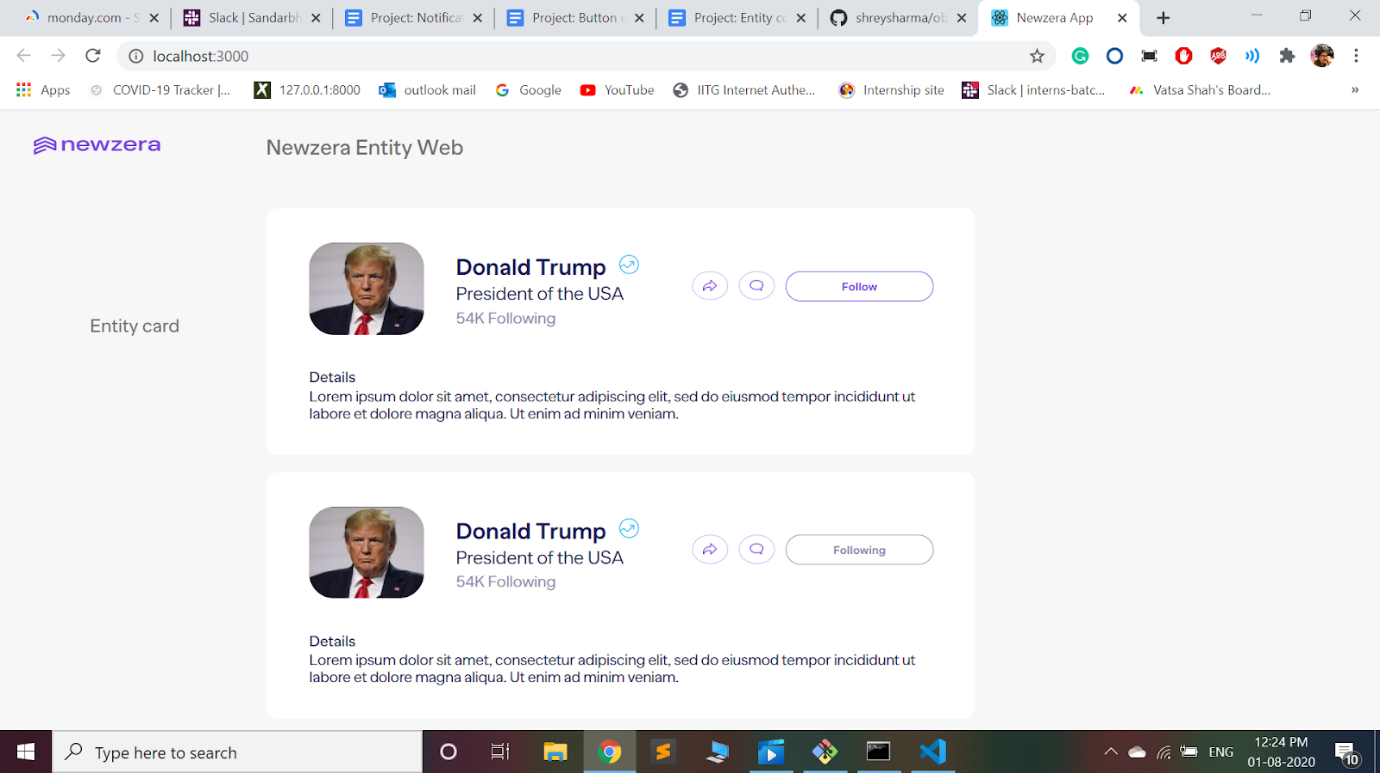
Reviewing process can start.

Deliverables

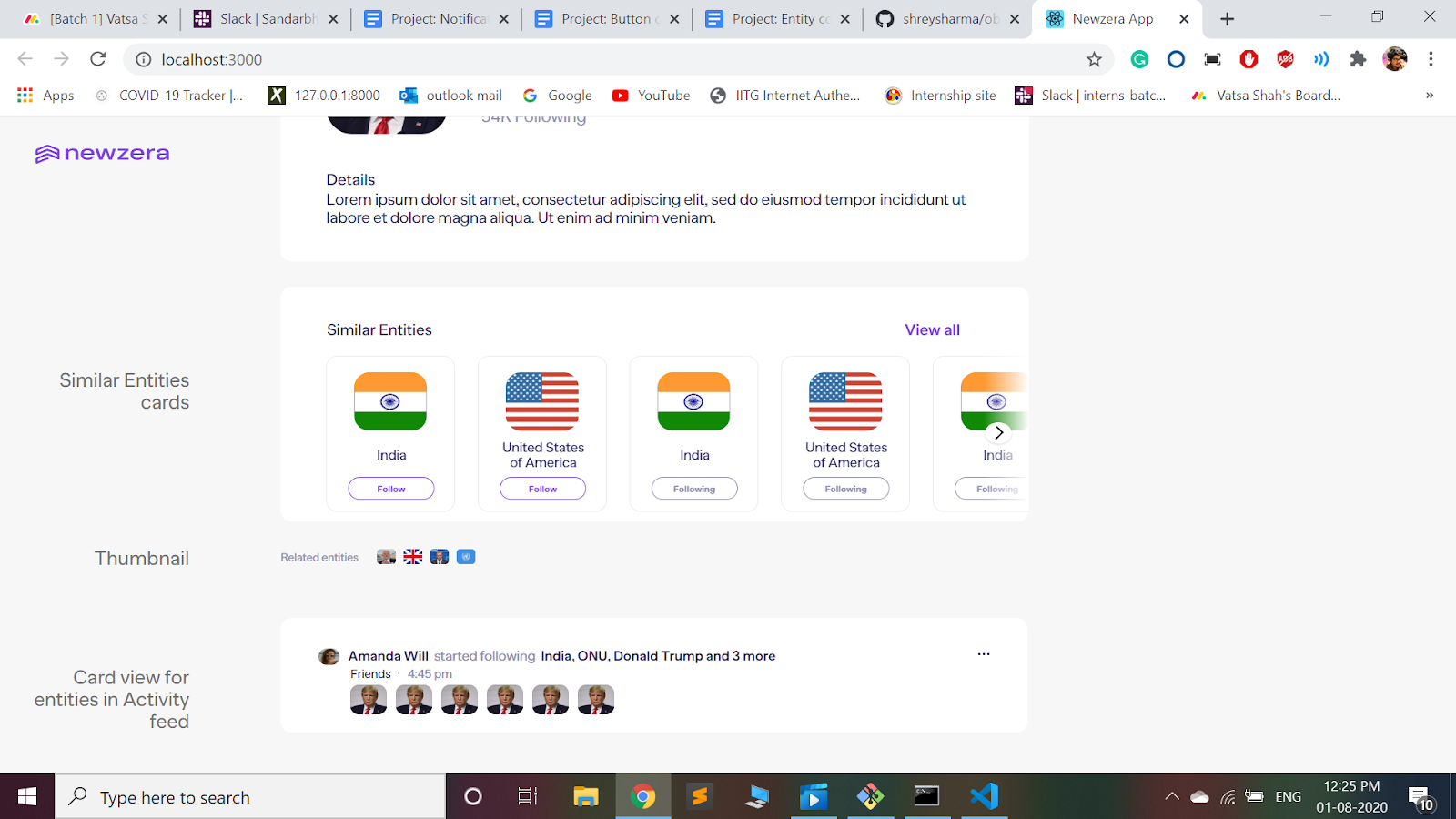
Entity components is the end result of this project. These entity components will be imported in the “Entities” section of the main flow page.

User Interface

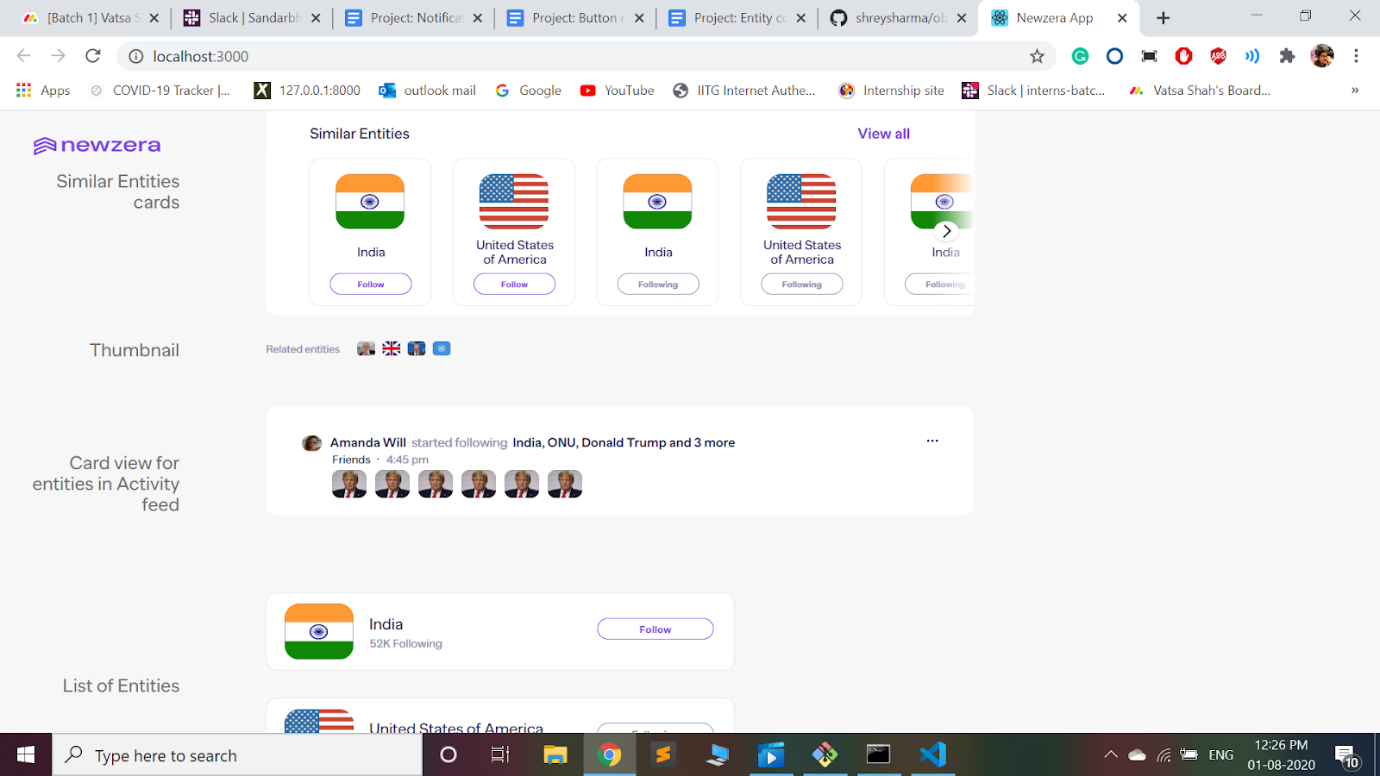
1. Entity Card -



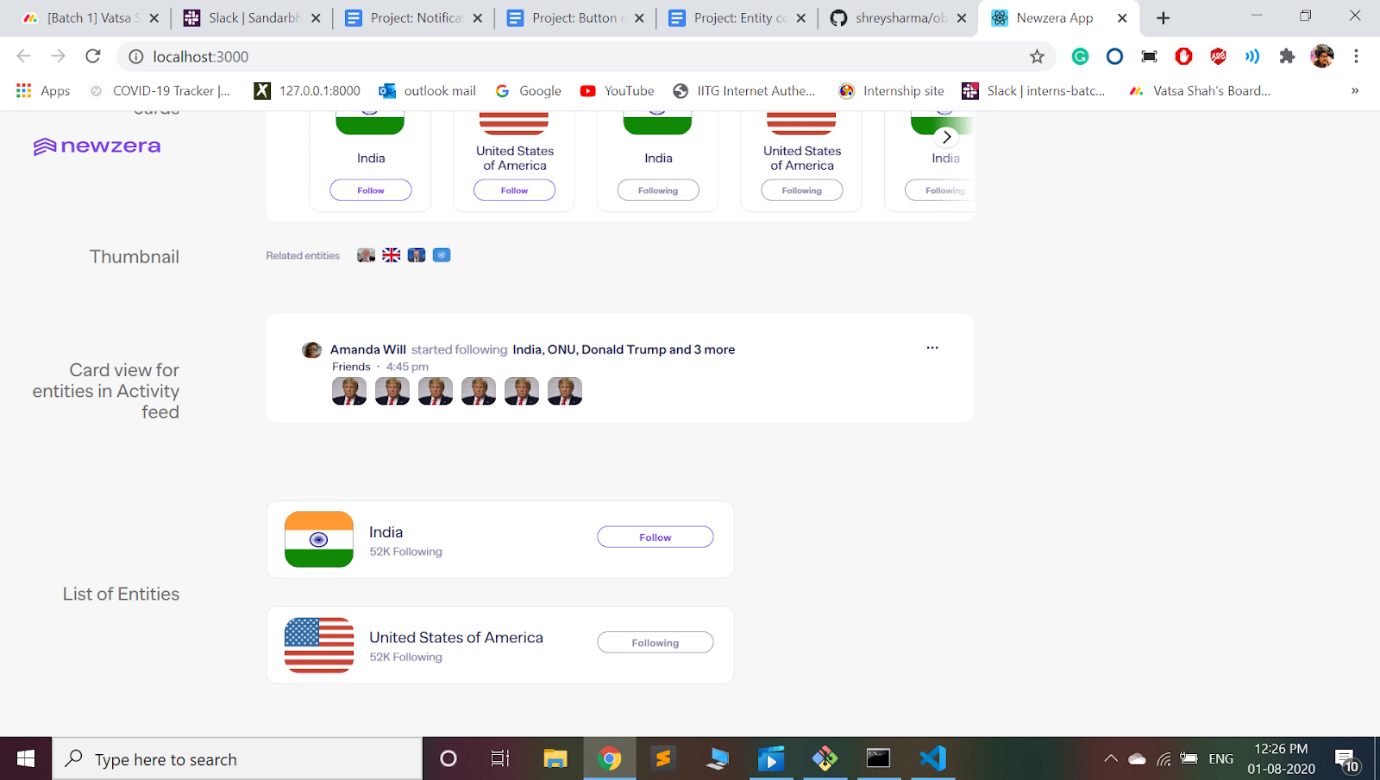
1. Similar entity card -



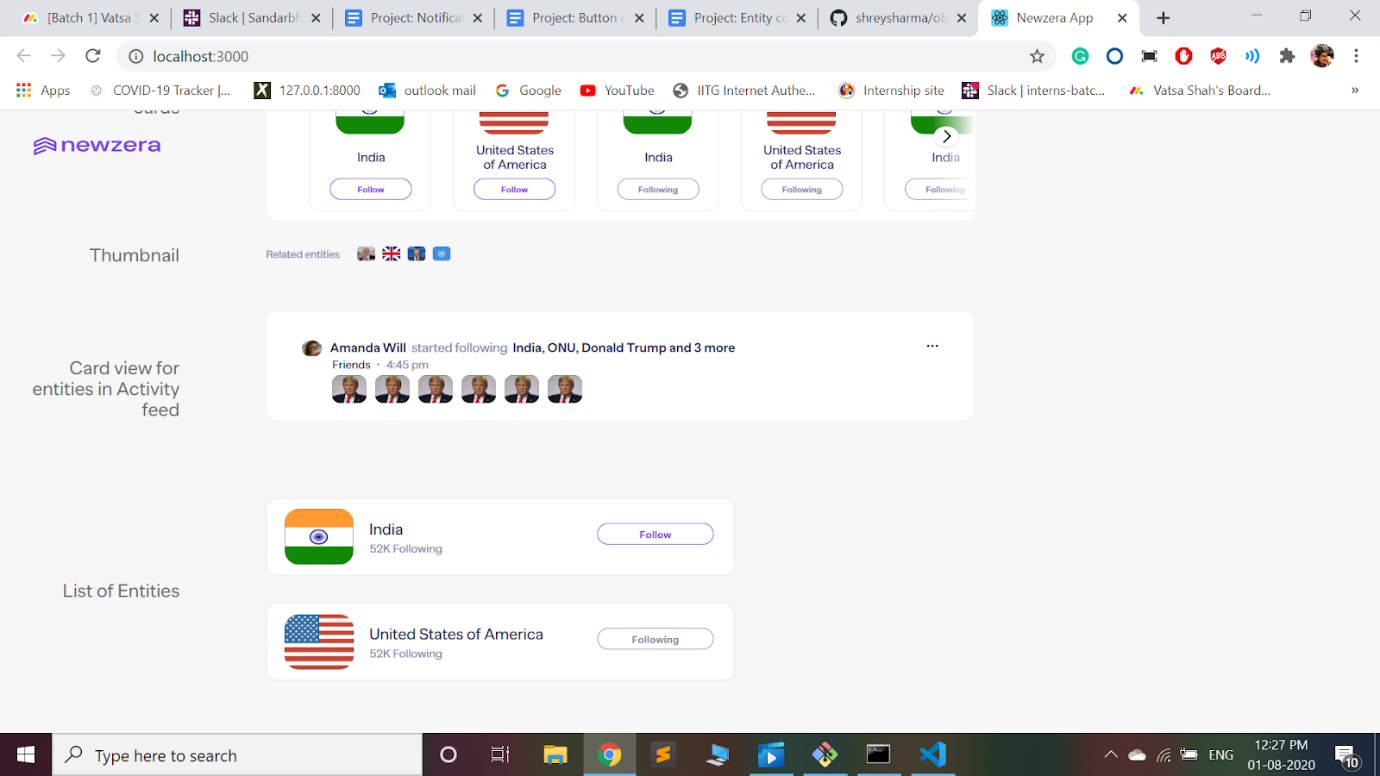
1. Thumbnails -



1. Entity in Activity Feed -



1. List of Entities -



Test plan

1. **Unit testing** using mount -

* To check if buttons are rendering correctly
* Array size of Entity name and Entity images should match

1. **Snapshot testing** using shallow to ensure the proper rendering of the components.
2. **Code coverage report** -

**@@            Coverage Diff             @@**

##           master     #207      +/-   ##

==========================================

+ Coverage   35.26%   45.10%   +9.83%

==========================================

  Files          43       88      +45

  Lines         896     1747     +851

  Branches      155      296     +141

==========================================

+ Hits          316      788     +472

- Misses        544      884     +340

- Partials       36       75      +39

Milestones

1. Entity Card, Similar entity card, thumbnails, Entity in activity feed and list of entities should be developed as per the design.
2. The props should be rendered correctly. i.e snapshot and unit tests should pass.

Follow up work

Would be working on the entity flow main page and check if the entity components created in this project will render correctly or not.

References

1. For creating a carousel from scratch, Follow the source - [Simple Plain Vanilla Javascript Carousel](https://codepen.io/gkaza/pen/eYJQxLy?__cf_chl_jschl_tk__=7d0d18e4d7f7b0ae8e98307e25b319d720d3ca64-1596190150-0-AXNL8BW4Li4-s2P5n0LYDgA8WwehT13JHYGw8D85c2xiGY5rHfdj9cwh1IMeqT41eXuUojNVN3tLvo5paEuKNnlbXsxsDrYUDlEB9UkEze_LQdM6cCFlPC7jTP_RIvVGJiiTnR6uIX5NWST7ax9wcs9q9vZyeW-5G8Csx0SqcEEb54DMP3nfF3kdyGV4j2TnwtbpXn702GT5QowgpxZagVi_KoPLrb6oAZLPgqh7GbZAwh3Rx0uz_iDqC_dqbHukCLtM2eVv6FSTG1hTjgcEEKqHqWbJY9OpAYYilU5a974hinDK_NGjhu9ANI0k5YQxxV4AXrvxqsqMP3SDcAfrT9rC2QQe0oZgMuRy4JY-FQZz)

Glossary

Carousel - slideshow for cycling through a series of content