Title

Buttons component in Newzera website

Description of the problem

Develop the Button components(Circular Buttons, Activity Buttons and Rectangular Buttons) for the newzera website and test them if they are being rendered correctly to the display screen.

Functional requirements

We have to design the front end UI for the Button Components for Newzera Website. It Includes developing 3 different types of Button Components. The Button Component types are as follows:

1. Circular Buttons
2. Activity Buttons (Share, Raven and React)
3. Rectangular Buttons

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

Non-functional requirements

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

Non-goals

Will not handle the change of follow - > following and vice versa for Rectangular buttons.

Assumptions

1. Assumed all the default boolean props to be false.
2. Assumed the border thickness for all buttons to be 1px
3. Assumed default activity button to be “Raven”
4. Assumed same letter-spacing for all rectangular buttons.

Impact

The Button components developed here would be later used in all the flow pages. For using a particular button in any flow page, the person will just need to call the particular button 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. **One component for the circular button.**

A person importing the circular button can change it’s type using props. For example, if a person wants to have a bordered circular button, he/she just needs to provide “bordered” value as true in props.

1. **One component for the Activity button.**

A person importing the Activity button can change it’s type using props. For example, if a person wants to have a “share” button, he/she just needs to provide “activityType” prop value as “Share”. Prop values for using raven and react button are “Raven” and “React” respectively.

1. **One component for the Rectangular button.**

A person importing the Rectangular button can change it’s type using props. The different props associated with the rectangular button are -

RectangleButton.defaultProps = {

  title: 'Button',

  disabled: false,

  bordered: false,

  filled: false,

  height: 50,

  width: 300,

  radius: 0,

  fontSize: 18

}

A person can change the prop values as per his/her requirements.

External dependencies

No external dependencies were used.

Internal dependencies

1. Assets downloaded from the design xd link
2. 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 button 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 3 button components step by step.

* Circular Buttons - <https://newzera.monday.com/boards/611894737/pulses/652433601>
* Activity Buttons - <https://newzera.monday.com/boards/611894737/pulses/652433605>
* Rectangular Buttons - <https://newzera.monday.com/boards/611894737/pulses/652433610>

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

* Make sure to make the page responsive.
* Discuss with shruti if there are any design inconsistencies.
* Make buttons toggle.

1. Create and run tests for the button components -

* Circular Buttons - <https://newzera.monday.com/boards/611894737/pulses/652433603>
* Activity Buttons - <https://newzera.monday.com/boards/611894737/pulses/652433607>
* Rectangular Buttons - <https://newzera.monday.com/boards/611894737/pulses/652433614>

1. Test each button component after completing the development.

* Run snapshot tests for each component
* Run the unit tests to check if props are 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

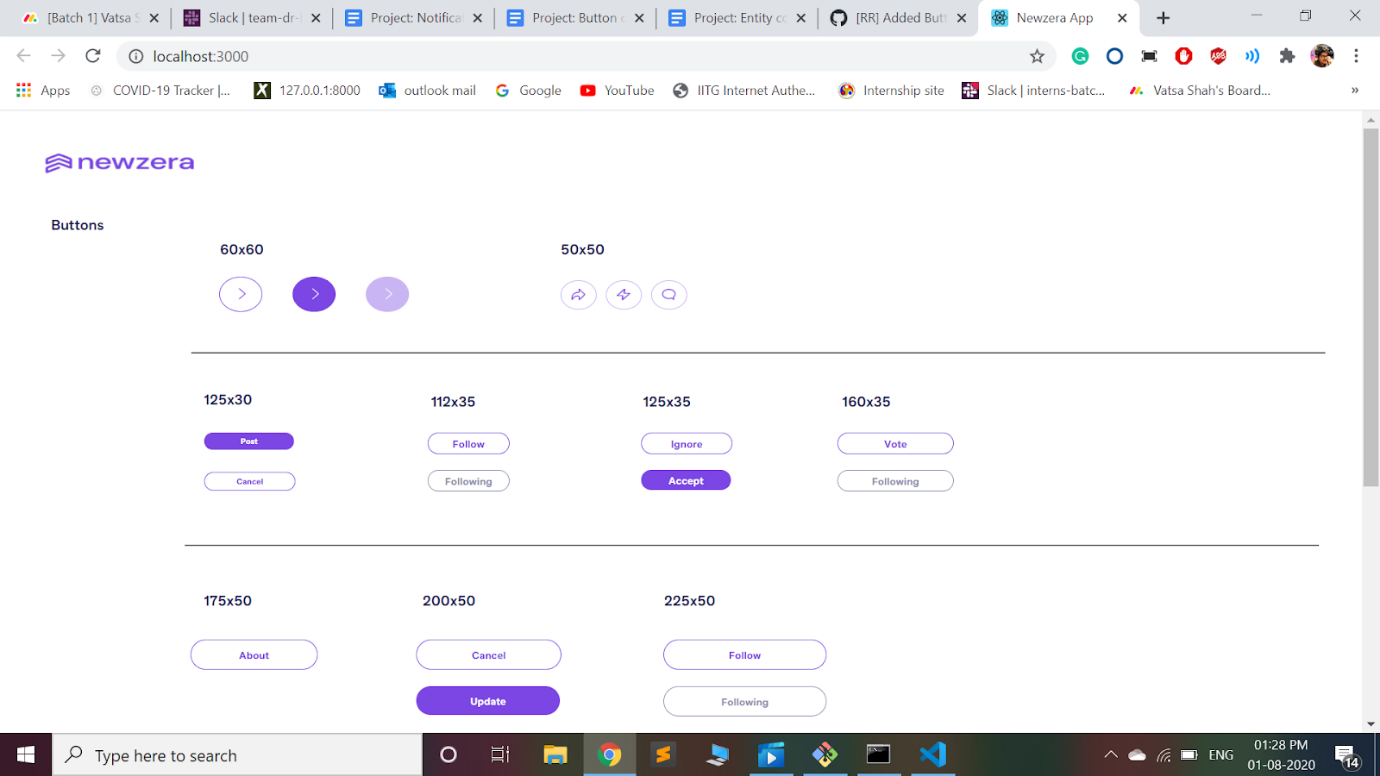
Reviewing process can start.

Deliverables

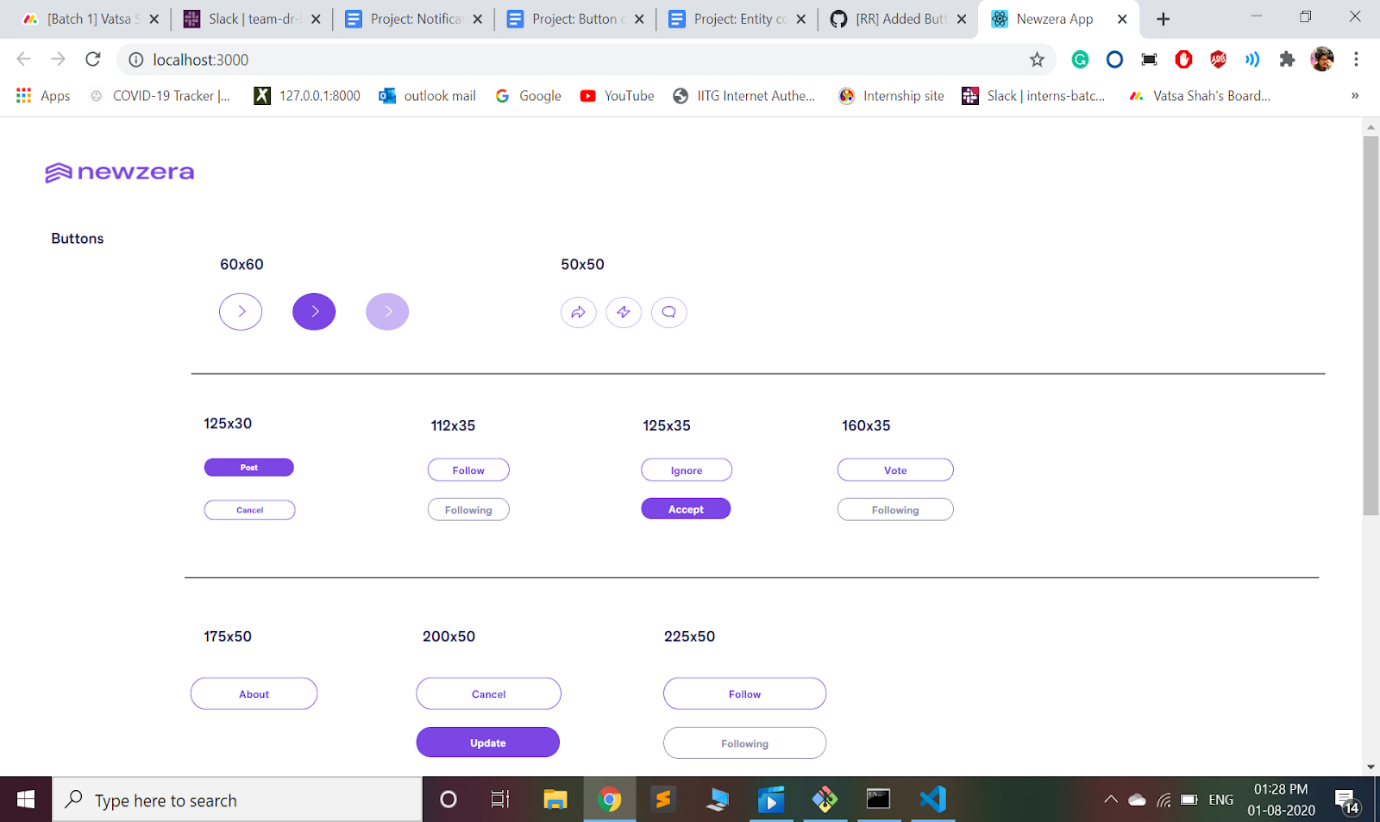
Button components is the end result of this project. These button components will be imported in the various sections of the main flow page.

User Interface

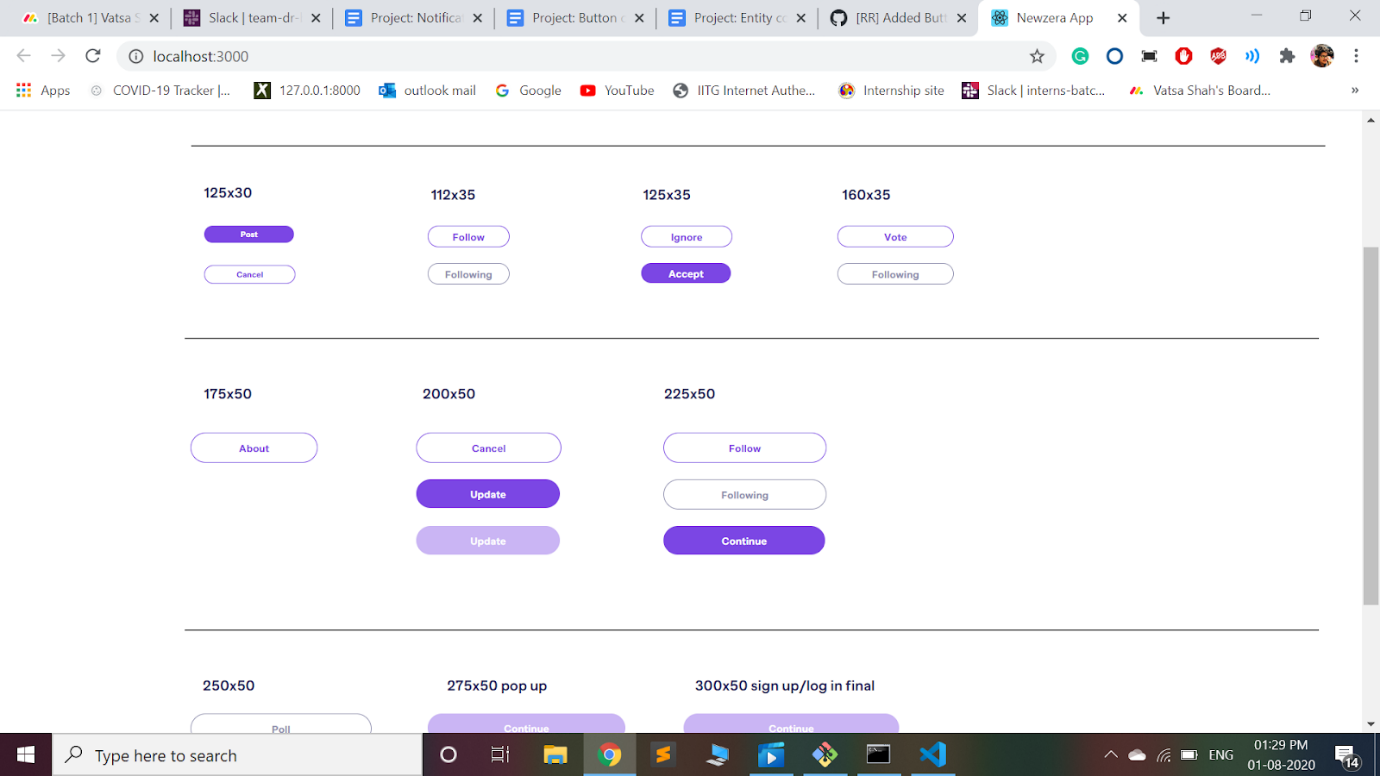
1. Circular Buttons -

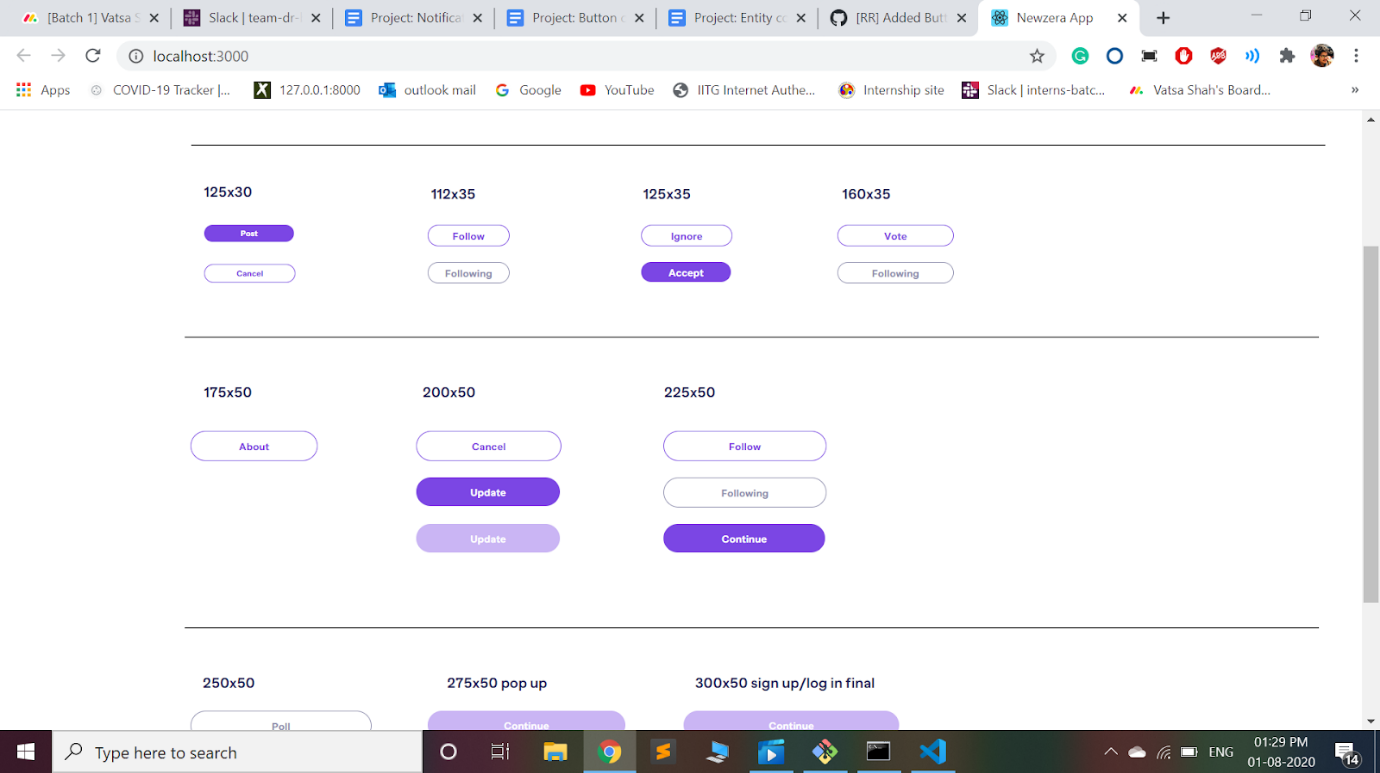


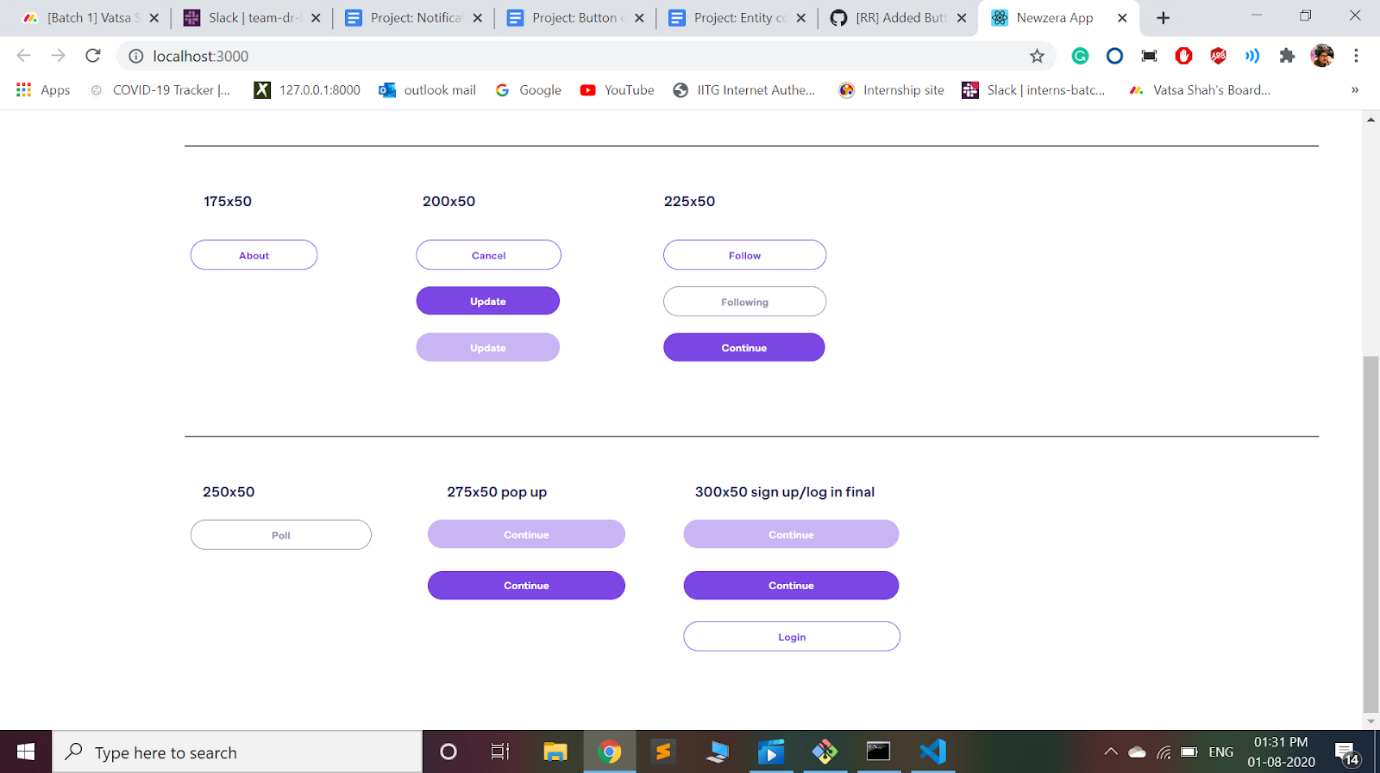
1. Activity Buttons (Share, Raven and React) -



1. Rectangular Buttons -







Test plan

1. **Unit testing** using mount to check if buttons are rendering correctly
2. **Snapshot testing** using shallow to ensure the proper rendering of the components.
3. **Code coverage report** -

**@@             Coverage Diff             @@**

##           master     #224       +/-   ##

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

+ Coverage   35.26%   48.37%   +13.10%

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

  Files          43       48        +5

  Lines         896      955       +59

  Branches      155      162        +7

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

+ Hits          316      462      +146

+ Misses        544      465       -79

+ Partials       36       28        -8

Milestones

1. Circular Buttons, Activity Buttons and Rectangular Buttons 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-component page and check if the button components created in this project will render correctly or not.