1. Dark Mode
    a. Feature 1 is a dark mode in the nav bar at the top right there is a button to switch to dark mode press the button until it says dark mode. Now the page is set to dark mode, it uses darker colors to be easier on the eyes.

2. Custom Templates for Pages
    a. The first part of feature 2 is setting custom colors for text and background for pages. To do this either edit the existing page and go to the bottom and put in a hex color value for the text and or the background or create a new page and put the hex value In at the bottom.

    b. The second part of this feature is the ability to see custom page colors and background colors. Do this by pressing the button in the nav bar until it says custom then go to a page with a custom color for text and or background. Note if a page does not have custom colors, it will use the default light theme.

3. Accessibility Features
    a. This feature adds a new button labeled "Accessibility" to the navbar of all of the pages. When clicked, this button creates a pop up window with two checkboxes: High Contrast and Large Text. When checked, the High Contrast mode turns backgrounds to black and text yellow. This is a common type of accessibility feature designed for users that may struggle to differentiate similar hues. The Large Text checkbox, when checked, increases text size by 50%, which is more drastic than it may sound. This is intended for users that may not be able to see small text as well as others.

    b. Feature 3 modifies base.html to add the button to the Navbar which is present on all pages. The logic for the high contrast mode and large text is also embedded into this html for simplicity since they're rather small functions.

4. Citation Generator
    a. Feature 4 is a button at the bottom of pages in the Riki site. Pages (articles) can be created and mimic articles in wikipedia in their form and can be indexed by tags. Given this site's similarity to wikipedia, we can infer that users may want to cite articles they find on the website for scholarly purposes. To assist them in this, we created a method that generates two of the more popular formats of citations using variables already present in the code.

    b. When the user navigates to a page on the site, they can scroll to the bottom of the page to find a button labeled "View Citations". When they click on it, a pop up window appears with the MLA and APA citation printed out as well as a button under each labeled either "Copy MLA" or "Copy APA" that will copy the citation to their clipboard.

    c. The features was added by modifying page.html which is the html used for pages/articles. It contains metadata such as the title and URL which are required. Then the publisher, Riki, is added, and we use formatDate() to get the time the article was accessed.

5. Tutorial Page
    a. To use and understand the instructions page on the website you simply need to be on the home page and select the instruction tab to view the HTML page with the information on it.

    b. The code view is used to go to the templates folder and locate the instruction.html to view the info. To adjust either the page linking or the address go to the routes folder to adjust the /instruction/ section to the adjusted address and go to the base.html to adjust the page link.

6. User Privilege
    a. To access and use both the roles page and the admin page you will first need to login in with your user account. Then click the roles and fill in your user name with the title of the role you want assigned. To access the admin page you will need to assign the role "admin". Once you access the admin page all your roles will be cleared and you will have to re-add the roles in the role page.

    b. In the code form, you can go to the roles.html under templates and adjust anything needed in the roles.html page. The methods below Assign Role is used to assign the user the roles inserted and is needed to allow the other code to function. The page access and role assignment are done within routes.py and the user.py page where routes.py has a creation and assignment for the role while the user.py allows the role to be passed into the list and saved. When using the admin.html page you have an if statement to check for the correct role to access the page and once it is correct it will clear the roles and let the user view the page.