Akila Induranga
akila.99g@gmail.com
Colombo, Sri Lanka
Timezone: IST - India Standard Time (UTC+5:30)

Adding a custom text color picker for Kolibri EPUB renderer

GSoC Project proposal for Learning Equality

Personal Details and Contact Information

- Github username : akila-i
- Email : akila.99g@gmail.com
- University : University of Moratuwa, Sri Lanka
- Time-zone : IST - India Standard Time (UTC+5:30)
- Social Profiles : LinkedIn
  : StackOverFlow

Synopsis

- Kolibri offers a well integrated EPUB renderer for its users with smooth responsiveness, control over the text size, and six color themes to choose from. The existing color themes include variations for both light and dark modes. However, those color themes are fixed to the application and cannot be fine tuned as per individual user needs. The project for this summer would be to design and implement a custom color picker component for the Kolibri EPUB renderer so that users can have their own custom color themes in the application.
- This project will stand mainly in the accessibility area of Kolibri's development.
- Primary goal of this project is to deliver an extension for the color picker of Kolibri EPUB renderer. Apart from that, improving the accessibility and user experience of the Kolibri platform, and supporting the open source software development community to grow can be identified as secondary goals of this project.
- The following repositories have been identified to work with during the project.
  - Kolibri
  - Kolibri-Design-System
  - Kolibri-Docs
- The project would be guided to success with the advice from the following mentors.

- Radina Matic @radina@learningequality.org
- Marcella Maki @marcella@learningequality.org

Benefits to the Community

- Accessibility of the contents has become an essential factor in software development nowadays. According to WHO, almost 15% of the world's population has some sort of disability and we as software developers should take all possible actions to include them in our user base where they can comfortably use the software products. Visual impairments are one kind of such disabilities. This project allows those groups of people to customize the theming (including the text color, background color, and hyperlink color) according to their individual contrast needs.
- By allowing them to save the customized color combinations as themes in the application, they would always feel like home when using Kolibri and will make them tend to come back, which will help progress Kolibri.
- For the future developments, one suggestion would be to have an option where visually impaired users can download a separate pack of themes just for themselves. We can introduce such theme packs through a series of surveys and research. That could be implemented on top of the custom color picker option.

Current Status of the Project

- The project is an extension for the already available themes selector of the EPUB renderer in Kolibri. Since it only has a fixed set of themes, this project will be built on top of the available EPUB renderer settings to have an additional set of custom created themes as per users' preferences.
- Few functions from the existing themes (applying a theme to the EPUB renderer, saving theme details to the local storage) to be used with slight modifications for this project.

Goals

Goal 1 :

Implement a feature for users to apply custom color themes in addition to the ones already provided by Kolibri

Goal 2 :

Integrate a color picker for users to customize the color themes they are adding to the EPUB renderer.

Goal 3 :

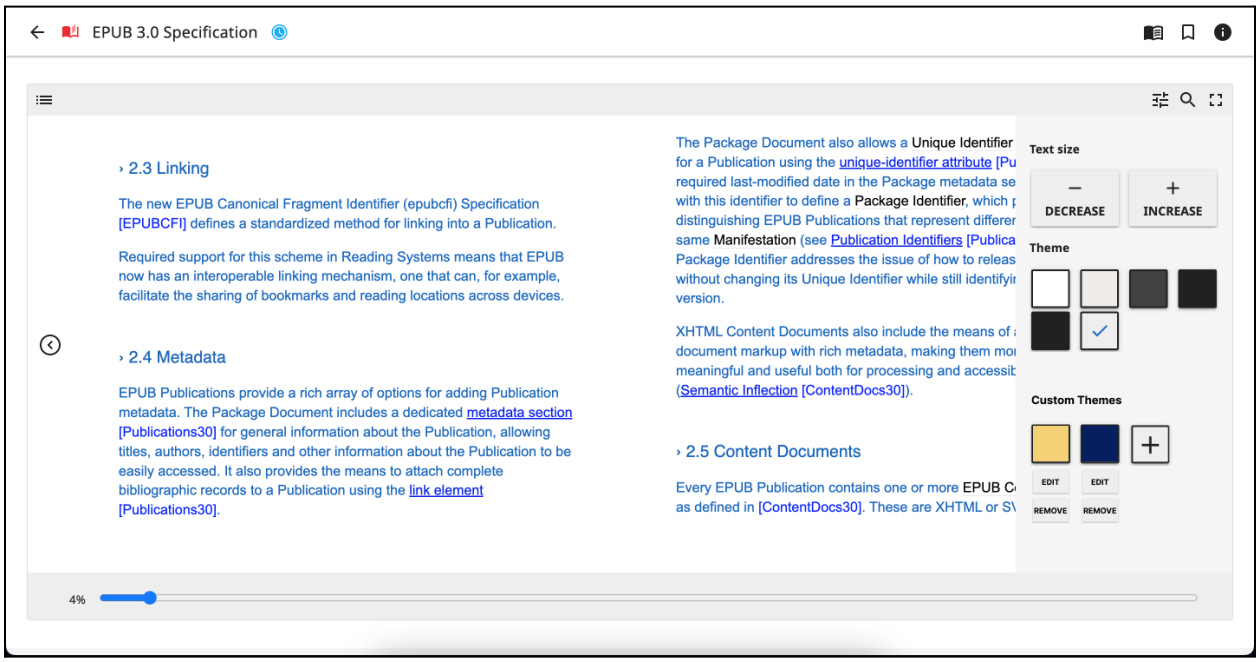
Implement a feature for users to save the custom color themes for future use, and to remove the custom themes when no longer needed.

Deliverables

Coding period is divided into mainly two phases, as before and after the mid evaluation. Following deliverables are planned to be produced during the mentioned phases.

Deliverable 1 - Display section for custom themes

- A separate section will be introduced just below the in-built themes to accommodate the custom themes. Right now it is planned to let users have up to 4 custom themes as shown in the wireframe below.
- Expected by Evaluation 1

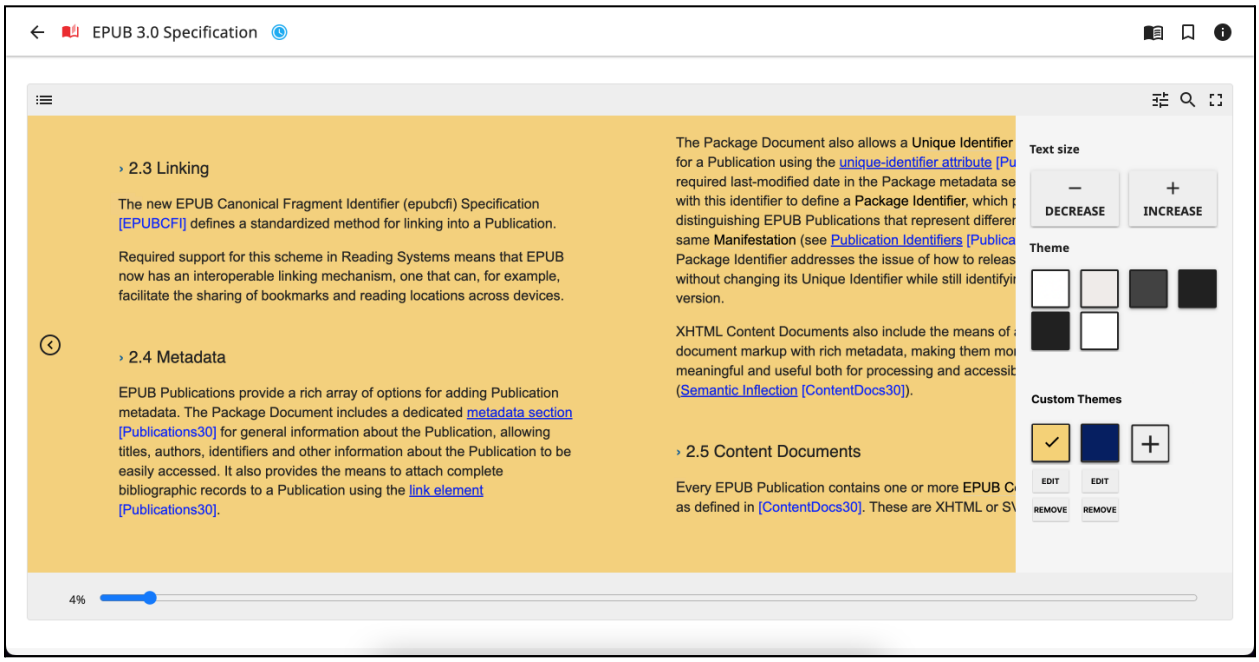


Deliverable 2 - Feature of changing EPUB renderer according to selected custom theme

- After having the space for sample custom themes on UI, next is to make the function to apply the colors in the selected custom theme when one is clicked.

This integration is evaluated to be easy as it can use the already existing functions (*setTheme()* function implemented in *EpubRendererIndex.vue* file.) from the in-built theme selector. At this stage , it is decided to use hard coded values of colors as themes and use them inside the *setTheme()* function, just to get the colors applied to the EPUB renderer.

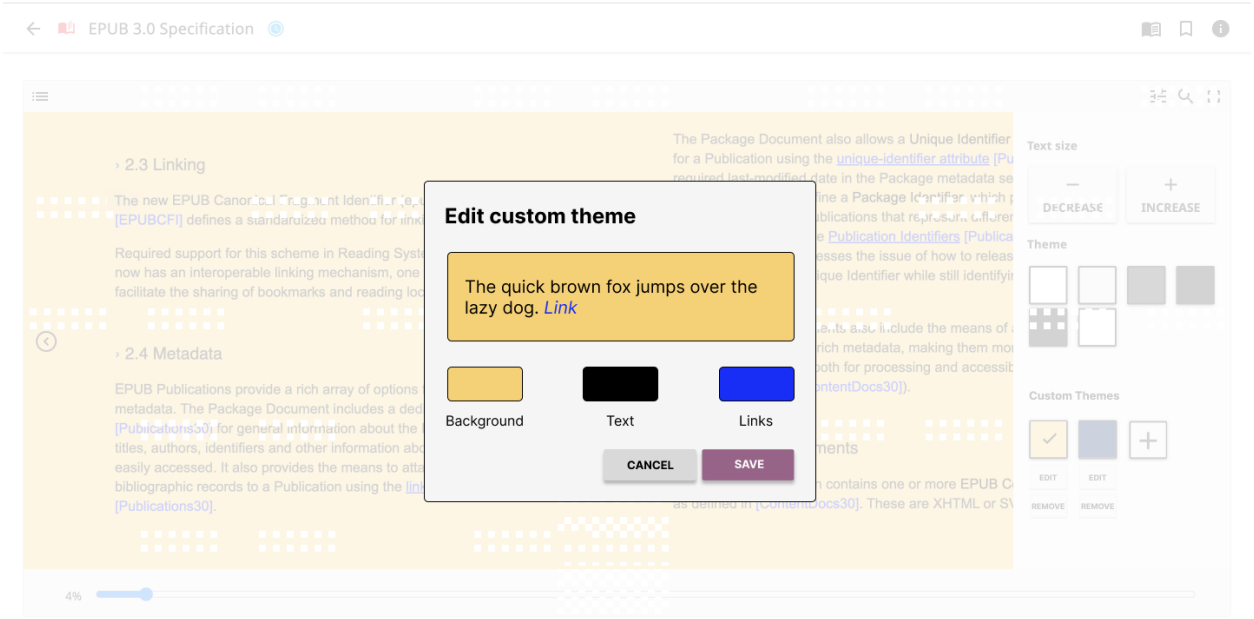
- Expected by Evaluation 1

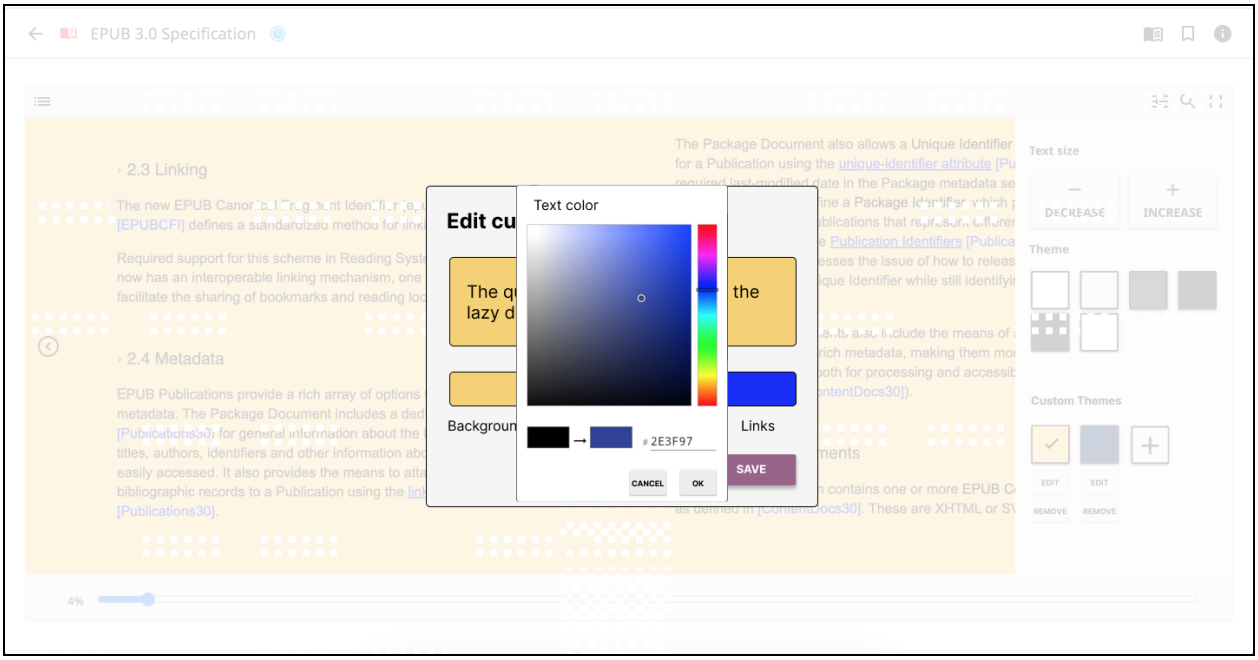


Deliverable 3 - Color pickers for text color, background color, and text color of a custom theme

- Now we have options to add a new theme, and the theme gets applied when selected, next part is to have the option to customize a selected theme. A separate modal is to be used which has the options to select text color, background color, links color using a color picker, and a preview of the theme that changes on the fly of selecting colors as shown in the wireframes below. When adding a new theme, or edit theme option is selected, this modal is supposed to pop up.

- Expected by the end of the coding period


EPUB 3.0 Specification
Edit custom theme
The quick brown fox jumps over the lazy dog. Link
Background
Text
Links
CANCEL
SAVE
Text size
DECREASE
INCREASE
Theme
Custom Themes
EDIT
REMOVE
4%


EPUB 3.0 Specification
Edit cu
Text color
Background
Links
SAVE
CANCEL
OK
Text size
DECREASE
INCREASE
Theme
Custom Themes
EDIT
REMOVE
4%

Deliverable 4 - Feature of making custom themes persistent for a user

- To enhance the convenience of using custom themes, it would be better to have the created custom themes saved per each user. For simplicity, the custom theme details are decided to be stored in the browser with local storage for this project. It leaves the flexibility to store those data in the backend later too. For now, simple JSON objects with the color codes of themes to be stored in the browser and to be retrieved when the EPUB renderer is loaded.

- Sample structure of the JSON object would be;

- Expected by the end of the coding period

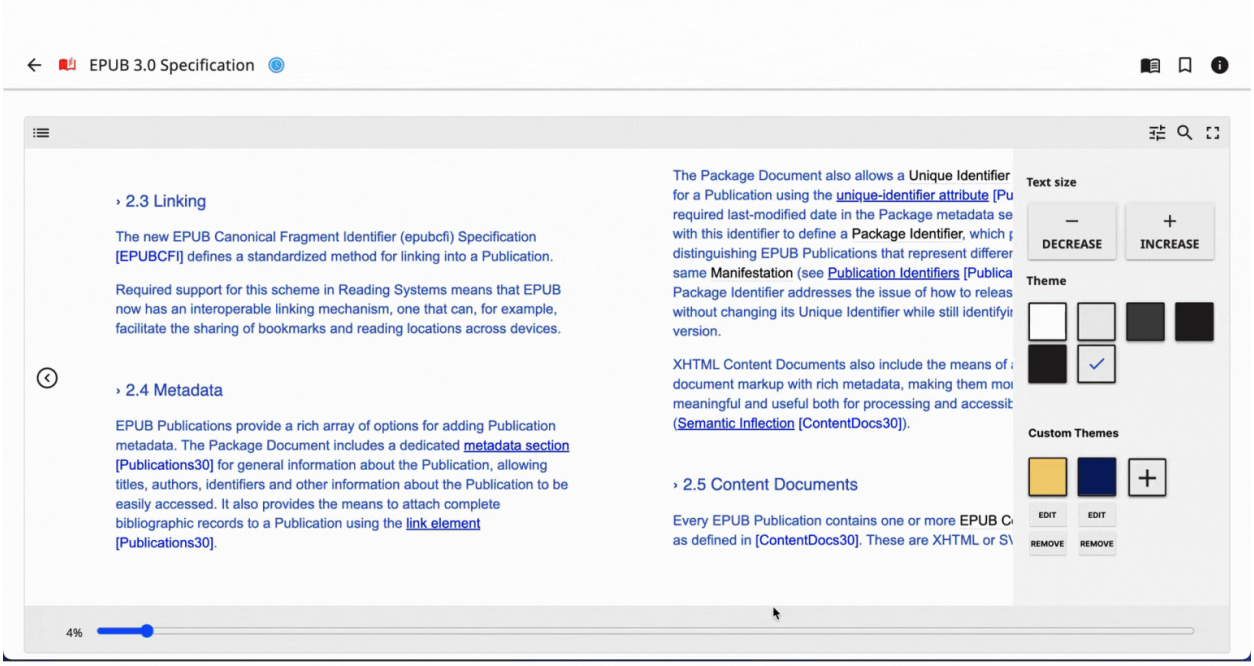
Expected Results

- After the project is finished, the following workflow can be experienced by a Kolibri user.

  - A user signs in to Kolibri, opens a learning material which is in EPUB format.
  - From the EPUB renderer where the learning material is displayed, a user selects one of the in-built themes of his/her preference.
  - If the user is not happy with the in-built themes, he/she clicks on the add item to create a custom theme of his own.
  - The user selects text color, background color, and links color of his choice, and clicks create.
  - A new custom theme gets added to the EPUB renderer settings and the user applies the new theme to the renderer to view the learning material with his own theme.

- If the user prefers, he/she creates more custom themes (up to 4), following the same procedure, and selects any one of the custom themes to apply.
- The user modifies a theme if he/she wishes; or delete custom themes if he/she no longer needs them.

- Following GIF shows the expected functionality of the final product. The prototype can be experienced here. To access the GIF from the PDF file, click here.



Approach

- The deliverables are explained in detail with the figures where necessary. The approach to accomplish the goals is explained better in the timeline.
- The technologies and steps for each deliverable can be explained as follows.
  - Deliverable 1 :
    - To accommodate the custom themes in the EPUB renderer settings, mainly working on the file *'SettingsSideBar.vue'*.
    - Add a *'KFixedGrid'* with 4 columns to accommodate up to 4 custom themes.
    - Themes would be presented as *'KButton's*, and for *edit* and *remove* items also use *'KButton's* with a separate class.
  - Deliverable 2 :
    - To apply the custom themes to the EPUB renderer, mainly working on the files *'SettingsSideBar.vue'*, and *'EpubRendererIndex.vue'*.
    - Add the click event to *'KButton'* and use the *setTheme()* function with the required changes to apply the selected custom theme.

- Deliverable 3 :
    - To select the text, background, link colors separately, a modal to be introduced at the click of edit button. A *'KModal'* similar to the one in *importing a channel from a source* is decided to be used.
    - As the color picker, it is decided to use the *'v-color-picker'* from **VUETIFY** because of its easy integration founded by research. More about the *v-color-picker* can be read here.
- Deliverable 4 :
    - To make the color themes persistent for the user, local storage is decided to be used as the first step.
    - To make the necessary changes and save the themes in local storage, *'useLocalStorage'* from *'@vueuse/core'* is identified as usable.
    - Necessary actions to change the local storage when saving, and removing are to be added as separate functions for clicks of *KButtons Edit*, and *Remove*.

Timeline

The timeline is divided into periods of 2 weeks except for the midterm evaluation week, and the final evaluation week. The tasks to be carried out during each time period are specified with the components to be working on. Even though the timeline is in periods of two weeks, I hope to provide the project update weekly, through an email thread, or through a meeting.

Period	Task
After proposal submission [April 4 - May 4]	- Continue working on Kolibri issues on Github (currently working on issue #10124) - Discuss on what can be improved further about the proposal.
Community bonding period [May 4 - May 28]	- Get to know more about Kolibri and Learning Equality with mentors and current contributors. - Have few online meetings if possible. - Discuss design decisions and approaches in advance to get them refined.

Week 1 and 2 [May 29 - June 12]	- Add *KFixedGrid* with 4 columns - Add *KButtons* for themes - Write tests for *KFixedGrid* and *KButtons*
Week 3 and 4 [June 12 - June 26]	- Add *KButtons* for *'Edit'* and *'Remove'* - Define the css class for edit and remove buttons - Write tests for *KButtons*
Week 5 and 6 [June 26 - July 10]	- Apply *setTheme*() function for click actions in *KButtons* of themes. - Write tests for *setTheme*() of custom themes.
Midterm Evaluation Week 7 [July 10 - July 14]	- Do the changes (if any required) in the *setTheme* function. - Do the changes (if any requested) from the midterm evaluation.
Week 8 and 9 [July 17 - July 31]	- Add *KModal* to change the text, background, link colors, with the preview. - Write tests for *KModal* functions.
Week 10 and 11 [July 31 - August 14]	- Add *v-color-picker* for each button which is there to change text, background, and link colors. - Write functions to change the preview on the fly. - Write tests for *v-color-picker*.
Week 12 [August 14 - August 21]	- Add function *useLocalStorage* to save themes in browser storage. - Add function to retrieve saved custom themes. - Write tests for *useLocalStorage* to save custom themes.
Final Evaluation Final Week [August 21 - August 28]	- Write integration tests for the whole color picker and custom themes feature. - Do the changes (if any requested) before the final evaluation.

About Me

- I'm a Computer Science & Engineering (3rd year) undergraduate at the University of Moratuwa, Sri Lanka. I have managed to maintain an overall GPA of 4.0 out of 4.2 over the 5 semesters I have studied.
- I have completed a number of software projects during my academic career. They have been mentioned in the resume on my Github profile. The concepts and technologies I'm familiarized with are;
    - Object oriented programming and Design patterns
    - Software Development Lifecycle
    - Networking and Computer Security principles
    - Java, Python, JavaScript as programming languages
    - ReactJS, VueJS, Bootstrap, ExpressJS, django as web development tools
    - Docker, Terraform as DevOps tools
- As I have gotten very interested in the Kolibri project, I have tried to work on several issues on Github, and I hope to keep working on many issues in the future as well.
    - Adding an ESLint to Kolibri-tools : issue #9867 , PR #10130 (Merged)
    - Bugfix for UI displaying English as a language option when its not available : issue #10124 (Currently working on)
    - Improve user experience when selecting lesson recipients in Coach : issue #9097 (Engaged through comments and hope to work on this next)
    - Two inputs are focused at the same time : issue #9077 (Engaged through comments)
    - Resources cards are not optimized for small screens : issue #9058 (Engaged through comments)
- It would be a huge opportunity for me if I get a chance to participate in GSoC 2023 as I'm finishing my degree program next year and am Hoping to engage in a role of full time software engineer. I hope to gather a few ideas for my final year project from these projects while working on them.
- I have chosen Learning Equality because it drew my attention from the first sight as a Learning platform. As a Sri Lankan, I feel the necessity of having a learning platform that can be used offline in rural areas because of the current situation in the country. Therefore, I decided to support this project, and to make people aware of these tools in my potential.
- The project I have chosen to work on is a medium scale project that requires 175 hours of work. As I have planned, sparing about 2 hours a day for the 12 weeks would suffice and I would definitely be able to allocate that amount of time per day managing my other commitments. As a well organized student, I believe I have the potential to deliver the project within the timeline with all of my skills.
- Google doc of the proposal is linked here for more interactive recordings (GIFs).