Implementing Versioning, adding Favourite Components, and Implementing Component Properties Popup

On

eSim on Cloud

Submitted by

Rajat Maheshwari

Under the guidance of

Prof. Kannan Moudgalya
Chemical Engineering Department
IIT Bombay

June 2021

Acknowledgement

I, the summer intern of the **FOSSEE - eSim Cloud Project** am overwhelmed in all humbleness and gratefulness to acknowledge my deep gratitude to all those who have helped me put my ideas to perfection and have assigned tasks well above the level of simplicity and into something concrete and unique. I wholeheartedly thank **Prof. Kannan M. Moudgalya** for having faith in me, selecting me to be a part of his valuable project and for constantly motivating me to do better. I thank **Mr. Nagesh Karmali** and **Ms. Firuza Aibara** for providing me the opportunity to work on this project. I am also very thankful to my mentors for their valuable suggestions. They were and are always there to show me the right track when needed help. With help of their brilliant guidance and encouragement, I was able to complete my tasks properly and were up to the mark in all the tasks assigned. During the process, I got a chance to see the stronger side of my technical and nontechnical aspects and also strengthen my concepts. Last but not the least, I sincerely thank all my other colleagues working in different projects under **Prof. Kannan M. Moudgalya** for helping me evolve better with their critical advice.

Declaration

I declare that this written submission represents my ideas in my own words and whenever others' ideas or words have been included, I adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this thesis.

I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/-source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken when needed.

Rajat Maheshwari

Contents

List of Figures

Chapter 1

Introduction

The current system allows the users to draw analog and digital circuits and simulate them. The users have a facility to drag and drop components from the left pane onto the schematic grid on the right pane. The components on the grid are connected using wires. The circuit can then be simulated using the different simulation parameters(DC Solver, DC Sweep, Transient analysis, and AC analysis). The basic ERC check enables the users to find out errors if any. The size of the schematic grid can be changed from A1 to A5 paper sizes along with portrait and landscape modes. The users can also print the circuit or save it in pdf format for documentation purposes.
Apart from this, contributors can create circuits and send them for review for publishing them on the platform for all to see.
All of this code is hosted in the repository available on GitHub

1.1 Problem Statement

Extending the existing functionality of the existing eSim on Cloud project, by adding features like Versioning and Favourite Components. Additionally made the Component Properties Tab, a popup next to the component itself, so as to make it easier to integrate with other aspects of the project.
Adding these features increase the usability and gives a better experience to the user. For example, earlier circuit maintenance was non-existent, as saving the circuit again resulted in previous changes being lost. With versioning, features like reverting to an older version coupled with making new variations results in proper segmentation of the different features a circuit has to offer, reducing the chances of overlapping and confusing circuits.

1.2 Project Objective

This project aims to provide quality improvements and increase ease of use of the existing system. Adding features like versioning increases the freedom a user has while working on the system, coupled with the facility to add favourite components,

increases the productivity of the user as he can focus on making the circuit rather than searching for the components.

1.3 Project Outcome

The project adds facility to create versions of a circuit along with variations of the same circuit. Earlier, saving a circuit would result in the earlier changes being overwritten, this resulted in needing the user to be incredibly careful about what he did. Now the user can create versions as he wants freely. Another facility that is implemented is the ability to add favourite components for users, which are displayed as a drop down list Lastly, I have made the Component Properties tab a fully draggable popup component, opening next to the component which was clicked on.

1.4 Project Requirements

Following Major Technologies have been used during development.

- Django (v2.2.12)
- React (v16.13.1)
- PostgreSQL

Chapter 2

Project Overview

The implemented features allow the user to add favourite components, to quickly access and use them without scrolling through a large list of components. The user can also create various versions of a circuit. With versioning the user has the freedom to try new things as he wishes, without the fear of losing progress. He can create various variations, commonly known as branches, on sites like GitHub. The user can save the circuit, create a new variation to try new ideas as he pleases, and can go back to any version whenever he pleases. This is especially useful as the user can select and add components to his favourites so that he doesn't need to waste time searching for specific components, he can quickly access the components he uses frequently. This is displayed as a drop down list, with the other components under the favourite component tab. Earlier, the Component Properties tab was situated on the right hand side of the screen, double clicking on a component allowed you to set the properties of that component. I have made the Component Properties a popup component.

2.1 Features

My task was divided into four main parts, Favourite Components, Versioning, Component Popup, and integrating Versioning with Workflow. More details are given below.

2.1.1 Favourite Components

The user can decide from a list of components which one he wishes to add to favourites, it works like any other adding to favourites facility available on any website. The drop down menu automatically re-renders each time the user makes a change to the favourites. It also un-renders when all favourite components are deleted and will not load till something is filled in it.

2.1.2 Versioning

Like Git, the project now supports versioning along with support for variations, commonly known as branches, in Git terminology. Versioning allows multiple changes

to a circuit which is useful in comparing the steps a circuit took to get completed, it allows for greater variety, as some mistake wouldn't result in going back to square one, and repeating the circuit creation again. He has more freedom to experiment, and try different variations of the circuit before settling on the one he likes. Creating a version, results in the circuit being saved at that point, with the page refreshing to show the new versions. Creating a variation(branch), is instant and is reflected instantly on the bottom right hand side of the screen. The various versions and variations are ordered in descending order of save time. Users can navigate to any circuit by clicking on any of the version he wishes.
The user can name his variation as he wishes, so that they are identified uniquely according to the features implemented in it. The user can also delete specific versions, and delete entire variations as well. Deleting variations deletes all of the versions contained underneath it.
Apart from this, the entire circuit can be deleted, from the dashboard, which will result in all of the variations and versions in it being deleted.

2.1.3 Component Popup

Earlier, to set the properties of a component, double clicking on it opened the properties tab replacing the component position tab. Now, the popup opens next to the component being clicked, this makes it easier to understand which component's properties are being set, and making it draggable allows the user to understand what the circuit is doing before setting the values. Additionally, this made it easy to integrate the Schematic Editor with the Workflow/Project aspect of the problem statement. This also fixed a bug with the zoom of the component position not being reset, as the component is not removed and rendered back onto the page anymore, it is now always present on the screen

2.1.4 Integrating Workflow with Versioning

Workflow was another aspect of the project, where the user can create a project, and a reviewer can publish it. This was made on an earlier version of eSim, which didn't support Versioning. Rugved and I handled the integration of workflow and versioning, which made the final result more powerful.
A contributor can create various versions and can select from a drop down menu of all the versions he has created to pick the one he would like to be published. Resolving conflicts was an important aspect of this integration.

Chapter 3

Feature Implementation

The following chapter describes the procedure I had undertaken to implemented the above listed features:

1) Favourite Components

2) Versioning

3) Component Properties Popup

3.1 Adding Favourite Components

The existing library model already contained the list of all components in the eSim project. I added a new model for supporting favourite components of a user. This model contains a Many to Many field referencing the components. This results in an list of the favourite components of a user.
This is served by a new endpoint called favouritecomponents, which returns a list of the favourite components of an authenticated user.
Favourite Components can be added by sending a post request with the ID of the component which needs to be added, again, this is only accessible to users who are authenticated.

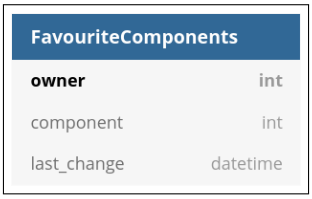

Figure 3.1: Schema Diagram of Favourite Components

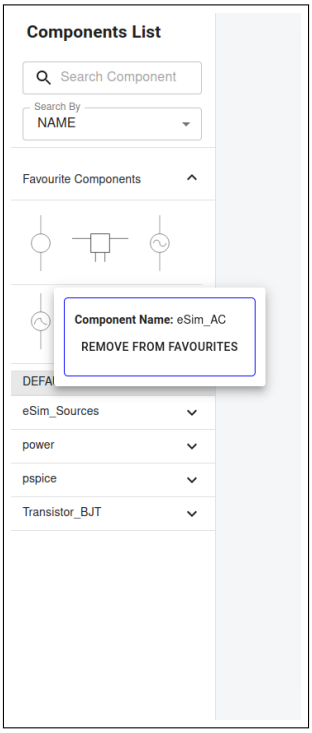

Figure 3.2: Remove From Favourites

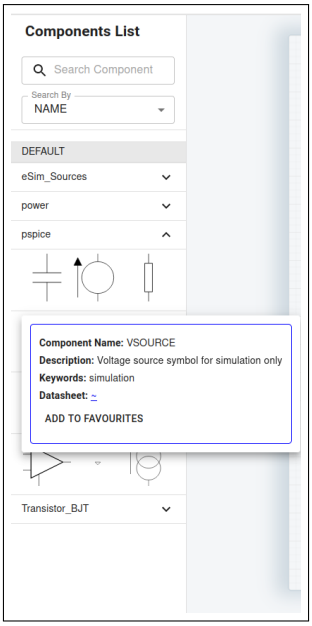

Figure 3.3: Add To Favourites

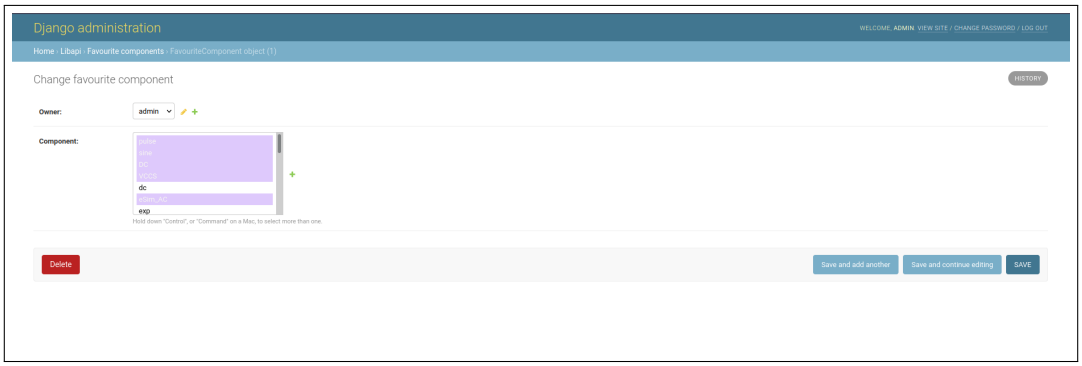

Figure 3.4: Admin Panel Look of Favourite Components

3.2 Versioning Support

Saving earlier resulted in overwriting of the earlier saves with the new ones. This resulted in there being only one absolute source of the circuit. Now, with versions and variations, the users can make checkpoints, and go in different directions as he pleases.

Prior to versioning, the save_id was a primary key, hence, it was unique always, editing this field to be a non unique key allowed me to save various versions of the same circuit, with the same save_id. The save_id is like a git repository, which contains all the information regarding the versions of the circuit.

To support versioning, the save model was edited, to have extra attributes with versions and variations.

Saving a circuit results in a new version being created with a random string as the version hash. Creating a new variation, prompts the user to give a new name to his variation, which is then associated with that version of the circuit.

On the bottom right of the screen, the history of the circuit is shown. The variations are a drop down menu, the versions related to that specific variation are listed in the drop down menu which toggles when clicked.

Figure 3.5: Schema Diagram of Versioning

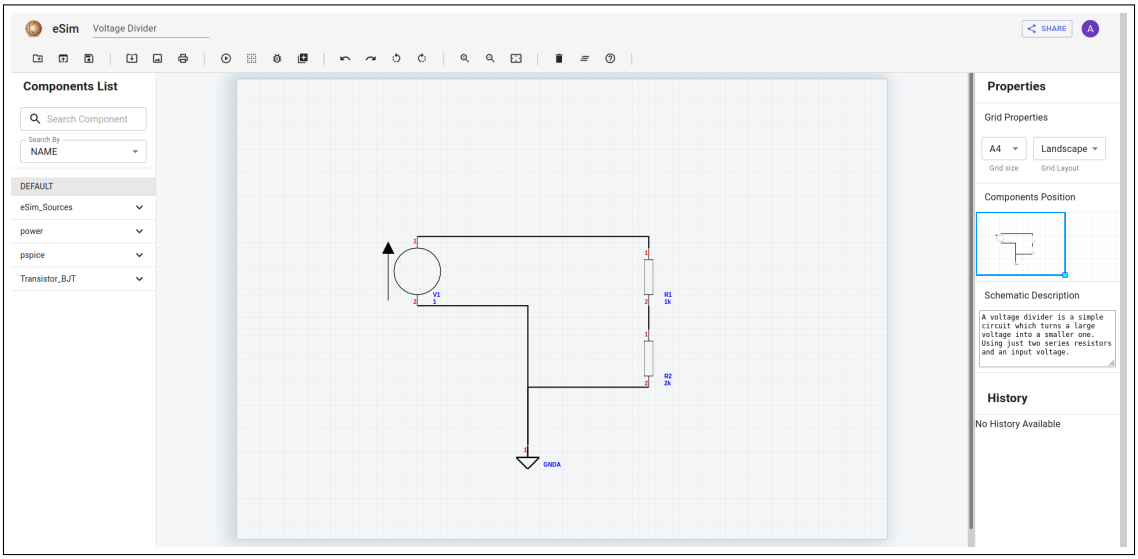

Figure 3.6: Schematic with no History saved

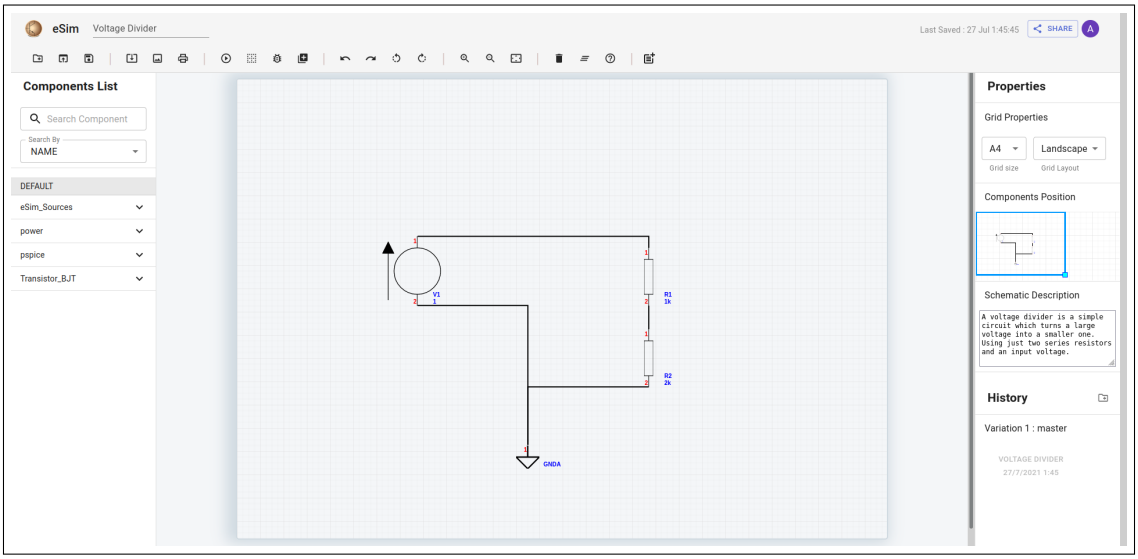

Figure 3.7: Schematic with one Version

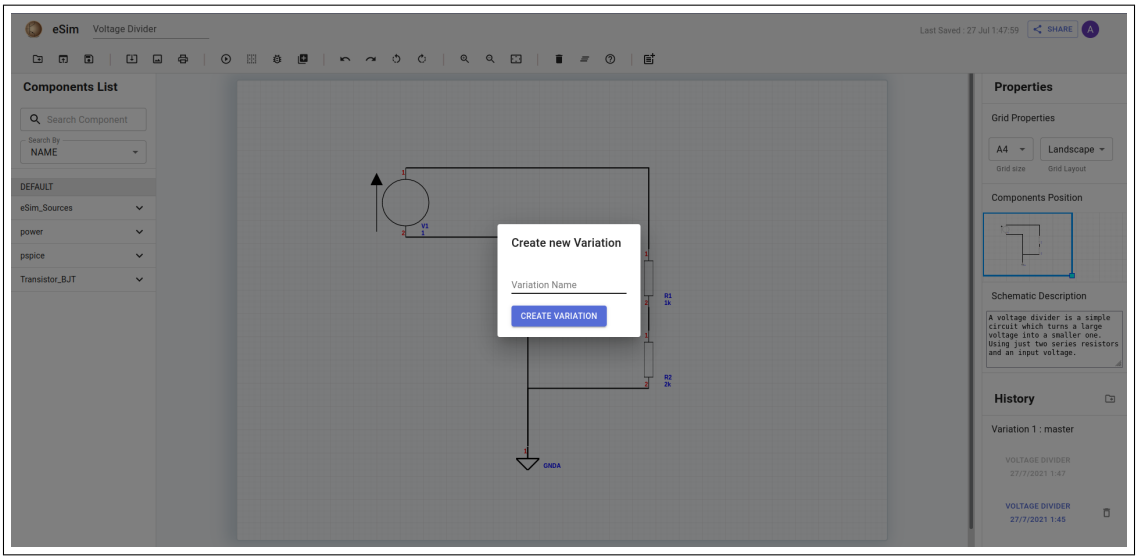

Figure 3.8: Popup for creating new Variation

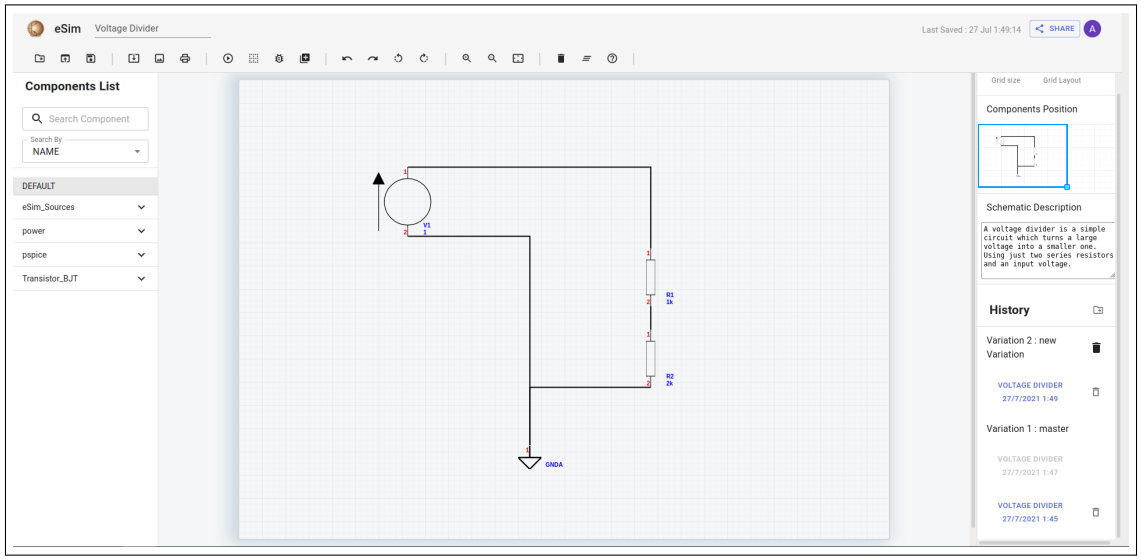

Figure 3.9: Schematic with two Variations

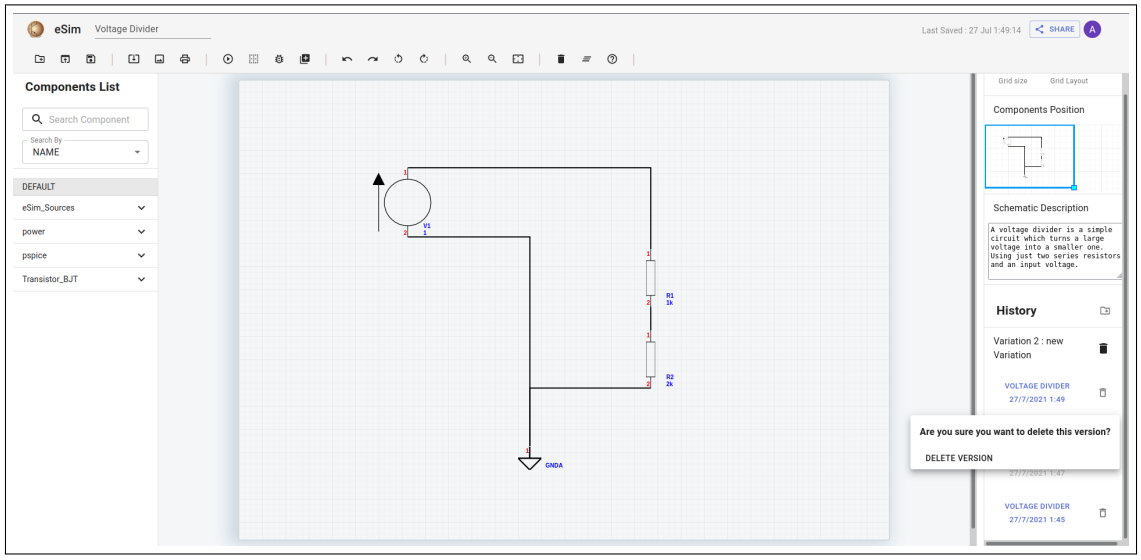

Figure 3.10: Delete a Version

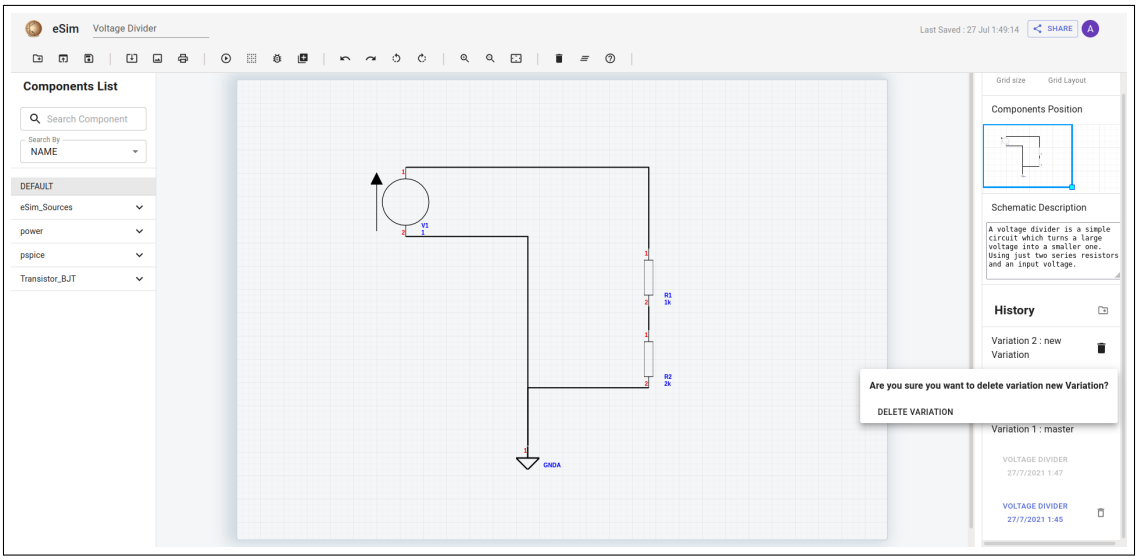

Figure 3.11: Delete a Variation

3.3 Component Properties Popup

Opening the Component Properties in a specific area of the editor made other components invisible to make space for this tab. With the Component Properties now being a popup, it opens next to the component.
Using geometry, I found the rendering position of the component properties tab, it is also scrollable, so exceeding a maximum height of the properties, will give a scroll bar, to fit all of the properties of that particular component.
The popup is also completely draggable within the confines of the screen, this is especially useful when integrating with Workflow, as it results in the reviewer being able to explore every aspect of the project and not have to divert away from the circuit to check the values.
This resulted in a bug fix, where the Component Position wouldn't reset when the properties tab was closed, resulting in the user not being able to navigate the circuit.

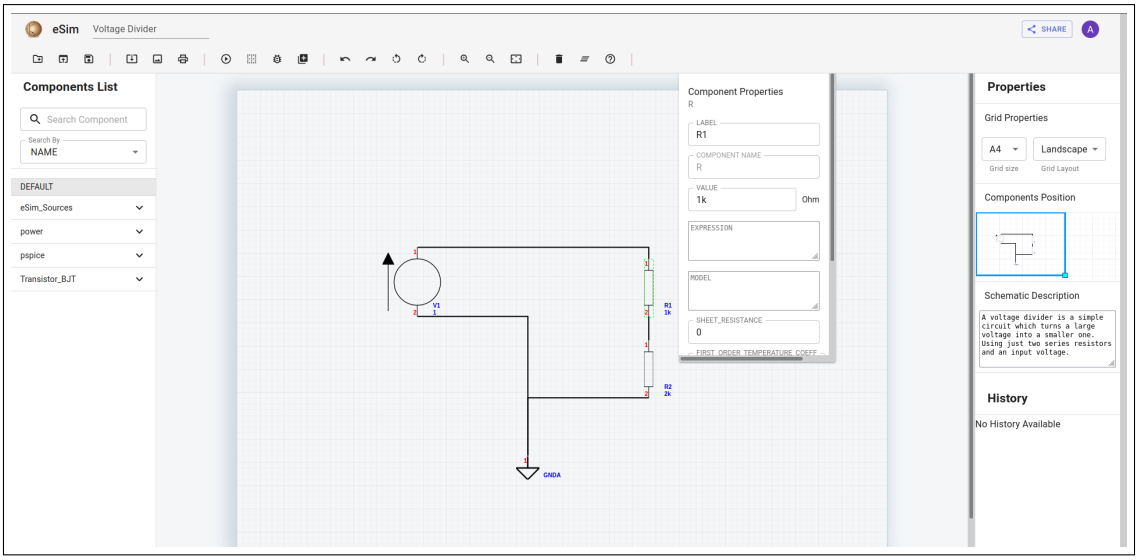

Figure 3.12: Component Popup in default location

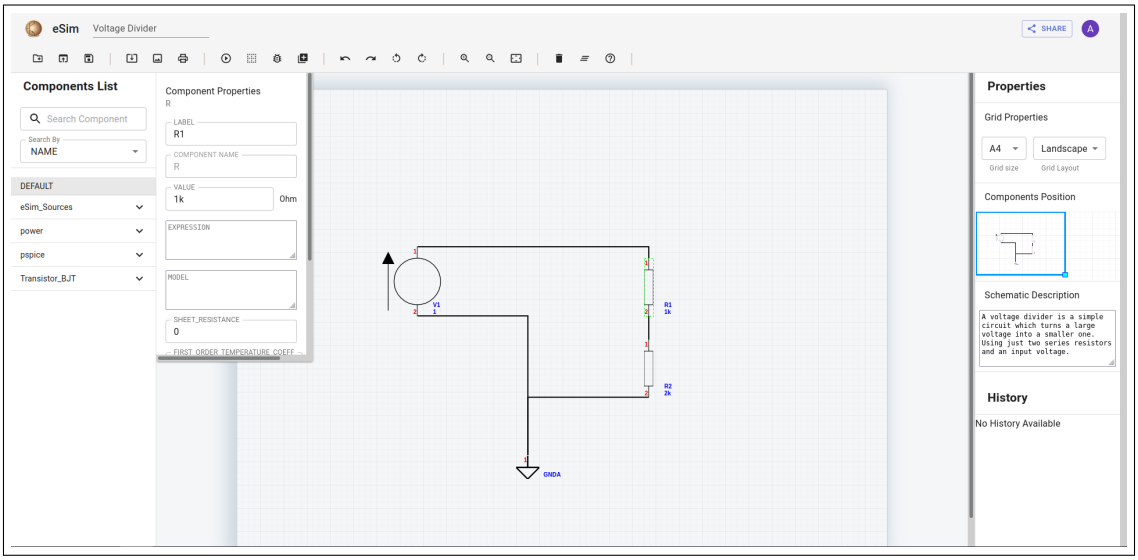

Figure 3.13: Component Popup dragged to the left

Chapter 4

Conclusion

Within the duration of my internship, I have learnt a lot of new technologies and I have done my best to implement my tasks to the best of my ability.
I believe that with the features I have implemented, the project will benefit, as they make the user interaction aspect a lot simpler and intuitive. Adding versioning was a crucial feature as it gives more freedom to users to try out whatever they wish to do.
Apart from this, versioning also allows the LTI implementation to work flawlessly, as having multiple versions allows the test setter to have a wider range of selections to pick from right on the same page without having to navigate many circuits by repeatedly going back to the dashboard.
Versioning also enables reverting to a previous save, which in turn allows users to make mistakes without the fear of having to start all over again.
Adding favourite components is a time saver as it allows users to pick out specific most used components rather than spend time scrolling through the many components eSim has to offer.
These simple improvements result in a much better user experience than before, which is always a main goal in a project of this scale.

Chapter 5

Future Work

There is a minor improvement in versioning which can be made .This involves creation of a new version. While creation of a branch is instant, creating a version requires a page reload to clear the stored variables otherwise it results in a bug where some components would duplicate, a minor improvement would be to make version creation instant as well.
One way to achieve this is clearing the dependencies of the useEffect hook of the properties sidebar and resetting them to the same values to simulate a page reload, but only for the sidebar component.

Bibliography

[1] https://www.django-rest-framework.org/ (Accessed on August 1, 2021).

[2] https://www.djangoproject.com/ (Accessed on August 1, 2021).

[3] https://swagger.io/ (Accessed on July 31, 2021).

[4] https://reactjs.org/ (Accessed on July 30, 2021).