Semester-Long Internship Report

On

Adding LTI Support, Simulation History and User management.

Submitted by

Akshat Sharma

Under the guidance of

Prof. Kannan Moudgalya

Chemical Engineering Department

IIT Bombay

April 2021 - August 2021

Acknowledgement

I, Akshat Sharma, summer intern of the **FOSSEE - eSim Cloud Project** are overwhelmed in all humbleness and gratefulness to acknowledge our 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 was 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 our 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.

Akshat Sharma

Contents

1 Introduction **5**
- 1.1 Problem Statement . . . 5
- 1.2 Project Objective . . . 5
- 1.3 Project Outcome . . . 5
- 1.4 Project Requirements . . . 6

2 Project Overview **7**
- 2.1 Features . . . 8
  - 2.1.1 LTI Support . . . 8
  - 2.1.2 Simulation History and Comparison . . . 8
  - 2.1.3 Improving User Experience . . . 8
  - 2.1.4 Improving Graph Result Screen . . . 8
  - 2.1.5 Integrating LTI with workflow and versioning . . . 8
- 2.2 DevOps . . . 9
  - 2.2.1 Migrating production server to new DB schema . . . 9

3 Feature Implementation **10**
- 3.1 LTI Support . . . 11
- 3.2 Simulation History and Comparison . . . 20
- 3.3 Improving Graphs in Simulation Screen . . . 22
- 3.4 Improving User Experience . . . 22

4 Conclusion **23**

5 Future Work **24**

List of Figures

3.1 Simulation history 11
3.2 Simulation history 12
3.3 Simulation history 13
3.4 Create LTI App button in Schematic Card 14
3.5 LTI Create Page 15
3.6 LTI Create Page after an LTI app is created 15
3.7 eSim embedded in LMS using LTI standard 16
3.8 Simulations being ran on eSim through LMS 16
3.9 Student can select from all his/her simulations for submission 17
3.10 Submission of a simulation whose result is of graphical nature 17
3.11 Submission where all the values of the student are correct 17
3.12 Submission where all the values of the student are wrong 18
3.13 Submission where student was partially correct 18
3.14 Submission where only some params where considered for grading . . 18
3.15 All student submission being shown to the teacher 19
3.16 Student submission being shown to teacher 19
3.17 Student submission(graphical in nature) being shown to teacher . . . 20
3.18 Simulation history 20
3.19 Simulation history 21
3.20 Comparing tabular simulations 21
3.21 Comparing graphical simulations 21
3.22 Download graph output 22

Chapter 1

Introduction

1.1 Problem Statement

- Add LTI support for circuits made on eSim.
- Add simulation history of circuits and comparison of simulation result.
- Fix the graph output for simulations and add support for downloading output of graph.
- Make the existing simulator compatible with desktop Ngspice syntax.
- Add reset password and forgot password feature for better user experience.

1.2 Project Objective

This project aims to provide quality improvements and increase the numerous use cases of the system. Addition of features like LTI helps teachers using the system to easily embed circuits they've created on the platform onto their preferred LMS without any hassle. Addition of circuit simulation history and comparison helps users to compare their previous simulations and figure out the simulation/component parameters that give out the correct output.

1.3 Project Outcome

A user will be able to embed his/her circuits in a LMS after saving it in his/her account. The user can also setup a test-case that can be used for comparing it with a student's simulation to check whether the parameters selected by the student is correct or not. A user can compare his/her circuits' simulations and go through all the simulations that have been ran on a particular circuit. Along with this password changing and forgot password functionality was added which would help the user get access of his/her account in case they want to change their credentials.

1.4 Project Requirements

Following Major Technologies were used by me during development.

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

Chapter 2

Project Overview

LTI allows the user to use the platform as an assignment creation tool and embed any circuit they want in their preferred choice of LMS(for example- Moodle, OpenEdx, Canvas, etc.). This helps the teacher focus on creating quality education resources for their students without worrying about how to give them access to that resource. After assignment creation, the teacher can have a look at the submissions, that have been made for that particular assignment, on the platform itself along with the details of the user who made the submission, the actual submission and from where the submission was made from.

Another feature implemented is that of storing simulation history and giving user the option to compare current simulation with a previous one. This allows the user to compare how their circuit has been performing and accordingly change their simulation parameters and/or component parameters to reach their desired output from that circuit. The simulation history feature has also been implemented in the provided NgSpice Simulator on the platform and can be utilised there based on type of simulation.

Earlier, the users where not provided with the option to either change their password if they want to or even get a change password email in case they don't remember their password. This feature was promptly implemented for better user experience and interaction.

2.1 Features

My task was divided into five main parts, adding LTI support, simulation history and comparison, improving user experience, integrating LTI with workflow and versioning and maintaining production server and migrating to new database schema with persistent data. More details are given below.

2.1.1 LTI Support

Learning Tools Interoperability, also known as LTI, is a way in which an LMS (learning management system like edX, Moodle, Canvas, Blackboard, etc. addressed as LTI Consumer) communicates with external tools/systems/utilities, addressed as LTI Producer (eSim on Cloud). To a normal user, it appears as if the external tool is embedded into the LMS in a frame. LTI(1.0/1.1) was implemented in the platform along with auto grading of circuits. After assignment creation, the teacher can have a look at the submissions, that have been made for that particular assignment, on the platform itself along with the details of the user who made the submission, the actual submission and from where the submission was made from.

2.1.2 Simulation History and Comparison

Storing the simulation history of a particular circuit along with its netlist and results was a feature that was heavily required on the platform that would enable the users to compare and arrive at a simulation that they feel is good enough for their use case. The storing of simulation history of both a schematic and that from the NgSpice Simulator on the platform makes the process of circuit simulation all the more easier and adds better user experience to the platform.

2.1.3 Improving User Experience

The user can now use the features forgot password and reset password to change his/her password according to his/her liking. Passwords provide the first line of defense against unauthorized access to one's account and personal information and it is necessary that users get the option to manage their passwords.

2.1.4 Improving Graph Result Screen

The user can now download the plot points of graphs in the form of a CSV(Comma Separated Values). Along with this the graph is now auto-scaled according to the values of plot points.

2.1.5 Integrating LTI with workflow and versioning

Workflow was another aspect of the project, where the user can create a project, and a reviewer can publish it. Apart from this, another big change to how schematics were saved was introduced in versioning where the user can create multiple versions and variations of a circuit. This added more complexity to how an LTI app could

be created. Rugved and Rajat assisted me in integrating LTI with workflow and versioning. A user 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 created as an LTI app. An additional check was added where a project can't be an LTI app and vice-versa. Resolving code as well as logical conflicts was an important aspect of this integration and took almost 2-3 days of work.

2.2 DevOps

2.2.1 Migrating production server to new DB schema

There were several changes to the database tables along the way which required us to migrate existing data on the server to the new tables.

- Migrating the user model in which the email field was changed was an issue. According to Django's table naming scheme the already existing user table would be removed which would cause data loss. This was handled by doing some intermediary changes before running migrations that would rename the table names for us. This fixed the issue of migrating the user model. [1]

- Migrating the schematic model and and an auto generated Django table was required when versioning was introduced. This was required because the primary key was changed in the schema but, there was data that needed to be migrated to the new schema. This was done by running some steps intermediary steps and changing the postgres tables a bit.

Both of these migrations are documented and stored in the patch folder on the main repository of eSim-Cloud.

Chapter 3

Feature Implementation

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

1) LTI Support

2) Simulation History and Comparison

3) Improving Simulation screen

4) User Experience

3.1 LTI Support

Adding LTI support required me to introduce some new models to the system and modify some existing ones as well.
Our platform needed us to have a new consumer and secret key for each circuit so a model was introduced named lticonsumer that contained a consumer and secret key field along with a foreign key to the schematic that it referenced to. Along with these the model also a score field that would contain the maximum marks that can be assigned to that particular assignment(between 0 and 1) and a boolean field that would be used to indicate whether the circuit was to be used as a demonstration or as an assignment. It also has a testcase foreign key that references to a simulation that will be used to auto grade and simParams field that will hold the parameters that the student circuit will be judged on.

Figure 3.1: Simulation history

Adding LTI support also required us to store the LTI params that were sent from

the LMS with the auth request. For this a model named ltiSession was introduced that has some character fields which are required while authenticating the LMS request.

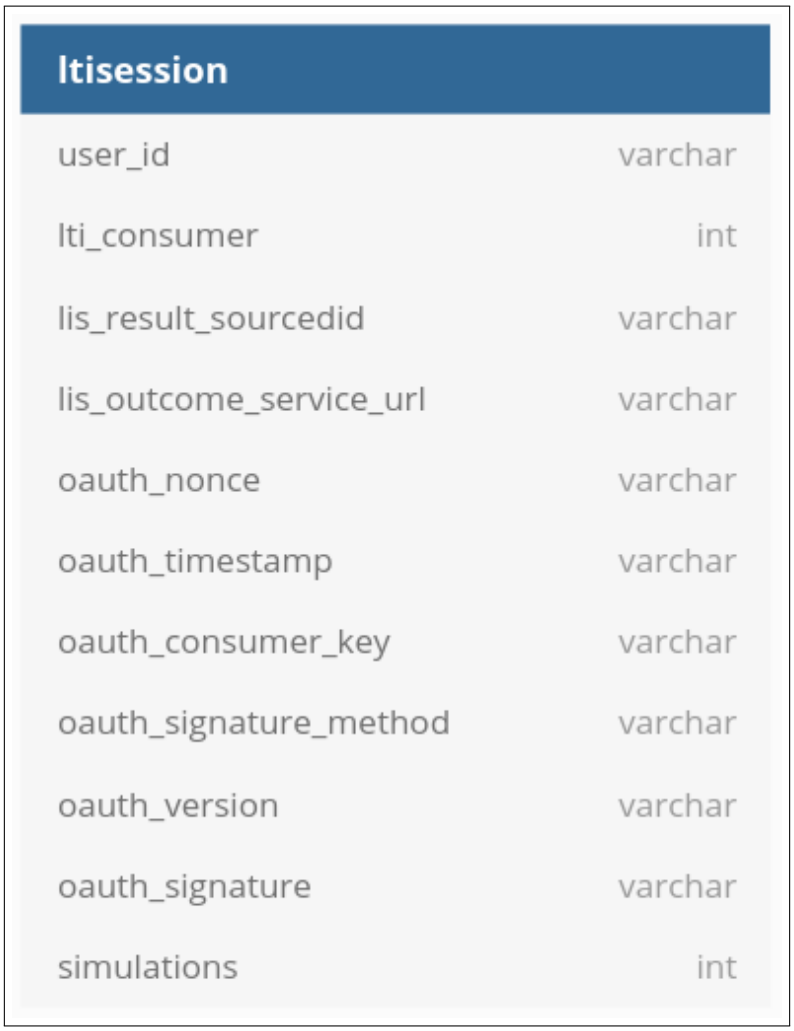

Figure 3.2: Simulation history

Assignment submission required us to store submission related details such as the consumer it was submitted for, the student that created that submission, the session from which the submission was made and whether or not the score was successfully returned to the LMS or not. For this a model named Submission was introduced.

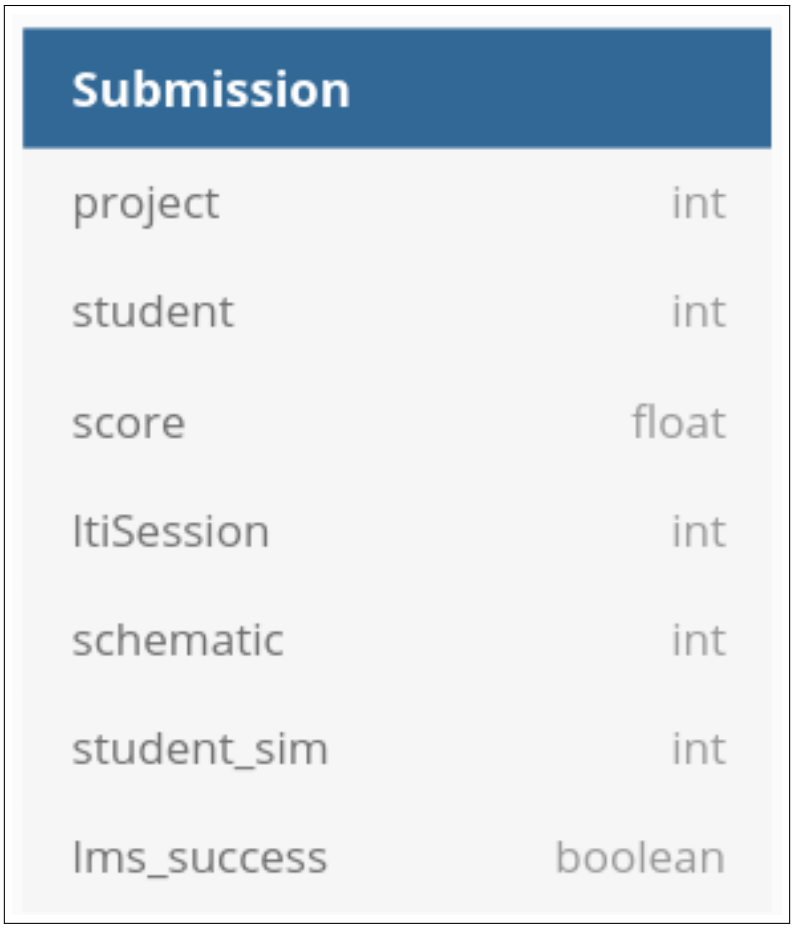

Figure 3.3: Simulation history

The workflow for LTI support was as follows:

- A teacher needs to create a circuit and save it on his/her account on the platform. This can be done on the dashboard page through the button on the schematic card. Doing this will redirect the teacher to a page that can be used to create an LTI app with different versions of that schematic and also set the consumer and secret key and the other parameters that are required for creation of an LTI app. Upon clicking the 'Create LTI App' button, the

  `api/lti/build/`

  route is hit with the request.

- After creation the teacher has options to update the details using the 'Update' button which hits on

  `api/lti/update/`

- There is also a delete LTI app button that sends a delete request to

  `api/lti/delete/<id>`

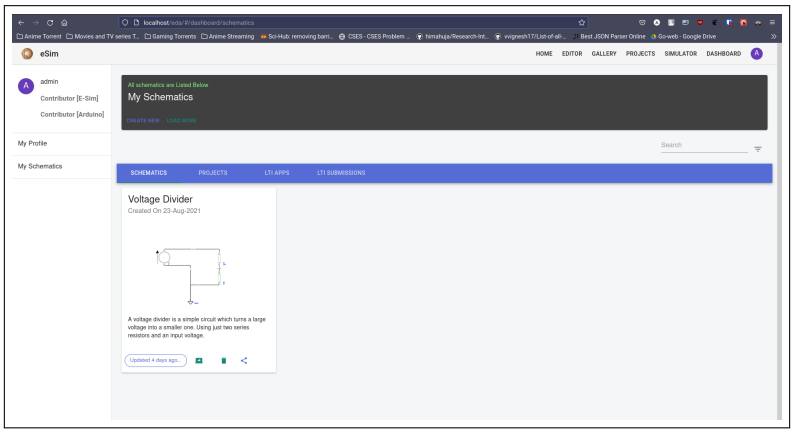

Figure 3.4: Create LTI App button in Schematic Card

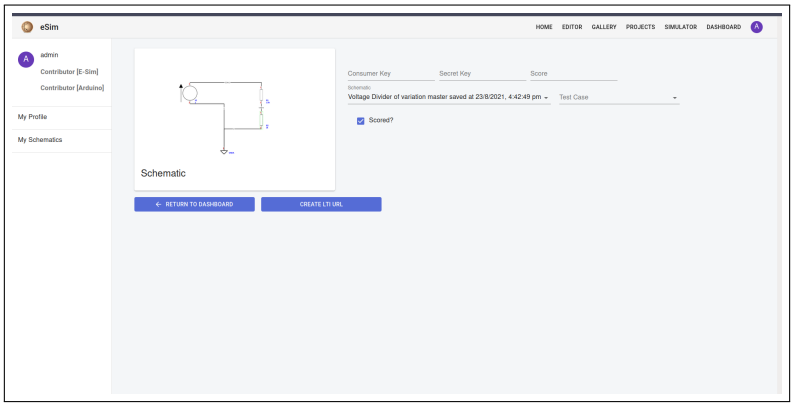

Figure 3.5: LTI Create Page

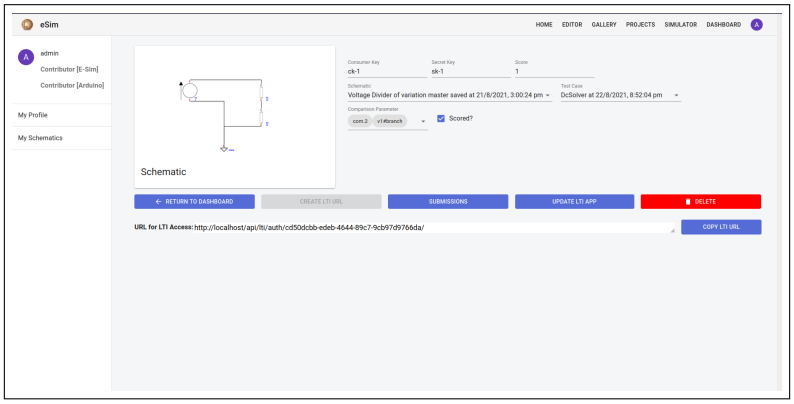

Figure 3.6: LTI Create Page after an LTI app is created

- Upon creation of LTI app, the teacher needs to enter the consumer and secret key and the config url into the LMS External Tool.
- Whenever the assignment is accessed through the LMS, a request is sent to

  `api/lti/auth/<save_id>`

  which authenticates the LMS and request and accordingly sends the resource URL that needs to be rendered in an iframe inside the LMS.

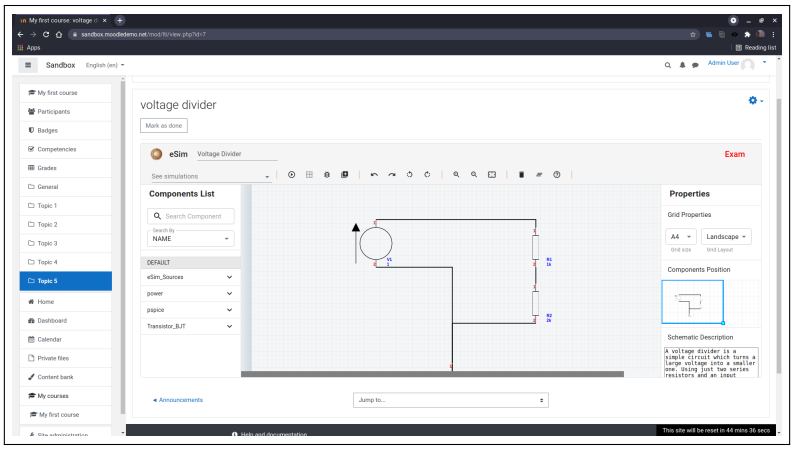

Figure 3.7: eSim embedded in LMS using LTI standard

- Whenever a submission is made through the LMS, a request is sent on

  `api/lti/submit/`

  upon which score is calculated and then the LMS is sent the score that was calculated.

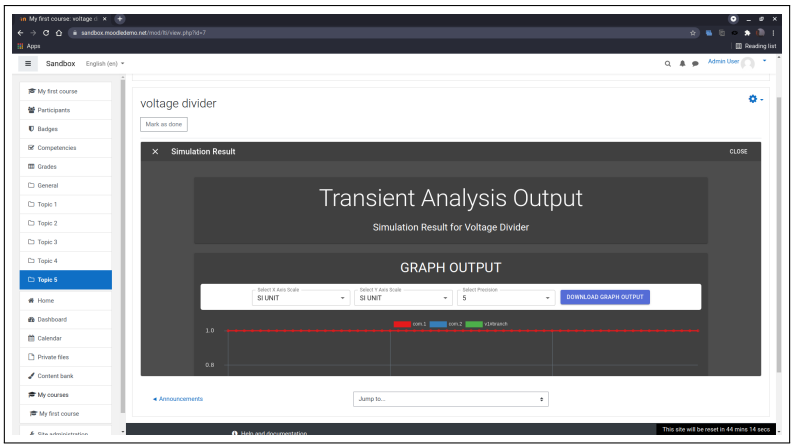

Figure 3.8: Simulations being ran on eSim through LMS

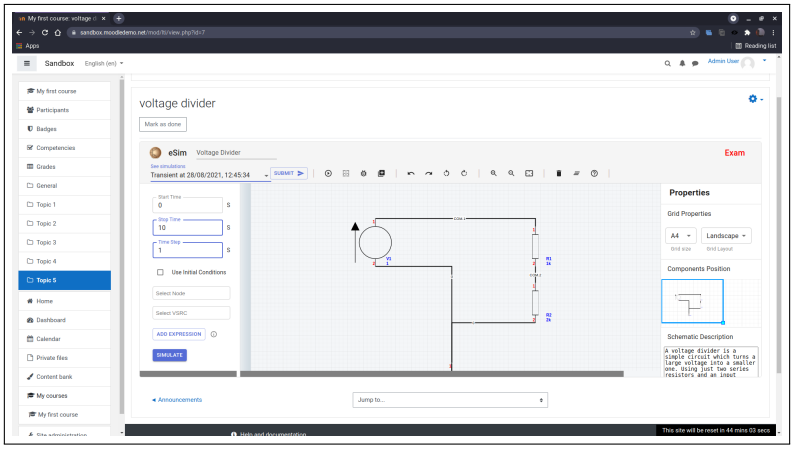

Figure 3.9: Student can select from all his/her simulations for submission

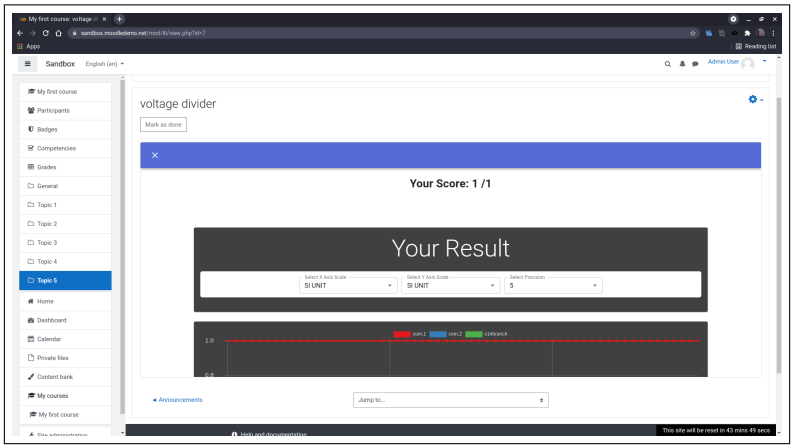

Figure 3.10: Submission of a simulation whose result is of graphical nature

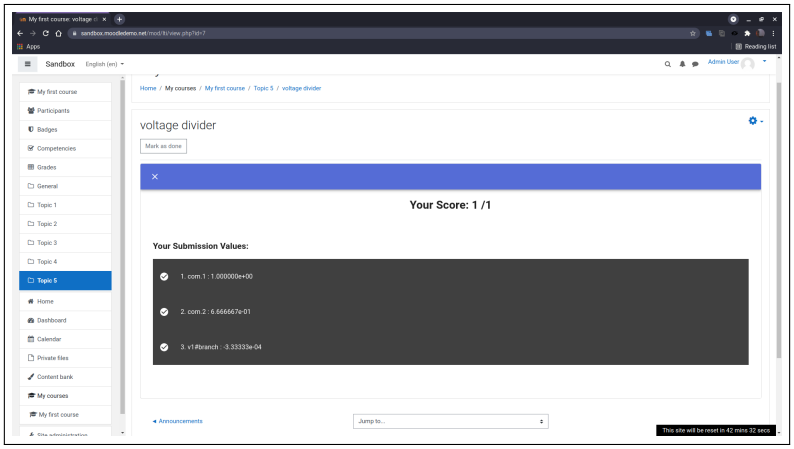

Figure 3.11: Submission where all the values of the student are correct

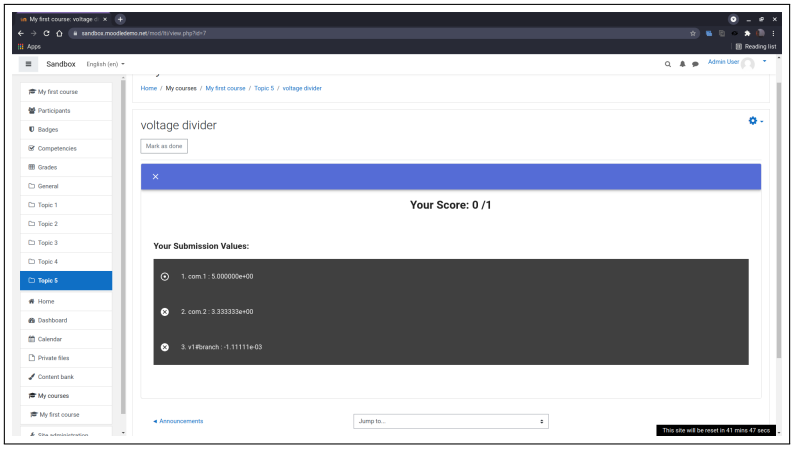

Figure 3.12: Submission where all the values of the student are wrong

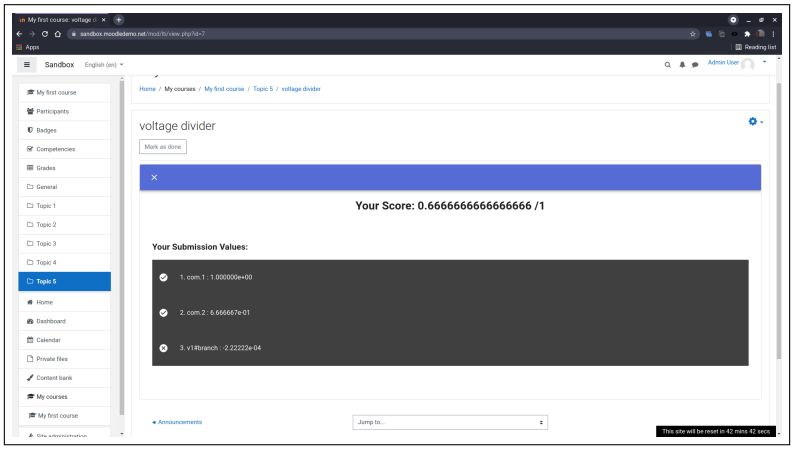

Figure 3.13: Submission where student was partially correct

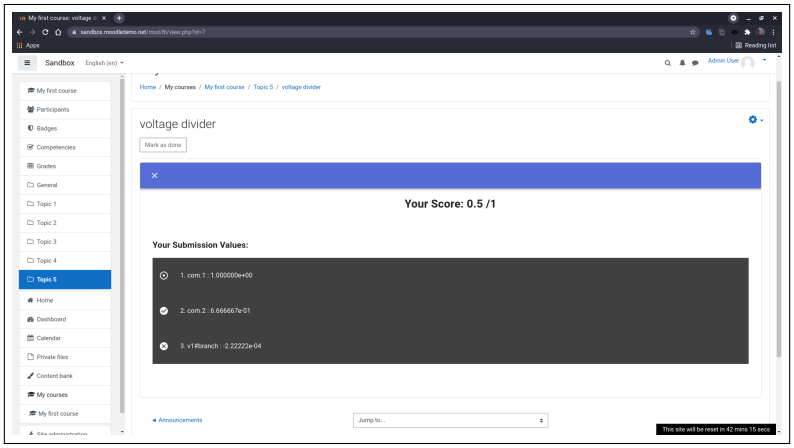

Figure 3.14: Submission where only some params where considered for grading

- A submission button is present on the create LTI app page that redirects the teachers to a page which contains all the submissions for a particular LTI app. As soon as the page is loaded, the endpoint

  `api/lti/submissions/<save_id>/<version>/<branch>`

  is hit which returns all the submissions for that particular LTI app. The teacher can look at all the metadata of the submission along with the student simulation and circuit.

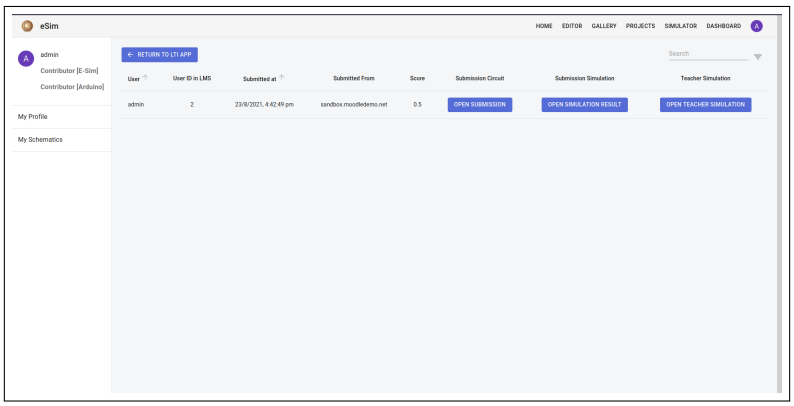

Figure 3.15: All student submission being shown to the teacher

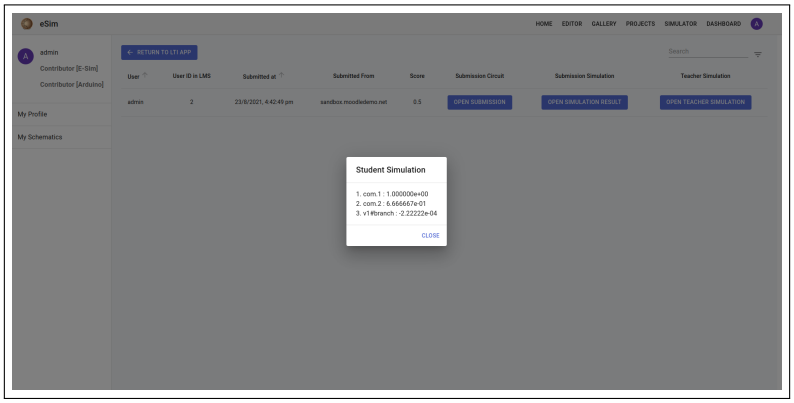

Figure 3.16: Student submission being shown to teacher

Figure 3.17: Student submission(graphical in nature) being shown to teacher

3.2 Simulation History and Comparison

Our platform needed us to have a simulation history which prompted the creation of a new model which had the fields of simulationType (storing the type of simulation), taskID (a foreign key which stored the taskID of the celery task), schematic (a foreign key which stored the schematic for which the simulation was run, if a schematic was used), owner (a foreign key which stored the user who ran the simulation) and netlist and result fields (which stored the corresponding netlist and result for the simulation)

Figure 3.18: Simulation history

Whenever a simulation result is returned, an api call is made to

`api/simulation/history/<save_id>/<version>/<branch>/<simType>`

which returns all the simulations for that circuit and that type of simulation which is then rendered on the frontend along with their netlists.

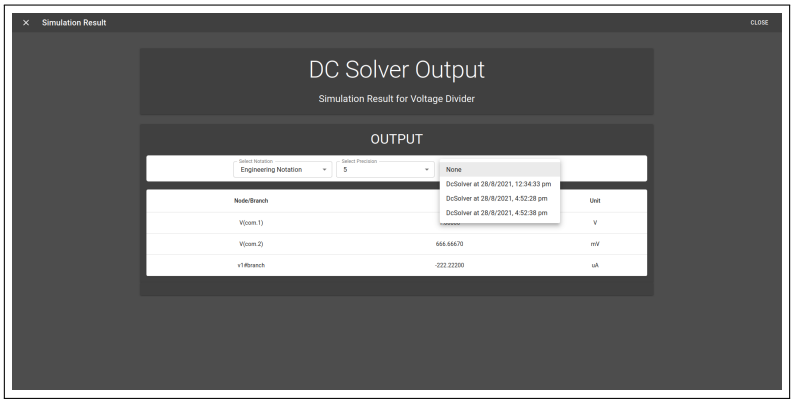

Figure 3.19: Simulation history

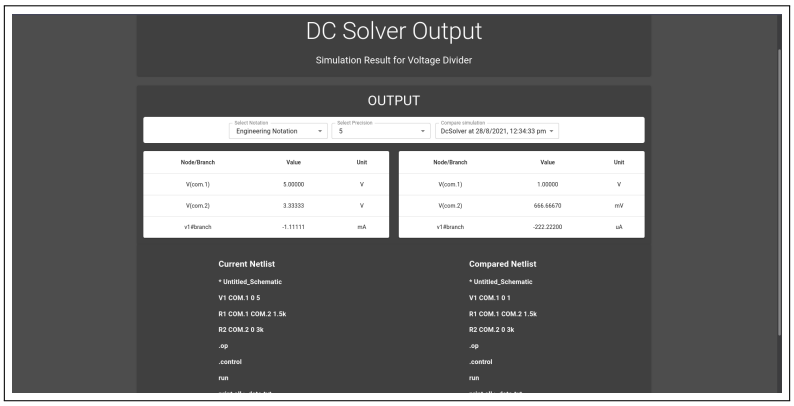

Figure 3.20: Comparing tabular simulations

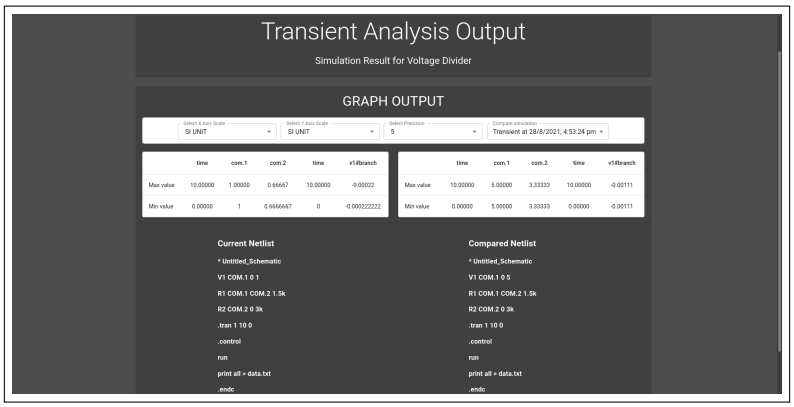

Figure 3.21: Comparing graphical simulations

3.3 Improving Graphs in Simulation Screen

Code was added in the front-end so that a button is added for downloading the graph plot points in the form of a CSV(Comma Separated Values). Along with this auto scaling was added in the graph so that user will not have to manually scale the graph according to the values of the graph.

Figure 3.22: Download graph output

3.4 Improving User Experience

The user model was changed so that each email was unique in the db which restricted users to register multiple accounts with the same email id.

Chapter 4

Conclusion

In conclusion, I have successfully implemented LTI, simulation history and comparison, merging LTI with the system and improving user experience.
LTI will aid in making the platform more usable for teachers. This will encourage more users to contribute and use the platform likewise.
Adding simulation history and comparison will help the user to come to conclusions about their circuits more easily and can use the feature to extract important pieces of information that can be useful to them. This project has helped me a lot of new technologies, good practices and designing a robust system for many users to use.

Chapter 5

Future Work

LTI can be improved a lot by using our own oAuth implementation instead of using the pyLTI library which is now archived, meaning no security patches will be added. Along with that pyLTI only support oAuth 1.0 which restricts our platform from further implementing LTI 1.3 (which requires oAuth 2.0).
In case of simulation history and comparison, a separate page can be created which displays all the simulations a user has ran, filtered according to the circuit on which the simulation was ran.

Bibliography

[1] T. McNulty, "How to switch to a custom django user model mid-project." https://www.caktusgroup.com/blog/2019/04/26/how-switch-custom-django-user-model-mid-project/, 2019.

[2] http://ngspice.sourceforge.net/docs/ngspice-manual.pdf (Last accessed 17th August 2021).

[3] https://www.django-rest-framework.org/ (Last accessed 17th August 2021).

[4] https://www.djangoproject.com/ (Last accessed 17th August 2021).

[5] http://www.imsglobal.org/activity/learning-tools-interoperability (Last accessed 17th August 2021.

[6] R. Jose, "django-lti-auth." https://github.com/rohitjose/django-lti-auth, 2018.

[7] https://www.nginx.com/ (Last accessed 17th August 2021).