GSoC‘19 Project Proposal

CircuitVerse

Interactive Book

Basic Information

Name and Contact Information

- **Name:** Aman Singla
- **Location:** Punjab, India
- **Timezone:** UTC +530 India Time Zone
- **Email:** amansingla97@gmail.com
- **Github**: https://github.com/amansingla97
- **Phone Number:** +91 9915 724 720
- **Resume**: Resume
- **Recommendation letter:** Creena Mehta (Senior Engineering Manager Microsoft)

University and Current Enrolment

- **University:** Indian Institute of Technology, Roorkee.
- **Field of Study:** Electronics and Communication Engineering (Batch of 2019)

Commitment

1. Are you planning any vacations during the GSoC period?
   No

2. How many classes are you taking during the GSoC period?
   None

3. Do you have any other employment during the GSoC period?
   No

4. How many hours per week do you expect to work on the project?
   I can easily contribute 50 hours per week

Statement of Motivation

I have been an enthusiastic developer and have always loved the concept of open source and wanted to contribute to the same. I have been trying to contribute in the small ways I could, by having all my projects on GitHub so that others can benefit from the same and trying to fix minor bugs that I can. GSoC provides a good platform to dive right into the middle of open source development with the opportunity to work in big open source projects with the core developers. This makes me excited and eager to participate in it.

I chose this particular project because it is very well aligned with my interests and also correlated to what I have worked on in the past. Other than that, this project provides a good opportunity to apply my learnings on a practical scale. This is my field of interest and therefore the natural inclination to this project.

Coding Skills

These are the languages that I can code in (in order of proficiency):

- Ruby on Rails
- Javascript
- PHP
- Python
- JAVA/C++ with sound knowledge of Object Oriented Programming

Related Experience

I am an Electronics and Communication Engineering student at IIT Roorkee. I have done 2 courses in Digital Logic Design (Basic and Advanced). Along with that, I have developed a deep interest in designing and learning about user behavior. I have done a design internship during my 2nd year. Further, I have experience of working closely with a college group called SDSLabs which involves building applications for the campus as well as helping other students by conducting lectures and engaging workshops.

Relevant Projects

- **Cerebro (Developed in React + Redux)**

  - Cerebro is an online sports programming platform built using React with custom Online Judge for hosting Data Science competitions developed and maintained by SDSLabs. It is hosted on the intranet in the IIT Roorkee campus.

  - **As a part of SDSLabs, I developed the frontend of the application in React and Redux.**

  - It involved building a large no. of modular components with a lot of user interactions and animations, managing authentication and interacting with an API, and writing unit tests and end to end tests.

- **Similar Case Detection** (**ML Research Paper and Implementation**)

  - Designed a simple but elegant architecture to solve the cold start problem of getting the labeled training data, which combines the un-supervised model results and human feedback to prepare accurate labeled data for supervised training model.

  - To increase the efficiency and reduce the turn around time of the customer service agents, we have implemented a machine learning based system to identify similar cases.

Contributions so far

- ## PRs in CircuitVerse

  #319 Pgsearch(search on projects)

  Previously we were using google custom search which doesn't do a very good job. This search would search over project names and descriptions.

  #297 Sidekiq

  I have dockerized sidekiq.

  Also added the Sinatra gem which is needed for the Sidekiq web UI (the dashboard is shown below)

  #290 Fix_Scroll

  Added a scroll to combinational analysis (Tools->combinational analysis) when the number of inputs is high as earlier, the truth table was no longer viewable

  #265 Empty_Project

  Saving an empty project produced an error. When the circuit is empty, then data_url is data:,, it otherwise is <actual_data>

  #232 Implement Subscriptions

  Added the functionality of subscriptions to comments. This was done through gem Commontator. More to add I even changed the inbuilt functioning of the gem.

  #247 Implement Voting and Mentions

  Have implemented voting (likes and dislikes) on comments along with mentions in the comment.

  #239 Drop Mysql support

  Supporting two databases causes unnecessary hurdles in development. We decided to drop Mysql support and just use PostgreSQL in all environments

  #187 Adding watermark to iframes

  A watermark "Made with CircuitVerse" is displayed on iframes on clicking to which directs to a new page in the simulator.

  #298 Watermark_clickable_fix

  The clickable area was hidden under the simulation area. Setting the z-index solved the problem

  #303 Watermark and tooltip overlap issue

  Changed the position of tooltip from the bottom right to bottom left.

  #178 Fixing Dropdown Position

  Added a bootstrap class that fixes the positioning

  #164 Css correction of alert in case of invalid login

  Css issues related to positioning were corrected.

  #351 Issue raised to Implement Elastic Search

  Provided a step by step guide + Resources to implement elastic search.

#356 Enable edit and delete for all the comments.

A quick fix in commontator.rb enabling the user to delete and edit their previous comments.

- ## PRs in CircuitVerseDocs

#113 Added documentation for saving projects

Added documentation for saving the project both offline and online.
It also includes the importance and usage of tags in online saving.

#114 Added documentation for forking projects

Added documentation for forking the project.
Tried to explain what is a fork and when do we use it w.r.t CircuitVerse.

#115 Added documentation Collaborations

Explained what do we mean by collaborations in CircuitVerse
Also, Explained how we can implement collaborations.

#116 Typo+Grammar+Spaces in the entire repo

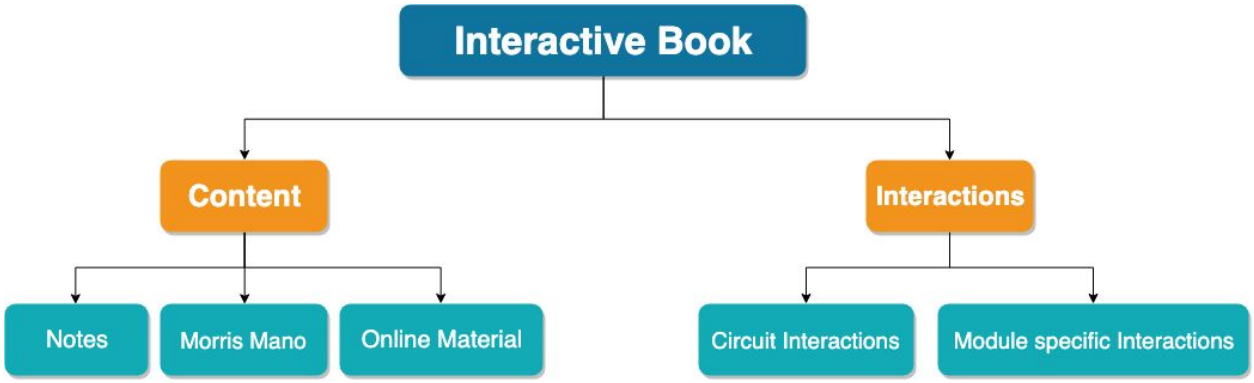
Went through the entire repo and found a ton of mistakes.
Was able to detect 35+ mistakes.

Project Proposal

Synopsis

The aim of this project is to create an online interactive guide for digital logic design. The primary goal is to develop an open source book with quality content which teaches digital logic design. It will enable students to learn digital design by interacting with circuits, truth table, and other interactive elements as they proceed through the book.The professors and students all over the world can read and contribute to the same.

Overview



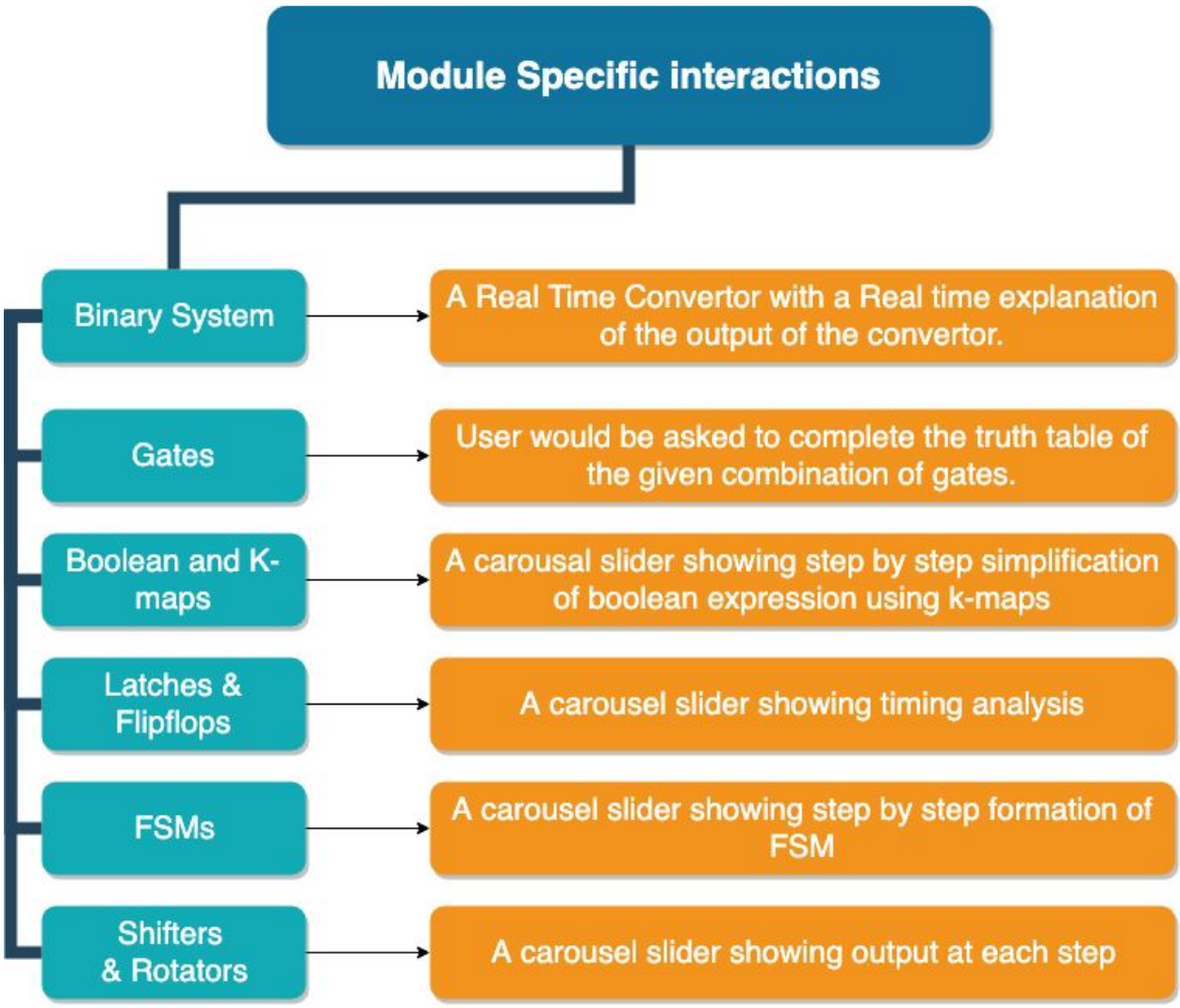
The Interactive book would have two components in it.

1. Content
2. Interactions

Content- It will include quality content which would be gathered from various books (primarily from '**Digital Design**' by **Morris Mano** as its copyright has expired and is in public domain) and online material. Basic notes have already been prepared to reference the aforementioned book and are attached in the table below.

Interactions- There would be two types of Interactions.

1. **Circuit Interactions-** This would include designing optimal circuits which would help to clearly understand the logic. The student would be able to clearly detect the variation of output with the change in input via these circuits. With each Interaction module, there would be a set of instructions that would guide the user to see the desired changes.
2. **Module Specific Interactions-** The entire content would be divided into different modules as shown below.



Layout

Below is the UI designed. According to me, a green and black color combination would suit best to represent CircuitVerse Interactive Book.

Binary Numbers

Binary is a scheme of numbers that only has two possible values for each digit: 0 and 1. The term also describes any encoding/ decoding system in which there are only two possible states. In digital data storage, memory, communications, or processing the 0 and 1 values are sometimes called "low" and "high" or "On" and "Off". The digital world is represented in binary, but hexadecimal, which is compatible with binary and more easily understood by people, is commonly used. Like previously mentioned, binary uses only the numbers 0 and 1. However, in some cases L/H is used as a counterpart to 0/1 notation. In order to indicate which base is used, d(decimal), b(binary), and h (hexadecimal) are added to the end of the number. Such information may further have mathematical relationships. In order for a computer (digital circuit) to perform computing (mathematical operations) on such information, it must be first able to perform arithmetic operations. The arithmetic operations are the basic mathematical operations. Only by performing arithmetic operations, other algebraic operations can be performed on numerical data. The digital circuits only understands and manipulate binary numbers. So, the arithmetic operations can be performed on data only in binary form. The digital circuits implement various arithmetic operations (binary arithmetic) with the help of logic gates. The electronic circuit of a binary adder (built by logic gates) with suitable shift registers can perform all arithmetic operations. Before understanding how digital circuits capable of performing mathematical operations can be built using logic gates, it is important to understand how various arithmetic operations are executed in binary form. So, let us learn the basics of Boolean algebra.

Binary Addition

The addition of two binary numbers is done the same way as is done for the decimal numbers. The addition is carried out from the least significant bits and it proceeds to higher significant bits, adding the carry resulting from the previous addition each time.

Binary Addition

Like the subtraction of decimal numbers is done, similarly is the subtraction of binary numbers done. The subtraction is carried out from the least significant bits and proceeds to the higher significant bits. For example consider subtracting (1101)2 by (1001)2 as follow -

CircuitVerse

Search Topic

Interactive Book

45% COMPLETED Reset

Compliment
Signed and unsigned
Parity bit
Binary codes

Boolean Algebra
- Addition and multiplication
- Properties
- Demorgan's theorem
- Commutative, associative and distributed law

Gates
- NOT,OR,AND,NOR,NAND XOR
- Combination of gates
- Exercise-NAND into

K-Maps
- The map method
- Two and three-variable Maps
- Four-variable maps
- Sums simplification

Mark Course as Completed

- Is there any single point of failure in our system? What are we doing to mitigate it?
- Do we have enough replicas of the data so that if we lose a few servers we can still serve our users?
- Similarly, do we have enough copies of different services running such that a few failures will not cause total system shutdown?
- How are we monitoring the performance of our service? Do we get alerts whenever critical components fail or their performance degrades?

Summary

In short, preparation and being organized during the interview are the keys to be successful in system design interviews. The above-mentioned steps should guide you to remain on track and cover all the different aspects while designing a system.

Let's apply the above guidelines to design a few systems that are asked in SDIs.

Completed

Next →
Designing a URL Short...

Send feedback 56 Recommendations

Workflow

The whole project objectives can be broadly classified into two phases through to completion:

Initial Work

0.1 Extract all the relevant information from Morris Mano and other sources

Following is the content material for different modules. I have made notes of must include items from various online sources. This phase would extract all the information from the book and combined with notes to get the most relevant content.

Module	Sub Modules	Notes	Morris Mano Page No.
Binary Numbers	• +,-,*,/ • Compliment • Signed and unsigned • Parity bit • Binary codes	Link	1-17
Boolean Algebra	• Addition and multiplication • Commutative, associative and distributive law • Properties • Demorgan's theorem	Link	36-55
Gates	• NOT,OR,AND,NOR,NAND XOR • Combination of gates • Exercise	Link	58-61
K-Maps	• The map method • Two and three-variable Maps • Four-variable maps • Sums simplification • Don't-care conditions	Link	72-100
Combinational Logic	• Adders • Subtractors • Code conversion • Multilevel NAND and NOR circuits	Link	114-138
MSI and PLD	• Binary adder and magnitude comparator • Multiplexers • Decoders and encoders	Link	154-179
Latches - Flipflop	• Introduction • Flip-flops • Triggering of flip-flops	Link	202-235

	• Excitation tables		
Counters - Registers	• Introduction • Registers • Shift registers • Ripple counter • Timing sequence	Link	257-288
FSMs	• Introduction • ASM chart • Timing considerations • Design with multiplexers	Link	307-329
Asynchronous Sequential Logic	• Introduction • Circuit with latches and design procedure • Reduction of state and flow tables	Link	341-373

0.2 Presenting all the information in HTML5 and Markdown

It includes designing layout of pages keeping in mind the user experience and providing them an interface which is easier to navigate and which also detects the progress as the student proceeds through the module. A brief and interactive summary will be included at the end of each module.

Primary Work

1.1 Make Interactive and relevant circuit diagrams

This includes designing specific interactive circuits required which would detect the change in output with the change in input. It will include designing relevant iframes and other interactive circuits.

1.2 Make different module specific Interactions

This includes developing module specific interactions as mentioned in the above flow chart. For eg some of them are:

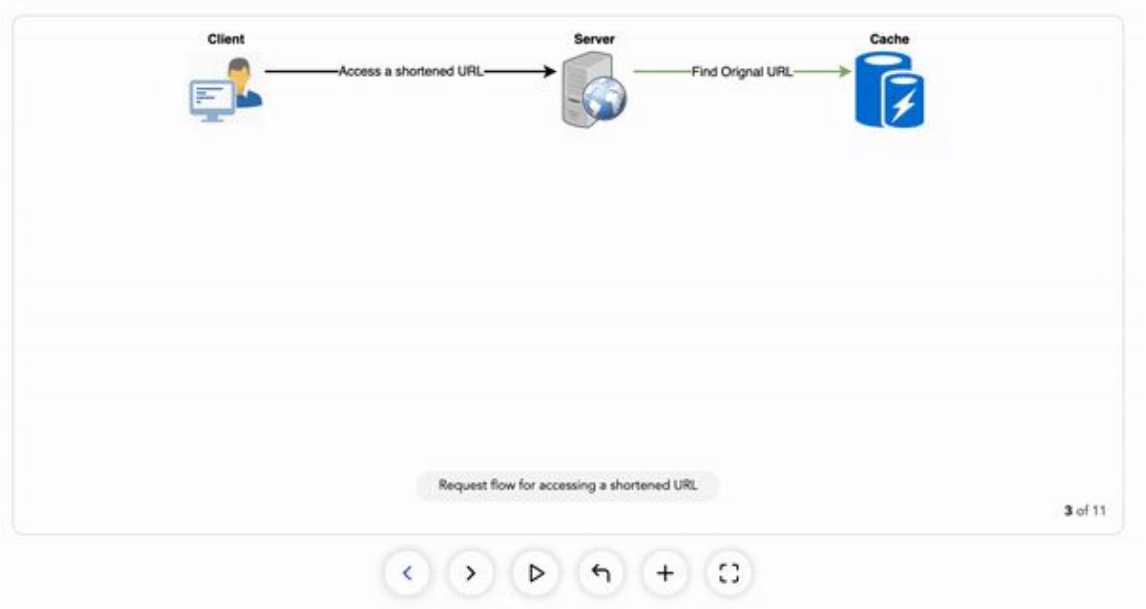
- ❏ In Binary numbers, an interactive module dynamically showing how a binary number is formed. An exercise would also be involved using this interaction to ask the user to form a binary representation of given numbers.

binary number:	1	1	1	0	0	1
power of 2:	2^5	2^4	2^3	2^2	2^1	2^0

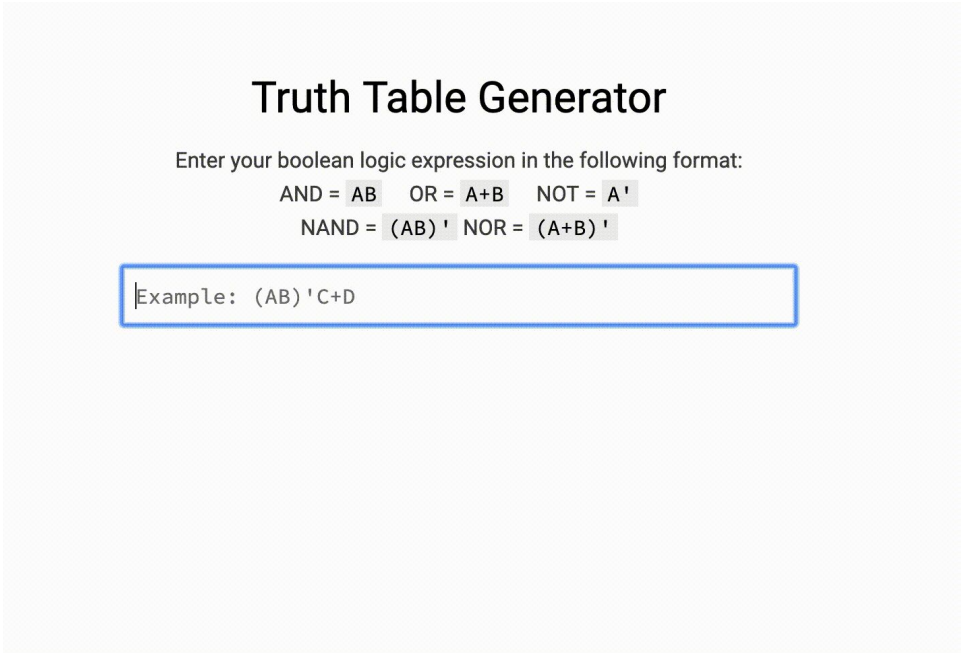
$$111001_2 = 1\cdot2^5+1\cdot2^4+1\cdot2^3+0\cdot2^2+0\cdot2^1+1\cdot2^0 = 57_{10}$$

- ❏ A carousel (shown below) would be used in a different module to show a step by step formation of KMaps, latches, and flip-flops, FSM's and shifters. Each step

would be properly labeled providing a clear image of the process.



- Truth Table generator could be used to visualize and relate boolean algebra with truth tables



Technical Details

- As the requirement of the project is to be serverless. I plan to use **Github Pages,** similar to CircuitverseDocs unless I find a better alternative.
- Tech Stack
  - HTML5
  - CSS
  - Javascript
  - Jquery
  - Bootstrap
- As all the interactions described above can be coded in javascript(frontend) Github pages would work perfectly.
- I experimented with Github pages and was successfully able to code a Truth-table generator.
  DEMO - https://amansingla97.github.io/Truth-Table/
- During the summers if something comes up that demands the usage of the server. I plan to code the required things in Circuitverse main repo and embed these in the form of iframes or in the form of plugins.

Contribution by new Users

- A detailed Documentation and code of conduct would be written to be able to help the new users to contribute to the book.
- Users would be able to contribute by following the given steps
  1. Fork the GitHub project
  2. Create your feature branch `(git checkout -b feature/feature_name)`
  3. Commit your changes `(git commit -am 'Add feature')`
  4. Push to the branch `(git push origin feature_name)`
  5. Create a new Pull Request

Project Plan

Preliminary Plan

May 4 - May 30 (Community Bonding Period)
- Understand existing codebase.
- Discuss with the mentor about the best way to go about the implementation.
- Go through the book and online content available.

Week Number	Start Date	End Date	Tasks to be completed
Week 1	27 May	2nd June	Compile notes up to module 5. This includes compiling from Morris Mano book and online material. Basically, I'll be typing content in a text editor.
Week 2	3rd June	9th June	Compile all notes. This includes compiling from Morris Mano book and online material. Basically, I'll be typing content in the text editor.
Week 3	10th June	16th June	Design the user interface of all Modules. Designing the wireframes, layouts of individual modules.
Week 4	17th June	23rd June	Completion of the basic book. Start presenting all the information in HTML5 and Markdown.
Week 5	24th June	30th June	Research more about the module specific interactions. Try to Include interactive exercise after each module
Week 6	1st July	7th July	Design all Circuit interactions using iframes of circuitverse Simulator.
Week 7	8th July	14th July	Code truth table generator, completion module, and binary interaction modules.Design example to illustrate the usage of these interactions.
Week 8	15th July	21 July	Code Carousels with instructions of use of all the modules.Complete the conversion of notes to markdown.
Week 9	22nd July	28th July	Code the remaining illustrations found during week 5. Try to include interactive summary after each module.
Week 10	29th July	5th Aug	Add features like Bookmarks, Highlights, etc. Try to Include advance digital logic design topics. Try adding tests and sample papers for teachers as well as students.
Week 11	6th Aug	12th Aug	Add documentation for the things added to help new contributors to contribute to the project. Tidy the code and increase test coverage.
Week 12	13th Aug	19th Aug	Add a manual and code of conduct for new contributors. Submit the code for final evaluation. Discuss future prospects with the mentors.
Submission			

Major Milestones

1. June 24 - 28, 2019
    - Compile material from all the resources
    - Get the layout and designs ready
    - Code the content in HTML5 and markdown
2. July 22 - 26, 2019
    - Design interactive summary at the end of each module
    - Design circuit Interactions using iframes etc
    - Design all module specific interactions
3. August 19 - 26, 2019
    - Features like bookmarks & highlights
    - Documentation and manual for new users to contribute

Additional Suggestions and Contributions

After project completion, which I am quite sure I'll be able to complete before the deadline, I would like to work on a feature that enables users to raise issues to track ideas, enhancements, tasks, or bugs for projects in the classroom. Issues can act as more than just a place to ask questions than to transfer knowledge.

Put another way, I would like to build "**GitHub issues**" for CirtcuitVerse where users can ask and discuss questions, comment on it, etc.

About Me

I am a 22-year-old, second-year student currently enrolled in Electronics and Communication Engineering (IV Year Course) at IIT Roorkee. I developed a passion for programming and web development in my freshman year. I have been contributing to open source regularly since about three months from now - looking over to repositories of the products I use/come across and trying to give my part of contribution back to the organization whose product has been an asset to me.

I am an active member of SDSLabs at IIT Roorkee, a bunch of passionate enthusiasts trying to foster software development culture in the campus. I have a previous experience to work as per my proposed timeline with a mentor on a summer project in SDSLabs. Also, most of our work is done according to a strict timeline with a lot of focus on test driven development and following best practices to structure code as per the design principles for the task.

I am also selected for the job of Software Developer in Microsoft, Redmond, USA and would be joining in the month end of september this year.

* * * * * * * * * *