**MATH CONVERTERS PROJECT**

Eetu Heikkinen

Nam Do

Oulu University of Applied Sciences

contents

1 INTRODUCTION 1

2 THE WORK ENVIROMENT 1

3 DEFENITION 1

3.1 Number Converter 1

3.2 Combination & Permutation 2

3.3 Truth Tables 2

3.4 Random value 2

4IMPLEMENTATION 2

4.1 Planning 2

4.2 Dec, Bin, Oct and Hex Converter 3

4.3 0–50 Decimal Table 3

4.4 Combination & Permutation 3

4.5 Truth Tables 4

4.6 Implementation conclusion 4

5 TESTING 4

5.1 Unit testing 4

5.2 Integration test 5

5.3 System testing 5

6 problems 5

6.1 Problems with Number converter and Number table5

6.2 Problems with Combination & Permutation 5

6.3 Problems with Truth table 5

6.4 Problems with Random value 5

7 POSSIBILITIES OF FURTHER DEVELOPMENT 6

8 CONCLUSION 6

# 1 Introduction

This document is going to be about what we have learned in two periods and how well we can build a web page and implement JavaScript. Project started on 3.12.2018 and teachers divided us into groups and pairs, so it would be more like realistic working environment. This documentary will go different phases what we have done, how did we do it, what problems we had and how did overcome those problems.

Firstly, we will go through working environment and how technical side of it and how its related to real world. After that there is going to be definitions where we will explain how everything is working and how we managed to do it. Then we will have implementation and what problems that caused to our project and how we fixed those problems. In testing we will describe how the testing was done and what we found with the testing, what bugs we found by doing testing and what limitations we had to set on some of the calculators.

We will talk about problems in a bigger scale too and tell what kind of problems we faced turning this project and tell what took most time and were the hardest to fix. Lastly, we will talk about the future of this project and how we could improve this in future and what plans we have for the future development.

**2 THE WORK ENVIRONMENT**

Working environment was regular class room with around 30 people working there most of the time and the noise level was through the roof sometimes and it was quite hard to concentrate at time. We also worked at home with this project by planning what we want to get finished the next day and we did also do some work turning weekends to be sure we can stay in our schedules with this project.

Working with laptops turned to be hard at times when you were doing some work on laptop and then checking it at home on desktop pc the layout looked completely different, so we had to come out with solution fix these issues. But overall working with laptops was easy since you could easily show the ideas or a code you did to the coworker before starting the process of implementing it into to the web page.

Both of us used the Visual Code Studio program for coding because we have found out that this has good tools build into it what comes to working with pushing and pulling to GitHub and it has good extensions you can use to make your work flow better. This program also helped us to make the styling and implementation of codes to the web page easier since it will automatically give you the correct “class” names when you start to type them into the CSS file.

**3 Definition**

This system is planned to help users with their mathematical problems, we will provide variety of different methods and calculators to solve user’s problems best we can. In total we have five different calculators to help users out and some of them can work in many ways.

**3.1 Number Converter**

Number converter was the first thing we decided to make, and we wanted it to work in many ways, so user can input Decimal, Binary, Octal and Hexadecimal and the calculator will calculate all the values no matter which one the users decides to input. We also made sure that there is no room for error, so we added limitations on the inputs, example: Binary will only accept one and zero or otherwise it will tell the user that the input is invalid, and it will not let users execute with invalid inputs.

**3.2 Combination & permutation**

Basically, this too helps with the calculation of three main probability problems. This includes combination, which means how many subsets can be created from the original set and permutation, which is to count how many possible ways there are to create an (un)ordered list.

**3.3 Truth Tables**

We made basic Truth Tables for users to check how they work and we added also definition on what the different symbols mean in there and that in mind they can look the symbol definition and understand how it works in these basic Truth Tables.

**3.4 Random Values**

This was just simple random value tester that will allow user to input a number which is the range for the numbers and then they can input the amount of numbers to be generated and after that they can press a button to see the statistics and this will show how many times the numbers occurred in the random generator.

**4 Implementation**

**4.1 Planning**

At the start of the project we discussed about the layout of the web page and how we wanted to present everything, based on those decisions we started to build our web layout. We started to work on the HTML + CSS files after we had basic idea about the layout and while other worked on the first JavaScript file other made the layout with styling.

In the layout we added separated tabs for each calculator except for the first two because those are close to each other, so it made sense to us to have them on the same page. Then we made some indicators where the users in the page so the tab the user is in is different color from the others and we made animated styling for underline when user mouseovers the tabs.

**4.2 Dec, Bin, Oct and Hex Converter**

Then when the first script file was ready to be implemented to page, we just put into own folder and made table for it and linked the file for it to show up on the web page, after that was done we just made sure everything worked correctly and then we started to style it to the style of the layout. Here again when other started to work on styling of the first script and other started to work on the next one.

There were several problems what came to styling tablets for us to be satisfied whit it and there started to be real problems after the second JavaScript was implemented to the first tab because when we tried to make it responsive they would just overlap with each other and we had to spent almost one day only to make it work as responsive web page.

**4.3 0**–**50 Decimal Table**

The second JavaScript was just simple for loop to print out the 0–50 Decimal to Hex, Oct and Bin. We used the first JavaScript file to get the information to printout the correct values for this table and after it was working, we made the show and hide button at this point it was done with the document write and Chrome would give us errors not to use this method because it can slow down the web page so we came up with different solution to make it work so we used table tag and then made the JavaScript file to generate the table cells with element ID. After that was done and everything was working once again we spilt the work so that other works on styling and other works on the next task. Styling for this was just simple borders and font size was set to same as the first converter to have that simple styling and it follows through the web page.

**4.4 Combination & Permutation**

Next up was the combinatorics and permutation and this took some time to do since we had to study how these work at all, so it took quite a while to understand this before we could work on the coding part. This point the styling for the first tab was done so other worked on this combinatorics and permutation and other started to work on the truth tables. Both were challenging to make and took us over two work days to even have somewhat working scripts, especially with the truth table since the instruction said that we cannot hard code it, so we started to make a script that would have user inputs and it would generate the table for the user for all the possible solutions.

The implementation of combination and permutation calculator was not very hard as all the formulas were provided by our Mathematical teacher. However, the only problem was that those formulas were written in human-readable way, which means it was not optimal for computer to process. Thus, we had to make some modifications and now the website can handle a wider range of input numbers.

**4.5 Truth Tables**

The truth tables were hard one to make with the input since you had to make it understand the inputs and what the different values mean in the truth table and for it work correctly it felt like it was too hard for us to make whit our current skill in JavaScripting. We got it to work with the inputs and printing the tables out but the problem was that the symbol wouldn’t mean anything to it so it just printed them into the table so that said the truth table answer was incorrect. This JavaScript took us over three days to make and after that, one of the teachers said that we can hard code some of the code and after that it took us one day to make code that automatically prints out four different tables with correct answers. This was done by two different loops and four if statements and constructions with arrays. We managed to make it so that the whole table with cells is printed into a one empty div, after we had it printing out correct tables and answers we could start to style it and make it responsive with the web page.

Styling these truth tables had some problems because they were going in column direction and after we got them to go in a row, we could start to style them. We made minimalistic looking styling to these tables which was simple borders and font size same as on the first page. Then came the time to make it responsive and this went good over all but there was minor problems whit it and we fixed them easily by testing different solutions for it.

**4.6 Implementation conclusion**

Over all the hardest part in these has been the truth table and styling everything and after the styling has been done came the part to make everything responsive since that is important now days because people use more and more mobile devices these days. So, we really wanted to make everything work with mobile devices and we managed to do that, over all we still had plenty of time to fix minor things after all the bigger parts were done.

**5 Testing**

**5.1 Unit testing**

We carried out this test every time a new file is created or there are any changes in the source code. To test if a new feature or not, first we tested its functionality without any interactions with other files. We created a file and named it something like “temp.js”, wrote only the main code in it and ran the file in Windows Command Prompt to see if it works as expected. The results are mostly within expectation as our algorithms are simple. Any changes leading to successful execution were committed immediately.

**5.2 Integration test**

After testing a new feature individually, we continued to test if it works in companion with other files. The implementation of this test first involved backing up the previous working state of the file, in case it no longer worked after we added changes. After that, we opened the website(s) and saw if the script works well with the HTML file. For most of the scenarios where it did not work, the reason was a small typo mistake, or the script file was not linked with the HTML file. However, at this point, we usually must take care of input exceptions, which means that we needed to consider what users may input that can break our website. Therefore, we added a validator and carried the unit test one more time.

**5.3 System testing**

We loaded our project to a server and carried the test with different devices to test the basic functions as well as responsive design. The only problem we encountered was the mismatches of file names, which was fixed quite easily.

**6 PROBLEMS**

From the aforementioned testing methods, we detected several bugs and problems and they will be presented hereafter.

## 6.1 Problems with Number converter and Number table

There is no existing validator for binary, octal and hexadecimal input so we had to build them for ourselves and we also had to come up with a way to inform users about the errors they may cause if they input an invalid number. This took us a few hours to not only figure out an optimal way to do it but also rearrange the whole web layout for those error messages to fit in. Currently, the converter can only work with natural numbers.

## 6.2 Problems with Combination & Permutation

The older version of it can only work with small numbers (less than 300) because I used the traditional way to deal with input, which is using while loop, and the same formula as was taught in the class. In that way, the processor had to calculate too many things, so I had to change the algorithm a little bit and now it can deal with much bigger numbers. I have not tested the limit yet, but it can at least work with numbers over 2500, if the difference between the length of the original set and that of each subset is less than 170, for example 2243 C 2234 (this can be calculated by our website). There are also some styling-related problems, but they can be fixed easily

## 6.3 Problems with Truth table

The only problem we encountered was when we wanted to add more style to the tables. It was more difficult since we generated them with a JavaScript file, so it took a little more time to add ID and class name to the elements.

## 6.4 Problems with Random values

The first problem we had to deal with was to validate the input of users. They are not supposed to input either a number with decimal point or a negative number as the length of the number set they want to generate. Secondly, because the test tool we built relied on the use of loops, it also cannot deal with a set of number whose length is bigger than 1500 items (it can but it may affect the performance of your devices).

**7 POSSIBILITIES OF FURTHER DEVELOPMENT**

Possible features for further development are listed below:

* Convert also negative numbers
* An option for users to choose between the ordinary layout or a more professional and mathematical layout.
* Include more number bases in the converter
* Bring all the number generator and other loop-involving to server-side process so that will not be a burden on users’ devices and allow them to work with bigger numbers
* More tools to calculate other math operations such as logarithm and trigonometry.
* Equation solving

**8 conclusion**

Over all we accomplished our goals good time before the project ending time, mainly because we planned ahead what we are going to do, and we set daily goals to be achieved which gave us even more motivation to work towards making everything work. There were some setbacks few times, but we managed to catch up with everything and maintain the speed we were having. Best result in this project was the how we worked together, and we got every task done including the optional one and when there still might be minor bugs somewhere in the web page or in the JavaScript’s we are happy with the results that we got.