Project 1: Exam Grade Calculator

Table Of Contents:

[**Design Section:**](#_3iq1jbxbgo23) **3**

[Success Criterias](#_xkyo7ss47x2q) 4

[Functional Criteria](#_9ko3lfd3abql) 4

[Cosmetic Criteria](#_awtteujxj2ft) 4

[Test Plan](#_pky8dnftkodq) 5

[**Development Section**](#_wxkxrzo7mrgt) **6**

[HTML Designing](#_94hfepbwf7hj) 7

[Flowchart](#_uf1gndcw6vxe) 8

[Pseudocode](#_uuh7bg713461) 9

[Submit Function](#_lgb5pxkycyds) 9

[Grade Generator Function](#_gmu4rcl1hf1y) 10

[JavaScript Design](#_7n8zi2ebucqi) 11

[Page Design](#_ev1ivv1ex1p) 13

[**CODE**](#_n6u3tre9chcy) **15**

[HTML Code](#_h1bc3qp9e245) 16

[CSS](#_cvh3qnpjo1jl) 18

[JavaScript](#_gkdoilhfqkcu) 23

[**Testing**](#_473g1huofudi) **29**

[Completed Test Plan](#_wcpgn02wkn40) 30

[**Evaluation**](#_icpazh24y72z) **33**



# Design Section:

## Success Criterias

### Functional Criteria

My program should:

* Give correct and accurate results, e.g. the correct grade for the marks
* Be able to validate all inputs from the user
* Clearly show the students grade
* Check the name and subject fields are not blank
* Check that the exam number field is exactly 4 characters long and a number
* Check that the exam mark field is between 0 and 100
* Display an apt error message and turn the label red if necessary

### Cosmetic Criteria

My program should:

* Have a clean clean look e.g. the page should not have mismatched colours
* The page should have a dark tone
* The page should have depth
* The page should have a modern and up to date look

## Test Plan

| Purpose | Input | Expected Output |
| --- | --- | --- |
| Presence check, this checks if the entry fields are empty | Name: “”  Subject:””  Examination Number:””  Mark:”” | Name: “Please enter name”  Subject: “Please enter subject”  Examination Number: “Please enter a number”  Mark: “Please enter a number” |
| Type check, this checks If the inputs are the correct type. | Examination Number: “One”  Mark: “Nine” | Examination Number: “Please enter a number”  Mark: “Please enter a number” |
| Length check, this checks if the input is four digits long | Examination Number: “124” | Examination Number: “Please enter a four digit number” |
| Range check, this checks the input is within 0 and 100 | Mark: “150” | Mark: “Please enter a number (0-100)” |
| Checks it gives the correct output | Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”10” | Grade: U |
| Checks it gives the correct output | Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”42” | Grade: E |
| Checks it gives the correct output | Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”56” | Grade: D |
| Checks it gives the correct output | Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”63” | Grade: C |
| Checks it gives the correct output | Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”77” | Grade: B |
| Checks it gives the correct output | Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”87” | Grade: A |

## HTML Designing

| Code | Page | Comments |
| --- | --- | --- |
|  |  | Created the title and named the tab “Exam Grade Calculator” |
|  |  | I added the labels, buttons and entry fields.  I had an issue where everything was on one line, but I had just forgotten to add the <br> tags |
|  |  | Finally I added the grade output and created the text fields for the errors to appear. |

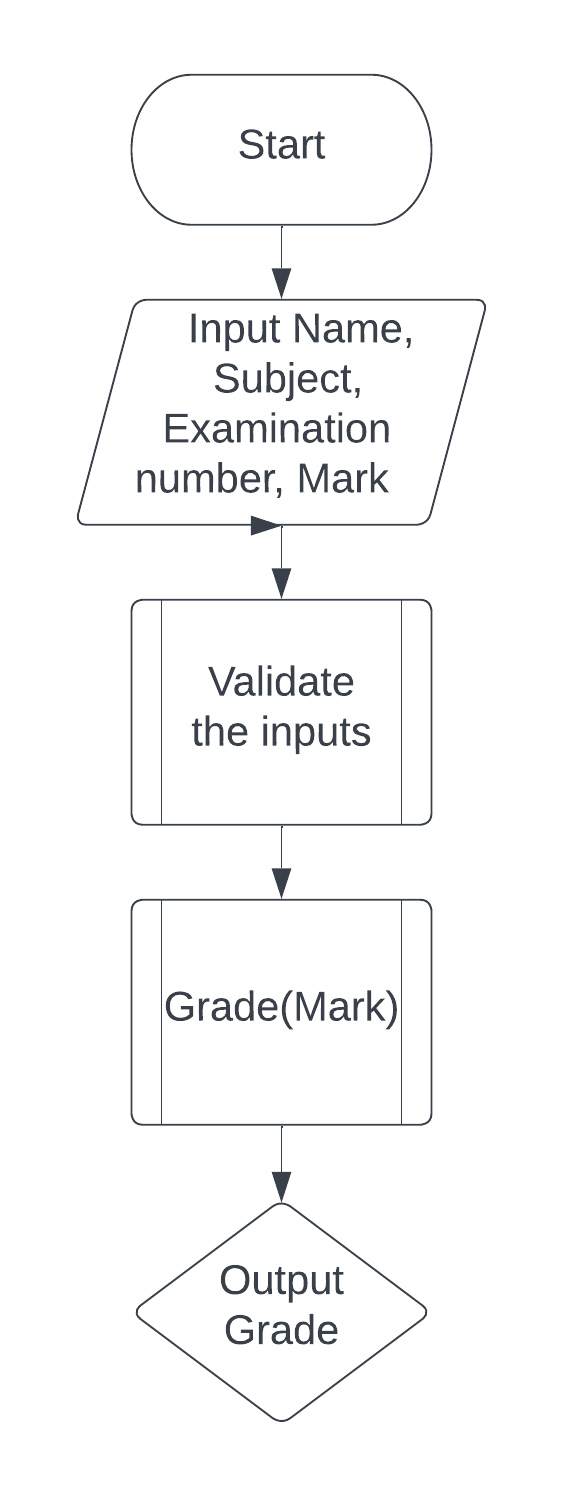
## 

## 

## 

## 

## Flowchart

I created this flowchart to give me a basic look at what my program must do and the order it must do it in.

## Pseudocode

### Submit Function

Function submit():

nameValid = False

subjectValid = False

subjectName = the value of studenNameField

studentSubject = the value of subjectField

if (studentName.length === 0){

document.getElementById("nameError").innerHTML = "Please Enter Name"

document.getElementById("nameLabel").style.color = "#e02f1e"

} else{

document.getElementById("nameLabel").style.color = "#d7d8db"

document.getElementById("nameError").innerHTML = ""

nameValid = true

}

if (studentSubject.length === 0){

document.getElementById("subjectError").innerHTML = "Please Enter Subject"

document.getElementById("subjectLabel").style.color = "#e02f1e"

} else{

document.getElementById("subjectLabel").style.color = "#d7d8db"

document.getElementById("subjectError").innerHTML = ""

subjectValid = true

}

### Grade Generator Function

function gradeGenerator(mark) {

//--This Function Generates The Grade For the Given Mark--//

if (mark === 0 || mark <40) {

return "U"

}

else if (mark === 40 || mark <50){

return "E"

}

else if (mark === 50 || mark <60){

return "D"

}

else if (mark === 60 || mark <70){

return "C"

}

else if (mark === 70 || mark <80){

return "B"

}

else if (mark === 90 || mark <=100){

return "A"

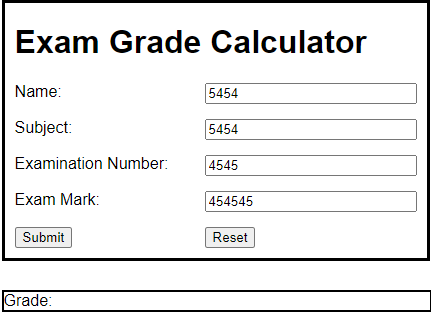
}

}

## JavaScript Design

| **Function** | **Code** | **Comments** |
| --- | --- | --- |
| Reset |  | I was almost able to copy my pseudocode directly into the code.  This code was relatively straight forward and i didn’t encounter any major errors |
| Grade Generator |  | I had a logic error that would only give an output of “U” no matter what the input was. The problem was I was only looking at values >0 rather than > 0 and < 40 |
| Submit |  | I made an error of changing the colour of the text in the input field rather than the labels.  At this point I hadn’t finished the code but the validation that was finished it worked perfectly |
| Submit |  | At this point I had finished creating the validation part which was fully functional.  The only error I had encountered was not putting in curly brackets |
| Submit |  | At this point all the code was finished and functioning and had been tested with a range of values |
| Enter Function |  | This was an addition I thought would be helpful through testing as I kept pressing enter to enter my values. |

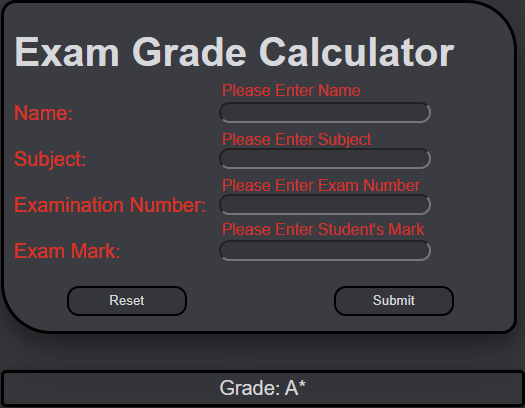
## Page Design



First Design:

I like this design because it encapsulates all the information in a single place. By having the grade output in a different box, it keeps it separate and stops any misunderstandings with how to use the program.

The only problems I have is it looks very outdated and does not have the dark colour scheme

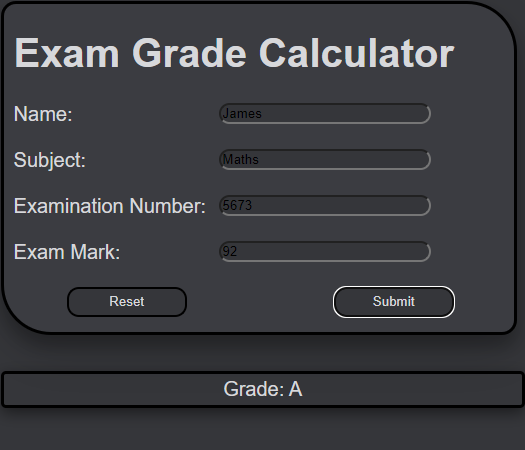


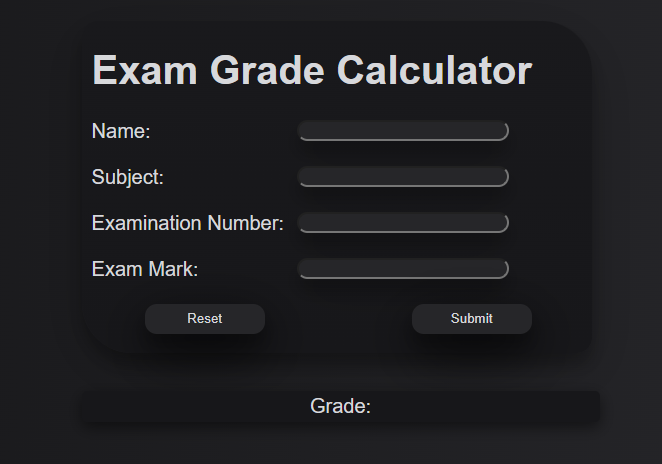
Second Design:

I like this design because it uses colours and helvetica which makes it feel more up to date.

The edges of the border, input fields and buttons are all curved which breaks up the outdated geometric feel of the original design.

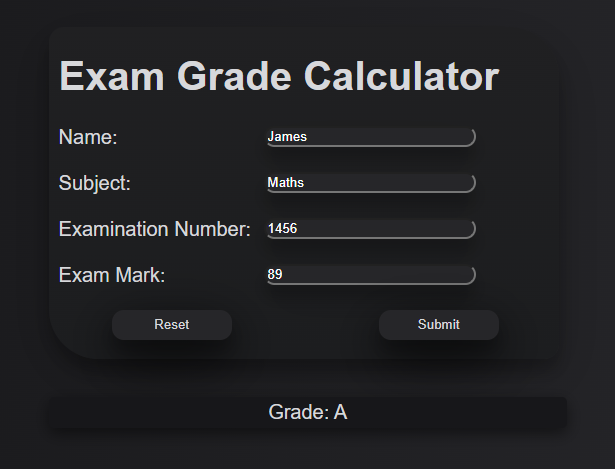
I also added the red colour to the error text and the labels giving an easy notice to the user they have done something wrong.





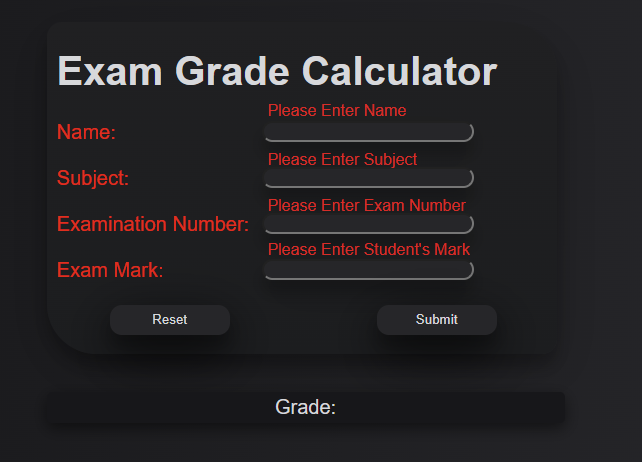
Third Design:

In this iteration I decided to remove the borders of the main input box, grade output box and buttons. By doing this it creates a sense of depth, as well as this I also added a box shadow to the boxes which further emphasises the depth.

Final Design:

In the final design I took the borderless design from the previous one, however I felt that the background and the Input box colour was too dark and it was merging so I lightened both and gave them a slight gradient further creating depth

I also changed the text colour of the input fields to match the rest of the text.

I had an error where the size of the error text and position changed depending on how zoomed the screen was. The problem was that the error text was in a completely different division to the rest of the elements.

## 

# CODE

## 

## 

## HTML Code

<html>

<div id ="main">

<div id="inputField">

<head>

<title>Exam Grade Calculator</title>

<h1>Exam Grade Calculator</h1>

</head>

<label id="nameLabel">Name:</label>

<input type="text" id = "studentNameField" value="" size="25"/><br><br>

<label id="subjectLabel">Subject:</label>

<input type="text" id = "subjectField" list="Subjects" size="25"/><br><br>

<datalist id="Subjects">

<option value="Mathematics">

<option value="English">

<option value="Physics">

<option value="Biology">

<option value="Chemisty">

</datalist>

<label id="examinationNumberLabel">Examination Number:</label>

<input type="text" id = "examinationNumberField" value="" size="25"/><br> <br>

<label id="examMarkLabel">Exam Mark:</label>

<input type="text" id = "examMarkField" value="" size="25"/><br> <br>

<button type="button" id = "reset" onClick="reset();">Reset </button>

<button type="button" id = "submit" onClick="submit();">Submit </button>

</div>

<div id="output">

<label size = "25" id="gradeOutput">Grade: </label>

</div>

<div id = "errorlabels">

<label id="nameError"></label><br> <br> <br>

<label id="subjectError"></label><br> <br> <br>

<label id="numberError"></label><br> <br> <br>

<label id="markError"></label><br> <br> <br>

</div>

</div>

</html>

## CSS

<style>

label{

transition: .1s;

}

/\*CSS for the entire page\*/

html{

font-family: helvetica;

background: linear-gradient(87deg,#121214, #2c2c30);

}

#main{

position: absolute;

left: 38vw;

top;

right;

float;

position: absolute;

margin: auto;

}

h1{

position: relative;

left: 21px;

}

/\*CSS for the error text\*/

#nameError {

transition: 1.s;

color: #d62e29;

position: relative;

top: -250px;

left: 220px;

resize: none;

}

#subjectError {

transition: .1s;

color: #d62e29;

position: relative;

top: -260px;

left: 220px;

}

#numberError {

color: #d62e29;

transition: .1s;

position: relative;

top: -267px;

left: 220px;

}

#markError {

color: #d62e29;

transition: .1s;

position: relative;

top: -278px;

left: 220px;

}

/\*CSS for the input box\*/

#inputField {

font-size: 20px;

background: linear-gradient(#202022, #1c1d1f);

padding-right: 60px;

padding-left:10px;

padding-bottom: 49px;

top: 50px;

border: solid 3px none;

border-radius: 15px 50px;

color: #d7d8db;

box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.9);

}

/\*General CSS for both buttons\*/

button {

color: #d7d8db;

width:120px;

height:30px;

border-radius: 12px;

border: none;

background-color: #262629;

box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.9);

}

/\*General CSS for both buttons when hovered over\*/

button:hover {

transition: .3s;

box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);

width:125px;

button:active {

transition: .1s;

background-color: #4f5056;

}

}

/\*General CSS for both buttons when pressed\*/

button:active {

transition: .1s;

background-color: #313136;

}

/\*CSS for the input fields\*/

input{

color: white;

position: absolute;

left: 215px;

border-radius: 12px;

background-color: #262629;

transition: .1s;

box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.10);

}

/\*CSS for when the input fields are in use\*/

input:focus {

background-color: #313136;

left: 220px;

}

/\*CSS for the output box\*/

#output{

background-color: #17171a;

font-size: 20px;

position: absolute;

top: 420px;

padding: 4px;

border: solid 3px none;

border-radius: 5px;

box-shadow: rgba(0, 0, 0, 0.5) 0px 5px 15px;

color: #d7d8db;

display: block;

width: 510px;

text-align: center;

}

/\*CSS for the reset button\*/

#reset{

position: absolute;

left: 63px;

}

/\*CSS for the submit button\*/

#submit {

position: absolute;

left: 330px;

}

</style>

## JavaScript

<script>

function submit() {

//--This Function Validates The Users Inputs And Outputs The Grade--//

//Boolean variables that check if the users inputs are valid

let nameValid = false;

let subjectValid = false;

let examNumbervalid = false;

let examMarkValid = false;

//Defines the values of the users inputs

let studentName = document.getElementById("studentNameField").value;

let studentSubject = document.getElementById("subjectField").value;

let examNumber = document.getElementById("examinationNumberField").value;

let examMark = document.getElementById("examMarkField").value;

//Presence Check

if (studentName.length === 0){

//If the entry box is empty it changes error text to "Please Enter Name" and changes the color of the label to red

document.getElementById("nameError").innerHTML = "Please Enter Name";

document.getElementById("nameLabel").style.color = "#e02f1e";

} else{

//If the entry box contains text change the color of the label to normal and change make the error text disappear

document.getElementById("nameLabel").style.color = "#d7d8db";

document.getElementById("nameError").innerHTML = "";

nameValid = true;

}

//Presence Check

if (studentSubject.length === 0){

//If the entry box is empty it changes error text to "Please Enter Name" and changes the color of the label to red

document.getElementById("subjectError").innerHTML = "Please Enter Subject";

document.getElementById("subjectLabel").style.color = "#e02f1e";

} else{

//If the entry box contains text change the color of the label to normal and change make the error text disappear

document.getElementById("subjectLabel").style.color = "#d7d8db";

document.getElementById("subjectError").innerHTML = "";

subjectValid = true;

}

//Presence Check

if (examNumber.length ===0){

document.getElementById("numberError").innerHTML = "Please Enter Exam Number";

document.getElementById("examinationNumberLabel").style.color = "#e02f1e";

} else{

//Type Check

if (isNaN(examNumber)){

//If the input is not a number change the error text to "Please Enter A Number" and change the color of the label to red

document.getElementById("numberError").innerHTML = "Please Enter A Number";

document.getElementById("examinationNumberLabel").style.color = "#e02f1e";

} else{

//Length Check

if (examNumber.length != 4) {

//If the number is not 4 digits long change the error text to "Please Enter A 4 Digit Number" and change the color of the label to red

document.getElementById("numberError").innerHTML = "Please Enter A 4 Digit Number";

document.getElementById("examinationNumberLabel").style.color = "#e02f1e";

} else{

//If the input is valid change the color of the label to normal and change make the error text disappear

document.getElementById("examinationNumberLabel").style.color = "#d7d8db";

document.getElementById("numberError").innerHTML = "";

examNumbervalid = true;

}

}

}

//Presence Check

if (examMark.length === 0){

document.getElementById("markError").innerHTML = "Please Enter Student's Mark";

document.getElementById("examMarkLabel").style.color = "#e02f1e";

}else{

//Type Check

if (!isNaN(examMark)){

//Range Check

if (examMark < 0 || examMark > 100) {

document.getElementById("examMarkLabel").style.color = "#e02f1e"

document.getElementById("markError").innerHTML = "Please Enter Number (0-100)";

} else{

document.getElementById("examMarkLabel").style.color = "#d7d8db"

document.getElementById("markError").innerHTML = "";

examMarkValid = true;

}

} else{

document.getElementById("markError").innerHTML = "Please Enter A Number";

document.getElementById("examMarkLabel").style.color = "#e02f1e"

}

}

//If everything is valid output the grade

if (nameValid === true && subjectValid == true && examNumbervalid == true && examMarkValid == true) {

document.getElementById("gradeOutput").innerHTML = "Grade: "+ gradeGenerator(examMark);

console.log(gradeGenerator(examMark))

}

}

function reset() {

//--This Function Removes All The Inputs From The Boxes And Changes The Colors Of all The text To Normal--//

document.getElementById("nameError").innerHTML = "";

document.getElementById("nameLabel").style.color = "#d7d8db";

document.getElementById("studentNameField").value = "";

document.getElementById("subjectError").innerHTML = "";

document.getElementById("subjectLabel").style.color = "#d7d8db";

document.getElementById("subjectField").value = "";

document.getElementById("numberError").innerHTML = "";

document.getElementById("examinationNumberLabel").style.color = "#d7d8db";

document.getElementById("examinationNumberField").value = "";

document.getElementById("markError").innerHTML = "";

document.getElementById("examMarkLabel").style.color = "#d7d8db";

document.getElementById("examMarkField").value = "";

document.getElementById("gradeOutput").innerHTML = "Grade: ";

}

function gradeGenerator(mark) {

//--This Function Generates The Grade For the Given Mark--//

let subject = document.getElementById("subjectField").value;

let UMark = 30;

if (subject === "Mathematics"){

UMark = 40;

} else if (subject === "English"){

UMark = 30;

}else if (subject === "Physics"){

UMark = 40;

}else if (subject === "Chemistry"){

UMark = 32;

}else if (subject === "Biology"){

UMark = 27;

}

if (mark === 0 || mark <UMark) {

return "U"

}

else if (mark === UMark || mark <UMark+10){

return "E"

}

else if (mark === UMark+10 || mark <UMark+20){

return "D"

}

else if (mark === UMark+20 || mark <UMark+30){

return "C"

}

else if (mark === UMark+30 || mark <UMark+40){

return "B"

}

else if (mark === UMark+40 || mark <=UMark+50){

return "A"

}

else if (mark === UMark+50 || mark <=UMark+60){

return "A\*"

}

else if (mark === UMark+60 || mark <=UMark+70){

return "A"

}

}

//When enter is pressed it runs the submit function

document.addEventListener("keypress", function(event) {

if (event.keyCode == 13) {

submit();

}

});

</script>

# Testing

## Completed Test Plan

| Input | Actual Output | |
| --- | --- | --- |
| Name: “”  Subject:””  Examination Number:””  Mark:”” |  | |
| Examination Number: “One”  Mark: “Nine” |  | |
| Examination Number: “124” |  | |
| Mark: “150” |  | |
| Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”10” |  | |
| Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”42” |  | |
| Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”56” |  | |
| Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”63” |  | |
| Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”77” |  | |
| Name: “James”  Subject:”Maths”  Examination Number:”1453”  Mark:”87” |  | |

# Evaluation

From my testing, I found that my program performed correctly, giving accurate results. I would say that I have satisfied all of the success criteria, the validation works correctly, there are precise error messages that define the problem and the result is clearly outputted.

Throughout my development I encountered a lot of errors, most notably the positioning of the elements on the screen, whenever you would zoom or change screen resolution, the error text would move positions and size. The problem was that all the elements on the screen were in different divisions so when the screen resolution would change, each element would change relative to their div position rather than the collective position. I fixed this by having nested divisions and changing the parent div’s position rather than each individual division.

In terms of improvements, I don’t feel like there is anything that I need to add to fit the criteria, however some extra things such as adding a way to save students data for many different subjects would add to the program's functionality. Another addition could be a space for an exam board to further verify the exam paper that they are inputting, reducing the chance of errors. As well as this, better validation could be added in terms of the name and subject inputs, the only validation that is in the program is a presence check. A type check could be added to make sure it’s a string and a length check could be added to further decrease errors by user misuse.

In all, I created a program that confirms to the success criteria and I learnt a lot

of new functions in JavaScript and I now know a lot more about CSS.