



**UNIVERSITI KUALA LUMPUR KAMPUS KOTA**  
**MALAYSIAN INSTITUTE OF INFORMATION TECHNOLOGY**

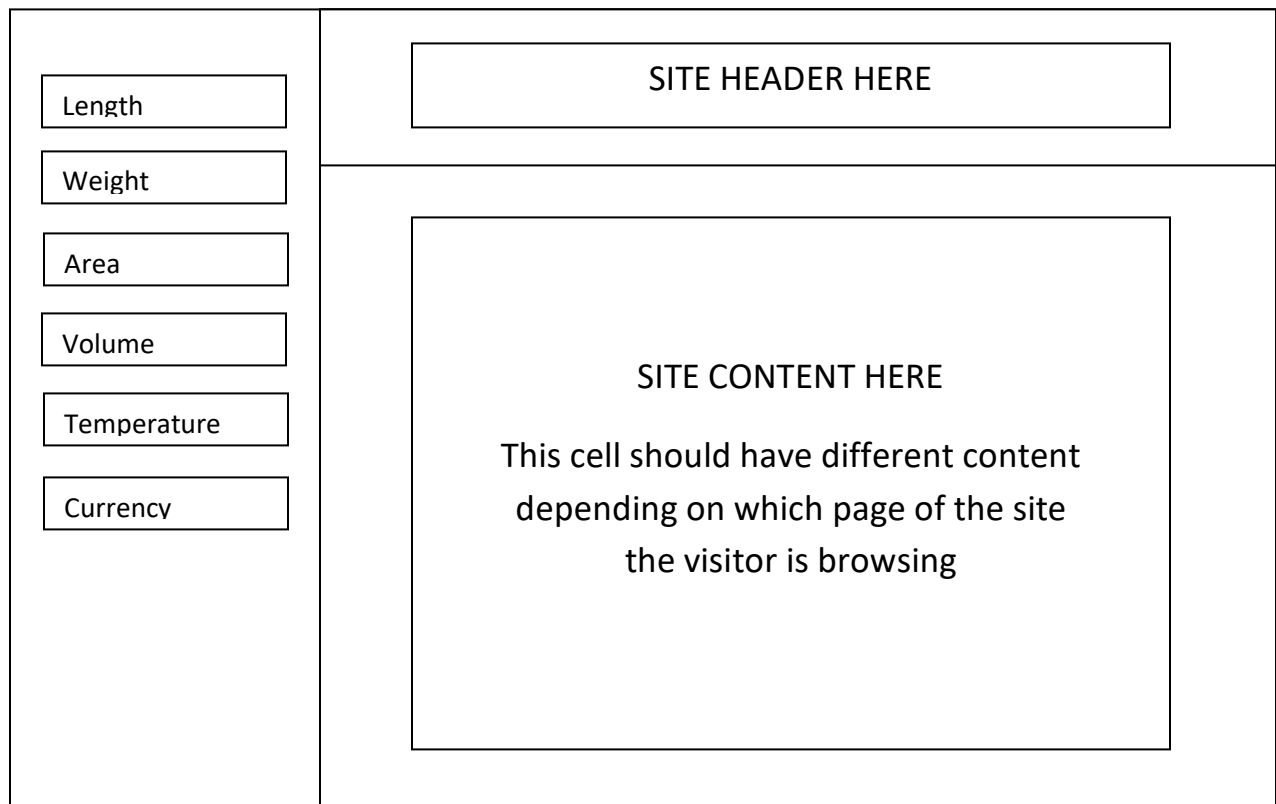
Name of Course	<b>WEB APPLICATION DEVELOPMENT</b>	Course Code	<b>ISB42403</b>
Lecturer	<b>MDM SURIANA ISMAIL</b>	Semester / Year	<b>June 2019</b>

Student Name	<b>Bazin Gwendal</b>
ID Number	<b>52200119004</b>
Programme	
Date	<b>22/09/2019</b>

Assessment	ASSIGNMENT	Duration	(1.5+WEEKs)	Weightage	100 marks (10%)
Course Outcome to achieve: CLO 1 - Create an ASP.NET Web Application project by using Visual Studio.NET CLO 2 - Add Functionality to an ASP.NET Web Form using suitable controls <del>CLO 3 - Integrate developed ASP.NET Web Application with ADO.NET in Visual Studio.NET</del> CLO 4 - Demonstrate the successful Web Application orally with a complete documentation.					

## INSTRUCTIONS

1. This is an individual assignment.
2. Create a website that converts various units of measurement from one form to another or calculate certain measurements. The website must incorporate an ASP.NET master page and several content pages. The figure below is an example of the website structure that you can use. You can design your own structure.
3. You must include at least 4 categories of measurements. For each category of measurement (see figure below), include 6 different measurements. The measurements must be useful and practical.  
  
(You can also create your own categories of any computations of similar example.)
4. Calculations for the measurements are done using C#. Measurements must incorporate web user controls. Suitable formatting of numbers must be used. Suitable input validation must be applied to the input.
5. You may include CSS to make the user interface professional looking and beautiful. You can use images where suitable. Suitable formatting of numbers must be used.



## List of Measurements

### Length

NAME	COEF
METERS	1.0
KILOMETERS	0.001
INCHES	0.0254
FEET	0.3048
LIGHT YEARS	9460730472580044.0
PARSECS	30856775812799588.0

### Time

NAME	COEF
SECONDS	1.0
MILISECONDS	0.001
MINUTES	60.0
HOURS	3600.0
DAYS	86400.0
WEEKS	604800.0
MONTHS	2629800.0
YEARS	31557600.0

## Area

NAME	COEF
KILOMETERS <sup>2</sup>	1.0
METERS <sup>2</sup>	1e-6
DECIMETERS <sup>2</sup>	1e-8
MILIMETERS <sup>2</sup>	1e-10
HECTARES	0.01
ARES	0.0001

## Weight

NAME	COEF
GRAM	1.0
TON	1000000
POUND	453.59237
CARAT	0.2
CENTNER	100000
STONE	6350.293

## Speed

NAME	COEF
KILOMETER / HEURE	1.0
METER / SECONDE	3.6
MILES / HEURE	1.60934
MILES / SECONDE	5793.64
MACH	1234.8
LIGHT SPEED	1079000000.0

## Currency

NAME	COEF
USD	1.0
EUR	1.10
RM	0.24
CFP	0.0093
BTC	10716.80
ETH	175.55

## Volume

NAME	COEF
LITERS	1.0
CUBIC METERS	1000
MILLILITERS	0.001
TEASPOONS	0.005
BRETON SHOOTERS	0.5
BUCKETS	18.92706

## Data Storage

NAME	COEF
BITS	1.0
BYTES	8.0
KILOBITS	1024.0
KILOBYTES	8192.0
MEGABITS	1048576.0
MEGABYTES	8388608.0
GIGABITS	1073741824.0
GIGABYTES	8589934592.0

## Formula and C# code

I use a formula that with coefficients allows me to switch from one measurement to another, thanks to this formula I do not need more than one line of code to convert.

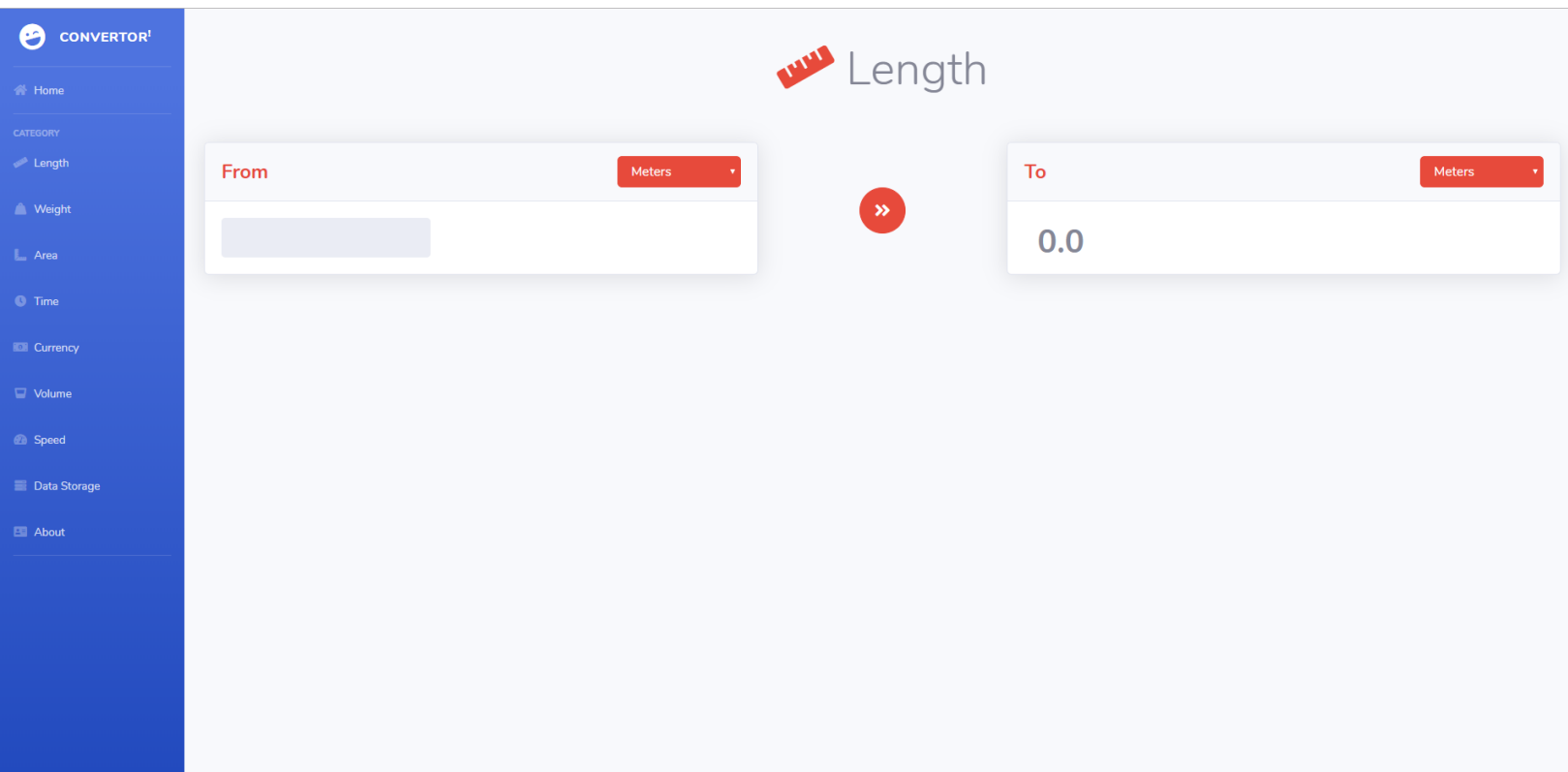
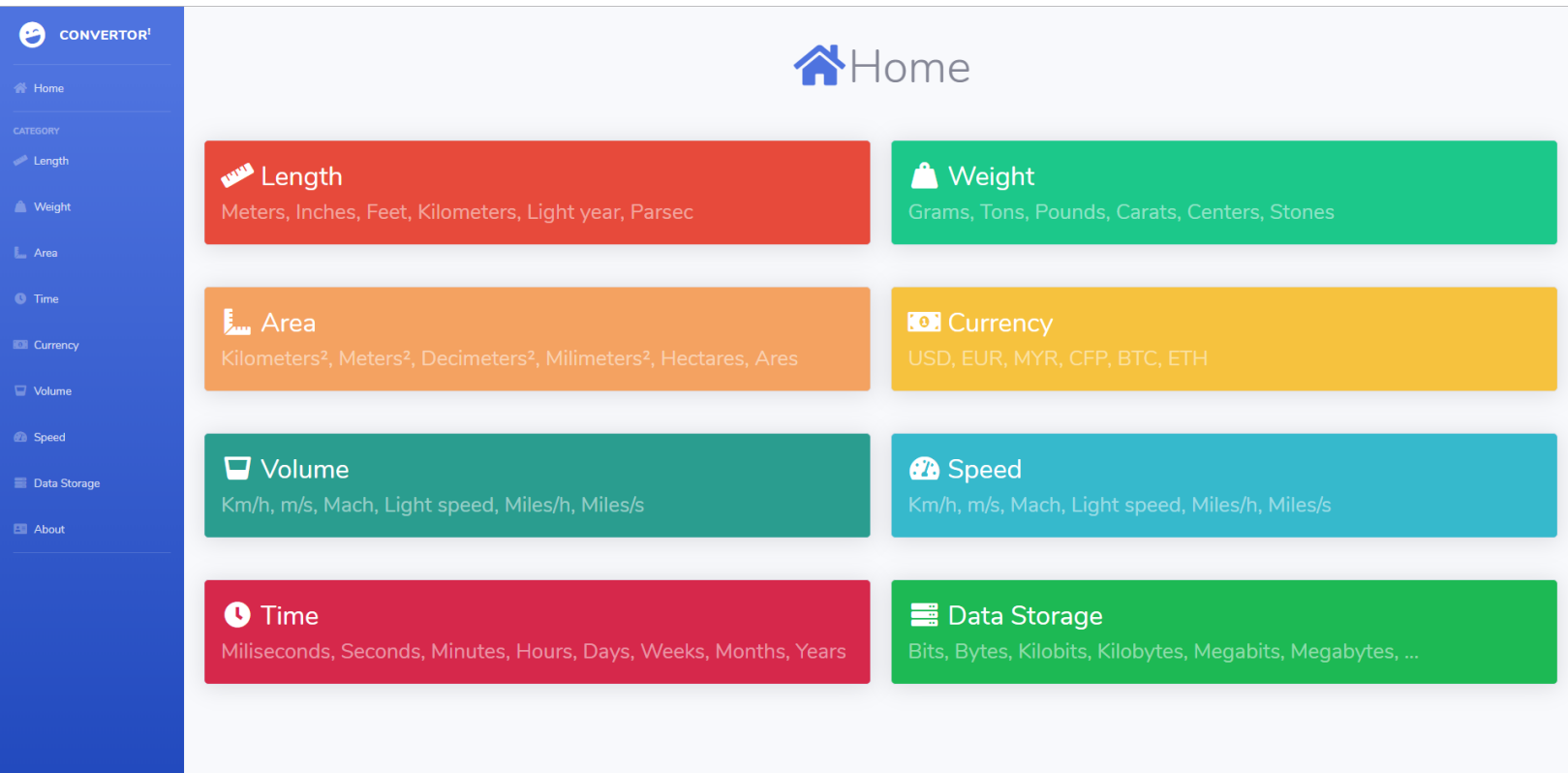
Here it is:  $\text{value} = (\text{value} * \text{fromCoef} * (1 / \text{toCoef}))$ ;


```
public IEnumerable<KeyValuePair<string, double>> DataCoefs = new List<KeyValuePair<string, double>>()
{
    new KeyValuePair<string, double>("USD", 1.0),
    new KeyValuePair<string, double>("EUR", 1.10),
    new KeyValuePair<string, double>("RM", 0.24),
    new KeyValuePair<string, double>("CFP", 0.0093),
    new KeyValuePair<string, double>("BTC", 10716.80),
    new KeyValuePair<string, double>("ETH", 175.55)
};
0 references
protected void Convert(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(TBFrom.Text))
        return;
    double value = double.Parse(TBFrom.Text);
    double fromCoef = DataCoefs.FirstOrDefault(x => x.Key == DropDownList1.Text).Value;
    double toCoef = DataCoefs.FirstOrDefault(x => x.Key == DropDownList2.Text).Value;
    value = (value * fromCoef * (1 / toCoef));

    Result.Text = value.ToString();
}
```

Above is an example of the code used to convert currencies between them.

## Screenshots



 CONVERTOR<sup>1</sup>

Home

CATEGORY

Length

Weight

Area

Time


Currency

Volume

Speed

Data Storage

About

 Weight

From


Gram

>>

To

Grams

0.0

 CONVERTOR<sup>1</sup>

Home

CATEGORY

Length

Weight

Area

Time


Currency

Volume

Speed

Data Storage

About

 Area

From

Kilometers<sup>2</sup>


>>

To

Kilometers<sup>2</sup>

0.0



 CONVERTOR!

Home

CATEGORY

Length

Weight

Area

Time


Currency

Volume

Speed

Data Storage

About

 Time

From


Milliseconds

>>

To

Milliseconds

0.0

 CONVERTOR!

Home

CATEGORY

Length

Weight

Area

Time


Currency

Volume

Speed

Data Storage

About

 Currency

From


USD

>>

To

USD

0.0

 CONVERTOR!

Home

CATEGORY

Length

Weight

Area

Time


Currency

Volume

Speed

Data Storage

About

 Volume

From


Cubic Meters

>>

To

Cubic Meters

0.0

 CONVERTOR!

Home

CATEGORY

Length

Weight

Area

Time


Currency

Volume

Speed

Data Storage

About

 Speed

From

Km/h

>>

To

Km/h

0.0

CONVERTOR!

Home

CATEGORY

Length

Weight

Area

Time

Currency

Volume

Speed

Data Storage

About

»»

Data Storage

From

Bits

»»

To

Bits

0.0

CONVERTOR!

Home

CATEGORY

Length

Weight

Area

Time

Currency


Volume

Speed

Data Storage

About

About



This converter application is a school project from Web Application Development class at UnikL.  
It have been developed with ASP.NET framwork using Bootstrap 4 and FontAwesome 5.

Svg illustrations come from [unDraw](#) website.

[View the assignment soft copy →](#)

### MARKING RUBRIC

Components	No work done <-- Marks --> Excellent work				
1. Basic Functionality (40%)	Almost nothing functions.	Limited functionality. Not achieve the minimum requirements.	Fulfil minimum functionality requirements. Some do not work.	All functions working correctly. Some application not so practical.	All functions working correctly. Application very practical.
	0	10	20	30	40
2. Interface Design (25%)	Interface design almost non-existent.	Too little design or too cluttered.	Acceptable but quite basic.	User friendly but not so beautiful.	Beautiful design and user friendly.
	0	6	12	18	25
3. Additional Features (15%)	No additional features.	1 additional features.	2 additional features.	3 additional features.	4 additional features.
	0	3	7	11	15
4. Documentation (20%)	No documentation given.	Some requirements are not fulfilled.	Fulfill all requirements. Structure and formatting not so good.	Fulfill all requirements. Structure and formatting quite basic.	Fulfill all requirements and proper structure and formatting.
	0	5	10	15	20
Total (100%)					