

Développement des applications mobiles (DAW)

Project theme : Create a personal website for a web designer (html, css, javascript)

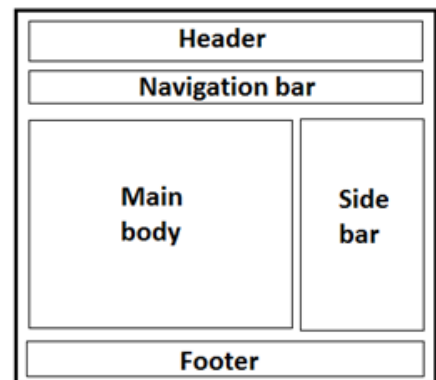
Prepared by : Grine Rayenne Section (1) Group (06)

The site contains five pages which are :

- « **HOME** » : Page contains name of designer, little introduction and her photo.
- « **ABOUT** » : Page contains her cv (work experience, education, personal informations, ...).
- « **CONTACT** » : Page for visitors to contact the web designer by sending a message or social profiles.
- « **WORK** » : Page of photos to see work examples.
- « **SIGN UP** » : Page containing a registration form and authentication.

Most of sites consist of :

- Header <header>
- Navigation bar <nav>
- Main body <body>
- Side bar <aside>
- Footer <footer>



I've worked with :

Html « Hypertext Markup language » a language responsible for the structure of the web page. Define headings, paragraphs, texts and images.

Css « Cascading style sheet » a language responsible for the style and layout of the web page. Define new formats for colors, fonts, alignment methods, and create some animations.

Javascript an object oriented scripting language. It allows, among other things, to introduce small animations or effects on a web or HTML page.

HOME

<!DOCTYPE html> it's an information to the browser about what document type to expect.

<html> tag represents the root of an HTML document.

<!-- --> comment tag used to insert comments in the source code.

<head> a container for metadata and is placed between the **<html>** tag and the **<body>**.

<link rel="stylesheet" type="text/css" href="style.css"> Link between HTML and CSS.

<title>Know ME</title> title of the document which must be text only.

</head> close of tag **<head>**.

<body> defines the document's body, it contains all the content of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc

<div id="nheader"> private ID represents a container for introductory content or a set of navigational links.

**** defines an unordered (bulleted) list.

|Home defines a list item(home's page)

|About defines a list item(about's page)

|Contact defines a list item(contact's page)

|Work defines a list item(work's page)

|Sign Up defines a list item(signup's page)

**** close of tag ****.

<div class="left"> Generic ID.

** **

Set a logo site picture with 85px width /Folder(images) /Name(icone)/ Type(png).

</div> close of class tag **<left>**.

<h1> KnowMe</h1> Web title defines a hyperlink (href) which link to the home's page

<p class="name"> Green Ryan</p> defines a paragraph(name of designer).

**
** inserts a single line break.

<p class="minipara"> defines a paragraph.

** ** leave a space **"As a freelance web designer ... and needs of my client."**

</p> close of tag **<p>**. **</div>** close of tag **<div>**. **</body>** close of tag **<body>**.

<div id="footer"> private ID of a footer for the website.

<p>©2020 All rights ... Grine Rayene. defines a paragraph of footer.

</p> close of tag **<p>**. **</div>** close of tag **<div>**. **</html>** close of tag **<html>**.

Work

```
<div class="service">
```

```
  <table>
```

```
    <tr>
```

```
      <td>  </td>
```

```
      <td>  </td>
```

```
      <td>  </td>
```

```
    </tr>
```

```
    <caption> <i> <b> Education </b> </i> </caption>
```

```
  </table>
```

The tag `<table>` used to insert an HTML table.

Inside this tag we find one or more `<tr>`, `<th>`, and `<td>` elements.

The `<tr>` element defines a table row, the `<th>` element defines a table header, and the `<td>` element defines a table cell.

It may also include `<caption>`, `<colgroup>`, `<thead>`, `<tfoot>`, and `<tbody>` elements.

About

```
<video width="700" height="490" controls>
```

```
  <source src="video/webs.mp4" type="video/mp4">
```

```
</video>
```

The tag `<video>` used to insert a video in the website.

used to embed video content in a document, such as a movie clip or other video streams.

Folder(video)/Name(webs)/Type(mp4).

Contact

`dir="ltr">` to list directory titles.

`<meta charset="utf-8">` defines metadata about an HTML document.

`<script type="text/javascript" src="scripts.js"></script>` used to embed a client-side script (JavaScript).

`<form id="myform" onsubmit="return validation()" >` used to create an HTML form for user input.

<input type="text" placeholder="Enter your name" id="name"> specifies an input field where the user can enter data.

<div class="btn">
<input type="submit" >
</div> } create a send button.

Sign up

lang="en" specifies the language of the element's content (English).

<script type="text/javascript" src="script.js"></script> > Link between the client-side script and the HTML, we found it in source (src).

placeholder="Enter your last name" specifies a short hint that describes the expected value of an input field.

<textarea id="about" placeholder="Describe yourself" ></textarea> defines a multi-line text input control.

value="Register" > specifies the value of an <input> element.

Css

* the star is used to designate all the elements of a page at the same time.

{} sign to the beginning and the end of styling.

margin: used to create space around elements, outside of any defined borders.

padding used to generate space around an element's content, inside of any defined borders (padding left, right, top, bottom).

color: Colors are specified using predefined color names or hashtags.

font-size: sets the size of the text.

font-family: a specific font family (like "Times New Roman" or "Arial").

font-weight: sets how thick or thin characters in text should be displayed.

text-decoration: specifies the decoration added to text, and is a shorthand property for:

- text-decoration-line (required)
- text-decoration-color
- text-decoration-style

float: specifies how an element should float (left, right, none).

display: specifies if/how an element is displayed (block, inline).

text-align: specifies the horizontal alignment of text in an element (center, right, justify).

height: sets the height of an element which is set to a numeric value (like pixels, percentages) .

background-image: specifies an image to use as the background of an element.

font-family: Type of font like Generic family (serif or monospace) and font family(times new roman or arial).

font-size: sets the size of the text by pixels.

background-attachment: specifies whether the background image should scroll or be fixed.

width: The width of an element does not include padding, borders, or margins.

background-size: specifies the size of the background images.

background-color: specifies the background color of an element.

margin-left: sets the left margin of an element.

margin-right: sets the right margin of an element.

display: specifies the display behavior (the type of rendering box) of an element.

max-width: If the content is larger than the maximum width, it will automatically change the height of the element.

margin-bottom: sets the bottom margin of an element.

border: allow to specify the style, width, and color of an element's border.

resize: defines if and how an element is resizable by the user.

letter-spacing: increases or decreases the space between characters in a text.

transition: allows to change property values smoothly, over a given duration.

Javascript "scripts.js"

```
function validation(){
```

```
    var name = document.getElementById("name").value;
```

```
    var subject = document.getElementById("subject").value;
```

```
    var phone = document.getElementById("phone").value;
```

```
    var email = document.getElementById("email").value;
```

```
    var message = document.getElementById("message").value;
```

```
    var error_message = document.getElementById("error_message");
```

```
    error_message.style.padding = "10px";
```

```
    var text;
```

variables are
containers for
storing data
values.

```
if(name.length < 5){  
  text = "Please Enter valid Name";  
  error_message.innerHTML = text;  
  return false;  
}
```

If (name) is less than 5 letters, this function alerts a message(Please Enter valid Name), and returns false, to prevent the form from being submitted.

```
if(subject.length < 10){  
  text = "Please Enter Correct Subject";  
  error_message.innerHTML = text;  
  return false;  
}
```

If (subject) is less than 10 letters, this function alerts a message(Please Enter Correct Subject), and returns false, to prevent the form from being submitted.

```
if(isNaN(phone) || phone.length != 10){  
  text = "Please Enter valid Phone Number";  
  error_message.innerHTML = text;  
  return false;  
}
```

If (phone) isn't equal 10 numbers, this function alerts a message(Please Enter valid Phone Number), and returns false, to prevent the form from being submitted.

```
if(email.indexOf("@") == -1 || email.length < 6){  
  text = "Please Enter valid Email";  
  error_message.innerHTML = text;  
  return false;  
}
```

If (email) is less than 6 letters, this function alerts a message(Please Enter valid Email), and returns false to prevent the form from being submitted.

```
if(message.length <= 140){  
  text = "Please Enter More Than 140 Characters";  
  error_message.innerHTML = text;  
  return false;  
}
```

If (message) is less than 140 letters, this function alerts a message(Please Enter more than 140 characters), and returns false to prevent the form from being submitted.

```
alert("Message sent successfully!");  
return true;  
}
```

If the previous conditions are available, this function alerts a message(Message sent successfully), and returns true.

Javascript "script.js"

```
function validation(){  
var firstname = document.getElementById("firstname").value;  
var lastname = document.getElementById("lastname").value;  
var email = document.getElementById("email").value;  
var password = document.getElementById("password").value;  
var npassword = document.getElementById("npassword").value;  
var about = document.getElementById("about").value;  
var error_message = document.getElementById("error_message");  
error_message.style.padding = "10px";
```

variables are
containers for
storing data
values.

```
var text;
```

```
if(firstname.length < 5){  
    text = "Please Enter valid first-name";  
    error_message.innerHTML = text;  
    return false;
```

If (firstname) is less than 5 letters, this
function alerts a message(Please Enter valid
first-name), and returns false, to prevent the
form from being submitted.

```
}
```

```
if(lastname.length < 5){  
    text = "Please Enter valid last-name";  
    error_message.innerHTML = text;  
    return false;
```

If (lastname) is less than 5 letters, this
function alerts a message(Please Enter
valid last-name), and returns false, to
prevent the form from being submitted.

```
}
```

```
if(email.indexOf("@") == -1 || email.length < 6){  
    text = "Please Enter valid Email";  
    error_message.innerHTML = text;  
    return false;
```

If (email) is less than 6 letters, this
function alerts a message(Please
Enter valid Email), and returns false
to prevent the form from being
submitted.

```
}
```

```
if(password.length < 5){  
    text = "Please Enter correct password";  
    error_message.innerHTML = text;  
    return false;
```

If (password) is less than 5 characters, this
function alerts a message(Please Enter correct
password), and returns false, to prevent the
form from being submitted.

```
}
```

```
if(npassword.length < 5){  
  text = "Please Enter your previous password"  
  error_message.innerHTML = text;  
  return false;  
}
```

If (npassword) is less than 5 letters, this function alerts a message(Please Enter your previous password), and returns false, to prevent the form from being submitted.

```
  alert("successfully registered!  
  Welcome to our family ♥");  
  return true;  
}
```

If the previous conditions are available, this function alerts a message (Successfully registered! Welcome to our family), and returns true.