Project Report

|  |  |
| --- | --- |
| Product Name | Advanced Certificate in Web Development |
| Qualification Name (NICF) | Advanced Certificate in Software Applications (Development and Deployment) |
| Product Name | Front End Web Development |
| Module Name (NICF) | NICF - Front End Web Development |

|  |  |  |  |
| --- | --- | --- | --- |
| Student name | | Assessor name | |
|  | |  | |
| Date issued | Completion date | | Submitted on |
|  |  | |  |
|  | |  | |
| Project title | Development of website for a Training Organization | | |

|  |
| --- |
| Learner declaration |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student signature: Date: |

Content

1. Project background
2. Project Objectives
3. Requirement Specifications
4. Task 1
5. Task 2
6. Task 3
7. Task 4
8. Task 5
9. Task 6
10. Task 7
11. Task 8
12. Task 9
13. Task 10
14. Task 11
15. Task 12
16. Task 13
17. Task 14
18. Task 15
19. Task 16
20. Task 17

Project Background

|  |
| --- |
| You have been approached by ‘ABC Learning Center’ as a website developer to develop a website for their institute. ‘ABC Learning Center’ is an IT Training institute that conducts courses like ‘Programming in Java’, ‘Programming in .Net’, ‘Microsoft Office’, ‘HTML’ etc. They want to design a website which can be used for providing information about their courses to students and for capturing leads generated by digital marketing. |

Project Objective

Tools & platform used

* Visual Studio Code
* Firefox browser
* Apache server
* Jquery & Javascript
* HTML
* CSS

Requirement Specification

1. Container Page (The page which acts as the container for loading main content of the below pages). This is used for creating a Single Page Application.

Note : Container page provides Header, Footer, Navigation and the html, head, body tags. This a single page application all other pages are loaded using Ajax

2. Home Page (Without html, head, navigation, body tag)

3. About Us Page / Profile Page (Without html, head, navigation, body tag)

4. Course Listing Page (All the 4 courses listed in this page must be loaded from a Web Service) (Without html, head, navigation, body tag)

5. Course Details Page (Develop 4 pages, 1 for each course) (Without html, head, navigation, body tag)

6. Course Schedule Page (Develop 4 pages, 1 for each course) (Without html, head, navigation, body tag)

7. Contact Us Page (Develop 1 Html page) (Without html, head, navigation, body tag)

8. Lead Registration Page (Develop 1 form) (Without html, head, navigation, body tag)

9. Registration Thank You Page ( Develop 1 HTML page) (Without html, head, navigation, body tag)

10. Sitemap Page (Develop 1 HTML page) (Without html, head, navigation, body tag)

11. Privacy Policy Page (Develop 1 HTML page) (Without html, head, navigation, body tag)

1. Task 4

1. Write briefly on the purpose of markup language pages.

2. Include notes on purpose of using CSS.

3. Include it as part of **Project Report**.

Solution

1. Markup language pages are the basis of all webpages. Pages must be in HTML for webpages to be displayed as part of the web. This includes the logic and the information that should be presented
2. CSS takes the information of a webpage and styles it in a certain way. This improves the way the information is presented to a user. CSS is mainly for presentation and improvement of user experience.
3. Task 5

Task Statement

* 1. Collate Videos and Images required for the Inner pages.
  2. a. You can use images from free stock photo websites.

Solution

2 images (1 logo and a background image) are used in the project. (Ref: img folder from source code)

1. Task 6

Task Statement:

Solution

1. Task 7

Task Statement

Solution

|  |  |  |
| --- | --- | --- |
| S. No. | Selector | Style to be created |
| 1 | Body | Background image |
| 2 | #logo | Height (100px),  width(100px)  Margin(0 5px 0 30px) |
| 3 | Nav | Background color (beige)  Height (5rem)  Margin(0 0 20px 0) |
| 4 | #top-nav | List-styles (None)  Padding (0.92em),  Text aligned (centre),  Font-size (1.3em),  Float left |
| 5 | .content-box | Min-height (400px)  Border (2px solid gray)  Overflow(auto)  Background color (white)  Margin (3em auto 4em auto)  Width (70%) |
| 6 | A | Colour (olive)  text decoration (none) |
| 7 | A:hover | Colour(chartreuse) |
| 8 | #bot-nav | Background color (Beige)  Bottom (0)  Height (125px) |
| 9 | #related-links | Float left,  Text-align (center)  font-size (1.3em)  width (13%)  padding (0 0 0 25px) |
| 10 | #other-links | list-style(none),  font-size (0.8em)  padding (0 0.9em) |
| 11 | Html | Font family:  Gill sans, Gill San MT, Calibri, Trebuchet MS, sans-serif |

1. Task 8

Task Statement

Create a .CSS file for the styles noted in Task 5 using Dreamweaver.

Solution

html{

padding: 0;

margin: 0;

}

body{

margin: 0;

padding: 0rem;

background-image: url("img/blackboard-chalkboard-communication-355988.jpg");

}

/\* Top navigation bar settings \*/

#logo{

float:left;

width: 80px;

height: 80px;

padding: 0 10px;

margin: 0 5px 0 30px;

}

nav{

background: beige;

height: 5rem;

width: 100%;

top:0;

left:0;

margin: 0 0 20px 0;

}

#top-nav ul{

margin: 0;

padding: 0;

}

#top-nav li{

list-style: none;

float: left;

padding: .92em;

text-align: center;

margin: 0;

font-size: 1.3em;

}

#tabs li:hover{

background: wheat;

}

/\* Link settings in general \*/

a{

color:olive;

}

a:hover{

color:chartreuse;

}

a:link{

text-decoration: none;

}

/\* Container settings \*/

.content-box{

background:white;

border: 2px solid gray;

margin: 3em auto 4em auto;

width: 70%;

min-height: 400px;

max-height: 400px;

overflow: auto;

}

/\* Footer settings \*/

#bot-nav{

background: beige;

bottom: 0;

height:125px;

margin:0;

}

#bot-nav h3{

color:greenyellow;

padding: 10px;

text-align: center;

margin:0;

}

#related-links{

height:80px;

}

#related-links ul{

margin: 0;

}

#related-links li{

list-style: none;

float: left;

padding: .92em;

text-align: center;

margin: 0;

font-size: 1.3em;

font-size: .98em;

width: 13%;

padding: 0 0 0 25px;

}

/\* Removes float left \*/

.seperator{

clear: both;

margin: 0;

padding: 0;

}

#other-links{

height:30px;

width:100%;

margin:0;

}

#other-links li{

list-style: none;

float: left;

padding: .92em;

text-align: center;

margin: 0;

font-size: 1.3em;

width:45%;

font-size: .8em;

padding: 0 0.9em;

}

1. Task 10

Task Statement

Demonstrate ways of inserting the style sheet into HTML pages.

Solution

1. Link in html page with <link rel=”stylesheet” href=”your-css-file.css”>
2. Style using style tag within head tag (<head><style></style></head>)
3. Individual styling on the individual tags themselves. <p style=”…”>…</p>
4. Task 13

**Task Statement:**

Determine if the JSON response from mock Course List & Currency webservice is suitable for display purposes.

Write briefly in project report on the suitability.

Solution

Yes it is, as it is easy to work with and easy to send. With little overhead, information can easily be sent between APIs and client programmes. The values are easily referenced using JS.

1. Task 14

**Task Statement:**

Solution

Getting course details from courses.json:

function showCourseDetail(id) {

changePage('course-template.html');

$(document).ready(function(){

$.getJSON("courses.json", function (data) {

$.each(data.courses, function () {

if (this.course\_id === id) {

//change page to course template

$("#c-title").append(this.course\_name);

$("#c-desc").text(this.course\_description);

$("#c-reccomend").text(this.course\_recommend);

$("#c-price").append(this.course\_price);

}

});

});

});

}

Getting currency.json details:

function calcCurrency(currCode, amount){

var url = "currency.json";

var details = "";

$.getJSON(url, function (data) {

// console.log("Currency Json started");

$.each(data.currencies, function(){

if (currCode == this.code) {

// console.log(this.code);

// console.log(currCode);

// console.log("Before change: " + amount);

amount \*= this.conversion;

details += this.name + ": "+ this.code + " " + amount;

// console.log(details);

// console.log("after change: " + amount)

}

})

$("#price-indicator").text(details);

});

}

A single application page strictly only has a single complete html page. Within that page, there will be container(s) that is/are updated only when a request is made to the server via Ajax. These requests then obtains snippets of html code and replaces the container(s) content(s) to show updated data.

1. Task 15

**Task Statement:**

Develop at least 5 Test Cases in a Test Case Document to Test the RIA and ensure it follows the technical design.

Solution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Number | Test Case | Expected Result | Actual Result | Pass/Fail |
| 1 | Test links in webpage | All links except “A link” should work | All links aside from “A link” works | Pass |
| 2 | Test spelling and grammar | Should be grammatically correct. Spelling should be alright | Grammar and spelling is okay | Pass |
| 3 | Title Bar testing | Changes within the address bar should bring about relevant pages | All except “#course-details” brings user to a different page | Pass |
| 4 | Cross browser Checks | All browsers should look similar | There are some difference when displaying #course-list between Chrome and Firefox.  In Internet Explorer, the contents fails to load. | Fail |
| 5 | Test Fonts are consistent | All fonts should follow the family stated in CSS | All fonts are similar. There’s no sudden change | Pass |

1. Task 16

**Task Statement:**

Solution

1. Task 17

**Task Statement:**

Solution