Web Application Development Project Submission 2022

Name: Cormac Hennigan.....

Student ID: G00398284 .....

Date: 16/05/2022.....

Home page/index page/start page (eg., page user should open first): index.html.....

Project Requirements Implementation

ITEM 1	Reference
Allow the customer to enter their login details:	See page “index.html” and the associated JavaScript functions from “scripts.js”.
Login details validated (via a login screen) before receiving a summary of the order:	Validation performed using HTML5 on “index.html” page and JavaScript on “scripts.js”.
Username:	admin@gmit.com
Password:	iloved3
Brief description of how this was implemented:	I used HTML5 form validation in combination with javascript for username and password authentication. The authentication was done using the JavaScript function called “authenticate()” that I created. The function is in “scripts.js” which was imported to “index.html”. The function uses a DOM query to extract the values from both inputs and checks them against the username and password above using an if statement.

ITEM 2	Reference
Perform form validation through JavaScript or HTML to ensure that text fields are not empty, and a valid email address is entered	Created a form in “index.html” using HTML5. The form consisted of an action to bring the user to the home page, two inputs for email and password and a submit input to submit and validate the data.
Brief description of how this was implemented:	Used the attribute “required” in the HTML Form to ensure input fields were not empty. Used the “type=email” attribute to ensure that the email was a valid email string. Used the “type=password” attribute to ensure that the password was displayed as dots.

ITEM 3	Reference
Access and change HTML on the web page through the DOM;	If the user entered the wrong username or password, a message is presented to the user through a <p> element with an id: feedback saying “Username or password invalid”.
Brief description of how this was implemented:	When the form is submitted and the “authenticate()” function is called, if the user input the wrong email or password the document method “getElementByID” is used to find and edit the text of the feedback element on the page “id=feedback”. The element was only updated if the incorrect login information was submitted.

ITEM 4	Reference
Access and change styling through the DOM;	An input element with type=”checkbox” was used on each page to toggle a “dark” class on the body element of the HTML
Brief description of how this was implemented:	In “scripts.js” an event listener was created to listen for a change event on the input checkbox with id = “dark”. When the user clicks the checkbox, the listener will call the callback function provided to it. In this case the callback is used to toggle the “dark” class on the body element of the html. Using CSS, the body selector sets a background and colour for the page and text. I also added a .dark class in the CSS so when the dark class is applied to the body, the body then uses the dark background and light text specified in the .dark class. This feature is on all pages of the site.

ITEM 5	Reference
Demonstrate the use of events;	Darkmode toggle is clicked. “change” event is fired.
Brief description of how this was implemented:	As above. When the input checkbox id: darkmode is clicked a change event is fired. Using an eventListener created in script.js I listen for this event and provide a callback function to the eventListener. This callback function allows me to perform actions only when the event is fired. In this case it toggles the “dark” class on the body element of the html.

ITEM 6	Reference
Contain two D3 data visualisations (e.g., Bar Chart) of your choosing a. One from a CSV file b. One from an array	I used a bar chart that shows the frequency of each letter in the English language as percentages for my CSV d3 visualisation and a graph that shows the average monthly rainfall as a percentage of the norm in Ireland in 2021 for my array d3 visualisation.
Brief description of how this was implemented:	Firstly, the D3 library is imported via a CDN into the page using a script tag. This allows me to use all the D3 functionality inside my own javascript files. The dimensions and margins of the charts are set using variables at the top of the respective js files. Using the “d3” global object I find the element I want to display the chart in, in one case it’s a div with the id set to “letterFrequency”. d3.select is a function that takes the id attribute and using append(“svg”), I append an svg element to the letterFrequency div. The .attr() function allows me to set attributes on the element appended, in the case of the svg, I set the viewbox attribute so the chart will scale with the size specified through the css. Next the data is imported from the csv file using the d3.csv function which takes the path to the csv file I added to the folder. Now with the data, I used more d3 functions to create both axes including the labels and ticks, and position them accordingly. Next I created the body of the graph using the data and finally create animations to apply to the charts/line on each of the graphs.

ITEM 7	Reference
Both visualisations should allow the user to specify display settings, including an option to change colour, display size and animations	Change the colour, size or trigger an animation of the d3 bar chart and graph on the “rain.html” and “letters.html” pages.
Brief description of how this was implemented:	Created 3 buttons to fire the above events. The onclick attribute calls one of 3 functions: "toggleColor()", "triggerAnimation()"and "changeSize()”. There are 3 boolean variables at the top of “letters.js” and “rain.js” with each of these events set to false (colorToggled, sizeToggled and animationTriggered). Within each of the functions when the function has performed its task, the boolean is toggled to allow the function to be called again and perform the reverse function.

ITEM 8	Reference
Have a minimum of 3 linked pages;	The pages created are “index.html”, “home.html”, “rain.html” and “letters.html”.
Brief description of how this was implemented:	The pages each contain a basic skeleton html structure. Navigation is performed using <a> tags within a semantic nav element. ./ is used at the beginning of the href attribute to denote a relative path as we are navigating within the same folder.

Additional information:.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....