ArMean

"your go-to website for calculating statistics"

Michael Samuel 191900135
Mohamed Ahmed 191900111
Abanoub Adel 191900173

Presented to: Dr. Abdalla Mohamed

May 25, 2023

PROJECT			
	NODE_MODULES		
	PUBLIC		
		USERS_DATA	//contains registered accounts
		REGISTER.HTML	Inside the registerUser() function, the values of the username, email, and password fields are retrieved from the respective input elements. A validation check is performed to ensure that all fields are filled. If any field is empty, an alert message is displayed.
			If all fields are filled, a data object is created with the user information. This data object is then sent to the server using a POST request through the fetch() function. The server expects the data to be in JSON format.
			If the server responds with an OK status, a success message is displayed through an alert, and the user is redirected to the "index.html" page. If there is an error during the registration process, an error message is displayed in the console and an alert is shown to the user.
		INDEX.HTML	Inside the login() function, the values of the email and password fields are retrieved from the respective input elements. The email is used to generate a filename for the user's data file.
			The function then makes a fetch request to retrieve the user's data file from the "users_data" directory. If the response is successful, the data is parsed as JSON.
			Next, the function compares the entered password with the stored password from the retrieved data. If the passwords match, a success message is displayed through an alert, and the user is redirected to the "ArMean.html" page. If the passwords do not match or there is an error during the login process, an error message is displayed.

PROJECT			
		ARMEAN.HTML	When the user clicks the "Calculate" button, the calculateStatistics() function is triggered, which is defined in the external "script.js" file. This function performs the necessary calculations to determine the arithmetic mean, variance, and standard deviation based on the entered values.
		ABOUT.HTML	provides a brief overview of the purpose of the website and acknowledges the group members
		SCRIPT.JS	The function calculateStatistics() begins by retrieving the input values from an HTML element with the ID "values". The values are expected to be entered as a comma-separated string. The string is then split into an array of individual values. The function proceeds to initialize variables for the sum, mean, variance, and standard deviation. It iterates over the values array, converting each element to a float and accumulating the sum. After calculating the sum, the mean is computed by dividing the sum by the number of values. Next, the function calculates the variance by iterating over the values array again. It subtracts the mean from each value, squares the result using Math.pow(), and accumulates the squared differences. The variance is then divided by the number of values to obtain the average variance. Finally, the standard deviation is calculated by taking the square root of the variance using Math.sqrt(). The results are displayed by updating the text content of specific HTML elements identified by their IDs: "meanResult", "varianceResult", and "standardDeviationResult".

PROJECT

	STYLE.CSS	The CSS styles begin with a universal selector (*) to reset margins, paddings, and box-sizing for all elements. The body element is set to flex display, center-aligned, with a background image and cover. The header is fixed at the top with a background color and contains a logo and navigation links. The navigation links have hover effects that display an underline animation. There is also a "Register" button with a hover effect. The main content is wrapped inside a container element with a specific width, height, background, border, and box-shadow. Form elements such as input boxes, labels, and icons are styled. The labels are positioned above the input boxes and move up when the input is focused or contains a value. The "Login" button has specific dimensions, background color, border, and text color. Additionally, there is a separate section of CSS styles for a container element outside the given code snippet. It sets the maximum width, padding, text alignment, and color for a different container element.
	BG.JPG	

PROJECT		
INDEX.JS		The code begins by requiring necessary modules such as express, path, and fs. An instance of the Express application is created, and static file serving is enabled from the "public" directory using express.static middleware. JSON body parsing is also enabled using express.json middleware.

Two routes are defined:
The root route ('/') responds with the 'index.html' file located in the "public" directory.

The '/register' route handles a POST request for user registration. It expects the request body to contain fields such as username, email, and password. Validation is performed to ensure all fields are provided. If any field is missing, a 400 response with an error message is sent.

If all fields are present, a data object is created with the user information, converted to JSON, and saved to a file in the "public/users_data" directory.

The filename is based on the user's email, and the file path is defined using path.join. The data is written to the file using fs.writeFile. If any error occurs during the file write operation, a 500 response with an error message is sent.

Otherwise, a success message is logged to the console, and a response indicating successful registration is sent.

Finally, the server is started on port 80, and a message is logged to the console indicating the server's running status. |
| **PACKAGE.JSON** | | //created upon installing express |
| **PACKAGE-LOCK.JSON** | | //created upon installing express |

Watch a Demo: Explore How the Website Works!

- [ArMean Website](#)