MISWA
TEAM PIONEERS

Developer Documentation (Website)

Authors

Mariam Jobe

Factors

- The programming language used to create the backend of the website is PHP and Javascript.
- The language for the user interface is HTML and CSS, a template from Bootstrap Admin.
- The platform tool used to develop the application is brackets and dreamweaver.
- The data is retrieved from phpmyadmin database , which uses mySQL as a language. The connection between the web page and the database is done via PHP.
- The website runs locally on the computer. Use something like XAMPP to set it up. Add the whole folder to htdocs and type localhost in your web browser to access the website.

Software build process

Backend development

The backend is split up into different files, all the coding is done in PHP and Javascript:

	FILENAME	FUNCTIONALITY
	test.php	Creates the database connection. It has all of the login information for phpmyadmin.
	test2.php	Checks if username and password is entered correctly else it gives an error message. Creates a session variable for the Company Id.
	signup.php	Adds a new company to the database using the posted information with a unique Company Id.
	add.php	Adds an employee to the database. Checks if the username is equal to the password and checks if the username has not been taken already, else it adds it to the database. Uses the session variable so the user is added to the right company.
	mission.php	Adds a mission to the database. Adds all the employees connected to the logged in company id and puts it in an array to be able to display it in a dropdown list. Then it gets all of the posted values and adds it in the insert query. If it is successful a pop-up message displays saying so, else it displays a pop-up message to try again. Also uses the session variable to display everything in the table.
	logout.php	Destroys all of the session variables, and exits all of the pages and goes to the index file(login page).

Design
The CSS folder contains style.css, which sets the size of the different components as well as adding fonts, positioning and coloring.

In the font-awesome folder contains Bootstrap.css, makes the website responsive, it handles table borders, sets font sizes, colors the background etc.

Custom-styles.css has similar features as the css files above.

In the images folder we have all of the pictures that are used in the UI as well as some of the icons.

The UI uses CSS and HTML, the design is a modified bootstrap template:

NAME	SECTION	FUNCTIONALITY
index.html	Login page	Uses **style.css**, javascript and **font.googleapis**. The form is executed with **test2.php**,as an action. If the user wants to sign up it has a reference to **companysignup.php**.
companysignup.php	Sign up page	Uses the **style.css** for graphics, javascript for the event listeners and some **googleapis** for the fonts. Displays text fields in HTML for user inputs and has a reference to **index.html** if the user already has an account. The action is executed with **signup.php**.
maindesign.html	Dashboard	Uses bootstrap.css, style.css and morris-0.4.3.min.css as well as the google apis for the fonts. The navbars have references to **logout.php**, **driverTable.php** and **table.php**. The javascript executes the image slideshow at the top. The icons in the textboxes are glyphicons which are compatible with bootstrap.

driverTable.php	Manage Employee	Úses the CSS files, the google apis and javascript and php. The backend php functionality at the top of the file checks if edit is clicked and redirects to **driveredit.php**. The value of the checkboxes are the Employee Id's of the users in the table. It checks if the delete is clicked and puts everything from the row of that checkbox into an array and executes a delete query of the array. Refreshes the page and displays pop-up message in javascript. If the reset password button is clicked, it puts everything from the row of that checkbox into an array and executes an update query in sql. This also displays a pop-up message in javascript. The table is populated with live database values. The values are getting retrieved using a select statement based on the Company Id. It is getting the values based on which company id is currently stored in the session variable. If the user click on new, it opens a modal pop up with a form of text fields. The form is executed with add.php. All the values the user enters in the text fields gets stored in variables and are user in an sql insert statement in **add.php**.

driveredit.php	Edit employee	The php code gets the Employee Id of the ticked checkbox to prefill the form when the page opens with a select statement that uses this Employee Id. The HTML form the CSS files and the javascript functions just as the previous files. The text the user enters in the input fields gets stored in variables which are used in the sql Update statement.
table.php	Manage missions	Uses the CSS files, the google apis and javascript and php. The backend php code is checking if the edit button is clicked and and redirects to **editMission.php**. The checkbox values are the Mission ID's. If the delete button is clicked it takes the Mission ID of the checkbox row and uses it in the sql delete statement. If the deleted row gives a result it displays a pop-up message in javascript. The table is populated with live database values. The values are getting retrieved using a select statement based on the session variable Company Id. It is getting the values based on which company id is currently stored in the session variable. If the user click on new, it opens a modal pop up with a form of text fields. The form is executed with **mission.php**. All the values the user enters in the text fields gets stored

		in variables and are user in an sql insert statement in mission.php. All employee names that belong to the session variable are displayed in a dropdown list when the user is adding a new mission.
editMission.php	Edit/Update mission	The php code gets the Mission ID of the ticked checkbox to prefill the form when the page opens with a select statement that uses this Mission ID. The HTML form the CSS files and the javascript functions just as the previous files. The text the user enters in the input fields gets stored in variables which are used in the sql Update statement.

Overview of the functionality and how they are related: