2DoList Web Application Report

Phase 2

Name:Hisham Osama Abd-Alsalam Ghazy
ID: 85

Name: Ahmed Khaled Saad Abd-Alaziz
ID: 07

Introduction to 2DoList Web Application:

2DoList is web application that allows you to track your To-Do and task lists. Example commercial applications are: Todoist and Wunderlist.

Here we are building this web application in multiple phases. In the first phase of the project, we created the main home page that lists all the tasks and all the options that a user is allowed to do to these tasks.

A user of your 2DoList should be able to add new tasks, update the state of a task, etc. This web application has used the proper frameworks that would allow us to apply 2DoList on different web browsers.

Code Organization:

In this phase the code:

*Node JS:

Used basically in catching any request from the user-URL at server side and make some operations based on that request, then response to the user with the required data. Also used to read data from file and send as response if required.

*HTML:

It's the part where some tags are used to display the basic table where the tasks should appear later, it is also used where other tags are used to display the pop up window that the user will enter the tasks in it later. Used in making user-interface that appears to the user on requesting our local-port.

*CSS(with boot-strap):

It's the part where styles are applied for different parts of the web page. However, Boost-strap used to generate good well-designed style. Used in making user-interface that appears to the user on requesting our local-port.

*JavaScript:

It's the main part of the project where all the functions controlling the flow of the application.

The main functions used in this section will be discussed later.

Main functions:

*validateEmail:

This function makes validations on the entered e-mail whether in registration or logging-in.

*validatePassword:

This function makes validations on the entered password whether in registration or logging-in.

*validateName:

This function makes validations on the entered user name whether in registration.

Note: The other functions are on request-response status.

The flow of the server-side:

*The user signed up with a new e-mail. After signing-up the data entered to the write-text is picked at the html-page interface and is sent to server-side which make validations on the entered text and checks if it is valid. If it's valid it response to the user the front-end of the main To Do List made in phase 1. The user add his own tasks and make operations and what he wants to do on them.
*After that the four arrays which stores all data of the tasks that the user made lately are then saved in a file for that user.
*Each user has each own file created on signing up and filled on making any operations on his own tasks.
*Finally when the user logs in with his email and password, the server-side requests the' To Do-list' html page. In that html file the required user file is uploaded and the four arrays are then filled with the last saved data by that user and displays completely in front of that user.

Finally:

Most of the main methods are made together by us, but there some of individually work
Team Member 1 ('Hisham');
Implementation of most of 'Node-JS' flow functions and home page interface.
Team Member 2 ('Ahmed');
Implementation of the requests related to files and some modifications on the phase-1 document to be prepeared to be used in phase-2.