WEB APPLICATIONS FINAL REPORT
CSCE 685 DIRECTED STUDIES

NIMOSHIKA JAYARAMAN
UIN: 631000852

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TEXAS A&M UNIVERSITY, COLLEGE STATION

TABLE OF CONTENTS

1. Summary

Three web applications “To-Do list”, “Timer” and “Temperature Convertor” were designed based on the day to day application of them in mobile devices.

This was done as a part of the Directed studies project inspired by the project that was done by our team as a part of a software engineering course. These applications were implemented using HTML 5, standard CSS and basic JavaScript. The requirement was to implement these applications in the local web browser provided the basic coding files. HTML5 is a markup language used for structuring and presenting contents on the web. Cascading style sheets(CSS) is a style sheet language used for describing the presentation of a document written in HTML. JavaScript is used to create dynamic and interactive web applications and browsers.

Agile software development methodology was followed and user stories were developed that reflected the requirements. This helped in the step by step development of the application to enhance the user experience. After each iteration, the changes were implemented and checked for better performance.

To-Do list application helps in creating new tasks, updating and deleting the existing task. Timer application helps in having a counter time, starting and stopping the timing whenever as needed. Temperature conversion application is used to convert the given temperature into celsius, fahrenheit and kelvin.

This report gives a detailed explanation on various user stories and how it was developed. This will help in the future enhancement or development of the given applications.

2 Development And Management

2.1 BDD/TDD

Jasmine could be used for unit testing the code. Testing was initially tried but as the priority was towards development of new applications, a decision was made to proceed on with the development part. The development of the application was successful and tested post the development.

2.2 Configuration Management

The code management was done using few releases and one branch as user stories were created completely modular and separate from other user stories. With one branch and few releases, it was easy to manage. All development was done towards the completion of user stories at the end of each iteration. There was no spike in the project.

3 User Stories

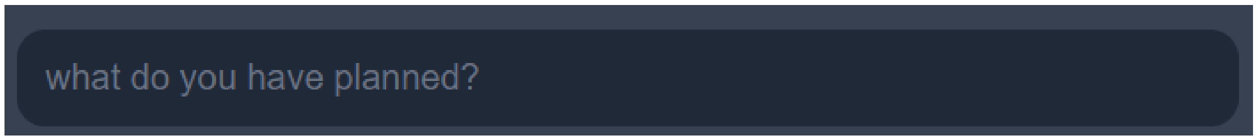
3.1 User stories related to the To-Do List Application:

3.1.1 User story: Implement user friendly style for the application with CSS and HTML

Points:1. Status: complete. Description: As an application user, the user interface should be clear and easy to understand and use. The implementation should be done appropriately so the user knows where the task has to be entered and the buttons should be clear. For example, the space where the task has to be entered should display “What do you have planned?” and so on.

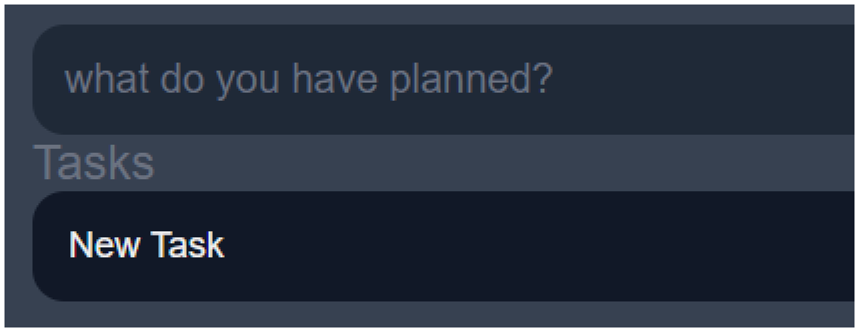
3.1.2 User story: Setup the page structure for the addition of new task

Points:2. Status: complete. Description: As a user only one task should be entered at a time. The implementation should be done appropriately so that the task added should be displayed properly and the new tasks should be following the old task.



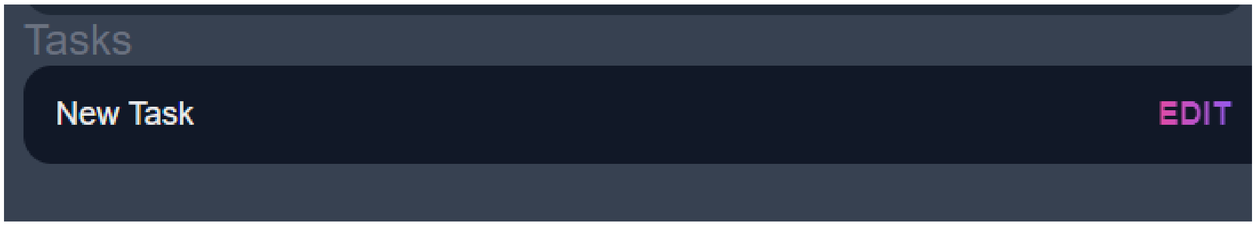
3.1.3 User story: Display the new task added

Points:2. Status: complete. Description: As a user the new task added should be displayed one followed by the other. The more the task is added, the more the task should be displayed one below the other.



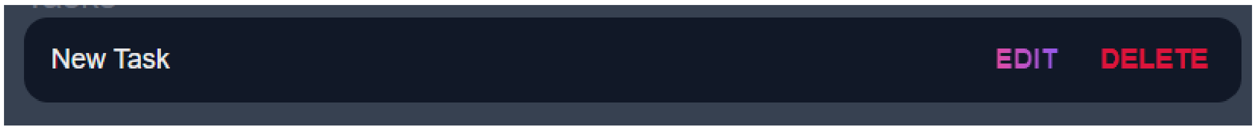
3.1.4 User story: Implementation of Edit button

Points:2. Status: complete. Description: As a user the Edit button has to be displayed for each task and the edit action should serve the purpose once it is implemented. The task has to be edited appropriately. Add task button is also added on the side of each task.



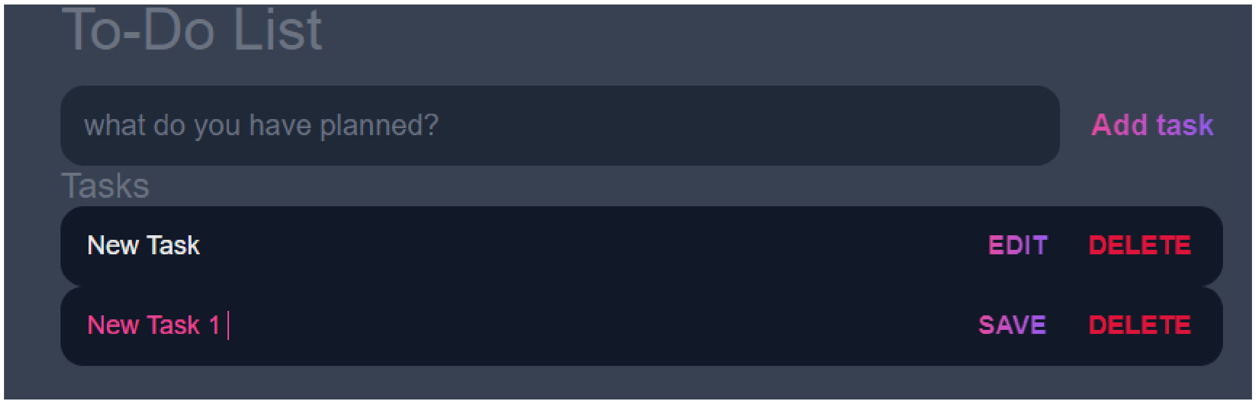
3.1.5 User story: implementation of Delete button

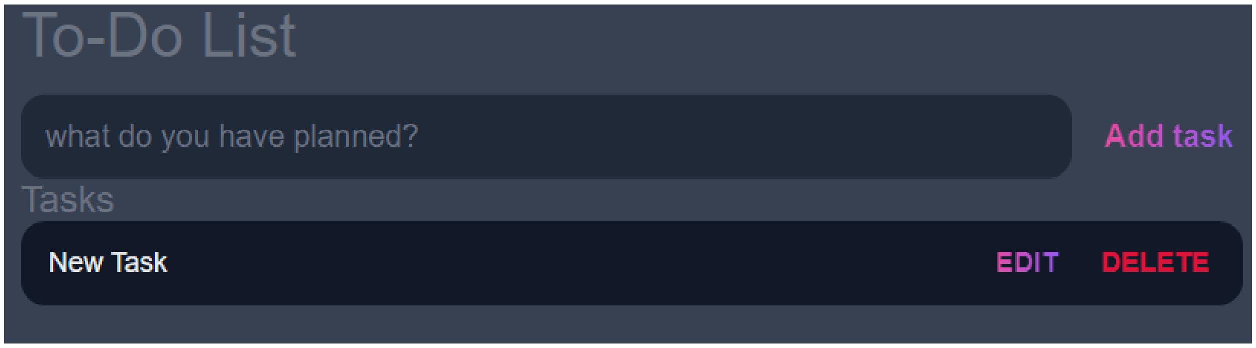
Points: 2. Status: complete. Description: As a user delete button should be displayed and implemented properly beside each task. Once the delete button is pressed, the task must be deleted from the list displayed.



3.1.6 User story: display the appropriate task after edit and delete

Points: 3. Status: complete. Description: As a user the edit and delete options should function correctly when used alternatively. When the edit button is pressed, the cursor should appear for the changes to be made in the task and the delete should delete the task from the entire list. After edit is pressed, the save button should appear.

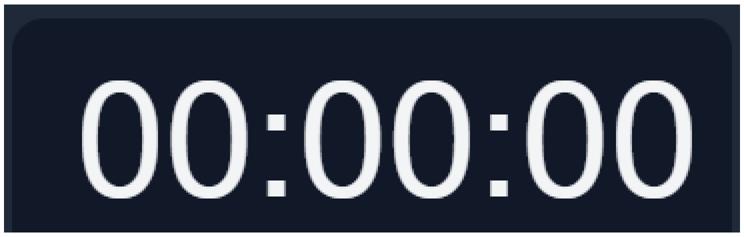




3.2 User stories related to the Timer Application:

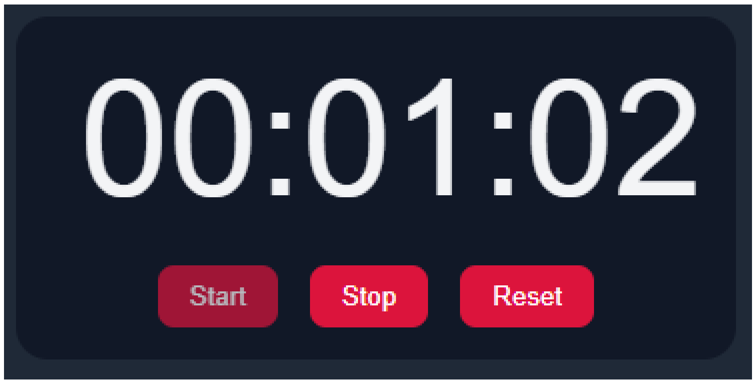
3.2.1 User story: Implement user friendly style for the application with CSS and HTML
Points 2. Status: complete. Description: As a user, the application should be easy and clear to use. The time should be displayed appropriately and easy to understand.

3.2.2 User story: setup the page structure for the running timer
Points 2. Status: complete. Description: As a user, the page structure should be right. The timer should be displayed on the center of the page with the hours, minutes and seconds correctly with a semicolon in between them.



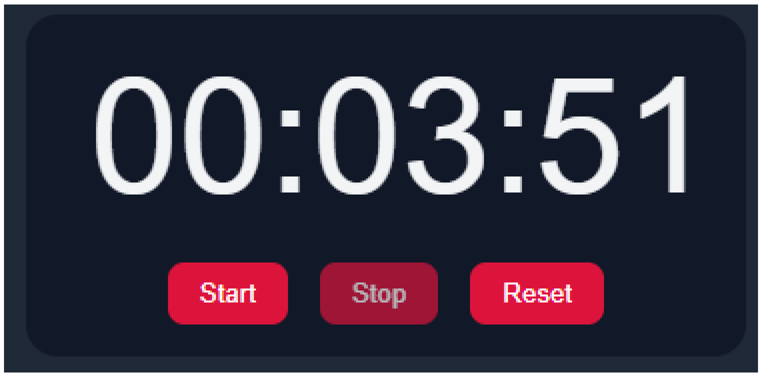
3.2.3 User story: Implementation of start button

Points 2. Status: completed. Description: Once the user clicks the start button, the time should start running from the seconds. A start button is implemented in such a way that when the user clicks on it, first the seconds start running until 60, then the minutes start running and so on.

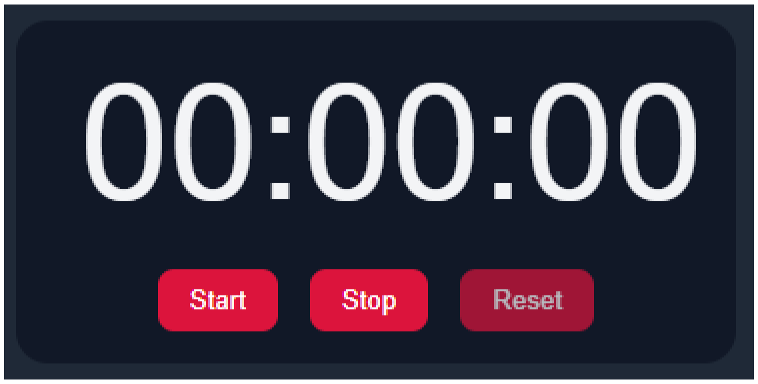


3.2.4 User story: Implementation of stop button

Points 2. Status: Completed. Description: As the user clicks the stop button the timer should stop abruptly at that point and display the time.



3.2.5 User story: Implementation of Reset button:
Points 2. Status: Completed. Description: As the user clicks on the reset button, the timer should go back to 00:00:00.

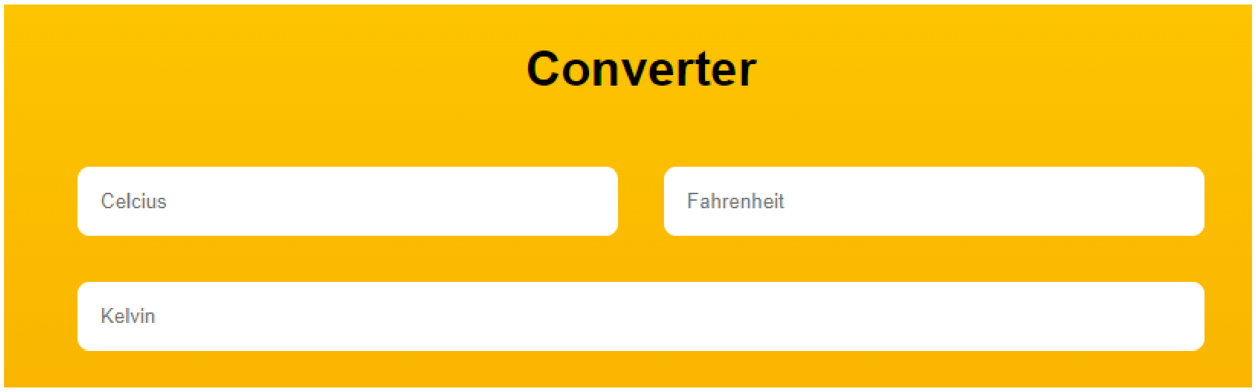


3.2.6 User story: Display the appropriate time after the start, stop and reset:
Points 3. Status: Completed. Description: As the user clicks on the buttons, it should work appropriately and serve the purpose. The time should work in such a way that when the user clicks on start and uses the timer and when the calculated time is over and clicks on stop the time should be calculated correctly and be resetted again to the beginning.

3.3 User stories related to the Temperature Convertor Application:

3.3.1 User story: Implement user friendly style for the application with CSS and HTML:

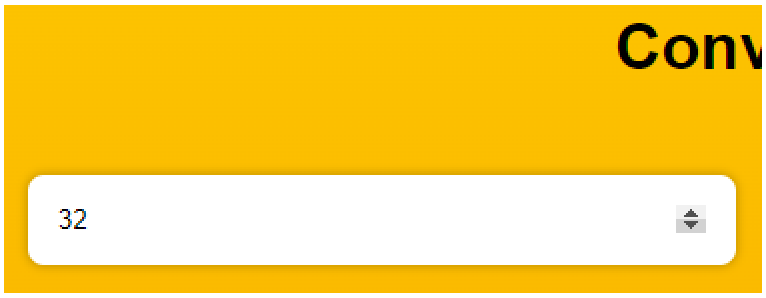
Points 2. Status: complete. Description: As the user uses the application, it should be easy and clear to use. The different settings (celcius, fahrenheit and kelvin) should be mentioned properly in the appropriate place.



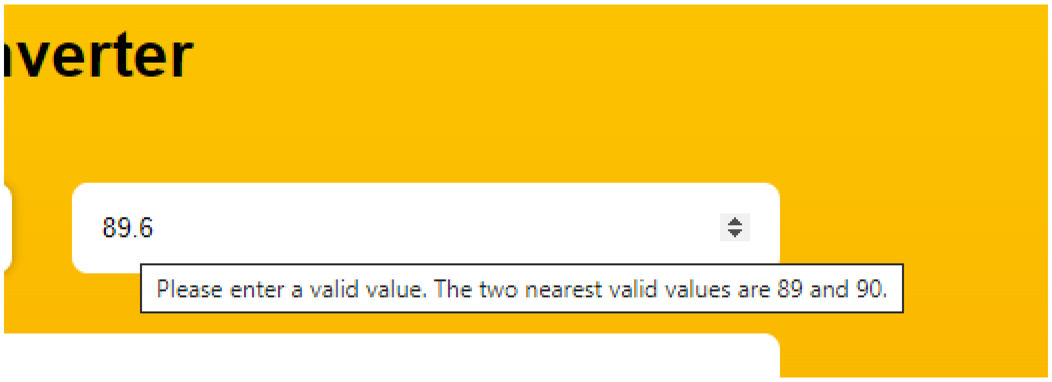
3.3.2 User story: setup the page structure for the Temperature Convertor display:

Points 2. Status: Complete.Description: As a user, the converter has to be aligned in the center of the page with celsius, fahrenheit and Kelvin in that order. When the user enters the temperature it has to be converted within the space and displayed correctly in the page.

3.3.3 User story: Implementation of celsius:
Points 2. Status: Completed. Description: As a user, the celsius has to be displayed first with up and down arrow adjustments. The box for celsius has to have the up and down arrow for adjustment of temperature dynamically without having to input it manually.



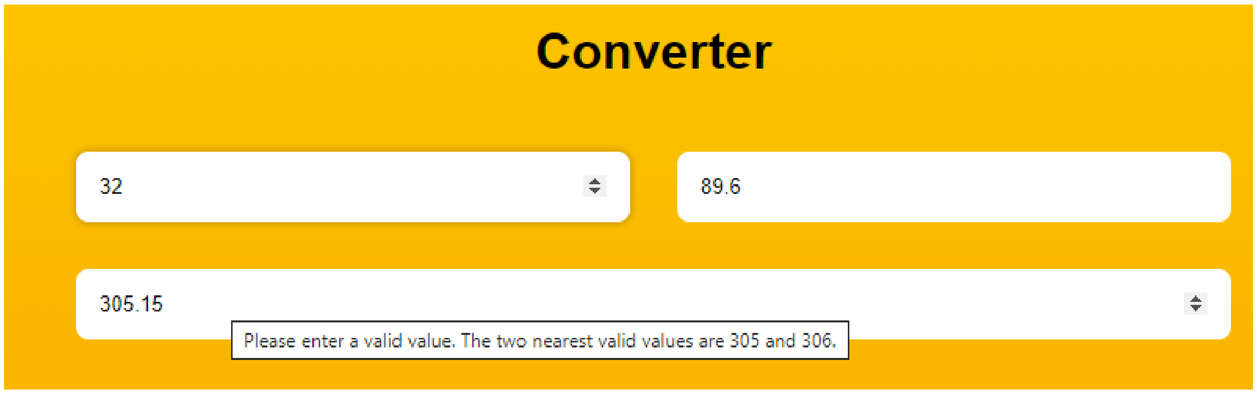
3.3.4 User story: Implementation of Fahrenheit:
Points 2. Status: Completed. Description: As a user, the Fahrenheit has to be displayed after celcius with up and down arrow adjustments. The box for Fahrenheit has to have the up and down arrow for adjustment of temperature dynamically without having to input it manually.



Please enter a valid value. The two nearest valid values are 89 and 90.

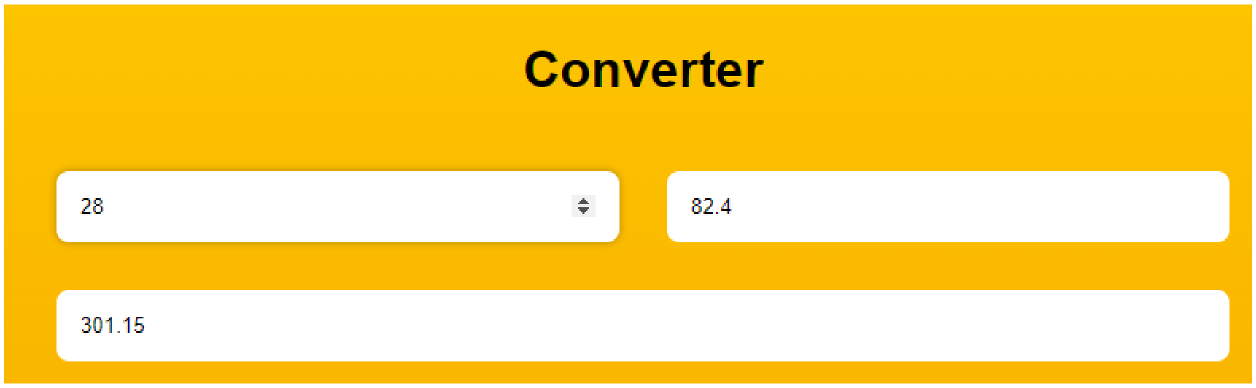
If the value is not rounded, the display would show as above where it would request to round off the values for correct calculation.

3.3.5 User story: Implementation of Kelvin:
Points: 2. Status: Completed. Description: As a user, the Kelvin has to be displayed after celsius and Fahrenheit with up and down arrow adjustments. Kelvin since it is a base, it has to be displayed below the celsius and fahrenheit space with the same adjustment arrows.



Please enter a valid value. The two nearest valid values are 305 and 306.

3.3.6 User story: Display the appropriate temperature:
Points: 3. Status: Completed. Description: As the user enters the temperature in any of the spaces, other spaces should be populated with the correct converted temperature.



As a user, the conversion has to be accurate for using it. We have to make sure it's done right.

4 Issues:

Since this project was done on a small scale, the test cases were not written but the functioning of the application was checked appropriately after it got completed. Jasmine unit tests could be done before developing this code to improve the project.

5 Deployment:

As this is a small scale project, it does not require any FTP deployment. The HTML file used under each application could be used to launch the application in the local browser. All three applications's HTML file was made to run and tested for its proper functioning.