Project Final Report

“Pomodoro Task Manager”

1. Introduction

- This App is about Daily Tasks Time Management through implementation of Pomodoro techniques

2. Design and Implementation

The Application consist of two Main Widgets :

“To-Do List” Widget for building the Tasks Pipeline(TaskListAppComponent)

“Pomodoro Clock” Widget for Time Management(PomodoroComponent)

These Widgets(React Components itself) are rendered inside the HomeComponent

They receive application state and function references through props and calling

the functions on MainComponent in order to update the application’s state

HomeComponent and AboutComponent,corresponds to two application pages

Which have different routes:’/home’ and ‘/about’ routes ,this is implemented

through the usage of Router (imported from 'react-router-dom' module)inside MainComponent,where HomeComponent and AboutComponent are rendered as different Routes inside Switch Component.The Application have Header and Footer implemented

through HeaderComponent and FooterComponent wrapped around

HomeComponent and AboutComponent,and they are always rendered, regardless

Which route user chooses

A typical flow of my application in terms of user experience and code implementation:

User will add new tasks using To-Do List Widget, building the Tasks Pipeline.

Tasks added and deleted by calling the handleAddTask(),handleDeleteTask(),

handleDeleteTasks() functions , which updates the **tasks[ ]** array located in application's state

inside the MainComponent

Afterwards with a help of Pomodoro Clock Widget he will choose length of

Task execution('Pomodoro' timeframe) and length of his Break Session

incrementTimer(),decrementTimer(),incrementBreakSession(),decrementBreakSession ()

After pressing the 'Play' button he will start the execution of Tasks Pipeline

Timer functionality implemented inside startTimer() function ,where through calling of

setInterval() asynchronious function, timer_value is updated

through continuous calling of decrementTimer() each 1000 milliseconds

```
 this.player = setInterval( () =>{
 this.setState({timer_value: this.state.timer_value - 1})
 if(this.state.timer_value<0)
 {this.stopTimer();
   this.setState({break:false});
  document.getElementById("beep").play();
  this.sleep(1000);
  this.nextTask();//increment pointer to next task
  console.log(this.state.task_index);
  this.startTimer(); }}
, 1000);
```

The Completion of the Task will be marked by Alarm Sound,

By playing the audio file ,which is hosted in the WEB and it's path url

Is saved inside **<audio>** tag on MainComponent then the color of

Timer will change(changing the color state: this.setState({timer_color:'#3f68aa'});

) and application will start the Break Session(startBreak()). By the end of it

User will hear Alarm Sound again ,and will start the execution of the next Task in the Pipeline

```
nextTask=()=>{
  var len=this.state.tasks.length;
  var task_indx=(this.state.task_index+1)%len;
  this.setState({task_index:task_indx});
};
```

Application will loop through the Complete Task List and when will start over again

During each step of the Pipeline Workflow ,user can Pause the Task\Break Session

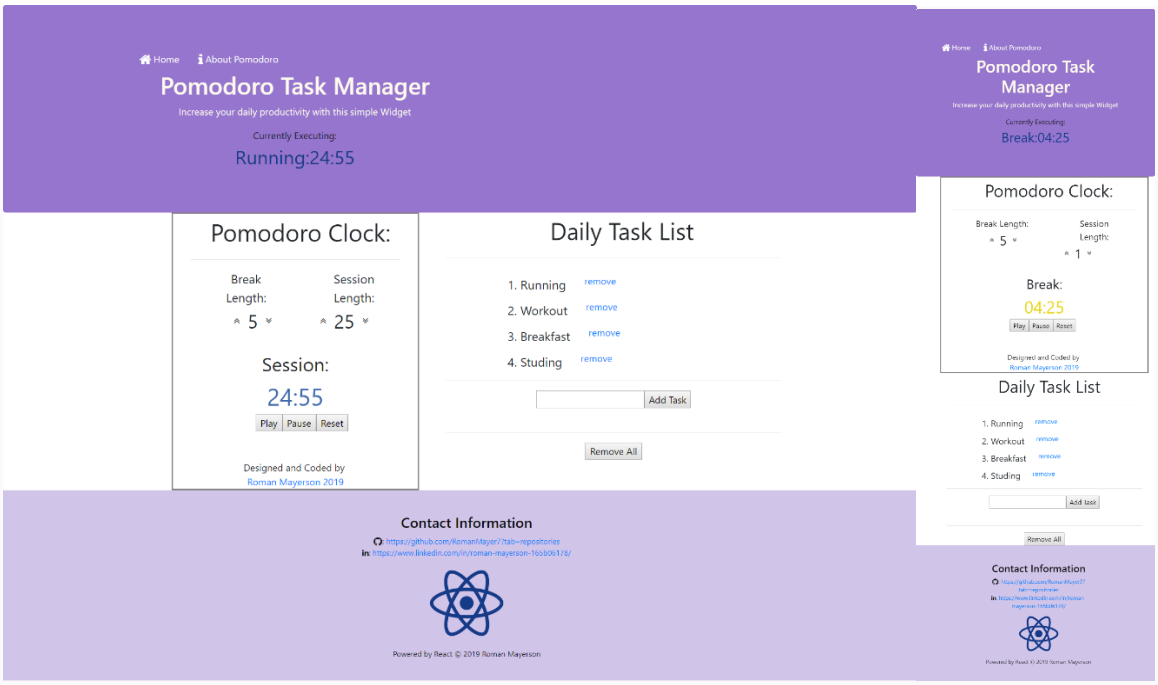
Or Reset the Timer and start from the Beginning ,by pressing on corresponding buttons inside

Timer Widget,which will call corresponding handler function on MainComponent and

Will make appropriate changes to the state of React Application,to maintain

The flow of Pomodoro Task Manager App Logic and Flow

- Include a few screen shots of your website in the report



3. Conclusions

- The results I've obtained from working on project:

Got Experience in Implementation and Deployment of React Application + Writing Documentation

.

4. References

- UI Design Wireframe: https://wireframe.cc/HbLG30
- Link to Pomodoro Task Manager: https://romanmayer7.github.io/Pomodoro_Task_Manager

- https://en.wikipedia.org/wiki/Pomodoro_Technique
- https://francescocirillo.com/pages/pomodoro-technique

by Roman Meyerson 18/07/2019