**ABIYAANTRIX & SAPIENCE ACADEMY**
**INTERNSHIP + TRAINING**

SAPIence®
Academy & Management Solutions

*INTERNSHIP MINI PROJECT*

**On**

# "POMODORO CLOCK"

***Submitted in partial fulfillment of the requirements for the 7th semester of Degree of***

**BACHELOR OF ENGINEERING**
***IN***
**COMPUTER SCIENCE ENGINEERING**

***SUBMITTED BY***

**ABDULRAJAK MULLA (4GE14CS001)**
**MOHAMMED USMAN (4mo15CS031)**
**KUMARSWAMY MATHPATI (4GE15CS016)**
**AJAYKUMAR R (4GE15CS001)**
**KARTHIK GOWDA T S (4GE15CS013)**
**MADAN KUMAR K M (4CA15CS017)**
**AKSHAY A (4VV15IS012)**
**ARAVINDAN S (4MO15CS005)**

**UNDER THE GUIDANCE OF**

**Mrs. ANJANA SHASHI KIRAN**
**DIRECTOR**

**Mr. SRIKRSHNA S KASHYAP**
**INTERNSHIP TRAINER**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GOVERNMENT ENGINEERING COLLEGE**

**CHAMARAJANAGAR**

**NANJANGUD ROAD, BEDARPURA**

**2017-2018**

## CERTIFICATE

This is to certify that the mini(INTERNSHIP) project report entitled **" POMODORO CLOCK”** is submitted by **ABDULRAJAK MULLA AND GROUP 4** students of **GOVERNMENT ENGINEERING COLLEGE CHAMARAJANAGAR**, in partial fulfillment of **7th** semester, **B.E** (**Computer Science and Engineering) INTERNSHIP** as prescribed by the VTU for the academic year 2017-2018. The report work has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the Bachelor of Engineering.

| | |
|---|---|
| **Mr. SHRIKSRISHN** | **Mrs. ANJANA SHASHI KIRAN** |
| INTERNSHIP TRAINER | Professor, DIRECTOR |

# ACKNOWLEDGEMENT

We are happy to present this POMODORO CLOCK Mini Project after completing it successfully. This Project would not have been possible without the guidance, assistance and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and everyone who has helped us make this project a success.

We Sincerely thank to our respected **MS. ANJANA SHASHI KIRAN** for his guidance and support for this internship Mini Project.

We heartily thank to our **TRAINER, Mr. SRIKRSHNA, internship** for his constant encouragement.

We gracefully thank our Internal Guides **Ms. ANJANA SHASHI KIRAN, Mr. SRIKRSHNA** for their encouragement and advice throughout the course of the internship Mini Project

Lastly we thank our parents, siblings and friends for their encouragement and support given to us in order to finish this precious work.

ABDULRAJAK MULLA AND GROUP

# CONTENTS

| | |
|---|---|
| ACKNOWLEDGEMENT | I |
| CONTENTS | II |
| LIST OF FIGURES | III |

**References**

## LIST OF FIGURES

**Chapter-1**

# Introduction

1. Introduction Time is a priceless and scarce resource for software development projects [4]. It is especially true in agile software development. A brief review of the 12 agile principles behind the Agile Manifesto reveals that time is an important dimension of agile processes, symbolized by terms such as “early”, “frequently”, “couple of weeks”, “daily”, “regular intervals” in these principles [1]. Agile teams work with short time-boxed iterations and need to maintain a fast yet sustainable pace throughout the project lifespan [3]. When moving to a distributed setting, the time dimension is further complicated by issues such as time zones, geographical distance, and different cultures [2]. However, there is very little reported evidence of effective time management techniques applied in agile software development, especially in the context of distributed teams. The Pomodoro Technique is a time management tool that was originally intended to optimize personal work and study. More recently, it has been widely applied by Italian agile teams [9]. Awareness of this technique is growing among the wider, international agile community (two tutorials on the Pomodoro Technique have beengiven in Agile 2009 - the international conference). The technique is named after the usage of a common kitchen timer in the shape of a tomato (“pomodoro” in Italian, see Figure 1). The heart of the Pomodoro Technique is 25 minutes of focused, uninterrupted work on one task, then 5 minutes of rest. There are also rules to keep the integrity of pomodoro, and tactics to deal with internal and external interruptions. However, starting as a personal time management tool, how is it applied by an agile team, especially when the team is working in a distributed environment? There is no ready answer in spite of the increasing popularity of the Pomodoro Technique in the agile community. Based on this observation, the objective of our study is to provide a better understanding of the application of the Pomodoro Technique in agile teams, especially when they work in distributed contexts. To this end, we studied in-depth one agile team that has applied the Pomodoro Technique extensively. The team collaborates with other remote sites of the company where the Pomodoro Technique is not used. This allows us to reflect on the impact of the Pomodoro Technique (and the lack of it) in a distributed context. The remaining part of the chapter is organized as follows. In the next section we review a set of time-related issues and argue the importance of time management in software development in general and agile software development in a distributed context in particular. It is followed by an introduction of the Pomodoro Technique. Then the experience of Sourcesense Milan Team using the Pomodoro Technique is presented. We analyse their experience and provide useful guidelines for implementing the Pomodoro Technique in the following section. The chapter ends with a conclusion section that highlights the contribution of our study and points out future studies

## 2.Time is An Enemy?

Time occupies a crucial place in software development projects. It is one of the most important factors dominating software development processes [10]. And just like many people have experienced, to various degrees, the anxiety associated with the passage of time, many software development projects have suffered from a set of time-related issues. Back in 1975, Brooks claimed that "more software projects havegone awry for lack of calendar time than for all other causes combined" [4]. In [10] a set of interlinked time-related problems in software development processes is summarised, including bottlenecks, which occur when one or more functions in the development process are dependent upon the output of another function within the process, resulting in developers having nothing to work on in the meantime; schedule problems, both construction of a feasible project schedule and to meet the schedule that has been set; difficulty in time estimation of large module/class/task; time pressure, which happens typically towards the end of a development process when the development team cannot meet the project schedule either because of poor time estimations or bottlenecks; and late delivery, which occurs as a result of inappropriate project planning, usually due to poor estimations. These inter-related problems, in essence, are all subjects of time management, one of the major knowledge areas of project management [13]. This area involves decomposition of project work into manageable tasks, estimation of task durations, scheduling of tasks, and controlling and monitoring the execution of tasks. Effective time management is crucial for addressing the time-related problems and leading to the success of software development projects. However, [10] argue that in early software engineering projects, when waterfall versions started to emerge, time issues were essentially neglected. As we proceed along the software engineering timeline, time issues receive increasingly more attention and their importance in software development processes is increasingly recognized, as evidenced in Team Software Process (TSP) [12], Rapid Application Development (RAD), and recently in agile software development. Time plays a more crucial role in agile software development than in conventional waterfall-like software development processes. This is demonstrated by a review of the 12 agile principles [1] from a time perspective. A review of these principles shows that 50% of them have an emphasis on time:  Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.   Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.   Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.   Business people and developers must work together daily throughout the project.   Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.   At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. These principles pose new time-related challenges to agile teams, apart from the previously discussed time issues. Agile teams need to work at a fast yet sustainable pace. The risk in having sole focus on velocity of development is the reduction of enthusiasm among team members. This may then have an impact on the sustainability of an agile team's daily work. It is important to achieve this equilibrium in agile software development, but it is underestimated in practice [9]. As a consequence, time management in agile software development not only means effective planning andmonitoring of the work to be performed, but also should help agile teams to create and maintain a fast yet sustainable pace. These two aspects of time management in agile software development can be further complicated as software development moves into global, distributed settings, which is an approach adopted by many current software projects. Issues such as time zones [6], geographical distance [7], and different cultures [14] will affect the previously discussed time-related issues and the effectiveness of time management techniques employed by agile teams. To better understand the impacts of distributed context, it is useful to consider different distributed team configurations (see [8] for different kinds of team configurations). Each different team structure presents different benefits to the work being undertaken, and would impact the time management technique used by agile teams in different ways. In spite of the importance of time and time management in agile software development, however, there is a paucity of both time management techniques and studies on the application of these techniques in agile software development, let alone in a distributed agile context. The Pomodoro Technique is one promising time management technique and is increasingly popular in the agile community. A growing number of agile teams use the pomodoro technique within their agile development processes [9].

# Chapter-3

# THE POMODORO TECHNIQUE

The goal of the Pomodoro Technique is to encourage consciousness, concentration, and clarity of thought through effective time management. A ‘pomodoro’ is 25 minutes of focused, uninterrupted work on one task then 5 minutes of complete rest. The inventor claims that, based on scientific proof, “20- to 45-minute time intervals can maximize our attention and mental activity, if followed by a short break” [5]. The 5-minute break aims to support team members in establishing and maintaining an optimal attention curve while engaged in project activities. In order to increase the impact of this effect, following every four consecutive pomodoros a longer pause of 15 minutes is recommended. The technique can improve the productivity of an individual. Improvements in productivity are achieved through increased motivation and the technique has also proved effective in supporting the management of complex situations. These benefits can be achieved through the following two inter-related aspects of the Pomodoro Technique: time-boxing and duration estimations. (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

The various methods used in this project are as follows:-

## 3.1 Pomodoro as Time-box

One of the primary inspirations behind the Pomodoro Technique is time-boxing [5]. Time-boxing suggests that, once a series of activities has been assigned to a given time interval, the delivery date for these activities should never change. If necessary, the unfinished activities can be reassigned to the following time interval. Corresponding to the idea of time-boxing, to maximize participant concentration, every time-box (or pomodoro), needs to be protected. “Protecting pomodoro” leads to fewer interruptions. There are two kinds of interruptions that need to be addressed: The dependability of a computer system is a property of the system that equates to its trustworthiness. T One of the primary inspirations behind the Pomodoro Technique is time-boxing [5]. Time-boxing suggests that, once a series of activities has been assigned to a given time interval, the delivery date for these activities should never change. If necessary, the unfinished activities can be reassigned to the following time interval. Corresponding to the idea of time-boxing, to maximize participant concentration, every time-box (or pomodoro), needs to be protected. “Protecting pomodoro” leads to fewer interruptions. There are two kinds of interruptions that need to be addressed: rustworthiness essentially means the degree of user confidence that the system will operate as they expect and that the system will not ‘fail’ in normal use.

Internal: These interruptions are triggered by the participant, e.g. “I should check email”, or “I need to get a coffee”; External: Triggered by other entities, e.g. a phone call or a request from a colleague. In order to handle these interruptions effectively, an “indivisible rule” needs to be enacted. A pomodoro represents twenty-five minutes of pure work that cannot be split up. There is no such a thing as a half or a quarter of a pomodoro. If a Pomodoro is interrupted definitively, i.e. the interruption is not deflected, then the pomodoro is considered to have never commenced – it is made void. Used as a time-boxing tool, the Pomodoro Technique can help enhance focus and concentration on work by cutting down on interruptions. Consequently, it can alleviate anxiety linked to the passage of time and reduce both time-wasting and overtime. A sustainable working pace can be obtained through the alternation of work and rest and the combination of short breaks and long pauses. Pomodoro Technique can help enhance focus and concentration on work by cutting down on interruptions. Consequently, it can alleviate anxiety linked to the passage of time and reduce both time-wasting and overtime. A sustainable working pace can be obtained through the alternation of work and rest and the combination of short breaks and long pauses.

## 3.2 SOUCE CODE

```
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">
        <meta name="description" content="A single page application featuring a fully
functional pomodoro timer">
        <meta name="viewport" content="width=device-width, initial-scale=1, user-
scalable=yes">
        <title>Pomodoro Timer | Jason Durant</title>
        <link rel="stylesheet" type="text/css" href="css/style.min.css">
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.5.0/css/font-awesome.min.css">
</head>
<body>
        <header class="content-center">
                <h1>Pomodoro Productivity Timer</h1>
        </header>
        <section class="container content-center">
                <div class="half-width-m breaks-container">
                                <div id="break-container" class="half-width-m break-
container">
                                        <p>Break Min</p>
                                        <div>
                                                <span class="subtract-time">-</span>
                                                <span id="break-time">7</span>
                                                <span class="add-time">+</span>
                                        </div>
                                </div><!--remove space
                                --><div id="work-container" class="half-width-m work-
container">
                                        <p>Work Min</p>
                                        <div>
                                                <span class="subtract-time">-</span>
                                                <span id="work-time">30</span>
                                                <span class="add-time">+</span>
                                        </div>
                                </div>
                </div>
        </section>
        <section class="content-center">
                <div class="timer-container">
                        <div class="timer-inner-circle">
                                        <h2 id="timer-title">Pomodoro Session</h2>
                                        <div id="timer-clock"  class="timer-clock">25</div>
                                        <button id="engage-button"
class="start">Start</button>
                                        <button id="reset-button" class="reset">Reset</button>
```

```
					<!-- absolute -->
					<div class="timer-filler timer-filler-work">
						<div class="gradient-animation"></div>
					</div>
				</div>
			</div>
			<audio id="timer-bell" src="workbell.mp3" type="audio/mpeg">
			</audio>
		</section>
		<footer class="container content-center">
			<ul>
				<li><a href="https://www.linkedin.com/in/durantjason"><i class="fa
fa-linkedin-square" title="LinkedIn profile link"></i>
LinkedIn</a></li>
				<li><a href="https://twitter.com/" title="Twitter profile link"><i
class="fa fa-twitter-square"></i>Twitter</a></li>
				<li><a href="https://github.com" title="Github profile link"><i
class="fa fa-github"></i>Github</a></li>
			</ul>
		</footer>
		<script type="text/javascript"
src="bower_components/jquery/dist/jquery.min.js"></script>
		<script type="text/javascript" src="js/main.min.js"></script>
</body>
</html>
```

**3.3 Pomodoro as Unit of Effort**

To master and improve the use of the Pomodoro Technique, an underlying daily process is suggested, which consists of five stages: Planning: to decide the activities to do in the day; Tracking: to gather raw data on the effort expended and other metrics of interest; Recording: to compile an archive of daily observations; Processing: to transform raw data into information; and Visualizing: to present the information in a format that facilitates understanding and clarifies paths to improvement. In each stage, the pomodoro plays the role of unit of estimation. Two rules apply:

1) if a task lasts more than 5-7 pomodoros, break it down. Complex activities should be divided into several activities; and

2) if it lasts less than one pomodoro, add it up. Simple tasks can be combined. Used as an effort estimation tool, the Pomodoro Technique can support the refinement of an effort estimation process through the use of continuous reflection of team activities (more detailed instructions of how to apply the Pomodoro Technique as a personal time management technique can be found in [5].

The Pomodoro Technique was invented initially for individual work, as a personal time management tool. But it has been developed and refined in the context of teamwork by the inventor and advocates of the technique over the time. [9] report the technique as a team time management tool used by several XP teams. They claim that the Pomodoro Technique is an unstressful - as well as efficient - way to help teams find their “natural” rhythm in daily work. Their study is a good starting point and provides a broad picture for investigating how the Pomodoro Technique can be applied in agile teams. Our study intends to go into more depth to understand the benefits, issues and concerns of using the Pomodoro Technique in an agile team within a distributed context.

# Chapter-4

## Snapshots

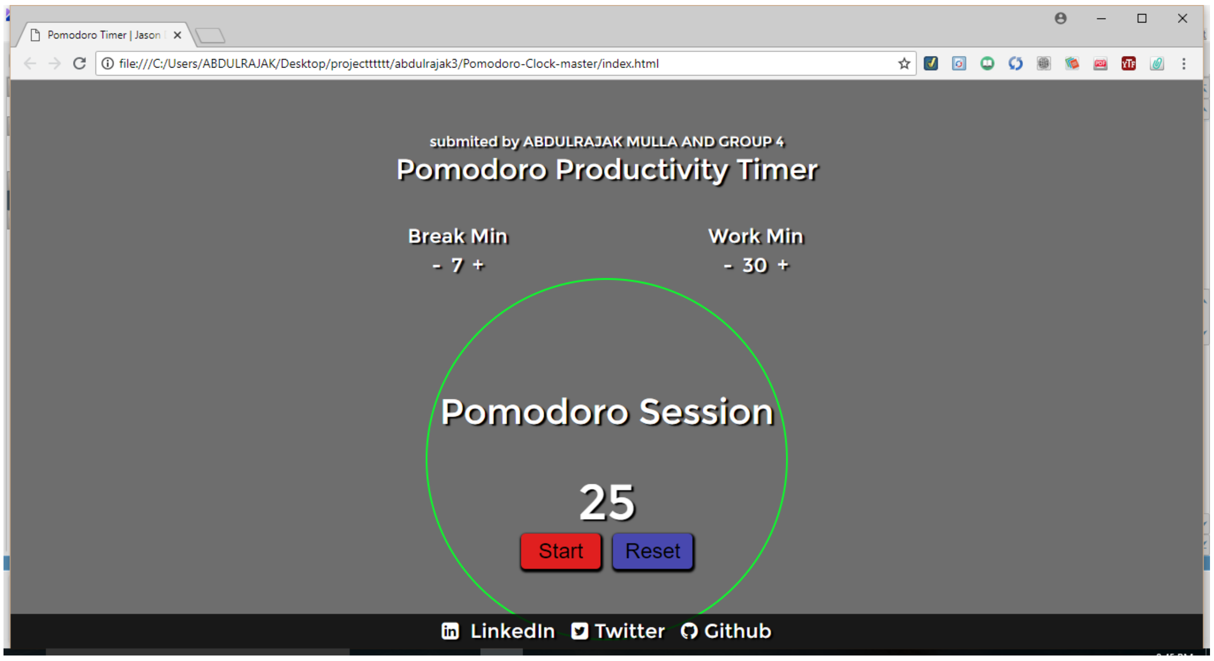

Pomodoro Timer | Jason
file:///C:/Users/ABDULRAJAK/Desktop/projectttttt/abdulrajak3/Pomodoro-Clock-master/index.html
submited by ABDULRAJAK MULLA AND GROUP 4
Pomodoro Productivity Timer
Break Min
- 7 +
Work Min
- 30 +
Pomodoro Session
25
Start
Reset
LinkedIn
Twitter
Github

**Figure 4.1: Start up window time**

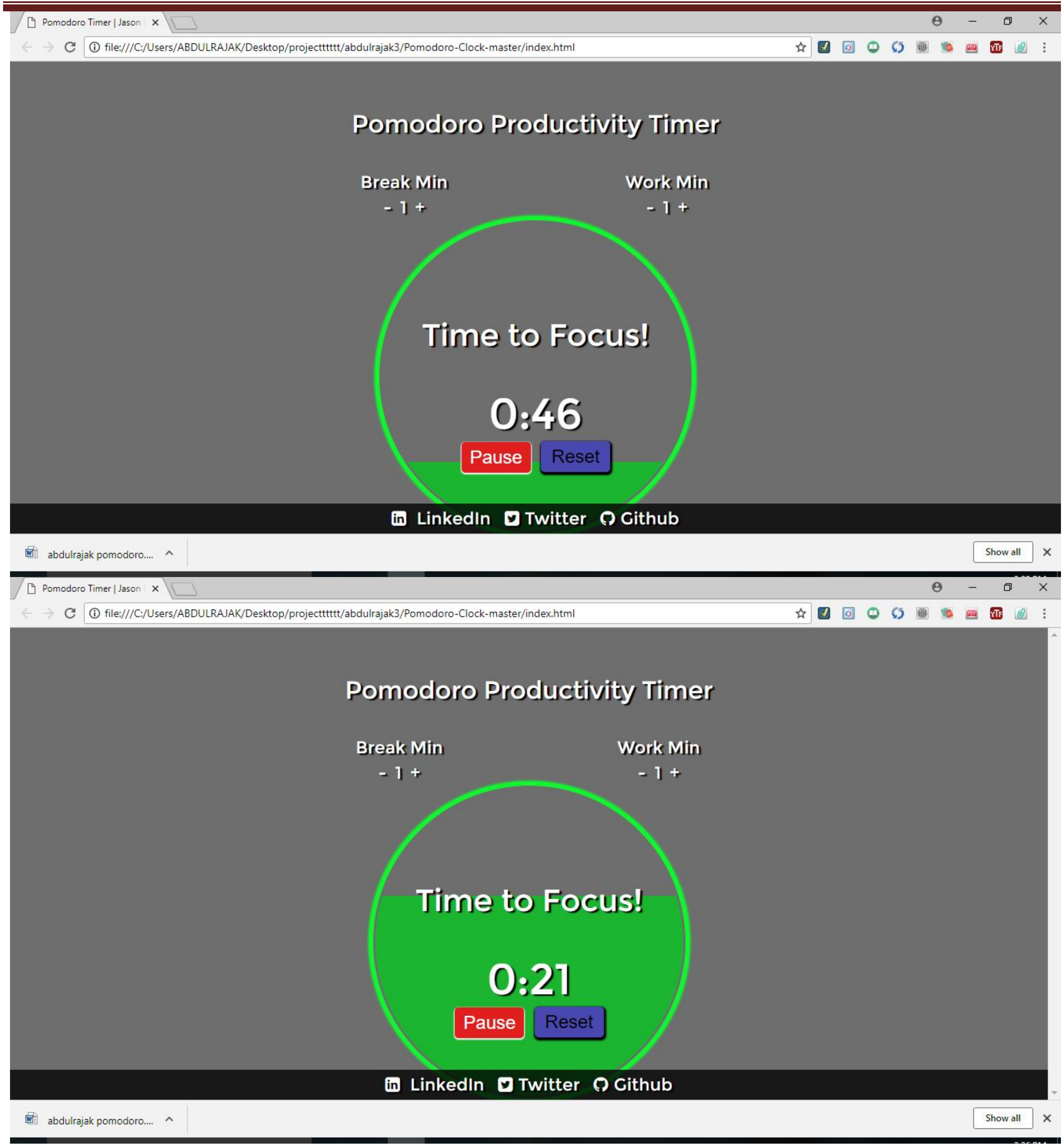
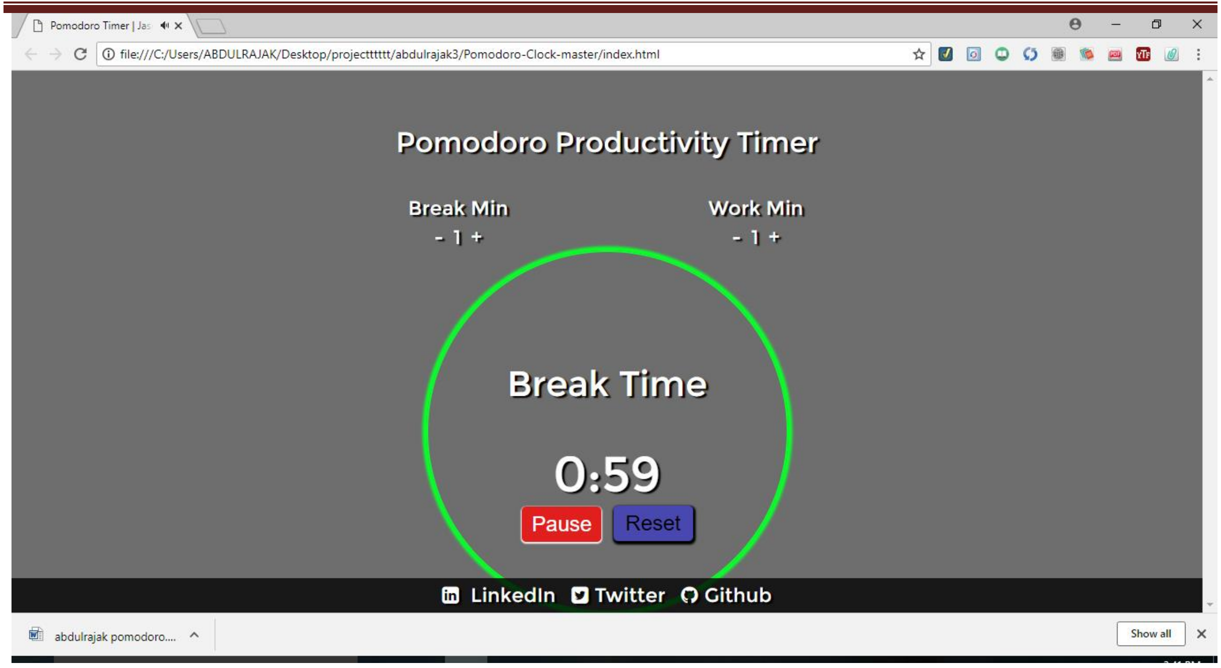


**Figure 4.2: Job tim to focus**



h ddddyhgrgh

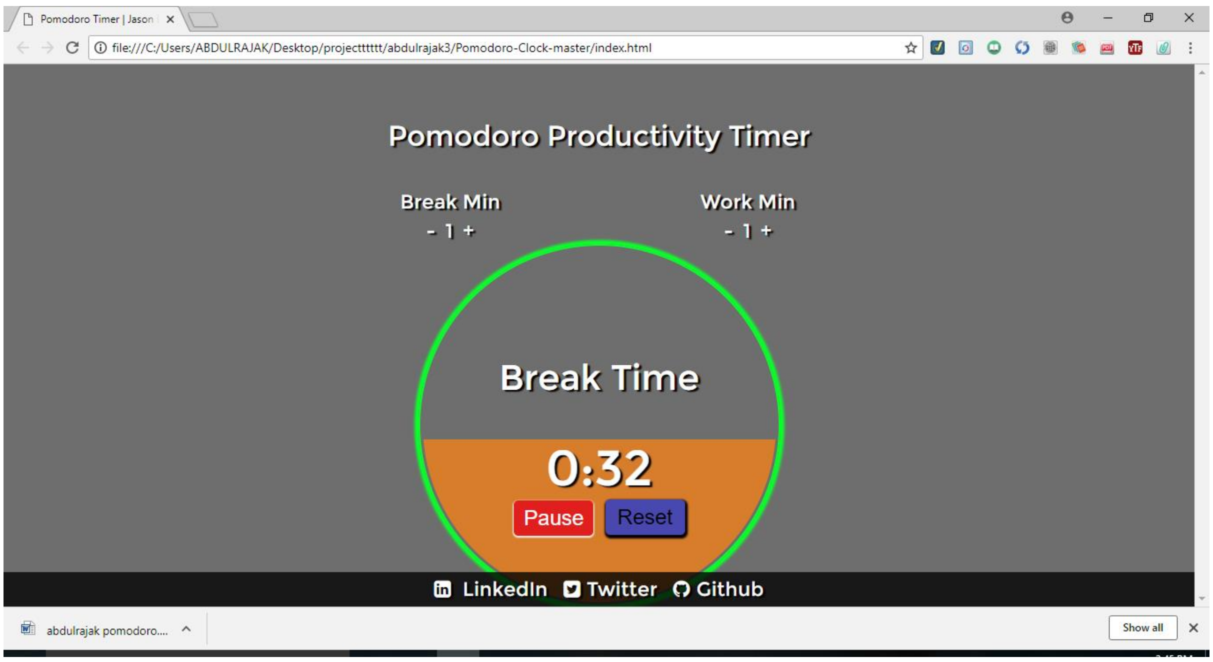


**Figure 4.2: Job time to focus**

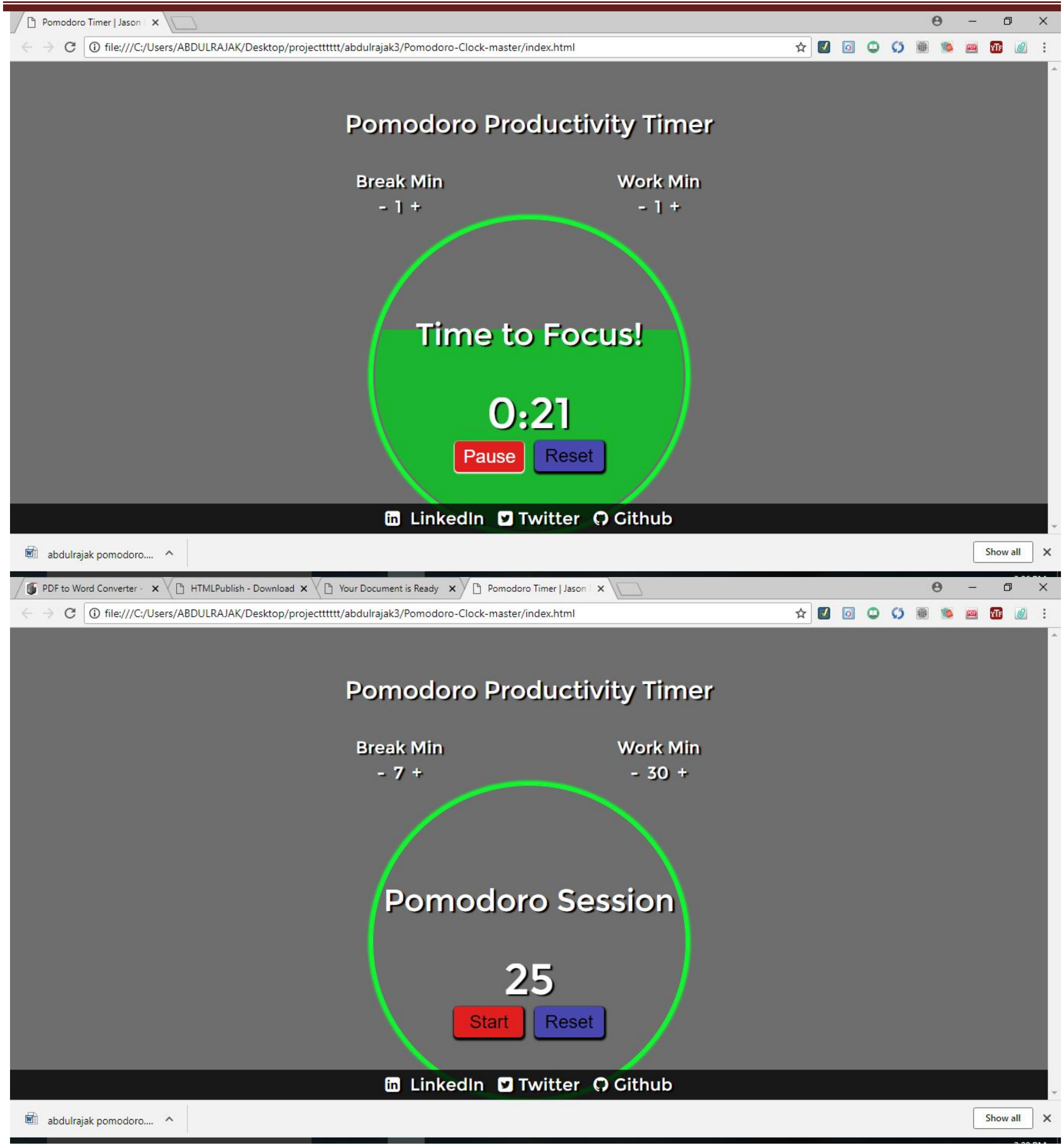
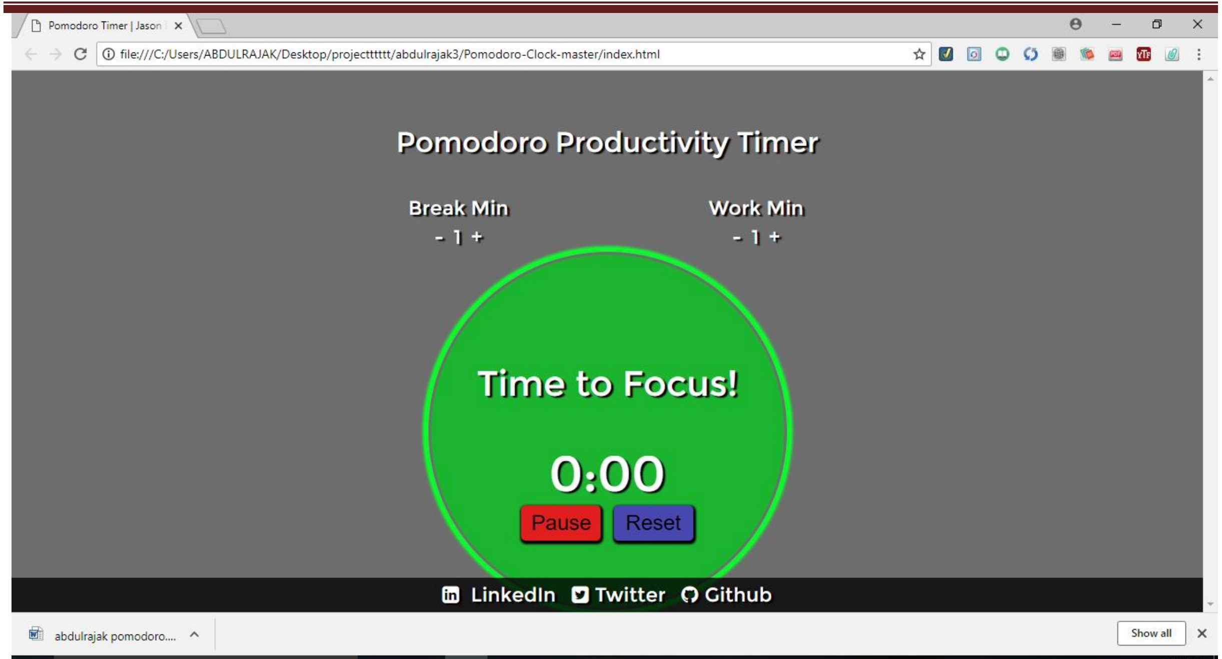


**Figure 4.3: pomodoro session on alarm**



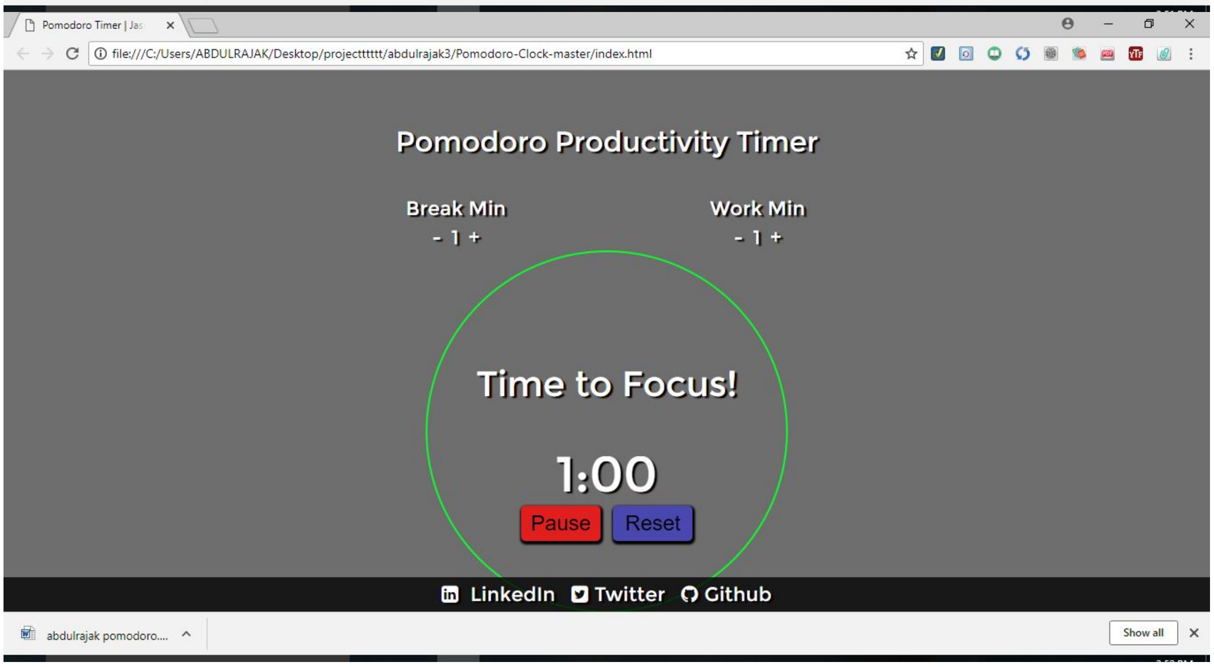


**Figure 6.4: time to focus alarm**

**Chapter 5**

# Conclusion & Future enhancement

In this chapter we presented a case study of the application of the Pomodoro Technique in an agile software development team. The effects of the technique on the team were analysed through two angles: pomodoro as timebox and pomodoro as unit of effort.

We argued the benefits and potential issues of using one pomodoro for the whole team, team breaks, estimating with pomodoro (real time) rather than abstract points. We also explored the scenarios where the Pomodoro Technique should not be applied. If and how the technique can be applied in a distributed setting was also examined. Even though the Pomodoro technique is not used in a truly distributed manner in the case we studied (which is a major limitation of our study),

we believe that the technique itself is a welcome addition to the agile development toolkit. We would hope that the lessons learnt from this case study could be easily adapted into distributed settings. Our study contributes to the body of knowledge of an under-developed theme within agile research: effective time management in fast-paced agile software development. The practical implication of our study is a better understanding of time management in agile software development, and several concrete suggestions of how