CSE 341 Project 2 Proposal

# General Info

Michael Norton, Carlos N Reina, Aaron Rooks, Leonardo Souza

Time Tracker

Contents:

Contents

[General Info 1](#_heading=h.gjdgxs)

[Application Info 2](#_heading=h.30j0zll)

[What will the app be? 2](#_heading=h.1fob9te)

[What will the app do? 2](#_heading=h.3znysh7)

[What kind of data layer will you incorporate? 2](#_heading=h.2et92p0)

[Will you choose to use a frontend JavaScript framework? If so, which? If not, why not? 2](#_heading=h.tyjcwt)

[How will your app utilize a login system? 2](#_heading=h.3dy6vkm)

[What views will change based on roles, or logged in status? 2](#_heading=h.1t3h5sf)

[What pieces of data in your app will need to be secured? How will you demonstrate web security principles in the development of this app? 2](#_heading=h.4d34og8)

[What file structure and program architecture will you use for this project? Why? 2](#_heading=h.2s8eyo1)

[What are potential stretch challenges that you could implement to get 100%? 2](#_heading=h.17dp8vu)

[Potential Risks and Risk Mitigation Techniques 3](#_heading=h.3rdcrjn)

[What are the risks involved with you being able to finish this project in a timely manner? 3](#_heading=h.26in1rg)

[How will you mitigate or overcome these risks? 3](#_heading=h.lnxbz9)

[Project Scheduling Timeline 3](#_heading=h.35nkun2)

# Application Info

## What will the app be?

Our application will be a daily productivity application helping users improve their studies and skills using a time tracker and link organizer.

## What will the app do?

Users will be able to create an account. With their accounts users will be able to create a playground with a name of a subject or skill the user plans to work on.

**Time Tracker**

Once a playground has been created users can enter the playground and click a button to clock-in. If a user clicks the clock-in button when a clock-in is still active a notification will let them know they are currently clocked-in and no action will be performed.

An optional timer will be available to remind users to take a break and not over work. Users may also have the option to use a stopwatch that will give them an idea of how long they have been working.

Once a user is ready to stop working on their subject or skill they click the clock-out button. Likewise if no clock-in is active and a user clicks clock-out they will be notified that no clock-in session is active.

For each clock-in and clock-out pair, users will be able to review their daily, weekly, monthly, quarterly and yearly progress based on time spent on their activities. They will also be able to view this in a weekly calendar view and graphics.

If a user forgets to clock-in or clock-in they will have an option to manually add or edit a time-stamp. They will also be able to delete time stamps if they make a punch by accident.

**Link Organizer**

Inside their playground the user will be able to add links they often access.

**Keyboard Shortcuts**

Keyboard shortcuts will be available for users to clock-in and out, make time edits, create new links and navigate links.

\*If the app is to have a more professional feel then dashboard may be a better name then playground. Time tracking though already is work in and of itself so the term playground was chosen to help gamify the work of being organized.

**Team Playgrounds**

Users will also be able to create a playground to work together with teammates. There users will be able to see time worked by other users on the playground but only be able to add, edit and delete their own time stamps. The team can also be able to create links and manage links. This will be up to the playground admin to decide whether these links are edited by just the admin, specific users or anyone on the team. Keyboard shortcuts will work the same for team playgrounds as they do for individual playgrounds.

## What kind of data layer will you incorporate?

We will use MongoDB.

## Will you choose to use a frontend JavaScript framework? If so, which? If not, why not?

We will use EJS, to incorporate JavaScript into our frontend.

## How will your app utilize a login system?

We will use sessions to allow our users to login and be recognized on the server side.

## What views will change based on roles, or logged in status?

The main view will have an additional link in it for going to the activities view when a user is logged in. Users will only be able to see their own activities/time logs in the activity view.

## What pieces of data in your app will need to be secured? How will you demonstrate web security principles in the development of this app?

Our authentication routes will need to all be validated. Our database will secure login info using hashed passwords in our database. We will use Csrf tokens to ensure that our sessions are protected. We will check the authentication status of requests before allowing them to access routes that only authenticated users should access.

## What file structure and program architecture will you use for this project? Why?

We will use the MVC pattern for this project.

## What are potential stretch challenges that you could implement to get 100%?

Importing files with activities and time stamps, reading and automatically logging these activites/times. Team Playgrounds.

# Potential Risks and Risk Mitigation Techniques

## What are the risks involved with you being able to finish this project in a timely manner?

Individual team members time constraints. Implementing the team playgrounds feature may be challenging, and use new principles that we have to learn on our own.

## How will you mitigate or overcome these risks?

Communication is key to mitigating time constraints, so that other team members may pick up the slack where others cannot.

# Project Scheduling Timeline

Plan out when you will complete each phase of development listed in the [Project 2 High-level Tasks webpage](https://byui-cse.github.io/cse341-course/projectStuff/prj2/prj2-breakdown.html) within the allotted time to complete the project.

| Week 7 Tasks |  |
| --- | --- |
| Week 8 Tasks |  |
| Week 9 Tasks | Initial Setup/Views |
| Week 10 Tasks | Authentication/Base Functionality |
| Week 11 Tasks | Error Handling/Security |
| Week 12 Tasks | Debugging/Clean Up/Stretch Goals |