

### Criterion B: Record of Tasks

Task number	Planned action	Planned outcome	Time estimated	Target completion date	Criterion
1	Brainstorm IA ideas	Created a document filled with possible ideas and decided to create a timeblocking app—an app that I wish existed and would love to use myself.	1 hour	October 8, 2021	A
2	Brainstormed features	Created a document filled with features that I'd need to implement in my app to be eventually incorporated	2 hours	October 10, 2021	A
3	Brainstorm solution mastery	Create a list of the techniques that I would need to incorporate into my app to satisfy the IB criteria, which influenced the types of features I decided to add to my app	2 hours	October 12, 2021	A
4	Researching the problem	Understanding more about timeblocking and how I should structure my app to make creating timeblockers easier	2 hours	October 13, 2021	A
5	Creating a detailed plan for the application	Have an outline of the functionality completed	3 hours	October 16, 2021	A
6	Initial discussion with my advisor—my father	To have ideas approved by my advisor and to have received some feedback on my idea.	0.5 hours	October 22, 2021	A

7	Consider the technologies I plan to use in my program	Decided to build a website that would be easily accessible from any device with a browser and Internet which would also sync data across multiple devices using a server	5 hours	October 25, 2021	A
8	Create testing plan	Create a list of tasks that my program needs to be able to accomplish in order for me to consider it finished.	2 hours	November 1, 2021	A
9	Find software for creating the flowchart	Searched and experimented with many design apps, considered using multiple free design apps, from GIMP (an open-source image editor), Inkscape, Gravit, but ended up using FigJam because for me it was the easiest to use.	4 hours	November 3, 2021	B
10	Create flowchart with FigJam	Used FigJam to create a basic flowchart of the user's actions on the frontend.	1 hour	October 20, 2021	B
11	Sketch GUI on paper	Have a basic GUI drawn on paper to get a better idea of how I want my UI to look (I plan to polish it further with using an actual design app)	5 hours	October 18, 2021	B
12	Find software for creating the design pages	Experimented with apps (the same apps I experimented with when finding a flowchart creator), eventually settled on Figma because it was popular and easy to use.	4 hours	October 20, 2021	B

13	Creating an initial UI design in Figma	Create a timeblock page in Figma containing month, week, and day views, as well as login and registration pages.	20 hours	November 15, 2021	B
14	Designing the class hierarchy of my program.	Creating a hierarchy of classes involving inheritance, composition, and polymorphism that represents the relationships between data in my code.	2 hours	November 20, 2021	B
15	Set up Vite	Installed Vite, a front-end build tool that allows local development of a website. It also has the extremely useful feature of automatically updating the website when changes are made to the code without needing to refresh the page.	2 hours	November 20, 2021	C
16	Set up TailwindCSS	Have TailwindCSS integrated into my project that allows me to use utility classes in my HTML code to make it easier to style content in my website. In addition, have a popular TailwindCSS UI library (daisyUI) installed that allows me to easily embed customizable, accessible, and battle-tested UI components like buttons and checkboxes.	4 hours	December 17, 2021	C

17	Set up TypeScript	Have TypeScript (a strongly-typed superset of JavaScript) set up so that common bugs like passing the wrong type to functions or typos in object keys can be caught at compile-time instead of runtime. Also improves IDE IntelliSense when coding.	4 hours	December 19, 2021	C
18	Setting up page routing on my website	Have a Vue Router configuration object containing routes that point to the various pages in my program. Adding Vue Router links to the code that allow users to navigate to different pages through button clicks in my website. Also configure GitHub pages to redirect user to proper 404 page when the user's link doesn't exist.	2 hours	January 16, 2022	C
19	Set up state management on my website	There will be state that many different components need to access, and the best practice is to use a global data store that can be accessed by all components. I decided to use Pinia, a type-safe store library for Vue since it has great TypeScript and IntelliSense support.	2 hours	January 16, 2022	C

20	Learn Vue 3.2's ref sugar syntax	Integrate Vue 3.2's ref sugar syntax into the project and use it for reactive variables to reduce boilerplate needed to write Vue template files.	2 hours	December 14, 2021	C
21	Create a basic landing page for ib.time-blocker.io	Added a large title, a header (Login, Register), and a call-to-action button.	2 hours	December 20, 2021	C
22	Hosting the website on GitHub pages and integrating GitHub Actions	Have the website publicly accessible on a GitHub domain ( <a href="https://&lt;username&gt;.github.io/time-blocker">https://&lt;username&gt;.github.io/time-blocker</a> ) and have 404 redirects work properly, and have a GitHub Action run every time the code is pushed to the main branch.	4 hours	December 22, 2021	C
23	Linking the website to a public domain for anybody to access	Have the website accessible on a public domain ( <a href="https://ib.time-blocker.io">https://ib.time-blocker.io</a> )	2 hours	December 24, 2021	C
24	Creating a Dockerfile and integrating Docker for easier local development	Having a docker-compose.yaml file in the root of the project to automatically build and start a PostgreSQL container with the database when 'docker compose up' is run on a local machine with Docker installed.	3 hours	December 31, 2021	C
25	Setting up Prisma on the backend	Having Prisma installed and integrated into the backend and having the ORM models defined in the schema.prisma file.	4 hours	January 4, 2022	C

26	Implementing authentication in the frontend and backend	Creating a functional registration and login page with proper web security. Implementing headers-based authentication in the backend.	15 hours	January 10, 2022	C
27	Fix weird JSX bug that's preventing Vue component type checking from working properly	Isolated the bug to the result of a package I'm using (FullCalendar) that imports its own types for JSX that is interfering with the Volar configs. Fixed the problem by creating a stub file for overriding FullCalendar's types.	5 hours	January 14, 2022	C
28	Learning and integrating the TailwindCSS library DaisyUI into my program	Have the UI elements of my program use battle-tested and accessible components from a popular TailwindCSS component library called DaisyUI.	4 hours	January 18, 2022	C
29	Creating a landing page for my website	Have a landing page that is presented to the user when they visit the home page ( <a href="https://ib.time-blocker.io">https://ib.time-blocker.io</a> ) that includes a call-to-action and buttons to create an account or log into an existing account.	3 hours	January 20, 2022	C
30	Implementing drag and drop API for dragging tasks and task blocks	Allow user to easily move tasks around by simply dragging and dropping them into the schedule.	5 hours	January 23, 2022	C

31	Fix bug with Volar type checking	Isolated the issue to using a union type signature in Vue's defineEmits function. Replaced the union with two separate function signatures to sidestep the bug. Created a minimal reproduction project and reported the bug using GitHub issues in the Volar GitHub repository.	2 hours	January 24, 2022	C
32			2 hours	January 24, 2022	C
33	Implement ability to resize task blocks on the schedule	Allow the user to drag up/down on the edges of the task blocks to easily resize their task. Had to add validation to prevent user from dragging it too high/low that it would make the task block "0 minutes long"	3 hours	January 24, 2022	C
34	Add a timeblock calendar that allows users to create a timeblock for each day	Use the FullCalendar library. Had to find out how to dynamically render a Vue component since I needed to return a DOM element for the dayCellContent property in the FullCalendar component, and the Vue component doesn't return HTML by default. Solved the problem by mounting the component into a dynamically created <div> element.	3 hours	January 25, 2022	C

35	Change the scoping of tasks	Before, tasks were scoped to individual timeblocks. However, it's more likely that users want to reuse their tasks across timeblocks. Thus, I need to scope the tasks to an account and not each individual timeblock.	1 hour	January 25, 2022	C
36	Figure out why Vue's global errorHandler isn't working with async callbacks	Learned that Vue's errorHandler only works with errors that are thrown in the 'setup()' function and don't work with errors that are thrown in callbacks (including async callbacks). Fixed by using window.addEventListener('error') and window.addEventListener('unhandledRejection')	2 hours	February 28, 2021	C
37	Finish implementing the fixed time tasks & task blocks and the fixed weekly time tasks & task blocks		2 hours	March 1, 2021	C
38	Learning a library called tRPC for creating type-safe API endpoints	Reading the documentation at trpc.io	3 hours	November 27, 2021	C



39	Learning web security and adding measures to avoid CSRF attacks and minimize the damage of XSS attacks	Learned about cookies, session tokens and session state, danger of XSS attacks. Browsed through the Amazon Cognito JavaScript library to find out how they store user tokens, and discovered they use the browser's localStorage. Thus, I decided to store session tokens for ib.timeblocker.io via the browser's localStorage.	20 hours	December 12, 2021	C
40	Set up Prisma	Have the prisma packages installed and create	1 hour	December 19, 2021	C
41	Set up tRPC	Installed the necessary libraries, set up the TypeScript configuration in tsconfig.json, installed the tRPC fastify adapter to integrate with Fastify, set up tRPC context so that all operations have access to the Fastify request and reply variables, as well as access to the Prisma client instance.	6 hours	December 19, 2021	C

42	Debug tRPC frontend issues	TypeScript throwing a compiler error in frontend saying that it can't find TypeScript definitions for fastify even though fastify is a back-end library. After some investigation, turns out this was because of how tRPC worked, as I needed to import types from the back-end. I fixed it by adding a manual entry in the frontend's tsconfig.json.	3 hours	December 19, 2021	C
43	Create registration and login routes	Implement the registration and login routers for tRPC and merge the routers sessions in the backend via session tokens.	2 hours	December 19, 2021	C
44	Create timeblock CRUD (Create, Read, Update, Delete) routes	Implement all the necessary routes along with their database implementations for creating new timeblocks, new timeblock tasks, and new timeblock task blocks. Also modified the prisma.schema to reflect these new models.	15 hours	December 21, 2021	C

45	Debug issue with creating new accounts where it gives the error message "token not found."	Fixed the bug with registration; I mistakenly thought the return value of createAccount was the sessionToken when in reality it was the accountId, so I ended up sending the accountId to the client and it was sending the server the accountId as if it was the session token, which lead to the "token not found" error.	1 hour	December 21, 2021	C
46	Hosting the backend server publicly on Heroku (a cloud application platform)	The server is public and linked to a Heroku domain and CORS (cross-origin resource sharing) is restricted to only accept requests from <a href="https://ib.timeblocker.io">https://ib.timeblocker.io</a>	4 hours	December 21, 2021	C
47	Debug pnpm issues with Heroku	Heroku doesn't natively support the package manager that I used for Timeblocker (pnpm), so I had to experiment with numerous workarounds to try and get it working with Heroku.	20 hours	December 25, 2021	C
48	Debug fastify issues with Heroku	The Heroku server kept failing because Fastify wasn't binding to the port; turns out there was two issues: one was I needed to explicitly specify the 0.0.0.0 address for Heroku, and the second issue was that I forgot to read the PORT variable from the environment.	2 hours	December 25, 2021	C