This is a purely hypothetical idea from some inspiration that I had, loosely based upon various task scheduling apps (Todoist, Microsoft To-Do) and Adapt.

TimeLearn

TimeLearn is an app that would enhance productivity, primarily amongst students in schools and universities, but also amongst anyone that wants to use the app, including those who are employed and those that are doing hobby projects. It would retrieve data about your tasks from various services (e.g Slack, Google Classroom) and also accept manual user input. It would arrange your tasks into realistic and achievable chunks, schedule them for you and encourage you to get working and meet your deadlines, by notifications and motivational quotes (from a database).

It will plot your tasks at the times when you're most productive based upon tasks that you've ticked off on the app, allowing for tasks that may take longer to not distort this productivity data.

If the user wants this to happen, the app would be able to retrieve background usage data on the device to track their usage of productivity apps to improve the algorithm for them, working out whether they're really working or whether they're just saying they're working. This would likely be unpopular due to privacy concerns and wouldn't be implemented.

Their information would stay within TimeLearn. The source code of this app would be publicly available on GitHub under the MIT licence.

Design

The app would look modern and be easy to use. It would have navigation buttons to see 'Tasks', 'Schedule' and 'Settings' at the bottom, which would have the symbols of a to-do list, a clock and a cog, respectively. The settings page would have app information for support purposes (version number, some information about the device - whether it's rooted/jailbroken and the OS version), allow connections for user input and enable a user to delete their account. Other forms of task adding will be done through 'Tasks'. The schedule can be adjusted in 'Schedule' by dragging and moving around the tasks, tapping the task and moving your finger to the right to lengthen the amount of time you spend on it and tapping the task and moving your finger to the left to lessen the amount of time. The app wouldn't be complicated to use. It'd have simple symbols, easy navigation and the error messages would be in plain English and understandable to anyone with an average reading age[1].

The app would be primarily in blue and white and have fairly rounded buttons. It'd support a dark theme and I hope to roll out additional themes and give the user the ability to change the design by programming a theme or by tweaking the theme using a simple interface. The acceptance of programmed themes would be dependent on the ability to ensure that malware that could infect the app can't be introduced, although the themes would be stored locally on the device.

At release, the app would only be available in English. As the app progresses, it could accept user translations to translate the app into a language they speak, which could be

[1] For the UK, sources put the average reading age between 9 and 11.

This is a purely hypothetical idea from some inspiration that I had, loosely based upon various task scheduling apps (Todoist, Microsoft To-Do) and Adapt.

voted on, approved and then written into the app. There would also be a suggestions interface to allow users to suggest features and a contact form to allow thm to report bugs.

When a user's task is scheduled, the app will notify them. The user will then confirm that they're getting started and update the app on their progress on the task or sub-tasks at the end of their session.

They'll be able to go back into the app and listen to calm, productive music, which would be instrumental to lessen focus on lyrics if they want to do so. I plan to offer support for the popular and sensible Pomodoro technique in the app, which consists of 25 minutes of work, followed by a 5-minute break and then another 25-minute burst.

Target demographic

It's my ambition that despite the slightly misleading name, the app and the website would be available to everyone in allowed markets globally. It wouldn't have a specific target audience in any shape or form.

Development and testing

This app would be tested using the three proper phases of development to ensure that it's robust and well built:

- Alpha stage - Would be carried out by a team of programmers and involve some scrutiny of the source code to check for errors, test data would be used to ensure that the program produces the expected response at all times.
- Beta stage - A group of interested users are given access to the app. They'd use it in a normal working situation and feedback their findings, including bugs and suggestions for improvement.

Corrective and adaptive maintenance would be done regularly.

The app would be built on Visual Studio Code, although I'd do some testing on Android Studio to test how it looks on various Android devices (that app has an interface that displays it). The code would contain comments that describe the functions of variables and certain parts of the code, these would be used sensibly to avoid clutter. Variables would be written in camel case (e.g 'firstName') and be logical: they wouldn't have cryptic names or names that wouldn't be recognised by a competent programmer.

In the early stages of development, flow charts and pseudocode algorithms would be made to show and plan various functions in this app, so they can be implemented.

The app would be programmed in Python or Java, with the front-end of the website obviously written using HTML and styled using CSS.

Availability

This app would be cross-platform and able to run on Android and iOS, including on the app store for Chromebooks. For other devices that wouldn't be supported by a native app at the time of release, there would be a web interface at https://time.learn (not available). You'd be able to sign up to TimeLearn using the site signup form, which

This is a purely hypothetical idea from some inspiration that I had, loosely based upon various task scheduling apps (Todoist, Microsoft To-Do) and Adapt.

would be fully validated to prevent SQL injection attacks that could damage database security and user trust, or using the OAuth API of Google on Android devices, which would allow your application data to be stored in a dedicated folder on your Google Drive, which only the app would be able to read and write to. The user would be able to revoke their permission for this online. For those signing up through the website, I would require: their username, a password and their date of birth. This would be retrieved automatically for those signing up via Google (they'll sign in via OAuth, not with a password). **This app would be free to all. It wouldn't be sold for profit.**

I repeat, this is a purely hypothetical idea.