

SyncScheduler

created by: [Mariia Portnykh](#), [Semen Pobrodilin](#)

Table of Contents

SyncScheduler.....	1
Project idea	2
Main features:	2
What problem our app solves.....	2
What we want to learn?	2
Libraries.....	3
Success requirments:	3

Project idea

smart planner. A calendar app capable of analysing your schedule and suggesting optimal dates and times for your events. This app will be especially useful for students or people with busy schedules.

Main features:

1. Two types of events:

- **Static:** Events that cannot be rescheduled. For example, a student's schedule
- **Dynamic:** Flexible events for which you can allocate different free time. For example, a meeting with a friend.

2. **Smart scheduling:** when creating a dynamic event, the user specifies their wishes, such as specific days, times and even the duration of the meeting. Based on the user's wishes, the app will analyse the user's schedule and suggest suitable dates for the user's request.

3. **Conflict notification:** when a user may need to add an unscheduled static event and there is a conflict that the time is already taken, the calendar will notify the user. In case a more important event with a limited time slot occurs and it cannot be scheduled without rescheduling, the calendar will offer to move the lower priority event if available.

4. **Event Prioritisation:** All static events are prioritized over dynamic events. However, the user also has the ability to specify the priority for dynamic events, which can simplify conflict resolution, and in case of a higher priority event with a narrower time frame in which the event cannot fit, the calendar can offer to move the less significant event.

5. **Maintain an event rescheduling counter:** If a user has been rescheduling an event for a long time the calendar will also notify the user.

What problem our app solves

Our app should help people with busy schedules to optimally organise their lives by selecting convenient dates and times and saving an hour on manual planning

What we want to learn?

1. Create a gui application in rust
2. Learn to implement complex algorithmic computations in Rust, that will be needed to optimise schedules and suggestion of optimal dates.
3. Integrate database into rust project to store events data
4. Write quality code in rust

Libraries

1. Tauri + egui / rust slint (for gui, we did not decide yet)
2. Chrono (for working with date)
3. Diesel (for database integration)
4. ICalendar

Success requirments:

1. The user must be able to schedule two types of events: static and dynamic events
2. Ability to schedule an event in 2 ways: via calendar, when the user searches for this or that date by flipping the calendar, or via dynamic window, so that the user does not have to flip the calendar manually, just set the date in the format 02.04.2025 in the window and all the data to the event and it will be automatically added to the calendar.
3. Optimised date suggestion taking into account the conditions set by the user
4. If there are conflicts between events, the application should notify the user and suggest new date options for rescheduling.
5. Ability to prioritize events
6. Taking priority into account when selecting optimal dates and organising