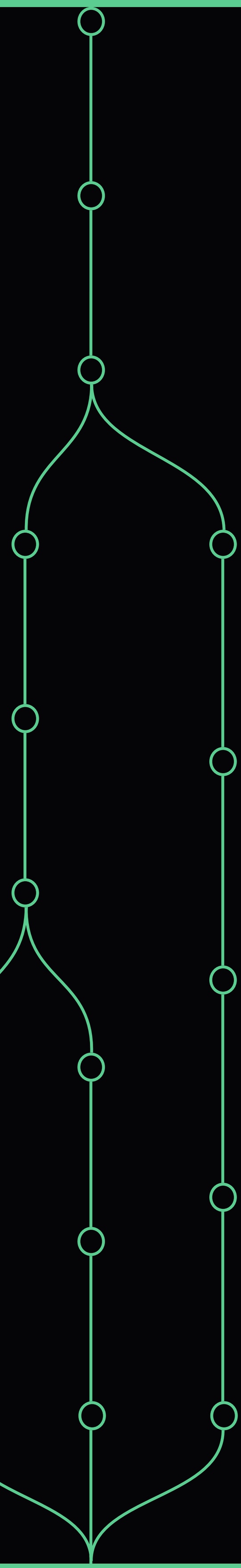
Chronos

Track Full Stack

September 1, 2025



INNOVATION CAMPUS

ucode connect

Contents

Innovation Campus

ucode connect

Engage

DESCRIPTION

Good day!

An event calendar is a useful tool. It can be used for creating, managing, and searching for events. Event calendars are a way to be informed about upcoming events and to keep people in the loop about meetings, conferences, seminars, etc.

Some of the benefits of such calendars include:

- organization and efficiency
- holding teams accountable
- saving time
- significantly improving your results

Planning plays a pivotal role in effective time management. Time management, in turn, is vital not only in the organization of the working process but also in streamlining one's personal life. Planning lets individuals split big tasks into smaller parts, thus helping them to complete assignments step by step and on time. It also makes it clear what requires urgent attention and what can be done a little later.

In this challenge you're going to dive into the ocean of time and dates. At first glance, it looks quite easy, but, in fact, it has a large number of pitfalls and is a quite complicated task. So, it requires patience and attention to all details.

In this challenge you will again encounter implementing third-party APIs in your solution. Most third-party APIs have requirements and regulations, so if you need to adapt the APIs for the program with particular specifications, this may not be an optimal solution. However, most of the time, third-party APIs will meet your needs and may become a more secure and efficient solution.

Use the best approaches to time management and planning to complete this challenge!

BIG IDEA

Time management.

ESSENTIAL QUESTION

How can technology help you stay organized and manage your time?

CHALLENGE

Build a simple and powerful time planner.

Investigate

GUIDING QUESTIONS

We invite you to find answers to the following questions. By researching and answering them, you will gain the knowledge necessary to complete the challenge. To find solutions, ask the students around you and search the internet. We encourage you to ask as many questions as possible. Note down your findings and discuss them with your peers.

- What is business logic?
- What do you think is the best framework for this solution?
- From a technical point of view, what are the differences between an arrangement, a reminder, and a task?
- What are the main advantages of process automation? What about disadvantages?
- Why do unit tests save a lot of time during development?
- What is the difference between relational and non-relational databases?
- What is a Docker container? Why do developers use Docker?
- How is a Docker container different from just installing the app on your computer?
- What is the difference between Docker and traditional deployment?
- What is an image in Docker? How does it relate to a container?

GUIDING ACTIVITIES

Complete the following activities. Don't forget that you have limited time to overcome the challenge. Use it wisely. Distribute tasks correctly.

- Find the information about where to start developing a full-stack service.
- Try to choose the best solution.
- Find the web service testing tool that you like the most.
- Write some JavaScript code that counts how many days are left from 'date1' to 'date2'.
- Investigate what default duration for arrangements is set in Google Calendar (or similar services).
- Research and decide how to implement the different views of the calendar (week, month) with minimum effort.
- Find out how the process of automatic user notification is implemented in any calendar you like.
- Clone your git repository that is issued on the challenge page.
- Start to develop your solution.

ANALYSIS

Analyze your findings. What conclusions have you made after completing guiding questions and activities? In addition to your thoughts and conclusions, here are some more analysis results.

- Be attentive to all statements of the story.
- Analyze all information you have collected during the preparation stages.
- Complete the Document section while developing a challenge. It is described after the Act section.
- You can proceed to Act: Creative only after you have completed all requirements in Act: Basic. But before you begin to complete the challenge, pay attention to the program's architecture. Take into account the fact that many features indicated in the Act: Creative require special architecture. And in order not to rewrite all the code somewhen, we recommend you initially determine what exactly you will do in the future. Note that the Act: Basic part gives the minimum points to validate the challenge.
- As you design the program, think of scalability. You may want to integrate this solution into your future projects.
- Be sure to validate all input fields with relevant error messages.
- Remember to properly handle all possible errors.
- Your service must not crash in no way.
- Make sure that your web app is responsive and is correctly displayed on different screen resolutions (mobile, tablet, etc.).
- Complete tasks according to the rules specified in the following style guides:
  - HTML and CSS: Google HTML/CSS Style Guide. As per section 3.1.7 Optional Tags, it doesn't apply. Do not omit optional tags, such as `<head>` or `<body>`
  - JavaScript:
    - JavaScript Style Guide and Coding Conventions
    - JavaScript Best Practices
    - Airbnb React/JSX Style Guide
    - Node.js Best Practices
    - MongoDB Best Practices
    - Mongoose style guide
    - Docker Best Practices
- The solution will be checked and graded by students like you. Peer-to-Peer learning.
- If you have any questions or don't understand something, ask other students or just Google it.

Act: Basic

ALLOWED STACK

`HTML, CSS, JS (Node, React), MongoDB, Mongoose, Docker`

DESCRIPTION

Create a service called chronos that helps organize meetings, tasks for the day/month/year, and events.

NOTE: this service will be a useful addition to some of your upcoming projects. Take this into consideration as you design the solution and make sure that it can be incorporated in another program.

BUSINESS LOGIC

We advise you to use the user registration form from your previous challenges, so that you to have time for the more interesting features in this service.

First of all, the user must be able to register in your app. The entity of the calendar must be created automatically for each user created. It will be the main and default calendar. The user can create more calendars in their profile and each of them can have individual events. Also, include a built-in calendar with national holidays in accordance with the user's region.

HINT: You can find the appropriate API if you make the right request on Google.

Each of these additional calendars is removable and can be hidden, unlike the main calendar. The user must have the possibility to choose a calendar name, its description or the color of its events. These settings must be editable in the future.

The user can create an event and choose a category for it:

- arrangement
- reminder
- task

Each category has its own functionality in accordance with its purpose, such as a reminder or task description. It is a great idea to set a default duration for arrangements. The important thing is that you must implement event generation in two ways:

- by clicking on the blank space on the calendar itself (here the user won't be redirected to another page)
- by clicking a special button (here the user will be redirected to the event creation page)

Regardless of the type, every event in your database must have a name and date, and must be assigned to one of the calendars. In each category, you are free to add as many fields as you think would be useful.

Your app must provide the possibility to invite friends to an event via email. The user can share their calendar and invite others. This means that several users can create events on the same calendar.

Each of the calendar users must have the possibility to choose the color of the events they create. If there is more than one user in a calendar, consider what rights each of them can have. You can allow anyone to delete any event (even if it was not created by this user) or you can limit it.

VISUAL/FRONT

Visual implementation is your field for creativity. You have no restrictions in how your service must look like. Everything is allowed if it does not interfere with the required functionality. However, it is possible that your implementation of displaying some features will be similar to someone else's. For example:

- a possibility for the user to choose the representation of the calendar (week, month and year). The calendar must display weeks' numbers in the weekly representation
- an ability for the user to choose which categories of events will be displayed in the calendar (work, home, etc.)
- a search bar that allows the user to find the desired event quickly

All events must be clickable and contain information for preview. You must properly display all events in accordance with their duration. But be attentive, not every display option fits every representation.

Make sure that forms to create new events or tasks are simple and understandable for inexperienced users.

DOCKER SETUP

Your task is to set up containerized environments for both backend and frontend services. You are free to organize the structure as you see fit, as long as the application runs correctly and supports further development or deployment. The visual structure of your Docker setup is not restricted — creativity and clarity are encouraged.

However, certain technical requirements must be met:

- The frontend and backend must each run in their own Docker containers
- Each service must include a valid and working Dockerfile
- You may organize services independently or together using docker-compose

As for the database, you have two options:

- Connect your application to a cloud-hosted MongoDB Atlas instance
- Or configure a local MongoDB container via Docker Compose

Optional but encouraged:

- Create a docker-compose.yml file that launches all services (frontend, backend, database) together
- Ensure all necessary environment variables are configured and passed securely
- The main goal is to allow a new developer to clone your repository, run one or two commands, and have a working full-stack environment with minimal setup effort.

SEE ALSO

TDD

MongoDB Atlas

Act: Creative

DESCRIPTION

It is the place where your imagination and creativity play a significant role. Implement additional features to make the program better and more unique. Listed below are a few ideas you can add to your program. You can come up with everything you want to improve your program. Creative features:

- push and email notifications
- event sharing functionality
- a simple chat for shared events and tasks
- various calendars and events representation
- other creative features

Document

DOCUMENTATION

One of the attributes of Challenge Based Learning is documentation of the learning experience from challenge to solution. Throughout the challenge, you document your work using text and images and reflect on the process. These artifacts are helpful for ongoing reflection, informative assessment, evidence of learning, portfolios, and telling the story of challenge. The end of each phase (Engage, Investigate, Act) of the challenge offers an opportunity to document the process.

Much of the most profound learning occurs by considering the process, thinking about one's own learning, analyzing ongoing relationships with the content and between concepts, interacting with other people, and developing a solution. During learning, documentation of all processes will help you analyze your work, approaches, thoughts, implementation options, code, etc. In the future, this will help you understand your mistakes, improve your work, and read the code.

It is vital to understand and do this at the learning stage, as this is one of the skills you will need in your future job. Naturally, the documentation should not be voluminous. It should be presented in an accessible, logical, and connected form.

So what should you do?

- A nice-looking and helpful README file. For people to want to use your product, their first introduction must be through the README on the project's git page. Your README file must contain:
  - Short description. It means that there must be some info about what your project is. For example, what your program does.
  - Screenshots of your solution. This point is about screenshots of your project "in use".
  - Requirements and dependencies. A list of everything that needs to be installed on any machine to build your project.
  - How to run your solution. Describe the steps from cloning your repository to the first launch of your program.
- Full-fledged documentation in any form is convenient for you. By writing this, you will get some benefits:
  - you have an opportunity to think through implementation without the overhead of changing code every time you change your mind about how something should be organized. You will have excellent documentation available for you to know what you need to implement
  - if you work with a development team and they have access to this information before you complete the project, they can confidently start working on another part of projects that will interact with your code
  - everyone can find how your project works
- Your documentation must contain:
  - description of progress after every competed CBL stage
  - description of the algorithm of your whole program

Keep in mind that peers will check the implementation of this stage at the assessment!

Also, several links can help you:

- Make a README
- How to write a readme.md file?
- A Beginners Guide to writing a README
- Google Tools - a good way to journal your phases and processes:
  - Google Docs
  - Google Sheets
- Git Wiki - a section for hosting documentation on Git-repository
- Haroopad - a markdown enabled document processor for creating web-friendly documents
- Canva - a good way to visualize your data
- code commenting - source code clarification method. The programming language determines the syntax of comments
- and others to your taste

Share

DESCRIPTION

Last but not least, the final stage of your work is to publish it on LinkedIn in a form of post. This step isn't just about showcasing your work — it's a crucial part of Challenge Based Learning framework. During this stage, you will discover ways of getting external evaluation and feedback on your work. Analyzing your process helps you to understand your strengths and areas for improvement, while sharing your insights invites valuable feedback from peers and professionals.

Pay attention to details:

- aim to make your post both informative and reflective
- start by briefly introducing the challenge and its purpose
- highlight the key challenges you faced and the technologies or solutions you used to overcome them
- share any significant findings or lessons learned during the process
- conclude with a reflection on how this experience has prepared you for future work and invite others to share their thoughts

Helpful tools:

- Canva - a good way to visualize your data
- QuickTime or OBS - an easy way to capture your screen, record video or audio

Don’t forget to tag your post with #InnovationCampusKhPI to connect with the community! Follow and mention via @ Innovation Campus of NTU "KhPI"!