Instructions and Information on the Project

Inhalt

1 Registration and team building

Follow the GitHub Classroom invitation from the CampUAS course to the lecture. If you are not yet registered in GitHub, you must do so now in the first step.

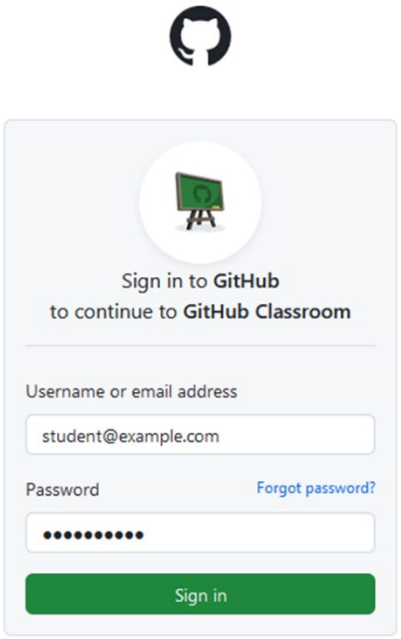


Figure 1 GitHub sign-in

In the next step, select your name from the list and join the specified classroom.

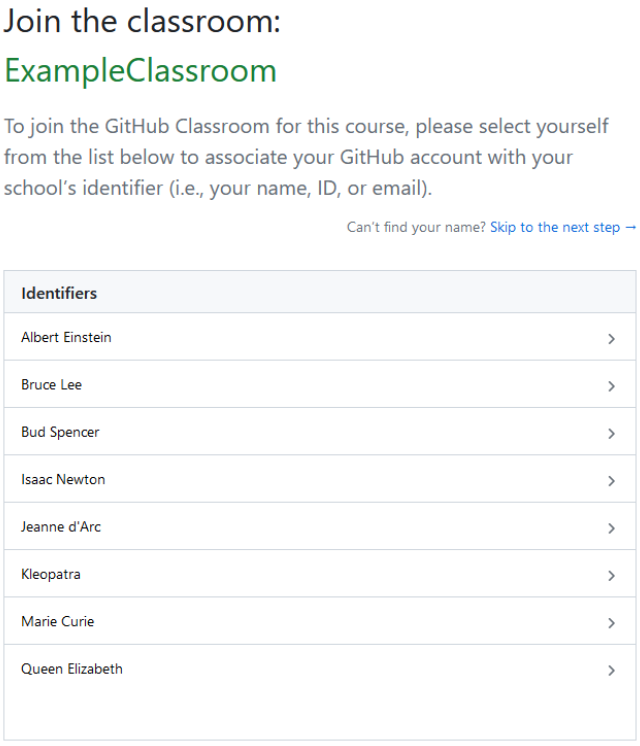


Figure 2 Choose your own name

Before you can accept the invitation to the project, you must create a team or join a team (only a limited number of teams and team members are possible).

Remember that once you have joined a team, this cannot be undone afterwards. So be absolutely sure which team you join! Communicate with your fellow students in advance.

The following is an example of a team selection or creation page.

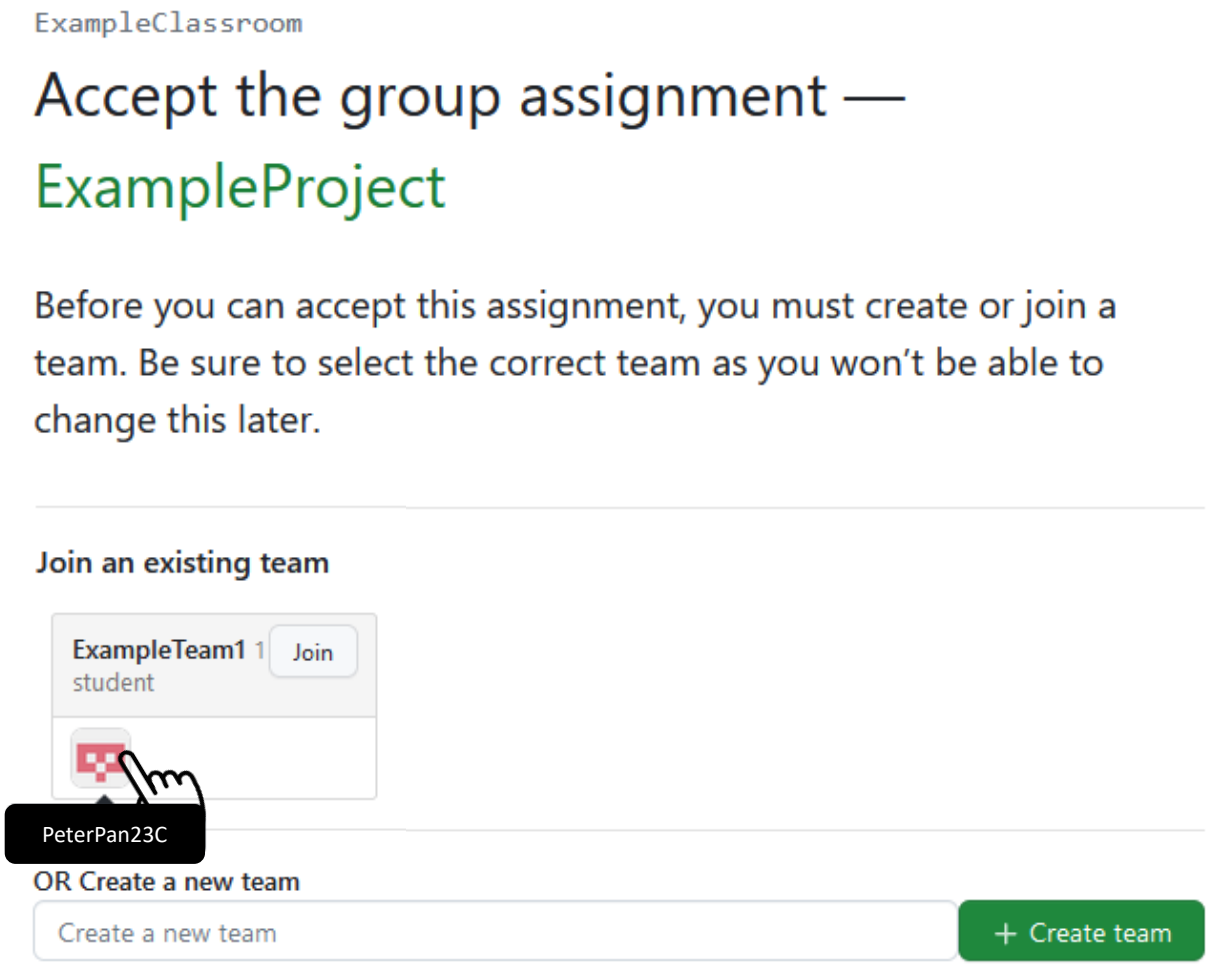


Figure 3 Team creation/assignment

You will then be asked to accept the assignment to the selected project group. Confirm this with „Accept this assignment".

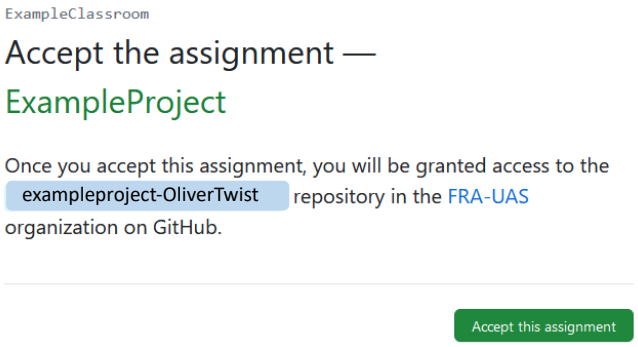


Figure 4 Project group assignment

Your membership has been accepted; you must now wait a little and refresh the page.

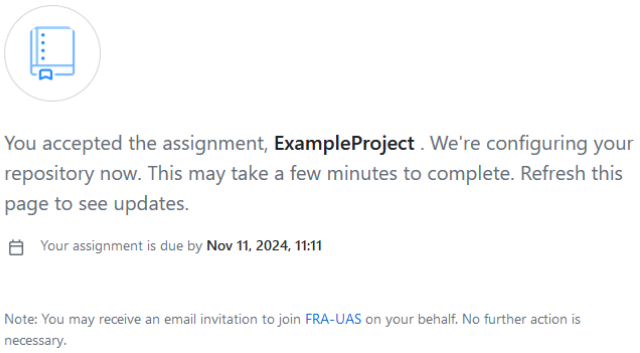


Figure 5 Acceptance of the membership

You should then be shown a link to your project.

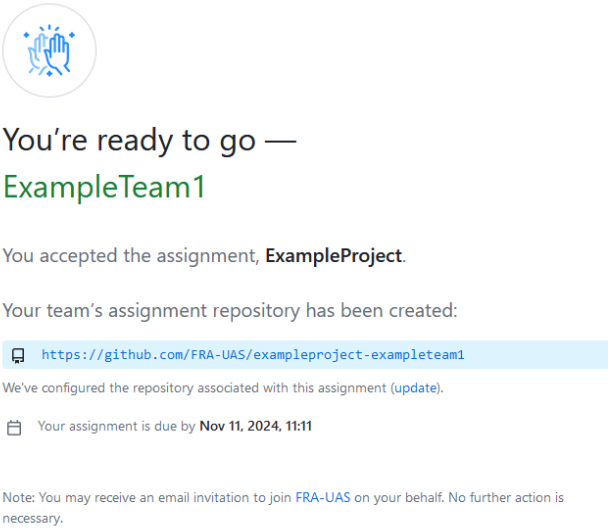


Figure 6 Link to the repository

You should make a note of the link displayed or bookmark it in your browser (note: the link shown here is only an example).

You are now in the GitHub repository of your project, which your entire team has access to. **During the project, work exclusively in the repository intended for this purpose. If you work in a different repository, this cannot be evaluated!**

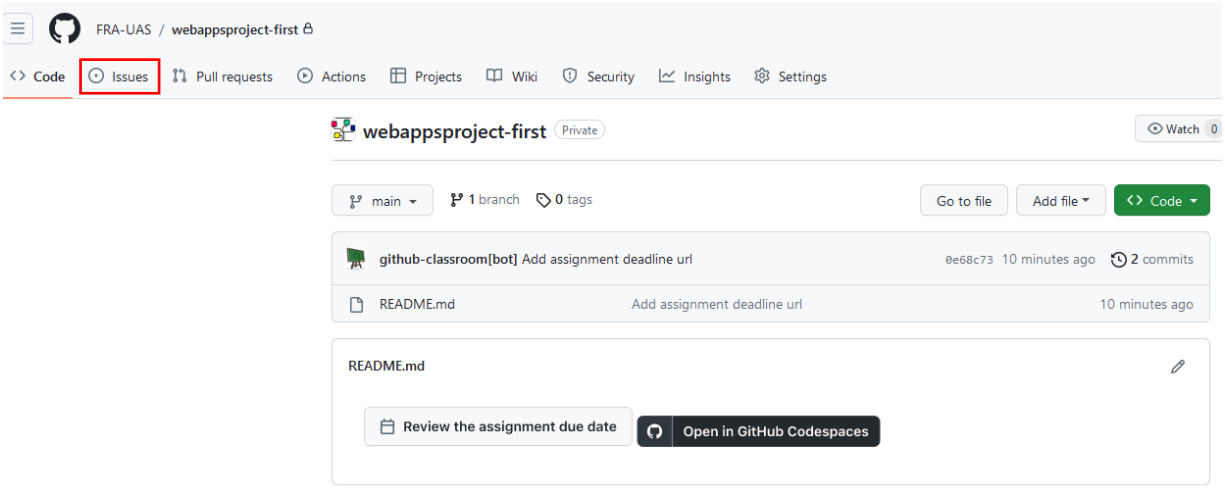
1.1 Project description

The project description is in your GitHub repository under ProjectDescription.md.

1.1.1 Project planning

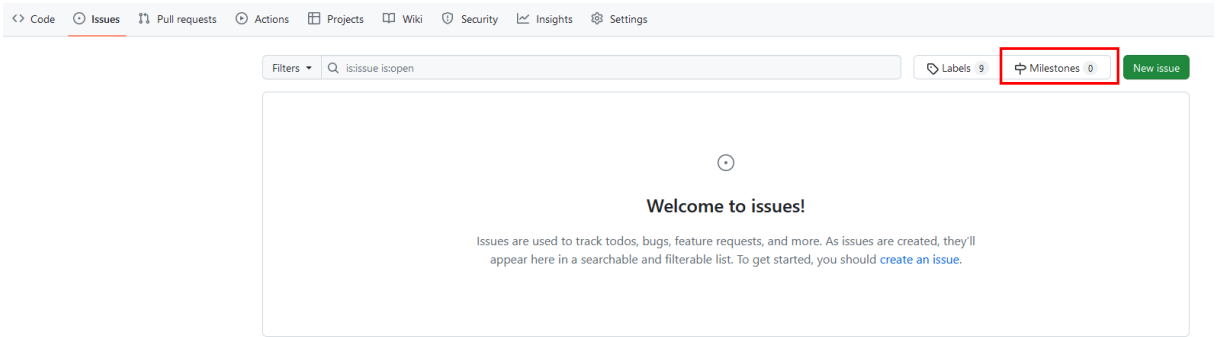
A central point of your project work in the team is project planning. This always takes place at the start of the project and is monitored and updated throughout the entire project duration. Planning is the basis for a successful project and, therefore, also for a corresponding evaluation. The following is a brief explanation of how to carry out project planning.

The following image shows an overview of an exemplary repository.



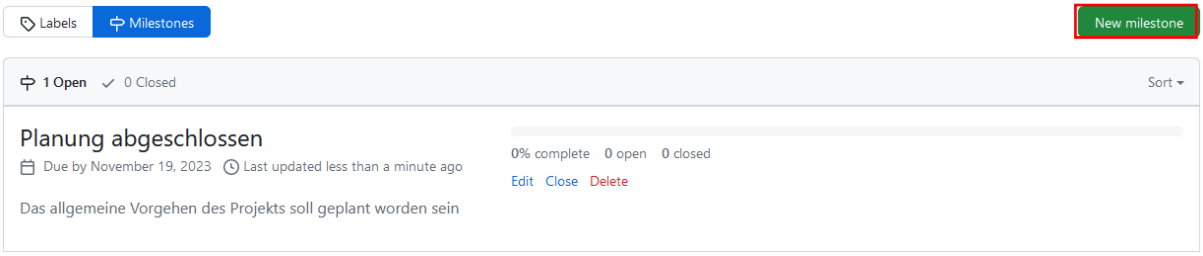
You will find the Issues tab (work packages) in the top menu bar. Here, you should define all topics or problems that must be solved during the project. Create a kind of work plan to track problems and discuss issues. With GitHub Issues, you can express ideas with GitHub Flavored Markdown, assign and mention contributors, respond with emojis, clarify with attachments and videos, and reference code such as commits, pull requests and deploys. With task lists, you can break down large issues into tasks, further organize your work with milestones and labels, and track relationships and dependencies.

Open the Issues view. And then switch to the milestones.

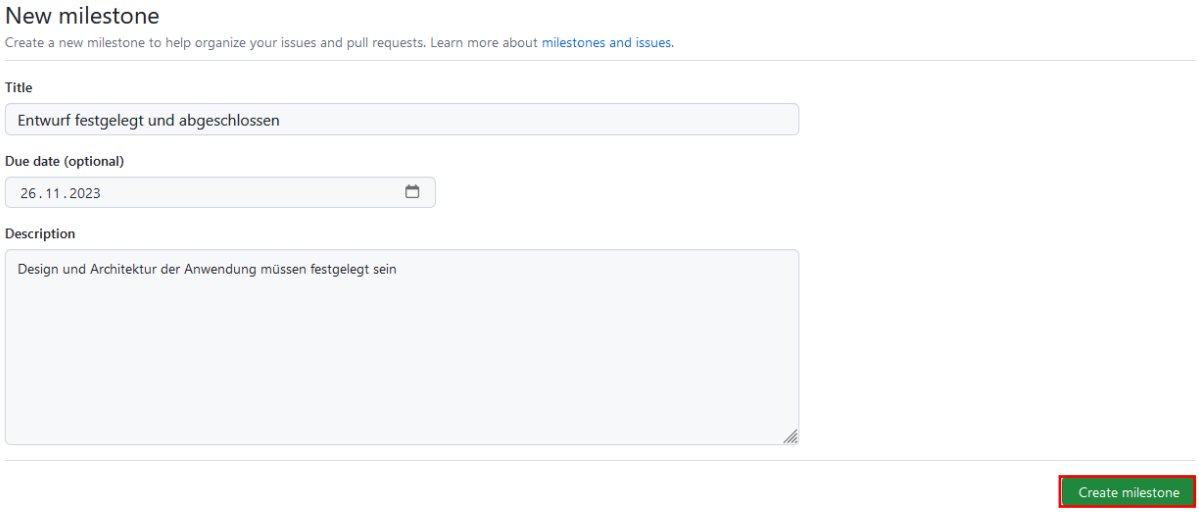


Initially, you should roughly subdivide your project based on the milestones' results. Exemplary milestones could be planning completion, design completion or prototype development. Remember to give the milestones a completion date and a more detailed description. Create the milestones in consultation with your team.

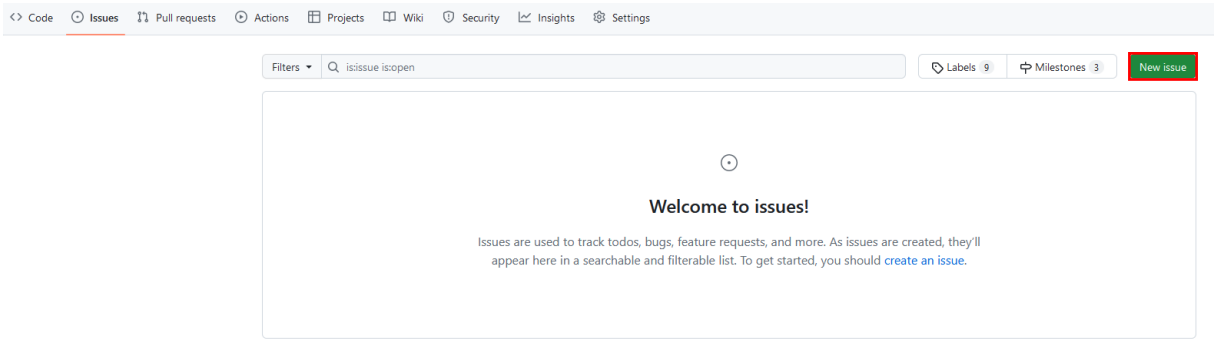
The following image shows how to create a new milestone. Create all the milestones you have worked on in consultation with the team.



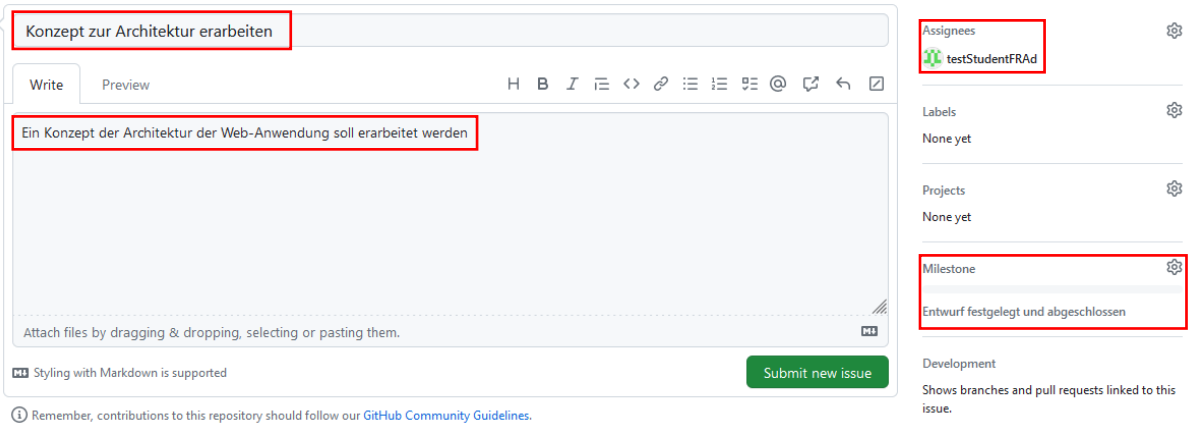
Name the milestones as meaningfully as possible. A designation such as Milestone 1 or First Step is not meaningful and, therefore not sufficient.



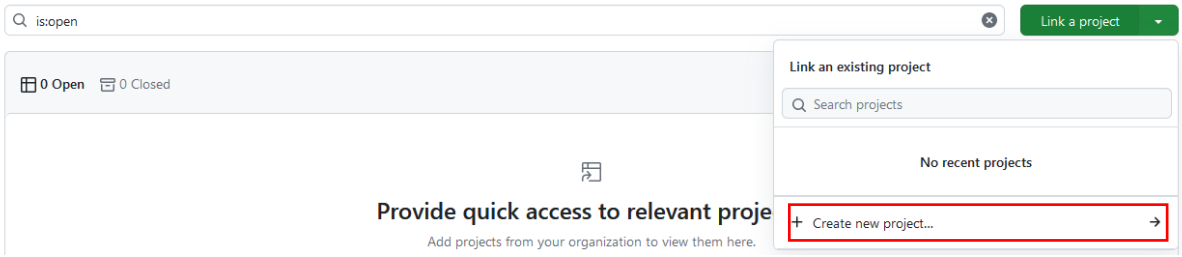
Once you have created the milestones as a rough plan for your project, you should now create your work packages for fine-grained project planning. Now, create these work packages using issues.



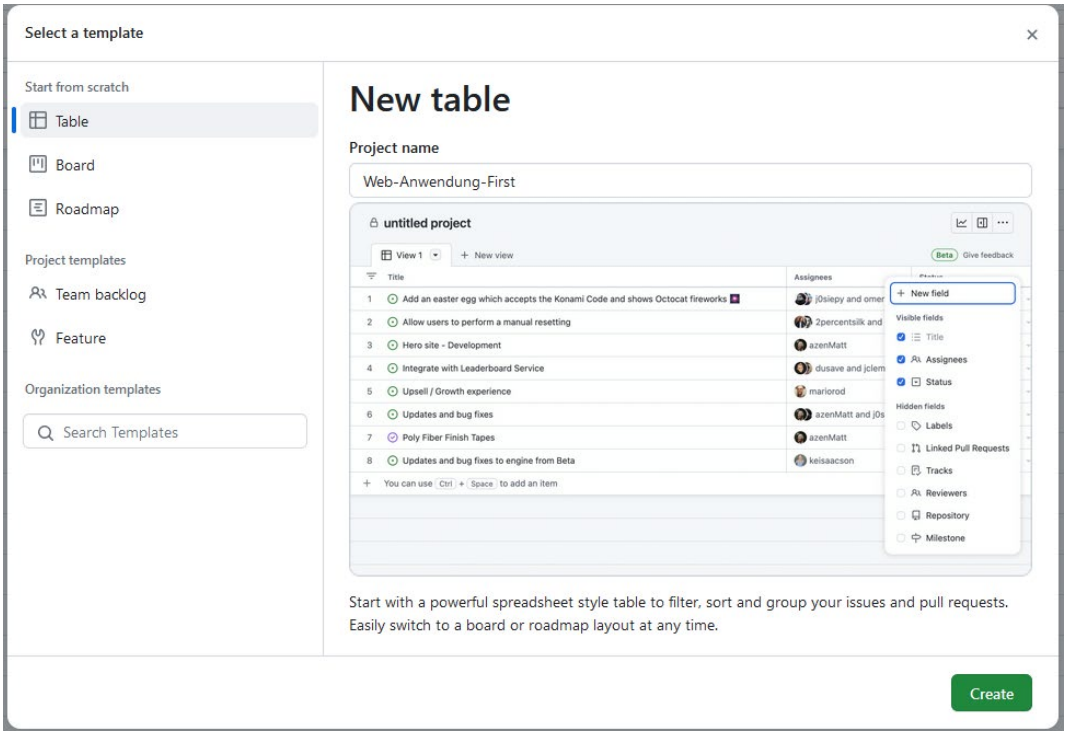
On the right-hand side, you can define the assignees of the work package (see Assignees below). Under Labels, you can use an existing label or create your own. However, you can also adapt and change this during the project. The only important thing here is the assignment to a milestone.



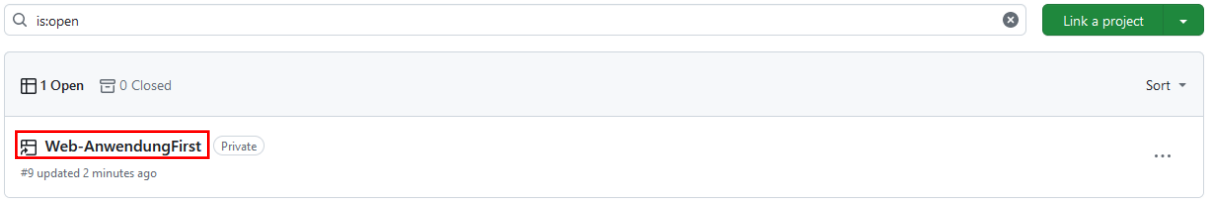
Once you have created several work packages for all your milestones, a project should now be created. Milestones always consist of several work packages.



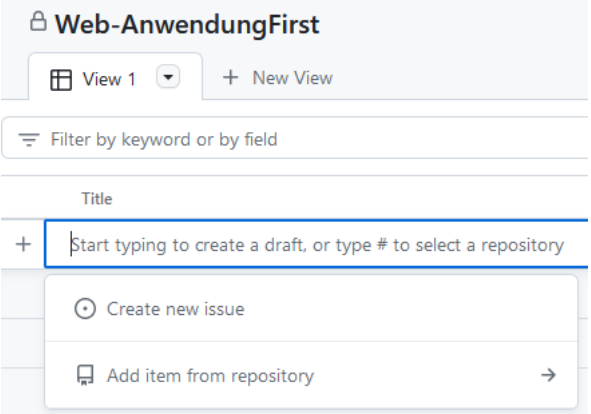
Create a new project and give it a meaningful name first, e.g. the name of your project. Leave the selection set to Table.



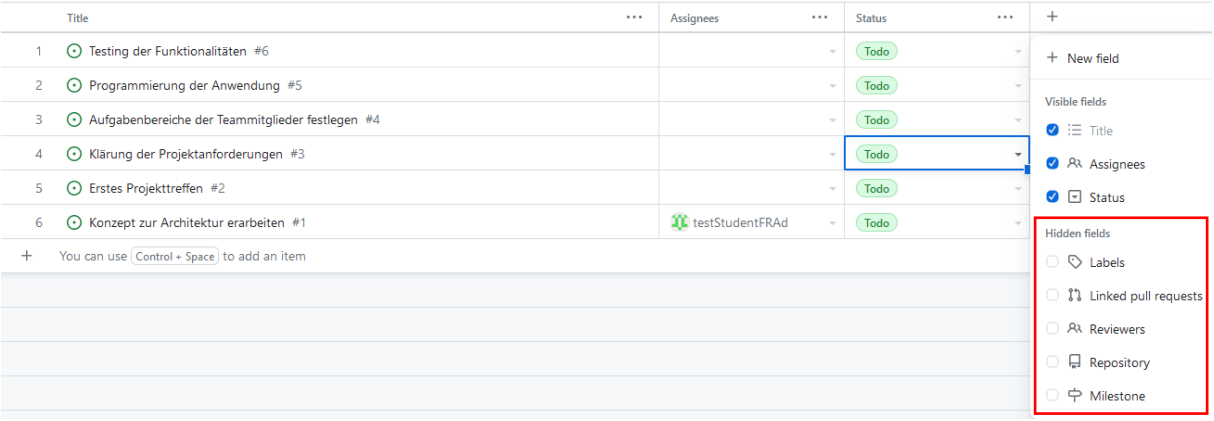
Then switch to the project view.



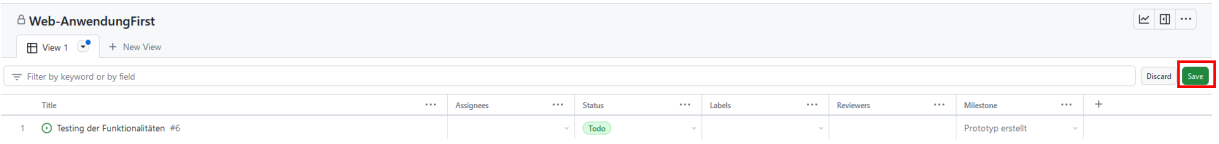
Here you can now add your previously created work packages by clicking on the + symbol under Title. And call up Add item from repository.



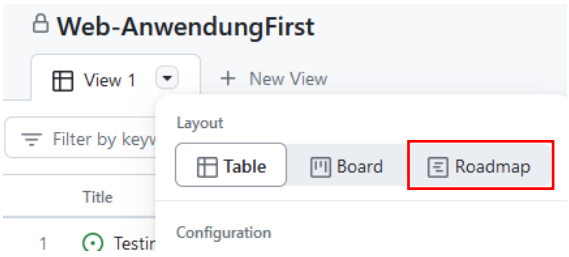
You will now see the individual work packages in the table view. Here you can assign the current status to the work packages, e.g. Todo, In Progress or Done. You can also define additional statuses if required. You can add further fields, such as labels or milestones, using the + symbol in the top line of the table.



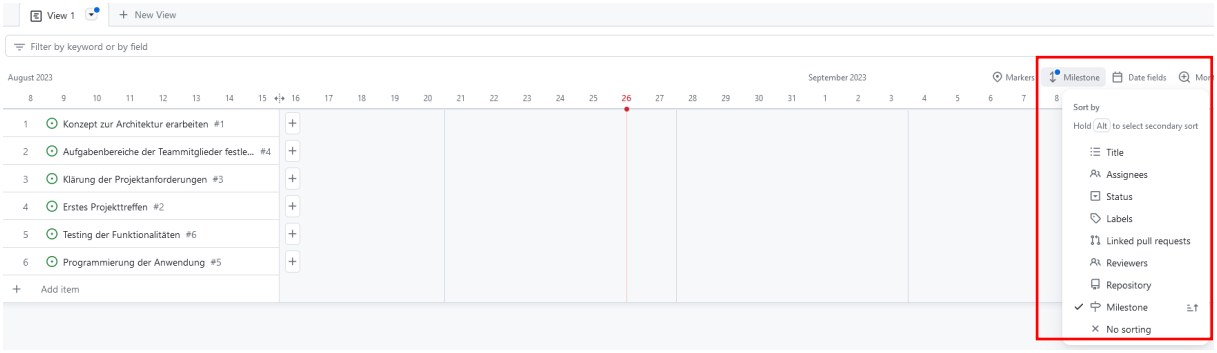
Next, save the table by clicking on Save.



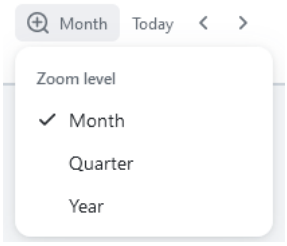
Then use the drop-down menu under View 1 to switch to the roadmap view. (Note: You can change the name of the view at any time!)



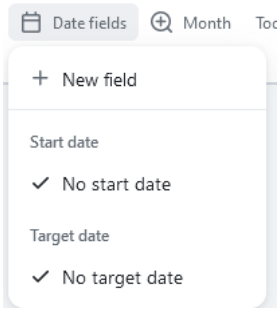
Here you can start by sorting the work packages according to milestones if you wish.



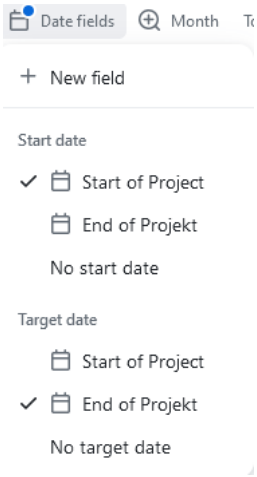
You should now define the timeline for the individual work packages. First, adjust the timeline by clicking on Month and selecting the desired time frame display, e.g. Quarter.



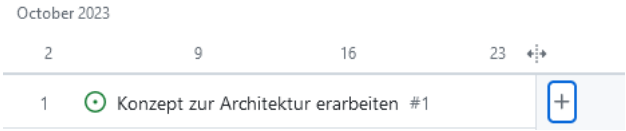
Now create two fields via the Date fields selection.



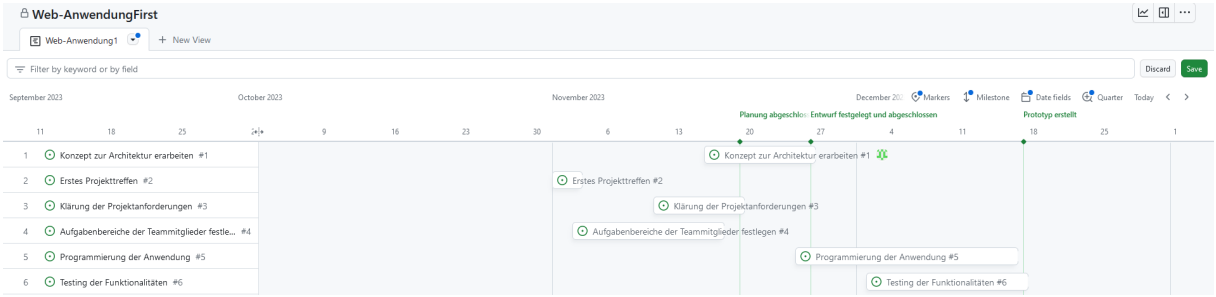
A field that defines the start of your project and a field that defines the end of your project needs to be defined.



Now select the respective + symbol in the line behind the first work package.



You can then arrange this on the timeline and define the duration of the work package by specifying the start and end dates. You can also display the milestones by selecting Markers. After the initial planning of your project, a roadmap overview such as this should have been created.



Always remember to keep the project overviews up to date as your project progresses. This includes, for example, adjusting the status of the work packages or adding new work packages.

2 Dates and project schedule

The project duration is approx. 10 weeks, excluding winter and Christmas vacations.

Project start: 04.12.2023

End of project: 25.02.2024 23:59 p.m.

Submissions must be made by no later than 25.02.2024 23:59 p.m.

In addition to the project planning in GitHub, a working implementation or configuration that meets the project description is required. A presentation of all team members involved in the application must also be created with Panopto or a similar application and submitted in CampUAS by the submission date.

- Setting up a complete 5G environment (**You define the scope in the team**)
  - 5G Core (5GC)
  - User Equipments (UEs) and Radio Access Network (RAN)
  - Data Network with Applications (DN)
- Project planning in GitHub
  - What are the tasks to work on?
  - Which group member works on what?
  - When are these tasks to be performed?
  - What is the progress?
  - Who is responsible for what?
- Documentation and Analysis in GitHub
  - README.md (written and formatted documentation)
  - PCAP-files (captured Wireshark traces)
  - Figures (e.g. MSCs, flow charts, wireshark traces)
- Linux use (no Windows-Subsystem for Linux!!!)
- GitHub Classroom must be used
  - Don’t create your own repositories; use the one automatically generated
  - private Repository per Group
- Continuous project plan (milestones and work packages)
- README.md
  - Format of documentation (formatting/consistency)
  - Outline of the document, e.g.:
    - Introduction, Requirements and Planning, Realization and Analysis, Conclusion, References (primary sources)
  - Spelling and grammar
  - Representation (e.g. readability of figures)
- Project structure in GitHub, e.g.
  - Documents
  - Configurations
  - Scripts
  - Figures
  - Traces
  - User guide
  - Presentation slides

Project Presentation incl. Demo

- Submission in CampUAS via Panopto
- Preparation of Slides
- Preparation of Demonstration
- Presentation
    - think about the time
    - equally distributed across all group members
    - clear structure
- Demonstration
    - recognizable for the viewer
- What do you want to tell us?

3 Project Assessment

The project and the final evaluation are based on the following criteria:

1. Project continuity (continuous working method) 15%
   - Only recognizable continuous performance can be evaluated
   - A realistic work plan is essential
   - Realistic structure of the workload
   - Individually assessable
2. Project planning 25%
   - Continuity (see point 1)
   - During the entire duration of the project
3. Project Documentation 20%
   - Continuity (see point 1)
   - Who contributed what (GitHub Commit?)
   - Quality of the documentation
   - Complexity/volume
4. Project Repository 25%
   - Project structure
   - README.md
5. Project Presentation incl. Demo 15%
   - Style
   - Structure
   - Integration in the presentation?
   - Special features?
   - Adherence to time?
   - Equally distributed among group members?
   - Content (Were important aspects explained?)
   - Special features?