
Contents

Introduction	v
1 Getting started	1
1.1 Getting an overview	1
1.2 Creating the wiki team	1
1.3 Let everyone know the essentials	2
1.4 Review tasks and timeline	2
1.5 Distributing responsibility	3
2 Design	5
2.1 Collaboration with the Creativity/Design Team	5
2.2 Design Parameters	6
3 Documentation	9
3.1 General considerations	9
3.2 Team Meetings	10
3.3 Lab	11
3.4 Integrated Human Practices	12
3.5 MeetUps	13
4 Best Wiki	15
5 File structure	17
5.1 Naming Conventions	18
6 Website and page structures	19
6.1 Standard URL Pages	19
6.2 Additional Pages and Considerations	20
6.3 Structuring for Easy Navigation	20
6.4 Enhancing the Navigation Bar	21
6.5 Footer	23

6.6	Header	24
6.7	Additional Elements	25
7	Media	27
8	Coding prerequisites	29
8.1	Hardware	29
8.2	GitLab Access and Cloning the Repository	29
8.3	Coding Environment (IDE) and Git	31
8.4	Cloning your repository	32
8.5	Other Software	33
9	General Coding	35
10	React	37
10.1	Getting started with your React project	37
11	Additional tools	39
12	Troubleshooting	41
13	Guide for team	43
14	The Wiki Freeze	45
15	The Wiki Thaw	47
	Associated software	I
	Attributions	III

© Copyright (c) Liliana Sanfilippo 2025

Published by Liliana Sanfilippo
liliana.sanfilippo@uni-bielefeld.de

This guide is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0). This license allows others to remix, adapt and create from this work for non-commercial purposes, provided they credit the authors and license the new creations under identical terms.

First impression, September 2025

Introduction

This guide is a comprehensive introduction to creating and managing an iGEM wiki from the absolute basics to the technical documentation of a modern React-based template wiki designed specifically for iGEM teams.

It is organized into various sections that cover specific aspects of project management, collaboration, design, documentation, and technical implementation.

Throughout this guide, I will assume a successful activation of the team Wiki but will walk the reader through the fundamental aspects of setting up the Wiki, as well as providing a general overview of project documentation. I will emphasize key elements that are essential for the Wiki team, such as effective documentation, clarity, and organizational strategies. The coding section will specifically focus on React applications, explaining the basics and offering guidance on best practices and methods for creating a dynamic and user-friendly interface.

At the end of the guide, there will be an index list to help you easily find specific topics and navigate through the content more efficiently.

This guide also serves as documentation to the associated template wiki, which teams are free to use as a starting point. It can be found at <https://github.com/liliana-sanfilippo/igem-wiki-guide>.

The Template Wiki is a React-based, easily customizable template for iGEM wikis. It is aimed at first-time teams with little experience as well as advanced iGEM teams. The goal is to make it easier to get started with the technical implementation of your wiki without having to compromise on functionality or flexibility.

The template is currently developed further, including some independent packages. It offers a modern, modular code base that can be easily adapted and extended. The aim is better usability and maintainability compared to the example wiki provided by iGEM through additional functions, including components designed to work with the template specifically.

It is a first version and I will be working on improving it, so any feedback or ideas can be sent to liliana.sanfilippo@uni-bielefeld.de or raised as issues on GitHub.

Documentation for other tools or packages by myself is not included, but will be linked.

While updating the guide, some already finished sections will be re-added and the following aspects will be added:

- An in-depth explanation on how to implement a citation React component
- Setting Global Variables in React
- Wiki Thaw
- Utilizing Bootstrap
- FAQ
- Merge Conflicts
- Utilizing ChatGPT
- Where can I get help?
- Definitions of terms such as CSS path, CSS selector
- Errors and Troubleshooting
- html link targets
- Further Collaborations and Considerations with other subteams (e.g. Sponsoring and logos)
- Scrolling animations

As well as additional images outlining the instructions and examples given.

Chapter 1

Getting started

1.1 Getting an overview

You need to familiarise yourself with the requirements by iGEM and your team members. Some teams set high expectations for themselves in regards of design or other aspects.

1.1.1 iGEMs resources

To successfully build your iGEM team Wiki, it is essential to thoroughly review iGEM's official Wiki Rules and Resources. Pay close attention to size limitations, time restrictions, and functionality constraints to ensure compliance with the competition requirements. Familiarise yourself with best practices and common pitfalls by examining previous Wikis, particularly those from past winning teams, as they can provide valuable insights into effective strategies and potential challenges.

Before you dive into coding, make sure to read iGEM's official information on the team Wikis and the GitLab Guide. These documents will help you understand the basic requirements, technical constraints, and available tools necessary for your Wiki's development. Key resources to consult include the iGEM GitLab Guide and the Team Wiki Deliverable page, which offer an overview of important documents that will guide you throughout the process.

By following these steps, you will be better prepared to create a well-structured and compliant Wiki that effectively showcases your team's project.

1.2 Creating the wiki team

Decide on one or multiple people taking the organisational lead for the wiki and determine which people will actively contribute to the wiki development. Everyone? A set group of people? Will everyone get a section to work on?

Depending on the answer you need to consider who will teach. Further, consider who will manage coordination with other subteams.

1.2.1 Deciding on a template

The wiki lead needs to get an overview about the skills and resources available to your team. Based on the availability, you need to choose a template.

- **Markdown & Frozen Flask (Beginner):** This simple template utilizes Markdown files to create a static website. It's user-friendly, making it an ideal choice for teams just starting out.
- **HTML & Frozen Flask (Intermediate):** This template employs HTML files to generate a static website. While it is slightly more complex than the Markdown option, it remains relatively straightforward to use, making it suitable for teams with some coding experience.
- **React (Advanced):** This option is for teams with advanced skills in React, a complex framework that allows for the creation of dynamic and interactive Wikis. Before choosing this template, ensure that your team is proficient in React, as we will not provide detailed guidance on its usage. It's crucial to review the requirements and thoroughly test your Wiki before the Wiki Freeze to ensure everything functions as intended.

1.3 Let everyone know the essentials

Everyone should be aware of the requirements and mistakes that could get you disqualified to avoid the responsibility being placed on one person. Important aspects are:

- GitLab Repository Link
- Licensing
- Standard URL Pages
- Content Hosting
- iframes Usage
- Source Code Submission

Do not rely on the knowledge of former teams only as the guidelines can change.

1.4 Review tasks and timeline

Get started with the following questions:

Who will oversee

What pages do we need?

How do we distribute

1.5 Distributing responsibility

To effectively build your iGEM team Wiki, it's essential to clearly define team roles and responsibilities. Consider assigning specific positions such as content creators, developers, and designers. Utilizing task management tools can help streamline the process and ensure everyone stays organized.

Each team member should be aware of their specific tasks related to the Wiki, which may include:

- **Page Layouts and Design:** Creating visually appealing and user-friendly layouts.
- **Coding:** Implementing the necessary code for functionality.
- **Automation:** Streamlining repetitive tasks where possible.
- **Data Curation:** Organizing and managing data sets relevant to your project.
- **Media Management:** Uploading videos, taking and editing photographs (including considerations for lighting, cropping, and naming conventions).
- **Text Creation and Editing:** Writing, editing, and correcting text to ensure clarity and accuracy.
- **Task Management:** Overseeing progress and reminding team members of their responsibilities.
- **Mobile Optimization:** Ensuring the site is accessible on mobile devices.
- **Illustrations and Animations:** Creating visual elements to enhance the content.
- **Accessibility:** Implementing features that make the Wiki usable for all visitors.
- **Documentation:** Maintaining accurate records of processes and changes.
- **Development of Additional Tools:** Creating helpful scripts or tools, such as Python scripts, to automate tasks.

By clearly defining roles and responsibilities and utilizing task management tools, your team can effectively collaborate to create a comprehensive and engaging Wiki for the iGEM competition. It is recommended to distribute these tasks on multiple people.

Most tasks concerning the Wiki overlap with other subteams and should therefore be handled in cooperation with these.

Chapter 2

Design

2.1 Collaboration with the Creativity/Design Team

To create an effective and visually appealing iGEM Wiki, collaboration with the Creativity/Design team is essential. You do not only want to simply collect information on your wiki, but present it in a coherent, readable and convincing way to both fellow iGEMers and judges. Consider the following design aspects:

Choosing Colors: Select a cohesive color palette that reflects your team's branding. If the team has no prior experience with design, color psychology[?] or simply color palette choosing tools can be a helpful starting point, especially in small teams with limited resources.

Initial Design Work: Ideally, start in a design program or layout tool (e.g. Figma) to visualize your ideas before implementing them on the Wiki. Alternatively, vision boards or analogue planning methods such as drawing work as well.

Choosing Illustration Styles: Decide on illustration styles that align with your overall design and branding. Consistency in style helps create a unified look across your Wiki, making it more visually appealing and easier for users to navigate. Style considerations can be the software used for illustrations and specific details such as the usage of borders and shadows for in-house designs.

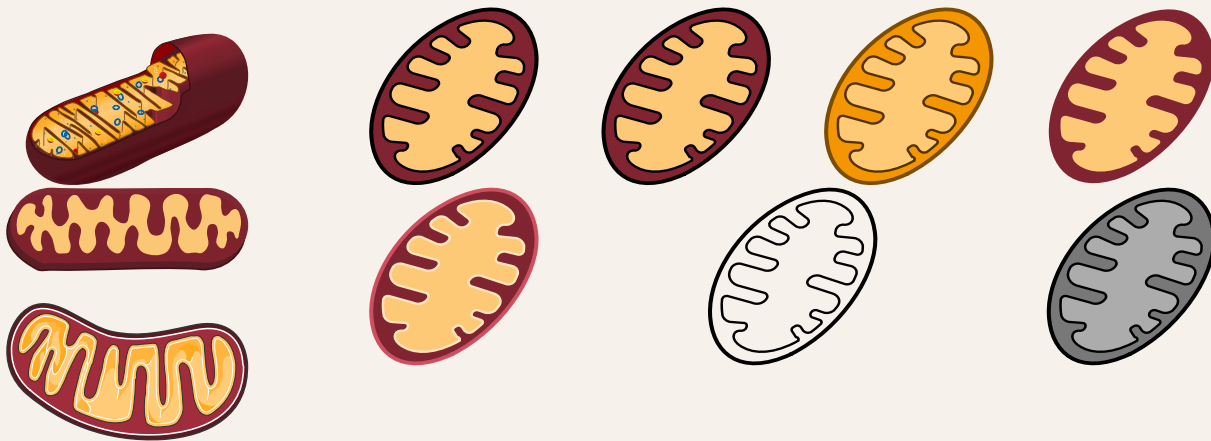


Figure 2.1: Different styles.

Logo usage: There are many possibilities to use your team logo on your wiki. Both for simple things such as buttons as well as more complex things such as animations, progress bars or a guide through your website.

2.2 Design Parameters

- **Typography:** Prioritize typography to enhance readability and aesthetics.
 - Maintain appropriate line length and spacing for readability.
 - Use standardized font sizes for consistency across the Wiki.
- **Simplicity:** Aim to simplify designs for a cleaner look.
 - Limit to 2–3 headings and colors to maintain clarity and focus.
 - Minimize the number of design parameters to streamline the design process.
 - Use colors purposefully; avoid unnecessary embellishments unless justified.
- **Consistency:** Unify illustration styles and design elements for a cohesive appearance.
 - Ensure consistent image formats and alignment (preferably left-aligned, avoiding justified text).
 - Align infographic styles to maintain a cohesive visual narrative.
- **Bootstrap Values:** Consider overriding Bootstrap defaults as needed for your design (See ??).
- **Call to action and clear navigation:** Include a welcoming call to action on the homepage, using colors meaningfully to guide users.

- Possibly add further guidance such as highlights, to guide the user through the wiki and help the judges find all necessary information.
 - Use a clear menu structure and possibly breadcrumbs.
 - Avoid deep nesting your pages and aim for a flat, user-friendly structure. You want the judges to easily find all you pages from the standard URLs.
- **Content Organization:** Clear information hierarchy helps users find what they’re looking for faster.
 - Utilize cards or expandable sections for content organization.
 - Ensure the basic information is always available and additional information is, while easily locatable, tucked away and not distracting the reader.
- **Responsiveness:** Ensure that all design elements adapt smoothly to different screen sizes (See ??).
 - Prioritize mobile usability since people may want to look at your wiki on their phones during or after presentations and poster sessions.
 - Verify that navigation, text, and visuals remain accessible and visually consistent across devices.
- **Accessibility:** Design for inclusivity by ensuring content is accessible to all users. Especially if you aim for the inclusivity special prize.
 - Maintain sufficient color contrast for readability.
 - Use semantic HTML and alt text for images.

Chapter 3

Documentation

Creating well-structured documentation is essential for effective communication and information retention. Here are some general considerations for documentation, including various methods to create PDFs and layout decisions for protocols and notes.

3.1 General considerations

Purpose and Audience

Clearly define the purpose of the document and be aware your target audience. Understanding the audience's knowledge level will guide your writing style and content depth. Of course judges will most likely come from a biotech background, but they can still be unfamiliar with niche topics. A different example where your documentation and documents need audience considerations would be flyers or videos for educational purposes, especially if you want to choose education as a special prize.

Clarity and Consistency

Use clear, concise language to convey information effectively. Maintain consistency in terminology, formatting, and style throughout the document to enhance readability and comprehension. There is no special prize for using the most complicated words or coming up with the most intricate way to describe simple procedures.

Sustainability

Decide on styles and approaches early on and think about using English from the get-go to avoid having to translate and rework your notes and protocols later on. Remember that you need to credit translators and potentially translating tools in the attributions.

Structure and Organization

Use a logical structure with a clear hierarchy, even if it seems basic or unnecessary to you.

- **Title Page:** Include the document title, author(s), date, and possibly your logo. Remember, the document is presented on your wiki, but it is possible it will be distributed in a different context, and therefore you should always add all necessary information to each document.
- **Table of Contents:** Do not forget to provide a navigation aid for longer documents. If possible, create a table of content with hyperlinks.
- **Structure** Divide content into clear sections with appropriate headings. Possibly use a glossary for intricate documents.
- **Conclusion:** Summarize key points to ensure the reader understands the main points and offer recommendations if applicable.

Choosing a Format for Documentation

Word Processing Software (e.g., Microsoft Word, Google Docs): These tools allow for straightforward document creation with various formatting options. You can export your documents as PDFs easily, ensuring compatibility across different devices and allowing collaborative writing.

LaTeX: Ideal for more complex documents, LaTeX offers precise control over formatting, making it perfect for scientific papers and technical documentation. It is particularly beneficial for documents with mathematical equations, citations, and references. Services such as Overleaf allow collaborative writing.

It is possible to create PDF-masks with design tools such as Canva, if you want to add further styles or branding to your documents.

Even if you are unsure what to use yet, a digital text file of any sort is always easier to edit and transfer than handwritten notes!

3.2 Team Meetings

It is advisable to have meeting protocols from the beginning. Depending on your team structure, experience and regarding time management, you should consider:

- **Preparing agendas:** Starting meetings with a pre-prepared agenda outlining the topics for discussion.

- This helps keep the meeting focused and provides a structure for your notes.
- **Minute taker:** Assign the task of note keeping to a specific person every meeting.
- **Tracking decisions:** Document any decisions or agreements reached during the meeting to provide a clear record of team choices.
- **File Storage and Accessibility:** The meeting notes should be accessible to the whole team afterward and multiple versions should be most urgently avoided.
- **Summaries:** Some teams may find it helpful to create meeting summaries after the fact.
- Though this can also be a time drain.

3.3 Lab

Well-maintained lab books are essential for tracking progress, reproducing experiments, and ensuring accountability. The following aspects seem basic and natural which unfortunately puts them at risk of being forgotten for this exact reason.

- **Digitalize:** Most projects get busier towards the end, and it is unlikely you will have the time to digitalize your notes later on. It is best to keep it digitally from the beginning, in whatever exact form. Even a `.txt` file is better than handwritten notes, because the text can simply be copied to the wiki in the worst case.
- **Backup:** Regularly back up entries to prevent data loss.
- **Raw Data:** Always include raw data alongside processed or analyzed data and save the data separately, too.

If you want to use a lab book software, please check beforehand if it allows you to export your (raw) data and notes in a useful way. Usually, useful ways are easily processable data files such as `.csv` or `.json`.

Protocol design examples

- JU Krakow 2024

3.4 Integrated Human Practices

Documenting Integrated Human Practices involves tracking contacts, conversations, permissions, and the use of any media, ensuring compliance with ethical and legal standards.

Both to avoid unnecessary work and to maintain a professional demeanor towards your stakeholders.

- **Tracking Contacts and Interactions:**
 - Maintain a list of **Contact Information** of the individuals and institutions you both want to contact and already contacted. Include details such as *role, affiliation, date of first contact*, etc.
 - Keep track of **Who Contacted Whom** to avoid contacting the same person multiple times. This helps in maintaining accountability and tracking follow-ups.
- **Consent Management**
 - Be sure to retain **Informed Consent for Media Usage** from the people you interviewed or otherwise created media content with. You should mention how the “freezing” of iGEM Wikis works and that it will not be possible to change or remove information after the project ends.
 - Be mindful to receive **Feedback for Quotes and Transcripts** of interviews. Especially if you need to translate conversations to English, the stakeholders should get the chance to comment and, if necessary, correct your translations.
 - Remember to ensure a tidy **Storage of Consent Information** that is accessible to the whole team.
- **Recording Conversations and Outcomes**
 - If possible, clear the **Type of Documentation** with the interviewee beforehand.
 - Try to organize **High Quality Tools** to record and test them beforehand.
 - Be aware of background noises and keep in mind that the videos could be useful for your project presentation video.
 - Create **Meeting Summaries** including *main topics, key takeaways, quotes* and *to-dos*.
- **Categorizing Stakeholders and Their Input**
 - The **Categorization of Contacts** into categories such as *Academia, Industry* or *Community* should be started as soon as possible to streamline documentation and to facilitate references back to relevant discussions.
 - Keep track of the **Implementation of Advice and Input** you received to be able to cross-reference from you Integrated Human Practice to other aspects of your project.

- **Ethical Considerations**

- Ensure that no **Sensitive Information** is included in your published material. This can include *confidential project data* or *compromising information*. If necessary, censor details to ensure the **Privacy** of individuals such as patients, children or other vulnerable stakeholders.
- Maintain proper **Attribution** for both intellectual input and media usage.
- **Maintaining Transparency**
 - Maintain **Transparency in Reporting** by including clear and concise references to your interactions with stakeholders when writing Wiki texts.
 - Be upfront about **Changes and Censorship** in your documentation.

3.5 MeetUps

When documenting an iGEM meetup, it's important to capture not only the logistics but also the valuable exchanges, collaborations, and outcomes from the event.

- **Photos and Videos:** Be sure to organize media documentation beforehand and take pictures of key moments. Not only for your team but to support the attending teams and allow them to concentrate on the event.
- **Expert Feedback:** If experts or judges were present, document the feedback they provided to your team or others. Highlight how this feedback can influence your project moving forward.
- **Material Exchange:** If any teams shared protocols, tools, or resources during the meetup, keep track of what was exchanged and which team shared it.
- **Lessons Learned:** Capture insights or takeaways from the event that could influence your project, approach, or team dynamics moving forward.
- **Online Resources:** If presentations, slides, or meeting recordings were shared, document how these materials can be accessed later (e.g., through a shared drive or link).
- **Meetup Agenda:** Include the event's agenda or a summary of the planned sessions, presentations, or workshops. Be sure to note changes.
- **Feedback:** If possible, gather feedback on-site at the end of the event to ensure high involvement and profit from fresh impressions.

Chapter 4

Best Wiki

Upload pending.

Chapter 5

File structure

To facilitate collaborative work on the wiki and minimize merge conflicts in GitLab, it's essential to establish a well-organized file structure. This will help multiple team members work simultaneously without issues, especially for lengthy sites.

Organized Components: For pages with multiple tabs or sections, create individual files for each tab and treat them as components. This modular approach enhances clarity and maintainability.

Suggested Folder Structure

- `project-folder/`
 - `code/`: Additional tools, e.g. Python scripts
 - `public/`: Files you need to be able to access with via url, e.g. java scripts
 - `src/`
 - * `app/`: Store the App and other main components.
 - * `components/`: Reusable components like buttons, cards, and sections.
 - * `data/`: Data sets used for automated components or other elements.
 - * `ic/`: ?? and ?? files. Alternatively, you can use a global definition file.
 - * `pages/`: The files used as page components.
 - * `styles/`: CSS or SCSS files to manage styling consistently.
 - * `utils/`: Type Script and JavaScript functions for global use such as ?? or ??.
 - * `main.tsx`
 - * `navigation.ts`
 - * `pages.ts`
 - * `index files`
 - `README`
 - `??`
 - `LICENSE`
 - `index.html`
 - `gitignore`
 - `pipeline file`

This structure can be extended to, for example, include files for headers, sidebars and references per page.

5.1 Naming Conventions

Implement clear and consistent naming conventions for files and folders to enhance navigation and understanding within the project.

- **PascalCase** for props, states and components: *UserProfile*, *Navbar*, *ItemList*.
- **camelCase** for variables: *wordCount*, *primaryColor*
- **Verb-Noun with context** for functions: *UseNavbar*, *CreateSidebar*
- **Lowercase with Hyphens** for classes and ids: *experiment-header*, *first-button*
- **Context** for all naming: *Timeline*, *InfoBoxes*, *createSidebar*

Chapter 6

Website and page structures

6.1 Standard URL Pages

iGEM provides standard URL pages that partly correspond to deliverables required for medal eligibility. These pages should be carefully curated to meet specific criteria, and judges will often look at them first. Here are the essential pages you must include:

Bronze Medal Criteria

- **Attributions:** This page outlines the contributions of each team member and any external help you received.
 - <https://year.igem.wiki/example/attributions>
- **Project Description:** A detailed description of your project, summarizing the scientific question you are addressing.
 - <https://year.igem.wiki/example/description>
- **Contribution:** Document how your team contributed to the iGEM community, whether through improved protocols, new parts, or educational resources.
 - <https://year.igem.wiki/example/contribution>

Silver Medal Criteria

- **Engineering Success:** Outline how your project follows the engineering design cycle, including design, build, test, and learn phases.
 - <https://year.igem.wiki/example/engineering>
- **Human Practices:** Demonstrate how your project interacts with and is shaped by the broader societal, ethical, and environmental context.
 - <https://year.igem.wiki/example/human-practices>

Gold Medal Criteria

For Gold, your team needs to document work for Special Prizes. The structure of these pages may vary depending on the prizes you select.

These standard URL pages are automatically linked to your Judging Form, so it's essential that all relevant achievements are documented here to ensure proper judging and recognition.

6.2 Additional Pages and Considerations

While the Standard URL Pages cover core requirements, your team may need to include additional pages to give a full picture of your project. Consider adding separate pages for the following:

- **Team Page:** Introduce the team members, their roles, and their contributions. This humanizes your project and gives credit to everyone involved.
- **Lab Books:** Document your experiments and lab work. Each experiment or set of related experiments can have its own section for better clarity.
- **Meeting Documentation:** Keep track of all important team meetings, especially those related to project decision-making, collaborations, and interactions with stakeholders.
- **Sponsors and Partners:** List all your project sponsors, partners, and external supporters. This can be a good way for acknowledging specific contributions and maintaining transparency. If you have official sponsoring contracts, you can offer the sponsor page as a way of advertising to your sponsors.
- **Materials and Methods:** Naturally, you need to provide detailed information about the techniques, equipment, and protocols used in your project for both judges and future teams.
- **Results:** Summarize your findings and data analysis to give the user overviews and to offer easy entry into every aspect of your project. Depending on your results' complexity, this may warrant its own section or multiple pages.
- **Parts:** List the biological parts you designed or used in your project, following iGEM's standards for documenting parts.
- **Sustainability:** Include this page to highlight how your project contributes to sustainable development goals or impacts the environment positively.

6.3 Structuring for Easy Navigation

It's important to ensure that every aspect of your project can be reached through the standard URL pages. For example, your *Human Practices* page can link to more detailed reports or outcomes in related

sections such as *Sustainability* or *Ethics*, but make sure the essential content is summarized within the *Human Practices* page itself.

To ensure easy navigation and avoid clutter, consider using clear menus and submenus that guide users through different sections. Items such as *Proof of Concept* or *Design* can be standalone pages but should also be cross-referenced in key standard pages such as *Engineering Success* to ensure judges can find relevant information quickly.

6.4 Enhancing the Navigation Bar

The navigation bar is one of the most crucial elements of your wiki, guiding users through the content and making sure they can quickly find what they need. Here are some key considerations to ensure your navigation bar is both functional and user-friendly:

- **Linking Sections, Not Just Pages:** It's important to know that your navigation bar doesn't have to link only to full pages. You can also link directly to specific sections within a page. This is especially useful if you have long pages with different subtopics. For example, in a dropdown menu for "Project", you could link to the "Design" section of the main project page without needing to create a separate page for it. This saves time and avoids unnecessary fragmentation of content while ensuring that all important sections are easily accessible.

```
{  
  name: "Project Description",  
  title: "Project Description",  
  path: "/description?scrollTo=Abstract"  
}
```

See full code example in the appendix on page ??.

- **Creating a 'Highlights' Dropdown:** A smart way to organize your navigation bar is by adding a Highlights dropdown, which can give users a quick tour of the key content. This is ideal for judges or visitors who want to get an overview without navigating through each section individually. Include the most critical pages like:
 - Project Description
 - Human Practices
 - Engineering Success
 - Contribution

This offers a streamlined navigation experience, allowing users to quickly jump to the core of your project.

- **Page and Folder Highlighting:** For enhanced user experience, it's a good idea to highlight the current page or folder that the user is on within the navigation bar. This visual cue helps users understand where they are within the site structure. Many modern websites use a simple color change or underline to show which menu item is active. This is especially useful when users are navigating through multiple pages or sections, preventing them from getting lost in a large wiki structure.
- **Dropdown Menus for Section and Page Links:** Dropdowns can help organize both pages and sections under a single heading. For example, if you have a "Project" dropdown, it can contain links to separate pages like *Project Description*, while also linking directly to sections of a single page, like *Design* or *Results*. This ensures that teams don't feel pressured to create separate pages just to mention something on the navigation bar—allowing a better balance between organization and simplicity.

Ensure that your navigation bar includes direct links to all key sections such as *Project Description*, *Engineering Success*, and *Human Practices*.

Also consider:

- **Clear Labeling:** Each item in the navigation bar should have a clear and descriptive label that indicates the content of the page or section it links to. Avoid using jargon or abbreviations that might confuse users. For example, instead of using "Proj. Des.", use "Project Description."
- **Consistent Structure:** Maintain a consistent structure throughout the navigation bar. For example, if you use dropdowns for one category, ensure that all categories are similarly organized. This consistency helps users understand how to navigate your site intuitively.
- **Mobile Responsiveness:** Ensure that the navigation bar is responsive and functions well on mobile devices. This may involve using a hamburger menu for smaller screens, where users can click to expand the menu. Test the navigation on various devices to confirm usability.
- **Search Functionality:** If your wiki has a lot of content, consider incorporating a search bar in the navigation area. This allows users to quickly find specific information without having to browse through multiple pages.
- **Breadcrumb Navigation:** Implement breadcrumb navigation to provide users with context about their location within the site. This shows the path they took to arrive at a specific page, allowing them to backtrack easily.
- **Accessibility Features:** Ensure that the navigation bar is accessible to all users, including those with disabilities. Use appropriate color contrasts, keyboard navigation support, and ARIA (Accessible Rich Internet Applications) roles for assistive technologies.
- **Sticky Navigation:** Consider making the navigation bar sticky (fixed at the top of the page as the user scrolls). This keeps essential navigation options visible, improving user experience, especially on long pages.

- **Dropdown Indicators:** Use visual indicators (like arrows or plus signs) to show which menu items contain dropdown options. This helps users understand that more content is available under certain categories.
- **Highlighting Active Links:** Highlight active links (the current page) in a way that distinguishes them from other links. This could involve changing the text color or background color to indicate which page is currently being viewed.
- **Limited Number of Items:** Try to limit the number of top-level items in the navigation bar to avoid clutter. A clean, concise navigation menu is easier for users to navigate than one that is overloaded with links. If you have many categories, consider grouping them logically or using dropdown menus.
- **User Testing:** After designing the navigation bar, conduct user testing with team members or potential users to gather feedback on usability. This can help identify areas for improvement before finalizing the design.
- **Linking to External Resources:** If your project references external resources, ensure that links to these sites open in a new tab. This prevents users from losing their place on your wiki while exploring additional content.

By implementing these features in your navigation bar, you'll create a smoother and more intuitive browsing experience that can help judges and other users easily navigate your content and quickly access essential information.

6.5 Footer

Be sure to always include the obligatory parts in the footer such as the licensing information and link to your team's GitLab Repository.

- **Licensing Information:** It is essential to mention the licensing terms for your content in the footer.
- iGEM requires that all content be available under the Creative Commons Attribution 4.0 license or any later version, and this should be clearly stated.
- **GitLab Repository Link:** Include a visible link to your team's GitLab repository in the footer, as mandated by iGEM. This allows judges and future teams to easily access the source code used to generate the wiki.

Apart from the above parts, the footer design is quite flexible. Possible additions include:

- **Sponsor Links:** Many teams choose to include links to their sponsors in the footer, often displayed

in a slider or as static images. This not only acknowledges support but also provides visibility for the sponsors.

- **Contact Information:** Some teams may include an Impressum (legal disclosure) and contact information in the footer. This typically consists of email addresses or physical addresses.
 - **Email Address Security:** If you choose to display email addresses, be mindful of security concerns. Simply listing an email address can expose it to spam bots. To mitigate this, consider using a contact form or disguise the email (e.g., using “teamname [at] domain [dot] com”).
- **Navigation and Additional Links:** Consider including quick links to important pages or sections within the wiki in the footer, enhancing user navigation.
- **Social Media Links:** If applicable, teams may also choose to link to their social media profiles to engage visitors further and provide updates on their project.

By thoughtfully incorporating these elements into the footer, you can create a functional and informative section that enhances user experience while fulfilling the necessary requirements set forth by iGEM.

6.6 Header

Page headers are the first interaction point for users on your iGEM wiki. They should clearly display the page name and set the tone for the content. If you are aiming for best wiki, consider that engaging visuals, such as images or videos, can enhance the user experience while maintaining consistency with your team’s branding and thoughtful headers create an inviting atmosphere that encourages exploration.

- **Clear Page Identification:** Every page header should clearly display the page name. This helps users quickly identify the content they are viewing and enhances navigability.
- **Visual Style:** Headers can vary in style, ranging from simple text to more elaborate designs. Some common styles include:
 - **Flashy Headers:** Utilizing bold graphics, vibrant colors, or animations to grab attention.
 - **Photographic Headers:** Incorporating relevant images that reflect the page content, creating an engaging visual experience.
 - **Video Headers:** Using short video clips or animations as headers can add dynamic content and enhance user engagement.
- **Call to Action:** If appropriate, headers can include calls to action (CTAs) to encourage users to engage further with the content, such as “Learn More” or “Get Involved.”
- **Alignment with Content:** The style and imagery used in the header should align with the content of the page, providing context and setting the tone for what follows.

By thoughtfully designing page headers with these considerations in mind, you can create an inviting and informative entry point for users that enhances their overall experience on your iGEM wiki.

6.7 Additional Elements

Integrating extra elements into your iGEM wiki can significantly enhance navigation and user engagement. These features improve the overall experience by making content more accessible and encouraging exploration, creating a more inviting and informative environment for visitors.

- **Sidebar Navigation:** A sidebar can significantly improve navigation, especially for long pages. It should include only H1 and H2 headings, allowing users to quickly jump to different sections. Incorporating a highlight feature can indicate the current section the user is viewing.
- **”Back to Top” Button:** This button can be placed at the end of the page or within the sidebar, providing an easy way for users to return to the top without scrolling manually.
- **Cards:** Consider adding cards at specific positions, such as the end of each page, to engage users and encourage them to explore related content. These can highlight important links or additional resources.
- **Scrolling Progress Bar:** A visual scrolling progress bar can indicate how far the user has scrolled down the page, giving them a sense of navigation and encouraging continued reading.
- **Search Functionality:** Incorporating a search bar allows users to quickly find specific content or sections within the wiki, enhancing usability.
- **Suggested Reading:** Including a section for related links or suggested reading at the end of pages can guide users to additional relevant content, e.g. guiding the reader from design to engineering.

Final Thoughts

When structuring your iGEM wiki, prioritize user experience, especially from the perspective of judges and future teams. Keep navigation simple and intuitive, ensuring that all critical deliverables are well-documented and easy to locate. By maintaining clarity and organization across your pages, your team will be well-positioned for success in the competition.

Chapter 7

Media

Upload pending.

Chapter 8

Coding prerequisites

8.1 Hardware

The most obvious requirement is a laptop or computer. In my experience, laptop-tablet 2-in-1 devices can be difficult to work with due to limitations in handling coding tasks. Additionally, not all laptops can smoothly run the required coding environments. We have encountered common issues with Windows laptops, particularly around scripts not being allowed to run or installations being blocked. On personal devices, these problems can usually be fixed (with an internet search), but on university-provided laptops, users may not have the administrative rights needed to make necessary changes.

I recommend to address these issues on debut, as deferred tasks tend to not get done in iGEM projects and trying to fix them last minute may lead to delays that are difficult to recover from.

8.2 GitLab Access and Cloning the Repository

To contribute to the Wiki, every team member needs to clone the repository from the iGEM GitLab. This requires each member to have their iGEM account credentials (username and password) ready. I recommend using SSH cloning for a more secure and stable connection.

HTTPS Cloning: This method requires users to enter their iGEM credentials (username and password) every time they push or pull code. It is straightforward but can become annoying due to repeated logins.

- **Advantages:** Easy to set up and doesn't require additional software.
- **Disadvantages:** Requires entering credentials frequently, less secure than SSH.

SSH Cloning: This method uses an SSH key to authenticate, avoiding the need to repeatedly enter a username and password. It is generally considered more secure and efficient.

- **Advantages:** No need for constant credential input, more secure, ideal for frequent use.
- **Disadvantages:** Requires setting up an SSH key and ensuring the necessary software is installed.

Setting Up SSH Cloning:

For SSH cloning, each team member needs an **SSH key generator** installed. You can check if your system has this by typing **ssh** in the terminal. If the system does not recognize the command, you'll need to install an SSH key generator that fits your operating system.

It is best to explain to the team what an SSH key is: it is a pair of cryptographic keys used for secure access. Always remember to **only use and share the public key**, never the private key. The public key typically has the file extension **.pub**.

Note: On some devices, **.pub** files may attempt to open with Microsoft Publisher or produce an error about missing software. You can simply open the file using the “Open with...” option and choose a text editor.

8.2.1 Step-by-Step explanation

1. **Check for SSH Key Generator:** Open your terminal and type **ssh** to check if the key generator is installed. If not, install it (e.g., OpenSSH for Windows or macOS/Linux).
2. **Generate the SSH Key:** In the terminal, enter the command to generate the SSH key (e.g. **ssh-keygen**).
You will be asked:

Enter file in which to save the key (/home/user/.ssh/id_ed25519):

Press **Enter** to accept the default file name or enter a different name to create a new key if the file already exists.

3. **Set a Passphrase (Optional):**
At the prompt:

Enter passphrase (empty for no passphrase):

Choose to set a passphrase or press **Enter** to skip. You will be prompted to confirm the passphrase.

4. **View Key Information:** After generating the key, the terminal will display the file name, its fingerprint, and a randomart image of the key.
5. **Log into iGEM GitLab:** Go to iGEM GitLab and log in with your credentials.
6. **Access SSH Key Settings:**
 - Click on your profile picture (top right) and select **Preferences**.
 - This will take you to the **User Settings** page.

7. Add New SSH Key:

- On the left, click **SSH Keys**.
- Then, click **Add new key**.

8. Copy Your Public Key:

- Open the public key file you just generated (it typically ends with `.pub`).
- *Note:* You may need to reveal hidden files to access the `.ssh` folder.
- Copy the entire content of the public key file. It will start with `ssh-ed25519` or `ssh-rsa`.

9. Paste the Key in GitLab:

- Paste the public key content into the **Key** box on GitLab.
- Add a **Title** (e.g., "My Laptop Key").
- Click **Add key** to save it.

After completing these steps, your SSH key will be added, and you can now clone repositories via SSH!

For that, install the necessary software as described in the following section.

8.3 Coding Environment (IDE) and Git

I recommend every team member installing a coding environment (IDE) to work efficiently. Popular choices include **Visual Studio Code** or **IntelliJ IDEA**.

While it's possible to edit files directly on GitLab through:

- **The Text Editor:** No helper tools to catch errors.
- **The Web IDE:** Limited error-checking capabilities.

A standard IDE provides (better) error detection, auto-completion, and other productivity tools that make coding more efficient and less error-prone.

But an IDE alone does not suffice. To start working with Git, you'll need to install it on your system and configure it with your username and email, which will identify your contributions.

1. Install Git : Either download the installer from <https://git-scm.com/downloads> and follow the setup instructions depending on your operating system or google a way to install it via the command line.

2. Set Up Username and Email : After installing Git, configure your username and email to be used in your commits. This ensures that every commit you make is properly attributed to you.

1. Open a terminal or command prompt.
2. Set your username by running:

```
git config --global user.name "Your Name"
```

3. Set your email address by running:

```
git config --global user.email "your.email@example.com"
```

You can check if your Git username and email are set correctly by running:

```
git config --global --list
```

8.4 Cloning your repository

After installing Git and an IDE, you are ready to clone your repository.

1. Go to Your Project Page on GitLab: Navigate to the iGEM GitLab project page you want to clone.

2. Open the Code Menu: On the project page, click on the blue **Code** button near the top of the screen.

3. Select "Open in Your IDE": From the dropdown, select **Open in your IDE** for simplicity. You will be presented with options to select between different IDEs. Choose either:

- Visual Studio Code (SSH)
- IntelliJ IDEA (SSH)

4. Choose a Folder: A file explorer will open, prompting you to select a folder on your local machine where the repository will be saved. I advise against saving the folder in a cloud. Git already backs your work up, if you push (upload it).

Trust the Authors (if prompted): If a pop-up appears asking whether you trust the project's authors, confirm by selecting **Trust** to ensure you can edit the files without restrictions.

8.5 Other Software

Depending on the type of project in your repository, you will need to install specific package managers and dependencies. For React projects, some of those will already be listed in your `package.json`, additional ones you need to install as the necessity arises.

This will be covered in React.

Additional Considerations

Regardless of the project type, be aware that there may be additional packages, scripts or software required depending on what is already installed on your device. The error messages you encounter will usually indicate what is missing.

Further, you might need to think about:

- **Using the Correct Shell:** Some commands might differ based on the shell you're using (e.g., bash, zsh, cmd, PowerShell). Ensure you are using the correct commands for the terminal of your operating system.
- **using terminals as an administrator** Some changes or installations can only be made if you do them as an administrator. It can differ from operating system to operating system how you install things as an administrator.
- **Common Dependency Errors:** If you encounter errors related to dependencies, search for the specific error message online, as there may be community solutions or discussions about the issue.
- **File Permissions:** On some operating systems, particularly Linux and macOS, you may need to adjust file permissions for certain operations. That does not automatically mean you should not do the operation.

Chapter 9

General Coding

Upload pending.

Chapter 10

React

10.1 Getting started with your React project

Updating Package Managers: If you already have `npm` or `yarn` installed, consider updating them to the latest version to avoid compatibility issues. You do not need both package managers, one should be sufficient. If you are unsure, read about the differences between the managers to make a decision.

Otherwise **Install Package Mangers:**

- **node:** Make sure you have Node.js installed. You can check by running `node --version`. If it's not installed, download it from the Node.js website.
- **npm:** It should come bundled with Node.js.
- **yarn:** If you don't already have yarn installed, you can usually do so by running the following command:

```
npm install --global yarn
```

Install Project Dependencies: Navigate to your project directory in the terminal and run:

```
yarn install
```

or

```
npm install
```

Ensure Compatible Node.js Version: Make sure your Node.js version is either `^8.18.0` or `>=20.0.0` to avoid compatibility issues, with 18.20.0 being the recommended version in 2024.

Chapter 11

Additional tools

Upload pending.

Chapter 12

Troubleshooting

Upload pending.

Chapter 13

Guide for team

Upload pending.

Chapter 14

The Wiki Freeze

Upload pending.

Chapter 15

The Wiki Thaw

Upload pending.

Associated software

Bibtex Parser

COMPLETED · package – script– component

Webpack parser script usable as TypeScript and React component

<https://github.com/liliana-sanfilippo/bibtex-ts-parser>

Author names Parser

IN PROGRESS · package – script– component

Name parsing grammar for ANTLRts, intended to be used to parse names of authors in bibtex sources and webpack parser script usable as TypeScript and React component.

<https://github.com/liliana-sanfilippo/author-name-parser>

Reference generator component

IN PROGRESS · package – component

TypeScript and React component using bibtex and nam parser to automatically parse references for wiki pages or scientific blogging.

<https://github.com/liliana-sanfilippo/react-bibtex-source-generator>

React Library

IN PLANNING · package – component

Library of TypeScript and React components aimed at iGEM wikis.

<https://github.com/liliana-sanfilippo/igem-react-lib>

Attributions

L^AT_EXcode

Layout

Rafael Kümmel

CC BY 4.0 · Apostila Template CP-USP-2021

Overall document layout was adapted from Kümmels template though the code changed and added to.

www.overleaf.com/latex/templates/

`apostila-template-cp-usp-2021/tfcwfrsgzhbz`

Attribution table

Sarah Lang

CC BY 4.0 · Modern Simple CV

Table layout was adapted.

<https://www.overleaf.com/latex/templates/modern-simple-cv/>

`kwrxbwthgrwr`

Illustrations



Mitochondria
JAIGANESH · CC0
https://bioicons.com/icons/cc-0/Intracellular_components/jaiganesh/mitochondria.svg



Animal mitochondrion diagram unlabelled
PUBLIC DOMAIN · LadyofHats
https://commons.wikimedia.org/wiki/File:protect\protect\leavevmode@ifvmode\kern+.2222em\relaxAnimal_mitochondrion_diagram_unlabelled.svg



Simple Mitochondria Rod
MARNIE-MADDOCK · CC-BY 4.0 Unported
https://bioicons.com/icons/cc-by-4.0/Intracellular_components/Marnie-Maddock/simple_mitochondria_rod.svg



Mitochondrium 6
SERVIER · CC-BY 3.0 Unported
https://bioicons.com/icons/cc-by-3.0/Intracellular_components/Servier/mitochondrium-3.svg

Upload pending.