



**POLITECNICO**  
**MILANO 1863**

# TECHNOLOGY REPORT

## HYPERMEDIA APPLICATIONS

Prof. Garzotto Franca – A.Y 2023/2024  
17/07/2024

[Helping Hands Website](#)

[GitHub Repository](#)

Submitted by  
*Thinking4heads*

Filippo Fini  
Matteo Fiorentino  
Luca Longinotti  
Davide Pessi

filippo.fini@mail.polimi.it  
matteo.fiorentino@mail.polimi.it  
luca.longinotti@mail.polimi.it  
davide.l.pessi@mail.polimi.it

Index

1    **WORK ORGANIZATION**..... 3

2    **OUR PROJECT**..... 4

    2.1    PROJECT EXPLANATION ..... 4

    2.2    CHOSEN THEME ..... 4

    2.3    HOSTING SERVICE ..... 4

    2.4    PROJECT STRUCTURE ..... 5

        2.4.1    *Pages* ..... 5

        2.4.2    *Server Endpoints*..... 6

    2.5    COMPONENTS IMPLEMENTED ..... 7

    2.6    EXTRA MODULES .....10

    2.7    ACCESSIBILITY AND SEO.....10

        2.7.1    *Accessibility* .....10

            2.7.1.1    *Responsiveness*.....10

        2.7.2    *SEO*.....10

    2.8    EXTRA FUNCTIONALITIES .....11

        2.8.1    *Sorting* .....11

        2.8.2    *Contact Form* .....11

        2.8.3    *Donation Form*.....12

        2.8.4    *Story Form*.....12

# 1 Work Organization

We divided our work by assigning specific pages to each member along with some broad guidelines; then we collaborated to unify the style and exchange ideas for improving each other's work. This process was done at each step, ensuring that each of us became familiar with every aspect of website development while maintaining the style as consistent as feasible.

Members	Main Areas of Work	% of Global Effort
Fini Filippo	UX Design, Low-Fidelity Wireframes, DB Design, DB Data Insertion, Implementation, Styling, Report creation, Chatbot implementation, Chatbot Prompt, Extra Functionalities, Interaction Scenarios.	25%
Fiorentino Matteo	UX Design, Low-Fidelity Wireframes, DB Design, Implementation, Styling, Chatbot Prompt, Accessibility, SEO, Report creation, Interaction Scenarios.	25%
Longinotti Luca	UX Design, Low-Fidelity Wireframes, DB Design, Implementation, Styling, Chatbot Prompt, Extra Functionalities, Report creation, Interaction Scenarios.	25%
Pessi Davide	UX Design, Low-Fidelity Wireframes, DB Design, DB Data Insertion, Implementation, Styling, Report creation, Accessibility, Chatbot Prompt, Extra Functionalities, Interaction Scenarios.	25%

## 2 Our Project

### 2.1 Project Explanation

The project was about the creation of a website for a no-profit center that offers support to women in difficulty and involved in situations of violence. The given name is *Helping Hands*. The project has been implemented using the Vue3 and Nuxt3 frameworks. The final website consists of several pages and for most of them the content is retrieved from the database.

### 2.2 Chosen Theme

Our website's theme is about women's violence. We focused on educating victims about various types of violence as well as providing them with tools and resources. Given our website's target audience, we chose warm, reassuring colors, and all the photos convey a sense of community and affection.

### 2.3 Hosting service

The platform we adopted to host our website is *Vercel*, in combination with *Supabase* to store our data Postgres DB. This solution allowed us to create a more dynamic website, with respect to the GitHub Pages alternative.

We opted for SSR (Server-Side Rendering) since we believe that the server on which we are relying is very powerful and we wanted to lighten processes on client machines to allow every kind of device to access the website.

Moreover, Vercel is integrated with Git, so the project is automatically deployed every time there are changes to the repository.

## 2.4 Project Structure

### 2.4.1 Pages

Our website is structured in the following way:

- index.vue
- Activities/projects/[id].vue
  - /index.vue
  - /services/[id].vue
    - /index.vue
  - /index.vue
- ask\_for\_help/for\_another/index.vue
  - /for\_you/index.vue
  - /index.vue
- our\_team/[id].vue
  - /index.vue
- support\_us/donate/index.vue
  - /volunteer/index.vue
  - /index.vue
- about\_us.vue
- contact\_us.vue

**Index.vue:** the homepage of our website. It is linked to the most relevant pages. It contains both textual content and images.

**Activities/projects/[id].vue:** page for each individual project at our center. It includes the image, supervisor, team, year, description, and a link to the project supervisor's page. Users can navigate to the next and previous projects, ordered by project\_id in the database, or to one of the three services displayed at the bottom of the page, which were randomly selected from the list of projects.

**Activities/projects/index.vue:** page that displays a list of all the center's projects (each described by its image, title and short description), which are ordered alphabetically by default. Users can choose between four sorting criteria: *alphabetically*, *alphabetically reversed*, *newest first*, and *oldest first*. If a user clicks on a project, he is directed to the project's specific page.

**Activities/services/[id].vue:** page for each individual project at our center. It includes the image, supervisor, team, service schedule, description, and a link to the supervisor's page. Users can navigate to the next and previous services, which are ordered by the service\_id in the database, to one of the three services displayed at the bottom of the page that were randomly selected from the list of services, or to the page '*contact\_us*'.

**Activities/services/index.vue:** page that displays a list of all the center's services (each described by its logo and title), which are ordered alphabetically by default. Users can choose between two sorting criteria: *alphabetically* or *alphabetically reversed*. If a user clicks on a service, he is directed to the service's specific page.

**Activities/index.vue:** page that introduces projects and services. Aside from the textual content, it includes a link to '*Projects*' and '*Services*'.

**Ask\_for\_help/for\_another/index.vue:** this page contains all the information that a third party, such as a friend or family member, needs to assist a woman who has been the victim of violence. Aside from the textual content, it includes images and links to the '*Services*' and '*Contact\_us*' pages.

**Ask\_for\_help/for\_you/index.vue:** this page contains all the information required by a woman who has been a victim of violence, including an explanation of the various types of violence and how to seek assistance. In addition to the textual content, it contains images and links to the '*services*' and '*contact\_us*' pages.

**Ask\_for\_help/index.vue:** this is an introductory page on the topic of assisting a victim of violence. Aside from the textual content, it includes images and links to the '*for\_you*' and '*for\_another*' pages.

**Our\_team/[id].vue:** is the personal page of the center's members. It includes their picture, CV, personal information, a brief description of the employee and two clickable carousels displaying the services and projects for which he or she is a supervisor.

**Our\_team/index.vue:** page displaying a list of all the center's employees (each described by their picture, name, and role), ordered by role as the default setting. Users can choose between three sorting criteria: *alphabetically*, *alphabetically reversed*, and *by role*. If a user clicks on an employee, he is directed to the employee's personal page.

**Support\_us/donate/index.vue:** this page allows users to support the center by making donations. It includes textual content, an image, and a form where users can donate selecting a fixed amount or a custom import.

**Support\_us/volunteer/index.vue:** users can find information about how to volunteer at the center on this page. It contains images with textual content and a button that takes the user to the '*contact\_us*' page.

**Support\_us /index.vue:** this is an introductory page that explains how a user can help the center. Furthermore, there is a form where a testimonial can share their story. Aside from the textual content, it includes links to the '*donate*' and '*volunteer*' pages.

**About\_us.vue:** this is an introductory page that describes the center's history, a quick introduction about the solutions it offers, and a brief overview of its staff. It contains images with textual content and a button that takes the user to the '*team*' page.

**Contact\_us.vue:** page with all relevant contact information for the organization (phone number, email, map with headquarters location, and written directions). A form is also available for users to submit feedback, messages or contact the center.

## 2.4.2 Server Endpoints

**GET api/our\_team/[id]:** endpoint that retrieves all the information of a specific person. Their id is passed as a URL parameter. The response is either a person Object (with their id, name, surname, phone number, email, cv link, role, image and description) or an error with its message.

**GET api/our\_team/index:** endpoint that retrieves the full list of people working at the association. The response is either an Object containing all the people (with their id, name, surname, role and image), ordered role, or an error with its message.

**GET api/projects/[id]:** endpoint that retrieves the details of a project. Its id is passed as a URL parameter. The response is either a project Object (with its id, date, description, image, title as well as the respective members involved, as supervisor or team) or an error with its message.

**GET api/projects/index:** endpoint that retrieves all the projects. The response is either an Object containing them all (with their id, title, year and small description), ordered alphabetically by title, or an error with its message.

**GET api/services /[id]:** endpoint that retrieves the details of a service. Its id is passed as a URL parameter. The response is either a service Object (with its id, description, image, small description, title, schedule as well as the respective members involved, as supervisor or team) or an error with its message.

**GET api/services/index:** endpoint that retrieves all the services. The response is either an Object containing them all (with their id, title, small description and image), ordered alphabetically by title, or an error with its message.

**GET api /peopleCount:** endpoint that retrieves the total number of people. The response is either an integer number or an error with its message.

**GET api /projectOfPers:** endpoint that retrieves all the persons and the projects in which they are involved. The response is either an Object containing them all (with is supervisor, person id, project title, project image and its id) or an error with its message.

**GET api /serviceOfPers:** endpoint that retrieves all the persons and the services in which they are involved. The response is either an Object containing them all (with is supervisor, person id, service title, service image and its id) or an error with its message.

**GET api /testimonials:** endpoint that retrieves all the testimonials of the services. The response is either an Object containing them all (with service id, name, image and quote) or an error with its message.

**POST api /chat:** endpoint that allows to interact with OpenAI Assistant. Receives a text message as input, retrieves the assistant, creates/uses a thread, sends message to thread conversation and gets a response. Returns either a response text or an error message.

## 2.5 Components Implemented

To reduce the redundancy of code, we decided to implement the following Vue components:

- **BackButton:** button that appears in every page, except for the Homepage, that allows the user to go back to the previously visited page.
- **BandDoubleDesc:** consists of two bands, one on top of the other, each composed of a title, a description, and a ButtonViolet. There are no props required because we only used this component on the 'support\_us' page.
- **BandForm:** contains a band that covers the entire width of the page. On the left of the band, there are a title placed above a description; below the description there is the mailing address of the center and the links to its social media; on the right side of the band there is instead the '**contactForm**' component.
- **BandLeftImg:** contains a band that spans the entire width of the page with an image on the left of the band, an optional text on the right of the image and a ButtonViolet at the end of the text centered on the right side of the band. Props: '**description**' text that will appear on the right, '**namebutton**' text that will appear on the button, '**imageName**' link to the image that will appear on the left, '**link**' link to the page of the button, '**aria\_label**' aria label of the button.
- **BandLeftNoButton:** contains a band that spans the entire width of the page with an image on the left of the band and an optional text on the right of the image. Props: '**description**' text that will appear on the right, '**imageName**' link to the image that will appear on the left.

- **BandMap:** contains a band that spans the entire width of the page. On the left side of the band, there is a map showing the center's position; on the right, there is a title placed on top of text.
- **BandRightImg:** contains a band that spans the entire width of the page with an image on the right of the band, an optional text on the left of the image and a ButtonViolet at the end of the text centered on the left side of the band. Props: 'description' text that will appear on the left, 'namebutton' text that will appear on the button, 'imageName' link to the image that will appear on the right, 'link' link to the page of the button, 'aria\_label' aria label of the button.
- **BandRightNoButton:** contains a band that spans the entire width of the page with an image on the right of the band and an optional text on the left of the image. Props: 'description' text that will appear on the left, 'imageName' link to the image that will appear on the right.
- **BandTripleButton:** contains a band that spans the entire width of the page with an image on the right, an optional text on the left of the image and three ButtonViolet at the end of the text centered on the left side of the band, equally spaced. Props: 'description' text that will appear on the left, 'namebutton1' text that will appear on the first button, 'link1' link to the page of the first button, 'namebutton2' text that will appear on the second button, 'link2' link to the page of the second button, 'namebutton3' text that will appear on the third button, 'link3' link to the page of the third button, 'imageName' link to the image that will appear on the right, 'aria\_label' aria label of the three buttons.
- **BreadCrumbs:** this component allows the user to determine which section of the website he is on and navigate to higher levels of the website architecture. It is present on all pages except the homepage. Props: 'crumbs' is an array in which each member is a tuple (label, URL) that, when processed by the component, will become a NuxtLink with that name (label) and path (URL); each member of the array will be separated from the others by a '>' symbol.
- **ButtonViolet:** It's based on a NuxtLink and serves as the template for most buttons used. Props: 'name\_button' the text displayed on the button, 'link' link of the page to jump to, 'aria\_label' aria label of the button.
- **Chatbot:** component that allows users to interact with the OpenAI assistant. When not activated, the bot appears as a button in the bottom right corner of the screen. When activated, it opens a chat window where users can see the bot and his/her messages, write messages in the designated box, and send messages using the dedicated button or the 'enter' key on the keyboard. When a new message is sent, the 'sending message' text is displayed. When a response is received, the text is either removed or replaced with an error message. When switching to the mobile version, the open chat fills the entire screen.
- **ContactForm:** contains a form used by users to contact the center, which has fields for name, surname, phone number, email, and text message, as well as a submit button. To submit the message to the center, the user must first compile all the form's fields; until then, the submit button will be disabled. Furthermore, the component checks to see if the email address and phone number provided are of the correct format and, if not, notifies the user via a message on the screen.
- **Donationform:** contains a form that allows people to donate to the facility. The form consists of some buttons indicating fixed donation amounts, a field where the user can enter a custom donation amount, and by a donate button that is disabled until an amount, either custom or fixed, is selected.
- **DropdownList:** includes a drop-down list that fills the page's whole width. Every item in the list has a title and a description, which is only shown when the element is clicked.



- **Footer:** the bottom bar that appears in every page of the website, containing the organization's phone number, email, headquarter address and social media profiles.
- **ImageUp\_textDown:** component that contains an image placed on top of a text composed of a title and a description. Props: '**description**' text that will appear on the bottom of the component, '**imageName**' link to the image that will appear on the top, '**title**' title that will be placed between the description and the image.
- **NetflixCarousel:** component that contains a title, and a list of cards, which could be projects or services, each of which has an image of the activity and a title; clicking on a card takes you to the appropriate page. It is used to highlight the services and projects that the person oversees within the single person page. Props: '**id**' person id, '**act**' identify if the list is of projects or services.
- **PersonOv:** component containing the team member's picture, full name, and role. It is also used to display testimonials; in this case, the component is not clickable and includes a quote in the role prop. Props: '**id**' person id, '**name**' person name, '**surname**' person surname, '**role**' person role/quote, '**link**' link to the page of the specific person, '**image**' person image, '**testimonial**' 0 if is not a testimonial, 1 otherwise.
- **ProjectCard:** component that contains the logo, the name and a small description of the project or service. Props: '**id**' project or service id, '**title**' project or service title, '**link**' link to the page of the specific project or service, '**year**' year of the project (null in case of service), '**small\_desc**' small description of the project or service, '**image**' boolean value to state whether the image should be displayed or not.
- **StoryForm:** contains a band that spans the entire width of the page with a form through which a user can send their story. It consists of a generic title layered over two title-input field pairs, the first of which allows the user to enter information, and the second of which allows them to provide their testimony.
- **TopBar:** the top bar that appears on every page of the website, containing the clickable company logo and links to the following pages: 'Activities', 'Projects', 'Services', 'Ask for Help', 'Our Team', 'About Us', 'Contact Us' and 'Support Us'. This component is especially responsive because, on smaller screens, all the page links disappear and are replaced with a so-called hamburger menu, which, when activated, opens a sidebar menu with the same entries, making the website mobile friendly.

## 2.6 Extra Modules

In this section we provide a list of all the external modules we installed to improve the website, each one with a short description of how we practically used it in our project:

- *Supabase*: used to create an easily manageable *Postgres Database*.
- *OpenAI*: used to implement the chatbot.

## 2.7 Accessibility and SEO

### 2.7.1 Accessibility

Accessibility is critical for ensuring that everyone can enjoy web content, especially on websites like ours. Specifically, we focused on:

Each page should be easily readable, with text at an appropriate font size and a good contrast between text and background colors.

Adding to each button and linking a label to describe its functionality.

Adding alternative text to each image to help users who are unable to view it understand what it is depicting.

We designed each page and the overall structure of the website to be as understandable as possible, allowing users to find the information they need in the simplest way possible.

To achieve this goal, we used *Lighthouse* and *Wave* to precisely test accessibility and performance.

#### 2.7.1.1 Responsiveness

By specifying every dimension using dynamic metrics and through the use of CSS media queries, we were able to adapt our website to multiple screen sizes, altering the content to better fit the device through which it was viewed.

### 2.7.2 SEO

The goal of our website is to assist as many women as possible; Search Engine Optimization is critical to achieving this goal because it creates the conditions for web crawlers to assign higher scores to such websites.

The techniques we used were:

The addition of title, description, and keywords attributes within the head meta tag, containing information about every aspect of our website.

Addition of various meta tags to improve the appearance of the website link when shared on social media (displaying thumbnail with title and logo).

Addition of descriptive text to every link, avoiding confusing or general words.

We tested that the optimization was done in the best way possible using *Lighthouse*.

## 2.8 Extra Functionalities

We added some features that were not mandatory, but that can have a great impact on website usability.

### 2.8.1 Sorting

In pages with lists, the showing order of the items can be changed to assist users discover what they're looking for more quickly. In Services and Projects, the user can choose whether to display the list of items alphabetically or in reverse order; only in Services is it possible to visualize objects in order of newest first or oldest first.

In Our Team, the list can also be displayed by role.



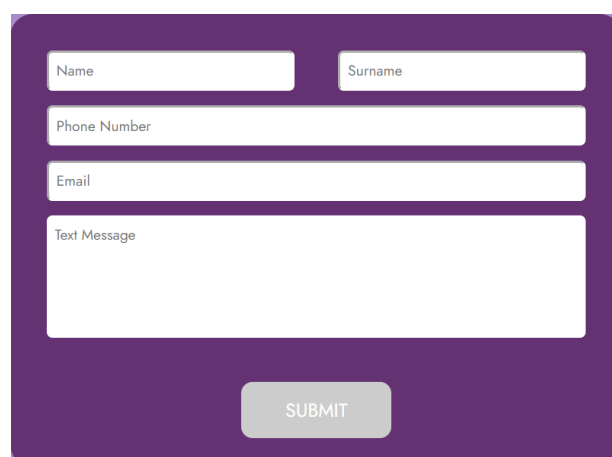
*Figure 1: Example of Sorting Button*

### 2.8.2 Contact Form

In the Contact us page, we added a form, with personal information (Name, Surname, Phone Number, E-mail) and Message fields through which the users can send questions, information requests or assistance requests to the center, and leave their e-mail to be contacted afterwards. The inclusion of first and last name is not mandatory.

All other fields obviously are prevented to be empty, and, moreover, the correctness of Phone Number and E-mail fields is checked by means of regular expressions in the ContactForm component.

Then if they are all suited, the content can be sent correctly.



*Figure 2: Contact Form*

### 2.8.3 Donation Form

In the Donate page, we add a form that allows users to make donations to the center. Each user can choose the amount of money to donate.

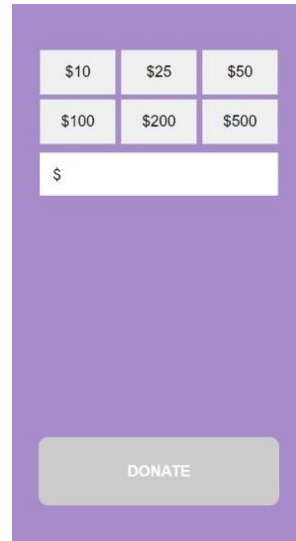
A mobile app interface for a donation form. It features a purple background. At the top, there are six buttons arranged in a 2x3 grid, labeled \$10, \$25, \$50, \$100, \$200, and \$500. Below these buttons is a text input field with a dollar sign (\$) as a placeholder. At the bottom of the form is a large, rounded rectangular button labeled "DONATE".

Figure 3: Donation Form

### 2.8.4 Story Form

In the Support Us page, we add a form that allows users to submit their stories to the center. Each user can choose to send it anonymously or not. The story field, obviously, cannot be empty.

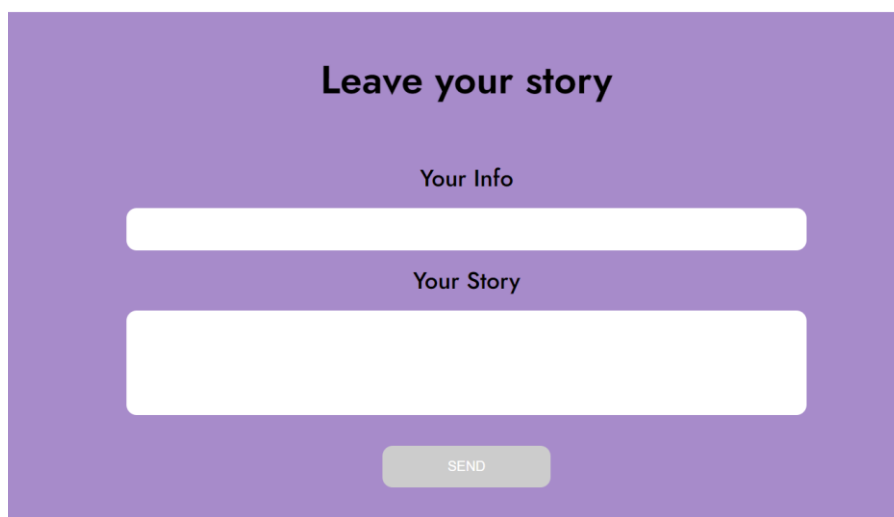
A mobile app interface for a story submission form. It has a purple background. The title "Leave your story" is centered at the top. Below the title is a section labeled "Your Info" with a single-line text input field. Underneath that is a section labeled "Your Story" with a larger, multi-line text input field. At the bottom of the form is a rounded rectangular button labeled "SEND".

Figure 4: Story Form