POLITECNICO
MILANO 1863

Hypermedia Application

Technical Documentation

first version - 17/07/2024

Link to website
Link to GitHub repository

Authors :

Federica Giannunzio (10726268) federica.giannunzio@mail.polimi.it
Isabella Guglielmelli (10701183) isabella.guglielmelli@mail.polimi.it
Léandre Le Bizec (10990212) leandremerlin.lebizec@mail.polimi.it

2024

Contents

1 Introduction

The following documentation describes the main implementation aspects of the website for *Women's Solidarity Center*, a non-profit organization that helps women victims of gender-based violence. The project was implemented using Nuxt3 and Vue3 technologies.

2 Work Organization

The front-end work was evenly split among all group members, while responsibilities regarding the ChatBot implementation and back-end were assigned to different group members, according to their areas of interest. The overall work was split evenly among group members.

Member	Work
Federica Giannunzio	Front-End Implementation, Back-End API Implementation
Isabella Guglielmelli	Front-End Implementation, DB Design and Implementation, Deployment
Léandre Le Bizec	Front-End Implementation, Chat-Bot Design and Implementation

Table 1: Work Organization

3 Hosting Service

To host our website, we decided to use Vercel because of the easy integration with Supabase, which was used for the implementation of a Postgres Database. The use of Vercel in combination with Supabase allowed us to use the Nuxt Serverless Functionality.

3.0.0.1 Rendering Mode

For what concers rendering mode, we decided to use Nuxt with Server Side Rendering (SSR). Our choice ws driven by the advantages that SSR provides, in particular:

- Search Engine Optimization: Pages rendered on the server are more search engine-friendly since the content is already present in the initial HTML sent to the client.
- Improved intital performance.

4 Structure of the project

4.1 Pages

The project was implemented with the following page structure.

```
pages/
    AllActivites/
        OurProjects/
            [id].vue
            index.vue
        OurServices/
            [id].vue
            index.vue
    HelpInformation/
        ForYou.vue
        ForOthers.vue
        index.vue
    OurPeople
        [id].vue
        index.vue
    Contact.vue
    index.vue
    TheCenter.vue
```

4.1.0.1 All Activities/OurProjects/[id].vue

Page for each project of the Center. Each page includes an image that represents the project, its description and a link to the project manager's page.

4.1.0.2 All Activities/OurProjects/index.vue

Introductory page for projects. It includes links to all projects of the Center, and a short description to each.

4.1.0.3 All Activities/OurServices/[id].vue

Page for each service of the Center. Each page includes an image that represents the service, its description, a link to the service manager's page. It also includes a Google Maps link to the center's location, and a list of testimonials of people that were helped by the service.

4.1.0.4 All Activities/OurServices/index.vue

Introductory page for services. It includes links to all services of the Center, and a short description to each.

4.1.0.5 HelpInformation/Foryou.vue

Informative page about what a victim of violence can do to to seek help. It also includes a link to the Help Information page and a link to the For Others page.

4.1.0.6 HelpInformation/ForOthers.vue

Informative page about what a person can do if they know someone that is experiencing gender-based violence. It also includes a link to the Help Information page and a link to the For You page.

4.1.0.7 HelpInformation/index.vue

Informative page about gender-based violence in general and its manifestations. It also includes a link to the For You page and a link to the For Others page.

4.1.0.8 OurPeople/[id].vue

Page for each employee of the Center. Each page includes an image of the employee, a short biography and links to the managed projects and services, with a short description .

4.1.0.9 OurPeople/index.vue

Introductory page for employees. It includes links to all employees of the Center, divided by role.

4.1.0.10 Contact.vue

Informative page about the Center's contacts, such as phone numbers and e-mail. It also includes the Center's location, Google Maps link, and opening hours.

4.1.0.11 index.vue

Homepage of the website. It displays the main purpose of the center, with links to the most important sections of the website, such as Help Information pages, Projects and Services.

4.1.0.12 TheCenter.vue

Informative page about history and mission of the Center.

4.2 Server endpoint

We used Supabase to develop our database. Those are the endpoints that we used to retrieve or publish data on the database.

4.2.0.1 POST /api/addSubscription

Endpoint that inserts a subscription to the corresponding table of the database (the user only inserts fields "name" and "email", while "id" is automatically generated). The response is either the record inserted or an error with its message.

4.2.0.2 POST /api/addTestimonial

Endpoint that inserts a testimonial to the corresponding table of the database (the user only inserts fields “description", “id” is automatically generated and "service_id" is retrieved directly from the context). The response is either the record inserted or an error with its message.

4.2.0.3 GET /api/allProjects

Endpoint that retrieves all the projects. The response is either an Object containing them all (with their id, name, description, img_path, manager_id), or an error with its message.

4.2.0.4 GET /api/allServices

Endpoint that retrieves all the projects. The response is either an Object containing them all (with their id, name, description, img_path, manager_id), or an error with its message.

4.2.0.5 GET /api/allWorkers

Endpoint that retrieves all the projects. The response is either an Object containing them all (with their id, name, surname, role, bio, img_path), or an error with its message.

4.2.0.6 GET /api/person

Endpoint that retrieves the details of a person. Its id is passed as a URL parameter. The response is either a project Object (with its id, name, surname, role, bio, img_path) or an error with its message.

4.2.0.7 GET /api/project

Endpoint that retrieves the details of a project. Its id is passed as a URL parameter. The response is either a project Object (with their id, name, description, img_path, manager_id) or an error with its message.

4.2.0.8 GET /api/service

Endpoint that retrieves the details of a service. Its id is passed as a URL parameter. The response is either a project Object (with their id, name, description, img_path, manager_id) or an error with its message.

4.2.0.9 GET /api/testimonials

Endpoint that retrieves the details of a testimonial. Its id is passed as a URL parameter. The response is either a project Object (with its id, description, service_id) or an error with its message.

5 Components

We developed the website components with a focus on maximizing their reusability across various pages. This approach not only enhances modularity but also improves system usability by presenting users with familiar components throughout their browsing experience.

- **NavBar:** Responsive Navigation Bar with navigation links and dropdown menu.
- **Footer:** Responsive Footer with contact information, social media links and NewsLetter subscription form.
- **Breadcrumbs:** Dynamic breadcrumbs display the path from the Home page to the current page. Pages other than the Home page are links. The path is retrieved from the URL and displayed with some regex rules.
- **ChatBox:** ChatBot using Open AI API, model chatgpt-turbo-3.5. The role of this chatbot is, firstly, to provide assistance to the user during his research on the website. Secondly, to be able to handle emergency situations by calming someone in panic and providing the good resources to the person.
- **Accordion:** Accordion component that displays different types of gender-based violence. Text is shown only on click, when the section is open. The component takes as props an array of items, each item has a title and a text attribute.
- **PersonCard:** Card component that represents an employee. It shows the employee's name, surname, role and image. The component takes as props all attributes of a Person, such as name, surname, role, id, img_path.
- **ActivityCard:** Card component that represents an activity in general. It shows the activity's name and description, clamped after 6 lines. The component takes as props the activity's name, description and path to respective service or project page.

6 ChatBot logic

- **Plugin:** openAI API v1 is added in the project as NuxtPlugin to allow request to the API with the key passed in the header of the request. Here is a list of the different function used in the chatBot component and how it is linked with OpenAI API.
- **Component logic:**
  - toggleChatBox(): Display or hide the chatBox when clicked;
  - addUserMessage(message): Format and add the user message passed in argument to the chatBox;
  - addBotMessage(message): Format and add the bot message passed in argument to the chatBox;

  - respondToUser(userMessage): Call the OpenAI with preloaded model, preloaded prompt stored in field systemMessage.js and the user message passed in argument then call addBotMessage(respond), with the openAI respond in argument;
  - finally, the listener are added on mounted.

7 Extra Modules

We used different extra modules in our project implementation, both for the front-end and the back-end.

- **Supabase:** We implemented the database using a Postgres DB on Supabase, because of it's easy integration both with Nuxt3 and Vercel.
- **TailwindCSS:** We used TailwindCSS for easier and more compact styling of pages.
- **Buefy:** We used Buefy components for more complex elements, such as input fields with alerts for the newsletter, dropdowns in the NavBar component and carousels in the homepage.
- **Material Design Icons**: Installed to support icons in Buefy components.

8 SEO and Accessibility

Search Engine Optimization and Accessibility were two key aspects we kept in consideration while developing our website. To test both Accessibility and SEO we used the Lighthouse tool, which returned a score between 90% and 100% on each page for both aspects.

8.1 SEO

Search Engine Optimization is the process of making a website better for search engines. We implemented the following features to make the website easily findable by web crawlers:

- Server Side Rendering: SSR allows the page to load all elements before being analyzed by crawlers.
- useSEOMeta(): we used this function in each page to describe page metadata, such as page title, description and keywords
- Use of relevant meta tags: to enforce page metadata

8.2 Accessibility

We followed accessibility guidelines and techniques to make the website compliant with accessibility web standards. We achieved this by implementing the following features

- Responsiveness: every page has fully responsive design and elements, so that the website can be viewed on every device without losing functionality.

- Alt Text: each element relevant for navigation has an alt text attribute with an appriopriate description, in order to make the website clear for screen readers
- aria-label: the aria-label attribute is used to specify the function of some elements, like links used as buttons