



POLITECNICO
MILANO 1863

**SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE**

Technology Report

SheRise - <https://sherise-teagreen.vercel.app>

Authors:

Emanuele Musto 10691229 emanuele.musto@mail.polimi.it

Riccardo Speroni 10671842 riccardo.speroni@mail.polimi.it

Matteo Spreafico 10669138 matteo2.spreafico@mail.polimi.it

Manuel Tacca 10707009 manuel.tacca@mail.polimi.it

Delivery date: **26-06-2024**

Course name: **Hypermedia Applications (Web and Multimedia)**

Academic Year: 2023-2024

Abstract

The aim of this technical document is to present the key technical choices made prior to the development process of the SheRise website, detailing the project's structure and the components designed and developed for it. Additionally, it includes a list of the extra modules and libraries used, and explains how the website was made compliant with accessibility and SEO guidelines.

Contents

Abstract	i
Contents	ii
1 Introduction	1
1.1 Group Information	1
1.2 Project Information	1
1.3 Work Breakdown	1
2 Documentation	2
2.1 Chosen Theme	2
2.2 Technological Choices	2
2.3 Project Structure	3
2.3.1 Links/Pages Structure	3
2.3.2 Available Server Endpoints	4
2.4 Custom Types	5
2.5 Custom Components	5
2.5.1 Buttons	6
2.5.2 Cards	6
2.5.3 Carousels	7
2.5.4 Containers	8
2.5.5 Slides	11
2.5.6 Other Components	12
2.6 Extra Modules	12
2.7 External Libraries	13
3 Extras	14
3.1 Accessibility Compliance	14
3.2 SEO Guidelines Compliance	15

1 | Introduction

1.1. Group Information

Group name: **Tea Green**

Group Components

Name	Surname	Person Code
Emanuele	Musto	10691229
Riccardo	Speroni	10671842
Matteo	Spreafico	10669138
Manuel	Tacca	10707009

Table 1.1: Tea Green's components

1.2. Project Information

Link to our website: **<https://sherise-teagreen.vercel.app>**

Link to our GitHub repository: **<https://github.com/manuel-tacca/HYP-Project-2024>**

1.3. Work Breakdown

We have tried the best we could to keep the work balance among all members as equal and as widespread as possible, in order to let everyone learn the most while doing this project. However, to speed up our work, we had to split the workload at some point: specifically, while Riccardo Speroni and Manuel Tacca focused on the Chatbot's design and implementation, Emanuele Musto and Matteo Spreafico focused on the documentation and the website's styling. Nonetheless, everybody contributed to the design and the implementation of the application's components.

Emanuele Musto, Riccardo Speroni, Matteo Spreafico, Manuel Tacca

2 | Documentation

2.1. Chosen Theme

We have decided to create a website for a company called "**SheRise**", based in Los Angeles, California, dedicated to supporting women who are victims of abuse and investing in the prevention of such abuse.

2.2. Technological Choices

- Server-side development: **Nuxt Nitro API endpoints**. This is because the project specification required us to use the Nuxt framework.
- Application hosting: **Vercel**. We preferred to use Vercel for application hosting instead of GitHub Pages because we believe static pages are not suitable for a company website where the staff, projects, and services details might change frequently.
- Database hosting: **Supabase**. The relational model is a good fit for a website where the stored data have interrelationships (Supabase is based on PostgreSQL).
- Rendering mode: **SSR**. Since visibility is critical for a company's website, rendering pages on server-side allows web crawlers to better index the website, resulting in better ranking in search results. Plus, we immediately discarded CSR because we knew since website design that we would not need a high level of interaction.
- Scripting language: **TypeScript**. Compared to JavaScript, TypeScript allowed us to write more robust and bug-free code. Also, it allowed us to define types related to the database entities, leading to a better developer experience when defining props for our components.

2.3. Project Structure

2.3.1. Links/Pages Structure

Links/Pages

Page	URL	Description
Home	/	It is SheRise's home page. It summarizes content from other pages and provides various links to them.
About Us	/about	It provides some details about SheRise's story and mission.
Our Team	/persons	It displays all the people working for SheRise.
Person	/persons/[id]	It displays all the details about a specific person working at SheRise.
Activities	/activities	It displays two carousels summarizing all the projects and services SheRise is involved in.
Projects	/activities/projects	It displays all the projects SheRise is involved in.
Project	/activities/projects/[id]	It displays all the details about a specific project.
Services	/activities/services	It displays all the services SheRise is involved in.
Service	/activities/services/[id]	It displays all the details about a specific service.
Contacts	/contact	It provides all the details about how to connect with SheRise.
Error	/[any other]	It is shown everytime a wrong navigation path is entered and gives the user some details about what went wrong. Additionally, it shows a button that easily allows the user to go back to the home page.

Table 2.1: SheRise's links/pages

2.3.2. Available Server Endpoints

Server Endpoints

API call	Parameters	Description
/chatbot/apiProxy	context	Returns the chatbot's response to the user's given utterance, taking into account the previous context.
/person/getAllPersons	none	Returns a subset of data about all the persons involved with SheRise, in order to build cards to show.
/person/getPersonById	person ID	Returns all the data about a person given their ID.
/person/getTotalNumberOfPersons	none	Returns the total number of persons working at SheRise, so that we can build group links without hardcoding.
/project/getAllProjects	none	Returns a subset of data about all the projects SheRise is involved in, in order to build cards to show.
/project/getProjectById	project ID	Returns all the data about a project given its ID.
/project/getRelatedProjects	supervisor ID	Returns all the data about the projects the person with the given ID is supervisor of.
/project/getTotalNumberOfProjects	none	Returns the total number of projects SheRise is involved in, so that we can build group links without hardcoding.
/service/getAllServices	none	Returns a subset of data about all the services SheRise is involved in, in order to build cards to show.
/service/getServiceById	service ID	Returns all the data about a service given its ID.
/service/getRelatedServices	supervisor ID	Returns all the data about the services the person with the given ID is supervisor of.

/service/ getTotalNumberOfServices	none	Returns the total number of services SheRise is involved in, so that we can build group links without hardcoding.
/testimonial/ getRandomTestimonials	testimonials count	Returns three random testimonials about SheRise's services, given the total number of testimonials.
/testimonial/ getTestimonialsByServiceId	service ID	Returns all the data about the testimonials related to the service with the given ID.
/testimonial/ getTotalNumberOf Testimonials	none	Returns the total number of SheRise testimonials, so that we can avoid hardcoding when fetching three random testimonials in the home page.

Table 2.2: SheRise's server endpoints

2.4. Custom Types

We have defined a few extra types to facilitate the developer experience. They can be distinguished into two categories:

1. **Database Types.** They are: Person, Project, Service, Testimonial. They are related to the corresponding database entities and have been generated following the official Supabase tutorial. Such types have made it easier for us to enable type safety, especially when retrieving data from Supabase.
2. **Utility Types.** This category comprises only the Message type, which has been created for easy storage and access of chatbot interactions.

2.5. Custom Components

Most of our website is built out of the components that we have designed and developed keeping in mind the following two principles:

- **Reusability.** Several components, such as cards and buttons, are widely used across our website. In order to keep them coherent, we have effectively reused their code through the definition of components. To allow for slight variations, we have

decided to make use of presence/absence of props or flag props.

- **Modularity.** Defining components allowed us to make each page's code much more readable. Indeed, we have defined some components that have been used only once in the codebase because of readability and ease of maintenance.

We will now describe our custom components in detail.

2.5.1. Buttons

CustomButton

This component is a SheRise-style button that allows to navigate to a new page, when clicked. It may optionally have either a left icon or a right icon, based on the fact that one of the two props is given. Additionally, one may decide to override the default width through the designated prop.

CustomButton

Prop	Type	Required	Short Description
value	string	Yes	The string that is displayed.
to	string	Yes	The path of the page to go to.
leftIcon	string	No	The name of the Nuxt icon to place on the left.
rightIcon	string	No	The name of the Nuxt icon to place on the right.
width	string	No	The string representation of the desired width.

Table 2.3: CustomButton details

FloatingButton

This component is a special kind of button that is always visible on the bottom-right corner of every page and, when pressed, toggles the chatbot interface. Such interface is included in the component since the chatbot is available only through the floating button and the floating button has no other uses in the website. This component has no prop.

2.5.2. Cards

ActivityCard

This component is a card that shows, for a given activity, its name and its picture. Additionally, one may decide to override the default height through the designated prop.

ActivityCard

Prop	Type	Required	Short Description
img	string	Yes	The path of the activity image.
name	string	Yes	The name of the activity.
height	string	No	The string representation of the desired height.

Table 2.4: ActivityCard details

PersonCard

This component is a card that shows, for a given person, their name, their picture, and their main role.

PersonCard

Prop	Type	Required	Short Description
img	string	Yes	The path of the person image.
name	string	Yes	The full name of the person.
main_role	string	Yes	The main role of the person.

Table 2.5: PersonCard details

2.5.3. Carousels

The following carousels have been made with the help of the Bootstrap CSS library. For accessibility reasons, they must be manually controlled (no auto-scroll is enabled).

ActivitiesCarousel

This component is a carousel that shows all the activities of a given type.

ActivitiesCarousel

Prop	Type	Required	Short Description
activities	Project[] or Service[]	Yes	The list of activities to show.

type	string	Yes	The type of the carousel cards (e.g. project, service).
------	--------	-----	---

Table 2.6: ActivitiesCarousel details

TestimonialsCarousel

This component is a carousel that shows all the testimonials of a certain service.

TestimonialsCarousel

Prop	Type	Required	Short Description
testimonials	Testimonial[]	Yes	The list of testimonials to show.

Table 2.7: TestimonialsCarousel details

2.5.4. Containers

The following components represent various sections of the website. They do not embody a component *per se* as they rather hold together multiple components and/or HTML pieces of code that are either reused together across the site or logically coherent (in the latter case, the container has been created for modularity and easy maintenance).

ActivityImageAndSupervisorCardContainer

This component is a container that holds together the activity image and the card of its supervisor.

ActivityImageAndSupervisorCardContainer

Prop	Type	Required	Short Description
activity	Project or Service	Yes	The activity to show.
supervisor	Person	Yes	The person who is responsible for the shown activity.

Table 2.8: ActivityImageAndSupervisorCardContainer details

CVInfoContainer

This component is a container that holds together the parts of the short CV related to a person.

CVInfoContainer

Prop	Type	Required	Short Description
bio	string	Yes	The bio of a person.
education	string	Yes	The education journey of a person.
past_experience	string	Yes	The past experience of a person.
main_expertise	string	Yes	The main expertise of a person.

Table 2.9: CVInfoContainer details

DescriptionContainer

This component is a container that nicely shows the description of an activity.

DescriptionContainer

Prop	Type	Required	Short Description
description	string	Yes	The description of an activity.

Table 2.10: DescriptionContainer details

GroupLinksContainer

This component is a container that holds together the "previous" and "next" group links when visiting the page of a kind of topic.

GroupLinksContainer

Prop	Type	Required	Short Description
id	string	Yes	The id of the kind of topic.
type	string	Yes	The type of the topic (e.g. person, project, service).
maxBound	number	Yes	The number of instances of the kind of topic currently available in the database.

Table 2.11: GroupLinksContainer details

MainPageInfoContainer

This component is a container that holds together three headers that are shown in many main pages of the website. However, since not every page has to have them, they are all optional.

MainPageInfoContainer

Prop	Type	Required	Short Description
pageTitle	string	No	The title of the page.
orientationalInfo	string	No	The orientational information of the page.
introduction	string	No	A brief introduction to what will be shown in the page.

Table 2.12: MainPageInfoContainer details

PersonInfoContainer

This component is a container that holds together a person's picture, full name, main role, and personal motto.

PersonInfoContainer

Prop	Type	Required	Short Description
img	string	Yes	The path to the person's picture.
name	string	Yes	The full name of the person.
main_role	string	Yes	The main role of the person.
motto	string	Yes	The personal motto of the person.

Table 2.13: PersonInfoContainer details

ServiceInfoContainer

This component is a container that holds together a service's opening hours, duration (if present), and specific contacts.

ServiceInfoContainer

Prop	Type	Required	Short Description
opening_hours	string	Yes	The opening hours of the service.
duration	string	No	The duration of the service.
contacts	string	Yes	The contacts of the service.

Table 2.14: ServiceInfoContainer details

2.5.5. Slides**ActivitySlide**

This component is a carousel slide that shows, for a given activity, its name, its picture, and its description. Additionally, it shows a CustomButton that gives the user the possibility to learn more about that activity, by being redirected to its page.

ActivitySlide

Prop	Type	Required	Short Description
id	number	Yes	The id of the activity.
img	string	Yes	The path of the activity image.
name	string	Yes	The name of the activity.
description	string	Yes	The description of the activity.
isActive	boolean	Yes	Whether the slide is currently shown (e.g. on a carousel).
type	string	Yes	The activity type (e.g. project, service).

Table 2.15: ActivitySlide details

TestimonialSlide

This component is a carousel slide that shows, for a given testimonial, their name, their picture, and their statement.

TestimonialSlide

Prop	Type	Required	Short Description
------	------	----------	-------------------

img	string	Yes	The path of the testimonial image.
name	string	Yes	The name of the testimonial.
statement	string	Yes	The statement of the testimonial.
isActive	boolean	Yes	Whether the slide is currently shown (e.g. on a carousel).

Table 2.16: TestimonialSlide details

2.5.6. Other Components

Now, we will briefly describe the remaining custom components that do not belong to any mentioned category. Notice that none of the following ask for any prop.

Footer

This component is the footer of SheRise, displaying contact links, EIN number, and more.

Loader

This component is a circular loading bar that is displayed as a temporary replacement of database data everytime that they are lazily loaded.

NavBar

This component is the navigation bar of SheRise, including all its landmarks. Its layout significantly changes with respect to the maximum available width of the screen.

2.6. Extra Modules

Extra Modules

Module Name	Usage Description
supabase	It was used to interact with our remote database.
google-fonts	It was used to import Google fonts inside the website.
nuxt-icon	It was used to access modern and standardized icons.

Table 2.17: Extra Modules usage description

2.7. External Libraries

As for external libraries, according to the project rules, we have used only the **Bootstrap CSS library**, which was imported to handle carousels more easily, since they are complex, interactive components. In order to override some default Bootstrap variables, we have included a "custom.scss" file in the project's "assets/scss" folder.

3 | Extras

In this chapter, we will discuss how our website is compliant with the accessibility and SEO guidelines. In order to evaluate such compliance, we have used the WAVE and Lighthouse web evaluation tools.

3.1. Accessibility Compliance

The following accessibility good practices have been implemented:

- **Alternative text.** Every non-textual content has its alternative textual representation.
- **High contrast.** The website's color palette has been designed with high contrast as a top priority. In every page, text is clearly readable and distinguishable from the background. Plus, buttons and other interaction elements are clearly visible and distinguishable from other content.
- **Clear structure.** Different pieces of content are clearly distinguished because of effective spatial allocation and proper use of headings.
- **No self-moving elements.** No element of the SheRise website is self-moving (notably, the carousels do not automatically move and need user interaction to display a new slide).
- **Simple language.** The content is written without using complex or technical words. In particular, if an error occurs, no technical word or error code is shown (instead, a custom error page is shown).
- **Coherence.** All the similar pages of the website are structured in the same way, so that the user knows what to expect when visiting pages with topics of the same kind.
- **Standards compliance.** All the HTML and CSS content is accessible from any browser.

3.2. SEO Guidelines Compliance

The following Search Engine Optimization (SEO) good practices have been implemented:

- **Unique title tags.** Each page of the website has a unique title tag that summarizes in a couple of words its content.
- **Effective meta descriptions.** Each page of the website also has a short but informative description that represents what to expect when navigating the page.
- **Clear content hierarchy.** Thanks to an effective use of different headers, it is clear which content is more important.
- **Clear URLs.** The URLs contain clear keywords that refer to the content that the related pages show.
- **Mobile friendliness.** The website adapts its layout based on the maximum available screen width, thanks to CSS media queries.

List of Tables

1.1	Tea Green's components	1
2.1	SheRise's links/pages	3
2.2	SheRise's server endpoints	5
2.3	CustomButton details	6
2.4	ActivityCard details	7
2.5	PersonCard details	7
2.6	ActivitiesCarousel details	8
2.7	TestimonialsCarousel details	8
2.8	ActivityImageAndSupervisorCardContainer details	8
2.9	CVInfoContainer details	9
2.10	DescriptionContainer details	9
2.11	GroupLinksContainer details	10
2.12	MainPageInfoContainer details	10
2.13	PersonInfoContainer details	10
2.14	ServiceInfoContainer details	11
2.15	ActivitySlide details	11
2.16	TestimonialSlide details	12
2.17	Extra Modules usage description	12