Hypermedia Application - Technical Report

Web Technology Report of `https://venture-capital.vercel.app/`

WAMM Team - Github repo

Marco Bendinelli
10673478
marco.bendinelli@mail.polimi.it

Angelo Capponcelli
10653805
angelo.capponcelli@mail.polimi.it

Mattia Carini
10618404
mattia.carini@mail.polimi.it

William Zeni
10613915
william.zeni@mail.polimi.it

Delivery date 19/06/2023

Project supported at Politecnico di Milano - Electronics, Information and Bioengineering Department - Computer Science and Engineering Course

Contents

1 Work division

In the process of implementing the website, we adopted a collaborative approach to ensure efficient and coordinated work. We began by discussing and analyzing the overall design and structure of the website as a team. In particular, we held collaborative discussions to finalize the correct links and navigation flow between the pages. Once we had a clear understanding of the project's requirements and objectives, to maintain clarity and organization, we decided to split the work by dividing the pages of the website among the team members, in particular:

- **Bendinelli's pages:** Area pages
- **Capponcelli's pages:** Homepage, Portfolio pages
- **Carini's pages:** Our Team pages
- **Zeni's pages:** About us, Contact us pages

So, we assigned specific jobs to each team member that included tasks such as creating wireframes, developing abstract tables, and implementing the design on Vue. Each team member was responsible for their designated pages and components, ensuring a focused and streamlined approach to the implementation process.

Throughout the implementation phase, we maintained open lines of communication, regularly sharing progress updates and seeking feedback from one another. This collaborative approach allowed us to address any challenges or concerns effectively, leveraging the diverse expertise and perspectives within the team.

By combining our individual skills, we ensured the successful implementation of the website while maintaining consistency among the pages of the website.

2 Documentation

2.1 Description of the project

Our Hypermedia project for the course focused on applying Nielsen's and Mile's heuristics to create an engaging and user-friendly website. The project aimed to simulate a small venture capital company's online presence. The website featured several key components, including a company page showcasing its members, individual pages for each member with their respective projects, a comprehensive list of available projects categorized by their respective areas of work, an 'About Us' page highlighting the company's mission and values, and a user-friendly contact page for users to get in touch with the company.

To build the website, we utilized the Nuxt + Vue3.js framework, ensuring a smooth and dynamic user experience. Throughout the development process, our team encountered design challenges due to the lack of a dedicated designer. However, we addressed this by diligently working with Figma and drawing inspiration from existing websites.

Overall, our Hypermedia project successfully applied the principles of good website design, offering visitors a visually appealing and intuitive platform to explore the venture capital company and its projects.

2.2 Hosting service

The team decided to use Vercel as the hosting service for our company's website for several reasons. Firstly, Vercel's free offering was a significant factor, as it aligned with our budget constraints for the educational purposes. Additionally, Vercel's user interface was straightforward and intuitive, making it easy for us to navigate and manage our website.

One key feature that attracted us to Vercel was its strong integration with GitHub. Being able to push our code to the main repository and instantly see the changes reflected on the live website was incredibly convenient and streamlined our development process.

Although scalability and performance were important factors, our main goal was to establish an online presence within our local community. As a result, while we recognized the importance of a reliable and performant hosting service, our website's anticipated traffic volume remained relatively modest.

2.2.1 Rendering mode

The rapid evolution of web technologies has introduced various rendering approaches that influence how web pages are constructed and delivered to users. CSR, SSR, and SSG have emerged as popular strategies, each offering unique features and trade-offs. After a long discussion, the team decided to adopt Server-Side Rendering (SSR) as the website rendering mode for the following several compelling reasons.

First and foremost, SSR provides a faster initial page load experience for users. By generating complete HTML documents on the server and sending them directly to the client's browser, the time required to display the content is significantly reduced. This ensures a seamless and engaging user experience right from the start, enabling users to access and interact with the website without delays caused by client-side rendering.

Another significant factor behind this choice is the search engine optimization (SEO) benefits offered by SSR. Search engines can easily crawl and index the fully rendered HTML pages, enhancing the website's discoverability and potential for organic traffic. This is crucial for improving online presence and facilitating business growth.

Additionally, in the context of our specific scenario, it is worth noting that the website is intended to be lightweight, and there is no significant concern regarding the rendering time and computational power required during the rendering process.

2.3 Structure of the project

The Nuxt.js framework's inherent functionality includes the automatic generation of the vue-router configuration by analyzing the organization of Vue files within the pages directory. Leveraging this mechanism, we strategically exploited it to align our web application's original design with the structure of the pages and folders within the project.

2.3.1 Pages structure

- *areas* folder: contains web pages related to the areas;
  - *[id]*: page of a specific area with parameter id that indicates the id of the area;
  - *index*: introductory page of the areas;
- *portfolio* folder: contains web pages related to the projects;
  - *[[area]]* parametric folder: is optional and is only present for single project pages that do not come directly from 'All projects'; it is set for a project that comes from 'Most relevant projects' (e.g. */portfolio/most-relevant-projects/6*) or that comes from the portfolio page of a specific area (e.g. */portfolio/HealthTech/19*)

        * *[id]*: can assume different meanings depending on the type of page required:
            1. id of a project from the 'All projects' page, the single project page will be displayed (e.g. */portfolio/1*)
            2. id of a project from the 'Most relevant page' page with [[area]] set to 'most-relevant-projects', the single project page will be displayed (e.g. */portfolio/most-relevant-projects/6*)
            3. id of a project from the portfolio specific area page with [[area]] set to area name, the single project page will be displayed (e.g. */portfolio/FinTech/21*)
            4. name of a specific area with [[area]] not set, portfolio specific area page will be displayed (e.g. */portfolio/FinTech*)
    - *index*: introductory page of the projects ('All projects' page of portfolio);
    - *most-relevant-projects*: introductory page of the most relevant projects ('Most relevant projects' page of portfolio);

- *team* folder: contains web pages related to the team people;
    - *[id]*: page of a specific person with parameter id that indicates the id of the person;
    - *index*: introductory page of the people;
- *about*: about us page;
- *contact*: contact us page;
- *index*: homepage;

2.3.2 Available server endpoints

- *index.get.js*: retrieves data for the homepage from the database; if successful, returns a maximum of 6 projects (id and startup id to get startup image), a maximum of 8 people (id, image), a maximum of 3 areas (id, name, description);
- *portfolio/index.get.js*: retrieves data for 'All projects' page from database; if successful, returns data useful for project overview (id, title, overview, startup id) sorted alphabetically by title;
- *portfolio/most-relevant-projects.get.js*: retrieves data for 'Most relevant projects' page from database; if successful, returns data useful for project overview (id, title, overview, isRelevant, startup id) sorted alphabetically by title;
- *portfolio/[area]/[id].js*: Depending on the parameters retrieves data for different type pages from database:
    - id = 'undefined', area = [area.name]: retrieves data for a specific area page within a portfolio, where the name of the area is equal to [area.name];
    - id = [project.id], area = 'all': retrieves data for a single project page coming from 'All projects' page where the id of the project is equal to [project.id];
    - id = [person.id], area = 'person': retrieves data of specific person projects where the id of the supervisor is equal to [person.id];
    - id = [project.id], area = 'Most relevant projects': retrieves data for a single project page coming from 'Most relevant projects' page where the id of the project is equal to [project.id];
    - id = [project.id], area = [area.name]: retrieves data for a single project page coming from a portfolio specific area page where the id of the project is equal to [project.id] and the name of the area is equal to [area.name];

if successful, returns data useful for project overview (id, title, overview, isRelevant, startup id) sorted alphabetically by title, or data of single project for the detailed page, and data of the next and previous project with respect from where the user is coming;

- *area/index.get.js*: retrieves data for the introductory page of the areas from the database; if successful, returns data useful for areas overview (id, name, description) sorted by id;
- *area/[id]*: retrieves data for single area page from the database; if successful, returns data of single area for the detailed page, and the data of the related projects;
- *team/index.get.js*: retrieves data for the introductory page of the team from the database; if successful, returns data useful for person overview (id, full_name, position, image) sorted by id;
- *team/[id]*: retrieves data for single person page from the database; if successful, returns data of single person for the detailed page;

2.4 Components implemented

- *TitleWithImage*: component used for the Title section of a page, displays a title in a big font size and a small subtitle under it. Can be modified to add an image as background instead of the plain white background color.
  Props: title, subtitle.
- *AreaCard*: the card component serves the purpose of showcasing an area within the Areas page of our website. It contains its image, along with its corresponding name and a concise description. The card also includes a "Discover More" Nuxt link, which directs users to the dedicated page for that specific area. The "index" and "areOdd" props serve to determine the layout and presentation of the area cards. They permit to alternate the colors between the cards. In particular, the areOdd prop, permits to always position the white card at the end.
  Props: areaId, name, description, index, areOdd.
- *TeamCard*: card for the Our Team page. Displays the person's image, name and position at the company. The card is itself a Nuxt link to the individual person's page, set by passing it the person's id as a prop.
  Props: name, position, id, image.
- *TeamBigCard*: card for the individual person page. Contains the person's image, name, position at the company, a short 1 line impactful description, links to social media and linkedin accounts, a long description containing education information, past experiences and expertise, and finally group links to the next and previous person and back to the Our Team page.
  Props: name, position, logline, image, description, id, next.
- *ProjectBigCard*: gives an overview of the projects in the introduction pages to them ('Most relevant project' page). It has the link connecting it to a specific project, an image of the project startup, the title, and a short introductory text.
  Props: title, overview, startupId, link.
- *ProjectSmallCard*: gives an overview of the projects in the introduction pages to them ('All project' page and specific area projects page). Compact version of *ProjectBigCard*. It has the link connecting it to a specific project, the id useful for filtering projects, an image of the project startup, the title, and a short introductory text.
  Props: title, overview, startupId, link, id.

- *ProjectSideDrawer*: side menu in the portfolio pages. It contains the labels and corresponding links of the page containing all projects, the most relevant projects, and pages containing projects from a specific area. The entries of the latter are dynamically loaded from the database (being that the areas may change). Receives as props the index of the active menu item on that page (it is highlighted in the display to let the user know what page it is on).
  Props: pageindex.

- *ProjectTab*: tab used on the individual project page. There are two types: one (tabType = 0) comprises startups and supervisors with an image, a title and subtitle, and a link on the entire tab; the other (tabType = 1) encompasses correlated areas without an image, featuring a title, and links to different areas on the individual label (not on the entire tab).
  Props: title, titleLabel, subtitle, subtitleLabel, image, links.

- *ContactsCard*: card for the contact page (also used in the about page for its compatible design). It contains a simple title and a short text. Eventually, you can add a google maps link to get on location rendered inside the card.
  Props: title, content, maps.

- *Breadcrumb*: implements breadcrumbs within the pages of the website. It takes an array of elements as a parameter, characterized by a label and a link, representing the hierarchy to be displayed. Using this parameter, the breadcrumb is constructed, arranging the labels in an ordered manner and implementing links to the various pages (except for the last label, which is assumed to have no link since it represents the current page).
  Props: crumbs [{label, link}].

- *SearchBar*: is implemented on the portfolio and area pages. Specifically, it consists of a text field that, upon receiving input, invokes the search() function, which retrieves the user's input and emits it as a value. Additionally, a useful parameter, 'id', is provided for mobile display purposes.
  Props: id.
  Emits: search-filter.

- *Navbar*: header of the website, encompassing all the relevant landmarks that aid the user in navigating the webpages.

- *MobileNavbar*: navigation menu used for mobile devices. Appears after pressing the button at the top right of the NavBar and contains the main section landmarks present in the NavBar for larger devices.

- *Footer*: footer of the website.

- *Logo*: represents the organization's logo displayed in both the header and footer sections of the website. It is designed to be responsive, adapting its size according to the dimensions of the layout.

2.5 Extra modules

The only extra module used in the implementation is Supabase, which is used to store and dynamically retrieve the information about people, projects and areas in the website. For the scope of this project and for simplicity, a public bucket was used to store images on Supabase without any authentication. In a commercial project it would be best to use a private bucket and set up authentication when fetching images from the DB.

2.6 Extra functionalities

- *Lazy image loading*: implemented for all images that in the website that don't need to be loaded until the user scrolls to the area of the page containing them. One example is the case with the images in the home page or the images of the person cards in the Our Team page, in which if a user never scrolls down, the lower images are never loaded. This has been implemented using Vue.Js's IntersectionObserver API which scans the viewport, or an area of the viewport, and can trigger events when an element enters the specified area. In the example of the person cards for the Our Team, one intersection observer that scans the entire viewport is used to lazy load the images when they enter the area, and another with a reduced height is used to activate the fading in animation of the cards when they appear in the viewport. The cards can also be made to disappear and to lazily unload the images when they move out of the viewport, but it was decided to not activate this feature, as it could be found annoying to have to watch the animations every time a user scrolls up and down.
- *Page fade in/out transitions*: page transitions have been implemented by adding a page transition attribute to the nuxt.config.ts file and a few lines of CSS code found in the App.vue file. These transitions allow a smooth and good looking transition between all pages in the website.
- *Search bar*: a filtering system has been implemented for the areas and projects to facilitate the search for a specific item. Specifically, the *SearchBar* component emits an event every time text is entered or the search icon is clicked. At this point, the filtering function is called on the pages where the component is implemented, filtering all elements that contain the received input in their title.
- *Google maps*: an easy google maps interactive location have been added in the Contacts page following the documentation provided by Google.

2.7 Optimizing website performance and user experience

2.7.1 SEO

To enhance the website's SEO, various procedures have been implemented that have resulted in excellent SEO evaluation outcomes.

Specifically, global parameters have been added in NuxtConfig concerning the website's title, the utilized charset (utf-8), meta information related to the viewport, which sets the width as the device's width in use, and the scale level set at 100%. This optimization not only ensures better display on mobile screens but also eliminates the 300-millisecond delay in user input. Global dependencies for the website's font have been included, utilizing a font from Google. Additionally, hreflang link was specified to inform search engines about the appropriate page version to display in search results for specific languages or regions.

Furthermore, on each page, an appropriate title and description reflecting the content of the pages have been specified. We dynamically set the title and description attributes using the following function:

```
useHead({
    title,
    meta: [{
        name: ‘description’,
        content: description
    }]
})
```

The links have been implemented in a way that they have descriptive text referring to the page they link to, rather than being generic. This approach enhances user understanding and improves navigation throughout the website.
Moreover, all images have been assigned alternative text to provide a descriptive explanation of the image. This alt text is crucial not only for accessibility purposes but also for search engine optimization (SEO). By including descriptive alt text, we ensure that search engines can properly index and understand the content of the images.

2.7.2 Accessibility

Excellent results have been achieved in terms of accessibility, as confirmed during the analysis phase. Particularly, as previously mentioned, implementing alternative text settings for images and descriptive text for links is not only a good practice for improving SEO but also for enhancing accessibility. While designing the website elements, efforts were made to maintain consistency in terms of colors and shapes, thereby ensuring intuitive navigation and user interaction. For instance, in project cards, in addition to the hover effect that provides access to the link, the text "Learn more" has been included, highlighted with a contrasting color to clearly indicate interactivity with the card element.

The use of Tailwind through the import of CSS classes enabled effective exploitation of framework practices to improve accessibility. Specifically, an initial color palette was defined in the Tailwind configuration file, which was later modified to achieve optimal contrast between elements and ensure clear and visible text.

The custom interactive controls implemented in the website are keyboard focusable and include a visible focus indicator, ensuring that users can navigate and interact with them using the keyboard alone. This feature enhances accessibility by accommodating individuals who rely on assistive technologies for navigation.

Furthermore, the headings on the website have been properly structured and ordered, without any level skipping. This adherence to a logical hierarchy conveys the semantic structure of the page, facilitating navigation and comprehension, particularly for users relying on assistive technologies. By following this approach, the website ensures a more seamless and intuitive browsing experience for all users.

The web application exhibits complete responsiveness, ensuring comprehensive coverage across all devices and screen sizes. Its adaptive design allows for seamless user experience and optimal display regardless of the device being used, be it desktop computers, laptops, tablets, or smartphones. This responsive approach eliminates any limitations posed by varying screen sizes, ensuring that the website's content remains accessible and user-friendly across the entire spectrum of devices.

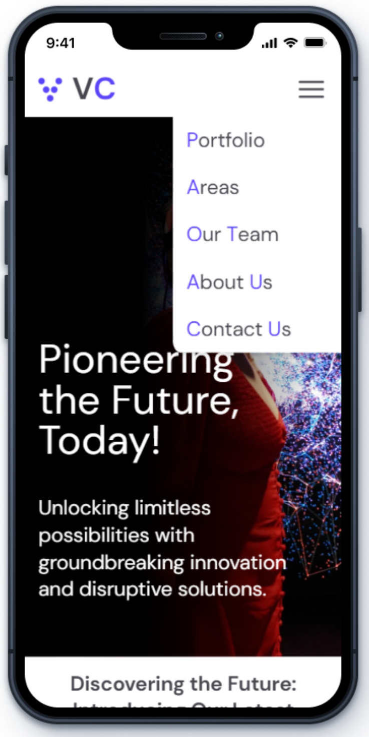

9:41
VC
Portfolio
Areas
Our Team
About Us
Contact Us
Pioneering the Future, Today!
Unlocking limitless possibilities with groundbreaking innovation and disruptive solutions.
Discovering the Future:

Figure 1: Responsive for smartphones

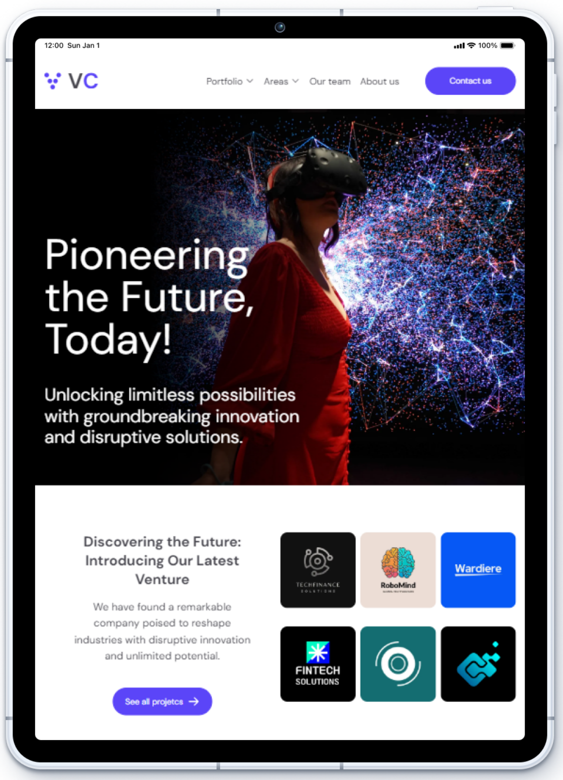

12:00 Sun Jan 1
100%
VC
Portfolio
Areas
Our team
About us
Contact us
Pioneering the Future, Today!
Unlocking limitless possibilities with groundbreaking innovation and disruptive solutions.
Discovering the Future: Introducing Our Latest Venture
We have found a remarkable company poised to reshape industries with disruptive innovation and unlimited potential.
See all projetcs
TECHFINANCE SOLUTIONS
RoboMind
Wardiere
FINTECH SOLUTIONS

Figure 2: Responsive for tablets

3 Site Lighthouse Analysis

Lighthouse is an open-source tool developed by Google that enables website developers to assess and enhance the performance, accessibility, and overall quality of their web pages. It provides detailed insights and recommendations based on a set of predefined metrics and best practices. Due to Google's significant impact and authority in the realm of the web, our team has made the decision to utilize Lighthouse as a testing tool for our website. This choice stems from acknowledging Google's wide-reaching influence, ensuring that our website aligns with their standards and guidelines, thus optimizing its visibility and accessibility within the digital landscape.

3.1 Performance

The "Performance" parameter in Lighthouse is determined by analyzing various aspects of a website's loading and execution. It measures metrics such as the time taken for the website to load, the efficiency of resource utilization, and the responsiveness of user interactions. However, it is important to note that the calculated value is merely a guideline. It can be influenced by the performance capabilities of the computer on which the analysis is conducted, as well as the geolocation from which the website is being accessed. Therefore, while the Performance score provides valuable insights, it should be considered in conjunction with real-world testing and user feedback to accurately assess and improve the website's overall performance.

3.2 Accessibility

The "Accessibility" parameter in Lighthouse is calculated by evaluating a website's compliance with accessibility standards and guidelines. Lighthouse performs automated audits to assess elements such as color contrast, proper labeling of form elements, presence of alternative text for images, keyboard navigability, and semantic markup. These audits analyze the underlying

HTML, CSS, and JavaScript code to identify potential accessibility issues. Lighthouse assigns a score based on the number of passed audits, with a higher score indicating better accessibility compliance. However, it is important to note that automated audits may not capture all accessibility concerns, and manual testing and user feedback are essential for a comprehensive assessment of a website's accessibility.

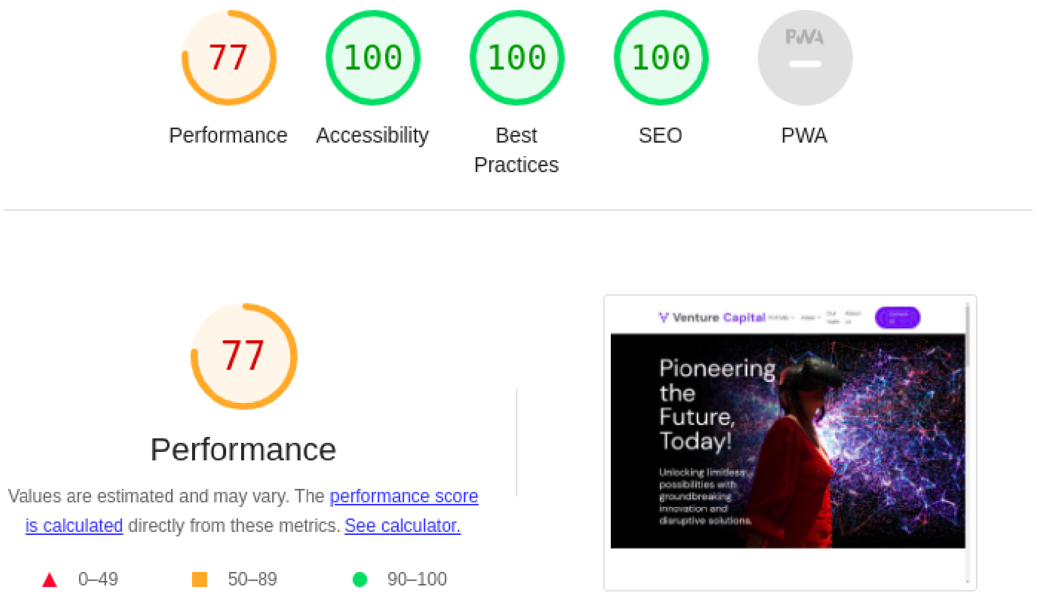
3.3 Best practices

The "Best Practices" parameter in Lighthouse is calculated by assessing how well a website adheres to recommended coding practices and industry standards. Lighthouse conducts automated audits to examine various aspects, such as using secure connections (HTTPS), implementing effective caching techniques, optimizing resource loading, employing appropriate image formats, and avoiding deprecated technologies. By analyzing the website's HTML, CSS, and JavaScript code, Lighthouse identifies any deviations from established best practices. The resulting score for the "Best Practices" parameter reflects the number of successfully passed audits. However, it is crucial to note that manual code reviews and adherence to specific project requirements are also vital for ensuring the overall quality of a website.

3.4 SEO

SEO, or Search Engine Optimization, refers to the practice of optimizing a website to improve its visibility and ranking in search engine results. It involves various strategies such as keyword optimization, content quality, backlink building, and user experience enhancement. Lighthouse does not calculate an explicit coefficient for SEO. Instead, it provides separate scores for parameters like performance, accessibility, and best practices. These scores reflect the compliance of the website with predefined metrics and guidelines. While Lighthouse's scores can indirectly contribute to SEO, the overall SEO performance depends on a broader set of factors beyond the scope of Lighthouse. It is important to consider Lighthouse's recommendations alongside other SEO practices to achieve the best possible search engine visibility.

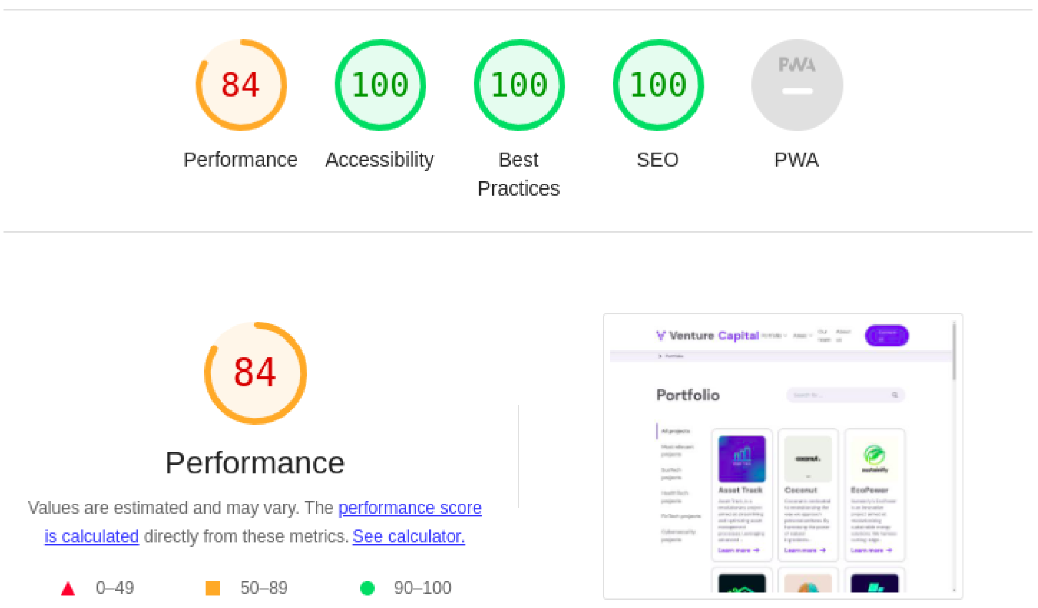
3.5 Result Screenshots


77
Performance
100
Accessibility
100
Best Practices
100
SEO
PWA
PWA
77
Performance
Venture Capital
Pioneering the Future, Today!
Unlocking limitless possibilities with groundbreaking innovation and disruptive solutions.

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Figure 3: Lighthouse results - Homepage


84 Performance

100 Accessibility

100 Best Practices

100 SEO

PWA

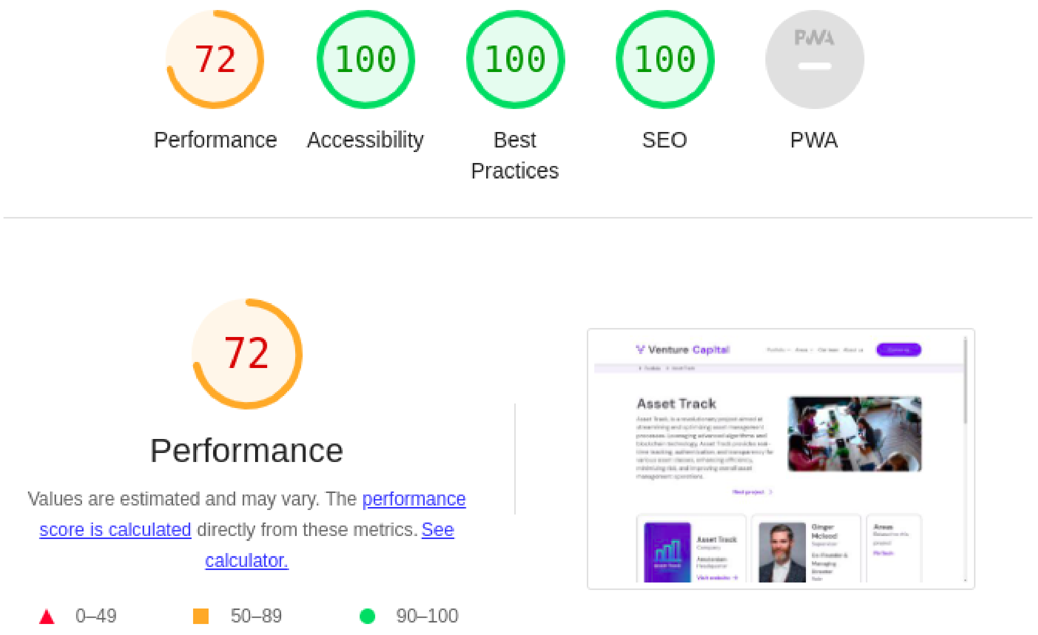
84

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Figure 4: Lighthouse results - Portfolio


72 Performance

100 Accessibility

100 Best Practices

100 SEO

PWA

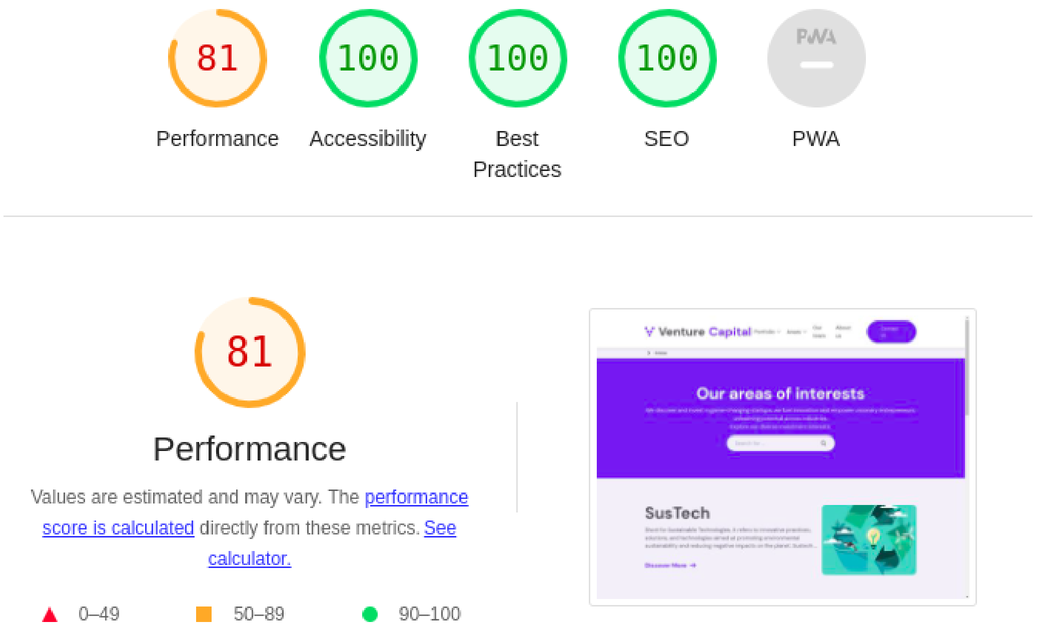
72

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Figure 5: Lighthouse results - Portfolio project


81 Performance
100 Accessibility
100 Best Practices
100 SEO
PWA

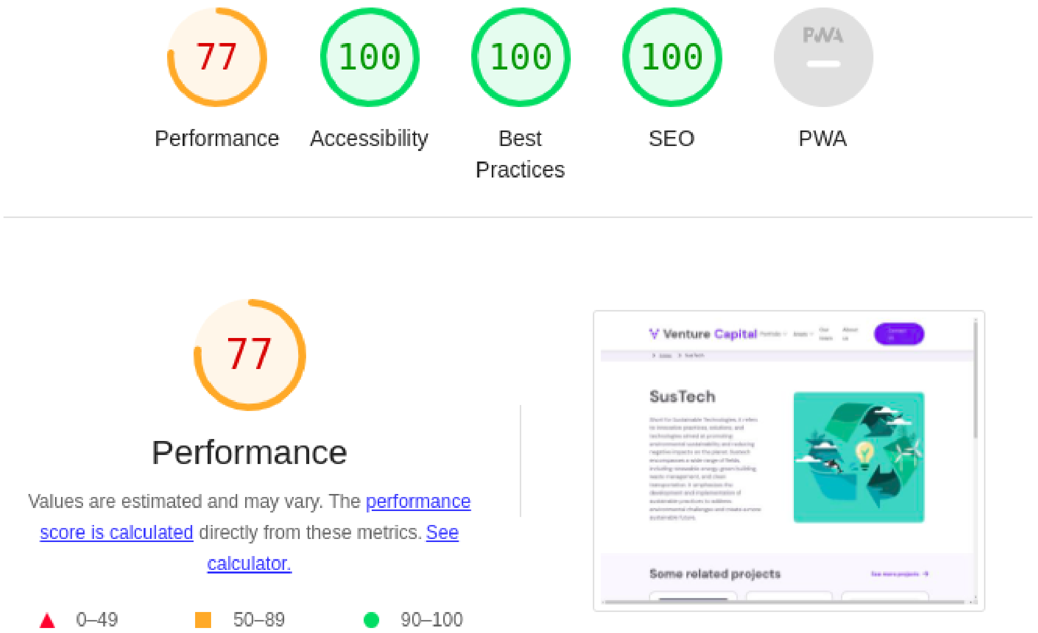
81

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Figure 6: Lighthouse results - Areas


77 Performance
100 Accessibility
100 Best Practices
100 SEO
PWA

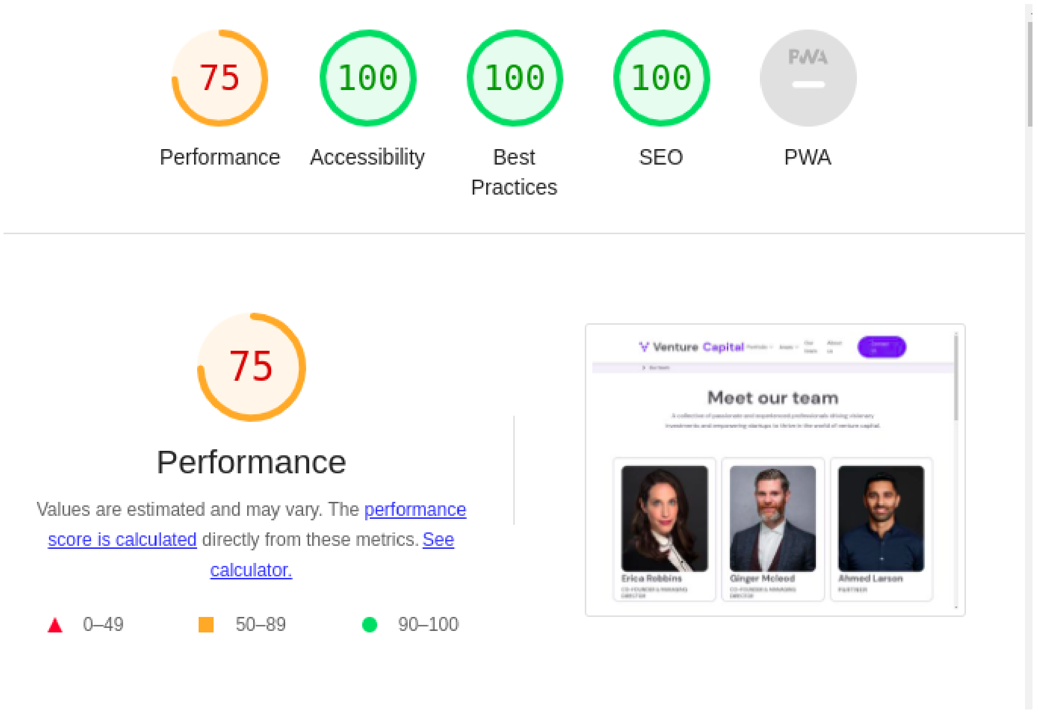
77

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Figure 7: Lighthouse results - Single area



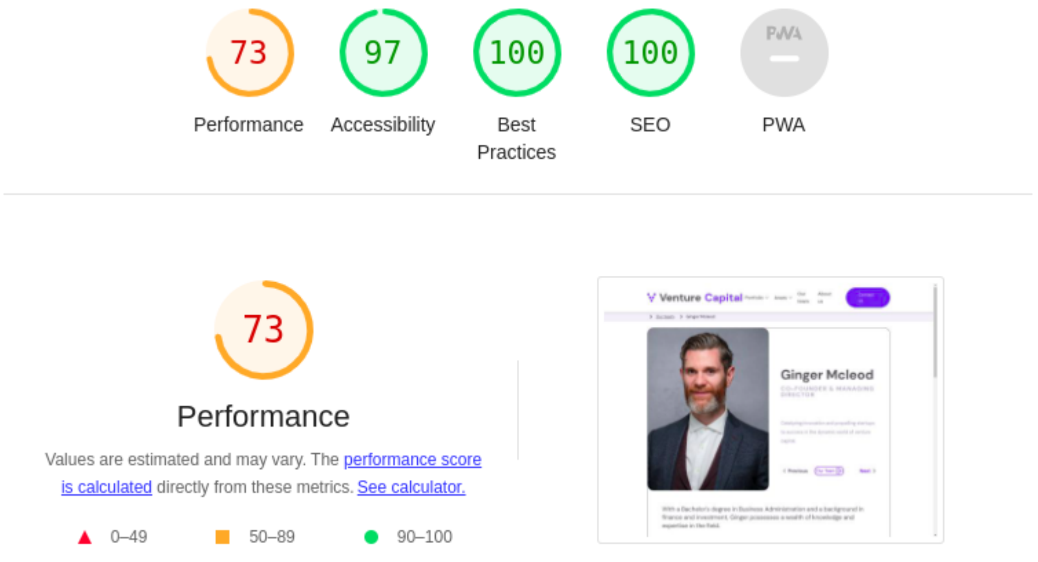
75 Performance
100 Accessibility
100 Best Practices
100 SEO
PWA

75
Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Figure 8: Lighthouse results - Our Team



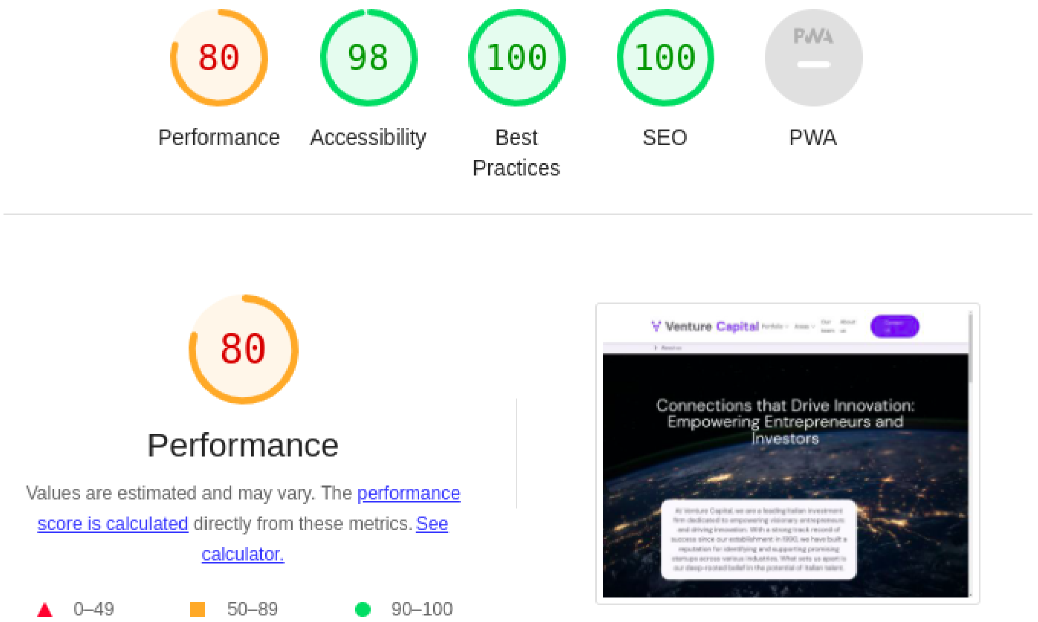
73 Performance
97 Accessibility
100 Best Practices
100 SEO
PWA

73
Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Figure 9: Lighthouse results - Team member


80 Performance | 98 Accessibility | 100 Best Practices | 100 SEO | PWA

80

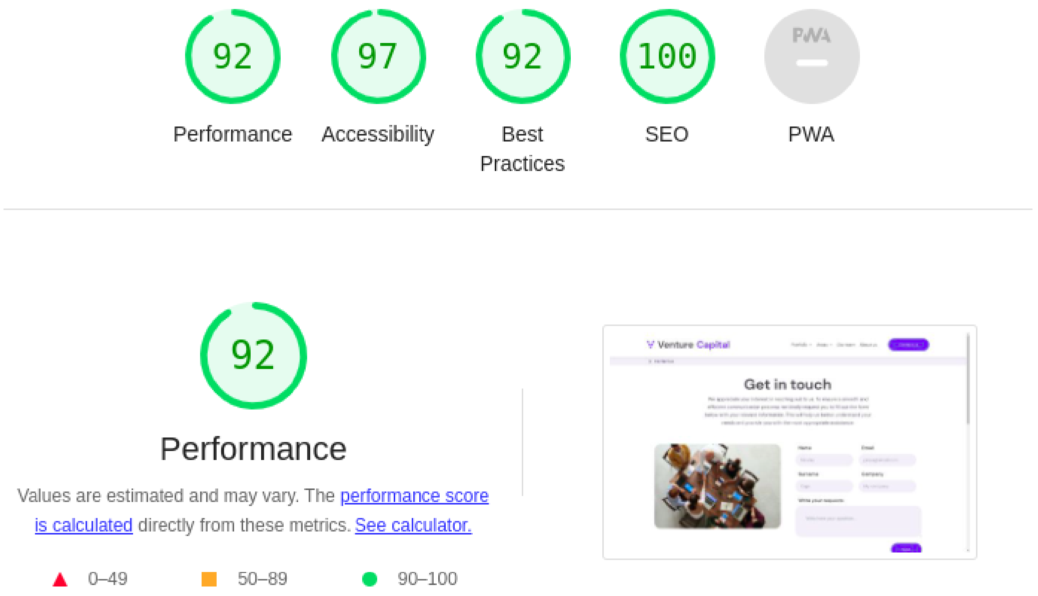
Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Venture Capital
Connections that Drive Innovation: Empowering Entrepreneurs and Investors

Figure 10: Lighthouse results - About us


92 Performance | 97 Accessibility | 92 Best Practices | 100 SEO | PWA

92

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

Figure 11: Lighthouse results - Contact us