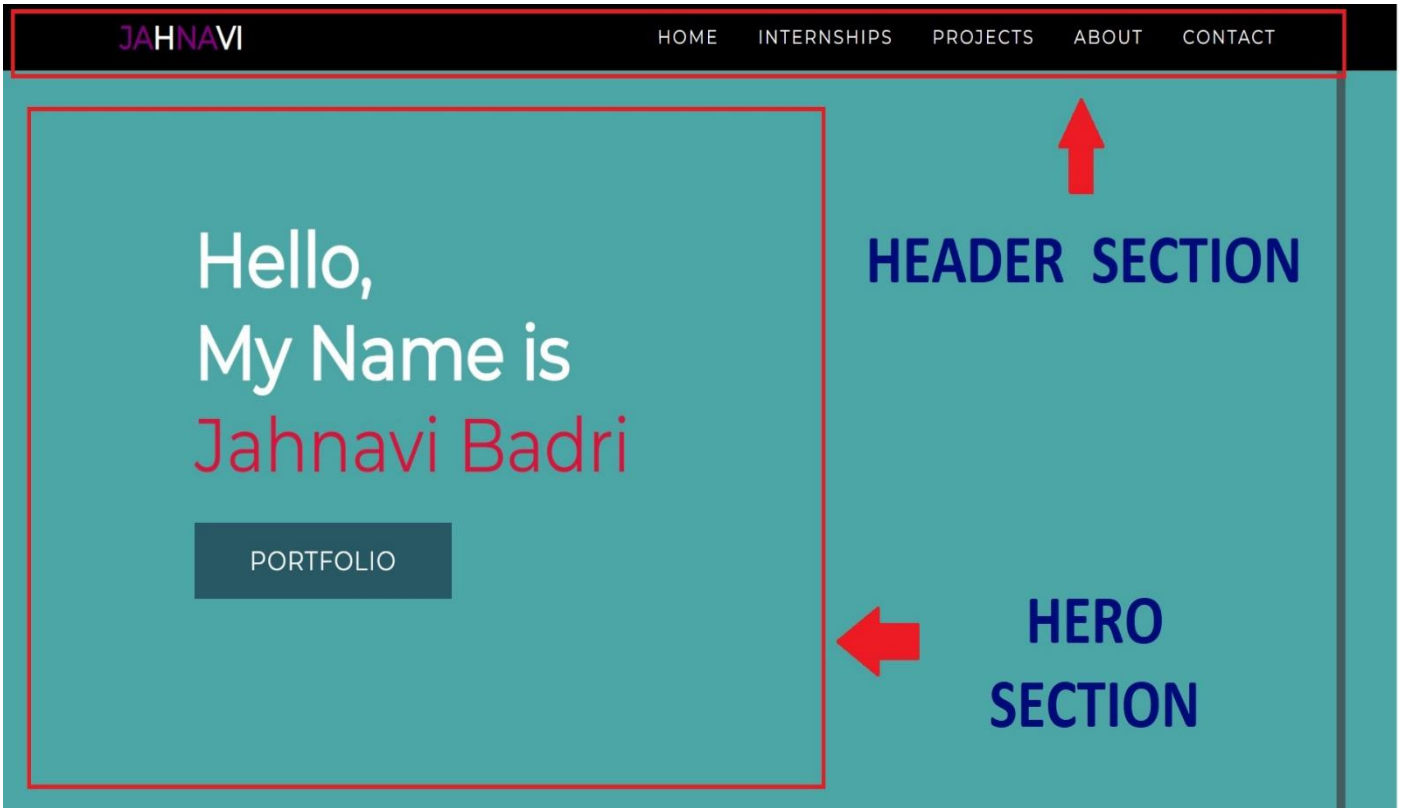
PROJECT REPORT

Project Description: The **Personal Technical Portfolio** is a static web application designed to showcase an individual's technical skills, projects, achievements, and experience in a visually appealing and organized manner. It aims to highlight their expertise, demonstrate their coding abilities, and provide a comprehensive understanding of their technical background to potential employers, clients, or collaborators. The goal of this application is to create a visually compelling and professional portfolio that impresses visitors and establishes a strong online presence.

STAGE-1

In this stage 1, we have created the basic html page with a header and hero section. In the header section, we have created html code with head tags with basic HTML5, like meta tags, div tags, anchor tags, unordered list tags etc, and CSS code using font-size, font-family, scroll-behaviour. Finally, for the smooth scrolling and creating a container for the hero section in the hero section, we have the introduction.

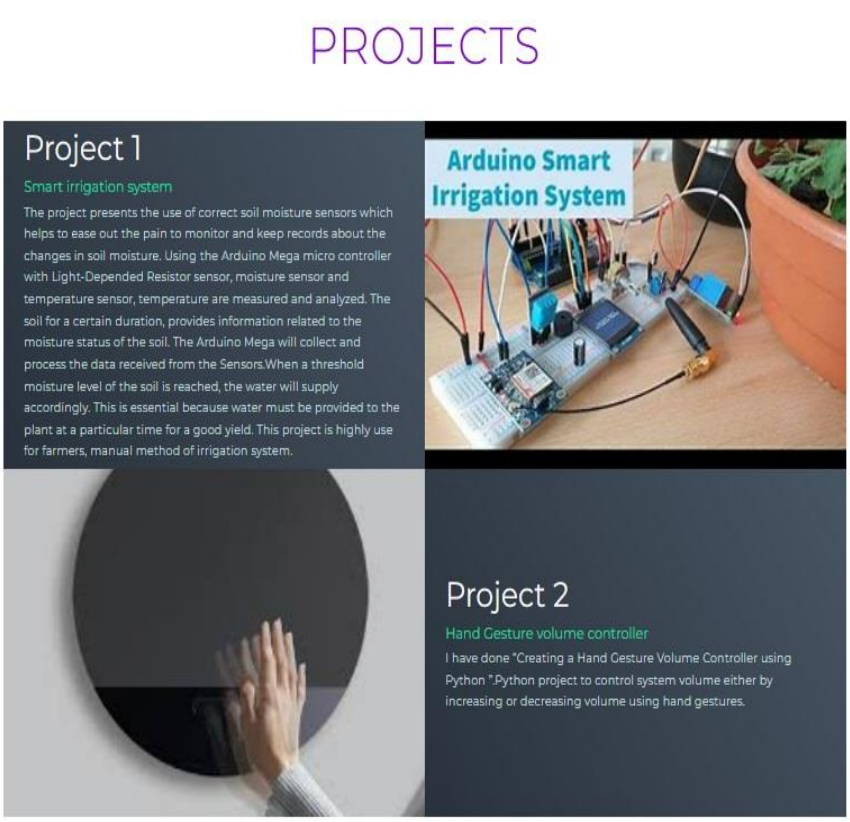


STAGE-2

In stage 2, we have created an internship section. We have placed the internships, whatever we have done with the description in paragraph tag. We have used div tags for a clear appearance and look of the section. For the CSS, we have inserted the images in this section and we have applied the width and height and we also have specified the transition effect with a slow start.

STAGE-3

In stage 3 we have the project section in this we have placed our projects we have done till now. For project section using html 5 we have about our project and created separate div tags for styling the project container. We have used the same reference as we created the internship section.



STAGE-4

In stage 4, we have the about section. In which we have placed our bio data. For the about section html5, we have inserted the image using image tag and also the CSS align is used for positioning the items along with setting the distribution of space between and around content items. The various methods and techniques are used to centre them, by taking care of the left and the right margin, etc.

ABOUT ME

Name:JAHNAVI BADRI

Email: jaanubadri71@gmail.com

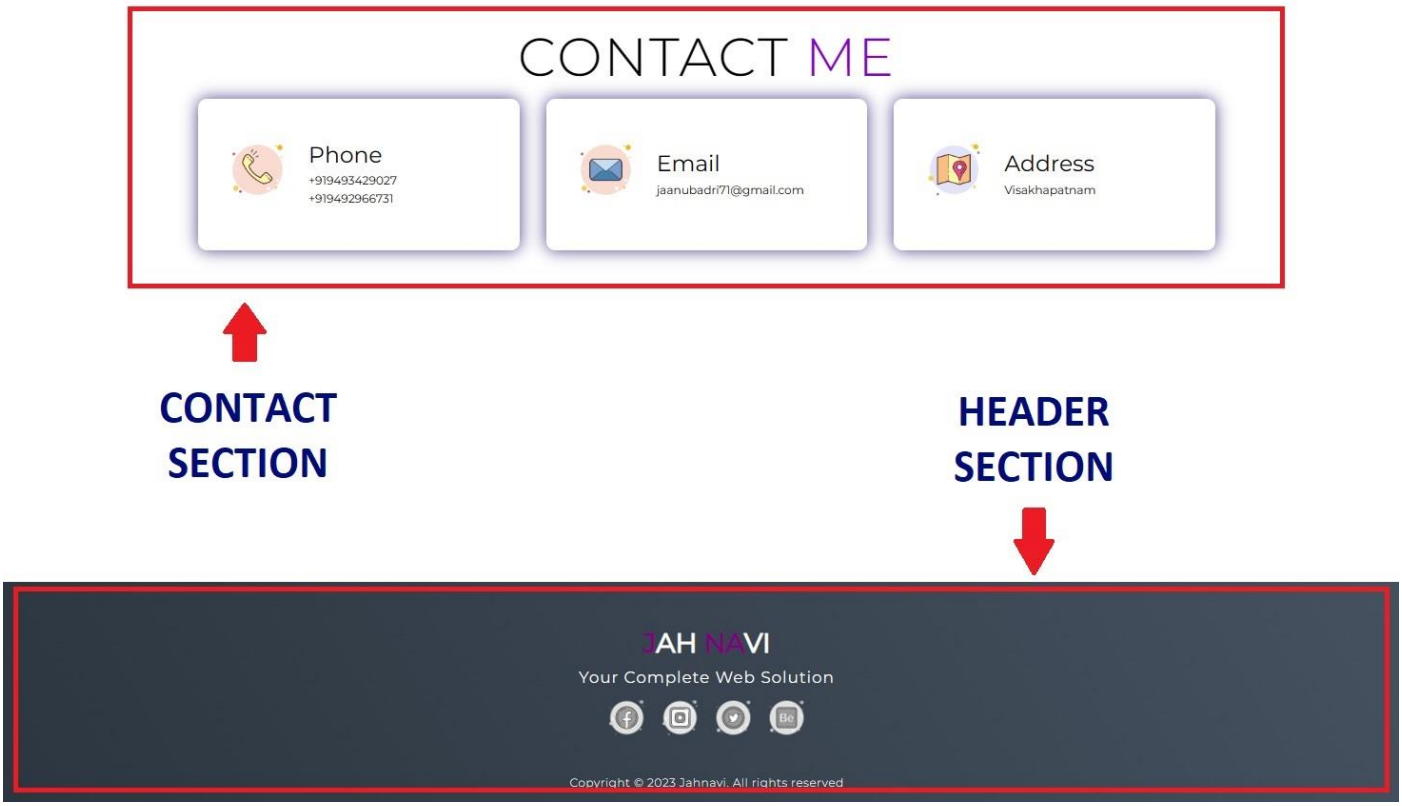
Contact No:9493429027

LinkedIn : https://www.linkedin.com/in/jahnavi-badri-594541266

DOWNLOAD RESUME

STAGE-5

In stage 5, we have the contact section and footer section. In this we have placed our contact details using containers in html5 and also used some CSS properties for inserting media icons in contact section. And in the footer section we have created social media icons (like Facebook, Twitter, Instagram, etc.) for some notable area on our website. Doing so, users can be able to communicate with other people and keep in touch with those who already have.

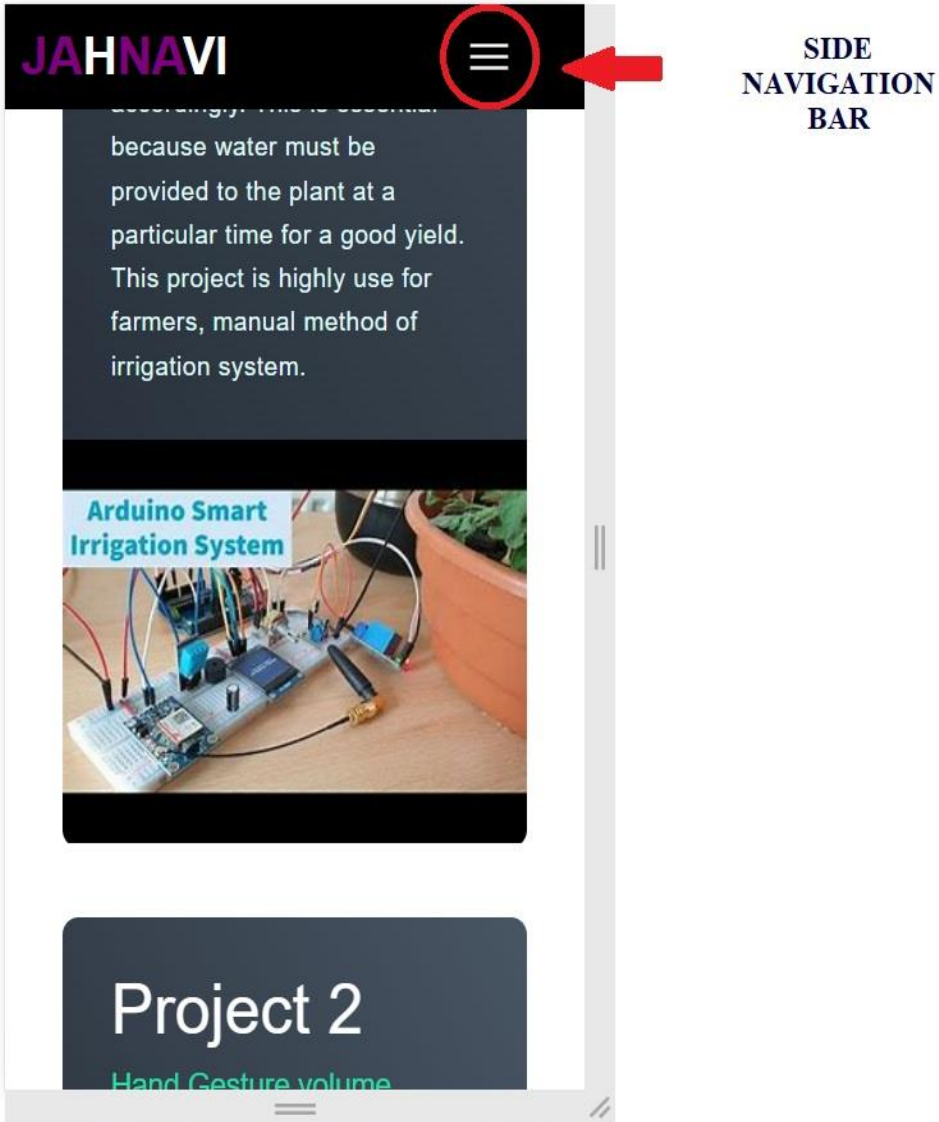


STAGE-6

MOBILE VIEW:

Mobile-first is when we start by writing our CSS for mobile devices and then use media queries to add in styling for larger screen sizes**.** In general, that means that media queries use a min-width. We're using media queries to add or overwrite styles for a set breakpoint and bigger.

Websites are naturally responsive before we even write a single line of CSS**.**
If you remove the CSS from any page on the internet, even some site made in for a very specific screen size back in 2001, you now have a responsive, mobile-friendly website!



Mobile layouts tend to be very simple, making it very easy to start there
For all the people who replied to me saying that their clients preferred seeing the desktop version, or that they were only given desktop comps by their designers, I would argue it's still easier to start mobile-first.

For many sites, once you've set up your typography, you're 70% of the way there. Things like:

- font-family
- font-size
- font-weight
- margin (on your text elements)

Next up, you can go and do some very basic layout styling on your layout elements, such as:

- padding
- background-color
- color
- and maybe some tweaks with margin

At that stage, things will be looking pretty good from a layout perspective on small screens. That means, without writing a single media query, you have a fully functional site on mobile.
If you were feeling particularly lazy, or have a very simple site, you could stick a max-width on your container and be done with the entire thing and not even have to worry about a media query at all!
Most of the time, we do want to up the game at bigger screen sizes though, and that's why I feel like mobile-first is the way to go. It's the natural progression upward.

DESKTOP VIEW:

Desktop styles tend to be more complex

When we style for desktop-first, we're adding widths, columns, and moving things around. We're adding complexity. We're doing this for good reason, as we have more real-estate to work with.

Not only do we want to take advantage of that to make things look more interesting, but if we didn't make things more complex on larger screens, things wouldn't look very good. Even if you have a *very* simple website, you don't want it to have text stretching from one side to the other.

Some things aren't so simple

Some components look completely different at different screen sizes, such as navigation menus. Other times, you have styles on mobile that need to be overwritten for desktop that end up being redundant**.**

Mobile First

```
/* Mobile styles */
.nav {
  position: fixed;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  padding-top: 2rem;
}
```

```
/* Desktop styles */
@media (min-width: 60rem) {
  .nav {
    position: initial;
    width: initial;
    height: initial;
    display: flex;
    align-items: center;
    padding-top: 0;
    background-color: blue;
  }
}
```

Desktop First

```
/* Desktop styles */
.nav {
  display: flex;
  align-items: center;
  background-color: blue;
}
```

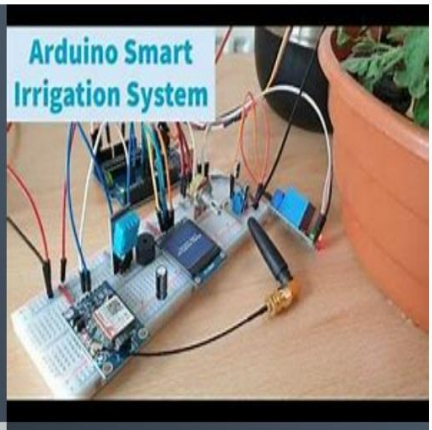
```
/* Mobile styles */
@media (max-width: 25rem) {
  .nav {
    display: block;
    position: fixed;
    background-color: initial;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    padding-top: 2rem;
  }
}
```

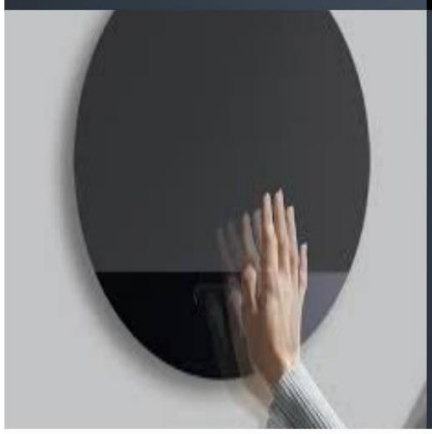
PROJECTS

Project 1

Smart irrigation system

The project presents the use of correct soil moisture sensors which helps to ease out the pain to monitor and keep records about the changes in soil moisture. Using the Arduino Mega micro controller with Light-Depended Resistor sensor, moisture sensor and temperature sensor, temperature are measured and analyzed. The soil for a certain duration, provides information related to the moisture status of the soil. The Arduino Mega will collect and process the data received from the Sensors.When a threshold moisture level of the soil is reached, the water will supply accordingly. This is essential because water must be provided to the plant at a particular time for a good yield. This project is highly use for farmers, manual method of irrigation system.





Project 2

Hand Gesture volume controller

I have done "Creating a Hand Gesture Volume Controller using Python ".Python project to control system volume either by increasing or decreasing volume using hand gestures.

CONCLUSION

Throughout this portfolio you can see that my work was a progress from the rough draft to the polished draft. All of the work I composed onto this file was the journey I went through this course. Each project has a prompt and its own individual purpose for us to seek and complete. Every material I worked on had its own different skills. Portfolio management is an effective method for organizations to manage their products through their development lifecycles, priority, gating, and consistent approaches, which are taken into consideration. Many organizations use portfolio management to ensure the right projects get selected and implemented. However, portfolio management can do a lot more than that, and it’s a useful approach for your PMO.

In today's Web development, a good page design is essential. A bad design willlead to the loss of visitors and that can lead to a loss of business. In general, a good page layout has to satisfy the basic elements of a good page design. This includes color contrast, text organization, font selection, style of a page, page size, graphics used, and consistency. In order to create a well-designed page for a specific audience.

It can be helpful when searching for a job, a freelancing gig, or showcasing your skills towards a new client. HTML, CSS and JavaScript are the foundational languages to create a website.

1. **Create the structure with HTML.** The first thing you have to learn, is HTML, which is the standard markup language for creating web pages.
2. **Style with CSS.** The next step is to learn CSS, to set the layout of your web page with beautiful colors, fonts, and much more.
3. **Make it interactive with JavaScript.** After studying HTML and CSS, you should learn JavaScript to create dynamic and interactive web pages for your users.