A Project Report

on

Personal Portfolio Website

B. Tech. (C.S.E.) 4th Semester

Submitted By:

Vikas Dwivedi (B2055R10106118)

Under the Guidance of:

Dr. Subhadra Shaw

(Assistance professor)

Department of Computer Science and Engineering (C.S.E.)

AKS University, SATNA (M.P.)

Declaration

Vikas dwivedi, hereby declare that project titled “Personal Portfolio Website” submitted by me for the partial fulfillment of the requirement for the award of Bachelor of Technology (Computer Science) to A.K.S (Amicable Knowledge Solution) University, Satna, comprises my work and due acknowledgment has been in text and all other material used**.**

Submitted By:

Vikas Dwivedi (B2055R10106118)

Certificate

This is to certify that Mr. Vikas Dwivedi worked on the Project "Personal Portfolio Website" from 2022 and has successfully completed the academic Minor Project, in order to partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science Engineering under my supervision and guidance.

-------------------------------------- -------------------------------

Internal **External**

Certificate

This is to certify that Mr. vikas dwivedi worked on the project “Personal Portfolio Website” from 2022 and has successfully completed the academic minor project, in order to partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science Engineering under my supervision and guidance.

Dr. Akhilesh A. Waoo

(H.O.D)

Acknowledgement

It is great opportunity to express thanks, respect, and gratitude to those who helped me in the different possible ways at the time of project development. I also thank the faculty members who helped me not directly but as guidance back to me. Mostly I wish to thank Prof. Dr. Akhilesh A. Waoo and Asst. prof. Dr. Subhadra Shaw, for guiding the successful completion of the project with great effort patiently, and smilingly.

At last, I’m thankful to our colleagues for their help and inspiration. During my project, I learned many things and got a lot of knowledge from this project. I have given my full efforts to this project. I wish that my project will be accepted and be successful.

Table of Contents

Introduction

It is a new term to me, My portfolio.

What is a portfolio?

In the most general term, a portfolio is documentation of professional growth and achieved Competence. Organized around a set of standards, my portfolio will help contain document that will provide tangible evidence of the wide range of knowledge, skills, and dispositions that I possess as a growing professional student.

Why is a portfolio needed?

- When I move through here and work as a developer, my portfolio will help me in several ways. My portfolio will help me keep track of how I’m doing, my area of strength, and the areas in which I need to make extra practice.
- Encourage me to reflect on the documents in my portfolio and determine how I can further develop them to demonstrate high levels of quality.
- Show the faculty that I’m making progress toward achieving the stated standards or outcomes of my program.
- Guiding me in preparing for the interviews and high stakes test.
- Help me for my job search by serving as a major method of presenting evidence of my competence to potential employers.

Design and framework

1. Software process model

As software engineering students, coding is our art. Similar to artists, many of us have a portfolio of projects. I've worked on that we might like to share. Not only is it helpful for employers, but it's also cool to show friends and family, and anyone else who might be interested.

But to share our work with people, it's not enough for us to throw our code up on Github and make a nice readme, but nowadays we have to make an actual website to showcase our projects.

Not only do design websites that people dedicate their whole careers to doing well, but also HTML and CSS can be very finicky to work with.

This is not going to be an HTML, CSS, or website design tutorial, but I'm just going to share some things I found useful that I learned while putting together my website.

As a background, I started coding 1 year ago, but only started seriously coding regularly this semester and a bit during this past summer. Although I have a little bit of front-end experience, I do not consider myself an expert by any means, and the things I mention in this report should be digestible for anyone who's a farm with the basics of HTML and CSS.

2. Why do I even need a site for my projects?

I don't need a personal site for my projects. I also don't take my resume look into nice, or fill out my LinkedIn profile, but I do those things for my own knowledge.

It's common for employers to a quick look at a resume, and no way are they going to download my code from Github and run it. If they only spend on average less than 10 seconds on my actual resume, I think they're going to click through all my repositories to see my readmes? Now I had, a page with images or gifs showing off my project's functionality, a recruiter could glance at it in the milliseconds that they were going to give it anyway, but this time they'll be able to see the results of my hard work, and the types of projects I like to work on.

Nowadays a lot of applications have an option for personal websites and other media links because these things help to get a better sense of who an applicant is. Having a personal portfolio website is a great opportunity to bring life to the work you’ve done.

3. Where do I start?

So, before I start making it, I need to know what I want it to look like. The average programmer doesn’t just have amazing design ideas to just throw around so, I have some inspiration. For this, you can go to places like dribble.

After that, I tried to draw it out on a piece of paper. Something like Figma is arguably better, but my site was going to be very simple, so pen and paper was convenient and good enough.

Some people can manage it all in their minds, but I like to have at least a simple outline that I can reference along the way.

4. My general approach to the design

I just wanted a site that showed my work, with little to nothing else.

People like to have very fancy pages that look cool, but I focused more on just making something that was concise, robust, and easy on the eyes.

Maybe my design itself won’t impress anybody, but the main focus is learning my work, and I think my design reflects that.

My first attempt at making it was actually kind of the opposite. Although I wanted to make the aesthetic of the design simple, I was going for a fancier layout. After an hour or two of implementing that, I got sick of it and threw something together that was just meant to showcase my projects. I ended up liking it a lot better, and that’s the one I’ve stuck with.

5. Actual design

For the design, the more complex and fancier I want my site to be, the more difficult it will be to implement it. Animations aren’t too complex with just raw HTML/CSS, but I chose to just have everything be static.

For the actual aesthetics, I think getting just two things right can go a long way:

1. A nice font, or pair of fonts.
2. A simple color on your pages.

I don’t think the layout of my website is anything special, but I think it looks clean because the fonts and colors go well together.

I kept it as basic as I could, so I went with Lobster cursive as my primary font, and I chose Roboto Mono as a secondary font to give a bit of a typewriter-Esque coder feel.

I think a good pair of two different font style (from google fonts) can go a long way in making something look good.

To get free fonts, and even find font pairings, it’s all on google fonts and they have options for you to just import it directly into CSS, which is what I used.

For my colors, I can never go wrong with white, black, and grey. I want to have colors; I’m choosing a single accent color and sticking to neutrals when in doubt.

For these visual aspects, experimenting, inspiration, and copying will be my best friends to make a unique website.

HTML/CSS tips and tricks

At this point, I might be thinking to myself, “vikas, I don’t care about myself experience, I’ve spent my whole career trying to center a div, and I still can’t do it.” Well, I’m finally getting to it.

1. Centering things

For actually centering things, I like to use the classic

```
element {
margin: auto;
}
```

Or put the element inside of a div and use flex-box.

```
div {
display: flex;
justify-content: center;
align-item: center;
}
```

It should get 80 percent of my centering struggles.

Units of measurement

Something I’ve always been confused about is when to use different units of measurement. Below are my personal preferences for these things, which I’ve found through experimentation. They are likely not what a professional front-end dev or UI/UX designer would say, but I experienced this things.

% — percent relative to the parent of an element. I like to use this very liberally for the width and height of elements because I think it makes sense for the size of things to scale with their parents.

1. **vw, vh** (view width and view height), these changes both between devices and if you change the size of your window. I use these for big picture spacing, such as the size of a div containing the majority of my content for a page.
2. **em, rem**, not 100 percent sure what each stand for but I think of them as element font size, and root **e**lement font size, this scale primarily between devices. I like to use rem for font sizes and some minor spacing that I want to vary between devices. I don't use em, but I can see where it'd be helpful if I wanted to scale the fonts of a single element all at the same time, e.g., if I wanted to scale all the text in my project sections, I could use them for the elements inside, and change the overarching parent div.'s font size.
3. **px**(pixels), Other resources will call this an absolute sizing metric, but it does scale between screens of different dpi. My use case is similar to rem, where I want to have things remain static sizing on a single device, but adapt for others. I use px most often when I know that a viewport is going to be large enough for this pseudo-absolute sizing metric.

4. Sizing of Images

```
    img {
 border-radius: 50%;
}
```

Nobody wants a squashed version of their image. The percent will fit an image's size to its container, and auto for height or width will make it scale to the image's original aspect ratio.

5. Flexbox

Aside from being able to center things, flexbox is a godsend for the layout of website.

Media queries and scaling your layout

I doubt that employers will be viewing my portfolio on a phone or tablet, but I'm releasing my site to the general public, it may be worth putting in a little extra effort to have the layout of my website accommodate different devices.

I just stuck to changing the layout based on different viewport widths, but there are also features to change upon different viewing devices.

I don't have to rewrite all of my styling for each media query, only what I want to change. This is all I have in the stylesheet:

```
@importURL('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
   list-style: none;
     }
```

This just tells the page that when the screen width is below, apply these styles, i.e., change the flex-direction, box-sizing, adjust image sizing, and tweak some spacing.

Icons

For icons, I used font-awesome. They have tons of icons that scale with a font size that are easy to import into page.

And I learned about this through an important technique inspecting elements (or ctrl+shift+I) on other people’s web pages. Just a little bit of googling later I had some awesome icons to use on my website.

Coding:

```
                        <div class="text-center m-top-50">
s
                   <a  class="btn  line-btn-dark  btn-icon  btn-radius”  href="vikas.docx"
title="Resume"<i class="fa fa-download" download></i> Download Resume</a>
        </div>
```

Here, I’m using a div attribute. inside div I used anchor tag href .with class name btn line-btn-dark bin-icon btn-radius.

The functionality of the class is to make a clickable button with a downloadable file.

```
    <img src="vicky.png" alt="developer image “class="center">

    </div>
```

Img tag is used for linking the image file inside the code.

```
<div class="topnav">

<nav>

<a class="click-me" href="index.html"><i class="fa fa-home"></i> Home</a>

<a class="click-me" href="blogs.html">Blogs</a>

<a class="click-me" href="about.html">About Me</a>

</nav>

</div>
```

Div class topnav is used for linking the pages from one to another.

Inside div, I used an anchor tag for linking the file.

```
<style>
      h1{




         color: rgba (238, 28, 157, 0.76);

         margin-left: 80vh;

         margin-bottom: auto;

     }


     h2{

         color: rgba (221, 118, 22);

         margin-left:10vh;

     }

      p {

    <style>

      h1{




         color: rgba (238, 28, 157, 0.76);

         margin-left: 80vh;

         margin-bottom: auto;

     }
    h2{
       color: rgba (221, 118, 22);
       margin-left:10vh;
    }
     p {
 @import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
       font-family: 'Lobster', cursive;
       color: grey;
       margin: 5 5 3 vh;
       margin-left: 10vh;
    }




    body {


    height: 100%;



/* The image used */




/* Full height */
height: 100%;


/* Center and scale the image nicely */
```

```
 background-position: center;

 background-repeat: no-repeat;

 background-size: cover;



        background-image: url('cool-background.png');


        }


        a {

              color: rgba (230, 15, 122);

               margin-left: 10vh;

          }



           a: hover {

                background-color: rgba (33, 128, 33);

           }

</style>





       body {


       height: 100%;
```

```
/* The image used */



/* Full height */
height: 100%;

/* Center and scale the image nicely */
background-position: center;
background-repeat: no-repeat;
background-size: cover;



       background-image: url('cool-background.png');

       }

       a {
             color: rgba (230, 15, 122);
              margin-left: 10vh;
         }



          a: hover {
               background-color: rgba (33, 128, 33);
          }

</style>
```

It’s a code for blog page style.

This is a CSS code that styles the page.

Here, I used to hover for adding a hover effect on my page.

I’m using multiple colors for different text sizes.

```
 <style>


img {

        border-radius: 50%;

        }


 h1{

              color: rgba (233, 158, 20, 0.911);

              margin-left: 10vh;


           }

 body {


   /* Full height */

      height: 100%;


  /* Center and scale the image nicely */
```

```
     background-position: center;

      background-repeat: no-repeat;

       background-size: cover;


     background-image: url('background.jpg');

    }




. btn {

  background-color: rgba (29, 165, 29);

  border: none;

  color: white;

  padding: 10px 25px;

  cursor: pointer;

  font-size: 15px;

}


/* Darker background on mouse-over */

. btn: hover {

  background-color: Royal Blue;

}

</style>
```

It is the CSS code for the index page.

Here, I used the background color for multiple headings and

Paragraph.

```
<div   class="centerdiv">
   <a href="https://web.whatsapp.com/"target="_blank">
   <i class="fa-brands fa-whatsapp">whatsapp<br> 8236014408</i>
   </a>

<ahref="https://www.facebook.com/profile.php?id=100069380677113"target="_bl
ank">
      <i class="fa-brands fa-facebook">facebook</i>
      </a>

  <ahref="https://www.linkedin.com/in/vikas-dwivedi-1a0b27233"target="_blank">
         < i class="fa-brands fa-linkedin">linkedin</i>
         </a>

                                                                            <a
href="https://www.instagram.com/direct/340282366841710300949128269260623
90513"target="_blank">
         <i class="fa-brands fa-linkedin">instagram</i>
         </a>
</div>
```

Here, I link my multiple social media profiles.

By clicking on these icons or text you can land on my social media profiles.

```
<style>
       h1{
           color: rgb(235, 8, 8);
           margin-left: 15cm;
           margin-top: 5vh;
           margin-bottom: 10vh;
       }
         body{
         background-image: url('wall.jpg');
         background-size: cover;
         background-repeat: no-repeat;

          }
          img{
              /* margin-left: 140vh; */
              margin-right: 30vh;
              margin-top: 10vh;
              position: absolute;
                top: 5px;
                 right: 5px;

              width: 300px;
          }
```

```
        a{
            color: #00ff88;
            margin-left: 10vh;
        }
        h2{
            color: rgb(226, 109, 109);
            margin-left: 10vh;
        }
        h3{
            color: #ff9900;
        }
      </style>
```

It’s a CSS code for the about me page.

Here, I’m giving the styling for h1, body, and image.

```
@Import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
@Import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    list-style: none;
```

```
}

body {
    background-color: var (--color-primary);
    font-family: ‘Lobster', cursive;
    font-size: 1.1rem;
    color: var (--color-white);
    transition: all .4s ease-in-out;



a {
    display: inline-block;
     text-decoration: none;
     color: inherit;
     font-family: inherit;
}
 header {
    height: 100vh;
     color: var (--color-white);
    overflow: hidden;
    color: #ddd;
 }
 p {
```

```
color: #ddd;

padding: 5vw;

 }

/* Add a black background color to the top navigation */

. topnav {

   background-color: #333;

   overflow: hidden;

  }

 /* Style the links inside the navigation bar */

 . topnav a {

   float: left;

   color: #f2f2f2;

   text-align: center;

   padding: 14px 16px;

   text-decoration: none;

   font-size: 17px;

  }

 /* Change the color of links on hover */

 . topnav a: hover {

   background-color: #ddd;

   color: black;

  }

 /* Add a color to the active/current link */

 . topnav a. active {

   background-color: #04AA6D;

   color: white;
  }


. btn {
    color: hsl (270, 16%, 87%);
    background-color: #42e211;
    margin-left: 10vw;
    margin-bottom: 14vh;
}
. btn: hover {
    background-color: Royal Blue;
  }
  . button: hover {
      background-color: royal blue;
  }


section {
    min-height: 100vh;
    width: 100%;
    position: absolute;
    left: 0;
    top: 0;
    padding: 3rem 18rem;
}
. section {
    transform: translate (-100%) scale (0);
    transition: all .4s ease-in-out;
    background-color: var (--color-primary);
}
. controls i {
    background-color: rgba (252, 122, 0);
    font-family: 'Lobster', cursive;
}
 . controls h2, h3, h4 {
    color: rgba (252, 122, 0);
    padding-left: 5vw;
    margin: auto;
    font-family: 'Lobster', cursive;
}
```

Problem Statement:

A problem statement is a concise description of an issue to be addressed or a condition to be improved upon. It identifies the gap between the current (problem) state and desired (goal) state of a process or product. Focusing on the facts, the problem statement should be designed to address the Five Ws. The first condition of solving a problem is understanding the problem, which can be done by way of a problem statement. Problem statements are widely used by most businesses and organizations to execute process improvement projects. A simple and well-defined problem statement will be used by the project team to understand the

problem and work toward developing a solution. It will also provide management with specific insights into the problem so that they can make appropriate project-approving decisions. As such, the problem statement must be clear and the problem, the statement is to identify and explain the problem. This includes describing the existing environment, where the problem occurs, and what impacts it has on users, finances, and ancillary activities. Additionally, the problem statement is used to explain what the expected environment looks like. Defining the desired condition provides an overall vision for the process or product. It makes clear the purpose for initiating the improvement project and the goals that it is meant to accomplish.

Another important function of the problem statement is to be used as a communication device. The problem statement helps with obtaining buy-in from those involved in the project. Before the project begins, the stakeholders verify the problem and goals are accurately described in the problem statement. Once this approval is received, the project team reviews it to ensure everyone understands the issue at hand and what they are trying to accomplish. This also helps define the project scope, which keeps the project concentrated on the overall goal.

The problem statement is referenced throughout the project to establish focus within the project team and verify they stay on track. At the end of the project, itis revisited to confirm the implemented solution indeed solves the problem. A well-defined problem statement can also aid in performing root-cause analysisto understand why the problem occurred and ensure measures can be taken to prevent it from happening in the future.

It is important to note that the problem statement does not define the solution or methods of reaching the solution. The problem statement simply recognizes the gap between the problem and goal states. It can be said that "a problem well stated is half solved".[4] However, there are often multiple, viable solutions to aproblem. Only after the problem statement is written and agreed upon shouldthe solution(s) be discussed and the resulting course of action determined.

Before the problem statement can be crafted, the problem must be defined. It is human nature to want to begin working on a solution as soon as possible and neglect the definition of the true problem to be solved. However, a poorly defined problem increases the risk of implementing a solution that does notfully meet the expected results. A problem cannot be solved if it is not completely understood.

The process of defining the problem is often a group effort. It starts with meeting with the stakeholders, customers, and/or users affected by the issue (if possible) and learning about their pain points. Since people often struggle with effectively communicating their issues, particularly to someone outside of the process, it is helpful to ask a series of "why" questions until the underlying reasoning is identified. This method, known as the "5 Why's", helps drill down to the core problem as many of the experienced frustrations could be mere symptoms of the actual problem. Asking these additional questions as well as paraphrasing what the stakeholder had said demonstrates a degree of empathy and understanding of the problem.

The information collected from these initial interviews is only one part of analysis. Many times, the problem extends to multiple areas or functions to which the stakeholders, customers, and users are unaware. They may also be familiar with what is happening on the surface but not necessarily the underlying cause.

Technologies used:

HTML: HTML stands for Hypertext Markup Language. It allows the user to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications.

HTML is not a programming language, meaning it can’t create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word.

When working with HTML, we use simple code structures (tags and attributes) to mark up a website page. For example, we can create a paragraph by placing the enclosed text within a starting ***<p>*** and closing ***</p>*** tag.

HTML was invented by Tim Berners-Lee, a physicist at the CERN research institute in Switzerland. He came up with the idea of an Internet-based hypertextsystem.

Hypertext means a text that contains references (links) to other texts that viewers can access immediately. He published the first version of HTML in 1991, consisting of 18 HTML tags. Since then, each new version of the HTML language came with new tags and attributes (tag modifiers) to the markup.

According to Mozilla Developer Network’s HTML Element Reference, currently, there are 140 HTML tags, although some of them are already obsolete(not supported by modern browsers).

Due to a quick rise in popularity, HTML is now considered an official web standard. The HTML specifications are maintained and developed by the World Wide Web Consortium (W3C). You can check out the latest state of the language is biggest upgrade of the language was the introduction of HTML5 in 2014. Itadded several new semantic tags to the markup, that reveal the meaning of their content, such as <article>, <header>, and <footer>.

CSS: Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language suchas HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, and enable multiple web pages to share formatting by specifying the relevant in a separate. CS file which reduces complexity and repetition in the structural content as well as enables' S file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), andon Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

JavaScript: JavaScript is a lightweight, cross-platform, interpreted scriptinglanguage. It is well-known for the development of web pages; many non-browserenvironments also use it. JavaScript can be used for Client-side developments aswell as Server-side developments. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

Client-side: It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse

clicks, form input, and page navigation. Use full libraries for the client-side are AngularJS, ReactJS, VueJS and so many others.

Server-side**:** It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is node.js.

It was created in 1995 by Brendan Eich while he was an engineer at Netscape. Itwas originally going to be named LiveScript but was renamed. Unlike most programming languages, the JavaScript language has no concept of input or output. It is designed to run as a scripting language in a host environment, and it is up to the host environment to provide mechanisms for communicating with the outside world. The most common host environment is the browser.

System Configuration:

Hardware / Software Requirements:

Hardware Requirement:

1. CPU
2. Monitor/ Projector
3. Keyboard
4. Mouse
5. 512MB RAM (Minimum)
6. Minimum HDD/SDD based on your OS.
7. Any Device with an internet connection via the browser to access a website.

Software Requirement:

1. Browser.
2. Any Operating System

Features of a project:

1. Design

The design I choose can mean the difference between driving most customers away and inspiring most of them to bite. This is especially true because as a web designer as my website *is* an example of my work. Minimalism is the way to go when it comes to portfolio sites. I'm trying to attract new customers with my work; therefore, I should choose a design that draws attention *toward* my portfolio and any information I list about my work, not away from them.

I don't need to choose a white or bright color scheme, but I should use whitespace and stick to a maximum of three fonts. My color scheme should contrast well, and my typography should have a dramatic flair that draws my eye to it. Also, I'm not trying to use animations that don't serve a purpose.

2. Logo

This doesn't need to be complicated, and I can use a simple text-based logo that displays my menus. What's important is my line as it can describe myself and the services I provide with a quick, short phrase. This provides a quick way for visitors to decide if my services are t for them before they contact me. Try to be short yet descriptive with my tagline. "Developer" is descriptive, but it's not as descriptive as "WordPress developer." In the same sense, "New York-based graphic designer" is more descriptive than "graphic designer."

3. Call to action on the page

Refer back to the decision I made when I determined the purpose of my portfolio site. It'll help me come up with calls to action for my site, particularly on the homepage. I'm trying t land more clients for my s e. Using "Request a Quote" as an action would be appropriate.

Determine the purpose my portfolio site serves, and the action I want interviewers to take on landing pages makes sense.

4. High-Quality Images

This ties into the first point I made about design. Always make sure I'm using the highest quality of images to represent my work. This isn't only true for photographers, artists, and graphic designers. Even if I don't create visual works, I should still be using images for presentation purposes, and those images *should be* high in quality.

The images should also complement the rest of my site. It uses a dark color scheme, which complements the black-and-white photography in the demo's portfolio beautifully.

5. Services

I provide services, on my site, you can navigate from the home page to multiple pages like blogs page and about page.

you can download my resume.

6. Contact Information

customers don't contact me for quotes to makes it impossible for them to go about doing that. Make things easy for them by inserting contact details on my homepage or a link to my Contact page, at the very least.

I do place contact info on the homepage, I'm also including business email addresses on the page

7. Information About Myself

Some would love to learn my professional story about how I learned to do whatever it is.

8. Content

Producing content related to my page. I think it is the best way to establish myself as an authoritative presence on a particular topic.

Blogs are the traditional way to go, but what happens if I'm not a strong writer? so, I can speak better than I write,

9. A High-Quality Image

This is optional, but it may help clients connect with you better if you share a picture of yourself on your website. Unless everything you post is highly professional, I wouldn't insert my Instagram feed on my site. Instead, opt for a high-quality image, and placed on homepage, About page, or both.

10. Relevant Social Media Accounts

Speaking of Instagram, I use my social media accounts professionally, Not only is it another form of marketing, but it can also provide yet another way for you to look like an authority in your niche.

Testing:

1. Desk checking

Desk checking is an informal manual test that programmers can use to verify coding and algorithm logic before a program launch. This enables developers to spot some errors that might prevent a program from working as it should. The modern debugging tool makes desk checking less essential than it was in the past, but it can still useful way of spotting logic errors.

2. Code complexity

Code complexity is defined as the amount of effort needed to understand and maintain the code properly. Complexity is often relativity to the modeled being is more complex than the long one. This is recommended to have a combination of code size metrics and complexity metrics to detect code complexity.

3. Pair Testing:

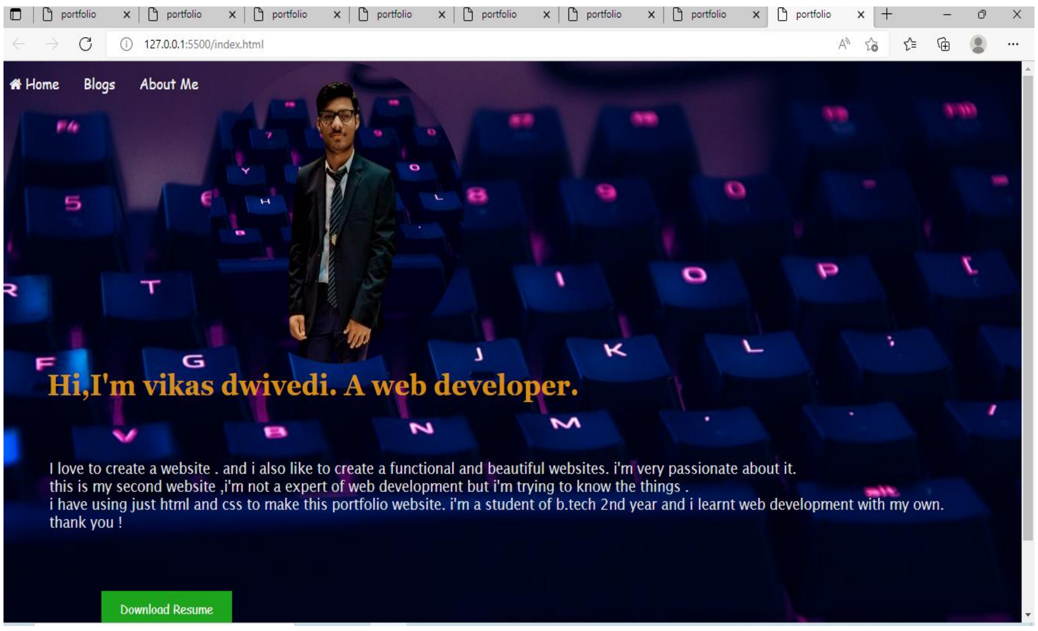
Pair testing is a software testing technique in which two people test the same feature at the same place at the same time by continuously sharing their ideas. It generates more ideas where resultants in better testing of the application under test.

Output screen

The interface on which users can Select:

1. **Home page**
2. **Blogs page**
3. **About me page**

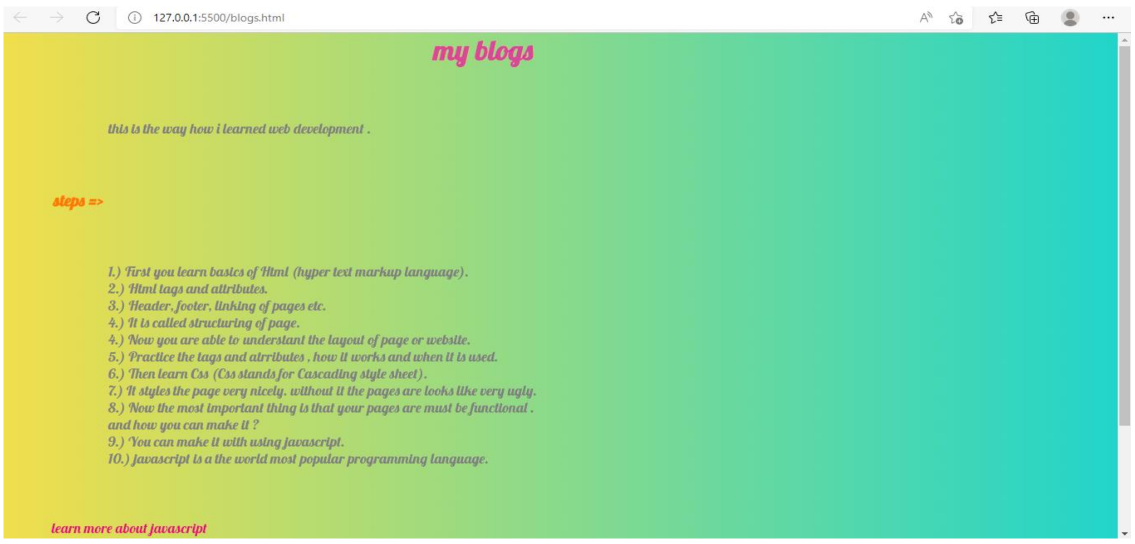
First Screen Interface:



Hi,I'm vikas dwivedi. A web developer.

I love to create a website . and i also like to create a functional and beautiful websites. i'm very passionate about it.
this is my second website ,i'm not a expert of web development but i'm trying to know the things .
i have using just html and css to make this portfolio website. i'm a student of b.tech 2nd year and i learnt web development with my own.
thank you !

Second Screen Interface:



127.0.0.1:5500/blogs.html

my blogs

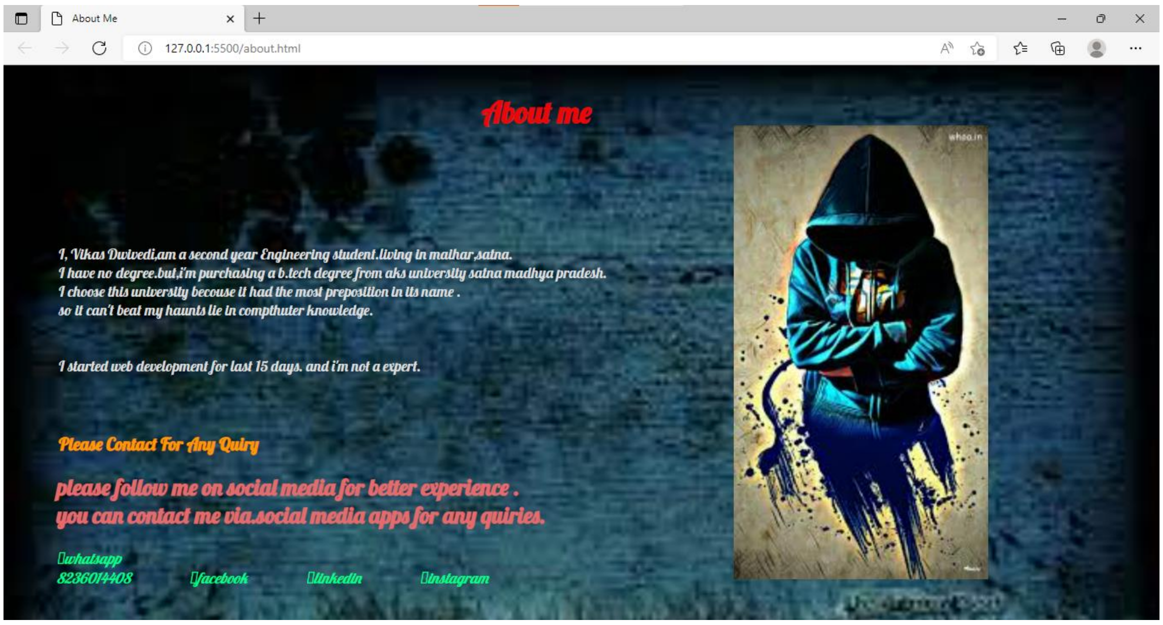
this is the way how i learned web development .

steps =>

1.) First you learn basics of Html (hyper text markup language).
2.) Html tags and attributes.
3.) Header, footer, linking of pages etc.
4.) It is called structuring of page.
4.) Now you are able to understant the layout of page or website.
5.) Practice the tags and atrributes , how it works and when it is used.
6.) Then learn Css (Css stands for Cascading style sheet).
7.) It styles the page very nicely. without it the pages are looks like very ugly.
8.) Now the most important thing is that your pages are must be functional .
and how you can make it ?
9.) You can make it with using javascript.
10.) javascript is a the world most popular programming language.

learn more about javascript

Third Screen Interface:



About me

I, Vikas Dwivedi,am a second year Engineering student.living in malhar,satna.
I have no degree.but,i'm purchasing a b.tech degree from aks university satna madhya pradesh.
I choose this university becouse it had the most preposition in its name .
so it can't beat my haunts lie in compthuter knowledge.

I started web development for last 15 days. and i'm not a expert.

Please Contact For Any Quiry

please follow me on social media for better experience .
you can contact me via.social media apps for any quiries.

whatsapp
8236014408 facebook linkedin instagram

1. Future Enhancement:

1.) In the future, I can add more pages to my website.

2.) And I can make my website dynamic.

3.) I can also add features, which make my website more efficient and functional.

4.) I can connect the database in the future.

5.) I can make a contact me page in future. And on the contact me page I can store the information of an user.

6.) In the future, I can also use JavaScript, jQuery, and many more languages to make my website more functional etc.

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 in this file was the journey I went through during this course. Each project has a prompt and its purpose for us to seek and complete.

Every material I worked on had its different skills.

For a while, I'd put off making my website because it felt like such a daunting task, but when it came down to it, making a static website is only a matter of formatting content on a page. It can be a bit of labor, but I'll be proud of my work by the end of it. I hope this could help.

THANK YOU