NATIONAL INSTITUTE OF TECHNOLOGY SURATHKAL

IT-254 WTA Lab

Responsive Web Design and Server Side Programming

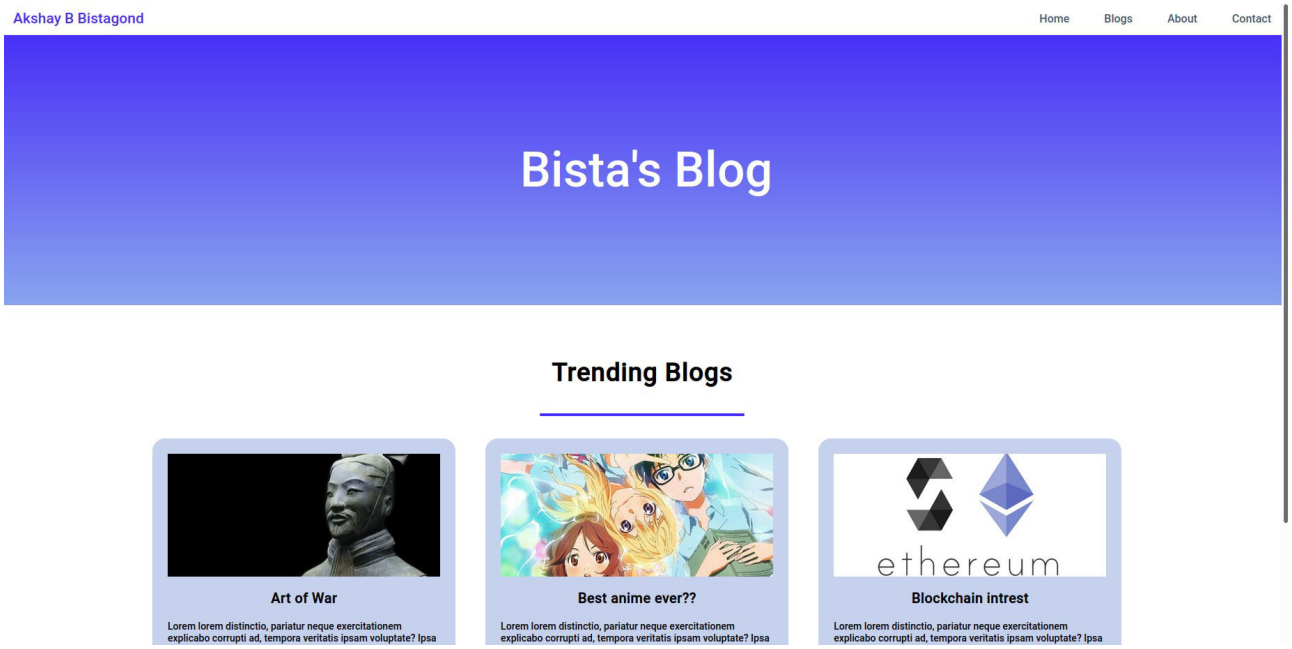
Submitted By-
Akshay B Bistagond
191EC203

Responsive Web Design

Problem 1: Design a blog of your choice (food blog, travel blog, your life experience etc). The website must include a good set of pictures and descriptions (recipe or describing the place) for the pictures. The descriptions can be added in a new page (as in the clickable image assignment you did in basic html). The blog should have a good layout with proper Heading, Sidebars for additional links, Navigation Bars, Footer element. The blog is expected to be responsive when it is accessed on a mobile device, tablet and laptop. Consider the different states and breakpoints for triggering the responsive style (use google chrome's developer tools option for emulating a tablet, laptop and mobile device). The blog must look realistic.

Home Page:-

Desktop view -



Mobile Veiw -

Akshay B Bistagond

Bista's Blog

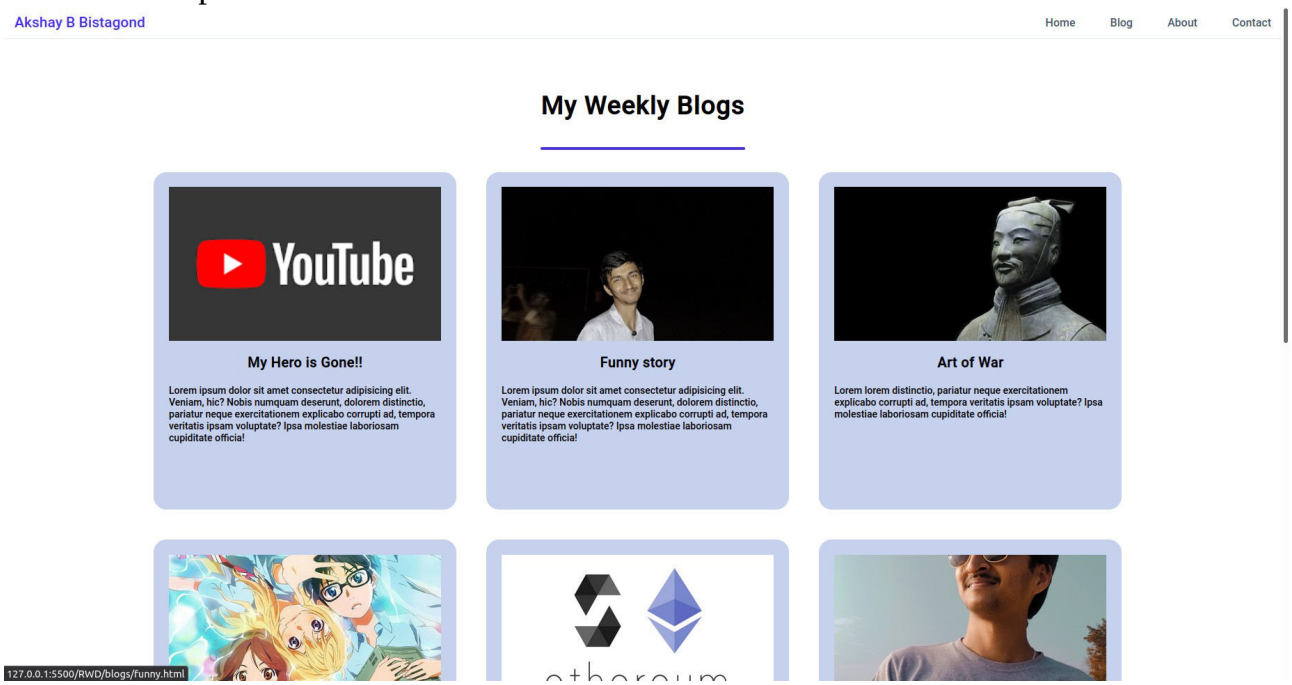
Trending Blogs

Art of War

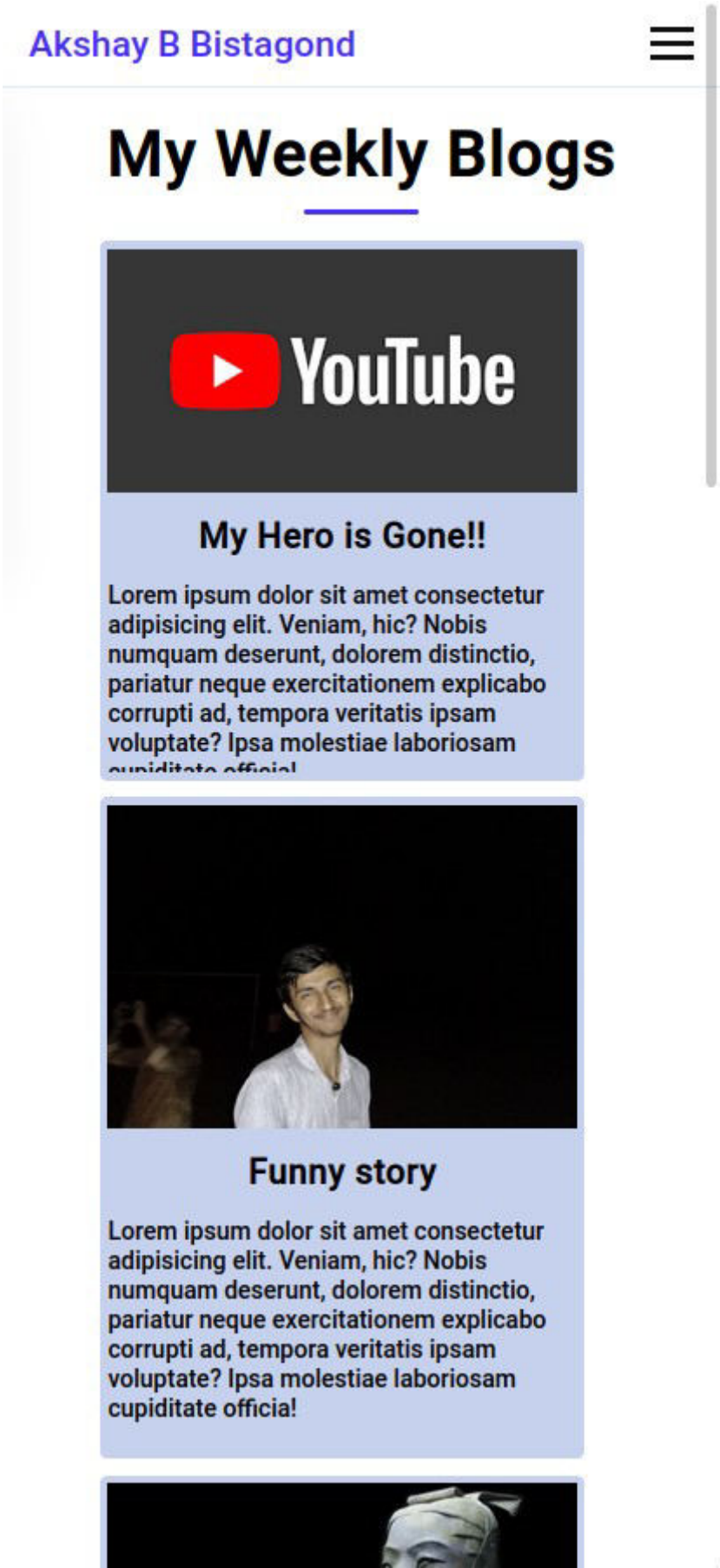
Lorem lorem distinctio, pariatur neque exercitationem explicabo corrupti ad, tempora veritatis ipsam voluptate? Ipsa molestiae laboriosam cupiditate officia!

Blogs Page:-

Desktop view -

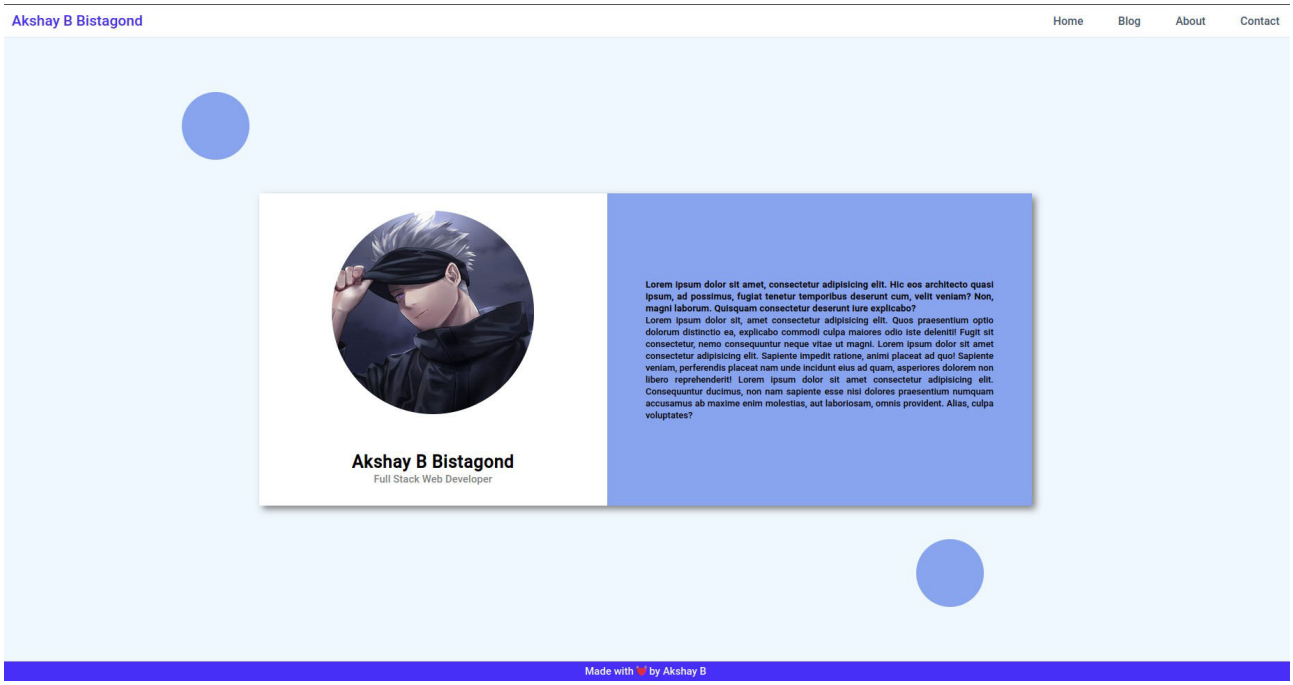


Mobile Veiw -


Akshay B Bistagond
My Weekly Blogs
YouTube
My Hero is Gone!!
Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam, hic? Nobis numquam deserunt, dolorem distinctio, pariatur neque exercitationem explicabo corrupti ad, tempora veritatis ipsam voluptate? Ipsa molestiae laboriosam
Funny story
Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam, hic? Nobis numquam deserunt, dolorem distinctio, pariatur neque exercitationem explicabo corrupti ad, tempora veritatis ipsam voluptate? Ipsa molestiae laboriosam cupiditate officia!

About Page:-

Desktop view -

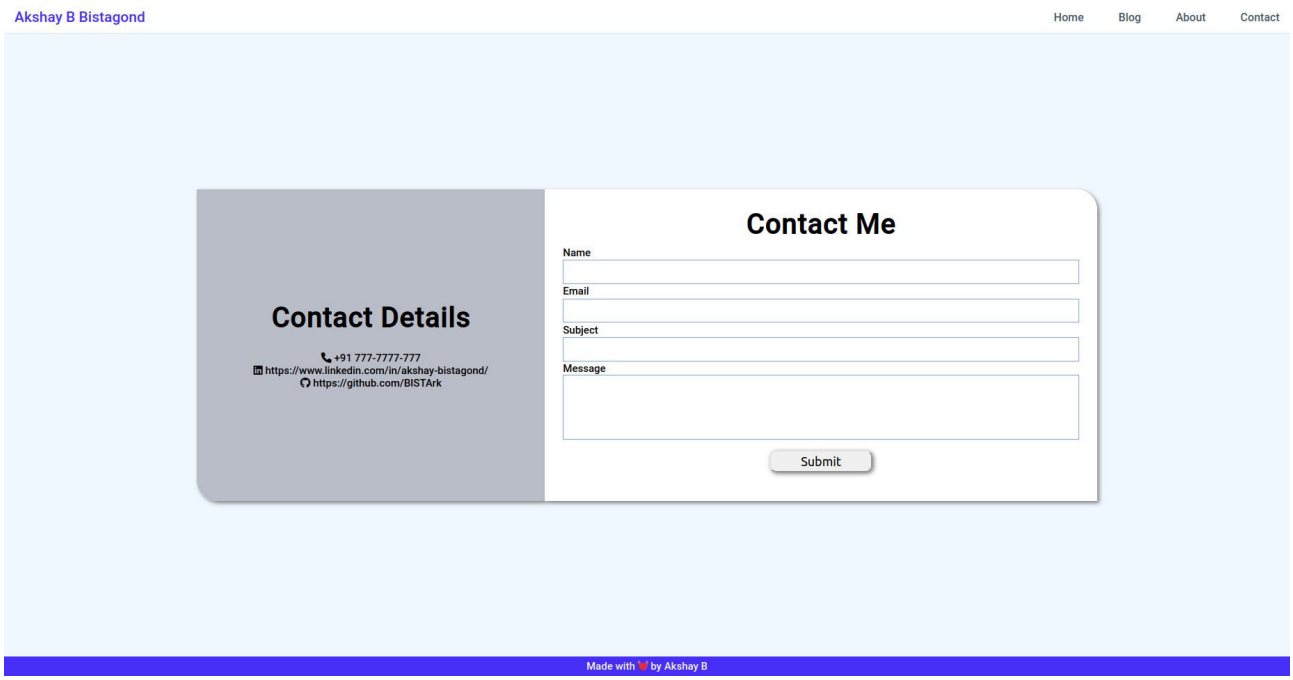


Mobile Veiw -

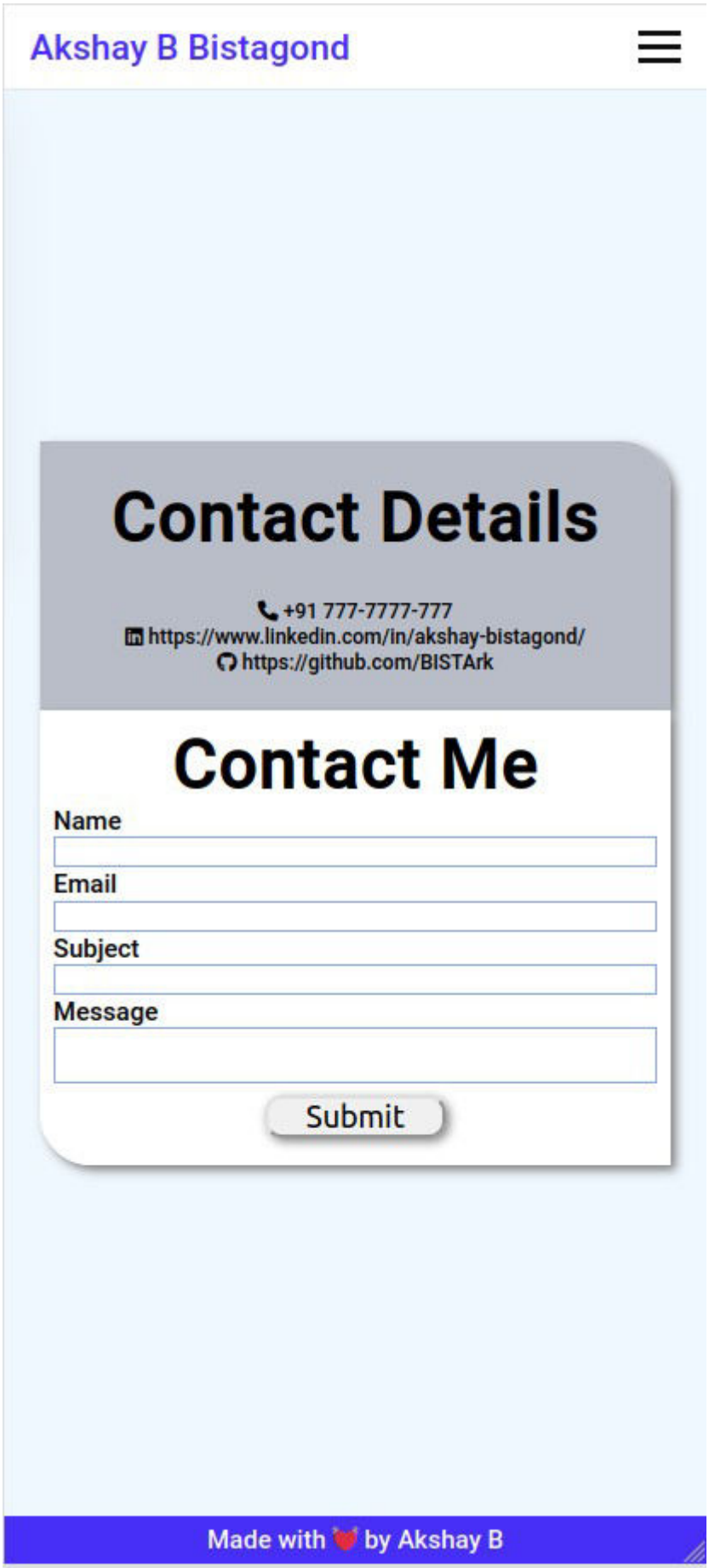

Akshay B Bistagond
Akshay B Bistagond
Full Stack Web Developer
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Hic eos architecto quasi ipsum, ad possimus, fugiat tenetur temporibus deserunt cum, velit veniam? Non, magni laborum. Quisquam consectetur deserunt iure explicabo?
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Quos praesentium optio dolorum distinctio ea, explicabo commodi culpa maiores odio iste deleniti! Fugit sit consectetur, nemo consequuntur neque vitae ut magni. Lorem ipsum dolor sit amet consectetur adipisicing elit. Sapiente impedit ratione, animi placeat ad quo! Sapiente veniam, perferendis placeat nam unde incidunt eius ad quam, asperiores dolorem non libero reprehenderit! Lorem ipsum dolor sit amet consectetur adipisicing elit. Consequuntur ducimus, non nam sapiente esse nisi dolores praesentium numquam accusamus ab maxime enim molestias, aut laboriosam, omnis provident. Alias, culpa voluptates?
Made with by Akshay B

Contact Page:-

Desktop view -



Mobile Veiw -


Akshay B Bistagond
Contact Details
+91 777-7777-777
https://www.linkedin.com/in/akshay-bistagond/
https://github.com/BISTArk
Contact Me
Name
Email
Subject
Message
Submit
Made with by Akshay B

Media Queries used -

```css
@media only screen and (max-width: 1024px){
    .blogcard{
        width: 45%;
        max-height: 50vh;
        margin: 1.5%;
        padding: 1.5%;
        background-color: #c6d1ee;
        display: flex;
        flex-direction: column;
        border-radius: 1vw;
    }

}


@media only screen and (max-width: 768px){
    .blogcard{
        width: 90%;
        max-height: 50vh;
        margin: 1.5%;
        padding: 1.5%;
        background-color: #c6d1ee;
        display: flex;
        flex-direction: column;
        border-radius: 1vw;
    }

}
```

```css
@media only screen and (max-width: 768px) {
    .nav-menu {
        position: fixed;
        left: -100%;
        top: 5rem;
        flex-direction: column;
        background-color: #fff;
        width: 100%;
        border-radius: 10px;
        text-align: center;
        transition: 0.3s;
        box-shadow:
            0 10px 27px rgba(0, 0, 0, 0.05);
    }

    .nav-menu.active {
        left: 0;
    }

    .nav-item {
        margin: 2.5rem 0;
    }

    .hamburger {
        display: block;
        cursor: pointer;
    }

    .hamburger.active .bar:nth-child(2) {
        opacity: 0;
    }

    .hamburger.active .bar:nth-child(1) {
        transform: translateY(8px) rotate(45deg);
    }

    .hamburger.active .bar:nth-child(3) {
        transform: translateY(-8px) rotate(-45deg);
    }
}
```

```css
@media only screen and (max-width: 1024px){
    .blogcard{
        width: 45%;
        max-height: 50vh;
        margin: 1.5%;
        padding: 1.5%;
        background-color: #c6d1ee;
        display: flex;
        flex-direction: column;
        border-radius: 1vw;
    }

}


@media only screen and (max-width: 768px){
    .blogcard{
        width: 90%;
        max-height: 50vh;
        margin: 1.5%;
        padding: 1.5%;
        background-color: #c6d1ee;
        display: flex;
        flex-direction: column;
        border-radius: 1vw;
    }

}
```

```css
@media only screen and (max-width: 1024px) {

    body{
        height: auto;
    }

    .about{
        margin: 0 5vw;
    }

  .aboutcard {
    flex-direction: column;
  }

  .mypic {
    width: 30vw;
    height: 30vw;}

  .picname {
    flex: 1;
    display: flex;
    align-items: center;
    width: 100%;
    flex-direction: column;
    background-color: white;
  }

  .desc {
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    flex: 1;
    padding: 5%;
    width: 100%;
    background-color: #89a4ee;
  }
}
```

Problem 2:

a) Demonstrate a simple HTTP communication between client and server in node.js. Make the http service is available at port 9000. When the client requests for the service at port 9000, the Server has to send an appropriate message to the client.

```
const http = require('http')

http.createServer((req,res)=>{
    res.write("HALLO!! Welcome to Hello World!!");
    res.end();
}).listen(9000);
```

b) Demonstrate a simple HTTP communication between client and server in node.js. Make the http service available at port 9001. When the client requests the service at port 9001, the server has to read and send the message stored in read.txt to the client. Assume any welcome note in hello.txt.

```
const http = require('http');
const fs = require('fs');

http.createServer((req,res)=>{
    fs.readFile('hello.txt','utf-8',(err,data)=>{
        if(err){
            console.log("No such file exist!");
        }
        res.write(data);
        res.end();
    })
}).listen(9001);
```

c) Create 3 html files (with relevant content) in the server. The client will request for a file (by specifying the URL : localhost:portnumber/home.html). The server must respond accordingly.

```
const http = require("http");
const url = require("url");
const fs = require("fs");

http
  .createServer((req, res) => {
    let myPath = url.parse(req.url).pathname;

    if(myPath === '/')myPath = '/home';
    fs.readFile(`./htmls${myPath}.html`, "utf-8", (err, data) => {
      if (err) {
        res.writeHead(404, { "Content-Type": "text/html" });
        res.write(data);
      } else {
        res.writeHead(200, { "Content-Type": "text/html" });
        res.write(data);
      }
      res.end();
    });
  })
  .listen(9002);
```