Image component:

<div className={styles.card}>

                <Image

                    src='/img/Avocado.png'

                    width="100%"

                    height="100%"

                    layout='fill'

                    objectFit="cover"

                    alt=""/>

            </div>

Layout fill will ultimately takes all space, to get provide space, make it parent component which is card relative

card{

    flex: 0.5;

    padding: 100px;

    padding-right: 0;

    display: flex;

    flex-direction: column;

    justify-content: space-around;

    position: relative; /\*for image fill\*/

}

Using Static Data in Nextjs:

import Head from 'next/head'

import Image from 'next/image'

import Footer from '../components/Footer'

import Intro from '../components/Intro'

import Navbar from '../components/Navbar'

import Services from '../components/Services'

import { data } from '../data'

import styles from '../styles/Home.module.css'

export default function Home({services}) {

  //console.log(services)

  return (

    <div>

      <Head>

        <title>Agency Website</title>

        <meta name="description" content="Made using Nextjs" />

      </Head>

      <Intro/>

      <Services services={services}/>

    </div>

  )

}

//Fetching data before rendering above components

//Below part should be done on that page where we

//want to show that data, here we show cards on services page

export const getStaticProps = async () => {

  // if api

  //  const services =  await axios.get('/', req,res,)=>{

  // }

  //Static data from file

  const services = data

  return {

    props : { services }

  }

//Now we use these props immediately in this page as shown in home function above

//Pass these services props in services components and thus we can use this in services component

}

To hover some inside class when hovering parent class:

.service:hover .media{

    transform: scale(1.1);

}

Showing text or desc when hovering over image or div:

div className={styles.service}>

                            <div className={styles.desc}>{service.desc}</div>

                            <span className={styles.cat}>

                                {service.title}

                            </span>

                            <div className={styles.media}>

                                {service.video ? (

                                    <video

                                    src={`/img/${service.video}`}

                                    autoPlay

                                    loop

                                    className={styles.video}

                                    />

                                ):(

                                    <Image

                                    src={`/img/${service.photo}`}

                                    width="100%"

                                    height="100%"

                                    layout="responsive"

                                    objectFit="cover"

                                    alt=""

                                    />

                                )}

                            </div>

                        </div>

.service{

    width: 500px;

    height: 500px;

    display: flex;

    align-items: center;

    justify-content: center;

    padding: 10px;

    margin-bottom: 50px;

    position: relative;

    background-color: whitesmoke;

}

.cat{

   position: absolute;

   top: 10px;

   left: 10px;

   color: gray;

   letter-spacing: 5px;

   font-weight: 300;

   text-transform: uppercase;

}

.service:hover .media{

    transform: scale(1.1);

}

.desc{

    position: absolute;

    width: 60%;

    text-align: center;

    z-index: 999;

    background-color: rgba(189, 183, 183, 0.575);

    border-radius: 10px;

    padding: 20px;

    opacity: 0;

    transition: all 1s ease;

}

.service:hover .desc{

    opacity: 1;

}

.media{

    width: 70%;

    height: 70%;

    border-radius: 50%;

    overflow: hidden;

    transition: all 1s ease;

}

.video{

    width: 100%;

    height: 100%;

    object-fit: cover;

}

Main point:

By default, text or description opacity will be 0, on hovering over service component, its opacity will become 1