Francis Road Allotments

Cost varies between £51 and £93 depending on size.

OAPs get a reduction of just under 50% and half plots are available at a reduced fee.

Landing page

Events page

News - about lockdown

Available plots (some already rotivated)

Location - Map

Contact

   // unworking tests to fix or remove

   beforeEach(() => {

      matchMedia = new MatchMediaMock();

      component = mount(

         <MemoryRouter>

            <NavBar

               routes={testRoutes}

            />

         </MemoryRouter>);

   });

   afterEach(() => {

      matchMedia.clear();

   });

   // const fireResize = (width) => {

   //    window.innerWidth = width;

   //    window.dispatchEvent(new Event("resize"));

   // };

   // test(

   //    "temp test", () => {

   //       console.error(

   //          "pre-test text: ", findByTestAttr(

   //             component, "tempTest"

   //          ).text()

   //       );

   //       fireResize(200);

   //       component.setProps({});

   //       const mediaQuery = "(min-width: 768px)";

   //       const listener = jest.fn();

   //       const mql = window.matchMedia(mediaQuery);

   //       mql.addEventListener(

   //          "change", ev => {

   //             console.error(

   //                "event: ", ev

   //             );

   //             // ev.matches && listener()

   //          }

   //       );

   //       act(() => {

   //          matchMedia.useMediaQuery(mediaQuery);

   //       });

   //       console.error(findByTestAttr(

   //          component, "tempTest"

   //       ).text());

   //       // console.error(

   //       //    mql.matches, mql.media

   //       // );

   //       // fireResize(100);

   //       // component.setProps({});

   //       // console.error(findByTestAttr(

   //       //    component, "tempTest"

   //       // ).text());

   //       // console.error(window.innerWidth);

   //    }

   // );

   // describe(

   //    "NavBar responsiveness tests", () => {

   //       it(

   //          "should show a burger icon on a mobile viewport only", () => {

   //             expect(findByTestAttr(

   //                component, "burgerIcon"

   //             ).length).toEqual(0);

   //             fireResize(700);

   //             expect(findByTestAttr(

   //                component, "burgerIcon"

   //             ).length).toEqual(1);

   //          }

   //       );

   //    }

   // );

         {/\* temp variable for temp test \*/}

         {/\* <p

            style={{display: "none"}}

            data-test="tempTest"

         >

            {mobNavHeight}

         </p> \*/}

//   Pre-caching not working (App.jsx)

import homeImg from "./images/randy-fath-ey6g0z\_fs0-unsplash.jpg";

import articlesImg from "./images/dana-devolk-n\_0wi\_oruce-unsplash.jpg";

import aboutImg from "./images/prince-abid-iy1k44aa4uq-unsplash.jpg";

import infoImg from "./images/thom-holmes-3w9aalszgo0-unsplash.jpg";

// const preload = src => new Promise((resolve, reject) => {

   //    const img = new Image();

   //    img.onload = resolve;

   //    img.onerror = reject;

   //    img.src = src;

   // });

   // const preloadAll = (srcs) => Promise.all(srcs.map(preload));

   // const preloaded = preloadAll(images);

   // const images = [

   //    homeImg,

   //    infoImg,

   //    articlesImg,

   //    aboutImg

   // ];

   const images = [

      homeImg,

      infoImg,

      articlesImg,

      aboutImg

   ];

   const components = {

      "About": <About image={images[0]} />,

      "Articles": <Articles image={images[1]}/>,

      "Home": <Home image={images[2]}/>,

      "Info": <Info image={images[3]}/>

   };

   useEffect(() => {

      cacheImages();

   }, []);

   const cacheImages = async () => {

      const promises = await images.map((src) => {

         return new Promise((resolve, reject) => {

            const img = new Image();

            img.src = src;

            img.onload = resolve();

            img.onerror = reject();

         });

      });

      await Promise.all(promises);

      setIsLoading(false);

   };

            {

               isLoading ?

                  <p style={{color: "black",

                     fontSize: "40px"}}>Loading mate</p> :

                  <Home image={images[0]} />

            }

// pre-caching Hero.jsx

// const [image, setImage] = useState(homeImg);

   // useEffect(() => {

   //    getLargeImage();

   // }, []);

   // const getLargeImage = async () => {

   //    const img = new Image();

   //    img.src = await lgHomeImg;

   //    img.onload = await setImage(img.src);

   // };