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 - NavBar

   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);

   //          }

   //       );

   //    }

   // );

// Unworking pre-cache tests - App

it("it should show a loader and hide the main content if isLoading === false", () => {

      console.error(findByTestAttr(component, "preCacheHidden"));

      // const realUseState = React.useState;

      // jest

      //    .spyOn(React, "useState")

      //    .mockImplementationOnce(() => realUseState(false))

      //    .mockImplementationOnce(() => realUseState(false));

      // component.setProps({});

      // console.error(component.debug());

      // expect(findByTestAttr(component, "loaderCont").length).toBe(0);

      // expect(findByTestAttr(component, "pageComponent")

      //    .find("Transition")

      //    .children()

      //    .length)

      //    .toBe(1);

   });

   it("should render one static page component at a time", () => {

      // console.error(screen.getAllByRole("img"));

      // expect(findByTestAttr(component, "pageComponent")

      //    .find("Transition")

      //    .children()

      //    .length)

      //    .toBe(1);

      // console.error(component

      //    .debug());

   });

it("should have a different transition time between large and small screen")

// Hero Tests

test("hero subtitle and homepage text should gain styles once main heading has finished animating");

// Unworking tests - HeroImage

   global.loadImageWithPromise = () => Promise.resolve();

const tick = () => {

      return new Promise(resolve => {

         setTimeout(resolve, 0);

      });

   };

   global.loadImageWithPromise = () => Promise.resolve();

   it("should make the opacity of the small image 0 once the large image has loaded", async () => {

      // global.Image = class {

      //    constructor() {

      //       setTimeout(() => {

      //          this.onload();

      //       }, 10);

      //    }

      // };

      const opacityStyle = findByTestAttr(

         component, "heroImage")

         .at(0)

         .props()

         .style

         .opacity;

      const opacityTestStr = findByTestAttr(

         component, "heroImage")

         .at(0)

         .props()

         .style

         .opacity

         .replace(/ /gu, "");

      const wrapper = findByTestAttr(

         component, "heroImage")

         .at(0)

         .props();

      // console.error(wrapper);

      // await tick();

      // component.update();

      // component.setProps({});

      // console.error("opacity: ", findByTestAttr(

      //    component, "heroImage")

      //    .at(0)

      //    .props()

      //    .style

      //    .opacity

      // );

      // setImmediate(() => {

      //    // within `setImmediate` all of the promises have been exhausted

      //    expect(findByTestAttr(

      //       component, "heroImage")

      //       .at(0)

      //       .props()

      //       .style

      //       .opacity).toBe("0");

      //    // have to call `done` here to let Jest know the test is done

      //    done();

      // });

   });

**Projects with tests**

https://www.reddit.com/r/reactjs/comments/iobj1a/do\_you\_know\_of\_a\_good\_sample\_react\_app\_with\_the/