**Abstract**

Whiteboards are used in classrooms, businesses and for example with myself, right at home! They are a great tool to express information onto a canvas and to encourage collaborative learning. In an online context this increase the possibilities and potential to allow for multiple people work on the same whiteboard, to view the whiteboard in remote locations and to allow the ability of extra actions like undo, redo and rescaling.

The purpose of this project was to create a whiteboard that could accessed over the internet delivering real-time drawing of lines and objects. With the extension of a permission system making it view-only, local-edit or fully interactive with everyone who is using this. It's also compatible with mobiles as well as desktops allowing people with varying technology to use the whiteboard.

The whiteboard allows users to draw lines and objects, images, text, undo, redo, create and join whiteboard sessions, set permissions, resize objects and the canvas itself all with a clean and simplistic UI.

The whiteboard uses a variety of the latest libraries and frameworks such as React, node.js, MongoDB, socket.io and Konva. These allows me to have a snappy front-end and modern backend which can handle a significant amount of users viewing a whiteboard at once.

**Main text**

Whiteboards can be seen in all contexts, in a educations, business and even personal scenarios. They can often be seen as tools to procide information withas seen by presenters or also as learning tools to allow group collaborations. With an extension, we have seen interactive whiteboards in the class using software to be displayed on and draw onto with a project and whiteboard or embedded within a presentation itself.

Personally I own a wall-mounted whiteboard myself which I frequently use to scribble out information that comes into my head to help me problem solve. Whilst this has been incredibly helpful, a huge extension onto this, in an online context, is the ability to save and return back to a whiteboard later on. Personally, im also an interactive learner. Being presented by a slideshow on a unused whiteboard feels like such a wasted opportunity, and with the potential of it being able to extended onto student’s electronic devices says to me there is a huge gap for improvement.

Pre development involved researching what tools would be the best. I knew I wanted a modern and interactive frontend so I explored the big 3 frontend frameworks, React, Vue.js and Angular. I decided to go with React as I had some experience with it before and I knew that it is incredibly popular with very good documentation.

Next I decided to choose the back-end server. My 3 main options were PHP, node.js and go. In comparison between PHP and node.js, node.js comes out the clear winner. Node.js is faster than PHP and in my opinion is much cleaner. Comparing node.js to go I wasn’t too comfortable with the documentation for go so I decided with node.js.

I then had to choose which database to use. I have an array of options being MySQL, Orcal,e Microsoft SQL Server, PostgreSQL, MongoDB, MariaDB. Here was my comparison of the databases.

|  |  |  |
| --- | --- | --- |
| **Database management software** | **Pros** | **Cons** |
| MySQL | Very standard, lots of documentation. Have experience with this before. | Not particularly suitable for my purpose not for a dynamic website. |
| Orcale | Nothing, it’s awful! | It’s slow & costs money. |
| Microsoft SQL Server | Developed by Microsoft, fast and stable. | Suitable for a business environment, not for a dynamic website like the whiteboard. Also costs money! |
| PostgreSQL | Incredibly scalable for if there was a surge on hits to the website | Limited documentation |
| MongoDB | Incredibly versatile, fast | Bit of hassle to setup, doesn’t use SQL so little bit of a learning curve |
| MariaDB | Fast & stable | Quite a new SQL type, but a better version of MySQL essentially. |

My final choice of software to use pre-development was deciding how to communicate clients to the server. I choose socket.io.

Development started with the creation of a canvas. I started my approach with using the basic HTML canvas element. The canvas element, as implied by the name, provides a blank canvas to which I can call functions on to draw shapes.

I implemented X colors. I progressed to implement more features, such as networking, undo, clearing the whiteboard.