This is a heading

and this is the note that I am making

```
window.onblur = () => {
    closeCtxMenu();
};



document.onmousedown = () => {
    closeCtxMenu();
};



contextMenu.onmousedown = (e) => {
    e.preventDefault();
};



contextMenu.oncontextmenu = (e) => {
    e.preventDefault();
};


lorem ipsum this is a really long line that will reach the end so that I can check out what' wrongt


async function loadNote(self) {
    if (window.editor) {
        window.editor.updateOptions({ readOnly: true });
        let toastEl = document.querySelector(".toast[data-toast-type='ongoing']");
        toastEl
            .querySelector(".toast-body")
            .textContent = "Opening Note";
        let toast = window.Bootstrap.Toast.getInstance(toastEl);
        toast.show();

        let noteToken = self.dataset.token;
        const token = document.head.querySelector('meta[name="csrf-token"]').getAttribute("content");
        let response = await fetch("{{ route('note.get') }}", {
            method: "POST",
            headers: {
                "Content-Type": "Application/json",
                "X-CSRF-TOKEN": token
            },
            body: JSON.stringify({
                "note_token": noteToken
            })
        });

        if (response.status === 404) {
            console.log("Note not found");
            return;
        }

        if (response.status === 200) {
            let result = await response.json();

            if (result.note_token !== noteToken) {
                console.log("An error occurred!\nNote Tokens do not match. Please refresh page and try again");

                // Show an error toast to user
                let toastEl = document.querySelector(".toast[data-toast-type='static']");
                toastEl
                    .querySelector(".toast-body")
                    .textContent = "Please refresh your page and try again (Protip: you can press CTRL + R to refresh your page)";
                toastEl.classList.add("bg-danger");
```

```
                let toast = window.Bootstrap.Toast.getInstance(toastEl);
                toast.show();
                return;
            }

            if (window.editor.note_token) {
                document.querySelector(`.notes-list-item[data-token="${window.editor.note_token}"]`).classList.remove("active");
            }
            self.classList.add("active");
            window.editor.setValue(result.body === null ? "" : result.body);
            window.editor.note_token = noteToken;
            window.editor.updateOptions({ readOnly: false });
            toast.hide();
        }
    }
}



function rightClick(self, event) {
    event.preventDefault();

    let topOffset = 0;

    if (contextMenu.offsetHeight + event.clientY > window.innerHeight) {
        topOffset = event.clientY - contextMenu.offsetHeight;
    } else {
        topOffset = event.clientY;
    }

    contextMenu.style.top = topOffset + "px";
    contextMenu.style.left = event.clientX + "px";
    contextMenu.classList.add("active");
}



function closeCtxMenu() {
    contextMenu.classList.remove("active");
}
```

And now we see something