```jsx
import { forwardRef, useEffect, useState } from 'react';
import { useNavigate } from 'react-router-dom';
import PropTypes from 'prop-types';
// @mui
import {
    Slide,
    Dialog,
    Stack,
    Typography,
    Container
} from '@mui/material';
// constants
import { MATCH_REQUEST, MATCH_FOUND, MATCH_NOT_FOUND,
ALL_USERS_MATCHING_CHANNEL } from '../../constants/socket-
events';
// components
import { CustomAvatar } from "../custom-avatar";
import { CustomHeader } from "../custom-header";
import NoMatchDialog from './NoMatchDialog';
// sockets
// import { socketService } from '../../services/socket-
service';
// routes
import { PATH_APP } from '../../routes/paths';
// config
import { HEADER } from '../../config-global';
import { useChannel } from '../../hooks/useChannel';
// hooks

//
------------------------------------------------------------------
------

const Transition = forwardRef((props, ref) => <Slide
direction="left" ref={ref} {...props} />);

FindingMatchDialog.propTypes = {
    isOpen: PropTypes.bool,
    onClose: PropTypes.func,
    imgSrc: PropTypes.string,
    user: PropTypes.object,
    question: PropTypes.object,
};

export default function FindingMatchDialog({ isOpen, onClose,
imgSrc, user, question }) {
```

```
    const navigate = useNavigate();

    const [noMatchDialogVisible, setNoMatchDialogVisible] =
useState(false);

    const [allUsersMatchChannel, allUsersMatchChannelAbly] =
useChannel({
        channelName: ALL_USERS_MATCHING_CHANNEL,
    });

    const [userChannel, userChannelAbly] = useChannel({
        channelName: `match-finding-user-${user.user_id}`,
        eventNames: [MATCH_FOUND, MATCH_NOT_FOUND],
        callbackOnMessageMap: {
            [MATCH_FOUND]: (message, _channel) => {
                const data = message.data;


navigate(PATH_APP.conversations.view(data.match_id));
            },
            [MATCH_NOT_FOUND]: (message, _channel) => {
                console.log("listening to Match not found")
                openNoMatchDialog();
            }
        },
    });

    useEffect(() => {
        const matchRequestTimer = setTimeout(() => {
            console.log("Sending out match request!!")
            allUsersMatchChannel.publish(MATCH_REQUEST, {
                question_id: question.question_id,
                user_id: user.user_id,
            });
        }, 3000);

        return () => {
            clearTimeout(matchRequestTimer);
        }
    }, [])

    const openNoMatchDialog = () =>
setNoMatchDialogVisible(true);

    return (
        <Dialog
```

```
            open={isOpen}
            fullScreen={true}
            TransitionComponent={Transition}
            data-qa-id="finding-match-dialog"
        >
            <CustomHeader
                onClose={onClose}
            />
            <Container sx={{ height: "inherit", pt: `${HEADER.H_MOBILE}px` }} data-qa-id={"finding-match-container"}>
                <Stack direction="column" justifyContent="flex-start" alignItems="center">
                    <Stack
                        direction="column"
                        justifyContent="center"
                        alignItems="center"
                        spacing={1}
                        sx={{
                            color: "common.white",
                            textAlign: "center",
                        }}
                    >
                        <CustomAvatar
                            src={imgSrc}
                            name={user?.first_name}
                            sx={{
                                mx: 'auto',
                                width: { xs: 80, md: 128 },
                                height: { xs: 80, md: 128 },
                            }}
                        />
                        <Typography variant="h4" sx={{ color: 'text.secondary' }}>Finding friend...</Typography>
                    </Stack>
                </Stack>
                <NoMatchDialog
                    open={noMatchDialogVisible}
                />
            </Container>
        </Dialog>
    );
}
```