

```

import React, { useState } from "react";
import { Box, Button, TextField, Typography } from "@mui/material";
import { ref, get } from "firebase/database";
import initializeFirebase from "../data/firebase/firebase";
import { Navigate } from "react-router-dom";

const LoginPage = ({ onLogin }) => {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const [error, setError] = useState("");
  const [isLoggedIn, setIsLoggedIn] = useState(false);
  const lastVisitedRoute = localStorage.getItem("lastVisitedRoute");

  const handleLogin = async () => {
    try {
      const database = initializeFirebase();
      const paramPath = "/Admin";
      const paramRef = ref(database, paramPath);
      const snapshot = await get(paramRef);

      if (snapshot.exists()) {
        const userData = snapshot.val();
        if (userData.username === username && userData.password === password) {
          setIsLoggedIn(true);
          onLogin(); // Removed rememberMe argument since it's not needed anymore
        } else {
          setError("Incorrect username or password");
        }
      } else {
        setError("User not found");
      }
    } catch (error) {
      console.error("Error during login:", error.message);
      setError("An unexpected error occurred");
    }
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    handleLogin();
  };

  if (isLoggedIn) {
    return <Navigate to={lastVisitedRoute || "/dashboard"} replace />;
  }
}

```

```

return (
  <Box
    display="flex"
    flexDirection="column"
    alignItems="center"
    justifyContent="center"
    minHeight="100vh"
    bgcolor="#f0f2f5"
  >
    <Box
      sx={{
        display: "flex",
        flexDirection: "column",
        alignItems: "center",
        width: 300,
        p: 4,
        backgroundColor: "white",
        borderRadius: 8,
        boxShadow: "0px 3px 15px rgba(0,0,0,0.2)",
      }}
    >
      <Box mb={2}>
        
      </Box>
      <Typography variant="h5" align="center" gutterBottom>
        Login
      </Typography>
      <form onSubmit={handleSubmit}>
        <TextField
          label="Username"
          variant="outlined"
          fullWidth
          margin="normal"
          value={username}
          onChange={(e) => setUsername(e.target.value)}
        />
        <TextField
          label="Password"
          variant="outlined"
          fullWidth
          type="password"
          margin="normal"

```

```

        value={password}
        onChange={(e) => setPassword(e.target.value)}
      />
      <Button
        type="submit"
        variant="contained"
        color="primary"
        fullWidth
        sx={{ mt: 2 }}
      >
        Login
      </Button>
    </form>
    {error && (
      <Typography variant="body1" color="error" align="center" mt={2}>
        {error}
      </Typography>
    )}
  </Box>
</Box>
);
};

```

```
export default LoginPage;
```