StudentLoginRegister/Login

module.exports() => (HTML)

This function, displays the login section, and handles the logic behind the login.

Kind: Exported function

Code

```
/** @module StudentLoginRegister/Login */

import "./style.css"
import {useState} from "react"
import {useNavigate} from "react-router-dom"

/**
* This function, displays the login section, and handles the logic behind the
login.
* @function
* @return {HTML}
*/
export default function Login(){
    /**
    * "email" stores the users email, and "set_email" is a function, which is used
to change the value of "email".
    * @constant
    * @typedef {Array.<String, Function>}
    */
    const [email, set_email] = useState("")
    /**
    * "password" stores the users email, and "set_password" is a function, which
is used to change the value of "password".
    * @constant
    * @typedef {Array.<String, Function>}
    */
    const [password, set_password] = useState("")

    let navigate = useNavigate()

    /**
    * This function sends a login request to the backend.
    * @function
    * @param {Object} user_details
    * @returns {null}
    */
    function _Login(e, user_details){
        e.preventDefault() /* Added this line, to stop page from refreshing,
```

```
    before a response is recieved from the backend. */
        fetch("http://localhost:8000/login/student",{
            method: "POST",
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify(user_details)
        })
        .then(response => response.json())
        .then(user => {
            if (user["exists"] === true){
                console.log("success")
                localStorage.setItem("logged_in", true)
                localStorage.setItem("student", true)
                localStorage.setItem("id", user["user_id"])
                localStorage.setItem("name", user["name"]) /* This is added, to
enable the use of custom usernames. */

                navigate("/user")
                window.location.reload() /* Addition of this line fixes the issue
of the navbar not re-rendering, when user logs in. */
            }
            else{
                localStorage.setItem("logged_in", false)
            }
        })
    }

    return(
        <form className="login" onSubmit={(e) => _Login(e, {"email": email,
"password": password})}>
            <label>
                Email
                <input className="email" type="email" value={email} onChange={(e)
=> set_email(e.target.value)} />
            </label>
            <label>
                Password
                <input className="password" type="password" value={password}
onChange={(e) => set_password(e.target.value)} />
            </label>
            <input type="submit" value="Login" />
        </form>
    )
}
```