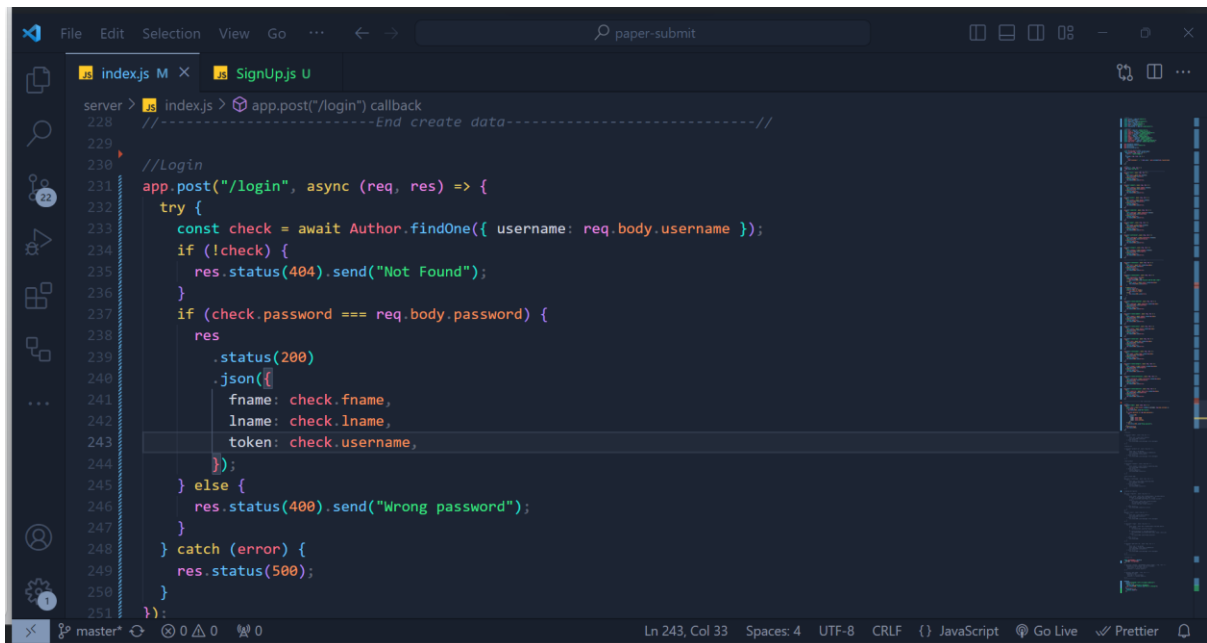
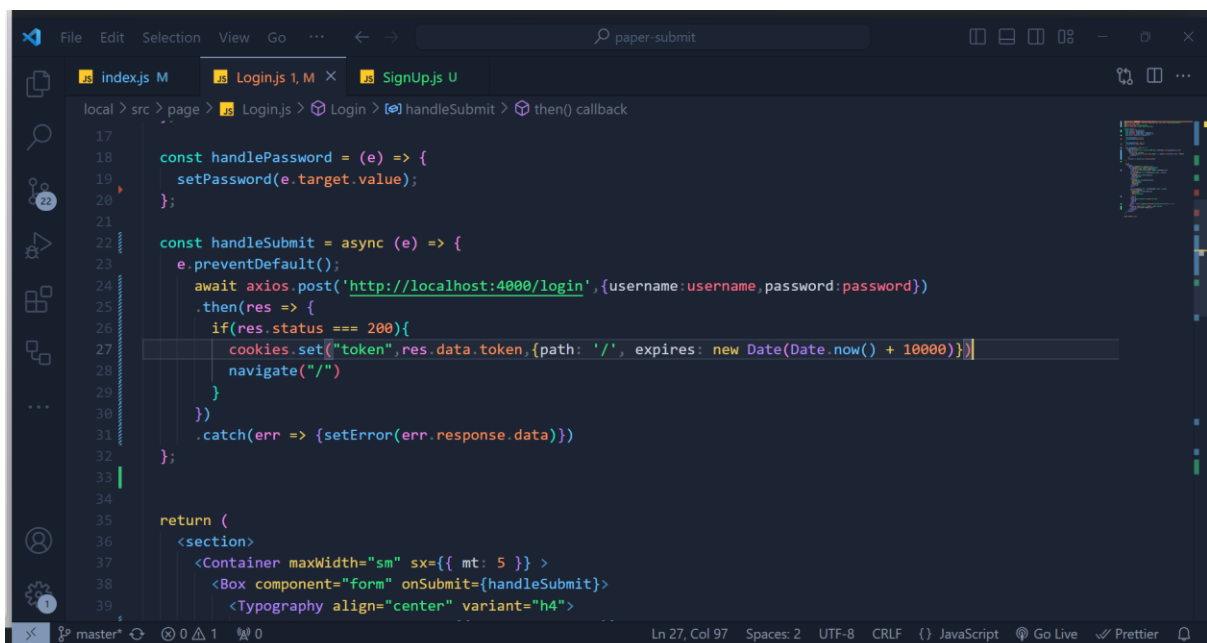


รวมอัปเดต วันที่ 8.12.66 – 13.12.66



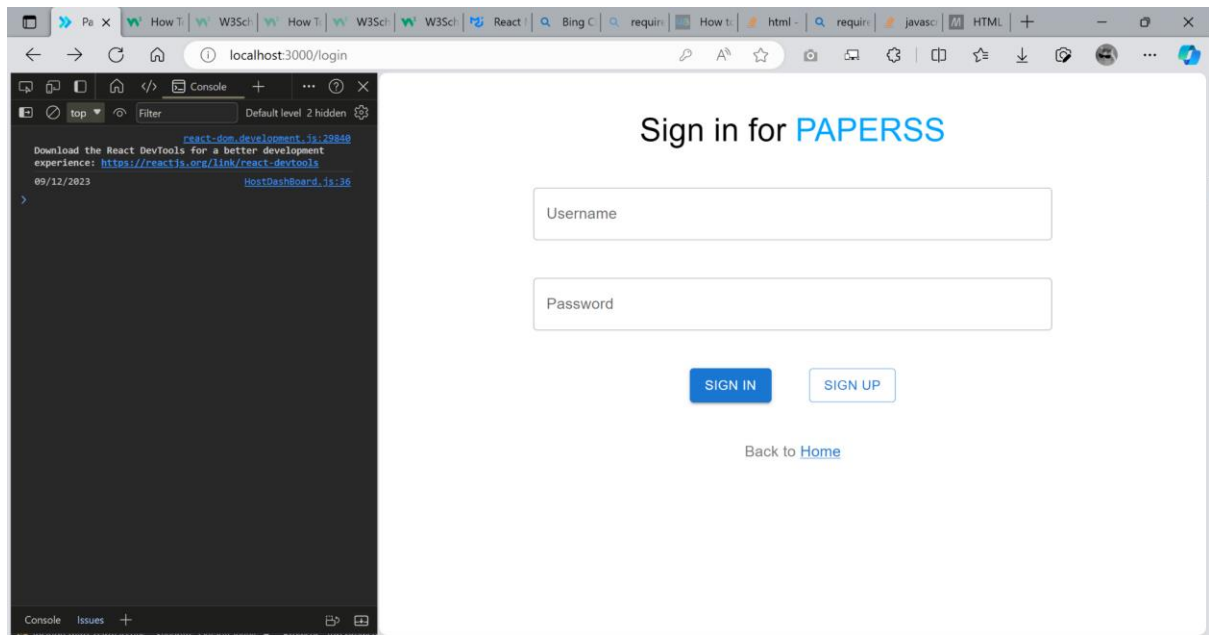
```
server > index.js > app.post("/login") callback
//-----End create data-----//
228
229
230 //Login
231 app.post("/login", async (req, res) => {
232   try {
233     const check = await Author.findOne({ username: req.body.username });
234     if (!check) {
235       res.status(404).send("Not Found");
236     }
237     if (check.password === req.body.password) {
238       res
239         .status(200)
240         .json({
241           fname: check.fname,
242           lname: check.lname,
243           token: check.username,
244         });
245     } else {
246       res.status(400).send("Wrong password");
247     }
248   } catch (error) {
249     res.status(500);
250   }
251 });
```

เพิ่มส่วนของ token เข้าไปใน api เพื่อจะใช้ในการ set cookie



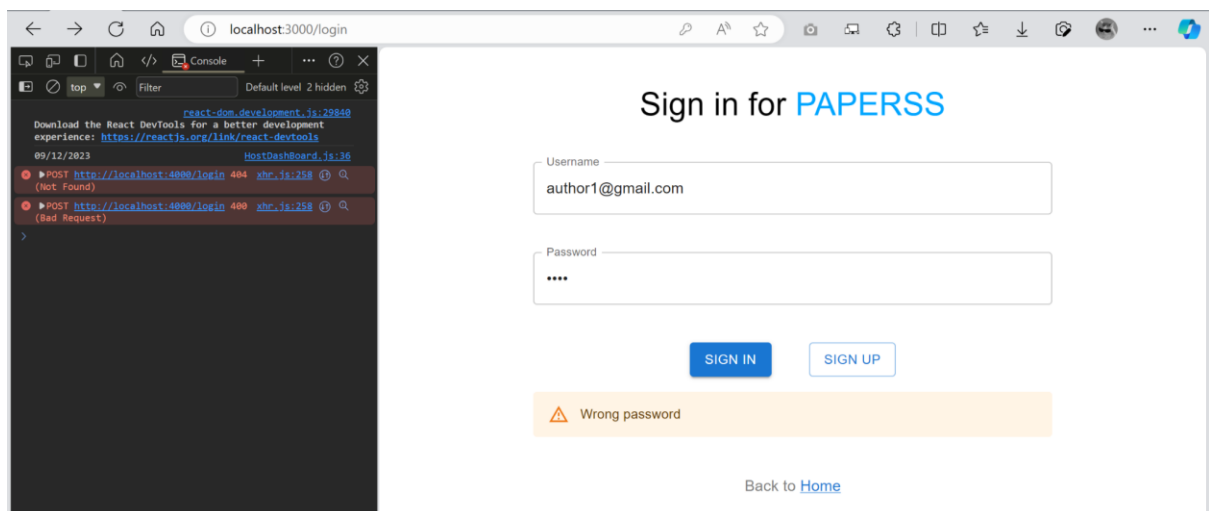
```
local > src > page > Login.js > Login > handleSubmit > then() callback
17
18 const handlePassword = (e) => {
19   setPassword(e.target.value);
20 };
21
22 const handleSubmit = async (e) => {
23   e.preventDefault();
24   await axios.post('http://localhost:4000/login', {username: username, password: password})
25     .then(res => {
26       if(res.status === 200){
27         cookies.set("token", res.data.token, {path: '/', expires: new Date(Date.now() + 10000)});
28         navigate("/")
29       }
30     })
31     .catch(err => {setError(err.response.data)})
32 };
33
34
35 return (
36   <section>
37     <Container maxWidth="sm" sx={{ mt: 5 }} >
38       <Box component="form" onSubmit={handleSubmit}>
39         <Typography align="center" variant="h4">
```

มีการติดตั้ง axios และ universal-cookie เพื่อที่จะใช้สำหรับการเรียก api และนำเอาค่าจาก api มาเป็น set cookie ไว้ที่เครื่องของ user และกำหนดให้หมดเวลาไว้ 10000 ms (เป็นการ test จะมีการเปลี่ยนเป็นให้หมดอายุ หลังจากวันที่ login 1 วัน)

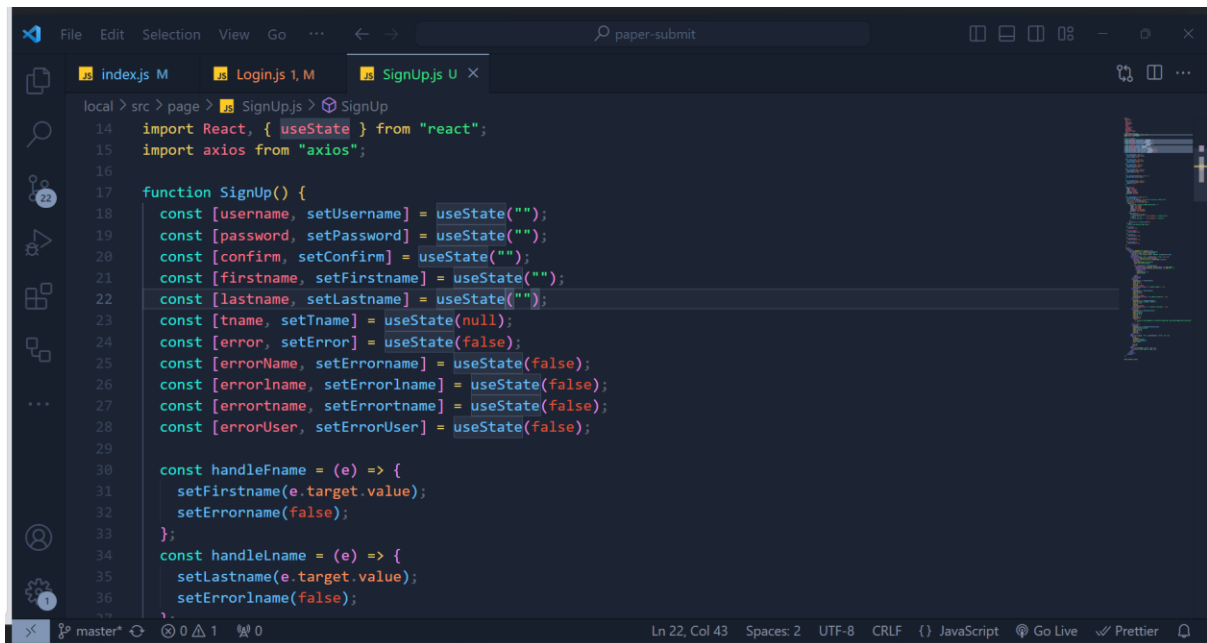


เมื่อ Login Success จะพา user ไปที่หน้า Home

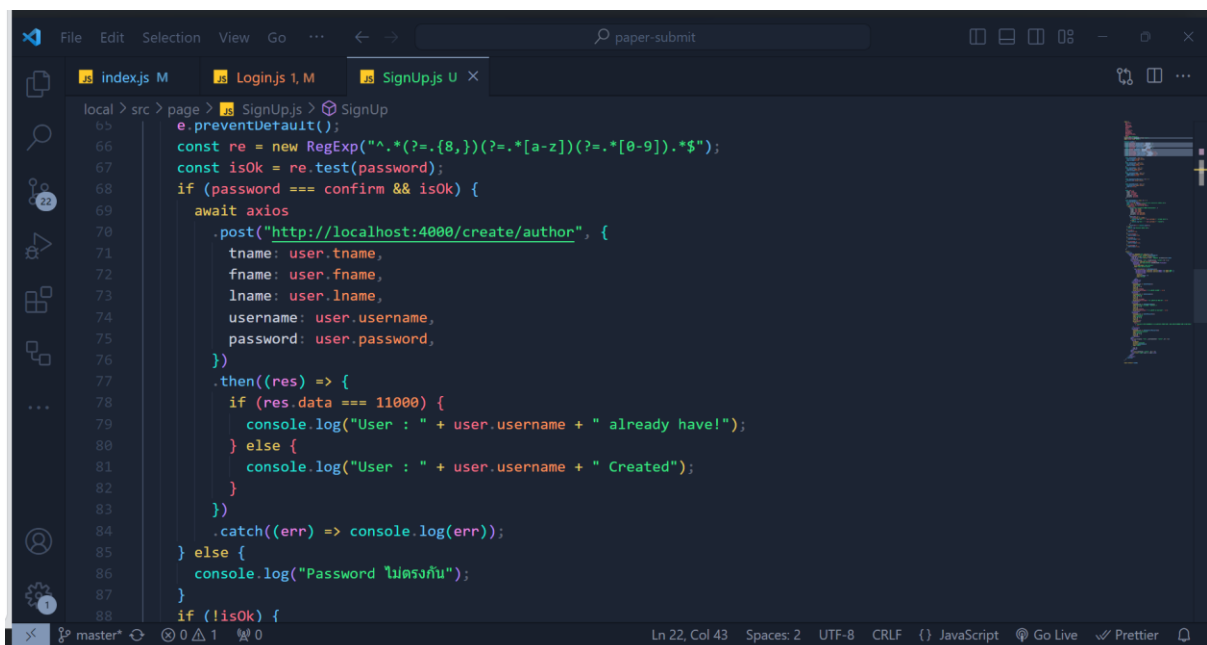
แต่ถ้าผิดจะมีการ แสดง alert text ออกมาเป็นตามหัวข้อที่ผิด



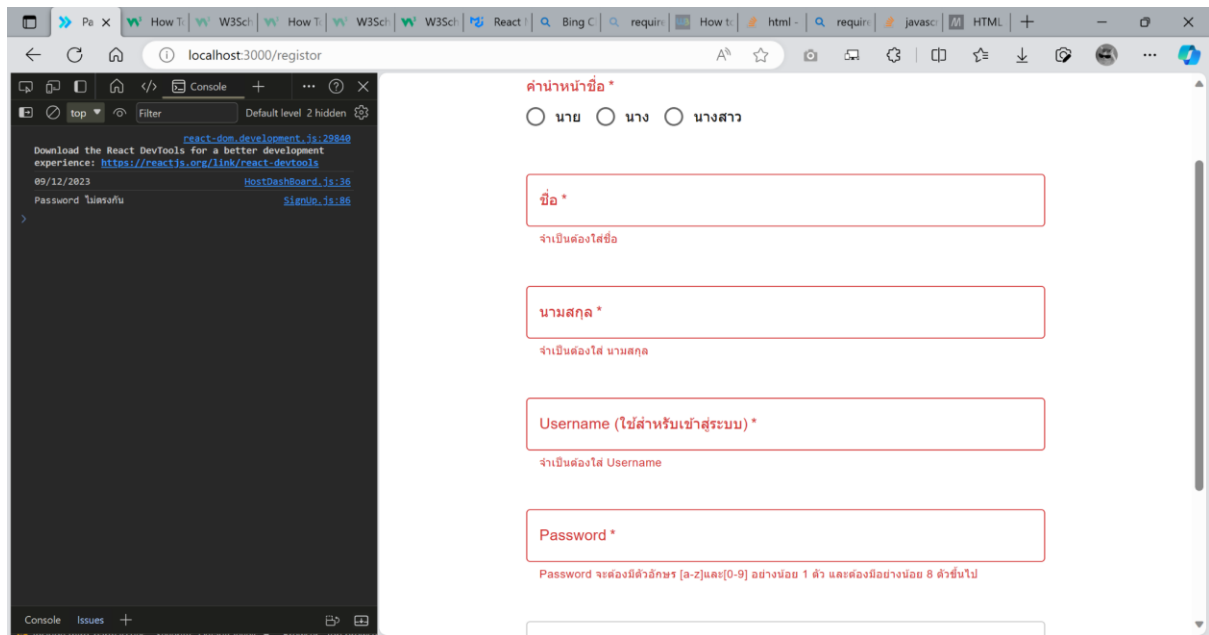
มีการเพิ่มการทำงานของ register ของ author ซึ่งมีปัญหาเกี่ยวกับการแยกว่าเมื่อ register แล้วจะแยกระหว่าง author กับ host ยังไง



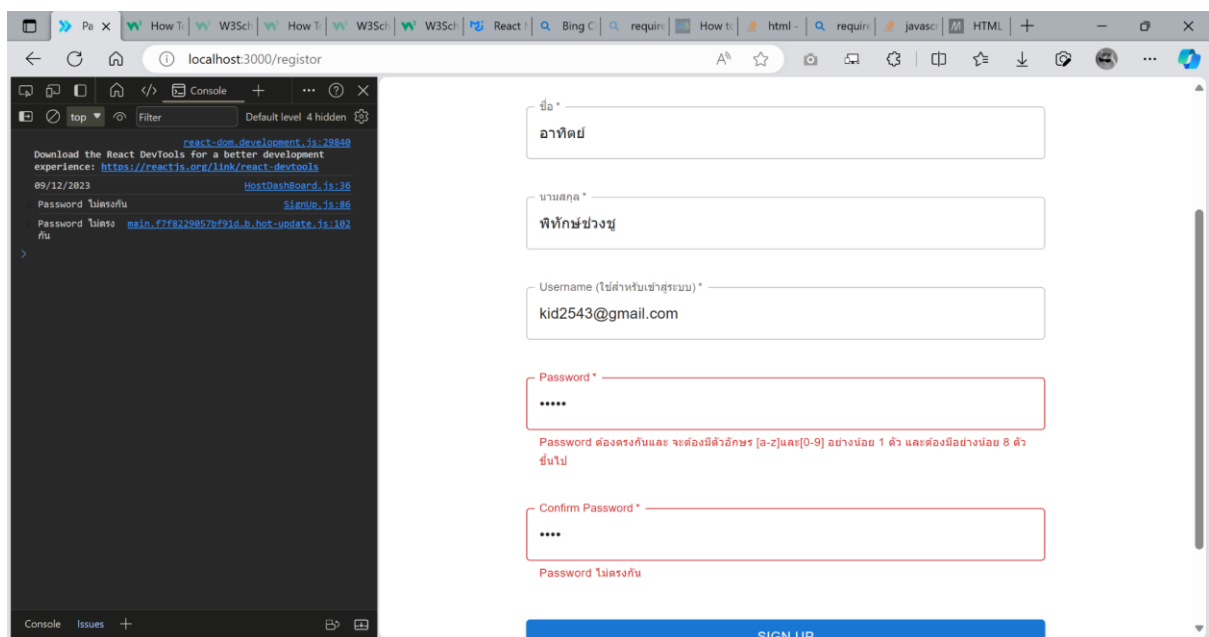
```
14 import React, { useState } from "react";
15 import axios from "axios";
16
17 function SignUp() {
18   const [username, setUsername] = useState("");
19   const [password, setPassword] = useState("");
20   const [confirm, setConfirm] = useState("");
21   const [firstname, setFirstname] = useState("");
22   const [lastname, setLastname] = useState("");
23   const [tname, setTname] = useState(null);
24   const [error, setError] = useState(false);
25   const [errorName, setErrorname] = useState(false);
26   const [errorIname, setErrorIname] = useState(false);
27   const [errorortname, setErrorortname] = useState(false);
28   const [errorUser, setErrorUser] = useState(false);
29
30   const handleFname = (e) => {
31     setFirstname(e.target.value);
32     setErrorname(false);
33   };
34   const handleLname = (e) => {
35     setLastname(e.target.value);
36     setErrorIname(false);
37   };
38 }
```



```
65 e.preventDefault();
66 const re = new RegExp("^(?=.*{8,})(?=.*[a-z])(?=.*[0-9]).*$");
67 const isOk = re.test(password);
68 if (password === confirm && isOk) {
69   await axios
70     .post("http://localhost:4000/create/author", {
71       tname: user.tname,
72       fname: user.fname,
73       lname: user.lname,
74       username: user.username,
75       password: user.password,
76     })
77     .then((res) => {
78       if (res.data === 11000) {
79         console.log("User : " + user.username + " already have!");
80       } else {
81         console.log("User : " + user.username + " Created");
82       }
83     })
84     .catch((err) => console.log(err));
85   } else {
86     console.log("Password ไม่ตรงกัน");
87   }
88   if (!isOk) {
```

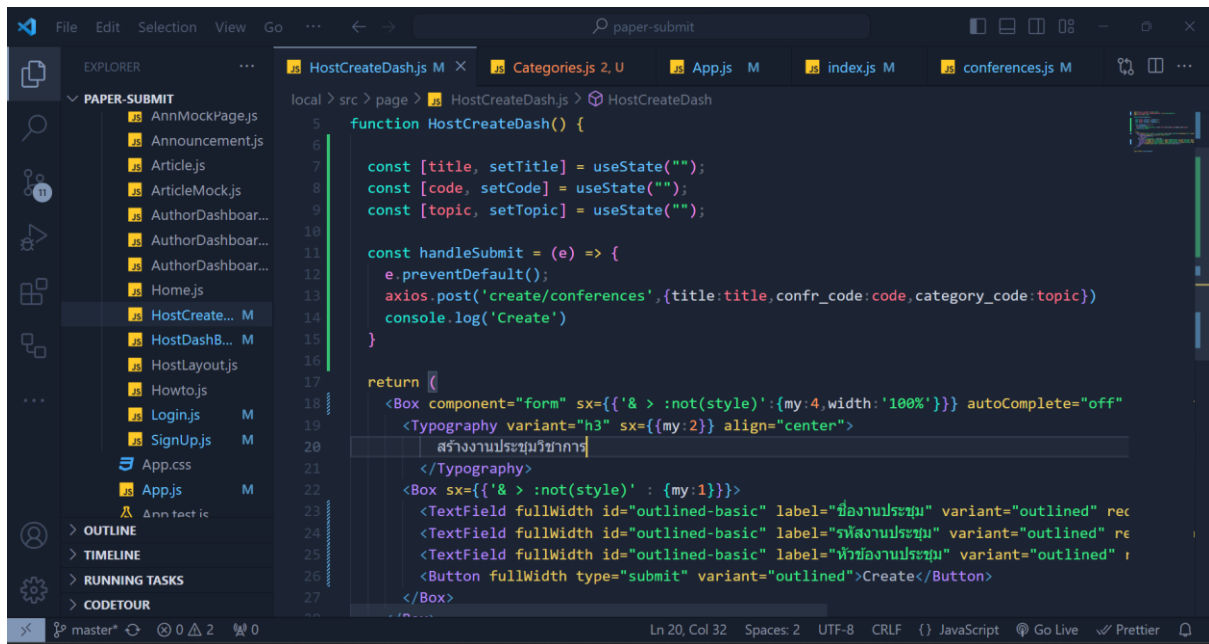


เมื่อไม่ได้กรอกข้อมูลจะทำการ ฟ้อง Error ตาม input นั้นๆ



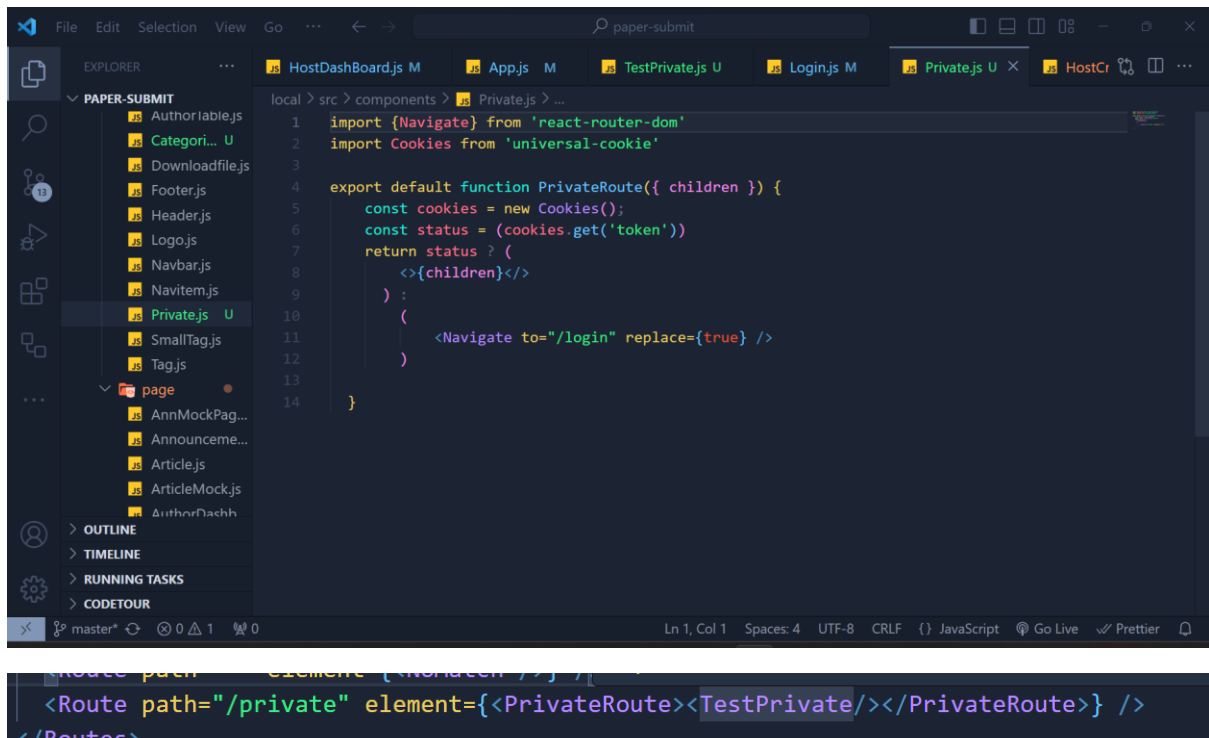
เมื่อทำการใช้ username เข้ากับ user ที่มีอยู่แล้วจะทำการ console.log แสดงว่า user นี้ถูกใช้ไปแล้ว

ทำการ ดึง api โดยใช้ axios แล้วส่งค่ากลับไปเพื่อสร้าง Page



```
function HostCreateDash() {  
  const [title, setTitle] = useState("");  
  const [code, setCode] = useState("");  
  const [topic, setTopic] = useState("");  
  
  const handleSubmit = (e) => {  
    e.preventDefault();  
    axios.post('create/conferences', {title: title, confr_code: code, category_code: topic})  
    console.log('Create')  
  }  
  
  return (  
    <Box component="form" sx={{'& >:not(style)': {my: 4, width: '100%'}}} autoComplete="off"  
      <Typography variant="h3" sx={{my: 2}} align="center">  
        สร้างงานประชุมวิชาการ  
      </Typography>  
      <Box sx={{'& >:not(style)': {my: 1}}>  
        <TextField fullWidth id="outlined-basic" label="ชื่องานประชุม" variant="outlined" rec  
        <TextField fullWidth id="outlined-basic" label="รหัสงานประชุม" variant="outlined" re  
        <TextField fullWidth id="outlined-basic" label="หัวข้องานประชุม" variant="outlined" r  
        <Button fullWidth type="submit" variant="outlined">Create</Button>  
      </Box>  
    )  
  )  
}
```

สร้าง Private Route ขึ้นเพื่อจำกัดการเข้าถึงของ User และนำเอา Cookie มาใช้เพื่อทำการแสดงข้อมูลเฉพาะของ User คนนั้นๆ



```
import {Navigate} from 'react-router-dom'  
import Cookies from 'universal-cookie'  
  
export default function PrivateRoute({ children }) {  
  const cookies = new Cookies();  
  const status = (cookies.get('token'))  
  return status ? (  
    <{children}</>  
  ) : (  
    <Navigate to="/login" replace={true} />  
  )  
}
```

```
<Route path="/private" element={<PrivateRoute><TestPrivate/></PrivateRoute>} />  
</Routes>
```

หลังจากนั้นก็เอามาครอบส่วนที่ต้องการจะจำกัด

เพิ่ม หน้าสำหรับ Update การประกาศบทความ

→ TOOLS

ข้อมูลจำเป็น

ชื่องานประชุม

รหัสงานประชุม

หัวข้อที่เกี่ยวข้อง

ข้อมูลหน้าประกาศ

ข้อมูลทั่วไป

คำอธิบายงานประชุม

Icon ของงานประชุม

ใบปลิว

วันสำคัญ

เพิ่ม Api เกี่ยวกับการ Update

```
server > index.js > app.put("/conferences/user/:id") callback
256 } else {
257   res.status(400).send("Wrong password");
258 }
259 } catch (error) {
260   res.status(500);
261 }
262 });
263
264 // -----update-----
265
266 app.put("/conferences/user/:id", async (req, res) => {
267   try {
268     const confrId = req.params.id
269     const userInput = req.body;
270     const confUpdate = await Conferences.findByIdAndUpdate(confrId, {$set: {title: userInput.title}})
271     res.status(200).json(confUpdate)
272   } catch (error) {
273     console.log(error)
274   }
275 })
276
277 //Find ALL
278
```