

กรชนก เขียวดนตรี 6210110446

Github url : <https://github.com/HowlKornchanok/DCW-Final-Assignment>

## รายงาน DCW Final Assignment

### Backend

#### Logger

ใช้ packages morgan และ winston ร่วมกับ express ซึ่งทำการ catching HTTP method ที่ request เข้ามา ในการสร้าง log file โดยใช้แพ็คเกจ morgan ในการสร้าง log file ที่เก็บข้อมูลการ request ที่มายัง server และ respond ที่ server ส่งออก ส่วนการใช้ Winston ใช้ในการสร้าง log file ที่เก็บข้อมูลการทำงานของ server

#### Authentication

ใช้ package passport ( passport, passport-google-oauth20, passport-github2, passport-facebook) ในการเก็บข้อมูล body client id และ client secret ในการใช้ OAuth โดยทำงานร่วมกับแพ็คเกจ express ในการ catching get /loginstatus และ respond สถานะไปยัง console และ frontend ตามสถานการณ์โดยใช้ cors เป็น middleware เพื่อให้ในการส่งข้อมูล use state ของ user ไปแสดงใน frontend และใช้แพ็คเกจ cookie-session ในการสร้าง session เก็บไว้เพื่อไม่ต้องทำการ login ใหม่ทุกครั้ง

```
passport.use(  
  new GoogleStrategy(  
    {  
      clientID: GOOGLE_CLIENT_ID,  
      clientSecret: GOOGLE_CLIENT_SECRET,  
      callbackURL: "/auth/google/callback",  
    },  
    function (accessToken, refreshToken, profile, done) {  
      done(null, profile);  
    }  
  )  
);
```

```
router.get("/google", passport.authenticate("google", { scope: ["profile"] }));

router.get(
  "/google/callback",
  passport.authenticate("google", {
    successRedirect: CLIENT_URL,
    failureRedirect: "/login/failed",
  })
);
```

```
router.get("/login/success", (req, res) => {
  if (req.user) {
    res.status(200).json({
      success: true,
      message: "successfull",
      user: req.user,
    });
  }
});
```

```
router.get("/logout", (req, res) => {
  req.logout();
  res.redirect(CLIENT_URL);
});
```

## TEST API

ใช้ express ร่วมกับ bodyparser และ cors เพื่อทดสอบการทำงานของ server โดยใช้ HTTP method ในทำการทดสอบโดยดึงข้อมูลที่แสดงมาจาก body ของ user.json ที่สร้างไว้เพื่อทดสอบ

```
app.get('/user', function (req, res, next) {
  return res.status(200).json({
    code: 1,
    message: 'OK',
    data: users
  })
});
```

กรชนก เขียตุนสุรณ 6210110446

Github url : <https://github.com/HowlKornchanok/DCW-Final-Assignment>

## Frontend

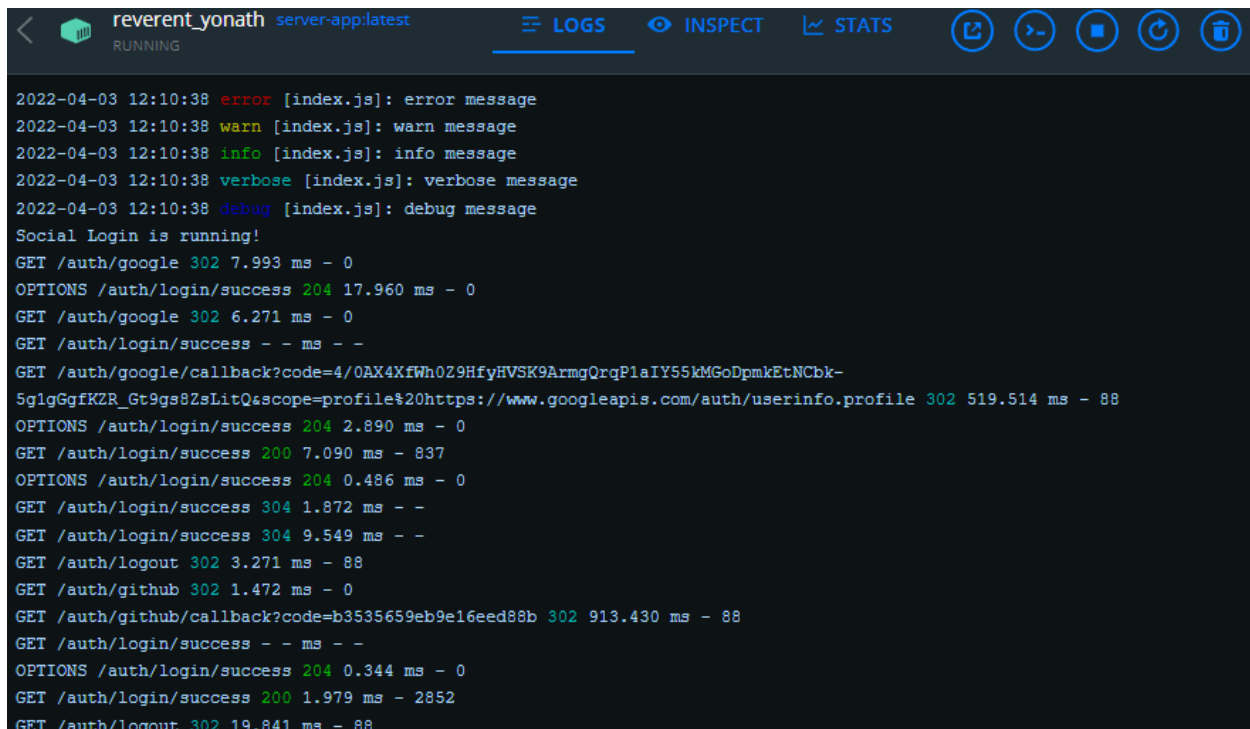
ใช้ react router-dom เพื่อเข้าถึง DOM ให้ทำการ render ในหน้าของแอปพลิเคชัน โดยแอปพลิเคชันจะทำงานร่วมกับ Authentication Application โดยดึงข้อมูล use state เป็นรูปและชื่อของ user ด้วย HTTP method GET มาจาก server เมื่อ login/success และให้อำนาจการเข้าถึง(Access Control Allow Credential) Post.jsx

```
useEffect(() => {  
  const getUser = () => {  
    fetch("http://localhost:5000/auth/login/success", {  
      method: "GET",  
      credentials: "include",  
      headers: {  
        Accept: "application/json",  
        "Content-Type": "application/json",  
        "Access-Control-Allow-Credentials": true,  

```

```
<Route  
  path="/post/:id"  
  element={user ? <Post /> : <Navigate to="/login" />}  
/>
```

## Docker Application



The screenshot shows the Docker Desktop interface with the 'LOGS' tab selected for a container named 'reverent\_yonath'. The logs display a series of HTTP requests and responses, including GET requests to /auth/google, OPTIONS requests to /auth/login/success, and GET requests to /auth/login/success and /auth/logout. The logs also show a 'Social Login is running!' message and a 'debug' message. The container is running on a server with the latest version of the application.

```
< reverent_yonath server-app:latest  
RUNNING  
LOGS INSPECT STATS  
2022-04-03 12:10:38 error [index.js]: error message  
2022-04-03 12:10:38 warn [index.js]: warn message  
2022-04-03 12:10:38 info [index.js]: info message  
2022-04-03 12:10:38 verbose [index.js]: verbose message  
2022-04-03 12:10:38 debug [index.js]: debug message  
Social Login is running!  
GET /auth/google 302 7.993 ms - 0  
OPTIONS /auth/login/success 204 17.960 ms - 0  
GET /auth/google 302 6.271 ms - 0  
GET /auth/login/success - - ms - -  
GET /auth/google/callback?code=4/0AX4XfWh0Z9HfyHVS9Arm9Qr9P1aIY55kMGoDpmkEtNCbk-  
5g1gGgfk2R_Gt9gs8ZsLitQ&scope=profile%20https://www.googleapis.com/auth/userinfo.profile 302 519.514 ms - 88  
OPTIONS /auth/login/success 204 2.890 ms - 0  
GET /auth/login/success 200 7.090 ms - 837  
OPTIONS /auth/login/success 204 0.486 ms - 0  
GET /auth/login/success 304 1.872 ms - -  
GET /auth/login/success 304 9.549 ms - -  
GET /auth/logout 302 3.271 ms - 88  
GET /auth/github 302 1.472 ms - 0  
GET /auth/github/callback?code=b3535659eb9e16eed88b 302 913.430 ms - 88  
GET /auth/login/success - - ms - -  
OPTIONS /auth/login/success 204 0.344 ms - 0  
GET /auth/login/success 200 1.979 ms - 2852  
GET /auth/logout 302 19.841 ms - 88
```