

컴퓨터네트워크 11주차 과제

- Oauth2.0 -

| | |
|------|-----------|
| 제출일자 | 21.11.24 |
| 분 반 | 02 |
| 이 름 | 배시현 |
| 학 번 | 201702022 |

1)과제목표

구글 API인 OAuth2.0을 이용하여서 Node.js로 만든 웹에 로그인환경을 구축한다. 구축 후 자신의 구글아이디를 이용하여 자신이 만든 웹에 로그인하여 화면에 사용자이름을 표시한다.

2)코드설명

이번 시간에 만들어야할 코드는 총 4가지이다. 제출파일 순서대로 설명하겠다.

Index.js

```
var express = require('express');
var app = express();
var passport = require('passport');
var session = require('express-session');
var https = require('https');
var fs = require('fs');

var options = {
  key: fs.readFileSync('key.pem'),
  cert: fs.readFileSync('cert.pem')
};

app.set('view engine', 'ejs');
app.use(session({secret: 'MySecret', resave: false, saveUninitialized: true}));

//passport setting
app.use(passport.initialize());
app.use(passport.session());

// Routes
app.use('/', require('./routes/main'));
app.use('/auth', require('./routes/auth'));

// Port setting
var port = 3000;
https.createServer(options, app).listen(port, () => {
  console.log('server on! https://localhost:'+port);
});
```

기본적인 코드는 실습자료에 주어져있고 여기서 본인이 구현해야 할 것은 http가 아닌 https로 접속하도록 port setting을 해주어야한다. 그러기 위해서 fs와 https모듈을 가져와서 fs모듈로 발급 받은 개인키와 인증서를 불러와서 옵션이란 변수에 넣은 후, https모듈의 createServer를 이용하여 파라미터로 생성한 options를 넣고 서버를 생성한다.

Passport.js

```
var passport = require('passport');
var GoogleStrategy = require('passport-google-oauth2').Strategy;

passport.use(new GoogleStrategy(
  {
    clientID : '55986542851-s2drdcm0c86p6a2ud8bkn1uv2cv79e2g.apps.googleusercontent.com',
    clientSecret : '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',
    callbackURL : '/auth/google/callback',
    passReqToCallback : true
  }, function(request, accessToken, refreshToken, profile, done) {
    console.log('profile: ', profile);
    var user = profile;

    done(null, user);
  })
);
passport.serializeUser((user, done) => {
  done(null, user);
});
passport.deserializeUser((user, done) => {
  done(null, user);
});
```

해당 코드에 클라이언트 id와 secret은 api 서비스를 생성할 때 받은 그대로 작성해서 넣으면 된다. 밑에 있는 코드인 serializeUser 메소드와 deserializeUser 메소드는 정확히 이해하지 못하여서 passport의 공식 매뉴얼을 참고하여 작성하였다.

Auth.js

```
var express = require('express');
var router = express.Router();
var passport = require('../config/passport.js');

router.get('/login', function(req,res){
  res.render('auth/login');
});

router.get('/logout', function(req,res) {
  req.logout();
  res.redirect('/');
});

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

router.get('/google/callback',
  passport.authenticate('google'), authSuccess
);

function authSuccess(req, res) {
  res.redirect('/');
}

module.exports = router;
```

Routes의 역할을 함으로 login시 auth/login으로 연결해주고 logout시에는 메인인 /로 연결해준다.

Main.js

```
var express = require('express');
var router = express.Router();

router.get('/', function(req, res){
    res.render('main', {user: req.user});
});

module.exports = router;
~
~
~
```

메인에서는 render시 key값이 user가 되는데 이경우 req.user를 넣었고 main으로 render한다.

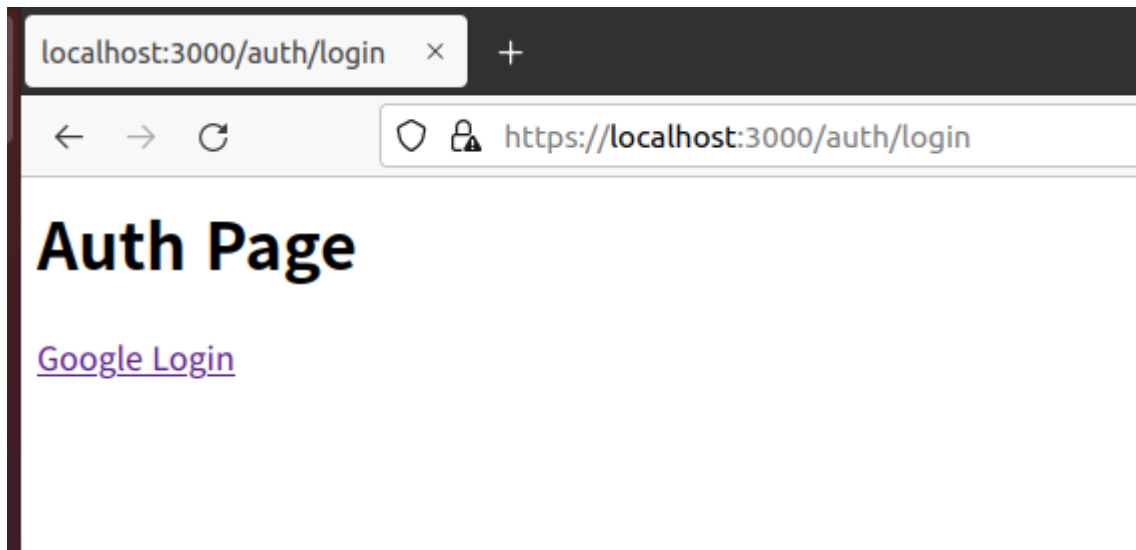
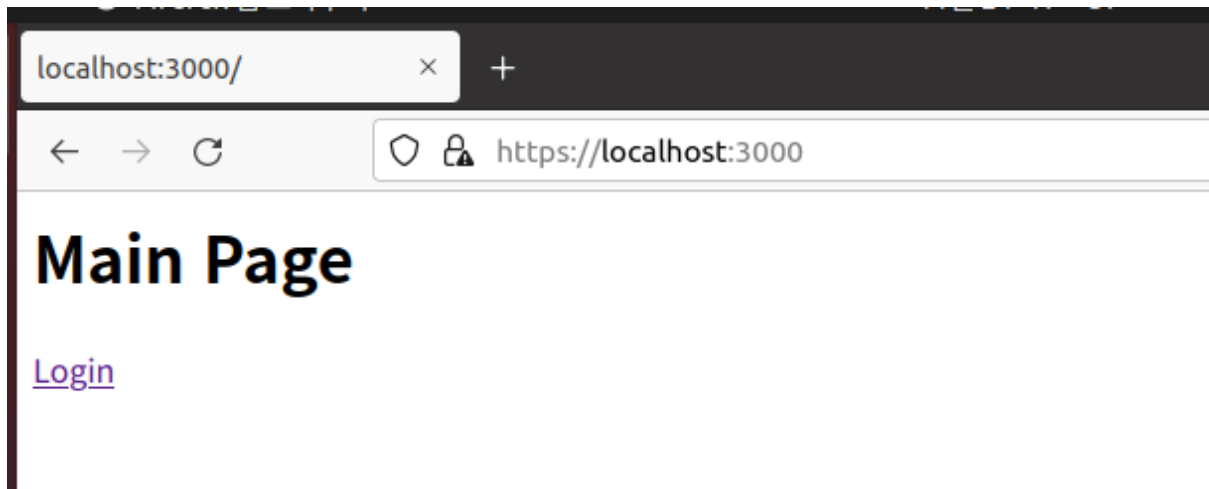
Dockerfile


```
FROM node:12
MAINTAINER Baesihyeon "baesh0408@naver.com"
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["node", "index.js"]
~
~
```

Dockerfile의 경우 저번 과제와 동일하게 작성하였다.

3) 실행결과

1. localhost로 실행



 Google 계정으로 로그인

계정 선택

Nodeweb(으)로 이동



배시

baesi4879@gmail.com



배시현

baesh0408@g.cnu.ac.kr



다른 계정 사용

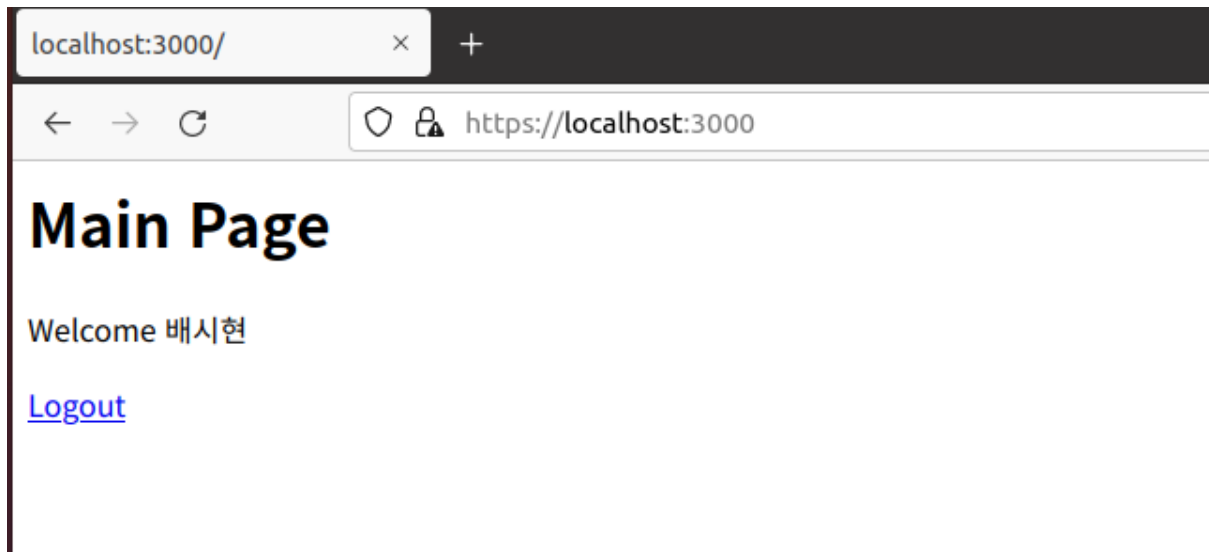
계속 진행하기 위해 Google에서 내 이름, 이메일 주소, 언어
환경설정, 프로필 사진을 Nodeweb과(와) 공유합니다.

한국어 ▼

도움말

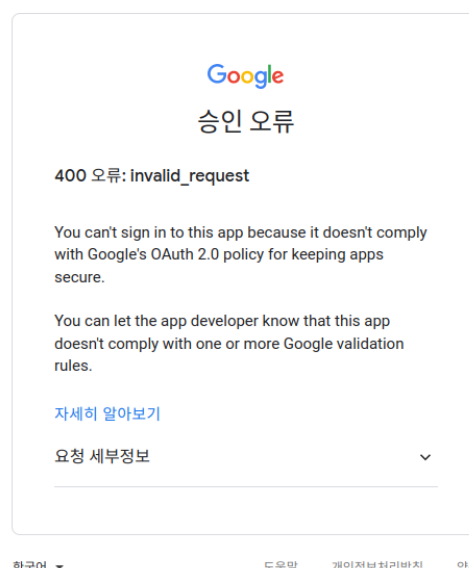
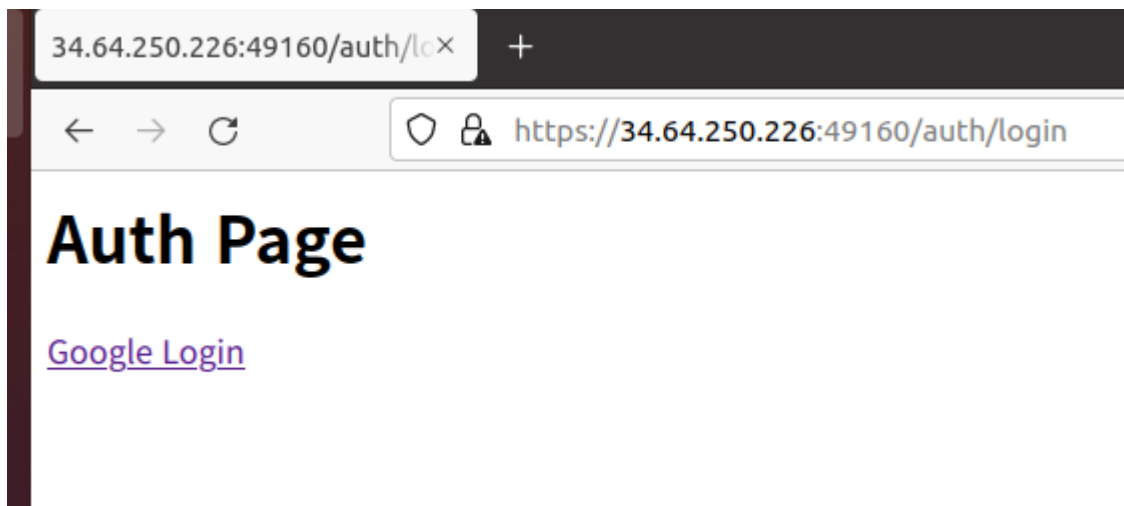
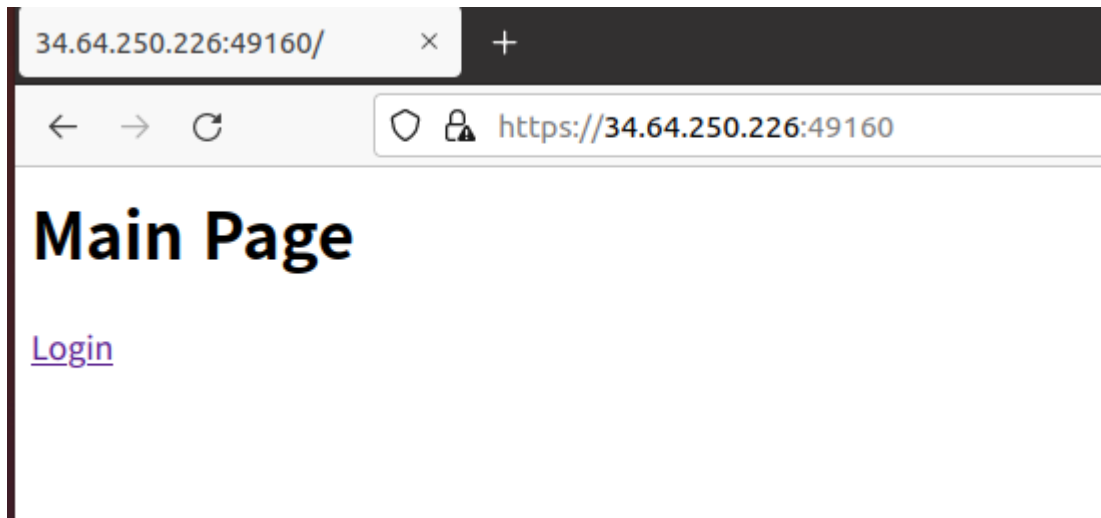
개인정보처리방침

약관



```
baesihyeon@baesihyeon-VirtualBox:~/5week$ sudo node index.js
server on! https://localhost:3000
profile: {
  provider: 'google',
  sub: '101571813213747370851',
  id: '101571813213747370851',
  displayName: '배시현',
  name: { givenName: '시현', familyName: '배' },
  given_name: '시현',
  family_name: '배',
  language: 'ko',
  locale: undefined,
  photos: [
    {
      value: 'https://lh3.googleusercontent.com/a/AATXAJxJ2whEnpi7tHFN5tp-PEZ8UgicykZNatxm_60=s96-c',
      type: 'default'
    }
  ],
  picture: 'https://lh3.googleusercontent.com/a/AATXAJxJ2whEnpi7tHFN5tp-PEZ8UgicykZNatxm_60=s96-c',
  _raw: '{\n' +
    '  "sub": "101571813213747370851",\n' +
    '  "name": "배시현",\n' +
    '  "given_name": "시현",\n' +
    '  "family_name": "배",\n' +
    '  "picture": "https://lh3.googleusercontent.com/a/AATXAJxJ2whEnpi7tHFN5tp-PEZ8UgicykZNatxm_60\\u003ds96-c"\n' +
    '  "locale": "ko"\n' +
    '}',
  _json: {
    sub: '101571813213747370851',
    name: '배시현',
    given_name: '시현',
    family_name: '배',
    picture: 'https://lh3.googleusercontent.com/a/AATXAJxJ2whEnpi7tHFN5tp-PEZ8UgicykZNatxm_60=s96-c',
    locale: 'ko'
  }
}
```

2. 외부 ip로 접속



여러방법을 사용해봤지만 문제를 찾아내지 못하였다.