20190706

기본 설치

- mkdir 1stproject
- cd 1stproject
- express --viewejs
- npm i
- npm i --save-dev nodemon
- package.json

```
{
  "name": "1stproject-test",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "nodemon ./bin/www"  //이부분 nodemon으로 수정
  },
  "dependencies": {
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "ejs": "~2.6.1",
    "express": "~4.16.1",
    "http-errors": "~1.6.3",
    "morgan": "~1.9.1"
  },
  "devDependencies": {
    "nodemon": "^1.19.1"
  }
}
```

- npm start

  -> http://localhost:3000 URL로 Browser에서 확인

로그인 페이지 만들기

- public>javscripts>login.css

```
@import url('https://fonts.googleapis.com/css?
family=Source+Sans+Pro&display=swap');
/*Google 폰트 web에서 가져오기*/
body{
  margin: 0;
```

```
  padding: 0;
  font-family: 'Source Sans Pro', sans-serif;
}
body{
  position: relative;
  z-index: 1;
}
body:after{
  width:100%;
  height:100vh;
  position: absolute;
  opacity: 0.5!important;
  filter: alpha(opacity=50);
  z-index: -1;
  content: "";
  background:url('../images/passport.jpg'); /* Background img 넣기*/
  background-size:cover;
  color:white;
}
.container{
  position: absolute;
  top:50%;
  left:50%;
  transform: translate(-50%,-50%);
  width:340px;
  text-align: center;
}
#login-form{
  position: relative;
  box-sizing: border-box;
  padding:60px 30px;
  transition: .5s;
  margin-top: 250%;
}

#login-form h1{
  font-size:32px;
  color:black;
  /* text-transform: uppercase; */
}
#login-form input,button,a{
  font-family: 'Source Sans Pro', sans-serif;
  display: block;
  width: 100%;
  padding: 10px 20px;
```

```
    box-sizing: border-box;
    margin-bottom: 20px;
    border-radius: 25px;
    outline: none;
    font-size: 14px;
    letter-spacing: 1px;
    color:black;
    /* text-transform: uppercase; */
    border:none;
    background: rgba(255, 255, 255, 0.1);
}
#login-form input::placeholder{
    color: black;
}
#login-form button
{
    color:#fff;
    background:#0072ff;
    font-size: 16px;
    border:none;
    cursor: pointer;
}
#login-form button:hover{
    background:#005aca;
}
/* #login-form button{
text-decoration: none;
color:black;
text-transform: uppercase;
} */
#login-form a{
    text-decoration: none;
    color:#FFF;
    text-transform: uppercase;
}
```

- public > images > passport.jpg 파일 넣어두기.
- views > index.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>LOGIN</title>
```

```
    <script src="https://code.jquery.com/jquery-3.4.1.js"
integrity="sha256-WpOohJOqMqqyKL9FccASB9O0KwACQJpFTUBLTYOVvVU="
crossorigin="anonymous"></script>
    <!-- jquery cdn 넣기 -->
    <link rel="stylesheet" href="stylesheets/login.css">   <!-- 위에서 만
들었던 css 가져오기 -->
  </head>
  <body>
    <section>
      <div class="container">
        <div id="login-form">
          <form>
            <h1>Login</h1>
            <input id="login_id" type="text" placeholder="ID">
            <input id="login_pwd" type="password"
placeholder="Password"><br>
            <button id="login_btn" type="submit">Login</button>
            <div id="signup">
              <a href="/signup">Sign up</a>
            </div>
          </form>
        </div>
      </div>
    </section>
  </body>
</html>
```

Sign up 페이지 만들기

login.ejs 페이지와 css는 동일하지만 UI는 다르게 만들기

- views > signup.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>LOGIN</title>
    <link rel="stylesheet" href="stylesheets/login.css">
    <script src="https://code.jquery.com/jquery-3.4.1.js"
integrity="sha256-WpOohJOqMqqyKL9FccASB9O0KwACQJpFTUBLTYOVvVU="
crossorigin="anonymous"></script>
    <script src="javascripts/javascript.js"></script>
  </head>
```

```
  <body>
    <section>
      <div class="container">
        <div id="login-form">
          <div>
            <h1>Sign up</h1>
            <div id="signup_div">
              <input id="user_id" type="text" placeholder="ID"> <!--로그
인시 입력할 ID-->
              <input id="pwd" type="password" placeholder="PASSWORD">
<!--로그인시 입력할 PWD-->
              <input id="name" type="text" placeholder="NAME"> <!--사용자
이름-->
              <input id="user_num" type="text" placeholder="ID NUMBER">
<!--사용자 주민번호-->
              <button id="sign_up" type="submit">Sign up</button>
            </div>
          </div>
        </div>
      </div>
    </section>
  </body>
</html>
```

login.ejs와 똑같지만 <head>부분에 javascript.js 스크립트를 추가한다.

- app.js 에 라우팅 추가

```
...
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/signup',require('./routes/signup')); //라우팅 추가
...
```

- routes > signup.js

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

/* GET signup page. */
router.get('/', function(req, res, next) {
  res.render('signup');
});

module.exports = router;
```

Sign up 버튼을 눌렀을 때 세션에 저장할 member_insert.js

- sign_up 버튼을 눌렀을 때 일어날 이벤트를 jquery로 만들기
  - public > javascripts > javascript.js

```
$(document).ready(function(){
  $("#sign_up").click(function(){
    const user_id = $("#user_id").val();
    const pwd = $("#pwd").val();
    const name = $("#name").val();
    const user_num = $("#user_num").val();
    const send_params={
      user_id,
      pwd,
      name,
      user_num
    };
    $.post('/member_insert',send_params,function(data,status){
      const parsed_data=JSON.parse(data);
      $("#signup_div").html("<h1>"+parsed_data.msg+"</h1><br><a
href='/'>Login</a>");
    });//post
  });//signup
});
```

입력 받은 데이터들을 받아서 Javascript 객체로 파싱해서 msg값을 h1태그에 넣고 로그인 페이지로 갈 <a>태그도 작성.

  - public > routes > member_insert.js

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

/* GET member_insert page. */
router.post('/', function(req, res, next) {
  const result={msg : req.body.name+", Signed up!"};
  res.json(JSON.stringify(result));
});

module.exports = router;
```

post 방식으로 받아옴.

Session-express 설치

- session에 id , pwd , name , user_num을 저장하기 위해 설치한다
- npm i express-session
- package.json

```
"dependencies": {
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "ejs": "~2.6.1",
    "express": "~4.16.1",
    "express-session": "^1.16.2",
    "http-errors": "~1.6.3",
    "morgan": "~1.9.1"
  },
```

express-session이 설치되어있는 것을 확인 할 수 있다.

- app.js

```
...
app.use(logger('dev'));
app.use(express.static(path.join(__dirname, 'public'))); //순서바꿔줌.
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(cookieParser('SSORRY CHOI')); //암호화 키 추가
app.use(require('express-session')({ //추가
  resave:false,
  saveUninitialized:true,  //Session의 넣어둘 내용이 없어도 저장할 것인가?
  secret:'SSORRY CHOI',    //필수,, Session ID값을 암호화 한 암호화 키
```

```
  cookie:{
    httpOnly:true,
    secure: false
  }
}));
...
```

- routes > member_insert.js

```
/* GET member_insert page. */
router.post('/', function(req, res, next) {
  req.session.user_id=req.body.user_id;
  req.session.pwd=req.body.pwd;
  req.session.name=req.body.name;
  req.session.user_num=req.body.user_num;
  const result={msg : req.body.name+", Signed up!"};
  res.json(JSON.stringify(result));
});
```

입력받은 값들을 session 값에 넣는다.

Login 하기

- Sign up 페이지에서 받아온 id값과 pwd값을 이용한다.
- views > index.ejs

```
...
<head>
  ...
  <script src="javascripts/javascript.js"></script>
</head>
<section>
  <div class="container">
    <div id="login-form">
      <form>
        <h1>Login</h1>
        <input id="login_id" type="text" placeholder="ID">
        <input id="login_pwd" type="password" placeholder="Password">
<br>
        <button id="login_btn" type="submit">Login</button>
        <div id="signup">
          <a href="/signup">Sign up</a>
        </div>
      </form>
```

```
      </div>
    </div>
  </section>
```

javascripts > javascript.js 를 head부분에 추가한다.

- javascripts > javascript.js

```
...
$("#login_btn").click(function(){
  const login_id = $("#login_id").val();
  const login_pwd = $("#login_pwd").val();
  const send_params={
    user_id: login_id,
    pwd: login_pwd
  }
  $.post('/login',send_params,function(data,status){
    try{
      alert(JSON.parse(data).msg);
      $("#login_id").val()="";
      $("#idnumber_div").html()
    }catch{
      window.location.reload(true);
    }
  });//post

});//login_btn
...
```

Login 버튼을 눌렀을때의 이벤트를 jquery로 만들어 준다.

routes>login.js 으로 post 방식으로 값들을 보낸다.

- routes > index.js 에서 loginState값을 받아준다.

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

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', {loginState:req.session.loginState});
});

module.exports = router;
```

- routes > login.js

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

/* GET login page. */
router.post('/', function(req, res, next) {
  const result={msg:""};
  if((req.session.user_id===req.body.user_id)&&
(req.session.pwd===req.body.pwd)){
    req.session.loginState=true;
    res.redirect('/');
  }else{
    result.msg="Try again";
    res.json(JSON.stringify(result));
  }
});

module.exports = router;
```

session에 저장된 id,pwd 값을 비교해서 동시에 맞을 경우 loginState값을 true로 바꾸고 index.ejs를 redirect한다.

- views > index.ejs

```
<body>
  <section>
    <div class="container">
      <div id="login-form">
        <%
           if(loginState){
           %>
          <h1>Hello World!<br> Success to login</h1>
          <button onclick="window.location.href='/about'">Go to
HomePage</button>

          <a href="/" id="logout_btn">Logout</a>
          <%}else{
                  %>

            <form>
              <h1>Login</h1>
              <input id="login_id" type="text" placeholder="ID">
              <input id="login_pwd" type="password"
placeholder="Password"><br>
              <button id="login_btn" type="submit">Login</button>
              <div id="signup">
```

```
                    <a href="/signup">Sign up</a>
                  </div>
                  <% }
                     %>
                    </form>
                  </div>
                </div>
              </section>
            </body>
```

body부분을 수정한다. login되었을 경우에 loginState값이 true이므로 Hello World!와 동시에 Homepage로 가는 버튼이 생기고 login에 실패 했을 경우엔 login form이 나오게 한다.

- app.js

```
...
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/signup',require('./routes/signup'));
app.use('/member_insert',require('./routes/member_insert'));
app.use('/login',require('./routes/login'));
...
```

login 라우터 추가

Contents가 보여질 about.ejs 페이지 만들기

- views > about.ejs

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>UNDEFINED</title>
    <!-- Bootstrap core CSS & JS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.
min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
```

```
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper
.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.mi
n.js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
integrity="sha256-pasqAKBDmFT4eHoN2ndd6lN370kFiGUFyTiUHWhU7k8="
crossorigin="anonymous"></script>
    <!-- Custom styles for this template -->
    <link href="stylesheets/style.css" rel="stylesheet">
    <script src="javascripts/javascript.js"></script>
  </head>
  <body id="page-top">
    <!-- Navigation -->
    <nav class="navbar navbar-expand-lg navbar-light bg-light shadow
fixed-top">
      <div class="container">
        <a class="navbar-brand" href="#">ID Check</a>
        <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarResponsive" aria-
controls="navbarResponsive" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item active">
              <a class="nav-link" href="#">Home
                <span class="sr-only">(current)</span>
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#">About</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#">Services</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#">Contact</a>
            </li>
```

```
            <li class="nav-item">
              <a class="nav-link" href="/" id="logout_btn">Logout</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>

    <!-- Full Page Image Header with Vertically Centered Content -->
    <header class="masthead">
      <div class="container h-100">
        <div class="row h-100 align-items-center">
          <div class="col-12 text-center ">
            <h1 class="font-weight-light">What is your ID Number?</h1>
            <p id="idnumber_div"class="lead">Your ID Number is : <br>
              <%=user_num%>
                </p>
          </div>
        </div>
      </div>
    </header>
    <section id="about">
      <div class="container">
        <div class="row">
          <div class="col-lg-8 mx-auto">
            <h2>About this page</h2>
            <p class="lead">This is a great place to talk about your
webpage. This template is purposefully unstyled so you can use it as a
boilerplate or starting point for you own landing page designs! This
template features:</p>
            <ul>
              <li>Clickable nav links that smooth scroll to page
sections</li>
              <li>Responsive behavior when clicking nav links perfect
for a one page website</li>
              <li>Bootstrap's scrollspy feature which highlights which
section of the page you're on in the navbar</li>
              <li>Minimal custom CSS so you are free to explore your
own unique design options</li>
            </ul>
          </div>
        </div>
      </div>
    </section>
```

```
    <section id="services" class="bg-light">
      <div class="container">
        <div class="row">
          <div class="col-lg-8 mx-auto">
            <h2>Services we offer</h2>
            <p class="lead">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Aut optio velit inventore, expedita quo laboriosam
possimus ea consequatur vitae, doloribus consequuntur ex. Nemo
assumenda laborum vel, labore ut velit dignissimos.</p>
          </div>
        </div>
      </div>
    </section>

    <section id="contact">
      <div class="container">
        <div class="row">
          <div class="col-lg-8 mx-auto">
            <h2>Contact us</h2>
            <p class="lead">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Vero odio fugiat voluptatem dolor, provident
officiis, id iusto! Obcaecati incidunt, qui nihil beatae magnam et
repudiandae ipsa exercitationem, in, quo totam.</p>
          </div>
        </div>
      </div>
    </section>

    <!-- Footer -->
    <footer class="py-5 bg-dark">
      <div class="container">
        <p class="m-0 text-center text-white">Copyright &copy; Your
Website 2019</p>
      </div>
      <!-- /.container -->
    </footer>
    <!-- Bootstrap core JavaScript -->
    <!-- Plugin JavaScript -->
    <script src="javascripts/jquery.easing.min.js"></script>
    <!-- Custom styles for this template -->
    <link href="stylesheets/scrolling-nav.css" rel="stylesheet">
  </body>

</html>
```

- app.js

```
...
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/signup',require('./routes/signup'));
app.use('/member_insert',require('./routes/member_insert'));
app.use('/login',require('./routes/login'));
app.use('/about',require('./routes/about'));
...
```

about 라우터 추가

- routes > about.js

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

/* GET home page. */
router.get('/', function(req, res, next) {
  console.log(req.session.user_num);
  res.render('about',
{id:req.session.user_id,name:req.session.name,user_num:req.session.user
_num});
});

module.exports = router;
```

Logout

- javascripts > javascript.js

```
...
$('#logout_btn').click(function(){
  $.get("logout",function(data,status){
    location.reload(true);
  });
});//logout_btn
...
```

logout 버튼 클릭시 이벤트 추가하기

- routes > logout.js

```
var express = require('express');
```

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

/* GET home page. */
router.get('/', function(req, res, next) {
    const result={msg:""};
    if(req.session.user_id){
        req.session.destroy(function(err){
            res.redirect('/');
        })
    }
});

module.exports = router;
```

- app.js

```
...
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/signup',require('./routes/signup'));
app.use('/member_insert',require('./routes/member_insert'));
app.use('/login',require('./routes/login'));
app.use('/about',require('./routes/about'));
app.use('/logout',require('./routes/logout'));
...
```

logout 라우터 추가

about.ejs 페이지에서 logout 버튼을 눌렀을 시

- about.ejs - menu bar

```
<div class="collapse navbar-collapse" id="navbarResponsive">
  <ul class="navbar-nav ml-auto">
    <li class="nav-item active">
      <a class="nav-link" href="#">Home
        <span class="sr-only">(current)</span>
      </a>
    </li>
    <%
       if(name){
      %>
      <li class="nav-item">
        <a class="nav-link" href="#">About</a>
      </li>
```

```
          <li class="nav-item">
            <a class="nav-link" href="#">Services</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Contact</a>
          </li>
          <li class="nav-item">

            <a class="nav-link" href="/about"
      id="about_logout">Logout</a>
          </li>
          <%
             }else{
             %>


            <li class="nav-item">
              <a class="nav-link" href="/" id="">Login</a>
            </li>
            <%
               }
               %>
              </ul>
            </div>
```

about.js에서 name에 req.session.name을 넣어 두었는데 그 값이 있으면(로그인했으면)menu bar가 전부 보이게 설정한다.

name값이 거짓(로그인이 되어있지 않다면) 이면 login 메뉴가 보이게 설정한다.

- about.ejs - 본문내용

```
    <div class="col-12 text-center ">
      <%
         if(name){
         %>
        <h1 class="font-weight-bold">What is your ID Number?</h1>
        <p id="idnumber_div"class="lead">
        <h3 class="font-weight-light"><%=name%>'s ID Number is : </h3>
          <%=user_num%>
            </p>
          <%
             }else{
             %>
            <h1 class="font-weight-bold">Please login</h1>
            <%
```

```
            }
            %>
          </div>
```

about.ejs 가운데 부분에도 변화를 준다.

- routes > about_logout.js

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

/* GET about_logout */
router.get('/', function(req, res, next) {
  const result={msg:""};
  if(req.session.name&&req.session.user_num){
    req.session.destroy(function(err){
      res.redirect('/about');
    })
  }
});

module.exports = router;
```

menu_bar 에서 logout버튼을 눌렀을때 발생하는 이벤트 설정

- app.js

```
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/signup',require('./routes/signup'));
app.use('/member_insert',require('./routes/member_insert'));
app.use('/login',require('./routes/login'));
app.use('/about',require('./routes/about'));
app.use('/logout',require('./routes/logout'));
app.use('/about_logout',require('./routes/about_logout'));
```

about_logout 라우터 추가