

고급웹프로그래밍 프로젝트

W e b P r o g r a m m i n g P r o j e c t

2014105022 김완기

목 차

1. 프로젝트 소개

—

2. 프로젝트 구현

—

3. 프로젝트 소감

—

프로젝트 소개

개요

이름

Project Name

경매왕

개요

Project Outline

다양한 종류의 물품을 경매할 수 있는
실시간 경매장 웹 서비스

효과

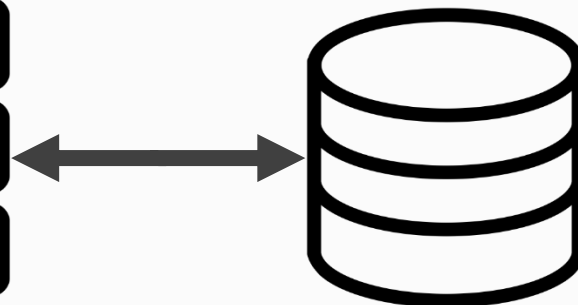
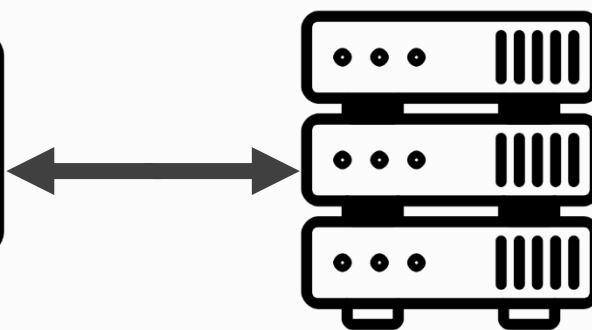
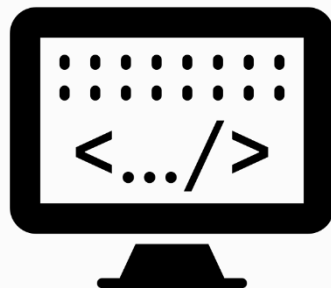
Effectiveness

온라인 경매를 통해 자유롭게 물건을 사고 팔 수 있다.

프로젝트 소개

시스템 개요도

경매왕



프로젝트 소개

개발 도구



Front-End

HTML + CSS + JavaScript

수업 시간에 배운 것,
+
jQuery, Bootstrap 사용



Server

Node.js

비동기식 실행,
NPM 이용
(Express 모듈)



Database

MongoDB

신뢰성보다는
성능에 중점,
JSON 형식

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

1. 회원 가입 - crypto 모듈 이용

crypto 모듈을 이용하여
비밀번호 암호화와 복호화를
자동으로 되게 해 놓았다.

LocalStrategy
Passport를 이용하여
아이디 중복 확인도 가능했다.

```
var UserSchema = mongoose.Schema({  
  id: {type: String, index: 'hashed', required: true,  
    unique: true, 'default' : ''},  
  hashed_password: {type: String, required: true,  
    'default' : ''},  
  salt: {type: String, required: true},  
  name: {type: String, index: 'hashed', 'default' :  
    ''},  
  age: {type: Number, 'default' : -1},  
  address : {type: String},  
  phone : {type: String},  
  created_at: {type: Date, index : {unique : false},  
    'default' : Date.now}  
});
```

```
passport.use('local-signup', new LocalStrategy({  
  usernameField: 'id',  
  passwordField: 'password',  
  passReqToCallback: true  
}, function (req, id, password, done) {  
  // 요청 파라미터 중 name 파라미터 확인  
  var paramName = req.body.name || req.query.name;  
  var paramAge = req.body.age || req.query.age;  
  var paramAddress = req.body.address ||  
    req.query.address;  
  var paramPhone = req.body.phone || req.query.phone;  
  console.log('passport의 local-signup 호출됨 : ' + id + ',  
    ' + password + ', ' + paramName);  
  
  // User.findOne이 blocking되므로 async 방식으로 변경할 수도  
  있음  
  process.nextTick(function () {  
    UserModel.findOne({  
      'id': id  
    }, function (err, member) {  
      // 오류가 발생하면
```

프로젝트 구현

기능

- 2. 로그인
 - 3. 내 정보 보기
 - 4. 경매 게시판
- mongoose 모듈 이용

LocalStrategyPassport
와 Mongoose 모듈을 이용
하여, UserModel과
PostModel에 있는 데이터들
을 웹에 가져 왔다.

```
/* 몽고디비 모듈 사용 */  
var mongoose = require('mongoose');  
mongoose.set('useCreateIndex', true);  
/* 데이터베이스 객체를 위한 변수 선언 */  
var database;  
  
/* 데이터베이스 모델 객체를 위한 변수 선언 */  
var UserModel;  
var PostModel;
```

```
passport.use('local-login', new LocalStrategy({  
  usernameField: 'id',  
  passwordField: 'password',  
  passReqToCallback: true  
}, function (req, id, password, done) {  
  console.log('passport의 local-login 호출됨 : ' + id + ',  
    ' + password);  
  
  UserModel.findOne({  
    'id': id  
  }, function (err, member) {  
    if (err) {  
      return done(err);  
    }  
  })  
});
```

```
var newPost = new PostModel({  
  'id': req.user.id,  
  'title': req.body.title,  
  'content': req.body.content,  
  'photo': req.user.id + "_" + count + extension,  
  'price': req.body.price  
});  
  
newPost.save(function (err) {  
  if (err) {  
    throw err;  
  }  
  console.log('게시글 추가함.');
```

```
res.redirect('/auction');
```

```
});
```

```
var socketio = require('socket.io');
```

5. 실시간 채팅(경매장)

6. 입찰

- socket.io 모듈 이용

socket.io 모듈을 이용하여
실시간 웹 서버 통신이 가능하
도록 했다.

```
var io = socketio.listen(server);
console.log('socket.io 요청을 받아들일 준비가 되었습니다.');
```

```
io.on('connection', function (socket) {
  console.log('connection info : ',
    socket.request.connection._peername);

  socket.remoteAddress =
    socket.request.connection._peername.address;
  socket.remotePort =
    socket.request.connection._peername.port;

  // 'message' 이벤트를 받았을 때의 처리
  socket.on('message', function (message) {

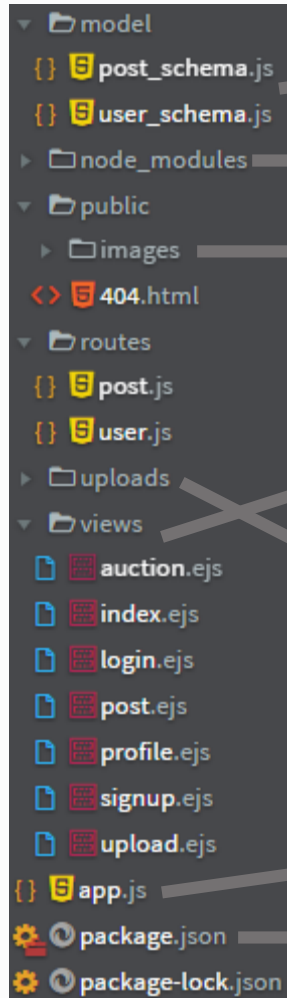
    if (message.receipient == 'ALL') {
      // 나를 포함한 모든 클라이언트에게 메시지 전달
      io.sockets.emit('message', message);
    }
  });

  // 'price' 이벤트를 받았을 때의 처리
  socket.on('price', function (price) {
    PostModel.update({
      _id: price.objId
    }, {
      $set: {
        price: parseInt(price.data),
        bidder: price.sender
      }
    }, function(err, docs){
    });
  });
});
```

```
<script src="https://cdn.socket.io/socket.io-1.7.0.js">
</script>
```


프로젝트 구현

구성 요소



MVC Pattern
Model

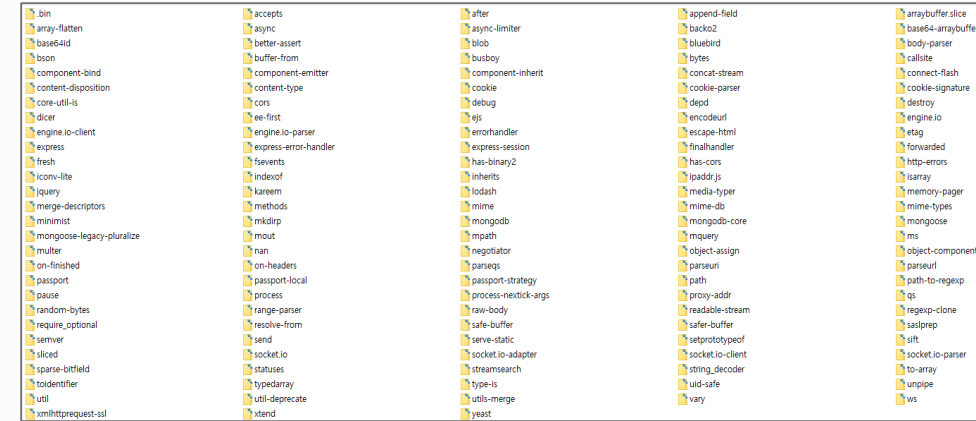
public : 웹 페이지에
필요한 리소스들
저장하는 디렉토리

MVC Pattern
View

uploads : 웹 서버에
클라이언트가 전송한
파일 저장하는 디렉토리

MVC Pattern
Controller

Node.js를 이용하여
웹 서버를 구축할 때
사용하는 module들이
저장되는 디렉토리

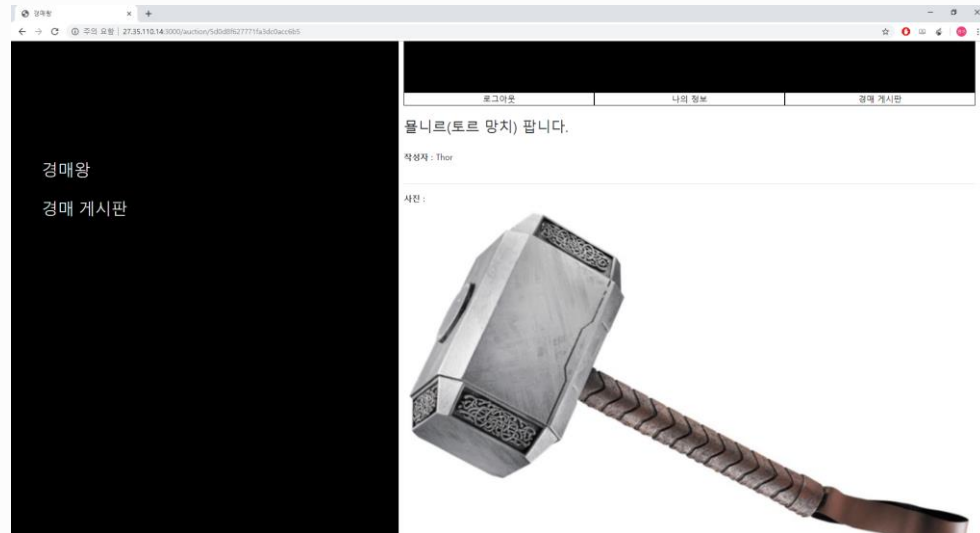
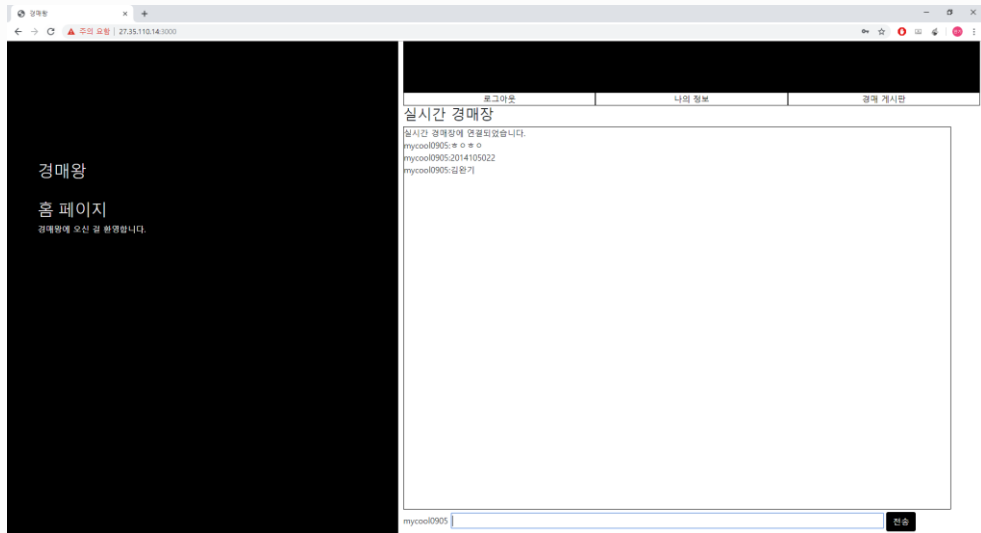
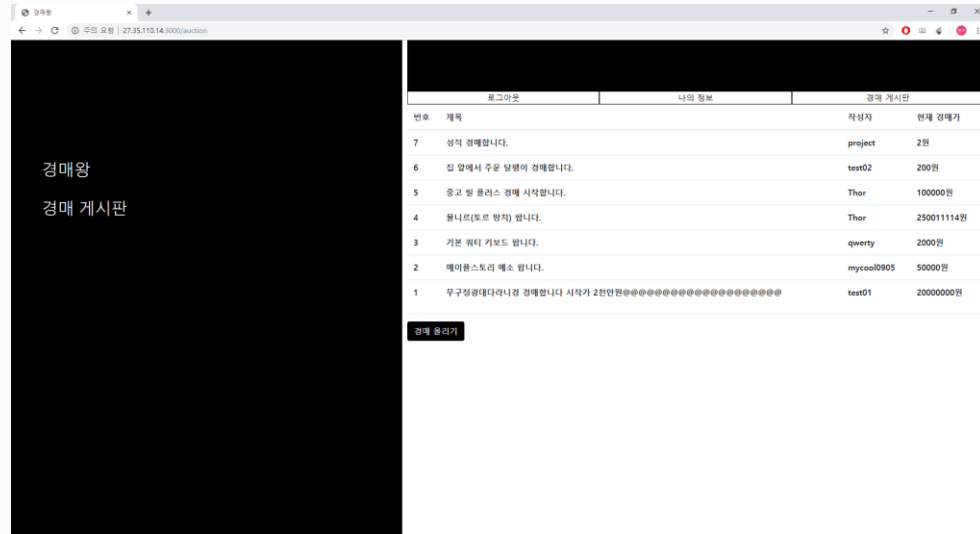
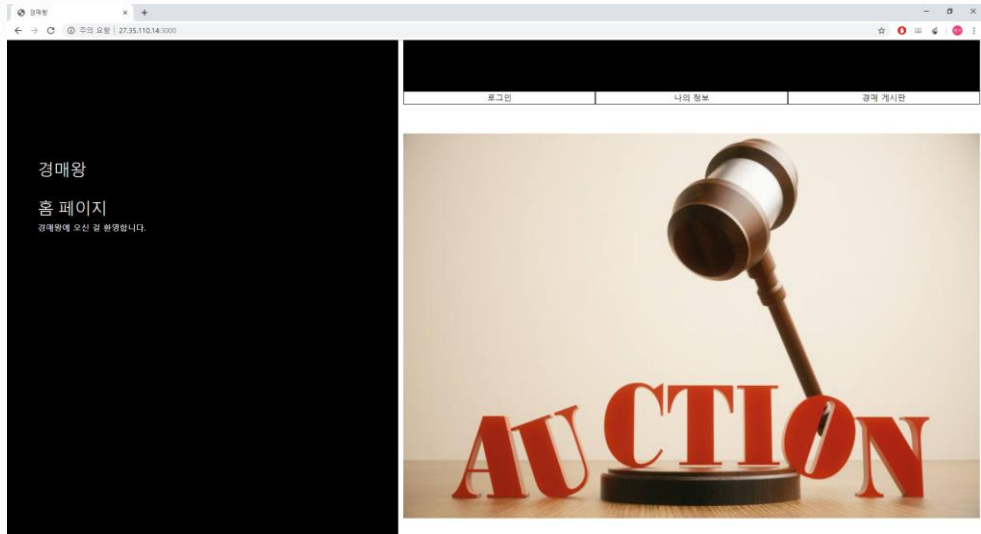


Node.js를 이용하여
웹 서버를 구축할 때
사용하는 module들을
저장하는 JSON

```
{
  "name": "WebProgrammingProject",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "connect-flash": "^0.1.1",
    "cookie-parser": "^1.4.4",
    "cors": "^2.8.5",
    "ejs": "^2.6.1",
    "errorhandler": "^1.5.1",
    "express": "^4.17.1",
    "express-error-handler": "^1.1.0",
    "express-session": "^1.16.1",
    "jquery": "^3.4.1",
    "mongodb": "^3.2.6",
    "mongoose": "^5.5.11",
    "multer": "^1.4.1",
    "passport": "^0.4.0",
    "passport-local": "^1.0.0",
    "path": "^0.12.7",
    "serve-static": "^1.14.1",
    "socket.io": "^2.2.0"
  }
}
```

프로젝트 구현

웹 페이지



프로젝트 구현

깃 허브

mycool0905 / WebProgramming

Watch 0

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

WebProgrammingLecture

Edit

Manage topics

17 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

mycool0905 2019-06-22-09:03 => ProjectFirstDraft

Latest commit b2bcee3 3 hours ago

Assignments

2019-05-26-02:36

27 days ago

Project

2019-06-22-09:03 => ProjectFirstDraft

3 hours ago

README.md

Initial commit

27 days ago

고급웹프로그래밍프로젝트.pdf

2019-06-07-06-22

15 days ago

고급웹프로그래밍프로젝트.pptx

2019-06-07-06-22

15 days ago

README.md

WebProgramming

WebProgrammingLecture

<https://github.com/mycool0905/WebProgramming>



프로젝트를 진행할 때,
수업 시간에 배웠던
기본이 되는
웹 프로그래밍 언어의 복습이
됐을 뿐만 아니라
요즘 많은 곳에서
사용하고 있는 Node와
MongoDB까지 다뤄보는
좋은 기회였던 것 같다.



**THANK YOU
FOR
WATCHING**

Web Programming Project