

포트폴리오

정 기 훈

<https://github.com/jung-ki-hun>

보유 기술

1. 언어

1) Java

- ① Multi Thread Programming
- ② java swing을 활용한 gui 개발

2) JavaScript

- ① node.js를 활용한 restAPI 개발
- ② jquery라이브러리
- ③ dom 활용가능

3) C/C++/C#

- ① Multi Thread Programming
- ② dll 라이브러리 제작
- ③ 아두이노 코딩

4) db

- ① sql로 CRUD
- ② nosql로 CRUD

2. 사용 가능 툴

1) git

- ① github / bitbucket 연동
- ② issues 관리

2) Jira

- ① issues 관리
- ② 프로젝트 일정관리

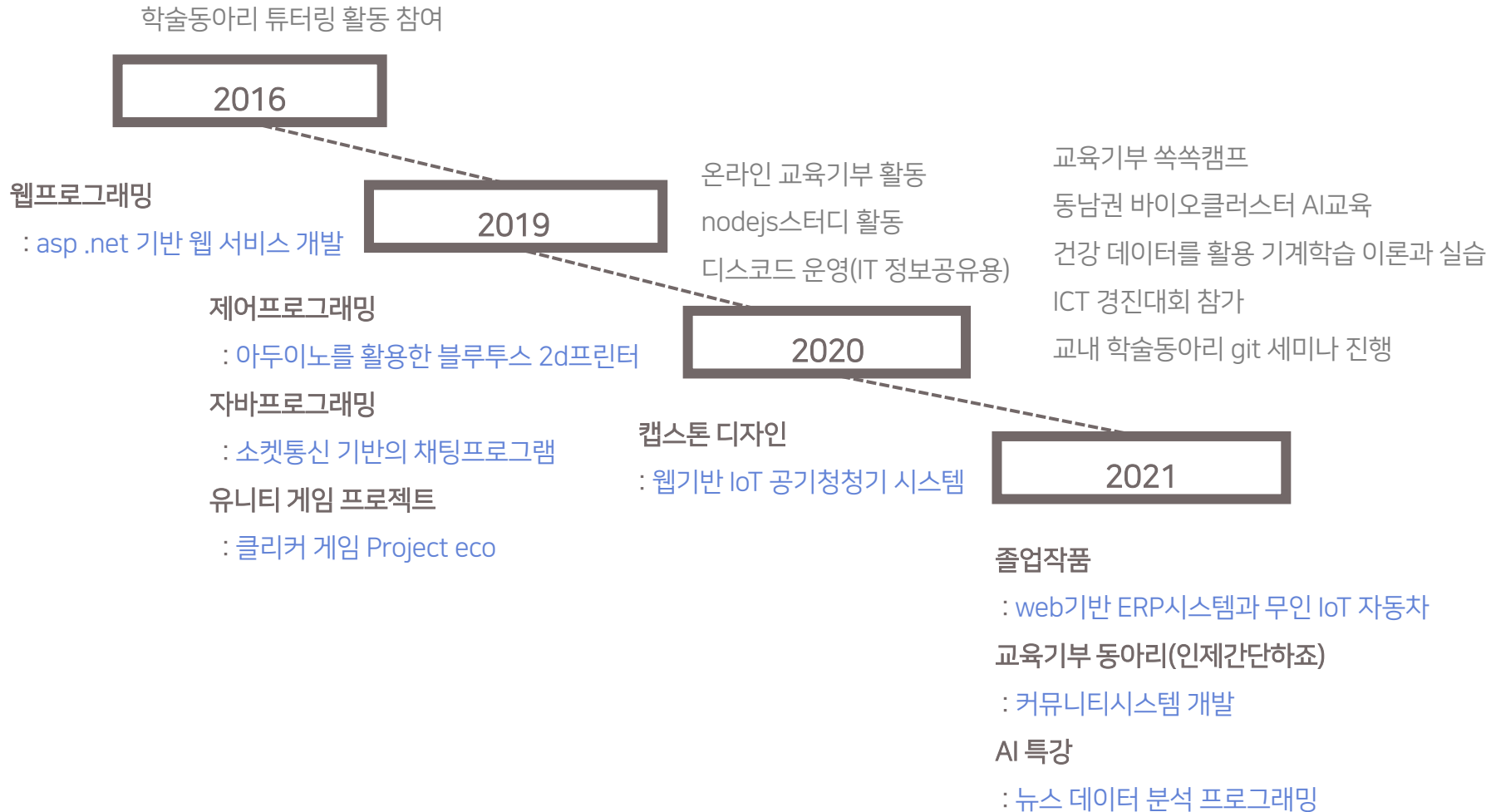
3) 운영체제(OS)

- ① centos 설치 및 설정

4) etc

- ① Rhinoceros 5
- ② Adobe XD

걸어온 길



프로젝트 Programing Box

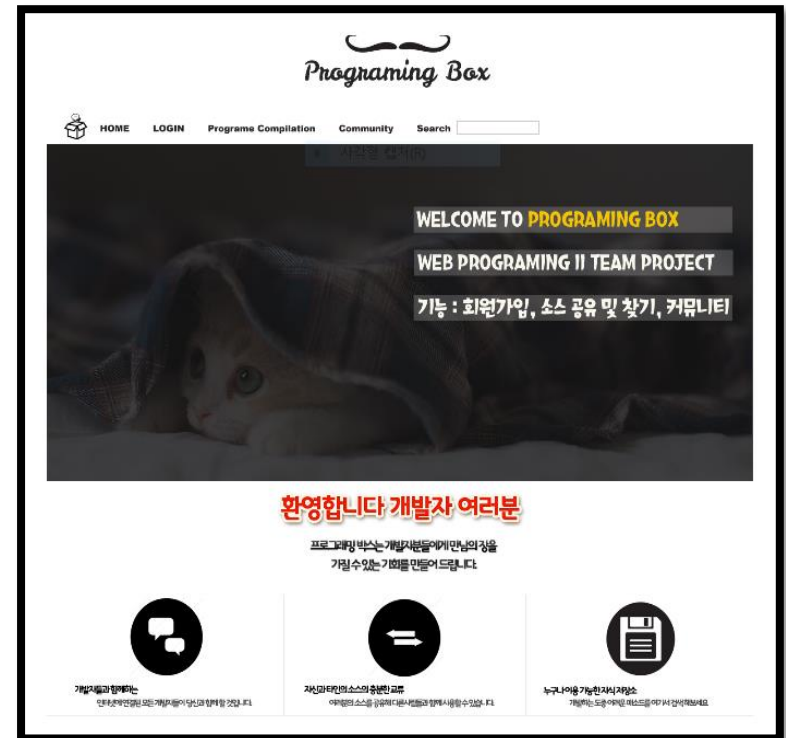
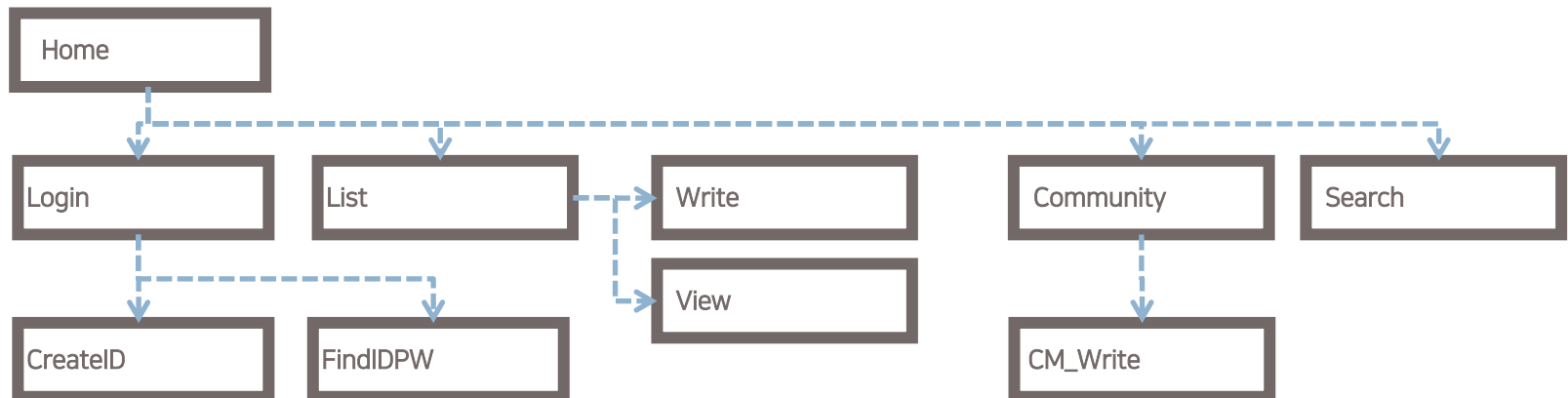
1. 특징

- ① asp.net와 mysql로 개발
- ② 개발자 커뮤니티 사이트
- ③ 방명록과 게시판 기능 구현
- ④ mysql를 db로 사용
- ⑤ 튜장으로 활동

2. 소스코드 위치

- ① https://github.com/jung-ki-hun/project_Asp

파일 구조



프로젝트 Programing Box

프로젝트의 주요 코드

```
protected void ImageButton1_Click(object sender, ImageClickEventArgs e) //로그인하기 버튼
{
    try
    {
        string connectionString = "server=(local)\\SQLExpress; Integrated Security=true; database=userdb";
        SqlConnection conn = new SqlConnection(connectionString);

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conn;
        cmd.CommandText = "select * FROM userdb WHERE id = '" + TextBox3.Text + "' AND Password = '" + TextBox2.Text + "'";

        conn.Open();
        SqlDataReader reader = cmd.ExecuteReader();
        reader.Read();
        if (reader.HasRows)
        {
            Session["User"] = TextBox3.Text;
            Response.Redirect("Home.aspx");
        }
        else
        {
            Label2.Text = "아이디 또는 비밀번호가 잘못되었습니다.";
            TextBox3.Text = "";
        }
        conn.Close();
    }
    catch (Exception ee)
    {
        TextBox3.Text = ee.Message;
    }
}
```

```
protected void Button1_Click(object sender, EventArgs e)
{
    string connectionString = "server=(local)\\SQLExpress; Integrated Security=true; database=userdb";
    SqlConnection conn = new SqlConnection(connectionString);
    conn.Open();
    if (Request["Action"].CompareTo("Reply") != 0) //답변
    {
        string strSql = "INSERT INTO tblboard(name, title, contents, type, readcount, refer, depth, pos) VALUES ('"
            + TextBox3.Text + "', '"
            + TextBox2.Text + "', '"
            + TextBox4.Text + "', '"
            + DropDownList1.SelectedValue + "', "
            + "0, 0, 0, 0)";
        SqlCommand cmd = new SqlCommand(strSql, conn);
        cmd.ExecuteNonQuery();
        strSql = "UPDATE tblboard SET refer=num WHERE refer=0"; //번호넘기는거
        cmd = new SqlCommand(strSql, conn);
        cmd.ExecuteNonQuery();
    }
    else //일반
    {
        int iRefer = int.Parse(Request["refer"]);
        int iDepth = int.Parse(Request["depth"]);
        int iPos = int.Parse(Request["pos"]);
        string strSql = "UPDATE tblboard SET pos=pos+1 WHERE refer=" + iRefer + " and pos>=" + iPos;
        SqlCommand cmd = new SqlCommand(strSql, conn);
        cmd.ExecuteNonQuery();

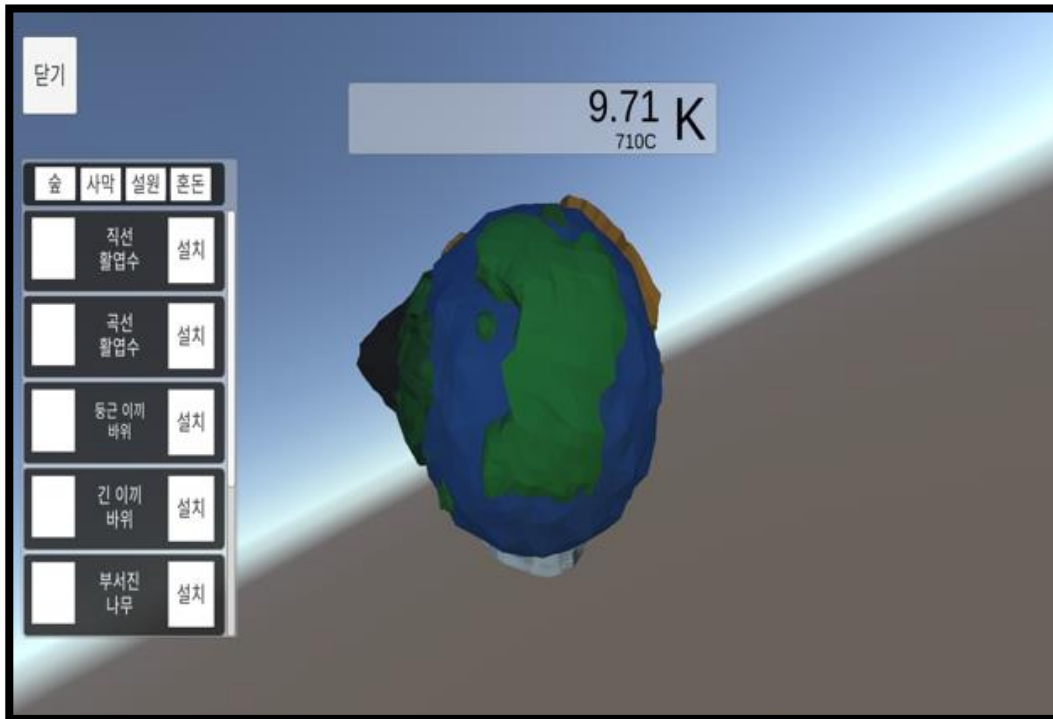
        iDepth++;
        iPos++;

        strSql = "INSERT INTO tblboard(name, title, contents, type, readcount, refer, depth, pos) VALUES ('"
            + TextBox3.Text + "', '"
            + TextBox2.Text + "', '"
            + TextBox4.Text + "', '"
            + DropDownList1.SelectedValue + "', "
            + "0, "
            + iRefer + ", "
            + iDepth + ", "
            + iPos + ")";
        cmd = new SqlCommand(strSql, conn);
        cmd.ExecuteNonQuery();
    }
}
```

sql문법을 통해 db에 접근 후 전역 객체로 생성한 Session을 활용하여 로그인하고 사용자 이름을 Session에 저장하는 방식으로 구현하였습니다. 또한 대학교에서 처음으로 SQL을 사용하여 프로젝트를 개발해서 의의하는데 어려움이 많았습니다.

해당 코드는 view 페이지와 list 페이지에서 오는 매개변수 별로 차이를 두고 개발할 필요가 있었습니다. 이유는 view 페이지는 답변을 작성하기 위해서 요청을 보낸 것이고 list 페이지는 답변이 아닌 일반 게시물을 작성하고자 write 페이지에 요청을 보냈기 때문입니다. 따라서 view 페이지에서 온 요청인지 아니면 list 페이지에서 온 요청인지 확인 후 Request의 값을 조회하여 "Reply"인지 확인하고 처리해주는 방식으로 구현했습니다.

프로젝트 Project eco

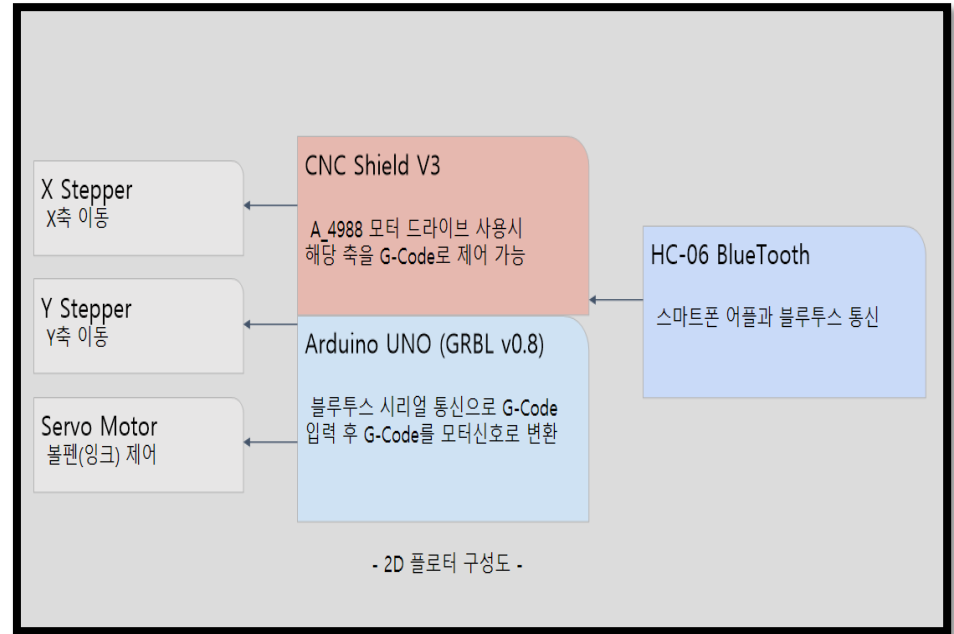
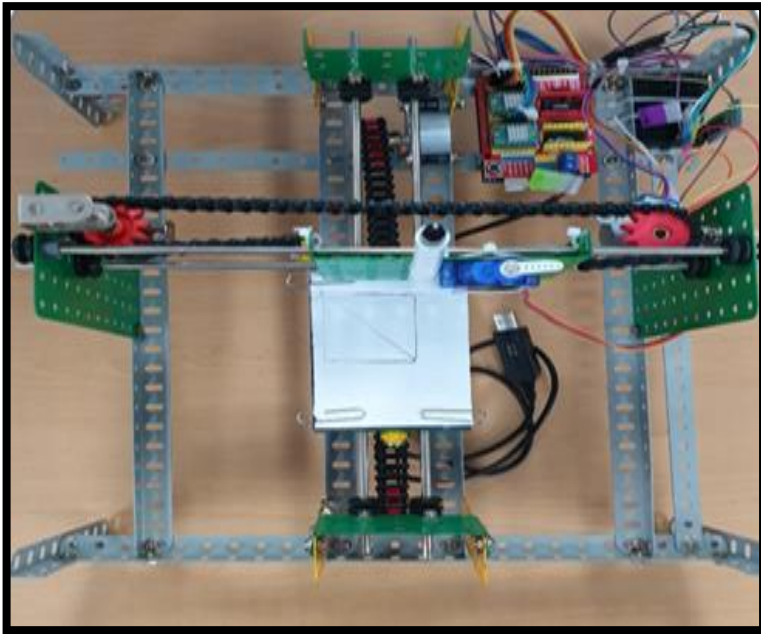


1. 특징

- ① unity 로 개발
- ② 안드로이드 환경을 기준으로 개발
- ③ UI 디자이너로 참여

2. 소스코드 ([google drive](#) 바로가기)

프로젝트 블루투스 통신이 가능한 2차원 플로터



1. 특징

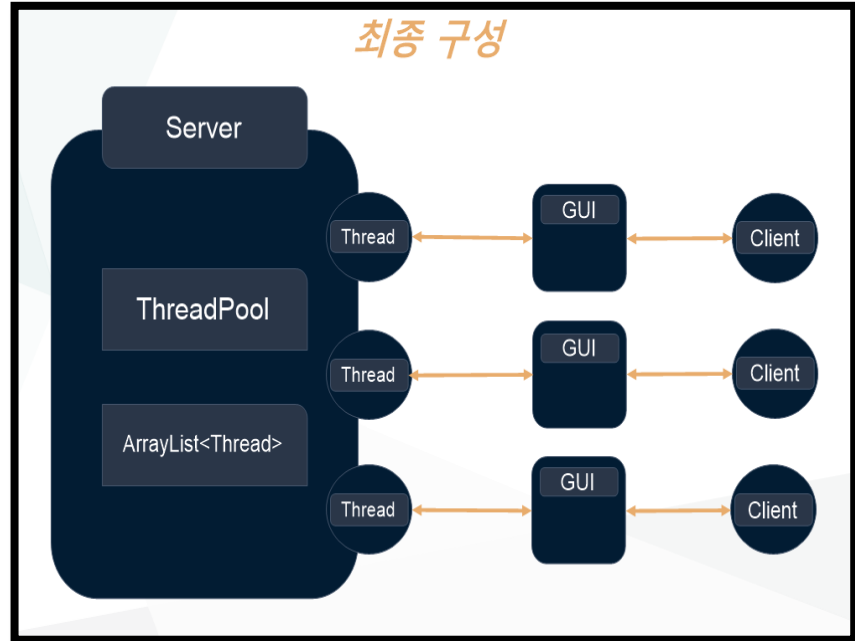
- ① 아두이노를 사용하여 개발
- ② 회로 제작과 하드웨어 제작을 담당
- ③ 기존의 라이브러리를 활용하여 개발
- ④ 팀장으로 참여

2. 시연 영상 ([youtube](#) 바로가기)

우노 보드에 cncшил드를 사용하여 gcode 제어 기능을 보강하였습니다. 블루투스 모듈 장착으로 안드로이드 마켓에 있는 cnc 어플을 통해 사용자가 직접 조작 가능합니다.

프로젝트

소켓 통신을 활용한 채팅 프로그램



1. 특징

- ① threadPool을 활용
- ② 자바 swing 라이브러리
- ③ gui 개발로 참여

2. 링크

- ① 소스코드 : [깃허브](#) 바로가기
- ② 시연영상 : [유튜브](#) 바로가기

스래드 풀을 활용하여 새로운 참여자가 등장하면 Theard를 하나 더 생성하고 관리합니다. 또한 채팅방 gui에도 현재시간을 알려주는 Theard가 하나씩 존재합니다.

프로젝트

소켓 통신을 활용한 채팅 프로그램

```
class Timeth extends JLabel implements Runnable {
    SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss ");
    Calendar time; // = Calendar.getInstance();

    public Timeth() {
        super();
        Thread th = new Thread(this);
        th.start();
    }

    @Override
    public void run() {
        // TODO Auto-generated method stub
        while (true) {
            time = Calendar.getInstance(Locale.KOREA);
            int hour = time.get(Calendar.HOUR_OF_DAY);
            int min = time.get(Calendar.MINUTE);
            int sec = time.get(Calendar.SECOND);
            String format_time1 = format1.format(time.getTime());
            String nowtime = hour + ":" + min + ":" + sec;
            this.setText(format_time1);
            // time.clear(); // 0이 넘으면 000으로 뜸
            try {
                Thread.sleep(10); // 혹시나 생길 과부하 생각해서 휴식 시간 넣어줌 크이 4대보협
            } catch (Exception e) {
                return;
            }
        }
    }
} // 시간 스래드
```

thread를 활용하여 채팅방에 현재시간을 표시해주는 기능을 구현했습니다
컴퓨터의 부하를 줄이기 위해서 10ms 정도의 지연을 주었습니다.
해당 시간은 "Jlabel"로 사용자에게 표시가 되도록 하였습니다.

```
class Help extends JDialog {

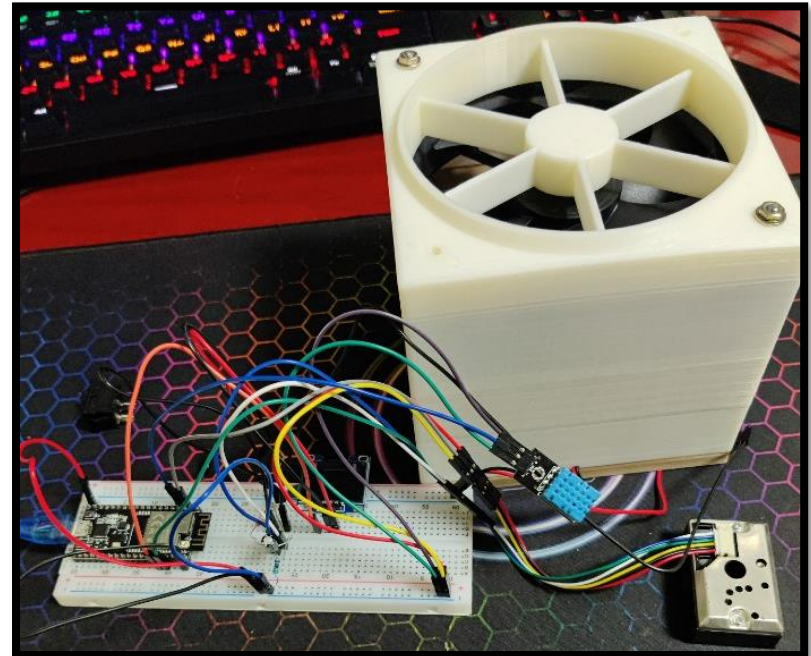
    private final JPanel contentPanel = new JPanel();

    /**
     * Create the dialog.
     */
    public Help() {
        setBounds(100, 100, 450, 600);
        getContentPane().setLayout(new BorderLayout());
        contentPanel.setLayout(null);
        contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
        this.setTitle("도움말");
        this.setResizable(false);
        getContentPane().add(contentPanel, BorderLayout.CENTER);
        {
            JTextArea textArea = new JTextArea();
            textArea.setSize(415, 500);
            textArea.setLocation(5, 5);
            textArea.setEditable(false);
            contentPanel.add(textArea);
            String help = new String("\n [ !명령어 를 입력하면 명령어를 조회할 수 있음. ]\n\n" + " - " + "!현인화 : 현재 방에 있는 인원 수와 닉네임 조회\n\n"
                + " - !금지어 : 금지어 목록 조회 (사용하면 (금지어) 처리)\n\n" + " - to(컨속말) : to 닉네임 대화내용으로 대상에게만 채팅");
            textArea.setText(help);
        }
        {
            JPanel buttonPane = new JPanel();
            buttonPane.setLayout(new FlowLayout(FlowLayout.RIGHT));
            getContentPane().add(buttonPane, BorderLayout.SOUTH);
            {
                JButton cancelButton = new JButton("확인");
                cancelButton.setActionCommand("cancel");
                cancelButton.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {
                        // TODO Auto-generated method stub
                        dispose();
                    }
                });
            }
            buttonPane.add(cancelButton);
        }
    }
}
```

JDialog를 상속받아서 사용자에게 도움말 기능을 제공해주는
Help클래스 클래스 입니다.

프로젝트 IoT 공기청정기 시스템











1. 특징

- ① db(mysql)를 중심으로 통신
- ② nodejs 기반의 웹서버
- ③ 반응형 웹페이지
- ④ 아두이노를 활용한 wifi통신
- ⑤ 공공 api를 받아와 사용자에게 제공
- ⑥ 팀장으로 참여

2. 링크 (소스코드 : [깃허브](#) 바로가기)

프로젝트 IoT 공기청정기 시스템

	air_database
	obid
	observation_time
	misae
	temperature
	humidity
	is_turned_on
	machine_id

```

1 global__approot = __dirname;
2 const express = require('express');
3 const passport = require('passport');
4 const cors = require('cors');
5
6 const { NODE_APP_INSTANCE, NODE_ENV } = process.env;
7
8 // 서버 초기화
9 const app = express();
10 app.disable('powered-by'); // x-powered-by 제거
11 app.use(cors()); // CORS 설정
12 app.options('*', cors()); // CORS Pre-flight 응답
13 app.use(express.urlencoded({ extended: true, limit: '100mb' })); // nginx도 같은 값으로
14 app.use(express.json({ limit: '100mb' }));
15
16 const db_config = require(__dirname + '/db.js');
17 const conn = db_config.init();
18 db_config.connect(conn);
19 app.get('/', (req, res) => {
20   const ver1 = '1';
21   res.send({ API: '안녕하세요' });
22 })
23
24 //json에 ON/OFF, 미세먼지농도, 온도, 습도
25
26 app.get('/dashboard/info', (req, res) => {
27
28   const response = {
29     state: 1,
30     query: null,
31     msg: 'Successful'
32   };
33   var sql = 'SELECT AIR_database.misae FROM AIR_database WHERE obid=1'; //가짜데이터
34   //console.log(sql);
35   conn.query(sql, function (err, results, field) {
36     let json1 = results[0].misae;
37     response.query = json1 + "%"; // 결과 가져오기
38     return res.status(200).json(response);
39   });
40 }) //가짜데이터 데이터
41
42 app.get('/dashboard/info2', (req, res) => {
43   //유관공인 미세먼지 (pm2.5, 온도, 습도) 데이터
44   //가짜데이터
45   app.get('/dashboard/info3', (req, res) => {
46     //가짜데이터
47   });
48 }) //가짜데이터
49
50 app.get('/ui/data', (req, res) => {
51   //UPDATE AIR_database SET observation_time=TIMESTAMP '2020-12-18 19:18:25', misae=20, temperature=22.5, humidity=10, is_turned_on=0 WHERE obid = 1
52   app.get('/user/list', async (req, res) => {
53     //가짜데이터
54   });
55
56 const server = app.listen(9595, '0.0.0.0', () => {
57   console.log(
58     `>>> API SERVER ${Number.isNaN(Number(NODE_APP_INSTANCE)) ? 'local' : NODE_APP_INSTANCE} STARTED ON http://0.0.0.0:9595 AS ${NODE_ENV} [! 'NO_NODE_ENV']`,
59   );
60 });
61
62 // 서버 종료
63 process.on('SIGINT', () => {
64   server.close();
65 });
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

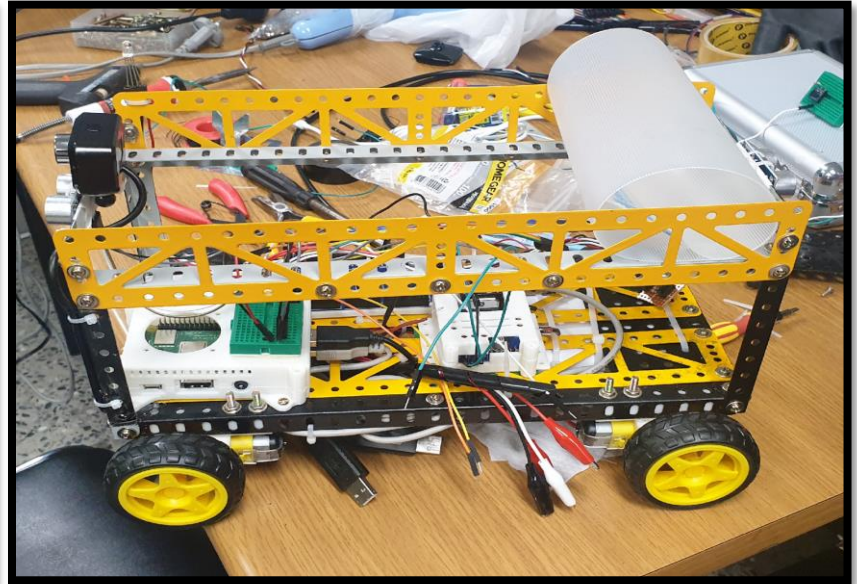
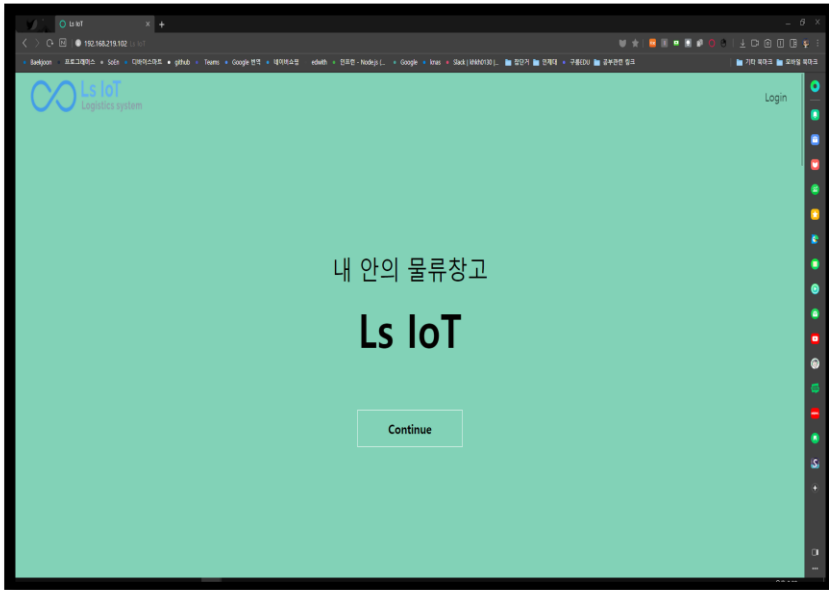
```

1) db 구조

REST_API를 활용하여 web 페이지에서 get 요청하면DB에서 자료를 수집하여 전송하는 방식으로 구현했습니다. 첫 node.js 개발이라서 라우팅 파일은 따로 만들어 사용하지 않았습니다.

수집한 기기로부터데이터를 받아 web 페이지에 표시하기 위해서 air_database 테이블을 제작하였습니다. 공기청정기에서 직접 DB를 조회 할 수 있도록 MYSQL 아이피 개방을 하여 외부 IP로 접근이 가능하도록 구성하였습니다.

프로젝트 IoT 무인 물류창고 시스템



1. 특징

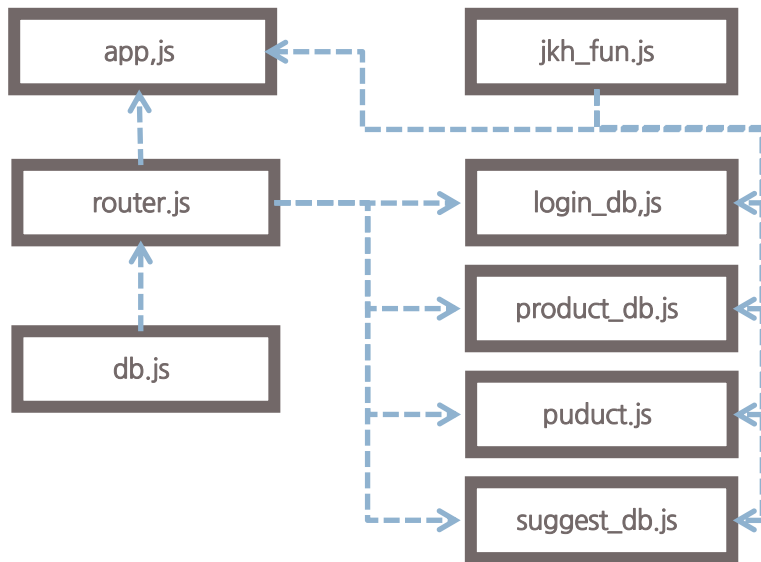
- ① DB(mysql)를 중심으로 통신
- ② Node.js 기반의 웹서버
- ③ 반응형 웹페이지
- ④ 라즈베리를 이용한 wifi통신
- ⑤ 무인 배송 로봇
- ⑥ 팀장으로 참여

2. 링크

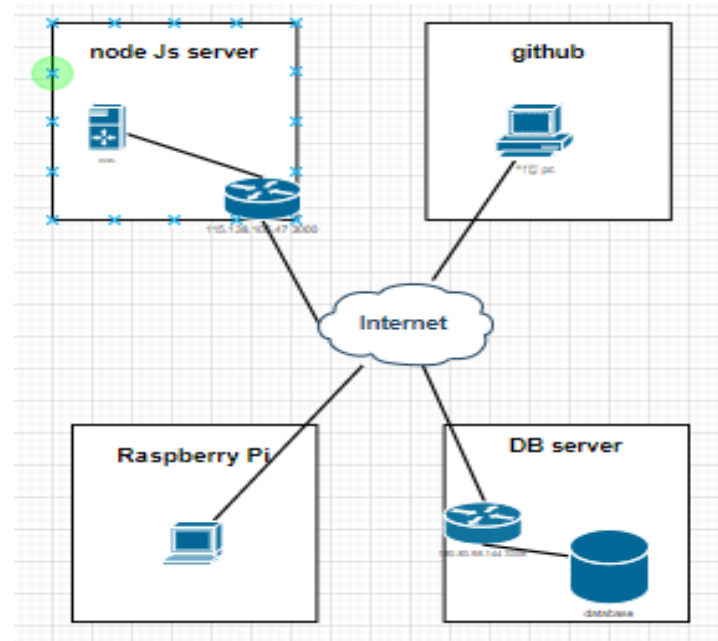
- ① 소스코드 : [깃허브](#) 바로가기

프로젝트 IoT 무인 물류창고 시스템

파일 구조



서비스 네트워크 구조



node.js는 개인 나스에서 nodemon을 활용하여 호스팅하였습니다. 해당 이유는 다양한 서버 환경에서 구동해보고 싶었고 온종일 컴퓨터를 켜두기 힘들 거 같아서 개인 NAS를 통해 호스팅하였습니다. 소스 코드는 github를 통해 형상관리를 진행하였습니다. 로봇의 라즈베리에서 직접적으로 DB에 접근해서 응답속도를 빠르게 하려고 했으나 외부 IP 접근 허용을 통해 DB의 보안이 많이 떨어지는 결과를 초래하게 되었습니다.

프로젝트 IoT 무인 물류창고 시스템

```

/*****로그인 설정*****/
/*****로그인 설정*****/
/*****로그인 설정*****/
router.post('/login', (req, res) => {
  var req_data = {
    email: req.body.id,
    pw: req.body.password
  }
  jkh_db_config.userSelect_post(req, res, conn, req_data);
});
//로그인 - 세션등록
router.get('/login', (req, res) => {
  try{
    var req_data = {
      name: req.session.user.name,
      email: req.session.user.email,
      pw: req.session.user.password
    }
  } catch(e){
    jkh_db_config.userSelect_get(req, res, conn, req_data);
  }
});
//로그인 - 비밀번호 추출
router.post('/logout', (req, res) => {
  jkh_db_config.userdisable(req, res, conn);
});
//로그아웃
router.post('/regi', (req, res) => {
  var req_data = {
    email: req.body.email,
    pw: req.body.password,
    name: req.body.username,
  }
  jkh_db_config.userCreate(req, res, conn, req_data);
});
//회원 가입
router.post('/repw', (req, res) => {
  var email = req.body.email;
  jkh_db_config.userchage(req, res, conn, email);
});
//비밀번호 찾기

```

라우팅 모듈로 로그인 처리 부분 중 클라이언트로부터 파라미터만 받고 db CURD는 따로 함수로 생성하여 router.js에서 인자로 전달하는 방식으로 구현했습니다. 해당 방식 덕분에 팀원들과 개발하면서 업무를 분배하기 편리하고 한쪽 기능에서 버그가 발생해도 각자 서로 다른 기능을 개발할 수 있었습니다.

```

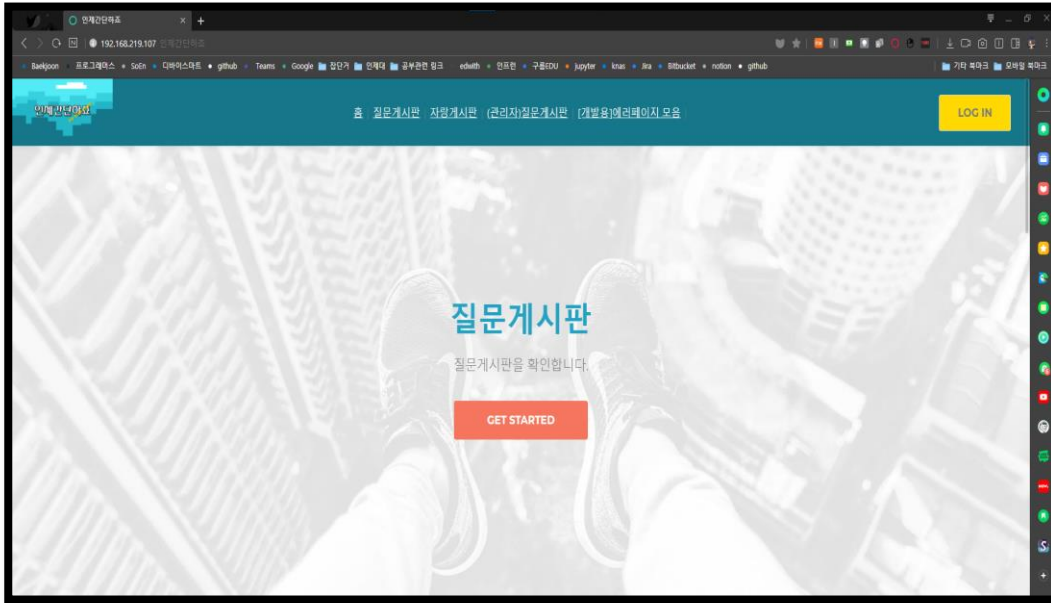
userSelect_post: (req, res, conn, req_data) => {
  //console.log('이메일 : ' + email); //확인용
  let sql = 'SELECT * FROM user_database WHERE user_email = ? AND user_password = ?'; //가져오기
  var session = req.session;

  conn.query(sql, [req_data.email, req_data.pw], function (err, results) {
    if (err) {
      console.log(`${jkh_fun.date_time()}: error => ${err}`);
    }
    else {
      try {
        if (jkh_fun.isEmpty(results)) {
          console.log(`${jkh_fun.date_time()}: No request data`);
          response.query = false; //이름없음
          response.msg = 'failed';
          response.state = 0;
          return res.status(200).json(JSON.stringify(response));
        } //조회 실패
        else {
          console.log(`${jkh_fun.date_time()}: select is data => ${results[0].user_name}`);
          //db_data.db_name = results[0].user_name;
          response.query = results[0].user_name; // db_data.db_name;
          response.msg = 'Successful';
          session.user = {
            name: response.query, //results[0].user_name; //results[0];
            password: req_data.pw,
            email: req_data.email
          };
          response.state = 1;
          console.log(response); //결과 출력
          return res.status(200).json(JSON.stringify(response));
        } //세션에다가 결과 저장해야함
      } catch (e) {
        console.log(e + ' // db조회중 오류 발생');
      }
    }
  });
}
//세션등록

```

router.js 모듈로부터 전달받은 파라미터를 이용하여 처리하는 모듈로서 모든 조건절에 "console.log"를 넣어 오류 발생 위치를 쉽게 파악할 수 있었고 유연한 대처를 가능하게 만들었습니다.

프로젝트 Project eco



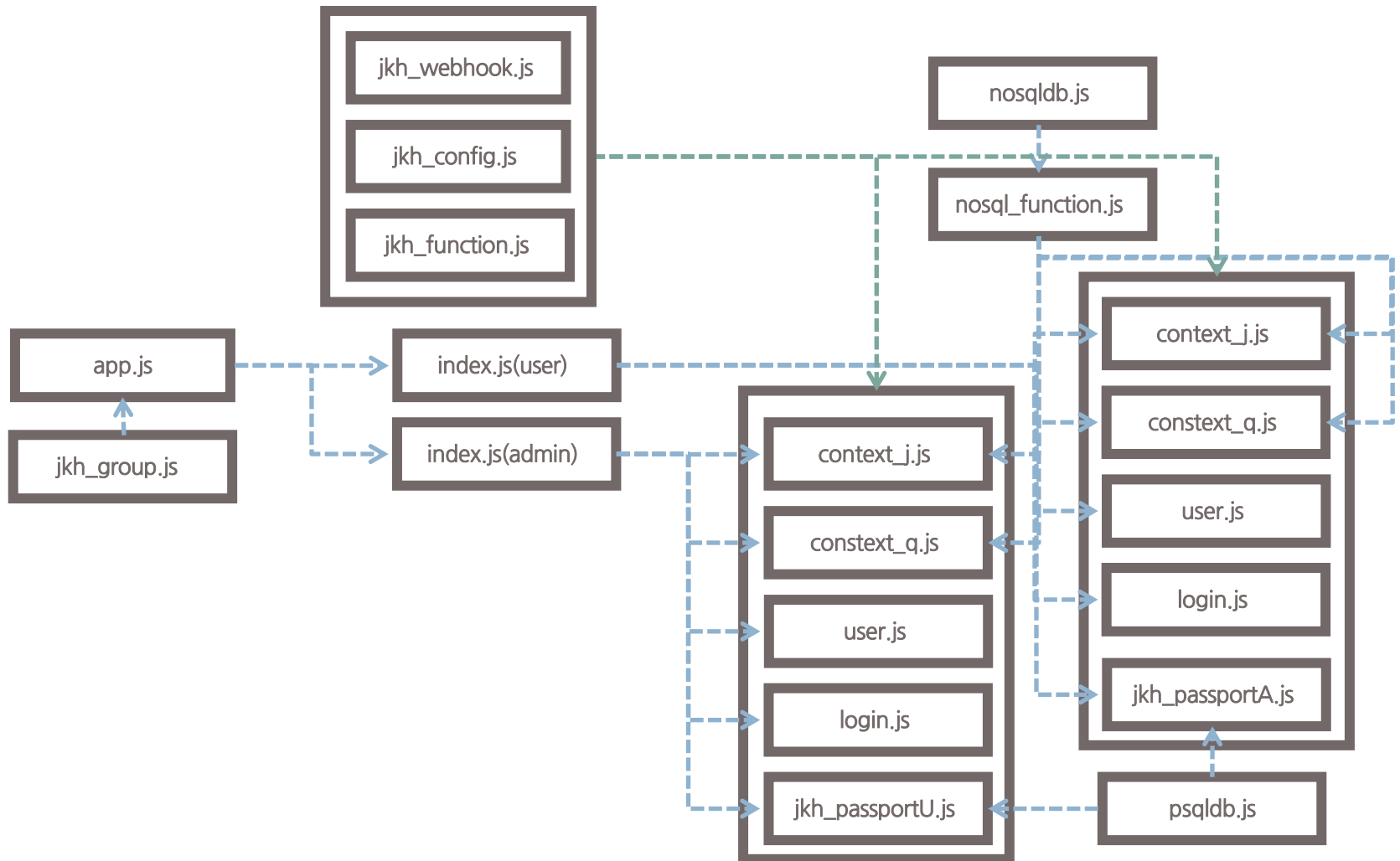
1. 특징

- ① postgresql 사용
- ② mongodb 사용
- ③ nodejs 기반의 웹서버
- ④ 반응형 웹페이지
- ⑤ 팀장으로 참여
- ⑥ api 서버와 웹서버를 분리하여 개발
- ⑦ 현재 스케일 아웃 작업 진행중
- ⑧ jira를 통해 남은 이슈를 점검한 결과 60%정도 완성됨

2. 링크 (소스코드 : [깃허브](#) 바로가기)

프로젝트 Project eco

파일 구조

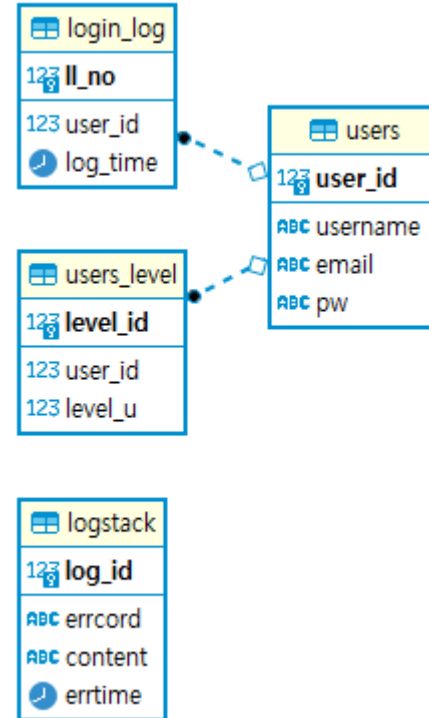


프로젝트 Project eco

```

1 const express = require('express');
2 const path = require('path');
3 const morgan = require('morgan');
4 const passport = require('passport');
5 const cookieParser = require('cookie-parser'); //쿠키파싱
6 const cors = require('cors');
7 require('./api/v1/function/jkh_group.js');
8 const jkh_f = require('./api/v1/function/jkh_function');
9 const jkh_c = require('./api/v1/function/jkh_config');
10 const users = require('./api/v1/user');
11 const admin = require('./api/v1/admin');
12 const ipfilter = require('express-ipfilter').Ipfilter
13 const app = express();
14
15 var iplist = [];
16 const iplist_maker = (list) => {
17   //파일로 되어 있는 데이터를 실시간으로 받아서 목록 구성
18   //파일로 iplist에 추가하는 함수
19   const data = jkh_f.file_r(path.join(__dirname, 'api/v1/function/'), 'config');
20   if (jkh_f.isEmpty(data)) { //null이면
21     return iplist;
22   } else {
23     var ip = data.split(' '); //save_data lode
24     iplist.push(ip); //save
25   }
26   return iplist;
27 }
28 app.use(ipfilter(iplist_maker())); //모든 deny //https://www.npmjs.com/package/express-ipfilter
29 //ip 주소(200.0.0.0) 접속 차단
30 //app.use(mongodb.init());
31 app.disable('x-powered-by'); // x-powered-by 제거
32 app.use(cors({
33   exposedHeaders: ['Content-Disposition'], // 디스포지션 시 헤더를 노출
34 })); // CORS 설정
35 app.options('*', cors()); // CORS Pre-flight
36 app.use(express.json());
37 app.use(express.urlencoded({ extends: true }));
38 app.use(cookieParser());
39 app.use(passport.initialize()); //passport 설정
40
41 ///// ip 차단 코드
42 app.get('/', (req, res) => {
43   var rrq_ip = jkh_f.ip_denying(req); //요청을 받아서 분석하여 차단여부 판단
44   if (rrq_ip.state == 1) {
45     iplist.push(rrq_ip.ip); //차단리스트 등록
46     console.log('ip 차단 : $(rrq_ip.ip)');
47     jkh_f.webhook('warn', '$(rrq_ip) country '$(rrq_ip.country)' api '/' enter and denying');
48     let str = rrq_ip.ip + ' ';
49     jkh_f.file_s('./api/v1/function/','config', str); //파일로 저장하여 관리하기 용이 // 전역으로 관리하면 좋을 것 같아 추가
50   }
51   else{
52     jkh_f.webhook('success', '$(rrq_ip) api '/' enter');
53     return res.send(str);
54   }
55 })
56
57 /////
58 app.use(morgan('combined', { stream: jkh_f.logstream })); //로그파일을 관리 할 수 있게
59
60 app.use('/api/v1/user', users); //사용자
61 app.use('/api/v1/admin', admin); //관리자
62 app.listen(jkh_app.port, jkh_app.host, () => {
63   let str = 'http://$(jkh_app.host):$(jkh_app.port)'; //api 접근 url
64   console.log('$(jkh_f.date_time())start server');
65   jkh_f.webhook('info', '$(jkh_f.date_time())node.js server starting!');
66   jkh_f.webhook('info', str);
67 });
68

```



서버 실행할 때 중국의 접근이 많아 한국 IP가 아니면 차단하는 소스코드를 추가했습니다. 또한 패스포트 모듈을 활용하여 인증을 구현하기 위해 미들웨어로 설정하였습니다. 또한 일반 사용자와 관리자 그룹별로 따로 관리하기 위해서 파일을 분리하여 제작하였습니다.

postgresql를 활용하여 로그인 로그를 저장하는 테이블과 사용자의 권한을 판단하는 테이블을 따로 만들어 구성하였습니다. 사용자 정보 저장과 error 발생 시 저장하는 용도로 사용하고 게시판 데이터는 mongodb를 사용하여 개발할 예정입니다.

프로젝트

뉴스 키워드 분석



1. 특징

- ① python 활용함
- ② 전기관련 기사 1만5천여개
- ③ 뉴스 기사를 크롤링해서 Kmeans 알고리즘 적용
- ④ 추출한 키워드를 바탕으로 wordcloud로 표현

2. 링크

- ① 소스코드 : [깃허브](#) 바로가기

대외활동

2020년 대학생 온라인 교육기부 동아리(하반기)

1. 참여 동기

학생들에게 “컴퓨터를 이용하여 게임뿐만 아니라 다양한 활동들을 해볼 수가 있다.”라는 사실을 알려주고 싶은 마음에 참여하였습니다.

2. 특징

- ① 동아리 팀장으로 진행
- ② 촬영, 기획, 편집, 키트제작을 담당
- ③ 현동지역아동센터 학생들 대상 총 20명에게 교육기부활동 진행
- ③ 초등학생들에게 아두이노와 엔트리를 활용한 it 교육을 비대면으로 진행

3. 링크

- ① 유튜브 :
 - ① [영상 1](#) 바로가기
 - ② [영상 2](#) 바로가기
 - ③ [영상 3](#) 바로가기



대외활동

2021년 대학생 교육기부 여름방학 '온라인 쏙쏙캠프'

1. 참여 동기

작년 수혜 학생들 반응이 좋아서 조금 더 많은 학생에게 콘텐츠를 제공해주고 싶어서 한 번 더 활동하였습니다.

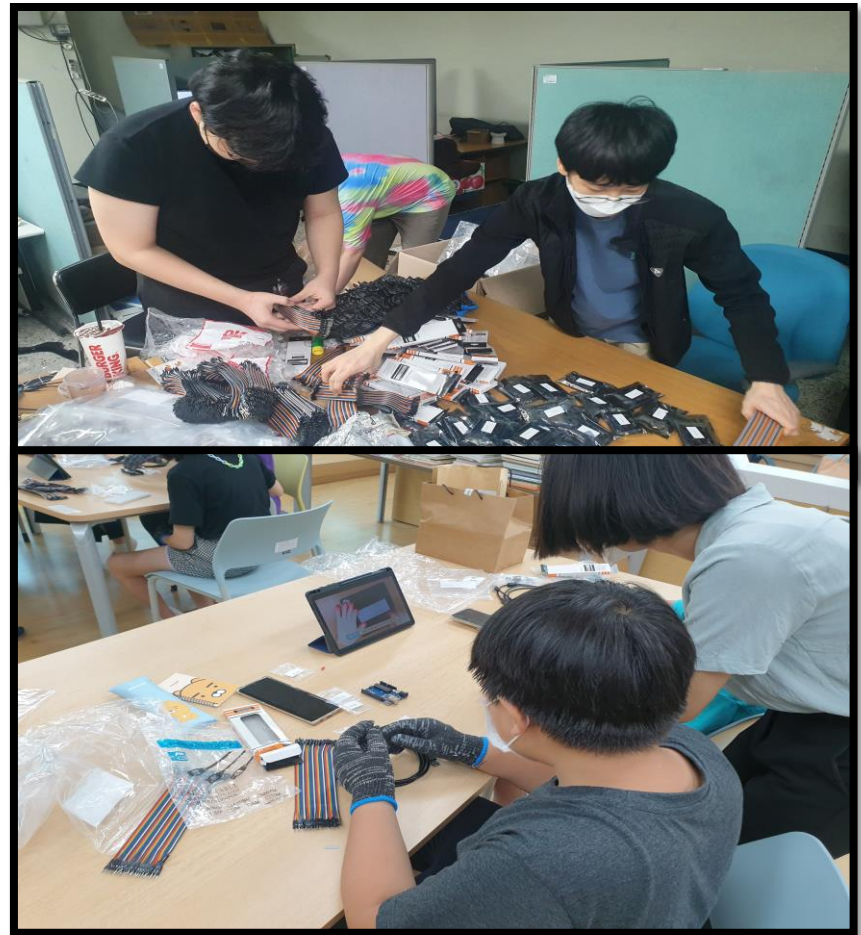
2. 특징

- ① 동아리 팀장으로 진행
- ② 촬영, 기획, 키트제작, 실시간 강의를 담당
- ③ 광주서초등학교 학생들 대상 총 20명에게 교육기부활동 진행
- ③ 초등학생들에게 아두이노와 엔트리를 활용한 it 교육을 비대면 실시간 강의를 진행

3. 링크

① 유튜브 :

- ① [실시간 강의 영상](#) 바로가기
- ② [영상1](#) 바로가기
- ③ [영상2](#) 바로가기
- ④ [영상3](#) 바로가기



대외활동

학술 동아리 세미나 강의 진행

1. 참여 동기

학교 후배들이 git과 github에 대해서 모르는 학생들이 많아 깃 사용법에 대해서 강연을 한적이 있습니다.

2. 특징

- ① 2021년도 3월 24~ 4월 7일 주 1회 진행
- ② 학과 신입생과 깃에 대해 잘 모르는 후배들 대상으로 진행
- ③ git과 github 자주 사용하는 툴을 소개

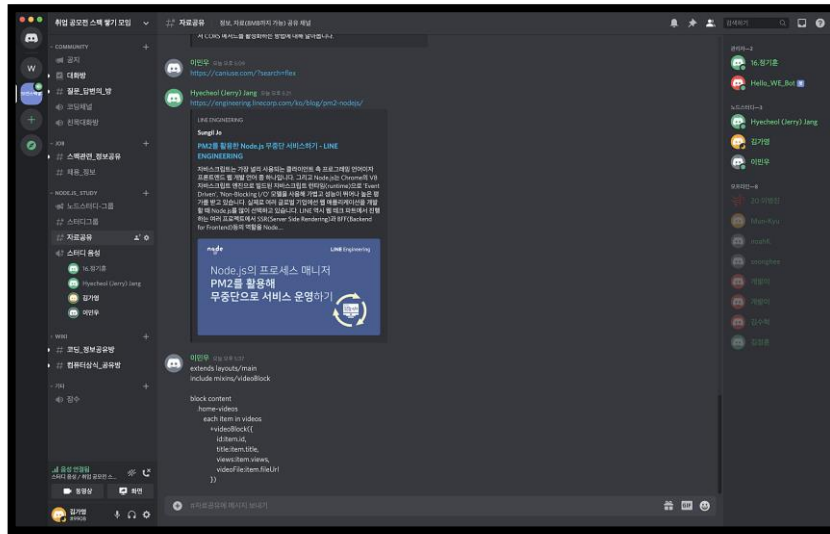
3. 링크

- ① 구글 드라이브:
 - ① 강의 자료 바로가기
 - ② 강의 자료 바로가기



대외활동

디스코드를 활용한 비대면 활동



2020년 02월부터 개발 공부하면서 참고한 기술 블로그 및 자료 공유용 디스코드(discord) 채널 운영하고 있습니다.
해당 채널은 정보 공유와 더불어 부산지역 node.js 스터디 활동을 [팀장](#)으로 하면서 비대면 모임의 장으로 활용하였습니다.

수료증

제약21-07-012호

수료증

성명: 정기훈
 생년월일: 1997년 1월 30일
 소속: 인제대학교 컴퓨터공학부
 교육명: 제1회 동남권 바이오올러스터 AI 교육

위 사람은
 대웅제약이 경남김해장소특구
 빅드림웨이오사업단과 함께 주관하는
 '2021년도 제1회 동남권 바이오올러스터 AI'
 교육을 성실히 수료하였으므로
 이에 이 증서를 수여합니다.

2021년 7월 21일

주식회사 대웅제약
 대표이사 윤재준, 전승호

제2021-교육30-789832

수료증

교육구분: 은관인교육
 교육과정: (2021) 4차 산업혁명 시대, 알아야 하고 계획야하는
 직업세계
 수강신청일: 2021년 03월 30일
 수료일: 2021년 06월 20일
 교육시간: 5시간
 소속: 인제대학교
 성명: 정기훈

위하에서는 상기와 같이 한국고용정보원
 사이버정보교육센터에서 실시한 교육과정을
 수료하였으므로 이 증서를 수여합니다.

2021년 06월 20일

한국고용정보원

제2021-AI-253호

수료증

교육과정: 비정형 의료/건강 데이터를 활용한 기계학습 이론과 실습
 교육기간: 2021.07.15.~08.31.(160h)
 생년월일: 1997.01.30
 교육생명: 정기훈

위 사람은 2021년 ICT이노베이션스퀘어 확산사업의
 「인공지능(AI) 교육과정(160시간)」을 위와 같이
 수료하였으므로 이 증서를 드립니다.

2021.08.31

(재)경남테크노파크

제 2020 - 10144호

수료증

소속: 컴퓨터공학부
 학번: 20163164
 성명: 정기훈

위 학생은 인제대학교 취업진로과에서 주관한 [선후배
 와의 멘토 특강 프로그램(컴퓨터공학부)]를(을) 아래와
 같이 이수하였으므로 이 수료증을 수여함.

• 프로그램: 선후배와의 멘토 특강 프로그램(컴퓨터공학부)
 • 활동기간: 2020-11-05 ~ 2020-11-05

2021년 06월 20일

학생취업처장

발급번호: 제 2021A007088 호

교육이수확인서

인계사항	성명	정기훈	주민등록번호	970130-*****	
	소속	인제대학교 컴퓨터공학부	소속지역		
	기간	2021.05.14 ~ 2021.06.11	총일수		
	1일시간		총시간	2	
특정사항	장소	교육부/한국과학창의재단			
	내용	○ 교육명: 2021년 대학생 교육기부자 필수 역량교육(목욕캠프) ○ 주최/주관: 교육부/한국과학창의재단 ○ 이수내용: 실시간 온라인 연수 이수(총 1일, 각 2시간) 1) 비대면 교육기부 방법			
	담당자	성명	이동은	연락처	02-059-3973
	위와 같이 확인합니다.				

2021년 08월 11일

한국과학창의재단이사장

발급번호: 2021782145809_03080603

자원봉사활동 확인서

성명: 정기훈
 주민등록번호: 970130-*****
 주소: 50930 경상남도 김해시 분성로512번길 51 동주빌리지 101호

• 자원봉사 활동기간: 2020-09-25 ~ 2020-11-09
 • 자원봉사 활동시간: 총 6시간 0분
 • 자원봉사 활동횟수: 총 3회
 • 봉사활동 내용: 대학생 온라인 교육기부, 영상 콘텐츠 제작 및 멘토링 자원 봉사 2건

위와 같이 자원봉사 활동에 참여하였음을 확인함

2021년 07월 08일

(재)한국중앙자원봉사센터장

◆ 본 증명서는 인터넷으로 발급되었으며, 자원봉사 포털시스템(www.1365.go.kr)의
 확인서 조회 메뉴를 통해 문서발급 번호 입력으로 내용의 위변조 여부를 확인해 주십시오.
 다만 문서 확인 번호를 통한 확인은 발급일로부터 90일까지 가능합니다.
 * 발급확인서에 표시한 유관기관의 자원봉사 실적은 자원봉사 포털 시스템과 연계됨을 통해
 확인할 실적 정보입니다.

증명서

제 501709 호

장학금수혜증명서


학번: 20163164
성명: 정기훈
대학: 소프트웨어대학
학과(부)·학년: 컴퓨터공학부 전문컴퓨터공학 4학년

장학금수혜내역

주최 기관	장학명	장학금액(원)		
		입학금	등록금	합계
2020년				
2020년	학업장학(보조나눔 가계곤란지원)	0	100,000	100,000
2020년	학업장학(보조나눔 가계곤란지원)	0	200,000	200,000
2021년	장수장학(장수세대지원)	0	200,000	200,000
2021년	기타근로(컴퓨터공학부)	0	800,000	800,000
2021년	다녀나갈학금	0	2,250,000	2,250,000
2021년	전체인원(장학대상)	0	400,000	400,000
2021년	장학대상(소속분위)	0	200,000	200,000
2021년	국가장학	0	600,000	600,000
- 이외사항 -				

상기와 같이 장학금수혜 내역을 확인합니다.

2021년 9월 29일

인제대학교 

* 이 증명서는 장학금수혜를 위해 학위를 수여받은 학생에게만 효력이 있습니다.

fast campus Life Changing Education fastcampus.kr

패스트캠퍼스 교육과정 수료 증명서

문서번호: FC-210823-CA002
발행일자: 2021. 08. 23


성명: 정기훈

수료정보

수료 과정	수료일
알고리즘 / 기술면접 완전 정복 끝까지 OnLine	2021. 08. 22

위 사항은 패스트캠퍼스 학습 과정에서
성실하게 수료하였음을 증명합니다.

2021. 08. 23

(주)데이원컴퍼니 

제 90-008043 호

자격취득확인서

성명: 정기훈
생년월일: 1997년 01월 30일

종목: 인공지능응용 2급

자격번호: 14-93-009137

2014년 02월 21일 국가기술자격검정시험 합격의 자격을 취득하였음을 확인합니다.

2021년 09월 29일

대한상공회의소 회장 

한국상공회의소 2021년 9월 29일

kpc 한국생산성본부

자격취득내역서

성명: 정기훈
생년월일: 1997년 01월 30일

자격종목: 정보기술자격(ITQ)

등록번호: AG01-2012115-007159

취득자격	등록일
아래한글 A등급	2012-11-01
한글역설 A등급	2012-12-27
한글퍼워포인트 A등급	2012-09-13

위와 같이 취득하였음을 확인합니다.

2021. 09. 29

한국생산성본부 회장