Web Service Development	Student number	21600193
Homework 7 – Project Final Report	Name	Kim, Hyo Rim

1. Service Description

1) Theme

Big Data Analyzer

2) Reason

In the case of Daba bases, data can be analyzed quickly and easily through queries. However, it is difficult to draw a graph. Several analysis programs such as 'Weka' can be visualized at the same time as analysis, but the program uptime is long.

So I decided to provide a service that can quickly and easily draw graphs that analyze Big Data.

3) Service

(1) Data Analysis

① It receives the data file from the user and provides db.

② It provides data analysis and data graph easily and quickly through mysql query.

③ Count: It sends a count query after allowing or disallowing duplication of data.

④ Bar Chart: Using a JavaScript canvas, graph the data obtained from the query.

(2) Discussions between users

① It enables discussions among users through bulletin boards.

② Enable comment function to easily exchange opinions for each article.

(3) Request Data Analysis

① A user can request a DB analysis through a bulletin board which can be viewed only by the user and the manager.

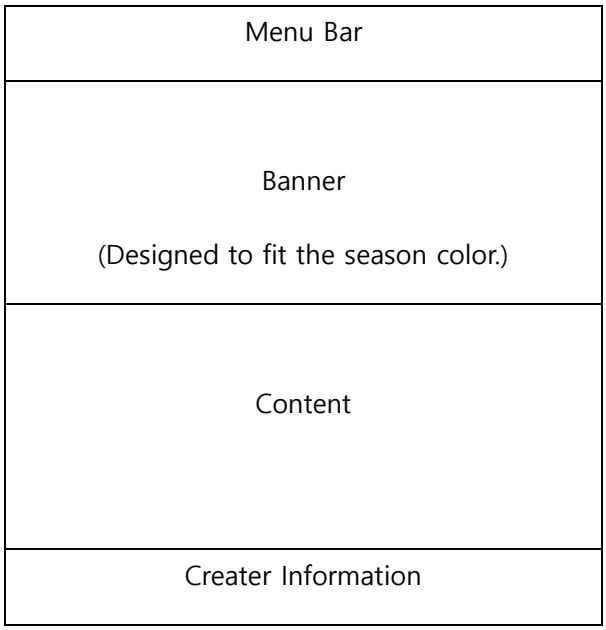
2. Overall System Design

1) Designs

(1) Layout

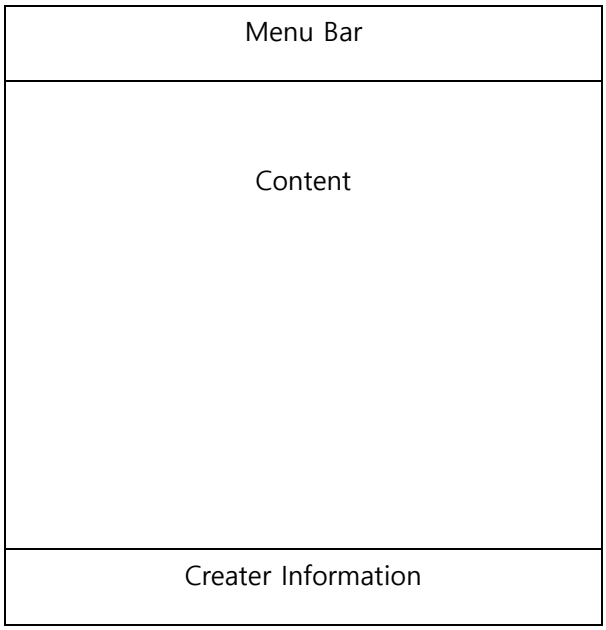
It will be designed to be as simple and clear as possible.

① Except bulletin board and login page.



<Figure 1> page design 1

② Bulletin Board and login page



<Figure 2> page design 2

③ Code

```
<!DOCTYPE html>
<html>
<head>
        <title>Welcome!</title>
        <link rel="stylesheet" type="text/css" href="./css/english.css">
```

```
</head>
<body>
        <div id="nav">
                <!--Top menu-->
        </div>
        <div>
                 <!-- contents   -->
        </div>
        <div id="Hyorm">
                 <!-- Creator Information -->
        </div>
</body>
</html>
```

(2) Interaction

① app: analyze.php, data.php, info.php, main.php, printData.php, quest.php, read.php,

showData.php, world.php, write.php

- signUp.html & write.php

Input User Data

```
<form action="../process/register.php" method="POST">

        <div class="container">

                <!– input Data -->

        </div>

</form>
```

write Article

```
<form id="article" action="../process/write_process.php" method="POST”>

        <!– Write Article -->

</form>
```

- main.php & info.php

Banner

```
<a id="big" href="../app/main.php">
```

```
        <img src = "../img/banner.png" width="100%" height="35%">
</a>
```

Mouseover Menu

```
<a href="../app/main.php">
        <img src = "../img/trevel.jpg"        onmouseover="this.src='../img/trevel_hover.jpg'"
        onmouseout="this.src='../img/trevel.jpg'" width="48%"         height="23%">
</a>
```

- data.php & showData.php

multipart/form-data

```
<div id="img">
        <p id="name">Welcome to Big Data World!</p>
        <form action="../process/fileI.php" method="POST"
        enctype="multipart/form-data">
        <input class="browse" type="file" name="userfile">
</div>
<div>
        <textarea name='val' rows="10" cols="40"></textarea>
        <input class="submitbtn" type="submit" value="INPUT">
        </form>
</div>
```

- analyze.php

Data Base information

```
$sql = "SELECT COLUMN_name, COLUMN_TYPE  FROM INFORMATION_SCHEMA.COLUMNS WHERE
TABLE_NAME='".$id."'";
```

Count

```
$que = "SELECT count(distinct ".$_POST['name'].") FROM ".$id;
```

```
$que = "SELECT count(".$_POST['name'].") FROM ".$id;
```

Bar Chart - Query

```
$sql = "select ";

if($rowPro=="count")$sql .="count(".$row."), ";

else $sql .=""".$row.", ";

if($colPro=="count"){

            $sql .="count( ";

            if($colDed=="dedupNo")$sql .="distinct ".$col.") ";

            else $sql .= $col.") ";

}

else {

            $sql .= $col." ";

}

$sql .="from ".$id." ";

if($rowDed=="dedupNo")$sql .="group by ".$row." ";
```

Bar Chart – Draw

```
<canvas id="canvas" width="800" height="520"></canvas>

        <script type="text/javascript">

            var elem = document.getElementById('canvas');

            var context = elem.getContext('2d');

            context.font = '20px sans-serif'; context.fillStyle = '#000';

            context.lineWidth = 1;

           getRod();

            function drawLine(x1, y1, x2, y2){

                context.moveTo(x1, y1); context.lineTo(x2, y2);

            }
```

```
    function getRod(){
        context.clearRect(0, 0, canvas.width, canvas.height);
        var selectX = "<?=$row?>"; var selectY = "<?=$col?>";
        if(selectX=="Empty"){
            context.fillText('No Element', 350, 100);
            context.closePath(); context.stroke();
        }else{
            var rowArr = <?=json_encode($rowArr)?>;
            var colArr = <?=json_encode($colArr)?>;
            var arrNum = '<?=$arrNum?>'; var colNum = '<?=$colNum?>';
            var sql = <?=json_encode($sql)?>
            context.fillText(sql, 200, 30);
            drawLine(80, 450, 80, 50); drawLine(80, 450, 640, 450);
            context.fillText(selectX, 650, 450); context.fillText(selectY, 70, 30);
            context.closePath(); context.stroke();
            context.fillStyle= '#ab162a'; context.lineWidth = 560/arrNum/5;
            if(560/arrNum/5<1)context.lineWidth = 1;
            context.beginPath();
            var i = 0;
            for(i = 0; i < arrNum; i++){
              drawLine(120+560/arrNum*i, 450, 120+560/arrNum*i, 400-    400*colArr[i]/colNum);
            }
            context.closePath(); context.stroke();
        }
    }
</script>
```

- printData.php

Query
$sql = "select count(*) FROM information_schema.columns where table_name='".$id."'"; $sql = "SELECT * FROM ".$id."";
Table

function expandTable($id){
 	
html = $id."</td><td>";
 	
return $html;
 	
}
 	
while ($row = mysqli_fetch_row($result)) {
 	
echo "<tr><td>";
 	
for($i = 0; $i<$num ; $i++){
 	
echo expandTable($row[$i]);
 	
}
 	
}

② process: authentication.php, db_connect.php, filel.php, fileO.php, login.php, logout.php,

logout.php, register.php, wirte_answer.php, write_process.php

- filel.php & fileO.php

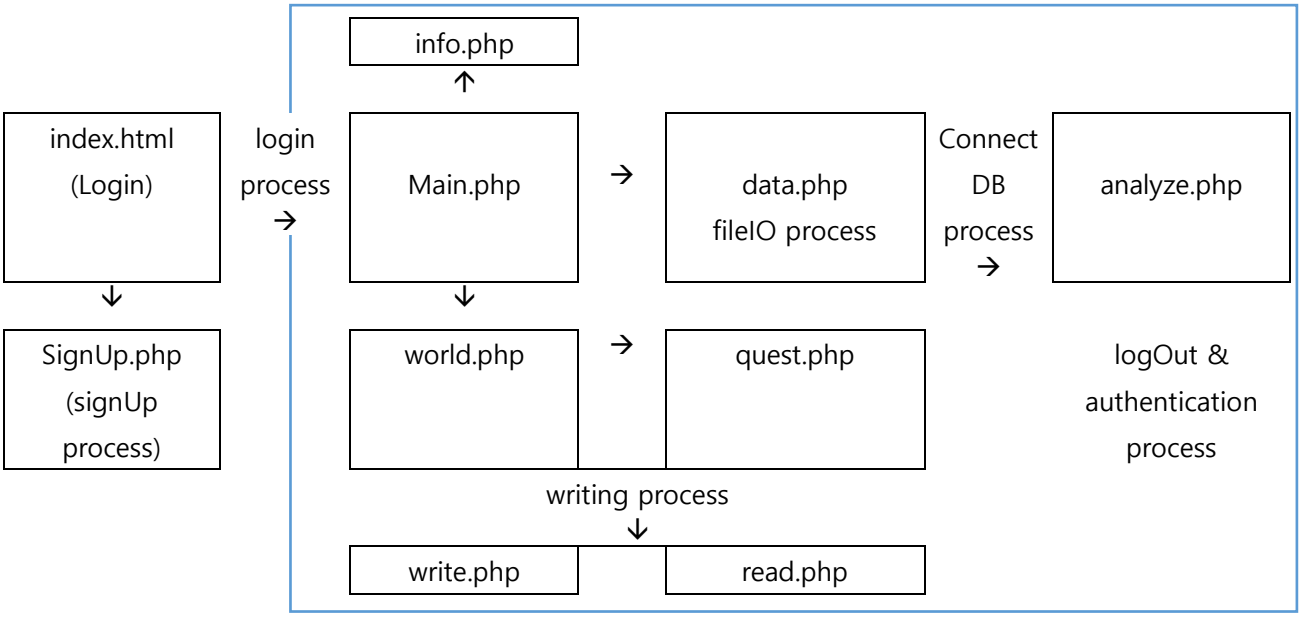
Create
$sql = "create table `bda`.`".$id."`(".$_POST['val'].")"; $uploaddir = '../data/'; $uploadfile = $uploaddir.basename($_FILES['userfile']['name']); $fileName = $_FILES['userfile']['name']; $sql = "LOAD DATA LOCAL INFILE '/var/www/html/data/".$fileName."' INTO TABLE `".$id."` FIELDS TERMINATED BY ','";
Delete

$sql = "drop table ".$id."";

- write_answer.php & write_process.php

Write Article
if($board == 'Free') $board = "free"; else if($board == 'Credit') $board = "credit"; $sql = "INSERT INTO ".$board." (`title`, `writer`, `text`, `time`) VALUES ('$title', '$id','$content', CURRENT_TIMESTAMP)";
Write Answer
$sql = "INSERT INTO answer (`postid`, `type`, `writer`, `text`, `time`) VALUES ('$post_id', '$board', '$id','$content', CURRENT_TIMESTAMP)";

③ first: index.html



<figure 3. html&php flow>

2) DB designs (DB Name: bda)

(1) Sign Up DB

① This DB stores information when a user subscribes to this service.

② signup

-Form

ID: VARCHAR(100), NOT NULL, PRIMARY KEY

NAME: VARCHAR(100), NOT NULL

PASSWORD: VARCHAR(100), NOT NULL

BIRTH: INT, NOT NULL

-Query

create table `bda`.`signup`(`ID` VARCHAR(100) NOT NULL, `NAME` VARCHAR(100) NOT NULL, `PASSWORD` VARCHAR(100) NOT NULL, `BIRTH` INT NOT NULL, PRIMARY KEY(`ID`));

(2) Board DB

① This DB provides user-to-user discussions, users-to-service administrators, or users' personal space.

② credit

-Form

postId: INT, NOT NULL, AUTO_INCREMENT, PRIMARY KEY

title: VARCHAR(100), NOT NULL

writer: VARCHAR(100), NOT NULL

text: VARCHAR(50000), NOT NULL

time: FLOAT, NOT NULL

-Query

create table `bda`.`credit`(`postId` INT NOT NULL AUTO_INCREMENT, `title` VARCHAR(500) NOT NULL, `writer` VARCHAR(100) NOT NULL, `text` VARCHAR(50000) NOT NULL, `time` FLOAT NOT NULL, PRIMARY KEY(`postId`));

③ free

-Form

postId: INT, NOT NULL, AUTO_INCREMENT, PRIMARY KEY

title: VARCHAR(100), NOT NULL

writer: VARCHAR(100), NOT NULL

text: VARCHAR(50000), NOT NULL

time: FLOAT, NOT NULL

-Query

create table `bda`.`free`(`postId` INT NOT NULL AUTO_INCREMENT, `title` VARCHAR(500) NOT NULL, `writer` VARCHAR(100) NOT NULL, `text` VARCHAR(50000) NOT NULL, `time` FLOAT NOT NULL, PRIMARY KEY(`postId`));

④ answer

-Form

postId: INT, NOT NULL

type: VARCHAR(20), NOT NULL

writer: VARCHAR(100), NOT NULL

text: VARCHAR(5000), NOT NULL

time: FLOAT, NOT NULL

-Query

create table `bda`.`answer`(`postId` INT NOT NULL, `type` VARCHAR(20) NOT NULL, `writer` VARCHAR(100) NOT NULL, `text` VARCHAR(5000) NOT NULL, `time` FLOAT NOT NULL);

(3) User's Data DB

① It is a space to store data for user's data analysis.

3) AWS Server design

1) AWS

The Amazon Web server is easy to build and cheap to use.

2) Setting

(1) Instance

① TYPE: t2.micro

② Availability zone: us-west-2a

③ AMD ID: ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20171121.1 (ami-0def3275)

④ Inbound: HTTP – TCP(Protocol), 80(Port Range), Anywhere(Source)

SSH – TCP(Protocol), 22(Port Range), Anywhere(Source)

(2) Server (I added a user named 'awsRoot' and sent the data using sftp.)

① Install: Apache/2.4.18 (Ubuntu)

PHP 7.0.22-0ubuntu0.16.04.1 (cli) (NTS)

MySQL 5.7.20-0ubuntu0.16.04.1 (Ubuntu)

(3) PHP

① pho.ini: /etc/php/7.0/apache2/php.ini

/etc/php/7.0/cli/php.ini

② Edit: upload_max_filesize

post_max_size

max_execution_time

max_input_time

memory_limit

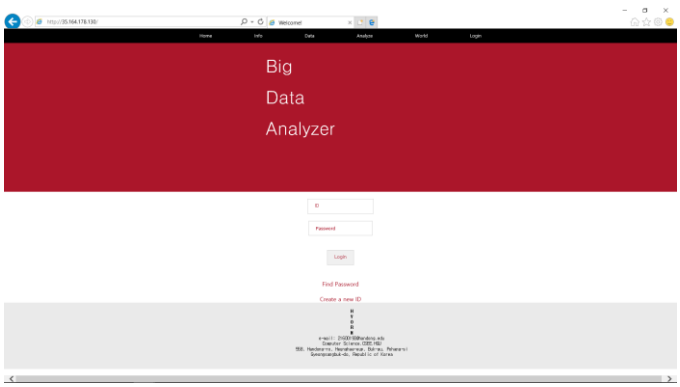
(post_max_size > upload_max_filesize >= memory_limit)

3. Service Scenarios

1. Browser

1)Internet Explorer

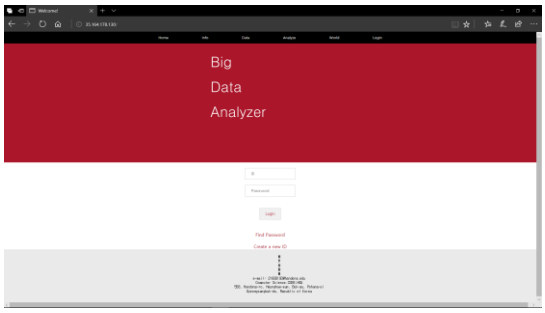
(1) index



(2) main

2) Microsoft Edge

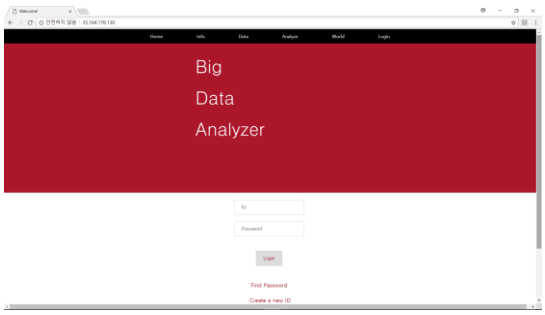
(1) index



(2) main

3) Chrome

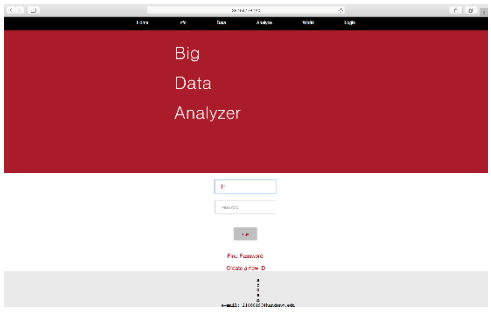
(1) index



(2) main

4) Safari

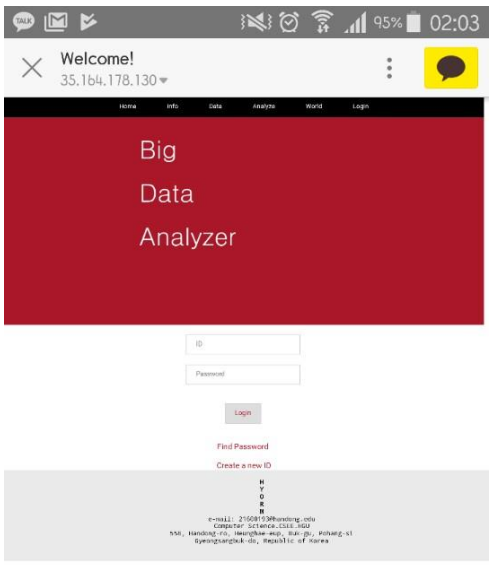
(1) index



(2) main

5) Mobile

(1) index

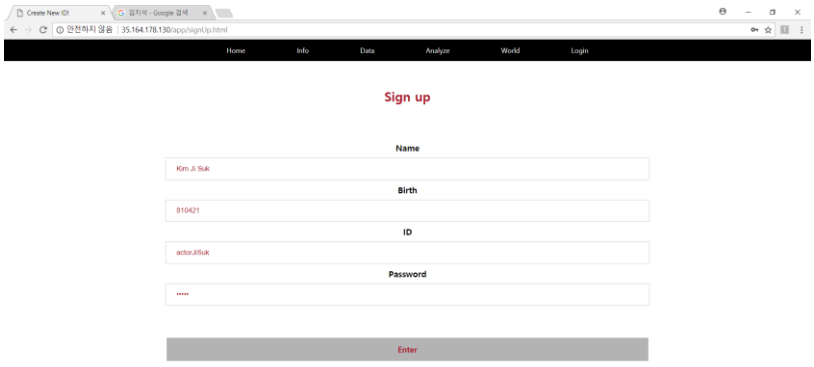


(2) main

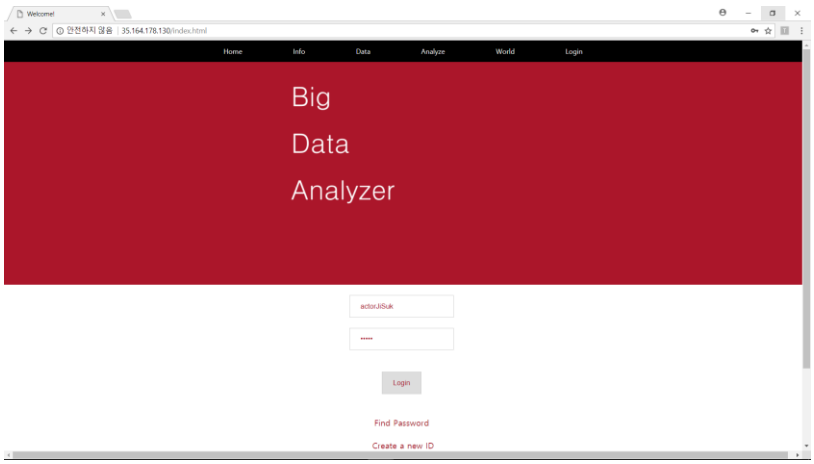
2. Virtual execution - Chrome

1) Sign

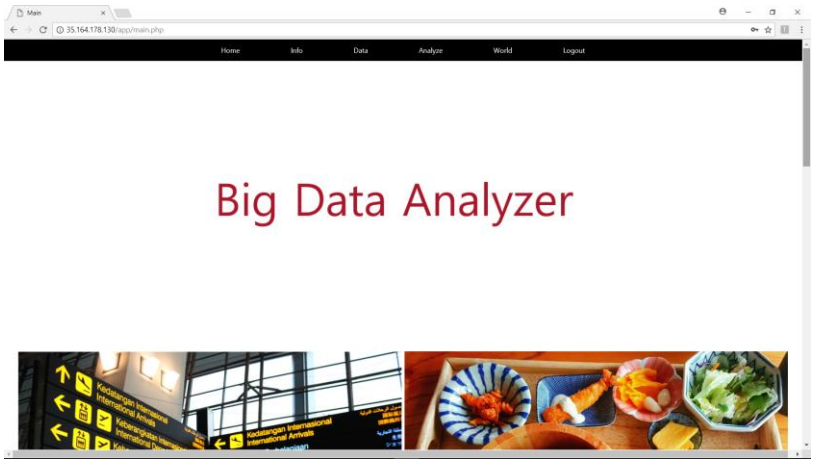
(1) Sign Up



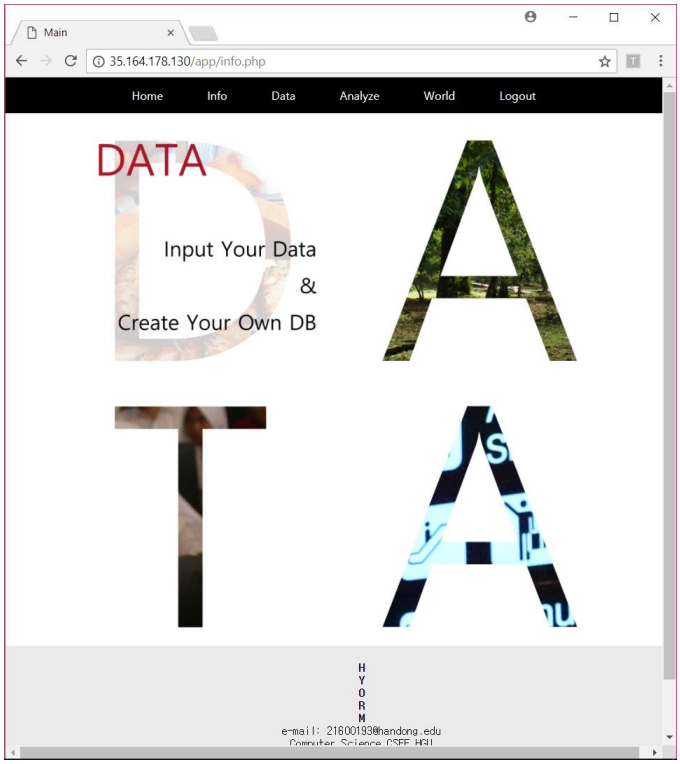
(2) Login



2) Main

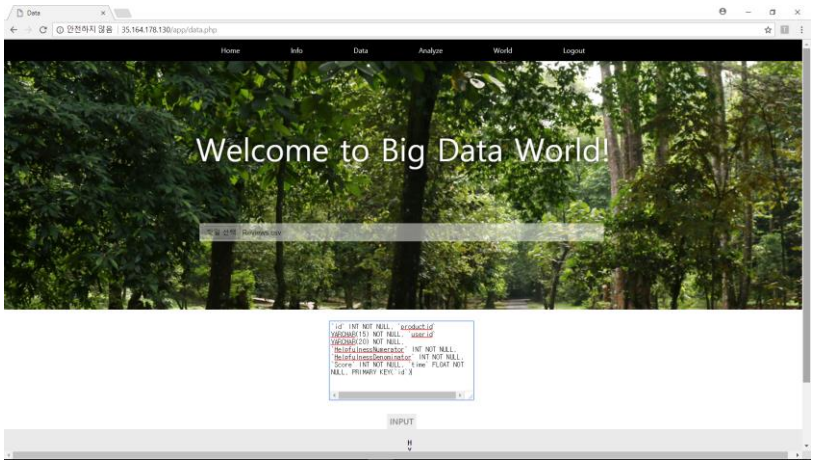


3) Info

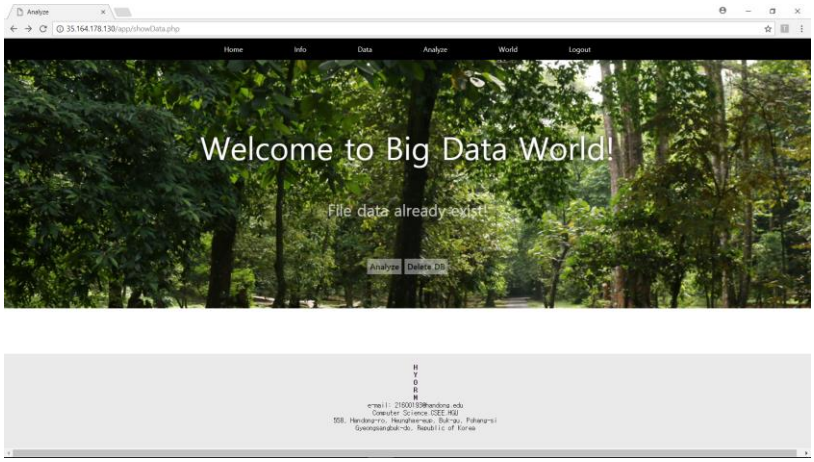


4) Data

(1) Input Data – open your file & input data query

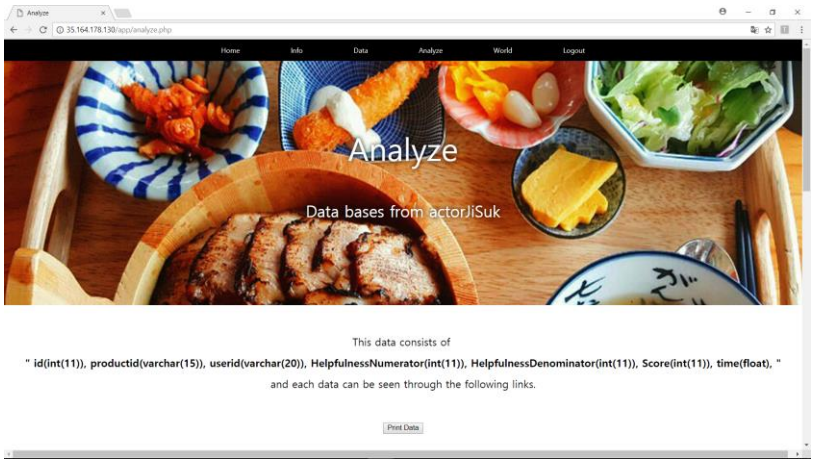


(2) Data already exist

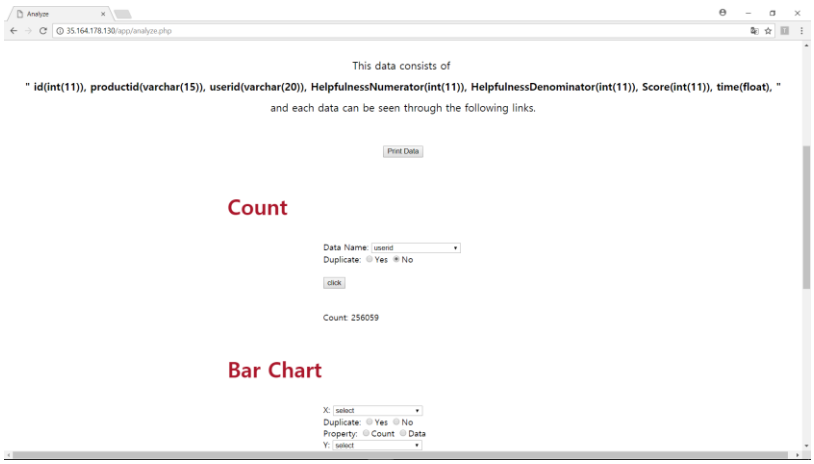


5) Analyze

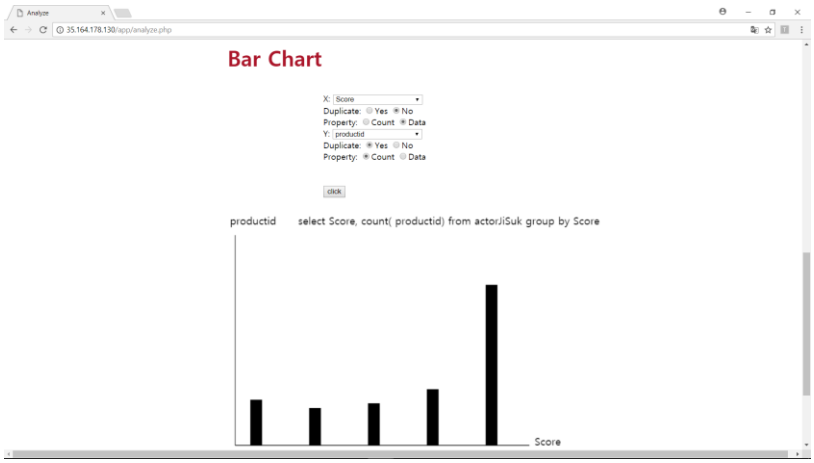
(1) Analyze



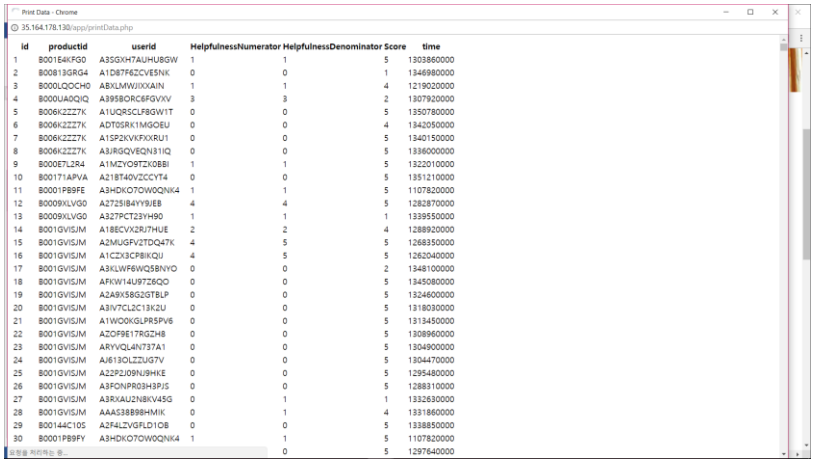
(2) Count



(3) Bar Chart



(4) Print Data

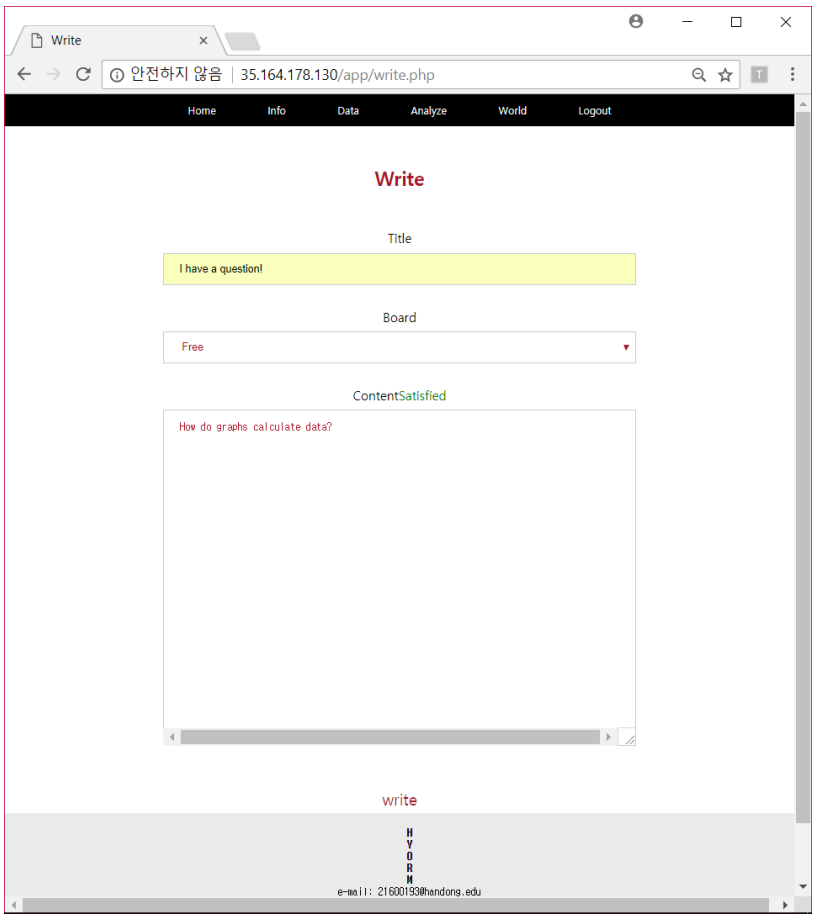


id	productid	userid	HelpfulnessNumerator	HelpfulnessDenominator	Score	time
1	B001E4KFG0	A3SGXH7AUHUBGW	1	1	5	1303860000
2	B00813GRG4	A1D87F6ZCVE5NK	0	0	1	1346980000
3	B000LQOCH0	ABXLMWJIXXAIN	1	1	4	1219020000
4	B000UA0QIQ	A395BORC6FGVXV	3	3	2	1307920000
5	B006K2ZZ7K	A1UQRSCLF8GW1T	0	0	5	1350780000
6	B006K2ZZ7K	ADT0SRK1MGOEU	0	0	4	1342050000
7	B006K2ZZ7K	A1SP2KVKFXXRU1	0	0	5	1340150000
8	B006K2ZZ7K	A3JRGQVEQN31IQ	0	0	5	1336000000
9	B000E7L2R4	A1MZYO9TZK0BBI	1	1	5	1322010000
10	B00171APVA	A21BT40VZCCYT4	0	0	5	1351210000
11	B0001PB9FE	A3HDKO7OW0QNK4	1	1	5	1107820000
12	B0009XLVG0	A2725IB4YY9JEB	4	4	5	1282870000
13	B0009XLVG0	A327PCT23YH90	1	1	1	1339550000
14	B001GVISJM	A18ECVX2RJ7HUE	2	2	4	1288920000
15	B001GVISJM	A2MUGFV2TDQ47K	4	5	5	1268350000
16	B001GVISJM	A1CZX3CP8IKQU	4	5	5	1262040000
17	B001GVISJM	A3KLWF6WQ5BNYO	0	0	2	1348100000
18	B001GVISJM	AFKW14U97Z6QO	0	0	5	1345080000
19	B001GVISJM	A2A9X58G2GTBLP	0	0	5	1324600000
20	B001GVISJM	A3IV7CL2C13K2U	0	0	5	1318030000
21	B001GVISJM	A1WO0KGLPR5PV6	0	0	5	1313450000
22	B001GVISJM	AZOF9E17RGZHB	0	0	5	1308960000
23	B001GVISJM	ARYVQL4N737A1	0	0	5	1304900000
24	B001GVISJM	AJ613OLZZUG7V	0	0	5	1304470000
25	B001GVISJM	A22PJ09NJ9HKE	0	0	5	1295480000
26	B001GVISJM	A3FONPR03H3PJS	0	0	5	1288310000
27	B001GVISJM	A3RXAU2N8KV45G	0	1	1	1332630000
28	B001GVISJM	AAAS38B98HMIK	0	1	4	1331860000
29	B00144C10S	A2F4LZVGFLD1OB	0	0	5	1338850000
30	B0001PB9FY	A3HDKO7OW0QNK4	1	1	5	1107820000
				0	5	1297640000

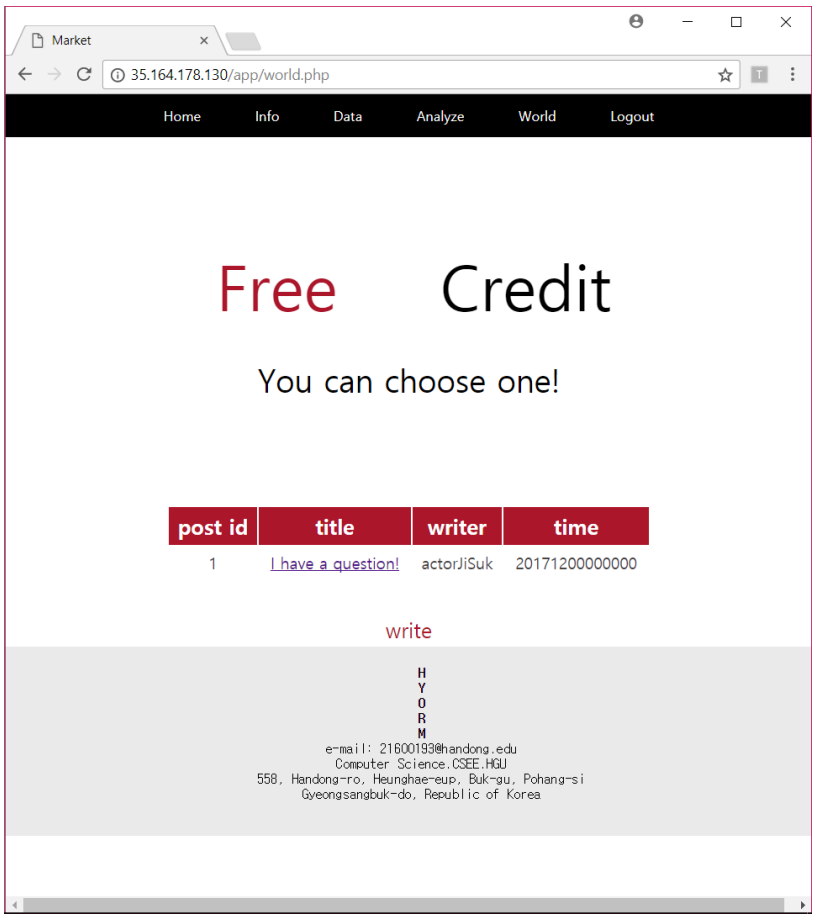
6) World

(1) Free

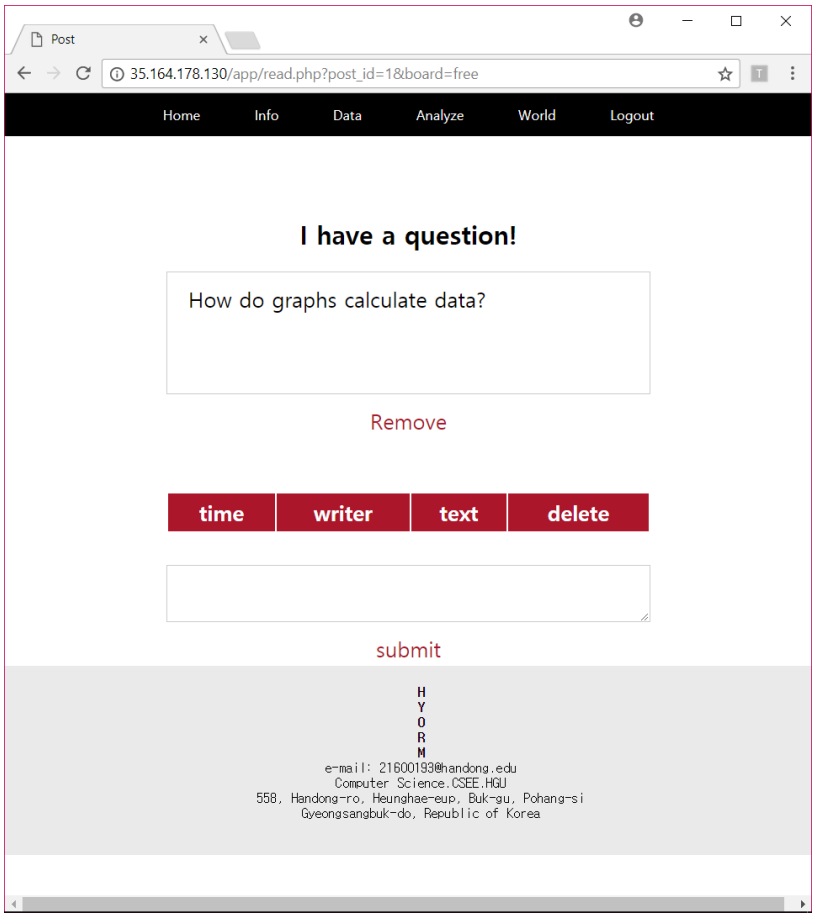
① write


Write
안전하지 않음 | 35.164.178.130/app/write.php
Home
Info
Data
Analyze
World
Logout
Write
Title
I have a question!
Board
Free
ContentSatisfied
How do graphs calculate data?
write
HYORM
e-mail: 21600193@handong.edu

② board


Market
35.164.178.130/app/world.php
Home
Info
Data
Analyze
World
Logout
Free
Credit
You can choose one!
post id
title
writer
time
1
I have a question!
actorJiSuk
20171200000000
write
HYORM
e-mail: 21600193@handong.edu
Computer Science.CSEE.HGU
558, Handong-ro, Heunghae-eup, Buk-gu, Pohang-si
Gyeongsangbuk-do, Republic of Korea

③ Read



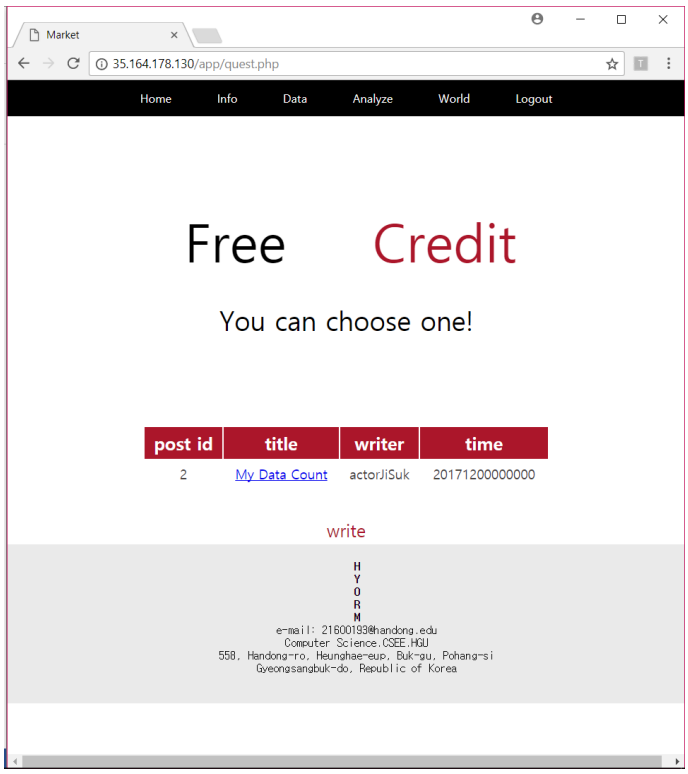
(2) Credit

① write

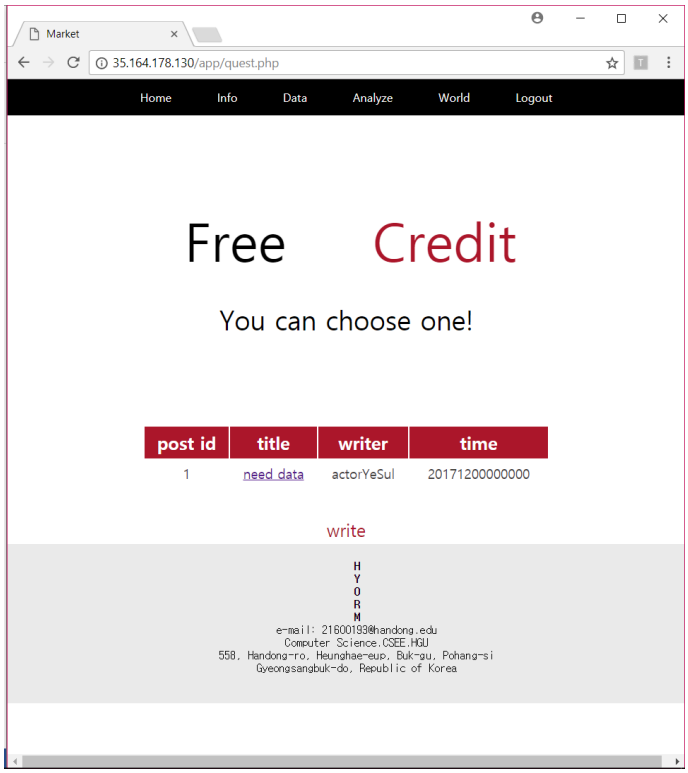
(same (1)-①)

② board

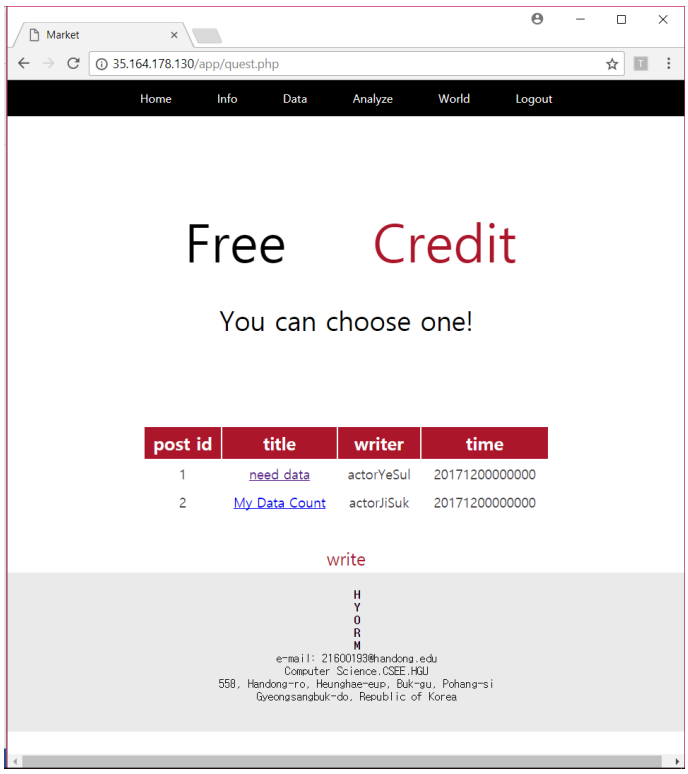
-user: actorJiSuk



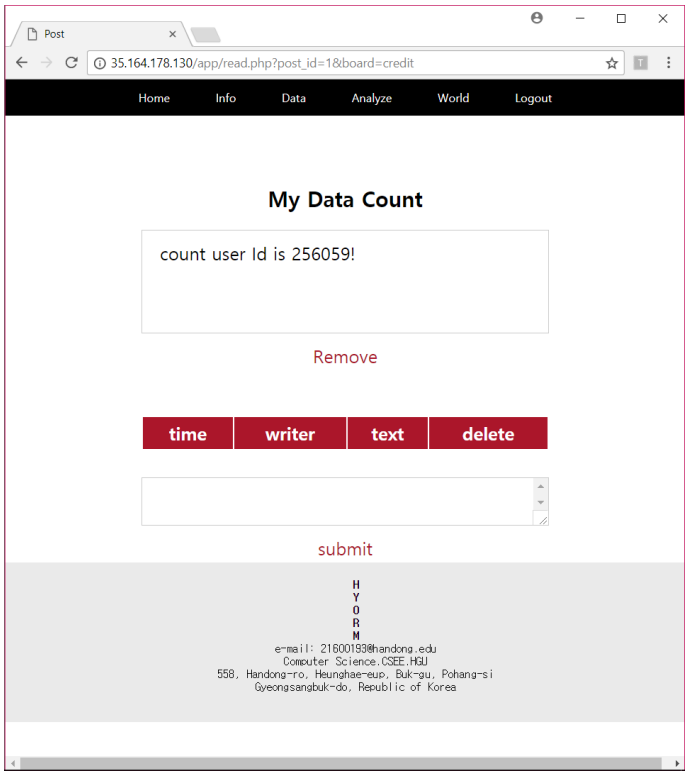
-user: actorYeSul



-user: root

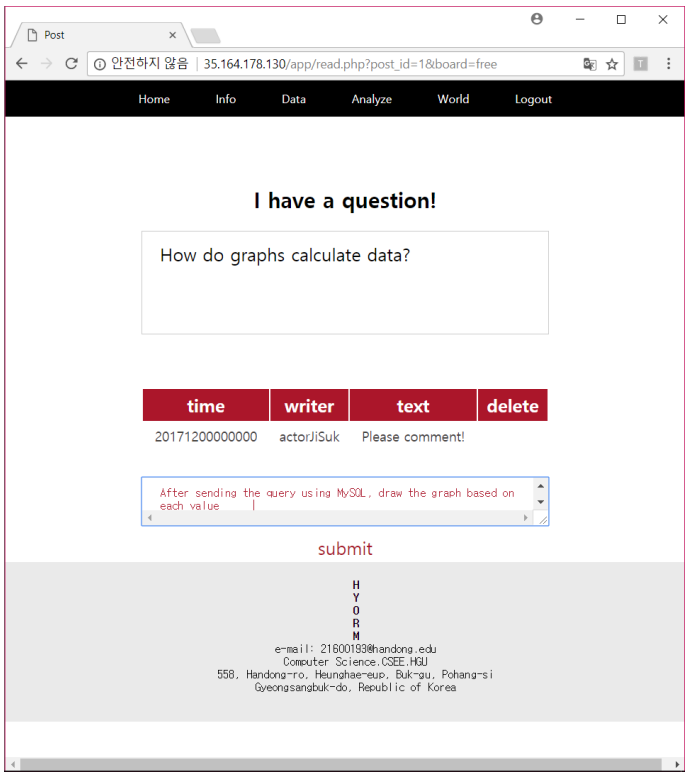


③ Read



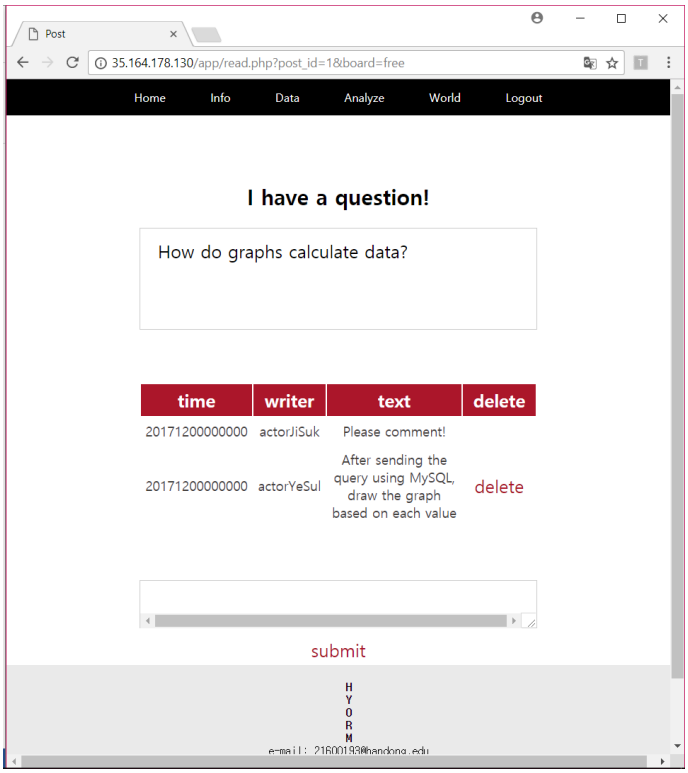
(3) Answer

① Write

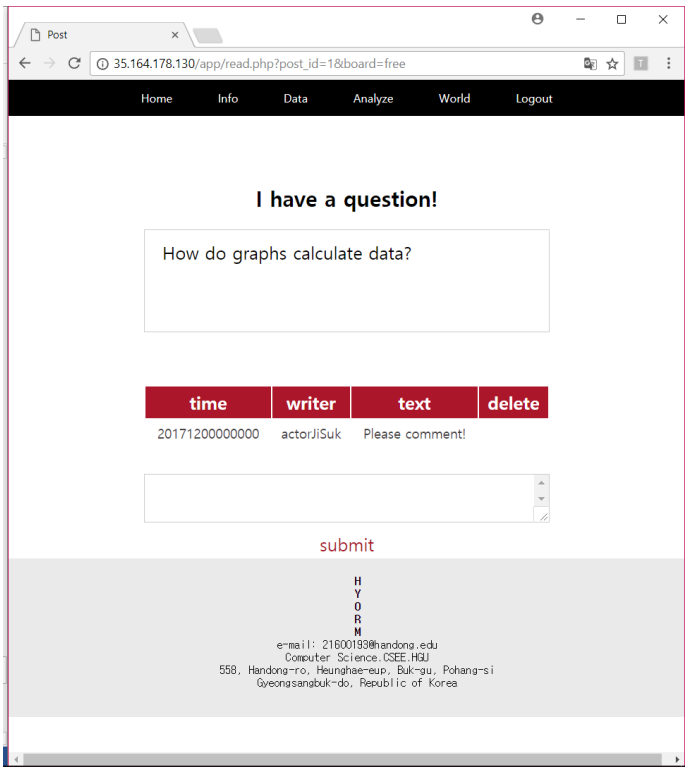


② remove

- Before



- After



4. Result

I enjoyed using Amazon Web Services to build and service real web services. When I first thought about the service, it seemed to be easy to provide the service. However, when I tried actual testing with various people, I experienced a lot of errors in the service. Also, analyzing big data was difficult because it had so many variables. I would like to keep updating the missing parts.