

Project Description

The project is a forum-style website with the main purpose of communication and interaction between like-minded individuals on the platform. The domain of the app is social media, particularly online interactions, news sharing, and general communication. Users are able to generate posts and comments with the ability to like or dislike them. The posts can have attachments including images, text, and video. Two types of users will be classified; guests and members. Any visitors to the site are classified as guests until they make an account. Only members are given access to all features stated above, guests can only view content.

Schema Modifications

We removed the 'Action' and all related actions (Like, dislike, edit, delete, censor) from the schema, as those were not necessary for the implementation of the project. Instead, we are tracking and storing information about the likes and dislikes directly in the post and comment itself. Any other actions like edit and delete will be reflected directly in the content stored as part of "UserContent." Additionally, we removed the separate tables from attachment and combined them all into a separate field of attachment, as it makes more sense for the attachment types and information to be fields of the attachment. Lastly, we removed custom figures and badges, as those were not necessary for the implementation of the project.

Updated Schema

Visitor(ip: string, experience: integer NOT NULL, vid: string PK, datecreated: date, name: string)

Guest(gid:string PK/FK(Visitor.vid))

Member(mid:string PK/FK (Visitor.vid), email:string NOT NULL, password: string NOT NULL)

UserContent(cid:string PK, mid: stringFK(member.mid), datecreated: date)

Post(pid: string PK/FK (UserContent.cid), upvotes: integer NOT NULL, downvotes: integer NOT NULL, title: string NOT NULL)

Attachment(attid: string PK, pid: string FK (Post.pid), type: string, content: string)

Views(vid: string NOT NULL, pid: string NOT NULL, PK (vid, pid))

Data in each Relation:

Visitor:

```
SQL> select * from visitor;
```

IP	EXPERIENCE	VID	DATECREAT	NAME
192.010.303	50	10302	01-SEP-22	Bob
192.163.093	1	20530	28-FEB-19	Alice
192.156.525	1354	778	10-JAN-20	Jonathan
192.321.444	642	1053	12-JUN-21	Faker
192.111.593	123	10111	22-AUG-12	doublelift
192.121.545	12533	17	17-MAR-23	soccerplayer1
192.101.351	53433	3350	05-APR-11	kosmos
192.113.041	32987	351	31-DEC-22	zeezan
192.164.013	25921	41	03-JUL-13	snakes
192.151.343	8932	873	04-APR-15	cr7ronaldo

10 rows selected.

Guest:

```
SQL> select * from guest;
```

```
GID
-----
10111
10302
1053
20530
778
```

Member:

```
SQL> select * from member;
```

MID	EMAIL	PASSWORD
17	soccerplayer4@outlook.com	logindude
3350	throwawayabc@email.com	logindude
351	olderperson@gmail.com	logindude
41	gamer12@outlook.com	logindude
873	cpscstudent@yahoo.com	logindude

Post:

```
SQL> select * from Post;
```

PID	MID	UPVOTES	DOWNVOTES	TITLE
158	17	5001		2500 Ronaldo Leaves Real Madrid!
9932	3350	10		5345 Dr. Recommends an apple a day.
8168	351	947		5 Elden Ring most played game of 2022
554	41	8811		23 Why dont we have any moderators on this site!?!?
9002	873	1000		9923 Unpopular Opinion: water sucks

UserComment:

```
SQL> select * from usercomment;
```

CUID	PID	MID	UPVOTES	DOWNVOTES	CONTENT
31	158	3350	31	5	wow thats really insightfull!
33324	9932	351	40	10	youre just wrong here, how can you say that!?
8217	8168	41	35	100	ill have you know I graduated at the top of my class
132	554	873	4512	23	actually if you use this weapon you can get 100 more damage
53452	9002	17	9529	501	the site has come a long way now!

Attachment:

```
SQL> select * from attachment;
```

ATTID	PID	TYPE	CONTENT
5	158	image	www.image1.com
6	9932	link	www.github.com
7	8168	video	www.youtube.com/watch?v=vS3_72Gb-b1
8	554	text	I think that there are many things we should consider regarding laws relating to gambling.
9	9002	image	www.image3.com

Views:

```
SQL> select * from views;
```

VID	PID
10302	158
10302	554
10302	8168
10302	9002
10302	9932
41	554
778	9932
873	9932

SQL Queries:

All found in API/server.js (referenced by line number)

Example: INSERT QUERY

Line 67: INSERT INTO Visitor (ip, experience, vid, datecreated) VALUES(:ip, :experience, :vid, :time)

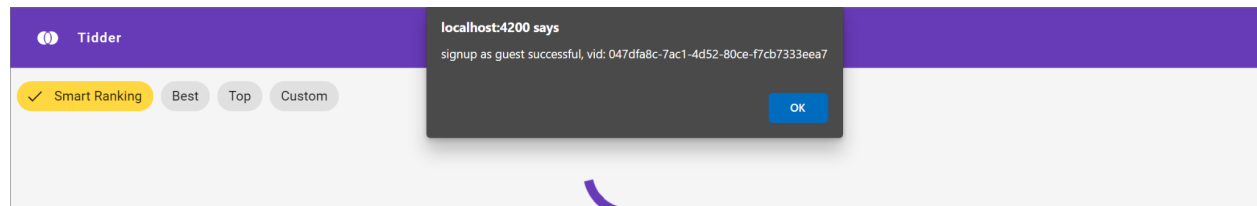
Before:

```
SQL> select * from visitor;
```

IP	EXPERIENCE	VID	DATECREAT	NAME
192.010.303	50	10302	01-SEP-22	Bob
192.163.093	1	20530	28-FEB-19	Alice
192.156.525	1354	778	10-JAN-20	Jonathan
192.321.444	642	1053	12-JUN-21	Faker
192.111.593	123	10111	22-AUG-12	doublelift
192.121.545	12533	17	17-MAR-23	soccerplayer1
192.101.351	53433	3350	05-APR-11	kosmos
192.113.041	32987	351	31-DEC-22	zeezan
192.164.013	25921	41	03-JUL-13	snakes
192.151.343	8932	873	04-APR-15	cr7ronaldo

10 rows selected.

During:



After:

```
SQL> select * from visitor;
```

IP	EXPERIENCE	VID	DATECREAT	NAME
0	0	a8e08eb6-ec6b-4f75-b33f-4c4d0bcaf0a2	22-APR-21	
0	0	240a7644-5998-4006-9f1d-09c3c0f81a6b	22-APR-21	
192.010.303	50	10302	01-SEP-22	Bob
192.163.093	1	20530	28-FEB-19	Alice
192.156.525	1354	778	10-JAN-20	Jonathan
192.321.444	642	1053	12-JUN-21	Faker
192.111.593	123	10111	22-AUG-12	doublelift
192.121.545	12533	17	17-MAR-23	soccerplayer1
192.101.351	53433	3350	05-APR-11	kosmos
192.113.041	32987	351	31-DEC-22	zeezan
192.164.013	25921	41	03-JUL-13	snakes

IP	EXPERIENCE	VID	DATECREAT	NAME
192.151.343	8932	873	04-APR-15	cr7ronaldo
0	0	047dfa8c-7ac1-4d52-80ce-f7cb7333eea7	22-APR-21	
0	0	28188faf-108d-43b3-8cc9-081d1c695d9e	22-APR-21	
0	0	f2704d45-16c5-4124-b68c-c0d3b21a2abd	22-APR-21	
0	0	3f497f60-2281-4fed-89d1-25c6eb4d59f2	22-APR-21	

16 rows selected.

Line 77: INSERT INTO Guest (gid) VALUES(:vid)

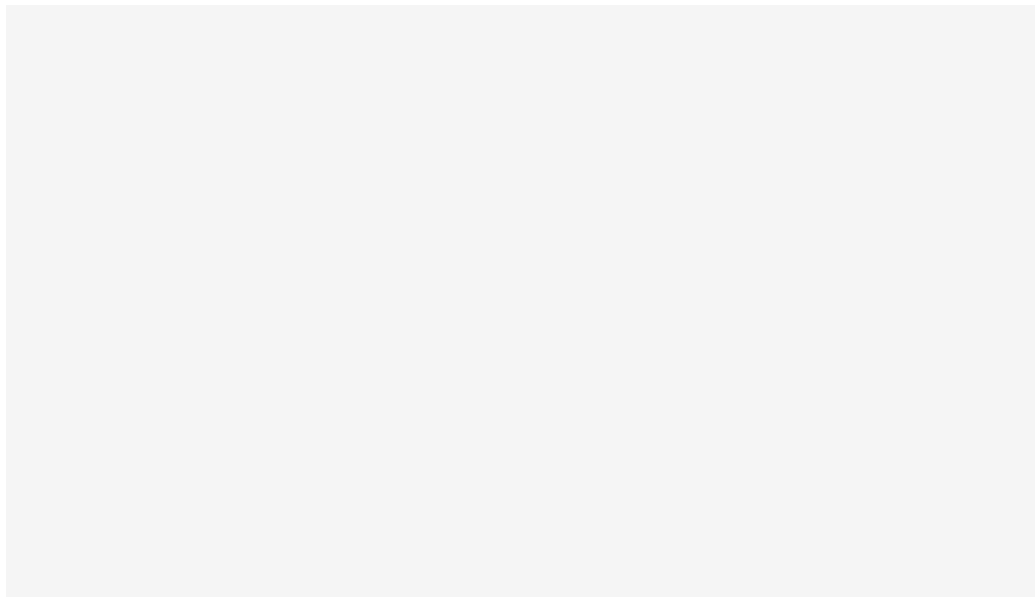
Line 107: DELETE FROM Guest WHERE gid = :vid

Line 117: INSERT INTO Member (mid, email, password) VALUES(:vid, :email, :password)

EXAMPLE: JOIN QUERY

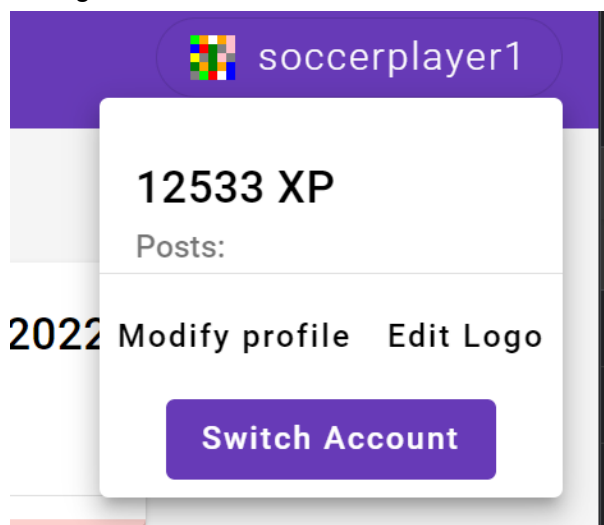
Line 179: SELECT DISTINCT vid, name, experience, email FROM Visitor, Member, Guest
WHERE vid = mid AND vid = :vid

Before:



(empty)

During/After:



Line 220: DELETE FROM Visitor WHERE vid = :vid

Line 249: UPDATE Visitor SET NAME=:name WHERE vid=:vid

Line 290: INSERT INTO Post (pid, mid, upvotes, downvotes, title) VALUES(:cid, :mid, :upvotes, :downvotes, :title)

Line 304: INSERT INTO Attachment (attid, pid, type, content) VALUES(:attid, :pid, :type, :content)

EXAMPLE: DELETE QUERY

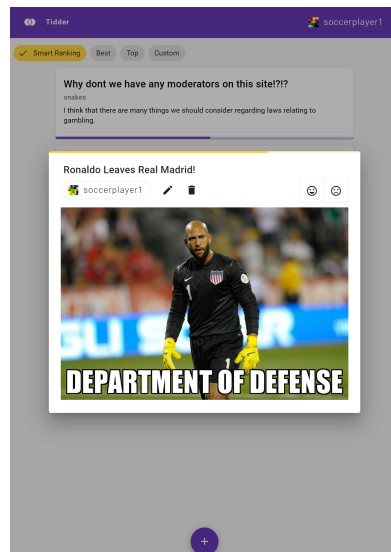
Line 338: DELETE FROM Post WHERE pid = :cid

Before:

```
SQL> select * from post;
```

PID	MID	UPVOTES	DOWNVOTES	TITLE
158	17	5001	2500	Ronaldo Leaves Real Madrid!
9932	3350	10	5345	Dr. Recommends an apple a day.
8168	351	947	5	Elden Ring most played game of 2022
554	41	8811	23	Why dont we have any moderators on this site!?!?
9002	873	1000	9923	Unpopular Opinion: water sucks

During:



After:

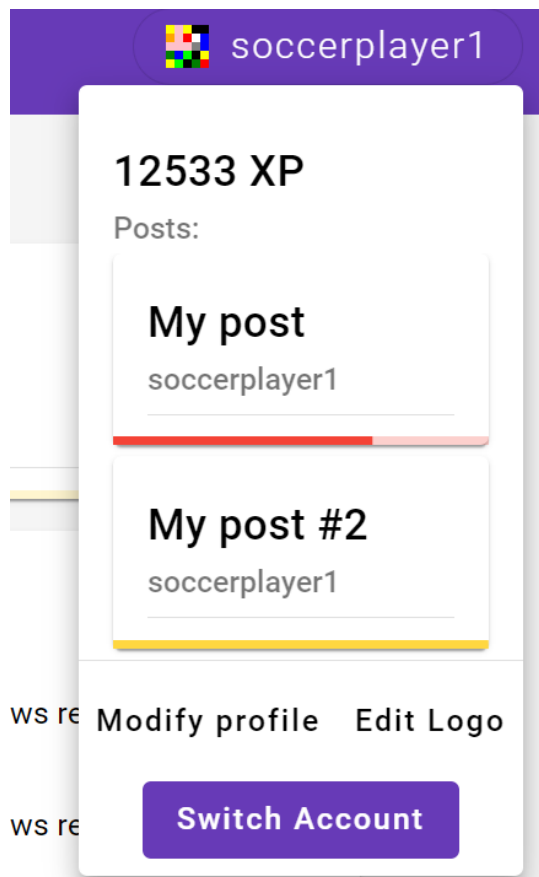
```
SQL> select * from post;
```

PID	MID	UPVOTES	DOWNVOTES	TITLE
9932	3350	10	5345	Dr. Recommends an apple a day.
8168	351	947	5	Elden Ring most played game of 2022
554	41	8811	23	Why dont we have any moderators on this site!?!?
9002	873	1000	9923	Unpopular Opinion: water sucks

EXAMPLE: AGGREGATION BY GROUP

Line 366: SELECT pid, COUNT(pid) FROM Views GROUP BY pid ORDER BY COUNT(pid) DESC

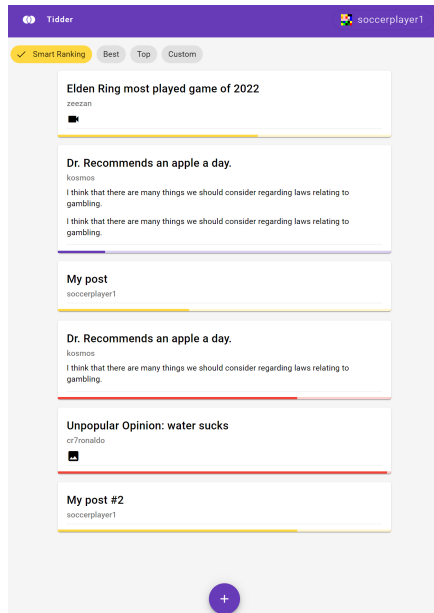
Result:



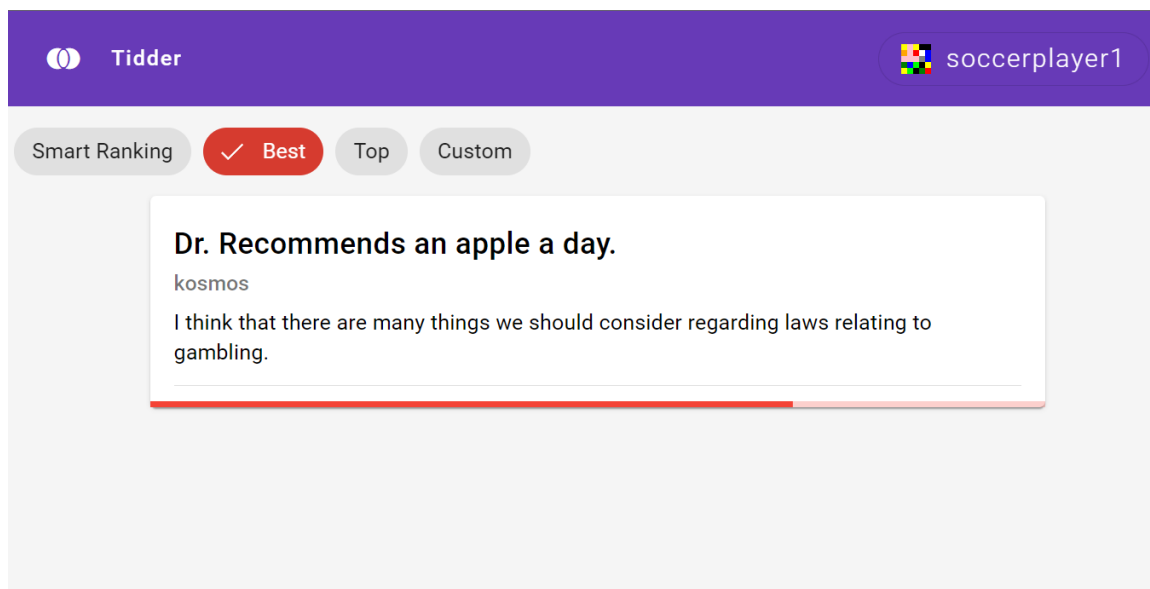
EXAMPLE: NESTED AGGREGATION WITH GROUP BY

Line 390: SELECT p.pid from Post p GROUP BY p.pid having MAX(p.upvotes) >= ALL (SELECT MAX(p1.upvotes) FROM post p1 GROUP BY pid)

Before:



After:



Line 411: SELECT PID FROM POST

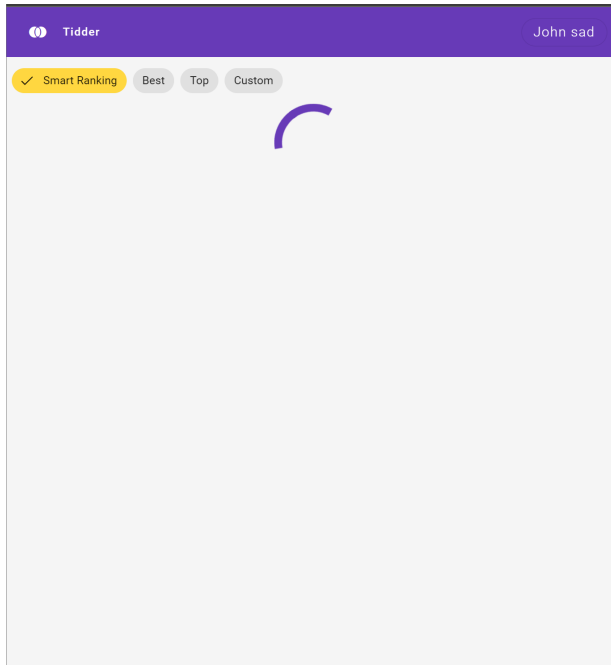
Line 452: SELECT pid FROM Post WHERE mid = :cid

Line 497: SELECT DISTINCT pid, mid, name, upvotes, downvotes, title FROM Visitor v, Post p
WHERE p.pid = :cid AND p.mid = v.vid

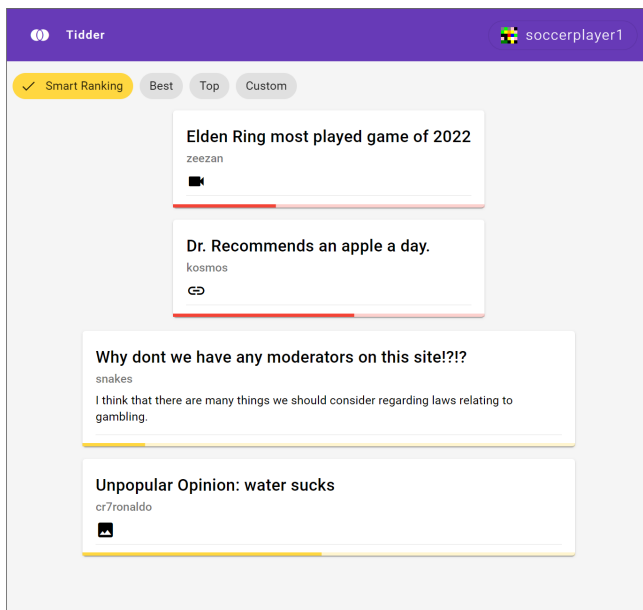
EXAMPLE: PROJECTION

Line 517: SELECT attid, type, content FROM Attachment WHERE pid = :cid

Before:



During:



After:

```
▼ (4) ['554', '8168', '9002', '9932'] ⓘ
  0: "554"
  1: "8168"
  2: "9002"
  3: "9932"
  length: 4
  ► [[Prototype]]: Array(0)
```

EXAMPLE: UPDATE QUERY

Line 558: UPDATE Post SET upvotes = upvotes + 1 WHERE pid=:cid

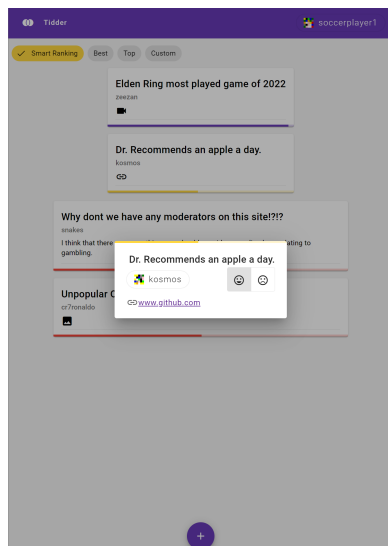
Before:

```
SQL> select * from post where pid = '9932';
```

PID	MID	UPVOTES	DOWNVOTES	TITLE
9932	3350	10	5345	Dr. Recommends an apple a day.

```
SQL>
```

During:



After:

```
SQL> select * from post where pid = '9932';
```

PID	MID	UPVOTES	DOWNVOTES	TITLE
9932	3350	11	5345	Dr. Recommends an apple a day.

Line 587: UPDATE Post SET downvotes = downvotes + 1 WHERE pid=:cid

Line 616: SELECT * FROM Attachment WHERE attid = :attid

Line 653: INSERT INTO UserComment (coid, pid, mid, upvotes, downvotes, content)
VALUES(:coid, :pid, :mid, :upvotes, :downvotes, :content)

Line 682: UPDATE UserComment SET CONTENT=:content WHERE coid=:id

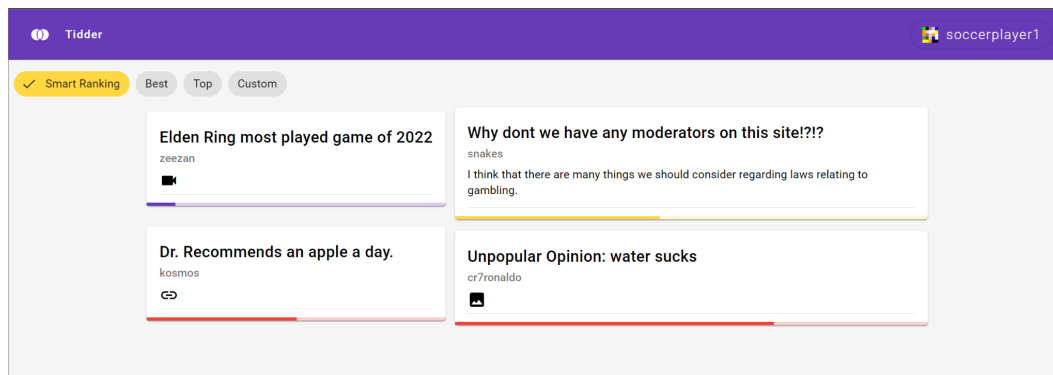
Line 712: DELETE FROM UserComment WHERE coid = :coid

Line 743: INSERT INTO Views VALUES(':vid', ':pid')

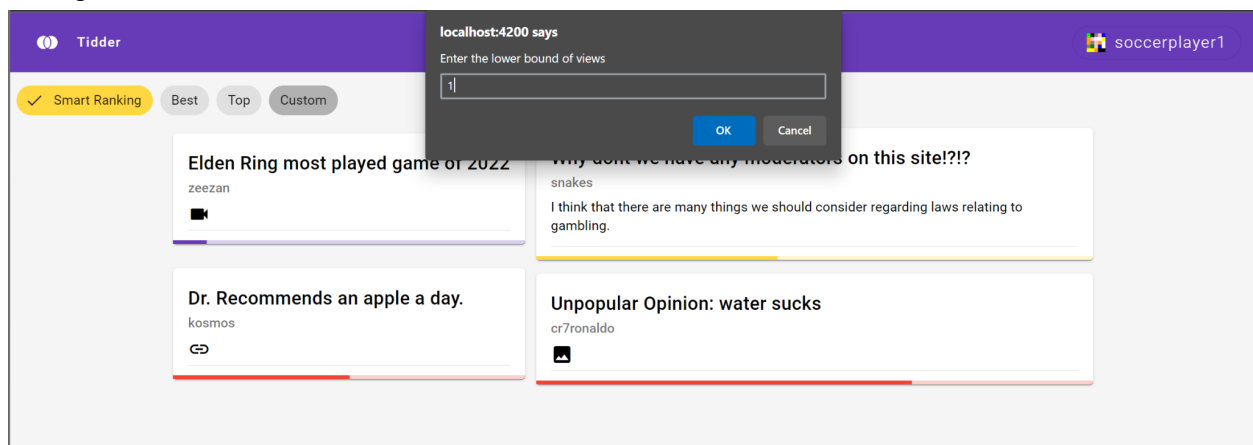
EXAMPLE: AGGREGATION WITH HAVING QUERY:

Line 771: SELECT DISTINCT p.pid from Post p, Views v WHERE p.pid = v.pid GROUP BY
p.pid HAVING COUNT(*) > :input

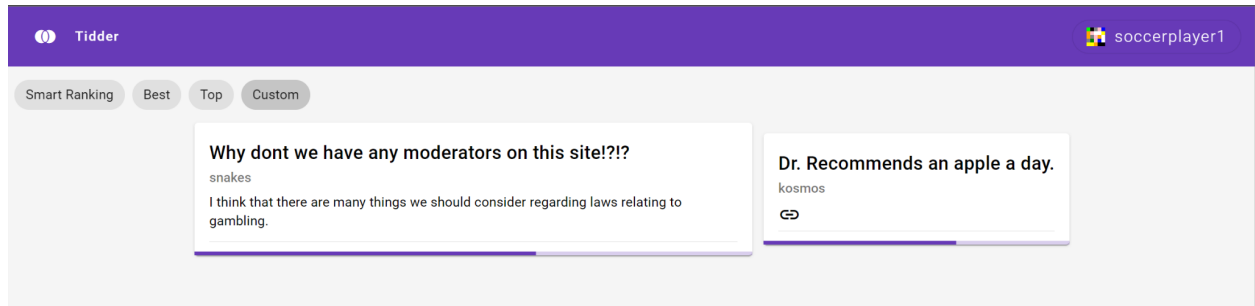
Before:



During:



After:



EXAMPLE: DIVISION QUERY:

Line 808: `SELECT v.name FROM Visitor V WHERE NOT EXISTS((SELECT P.pid FROM Post P) MINUS (SELECT w.pid FROM Views w WHERE w.vid = v.vid))`

Before:

During:

After: