

Project Final Report

CS 353 - Database Systems

Group 7
Figali Taho
Dias Alymbekov
Elena Cina
Erkam Berker Senol

December 2017



Faculty of Engineering
Department of Computer Engineering

Contents

1	Project Description	4
2	Final ER-Diagram	5
3	Final list of tables.	6
3.1	User	6
3.2	Admin	6
3.3	Moderator	6
3.4	Content	6
3.5	Post	6
3.6	Comment	6
3.7	Category	6
3.8	Topic	6
3.9	Message	6
3.10	Category_Topic	7
3.11	Vote	7
3.12	Follows	7
3.13	Banned	7
4	Implementation details	8
4.1	Technologies used	8
4.2	Issues faced	8
5	Advanced Database Components	9
5.1	Triggers	9
5.1.1	Voting triggers	9
5.1.2	Comment triggers	10
5.1.3	Post triggers	10
5.1.4	Category_Topic trigger	10
5.1.5	Category trigger	10
5.2	Procedures	11
5.2.1	Register Admin	11
5.2.2	Register Moderator	11
5.3	Reports	11
5.4	Indexes	12
5.5	Views	12
5.5.1	Homepage	12
5.5.2	Best of last week	12
6	A User's Manual describing the usage of our system	13
6.1	Introduction	13
6.2	For all users	13
6.2.1	Home screen	13
6.3	View content	14
6.3.1	Search	14
6.3.2	Category screen	15
6.3.3	Topic screen	16
6.3.4	User Profile	16
6.4	How to register	17
6.5	Login	17
6.6	Homepage	18
6.7	Profile and user settings page	19
6.8	Request adminship and moderatorship	19
6.9	Admin functionalities	20
6.9.1	Delete category	20
6.9.2	Delete user	21
6.10	Moderator functionalities	21

6.10.1 Delete post	21
6.11 Adding posts	21
6.11.1 Add a text post	21
6.11.2 Add a link post	22
6.12 Create new categories	22
6.13 Create New Topic	23
6.14 Follow other users	23
6.15 Message other users	23

1 Project Description

The project is a system for a social discussion website similar to Reddit. There are many topics which users can open themselves and on which every user can discuss with each other via commenting. Each topic belongs to a certain category. Categories are predefined prior to being opened or users can create a new category. Users of this system can follow other users and should be able to comment others posts or comments on a particular discussion topic and also they can send messages to each other. There will be 3 user types: normal user, moderator and admin.

A database is used to store the information regarding users, categories and topics, subtopics and comments. This database will allow the user to search information regarding other users and categories. For example, the user can select to see a list of topics that fall under a certain category via the web-page interface. This results in a query being used to project only the topics under the category, sorted according to time. The user can also sort the list of topics by other attributes such as "popularity", derived from the number of people who have upvoted the topic, or "latest", derived from the date when the topic is posted.

Similarly, users are also stored in the database, and are identified by their unique user name, which is the primary key. The user information will also consist of attributes such as joined date, password and email address. Moreover, the various relationships that will be present, such as when the user opens a new category or creates a new topic are supported by the entities of the relationships.

The database is an integral part of the system and most of the project's features are based around the database. It is loaded onto the webpage which will be used as the interface by which the user access the database. However, the user are not able to see the database structure and instead can only see relevant information displayed on the webpage. Hence the queries that are used to retrieve the information and alter the tables will be automatically generated upon the selection of the user on the webpage interface. Additional information about our project is provided on our website: <https://ebsenol.github.io/fara.github.io/>

2 Final ER-Diagram

Below we provide the revised version of the design of the database for social discussion platform.

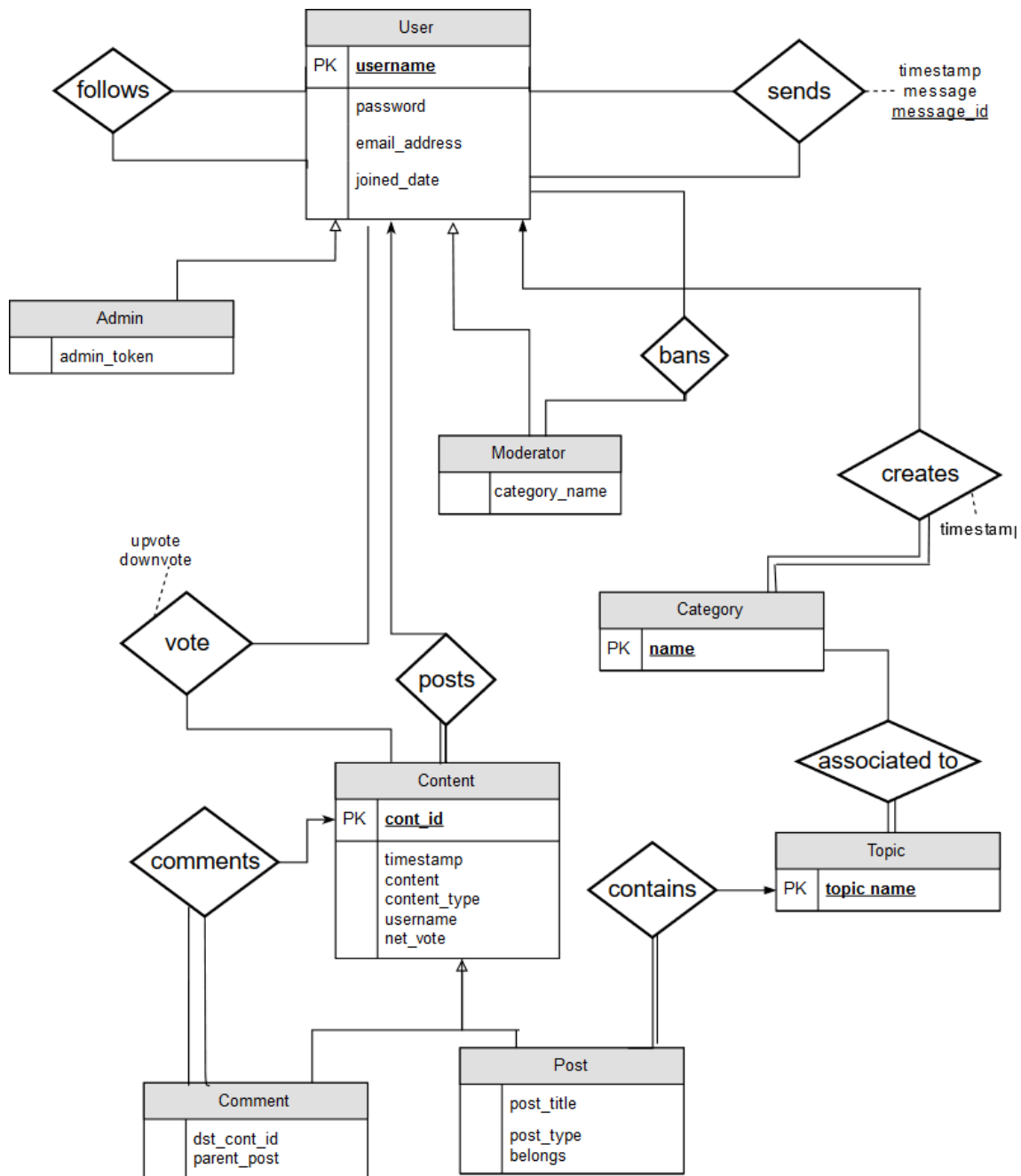


Figure 1: Final E/R diagram.

3 Final list of tables.

The final version of our project has 13 tables which are provided below.

3.1 User

User(username , password , email_address , joined_date)

3.2 Admin

Admin(username , password , email_address , joined_date, admin_token)

3.3 Moderator

Moderator(username, password , email_address , joined_date)

3.4 Content

Content(cont_id, timestamp, content, content_type, username, net_vote)

- FOREIGN KEY (username) REFERENCES User(username)

3.5 Post

Post(cont_id, post_title, post_type, belongs)

- FOREIGN KEY (belongs) REFERENCES Topic(topic_name)
- FOREIGN KEY (cont_id) REFERENCES Content(cont_id)

3.6 Comment

Comment(cont_id, username, dst_cont_id)

- FOREIGN KEY (cont_id) REFERENCES Content(cont_id)
- FOREIGN KEY (dst_cont_id) REFERENCES Content(cont_id)
- FOREIGN KEY (username) REFERENCES User(username)

3.7 Category

Category(name)

3.8 Topic

Topic(topic_name)

3.9 Message

Message(message_id, dst_name, rcv_name, timestamp, message)

- FOREIGN KEY (dst_name) REFERENCES User(username)
- FOREIGN KEY (rcv_name) REFERENCES User(username)

3.10 Category_Topic

Category_Topic(category_name, topic_name)

- FOREIGN KEY (category_name) REFERENCES Category(name)
- FOREIGN KEY (topic_name) REFERENCES Topic(topic_name)

3.11 Vote

Vote(username, cont_id, vote)

- FOREIGN KEY (username) REFERENCES User(username)
- FOREIGN KEY (cont_id) REFERENCES Content(cont_id)

3.12 Follows

Follows(follower, following)

- FOREIGN KEY (follower) REFERENCES User(username)
- FOREIGN KEY (following) REFERENCES User(username)

3.13 Banned

Banned(username, category_name)

- FOREIGN KEY (username) REFERENCES User(username)
- FOREIGN KEY (category_name) REFERENCES Category(name)

4 Implementation details

4.1 Technologies used

We have implemented this project on three different layers: data layer, service layer and presentation layer. We have utilized different stacks of technologies to implement each layer and to achieve a successful communication between layers.

In the data layer, we have used standard MySQL, and MySQL Workbench for local testing. In order to create and populate the database we have implemented Java code, thus utilizing the Java Database Connectivity API (JDBC).

In the service layer we have used PHP. Utilizing PHP we have implemented all operations of the system. All the transitions between pages, error/confirmation messages, login/logout, signin and session information managements are done with PHP. Most of PHP files contains HTML code since service layer is connected with presentation layer. Additionally, few Javascript functions embedded in HTML were utilized for transitional error or confirmation messages.

In the presentation layer have used HTML, CSS and Bootstrap for responsiveness.

4.2 Issues faced

On the technical side we made good efforts to reduce any trouble we could face with the database, taking care carefully of the primary and foreign keys, and how the tables would communicate. It was not until we implemented triggers that we realized some data was depending on other data, and deleting seemed a bit challenging. Specifically, to delete a category, every topic that is under category should be removed, for that every comment, post, vote should be removed, other than that moderators, banned users from that category should also be removed and so on. Even though implementing these triggers were challenging, it made our job much easier on the application layer.

On the organization side of the project, a problem we faced was synchronization and efficiency of each others work. Since each of us worked on different parts of the project combining the code and meticulously making sure the UI seemed good, and the functionality remained intact, was a challenge. However, through extensive group meetings and communication with each other we overcome this problem. The usage of GitHub facilitated our work a lot.

5 Advanced Database Components

5.1 Triggers

5.1.1 Voting triggers

```
CREATE TRIGGER update_netvote_after_insert
AFTER INSERT ON Vote
  FOR EACH ROW BEGIN
    IF NEW.vote THEN
      SET @val = 1;
    ELSE
      SET @val = -1;
    END IF;
    SET @cont_id = NEW.cont_id;
    UPDATE Content SET net_vote = net_vote + @val
    WHERE cont_id = @cont_id;
END;
```

```
CREATE TRIGGER update_netvote_after_delete
AFTER DELETE ON Vote
  FOR EACH ROW BEGIN
    IF OLD.vote THEN
      SET @val = -1;
    ELSE
      SET @val = 1;
    END IF;
    SET @cont_id = OLD.cont_id;
    UPDATE Content SET net_vote = net_vote + @val
    WHERE cont_id = @cont_id;
END;
```

```
CREATE TRIGGER update_netvote_after_update
AFTER UPDATE ON Vote
  FOR EACH ROW BEGIN
    IF NEW.vote THEN
      SET @val = 2;
    ELSE
      SET @val = -2;
    END IF;
    SET @cont_id = OLD.cont_id;
    UPDATE Content SET net_vote = net_vote + @val
    WHERE cont_id = @cont_id;
END;
```

5.1.2 Comment triggers

```
CREATE TRIGGER before_delete_comment BEFORE DELETE ON Comment
FOR EACH ROW BEGIN
    DELETE FROM Vote WHERE cont_id = OLD.cont_id;
END;
```

```
CREATE TRIGGER after_delete_comment AFTER DELETE ON Comment
FOR EACH ROW BEGIN
    DELETE FROM Content WHERE cont_id = OLD.cont_id;
END;
```

5.1.3 Post triggers

```
CREATE TRIGGER before_delete_post BEFORE DELETE ON Post
FOR EACH ROW BEGIN
    DELETE FROM Vote WHERE cont_id = OLD.cont_id;
    DELETE FROM Comment WHERE parent_post = OLD.cont_id;
END;
```

```
CREATE TRIGGER after_delete_post AFTER DELETE ON Post
FOR EACH ROW BEGIN
    DELETE FROM Content WHERE cont_id = OLD.cont_id;
END;
```

5.1.4 Category_Topic trigger

```
CREATE TRIGGER after_delete_topic_cattopic AFTER DELETE ON Category_Topic
FOR EACH ROW BEGIN
    DELETE FROM Post WHERE belongs = OLD.topic_name;
    DELETE FROM Topic WHERE topic_name = OLD.topic_name;
END;
```

5.1.5 Category trigger

```
CREATE TRIGGER before_delete_category BEFORE DELETE ON Category
FOR EACH ROW BEGIN
    DELETE FROM Moderator WHERE category_name = OLD.name;
    DELETE FROM Banned WHERE category_name = OLD.name;
    DELETE FROM Category_Topic WHERE category_name = OLD.name;
```

```
END;
```

5.2 Procedures

5.2.1 Register Admin

```
CREATE PROCEDURE register_admin
    (username VARCHAR(32),
    password VARCHAR(32),
    email_address VARCHAR(32),
    joined_date DATETIME ,
    admin_token VARCHAR(32))
INSERT INTO Admin (username, password, email_address, joined_date, admin_token)
VALUES (username, password, email_address, joined_date, admin_token);
```

5.2.2 Register Moderator

```
CREATE PROCEDURE register_moderator
    (username VARCHAR(32),
    password VARCHAR(32),
    email_address VARCHAR(32),
    joined_date DATETIME ,
    category_name VARCHAR(100))
INSERT INTO Moderator (username, password, email_address, joined_date, category_name)
VALUES (username, password, email_address, joined_date, category_name);
```

5.3 Reports

The following report calculates the number of posts and comments left by users in a given period of time. In this way it finds the most active users.

```
SELECT username, count(content) as activity_count
FROM Content
WHERE DATETIME(timestamp) BETWEEN '2017-11-08' AND '2017-25-012'
GROUP BY Content.username
HAVING activity_count > 1
ORDER BY activity_count DESC;
```

The following report measures the engagement of the users in the post. The engagement is measured by counting votes and comments of the post.

```
SELECT P.post_title, C1.net_vote + count(C2.cont_id) as engagement
FROM Content as C1, Post as P, Comment as C2
```

```

WHERE C1.cont_id = P.cont_id AND C2.parent_post = P.cont_id
      AND DATETIME(C1.timestamp) BETWEEN '2017-11-08' AND '2017-25-012'
GROUP BY P.cont_id
HAVING engagement > 2
ORDER BY engagement DESC;

```

5.4 Indexes

```

CREATE INDEX username_index USING BTREE ON User(username);

CREATE INDEX category_name_index USING BTREE ON Category(name);

CREATE INDEX topic_name_index USING BTREE ON Topic(topic_name);

CREATE INDEX post_title_index USING BTREE ON Post(post_title);

```

5.5 Views

5.5.1 Homepage

```

CREATE VIEW homepage_view AS
  (SELECT net_vote, C.cont_id, timestamp, content, content_type,
         username, post_title, post_type, belongs, category_name
   FROM Post AS P, Content AS C, Category_Topic AS CT
   WHERE P.cont_id = C.cont_id AND P.belongs = CT.topic_name
   ORDER BY net_vote DESC);

```

5.5.2 Best of last week

```

CREATE VIEW bestofea_week_view AS
  (SELECT max(net_vote) as net_vote, C.cont_id, timestamp, content, content_type,
         username, post_title, post_type, belongs, category_name
   FROM Post AS P, Content AS C, Category_Topic AS CT
   WHERE P.cont_id = C.cont_id AND P.belongs = CT.topic_name
         AND timestamp > now() - INTERVAL 1 WEEK
   GROUP BY category_name
   ORDER BY net_vote);

```

6 A User's Manual describing the usage of our system

6.1 Introduction

Welcome to Fara, the most awesome platform for sharing your thoughts with all the 'online' people, just like you! We have created this guide to help you gain acquaintance with this new and exciting platform!

6.2 For all users

6.2.1 Home screen

The Homepage is the first thing a user will be redirected to when facing the website. The homepage by default will contain the posts by the most number of votes they have collected. This applies to all time, while the user is offered with the choices to have various popular queries as explained below.

One of the most prominent features one comes across with in the homescreen page, is the navigation bar at the top, which is dressed in a funky color in all the rest of the pages. The navigation bar allows for quick navigation into the platform, whether it is on different categories, topics, users, setting etc.

The screenshot shows the Fara Home screen. At the top is a dark purple navigation bar with links: Fara, Home (active), Categories (dropdown), + Add Category, a search bar, and user status (Logged in as figali) with a settings gear and Log out link. Below the bar, the main content area is titled 'recent posts'. It features a table with columns: Votes, Title, Date, Category, Topic, and User. The table lists 10 posts, each with a vote count (1 or 0) and a vote icon. The first post has a 'comments' link. Below the table are three filters: 'Show 10 per page', 'Show 25 per page', and 'Show 50 per page'. At the bottom left, it says 'Pages: 1'.

Votes	Title	Date	Category	Topic	User
1	Who is your favorite footballer of all time?	56 minutes ago	Sports	Football	berku
1	Tips on getting A+ on biology	57 minutes ago	Science	Biology	figali
1	Basketball post comments	31 minutes ago	Sports	Basketball	berku
1	Football post	31 minutes ago	Sports	Football	dias
0	Who is the new coach of Galatasaray?	54 minutes ago	Sports	Football	berku
0	New hashing technique found!!	59 minutes ago	Science	Computer Science	figali
0	Thoughts on Besiktas and Bayern Muni	55 minutes ago	Sports	Football	berku
0	Last week of biology	31 minutes ago	Science	Biology	berku

Figure 2: Home screen

From the list of posts you can click on the title of the post. Then you will be redirected to see content of the posts and comments left under that post. If you click on a link post, you will be redirected to the url linked to the post. You can click on the comments link next to the title if you want to see the comments left under a link post.

6.3 View content

When you click on the title of the post or the comments link, you will be redirected to the page which displays the content of the post and comments posted under the post.

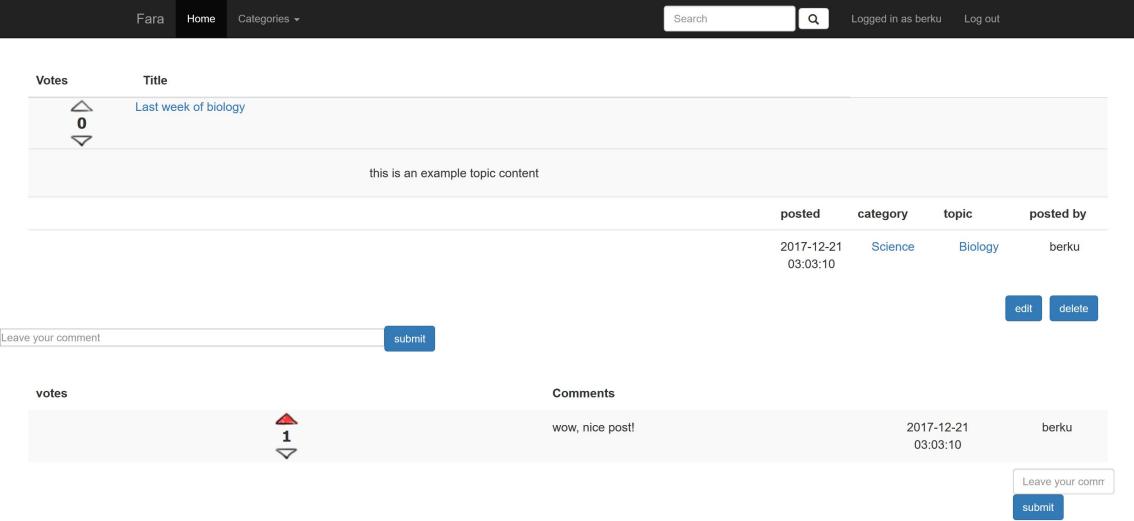


Figure 3: View of the content

6.3.1 Search

Search is a feature provided to all users of Fara, registered already or mere guests. Search is performed on the usernames, on posts titles, topics names as well as category names. The respective results come up in the search page. Again, the navigation bar, this essential feature, is used for the actual searching procedure.



Figure 4: Search bar

Search is done on:

- Username
- Category names
- Topic names
- Post titles

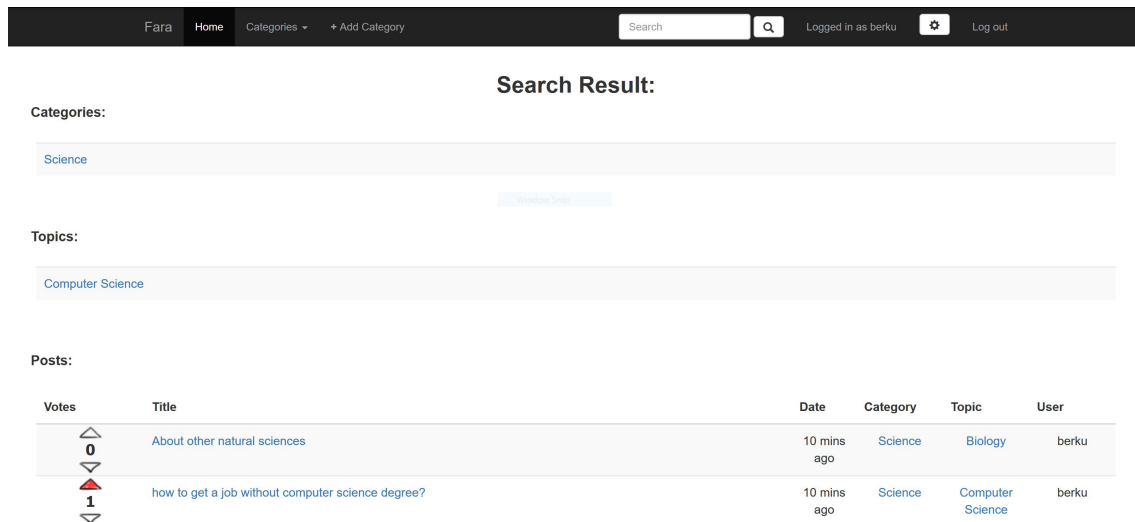


Figure 5: Sample search result

6.3.2 Category screen

In category screen one would see the respective posts of a certain category. Users can choose the category from the navigation bar, and when redirected to respective page, users can see these posts of the category, ordered by net vote count, or popularity as some might call it.

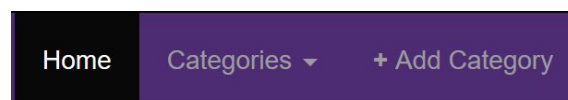


Figure 6: Category bar

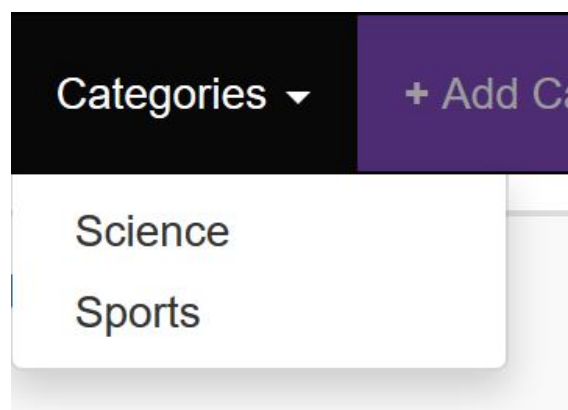


Figure 7: Categories dropdown

Fara

Home

Topics + Add Topic

Search

Q

Logged in as figali

⚙

Log out

Science

Top posts of today

Top posts of all week

Top posts of all

Votes	Title	Date	Category	Topic	User
<div><div>▲</div><div>1</div><div>▼</div></div>	Tips on getting A+ on biology	47 minutes ago	Science	Biology	figali
<div><div>▲</div><div>0</div><div>▼</div></div>	Last week of biology	40 minutes ago	Science	Biology	berku
<div><div>▲</div><div>0</div><div>▼</div></div>	Here is why biology should be taught to everyone	55 minutes ago	Science	Biology	figali
<div><div>▲</div><div>0</div><div>▼</div></div>	New hashing technique found!!	49 minutes ago	Science	Computer Science	figali
<div><div>▲</div><div>0</div><div>▼</div></div>	My journey of getting into bioinformatics	54 minutes ago	Science	Biology	figali

Show 10 per page

Show 25 per page

Show 50 per page

Pages:

1

Stop being moderator

Figure 8: Topic screen

6.3.3 Topic screen

In topic screen, much like category screen, the user can see this time, the topics of the specific category. The functionality and interface provided resembles the previous pages very much, which we hope provides a homogeneous interface the user feels comfortable in.

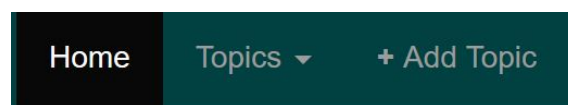


Figure 9: Topics bar

6.3.4 User Profile

The user profile belongs to all users, specifically it is helpful when a user might want to message

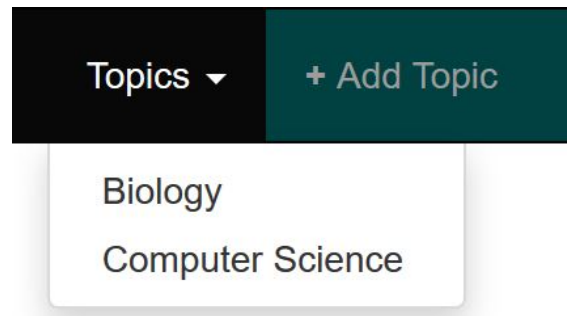


Figure 10: Topic dropdown

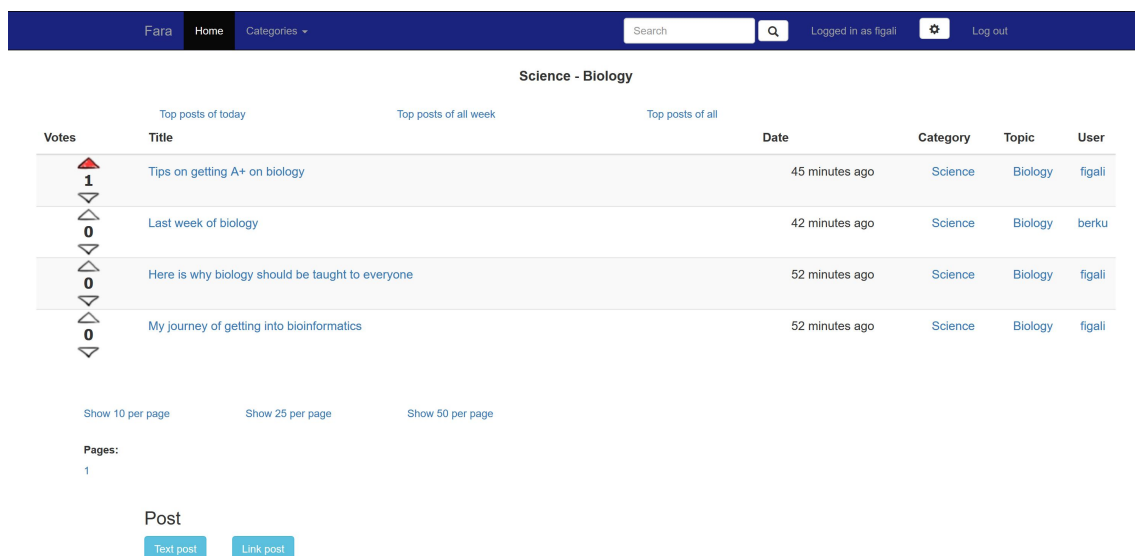


Figure 11: Topic screen

6.4 How to register

In order to have access to more special features of the website users can create a personal account through Sign up page. This a simple and user-friendly page, user needs to provide a unique email and username and a password in order to join. After checking the validity of the information entered, such as if email address is valid and if the user already exists user will be registered and will be redirected to Login page. The informations provided by user in this page are inserted into User table in database. Below we have provided a picture of how SignUp page looks.

6.5 Login

All register users can login into the system through this simple Login page. In order to login the unique username and password are required. The validity of both password and username is checked before user is logged in. After pressing the Log in button user is redirected to homepage.

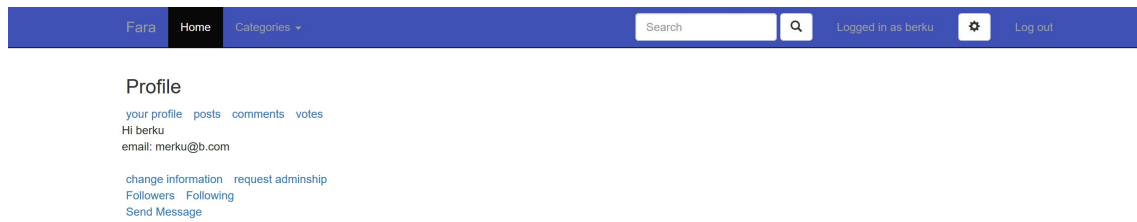


Figure 12: User Profile Page

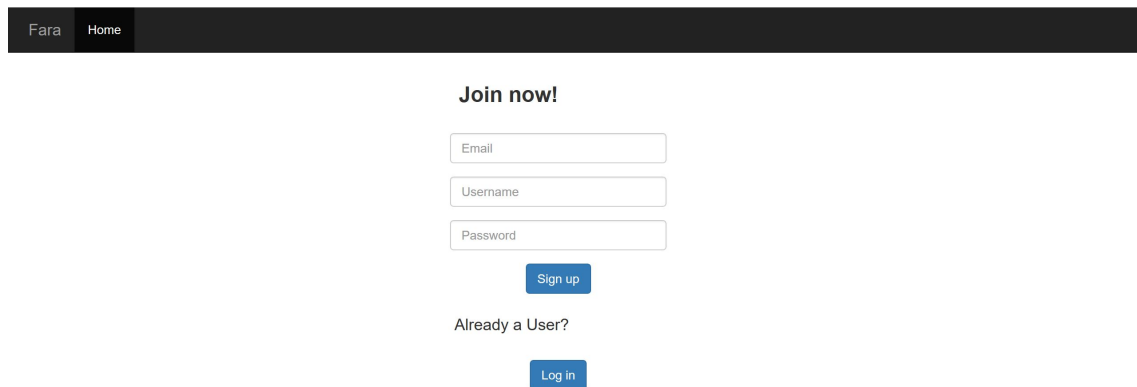


Figure 13: Signup page

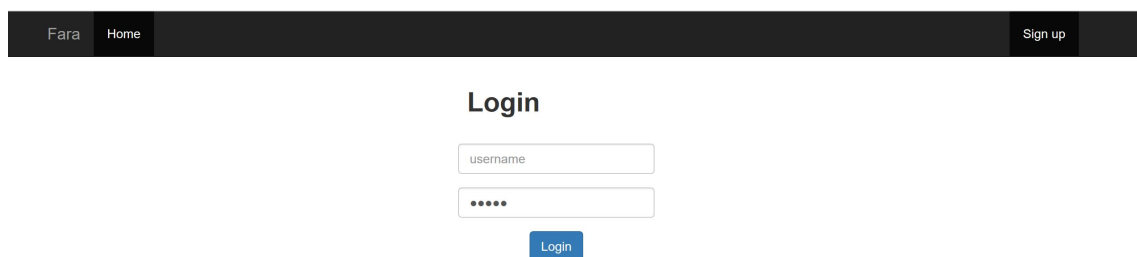


Figure 14: Login page

6.6 Homepage

For all registered users, homepage will be the main page. In the Homepage they can see a list of most recent posts along with detailed information about each post, such as date, category, topic and username of the user who have posted a specific post. The register user will be able to open

other user's profile or a specific category by clicking on the specific name. Additionally, through homepage user will be able to create a new category, search for a specific category or user, we have provided a more detailed explanation of these features below.

6.7 Profile and user settings page

In the profile and user settings page, the user can navigate to through the special settings icon button near the top left corner. The user can preview their activity in this page, in terms of posts, comments or votes they have done in the past. The user can click on these and see a detailed view of their past activity, much similar to the previous previews of Fara, for much ease for the user. Additionally this is where the user can find the option of changing their username and password by clicking on the change information link provided.

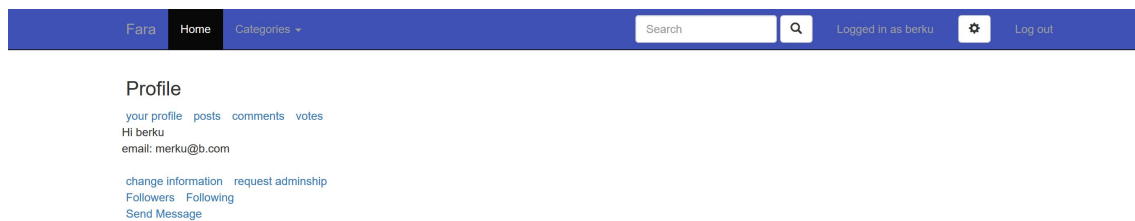


Figure 15: View User

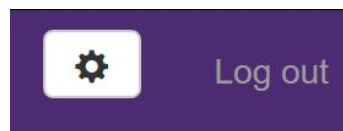


Figure 16: Settings and logout buttons

6.8 Request adminship and moderatorship

If a user wants to become an admin, he/she should go to their user page and click on Adminship request. After this a token will be generated and emailed to the user. Then a new page will come up as shown in Figure 17. The user will enter the token they received and find out if they get granted with admin access.

If a user wants to become a moderator, they should go into the category they want to be moderator for and click on request moderatorship button below. A user can be a moderator if they have been active more than 3 years and if they have lots of upvoted posts in the category they chose.

[Fara](#)
[Home](#)
[Categories](#)

Logged in as berku

[Log out](#)

Adminship request

Enter token received in your email address:

Figure 17: Request being an administrator

[Fara](#)
[Home](#)
[+ Add Topic](#)

Logged in as berku

[Log out](#)

Sports

Top posts of today		Top posts of all week		Top posts of all		Date	Category	Topic	User
1	Football post					48 minutes ago	Sports	Football	berku
2	Who is your favorite footballer of all time?					21 minutes ago	Sports	Football	berku
3	Who is the new coach of Southampton?					23 minutes ago	Sports	Football	berku
4	Thoughts on Darwin and Bayesian Models					23 minutes ago	Sports	Football	berku

[Show 10 per page](#)
[Show 25 per page](#)
[Show 50 per page](#)

Pages: 1

Figure 18: Request being a moderator

6.9 Admin functionalities

6.9.1 Delete category

An admin can delete a category. After this category is deleted all the posts, comments, topics of that category are also removed. An admin can go to any category and click on Delete Category to delete.

[Fara](#)
[Home](#)
[Topics](#)
[+ Add Topic](#)

Logged in as berku

[Log out](#)

Science

Top posts of today		Top posts of all week		Top posts of all		Date	Category	Topic	User
1	how to get a job without computer science degree?					51 minutes ago	Science	Computer Science	berku
0	Last week of biology					22 minutes ago	Science	Biology	berku
0	Here is why biology should be taught to everyone					46 minutes ago	Science	Biology	figali
0	New hashing technique found!!					52 minutes ago	Science	Computer Science	figali
0	My journey of getting into bioinformatics					47 minutes ago	Science	Biology	figali

[Show 10 per page](#)
[Show 25 per page](#)
[Show 50 per page](#)

Pages: 1

Figure 19: Delete a category

6.9.2 Delete user

An admin can delete a user. After the user is deleted, their comments, posts and votes stay in the system but user will not be able to log in to their accounts anymore. An admin can go to view profile of the user and click Delete User to prevent their access to their account.

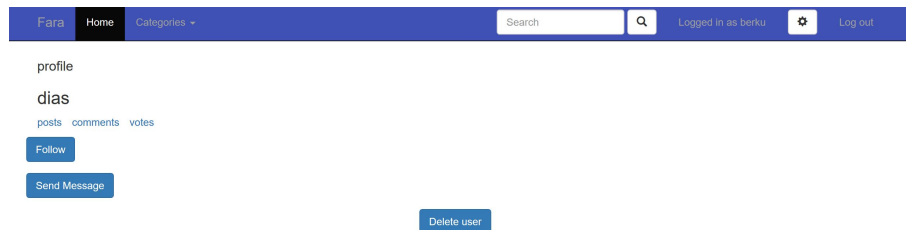


Figure 20: Delete a user

6.10 Moderator functionalities

6.10.1 Delete post

A moderator can delete posts in the category that he/she moderates on. In order to do it, moderator should go in post and click on Delete Post.

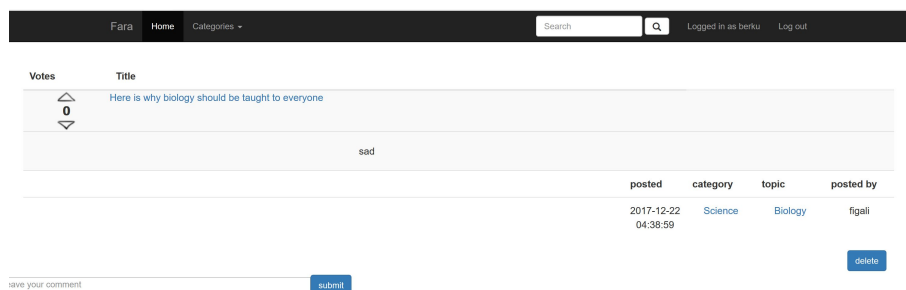


Figure 21: Delete a post

6.11 Adding posts

Users are able to post two different kind of post and text post and link post. In order to post a new post to a topic they want, they need to go to the view of topic and choose whether they want to post a text post or a link post.

6.11.1 Add a text post

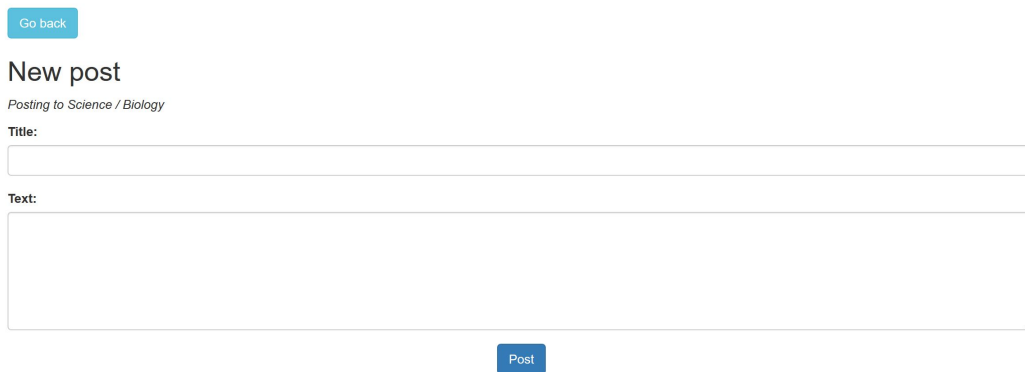
The user should provide a title to their post and a text. Then after clicking on post, the post will be posted to the respective topic.

Post

Text post

Link post

Figure 22: Post buttons

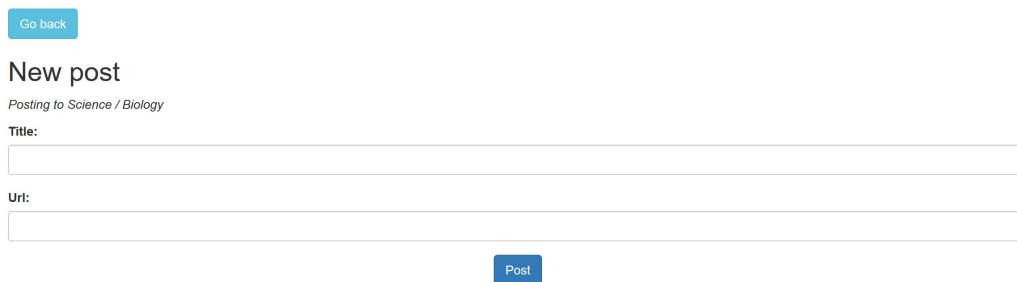


The screenshot shows a web interface for creating a new post. At the top left is a 'Go back' button. Below it is the heading 'New post' followed by the text 'Posting to Science / Biology'. There are two input fields: 'Title:' and 'Text:'. The 'Text:' field is a larger text area. At the bottom right is a 'Post' button.

Figure 23: Add a text post

6.11.2 Add a link post

A special type of post are links. Users are able to post just a link as a post. They need to provide a title and the URL of the link and through the simple form provided below they can successfully post a link.



The screenshot shows a web interface for creating a new link post. At the top left is a 'Go back' button. Below it is the heading 'New post' followed by the text 'Posting to Science / Biology'. There are two input fields: 'Title:' and 'Uri:'. At the bottom right is a 'Post' button.

Figure 24: Add a link post

6.12 Create new categories

Register users are able to add a new category. This can be done by clicking the "Add Category" button of Homepage which will lead to a simple and user friendly window where user can enter the desired topic and press "Add" button in order to submit it.

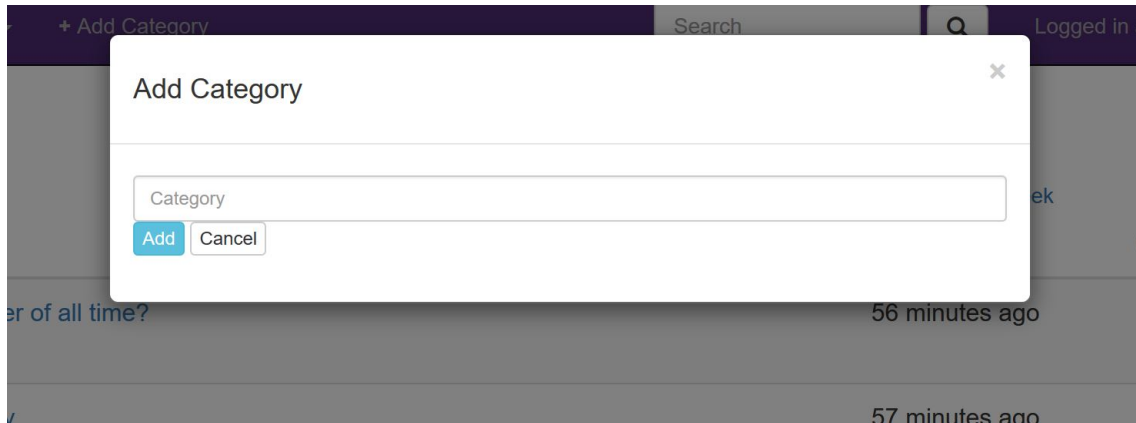


Figure 25: Adding a new category

6.13 Create New Topic

Adding a new topic will be very similar to adding a new category. User just needs to enter the desired topic to the respective field and click "Add" button.

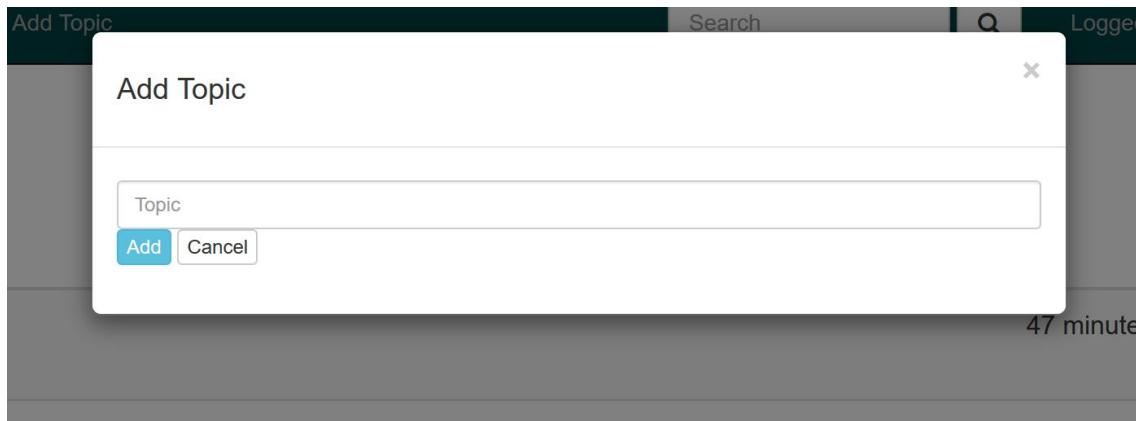


Figure 26: Adding a new topic

6.14 Follow other users

Users can follow each other through the "Follow" button that is on every user profile. Additionally, users can check the list of their followers and people they follow through links that are provided on their personal profile page as it can be seen from Figure 27.

6.15 Message other users

Users can communicate with each other through private messages. There is a button on each user's profile and by clicking that they will be redirected to send message page, on which user needs to provide the user name of the person to whom he/she wants to send a message and a

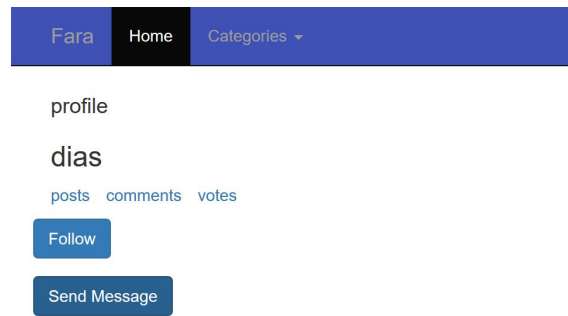


Figure 27: Following other users.

text message. Sending an empty message is not allowed, so the validity of the message and of the username of the recipient is checked before sending the message.

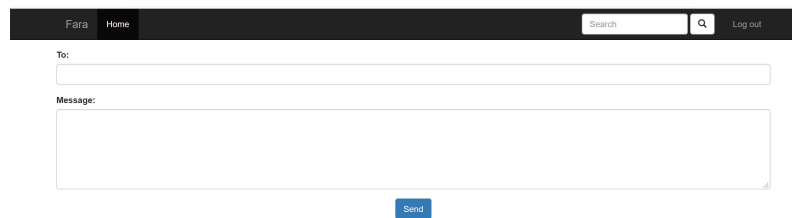


Figure 28: Messaging other users.