

# Technical and manual

<b>Group members:</b>	<b>1</b>
<b>Installation process</b>	<b>2</b>
In case of error messages during or after composing:	2
The client side requires:	2
The server side requires:	2
<b>Validation of input:</b>	<b>4</b>
User validation	4
Profile picture validation	4
Create forum validation	4
Test-Data:	4
<b>Using the website</b>	<b>4</b>
Use of the website	5
Register a new user	5
Login to the forum	6
Login with another user	6
Changing profile picture or password	7
Request Moderator as user	7
Accept or decline moderator request	8
See all users:	8
See blocked posts / comments:	9
Forum navigation	10
Navigate to a forum	10
Forum search:	12
See and comment on a post:	12
Delete or block comment or post:	14
Sort on post and comments	15
Create a new forum	16
Create a new post	16
See useractivity	18

## Group members:

- Oddbjørn S. Borge-Jensen - [oddbjosb@stud.ntnu.no](mailto:oddbjosb@stud.ntnu.no)
- Odd Hald Bliksås - [oddhb@stud.ntnu.no](mailto:oddhb@stud.ntnu.no) - Phone 45247790
- Nicholas Bodvin Sellevåg - [nicholbs@stud.ntnu.no](mailto:nicholbs@stud.ntnu.no)
- Wojciech Kurp

## Installation process

All dependencies for the project are easily accessible through two package.json files. Namely client/package.json and server/package.json.

The installation process consists of three steps: (the same steps that is provided from the start of the project)

1. In the root folder of the project, **prog2053-ny**, open a terminal and navigate to the **client** folder and run the command: ***npm i***
2. Navigate back to **prog2053-ny**, and step into the **server** folder. Yet again run the command ***npm i***. After this step all dependencies will be applied to your local repository.
3. Finally, the application is built and initialized through ***docker-compose up -d*** while being in the root folder **prog2053-ny**.

Please read the next chapter if some problems occur during the installation

## In case of error messages during or after composing:

These steps should only be followed in the case of a misfired *docker-compose*. The leading paragraphs explains in detail how to manually install all required packets/dependencies from our project.

The client side requires:

**NB!** Navigate to the “client” folder inside the project.

### **Vaadin router**

- We are using “Vaadin router” for front-end routing.
- **Installation:**
  - a. Run the command: ***npm install @vaadin/router --save***

## The server side requires:

**NB!** Navigate to the “server” folder inside the project.

### **MYSQL:**

- We are using the npm package “mysql” to communicate with our database
- **Installation:**
  - a. Run the command: *npm install mysql --save*

### **MULTER:**

- We are using the npm package “multer” for decoding formdata, in addition to uploading images.
- **Installation:**
  - a. Run the command: *npm install multer --save*

### **CORS:**

- We are using the npm package “cors” for authentication and verifying that requests to the back-end follows a rule set.
- **Installation:**
  - Run the command: *npm install cors --save*

### **FS**

- We are using the npm package “FS” for file handling.
- **Installation:**
  - a. Run the command: *npm install fs --save*

### **Randomstring:**

- We are using the npm package “randomstring” for generating random file names when uploading pictures.
- **Installation:**
  - a. Run the command: *npm install randomstring --save*

### **Express-session**

- The npm package “express-session” is in coherence with cookie handling
- **Installation:**
  - a. Run the command: *npm install express-session --save*

### **Bcryptjs**

- The npm package “bcryptjs” is used to create hashed and “salted” passwords, in addition to comparing hashed passwords for authentication.
- **Installation:**
  - Run the command: *npm install bcryptjs --save*

### **Cookie-parser**

- The npm package “cookie-parser” is used for handling cookies.
- **Installation:**
  - `npm install cookie-parser --save`

### Body-parser

- The npm package “bodyparser” is used as middleware to parse some requests.
- **Installation:**
  - `npm install body-parser --save`

## Validation of input:

### User validation

#### email-address:

An email address can consist of: `xxx [a-z0-9._%+~]+@[a-z0-9.-]+\.[a-z]{2,63}$`

#### username:

A username can consist of the characters (both capital and small letters) a-z , digits and the length can be from 2 to 63 characters

#### Password:

The password must be at least 8 characters long.

### Profile picture validation

- A profile picture can only be a png or jpg file.

### Create forum validation

A forum name can consist of the characters (both capital and small letters) a-z , digits and the length can be from 2 to 63 characters

## Test-Data:

We are providing 3 users which covers all use-cases of our application.

username	e-mail	password	usertype
Alpha	alpha@domain.com	Password1	admin
Bravo	bravo@domain.com	Password2	moderator
Charlie	charlie@domain.com	Password3	user

# Using the website

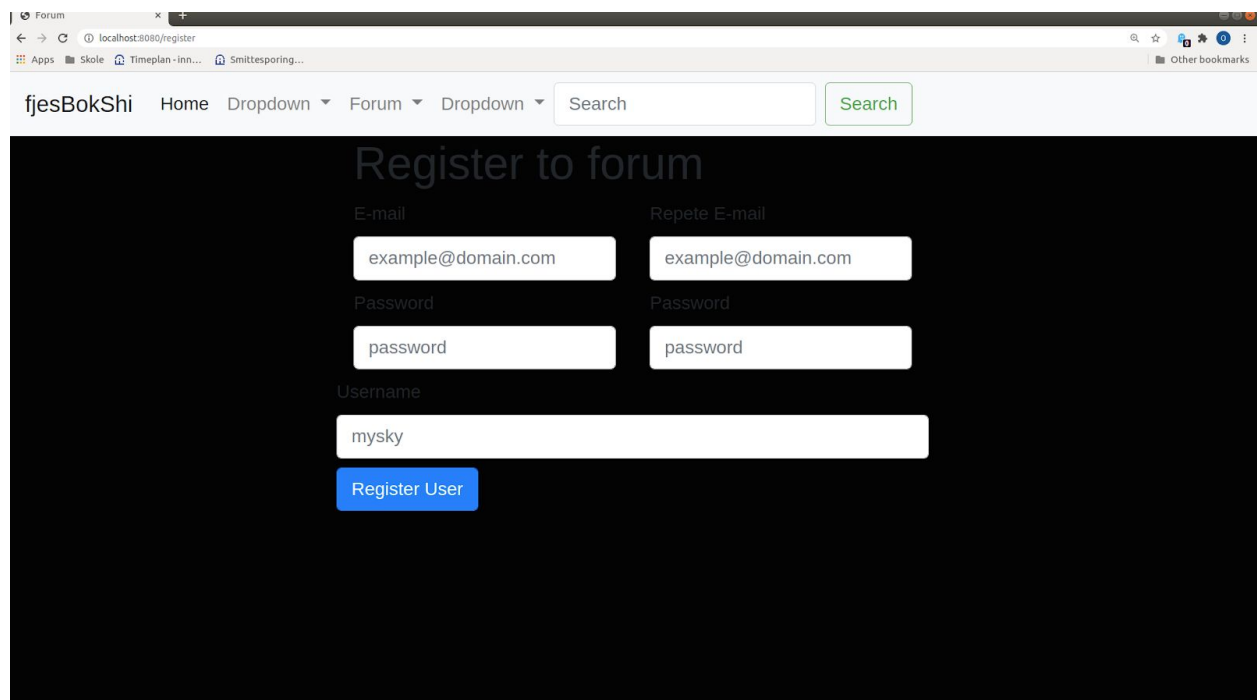
Homepage: <http://localhost:8080/>

Database: <http://localhost:8082/>

## Use of the website

### Register a new user

1. From the home page click on “First step is to register an account!”
2. You are directed to the register page:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/register'. The page title is 'Register to forum'. The form includes the following fields and elements:

- E-mail:** Input field containing 'example@domain.com'.
- Repete E-mail:** Input field containing 'example@domain.com'.
- Password:** Input field containing 'password'.
- Username:** Input field containing 'mysky'.
- Register User:** A blue button to submit the registration form.

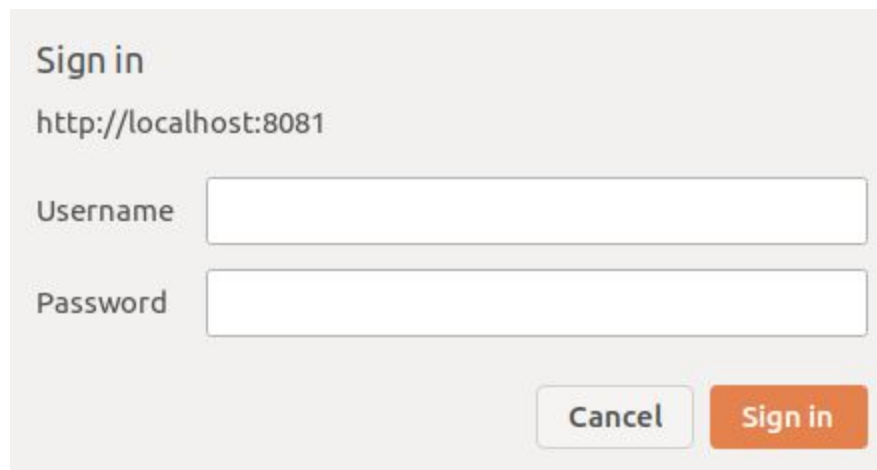
The browser's address bar also shows 'localhost:8080/register'. The page has a dark background and a light-colored header with navigation links: 'fjesBokShi', 'Home', 'Dropdown', 'Forum', 'Dropdown', and a search bar with a 'Search' button.

3. It is noteworthy that validation of input is implemented when registering a new user
4. Click register when you have filled all the data necessary to create and register a new user which can be viewed inside the database.
5. Based on the input you can get different results back: (It should pop up a alert box with the message)
  - a. **User registered!** = user is registered correctly

- b. **Username exists!** = username already exist
- c. **User exist** = the email address already exist
- d. **input value not valid username, password mismatch** = the repeated email or password doesn't match each other
- e. **Username character not valid** = Invalid characters in username
- f. **Invalid characters in Email** = Email characters invalid

## Login to the forum

1. We have chosen to implement a continuous authorization check into our application. In practice, you will be asked to login whenever you try to access a web-resource with content and functionalities depending on user privileges.
2. Naturally, a user is able to sign in from the home page. Simply click on “Secondly you can log into your account!” and a pop-up will appear.



Sign in

http://localhost:8081

Username

Password

3. Alternatively, the user can also sign in from the dropdown menu **User Info => login**
4. Add your username (**NOT your email**). One should note that the input is case sensitive for both your username and password.
5. Consequently after filling out the form click on “Sign in”.
6. Mismatching data between user input and database registers prohibits the user from gaining access to the web resource.
7. With a successful login, the user is redirected to the homepage.

## Login with another user

- For changing the user please delete the cookie in your web browser (the cookie name is user) then:
  1. Close your web browser and open the web browser again
  2. Navigate to the homepage: <http://localhost:8080/>
  3. Sign in again with the desired user.

Application											
Filter											
Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Priority		
user	s%3Aadmin%2015plmpbSG/CVK7R1H4uZ2eZ7YpZ3AWC3mOfCEw	localhost	/	Session	61				Medium		

Forum

localhost:8080/settings

Apps Skole Timeplan - inn... Smittesporing...

fjesBokShi Home User Info Forum Administer Search

Search

## Upload Picture

Upload Picture

Choose File No file chosen

set Profile Picture

## Change Password

Password new password

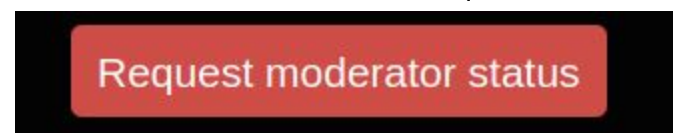
Change Password

1. Choose "User Info" from the navbar and click on "settings".

- Click on “choose file” and select the picture to be uploaded as your profile picture.
- Click on “set profile picture” to confirm your selection which updates the user database. **NB!** only *jpg* or *png* format of pictures are allowed.
- NB it is a bug in the fileupload, Be sure to have uploaded a picture before pushing the button set profile picture else something terrible will happen in the backend / the picture name will be set to undefined.

1. Type your new password in the password field
2. Click on button "Change password"

1. Click on the navbar on 'Administer' and then on request
2. Click on the red button "Request moderator status" as below:

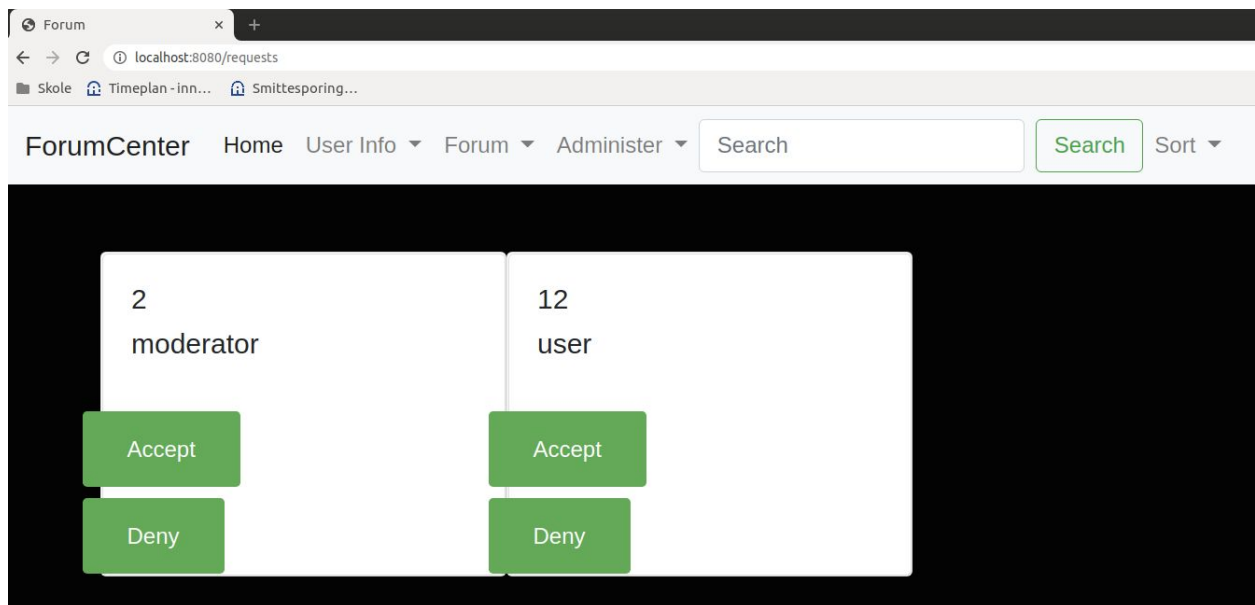


3. A message should pop up saying that: "your request has been noticed!"

## Accept or decline moderator request

You have to be logged into the system as administrator for accepting or decline moderator requests

1. On the navbar click on Administer and then on requests
2. You should see a list on all requests like the picture above:



3. Click what you want, Accept for moderator and Deny for denying the request. The number is the users UID.

## See all users:

You have to be logged in as moderator or administrator

1. Click on the nav bar -> Administer -> List of users
2. A list should show up with all users:



← → ↻ localhost:8080/userList

Skole Timeplan - inn... Smittesporing...

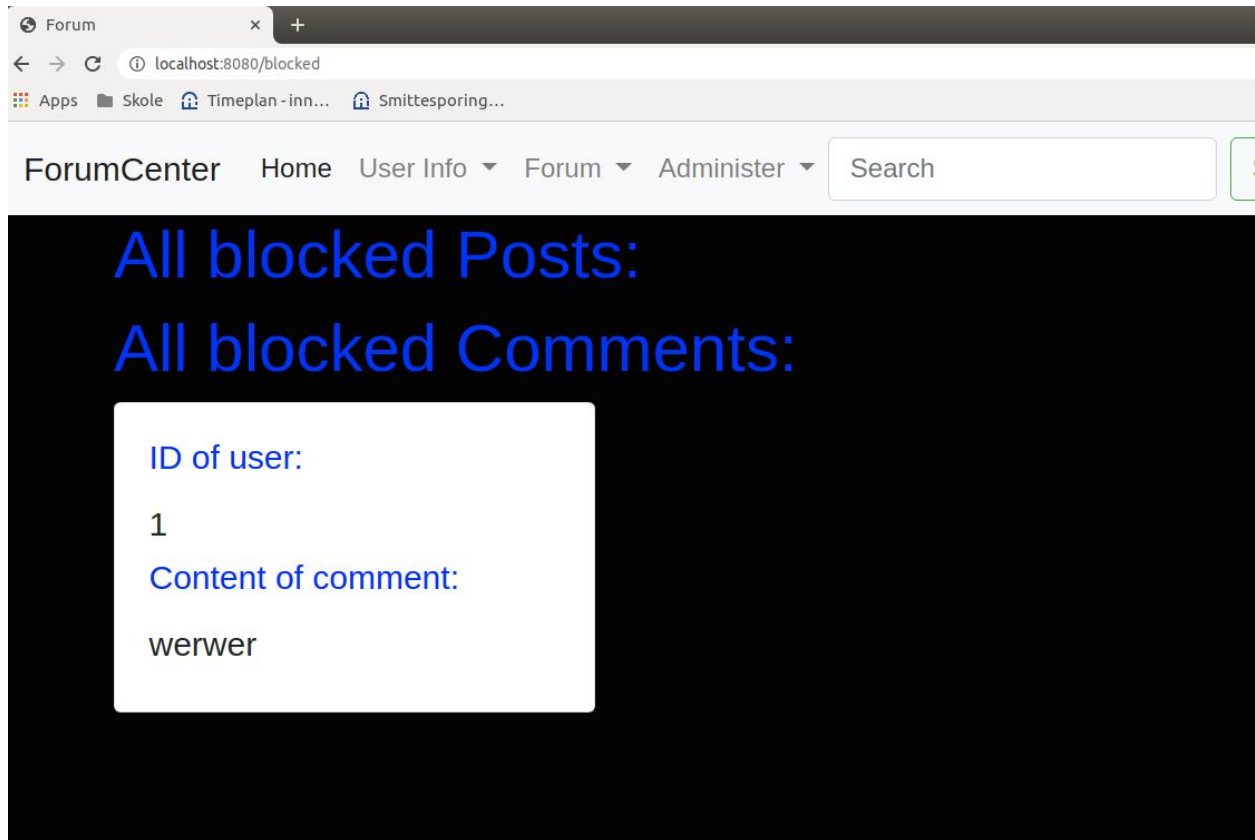
ForumCenter Home User Info ▾ Forum ▾ Administer ▾ Search Search Sort ▾

test admin	user	user
Alpha admin	Bravo moderator	Charlie user
delta user	papa moderator	digi moderator
demo user		

See blocked posts / comments:

You have to be logged in as administrator ore moderator

1. From the navbar click on administer -> blocked
2. A list of blocked content is listed, See screenshot above:

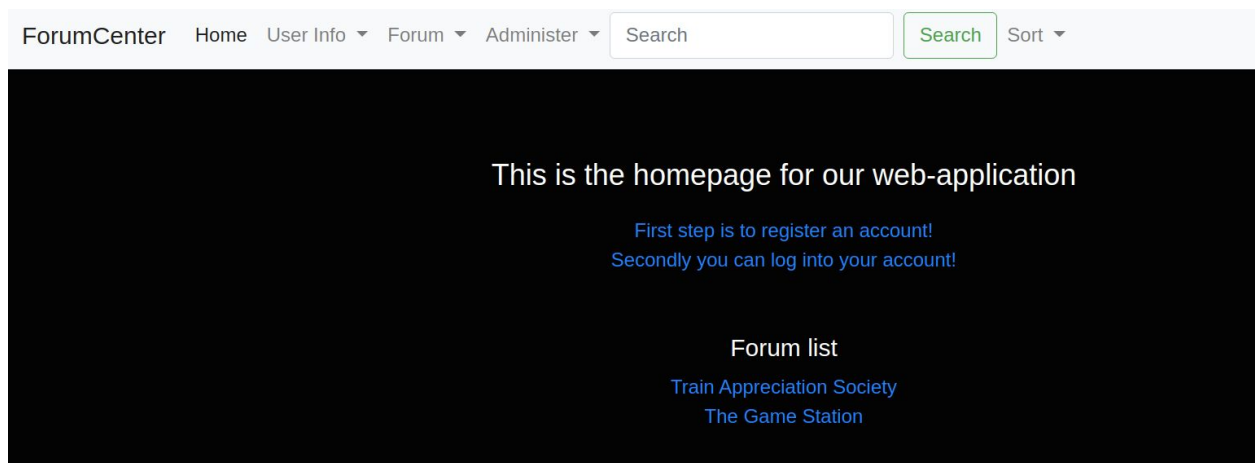


## Forum navigation

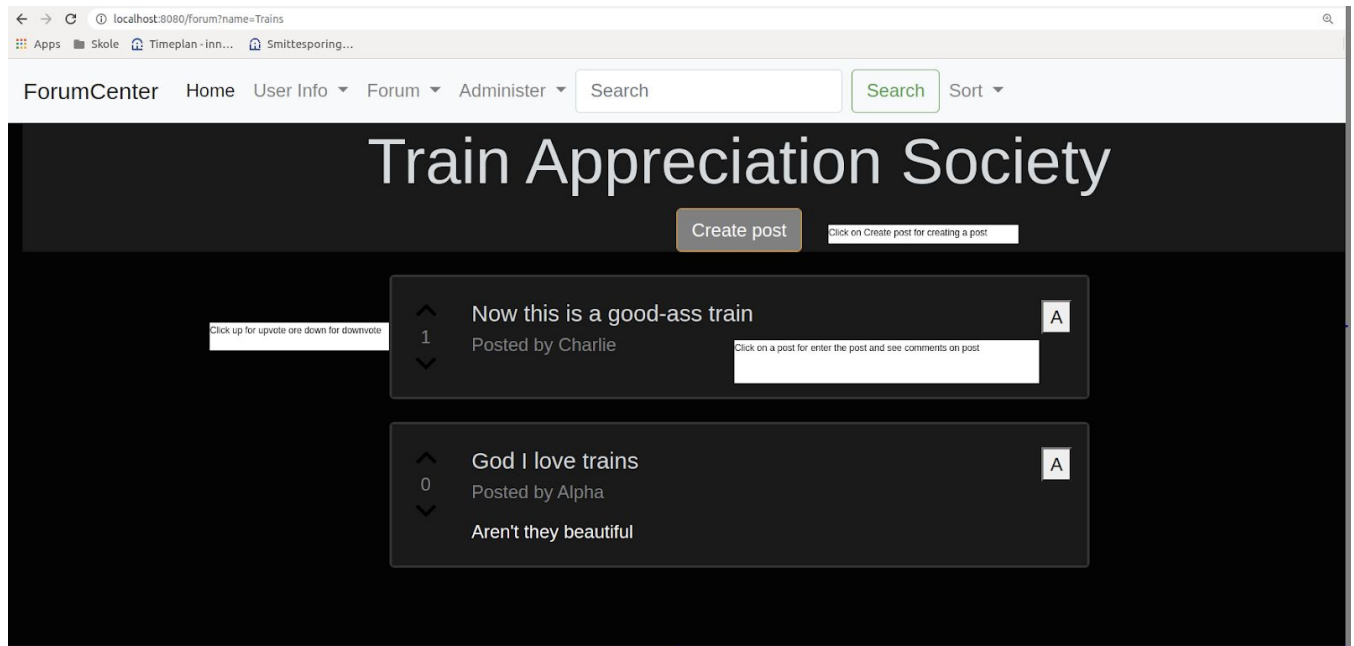
You have to be logged in to use the form functions properly

Navigate to a forum

From the homepage under Forum list, click on desired forum:



From the main site on a specific forum:

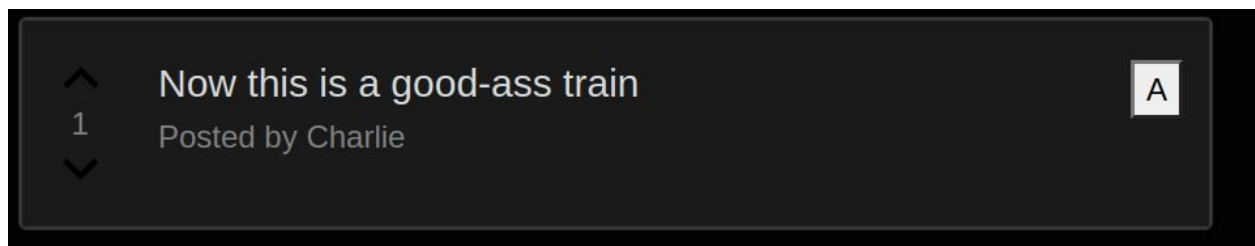


From the specific forum you can do multiple actions as described in the picture above with the textboxes in.

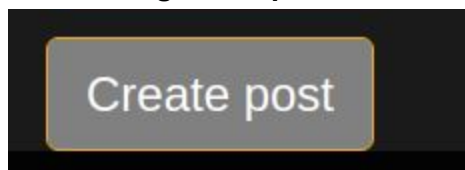
**For up or down vote click on the arrows:**



**For entering a specific post click in the post box:**



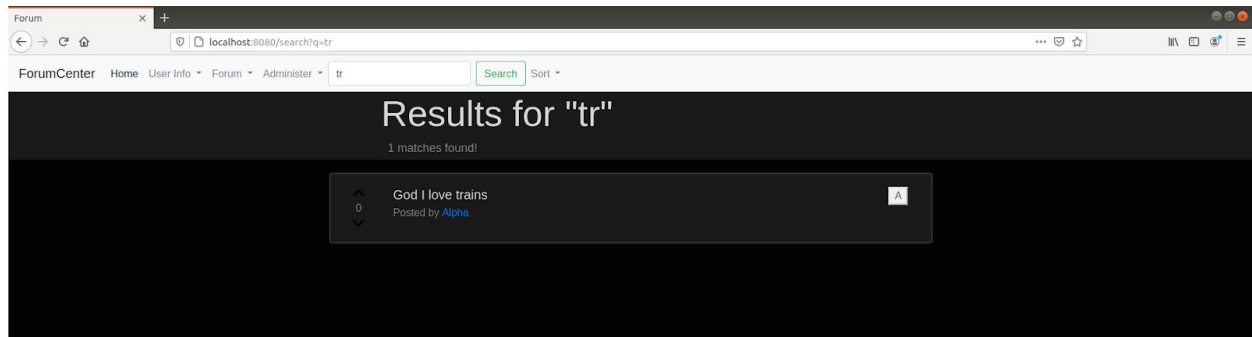
**For creating a new post click on the button "Create post"**



Forum search:

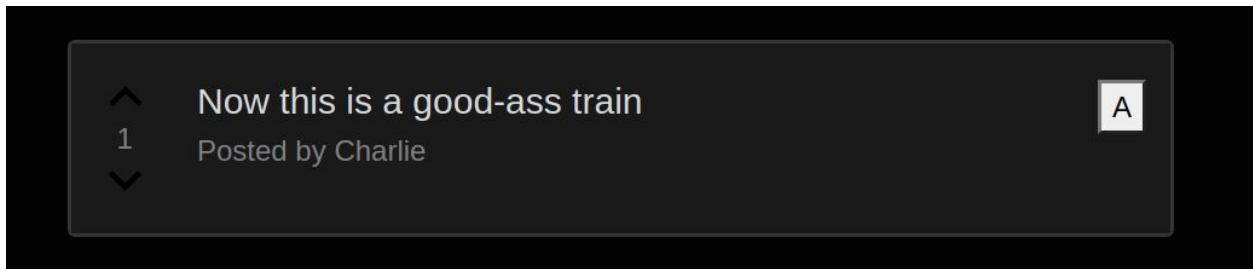
Known bugs: please click on Search button, do not use enter button for submitting the search

1. Please be signed in before use the search
2. In the navbar type what you want to search for ant then click Serach
3. The search result is displayed as shown in the screenshot below:

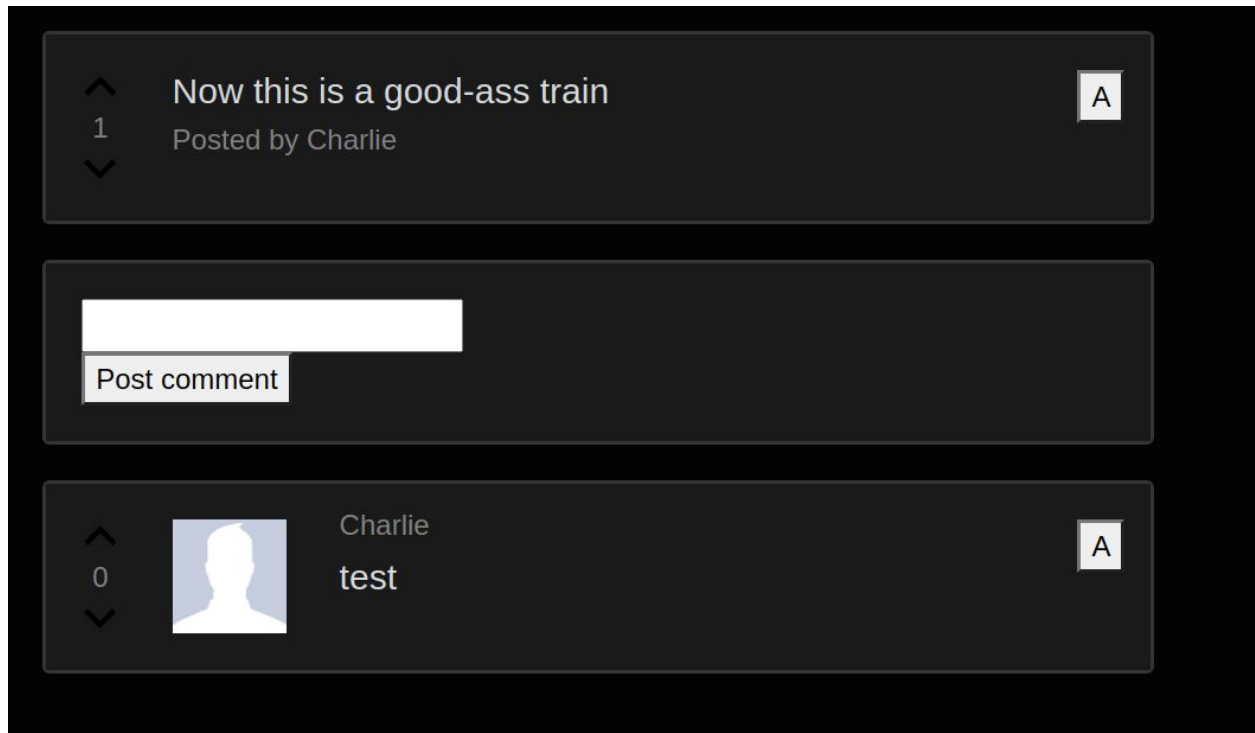


See and comment on a post:

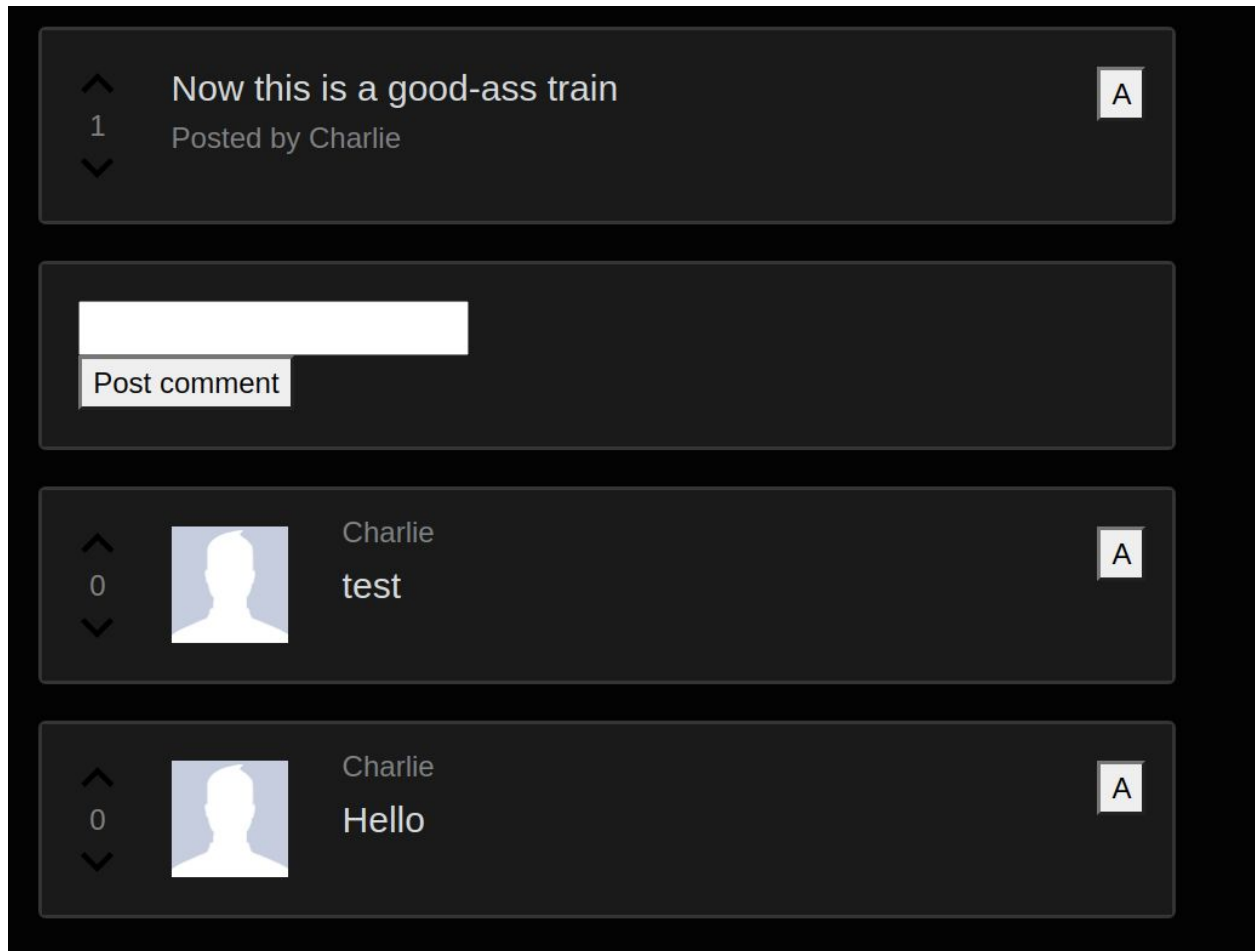
From the specific forum page click on the specific post you want to see or comment on as shown below:



When you have clicked on the specific post you are directed to the post with comments:

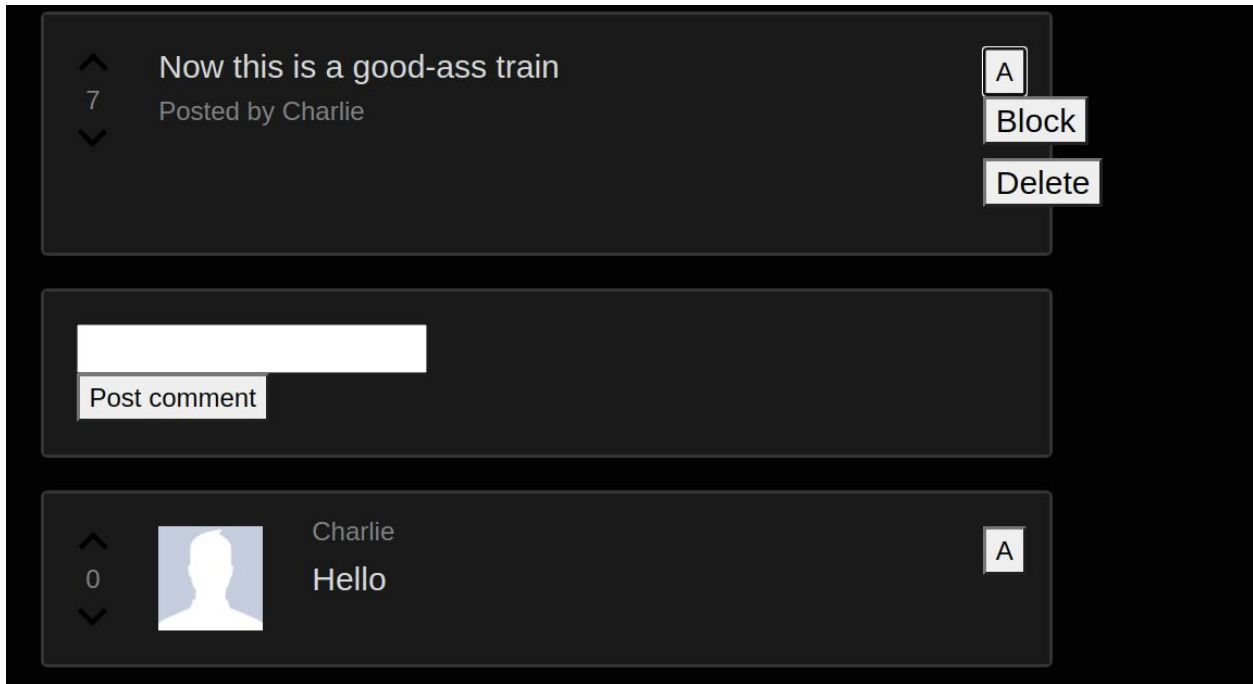


Here you can see already that user Charlie have comment on the post. If you want to comment just type your comment in the white field and click on post comment. The comment will be placed on the bottom as shown in the picture bellow:



Delete or block comment or post:

1. Go to the post or comment you want to delete
2. Click on the A and select delete or block:

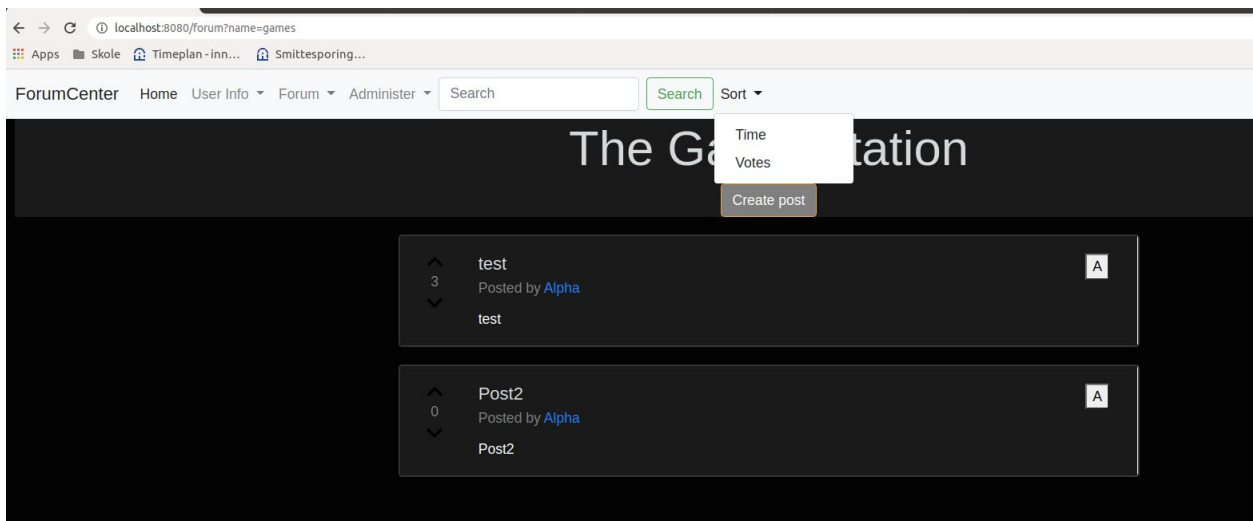


3. Chose Block for blocking ore delete for deleting the post ore comment

## Sort on post and comments

You can sort comments and post by time or votes.

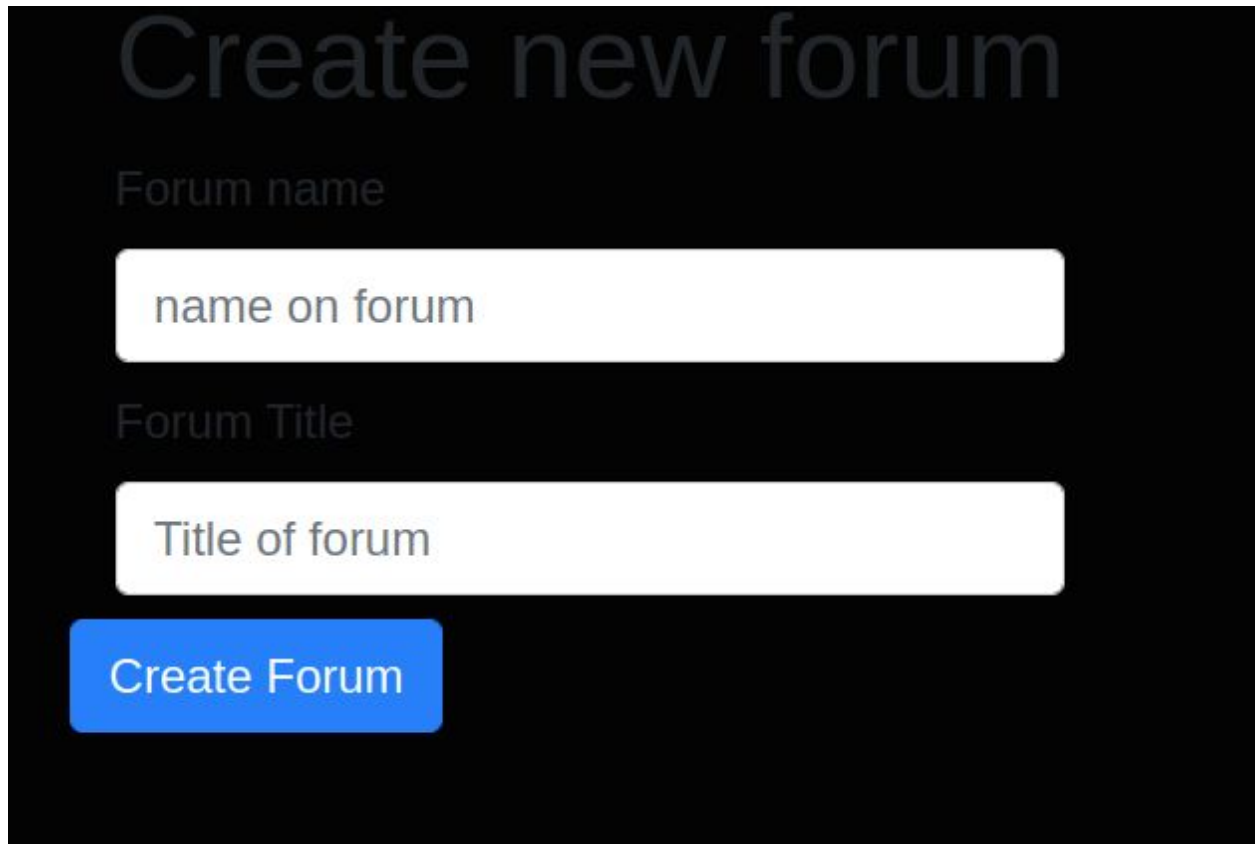
1. In a forum ore inside a post
2. Click on sort located to the right in the nav bar
3. Chose Time or Votes:



## Create a new forum

You have to be signed in to create a new forum.

1. From the homepage click on Forum located on the navbar and chose "Create forum"

A screenshot of a web form titled "Create new forum" on a dark background. The form has two input fields: "Forum name" with the placeholder text "name on forum" and "Forum Title" with the placeholder text "Title of forum". Below these fields is a blue button labeled "Create Forum".

Create new forum

Forum name

name on forum

Forum Title

Title of forum

Create Forum

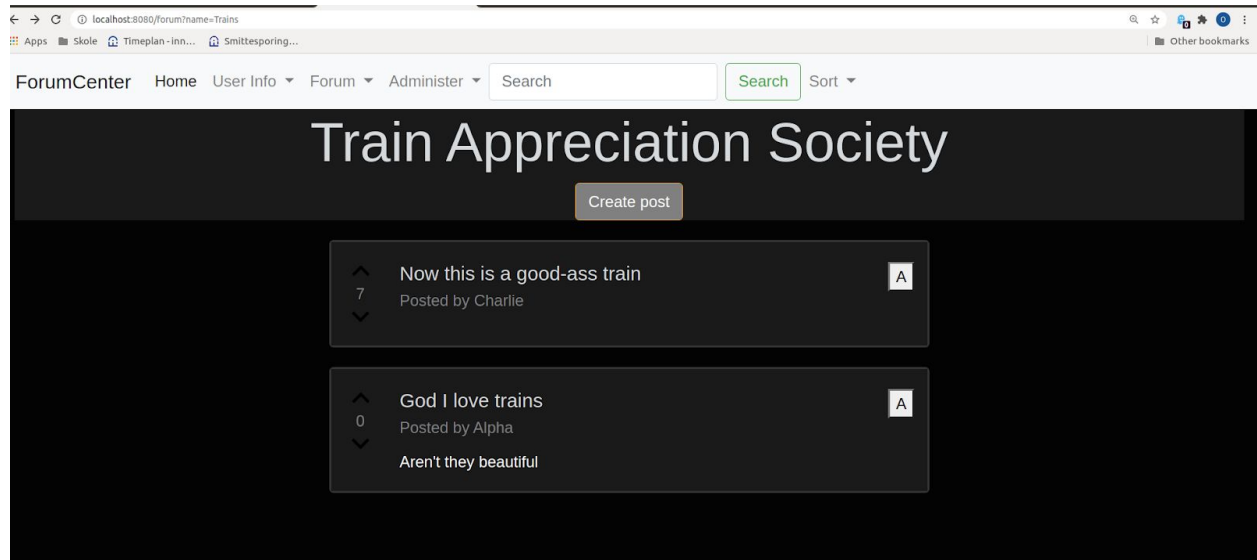
2. Enter desired Forum name and Title and then press "Create forum"
3. A box should pop up:
  - a. Forum Created = Forum created successfully
    - i. User is then redirected to the home page
  - b. Forum name already exist, please choose another one! = forum name already exist
  - c. Title or name character not valid, can only contain 0-9,A-Z,a-z = Invalid character

## Create a new post

You have to be signed in to the system for creating a new post

1. From the forum page where you want to create a new post, click on the button Create post





2. Fill in Title, content, then click on Create post:

3. Upon successful creation user is redirected to the relevant forum page

# Train Appreciation Society

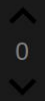
Create post



Now this is a good-ass train

Posted by Charlie

A



God I love trains

Posted by Alpha

Aren't they beautiful

A



mypost

Posted by Alpha

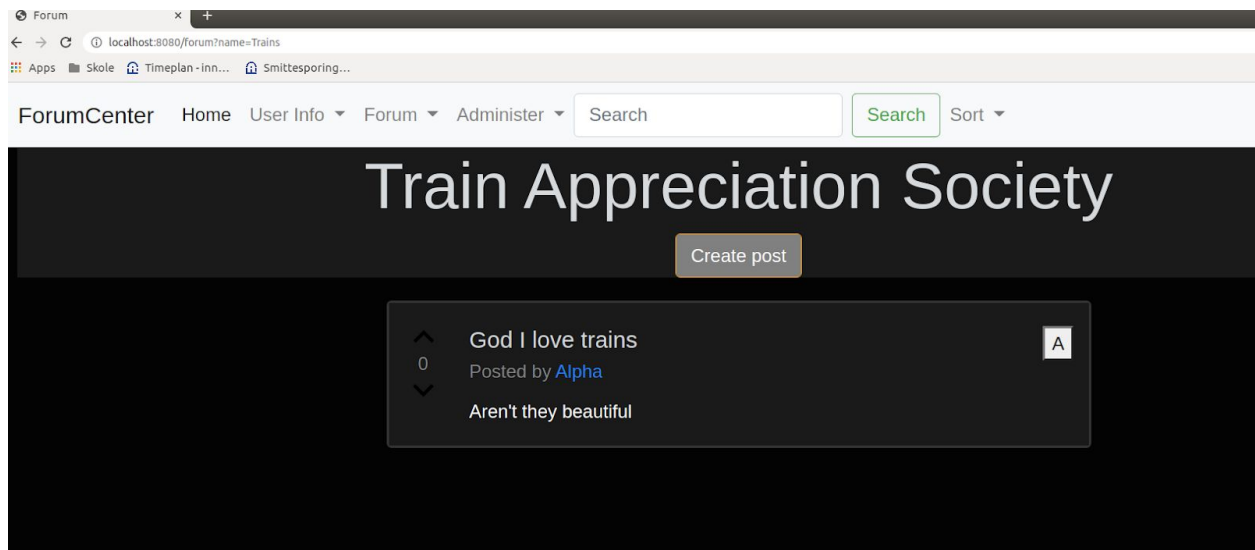
mycontentPost

A

## See useractivity

Accessing user activity:

1. Click on username on a post
2. Click on username on a comment
3. As admin: Click on username in user list



4. After clicking on the username, the user activity should appear as shown below:

