Lappeenrannan teknillinen yliopisto March 5, 2023
CT30A3204 Advanced Web Applications
Kokkonen Santeri

Contents

1 Technology choices

1.1 bootstrap

Client side components and responsive design.

1.2 highlight.js

Code snipped highlighting

1.3 react

Most familiar frontend framework

1.4 react-router

Client side routing

1.5 bcrypt

Password hashing

1.6 express

Easy to use

1.7 mongoose

Easy to use

1.8 jsonwebtoken

JWT Token signing and verification

1.9 multer

Handling multipart/form-data

2 Installation guidelines

2.1 Clone Repo

```
git clone https://github.com/mrleecock/webproject23.git
```

2.2 Installing dependencies

```
cd webproject23
npm install
```

2.3 Building frontend

```
npm run build
```

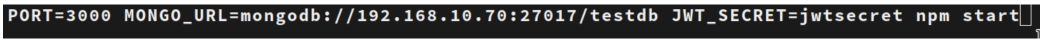
3 User manual

3.1 Before running

Make sure that mongoDB is running and url is known

3.2 Running

Run 'npm start' in root directory. Three(3) environment variables needs to be set; PORT, MONGO_URL and JWT_SECRET

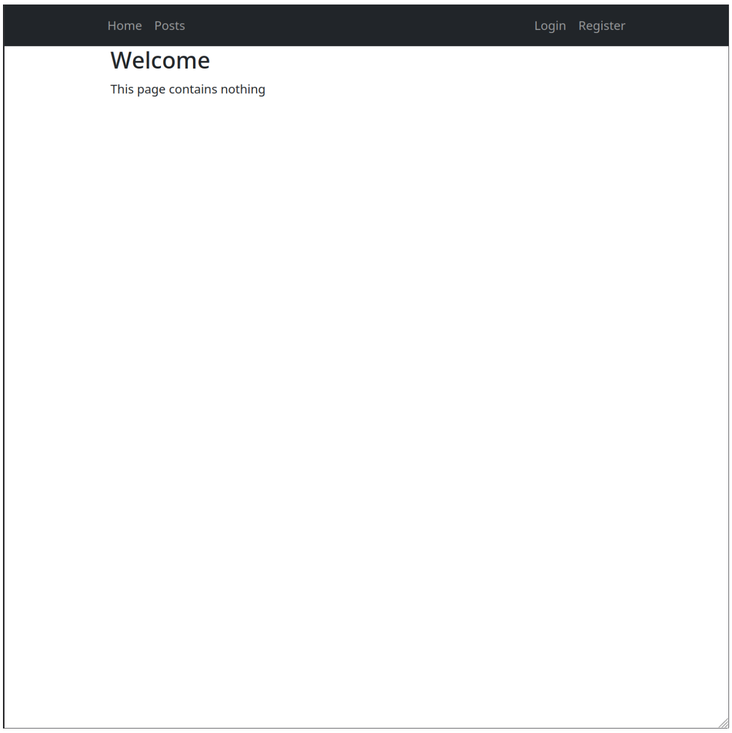


```
PORT=3000 MONGO_URL=mongodb://192.168.10.70:27017/testdb JWT_SECRET=jwtsecret npm start
```

Figure 1: Example

3.3 Using application

Open application in browser



Welcome

This page contains nothing

Figure 2: Home page

3.3.1 User registration

1. Navigate to 'Register' page
2. Fill username and password
3. If admin user check this checkbox
4. Click register button

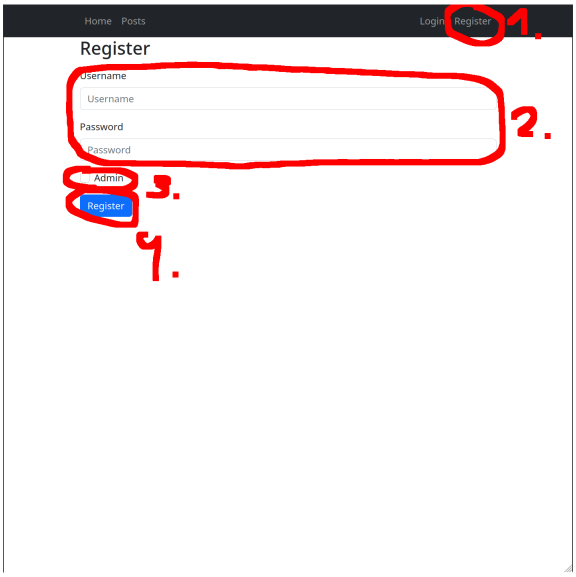
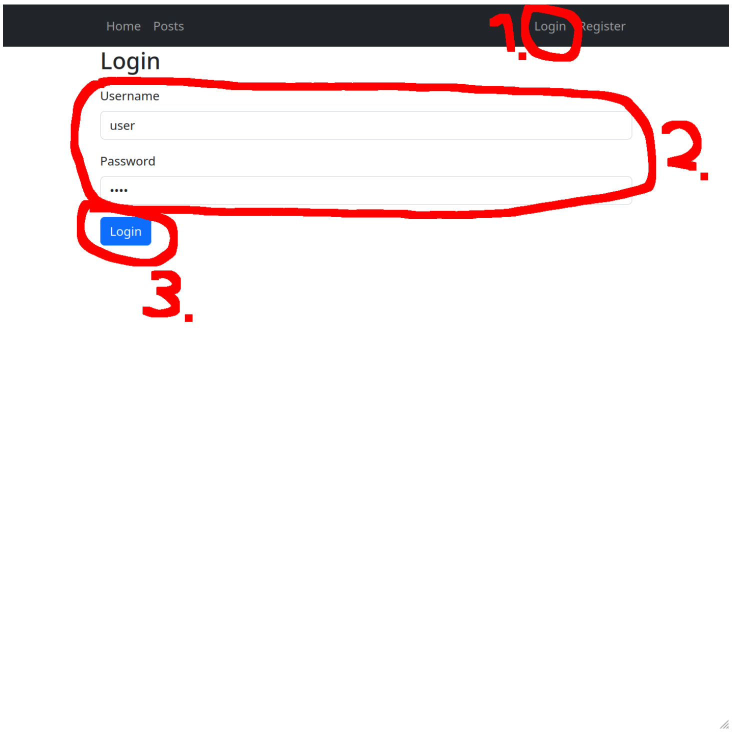


Figure 3: Registration page

After succesful registration application will redirect user to login page.

3.3.2 Login

1. Navigate to 'Login' page
2. Fill username and password
3. Click login button



Login

Username

user

Password

••••

Figure 4: Login Page

After succesful login application will redirect user to home page.

3.3.3 Changing user image

1. Navigate to 'User' page
2. Select new image
3. Click 'change picture' button

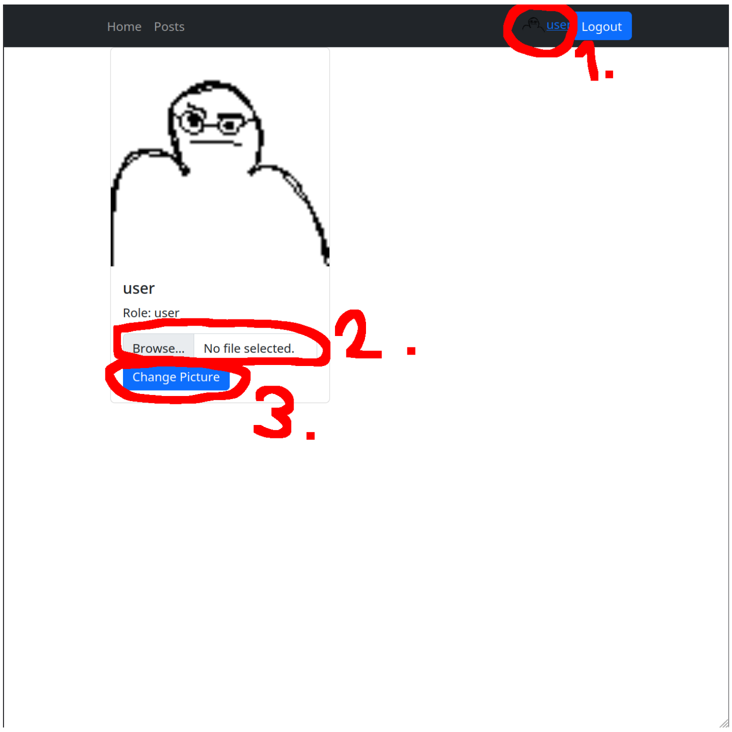


Figure 5: User Page

3.3.4 Adding post

1. Navigate to 'Posts' page
2. Click 'Add New Post' button
3. Fill the form
4. press 'Submit' button

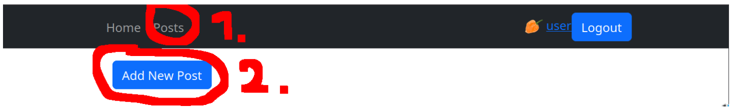
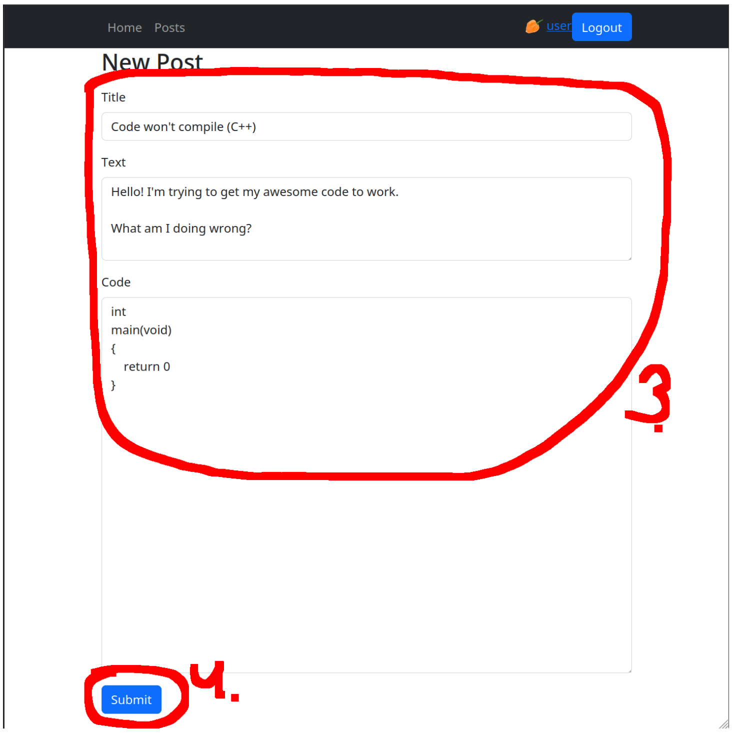


Figure 6: Posts page



New Post

Title

Code won't compile (C++)

Text

Hello! I'm trying to get my awesome code to work.

What am I doing wrong?

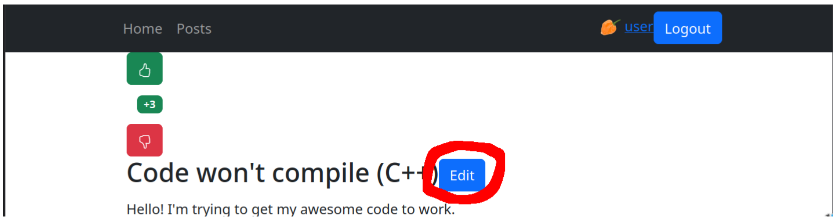
Code

```
int
main(void)
{
   return 0
}
```

Figure 7: Add new post page

3.3.5 Editing post

Users can edit their own posts by clicking 'Edit' button in 'post' page.



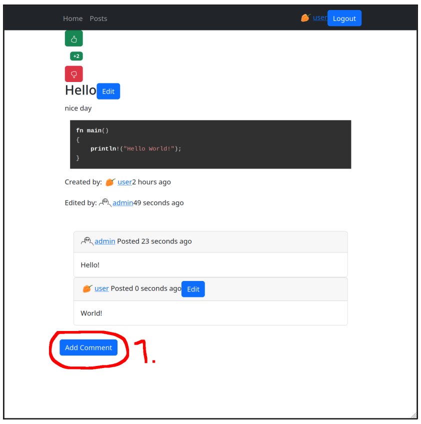
Code won't compile (C++)

Hello! I'm trying to get my awesome code to work.

Figure 8: Post page

3.3.6 Commenting

1. Click 'Add Comment' button
2. Fill the form
3. press 'Submit' button



Hello

nice day

```
fn main()
{
    println!("Hello World!");
}
```

Created by: user2 hours ago

Edited by: admin49 seconds ago

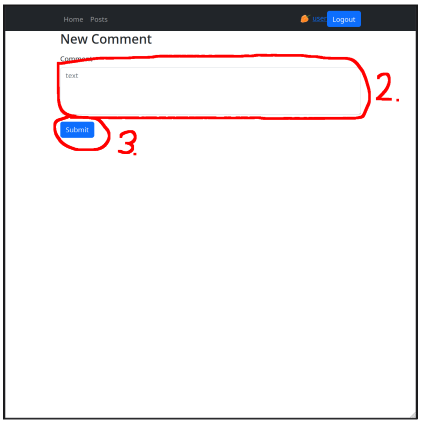
admin Posted 23 seconds ago

Hello!

user Posted 0 seconds ago

World!

Figure 9: Post page



New Comment

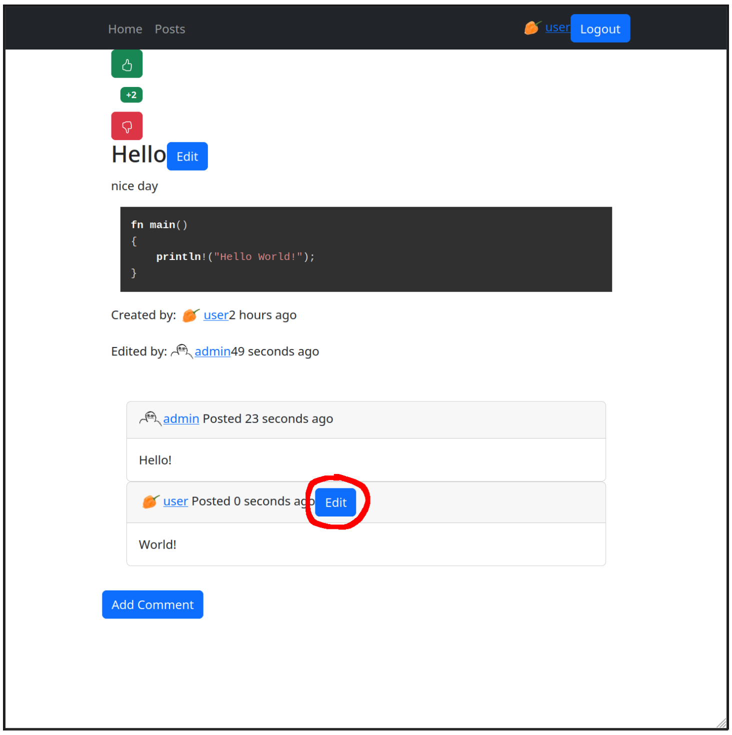
Comment

text

Figure 10: Add comment page

3.3.7 Edit comment

Users can edit their own comments by clicking 'Edit' button in 'post' page.



Hello

nice day

```
fn main()
{
    println!("Hello World!");
}
```

Created by: user2 hours ago

Edited by: admin49 seconds ago

admin Posted 23 seconds ago

Hello!

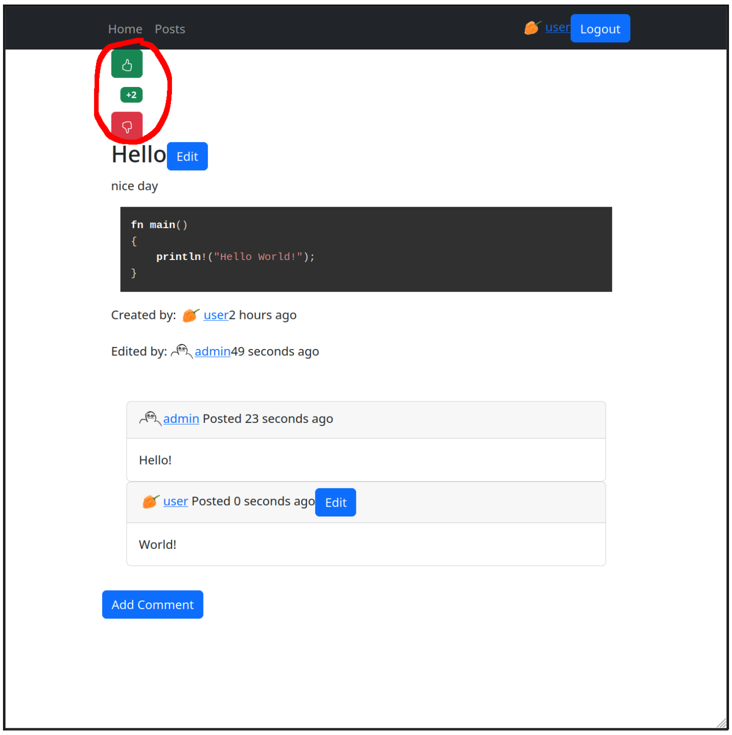
user Posted 0 seconds ago

World!

Figure 11: Post page

3.3.8 Voting

Users can vote each post.



Hello

nice day

```
fn main()
{
    println!("Hello World!");
}
```

Created by: user2 hours ago

Edited by: admin49 seconds ago

admin Posted 23 seconds ago

Hello!

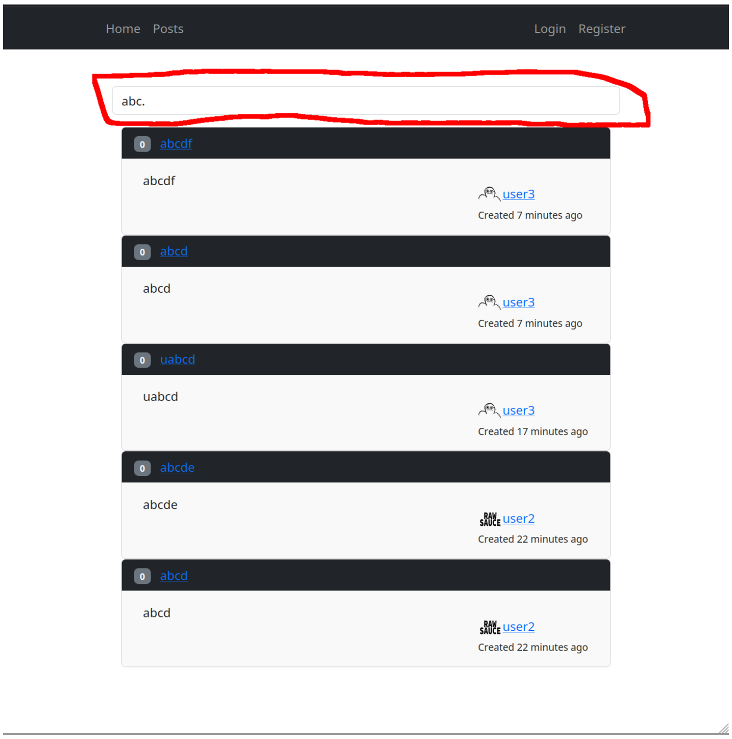
user Posted 0 seconds ago

World!

Figure 12: Post page

3.3.9 Filtering posts

Posts can be filtered using RegEx.



abc.

0 abcdf

abcdf

user3

Created 7 minutes ago

0 abcd

abcd

user3

Created 7 minutes ago

0 uabcd

uabcd

user3

Created 17 minutes ago

0 abcde

abcde

user2

Created 22 minutes ago

0 abcd

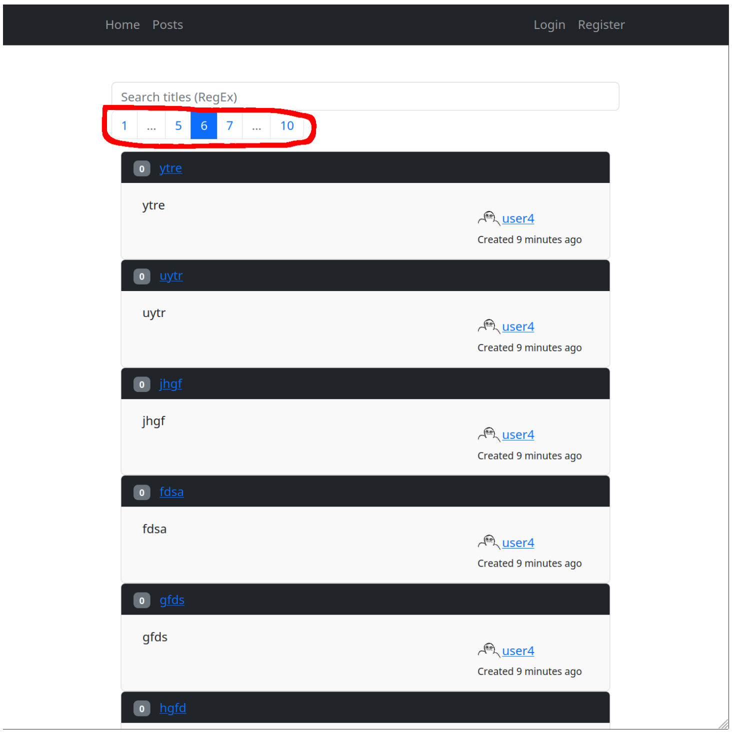
abcd

user2

Created 22 minutes ago

Figure 13: Posts page

3.3.10 Pagination



0 ytre

ytre

user4
Created 9 minutes ago

0 uytr

uytr

user4
Created 9 minutes ago

0 jhgf

jhgf

user4
Created 9 minutes ago

0 fdsa

fdsa

user4
Created 9 minutes ago

0 gfds

gfds

user4
Created 9 minutes ago

0 hgfd

Figure 14: Posts page

4 Features

Feature	Max Points	Points	Comment
Basic features with well written documentation	25	20	minus points from documentation
Users can edit their own comments/posts	4	4	
Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other	5	5	
Use some highlight library for the code snippets	2	2	
Use of a pager when there is more than 10 posts available	2	2	
Admin account with rights to edit all the post and comments and delete content	3	1.5	Deleting posts and comments is not possible in client side. Server side implementation done.
Provide a search that can filter out only those messages that have the searched keyword	2	2	
Vote posts and comments (only one vote per user)	3	3	
User profiles can have images which are show next to posts/comments	3	3	
User can click username and see user profile page where name, register date, (user picture) and user bio is listed	2	1	Only name, role and picture.
Last edited timestamp is stored and shown with posts/comments	2	2	
Code is not commented properly	-5	-5	-
		40.5	