Project Documentation

1. Technology Choices

The project is built using JavaScript, React for the frontend, and MongoDB for the database. These technologies were chosen for their robustness, scalability, and the extensive community support available. React's component-based architecture enables efficient development of dynamic user interfaces, while MongoDB offers a flexible, NoSQL database solution that scales well with complex data structures.

2. Installation Guidelines

To build the project, make sure you have Node.js and MongoDB installed. clone the repository, navigate to the project directory and jump to the client folder "npm install" to install the dependencies, then jump to the server folder "npm install" to install the dependencies, and then run respectively "npm start" to start the React application. Access the application in a web browser via "http://localhost:3000".

3. User Manual

The application allows users to create, view, update, and delete posts. Users can register, log in, and perform operations based on their authentication status. The navigation bar provides easy access to different sections of the application, including home, login, register, and create post pages.

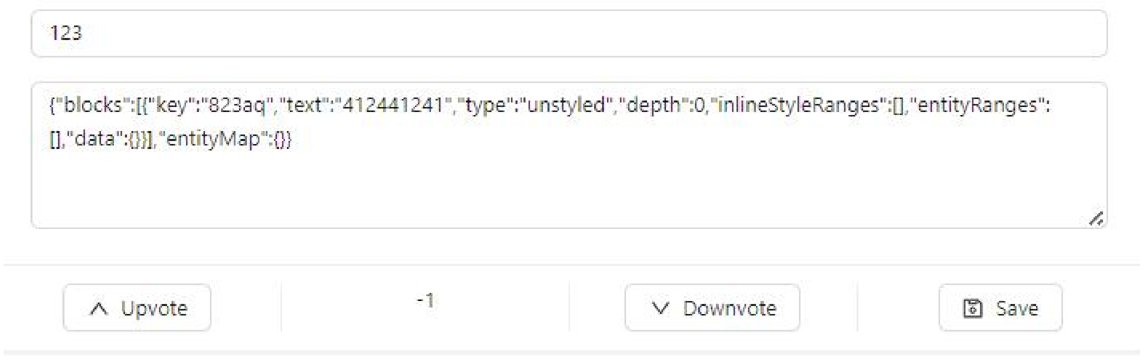
4. Features Implemented

Feature	**Max points**
Basic features (as stated in the previous chapter) with well written documentation	25

Users can edit their own comments/posts	4
Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other	5
Use some highlight library for the code snippets, for example https://highlightjs.org/	2
Use of a pager when there is more than 10 posts available	2
Admin account with rights to edit all the post and comments and delete content (if a post is removed, all its comments should be removed too)	3
Test software for accessibility; can it be used only with keyboard / voice command? Can screen readers work with your application?	3
Vote (up or down) posts and comments (only one vote per user)	3
Last edited timestamp is stored and shown with posts/comments	2
Create (unit) tests and automate some testing for example with https://www.cypress.io/ (at least 10 cases have to be implemented)	5

带格式表格[Sauli Tynkkynen]

Users can edit their own comments/posts	4


123

{"blocks":[{"key":"823aq","text":"412441241","type":"unstyled","depth":0,"inlineStyleRanges":[],"entityRanges":[],"data":{}}],"entityMap":{}}

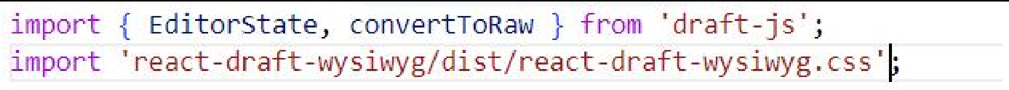
Utilization of a frontside framework, such as React, but you can also use Angular,	5

Vue or some other	

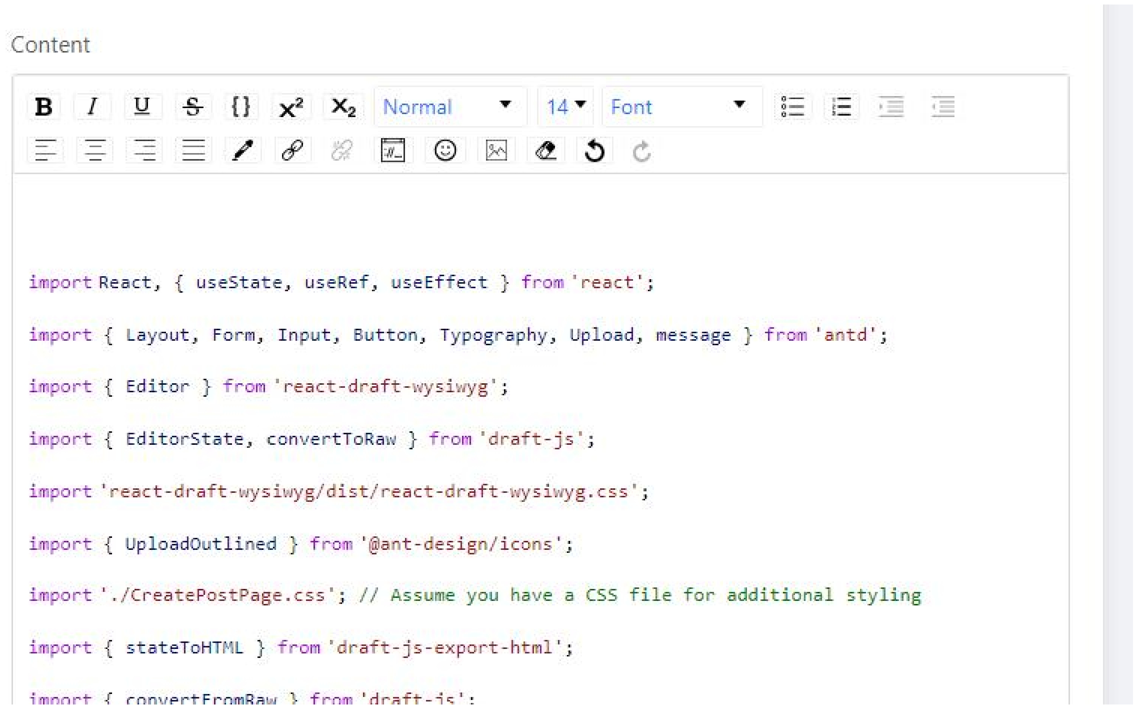
Use react

Use some highlight library for the code snippets, for example https://highlightjs.org/	2

带格式表格[Sauli Tynkkynen]



```
import { EditorState, convertToRaw } from 'draft-js';
import 'react-draft-wysiwyg/dist/react-draft-wysiwyg.css';
```



```
import React, { useState, useRef, useEffect } from 'react';

import { Layout, Form, Input, Button, Typography, Upload, message } from 'antd';

import { Editor } from 'react-draft-wysiwyg';

import { EditorState, convertToRaw } from 'draft-js';

import 'react-draft-wysiwyg/dist/react-draft-wysiwyg.css';

import { UploadOutlined } from '@ant-design/icons';

import './CreatePostPage.css'; // Assume you have a CSS file for additional styling

import { stateToHTML } from 'draft-js-export-html';

import { convertFromRaw } from 'draft-js';
```

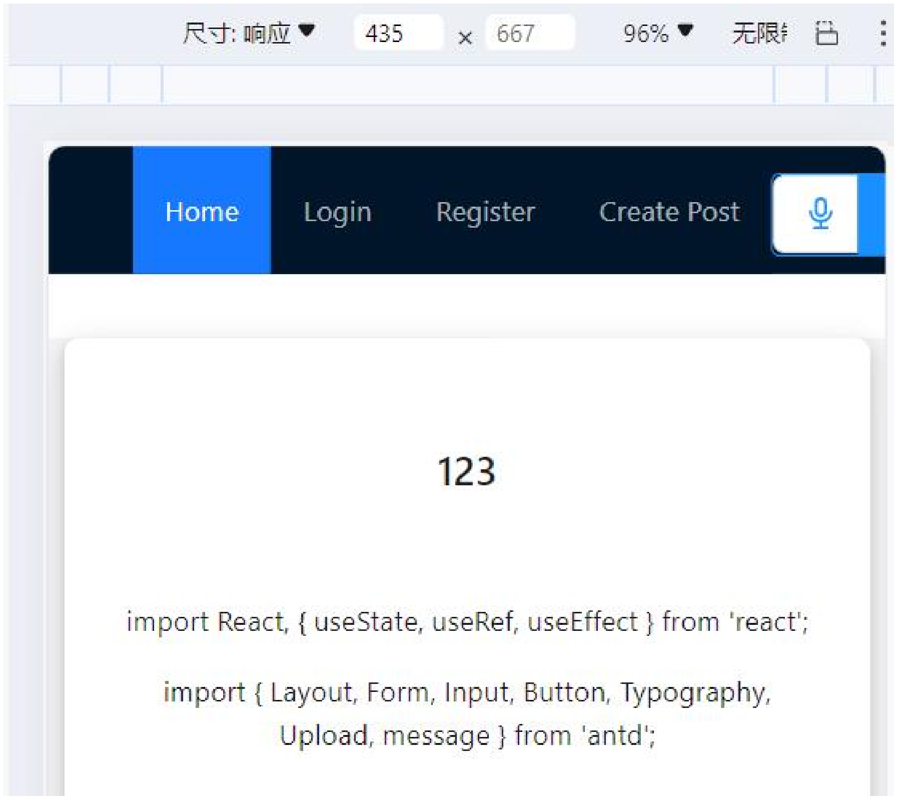
Use of a pager when there is more than 10 posts available	2

Scrolling pagination is used, with ten posts per page

Admin account with rights to edit all the post and comments and delete content (if a post is removed, all its comments should be removed too)	3

You can enter the code and 123456 when you register

Test software for accessibility; can it be used only with keyboard / voice command? Can screen readers work with your application?	3

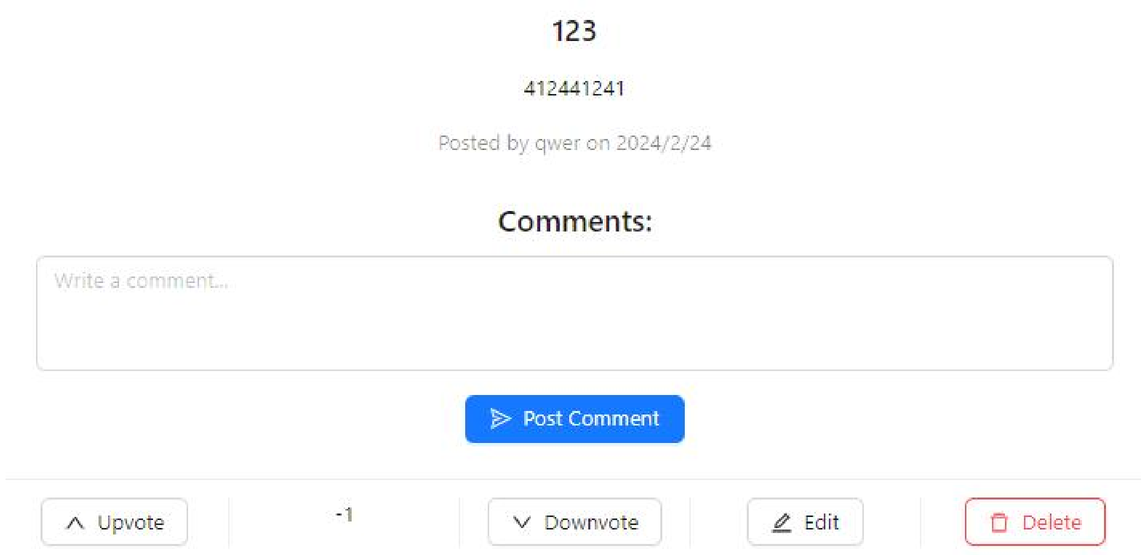


123

import React, { useState, useRef, useEffect } from 'react';

import { Layout, Form, Input, Button, Typography, Upload, message } from 'antd';

Vote (up or down) posts and comments (only one vote per user)	3
Last edited timestamp is stored and shown with posts/comments	2



Create (unit) tests and automate some testing for example with https://www.cypress.io/ (at least 10 cases have to be implemented)	5

Run npm run cypress:open for e2e testing

The project aims to demonstrate a comprehensive understanding of full-stack development, targeting a high level of functionality and user engagement.