

DV 200 Term 4

[interro]

1.1 Group Members

- Emilio Carreira 221350
- Erik Conradie 221147
- Ruan Jordaan 150139
- Anke Du Raan 221202



[]interro

[interro]

1.2 Brief Summary

Working in a team, we were expected to construct a full MERN Stack Application, that functions similar to that of Stack Overflow.

We built the programme for future Open Window Development research. Our platform will provide a place for students to ask and answer questions about software engineering.

[interro]

Open Window QnA Dashboard



2. Development Process :

2.1 Problem

Create an application similar to Stack overflow that will allow students at Open Window to post specific questions and answers to related to development.



2. Development Process :

2.2.1 Design

The screenshot shows the mobile application interface for 'interro'. At the top, there's a header bar with the 'interro' logo and a search bar. Below the header, the main content area is divided into three main sections:

- Announcements:** Displays three cards, each containing a 'Question Title' (e.g., 'Question 1', 'Question 2', 'Question 3') and a brief description.
- Events:** Displays three cards, each featuring a thumbnail image of two people in a dynamic pose, a 'Event Title' (e.g., 'Event 1', 'Event 2', 'Event 3'), and a short description. Each card has a yellow 'View Event' button at the bottom.
- Discover:** Displays three cards, each showing a snippet of code (e.g., 'header.css', 'header.js', 'header.html') and a 'Development' category. Each card has a yellow 'View Code' button at the bottom.

On the left side of the screen, there's a vertical navigation bar with icons for Home, Subjects (Development, Development), and Clubs (Horror Club). The 'Development' icon is highlighted.

The screenshot shows the 'Create a post' page of the 'interro' web application. At the top, there's a header bar with the 'interro' logo and a user profile icon. Below the header, the main content area is titled 'Create a post'.

The form fields include:

- A dropdown menu for 'Question' which is currently set to 'Development'.
- A 'Title' field with placeholder text 'Try to be as direct as you possibly can'.
- A 'Description' field with placeholder text 'Try to be as direct as you possibly can'.
- A rich text editor toolbar with various formatting options like Heading, Bold, Italic, etc.
- A 'Text (Optional)' input field.
- A 'Tags' section where users can select up to 3 relevant options from a list: HTML, CSS, JavaScript, React, MongoDB, and MySQL.

On the right side, there's a sidebar with navigation links: Home, Blog, Trending, Development, Inspiration, and Feedback. A 'New post guidelines' section lists the following rules:

1. Be nice
2. Lorem ipsum dolor sit
3. Lorem ipsum dolor sit
4. Lorem ipsum dolor sit
5. Read the [community guidelines](#)

At the bottom right of the form, there's a yellow 'Post' button.

2. Development Process :

2.2.2 Design

Search posts...

All Posts Questions Announcements

End year presentation Assignment Due

Development Posted by Admin 10 min ago

100 Comments

Question Title Development Posted by Username 10 min ago

100 Comments

Question Title Development Posted by Username 10 min ago

100 Comments

Rules

- Only development related posts
- Lowest ipsum dolor sit amet, consectetur adipiscing elit.
- Lowest ipsum dolor sit
- Lowest ipsum dolor sit
- Read the [community guidelines](#)

New Post

Search posts...

All Posts Questions

Question Title Development Posted by Username 10 min ago

100 Comments

Question Title Development Posted by Username 10 min ago

100 Comments

Question Title Development Posted by Username 10 min ago

100 Comments

Name Surname Illustration Student Due

100 150 1.2k 15k

New Post

2. Development Process :

2.2.3 Design

The screenshot shows a dark-themed user interface for a platform called [interro]. The left sidebar includes navigation links for Home, Subjects (Development, Interaction Design, Visual Culture, Product Design), and a search bar. The main content area displays a post by Erik Conradie titled "Seeking Guidance: JavaScript Challenge". The post discusses a complex task and includes a code snippet:

```
// JavaScript code causing me some issues
function trickyTask() {
    // My complex logic here...
}

trickyTask();
```

The post has a "Development" category badge and a "Most Upvoted" comment from Erik Conradie. The comment provides feedback on the code:

It looks like your code to generate the password needed an update, take a look at this.

```
function generateRandomPassword(length) {
    const charset = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$";
    let password = "";

    for (let i = 0; i < length; i++) {
        const randomIndex = Math.floor(Math.random() * charset.length);
        password += charset[randomIndex];
    }
    return password;
}

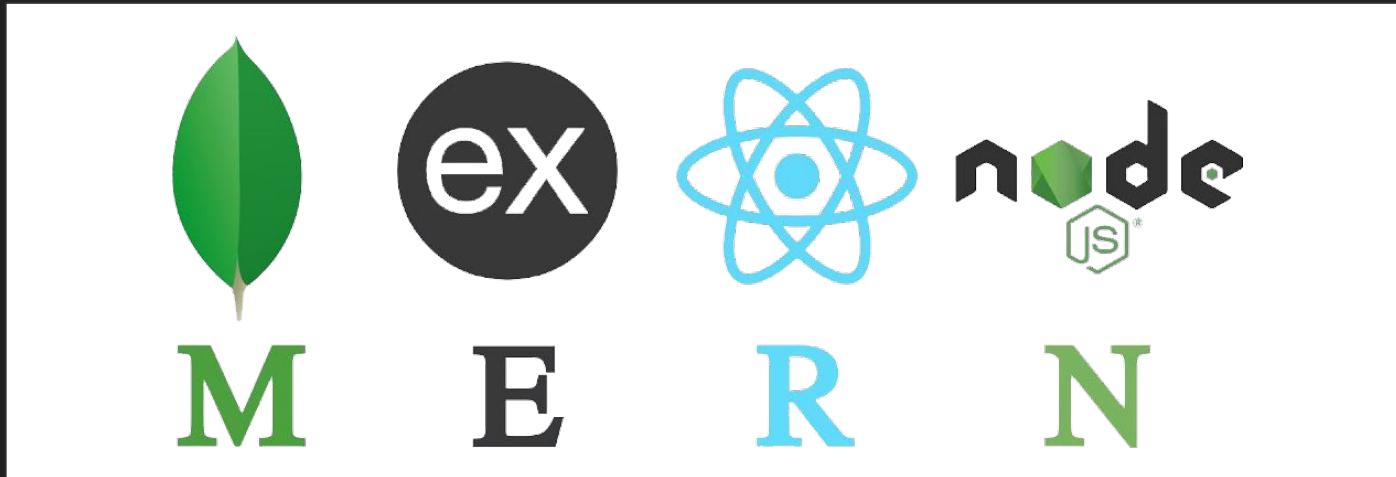
const randomPassword = generateRandomPassword(12); // You can specify the length you
console.log("Random Password: " + randomPassword);
```

2. Development Process:

2.3 Technologies and/or frameworks used.

- React
- Node.js

- MongoDB
- Axxess/Domains



3. Planning and Documentation

3.1. Visual Concept.

3.2. Features and Functionality.

3.3. Additional features.

3.1 Visual Concept

Icons:



interro

[]interro

[interro]

M

Montserrat

A B C D E F G H I
J K L M N O P Q R
S T U V W Z Y Z

Type scale: Montserrat

Category	Size	Weight
Heading	25 px	Bold
Subtitle 1	20 px	Bold
Body 1	16 px	Regular
BUTTON	12 px	Bold
Body 2	10 px	Regular
Caption	8 px	Regular

#000000

#161616

#DEF81D

#D8D8D8

#A5A5A5

#FFFFFF

Demo walk-through

3.2 Features and functionality

Signup

Users can sign up for the user on the sign up website. We implemented JWT to generate a user token and saved it alongside the rest of the relevant data in a the mongoDB database repository. I'll show you the various places on the site where this data has been used.

```
const [data, setData] = useState({ email: "", password: "" });
const [error, setError] = useState("");
const navigate = useNavigate();

const handleChange = ({ currentTarget: input }) => {
  setData({ ...data, [input.name]: input.value });
};

const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    const url = "http://localhost:3001/api/registerUser";
    const { data: res } = await axios.post(url, data);
    localStorage.setItem("token", res.data);
    navigate("/Login");
  } catch (error) {
    if (
      error.response &&
      error.response.status >= 400 &&
      error.response.status <= 500
    ) {
      setError(error.response.data.message);
    }
  }
};
```

Login

This brings us to the meat and potatoes of jwt's capabilities. Here's a screenshot from the perspective of a user who already exists in the system, showing the annoyingly encrypted input box. Passwords are encrypted so that they cannot be copied and pasted directly from a database to a website. This bodes well for the user.

The screenshot shows a login form for 'interro'. The title 'Login to Your Account' is at the top. Below it is an input field containing the email '221147@virtualwindow.c'. Below that is a placeholder 'Enter Email'. The next input field is completely filled with dots, representing an encrypted password. Below that is a placeholder 'Enter Password'. At the bottom is a green 'Sign In' button. To the right of the form, the text 'New Here ?' is displayed above a yellow 'Sign Up' button.

```
_id: ObjectId('652686ee0da8db5a346b62bf')
firstName: "Emilio"
lastName: "Carreira"
email: "221358@virtualwindow.co.za"
password: "$2b$10$dpDcr/7l.3uxacRaptVkJ.n8uVgU9YURcnp.HRtTKr5odhdKMg5vK"
__v: 0
image: "1698242669373.jpeg"
```

Profile Settings

If the user wishes to edit their profile they can do so by going to the edit profile page, as shown in the screen shot, which is linked on their profile page. Users can change their name, last name, email and password if they wish.

Profile Settings

Edit profile image

Name of file Choose image

Save Profile Image

Edit your profile details

First Name Last Name

Email

Password

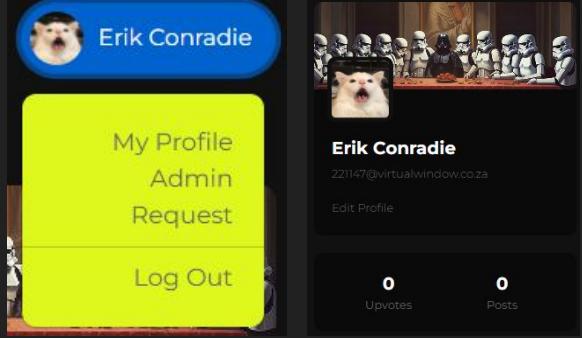
Save Profile

```
<Form onSubmit={handleProfileSubmit}>
  <Row className='mt-5'>
    <Col xs={12} className='pb-20'>
      <h2>Edit your profile details</h2>
    </Col>
  </Row>
  <Row className='mt-5 mb-20'>
    <Col xs={6}>
      <Form.Group className="mb-3">
        <Form.Label>First Name</Form.Label>
        <Form.Control
          type="text"
          placeholder={decodedToken.firstName}
          onChange={(e) => {
            setEditFirstName(e.target.value);
          }}
        />
      </Form.Group>
    </Col>
    <Col xs={6}>
      <Form.Group className="mb-3">
        <Form.Label>Last Name</Form.Label>
        <Form.Control
          type="text"
          placeholder={decodedToken.lastName}
          onChange={(e) => {
            setEditLastName(e.target.value);
          }}
        />
      </Form.Group>
    </Col>
  </Row>
```

Profile

If the user wishes to view their profile they need to click on the drop down menu with their name on it and then navigate to My Profile, as shown in the screen shots. From where they'll be able to view all the posts they posted and the number of upvoted they voted on other posts.

```
return (
  <>
  <div className="text-end profile-dropdown-container">
    <Dropdown>
      <Dropdown.Toggle className="user-dropdown">
        {decodedToken.image ? (
          <img
            className="nav-profile-pic"
            src={`${http://localhost:3001/profileImages/${decodedToken.image}}`}
            alt="..."
          />
        ) : (
          <img
            className="nav-profile-pic"
            src={placeholderImageUrl}
            alt="Placeholder"
          />
        )}
        <span>
          {decodedToken.firstName} {decodedToken.lastName}
        </span>
      </Dropdown.Toggle>
      <Dropdown.Menu>
        <div className="profile-dropdown-links">
          <Link to="/profile">My Profile</Link>
          {/* <Link to="/setting">Settings</Link> */}
          {'loggedIn' && 'isAdmin' && <Link to="/admin">Admin</Link>}
          <Link to="/CommunityRequests">Request</Link>
          <Dropdown.Divider />
          <Link to="/Login" onClick={handleLogout}>
            Log Out
          </Link>
        </div>
      </Dropdown.Menu>
    </div>
  </>
)
```



Erik Conradie
erik.conradie@virtualwindow.co.za
Edit Profile
Upvotes 0 Posts 0

```
return (
  <>
  <div className="dark-rounded-container mb-20 mt-20 text-start">
    <div className="profile-banner" style={{ backgroundImage: `url(${userProfile.banner})` }}>
    </div>
    <div className="pr-20 pl-20 pb-20">
      <div>
        {decodedToken.image ? (
          <img
            className="profile-avatar"
            src={`${http://localhost:3001/profileImages/${decodedToken.image}}`}
            alt="..."
          />
        ) : (
          <img
            className="nav-profile-pic"
            src={placeholderImageUrl}
            alt="Placeholder"
          />
        )}
      </div>
      <div className="w-100 mt-n20">
        <div className="w-100">
          <h2 className="text-xlarge">{decodedToken.firstName} {decodedToken.lastName}</h2>
        </div>
      </div>
    </div>
  </>
)
```

Posts

For the Boards. By clicking the upvote or downvote arrow on the post's left side, the user may tell if the content is well-liked or not. The problem is briefly outlined in the post's headline and body. The problem is elaborated on the next slide if the post is clicked.

All Posts Questions Announcements

Seeking Guidance: JavaScript Challenge Ahead 🔥🔥 Question

Development Posted by Ruan Jordaan 1 hours ago

I'm facing a bit of a head-scratcher with my JavaScript project, and I could use some assistance from the pros out there.

Solving the Mystery of JavaScript's Mysterious NaN Post

Development Posted by Ruan Jordaan 1 hours ago

Have you ever encountered the infamous NaN (Not-a-Number) error in your JavaScript code and wondered what's causing it?

```
return (
  <div>
    <posts.map((post) => (
      <div>
        <div>
          <div><span>{post.title}</span></div>
          <div><span>{post.description}</span></div>
        </div>
        <div><span>{post.votes}</span></div>
        <div><span>{post.time}</span></div>
        <div><span>{post.author}</span></div>
      </div>
    ))</div>
  </div>
)
```

Individual Post

Now that you've read the post, you can examine the enlarged version of the issue. The same header appears here, but a more detailed explanation of the issue is provided below. Users can share their thoughts on the issue at hand and possibly offer a workable solution in the comments section.

Development Posted by Ruan Jordaan

Seeking Guidance: JavaScript Challenge Ahead 🔥🔥

Hello, coding community! 🌟💻 I'm facing a bit of a head-scratcher with my JavaScript project, and I could use some assistance from the pros out there. I'm sure many of you have encountered similar hurdles before.

Here's the code snippet that's giving me trouble:

```
// JavaScript code causing me some issues
function trickyTask() {
  // My complex logic here...
}

trickyTask();
```

It might look simple at first glance, but trust me, it's a lot trickier than it appears. I've tried various solutions and spent quite a bit of time debugging, but I can't seem to get it right.

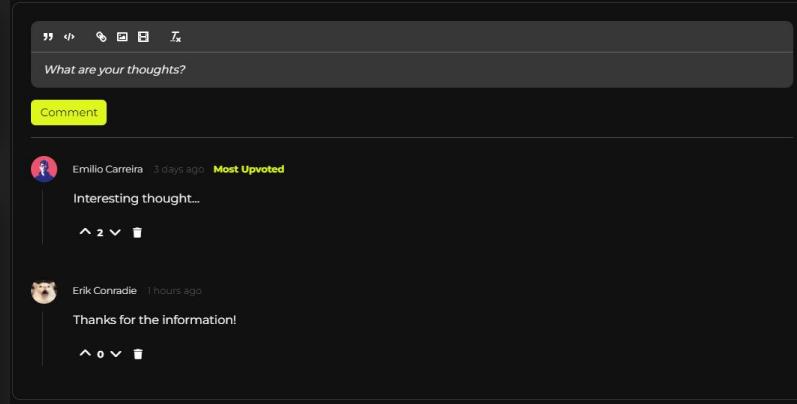
If you've got any insights, suggestions, or even just a fresh pair of eyes to take a look, I'd greatly appreciate your help. Please share your thoughts in the comments, and let's tackle this JavaScript challenge together! 🙏💬

```
return (
  <>
  <Col xs={7} className='pt-5 mt-5'>
    <div>
      {postData ? (
        <Col xs={12} key={postData.id}>
          {postData.postType === 'Announcement' ? (
            <div className='grey-outline d-flex'>
              <div xs={12} className='post-content'>
                <p className='post-descriptor mb-2'><span className='yellow bold'>{postData.community}</span> Posted by <span className='bold'>{postData.postAuthorName}</span> {postData.date}</p>
                <div className='d-flex w-100'>
                  <div className='w-75'>
                    <h2 className='mb-0'>{postData.title}</h2>
                    {"/<p className='mb-3'>{postTags}</p> "}
                  </div>
                </div>
                <div className='w-25 text-end'>
                  <Dropdown className='post-dropdown-button'>
                    <Dropdown.Toggle variant="success" id="dropdown-basic">
                      <span><i className='fa-solid fa-ellipsis-vertical white'></i></span>
                    </Dropdown.Toggle>
                    <Dropdown.Menu className='post-dropdown'>
                      <Dropdown.Item href="#/action-1"><span className='delete'><i className='fa-solid fa-trash'></i>Delete</span></Dropdown.Item>
                    </Dropdown.Menu>
                  </Dropdown>
                </div>
              </div>
            </div>
          ) : (
            <div dangerousInnerHTML={convertDeltaToHtml(postData.postContent)} />
            {"/<p className='post-descriptor mb-2'>{postData.comments}</p> "}
          )}
        </Col>
      ) : null}
    </div>
  </Col>
)
```

Comments

The comment section enhances user engagement by providing a platform for feedback and discussion.

- Leave comments under posts.
- Profile image, username, and comment content.
- Upvote and downvote comments.
- Users can delete their own comments.
- Utilized a POST request for comments.



```
// Create a comment for a specific post
router.post('/api/posts/:id/comments', async (req, res) => {
  const postId = req.params.id;
  const commentData = req.body;

  try {
    const post = await postsSchema.findById(postId);

    if (!post)
      return res.status(404).json({ error: 'Post not found' });

    post.comments.push(commentData);

    const updatedPost = await post.save();
    res.json(updatedPost);
  } catch (error) {
    console.error('Error adding comment:', error);
    res.status(500).json(error);
  }
});
```

```
const CommentSubmit = async (e) => {
  e.preventDefault();
```

```
const DeleteComment = async (commentId) => {
  try {
```

```
const UpdateComment = async (commentId, voteType) => {
```

```
/* COMMENTS */
(postData && postData.comments && postData.comments.length > 0 ? (
  postData.comments.sort((a, b) => b.votes - a.votes).map((comment, index) => (
    <div className='mt-4' key={index}>
      <Row>
        <Col xs={12}>
```

Voting System

Empower user feedback and interaction with the following features:

- Users can express their opinion on both comments and posts.
- Highest-voted comment is prominently displayed.
- Initial Vote: Users have the flexibility to add or remove their votes.
- The functionality is achieved using a PATCH request.

Interaction Design

1 Struggling w/
Advice!

```
router.patch('/api/posts/:postId/comments/:commentId/votes', async (req, res) => {
  const { postId, commentId } = req.params;
  const voteType = req.body.voteType;
  const UserId = req.body.UserId;

  try {
    const post = await postsSchema.findById(postId);

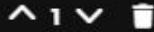
    if (!post) {
      return res.status(404).json({ error: 'Post not found' });
    }

    const comment = post.comments.id(commentId);
    if (!comment) {
      return res.status(404).json({ error: 'Comment not found' });
    }

    const hasVoted = comment.hasVoted.find(vote => vote.UserId === UserId);
    if (!hasVoted) {
      comment.hasVoted.push({ UserId, VoteType: voteType });
      if (voteType === 'Up') {
        comment.votes += 1;
      } else if (voteType === 'Down') {
        comment.votes -= 1;
      }
    } else {
      const existingVoteType = hasVoted.VoteType;
      if (existingVoteType === voteType) {
        const voteIndex = comment.hasVoted.findIndex(vote => vote.UserId === UserId);
        comment.hasVoted.splice(voteIndex, 1);
        if (voteType === 'Up') {
          comment.votes -= 1;
        } else if (voteType === 'Down') {
          comment.votes += 1;
        }
      } else {
        hasVoted.VoteType = voteType;
        if (voteType === 'Up') {
          comment.votes += 2;
        } else if (voteType === 'Down') {
          comment.votes -= 2;
        }
      }
    }
  }
});
```

Erik Conradie 17 hours ago **Most Upvoted**

Thanks for the information!



```
const UpdateComment = async (commentId, voteType) => {
  const UserId = decodedToken._id;
  try {
    // Send a PUT request to your server to update the specific comment
    const response = await axios.patch(`http://localhost:3001/api/posts/${id}/comments/${commentId}/votes`, { votetype, UserId });
    console.log('Comment updated successfully:', response.data);
    refreshPosts();
  } catch (error) {
    console.error('Error updating comment:', error);
  }
};
```

New Post

On the right side of the posts there is a "new post" area for each subject. If a user wants to make a new post, they must do it by filling up every available input area. Tags for title, group, summary, body, and origin. The user must choose the tags to specify the language(s) he is studying.

```
<div
  id="editor"
  className="mb-20"
  type="text"
  placeholder="PostContent"
  value={postContent}
  onChange={(e) => setPostContent(e.target.value)}
>
</div>

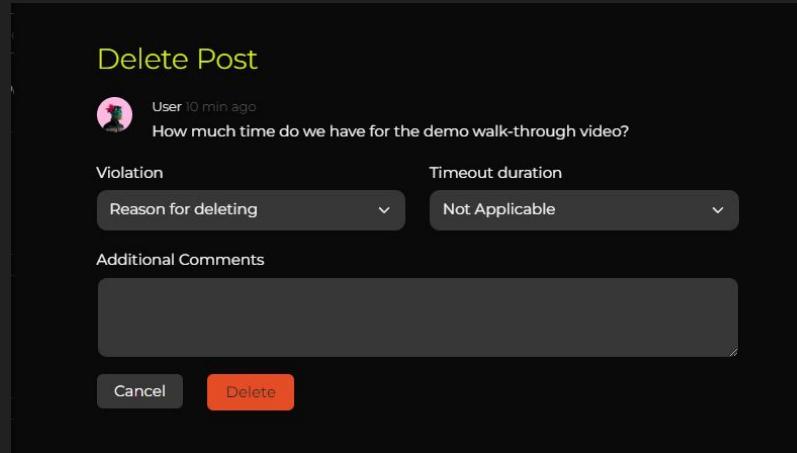
<h2 className="font-weight-400">Tags</h2>
<p className='post-descriptor'>Choose 1 - 3 of the most relevant options</p>

<div className='d-flex flex-wrap mb-20'>
  {tags.map((tag) => (
    <div
      className='create-post-tags ${selectedTags.includes(tag) ? 'selected' : ''}'
      key={tag}
      onClick={() => handleTagSelection(tag)}
    >
      <span className="mb-0 text-small">{tag}</span>
    </div>
  ))}
</div>

<Button variant="primary text-medium mb-5" type="submit"> Post</Button>
</Col>
```

Delete Post

- Post Deletion:
Users can delete their posts for content management.
- Modal Prompt: A user-friendly modal dialog appears to confirm post deletion.
- Improved User Experience:
Ensures a deliberate choice when removing posts.



```
const handleDelete = async (e) => {
  e.preventDefault();

  const payload = {
    postBy: 'postBy',
    deletedBy: 'deletedBy',
    violation: violation,
    timeout: timeout,
    postContent: postContent,
    additionalComment: additionalComment,
  };

  try {
    const response = await axios.post('http://localhost:3001/api/postDelete', payload);
    console.log('Post successfully submitted:', response.data);

    const deletePostResponse = await axios.delete(`http://localhost:3001/api/posts/${postId}`);
    console.log('Post successfully deleted:', deletePostResponse.data);

    refreshPosts();
  } catch (error) {
    console.error('Error submitting Post:', error);
  }
};
```

Search & Result

Empower users to find specific content easily with these features:

- Users can search for specific posts using the search bar.
- The displayed content is dynamically filtered based on user search queries.
- Provides a user-friendly way to access relevant content.

A screenshot of a search interface. At the top, there is a search bar with the text "tips". Below it, a navigation bar has "All" underlined and other options: Posts, Questions, Announcements. A post card is shown with the title "Tips", a "Post" badge, and a "Development" category. It was posted by "Anke Du Raan" 1 week ago. The post content is "Tips on to solve errors". On the right side of the card is a vertical ellipsis (...).

A screenshot of a code editor displaying a component's render function. The code uses React's functional component syntax. It includes a search input field with placeholder "Search Posts...", controlled by the state variable `searchQuery`. The `onChange` event handler updates `searchQuery` with the new value from the input. The component also includes logic to filter a list of posts based on the search query, using the `filter` method and logical OR operator `||` to check both title and description for matches.

```
<Col xs={7} className='pt-5'>
  <div>
    <Form>
      <Form.Group controlId="exampleForm.ControlInput1">
        <Form.Control
          type="text"
          placeholder="Search Posts..."
          value={searchQuery}
          onChange={(e) => setSearchQuery(e.target.value)}
        />
      </Form.Group>
    </Form>
  </div>

{posts
  .filter((post) => post.community === parsedSubjectsData && post.postType === 'Post')
  .filter((post) => post.title.toLowerCase().includes(searchQuery.toLowerCase()) || post.description.toLowerCase().includes(searchQuery.toLowerCase()))
  .reverse()}
```

Profile image upload

The user has the option to alter the default profile image if they so like. An image that the user uploads for their profile can then be saved, changed, and reloaded into the database (mongoDB) using the token decoder as displayed in the screenshots.

Profile Settings

Edit profile image

Name of file

Choose Image

Save Profile Image

```
<Form onSubmit={handleImageSubmit}>
  <Row>
    <Col xs={3} className='mt-5 mb-20'>
      {previewImage && (
        <img src={previewImage} alt="Preview" style={{ width: "100%" }} />
      )}
      <span>{imgName}</span>
    </Col>
    <Col xs={9} className='mt-5 mb-20'>
      <Form.Group className="mb-3">

        <label className="btn btn-primary">
          Choose Image
          <input
            type="file"
            accept="image/*"
            style={{ display: "none" }}
            onChange={getImage}
          />
        </label>

      </Form.Group>
    </Col>
  </Row>

  <Button variant="primary text-medium mb-5" type="submit">
    Save Profile Image
  </Button>
</Form>
```

3.3 Additional features

Add communities

Users can initiate the process of creating a new community by submitting a request through the Community Request Form, as shown in the accompanying screenshots.

Community Request Form

Your community request will be sent to the community council for review.

Title
Try to be as descriptive as possible

Description
Try to be as descriptive as possible

Upload Image Profile

Upload URL
Enter the image URL

Upload Image Background

Upload URL
Enter the image URL

Bottom Banner Color
Black

```
<Form onSubmit={handleSubmit} encType="multipart/form-data">
<Row>
  <Col>
    <Form.Group controlId="exampleForm.ControlInput1">
      <Form.Control
        type="text"
        placeholder="Title"
        value={title}
        onChange={(e) => setTitle(e.target.value)} />
      <p className='post-descriptor'>Try to be as descriptive as possible</p>
    </Form.Group>

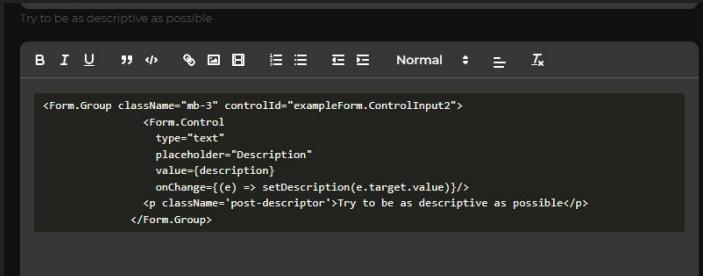
    <Form.Group controlId="exampleForm.ControlInput2">
      <Form.Control
        type="text"
        placeholder="Description"
        value={description}
        onChange={(e) => setDescription(e.target.value)} />
      <p className='post-descriptor'>Try to be as descriptive as possible</p>
    </Form.Group>

    <Form.Group controlId="formFileInput" >
      <Form.Label>Upload Image Profile</Form.Label>
      <Form.Control
        className='file-input-Community'
        type="text"
        placeholder='Upload URL'
        value={imageProfile} // Use the state variable
        onChange={handleProfileImageChange} />
      <p className='post-descriptor'>Enter the image URL</p>
    </Form.Group>
```

Rich text editor

Users have the option to include a piece of code in the comment section when they add a post or ask a question, as demonstrated in the provided screenshot.

Try to be as descriptive as possible:



```
B I U " <> ⌂ ☰ ⌂ ⌂ ⌂ ⌂ Normal ⌂ Tx

<Form.Group className="mb-3" controlId="exampleForm.ControlInput2">
  <Form.Control
    type="text"
    placeholder="Description"
    value={description}
    onChange={(e) => setDescription(e.target.value)}>
    <p className="post-descriptor">Try to be as descriptive as possible</p>
</Form.Group>
```

```
<Form onSubmit={handleSubmit}>

<Row>
  <Col xs={6}>
    <Form.Select className="custom-select"
      value={postType}
      onChange={(e) => setPostType(e.target.value)}
      name='postType'
    >
      <option>Post Type</option>
      <option value="Question">Question</option>
      <option value="Post">Post</option>
      <option value="Announcement">Announcement</option>
    </Form.Select>
  </Col>
  <Col xs={6}>
    <Form.Select aria-label="Default select example"
      value={community}
      onChange={(e) => setCommunity(e.target.value)}
      name='community'
    >
      <option>Open this select menu</option>
      <option value="Development">Development</option>
      <option value="Interaction Design">Interaction Design</option>
      <option value="Product Design">Product Design</option>
      <option value="Visual Culture">Visual Culture</option>
    </Form.Select>
  </Col>
</Row>
```

Add announcement

Furthermore, users have the capability to make announcements when creating a new post, as illustrated in the screenshots.

Create a post

Announcement

Open this select menu

Title
Try to be as descriptive as possible

Description
Try to be as descriptive as possible

```
<Form onSubmit={handleSubmit}>
  <Row>
    <Col xs={6}>
      <Form.Select className="custom-select"
        value={postType}
        onChange={(e) => setPostType(e.target.value)}
        name='postType'
      >
        <option>Post Type</option>
        <option value="Question">Question</option>
        <option value="Post">Post</option>
        <option value="Announcement">Announcement</option>
      </Form.Select>
    </Col>

    <Col xs={6}>
      <Form.Select aria-label="Default select example"
        value={community}
        onChange={(e) => setCommunity(e.target.value)}
        name='community'
      >
        <option>Open this select menu</option>
        <option value="Development">Development</option>
        <option value="Interaction Design">Interaction Design</option>
        <option value="Product Design">Product Design</option>
        <option value="Visual Culture">Visual Culture</option>
      </Form.Select>
    </Col>
  </Row>
```

Add tags to post & questions

In order to facilitate the process of locating posts using the search box, users can add tags to their posts as they are being created, as depicted in the screenshot.

Tags

Choose 1-3 of the most relevant options

HTML

CSS

JavaScript

React

Node.js

Express

MongoDB

SQL

Python

Java

C++

C#

```
<div className='d-flex flex-wrap mb-20'>
  {tags.map((tag) => (
    <div
      className={`${`create-post-tags ${selectedTags.includes(tag) ? 'selected' : ''}`}
      key={tag}
      onClick={() => handleTagSelection(tag)}
    >
      <span className='mb-0 text-small'>{tag}</span>
    </div>
  ))}
</div>
```

4. Final output

[interro]

[Open Window](#) QnA Dashboard

Problems we've encountered

Adding Profile image

[interro]

Profile Settings

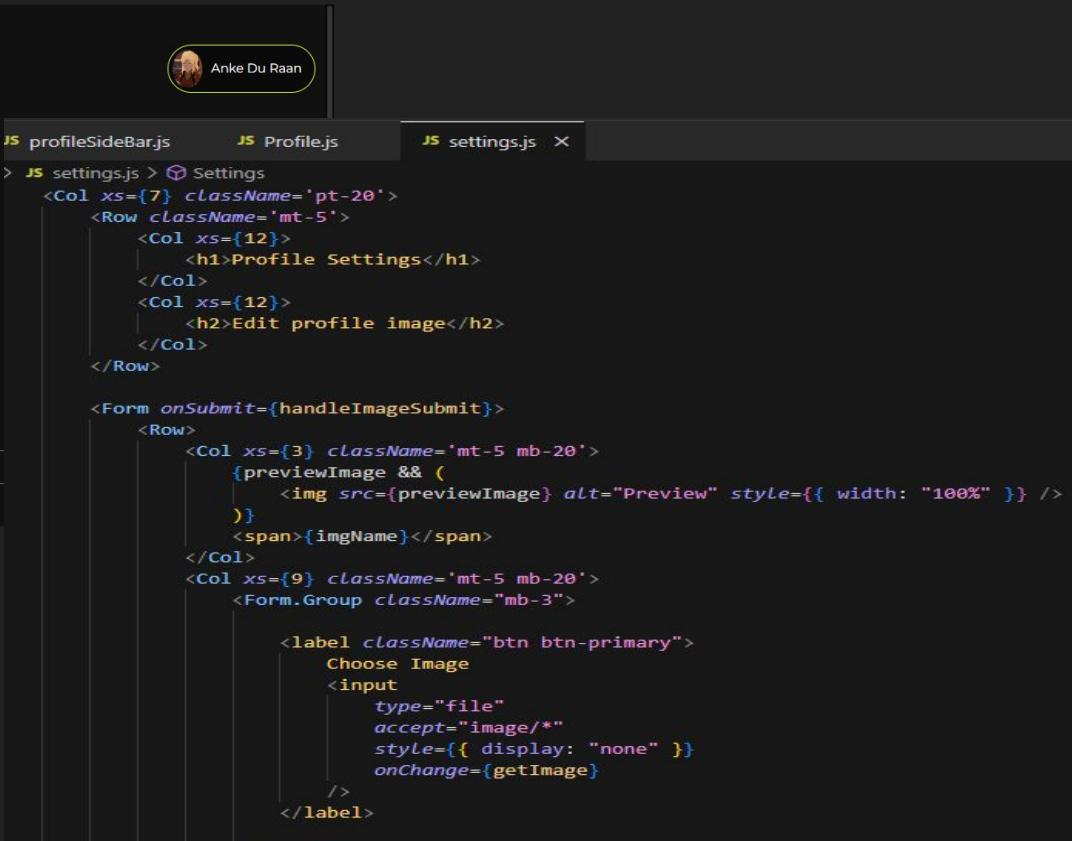
Edit profile image

Name of file Choose Image

Save Profile Image

Edit your profile details

First Name Last Name



The screenshot shows a dark-themed web application interface. On the left, there's a sidebar with navigation links: Home, Subjects (with Development, Interaction Design, Visual Culture, and Product Design), and a [interro] logo. The main content area has a title "Profile Settings" and a sub-section "Edit profile image". It includes fields for "Name of file" (with a "Choose Image" button) and a "Save Profile Image" button. Below this is a section titled "Edit your profile details" with "First Name" and "Last Name" input fields, both containing the value "Anke Du Raan". At the top right, there's a user profile card with a placeholder image and the name "Anke Du Raan". To the right of the profile card is a code editor window showing three files: profileSideBar.js, Profile.js, and settings.js. The code is written in JSX, handling profile settings and image upload logic.

```
> JS profileSideBar.js > JS Profile.js > JS settings.js < />
> JS settings.js > ⚡ Settings
<Col xs={7} className='pt-20'>
  <Row className='mt-5'>
    <Col xs={12}>
      <h1>Profile Settings</h1>
    </Col>
    <Col xs={12}>
      <h2>Edit profile image</h2>
    </Col>
  </Row>

<Form onSubmit={handleImageSubmit}>
  <Row>
    <Col xs={3} className='mt-5 mb-20'>
      {previewImage && (
        <img src={previewImage} alt="Preview" style={{ width: "100%" }} />
      )}
      <span>{imgName}</span>
    </Col>
    <Col xs={9} className='mt-5 mb-20'>
      <Form.Group className="mb-3">
        <label className="btn btn-primary">
          Choose Image
        </label>
        <input
          type="file"
          accept="image/*"
          style={{ display: "none" }}
          onChange={getImage}
        />
      </Form.Group>
    </Col>
  </Row>
</Form>
```

Struggled to get the upload for the profile image to be saved to database and then shown on website. Made sure everything was correctly spelled.

Edit Profile

Struggled to get the user's data to show as a placeholder on the edit profile page. Used a placeholder with `{decodedToken.firstName}` to fix it.

```
leSideBar.js      JS Profile.js      JS settings.js ×  
ettings.js > ⚡ Settings  
<Form onSubmit={handleProfileSubmit}>  
  <Row className='mt-5'>  
    <Col xs={12} className='pb-20'>  
      <h2>Edit your profile details</h2>  
    </Col>  
  </Row>  
  <Row className='mt-5 mb-20'>  
    <Col xs={6}>  
      <Form.Group className="mb-3">  
        <Form.Label>First Name</Form.Label>  
        <Form.Control  
          type="text"  
          placeholder={decodedToken.firstName}  
          onChange={(e) => {  
            setEditFirstName(e.target.value);  
          }}  
        />  
      </Form.Group>  
    </Col>  
    <Col xs={6}>  
      <Form.Group className="mb-3">  
        <Form.Label>Last Name</Form.Label>  
        <Form.Control  
          type="text"  
          placeholder={decodedToken.lastName}  
          onChange={(e) => {  
            setEditLastName(e.target.value);  
          }}  
        />  
      </Form.Group>  
    </Col>  
  </Row>
```

Edit your profile details

First Name

Last Name

Email

Password

Save Profile



Edit Profile image

[interro]

Profile Settings

Edit profile image

Name of file

Choose Image

Save Profile Image

Edit your profile details

First Name

Anke

Last Name

Du Raan



Anke Du Raan

```
<Form onSubmit={handleImageSubmit}>
  <Row>
    <Col xs={3} className='mt-5 mb-20'>
      {previewImage && (
        <img src={previewImage} alt="Preview" style={{ width: "100%" }} />
      )}
      <span>{imgName}</span>
    </Col>
    <Col xs={9} className='mt-5 mb-20'>
      <Form.Group className="mb-3">
        <label className="btn btn-primary">
          Choose Image
        </label>
        <input
          type="file"
          accept="image/*"
          style={{ display: "none" }}
          onChange={getImage}
        />
      </Form.Group>
    </Col>
  </Row>
<Button variant="primary text-medium mb-5" type="submit">
  Save Profile Image
</Button>
</Form>
```

Struggled to get the image to upload into the database then show. Had to make sure everything was spelled correctly.

Voting

Interaction Design
Struggling w/
Advice!

Erik Conradie 17 hours ago **Most Upvoted**

Thanks for the information!

^ 1 ✓

We encountered challenges in implementing the ability for users to revert their votes within the voting system.

```
router.patch('/api/posts/:postId/comments/:commentId/votes', async (req, res) => {
  const { postId, commentId } = req.params;
  const voteType = req.body.voteType;
  const UserId = req.body.UserId;

  try {
    const post = await postsSchema.findById(postId);

    if (!post) {
      return res.status(404).json({ error: 'Post not found' });
    }

    const comment = post.comments.id(commentId);
    if (!comment) {
      return res.status(404).json({ error: 'Comment not found' });
    }

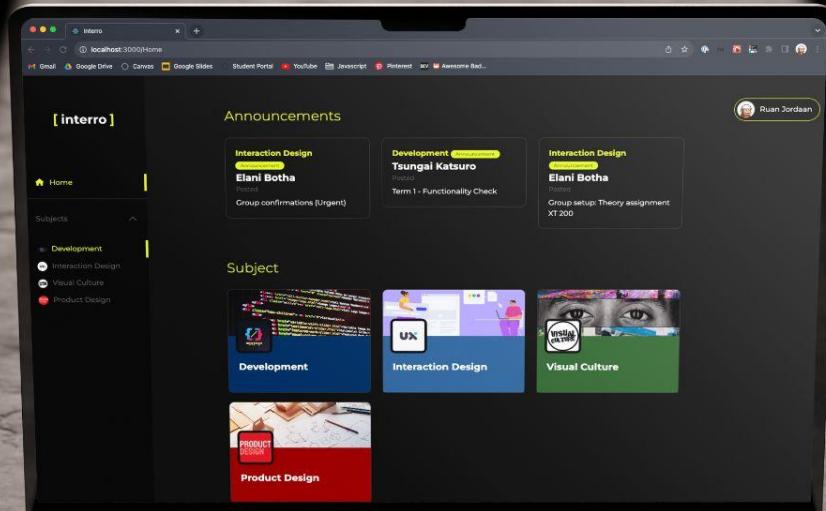
    const hasVoted = comment.hasVoted.find(vote => vote.UserId === UserId);
    if (!hasVoted) {
      comment.hasVoted.push({ UserId, VoteType: voteType });
      if (voteType === 'Up') {
        comment.votes += 1;
      } else if (voteType === 'Down') {
        comment.votes -= 1;
      }
    } else {
      const existingVoteType = hasVoted.VoteType;
      if (existingVoteType === voteType) {
        const voteIndex = comment.hasVoted.findIndex(vote => vote.UserId === UserId);
        comment.hasVoted.splice(voteIndex, 1);
        if (voteType === 'Up') {
          comment.votes -= 1;
        } else if (voteType === 'Down') {
          comment.votes += 1;
        }
      } else {
        hasVoted.VoteType = voteType;
        if (voteType === 'Up') {
          comment.votes += 2;
        } else if (voteType === 'Down') {
          comment.votes -= 2;
        }
      }
    }
  }
}
```

Deployment

The server frontend has been deployed, but we encountered difficulties in getting the server end to become operational.



Mockups



The screenshot shows a laptop displaying a web application titled '[interro]' on its screen. The application has a dark theme and features a sidebar on the left with a navigation bar and a 'Subjects' section containing categories like Development, Interaction Design, Visual Culture, and Product Design. The main content area displays several posts from a user named Ruan Jordaan:

- How Can We Improve the User Experience of Our Smart Home Control Panel?** (Product Design) posted by Ruan Jordaan 27 minutes ago. The post discusses enhancing smart home control panels.
- Designing for Inclusivity: Smart Home Interfaces** (Product Design) posted by Ruan Jordaan 29 minutes ago. The post explores creating user-friendly interfaces for everyone.
- Behind the Scenes of Product Design: From Concept to Creation** (Product Design) posted by Ruan Jordaan 30 minutes ago. The post discusses the process of product design from concept to creation.
- Struggling with Mobile App Navigation – Need UX Advice!** (Interaction Design) posted by Ruan Jordaan 31 minutes ago. The post asks for advice on mobile app navigation.
- Uncovering the User Experience Challenges in Mobile Banking**

Ruan Jordaan's profile is visible on the right, showing an icon, the name 'Ruan Jordaan', the email 'admin@admin.com', a 'Edit Profile' link, and activity statistics: 1 Upvotes and 7 Posts. A 'New Post' button is also present.





5. Conclusion

In conclusion, our journey in creating a Q&A website was a mix of challenges and rewards. While we encountered some difficulties along the way, we are ultimately satisfied with the end product.

Our Q&A website stands as a testament to our teamwork, perseverance, and dedication. It serves as a valuable resource for knowledge-sharing and learning, and we're proud of what we've achieved together.

The lessons learned and experiences gained during this project will continue to benefit us in future endeavors.