Milestone 4 (Team C)

Product summary

MentorMe is a web-based platform with the goal of revolutionizing the tutoring industry. Its goal is to link students with the top teachers and SMEs in the field.

List of final product functions:

- New students should be able to register using their university email addresses.
- Students must log in to the system using their university email.
- Students should have access to the system and be able to search for tutors.
- A message request to the tutor can be sent by students.
- Students must be able to have real-time messages with the tutor once the tutor has accepted their request.
- A tutor's course must be available for students to evaluate.
- The tutor must be able to register in the system and use it without any problem once the verification is completed by the Moderator.
- The tutor must be able to log in to the system.
- The tutor shall be able to upload his/her CV & image.
- The tutor must be able to accept a student's communication request.
- When new Tutor information arrives, the system must tell the Moderator.
- The moderator should be able to approve/reject content posted by the tutor before making it live (CV, Image).
- The tutor and the Students must be able to update their profiles.

What's Unique??

Ratings

Ban of a Student Profile or of a tutor profile by a moderator.

Chat Request Functionality

URL of the deployed application: http://20.113.70.224/

Usability Test Plan

Test objectives

On our website MentorMe, we have decided to test the functionality of Search along with the filters that we have on the website.

Our main concern is to check that if a naive/new user tries to use our application for the first time, how easily/comfortably he/she can use the application without any problems. We want to make sure that the design of the application is user-friendly in every possible way so that they don't need any kind of manual to use our application, even for the very first time. We check for things like: "Will users easily find the search box in its present location?" or "Will the user be able to find the filter functionality and use it easily along with the search option that he has."

Test background and setup

System setup: To perform this test Internet connection and a device with a browser.

Starting point: In order to conduct usability testing the starting point for a user is to open a browser and enter our website's URL.

Who are the intended users: We have selected our users and they can be any student of HS Fulda who is not part of our project, directly or indirectly.

URL of the system to be tested: http://20.113.70.224/searchTutors

What is to be measured: We have to measure the flexibility of our website, we plan to work further if we find some of the things in the Usability testing is not so easy or comfortable for our users to find/navigate through.

So, for user Satisfaction evaluation we will be using the Likert Test which includes assessing opinions, attitudes, or behaviors of our users.

Usability Task description

In order to perform the test, the tester first need to navigate to the URL provided above and then they should try to search something by putting text into the search box then they should observe that did they find anything relevant or not and to carry forward this test further they can also sort this result according to price or rating. Apart from this testing tester should also check this page by loading in various different sizes of devices like mobile, tablet, laptop to check how UI is getting rendered on different devices. Finally, they should evaluate the result.

To measure effectiveness, first, the tester should check, are results actually according to what they wrote in the search box and if the tester has selected any sorting option then, is the result sorted according to the given criteria?

Questionnaire

Question - 1. How satisfied were you with the UI and Functionality of this page?

1. Very dissatisfied
2. Somewhat dissatisfied.
3. Neither satisfied nor dissatisfied.
4. Somewhat satisfied.
5. Very satisfied.

Question - 2. How well does this page meet your needs?

1. It did not meet my needs at all.
2. It met very few of my needs.
3. It met some of my needs.
4. It met the majority of my needs.
5. It met all of my needs.

Question - 3. How intuitive did you find using Search Box and Sorting Options?

1. Not intuitive at all.
2. Not very intuitive.
3. Somewhat intuitive.
4. Mostly intuitive.
5. Extremely intuitive.

QA Test Plan

Test objectives:

The objective is to test the correctness and robustness of the search feature in the **MentorMe** product. The central idea was to regressively test the search with various filters and text searches from an end-user point of view. The testing methodologies employed was manual and postman-driven automation.

We used industry-standard testing techniques –

1. Smoke Testing
2. Regression Testing
3. Load Testing

Hardware Setup:

Linux based dual-core machine with 2GB RAM on Microsoft Azure Instance.

Software Setup:

1. Node package
2. MySQL Server
3. MySQL Client
4. Nginx Proxy Server
5. Postman (For Testing)

URL: http://20.113.70.224

Feature to be tested: The search feature was taken into consideration while applying the various testing methods.

Test Suite:

Sr. No.	Test Description	Input	Expected Output	Actual Output	Result	Google Chrome	Mozilla Firefox
1.	Responsiveness	Resolution-IpadMini	Screen Adapt to size	Screen adapted to size	Passed	✔	✔
2.	Responsiveness	Resolution – Macbook Pro	Screen Adapt to size	Screen adapted to size	Passed	✔	✔
3.	Case sensitive Tutor Search	Machine LeARning	List tutors with machine learning	Listed tutors with machine learning	Passed	✔	✔
4.	Empty search bar	Blank input	Show entire list	Showed entire list.	Passed	✔	✔
5.	Sorting - Rating	Sort Criteria – Rating	Sorted rate for tutors	Sorted rate for tutors	Passed	✔	✔
6.	Sorting - Price	Sort Criteria – Price	Sorted price for tutors	Sorted price for tutors	Passed	✔	✔
7.	Regression testing	Taking different builds on deployment to see breakage	Search shouldn't break	Search works for last 3 builds	Passed	✔	✔
8.	Load Testing	Multiple client search parallelly	Search shouldn't break	Search works	Passed	✔	✔
9.	Basic Smoke test	Searching for various subjects and tutor names	Tutor list	Tutor list fetched	Passed	✔	✔

Results:

All the various test cases drafted were found to be working as expected on the subject of test. Some of the results are formulated below for reference.

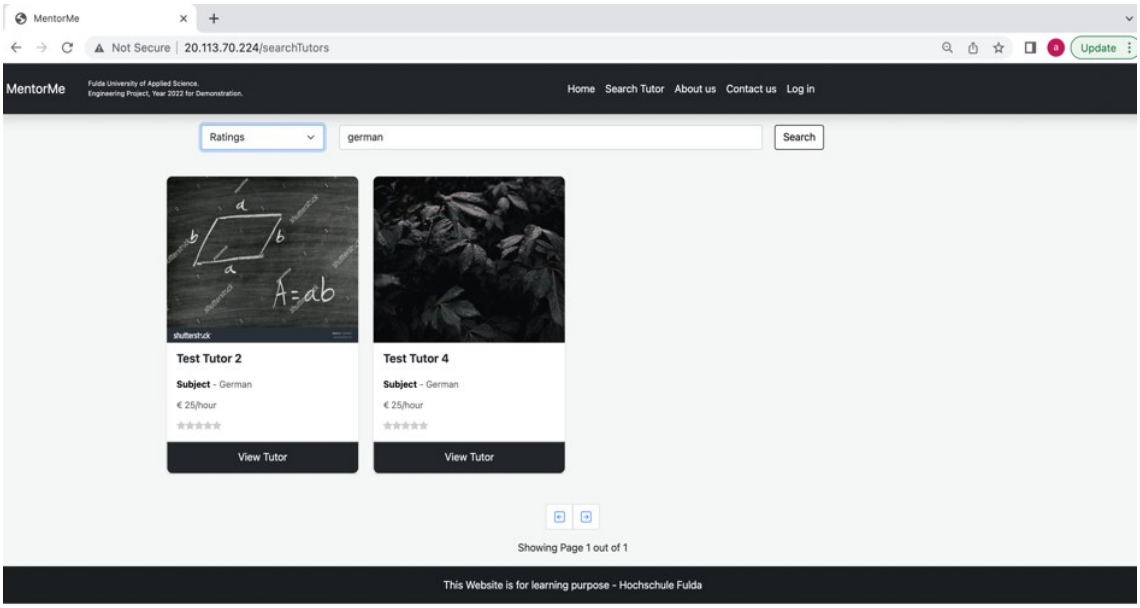

Figure-1: Search tutors teaching German, sort by Ratings

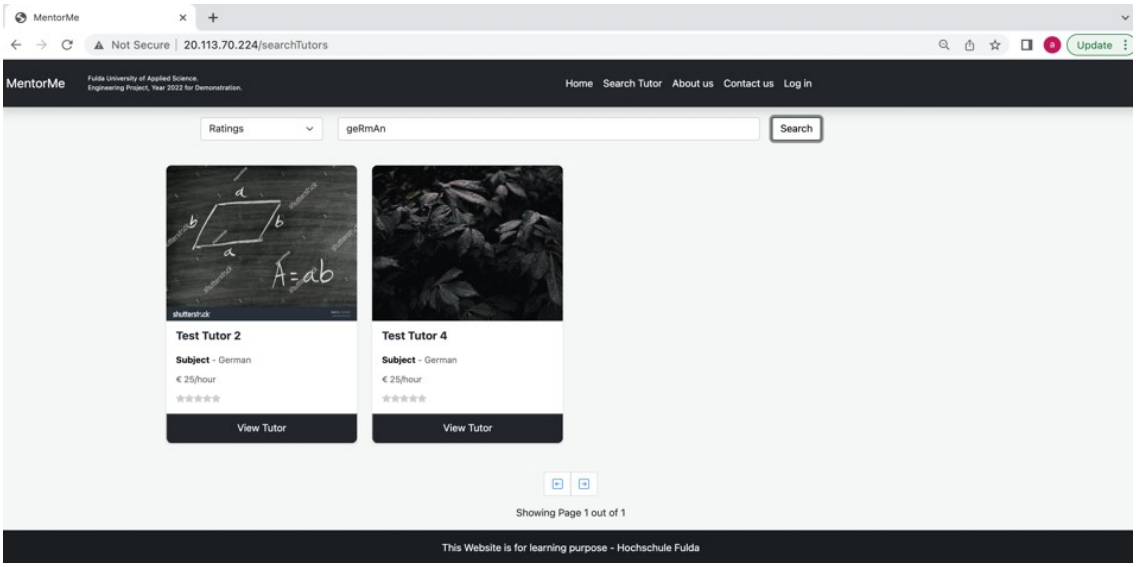

Figure-2: Search tutors teaching German case-sensitive, sort by Ratings

Code Review

We have chosen the portion of the frontend for search tutor and also the backend for the same.

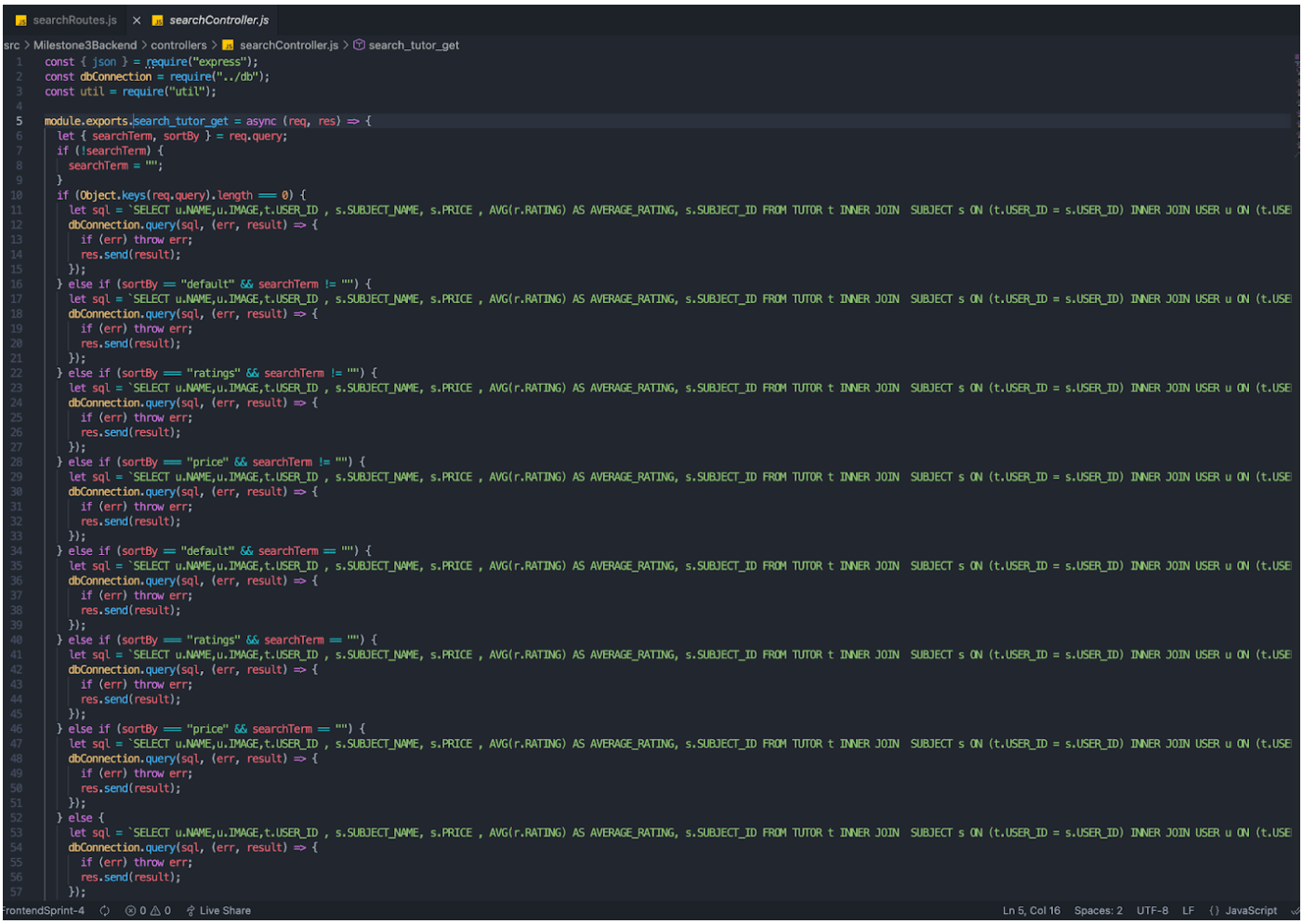

Figure 3: Backend Snippet Screenshot

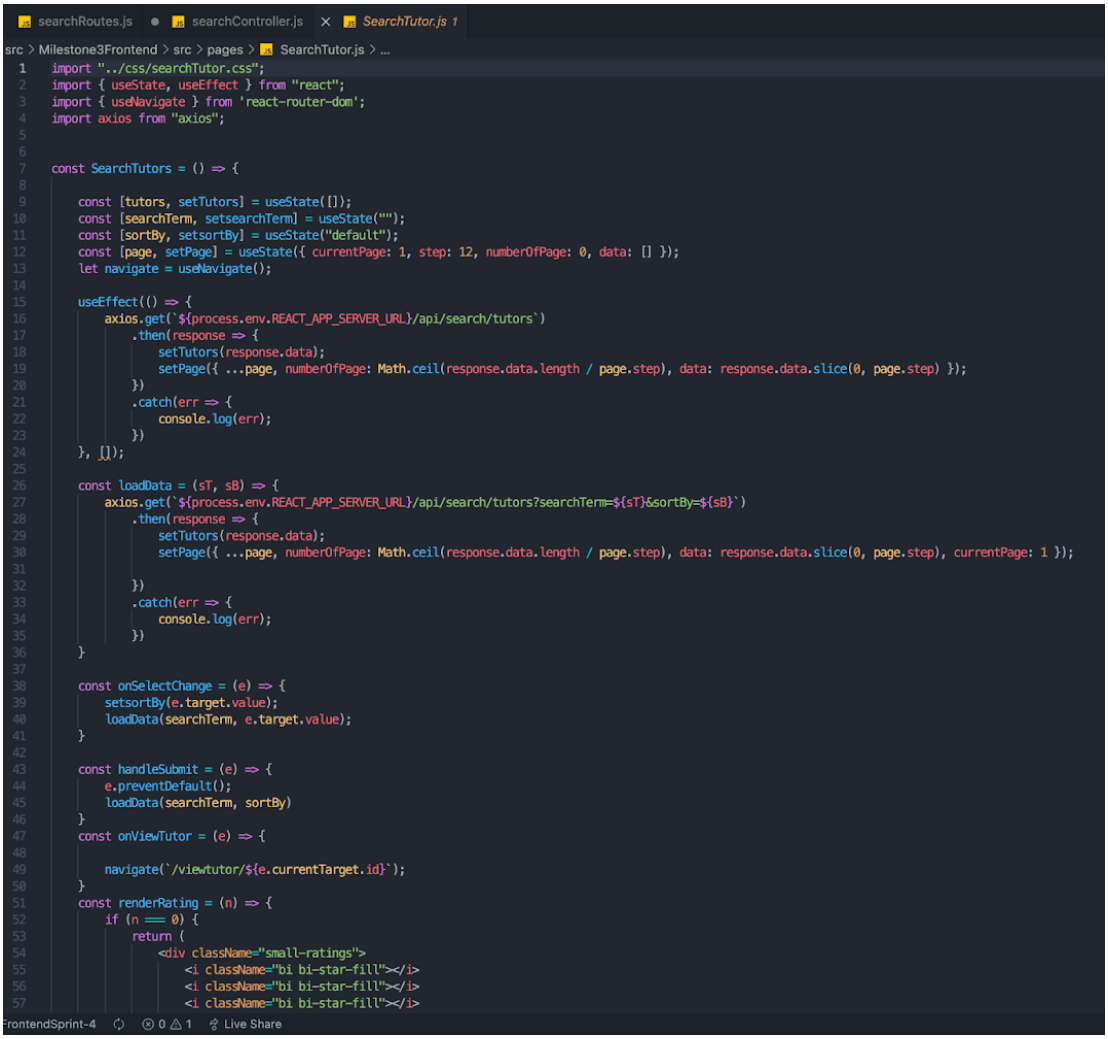

Figure 4: Frontend Snippet Screenshot

Our Steps for code Review:

1. As a peer we assigned each other the file for Searching tutor on backend and frontend using Github review and pull request feature.
2. We commented out improvements on code on a new review branch so that it can be seen by everyone.

Search Controller Review:

Assigned by: Mohammad Afwan Reviewed by: Bibek Gaihre
Link for Review:
https://github.com/Yogeeta31/Student-Tutor/pull/12#pullrequestreview-880864458

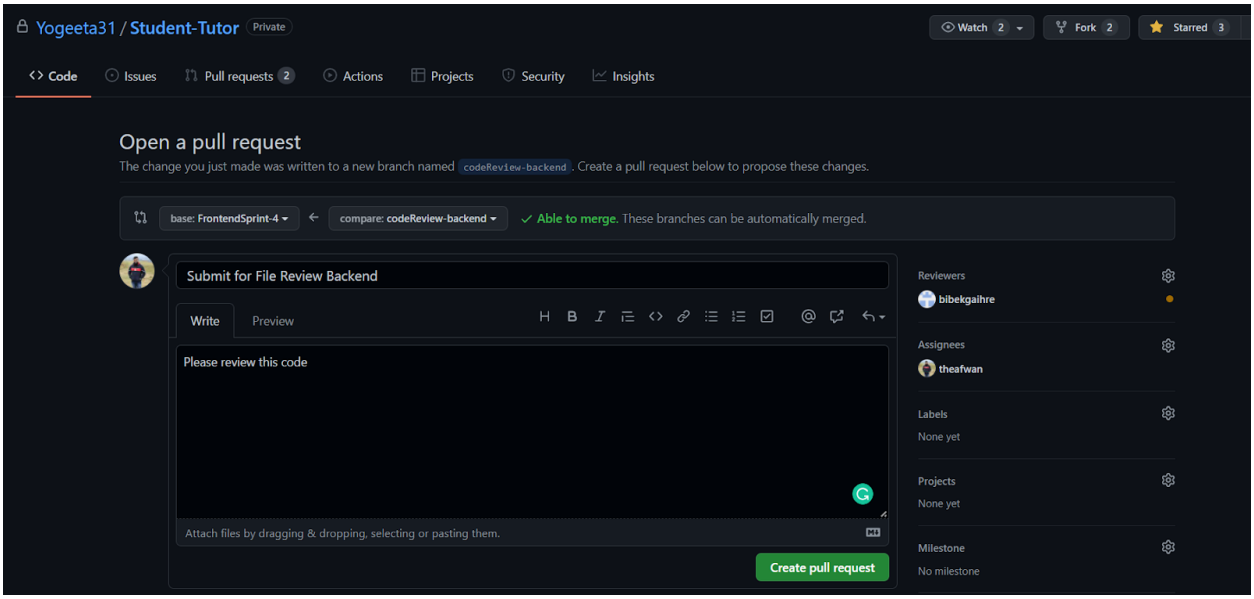

Figure 5: Screenshot 1

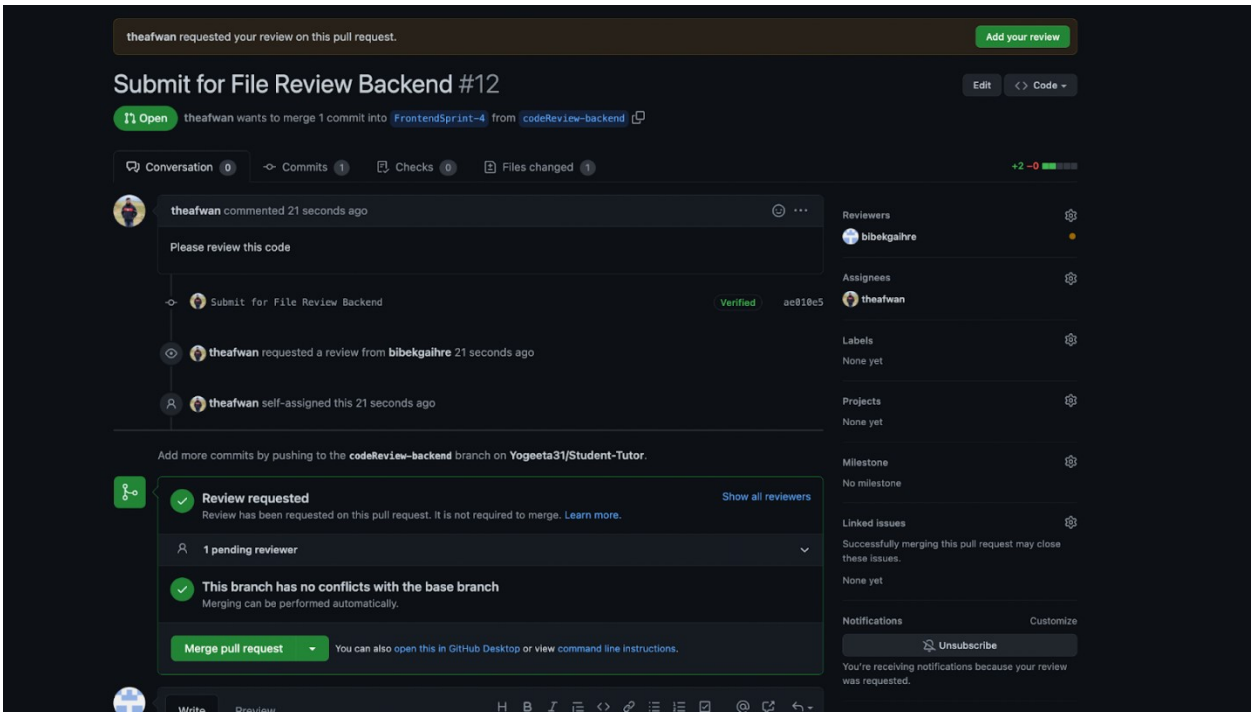

Figure 6: Screenshot 2

Review Comments:

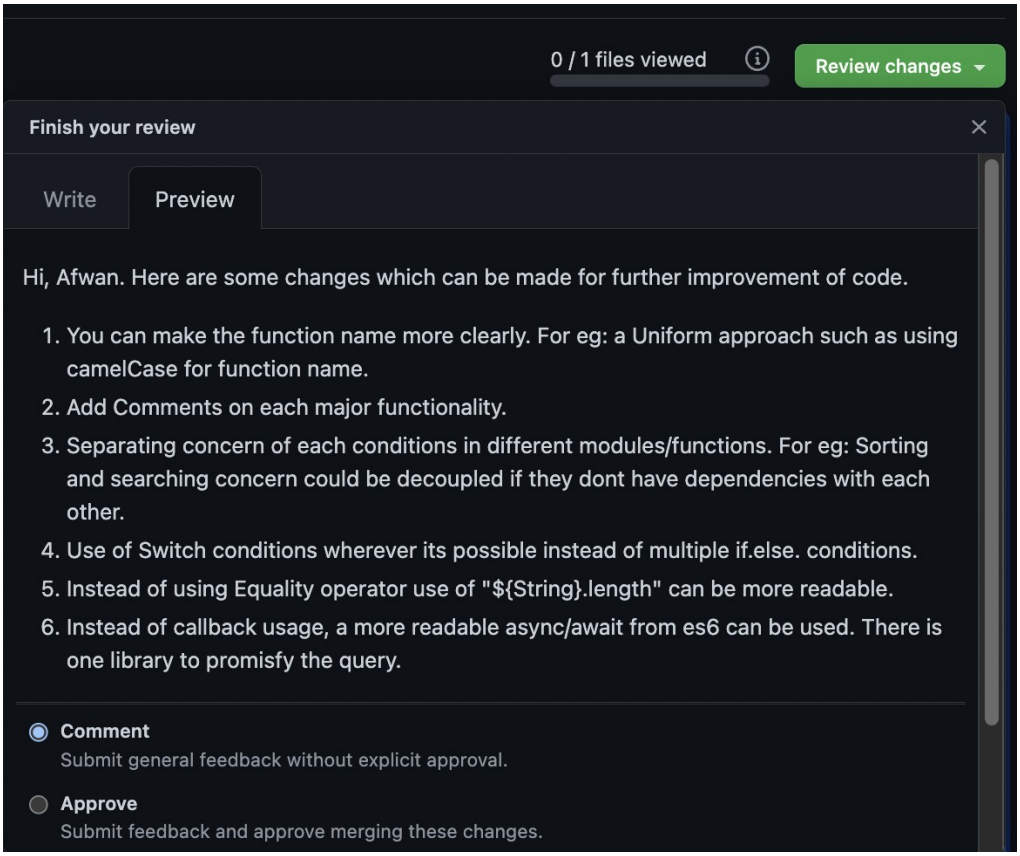

Figure 7: Review written by Bibek

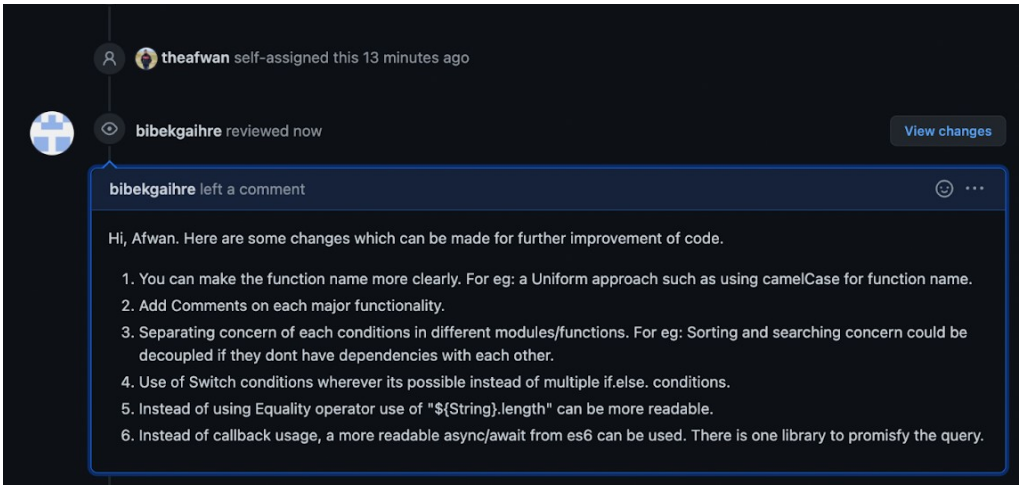

Figure 8: Review received by Afwan

SearchTutor component Review:

Assigned by: Bibek Gaihre Reviewed by: Mohammad Afwan
Link for Review:
https://github.com/Yogeeta31/Student-Tutor/pull/13#pullrequestreview-880866345

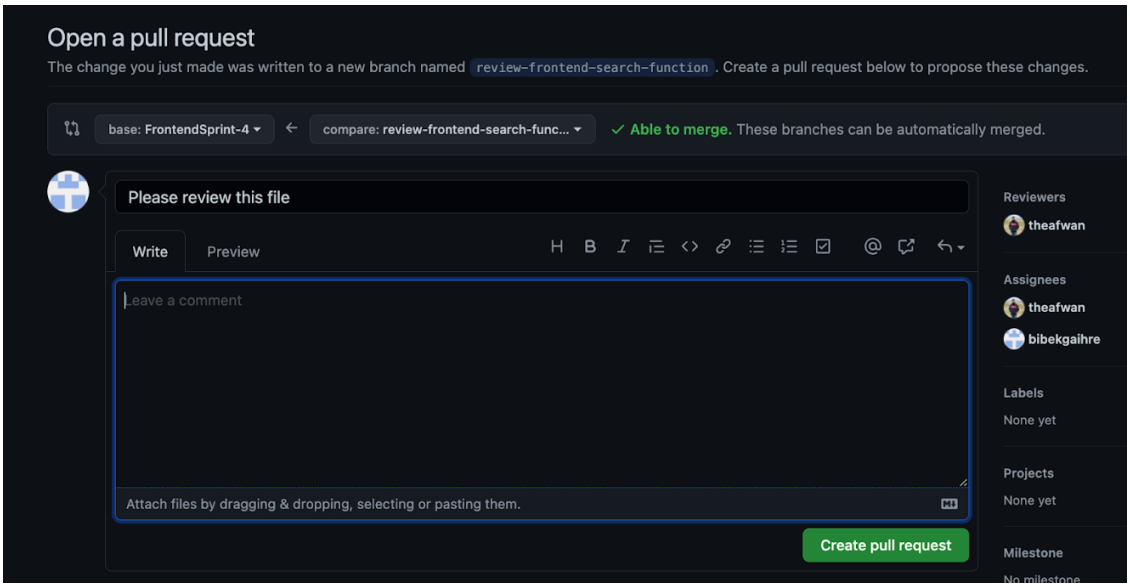

Figure 9:

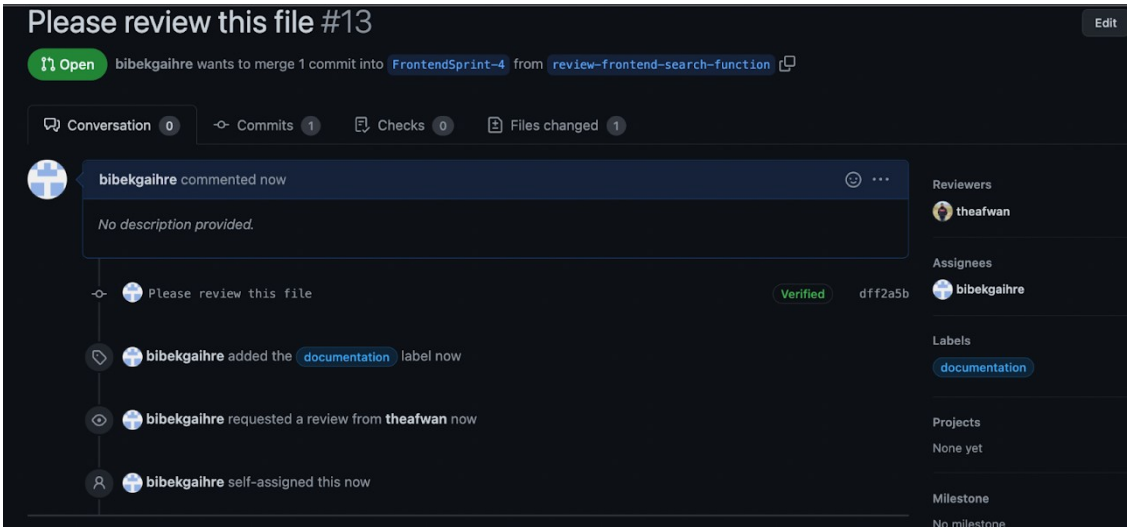

Figure 10:

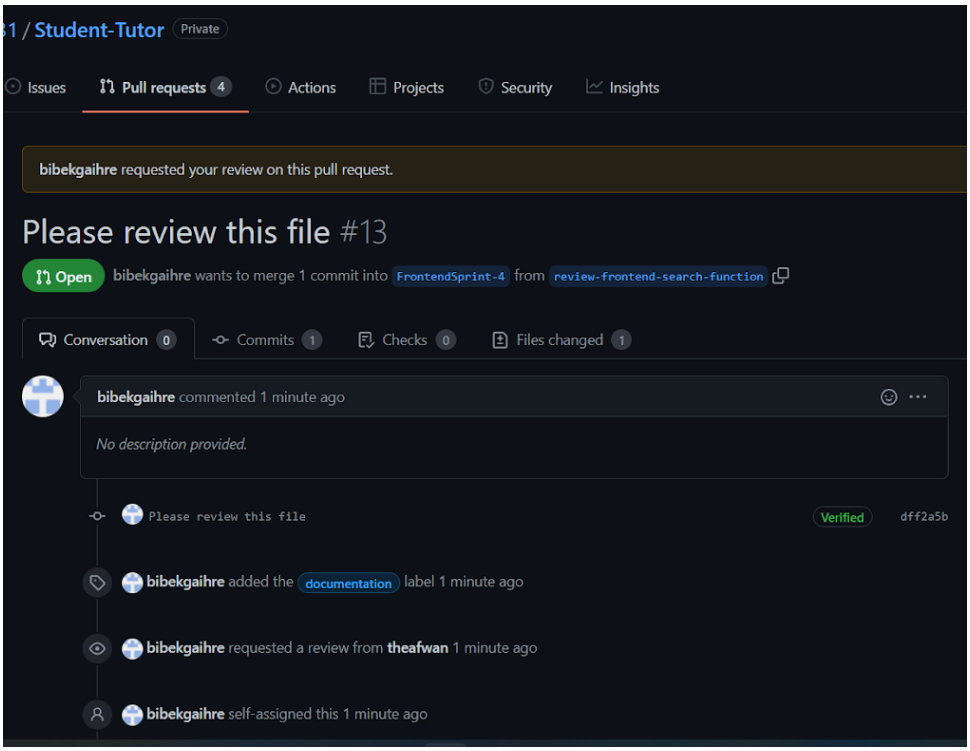

Figure 11:

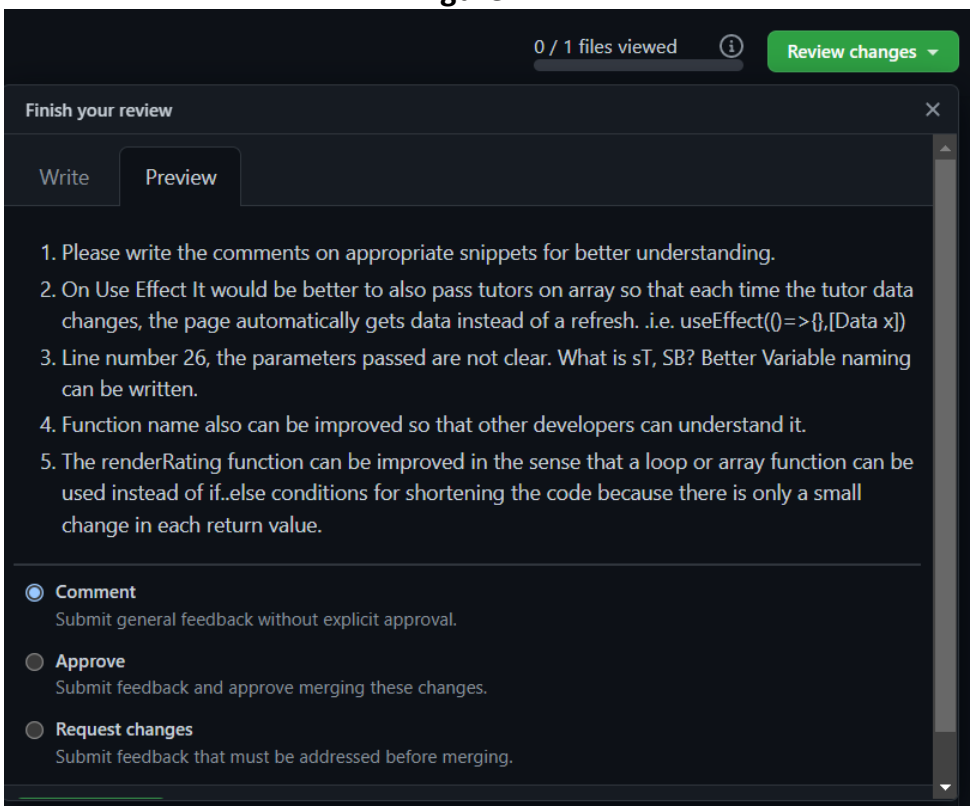

Figure 12:

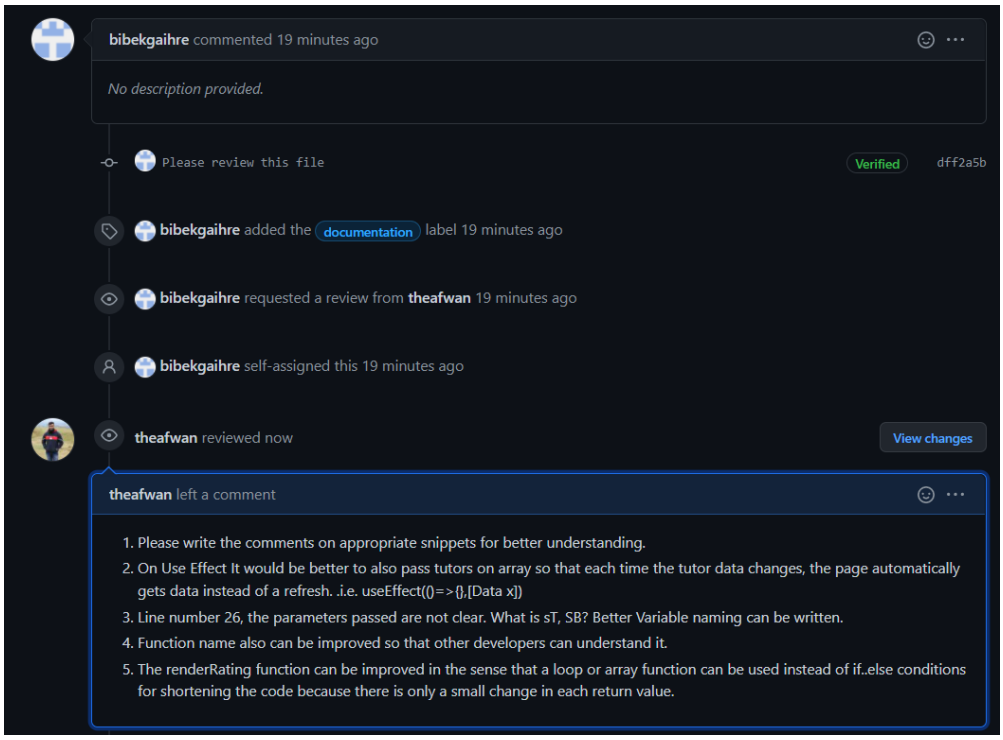
bibekgaihre commented 19 minutes ago
No description provided.
Please review this file
Verified
dff2a5b
bibekgaihre added the documentation label 19 minutes ago
bibekgaihre requested a review from theafwan 19 minutes ago
bibekgaihre self-assigned this 19 minutes ago
theafwan reviewed now
View changes
theafwan left a comment
1. Please write the comments on appropriate snippets for better understanding.
2. On Use Effect It would be better to also pass tutors on array so that each time the tutor data changes, the page automatically gets data instead of a refresh. .i.e. useEffect(()=>{},[Data x])
3. Line number 26, the parameters passed are not clear. What is sT, SB? Better Variable naming can be written.
4. Function name also can be improved so that other developers can understand it.
5. The renderRating function can be improved in the sense that a loop or array function can be used instead of if..else conditions for shortening the code because there is only a small change in each return value.

Figure 13:

Self-check on best practices for security

Major Assets to Protect

1. **User Content**
    - Profile Information
    - Passwords
    - Profile Image
2. **Tutor Specific Content**
    - CV
    - Student Reviews and Ratings (Tutor Profile Info.)
    - Subjects (Tutor Profile Info.)
    - Bio (Tutor Profile Info.)
3. **Chats**

Major Threats on Assets and Protection Mechanism

1. **User Content**
    - Profile Information: Only the user with valid credentials should be able to access this information. To protect this we have implemented login.
    - Passwords: If the database gets exposed or breached, then illegal parties can access the credentials. In order to protect this we are doing one way hashing and then storing it in the Database.
    - Profile Image: Profile image could be collected by some illegal/unauthorized party, to protect this we are just saving randomly generated unique name of the image in the database, and uploading real data to the secure Azure cloud.
2. **Tutor Specific Content**
    - CV: CV File could be collected by some illegal/unauthorized party, to protect this we are just saving randomly generated unique names of the file in the database, and uploading real data to the secure Azure cloud.
3. **Chats**
    - Chats Should be one to one and no other user should be able to access others chats, we are maintaining one to one chatting by uniquely identifying each message with the help of userID.

We are encrypting Passwords in the Database!

- We are validating the input data on each and every form, for empty fields and field specific information with the help of regular expression string comparison.

Self-check: Adherence to original Non-functional specs – performed by team leads