

Test Report

API

The backend is run using node server.js locally or in the docker container for backend. After the server is run, it should print out messages showing that the database, tables, and admin user were created. No error should be reported in this process (and my test passed that no error was reported).

I tested the backend using **loadtest** for sending bulk requests and **Httpie** (<https://httpie.io>) for sending individual requests.

Each API request and response should be correctly formatted and have the required field in the body of the JSON sent, which I checked using Httpie and they all passed the test.

Database

I tested the database by running the backend locally and used **MySQL workbench** to check if the tables and their columns and rows were created/updated/deleted as expected.

Every requested change from the backend to the database was reflected correctly in MySQL workbench.

Some tables in the database have foreign key constraints that perform cascade deletion. This was tested by deleting the parent table row and all the cascade deletions worked successfully by deleting the children table rows.

Frontend

I tested the frontend by creating a new user and posting new channels, questions, and comments. All these operations were reflected correctly in the front end. I tested that without logging in a user cannot access the mainwindow by changing the address to /mainwindow and it was not accessible, so the test passed. I logged in as admin and I was able to see the fully functioning delete buttons for channels, questions, and comments.

Searching for keywords and users worked correctly in all 3 panels.

I used multiple browser windows to login as different users and post new channels, questions, and comments. The changes were reflected correctly for other users.

I tested the like/dislike buttons by pressing them multiple times. It only works on the first time which is expected because each user can like/dislike only once. This like/dislike is visible to other users and is incremented every time a user likes/dislikes a comment for the first time.