CPSC 319
Zenith Blog
User Guide

Presented By:

Muhammad Saad Shahid
Param Tully
Eric Wong
Cheryl Yu
Anusha Saleem
Shawn Zhu
Andrew Liu
Ruchir Malik
Anthony Baek

Sponsors:

Evan Seabrook
Alvin Madar

TA:

Asem Ghaleb

Version: 1
Date: Apr/2/2023

INDEX

PREFACE

Our blog behaves differently depending on user permissions. The following table highlights the permissions for the 4 key users of our blog, Guest, Reader, Contributor, and Admin. Please use the login credentials given to explore different functionalities of Zenith Blog.
(Guests are those who do not login).

	Reader Username: zenithreader@gmail.com Password: zenithreader1!	Contributor Username: zenithcontributor@gmail.com Password: zenithcontributor1!	Admin Username: zenithblogteam@gmail.com Password: cpsc319zenith
View Posts/ Comments			
Create/ Delete Posts		• Can only delete/edit their OWN posts	• Can create posts and delete any post, but cannot edit other users' posts
Create/ Delete Comments		• Can only delete/edit their OWN comments	• Can create comments and delete any comment, but cannot edit other users' comments
Upvote/ Downvote Posts			
Upvote/ Downvote Comments			
Grant/ Deny Contributor Permissions			

1.0 VIEW POSTS AS A GUEST

Viewing posts on our application is easy and can be performed by any user irrespective of their type. Follow the steps below:

- Go to zenithblog.ca
- You'll be redirected to the home page of our application where you'll see a web carousel and multiple tiles below it (see fig 1.1 below).

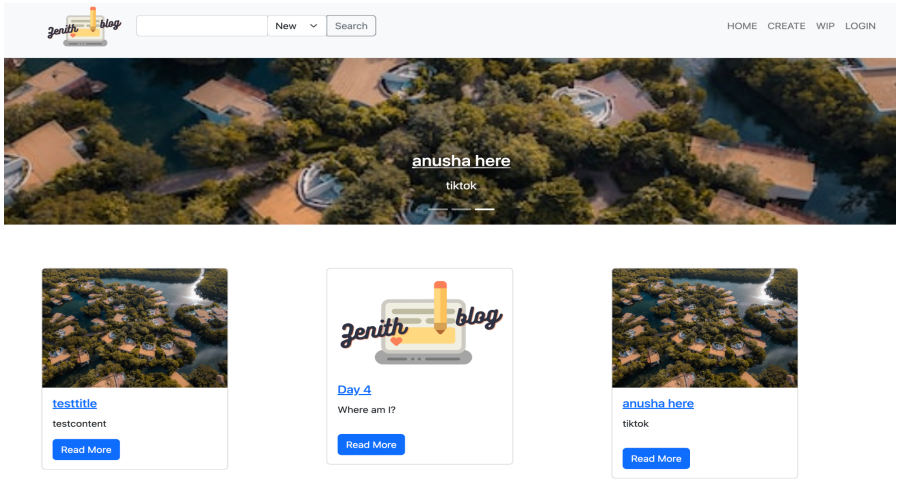


Fig 1.1

- The carousel and each tile below it is a summarized post and can be opened by clicking on the image or the 'Read More' button.

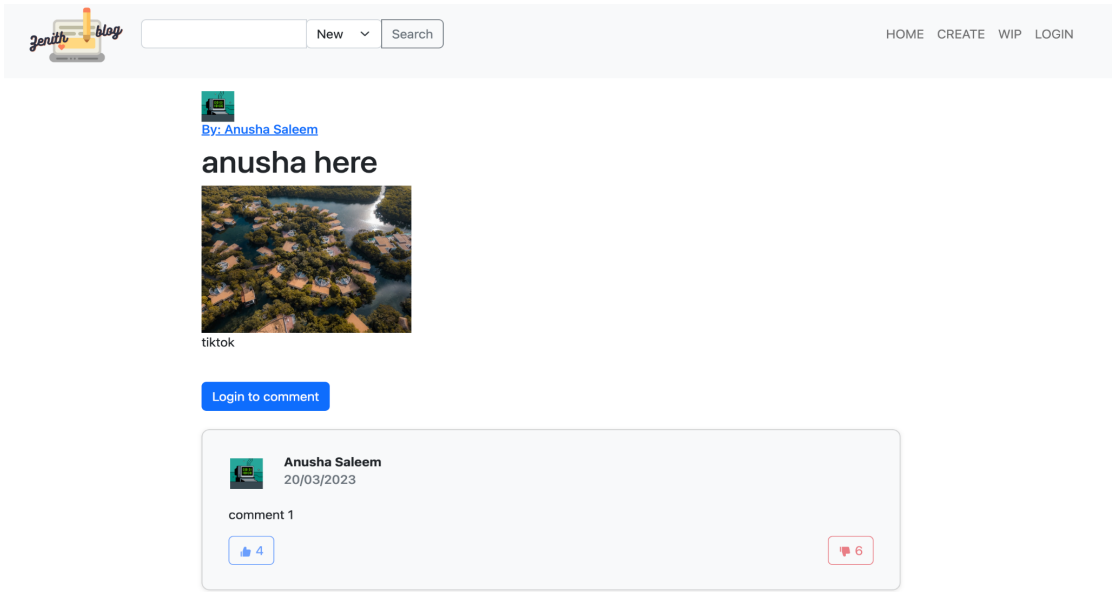


Fig 1.2

2.0 (HOW TO) LOGIN TO OUR SYSTEM

Zenith blogs uses the Google OAuth 2.0 (?) protocol for authorization and authentication. You become a Reader as soon as you login to our system. Follow these simple steps below to login to our system:

- Find the ‘Login’ button on the top-left corner of our application and click on it.

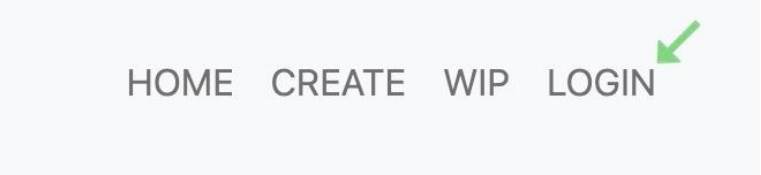


Fig 2.1

- You’ll be redirected to our Login page where you can easily login through your Google Account.

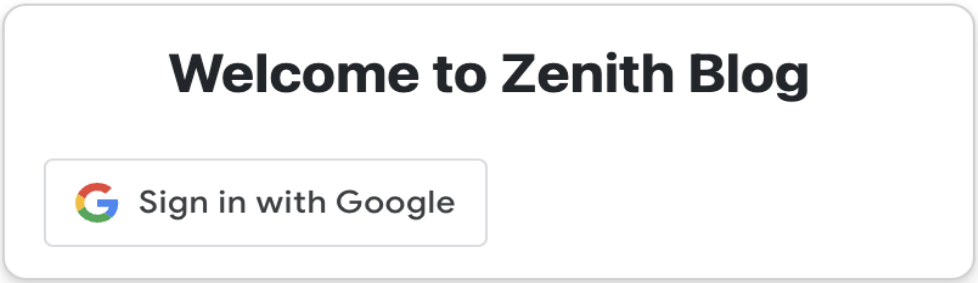


Fig 2.2

3.0 VIEW USER PROFILE

Once you've logged into Zenith Blog using your Google Account, you can view your profile. The profile page displays your 'account creation date', 'last login date' and the user status (i.e READER, CONTRIBUTOR or ADMIN). The page also has an option to request a level upgrade. Follow the simple instructions below ***after logging in*** to view the profile:

- Click on the 'Profile' button on the top-right corner of your screen.

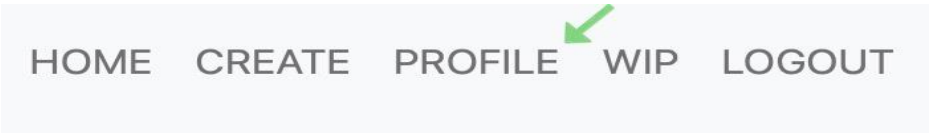


Fig 3.1

- You'll be redirected to the Profile Page, which will look something like Fig 3.2 below.

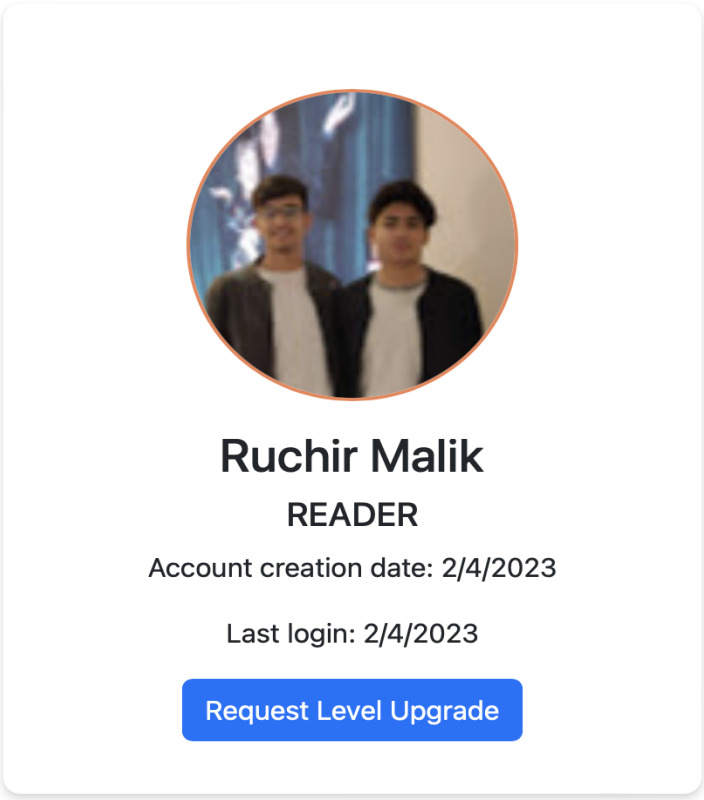
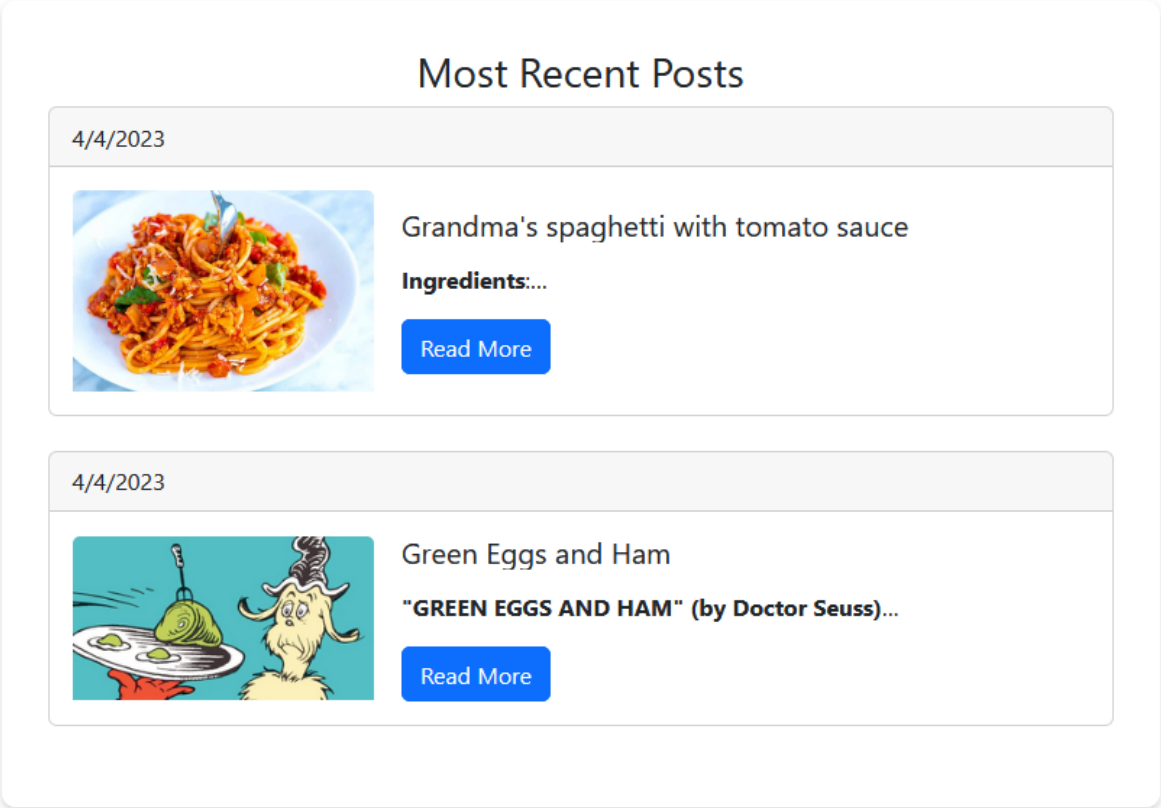


Fig 3.2

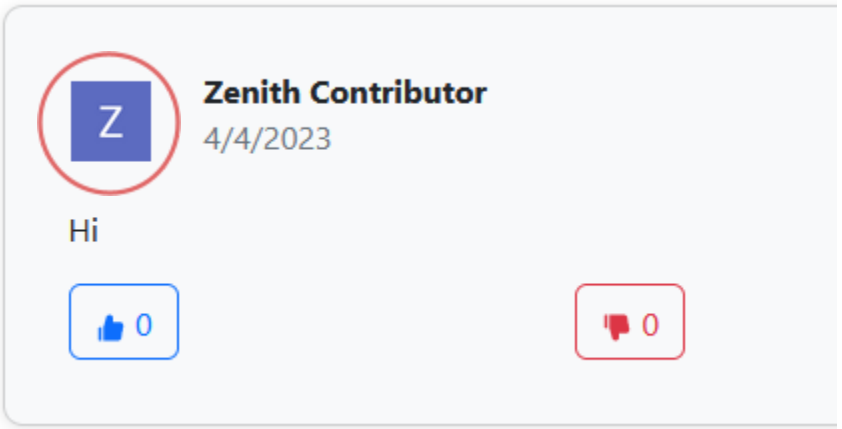
Profiles also have a list of the 5 most recent posts the user made. Each post can be visited by clicking on the image or the 'Read More' button.



You can also visit other users' profiles by clicking on the user profile pictures in posts.

4.0 UPVOTE/DOWNVOTE A POST

A user ***must be logged*** in to be able to



upvote/downvote a post. In other words, any user except a GUEST user can like/dislike a post.

Upvoting/downvoting a post is easy and can be achieved through the following steps:

- Make sure you're logged in to our system.
- Go to any post you'd want to upvote or downvote (see section 1.0). Refer to Fig 4.1 to see a sample post with **1 upvote and 0 downvotes.**

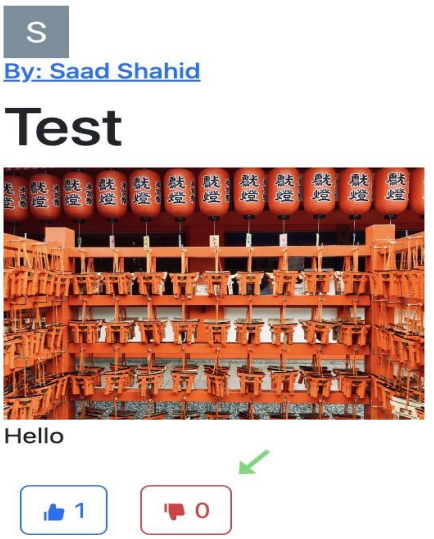


Fig 4.1

- Click on the blue thumbs-up button to upvote or the red thumbs-down button to downvote the post. Note that you CANNOT both upvote and downvote the same post. Clicking twice on any button undoes the action. Fig 4.2 shows the post after it has been upvoted by us.

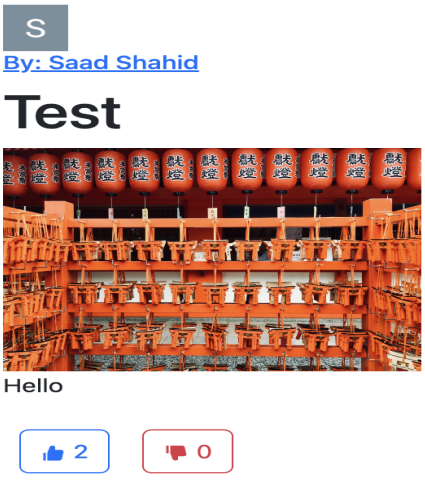


Fig 4.2

5.0 MAKING COMMENTS ON POSTS

Just like upvotes and downvotes, a user ***must be logged in*** to be able to make comments on a post. In other words, any user except a GUEST user can comment on a post. Follow the steps below to make a comment:

- Go to a post that you'd want to comment on (make sure you're logged in first). Refer to Fig 5.1 to see a sample post with a text box under 'Leave a comment'.

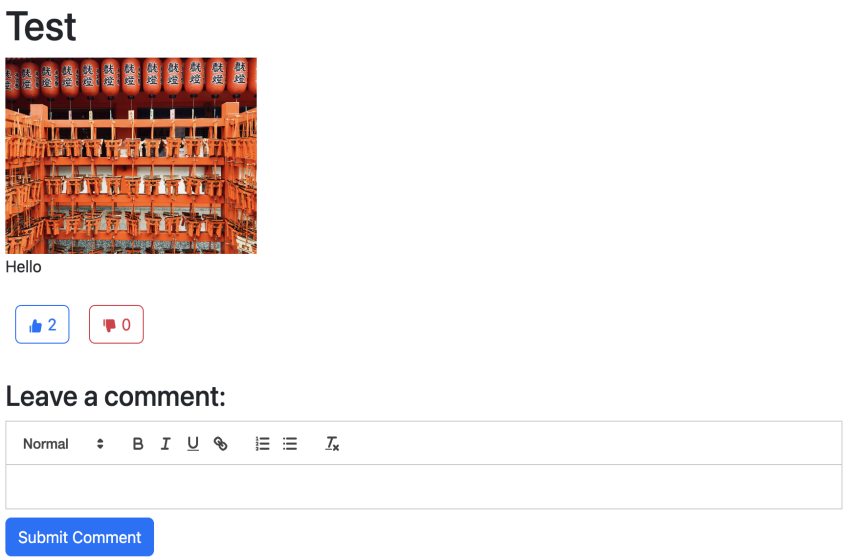


Fig 5.1

- Enter the text that you want to include in the comment and edit it as you wish. Click the 'Submit Comment' button to submit the comment. Fig 5.2 shows what a comment looks like once it has been submitted.

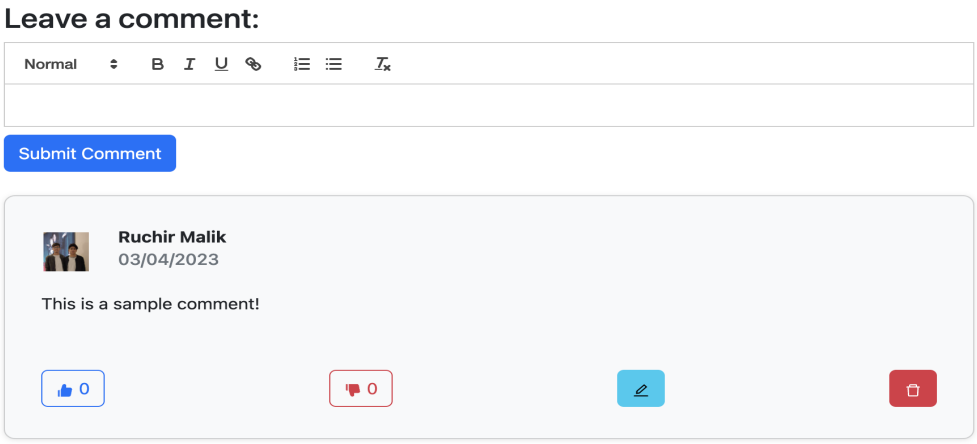


Fig 5.2

6.0 DELETE COMMENTS

Once you've made your comment, it's very easy to delete it. It's important to note that you can only delete your comment.

Follow the steps below to delete one of your comments:

- Any comment that has been made by you will have a red bin-button on the bottom-right corner. Refer to the green arrow in Fig 6.1 to see the 'delete' button.

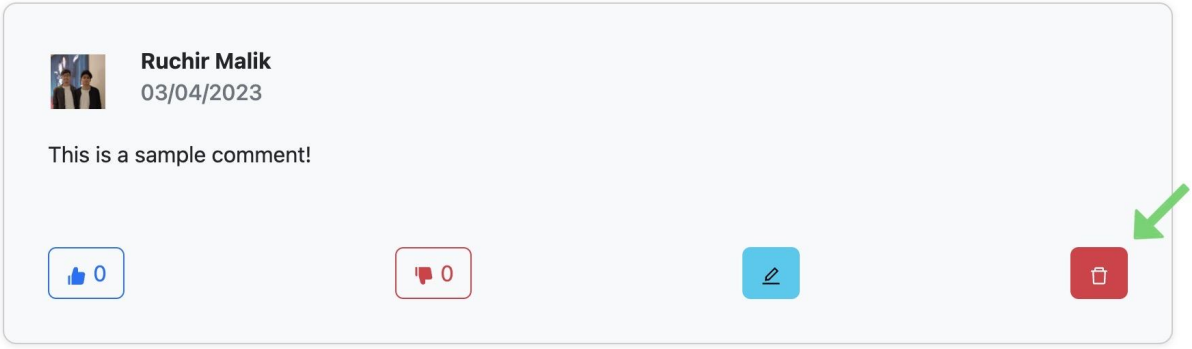


Fig 6.1

- Click on it and your comment disappears!

7.0 EDIT COMMENTS

Editing a comment is easy and can be achieved with the click of a button. Similar to how deleting a comment works, you can only edit your comment.

Follow the steps shown below to edit a comment:

- Any comment that has been made by you will have a blue pencil-button at the bottom of it. Refer to the green arrow in Fig 7.1 to see the 'edit' button.

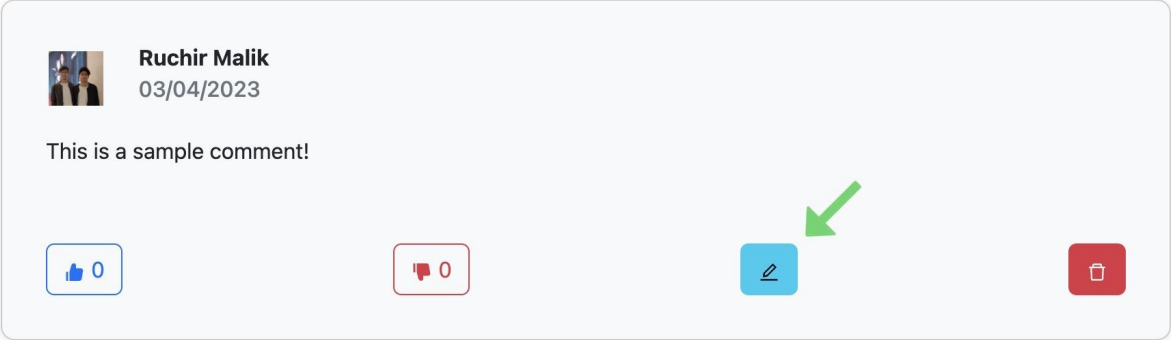


Fig 7.1

- Clicking on it reopens the text-box for you with your current comment already in it. You can edit as you like and click on ‘Save’ to save the changes or on ‘Cancel’ to undo the changes. Fig 7.2 shows what happens when you click on the ‘edit’ button on one of your comments.

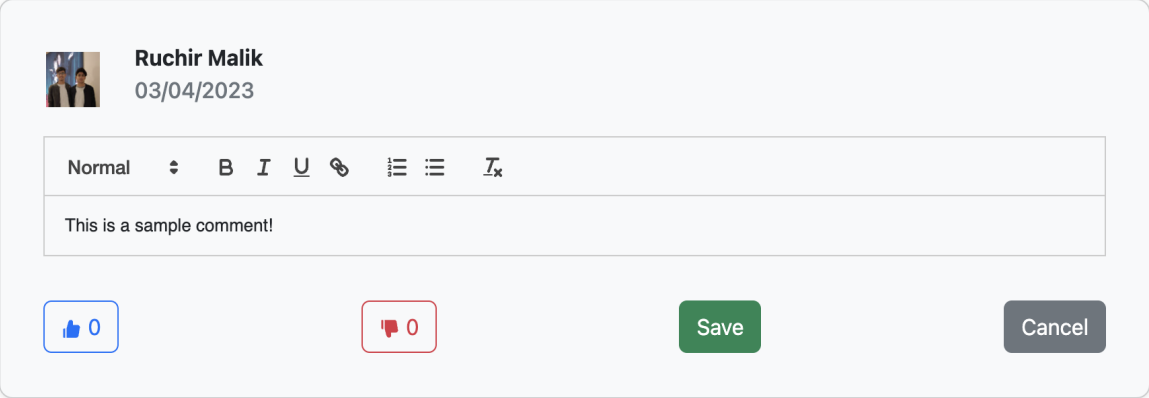


Fig 7.2

8.0 UPVOTE/DOWNVOTE A COMMENT

Upvoting or downvoting a comment is very similar to upvoting or downvoting a post. Moreover, you can upvote/downvote anyone’s comment on anyone’s post ***as long as you’re signed in***. Follow the steps below to upvote/downvote a comment:

- Login to our system.
- Go to a post whose comment you want to upvote/downvote (see section 1.0).

- Go to the comment you want to upvote/downvote. Fig 8.1 shows a random comment from one of the posts in our application.

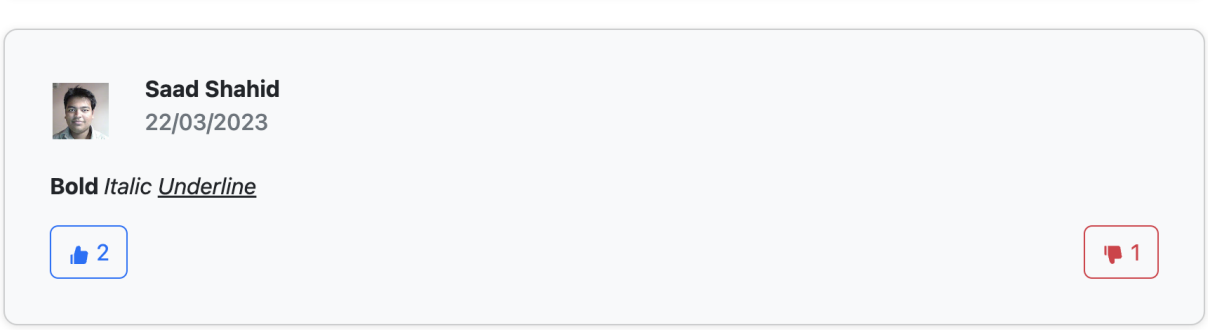


Fig 8.1

- Click the blue thumbs-up button to upvote or the red thumbs-down button to downvote the comment. Fig 8.2 shows the same comment after having upvoted it.

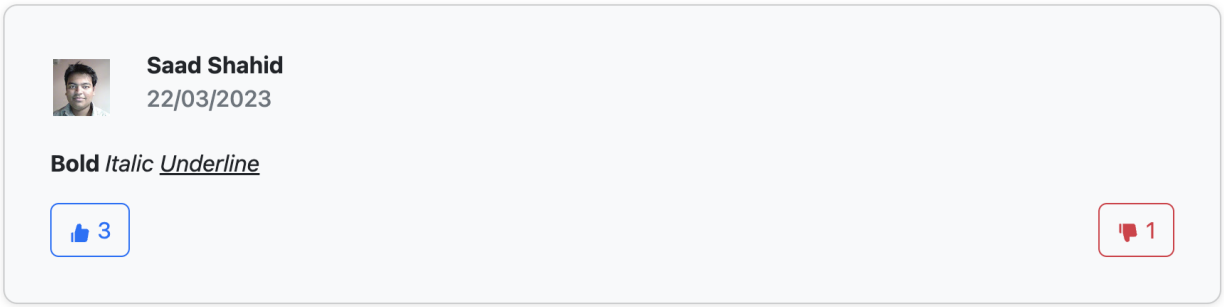


Fig 8.2

9.0 CREATE A POST

Creating a post is one of the most fun tasks and is as simple as it gets! However, a user must either be a CONTRIBUTOR or an ADMIN to be able make a post. Follow these simple steps below to create your first post:

- Make sure ***you're logged in*** and are either a Contributor or an Admin.

- Hit the ‘Create’ button from the menu on the top-left corner of our application. If you’re not logged in yet, it’ll ask you to login before you can create a post.

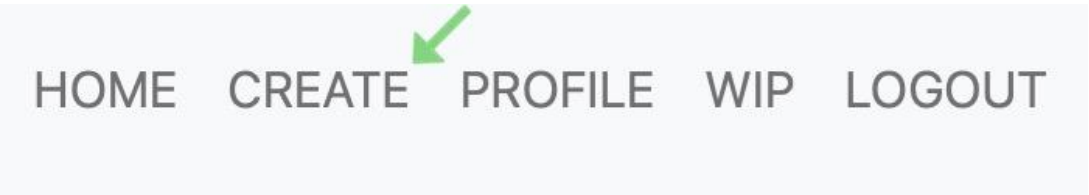


Fig 9.1

- The create post dashboard can be seen in fig 9.2 below. Enter title, content and upload image (optional) and hit Publish when you’re ready to publish your post on our application.

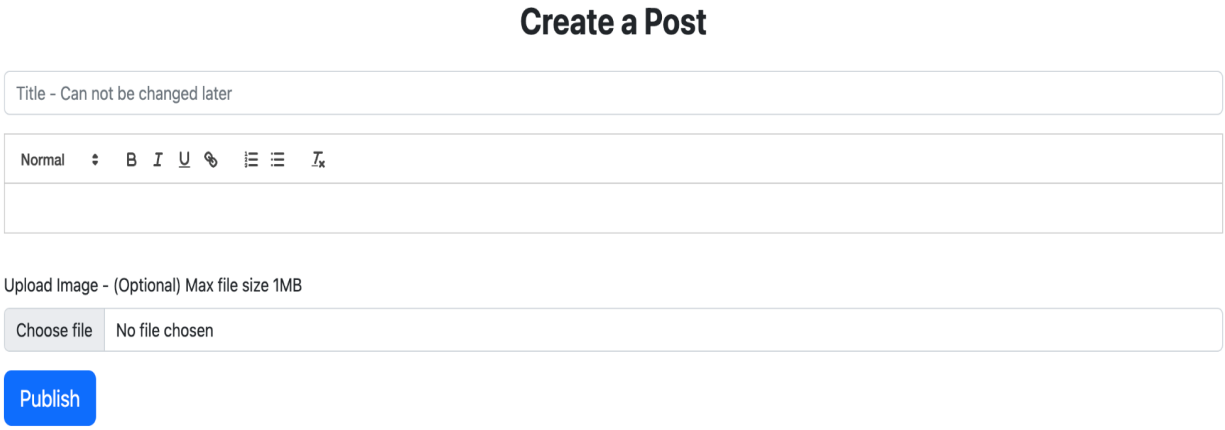


Fig 9.2

10.0 DELETE A POST

Deleting a post is as easy as creating one and can be deleted with the click of a button. However, a CONTRIBUTOR can delete only their own posts while an ADMIN can delete any post on our application.

Follow the steps below to delete one of your posts:

- Go to the post you want to delete once you’re logged in.
- You’ll see a red bin-button at the bottom right corner under your post. This is the delete button. See fig. 10.1.

Fig 10.1

- Hit the ‘delete’ button and your post disappears!
- Similarly if you’re an ADMIN, you’ll see the delete button under every post, which means you can delete any post irrespective of who has created it.

11.0 EDIT A POST

A CONTRIBUTOR can edit their and only their posts. Not even an ADMIN can edit someone else’s posts. Similarly, an ADMIN can edit their own posts. Follow the steps below to edit a post:

- Go to the post you want to edit.
- You’ll see a blue pencil button at the bottom right corner under your post. This is the ‘edit’ button. See the green arrow in fig 11.1 to find the ‘edit’ button.

Fig 11.1

- Click on it and the text box will reopen. Make changes as desired and hit ‘save’ to keep those changes. Hit ‘cancel’ to revert to the original post. Fig 11.2 shows what happens when you click on the ‘edit’ button on one of your posts.

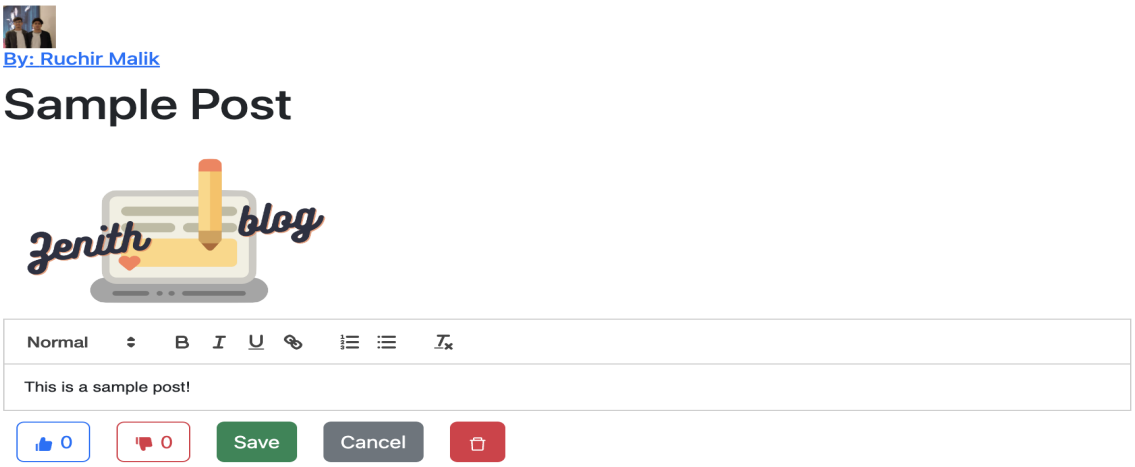


Fig 11.2

12.0 SEARCH AND SORT POSTS

Search is a key feature in any application. Our implementation of search is easy to use and efficient. Note that all four user types can perform search on our system.

Follow the steps below to perform search:

- Go to zenithblog.ca

- Look for the search bar on the top-left corner of your screen. The green arrow in fig 12.1 points towards the search bar.

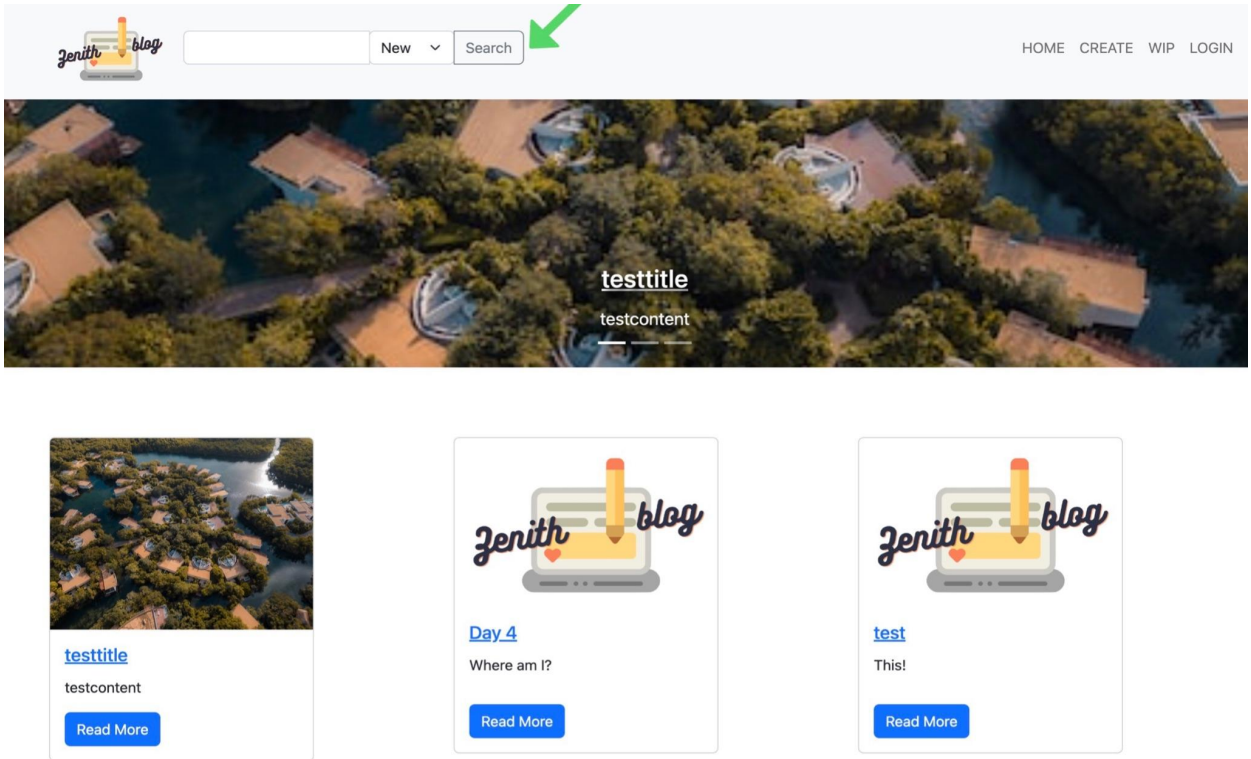


Fig 12.1

- Type in the keywords you want to use to search the posts, select the desired search filter from the drop-down menu and hit the search button!

- The posts containing the keyword(s) that you used to perform search will be filtered and displayed in front of you. Fig 12.2 displays results of a search that uses the keyword ‘hello’.

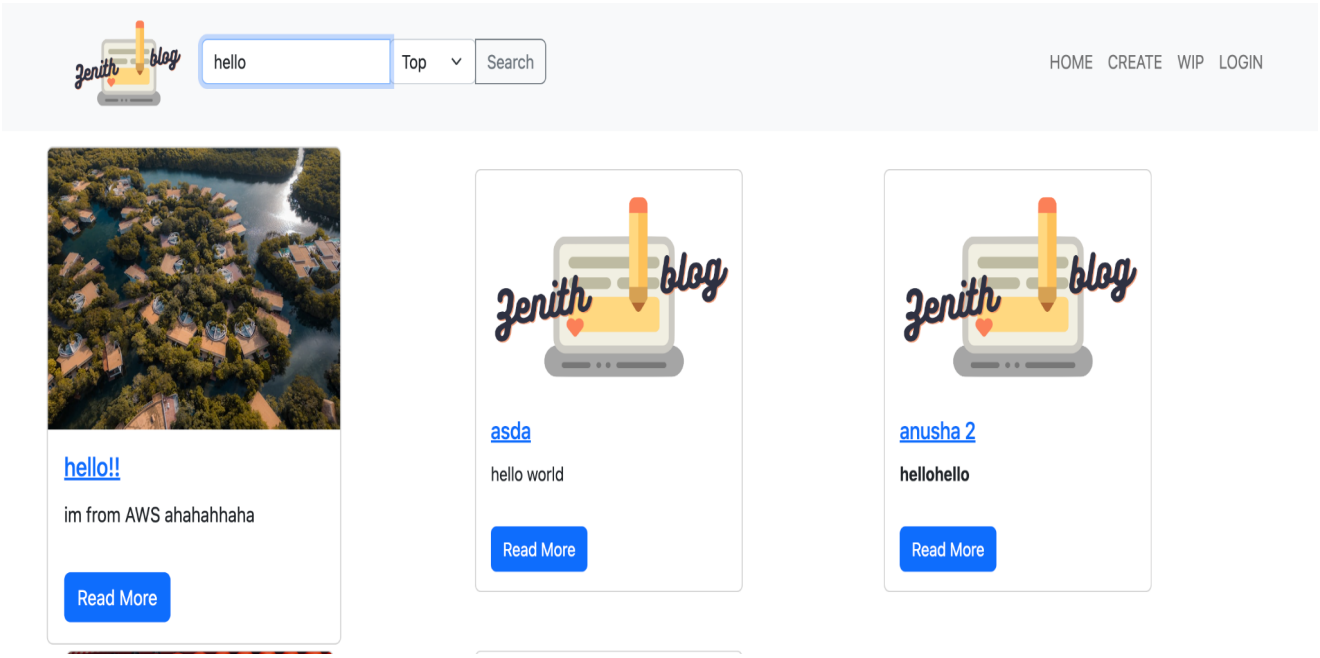


Fig 12.2

As far as **sorting of posts** is concerned, we have a toggle button that enables sorting of posts from either ‘oldest to newest’ or ‘newest to oldest’. The green arrow in fig 12.3 points towards the toggle sort button on our application.

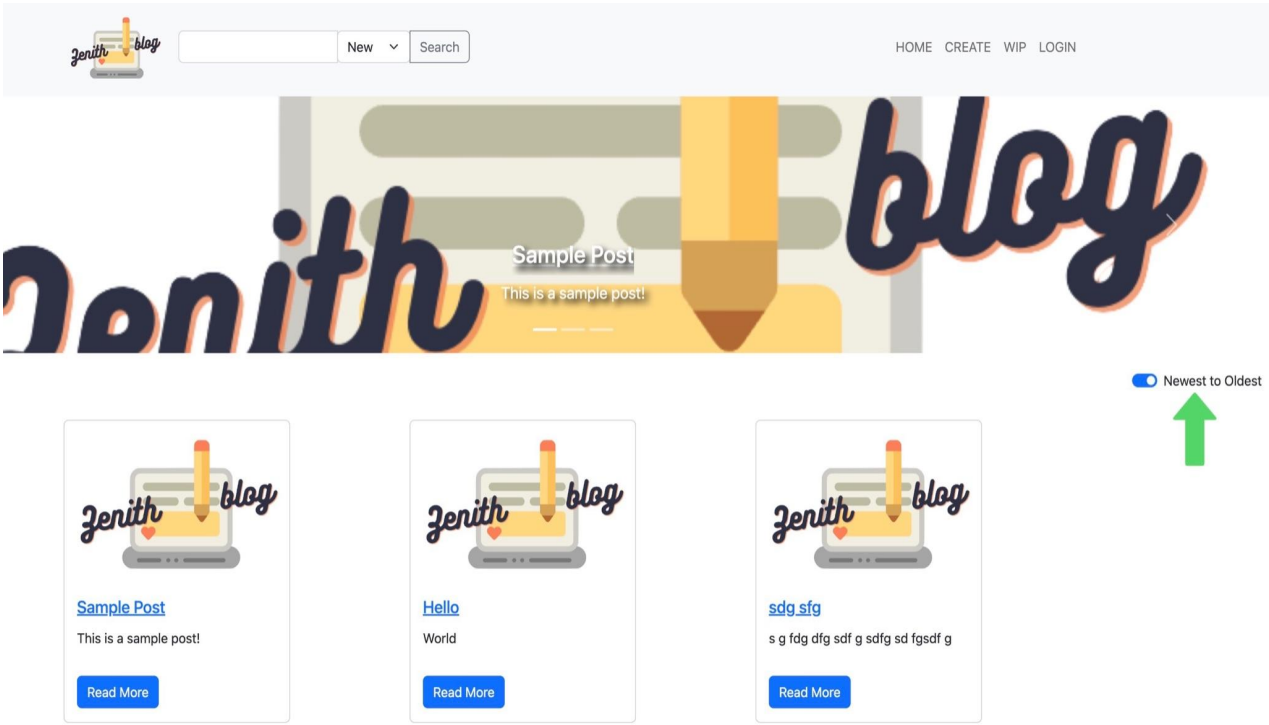


Fig 12.3

13.0 USER LEVEL UPGRADE REQUEST

A user is a guest user unless they've logged in. And when a user first logs in to our system, they're assigned the status of a READER, which means that they can upvote/downvote posts and comments, and make comments on posts that already exist. However, what a READER cannot do is create new posts. The user must be a CONTRIBUTOR or an ADMIN to be able to create posts. For a READER to become a CONTRIBUTOR or an ADMIN, they must send the current ADMIN(s) a level upgrade request. ADMIN has the power to accept or decline that request.

Follow the steps below to send a user level upgrade request:

- Make sure you're logged in.
- If this is your first time logging in to our system, you are a READER. You can check your user level by viewing your profile (see section 3.0).
- Go to your profile, you'll see a 'Request Level Upgrade' button (fig 13.1).

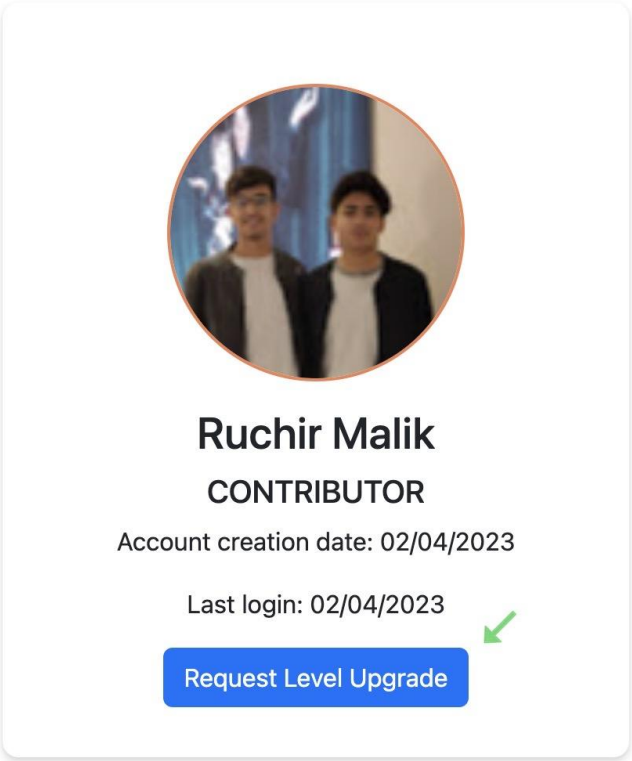


Fig 13.1

- Once you click on it, fill in the position you would like to get promoted to and state the reason. Hit ‘Submit Request’ and wait for an ADMIN to respond to your request (see fig 13.2).

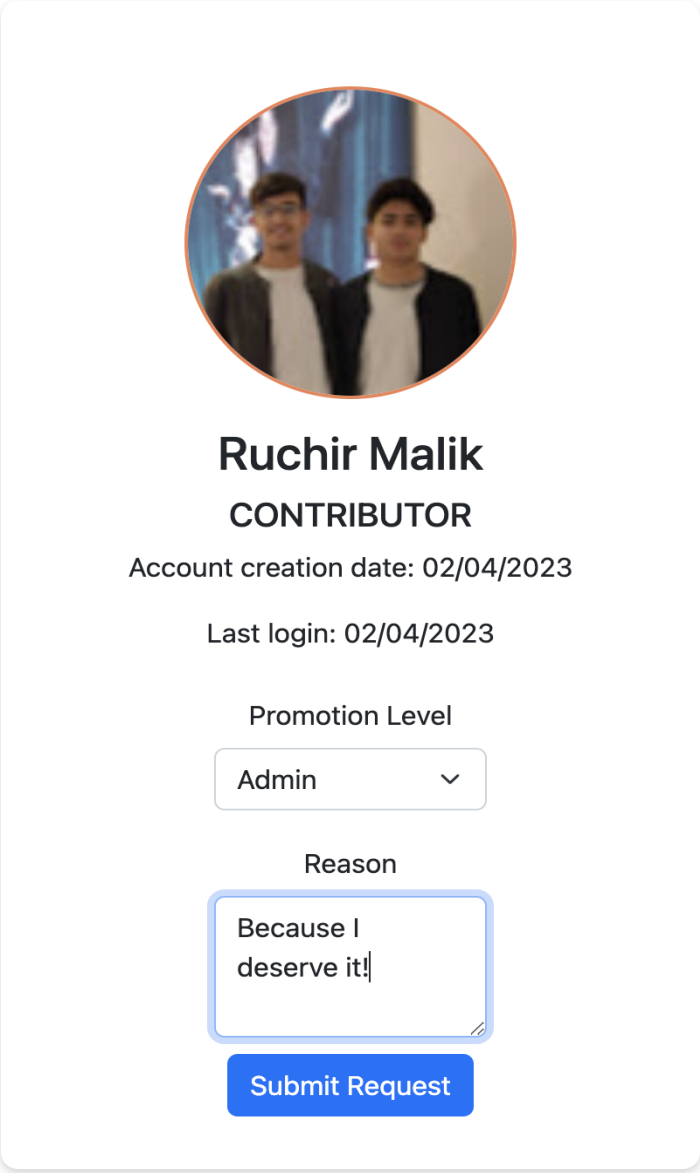


Fig 13.2

14.0 RESPOND TO AN UPGRADE REQUEST AS AN ADMIN

Any ADMIN on our system gets access to an additional tab where they can view pending upgrade requests. They have the authority to approve or reject any of them. To view pending requests, follow the simple steps below:

- Make sure ***you're logged in*** as an ADMIN on our system.
- Click on the ADMIN tab from the menu on the top-left corner (see fig 14.1).

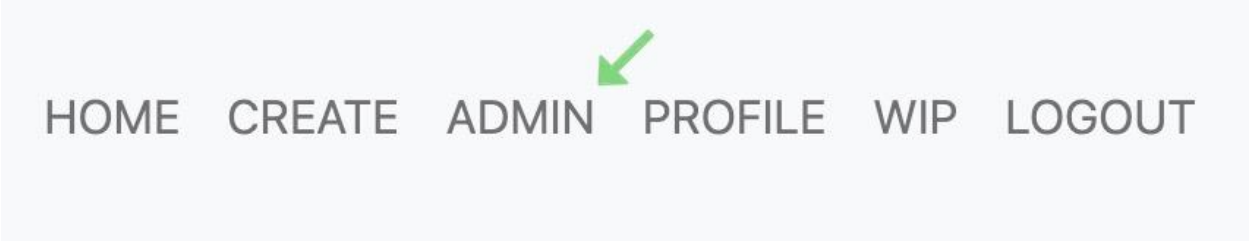


Fig 14.1

- You'll be redirected to a page that'll contain a list of pending requests, which you can either approve or deny (see fig 14.2)

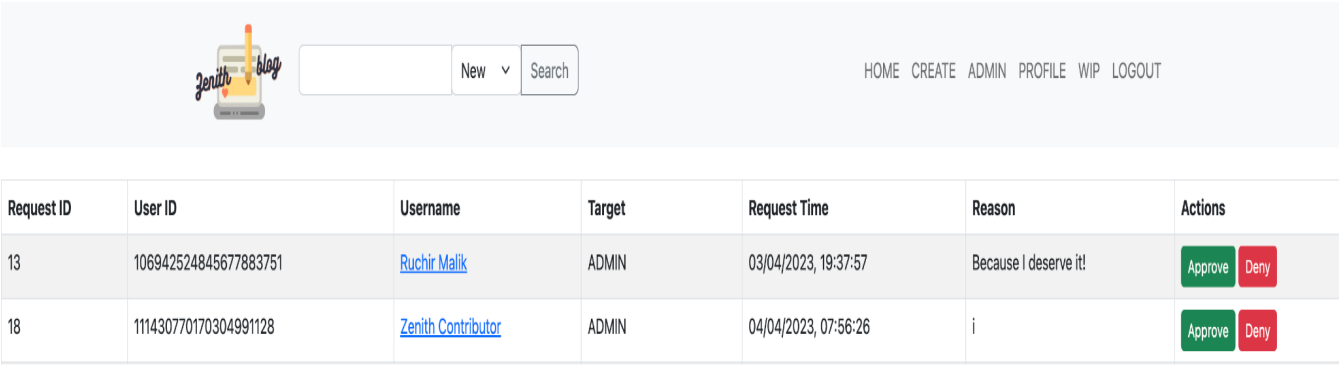


Fig 14.2

15.0 TRIGGER ON DEV ENVIRONMENT / FEATURE BRANCH

Upon any push or merge to the ‘dev’ branch or feature branches in the project repo, a Github Action is triggered. The Github Action sets up the dependencies and runs automated tests on both the frontend and the backend, both of which are integrated into the building of a docker image. This allows the tests to run in a third party, standardized environment. If either of the docker images fail to build, the build as a whole fails (see fig 15.2).

In the case that the images are built successfully in the ‘dev’ branch, meaning that all tests pass, the changes made will be deployed. This will happen in the form of a merge from the ‘dev’ branch into the ‘qa’ branch (see fig 15.1).

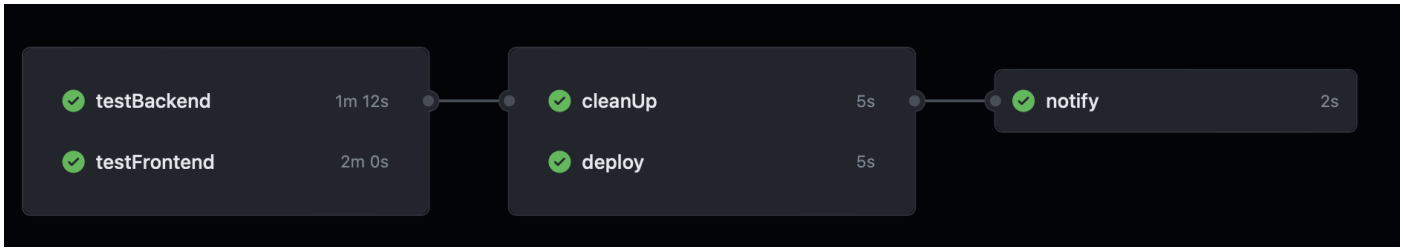


Fig 15.1
Example: success on push to dev branch

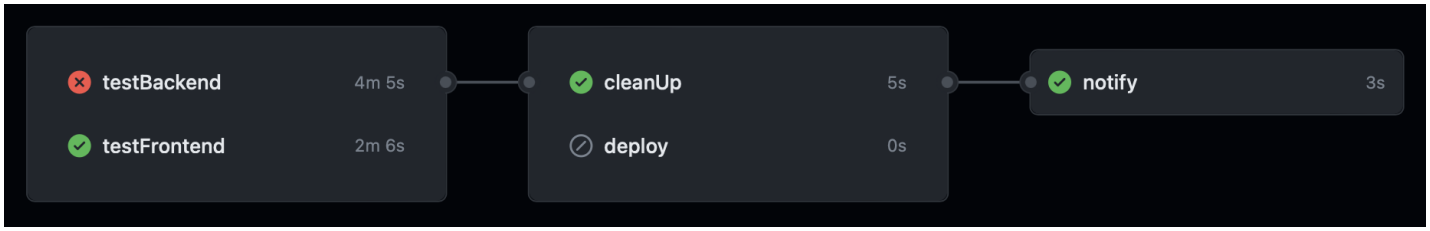


Fig 15.2
Example: fail on push to dev branch

16.0 TRIGGER ON QA AND PROD ENVIRONMENTS

Upon any push or merge to the ‘qa’ or ‘prod’ branch in the project repo, a Google Cloud Build process will trigger. The cloud build process involves building images for the frontend and backend, in which dependencies will be set up and automated tests will run. If both images are built successfully, the images will be pushed to a Google Artifact registry and then deployed with the Google Cloud Run service (see fig 16.1).

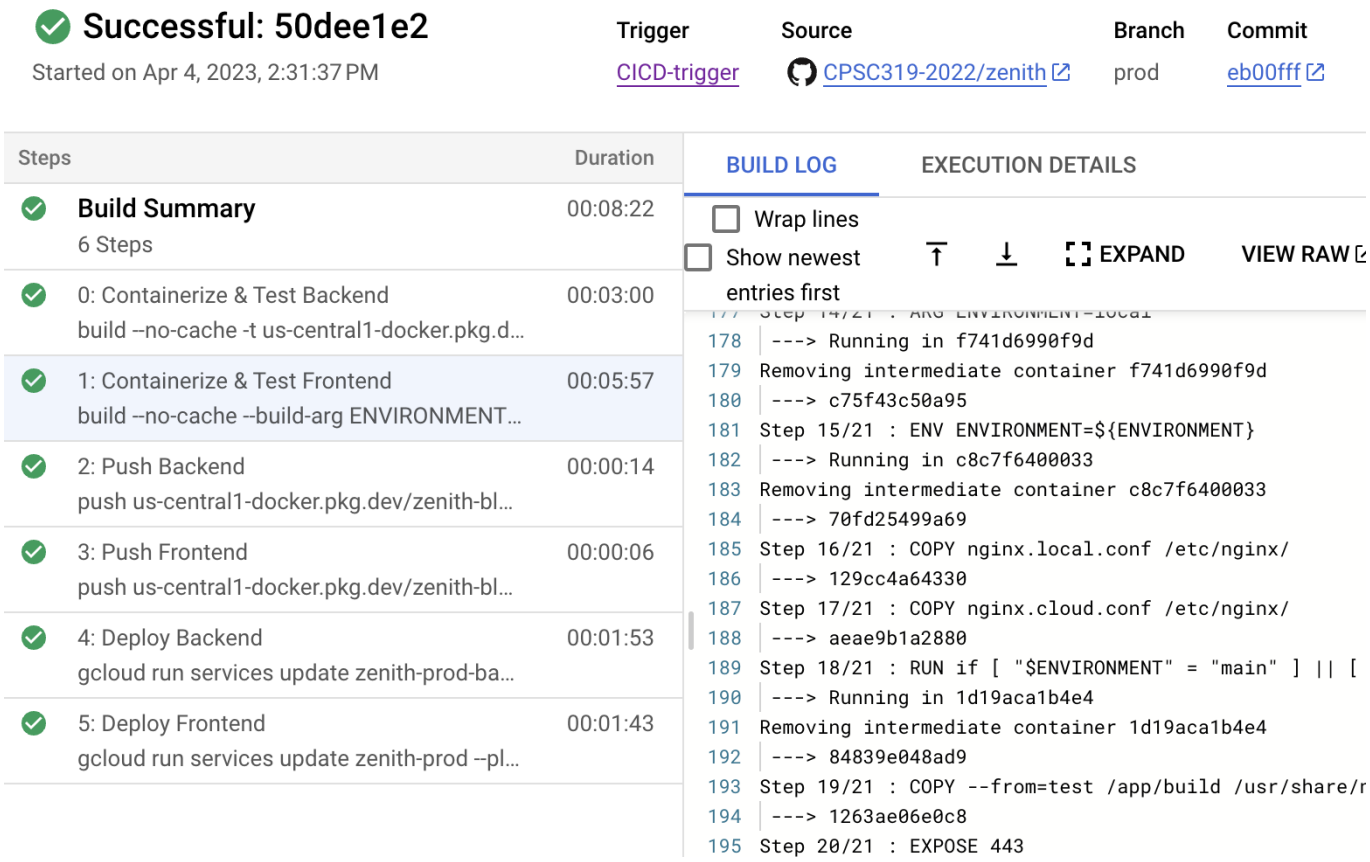


Fig 16.1
Example: success on push to prod branch

Both the ‘qa’ and ‘prod’ environments deploy to separate cloud run services. This allows the QA team to test a live version of the application without affecting the production version that is made openly accessible to the end users.

If the automated testing fails for either the frontend or the backend, neither of the images from that commit will be deployed. This ensures there will always be a stable version of the application deployed (see fig 16.2).

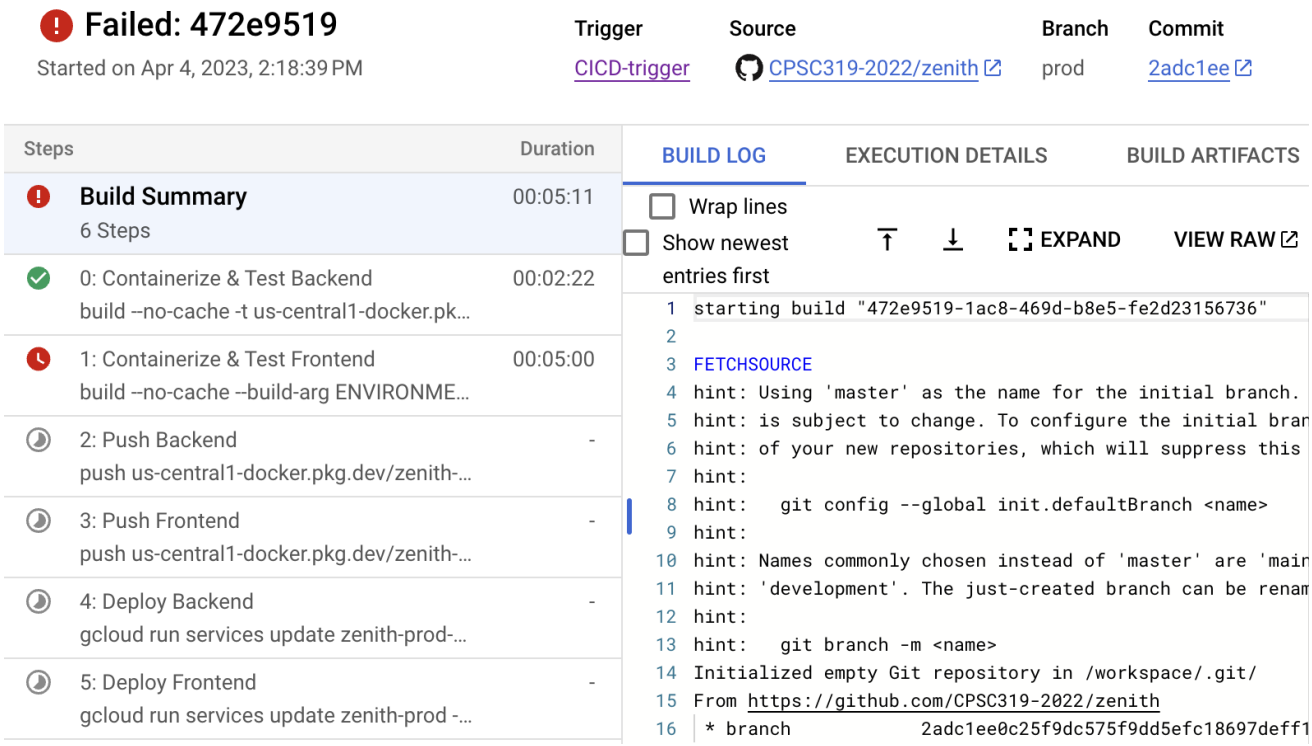


Fig 16.2
Example: fail on push to prod branch

In the case that an unstable version makes it past testing and gets deployed on a GCP server, it is easy to revert to a stable build through cloud run, which saves previously deployed containers. This can be accessed through the Manage Traffic button in the Cloud Run service of the gcloud console.

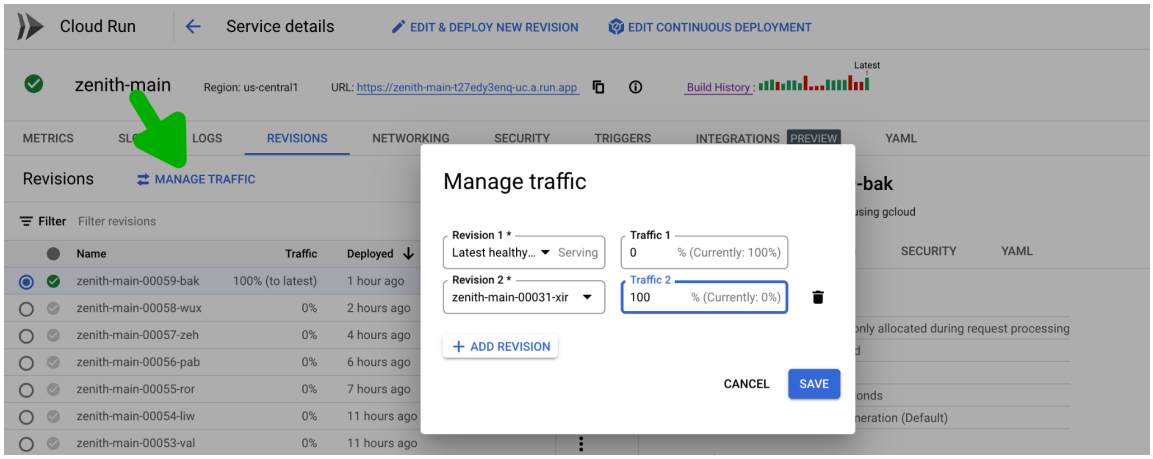


Fig 16.3
Example: Traffic Management of Cloud Run Service

17.0 SLACK NOTIFICATIONS

After a Google Cloud Build process is concluded, following a push to the ‘prod’ or ‘qa’ branch, a slack notification will automatically be sent to the “gc-build-notification” channel of the project’s Slack server, providing details about the commit and its build status (see fig 17.1).

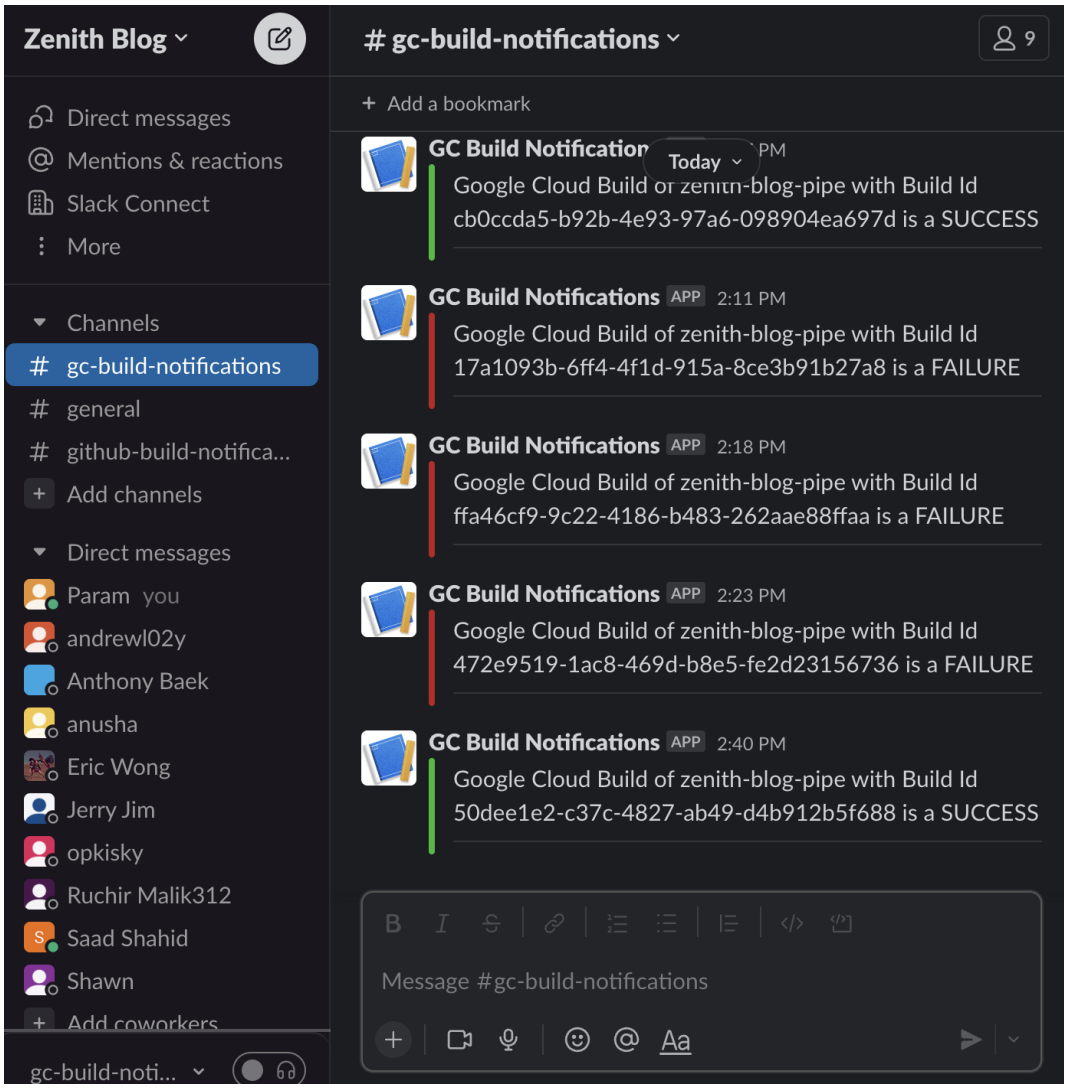


Fig 17.1

Similarly, after a Github Action has run the automated tests following a push to the dev branch or any feature branch, a slack notification will automatically be sent to the “github-build-notification” channel within the project’s Slack server. The notification will include the branch name the push was made to, the build status, and a link to the changes made on that commit (see fig 17.2).

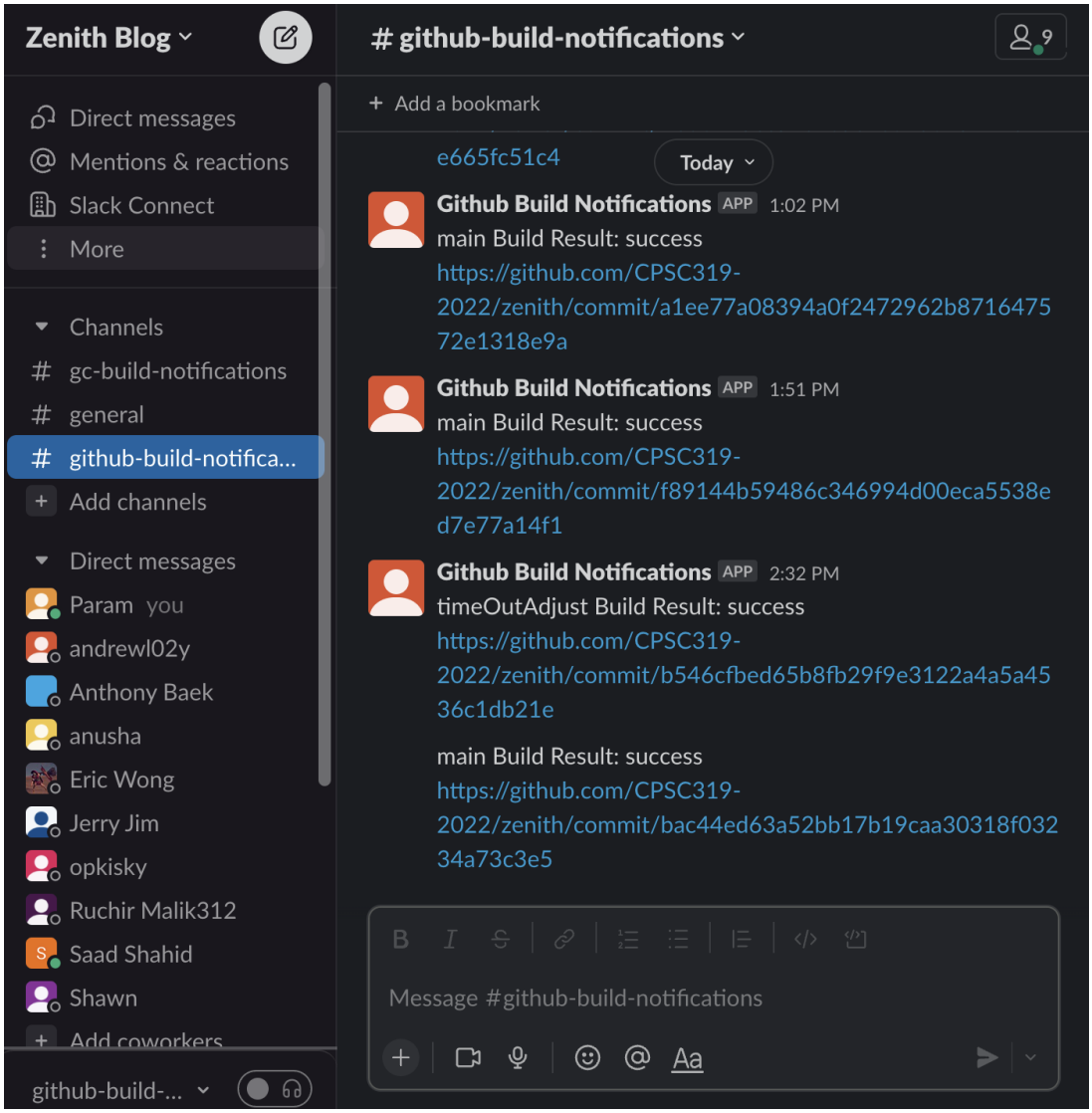

Zenith Blog
github-build-notifications
9
Direct messages
Mentions & reactions
Slack Connect
More
+ Add a bookmark
Channels
gc-build-notifications
general
github-build-notifica...
Add channels
Direct messages
Param you
andrewl02y
Anthony Baek
anusha
Eric Wong
Jerry Jim
opkisky
Ruchir Malik312
Saad Shahid
Shawn
Add coworkers
github-build-...
e665fc51c4
Today
Github Build Notifications APP 1:02 PM
main Build Result: success
https://github.com/CPSC319-2022/zenith/commit/a1ee77a08394a0f2472962b871647572e1318e9a
Github Build Notifications APP 1:51 PM
main Build Result: success
https://github.com/CPSC319-2022/zenith/commit/f89144b59486c346994d00eca5538ed7e77a14f1
Github Build Notifications APP 2:32 PM
timeOutAdjust Build Result: success
https://github.com/CPSC319-2022/zenith/commit/b546cfbed65b8fb29f9e3122a4a5a4536c1db21e
main Build Result: success
https://github.com/CPSC319-2022/zenith/commit/bac44ed63a52bb17b19caa30318f03234a73c3e5
Message #github-build-notifications
Aa

Fig 17.2