

Group 2

Group Member Information

| <u>Name</u> | <u>ID</u> |
|--------------------|-----------|
| Md. Fahim Bin Amin | 011201158 |
| Israt Jahan Khan | 011201142 |
| Abtahi Arifeen | 011212044 |
| Sadia Afrin Mou | 011211102 |

Project Title: StudentBook

Project Features:

1. Connectivity for Unique Educational Social Media:

- Create a secure and exclusive social media platform tailored to the specific educational institution.
- Implement user authentication mechanisms to ensure only valid students, faculty, alumni, and staff can access the platform.
- Foster a sense of community by providing a dedicated space for students, faculty, alumni, and staff to interact and collaborate.

2. Sharing Posts and Resources:

- Enable users to create, share, and engage with posts, articles, announcements, and educational resources.
- Facilitate discussions, comments, and likes on posts to encourage meaningful interactions.

3. Online Meetings and Seminars:

- Integrate video conferencing tools to host virtual classrooms, seminars, workshops, and meetings.

4. Official Job Portal:

- Develop a job portal where students can explore internship, part-time job, and full-time job opportunities.
- Allow employers to post job listings and connect with potential candidates directly through the platform.

5. Accessibility and User Features:

- Design the platform with a responsive and accessible user interface to cater to diverse user needs.
- Implement customizable user profiles to enhance user experience.

6. Learning Resources Repository:

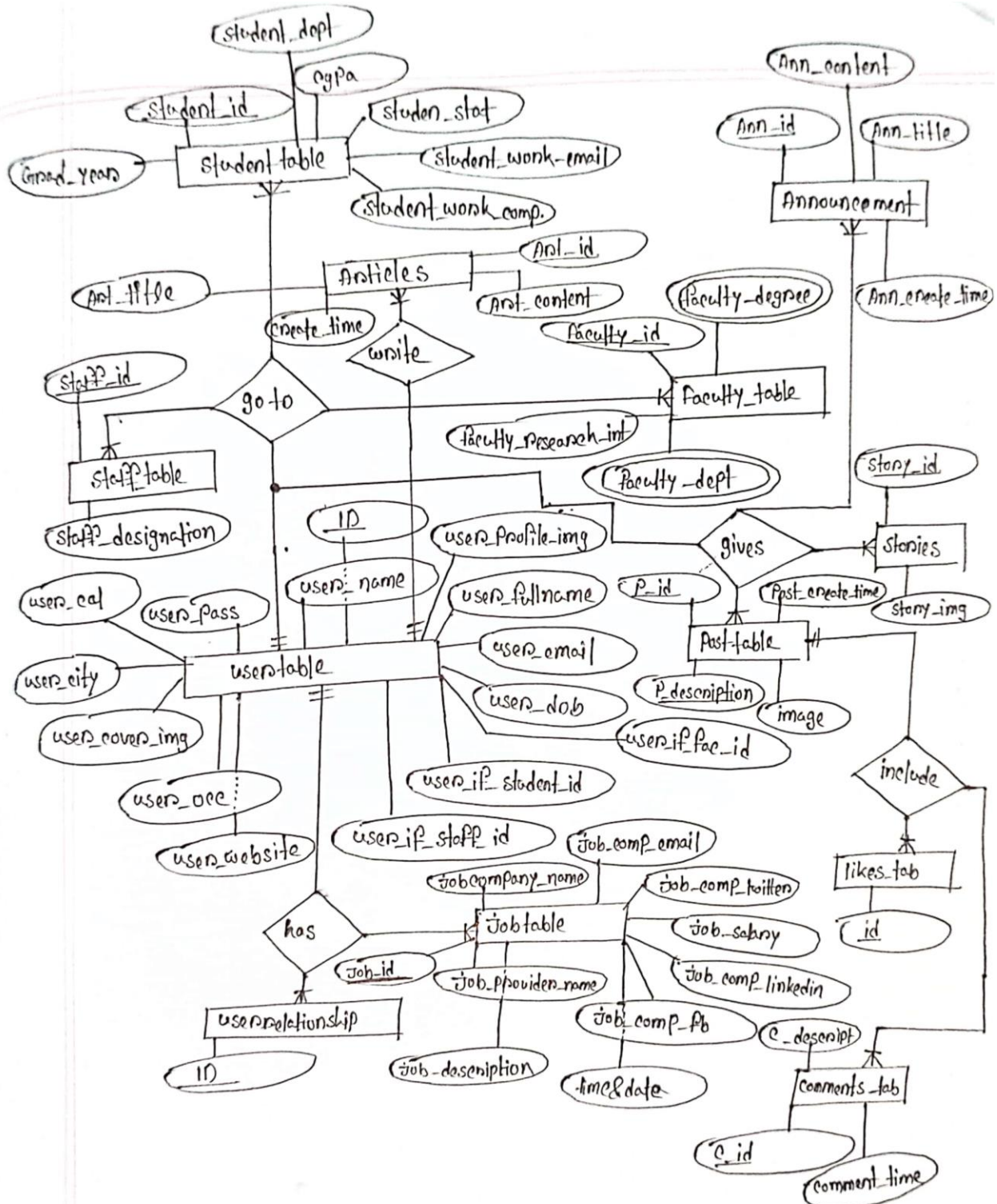
- Create a centralized repository for academic resources, including lecture notes, study materials, and research papers.
- Implement search and filtering options to help users quickly find other users along with their CGPA (if they are students).

7. Security and Privacy Measures: -

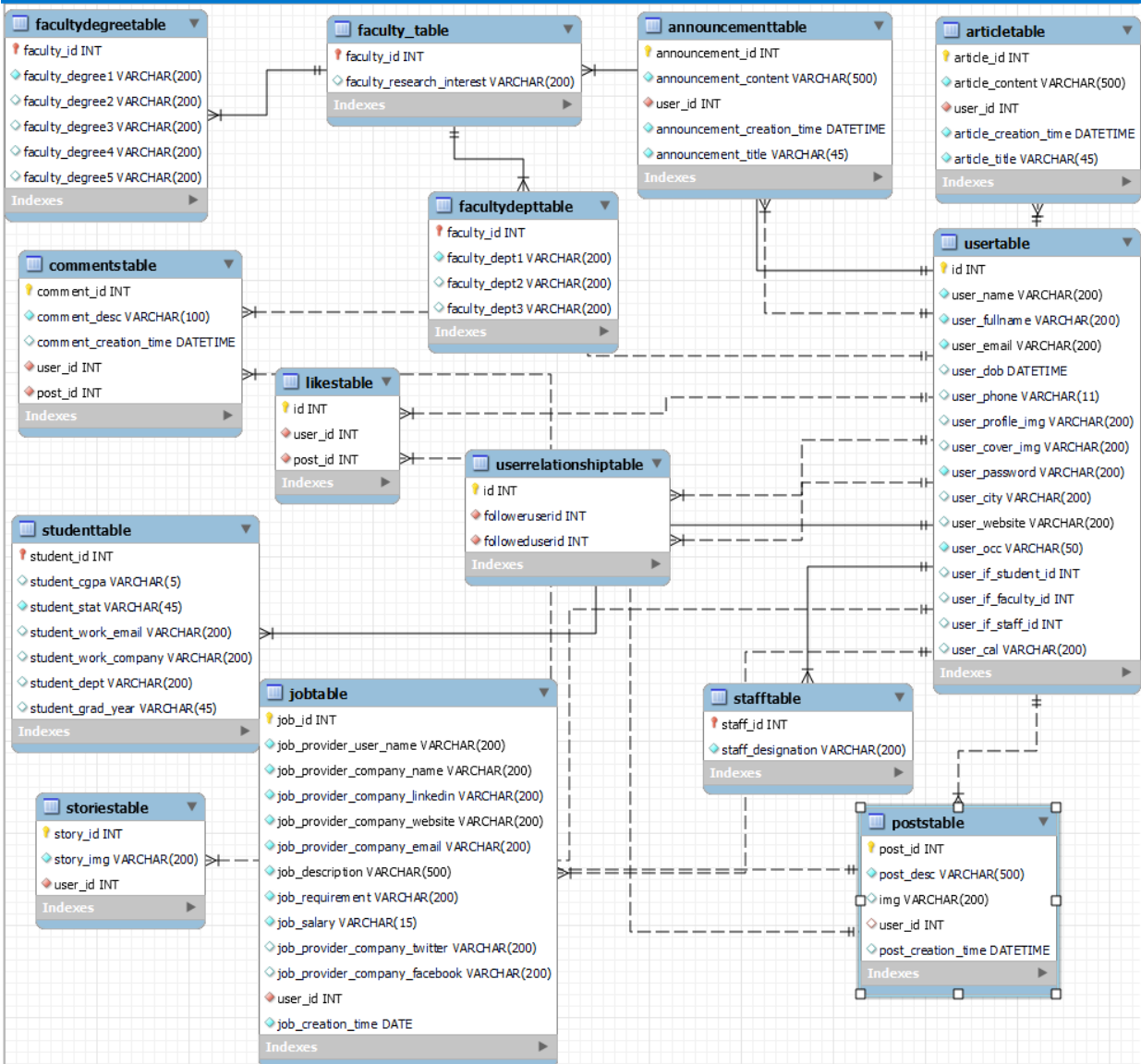
- Implement robust security measures to protect user data, including encryption and secure authentication.
- Using hashed password protection for enhanced security.

By incorporating these features, the StudentBook project aims to create an all-inclusive educational social platform that enriches the academic experience and fosters a sense of community within the educational institution.

ERD



Schema



MySQL Queries

1. getUser function for selecting all the users.
`const q = "SELECT * FROM usertable WHERE id=?"`
2. updateUser function for updating the user profile data.
`const q = "UPDATE usertable SET `user_fullname`=?,
`user_city`=?, `user_website`=?, `user_profile_img`=?,
`user_cover_img`=? WHERE id=?"`
3. searchByUsername function for searching user's full name along with the CGPA (if applicable).
`const q = `SELECT ut.*, st.student_cgpa
FROM usertable ut
LEFT JOIN studenttable st ON ut.user_if_student_id =
st.student_id
WHERE ut.user_fullname LIKE ?
,``
4. getRelationships function for identifying the follower and followed relationship among users.
`const q = "SELECT followeruserid FROM userrelationshipstable
WHERE followeduserid = ?";`
5. addRelationship function for adding relationship (follower/following) among users.
`const q = "INSERT INTO userrelationshipstable
(`followeruserid`,`followeduserid`) VALUES (?);"`
6. deleteRelationship function for deleting the user relationship.
`const q = "DELETE FROM userrelationshipstable WHERE
`followeruserid` = ? AND `followeduserid` = ?";`
7. getPosts function for fetching the posts.
`const q = userId !== "undefined"
? `SELECT p.*, u.id AS userId, user_name, user_profile_img
FROM poststable AS p
JOIN usertable AS u ON u.id = p.user_id
WHERE p.user_id = ?`

```
ORDER BY p.post_creation_time DESC`
: `SELECT p.*, u.id AS userId, user_name, user_profile_img
FROM poststable AS p
JOIN usertable AS u ON u.id = p.user_id
LEFT JOIN userrelationshipstable AS r ON p.user_id =
r.followeduserid AND r.followeruserid = ?
WHERE r.followeruserid IS NULL OR p.user_id = ?
ORDER BY p.post_creation_time DESC`;
```

8. addPost function for creating a new post from user's homepage and adding that to our database.

```
const q = "INSERT INTO poststable (`post_desc`, `img`,
`post_creation_time`, `user_id`, `user_fullname`) VALUES (?, ?, ?, ?,
?)";
```

9. deletePost function for deleting a post. The user only gets the option to delete his/her posts, not others' posts. The delete button only appears when the user is the same user who made the post. We implemented that in ReactJS instead of using the MySQL query.

```
const q = "DELETE FROM poststable WHERE `post_id`=?
AND `user_id`=?";
```

10. getLikes function for keeping updated data of how many likes a post has and who liked them.

```
const q = "SELECT user_id FROM liketable WHERE post_id =
?";
```

11. addLike function for adding a new like to a post if the current user has not liked the post yet.

```
const q = "INSERT INTO liketable (`user_id`, `post_id`)
VALUES (?)";
```

12. deleteLike function for deleting (removing) like.

```
const q = "DELETE FROM liketable WHERE `user_id` = ?
AND `post_id` = ?";
```

13. createJob function for creating a new Job post.

```
const q = "INSERT INTO jobtable (`job_provider_user_name`,
`job_provider_company_name`, `job_provider_company_linkedin`,
`job_provider_company_website`, `job_provider_company_email`,
`job_description`, `job_requirement`, `job_salary`,
`job_provider_company_twitter`, `job_provider_company_facebook`,
`user_id`, `job_creation_time`) VALUES (?)";
```

14. getAllJobs function to fetch all the job posts available on the database.

```
const q = `
    SELECT j.job_id, j.job_provider_user_name,
    j.job_provider_company_name, j.job_provider_company_linkedin,
    j.job_provider_company_website, j.job_provider_company_email,
    j.job_description, j.job_requirement, j.job_salary,
    j.job_provider_company_twitter, j.job_provider_company_facebook,
    j.user_id, j.job_creation_time,
    u.user_fullname, u.user_profile_img
    FROM jobtable j
    INNER JOIN usertable u ON j.user_id = u.id
`;
```

15. deleteOneJob function for deleting one job at a time. We checked the condition whether the user is the specific user who made the post in the api. One user can delete the job only he is the one who created the job post.

```
const q = "DELETE FROM jobtable WHERE job_id = ?";
```

16. getComments function for fetching all the comments a post has.

```
const q = `SELECT c.*, u.id AS userId, user_name,
user_profile_img FROM commentstable AS c JOIN usertable AS u
ON (u.id = c.user_id)
WHERE c.post_id = ? ORDER BY c.comment_creation_time
DESC
`;
```

17. addComment function for adding a new comment in a post.

```
const q = "INSERT INTO commentstable (`comment_desc`,
`comment_creation_time`, `user_id`, `post_id`) VALUES (?)";
```

18. deleteComment function for deleting a comment. We checked the condition where the user is the specific user who made the comment in the api. One user can only delete his comment if and only if he is the one who made the comment.

```
const deleteQuery = "DELETE FROM commentstable
WHERE id = ?";
```

19. Checking if a user already exists in auth.js during registration for a new user.

```
const q = "SELECT * FROM usertable WHERE user_name = ?"
```

20. Registering user.

```
const q = "INSERT INTO usertable (`user_name`,  
`user_fullname`, `user_email`, `user_occ`, `user_password`) VALUE  
(?)"
```

21. Login.

```
const q = "SELECT * FROM usertable WHERE user_name = ?"
```

22. Logout. The logout process has been implemented using the accessToken, not using any Query of MySQL.

23. createArticle function for creating a new article.

```
const q = "INSERT INTO articletable (`article_content`,  
`article_creation_time`, `user_id`, `article_title`) VALUES (?)";
```

24. getAllArticles function for fetching all the articles.

```
const q = `  
SELECT a.article_id, a.article_content, a.user_id,  
a.article_title, a.article_creation_time,  
u.user_fullname, u.user_profile_img  
FROM articletable a  
INNER JOIN usertable u ON a.user_id = u.id  
`;
```

25. deleteOneArticle function for deleting one article at a time. We checked the condition where the user is the specific user who posted the article in the api. One user can only delete his article if and only if he is the one who posted the article.

```
const q = "DELETE FROM articletable WHERE article_id = ?";
```

26. createAnnouncement function for creating a new announcement.

```
const q = "INSERT INTO announcementtable  
(`announcement_content`, `announcement_creation_time`,  
`announcement_title`, `user_id`) VALUES (?)";
```

27. getAllAnnouncements function for fetching all the announcements available on the database.

```
const q = `SELECT a.announcement_id,  
a.announcement_content,  
a.announcement_creation_time, a.announcement_title,  
a.user_id, u.user_fullname, u.user_profile_img FROM  
announcementtable a`
```



```
INNER JOIN usertable u ON a.user_id = u.id ORDER BY  
a.announcement_creation_time DESC`;
```

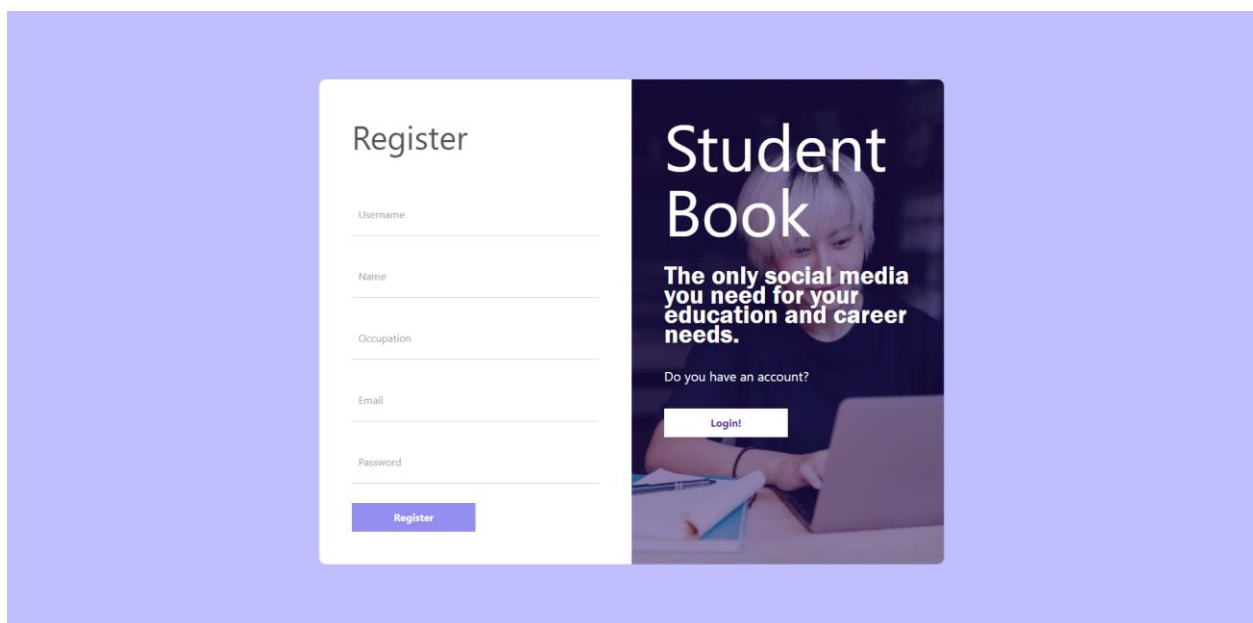
28. deleteOneAnnouncement function for deleting one announcement at a time. We checked the condition where the user is the specific user who made the announcement in the api. One user can only delete his announcement if and only if he is the one who made the announcement.

```
const q = "DELETE FROM announcementtable WHERE  
announcement_id = ?";
```

User Manual

Register:

If you are a new user, then register yourself in the platform by visiting the Register page.

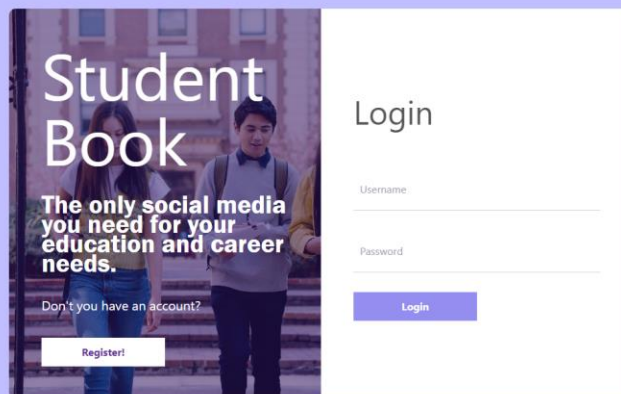


The username must be unique. Therefore, if you give any username that is already in use by someone, then it would notify you instantly and you can not create the account until you give any unique username.

Your password remains safe as the platform uses an encrypted method to create a hashed string using your password and only the hashed password gets stored in the database. Therefore, if our database gets breached, your password and data would remain safe from hackers.

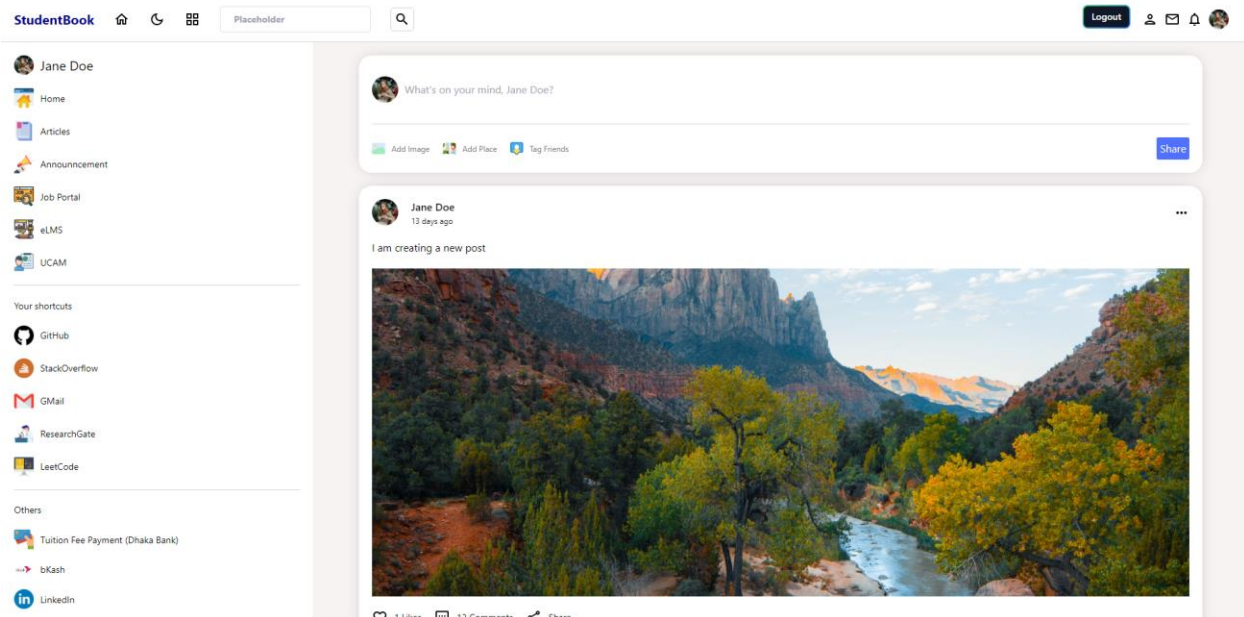
Login:

For Login, you need to provide your valid username and password.



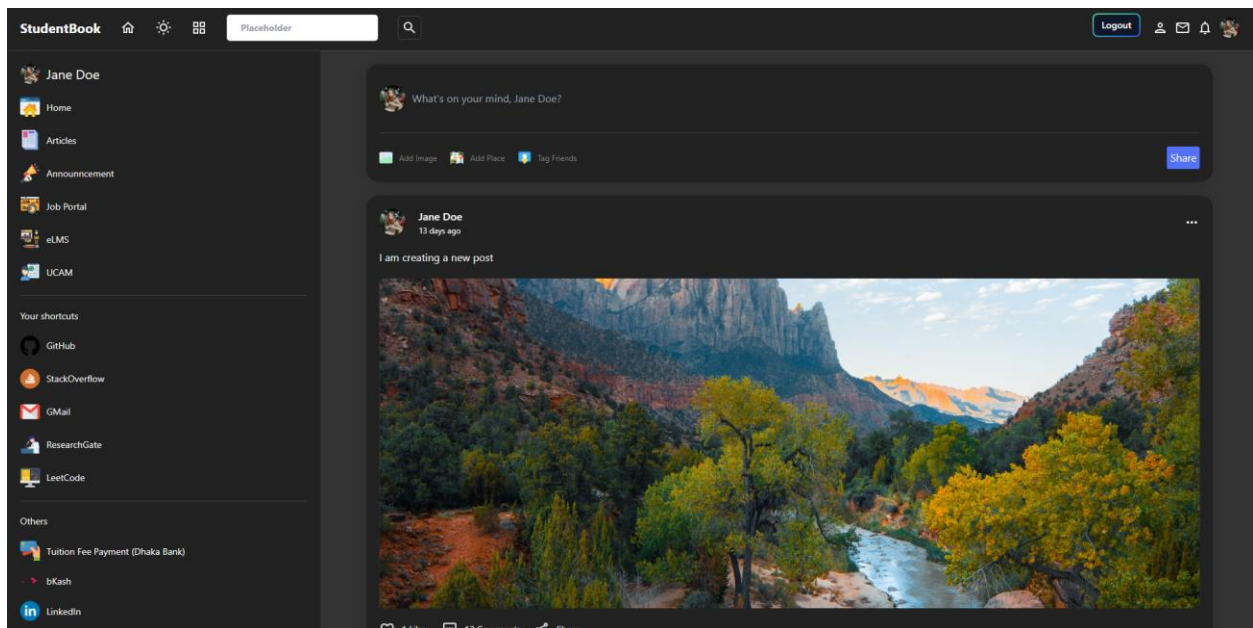
Homepage:

Homepage is the main page after you login to your account.



It contains a newsfeed, a navbar, a sidebar on the left which contains necessary pages and navigation throughout the entire website.

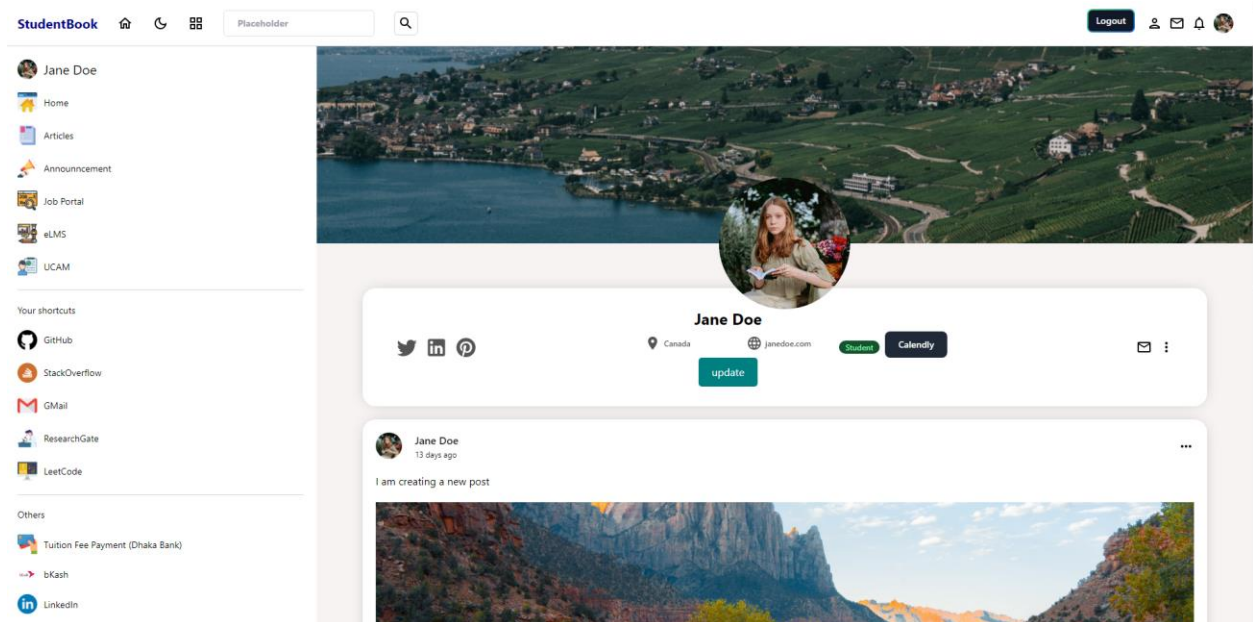
The platform has global light/dark mode features. You can switch them whenever you feel the need.



The initial color mode theme gets displayed based on your device's color mode. For example, if you are using light mode on your computer, then if you login using that computer, you will get light theme during login. Obviously, you can change them as much as you want!

Profile Page:

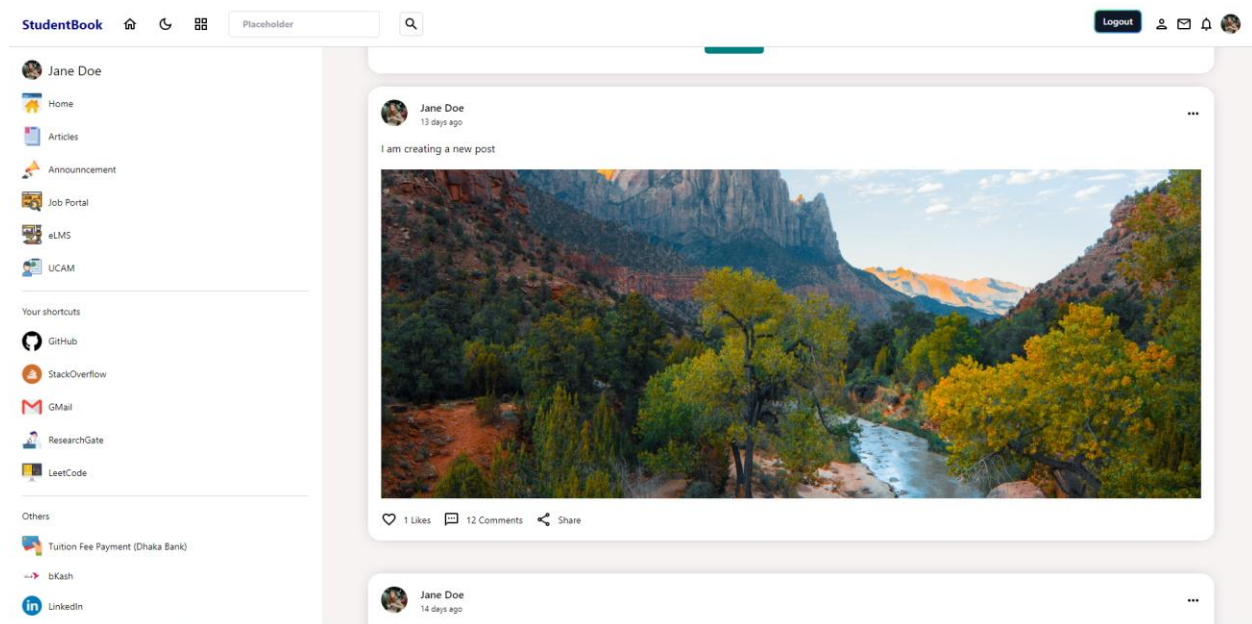
Your personal profile section would look like this.



You can add or edit the information, but you cannot edit all the information. For example, you can not edit your username, student id.

You can also visit whether a user is actually a student, or alumni or a staff member. You can see all the posts made by a user in their specific profile page.

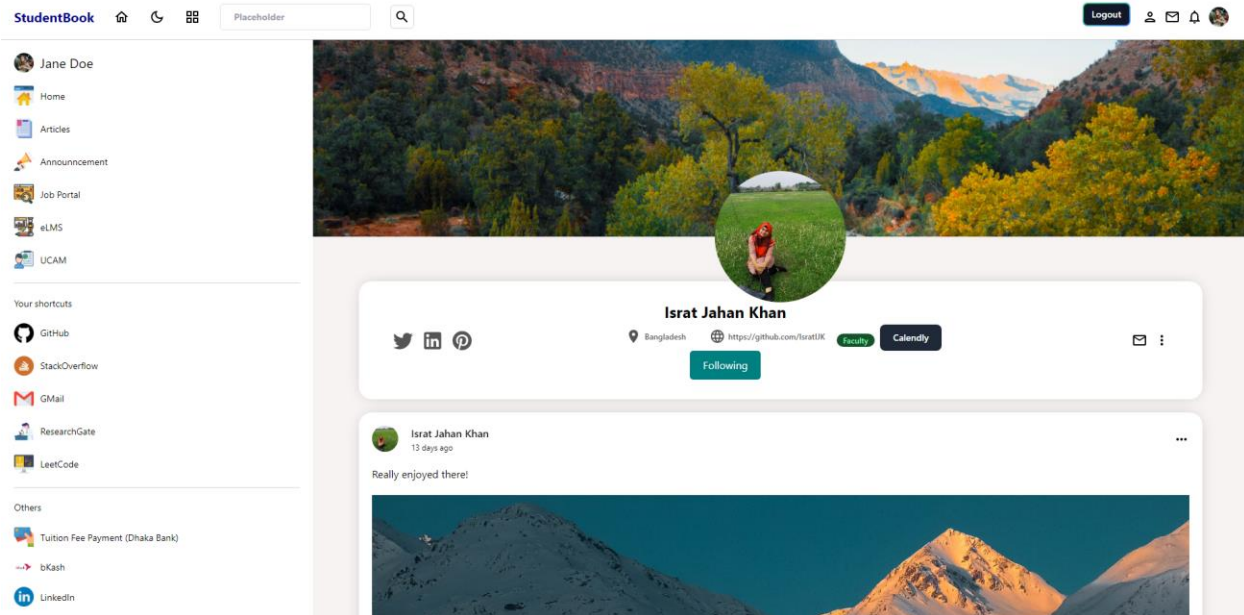
You can like their posts and comment in them; also share them from their profile section.



Also, the same activities can be done from your newsfeed section as well.

Follow/Unfollow:

When you create an account in StudentBook, you are following each user by default. Therefore, you get all the posts of them and so on. But let say, for some reason, you do not want to receive anyone's posts, you can simply unfollow them from their profile section.

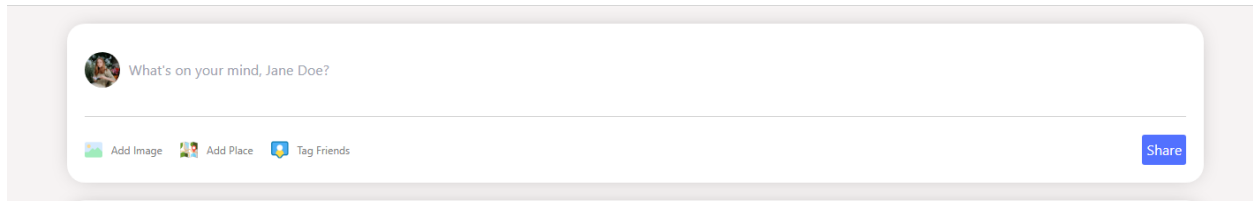


For that, you need to toggle the “Following” button from that specific user’s profile page.

If you unfollow someone, you will not get any post from that person ever again until you follow them again.

Post:

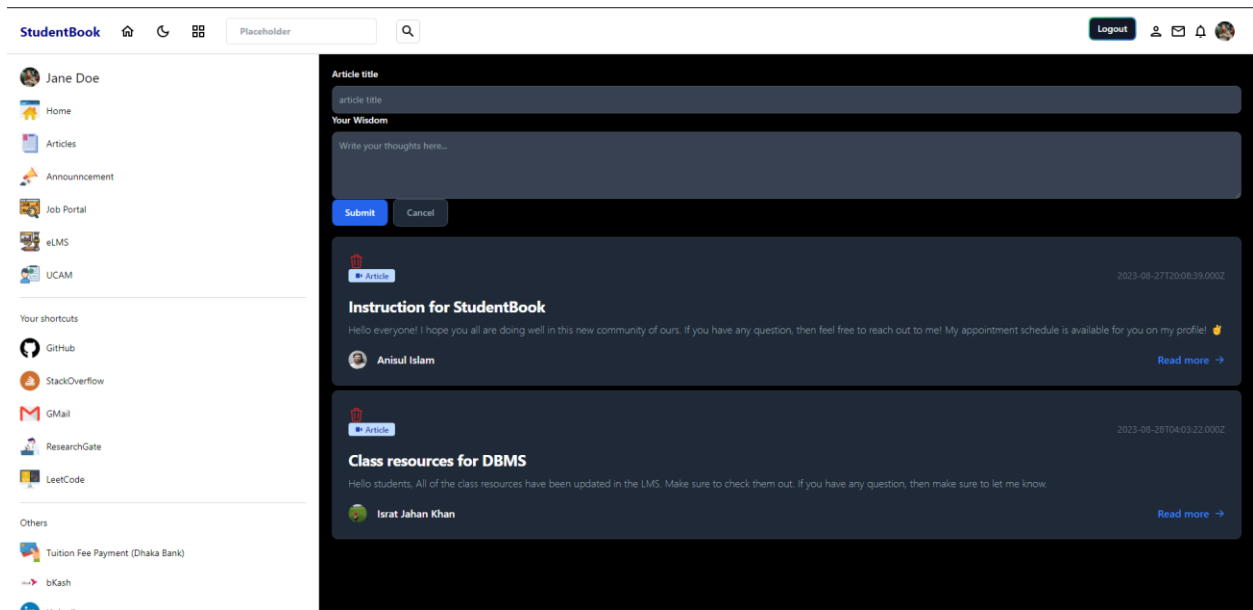
You can post anything from your newsfeed. You can add any image file you want to your post as well.



You can also delete your post whenever you want.

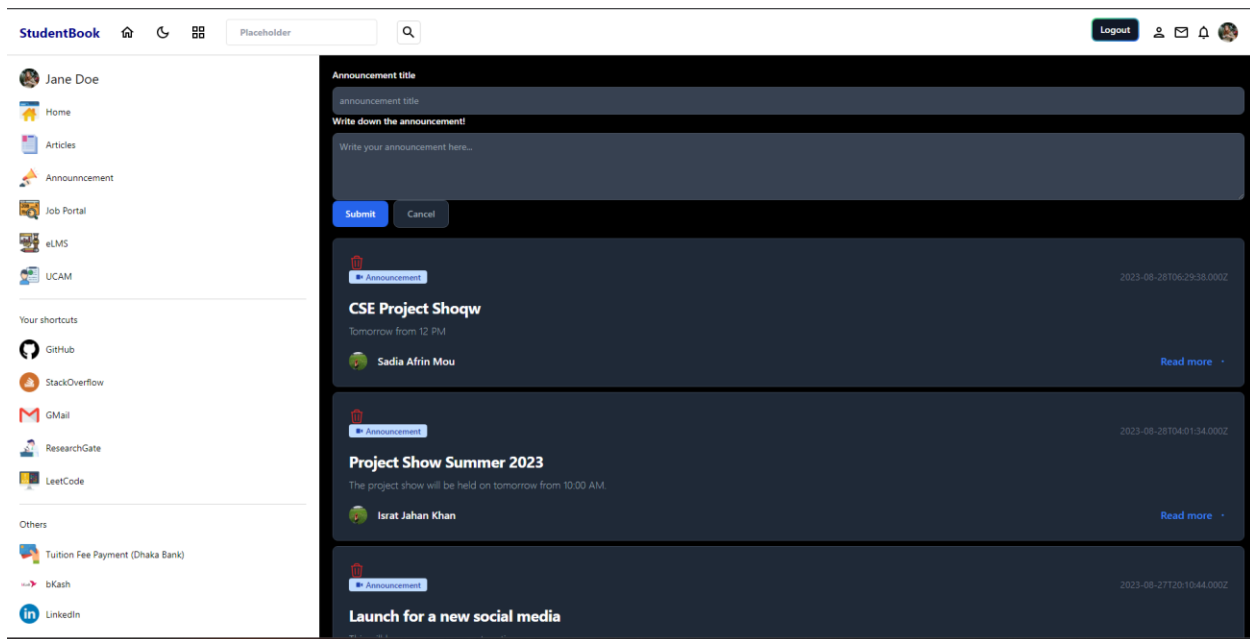
All posts use real timeline when you share the post.

Article:



The platform has a dedicated page for posting articles and getting to see others' articles. You can also delete your articles after posting.

Announcement:



The platform also has a dedicated section for the announcement. You do not need to find the latest announcements from other places anymore. You can simply visit this announcement page and get all the announcements.

If you want to publish any announcements, then you are also welcome to do so. Later, you can also delete any announcements you have made.

Job Portal:

StudentBook

Placeholder

Logout

Jane Doe

Home

Articles

Announcement

Job Portal

eLMS

UCAM

Your shortcuts

GitHub

StackOverflow

GMail

ResearchGate

LeetCode

Others

Tuition Fee Payment (Dhaka Bank)

bKash

LinkedIn

Recruiter's Full Name

Write your full name...

Company Name

Brain Station 23

Website

brainstation23.com

Official LinkedIn Page

https://www.linkedin.com/company/brain-station-23/

Official Twitter/X

https://twitter.com/BrainStation23

Official Facebook Page

https://www.facebook.com/brainstation23

Email address

forbrainstation23@gmail.com

Job Description

Write down the complete job description

Job Requirement

Write down the complete job requirement

Salary/Salary Range

Enter your salary range...

Submit


Cancel


This platform also has a dedicated job portal. You can get all the job circulars that are being posted in this platform. Additionally, you can post your own job circulars here as well and delete them if that becomes obsolete in time.


Handful Resources:


You do not need to browse a regularly used handful of resources individually on your browser as almost all of them are listed in one platform, StudentBook.


If you look at your left side bar, then you will see a handful of useful resources you need!


 Jane Doe


 Home

 Articles


 Announcment


 Job Portal


 eLMS


 UCAM


Your shortcuts

 GitHub


 StackOverflow


 GMail


 ResearchGate


 LeetCode


Others


 Tuition Fee Payment (Dhaka Bank)

 bKash


 LinkedIn


 Job Portal


 eLMS


 UCAM


Your shortcuts

 GitHub


 StackOverflow


 GMail


 ResearchGate


 LeetCode


Others


 Tuition Fee Payment (Dhaka Bank)


 bKash

 LinkedIn

 freeCodeCamp

 Coursera

 GoodRead

 Quora

Video Demonstration:

We have a good video demonstration available for your concern. Make sure to check that out!

[Project StudentBook \(Project Demonstration\)](#)

Project Codebase:

The whole project has been made an open-source project. You can check the full codebase and commit history as much as you want. We have not deployed yet.

[StudentBook](#)