A PROJECT REPORT ON

Online Article & Discussion Platform for CSE Students

CSE Hankers

By

Jwalit Shah (CE121) (18CEUOS057)

B. Tech CE Semester - VI
Subject : Object Oriented Software Engineering

Guided By:

Prof. Mrudang T. Mehta	**Prof. Jatayu H. Baxi**
Associate Professor	**Assistant Professor**
Dept. Of Computer	**Dept. Of Computer**
Engineering	**Engineering**

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

<u>CERTIFICATE</u>

This is to certify that the practical / term work carried out in the subject of **Object Oriented Software Engineering** and recorded in this journal is the bonafide work of

Jwalit Shah (CE121) (18CEUOS057)

of B.Tech semester **VI** in the branch of **Computer Engineering** during the academic year **2020-2021**.

Prof. Mrudang T Mehta	Prof. Jatayu H. Baxi	Dr. C. K. Bhensdadia,
Associate Professor,	Assistant Professor,	Head,
Dept. of Computer Engg.,	Dept. of Computer Engg.,	Dept. of Computer Engg.,
Faculty of Technology,	Faculty of Technology,	Faculty of Technology,
Dharmsinh Desai University, Nadiad	Dharmsinh Desai University, Nadiad	Dharmsinh Desai University, Nadiad

Table of Contents

Abstract

This CSE Hankers System is a web application that provides the platform for users to publish and manage their articles. Users can also ask questions related to the field and also upvote the articles or answers by giving likes or comments to it.

Purpose

The purpose of the Software Requirements Specification is to outline the requirements for the "Online Article & Discussion Platform for CSE Students". This product will be built on Dotnet core and SQLite server using provided tech-stacks. It will be operating system independent and accessible with any standard compliant browser having python installed on that local machine.

Scope

"CSE Hankers" is intended for Computer Science related community which helps them to find the certain related topics via Blogs/Articles written by other person and also, they can have a Questionnaires & give answers on the platform which makes platform kind a Forum for discussion and helps them to find their way.

Introduction

- Brief Introduction

The "CSE Hankers" is a web application for users to get the way out in their field when they get stuck. The system mainly works around two users, End Users and one Admin user. Users can Register and login to the system. Users can View the articles of the users to whom the user follows. Users can also create, update and delete their Articles. Users can ask Questions to the forum. Upon finding the content useful users can give like or comment. And upon seeing pirated content they can report the content. System checks all the corner cases for authentication, authorization of the users, validation of input data, duplication of data and gives appropriate error response messages according to the situation.

- Technology/Platform/Tools used

Technology :
- Dot net core framework
- SQL Server database.
- Bootstrap

Platform :
- Windows

Tools :
- Visual Studio

Software Requirements Specification - SRS

Total 2 type of users are there in system:

1. User
2. Admin user

Functional Requirements

User wise defined Functional Requirements.

Common Functional Requirements

1. **Login**

   Description: Login to the Administrator dashboard

   Input: User credentials

   Output: Appropriate status message and Redirection

2. **Logout**

   Description: Logout from the currently logged in session

   Input: User selection

   Output: Success message and Redirect to login page

3. **Forgot Password**

   Description: Reset the password if the password is forgotten.

   Input: User Selection

   Output: Redirection to Login page after resetting password.

R.1 Administrator

R.1.1 Accept / Discard the Report Content request

Description: Administrator can Discard the Reported Content request or can delete the content by accepting the request.

Input: Article details

Output: Response message

R.1.2 View Contents of platform

Description: All of the contents (etc. Questions, Articles, Answers) can be viewed according to the user selection criteria

Input: User selection

Output: Details of Articles

R.1.3 Statistical data

Description: Some statistical data can be viewed of the platform

Input: User selection

Output: Statistical Details

R.2 End Users

R.2.1 Register

Description: Register the self to the platform.

Input: User details and credentials

Output: Appropriate status message and Redirection

R.2.2 Follow users

Description: Any user can follow the other user

Input: User selection

Output: Success message

R.2.3 Publish Articles

Description: User can publish the articles to all other users

Input: Article details

Output: Response message

R.2.4 View Articles

Description: Articles can be viewed according to the user selection criteria

Input: User selection

Output: Details of Articles

R.2.5 Update/Delete Articles

Description: User can Update and Delete the articles

Input: Article details & User selection

Output: Response message

R.2.6 Comment on the Articles

Description: User can give comments on the articles

Input: User selection & comment description

Output: Response message

R.2.7 Ask Questionnaires

Description: Users can put questions on the platform.

Input: Question description

Output: Success message

R.2.8 Modify and Delete asked Questions

Description: User can modify and delete the asked questions on the platform asked by the user itself.

Input: User selection and Questions description

Output: Success message

R.2.9 Like Articles/ Answers

Description: User can give like to their favorite articles/ answers to the question

Input: User selection

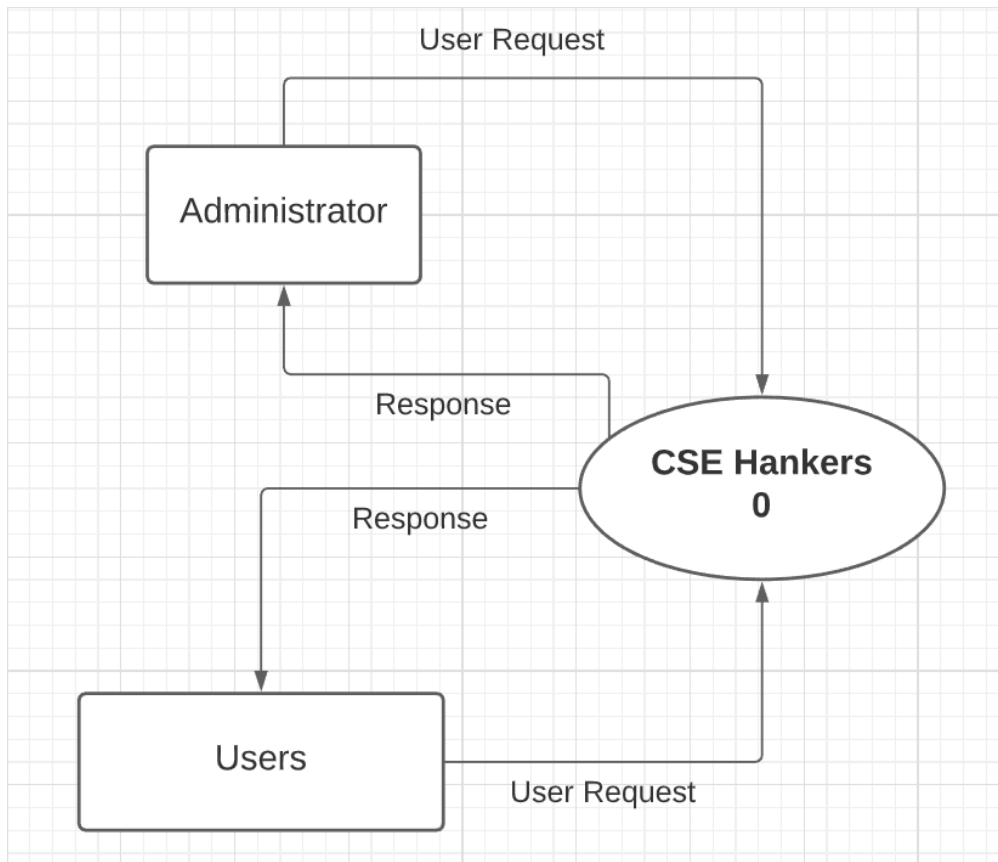Output: Success notification

R.2.10 Report for Contents

Description: Users can request the Administrator to perform certain actions on selected Content requests like (selected content can be Article, answer or question).
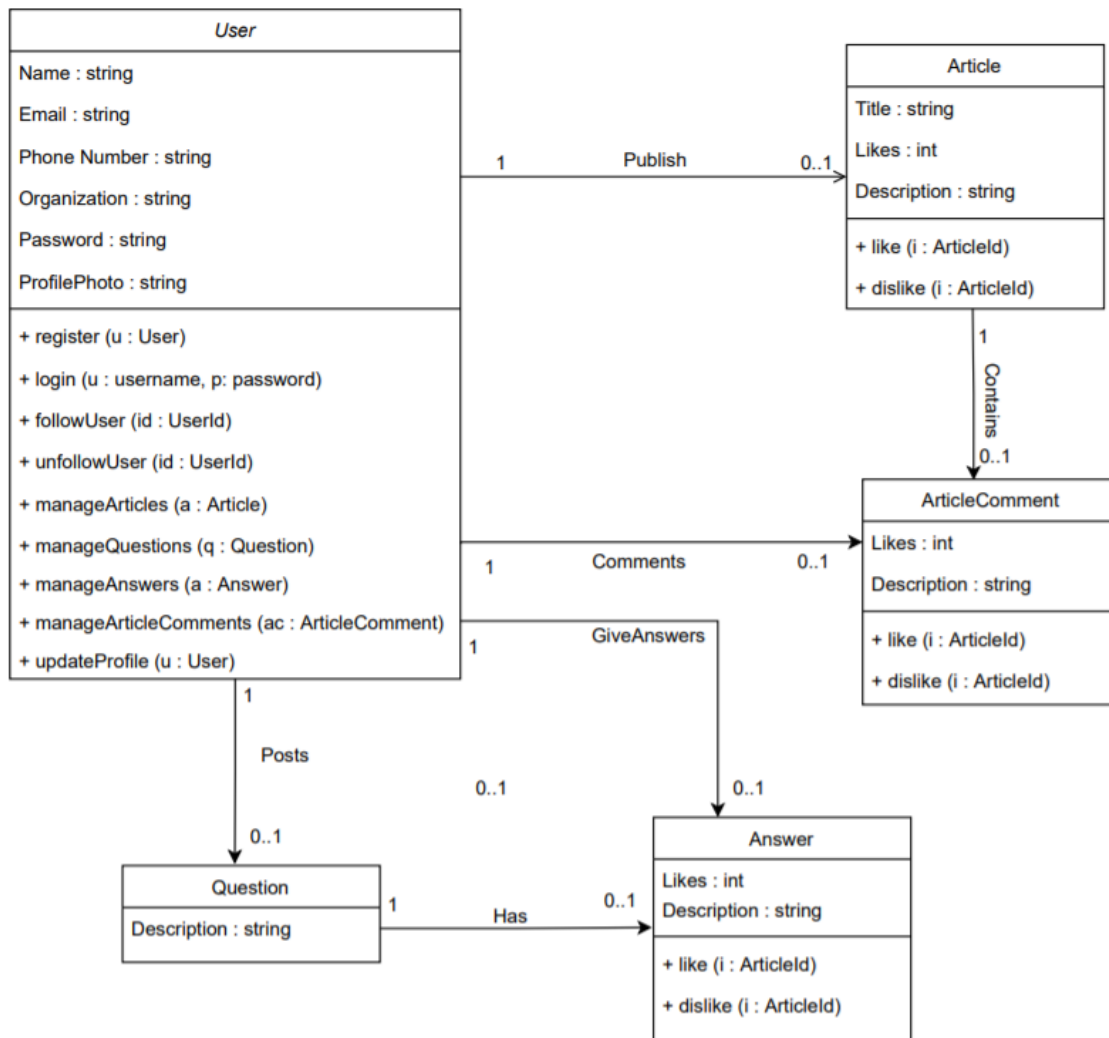
Input: Article details

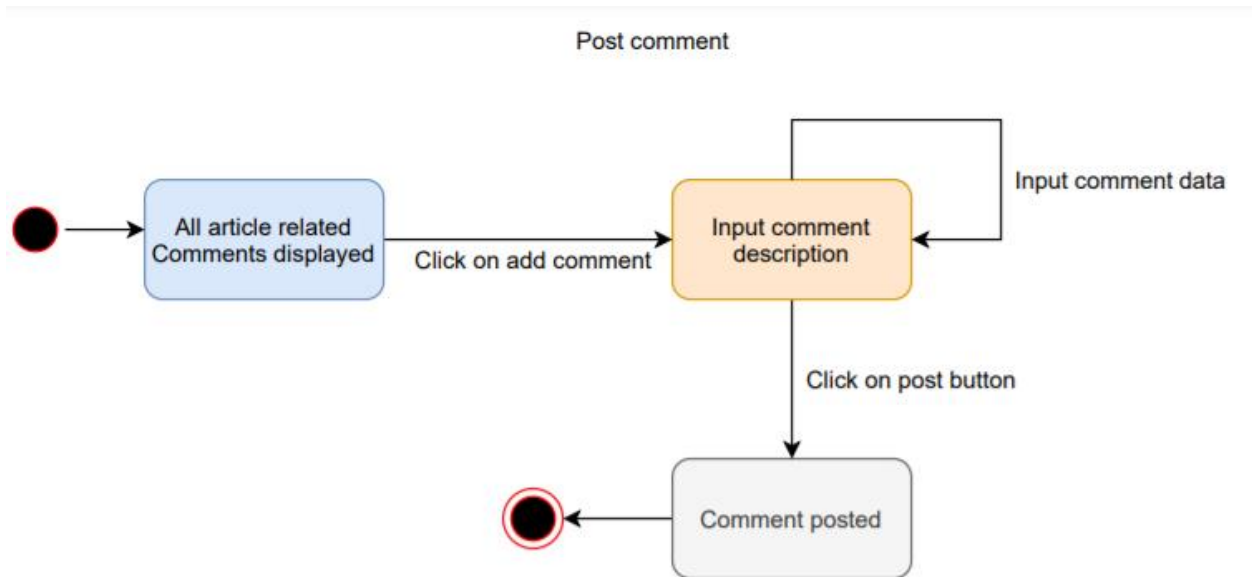Output: Response message

Design

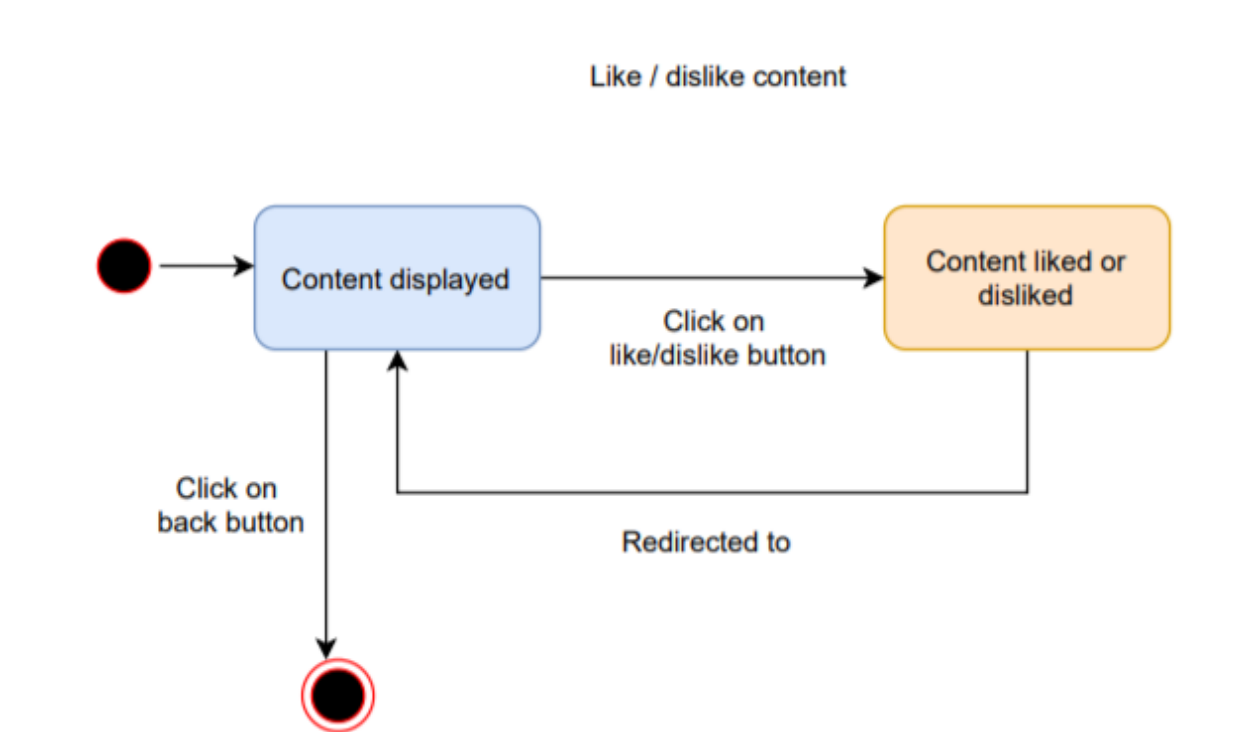## Data flow Diagram - Level 0

Class Diagram

Class Diagram

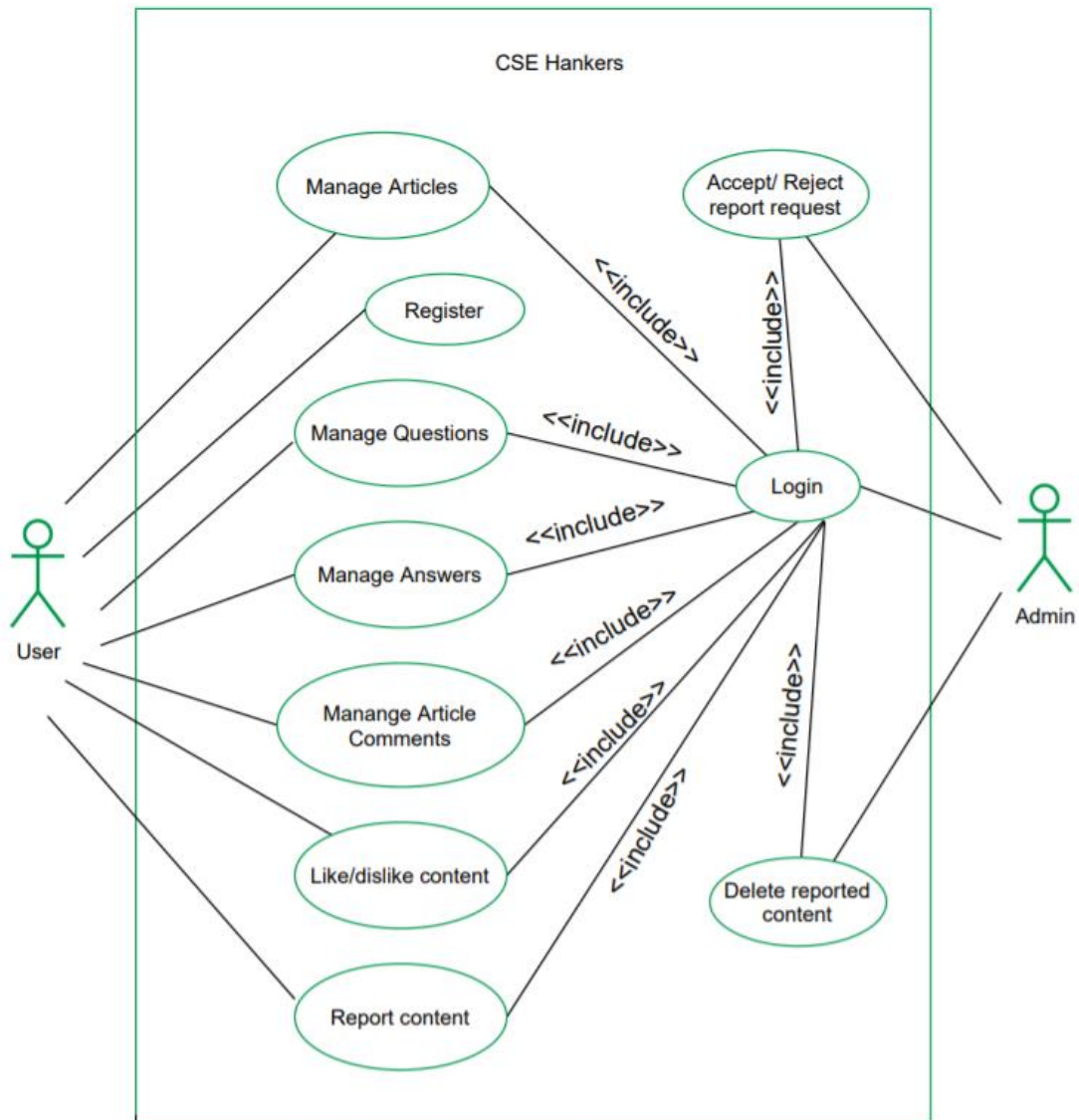**User** *(italic)*

Name : string
Email : string
Phone Number : string
Organization : string
Password : string
ProfilePhoto : string

+ register (u : User)
+ login (u : username, p: password)
+ followUser (id : UserId)
+ unfollowUser (id : UserId)
+ manageArticles (a : Article)
+ manageQuestions (q : Question)
+ manageAnswers (a : Answer)
+ manageArticleComments (ac : ArticleComment)
+ updateProfile (u : User)

Article

Title : string
Likes : int
Description : string

+ like (i : ArticleId)
+ dislike (i : ArticleId)

1   Publish   0..1

1   Contains

ArticleComment

Likes : int
Description : string

+ like (i : ArticleId)
+ dislike (i : ArticleId)

1   Comments   0..1

0..1

1   GiveAnswers

Answer

Likes : int
Description : string

+ like (i : ArticleId)
+ dislike (i : ArticleId)

0..1   0..1

1

Posts

0..1

Question

Description : string

1   Has   0..1

State Diagram

Post Comment

Post comment

All article related Comments displayed

Click on add comment

Input comment description

Input comment data

Click on post button

Comment posted

Like/ dislike content

Like / dislike content

Content displayed

Click on like/dislike button

Content liked or disliked

Click on back button

Redirected to

Use case Diagram

CSE Hankers

- Manage Articles
- Register
- Manage Questions
- Manage Answers
- Manange Article Comments
- Like/dislike content
- Report content
- Login
- Accept/ Reject report request
- Delete reported content

<<include>>

User

Admin

Sequence Diagram

Sequence diagram - forgot password

:user	:system	:database

dispatch

select forgot password

loads page

input data

validate data

error message on invalid input

reset password page sent

send email

enter token and password

validate data

error message on invalid data

new password

success message

success message

Activity Diagram

Activity Diagram - Like Content

User	CSE Hankers (System)

- Like content
- Check user Authentication
- is authenticated
- Redirect to Login page
- Check if resource exists
- is exists
- Display error
- is liked
- Check if content is already like by user
- like the content

ER Diagram

ArticleLikes

Id (PK)	int
AuthorId (FK)	int
Author	User
ArticleId (FK)	int

Article

Id (PK)	int
AuthorId (FK)	int
Author	User
Title	string
Description	string
Likes	int

Users

Id (PK)	int
Email	string
Name	string
Organization	string
MobileNumber	string
Password	string
ProfilePhoto	string

ArticleComments

Id (PK)	int
AuthorId (FK)	int
Author	User
Comment	string
Likes	int

Answers

Id (PK)	int
AuthorId (FK)	int
Author	User
Description	string
Likes	int

Questions

Id (PK)	int
AuthorId (FK)	int
Author	User
Description	string

Data Dictionary

User

No	Field name	Data type	Required	Unique	PK / FK	Ref. Table
1	UserId	int	true	true	PK	-
2	Email	string	true	true	-	-
3	Name	string	true	false	-	-
4	Password	string	true	false	-	-
5	PhoneNumber	int	true	false	-	-
6	Organization	string	true	false	-	-
7	ProfilePhoto	string	true	false	-	-

Article

No	Field name	Data type	Required	Unique	PK / FK	Ref. Table
1	ArticleId	int	true	true	PK	-
2	AuthorId	int	true	false	FK	User
3	Title	string	true	false	-	-
4	Description	string	false	false	-	-
5	likes	int	false	false	-	-

ArticleComments

No	Field name	Data type	Required	Unique	PK / FK	Ref. Table
1	ArticleCommentId	int	true	true	PK	-
2	AuthorId	int	true	false	FK	User
3	ArticleId	int	true	false	FK	Article
4	Description	string	false	false	-	-
5	likes	int	false	false	-	-

Answers

No	Field name	Data type	Required	Unique	PK / FK	Ref. Table
1	AnswerId	int	true	true	PK	-
2	AuthorId	int	true	false	FK	User
3	QuestionId	int	true	false	FK	Question
4	Description	string	false	false	-	-
5	likes	int	false	false	-	-

Questions

No	Field name	Data type	Required	Unique	PK / FK	Ref. Table
1	QuestionId	int	true	true	PK	-
2	AuthorId	int	true	false	FK	User
4	Description	string	false	false	-	-

ArticleLikes

No	Field name	Data type	Required	Unique	PK / FK	Ref. Table
1	ArticleLikeId	int	true	true	PK	-
2	AuthorId	int	true	false	FK	User
3	ArticleId	int	true	false	FK	Article

Implementation Detail

Modules

Total modules :- 2

1. **Account Module :-**

   This module manages operations related to users' accounts like login, signup and forgot password. Existing users can login to the system by providing their email-id and password. New users can create their account by providing the required details. Users can follow other users to see their articles. And also can unfollow the user.

2. **Article Module :-**

   This module manages operations related to articles. Users can publish new articles. They can view a list of their own published articles.They can update & delete their own published articles. Users can see other user's articles by following that user. And users can provide their feedback by providing comments and likes to the articles. Users can report the content. Users can dislike the articles if they don't like them.

Function prototypes which implements major functionality.

- Public void register (User user)
- Public void login (string username, string password)
- Public void followUser (int userId)
- Public Article addArticle (Article article)
- Public ArticleComment postComment (ArticleComment articleComment)
- Public ArticleComment[] getArticleComments (int articleId)
- Public Article[] getAllArticles ()
- Public void likeArticle (int articleId)

Testing

Unit testing of each module was done after successfully completing the module. Each module was tested individually before integrating them with the whole system. After integrating each module with the system, integration testing was done in order to check if modules are working properly together. After completing all integrations, black-box testing of the whole system was carried out to ensure the system works in a correct manner.

Black box testing of Major functions of the system

1. **Log in to the system.**

   Case 1 : Invalid Username or password entered by the user.
   Output : Error message on the screen saying "Invalid credentials"

   Case 2 : Valid credentials.
   Output : The user is redirected to the Dashboard page.

2. **Publish Article**

   Case 1 : Some of required fields missing in input.
   Output : Model validation errors will be displayed to the user.

   Case 2 : All input data are valid.
   Output : Article published successfully.

3. **Like/Dislike Article.**

   Case 1 : User is not logged in.
   Output : Redirected to the login page with error message "Please login..!".

   Case 2 : If an article is already liked.
   Output : Error message will be displayed, "Article already liked.!"

   Case 3 : If an article is already disliked.
   Output : Error message will be displayed, "Article not liked yet.!"
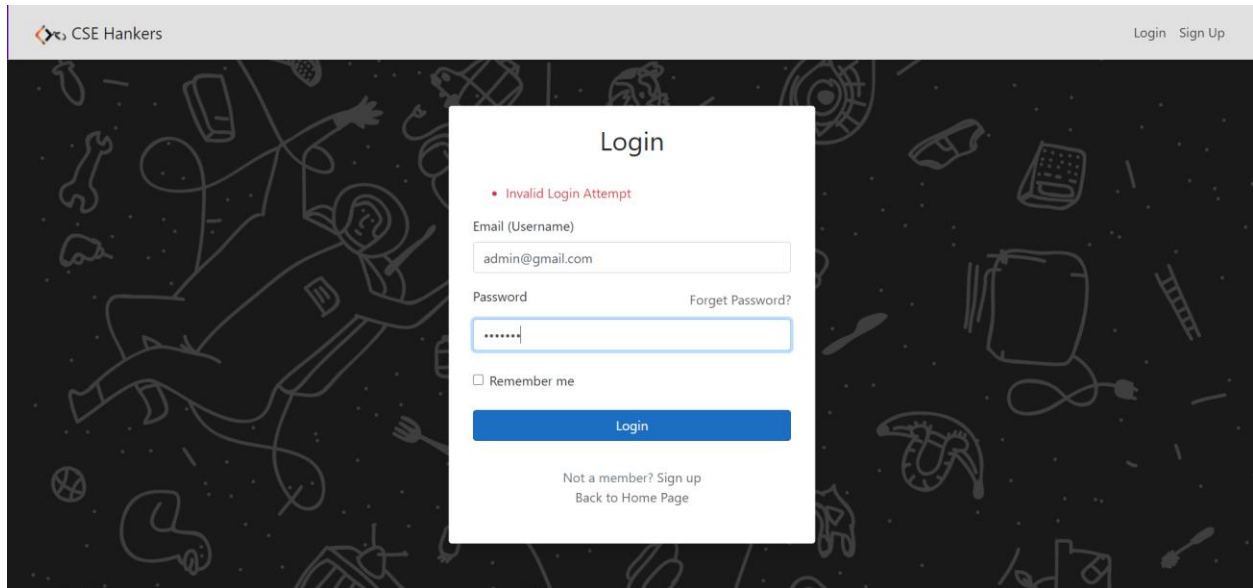
   Case 4 : If an article is not liked and like operation is performed.
   Output : Success message "Article liked" shown and like symbol color changes.

Screenshots

1. **Login page - with error messages**

2. **Registration page- with error messages**

3. **Dashboard**

4. Publish article
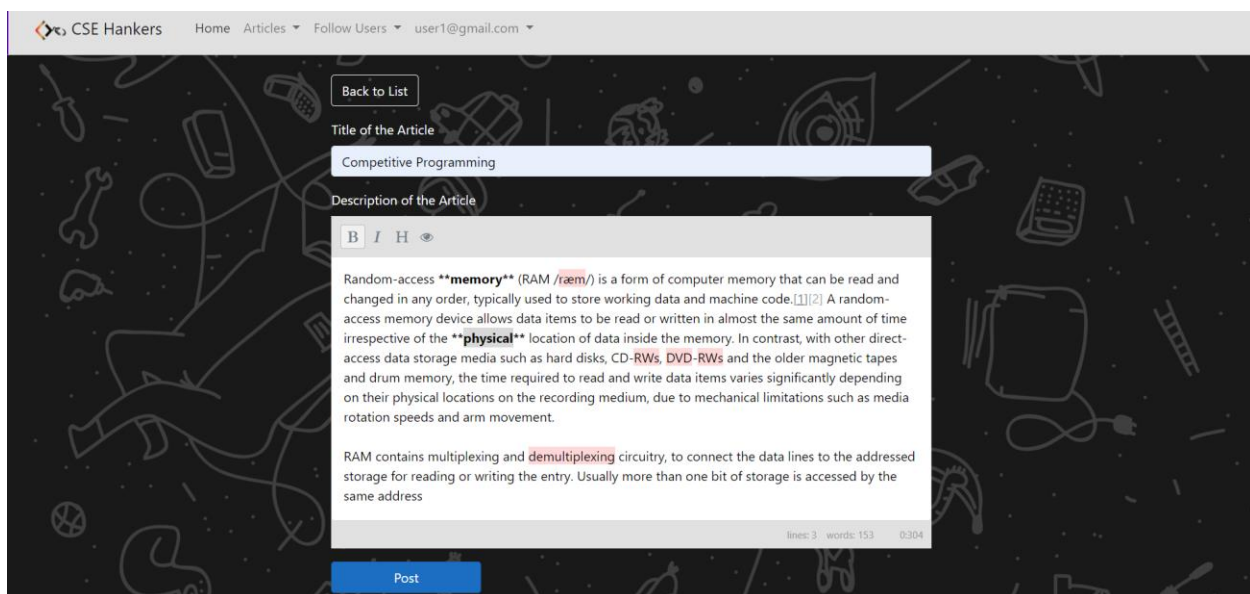
5. Pagination
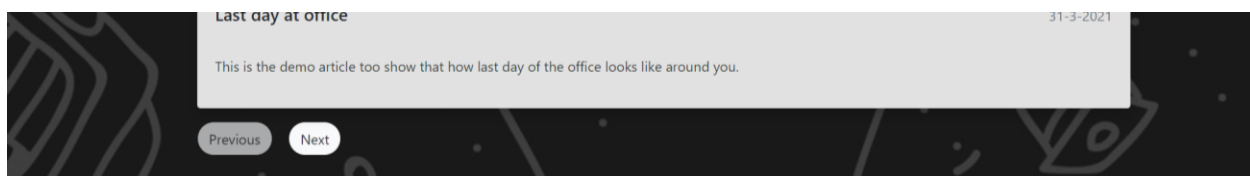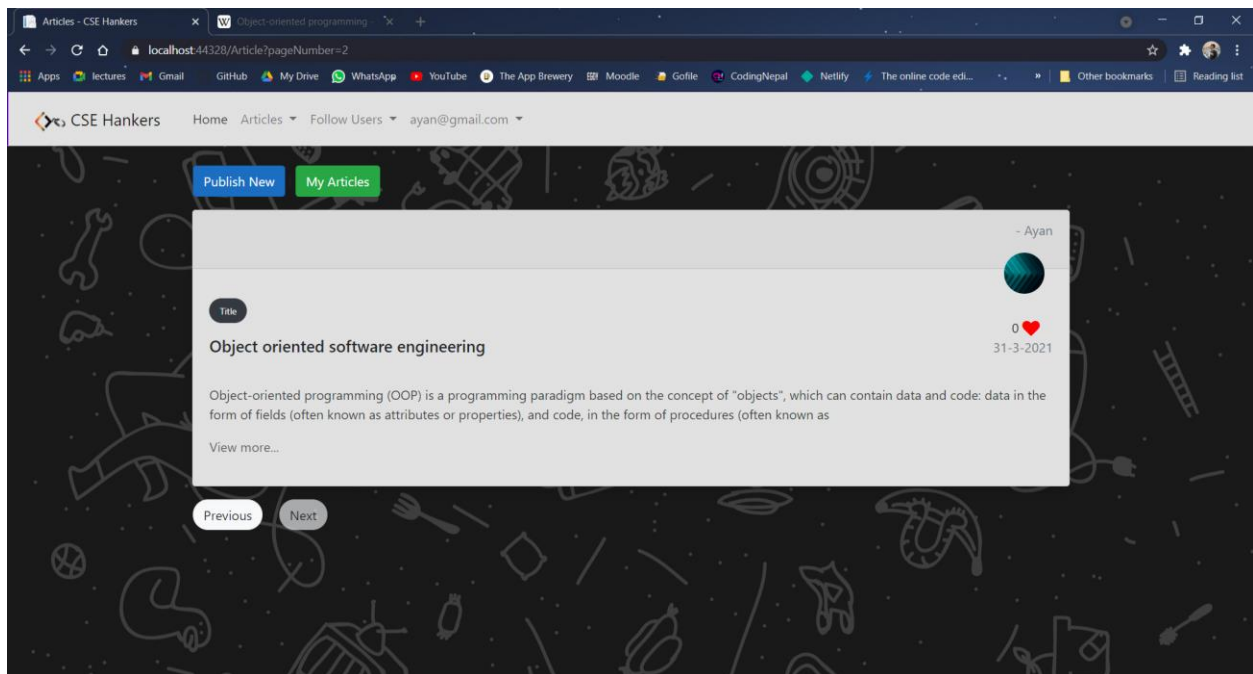
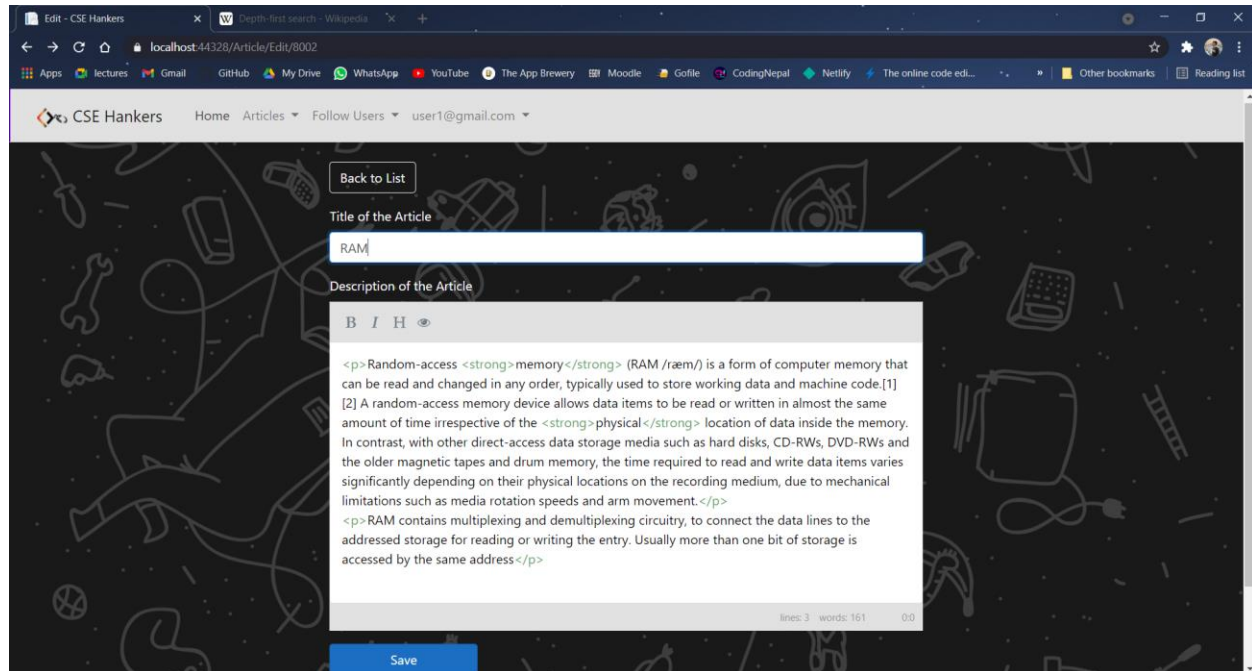## 6. View article details & add comment
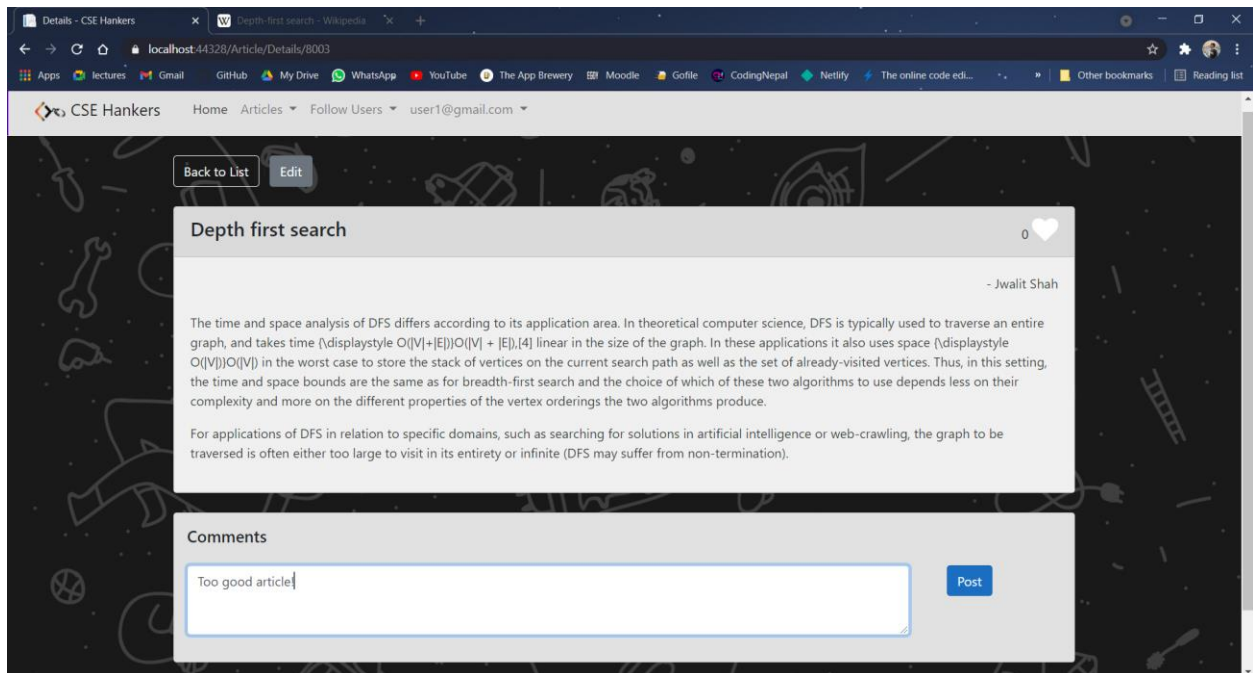
7. View comments
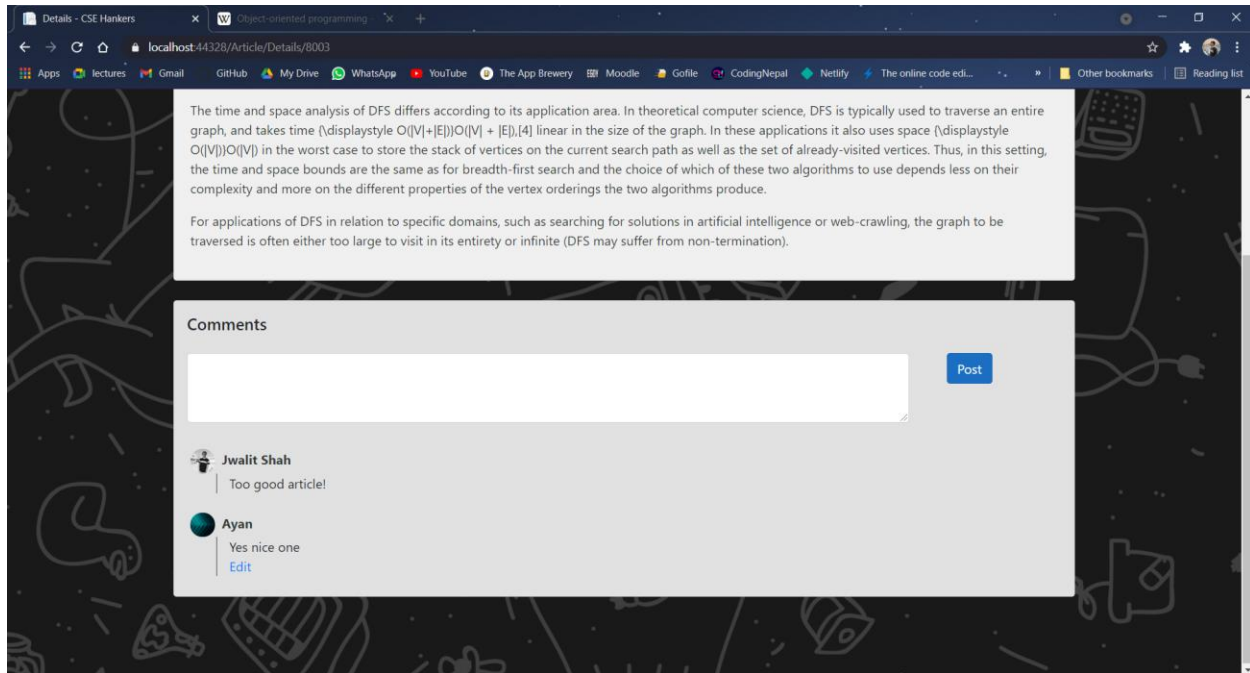
8. Edit comment
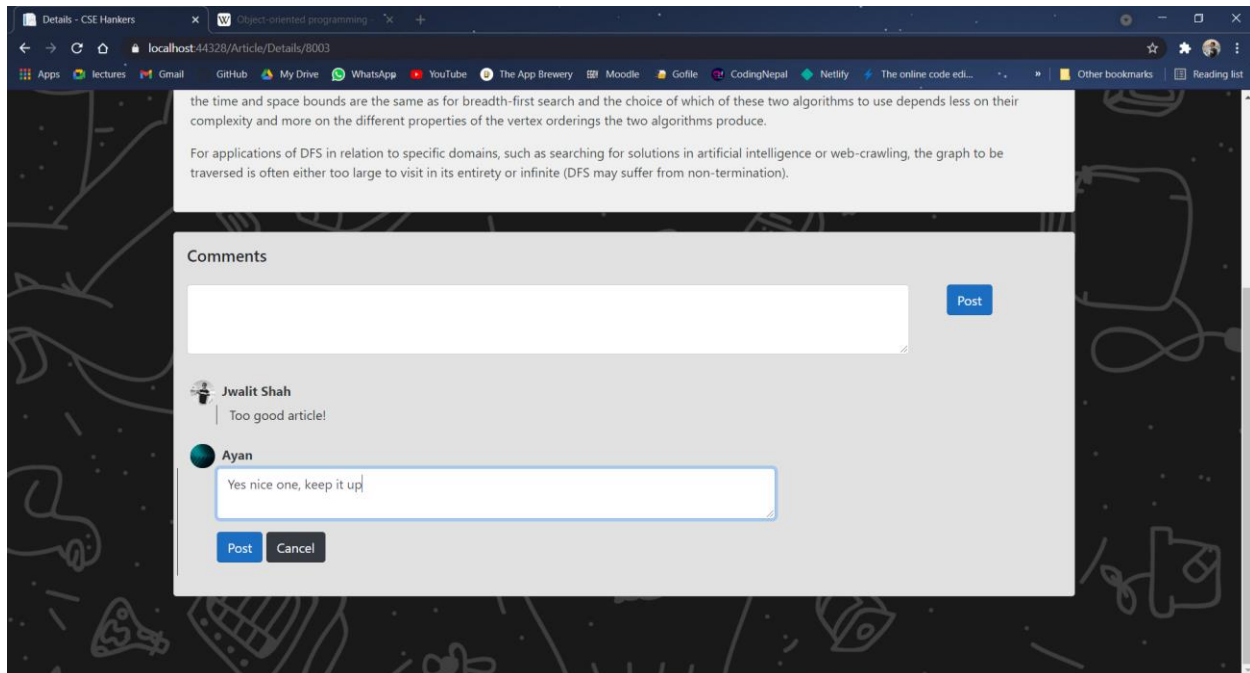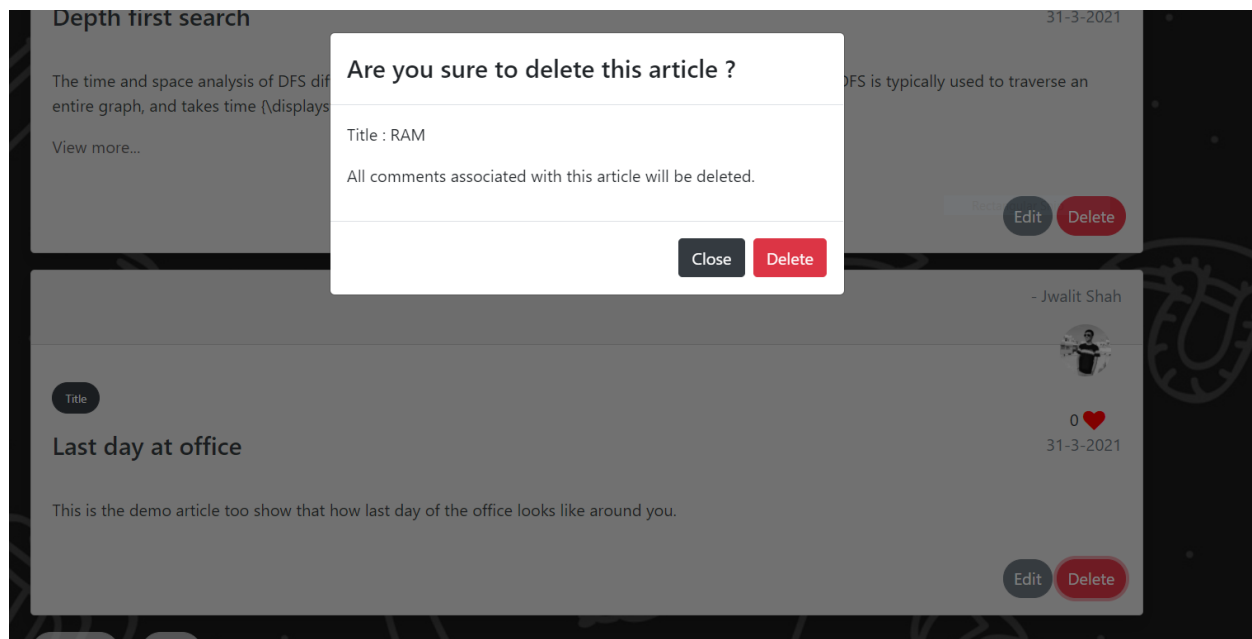
9. Unfollowed Users that are on platform
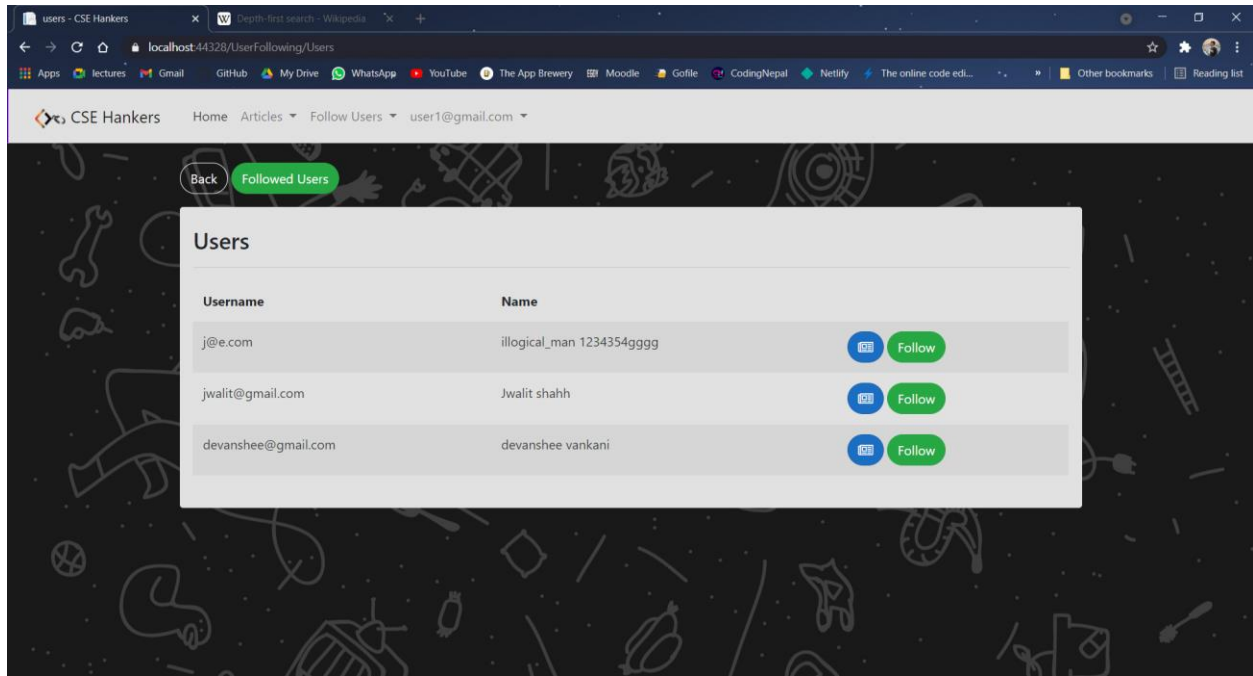
10. My Followed users
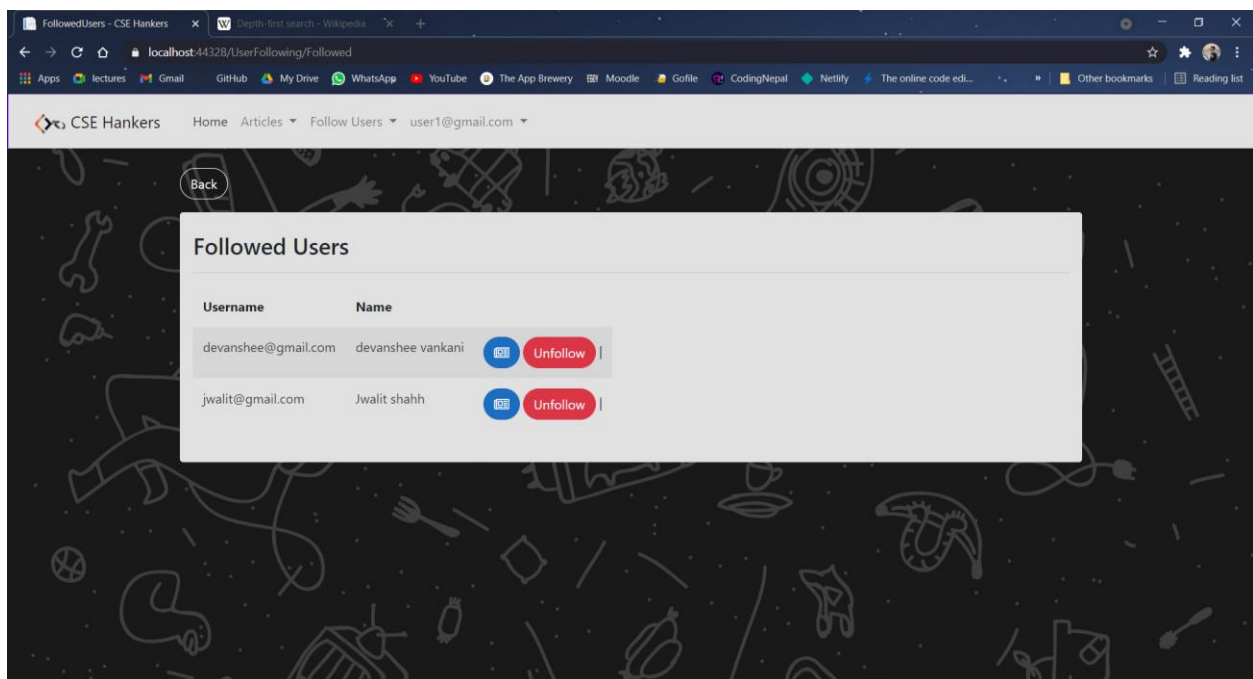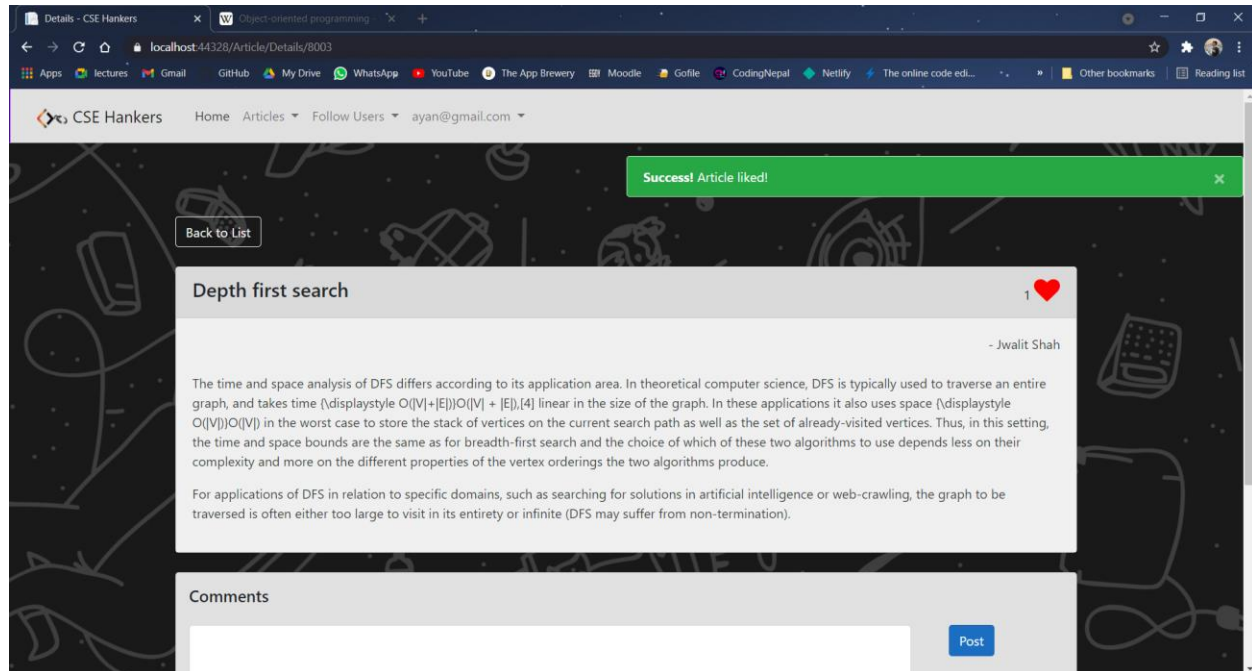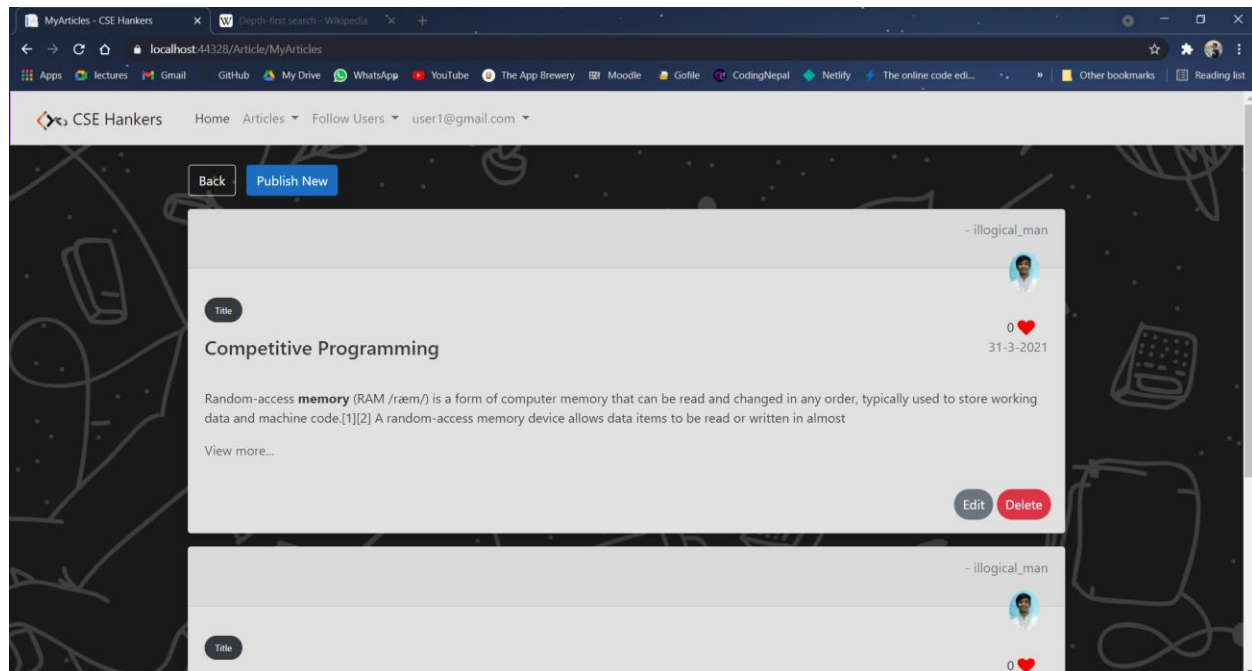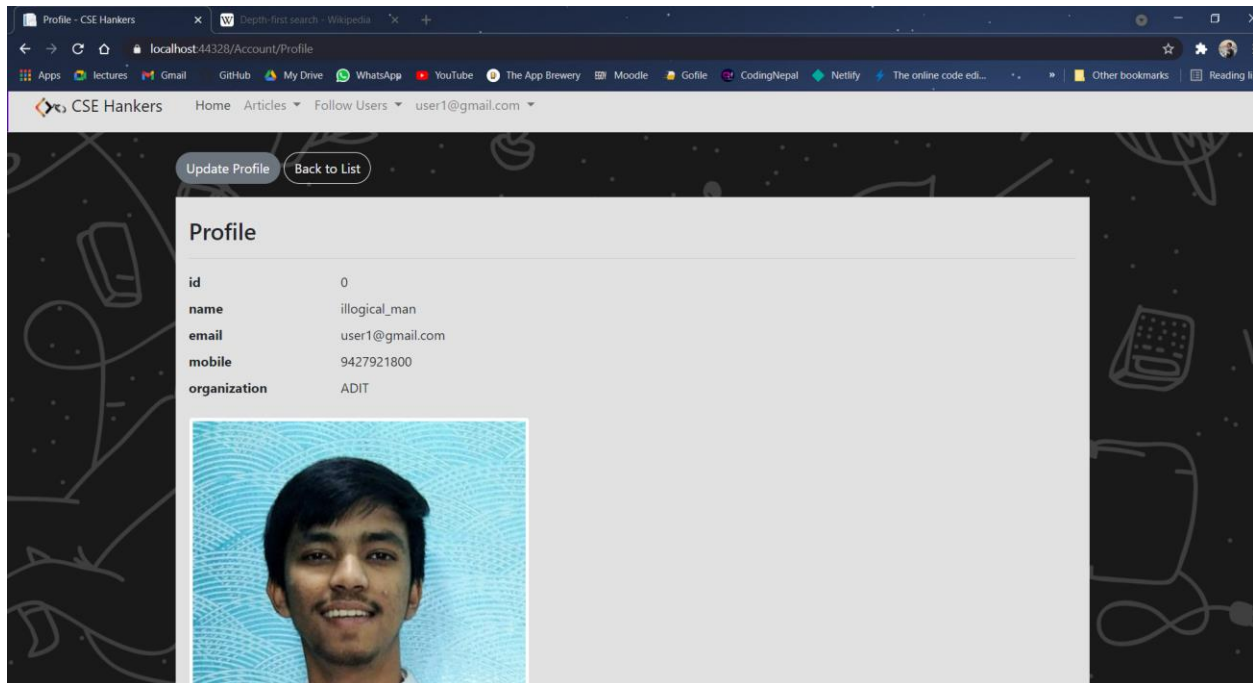
11. Edit article

12. Delete article
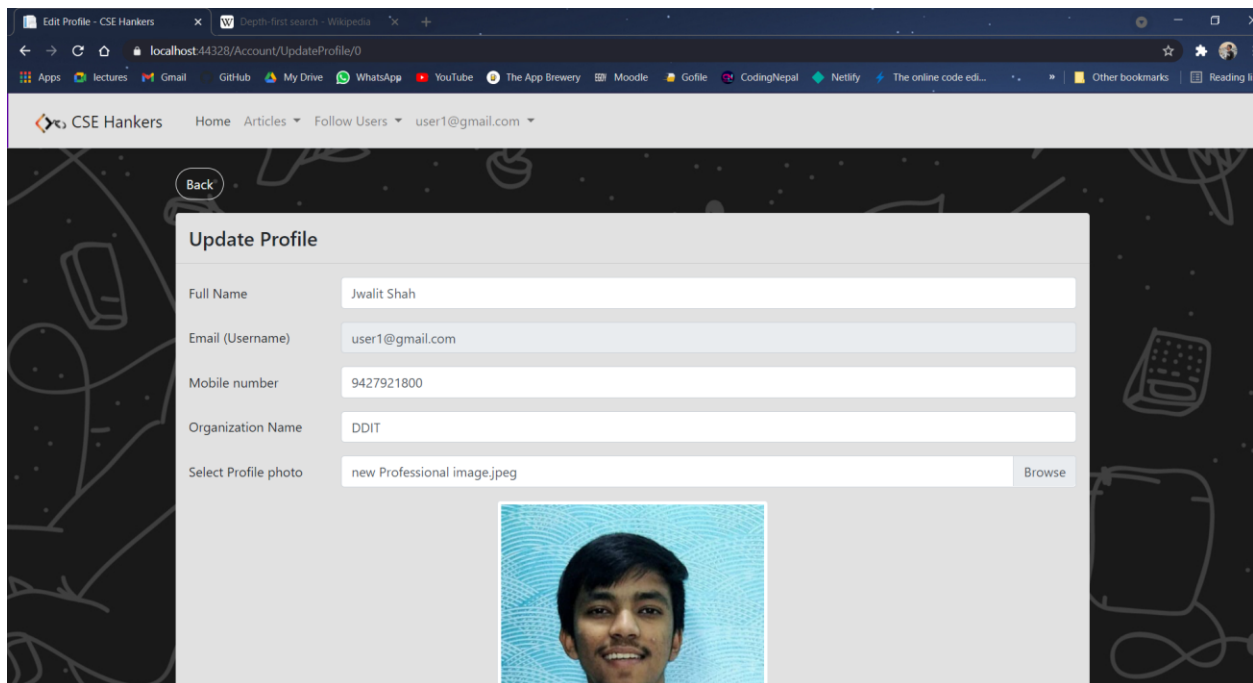
13. Like article
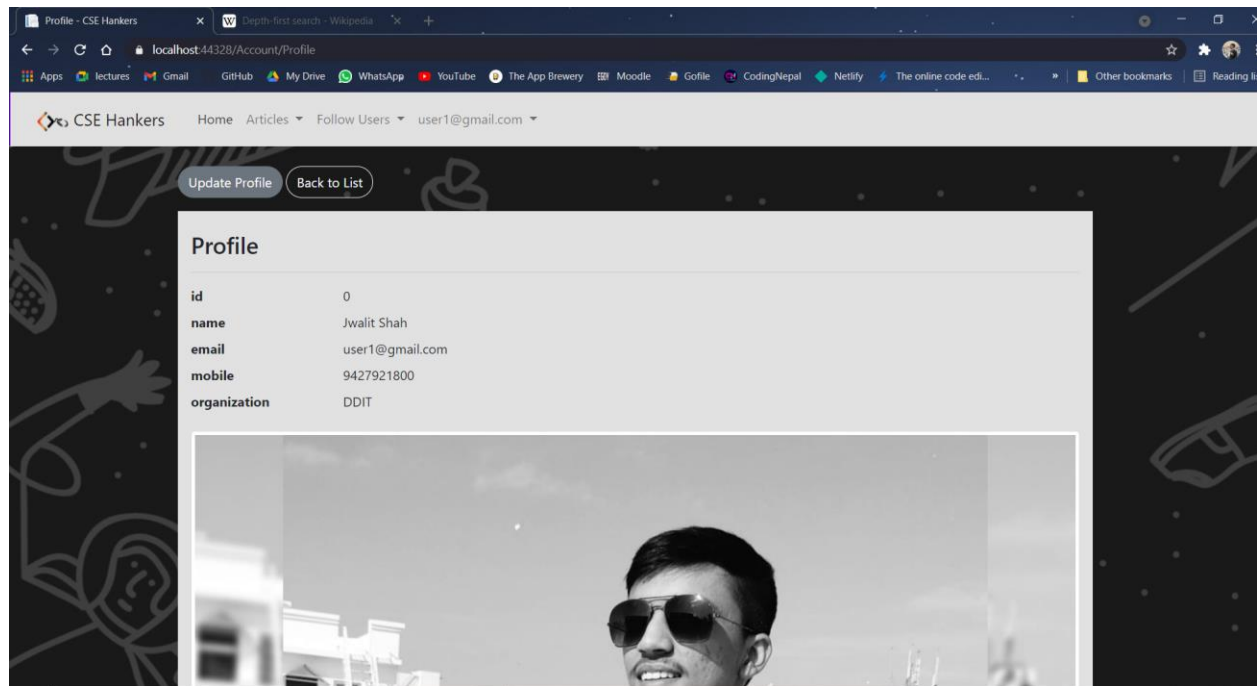

14. My article list
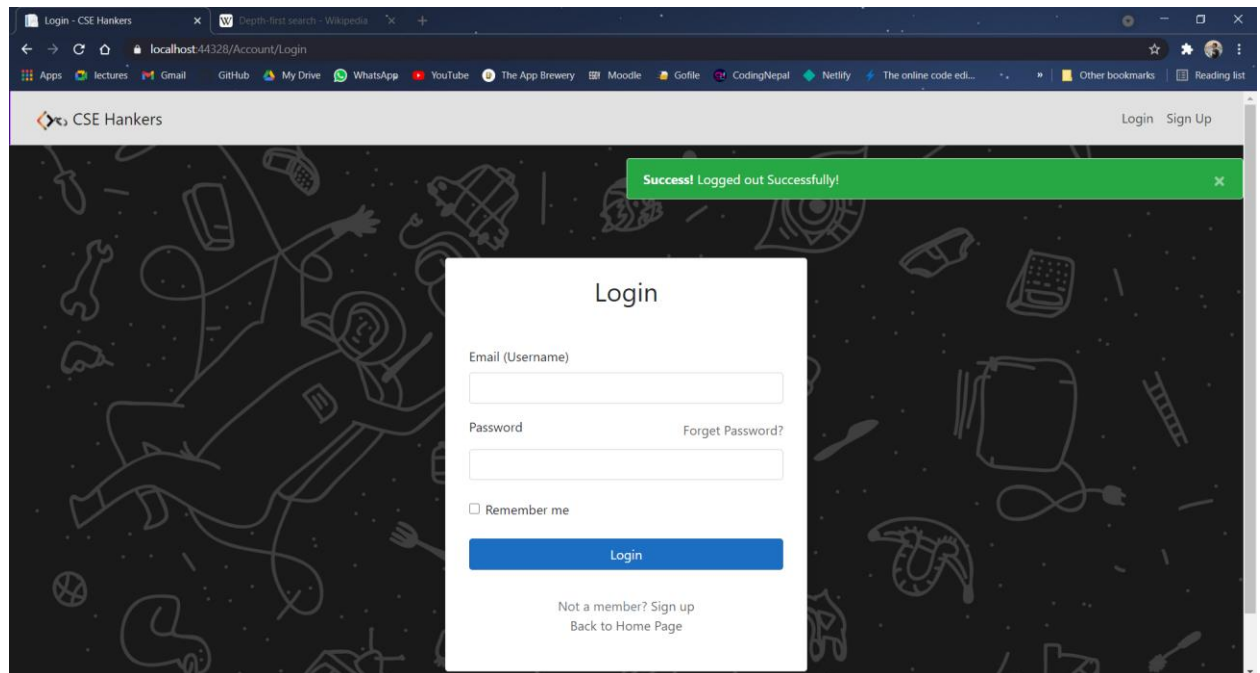
15. View profile

16. Update profile

17. After updating profile

18. Logout

Conclusion

Functionalities implemented successfully:

- Login by user.
- Registration by user.
- Follow user
- Unfollow user
- Publish article
- Update article details
- View articles list of own
- View followed user's articles list
- View article
- Delete article
- Give comment on article
- Update comment
- Like article
- Dislike article
- View other user's profile
- Update own profile
- Upload the profile picture
- Logout from the system

Limitation and Future Extensions

- ## Limitations
  - User can't login in case if password is not known
  - User can not change their username after registration

- ## Functionalities not implemented
  - Forgot password
  - Add, update, delete, view Questions
  - Add, update, view Answers
  - Like/ Dislike to answers

- ## Possible future extensions
  - Users can update their old password to the new password using forgot password in case if password is not known.
  - Users can publish and manage the questions posted on the forum.
  - Users can give answers to the questions posted on the forum
  - Users can provide their feedback to posted answers by liking or disliking the answers.
  - Users can give their feedback to the system administrator.

Bibliography

To build this project we have taken references from following websites,

- [Quora](#)

- [Article Services](#)

- [C-Sharp corner](#)

- [Stack overflow](#)