

PeerHub
A
MINOR PROJECT-I REPORT
Submitted in partial fulfillment of the requirements
for the degree of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

By
GROUP NO. 60
Amit Ranjan 0187CS191020 Ananya Tapi 0187CS191024
Ashutosh Pandey 0187CS191039 Hamza Kamal 0187CS191059

Under the guidance of
Dr. Vasima Khan
(Associate Professor)



Sep-2021

Department of COMPUTER SCIENCE & ENGINEERING
Sagar Institute of Science & Technology (SISTec)
Bhopal (M.P.)

Approved by AICTE, New Delhi & Govt. of M.P.
Affiliated to Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal (M.P.)

Sagar Institute of Science & Technology (SISTec), Bhopal
Department of COMPUTER SCIENCE & ENGINEERING
Bhopal (M.P.)



Dec-2021

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Minor Project-I Report entitled **Peerhub**, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** and submitted to the Department of Computer Science & Engineering, *Sagar Institute of Science & Technology (SISTec)*, Bhopal (M.P.) is an authentic record of my own work carried out during the period from July-2021 to Dec-2021 under the supervision of Dr. Vasima Khan (Project Guide). The content presented in this project has not been submitted by me for the award of any other degree elsewhere.

Signature

Amit Ranjan 0187CS191020

Ananya Tapi 0187CS191024

Ashutosh Pandey 0187CS191039

Hamza Kamal 0187CS191059

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Project Guide

HOD

Principal

ACKNOWLEDGEMENT

Presentation inspiration and motivation have always played a key role in the success of any venture.

In the accomplishment of this project successfully many people have bestowed upon us their blessings and the heart pledged support, this time we are utilizing to thank all people who have been concerned with this project.

We express our sincere thanks to Mr. Keshavendra Choudhary (Principal of Sagar Institute of Science and Technology (SISTec), Gandhi Nagar, Bhopal).

We pay our deep sense of gratitude to Bhavna Gupta (HOD) CSE department, Sagar Institute of Science and Technology (SISTec), Gandhi Nagar, Bhopal to encourage us to the highest peak and to provide us the opportunity to prepare the project. We feel to acknowledge our indebtedness and deep sense of gratitude to both our teachers as well as our project guide Dr. Vasima Khan (Associate Professor) kind supervision was given to us throughout the project.

Last, but not the least, we immensely obliged our friends for their elevation inspiration, encouraging guidance and kind supervision in completion of our project.

Group Members: -

Amit Ranjan 0187CS191020

Ananya Tapi 0187CS191024

Ashutosh Pandey 0187CS191039

Hamza Kamal 0187CS191059

Abstract

PeerHub as the name suggests it's a concept that we came up with and materialized in the form of a forum where your “ Peers ” can meet on a “ Hub” and make their own communities which like-minded people can join. Also, peers can comment on a post in the community as well as reply to the comments made by their peers. People can also choose their favorite communities so that they can see the latest posts and comments on their personal feed. Users will get recommended posts and communities based on their interactions i.e. post likes, the tags and communities they follow. This platform will bring like-minded people together in which they can grow as a whole community, learn new things about their favorite topics and also discuss ongoing things going on in the world of that community. We are using the MVC design pattern for application architecture and REST as the API architectural style.

Functional and Non-Functional Requirements Of PeerHub

Requirement

1.1 Functional Requirements:

The functional requirements provide a basic system behavior where requirements define the basic behavior of the application.

User Requirements:

1. User authentication (username, password)
2. Check Communities.
3. Check Post with associated Comments and Replies.
4. Vote (positive or negative) if a post is good.
5. Create Communities
6. Create Posts.
7. Create Comments and Replies.
8. Modify or delete Posts, Communities, Comments and Replies.
9. Change password (with Registered email ID)
10. Join a community.

Registration:

- 1) User registration
 - i) Personal information
 - ii) Professional information

Registration:

1. User Registration

Input – Username, Email and Password.

Output - Either Registration successful or Registration failed.

Description – Using this feature, users can register and gain authorization to create/modify Posts, Communities, Comments, Replies and other Users.

User Requirements:

1. User Authentication

Input – Username, password

Output – Login successful or login failed.

Description – With this feature, a user can login and the server will provide a token for authentication so the user's session will be maintained (expire after 6hrs).

2. Check Communities

Input – Click on a community or go to a URL of one.

Output – Display community with a post feed.

Description – Users can view a community and read the posts associated with that community.

3. Check Post with associated Comments and Replies.

Input – Click on a Post or go to a URL of one.

Output – Display Post with a comment feed.

Description – Users can view a Post and read the comments and replies associated with that community.

4. Vote (positive or negative) if a post is good.

Input – On a post, hitting the upvote or downvote button.

Output – Make a vote to that post.

Description – Users can vote posts and decide which posts they liked and disliked

5. Create Communities

Input – Community name, Description, Tags (Optional) and a logged in user.

Output – Creates a community with specified name and description.

Description – Users are able to build communities and Create posts and discussions relevant to the community.

6. Create Posts

Input – Post Title, Content, Tags (Optional) and a logged-in User.

Output – Creates a post in a community

Description – Users can create posts using this function and have discussions and/or share information relevant to the community they are posting into.

7. Create Comments and Replies

Input – Comment/Reply Content and a logged-in User.

Output – Creates a comment/Reply in a post

Description – Users can create comments using this function and have discussions and/or share information relevant to the post they are commenting on. Users can also reply to said comments.

8. Modify or Delete Posts, Communities, Comments and Replies.

Input – Posts/Communities/Comments/Replies and a logged-in User.

Output – Modifies or Delete said content.

Description – Users can delete Posts/Communities/Comments/Replies they created (Not another user's content)

9. Change password (with Registered email ID)

Input – username

Output – New Password.

Description – Users can change their password by getting an OTP on their mail address.

9. Join a community.

Input – Community to join

Output – Joined the provided community

Description – Users can join communities to get a better feed of subscribed content.

1.2 Non-Functional Requirements:

Non-Functional Requirements are the constraints or the requirements imposed on the system. They specify the quality attribute of the software. Non-Functional Requirements deal with issues like scalability, maintainability, performance, portability, security, reliability. They are contrasted with functional requirements that define specific behavior or functions.

The non-functional requirements of PeerHub are as:

1. Security
2. Reliability
3. Availability
4. Maintainability
5. Supportability

Each non-functional requirement is described below:

1. Security

The system uses SSL (secured socket layer) in all transactions that include any confidential user information. The system must automatically log out all users after the token expires. Creating, Updating and Modification capabilities are only provided to a logged in user and thus only authenticated entities can make change. User's password is not saved directly but in the form of a bcrypt hash with salting. Only user id and username are transferred in the token (encrypted JWT) to maintain the session.

2. Reliability

Project reliability depends on the reliability of the components. We are using MongoDB Atlas to host the Database (mongoDB) which has an uptime of 99.995% and uses sharding to distribute data across machines while maintaining continuous backups. Also, MongoDB allows Horizontal scaling so in case the storage capacity is filled then another system can be added easily.

3. Availability

The Application will be available 24/7 and users can access it through the means of a web browser (excluding IE11). If the application is down, an alternate page should be shown.

4. Maintainability

The code is written in a modular and functional way so changes have very few regressions as state is not manipulated a lot. Using express as the backend has allowed the use of middleware to further modularize code and increase cohesion meanwhile decreasing coupling between modules. We have unit tests with a test coverage of over 90% so in case of a failure, we can find the regressions early on. MongoDB is used as the database with Atlas hosting it, so database is taken care of by MongoDB.

5. Supportability

The code and supporting modules of the system will be well documented and easy to understand. Online User Documentation and Help System Requirements.