The APSIT Community

Submitted in partial fulfillment of the requirements of the degree of

BACHELOR OF COMPUTER ENGINEERING

by

Sana Sheikh (20102081)

Abhay Sharma (20102065)

Abhishek Tiwari (20102064)

Vaidik Vadhavana (20102197)

Guide:

Dr. Rahul Ambekar

Department of Computer Engineering

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

(2022-2023)

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

CERTIFICATE

This is to certify that the Mini Project 2B entitled “**The APSIT Community**” is a bonafide work of **“Sana Sheikh (20102081), Abhay Sharma (20102065), Abhishek Tiwari (20102064), Vaidik Vadhavana (20102197)**” submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Engineering.**

Guide: Prof. Rahul Ambekar	Project Coordinator: Prof. D.S. Khachane	Head of Department: Prof. S.H. Malave

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

Project Report Approval for Mini Project-2B

This project report entitled **“The APSIT Community”** by ***Sana Sheikh (20102081), Abhay Sharma (20102065), Abhishek Tiwari (20102064), Vaidik Vadhavana (20102197)*** is approved for the partial fulfillment of the degree of ***Bachelor of Engineering*** in ***Computer Engineering***, ***2022-23***.

Examiner Name Signature

1. ______________ ______________

2. ______________ ______________

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sana Sheikh- 20102081

Abhay Sharma- 20102065

Abhishek Tiwari- 20102064

Vaidik Vadhavana- 20102197

Date:

Abstract

The APSIT Community is a student-centric platform designed to promote awareness of activities on and around the APSIT campus. Its main goal is to create a community culture among students and encourage an interdisciplinary approach to problem-solving. The platform provides a central site for deadline monitoring and encourages seniors to mentor their fellow peers and juniors. Students can publish about particular areas of interest and share personal accomplishments to assist others. The platform is designed to be interactive, with a comment feature for viewers to ask doubts related to the post and react to posts. The achievement page displays breakthroughs of star students, while the news page provides information about recent activities, events, competitions, and placement notices. Overall, The APSIT Community is a website for APSIT students to post their doubts, achievements, milestones, and connect with like-minded people on campus to facilitate overall growth according to their domain of interest.

Keywords: Post, community, inter-disciplinary, student-centric.

CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

The site titled APSIT Community is designed keeping students' comfort and convenience in mind. The objective of the project is to build an online platform that is centered on the requirements of APSIT students and is developed and designed keeping that purpose in mind. The APSIT Community aims to surpass the current systems in terms of effectiveness and efficiency. The primary goal is to encourage connectivity among different departments.

An interdisciplinary approach is essential today owing to the industry's rapidly increasing demand and competitiveness. Hopefully, this project will prove beneficial to all the aspiring engineers registered under APSIT. This project will offer comprehensive information about what is occurring in APSIT. The Post feature will enable discussion with peers. By creating a new post, the student can have the authority to edit as well as delete it. The student can bookmark any post according to his/her area of interest. The user can like or comment on any post on which they want to provide their input. On the News page, it also provides regular news regarding what really is ongoing in the institution. Displaying the accomplishments of the existing APSIT students and commemorating noteworthy achievements on the Achievement Page will boost student’s achievement by fostering healthy peer competition.

Once a student has set up an account, they can access the website (proper validation is implemented). The student can freely surf the website after making an account in accordance with his or her preferences. The About Us page gives an overview of the developers of the community. The Help Desk provides important information to students as well as a feature to report any unauthorized activity on the website by Contact Us feature. If the site is idle for 2 hours the student has to log in again. By creating a new post and highlighting milestones, experiences, projects, etc., students can contribute to the website.

The Post feature is the spotlight of this project. It will cater to the interests of all students. The students of APSIT can create an account with their unique Moodle id and password. Once an account is created, they will be able to login to the website. Students can browse freely throughout the website. They can report any errors they face. Students are also given the terms and conditions of website use. The site is created for the benefit of students and hence is aimed to boost productivity among APSIT students. The current existing system does not have a facility for students to post their ideas and interests.

The APSIT Community is developed keeping in mind a student centric approach. An extensive study of SNS platforms available for students was studied. There was no common platform for students other than LinkedIn, Moodle, Stack overflow to ask and post their queries on. There is a noticeable gap in the information available for students on these sites. The main issue here was studied, and it was decided to make a website where students have access to explore new things by the thoughts of other students. The age gap won’t be so diversified on The APSIT community henceforth the queries of students will be better understood and appreciated.

To provide a holistic approach it is kept in mind to include students’ input to make the website more catered to their use. In the future implementation of the project, we’ll work on algorithms which will deal with censorship issues and eliminate any before it arises. It is also under plan to integrate the entire site with Moodle so students can easily keep track of their submissions and get timely reminders, so they don’t miss deadlines.

The main purpose is to network within the campus. Users are also expected to upload their achievements, projects they work on, doubts regard to career and area of interest. It will make APSIT a digital campus. By providing everything under one roof, it’ll not only bridge the gap between the students of different departments but also provide a healthy and friendly environment all over the campus.

Chapter 2

Literature Survey

1. **Nike Arnolda, Trena Paulusb. (2010).Using a social networking site for experiential learning. Appropriating, Lurking, modeling and community building.**

   Social networking sites can be valuable tools for experiential learning, allowing individuals to engage in appropriation, lurking, modeling, and community building. Appropriation involves utilizing existing information and resources from social networking sites to enhance learning, while lurking involves observing online conversations and interactions without actively participating. Modeling entails learning from others who are active participants in the community, while community building involves developing relationships with others on the platform and collaborating on projects or assignments. By leveraging these tools and approaches, individuals can deepen their understanding of various topics and enhance their learning experiences in online environments.

2. **Bin Masrek M (2007). Measuring campus portal effectiveness and the contributing factors**
   Interdisciplinary integration in engineering education is a growing trend that recognizes the importance of preparing engineering graduates with a broad set of skills and knowledge. The traditional approach to engineering education has been focused on the development of technical skills in a specific discipline, such as civil, mechanical, or electrical engineering. However, as the problems faced by society become more complex and interconnected, there is a need for engineers who can work across disciplines and apply a variety of approaches to solve real-world challenges. Interdisciplinary integration in engineering education involves incorporating concepts and techniques from other fields, such as biology, chemistry, physics, mathematics, and social sciences, into engineering curricula. For example, engineering students may study biology to understand how living organisms can be used to create sustainable technologies, or economics to learn about the financial implications of engineering projects. By learning about these diverse fields, students gain a broader perspective that enables them to see the connections between different disciplines and apply a multidisciplinary approach to problem-solving.

3. **Fonseca Cacho, Jorge. Using Discord to Improve Student Communication, Engagement, and Performance**

   Since the public release of Discord in May 2015, gamers have widely used it as an additional application to communicate with their teammates while playing games. However, this application is also used as an alternative media of communication in teaching and learning activities. Therefore, this study aims to determine students' acceptance of Discord as alternative teaching and learning media. Data were obtained using a simple questionnaire. Furthermore, their experience and acceptance towards this application are explored using the technology acceptance model (TAM) framework. The result showed that most participants confirmed that Discord is a favorable alternative media due to the attractive user interface, completeness of features, and its ease of use. Therefore, this application, which was originally intended for the gaming, can surprisingly be used as an alternative online learning media.

4. **Secreto, Percia V. Pamulaklakin, Rhodora L.(2015).Learners' satisfaction level with online student portal as a support system in a distance e-learning environment.**

   The online student portal is an essential support system in distance e-learning environments, providing learners with access to a variety of resources and services, including course materials, discussion forums, virtual libraries, and administrative support. Learners' satisfaction level with the online student portal is a crucial factor that can impact their overall experience and success in the program. Research has shown that learners generally have a positive perception of the online student portal, citing its convenience, accessibility, and ease of use. However, factors such as technical issues, lack of personal interaction, and limited engagement with faculty and peers can negatively impact learners' satisfaction level. Overall, the online student portal plays a vital role in supporting learners in distance e-learning environments, and efforts should be made to ensure its accessibility, functionality, and user-friendliness to enhance learners' satisfaction and success.

RESEARCH PAPER	ANALYSIS
1.. Nike Arnolda, Trena Paulusb. (2010).Using a social networking site for experiential learning: Appropriating, Lurking, modeling and community building. 158 Citations. The Internet. Volume 13, Issue 4. Dec. 2010	Social networking sites can be valuable tools for experiential learning, allowing individuals to engage in appropriation, lurking, modeling, and community building. Appropriation involves utilizing existing information and resources from social networking sites to enhance learning, while lurking involves observing online conversations and interactions without actively participating.
2. Bin Masrek M (2007). Measuring campus portal effectiveness and the contributing factors. Campus Wide Information Systems 24 (5). Nov. 2007.	As the problems faced by society become more complex and interconnected, there is a need for engineers who can work across disciplines and apply a variety of approaches to solve real-world challenges.
3. Berezin (2020). Interdisciplinary integration in engineering education.42 Citations. JEE Vol. 109, Issue 3.	Interdisciplinary integration also encourages collaboration and communication between students and faculty from different disciplines, which leads to innovative solutions.
4. Secreto, Percia V. Pamulaklakin, Rhodora L.(2015).Learners' satisfaction level with online student portal as a support system in a distance e-learning environment.	This paper explains the use of social media in the recruitment process. The applications like LinkedIn are widely used.

Chapter 3

Problem Statement, Objective & Scope

Problem Statement

The goal is to design and implement a college community web-app using Nodejs, Python, Flask, and MongoDB for the students of APSIT. The app aims to inculcate inter-department and intra-department interaction, as there is a noticeable lack of awareness of ongoing activities and a common platform for communication. The current Moodle system does not offer real-time communication, making it difficult to keep track of individual activities and new clubs. Events need to be effectively included in the curriculum as part of overall activities to make campus life worthwhile. There is a lack of a suitable venue to highlight student's on-campus activities and inform them of available resources. Clubs and groups do not currently have a widespread presence on campus. Many current students are unaware of the qualifications and standards required by the industry, leading to a lack of networking opportunities. Some pupils are reluctant to express their opinions in person, making it challenging to share ideas freely. A suitable platform is needed to circulate internship and placement reviews from senior year students throughout the campus.

Objective

- To encourage communication amongst all APSIT students, regardless of branch.
- To offer a common platform for uniting all the clubs under a single community and eliminating ignorance across the university.
- To showcase students' achievements and the most recent updates.
- To alleviate uncertainties by teaching multidisciplinary problem-solving techniques.
- To give students access to a user-friendly platform that is focused on them.

Scope

- To provide a common site where every student enrolled in the institute can come together and form a community.
- To give information that will assist undergraduate students by giving them a variety of opportunities to interact, express their opinions, and discuss specific interest areas.

Chapter 4

Proposed System Architecture

- **Architecture / Block Diagram**

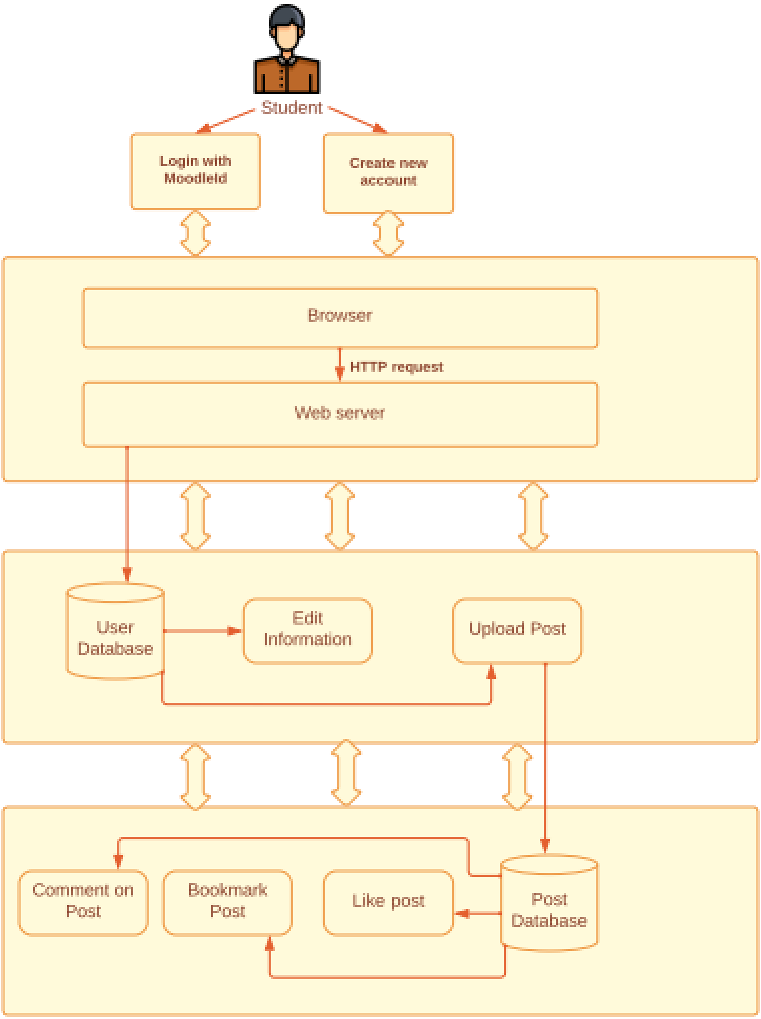


Figure 4.1 Architecture Diagram

An architectural diagram is a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element. Our focus is on the students as it is a student-centric platform. We have two databases, one for the user and the other for the post.

- **Data Flow Diagram (Level 0, Level 1 & Level 2)**

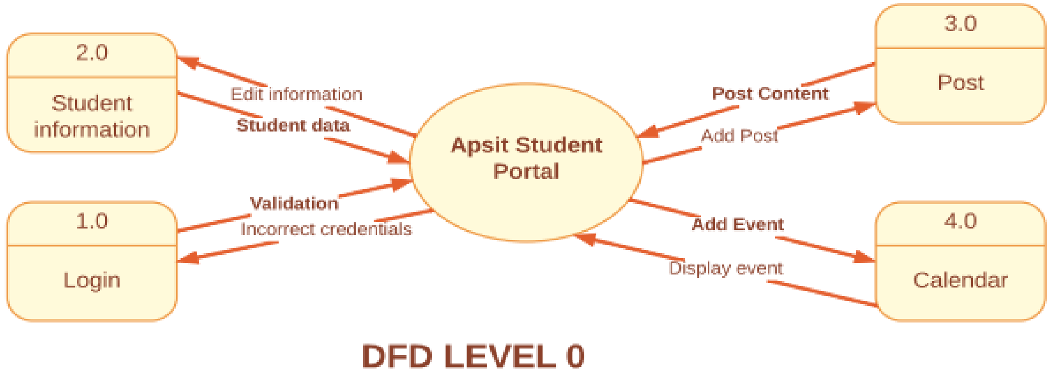


Figure 4.2 DFD Level 0

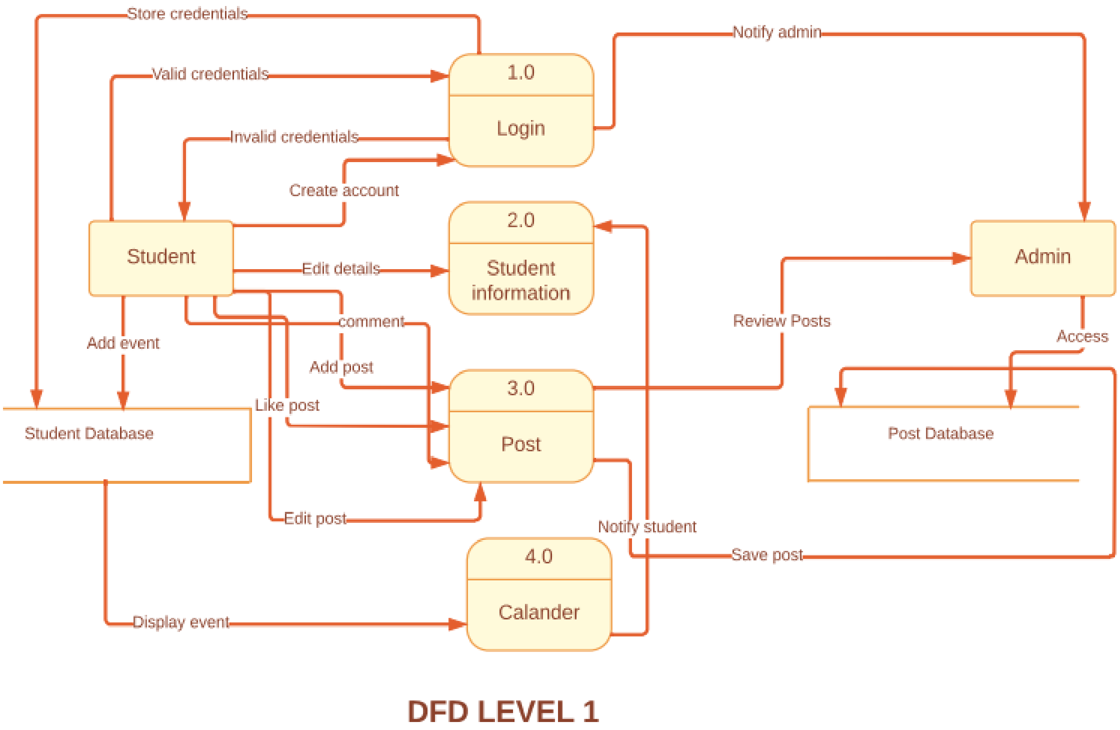


Figure 4.3 DFD Level 1

DFD is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.

- **Use Case Diagram**

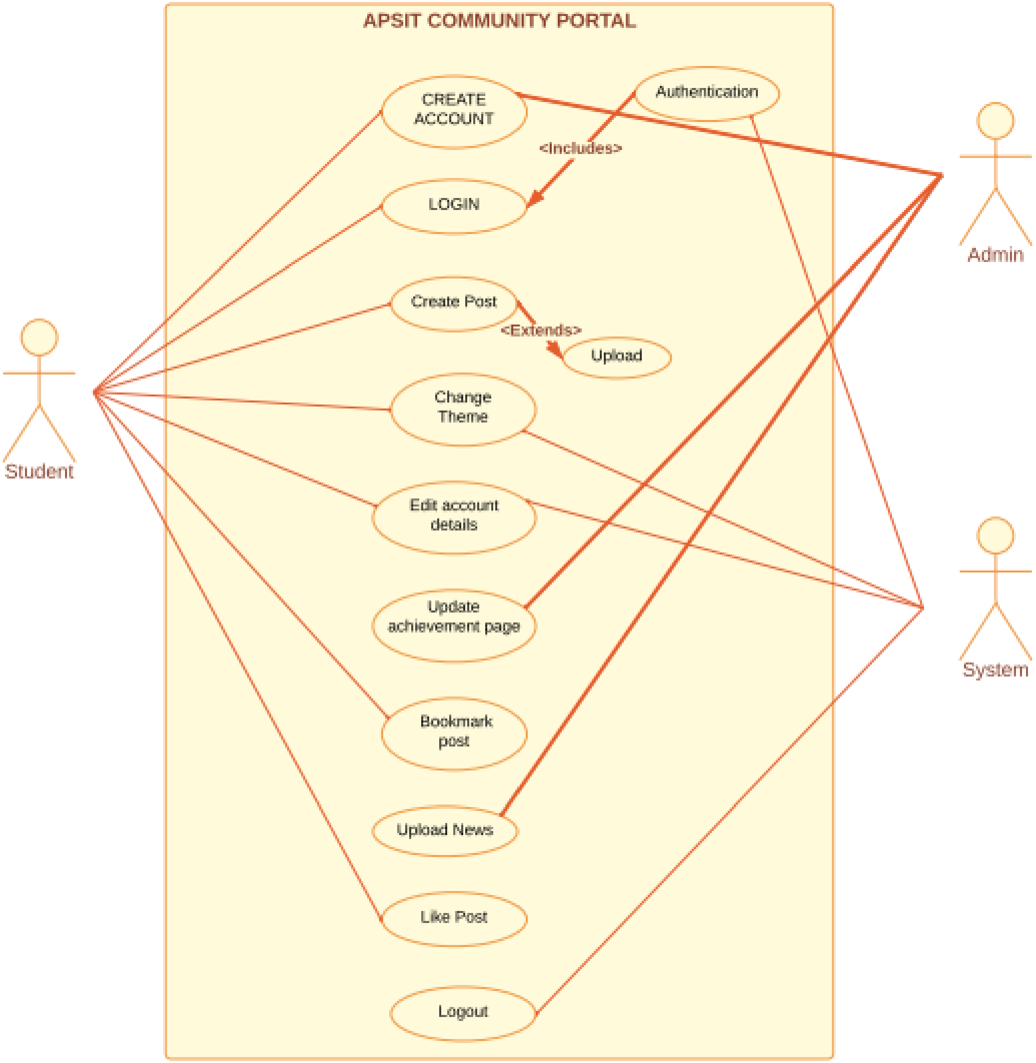


Figure 4.4 Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

- **Sequence Diagram**

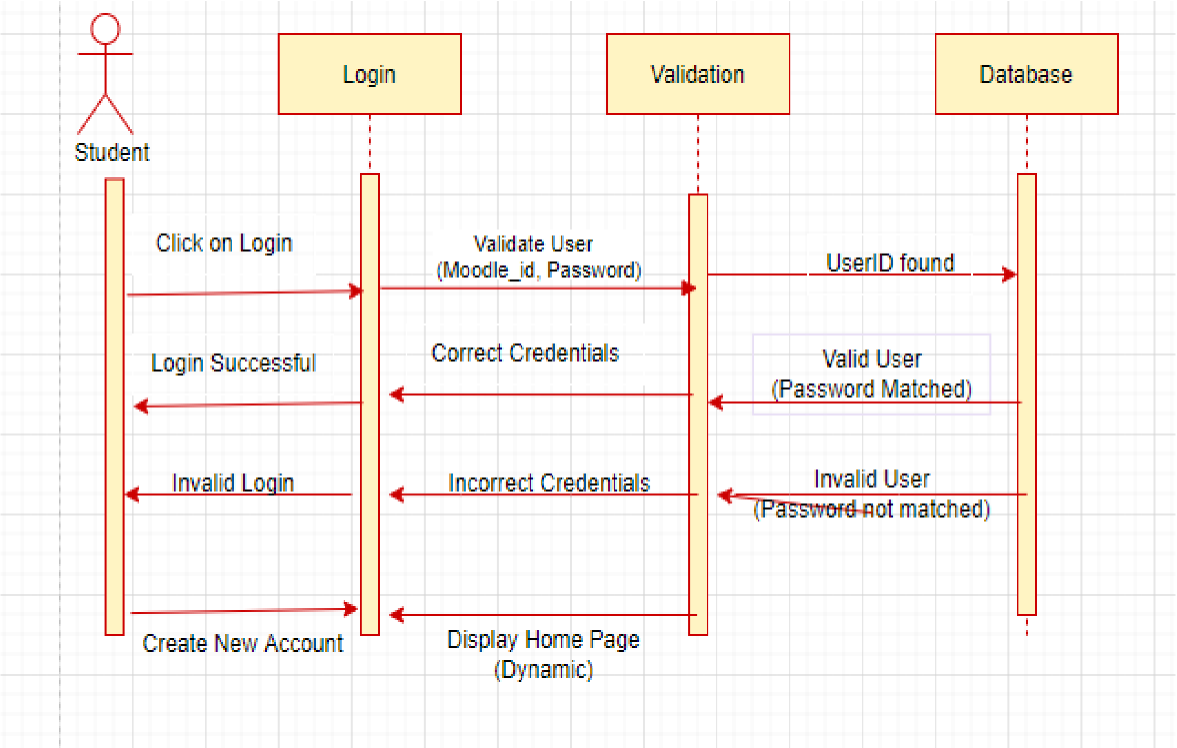


Figure 4.5 Sequence Diagram

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

- **Activity Diagram**

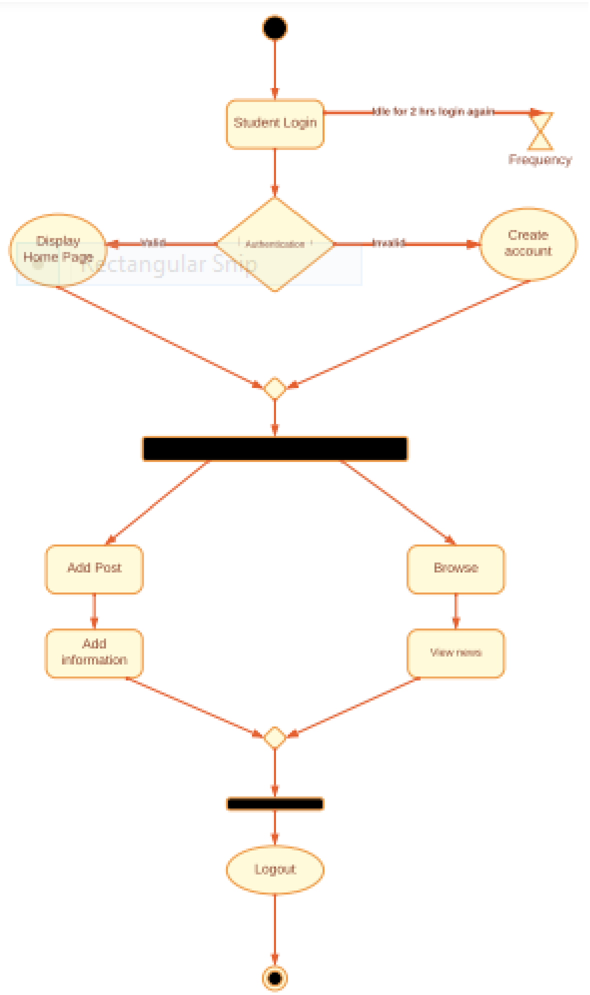
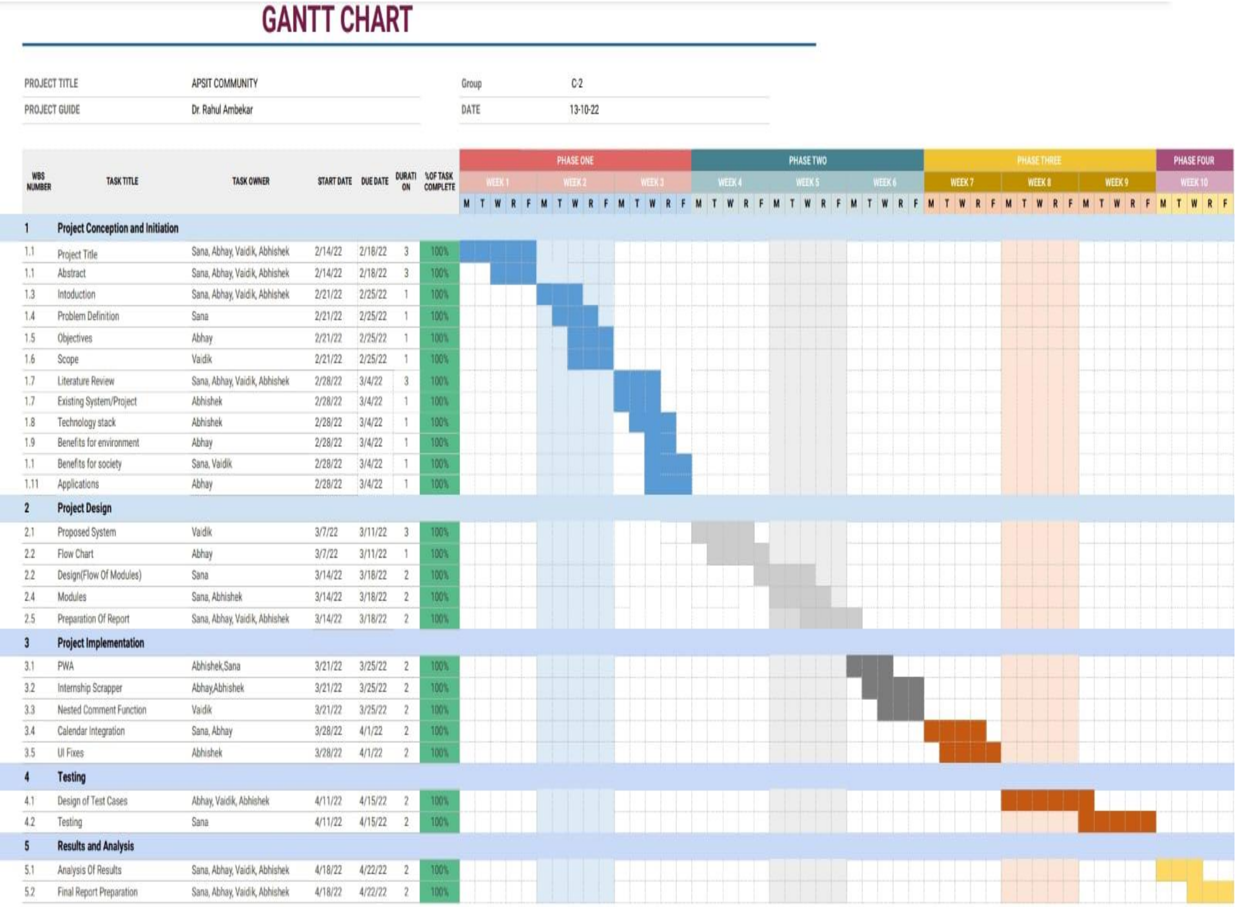


Figure 4.6 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

Chapter 5

Project Planning



GANTT CHART

PROJECT TITLE	APSIT COMMUNITY	Group	C-2
PROJECT GUIDE	Dr. Rahul Ambekar	DATE	13-10-22

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	%OF TASK COMPLETE
1	**Project Conception and Initiation**					
1.1	Project Title	Sana, Abhay, Vaidik, Abhishek	2/14/22	2/18/22	3	100%
1.1	Abstract	Sana, Abhay, Vaidik, Abhishek	2/14/22	2/18/22	3	100%
1.3	Intoduction	Sana, Abhay, Vaidik, Abhishek	2/21/22	2/25/22	1	100%
1.4	Problem Definition	Sana	2/21/22	2/25/22	1	100%
1.5	Objectives	Abhay	2/21/22	2/25/22	1	100%
1.6	Scope	Vaidik	2/21/22	2/25/22	1	100%
1.7	Literature Review	Sana, Abhay, Vaidik, Abhishek	2/28/22	3/4/22	3	100%
1.7	Existing System/Project	Abhishek	2/28/22	3/4/22	1	100%
1.8	Technology stack	Abhishek	2/28/22	3/4/22	1	100%
1.9	Benefits for environment	Abhay	2/28/22	3/4/22	1	100%
1.1	Benefits for society	Sana, Vaidik	2/28/22	3/4/22	1	100%
1.11	Applications	Abhay	2/28/22	3/4/22	1	100%
2	**Project Design**					
2.1	Proposed System	Vaidik	3/7/22	3/11/22	3	100%
2.2	Flow Chart	Abhay	3/7/22	3/11/22	1	100%
2.2	Design(Flow Of Modules)	Sana	3/14/22	3/18/22	2	100%
2.4	Modules	Sana, Abhishek	3/14/22	3/18/22	2	100%
2.5	Preparation Of Report	Sana, Abhay, Vaidik, Abhishek	3/14/22	3/18/22	2	100%
3	**Project Implementation**					
3.1	PWA	Abhishek,Sana	3/21/22	3/25/22	2	100%
3.2	Internship Scrapper	Abhay,Abhishek	3/21/22	3/25/22	2	100%
3.3	Nested Comment Function	Vaidik	3/21/22	3/25/22	2	100%
3.4	Calendar Integration	Sana, Abhay	3/28/22	4/1/22	2	100%
3.5	UI Fixes	Abhishek	3/28/22	4/1/22	2	100%
4	**Testing**					
4.1	Design of Test Cases	Abhay, Vaidik, Abhishek	4/11/22	4/15/22	2	100%
4.2	Testing	Sana	4/11/22	4/15/22	2	100%
5	**Results and Analysis**					
5.1	Analysis Of Results	Sana, Abhay, Vaidik, Abhishek	4/18/22	4/22/22	2	100%
5.2	Final Report Preparation	Sana, Abhay, Vaidik, Abhishek	4/18/22	4/22/22	2	100%

Figure 5.1 Gantt Chart

Chapter 6

Experimental Setup

- **Software Requirements**

1) Nodejs. v.14 +
2) Python v. 3.9+
3) Any modern browser

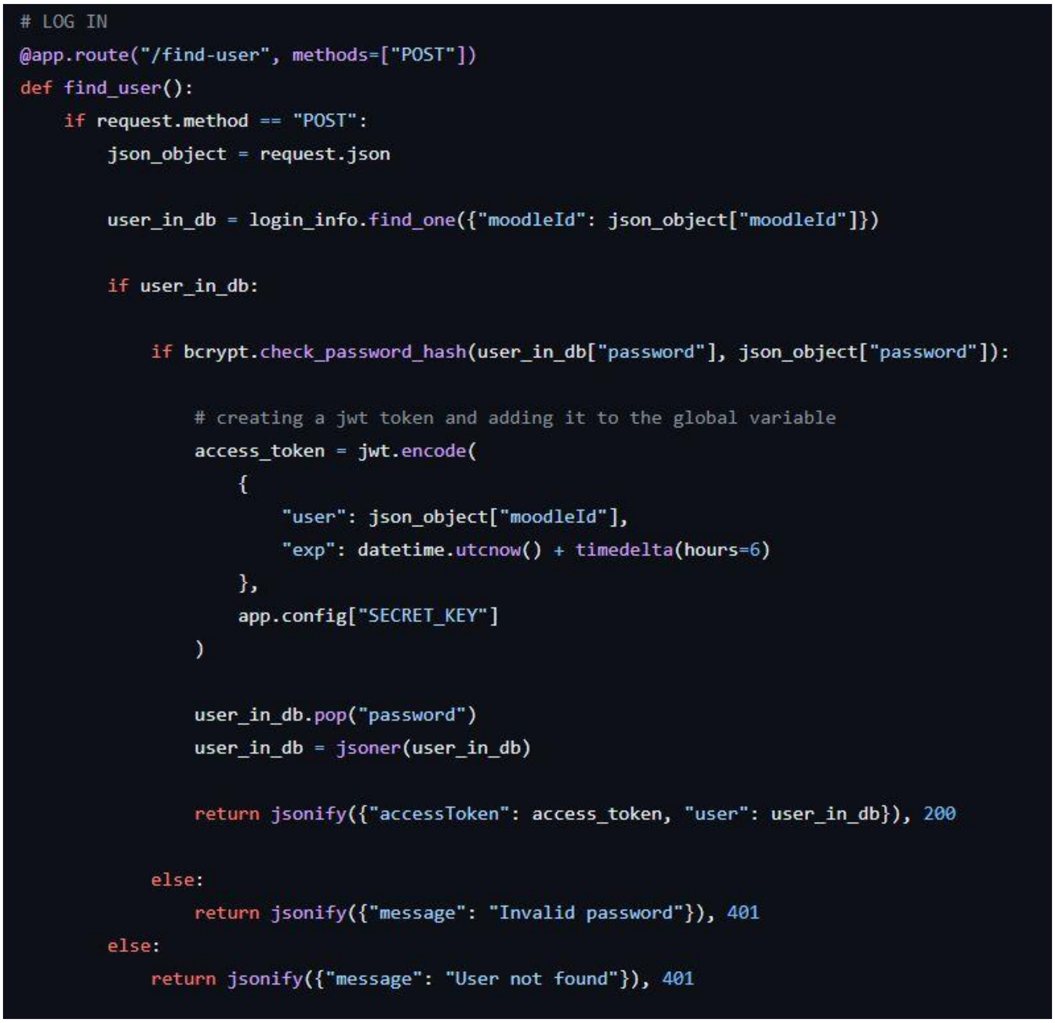
- **Hardware Requirements**

1) CPU: I3 Processor
2) RAM: 4 GB
3) STORAGE: 20 GB
4) OS: Windows
5) Internet Connection

Chapter 7

Implementation Details

Module 1: Secure Login and User Authentication



```
# LOG IN
@app.route("/find-user", methods=["POST"])
def find_user():
    if request.method == "POST":
        json_object = request.json

        user_in_db = login_info.find_one({"moodleId": json_object["moodleId"]})

        if user_in_db:

            if bcrypt.check_password_hash(user_in_db["password"], json_object["password"]):

                # creating a jwt token and adding it to the global variable
                access_token = jwt.encode(
                    {
                        "user": json_object["moodleId"],
                        "exp": datetime.utcnow() + timedelta(hours=6)
                    },
                    app.config["SECRET_KEY"]
                )

                user_in_db.pop("password")
                user_in_db = jsoner(user_in_db)

                return jsonify({"accessToken": access_token, "user": user_in_db}), 200

            else:
                return jsonify({"message": "Invalid password"}), 401
        else:
            return jsonify({"message": "User not found"}), 401
```

Figure 7.1 Login

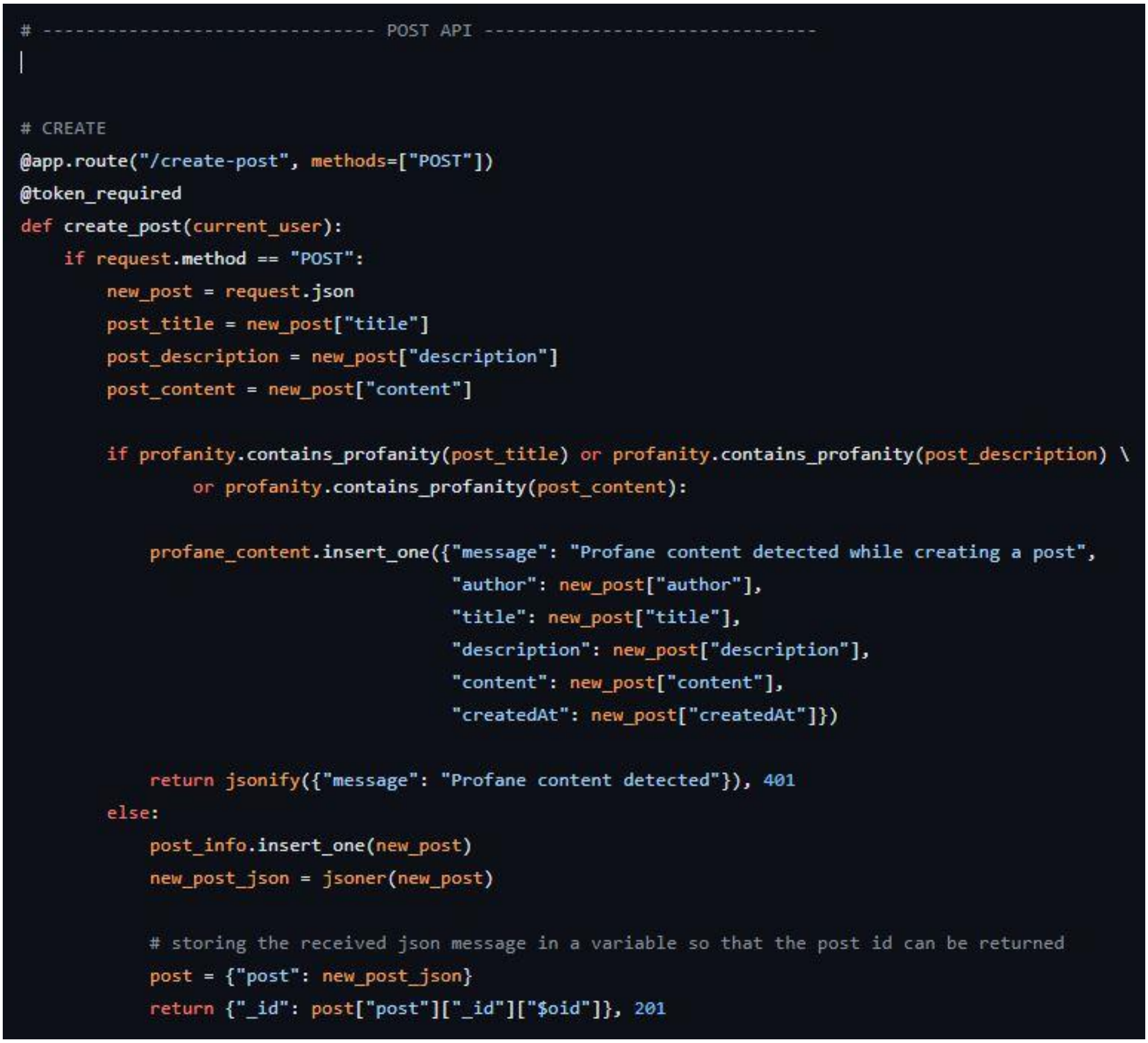
The secure login and user authentication features in the APSIT Community app provide a safe and reliable way for users to access our platform. These features ensure that only registered users with valid credentials can access the app, protecting user data and preventing unauthorized access.

To implement this feature, we require users to register using their college email and roll number. This provides an added layer of security, as it ensures that only verified students are able to access the app. Additionally, we hash the user's password before storing it in our database, providing further protection against unauthorized access.

To streamline the login process, we also use JSON Web Tokens (JWT). These tokens are generated upon successful login and stored on the user's device. This means that users don't need to log in again and again, as the token allows them to access the app without entering their credentials each time.

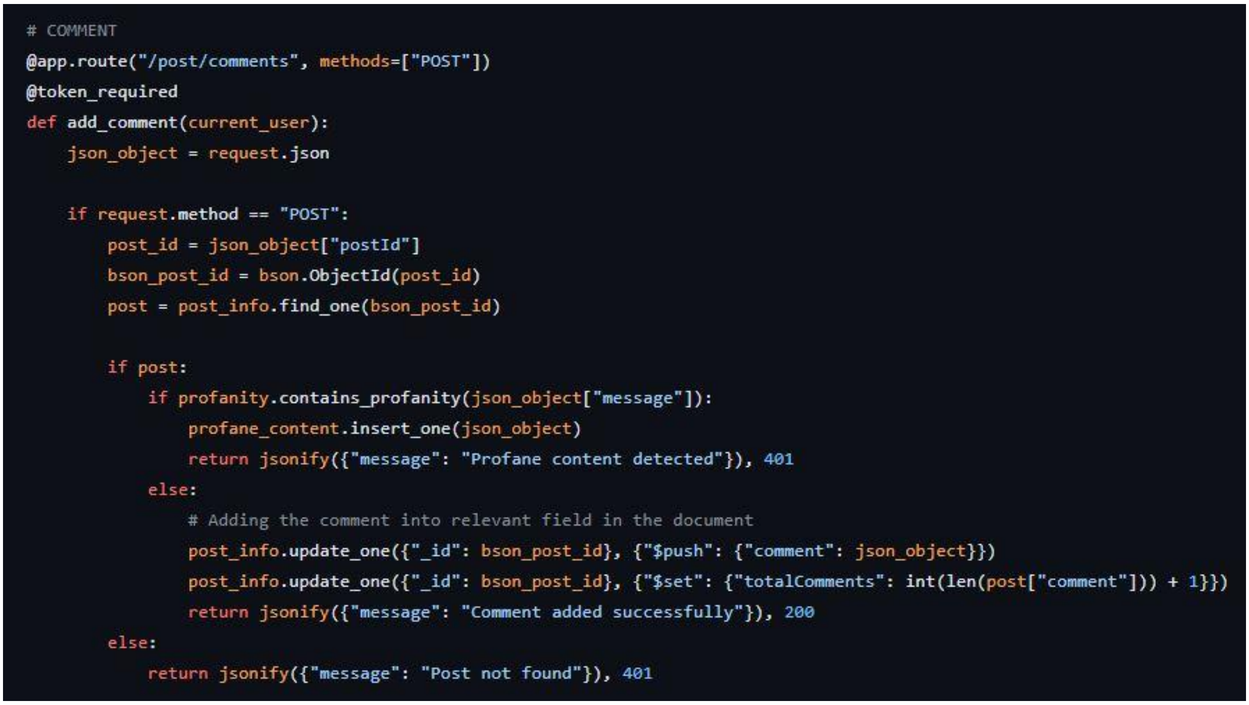
Overall, the secure login and user authentication features in the APSIT Community app provide a secure and reliable way for users to access our platform. By requiring registration with a valid college email and roll number, hashing passwords, and using JWT tokens, we are able to protect user data and prevent unauthorized access, creating a safe and trustworthy online community.

Module 2: User Posts and Commenting



```
# ------------------------------ POST API ------------------------------


# CREATE
@app.route("/create-post", methods=["POST"])
@token_required
def create_post(current_user):
    if request.method == "POST":
        new_post = request.json
        post_title = new_post["title"]
        post_description = new_post["description"]
        post_content = new_post["content"]

        if profanity.contains_profanity(post_title) or profanity.contains_profanity(post_description) \
                or profanity.contains_profanity(post_content):

            profane_content.insert_one({"message": "Profane content detected while creating a post",
                                        "author": new_post["author"],
                                        "title": new_post["title"],
                                        "description": new_post["description"],
                                        "content": new_post["content"],
                                        "createdAt": new_post["createdAt"]})

            return jsonify({"message": "Profane content detected"}), 401
        else:
            post_info.insert_one(new_post)
            new_post_json = jsoner(new_post)

            # storing the received json message in a variable so that the post id can be returned
            post = {"post": new_post_json}
            return {"_id": post["post"]["_id"]["$oid"]}, 201
```

Figure 7.2 Post



```
# COMMENT
@app.route("/post/comments", methods=["POST"])
@token_required
def add_comment(current_user):
    json_object = request.json

    if request.method == "POST":
        post_id = json_object["postId"]
        bson_post_id = bson.ObjectId(post_id)
        post = post_info.find_one(bson_post_id)

        if post:
            if profanity.contains_profanity(json_object["message"]):
                profane_content.insert_one(json_object)
                return jsonify({"message": "Profane content detected"}), 401
            else:
                # Adding the comment into relevant field in the document
                post_info.update_one({"_id": bson_post_id}, {"$push": {"comment": json_object}})
                post_info.update_one({"_id": bson_post_id}, {"$set": {"totalComments": int(len(post["comment"])) + 1}})
                return jsonify({"message": "Comment added successfully"}), 200
        else:
            return jsonify({"message": "Post not found"}), 401
```

Figure 7.3 Comment

The user posts and commenting features in the APSIT Community app provide a powerful tool for sharing information and engaging with other users. These features allow students to post custom content with images, ask questions, and share knowledge, creating a vibrant and dynamic online community.

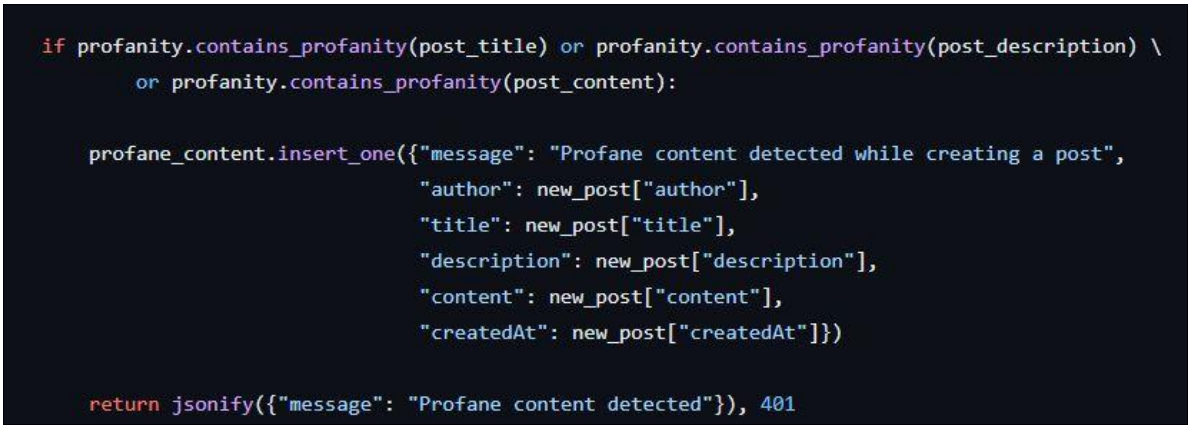
To implement this feature, we created a custom post form that allows users to create a new post with a custom heading and content. Users can also attach images to their post, providing additional context and visual interest.

To give users greater control over their posts, we added editing and deleting functionality. This means that if a user decides they no longer want a post to be visible, they can easily remove it from the platform.

To facilitate discussion and engagement, we also added a commenting feature. This allows other users to add their thoughts and insights to a post, creating a dynamic and interactive community. Additionally, we added a report button to ensure that any inappropriate content or behavior can be flagged for review by our team of moderators.

Overall, the user posts and commenting feature in the APSIT Community app provides a powerful tool for sharing information and engaging with other users. By allowing custom content creation, image attachment, editing and deleting functionality, commenting, and a report button, we are able to create a vibrant and dynamic online community where users can share knowledge and engage in thoughtful discussion.

Module 3: Profanity Detection and Blocking



```
if profanity.contains_profanity(post_title) or profanity.contains_profanity(post_description) \
        or profanity.contains_profanity(post_content):

    profane_content.insert_one({"message": "Profane content detected while creating a post",
                                "author": new_post["author"],
                                "title": new_post["title"],
                                "description": new_post["description"],
                                "content": new_post["content"],
                                "createdAt": new_post["createdAt"]})

    return jsonify({"message": "Profane content detected"}), 401
```

Figure 7.4 Profanity

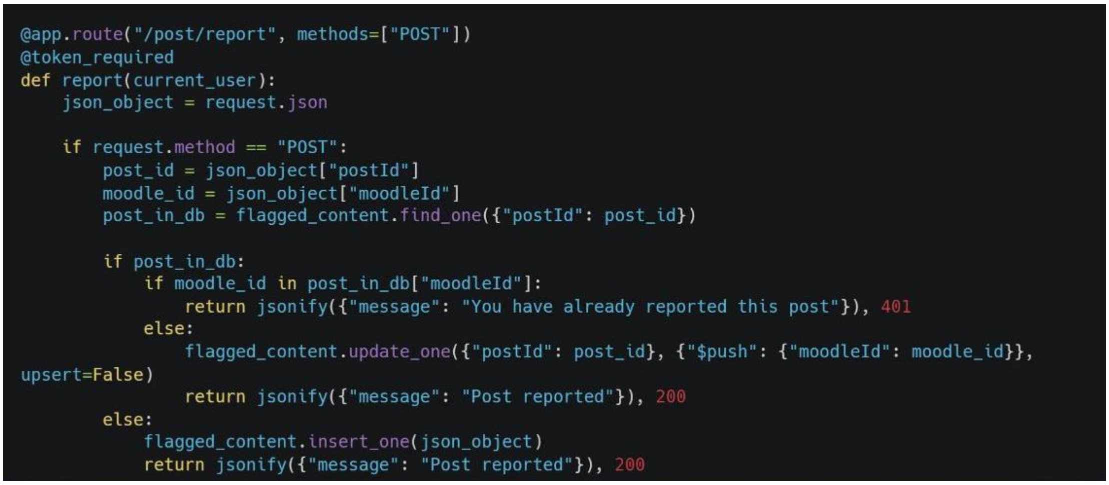
The profanity detection and blocking feature in the APSIT Community app is an important tool for creating a safe and respectful online environment. This feature automatically detects and filters out offensive language and abusive behavior, ensuring that all user-generated content is appropriate and respectful.

To implement this feature, we used the better_profanity library. This library is a powerful tool that can automatically detect and filter out a wide range of profanity and offensive language. By integrating this library into our app, we were able to ensure that users cannot post inappropriate content, thereby promoting a more positive and respectful online community.

However, it is important to note that profanity detection is not a foolproof solution. While the library can identify and block certain words, it may not be able to detect more nuanced forms of offensive language or hate speech. To address this, we also implemented a report button that users can use to flag inappropriate content or behavior. This allows our team of moderators to review and address reports in a timely manner, further promoting a safe and inclusive online community.

Overall, the profanity detection and blocking feature is a key component of our APSIT Community app. By ensuring that all user-generated content is appropriate and respectful, we are promoting a more positive and welcoming environment for all users.

Module 4: Report Feature



```
@app.route("/post/report", methods=["POST"])
@token_required
def report(current_user):
    json_object = request.json

    if request.method == "POST":
        post_id = json_object["postId"]
        moodle_id = json_object["moodleId"]
        post_in_db = flagged_content.find_one({"postId": post_id})

        if post_in_db:
            if moodle_id in post_in_db["moodleId"]:
                return jsonify({"message": "You have already reported this post"}), 401
            else:
                flagged_content.update_one({"postId": post_id}, {"$push": {"moodleId": moodle_id}},
upsert=False)
                return jsonify({"message": "Post reported"}), 200
        else:
            flagged_content.insert_one(json_object)
            return jsonify({"message": "Post reported"}), 200
```

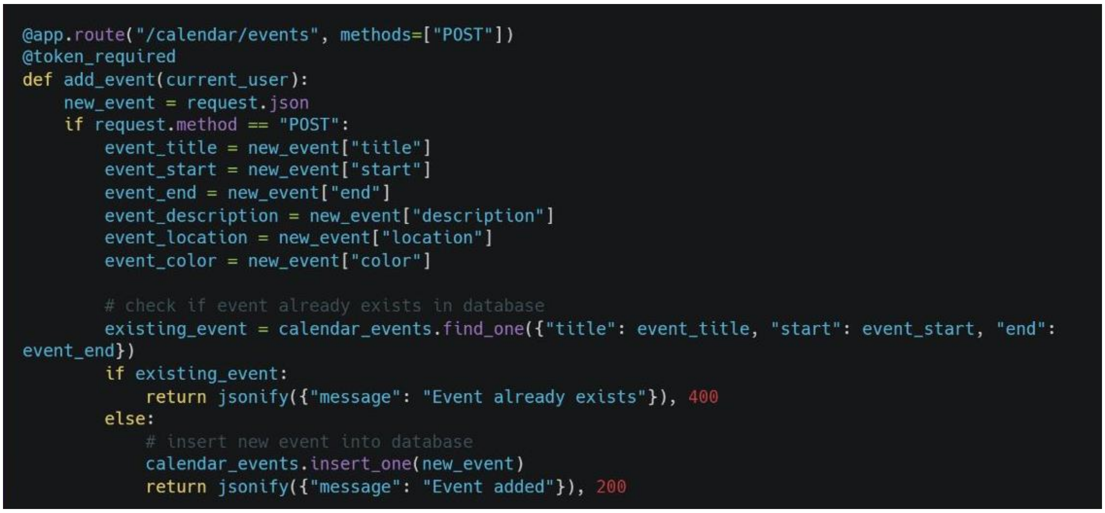
Figure 7.5 Report

The "Post and comment report" feature is implemented using the Flask framework with Python. The feature allows users to report inappropriate content in posts or comments. The endpoint for this feature is "/post/report" and the request method is "POST".

When a request is received, the function extracts the JSON data from the request object. It then checks if the request method is "POST". If it is, the function gets the post ID and moodle ID from the JSON data. It then checks if the post ID exists in the "flagged_content" collection in the database. If it exists, the function checks if the moodle ID of the user who made the report is already in the "moodleId" array of the post in the database. If the moodle ID exists in the array, the function returns a response indicating that the user has already reported the post. Otherwise, the function updates the "moodleId" array for the post in the database with the new moodle ID and returns a response indicating that the post has been reported.

If the post ID does not exist in the "flagged_content" collection, the function adds the JSON data to the collection as a new document and returns a response indicating that the post has been reported.

Module 5: Calendar and Events



```
@app.route("/calendar/events", methods=["POST"])
@token_required
def add_event(current_user):
    new_event = request.json
    if request.method == "POST":
        event_title = new_event["title"]
        event_start = new_event["start"]
        event_end = new_event["end"]
        event_description = new_event["description"]
        event_location = new_event["location"]
        event_color = new_event["color"]

        # check if event already exists in database
        existing_event = calendar_events.find_one({"title": event_title, "start": event_start, "end":
event_end})
        if existing_event:
            return jsonify({"message": "Event already exists"}), 400
        else:
            # insert new event into database
            calendar_events.insert_one(new_event)
            return jsonify({"message": "Event added"}), 200
```

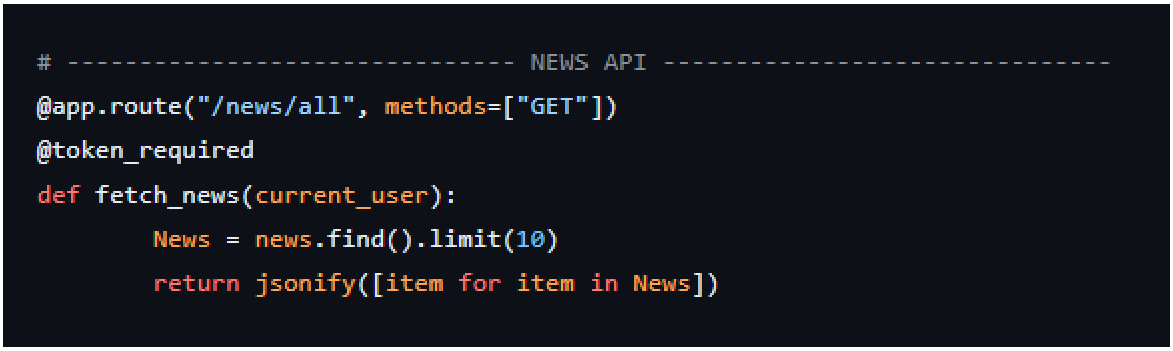
Figure 7.6 Calendar and Events

This code handles the addition of new events to a custom calendar. The function add_event is accessed through a POST request to the "/calendar/events" endpoint. It first extracts the JSON data from the request and stores it in the new_event variable. Then it extracts relevant event data such as the event title, start and end times, description, location, and color.

Next, the code checks if the event already exists in the database by querying the calendar_events collection for any events with the same title, start time, and end time as the new event. If such an event is found, the function returns a JSON response with a "Event already exists" message and 400 status code.

If the event does not exist in the database, the code inserts the new event into the calendar_events collection and returns a JSON response with a "Event added" message and 200 status code.

Module 6: News API



```
# ------------------------------- NEWS API -------------------------------
@app.route("/news/all", methods=["GET"])
@token_required
def fetch_news(current_user):
        News = news.find().limit(10)
        return jsonify([item for item in News])
```

Figure 7.7 News

This code defines a route in a Flask-based REST API that returns the latest 10 news items from a MongoDB database

The @token_required decorator is used to protect the route from unauthorized access. This decorator ensures that a valid access token is present in the request headers, which is used to authenticate the user.

The fetch_news function is defined as the handler for the route. It takes current_user as an argument, which is the user object of the authenticated user.

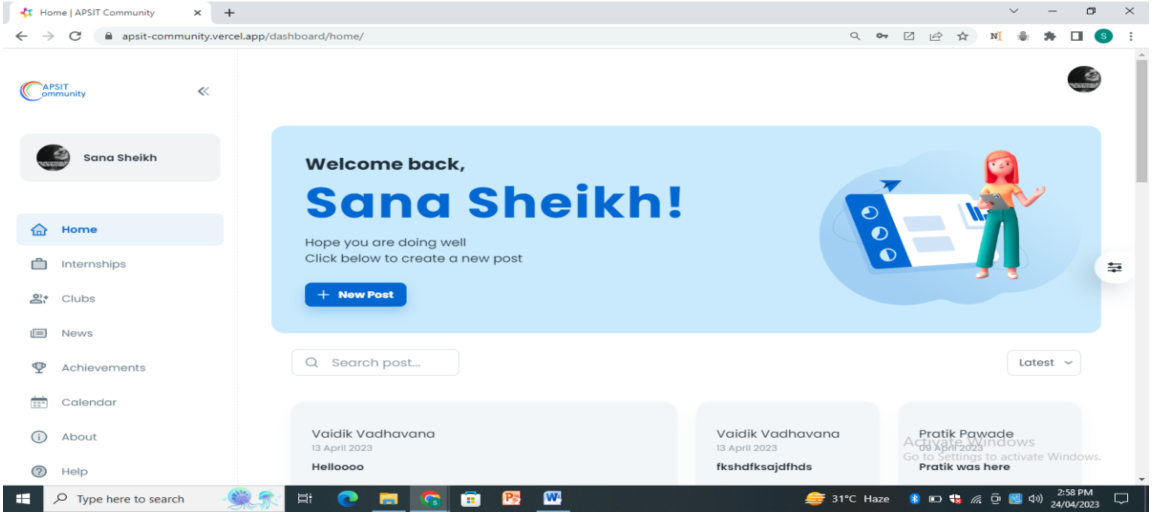
Inside the function, the news collection is accessed using the find() method. The limit(10) method limits the number of documents returned by the query to 10. The results are then converted to a list using a list comprehension and returned as a JSON response using the jsonify method.

In summary, this route fetches the latest 10 news items from the MongoDB news collection and returns them as a JSON response to authorized users who have a valid access token.

Chapter 8

Result

Landing Page:



Welcome back,

Sana Sheikh!

Hope you are doing well
Click below to create a new post

Vaidik Vadhavana
13 April 2023
Helloooo

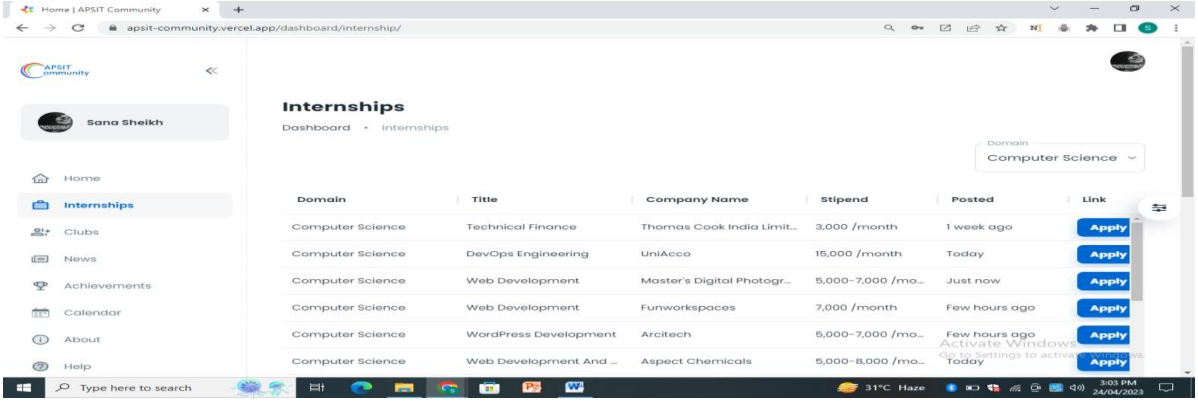
Vaidik Vadhavana
13 April 2023
fkshdfksajdfhds

Pratik Pawade
09 April 2023
Pratik was here

Figure 8.1 Landing Page

This is the landing page of APSIT Community. The students can create a post, like, comment and bookmark the posts.

Internship Page:



Internships

Dashboard • Internships

Domain: Computer Science

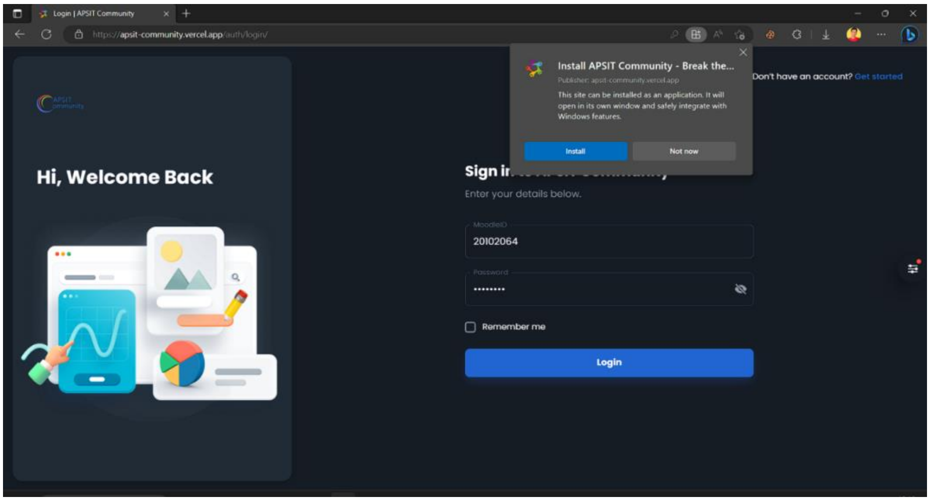
Domain	Title	Company Name	Stipend	Posted	Link
Computer Science	Technical Finance	Thomas Cook India Limit...	3,000 /month	1 week ago	Apply
Computer Science	DevOps Engineering	UniAcco	15,000 /month	Today	Apply
Computer Science	Web Development	Master's Digital Photogr...	5,000-7,000 /mo...	Just now	Apply
Computer Science	Web Development	Funworkspaces	7,000 /month	Few hours ago	Apply
Computer Science	WordPress Development	Arcitech	5,000-7,000 /mo...	Few hours ago	Apply
Computer Science	Web Development And ...	Aspect Chemicals	5,000-8,000 /mo...	Today	Apply

Figure 8.2 Internship Page

Internship page represents all real-time internship scrapped from internshala.

PWA:

Desktop view



Login | APSIT Community

https://apsit-community.vercel.app/auth/login/

Install APSIT Community - Break the...

Publisher: apsit-community.vercel.app

This site can be installed as an application. It will open in its own window and safely integrate with Windows features.

Hi, Welcome Back

Don't have an account? Get started

Sign in

Enter your details below.

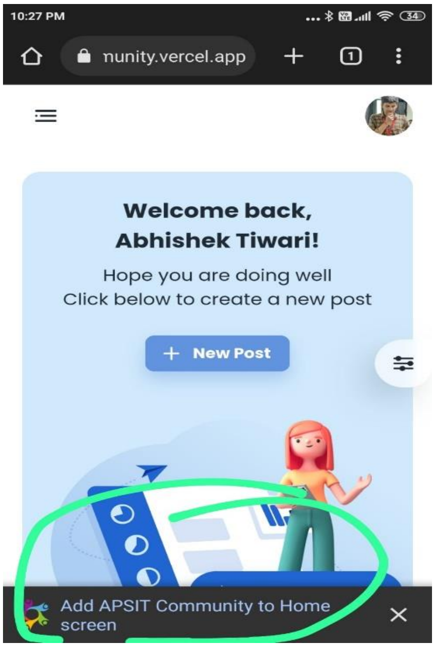
MoodleID: 20102064

Password: ••••••••

□ Remember me

Figure 8.3 PWA desktop view

Phone view



Welcome back, Abhishek Tiwari!

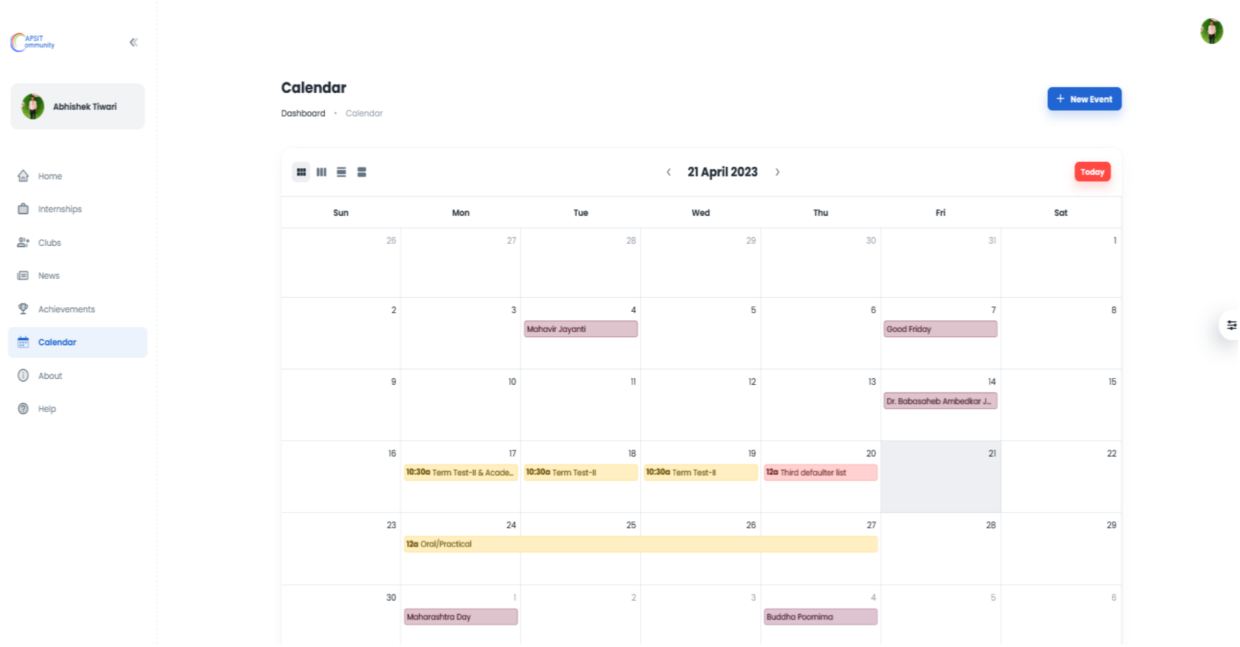
Hope you are doing well
Click below to create a new post

Add APSIT Community to Home screen

Figure 8.3 PWA phone view

PWA stands for progressive web application.

Calendar



Calendar

Dashboard • Calendar

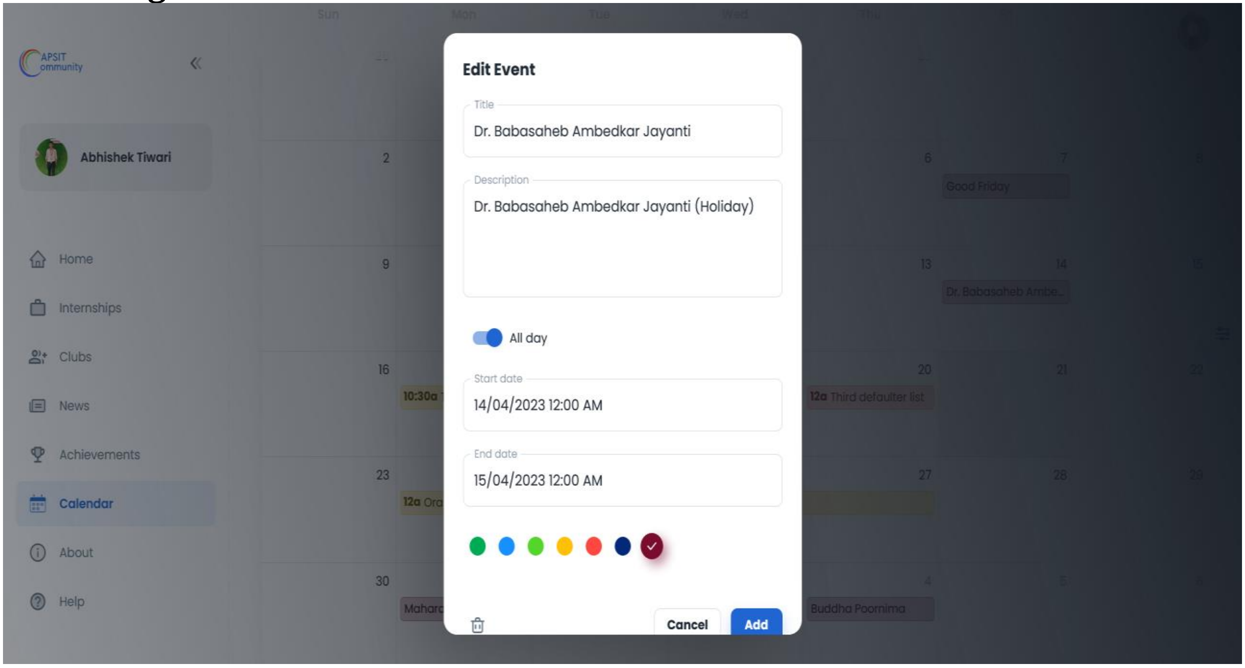
21 April 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4 Mahavir Jayanti	5	6	7 Good Friday	8
9	10	11	12	13	14 Dr. Babasaheb Ambedkar J...	15
16	17 10:30a Term Test-II & Acade...	18 10:30a Term Test-II	19 10:30a Term Test-II	20 12a Third defaulter list	21	22
23	24 12a Oral/Practical	25	26	27	28	29
30	1 Maharashtra Day	2	3	4 Buddha Poornima	5	6

Figure 8.4 Calendar

Calendar page is integrated with academic calendar.

Event Page:



Edit Event

Title: Dr. Babasaheb Ambedkar Jayanti

Description: Dr. Babasaheb Ambedkar Jayanti (Holiday)

All day

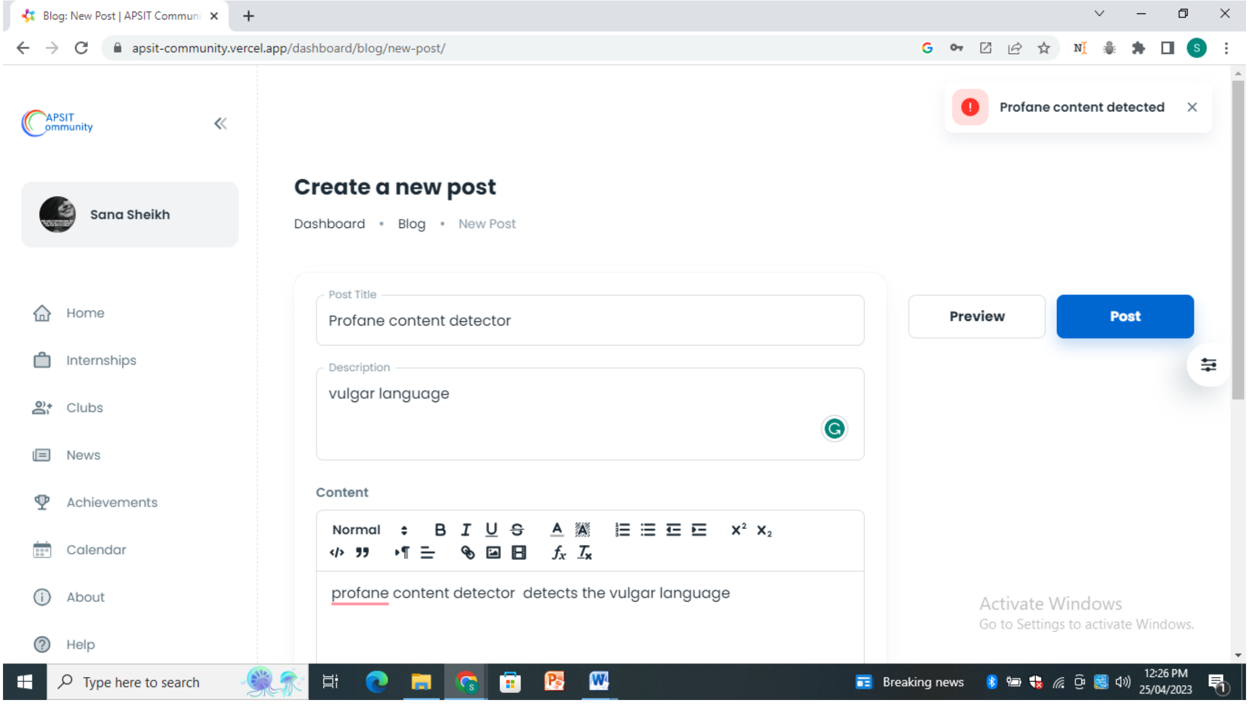
Start date: 14/04/2023 12:00 AM

End date: 15/04/2023 12:00 AM

Figure 8.5Event

Users can create new events.

Profane content detector:



Profane content detected

Create a new post

Post Title: Profane content detector

Description: vulgar language

Content: profane content detector detects the vulgar language

Figure 8.6 Profane content detector

Profane content detector detects profane language and the user is not able to post.

Chapter 9

Conclusion

We have successfully proposed ‘The APSIT Community’ a website where the students can come together and build a platform which will benefit everyone. It was designed to make student’s work easier and provide a user-friendly interface. The project’s aim was to fulfill the feature of posting content so that it would be visible to other students to comment, bookmark and like. This platform will not only help the students to post content but in the future implementation it will benefit them by keeping track of all deadlines as well. The main objective of the project was satisfied.

Chapter 10

References

[1] Nike Arnolda, Trena Paulusb. (2010).Using a social networking site for experiential learning: Appropriating, Lurking, modeling and community building. 158 Citations. The Internet. Volume 13, Issue 4. Dec. 2010

[2] Bin Masrek M (2007). Measuring campus portal effectiveness and the contributing factors. Campus Wide Information Systems 24(5). Nov. 2007.

[3] Berezin A(2020). Interdisciplinary integration in engineering education.42 Citations. JEE Vol. 109, Issue 3.

[4] Secreto, Percia V. Pamulaklakin, Rhodora L.(2015).Learners' satisfaction level with online student portal as a support system in a distance e-learning environment. Turkish Online Journal 16(3). July 2015.

[5] Fonseca Cacho, Jorge (2012).Using Discord to Improve Student Communication, Engagement, and Performance.

Acknowledgement

We have great pleasure in presenting the mini project report on **"Apsit Community"**. We take this opportunity to express our sincere thanks towards our guide **Dr. Rahul Ambekar**, Department of Computer Engineering, APSIT Thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of the project.

We thank Prof. Sachin Malave, Head of Department, Computer Engineering, APSIT for his encouragement during the progress meeting and providing guidelines to write this report.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Sana Sheikh (20102081)

Abhay Sharma (20102065)

Abhishek Tiwari (20102064)

Vaidik Vadhavana (20102197)