B.TECH. (CSE)

V Semester

UE21CS341A –Software Engineering

PROJECT REPORT

on

*Project Title*

**COLLEGE MANAGEMENT SYSTEM**

Submitted by :Name:

Gaurav BV-PES2UG21CS175

Charan S Gowda-PES2UG21CS140

Gagan R \_PES2UG21CS172

August – Dec 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BENGALURU – 560100, KARNATAKA, INDIA

|  |  |  |
| --- | --- | --- |
| *Table of Contents* | | |
| **Sl. No** | **Topic** | **Page No.** |
|  | Project Proposal | 3 |
|  | Project Plan | 4 |
|  | Software Requirements Specification | 11 |
|  | Design Diagrams | 25 |
|  | Test Plan, Cases | X |
|  | Screenshots of Test Output | X |

**Project Description:**

Coviteam revolutionizes education by encouraging collaborative learning through study

groups and project teams. With intelligent teammate matching, real-time group

collaboration tools, a vibrant learning community, and a vast resource repository,

Coviteam empowers students to excel academically while fostering a sense of belonging

and skill development. Potential customers include students from various educational

levels, including high school students, undergraduates, and postgraduates, who will use

Coviteam to find study groups, collaborate on projects, and enhance their learning

experience.

The functional features will include:

● Users can create accounts and profiles, including personal details and academic

interests and edit and update their profiles.

● A user can search for study groups or project teams based on their domain, skills, or

department. They can also send connection requests to join the study/project

groups.

● The system provides intelligent matching, suggesting suitable study groups or

teammates.

● Users can create study groups, defining the group's purpose, subject, and goals.

Group creators can invite members and assign roles (e.g., admin, moderator,

member).

● Study groups have discussion boards, file sharing, and real-time chat for

collaboration.

● Users can upload and share study resources within study groups, including notes,

documents, and links.

● Project creators or assigned moderators have access to project management tools.

Project details, progress, and timelines can be tracked within the platform.

● Users receive notifications about study group activity, connection requests.

.

Plan of Work and Project Ownership:

1. User Registration and Profile Management (Gagan R):

2. Teaming Up and Matching (Gagan R):

3. Study Group Creation and Management(Gagan R):

4. Resource Sharing (Gaurav B V):

5. Discussion Forums and Community Interaction (Gaurav B V):

6. Notifications and Updates(Gaurav B V):

7. User Authentication and Authorization (Charan S Gowda):

8. Resource Repository(Charan S Gowda):

9. Project Management (Charan S Gowda):

# **PROJECT PLAN:**

**1.LIFECYCLE MODEL**

We will be using ITERATIVE lifecycle for the execution of our  project in which continuous refinement and incremental

development are essential.

Reasons why we feel that iterative lifecycle is best suited for  our project:

• **Continuous Improvement**: The iterative model

emphasizes continuous improvement through repeated

cycles. This is best suited with our goal of creating a

dynamic and evolving educational platform. Each iteration  allows for the incorporation of user feedback and the

refinement of features.

• **Incremental Development:** Coviteam can be developed in  increments, focusing on specific features or modules in

each iteration. This approach enables us to deliver

functional components of the platform early and receive  feedback.

• **User-Centric Approach:** An iterative model encourages  strong user involvement throughout the development

process. This applies particularly in our project, where user  feedback and needs drive feature enhancements.

4

• **Risk Mitigation:** By dividing the project into manageable  iterations, risks can be identified and addressed early in our  project.

**2.TOOLS**

The following tools will be used throughout the lifecycle of our  project

• Planning Tool:

**Jira** is a versatile project management and issue tracking  tool that can be used for project planning, task assignment,  and Agile project management.

• Design Tools:

**Figma** is a cloud-based design and prototyping tool that  supports collaborative design and wireframing, which is  valuable for creating the user interface of Coviteam.

• Version Control:

**Git** is a distributed version control system, and **GitHub** is a  web-based platform that provides Git repository hosting.  This combination allows for version control, collaboration,  and code review.

• Development Tools:

5

**Visual Studio Code** is a versatile code editor with a wide  range of extensions for different programming languages. • Bug Tracking:

Besides its planning capabilities, **Jira** also provides  robust issue tracking, making it suitable for tracking and  managing bugs throughout the development process.

• Testing Tools:

**Selenium** is an open-source testing framework for web  applications. It's suitable for automated testing of the  Coviteam web-based user interface.

**.3.DELIVERABLES**

o **Reuse Components**:

● Third-Party Libraries and Frameworks:

Justification: Many software projects, including Coviteam,  rely on third-party libraries and frameworks.These

components are considered "reuse" as they are not

developed from scratch but incorporated to provide

specific functionality. For example, the use of a web

framework like Vite+React (front-end) and Go (back-end) can  be considered reuse.

● Database Management System (DBMS):

Justification: The Coviteam project utilizes a DBMS, such  as MySQL 8.0, for data storage and retrieval. The DBMS

6

is a fundamental component for most software applications  and can be categorized as "reuse" because it is an existing  technology used as is.

• Operating Systems:

Justification: The project specifies compatibility with

Linux and Windows Server as operating systems. These  are standard, existing platforms that are not developed but  leveraged for system deployment.

• Web Servers:

Justification: The Coviteam system relies on web servers  like Apache or Nginx. These are mature, existing solutions  that are reused for hosting the application.

• User Authentication Mechanisms:

Justification: User authentication is a crucial component,  and many projects leverage existing solutions for this

purpose. Coviteam may integrate with third-party

authentication providers or use industry-standard

authentication mechanisms like OAuth. These are

considered reuse components.

o **Build Components:**

7

• User Interface (UI):

Justification: The user interface of Coviteam, including the  web-based interface, is a custom design and build component.  It is tailored to the project's unique requirements and user  experience goals.

• Business Logic:

Justification: The business logic of the Coviteam application,  which includes features like project management, task

assignment, and document collaboration, is a custom-built  component. It is developed specifically for the project's

unique functional requirements.

• Project-Specific Code:

Justification: Any code, scripts, or components that are

designed to meet the project's unique requirements and

functionalities are considered "build" components. This

encompasses all the custom code developed exclusively for  Coviteam.

• Custom Reports and Analytics:

8

Justification: Coviteam's reporting and analytics features are  customized for the platform's specific needs. This code is

developed from scratch and is tailored to generate insights for  improving team efficiency.

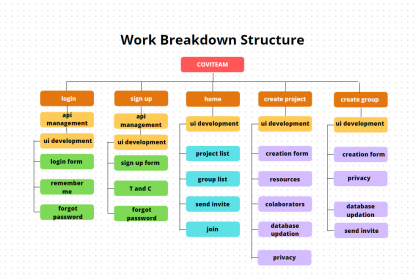
• Unique Documentation:

Justification: While there may be references to existing

educational platforms, the documentation for Coviteam, such  as user guides, tutorials, and project-specific documents, is

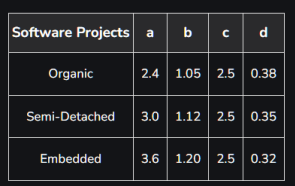
unique and built to cater to the platform's users.

**4.WBS**

****

9

**5.EFFORT ESTIMATION**

****

Using Cocomo model (semi-detached),

a=3.0

b=1.12

c=2.5

d=0.35

������������ = a × (��������) b

�������� = c × (������������) d

KLOC =1

Effort = 3 × (1)1.12

= 3 Person months

Time = 2.5 × (3)0.35

= 3.6 months

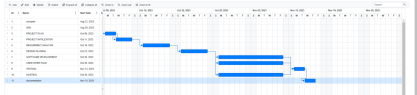
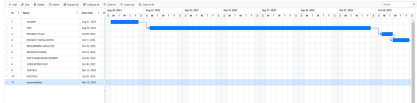
With 3 people in team,

Time taken = 3.6722 /3

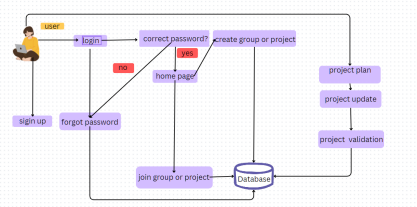
= 1.22 months

10

**6.GANTT CHART**

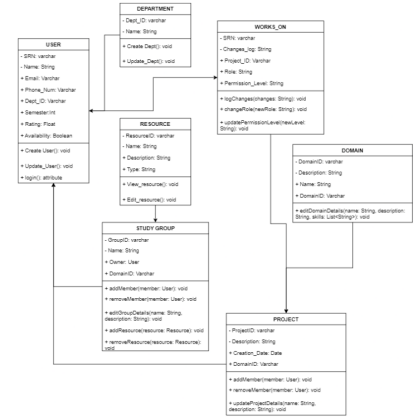
****

**7.ARCHITECTURE DIAGRAM**

****

11

**8.CLASS DIAGRAM**

****\*\*\*\*\*

**SOFTWARE REQUIREMENTS SPECIFICATIONS**

**Table of Contents**

**TableofContents...........................................................................................................................ii Revision History.............................................................................................................................ii 1. Introduction..............................................................................................................................1**

1.1 Purpose..............................................................................................................................................1 1.2 Intended Audience and Reading Suggestions...................................................................................1 1.3 Product Scope...................................................................................................................................1 1.4 References.........................................................................................................................................1

**2. Overall Description..................................................................................................................2** 2.1 Product Perspective...........................................................................................................................2 2.2 Product Functions.............................................................................................................................2 2.3 User Classes and Characteristics......................................................................................................2 2.4 Operating Environment.....................................................................................................................2 2.5 Design and Implementation Constraints...........................................................................................2 2.6 Assumptions and Dependencies.......................................................................................................3

**3. External Interface Requirements...........................................................................................3** 3.1 User Interfaces..................................................................................................................................3 3.2 Software Interfaces...........................................................................................................................3 3.3 Communications Interfaces..............................................................................................................3 **4. Analysis Models**

**5. System Features.......................................................................................................................4** 5.1 System Feature 1...............................................................................................................................4 5.2 System Feature 2 (and so on)............................................................................................................4

**6. Other Nonfunctional Requirements.......................................................................................4** 6.1 Performance Requirements...............................................................................................................4 6.2 Safety Requirements.........................................................................................................................5 6.3 Security Requirements......................................................................................................................5 6.4 Software Quality Attributes..............................................................................................................5 6.5 Business Rules..................................................................................................................................5 **7. Other Requirements................................................................................................................5 Appendix A: Glossary....................................................................................................................5 Appendix B: Field Layouts...........................................................................................................5 Appendix C: Requirement Traceability matrix..........................................................................6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name Date Reason For Changes Version** | | | |
|  |  |  |  |
|  |  |  |  |

Page v / xvi

**PES UNIVERSITY**

BENGALURU

Introduction

Purpose

*This SRS document outlines the functional and non-functional requirements for the Coviteam platform. It encompasses all features and aspects of the software system, covering user registration, teammate matching, study group and project collaboration, resource sharing, user interfaces, security measures, and system performance.*

*Coviteam serves as a collaborative learning and project collaboration platform designed for diverse user groups, including students, educators, professionals, and individual learners. This SRS defines the requirements for the entire system to ensure that it effectively fulfills its intended purpose: facilitating collaborative learning, project management, and community engagement.*

Intended Audience

*This document serves as a comprehensive guide for software development, enabling the development team to gain a thorough understanding of the project's objectives, functionalities, and quality standards. Additionally, it facilitates effective communication among stakeholders, ensuring that the final product aligns with the project's vision and objectives.*

Product Scope

*Coviteam is a collaborative learning and project management platform designed to empower students, educators, professionals, and individual learners. Its purpose is to facilitate collaborative learning, promote project collaboration, and build a vibrant learning community. Coviteam's objectives include academic excellence, skill development, community engagement, and alignment with educational goals.*

References

*We draw inspiration and insights from existing educational platforms like Zoom for Education, Google Classroom, Microsoft Teams for Education, Canvas by Instructure, and Blackboard Learn. These platforms serve as valuable references for understanding the landscape of collaborative learning, virtual classrooms, and educational collaboration. Analyzing their features and user experiences helps us refine and enhance Coviteam's capabilities to meet the diverse needs of our users.*

Overall Description

Product Perspective

*The COVITEAM - Collaborative Virtual Teammate Network is envisioned as a self-contained software solution designed to foster efficient collaboration and teamwork within virtual teams. It does not belong to an existing product family but serves as an independent platform with a specific focus on enhancing remote collaboration. COVITEAM is not intended as a direct replacement for any existing systems but as a complementary tool to facilitate virtual teamwork.*

Product Functions

*1. User Authentication and Management:*

• *Enable users to register, log in, and manage their profiles.*

• *Administer user roles and permissions for secure access control.*

*2. Collaboration Workspace:*

• *Provide a virtual workspace for real-time document collaboration.*

• *Facilitate chat and instant messaging for team communication.*

• *Support video conferencing and screen sharing for meetings.*

*3. Resource Sharing and Storage:*

• *Allow users to upload, access, and manage shared documents and resources.* • *Implement version control for document tracking and history.*

*4. Task and Project Management:*

• *Enable users to create, assign, and manage tasks.*

• *Support project planning, scheduling, and progress tracking.*

*5. Communication Tools:*

• *Offer internal communication channels for team discussions.*

• *Allow integration with external communication tools if desired.*

*6. Reporting and Analytics:*

• *Provide reporting capabilities to monitor user activity and collaboration metrics.* • *Generate insights for improving team efficiency.*

*7. Security and Data Protection:*

• *Ensure data security through encryption and access controls.*

• *Comply with privacy regulations and data protection standards.*

*8. External Integration:*

• *Allow integration with third-party applications and services via APIs.* • *Support data exchange with external systems and tools.*

*9. User Support and Help Center:*

• *Offer user assistance, tutorials, and a help center for guidance.*

• *Provide customer support and feedback mechanisms.*

User Classes and Characteristics

• *Students: Students from various educational levels, including high school, undergraduate, and postgraduate, who use Coviteam to find study groups, collaborate on projects, and enhance their learning experiences.*

• *Tutors and Instructors: Educators who leverage Coviteam to facilitate group learning, provide additional resources, and engage with their students.* • *Study Group Creators and Moderators: Users who take the initiative to create and manage study groups, ensuring a conducive learning environment for participants.* • *Community Contributors: Users who share educational resources, participate in discussions, and actively engage in Coviteam's learning community.*

• *Administrators and Moderators: Individuals responsible for maintaining and moderating the Coviteam platform, ensuring a safe and supportive environment for users*

Operating Environment

*COVITEAM - Collaborative Virtual Teammate Network operates in an environment that encompasses various technologies and components. Below is a description of the operating environment for COVITEAM, highlighting the hardware platform, operating systems, and software components it needs to coexist with:*

• *Hardware Platform:*

*COVITEAM is designed to run on standard computing hardware commonly used by individuals and organizations, including desktop computers, laptops, tablets, and smartphones.*

*It should be compatible with a range of hardware specifications, considering both high-end and lower-end devices to ensure accessibility.*

• *Operating Systems and Versions:*

o *Server-side:*

*The server-side components of COVITEAM, which include the web server and database server, are typically hosted on a variety of operating systems. Common choices include Linux distributions (e.g., Ubuntu, CentOS), Windows Server, or cloud-based platforms like AWS, Azure, or Google Cloud.*

*The specific operating system version may depend on the hosting environment and organizational preferences.*

o *Client-side:*

*The web-based user interface of COVITEAM is designed to be cross-platform and should be compatible with modern web browsers, including but not limited to:*

*Google Chrome*

*Mozilla Firefox*

*Apple Safari*

*Microsoft Edge*

• *Software Components and Technologies:*

o *Programming Languages:*

*Server-side components are implemented using technologies like Go (Golang) for server logic, JavaScript for client-side scripting, HTML for web page structure, and CSS for styling.*

*SQL is used for database management and data retrieval.*

o *Web Server:*

*COVITEAM typically runs on web servers like Apache, Nginx, or cloud-based serverless platforms.*

o *Database:*

*SQL-based relational databases such as MySQL or SQL Server may be used for data storage and retrieval.*

Design and Implementation Constraints

*The project is subject to various constraints that can significantly influence the design and implementation of the system. These constraints encompass budgetary limitations, project timeline considerations, and technological restrictions. They play a pivotal role in shaping project decisions, impacting resource allocation, development approaches, and the overall project trajectory. Managing these constraints effectively is essential to ensure the successful delivery of the system within the defined parameters.*

2.6 Assumptions and Dependencies

*Assumed that third-party components, libraries, and services, such as authentication providers and cloud storage platforms, will be available and functional as required and the organization's data privacy policies and regulatory compliance measures will be followed during the development and deployment of COVITEAM.*

Dependencies

Data Center Infrastructure: The availability and reliability of the data center infrastructure on which COVITEAM is hosted are critical to system uptime and performance.

Release of Third-Party Updates: COVITEAM may rely on third-party software components that receive updates. The project depends on timely and compatible releases from these third-party providers.

External APIs: The project depends on external APIs for services such as document storage and authentication, and any changes or discontinuation of these APIs may impact COVITEAM's functionality.

External Interface Requirements

User Interfaces

• *Web-based User Interface*

• *Interface elements for integrating with external applications and services, including APIs and third-party app connections.*

• *The help center and support interface provide user assistance, tutorials, FAQs, and support request options.*

Software Interfaces

• *Database Management System (DBMS) :*

*MySQL 8.0*

• *Operating System Compatibility:*

*Linux, Windows Server*

• *Front-End Framework:*

*Flutter or React*

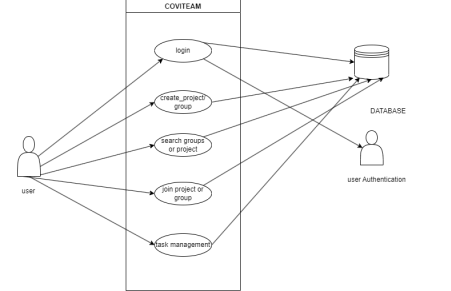
• *Back-End Framework*

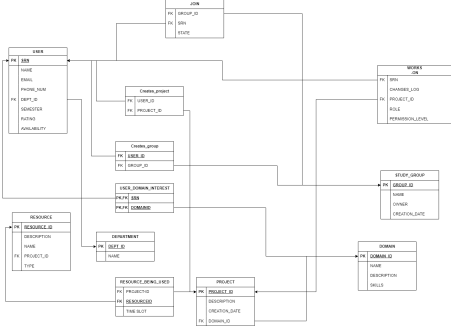
*Golang or nodejs*

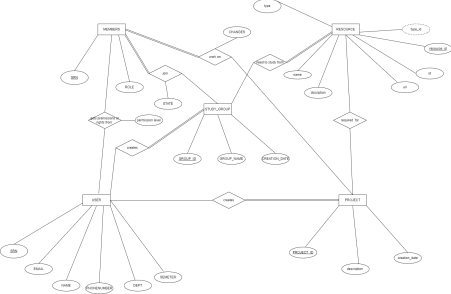
Communications Interfaces

*COVITEAM's communication interfaces are vital for effective collaboration and user interaction. These interfaces include email notifications, web browser access, secure network protocols, real-time chat, electronic forms, and data transfer optimization. Key requirements include adhering to email standards, ensuring browser compatibility, implementing secure communication protocols, enabling real-time chat with encryption, and accommodating various data transfer rates. Additionally, synchronization mechanisms are employed to minimize conflicts and maintain data consistency. These measures collectively enhance user experience, data security, and smooth communication within COVITEAM.*

Analysis Models







System Features

***1. User Management***

*1.1 User Registration*

• *Users should be able to register by providing essential information.* • *The system must validate and store user registration data securely.* • *Confirmation emails should be sent for account verification.*

*1.2 User Authentication*

• *Users must be able to log in securely with their credentials.*

• *Authentication should be protected against unauthorized access attempts.* • *Password reset functionality should be available.*

***2. Project Management***

*2.1 Create and Edit Projects*

• *Users with appropriate permissions should be able to create and edit projects.* • *Projects must have a name, description, and associated team members. 2.2 Project Assignment*

• *Project managers should be able to assign team members to projects.* • *Assignments should include roles and responsibilities.*

*2.3 Project Goals and Objectives*

• *Users should set clear goals and objectives for each project.*

• *Goals should be measurable and trackable.*

***3. Task Management***

*3.1 Create and Assign Tasks*

• *Users should create tasks within projects.*

• *Tasks must include titles, descriptions, due dates, and priorities.*

• *Task assignment to team members should be supported.*

*3.2 Task Progress Tracking*

• *Users and project managers should be able to track task progress.* • *Task completion should be marked, and status should be updated.*

***4. Document Management***

*4.1 Upload and Share Documents*

• *Users should upload and store project-related documents.*

• *Document sharing and access control based on user roles should be available.* • *Version control for documents is required.*

*4.2 Document Collaboration*

• *Users should collaborate on documents in real-time.*

• *Comments and annotations on documents should be supported.*

•

***5. Search and Filtering***

*5.1 Search Functionality*

• *Users should be able to search for tasks, documents, and discussions.* • *Advanced search options and filtering should be provided.*

***6. Resource Management***

*6.1 Resource Allocation*

• *Project managers should be able to allocate resources (e.g., team members, equipment, materials) to specific tasks and projects.*

• *Resource allocation should consider availability, skills, and workload. 6.2 Resource Tracking*

• *The system should provide real-time tracking of resource usage and availability.* • *Resource status (e.g., allocated, available, on leave) should be visible to project managers and team members.*

*6.3 Resource Reports*

• *Generate resource utilization reports to help project managers make informed decisions.*

• *Reports should include resource allocation history, workload distribution, and resource cost analysis.*

Other Non Functional Requirements

Performance Requirements

• *Response Time*

• *Scalability*

• *Data Retrieval Efficiency*

• *Real-Time Collaboration Synchronization*

• *Resource Utilization*

Safety Requirements

• *Data Security and Privacy*

• *User Authentication*

• *System Availability*

• *Security Incident Response*

• *Compliance with Privacy Regulations*

Security Requirements

• *Data Encryption*

• Access Control

• Data Privacy Compliance

• Data Backup and Recovery

• Security Certifications

Software Quality Attributes

*COVITEAM prioritizes key software quality attributes to ensure user satisfaction and facilitate efficient development and maintenance. These attributes encompass usability, reliability, maintainability, portability, security robustness, interoperability, scalability, adaptability, testability, and availability. These attributes guide the development of COVITEAM, ensuring a robust and user-centric platform.*

Business Rules

• *User Authentication*

• *User Roles*

• *Data Access*

• *Document Sharing*

• *Task Assignment*

• *Security and Compliance*

• *Communication*

• *Data Backup*

• *User Conduct*

• *Usage Policies*

Other Requirements

NO other requirments.

**Appendix A: Glossary**

*COVITEAM: Collaborative Virtual Teammate Network - The software product designed to facilitate collaboration and project management among virtual teams.*

*GUI: Graphical User Interface - The visual interface that allows users to interact with and navigate the COVITEAM software.*

*API: Application Programming Interface - A set of rules and protocols that allows different software applications to communicate and interact with each other.*

*DBMS: Database Management System - Software that manages and organizes the storage and retrieval of data in a database.*

**Appendix B: Field Layouts**

Project table

Project table

+------------------+--------------+---------------------------+-----------------+

| Field Name | Data Type | Description | Required |

+------------------+--------------+---------------------------+-----------------+

| Project Name | Text | Name of the project | Yes |

| Project ID | Auto Number | Unique project identifier | Automatically |

| Project Manager | User/Picker | Assigned project manager | Yes |

| Start Date | Date | Project start date | Yes |

| End Date | Date | Project end date | Yes |

| Description | Text Area | Project description | No |

+------------------+--------------+---------------------------+-----------------+

Study group table

+------------------+--------------+---------------------------+-----------------+

| Field Name | Data Type | Description | Required |

+------------------+--------------+---------------------------+-----------------+

| Group Name | Text | Name of the study group | Yes |

| Group ID | Auto Number | Unique group identifier | Automatically |

| Course Name | Text | Name of the course | Yes |

| Instructor | Text | Instructor's name | No |

| Start Date | Date | Start date of the group | Yes |

| End Date | Date | End date of the group | No |

| Description | Text Area | Group description | No |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

USER table

+------------------+--------------+---------------------------+-----------------+

| Field Name | Data Type | Description | Required |

+------------------+--------------+---------------------------+-----------------+

| User ID | Auto Number | Unique user identifier | Automatically |

| Username | Text | User's username | Yes |

| First Name | Text | User's first name | Yes |

| Last Name | Text | User's last name | Yes |

| Email | Email | User's email address | Yes |

| Role | Dropdown | User's role(e.g.,Admin,Moderator,Member)|Yes|

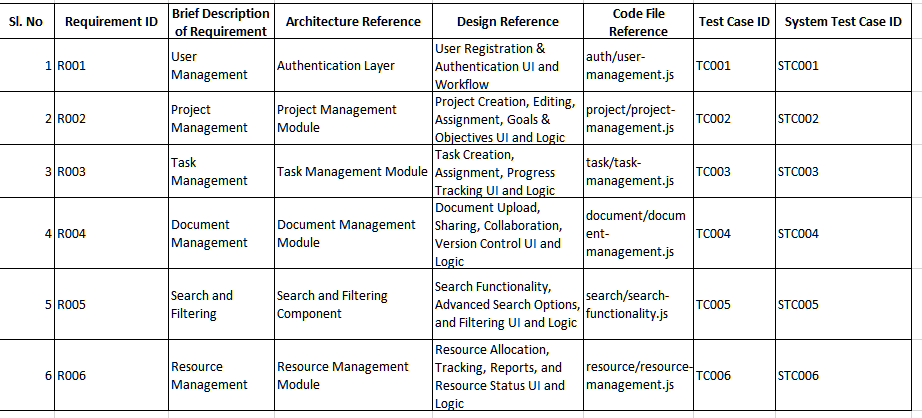
| Status | Dropdown | User's status (e.g., Active,Inactive) | Yes |

| Registration Date| Date | Date of user registration | Yes |

| Profile Picture | Image | User's profile picture | No |

+------------------+--------------+---------------------------+-----------------+

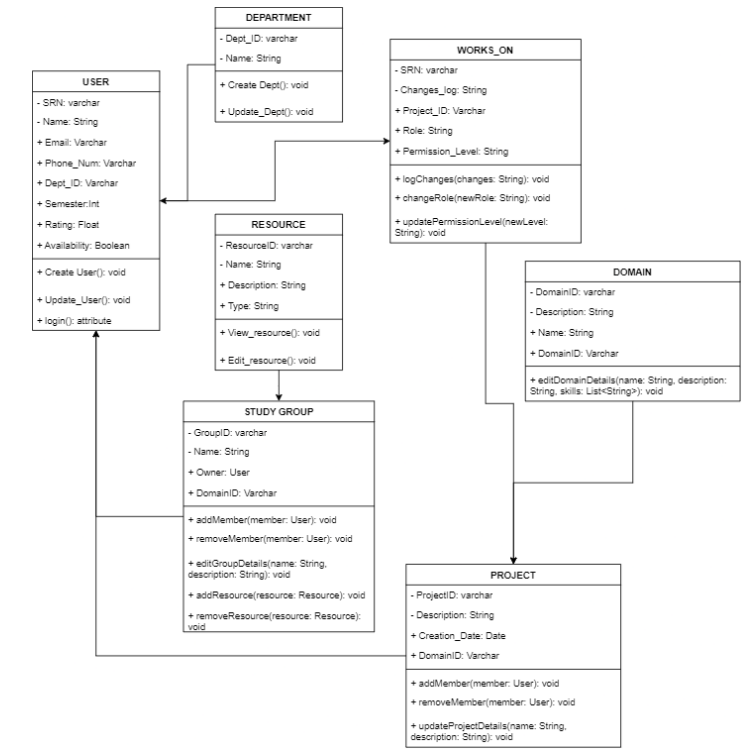
**C: Requirement Traceability Matrix**

****

# **DESIGN DIAGRAM:**

|  |  |
| --- | --- |
| **SL NO.** | **TOPIC** |
| **1.** | **CLASS DIAGRAM** |
| **2.** | **DATA FLOW DIAGRAM (L0)** |
| **3.** | **DATA FLOW DIAGRAM (L1)** |
| **4.** | **ARCHITECTURE DIAGRAM** |
| **5.** | **ARCHITECTURAL STYLE** |

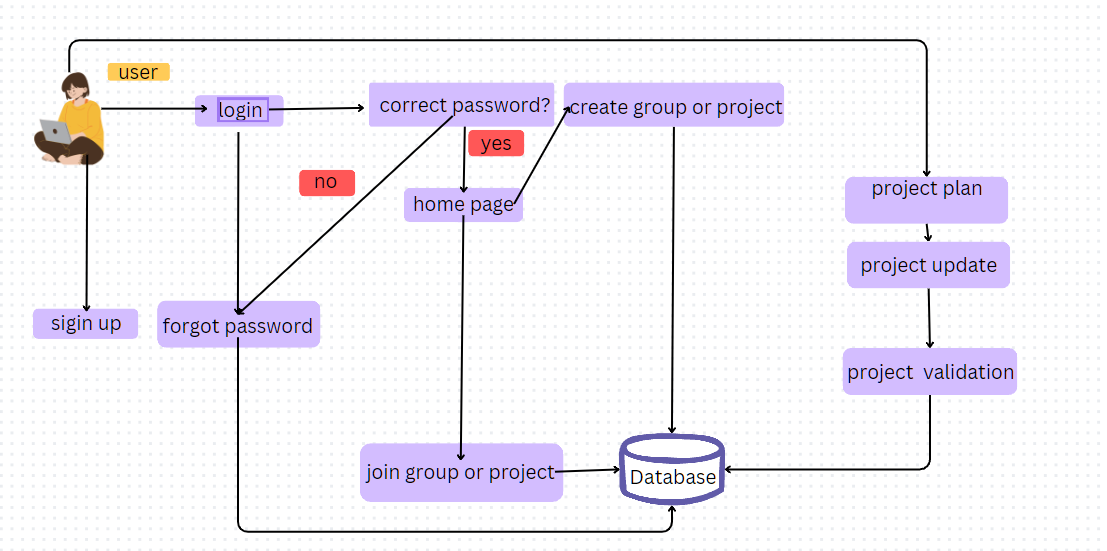
**1.CLASS DIAGRAM :**



**2.DATA FLOW DIAGRAM (L0)**

|  |  |
| --- | --- |
|  | **3.DATA FLOW DIAGRAM (L1)** |
|  |  |

**4.ARCHITECTURE DIAGRAM :**



|  |  |
| --- | --- |
|  |  |

**5.ARCHITECTURE STYLE (client server)**

**Client-Server:** The client–server model is a distributed application structure that partitions tasks or workloads between the providers and service requesters.

**Justification:**

**Scalability:** Client-server architecture allows for horizontal scalability. You can scale the system by adding more servers to handle an increasing number of clients. This makes it suitable for applications that may experience variable or high loads, as you can distribute the load across multiple servers.

**Security Enforcement:** By centralizing authentication and authorization mechanisms on the server, you can control and enforce security policies more effectively.

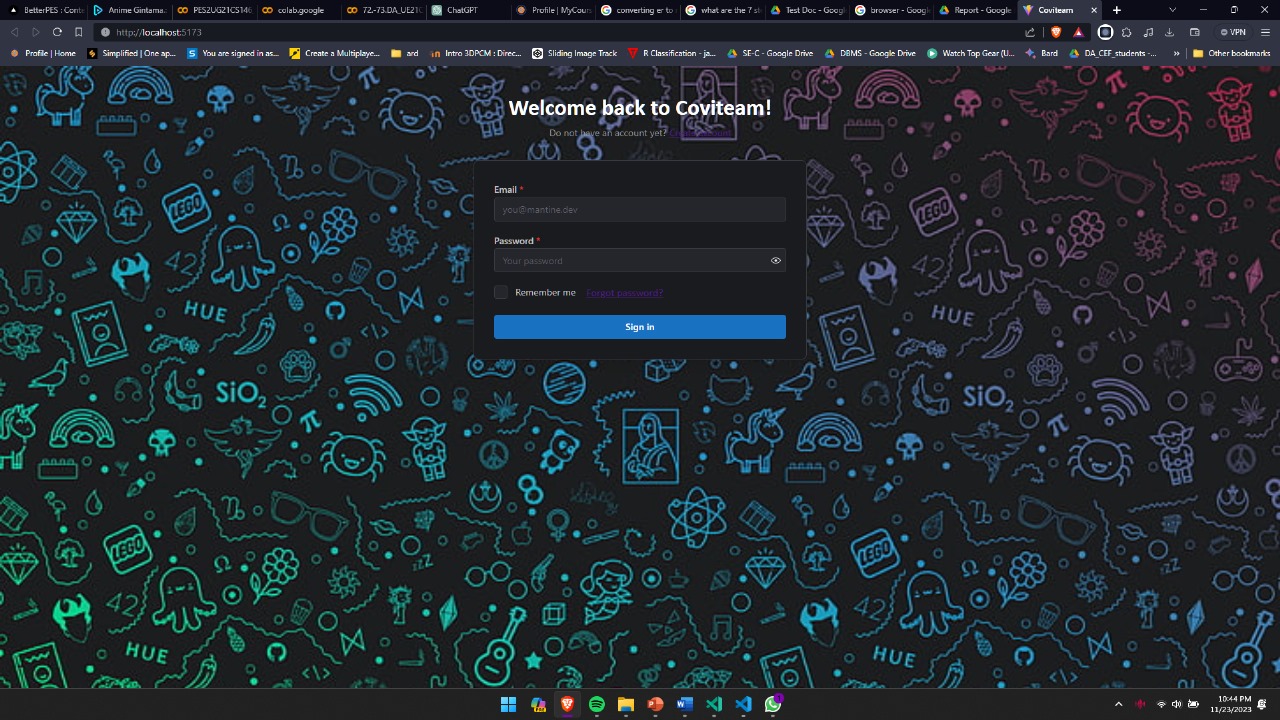
**Centralized Data Management:** Storing data on a central server allows for consistent data management, backup, and recovery. This is crucial for applications that require data integrity and availability.

**Resource Sharing:** Client-server architecture allows multiple clients to share a common set of resources or services provided by the server, making it an effective choice for scenarios where resource sharing is required.

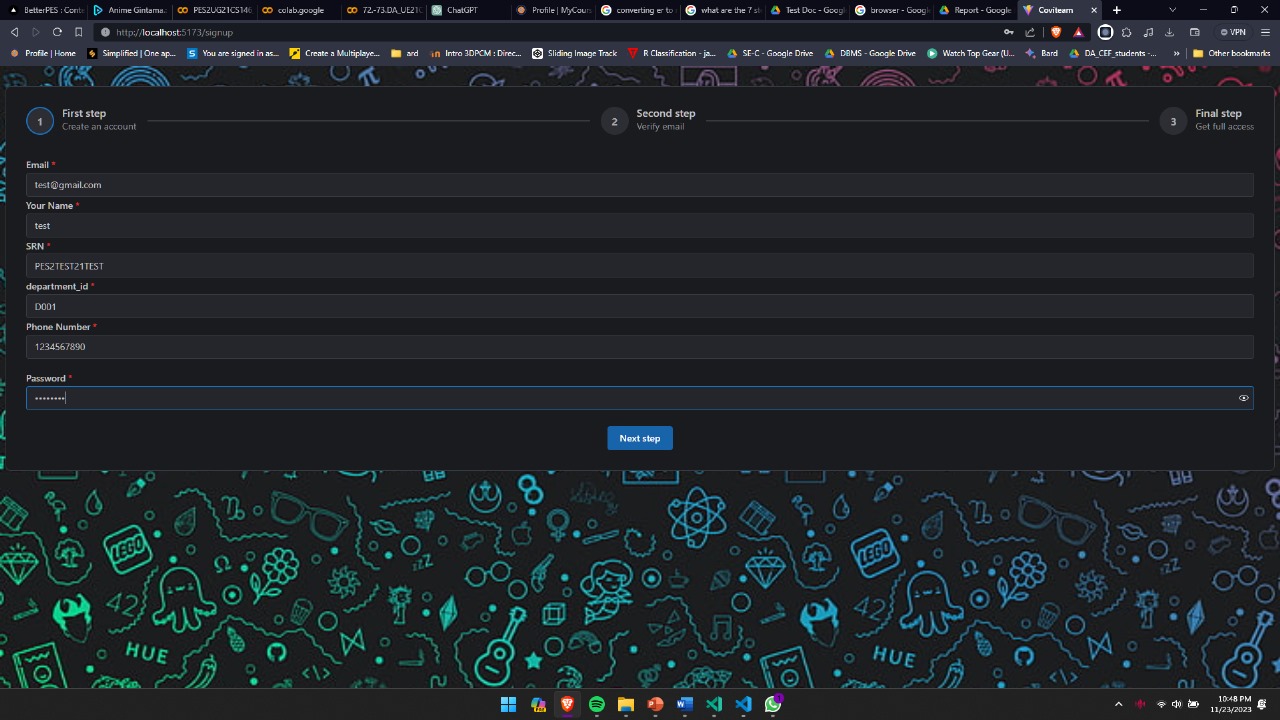
**Test case:**

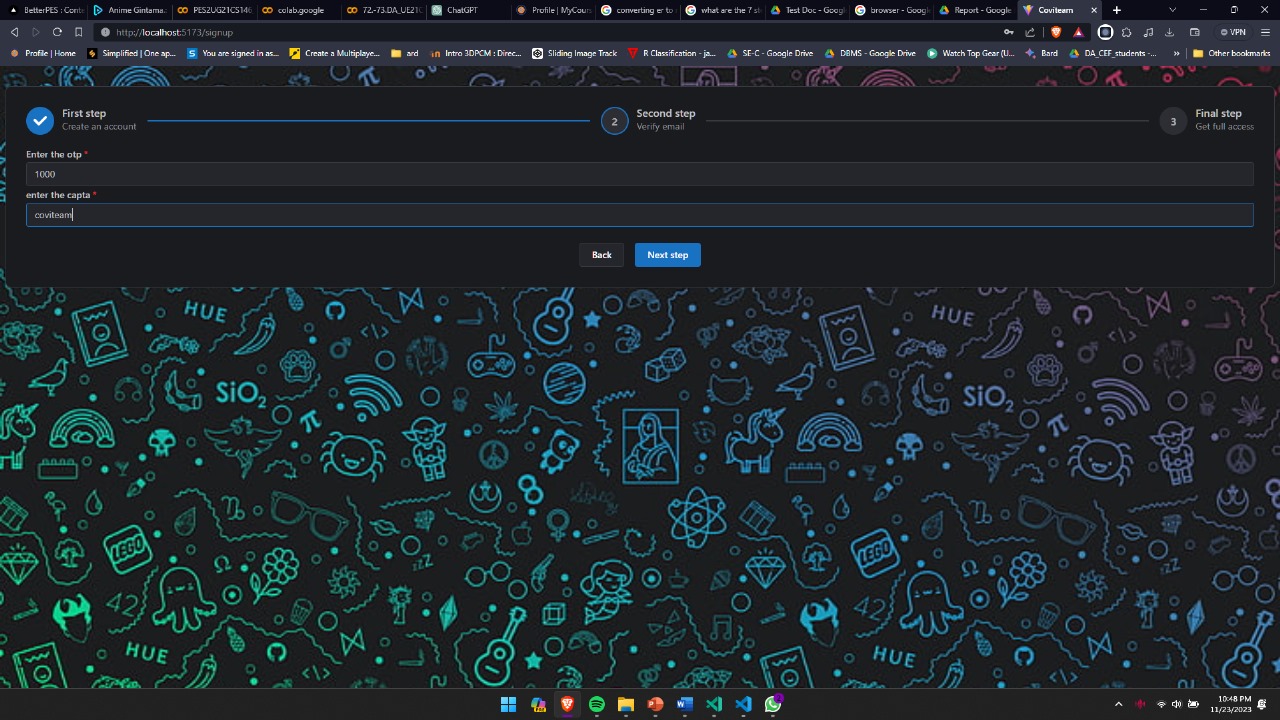
| Test Case ID | Name of Module | Test Case  Description | Pre-conditions | Test Steps | Test Data | Expected Results | Actual Results | Test Result |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UT-01** | **Sign-up** | **To test creating account** | **Need to have email ,phone number and a browser** | **Step1:Fill the form**  **Step 2:**  **Enter capta**  **Step3:**  **Enter your intressts**  **Step4:click finish** | **New credentials** | **Redirected to login page** | **Redirected to ligin page** | **Pass** |
| **UT-02** | **Login** | **To test login functionality** | **User should have signed up** | **Enter email**  **And password**  **Click login in** | **Valid**  **Credentials** | **Redirected to home page** | **Redirected to home** | **Pass** |
| **UT-03** | **Creation of groups** | **To create study groups** | **Logged in as a legit uses** | **Click on the your groups create on (+) button enter group name**  **And description**  **And press create group** | **Group name should not be null** | **Redirected to home page** | **Redirected to home** | **Pass** |
| **UT-04** | **Creation of project** | **To create**  **Project** | **Logged in as a legit user** | **Click on the your projects create on (+) button enter project name, description and select the domain**  **And press create group** | **Project name should not be null**  **Should have a domain** | **Redirected to home** | **Redirected to home** | **Pass** |
| **UT-05** | **Requesting access for project** | **To join project** | **Logged in as a legit user** | **Click on recommended projects press on request access on the desired project** | **NULL** | **Notification sent to owner** | **Notification sent to owner** | **Pass** |
| **UT-06** | **Requesting access for study group** | **To join study group** | **Logged in as a legit user** | **Click on recommended projects press on request access on the desired project** | **NULL** | **Notification sent to owner** | **Notification sent to owner** | **Pass** |
| **UT-07** | **Messaging** | **To check messaging** | **Logged in as a legit user**  **And should be a member of the group** | **Enter the text u want to send and click send** | **Message should not be null** | **Message sent** | **Message sent** | **Pass** |
| **UT-08** | **Uploading file for projects** | **Upload file for projects** | **Logged in as valid user and part of the project** | **Click on upload select file**  **Click on confirm upload** | **File should not be video or audio** | **Returns uploaded successfully** | **Returns uploaded successfully** | **Pass** |
| **UT-09** | **Editing groups**  **Info** | **Editing**  **Group info** | **Logger in as valid user**  **Should group owner** | **Click on the 3 lines(Burger)**  **edit it**  **press on save** | **Group name should not be null** | **Redirected to home page** | **Redirected to home page** | **Pass** |
| **UT-10** | **Editing project info** | **Editing project info** | **Logger in as valid user**  **Should have the permissions** | **Click on the 3 lines(Burger)**  **edit it**  **press on save** | **Project name should not be null** | **Redirected to home page** | **Redirected to home page** | **Pass** |
| **UT-11** | **Forgot passord page** | **Forgot password** | **Should know email** | **Click forgot password reset password** | **User should have signed up before** | **Redirected to login page** | **Redirected to login page** | **pass** |

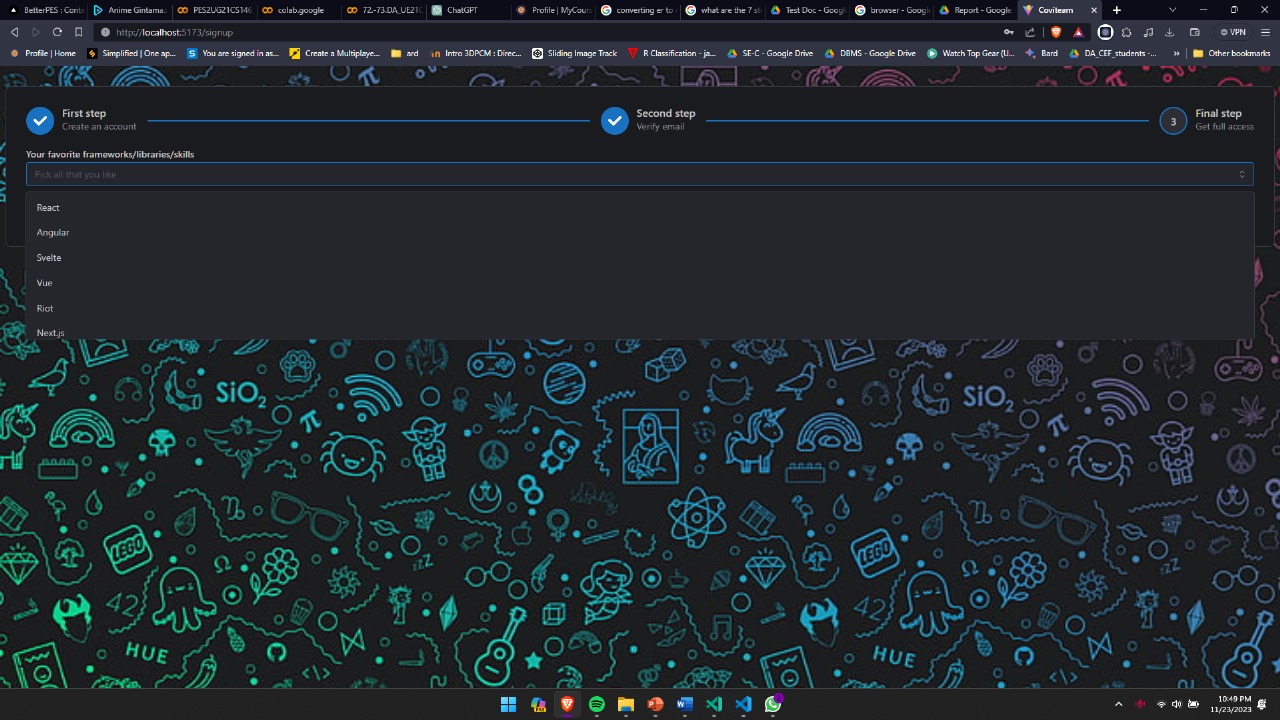
**SCREENSHOTS:**

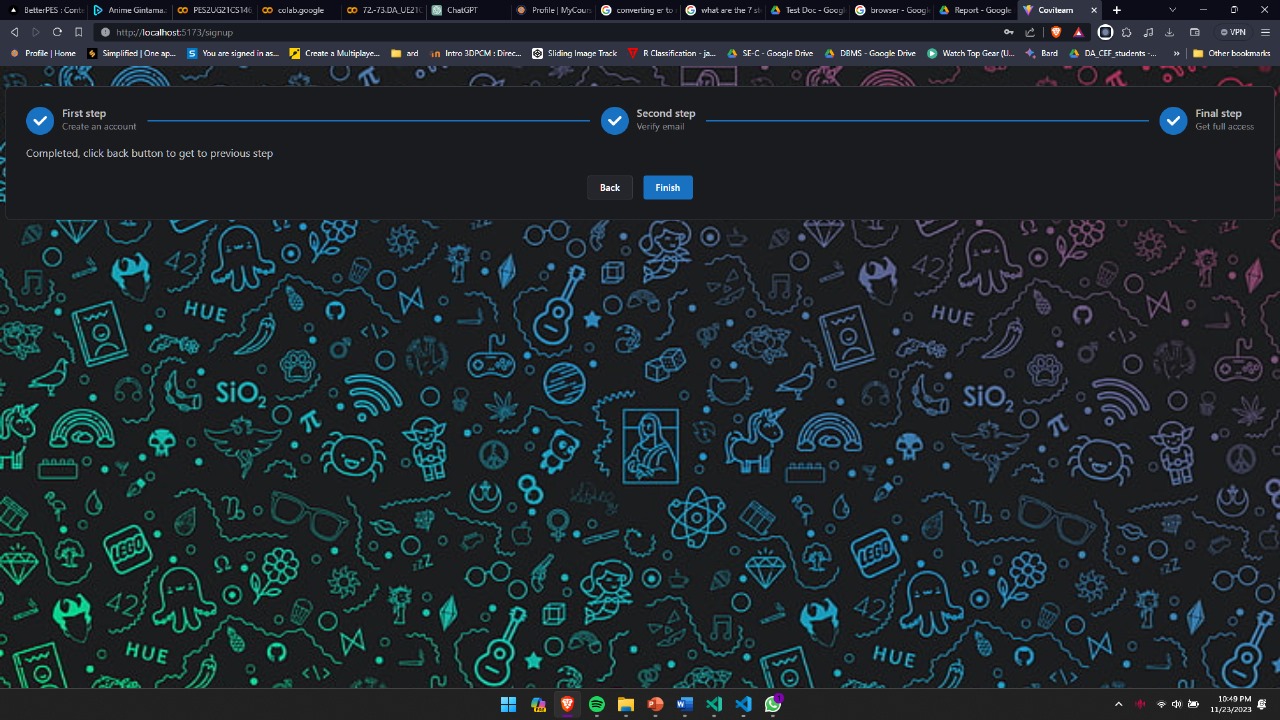
**Login page:**

**Sign up page:**

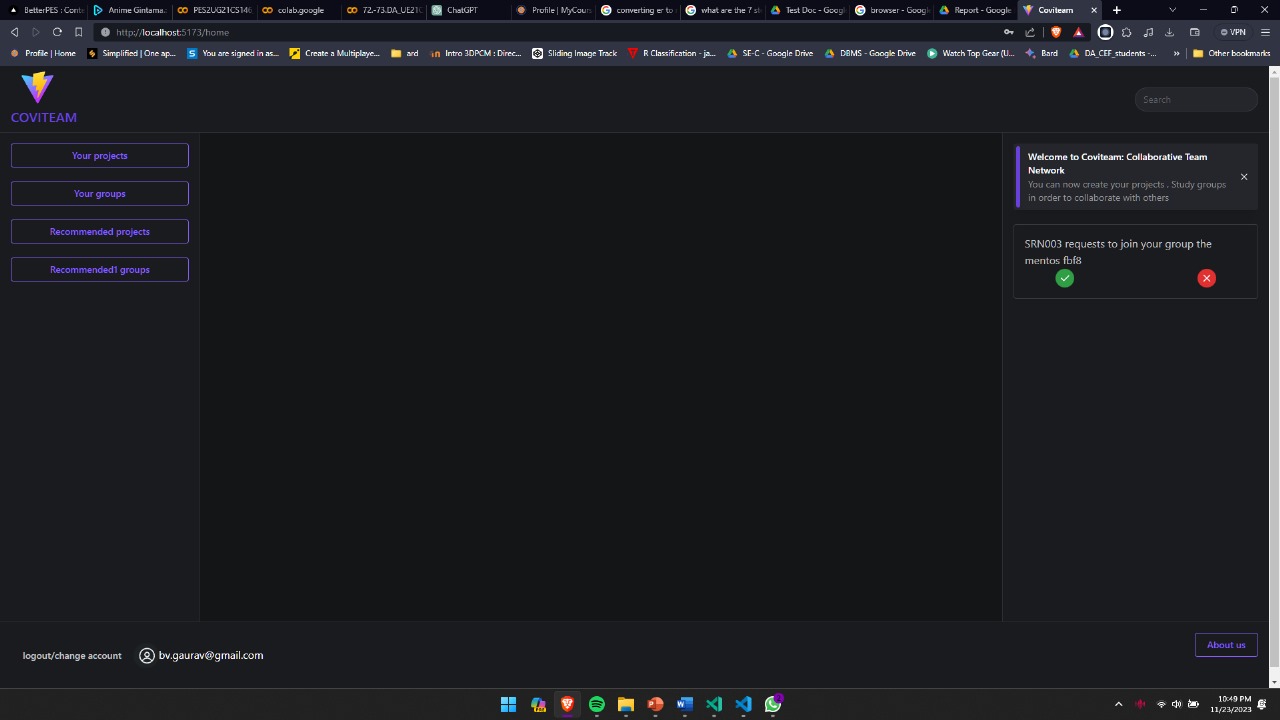




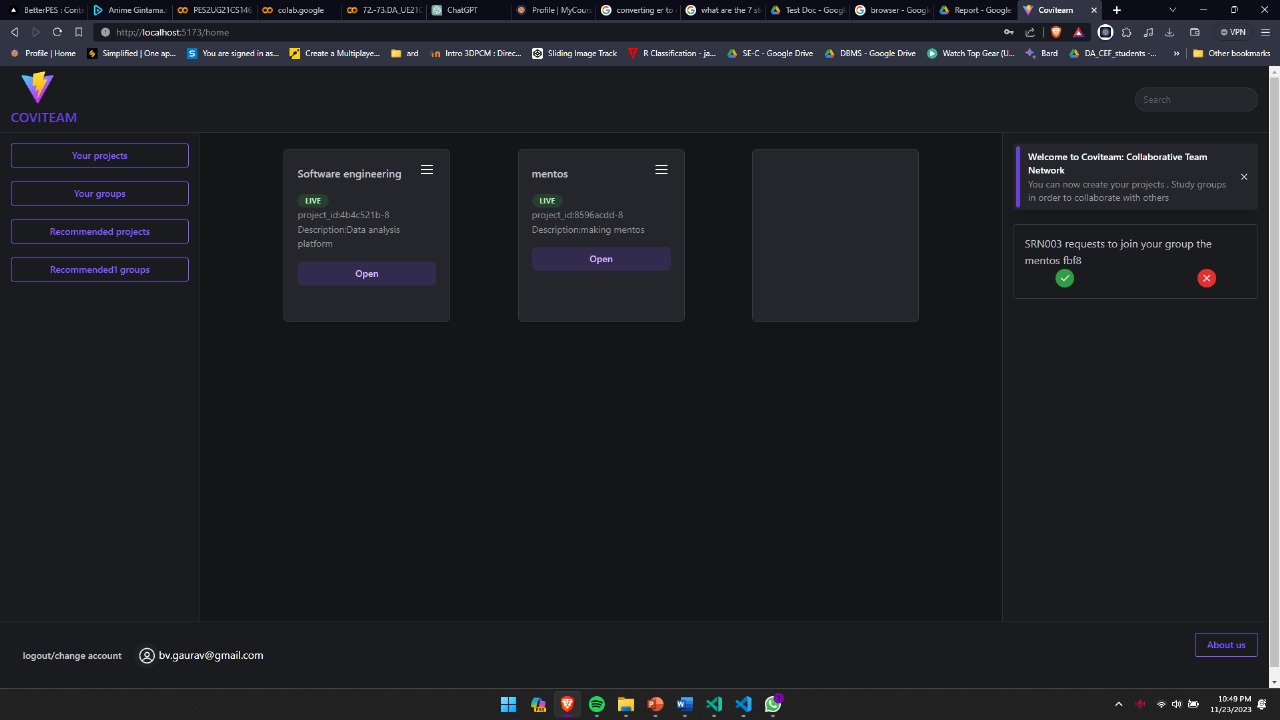




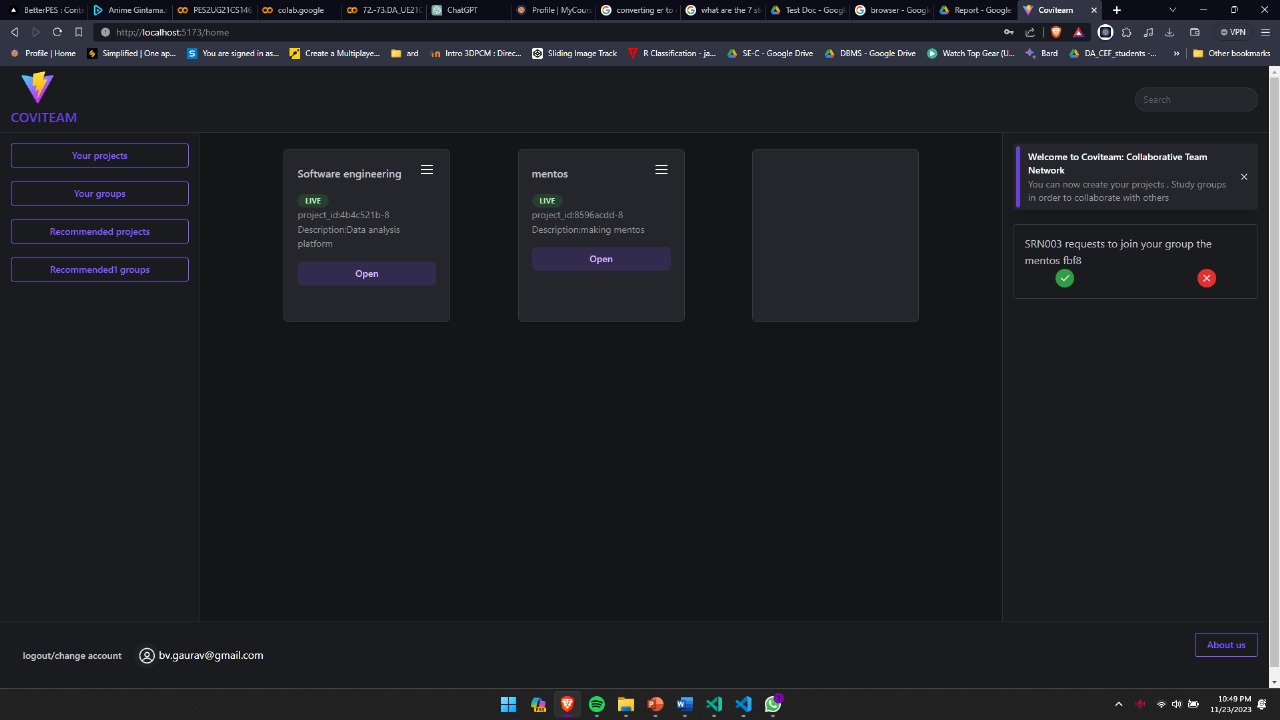
**Home page:**



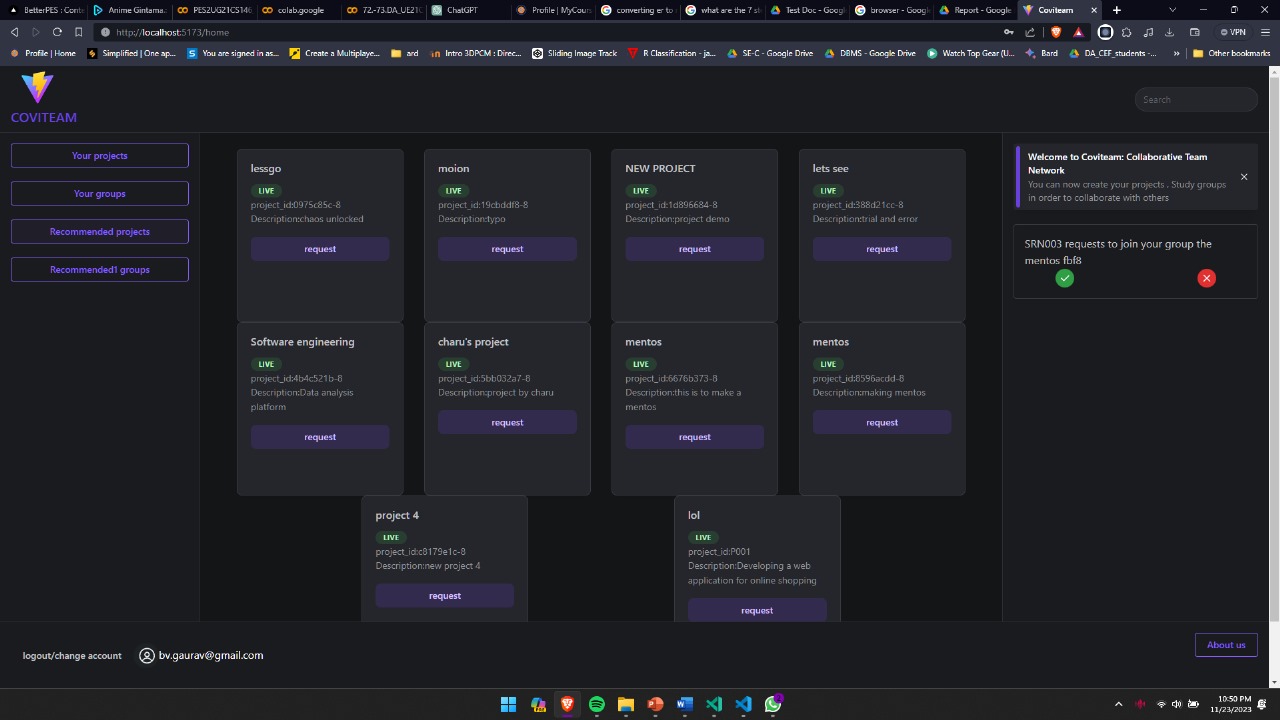
**Your projects:**



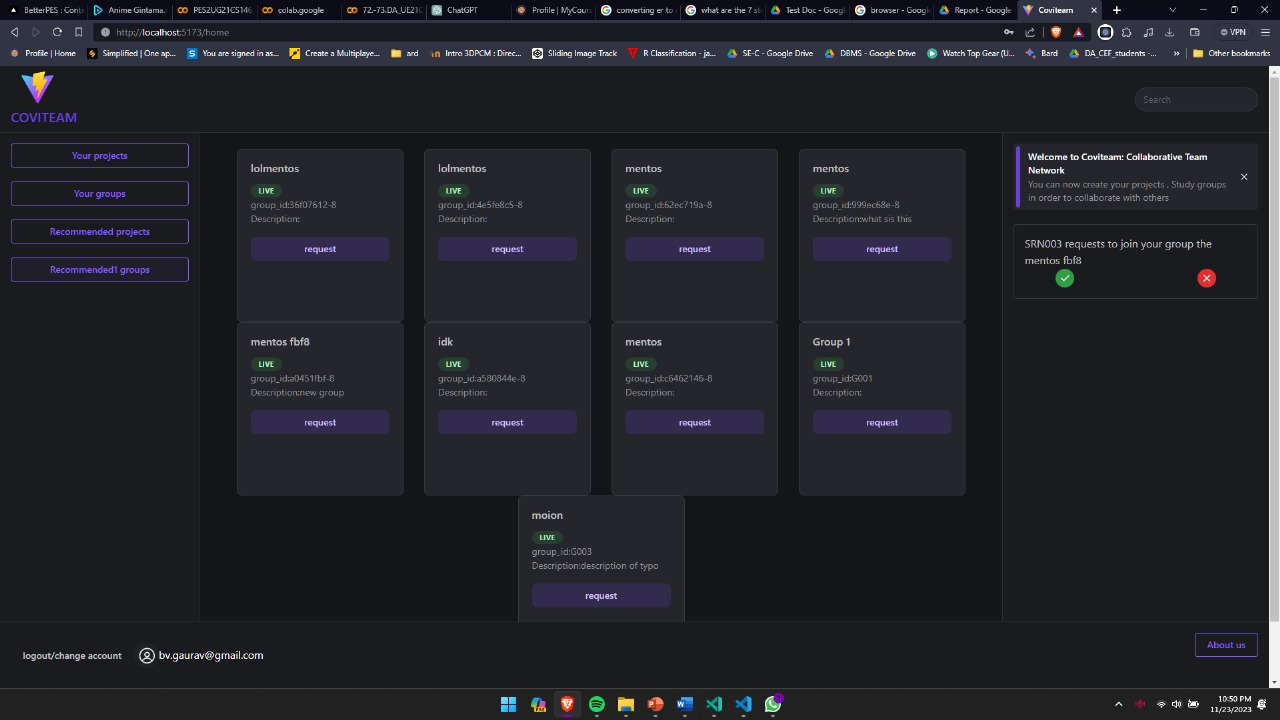
**Your groups:**



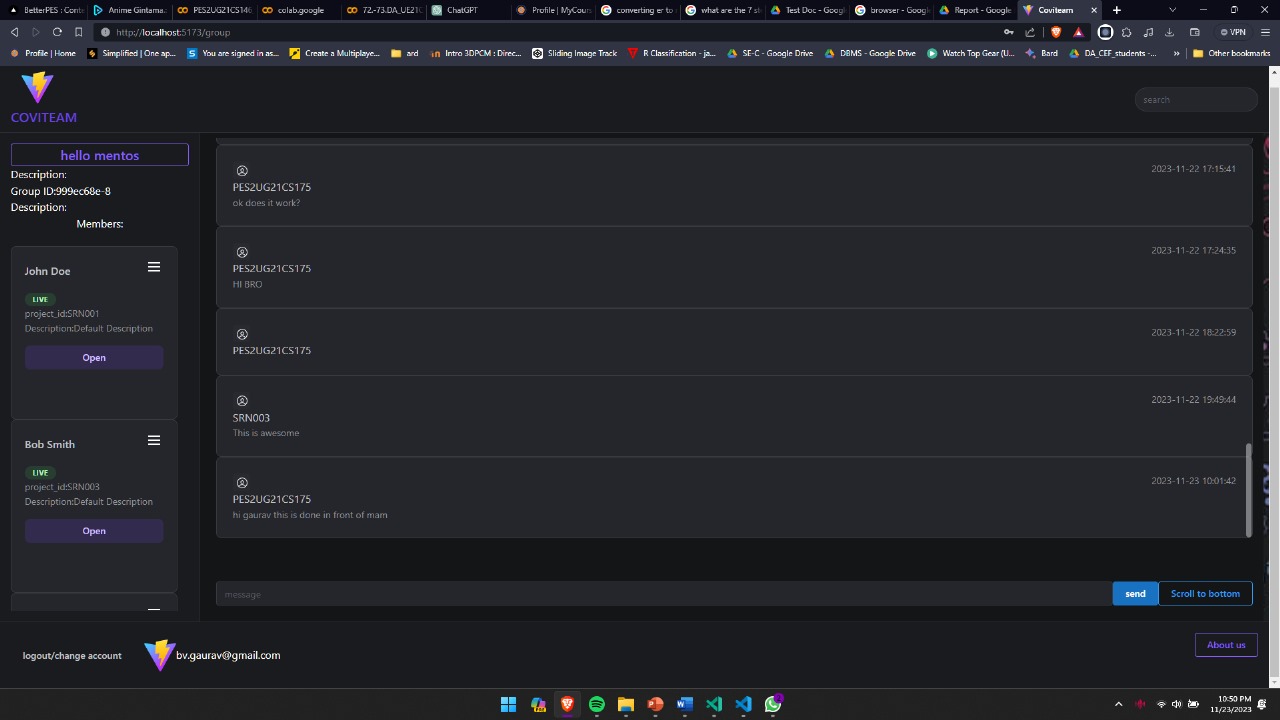
**All projects :**



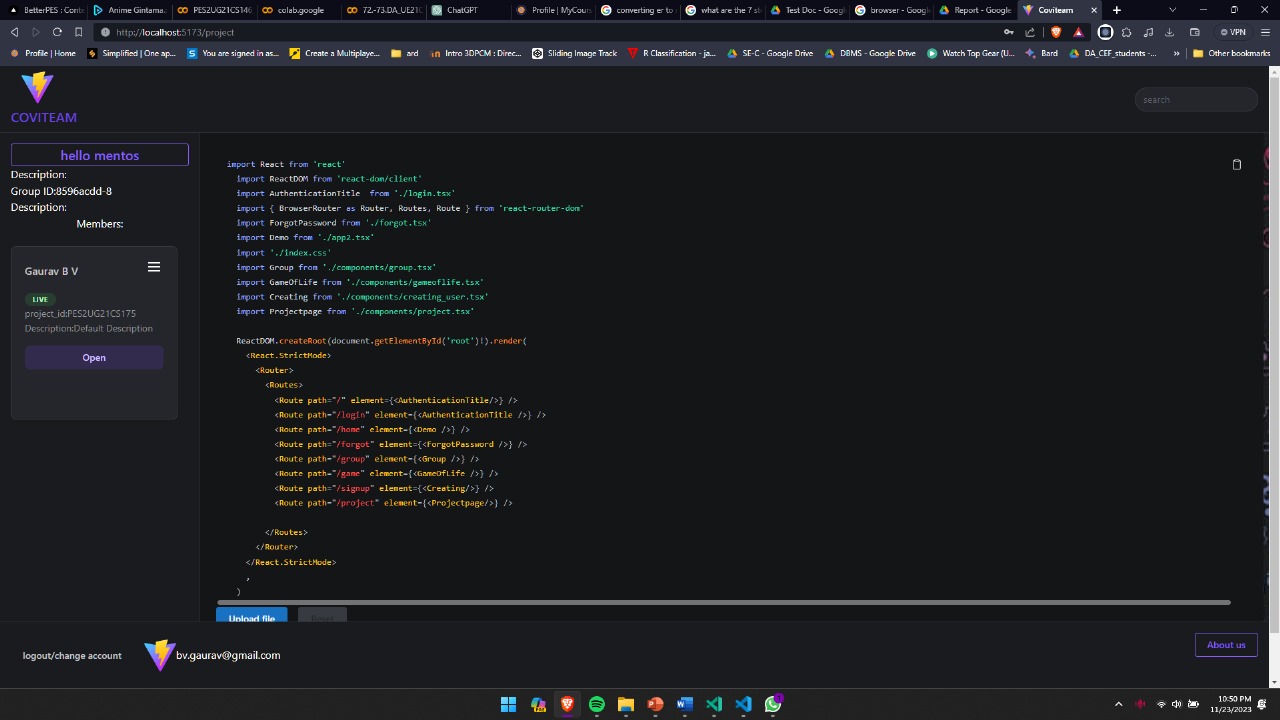
**All groups:**



**Groups descussion page:**



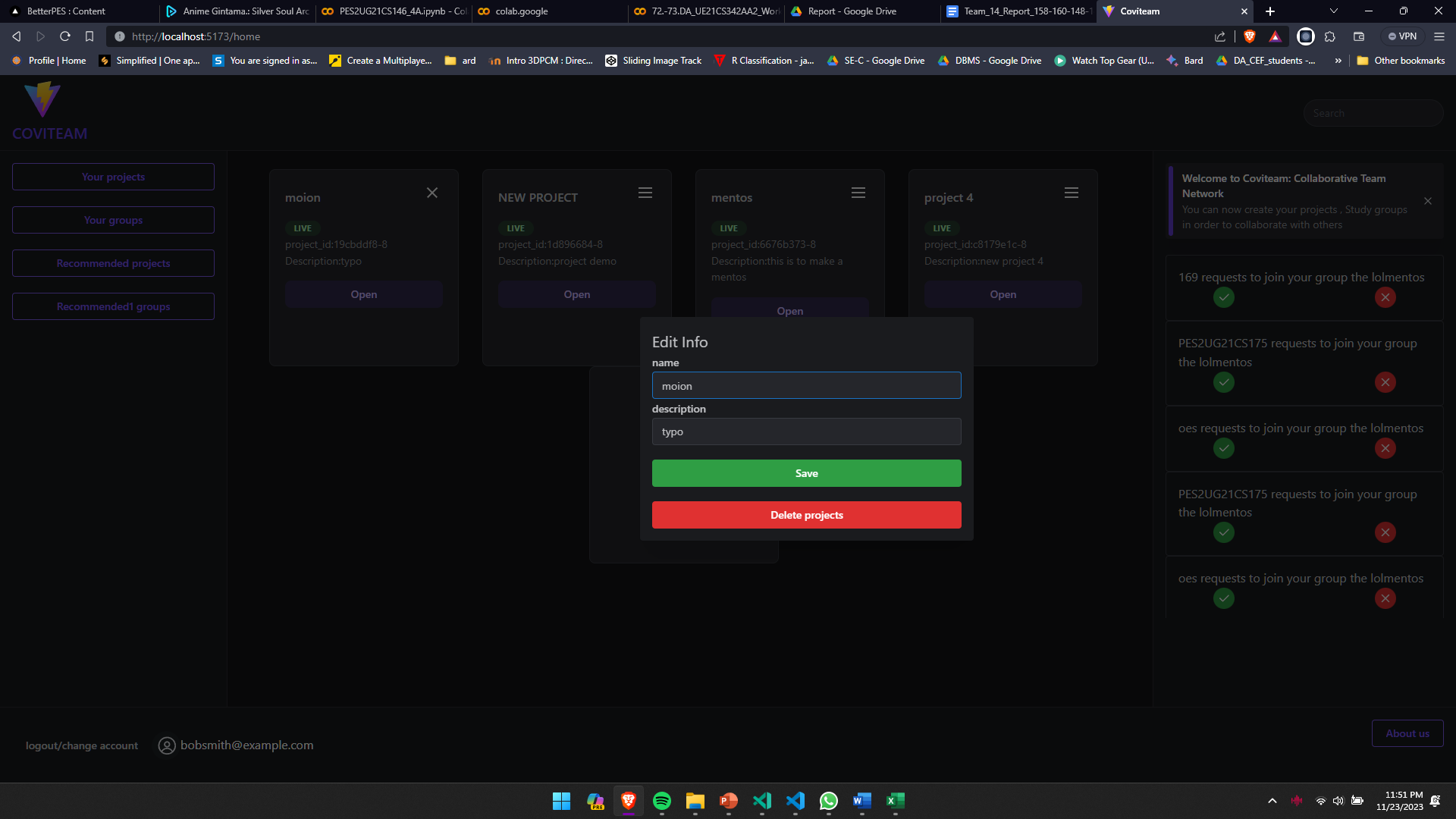
**Projects page:**



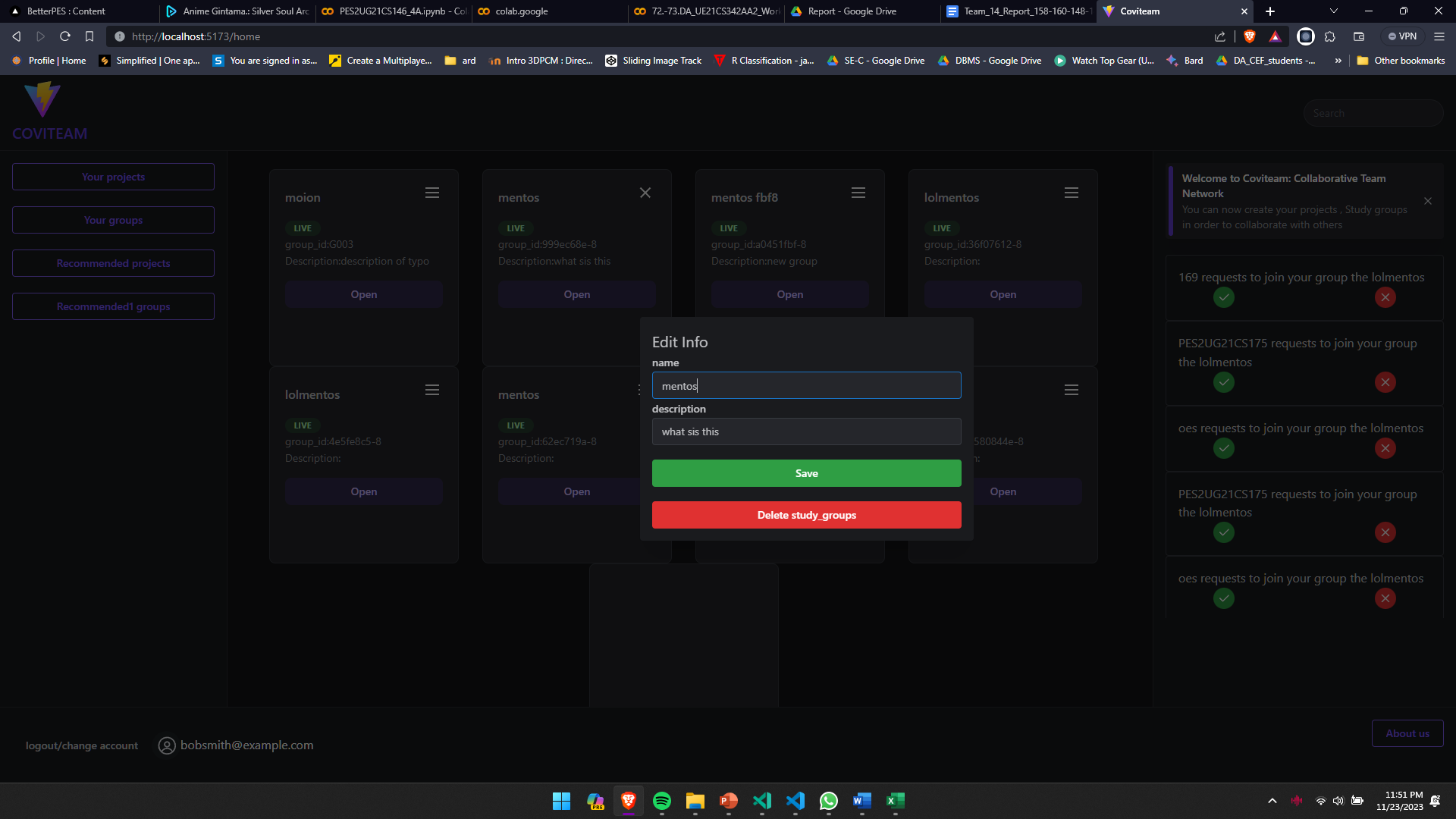
**Notfications right side of the page requesting access for groups and projects:**



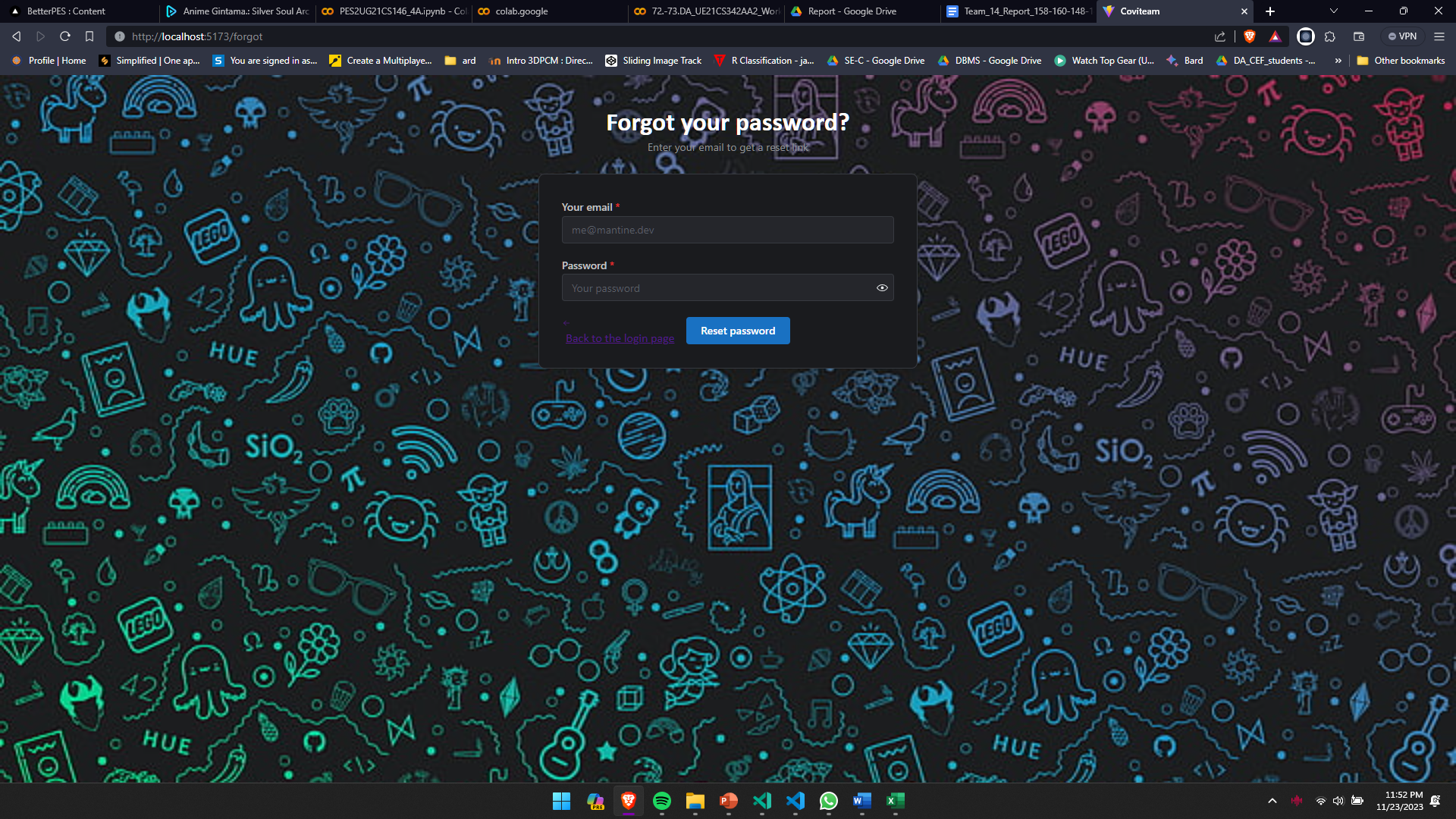
**Editing project:**



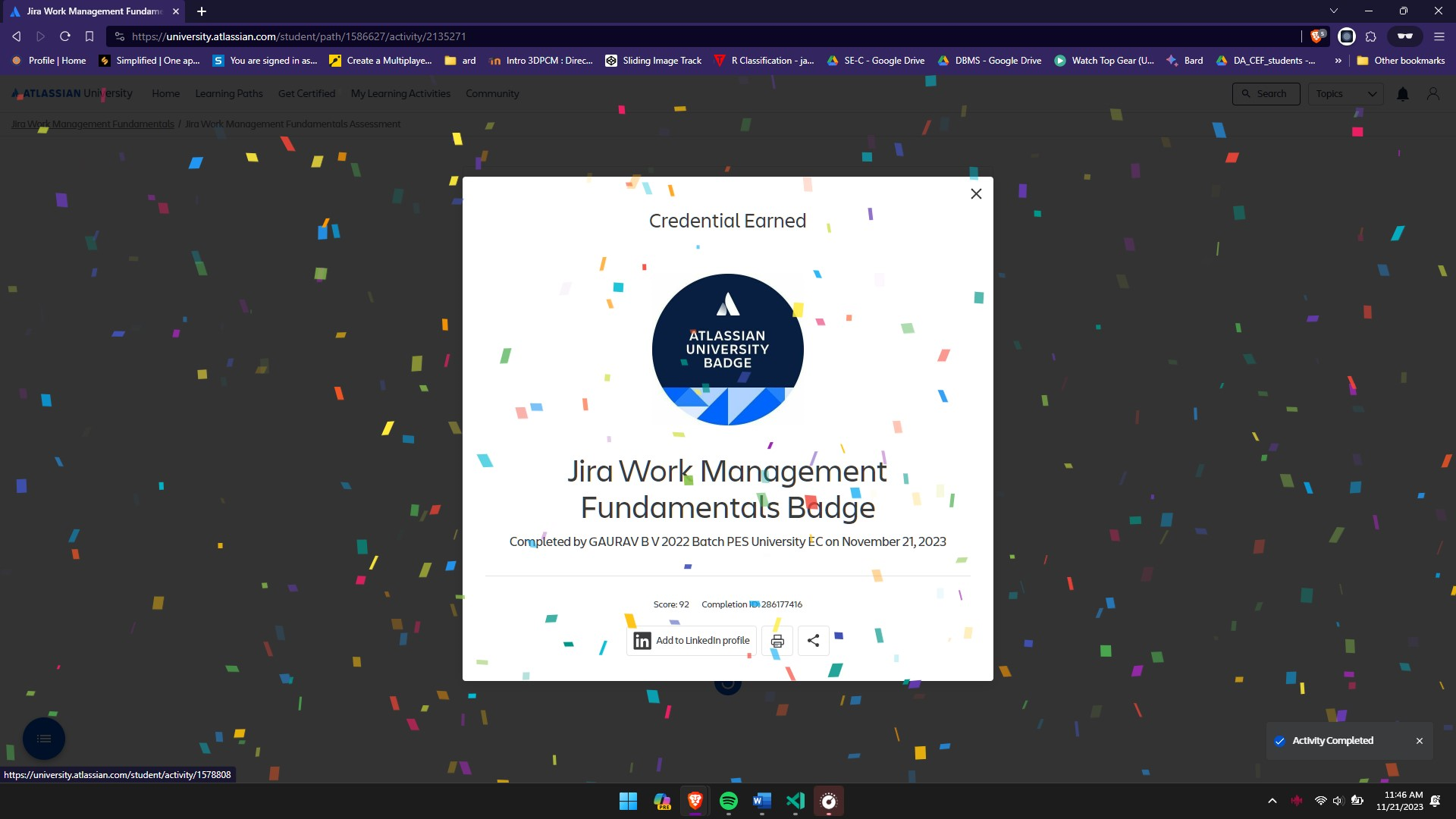
**Editing groups:**

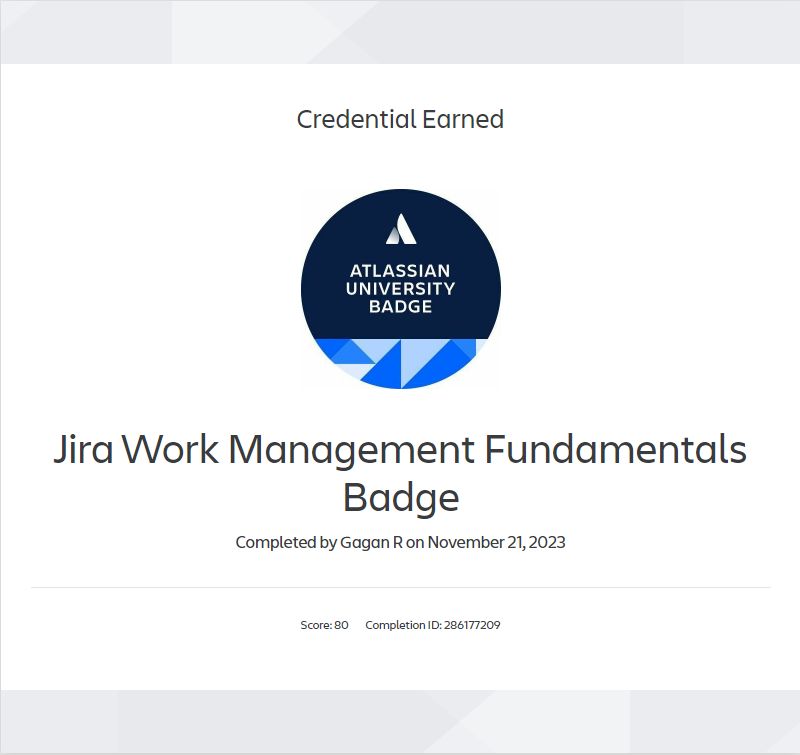


**Forgot password:**



**Jira Certificates:**





**References:**