

**INDIAN INSTITUTE OF TECHNOLOGY
(IIT) MADRAS – BS DEGREE**

Software Engineering – Group Project

ONLINE SUPPORT TICKETING SYSTEM

Project Report

- By ANSH KUSHWAHA

Roll No.: 21F1006019

&

PRAHLAD SINGHANIA

Roll No.: 21F1006059

TABLE OF CONTENTS

S. No.	Title	Page No.
1	Honour Code	3
2	Introduction	4
3	User Requirements	5
4	User Interfaces	7
5	Scheduling and Design	10
6	API Endpoints	15
7	Test Cases & Test Suite	18
8	Working Prototype	41
9	Implementation Details	42
10	Code Review, Issue Reporting & Tracking	43

HONOUR CODE

By ANSH KUSHWAHA
Roll No.: 21F1006019

I **ANSH KUSHWAHA** with roll no. **21F1006019** declare that I will not use any ideas, writings, code or work that is not my own or my group's with the intention of claiming it my or my group's work. For all the work that I will submit as part of this project, I will not share it outside my group with anybody directly or indirectly or upload it to any of the public forums on the internet.

I acknowledge that failing in any of the above constitutes plagiarism and in that case, the institute will take appropriate disciplinary action.

Sign: **ANSH KUSHWAHA**

Date: **2023-02-11**

By PRAHLAD SINGHANIA
Roll No.: 21F1006059

I **PRAHLAD SINGHANIA** with roll no. **21F1006059** declare that I will not use any ideas, writings, code or work that is not my own or my group's with the intention of claiming it my or my group's work. For all the work that I will submit as part of this project, I will not share it outside my group with anybody directly or indirectly or upload it to any of the public forums on the internet.

I acknowledge that failing in any of the above constitutes plagiarism and in that case, the institute will take appropriate disciplinary action.

Sign: **PRAHLAD SINGHANIA**

Date: **2023-02-12**

INTRODUCTION

Online Support Ticketing System:

Support Ticketing Systems are software applications that help businesses keep track of customer service requests. They generate a ticket for every inquiry and assign it to the appropriate team members. They also allow businesses to prioritise, manage, and resolve customer issues.

— Source: Internet

In the context of the IIT Madras BS Degree, the Online Support Ticketing System helps the support team (the POD Team) keep track of student's requests and queries.

About Project:

The project is a working web application of an online support ticketing system for IIT Madras BS degree. It is built mainly using Vue.js as a frontend, Python Flask as a backend, and SQLite as a database. It uses APIs for the interaction between the components and is currently hosted on Replit only for demonstration purposes.

Agile software development methodology was followed during the development of this web application.

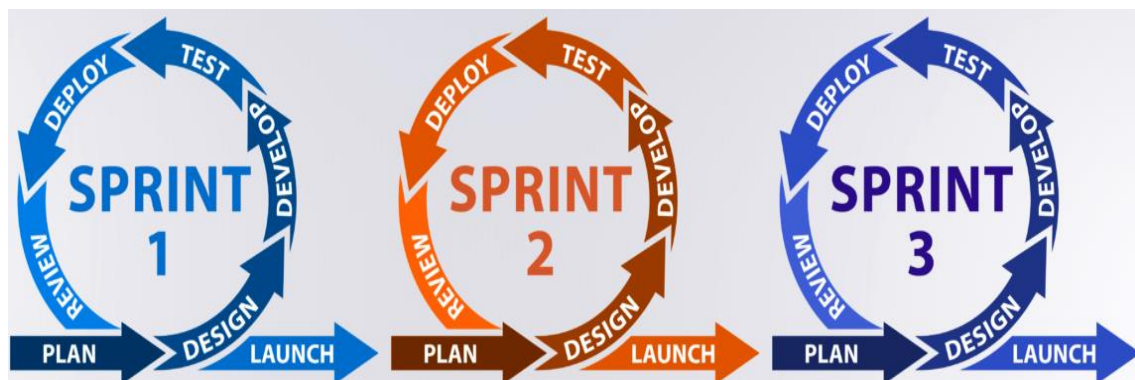
Agile Software Development Methodology:

Agile is an iterative approach to software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously, so teams have a natural mechanism for responding to change quickly.

— Source: Internet

Sprint is a short, time-boxed period when a team works to complete a set amount of work.

User Stories are short requirements or requests written from the perspective of an end user.



We have completed the whole work in six different sprints, which are as follows:

USER REQUIREMENTS

❖ Users of the web application - primary, secondary and tertiary users.

Primary Users:

Students enrolled in the IITM BSc degree program are the primary users of the application. They can create support tickets, track the status of their tickets, search for FAQs, and receive notifications when their tickets are resolved.

Secondary Users:

Support staff and **admins** are the secondary users of the application. They are responsible for addressing the support tickets, marking them as resolved, and updating the FAQs. Support staff should also be able to view reports and analyze ticket data to identify common concerns and track response times.

Tertiary Users:

Future students who may need support services are the tertiary users of the application. They can benefit from the FAQs and the knowledge base generated by the support team, which can help them find solutions to their queries and concerns. Additionally, tertiary users can also benefit from the feedback and ratings provided by primary users, which can help improve the quality of support services in the future.

❖ User Stories for the requirements, based on the SMART guidelines, in the format:

**As a [type of user],
I want [an action],
So that [a benefit/value]**

User Stories as a Student

- As a student,
I want to create a support ticket,
So that I can submit my concern or query to the support team.
- As a student,
I want to view a list of tickets before I create my ticket,
So that I can see if my concern or query has already been addressed.
- As a student,
I want to like or +1 an existing ticket,
So that popular concerns or queries can be prioritized by the support team.

- As a student,
I want to receive a notification when my ticket is resolved,
So that I know that my concern or query has been addressed.
- As a student,
I want to search for existing tickets,
So that I can find a solution to my concern or query quickly.
- As a student,
I want to attach files to my ticket,
So that the support team can have more information about my concern or query.
- As a student,
I want to view my previous tickets,
So that I can refer to them when I have a similar concern or query in the future.
- As a student,
I want to have the ability to cancel or re-open my ticket,
So that I can change my mind if I decide that my concern or query is no longer relevant.

User Stories as a Support Staff

- As a support staff,
I want to view and reply to existing tickets,
So that I can address the concerns and queries submitted by students.
- As a support staff,
I want to mark a ticket as resolved,
So that students know that their concern or query has been addressed.
- As a support staff,
I should tag other support staff on a ticket,
So that I can address complex concerns or queries more efficiently.
- As a support staff,
I want to view the history of all tickets resolved by me,
So that I can see how the support team has addressed similar concerns or queries in the past.
- As a support staff,
I want to view my performance metrics,
So that I can track my progress and identify areas for improvement.

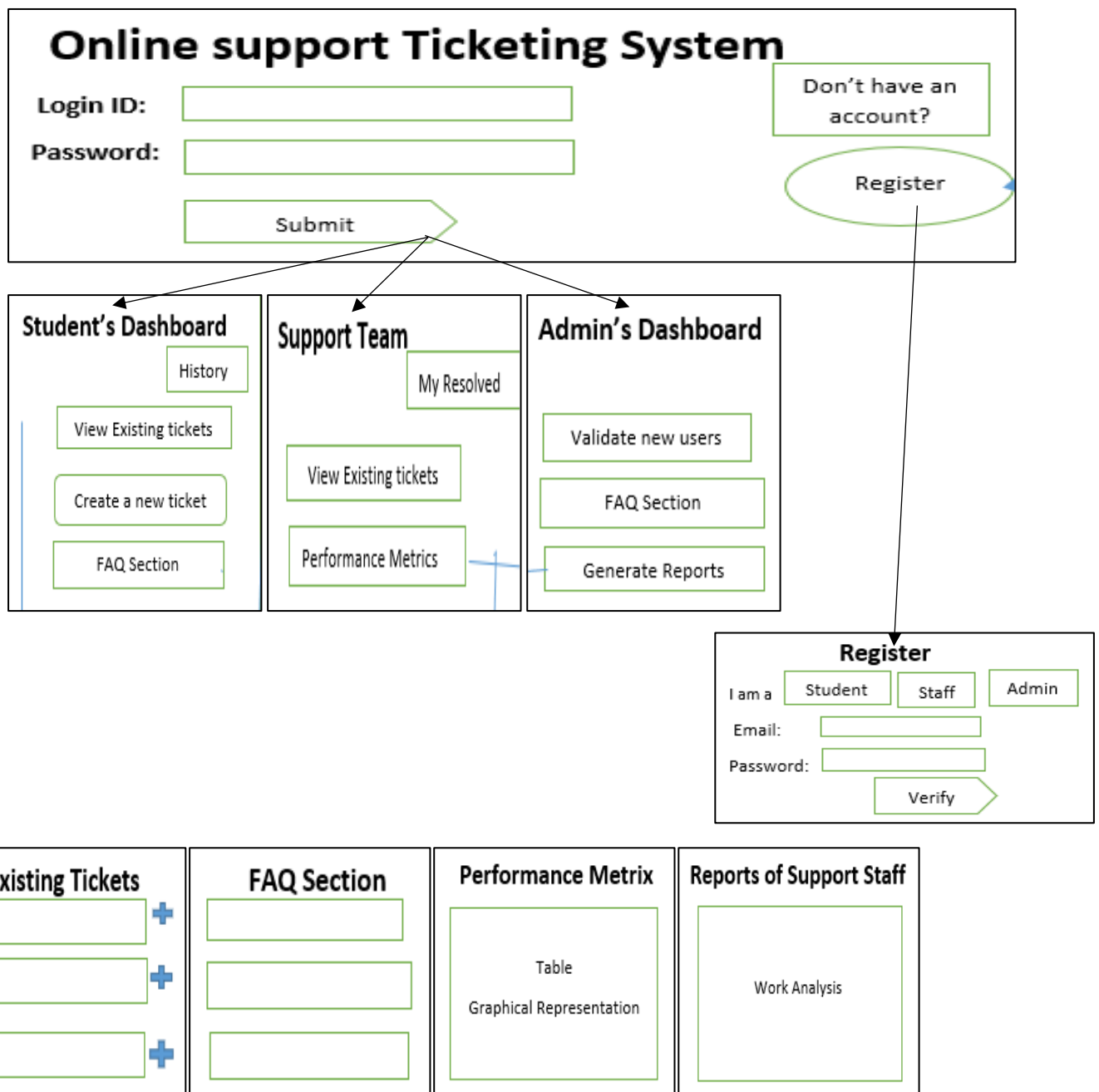
User Stories as an Admin

- As an admin,
I want to manage the FAQ section,
So that students have access to up-to-date information about common concerns and queries.

- As an admin,
I want approve newly registered users,
So that I can give them respective access.
- As an admin,
I want to generate reports on ticket activity,
So that I can monitor the performance of the support team and identify areas for improvement.

USER INTERFACES

❖ Low-Fidelity Wireframes for the Identified User Stories



❖ Usability Design Guidelines & Heuristics

Effectiveness:

- Provide clear and concise instructions for creating and managing support tickets.
- Allow users to easily find and view their existing support tickets.
- Ensure that support staff have access to all necessary information to address support tickets in a timely and effective manner.
- Allow users to provide feedback on the effectiveness of support responses and the resolution of their issues.
- Ensure that the support staff can respond to support tickets promptly and accurately, by providing them with the necessary tools and resources.
- Prioritize support tickets based on the popularity and urgency of the concerns or queries, so that the most pressing issues are addressed first.
- Provide support staff with a way to track the status of support tickets, so they can ensure that all support tickets are addressed in a timely and effective manner.

Efficiency:

- Provide a search function to allow users to quickly find similar support tickets and avoid creating duplicates
- Use automation to help streamline the support process and reduce the workload on support staff
- Allow support staff to easily categorize and prioritize support tickets to improve efficiency
- Provide keyboard shortcuts and other time-saving features to improve efficiency for power users
- Design the system to be quick and easy to use, with minimal steps required to create and respond to support tickets.
- Allow users to quickly search for existing support tickets or FAQs, to avoid duplicating support tickets or wasting time on questions that have already been answered.
- Provide support staff with tools to automate repetitive tasks, such as categorizing and tagging support tickets.

Safety:

- Ensure that user data is kept secure and that privacy policies are clearly communicated to users
- Provide clear warnings and confirmation messages before performing actions that could have serious consequences, such as deleting a support ticket
- Implement measures to prevent fraud and abuse, such as CAPTCHA verification for creating support tickets

- Ensure that the system protects user privacy and sensitive information, by requiring secure login credentials and using appropriate encryption and security protocols.
- Provide users with clear instructions and warnings when performing potentially risky actions, such as deleting support tickets or updating sensitive information.
- Provide support staff with tools to identify and remove inappropriate content, such as spam or abusive messages.

Learnability:

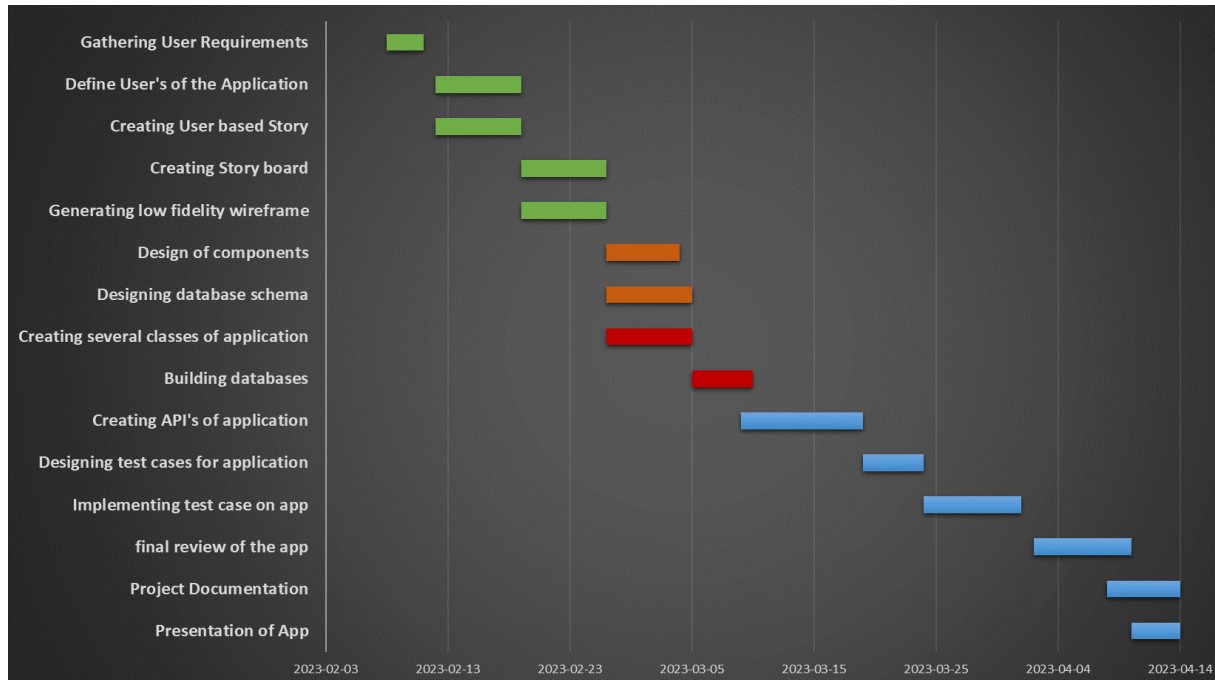
- Use a simple and consistent design to reduce the learning curve and help users quickly understand how to use the system
- Provide tutorials, tooltips, and help icons to guide users through the system
- Conduct user testing to identify areas of the system that are difficult to understand or use, and make improvements based on user feedback
- Provide a FAQ section that is easy to access and search, and that is regularly updated based on common support tickets and user feedback
- Design the system to be intuitive and easy to use, with clear labels and instructions that guide users through the support ticket creation process.
- Provide users with access to contextual help and documentation, such as tooltips, pop-ups, or a searchable FAQ section.
- Allow users to experiment with the system without fear of making irreversible mistakes, by providing undo and redo features.

Memorability:

- Use clear and concise language and labels to help users remember how to use the system
- Provide a history of support tickets and responses so that users can easily refer back to previous interactions
- Use consistent design patterns and navigation to help users remember how to use the system
- Use visual cues, such as icons and colors, to help users remember the meaning of different actions and elements in the system
- Design the system to be memorable and easy to use, by using consistent design elements, such as color schemes, fonts, and icons.
- Use clear and concise language and instructions that are easy to remember and understand, so users don't have to rely on external documentation or help.
- Provide users with tools to easily save and retrieve support tickets or frequently asked questions, such as bookmarking or a favorites list.

SCHEDULING & DESIGN

❖ Project Schedule

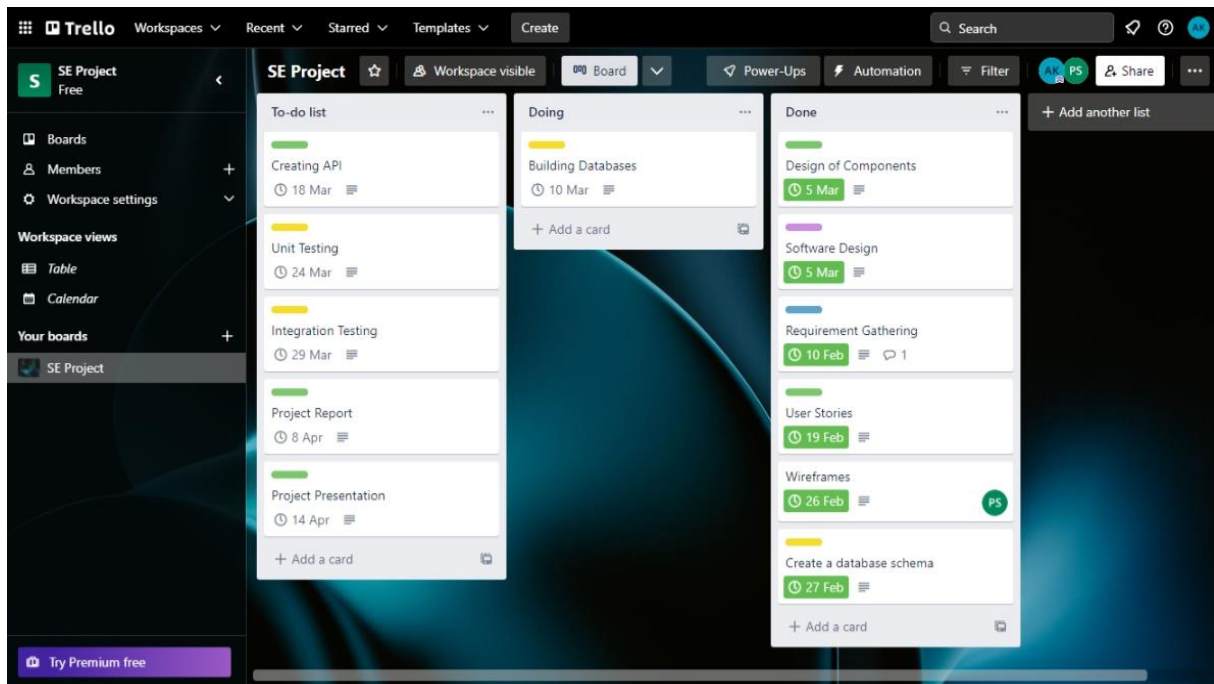


Gantt Chart

Task Name	Start (Date)	End (Date)	Duration (Days)
Gathering User Requirements	2023-02-08	2023-02-11	3
Define User's of the Application	2023-02-12	2023-02-19	7
Creating User based Story	2023-02-12	2023-02-19	7
Creating Story board	2023-02-19	2023-02-26	7
Generating low fidelity wireframe	2023-02-19	2023-02-26	7
Design of components	2023-02-26	2023-03-05	6
Designing database schema	2023-02-26	2023-03-05	7
Creating several classes of application	2023-02-26	2023-03-05	7
Building databases	2023-03-05	2023-03-10	5
Creating API's of application	2023-03-09	2023-03-19	10
Designing test cases for application	2023-03-19	2023-03-24	5
Implementing test case on app	2023-03-24	2023-04-01	8
final review of the app	2023-04-02	2023-04-10	8
Project Documentation	2023-04-08	2023-04-14	6
Presentation of App	2023-04-10	2023-04-14	4

❖ Project Scheduling Tool

Tool Use: Trello Board



Trello Board

❖ Design of Components – Description of different components of the system

User Management Component:

The user management component will handle user authentication, registration, and management. It will help to provide different access levels to users based on their roles, such as students, support staff, and admins. Users will be able to create accounts, update their profiles, change passwords, and view their support tickets. The component will also ensure that user data is secure by implementing security measures like encryption and access control.

Support Ticketing Component:

The support ticketing component will allow users to create new support tickets, view their existing tickets, and search for tickets by tags and title. The support staff will be able to view and respond to tickets, mark them as resolved, and close them. This component will also include features like ticket tracking, ticket assignment, and ticket escalation. The component will ensure that support tickets are handled efficiently and effectively, and that users are updated on the status of their tickets in a timely manner.

Notification Component:

The notification component will handle the notification system. It will send notifications to users when their tickets are created, updated, or resolved. It will also send notifications to support staff when new tickets are created, and to admins when critical tickets are not resolved within a certain time frame. The component will ensure that users are informed of the status of their tickets and that the support team is aware of new and critical tickets.

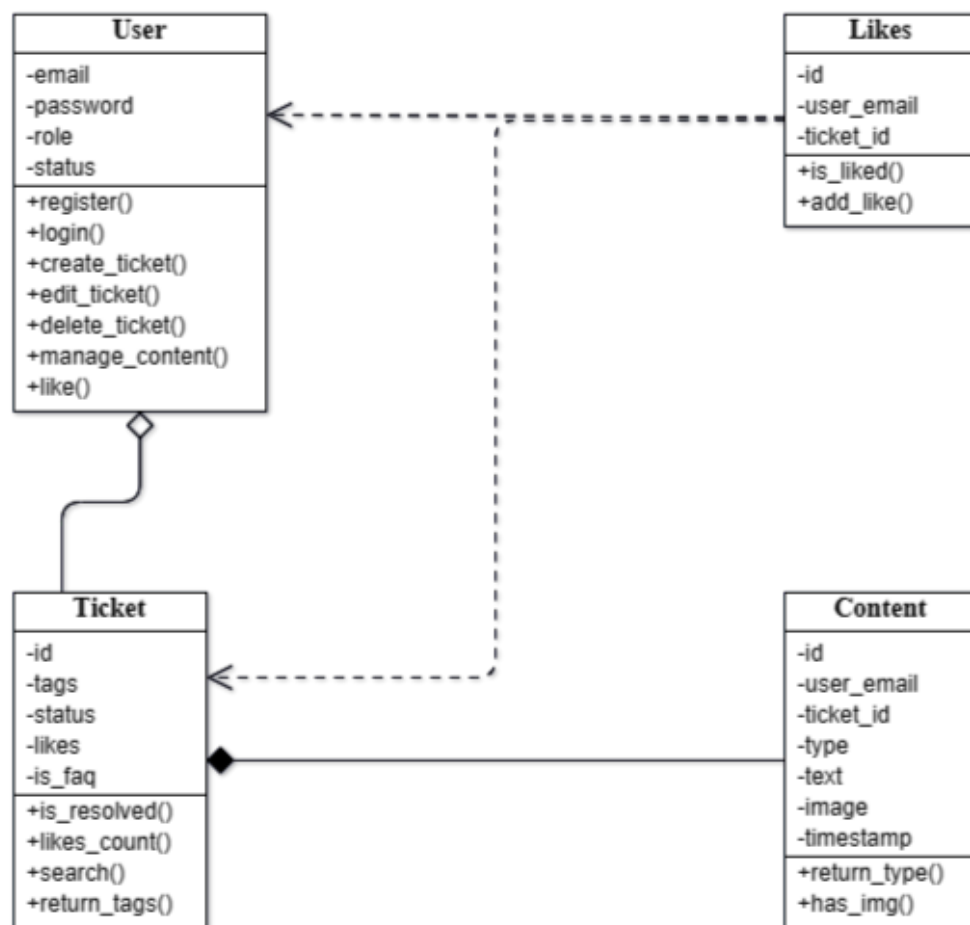
FAQ Management Component:

The FAQ management component will allow support staff to manage the FAQ section of the system. They will be able to create, update, and categorize support queries and responses. This component will also include features like search, filtering, and sorting, so that users can easily find the relevant information. The component will ensure that users have access to up-to-date and relevant information and that support staff can efficiently manage the FAQ section.

Analytics Component:

The analytics component will provide admins with analytical insights into the support process. It will track metrics like the number of tickets created, resolution time, user satisfaction, and support staff performance. This information will be used to optimize the support process and improve the system's performance. The component will ensure that the system is operating efficiently and that user needs are being met.

❖ Software Design – Basic Class Diagram of the proposed system



❖ **Details / Minutes of Scrum Meetings****Date: 20th Feb**

First name ▼	Last name ▼	Email ▼	Duration ▼	Time joined ▼	Time exited ▼
ANSH	KUSHWAHA	21f100601	27 min	05:15 PM	05:42 PM
PRAHLAD	SINGHANIA	21f100605	30 min	05:11 PM	05:42 PM

Work Reported:

- Requirements gathered
- Identified users of the application
- User stories created & presented based on smart guidelines

Upcoming Task:

- To create a storyboard of the application
- Create low-fidelity wireframes using usability guidelines

Issues Resolved:

- Some user stories look challenging for us. Ansh sorted out this problem by discussing and researching various method that we can implement in the long run.

Date: 26th Feb

First name ▼	Last name ▼	Email ▼	Duration ▼	Time joined ▼	Time exited ▼
ANSH	KUSHWAHA	21f1006019	1 hr 9 min	10:40 PM	11:49 PM
PRAHLAD	SINGHANIA	21f1006059	1 hr 11 min	10:39 PM	11:49 PM

Work Reported:

- Created a low-fidelity wireframe that follows usability guidelines
- Designed various interactions of functions for each UI

Upcoming task:

- To prepare Project schedule in Tello workspace.
- To come up with different design of component
- Preparing a class diagram for the application.

Date: 1st March

First name ▼	Last name ▼	Email ▼	Duration ▼	Time joined ▼	Time exited ▼
ANSH	KUSHWAHA	21f1006019@	30 min	08:02 PM	08:32 PM
PRAHLAD	SINGHANIA	21f1006059@	38 min	07:54 PM	08:32 PM

Work Reported:

- Created a workspace to organize project in an effective way.
- Used a tool Tello for Project Scheduling. Added the task done and needs to be done so as to bring everyone on the same page
- Designed components of the Application

Upcoming Task:

- Creating a database for the application.
- Generating APIs for each component
- Started preparing class diagrams

Issue Resolved:

- Clarified all the methods of the class required for the application.

Date: 4th March

First name ▼	Last name ▼	Email ▼	Duration ▼	Time joined ▼	Time exited ▼
ANSH	KUSHWAHA	21f1006019	23 min	06:37 PM	07:00 PM
PRAHLAD	SINGHANIA	21f1006059	28 min	06:32 PM	07:00 PM

Work Reported:

- Designed database schema of the application
- Created an SQLite database for the application
- Prepared a class diagram for the application

Upcoming task:

- ❖ Creating API and different methods corresponding to it.

Issues Resolved:

- ❖ Ansh came up with an idea of storing image in SQLite database.

API ENDPOINTS

Software Engineering Project 1.0.11 OAS3

IIT Madras - BS Degree - Online Support Ticketing System

[Contact the developer](#)

Servers

▼

HomePage Operations on HomePage ^

GET / Display homepage ▼

POST / Redirect to login ▼

User Operations about User ^

GET /user Fetch all users ▼

POST /user Add details of user ▼

GET /user/{email} Returns User ▼

PUT /user/{email} Update User ▼

DELETE /user/{email} Delete user ▼

Ticket Operations on Ticket ^

GET /ticket Get ticket details ▼

POST **/ticket** Add ticket details



GET **/ticket/{id}** Get Ticket



PUT **/ticket/{id}** Update Ticket



DELETE **/ticket/{id}** Delete Ticket



Content

Operations on Content



GET **/ticket/{id}/content** Returns All Contents of Given Ticket



POST **/ticket/{id}/content** Create Content for the Given Ticket



GET **/ticket/{id}/content/{i}** Returns Content



PUT **/ticket/{id}/content/{i}** Modify Content



DELETE **/ticket/{id}/content/{i}** Remove Content



Like

Operations about Like



GET **/like** Returns All Likes



GET **/ticket/{id}/like** Returns All Likes on Given Ticket



POST **/ticket/{id}/like** Add Like on Given Ticket



DELETE **/ticket/{id}/like** Remove Like on Given Ticket



Schemas

User ▾ {
 email > [...]
 password > [...]
 role > [...]
 status > [...]

}

Ticket ▾ {
 id > [...]
 user_email > [...]
 title > [...]
 tags > [...]
 desc > [...]
 image > [...]
 status > [...]
 likes > [...]
 is_faq > [...]
 timestamp > [...]
 contents > [...]

}

Content ▾ {
 id > [...]
 user_email > [...]
 ticket_id > [...]
 text > [...]
 image > [...]
 timestamp > [...]

}

Like ▾ {
 id > [...]
 user_email > [...]
 ticket_id > [...]

}

TEST CASES & TEST SUITE

Page Being Tested	HomePage
Inputs	Email = @web.app
Expected Output	Enter a part followed by '@'
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Email = emailweb.app
Expected Output	Include an @ in the email address
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Email = email@.app
Expected Output	'.' is used at a wrong position
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Email = email@webapp
Expected Output	Correct Input Format
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Email = email@web.app
Expected Output	Correct Input Format
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Email = email@web.
Expected Output	‘.’ is used at a wrong position
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Email = email@web app
Expected Output	Part following ‘@’ should not contain the symbol ‘ ‘
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Password = blank (empty input)
Expected Output	Fill out this field
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Password = p@ssw0rd
Expected Output	Match the requested format
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Password = P@SSWORD
Expected Output	Match the requested format
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Password = P@ssword
Expected Output	Match the requested format
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Password = Passw0rd
Expected Output	Match the requested format
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Password = P@ss w0rd
Expected Output	Match the requested format
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	Password = P@ssw0rd
Expected Output	Correct Input Format
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	On Login with Any Incorrect Password
Expected Output	Error Page
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	On Login with Correct Password
Expected Output	DashBoard
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	If New Password Input didn't match with Confirm Password Input
Expected Output	Retyped Password didn't match
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	While Registering, If User Already Exists
Expected Output	Go Back & Log In
Actual Output	As Expected
Result	Success

Page Being Tested	HomePage
Inputs	While LogIn, If User isn't Registered
Expected Output	Go Back & Register
Actual Output	As Expected
Result	Success

Page Being Tested	User API (/user)
Status	Admin logged in get all users
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Admin logged in get particular user detail
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	User API (/user)
Status	Admin logged out get all users
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Admin logged in get particular user detail
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	User API (/user)
Status	Add users details
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	User API (/user)
Status	Add user improper role
Expected Status Code	400
Actual Status Code	400
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	User logged out edit details
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Admin logged edit user status that doesn't exist
Expected Status Code	404
Actual Status Code	404
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	User logged in change password
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	User logged in edit role
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Admin logged in edit role
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Admin logged in change status
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Admin logged in change password
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Support login change role
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Student login change status
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	User logged out delete user
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	User logged in delete user that doesn't exist
Expected Status Code	404
Actual Status Code	404
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	User logged in delete user
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Admin logged in delete admin
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	User API (/user/<string:email>)
Status	Support staff logged in delete user
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket)
State	Student not logged in add ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket)
State	Student logged in add ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket)
State	User not logged in fetch all existing tickets
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket)
Status	User logged in fetch all existing tickets.
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	User not logged in fetch details of ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	User logged in fetch details of ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API(/ticket)
Status	Student logged in fetch ticket details of user
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	User not logged in edit details of a ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Student logged in edit details of a ticket which doesn't exist
Expected Status Code	404
Actual Status Code	404
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Student logged in edit title of a ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Student logged in edit tags of a ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Student logged in edit description of a ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Student logged in change image of a ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Support staff logged in change status of a ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Admin logged in edit faq section with a ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Support staff logged in edit title of a ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Support staff logged in edit tags of a ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Support staff logged in edit description of a ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Support staff logged in change image of a ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Student logged in change status of a ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Support staff logged in edit faq section with a ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	User not logged in delete a ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Student logged in delete a ticket that didn't exist
Expected Status Code	404
Actual Status Code	404
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Student logged in delete a ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Admin logged in delete a ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Ticket API (/ticket/<int:id>)
Status	Support staff logged in delete a ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content)
Status	User logged out add content
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content)
Status	Student logged in add content
Expected Status Code	404
Actual Status Code	404
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content)
Status	Student logged in add content
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content)
Status	User logged out get all content
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content)
Status	User logged in get all content
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	User logged out get particular content
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	User logged in get particular content
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content)
Status	User logged in get all for that user
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	User logged out edit content
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	User logged in edit content that doesn't exist
Expected Status Code	404
Actual Status Code	404
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	Support logged in edit text
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	Support logged in edit image
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	Student logged in edit text
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	Student logged in edit image
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	User logged out delete content
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	Student logged in delete content that doesn't exist
Expected Status Code	404
Actual Status Code	404
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	Student logged in delete content
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	Admin logged in delete content
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Content API (/ticket/<int:id>/content/<int:content_id>)
Status	Support staff logged in delete content
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Like API (/ticket/<int:id>/like)
Status	User logged out like ticket
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Like API (/ticket/<int:id>/like)
Status	Student logged in like ticket that doesn't exist
Expected Status Code	404
Actual Status Code	404
Result	Success

Page Being Tested	Like API (/ticket/<int:id>/like)
Status	Student logged in like ticket
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Like API (/ticket/like)
Status	User logged out get all likes
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Like API (/ticket/like)
Status	User logged in get all likes
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Like API (/ticket/<int:id>/like)
Status	User logged out get all ticket likes
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Like API (/ticket/<int:id>/like)
Status	User logged in get all ticket likes
Expected Status Code	200
Actual Status Code	200
Result	Success

Page Being Tested	Like API (/ticket/<int:id>/like)
Status	User logged out delete like
Expected Status Code	401
Actual Status Code	401
Result	Success

Page Being Tested	Like API (/ticket/<int:id>/like)
Status	User logged in delete like for ticket that doesn't exist
Expected Status Code	404
Actual Status Code	404
Result	Success

Page Being Tested	Like API (/ticket/<int:id>/like)
Status	User logged in delete like
Expected Status Code	200
Actual Status Code	200
Result	Success

WORKING PROTOTYPE

The Web Application is hosted on Replit & can be accessed publicly by the link given below:

SE Project (ansh-kushwaha.repl.co) [<https://se-project.ansh-kushwaha.repl.co/>]

IMPLEMENTATION DETAILS

Technologies Used:

- ❖ Python
 - Flask
 - flask-marshmallow
 - Flask-RESTful
 - Flask-SQLAlchemy
 - marshmallow
 - marshmallow-sqlalchemy
 - SQLAlchemy
- ❖ SQLite
- ❖ HTML
- ❖ CSS
- ❖ JavaScript [Vue.js]
- ❖ Bootstrap

Tools Used:

- ❖ Replit IDE (for development environment)
- ❖ Trello (for schedule tracking)
- ❖ Google Drive (for documents collaboration)
- ❖ Google Meet (for collaborative development – pair programming)

Application Hosting:

- ❖ Hosting Platform: Replit
- ❖ Hosted Link: [SE Project](https://se-project.ansh-kushwaha.repl.co/) [<https://se-project.ansh-kushwaha.repl.co/>]

Instruction to Run the Application:

- ❖ Access the running web application via Hosted Link OR
- ❖ Run on your Local System:
 - Download the Code & Extract it
 - Open the parent folder of “main.py” file in a Linux Terminal
 - Create a Virtual Environment using command:


```
python -m venv ./
```
 - Activate Virtual Environment using command:


```
source ./bin/activate
```
 - Install required packages using command:


```
pip install -r requirements.txt
```
 - Run the Application using command:


```
python main.py
```
 - Access the running web application of any JavaScript enabled browser at:


```
https://127.0.0.1/5000
```
 - To stop the running application, press keys:


```
Ctrl + C
```
 - To shut down the Virtual Environment, use command:

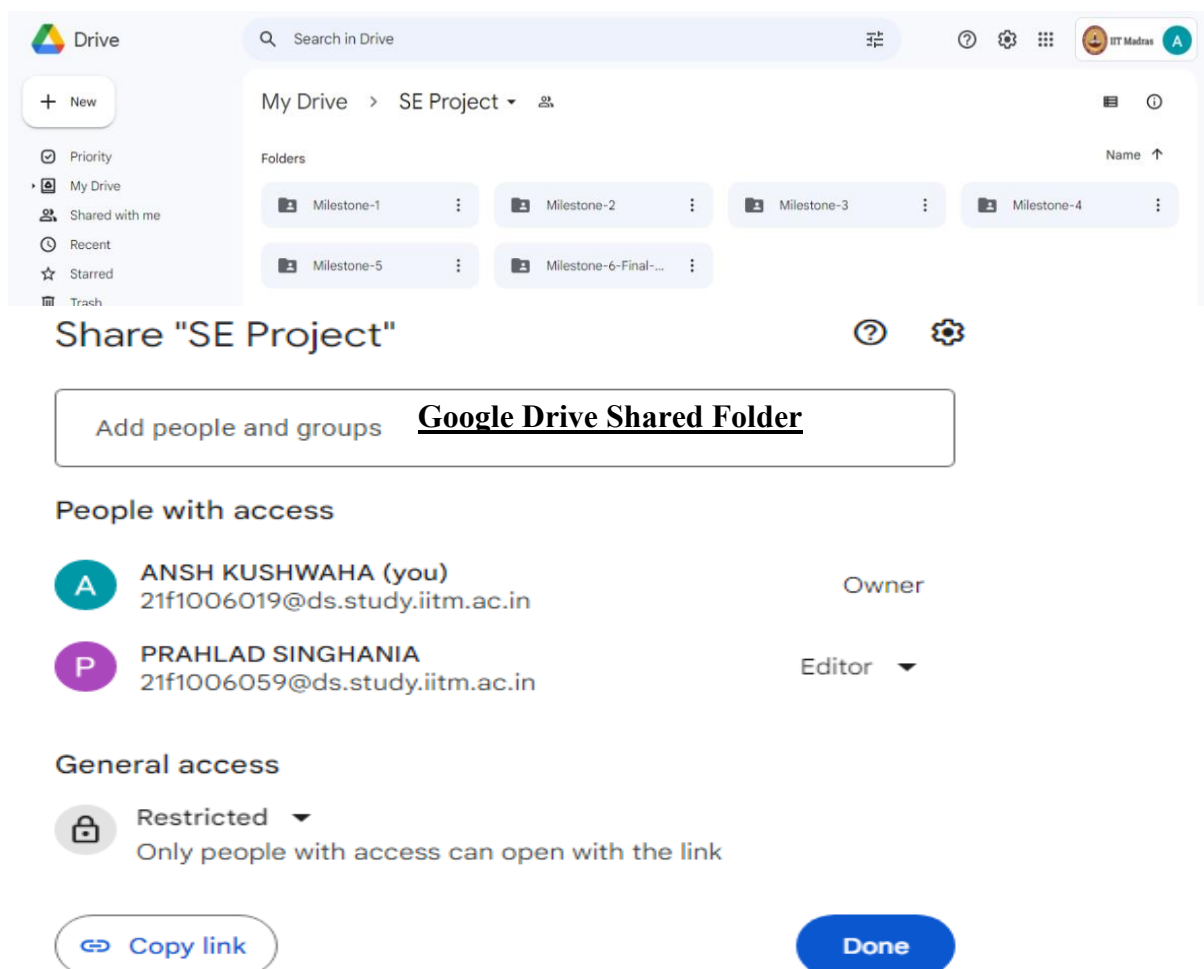
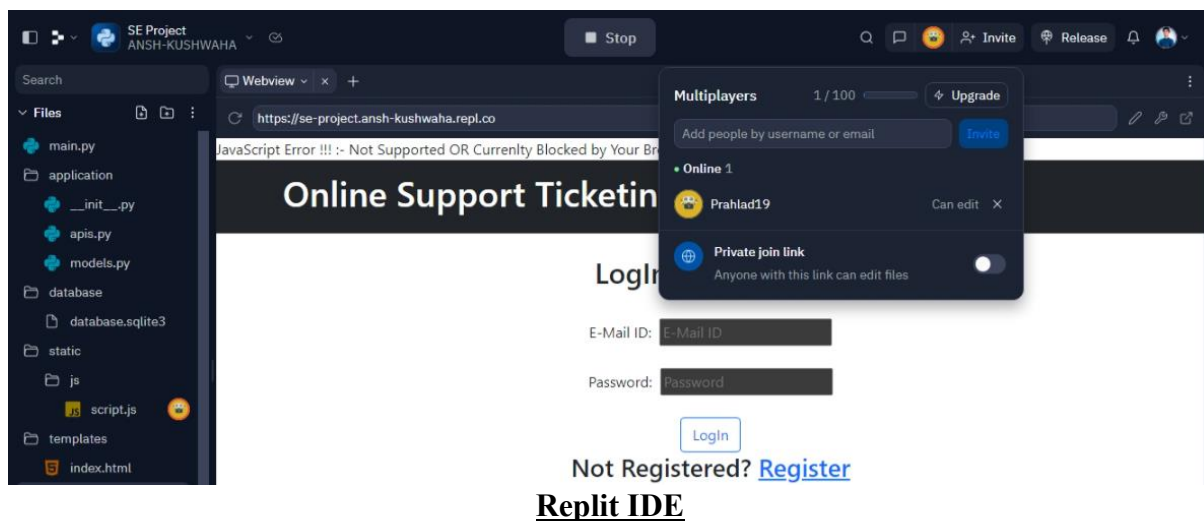

```
deactivate
```

CODE REVIEW, ISSUE REPORTING & TRACKING

For this we have used several platforms:

- ❖ Trello (for Issue Reporting & Tracking)
- ❖ Replit (for Collaborative Development & Testing)
- ❖ Google Drive shared folder (for documents reviewing)
- ❖ Google Meet sessions (for Pair Programming)

Here are the relevant screenshots of the above platforms:



Version history

All versions ▼

TUESDAY

April 11, 7:13 PM

● ANSH KUSHWAHA

THIS MONTH

▶ April 5, 11:32 PM

● ANSH KUSHWAHA

MARCH

March 20, 1:43 AM

● ANSH KUSHWAHA

March 20, 12:24 AM

● ANSH KUSHWAHA

▶ March 5, 8:13 PM

● PRAHLAD SINGHANIA

● ANSH KUSHWAHA

▶ March 5, 4:28 PM

● ANSH KUSHWAHA

▶ March 4, 8:16 PM

● ANSH KUSHWAHA

☒ Show changes

Version history

All versions ▼

FEBRUARY

▶ February 26, 11:11 PM

● PRAHLAD SINGHANIA

▶ February 26, 9:17 PM

● ANSH KUSHWAHA

February 17, 6:11 PM

● ANSH KUSHWAHA

▶ February 16, 6:21 PM

● ANSH KUSHWAHA

▶ February 15, 9:21 PM

● PRAHLAD SINGHANIA

▶ February 12, 9:12 AM

● ANSH KUSHWAHA

▶ February 11, 10:18 PM

● ANSH KUSHWAHA

February 11, 9:55 PM

● ANSH KUSHWAHA

☒ Show changesGoogle Docs Shared File

Google Meet Sessions Log

Monday, February 20, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	27 min	05:15 PM	05:42 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	30 min	05:11 PM	05:42 PM
Sunday, February 26, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	1 hr 9 min	10:40 PM	11:49 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	1 hr 11 min	10:39 PM	11:49 PM
Wednesday, March 01, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	30 min	08:02 PM	08:32 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	38 min	07:54 PM	08:32 PM
Saturday, March 04, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	23 min	06:37 PM	07:00 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	28 min	06:32 PM	07:00 PM
Sunday, March 05, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	29 min	09:59 PM	10:27 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	35 min	09:53 PM	10:27 PM
Wednesday, March 15, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	39 min	06:12 PM	06:51 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	44 min	06:07 PM	06:51 PM
Thursday, March 16, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	23 min	10:12 PM	10:34 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	30 min	10:04 PM	10:34 PM
Sunday, March 19, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	15 min	04:32 PM	04:47 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	17 min	04:30 PM	04:47 PM
Friday, March 24, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	21 min	07:42 PM	08:03 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	24 min	07:39 PM	08:03 PM
Wednesday, April 05, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	1 hr 10 min	11:04 PM	12:14 AM

PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	1 hr 12 min	11:01 PM	12:14 AM
Thursday, April 13, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	1 hr 15 min	02:38 PM	03:58 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	1 hr 26 min	02:37 PM	04:11 PM
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	4 hr 12 min	11:27 PM	03:38 AM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	4 hr 13 min	11:25 PM	03:38 AM
Friday, April 14, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	1 hr 17 min	12:31 PM	01:48 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	1 hr 19 min	12:29 PM	01:48 PM
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	1 hr 42 min	06:57 PM	08:39 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	1 hr 45 min	06:54 PM	08:39 PM
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	4 hr 48 min	09:48 PM	02:36 AM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	4 hr 49 min	09:47 PM	02:36 AM
Saturday, April 15, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	2 hr 3 min	02:48 PM	04:52 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	1 hr 55 min	02:41 PM	04:52 PM
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	1 hr 46 min	06:44 PM	08:30 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	1 hr 50 min	06:40 PM	08:30 PM
Sunday, April 16, 2023					
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	3 hr 15 min	03:35 PM	06:50 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	3 hr 20 min	03:30 PM	06:50 PM
First name	Last name	Email	Duration	Time joined	Time exited
ANSH	KUSHWAHA	21f1006019@ds.study.iitm.ac.in	1 hr 27 min	10:13 PM	11:40 PM
PRAHLAD	SINGHANIA	21f1006059@ds.study.iitm.ac.in	1 hr 32 min	10:08 PM	11:40 PM

THANK YOU